POLITECNICO DI TORINO

Corso di Laurea Magistrale
in Ingegneria Matematica

Tesi di Laurea Magistrale

Metodi di machine learning per caratterizzare i clienti aziendali

Relatore
Prof. Gianluca Mastrantonio

Candidato
Jacopo Dominici

Anno Accademico 2021-2022

Indice

Elenco delle tabelle

Elenco delle figure

Introduzione

Nel seguente elaborato, viene illustrato il progetto svolto nell'azienda Omicron Consulting s.r.l. per capire quali possono essere i loro possibili clienti. La ricerca dei potenziali clienti è una tematica molto importante per un'azienda. Per prima cosa permette a qust'ultima di evitare di proporre beni e servizi a tutte le aziende ma solo a quelle interessate. Grazie alla caratterizzazione dei possibili clienti è possibile capire su cosa basare l'offerta da parte dell'azienda. Viene aiutata così anche la sezione marketing poichè conoscendo a priori il target dei clienti è più facile indirizzare le strategie. Una strategia di marketing mirata ad un determinato target di mercato consente l'utilizzo di tecniche più specifiche, come per esempio il linguaggio da utilizzare. Questo lavoro ha quindi l'obiettivo di fornire un punto di riferimento per l'azienda e consentirà di ottimizzare tempo e risorse.
Per prima cosa è stata effettuata una ricerca per capire a quale settore appartengono le aziende che sono clienti di Omicron Consulting s.r.l. o che lo sono state in passato. Tutto ciò è stato fatto tramite l'utilizzo del codice ATECO. Da questa ricerca è emerso che l'azienda ha clienti in vari settori ma quello predominante è quello relativo al codice ATECO 62, ovvero aziende di consulenza informatica e produzione di software. Successivamente è stata effetuata una ricerca sulla località della sede legale di queste aziende selezionate e come risultato è emerso che circa il 60% delle aziende clienti con codice ATECO 62 si trovano nelle province di Roma, Milano e Torino. L'analisi si è quindi basata nell'individuare i possibili clienti tra le aziende situate nelle province di Torino, Milano e Roma che appartengono al settore di consulenza informatica e produzione di software.
Nel dettaglio la tesi presenta un primo capitolo dove viene illustrata la parte iniziale del progetto, che tratta la raccolta dei dati e la gestione dei valori mancanti. Tutto ciò, per ottenere un dataset pronto per le analisi. In particolare sono state raccolte le informazioni relative a tutte le aziende con le caratteristiche enunciate in precedenza. Nel secondo capitolo viene riportata la cluster analysis con la quale sono state trovate e caratterizzate tutte le aziende che possono essere possibili clienti per l'azienda Omicron Consulting s.r.l. In questo capitolo troviamo la descrizione dei vari algoritmi di clustering e il confronto dei risultati ottenuti con essi. Infine nell'ultimo capitolo vengono illustrate le varie tecniche di classificazione e confrontati i risultati ottenuti con essi. Questo è stato fatto per classificare in futuro le aziende non presenti nel database come possibili clienti o meno.

Capitolo 1

Raccolta ed esplorazione dei dati

La fase iniziale del progetto si è concentrata sull'acquisizione del dato da utilizzare per le analisi. Le fonti utilizzate sono state la base dati AIDA, dove sono presenti diverse variabili riguardanti dati finanziari delle aziende italiane e la base dati dell'azienda Omicron Consulting s.r.l. contenente le informazioni riguardanti i suoi clienti. All'interno della base dati AIDA (van Dijk [2022]) sono presenti le informazioni economico-finanziarie, anagrafiche e commerciali di tutte le società di capitali che operano in Italia. In Aida sono disponibili informazioni quali settore di attività, numero dipendenti e varie informazioni di carattere finanziario. La base dati AIDA è stata scelta poichè permetteva di acquisire i dati delle aziende relativi sia all'anno 2021 ma anche agli anni precedenti, in modo da lavorare su dati non totalmente influenzati dalla pandemia legata al virus Covid-19. Successivamente sono state selezionate le variabili da utilizzare per le analisi, tra quelle presenti nella base dati AIDA. In particolare sono state selezionate variabili prevalentemente di carattere economico escludendo quelle riguardanti il rendimento dei dipendenti dato il settore delle aziende considerato. L'analisi si è basata sulle aziende appartenenti al settore consulenza informatica e produzione di software, quindi aziende che non utilizzano i dipendenti per la produzione di prodotti finiti da vendere poi al dettaglio. Nell'analisi sono state utilizzate variabili identificative come "RagioneSociale" che indica il nome dell'azienda e "PartitaIVA" che è stata utilizzata come codice identificativo per le aziende, poichè è un codice univoco ed obligatorio per ogni azienda. Sono state inoltre utilizzate variabili come "Anno", "Provincia" e "ATECO2007" per avere indicazioni rispettivamente sull'anno di nascita, sulla posizione geografica e sulla sottocategoria dell'azienda. Un'altra tipologia di variabili utilizzate sono quelle di carattere economico come "RVXX" che rappresenta i ricavi dell'anno XX, con XX che può essere 2018, 2019, 2020, 2021 ed "EBITDAXX" che rappresenta l'indicatore della redditività aziendale nell'anno XX, con XX che può essere 2019, 2020, 2021. Nell'analisi sono state anche utilizzate altre variabili di carattere economico relative all'anno 2021 come "TotAttivo" e "TotPassivo" che rappresentano rispettivamente il totale delle attività e delle passività dell'azienda. Sono state utilizzate

anche altre variabili che mi forniscono un'indicazione sia in merito alla liquidità dell'azienda sia ai suoi debiti come "TotDisponLiquide" e "TotDebiti". Relativamente all'anno 2021 sono state utilizzate anche le variabili "TotCrediti", "UtileNetto", "CapCircNetto", "MargineConsumi" e "FlussoCassa" per avere un quadro economico dell'azienda il più completo possibile. Infine è stata utilizzata anche la variabile "Dipendenti" che indica il numero di dipendenti dell'azienda e la sua rispettiva dimensione. Successivamente è stata aggiunta la variabile binaria "Cliente" che indica con valore "1" le aziende che sono clienti dell'azienda Omicron consulting s.r.l. e con valore "0" le aziende che non lo sono. Per la costruzione di questa variabile è stata utilizzata la variabile identificativa "PartitaIva", in modo da legare il dataset della base dati AIDA e il dataset dove sono presenti le aziende clienti dell'azienda Omicron Consulting s.r.l. In particolare si è ottunuto che delle 13921 aziende con sede legale nella provincia di "Roma" o "Milano" o "Torino" e con codice ATECO 62, 13737 non sono clienti dell'azienda Omicron consulting s.r.l. mentre 184 lo sono.

1.1 Gestione dei valori mancanti

All'interno del dataset ci sono diversi valori mancanti che devono essere gestiti prima di iniziare l'analisi. I metodi che consentono di gestire i valori mancanti si distinguono in eliminazione dei record o delle variabili oppure imputazione del valore mancante. L'eliminazione dei record e quindi in questo caso delle aziende avviene se per quell'azienda sono presenti dei valori mancanti. Questo metodo ha i suoi vantaggi, infatti non necessita di nessun calcolo ma può portare ad una grande diminuzione del numero di record da utilizzare nell'analisi. Allo stesso modo si può utilizzare il metodo dell'eliminazione della variabili che hanno troppi valori mancanti. Ci sono infine,i metodi di imputazione che permettono di sostituire il valore mancante con un valore ragionevole e quindi di effettuare l'analisi senza la necessità di diminuire la dimensione del campione originale. Il classico metodo di imputazione consiste nel calcolo della media della variabile X tramite l'utilizzo dei valori non mancanti e dell'utilizzo di questo valore in sostituzione ad ogni valore mancante della variabile. Questo metodo però ha grandi limitazioni infatti per la stessa variabile ogni valore mancante viene sostiuito dallo stesso valore e questo può portare a stime distorte. Per risolvere questo problema si possono utilizzare metodi di imputazione che portano alla sostituzione dei valori mancanti con stime più precise. In particolare possono essere utilizzati modelli di regressione multipla con variabili indipendenti tra loro (Soley-Bori [2013]).
Per questa analisi, data l'elevata numerosità dei valori mancanti, è stato utilizzato il metodo di imputazione tramite modelli di regressione per evitare di ridurre troppo la dimensionalità del dataset. La regressione infatti viene utilizzata per predire una risposta quantitativa Y_i, che in questo caso corrisponde al valore mancante della variabile $\boldsymbol{Y}$ per l'i-esima osservazione, sulla base dei valori delle variabili predittive X_{ij}, dove l'indice i indica il record e l'indice j indica la variabile. Si assume inoltre che Y_i sia data dalla combinazione lineare delle variabili X_{ij} e matematicamente si può formalizzare con l'equazione (1.1).

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + ... + \beta_m X_{im} + \epsilon_i \tag{1.1}$$

dove $m \leq p-1$ e p è il numero di variabili del dataset, mentre ϵ_i rappresente l'i-esimo errore, o componente non spiegata dal modello, che si assume essere generato da una distribuzione normale di media zero e varianza σ^2. Nel dettaglio deve valere che i residui siano scorrelati tra loro e omoschedastici, ovvero che tutti i residui devono avere la stessa varianza. I coefficienti della regressione β_i sono incogniti e deve quindi essere utilizzata una loro stima per calcolare la previsione di Y_i, ovvero $\hat{Y}_i$ come si può vedere nella formalizzazione (1.2).

$$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_{i1} + \hat{\beta}_2 X_{i2} + ... + \hat{\beta}_m X_{im} \tag{1.2}$$

I $\hat{\beta}_j$ si possono ottenere tramite l'approccio dalla minimizzazione del Mean Squared Error (Gareth James [2013]), dove la quantità Mean Squared Error (MSE) è ugale a

$$\begin{aligned} MSE = \frac{1}{N}\sum_{i=1}^{N}\left(Y_i - \hat{Y}_i\right)^2 = \frac{1}{N}\left[\left(\mathbf{Y} - \mathbf{X}^T\hat{\boldsymbol{\beta}}\right)\left(\mathbf{Y} - \mathbf{X}^T\hat{\boldsymbol{\beta}}\right)\right] = \\ = \frac{1}{N}\left[\mathbf{Y}^T\mathbf{Y} - 2\mathbf{Y}^T\mathbf{X}\hat{\boldsymbol{\beta}} + \hat{\boldsymbol{\beta}}^T\mathbf{X}^T\mathbf{X}\hat{\boldsymbol{\beta}}\right] \end{aligned} \tag{1.3}$$

dove N è il numero di osservazioni del dataset, Y_i è la quantità misurata della variabie $\mathbf{Y}$ per l'i-esima osservazione mentre $\hat{Y}_i$ è la sua stima. La matrice $\mathbf{X}$ di dimensione $N \times (m+1)$ è data dalle osservazioni delle variabili utilizzate come predittori e dalla prima colonna composta da tutti 1.
Quindi minimizzando rispetto a $\hat{\boldsymbol{\beta}}$ il MSE, ovvero ponendo uguale a zero la derivata parziale dell'equazione (1.3) rispetto a $\hat{\boldsymbol{\beta}}$, si ottiene la relazione

$$-2\mathbf{X}^T\mathbf{Y} + 2\mathbf{X}^T\mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{0} \tag{1.4}$$

Dall'equazione (1.4) si ottiene infine la stima di β

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{Y} \tag{1.5}$$

Si può vedere quindi come la stima dei β_j si ottiene solo se la matrice $\mathbf{X}^T\mathbf{X}$ è invertibile, cioè la matrice $\mathbf{X}$ deve avere rango pieno. Questo significa che i predittori devono essere linearmente indipendenti tra loro. Si costruisce quindi il grafico delle correlazioni, Figura 1.1, per evitare di utilizzare variabili troppo correlate tra loro nei modelli di regressione. Nella Figura 1.1 si può osservare che le variabili "TotPassivo" e "RicaviVendite2018" sono fortemente correlate con le altre variabili e quindi non devono essere utilizzate nei modelli di regressione per quanto detto in precedenza. Questo approccio viene applicato con il dataset composto solo dalle osservazioni che non hanno valori mancanti poichè occorre conoscere il valore reale delle variabili che si vogliono predire.
Si costruiscono quindi i modelli di regressione per stimare i valori mancanti all'interno del dataset per le variabili "Dipendenti", "EBITDA21", "UtileNetto", "MargineConsumi" e

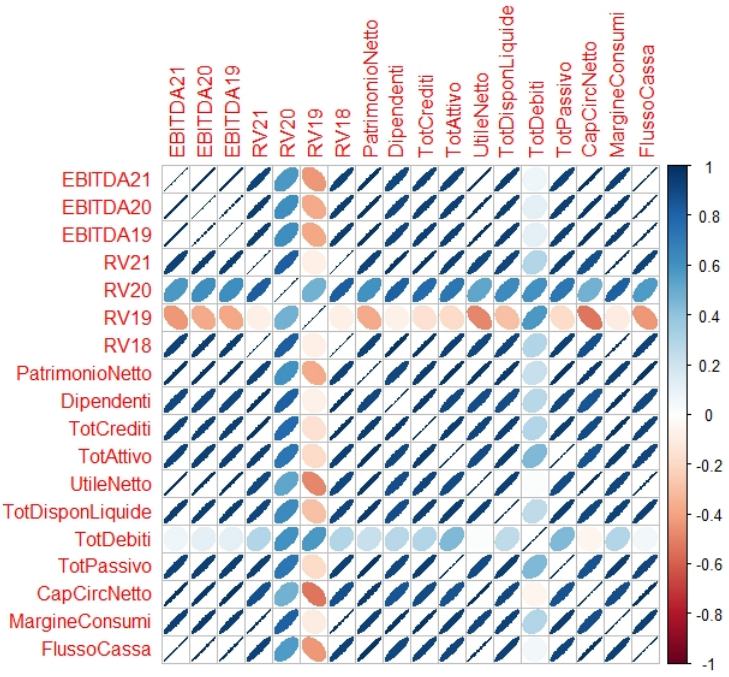

Figura 1.1. Matrice Covarianza delle variabili numeriche del dataset

"FlussoCassa". Il primo modello di regressione costruito è relativo alla stima dei valori mancanti della variabile "Dipendenti". Per stimare questi valori si dovrebbero utilizzare come predittori nel modello di regressione solamente le variabili che non hanno valori mancanti per le aziende che hanno come valore mancante il campo "Dipendenti", per poter poi calcolare con i coefficienti della regressione trovati il valore da sostituire al valore mancante. Questa richiesta potrebbe portare ad una stima non troppo accurata del valore mancante poichè vengono utilizzate poche informazioni. Si potrebbe quindi aumentare l'accuratezza della stima utilizzando anche altre variabili come predittori. Questo però comporterebbe l'eliminazione dal dataset di aziende che hanno come valori mancanti sia il campo "Dipendenti" sia la variabile che si vuole aggiungere al modello di regressione. Quindi per avere una stima abbastanza accurata del valore mancante bisogna tenere conto del trade-off tra accuratezza della stima e il numero di aziende da eliminare dal dataset. Per misurare l'accuratezza della stima si può utilizzare l'indice R^2, un indice compreso tra 0 e 1 che indica quanta varianza viene spiegata dal modello di regressione (Gareth James [2013]). Per ottenere quindi, un valore alto dell'indice R^2 nel modello di regressione che aveva come variabile risposta "Dipendenti", oltre alle variabili che non hanno nessun valore mancante sono state utilizzate come predittori nel modello di regressione anche altre variabili come "MargineConsumi" e "UtileNetto". Questo ha portato ad una eliminazione di alcune aziende dal dataset passando da 13921 aziende a 13885 aziende.
Il procedimento appena descritto viene ripetuto per la stima dei valori mancanti delle variabili "RV20", "RV19", "EBITDA20", "EBITDA19", "FlussoCassa", "MargineConsumi"

Variabile	Indice Variabile
RagioneSociale	1
Provincia	2
Anno	3
ATECO2007	4
EBITDA21	5
EBITDA20	6
EBITDA19	7
RV21	8
RV20	9
RV19	10
PatrimonioNetto	11
Dipendenti	12
TotCrediti	13
TotAttivo	14
UtileNetto	15
TotDisponLiquide	16
TotDebiti	17
CapCircNetto	18
MargineConsumi	19
FlussoCassa	20
Cliente	21

Tabella 1.1. Variabili del Dataset con i relativi indici identificativi

però senza l'esigenza di eliminare nessuna azienda dal dataset.
Ci sono ancora delle variabili che hanno dei valori mancanti. In particolare si ha che le variabili "EBITDA21", "PatrimonioNetto", "TotCrediti", "TotDisponLiquide", "TotDebiti" e "CapCircNetto" hanno un unico valore mancante e coincide per tutte le variabili con la stessa azienda, quindi in questo caso si decide di eliminare direttamente l'azienda, passando da 13885 aziende a 13884 aziende. Per i valori mancanti legati alla variabile "Anno", dal momento che erano solo 12, si è svolta una ricerca riguardo all'anno di fondazione delle aziende ottenendo risultati positivi per 8 aziende, mentre le altre 4 sono state eliminate dal dataset.
Si è ottenuto quindi un dataset composto da 13880 aziende di cui 184 sono clienti di Omicron Consulting s.r.l. e dalle variabili presenti nella Tabella 1.1.
Di seguito vengono elencati i modelli di regressione utilizzati per la stima dei valori mancanti con i relativi valori di R^2. Dove per identificare gli indici j delle variabili $X_{i,j}$ si fa riferimento alla Tabella 1.1.

