



Classification

Stroke Dataset

Report progetto

Laboratorio di Big Data

Classificazione sul dataset stroke



A.a. 2022/2023

*Corso di studi: Data Science, Business Analytics e
Innovazione*

Prof. Giuliano Armano

Alessandro Piroddi 11/82/00205



Le librerie

INTRODUCTION

Il dataset oggetto di analisi, disponibile al seguente [link](#), contiene le informazioni relative all'insorgenza di ictus nei pazienti, che secondo l'Organizzazione della Sanità Mondiale (OMS) risulta essere la seconda causa di morte a livello globale, responsabile di circa l'11% dei decessi totali.

Con tale analisi si ha l'obiettivo di applicare diversi modelli statistici al fine di classificare la variabile di risposta "**stroke**" in funzione delle caratteristiche dei pazienti, che verranno trattate come variabili indipendenti.

LOADING LIBRARIES AND FILES

La prima operazione concerne l'importazione delle varie librerie utili all'esecuzione delle funzioni applicate nel corso della stesura del codice. Le varie librerie sono state suddivise, attraverso l'utilizzo dei commenti, in base alla loro funzione.

Ci si è avvalsi della libreria **pyspark.sql** per il caricamento, la cancellazione e la manipolazione dei dati e per l'inizializzazione dello stesso spark. I moduli di **pyspark.ml**, relativi al machine learning, sono stati utilizzati invece per la gestione dei dati nella fase di pre-processing e per la creazione dei vari modelli di classificazione.

Per la parte di visualizzazione grafica, e più in generale dell'analisi esplorativa dei dati, sono state utilizzate le librerie di **matplotlib**, **seaborn** e **plotly**.

DATASET OVERVIEW

Dopo aver inizializzato spark e aver opportunamente caricato il dataset , si è proceduto ad effettuare una panoramica generale delle osservazioni che vanno a comporre il dataset.

Il dataset è composto da 5110 osservazioni e da 12 colonne relative alle caratteristiche dei pazienti che hanno o non hanno avuto un ictus. Le colonne sono le seguenti:

- ***id*** : codice identificativo paziente
- ***gender*** : maschio/femmina/altro
- ***age*** : rappresentante l'età del paziente
- ***hypertension*** : "1" il paziente soffre di ipertensione, "0" non soffre di ipertensione
- ***heart_disease*** : "1" il paziente ha problemi cardiopatici, "0" non soffre di cardiopatia
- ***work_type*** : identifica la tipologia lavorativa del paziente
- ***residence_type*** : rurale/urbana
- ***avg_glucose_level*** : rappresenta il valore medio di glucosio nel sangue
- ***bmi***: relativo all'indice di massa corporea
- ***smoking_status***: identifica il tipo di fumatore
- ***stroke***: "1" il paziente ha avuto un ictus, "0" non ha avuto un ictus

Attraverso la funzione ***printSchema()*** è stato possibile identificare il tipo e la determinazione di ogni variabile. Il Dataset risulta essere composto prevalentemente da variabili categoriche, mentre le variabili quantitative sono solo 4: "***id***", "***age***", "***avg_glucose_level***", "***bmi***".

La funzione ***describe()*** ha permesso di ottenerne una sintesi, contenente il valore minimo, medio e massimo per ogni variabile. In questo modo è stato possibile individuare eventuali valori errati. E' il caso dell'indice di massa corporea, il quale ha presentato valori massimi "N/A", cioè la presenza di osservazioni mancanti.

DATA CLEANING & MANIPULATION

Successivamente alla ridenominazione di alcune colonne per risolvere eventuali conflitti di case sensitive in fase di compilazione del codice e al *casting* delle variabili quantitative, sono state svolte le operazioni di pulizia e mantenimento dei dati. Nello specifico, per l'indice di massa corporea, sono state sostituite 201 osservazioni mancanti con il valore medio, calcolato sul totale delle osservazioni della variabile stessa.

