CREDIT SCORE: DETERMINAZIONE DELLA CLASSE DI CREDITO TRAMITE ALGORITMI DI MACHINE LEARNING

PERRELLI ADALGISA[1,2], NICOLE SANTERO[1,3] E MAGDA MORETTI[1,4]

[1]Pre-Processing e Reti Neurali

[2]Clustering Agglomerativo Gerarchico e Random Forest

[3]Algoritmo Iterative Distance-Based e Support Vector Machine

[4]DBscan e Algoritmo K-Nearest Neighbours

ABSTRACT

Il seguente elaborato ha avuto come obiettivo la determinazione degli algoritmi di clustering e classificazione più performanti per la discriminazione tra classi di credito. I risultati inerenti ai metodi di clustering non hanno condotto a differenze significative in termini di purezza. Il metodo di classificazione che ha prodotto il valore più elevato di Accuracy è stato l'algoritmo Radial Support Vector Machine. In generale, con tutti i metodi è stata riscontrata un'elevata similarità delle istanze di classe 'Standard' con quelle appartenenti alle altre due.

1 INTRODUZIONE

Il dataset in analisi, disponibile al seguente link https://www.kaggle.com/datasets/parisrohan/credit-score-classification, originalmente comprende 28 variabili e 100.000 istanze. Si tratta di un panel data in cui ogni osservazione illustra il profilo di un cliente, attraverso le informazioni bancarie, per i primi otto mesi dell'anno di raccolta dati presso un istituto di credito.

1.1 OBIETTIVI

Il principale scopo della seguente analisi è determinare il profilo creditizio di ciascun cliente, sulla base delle informazioni personali (età, occupazione…), del conto corrente registrato e delle spese sostenute mensilmente. L'obiettivo è stato perseguito attraverso l'uso di diversi strumenti raggruppabili in due tecniche. La tecnica 'unsupervised', che prevede l'utilizzo in un contesto non supervisionato di algoritmi finalizzati alla formazione di cluster preferibilmente omogenei internamente ed

eterogenei tra loro e che corrispondono ai tre profili creditizi, le modalità della variabile target, di cui l'apporto di informazione è presunto assente per questa metodologia. La tecnica 'supervised', per la quale si utilizza la variabile dipendente per validare e studiare i risultati, si basa sul concetto di classificazione: lo scopo è quindi prevedere e non raggruppare. Ciò permetterebbe all'istituto di individuare con maggior certezza, avendo a disposizione algoritmi allenati e affinati per tale obiettivo, i profili creditizi di nuovi clienti.

2 PRE-PROCESSING

Preliminarmente, si è ritenuta opportuna l'esclusione delle seguenti variabili, fornendo un'informazione ridondante rispetto la variabile CUSTOMER_ID: ID (riferimento univoco della rilevazione), SSN (codice di previdenza sociale del cliente), NAME (nome e cognome del cliente). Inoltre, sono state eliminate le covariate TYPE_LOAN (tipologie di prestiti attivi del cliente) e CREDIT_AGE (età creditizia del cliente) ritenendole problematiche per l'analisi.

2.1 INDIVIDUAZIONE DEI VALORI ANOMALI E IMPUTAZIONE DEI DATI MANCANTI

Durante la prima fase di pulizia del dataset sono state corrette le anomalie individuate tra le osservazioni, rispettando la natura di ciascuna variabile. Si è proceduto con il seguente ragionamento: essendo presenti variabili dai valori costanti in tutti i mesi di rilevazione, facenti riferimento al medesimo cliente, si è ritenuta opportuna, in caso di valori non disponibili oppure errati, l'imputazione tramite il valore della moda*, calcolabile dai valori dei sottogruppi ottenuti discriminando secondo la variabile CUSTOMER_ID.

Si è proceduto con un controllo e l'eventuale correzione delle tipologie delle variabili. Ulteriori dettagli sulle operazioni di pre-processing compiute per ciascuna covariata sono specificati nell'elenco sottostante, oltre alla descrizione della variabile e la specificazione della tipologia della stessa:

- CUSTOMER_ID: identificativo univoco del cliente – alfanumerico;
- MONTH: mese della rilevazione - fattoriale;
- AGE: età del cliente - numerico – imputazione con il valore modale*;
- OCCUPATION: attuale occupazione lavorativa del cliente – fattoriale – imputazione con il valore modale*;
- ANNUAL_INCOME: entrate annuali totali del cliente – numerico – imputazione con il valore modale*;
- MONTHLY_SALARY: salario base mensile del cliente – numerico – imputazione tramite il valore modale*;
- NUM_BANK_ACCOUNTS: numero di conti correnti posseduti dal cliente – numerico – intervallo $[0, \infty)$ e imputazione tramite il valore modale*;
- NUM_CREDIT_CARD: numero di carte di credito possedute dal cliente – numerico – imputazione tramite il valore modale*;
- INTEREST_RATE: tasso d'interesse stipulato da contratto – numerico – imputazione tramite il valore modale*;
- NUM_LOAN: numero di prestiti attivi del cliente – numerico – conteggio da TYPE_LOAN;
- DELAY_DAYS: numero medio di giorni tardivi di pagamento del debito – fattoriale – accorpamento dei valori in cinque categorie;