$$\begin{aligned} Dipendenti_i =& \beta_0 + \beta_5 X_{i,5} + \beta_{11} X_{i,11} + \beta_{14} X_{i,14} + \beta_{15} X_{i,15} + \beta_{16} X_{i,16} + \\ & \beta_{17} X_{i,17} + \beta_{18} X_{i,18} + \beta_{19} X_{i,19} + \beta_{20} X_{i,20} + \epsilon_i \end{aligned} \tag{1.6}$$
$$R^2 = 0.8517$$

$$\begin{aligned} RV20_i =& \beta_0 + \beta_5 X_{i,5} + \beta_8 X_{i,8} \beta_{11} X_{i,11} + \beta_{13} X_{i,13} \beta_{14} X_{i,14} + \beta_{15} X_{i,15} + \\ & \beta_{17} X_{i,17} + \beta_{18} X_{i,18} + \beta_{19} X_{i,19} + \beta_{20} X_{i,20} + \epsilon_i \\ & R^2 = 0.9937 \end{aligned} \tag{1.7}$$

$$\begin{aligned} RV19_i =& \beta_0 + \beta_5 X_{i,5} + \beta_8 X_{i,8} \beta_{11} X_{i,11} + \beta_{13} X_{i,13} \beta_{14} X_{i,14} + \beta_{15} X_{i,15} + \\ & \beta_{17} X_{i,17} + \beta_{19} X_{i,19} + \beta_{20} X_{i,20} + \epsilon_i \\ & R^2 = 0.9833 \end{aligned} \tag{1.8}$$

$$\begin{aligned} EBITDA20_i =& \beta_0 + \beta_5 X_{i,5} + \beta_{13} X_{i,13} \beta_{14} X_{i,14} + \beta_{15} X_{i,15} + \beta_{16} X_{i,16} + \\ & \beta_{17} X_{i,17} + \beta_{19} X_{i,19} + \beta_{20} X_{i,20} + \epsilon_i \\ & R^2 = 0.9759 \end{aligned} \tag{1.9}$$

$$\begin{aligned} EBITDA19_i =& \beta_0 + \beta_5 X_{i,5} + \beta_{11} X_{i,11} \beta_{13} X_{i,13} + \beta_{14} X_{i,14} + \beta_{15} X_{i,15} + \beta_{16} X_{i,16} + \\ & \beta_{17} X_{i,17} + \beta_{18} X_{i,18} + \beta_{19} X_{i,19} + \beta_{20} X_{i,20} + \epsilon_i \\ & R^2 = 0.9785 \end{aligned} \tag{1.10}$$

$$\begin{aligned} FlussoCassa_i =& \beta_0 + \beta_5 X_{i,5} + \beta_6 X_{i,6} + \beta_7 X_{i,7} + \beta_8 X_{i,8} + \beta_9 X_{i,9} + \beta_{11} X_{i,11} + \\ & \beta_{12} X_{i,12} + \beta_{13} X_{i,13} + \beta_{14} X_{i,14} + \beta_{15} X_{i,15} + \beta_{16} X_{i,16} + \\ & \beta_{17} X_{i,17} + \beta_{18} X_{i,18} + \beta_{19} X_{i,19} + \epsilon_i \\ & R^2 = 0.989 \end{aligned} \tag{1.11}$$

$$\begin{aligned} MargineConsumi_i =& \beta_0 + \beta_5 X_{i,5} + \beta_7 X_{i,7} + \beta_8 X_{i,8} + \beta_9 X_{i,9} + \beta_{11} X_{i,11} + \beta_{12} X_{i,12} + \\ & \beta_{13} X_{i,13} + \beta_{14} X_{i,14} + \beta_{15} X_{i,15} + \beta_{18} X_{i,18} + \beta_{20} X_{i,20} + \epsilon_i \\ & R^2 = 0.9855 \end{aligned} \tag{1.12}$$

1.2 Outlier Detection

Il rilevamento dei valori anomali appartiene ai compiti più importanti nell'analisi dei dati, dove con valori anomali si intendono quelle osservazioni che si discostano dalla naturale variabilità dei dati.
Per l'individuazione dei dati anomali una tecnica molto utilizzata nel caso del multivariato è la distanza di Malhanobis, (Filzmoser [2004]). La distanza di Malhanobis tiene conto della correlazione tra le variabili e quindi in particolare della matrice di Covarianza che si può vedere nella Figura 1.2. Per un campione multivariato p-dimensionale, dove p è il

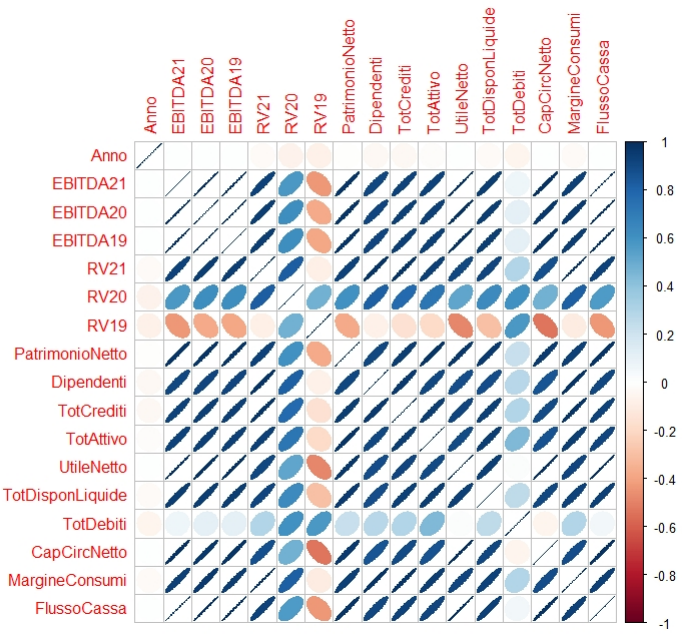

Figura 1.2. Matrice Covarianza finale delle variabili numeriche del dataset

numero delle variabili, la distanza di Mahalanobis per l'osservazione i-esima $\mathbf{X}_{\{i,.\}}$ $(i = 1, ..., N)$ è definita da

$$MD_i = \left((\mathbf{X}_{\{i,.\}} - \bar{\mathbf{X}})^T \mathbf{S^{-1}} (\mathbf{X}_{\{i,.\}} - \bar{\mathbf{X}})\right)^{1/2} \tag{1.13}$$

dove $\mathbf{X}_{\{i,.\}}$ è il vettore p-dimensionale corrispondente all'i-esima osservazione, $\bar{\mathbf{X}}$ è il vettore p-dimensionale della media del campione di osservazioni e $\mathbf{S}$ è la matrice di covarianza, Figura 1.3. Vengono individuate come valori anomali quelle osservazioni che hanno alti valori della distanza di Malhanobis.
Si calcola quindi la distanza di Malhanobis per il campione di osservazioni e quello che si ottiene si può visualizzare nella Figura 1.3. Dalla Figura 1.3 si può notare la possibile presenza di outliers, ovvero di dati anomali. Approfondendo le aziende che sembrano avere dei dati anomali si scopre che si tratta di aziende che si discostano dalle altre poiché sono le multinazionali del settore considerato. Grazie alla verifica effettuata si può affermare quindi che le osservazione che sembravano essere valori anomali non lo sono e non devono essere eliminate dal dataset. Questo risultato viene poi confermato nel capitolo seguente della Cluster analysis.

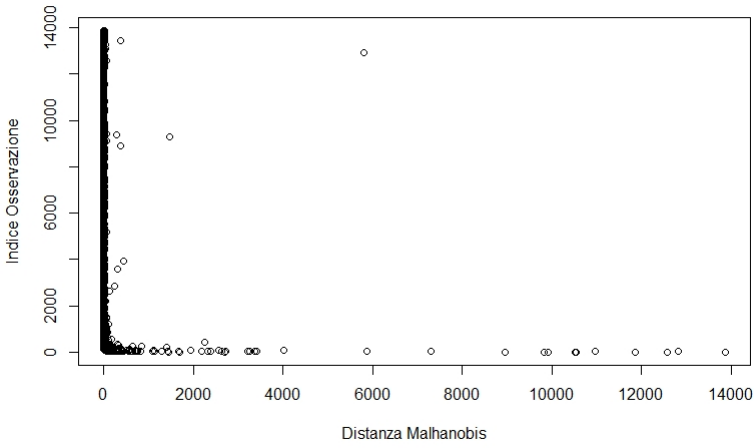

Figura 1.3. Distanza di Malhanobis

Capitolo 2

Cluster Analysis

L'obiettivo dell'analisi ora è quello di individuare tramite la Cluster analysis i possibili clienti dell'azienda Omicron consulting s.r.l. La Cluster analysis per questa analisi consente infatti di individuare sottogruppi di osservazioni simili tra loro e distinguerle dalle altre. Una volta individuati i diversi sottogruppi di aziende simili tra loro si possono caratterizzare come sottogruppi composti da aziende che potranno essere possibili clienti o meno. In particolare si attribuisce ai cluster l'etichetta "Possibili Clienti" se la percentuale di aziende che sono clienti al loro interno supera un determinato target. Come target è stata scelta la percentuale di aziende che sono clienti all'interno dell'intero dataset, usando la seguente formula:

$$target = \frac{n.°Aziende\ clienti}{n.°Aziende} = \frac{184}{13880} \approx 0.0133 = 1.33\% \tag{2.1}$$

Il primo passo fondamentale per la Cluster Analysis, quando ci sono variabili con unità di misura differente, è la standardizzazione del dataset. In questo caso per esempio ci sono unità di misura differenti come "anno" e "euro", e la standardizzazione del dataset consente di dare la giusta importanza a tutte le variabili per il calcolo delle metriche.
Una volta standardizzato il dataset si può procedere con la Cluster Anaysis. In questo capitolo vengono confrontati gli algoritmi K-Means e DBSCAN, utilizzati per la Cluster Analysis, in modo da capire quale tra i due algoritmi performa al meglio al fine delle analisi. Per confrontare gli output ottenuti da questi algoritmi vengono utilizzate soprattutto le rappresentazioni grafiche ottenute grazie all'analisi delle componenti principali.

2.1 Analisi delle componenti principali - PCA

Questa analisi ha come scopo quello di trovare una rappresentazione delle osservazioni in uno spazio di bassa dimensione mantenendo allo stesso tempo una buona frazione di varianza spiegata. In questo modo si può ottenere quindi una rappresentazione grafica anche dei Cluster. L'idea alla base è che ciascuna delle osservazioni viva nello spazio p-dimensionale, dove p è il numero delle variabili, ma non tutte queste dimensioni sono ugualmente interessanti, di fatti la PCA cerca di individuare un piccolo numero di dimensioni quanto più interessanti possibile, ovvero che riescano a catturare la maggior quantità

di informazioni. Ciascuna delle dimensioni trovate dalla PCA è una combinazione lineare delle p variabili normalizzata.
Vediamo ora come si trovano le componenti principali. La prima componente principale (***PC*1**), Equazione (2.2), è la combinazione lineare delle variabili del dataset che riesce a catturare più informazioni possibili, ovvero che ha la varianza più alta. Quindi ***PC*1** è un vettore N-dimensionale, dove N è il numero di osservazioni del dataset, e alla posizione k del vettore si ha la seguente quantità

$$PC1_k = \beta_{11}\mathbf{X}_{\{k,1\}} + \beta_{12}X_{\{k,2\}} + ... + \beta_{1p}X_{\{k,p\}} = \boldsymbol{\beta}_1^T \boldsymbol{X}_{\{k,.\}} \tag{2.2}$$

dove p è il numero di variabili numeriche utilizzate per l'analisi, $X_{\{k,j\}}$ è la quantità assunta dal k-esimo record in corrispondenza della j-esima variabile del dataset, mentre β_{1j} è il coefficiente della combinazione lineare per la prima componente principale relativi alla j-esima variabile del dataset. L'equazione (2.2) può essere anche scritta in forma compatta:

$$PC1_k = \boldsymbol{\beta}_1^T \boldsymbol{X}_{\{k,.\}} \tag{2.3}$$

dove $\boldsymbol{X}_{\{k,.\}}$ è il vettore p-dimensionale composto dalle variabili numeriche del record k e $\boldsymbol{\beta}_1$ è il vettore p-dimensionale dove si hanno i coefficiente della combinazione lineare per la prima componente principale. L'obiettivo è quello di massimizzare la varianza di **PC1**:

$$Var[\mathbf{PC1}] = \frac{1}{N}\sum_{k=1}^{N} \boldsymbol{\beta}_1^T \boldsymbol{X}_{\{k,.\}} \boldsymbol{X}_{\{k,.\}}^T \boldsymbol{\beta}_1 = \boldsymbol{\beta}_1^T \left(\frac{1}{N}\sum_{k=1}^{N} \boldsymbol{X}_{\{k,.\}} \boldsymbol{X}_{\{k,.\}}^T \right) \boldsymbol{\beta}_1 = \boldsymbol{\beta}_1^T \boldsymbol{S} \boldsymbol{\beta}_1 \tag{2.4}$$

dove $\boldsymbol{S}$ è la matrice di covarianza dato che il dataset era stato precedentemente standardizzato ponendo le variabili a media zero.
La combinazione lineare così ottenuta deve essere normalizzata come detto in precedenza. Per normalizzata si intende che $\|\boldsymbol{\beta}_1\|^2 = 1$. Viene imposto questo vincolo perchè aumentando arbitrariamente la grandezza in valore assoluto del vettore $\boldsymbol{\beta}_1$ aumenta $Var[\boldsymbol{PC1}]$, e dato che l'obiettivo è trovare la direzione in cui i dati variano di più si impone ciò.
Il problema si restringe quindi nella ricerca del versore $\boldsymbol{\beta}_1$ che restituisce la direzione della massima varianza:

$$\begin{aligned} &\max_{\boldsymbol{\beta}_1} \boldsymbol{\beta}_1^T \boldsymbol{S} \boldsymbol{\beta}_1 \\ &tale\ che: \\ &\|\boldsymbol{\beta}_1\|^2 = 1 \end{aligned} \tag{2.5}$$

Dato che il problema è convesso e ha vincoli di uguaglianza si può trovare la soluzione grazie alla lagrangiana associata ad esso

$$\mathcal{L}(\boldsymbol{\beta}_1, \lambda) = \boldsymbol{\beta}_1^T \boldsymbol{S} \boldsymbol{\beta}_1 + \lambda(1 - \boldsymbol{\beta}_1^T \boldsymbol{\beta}_1) \tag{2.6}$$

Si possono ricavare le derivate parziali dell'equazione (2.6) rispetto a $\boldsymbol{\beta}_1$

$$\frac{\partial \mathcal{L}(\boldsymbol{\beta}_1, \lambda)}{\partial \boldsymbol{\beta}_1} = 2\boldsymbol{\beta}_1^T \boldsymbol{S}\boldsymbol{\beta}_1 - 2\lambda\boldsymbol{\beta}_1^T \tag{2.7}$$

e rispetto a λ

$$\frac{\partial \mathcal{L}(\boldsymbol{\beta}_1, \lambda)}{\partial \lambda} = 1 - \boldsymbol{\beta}_1^T \boldsymbol{\beta}_1 \tag{2.8}$$

Ponendo le derivate parziali dell'equazioni (2.7) e (2.8) uguali a zero si ottengono le relazioni:

$$\boldsymbol{S}\boldsymbol{\beta}_1 = \lambda\boldsymbol{\beta}_1 \tag{2.9}$$

$$\boldsymbol{\beta}_1^T \boldsymbol{\beta}_1 = 1 \tag{2.10}$$

Si può facilmente vedere dalle relazioni ottenute come $\boldsymbol{\beta}_1$ sia un autovettore della matrice di covarianza $\boldsymbol{S}$ e che quindi il moltiplicatore di Lagrange λ corrisponde all'autovalore associato all'autovettore $\boldsymbol{\beta}_1$.
Inserendo i risultati ottenuti dall'equazioni (2.9) e (2.10) nell'equazione (2.4) si ottiene la seguente relazione:

$$Var[\boldsymbol{PC1}] = \boldsymbol{\beta}_1^T \boldsymbol{S}\boldsymbol{\beta}_1 = \lambda\boldsymbol{\beta}_1^T \boldsymbol{\beta}_1 = \lambda \tag{2.11}$$

