POLITECNICO DI TORINO

Facoltà di Ingegneria

Corso di Laurea in Ingegneria Gestionale

Tesi di Laurea Magistrale

Valutazione dei Non Performing Loan con Machine Learning:

Caso di studio Gruppo Intesa Sanpaolo

Relatore:

Franco Varetto

Candidato:

Ramiro R. Ruiz Alvarez

Luglio 2020

A Roberto Trabucco e Ana Maria Alvarez,

che non smetterò mai di ammirare e ringraziare

Abstract

I due argomenti principali trattati e combinati in questa tesi sono stati protagonisti, a modo loro, dell'ultimo decennio. Da una parte si parla della valutazione dei *Non-Performing Loan* negli intermediari finanziari, e quindi l'utilizzo della statistica per misurare il livello di rischio che comporta intraprendere dei rapporti con un determinato cliente; dall'altra si parla di *Machine Learning*, sotto-argomento del più vasto mondo dell'Intelligenza Artificiale, ovvero quell'insieme di algoritmi che apprendono automaticamente da un ammontare grande di dati in input per svolgere una determinata attività e fornire un desiderato output.

Lo scopo principale dell'elaborato è confrontare i modelli predittivi che vengono ad oggi impiegati in molti istituti bancari e provenienti dalla statistica "classica" (regressioni semplici, regressioni multivariate, ecc) con alcuni modelli predittivi proposti dal Machine Learning, per stimare la variabile target *Loss Given Default*. Questa misura di rischio indica la percentuale di perdita che subisce l'intermediario finanziario nel caso la controparte vada in default e rappresenta la variabile fondamentale nel caso di controparti già in default per le quali bisogna quantificare la perdita che ci si attende. Gli algoritmi di machine learning utilizzati sono: Decision Tree, Random Forest, Support Vector Machine e Rete Neurale. In questo intento è stato utilizzato un dataset di controparti in default fornito dal *Risk Management* del Gruppo Intesa Sanpaolo.

Nel Capitolo 1 vengono dunque introdotti i concetti basici del Rischio di Credito e le varie componenti che servono a quantificarlo facendo un ripasso delle varie metodologie comunemente utilizzate. Nel Capitolo 2 invece ci si focalizza sui Non Performing Loan e si cerca di contestualizzare la situazione pregressa ed attuale nel contesto principalmente italiano, prima di introdurre le metodologie che il gruppo Intesa Sanpaolo adotta per il calcolo della Loss Given Default.

Nel Capitolo 3 viene introdotta la teoria alla base delle metodologie utilizzate per la preparazione dei dati, della selezione delle variabili, degli algoritmi utilizzati e dell'ottimizzazione dei parametri di ognuno di essi. Nel Capitolo 4 vengono mostrati gli sforzi fatti per la scelta e l'elaborazione del set di variabili, i risultati dei singoli algoritmi partendo da una versione di "*default*" e la progressiva evoluzione delle performance con l'ottimizzazione sia del set di variabili utilizzate, sia dei parametri degli algoritmi. Parallelamente i risultati vengono confrontati sia tra gli algoritmi stessi, sia con l'algoritmo semplice di regressione multivariata in uso preso il Gruppo. Infine, si cerca di introdurre due algoritmi composti che combinino i risultati dei singoli per fornire una nuova stima.

I risultati evidenziano come tutti gli algoritmi di machine learning scelti performino meglio della regressione multivariata. In particolare, il Random Forest ottiene miglioramenti nell'accuratezza significativi.

Indice

Parte I: Rischio di Credito e i Non Performing Loan

Capitolo 1: Il Rischio di credito

1.1 Introduzione al Rischio di Credito

Fra tutti i rischi a cui gli istituti finanziari fanno fronte, il rischio di credito è senza ombra di dubbio il più comune ed intrinseco dell'attività di intermediazione finanziaria. Fin dalla nascita medievale di quel che oggi si riconosce in un istituto bancario, questo rischio è stato individuato e gestito con diverse metodologie che si sono evolute nel corso del tempo, e più recentemente, anche in parallelo al mercato finanziario. Principalmente a partire dagli anni successivi alla fine del secondo e più grande conflitto bellico mondiale, si ebbe uno sviluppo dei mercati finanziari senza precedenti grazie anche all'introduzione di svariati strumenti come, ad esempio, i derivati. Con oggetti così sofisticati, che rendevano il mercato ancora più vario, complesso e dinamico, lo studio, la misurazione e la gestione del rischio di credito diventò talmente importante che nacquero apposite direzioni funzionali negli istituti finanziari: quel che oggi si definisce *Risk Management*. Furono le crisi, durante gli anni '90 del secolo scorso e la più recente del 2007-08, a sollevare l'evidente bisogno di una maggior attenzione da parte delle autorità di Vigilanza. La gestione del rischio diventa quindi non soltanto un bisogno dell'istituto bancario per salvaguardare i propri profitti ma un sollecito esterno da parte delle autorità per salvaguardare l'intero sistema economico.

Ma cosa si intende, in senso moderno, con rischio di credito?

1.2 Definizione e classificazione del rischio di credito

La definizione classica prevede l'impiego del termine *rischio* soltanto nella sua accezione negativa, ovvero di "*perdita*" e mai di "*guadagno*", dunque si intende la probabilità che emesso un credito il debitore si trovi nella condizione di non riuscire a ripagare completamente o parzialmente quanto dovuto. Per essere precisi si riporta la definizione di A. Sironi[1]:

"Il rischio di credito rappresenta la possibilità che una variazione inattesa del merito creditizio di una controparte generi una corrispondente variazione inattesa del valore corrente della relativa esposizione creditizia":

Da questa definizione si capisce come il rischio di credito non riguardi soltanto le controparti insolventi ma anche quelle per cui si vanta un credito *deteriorato*, ovvero di qualità inferiore rispetto alle condizioni iniziali. Il valore di un prestito viene calcolato tenendo conto dei flussi di cassa futuri opportunamente attualizzati. Il tasso di attualizzazione coinvolto è formato da una componente *risk-free* e da uno *spread* che rispecchia il rischio di insolvenza della controparte. Per diversi motivi, il rischio di insolvenza può crescere aumentando di conseguenza lo spread (o premio per il rischio) che rende la posizione, ad attualizzazione fatta, di valor inferiore.

Volendo fare un esempio scolastico supponiamo che la banca abbia fornito un prestito ad un cliente di 150 mila euro e che il cliente lo debba ripagare entro 3 anni con cedole uguali di 50 mila euro. Il tasso risk free è l'1% mentre il premio per il rischio di insolvenza del cliente è del 6% inizialmente e del 9% successivamente. Utilizzando le seguenti formule:

$$R_{cliente} = R_{riskfree} + PremioRischio \tag{1.1}$$

[1] [41] Resti A. & Sironi A., (2008), Rischio e valore nelle banche, Milano, Egea, p. 351.

$$Valore\ Attuale = \sum_{t}^{T} [\frac{FlussoCassa_t}{(1 + R_{cliente})^t}] \qquad (1.2)$$

Si riescono a confrontare i due scenari ed a sottolineare la perdita dovuta al deterioramento del credito concesso:

Tabella 1.1: Esempio di deterioramento del credito

Scenari	Flusso Attualizzato 1	Flusso Attualizzato 2	Flusso Attualizzato 3	Valore attuale totale
Rischio cliente 7%	93,45	87,34	81,62	262,43
Rischio cliente 10%	90,90	82,64	75,13	248,68
Perdita di valore				-13,74

(Valori in migliaia di euro)

Bisogna puntualizzare che si parla di rischio di credito quando questo è inatteso. Gli istituti finanziari monitorano costantemente le variazioni attese delle condizioni economico-finanziarie delle varie controparti ma il vero rischio deriva dalle variazioni che sfuggono completamente a queste previsioni.

Per i clienti insolventi il credito viene recuperato seguendo diversi metodi che variano in base alle caratteristiche di questo. Generalmente si apre una pratica legale a cui sussegue un dedicato processo di recupero. La banca, ex-ante l'insolvenza, calcola una stima del credito recuperabile in caso di fallimento della controparte che spesso non coincide con quanto effettivamente si recupera ex-post. Questo delta di recupero, in senso negativo, rappresenta una terza componente del rischio di credito, ovvero la probabilità di non recuperare quanto previsto.

L'esposizione stessa risulta, per alcuni strumenti finanziari, componente del rischio di credito in quanto soggetta ad una certa discrezionalità della controparte. In questi casi, spesso succede che momenti prima dell'avvento dell'insolvenza l'esposizione aumenti di molto le sue dimensioni nella speranza di evitare il default ma creando di conseguenza una maggior perdita nel caso questo si verifichi.

Per quegli strumenti che hanno un mercato secondario, ad esempio le obbligazioni, si potrebbe verificare anche una crisi di mercato che si rifletterebbe in un aumento del premio per il rischio richiesto. Anche questo fattore, denominato rischio di spread, va tenuto in conto.

Per sintetizzare quanto detto si riprende la classificazione fatta da Sironi[1]:

Tabella 1.2: Tipologie di rischio di credito

Tipo Rischio	Manifestazione	Cause	Esposizioni
Insolvenza	Incremento di controparti insolventi.	Diminuzione della crescita economica, incremento dei tassi di interesse	Tutte
Migrazione	Deterioramento del credito	Diminuzione della crescita economica, incremento dei tassi di interesse	Tutte
Recupero	Recupero inferiore a quanto previsto dal creditore in caso d'insolvenza	Aumento dei tassi di interesse, diminuzione del valore delle attività reali	Tutte
Esposizione	Aumento dell'esposizione a rischio	Multiaffidamento se la banca si muove in ritardo rispetto alle altre	Esposizioni per le quali il debitore gode di discrezionalità
Spread	Aumento del rischio richiesti dal mercato	Crisi di mercato e/o aumento avversione al rischio degli investitori	Titoli obbligazionari e attività aventi un mercato secondario

Sironi parla anche di rischi di Pre-Regolamento e di Regolamento che hanno natura del tutto simile al rischio di insolvenza principalmente per quel che riguarda strumenti derivati. Si intende infatti il rischio che la controparte

[1] [41]

si rilevi inadempiente prima della data di regolamento delle prestazioni o della data prevista dal regolamento.

1.3 Componenti del rischio di credito

Come già accennato, il rischio di credito può essere suddiviso in due principali componenti: la *Perdita Attesa* (*EL - Expected Loss*) e la *Perdita Inattesa* (*UL - Unexpected Loss*). È importante notare che la prima componente non può essere eliminata con la diversificazione del portafoglio mentre la seconda può invece essere sensibilmente ridotta.

1.1.1 Expected Loss

L'*Expected Loss* risulta essere la perdita che l'istituto finanziario si attende dalla controparte ed è composta a sua volta da altre tre importanti componenti:

- Probabilità di insolvenza (*PD - Probability of Default*)
- Il tasso di perdita atteso in caso di Default (*LGD - Loss Given Default*) oppure, il complemento del tasso di recupero ((1- *RR) - Recovery Rate*)
- L'esposizione vantata nei confronti del debitore (*AE - Adjusted Exposure*)

La perdita attesa si esplicita analiticamente con questi tre fattori nel seguente modo:

$$EL = AE \cdot PD \cdot LGD \qquad (1.3)$$

Si analizzano in modo sintetico e singolarmente ognuna di queste componenti.

1.1.1.1 Probability of Default

Essendo l'evento d'insolvenza del cliente il fattore scatenante di una situazione molto grave, la letteratura si è concentrata maggiormente sullo studio di metodi e algoritmi sempre più adatti e precisi nella stima della PD. Generalmente si utilizzano tre diversi approcci (**Figura 1.1**):

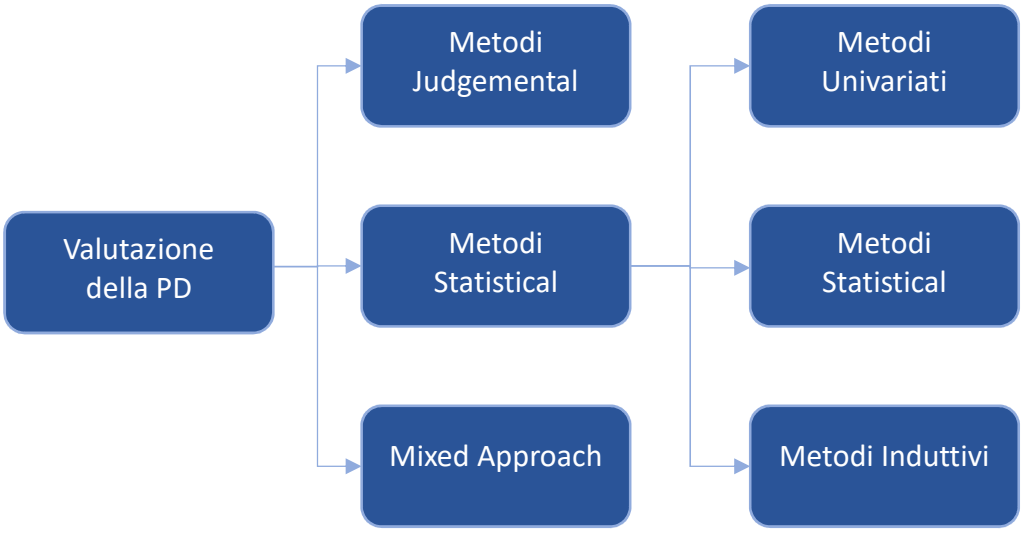


Figura 1.1: Metodi di calcolo della PD

1) *Judgemental*: Ci si basa sull'opinione di un esperto che analizza a fondo le caratteristiche del cliente e le confronta con la sua esperienza pregressa. Questo approccio viene utilizzato spesso per clienti *Corporate*, ovvero con esposizioni al di sopra di una soglia predefinita di capitale.
2) *Statistical*: Sono metodi statistici, detti di "*scoring*", nei quali si cerca di assegnare un certo *score* al cliente analizzando le caratteristiche ritenute importanti (solitamente quelle economico-finanziare). Si parla generalmente di:

a) *Modelli Univariati*: si cerca di costruire una funzione discriminante fra cliente sano e cliente non sano sulla base di un unico indicatore;
b) *Modelli Multivariati*: a differenza dei modelli univariati si considerano più variabili che meglio segmentano la tipologia di cliente. Fra i più utilizzati ci sono i modelli dello *Z-Score*[1] e la *Regressione Logistica*.
c) *Modelli Induttivi*: si parla principalmente di tutti quei modelli di intelligenza artificiale che si prestano, con ottimi risultati, al calcolo della PD. Questi modelli hanno bisogno di un continuo training dei loro parametri per affinare nel tempo le loro previsioni. Esempi di questi sono: le Reti Neurali, Support Vector Machine, Random Forest, ecc.

3) *Mixed-approach:* Un approccio misto fra i primi due. Si parte dalle analisi statistiche e si corregge il valore della PD secondo le considerazioni in merito di un esperto.

1.1.1.2 Loss Given Default

La stima della LGD ha un grado superiore di difficoltà rispetto a quella della PD per alcuni intuitivi aspetti. Innanzitutto, il numero di LGD osservate è relativamente basso in quanto si tratta soltanto di quel sottoinsieme di aziende effettivamente fallite. Inoltre, mentre nel calcolo della PD si ragiona osservando una variabile binaria che prevede dunque soltanto due scenari, default e non-default, nel calcolo della LGD abbiamo una variabile continua in quanto la percentuale recuperabile può variare da 0 ad 1.

Come accennato precedentemente la stima della LGD in realtà viene fatta spesso attraverso il suo complemento, ovvero il *Recovery Rate* (RR) che altro non è che il tasso atteso di recupero sul credito vantato

[1] Si Veda E. Altman,1968

in caso di insolvenza della controparte. Questo parametro è difficilmente noto al momento della concessione e spesso neppure al momento del verificarsi dell'insolvenza del debitore.

Notoriamente il tasso di recupero dipende da fattori quali:

- Caratteristiche del cliente (settore economico, posizione geografica, condizioni giuridico-politiche del paese in cui risiede il cliente, ecc)
- Caratteristiche della banca (efficienza dei processi di recupero crediti, efficienza dei servizi legali interni e/o delegati, ecc)
- Caratteristiche del finanziamento (accompagnato da garanzie o meno, grado di liquidità delle garanzie, la seniority vantata sul cliente rispetto ad eventuali altri istituti finanziari, ecc)
- Fattori Esogeni (Ciclo Economico, livello dei tassi di interesse, ecc)

Come analizzato e suggerito da Querci[1] (2007) il calcolo della LGD può esser diviso in due macro-insiemi:

- LGD calcolata ex-ante l'evento di insolvenza (come stima nel caso in cui questo capitasse)
- LGD calcolata ex-post l'evento d'insolvenza

È facile intuire che mentre per il secondo insieme si possono sviluppare dei calcoli effettivi basati sull'esperienza di clienti realmente insolventi, nel primo ci si deve accontentare di una stima basata sul comportamento prima del passaggio in stato di default di clienti con caratteristiche simili a quello analizzato. Di conseguenza, la variabilità del primo caso è superiore di molto a quella del secondo.

Più genericamente, i metodi di stima della LGD si possono sintetizzare secondo lo schema in **Figura1.2**.

[1] [5] Querci F. (2007). Rischio di credito e valutazione della loss given default. Bancaria Editrice, Roma

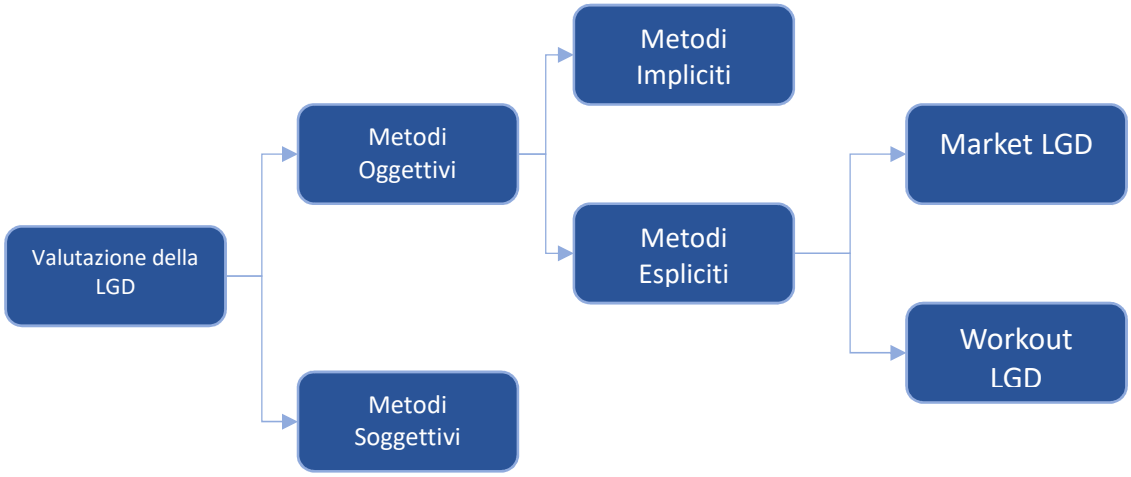


Figura 1.2: Metodi di calcolo della LGD

La valutazione può essere:

- *Soggettiva*: basata sull'opinione di un esperto;
- *Oggettiva*: basata su dati numerici e statistiche.

La *Valutazione Oggettiva* inoltre si suddivide in:

- *Metodi Impliciti*: si assume che il tasso di recupero sia implicitamente contenuto nel prezzo di un titolo obbligazionario emesso dalle imprese. La differenza fra il prezzo del titolo *risk free*, che potrebbe essere un titolo di stato, ed il prezzo di un'obbligazione *corporate* dipende anche dalla perdita attesa, ovvero dal rischio che si associa al bond corporate emesso. La perdita attesa altro non è, come già visto precedentemente, che la moltiplicazione fra PD e LGD. Stimando la PD, la LGD diventa implicita e ricavabile di conseguenza[1].
- *Metodi Espliciti*: La LGD viene stimata sulla base di dati riguardanti controparti insolute.

La *Valutazione Oggettiva Esplicita* si suddivide a sua volta in:

[1] Si vedano [16] e [17].

- *Market LGD:* Nel caso esista un mercato secondario dei crediti deteriorati, si possono registrare i prezzi di vendita delle posizioni insolute (*price at default*) ed elaborare opportune statistiche sulla base di queste informazioni. In Italia non esiste un mercato secondario dei crediti deteriorati sufficientemente liquido tale da rendere attraente questa opzione.
- *Workout LGD:* La LGD viene calcolata come il complemento del rapporto fra il valore attuale dei flussi di cassa netti attualizzati (recuperi – perdite) e l'ammontare dell'esposizione creditizia al momento del default:

$$LGD = 1 - \frac{(\sum_i R_i - \sum_i C_i)}{Esposizione} + CI \tag{1.4}$$

Con:

- R_i i recuperi attualizzati;
- C_i sono le perdite durante il processo di recupero;
- CI sono eventuali costi indiretti.

1.1.1.3 Adjusted Exposure

L'esposizione è, nella maggior parte dei casi, fissa e non soggetta a variazioni. Ci sono tuttavia alcune situazioni per le quali, soprattutto prossimi all'evento di default, questa esposizione varia notevolmente. Un esempio di questo fenomeno si verifica quando la controparte della banca ha a disposizione, nel proprio conto corrente, un fido. È intuitivo immaginare come in una situazione prossima al fallimento, con l'obbiettivo di evitarlo, la controparte cerchi di usufruire di tutte le risorse disponibili arrivando a toccare facilmente i massimali del fido concesso. Pertanto, il creditore non saprà con certezza l'esposizione fino al momento di insolvenza. Tuttavia,

possono essere fatte delle stime aggiungendo una quota aleatoria e comporre l'esposizione come evidenziato di seguito:

$$AE = Quota\ U + Quota\ NU \cdot UGD \tag{1.5}$$

Con:

- *Quota U*, quota parte del fido utilizzata al momento della stima;
- *Quota NU*, la differenza fra il massimale previsto dal contratto con la controparte e la quota già utilizzata;
- *UGD (Usage Given Default):* È la percentuale del credito a disposizione non ancora usato dalla controparte che si ritiene verrà usata.

1.1.2 Unexpected Loss

L'altra componente del rischio di credito rappresenta il vero e proprio rischio in quanto cerca di ricavare la variabilità che la perdita attesa ha intorno al suo valore medio dando di conseguenza una misura dell'affidabilità dello stesso.

La forma analitica più elementale della *UL*, in caso la *AE* e la *LGD* siano deterministiche, è funzione della probabilità di default *PD*:

$$UL = LGD \cdot \sqrt{PD \cdot (1 - PD)} \tag{1.6}$$

Se invece si considerasse la variabilità della *LGD,* trattandosi ora della variazione standard del prodotto di due variabili aleatorie, la *UL* assumerebbe la forma:

$$UL = \sqrt{PD \cdot (1 - PD) \cdot LGD^2 + PD^2 \cdot \sigma_{LGD}^2 + PD \cdot (1 - PD) \cdot \sigma_{LGD}^2} \tag{1.7}$$

Con σ_{LGD}^2 la varianza della LGD.

La correlazione tra le variabili *PD* e *LGD* è stata studiata e non tralasciata. Tuttavia, la scelta dei modelli di stima impatta fortemente sul valore della correlazione per cui si preferisce spesso stimare le variabili per separato come se risultassero indipendenti. L'intuizione economica che una correlazione esista deriva dal fatto che sia la PD sia la LGD sono correlate al ciclo economico. Essendo correlate quindi allo stesso fenomeno di rischio sistematico, si ipotizza che lo siano anche fra di loro. Nello specifico ci si aspetta una correlazione positiva: in situazione di recessione economica la probabilità di default aumenta cosi come la percentuale non recuperabile di credito.

Capitolo 2: Non performing loans (NPL)

2.1 Definizione ed introduzione ai NPL

I crediti deteriorati, ovvero i Non Performing Loans (NPL), sono esposizioni verso soggetti che, a causa di un peggioramento della loro situazione economica e finanziaria, non sono in grado di adempiere in tutto o in parte alle proprie obbligazioni contrattuali.

I NPL sono stati protagonisti del settore bancario italiano (e non solo) quasi per un decennio, a partire dalla crisi economica del 2007/08. La forte contrazione dell'economia (quasi dieci punti di PIL[1] e circa un quarto di produzione industriale, contro il -5,7% del PIL e il -19% della produzione industriale nell'area dell'euro)[2] dovuta alla crisi, ha colpito fortemente il sistema bancario italiano, il quale ha accumulato, anno dopo anno, un ammontare sempre più elevato di crediti deteriorati registrando un picco di massima nel 2015. Si pensi che il rapporto fra i NPL ed il totale dei crediti erogati, detto *NPL ratio*, aveva raggiunto il 15-17%, per le banche più solide, e oltre il 35% per le banche più deboli[3]. A partire da quest'anno, grazie soprattutto a operazioni massive di cessione, il livello di NPL è diminuito progressivamente.

Nel grafico di **Figura 2.1**, creato da Banca Ifis su dati di Banca d'Italia, si evince l'andamento dell'ammontare dei NPL in Italia dal 2010 fino a fine 2019. Si nota come le Sofferenze siano sempre in maggioranza fra i NPL.

[1] Prodotto Interno Lordo - PIL
[2] [6] www.bancaditalia.it
[3] [7] C. Barbagallo, *"I crediti deteriorati delle banche italiane: problematiche e tendenze recenti",* Roma, 6 giugno 2017

Inoltre, pur avendo ridotto l'ammontare totale di più del 50% rispetto al picco del 2015, il valore dei crediti deteriorati resta comunque alto.

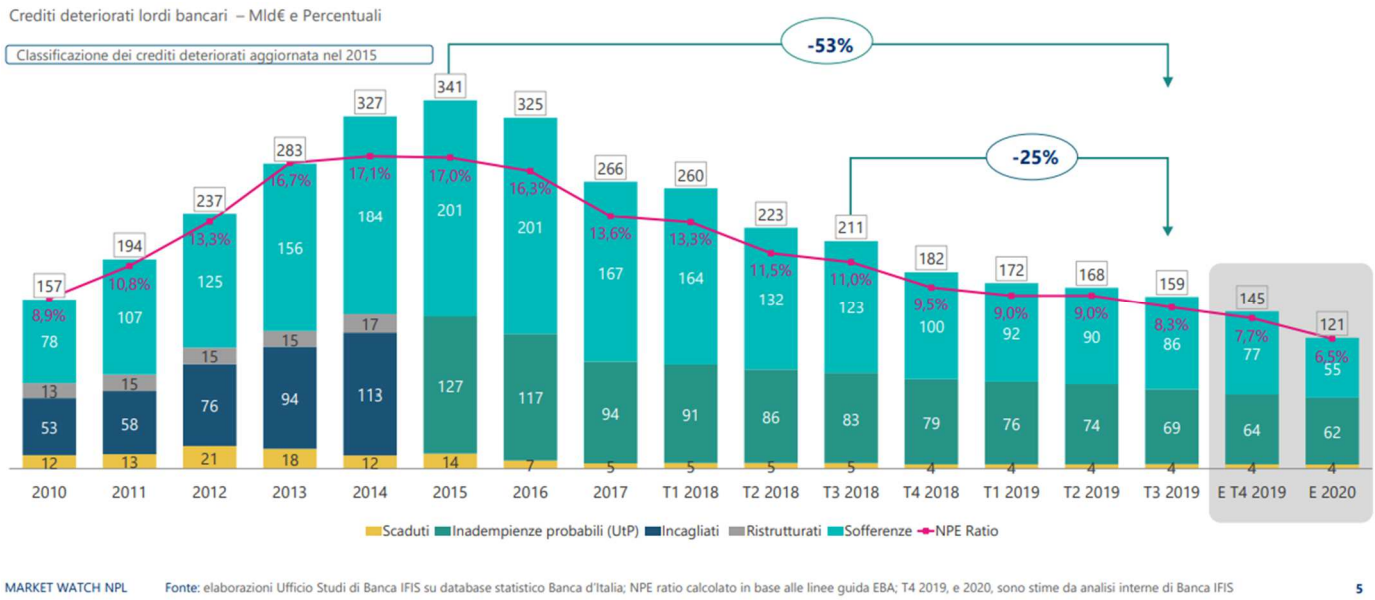


Figura 2.1: Andamento NPL Ratio, realizzato da Banca Ifis[1]

2.1.1 Impatti dei NPL sugli Intermediari Finanziari

Esattamente come la presenza di crediti vantati in una banca è sinonimo di guadagno (attraverso i tassi di interesse), la presenza di crediti deteriorati è sinonimo di perdita. I costi derivanti sono molteplici, tra gli altri si possono contare: le spese legali da sostenere per le procedure di recupero, il costo del personale e delle risorse che occupano della gestione, la perdita stessa del mancato pagamento, i costi indiretti derivanti dagli impatti negativi sull'immagine.

La presenza di elevati stock di NPL potrebbe essere anche letto come un indice esplicito di inefficienza, sia in termini di valutazione in fase di concessione dei prestiti, sia di recupero crediti. Devono essere stanziati appositi fondi per coprire il rischio che i NPL rappresentano, sia per volontà propria sia per l'obbligo normativo di mantenere un sufficientemente alto

[1] [8] https://www.bancaifis.it

patrimonio di vigilanza[1], il che si traduce in una riduzione dei profitti ed una mancata opportunità di reinvestimento. La somma di questi aspetti portano gli investitori a diffidare delle performance dell'intermediario in questione ed a innalzare sensibilmente il costo di finanziamento sostenuto da quest'ultimo (o il premio per il rischio richiesto dai primi).

È stato provato[2] come un alto livello di crediti deteriorati modifichi il comportamento dell'intermediario che a sua volta diffiderà maggiormente nella concessione di crediti (*credit crunch*). La sua avversione al rischio limita la sua possibilità di espansione e la capacità di cogliere opportunità di investimento convenienti. È dunque intuitivo capire come sia altamente prioritario, per la banca, minimizzare l'ammontare dei propri NPL.

2.1.2 Cause iniziali dell'incremento di NPL in Italia

Sono stati sicuramente tre i fattori principali a causare l'aumento dei NPL in Italia nell'ultimo decennio:

- la recessione economica (dovuta all'estensione della crisi finanziaria del 2007/2008);
- l'estrema lentezza delle procedure di recupero dei crediti;
- l'assenza di un mercato secondario di NPL.

Inoltre, politiche creditizie imprudenti, un'eccesiva tolleranza nei confronti dei debitori (dai profili di rischio talvolta dubbiosi), erogazioni in conflitto di interessi o apertamente fraudolente costituirono altrettante aggravanti[3].

---

[1] Obbligo normativo derivante dagli accordi di Basilea, di cui l'ultimo, "*Basilea 3 – Schema di regolamentazione internazionale per il rafforzamento delle banche e dei sistemi bancari*", è reperibile all'indirizzo web specificato in [42]

[2] [9] Doriana Cucinelli, "*The Impact of Non-performing Loans on Bank Lending Behavior: Evidence from the Italian Banking Sector",* Eurasian Journal of Business and Economics, 2015

[3] [7] Carmelo Barbagallo ,"I crediti deteriorati delle banche italiane: problematiche e tendenze recenti", Roma, 6 giugno 2017

La capacità di smaltire le procedure di recupero crediti in Italia era dimezzata rispetto al resto dei paesi europei. In altre parole, i tempi di recupero crediti erano il doppio (a sua volta un riflesso dei tempi lunghi in generale delle procedure civili in Italia), il che significava che lo stock dei NPL tendeva ad assumere dimensioni importanti se confrontati con gli altri stati europei. Tra il 2014 e 2017 il tempo medio di recupero era di 8,5 anni[1].

La problematica non derivava soltanto dalle procedure "fuori banca", ma anche da procedure interne. L'attenzione dedicata dagli intermediari finanziari al recupero credito era inferiore di molto se confrontata con l'attenzione dedicata ad altre operatività. Il riflesso empirico di questo fatto era visibile nello scarso quantitativo di personale predisposto, nelle altrettante scarse risorse materiali, nel basso livello di informatizzazione (le pratiche erano gestite principalmente in formato cartaceo) e nella mancanza di opportuni database che strutturassero in modo opportuno le informazioni. Operando "*manualmente*", un aumento improvviso delle pratiche per un già ridotto personale a disposizione, ha generato uno stock di queste anche all'interno delle banche. Questa manualità aveva anche l'aggravante di non rendere facilmente possibile una *prioritizzazione* delle pratiche. Algoritmi applicati in tempi recenti hanno evidenziato come si riesce a costruire un elenco di pratiche da *prioritizzare* permettendo così ai gestori di trattare per primi quei clienti con esposizioni rilevanti per i quali è previsto, in base alle proprie caratteristiche, di ottenere una maggior percentuale dei crediti vantati. Ci si focalizza quindi sulle controparti per le quali si ha un potenziale di recupero superiore evitando di disperdere energie. Per il gestore che lavorava con formati cartacei, questo tipo di *prioritizzazione* non era facilmente fattibile.

Il fatto stesso che esistano nel mercato operatori specializzati nel recupero credito con rendimenti rilevanti, è un'evidenza almeno parziale dell'inefficienza delle procedure di recupero credito nelle banche. La quasi

[1] [10] Paolo Angelini, *"I crediti deteriorati: mercato, regole e rafforzamento del sistema*", Intervento del Vice Capo del Dipartimento Vigilanza bancaria e finanziaria della Banca d'Italia, Roma, 9 ottobre 2018

totale assenza di questi operatori nel mercato italiano era un'altra motivazione degli alti livelli di stock di NPL in Italia. Pur non trattandosi di una causa diretta di *genesi* di NPL, il non avere un mercato sviluppato di compra/vendita di crediti deteriorati era sicuramente un ostacolo in più nello smaltimento di questi, accrescendo quindi lo stock presente ed accrescendo il *gap* con altri paesi nei quali esisteva tale mercato.

Il motivo principale per il quale in Italia non esisteva un mercato secondario di NPL sufficientemente sviluppato va ricercato nel cosiddetto *bid-ask spread*. Si parla infatti della differenza fra il prezzo al quale le banche erano disposte a vendere i loro crediti deteriorati ed il prezzo al quale gli operatori di mercato specializzati in recupero credito erano disposti a comprare. A sua volta, il motivo di questa differenza nei due prezzi, di offerta e domanda, derivava dai diversi criteri di valutazione utilizzati dalle banche per scrivere il valore a bilancio di questi crediti e i criteri utilizzati dagli investitori potenziali. La differenza dei criteri si può sintetizzare nei seguenti due punti[1]:

1- Essendo dei crediti ad alto rischio di rimborso, il tasso di rendimento richiesto dagli investitori risultava molto più elevato rispetto al tasso utilizzato dalle banche per le scritture contabili (nel rispetto dei principi contabili IAS/IFRS allora seguiti), ovvero il tasso effettivo originario su questi attivi (molto più basso). Nel valutare i flussi futuri di incasso, utilizzando un tasso di rendimento maggiore rispetto alle banche, gli investitori si trovavano ad avere un valore attuale molto più basso rispetto a quanto calcolato nei bilanci bancari;
2- I costi indiretti della gestione degli NPL venivano considerati dalle banche, nel rispetto dei principi contabili, nell'esercizio di competenza. Gli investitori invece deducevano complessivamente questo importo immediatamente riducendo ulteriormente il valore netto e quindi il prezzo di acquisto.

[1] [19] L. G. Ciavoliello, F. Ciocchetta, F. M. Conti I. Guida, A. Rendina, G. Santini, "Quanto valgono i crediti deteriorati", Note di stabilità Finanziaria N. 3 aprile 2016

Per ovvi motivi, risulta evidente come il prolungato tempo di recupero aggravi ulteriormente questa situazione. I costi, diretti e indiretti, di gestione sono maggiori se il tempo di recupero è maggiore. Allo stesso modo, tempi maggiori di recupero implicano una maggior probabilità di non recuperare affatto i crediti o un deterioramento del valore attuale (anche solo tenendo conto l'inflazione), il che si riflette in un maggior tasso di rendimento (o di rischio) richiesto dall'investitore.

2.1.3 Cause della recente riduzione dei NPL in Italia

La riduzione visibile dei NPL nella **Figura 2.1** dopo il 2015 deriva in primis da una forte reazione legislativa in Italia. Fra i vari interventi si elencano:

- Interventi normativi sul regime fiscale applicato alle perdite sui crediti delle banche. È stata rivista la tassazione delle perdite su crediti delle banche, attenuandone la pro-ciclicità, incentivando l'adozione di politiche di valutazione dei crediti più prudenti e contribuendo alla trasparenza dei bilanci bancari[1];
- Riforme della legge fallimentare[2] e del codice di procedura civile[3] approvate nel 2015 e 2016, volte a ridurre i tempi e accrescere l'efficacia delle procedure concorsuali ed esecutive;
- Provvedimento che istituisce un meccanismo di garanzia statale [4](la "*GACS*") sulle operazioni di cartolarizzazione dei crediti deteriorati;

[1] [18] A. De Vincenzo, G. Ricotti,*" L'utilizzo della fiscalità in chiave macroprudenziale: l'impatto di alcune recenti misure tributarie sulla prociclicità e sulla stabilità delle banche*", Note di stabilità finanziaria e vigilanza N. 1, Aprile 2014
[2] [43] M. Marcucci, A. Pischedda, V. Profeta*, "The changes of the Italian insolvency and foreclosure regulation adopted in 2015"*, Notes on Financial Stability and Supervision No. 2, November 2015
[3] [44] E. Brodi, S. Giacomelli, I. Guida, M. Marcucci, A. Pischedda, V. Profeta, G. Santini, *"Nuove misure per velocizzare il recupero dei crediti: una prima analisi del D.L. 59/2016",* Note di stabilità finanziaria e vigilanza N. 4, Agosto 2016
[4] Si veda [11]

- Dal 2016 la Banca d'Italia richiede alle banche di compilare una segnalazione statistica contenente dati molto dettagliati sulle singole posizioni in sofferenza[1];

Inoltre, nel marzo 2017 la Banca Centrale Europea ha pubblicato le "*Linee guida per le banche sui crediti deteriorati (NPL)*"[2] con lo scopo di individuare le *best practices*, di monitorare e presidiare costantemente gli sviluppi della gestione dei NPL, di promuovere una maggiore tempestività di accantonamenti e cancellazioni.

Un altro cambiamento sostanziale per il sistema bancario italiano è costituito dall'entrata in vigore, dal 01 gennaio 2018, del nuovo principio contabile internazionale *IFRS9*, che va a sostituire lo *IAS39*. Fra le tante novità, sono principalmente due quelle che maggiormente interessarono i crediti deteriorati:

- Nuova classificazione e misurazione degli strumenti finanziari;
- Nuovo modello di svalutazione dei crediti (*impairment*);

Nella nuova classificazione bisogna considerare congiuntamente sia il *business model* adottato dalla banca nella gestione degli strumenti finanziari, andando a capire se le attività finanziarie sono detenute per incassare e/o per vendere, sia delle caratteristiche contrattuali dei flussi di cassa dei singoli asset utilizzando il SPPI test (*Soley payment of principal and interest test*). Gli asset possono essere dunque classificati nelle seguenti categorie:

A) *Attività finanziarie detenute al fair value con impatto a Conto Economico*: Si tratta di strumenti trattenuti principalmente con finalità di trading, strumenti per i quali si è già deciso si esercitare la *fair value option* e tutti gli altri strumenti finanziari che non sono classificate nelle successive categorie;

[1] Si veda [12]
[2] Si veda [13]

B) *Attività finanziarie valutate al fair value con impatto sulla redditività complessiva*: Questa valutazione viene fatta al fair value ed ha impatto sulla voce del patrimonio netto "*Riserve di valutazione*". Si parla di attività finanziarie: possedute nell'ambito di un *business model* il cui obiettivo è conseguito sia incassando i flussi di cassa che cedendo l'attività; i cui termini contrattuali danno origine, per specifiche date, a flussi di cassa derivanti da rimborsi di capitale nominale e da interessi calcolati in relazione all'ammontare del valore nominale residuo.

C) Attività finanziarie valutate al costo ammortizzato: Si tratta di attività che devono rispettare le stesse condizioni della lettera precedente con la differenza che la valutazione contabile non viene fatta con il *fair value* ma al *costo ammortizzato*.

Il nuovo modello di svalutazione dei crediti prevede per gli strumenti finanziari un nuovo sistema di calcolo per definire le rettifiche di valore sui crediti (*impairment*), in relazione al relativo peggioramento della loro qualità creditizia. Visto che il modello precedente, detto *incurred loss model*, basato sulle perdite subite, non ha correttamente perseguito l'obiettivo di rilevare le perdite, il nuovo modello, denominato *three buckets model*, prevede il riconoscimento,dunque la rilevazione degli accantonamenti, delle perdite attese in funzione del grado di deterioramento del rischio di credito degli strumenti finanziari. Non è necessario dunque che ci sia un evento o segnale esplicito che comunichi il deterioramento creditizio ma è sufficiente "sospettarlo", date le informazioni a disposizione. Per quanto riguarda la valutazione delle perdite attese, il nuovo modello si presenta come *prospettico* (*forward looking*) in quanto la stima delle perdite attese deve essere effettuata ricorrendo ad informazioni verificate e disponibili che tengano conto non solo dati storici ed attuali, ma anche prospettici. Le tre categorie previste per la valutazione sono:

1- *Performing (Stage 1):* Posizioni con rischio di credito basso. La stima della perdita attesa viene considerata con riferimento ai relativi portafogli collettivi per un periodo di un anno;

2- *Under Performing (Stage 2):* Rischio di credito di livello intermedio. Si tratta di posizioni che pur restando non critiche hanno subito un peggioramento del proprio *rating* o manifestato evidenti difficoltà economiche ed inadempienze creditizie. Le perdite attese (*forward looking*) sono valutate su un arco temporale pari alla durata contrattuale residua (*lifetime*);
3- *Non Performing (Stage 3)*: Posizioni con un elevato rischio creditizio per le quali la perdita è già avvenuta o è quasi certo avverrà. Il calcolo della perdita attesa si effettua in modo analitico in relazione alle singole posizioni deteriorate, proporzionandole alla vita residua della singola esposizione (lifetime).

Infine, come detto precedentemente la riduzione del 53% dei crediti deteriorati negli ultimi anni mostratati nella **Figura 2.1** deriva principalmente dalle cessioni. Il mercato dei NPL in Italia, pur non essendo sviluppato come in altri paesi, ha visto un aumento delle sue dimensioni grazie all'operatività di soggetti interessati nella compra/vendita di questi crediti. Nel solo 2018 sono stati ceduti circa 70 miliardi di NPL. Nel terzo trimestre del 2019 l'ammontare dei NPL lordi in Italia era di 325 miliardi[1]. Di questi 141 miliardi erano posseduti dalle banche mentre il restante era in mano a Fondi di Investimento, GACS e *Servicer specializzati*[2]. Pur registrando le banche una netta diminuzione dell'ammontare dei crediti deteriorati posseduti, bisogna notare come questi ultimi siano ancora presenti nel sistema, infatti soltanto il 7% è stato effettivamente recuperato o dichiarato come inesigibile. Alcuni[3] hanno anche parlato di "bolla" riguardo il mercato dei crediti deteriorati, mettendo in dubbio l'effettiva efficienza dei Servicer operanti. Seppur è vero che questi ultimi dovrebbero coprire l'inefficienza delle singole banche nel recuperare i crediti, essendo questa l'attività *core* di questa tipologia di società, è anche vero che la massa di crediti da gestire

[1] Si veda [14]
[2] Società specializzate nel recupero crediti.
[3] [15] Morya Longo, *"Npl, rischio «bolla» per il mercato dei crediti deteriorati"*, Il Sole 24 Ore, 11 febbraio 2019

viene accentrata su pochi attori con un conseguente problema di dimensione.

2.1.4 NPL: Situazione attuale

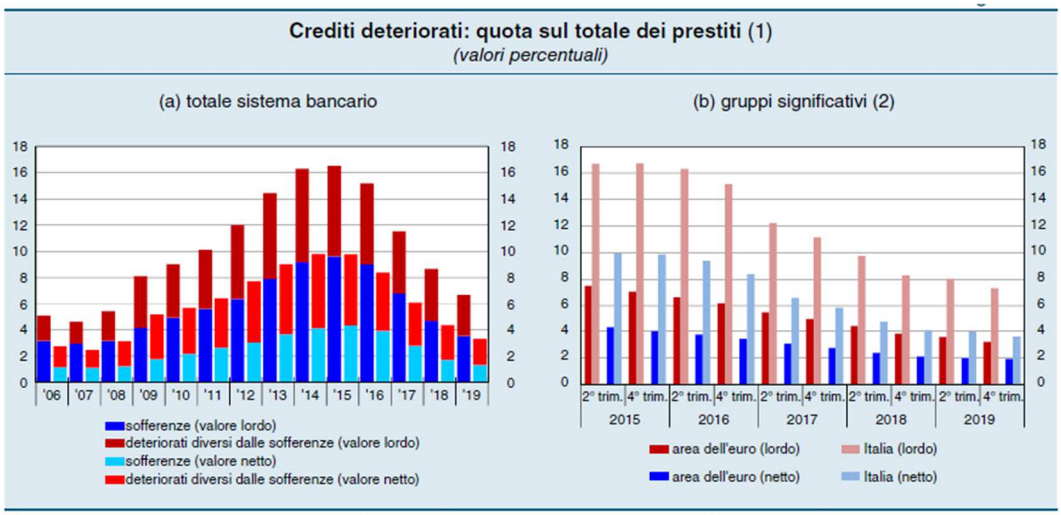
Come dichiarato da Banca d'Italia[1], "gli intermediari italiani si trovano ad affrontare i nuovi rischi da una posizione nel complesso più solida rispetto all'avvio della crisi finanziaria del 2008. Tra il 2007 e il 2019 il rapporto tra il capitale di migliore qualità e gli attivi ponderati per il rischio è quasi raddoppiato, i prestiti a famiglie e imprese sono ora finanziati interamente dai depositi e non emergono segnali di sfiducia dei risparmiatori nei confronti delle banche. Gli squilibri nei bilanci causati dalla crisi dei debiti sovrani in Europa sono stati in larga parte riassorbiti o contenuti: l'incidenza dei crediti deteriorati si è ridotta di due terzi rispetto al picco del 2015; l'impatto sul capitale delle perdite di valore dei titoli di Stato è mitigato dalla diminuzione della quota di quelli valutati al *fair value* avvenuta negli ultimi anni. L'incidenza dei nuovi crediti deteriorati sui mutui concessi negli anni 2015-19, a parità di tempo trascorso all'erogazione del prestito, è inferiore a quella sui finanziamenti degli anni precedenti. A seguito delle ingenti cessioni di sofferenze realizzate negli ultimi anni, circa la metà dei crediti deteriorati nei bilanci bancari è rappresentato da inadempienze probabili (44% e 54% del totale, rispettivamente, al lordo e al netto delle rettifiche)."

