SUS 4

team: SUDATA Alfonso D'Amelio & Valerio Guarrasi

20/6/2018

Data Pre-processing

Una volta importati i dati (riguardanti un problema di decisione sull'erogazione dei finanziamenti) che l'azienda *Findomestic* ci ha messo a disposizione, abbiamo prima di tutto applicato le seguenti tecniche di *Data-preprocessing*:

1. Divisione della variabile target (y) dalle variabili dipendenti (X).
2. Dalla X abbiamo innanzitutto eliminato (drop) la variabile 'identificatore'(id) non utile ai fine della classificazione.
3. Osservando che ci sono valori mancanti nelle seguenti variabili:

- *regione*
- *anz_ban*

Abbiamo deciso di imputare questi missing data (nan) mediante una *media*.

4. *'Encoding categorical data'*, ovvero per tutte le variabili qualitative abbiamo trasformato da stringa a intero.
5. Osservando che ci fossero pochi records con target uguale a 'CONTENZIOSO' e 'RECUPERO' abbiamo applicato un algoritmo di OVER-SAMPLING:

I cosiddetti "Unbalanced data" si riferiscono in genere a problemi di classificazione in cui le classi non sono rappresentate allo stesso modo.

Infatti nella maggior parte dei dataset riguardanti '*classificazione*' non si ha esattamente un numero uguale di istanze in ogni classe.

DATASET che hanno questo problema sono ad esempio quelli riguardanti transazioni fraudolente, dove la stragrande maggioranza delle transazioni sarà nella classe "Non-Fraud" e una piccola minoranza sarà nella classe "Fraud", oppure casi di set di dati come questi, nei quali la maggior parte dei consumatori risulta essere un cliente 'REGOLARE'.

È possibile quindi modificare il dataset che si utilizza per avere dati più bilanciati. Questo approccio è chiamato campionamento del set di dati, e ci sono due metodi principali che è possibile utilizzare per uniformare le classi:

- aggiungere copie di istanze dalla classe sottorappresentata denominata sovracampionamento
- eliminare le istanze dalla classe sovrarappresentata, chiamata sottocampionamento.

Come precedentemente detto, noi abbiamo deciso di implementare il secondo metodo, mediante l'utilizzo di un algoritmo di OVER-SAMPLING chiamato **SMOTE** (Synthetic Minority Over-sampling Technique), in modo tale che il numero di record per ogni tipo di target (*regolare,contenzioso,recupero*) fosse uguale.

6. Con *OneHotEncoder* creiamo dummy variables per le variabili qualitative nella X.
7. A questo punto abbiamo splittato il dataset in $X_{train}, X_{test}, y_{train}, y_{test}$ con ratio uguale a 0.8 e 0.2.
8. Abbiamo applicato un "feature scaling" per avere tutte le variabile nella stessa scala.

Fitting model

Per scegliere l'algoritmo di classificazione migliore abbiamo provato (con i parametri di default):

- *Logistic Regression*

- *K-nn*
- *SVM*
- *Kernel SVM*
- *Naive Bayes*
- *Decision Tree*
- *Random Forest*
- *XGBoost*
- *Gradient Boosting*

L'ultimo è sembrato essere il migliore.

Quindi per settare al meglio i parametri abbiamo applicato un Grid Search su:

- loss, learning rate, n_estimators, max_depth.

Una volta "fiatato" il modello, prima ancora di fare la predizione sul **TEST**, abbiamo applicato i seguenti passaggi al X_{TEST}:

- Eliminare le variabili identificatrici
- Eliminare i valori nulli (nan)
- Trasformare le stringhe in interi
- Creare le dummy variables
- Feature scaling.

A questo punto abbiamo effettuate la predizione e creato il file txt come richiesto, ottenendo il seguente punteggio:

- punteggio 13.06%