Dato che l'obiettivo è quello di trovare la direzione $\boldsymbol{\beta}_1$ dove si ha la massima varianza, allora la soluzione ottima è l'autovettore associato all'autovalore più grande della matrice di covarianza $\boldsymbol{S}$ (Marc Peter Deisenroth [2020]).
Una volta trovata la prima componente principale ($\boldsymbol{PC1}$) si procede con la ricerca della seconda componente principale. La seconda componente principale ($\boldsymbol{PC2}$) è la combinazione lineare delle variabili del dataset che riesce a catturare più informazioni possibilie che non sia correlata a $\boldsymbol{PC1}$. Si ha quindi il vincolo che $\boldsymbol{PC2}$ deve essere ortogonale a $\boldsymbol{PC1}$. Come in precedenza si ha quindi che $\boldsymbol{PC2}$ viene descritta dalla seguente relazione:

$$\boldsymbol{PC2} = \beta_{21}\boldsymbol{X}_{\{.,1\}} + \beta_{22}\boldsymbol{X}_{\{.,2\}} + ... + \beta_{2p}\boldsymbol{X}_{\{.,p\}} \tag{2.12}$$

dove i β_{2j} sono i coefficienti della combinazione lineare per la seconda componente principale. Per trovare la direzione della seconda componente principale si risolve un problema analogo al problema (2.5), dove deve essere aggiunto il vincolo che la direzione $\boldsymbol{\beta}_2$ deve essere ortogonale alla direzione $\boldsymbol{\beta}_1$.
In generale si ripete il procedimento appena descritto per trovare le componenti principali successive alla prima aggiungendo il vincolo che la direzione deve essere ortogonale alle direzioni trovate in precedenza. Il problema per la ricerca della generica K-esima

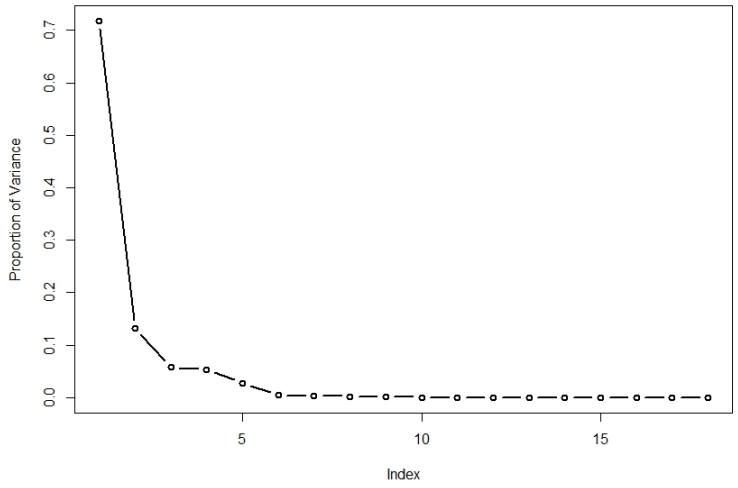

Figura 2.1. Proporzione di varianza componenti principali

componente principale si può quindi vedere nella formulazione (2.13).

$$\begin{gathered}\max_{\boldsymbol{\beta}_K} \boldsymbol{\beta}_K^T \boldsymbol{S} \boldsymbol{\beta}_K \\ tale\ che: \\ \|\boldsymbol{\beta}_K\|^2 = 1 \\ \boldsymbol{\beta}_K^T \boldsymbol{\beta}_j = 0 \quad \forall j = 1, \ldots, K-1\end{gathered} \tag{2.13}$$

2.1.1 Analisi delle componenti principali nel dataset

Il numero massimo di componenti principali che si possono individuare è il numero delle variabili originali, ma l'obiettivo della PCA è cercare di spiegare la gran parte di varianza del dataset ma con un numero inferiore di variabili. In questo caso dato che si vuole utilizzare le componenti principali per una rappresentazione grafica vengono prese in considerazione solamente le prime tre componenti principali. Dalla Figura 2.1 si può vedere la proporzione di varianza spiegata da ciascuna componente. Nella Figura 2.2 si può vedere la varianza cumulativa, cioè la somma della proporzione di varianza di ciascuna componente dalla prima alla corrente. Dai grafici si può vedere come la prima componente principale spiega il 71.8% della varianza, la seconda componente principale spiega il 13.19% della varianza e la terza componente principale spiega il 5.782% della varianza. Quindi in termini di varianza cumulativa si ha che prendendo le prime tre componenti principali si riesce a spiegare il 90.767%.

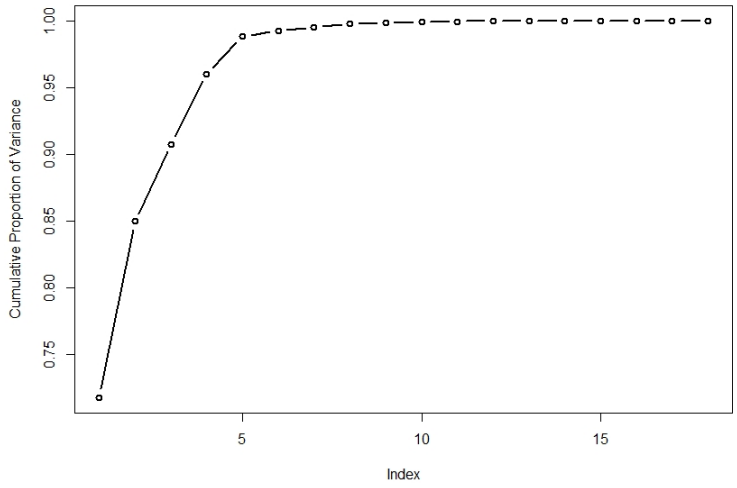

Figura 2.2. Varianza cumulativa componenti principali

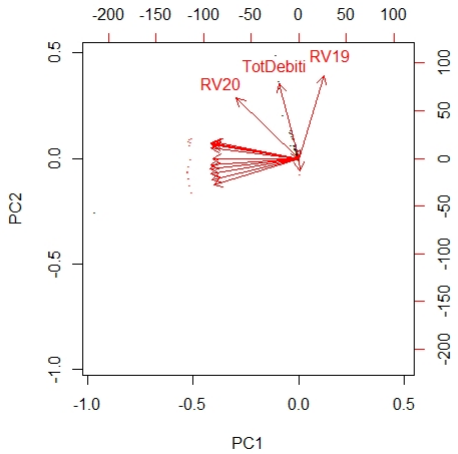

Figura 2.3. Grafico dei coefficienti di PC1 e PC2. Mostra quanto fortemente ciascuna variabile influenza PC1 e PC2

Per capire a cosa corrisponde in termini di variabili originali la posizione spaziale di un'azienda nel grafico con le componenti principali si possono utilizzare le Figure 2.3, 2.4 e la Tabella 2.1 con i coefficienti delle combinazioni lineari per le prime tre componenti principali. Si può dire quindi che ad alti valori della prima componente principale corrispondono alti valori della variabile "RV19", mentre per bassi valori della prima componente principale corrispondono valori alti per tutte le altre variabili. Dalla Tabella 2.1

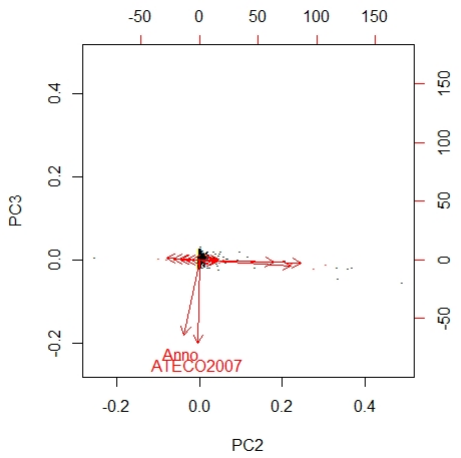

Figura 2.4. Grafico dei coefficienti di PC2 e PC3. Mostra quanto fortemente ciascuna variabile influenza PC2 e PC3

Variabile	PC1	PC2	PC3
Anno	0.004	-0.089	-0.665
ATECO2007	0.001	-0.007	-0.743
EBITDA21	-0.273	-0.112	0.003
EBITDA20	-0.275	-0.075	-0.001
EBITDA19	-0.275	-0.077	0
RV21	-0.271	0.114	-0.006
RV20	-0.194	0.437	-0.022
RV19	0.078	0.597	-0.028
PatrimonioNetto	-0.273	-0.045	-0.003
Dipendenti	-0.265	0.116	0.005
TotCrediti	-0.271	0.080	0.002
TotAttivo	-0.268	0.096	-0.016
UtileNetto	-0.270	-0.153	0.007
TotDisponLiquide	-0.267	-0.002	0.005
TotDebiti	-0.060	0.541	-0.057
CapCircNetto	-0.264	-0.193	0.015
MargineConsumi	-0.272	0.104	-0.008
FlussoCassa	-0.273	-0.115	0.004

Tabella 2.1. Coefficienti delle variabili per la combinazione lineare delle prime tre componenti principali

si può vedere che per alti valori della seconda componente principale corrispondono alti valori delle variabili “RV19”, “RV20”, “TotDebiti”, “Dipendenti”, “MargineConsumi”,

"RV21" e "TotAttivo", mentre per bassi valori della seconda componente principale corrispondono alti valori per tutte le altre variabili. Infine la terza componente principale mi riesce a spiegare solamente le variabili "Anno" e "ATECO2007", in particolare ho che per bassi valori della terza componente principale ho alti valori delle variabili "Anno" e "ATECO2007". Grazie a queste indicazioni si possono caratterizzare le aziende dalla loro posizione spaziale nei grafici che hanno come assi di riferimento le componenti principali.

2.2 K-Means

Il K-means è un algoritmo di clustering che consente di dividere il dataset in K distinti clusters (Tan [2006]). Vediamo ora nel dettaglio come lavora questo algoritmo. Per trovare la clusterizzazione l'algoritmo prevede che si parte assegnando casualmente ad ogni osservazione un'etichetta che va da 1 a K, ovvero ogni osservazione viene assegnata casualmente ad un cluster. Vengono quindi definiti gli insiemi $C_1, ..., C_K$, dove l'insieme C_i contiene gli indici delle osservazioni appartenenti all'i-esimo cluster. Questi insiemi devono soddisfare le seguenti condizioni:

1. $C_1 \cup \ldots \cup C_K = \{1, ..., N\}$, dove N è il numero di osservazioni del dataset.
2. $C_i \cap C_j \neq \emptyset$, per ogni $i \neq j$.

Si passa ora alla seconda fase dell'algoritmo. Per ogni cluster viene calcolato il relativo centroide, dove il centroide del cluster k è un vettore di dimensione p, e alla posizione p-esima del vettore si ha la media della p-esima variabile per le osservazioni nel cluster k. A questo punto si assegna ogni osservazione al cluster il cui centroide è più vicino in termini di distanza Euclidea

$$d(\boldsymbol{X}_{\{i,.\}}, \boldsymbol{X}_{\{k,.\}}) = \sqrt{\sum_{j=1}^{p} \left(X_{\{i,j\}} - X_{\{k,j\}}\right)^2} \tag{2.14}$$

dove $X_{\{i,.\}}$ e $X_{\{k,.\}}$ sono due osservazioni distinte del dataset.
Viene utilizzata la distanza Euclidea dato che le variabili considerate sono variabili numeriche. La seconda fase si ripete fino a quando le osservazioni non cambiano cluster. Questo algoritmo è un problema di ottimizzazione dove ad ogni iterazione si vuole minimizzare la distanza dei punti all'interno di uno stesso cluster.
In questo algoritmo per scegliere la configurazione iniziale più vicina possible a quella ideale vengono generate più configurazioni iniziali casuali e tra queste viene scelta la migliore per l'algoritmo K-Means. Per migliore si intende la configurazione che porta ad un risultato finale che ha il valore minimo di SSE

$$SSE = \sum_{i=1}^{K} \sum_{j \in C_i} \left(d(\boldsymbol{X}_{\{j,.\}}, \boldsymbol{m_i})\right)^2 \tag{2.15}$$

dove $\boldsymbol{X}_{\{j,.\}}$ è un osservazione del dataset, C_i è l'insieme degli indici delle osservazioni appartenenti all'i-esimo cluster e $\boldsymbol{m_i}$ è il centroide dell'i-esimo cluster.
In questo algoritmo abbiamo che il numero di cluster che vengono generati viene scelto a

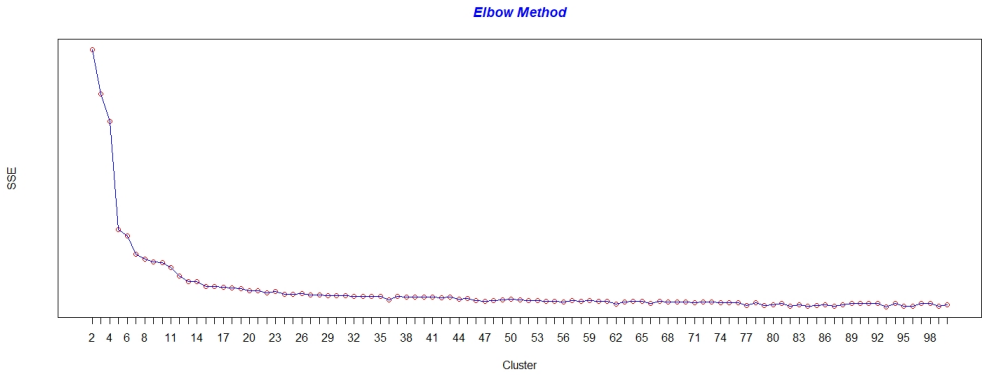

Figura 2.5. Calcolo SSE in funzione del numero di cluster generati tramite l'algoritmo K-Means

priori ed è pari al parametro K. Per la scelta più ottimale di questo parametro si vanno ad analizzare diversi fattori. Per prima cosa si va ad individuare quale scelta di K porta ad avere il più piccolo sum of square error (SSE), Formula (2.15). Dalla equazione (2.15) si vede quindi che voler avere un SSE piccolo equivale a voler avere per ogni osservazione appartenente ad un dato cluster la più piccola distanza dal proprio centroide.
Nella Figura 2.5, si può quindi vedere il valore del SSE in relazione al parametro K che viene utilizzato. Naturalmente all'aumentare del parametro K diminuisce il valore del SSE, però scegliere un parametro K troppo grande potrebbe portare ad un errore di overfitting. Per evitare questo problema si individua il valore di K ideale nel punto di gomito della Figura 2.5. Si sceglie il punto di gomito perchè è dove cambia la pendenza della retta che indica l'errore SSE, si passa da un'alta pendenza ad una bassa pendenza, il che significa che aggiungere un cluster non porta più ad avere un vantaggio rilevante. Dalla Figura 2.5 possiamo individuare diversi valori K come punto di gomito, ovvero K: 5, 7, 8, 10, 12, 13, 15. Per decidere quale valore di K utilizzare per l'algoritmo K-Means, tra i possibili valori individuati come punto di gomito, si valutano altri due fattori. Un fattore è la composizione dei cluster generati con questi diversi parametri K, che si può osservare nelle Figure 2.7,2.8, 2.9, 2.10, 2.11, 2.12, 2.13 e nelle Tabelle 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8. Analizzando la composizione dei cluster generati si può notare la presenza di due cluster composti da poche aziende per ogni valore di K. Dato che K pari a 5 è stato il più piccolo valore considerato, questa informazione potrebbe suggerire l'utilizzo di K pari a 3. Visualizzando la rappresentazione grafica dei cluster generati con K pari a 5, Figura 2.6, si può però osservare che le aziende appartenenti ai cluster 3 e 4, ovvero i cluster composti da poche aziende, sono molto distanti dalle altre. Questo risultato fornisce l'informazione che le aziende appartenenti ai cluster poco numerosi hanno caratteristiche diverse rispetto alle altre e che quindi è giusto assegnarle a dei cluster diversi anche se poco numerosi. Questo conferma quanto riportato dallo studio dei possibili outliers tramite la distanza di Malhanobis, Sezione 1.2, ovvero che le aziende distanti dalle altre non sono valori anomali ma appartengono alla sottocategoria delle multinazionali del settore considerato.