Tuttavia, lo shock macroeconomico causato dalla pandemia Covid-19 potrebbe generare un forte incremento del tasso di deterioramento dei prestiti. Esistono correlazioni fra il *Prodotto Interno Lordo* (PIL) e il livello di NPL infatti, secondo Banca d'Italia, "per ogni riduzione del PIL di un punto percentuale – mantenendo le altre variabili costanti – il flusso di nuovi crediti deteriorati, valutato in rapporto al totale dei prestiti in bonis, tende ad aumentare di 2 decimi di punto per le imprese e di 1 per le famiglie. Queste

[1] [1] Banca d'Italia, *"Rapporto sulla stabilità finanziaria"*, aprile 2020.

valutazioni, basate su regolarità storiche, non includono gli effetti dei provvedimenti legislativi sulle moratorie, sulle garanzie pubbliche ai finanziamenti sugli interventi a sostegno dei redditi delle famiglie". La sospensione della produzione così come quasi tutte le attività in generale, comporterà sicuramente un prolungamento anche nei tempi di recupero, dunque un prolungamento di tali crediti a bilancio, con le già citate conseguenze.

In **Figura 2.2** si riporta l'andamento dei NPL suddiviso per tipologia e confrontato con i numeri dell'area euro.



Fonte: segnalazioni di vigilanza consolidate per i gruppi bancari italiani, individuali per il resto del sistema; BCE, *Supervisory Banking Statistics* per l'area dell'euro. (1) I prestiti includono i finanziamenti verso clientela, intermediari creditizi e banche centrali. Sono compresi i gruppi e le banche filiazioni di intermediari esteri; sono escluse le filiali di banche estere. Le quote sono calcolate al netto e al lordo delle relative rettifiche di valore. I dati di dicembre 2019 sono provvisori. – (2) Il perimetro delle banche significative e di quelle meno significative non è omogeneo tra le date esposte in figura: a partire da giugno del 2019, con il perfezionamento della riforma del settore del credito cooperativo, Cassa Centrale Banca è diventata il dodicesimo gruppo significativo ai fini di vigilanza e nel gruppo ICCREA, già classificato come significativo prima della riforma, sono confluite 143 banche di credito cooperativo (BCC).

Figura 2.2: Andamento NPL Ratio suddiviso per tipologia. Fonte: Banca d'Italia [1]

2.2 Definizione dei NPL nel Gruppo ISP

Il *Risk Management* del Gruppo Intesa Sanpaolo utilizza diverse strategie per approcciare e modellare il rischio di credito in base alle caratteristiche del cliente e dell'esposizione nei suoi confronti. Una prima macro-suddivisione è quella fra portafogli *Performing* (Stage 1 e 2) e *Non-*

Performing (Stage 3), in accordo con quanto previsto dal principio contabile IFRS9. Come già visto, la differenza è intuibile già dai semplici nomi: nel primo caso si tratta di controparti adempienti e con basso rischio creditizio (o *Bonis*), mentre nel secondo caso si tratta di controparti con problematiche creditizie che riflettono uno stato di default.

Il Gruppo Intesa Sanpaolo si allinea alle definizioni normative degli stati di rischio, intesi come *Non Performing*, specificate come:

- "*Impairment*", contenuta nel principio contabile internazionale *IFRS9*
- "*Default*", contenuta nell'art. 178 del Regolamento 575/2013 (*Capital Requirements Regulation*).

In particolare, come crediti *Non performing*, con crescente gravità, si considerano:

- Crediti a *Past Due* – Scaduti/Sconfinanti;
- Crediti *Unlikely to pay* (UTP) – Inadempienze probabili;
- Crediti a Sofferenza (*Doubtful*);

2.2.1 Crediti Scaduti/Sconfinanti

Le posizioni *Scadute e/o Sconfinanti* si definiscono tali se non rientrano fra le esposizioni per cassa già definite come inadempienze probabili o sofferenza e risultano inadempienti, con superamento di determinate soglie di esposizione (*Soglie di Rilevanza*), consecutivamente per più di 90 giorni.

Le soglie sono di due tipologie: *Soglia Assoluta* e *Soglia Relativa*. La prima confronta l'esposizione scaduta con un predeterminato ammontare che varia dalla tipologia di cliente (si distingue principalmente fra *retail* e *non-retail*). La seconda soglia è determinata dal confronto tra l'esposizione in sconfino del debitore e il totale dell'esposizione con la banca per quella controparte.

2.2.2 Inadempienze Probabili: Forborne e Non-Forborne

Le *Inadempienze Probabili* sono "tutte le esposizioni per cassa e <<fuori bilancio>> di un debitore nei confronti del quale la banca, a suo giudizio, ritiene improbabile che lo stesso adempia integralmente (in linea capitale e/o interessi) alle sue obbligazioni creditizie, senza il ricorso ad azioni quali l'escussione delle garanzie. Tale valutazione prescinde dalla presenza di eventuali importi (o rate) scaduti e non pagati."[1] Non è quindi necessario che ci sia un esplicito sintomo di anomalia creditizia quale mancato rimborso ma potrebbe essere sufficiente un evento che decreti la probabile inadempienza (potrebbe ad esempio esserci una grave crisi nel settore in cui opera il debitore).

Tra le inadempienze probabili va incluso anche il complesso delle esposizioni verso gli emittenti che non abbiano onorato puntualmente gli obblighi di pagamento (in linea capitale e/o interessi) relativamente ai titoli di debito quotati. A tal fine si riconosce il "*grace period*" previsto dal contratto o, in assenza, riconosciuto dal mercato di quotazione del titolo.

In base alla Circolare 272 rientra fra le inadempienze probabili "il complesso delle esposizioni verso debitori che hanno proposto il ricorso per concordato preventivo c.d. "in bianco" (*art. 161 della Legge Fallimentare*), la cui segnalazione va effettuata "dalla data di presentazione della domanda e sino a quando non sia nota l'evoluzione dell'istanza. Resta comunque fermo che le esposizioni in questione vanno classificate tra le sofferenze qualora:

- ricorrano elementi obiettivi nuovi che inducano gli intermediari, nella loro responsabile autonomia, a classificare il debitore in tale categoria;
- le esposizioni erano già in sofferenza al momento della presentazione della domanda.

[1] [20] *"Manuale per la compilazione della matrice dei conti "*, Banca d'Italia

Medesimi criteri si applicano nel caso di domanda di concordato con continuità aziendale (*art. 186-bis della Legge Fallimentare*), dalla data di presentazione sino a quando non siano noti gli esiti della domanda (mancata approvazione ovvero giudizio di omologazione). In quest'ultimo caso la classificazione delle esposizioni va modificata secondo le regole ordinarie. Qualora, in particolare, il concordato con continuità aziendale si realizzi con la cessione dell'azienda in esercizio ovvero il suo conferimento in una o più società (anche di nuova costituzione) non appartenenti al gruppo economico del debitore, l'esposizione va riclassificata nell'ambito delle attività in bonis. Tale possibilità è invece preclusa nel caso di cessione o conferimento a una società appartenente al medesimo gruppo economico del debitore, nella presunzione che nel processo decisionale che ha portato tale ultimo a presentare istanza di concordato vi sia stato il coinvolgimento della capogruppo/controllante nell'interesse dell'intero gruppo. In tale situazione, l'esposizione verso la società cessionaria continua a essere segnalata nell'ambito delle attività deteriorate; essa va inoltre rilevata tra le "*esposizioni oggetto di concessioni deteriorate*".

2.2.3 Sofferenze

Ai sensi della Circolare 272, rientrano in questa categoria le esposizioni per cassa e fuori bilancio nei confronti di un soggetto in stato di insolvenza (anche non accertato giudizialmente) o in situazioni sostanzialmente equiparabili, indipendentemente dalle eventuali previsioni di perdita formulate dalla Banca. Si prescinde, pertanto, dall'esistenza di eventuali garanzie (reali o personali) poste a presidio delle esposizioni. Sono escluse le esposizioni la cui situazione di anomalia sia riconducibile a profili attinenti al rischio Paese.

Sono inclusi anche:

- le esposizioni nei confronti degli enti locali (comuni e province) in stato di dissesto finanziario per la quota parte assoggettata alla pertinente procedura di liquidazione;
- i crediti acquistati da terzi aventi come debitori principali soggetti in sofferenza, indipendentemente dal portafoglio di allocazione contabile.

Un cliente deve pertanto essere classificato a Sofferenza

- in ogni caso, qualora sia intervenuta una delle seguenti fattispecie:
    - dichiarazione di fallimento o di liquidazione coatta amministrativa;
    - avvio di atti giudiziali da parte della Banca, secondo l'iter previsto nell'attuale normativa;
    - quando il numero di rate arretrate impagate supera i limiti oggettivi (12 rate mensili impagate per tutte le forme tecniche) in riferimento alle controparti con finanziamenti rateali, fatta salva la presenza di accordi stragiudiziali e/o piani di rientro formalizzati;
- previa approfondita valutazione qualora siano intervenuti i seguenti eventi:
    - ammissione alla procedura di amministrazione straordinaria, nell'ipotesi in cui non sussistano concrete prospettive di recupero dell'equilibrio economico-finanziario e patrimoniale delle attività imprenditoriali;
    - atti giudiziali promossi da terzi;
    - cessazione dell'attività aziendale;
    - messa in liquidazione volontaria;
    - richiesta/ammissione al concordato preventivo qualora si possa ritenere che lo stato di crisi coincida, di fatto, con lo stato di insolvenza.

In linea generale, come da disposizioni regolamentari (Centrale dei Rischi – Istruzioni per gli Intermediari Partecipanti' l'Organo di Vigilanza),

l'appostazione a sofferenza implica una valutazione da parte dell'intermediario della complessiva situazione finanziaria del cliente e non può scaturire automaticamente da un mero ritardo di quest'ultimo nel pagamento del debito. La contestazione del credito non è di per sé condizione sufficiente per l'appostazione a sofferenza

2.3 Valutazione dei NPL nel Gruppo ISP

La valutazione del portafoglio Non-performing avviene attraverso due diversi macro-approcci in base all'ammontare dell'esposizione nei confronti del debitore. Nello specifico, si parla di:

- Valutazione *Analitico-statistica*, per tutte quelle posizioni con esposizione inferiore alla *soglia di separazione*. È basata su un modello statistico di previsione delle perdite ed integrata da un modello statistico per tenere conto degli scenari macroeconomici attesi (componente *Add-on*);
- Valutazione *Analitica*, per tutte quelle posizioni al di sopra della *soglia di separazione*. In questo caso, trattandosi di controparti di una certa rilevanza, il modello statistico viene sostituito dalla valutazione analitica del gestore. La valutazione viene comunque integrata dagli *Add On* come nel punto precedente.

Una volta classificata la posizione, la metodologia fra analitico-statistica e analitica specifica non cambia per tutta la durata del recupero, al netto di alcuni casi specifici. Ad esempio, se l'accumulo di addebiti di una controparte, inizialmente sotto soglia, aumenta in modo tale da superare la soglia, dovrà essere valutata non più con l'approccio analitico-statistico ma analitico specifico.

2.3.1 Modello Analitico-Statistico per il calcolo della LGD

Trattandosi di portafogli *Non-Performing*, per i quali la situazione di default si è già verificata, nella formula dell'*EL* la probabilità di default PD assume l'unità e pertanto diventa: *EL = AE·LGD*. La variabile rilevante è chiaramente la LGD ed è pertanto questa che va calcolata per determinare la perdita attesa per quella controparte.

Il calcolo della LGD avviene secondo la costruzione delle cosiddette *griglie LGD* che fanno riferimento al già citato metodo *Workout LGD*. Queste vengono costruite attraverso modelli differenziati e specializzati per segmento di operatività quali:

- Corporate
- SME Retail (*Small Medium Entities*)
- Mutui
- Altri Retail
- Factoring
- Leasing
- Enti pubblici

Per le banche invece (e soltanto per Stage 1 e Stage 2) viene calcolata la *Market LGD*, accennata nel paragrafo 1.1.1.2 del Capitolo 1, sulla base dei prezzi degli strumenti di debito, osservati nei 30 giorni successivi alla data ufficiale di default, appartenenti a banche di tutto il mondo che sono andate in default in passato. Per lo *Stage 3* viene invece utilizzata la valutazione Analitica. Una differenza caratterizzante i modelli di LGD per i portafogli Non Performing rispetto a quelli Performing, è la considerazione del tempo di permanenza della posizione in quello stato di rischio e la valutazione dei tassi di perdita per attivazione delle procedure di recupero giudiziali.

2.3.1.1 Griglie LGD

L'approccio di costruzione delle griglie di LGD prevede il calcolo attraverso un modello econometrico per le posizioni in Sofferenza che funge da base per il calcolo della LGD degli altri stati. Infatti, la LGD per gli Unlikely to Pay e Past Due, viene dedotta da quanto calcolato per lo stato Sofferenza utilizzando un "fattore correttivo", detto *Danger Rate*.

Per costruire il modello econometrico si seguono i seguenti passi logici:

- Si individua il set di variabili che possono individualmente contribuire nella stima della LGD (Area Geografica, Forma Tecnica, Garanzie, Informazioni generiche della controparte, ecc);
- Per ogni variabile scelta nel punto precedente, si definisce l'insieme di valori assumibili (ad esempio, se si sceglie Area Geografica per una controparte in Italia, si potrebbe decidere che i valori assumibili sono tutte le venti regioni, oppure soltanto due, Nord e Sud, ecc);
- Si studia la correlazione delle singole variabili con la LGD - variabile target per capire quale di queste meglio contribuisce alla stima (*Univariate Analysis*);
- Selezionate le variabili più adatte si esegue una analisi di correlazione (*Correlation Analysis*) fra queste. Variabili correlate in modo consistente possono distorcere in modo altrettanto consistente la stima;
- Verificata l'assenza di forti correlazioni fra le variabili, si esegue l'analisi multivariata (*Multivariate Analysis*) attraverso più modelli di regressione lineare multivariati con variabili qualitative (o *dummies*) combinando set diversi di quest'ultime ad ogni prova;
- Si sceglie il modello più performante e che rifletta meglio la realtà economica in contesto (*Comparison between models*);
- Si costruiscono le griglie della LGD ottenuti i coefficienti e quindi le contribuzioni delle variabili (e loro combinazioni) del modello multivariato scelto nel punto precedente.

A questo punto, date le informazioni riguardanti una controparte, può essere calcolata la LGD in base alla griglia costruita. Volendo fornire un esempio di utilizzo del modello econometrico appena spiegato per il calcolo della LGD media di un determinato pool *m*, nel caso di tre caratteristiche/variabili della controparte, *A*, *P* ed *M*, si consideri la seguente formula:

$$LGD_m = f\left(\mu + \sum_{k=1}^{K-1} \alpha_k^A D_{mk}^A + \sum_{j=1}^{J-1} \alpha_j^G D_{mj}^G + \sum_{i=1}^{I-1} \alpha_i^F D_{mi}^F + \sum_{k=1}^{K-1}\sum_{j=1}^{J-1} \beta_{k,j}^{A,G} D_{mk}^A D_{mj}^G + \sum_{k=1}^{K-1}\sum_{i=1}^{I-1} \beta_{k,i}^{A,F} D_{mk}^A D_{mi}^F + \sum_{i=1}^{I-1}\sum_{j=1}^{J-1} \beta_{i,j}^{F,G} D_{im}^F D_{mj}^G + \sum_{k=1}^{K-1}\sum_{j=1}^{J-1}\sum_{1=1}^{I-1} \gamma_{k,j,i}^{A,G,F} D_{mk}^A D_{mj}^G D_{mi}^F\right) + \varepsilon_m$$

(2.1)

Con

- μ l'intercetta all'origine (o la LGD di partenza, detta *Baseline*);
- *A*, *G* ed *F* sono caratteristiche della controparte (ad esempio, Area geografica, Garanzie, Forma tecnica) che possono rispettivamente assumere *K*, *J* e *I* valori;
- Le variabili *D* sono le *dummies*, che dipendono dalle variabili *A*, *G* ed *F* e dai rispettivi valori *K*, *J*, *I*;
- $\alpha, \beta\ e\ \gamma$ sono rispettivamente le contribuzioni individuali delle variabili (*main effects*), le contribuzioni in combinazione di due variabili (*2-way interactions)*, la contribuzione della combinazione delle tre variabili (3- *way interactions)*.
- ε_m è l'errore rispetto all'osservazione *m*;

Il modello esemplificativo appena esposto mostra la forma della LGD in caso di tre variabili. Non esiste un limite al numero di variabili utilizzabili, queste possono essere molte di più dell'esempio esposto ed in quel caso si andranno a considerare le contribuzioni combinate di ognuna di queste, che

saranno tanto più numerose quanto più numerose sono le variabili coinvolte.

Provando a fare un esempio numerico per meglio chiarire l'utilizzo delle griglie di LGD, si pensi al calcolo per un cliente con caratteristiche tali da essere assegnato alla Baseline *U*, Area Geografica *A,* avente una forma tecnica Mutuo *FTM*, ed una garanzia reale *GR*. I contributi, dalle griglie LGD costruite, risultano essere:

Tabella 2.1: Esempio di calcolo con griglie LGD

Baseline U	*72%*		
Main Effects	***A***	***FTM***	***GR***
	13%	-40%	-18%
2-way interactions	***A/FTM***	***A/GR***	***FTM/GR***
	-10%	1%	15%
3- way interactions	***A/FTM/GR***		
		1%	

L'assegnazione iniziale è quella della Baseline *U*, ovvero 72%. Dopodiché, si verifica che il cliente appartenga all'area specifica *A* che, in media, incrementa la LGD del 12%. Lo stesso viene fatto per la forma tecnica del contratto (Mutuo) e la presenza o meno di garanzie (in questo caso reale). L'effetto totale dei contributi individuali è un decremento della LGD di baseline del 13%-40%-18%= -45%.

Successivamente si osservano gli effetti combinati a due variabili. Per una controparte che ha un Mutuo e contemporaneamente opera nell'Area Geografica *A*, si osserva in media un decremento della LGD di -10%. La stessa osservazione viene fatta per controparti operanti nell'area *A* ed aventi garanzie reali *GR*; controparti con Mutui e garanzie reali *GR*. Si evince che il contributo degli effetti combinati a due variabili è un incremento del -10%+1%+15%= 6%.

Controparti operanti nella zona A ed aventi un Mutuo con garanzie reali hanno in media un contributo incrementativo del 1%. Si conclude quindi che la LGD media assegnata alla singola controparte è:

$$LGD_m = 72\% - 45\% + 6\% + 1\% = 34\%$$

2.3.1.2 Danger rate

Come precedentemente esposto, il *Danger Rate* viene utilizzato per "correggere" la LGD stimata per una determinata controparte, date le sue caratteristiche, e quindi adattarla al suo reale stato di rischio: Past Due, Unlikely to pay oppure Bonis. Si potrebbe dunque generalizzare con la formula:

$$LGD_m = DangerRate \cdot LGD_{soff} \tag{2.2}$$

Per calcolare il *Danger Rate* bisogna costruire il concettuale albero degli eventi e determinare le singole probabilità che costituiscono ogni passaggio. Considerando gli stati Past Due, Unlikely to pay, Bonis e Sofferenza; considerando le probabilità di passare da uno stato ad un altro, dette anche *tassi di migrazione*, l'albero degli eventi potrebbe essere sintetizzato come in **Figura 2.3**.

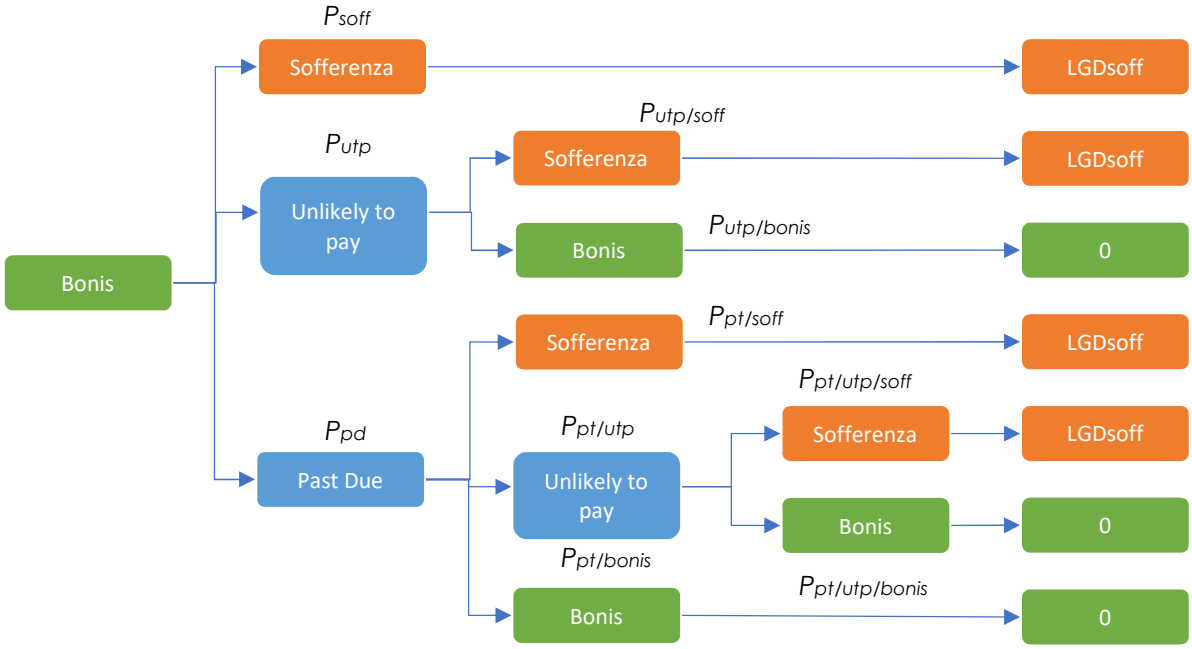


Figura 2.3: Albero dei passaggi di stato controparte in default

Una controparte, partendo da una posizione di Bonis, potrebbe entrare direttamente in Sofferenza con probabilità P_{soff}, in Unlikely to Pay con probabilità P_{utp}, oppure in Past Due con probabilità P_{pd}. Il totale delle controparti considerate sono quelle per cui uno dei tre precedenti passaggi di stato è stato verificato. Di conseguenza, la somma delle tre probabilità citate è pari all'unità.

Le controparti possono entrare in stato Unlikely to Pay direttamente dal Bonis e, a sua volta, successivamente entrare in Sofferenza o tornare in Bonis rispettivamente con probabilità $P_{utp/soff}$ e $P_{utp/bonis}$. La stessa logica va applicata alle controparti che entrano in Past Due e possono successivamente entrare in Sofferenza, in Unlikely to Pay o in Bonis rispetivamente con probabilità $P_{pd/soff}$, $P_{pd/utp}$ e $P_{pd/bonis}$.

Infine, ci sono controparti che possono entrare in Past Due, successivamente in Unlikely to Pay e poi andare in Sofferenza oppure tornare in Bonis, rispettivamente con probabilità, $P_{pd/utp/soff}$ e $P_{pd/utp/bonis}$.

Le probabilità vengono calcolate come il numero di controparti che "transitano" attraverso quel passaggio di stato rispetto al totale delle

controparti considerate. P_{utp} ad esempio, è il numero di controparti che dallo stato Bonis sono passate allo stato UTP sul totale di controparti in stato Bonis che sono passate a Sofferenza, UTP o Past Due. Le probabilità composte devono essere lette come le probabilità condizionate della statistica. Ad esempio, la probabilità $P_{pd/bonis}$ è la probabilità che una posizione entri in stato Bonis una volta verificatosi il passaggio in Past Due: $P_{pd/bonis} = \mathbb{P}(Bonis \mid Past\ Due)$.

Seguendo l'albero dunque si riescono a ricavare le formule per le LGD specifiche degli altri stati partendo dalla LGD_{soff}:

$$LGD_{bonis} = (P_{soff} + P_{utp}P_{utp/soff} + P_{pd}P_{pd/soff} + P_{pd}P_{pd/utp}P_{pd/utp/soff})LGD_{soff}$$

$$LGD_{utp} = (P_{utp/soff})LGD_{soff}$$

$$LGD_{pd} = (P_{pd/soff} + P_{pd/utp}P_{pd/utp/soff})LGD_{soff}$$

Rispettivamente (2.3), (2.4), (2.5)

La LGD_{utp} è stata espressa per semplicità considerando tutte le posizioni in stato Inadempienza Probabile indipendentemente dalla provenienza da Bonis o Past Due.

2.3.1.3 Evoluzione delle esposizioni e LGD per rientri in bonis

Nel modello precedentemente presentato si dà per scontato che l'ammontare di credito vantato con la controparte si mantenga costante fra i vari passaggi. Empiricamente però si verifica che questo non è esatto in quanto, ad esempio, una controparte in stato di Inadempienza Probabile potrebbe avere un'esposizione di 100 crediti nel momento in cui le è stato assegnato questo stato di rischio ed un'esposizione di 60 nel momento della dichiarazione di insolvenza. Nell'arco temporale fra l'entrata in Inadempienza Probabile e l'entrata in Sofferenza, la banca è riuscita ad incassare 40 crediti dalla controparte.

Per tenere conto di questi eventi, si introduce nel modello un tasso di evoluzione dell'esposizione q_s nello stato *s* descritto come:

$$q_s = 1 - \frac{\sum(R_t - C_t)}{Esposizione_s} \tag{2.6}$$

Con

- R_t sono i recuperi attualizzati verificatesi nel periodo t;
- C_t sono i costi/aggravi attualizzati verificatesi nel periodo t;
- $Esposizione_s$ è l'esposizione dello stato di rischio s.

Un ulteriore considerazione sul modello esposto precedentemente è l'implicita assunzione che, per quelle controparti che entrano in stato di Inadempienza Probabile o Past Due e poi tornano in stato Bonis, non ci siano delle perdite (*LGD = 0*). Anche in questo caso si evince empiricamente che la perdita, pur non essendo rilevante come quella del passaggio a Sofferenza, non è nulla.

Integrando il modello con entrambe le considerazioni fatte si ottiene un modello generalizzato per le posizioni in Bonis (stato più generico):

$$\begin{aligned} LGD_{bonis} = {} & P_{soff} LGD_{soff} + P_{utp} P_{utp/soff} q_{utp} LGD_{soff} + P_{utp} P_{utp/bonis} LGD_{utp/bonis} \\ & + P_{pd} P_{pd/soff} q_{pd} LGD_{soff} + P_{pd} P_{pd/utp} P_{pd/utp/soff} q_{pd} q_{utp} LGD_{soff} \\ & + P_{pd} P_{pd/utp} P_{pd/utp/bonis} q_{pd} LGD_{pd/utp/bonis} \\ & + P_{pd} P_{pd/bonis} LGD_{pd/bonis} \end{aligned}$$

(2.7)

2.3.2 Calcolo degli Add-On

Gli Add-On sono quelle componenti aggiuntive di correzione della LGD calcolata che tengono conto di informazioni più di alto livello. Coerentemente con le griglie utilizzate per la valutazione analitico-statistica, si stimano i seguenti *Add-On*:

- *Add-On current conditions*: Componente basata su variabili gestionali, si considerano principalmente le condizioni economiche correnti;
- *Add-On forward looking*: Componente legata alle prospettive macroeconomiche e basata su gli scenari *Most-Likely* e peggiorativo previsti nell'orizzonte temporale dei successivi tre anni.
- *Add-On Sofferenze cedibili*: Componente che cerca di includere i possibili scenari di vendita/cessione di crediti previsti dal Gruppo Intesa Sanpaolo.

Le prime due componenti vengono calcolate separatamente e successivamente sommate per comporre l'*Add-On Complessivo*. La terza componente invece viene integrata nel calcolo del Add-On complessivo solo per le Sofferenze che hanno tali caratteristiche da essere ritenute cedibili.

2.3.2.1 Add-On current conditions

I modelli di LGD si basano su serie storiche di lungo periodo (cd. *TTC*-Through). Secondo i principi contabili è opportuno rendere questi parametri più aderenti alle condizioni economiche correnti e pertanto Intesa Sanpaolo condiziona le stime TTC attraverso un fattore di calibrazione definito "Add-On current conditions".

Dall'osservazione di lungo periodo della relazione tra le perdite osservate sulle posizioni a sofferenza chiuse ed alcune variabili gestionali è stata individuata una dipendenza tra le perdite e la variabile *NPL Ratio*[1]. Questa correlazione risulta significativa soprattutto tra le perdite osservate in un determinato anno, e il livello di NPL Ratio del Gruppo ISP. Il razionale sottostante potrebbe essere spiegato dal fatto che in presenza di un elevato

[1] Il NPL Ratio è la percentuale di NPL sul totale dei crediti dell'intermediario.

NPL ratio, la banca attiva leve gestionali atte a ridurre rapidamente tale indicatore, accelerando le lavorazioni e le chiusure delle posizioni, anche a costo di un maggiore impatto economico. Facendo un paragone con il triangolo *Costo-Tempi-Qualità* utilizzato nella disciplina di *Project Management,* si potrebbe dire che anche in questo caso, cosi come per certe situazioni specifiche dei progetti, si cerca di ridurre i tempi aumentando l'impatto negativo sulle altre due variabili, dunque accrescendo i costi operativi/gestionali e riducendo leggermente la qualità dell'operato. Essendo in questo caso la qualità dell'operato la capacità di recuperare il credito, una diminuzione della qualità è equivalente ad una diminuzione di quanto recuperato, ovvero una ulteriore perdita economica.

Di converso, quando l'NPL ratio è più basso, la banca si concentra nel massimizzare il recupero su ogni posizione, potendo permettersi di impiegare un tempo maggiore per la lavorazione delle stesse.

Il modello definito prevede la stima di relazioni matematiche tra l'NPL Ratio annuale e la Perdita su Posizioni Chiuse (PPC) media osservata annualmente, differenziate per segmento. Le variabili significative utilizzate in questi modelli risultano essere spesso variabili non stazionarie, ovvero variabili che cambiano distribuzione se traslate nel tempo. L'utilizzo di relazioni matematiche, quali regressioni lineari e non, tra variabili non stazionarie può portare a risultati spuri o casuali a meno che non esista tra loro una relazione di equilibrio di lungo periodo, detta di *cointegrazione*[1].

Le analisi condotte dal Gruppo Intesa Sanpaolo, suggeriscono la presenza di una relazione lineare tra le variabili del tipo:

$$PPC = \alpha + \beta \cdot NPL_{t-2} \tag{2.8}$$

[1] Si intende il caso in cui due o più serie temporali con trend stocastici si muovono congiuntamente in modo simile nel lungo periodo, tanto che sembrano possedere lo stesso trend. La definizione di c. è dovuta all'econometrico C.W.J. Granger che per tale ricerca è stato insignito, insieme a R.F. Engle, del premio Nobel per l'economia nel 2003 [21]

I test di cointegrazione, test di *Pesaran* e *Shin*, confermano la validità delle relazioni lineari di cui sopra indicando che pur utilizzando il modello variabili non stazionarie queste si muovono indicativamente allo stesso modo nel lungo periodo.

Mentre la PPC è limitata al dominio di valori compresi tra 0 e 1, la relazione lineare ha un dominio infinito. Per questo motivo e per limitare il modello a valori economicamente ragionevoli di PPC, si considera la relazione lineare tra un *floor (A)* ed un *cap* (B) opportunamente scelti sulla base dello specifico segmento esaminato, smussandola asintoticamente in una relazione logistica:

$$PPC = \alpha + \beta \cdot NPL_{t-2} \qquad \rightarrow \qquad PPC = A + \frac{B - A}{1 + e^{-(C + D \cdot NPL_{t-2})}}$$

(2.9)

dove:

- *A* rappresenta il limite inferiore della funzione per valori decrescenti di NPL;
- *B* rappresenta il limite superiore della funzione per valori crescenti di NPL.

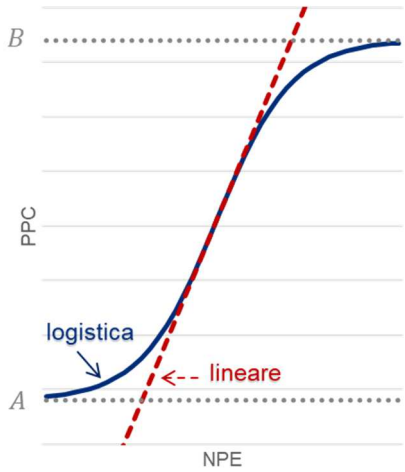


Figura 2.4: Confronto andamento lineare e Logistico.

I parametri A e B, prima di eseguire il fit, vengono calibrati secondo la seguente logica:

$$\begin{cases} B = PPC_{media} + \Delta \\ A = PPC_{media} - \Delta \end{cases} \tag{2.10}$$

dove Δ, la distanza degli asintoti dal valore medio della Perdita su Posizioni Chiuse, è:

$$\Delta = \max\left[PPC_{max} - PPC_{media}; PPC_{media} - PPC_{min}\right] \tag{2.11}$$

Risulta quindi che A e B sono stati calibrati in modo da disporsi simmetricamente intorno al valore medio di PPC osservata. Una volta calibrati A e B, i parametri C e D vengono calcolati attraverso un approccio *non-linear least squares.* Le relazioni matematiche così ottenute vengono utilizzate per calcolare le PPC stimate per anno di chiusura futuro in funzione dell'NPL Ratio. La PPC stimata viene confrontata con la media semplice delle PPC annuali osservate nel campione di sviluppo ai fini della determinazione dell'Add On.

2.3.2.2 Add-On forward looking

Come richiesto dal Principio contabile IFRS9, occorre considerare anche elementi *forward looking* sulle stime di LGD ancorate alle *"current conditions"* attraverso la prima componente di Add-On trattata al paragrafo precedente. Questo Add On va a considerare variabili macroeconomiche che cerchino di cogliere la relazione non lineare fra la perdita attesa egli eventi futuri.

Indicando $LGD_{most\ likely}$ e $LGD_{peggiorativo}$ rispettivamente le perdite attese nello scenario più probabile, e lo scenario peggiore stimato; considerando LGD_{TTC} come la perdita attesa media nell'arco temporale di interesse (tre anni), la componente forward looking incide sulle stime di LGD mediante due effetti separatamente quantificati e poi sommati:

- scenario *Most-Likely* rispetto allo scenario *Through the Cycle*;
- scenario peggiorativo rispetto a scenario *Most-Likely*.

Ovvero:

$$Add\_on_{most\ likely} = LGD_{most\ likely} - LGD_{TTC}$$

$$Add\_on_{peggiorativo} = LGD_{peggiorativo} - LGD_{most\ likely}$$

rispettivamente (2.12) e (2.13)

e conseguentemente:

$$Add\_on_{forward\ looking} = Add\_on_{most\ likely} + Add\_on_{peggiorativo} \quad (2.14)$$

Tale componente è soggetta ad un floor pari a zero, ovvero per le posizioni in *Stage 3* non viene prudenzialmente prevista la possibilità che lo scenario Most-Likely prevalga rispetto a quello peggiorativo nel computo finale dell'Add-On.

Infine, tenendo conto di entrambi gli Add On discussi, si calcola quello complessivo come:

$$Add\_on_{complessivo} = Add\_on_{current\ conditions} + Add\_on_{forward\ looking} \quad (2.15)$$

2.3.2.3 Add-On Sofferenze cedibili

Per un perimetro definito di crediti in Sofferenza aventi caratteristiche di cedibilità, al fine di determinare l'Add-On finale, l'Add-On da scenario macroeconomico complessivo viene integrato, in proporzione alle probabilità di accadimento, con un Add-On che riflette i possibili impatti sugli importi recuperabili in caso si concretizzi uno scenario di vendita delle stesse.

Per la realizzazione di tale framework sono necessari dunque i seguenti input:

- perimetro posizioni in Sofferenza cedibili;
- % probabilità di vendita del perimetro delle Sofferenze ritenute cedibili, calcolata rapportando l'ammontare delle vendite previste dall'NPL plan all'ammontare del portafoglio cedibile;

- fair value della quota ipotecaria del portafoglio cedibile espresso in percentuale del *Gross Book Value* (GBV);

- fair value della quota non ipotecaria del portafoglio cedibile espresso in percentuale del GBV.

Ai fini della stima del fair value, ossia del prezzo che si percepirebbe per la vendita dei crediti in Sofferenza individuati come cedibili in una regolare operazione tra operatori di mercato alla data della valutazione, sono prese in considerazione tutte le informazioni disponibili, adottando le assunzioni che un normale operatore di mercato ragionevolmente utilizzerebbe nella determinazione del prezzo.

La stima del fair value segue un approccio valutativo integrato, basato sull'applicazione di un metodo analitico e di un metodo empirico.

Il metodo analitico si basa sull'attualizzazione dei flussi di cassa attesi. Tale tecnica viene applicata ai singoli rapporti, eventualmente classificati in cluster omogenei (es. Secured Corporate/Secured Retail e Unsecured Corporate/Unsecured Retail). Gli elementi chiave ai fini dello sviluppo di tale metodologia sono:

- la stima della distribuzione dei flussi di cassa, in termini di ammontare e tempi di recupero, al netto di spese e costi legali;

- il tasso di attualizzazione, che deve considerare il profilo di rischio dei flussi oggetto di attualizzazione, tenendo anche conto delle condizioni correnti di mercato.

I risultati ottenuti vengono infine sottoposti ad analisi di sensitività, al fine di apprezzare la variabilità dei valori ottenuti al variare delle principali assunzioni valutative adottate. In particolare, vengono sviluppati differenti scenari in relazione ai seguenti fattori: tasso di attualizzazione, ammontare

dei recuperi attesi (abbassamento di valore per ogni asta, abbassamento di valore per definizioni stragiudiziali, ponderazione per i differenti gradi ipotecari), tempi di recupero attesi (tempi medi delle aste, tempi medi delle definizioni stragiudiziali). Gli scenari di *sensitivity* devono essere coerenti in termini di ammontare, tempo di recupero, tasso di sconto, al fine di giungere ad un intervallo di valori del portafoglio oggetto di valutazione.

Il metodo empirico si basa sull'analisi delle recenti transazioni osservate sul mercato. La significatività dei risultati ottenuti attraverso tale metodologia è funzione della comparabilità delle transazioni osservate con i crediti oggetto di stima. Occorre pertanto analizzare le transazioni di mercato al fine di costruire cluster omogenei e rilevare, per ciascuno di essi, i multipli impliciti in termini di prezzo rapportato al GBV dei crediti. Ai fini dell'analisi e della definizione di un intervallo di moltiplicatori, rileva la numerosità delle transazioni riscontrate e la variabilità dei multipli impliciti osservati, eventualmente procedendo a normalizzazioni, opportunamente argomentate, per tenere conto di osservazioni anomale. Nell'ambito della definizione del campione di transazioni di mercato, vengono identificate le transazioni concluse dal Gruppo Intesa Sanpaolo, al fine di confrontare l'intervallo di multipli riscontrato sul mercato con l'intervallo delle operazioni del Gruppo e valutarne complessiva coerenza, anche ai fini di definire l'intervallo dei moltiplicatori.

I moltiplicatori identificati per cluster omogenei, vengono infine applicati ai corrispondenti crediti oggetto di stima. Alla luce delle analisi svolte, il fair value viene determinato all'interno del range dei risultati ottenuti tramite i due metodi, privilegiando l'area di sovrapposizione fra i risultati degli stessi.

Una volta definiti questi input, per ciascuna posizione in Sofferenza appartenente al perimetro cedibile, si procede al calcolo dell'Add-On finale nel seguente modo:

$$Add_{on_{finale}} = P_{vendita} \cdot Add\_on_{sale} + (1 - P_{vendita}) \cdot Add_{on_{complessivo}}$$

(2.16)

dove:

- $P_{vendita}$: definita coerentemente con il piano NPL di Gruppo ed identicamente applicata su tutte le posizioni ricomprese nel perimetro cedibile

- Add_on_{sale}: definito dalla differenza tra la percentuale di copertura prima dell'applicazione dello scenario di vendita ed il complemento ad uno del "fair value" delle Sofferenze cedibili definito in input

- $Add\_on_{complessivo}$: declinato secondo quanto illustrato in precedenza in funzione delle caratteristiche del rapporto oggetto di valutazione (es. segmento regolamentare, forma tecnica, tipologia di garanzia, ecc.)

Per tutti i crediti deteriorati non ricompresi nel perimetro delle Sofferenze cedibili l'Add_on_{finale} è equivalente all'$Add\_on_{scenario\ complessivo}$.

Parte II: Machine Learning e Calcolo della LGD

Capitolo 3: Introduzione al Machine Learning[1]

Come indica il nome stesso, *Machine Learning* (ML) è quell'insieme di tecniche che se applicate permettono ad una macchina di *imparare* da un determinato ammontare di dati, conoscenze e modelli impliciti in esso. Per dirla con le famose e più precise parole di Tom Mitchell nel suo libro *Machine Learning*[2]:

"Si dice che un programma apprende dall'Esperienza E con riferimento ad alcune classi di compiti T e con misurazione della performance P, se le sue performance P nel compito T migliorano con l'esperienza E"

Il vantaggio dell'utilizzare algoritmi di apprendimento automatico può essere facilmente intuibile anche per coloro che non conoscono neanche in minima parte la materia. Le macchine hanno un potenziale di calcolo su un singolo task superiore a quello che un qualsiasi essere umano riuscirebbe a fare con penna e foglio in mano, specialmente se si prendono in considerazione le ultime novità riguardanti i computer quantistici in grado di svolgere gigantesche moli di calcolo in pochi secondi. Per farci una idea, mentre un computer classico impiegherebbe circa 300 trilioni di anni (!!) per decifrare una chiave crittografica di tipo RSA 2048-bit [3](risolvendo quindi il problema della fattorizzazione in numeri primi di un numero intero gigantesco), un computer quantistico impiegherebbe 10 secondi. Con un tale potenziale di calcolo, secondo molti nel futuro le macchine saranno in grado di eguagliare il cervello umano e probabilmente anche di superarlo. Per lo stesso motivo, saranno in grado di imparare ad una maggior velocità e di immagazzinare una maggior quantità di informazioni. Si potrebbe

[1] Per la realizzazione di questo capitolo sono stati utilizzati principalmente due libri: di stampo più pratico [2], di stampo più teorico [38].
[2] [36] Mitchell, T. (1997), *"Machine Learning"*, McGraw Hill.
[3] [37]

dunque generalizzare dicendo che il vantaggio di tali algoritmi deriva dal fatto che possono sostituirsi in molte delle attività che oggi affrontiamo con l'aiuto del nostro cervello.

Ai giorni nostri le applicazioni pratiche sono variegate e ben visibili. Partendo dall'esempio più classico di smistamento automatico delle mail (Spam/non Spam) e arrivando ai vari assistenti vocali già presenti all'interno di molte case, alle macchine che si guidano da sole, al riconoscimento facciale e ai primi robot umanoidi, si evince come il futuro verrà guidato da questa disciplina in modo sempre più decisivo.

In questo capitolo si cerca di introdurre le nozioni necessarie a capire dal punto di vista teorico e pratico tutti gli algoritmi utilizzati per il calcolo della LGD nel Capitolo 4. Saranno inoltre illustrate le varie metodologie classiche di trattamento iniziale dei dati.

Si parlerà spesso di:

- *Dataset*: È l'insieme di dati, organizzati in modo tabellare, a disposizione.
- *Training set*: È il sottoinsieme del dataset che si predispone per la fase di Training, ovvero di apprendimento.
- *Test set*: È il sottoinsieme del dataset predisposto per verificare le performance dell'algoritmo allenato sul Training set, di conseguenza, risulta complementare a quest'ultimo.
- *Istanze*: Sono le singole righe (o record) del *set* a cui si fa riferimento.
- Attributi/ Features: Sono le colonne (al netto della variabile target per gli algoritmi supervisionati) del *set* a cui si fa riferimento. Quando si spiegano gli algoritmi, gli attributi vengono denominati *variabili*, in quanto lo sono a tutti gli effetti dal punto di vista dei primi.
- Label/Variabile target: Per algoritmi supervisionati, si tratta della colonna nella quale viene inserito il valore veramente osservato/desiderato/di label (etichetta) che ci serve per confrontarlo con la stima prodotta dall'algoritmo e ricavare quindi l'errore commesso.

Inoltre, verranno utilizzati spesso i nomi degli algoritmi e delle metodologie in lingua inglese in quanto più utilizzati nel comune linguaggio delle versioni in italiano.

3.1 Classificazione degli algoritmi di Machine learning

Gli algoritmi di Machine Learning vengono utilizzati per affrontare diversi problemi con diverse condizioni ed è sulla base di questi fattori che vengono classificati. Nello specifico si parla principalmente di:

- Algoritmi *Supervisionati* vs *Non supervisionati*
- Algoritmi ad *Apprendimento Online* vs *Apprendimento a Batch*
- Algoritmi *Instance-based* vs *Model-based*

Questi tre insiemi non sono mutuamente esclusivi. Un unico algoritmo, in base alle proprie caratteristiche, può rientrare contemporaneamente in più di una di queste classi.

3.1.1 Algoritmi Supervisionati vs Non supervisionati

Si parla di *Algoritmi Supervisionati* quando, avendo gli attributi del dataset, si hanno a disposizione anche i risultati osservati/desiderati (*labels*). Questi algoritmi vengono utilizzati nella maggior parte dei casi per risolvere due tipologie di problemi: *Classificazione* e *Regressione*.

I problemi di classificazione di una variabile target entro determinate categorie, possono essere affrontati avendo a disposizione le informazioni (o gli attributi) che in passato hanno permesso di determinare la categoria di appartenenza dell'istanza osservata. Si potrebbe, ad esempio, alimentare l'algoritmo con tutte le informazioni rilevanti (gli attributi) di tutti i clienti (istanze) di una determinata società specificando quelli che si sono rivelati dei cattivi creditori da quelli che hanno sempre pagato quanto dovuto nei tempi prestabiliti. In questo modo l'algoritmo imparerà a classificare, in base alle proprie caratteristiche della controparte, tutti i clienti futuri come cattivi o buoni creditori.

Lo stesso vale per i problemi di regressione nei quali vogliamo determinare non una categoria di appartenenza ma un valore continuo che potrebbe essere ad esempio un prezzo o una quantità.

Esempi di algoritmi supervisionati sono:

- k-Nearest neighbors
- Regressione Lineare
- Regressione Logistica
- Support Vector Machines
- Decision Tree
- Random Forest

Nelle successive pagine si entrerà nel dettaglio di alcuni di questi.

Con *Algoritmi Non-Supervisionati* si intende dire che pur avendo a disposizione gli attributi, non sono disponibili le labels. Si potrebbe dire che mentre nel caso supervisionato l'algoritmo ha modo di correlare o confrontare per ogni istanza un risultato finale, quindi una variabile target, in questo caso il risultato finale non esiste e dev'essere dedotto. Per questo motivo le problematiche che vengono affrontate con gli algoritmi non-supervisionati differiscono da quelle affrontate con quelli supervisionati.

