Statlog - German credit
Report per l'Esame di Fondameti di Machine Learning

Alessandro Pansera

152543
Ing. Informatica
282423@studenti.unimore.it

Abstract

Il sistema bancario moderno si basa sulla cessione di prestiti ai propri correntisti al fine di realizzare profitto sugli interessi e fidelizzare il cliente. Alla base del modello di business descritto c'è la necessità di ponderare in maniera occulta l'affidabilità creditizia di un cliente affinché non risulti in futuro insolvente. Il Machine Learning permette di risolvere problemi di questa natura mediante la classificazione tramite la quale offrire un supporto al normale svolgimento delle mansioni di un operatore bancario.

1 Dataset

Il dataset è composto da dati bancari raccolti nel 1994 in Germania, è stato donato all'università di Amburgo e pubblicato nel 2019 all'interno dell'archivio dell'università della California. All'interno del dataset troviamo sia features categoriche che numeriche, non sono presenti sample con attributi nulli. Essendo il dataset, per la natura dei features che lo compongono, multivariato, occorre applicare delle tecniche di replacing per permettere il training di un classificatore. Nella sezione 1.1 verranno definiti i criteri con cui eseguire le operazioni di replacing. Non è necessario eseguire operazioni di normalizzazione o standardizzazione in quanto gli importi di natura maggiore si suppone, per la definizione del dominio delle features, non superi i 2 ordini di grandezza rispetto al valore minimo ammesso. per altre features.

1.1 Features

Di seguito è riportato l'elenco delle features presenti all'interno del dataset:

1. Stato del conto corrente (categorica)
    - A11 : $x < 0DM$
    - A12 : $0 \leq x < 200DM$
    - A13 : $x \geq 200DM$ / anticipo di uno stipendio annuale
    - A14 : sprovvisto di un c.c.
2. Durata in mesi del prestito (numerica)
3. Storia creditizia (categorica)
    - A30 : nessun prestito rate pagate puntualmente
    - A31 : tutti i prestiti, in questa banca, sono stati tutti ripagati puntualmente
    - A32 : i prestiti esistenti per ora sono stati pagati puntualmente

- A33 : ritardi nel ripagare prestiti esistenti
- A34 : pagatore 'sofferente' / ha altri debiti presso altre banche.

4. Obbiettivo (categorica)
    - A40 : macchina (nuova)
    - A41 : macchina (usata)
    - A42 : arredamento
    - A43 : TV
    - A44 : elettrodomestici
    - A45 : manutenzione
    - A46 : educazione
    - A47 : vacanza
    - A48 : formazione professionale
    - A49 : business
    - A410 : altro
5. Credito richiesto (numerica)
6. Conto di risparmio / azioni (categorica)
    - A61 : $x < 100DM$
    - A62 : $100 \leq x < 500DM$
    - A63 : $500 \leq x < 1000DM$
    - A64 : $x \geq 1000DM$
    - A65 : sconosciuto non esistente
7. Lavora da anni (categorica)
    - A71 : disoccupato
    - A72 : $x < 1anno$
    - A73 : $1 \leq x < 4anni$
    - A74 : $4 \leq x < 7anni$
    - A75 : $x > 7anni$
8. Rata in percentuale del reddito disponibile (numerica)
9. Status (categorica)
    - A91 : uomo: divorziato / separato
    - A92 : donna: divorziata / separata / sposata
    - A93 : uomo: single
    - A94 : uomo: sposato / vedovo
    - A95 : donna: single
10. Garanti (categorica)
    - A101 : Nessuno
    - A102 : Co-intestatari
    - A103 : Garante
11. Residente da anni (numerica)

12. Proprietà (categorica)
    - A121 : immobili
    - A122 : assicurazioni sulla vita / polizze
    - A123 : auto / altro
    - A124 : sconosciute / nessuna proprietà
13. Età (numerica)
14. Altre rate (categorica)
    - A141 : banca
    - A142 : negozi
    - A143 : nessuna
15. Abitazione (categorica)
    - A151 : affitto
    - A152 : proprietario
    - A153 : gratis
16. Credito esistente presso questa banca (numerica)
17. Lavoro (categorica)
    - A171 : disoccupato
    - A172 : impiegato / operaio non specializzato
    - A173 : impiegato / operaio specializzato
    - A174 : manager / autonomo / dip. altamente qualificato
18. Numero di persone da mantenere (numerica)
19. Telefono (categorica)
    - A191 : no
    - A192 : si, registrato con il nome del correntista
20. Lavoratore straniero (categorica)
    - A201 : si
    - A202 : no

