Classificazione : previsioni di bancarotta

Aina Belloni, 829768

June 23, 2020

Abstract

Nel presente documento ho affrontato il problema di classificazione di 7027 società polacche, suddivise in due classi : fallite dopo 5 anni e ancora operative. Il lavoro che ho svolto si è articolato su più fasi, una prima fase di pre-processing e pulizia del dataset, è stata la fase più impegnativa in quanto il dataset presentava diversi problemi. Successivamente ho utilizzato una porzione di unità statistiche per la fase di apprendimento in modo da riuscire a classificare altre società di cui non si conosce la classe. Ho creato diversi dataset, i quali differivano per trasformazioni e variabili ridotte. I metodi utilizzati su questi dataset sono stati la Regressione Logistica, Analisi Discriminante Lineare e Quadratica e K-Nearest Neighbours. Infine ho analizzato, tramite strumenti appositi come la Confusion Matrix e le curve ROC, quali avrebbero prodotto una stima più accurata della classe di appartenenza.

1 Introduzione

Il Data-Mining, e soprattutto le classificazione, sono diventati ormai fondamentali nella realtà aziendale. La previsione di un fallimento è di grande importanza nel processo decisionale economico. L'obiettivo di questo report è appunto quello di arrivare ad un modo sufficientemente accurato per determinare se una nuova società con determinate caratteristiche, può considerarsi in pericolo di fallimento o no.

Per procedere con la classificazione è necessario suddividere le unità statistiche presenti in un dataset in training set e test set. Questo ci permette di far comprendere alla macchina l'ambiente che si vuole studiare, e fornisce esempi di come sono state classificate determinate osservazioni.

2 Materiali e metodi

2.1 Materiali

Il dataset di riferimento (1year.arff) contiene dati riguardanti la previsione di fallimento delle società polacche. Il periodo considerato per le società fallite è dal 2000 al 2012, mentre per le società ancora operative dal 2007 al 2013. I dati sono stati raccolti dal servizio di informazione sui mercati emergenti (EMIS), che è un database contenente informazioni sui mercati emergenti di tutto il mondo. In particolare il dataset considerato contiene 64 variabili che sono i tassi finanziari dal 1° anno del periodo di previsione ed una 65° variabile che è un'etichetta di classe corrispondente che indica se la società è fallita o no dopo 5 anni. I casi (bilanci) considerati sono 7027, di cui 271 rappresentano società in bancarotta e 6756 aziende operative nel periodo di previsione.

Il dataset presenta diversi problemi :

- Missing Values
- Classi altamente sbilanciate
- Valori anomali e outliers
- Multicollinearità (alta correlazione tra diverse variabili)

2.1.1 Preparazione del dataset

Utilizzando il software Rstudio ho creato dal file .arff un dataframe 7027x65, trasformando la variabile class in una variabile factor con due livelli (0: not bankrupted, 1: bankrupted) mentre le altre 64 variabili sono di tipo numerico. Inoltre i missing values si presentavano sottoforma di punto di domanda "?" e per comodità ho sostituito questo carattere con il valore NA.

2.2 Metodi

2.2.1 Pre - processing e pulizia del dataset

Una delle fasi più critiche e più importanti è la fase di preparazione del dataset. Il dataset che avevo a disposizione conteneva dei missing values, ho dovuto così analizzarli, capire che natura avessero e trattarli nel modo adeguato. Ci sono in particolare due strategie per trattare i missing values :

Strategia passsiva –eliminare le osservazioni o le variabili contenenti missing (tramite la casewise deletion o la pairwise deletion)

Strategia attiva – imputare valori plausibili al loro posto, come il valore medio, il valore medio condizionato alla classe o stimando il valore con un modello di regressione.

Nella classificazione un passaggio fondamentale è quello di suddivisione del dataset in : training set, validation set e test set. In particolare il training (solitamente contenente il maggior numero di osservazioni) viene utilizzato per l'apprendimento del metodo di classificazione. Il validation set serve per approfondire le prime analisi così da stabilire i parametri. Il test set invece è utilizzato solo per la verifica e validazione finale, non viene utilizzato nella fase di apprendimento.

La suddivisione è avvenuta nel seguente modo :

Training set 80%, ulteriormente diviso in Training 65% e Validation 45%;

Test set : 20%.

Procedo con un'analisi esplorativa del training set, noto che ci sono molti valori anomali e dati non confrontabili, così standardizzo le variabili ed elimino le osservazioni con valori in modulo maggiori di 9.

Altra analisi importante è verificare che le classi siano bilanciate, nel mio caso non lo erano, anzi presentavano un forte sbilanciamento. Sono previste diverse soluzioni le due più comuni sono:

- SMOTE (Synthetic Minority Oversampling Technique)
- RUS (Random Under Sampling)

La prima prevede la generazione di nuove unità statistiche per la classe minore, così da arrivare ad essere in pari con la classe magiore. La seconda invece seleziona casualmente delle osservazioni appartenenti alla classe maggiore così da essere in numero uguale alla classe con meno osservazioni.