Si parla spesso di:

- *Clustering*: Avendo un insieme di informazioni cercare di capire se ci sono dei sottoinsiemi che differiscono seguendo un determinato modello con determinate caratteristiche. Ad esempio, avendo a disposizione i dati della clientela di una azienda si può scoprire che ci sono gruppi di clienti molto simili tra loro e molto dissimili con gli altri gruppi. Questo potrebbe aiutare a personalizzare l'esperienza, i servizi o i prodotti offerti e ad incrementare il beneficio del cliente.

- *Visualizzazione*: Avendo moltissimi dati potrebbe essere utile predisporli in modo visibilmente capibile per indagare l'esistenza di pattern non direttamente ricavabili dalla sola analisi dei dati. Questo punto potrebbe essere visto come una coda di quello precedente.

- *Riduzione della dimensione*: Un problema molto diffuso quando si hanno grosse quantità di dati è quello di capire quali di questi possono essere effettivamente utili. Algoritmi non-supervisionati possono aiutare ad individuare gli attributi più esplicativi ma anche

a crearne di nuovi come combinazione di quelli presenti, per meglio spiegare il contenuto informativo dell'insieme.

- *Individuazione delle anomalie*: Si possono usare algoritmi non supervisionati per trovare i cosiddetti *outliers*[1], che possono rivelarsi come dati anomali.

- *Ricerca delle correlazioni*: L'esplorazione e combinazione degli attributi può aiutare a scoprire correlazioni nascoste e non intuitive che spesso e volentieri portano ad applicazioni pratiche estremamente utili.

Esistono anche gli algoritmi *Semi-supervisionati*. È facile intuire che in questo caso si parla di insiemi per i quali abbiamo le labels ma non su tutto il set informativo. Il riconoscimento degli elementi nelle fotografie è un esempio lampante di questi algoritmi. Dato un insieme di fotografie di animali, l'algoritmo non supervisionato probabilmente individuerà efficacemente che un cane è diverso da un gatto senza tuttavia sapere che si tratta di un cane. Basta fornire all'algoritmo un insieme piccolo di fotografie nelle quali indichiamo l'insieme di pixel nel quale è presente un cane per abilitare l'algoritmo a classificare come cani tutti quegli elementi che aveva precedente indicato come simili senza tuttavia averne la *label*.

3.1.2 Algoritmi ad Apprendimento Online vs Apprendimento a Batch

La differenza fra algoritmi *online* ed a *batch* è molto intuitiva già solo dal nome. Nel primo caso si tratta di un algoritmo in grado di imparare progressivamente e sul momento "da nuovi arrivi", ovvero online, mentre nel secondo abbiamo un algoritmo che studia l'intera storia fino a quel punto ed in questo modo si prepara per affrontare tutti i "futuri arrivi". Con arrivi si intendono informazioni nuove che l'algoritmo dovrà analizzare per svolgere una determinata operazione.

[1] Si tratta di valori tendenzialmente anomali in quanto molto rari e diversi dalla media.

L’algoritmo online è un’ottima soluzione quando si hanno sistemi che ricevono dati costantemente e che necessitano di una capacità di giudizio flessibile e adattabile a forti oscillazioni dei parametri di valutazione. Inoltre, scegliendo un tasso di apprendimento (o *learning rate*) basso, solitamente hanno il vantaggio di evitare di salvare costantemente tutte le informazioni ricevute in memoria in quanto l’algoritmo “se ne può dimenticare” con risparmio di memoria. Tuttavia, il grosso svantaggio dell’avere alimentazioni e apprendimento online con un basso tasso di apprendimento è che in caso di scarsa qualità dei dati di input, l’algoritmo viene molto rapidamente influenzato. Pertanto, bisogna garantire la massima qualità dei dati entranti.

3.1.3 Algoritmi Instance-based vs Model-based

Si fa riferimento al metodo di apprendimento di cui l’algoritmo dispone o per essere più precisi, al modo di *generalizzare*. Gli algoritmi vengono allenati con il training set e poi vengono utilizzati su nuove istanze, mai viste prima, in modo che *generalizzino* quanto imparato durante la fase di allenamento.

Nel caso si tratti di *Instance-based*, l’algoritmo imparerà e svolgerà il suo compito *per similitudine,* senza quindi parametri definiti a priori. Ad esempio, dovendo classificare tutti i nuovi clienti come rischiosi o meno, l’algoritmo assegnerà l’etichetta “*rischioso*” a tutti i clienti che presentano caratteristiche *simili* ma non esattamente *uguali* a quelle di precedenti (e verificati) clienti rischiosi. Questa misura di similitudine può essere manipolata per ottenere le performance desiderate. Il Decision Tree è un esempio di questa tipologia di algoritmo.

Un altro approccio è il metodo *Model-based*. In questo caso si crea un modello prestabilito e si insegna l’algoritmo a stimare i parametri del modello che permettono di ottenere le migliori performances (ad esempio la Regressione Lineare).

3.2 Checklist in un progetto di Machine Learning

Si supponga di voler stimare una determinata variabile attraverso l'applicazione di un algoritmo di machine learning. Come si evince non appena si entra in questo mondo, l'applicazione dell'algoritmo finale e l'ottenimento dei risultati è in realtà la parte meno onerosa in termini di tempo e di *effort*. Ci sono una serie di domande alle quali dar risposta, verifiche da fare e controlli da eseguire affinché gli algoritmi possano funzionare prima incluso di prenderli in considerazione.

Per questo motivo si parla spesso di *Checklist del Data Scientist*, ovvero quella "scaletta" di questioni che si devono seguire al fine di assicurare la migliore delle performance.

Di seguito viene elencata la scaletta proposta da Aurélien Géron in [2]:

- *Definire il problema e l'obiettivo*
- *Trovare i dati*
- *Esplorare i dati*
- *Preparare i dati*
- *Provare più modelli e scegliere quelli migliori*
- *Fine-tuning dei modelli scelti*
- *Analizzare e presentare la soluzione*

Solitamente si ha un obiettivo, che spesso si traduce nel trovare la soluzione ad un problema. In questa sede, l'obiettivo è stimare la LGD con algoritmi di machine learning e confrontare i risultati con i metodi classici utilizzati in modo da capire se l'accuratezza della previsione incrementa o meno.

La creazione di un dataset composto da attributi che potrebbero potenzialmente aiutare il nostro algoritmo a meglio prevedere la variabile target è ovviamente un passaggio fondamentale. In questa fase si decide quali dati si necessitano e in quali quantità. Il dataset utilizzato in questa sede è stato gentilmente fornito, con le dovute alterazioni per questioni di

privacy aziendale, dal gruppo Intesa Sanpaolo ed è stato successivamente elaborato dal sottoscritto.

Una volta aventi i dati organizzati in una tabella e formattati in modo appropriato, l'esplorazione di questi è un'ottima pratica per iniziare ad ottenere le prime intuizioni che possano portare in futuro a scegliere l'algoritmo più adatto. Dalle esplorazioni si possono infatti velocemente individuare le variabili categoriche, booleane, numeriche, la quantità di valori *missing* in un determinato attributo, gli *outliers*, le correlazioni, le distribuzioni, le trasformazioni necessarie e propedeutiche al training dell'algoritmo ed ulteriori informazioni utili.

Nei prossimi paragrafi verranno trattati in modo più approfondito i successivi punti della scala.

3.3 Preparazione dei dati

Molti degli algoritmi di machine learning necessitano di dati in alimentazione con un alto livello di qualità e, a volte, anche un determinato format. Ci sono inoltre alcune operazioni che possono indirettamente o direttamente incrementare le performance del modello. Con *Preparazione dei dati*, solitamente si intende:

- *Data Cleaning*
- *Feature Selection*
- *Feature Engineering*
- *Feature Scaling*

3.3.1 Data Cleaning

Quando si fa *Data Cleaning* (*Pulizia dei dati*), solitamente ci si concentra su:

- Ricerca e gestione dei valori *missing* (mancanti);
- Ricerca e gestione dei valori *outliers*.

Molti algoritmi non trattano i valori *missing* e per questo motivo bisogna adottare un approccio che comprometta il meno possibile il contenuto informativo del dataset a disposizione.

L'approccio varia dipendendo dalla situazione e della variabile in questione. Generalmente si possono:

- *Cancellare le sole righe con valori missing*: Questo comporta ovviamente una perdita di informazione. Infatti, cancellando l'intero record si potrebbero perdere le informazioni di altri attributi che invece potrebbero essere valorizzati;
- *Cancellare l'attributo*: Se un attributo ha un'alta percentuale di valori missing, non otterremmo molta informazione da esso in ogni caso, pertanto potrebbe essere una buona scelta cancellare l'attributo direttamente;
- *Sostituire i valori missing*: Un'ottima strategia potrebbe essere quella di stimare il dominio che potrebbero avere i valori missing all'interno di un attributo. Se si tratta di un attributo numerico si potrebbero sostituire i missing con il valore medio, se si tratta di un attributo categorico con la moda. Ovviamente queste operazioni di sostituzione devono essere condotte consapevoli che la distribuzione del valore dell'attributo potrebbe essere impattata sensibilmente al crescere dei valori mancanti sostituiti. Un altro approccio più laborioso è quello di utilizzare gli altri attributi disponibili per stimare il valore missing attraverso l'utilizzo di un algoritmo (Random Forest, Decision Tree, Regressione lineare, ecc).

L'approccio migliore di Data Cleaning è probabilmente una combinazione dei punti precedenti, soprattutto se si tratta di un dataset con numerosi attributi aventi valori mancanti.

Gli *Outliers*, valori che differiscono di molto rispetto agli altri valori presenti nell'attributo di riferimento e possono diventare un problema.

Sono per definizione dei disturbatori delle distribuzioni degli attributi e di conseguenza possono impattare negativamente l'accuratezza con cui l'algoritmo performa.

Bisogna quindi accettarsi che questi siano veritieri e non abbiano una giustificazione tecnica quali errori di compilazione, problemi di formattazione, valori di default, ecc. Se si tratta di un outlier privo di senso in quanto derivante da problemi di data quality, lo si può correggere, rimuovere o sostituire tenendo conto sempre dell'impatto di ogni singola scelta, simile a quanto visto per i valori missing. Se l'esistenza dell'outlier non ha una giustificazione nota, possono essere gestiti tramite standardizzazione, come vedremo più avanti.

3.3.2 Feature Selection e Riduzione della Dimensione

L'obiettivo, come suggerito dal titolo, è quello di selezionare gli attributi che meglio aiutano a stimare la variabile target a dispetto di quelle che invece forniscono un contributo irrilevante.

Avere tante variabili può creare problemi di:

- O*verfitting* (o sovradattamento del modello sui dati del training set con successive performance scadenti nel test set);
- Difficile interpretazione dei risultati;
- Innecessarie lavorazioni (variabili duplicate o molto simili).

È pertanto consigliato, in molti casi, ridurre il numero degli attributi pur tenendo presente che non bisogna mai eliminarli indiscriminatamente. Questi potrebbero contenere informazioni importanti ai fini della stima.

Ci sono vari metodi per selezionare i *migliori* attributi. L'efficienza di ognuno di essi dipende da quelli che sono gli obiettivi e dalle specificità del problema. In questa sede, l'intenzione non è quella di elencare tutti gli approcci esistenti ma soltanto alcuni dei più utilizzati in modo da dare un'idea delle logiche alla base di questo necessario processo.

Il primo e più intuitivo step è quello di rimuovere gli attributi duplicati, in quanto una volta analizzato il primo di essi, il secondo non aggiunge informazioni nuove. Il secondo step è quello di rimuovere gli attributi che assumono soltanto un unico valore: una feature con varianza uguale a zero non dà nessuna informazione che aiuti a stimare la variabile target. Il terzo step, molto vicino al secondo, è quello di analizzare le colonne che hanno una varianza molto piccola: si tratterebbero di variabili che hanno quasi nella totalità dei casi lo stesso valore. Bisogna fare attenzione a non eliminare variabili con varianze "piccole" per definizione (domino compreso fra 0 e 1). Per attributi con un dominio di valori ridotto, ad esempio attributi categorici, invece che la varianza si potrebbe analizzare la distribuzione dei valori. L'intuizione è che se un determinato valore è presente in un'alta percentuale (98%-99%) delle istanze allora l'informazione che quella feature fornisce è pressoché nulla.

Eliminati i casi più basici, si passa ad analizzare alcuni dei metodi più utilizzati, ovvero:

- *Univariate Feature Selection*
- *Modelli Regolarizzati: Regressione Lasso e Ridge*
- *Modelli basati sugli alberi decisionali: Decision Tree e Random Forest*
- *Analisi delle Componenti Principali*
- *Recursive Feature Elimination*

3.3.2.1 Univariate feature selection

Univariate Feature Selection consiste nell'analizzare ogni singolo attributo ed associare a questo un'*importanza* che aiuti a capire qual è la forza della relazione con la variabile target. Il come associare questa importanza dipende dalla tipologia del problema e dalle caratteristiche del dataset. Si possono utilizzare indicatori come il *p-value*, il *Coefficiente di Massima Informazione* (o *Maximal information coefficient* – MIC), la correlazione della

distanza (*distance correlation*), ecc. In caso si trattasse di una regressione, si potrebbe ad esempio utilizzare la *Correlazione di Pearson*:

$$r = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{N}(x_i - \bar{x})^2(y_i - \bar{y})^2} \quad (3.1)$$

Dove

- x_i è l'istanza *i* dell'attributo preso in considerazione;
- $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$ è la media delle istanze dell'attributo preso in considerazione;
- y_i è l'istanza *i* della variabile target;
- $\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$ è la media delle istanze della variabile target.

Per come impostata la formula (3.1), il dominio varia tra -1 e 1. Più ci si avvicina all'unità negativa e più forte è la correlazione negativa: se una delle variabili cresce l'altra in media decresce. La correlazione positiva, vicina all'unità, indica che le due variabili analizzate crescono o decrescono insieme. Se l'indicatore è vicino allo zero allora non esiste correlazione lineare alcuna fra le due variabili.

Se si calcola questa correlazione lineare fra tutte le variabili nella cosiddetta *Matrice delle correlazioni* si può, sia individuare quegli attributi fortemente correlati alla variabile target, sia individuare forti correlazioni fra coppie di attributi che possono portare ad eliminarne uno.

3.3.2.2 Modelli Regolarizzati: Regressione Lasso e Ridge

I Modelli Regolarizzati semplicemente aggiungono ad un problema di ottimizzazione una penale sui coefficienti degli attributi in modo da forzare quelli poco partecipi nella stima verso lo zero. Di conseguenza, è intrinseca nell'algoritmo la ricerca delle variabili più "pesate" e quindi più importanti.

I due algoritmi più utilizzati sono la *Regressione Ridge* e la *Regressione Lasso*. Entrambi aggiungono una penalità al problema di ricerca dei minimi

quadrati (*Ordinary Least Square - OLS*) utilizzato per costruire la comune *Regressione Lineare*. La differenza sostanziale fra i due algoritmi è la forma che assume la penalità. Per esplicitarlo in formule:

$$OLS = \sum_{i=1}^{N} (y_i - \hat{y_i})^2 = \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{M} \beta_j x_{ij} \right)^2 \quad (3.2)$$

Regressione Ridge (RR)

$$RR = OLS + \lambda \sum_{j=1}^{m} \beta_j^2 \quad (3.3)$$

Regressione Lasso (RL)

$$RL = OLS + \lambda \sum_{j=1}^{m} |\beta_j| \quad (3.4)$$

Con

- x_{ij} è l'istanza *i* dell'attributo *j*;
- y_i è l'osservazione *i* reale della variabile target;
- $\hat{y_i}$ è la stima *i* della variabile target;
- β_0, β_j sono rispettivamente la costante rispetto all'origine e *i* coefficienti/pesi degli attributi *j*;
- λ penalità assegnata ai pesi β_j

L'aggiunta della penalità λ, durante la minimizzazione successiva, forza i coefficienti β meno influenti verso lo zero. Come già detto, la differenza fra i due algoritmi sta proprio nell'influenza che la penalità assume in questa forzatura. Se $\lambda = 0$ evidentemente ci troviamo a fare una normale Regressione Lineare. Regolare λ significa fare una scelta fra quanta *varianza* vogliamo permetterci rispetto ai *bias* (errori nella stima), nella costruzione del modello. Infatti, aumentando λ di molto otterremmo moltissimi coefficienti vicino allo zero che ci forniranno quindi una varianza inferiore ma un maggiore bias nella stima in quanto "scartiamo" informazioni. L'azzeramento totale dei coefficienti degli attributi viene

raggiunto con più discrezione nel modello Ridge rispetto al Lasso, nel quale l'esclusione degli attributi aumenta di molto all'aumentare di λ. Calcolando i pesi penalizzati dei singoli attributi si riesce a capire quali di questi siano più importanti.

3.3.2.3 Modelli basati sugli alberi decisionali

Gli alberi decisionali e quindi anche il Random Forest, che saranno trattati nelle prossime pagine, hanno una selezione implicita delle variabili più importanti durante il loro apprendimento. Gli attributi più importanti verranno scelti dal modello per costruire i nodi decisionali più vicini alle radici, mentre gli attributi meno importanti saranno quelli utilizzati dai nodi vicini alle foglie. In questo modo si può assegnare un'importanza ad ogni attributo in base alla posizione che l'attributo assume all'interno dell'albero decisionale (o alla media delle posizioni degli attributi negli alberi costituenti la Random Forest).

3.3.2.4 Analisi delle componenti principali

L'*Analisi delle Componenti Principali*, più conosciuta come PCA (*Principal Component Analysis*), è uno degli algoritmi più utilizzati per ridurre la dimensione di un dataset. Prima di entrare nel dettaglio matematico di come funziona l'algoritmo, si cerca di spiegare quella che è l'idea.

Avendo un dataset composto da *m* attributi che creano un iperspazio di *D* dimensioni, la PCA "proietta" questi *m* attributi in un altro sistema (o iperpiano) che viene costruito in modo da massimizzare la varianza contenuta nel sistema originario con un minor numero di dimensioni. La prima dimensione (prima componente principale) del nuovo sistema verrà costruita come combinazione lineare degli attributi con maggior varianza del sistema originario, la seconda dimensione (seconda componente principale

ed ortogonale alla prima) verrà costruita cercando di catturare la varianza residua per quanto più possibile, così via con la terza ed ulteriori dimensioni.

In questo modo si potrebbe passare, come empiricamente verificatosi in molti casi, da un dataset con molte dimensioni (attributi), ad esempio 20, ad un dataset, combinazione lineare del primo, con un numero inferiore di dimensioni, ad esempio 5, che catturano un'alta percentuale della varianza del primo sistema, ad esempio 95%. Si utilizza cosi un dataset con 5 attributi sacrificando una bassa percentuale di informazione.

Provando a dare una formulazione matematica, si consideri un dataset con x_n osservazioni avente dimensioni *D.* L'obiettivo è quello di proiettare i dati in un altro spazio di dati che abbia una dimensione *M* < *D*, massimizzando la varianza. Supponendo che *M* sia conosciuto e che sia (per comodità di esposizione) pari a *M = 1*, si può definire la direzione dello spazio con un *versore* $\boldsymbol{u_1}$ (quindi tale per cui $\boldsymbol{u_1^T u_1 = 1}$). Ogni istanza x_n è proiettata dunque in un valore scalare $\boldsymbol{u_1^T x_n}$. La media di questi dati proiettati è $\boldsymbol{u_1^T \overline{x}}$ con $\overline{\boldsymbol{x}} = \frac{\boldsymbol{1}}{\boldsymbol{N}} \sum_{\boldsymbol{n=1}}^{\boldsymbol{N}} \boldsymbol{x_n}$ e di conseguenza la varianza risulta essere:

$$\frac{\boldsymbol{1}}{\boldsymbol{N}} \sum_{\boldsymbol{n=1}}^{\boldsymbol{N}} \{\boldsymbol{u_1^T x_n} - \boldsymbol{u_1^T \overline{x}}\}^{\boldsymbol{2}} = \boldsymbol{u_1^T S u_1} \tag{3.5}$$

$$con\ \boldsymbol{S} = \frac{\boldsymbol{1}}{\boldsymbol{N}} \sum_{\boldsymbol{n=1}}^{\boldsymbol{N}} (\boldsymbol{x_n} - \overline{\boldsymbol{x}})(\boldsymbol{x_n} - \overline{\boldsymbol{x}})^{\boldsymbol{T}} \tag{3.6}$$

S è la matrice di covarianza. Volendo massimizzare la varianza rispetto a $\boldsymbol{u_1}$ bisogna innanzi tutto vincolare con $\boldsymbol{u_1^T u_1 = 1}$ per evitare $\|\boldsymbol{u_1}\| \to \infty$**.** Questo vincolo viene introdotto tramite il moltiplicatore di Lagrange λ_1:

$$\boldsymbol{u_1^T S u_1} + \lambda_1(1 - \boldsymbol{u_1^T u_1}) \tag{3.7}$$

Facendo le derivate parziali rispetto a $\boldsymbol{u_1}$ e λ_1 si ottiene:

$$\boldsymbol{S u_1} = \lambda_1 \boldsymbol{u_1} \tag{3.8}$$

Ovvero $\boldsymbol{u_1}$ è un *autovettore* di $\boldsymbol{S}$. Moltiplicando a sinistra per $\boldsymbol{u_1^T}$ e ricordando che si tratta di un versore si ottiene:

$$\boldsymbol{u_1^T S u_1} = \lambda_1 \tag{3.9}$$

Il che significa che la varianza sarà massima quando si uguaglia $\boldsymbol{u_1}$ con l'autovettore avente l'autovalore λ_1. Questo autovettore si riconosce come la prima componente principale. Si può successivamente individuare la seconda componente principale definendo il versore $\boldsymbol{u}_2$, ortogonale a $\boldsymbol{u}_1$, con direzione tale da massimizzare la varianza dei dati proiettati. In generale, si otterrà un insieme di *M* componenti principali (autovettori) $\boldsymbol{u}_1, \dots \boldsymbol{u}_M$ della matrice di covarianza $\boldsymbol{S}$ corrispondente ai *M* più grandi autovalori $\lambda_1, \dots, \lambda_M$. Il che significa che l'analisi delle componenti principali altro non è che il calcolo della media dei dati $\overline{\boldsymbol{x}}$ e della matrice di covarianza $\boldsymbol{S}$ per trovare gli *M* autovettori di $\boldsymbol{S}$ che corrispondano agli autovalori più grandi.

Nonostante sembri un approccio estremamente conveniente e quindi d'attuare tutte le volte che si ha un dataset sufficientemente grande, la PCA dev'essere utilizzata con attenzione. Anche se può essere tecnicamente applicata su variabili non continue, questo non è consigliato. Queste variabili assumono un significato soltanto nel sistema originale. Inoltre, non bisogna confondere le componenti principali con l'importanza degli attributi. La prima componente non prende in considerazione gli attributi *più importanti* ma quelli *più variabili.*

3.3.2.5 Recursive Feature Elimination

Come dice il nome stesso, *Recursive Feature Elimination – RFS*, si tratta di un processo di eliminazione progressiva di attributi. I passi logici si possono sintetizzare come di seguito:

- Si sceglie un algoritmo *tester* che intrinsecamente valuti gli attributi fornendo una misura dell'*importanza* di essi (come ad esempio, il Random Forest);
- Si allena l'algoritmo sul set di attributi presi in considerazione e si ricavano le misure di importanza;
- Si eliminano gli attributi ritenuti non rilevanti dall'algoritmo scelto (si possono eliminare singolarmente o a gruppi per ogni iterazione);
- Si ripete il processo considerando il nuovo subset di variabile.

Il numero di attributi desiderati può essere definito in partenza oppure può non essere conosciuto a priori. In quest'ultimo caso il RFS consiglia come subset migliore la combinazione di attributi che ha fornito le previsioni più accurate.

Ovviamente la valutazione dell'importanza delle variabili dipende dall'algoritmo che si utilizza per testare. Questo metodo viene molto spesso combinato con la metodologia di validazione detta *Cross-Validation*, spiegata successivamente nel paragrafo 3.5.

3.3.3 Feature Engineering

Il processo di *Feature Engineering* si pone l'obiettivo generico di trasformare, aggregare e/o inventare le variabili del set utilizzato con lo scopo di migliorare la qualità della predizione.

Alcuni esempi di feature engineering sono:

- *Binning*: Si tratta, in poche parole, di ridurre il dominio di un attributo raggruppando in *Bin*, ovvero un intervallo che contiene in sé più valori del dominio originale. Alcuni attributi, pur avendo un dominio continuo, possono essere resi variabili categoriche. Ad esempio, avendo un attributo con la temperatura dell'acqua (a pressione atmosferica) per ogni istanza, si potrebbe essere interessati ad estrapolare lo stato della materia assunto categorizzando tutte le osservazioni con temperatura fino ai 0 °C come "Solido", da 1 °C a 100 °C come "Liquido" e superiori a 100

°C come “Gassoso”. Allo stesso modo si potrebbe ridurre il dominio di una variabile categorica trasformandola in un'altra variabile categorica. Avendo un dataset contenente l’attributo “Paese” si potrebbe pensare di trasformarlo in “Continente” riducendo il dominio dell’attributo da centinaia di valori a soltanto cinque[1]. Si veda la Tabella 3.1 per un esempio;

- *Scomposizione di variabili*: Ci sono alcuni attributi che possono essere scomposti in più attributi per meglio rappresentare il contenuto informativo di essi. L’esempio più immediato è la trasformazione *One Hot Encode*: si supponga di avere sempre la variabile categorica sullo stato della materia dell’acqua, Solido, Liquido e Gassoso, si potrebbe voler rappresentare questi tre stati come tre diversi attributi *Booleani* che assumono il valore 1 laddove la materia abbia quello specifico stato e 0 in tutte le altre osservazioni. Un ulteriore esempio è la scomposizione delle date in tre diversi attributi: giorno, mese ed anno. Si veda la Tabella 3.1 per un esempio

Tabella 3.1: Esempio di Binning e di One Hot Encode

Dataset di partenza		*Binning da “Temperatura°C”*	*One Hot Encode dell’attributo “Stato”*		
ID Istanza	**Temperatura °C**	**Stato**	**Liquido**	**Solido**	**Gassoso**
1	25	Liquido	1	0	0
2	120	Gassoso	0	0	1
3	-10	Solido	0	1	0

- *Trasformazione di variabili presenti*: Una variabile cosi per come presentata potrebbe non risultare in apparenza correlata con la variabile da stimare, dunque potrebbe anche essere scartata. Tuttavia, appena si applica una trasformazione matematica, come il logaritmo, la radice quadrata, l’esponenziale o ancora altre operazioni, ci si potrebbe accorgere della effettiva valenza ai fini della stima. È buona pratica quindi aggiungere al dataset di partenza variabili che siano funzioni matematiche di variabili già presenti;

[1] Secondo la convenzione più comune.

- *Combinazioni di più variabili*: Quasi una generalizzazione del punto precedente, a volte risulta utile combinare più variabili all'interno del dataset per crearne una nuova che aiuti nella stima meglio delle singole componenti. È facile capire che l'altezza media di un piano moltiplicato per il numero di piani di un edificio aiuta molto di più nello stimare l'altezza totale dell'edificio che non le singole informazioni per separato.

3.3.4 Feature Scaling: Normalizzazione e Standardizzazione

Quando si hanno variabili che hanno domini con scale molto diverse, una buona pratica, che migliora in molti dei casi l'accuratezza della stima, è quella di *Normalizzare* o *Standardizzare* gli attributi. In molti contesti le due operazioni vengono prese come la medesima, in questa sede si intendono come due operazioni molto diverse.

La *Normalizzazione*, chiamatasi anche *min-max scaling*, ha lo scopo di condurre tutti i possibili valori di un attributo all'intervallo 0-1. In questo modo, l'algoritmo utilizzato potrebbe confrontare variabili come l'Età e il Reddito Annuale Lordo più facilmente. Questo lo si fa solitamente applicando la seguente formula:

$$X_i\ normalizzato = \frac{x_i - x_{min}}{x_{max} - x_{min}} \tag{3.10}$$

Dove,

- x_i è la istanza *i* dell'attributo preso in considerazione;
- x_{min} è il valore minimo dell'intero dominio dell'attributo;
- x_{max} è il valore massimo dell'intero dominio dell'attributo.

Questa trasformazione non cambia la distribuzione dell'attributo ed è infatti consigliabile quando non conosciamo quest'ultima. Prima di normalizzare bisogna aver eliminato e/o gestito eventuali *outliers* presenti nel dominio in quanto la trasformazione è estremamente sensibile a questi. Avendo, ad esempio, 99 numeri compresi fra 1 e 20, e soltanto un numero

uguale a 100, dopo la normalizzazione si avranno 99 valori compresi fra 0 e 0,4 perché verrebbe considerato l'outlier "100" come il x_{max}.

La *Standardizzazione* è invece un'operazione che porta l'attributo ad avere un valore medio uguale a zero ed una deviazione standard pari a 1. Si utilizza la seguente formula:

$$Z_i = \frac{x_i - \mu}{\theta} \tag{3.11}$$

Dove,

- x_i è la istanza *i* dell'attributo preso in considerazione;
- $\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$ è la media del dominio dell'attributo;
- $\theta = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$ è la deviazione standard del dominio dell'attributo.

La standardizzazione quindi non conduce i valori degli attributi verso un predeterminato intervallo come nel caso della normalizzazione e questo è un problema per alcuni algoritmi, come alcune reti neurali che richiedono in input variabili con valori nell''intervallo [0,1]. Tuttavia, non soffre della presenza di *outliers* come la normalizzazione ed è anzi un modo alternativo di trattarli in quanto ne riduce l'impatto.

3.4 Scelta del Modello

I modelli di machine learning utilizzabili sono molteplici e ancora più numerose sono le combinazioni che si possono fare, sia di parametri all'interno di un singolo modello che fra modelli differenti. Pedro Domingos nel suo libro *"L'Algoritmo Definitivo: La macchina che impara da sola e il futuro del nostro mondo"*[1], dopo aver ampiamente spiegato le cinque diverse "*tribù*" esistenti fra i data scientist (*Connessionisti*, *Simbolisti*,

[1] [39] Pedro Domingos, *"L'Algoritmo Definitivo: La macchina che impara da sola e il futuro del nostro mondo"*, Bollati Boringhieri, 2016.

Bayesiani, *Evoluzionisti* ed *Analogisti*), che differiscono principalmente per il credere più in specifici modelli di machine learning che in altri, ipotizza l'esistenza di un algoritmo unificante che sia adattabile ad ogni tipologia di problema (appunto, "Definitivo"). I connessionisti, ad esempio, credono ampiamente che tutto possa essere risolto con le reti neurali mentre gli evoluzionisti puntano tutto sugli algoritmi genetici. Secondo Domingos si sbagliano entrambi in quanto, come provato, alcuni algoritmi sono più efficienti di altri in base alle caratteristiche del problema e nessuno è in modo assoluto superiore agli altri. Nell'attesa che si trovi l'Algoritmo Definitivo ipotizzato da Domingos, che sia quindi in grado di risolvere indiscriminatamente qualsiasi problema, il data scientist odierno deve armarsi dei migliori strumenti offerti dalle singole tribù e capire, problema per problema, quale di questi è più performante date le circostanze.

Trattare tutti gli algoritmi esistenti non è lo scopo di questa tesi. Si cerca invece di confrontare alcuni di questi con i classici modelli di regressione lineare, in utilizzo preso le banche nel calcolo della LGD, per capire i potenziali utilizzi. Sono dunque stati scelti quattro diversi modelli che rappresentano tuttavia dei *"must"* nel bagaglio nozionale di un comune data scientist:

- *Support Vector Machine*
- *Decision Tree*
- *Random Forest*
- *Reti Neurali Artificiali*

3.4.1 Support Vector Machine

Le Macchine a Vettori di Supporto (o Support Vector Machine – SVM) sono dei sofisticati algoritmi di machine learning che vengono ampiamente utilizzati sia per problemi di classificazione, sia per problemi di regressione. In questa sede si vedranno prima le logiche utilizzate per la classificazione

che semplificheranno la comprensione di quelle utilizzate per la regressione.

Quando si tratta di classificare, le SVM cercano di tracciare una linea (o iperpiano, per essere precisi) nell'iperspazio che delimiti le diverse categorie entro le quali ogni istanza può ricadere. In questo modo, si riconosce facilmente che "al di là della linea" si avrà una determinata classe mentre "al di qua" se ne avrà un'altra. Questo funziona sia per due o più dimensioni, sia per gruppi linearmente e non-linearmente separabili.

Si supponga, ad esempio, di avere un dataset di due attributi linearmente separabili come nella **Figura 3.1**:

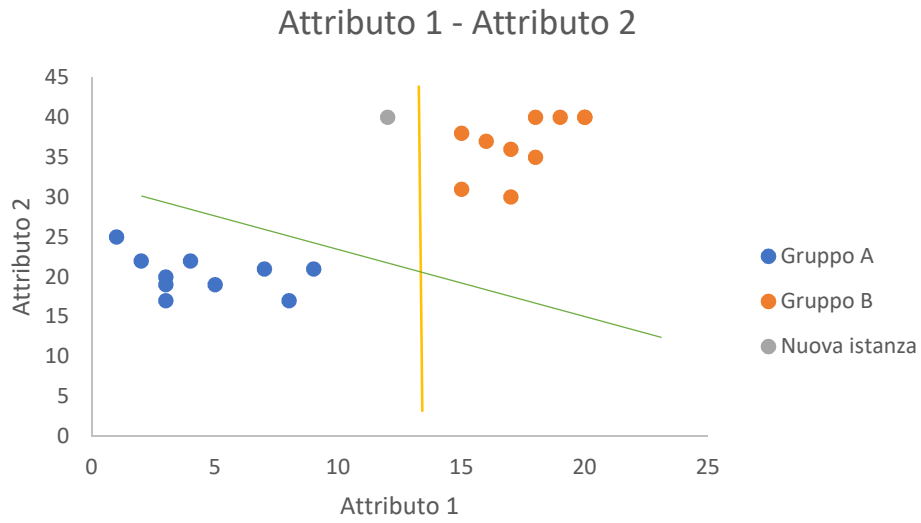


Figura 3.1: Classificazione Gruppo A e Gruppo B in base agli attributi 1 e 2.

Si nota facilmente che osservando la coppia *Attributo 1 – Attributo 2* si definiscono (anche visivamente) due gruppi diversi: l'insieme dei punti in basso a sinistra (blu) e quello in alto a destra (arancione). Ci sono infinite rette che in un esempio così semplice riuscirebbero a dividere efficacemente i due insiemi. La retta gialla e la retta verde sono due degli infiniti esempi. Tuttavia, aggiungendo nuovi punti al grafico (punto grigio), si potrebbe scoprire come molte di queste rette non siano tanto più adatte a

classificare (retta gialla) la nuova istanza che, molto probabilmente, appartiene al gruppo B. Ci sono dunque infinite soluzioni, vero, ma alcune sono migliori di altre.

La logica delle SVM è quella di trovare la retta di confine fra i due gruppi che massimizzi la distanza fra questi. In questo modo si presuppone di ridurre gli errori nelle classificazioni future presupponendo che più *distanti* siano due *oggetti* più *diversi* siano. Per dirlo in parole povere (o nel comune linguaggio), si cerca di trovare la *strada* che abbia la massima *ampiezza* senza includere nessuna delle istanze al suo interno, come mostrato in **Figura 3.2**:

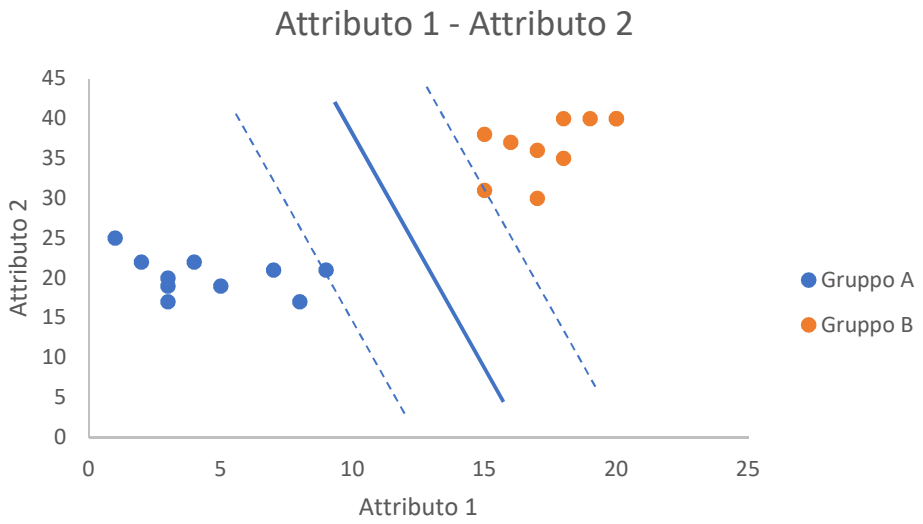


Figura 3.2: Classificazione con le SVM

Le due rette tratteggiate rappresentano i margini della strada mentre la retta centrale è il vero confine fra i gruppi. I margini sono visti come la distanza fra l'istanza di un gruppo e il confine che li separa dall'altro ed è la grandezza che si vuole massimizzare. Pertanto, se ci sono delle nuove istanze, queste verranno classificate come Gruppo A o Gruppo B facendo riferimento alla retta centrale della *strada*. Le istanze che avranno una distanza *in negativo* (verso sinistra) rispetto alla retta centrale e superiore al margine saranno classificate come Gruppo A; le istanze che avranno una

distanza *in positivo* (verso destra) rispetto alla retta centrale e superiore al margine saranno classificate come Gruppo B.

3.4.1.1 SVM per Classificazione - Dati linearmente separabili

Cercando di darne una forma matematica, quanto espresso precedentemente si traduce nella risoluzione del problema seguente:

$$\begin{cases} \boldsymbol{w} \cdot \boldsymbol{x}_i + b \geq 1 & per\ y_i = +1 \\ \boldsymbol{w} \cdot \boldsymbol{x}_i + b \leq -1 & per\ y_i = -1 \end{cases} \tag{3.12}$$

Dove

- $\boldsymbol{w}$ è un vettore perpendicolare alla retta di separazione dei gruppi;
- $\boldsymbol{x}_i$ è il vettore che rappresenta l'istanza "i";
- b è la costante intercetta all'origine;
- y_i è il label dell'istanza *"i"*, assume il valore +1 se l'istanza appartiene al Gruppo B e -1 se appartiene al Gruppo A, nell'esempio grafico delle **Figure 3.1** e **3.2**.

Si vorrebbe dunque trovare **w** e b, ovvero i parametri che definiscono il confine fra le classi e che permetterebbero di classificare nuove istanze. Sostituendo la funzione:

$$y = \begin{cases} +1 & se\ gruppo\ B \\ -1 & se\ gruppo\ A \end{cases}$$

nella (3.12), si ottiene:

$$y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) - 1 \geq 0 \quad \forall i \tag{3.13}$$

Considerando due istanze $\boldsymbol{x}_A, \boldsymbol{x}_B$ posizionate rispettivamente sui margini della strada dei gruppi A e B, si può ricavarci la larghezza della strada come:

$$Larghezza = \frac{(\boldsymbol{x}_A - \boldsymbol{x}_B) \cdot \boldsymbol{w}}{\|\boldsymbol{w}\|} \tag{3.14}$$

Ricavando $\boldsymbol{x}_i$ in (3.13), sostituendo in (3.14), facendo attenzione ai segni e ricordando che $\boldsymbol{x}_A, \boldsymbol{x}_B$ sono istanze nei margini, per cui la disuguaglianza (3.13) diventa eguaglianza, si ottiene:

$$Larghezza = \frac{1 - b + 1 + b}{\|\boldsymbol{w}\|} = \frac{2}{\|\boldsymbol{w}\|} \tag{3.15}$$

Massimizzare la larghezza per come espressa in (3.15) è lo stesso problema di minimizzare $\|\boldsymbol{w}\|$. A sua volta, minimizzare $\|\boldsymbol{w}\|$ è come minimizzare $\frac{1}{2}\|\boldsymbol{w}\|^2$, per cui verrà fatta questa trasformazione che facilità l'analisi matematica in quanto permetterà di risolvere il problema come un problema di *programmazione quadratica*.

La funzione obbiettivo vincolata diventa quindi:

$$min_{w,b}\ \frac{1}{2}\|\boldsymbol{w}\|^2 \tag{3.16}$$

$$y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b)\ \geq\ 1\quad \forall i \tag{3.17}$$

Che può essere risolta con i moltiplicatori di Lagrange sia per la facilitazione dei calcoli sia per la possibilità di esprimere i dati di training come prodotto scalare e quindi, come si vedrà successivamente, di estendere il ragionamento anche a problemi non lineari:

$$L_p(\boldsymbol{w}, b, \alpha) = \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{N} \alpha_i \left[y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b)\right] + \sum_{i=1}^{N} \alpha_i \tag{3.18}$$

Dove α_i sono i moltiplicatori di Lagrange

Si minimizza rispetto a $\boldsymbol{w}$ e b e si massimizza rispetto a α, rispettando sempre la condizione $\alpha_i \geq 0$. La Lagrangiana così formata rappresenta un problema di *programmazione quadratica convessa*, visto che la funzione da minimizzare risulta convessa e convessi sono anche i vincoli. Possiamo dunque risolvere anche il problema duale: massimizzare la Lagrangiana

eguagliando a zero le derivate parziali rispetto a $\boldsymbol{w}$ e b sotto il vincolo dei moltiplicatori positivi[1].

Derivando rispetto a $\boldsymbol{w}$ e b ed uguagliando a zero si ottiene:

$$\frac{\partial L_p}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{1=1}^{N} \alpha_i y_i \boldsymbol{x}_i = 0 \rightarrow \boldsymbol{w} = \sum_{1=1}^{N} \alpha_i y_i \boldsymbol{x}_i \tag{3.19}$$

$$\frac{\partial L_p}{\partial b} = \sum_{1=1}^{N} \alpha_i y_i = 0 \tag{3.20}$$

Sostituendo (3.19) in (3.18) si ottiene:

$$L_d(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j \tag{3.21}$$

$L_d(\alpha)$ va massimizzata rispetto alla variabile α rispettando i vincoli di positività dei moltiplicatori e (3.20). Il training test quindi sarà utilizzato per massimizzare la Lagrangiana rispetto ai moltiplicatori e ricavare le soluzioni sostituendoli nella (3.19). È dimostrabile che questo tipo di ottimizzazione rispetti le condizioni di *Karush-Kuhn-Tucker* (KKT), tra le quali troviamo:

$$\alpha_i [y_i (\boldsymbol{w} \cdot \boldsymbol{x}_i + b) - 1] = 0 \tag{3.22}$$

Che può essere rispettata soltanto se i moltiplicatori sono nulli oppure se quanto espresso dentro le quadra è nullo. Questo secondo caso si verifica, come visto in precedenza, solo se $\boldsymbol{x}_i$ si trova in uno dei margini della *strada* rappresentata graficamente. Pertanto, si conclude che se $\alpha_i > 0$ allora $\boldsymbol{x}_i$ è un vettore di supporto. Per calcolare $\boldsymbol{w}$ e b, dopo opportuni passaggi matematici, si ottiene:

[1] Detto anche problema duale di Wolfe. [40] Philip Wolfe (1961). "A duality theorem for non-linear programming". Quarterly of Applied Mathematics. 19: 239–244.

$$\boldsymbol{w} = \sum_{x \in SV}^{N_{sv}} \alpha_i y_i \boldsymbol{x}_i \tag{3.23}$$

$$b = \frac{1}{N_{SV}} \sum_{x_i \in SV}^{N_{sv}} (y_i - \sum_{x_j \in SV}^{N_{sv}} \alpha_j y_j \boldsymbol{x}_j \cdot \boldsymbol{x}_i) \tag{3.24}$$

con N_{SV} il numero di vettori di supporto e la giusta indicazione che per b si preferisce calcolare il b medio invece che basarsi sul b dedotto dal singolo vettore di supporto.

La classificazione appena evidenziata viene detta *hard margin classification* perché non si ammettono violazioni dei margini. In presenza di dati non linearmente separabili o in presenza di *outliers,* che ridurrebbero di molto l'ampiezza della strada, come raffigurato in **Figura 3.3**, sarebbe conveniente ammettere un numero contenuto di violazioni con il fine di ampliare i margini per migliorare la performance del modello. Si parla di *soft margin classification*.

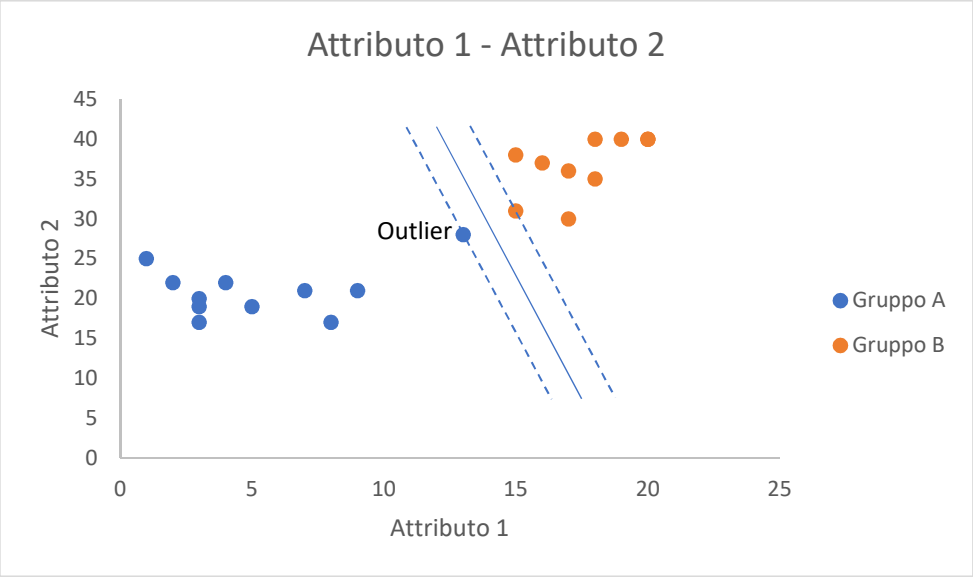


Figura 3.3: Impatto degli outliers sulla classificazione SVM

Fare questo, matematicamente significa ammettere che si possono avere delle violazioni dei margini. Questo errore viene registrato nelle cosiddette

variabili di slack ε_i, che saranno quindi valorizzate soltanto se l'istanza considerata viola i margini. Pertanto, le condizioni (3.12) diventano:

$$\begin{cases} \boldsymbol{w} \cdot \boldsymbol{x}_i + b \geq 1 - \varepsilon_i & per\ y_i = +1 \\ \boldsymbol{w} \cdot \boldsymbol{x}_i + b \leq -1 + \varepsilon_i & per\ y_i = -1 \end{cases} \tag{3.25}$$

E la (3.13), diventa:

$$y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) - 1 + \varepsilon_i \geq 0 \quad \forall i, \varepsilon_i > 0 \tag{3.26}$$

Incorporando le variabili slack nella funzione obiettivo, si ottiene:

$$min\ \frac{1}{2}\|\boldsymbol{w}\|^2 + C\left(\sum_{i=1}^{N} \varepsilon_i\right)^k \tag{3.27}$$

$$y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b)\ \geq\ 1 - \varepsilon_i \quad \forall i, \varepsilon_i \geq 0 \tag{3.28}$$

Dove,

- *C* è la penalità che si vuole assegnare agli errori;
- *k* è il grado che si vuole dare agli errori. Per qualsiasi k si tratta di un problema di programmazione convessa; per k = 2 o k= 1 è inoltre un problema di programmazione quadratica.