Nel caso della variabile “**smoking status**”, che presentava valori “*unknown*”, non è stato possibile ricostruire il dato per mancanza di informazioni da cui attingere nelle altre righe. Inoltre, avendo “*unknown*” un peso pari al 30% delle osservazioni totali, e considerato il trade off tra informazioni mancanti e restanti osservazioni sulla stessa riga, si è optato per il mantenimento immutato della variabile.

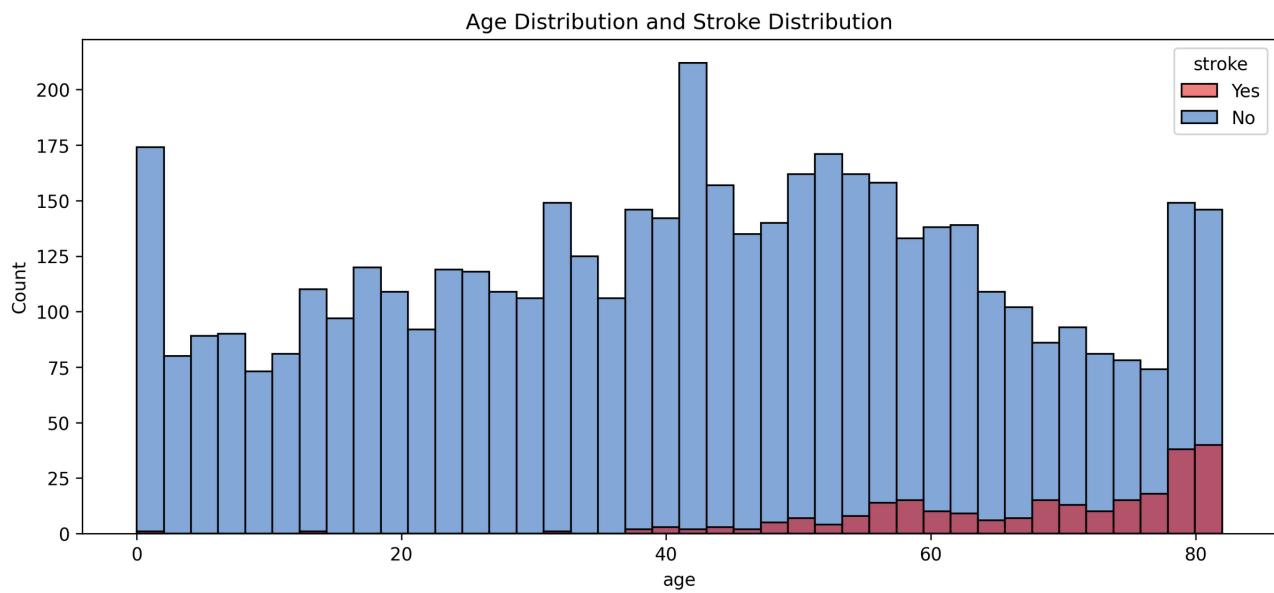
Discorso differente può essere fatto nel caso della variabile “**gender**”, formata da tre livelli “*male*”, “*female*” e “*other*”. In questo caso essendoci un solo valore “*other*” si è deciso di eliminarlo, passando così a 5109 osservazioni totali.