Ultimo problema di pre-processing da trattare è la multicollinearità e la grande quantità di variabili. La multicollinearità è la presenza di molte variabili altamente correlate tra loro. E' necessaria una feature reduction o selection così proviamo a procedere in due modi :

- Stepwise selection tramite modello di regressione logistica
- PCA (Analisi delle componenti principali)

2.2.3 Regressione logistica semplice

Applichiamo il primo metodo di classificazione a tutti i 6 dataset creati. Il modello di regressione logistica permette di mettere in relazione di dipendenza di un attributo dicotomico con diverse variabili indipendenti.

Questo modello ci ha permesso precedentemente di individuare le variabili indipendenti con maggiore potere esplicativo, adesso la usiamo per cercare la combinazione lineare di variabili che meglio discrimina tra i gruppi; così da riuscire a stimare la probabilità del possesso dell'attributo per una nuova unità statistica su cui sono state osservate le variabili considerate.

Calcolo il seguente modello :

$$logit\big(\pi(x)\big) = \beta_0 + \sum_{j=1}^{p} \beta_j \, x_j$$

Dove $\pi(x)$ è la probabilità a posteriori che la classe di appartenenza rispetto alla variabile x sia la classe positiva. Utilizziamo la funzione di questa probabilità che è il link logit. β_0 indica l'intercetta e β_j sono i coefficienti delle variabili x che rappresentano la variazione nel logit corrispondente ad un incremento unitario di x. I parametri vengono stimati con il metodo della massima verosimiglianza.

Poi con la regola di Bayes decido che classe assegnare alla nuova unità statistica (utilizzando un cut-off di 0.5).

2.2.4 Analisi discriminante lineare

Passiamo al secondo metodo, simile alla regressione logistica, ma stima la distribuzione delle variabili indipendenti X separatamente per ogni classe di risposta. Date le distribuzioni così calcolate, si ottiene attraverso il teorema di Bayes la regola per la scelta della classe.

Questa analisi prevede però delle ipotesi :

- Le variabili devono essere normalmente distribuite
- Tutte le classi devnono avere matrice di varianza e covarianza in comune.

$$f_k(x) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T \boldsymbol{\Sigma}^{-1}(x-\mu)\right) \sim N(\mu_k, \Sigma)$$

2.2.5 Analisi discriminante quadratica

Questo approccio estende quello di LDA, infatti per applicarlo basta che le variabili siano normalmente distribuite, ma le classi possono avere varianza e covarianza non in comune. E' un metodo più flessibile e solitamente da risultati più accurati.

2.2.6 KNN

Ultimo metodo considerato è quello KNN (K-nearest neighbour), è un metodo non parametrico in quanto non sono necessarie assunzioni sulla distribuzione di partenza. L'unico parametro considerato è K, il parametro di tuning, che corrisponde ai numeri di vicini che il metodo considera.

E' necessario definire una misura di distanza tra le osservazioni, in questo caso definiamo la distanza euclidea :

$d_{euc}(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$ dove n è il numero di dimensioni nel vettore

Per determinare il parametro k facciamo una simulazione in modo da individuare il valore di k che minimizza l'errore. Una volta individuato si calcola la distanza tra un nuovo punto e le k osservazioni più vicine, si individua la categoria di target di tutte le k osservazioni vicine e si assegna al novo punt la classe che maggiormente è presente in queste.

2.2.7 Calcolare qualità del modello e accuratezza

Per confrontare i diversi metodi e vedere i risultati della previsione, calcoliamo la confusion matrix, ovvero una matrice contenente tutte le classi stimate delle nuove osservazioni.

Da questa matrice possiamo calcolare l'accuratezza, la sensibilità e la specificità nel seguente modo :

		Predicted class		
		− or Null	+ or Non-null	Total
True class	− or Null	True Neg. (TN)	False Pos. (FP)	N
	+ or Non-null	False Neg. (FN)	True Pos. (TP)	P
	Total	N*	P*	