La (3.18) diventa:

$$L_p = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N} \varepsilon_i - \sum_{i=1}^{N} \alpha_i\left[y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) - 1 + \varepsilon_i\right] - \sum_{i=1}^{N} \beta_i \varepsilon_i \tag{3.29}$$

E derivando rispetto a $\boldsymbol{w}$, b e ε_i si ottengono:

$$\frac{\partial L_p}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{1=1}^{N} \alpha_i y_i \boldsymbol{x}_i = 0 \rightarrow \boldsymbol{w} = \sum_{1=1}^{N} \alpha_i y_i \boldsymbol{x}_i \tag{3.30}$$

$$\frac{\partial L_p}{\partial b} = \sum_{1=1}^{N} \alpha_i y_i = 0 \tag{3.31}$$

$$\frac{\partial L_p}{\partial \varepsilon_i} = C - \alpha_i - \beta_i = 0 \tag{3.32}$$

E facendo l'analoga sostituzione fatta per il caso di classificazione *hard margin*, di (3.30)-(3.32) nella (3.29) si ottiene:

$$L_d(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j \qquad (3.33)$$

Da massimizzare rispetto ad α_i rispettando il vincolo $0 \leq \alpha_i \leq C$. Quindi l'unica differenza rispetto al caso dei dati linearmente separabili è l'imposizione di un limite superiore dei moltiplicatori α_i rappresentato dal parametro di penalizzazione *C*.

3.4.1.2 SVM per Classificazione: dati non linearmente separabili

Regolare il parametro *C* è un buon modo di superare gli eventuali outliers presenti in un dataset composto di dati linearmente separabili. Per affrontare dati non linearmente separabili con distribuzione più complesse, in generale, è meglio utilizzare le *funzioni Kernel*, che vengono genericamente rappresentate come:

$$K(\boldsymbol{x}, \boldsymbol{x}') = \Phi(\boldsymbol{x}) \cdot \Phi(\boldsymbol{x}') \qquad (3.34)$$

Come visto nella (3.21) o (3.33), un passaggio fondamentale è il prodotto scalare fra i vettori $\boldsymbol{x}_i \cdot \boldsymbol{x}_j$. Quest'ultimo può essere sostituito da una funzione Kernel che ha lo scopo di moltiplicare due vettori in uno spazio *Z* diverso da quello rappresentato dai dati X di input, senza interessarsi delle trasformazioni fra i due spazi. Il desiderio è che in questo nuovo spazio Z i dati siano linearmente separabili. Per fare un banale esempio grafico del concetto appena espresso si pensi di avere, in un piano cartesiano, dei dati *x* distribuiti sulla retta *y= 3* come rappresentato in **Figura 3.4**:

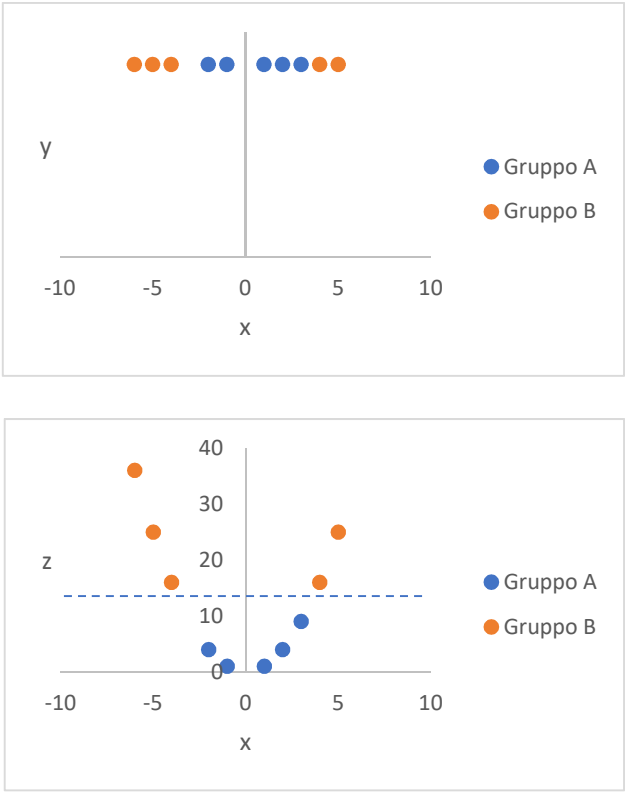


Figura 3.4: Esempio grafico di trasformazione Kernel

Come si vede nel grafico superiore di **Figura 3.4**, i gruppi non sono linearmente separabili. Tuttavia, se si applica la trasformazione $z(x) = x^2$, come visualizzato nel grafico inferiore, diventano facilmente e linearmente separabili.

Alcune delle funzioni Kernel più utilizzate, e quindi anche ammissibili (in quanto rispettano determinate condizioni, dette *condizioni di Mercer*), sono:

- Kernel Lineare:

$$K(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{x} \cdot \boldsymbol{x}' \tag{3.35}$$

- Kernel Polinomiale ($p \in \mathbb{N}$, $c > 0$)

$$K(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x} \cdot \boldsymbol{x}' + c)^p \tag{3.36}$$

- Kernel Gaussian RBF (*Radial Basis function*):

$$K(\boldsymbol{x}, \boldsymbol{x}') = e^{-\frac{\|\boldsymbol{x}-\boldsymbol{x}'\|^2}{2\sigma^2}} \tag{3.37}$$

3.4.1.3 SVM per Regressione

Volendo utilizzare i vettori di supporto per affrontare un problema di regressione, invece che di classificazione, la logica si inverte. Invece di massimizzare l'ampiezza della strada cercando di lasciare fuori quanti più punti possibili, ora si cerca di includere dentro i margini quanti più punti possibili. Si vuole dunque trovare una funzione che abbia un grado di precisione ξ e che approssimi la distribuzione dei dati. Si avrà pertanto il modello di regressione lineare:

$$p(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x} + b \tag{3.38}$$

Vincolato con

$$|y_i - \boldsymbol{w} \cdot \boldsymbol{x}_i - b| \leq \xi \tag{3.39}$$

Visto che non è sempre possibile ottenere delle previsioni con un errore massimo ξ ,si introducono le variabili di slack $\varepsilon \geq 0$ e $\varepsilon^* \geq 0$, con ε l'errore per punti $\boldsymbol{x}_i$ tali per cui $y_i > p(\boldsymbol{x}_i) + \xi$ e ε^* l'errore per punti $\boldsymbol{x}_i$ tali per cui $y_i < p(\boldsymbol{x}_i) - \xi$. Si arriva, come fatto precedentemente a:

$$min\ \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N}(\varepsilon_i + \varepsilon_i^*) \tag{3.40}$$

$$con\ vincoli \begin{cases} y_i - \boldsymbol{w} \cdot \boldsymbol{x}_i - b \leq \xi + \varepsilon_i \\ \boldsymbol{w} \cdot \boldsymbol{x}_i - y_i + b \leq \xi + \varepsilon_i^* \end{cases} \tag{3.41}$$

La costante *C* svolge lo stesso lavoro di regolazione fatto in (3.27). Andando a formulare questo problema utilizzando i moltiplicatori di Lagrange, si ottiene:

$$L = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N}(\varepsilon_i + \varepsilon_i^*) - \sum_{i=1}^{N}\alpha_i\left(\varepsilon_i + \varepsilon_i^* - y_i + \boldsymbol{w}\cdot\boldsymbol{x}_i + b\right)$$
$$- \sum_{i=1}^{N}\alpha_i^*\left(\varepsilon_i + \varepsilon_i^* + y_i - \boldsymbol{w}\cdot\boldsymbol{x}_i - b\right)$$
$$- \sum_{i=1}^{N}(\varepsilon_i\beta_i + \varepsilon_i^*\beta_i^*) \quad con\ i\ moltiplicatori\ \ \beta_i, \beta_i^*, \alpha_i, \alpha_i^* \geq 0$$

(3.42)

Derivando la Lagrangiana rispetto alle variabili $\varepsilon_i, \varepsilon_i^*, \boldsymbol{w}, b$ ed uguagliando a zero si ottengono le condizioni di ottimo:

$$\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{w} - \sum_{1=1}^{N}(\alpha_i - \alpha_i^*)\boldsymbol{x}_i = 0 \rightarrow \boldsymbol{w} = \sum_{1=1}^{N}(\alpha_i - \alpha_i^*)\boldsymbol{x}_i \qquad (3.43)$$

$$\frac{\partial L}{\partial b} = \sum_{1=1}^{N}(\alpha_i - \alpha_i^*) = 0 \qquad (3.44)$$

$$\frac{\partial L}{\partial \varepsilon_i} = C - \alpha_i - \beta_i = 0 \qquad (3.45)$$

$$\frac{\partial L}{\partial \varepsilon_i^*} = C - \alpha_i^* - \beta_i^* = 0 \qquad (3.46)$$

Sostituendo le condizioni appena ottenute nella (3.42) ed introducendo la funzione generica di Kernel $K(\boldsymbol{x}, \boldsymbol{x}')$,si ottiene:

$$L(\boldsymbol{x}_i, \boldsymbol{x}_j) = -\xi\sum_{i=1}^{N}(\alpha_i + \alpha_i^*) - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$
$$+ \sum_{i=1}^{N} y_i(\alpha_i - \alpha_i^*)$$

$$con\ vincoli \begin{cases} \sum_{1=1}^{N}(\alpha_i - \alpha_i^*) = 0 \\ 0 \leq \alpha_i, \alpha_i^* \leq C \end{cases}$$

(3.47)

Sostituendo (3.43), che dà il valore di ***w***, in (3.38) si ottiene quindi la funzione che aiuta a determinare la previsione voluta per ogni **x** futura:

$$p(\boldsymbol{x}) = \sum_{1=1}^{N} (\alpha_i - \alpha_i^*) K(\boldsymbol{x}_i, \boldsymbol{x}) + b \tag{3.48}$$

Il parametro *b* può essere calcolato usufruendo della già citata condizione di KKT, ottenendo così:

$$b = y_i - \xi - \sum_{1=1}^{N} (\alpha_i - \alpha_i^*) K(\boldsymbol{x}_i, \boldsymbol{x}) \tag{3.49}$$

3.4.1.4 Vantaggi e Svantaggi delle SVM

Tra i principali vantaggi si evincono:

- *Efficacia in dimensioni spaziali elevate* (molti attributi);
- *Efficacia quando il numero di attributi supera il numero di istanze*;
- *Efficienza della memoria*: Una volta individuati i vettori di supporto, soltanto questi devono essere memorizzati;
- *Versatilità*: Come visto, possono essere utilizzati in diverse forme per soddisfare le specificità dei singoli problemi.

Svantaggi:

- *Interpretazione dei risultati non semplice*: Come molti (ma non tutti) gli approcci non parametrici l'interpretazione dei risultati potrebbe non essere facile. In questo senso, tecniche di visualizzazione grafica possono aiutare.
- *Poco adatto a dataset numerosi*: Per dataset con numerose istanze questo algoritmo non è consigliato. i tempi impiegati nel trovare i vettori di supporto può crescere più che *quadraticamente* al crescere delle istanze.

- *Non probabilistico*: Non esiste una interpretazione probabilistica della classificazione.

3.4.2 Decision Tree

L'Albero decisionale, o Decision Tree, è un algoritmo di machine learning che può essere utilizzato sia per la classificazione che per la regressione. La logica sottostante all'algoritmo è di facile comprensione. Supponendo di voler classificare a quale Nazione fra Cuba, Italia e Francia, appartiene una determinata bandiera, si può costruire il seguente grafico/albero (**Figura 3.5**):

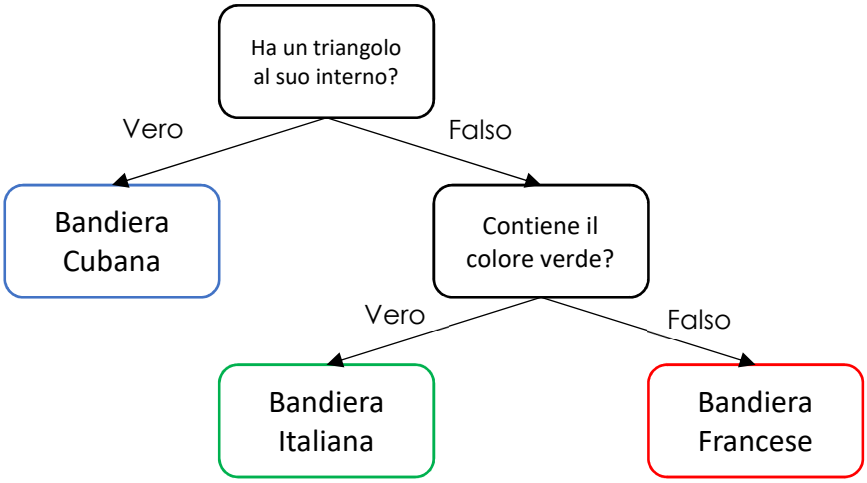


Figura 3.5: Esempio concettuale di albero decisionale

Si nota come ci siano due elementi costituenti l'albero: nodi e archi. I nodi sono i punti logici dell'albero nei quali si analizzano determinate condizioni prima di proseguire, attraverso un arco di collegamento, verso un altro nodo che specificherà ulteriori condizioni. I nodi finali, ovvero quelli che rappresentano la stima vera e propria, vengono chiamati anche *nodi foglie*.

In questo esempio, data una determinata bandiera della quale non si sa ancora la Nazione di appartenenza, ci si interroga sulla presenza o meno di un triangolo al suo interno nel primo nodo. Dato che l'unica bandiera fra le

tre Nazioni a scelta ad avere un triangolo è quella cubana, in caso questa condizione si verifichi la classificazione è automatica. In assenza di triangoli si prosegue verso il nodo successivo nel quale ci si interroga sulla presenza o meno del colore verde nella bandiera. In caso positivo, si deduce che si tratta della bandiera italiana; in caso negativo, di quella francese. Si nota inoltre come questo albero non sia l'unica soluzione: scambiando il primo nodo con il secondo ci si trova in una situazione simile con l'unica differenza che si accetta/scarta per prima la bandiera italiana invece che quella cubana.

Le condizioni vengono imposte ovviamente sugli attributi presenti nel dataset. In questo esempio quindi il dataset probabilmente ha al suo interno due attributi del tipo:

- '*Figure Geometriche*' = ['Rettangolo verticale', 'Triangolo', Rettangolo Orizzontale];
- '*Colori*' = ['Rosso', 'Bianco', 'Verde', 'Blue']

E le possibili classificazioni, quindi la variabile target:

- '*Bandiera'* = ['Bandiera Cubana', 'Bandiera Francese', 'Bandiera Italiana']

Le ramificazioni dei nodi possono essere binarie oppure multi-ramificate. Per capire la differenza, si supponga di avere un attributo descrivibile con una variabile $x \in \mathbb{N}, x \in [1,100]$. La ramificazione binaria in un determinato nodo dell'albero potrebbe essere rappresentata, ad esempio, come: $x_i > 60?$, perché 60 è stato individuato come il valore (parametro) all'interno dell'intervallo che meglio differenzia le categorie in gioco, partendo dall'attributo *x*; mentre la multi-ramificazione non fa altro che aumentare il numero di nodi sottostanti ai quali ci si può collegare in base ad un uguale numero di sottoinsieme della variabile x (se 0< x < 20, proseguire verso nodo A; se 19< x < 40, proseguire verso nodo B e così via). Quest'ultimo caso si può estendere fino ad associare un nodo sottostante per singolo valore del dominio dell'attributo: nella variabile di esempio x, si potrebbero avere quindi 100 nodi sottostanti: un nodo, un valore di x.

Per costruire un albero decisionale quindi si parte dal training set e si è interessati a definire sia le varie ramificazioni (o percorsi o sequenza di archi) e quindi la struttura stessa dell'albero, sia le condizioni all'interno di ogni nodo e quindi le variabili con i rispettivi parametri. Per fare questo si utilizza molto spesso l'algoritmo *Classification And Regression Tree* (CART).

3.4.2.1 Decision Tree per Classificazione

L'algoritmo CART suddivide il training set in due sottoinsiemi sulla base di un attributo *k* e di una soglia t_k. Per scegliere la prima coppia di attributo-soglia, che costituirà quindi il primo nodo dell'albero, si cerca la combinazione che produca i sottoinsiemi più "puri" minimizzando la seguente funzione:

$$J(k, t_k) = \frac{m_{sinistra}}{m} G_{sinistra} + \frac{m_{destra}}{m} G_{destra} \quad \forall k. t_k \qquad (3.50)$$

Con

- $G_{sinistra/destra}$ l'impurità dei sottoinsiemi di *sinistra/destra;*
- $m_{sinistra/destra}$ il numero di istanze presenti nel sottoinsieme di *sinistra/destra.*

L'impurità di un nodo *n* (o sottoinsieme) viene solitamente calcolata con l'*Indice di Gini*:

$$G_n = 1 - \sum_{c=1}^{C} {p_{n,c}}^2 \qquad (3.51)$$

Con

- $p_{n,c}$ la quantità di istanze di tipo (o categoria o classe) *c* presenti nel nodo *n* rispetto al totale. Può essere anche vista come la probabilità di trovare una istanza di tipo *c* nel nodo *n*;

Si definisce un nodo totalmente puro quando $G_n = 0$. Un'alternativa all'indice di Gini per misurare l'impurità di un nodo, ma che porta comunque

a simili soluzioni per la maggior parte dei casi, è la presa in prestito dalla Termodinamica della *Entropia H*, definita come:

$$H_n = -\sum_{c=1, p_{n,c}\neq 0}^{C} p_{n,c}\log(p_{n,c}) \tag{3.52}$$

L'indice di Gini rende leggermente più veloce i calcoli rispetto all'utilizzo dell'entropia H e per questo motivo viene spesso lasciato come criterio di default nelle varie funzioni che molte librerie di machine learning mettono a disposizione[1].

Una volta ottenuto i primi due sottoinsiemi si procede ad applicare la stessa logica singolarmente su questi per determinare via a via le varie ramificazioni dell'albero. La lunghezza dei vari livelli dell'albero può essere determinata a priori per evitare di caricare onerosamente la macchina di calcolo oppure può essere fermata quando non si riesce a ridurre ulteriormente l'impurità di un significativo valore.

3.4.2.2 Decision Tree per Regressione

Le logiche utilizzate nella regressione non cambiano di molto rispetto a quelle utilizzate per la classificazione. Di fatto, invece di prevedere una determinata classe nei nodi foglia, si determineranno dei valori numerici. Nello specifico, il valore previsto sarà la media di tutti i valori osservati per le istanze coinvolte in quel determinato nodo.

Il CART resta lo stesso di prima con la differenza che invece di calcolare e minimizzare l'impurità del nodo si cerca di minimizzare il *Mean Squared Error* (MSE o Errore medio quadratico):

$$J(k, t_k) = \frac{m_{sinistra}}{m} MSE_{sinistra} + \frac{m_{destra}}{m} MSE_{destra} \tag{3.53}$$

[1]Scikit-learn ad esempio utilizza di default Gini nella funzione: `DecisionTreeClassifier`

$$MSE_n = \sum_{i \in n} (\hat{y}_n - y_i)^2 \tag{3.54}$$

$$\hat{y}_n = \frac{1}{m_n} \sum_{i \in n} y_i \tag{3.55}$$

dove

- $\hat{y}_n$ è la stima del valore di quel determinato nodo, calcolata come media deli singoli valori target y_i delle istanze del nodo *n* di riferimento.
- m_n sono tutte le istanze presente nel nodo *n*.

Il CART è un algoritmo detto "greedy" in quanto non garantisce la soluzione di ottimo. Trovare l'albero ottimale in senso assoluto sarebbe l'equivalente di risolvere un problema di complessità computazionale di classe *NP-Completo*[1] ed è per tanto quasi un obbligo indagare ed utilizzare soluzioni come quella offerta dal CART.

3.4.2.3 Vantaggi e Svantaggi dei Decision Tree

Gli alberi decisionali sono molto comodi in quanto:

- Non richiedono una particolare preparazione dei dati: possono trattare variabili categoriche non trasformate; variabili non standardizzate o normalizzate; valori missing;
- Si ha una facile comprensione del "perché" è stata presa una determinata decisione visualizzando il percorso di nodi, e quindi di decisioni prese, che hanno portato alla stima fornita (pur essendo un modello non parametrico a priori);
- La data selection avviene in modo intrinseco nell'algoritmo.

[1] Nella teoria di complessità computazionale si definisce una classe P come l'insieme dei problemi decisionali che possono essere **risolti** da una macchina di Turing deterministica in un tempo, polinomiale rispetto alla dimensione dei dati in input, "utile"; un problema decisionale si dice di classe NP se le sue soluzioni si possono **verificare** in un tempo polinomiale utile (o equivalentemente se le soluzioni possono essere trovate in un tempo polinomiale utile con una macchina di Turing non-deterministica). La classe NP-Completo fa riferimento a quei problemi decisionali che sicuramente non appartengono a P se P ≠ NP. Sono i problemi NP più difficili e sono anche i meno probabile di appartenere a P. Se P ≠ NP allora non sarà mai trovato un algoritmo polinomiale per un problema NP - Completo, tranne forse con un computer quantistico (macchina di Turing non-deterministica) [22]

Tuttavia, presentano anche alcuni svantaggi:

- Le suddivisioni dei dati sono sempre ortogonali agli assi e questo potrebbe creare alcuni problemi laddove, per motivazioni particolari (durante la fase di preparazione del training set, ad esempio), i dati venissero ruotati modificando l'accuratezza del modello;
- Generalmente sono molto sensibili anche a piccole variazioni dei dati del training set.

Questi rischi possono tuttavia essere mitigati utilizzando un Random Forest.

3.4.3 Random Forest e l'Ensemble Learning

Come indica creativamente il nome dell'algoritmo, il *Random Forest* (Foresta Casuale) è un algoritmo di *Ensemble learning*[1], composto dunque non da un unico albero decisionale ma da tanti. La stima, in caso si tratti di una classificazione, sarà la classe più "votata" fra le stime dei singoli alberi decisionali (quindi la moda, **Figura 3.6**). Per problemi di regressione si tratterà della media dei singoli valori previsti dagli alberi. Pur essendo un algoritmo molto semplice si è rivelato essere uno degli strumenti più potenti e versatili utilizzato ad oggi dai data scientist.

[1] Gli algoritmi di Ensemble sono algoritmi composti che prendono in considerazione le singole stime di ogni componente (solitamente algoritmi detti *deboli*) per creare una stima finale pesata.

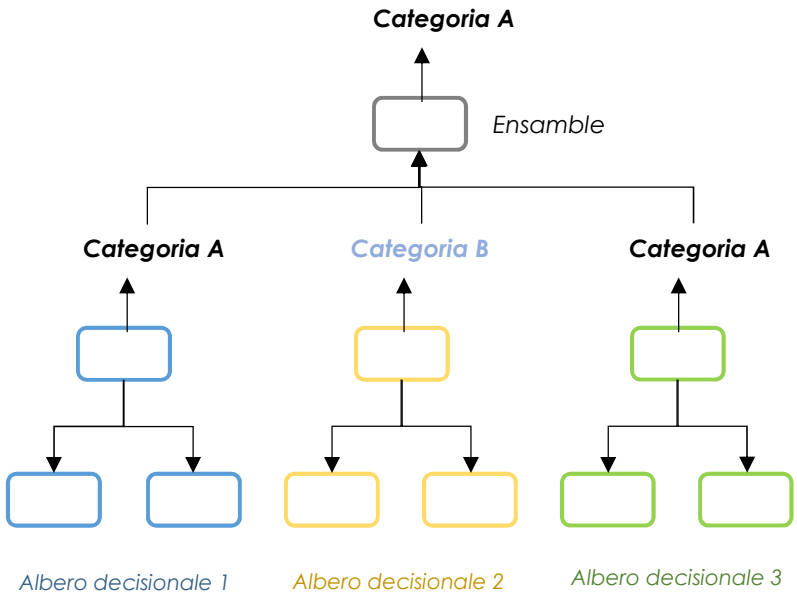


Figura 3.6: Schema di struttura del Random Forest.

I metodi di *Ensemble Learning* hanno dimostrato più e più volte di funzionare meglio rispetto ai singoli algoritmi. Addirittura, ci sono prove che un algoritmo di *Ensemble* composto da molti stimatori "deboli" forniscono una stima più accurata del più potente degli stimatori "singoli". L'intuizione che spiega il perché di questo fenomeno la si può delineare anche con il comportamento umano. Di fronte ad un problema spesso è utile prendere in considerazione l'opinione di più esperti contemporaneamente, invece che quella del singolo, per via dei vari punti di vista e le diverse esperienze e competenze. Questo si riflette in molte delle nostre organizzazioni odierne: parlamenti, consigli di amministrazione, compagni dell'università organizzando una cena di classe, ecc. La Democrazia stessa, sistema di governo più diffuso oggi nel mondo[1], è guidata da questo concetto e la sua separazione dei poteri (fra legislativo, giudiziario ed esecutivo) teorizzata dal francese Montesquieu[2], ne riflette l'anima. Così come una persona è diversa dalle altre e riesce a notare aspetti che alle altre possono sfuggire,

[1] [23]
[2] [24] C.L. DE SECONDAT DE MONTESQUIEU, *Lo spirito delle leggi*, trad it. a cura di B. Boffito Serra, Milano, 1967, p. 207 e ss.

un algoritmo costruito secondo criteri diversi da altri algoritmi può raccogliere aspetti che sfuggono ai secondi.

Le forme più famose ed utilizzate di Ensamble learning sono:

- *Bagging*[1]: Si utilizza la stessa tipologia di algoritmo ma su diversi subset del training set creati con *rimpiazzi* (una istanza può finire più volte in più campioni). Solitamente il Random Forest utilizza questa logica riducendo fortemente la varianza nella previsione rispetto ad un singolo albero decisionale;
- *Pasting*: Esattamente come il Bagging ma senza il rimpiazzo;
- *Stacking*: Invece di utilizzare una predeterminata funzione, la media o la moda, come criterio decisionale per aggregare le singole previsioni degli stimatori, lo *Stacking* utilizza a sua volta un algoritmo di apprendimento.
- *Boosting*: Si tratta di allenare gli algoritmi non parallelamente, come fatto per il Bagging e il Pasting, ma in modo sequenziale dando sempre più peso agli errori commessi dagli algoritmi "precedenti". In questo modo si spera che l'algoritmo successivo corregga quello precedente andando a migliorare l'accuratezza complessiva della previsione. Infine, si sceglie la stima finale considerando le stime pesate dei singoli algoritmi secondo un determinato criterio. Esempi di questi algoritmi sono:
    - AdaBoost (*Adaptive Boost*)[2]
    - Gradient Boosting [3]

3.4.3.1 AdaBoost

Volendo capire l'idea alla base e per presentare l'AdaBoost senza entrare nel particolare di ogni sua variante, si elencano i seguenti passi logici:

0- Considerando un problema di classificazione binario con:
    - *N* $x_1, x_2 \dots x_n$ vettori (istanze) in input;
    - *M* stimatori in sequenza;
    - $t_1, t_2 \dots t_n, t_n \in \{-1,1\}$ le label del training set;
    - w_ni pesi assegnati ad ogni istanza;
    - $y(x) \in \{-1,1\}$ la funzione segno di classificazione.

[1] Breiman, L. Bagging Predictors. *Machine Learning* 24, 123–140 (1996).
[2] [25] Yoav Freund and Robert E. Schapire. "*A decision-theoretic generalization of on-line learning and an application to boosting"*. Journal of Computer and System Sciences, 55(1):119–139, August 1997;
[3] [26] Jerome H. Friedman. Greedy Function Approximation: A Gradient Boosting Machine, Aprile 2001

1- Si inizializza assegnando il peso di default $w_n = \frac{1}{N} \quad \forall n$

2- Per *m*= 1,…,M:

2.1- Si allena il classificatore $y_m(x)$ sul training set minimizzando la funzione di errore:

$$J_m = \sum_{n=1}^{N} w_n^m I(y_m(x_n) \neq t_n) \tag{3.56}$$

$$con\ I(y_m(x_n) \neq t_n) \begin{cases} 1 & se\ y_m(x_n) \neq t_n \\ 0 & se\ y_m(x_n) = t_n \end{cases}$$

2.2- Si calcolano le quantità:

$$\epsilon_m = \frac{\sum_{n=1}^{N} w_n^m I(y_m(x_n) \neq t_n)}{\sum_{n=1}^{N} w_n^m} \tag{3.57}$$

$$\alpha_m = \ln\left(\frac{1-\epsilon_m}{\epsilon_m}\right) \tag{3.58}$$

Con ϵ_ml'errore nella misurazione dello stimatore *m*, α_m il peso che si darà alla stima dello stimatore *m*.

2.3- Si aggiornano I pesi delle istanze:

$$w_n^{m+1} = w_n^m exp[\alpha_m I(y_m(x_n) \neq t_n)] \tag{3.59}$$

3- Si esegue la stima finale Y_M:

$$Y_M(x) = sign\left(\sum_{m=1}^{M} \alpha_m y_m(x)\right) \tag{3.60}$$

Secondo questa logica quindi, allenato un primo stimatore, si evince quelle che sono le istanze per le quali la sua previsione non è stata corretta. Queste istanze vedranno un aumento del loro peso w_n (***boosting*** appunto) e saranno così considerate maggiormente dal successivo stimatore grazie allo step di aggiornamento 2.3. Infine, nella stima finale, si darà una maggior considerazione ai classificatori con un errore di previsione ϵ_m inferiore.

3.4.3.2 Gradient Boosting

Questo algoritmo si basa più sulla "discesa del gradiente", ovvero data una determinata funzione di costo (solitamente si utilizza l'errore quadratico medio per i problemi di regressione e la perdita logaritmica per problemi di classificazione), si cerca di minimizzare gli errori commessi allenando in sequenza stimatori deboli (solitamente degli alberi decisionali):

0- Considerando:
   - $x = x_1, \dots, x_n$il vettore degli input (istanze);
   - $m = 1, \dots, M$ stimatori in sequenza;
   - $F(x, P) = \sum_{m=0}^{M} \beta_m h(x, \alpha_m)$ la funzione di previsione con parametri $P = \{\beta_m, \alpha_m\}$ che prende in input il vettore x per stimare la variabile target $y = y_1, \dots, y_n$;
   - $h(x, \alpha_m)$ una funzione di x avente come parametri $\alpha = \alpha_1, \dots, \alpha_M$
   - $L(y_i, \rho)$ la funzione di costo con ρ il "tasso di apprendimento" , con il quale si correggono gli errori di stima;

1- Si inizializza con $F_0(x) = arg\ min_\rho \sum_{i-1}^{N} L(y_i, \rho)$

2- Per m=1,…,M:

2.1- Si calcola il gradiente come:

$$\tilde{y}_i = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right] \tag{3.61}$$

2.2- Si allena il modello:

$$\alpha_m = \arg min_{\alpha,\beta} \sum_{i=1}^{N} [\tilde{y} - \beta h(x_i, \alpha_m)]^2 \tag{3.62}$$

2.3- Se sceglie il grado di apprendimento:

$$\rho_m = \arg min_\rho \sum_{i-1}^{N} L(y_i, F_{m-1}(x_i) + \rho h(x_i, \alpha)) \tag{3.63}$$

2.4- Si aggiornano le stime rispetto al modello precedente:

$$F_m(x) = F_{m-1}(x) + \rho_m h(x, \alpha_m) \tag{3.64}$$

3- Fine dell'algoritmo.

3.4.3.3 Vantaggi e Svantaggi del Random Forest

Tra i vantaggi si hanno:

- Comporta *tutti i vantaggi del Decision Tree* ma, rispetto a questo, riduce inoltre il rischio di overfitting, riduce la varianza e spesso viene migliorata di molto l'accuratezza;
- Versatilità, infatti è molto performante sia su problemi di classificazione che di regressione;

Svantaggi:

- Richiede maggiori tempi di calcolo rispetto al singolo Decision Tree;
- La complessità dell'interpretazione dei risultati aumenta notevolmente dal momento che si considerano tanti singoli Decision Tree e vengono combinati i singoli risultati.

3.4.4 Rete Neurale

Per capire il funzionamento delle *Reti Neurali Artificiali* (chiamate comunemente *Reti Neurali*) si può iniziare spiegando il modello reale dal quale venne presa l'idea: le reti neurali biologiche.

Il nostro cervello è composto da circa 86 miliardi [1]di specifiche cellule, comunemente conosciute come neuroni. Queste cellule sono composte da un *corpo cellulare,* contenente il nucleo, ma anche da tante estensioni dette *dendriti* ed un'estensione, molto più grande rispetto ai dendriti, detta *assone* (**Figura 3.7**). Al suo estremo, l'assone si divide in tante piccole parti che prendono il nome di *sinapsi*. Le sinapsi di un neurone si congiungono con i dendriti di un altro neurone per comunicare attraverso impulsi elettrici.

[1] [27] Suzana Herculano-Houzel, *"The Human Brain in Numbers: A Linearly Scaled-up Primate Brain",* Published online Front Hum Neurosci, 2009

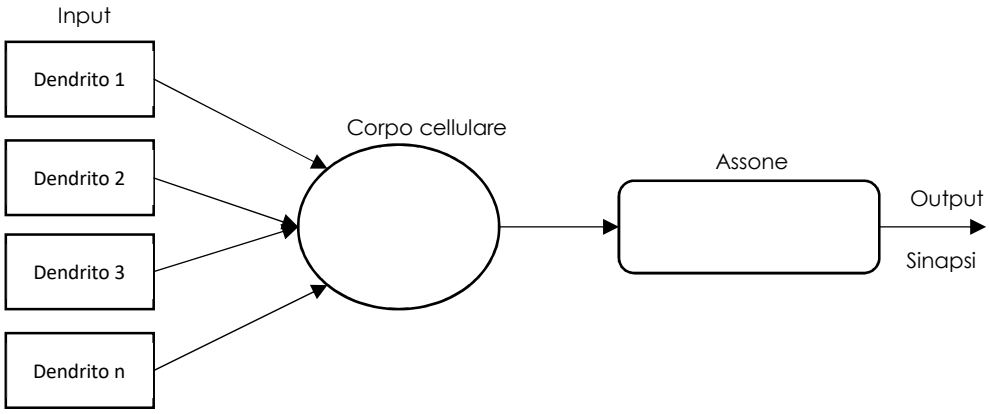


Figura 3.7: Schema concettuale di un Neurone

Come funzionano queste comunicazioni? Per via della presenza di ioni sia fuori che dentro la cellula, si creano differenze di potenziale. Quando il neurone presinaptico si attiva rilascia nel neurone postsinaptico degli ioni di sodio e potassio. Se la concentrazione di ioni nel neurone postsinaptico è sufficientemente elevata, tale da superare una determinata soglia, il neurone si attiva e rilascia a sua volta una scarica elettrica verso i neuroni ai quali è collegato tramite l'assone.

Lo psicologo canadese Donald Hebb spiegò [1]nel 1949 come un neurone A, avente l'assone vicino ad un neurone B, che partecipa frequentemente nell'attivazione del neurone B, creerà una situazione tale per cui, con il tempo, l'attivazione del neurone B da parte del neurone A sarà sempre più efficiente. Questo implica che le sinapsi fra i neuroni cambiano nel tempo in base al numero di volte che i due interagiscono, rafforzandosi ed efficientando il processo. In questo modo si capisce la connessione che esiste fra i numerosi neuroni presenti nel nostro cervello ed anche il meccanismo di apprendimento di questi.

[1] [28] Donald O. Hebb, *The organization of behavior; a neuropsychological theory*. Wiley, New York, 1949.

Per fare un esempio semplicistico, se vedessimo un nostro parente, probabilmente nel nostro cervello si attiverebbe un neurone indicando che si tratta di una persona, questo neurone farebbe attivare un altro, fra i tanti, che ci ricorda come quella persona sia un nostro parente, successivamente un altro, che ci ricorda come quel parente è nostra madre, poi uno che ci ricorda come i suoi capelli siano mori e così via. L'attivazione di questi neuroni è quasi istantanea perché pensiamo, vediamo o comunque abbiamo visto spesso nostra madre. Agli studenti di ingegneria gestionale non succede altrettanto con la *dimostrazione* del teorema di Binet, fatto il primo anno universitario in Geometria - Algebra Lineare e mai più ripresa. È facile aspettarsi che le sinapsi costruite durante il periodo di studio che ci ricordavano la dimostrazione passo dopo passo siano andate perdute del tutto, perché mai più sollecitate.

Come trasformare questo in un modello matematico che simuli sia il comportamento neuronale che il suo apprendimento? Il primo passo è stato fatto, nel 1943, da Warren McCulloch e Walter Pitts[1], che hanno modellato la connessione fra neuroni utilizzando degli schemi molto somiglianti alle porte logiche utilizzate nei programmatori: OR, AND e NOT (**Figura 3.8**). Questo modello, in termini binari, spiega in modo chiaro come le connessioni fra i neuroni possono fare calcoli e ragionamenti logici esattamente come le porte informatiche spiegano i ragionamenti di un computer.

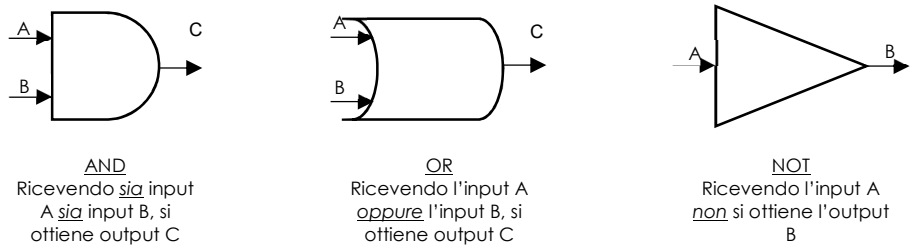


Figura 3.8: Porte informatiche And, Or e Not.

[1] [29] W. McCulloch, W. Pitts, "A Logical Calculus of Ideas Immanent in Nervous Activity", 1943, Bulletin of Mathematical Biophysics 5:115–133.

Mancava tuttavia la componente di apprendimento, che venne spiegata nel 1957 dal modello di *percettrone* di Frank Rosenblatt[1]. In questo modello gli input, non più binari, hanno dei pesi e l'output, di tipo binario, rappresenta con 1 e 0 rispettivamente gli stati "neurone attivato" e "neurone non attivato". L'attivazione del neurone avviene se la somma pesata degli input supera una determinata soglia. Per schematizzare si rimodula la **Figura 3.8** come segue:

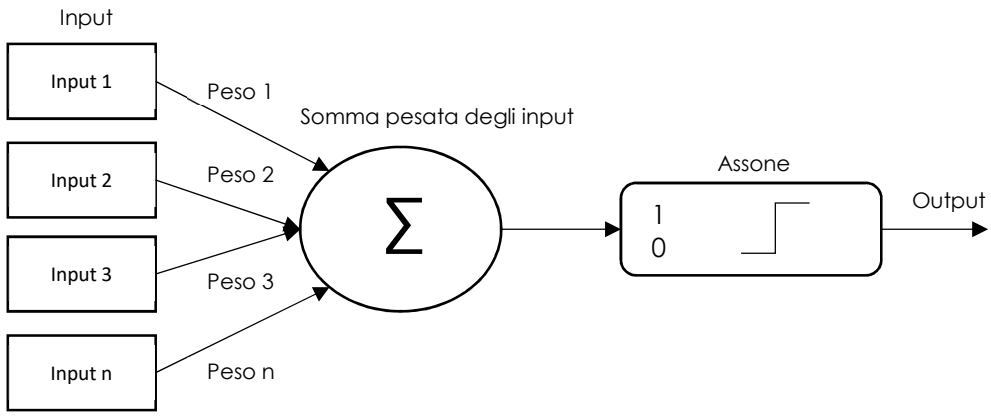


Figura 3.9: Schema del Percettrone.

L'apprendimento è determinato dall'aggiornamento dei pesi degli input con il susseguirsi delle istanze. Il percettrone, utilizzato per problemi di classificazione binaria, tuttavia, ha dei limiti non indifferenti. Come dimostrato nel 1969 da Marvin Minsky e Seymour Papert nel loro libro *Perceptrons*[2], l'algoritmo non è in grado di apprendere funzioni elementali come l'"*OR-esclusivo*" (o XOR). Questa funzione è vera se uno dei due input è vero, ed è falsa se i due input sono uguali. Questo problema può essere ovviato se invece di utilizzare un unico strato neuronale, se ne

[1] [30] Frank Rosenblatt, "The Perceptron: A Probabilistic Model For Information Storage and Organization in the Brain", Psychological Review vol. 65, 1958
[2] [31] Marvin Minsky e Seymour Papert, "Perceptrons: An Introduction to Computational Geometry",The M.I.T press,1969

utilizzano di più. Il problema del modello a più neuroni, conosciuto come *Multi-layer perceptron* (o percettrone multistrato), era che non si sapeva come approcciare il training dell'algoritmo e quindi aggiornare i vari pesi fra i vari neuroni connessi nella rete.

Questo fu possibile grazie alla sostituzione della funzione a step in output (corrispondente all'assone nella analogia visualizzata precedentemente) con una funzione derivabile. Nello specifico si utilizzarono curve *"ad S"*, nominate in questo modo per la caratteristica forma. La differenza fra la funzione step e la funzione ad S si esplicita fisicamente nel trattare un fenomeno non più come "istantaneo" ma come "di transizione". Quando la temperatura scende e raggiunge lo 0°C, l'acqua non diventa *istantaneamente* ghiaccio, segue piuttosto una transizione di fase nella quale, prima in modo accelerato, la maggior parte delle molecole di acqua costruiscono i legami caratteristici del ghiaccio e poi, in modo decelerato, le ultime molecole d'acqua rimaste si uniscono alla massa di ghiaccio. Per un breve istante la materia si trova sia in stato liquido che in stato solido. L'introduzione della curva ad S nel percettrone cambiava leggermente quindi il fenomeno fisico rappresentato. Il neurone iniziava ad emettere elettricità in modo sempre più crescente per poi completare l'emissione in modo decrescente all'aumentare del peso degli input, ma, e soprattutto, rendeva derivabile la funzione in output.

Quest'ultimo aspetto ha avuto dei risvolti pratici notevoli in quanto permise di introdurre, nel 1986 da parte di David E. Rumelhart, G. Hinton e R. J. William, il metodo di *retropropagazione dell'errore*, che si trasformò in elemento chiave per l'algoritmo di training delle reti multi-strato. In una prima fase si assegnano ai vari pesi dei valori più o meno casuali in modo che venga fornita in output alla rete una prima previsione. Questa previsione viene confrontata con il valore reale osservato e si calcola in questo modo l'errore commesso. Il calcolo dell'errore viene propagato lungo tutta la rete neurale partendo dagli ultimi strati vicini all'output e finendo nei primi strati. I neuroni che hanno avuto una migliore previsione vedranno aumentati i loro pesi, mentre i neuroni che hanno avuto una cattiva previsione verranno

penalizzati. Questo processo avviene per ogni nuova istanza finché non finisce il training.

3.4.4.1 Modello Percettrone Multistrato

Provando ora a formalizzare le idee espresse precedentemente, si illustra il modello di una rete con uno strato neuronale fra l'input e l'output, ovvero un modello di Percettrone Multistrato:

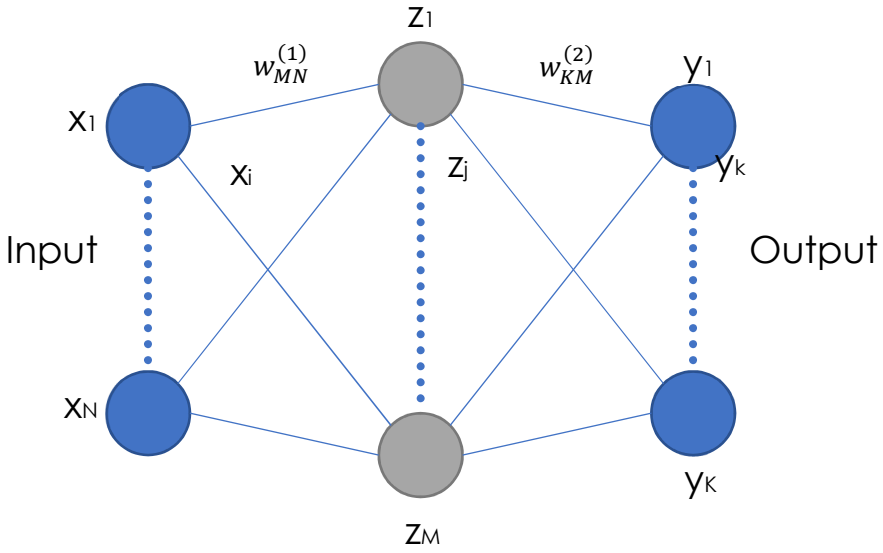


Figura 3.10: Schema di un Percettrone con strato nascosto.

