POLITECNICO
DI TORINO

Corso di Laurea in Ingegneria Informatica

Tesi di Laurea Magistrale

Algoritmi di Data Mining applicati al processo produttivo

Progettazione di un modello predittivo per il controllo della qualità

Relatore
Prof. Paolo Garza

Candidato
Marco Schiuma

Tutore Aziendale
FCA ITALY
Dot.ssa Anna Adilardi

Settembre 2018

Ringraziamenti

Vorrei ringraziare l'ufficio ICT dello stabilimento di Mirafiori per avermi fornito il supporto necessario per svolgere il progetto presentato in questo elaborato, in particolare la Dott.ssa Anna Adilardi e l'Ing. Vittorio De Carlo. Un ringraziamento al professore Paolo Garza, relatore di questa tesi di laurea, per la grande disponibilità e precisione dimostratemi durante il periodo di stesura.

Indice

Elenco delle figure

Capitolo 1

Introduzione

In questo capitolo viene fornita un'introduzione al lavoro svolto durante il mio progetto di tesi. Dopo una breve descrizione del contesto aziendale in cui si è svolto il tirocinio, vengono introdotti alcuni concetti che hanno guidato le aziende in un processo di rinnovamento, ovvero Industry 4.0, Internet of Things e Knowledge Data Discovery, e che sono alla base del mio progetto personale.

Il lavoro descritto nel presente elaborato si è svolto presso l'ufficio ICT di Maserati, gruppo FCA[1], all'interno dello stabilimento Mirafiori di Torino. Fiat Chrysler Automotive (FCA) è un'azienda italo-statunitense, nata nel 2014 dall'acquisizione da parte di Fiat S.p.A dell'azienda statunitense Chrysler Group.

L'azienda figura tra le principali case automobilistiche mondiali e fin dalla sua nascita ha adottato, in tutti i suoi stabilimenti, i principi del World Class Manufacturing (WCM).

1.1 WCM: World Class Manufacturing

Il termine WCM fu coniato dallo studioso americano Richard Schonberger negli anni Ottanta, per definire l'insieme di metodologie di ottimizzazione della produzione adottate dalle migliori industrie giapponesi [11].

Il WCM definisce aspetti tecnici e gestionali che le industrie devono rispettare per realizzare prodotti in maniera più rapida, economica e di maggiore qualità. Per ottenere tutto ciò, bisogna adottare un approccio di continuo sviluppo e miglioramento di tutti i processi, seguendo le linee guida definite dal WCM.

[1] https://www.fcagroup.com/it-it/pages/home.aspx

1.1.1 Concetti

Il World Class Manufacturing descrive una serie principi, standard, e tecniche da seguire, finalizzati alla riduzione al minimo di tutti i tipi di costi e perdite, in ogni fase aziendale.

La figura che segue riassume i principali concetti del WCM e si può subito notare come gli ambiti coinvolti siano molteplici e varino dalla produzione alla gestione degli ordini, alla manutenzione dei macchinari. In un sistema come questo infatti, è continuamente richiesto a ogni impiegato di creare valore aggiunto nello svolgimento delle sue funzioni.

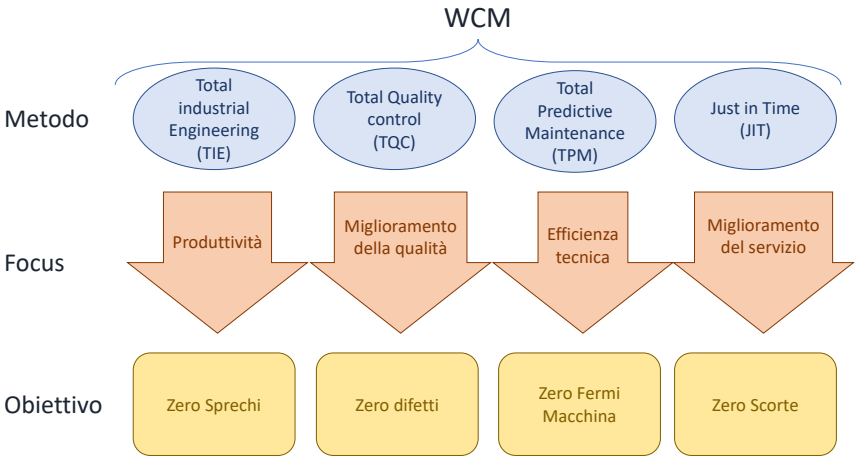


Figura 1.1. Concetti fondamentali del WCM [2]

Pur basandosi su metodologie già esistenti e applicate da alcune realtà industriali, come TIE, TQC, TPM, JIT (Figura 1.1), il WCM ha inglobato queste logiche in un nuovo modello, ponendo particolare attenzione alla riduzione di sprechi, scorte, difetti e fermi macchina, per ottimizzare l'efficienza.

1.1.2 Pilastri

Alla base del WCM ci sono 20 *Pillars* (Pilastri) di cui 10 tecnici e 10 gestionali. Ogni pilastro tecnico descrive precisi standard da seguire per ottenere un determinato obiettivo, mentre i pilastri gestionali, più generici, fungono da supporto per quelli tecnici. Nonostante la grande varietà di argomenti toccati, i pilastri non devono essere visti come entità a sé stanti, ma piuttosto come un insieme di azioni e standard da seguire, per ottenere un obiettivo comune. Il richiamo visivo a una costruzione simile a un tempio, con fondamenta e pilastri, non è casuale, ma vuole sottolineare l'imprescindibilità di ogni elemento.

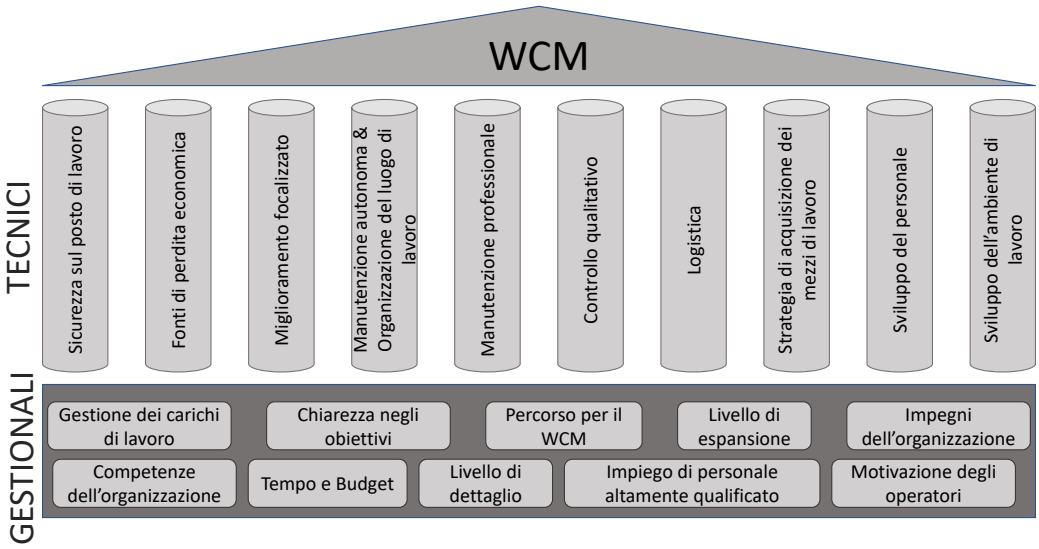


Figura 1.2. Pilastri del WCM [2]

Molti degli aspetti tecnici del WCM si integrano con il modello di Industria 4.0, il quale prevede la digitalizzazione dei processi produttivi, come stampa 3D, robotica avanzata, sistemi di simulazione e di realtà virtuale e aumentata, collegamento continuo di tutti gli oggetti, le macchine e le persone tramite reti IoT [9].

1.2 Industria 4.0

Negli ultimi anni molte aziende manifatturiere, hanno modificato i loro processi produttivi, rendendoli sempre più automatizzati e integrando nuove tecnologie al fine di migliorare la qualità e l'efficienza della produzione. Questa tendenza ha preso il nome di "Industria 4.0" oppure di "Quarta Rivoluzione Industriale", mettendo in risalto l'accezione storica di questo cambiamento, oppure ancora "Impresa 4.0" indicando più in generale i piani economico-strategici di governi e aziende, per adattarsi a questo cambiamento.

I cambiamenti introdotti dall'Industria 4.0 hanno avuto un impatto notevole tanto da arrivare a parlare di *Rivoluzione*, in modo da differenziare in maniera netta ciò che è stato fino ad ora e ciò che sarà dopo. Tuttavia, per comprendere a pieno l'entità dell'evoluzione, è necessario fare un breve excursus sulle altre grandi rivoluzioni che hanno caratterizzato l'industria nel corso della sua storia.

La prima grande Rivoluzione Industriale avvenne tra gli anni 1760 e 1830, durante i quali la produzione si è evoluta a seguito dell'introduzione delle macchine a discapito della forza fisica. In questo periodo è di grande importanza l'uso del

carbone al posto della legna, che era in grado di fornire una maggiore resa. Con l'introduzione delle macchine la produzione aumentò in quantità e anche in qualità e la rivoluzione che nacque in Inghilterra, presto si diffuse nel resto dell'Europa e negli Stati Uniti.

La seconda Rivoluzione Industriale, nota anche come Rivoluzione Tecnologica, avvenne tra gli anni 1840 e 1870. La nascita della ferrovia e l'utilizzo della corrente elettrica al posto del carbone, trainarono l'industria verso un nuovo modo di produrre. La produzione di massa nasce in questo periodo.

La terza Rivoluzione Industriale si ebbe nella seconda metà del ventesimo secolo. Dopo le due Guerre Mondiali, che causarono un rallentamento nello sviluppo tecnologico e industriale, oltre alle crisi economiche che ne conseguirono, l'industria riprese la sua evoluzione. In particolare l'invenzione dei primi calcolatori, la nascita della prima rete telematica ARPANET dalla quale poi nacque Internet, portarono un nuovo e profondo cambiamento nell'industria e nella vita personale di ognuno.

Nella quarta Rivoluzione Industriale, infine, le macchine hanno reso autonomi molti processi che prima venivano effettuati manualmente, la gestione degli ordini è gestita automaticamente da un sistema, e molto spesso anche i rapporti con i fornitori sono gestiti automaticamente. Nelle industrie diventa di estrema importanza l'integrazione tra le macchine e le nuove tecnologie come la rete internet, i sensori dell'Internet of Things, oltre alla competenza degli operatori che supervisionano i processi.[4]

Il termine Industria 4.0 è stato coniato in Germania nel 2013, in particolare da tre studiosi tedeschi: Henning Kagermann, Wolf-Dieter Lukas e Wolfgang Wahlster che dopo aver introdotto il concetto in varie conferenze e convegni in Germania, stilarono il "*Final report of the Industrie 4.0 Working Group*"[8], nel quale viene definito un nuovo modo di concepire l'industria, delineando gli investimenti, le strategie e i cambiamenti da apportare alle industrie tradizionali per riportare la manifattura tedesca ai vertici europei e mondiali.

I risultati ottenuti dalle aziende tedesche si sono rivelati sin da subito notevoli, tanto da convincere anche altre aziende europee e non ad adottare questo nuovo livello produttivo nelle proprie realtà.

Le principali tecnologie che stanno trasformando la produzione industriale sono nove: Automazione, Simulazione, Integrazione dei Sistemi, IoT (Internet of Things), Cybersecurity, Cloud, Produzione Additiva, Realtà Aumentata, Big data Analytics [10]. Molte aziende, tra cui anche FCA, investono per introdurre queste nuove tecnologie al fine di ottenere risultati in termini di efficienza e qualità. Oggigiorno, non è raro trovare in aziende e industrie, oltre a cellulari e PC aziendali, oggetti come smartwatch e stampanti 3d utilizzati quotidianamente, per migliorare e digitalizzare alcuni processi.

L'introduzione di queste tecnologie, in particolare l'automazione raggiunta grazie all'utilizzo di numerosi robot nella linea produttiva e la trasformazione portata dall'IoT, hanno favorito l'acquisizione e l'archiviazione di dati provenienti da quasi

tutti i processi aziendali. Questi dati possono riguardare le macchine, il prodotto, il processo, la manutenzione, il controllo della qualità, il rilevamento di guasti, e sono raccolti dai sistemi che gestiscono l'intero processo come il *MES*, descritto nel capitolo seguente.

I dati che vengono immagazzinati per tenere traccia degli step della catena di montaggio, possono essere utilizzati per migliorare il processo stesso. La sfida che oggi le aziende si trovano ad affrontare, è quella di riuscire a utilizzare questa grande mole di dati grezzi al meglio. In quest'ambito il Data Mining e il KDD (Knowledge Data Discovery) offrono numerose opportunità e campi di applicazione.

1.3 KDD: Knowledge Data Discovery

Il termine KDD è stato formalizzato per la prima volta da Usama Fayyad durante la conferenza *First International Conference on Knowledge Discovery* [5] tenuta a Montreal nel 1995 e considerata ancora una delle principali conferenze su questo argomento [6]. Usama Fayyad è stato anche il primo a definire KDD come segue:

> *Knowledge Discovery in Databases is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data.*

Il Knowledge Data Discovery, quindi, è quel processo che comprende una serie di strumenti e tecniche usati per estrarre informazioni, non facilmente deducibili, da grandi basi di dati, warehouse, direttamente dal Web o da altre fonti di data stream[7].

Il processo è interattivo e iterativo (Figura 1.3), infatti, comprende 9 punti nei quali l'utente prende decisioni importanti che determineranno l'esito dell'analisi, inoltre spesso è necessario iterare più volte per trovare la struttura migliore da analizzare.

Di seguito vengono brevemente descritti i principali punti del KDD[5], che costituiranno la struttura del mio progetto:

1. Comprendere come funziona l'intero sistema, dove e come avviene la raccolta dei dati, identificare l'obiettivo dell'intero processo dal punto di vista del cliente;

2. Creare un dataset integrando più fonti di dati oppure selezionando una serie di attributi o tuple rilevanti sul quale basare la propria analisi;

3. *Data cleaning* e *Data preprocessing*: due step molto importanti descritti nel capitolo successivo, che comprendono ad esempio la riduzione del rumore, l'eliminazione degli outliers, e la gestione dei valori mancanti [2.2.1] [2.2];

4. *Data reduction and projection*: spesso conviene ridurre la cardinalità di un dataset o il numero di attributi, in modo da analizzare più facilmente i dati ottenendo risultati simili.

5. Scegliere la tecnica di analisi più adatta per ottenere l'obiettivo preposto (step 1), ad esempio classificazione, regressione o clustering;

6. Scegliere l'algoritmo da utilizzare, modificando appropriatamente i dati in ingresso (ad esempio alcuni algoritmi accettano valori discreti, mentre altri valori continui) 2.

7. *Data mining*: ricerca di pattern frequenti, oppure di una particolare rappresentazione dei dati tramite classificazione, alberi decisionali, clustering, ecc. Il risultato di questo step, dipende fortemente da quelli precedenti.

8. Interpretazione dei risultati, e possibile iterazione dei passaggi 1-7.

9. Utilizzare le informazioni ricavate in altri sistemi, o risolvere conflitti in caso di risultati attesi differenti.

Come è possibile notare, nella definizione di Usama Fayyad[5], il *Data Mining* costituisce uno dei punti del *KDD* ed è, quindi, incorporato in un processo più ampio. Successivamente però, il termine *Data Mining* ha inglobato le fasi di *preprocessing* e interpretazione dei dati, diventando sinonimo dell'intero processo di estrazione dei dati[6].

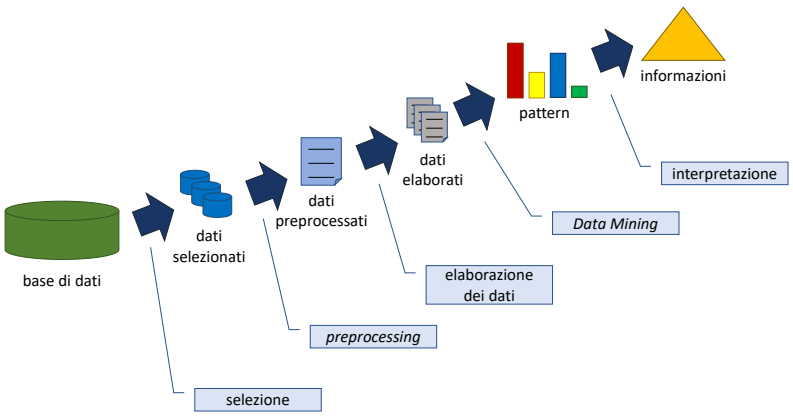


Figura 1.3. Processo del Knowledge Discovery from Data

1.4 Data Mining nel Processo Produttivo

Alcune industrie utilizzano potenti sistemi per l'acquisizione e l'analisi dei dati sulla maggior parte dei processi aziendali. I dati raccolti possono riguardare la produzione, i macchinari, il prodotto, i processi, la manutenzione, il rilevamento dei guasti o il controllo qualità e spesso vengono archiviati a vari livelli, in delle basi di dati, per formare uno storico più o meno grande[3].