- NUM_DELAY_PAYMENT: numero medio di pagamenti tardivi – numerico – intervallo [0, 30] con conseguenti NA;
- CHANGE_CREDIT_LIMIT: percentuale di variazione dei massimali delle carte – numerico;
- NUM_CREDIT_INQUIRIES: numero di richieste di informazioni sullo stato bancario – numerico – intervallo [0, 100] con conseguenti NA e imputazione tramite valore modale*:
- CREDIT_MIX: classificazione della composizione del portfolio creditizio del cliente - fattoriale;
- OUTSTANDING_DEBT: ammontare del debito rimanente da pagare - numerico;
- UTILIZATION_RATIO: rapporto di utilizzo del credito - numerico;
- MINIMUM_AMOUNT: pagamento dell'importo minimo richiesto - binario;
- MONTHLY_EMI: somma dei ratei mensili da pagare – numerico – imputazione tramite valore modale*;
- MONTHLY_INVESTMENT: somma d'investimento mensile - numerico;
- PAYMENT_BEHAVIOR: profilo comportamentale del cliente - fattoriale;
- MONTHLY_BALANCE: saldo mensile del cliente - numerico;
- CREDIT_SCORE: profilo creditizio del cliente, fattoriale ('Poor', 'Standard', 'Good').

Successivamente, sono state individuate, tramite i quartili della distribuzione, ed eliminate le righe contenenti almeno un outlier.

Per l'imputazione dei restanti dati mancanti è stato adottato un metodo non parametrico, implementato tramite la libreria 'mice', applicando l'algoritmo sull'intero dataset.

Si è proseguito osservando il correlogramma (Figura 1), sono stati valutati i valori di correlazione riguardanti le coppie di variabili ed infine rimossa la covariata MONTHLY_SALARY, poiché altamente correlata con ANNUAL_INCOME, MONTHLY_EMI, MONTHLY_BALANCE e MONTHLY_INVESTMENT.

Per le variabili qualitative, si è considerato l'indice Chi-quadro normalizzato. Nessuna coppia presenta valore della statistica test superiore a 0.80, di conseguenza non è stata eliminata alcuna covariata.

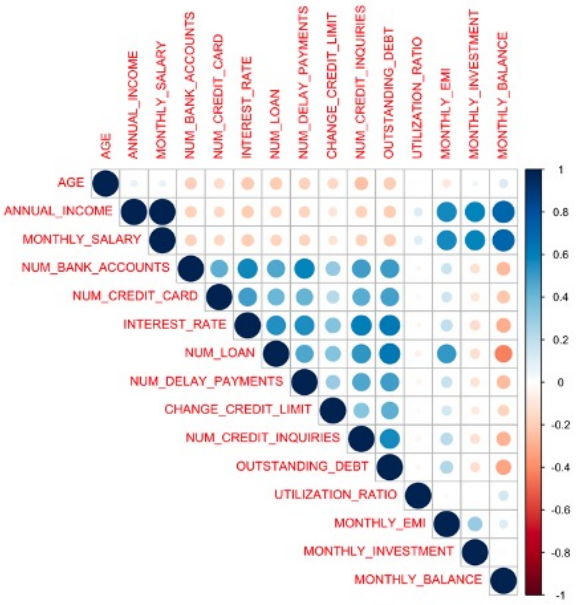


Figura 1 Correlogramma

Con l'ausilio della variabile CUSTOMER_ID, per ogni cliente è stata selezionata in modo casuale una sola riga, ottenendo un dataset dalle dimensioni di 11257 righe e 22 colonne.

2.2 MODEL SELECTION

La selezione delle variabili è stata effettuata tramite l'algoritmo random forest, utilizzando 500 alberi e 12 variabili per la creazione di ogni nodo all'interno di questi. Le variabili in questione sono state individuate tramite un indice di importanza con soglia pari al 30% (Figura 2).