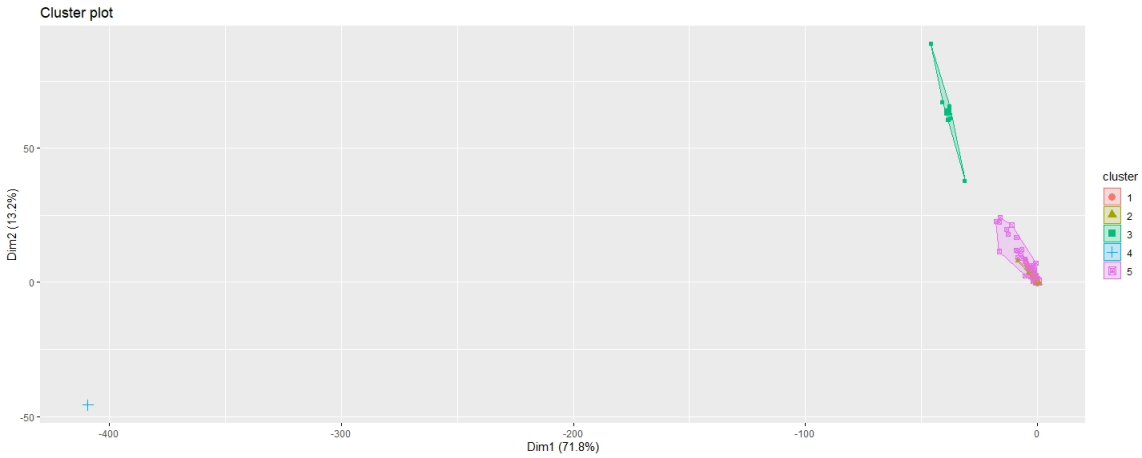

Figura 2.6. Rappresentazione grafica 2D dei cluster ottenuti con l'algoritmo K-Means con K=5

In conclusione quindi si può affermare che è giusto andare a considerare i K individuati tramite il punto di gomito della Figura 2.5. Il secondo fattore che viene considerato per individuare il valore K da utilizzare nell'algoritmo K-Means è l'indice Silhouette in funzione del numero di cluster, che si può osservare nella Figura 2.14. L'indice Silhouette indica la correttezza dell'appartenenza delle osservazioni all'interno dei cluster, quindi viene considerata la media degli indici calcolati per ogni osservazione e più è grande il valore dell'indice medio maggiore è la bontà dei cluster. Il calcolo dell'indice Silhouette viene poi approfondito nel dettaglio nella Sezione (2.4). Confrontando l'indice medio della silhouette in funzione del numero di cluster, per i K individuati come punto di gomito, si può vedere dalla Figura 2.14 che i risultati migliori si hanno per K pari a 5 e per K pari a 8. Come detto in precedenza però si deve scegliere K tra quelli individuati come punto di gomito nella Figura 2.5, che combini alto valore dell'indice medio Silhouette e con cui si generano cluster abbastanza omogenei in dimensione. Andando a valutare quindi anche la composizione dei cluster generati con K pari a 5 si può vedere dalla Tabella 2.8 e dalla Figura 2.14 che vengono generati cluster poco omogenei, in particolare si ottiene un cluster composto da più della metà delle aziende. Allo stesso modo andando a valutare la composizione dei cluster generati con K pari a 8 si può vedere dalla Tabella 2.6 e dalla Figura 2.11 che vengono generati cluster poco omogenei, in particolare si ottiene un cluster composto da circa la metà delle aziende. Questo risultato non è ottimale ed inoltre per l'analisi è utile avere cluster poco numerosi per diversificare bene le caratteristiche delle aziende che vi appartengono. Il K che riesce a soddisfare al meglio il trade-off appena descritto, ovvero che sia punto di gomito della Figura 2.5, che ha un buon valore dell'indice medio Silhouette, che generi cluster poco numerosi e abbastanza omogenei in dimensione, è quindi K pari a 15.
Si rappresentano ora con l'utilizzo delle componenti principali i clusters generati dall'algoritmo K-Means con K=15, Figure 2.15 e 2.16.

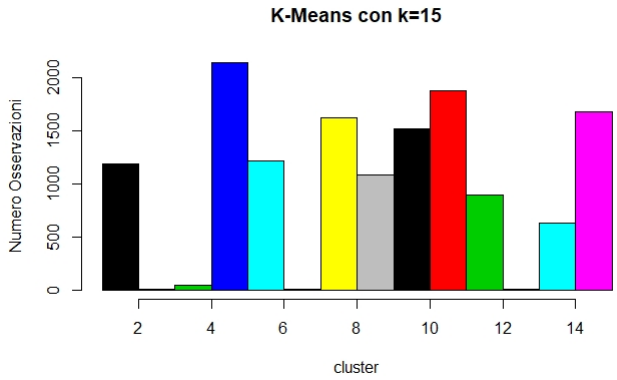

Figura 2.7. Numero osservazioni nei cluster ottenuti dall'algoritmo K-Means con K=15

Cluster	Numero osservazioni
1	4
2	1181
3	9
4	45
5	2136
6	1218
7	1
8	1624
9	1077
10	1516
11	1872
12	891
13	1
14	631
15	1674

Tabella 2.2. Numero osservazioni nei cluster ottenuti dall'algoritmo K-Means con K=15

In particolare analizzando da un punto di vista qualitativo le aziende che si discostano maggiormente dalla grande maggioranza, nelle Figure 2.15 e 2.16, si può vedere che si tratta delle multinazionali del settore considerato. Questo conferma quanto riportato dallo studio dei possibili outliers tramite la distanza di Malhanobis, Sezione 1.2. Successivamente verranno utilizzati i grafici costruiti con le componenti principali per caratterizzare al meglio i cluster dove sono presenti i possibili clienti. Infine nella Tabella 2.9 viene riportata la percentuale dei clienti all'interno di ogni cluster per stabilire quale tra questi cluster possono essere classificati come cluster dei possibili clienti. Per capire quali sono i cluster individuati come possibili clienti si confronta la percentuale trovata per ogni

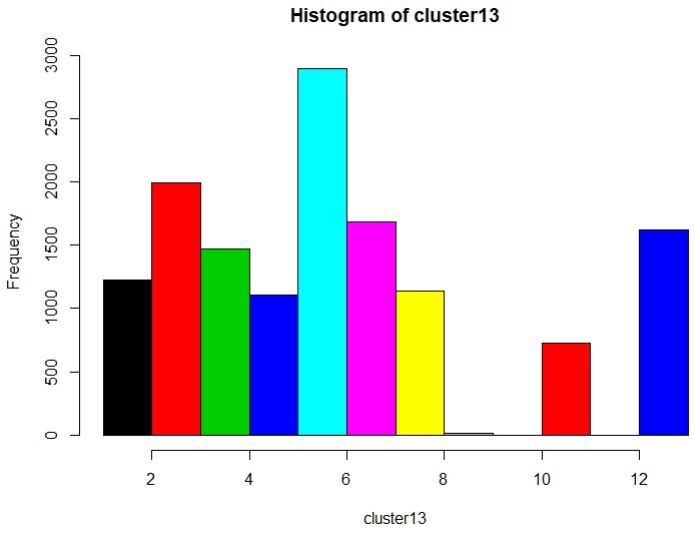

Figura 2.8. Numero osservazioni nei cluster ottenuti dall'algoritmo K-Means con K=13

Cluster	Numero osservazioni
1	4
2	1221
3	1996
4	1470
5	1110
6	2895
7	1686
8	1136
9	15
10	1
11	724
12	1
13	1621

Tabella 2.3. Numero osservazioni nei cluster ottenuti dall'algoritmo K-Means con K=13

cluster nella Tabella 2.9, con il target dell'equazione (2.1). Da questo confronto vengono etichettati i cluster 1, 3, 4, 8, 9, 14 e 15 come cluster dei possibili clienti poiché hanno una percentuale superiore al target.

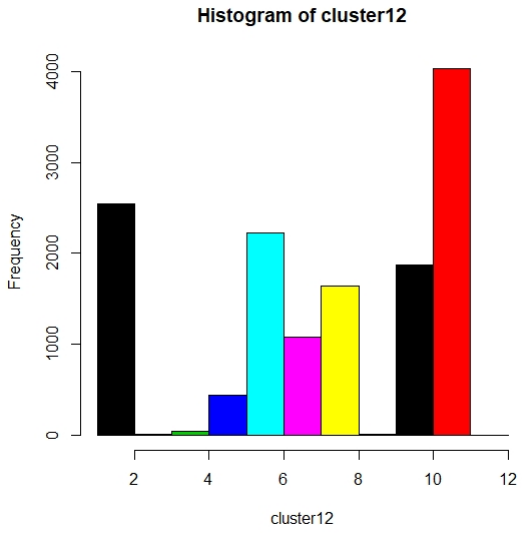

Figura 2.9. Numero osservazioni nei cluster ottenuti dall'algoritmo K-Means con K=12

Cluster	Numero osservazioni
1	821
2	1719
3	9
4	45
5	438
6	2225
7	1074
8	1638
9	5
10	1872
11	4033
12	1

Tabella 2.4. Numero osservazioni nei cluster ottenuti dall'algoritmo K-Means con K=12

2.3 DBSCAN

L'algoritmo DBSCAN è un algoritmo basato sulla densità dei punti, dove con densità si intende il numero di punti all'interno di una regione di spazio di raggio Epsilon (Tan [2006]).
Si comincia con un punto scelto in maniera casuale e viene calcolato il suo Epsilon-vicinato. Per Epsilon-vicinato dell'osservazione $\boldsymbol{X}_{\{i,.\}}$ si intende l'insieme delle osservazioni $\boldsymbol{X}_{\{k,.\}}$,

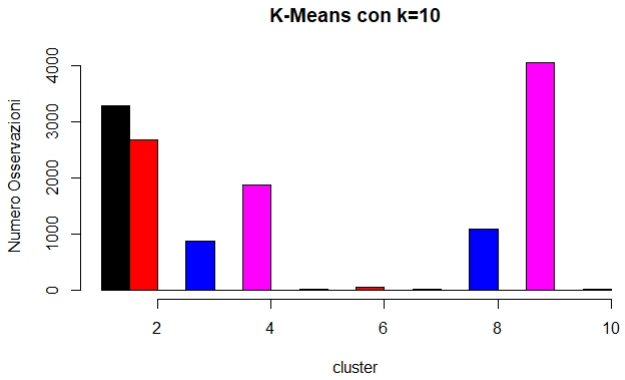

Figura 2.10. Numero osservazioni nei cluster ottenuti dall'algoritmo K-Means con K=10

Cluster	Numero osservazioni
1	3278
2	2679
3	869
4	1872
5	1
6	45
7	9
8	1078
9	4044
10	5

Tabella 2.5. Numero osservazioni nei cluster ottenuti dall'algoritmo K-Means con K=10

con $k \neq i$, che soddisfano

$$d(\boldsymbol{X}_{\{i,.\}}, \boldsymbol{X}_{\{k,.\}}) = \sqrt{\sum_{j=1}^{p} \left(X_{\{i,j\}} - X_{\{k,j\}}\right)^2} \leq Epsilon \tag{2.16}$$

Se l'Epsilon-vicinato contiene un numero sufficiente di punti, indicato con MinPoints, viene creato un nuovo cluster di cui ne farà parte. Se ciò non avviene il punto viene etichettato come rumore e successivamente potrebbe essere ritrovato in un Epsilon-vicinato sufficientemente grande riconducibile ad un punto differente entrando a far parte di un cluster. Se invece viene creato un nuovo cluster tutti i punti appartenenti al suo Epsilon-vicinato faranno parte di quel cluster. Questo processo si ripete per tutti i punti che quando vengono visitati ancora non appartengono ad un cluster, fino a quando ogni punto non viene assegnato ad un cluster. I punti che non vengono assegnati ad un cluster perché troppo distanti dagli altri punti finiscono nel cluster 0.

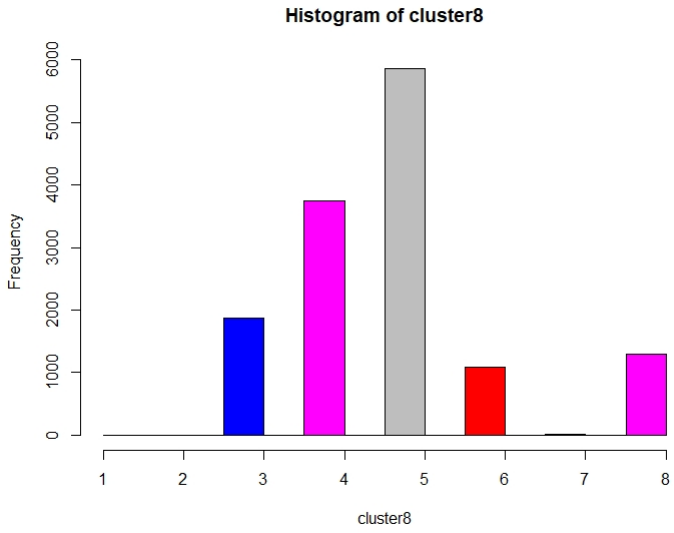

Figura 2.11. Numero osservazioni nei cluster ottenuti dall'algoritmo K-Means con K=8

Cluster	Numero osservazioni
1	1
2	5
3	1875
4	3744
5	5861
6	1080
7	15
8	1299

Tabella 2.6. Numero osservazioni nei cluster ottenuti dall'algoritmo K-Means con K=8

Vediamo quindi come anche questo algoritmo necessita di due parametri di input che sono il MinPoints, ovvero il numero minimo di punti che servono per creare un cluster ed Epsilon, ovvero il raggio che individua la regione in cui si deve cercare i punti per creare il cluster. Per trovare i parametri MinPoints ed Epsilon viene utilizzato il grafico K-NN dove i punti vengono ordinati in base alla distanza dal K-esimo vicino. Per scegliere K vengono plottati tutti i grafici K-NN partendo da K=1 fino a quando il grafico non cambia. Viene utilizzata questa tecnica poiché significa che le distanza dal K vicino e dal K+1 vicino sono molto simili e quindi, fissato Epsilon, i cluster generati con MinPoints uguale a K o con MinPoints uguale a K+1 saranno gli stessi. Per l'analisi è stato scelto K uguale a 30 poiché da questo valore il cambiamento del grafico da K uguale a 30 e K uguale a 31 è veramente minimo come si può vedere nelle Figure 2.17 e 2.18.

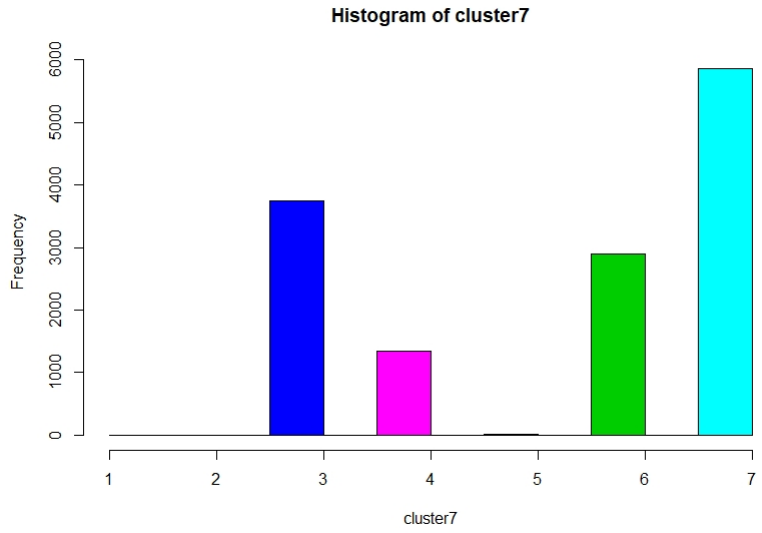

Figura 2.12. Numero osservazioni nei cluster ottenuti dall'algoritmo K-Means con K=7

Cluster	Numero osservazioni
1	1
2	5
3	3747
4	1347
5	15
6	2904
7	5861

Tabella 2.7. Numero osservazioni nei cluster ottenuti dall'algoritmo K-Means con K=7

Una volta scelto il parametro MinPoints da utilizzare si va ad individuare il parametro Epsilon mediante la ricerca del punto di gomito nel grafico K-NN, con K uguale a MinPoints. Si individua Epsilon nel punto di gomito poiché sopra quel valore le distanze dal K-esimo punto aumentano sensibilmente. Seguendo la logica dell'individuazione dei parametri MinPoints ed Epsilon ci si aspetta che i punti interni al cluster avranno una K-distance minore di Epsilon mentre i punti del "cluster 0" avranno una K-distance maggiore di Epsilon, con K uguale a MinPoints.
Sono stati generati i clusters considerando come valori $Minpoints = 30$ e $Epsilon = 0.1$, ottenendo 78 cluster ben proporzionati. Nel "cluster 0" finiscono invece 1909 aziende, che quindi per quanto descritto in precedenza vengono considerate come osservazioni rumorose. Avevamo però visto in precedenza, con lo studio dei possibili outliers nella Sezione 1.2, che queste osservazioni non sono rumorose ma questo risultato è dovuto dalla natura

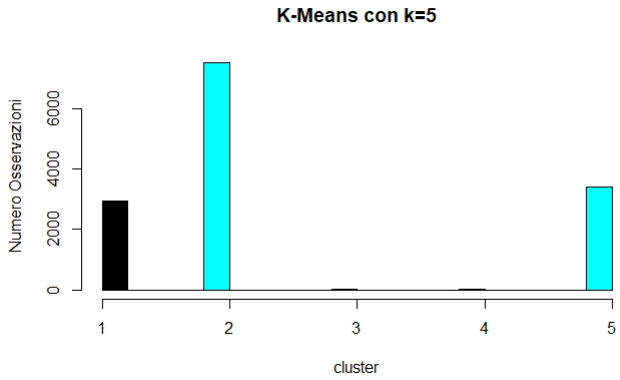

Figura 2.13. Numero osservazioni nei cluster ottenuti dall'algoritmo K-Means con K=5

Cluster	Numero osservazioni
1	2932
2	7531
3	6
4	1
5	3410

Tabella 2.8. Numero osservazioni nei cluster ottenuti dall'algoritmo K-Means con K=5

del dominio di applicazione. Si potrebbe provare ad utilizzare un valore di Epsilon più grande per evitare di considerare osservazioni come rumorose anche se non lo sono. Così facendo però si ottengono risultati poco utili al fine dell'analisi perchè vengono generati pochi cluster e di grandi dimensioni difficili poi da caratterizzare. Questo problema si può vedere anche graficamente tramite la rappresentazione dei cluster trovati con l'algoritmo DBSCAN nella Figura 2.19.

