MOBD
relazione di progetto

Christian Santapaola
0294464
Università di Roma Tor Vergata
Roma, Italia

***Abstract*—Questa relazione si pone l'obiettivo di documentare e razionalizzare le scelte effettuate durante il progetto di MOBD.**

I. Introduzione

Il dataset su cui è stato effettuato il modello di predizione è composto da 14 features è una classe binaria. Il primo obiettivo è stato di studiare la struttura del dataset, in particolare di partizionare le features in categoriche e non categoriche, di studiare la presenza di valori nulli e la loro gestione, si è studiato come gestire la presenza di sbilanciamento nella classe che ammonta a 0.75 per la classe 0, 0.25 per la classe 1 tramite tecniche di sampling, infine si è cercato il classificatore più efficace a classificare questo dataset, questo ciclo è stato ripetuto in un feedback loop finché non si è arrivato ad un risultato soddisfacente.

II. Analisi delle Features

La prima partizione fatta è stata suddividere le features tra features categoriche e non categoriche, definiamo una features categorica, come una features il cui l'insieme dei valori che può assumere ha dimensione finita. Una assunzione importante fatta in questa fase per le features categoriche e che tutti i valori che la features può assumere siano presenti nel dataset di addestramento. Per identificare le features come categoriche si è quindi analizzato il numero di valori che una features ha assunto. Le features F1, F3, F4, F5, F6, F7, F8, F9, F13 sono state considerate categoriche, tra queste possiamo effettuare una seconda partizione, tra features numeriche e features non numeriche, nelle prime rientra la sola F4, la quale contiene tutti i valori interi compresi tra 1 e 16. La features F0, nonostante sia simile ad F4, non è stata considerata come una features categorica per due motivi, il primo il numero di valori molto più grande di F4, il secondo è dovuto al tipo di dato, F0 nel contesto del dataset, ha la struttura di un eta, la quale è stato preferito considerarla una variabile quantitativa, questo permette ad un modello di mantenere il concetto di distanza tra i valori, cosi che una persona di 80 anni, risulti più vicina ad una di 70 rispetto ad una di 17, questa relazione nel modello categorico andrebbe persa. Sulle features categoriche non numeriche, le features sono state a loro volta partizionate in Nominali e Ordinali, una features Nominale è una features categorica dove i suoi dati non hanno una relazione logica, mentre una features Ordinale è una features categorica dove i suoi dati seguono una relazione logica. Le features F8, F9, F13 sono state considerate come features nominali, mentre le restanti features F1, F3,F4,F5,F6 e F7 sono state considerate features ordinali, la relazione logica è stata assunta dalla forma che queste features prendono con valori del tipo 'lettera numero' che permette un qualche ordine ragionevole di essere scelto per le features ad esempio che il valore 'Q1' sia minore di 'Q2', questo risulta non possibile per le features nominali, tra 'USA' e 'Italy' non è possibile asserire alcuna relazione ragionevole sui dati. Le features non categoriche sono tutte features numeriche, F0,F2,F10,F11,F12. La seconda analisi effettuata comprende la presenza di valori nulli, tra le features presenti otteniamo il seguente risultato:

TABLE I
Valori nulli nel dataset per features

Feature	valori nulli
F1	1836
F6	1843
F13	583
Totale	2399

Dalla tabella possiamo vedere come il numero di righe con valori nulli non è trascurabile, il totale ammonta a quasi il 7% delle righe totali del dataset. Le terza analisi è il bilanciamento del dataset, contanto il numero di istanze con classe 0 e classe 1 otteniamo i seguenti valori:

TABLE II
Tabella di bilanciamento delle classi

classe	istanze	percentuale
0	24720	0.75919044
1	7841	0.24080956

Come visto dalla tabella possiamo osservare che il dataset è sbilanciato, quindi dovremmo cercare di mitigare questo sbilanciamento con un algoritmo di sampling per ottenere un modello di predizione più accurato.