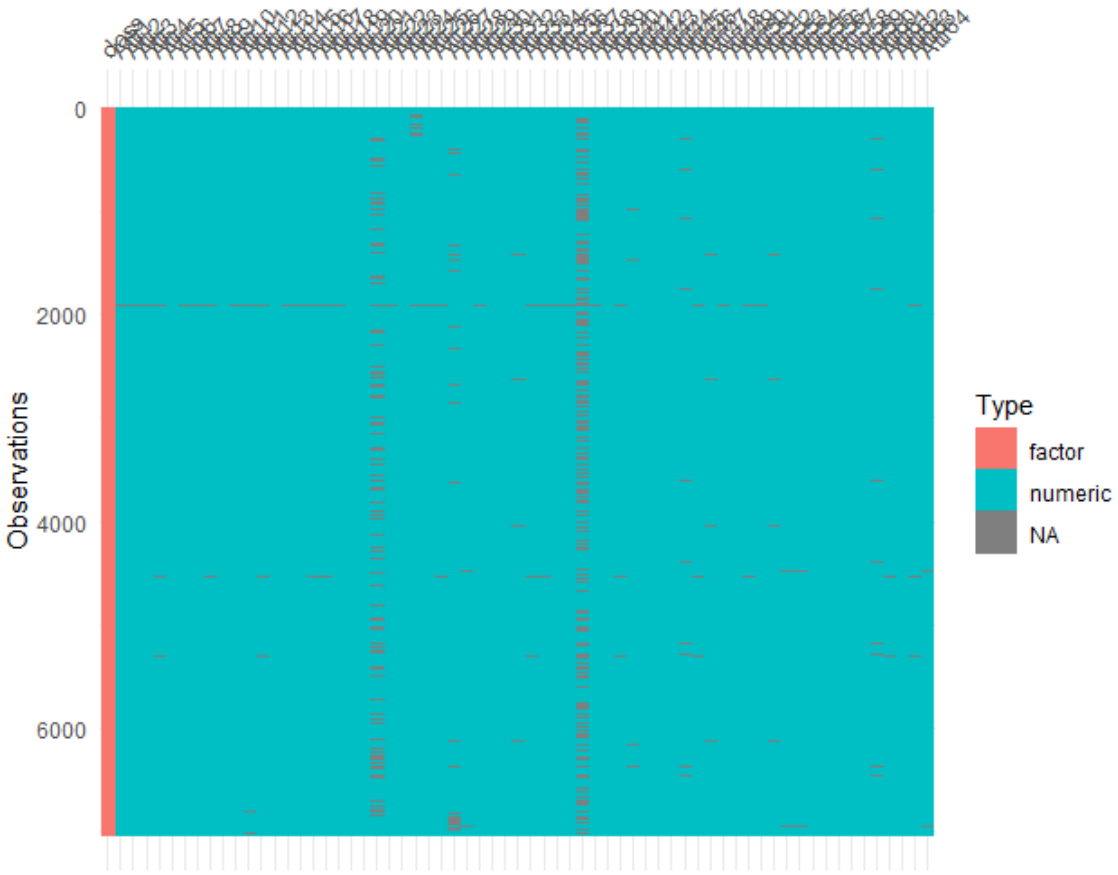
Name	Definition	Synonyms
False Pos. rate	FP/N	Type I error, 1−Specificity
True Pos. rate	TP/P	1−Type II error, power, sensitivity, recall
Pos. Pred. value	TP/P*	Precision, 1−false discovery proportion
Neg. Pred. value	TN/N*	

Possiamo anche visualizzare graficamente la capacità discriminatoria del test, grazie alle curve ROC che traccia la probabilità di un risultato vero positivo (sensibilità) in funzione della probabilità di un risultato falso positivo per una serie di punti di cut-off.

3 Risultati

Missing Data

Da una prima analisi dei dati si nota un'elevata presenza di missing values, in particolare più della metà delle osservazioni contiene NA (54%).



Osserviamo che gli attributi 21 e 37 hanno un'elevata concentrazione di NA, rispettivamente il 24% e 39%. Studiando la matrice di correlazione con altre variabili notiamo che non ci sono correlazioni particolarmente elevante con questi due attributi quindi scarto la possibilità di eliminare interamente le colonne dal dataset.

Essendo in numero così elevato non procedo con l'eliminazione di ogni osservazione contenente missing, bensì, prima di passare ad altre analisi, decido di imputare al loro posto il valore medio condizionato alla classe.

Training: analisi esplorativa e standardizzazione

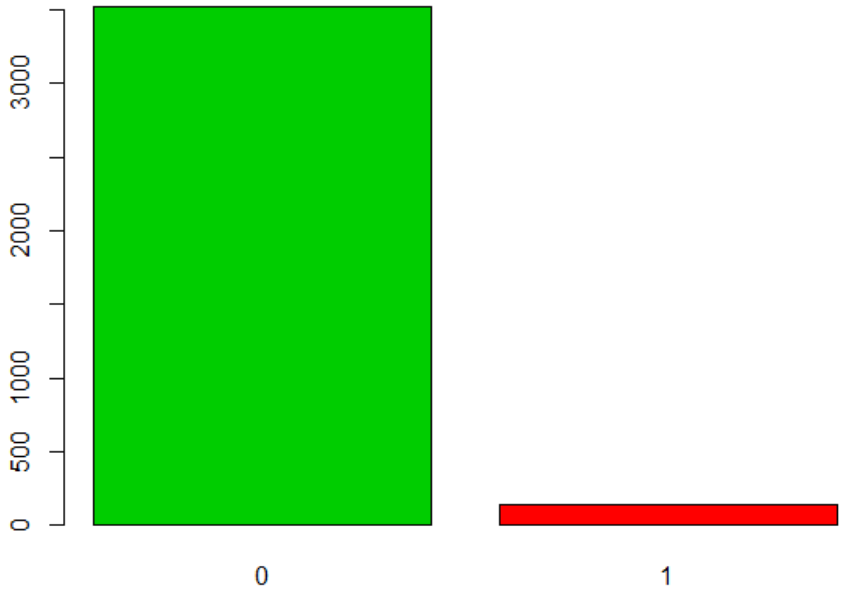
Facendo un'analisi esplorativa del training utilizzando soprattutto boxplot viene subito all'occhio la presenza di molti valori anomali e outliers, e media e mediana che non solo non corrispondono ma hanno valori decisamente diversi.