EXPLORATORY DATA ANALYSIS & OUTLIERS DETECTION

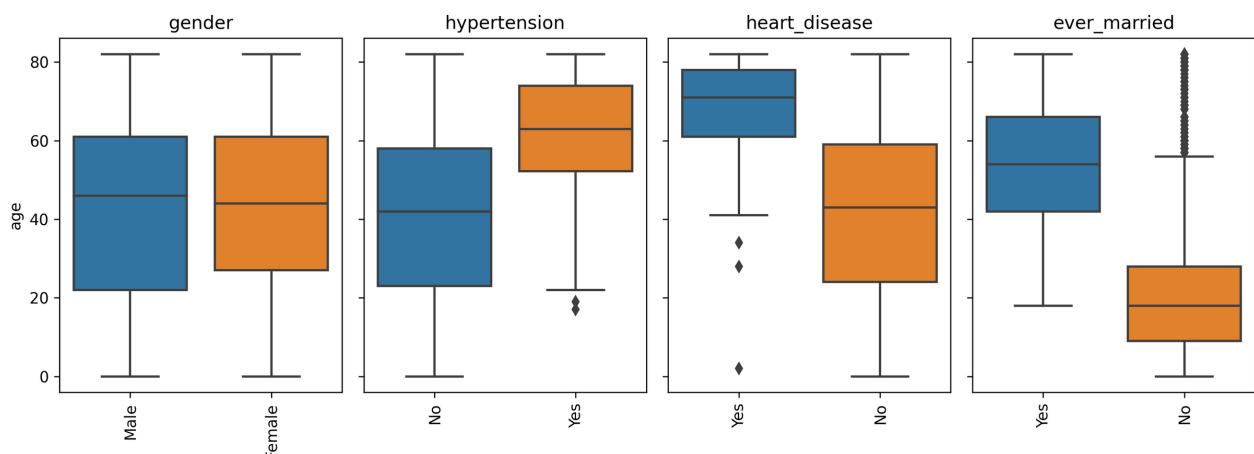
Dopo aver adeguatamente ripulito il Dataset, si è passati a una descrizione delle caratteristiche dello stesso attraverso un'analisi grafica delle variabili, che ne permettesse una comprensione più facile e immediata. A tal fine sono stati predisposti un insieme di Grafici a ciambella, utili per la descrizione delle singole variabili qualitative, e gli Istogrammi/Boxplot per le variabili quantitative, con lo scopo di evidenziare la propria distribuzione e possibili valori anomali.

I Grafici a ciambella hanno permesso di identificare immediatamente le varie classi delle variabili e attraverso il calcolo percentuale la loro suddivisione. Osservando il grafico a ciambella relativo alla variabile target “**stroke**” è possibile notare come questa sia fortemente sbilanciata verso valori di “*no_stroke*” pari al 95,13%, mentre solo il 4,87% è relativo ai casi in cui il paziente ha avuto un ictus.

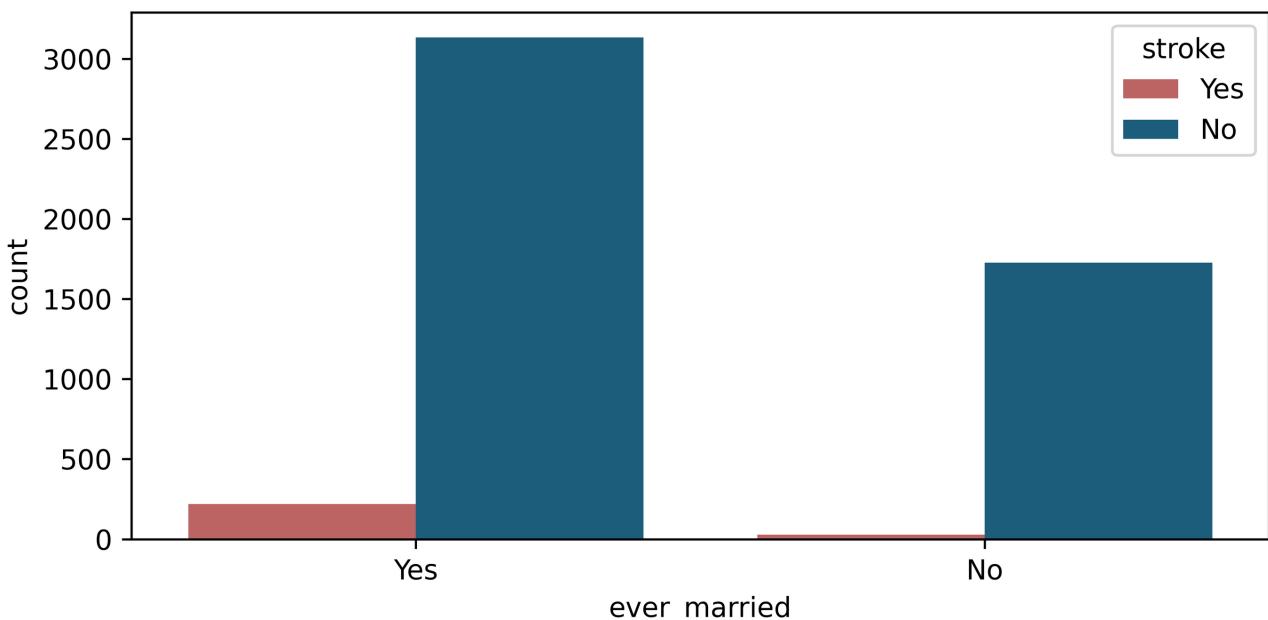
Tale situazione porterà, come verrà specificato più avanti, all'applicazione della tecnica di *Resampling* per bilanciare così la variabile di risposta oggetto dell'analisi.