Si può definire la funzione generica di previsione come:

$$y(\boldsymbol{x}, \boldsymbol{w}) \tag{3.65}$$

Con

- $\boldsymbol{x}$ il vettore degli attributi *i* in input e **w** il vettore dei pesi nelle connessioni fra neuroni;

Si costruisce ora una combinazione lineare degli attributi in input per ogni neurone *j* del primo layer (o strato) segnalato con l'apice (1):

$$a_j = \sum_{i=1}^{N} w_{ji}^{(1)} x_i + w_{j0}^{(1)} \quad \forall j \tag{3.66}$$

I parametri $w_{j0}^{(1)}$ vengono detti *"bias"*, non in senso di errore statistico, si tratta delle costanti previste nei basilari modelli di combinazione lineare. I a_j vengono chiamati "attivazioni" (in analogia con la concentrazione di ioni nei neuroni) e sono l'input per la trasformazione:

$$z_j = h(a_j) \quad \forall j \tag{3.67}$$

La funzione di attivazione h(.) dello strato intermedio, come già detto, assume la forma di una curva ad S, indipendentemente da quella che sarà la funzione d'output y. Particolarmente utilizzate sono:

- Funzione sigmoidea logistica:

$$h(a_j) = \frac{1}{1 + e^{-a_j}} \tag{3.68}$$

- Funzione tangente iperbolica:

$$h(a_j) = \frac{e^{a_j} - e^{-a_j}}{e^{-a_j} + e^{-a_j}} \tag{3.69}$$

- Funzione Relu[1]:

$$h(a_j) = \max(0, a_j) \tag{3.70}$$

Ottenuti gli output del primo layer z_j, la logica nel secondo layer si ripete, questi diventano gli input per il calcolo delle attivazioni nel secondo layer (2):

$$a_k = \sum_{j=1}^{M} w_{kj}^{(2)} z_j + w_{k0}^{(2)} \tag{3.71}$$

[1] La funzione Relu non ha la classica forma ad S ma viene molto spesso utilizzata in quanto la derivata è zero per i numeri negativi e 1 per i positivi, cosa che facilita di molto i calcoli, specialmente nelle reti complesse. Non risulta derivabile all'origine ma viene spesso definito zero come parametro di default in questo caso.

Con *k=1,…,K* il numero degli output. A sua volta viene fatta la trasformazione come fatto per lo strato (1):

$$y_k = \sigma(a_k) \tag{3.72}$$

Come detto precedentemente, trattandosi di una classificazione si possono utilizzare le curve ad S. Se invece si trattasse di una regressione si utilizza la funzione identità $y_k = a_k$. La funzione di previsione, sostituendo opportunamente quanto visto precedentemente, diventa:

$$y(\boldsymbol{x}, \boldsymbol{w}) = \sigma\left(\sum_{j=1}^{M} w_{kj}^{(2)} h\left(\sum_{i=1}^{N} w_{ji}^{(1)} x_i + w_{j0}^{(1)}\right) + w_{k0}^{(2)}\right) \tag{3.73}$$

Per non esplicitare i *bias* solitamente in letteratura si introduce un ulteriore input $x_0 = 1$ da associare al peso $w_{j0}^{(1)}$, e lo stesso viene fatto per gli strati intermedi. In questo modo si ottiene la (3.73) come:

$$y(\boldsymbol{x}, \boldsymbol{w}) = \sigma\left(\sum_{j=0}^{M} w_{kj}^{(2)} h\left(\sum_{i=0}^{N} w_{ji}^{(1)} x_i\right)\right) \tag{3.74}$$

Questo processo descritto viene detto di *forward propagation* in quanto i neuroni non hanno dei collegamenti tali da creare dei "cerchi", l'informazione viene assorbita nei primi strati e propagata verso quelli successivi. La generalizzazione a modelli più elaborati, come ad esempio contenenti più strati intermedi fra input ed output, è facilmente fattibile e deducibile da quanto esposto.

Come detto precedentemente, il training delle reti neurali avviene secondo un sofisticato meccanismo di aggiornamento dei pesi con *retropropagazione degli errori*. Una volta concluso il processo di *forward propagation*, si calcola l'errore commesso secondo una funzione di errore

E e si utilizza la *discesa del gradiente* per aggiornare i vari pesi coinvolti nella rete neurale:

$$w_{n+1} = w_n - \eta \nabla E(w_n) \tag{3.75}$$

Con

- *n* una determinata istanza del training set (in questa formula risulta l'unico indice in quanto vale per qualsiasi peso in qualsiasi strato della rete);
- $\nabla E(w_n)$ il gradiente della funzione errore *E* rispetto ai pesi *w*;
- η è il *learning rate* (o *tasso di apprendimento*) positivo e deciso a piacere.

La funzione di errore solitamente assume la forma quadratica:

$$E = \frac{1}{2}\sum_k (y_k - t_k)^2$$

Con t_k la label, o valore osservato. Il gradiente dell'errore E_n può anche essere scritto come:

$$\frac{\partial E_n}{\partial w_{jk}} = \frac{\partial E_n}{\partial a_k}\frac{\partial a_k}{\partial w_{jk}} \tag{3.76}$$

Supponendo di affrontare una regressione, per cui $y_k = a_k = \sum_{j=0}^{M} w_{jk} z_j$ si introducono le seguenti uguaglianze:

$$\delta_k = \frac{\partial E_n}{\partial a_k} = y_k - t_k \tag{3.77}$$

$$\frac{\partial a_k}{\partial w_{jk}} = z_j \tag{3.78}$$

E sostituendo in (3.76), si ottiene:

$$\frac{\partial E_n}{\partial w_{jk}} = \delta_k z_j = (y_k - t_k) z_j \tag{3.79}$$

Si ricava quindi:

$$w_{n+1,jk} = w_{n,jk} - \eta \nabla E(w_{n,jk}) \tag{3.80}$$

$$w_{n+1,jk} = w_{n,jk} - \eta \delta_k z_j \tag{3.81}$$

Allo stesso modo si prosegue con la propagazione degli errori aggiornando i pesi del primo strato w_{ij}.

$$\frac{\partial E_n}{\partial w_{ij}} = \frac{\partial E_n}{\partial a_j}\frac{\partial a_j}{\partial w_{jk}} = \delta_j \frac{\partial \sum_{i=1}^{N} w_{ij} x_i}{\partial w_{jk}} = \delta_j x_i \tag{3.82}$$

$$\delta_j = \frac{\partial E_n}{\partial a_j} = \sum_k \frac{\partial E_n}{\partial a_k}\frac{\partial a_k}{\partial a_j} = \sum_k \delta_k \frac{\partial \sum_j w_{jk} h(a_j)}{\partial a_j} = h'(a_j)\sum_k \delta_k w_{jk} \tag{3.83}$$

E combinando le due espressioni (3.82) e (3.83):

$$\frac{\partial E_n}{\partial w_{ij}} = x_i h'(a_j)\sum_k \delta_k w_{jk} \tag{3.84}$$

Si ricavano i pesi come:

$$w_{n+1,ij} = w_{n,ij} - \eta x_i h'(a_j)\sum_k \delta_k w_{jk} \tag{3.85}$$

Supponendo che la funzione di attivazione h(.) sia la tangente iperbolica (3.69), per fare un esempio, la sua derivata sarebbe:

$$h'(a_j) = 1 - h(a_j)^2 \tag{3.86}$$

E quindi, sostituendo in (3.85):

$$w_{n+1,ij} = w_{n,ij} - \eta x_i (1 - z_j^2)\sum_k \delta_k w_{jk} \tag{3.87}$$

Questo processo viene iterato per tutte le *n* istanze presenti nel training set.

Il modello presentato è quello relativamente più semplice che aiuta a generalizzare al meglio. Intuitivamente ci si accorge che potenzialmente le reti possono differire infinitamente. Una rete può variare nel numero di layer interni fra input ed output ma può anche variare il numero di neuroni presenti per ogni layer. Pur essendo un algoritmo con così tante opzioni esistono alcune regole empiriche che aiutano nella costruzione di una rete.

Il giusto numero di layer e il giusto numero di neuroni per ognuno di essi non è di facile determinazione. Tuttavia, si sa che il semplice percettrone multistrato con un unico layer intermedio è capace di risolvere le più complesse funzioni se contenente un sufficiente numero di neuroni. Si sa anche che l'impiego di più layer aiuta a ridurre esponenzialmente l'utilizzo di neuroni a parità di funzione rispetto al singolo percettone. Per questo motivo il *Deep Learnig,* ovvero l'utilizzo di reti a più strati, è esploso negli ultimi anni.

Definire il numero di neuroni per layer è tutt'oggi più una decisione presa sulla base di risultati empirici e/o continue iterazioni che su regole definite a priori. Sicuramente è consigliabile avere un maggior numero di neuroni nei layer intermedi rispetto agli input o gli output. Se così non fosse, si saprebbe a priori che una parte delle informazioni in input viene persa per via della riduzione dimensionale negli strati intermedi.

3.4.4.2 Vantaggi e svantaggi delle Reti Neurali

Tra i vantaggi si possono elencare:

- Capacità di affrontare problemi estremamente complessi individuando pattern anche non lineari (classificazione di video, suoni, immagini, ecc);
- Tolleranza agli errori e al rumore;
- Indipendenza di assunzioni a priori;
- Versatilità nell'affrontare diversi problemi di classificazione e regressione.

Svantaggi:

- Occorre una specifica preparazione dei dati (Normalizzazione, Standardizzazione, assenza di valori missing, ecc);
- Tempi di training mediamente più lunghi di altri algoritmi;
- Estrema difficoltà nell'interpretazione dei risultati e nella rialita delle logiche di aggiornamento dei pesi, quindi poca trasparenza nel processamento delle istanze;
- Calcoli onerosi per i computer.

3.5 Fine Tuning del modello scelto

Come visto nei paragrafi precedenti, in media, i modelli di machine learning coinvolgono un gran numero di parametri e ci sono diverse versione degli stessi: nell'albero decisionale bisogna decidere quanto profondi devono essere, quali e quanti attributi utilizzare per nodo, con quale/i parametro di separazione, numero di nodi figli da collegare con il nodo padre, ecc; per il random forest bisogna decidere anche il numero di alberi da utilizzare; per le reti neurali il numero di layer, il numero di neuroni per layer, le funzioni di attivazione, il tasso di apprendimento, ecc.

Quindi una volta provati vari modelli e confrontati fra di loro capiamo quale di questi meglio si adatta alle necessità (dunque quale tribù di Domingos risulta vincitrice), ma si è sicuri che il modello scelto si manifesti nella sua miglior versione? L'albero decisionale con 5 nodi ha previsto una LGD migliore del SVM, ma è possibile trovare un altro albero decisionale con più o meno nodi in grado di battere suo fratello?

Come molte domande in ambito machine learning, la risposta non è data a sapere a priori, per cui la si ricava *testando*. Le librerie di machine learning in python offrono delle funzioni che aiutano in questo senso. Si parla di *Fine tuning* del modello quando si cerca la combinazione ottimale di parametri che permette di avere la miglior performance. A disposizione si ha:

- *GridSearch*: La griglia di ricerca lavora in modo estremamente intuitivo. Viene deciso a priori quali sono i parametri sui quali si vogliono testare più varianti, si specificano i valori che questi parametri devono assumere e la GridSearch si preoccupa di testare tutte le combinazioni esplicitate. Testare le combinazioni significa allenare il modello utilizzando il primo set di parametri sul training set, stimando la variabile target sul test set, calcolando l'errore commesso e confrontandolo con il risultato dello stesso modello con altri parametri. Dopo che tutte le combinazioni sono state provate viene evidenziata la combinazione più performante;

- *Randomized Search*: Quando lo spazio di scelta dei parametri è così vasto da rendere non facile la scelta a priori di quali si vogliono testare, una buona opzione è utilizzare la Randomized Search che funziona come la GridSearch con la differenza che sceglie le combinazioni dei parametri in modo *random*. In questo caso quindi non si fissano le combinazioni da provare ma solo il numero che si vuole siano provate, sarà poi l'algoritmo a scegliere, fra le combinazioni possibili che sono state specificate, quelle da testare;

- *Cross Validation*: I due metodi precedenti fanno largo utilizzo di questa tecnica per ovviare i problemi di overfitting che possono nascere durante la fase di training dei vari algoritmi. Per illustrare il funzionamento della Cross Validation, supponendo di suddividere il training set in tre diversi sottoinsiemi, si faccia riferimento alla **Figura 3.11**:

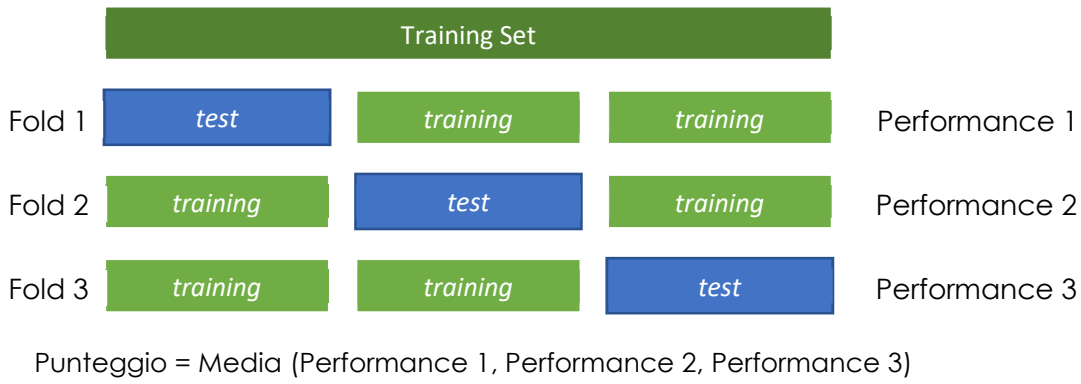


Figura 3.11: Schema Cross-Validation con 3 Fold

Il training set viene diviso in tre *Fold*, sottoinsiemi. Nel primo caso si sceglie uno dei tre sottoinsiemi come test set e si utilizzano gli altri due sottoinsiemi per fare training dell'algoritmo, ottenendo dunque la Performance 1. Lo stesso viene fatto nelle altre due casistiche con la differenza che si utilizzano diverse combinazioni di training e test set. Il punteggio finale dato alla combinazione di parametri proviene dalla media delle singole performance. In questo caso ci si assicura che ottime performance di combinazioni di parametri non derivino da *overfitting* sui dati.

- *Ensamble*: Come già detto precedentemente, "l'unione fa la forza" vale anche per gli algoritmi di machine learning. Si potrebbero utilizzare varie versioni dell'algoritmo originale e fornire la stima finale come media delle singole stime.

Capitolo 4: Calcolo della LGD con algoritmi di Machine Learning. Caso di studio Gruppo Intesa Sanpaolo.

Un calcolo più accurato delle variabili di rischio quali la LGD o la PD comportano degli impatti non indifferenti nell'operatività e nella strategia della Banca. Queste variabili di rischio, come già accennato, servono a calcolare il capitale di vigilanza, ovvero l'ammontare che l'intermediario non può utilizzare ma deve mantenere nei propri depositi per far fronte ad uno scenario di crisi. Se si riuscisse a provare una metodologia di calcolo che meglio stimi queste variabili si ridurrebbero sia i casi in cui l'accantonamento è stato troppo, sia i casi in cui è stato troppo poco.

In particolare, avere una LGD stimata *inferiore* significa dover accantonare meno capitale per far fronte ad eventuali crisi. Questo capitale non accantonato potrebbe ovviamente essere utilizzato per ulteriori investimenti. Alcuni potrebbero trovare preferibile un calcolo poco preciso che obblighi l'intermediario ad accantonare un sufficientemente abbondante capitale per essere *più protetti* da fallimenti e crolli economici, cosi come successo alla Lehman Brothers nel 2008. Questo, tuttavia, non è corretto in quanto un calcolo poco preciso lo è in entrambi i sensi: sia che si stimi di perdere molto e poi si perda poco, sia che si stimi di perdere poco e poi si perda molto. Inoltre, se il calcolo è accurato, significa che la realtà economica viene meglio rappresentata per cui laddove si verifichino dei segnali di crisi, questi saranno meglio catturati dai calcoli più accurati rispetto ai calcoli meno precisi.

L'utilizzo del machine learning, con l'obiettivo di migliorare l'accuratezza delle stime della LGD, in ambito di Credit Risk è già stato oggetto di interesse di ricercatori nel corso degli ultimi anni. Si citano, tra gli altri:

- Bastos, 2014: [1]Si interessa nel calcolare i Recovery Rate utilizzando algoritmi di Ensemble.
- Yao, Xiao, Jonathan Crook, e Galina Andreeva, 2017[2]: Studiano un modello di classificazione a due tempi per calcolare i Recovery Rate con l'utilizzo delle Support Vector Machine.

Sono poi molteplici i casi di studio portati avanti da società di consulenza per evidenziare quello che sembra ormai una verità confermata. Il machine learning comporta un miglioramento delle performances rispetto ai classici modelli utilizzati per il calcolo delle variabili di rischio.

Quali sono dunque le difficoltà dell'applicazione di questi e per quale motivo non sono tutt'oggi implementati immediatamente?

Il primo ostacolo è di tipo pratico. Gli algoritmi di Machine learning lavorano bene con i molti dati che le tecnologie odierne ci permettono di accumulare. Tuttavia, la raccolta di dati e l'implementazione di logiche di *Data Quality* e di *Data Governance* con il fine di renderli significativi e leggibili, comportano dei costi non indifferenti. Le grandi trasformazioni tecnologiche non sono sempre accompagnate da un veloce processo di adattamento (si pensi all'introduzione di energie rinnovabili o delle *smart grid* per il sistema elettrico).

Una parte di questi investimenti va sicuramente alla raccolta dati. Infatti, molti degli algoritmi di machine learning sono *supervised*, e quindi hanno bisogno di un riscontro reale sul comportamento da adottare (le label), altrimenti la fase di training risulta inefficiente o infattibile. Questo riscontro non è sempre scontato e in molti casi bisogna pagare persone che facciano attività ripetitive in modo da ottenere un numero sufficientemente grande di label che a sua volta permetta il training di algoritmi oppure un fornitore

[1] [32] Bastos, João A. 2014. "*Ensemble Predictions of Recovery Rates".* Journal of Financial Services Research 46: 177–93.
[2] [33] Yao, Xiao, Jonathan Crook, and Galina Andreeva. 2017. *"Enhancing two-stage modelling methodology for loss given default with support vector machines"*. European Journal of Operational Research 263: 679–89.

esterno che possiede l'informazione ricercata. Secondo il *The Economist*[1], l'azienda *Cognilytica* riconosce come la sola *Data Preparation* rappresenti un mercato da 1,5 miliardi (2019) e ci si aspetta cresca fino a 3,5 miliardi nel 2024.

Il secondo problema potrebbe essere indirizzato verso gli algoritmi stessi. Seppure vero che questi sembrano migliorare di molto l'accuratezza di certi *task,* è anche vero che non riescono a raggiungere un senso di generalizzazione come farebbe un cervello biologico. Questi algoritmi connettono input con output e non tutte le volte ragionano come ci si aspetta. Nella classificazione fra un cane ed un lupo delle nevi l'algoritmo potrebbe arrivare ad una accuratezza altissima individuando correttamente l'animale, avendo come input una fotografia di esso. Si potrebbe però scoprire che in realtà, il criterio utilizzato dall'algoritmo per individuare il lupo è la quantità di pixel bianchi, indicando quindi un'alta percentuale di neve nella fotografia. Questo potrebbe derivare dal fatto che le foto segnalate come di lupo durante la fase di training, visualizzano l'animale nel suo habitat naturale. L'algoritmo quindi è in grado di classificare correttamente fintanto che gli si mostrino fotografie di cani in casa e lupi delle nevi immersi, appunto, nella neve. Questa tipologia di anomalia deriva da *bias* nei dati di training di cui non è semplice accorgersene. Nel caso del lupo delle nevi la classificazione potrebbe non avere impatti consistenti, ma ci sono altri contesti in cui questi bias diventano letali. Discriminazione per via del colore della pelle, del sesso, dell'età, delle proprie origini, dei titoli di studio e di altri fattori potrebbero essere esempi di bias gravi intrinseci nei dati di input che viene dato all'algoritmo in fase di training che, non avendo una capacità critica o coscienza che sia, non può fare altro che ereditare tali bias. Impara ciò che gli è stato insegnato.

La stessa logica comporta ulteriori problemi di etica e di privacy o gestione delle informazioni sensibili in generale. Tutto questo viene aggravato dall'impossibilità odierna di spiegare il perché alcuni algoritmi decidono

[1] [34] The Economist, "*Not so big*", 13th June 2020

quello che decidono. In reti neurali complesse (*Deep Learning*), composte da centinaia di layer con migliaia di neuroni, non è facile (e in alcuni casi nemmeno fattibile) risalire al processo decisionale che ha dato vita ad una determinata decisione. Questo crea degli ovvi problemi in alcuni ambiti. Nel caso specifico della LGD, non sarebbe facilmente spiegabile il perché ad un cliente A è stato assegnato una determinata LGD mentre ad un cliente B è stata assegnata un'altra pur avendo questi, caratteristiche simili. Oppure perché ad un cliente A è stato assegnato una determinata LGD in un momento e poi successivamente gli è stata assegnata un'altra senza apparenti motivazioni. Ad un *Regulator,* che chiede una chiara visione delle metodologie decisionale degli algoritmi, non è sufficiente rispondere “non lo sappiamo”.

Situazioni peggiore possono verificarsi laddove queste decisioni abbiano un impatto sulla vita di un essere umano. Si pensi ad un’automobile che si autoguida che decide ad un certo punto di investire una persona in quanto non la si considera tale, o una diagnosi errata su un cancro che ritarda l’intervento dei dottori. È difficile spiegare poi alle famiglie delle vittime che la ragione per cui si sono commessi questi errori sono “non note” o “non ricavabili”.

Pur essendoci ostacoli più che validi per diffidare di questa nuova generazione di algoritmi, non bisogna tuttavia pensare che non saranno trovate soluzioni adatte. Il vantaggio dell’utilizzo di questi algoritmi non è solo teorico ma è pratico e tangibile. Consapevole o inconsapevolmente, siamo tutti a contatto con il machine learning ogni giorno.

In questo capitolo si cerca di confrontare i risultati in termini di errori commessi nella stima della LGD ottenuti dalla applicazione di quattro tra i più famosi algoritmi di machine learning (Decision Tree, Random Forest, Support Vector Machine e Rete Neurale) e quelli ottenuti dalla regressione multivariata utilizzata in Intesa Sanpaolo (ma anche in molti altri ambiti e società). Il confronto avverrà dal primo momento di applicazione dei citati

algoritmi fino alla fine del processo di selezione degli attributi, ottimizzazione dei parametri e perfezionamento delle performance.

Per lo svolgimento delle analisi e dei calcoli è stato utilizzato il linguaggio di programmazione *Python* con il supporto principale della libreria *Sklearn*. Il codice è visibile nell'Appendice B. Tranne alcuni casi che verranno specificati, la suddivisione fra *training set* e *test set* è stata rispettivamente del 80% e 20%.

4.1 Dataset e preparazione dei dati

Il dataset di partenza, fornito gentilmente dalla direzione di *Risk Management* del Gruppo Intesa Sanpaolo, è costituito da 134.901 istanze e 25 attributi più la LGD effettivamente osservata. Le LGD stimate/osservate sono quelle "totali", ovvero facenti riferimento all'intero arco temporale dal momento dell'entrata in default e non quindi le singole LGD stimate anno dopo anno. Si tratta delle posizioni al di sotto della *soglia di separazione* e quindi trattate con l'approccio statistico-analitico visto nel paragrafo 2.3.1 del Capitolo 2, dunque con un modello di regressione multivariata a variabili *dummies*. Inoltre, sono tutte posizioni facente parte dello Stage 3 spiegato nel paragrafo 2.2 del Capitolo 2, ovvero portafogli *Non Performing*. Non si considerano posizioni in Bonis, Inadempienza Probabile o Sconfini. Più specificamente, si tratta delle posizioni, opportunamente modificate con un fattore deciso dalla banca per rispettare la policy aziendale, utilizzate per la costruzione delle griglie della LGD di cui si è parlato precedentemente nel paragrafo 2.3.1.1 del Capitolo 2.

4.1.1 Prime elaborazioni degli attributi

Il primo passo verso l'avvio delle analisi è quello di adattare le variabili fornite alle condizioni richieste dai singoli algoritmi di machine learning. Molti data scientist riconoscono che due terzi del loro lavoro riguarda la gestione del dataset in termini di preparazione dei dati, *feature engineering*, *feature selection* e *data quality*. L'elenco degli attributi ricevuti con l'indicazione delle lavorazioni fatte su ognuno di essi, in primis, è riportato in Appendice A -*Tabella degli attributi ed elaborazioni,* e costituisce, se non due terzi, sicuramente una buona parte del tempo dedicato alla redazione di questo capitolo.

Da questa prima analisi sono state anche aggiunte ulteriori variabili nate da logiche di Feature Engineering quali One Hot Encode, scomposizione di variabili, Binning, trasformazione e combinazione di più variabili originali. Per il dettaglio si rimanda all'Appendice A - *Tabella nuovi attributi*.

Date le caratteristiche iniziali del dataset di partenza, non è stato necessario trattare valori missing e outliers in modo consistente in quanto quasi assenti (si veda l'Appendice A - *Tabella degli attributi ed elaborazioni*).

4.1.2 Esplorazione dei dati

La variabile da stimare, attraverso regressioni secondo le differenti logiche degli algoritmi spiegati nel Capitolo 3, è la LGD. Come già citato, questa variabile ha un dominio compreso fra 0 ed 1 e la distribuzione assume solitamente una forma ad U. Questa forma bimodale implica che ci sia una maggior probabilità di recuperare tutto il credito vantato così come di perderlo totalmente rispetto a situazioni intermedie di recupero parziale.

In **Figura 4.1** si illustra la distribuzione campionaria della LGD osservata nel dataset a disposizione:

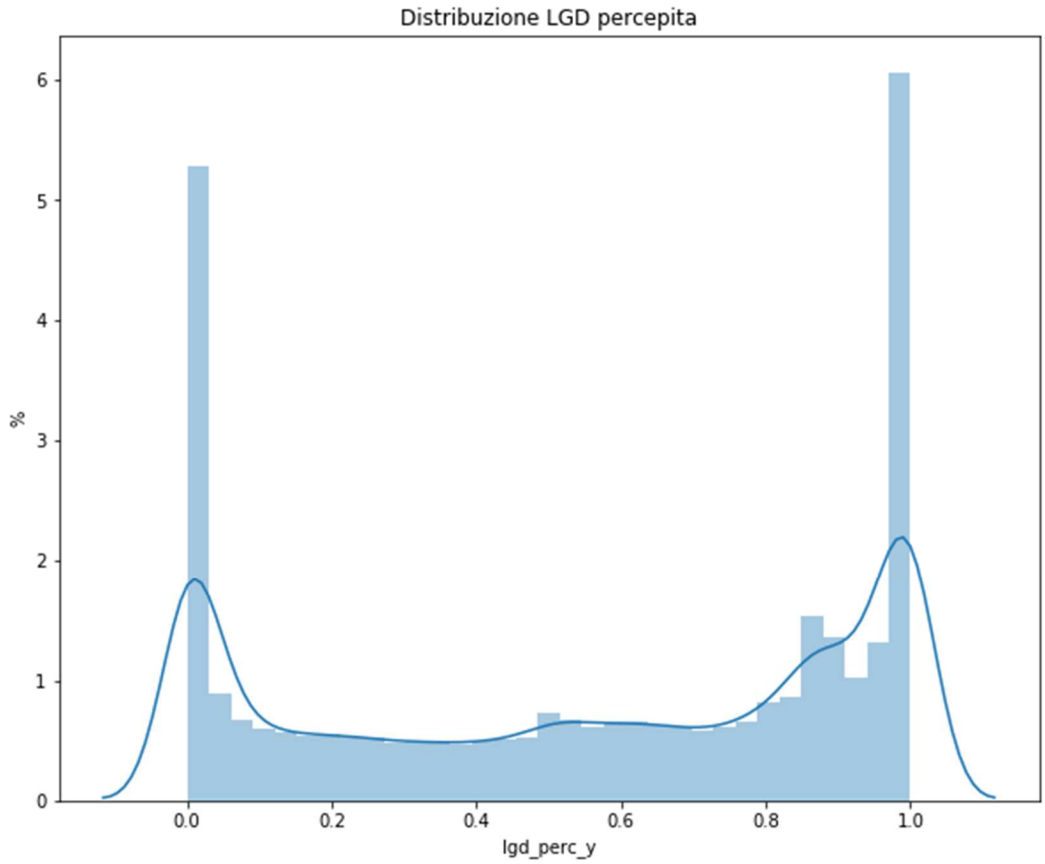


Figura 4.1: Distribuzione campionaria della LGD

Si evincono chiaramente le due mode agli estremi con una leggera prevalenza sulla perdita totale dei crediti vantati. È interessante notare come questa prevalenza tuttavia non sia una regola. Analizzando le perdite su segmenti di specifiche variabili, si nota come in alcuni casi gli scenari di perdite nulle siano superiori agli scenari di perdite totali.

Prendendo ad esempio la distribuzione della LGD per macro-tipologia di rapporto: *Breve Termine*, *Medio Lungo Termine Ipotecario* e *Medio Lungo Termine Non Ipotecario* (**Figura 4.2**), si nota come la distribuzione sia bilanciata nel caso di rapporti di medio lungo termine senza ipoteca (addirittura con una discreta probabilità di recuperare almeno metà dell'esposizione) mentre in presenza di Ipoteca la distribuzione viene nettamente sbilanciata verso lo scenario con perdite nulle. La motivazione è abbastanza intuitiva. I rapporti a breve termine rispettano lo sbilancio verso le perdite totali in quanto probabilmente sono compresi finanziamenti

di ammontare non elevati verso soggetti dal profilo leggermente più rischioso.

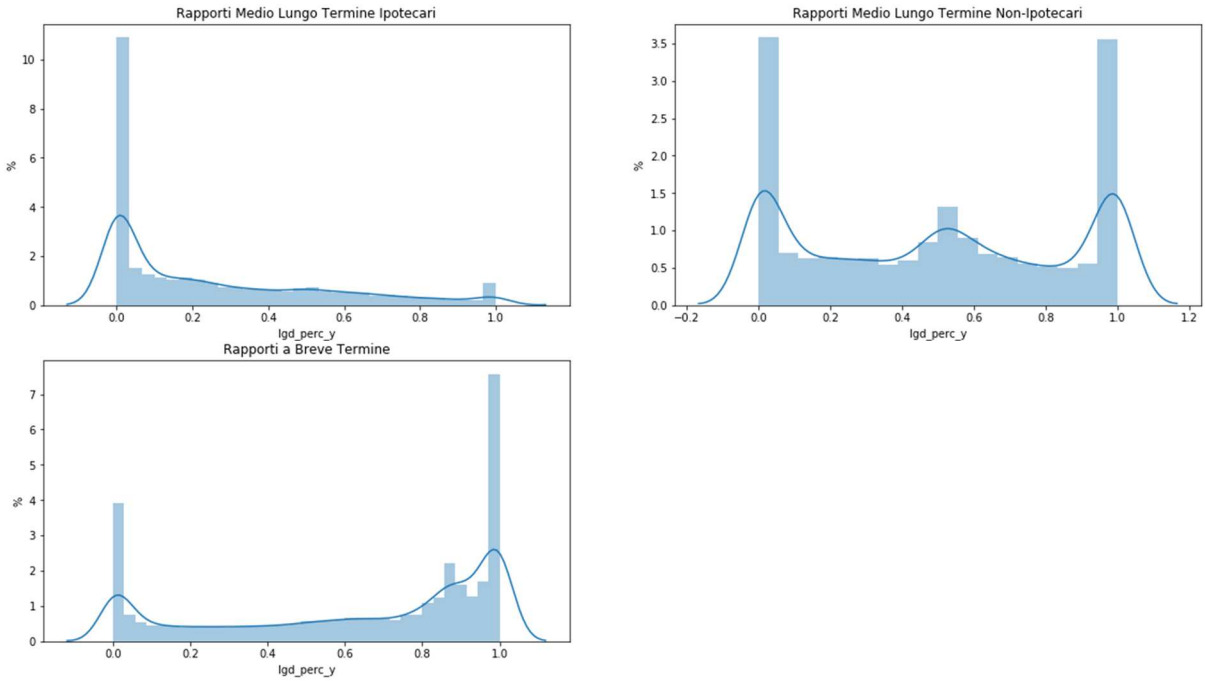


Figura 4.2: Distribuzione campionaria della LGD in base alla tipologia di rapporto.

Lo stesso è riscontrabile per quanto riguarda quelle controparti per le quali è riscontrabile la presenza di una *Garanzia Personale*, anche se con un minor impatto rispetto alla *Garanzia Ipotecaria* (**Figura 4.3**). Anche in questo caso si verifica come in caso di garanzia personale i recuperi totali siano superiori ai recuperi nulli.

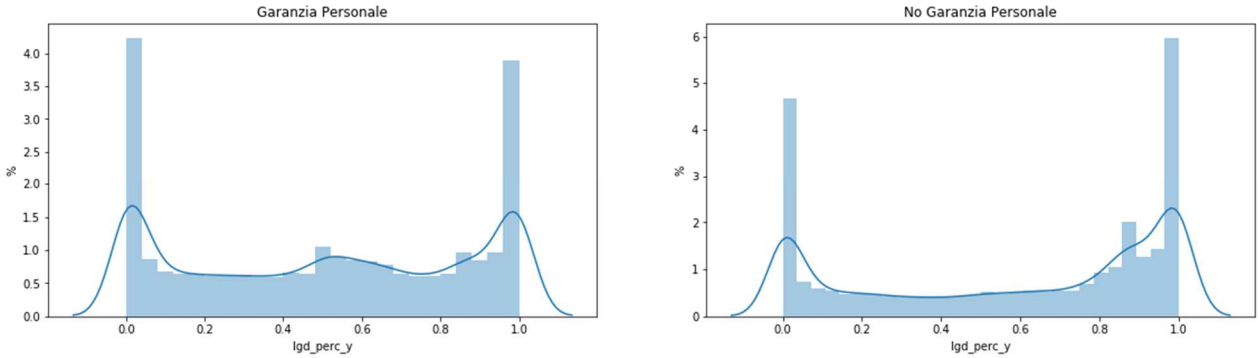


Figura 4.3: Distribuzione campionaria della LGD in presenza/assenza di garanzia personale

Facendo la stessa analisi sull'area geografica si nota come si abbia in entrambe le zone considerate, Nord e Sud (più isole), un rispetto dello sbilanciamento totale verso le perdite totali, con una leggera accentuazione nel Sud/Isole (**Figura 4.4**).

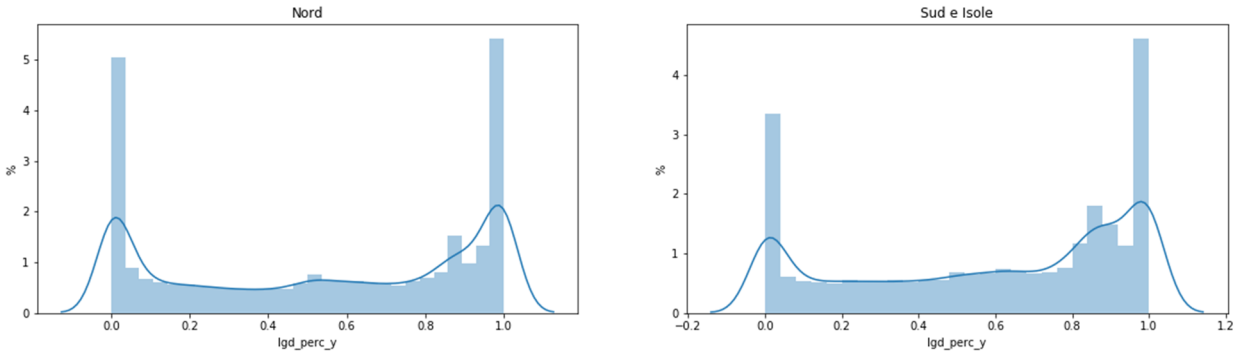


Figura 4.4: Distribuzione campionaria della LGD sulla base dell'area geografica.

Non ci sono ribaltamenti sostanziali nella logica originaria nemmeno per quanto riguarda la macro-classificazione del *Settore Economico* tra *Famiglie Produttrici/Consumatrici* e *Società Finanziarie/ Non Finanziarie*. Soltanto il caso delle Famiglie Produttrici riesce ad avere quel che sembra una parità di casi fra perdite totali e perdite nulle (**Figura 4.5**).

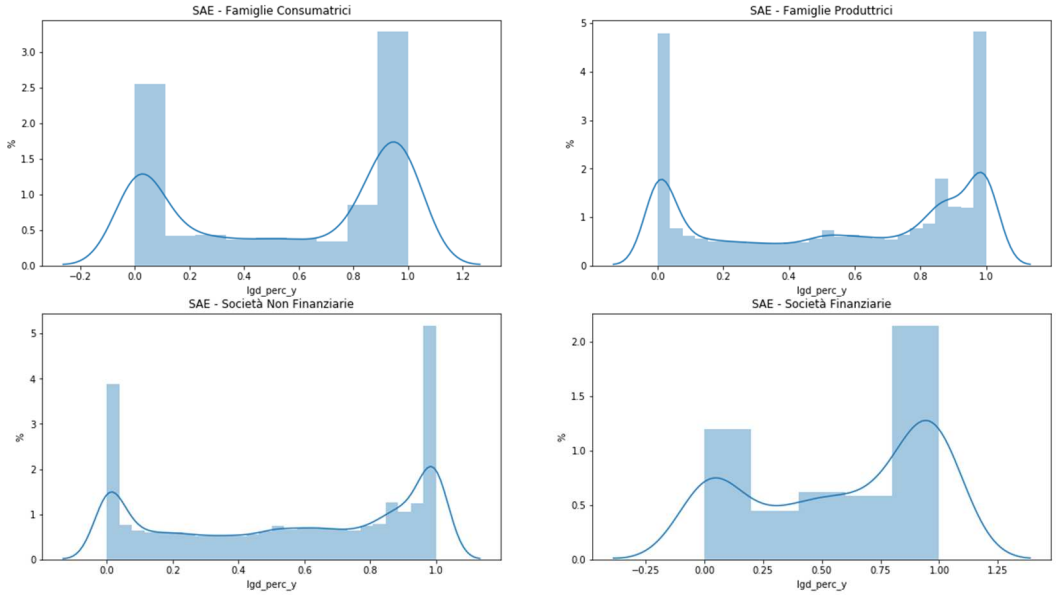


Figura 4.5: Distribuzione campionaria della LGD sulla base del SAE.

Guardando invece i 24 *Rami Attività Economica,* per i quali si rimanda all'Appendice A -*Tabella dominio SAE* per il dettaglio, soltanto *l'Agricoltura*

ha un maggior numero di casi di perdite nulle; *Holding Finanziarie e altro*, *Costruzione e materiale per costruzioni* ed i *Servizi* hanno due mode quasi equivalenti; tutti gli altri Rami seguono lo sbilancio verso lo scenario di perdita totale.

In **Figura 4.6** si evidenzia, in alto a sinistra, quella che è la distribuzione del periodo di permanenza in default delle posizioni. Negli altri riquadri si visualizza la distribuzione della LGD percepita per i segmenti: minore di 5 anni, tra 5 anni e 10 anni e maggiore di 10 anni. Per controparti che restano in default un tempo relativamente breve, gli scenari di recupero totale superano quelli di recupero nullo. L'opposto si verifica con l'aumentare degli anni in permanenza nello stato di default.

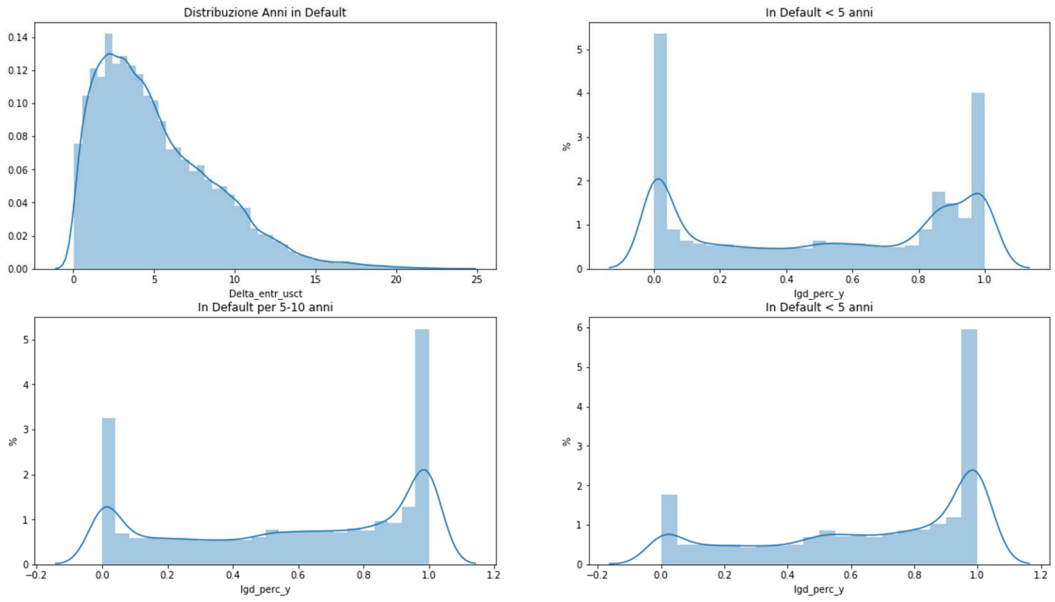


Figura 4.6: Distribuzione campionaria della LGD sulla base del tempo di permanenza in default.

La forma ad *U* della LGD percepita fa pensare che la sua *distribuzione reale* sia di tipo *Beta*[1]. Questa distribuzione, estremamente versatile, è definita

[1] Alcuni hanno anche dimostrato la validità di questa affermazione: Pesaran M.; Schuermann T.; Treutler B.; Weiner S. (2004). *Macroeconomic Dynamics and credit risk: a global perspective*. University of Cambridge.

nell'intervallo [0,1] ed in base ai due parametri $(\alpha, \beta) > 0$, le forme che può assumere sono davvero tante in quanto la definizione prevede:

$$Beta(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\int_0^1 x^{\alpha-1}(1-x)^{\beta-1}dx} \qquad per\ 0 < x < 1$$

$$Beta(x, a, b) = \frac{(x-a)^{\alpha-1}(b-x)^{\beta-1}}{\left(\int_0^1 x^{\alpha-1}(1-x)^{\beta-1}dx\right)(b-a)^{\alpha+\beta-1}} \qquad per\ a < x < b$$

(4.1), (4.2)

4.2 Feature Selection

Per via di tutte le operazioni di feature engineering svolte (e principalmente per via dell'One Hot Encode necessario per il trattamento delle variabili categoriche che altrimenti gli algoritmi non riuscirebbero a interpretare), dal dataset iniziale ci si trova uno nuovo composto da 343 variabili. Le dimensioni dell'iperspazio dato da un dataset con così tante variabili possono essere calcolate approssimativamente come spiegato di seguito. Si supponga che tutti i 343 attributi siano variabili booleane (affermazione quasi vera al netto delle varie date scomposte in numerici, che naturalmente hanno più valori, si veda l'Appendice A – *Tabella degli attributi ed elaborazioni*) e quindi aventi soltanto due valori. Il dataset contenente tutte le combinazioni possibili dei valori di questi attributi dovrebbe essere grande almeno (senza ripetizioni) 2^{343}. Si pensi che $2^{32} = 4\,294\,967\,296$, quindi un dataset con più di 10 volte *in meno* il numero di attributi del dataset utilizzato in questa sede, avrebbe bisogno di più di quattro miliardi di istanze per coprire tutte le possibili combinazioni. Visto

che il numero 2^{343} non è facilmente calcolabile, si riporta, per dare un'idea delle dimensioni in gioco, il numero seguente, ampiamente inferiore[1]:

$$2^{256} =$$

$$= 115\,792\,089\,237\,316\,195\,423\,570\,985\,008\,687\,907\,853\,269\,984\,665\,640\,564\,039\,457\,584\,007\,913\,129\,639\,936$$

Con un dataset composto soltanto di poco più di 134 mila istanze, la percentuale *potenziale* dell'iperspazio coperta è pari ad un numero così piccolo da poter essere considerata zero.

Per questo motivo, la *feature selection* si rende un'operazione estremamente necessaria. Riducendo il numero di variabili, si riduce esponenzialmente lo spazio di interesse. Sarebbe quindi ottimale rimuovere quelle variabili che non danno nessun tipo di informazione o contributo ai fini di stima (e quindi porzioni di iperspazio che in realtà non interessa esplorare) in modo da rendere la percentuale dell'iperspazio esplorato significativa, rendere l'iperspazio significativo e soprattutto i calcoli fattibili e significativi.

4.2.1 Matrice di Correlazione e Analisi Distribuzione Valori

Inizialmente sono stati utilizzati due approcci basici per ridurre il gran numero di variabili creato:

- *Matrice di correlazione di Pearson;*
- *Analisi della distribuzione dei valori degli attributi.*

Nel primo caso si va a calcolare la matrice di correlazione lineare di Pearson e viene applicato un algoritmo su di essa che vada ad eliminare quelle variabili che sono correlate (in senso assoluto) per più di una determinata soglia.

[1] [35]

$$M = \begin{bmatrix} \rho(x_1, x_1) = 1 & \rho(x_1, x_2) & \dots \rho(x_1, x_N) \\ \rho(x_1, x_2) & 1 & \dots \rho(x_2, x_N) \\ \dots \rho(x_1, x_N) & \dots \rho(x_2, x_N) & \dots \rho(x_N, x_N) = 1 \end{bmatrix} \quad (4.3)$$

In particolare, i passi dell'algoritmo sono:

- si individuano quelle coppie di attributi che abbiano una correlazione assoluta al di sopra di una determinata soglia;
- Per ogni attributo della coppia si calcola la correlazione con la variabile target;
- Si confrontano le due correlazioni calcolate al punto precedente;
- Si esclude l'attributo che abbia una correlazione assoluta con la variabile target inferiore.

In questo modo ci si aspetta di mantenere l'attributo che abbia un impatto maggiore, almeno lineare, nella stima. Attributi con un alto coefficiente di Pearson (in senso assoluto) fra di essi, sono attributi che forniscono all'algoritmo quasi lo stesso contenuto informativo e risulta pertanto altamente ridondante mantenerli entrambi.

Nell'*Analisi della distribuzione dei valori* per attributi, questi non vengono più presi in considerazione a coppie, ma singolarmente. Se una variabile, in alta percentuale, assume un unico valore, quasi come dire che assume soltanto quel valore, l'informazione che fornisce ai fini della stima è quasi nulla. Si vanno dunque ad eliminare quelle variabili che hanno un unico valore presente in una percentuale superiore ad una predeterminata soglia. Visto la generazione delle molte colonne per via dell'One Hot Encode, è intuitivo pensare che questo secondo approccio andrà ad eliminare tutte quelle variabili che, nell'attributo di origine, assumevano un valore molto raro.

Per fare un esempio, si pensi ad una variabile fittizia *Banca* che indica le operazioni di 3 diverse banche B1, B2 e B3. Si supponga dunque che su un dataset di 100 istanze, siano distribuite nel seguente modo:

Tabella 4.1: Distribuzione variabile fittizia Banca

Banca	Numero Istanze nel dataset
B1	46
B2	52
B3	2

Significa che nel dataset ci sono soltanto due operazioni provenienti dalla banca B3. Applicando l'One Hot Encode e guardando la distribuzione dei valori {0, 1}, si ottiene:

Tabella 4.2: Distribuzione variabile fittizia Banca dopo il OHE.