Una prima operazione da fare sicuramente è **standardizzare** e centrare le variabili (esclusa la class).

Successivamente approfondisco l'esplorazione dei valori anomali e outlier, calcolo che sono presenti nell'82% delle osservazioni. Provo quindi a eliminare osservazioni che hanno almeno un valore assoluto maggiore di 9.

Training : bilanciamento delle classi

La variabile target class contiene due livelli, ma le osservazioni appartenenti a questi sono altamente sbilanciati, infatti il 96% ha livello 0 mentre solo il 4% ha livello 1, come mostra il grafico.

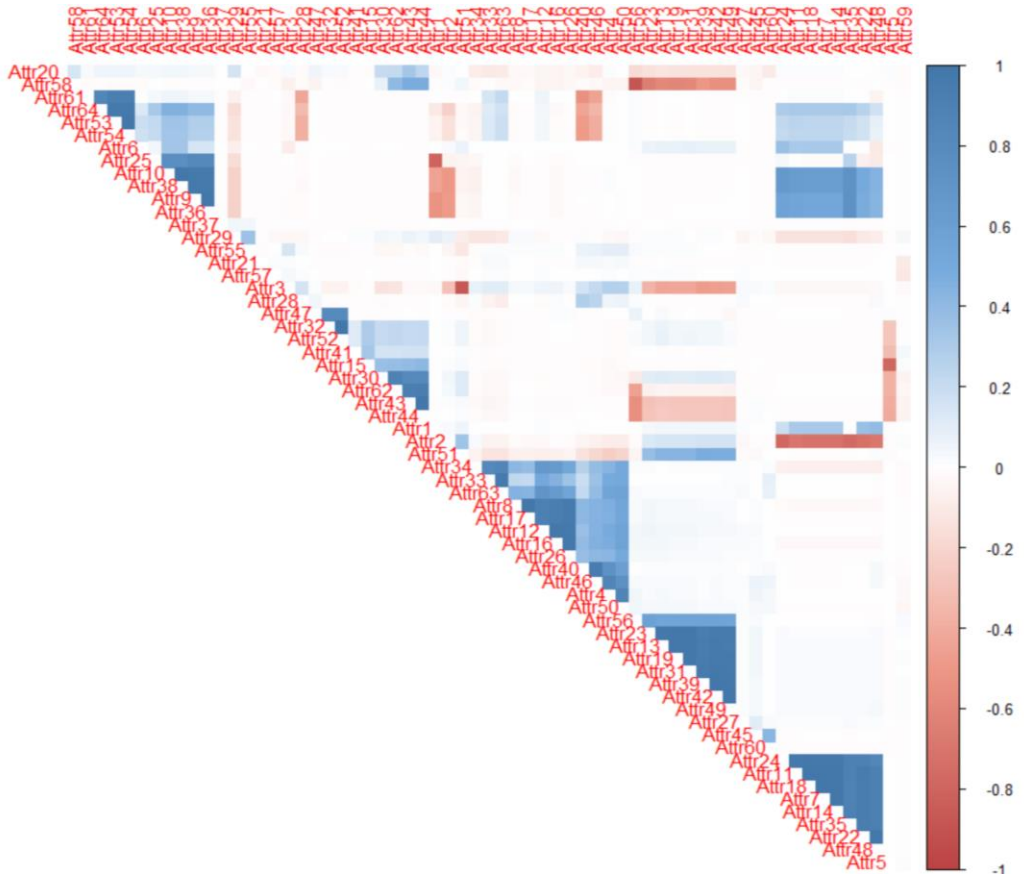


Essendo così sbilanciate le classi ho deciso di effettuare prima la tecnica RUS in modo da ridurre del 50% le aziende corrispondenti al livello 0. Successivamente con SMOTE incrementare la classe minore. In questo modo otteniamo un training dataset bilanciato composto da 3315 righe di cui 1691 "not bankrupt" e 1624 "bankrupt".

Poi ho provato a creare un altro dataset ribilanciato solo con SMOTE, ottenendo da 3606 osservazioni a 6881.

Training : matrice di correlazione e multicollinearità

Un altro grande problema di questo dataset è l'elevata presenza di variabili altamente correlate, come mostra il grafico sottostante. Ci troviamo chiaramente in un caso di multicollinearità.