Prima dell'avvento dell'Industria 4.0 e in particolare dell'affermazione del *KDD*, queste basi di dati venivano analizzate esclusivamente utilizzando tecniche matematiche e statistiche, al fine di trovare dei pattern noti, che potessero aiutare a migliorare i vari processi aziendali. La crescente complessità dei macchinari utilizzati, e l'introduzione di molteplici sensori costantemente connessi tra di loro, abbinati ai progressi tecnologici nei sistemi di acquisizione e archiviazione, però, hanno causato una crescita esponenziale delle basi di dati sia nella cardinalità che nella dimensionalità. Molto spesso, per modellare correttamente alcuni comportamenti del sistema, questi attributi devono essere considerati contemporaneamente. Tutto ciò ha portato all'introduzione di nuove tecniche di analisi per l'estrazione di informazione utile partendo da dati grezzi[3].

Data la grande disponibilità di dati in aziende manifatturiere, le possibili applicazioni di tecniche di *Data Mining* e *Machine Learning* al processo produttivo, sono molteplici. Come già detto in precedenza, gli ambiti variano dai processi, al prodotto, ai macchinari e molto spesso, i dati che vengono prelevati e archiviati non sono oggetto di analisi approfondite o, a volte, non vengono utilizzati affatto. Nel caso dei dati prelevati dai macchinari, ad esempio, spesso lo storico copre una finestra temporale finita e limitata, utilizzata prevalentemente per tracciare eventuali e, per lo più rari, casi di fermo macchina o guasti.

Convertire i dati grezzi in informazione è la grande sfida che un'azienda manifatturiera si trova ad affrontare oggi. Analizzando i dati, infatti, si possono ricavare informazioni preziose, utili a migliorare i processi di produzione in termini di efficienza, qualità o costi. Un'industria che è in grado integrare nei propri processi aziendali, una sistematica analisi dei dati, risulta più competitiva a livello globale.

Nel caso di questo elaborato, come specificato nel capitolo 4, si analizza la possibile applicazione di algoritmi di classificazione, con conseguente addestramento di un modello predittivo, alle macchine saldatrici dell'unità di saldatura del plant di Mirafiori. Infine, vengono presentati i risultati ottenibili dall'utilizzo di un modello simile, per l'analisi dei dati in tempo reale, e i possibili sviluppi futuri.

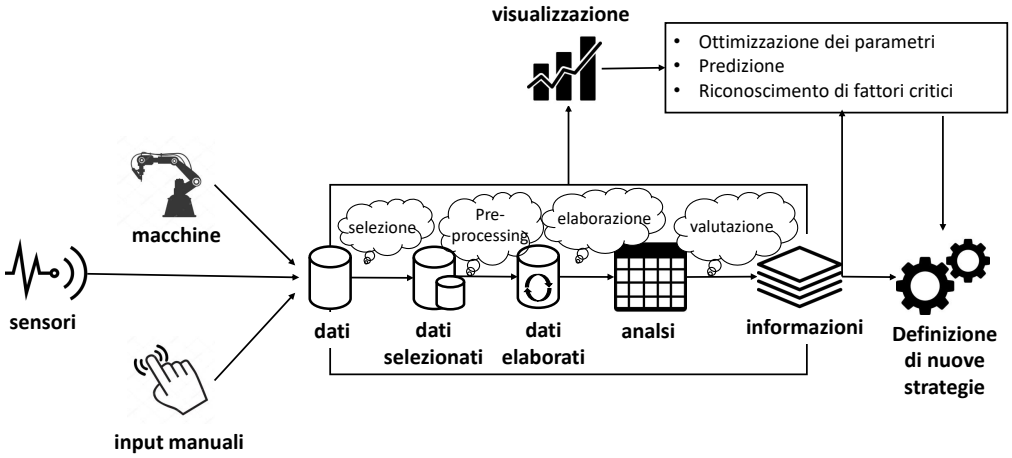


Figura 1.4. Analisi dei dati prelevati dal processo produttivo

Capitolo 2

Richiami teorici

In questo capitolo si vogliono richiamare alcune definizioni e concetti del *Data Mining* applicati in seguito.

Il primo *step* del *Data Mining* consiste nella preparazione dei dati che si vogliono analizzare. I dati raccolti dal mondo reale, essendo dati grezzi, possono essere affetti da *rumore*, ovvero valori anomali dovuti a errori di misurazione o trascrizione; spesso questi dati contengono molta più informazione di quella necessaria per l'analisi che si vuole effettuare, oppure attributi che non sono strettamente correlati con l'informazione che si vuole estrarre e quindi inutili se non, addirittura, controproducenti. In fase preliminare, quindi, è necessario uno studio approfondito sui dati, che comprende un'analisi sulla tipologia del dato, sul significato di tutti gli attributi presenti e il processo che l'ha generato.

2.1 Tipologia dei dati

Ci sono molti tipi di *dataset* e con il progressivo svilupparsi di tecniche di analisi e tecnologie di archiviazione la lista è in continuo aggiornamento. Una possibile classificazione dei vari tipi di dataset può essere la seguente[12]:

- **Record**: molto spesso, anche in ambito industriale, ci si trova ad analizzare dati archiviati sotto forma di *record*, ognuno dei quali è descritto da una serie definita di attributi. A priori, in genere, non è deducibile alcuna relazione tra record diversi, o tra attributi dello stesso record. Spesso questo tipo di data set è archiviato sotto forma di database relazionali, come ad esempio:

  1. *Tabelle*: classica rappresentazione tabulare, nella quale ogni riga rappresenta un oggetto, chiamato anche tupla, mentre le colonne rappresentano un numero prefissato di attributi che descrivono un determinato oggetto;
  2. *Documenti*: una rappresentazione alternativa alle tabelle, ma sempre di forma relazionale, è quella sotto forma di documenti. Ogni documento

rappresenta un oggetto sotto forma di vettore, ogni elemento del vettore rappresenta un attributo di quell'oggetto.

3. *Transazioni*: un tipo speciale di record dataset, nel quale ogni record rappresenta una transazione, e quindi include una serie di oggetti. Un esempio si può trovare nell'analisi degli oggetti acquistati in un alimentari. In questo genere di dataset è spesso interessante analizzare quali prodotti vengono acquistati insieme in una stessa transazione che, in questo caso, rappresenterebbe la spesa, mentre i singoli prodotti acquistati rappresentano gli oggetti.

- **Graph**: è una rappresentazione molto utile nel caso in cui:

  1. è necessario evidenziare una relazione tra oggetti, come ad esempio nel caso di generici link HTML, in cui la relazione tra pagine web è l'informazione rilevante;
  2. l'oggetto stesso è rappresentabile sotto forma di grafo, come ad esempio nelle formule chimiche delle molecole.

- **Ordered**: in altri casi, invece, è importante l'ordine nel tempo o nello spazio; come ad esempio sequenze di transazioni, in cui la precedenza è fondamentale, o l'analisi della sequenza genomica, oppure l'analisi di dati spazio-temporali come può esserlo uno studio della temperatura media mensile nelle varie aree geografiche terrestri.

Un *data set* è formato da oggetti, se un oggetto è archiviato in un database, questo corrisponderà a una *tupla*; in un database relativo al processo produttivo, ad esempio, un oggetto potrebbe essere un singolo processo effettuato sul prodotto. Ogni oggetto è rappresentato da una serie di attributi; nell'esempio in questione, a seconda del prodotto e del processo su esso svolto, che sia una laminazione, o una piegatura o una saldatura, gli attributi possono essere diversi, ma descrivono, comunque, una caratteristica del processo.

Gli attributi possono essere di diverso tipo. Un modo efficiente per distinguere le varie tipologie di attributo, è considerare le operazioni che è possibile effettuare[12]. Di seguito sono descritte 4 categorie di attributi:

- *Nominal*: attributo che ammette nomi predefiniti come valori. Attributi nominali forniscono informazione utile solo a distinguere un oggetto da un altro ($\neq$, $=$).

- *Ordinal*: attributo che ammette valori di una serie ordinata. Attributi di questo tipo forniscono informazione utile non solo per distinguere gli oggetti, ma anche per compararli tra di loro ($\neq$, $=$, $>$, $<$, $\leq$, $\geq$).

- *Interval*: attributi che, oltre alle operazioni ammesse dalle tipologie precedenti, sono caratterizzati da una unità di misura e quindi anche la differenza risulta significativa ($\neq$, $=$, $>$, $<$, $\leq$, $\geq$, $+$, -).

- *Ratio*: attributi che consentono anche operazioni di rapporto ($\neq$, $=$, $>$, $<$, $\leq$, $\geq$, $+$, -, *, /).

Gli attributi possono essere inoltre *discreti* o *continui*. Spesso, è necessario *discretizzare* (2.2) gli attributi continui, in modo da diminuire la complessità dell'analisi, ottenendo risultati comunque ottimali.

Conoscere la tipologia dei dati e, quindi, quella degli attributi degli oggetti, risulta molto importante durante l'analisi, poiché in questo modo è possibile scegliere l'approccio da utilizzare e gli algoritmi applicabili e comprendere se è necessario manipolare alcuni dati tramite discretizzazione o aggregazione.

2.2 Data Preprocessing

Una conoscenza approfondita della tipologia degli attributi e delle caratteristiche dei dati è necessaria per poter identificare valori errati o *outliers* che verranno gestiti in questa fase.

Come già accennato all'inizio del capitolo, i dati prelevati dai sistemi di acquisizione sono spesso affetti da rumore. Più in generale si può parlare di qualità dei dati in termini di *accuratezza, completezza* e *consistenza*.

I grossi dataset prelevati dal mondo reale sono spesso poco accurati, incompleti e inconsistenti, quindi è necessario elaborarli prima di analizzarli.

- I dati possono non essere accurati, ad esempio, a causa di errori nei sistemi di acquisizione, che registrano valori errati degli attributi, oppure a causa di errori di trasmissione da un sistema di acquisizione a uno di archiviazione. Gli errori più comuni si hanno in casi di inserimento manuale dei valori degli attributi; in questo caso gli errori possono essere sia involontari che volontari (come ad esempio l'inserimento di dati personali fittizi). Questi errori portano a un dataset inaccurato che necessita di tecniche avanzate di data cleaning, profondamente diverse in base al dataset preso in considerazione.

  In generale, gli errori si concretizzano in

-

  , ovvero oggetti con attributi i cui valori si discostano di molto rispetto a tutti gli altri.

- Si parla di dataset incompleti, invece, quando sono presenti degli oggetti con alcuni attributi non valorizzati. I motivi possono essere vari: l'attributo, erroneamente, non è stato salvato, o non ha significato per quel particolare oggetto, o al momento dell'archiviazione era stato considerato non rilevante. Gli attributi mancanti vengono chiamati *missing values* e sono gestiti in fase di data cleaning (Sezione 2.2.1).

- I dataset possono anche presentare inconsistenza. In questi casi, i dati inconsistenti tra di loro possono essere eliminati, essendo completamente inutilizzabili, se non dannosi ai fini dell'analisi.

In una prima fase di *data preprocessing*, vengono gestiti eventuali problemi di inaccuratezza, incompletezza e inconsistenza (Data Cleaning 2.2.1).

Spesso può essere necessario includere dati provenienti da diverse sorgenti per l'analisi. In questi casi è anche necessario integrare diversi database o file sulla base di attributi comuni, e successivamente bisogna gestire eventuali dati ridondanti causati dall'integrazione. In dataset prelevati dai sistemi di acquisizione reali, inoltre, può succedere che gli stessi attributi abbiano nomi differenti, poiché spesso in una stessa azienda vengono adoperati sistemi di acquisizione di marche differenti, che adoperano una propria codifica dei dati. In questi casi un'ulteriore fase di Data Cleaning è necessaria dopo la fase di integrazione (Data Integration 2.2.2).

Come già detto in precedenza, spesso i dataset prelevati da sistemi di acquisizione industriali sono di dimensioni molto elevate, poiché all'interno vengono archiviati la maggior parte dei dati misurabili. Utilizzando questi dataset, spesso integrando database provenienti da più sorgenti, ci si trova ad analizzare un'enorme mole di dati, che può rendere molto dispendiosa l'analisi in termini di risorse computazionali e tempo; a volte l'analisi può risultare del tutto irrealizzabile. Per questo, come ultima fase di preprocessing dei dati, è necessario ridurre la mole di dati da analizzare, cercando di mantenere esclusivamente i dati necessari. La riduzione può avvenire sia per cardinalità,che per dimensionalità (Sezione 2.2.3).

Un'altra tecnica di riduzione dei dati, consiste nel trasformare, ovvero elaborare i dati, per ottenere un dataset contenente le stesse informazioni, ma composto da meno dati. Questa fase, chiamata anche Data Transformation (Sezione 2.2.4), non è sempre possibile o necessaria, ma può ridurre le risorse richieste per il mining. Spesso è facile elaborare dati che esprimono dimensioni fisiche, correlate tra loro tramite relazioni notevoli.

Le tecniche di *data preprocessing*, quindi, sono tutte quelle tecniche che vengono applicate prima del *mining*, e che consentono di migliorare la qualità dell'analisi e/o il tempo necessario per il *mining*.

I metodi per *preprocessare* i dati sono raggruppati in 4 categorie che vengono qui elencate e in seguito approfondite: *data cleaning* (Sezione 2.2.1), *data integration* (Sezione 2.2.2), *data reduction* (Sezione 2.2.3), *data transformation* (Sezione 2.2.4).

2.2.1 Data cleaning

In questa fase, vengono gestiti i valori mancanti (*missing values*), gli outliers, e viene corretta l'eventuale inconsistenza dei dati. Più in generale viene migliorata la qualità del dataset.

I *missing values* sono tutti quei valori degli attributi mancanti perché erroneamente non sono stati salvati o semplicemente perché un attributo non ha significato per un determinato oggetto. Nel caso in cui un dataset presenti dei missing values, è necessario gestirli in uno dei seguenti modi:

1. **Eliminare l'oggetto**: ignorare completamente una specifica tupla che presenta uno o più attributi senza valore. In genere si elimina una tupla se l'attributo mancante ha un'importanza rilevante ai fini dell'analisi, oppure se gli attributi mancanti sono più di uno; in questo caso la tupla risulterebbe inutilizzabile;

2. **Inserire manualmente un valore**: in questo caso si può stimare un valore che rispetti la distribuzione dell'intero dataset, o inserire un valore costante, oppure ancora provare a prevedere il valore di quell'attributo ad esempio tramite l'utilizzo di alberi decisionali o regressione. Questo approccio è dispendioso in termini di tempo e non è efficiente in caso di dataset molto grandi;

3. **Ignorare l'attributo mancante**: nel caso in cui l'attributo non sia determinante ai fini dell'analisi, si può semplicemente ignorare;

4. **Sostituire con tutti i valori possibili**: al fine di mantenere la distribuzione della probabilità, si sostituisce il valore mancante con tutti i valori possibili, pesati in base alla loro rispettiva probabilità.

Gestire i missing values e gli outliers è estremamente importante ai fini di un buon risultato. La tecnica 2 è quella usata più di frequente; è necessario, però, comprendere bene la natura dei dati, in quanto spesso un attributo è privo di valore, semplicemente perché non ha significato per quello specifico oggetto. Personalmente, durante la fase di Data Cleaning del presente elaborato, ho preferito non considerare le tuple che presentavano missing values o outliers, poiché il numero di tuple a disposizione era molto elevato, e una predizione di tali valori sarebbe risultata troppo dispendiosa. In ogni caso un buon sistema di acquisizione, come nel caso in esame, può aiutare a ridurre al minimo questi errori, rendendo più facile la fase di Data Cleaning.

2.2.2 Data integration

I dati necessari per un'analisi, non sono sempre contenuti in un database unico. Può capitare che i sistemi di acquisizione archivino solo i dati di uno specifico processo. In questi casi è necessario integrare i dati mediante l'unione di più database, ovvero effettuare un'operazione di *Data Integration*.

L'integrazione di dati provenienti da sorgenti differenti, può produrre ridondanza e inconsistenza nel dataset. Questo perché l'informazione può risultare duplicata in database differenti, oppure può essere archiviata tramite codifica differente; infatti, una diversa unità di misura o l'utilizzo di codici differenti negli attributi nominali, deve essere gestita in questa fase.