Valori	Banca_B1	Banca_B2	Banca_B3
0	54	48	**98**
1	46	52	**2**

La nuova variabile creata, *Banca_B3,* ha 98 istanze con valore 0 e soltanto 2 con valore 1. Applicando l'analisi della distribuzione si nota che nel 98% dei casi questa variabile assume un unico valore e quindi l'informazione potrebbe non essere di gran aiuto. Quindi, anche se fisicamente si elimina un attributo, quello che in realtà si sta facendo è eliminare un valore estremamente raro dall'attributo di origine.

Bisogna dunque fare attenzione alla presenza di attributi categorici che hanno molti valori rari (ad esempio il *Settore Economico* o il *Ramo Attività Economica*) in quanto la somma delle singole presenze in percentuali dei singoli valori rari, potrebbe risultare significativa. Si pensi ad un attributo pre-OneHotEncode contenente nel 60% delle sue istanze un unico valore e per il restante 40% altri 80 valori ognuno rappresentante una porzione del 0,5% sul totale. Applicato l'One Hot Encode si ottengono 81 nuove colonne di cui, applicando l'approccio di eliminazione citato con una soglia del 99%, 80 verrebbero eliminate in quanto avrebbero per il 99,5% il valore "0". Le 80

colonne eliminate, tuttavia, rappresentano il 40% dell'informazione dell'attributo originale. È chiaro si tratti di una perdita consistente.

Per questo motivo sono state analizzate le variabili originali contenti molti valori ed è stato confermato che la somma delle contribuzioni dei singoli domini non rappresenti una percentuale penalizzante.

Per ogni approccio sono state scelte due soglie: nel caso della correlazione di Pearson sono stati considerati due dataset contenenti attributi correlati per meno del 50% e per meno del 95% in senso assoluto; nel caso dell'analisi della distribuzione sono stati considerati due dataset escludendo attributi con un unico valore presente nel 99% e nel 99,9% del dominio. Successivamente questi due criteri sono stati combinati per generare l'insieme di quattro dataset mostrato in tabella:

Tabella 4.3: Combinazione di soglie vs numerosità attributi

Soglia Analisi distribuzione	Soglia Correlazione Pearson	Numero attributi dataset
99%	50%	52
99%	95%	78
99,9%	50%	166
99,9%	95%	198

Verranno costituiti dunque quattro diversi dataset nominati rispettivamente: *Dataset 99_50, Dataset 99_95, Dataset 999_50 e Dataset 999_95*. Le due soglie in entrambi i casi hanno lo scopo di considerare idealmente macro-scenari che si possono definire "*Ridotto*" e "*Abbondante*", in termini di attributi. Gli algoritmi impiegati, preferiscono più o meno attributi? È più forte il criterio della correlazione o quello dell'analisi della distribuzione? Sono sufficienti questi due criteri? Come quasi tutto nel Machine learning, non è possibile definire a priori quale scenario sia migliore, pertanto il modo migliore per capire quale di questi due macro-insiemi sia più vantaggioso per gli algoritmi è testando quest'ultimi su ognuno dei primi.

Sono stati quindi considerati i 4 algoritmi scelti nelle loro versioni di default (al netto del valore *random_state*[1] fissato a 42 in modo da permettere la riproducibilità dei risultati) della libreria in Python, sklearn. Le Support Vector Machine (SVM) impiegano un tempo *più che quadratico* al crescere delle istanze da considerare ed è quindi sconsigliato utilizzare i Kernel non lineari su dataset con un numero di istanze superiori alle 20-40 mila unità. Per evitare di perdere scenari vantaggiosi per strada, le SVM non lineari, con Kernel gaussiano RBF e sigmoidea, sono state testate con una divisione fra training set e test set ribaltata, ovvero il training set contenente il 20% delle istanze del dataset di partenza. I risultati ottenuti con il kernel gaussiano sono stati simili e leggermente svantaggiosi rispetto a quelli ottenuti con il kernel lineare. I risultati ottenuti con il kernel sigmoidale sono stati decisamente svantaggiosi rispetto agli altri casi. I dati sono sempre stati standardizzati prima dell'utilizzo delle SVM. Allo stesso modo, la rete neurale è stata provata con i dati normalizzati e standardizzati e si è notata una leggera preferenza per la standardizzazione.

In tabella si riportano i valori di default utilizzati [2] per gli algoritmi:

Tabella 4.4: Algoritmi, funzione sklearn e valori di defaul utilizzati in questa sede.

Algoritmo	Funzione sklearn	Valori di default
Decision Tree	*Sklearn.tree. DecisionTreeRegressor*	*criterion='mse', splitter='best',max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, presort='deprecated', ccp_alpha=0.0*

[1] Il *random_state* è un parametro utilizzato nelle librerie di sklearn per inizializzare i calcoli. Si sceglie dunque un punto di partenza nell'iperspazio delle possibilità. La conseguenza diretta e vantaggiosa è che i calcoli sono sempre riproducibili da terzi.
[2] Per il significato di ognuno di essi si rimanda al sito ufficiale https://scikit-learn.org

Random Forest	*Sklearn.ensamble. RandomForestRegressor*	*n_estimators=100, criterion='mse', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='auto', max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None, bootstrap=True, oob_score=False, n_jobs=None, random_state=None, verbose=0, warm_start=False, ccp_alpha=0.0, max_samples=None*
Support Vector Machine	*Sklearn.svm.SVR*	*kernel='rbf'/'sigmoid', degree=3, gamma='scale', coef0=0.0, tol=0.001, C=1.0, epsilon=0.1, shrinking=True, cache_size=200, verbose=False, max_iter=-1*
Support Vector Machine	*Sklearn.svm.LinearSVR*	*epsilon=0.0, tol=0.0001, C=1.0, loss='epsilon_insensitive', fit_intercept=True, intercept_scaling=1.0, dual=True, verbose=0, random_state=None, max_iter=1000*
Rete Neurale	*sklearn.neural_network.MLP Regressor¶*	*hidden_layer_sizes=(100,), activation='relu'/logistic, solver='adam', alpha=0.0001, batch_size='auto', learning_rate='constant', learning_rate_init=0.001, power_t=0.5, max_iter=200, shuffle=True, random_state=None, tol=0.0001, verbose=False, warm_start=False, momentum=0.9, nesterovs_momentum=True, early_stopping=False, validation_fraction=0.1, beta_1=0.9, beta_2=0.999, epsilon=1e-08, n_iter_no_change=10, max_fun=15000*

Nel fare le valutazioni sono state utilizzati come indicatori di performance:

- MAE (*Mean Absolute Error*): Si tratta dell'Errore Medio Assoluto:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{4.4}$$

- MSE (*Mean Squared Error*) – Errore Medio Quadratico (già trattato precedentemente):

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{4.5}$$

- RMSE (*Root MSE*) – Radice dell'Errore Medio Quadratico:

$$RMSE = \sqrt{MSE} \tag{4.6}$$

- R2 (*R squared*) – R quadro[1]:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y}_i)^2} \tag{4.7}$$

I risultati ottenuti con il calcolo di suddetti indicatori, dall'applicazione di suddetti algoritmi, sui cinque dataset predisposti (includendo quello completo di 345 attributi) sono visibili in **Tabella 4.5**.

[1] Per algoritmi non lineari, come il Decision Tree e il Random Forest, l'R^2 perde la sua forza d'interpretazione in quanto trattasi di un indicatore studiato per modelli lineari. E' stato mantenuto per fornire interpretazioni laddove venissero utilizzate le forme lineari degli algoritmi.

Tabella 4.5: Risultati algoritmi con valori di default sui diversi dataset prescelti

Algoritmo	*Indicatori*	*Dataset 99_50*	*Dataset 99_95*	*Dataset 999_50*	*Dataset 999_95*	*Dataset Iniziale*
		Numero di attributi				
		52	78	166	198	343
Decision Tree	MAE	0,329029734	0,304064839	0,323747047	0,299294554	0,300746624
	MSE	0,216566092	0,193970515	0,212056126	0,191316928	0,192674283
	RMSE	0,465366621	0,440420839	0,460495522	0,437397906	0,438946788
	R2	-0,34513234	-0,204787006	-0,317120106	-0,188305082	-0,196735864
Random Forest	MAE	0,271068189	0,244605661	0,267268408	0,242496385	0,242044411
	MSE	0,121165465	0,104189784	0,120381947	0,104067013	0,103820033
	RMSE	0,348088301	0,322784424	0,346961017	0,322594192	0,322211162
	R2	0,247418723	0,352857838	0,252285297	0,353620397	0,355154434
LinearSVR	MAE	0,297248702	0,283919641	0,313484877	0,299673202	0,307754961
	MSE	0,156909899	0,142434068	0,17171239	0,152541353	0,164272346
	RMSE	0,396118542	0,377404382	0,414381938	0,390565427	0,405305251
	R2	0,025403385	0,11531547	-0,066537645	0,052537238	-0,020326147
Rete Neurale	MAE	0,29404508	0,277288717	0,296882042	0,275668145	0,287423835
	MSE	0,129500123	0,117151384	0,1342178	0,122906226	0,124456864
	RMSE	0,359861256	0,342273844	0,366357475	0,350579843	0,352784444
	R2	0,195650608	0,272350927	0,166348242	0,236606532	0,226975234

Tenendo conto che le tonalità di blu più forti corrispondono ai valori preferibili, si notano immediatamente una serie di considerazioni.

- L'algoritmo con le migliori *performances* è il Random Forest, seguito dalla Rete Neurale, le SVM ed infine il Decision Tree. Il fatto che il Decision Tree sia l'algoritmo dai risultati meno accurati mentre il Random Forest, composto da cento Decision Tree, sia quello con i risultati migliori è una forte indicazione della forza degli algoritmi di *Ensamble*.
- Il Random Forest e il Decision Tree ottengono risultati leggermente migliori con più variabili, il che potrebbe implicare che esistano delle correlazioni non lineari che vengono perse con l'eliminazione degli attributi (eliminazione basata principalmente, infatti, su un indice di correlazione lineare). Viceversa, vale per le SVM e la Rete Neurale, che vengono impattati probabilmente dalla abbondanza di

correlazioni lineari fra le variabili e dalla ridondanza di attributi nei dataset più estesi.

- Si nota sempre un forte miglioramento nelle stime quando vengono esclusi gli attributi correlati al di sopra della soglia del 95% rispetto allo scenario in cui vengono esclusi gli attributi correlati al di sopra della soglia del 50%, indicando nuovamente che l'eliminazione di attributi per via di un indicatore di correlazione lineare comporta probabilmente una perdita di correlazioni non lineari nel dataset.

Prendendo in considerazione le osservazioni fatte si conclude che il più promettente degli algoritmi è il Random Forest (da confermare una volta effettuati i fine tuning dei singoli algoritmi) mentre il dataset *preferibile* è il *Dataset 99_95*. Infatti, pur avendo dei risultati leggermente inferiori rispetto al *Dataset Iniziale,* quando si utilizza il Decision Tree e il Random Forest, la differenza di attributi (78 vs 343), e quindi l'effort richiesto agli algoritmi in termini di numero di operazioni e tempistiche, fa preferire, in questa sede, il primo al secondo.

Si consideri inoltre, che i valori ottenuti dalla stima della regressione multivariata su questo dataset fornito, sono i seguenti:

Tabella 4.6: Risultati della Regressione Multivariata sul dataset iniziale

Algoritmo	Indicatori	Valori
Regressione Multivariata	MAE	0,32191144
	MSE	0,14153914
	RMSE	0,37621688
	R2	0,11236879

Pertanto:

- il MAE è inferiore *soltanto* a quello stimato dal Decision Tree nei *più svantaggiosi* dataset;
- Il RMSE è inferiore a quanto stimato dal Decision Tree e dalla SVR;
- Complessivamente ed in ogni scenario, il Random Forest e la Rete Neurale, senza fine tuning dei modelli e senza ulteriori elaborazioni

del dataset, si performano meglio rispetto alla regressione multivariata.

4.2.2 Recursive Feature Elimination

Sulla base delle considerazioni del precedente paragrafo, e focalizzandocisi principalmente sul *Dataset 99_95,* ci si interroga sulla possibilità di migliorare ulteriormente la combinazione di attributi da dare in pasto agli algoritmi prima di iniziare il fine tuning dei parametri. Sembra inefficiente usufruire di più di 78 variabili (e meno di 52), ma questo numero non può essere ulteriormente ridotto? Inoltre, riducendolo, quali delle 78 variabili sarebbero da escludere e quali da tenere? Ci si interroga quindi ancora una volta su *quante* e *quali*.

A questo scopo viene utilizzato il metodo *Recursive Feature Elimination - Cross Validation* [1](RFECV) con il Random Forest con validazione (*CrossValidation*) su tre diversi subset del dataset considerato. L'utilizzo del Random Forest non deriva soltanto dall'essere stato l'algoritmo più promettente durante le prime analisi, ma anche dal fatto che potrebbe meglio cogliere le correlazioni non lineari probabilmente perse durante l'eliminazione attraverso il coefficiente di Pearson. In questo modo, presumibilmente, si eliminano le variabili linearmente correlate che non hanno un contributo non-lineare agli effetti della stima. Utilizzando questo metodo quindi si otterranno le risposte ad entrambe le domande appena poste.

Il RFECV[2] ha individuato come migliore combinazione degli attributi, testato su tre diversi subset con eliminazione di due variabili per *run* (per un totale di 40 diversi combinazioni provate), un dataset contenente 56 variabili, riducendo quindi di 22 variabili il *Dataset 99_95,* che permettono

[1] Visto nel Capitolo 3, paragrafo 3.3.2.5.
[2] Con configurazione RFECV (RandomForestRegressor, step=2, cv=3), pertanto con 39 combinazioni di attributi per tre diversi subset in un totale di 117 run.

di ottenere la miglior performance media. In **Figura 4.7** è possibile vedere l'andamento dello score assegnato dall'algoritmo RFECV alle performance di ogni combinazione sui tre subset.

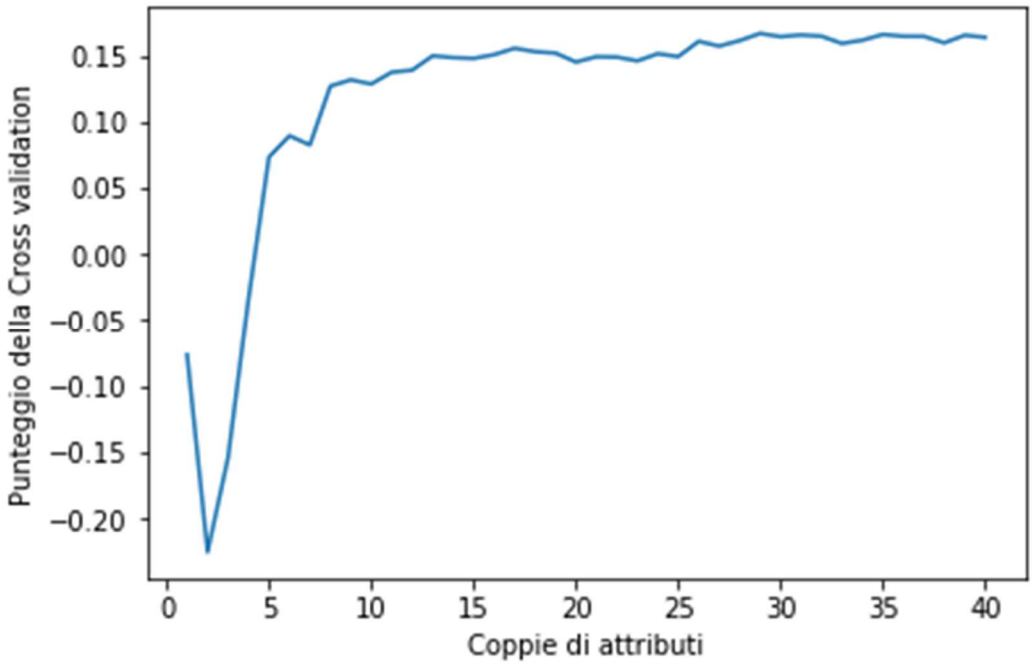


Figura 4.7: Andamento del punteggio di Cross-Validation con l'aumentare delle variabili utilizzate[1].

Si nota come già con poche variabili, 15-20, si ottengano risultati simili a quanto ottenuto con 78. I risultati del nuovo dataset, che verrà chiamato *Dataset_56*, sono messi a confronto per algoritmo con quelli del *Dataset 99_95,* nuovamente prendendo il training set come 80%, nella **Tabella 4.7**:

[1] Notare che sulle ascisse vengono riportate le *coppie* di attributi per via del fatto che ad ogni iterazione sono state eliminate 2 attributi. Il numero 25, ad esempio, fa quindi riferimento ad 25 coppie di attributi, quindi 50 attributi nel dataset.

Tabella 4.7: Confronto dei risultati per algoritmi testati sul Dataset 99_95 e Dataset 56.

Algoritmo	Indicatori	Dataset 56	Dataset 99_95
		Numero di attributi	
		56	78
Decision Tree	MAE	0,307851883	0,304064839
	MSE	0,196783841	0,193970515
	RMSE	0,443603247	0,440420839
	R2	-0,2222611	-0,204787006
Random Forest	MAE	0,245725609	0,244605661
	MSE	0,104671668	0,104189784
	RMSE	0,323530011	0,322784424
	R2	0,34986477	0,352857838
LinearSVR	MAE	0,283291025	0,283919641
	MSE	0,13958105	0,142434068
	RMSE	0,373605474	0,377404382
	R2	0,133036094	0,11531547
Rete Neurale	MAE	0,271534203	0,277288717
	MSE	0,116358867	0,117151384
	RMSE	0,341114155	0,342273844
	R2	0,277273401	0,272350927

Considerando le tonalità di blu più forti come i valori preferibili, si nota come in apparenza il Random Forest e il Decision Tree peggiorino le loro stime rispetto al caso precedente. Questo non deve trarre in inganno in quanto grazie al RFECV, testato su entrambe le combinazioni di attributi (78 e 56) e validate su tre diversi subset, è stato dimostrato come in media la combinazione di 56 attributi ottenga leggermente dei risultati più performanti. Il caso specifico è da considerare come un training set nel quale la combinazione di 78 attributi comporta una miglioria nella performance, tuttavia il RFECV spiega come la combinazione delle 56 variabili si comporti meglio con “nuove istanze” (che sarà lo scenario a target una volta che l’algoritmo sarà in produzione). Le SVM e la Rete Neurale hanno visto un miglioramento netto nelle loro stime.

Il RFECV poteva essere applicato fin dall’inizio sul dataset di partenza utilizzando tutti gli algoritmi e poi confrontando i dataset ottenuti per dare

uno score ai diversi attributi. Risultando questo processo abbastanza oneroso sia in termini di tempistiche che di memoria del calcolatore, è stato preferito adottare un approccio guidato dalle intuizioni derivanti dalle prime analisi.

Gli attributi del *Dataset 56* sono visibili nell'Appendice A - *Tabella Dataset 56*.

4.2.3 Analisi delle Componenti Principali

Visto il problema esplicito di scarsa variabilità "catturata" dagli algoritmi utilizzati, anche per via della distribuzione della variabile target stimata, ci si chiede se l'applicazione dell'algoritmo di calcolo delle Componenti Principali, visto nel Capitolo 3 paragrafo 3.3.2.4, avente l'obiettivo di ridurre la dimensione dei dataset catturando la variabilità di essi per quanto più possibile, potesse aiutare in questo senso. Si potrebbe, ad esempio, scoprire che, a valle di rotazioni e quindi di proiezione dei dati in uno spazio di inferiori dimensioni, sono sufficienti soltanto un numero esiguo di componenti principali nel nuovo spazio, per spiegare un'alta percentuale della variabilità del precedente spazio senza impattare fortemente le performances.

Costruendo un grafico in cui si riporta la cumulata della varianza spiegata al crescere del numero di componenti principali (e quindi di dimensioni del nuovo spazio), la "scoperta" descritta avverrebbe se la cumulata avesse un'accelerazione, già per i primi valori, andando rapidamente verso un limite di 1 al crescere del numero di componenti.

Se la curva invece si avvicinasse alla bisettrice significherebbe che non è così conveniente sacrificare l'informazione riguardante la varianza dei dati per ridurre la dimensione in quanto le componenti principali catturate pesano tutte allo stesso modo.

Il calcolo delle CP è stato eseguito sul dataset di partenza ottenendo la curva cumulata di **Figura 4.8**:

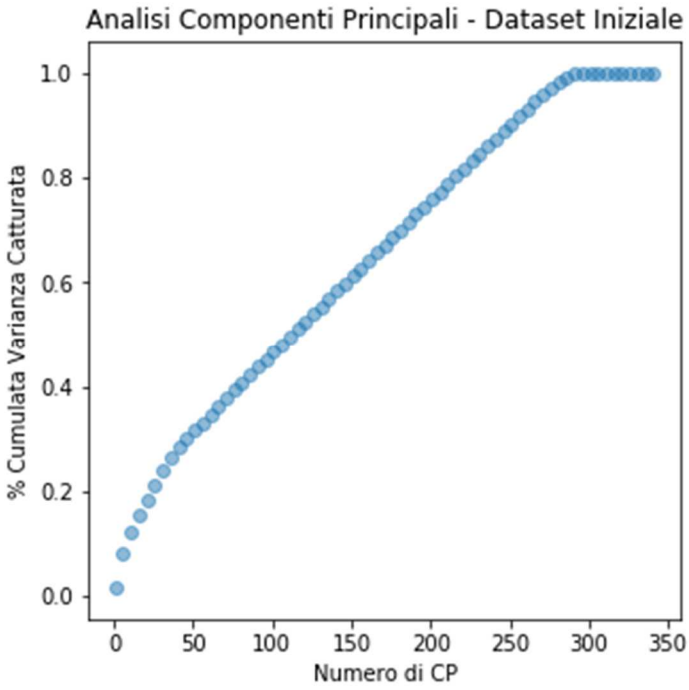


Figura 4.8: Cumulata della varianza all'aumentare il numero di Componenti Principali (CP) utilizzate – Dataset Iniziale

Sicuramente si evince come le ultime 50 componenti principali non aggiungono ulteriore informazione rispetto alle precedenti. Tuttavia, risulta anche evidente che, essendo la curva così vicina alla bisettrice, l'applicazione della PCA non aiuti a catturare la variabilità del dataset in modo conveniente con la riduzione della dimensione. Si esclude quindi l'ipotesi tale per cui la riduzione della dimensione poteva essere stata fatta come primo approccio con la PCA.

La stessa analisi viene fatta sul *Dataset 56* del paragrafo precedente, in quanto il più promettente, per capire se esiste ulteriore possibilità di ridurne la dimensione senza grandi sacrifici di performance. La curva cumulata può essere osservata nella **Figura 4.9**:

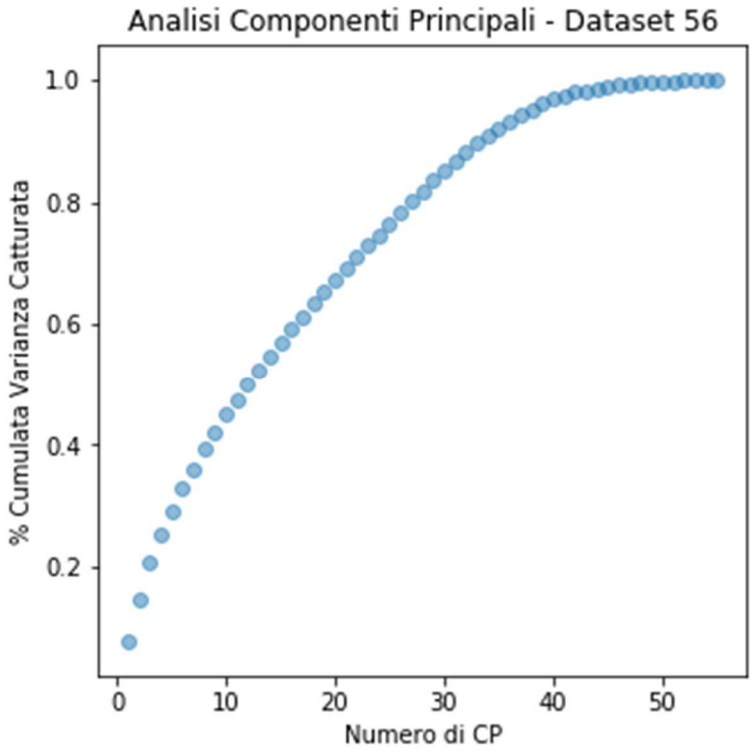


Figura 4.9: Cumulata della varianza all'aumentare il numero di Componenti Principali (CP) utilizzate – Dataset 56

Rispetto al caso precedente, pur restando non un grande vantaggio, il calcolo delle CP evince come si potrebbe leggermente ridurre la dimensione del Dataset 56 non perdendo così tanta informazione sulla varianza.

In **Tabella 4.8** vengono riportati i risultati ottenuti con i quattro algoritmi nel caso di utilizzo della PCA *senza riduzione* della dimensione ed utilizzo della PCA *con riduzione* della dimensione di 10 componenti. I dati sono stati standardizzati per essere trasformati in componenti principali e successivamente standardizzati nuovamente per alimentare gli algoritmi.

Tabella 4.8: Confronto dei risultati per algoritmi testati sul Dataset 56, 45 con PCA e 56 con PCA.

Algoritmo	Indicatori	Dataset 56	CP45	CP56
		Numero di attributi		
		56	45	56
Decision Tree	MAE	0,307851883	0,342305424	0,348314104
	MSE	0,196783841	0,225166	0,232696236
	RMSE	0,443603247	0,474516596	0,482385983
	R2	-0,222261096	-0,398547975	-0,445319675
Random Forest	MAE	0,245725609	0,283883387	0,283541363
	MSE	0,104671668	0,127948471	0,127255928
	RMSE	0,323530011	0,357698855	0,356729488
	R2	0,34986477	0,205288211	0,209589725
LinearSVR	MAE	0,283291025	0,283381078	0,282889955
	MSE	0,13958105	0,140942266	0,138462363
	RMSE	0,373605474	0,375422783	0,372105312
	R2	0,133036094	0,12458133	0,139984467
Rete Neurale	MAE	0,271534203	0,269087204	0,267956824
	MSE	0,116358867	0,118171834	0,114653187
	RMSE	0,341114155	0,343761304	0,338604765
	R2	0,277273401	0,266012722	0,287867695

Considerando nuovamente le tonalità più forti di blu come i valori preferibili, l'evidenza immediata è che la PCA non comporti una trascurabile riduzione a priori nelle performance a fronte di una riduzione della dimensione. Le SVM e la Rete Neurale ottengono un leggero vantaggio dall'utilizzo della PCA anche pur riducendo la dimensione del dataset. Il Random Forest e il Decision Tree peggiorano significativamente le loro performance.

Per gli algoritmi di SVR e Rete Neurale (e specialmente per quest'ultimo), le performance migliorano sensibilmente anche con la sola trasformazione in queste componenti. Si tratta sicuramente di un caso insolito e la spiegazione potrebbe risiedere nel fatto che la proiezione di dati nel nuovo spazio equidimensionale renda i dati più "leggibili" e meglio distribuiti rispetto al dataset pre-trasformazione.

Per il Random Forest e il Decision Tree si conferma la non convenienza nell'utilizzo della PCA né per la riduzione delle dimensioni né per il miglioramento delle performance.

In generale i risultati erano anticipabili anche dalla forte presenza di variabili non continue all'interno dei dataset utilizzati che non permettono un ottimale funzionamento della PCA, più performante con variabili continue.

4.3 Fine Tuning dei modelli

Il fine tuning dei vari modelli è stato fatto sulla combinazione di variabili scelte nei paragrafi precedenti a valle della Data Selection.

In **Tabella 4.9** sono visibili le combinazioni di parametri provate per ogni algoritmo, dunque la combinazione più performante. È stata utilizzata la funzione *RandomizedGridSearch* di Sklearn, vista nel Capitolo 3 paragrafo 3.5, in modo da navigare ampiamente l'iperspazio delle possibilità senza tuttavia provarle tutte. In particolare, per ogni algoritmo sono state utilizzate 20 diverse combinazioni di parametri testate con l'approccio *Cross-Validation* su tre diversi subset del training set per un totale di 60 *run*. In particolare:

- Decision Tree:
    - Criterion: Si tratta del criterio di ottimizzazione dell'albero, basato dunque sull'errore medio quadratico, il valore medio assoluto e l'errore medio quadratico di friedman[1].
    - Min_sample_split: Il numero minimo nel quale dividere un singolo nodo padre in nodi figli.
    - Min_sample_leaf: Il minimo numero di campioni richiesti in un nodo foglia.

[1] Si tratta di una versione pesata dell'errore medio quadratico proposta da J. Friedman in: Jerome H. Friedman, "Greey Aproximation: A Gradient Boosting Machine", February 24, 1999.

Tabella 4.9: Combinazioni potenziali testate e miglior parametri.

Algoritmo	Potenziali Combinazioni da testare	Migliore combinazione
Decision Tree	criterion = mse, mae, friedman_mse min_samples_split = [2, 10, 20, 40] max_samples_leaf = [1, 20, 40, 100] max_leaf_nodes = [5, 20, 100] max_depth = [2, 6, 8, None]	Criterion = friedman_mse min_samples_split = 40 min_samples_leaf = 100 max_leaf_nodes = 100 max_depth = 8
Random Forest	n_estimators = [50, 100, 300, 500, 1000] max_depth = [100, None] min_samples_split = [2, 4] min_samples_leaf = [1, 2]	n_estimators = 1000, min_samples_split = 2, min_samples_leaf = 2, max_depth = None
Support Vector Machine - LinearSVR	epsilon = [0, 0.3, 0.5, 0.7, 1] C = [0.5, 1, 2, 5]	epsilon = 0.3 C = 0.5
Support Vector Machine - SVR	kernel = [linear, poly, rbf, sigmoid] C = [0.5, 1, 2, 5] gamma = [auto, scale] epsilon = [0, 0.3, 0.5, 0.7, 1]	kernel= rbf gamma= scale epsilon= 0.3 C= 0.5
Neural Network Senza PCA	hidden_layer_sizes[1] activation= [identity, logistic, tanh, relu] solver= [sgd, adam] max_iter= [100, 200]	hidden_layer_sizes = (100,) activation = logistic solver = adam max_iter = 100
Neural Network Con PCA	hidden_layer_sizes activation= [identity, logistic, tanh, relu] solver= [sgd, adam] max_iter= [100, 200]	hidden_layer_sizes = (100,) activation = Relu solver = adam max_iter = 100

  - Max_leaf_nodes: Il numero massimo di campioni richiesti in un nodo foglia
  - Max_depth: La massima profondità dell'albero.
- Random Forest:
  - N_estimators: Il numero di alberi decisionale da attivare in parallelo

[1] Sono stati ipotizzate quattro diverse strutture: 1) Un solo layer con 100 neuroni (default); 2) Ventuno layer con neuroni decrescenti da 60 nel primo a 5 nell'ultimo; 3) Nove layer con neuroni decrescenti da 60 a 5; 4) Cinque layer con neuroni decrescenti da 50 a 5.

    - max_depth: La massima profondità degli alberi.
    - min_samples_split: Il numero minimo nel quale dividere un singolo nodo padre in nodi figli.
    - min_samples_leaf: Il minimo numero di campioni richiesti in un nodo foglia.
- Support Vector Machine (Lineare)
    - Epsilon: Si tratta del valore ε utilizzato nella formula (3.26) del Capitolo 3;
    - C: Si tratta del parametro di regolazione utilizzato in (3.27) del Capitolo 3;
- Support Vector Machine (Kernels)
    - Kernel: Si tratta della funzione kernel da utilizzare: Polinomiale, sigmoidale, lineare e gaussiana rbf.
    - Gamma: Si tratta del parametro richiesto nel caso si utilizzi la funzione kernel rbf, polinomiale o sigmoidea.
    - Epsilon: Si tratta del valore ε utilizzato nella formula (3.26) del Capitolo 3;
    - C: Si tratta del parametro di regolazione utilizzato nella formula (3.27) del Capitolo 3;
- Rete Neurale:
    - Hidden_layer_size: Si specificano il numero di layer intermedi fra quello in input e quello in output con anche il numero di neuroni per ognuno di essi;
    - Activation: È la funzione d'attivazione del neurone;
    - Solver: Si tratta del metodo correzione dei pesi;
    - Max_iter: Il numero massimo di iterazioni senza ottenere un miglioramento delle performance superiore al valore di default (solitamente 0,0004).

Nel caso delle SVM sono state provate anche le funzioni Kernel polinomiali, gaussiana rbf e sigmoidea con un dataset *ribaltato*, ovvero avente il training set pari soltanto al 20%. La combinazione più promettente è stata testata con dataset via via più grandi fino al 50% di training set,

ottenendo risultati sempre più affinati. Per le Reti Neurali, la parametrizzazione è stata fatta anche con l'utilizzo della PCA ottenendo però come miglior parametri proprio quelli di default. Inoltre, ottenendo delle stime al di sotto dello zero, alcune stime della Rete Neurale sono state forzate a zero per dare un senso economico al risultato.

In **Tabella 4.10** si possono osservare i risultati ottenuti con suddette combinazioni di parametri sul Dataset 56. Questi risultati vengono confrontati sia con quanto ottenuto utilizzando i valori di default, sia con i risultati ottenuti dalla regressione multivariata.

Tabella 4.10: Confronto risultati per algoritmo di regressione multivariata, algoritmi con parametri di default e algoritmi con parametri ottimizzati.

Algoritmo	Indicatori	Dataset 56		
		Regressione Multivariata	Parametri di Default	Parametri Ottimizzati
Decision Tree	MAE	0,321911443	0,307851883	0,263611852
	MSE	0,141539144	0,196783841	0,111759295
	RMSE	0,376216884	0,443603247	0,334304195
	R2	0,112368792	-0,222261096	0,3058422
Random Forest	MAE	0,321911443	0,245725609	0,243174209
	MSE	0,141539144	0,104671668	0,102155773
	RMSE	0,376216884	0,323530011	0,319618168
	R2	0,112368792	0,34986477	0,365491462
LinearSVR	MAE	0,321911443	0,283291025	0,306150611
	MSE	0,141539144	0,13958105	0,128446189
	RMSE	0,376216884	0,373605474	0,358393902
	R2	0,112368792	0,133036094	0,202196789
SVR	MAE	0,321911443	0,283291025	0,29374687
	MSE	0,141539144	0,13958105	0,119892796
	RMSE	0,376216884	0,373605474	0,346255391
	R2	0,112368792	0,133036094	0,248612382
Rete Neurale	MAE	0,321911443	0,271534203	0,26306628
	MSE	0,141539144	0,116358867	0,112913071
	RMSE	0,376216884	0,341114155	0,336025402
	R2	0,112368792	0,277273401	0,298675883

Prendendo ancora una volta i valori blu con tonalità forti come i più positivi, si evincono una serie di novità. Si nota innanzitutto che, dopo la parametrizzazione degli algoritmi, tutti e quattro gli algoritmi performano meglio della regressione multivariata, compreso il Decision Tree. Infatti, quest'ultimo ora ottiene dei risultati migliori anche delle SVM diventando performante quasi come la Rete Neurale. Il Random Forest si conferma l'algoritmo più adatto alla stima della LGD fra quelli utilizzati in questa sede.

Le SVM funzionano leggermente meglio con Kernel non lineari. In apparenza e paradossalmente sembra che la SVM lineare commetta più errori con i valori ottimizzati rispetto all'utilizzo dei semplici valori di default. Tuttavia, bisogna tener conto della *Cross-Validation*. Ancora una volta si tratta di un caso in cui i valori di default permettono di ottenere per lo specifico dataset (training set 80%) dei risultati migliori rispetto ai supposti parametri ottimizzati. Quest'ultimi però sono stati validati su tre differenti test subset ottenendo un punteggio superiore ai valori di default. Si presume dunque che in un ambito più esteso e generalizzato permettano di ottenere dei risultati medi superiori.

Si noti che i risultati della Rete Neutrale con PCA non è stato riportato in quanto coincide con quanto già ottenuto nel paragrafo 4.2.3. Le performance di default del Decision Tree sono state colorate di grigio per evidenziare come questi indicatori siano degli *outliers* all'interno della tabella.

4.4 Provando l'Ensemble

Visto che il Random Forest è l'algoritmo più performante sembrerebbe che anche in questa sede sia la combinazione di tanti differenti algoritmi ad avere la meglio sulle stime, quindi un approccio di *Ensemble*. È per questo motivo che nasce l'idea di provare a creare un algoritmo composto, senza

la presunzione di migliorare nettamente le stime ottenute, ma più per capire, esattamente come fatto nei paragrafi precedenti, se potrebbe essere una buona strada da intraprendere per affinare le performance.

Si provano in sintesi due approcci diversi:

- Si utilizzano tutti e quattro gli stimatori precedenti e si fa una media delle singole stime per ottenere la stima finale, da confrontare con il valore osservato di LGD;
- Si utilizza un algoritmo di Random Forest che generi la stima finale ricevendo come dati di input le singole stime dei singoli algoritmi (inclusa la media di questi).

In **Tabella 4.11** sono visibili i risultati dei due approcci confrontati con i risultati del Random Forest.

Tabella 4.11: Confronto risultati per Random Forest, Ensemble – Media e Ensemble – Random Forest.

Algoritmo	Indicatori	Dataset 56 - Performance
Random Forest	MAE	0,243174209
	MSE	0,102155773
	RMSE	0,319618168
	R2	0,365491462
Ensamble - Media	MAE	0,264776538
	MSE	0,106548482
	RMSE	0,32641765
	R2	0,338207527
Ensamble - Random Forest	MAE	0,241174897
	MSE	0,114146154
	RMSE	0,337855226
	R2	0,291016973

Si nota come non risulti conveniente nessuno dei due approcci rispetto all'utilizzo del Random Forest. Nel caso dell'*Ensemble – Media* si ottengono valori che intuitivamente potevano essere previsti. Trattandosi della media

delle stime, si ottengono valori migliori rispetto agli algoritmi meno performanti (SVM e Rete Neurale) e si ottengono valori meno performanti rispetto agli algoritmi più performanti (Random Forest e Decision Tree).

Nel caso dell'*Ensemble – Random Forest*, da un lato si riesce ad ottenere un errore medio assoluto significativamente più basso mentre dall'altro si ottiene un errore medio quadratico superiore. Questa evidenza potrebbe essere letta come: l'algoritmo di Ensemble in media sbaglia di meno ma quando succede sbaglia in modo più marcato rispetto al Random Forest.

4.5 Analisi della distribuzione delle stime prodotte

In questo breve paragrafo si cerca di illustrare le differenti distribuzioni che le stime dei singoli algoritmi, nella versione ottimizzata, assumono con lo scopo di ricavare ulteriori intuizioni.

Partendo dagli algoritmi meno performanti, si propone in **Figura 4.9** la distribuzione delle stime prodotte dalle SVM, lineare e non. La differenza fra le due distribuzioni è sottile. Entrambe sono sbilanciate verso lo scenario di perdite consistenti ma non prevedono, a differenza della distribuzione della LGD osservata rappresentata in **Figura 4.1**, un picco attorno all'unità.

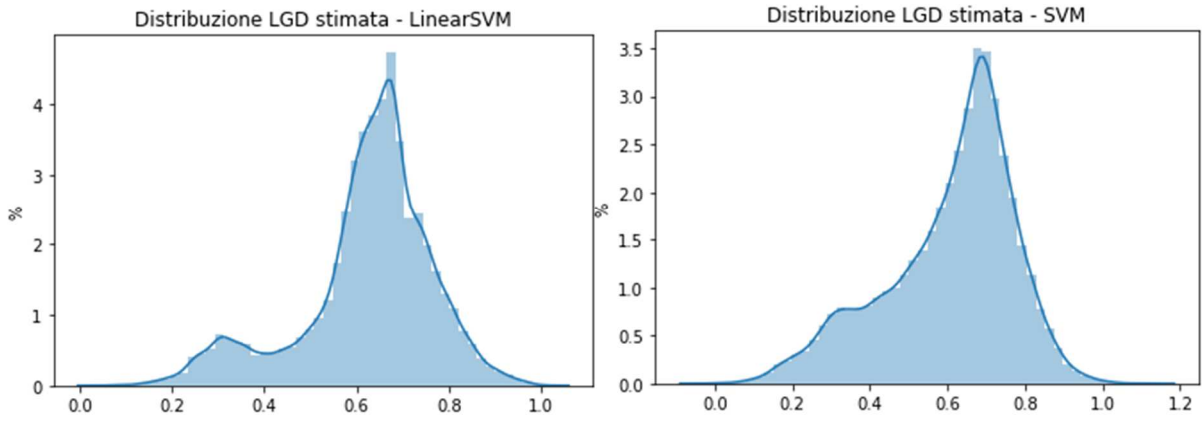


Figura 4.9: Distribuzione delle stime prodotte dalle SVM lineare (sinistra) e la SVM Gaussiana rbf (destra).

Entrambe le distribuzioni infatti hanno un picco intorno ai valori 0,62 – 0,67. Nel caso della SVM non lineare risulta quasi simmetrico, mentre nel caso delle SVM lineare è leggermente asimmetrico verso valori inferiori. È probabile che questa differenza sia la causa principale del piccolo gap prodotto nelle stime. Visto che la distribuzione della LGD osservata tende ad avere uno sbilancio vero lo scenario di perdite consistenti, l'asimmetria della SVM lineare vero scenario con perdite meno consistenti potrebbe risultare penalizzante.

In **Figura 4.10** si può vedere la distribuzione delle stime della Rete Neurale.

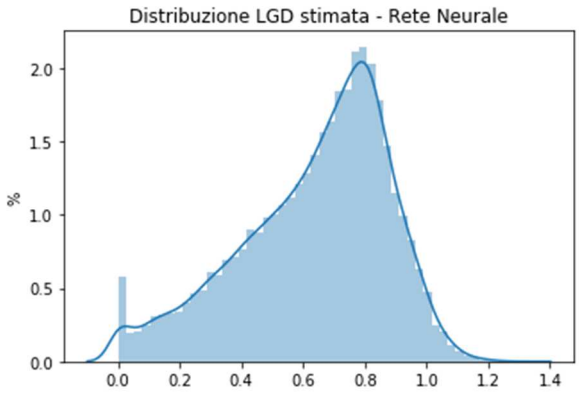


Figura 4.10: Distribuzione delle stime prodotte dalla Rete Neurale.

Anche in questo caso l'algoritmo è leggermente sbilanciato verso lo scenario di perdite consitenti con un picco intorno al valore 0,8. Si nota un accenno di massimo locale attorno allo zero, questo deriva principalmente dalla forzatura a zero dei valori negativi.

In **Figura 4.11** si osserva la particolare distribuzione del Decision Tree. È curioso notare come pur avendo indicatori di performance simili (**Tabella 4.10**), le distribuzioni della stima del Decision Tree e della Rete Neurale differiscono in modo visibile.

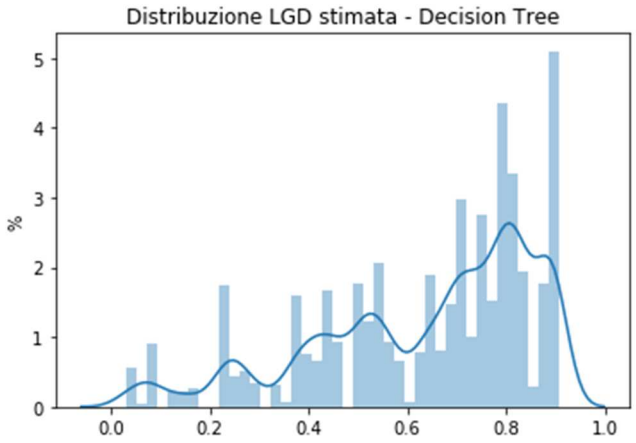


Figura 4.11: Distribuzione delle stime prodotte dal Decision Tree.

La segmentazione visibile nella distribuzione del Decision Tree, deriva basicamente dalle logiche dell'algoritmo. Infatti, come spiegato nel Capitolo 3, il Decision Tree nel caso della regressione individua prima delle *aree* nelle quali sono presenti più istanze del training set e definisce come stima la media della LGD di queste istanze. Questo significa che tutte le istanze che cadono in una determinata *area*, avranno la medesima stima. Per questo motivo nella distribuzione è possibile osservare come ci siano valori molto più probabili, rispetto alle distribuzioni finora viste, e come ci siano valori che non risultano mai prodotto della stima del Decision Tree. Anche in questo caso i valori più probabili sono sbilanciati verso gli scenari con perdite più consistenti con un picco intorno al valore 0,8.

In **Figura 4.12** si illustra infine la distribuzione delle stime prodotte dal Random Forest. Si nota subito la similitudine con la distribuzione della Rete Neurale con la differenza che il Random Forest tende a spingersi oltre l'unità in modo più decisivo (prevedendo quindi costi indiretti più consistenti). Rispetto al Decision Tree si osserva come l'effetto a *segmenti* venga perso per via della combinazione di mille diversi alberi decisionali.

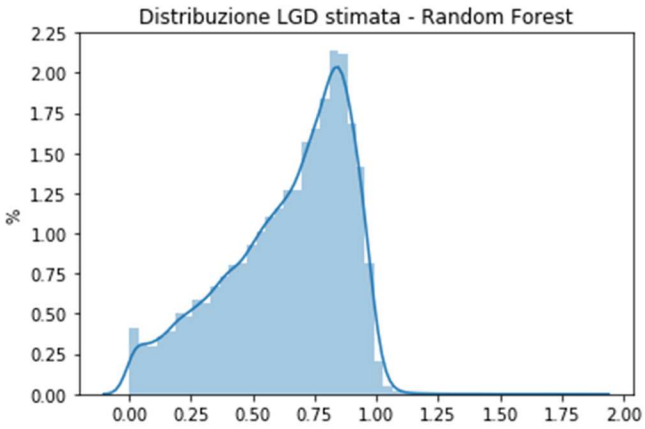


Figura 4.12: Distribuzione delle stime prodotte dal Random Forest.

È opportuno notare come nessuno degli algoritmi utilizzati colga il picco nell'origine che si evince nella distribuzione della LGD osservata. Pur essendo un elemento di errore statistico, questa anomalia implica economicamente una sovrastima della LGD, ovvero gli algoritmi si attendono di perdere in molti più casi rispetto a quelli effettivamente osservati, un'alta percentuale dell'esposizione vantata, seguendo di conseguenza un approccio *prudenziale*.