2.4 Confronto tra K-Means e DBSCAN

Per valutare la bontà dei cluster ottenuti con i diversi algoritmi sono stati utilizzati indici interni dato che si tratta di un'analisi esplorativa e a priori non si conoscono i risultati. In particolare sono stati utilizzati l'indice Silhouette (J.Rousseeuw [1964]) e l'indice Dunn (Eréndira Rendón and Quiroz [2011]). Nel dettaglio l'indice Silhouette, come detto in precedenza, indica la correttezza dell'appartenenza delle osservazioni all'interno dei cluster, più è grande il valore dell'indice maggiore è la bontà dei cluster. L'indice Silhouette viene calcolato per ogni osservazione del dataset e tiene conto sia della distanza dei punti

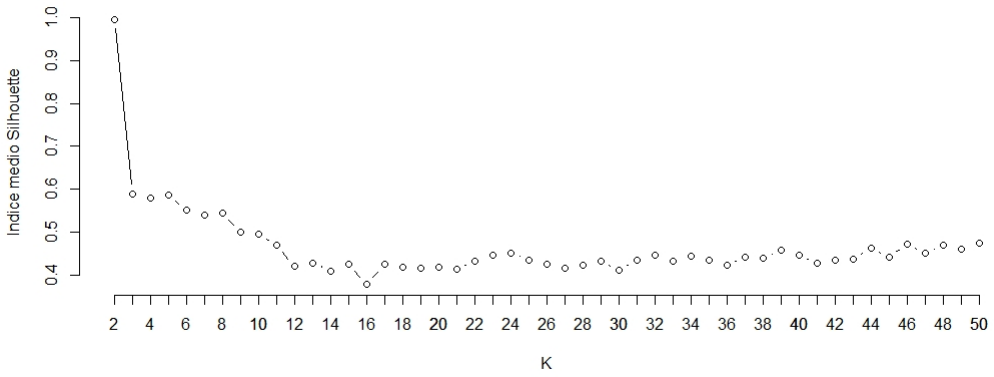

Figura 2.14. Indice Silhouette medio calcolato con i cluster ottenututi tramite gli algoritmi K-Means in funzione del numero di cluster K utilizzati

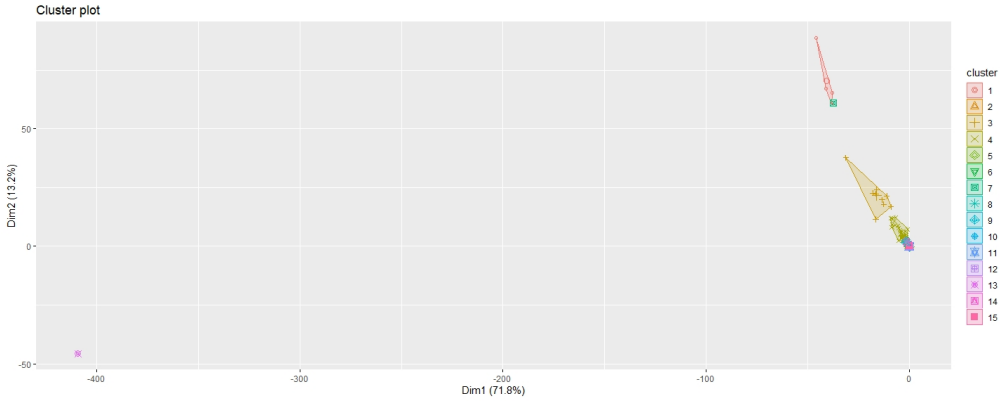

Figura 2.15. Rappresentazione grafica 2D dei cluster ottenuti con l'algoritmo K-Means con K=15

all'interno dei cluster:

$$a(\boldsymbol{X}_{\{k,.\}}) = \frac{1}{|C_i| - 1} \sum_{q \neq k} d(\boldsymbol{X}_{\{k,.\}}, \boldsymbol{X}_{\{q,.\}}) \tag{2.17}$$

sia della distanza tra cluster:

$$b(\boldsymbol{X}_{\{k,.\}}) = \min_{j \neq i} \frac{1}{|C_j|} \sum_{q \in C_j} d(\boldsymbol{X}_{\{k,.\}}, \boldsymbol{X}_{\{q,.\}}) \tag{2.18}$$

dove $\boldsymbol{X}_{\{k,.\}}$ è il vettore p-dimensionale composto dalle variabili numeriche del record k appartenente all'i-esimo cluster e $d(\boldsymbol{X}_{\{k,.\}}, \boldsymbol{X}_{\{q,.\}})$ è la quantità dell'equazione (2.14).

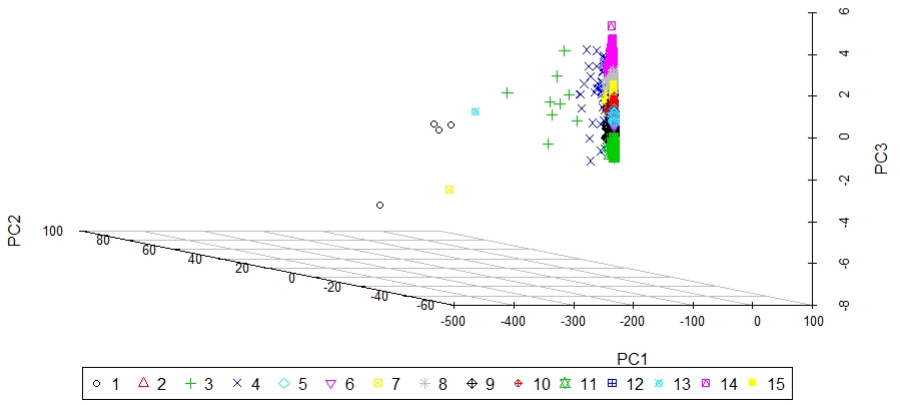

Figura 2.16. Rappresentazione grafica 3D dei cluster K-Means ottenuti con l'algoritmo K-Means con K=15

Cluster	Percentuale Clienti
1	100 %
2	0.68 %
3	33.33 %
4	37.78 %
5	0.33 %
6	0.74 %
7	0.00 %
8	1.91 %
9	1.39 %
10	1.32 %
11	0.43 %
12	1.23 %
13	0.00 %
14	2.69 %
15	2.03 %

Tabella 2.9. Percentuale clienti nei cluster ottenuti con l'algoritmo K-Means con K=15

Combinando le equazioni (2.17) e (2.18) si ottiene l'indice Silhouette

$$s(\boldsymbol{X}_{\{k,.\}}) = \begin{cases} 1 - \frac{a(\boldsymbol{X}_{\{k,.\}})}{b(\boldsymbol{X}_{\{k,.\}})} & se \quad a(\boldsymbol{X}_{\{k,.\}}) < b(\boldsymbol{X}_{\{k,.\}}) \\ 0 & se \quad a(\boldsymbol{X}_{\{k,.\}}) = b(\boldsymbol{X}_{\{k,.\}}) \\ \frac{b(\boldsymbol{X}_{\{k,.\}})}{a(\boldsymbol{X}_{\{k,.\}})} - 1 & se \quad a(\boldsymbol{X}_{\{k,.\}}) > b(\boldsymbol{X}_{\{k,.\}}) \end{cases} \tag{2.19}$$

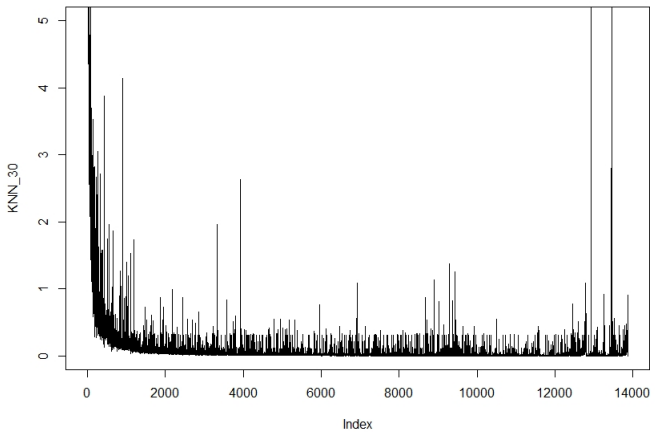

Figura 2.17. K-NN con K=30

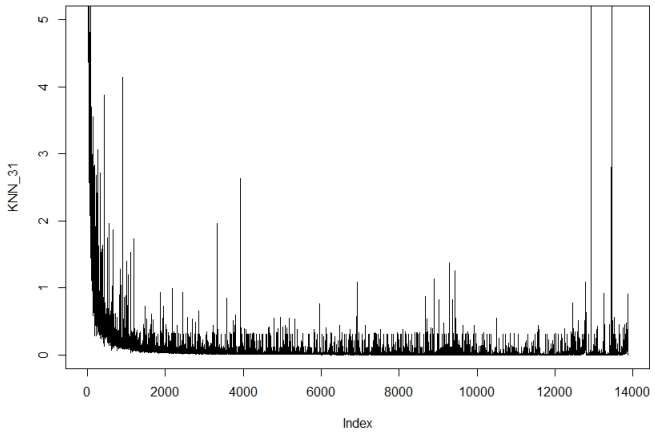

Figura 2.18. K-NN con K=31

Dato che l'indice Silhouette viene calcolato per ogni osservazione viene poi considerata la sua media per avere un'indicazione sulla bontà dei cluster.
Anche l'indice Dunn indica la correttezza dell'appartenenza delle osservazioni all'interno dei cluster, più è piccolo il valore dell'indice maggiore è la bontà dei cluster. L'indice Dunn tiene conto sia della distanza dei punti all'interno dei cluster

$$\Delta_i = \max_{k,j \in C_i} d(\boldsymbol{X}_{\{k,.\}}, \boldsymbol{X}_{\{j,.\}}) \tag{2.20}$$

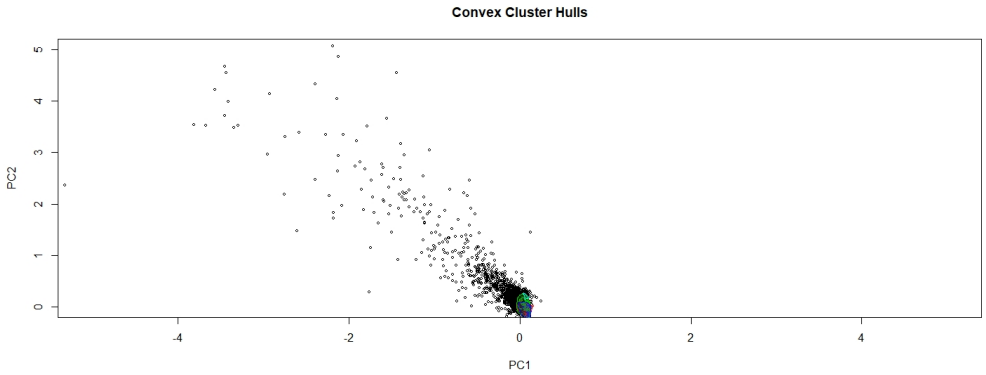

Figura 2.19. Rappresentazione grafica 2D dei cluster ottenuti con l'algoritmo DBSCAN

	Indice Silhouette medio
K-Means	0.4296052
DBSCAN	0.4921417

Tabella 2.10. Indice Silhouette medio calcolato con i cluster ottenututi tramite gli algoritmi K-Means con K=15 e DBSCAN

sia della distanza tra cluster

$$\delta(C_i, C_j) = d(\boldsymbol{m_i}, \boldsymbol{m_j}) \tag{2.21}$$

dove m_i e m_j sono rispettivamente i centroidi dell'i-esimo e del j-esimo cluster. Combinando le equazioni (2.20) e (2.21) si ottiene l'indice Dunn:

$$DI_M = \frac{\min_{1 \leq i \leq j \leq M} \delta(C_i, C_j)}{\max_{1 \leq k \leq M} \Delta_k} \tag{2.22}$$

dove M è il numero di cluster e $\delta(C_i, C_j)$ identifica la distanza tra il centroide dell'i-esimo cluster e il centroide del j-esimo cluster. Nella Tabella 2.10 vengono riportati i valori dell'indice Silhouette medio e nella Tabella 2.11 i valori dell'indice Dunn, per i cluster ottenuti con il K-Means e con il DBSCAN.
Dai risultati che si possono vedere nelle Tabelle 2.10 e 2.11 sembra performare meglio l'algoritmo DBSCAN. Ci si poteva aspettare questo risultato perché l'algoritmo DBSCAN lavora direttamente sulle distanze delle osservazioni nel dataset.
Tenendo però in considerazione le osservazioni negative fatte in precedenza sulla costruzione dei cluster con l'algoritmo DBSCAN e dato che i valori degli indici per il K-Means sono comunque buoni si vanno ad utilizzare i cluster ottenuti con il K-Means per caratterizzare e identificare i possibili clienti.
Si procede ora a caratterizzare i cluster dei possibili clienti trovati tramite l'algoritmo K-Means, quindi i cluster 1, 3, 4, 8, 9, 14 e 15. Nella Sezione 2.2 troviamo le Figure 2.15 e

	Indice Dunn
K-Means	0.0006581031
DBSCAN	0.0000487022

Tabella 2.11. Indice Dunn calcolato con i cluster ottenututi tramite gli algoritmi K–Means con K=15 e DBSCAN

Cluster	PC1	PC2	PC3
Cluster 1	Valori bassi	Valori alti	Valori bassi
Cluster 3	Valori bassi	Valori alti	Valori nulli
Cluster 4	Valori bassi	Valori alti	Valori nulli
Cluster 8	Valori nulli	Valori bassi	Valori alti
Cluster 9	Valori nulli	Valori bassi	Valori alti
Cluster 14	Valori nulli	Valori bassi	Valori alti
Cluster 15	Valori nulli	Valori bassi	Valori alti

Tabella 2.12. Valori componenti principali nei cluster dei possibili clienti

2.16 dove vengono mostrate le osservazioni differenziate in base al colore e alla forma per cluster di appartenenza. In particolare si può riassumere nella Tabella 2.12 la posizione spaziale dei cluster dei possibili clienti, nel grafico che ha come assi di riferimento le prime tre componenti principali. Grazie alle considerazioni fatte nella Sezione 2.1.1 tramite le Figure 2.3, 2.4 e la Tabella 2.1 con i coefficienti delle componenti principali si possono quindi caratterizzare i cluster dei possibili clienti in base alla loro posizione spaziale riassunta nella Tabella 2.12. Nei cluster 1, 3 e 4 si hanno aziende con valori alti per le variabili “RicaviVendite2020”, “TotDebiti”, “Dipendenti2021”, “MargineConsumi2021”, “RicaviVendite2021”, “TotAttivo2021”. Nei restanti cluster dei possibili clienti per queste variabili non si hanno informazioni.
Nei cluster 8, 9, 14 15 si ottengono le informazioni grazie ai valori alti di PC3, ovvero troviamo aziende con valori negativi delle variabili “Anno” e “ATECO2007” standardizzate. Quindi aziende che sono nel mercato da più anni e che appartengono a sottocategorie con il codice ATECO medio bassi tra quelli presenti nel dataset, ovvero le sottocategorie "Consulenza nel settore delle tecnologie dell’informatica" e "Gestione di strutture e apparecchiature informatiche hardware".