Avendo molte variabili e altamente correlate decido di applicare due diversi approcci :

- la PCA, prendendo in considerazione in entrambi i dataset le prime 20 componenti (che spiegano il 90% della varianza spiegata).

- Stepwise selection con il modello di regressione logistica

Classificazione

Finite tutte le analisi iniziali sul training dataset, passo a verificare l'applicabilità dei diversi metodi di classificazione con il validation.

I problemi non sono pochi, così ho creato diversi dataset per testare tutte le possibili trasformazioni :

- Dataset ridotto con stepwise ma sbilanciato
- Dataset ridotto con stepwise e bilanciato con RUS+SMOTE
- Dataset ridotto con stepwise e bilanciato con SMOTE
- PCA
- PCA bilanciato con RUS+SMOTE
- PCA bilanciato con SMOTE

Verifica assunzioni analisi discriminante

Prima di svolgere un'analisi discriminante lineare ho bisogno che le variabili soddisfino determinati requisiti:

- Varianza comune condizionatamente alle classi
- Covarianza comune condizionatamente alle classi
- Ipotesi di normalità (con Shapiro Test)

Per l'analisi discriminante quadratica l'unica richiesta invece è che siano normalmente distribuite.

Boxplot condizionati alla classe di alcune variabili :

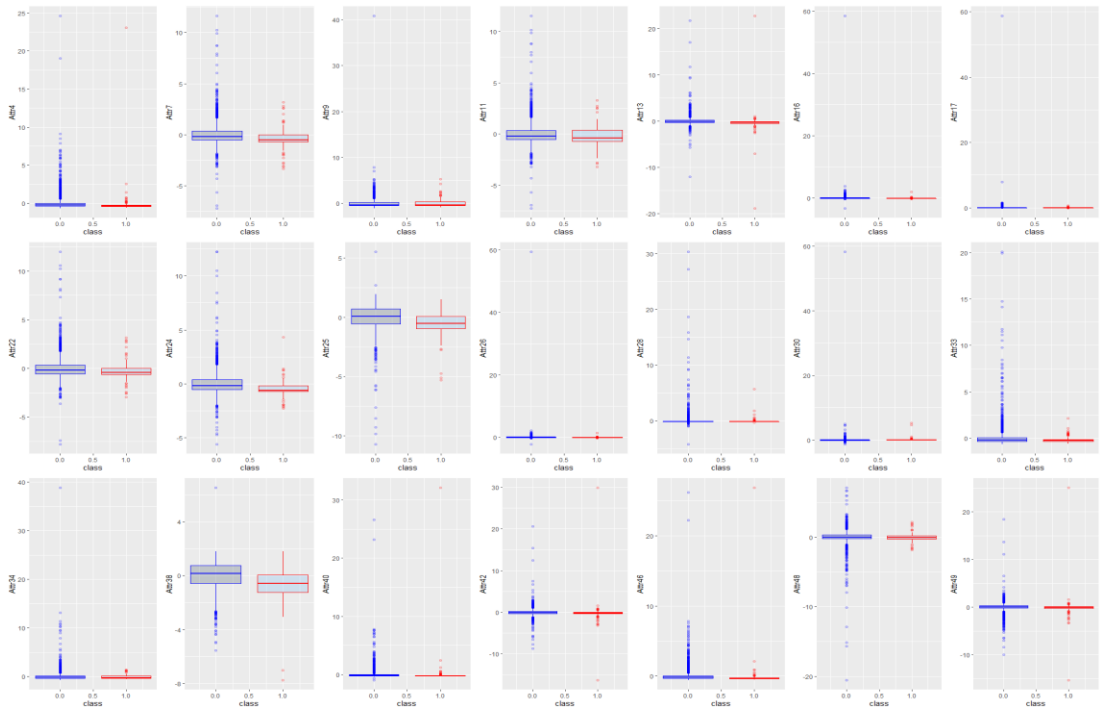
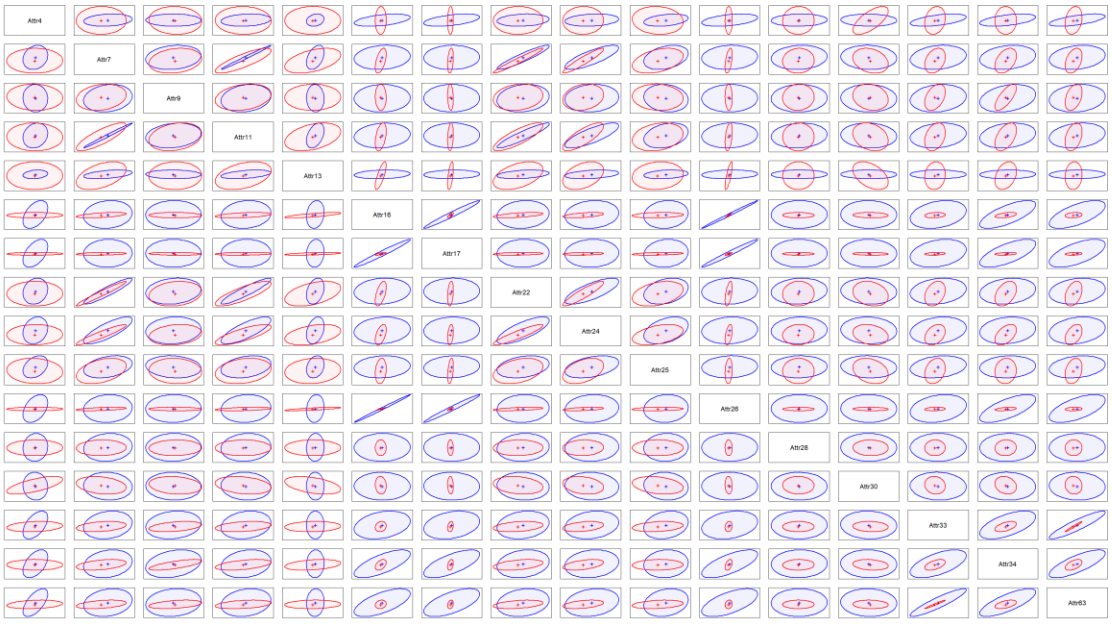