Integrando più dataset, infine, si può ottenere un database con più informazioni di quelle necessarie. Anche in questo caso si tratta di ridondanza dei dati, rilevabile empiricamente tramite uno studio del significato di ogni singolo attributo, che può risultare subito irrilevante dal punto di vista dell'analisi, oppure analiticamente tramite un'analisi della correlazione tra gli attributi.

In particolare, l'indice di correlazione di Pearson, è un coefficiente che, dati due attributi x_i e x_j con valori continui, determina come il valore di un attributo varia in funzione dell'altro[12]:

$$r_\mathrm{ij} = correlation(x_\mathrm{i}, x_\mathrm{j}) = \frac{covariance(x_\mathrm{i}, x_\mathrm{j})}{s_\mathrm{i} s_\mathrm{j}}$$

Dove sono state utilizzate le seguenti definizioni:

$$covariance(x_\mathrm{i}, x_\mathrm{j}) = \frac{1}{n-1} \sum_{k=1}^{n} (x_\mathrm{k} - \overline{x})(y_\mathrm{k} - \overline{y})$$

Il coefficiente di correlazione varia in un intervallo tra -1 e 1. Una correlazione di 1 (-1) indica che x e y hanno una relazione lineare positiva (negativa). Calcolando, quindi, la matrice di correlazione, è possibile determinare gli attributi correlati e quelli non correlati, che possono essere scartati durante l'analisi.

2.2.3 Data reduction

In questa fase, viene manipolato il dataset in modo da ridurre la cardinalità e/o la dimensionalità dello stesso. La cardinalità di un dataset corrisponde al numero di tuple che esso contiene, mentre la dimensionalità indica il numero di attributi che descrivono ogni singola tupla.

Le tecniche di *Data Reduction* possono essere applicate, quindi, per ridurre il volume della rappresentazione dei dati, pur mantenendo l'integrità originale del dataset. In questo modo, l'applicazione di tecniche di *mining* sul dataset ridotto, sarà più efficiente, producendo risultati molto simili (o uguali).

Le principali tecniche di *data reduction* sono:

Aggregation

Spesso è possibile combinare due o più attributi (o oggetti) in un singolo attributo (o oggetto), aumentando così l'astrazione dei dati. L'aggregazione può essere *senza*

perdita, quando è possibile ricostruire i dati nella forma originale, o *con perdita* quando l'aggregazione elimina alcune informazioni necessarie per applicare il processo inverso e tornare ai dati di partenza. Tramite l'aggregazione si ottiene un dataset di dimensionalità ridotta poiché diminuisce il numero degli attributi, cambia la scala di dettaglio poiché aumenta l'astrazione del dataset, e si ottengono dei dati più stabili poiché dati aggregati hanno tendenzialmente una variabilità minore.

Sampling

Il campionamento è una delle tecniche più utilizzate per la riduzione dei dati. Viene impiegata sia in fase preliminare che in fase finale. Campionare è spesso indispensabile per ridurre la cardinalità del dataset poiché l'utilizzo dell'intera base di dati è costoso in termini di risorse e tempo.

Usare il dataset campionato in luogo di quello completo produrrà risultati simili se il campione è rappresentativo; un campione è rappresentativo se contiene le stesse proprietà della base di dati originale.

Le tipologie di campionamento utilizzabili sono prevalentemente 4:

- *Simple Random Sampling*: consiste in un campionamento casuale semplice; in questo modo ogni oggetto ha la stessa probabilità di essere selezionato;
- *Sampling without Replacement*: ogni volta che un elemento viene selezionato, viene rimosso dalla popolazione; in questo modo dopo ogni singola selezione, la probabilità cambia.
- *Sampling with Replacement*: a differenza della tecnica precedente, ogni volta che viene selezionato un oggetto, questo non viene rimosso dalla popolazione, lasciando inalterata la probabilità; in questo modo un oggetto può essere selezionato più volte; questa tecnica infatti, viene utilizzata per sovra-campionare classi poco probabili.
- *Stratified Sampling*: prima di campionare, i dati vengono suddivisi in partizioni, successivamente viene eseguito il campionamento casuale su ogni partizione. In questo modo si conserva la distribuzione originale, ma è possibile considerare un attributo per volta.

Un ulteriore vantaggio nell'utilizzo di tecniche di campionamento, è che il costo per ottenere un campione è proporzionale alla grandezza del campione stesso e non alla dimensione dell'intero dataset. Quindi la complessità del campionamento è sublineare nella grandezza del dataset[7].

Dimensionality Reduction

Con l'aumentare del numero di attributi per tupla, ovvero la dimensionalità del dataset, i dati diventano sempre più sparsi nello spazio che occupano; diventa sempre

più difficile, quindi, distinguere oggetti simili e non, perché il concetto di distanza diventa sempre meno significativo. Questo principio è detto anche "*Curse of Dimensionality*".

Ridurre la dimensionalità di un database, risulta quindi importante per: evitare la "maledizione della dimensionalità" precedentemente introdotta, ridurre la quantità di risorse e tempo necessari nell'analisi (analogamente al *Data Reduction*2.2.3), migliorare la visualizzazione dei dati stessi, eliminare il rumore dovuto alla presenza di attributi ininfluenti.

Una delle tecniche per ottenere *Dimensionality Reduction* è la cosiddetta *Feature Subset Selection*, ovvero la selezione di un gruppo di attributi ritenuti rilevanti ai fini dell'analisi. Per determinare quali attributi sono rilevanti e quali no, si può ricorrere ai seguenti metodi:

- *Brute-force approach*: si provano tutti i possibili insiemi di attributi come input all'algoritmo di data mining, fino a ottenere il risultato migliore;
- *Embedded approaches*: la selezione degli attributi è parte integrante dell'algoritmo di data mining;
- *Filter approach*: la selezione degli attributi avviene prima di eseguire l'algoritmo di data mining, tramite l'applicazione di un filtro;
- *Wrapper approaches*: si utilizza l'algoritmo di data mining come una *black box* per ottenere il migliore sottoinsieme di attributi.

2.2.4 Data transformation

L'ultima fase che completa il processo di Data Preprocessing, consiste nella trasformazione dei dati necessaria per applicare alcuni algoritmi di data mining, che accettano in input specifici tipi di dati, oppure semplicemente per migliorarne l'efficienza.

In questa sezione sono richiamate due tecniche principali: la creazione e la trasformazione di attributi.

Creazione di Attributi

Consiste nella creazione di nuovi attributi che possano evidenziare importanti informazioni in maniera più efficiente degli attributi originali.

Le principali tecniche sono:

- Estrazione di nuovi attributi da quelli originali;
- Mappatura dei dati in un nuovo spazio;
- Costruzione di nuovi attributi, ad esempio, tramite combinazione di attributi originali.

Trasformazione di Attributi

Consiste nell'applicazione di una funzione in grado di mappare l'intero set di valori di un attributo in un nuovo set di valori. Una delle tecniche maggiormente utilizzate per trasformare gli attributi è senza dubbio la **Normalizzazione**, tramite la quale è possibile ridurre l'intervallo di variazione di un attributo, in un intervallo predefinito; tramite normalizzazione gli attributi diventano più facilmente confrontabili.

Le principali formule utilizzabili per normalizzare i dati sono:

- Normalizzazione min-max: $\nu' = \frac{\nu - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$
- Normalizzazione Z-score: $\nu' = \frac{\nu - mean}{stand\_dev}$
- Decimal scaling: $\nu' = \frac{\nu}{10^j}$, dove$j$ è l'intero minore tale per cui $max(|\nu'|) < 1$.

Un'altra tecnica molto utilizzata in questa fase è la **Discretizzazione**, tramite la quale è possibile trasformare un attributo continuo in uno discreto, dividendo il dominio in un numero definito di intervalli. Le tecniche utilizzabili sono:

- Dividere il dominio in N intervalli della stessa grandezza $L = \frac{v_{max} - v_{min})}{N}$; questa tecnica semplice da implementare, è incrementale, ma risente della presenza di outliers e rumore nei dati.
- Dividere il dominio in N intervalli *quasi* della stessa grandezza; tecnica non incrementale, ma più robusta in presenza di outliers.
- Utilizzare tecniche di *clustering* per identificare gli intervalli ottimali.

2.3 *Supervised* e *Unsupervised* Machine Learning

La fase successiva alla preparazione dei dati è l'applicazione degli algoritmi di Mining. In questo elaborato vengono introdotti, in particolare, gli algoritmi di Machine Learning utilizzati per addestrare un modello predittivo. L'analisi predittiva è un sottoinsieme del Data Mining, utilizzata principalmente in ambito commerciale (un esempio è la profilazione degli utenti), e in ambito industriale e manifatturiero. Nel contesto dell'analisi predittiva, quindi, per Data Mining si intende la generazione di un modello predittivo tramite algoritmi e funzioni matematiche. Un modello predittivo è una funzione matematica in grado di rappresentare una relazione tra gli attributi degli oggetti; descrive come una o più variabili di un oggetto, sono correlate ad altre variabili. Gli algoritmi di Machine Learning possono essere suddivisi in due macro gruppi: *Supervised* e *Unsupervised*, in base al modo in cui gli algoritmi "apprendono" dai dati, per fornire poi una predizione.

Gli algoritmi di Supervised Machine Learning, sono i più comuni e includono algoritmi di regressione, classificazione e le SVM (Support Vector Machines). Il

Supervised learning (letteralmente apprendimento supervisionato) è chiamato così poiché l'utente si comporta come una guida per l'algoritmo, in quanto è consapevole di quali conclusioni si vogliono ottenere dall'analisi. Gli algoritmi *supervised*, infatti, richiedono che i possibili output siano conosciuti e che i dati usati per addestrare il modello siano già classificati correttamente.

Gli algoritmi *unsupervised*, invece, non richiedono una conoscenza iniziale dei possibili output, ma sono in grado di modellare la distribuzione dei dati in input e di classificarli, senza conoscere una classificazione iniziale "corretta". Gli algoritmi *unsupervised* sono utilizzati in ambiti riguardanti l'intelligenza artificiale e meno in contesti commerciali o manifatturieri. Alcuni esempi di algoritmi di *Unsupervised Machine Learning* sono gli algoritmi di Clustering e le Regole di associazione.

In alcuni casi può accadere di avere un grande dataset da analizzare in cui solo alcuni dati sono già classificati correttamente, mentre la maggior parte non presentano un *label*. In questi casi, si tratta di *Semi-supervised Machine Learning*.

2.4 Classificazione

In questa sezione verranno richiamati alcuni algoritmi di Classificazione, che sono stati impiegati nel mio progetto presso FCA. La classificazione è un tipo di data analisi utilizzato per estrarre dei modelli in grado di descrivere delle classi di dati. Il modello una volta sviluppato e validato, è in grado di predire la classificazione dei dati in categorie. I campi di applicazione degli algoritmi di classificazione sono molteplici e includono , il rilevamento di illeciti, profilazione utenti, predizione delle performance, industria e diagnosi mediche.

La classificazione dei dati è un processo composto da due step il primo di addestramento del modello e il secondo di applicazione del modello nella vera e propria classificazione. Anche il dataset viene diviso in due partizioni, la prima chiamata *training set* e la seconda *test set*.

Nel primo step, viene sviluppato il classificatore per descrivere un insieme di classi di dati o concetti. In questa fase chiamata anche *training phase*, un algoritmo di classificazione crea un classificatore analizzando il *training set* in cui ogni tupla è associata a un'etichetta di classe. Ogni tupla del set è rappresentata da un vettore di attributi n-dimensionale che si assume appartenga a una specifica classe che include insiemi di tuple "simili". La classe è determinata da un attributo etichetta che è discreto, e ogni valore corrisponde a una classe. Poiché l'etichetta è conosciuta in partenza, questa prima fase è un esempio di *supervised learning*. Alla fine della *training phase* viene prodotta una funzione $y = f(X)$ in grado di associare una classe y data una tupla X in input.

Nel secondo step, il modello viene utilizzato per classificare dati senza etichetta. Prima di utilizzare il modello, però, è necessario stimare l'accuratezza dello stesso. Utilizzare il training set per testare l'accuratezza del modello sarebbe errato, poiché

sarà perfettamente adattato a tali dati, risultando così quasi perfetto nella predizione (durante la fase di addestramento, infatti, il modello potrebbe incorporare anche particolari anomalie presenti nel set di addestramento che non sono, però, caratteristiche generali di tutti i set di dati). È necessario, quindi, utilizzare un set differente e indipendente dal primo. In questa fase viene utilizzato un insieme di tuple di cui si conosce la classe, ma questa non viene fornita al modello come nella prima fase; dopo che il modello effettua la previsione delle classi di ogni tupla, la classe risultante viene confrontata con la classe reale della tupla.

2.5 Validazione del modello

Generalmente non esiste un algoritmo migliore di altri; tutto dipende dai dati che stiamo analizzando. Per questo, spesso vengono impiegati diversi algoritmi per addestrare altrettanti modelli predittivi, successivamente i modelli vengono testati e validati tramite misure specifiche sui risultati della predizione. In questa sezione vengono introdotte le definizioni di alcune misure utilizzate nel presente elaborato.

Come accennato nella precedente sezione, i dati etichettati vengono divisi in due set: uno di addestramento e uno di test. Le tecniche utilizzate per partizionare il set a disposizione sono prevalentemente tre:

- **Holdout**: consiste in un partizionamento fisso, tale per cui 2/3 del dataset originale vengono utilizzati per la fase di training, mentre il restante 1/3 viene impiegato per la fase di test. È un metodo meno affidabile di altri, ma preferibile in casi di dataset molto grandi.

- **Cross validation**: tramite questo metodo, i dati vengono partizionati in k sottoinsiemi chiamati *fold*. Il modello usa $k-1$ partizioni per la fase di training e la restante partizione per la fase di test. La validazione viene poi ripetuta per tutte le partizioni. La stima risultante da questa tipologia di partizionamento è particolarmente affidabile, ma non è utilizzabile in dataset molto grandi.

- **Leave-one-out**: l'intero dataset viene utilizzato per la fase di training, lasciando solo un oggetto per la fase di testing. Corrisponde a un Cross Validation con $k = 1$. È un metodo utilizzabile per dataset abbastanza piccoli.

Dopo aver partizionato il dataset di partenza, addestrato e validato il modello, si possono ricavare alcune misure che descrivono la qualità del modello risultante. L'**accuratezza** è una delle misure principali per valutare il modello. Partendo dalla matrice di confusione in figura, l'accuratezza si calcola come segue.

$$Accuracy = \frac{Numero\_di\_oggetti\_correttamente\_classificati}{Numero\_di\_oggetti\_classificati} = \frac{TP + TN}{TP + TN + FP + FN}$$

		Classe predetta	
	Classe	Yes	No
Classe reale	Yes	TP (True positive)	FN (False negative)
	No	FP (False positive)	TN (True negative)

Figura 2.1. Matrice di confusione

La misura di accuratezza ha, però, dei limiti. Nel caso in cui il dataset sia fortemente sbilanciato, infatti, il modello risulterà eccessivamente adattato alla classe maggiormente rappresentata, che assegnerà sempre la stessa classe, qualsiasi sia la tupla in input. Se si prova a calcolare l'accuratezza di un modello del genere, si otterrà un valore vicino al 100%, ma in realtà la qualità generale del modello è molto più bassa poiché nessuna tupla appartenente alla classe con minore cardinalità verrà identificata.

Per questo motivo, valutare un modello solo sulla base dell'accuratezza non è sufficiente. Altre misure specifiche per classe sono:

- **Recall**: indica la percentuale di oggetti appartenenti a una specifica classe correttamente identificati.

$$Recall = \frac{Numero\_di\_oggetti\_correttamente\_assegnati\_a\_C}{Numero\_di\_oggetti\_appartenenti\_a\_C}$$

- **Precision**: indica la percentuale di oggetti correttamente assegnati a una classe.

$$Precision = \frac{Numero\_di\_oggetti\_correttamente\_assegnati\_a\_C}{Numero\_di\_oggetti\_assegnati\_a\_C}$$

- **F-Measure**: la media armonica tra Recall e Precision.

$$F - measure(F) = \frac{2rp}{r+p}$$

2.6 Modelli utilizzati

Al fine di studiare le potenzialità dell'applicazione di tecniche di Data Mining al caso in esame, sono stati validati modelli implementati con diverse configurazioni e confrontate le performance di ognuno. Di seguito verranno richiamati gli algoritmi di Machine Learning utilizzati nei capitoli successivi [7].