III. Metodo di valutazione modello di predizione

Per valutare i modelli di predizione si è scelto una strategia di cross validation basata su kfold a 5 iterazioni, stratificata e con le righe mischiate fra loro, i parametri scelti per misurare l'efficacia di un modello sono stati l'accuracy, precision, recall ed f1. Questa strategia è stata favorita rispetto ad una

strategia a 10 iterazione per un incremento nelle performance, è stato provata per un sottoinsieme dei classificatori testati la strategia a 10 iterazioni, ma ha richiesto un aumento nel tempo di esecuzione è non ha dato risultati più accurati per giustificare tale aumento. La strategia effettuata per trovare il modello migliore è stata di provare in modo iterativo vari tipi di classificatore, e di tecniche di features engineering che potessero aiutare il classificatore a migliorare la sua efficacia, ogni loop di queste prove è stato accompagnato con i risultati dei loop precedenti allo scopo di migliorare la predizione, infine la migliore combinazione è stata scelta.

IV. Gestione dei valori nulli

Come detto in precedenza il dataset contiene un valore non trascurabile di valore nulli, le features coinvolte hanno in comune di essere features categoriche non numeriche, la strategia preferita è stata una strategia di imputing basata sulla moda, che sostituisce il valore mancante con la moda della features.

V. Gestione dei valori categorici

Per i valori categorici nella nostra analisi abbiamo distinto tra features categoriche Nominali e Ordinali, le features nominali F8, F9 e F10 sono state trasformate tramite la tecnica di Normalizzazione chiamata One Hot Encoding, mentre le features Ordinali sono state trasformate mappando l'insieme delle valori delle features con una serie ordinata di interi, tramite l'algoritmo di OrdinalEncoding di sklearn. Il motivo di questa sceltà è stato che l'algoritmo di One Hot Encoding aumenta il numero delle features nel dataset, quindi applicarlo ciecamente su tutte le features categoriche rischia di farci esplodere la dimensione delle features e di darci risultati peggiori, quindi dove è stato possibile l'algoritmo di OrdinalEncoding permette di trasformare le features categoriche ordinali in un formato leggibile dal modello senza far esplodere il numero di features, questo metodo può causare delle problematiche che l'algoritmo di One Hot Encoding non causa, dovuto principalmente al caso in cui delle righe del dataset diventino con questa trasformazione linearmente dipendenti tra loro che può portare alcuni algoritmi a non terminare, ma nel nostro caso questo non è problema non è stato riscontrato, tuttavia applicare ciecamente a tutte le features categoriche One Hot Encoding causa problema per alcuni classificatori tipo SVC() il cui tempo di addestramento ha complessita approssimabile a $O(n\_samples^2 \cdot n\_features)$, si traduce in lunghi tempi di addestramento.

VI. Normalizzazione e standardizzazione dei dati

Le features non categoriche sono state provate a normalizzare con l'algoritmo di sklearn MinMaxScaler(), standardizzare con l'algoritmo StandardScaler(), o non applicare niente, in un loop trial and error, alcuni classificatori sono molto sensibili ai range tra i valori di diverse features, altri molto meno, questa sensibilità implica la possibilità di aumentare l'efficienza e l'accuratezza dei classificatori andando a trasformare le sue features.

VII. Risultati dei classificatori

I classificatori testati sono stati SVC, ADABoostClassifier, e GradientBoostingClassifier. I risultati di accuracy sono stati i seguenti:

TABLE III
Risultati di accuracy dei classificatori

	SVC	RandomForest	ADABoost	GradientBoosting
UnderSample	0.797	0.814	0.814	0.828
OverSample	0.796	0.852	0.814	0.840
SMOTE	0.797	0.852	0.843	0.871

Il valore migliore lo si è ottenuto dal classificatore GradientBoostingClassifier unito alla tecnica di sampling SMOTE e senza alcuna tecnica di standardizzazione o normalizzazione sui dati non categorici, con un accuracy di 0.871. Un secondo valore tenuto in considerazione è stata il valore f1, che per il miglior classificatore ha un valore di 0.714. Un valore di F1 basso con un accuracy alta, può implicare che il classificatore non stia in realtà classificando, ma stia assegnando l'etichetta della classe maggioritaria alla maggior parte delle righe del dataset di prova, quindi il valore di accuracy risulta alto a causa del numero maggiore di istanze della classe maggioritaria che sono state classificate correttamente, nel nostro caso il valore è stato ritenuto buono in relazione al valore di accuracy ottenuto e dal benchmark richiesto.