Capitolo 3

Classificazione

3.1 Rielaborazione del dataset

Tramite la cluster analysis è stata ottenuta un'indicazione riguardante le aziende che possono essere potenziali clienti.
Per rendere il progetto più completo si è pensato di costruire modelli di classificazione in modo tale da poter predire in futuro aziende non presenti nel dataset come possibili clienti o meno. Naturalmente per le nuove aziende che si vorranno classificare devono valere le seguenti ipotesi: avere la sede legale a Roma o a Milano oppure a Torino e di far parte della categoria consulenza informatica e produzione di software.
La classificazione è un processo che si usa per assegnare un'osservazione ad una classe. Per prima cosa è stata costruita una nuova variabile binaria ("LabelClienti") che vale "1" se l'azienda appartiene ad uno dei cluster dei possibili clienti, altrimenti vale "0". Una volta creata la nuova variabile si passa all'analisi esplorativa, in particolare si possono vedere le distribuzioni delle variabili numeriche nei due diversi valori di "LabelClienti", tramite i boxplot nella Figura 3.1.
Dai Boxplot si può vedere che la variabile "Anno" è la variabile che maggiormente tende ad avere valori diversi nelle due diverse classi della variabile "LabelClienti". Ci si poteva aspettare questo risultato poichè tramite l'analisi grafica dei cluster individuati come possibili clienti la variabile "Anno" rappresentava la misura che maggiormente riusciva a caratterizzarli. Questo ci può dare un'indicazione riguardo al fatto che la variabile "Anno" potrà essere più significativa rispetto alle altre variabili per l'analisi di classificazione.
Inoltre prima di iniziare l'analisi è stato verificato che il dataset fosse abbastanza bilanciato rispetto alla variabile "LabelClienti". In particolare si ha che circa il 40% delle aziende sono etichettate come possibili clienti, mentre il restante 60% non appartiene a quella categoria.
Per studiare i modelli di classificazione è stato diviso il dataset in due sottoinsiemi chiamati training set e testing set. Il primo viene utilizzato per costruire i modelli e rappresenta il 70% del set di dati. Il secondo viene utilizzato per vedere come si comportano i classificatori e rappresenta la parte rimanente del set di dati. Viene utilizzato questo metodo per ottenere una corretta valutazione dell'errore di classificazione ottenuto con i diversi

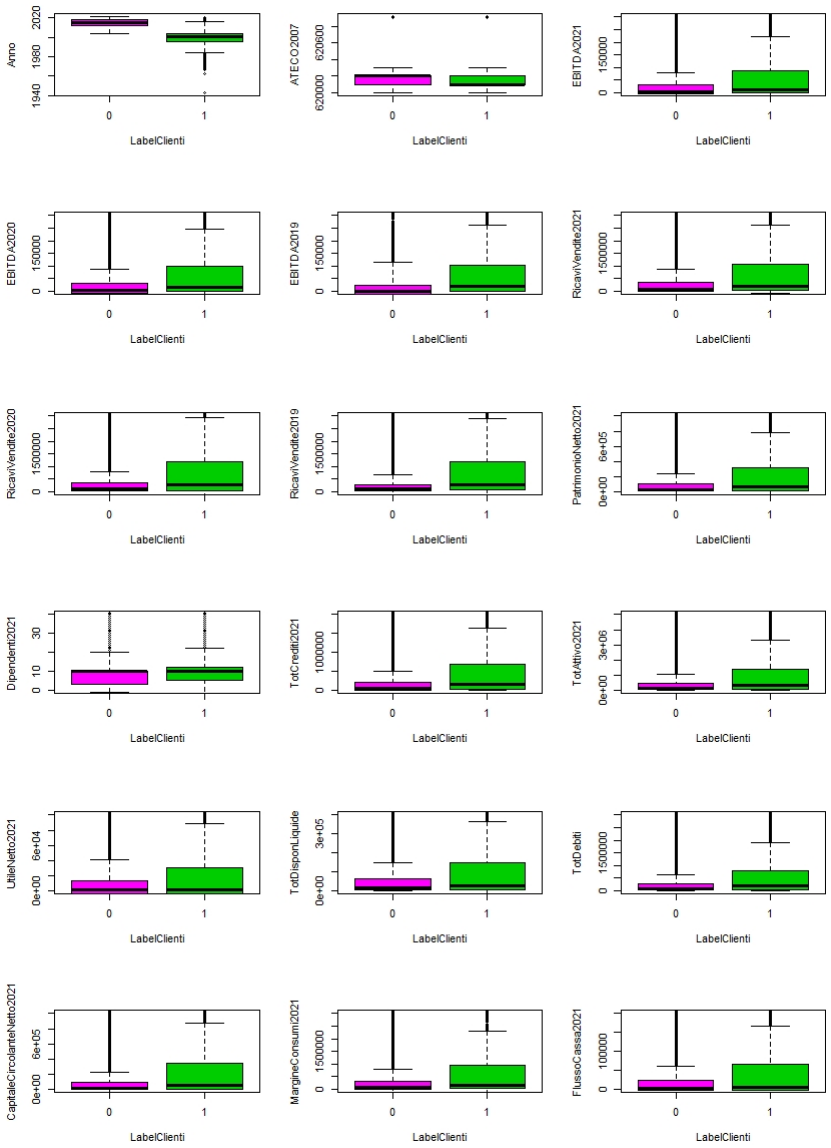

Figura 3.1. Distribuzione delle variabili numeriche rispetto alla variabile "LabelClienti"

modelli. Convalidando il modello sullo stesso set di dati con cui è stato costruito si ottine infatti un errore inferiore a quello che si ha utilizzando un diverso set di dati.

3.2 Support Vector Machines

Il classificatore Support Vector Machines è stato costruito sia con l'utilizzo del dataset originale sia con il dataset che ha come variabili le prime tre componenti principali introdotte nel capitolo precedente. Viene fatto questo per visionare graficamente come lavora questo classificatore. È stato quindi creato un training set e un testing set per entrambi i dataset utilizzando lo stesso "indice di riga" in modo da avere un giusto confronto. Naturalmente nella costruzione del training set e del testing set è stato verificato che la distribuzione di "LabelClienti" nei sottoinsiemi generati rimanesse simile a quella del dataset completo.
Si vede ora come viene costruito il classificatore. Con i dati presenti nel training set viene trovato l'iperpiano, nello spazio p-dimensionale dove p è il numero delle variabili, che separa al meglio le due classi della variabile "LabelClienti". Si può vedere SVM come un problema di ottimizzazione dove si vuole massimizzare la distanza tra i dati e l'iperpiano che li divide nelle due classi di "LabelClienti" a cui appartengono. Per spiegare la tecnica SVM si parte dal problema del classificatore del margine massimo. La soluzione ottimale di questo problema viene chiamata iperpiano di margine massimo.

$$
\begin{aligned}
&maxM \\
&tale\ che: \\
&\sum_{j=1}^{p} \beta_j^2 = 1 \\
&Y_i(\beta_0 + \beta_1 X_{i,1} + \beta_2 X_{i,2} + ... + \beta_p X_{i,p}) \geq M \qquad \forall i = 1, \ldots, n
\end{aligned}
\tag{3.1}
$$

dove n è il numero osservazioni del training set, Y_i è l'etichetta di classe dell'osservazione i-esima, p è il numero di predittori.
Il secondo vincolo del problema (3.1), garantisce che ogni osservazione è sul lato corretto dell'iperpiano e almeno ad una distanza M dall'iperpiano. In molti casi non esiste un iperpiano separatore in grado di separare le due classi, e quindi non esiste un classificatore di margine massimo.

3.2.1 Linear SVM

Per risolvere questo problema viene introdotto il **"soft margin"**. In questo caso possono essere ammesse qualche violazioni, in particolare viene scelto l'iperpiano che separa la maggior parte delle osservazioni nelle due classi, ma potrebbe mal classificare qualche

osservazione.

$$
\begin{aligned}
&maxM \\
&tale\ che: \\
&\sum_{j=1}^{p} \beta_j^2 = 1 \\
&Y_i(\beta_0 + \beta_1 X_{i,1} + \beta_2 X_{i,2} + ... + \beta_p X_{i,p}) \geq M(1 - \epsilon_i) \qquad \forall i = 1, \ldots, n \\
&\epsilon_i \geq 0 \\
&\sum_{i=1}^{n} \epsilon_i \leq C
\end{aligned}
\tag{3.2}
$$

Nella formulazione (3.2) vengolo aggiunte le variabili ϵ_i e il parametro C. Le variabili ϵ_i permettono alle singole osservazioni di trovarsi dalla parte sbagliata del margine o dell'iperpiano. In particolare se

$$\epsilon_i > 0$$

allora l'i-esima osservazione è dalla parte sbagliata del margine, e se

$$\epsilon_i > 1$$

l'i-esima osservazione è dalla parte sbagliata dell'iperpiano.
Il parametro C, chiamato anche parametro di costo, serve per limitare il numero di violazioni. Per questo motivo il primo passo è trovare il valore ottimale di C tramite la k-fold cross validation. Quello che viene fatto con la k-fold cross validation è dividere il training set in k sottoinsiemi di uguali dimensioni. Vengono utilizzati k-1 sottoinsiemi come set di allenamento e il sottoinsieme rimanente come testing set. Viene calcolato l'errore sul testing set, in questo caso l'errore è la percentuale di osservazioni mal classificate, e ripetuto il tutto per k volte, ogni volta utilizzando come testing set un sottogruppo differente. La k-fold cross validation è la media degli errori ottenuti. Viene ripetuto questo procedimento per ogni valore di C, in modo tale che viene scelto come C ottimale quel parametro con cui si ha l'errore medio più basso. Si può vedere anche graficamente come il valore ottimale, ovvero il valore per cui si ha il minimo errore, è per il dataset originale C = 0.04, Figura 3.2, mentre per il dataset costruito con le componenti principali è C = 16.2, Figura 3.3.
Grazie alla dimensionalità minore e alla grande varianza spiegata dalle prime componenti principali si può vedere graficamente come funziona il classificatore, nelle Figure 3.4 e 3.5. Nelle Figure vengono rappresentati infatti i support vectors, ovvero i punti che rimangono sul margine o nella parte sbagliata del margine per la loro etichetta. In particolare si può vedere nella Figura 3.4 che come assi di riferimento le prime due componenti principali che il classificatore funziona molto bene mentre la Figura 3.5 che ha come assi di riferimento la seconda e la terza componente principale non da alcuna informazione. Si ottiene questo risultato perché le prime due componenti principali spiegano la parte più grande della varianza mentre le componenti principali 2 e 3 spiegano solo una piccola parte della varianza, per questo la separazione grafica è meno evidente.

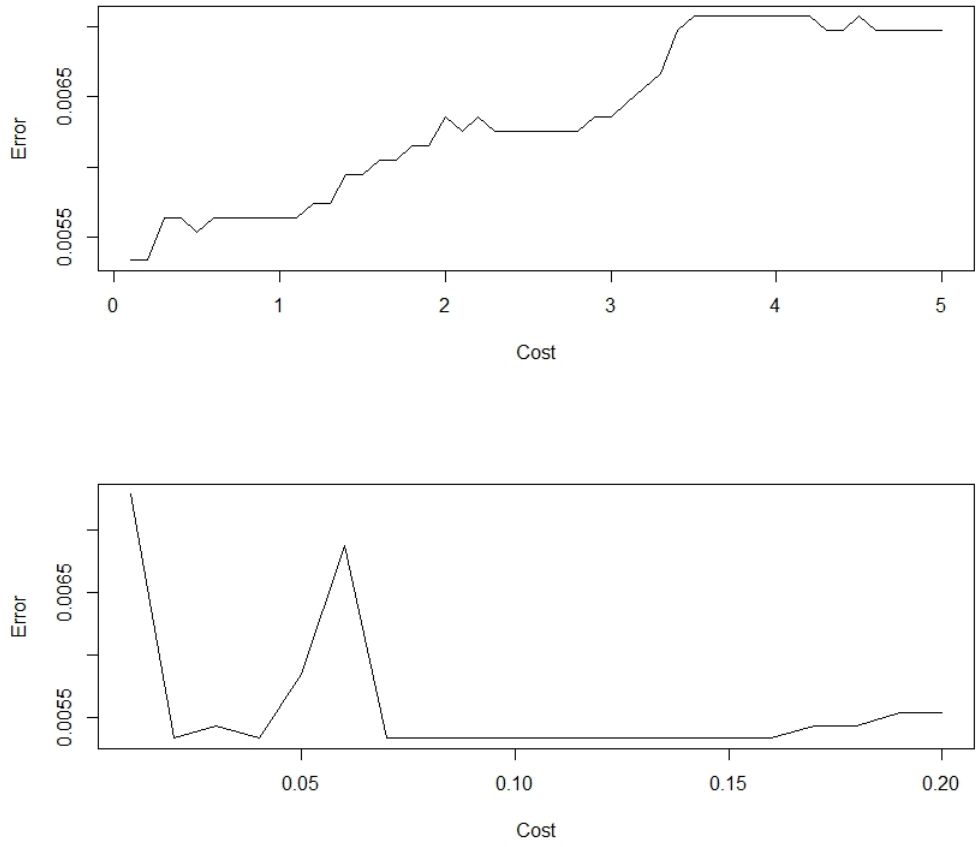

Figura 3.2. Errore previsionale (linear SVM) in funzione del parametro C sul dataset originale

3.2.2 Radial SVM

In precedenza è stato illustrato il metodo in cui il separatore tra le due classi di "Label-Clienti" era lineare. In questa analisi può essere utile un separatore non lineare. Quindi si considera la generalizzazione del support vector classifier che si chiama support vector machine. Si può vedere dal problema (3.2) che il support vector classifier lineare può essere rappresentato come:

$$f(\boldsymbol{x}) = \beta_0 + \sum_{i=1}^{n} \beta_i < \boldsymbol{x}, \boldsymbol{X}_{\{i,.\}} > \tag{3.3}$$

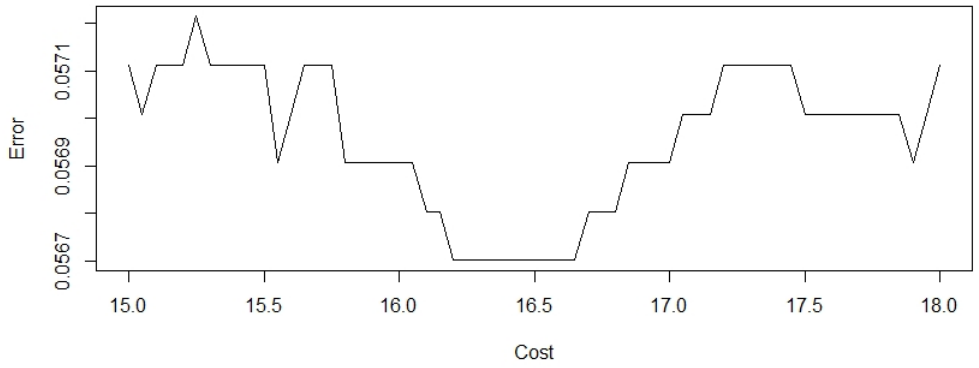

Figura 3.3. Errore previsionale (linear SVM) in funzione del parametro C sul dataset con le componenti principali

Figura 3.4. Linear SVM sul dataset con le componenti principali con assi di riferimento PC1-PC2. Con "x" vengono indicate le osservazioni che sono support vectors mentre con "o" le restanti osservazioni. I punti in rosso hanno l'etichetta per LabelClienti "1" e i punti in nero hanno l'etichetta per LabelClienti "0"

Per stimare i parametri $\beta_0, ..., \beta_n$ devono essere calcolati i $\binom{n}{2}$ prodotti interni $< \boldsymbol{X}_{\{i,.\}}, \boldsymbol{X}_{\{i',.\}} >$ tra le osservazioni del dataset. Sostituendo il prodotto interno con una sua generalizzazione si ottiene il kernel:

$$K(\boldsymbol{X}_{\{i,.\}}, \boldsymbol{X}_{\{i',.\}})$$

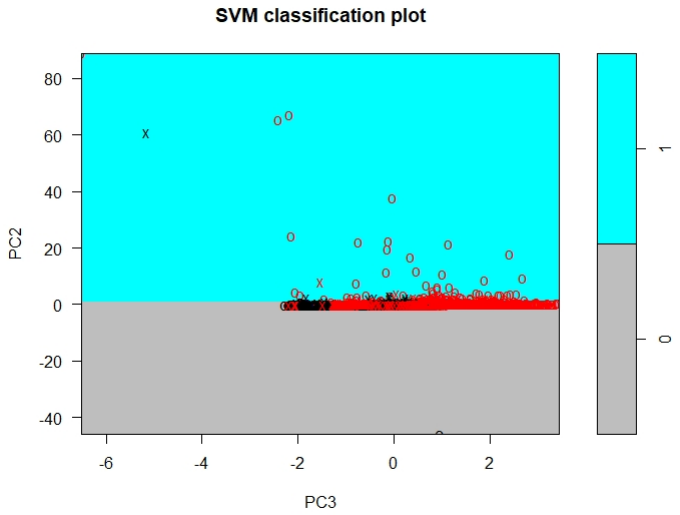

Figura 3.5. Linear SVM sul dataset con le componenti principali con assi di riferimento PC2-PC3. Con "x" vengono indicate le osservazioni che sono support vectors mentre con "o" le restanti osservazioni. I punti in rosso hanno l'etichetta per LabelClienti "1" e i punti in nero hanno l'etichetta per LabelClienti "0"

Per il prodotto interno si parla di kernel lineare ma una scelta tipica è il kernel radiale:

$$K(\boldsymbol{X}_{\{i,.\}}, \boldsymbol{X}_{\{i',.\}}) = exp(-\gamma \sum_{j=1}^{p} (X_{ij} - X_{i'j})^2) \tag{3.4}$$

In questo caso bisogna trovare il miglior valore di γ ed il miglior valore di C. Quindi il primo passo è trovare questi due valori e successivamente costruire l'algoritmo SVM radiale con questi due parametri.
Per trovare i valori ottimi di γ e C si utilizza come in precedenza la k-fold cross validation dove viene calcolato l'errore medio di previsione che si ottiene utilizzando diverse combinazioni dei parametri γ e C.