Decision Tree

Un albero decisionale è una struttura ad albero simile ad un flow chart, dove ogni nodo interno rappresenta un test su un determinato attributo, ogni ramo rappresenta un possibile risultato del test, mentre i nodi foglia (nodo terminale) determina una etichetta. Il nodo iniziale è chiamato radice dell'albero.

I principali algoritmi utilizzati per costruire un albero decisionale sono: algoritmo di Hunt, CART, ID3, C4.5, C5.0, SLIQ, SPRINT. L'algoritmo implementato nell'operatore di Rapid Miner utilizzato in seguito, ad esempio, è stato implementato tramite l'utilizzo dell'algoritmo C4.5 con lievi modifiche.

La maggior parte degli algoritmi adottano una strategia *greedy* costruendo l'albero decisionale in maniera ricorsiva, individuando di volta in volta il punto più conveniente per effettuare uno split. La parte più importante durante la costruzione di un albero decisionale è la scelta del punto in cui effettuare lo split; in base al tipo dell'attributo, lo split può essere binario (in questo caso è necessario scegliere bene il punto in cui partizionare il dominio) o multipli (per attributi categorici o in caso di partizioni multiple). La scelta del punto di split è guidata da misure di impurità; la misura può basarsi su diversi indici, che vengono di seguito riassunti:

- Indice GINI per un nodo t: $GINI(t) = 1 - \sum_j [p(j\backslash t)]^2$
  dove $p(j\backslash t)$ è la frequenza relativa della classe j al nodo t
- Entropia per un nodo t: $Entropia(t) = -\sum_j p(j\backslash t) \log p(j\backslash t)$
  dove $p(j\backslash t)$ è la frequenza relativa della classe j al nodo t
- Errore di classificazione per un nodo t: $Errore(t) = 1 - \max P(i\backslash t)$
  dove $p(j\backslash t)$ è la frequenza relativa della classe j al nodo t

L'elenco non vuole essere esaustivo di tutti gli indici di misura di impurità, ma ne esistono molti altri non citati per brevità.

Gli alberi decisionali sono soggetti al fenomeno di overfitting, che si verifica, in generale, quando un sistema di classificazione crea un modello che si adatta ai dati utilizzati in fase di training utilizzando un numero eccessivo di parametri, nel caso degli alberi decisionali, utilizzando un numero eccessivo di nodi. Il modello risultante, quindi, sarà troppo specializzato sul set di addestramento, ma poco generalizzabile, quindi inutilizzabile. Per evitare il verificarsi del fenomeno di overfitting, e quindi di far crescere eccessivamente l'albero decisionale in fase di addestramento, possono essere utilizzate due tecniche:

1. **Pre-pruning:** tecnica che consiste nel fermare lo split del dataset quando lo split di un nodo risulta minore di una soglia predefinita, tipicamente vicinaal 5% della cardinalità del dataset iniziale; in questo modo si evita di prendere decisioni troppo specifiche su un set troppo piccolo di dati, che potrebbero essere dovute ad anomalie di quel preciso set e non su caratteristiche generalizzabili;

2. **Post-pruning:** tecnica che consiste nel costruire l'albero per intero e tagliare i rami non necessari solo in seguito, sostituendo la parte di albero tagliata con un nodo foglia (etichetta), cercando di massimizzare le prestazioni, minimizzando la complessità del modello risultante.

K-Nearest-Neighbors

L'algoritmo di K-NN non crea un modello vero e proprio, ma misura la distanza tra gli elementi del training set. Per classificare una tupla, viene misurata la distanza di quello specifico oggetto rispetto a tutti gli oggetti del training set e viene assegnata l'etichetta corrispondente ai k oggetti più vicini (nearest neighbors), nel caso in cui gli oggetti vicini appartengano a classi differenti, si può scegliere se dedurre la classe per maggioranza o per media pesata secondo la distanza. L'algoritmo K-NN necessita di una configurazione iniziale, ovvero:

- Training set di partenza, che non viene utilizzato come un vero e proprio set di addestramento del modello, ma come set di riferimento per la classificazione di oggetti sconosciuti;

- Parametro k che identifica il numero di oggetti più vicini da identificare durante il processo di classificazione; la scelta di questo parametro è importante perchè un valore troppo basso causa una elevata sensibilità al rumore e agli outliers, mentre un valore troppo elevato rischia di includere punti appartenenti ad altre classi;

- Tipologia di misura della distanza da utilizzare nel calcolo.

L'algoritmo K-NN necessita di una normalizzazione iniziale degli attributi, poiché è basato sul calcolo di una distanza; è più lento di altri algoritmi durante la classificazione e risente particolarmente di dataset con dimensionalità elevata.

Neural Network

Il modello a reti neurali è chiamato in questo modo perché ispirato alla struttura del cervello umano. In analogia si neuroni come unità di elaborazione e le sinapsi come rete di connessione, una rete neurale è costituita da un insieme di nodi decisionali collegati tra loro in cui ogni connessione è associata ad un peso diverso. Durante la fase di apprendimento, il modello tara il peso associato ad ogni connessione in modo da fornire una precisione accurata nella classificazione delle tuple in ingresso.

I nodi di una rete neurale possono essere di tre tipi:

1. Nodo di input;
2. Nodo intermedio;
3. Nodo di output.

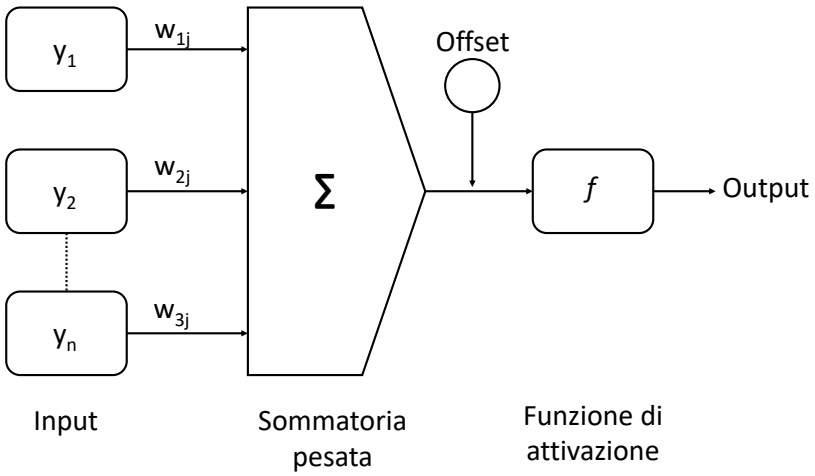


Figura 2.2. Struttura di un neurone

Ogni nodo è definito da un vettore di pesi, un valore di offset e una funzione di attivazione, utilizzati in fase decisionale. L'algoritmo di costruzione più utilizzato è quello di retropropagazione e viene di seguito riassunto:

- vengono assegnati dei valori di peso e offset casuali ad ogni nodo;
- per ogni nodo viene calcolato il risultato considerando i pesi, il valore di offset e della funzione di attivazione per ogni oggetto;
- propagazione fino ad ottenere un valore di output;
- confronto del valore ottenuto con quello desiderato e valutazione dell'errore ottenuto;
- propagazione in senso inverso a quello delle connessioni della rete aggiornando la stima di pesi e offset di ogni nodo;

Il processo termina quando si ottiene un'accuratezza accettabile, oppure si ottiene una percentuale di attributi con un errore minore di una certa soglia, oppure si è raggiunto un numero massimo di epoche settato all'inizio per evitare di ottenere una fase di training eccessivamente lunga.

Il modello a rete neurale può risultare molto lento in fase di apprendimento e produce un classificatore scarsamente interpretabile rispetto ad altri algoritmi, ma consente di ottenere risultati altamente accurati, essendo tollerante a rumore e outliers, è efficiente in fase di classificazione ed può essere applicato anche a problemi non lineari.

Capitolo 3

Contesto di Lavoro

Il presente capitolo introduce brevemente il contesto di lavoro presso il quale ho svolto il progetto in esame. Il primo passo in un lavoro di analisi di dati, infatti, consiste nello studio dell'ambiente di lavoro e degli obiettivi che l'azienda vuole ottenere dall'analisi dei dati(Step 1 di KDD 1.3). Spesso, come in questo caso, è necessaria un'analisi preliminare in grado di determinare vantaggi economici e di efficienza che si possono raggiungere e solo successivamente si procede con l'utilizzo di tecniche di Data Mining direttamente sul processo produttivo. Inoltre, nella maggior parte delle grandi industrie, come ad esempio FCA, i processi produttivi sono frutto di anni di esperienza e di lavoro consolidati; una modifica a tali processi, seppur minima, deve essere analizzata e studiata nello specifico prima di una sua possibile introduzione.

3.1 Fasi di produzione

Il mio lavoro si è svolto presso l'ufficio ICT del *plant* di Mirafiori a Torino. In questa struttura, nel periodo in cui si è svolto il mio tirocinio, venivano prodotti i modelli Levante di Maserati, e Mito di Alfa Romeo. In questo stabilimento, il processo produttivo si suddivide in 4 grandi unità: Lastratura, Verniciatura, Montaggio e Qualità.

- L'unità di Lastratura è la più automatizzata, sono presenti numerosi robot e stazioni automatiche di saldatura, rivettatura e spalmatura. Poiché la scocca è costituita da telaio in acciaio e parti mobili in alluminio, sono impiegate anche tecniche avanzate come la rivettatura Tucker. Con un tempo ciclo[1] di circa 6

[1]Per tempo ciclo si intende un intervallo temporale prefissato, durante il quale gli operatori possono effettuare una qualsiasi operazione. Allo scadere di questo intervallo, la scocca procede nella linea di produzione.

minuti, la lastratura è in grado di comporre le scocche partendo dalle lamiere stampate impiegando circa 2300 punti di saldatura, sottoposti a numerosi controlli geometrici automatizzati.

- L'unità di Verniciatura consta di 4 aree, in ognuna delle quali la superficie della scocca riceve un trattamento specifico (es. anticorrosione, sigillatura, spruzzatura dello smalto e applicazione di resina trasparente) e ovviamente assume il colore richiesto dall'ordine. In questa unità, inoltre, viene applicato alla scocca il CIS (Codice Identificativo Scocca): un codice univoco di 7 cifre più una di controllo, che identifica le caratteristiche che dovrà avere quella particolare automobile in base alle richieste dell'ordine. Da questo punto in avanti, ogni operatore della linea, scansionando il codice a barre corrispondente al CIS, applicato su ogni scocca, conoscerà le precise operazioni da eseguire su quella unità in base alle caratteristiche richieste dal cliente.

- L'unità di Montaggio si suddivide in tre aree principali: *trim, chassis, final*. Essa è prevalentemente manuale, ma assistita da tecnologie avanzate. Grazie alla supervisione del sistema MES, questa unità è in grado di produrre tutte le personalizzazioni richieste su un'unica linea di produzione. La linea è composta da zone ad avanzamento continuo e altre ad avanzamento a step, in queste ultime l'operatore ha a disposizione un certo tempo ciclo per completare le operazioni necessarie che, per il modello Levante, in genere è di 6 minuti.

- L'unità di Qualità è ramificata su tutta la linea produttiva ed esegue controlli di vario genere in più fasi della produzione. L'unità è suddivisa in Processo, Prodotto, Forniture e Diagnosi ed esegue controlli di natura estetica, funzionale e statistica. La scocca ad esempio, subisce un controllo a campione sulla qualità delle saldature, che verrà approfondito nella sezione 3.3. Ogni vettura attraversa 5 *gate* di delibera, per essere in fine sottoposta al Test Dinamico Funzionale, durante il quale viene testata su pista e su strada.

Durante la fase di montaggio, gli operatori interagiscono col sistema tramite gli *OT (Operator Terminal)*, sui quali vengono visualizzate le operazioni da effettuare in base al *CIS* che passa sulla linea. Interagendo con l'*OT* l'operatore può certificare le operazioni eseguite. Le certificazioni sono poi archiviate dal *MES* congiuntamente al *CIS*. Lungo tutta la linea di produzione sono presenti dei monitor chiamati *Hands-On*, sui quali vengono visualizzate varie informazioni riguardo il processo, come ad esempio le automobili precedenti e successive a quel preciso punto della linea, le automobili già processate in quel turno, eccetera. Tramite gli OT, inoltre, è possibile segnalare anomalie durate il processo produttivo che verranno inoltrate direttamente allo smartwatch personale del team leader appropriato al tratto di linea, il quale potrà intervenire, eventualmente generando un fermo. Sugli *Hands-On*, inoltre, viene visualizzato il numero di automobili da completare entro la fine del turno, secondo una stima effettuata dal sistema per un'efficienza ottimale.

3.2 MES

Un'azienda che mira a essere competitiva a livello globale, oggigiorno, deve necessariamente raggiungere un'efficienza molto elevata nel processo produttivo (WCM 1.1). A tal proposito, è necessario implementare logiche e strumenti che consentano all'azienda di monitorare e ottimizzare le diverse attività relative alla produzione e alla manutenzione degli impianti.

L'avvento dell'*Industria 4.0* ha determinato la fine dei sistemi centralizzati tradizionali per il controllo della produzione, strettamente correlati alla linea produttiva e quindi statici. Infatti, un'industria composta da elementi produttivi autonomi è intrinsecamente decentralizzata. L'introduzione di nuove tecnologie ha consentito la creazione di un nuovo ambiente altamente efficiente e dinamico in grado di adattarsi per soddisfare le richieste sempre più personalizzate dei clienti. Per supportare questo cambiamento di paradigma, il sistema *Manufacturing Execution System* (MES) deve necessariamente essere riadattato sulla base delle nuove esigenze[1].

3.2.1 MES: Manufacturing Execution System

L'acronimo *MES (Manufacturing Execution System)*, indica un sistema utilizzato dalle aziende manifatturiere, in grado di controllare e gestire tutte le attività legate alla produzione industriale. Lo scopo di un sistema MES, in un'azienda, è quello di permettere lo scambio di informazioni tra il livello gestionale, noto anche come *livello 4*, dove i sistemi utilizzati sono generalmente ERP (Enterprise Resource Planning), e il livello produttivo, noto anche come *livello 2*, nel quale vengono utilizzati prevalentemente sistemi *PLC (Programmable Logic Controller)* e *SCADA (Supervisory Control And Data Acquisition)* per la gestione e il controllo dei segnali digitali e analogici di sensori e attuatori di un impianto industriale. Il MES si colloca quindi in un livello intermedio, noto anche come livello 3, ed è in grado di gestire e controllare tutte le operazioni dell'ambiente industriale, e di controllarne tutte le relative interfacce.

Secondo il modello *MESA (Manufacturing Enterprise Solutions Association)*, le funzionalità di un sistema MES sono le seguenti:

- **Pianificazione della produzione:** gestione di un insieme di operazioni da effettuare per soddisfare i requisiti di produzione, generalmente ricevuti da sistemi di livello superiore, di tipo ERP. Sono incluse tutte le attività di approvvigionamento e ottimizzazione delle risorse necessarie.

- **Allocazione e stato delle risorse:** gestione delle risorse intese come macchinari, materiali, personale, strumenti. Il MES è in grado anche di fornire informazioni in tempo reale sullo stato delle risorse e gestire l'allocazione e la prenotazione delle stesse.

- **Assegnazione delle unità di produzione:** gestione del flusso delle unità produttive in termini di istruzioni di lavoro. Gli ordini di assegnazione vengono presentati nell'ordine in cui devono essere effettuate le operazioni, ma possono cambiare in tempo reale. Inoltre, vengono fornite informazioni riguardo il lavoro effettuato fino a un certo momento, e quello ancora da effettuare, tramite una logica a *buffer*.

- **Controllo dei documenti di produzione:** gestione e distribuzione delle informazioni su unità produttive, prodotti, processi, progetti e ordini. Tutto viene archiviato sotto forma di storico, insieme alla documentazione relativa alle *policy*.

- **Raccolta dati:** dati relativi a processi, materiali e operazioni svolte da personale, macchine o certificazioni vengono raccolti e archiviati tramite interfacce, talvolta direttamente dalla linea produttiva in maniera manuale o automatica.

- **Gestione della manodopera:** gestione dello stato del personale, inclusi turni e presenze; gestione indiretta delle risorse tramite la preparazione dei materiali o degli strumenti da parte del personale.

- **Controllo qualità:** analisi in tempo reale delle misurazioni effettuate sulla produzione per assicurare un controllo sulla qualità del prodotto e l'efficienza dell'impianto, per identificare eventuali problemi che richiedono un'attenzione particolare da parte di personale specializzato come ad esempio manutenzione, o guasti IT.

- **Controllo del processo:** gestione del flusso di lavoro dell'impianto. Particolare attenzione alle attività di produzione pianificate ed effettuate.