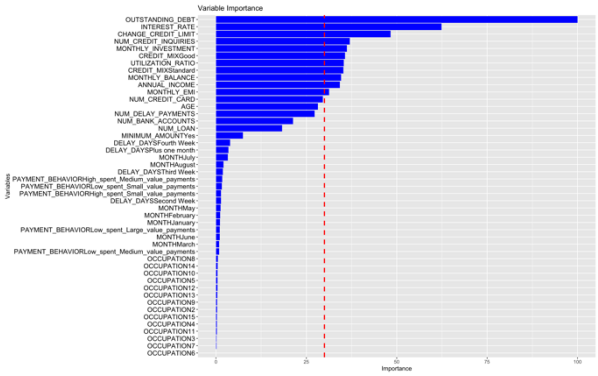


Figura 2 Grafico d'importanza delle variabili

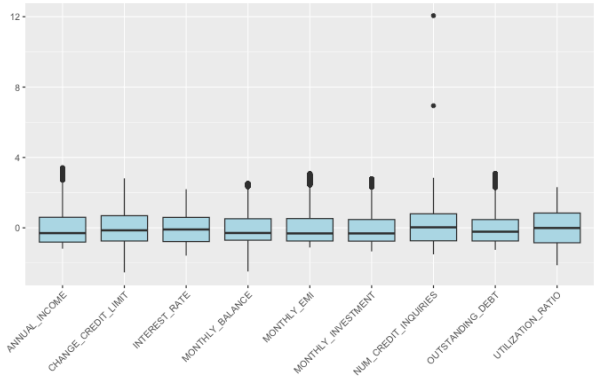
Il dataset così ottenuto si struttura di 11 variabili e 11257 osservazioni. Le variabili esplicative restanti a seguito della model selection sono:

- ANNUAL_INCOME
- INTEREST_RATE
- CHANGE_CREDIT_LIMIT
- NUM_CREDIT_INQUIRIES
- CREDIT_MIX
- OUTSTANDING_DEBT
- UTILIZATION_RATIO
- MONTHLY_EMI
- MONTHLY_INVESTMENT
- MONTHLY_BALANCE

3 CLUSTERING

È stato deciso di trasformare la variabile CREDIT_MIX in una variabile di tipo dummy così da includerla nell'impiego degli algoritmi, ottenendo tre colonne aggiuntive a partire dalle modalità della stessa. Il passaggio successivo ha previsto la standardizzazione delle variabili indipendenti e l'esclusione della variabile target CREDIT_SCORE (Figura 3).

Figura 3 Box-plot delle variabili standardizzate



3.1 ITERATIVE DISTANCE-BASED

Per la famiglia dei metodi iterativi basati sulla distanza, sono stati adottati gli algoritmi K-Means e K-Medians, che utilizzano rispettivamente le misure di distanza Euclidea e Manhattan. Per determinare il valore ottimale di k (numero di cluster) con K-Means, sono stati impiegati due metodi distinti: il Metodo del Gomito e il Metodo della Silhouette.

Il Metodo del Gomito (Figura 4) si basa sull'analisi della Somma delle Differenze al Quadrato, che misura la varianza totale all'interno dei cluster. Il punto in cui il decremento di WSS si stabilizza viene identificato come il "gomito" del grafico, indicando il numero ottimale di cluster.

In parallelo, è stato utilizzato il Metodo della Silhouette (Figura 5) per valutare la coerenza interna dei cluster, calcolando l'indice di silhouette per ciascun valore di k.

Figura 4 Metodo del Gomito

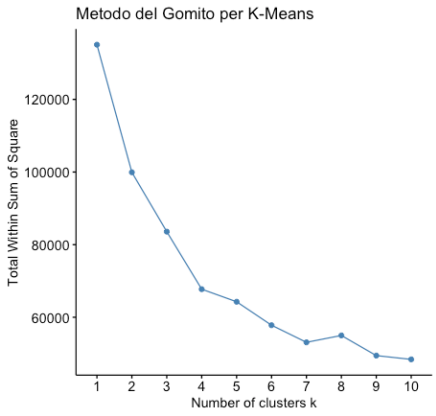
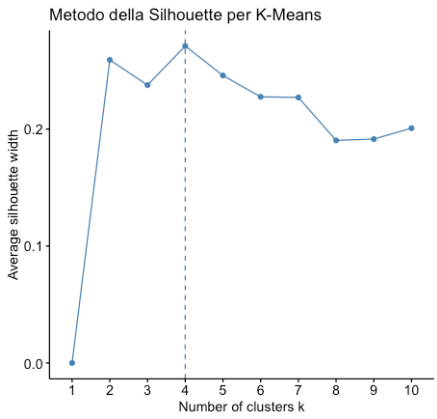


Figura 5 Metodo della Silhouette



In entrambi i casi, il valore più plausibili per k risulta essere quattro. Si è però deciso di fissare il numero di cluster a tre, per permettere un confronto diretto con i risultati ottenuti tramite altri algoritmi.