Pertanto, pur riducendo l'errore commesso rispetto alla regressione multivariata, il miglioramento non deriva dal riuscire a prevedere meglio quando si perde poco, ma dal riuscire a prevedere meglio quando si perde tanto.

4.6 Possibili Evolutive

È chiaro che quanto fatto non è che l'esplorazione di una micro-parte dell'iperspazio delle possibilità, visti i limiti strutturali evidenti. Si tenta pertanto in questo breve paragrafo di elencare quelli che potrebbero essere eventuali evolutive nello studio del calcolo della LGD con l'utilizzo di algoritmi di machine learning rispetto a quanto fatto in questa sede:

- *Dataset più ricco*: Il dataset di partenza potrebbe facilmente essere arricchito con informazioni anche non direttamente correlate con la variabile target. Uno dei pregi più citati del machine learning è la capacità di trovare correlazioni nascoste laddove normali algoritmi o la semplice logica non ci arrivano;
- *Feature Engineering più approfondito*: La combinazione delle variabili a disposizione può essere la più disparata possibile. In questo elaborato sono state fatte soltanto quelle che avevano un esplicito significato di business. Ci si potrebbe sorprendere di quanto sia utile ottenere variabili altamente esplicative da operazioni di feature engineering senza un apparente scopo logico (che viene successivamente scoperto/interpretato).
- *Data Selection più approfondita*: Come già accennato, la selezione delle variabili più performanti è un processo iterativo. Non è nemmeno detto che esista una combinazione migliore in assoluto (in quanto potrebbe esserci una combinazione dipendente dalle caratteristiche dell'algoritmo impiegato). Con strumenti più potenti, che riescono ad esplorare in tempi utili porzioni dell'iperspazio più ampie rispetto a quanto fatto in questa sede, si potrebbero ampliare gli scenari di test.
- *Altri algoritmi*: Sono stati testati quattro diversi algoritmi. Tuttavia, ne esistono molti altri che potrebbero performare meglio di quanto analizzato in questa sede. Un algoritmo molto versatile e quotato ultimamente per via delle performance ottenute è il *XGBoost*.

- *Ottimizzazione dei parametri più approfondita*: Anche in questo caso, avendo a disposizione un hardware più potente, il numero di combinazioni di parametri esplorabili per ogni algoritmo potrebbe scoprire casisitche più performanti da quelle trovate in questa sede.
- *Algoritmi di Ensemble*: Sono stati infine proposti due semplici approcci di Ensemble, ma risulta evidente che l'individuazione della combinazione e strutture di algoritmi che possano meglio capire e stimare la LGD potrebbe essere potenzialmente un'altra (o più) Tesi.

Conclusioni

Come accennato precedentemente, l'impiego delle tecniche di machine learning non è immediato e tantomeno a costo zero. Nonostante questo, risulta evidente, anche da un semplice elaborato come questo, che l'utilizzo dei più basilari algoritmi di questo ramo della statistica comporta una netta riduzione dell'errore commesso nella stima di una importante variabile di rischio di credito come la Loss Given Default.

È stato inoltre dimostrato quanto sia importante il processo di selezione degli attributi e quanto possano migliorare le performance con il fine tuning dei modelli. Ancora una volta è stata confermata la difficoltà di ottenere errori statistici trascurabili nello stimare questa difficile variabile, nonostante la diversità delle logiche di costruzione degli algoritmi impiegati.

Eppure, si evince come si è capaci di ottenere errori di stima nettamente inferiori con *tutti* gli algoritmi impiegati rispetto ai risultati ottenuti con la regressione multivariata in uso preso il gruppo di Intesa Sanpaolo. È vero che il Machine learning comporta dei costi di implementazione, ma forse sarebbe il caso di aprire uno studio di costi/opportunità per capire se effettivamente questi costi non vengano totalmente assorbiti dai guadagni diretti ed indiretti che comporterebbe l'applicazione degli algoritmi trattati.

Sono in molti a credere che il futuro non possa prescindere dall'utilizzo sempre più frequente di sofisticati algoritmi di apprendimento automatico. L'automazione ha rivoluzionato la società moderna cambiandone totalmente le connotazioni. L'automazione intelligente potrebbe fare altrettanto e il sottoscritto fa parte del gruppo che crede in un domani fortemente legato all'utilizzo delle tecniche di machine learning in ogni micro-aspetto delle nostre vite quotidiane.

Appendice A

Tabella degli attributi ed elaborazioni

ID	Nome Attributo	Descrizione Attributo	Note sull'Attributo	Trattamento variabile
1	*cd_istituto_lgd*	Codice ABI (Associazione Bancaria Italiana), identifica la Banca di riferimento	Variabile categorica con 28 diversi valori	Trasformazione con il One Hot Encode
2	*cd_ndg_lgd_appo*	Codice identificativo del cliente. Trattasi di un codice fittizio per non violare politiche aziendali	Trattandosi potenzialmente di un progressivo, il contenuto informativo che apporta alla stima non è rilevante se si tiene già in considerazione le varie date che danno un'idea dell'andamento della LGD nel tempo. L'86% delle controparti hanno soltanto 1 rapporto, il 16% ne ha due e il restante 2% ha da 3 a 28 rapporti. Escludendo la variabile si perde dunque l'informazione sull'impatto in termini di LGD dell'avere più di un rapporto con la Banca. Tuttavia, la LGD viene stimata con una granularità di rapporto, non si considera l'esclusione una aggravante fondamentale.	Variabile Esclusa
3	*cd_rapplgd_appo*	Codice identificativo del rapporto con il cliente (mutuo, prestiti personali, ecc). Trattasi di un codice fittizio non violare politiche aziendali	Anche questa variabile sarebbe da escludere ma a differenza del codice controparte, dal codice rapporto si può estrapolare un'ulteriore informazione (dai primi caratteri), ovvero quella che sembra essere la "tipologia di rapporto". Diventa in questo modo una variabile categorica.	Sostituzione della variabile con un'altra dal dominio più ridotto (Da 29.000 valori a 6): *cd_rapplgd_appo_SEMPL*= MUT(Mutuo), ML (Medio-Lungo Termine), BR1, BR2, BR3, BR4 (Breve termine di tipologia 1,2,3 e 4). Quanto scritto fra parentesi sono proprie interpretazioni, basate anche sulla correlazione con la variabile CD_PERIMETRO_NEW, in quanto l'informazione non è stata esplicitata. È tuttavia opportuno notare che pur non conoscendo il significato funzionale di tale codifica, il fatto che ci sia una evidente *clusterizzazione* aiuta l'algoritmo (che a prescindere non conosce mai il significato funzionale di quello che gli diamo in alimentazione) nella previsione. Trasformazione con l'One Hot Encode

4	*dt_entrata*	Data di ingresso nello stato di Default		Le date vengono trattate nel seguente modo: 1) Si divide la data in tre diverse informazioni (attributi numerici): "giorno", "mese", "Anno" 2) Si cancella la variabile originaria per evitare ridondanze (ed anche perché alcuni algoritmi faticano a leggere il formato data)
5	*SAE*	Settore Economico della Controparte	Si tratta di una variabile categorica con 56 diversi valori.	Vista la possibilità di inglobare più valori sotto uno stesso cluster, seguendo la pubblicazione di Banca d'Italia, sono state create due ulteriori variabili che rappresentano una classificazione di più alto livello: SAE_cl1 (5 valori) SAE_cl2 (3 valori) I rispettivi domini possono trovarsi nelle apposite tabelle delle successive pagine. C'erano 3 valori missing sostituiti con il valore nullo 0 e classificato come Altro. Trasformazione con One Hot Encode
6	*RAE*	Ramo Economico della Controparte	Si tratta di una variabile categorica con 192 diversi valori.	Vista la possibiltà di inglobare più valori sotto uno stesso cluster, seguendo la pubblicazione di Banca d'Italia, sono state create due ulteriori variabili che rappresentano una classificazione di più alto livello: RAE_cl1 (23 valori) RAE_cl2 (4 valori) I rispettivi domini possono trovarsi nelle apposite tabelle delle successive pagine. C'erano 4179 valori missing sostituiti con il valore neutro 8 e classificato come "Valore Missing". Trasformazione con One Hot Encode
7	*DT_USCITA*	Data di uscita dallo stato di Default		Si veda ID 4 Sono stati trovati 4716 valori di default "31/12/9999" appartenenti, probabilmente alle posizioni stralciate, ma ancora in essere, tali per cui IW = 1 (infatti 9999 si riscontra solo per quelle posizioni IW = 1 e viceversa). Sostituire con il valore medio della colonna non aveva senso e pertanto si è ipotizzata una data di "chiusura" default pari alla data di entrata in sofferenza più il valore medio della permanenza in sofferenza (circa 5 anni)

8	*DT_ENTRATA_SOFF*	Data di ingresso in stato Sofferenza		Si veda ID 4 In 5 casi è stata riscontrata una data di entrata in sofferenza precedente, anche se di massimo un paio di mesi, all'entrata in default. Essendo lo stato di insolvenza un sottoinsieme dello stato di default questa evidenza sembra un'anomalia. La data di entrata in sofferenza è stata posta pari alla data di entrata in default.
9	*YEAR_OUT*	Anno uscita dal default: year(dt_uscita)		Visto la divisione fatta su DT_USCITA, descritta nel ID 4, questa variabile diventa ridondante (di fatto viene ricavata da ID 7)
10	*DUMMY_PER*	Presenza/assenza garanzia personale/fidejussoria	Variabile Booleana	
11	*DUMMY_IMM_FINALE*	Presenza/assenza garanzia ipotecaria	Variabile Booleana	
12	*CD_PERIMETRO_NEW*	Macro aggregato forma tecnica (BT: breve termine - MLT IPO medio lungo termine ipotecario - MLT NON IPO: medio lungo termine non ipotecario)		Variabile esclusa in quanto le ID 16, 17 e 18 rappresentano già le dummies ottenibili con One Hot Encode.
13	*IW*	Posizioni ancora in essere ma considerate come se fossero chiuse (con stralcio dell'esposizione residua). Dominio: SI/NO	Ci sono soltanto 4716 "SI". Coincidono con i valori per cui la data di uscita da default è "9999".	
14	*PROC_CNCS_CALC3*	Procedura concorsuale in essere alla data di calcolo: FALL: fallimento o altre procedure giudiziali / ALTRO (altre procedure non giudiziali)		Variabile esclusa in quanto le ID 19 e 20 rappresentano già le dummies ottenibili con One Hot Encode.
15	*marea5*	Area geografica. Dominio: SUD ISOLE /NORD CENTRO		Variabile esclusa in quanto le ID 23 e 24 rappresentano già le dummies ottenibili con One Hot Encode.
16	*D_PER_BT*	Dummy variabile CD_PERIMETRO NEW - BT	Variabile Booleana	
17	*D_PER_MLT_IPO*	Dummy variabile CD_PERIMETRO NEW - MLT IPO	Variabile Booleana	
18	*D_PER_MLT_NO_IP*	Dummy variabile CD_PERIMETRO NEW - MLT NON IPO	Variabile Booleana	
19	*D_PROC3_ALTRO*	Dummy variabile PROC_CNCS_CALC3 (ALTRO)	Variabile Booleana	
20	*D_PROC3_FALL*	Dummy variabile PROC_CNCS_CALC3 (FALL)	Variabile Booleana	

21	*D_SI_GAR_PER*	Dummy variabile presenza garanzia personale		Si tratta di una variabile Booleana complementare a quella successiva e totalmente uguale alla ID 10. Sia questa variabile che la successiva vengono escluse in quanto non aggiungono informazioni ulteriori.
22	*D_NO_GAR_PER*	Dummy variabile assenza garanzia personale		Si veda variabile precedente
23	*D_GEO_NCE*	Dummy variabile area geo (NORD)	Variabile Booleana	
24	*D_GEO_SUD*	Dummy variabile area geo (SUD)	Variabile Booleana	Essendo il complemento della precedente, viene esclusa.
25	*lgd_perc*	LGD osservata	La variabile dovrebbe avere un dominio compreso tra 0 e 1. Ci sono tuttavia valori superiori a 1 (meno dello 0,6%), il che potrebbe indicare che si ha avuto una perdita superiore all'esposizione vantata verso il debitore. Si è deciso di non forzare questi valori a 1.	
26	*lgd_stim*	LGD Stimata	Si tratta della LGD stimata dal modello di regressione multivariata di Intesa Sanpaolo	

Tabella nuovi attributi

ID	Intervento	Nome Attributo	Descrizione Attributo
1	Binning	cd_rapplgd_appo_SEMPL	Come descritto nella tabella precedente, si tratta di un attributo prodotto del Binning su cd_rapplgd_appo
		SAE_cl1	Come descritto nella tabella precedente, si tratta di un attributo prodotto del Binning su SAE
		SAE_cl2	Come descritto nella tabella precedente, si tratta di un attributo prodotto del Binning su SAE
		RAE_cl1	Come descritto nella tabella precedente, si tratta di un attributo prodotto del Binning su RAE
		RAE_cl2	Come descritto nella tabella precedente, si tratta di un attributo prodotto del Binning su RAE
2	Separazione Variabili	*dt_entrata_xx*	con _xx= day, month, year. Quindi tre ulteriori attributi
		Data di uscita dal default_xx	con _xx= day, month, year. Quindi tre ulteriori attributi
		Data di ingresso in sofferenza_xx	con _xx= day, month, year. Quindi tre ulteriori attributi
3	Combinazione variabili	Delta_entr_sof	Delta_entr_sof = DT_ENTRATA_SOFF - dt_entrata Si tratta del periodo (in giorni) che trascorre tra l'entrata in default e l'entrata in sofferenza
		Delta_entrsof_uscsof	Delta_entrsof_uscsof = DT_USCITA - DT_ENTRATA_SOFF Si tratta del periodo (in giorni) che trascorre tra l'entrata in sofferenza e l'uscita dal default
		Delta_entr_usct	Somma dei due precedenti
		Rpp_entr_sof	Delta_entrsof_uscsof/Delta_entr_usct
		Rpp_entrsof_uscsof	Il complemento del precedente: 1- Rpp_entr_sof
4	One Hot Encode	*cd_istituto_lgd_xx*	Dove _xx assume tutti e 28 i valori dell'attributo originale cd_istituto_lgd, quindi altrettante colonne.
		cd_rapplgd_appo_SEMPL_xx	Dove _xx assume tutti e 6 i valori dell'attributo originale cd_rapplgd_appo_SEMPL, quindi altrettante colonne.

		SAE_xx	Dove _xx assume tutti e 56 i valori dell'attributo originale SAE, quindi altrettante colonne.
		SAE_cl1_xx	Dove _xx assume tutti e 5 i valori dell'attributo originale SAE_cl1, quindi altrettante colonne.
		SAE_cl2_xx	Dove _xx assume tutti e 3 i valori dell'attributo originale SAE_cl2, quindi altrettante colonne.
		RAE_xx	Dove _xx assume tutti e 192 i valori dell'attributo originale RAE, quindi altrettante colonne.
		RAE_cl1_xx	Dove _xx assume tutti e 23 i valori dell'attributo originale RAE_cl1, quindi altrettante colonne.
		RAE_cl2_xx	Dove _xx assume tutti e 4 i valori dell'attributo originale RAE_cl2, quindi altrettante colonne.

Tabella dominio SAE

SAE	cl1	cl2
775	famiglie consumatrici	Famiglia
774	famiglie consumatrici	Famiglia
772	famiglie produttrici	Famiglia
768	famiglie produttrici	Famiglia
759	societa non finanziarie	Societa
758	societa non finanziarie	Societa
757	societa non finanziarie	Societa
748	societa finanziarie	Societa
621	famiglie produttrici	Famiglia
620	famiglie produttrici	Famiglia
615	famiglie produttrici	Famiglia
614	famiglie produttrici	Famiglia
600	famiglie consumatrici	Famiglia
552	Altro	Altro
551	Altro	Altro
550	Altro	Altro
501	Altro	Altro
500	Altro	Altro
492	societa non finanziarie	Societa
491	societa non finanziarie	Societa
490	societa non finanziarie	Societa
482	societa non finanziarie	Societa
481	societa non finanziarie	Societa
480	societa non finanziarie	Societa
473	societa non finanziarie	Societa
472	societa non finanziarie	Societa
471	societa non finanziarie	Societa
470	societa non finanziarie	Societa
450	societa non finanziarie	Societa
442	societa non finanziarie	Societa
441	societa non finanziarie	Societa
440	societa non finanziarie	Societa

431	societa non finanziarie	Societa
430	societa non finanziarie	Societa
420	Altro	Altro
352	societa finanziarie	Societa
350	societa finanziarie	Societa
346	societa finanziarie	Societa
344	societa finanziarie	Societa
340	societa finanziarie	Societa
294	societa finanziarie	Societa
284	societa finanziarie	Societa
283	societa finanziarie	Societa
280	societa finanziarie	Societa
276	societa finanziarie	Societa
273	societa finanziarie	Societa
268	societa finanziarie	Societa
263	Altro	Altro
259	societa finanziarie	Societa
258	societa finanziarie	Societa
257	societa finanziarie	Societa
256	societa finanziarie	Societa
220	societa finanziarie	Societa
201	societa finanziarie	Societa
177	Altro	Altro

Tabella dominio RAE

RAE	RAE_cl2	RAE_cl3
0	Holding, finanziarie ed altro	Terziario
8	Valore missing	Valore missing
11	Agricoltura	Primario
12	Alimentare	Secondario
13	Alimentare	Secondario
14	Agricoltura	Primario
19	Agricoltura	Primario
20	Agricoltura	Primario
30	Agricoltura	Primario
111	Energia ed estrazione	Secondario
112	Energia ed estrazione	Secondario
120	Energia ed estrazione	Secondario
130	Energia ed estrazione	Secondario
140	Energia ed estrazione	Secondario
151	Energia ed estrazione	Secondario
152	Energia ed estrazione	Secondario
161	Utility	Terziario

162	Utility	Terziario
163	Utility	Terziario
170	Utility	Terziario
211	Energia ed estrazione	Secondario
212	Energia ed estrazione	Secondario
221	Metallurgia e prodotti in metallo	Secondario
222	Metallurgia e prodotti in metallo	Secondario
223	Metallurgia e prodotti in metallo	Secondario
224	Metallurgia e prodotti in metallo	Secondario
231	Energia ed estrazione	Secondario
232	Energia ed estrazione	Secondario
233	Energia ed estrazione	Secondario
239	Energia ed estrazione	Secondario
241	Costruzioni e materiali per costruzioni	Secondario
242	Costruzioni e materiali per costruzioni	Secondario
243	Costruzioni e materiali per costruzioni	Secondario
244	Costruzioni e materiali per costruzioni	Secondario
245	Costruzioni e materiali per costruzioni	Secondario
246	Costruzioni e materiali per costruzioni	Secondario
247	Costruzioni e materiali per costruzioni	Secondario
248	Costruzioni e materiali per costruzioni	Secondario
252	Chimica di base e intermedi	Secondario
253	Chimica di base e intermedi	Secondario
255	Chimica di base e intermedi	Secondario
256	Chimica di base e intermedi	Secondario
257	Farmaceutica	Secondario
258	Largo consumo	Secondario
259	Largo consumo	Secondario
260	Chimica di base e intermedi	Secondario
311	Metallurgia e prodotti in metallo	Secondario
312	Metallurgia e prodotti in metallo	Secondario
313	Metallurgia e prodotti in metallo	Secondario
314	Metallurgia e prodotti in metallo	Secondario
315	Metallurgia e prodotti in metallo	Secondario
316	Metallurgia e prodotti in metallo	Secondario
321	Meccanica	Secondario
322	Meccanica	Secondario
323	Meccanica	Secondario
324	Meccanica	Secondario
325	Meccanica	Secondario
326	Meccanica	Secondario
327	Meccanica	Secondario
328	Meccanica	Secondario

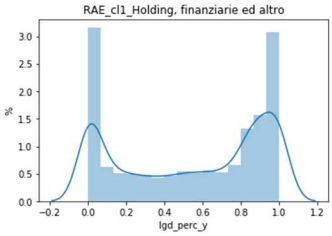
330	Elettrotecnica ed elettronica	Secondario
341	Elettrotecnica ed elettronica	Secondario
342	Elettrotecnica ed elettronica	Secondario
343	Elettrotecnica ed elettronica	Secondario
344	Elettrotecnica ed elettronica	Secondario
345	Elettrotecnica ed elettronica	Secondario
346	Elettrodomestici	Secondario
347	Elettrotecnica ed elettronica	Secondario
351	Mezzi di trasporto	Terziario
352	Mezzi di trasporto	Terziario
353	Mezzi di trasporto	Terziario
361	Mezzi di trasporto	Terziario
362	Mezzi di trasporto	Terziario
363	Mezzi di trasporto	Terziario
364	Mezzi di trasporto	Terziario
365	Mezzi di trasporto	Terziario
371	Elettrotecnica ed elettronica	Secondario
372	Elettrotecnica ed elettronica	Secondario
373	Elettrotecnica ed elettronica	Secondario
374	Elettrotecnica ed elettronica	Secondario
411	Alimentare	Secondario
412	Alimentare	Secondario
413	Alimentare	Secondario
414	Alimentare	Secondario
415	Alimentare	Secondario
416	Alimentare	Secondario
417	Alimentare	Secondario
418	Alimentare	Secondario
419	Alimentare	Secondario
420	Alimentare	Secondario
421	Alimentare	Secondario
422	Alimentare	Secondario
423	Alimentare	Secondario
424	Alimentare	Secondario
425	Alimentare	Secondario
426	Alimentare	Secondario
427	Alimentare	Secondario
428	Alimentare	Secondario
429	Alimentare	Secondario
431	Sistema moda	Secondario
432	Sistema moda	Secondario
436	Sistema moda	Secondario
438	Sistema moda	Secondario

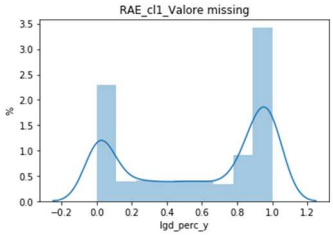
439	Sistema moda	Secondario
441	Sistema moda	Secondario
442	Sistema moda	Secondario
451	Sistema moda	Secondario
453	Sistema moda	Secondario
455	Sistema moda	Secondario
456	Sistema moda	Secondario
461	Intermedi per l'industria: beni vari	Secondario
462	Intermedi per l'industria: beni vari	Secondario
463	Intermedi per l'industria: beni vari	Secondario
464	Intermedi per l'industria: beni vari	Secondario
465	Intermedi per l'industria: beni vari	Secondario
466	Altri beni di consumo	Secondario
467	Mobili	Secondario
471	Intermedi per l'industria: beni vari	Secondario
472	Intermedi per l'industria: beni vari	Secondario
473	Editoria e stampa	Terziario
474	Editoria e stampa	Terziario
481	Intermedi per l'industria: beni vari	Secondario
482	Intermedi per l'industria: beni vari	Secondario
483	Intermedi per l'industria: beni vari	Secondario
491	Altri beni di consumo	Secondario
492	Altri beni di consumo	Secondario
493	Editoria e stampa	Terziario
494	Altri beni di consumo	Secondario
495	Altri beni di consumo	Secondario
505	Costruzioni e materiali per costruzioni	Secondario
506	Costruzioni e materiali per costruzioni	Secondario
507	Costruzioni e materiali per costruzioni	Secondario
509	Costruzioni e materiali per costruzioni	Secondario
611	Distribuzione	Terziario
612	Distribuzione	Terziario
613	Distribuzione	Terziario
614	Distribuzione	Terziario
615	Distribuzione	Terziario
616	Distribuzione	Terziario
617	Distribuzione	Terziario
618	Distribuzione	Terziario
619	Distribuzione	Terziario
620	Servizi	Terziario
630	Distribuzione	Terziario
641	Distribuzione	Terziario
642	Distribuzione	Terziario

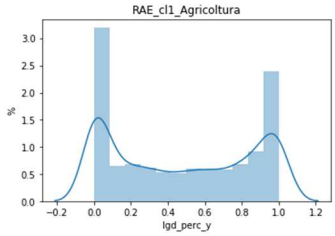
643	Distribuzione	Terziario
644	Distribuzione	Terziario
645	Distribuzione	Terziario
646	Distribuzione	Terziario
647	Distribuzione	Terziario
648	Distribuzione	Terziario
649	Distribuzione	Terziario
651	Distribuzione	Terziario
652	Distribuzione	Terziario
653	Distribuzione	Terziario
654	Distribuzione	Terziario
655	Distribuzione	Terziario
656	Distribuzione	Terziario
660	Servizi	Terziario
671	Distribuzione	Terziario
672	Servizi	Terziario
710	Trasporti	Terziario
721	Trasporti	Terziario
722	Trasporti	Terziario
723	Trasporti	Terziario
724	Trasporti	Terziario
725	Trasporti	Terziario
730	Trasporti	Terziario
741	Trasporti	Terziario
742	Trasporti	Terziario
750	Trasporti	Terziario
761	Trasporti	Terziario
762	Trasporti	Terziario
763	Trasporti	Terziario
764	Trasporti	Terziario
771	Servizi	Terziario
772	Trasporti	Terziario
773	Trasporti	Terziario
790	Utility	Terziario
810	Holding, finanziarie ed altro	Terziario
820	Holding, finanziarie ed altro	Terziario
830	Holding, finanziarie ed altro	Terziario
840	Servizi	Terziario
850	Servizi	Terziario
910	Holding, finanziarie ed altro	Terziario
920	Utility	Terziario
930	Servizi	Terziario
940	Servizi	Terziario

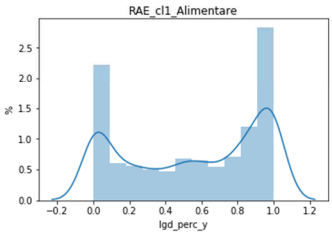
950	Servizi	Terziario
960	Servizi	Terziario
970	Servizi	Terziario
981	Servizi	Terziario
982	Servizi	Terziario
983	Servizi	Terziario
984	Servizi	Terziario
999	Holding, finanziarie ed altro	Terziario

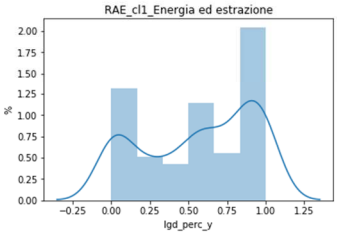
Grafico distribuzione della LGD per RAE

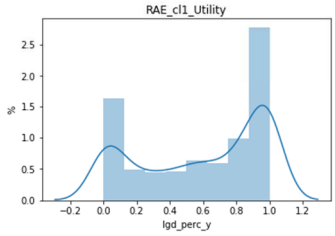

RAE_cl1_Holding, finanziarie ed altro
%
lgd_perc_y

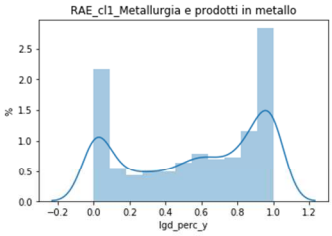

RAE_cl1_Valore missing
%
lgd_perc_y

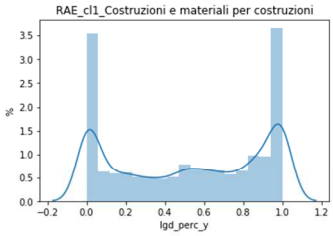

RAE_cl1_Agricoltura
%
lgd_perc_y

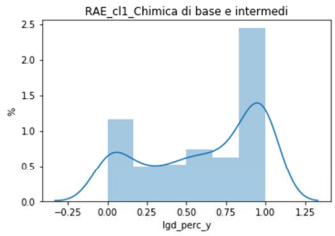

RAE_cl1_Alimentare
%
lgd_perc_y

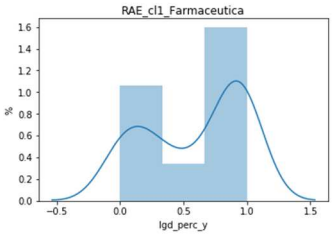

RAE_cl1_Energia ed estrazione
%
lgd_perc_y

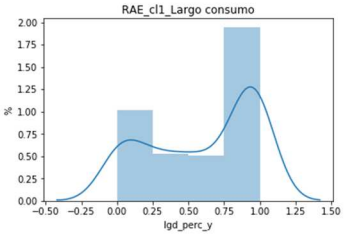

RAE_cl1_Utility
%
lgd_perc_y

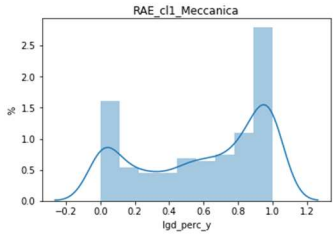

RAE_cl1_Metallurgia e prodotti in metallo
%
lgd_perc_y

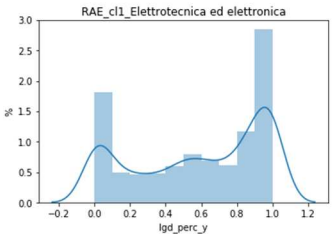

RAE_cl1_Costruzioni e materiali per costruzioni
%
lgd_perc_y

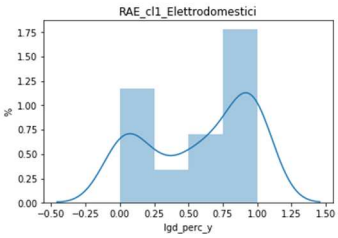

RAE_cl1_Chimica di base e intermedi
%
lgd_perc_y

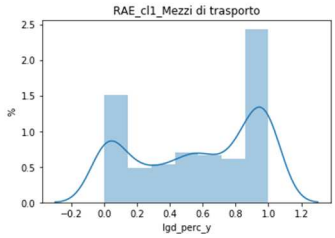

RAE_cl1_Farmaceutica
%
lgd_perc_y

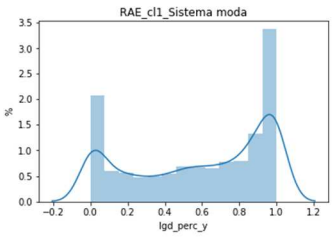

RAE_cl1_Largo consumo
%
lgd_perc_y

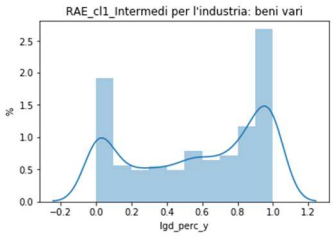

RAE_cl1_Meccanica
%
lgd_perc_y

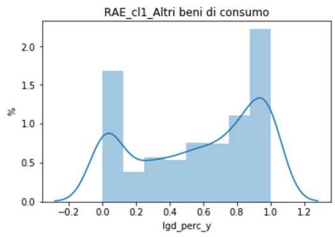

RAE_cl1_Elettrotecnica ed elettronica
%
lgd_perc_y

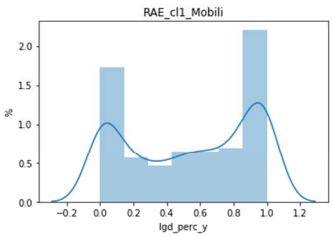

RAE_cl1_Elettrodomestici
%
lgd_perc_y

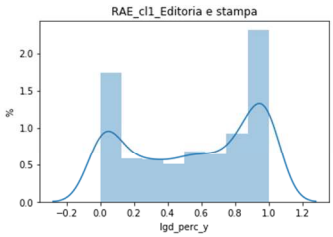

RAE_cl1_Mezzi di trasporto
%
lgd_perc_y

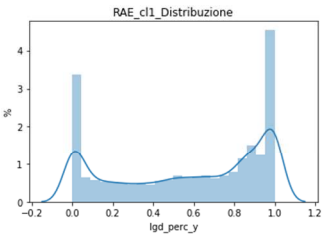

RAE_cl1_Sistema moda
%
lgd_perc_y

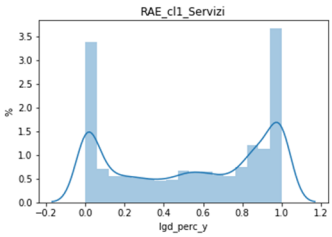

RAE_cl1_Intermedi per l'industria: beni vari
%
lgd_perc_y

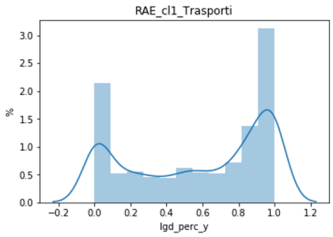

RAE_cl1_Altri beni di consumo
%
lgd_perc_y


RAE_cl1_Mobili
%
lgd_perc_y


RAE_cl1_Editoria e stampa
%
lgd_perc_y


RAE_cl1_Distribuzione
%
lgd_perc_y


RAE_cl1_Servizi
%
lgd_perc_y


RAE_cl1_Trasporti
%
lgd_perc_y

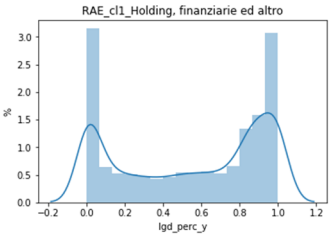

RAE_cl1_Holding, finanziarie ed altro
%
lgd_perc_y

Tabella Dataset 56

Dataset 56	
'cd_istituto_lgd_1010'	'RAE_cl1_Elettrotecnica ed elettronica'
'cd_istituto_lgd_6160'	'RAE_cl1_Servizi'
'cd_istituto_lgd_6225'	'RAE_cl1_Trasporti'
'cd_istituto_lgd_99999'	'RAE_cl2_Secondario'
'cd_rapplgd_appo_SEMPL_BR2'	'SAE_430'
'cd_rapplgd_appo_SEMPL_MUT'	'SAE_492'
'D_GEO_NCE'	'SAE_614'
'D_PER_BT'	'SAE_615'
'D_PER_MLT_NO_IP'	'SAE_cl1_societa non finanziarie'
'Delta_entr_sof'	cd_istituto_lgd_1025'
'Delta_entrsof_uscsof'	cd_istituto_lgd_6385'
'dt_entrata_day'	cd_rapplgd_appo_SEMPL_BR1'
'dt_entrata_month'	cd_rapplgd_appo_SEMPL_BR3'
'DT_ENTRATA_SOFF_day'	D_PROC3_ALTRO'
'dt_entrata_year'	DT_ENTRATA_SOFF_month'
'DT_USCITA_day'	DT_USCITA_year'
'DT_USCITA_month'	'DUMMY_PER'
'DUMMY_IMM_FINALE'	RAE_660'
'IW'	RAE_8'
'RAE_11'	RAE_cl1_Alimentare'
'RAE_505'	RAE_cl1_Distribuzione'
'RAE_630'	RAE_cl1_Intermediperl'industria:beni vari'
'RAE_645'	RAE_cl1_Metallurgiaeprodottiinmetallo'
'RAE_648'	RAE_cl1_Sistemamoda'
'RAE_830'	RAE_cl2_Terziario'
'RAE_984'	Rpp_entr_sof'
'RAE_cl1_Agricoltura'	SAE_482'
'RAE_cl1_Costruzioni e materiali per costruzioni'	SAE_cl1_famiglieproduttrici'

Appendice B

Codice Python di sintesi

```
#Librerie

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from scipy import stats

import seaborn as sns

from sklearn.pipeline import make_pipeline

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler

from sklearn.preprocessing import Normalizer

#Estimators

from sklearn.tree import DecisionTreeRegressor

from sklearn.ensemble import RandomForestRegressor

from sklearn.svm import LinearSVR

from sklearn.svm import SVR

from sklearn.svm import NuSVR

from sklearn.neural_network import MLPRegressor

from sklearn.feature_selection import VarianceThreshold

from sklearn.linear_model import Lasso

from sklearn.feature_selection import SelectKBest

from sklearn.feature_selection import f_regression

from sklearn.feature_selection import mutual_info_regression

from sklearn.feature_selection import RFE

from sklearn.feature_selection import RFECV

from sklearn.decomposition import PCA

from sklearn.model_selection import GridSearchCV

from sklearn.model_selection import RandomizedSearchCV
```

```
from sklearn import metrics

from sklearn.metrics import r2_score

from sklearn.metrics import explained_variance_score

import statistics

#Le colonne SAE_cl1, SAE_cl2, RAE_cl1, RAE_cl2, cd_rapplgd_appo_SEMPL sono state aggiunte direttamente sul file csv.

#Caricamento dataset

dt0 = pd.read_csv('REGR300.csv', sep = ';')

#Selezione Colonne utili

colonne_utili = ['cd_istituto_lgd','cd_rapplgd_appo_SEMPL', 'dt_entrata',

    'SAE','SAE_cl1', 'SAE_cl2','RAE','RAE_cl1', 'RAE_cl2','DT_USCITA', 'DT_ENTRATA_SOFF', 'DUMMY_PER',

    'DUMMY_IMM_FINALE','IW','lgd_perc','D_PER_BT',

    'D_PER_MLT_IPO', 'D_PER_MLT_NO_IP', 'D_PROC3_ALTRO', 'D_PROC3_FALL','D_GEO_NCE', 'D_GEO_SUD','lgd_stim']

dt1 = dt0[colonne_utili]

#Trattamento delle date

dt00 = dt1.copy()

#Trasformazione campi interessati in datatime

dt00.dt_entrata = pd.to_datetime(dt00.dt_entrata, format="%d/%m/%Y")

dt00.DT_USCITA = pd.to_datetime(dt00.DT_USCITA, format="%d/%m/%Y")

dt00.DT_ENTRATA_SOFF = pd.to_datetime(dt00.DT_ENTRATA_SOFF, format="%d/%m/%Y")

#Creazione colonne di differenze data ed eventuali elaborazioni

dt00['Delta_entr_sof'] = (dt00['DT_ENTRATA_SOFF'] - dt00['dt_entrata']).dt.days #periodo in stato di default prima della sofferenza

dt00['Delta_entrsof_uscsof'] = (dt00['DT_USCITA'] - dt00['DT_ENTRATA_SOFF']).dt.days #periodo in sofferenza

dt00['Delta_entr_usct'] = (dt00['DT_USCITA'] - dt00['dt_entrata']).dt.days #totale periodo di default

dt00['Rpp_entr_sof'] = dt00['Delta_entr_sof']/dt00['Delta_entr_usct']

dt00['Rpp_entrsof_uscsof'] = dt00['Delta_entrsof_uscsof']/dt00['Delta_entr_usct']
```

#Separazione singole colonne in 3 colonne diverse

dt00['dt_entrata_year'],dt00['dt_entrata_month'],dt00['dt_entrata_day'] = dt00.dt_entrata.dt.year, dt00.dt_entrata.dt.month, dt00.dt_entrata.dt.day

dt00['DT_USCITA_year'],dt00['DT_USCITA_month'],dt00['DT_USCITA_day'] = dt00.DT_USCITA.dt.year, dt00.DT_USCITA.dt.month, dt00.DT_USCITA.dt.day

dt00['DT_ENTRATA_SOFF_year'],dt00['DT_ENTRATA_SOFF_month'],dt00['DT_ENTRATA_SOFF_day'] = dt00.DT_ENTRATA_SOFF.dt.year, dt00.DT_ENTRATA_SOFF.dt.month, dt00.DT_ENTRATA_SOFF.dt.day

#Eliminazione delle colonne date che non sono utili per gli algoritmi

dt2 = dt00.drop(["dt_entrata", "DT_USCITA", "DT_ENTRATA_SOFF"], axis=1)

#Gestione Outliers sulle date e derivati

dt2.loc[dt['Delta_entr_sof'] < 0, 'Delta_entr_sof'] = 0

dt2.loc[dt['DT_USCITA_year'] == 9999, 'Delta_entrsof_uscsof'] = 1825

dt2.loc[dt['DT_USCITA_year'] == 9999, 'Delta_entr_usct'] = dt2.loc[dt['DT_USCITA_year'] == 9999, 'Delta_entr_sof']+1825

dt2.loc[dt['DT_USCITA_year'] == 9999, 'DT_USCITA_year'] = dt2.loc[dt['DT_USCITA_year'] == 9999, 'DT_ENTRATA_SOFF_year']+5

dt2['Rpp_entr_sof'] = dt2['Delta_entr_sof']/dt2['Delta_entr_usct']

dt2['Rpp_entrsof_uscsof'] = dt2['Delta_entrsof_uscsof']/dt2['Delta_entr_usct']

#One Hot Encode sulle variabili categoriche

categoriche = ['cd_istituto_lgd','cd_rapplgd_appo_SEMPL','SAE', 'SAE_cl1', 'SAE_cl2','RAE', 'RAE_cl1', 'RAE_cl2']

dt3 = pd.get_dummies(dt2,columns = categoriche)

 # GENERAZIONE DEI DATASET UTILI

#Dataset Iniziale - serve come dataset di input nella feature selection

xselec = dt[[n for n in dt.columns if n not in ('lgd_perc_y','lgd_stim')]].copy()

#Colonne da non considerare ricavate dalle analisi di Correlazione di Pearson e Distribuzione valori

Analisi_distrib99 = ('RAE_111', 'RAE_112', 'RAE_12', 'RAE_120', 'RAE_13', 'RAE_130', 'RAE_14', 'RAE_140', 'RAE_151', 'RAE_161', 'RAE_162', 'RAE_163', 'RAE_170', 'RAE_19', 'RAE_20', 'RAE_211', 'RAE_212', 'RAE_221', 'RAE_222', 'RAE_223', 'RAE_224', 'RAE_231', 'RAE_232', 'RAE_233', 'RAE_239', 'RAE_241', 'RAE_242', 'RAE_243', 'RAE_244', 'RAE_245', 'RAE_246', 'RAE_247', 'RAE_248', 'RAE_252', 'RAE_253', 'RAE_255', 'RAE_256', 'RAE_257', 'RAE_258', 'RAE_259', 'RAE_260', 'RAE_30', 'RAE_311', 'RAE_312', 'RAE_313', 'RAE_314', 'RAE_315', 'RAE_316', 'RAE_321', 'RAE_322', 'RAE_323', 'RAE_324', 'RAE_325', 'RAE_326', 'RAE_327', 'RAE_328', 'RAE_330', 'RAE_341', 'RAE_342', 'RAE_343', 'RAE_344', 'RAE_345', 'RAE_346', 'RAE_347', 'RAE_351', 'RAE_352', 'RAE_353', 'RAE_361', 'RAE_362', 'RAE_363', 'RAE_364', 'RAE_365', 'RAE_371', 'RAE_372', 'RAE_373', 'RAE_374', 'RAE_411', 'RAE_412', 'RAE_413', 'RAE_414', 'RAE_415', 'RAE_416', 'RAE_417', 'RAE_418', 'RAE_419', 'RAE_420', 'RAE_421', 'RAE_422', 'RAE_423', 'RAE_424', 'RAE_425', 'RAE_426', 'RAE_427', 'RAE_428', 'RAE_429', 'RAE_431', 'RAE_432', 'RAE_436',

'RAE_438', 'RAE_439', 'RAE_441', 'RAE_442', 'RAE_451', 'RAE_455', 'RAE_456', 'RAE_461', 'RAE_462', 'RAE_463', 'RAE_464', 'RAE_465', 'RAE_466', 'RAE_471', 'RAE_472', 'RAE_473', 'RAE_474', 'RAE_481', 'RAE_482', 'RAE_483', 'RAE_491', 'RAE_492', 'RAE_493', 'RAE_494', 'RAE_495', 'RAE_506', 'RAE_507', 'RAE_509', 'RAE_611', 'RAE_612', 'RAE_613', 'RAE_614', 'RAE_615', 'RAE_616', 'RAE_618', 'RAE_619', 'RAE_620', 'RAE_643', 'RAE_644', 'RAE_646', 'RAE_647', 'RAE_649', 'RAE_652', 'RAE_653', 'RAE_655', 'RAE_656', 'RAE_671', 'RAE_672', 'RAE_710', 'RAE_721', 'RAE_722', 'RAE_724', 'RAE_725', 'RAE_730', 'RAE_741', 'RAE_742', 'RAE_750', 'RAE_761', 'RAE_762', 'RAE_763', 'RAE_764', 'RAE_771', 'RAE_772', 'RAE_773', 'RAE_790', 'RAE_840', 'RAE_850', 'RAE_920', 'RAE_930', 'RAE_940', 'RAE_950', 'RAE_960', 'RAE_981', 'RAE_982', 'RAE_983', 'RAE_999', 'RAE_cl1_Altri beni di consumo', 'RAE_cl1_Chimica di base e intermedi', 'RAE_cl1_Elettrodomestici', 'RAE_cl1_Energia ed estrazione', 'RAE_cl1_Farmaceutica', 'RAE_cl1_Largo consumo', 'RAE_cl1_Mezzi di trasporto', 'SAE_0', 'SAE_177', 'SAE_201', 'SAE_220', 'SAE_256', 'SAE_257', 'SAE_258', 'SAE_259', 'SAE_263', 'SAE_268', 'SAE_273', 'SAE_276', 'SAE_280', 'SAE_283', 'SAE_284', 'SAE_294', 'SAE_340', 'SAE_344', 'SAE_346', 'SAE_350', 'SAE_352', 'SAE_420', 'SAE_431', 'SAE_440', 'SAE_441', 'SAE_442', 'SAE_450', 'SAE_470', 'SAE_471', 'SAE_472', 'SAE_473', 'SAE_480', 'SAE_481', 'SAE_491', 'SAE_500', 'SAE_501', 'SAE_550', 'SAE_551', 'SAE_552', 'SAE_620', 'SAE_748', 'SAE_757', 'SAE_758', 'SAE_759', 'SAE_768', 'SAE_772', 'SAE_774', 'SAE_775', 'SAE_cl1_Altro', 'SAE_cl1_Valore missing', 'SAE_cl1_societa finanziarie', 'SAE_cl2_Altro', 'SAE_cl2_Valore missing', 'cd_istituto_lgd_10637', 'cd_istituto_lgd_3240', 'cd_istituto_lgd_32896', 'cd_istituto_lgd_3309', 'cd_istituto_lgd_3359', 'cd_istituto_lgd_6030', 'cd_istituto_lgd_6065', 'cd_istituto_lgd_6080', 'cd_istituto_lgd_6125', 'cd_istituto_lgd_6130', 'cd_istituto_lgd_6165', 'cd_istituto_lgd_6220', 'cd_istituto_lgd_6280', 'cd_istituto_lgd_6315', 'cd_istituto_lgd_6345', 'cd_istituto_lgd_6380', 'cd_istituto_lgd_6930')