Come è possibile vedere è un dataset non molto recente e lo si intuisce dalla moneta vigente in Germania, non c'è il marco tedesco.
Le 13 variabili categoriche verranno tutte rimpiazzate mediante un operazione di replacing basata sul codice presente per ogni categoria utilizzabile:

```
# original category
cat = "A201"
# replaced category
new_cat = int(cat[-2:])
```

Per consentire la modifica dinamica delle codifiche con cui eseguire il replacing è stato predisposto un sistema basato su file .json importabili i quali indicano per ogni feature le regole di sostituzione. L'obbiettivo è quello non di variare il codice ma di cambiare il .json importato per il replace rendendo il processo più rapido e meno soggtto a potenziali errori.

1.2 Var. Target

La variabile target presente all'interno del dataset ammette 2 valori interi che sono 1 ('Good borrower') e 2 ('Bad borrower'). Considerando la natura del problema si possono eseguire le predizioni mediante un modello di classificazione binaria. Non è necessario eseguire il replace per i valori della variabile target.

1.3 Considerazioni

Prima di scegliere il tipo di algoritmo da impiegare per la classificazione binaria occorre studiare con attenzione il dataset che si ha a disposizione. Il primo problema facilmente osservabile riside nel mancato bilanciamento delle classi all'interno del dataset. Il rapporto tra la classe 'Bad borrower' e 'Good borrower' è 1:3, per evitare problemi legati a predizioni 'positive' sbagliate occorre applicare delle tecniche di over-sampling o under-sampling che verranno elencate alla voce 2.1.
Per poter creare un algoritmo di Machine Learning realmente applicabile ad un contesto lavorativo è importante valutare anche aspetti indipendenti dalla pura natura dei dati. Il classificatore che si vuole realizzare deve essere in grado predirre con precisione 'Good borrower' e 'Bad borrower', a livello applicativo per un istituto di credito è più importante predirre con precisione i casi appartenente alla prima classe che alla seconda. Se infatti sbagliamo la predizione legata ad un 'Finto good borrower' rischiamo di incorrere in perdite economiche al contrario di un 'Finto bad borrower' per cui la perdita economica è solo ipotetica. Dal testo allegato al dataset è presente una matrice dei costi per cui un 'Finto good borrower' ha peso 5 mentre un 'Finto bad borrower' ha peso unitario.
Sarà necessario in fase di training andare allora a considerare le due problematiche appena citate per andare a realizzare un classificatore in grado di fornire delle buone performance.

2 Modello

Come già detto il problema di Machine Learning trattato è di classficazione binaria. Per realizzare un modello di classificazione binaria sono presenti svariate tecniche ma quelle implementate per la realizzazione del progetto sono la Logical Regression e il Random Forest che è un algoritmo di ensemble.
Se la Logical Regression ragiona in maniera simile alla Linear Regression, per cui fornita un singmoide, viene stimata la percentuale e, mediante una soglia, viene scelta la classe più adatta agli input forniti il Random Forest opera in maniera differente. L'idea alla base del Random Forest è quella di creare più modelli (alberi decisionali) scorrelati da confrontare per raggiungere una buona predizione. La tecnica con cui vengono generati gli alberi decisionali è detta 'bagging' e consiste nel prendere un set di dati iniziali e crearne di nuovi tramite campionamento casuale con rimpiazzo. Sui nuovi set di dati creati verranno eseguite delle predizioni e quella statisticamente più presente verrà assunta come corretta.

2.1 Pre-processing

Come detto in fase di descrizione del dataset sarà necessario eseguire delle operazioni di pre-processing sui dati oltre al replacing. All'interno del progetto sono state implementate diversi metodi:

- Under-sample: tecnica basata sul ridurre i samples appartenenti alla classe dominante.
  - Near miss: è un algoritmo di under-sampling basato sulla rimozione randomica dei sample appartenenti, per il nostro problema, alla classe 'Good borrower'.
- Over-sample: consiste nel generare fedelmente dei samples sulla base di assunzioni relative alla classe di appartenenza. Gli algoritmi implementati per il progetto appartengono alla famiglia dei 'Synthetic Minority Oversampling Technique' . Gli algoritmi SMOTE non operano semplicemente duplicando i dati ma selezionano i samples più simili basandosi

sui features, e ne generano di nuovi sull'interpolazione che connette a due a due i samples selezionati nello spazio multidimensionale.