I risultati ottenuti con i metodi K-Means e K-Medians si sono rivelati molto simili; pertanto, si è deciso di riportare i risultati del K-Means.

Di seguito vengono presentati i grafici a torta rappresentanti le frequenze percentuali delle modalità della variabile risposta all'interno dei tre cluster rilevati (Figura 6):

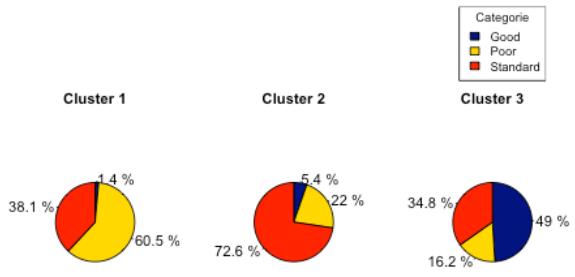


Figura 6 Areogrammi K-Means

Per la valutazione K-Means si presentano gli indici Cluster-Purity e Silhouette:

CLUSTER-PURITY	SILHOUETTE
0.6310	0.2808

3.2 CLUSTERING GERARCHICO AGGLOMERATIVO

Gli algoritmi agglomerativi con approccio "bottom-up" considerano, inizialmente, le singole osservazioni come cluster a sé. Procedono unendo i cluster più vicini tra loro, utilizzando una funzione che ne misura le distanze, ovvero la similarità. La distanza tra due cluster viene calcolata come funzione delle distanze esistenti tra ciascun punto di un cluster e ogni punto appartenente all'altro cluster. L'esecuzione termina quando viene raggiunto il n° di cluster specificato come regola di arresto. Le diverse implementazioni utilizzano metodi di selezione dei cluster da unire differenti ('ward.D', 'single', 'complete', 'average' e 'centroid'). Tali metodi prevedono in un primo momento la determinazione della matrice delle distanze tra le istanze del dataset. La scelta della combinazione ottimale tra metrica di distanza per il calcolo della matrice ('euclidean', 'maximum', 'manhattan', 'canberra' e 'minkowski') e metodo di scelta dei cluster da accorpare è dipesa dai valori degli indici Cluster-Purity e Silhouette. La

coppia più performante è risultata composta da distanza di Manhattan e metodo di Ward.

CLUSTER-PURITY	SILHOUETTE
0.6308	0.3707

Il dendrogramma (Figura 7) è stato tagliato a tre gruppi:

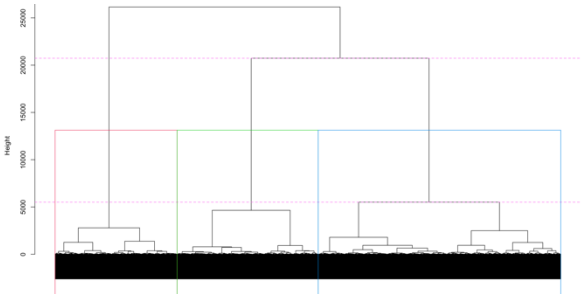


Figura 7 Dendrogramma

In particolare, il criterio adottato dal metodo di Ward è la minimizzazione della somma degli errori quadratici, rispetto al centroide, a seguito dell'unione di due gruppi di osservazioni. A ogni passo viene individuata la fusione che genera il minor incremento della somma dei quadrati delle distanze, così da definire l'ordine di aggregazione dei cluster. La distanza di Manhattan, invece, definisce il grado di similarità delle osservazioni ed è calcolata a partire dalle differenze assolute tra i valori delle variabili.

Dal grafico (Figura 7) si può notare come i tre cluster si formino a un'altezza compresa tra 5510 e 20734; il cluster rosso si forma prima, seguito dal verde e dall'azzurro. Quindi, è possibile affermare che i cluster verde e azzurro contengano osservazioni più dissimili rispetto a quelle del cluster rosso.

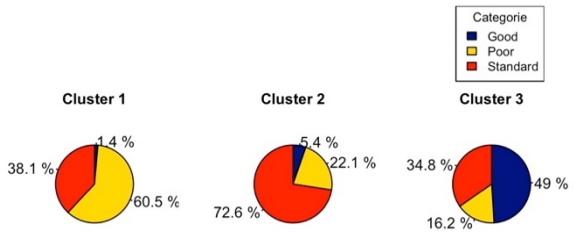


Figura 8 Areogrammi Clustering Gerarchico

Gli areogrammi in Figura 8 mostrano la distribuzione delle modalità della variabile dipendente all'interno dei tre cluster.