Grafico con ellissi per valutazione delle covarianze : (visualizzate 16 variabili)



Le ipotesi di varianza e covarianza comune non sono del tutto soddisfatte.

Effettuando lo Shapiro Test osservo che i pvalue non sono sufficientemente elevati per poter accettare l'ipotesi di normalità, così posso concludere che le variabili non sono normodistribuite.

Nonostante i pessimi risultati della verifica delle assunzioni preliminari per l'applicazione dei metodi LDA e QDA, provo comunque ad applicarli.

Verifica metodi su validation

Testando i metodi di classificazione sul dataset di validazione otteniamo:

	LDA	QDA	Regressione.Logistica	KNN	k.scelto
acc.1	0.9547764	0.3917683	0.9598577	0.9634146	6
acc.3	0.7586382	0.5381098	0.7550813	0.8485772	1
acc.4	0.7433943	0.4806911	0.7576220	0.8734756	1

In tutti e tre i casi è stato utilizzato nella fase di apprendimento un dataset risotto tramite stepwise selection. Nel primo caso è stato non è stato fatto alcun bilanciamento, nel secondo caso è stato fatto un bilanciamento con RUS+SMOTE, nel terzo caso il bilanciamento con SMOTE.

Osserviamo che a parte la QDA , gli altri metodi hanno prodotto un'accuracy molto alta. Il caso che ha prodotto un'accuracy maggiore è stato quello senza alcun bilanciamento, mentre quasi non si percepisce differenza tra le due modalità distinte di bilanciamento.

Per quanto riguarda invece il training a cui abbiamo applicato la PCA vediamo :

	LDA	QDA	Regressione.Logistica	KNN	k.scelto
acc.5	0.9608740	0.17530488	0.9613821	0.9613821	2
acc.6	0.9512195	0.09044715	0.9308943	0.9598577	1
acc.7	0.8231707	0.11432927	0.8445122	0.9435976	1

La prima riga si riferisce alla PCA non bilanciata, la seconda riga bilanciata con RUS+SMOTE e la terza solo con SMOTE.

Anche per quanto riguarda questo confronto notiamo che la QDA ha valori sorprendentemente bassi, mentre gli altri metodi hanno accuracy molto alta.

Osservazioni :

- L'accuracy è molto alta perché la maggior parte dei casi è di classe 0, e guardando la confusion matrix osservo che in quasi la totalità dei casi il modello ha predetto "appartenenza alla classe 0"
- In molti casi il K suggerito nel metodo KNN è 1, questo potrebbe portare ad errori di overfitting. Successivamente verifichiamo.

Fase di test

Utilizzo ora come dataset di training l'unione di sub training e validation. Applico le trasformazioni concordate precedentemente, ovvero standardizzazione ed eliminazione di outlier, e creo 3 diversi dataset con le seguenti caratteristiche :

- Ridotto con stepwise selection
- Ridotto con stepwise selection e bilanciato con RUS+SMOTE
- Ridotto con PCA

Anche il test set deve essere standardizzato e ridotto e quando utilizzato con il training ridotto con PCA deve essere trasformato con i pesi utilizzati.