3.2.3 Previsione

Dopo aver costruito i diversi modelli, vengono testati i classificatori utilizzando il testing set. In particolare per ogni record del testing set si prevede la classe di appartenenza e successivamente vengono confrontati i risultati ottenuti con le vere etichette.
Nei boxplot della Figura 3.6 si ha la probabilità che le aziende abbiano "LabelClienti" uguale a "1" e sono separati nelle loro classi reali. In tutti i boxplot possiamo vedere che ci sono pochi punti la cui vera etichetta è "0" ma hanno la probabilità di essere "1" molto vicina a uno. Tuttavia, vediamo che la probabilità media rimane anche nei classificatori sul dataset con le componenti principali vicina ad 1 per le aziende che hanno realmente "LabelClienti" pari ad "1" e vicina a 0 per le aziende che hanno realmente "LabelClienti"

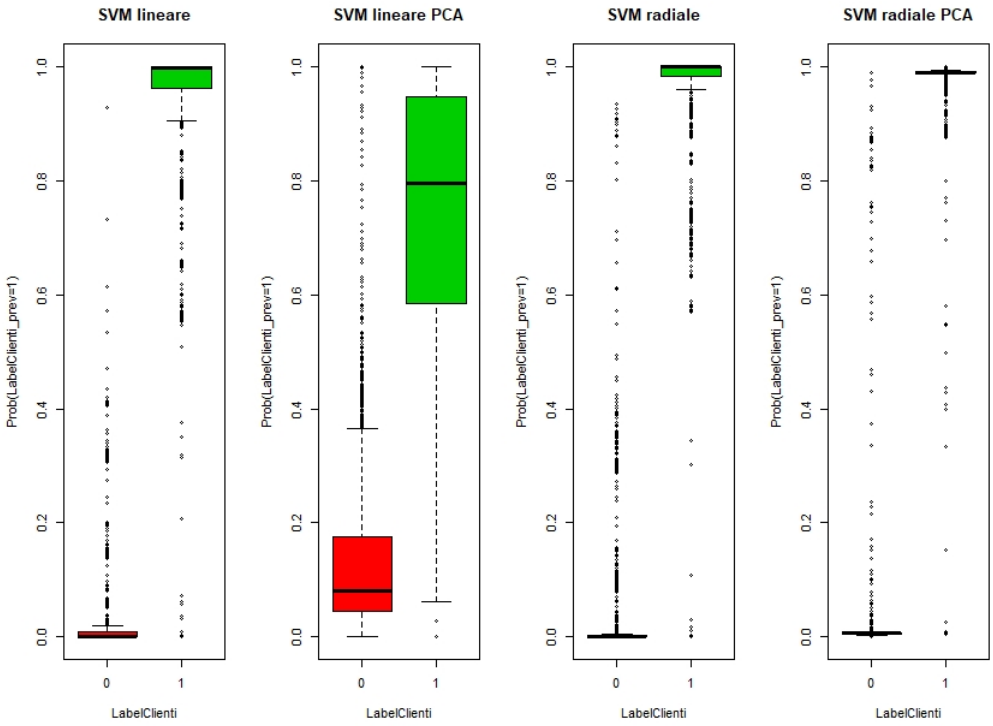

Figura 3.6. Boxplot previsione con Support Vector Machines. Probabilità che le aziende abbiano "LabelClienti" uguale a "1" separate nelle loro classi reali.

	LabelClienti = 0	LabelClienti = 1
Predict 0	2567	23
Predict 1	5	1532

Tabella 3.1. Tabella classificazione Linear SVM dataset originale

pari ad "0", soprattutto utilizzando il kernel radiale.

Nelle Tabelle 3.1, 3.2, 3.3, 3.4, viene riportato il numero di record ben classificati e il numero di record mal classificati con i diversi classificatori. Dai risultati ottenuti con il dataset originale si ha che il classificatore con kernel lineare sembra funzionare meglio mentre nel dataset con le componenti principali sembra funzionare meglio il classificatore con kernel radiale. In ogni modo è davvero molto difficile scegliere il modello che si comporta meglio con le Tabelle 3.1, 3.2, 3.3, 3.4, per questo motivo viene utilizzata la curva ROC per prendere questa decisione. La curva ROC, Figura 3.7, mostra contemporaneamente il "False positive rate" o "sensitivity", e il "True positive rate" o "1-specificity", che si ottiene per ogni soglia. Per soglia si intende quel valore tale per cui l'osservazione che ha la probabilità di avere "LabelClienti" uguale a "1" maggiore della soglia, viene assegnata

	LabelClienti = 0	LabelClienti = 1
Predict 0	2508	250
Predict 1	64	1305

Tabella 3.2. Tabella classificazione Linear SVM dataset dataset con le componenti principali

	LabelClienti = 0	LabelClienti = 1
Predict 0	2550	19
Predict 1	22	1536

Tabella 3.3. Tabella classificazione Radial SVM dataset originale

alla classe "1" altresi verrà assegnata alla classe "0". Il True positive rate (TPR) è la quantità descritta dalla seguente equazione:

$$TPR = \frac{TP}{TP + FN} \tag{3.5}$$

dove TP è il numero di osservazioni appartenenti alla classe "1" correttamente classificate e FN è il numero di osservazioni appartenenti alla classe "0" mal classificate. Il False positive rate (FPR) è la quantità descritta dalla seguente equazione:

$$FPR = \frac{FP}{FP + TN} \tag{3.6}$$

dove FP è il numero di osservazioni appartenenti alla classe "1" mal classificate e TN è il numero di osservazioni appartenenti alla classe "0" correttamente classificate. Ovviamente idealmente si vuole avere che il primo indice sia 0 e il secondo indice sia 1. Viene utilizzato l'indice AUC per riassumere le prestazioni complessive e scegliere il modello migliore. L'indice AUC è pari al valore dell'area al di sotto della curva ROC e il modello con AUC più alto corrisponde al modello migliore. Dal grafico della curva ROC, Figura 3.7, si può vedere che tutti i classificatori sono buoni perché tutti sono lontani dalla linea che divide in due il quadrato. Questa linea corrisponde al classificatore che sceglie casualmente con probabilità 0.5 la classe "1" o la classe "0". Dall'indice AUC, che si può osservare nella Tabella 3.5, possiamo dire che il miglior classificatore sul dataset originale è il classificatore con kernel lineare mentre il miglior classificatore sul dataset con le componenti principali è il classificatore con kernel radiale.

3.3 Regressione logistica

Un altro metodo di classificazione è il modello logistico. Questo metodo viene applicato ai problemi di classificazione e mira a prevedere la probabilità che un osservazione appartenga

	LabelClienti = 0	LabelClienti = 1
Predict 0	2525	26
Predict 1	47	1529

Tabella 3.4. Tabella classificazione Radial SVM dataset con le componenti principali

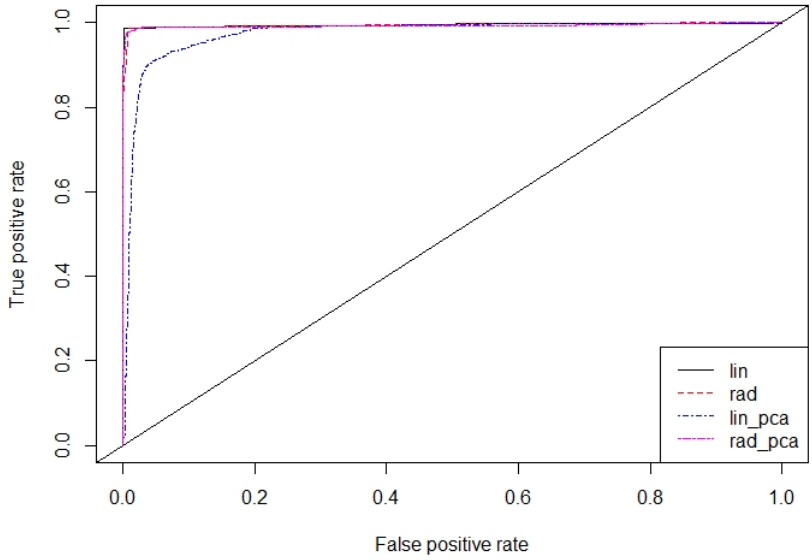

Figura 3.7. Curva ROC dei modelli SVM

a una particolare classe, dati i valori dei predittori. In questo problema di classificazione si ha una variabile di risposta binaria, ovvero "LabelClienti", che può assumere "0" e "1" come valori. Si vuole prevedere la probabilità che l'azienda sia un possibile cliente, cioè che la variabile dipendente "LabelClienti" sia uguale a 1. Si definisce $p(\boldsymbol{X}_{\{i,.\}}) = Pr(\boldsymbol{Y}_i = 1|\boldsymbol{X}_{\{i,.\}})$ dove $\boldsymbol{Y}_i$ è la variabile dipendente "LabelClienti" con una distribuzione Bernoulliana $Y_i \sim Be(p(\boldsymbol{X}_{\{i,.\}}))$ e $\boldsymbol{X}_{\{i,.\}}$ è il vettore delle covariate $(X_{\{i,1\}}, ..., X_{\{i,p\}})$. Questo modello nasce dall'idea di adattare la regressione lineare alla risposta binaria "LabelClienti" e prevedere "1" se la stima della probabilità che "LabelClienti" sia uguale a "1" è maggiore di 0.5. Infatti per mantenere questa probabilità nell'intervallo [0,1] non possiamo eseguire la regressione lineare, ma abbiamo bisogno di una regressione logistica. Viene quindi mappata la combinazione lineare delle covariate nell'intervallo [0,1] grazie alla funzione logistica $f(x) = \frac{e^x}{1+e^x}$ ottenendo la relazione seguente relazione

$$p(\boldsymbol{X}_{\{i,.\}}) = \frac{e^{\beta_0+\beta_1 X_{\{i,1\}}+...+\beta_p X_{\{i,p\}}}}{1 + e^{\beta_0+\beta_1 X_{\{i,1\}}+...+\beta_p X_{\{i,p\}}}} \tag{3.7}$$

Tipo Classificatore	AUC
LinearSVM	0.9947
PCALinearSVM	0.9747
RadialSVM	0.9946
PCARadialSVM	0.9927

Tabella 3.5. Indice AUC classificatori SVM

Dopo un pò di manipolazione dell'equazione (3.7), viene individuata la quantità che si chiama odds, ovvero il rapporto tra la probabilità che un evento accade e la probabilità che tale evento non accade:

$$\frac{p(\boldsymbol{X}_{\{i,.\}})}{1-p(\boldsymbol{X}_{\{i,.\}})} = e^{\beta_0+\beta_1 X_{\{i,1\}}+...+\beta_p X_{\{i,p\}}} \tag{3.8}$$

Prendendo il logaritmo di entrambi i membri dell'equazione (3.8) si ottiene la quantità che si chiama log-odds o logit:

$$log\left(\frac{p(\boldsymbol{X}_{\{i,.\}})}{1-p(\boldsymbol{X}_{\{i,.\}})}\right) = \beta_0 + \beta_1 X_{\{i,1\}} + ... + \beta_p X_{\{i,p\}} \tag{3.9}$$

Quindi quello che si vuole fare è stimare il vettore dei coefficienti β per poi stimare la probabilità che "LabelClienti" sia uguale a "1". Assumendo che per ogni osservazione le Y_i sono indipendenti e identicamente distribuite si ottiene la relazione seguente

$$\begin{aligned} P(Y_1 = y_1, ..., Y_n = y_n \quad | \ \boldsymbol{X}_{\{1,.\}}, ..., \boldsymbol{X}_{\{N,.\}}) &= \prod_{i=1}^{N} P(Y_i = y_i \quad | \ \boldsymbol{X}_{\{i,.\}}) = \\ &= \prod_{i:y_i=1} p(\boldsymbol{X}_{\{i,.\}}) \prod_{i':y_{i'}=0} (1-p(\boldsymbol{X}_{\{i',.\}})) \end{aligned} \tag{3.10}$$

dove $y_i \in \{0,1\}$ e N è il numero delle osservazioni. A questo punto per stimare il vettore dei β viene massimizzata la seguente funzione di verosomiglianza:

$$\ell(\beta_0, \beta_1, ..., \beta_p) = \prod_{i:y_i=1} p(\boldsymbol{X}_{\{i,.\}}) \prod_{i':y_{i'}=0} (1-p(\boldsymbol{X}_{\{i',.\}})) \tag{3.11}$$

Infatti tramite la massimizzazione rispetto a β si trova la stima dei β e la relativa stima della probabilità che "LabelClienti" sia uguale a uno.
Un primo step per il modello di regressione logistica è quello di individuare quali sono i predittori ideali tra quelli a disposizione per costruire il modello. Prima di tutto viene selezionato il modello migliore nel training set per ogni numero di predittori, dal modello con tutte le variabili come predittori fino al modello con 11 variabili. In particolare viene

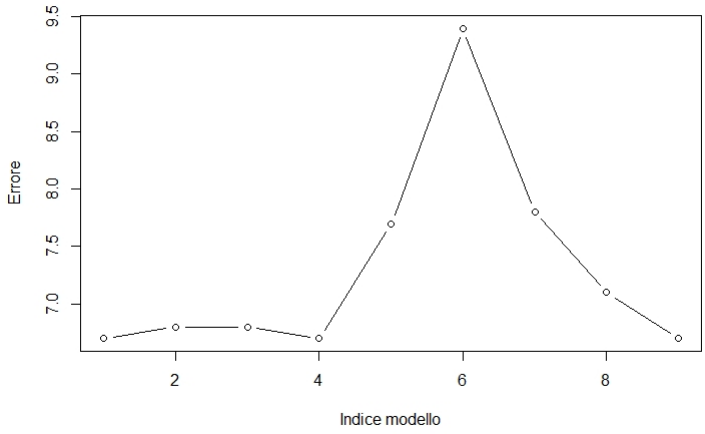

Figura 3.8. Errore medio di mal classificazione per i modelli di regressione logistica

utilizzata una step-function implementata con l'indice AIC e con direzione "backward". AIC è il criterio di informazione di Akaike, ed è un metodo per confrontare i modelli statistici. In particolare valuta la quantità di informazioni perse quando un determinato modello viene utilizzato per descrivere la realtà ed è pari alla seguente quantità:

$$AIC = 2ln(L) + 2k \tag{3.12}$$

dove k è il numero di parametri del modello e L è il valore di massima verosomiglianza del modello. La step-function quindi ad ogni passo rimuove una variabile dal modello precedente e la variabile che viene rimossa è quella con cui si ha il valore di AIC più alto. Successivamente per scegliere il modello migliore tra quelli ottenuti per ogni numero di predittori, grazie alla step-function implementata con l'indice AIC, viene utilizzata la k-fold cross validation sul training set. Viene diviso il training set in k sottogruppi e calcolato l'errore assoluto di previsione, ovvero quante osservazioni sono state mal classificate, k volte utilizzando ad ogni giro un sottogruppo differente come testing set. Come si può vedere nella Figura 3.8 si ottengono tre possibili modelli ottimali, che corrispondono al primo, al quarto e al nono modellodello e sono denominati rispettivamente m1, m4 e m9. Il modello m1 ha come predittori tutte le variabili, Equazione (3.13), il modello m4 ha un numero di predittori pari a 16, Equazione (3.14), e il modello m9 ha un numero di predittori pari a 11, Equazione (3.15). Di seguito vengono elencati i modelli di regressione logistica m1, m4 e m9 dove per identificare gli indici j dei predittori $X_{\{i,j\}}$ si fa riferimento

	LabelClienti = 0	LabelClienti = 1
Predict 0	2549	49
Predict 1	23	1506

Tabella 3.6. Tabella classificazione modello di regressione logistica: m4

Tipo Classificatore	AUC
m1	0.9865
m4	0.9868
m9	0.9865

Tabella 3.7. Indice AUC modelli di regressione logistica: m1, m4 e m9

alla Tabella 1.1.

$$log\left(\frac{Pr(LabelClienti_i = 1)}{1 - Pr(LabelClienti_i = 1)}\right) = \beta_0 + \beta_2 X_{\{i,2\}} + \beta_3 X_{\{i,3\}} + \beta_4 X_{\{i,4\}} + \beta_5 X_{\{i,5\}} + \beta_6 X_{\{i,6\}} + \beta_7 X_{\{i,7\}} + \beta_8 X_{\{i,8\}} + \beta_9 X_{\{i,9\}} + \beta_{10} X_{\{i,10\}} + \beta_{11} X_{\{i,11\}} + \beta_{12} X_{\{i,12\}} + \beta_{13} X_{\{i,13\}} + \beta_{14} X_{\{i,14\}} + \beta_{15} X_{\{i,15\}} + \beta_{16} X_{\{i,16\}} + \beta_{17} X_{\{i,17\}} + \beta_{18} X_{\{i,18\}} + \beta_{19} X_{\{i,19\}} + \beta_{20} X_{\{i,20\}} \tag{3.13}$$

$$log\left(\frac{Pr(LabelClienti_i = 1)}{1 - Pr(LabelClienti_i = 1)}\right) = \beta_0 + \beta_2 X_{\{i,2\}} + \beta_3 X_{\{i,3\}} + \beta_4 X_{\{i,4\}} + \beta_5 X_{\{i,5\}} + \beta_6 X_{\{i,6\}} + \beta_7 X_{\{i,7\}} + \beta_9 X_{\{i,9\}} + \beta_{10} X_{\{i,10\}} + \beta_{11} X_{\{i,11\}} + \beta_{12} X_{\{i,12\}} + \beta_{13} X_{\{i,13\}} + \beta_{15} X_{\{i,15\}} + \beta_{16} X_{\{i,16\}} + \beta_{17} X_{\{i,17\}} + \beta_{19} X_{\{i,19\}} + \beta_{20} X_{\{i,20\}} \tag{3.14}$$

$$log\left(\frac{Pr(LabelClienti_i = 1)}{1 - PR(LabelClienti_i = 1)}\right) = \beta_0 + \beta_2 X_{\{i,2\}} + \beta_3 X_{\{i,3\}} + \beta_5 X_{\{i,5\}} + \beta_6 X_{\{i,6\}} + \beta_7 X_{\{i,7\}} + \beta_9 X_{\{i,9\}} + \beta_{10} X_{\{i,10\}} + \beta_{13} X_{\{i,13\}} + \beta_{16} X_{\{i,16\}} + \beta_{19} X_{\{i,19\}} + \beta_{20} X_{\{i,20\}} \tag{3.15}$$

Si deve ora scegliere il modello ottimale tra: m1, Equazione (3.13), m4, Equazione (3.14), e m9, Equazione (3.15). Si costruisce quindi la curva ROC, Figura 3.9, e come visto per il SVM per decidere quale è il classificatore migliore viene selezionato il modello con l'indice AUC maggiore. Come si può vedere nella Tabella 3.7 si ha che il classificatore per la regressione logistica che performa meglio è m4. Viene inoltre riportata la Tabella di classificazione 3.6 per il modello m4 che sarà poi utile per confrontare i risultati ottenuti dai diversi modelli di classificazione nella Sezione 3.6.