3.3 DBSCAN

Un ulteriore metodo considerato per determinare i cluster è stato DBscan. Si basa sull'impiego del concetto di densità definito dal numero di punti che cadono a una certa distanza. È richiesta pertanto la determinazione di due iperparametri che influiscono sull'individuazione dei cluster:

- ϵ : misura del raggio per definire l'appartenenza di due punti allo stesso cluster, questo consente di ben definire i cluster; Per determinare il valore di questo si è considerata la regola del gomito (Figura 9), visualizzando il grafico che mette in relazione la misura delle distanze dai K punti più prossimi e il numero di punti che riportano il valore di tali distanze. Si definisce come ϵ il corrispondente valore sull'asse delle y di cui il grafico riporta l'inizio della crescita 'vertiginosa'. Nel caso in analisi, $\epsilon = 2.05$, corrispondente alla linea verde.

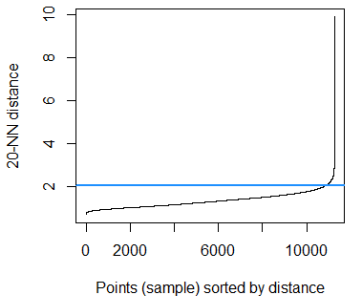


Figura 9 Metodo del Gomito

- k : un cluster può essere considerato tale se al suo interno 'cadono' almeno k punti, iperparametro utile a distinguere chiaramente i cluster. È convenzione imputare a questa costante il valore della dimensionalità del dataset incrementato di un'unità. È stato ritenuto comunque opportuno effettuare uno studio del parametro valutando il suo valore da 13 a 22, registrando poche differenze si è scelto come da prassi k=13.

Si utilizza la distanza di tipo 'euclidea' all'interno dell'algoritmo, data la previa standardizzazione.

Di seguito la tabella che confronta il numero di noise points e il numero di istanze classificate nei vari cluster ottenute con le corrispondenti etichette reali.

	Good	Poor	Standard	Totale punti
Noise points	6	2	8	16
Cluster 1	39	1642	1034	2715
Cluster 2	285	1190	3913	5388
Cluster 3	1538	509	1091	3138

Come si può osservare sono stati rilevati 16 noise points e 3 cluster. Sono 2715 i punti appartenenti al cluster 1, 5388 quelli raggruppati nel secondo cluster e il terzo cluster contiene 3138 punti. Si procede osservando la percentuale di appartenenza delle istanze di ciascun cluster alle etichette reali (Figura 10).

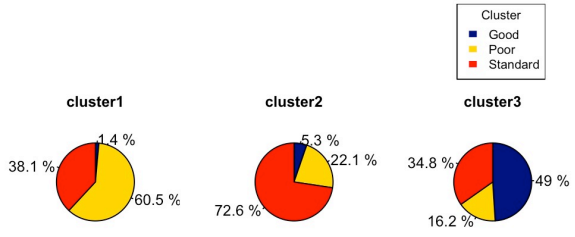


Figura 10 Areogrammi DBscan

Si riportano di seguito i valori dell'indice di Silhouette e Purezza ai fini di un confronto finale, ottenuti escludendo i noise points:

CLUSTER-PURITY	SILHOUETTE
0.6310	0.2815

3.4 CONSIDERAZIONI SUI RISULTATI

I tre metodi testati hanno riportato risultati pressoché analoghi. Il commento sulla distribuzione delle modalità della variabile CREDIT_SCORE all'interno dei tre cluster è comune ai tre algoritmi implementati. Nel cluster 1 si nota una prevalenza di istanze appartenenti alla categoria 'Poor' e un'elevata percentuale di osservazioni di livello 'Standard'. All'intero del cluster 2 si osserva una maggior presenza della modalità 'Standard'. Infine, nel cluster 3 all'incirca la metà delle osservazioni è di classe 'Good', seguita dalla modalità 'Standard'. Si evince, in particolare dal primo e dal terzo cluster, una sovrapposizione del livello intermedio 'Standard' sugli altri due e ciò potrebbe derivare dalla difficoltà di discriminare da parte delle variabili incluse. In alternativa, potrebbe dipendere da un'assenza di differenze

marcate tra i profili creditizi dei clienti 'Standard' e i clienti degli altri due livelli.

Tali risultati sono confermati dai valori degli indici di valutazione dei cluster, indicativamente costanti nei tre metodi.