- K SMOTE: attraverso l'algoritmo K-means vengono generati dei cluster all'interno del quale vengono selezionati i sample che verranno usati come base per l'over-sampling.Successivamente vengono scartati i cluster in cui abbiamo una sproporzione tra sample appartenenti alla classe maggioritaria e minoritaria. Per i cluster rimanenti applico l'algoritmo SMOTE base basato sulla ricerca dei K-Neighbour. K-means opera creando un numero K di cluster per i quali cerca un 'centroide' in modo iterativo e in seguito collega ad ogni cluster i sample sulla base di un principio di vicinanza.
- SVM SMOTE: l'algoritmo è un successore di SMOTE e in questo caso genera dei dati sintetici sul confine tra due classi, confine rilevato grazie ad SVM.
- ADASYN: il principio di funzionamento è differente dai precedenti due in quanto vengono creati dei dati sintetici negli spazi multidimensionali in cui la classe minoritaria è poco densa.

2.2 Performance

Di seguito sono riportate tutte i punteggi per ciascun classificatore per ciasuna tecnica impiegata in pre-processing.
Alla base di ciascuna metrica è presente una confusion matrix, le colonne rappresentano i valori reali e le righe le predizioni.

- Logical Regression - Near Miss

	Val. Reali		
		1	2
Pred.	1	**36**	32
	2	7	**25**

F1 score	0.64
F.D.R.	0.47
Precision	0.52
Recall	0.84

- Logical Regression - SVM SMOTE

	Val. Reali		
		1	2
Pred.	1	**54**	14
	2	10	**22**

F1 score	0.82
F.D.R.	0.21
Precision	0.84
Recall	0.79

- Logical Regression - K SMOTE

	Val. Reali		
		1	2
Pred.	1	**57**	11
	2	12	**20**

F1 score	0.83
F.D.R.	0.16
Precision	0.84
Recall	0.83

- Logical Regression - ADASYN

Val. Reali				
		1	2	
Pred.	1	**53**	15	
	2	9	**23**	

F1 score	0.81
F.D.R.	0.22
Precision	0.78
Recall	0.85

- Random Forest - Near Miss

Val. Reali				
		1	2	
Pred.	1	**51**	17	
	2	10	**22**	

F1 score	0.79
F.D.R.	0.25
Precision	0.75
Recall	0.84

- Random Forest - SVM SMOTE

Val. Reali				
		1	2	
Pred.	1	**59**	9	
	2	13	**19**	

F1 score	0.84
F.D.R.	0.13
Precision	0.87
Recall	0.82

- Random Forest - K SMOTE

Val. Reali				
		1	2	
Pred.	1	**63**	5	
	2	21	**11**	

F1 score	0.83
F.D.R.	0.07
Precision	0.92
Recall	0.75

- Random Forest - ADASYN

Val. Reali				
		1	2	
Pred.	1	**57**	11	
	2	13	**19**	

F1 score	0.83
F.D.R.	0.14
Precision	0.84
Recall	0.81

2.3 Analisi performance

Come già asserito alla voce 1.3 è altamente vincolante la precisione con cui vengono predetti i 'Good borrower'. Di seguito con FP verranno indicati i 'Finti good borrower' e con TP i 'True good borrower'. All'interno delle Confusion Matrix riportate il risultato desiderato si traduce in un gap il più elevato possibile tra gli elementi all'interno della prima riga. E' comunque importante avere una buona precisione in fase di predizione dei TN altrimenti il modello a livello generale perderebbe di precisione ed è comunque un risultato non accettabile.
Fissato l'obbiettivo che si vuole raggiungere si può procedere allo studio di quelli che sono gli score a cui si è maggiormente interessati per lo studio della bontà del modello. Se l'F1 score è più un indicatore generico che è sempre buona norma valutare per stimare la bontà generale di un classificatore F.D.R., Precision e Recall sono specifici per il tipo di problema in questione. L'F.D.R., 'False discovery rate', permette di valutare il rapporto tra FP e TP; la metrica è perciò una delle principali da considerare in quanto valuta esattamente quello che è il principale vincolo del problema di classificazione in questione. Precision e Recall misurano entrambe la qualità con cui vengono rilevati i TP e FP.