- **Gestione della manutenzione:** monitoraggio ed esecuzione delle attività relative alla manutenzione delle apparecchiature e degli strumenti dell'impianto. La manutenzione viene organizzata in maniera periodica o preventiva, oppure può essere effettuata immediatamente in risposta a guasti. Il sistema generalmente mantiene uno storico degli interventi effettuati per facilitare la diagnosi dei problemi.

- **Tracciamento e genealogia del prodotto:** il prodotto viene monitorato durante il suo percorso lungo la linea produttiva, spesso gli viene associato un codice univoco in modo da distinguerlo da tutti gli altri. In ogni momento il sistema è in grado di determinare dove si trova un determinato prodotto, chi ci sta lavorando, quale operazione viene effettuata. Anche in questo caso viene costruito uno storico in modo da conoscere il percorso che ha effettuato un prodotto, anche alla fine del suo ciclo produttivo.

- **Analisi delle prestazioni:** il sistema fornisce un confronto delle prestazioni attuali dell'impianto, con gli obiettivi stabiliti dal cliente o dall'azienda sulla base di dati storici. Il risultato delle prestazioni dipende da misure sull'utilizzo ottimale delle risorse, la disponibilità di materiale o il tempo ciclo delle varie operazioni. In questa fase, per misurare le prestazioni, vengono calcolati coefficienti come ad esempio l'*OEE (Overall Equipment Effectiveness)* (Capitolo 4), consultabili dal personale competente che è successivamente in grado di valutare l'effettiva efficienza dell'impianto.

Il *MES*, quindi, è un sistema che organizza, distribuisce, e monitora i macchinari e le informazioni a essi relative, a livello di fabbrica. Simultaneamente, è in grado di fornire importanti misurazioni relative all'efficienza e alle prestazioni del processo produttivo, effettuando le relative statistiche che possono essere consultate da personale specializzato.

3.3 Controllo qualità

In un plant di produzione di autovetture appartenenti ad un settore di mercato *luxury*, la qualità del prodotto è, ovviamente, un aspetto molto importante. I controlli di qualità a cui vengono sottoposti i veicoli sono molteplici e terminano con un test su pista e su strada per ogni automobile prodotta.

Anche in questa fase molti dati vengono prodotti, e spesso raccolti e archiviati in basi di dati. L'integrazione di questi dati con quelli prodotti da altre unità del plant, non è sempre immediata e spesso l'analisi dei dati risulta esclusivamente un'analisi statistica.

Lungo tutta la linea produttiva, l'unità di Qualità preleva casualmente delle scocche da sottoporre a dei controlli; le unità che non presentano difetti, o presentano difetti trascurabili entro certe soglie, vengono reintrodotte nella linea produttiva, mentre le unità che presentano difetti gravi, tornano indietro nella linea per subire interventi di riparazione.

Il progetto in esame, approfondito nel capitolo seguente, si colloca tra la produzione e il controllo qualità, e mira a diminuire la necessità di effettuare più volte una stessa operazione a causa di un risultato di scarsa qualità, colmando almeno in parte la mancanza di un collegamento tra la l'unità di produzione e manutenzione, e quella del controllo qualità, causa di una diminuzione dell'efficienza dell'intero plant.

Capitolo 4

Caso di studio

Nel capitolo precedente è stata introdotta la struttura della linea di produzione del plant di Mirafiori. In questo capitolo verrà illustrato il progetto in esame, concentrando l'analisi prevalentemente sulle unità di Lastratura (3.1) e Qualità (3.1).

Normalmente i dati raccolti dai sensori delle macchine saldatrici, sono analizzati dal team della Sala Metrologica, per studiare la frequenza e le possibili cause di eventuali fermate o micro-fermate durante i turni di lavoro. I dati prodotti dai controlli sulla qualità, invece, sono analizzati dal team dell'unità Qualità, a fini statistici; tali analisi vengono presentate al team della Manutenzione, che valuterà solo a posteriori se tarare le soglie del CBM (Condition Based Maintenance) ovvero le soglie di lavoro oltre le quali un macchinario necessita di manutenzione. Poiché l'analisi sui dati della qualità delle saldature viene effettuata alla fine del processo, l'efficienza stimata della linea produttiva, calcolata in termini di OEE (Overall Equipment Effectiveness) (Figura 4.1)[1], spesso non corrisponde all'efficienza effettiva, perché non sono considerate tutte le unità (scocche o parti mobili) che, a causa dei vincoli di qualità molto stringenti, necessitano di un intervento aggiuntivo.

Il progetto proposto consiste nell'implementazione di uno strumento di data analisi, nello specifico un classificatore, in grado di fornire una previsione sulla qualità di una saldatura effettuata da un macchinario, analizzando i suoi parametri di lavoro monitorati dai sensori. I dati sono stati prelevati dalle due unità sopracitate. La prima parte del progetto, infatti, consiste nell'integrazione dei dati provenienti da queste due unità. Successivamente, è stato costruito uno storico archiviando i dati quotidianamente, in modo da ottenere un dataset abbastanza grande con cui è stato addestrato e validato il modello per testarne la qualità.

[1]Il coefficiente OEE (Overall Equipment Effectiveness) esprime il valore di efficienza di una linea e delle sue componenti, i dati prodotti possono essere utili per comprendere perché si verificano delle efficienze e le condizioni di lavoro delle macchine.

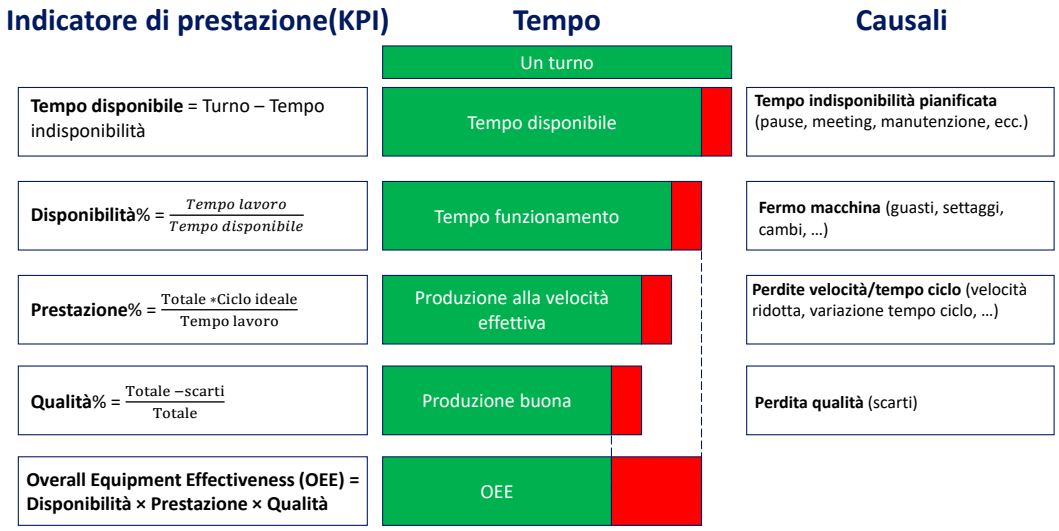


Figura 4.1. Calcolo OEE (Overall Equipment Effectiveness)

4.1 Dati estratti dall'unità Lastratura

Come accennato alla fine del capitolo precedente (Capitolo 3), l'unità di Lastratura effettua operazioni di saldatura e rivettatura su scocche e parti mobili. Le operazioni sono completamente automatizzate e le macchine saldatrici sono monitorate da numerosi sensori e gestite da PLC. Le macchine saldatrici sono in grado di determinare il successo o l'insuccesso di un'operazione, inteso come esito della saldatura e codificato con "OK" o "KO"[2], ma non sono in grado di determinare la qualità della saldatura (o rivettatura) effettuata. I dati, raccolti dai PLC, e aggregati da elaboratori appositamente collocati in vari punti della linea, vengono poi inviati al MES per essere archiviati in uno storico giornaliero.

I dati prelevati dal MES o direttamente dai PLC, sono raccolti in forma relazionale, dove ogni tupla rappresenta un'operazione di saldatura effettuata da una specifica macchina su uno specifico punto, entrambi identificati da codici univoci. Per ogni tupla vengono archiviate le informazioni riguardanti il punto di saldatura (*spot*) associate a uno specifico CIS [3](Capitolo 3) o al codice identificativo del lotto di parti mobili. Per ogni punto vengono archiviati l'ID e il Programma: in base alla tipologia del punto, infatti, esiste un programma preimpostato che configura

[2]Nel caso in cui una macchina rilevi un esito negativo di un'operazione di saldatura, può ripeterla senza provocare un fermo macchina.

[3]CIS (Codice Identificativo Scocca).

Nome variabile	Spiegazione	Unità di misura
protRecord_ID	numero progressivo dei record	numero
dateTime	data	data
progNo	numero del programma	numero
wear	consumo dell'elettrodo	numero
wearPerCent	consumo percentuale	%
iDemandStd	corrente richiesta standard	Ampere
ilsts	corrente Ists	Ampere
phaStd	fase del circuito standard	rad
pha2	fase 2	rad
tipDressCounter	contatore cambio elettrodo	numero
voltageActualValue	tensione	Volt
voltageRefValue	tensione di riferimento	Volt
currentActualValue	corrente	Ampere
currentReferenceValue	corrente di riferimento	Ampere
energyActualValue	energia circuito	Joule
energyRefValue	energia di riferimento	Joule
powerActualValue	potenza del circuito	Watt
powerRefValue	potenza di riferimento	Watt
resistanceActualValue	valore resistenza	mOhm
resistanceRefValue	resistenza di riferimento	mOhm
pulseWidthActualValue	larghezza dell'impulso	Hz
pulseWidthRefValue	larghezza impulso di riferimento	Hz

Figura 4.2. Tipologia dei dati estratti dall'unità di Lastratura

la macchina con dei valori di riferimento tarati appositamente per quello specifico punto; ogni parametro di lavoro, invece, viene archiviato tramite una coppia di valori: uno di riferimento che è quello impostato dal programma di lavoro e uno effettivo che è il valore fisico misurato dal sensore.

4.2 Dati estratti dall'unità Qualità

L'unità di Qualità preleva ogni giorno dalla linea, in modo casuale, alcune unità da sottoporre a stringenti controlli di vario genere (Sezione 3.1); in particolare dall'unità di Lastratura, vengono prelevate delle scocche e alcuni banchi di parti mobili

[4]. Ogni unità prelevata, viene sottoposta a un controllo effettuato da personale specializzato, con l'ausilio di una sonda ad ultrasuoni collegata a un computer sul quale è installato un software di rilevazione. La sonda viene messa a contatto con ogni punto di saldatura e tramite la misura di alcuni parametri, stabilisce la qualità dell'operazione effettuata. Il risultato della scansione viene codificato con un valore numerico da 0 a 17, in cui il valore 0 indica una saldatura di buona qualità, mentre gli altri valori indicano un difetto. Una saldatura di buona qualità, viene automaticamente convalidata dallo strumento, alcuni esiti negativi possono comunque essere validati manualmente dall'operatore se considerati difetti non bloccanti[5], possibilmente dovuti a errori di misurazione. Nella figura 4.3 un estratto del database del sensore ad ultrasuoni, contenente i possibili valori dell'esito del controllo, codificati.

CatNo	EnglishText	LocalText	ValidManua
0	Buono	Buono	☐
1	Bruciato	Bruciato	☑
2	Non Saldato	Non Saldato	☑
3	Sottile	Sottile	☑
4	Freddo+Sottile	Freddo+Sottile	☑
5	Freddo	Freddo	☑
6	Scarto	Scarto	☐
7	Scarto	Scarto	☑
8	Scarto 2	Scarto 2	☐
9	Nocciolo Picco	Nocciolo Picco	☑
10	Difetto Interno	Difetto Interno	☑
11	Defettoso	Defettoso	☑
12	Piccolo	Piccolo	☑
13	Ripetere	Ripetere	☑
14	Omesso	Omesso	☐
15	Causa cliente	Causa cliente	☑
16	Causa cliente	Causa cliente	☑
17	Causa cliente	Causa cliente	☑

Figura 4.3. Codifica dei possibili risultati del controllo tramite sonda a ultrasuoni

[4]Per parti mobili si intendono tutti gli elementi removibili della scocca, come ad esempio le porte, che in Lastratura vengono gestiti in banchi o lotti.

[5]Un difetto è considerato bloccante quando è tale da non consentire la continuazione nella linea produttiva, questo causa un fermo della produzione che deve essere convalidato dal personale responsabile (team leader).

4.3 Integrazione e analisi

I dati prelevati dalle due unità sono stati integrati tramite una join sul codice identificativo e l'esito della qualità della saldatura è stato codificato in un valore binario, concentrando i valori da 1 a 17 in un unico valore che identifica una saldatura di scarsa qualità. In questo modo è stato costruito un database contenente uno storico di saldature effettuate, l'esito del controllo della qualità tramite sonda ad ultrasuoni, in relazione ai parametri di lavoro della saldatrice al momento in cui ha effettuato quello specifico punto di saldatura.

Il database risultante, è stato diviso in due set distinti: il primo (training set) è stato utilizzato per addestrare un modello predittivo (training phase), mentre il secondo (test set), privato dell'attributo riguardante l'esito del controllo (label), è stato utilizzato per validare il modello (testing phase). Il classificatore è stato sviluppato utilizzando diversi algoritmi, e nella fase di validazione, le prestazioni dei vari algoritmi sono state confrontate per determinare l'approccio migliore.

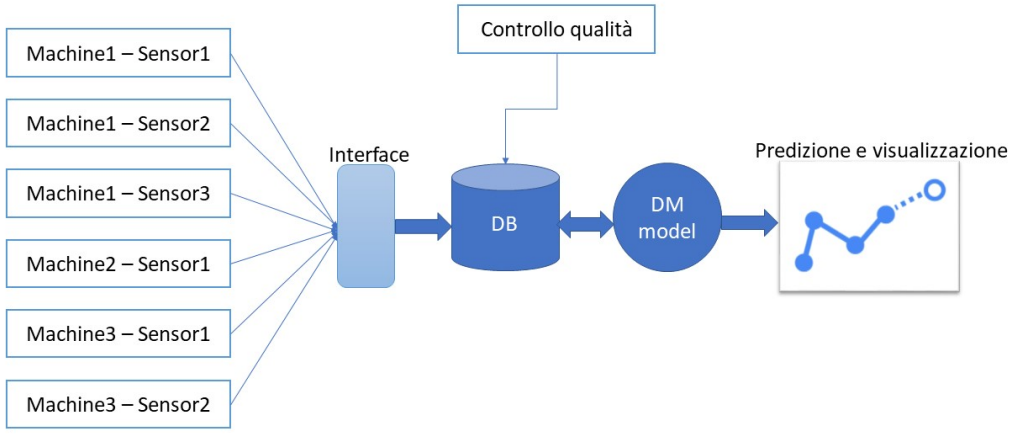


Figura 4.4. Integrazione dei dati e sviluppo del classificatore

L'analisi è stata effettuata su uno storico di dati creato manualmente, prelevando i dati quotidianamente. Congiuntamente al modello, però, sono stati implementati degli script in grado di prelevare automaticamente i dati dai computer della linea produttiva e inviarli su server condivisi, in modo da consentire un'analisi in tempo reale dei dati. Il risultato dell'analisi viene presentato agli addetti alla manutenzione, che saranno in grado di stabilire l'efficienza reale delle macchine ed eventualmente pianificare interventi straordinari di manutenzione o utilizzare

queste informazioni per tarare in modo più efficace le soglie CBM (Condition Based Maintenance) passando da un approccio reattivo, a uno proattivo riguardo la manutenzione delle macchine. Alcuni dei parametri di lavoro monitorati sono risultati altamente correlati con la qualità risultante della saldatura; di conseguenza i risultati analizzati nel capitolo successivo, possono essere considerati affidabili.

4.4 Gestione del dataset fortemente sbilanciato

Il dataset derivante dalla fase di integrazione e preprocessing, come spesso accade in casi di analisi dei dati prelevati da sistemi industriali, è risultato fortemente sbilanciato. Le due classi prese in considerazione, infatti, si riferiscono all'esito di un'operazione della linea produttiva altamente efficiente, per questo motivo il rapporto tra le due classi è stato verificato essere prossimo a $20 : 1$.

Analizzare un dataset così distribuito può portare a risultati scadenti nonostante valori elevati di misure come l'accuratezza. Calcolando l'accuratezza di un modello addestrato con un dataset sbilanciato, infatti, si otterrebbe un valore molto vicino al 100%, ma un modello non performante. Questo accade perché il modello tenderà ad assegnare la classe più probabile a tutti gli oggetti, per cercare di ottenere un'accuratezza elevata. Nel caso in questione, ad esempio, il modello sviluppato inizialmente raggiungeva un'accuratezza di circa il 95%, ma una percentuale di saldature di scarsa qualità identificate molto vicina allo 0%.