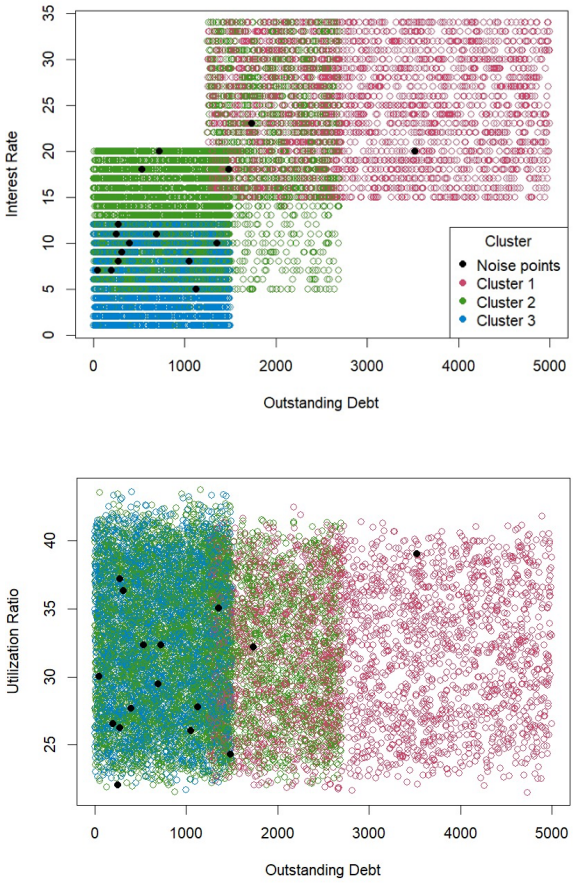


Figura 11 Scatterplot delle covariate Outstanding Debt – Interest Rate e Outstanding Debt – Utilization Ratio

Dal grafico in alto in Figura 11 i tre cluster, ottenuti mediante DBscan, risultano ben definiti, in quanto tale rappresentazione è stata ottenuta discriminando mediante le covariate dall'importanza maggiore (vedi Figura 2). Una situazione opposta si presenta nel secondo grafico, con i cluster sovrapposti rispetto alla variabile UTILIZATION_RATIO, dalla minor capacità divisoria rispetto alla covariata INTEREST_RATE.

4 CLASSIFICAZIONE

Il dataset ottenuto a seguito del pre-processing è stato diviso in 'dataset', contenente l'80% delle istanze di partenza selezionate casualmente, e in 'test', include il restante 20%. Inoltre, per le reti neurali è stata effettuata la trasformazione della covariata categoriale CREDIT_MIX in una variabile di tipo dummy. Invece, i metodi di classificazione KNN, SVM e reti neurali hanno richiesto la standardizzazione dei valori.

4.1 KNN

Il primo metodo considerato è il 'KNN', K-nearest neighbours. L'algoritmo consiste nella definizione delle distanze tra le osservazioni (si è utilizzata la distanza euclidea) e l'assegnazione dell'etichetta più presente dei K punti più vicini.

Per poter implementare l'algoritmo si deve decidere il valore di K, il criterio adottato è stata quello di procedere a tentativi: si è assegnato ciclicamente un valore a K (da 1 a 200) e se ne è valutato il tasso di errore (Figura 12), la percentuale di errori commessi nelle previsioni, di cui si è individuato il minor valore e il K corrispondente, conseguentemente assegnato.

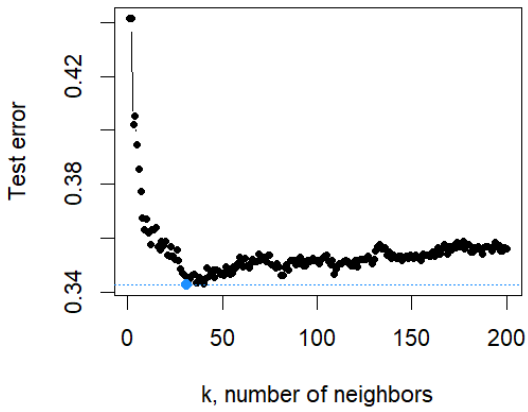


Figura 12 Andamento dell'errore al variare di k

Nell'analisi K=31 si riportano i risultati di seguito:

	Reference		
Prediction	**Good**	**Poor**	**Standard**
Good	82	31	84
Poor	4	404	138
Standard	288	234	988

Globalmente, il modello classifica correttamente il 65,42% delle istanze, all'interno dei gruppi però ci sono delle differenze, se ne riportano le metriche di valutazione:

	Good	**Poor**	**Standard**	**Overall**
Accuracy	-	-	-	0,6542
Sensitivity	0,21925	0,6039	0,8165	-
Specificity	0,9388	0,9104	0,4995	-
Pos Pred Value	0,41624	0,7399	0,6543	-
Neg Pred Value	0,85798	0,8448	0,7012	-

4.2 SVM

L'obiettivo principale di un Support Vector Machine (SVM) è individuare l'iperpiano ottimale che separa le classi di dati con il massimo margine. In altre parole, si mira a massimizzare la distanza tra i punti di dati di classi diverse più vicini all'iperpiano, definiti support vectors.