$$F1score = \frac{2 * precision * recall}{precision + recall}$$

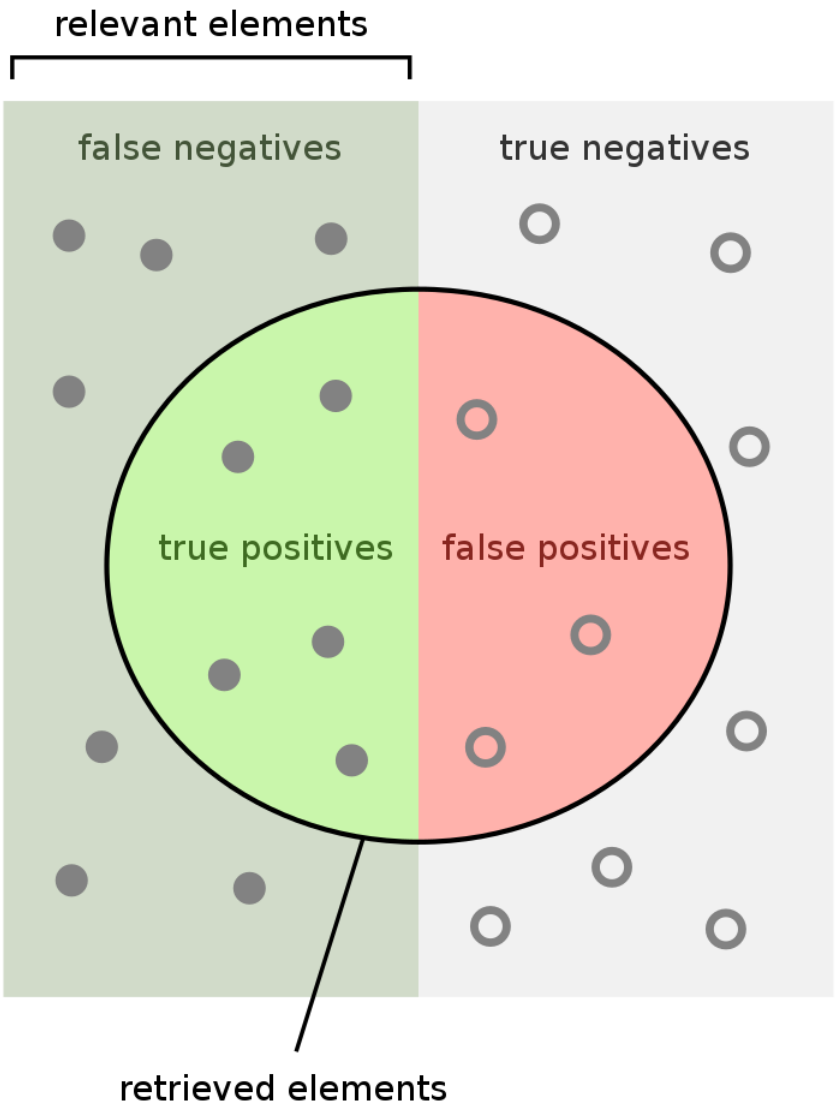
$$F.D.R. = \frac{FP}{TP + FP}$$

Se la Precision misura la precisione delle predizioni positive invece la Recall ne misura la completezza, di seguito sono riportate le formule di calcolo.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{FP}{TP + FN}$$

come verrà spiegato in seguito Precision e Recall sono inversamente proporzionali ed è possibile osservarlo anche dai test eseguiti sul classificatore. Se infatti si confrontano i risultati prodotti con il Random Forest applicando la tecnica di K SMOTE e SVM SMOTE per l'over-sampling sarà possibile vedere un andamento inversamente proporzionale; se con il K SMOTE Precision è più alta rispetto a SVM SMOTE invece accade l'opposto con il Recall.
L'inversa proporzionalità è legata al denominatore per cui se salgono gli FP decrescono i FN per quella che è la natura della classificazione binaria. Più si è precisi a definire ciò che si cerca minore è la completezza delle predizioni positive. La Precision può essere vista come una misura di esattezza mentre la Recall come una misura di completezza.



How many retrieved items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{retrieved elements}}$$

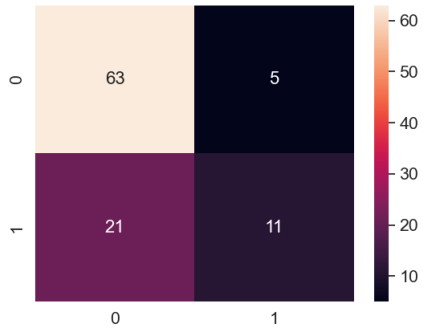
How many relevant items are retrieved?