Per gestire casi di questo genere, possono essere adottate alcune strategie di seguito elencate:

- In alcuni casi è possibile che il rapporto sproporzionato sia dovuto semplicemente ad un errore in fase di raccolta dati, quindi è sufficiente ripetere l'operazione di acquisizione iniziale;
- La misura dell'accuratezza, come spiegato in precedenza, non è un valore affidabile singolarmente, ma è necessario valutare il modello implementato con l'ausilio di altre metriche (Capitolo 2);
- Nel caso in cui il dataset sia sufficientemente grande, è possibile campionare la classe maggiormente rappresentata in numero simile alla cardinalità della classe meno rappresentata, oppure inserire nel dataset copie di tuple corrispondenti alla classe minore;
- Per non inserire copie di tuple già esistenti nel dataset, è possibile ricorrere a tecniche di generazione di oggetti "sintetici", ovvero tuple simili ma non perfettamente identiche a quelle già presenti. Nel caso in esame, è stata scelta questa strategia, utilizzando l'algoritmo SMOTE (Synthetic Minority Over-Sampling);

Capitolo 5

Descrizione della soluzione progettata e implementata

In questo capitolo viene trattata l'implementazione e la validazione del modello predittivo analizzando il dataset descritto nel capitolo 4. Di seguito saranno analizzati i processi implementati nelle fasi di preprocessing, analisi e validazione. Il framework prevalentemente utilizzato in questa fase è stato RapidMiner, un software di Data Mining in grado di effettuare operazioni di preparazione dei dati, Machine Learning, e implementazione di modelli predittivi. In particolare per il mio progetto ho utilizzato il software Rapid Miner Studio: un tool che, tramite una GUI, permette di configurare ed eseguire dei workflow (chiamati "Processi") costituiti da vari blocchi (chiamati anche "Operatori"). Gli operatori eseguono una singola operazione e l'output di uno, costituisce l'input di quello successivo. Soprattutto nella prima fase di preparazione dei dati, l'utilizzo di RapidMiner è stato integrato con degli script scritti in Python che sono poi stati 'incapsulati' all'interno di operatori del processo di RapidMiner. La scelta del framework da utilizzare è ricaduta su RapidMiner in quanto consente di applicare rapidamente diversi algoritmi e confrontarne i risultati: caratteristica molto utile in questa prima fase di valutazione delle potenzialità di un'analisi di questo tipo applicata al processo produttivo.

5.1 Preprocessing dati

5.1.1 Unità di Qualità

In Figura 5.1 è stato riportato un estratto del Database dell'unità di Qualità, dal quale sono stati prelevati i dati riguardanti il controllo della qualità. Nella figura sono presenti 3 tabelle del database e le relative relazioni tra le chiavi. La tabella "*AlgResult*", il cui contenuto è stato riportato nel capitolo precedente (Figura 4.3), contiene una legenda dei possibili esiti del controllo; la chiave "*CatNo*" della tabella

"*AlgResult*" è in relazione uno a molti con la chiave "*Category*" della tabella "*Results*", che contiene tutti i parametri rilevati in un singolo controllo effettuato con la sonda. La tabella "*Results*" ha una coppia di chiavi che identificano la tupla, ovvero "*TestID*" e "*SpotID*"; l'attributo "*TestID*" mette in relazione la tabella "*Results*" con "*ResultsHeader*", la quale contiene dati secondari riguardo il test, come ad esempio l'operatore (*Operator*) che ha effettuato il controllo, la data (*CreationDate*) e opzionalmente delle note (*Remarks*) riguardo il test.

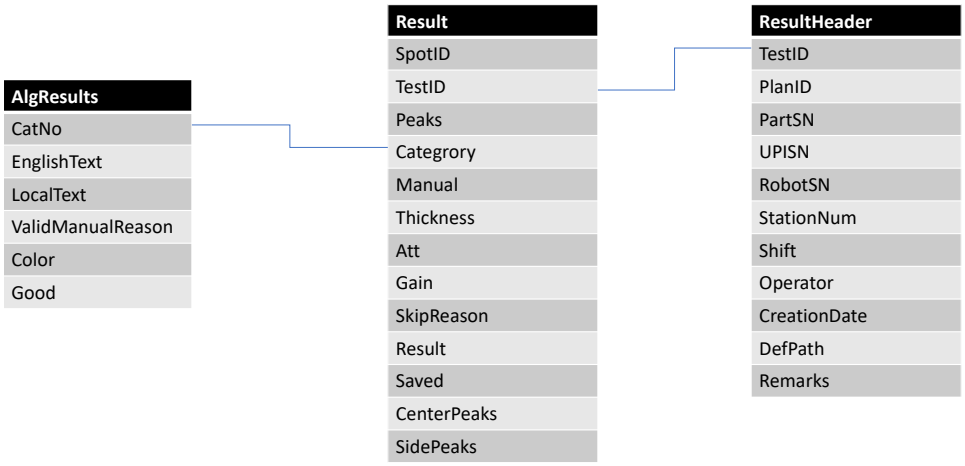


Figura 5.1. Relazione tra le tabelle del DB Qualità

I dati sono stati estratti dal database tramite uno script installato sui PC utilizzati per il controllo della qualità, in grado di eseguire giornalmente, a fine turno, una query per estrarre i dati rilevanti relativi alla sola data corrente. Di seguito un estratto del codice utilizzato. Poiché in questa fase, il CIS[1] è stato inserito manualmente dagli operatori tramite il campo "Remarks", ovvero un campo dedicato all'inserimento di note opzionali, il database non presentava un campo dedicato al CIS, bensì l'attributo "Remarks" rappresentava in maniera non standardizzata il codice identificativo della scocca e molto spesso non era proprio presente (data la natura opzionale del campo), rendendo impossibile l'integrazione tra i database. All'interno della query, infatti, sono state inserite alcune condizioni necessarie per estrarre le tuple che presentavano l'attributo contenente il CIS.

```
1 $reg1 = "cis[0-9]{8}"
```

[1]CIS = Codice Identificativo Scocca

```
2  $reg2 = "[0-9]{8}"
3  $reg3 = "cis[0-9]{8}.*"
4  $reg4 = "[0-9]{8}.*"
5  $query = "SELECT Results.TestID, ResultHeader.Remarks,
       SpotID, Operator, CreationDate, Thickness, Att, Gain,
       CatNo, AlgResults.EnglishText
6  FROM ResultHeader, Results, AlgResults
7  WHERE Results.Category=AlgResults.CatNo
8          AND ResultHeader.TestID=Results.TestID
9          AND (ResultHeader.Remarks LIKE '$reg1'
10         OR ResultHeader.Remarks LIKE '$reg2'
11         OR ResultHeader.Remarks LIKE '$reg3'
12         OR ResultHeader.Remarks LIKE '$reg4')"
```

Il set di dati prelevato dall'unità Qualità presentava numerosi outliers dovuti alla natura manuale del controllo a ultrasuoni; inoltre la possibilità di validare manualmente una misurazione considerata non valida dallo strumento, ha aumentato notevolmente il rumore nei dati.

Il primo processo in figura 5.2, utilizzato per eliminare gli outliers, è costituito da tre operatori:

- **Filter**: operatore necessario per diminuire quanto più possibile il rumore del set di dati. Le regole impostate all'interno del filtro escludono tutte le tuple che:
  a. sono state validate manualmente;
  b. contengono valori anomali dell'attributo 'Result' (mantenendo esclusivamente i valori 1 per le saldature di buona qualità e 2 per quelle di scarsa qualità);
  c. presentano misurazioni anomale degli attributi 'Thickness', 'Gain', 'Attenuation'.
- **Select attributes**: operatore tramite il quale vengono selezionati solo gli attributi di interesse;
- **Set Role**: operatore necessario per indicare l'attributo che il classificatore dovrà predire, tramite l'assegnazione del ruolo *label*; in questa prima fase, è stato deciso di considerare solo l'esito del risultato[2] e non la categoria (Figura 4.3)

[2]Esito controllo qualità:

– 1: OK

– 2: KO

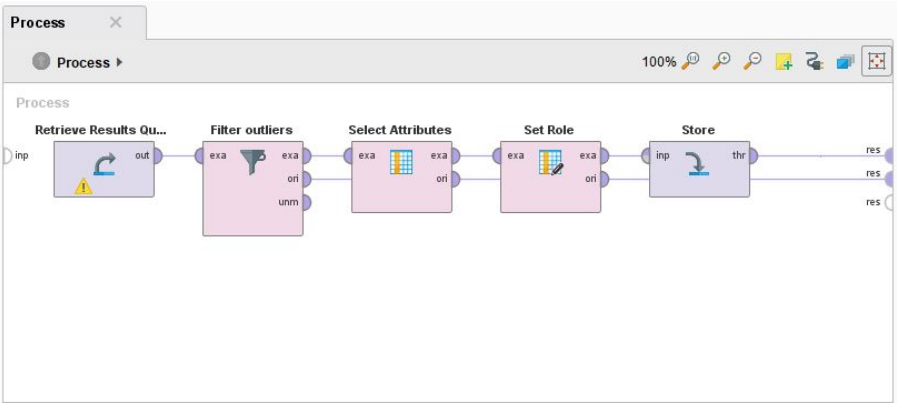


Figura 5.2. Preprocessing dati Qualità

5.1.2 Unità di Lastratura

I dati prelevati dall'unità di Lastratura non presentavano la stessa quantità di rumore e outliers del dataset precedente, nonostante ciò, è stato necessario un lavoro di preprocessing in termini di *feature selection* e *feature transformation*.

Quando lo stabilimento è attivo su tutti i turni, vengono effettuate oltre 20.000 saldature in un giorno. I PLC della linea di produzione, durante la raccolta dei dati generano uno storico giornaliero dei parametri di lavoro delle macchine saldatrici. Per ottenere un database da integrare con quello dell'unità di lastratura e da analizzare, sono stati raccolti quotidianamente i dati sulle saldature e successivamente uniti in un unico database.

I parametri monitorati in questa fase sono molteplici e riguardano tutti i vari componenti della macchina; per l'analisi in esame sono stati selezionati solo gli attributi descritti in Figura 4.2, che descrivono i valori dei parametri elettronici della macchina. Un ulteriore approfondimento riguardo il parametro "ProgNo" è doveroso per comprendere alcune scelte effettuate in fase di preprocessing di questo dataset. Ogni punto di saldatura, associato al proprio codice identificativo che è unico su ogni scocca (o parte mobile), è configurato sulla macchina saldatrice con uno specifico programma che contiene il settaggio di tutti i parametri di lavoro. La macchina che deve effettuare una saldatura in un punto, quindi, imposta il programma predefinito corrispondente, che configurerà automaticamente i valori di riferimento di tutti i parametri; durante l'operazione di saldatura, i sensori monitorano i parametri impostati dal programma e archiviano i valori effettivamente misurati di questi parametri. Poiché l'analisi è stata effettuata considerando contemporaneamente tutti i programmi dei vari punti, addestrare il classificatore basandosi esclusivamente sui valori effettivi dei parametri fisici non avrebbe prodotto risultati ottimali, poiché valori differenti nei parametri non necessariamente implicano qualità differenti della

saldatura. Per ovviare a questo problema è stato deciso di calcolare una variazione relativa di tutti i parametri per poter comparare tuple corrispondenti a saldature effettuate su punti diversi con programmi diversi.

Di seguito viene riportato uno script scritto in Python e utilizzato per unire i dati raccolti in tre giorni di lavoro e per calcolare la variazione relativa dei parametri monitorati, successivamente la nuova colonna generata è stata aggiunta al dataset risultante.

```
1  import pandas as pd
2  import numpy as np
3
4
5  def variation(x, actual, ref):
6  return 1 - (float(x[actual]))/(float(x[ref]))
7
8
9  def create_var_col(df):
10 df['ilstsVar'] = df.apply(lambda x: variation(x, 'ilsts',
       'iDemandStd'), axis=1)
11 df['phaseVar'] = df.apply(lambda x: variation(x, 'pha2',
       'phaStd'), axis=1)
12 df['currentVar'] = df.apply(lambda x: variation(x,
       'currentActualValue', 'currentRefValue'), axis=1)
13 df['voltageVar'] = df.apply(lambda x: variation(x,
       'voltageActualValue', 'voltageRefValue'), axis=1)
14 df['energyVar'] = df.apply(lambda x: variation(x,
       'energyActualValue', 'energyRefValue'), axis=1)
15 df['powerVar'] = df.apply(lambda x: variation(x,
       'powerActualValue', 'powerRefValue'), axis=1)
16 df['resistanceVar'] = df.apply(lambda x: variation(x,
       'resistanceActualValue', 'resistanceRefValue'),
17 axis=1)
18 df['pulseWidthVar'] = df.apply(lambda x: variation(x,
       'pulseWidthActualValue', 'pulseWidthRefValue'),
19 axis=1)
20
21
22 # Read excel files
23 df_8_9 = pd.read_excel('C:/Users/marco/Desktop/File
       lastratura/dati scocca/mod/08-09-mag-2018.xlsx')
24 df_9_10 = pd.read_excel('C:/Users/marco/Desktop/File
       lastratura/dati scocca/mod/09-10-mag-2018.xlsx')
25 df_9 = pd.read_excel('C:/Users/marco/Desktop/File
       lastratura/dati scocca/mod/09-mag-2018.xlsx')
26 df_10 = pd.read_excel('C:/Users/marco/Desktop/File
       lastratura/dati scocca/mod/10-mag-2018.xlsx')
```

```
27
28 # Remove null values
29 df_8_9 = df_8_9.loc[df_8_9['currentRefValue'] != 0]
30 df_9_10 = df_9_10.loc[df_9_10['currentRefValue'] != 0]
31 df_9 = df_9.loc[df_9['currentRefValue'] != 0]
32 df_10 = df_10.loc[df_10['currentRefValue'] != 0]
33
34 # Create percentage error column
35 create_var_col(df_8_9)
36 create_var_col(df_9_10)
37 create_var_col(df_9)
38 create_var_col(df_10)
39
40 # Create one big file
41 all_data = pd.DataFrame()
42 all_data = all_data.append(df_8_9, ignore_index=True)
43 all_data = all_data.append(df_9, ignore_index=True)
44 all_data = all_data.append(df_9_10, ignore_index=True)
45 all_data = all_data.append(df_10, ignore_index=True)
46
47
48 # Write results
49 df_8_9.to_csv('out_8_9.csv', sep='|')
50 df_9_10.to_csv('out_9_10.csv', sep='|')
51 df_9.to_csv('out_9.csv', sep='|')
52 df_10.to_csv('out_10.csv', sep='|')
53 all_data.to_csv('out.csv', sep='|')
```

5.2 Implementazione del classificatore

Dopo una fase preliminare di preprocessing descritta precedentemente, il dataset risultante è stato utilizzato per addestrare un primo classificatore. A prescindere dall'algoritmo scelto (a meno di piccole variazioni nei risultati), il modello risultante tendeva a riportare comportamenti anomali come descritto nel capitolo precedente (Capitolo 4.4). Il modello presentava un'accuratezza molto prossima al 100%, poiché tutte le tuple venivano classificate come saldature di buona qualità, essendo effettivamente in numero molto maggiore, considerando le altre tuple come outliers della stessa classe.

Un modello del genere sarebbe stato del tutto inutilizzabile dato che la classe meno rappresentata, ovvero la saldatura di scarsa qualità, è la classe più interessante da individuare.