Per quanto concerne i grafici bivariati invece, risulta particolarmente interessante il confronto tra le distribuzioni sovrapposte delle variabili "*age*" e "*stroke*". Si può notare come i casi in cui si verificano gli ictus aumentino con l'avanzare dell'età del paziente, con frequenza maggiore nei casi che vanno dai 60 anni fino agli 80. Il comportamento risulta essere analogo anche nei casi di ipertensione e cardiopatia, come segnalato dai boxplot correlati all'età rappresentati di seguito.



Degno di attenzione è inoltre il grafico a barre relativo al confronto tra il predittore "*ever_married*" e la variabile target "*stroke*". La maggior parte dei pazienti che ha avuto un ictus è stata sposata almeno una volta, mentre gli ictus risultano essere pochissimi nei casi di pazienti mai sposati.



L'ultimo passo dell'analisi esplorativa riguarda l'individuazione e la gestione di eventuali outlier, ovvero valori anomali presenti nelle variabili. Per tale scopo è stata creata una funzione `find_outliers()` che individua, per ciascuna variabile passata alla funzione, gli outlier basandosi sull' Interquartile Range Rule, ovvero sul calcolo, per ogni variabile, dell'Interquartile Range (IQR). L'IQR di una variabile è la differenza tra il terzo quartile e il primo quartile di quella stessa variabile.

Una volta calcolato l'IQR questo va moltiplicato per 1,5 ed il prodotto ottenuto va:

- sommato al terzo quartile Q3 ottenendo così la soglia superiore, per la quale ogni valore che supera questa soglia viene considerato un outlier;
- sottratto al primo quartile Q1 ottenendo così la soglia inferiore, per la quale ogni valore inferiore ad essa viene considerato un outlier.

L'eliminazione degli outlier avviene soltanto se questi non eccedono il 3% del totale delle osservazioni del dataset.

Attraverso la funzione sono state così ripulite dagli outliers le variabili "`age`" e "`bmi`" mentre la variabile "`avg_glucose_lv`" è rimasta immutata.

DATA PRE-PROCESSING

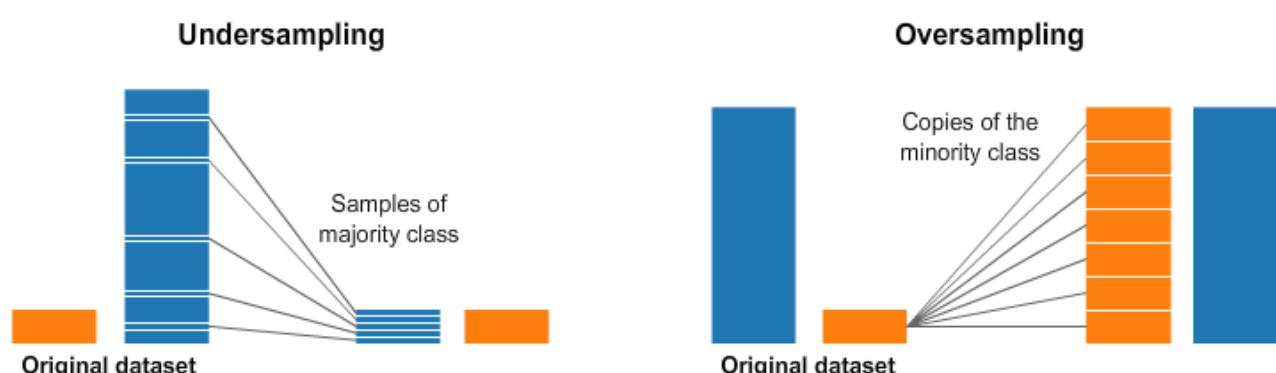
Successivamente all'analisi grafica e alla rimozione degli outliers più significativi, si è proceduto all' Encoding delle variabili qualitative e all'applicazione di metodi di Scaling per le variabili quantitative.