$$\text{Recall} = \frac{\text{true positives}}{\text{relevant elements}}$$

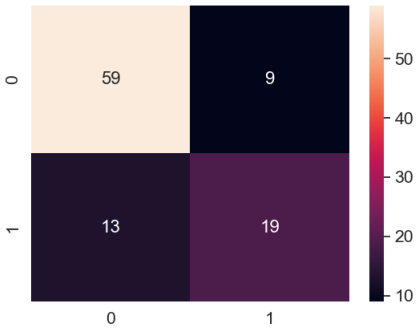
Per la tipologia di problema in questione, considerando soprattutto il settore in cui si opera, si predilige una buona Precision a discapito di un miglior Recall.
Le soluzioni migliori per la tipologia di problema considerato sono raggiunte applicando l'algoritmo di ensemble Random Forest applicando o il K SMOTE o l'SVM SMOTE. Le due strade appena citate presentano due differenze:

- FDR
- Precision/Recall

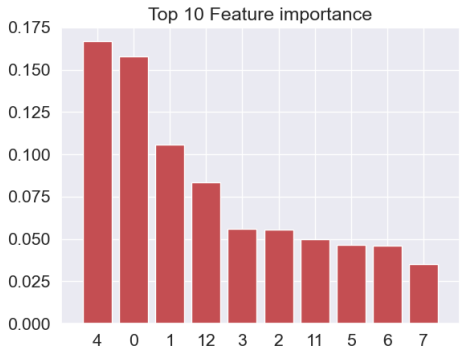
Le due voci appena citate servono per valutare il giusto tradeoff per la soluzione che si vuole applicare. K SMOTE permette di raggiungere l'FDR minimo registrato tra tutte le tecniche portando una Precision molto elevata a discapito della recall. SVM SMOTE mostra un comportamento più bilanciato con Precision e FDR rispettivamente peggiori di 5 pt. percentuali. L'upgrade legato all'applicazione del SVM SMOTE è riconducibile ad un incremento del Recall e alle solide performance dell'algoritmo che riporta una Precision maggiore di 5 pt. percentuali rispetto a quanto indicato all'interno della piattaforma che rende fruibile il dataset. Per concludere se l'over-sampling con K SMOTE permette di avere un estremizzazione del prinicpio per cui si vogliono evitare in maniera assoluta i falsi positivi SVM SMOTE offre una maggiore correttezza in fase di predizione offrendo una soluzione più bilanciata e, nonostante tutto, fedele.



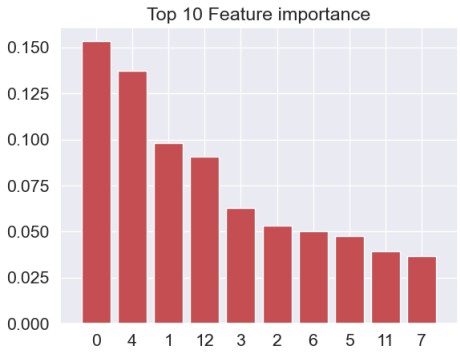
(a) Confusion Matrix - K SMOTE



(b) Confusione Matrix - SVM SMOTE



(a) Feature importance - K SMOTE



(b) Feature importance - SVM SMOTE

2.4 Implementazione

Il modello di classificazione può essere addestrato attraverso differenti combinazioni tra algoritmi e tecniche di unser-sampling/over-sampling. Per permettere un rapido interscambio tra tecniche e algoritmi usati è stato implementato, tramite incapsulamento, il Factory Pattern che permetta di pilotare tramite 2 stringhe tutti i fattori che portano all'addestramento del modello. E' stato inoltre prevista la possibilità di salvare il classificatore affinchè sia utilizzabile in seguito al training. Per rendere agevole il testing dell'algoritmo è stata creata una piattaforma web che permette l'input dei feature values impiegati dal classificatore. I dati provenienti dal web sono stati inseriti all'interno di una cartella apposita che, in un ottica applicativa, potrebbe permettere di raccogliere ulteriori samples ri-utilizzaibili, post correzione, per ulteriori re-training.
L'algoritmo Random Forest permette di definire dei pesi da associare alle classi; verranno applicati i pesi 1 per la classe 'Good borrower' e 2 per la classe 'Bad borrower'. A seguito di numerosi training la scelta si è dimostrata vincente permettendo di migliorare notevolmente tutte e 4 le voci con cui valutare i classificatori prodotti attraverso differenti tecniche. All'interno dello script che esegue il training è presente la rilevazione di esecuzione delle singole operazioni che portano al risultato finale con annessa scrittura all'interno di file di log.
Il download e la normalizzazione del dataset sono definiti in apposite classi che favoriscono l'intercambiabilità delle operazioni senza dover ristrutturare l'interno codice. L'obbiettivo è evidente sopratutto all'interno dei processi che gestiscono la normalizzazione in quanto è stata definita una struttura all'interno di file JSON richiamabili tramite i quali verranno definite tutte le potenziali operazioni di Replacing impiegate sul dataset.