Visti i risultati ottenuti, si è deciso di bilanciare il dataset tramite due tecniche distinte:

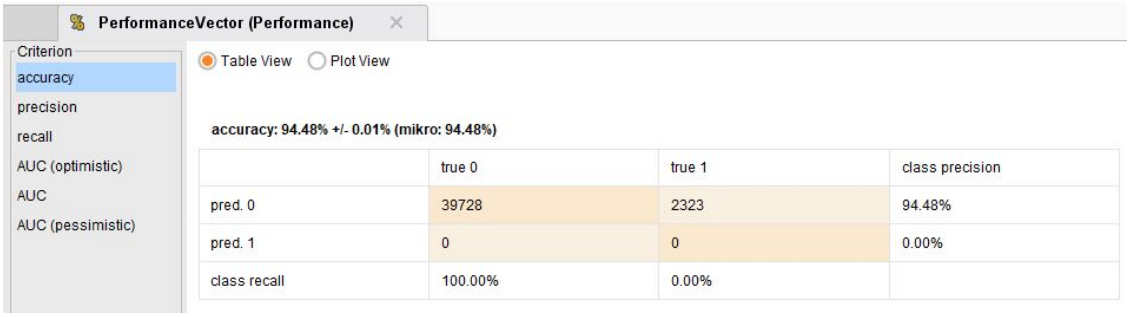


Figura 5.3. Accuratezza del classificatore con dataset fortemente sbilanciato

- *Sovradimensionamento* della classe meno rappresentata tramite la tecnica SMOTE in grado di creare oggetti fittizi della classe meno rappresentata;
- *Sottodimensionamento* della classe maggiormente rappresentata tramite campionamento;

Nel primo caso, è stato utilizzato un modulo di Python chiamato *imblearn* in combinazione con *pandas* e *numpy* per scrivere uno script in grado di sovracampionare la classe corrispondente alle saldature di scarsa qualità.

```
1  import pandas as pd
2  import numpy as np
3  from collections import Counter
4  from imblearn.over_sampling import SMOTE
5
6  df =
       pd.read_excel('C:/Users/marco/Desktop/welding_data_variations_labelled.
7  # print(df.columns)
8
9  X = df.drop(['result'], axis=1)
10 # X = df.drop(['result', 'progNo', 'protRecord_ID'], axis=1)
11 y = df['result']
12
13 print('Original dataset shape {}'.format(Counter(y)))
14
15 sm = SMOTE(random_state=42)
16
17 X_res, y_res = sm.fit_sample(X, y)
18
19 print('Resampled dataset shape {}'.format(Counter(y_res)))
20
21 df1 = pd.DataFrame(data=np.column_stack((X_res, y_res)))
22 df1.to_excel('C:/Users/marco/Desktop/welding_data_variations_labelled_bala
       sheet_name='sheet1', index=False)
```

Ottenendo come output un file contente il dataset bilanciato:

```
1 Original dataset shape Counter({0L: 39728, 1L: 2323})
2 Resampled dataset shape Counter({0: 39728, 1: 39728})
```

Nel secondo caso è stato utilizzato Rapid Miner per implementare un modello in grado di identificare la classe più rappresentata e campionarla in numero uguale a quella meno rappresentata (Figura 5.4).

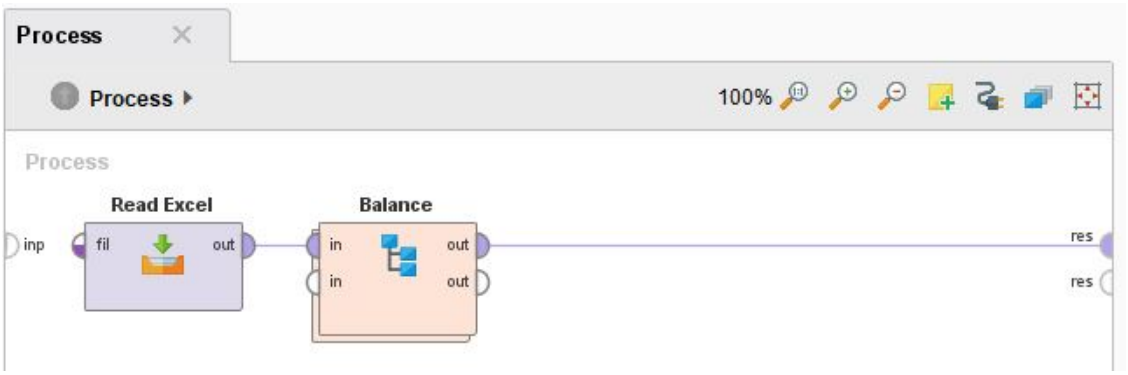


Figura 5.4. Processo di Rapid Miner utilizzato per sottocampionare la classe più rappresentata identificata con il valore 1 dell'attributo *result* e corrispondente a saldature di scarsa qualità.

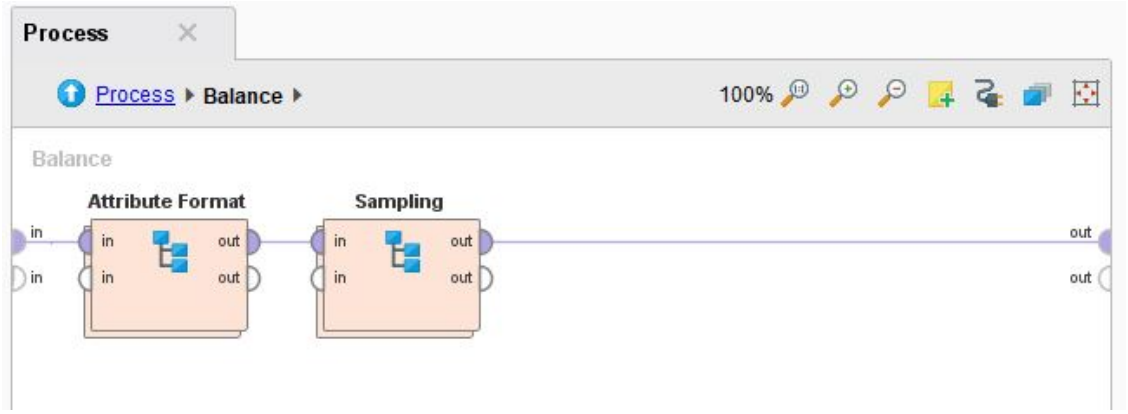


Figura 5.5. Focus sull'operatore *Balance*.

Come risultato, in questo caso, si ottiene un dataset molto ridotto in quanto la classe positiva è stata fortemente sottodimensionata (Figura 5.8).

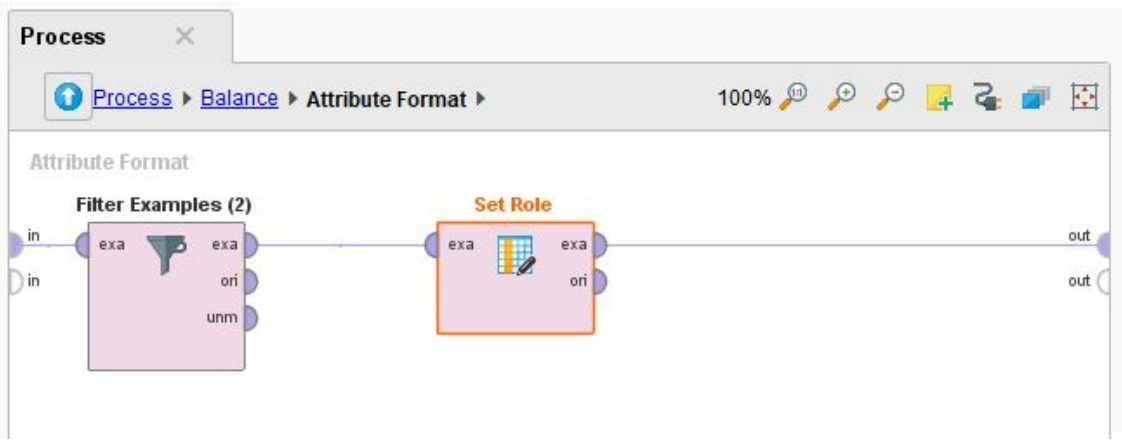


Figura 5.6. Focus sull'operatore *Attribute Format*.

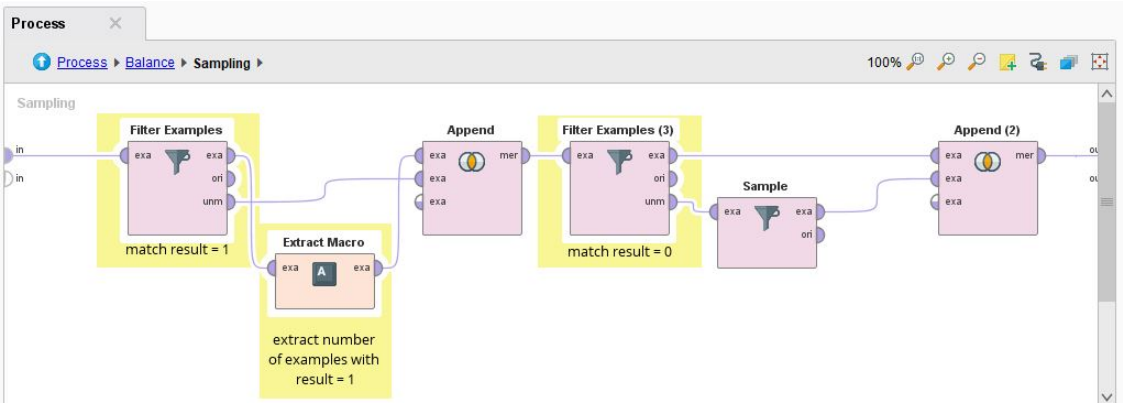


Figura 5.7. Focus sull'operatore *Sampling*.

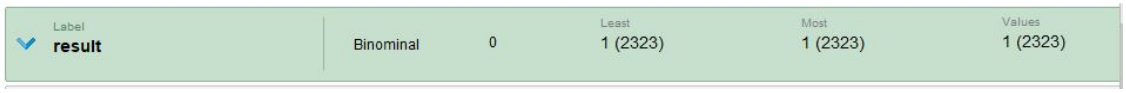


Figura 5.8. Risultato del campionamento

Capitolo 6

Risultati sperimentali

Dopo aver preparato il dataset eliminando gli outliers e averlo bilanciato per evitare problemi di overfitting, sono stati applicati diversi algoritmi per valutarne le prestazioni. In alcuni casi è stato necessario utilizzare un dataset ridotto (ottenuto tramite campionamento), per ridurre i tempi di esecuzione dell'algoritmo. Di seguito verranno presentati i processi implementati su Rapid Miner, i principali parametri settati e i risultati ottenuti con tutti i modelli.

Decision Tree

Il primo modello è stato implementato tramite un albero decisionale. Il processo di partenza è costituito da 4 operatori:

- **Retreive**: utilizzato per recuperare il dataset preprocessato;
- **Select Attribute**: utilizzato per selezionare gli attributi da analizzare e l'attributo *result* da predire;
- **Set Role**: utilizzato per indicare al programma il ruolo di *label* per l'attributo *result*;
- **Cross Validation**: utilizzato per addestrare e testare il modello; costituito da un input (il dataset iniziale) e 3 output (il modello implementato, il dataset iniziale e le performance misurate).

L'operatore *Cross Validation* (Figura 6.2) effettua, per l'appunto, un'operazione di validazione di tipo *Cross Validation* (Capitolo 2) tramite una suddivisione in 10 *folds*. Le performance raggiunte per mezzo di questo modello sono mostrate in figura 6.3.

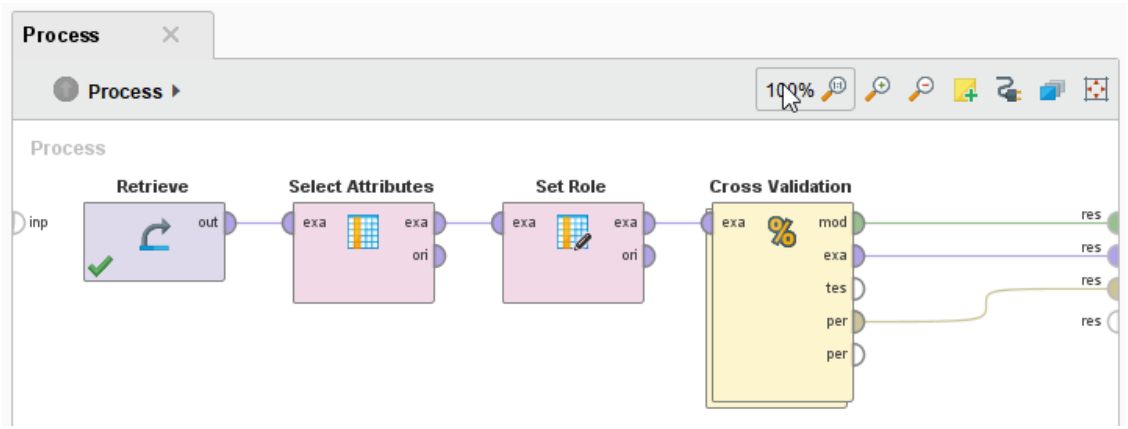


Figura 6.1. Processo implementato in Rapid Miner per l'applicazione dell'albero decisionale

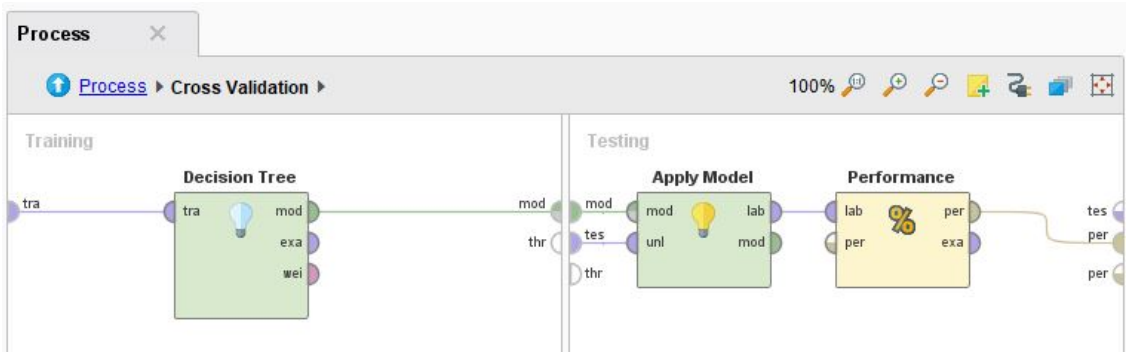


Figura 6.2. Cross Validation con Decision Tree

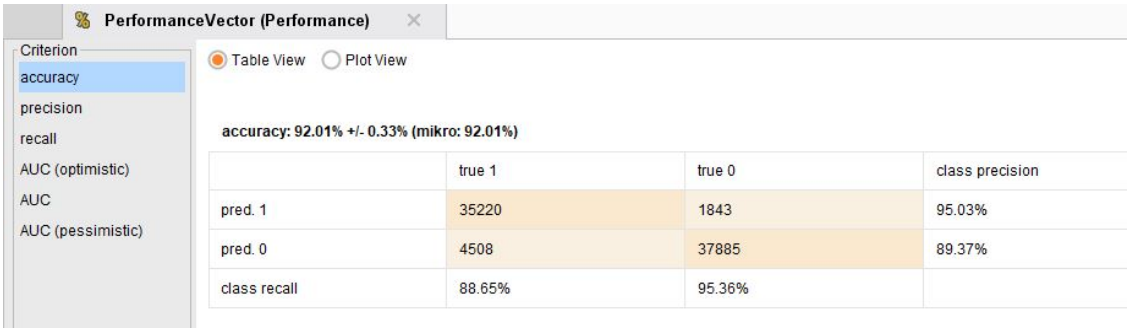


Figura 6.3. Risultati ottenuti dalla validazione del modello con Decision Tree

K-Nearest Neighbors

Il secondo modello è stato implementato utilizzando un algoritmo K-Nearest-Neighbors. Il processo di partenza è analogo al caso precedente, ma cambia l'algoritmo utilizzato all'interno dell'operatore *Cross Validation* (Figura 6.4). Per la configurazione

dell'algoritmo K-NN sono stati scelti $k = 5$ e misure della distanza di tipo Euclideo. I risultati ottenuti sono presentati in figura **??**.