Sono state utilizzate funzioni kernel di tipo lineare e radiale. L'algoritmo prevede la scelta di alcuni parametri:

- Parametro di penalizzazione C: questo parametro controlla il compromesso tra una classificazione corretta dei dati di training e la massimizzazione del margine. Un valore elevato di C tende a classificare correttamente tutti i dati di training, mentre un valore più basso consente un margine più ampio, favorendo una migliore generalizzazione.
- Parametro sigma (nel caso di kernel radiale): sigma controlla l'estensione dell'influenza dei punti di dati. Un sigma ridotto implica che solo i punti vicini abbiano un'influenza significativa, mentre un sigma maggiore amplia questa influenza.

Il confronto tra i modelli è stato effettuato tramite validazione incrociata (cross-validation) sul set di training, con l'obiettivo di valutare le prestazioni di ciascun modello e individuare la combinazione ottimale di parametri.

I risultati migliori sono stati riscontrati con l'utilizzo del kernel radiale, il quale ha mostrato una maggiore capacità di generalizzare i dati non linearmente separabili. Sono stati utilizzati i seguenti valori ottimali per i parametri: C=1 e sigma=0,1.

Dopo aver effettuato la previsione sul test set, abbiamo ottenuto la seguente matrice di confusione:

	Reference		
Prediction	**Good**	**Poor**	**Standard**
Good	308	106	196
Poor	2	387	125
Standard	64	176	889

Nella tabella che segue sono riportate le metriche di valutazione sul dataset di test:

	Good	Poor	Standard	Overall
Accurancy	-	-	-	0.7031
Sensitivity	0.8235	0.5785	0.7347	-
Specificity	0.8393	0.9198	0.7699	-
Pos Pred Value	0.5049	0.7529	0.7874	-
Neg Pred Value	0.9598	0.8378	0.7144	-

4.3 RANDOM FOREST

Il terzo metodo di classificazione implementato è la random forest, una tecnica di apprendimento che genera una moltitudine di alberi decisionali durante la fase di addestramento e definisce le previsioni sulla base della classe modale individuata. Per determinare il n° ottimale di alberi, 'ntree', e quante variabili scegliere casualmente per la creazione dei nodi di ciascun albero, 'mtry', è stata applicata la tecnica 'repeated cross-validation', utilizzando 10 fold e valutando i diversi modelli testati tramite la metrica di accuratezza. Per incrementare la robustezza dei risultati, la cross-validation è stata ripetuta tre volte. Il seguente grafico mostra l'andamento della metrica di valutazione al variare dei due iperparametri (Figura 13).

La combinazione migliore è risultata essere:

- 'mtry' pari a 2
- 'ntree' pari a 2000

Con gli iperparametri così determinati è stato eseguito un modello random forest sul dataset di training, 'dataset', ottenendo una stima dell'errore 'Out-of-Bag' pari al 31.55%. Tale valore, relativamente alto, potrebbe essere causato dalla scarsa separabilità della classe 'Standard' dalle altre, condizione di cui si ha avuto prova in precedenza.

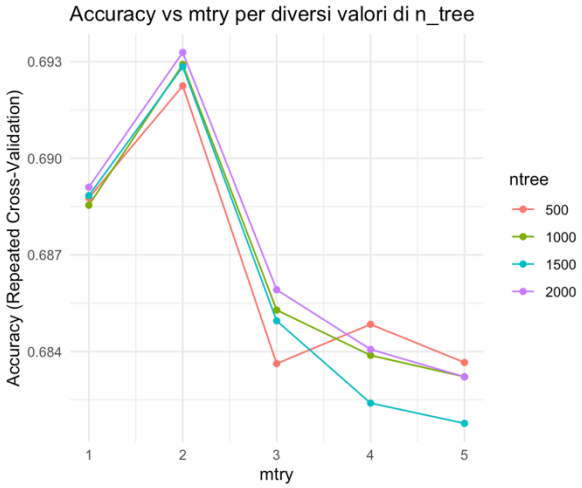


Figura 13 Andamento dell'Accuracy al variare degli iperparametri

Nel grafico sottostante (Figura 14) è rappresentato l'andamento dell'errore 'Out-of-Bag' generale e specifico per ciascuna classe, in funzione del numero di alberi costruiti. Analogamente per le quattro curve si apprezza una rapida riduzione dell'errore, che si stabilizza intorno ai 500 alberi. Qualora i tempi di calcolo fossero risultati eccessivamente elevati si sarebbe potuto ridurre il numero a 500, senza compromettere significativamente le prestazioni del modello. La classe 'Standard' presenta il tasso d'errore più basso, seguita dalle modalità 'Poor' e 'Good'. Inoltre, l'andamento dell'errore per ciascuna classe mostra tendenze simili rispetto all'errore OOB complessivo. Questi risultati indicano che il modello è facilitato nella previsione della classe 'Standard' e, al contrario, riscontra maggiori difficoltà a individuare i clienti 'Good'.