Dai risultati precedenti decido di applicare solo 3 dei 4 metodi :

- LDA
- Regressione logistica
- KNN (con i k scelti)

Accuracy :

	Accuracy Regressione Logistica	Accuracy LDA	Accuracy KNN	K usato
Training ridotto	0.9454	0.9559	0.9559	6
Training ridotto e bilanciato	0.739	0.7489	0.8378	2
Training ridotto con PCA	0.9552	0.9545	0.9559	2

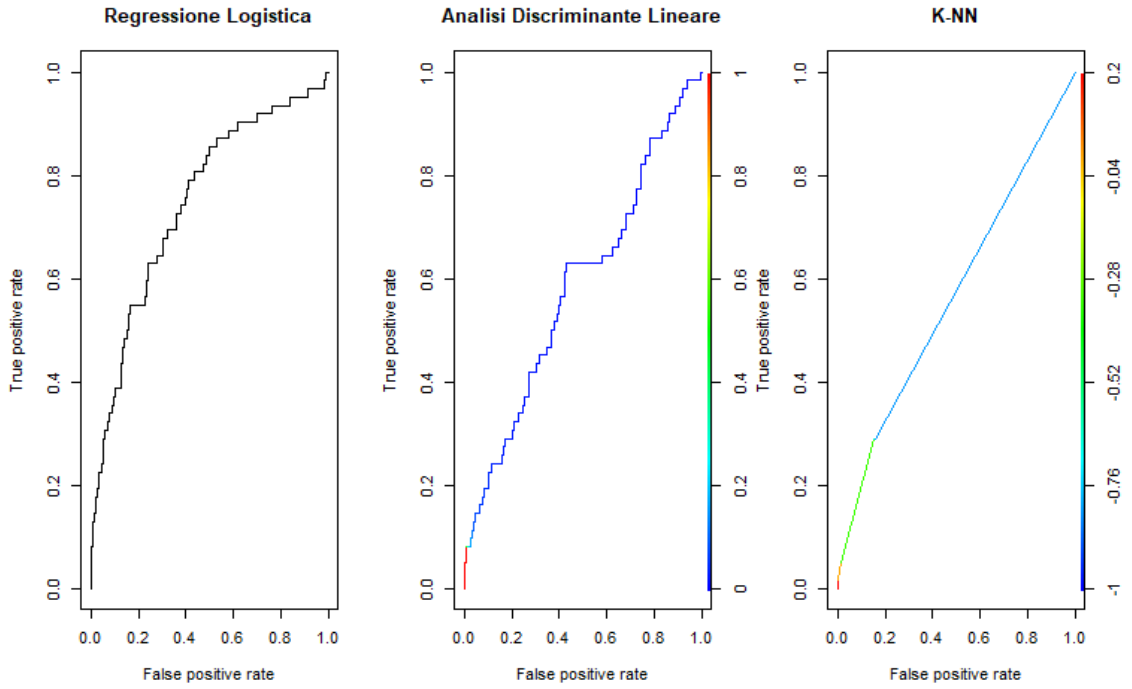
Specificity (classe positiva : “0”) :

	Regressione Logistica	LDA	KNN
Training ridotto	0.08	0.08	0.016
Training ridotto e bilanciato	0.14	0.43	0.41
Training ridotto con PCA	0	0	0

Training e test error :

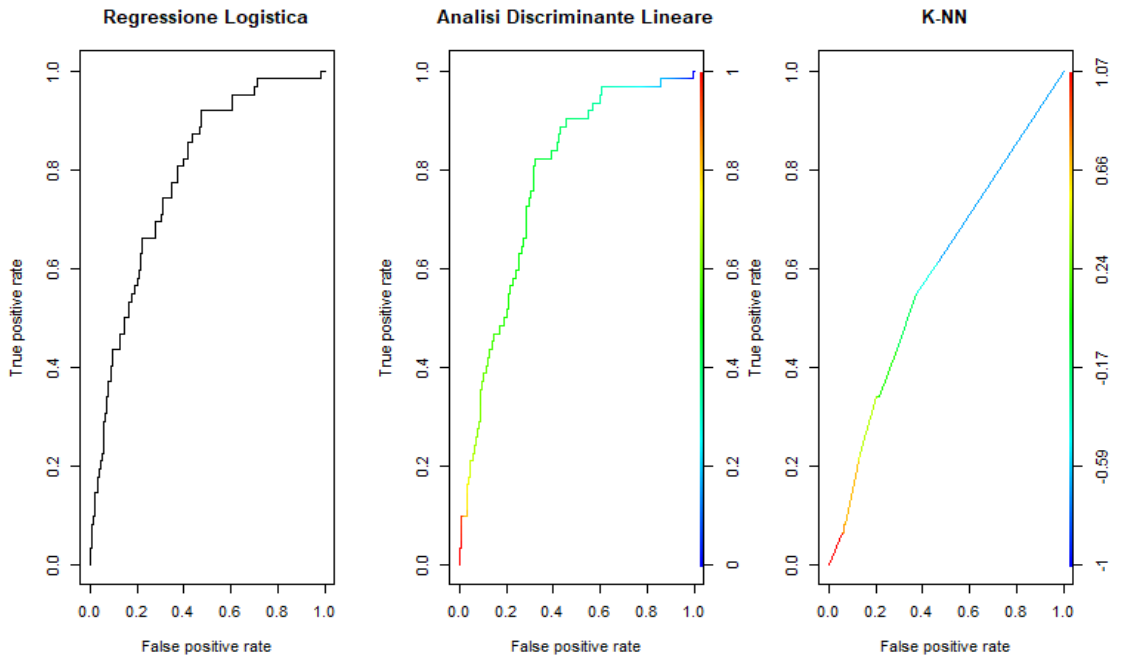
	Training error	Test error
Training ridotto	0.031	0.044
Training ridotto e bilanciato	0.23	0.25
Training ridotto con PCA	0.031	0.044