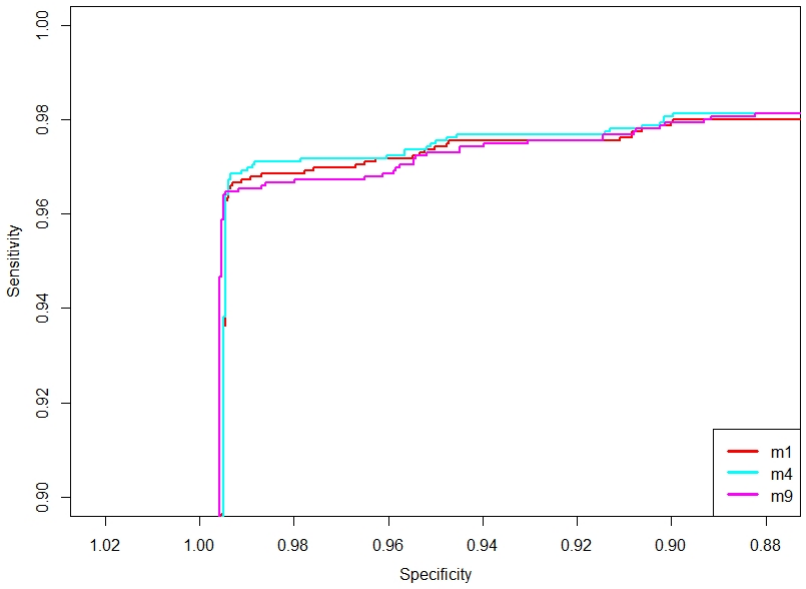

Figura 3.9. Curva ROC dei modelli di regressione logistica: m1, m4 e m9

3.4 Decision Tree

L'algoritmo Decision Tree opera tramite la stratificazione o la segmentazione dello spazio predittivo in più regioni. Il nome dell'algoritmo deriva dal fatto che le regole di divisione utilizzate per segmentare lo spazio predittivo può essere riassunto in un albero. Questo metodo può essere molto utile da un punto di vista interpretativo tuttavia, solitamente non sono competitivi con gli altri metodo di classificazione. In questo metodo si divide lo spazio dei predittori in J regioni distinte e non sovrapposte $(R_1, ..., R_J)$. Ogni osservazione che cadrà nella regione R_j avrà la stessa previsione che è semplicemente la classe maggioritaria per l'osservazioni del training set nella regione R_j.
Vediamo ora come costruire le regioni $R_1, ..., R_J$. Per tutti i predittori $\boldsymbol{X}_{\{.,j\}}$ viene selezionato il punto di taglio ottimale s in modo tale che la divisione dello spazio nelle regioni $\{\boldsymbol{X} \mid \boldsymbol{X}_{\{.,j\}} < s\}$ e $\{X \mid \boldsymbol{X}_{\{.,j\}} \geq s\}$ porta alla massima riduzione dell'errore di classificazione. Con $\{\boldsymbol{X} \mid \boldsymbol{X}_{\{.,j\}} < s\}$ si intende la regione all'interno dello spazio predittivo dove $\boldsymbol{X}_{\{.,j\}}$ assume valori minori di s. Quindi viene scelto il predittore e il punto di taglio in modo tale che viene minimizzata l'equazione

$$\sum_{i:\boldsymbol{X}_{\{i,.\}i} \in R_1} (y_i - \hat{y}_{R_1})^2 + \sum_{i:\boldsymbol{X}_{\{i,.\}i} \in R_2} (y_i - \hat{y}_{R_2})^2 \tag{3.16}$$

dove y_i è la classe reale dell'i-esima osservazione e $\hat{y}_{R_m}$ è la risposta associata alla regione R_m. Successivamente il processo viene ripetuto, cercando il miglior predittore e il migliore cutpoint per dividere ulteriormente i dati in modo da ridurre il numero di osservazioni mal classificate. Tuttavia questa volta viene divisa una delle due regioni precedentemente

identificate, trovando così tre regioni. Ancora una volta, si cerca di dividere una di queste tre regioni ulteriormente, in modo da ridurre al minimo il numero di osservazioni mal classificate. Le regioni successive verrano quindi costruite seguendo lo stesso procedimento. Si può vedere facilmente che l'approccio per costruire il Decision Tree è un approccio "greedy", ovvero la divisione migliore viene effettuata in quel particolare passaggio, piuttosto che guardare avanti e scegliere una divisione che porterà a un albero migliore nei passaggi successivi.
È stato ottenuto un albero decisionale composto da un unico split ottenuto con l'utilizzo della variabile "Anno", dove gli split sono i punti di taglio della regione dei predittori. Il risultato così ottenuto non fornisce un classificatore utile per le analisi, infatti basare la classificazione di un'azienda come possibile cliente o meno con l'utilizzo di una sola informazione non può essere affidabile. Inoltre gli alberi decisionali possono essere molto poco robusti. In altre parole, una piccola modifica dei dati può causare un grande cambiamento nella stima finale. Il modello ottenuto con il Decision Tree non è stato quindi preso in considerazione. Questo algoritmo però è il punto di partenza per il classificatore Random Forest con il quale si ottengono risultati interessanti come vedremo nelle sezioni successive.

3.5 Random Forest

Il Random Forest è un algoritmo decisionale che migliora un'idea che era stata applicata all'algoritmo decisionale Decision Tree, ovvero il Bagging. Il Bagging consiste nel costruire B diversi alberi decisionali basati su B diversi training set di dati, dove i training set vengono costruiti partendo dal training set originale tramite la tecnica di Bootstrap. La tecnica di Bootsrap consente di generare B training set campionando casualmente dal training set originale, e i training set così creati possono essere anche sovrapposti. Nel Bagging il classificatore finale si ottiene come "voto di maggioranza" dei B diversi classificatori, in pratica la previsione finale di una osservazione è data dalla classe più comune tra le B diverse previsioni. Il Bagging migliora la precisione ma gli alberi generati sono fortemente correlati. L'algoritmo Random forest può risolvere il problema della correlazione. Il Random forest opera come il Bagging, con la differenza che per ogni albero decisionale sceglie il migliore split in base a m predittori scelti casualmente e non sulla base di tutti i predittori come con il Bagging. La scelta tipica di m è tipicamente $\sqrt{p}$ dove p è il numero di predittori nel dataset, oppure come è stato fatto nell'analisi, viene scelto m in modo tale da avere la migliore "Accuracy". "Accuracy" è una metrica che riassume le prestazioni di un modello di classificazione, in particolare viene calcolata come il rapporto tra il numero di osservazioni correttamente classificate e il numero totale di osservazioni. Nel caso di una classificazione binaria, come in questo caso, può essere espressa nel modo seguente:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{3.17}$$

dove TP è il numero di osservazioni appartenenti alla classe "1" correttamente classificate, TN è il numero di osservazioni appartenenti alla classe "0" correttamente classificate, FP è il numero di osservazioni appartenenti alla classe "1" mal classificate, FN è il numero di

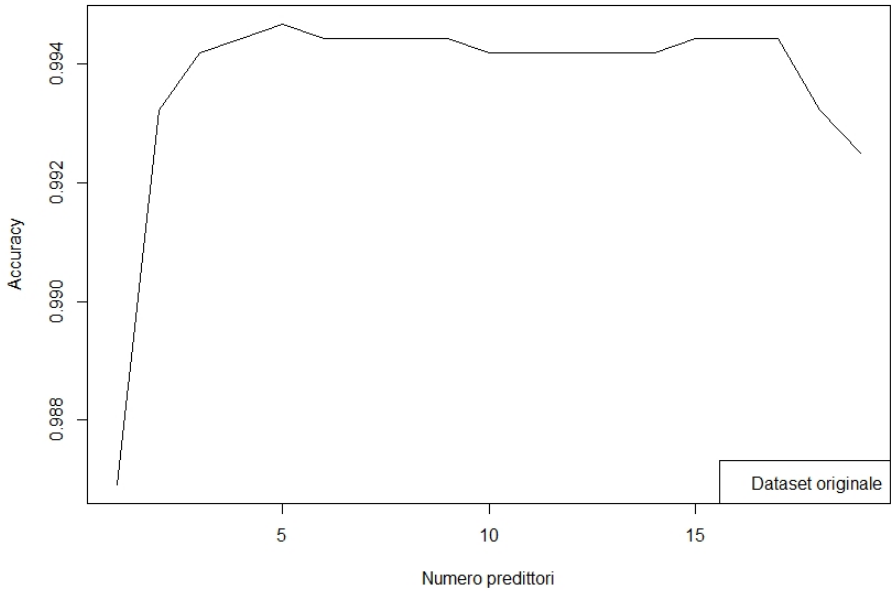

Figura 3.10. Accuracy del modello Random Forest in base al numero di predittori utilizzati nei Decision Tree

	LabelClienti = 0	LabelClienti = 1
Predict 0	2569	19
Predict 1	3	1536

Tabella 3.8. Tabella classificazione Random Forest

osservazioni appartenenti alla classe "0" mal classificate.
Quindi la prima cosa che è stata fatta riguardava l'individuazione del numero m di predittori da utilizzare nell'algoritmo Random Forest sulla base del parametro "Accuracy". Come si può vedere nella Figura 3.10 è stato ottenuto m = 5. Nell'analisi è stato quindi costruito il modello Random forest utilizzando il training set e il parametro m = 5. A questo punto il classificatore è stato testato sul testing set ottenenendo la Tabella di classificazione 3.8 e la curva ROC nella Figura 3.11 con il relativo valore di AUC pari a 0.9944.

3.6 Conclusione modelli previsionali

In questo capitolo è stata illustrata la costruzione dei diversi modelli previsionali con l'obiettivo di individuare il miglior modello che permettesse di classificare un'azienda come

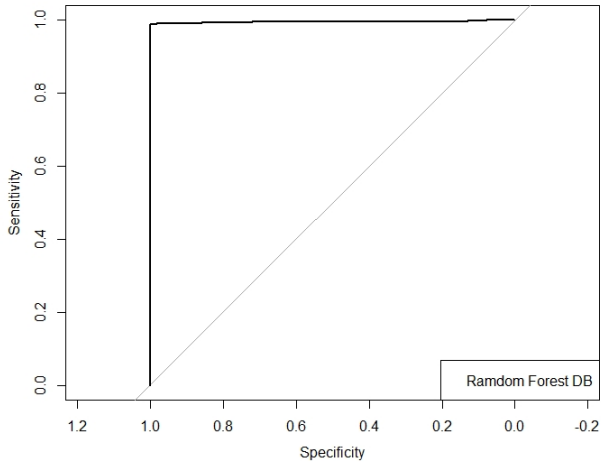

Figura 3.11. Curva ROC del modello Random Forest

possibile cliente o meno.
Vengono ora analizzati i risultati ottenuti con i diversi modelli previsionali. Per prima cosa vengono confrontate le curve ROC ottenute con i diversi modelli di classificazione, Figura 3.12. Andando a vedere più nel dettaglio la curva ROC, Figura 3.13, i modelli che sembrano performare meglio con questo confronto sono il Random forest e il support vector machines con kernel lineare. Vengono ora analizzati più nel dettaglio gli output ottenuti dai modelli di classificazione costruiti. I primi modelli che sono stati descritti nel capitolo sono: il support vector machines con kernel lineare e il support vector machines con il kernel radiale. Confrontando questi due modelli si nota che il support vector machines con kernel lineare risulta performare meglio in termini dell'indice AUC anche se i risultati sono molto simili tra loro. L'obiettivo dell'analisi però è quello di individuare le aziende che possono essere clienti e quindi si preferisce un classificatore che performi al meglio nel classificare bene le aziende che hanno 'LabelClienti' uguale ad "1". Infatti per l'obiettivo dell'analisi si preferisce classificare come potenziali clienti aziende che non lo sono piuttosto che perdere potenziali clienti. Tra i due modelli del support vector machines, come si può vedere dalle Tabelle 3.1 e 3.3, si preferisce il Support Vector Machines con kernel radiale anche se questo ha un valore dell'indice AUC inferiore. In generale per questi due modelli soprattutto per quello con kernel radiale viene riscontrato un alto tempo di calcolo dovuto soprattutto per la "stima" dei parametri del modello.
Con la regressione logistica si ottengono invece i risultati peggiori come si può vedere dalla curva ROC nella Figura 3.13 e dalla Tabella previsionale 3.6
Infine si analizzano i risultati ottenuti con l'algoritmo Random forest. Questo modello è quello che performa meglio in termini di AUC e come si può vedere dalla sua Tabella previsionale 3.8 è il modello che sbaglia di meno la previsione per le aziende con "LabelClienti" pari a "0". Insieme al classificatore support vector machines con kernel radiale è anche l'algoritmo che sbaglia di meno la classificazione di aziende con "LabelClienti" pari a "1".

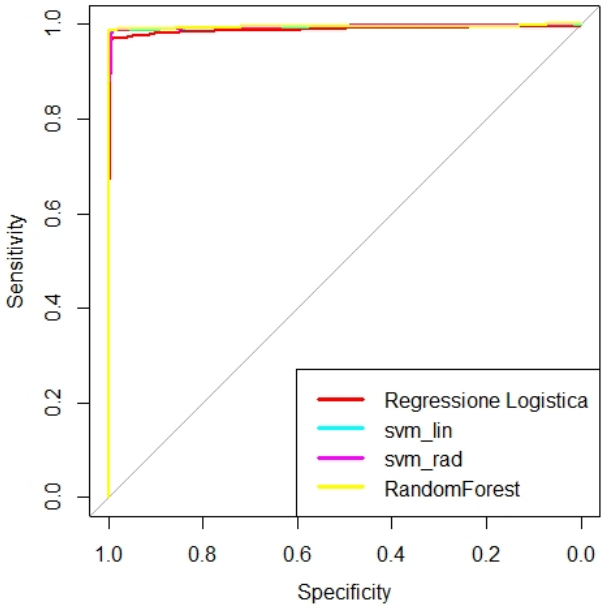

Figura 3.12. Curva ROC dei modelli previsionali analizzati

In linea di massima il Random forest necessita di una stima dei parametri di tuning anche se meno onerosa di quella effettuata per gli algoritmi support vector machines. Quindi dalle varie considerazioni fatte si ha che il Random forest è il classificatore che performa meglio per questa analisi.

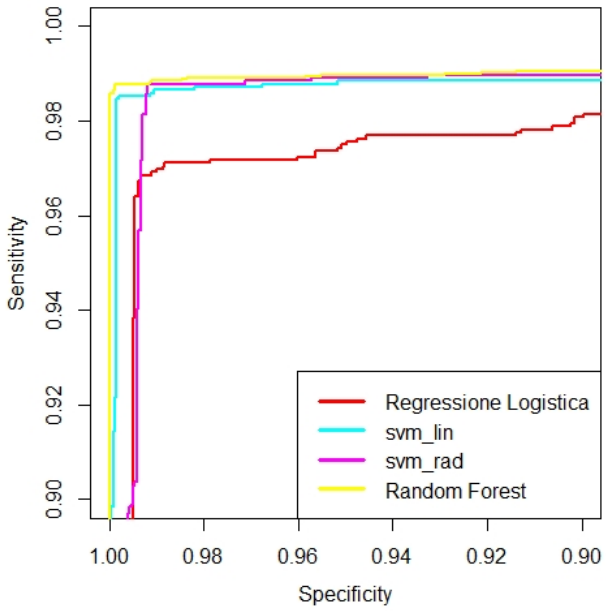

Figura 3.13. Curva ROC dei modelli previsionali analizzati

Bibliografia

Alejandra Arizmendi Eréndira Rendón, Itzel Abundez and Elvia M. Quiroz. Internal versus external cluster validation indexes. *International journal of computers and comunications*, 5:27–34, 2011.

P. Filzmoser. A multivariate outlier detection method. 2004.

Trevor Hastie Robert Tibshirani Gareth James, Daniela Witten. *An Introduction to Statistical Learning*. Springer, 2013.

Peter J.Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1964.

Cheng Soon Ong Marc Peter Deisenroth, A. Aldo Faisal. *Mathematics for machine learning*. Cambridge University Press, 2020.

Marina Soley-Bori. Dealing with missing data: Key assumptions and methods for applied analysis. 2013.

Kumar Tan, Steinbach. *Introduction to Data Mining*. McGraw Hill, 2006.

Bureau van Dijk. *AIDA*. Bureau van Dijk Electronic Publishing Ltd., 2022.