Nel caso delle variabili categoriche, si è applicato l'algoritmo di *string indexing*, passando in seguito quest'ultime al *One Hot Encoding*, algoritmo che risulta particolarmente adatto per la gestione delle variabili nominali non ordinate.

Per le variabili quantitative invece, essendo state gestite in maniera differente dalla funzione `find_outliers()`, sono state utilizzate due tipologie di Scaler:

- Per le variabili "**bmi**" e "**age**" si è optato per l'utilizzo del *MinMax Scaler*. Questo scaler risulta decisamente sensibile agli outliers, pertanto deve essere utilizzato in quei contesti in cui si voglia scalare una variabile già ripulita dai suoi valori anomali.
- Per la variabile "**avg_glucose_lv**" invece, è stato utilizzato l'algoritmo RobustScaler. Il motivo di tale scelta è legato al fatto che il *RobustScaler* permette di gestire la presenza degli outlier nella variabile che si sta scalando, e si utilizza appunto quando si hanno outlier nei nostri dati.

Come accennato nel paragrafo precedente la variabile di risposta "**stroke**" risultava altamente sbilanciata con percentuali legate ai valori binari 0 e 1 rispettivamente del 95% e del 5%. Nei casi come questo potrebbe risultare utile l'applicazione delle tecniche di *Resampling* con lo scopo di bilanciare le due classi.



Tra le varie tecniche di ricampionamento sono state prese in considerazione le tecniche di *undersampling* e *oversampling*. Nel primo caso, lo scopo di tale metodologia risulta essere quello di selezionare un sottoinsieme di osservazioni appartenenti alla classe maggioritaria fino al raggiungimento della proporzione di equilibrio rispetto la classe minoritaria. Nel caso dell'oversampling invece, le osservazioni della classe minoritaria sono duplicate fino al raggiungimento, in termini di quantitativi, della classe più popolosa.

E' necessario tenere in considerazione che entrambe le tecniche si discostano dalla realtà osservata poiché vanno a creare dei dati simulati, aumentando così il *bias statistico*, ovvero il pregiudizio dato dalla differenza tra valori teorici e reali. Per tale motivo l'applicazione di queste tecniche deve essere effettuata successivamente alla suddivisione del dataset ed esclusivamente sul set di addestramento, lasciando il test set immutato da osservazioni duplicate.

Nel caso analizzato, visto lo scarso numero di osservazioni appartenenti alla classe minoritaria, si è deciso di optare per l'applicazione della tecnica di oversampling, che ha il vantaggio rispetto l'undersampling, di mantenere tutte le osservazioni della classe maggioritaria.

CLASSIFICATION MODELS

Successivamente alla suddivisione del dataframe in training/test set e all'applicazione della tecnica di Oversampling sul set di addestramento si è proceduto all'applicazione degli algoritmi di Machine Learning per la classificazione dei pazienti colpiti da ictus.

Si è deciso di utilizzare sei differenti algoritmi di classificazione, di seguito elencati:

- *Logistic Regression*
- *Decision Tree*
- *Random Forest*
- *Naive Bayes*
- *Gradient Boosted Tree*
- *Super Vector Machines*

L'adozione della Regressione Logistica come primo modello non è stata un caso: infatti la Logistica, nonostante fornisca dei risultanti meno performanti rispetto gli altri modelli, ha il vantaggio di essere un modello "semplice" e di conseguenza i risultati ad esso associati risulteranno facilmente interpretabili.

In casi come la Random Forest invece, la funzione associata alla variabile di risposta è trattata come una **black box** e i risultati, derivanti dalle associazioni casuali, devono essere ritenuti attendibili a priori senza possibilità di essere interpretati.