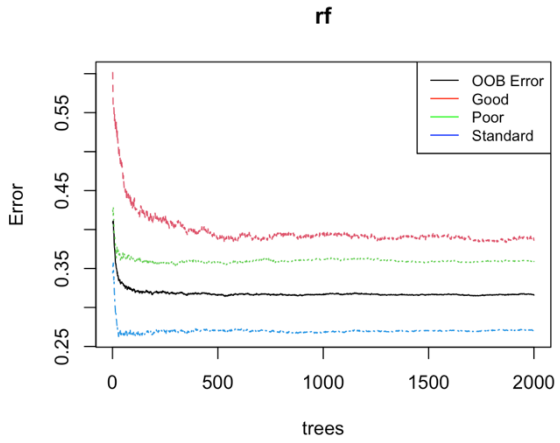


Figura 14 Andamento dell'errore Out-of-Bag

Dopo aver realizzato le previsioni sul dataset di test, è stata ottenuta la seguente matrice di confusione:

	Reference		
Prediction	**Good**	**Poor**	**Standard**
Good	233	3	138
Poor	85	418	166
Standard	153	143	914

Le metriche di valutazione sul dataset di test hanno prodotto i seguenti valori:

	Good	Poor	Standard	Overall
Accurancy	-	-	-	0.6946
Sensitivity	0.4947	0.7411	0.7504	-
Specificity	0.9209	0.8514	0.7140	-
Pos Pred Value	0.6230	0.6248	0.7554	-
Neg Pred Value	0.8733	0.9078	0.7085	-

4.4 RETI NEURALI

Le reti neurali realizzano classificazioni combinando la potenza dei neuroni, contenuti negli strati nascosti, tramite le funzioni di attivazione.

Si è scelto di implementare una rete neurale con un solo strato nascosto, composto da 512 neuroni con funzione di attivazione 'relu', e layer di output formato da 3 unità (pari al n° di livelli della variabile target) con funzione di attivazione 'softmax', specifica per la classificazione. L'addestramento della rete sul dataset di addestramento è stato ripetuto per 50 epoche.

Dopo aver realizzato le previsioni sul dataset di test, è stata ottenuta la seguente matrice di confusione:

	Reference		
Prediction	**Good**	**Poor**	**Standard**
Good	228	83	147
Poor	3	415	152
Standard	143	171	911

Nella tabella sottostante si possono leggere i valori delle metriche di valutazione, calcolate sul dataset di test:

	Good	Poor	Standard	Overall
Accurancy	-	-	-	0.6874
Sensitivity	0,6096	0,6203	0,7529	-
Specificity	0,8776	0,9021	0,6989	-
Pos Pred Value	0,4978	0,7281	0,7437	-
Neg Pred Value	0,9187	0,8491	0,7091	-

4.5 CONSIDERAZIONI SUI RISULTATI

L'accuracy rilevata nei quattro algoritmi si stabilizza su valori mediocri, dal più elevato al più basso: SVM con 0.70, random forest 0.69, reti neurali 0.69 e KNN con 0.65. Al fine di incrementare le performance si è utilizzato un vettore di pesi, all'interno degli algoritmi, per bilanciare le frequenze delle categorie. Tale strategia non ha portato al miglioramento atteso. In conclusione, le scarse prestazioni possono essere imputate nuovamente alle variabili considerate e alla poca distanza dei tre profili creditizi nel campione.

In merito all'indice di specificità, i risultati dai quattro metodi sono comuni e indicano 'Standard' come classe più difficoltosa da individuare, con i valori più bassi della metrica e compresi nell'intervallo [0.50, 0.77]. Al contrario, le categorie 'Good' e 'Poor' riportano valori al di sopra dell'83%. L'attendibilità delle previsioni è confermata dal Negative Predictive Value.

BIBLIOGRAFIA E SITOGRAFIA

https://www.datacamp.com/tutorial/hierarchical-clustering-R

https://cran.r-project.org/web/packages/clValid/vignettes/clValid.pdf

https://medium.com/@vdeshpande551/understanding-the-inner-workings-of-svm-from-data-preprocessing-to-model-deployment-cdc9d72a2d34#:~:text=Data%20Preprocessing%3A%20Data%20preprocessing%20is,can%20be%20used%20by%20SVM.

https://machinelearningmastery.com/tune-machine-learning-algorithms-in-r/

https://sefidian.com/2022/12/18/how-to-determine-epsilon-and-minpts-parameters-of-dbscan-clustering/

https://machinelearningmastery.com/repeated-k-fold-cross-validation-with-python/Statistica per Data Science con R - V. 03