ROC curve per dataset ridotto



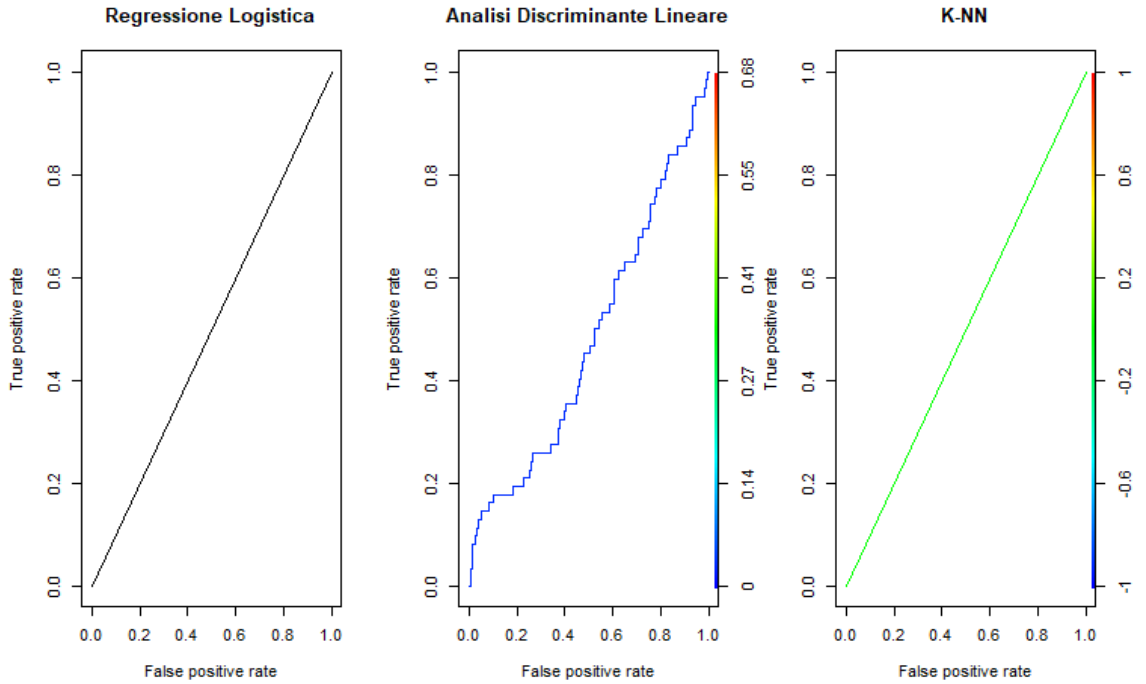
AUC = 0.74 AUC= 0.58 AUC=0.56

ROC curve per dataset ridotto utilizzando training bilanciato



AUC = 0.78 AUC= 0.77 AUC= 0.59

ROC curve per dataset ridotto con PCA



AUC=0.49 AUC=0.47 AUC=0.49

4 Discussioni

Nel presente elaborato ho affrontato un problema di classificazione relativo ad un dataset di dati e tassi finanziari appartenenti a 7027 società, applicando diversi metodi di data mining e diversi approcci di pre-processing. L'analisi esplorativa ha rilevato diversi problemi come la notevole presenza di missing values e outlier, e il problema di classi fortemente sbilanciate. Nonostante questo ho cercato soluzioni in modo che l'impatto generale sul dataset fosse minimo, e l'accuratezza dei metodi fosse alta. Osservando la Confusion Matrix osservo però che la specificità è molto bassa (considerando la classe "0" come positiva e "1" negativa) , poiché tende a stimare la maggior parte delle osservazioni come appartenenti alla classe "0", e solo poche vengono classificate in bancarotta. Questa problematica è visibile dalle curve ROC e dai valori AUC, ed è sicuramente legata al forte sbilanciamento delle classi (96% vs 4%).

Abbiamo visto che il metodo QDA in questo caso ha portato valori non soddisfacenti. In termini di accuracy il metodo migliore per tutti i training utilizzati pare essere il K-NN, ma presenta una percentuale di classe stimata fallimentare molto bassa. La regressione logistica e la LDA anch'esse presentano un'accuracy alta. E' difficile selezionare un metodo veramente accurato a causa dello sbilanciamento. Potrebbero esserci ulteriori metodi a me ancora non noti che siano in grado di classificare meglio il dataset.