Analisi_distrib999 = ('RAE_111', 'RAE_112', 'RAE_120', 'RAE_13', 'RAE_130', 'RAE_140', 'RAE_151', 'RAE_161', 'RAE_162', 'RAE_163', 'RAE_170', 'RAE_19', 'RAE_20', 'RAE_211', 'RAE_212', 'RAE_221', 'RAE_222', 'RAE_224', 'RAE_232', 'RAE_233', 'RAE_239', 'RAE_241', 'RAE_244', 'RAE_246', 'RAE_252', 'RAE_253', 'RAE_256', 'RAE_257', 'RAE_259', 'RAE_260', 'RAE_311', 'RAE_315', 'RAE_323', 'RAE_326', 'RAE_327', 'RAE_362', 'RAE_363', 'RAE_364', 'RAE_365', 'RAE_371', 'RAE_374', 'RAE_411', 'RAE_414', 'RAE_415', 'RAE_416', 'RAE_418', 'RAE_420', 'RAE_422', 'RAE_424', 'RAE_425', 'RAE_426', 'RAE_427', 'RAE_429', 'RAE_438', 'RAE_456', 'RAE_462', 'RAE_464', 'RAE_466', 'RAE_471', 'RAE_481', 'RAE_482', 'RAE_492', 'RAE_643', 'RAE_655', 'RAE_710', 'RAE_721', 'RAE_724', 'RAE_725', 'RAE_730', 'RAE_741', 'RAE_742', 'RAE_750', 'RAE_762', 'RAE_763', 'RAE_764', 'RAE_773', 'RAE_999', 'RAE_cl1_Farmaceutica', 'SAE_0', 'SAE_177', 'SAE_201', 'SAE_220', 'SAE_256', 'SAE_257', 'SAE_258', 'SAE_259', 'SAE_263', 'SAE_268', 'SAE_273', 'SAE_276', 'SAE_283', 'SAE_284', 'SAE_294', 'SAE_340', 'SAE_344', 'SAE_346', 'SAE_350', 'SAE_352', 'SAE_420', 'SAE_431', 'SAE_440', 'SAE_441', 'SAE_470', 'SAE_471', 'SAE_472', 'SAE_473', 'SAE_500', 'SAE_550', 'SAE_551', 'SAE_552', 'SAE_748', 'SAE_757', 'SAE_758', 'SAE_759', 'SAE_768', 'SAE_772', 'SAE_774', 'SAE_775', 'SAE_cl1_Valore missing', 'SAE_cl2_Valore missing', 'cd_istituto_lgd_3309', 'cd_istituto_lgd_3359', 'cd_istituto_lgd_6065', 'cd_istituto_lgd_6080', 'cd_istituto_lgd_6125', 'cd_istituto_lgd_6165', 'cd_istituto_lgd_6280', 'cd_istituto_lgd_6380', 'cd_istituto_lgd_6930')

Pearson95 = ('DT_ENTRATA_SOFF_year', 'D_GEO_SUD', 'D_PER_MLT_IPO', 'D_PROC3_FALL', 'Delta_entr_usct', 'RAE_cl1_Elettrodomestici', 'RAE_cl1_Farmaceutica', 'RAE_cl1_Holding, finanziarie ed altro', 'RAE_cl1_Mobili', 'RAE_cl1_Valore missing', 'RAE_cl2_Primario', 'RAE_cl2_Valore missing', 'Rpp_entrsof_uscsof', 'SAE_cl1_Valore missing', 'SAE_cl1_famiglie consumatrici', 'SAE_cl2_Altro', 'SAE_cl2_Famiglia', 'SAE_cl2_Societa', 'SAE_cl2_Valore missing')

Pearson50 = ('DT_ENTRATA_SOFF_year', 'DT_USCITA_year', 'D_GEO_SUD', 'D_PER_BT', 'D_PER_MLT_IPO', 'D_PER_MLT_NO_IP', 'D_PROC3_FALL', 'Delta_entr_usct', 'Delta_entrsof_uscsof', 'RAE_8', 'RAE_cl1_Agricoltura', 'RAE_cl1_Alimentare', 'RAE_cl1_Altri beni di consumo', 'RAE_cl1_Chimica di

base e intermedi', 'RAE_cl1_Costruzioni e materiali per costruzioni', 'RAE_cl1_Editoria e stampa', 'RAE_cl1_Elettrodomestici', 'RAE_cl1_Energia ed estrazione', 'RAE_cl1_Farmaceutica', 'RAE_cl1_Holding, finanziarie ed altro', "RAE_cl1_Intermedi per l'industria: beni vari", 'RAE_cl1_Largo consumo', 'RAE_cl1_Meccanica', 'RAE_cl1_Metallurgia e prodotti in metallo', 'RAE_cl1_Mobili', 'RAE_cl1_Servizi', 'RAE_cl1_Sistema moda', 'RAE_cl1_Trasporti', 'RAE_cl1_Utility', 'RAE_cl1_Valore missing', 'RAE_cl2_Primario', 'RAE_cl2_Secondario', 'RAE_cl2_Terziario', 'RAE_cl2_Valore missing', 'Rpp_entr_sof', 'Rpp_entrsof_uscsof', 'SAE_cl1_Altro', 'SAE_cl1_Valore missing', 'SAE_cl1_famiglie consumatrici', 'SAE_cl1_famiglie produttrici', 'SAE_cl1_societa finanziarie', 'SAE_cl1_societa non finanziarie', 'SAE_cl2_Altro', 'SAE_cl2_Famiglia', 'SAE_cl2_Societa', 'SAE_cl2_Valore missing', 'cd_istituto_lgd_1010', 'cd_istituto_lgd_99999', 'cd_rapplgd_appo_SEMPL_BR1', 'cd_rapplgd_appo_SEMPL_ML', 'cd_rapplgd_appo_SEMPL_MUT')

#combinazione dei 4 precedenti

colonne_indesiderate99_50 = ('lgd_perc_y','lgd_stim','RAE_111', 'RAE_112', 'RAE_12', 'RAE_120', 'RAE_13', 'RAE_130', 'RAE_14', 'RAE_140', 'RAE_151', 'RAE_161', 'RAE_162', 'RAE_163', 'RAE_170', 'RAE_19', 'RAE_20', 'RAE_211', 'RAE_212', 'RAE_221', 'RAE_222', 'RAE_223', 'RAE_224', 'RAE_231', 'RAE_232', 'RAE_233', 'RAE_239', 'RAE_241', 'RAE_242', 'RAE_243', 'RAE_244', 'RAE_245', 'RAE_246', 'RAE_247', 'RAE_248', 'RAE_252', 'RAE_253', 'RAE_255', 'RAE_256', 'RAE_257', 'RAE_258', 'RAE_259', 'RAE_260', 'RAE_30', 'RAE_311', 'RAE_312', 'RAE_313', 'RAE_314', 'RAE_315', 'RAE_316', 'RAE_321', 'RAE_322', 'RAE_323', 'RAE_324', 'RAE_325', 'RAE_326', 'RAE_327', 'RAE_328', 'RAE_330', 'RAE_341', 'RAE_342', 'RAE_343', 'RAE_344', 'RAE_345', 'RAE_346', 'RAE_347', 'RAE_351', 'RAE_352', 'RAE_353', 'RAE_361', 'RAE_362', 'RAE_363', 'RAE_364', 'RAE_365', 'RAE_371', 'RAE_372', 'RAE_373', 'RAE_374', 'RAE_411', 'RAE_412', 'RAE_413', 'RAE_414', 'RAE_415', 'RAE_416', 'RAE_417', 'RAE_418', 'RAE_419', 'RAE_420', 'RAE_421', 'RAE_422', 'RAE_423', 'RAE_424', 'RAE_425', 'RAE_426', 'RAE_427', 'RAE_428', 'RAE_429', 'RAE_431', 'RAE_432', 'RAE_436', 'RAE_438', 'RAE_439', 'RAE_441', 'RAE_442', 'RAE_451', 'RAE_455', 'RAE_456', 'RAE_461', 'RAE_462', 'RAE_463', 'RAE_464', 'RAE_465', 'RAE_466', 'RAE_471', 'RAE_472', 'RAE_473', 'RAE_474', 'RAE_481', 'RAE_482', 'RAE_483', 'RAE_491', 'RAE_492', 'RAE_493', 'RAE_494', 'RAE_495', 'RAE_506', 'RAE_507', 'RAE_509', 'RAE_611', 'RAE_612', 'RAE_613', 'RAE_614', 'RAE_615', 'RAE_616', 'RAE_618', 'RAE_619', 'RAE_620', 'RAE_643', 'RAE_644', 'RAE_646', 'RAE_647', 'RAE_649', 'RAE_652', 'RAE_653', 'RAE_655', 'RAE_656', 'RAE_671', 'RAE_672', 'RAE_710', 'RAE_721', 'RAE_722', 'RAE_724', 'RAE_725', 'RAE_730', 'RAE_741', 'RAE_742', 'RAE_750', 'RAE_761', 'RAE_762', 'RAE_763', 'RAE_764', 'RAE_771', 'RAE_772', 'RAE_773', 'RAE_790', 'RAE_840', 'RAE_850', 'RAE_920', 'RAE_930', 'RAE_940', 'RAE_950', 'RAE_960', 'RAE_981', 'RAE_982', 'RAE_983', 'RAE_999', 'RAE_cl1_Altri beni di consumo', 'RAE_cl1_Chimica di base e intermedi', 'RAE_cl1_Elettrodomestici', 'RAE_cl1_Energia ed estrazione', 'RAE_cl1_Farmaceutica', 'RAE_cl1_Largo consumo', 'RAE_cl1_Mezzi di trasporto', 'SAE_0', 'SAE_177', 'SAE_201', 'SAE_220', 'SAE_256', 'SAE_257', 'SAE_258', 'SAE_259', 'SAE_263', 'SAE_268', 'SAE_273', 'SAE_276', 'SAE_280', 'SAE_283', 'SAE_284', 'SAE_294', 'SAE_340', 'SAE_344', 'SAE_346', 'SAE_350', 'SAE_352', 'SAE_420', 'SAE_431', 'SAE_440', 'SAE_441', 'SAE_442', 'SAE_450', 'SAE_470', 'SAE_471', 'SAE_472', 'SAE_473', 'SAE_480', 'SAE_481', 'SAE_491', 'SAE_500', 'SAE_501', 'SAE_550', 'SAE_551', 'SAE_552', 'SAE_620', 'SAE_748', 'SAE_757', 'SAE_758', 'SAE_759', 'SAE_768', 'SAE_772', 'SAE_774', 'SAE_775', 'SAE_cl1_Altro', 'SAE_cl1_Valore missing', 'SAE_cl1_societa finanziarie', 'SAE_cl2_Altro', 'SAE_cl2_Valore missing', 'cd_istituto_lgd_10637', 'cd_istituto_lgd_3240', 'cd_istituto_lgd_32896', 'cd_istituto_lgd_3309', 'cd_istituto_lgd_3359', 'cd_istituto_lgd_6030', 'cd_istituto_lgd_6065', 'cd_istituto_lgd_6080', 'cd_istituto_lgd_6125', 'cd_istituto_lgd_6130', 'cd_istituto_lgd_6165', 'cd_istituto_lgd_6220', 'cd_istituto_lgd_6280', 'cd_istituto_lgd_6315', 'cd_istituto_lgd_6345', 'cd_istituto_lgd_6380', 'cd_istituto_lgd_6930','DT_ENTRATA_SOFF_year', 'DT_USCITA_year', 'D_GEO_SUD', 'D_PER_BT', 'D_PER_MLT_IPO', 'D_PER_MLT_NO_IP', 'D_PROC3_FALL', 'Delta_entr_usct', 'Delta_entrsof_uscsof', 'RAE_8', 'RAE_cl1_Agricoltura',

'RAE_cl1_Alimentare', 'RAE_cl1_Altri beni di consumo', 'RAE_cl1_Chimica di base e intermedi', 'RAE_cl1_Costruzioni e materiali per costruzioni', 'RAE_cl1_Editoria e stampa', 'RAE_cl1_Elettrodomestici', 'RAE_cl1_Energia ed estrazione', 'RAE_cl1_Farmaceutica', 'RAE_cl1_Holding, finanziarie ed altro', "RAE_cl1_Intermedi per l'industria: beni vari", 'RAE_cl1_Largo consumo', 'RAE_cl1_Meccanica', 'RAE_cl1_Metallurgia e prodotti in metallo', 'RAE_cl1_Mobili', 'RAE_cl1_Servizi', 'RAE_cl1_Sistema moda', 'RAE_cl1_Trasporti', 'RAE_cl1_Utility', 'RAE_cl1_Valore missing', 'RAE_cl2_Primario', 'RAE_cl2_Secondario', 'RAE_cl2_Terziario', 'RAE_cl2_Valore missing', 'Rpp_entr_sof', 'Rpp_entrsof_uscsof', 'SAE_cl1_Altro', 'SAE_cl1_Valore missing', 'SAE_cl1_famiglie consumatrici', 'SAE_cl1_famiglie produttrici', 'SAE_cl1_societa finanziarie', 'SAE_cl1_societa non finanziarie', 'SAE_cl2_Altro', 'SAE_cl2_Famiglia', 'SAE_cl2_Societa', 'SAE_cl2_Valore missing', 'cd_istituto_lgd_1010', 'cd_istituto_lgd_99999', 'cd_rapplgd_appo_SEMPL_BR1', 'cd_rapplgd_appo_SEMPL_ML', 'cd_rapplgd_appo_SEMPL_MUT')

colonne_indesiderate99_95 = ('lgd_perc_y','lgd_stim','RAE_111', 'RAE_112', 'RAE_12', 'RAE_120', 'RAE_13', 'RAE_130', 'RAE_14', 'RAE_140', 'RAE_151', 'RAE_161', 'RAE_162', 'RAE_163', 'RAE_170', 'RAE_19', 'RAE_20', 'RAE_211', 'RAE_212', 'RAE_221', 'RAE_222', 'RAE_223', 'RAE_224', 'RAE_231', 'RAE_232', 'RAE_233', 'RAE_239', 'RAE_241', 'RAE_242', 'RAE_243', 'RAE_244', 'RAE_245', 'RAE_246', 'RAE_247', 'RAE_248', 'RAE_252', 'RAE_253', 'RAE_255', 'RAE_256', 'RAE_257', 'RAE_258', 'RAE_259', 'RAE_260', 'RAE_30', 'RAE_311', 'RAE_312', 'RAE_313', 'RAE_314', 'RAE_315', 'RAE_316', 'RAE_321', 'RAE_322', 'RAE_323', 'RAE_324', 'RAE_325', 'RAE_326', 'RAE_327', 'RAE_328', 'RAE_330', 'RAE_341', 'RAE_342', 'RAE_343', 'RAE_344', 'RAE_345', 'RAE_346', 'RAE_347', 'RAE_351', 'RAE_352', 'RAE_353', 'RAE_361', 'RAE_362', 'RAE_363', 'RAE_364', 'RAE_365', 'RAE_371', 'RAE_372', 'RAE_373', 'RAE_374', 'RAE_411', 'RAE_412', 'RAE_413', 'RAE_414', 'RAE_415', 'RAE_416', 'RAE_417', 'RAE_418', 'RAE_419', 'RAE_420', 'RAE_421', 'RAE_422', 'RAE_423', 'RAE_424', 'RAE_425', 'RAE_426', 'RAE_427', 'RAE_428', 'RAE_429', 'RAE_431', 'RAE_432', 'RAE_436', 'RAE_438', 'RAE_439', 'RAE_441', 'RAE_442', 'RAE_451', 'RAE_455', 'RAE_456', 'RAE_461', 'RAE_462', 'RAE_463', 'RAE_464', 'RAE_465', 'RAE_466', 'RAE_471', 'RAE_472', 'RAE_473', 'RAE_474', 'RAE_481', 'RAE_482', 'RAE_483', 'RAE_491', 'RAE_492', 'RAE_493', 'RAE_494', 'RAE_495', 'RAE_506', 'RAE_507', 'RAE_509', 'RAE_611', 'RAE_612', 'RAE_613', 'RAE_614', 'RAE_615', 'RAE_616', 'RAE_618', 'RAE_619', 'RAE_620', 'RAE_643', 'RAE_644', 'RAE_646', 'RAE_647', 'RAE_649', 'RAE_652', 'RAE_653', 'RAE_655', 'RAE_656', 'RAE_671', 'RAE_672', 'RAE_710', 'RAE_721', 'RAE_722', 'RAE_724', 'RAE_725', 'RAE_730', 'RAE_741', 'RAE_742', 'RAE_750', 'RAE_761', 'RAE_762', 'RAE_763', 'RAE_764', 'RAE_771', 'RAE_772', 'RAE_773', 'RAE_790', 'RAE_840', 'RAE_850', 'RAE_920', 'RAE_930', 'RAE_940', 'RAE_950', 'RAE_960', 'RAE_981', 'RAE_982', 'RAE_983', 'RAE_999', 'RAE_cl1_Altri beni di consumo', 'RAE_cl1_Chimica di base e intermedi', 'RAE_cl1_Elettrodomestici', 'RAE_cl1_Energia ed estrazione', 'RAE_cl1_Farmaceutica', 'RAE_cl1_Largo consumo', 'RAE_cl1_Mezzi di trasporto', 'SAE_0', 'SAE_177', 'SAE_201', 'SAE_220', 'SAE_256', 'SAE_257', 'SAE_258', 'SAE_259', 'SAE_263', 'SAE_268', 'SAE_273', 'SAE_276', 'SAE_280', 'SAE_283', 'SAE_284', 'SAE_294', 'SAE_340', 'SAE_344', 'SAE_346', 'SAE_350', 'SAE_352', 'SAE_420', 'SAE_431', 'SAE_440', 'SAE_441', 'SAE_442', 'SAE_450', 'SAE_470', 'SAE_471', 'SAE_472', 'SAE_473', 'SAE_480', 'SAE_481', 'SAE_491', 'SAE_500', 'SAE_501', 'SAE_550', 'SAE_551', 'SAE_552', 'SAE_620', 'SAE_748', 'SAE_757', 'SAE_758', 'SAE_759', 'SAE_768', 'SAE_772', 'SAE_774', 'SAE_775', 'SAE_cl1_Altro', 'SAE_cl1_Valore missing', 'SAE_cl1_societa finanziarie', 'SAE_cl2_Altro', 'SAE_cl2_Valore missing', 'cd_istituto_lgd_10637', 'cd_istituto_lgd_3240', 'cd_istituto_lgd_32896', 'cd_istituto_lgd_3309', 'cd_istituto_lgd_3359', 'cd_istituto_lgd_6030', 'cd_istituto_lgd_6065', 'cd_istituto_lgd_6080', 'cd_istituto_lgd_6125', 'cd_istituto_lgd_6130', 'cd_istituto_lgd_6165', 'cd_istituto_lgd_6220', 'cd_istituto_lgd_6280', 'cd_istituto_lgd_6315', 'cd_istituto_lgd_6345', 'cd_istituto_lgd_6380', 'cd_istituto_lgd_6930','DT_ENTRATA_SOFF_year', 'D_GEO_SUD', 'D_PER_MLT_IPO', 'D_PROC3_FALL', 'Delta_entr_usct', 'RAE_cl1_Elettrodomestici', 'RAE_cl1_Farmaceutica', 'RAE_cl1_Holding, finanziarie ed altro', 'RAE_cl1_Mobili', 'RAE_cl1_Valore missing', 'RAE_cl2_Primario', 'RAE_cl2_Valore missing', 'Rpp_entrsof_uscsof', 'SAE_cl1_Valore missing',

'SAE_cl1_famiglie consumatrici', 'SAE_cl2_Altro', 'SAE_cl2_Famiglia', 'SAE_cl2_Societa', 'SAE_cl2_Valore missing')

colonne_indesiderate999_50 = ('lgd_perc_y','lgd_stim','RAE_111', 'RAE_112', 'RAE_120', 'RAE_13', 'RAE_130', 'RAE_140', 'RAE_151', 'RAE_161', 'RAE_162', 'RAE_163', 'RAE_170', 'RAE_19', 'RAE_20', 'RAE_211', 'RAE_212', 'RAE_221', 'RAE_222', 'RAE_224', 'RAE_232', 'RAE_233', 'RAE_239', 'RAE_241', 'RAE_244', 'RAE_246', 'RAE_252', 'RAE_253', 'RAE_256', 'RAE_257', 'RAE_259', 'RAE_260', 'RAE_311', 'RAE_315', 'RAE_323', 'RAE_326', 'RAE_327', 'RAE_362', 'RAE_363', 'RAE_364', 'RAE_365', 'RAE_371', 'RAE_374', 'RAE_411', 'RAE_414', 'RAE_415', 'RAE_416', 'RAE_418', 'RAE_420', 'RAE_422', 'RAE_424', 'RAE_425', 'RAE_426', 'RAE_427', 'RAE_429', 'RAE_438', 'RAE_456', 'RAE_462', 'RAE_464', 'RAE_466', 'RAE_471', 'RAE_481', 'RAE_482', 'RAE_492', 'RAE_643', 'RAE_655', 'RAE_710', 'RAE_721', 'RAE_724', 'RAE_725', 'RAE_730', 'RAE_741', 'RAE_742', 'RAE_750', 'RAE_762', 'RAE_763', 'RAE_764', 'RAE_773', 'RAE_999', 'RAE_cl1_Farmaceutica', 'SAE_0', 'SAE_177', 'SAE_201', 'SAE_220', 'SAE_256', 'SAE_257', 'SAE_258', 'SAE_259', 'SAE_263', 'SAE_268', 'SAE_273', 'SAE_276', 'SAE_283', 'SAE_284', 'SAE_294', 'SAE_340', 'SAE_344', 'SAE_346', 'SAE_350', 'SAE_352', 'SAE_420', 'SAE_431', 'SAE_440', 'SAE_441', 'SAE_470', 'SAE_471', 'SAE_472', 'SAE_473', 'SAE_500', 'SAE_550', 'SAE_551', 'SAE_552', 'SAE_748', 'SAE_757', 'SAE_758', 'SAE_759', 'SAE_768', 'SAE_772', 'SAE_774', 'SAE_775', 'SAE_cl1_Valore missing', 'SAE_cl2_Valore missing', 'cd_istituto_lgd_3309', 'cd_istituto_lgd_3359', 'cd_istituto_lgd_6065', 'cd_istituto_lgd_6080', 'cd_istituto_lgd_6125', 'cd_istituto_lgd_6165', 'cd_istituto_lgd_6280', 'cd_istituto_lgd_6380', 'cd_istituto_lgd_6930','DT_ENTRATA_SOFF_year', 'DT_USCITA_year', 'D_GEO_SUD', 'D_PER_BT', 'D_PER_MLT_IPO', 'D_PER_MLT_NO_IP', 'D_PROC3_FALL', 'Delta_entr_usct', 'Delta_entrsof_uscsof', 'RAE_8', 'RAE_cl1_Agricoltura', 'RAE_cl1_Alimentare', 'RAE_cl1_Altri beni di consumo', 'RAE_cl1_Chimica di base e intermedi', 'RAE_cl1_Costruzioni e materiali per costruzioni', 'RAE_cl1_Editoria e stampa', 'RAE_cl1_Elettrodomestici', 'RAE_cl1_Energia ed estrazione', 'RAE_cl1_Farmaceutica', 'RAE_cl1_Holding, finanziarie ed altro', "RAE_cl1_Intermedi per l'industria: beni vari", 'RAE_cl1_Largo consumo', 'RAE_cl1_Meccanica', 'RAE_cl1_Metallurgia e prodotti in metallo', 'RAE_cl1_Mobili', 'RAE_cl1_Servizi', 'RAE_cl1_Sistema moda', 'RAE_cl1_Trasporti', 'RAE_cl1_Utility', 'RAE_cl1_Valore missing', 'RAE_cl2_Primario', 'RAE_cl2_Secondario', 'RAE_cl2_Terziario', 'RAE_cl2_Valore missing', 'Rpp_entr_sof', 'Rpp_entrsof_uscsof', 'SAE_cl1_Altro', 'SAE_cl1_Valore missing', 'SAE_cl1_famiglie consumatrici', 'SAE_cl1_famiglie produttrici', 'SAE_cl1_societa finanziarie', 'SAE_cl1_societa non finanziarie', 'SAE_cl2_Altro', 'SAE_cl2_Famiglia', 'SAE_cl2_Societa', 'SAE_cl2_Valore missing', 'cd_istituto_lgd_1010', 'cd_istituto_lgd_99999', 'cd_rapplgd_appo_SEMPL_BR1', 'cd_rapplgd_appo_SEMPL_ML', 'cd_rapplgd_appo_SEMPL_MUT')

colonne_indesiderate999_95 = ('lgd_perc_y','lgd_stim','DT_ENTRATA_SOFF_year', 'D_GEO_SUD', 'D_PER_MLT_IPO', 'D_PROC3_FALL', 'Delta_entr_usct', 'RAE_cl1_Elettrodomestici', 'RAE_cl1_Farmaceutica', 'RAE_cl1_Holding, finanziarie ed altro', 'RAE_cl1_Mobili', 'RAE_cl1_Valore missing', 'RAE_cl2_Primario', 'RAE_cl2_Valore missing', 'Rpp_entrsof_uscsof', 'SAE_cl1_Valore missing', 'SAE_cl1_famiglie consumatrici', 'SAE_cl2_Altro', 'SAE_cl2_Famiglia', 'SAE_cl2_Societa', 'SAE_cl2_Valore missing','RAE_111', 'RAE_112', 'RAE_120', 'RAE_13', 'RAE_130', 'RAE_140', 'RAE_151', 'RAE_161', 'RAE_162', 'RAE_163', 'RAE_170', 'RAE_19', 'RAE_20', 'RAE_211', 'RAE_212', 'RAE_221', 'RAE_222', 'RAE_224', 'RAE_232', 'RAE_233', 'RAE_239', 'RAE_241', 'RAE_244', 'RAE_246', 'RAE_252', 'RAE_253', 'RAE_256', 'RAE_257', 'RAE_259', 'RAE_260', 'RAE_311', 'RAE_315', 'RAE_323', 'RAE_326', 'RAE_327', 'RAE_362', 'RAE_363', 'RAE_364', 'RAE_365', 'RAE_371', 'RAE_374', 'RAE_411', 'RAE_414', 'RAE_415', 'RAE_416', 'RAE_418', 'RAE_420', 'RAE_422', 'RAE_424', 'RAE_425', 'RAE_426', 'RAE_427', 'RAE_429', 'RAE_438', 'RAE_456', 'RAE_462', 'RAE_464', 'RAE_466', 'RAE_471', 'RAE_481', 'RAE_482', 'RAE_492', 'RAE_643', 'RAE_655', 'RAE_710', 'RAE_721', 'RAE_724', 'RAE_725', 'RAE_730', 'RAE_741', 'RAE_742', 'RAE_750', 'RAE_762', 'RAE_763', 'RAE_764', 'RAE_773', 'RAE_999', 'RAE_cl1_Farmaceutica', 'SAE_0', 'SAE_177', 'SAE_201', 'SAE_220', 'SAE_256', 'SAE_257', 'SAE_258', 'SAE_259', 'SAE_263', 'SAE_268',

```
'SAE_273', 'SAE_276', 'SAE_283', 'SAE_284', 'SAE_294', 'SAE_340', 'SAE_344', 'SAE_346', 'SAE_350', 'SAE_352', 'SAE_420', 'SAE_431', 'SAE_440', 'SAE_441', 'SAE_470', 'SAE_471', 'SAE_472', 'SAE_473', 'SAE_500', 'SAE_550', 'SAE_551', 'SAE_552', 'SAE_748', 'SAE_757', 'SAE_758', 'SAE_759', 'SAE_768', 'SAE_772', 'SAE_774', 'SAE_775', 'SAE_cl1_Valore missing', 'SAE_cl2_Valore missing', 'cd_istituto_lgd_3309', 'cd_istituto_lgd_3359', 'cd_istituto_lgd_6065', 'cd_istituto_lgd_6080', 'cd_istituto_lgd_6125', 'cd_istituto_lgd_6165', 'cd_istituto_lgd_6280', 'cd_istituto_lgd_6380', 'cd_istituto_lgd_6930')

#Colonne da considerare - Risultanti dal RFECV

colnRFECV56 = ['DUMMY_PER', 'DUMMY_IMM_FINALE', 'IW', 'D_PER_BT', 'D_PER_MLT_NO_IP','D_PROC3_ALTRO', 'D_GEO_NCE', 'Delta_entr_sof', 'Delta_entrsof_uscsof', 'Rpp_entr_sof', 'dt_entrata_year', 'dt_entrata_month', 'dt_entrata_day', 'DT_USCITA_year', 'DT_USCITA_month', 'DT_USCITA_day', 'DT_ENTRATA_SOFF_month', 'DT_ENTRATA_SOFF_day', 'cd_istituto_lgd_1010', 'cd_istituto_lgd_1025', 'cd_istituto_lgd_6160', 'cd_istituto_lgd_6225', 'cd_istituto_lgd_6385', 'cd_istituto_lgd_99999', 'cd_rapplgd_appo_SEMPL_BR1', 'cd_rapplgd_appo_SEMPL_BR2', 'cd_rapplgd_appo_SEMPL_BR3', 'cd_rapplgd_appo_SEMPL_MUT', 'SAE_430', 'SAE_482', 'SAE_492', 'SAE_614', 'SAE_615', 'SAE_cl1_famiglie produttrici', 'SAE_cl1_societa non finanziarie', 'RAE_8', 'RAE_11', 'RAE_505', 'RAE_630', 'RAE_645', 'RAE_648', 'RAE_660', 'RAE_830', 'RAE_984', 'RAE_cl1_Agricoltura', 'RAE_cl1_Alimentare', 'RAE_cl1_Costruzioni e materiali per costruzioni', 'RAE_cl1_Distribuzione', 'RAE_cl1_Elettrotecnica ed elettronica', "RAE_cl1_Intermedi per l'industria: beni vari", 'RAE_cl1_Metallurgia e prodotti in metallo', 'RAE_cl1_Servizi', 'RAE_cl1_Sistema moda', 'RAE_cl1_Trasporti', 'RAE_cl2_Secondario', 'RAE_cl2_Terziario']

#Diversi Dataset

x99_50 = dt3[[n for n in dt.columns if n not in colonne_indesiderate99_50]].copy()

x99_95 = dt3[[n for n in dt.columns if n not in colonne_indesiderate99_95]].copy()

x999_50 = dt3[[n for n in dt.columns if n not in colonne_indesiderate999_50]].copy()

x999_95 = dt3[[n for n in dt.columns if n not in colonne_indesiderate999_95]].copy()

xRFECV56 = dt3[[n for n in dt.columns if n in colnRFECV56]].copy()

#DATA SELECTION

#Correlazione Pearson

correlated_features = set()

correlation_matrix = xselect.corr()

for i in range(len(correlation_matrix.columns)):

    for j in range(i):

        if abs(correlation_matrix.iloc[i, j]) > soglia:#soglia ha assunto i valori 0.95 e 0.5

            if abs(correlation_matrix.iloc[i, len(correlation_matrix)-1])> abs(correlation_matrix.iloc[len(correlation_matrix)-1, j]):

                colname = correlation_matrix.columns[i]

                correlated_features.add(colname)
```

```
            else:
                colname = correlation_matrix.columns[j]
                correlated_features.add(colname)
            #questo ciclo ha permesso di determinare gli insiemi precedenti Pearson95 e Pearson50
#Analisi Distribuzione valori
colonne_varianza_minima = set()
for i in xselec.columns:
    for j in range(len(xselec[i].value_counts())):
        if len(xselec[i].value_counts())<3 and (xselec[i].value_counts()[j])/134901 > soglia:#soglia ha assunto i valori
0.99 e 0.999
            colonne_varianza_minima.add(i)
            #questo ciclo ha permesso di determinare gli insiemi precedenti Analisi_distrib99 e Analisi_distrib999
#Recursive Feature Elimination
estimator = RandomForestRegressor(random_state = 42)
selector = RFECV(estimator, step=2, cv=3)
selector = selector.fit(x99_95, y)
print('Optimal number of features :', selector.n_features_)
print('Best features :', x99_95.columns[selector.support_])
print('Original features :', x99_95.columns)
plt.figure()
plt.xlabel("Coppie di attributi")
plt.ylabel("Punteggio della Cross validation")
plt.plot(range(1, len(selector.grid_scores_) + 1), selector.grid_scores_)
plt.show()
#l'output di questo codice ha permesso di definire l'insieme colnRFECV56 e la Figura 4.7
#Principal Component Analysis
X_train, X_test, y_train, y_test = train_test_split(xRFECV56, y, test_size=0.2, random_state=42)
Standardizer = StandardScaler().fit(X_train)
X_train = Standardizer.transform(X_train)
X_test = Standardizer.transform(X_test)
pca = PCA(n_components=X_train.shape[1], random_state = 42)
```

```
X_train = pca.fit_transform(X_train, y_train)

X_test = pca.transform(X_test)

#creazione del dataframe CP

i= 1

col = []

while i <= X_train.shape[1]:

    col.append(f"principal cmp{i}")

    i = i+1

X_train = pd.DataFrame(data = X_train, columns = col)

X_test = pd.DataFrame(data = X_test, columns = col)

X_train = X_train.loc[:,:'principal cmpX']#X assume il valore del numero di CP desiderate. Utilizzati i valori 56 e 45.

X_test = X_test.loc[:,:'principal cmpX']#X assume il valore del numero di CP desiderate. Utilizzati i valori 56 e 45.

#Per SVM e Rete Neurale

Standardizer = StandardScaler().fit(X_train)

X_train = Standardizer.transform(X_train)

X_test = Standardizer.transform(X_test)

#Testando la PCA

estimator = Algoritmo(random_state=42)#Algoritmo assume le funzioni specificate in Tabella 4.4

estimator.fit(X_train, y_train)

y_pred = estimator.predict(X_test)

print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_pred))

print('Mean Squared Error:', metrics.mean_squared_error(y_test, y_pred))

print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))

print('Explained_Variance_score:',metrics.explained_variance_score(y_test, y_pred))

print('r2_score:',metrics.r2_score(y_test, y_pred))

#Testing degli insiemi nelle versioni di Default

insiemi = [x99_50, x99_95, x999_50, x999_95, xselec, xRFECV56]

nomi_insiemi = ['x99_50', 'x99_95', 'x999_50','x999_95', 'xselec','xRFECV56']

j=0

for i in insiemi:
```

```
    X_train, X_test, y_train, y_test = train_test_split(i, y, test_size=0.2, random_state=42)#per SVM non lineare test_size=0.8

    #Questo ciclo for è stato ripetuto per ogni algoritmo. Nella successiva riga, Algoritmo ha assunto tutte le funzioni specificate in Tabella 4.4

    regressor = Algoritmo(random_state=42)#per SVM e ReteNeurale: make_pipeline(StandardScaler(), Algoritmo(random_state=42))#per Rete Neurale anche Normalizer()

    regressor.fit(X_train, y_train)

    y_pred = regressor.predict(X_test)

    print(i.shape, nomi_insiemi[j])

    print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_pred))

    print('Mean Squared Error:', metrics.mean_squared_error(y_test, y_pred))

    print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))

    print('r2_score:',metrics.r2_score(y_test, y_pred))

    j=j+1

# FINE TUNING

random_grid = {}# In base all'algoritmo, questa griglia assume i valori specificati in Tabella 4.9

X_train, X_test, y_train, y_test = train_test_split(xRFECV56, y, test_size=0.2, random_state=42)#test_size=0.5 per SVM non Lineari

Alg = Algoritmo()#Algoritmo assume le funzioni specificate in Tabella 4.4

Alg_RSCV = RandomizedSearchCV(estimator = Alg, param_distributions = random_grid, n_iter = 20, cv = 3, verbose=2, random_state=42)

Alg_RSCV.fit(X_train, y_train)

Alg_RSCV.best_params_ #Per le Rete Neurali è stata applicata anche la PCA prima di RSCV


#I parametri ottenuti da questo codice sono stati utilizzati per testare nuovamente gli algoritmi secondo le logiche di test pregresse. Si evita di ripetere il codice.

# ENSEMBLE DEGLI ALGORITMI

#Ensamble su xRFECV56

X_train, X_test, y_train, y_test = train_test_split(xRFECV56, y, test_size=0.2, random_state=42)


#Decision Tree
```

```
tree_reg = DecisionTreeRegressor(random_state= 42,criterion = 'friedman_mse',min_samples_split = 40, min_samples_leaf = 40 ,max_leaf_nodes = 100, max_depth = 8)

tree_reg.fit(X_train, y_train)

y_pred_tree = tree_reg.predict(X_test)

#Random Forest

rf = RandomForestRegressor(n_estimators=1000, random_state=42, min_samples_split=2, min_samples_leaf= 2, max_depth= None)

rf.fit(X_train, y_train)

y_pred_rf = rf.predict(X_test)

#LSVR

pipelineLSVR = make_pipeline(StandardScaler(), LinearSVR(random_state=42, epsilon = 0.3, C=0.5))

pipelineLSVR.fit(X_train, y_train)

y_pred_lsvr = pipelineLSVR.predict(X_test)

#NN

pipelineMLP = make_pipeline(StandardScaler(), MLPRegressor(random_state=42, activation = 'logistic'))

pipelineMLP.fit(X_train, y_train)

y_pred_nn = pipelineMLP.predict(X_test)

#Creazione Dataframe dei risultati

dtE_columns = ['y_pred_tree','y_pred_rf','y_pred_lsvr','y_pred_nn']

dtEnsambl_test = pd.concat([pd.DataFrame(y_pred_tree),pd.DataFrame(y_pred_rf), pd.DataFrame(y_pred_lsvr),pd.DataFrame(y_pred_nn)],axis=1)

dtEnsambl_test.columns = dtE_columns

#calcolo della media come stimatore

dtEnsambl_test['Average'] = dtEnsambl_test.mean(axis=1)

y_pred = dtEnsambl_test.Average

#Nel caso di Ensemble con Random Forest finale

rfinal = RandomForestRegressor(n_estimators=500, random_state=42, min_samples_split=2, min_samples_leaf= 2, max_depth= None)

rfinal.fit(dtEnsambl_train, y_train)

y_pred = rfinal.predict(dtEnsambl_test)


print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_pred))
```

```
print('Mean Squared Error:', metrics.mean_squared_error(y_test, y_pred))

print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))

print('Explained_Variance_score:',metrics.explained_variance_score(y_test, y_pred))

print('r2_score:',metrics.r2_score(y_test, y_pred))

#Metriche della Regressione Multivariata

y_pred = dt3.lgd_stim

y_test = dt3.lgd_perc_y

print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_pred))

print('Mean Squared Error:', metrics.mean_squared_error(y_test, y_pred))

print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))

#print("R-Squaredon test dataset={}".format(tree_reg.score(X_test, y_pred))) #--> 26,16%

print('Explained_Variance_score:',metrics.explained_variance_score(y_test, y_pred))

print('R2:',metrics.r2_score(y_test, y_pred))

# GRAFICI

#E' stato utilizzato il seguente schema di codice per la maggior parte dei grafici rappresentanti distribuzioni

plt.ylabel('Titolo y')

plt.xlabel('Titolo x')

plt.title('Titolo Grafico')

graf = dati_ascisse # con dati_ascisse

sns.distplot(graf, norm_hist = True);

#Figure della PCA

num_componenti = np.arange(1,dataset.shape[1],5)# dataset ha assunto i valori xselec e xRFECV56

perc_varianza = []

for s in num_componenti:

    var = pca.explained_variance_ratio_

    categ = var[:s].sum()

    perc_varianza.append(categ)

perc_varianza

plt.figure(figsize=(5,5))

plt.title("Analisi Componenti Principali")

plt.xlabel('Numero di CP')
```

```
plt.ylabel('% Cumulata Varianza Catturata')

plt.scatter(num_componenti, perc_varianza, alpha=0.5)

plt.show()
```

Bibliografia

[1] B. d'Italia, «“Rapporto sulla stabilità finanziaria”,» Aprile 2020.

[2] A. Géron, "Hands-On Machine Learning with Scikit-Learn and TensorFlow Concepts, Tools and Techniques to Build Intelligent Systems", O'Reilly Media, 2017.

[3] B. f. I. Settlements, «https://www.bis.org/,» 2017. [Online].

[4] Wikipedia, «https://it.wikipedia.org/,» 2017. [Online].

[5] Q. Francesca, in *Rischio di credito e valutazione della Loss Given Default*, Roma, Bancaria Editrice, 2007.

[6] B. d'Italia, «www.bancaditalia.it/,» [Online]. Available: https://www.bancaditalia.it/media/views/2017/npl/faq/index.html#faq8761-7.

[7] C. Barbagallo, «“I crediti deteriorati delle banche italiane: problematiche e tendenze recenti”,» Roma, 6 giugno 2017.

[8] «https://www.bancaifis.it,» 29 01 2020. [Online]. Available: https://www.bancaifis.it/comunicati-stampa/market-watch-npl-gennaio-2020-2/. [Consultato il giorno 12 giugno 2020].

[9] D. Cucinelli, «“The Impact of Non-performing Loans on Bank Lending Behavior: Evidence from the Italian Banking Sector",» *Eurasian Journal of Business and Economics,* 2015.

[10] P. Angelini, «“I crediti deteriorati: mercato, regole e rafforzamento del sistema",» Roma, 9 ottobre 2018.

[11] «www.gazzettaufficiale.it,» [Online]. Available: https://www.gazzettaufficiale.it/eli/id/2016/2/15/16G00025/sg. [Consultato il giorno 12 06 2020].

[12] «www.bancaditalia.it,» [Online]. Available: https://www.bancaditalia.it/compiti/vigilanza/normativa/archivio-norme/comunicazioni/com-20160329/Comunic_20160329.pdf. [Consultato il giorno 13 06 2020].

[13] «www.bankingsupervision.europa.eu,» [Online]. Available: https://www.bankingsupervision.europa.eu/ecb/pub/pdf/guidance_on_npl.it.pdf . [Consultato il giorno 13 06 2020].

[14] «www.abbrevia.it,» [Online]. Available: https://www.abbrevia.it/it/npl-e-utp-2020-la-situazione-attuale-ed-i-protagonisti-del-mercato/. [Consultato il giorno 13 06 2020].

[15] M. Longo, «“Npl, rischio «bolla» per il mercato dei crediti deteriorati”,» *Il Sole 24 Ore,* 11 02 2019.

[16] G. Bakshi, D. Madan e F. Zhang, «"Understanding the role of Recovery in Default Risk Models: Empirical Comparison and Implied Recovery Rates",» 6 November 2001.

[17] H. Unal, D. Madan e L. Guntay, «"Pricing te Risk of Recovery in Default with APR Violations",» *Journal of Banking & Finance,* pp. 1001-1025, 2006.

[18] A. Ricotti e G. De Vincenzo, «” L’utilizzo della fiscalità in chiavemacroprudenziale: l'impatto di alcune recenti misure tributarie sullaprociclicità e sulla stabilità delle banche",» *Note di stabilità finanziaria,* n. 1, Aprile 2014.

[19] L. G. Ciavoliello, F. Ciocchetta, F. M. C. I. Guida, A. Rendina e G. Santini, «"Quanto valgono i crediti deteriorati",» *Note di stabilità Finanziaria,* n. 3, 3 aprile 2016.

[20] Banca d'Italia, «https://www.bancaditalia.it,» [Online]. Available: https://www.bancaditalia.it/compiti/vigilanza/normativa/archivio-norme/circolari/c272/CIRC272_integrale_8agg.pdf. [Consultato il giorno 07 Giugno 2020].

[21] «http://www.treccani.it,» [Online]. Available: http://www.treccani.it/enciclopedia/cointegrazione_%28Dizionario-di-Economia-e-Finanza%29/. [Consultato il giorno Giugno 2020].

[22] «www.wikipedia.it,» [Online]. [Consultato il giorno Aprile 2020].

[23] «www.wikipedia.org,» [Online]. Available: https://it.wikipedia.org/wiki/Democrazia#Democrazia_nel_mondo_contemporaneo. [Consultato il giorno Aprile 2020].

[24] C. De secondat De Montesquieu, "Lo spirito delle leggi", Milano, 1967.

[25] Y. Freund e R. E. Schapire, «A decision-theoretic generalization of on-line learning and an application to boosting,» *Journal of Computer and System Sciences,* pp. 119-139, Agosto 1997.

[26] J. H. Friedman, «Greedy Function Approximation: A Gradient Boosting Machine,» Aprile 2001.

[27] S. Herculano-Houzel, «“The Human Brain in Numbers: A Linearly Scaled-up Primate Brain”,» *Front Hum Neurosci,* 2009.

[28] D. O. Hebb, The organization of behavior; a neuropsychological theory., New York: Wiley, 1949.

[29] W. McCulloch e W. Pitts, «"A Logical Calculus of Ideas Immanent in Nervous Activity",» *Bulletin of Mathematical Biophysics,* n. 5, p. 115–133., 1943.

[30] F. Rosenblatt, «“The Perceptron: A Probabilistic Model For Information Storage and Organization in the Brain”,» *Psychological Review,* vol. 65, 1958.

[31] M. Minsky e S. Papert, "Perceptrons: An introduction to Computational Geometry", M.I.T Press, 1969.

[32] J. Bastos, «"Ensemble Predictions of Recovery Rates",» *Journal of Financial Services Research,* vol. 46, p. 177–93., 2014.

[33] Y. Xiao, J. Crook e G. Andreeva, «“Enhancing two-stage modelling methodology for loss given default with support vector machines”,» *European Journal of Operational Research,* vol. 263, p. 679–89, 2017.

[34] «"Not so big",» *The Economist,* 13 giugno 2020.

[35] «www.wikipedia.org,» [Online]. Available: https://it.wikipedia.org/wiki/Potenza_di_due.

[36] T. Mitchell, "Machine Learning", McGraw Hill, 1997.

[37] «https://www.weforum.org/,» [Online]. Available: https://www.weforum.org/agenda/2019/10/quantum-computers-next-frontier-classical-google-ibm-nasa-supremacy/. [Consultato il giorno 19 01 2020].

[38] C. M. Bishop, "Pattern Recognition and Machine Learning", Springer, 2006.

[39] P. Domingos, "L'Algoritmo Definitivo: La macchina che impara da sola e il futuro del nostro mondo", Bollati Boringhieri, 2016.

[40] P. Wolfe, «"A duality theorem for non-linear programming",» *Quaterly of Applied Mathematics,* vol. 19, pp. 239-244, 1961.

[41] A. Sironi e A. Resti, Rischio e valore nelle banche, Milano: Egea, 2008.

[42] Comitato di Basilea, «bis.org,» [Online]. Available: https://www.bis.org/publ/bcbs189_it.pdf. [Consultato il giorno 12 06 2020].

[43] M. Marcucci, A. Pischedda e V. Profeta, «“The changes of the Italian insolvency and foreclosure regulation adopted in 2015",» *Notes on Financial Stability and Supervision,* n. 2, 2015.

[44] E. Brodi, S. Giacomelli, I. Guida, M. Marcucci, A. Pischedda, V. Profeta e G. Santini, «“Nuove misure per velocizzare il recupero dei crediti: una prima analisi del D.L. 59/2016”,» *Note di stabilità finanziaria e vigilanza,* n. 4, Agosto 2016.