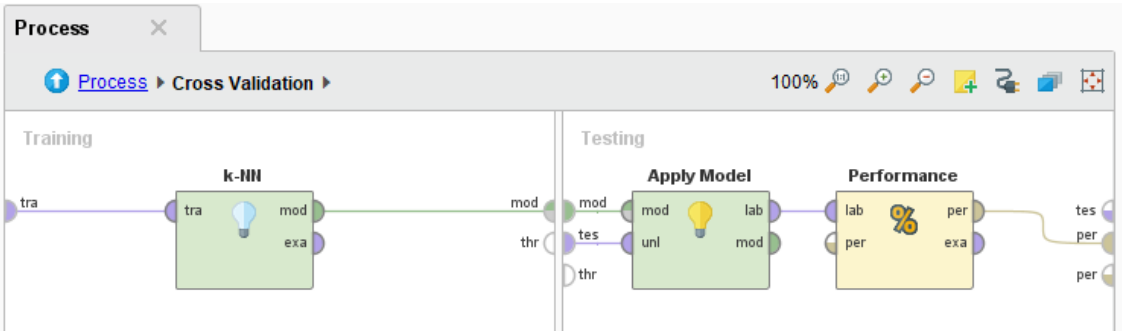
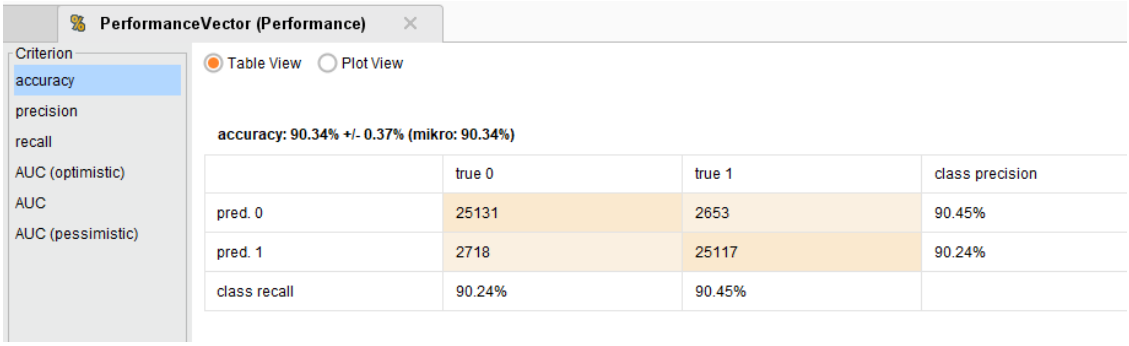


Figura 6.4. Cross Validation con K-Nearest-Neighbors



	true 0	true 1	class precision
pred. 0	25131	2653	90.45%
pred. 1	2718	25117	90.24%
class recall	90.24%	90.45%	

Figura 6.5. Risultati ottenuti dalla validazione del modello con K-Nearest-Neighbors

Neural Networks

Il terzo modello è stato implementato utilizzando un algoritmo di Rete Neurale. In questo caso, analizzare l'intero dataset di partenza come nei casi precedenti avrebbe richiesto una eccessiva quantità di risorse computazionali e tempo, perciò è stato effettuato un campionamento per ridurre la cardinalità del dataset (Figura 6.6). L'operatore di *Cross Validation* è stato configurato come nei casi precedenti, inserendo l'algoritmo di implementazione di Rete Neurale (Figura 6.7). La rete neurale ottenuta (Figura 6.8), è composta da 8 nodi di input relativi agli 8 parametri monitorati, 7 nodi intermedi di elaborazione, e 2 nodi di output corrispondenti alle due classi. I risultati ottenuti sono presentati in figura 6.9.

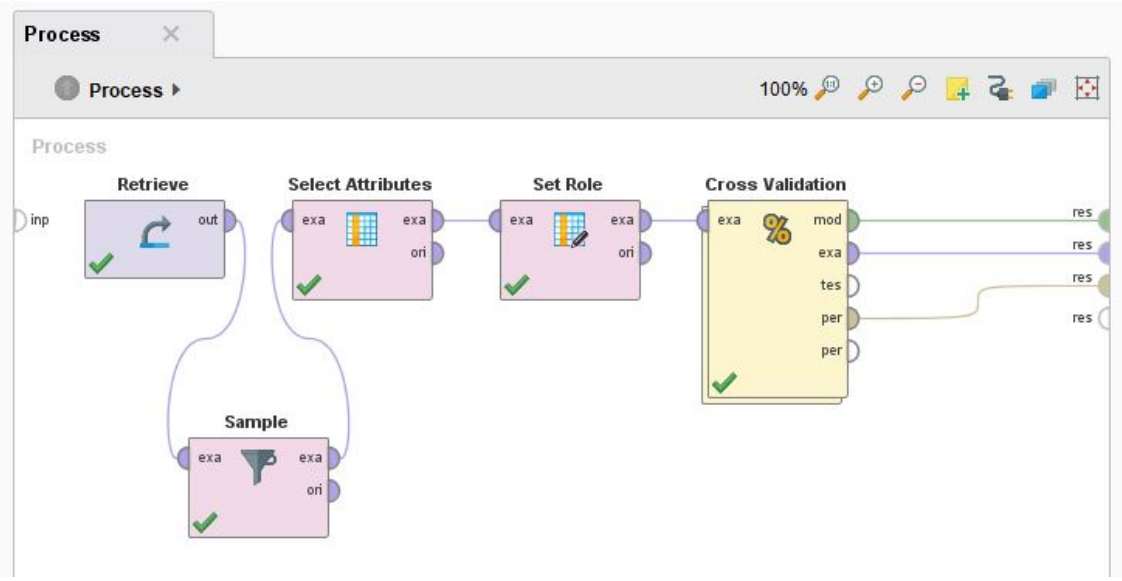


Figura 6.6. Processo implementato in Rapid Miner per l'applicazione della rete neurale

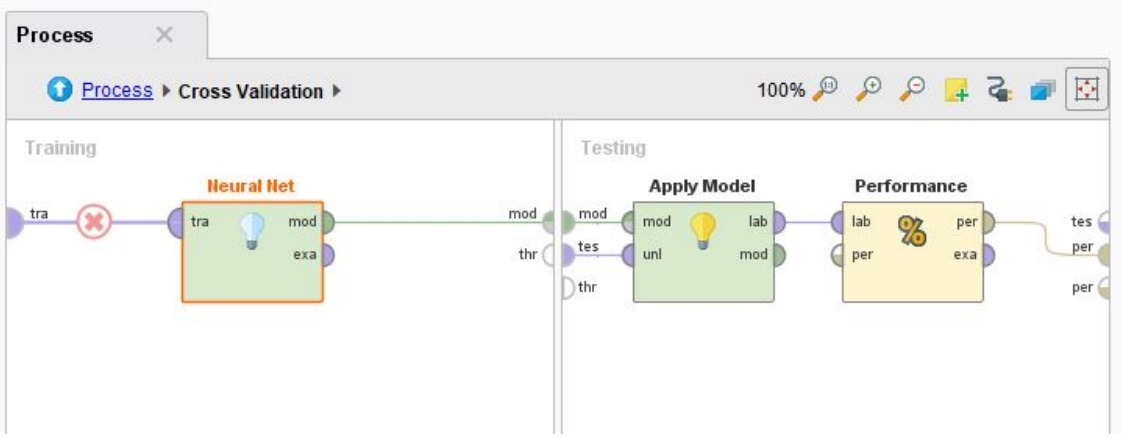


Figura 6.7. Operatore di Cross Validation con Rete Neurale

Naive Bayes

Il quarto modello è stato implementato utilizzando un algoritmo di Classificazione Bayes (Figura 6.10). I risultati ottenuti sono presentati in figura 6.11.

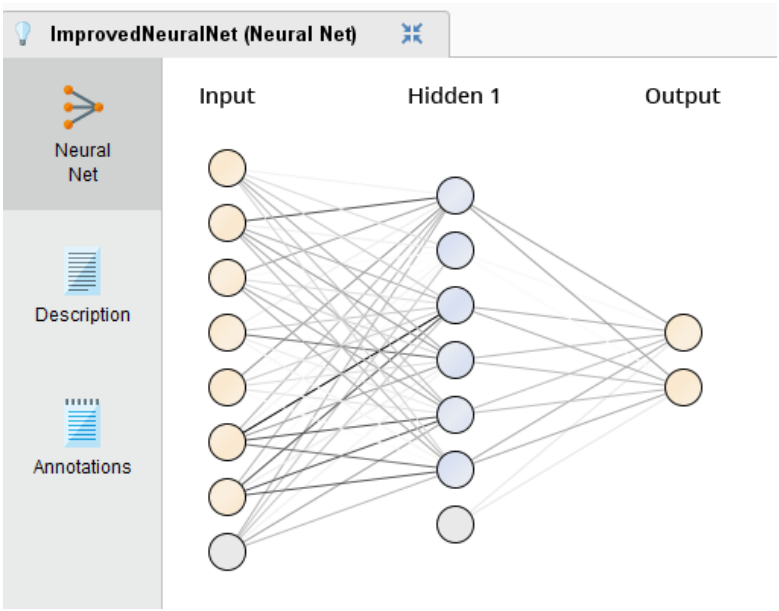
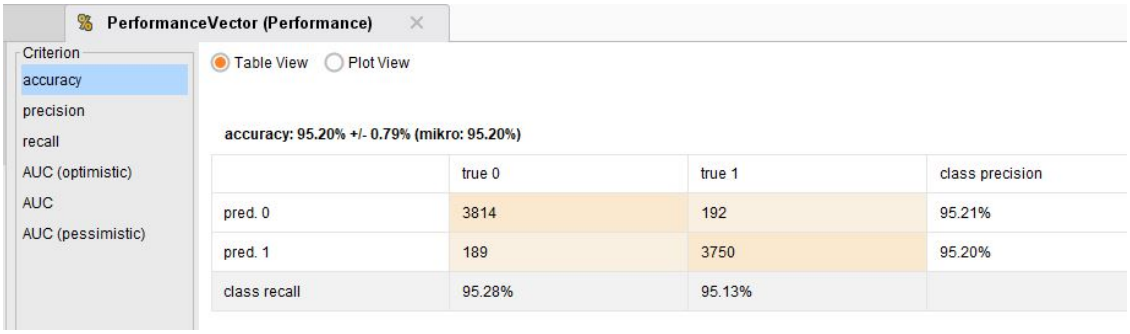


Figura 6.8. Neural Network



	true 0	true 1	class precision
pred. 0	3814	192	95.21%
pred. 1	189	3750	95.20%
class recall	95.28%	95.13%	

Figura 6.9. Risultati ottenuti dalla validazione del modello con Neural Network

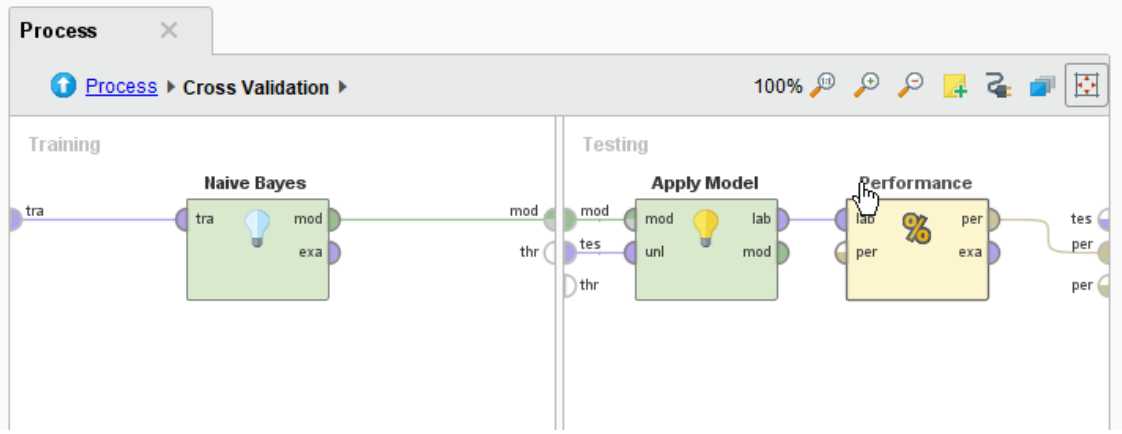
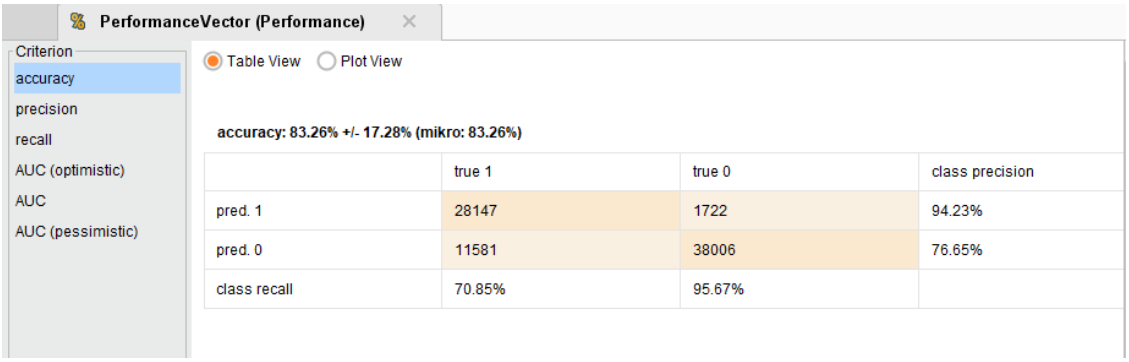


Figura 6.10. Operatore di Cross Validation con Naive Bayes



	true 1	true 0	class precision
pred. 1	28147	1722	94.23%
pred. 0	11581	38006	76.65%
class recall	70.85%	95.67%	

Figura 6.11. Risultati ottenuti dalla validazione del modello con Naive Bayes

Capitolo 7

Conclusioni e sviluppi futuri

L'introduzione di nuove tecnologie nelle industrie ha cambiato il modo di produrre e la concezione stessa del lavoro all'interno delle aziende. Molte imprese hanno iniziato a investire per introdurre i principi dell'Industria 4.0 all'interno del loro processo produttivo. In questo contesto, per competere a livello globale, ai giorni d'oggi, risulta di fondamentale importanza impiegare al meglio le tecnologie introdotte. Per sfruttare al meglio i Big Data prodotti e archiviati nei database di un'azienda manifatturiera come quella automobilistica, ad esempio, è possibile sfruttare tecniche di analisi dei dati per estrarre informazioni utili altrimenti non facilmente deducibili.

Il presente elaborato fornisce uno studio sulle prestazioni dei modelli predittivi implementati tramite l'addestramento di classificatori e applicati al processo produttivo per il controllo della qualità. I risultati ottenuti dalla validazione dei modelli illustrati nei capitoli precedenti mostrano come sia possibile fornire un'accurata stima dell'esito di una saldatura al momento della sua esecuzione. Alcuni degli algoritmi utilizzati sono in grado di classificare una saldatura con un'elevata precisione, grazie alla forte correlazione presente tra i parametri fisici monitorati, e l'esito della saldatura stessa. Lo studio si è focalizzato sulla variazione tra il valore dei parametri impostati sul macchinario e quelli effettivamente misurati, dovuta, in parte, anche allo stato di usura della macchina che effettua la saldatura.

L'implementazione di un modello simile a quelli descritti, fornirebbe uno strumento aggiuntivo per poter ottenere informazioni significative dai dati prodotti dalle macchine, in modo da passare da una manutenzione reattiva ad una predittiva, massimizzare l'efficienza e ridurre gli sprechi. Conoscere in anticipo il possibile esito di una operazione al momento in cui essa viene effettuata, infatti, garantirebbe ai responsabili della manutenzione un'informazione importante sullo stato di usura dei vari componenti.

L'analisi dei dati trattata in questo progetto può essere estesa, con ragionamenti analoghi, a molte altre operazioni automatizzate effettuate all'interno dello stabilimento di Mirafiori come, ad esempio, la rivettatura Tucker. Inoltre, i risultati ottenuti durante la classificazione, possono essere impiegati per affinare il calcolo

del coefficiente OEE di efficienza del plant, prevenendo costose rilavorazioni del prodotto.

Bibliografia

[1] Almada-Lobo, F. The industry 4.0 revolution and the future of manufacturing execution systems (mes). *Journal of Innovation Management 3*, 4 (2016), 16–21.

[2] Arsovski, S., Dokić, I., and Pesic-Dokic, S. Quality in world class manufacturing, 2011.

[3] BERGMANN, A. Data mining for manufacturing: Preventive maintenance, failure prediction, quality control.

[4] Bosch. A brief history of industry, 2018. [Online; accessed 121-June-2018].

[5] Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R. *Advances in knowledge discovery and data mining*, vol. 21. AAAI press Menlo Park, 1996.

[6] Giudici, P. *Applied data mining: Statistical methods for business and industry*. John Wiley & Sons, 2005.

[7] Han, J., Pei, J., and Kamber, M. *Data mining: concepts and techniques*. Elsevier, 2011.

[8] Kagermann, H., Helbig, J., Hellinger, A., and Wahlster, W. *Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0: Securing the Future of German Manufacturing Industry ; Final Report of the Industrie 4.0 Working Group*. Forschungsunion, 2013.

[9] Magnaghi, Giancarlo. World class manufacturing e industria 4.0 alla base della ripresa del gruppo fca, 2017. [Online; accessed 17-June-2018].

[10] Rüssmann, M., Lorenz, M., Gerbert, P., Waldner, M., Justus, J., Engel, P., and Harnisch, M. Industry 4.0: The future of productivity and growth in manufacturing industries. *Boston Consulting Group 9* (2015).

[11] Schonberger, R. J. *World class manufacturing*. Simon and Schuster, 2008.

[12] Tan, P.-n., Steinbach, M., and Kumar, V. *Data Mining Introduction*. Pearson Addison Wesley, 2006.