ANALYSIS RESULTS

In presenza di dataset con variabili di risposta sbilanciate è opportuno prestare particolare attenzione ai risultati ottenuti e non cadere nella "trappola" dell'accuratezza elevata derivante dai vari modelli.

Un modello di classificazione applicato alla variabile target sbilanciata tenderà infatti ad avere un bias verso i dati più numerosi compromettendo così l'accuratezza. In casi di sbilanciamento delle classi è consigliabile tenere in considerazione differenti metriche come la Precision e Recall, che da un punto di vista statistico risulteranno maggiormente attendibili.

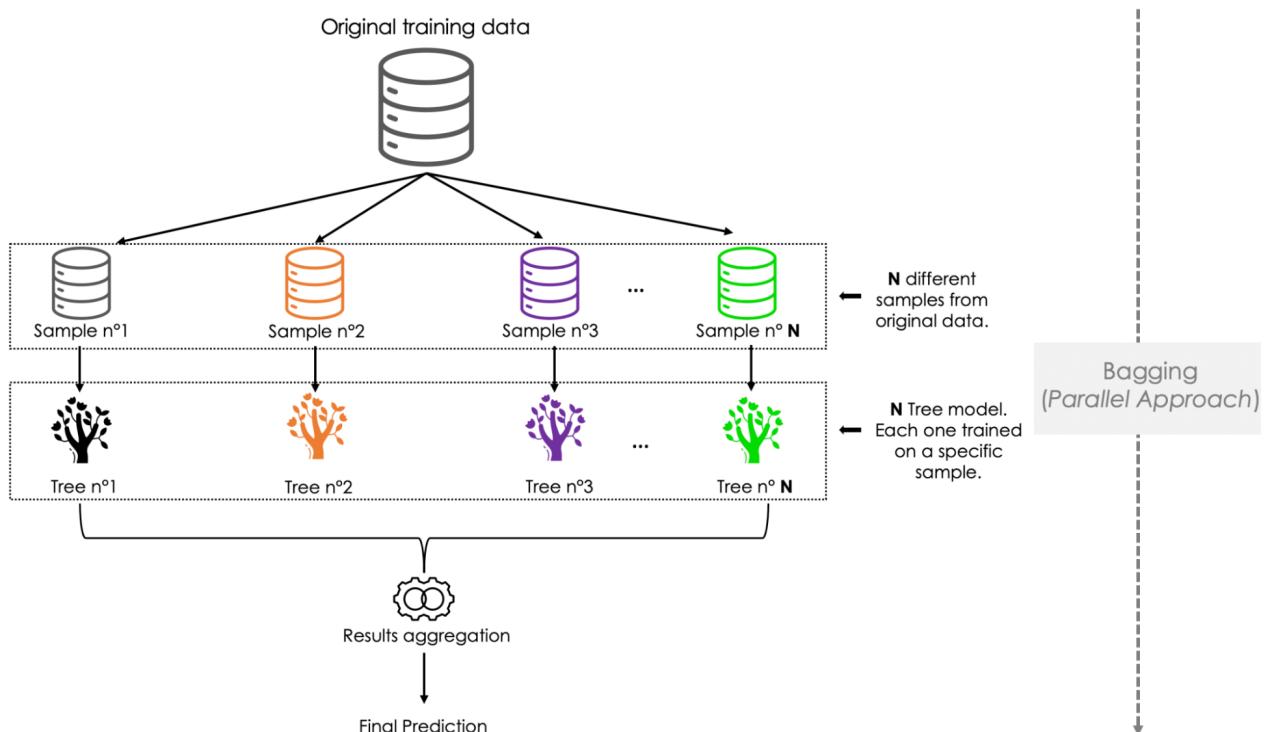
Nel caso specifico d'analisi, nonostante l'applicazione delle tecniche di Oversampling, i risultati ottenuti (Precision, Recall, F1 score), calcolati sulle matrici di confusione, non sono stati ottimali: i modelli non riuscendo a classificare correttamente i veri/falsi positivi e negativi hanno restituito valori di Precision estremamente bassi, nell'ordine dello 0,13, nella maggior parte dei casi.

IMPROVING RESULTS

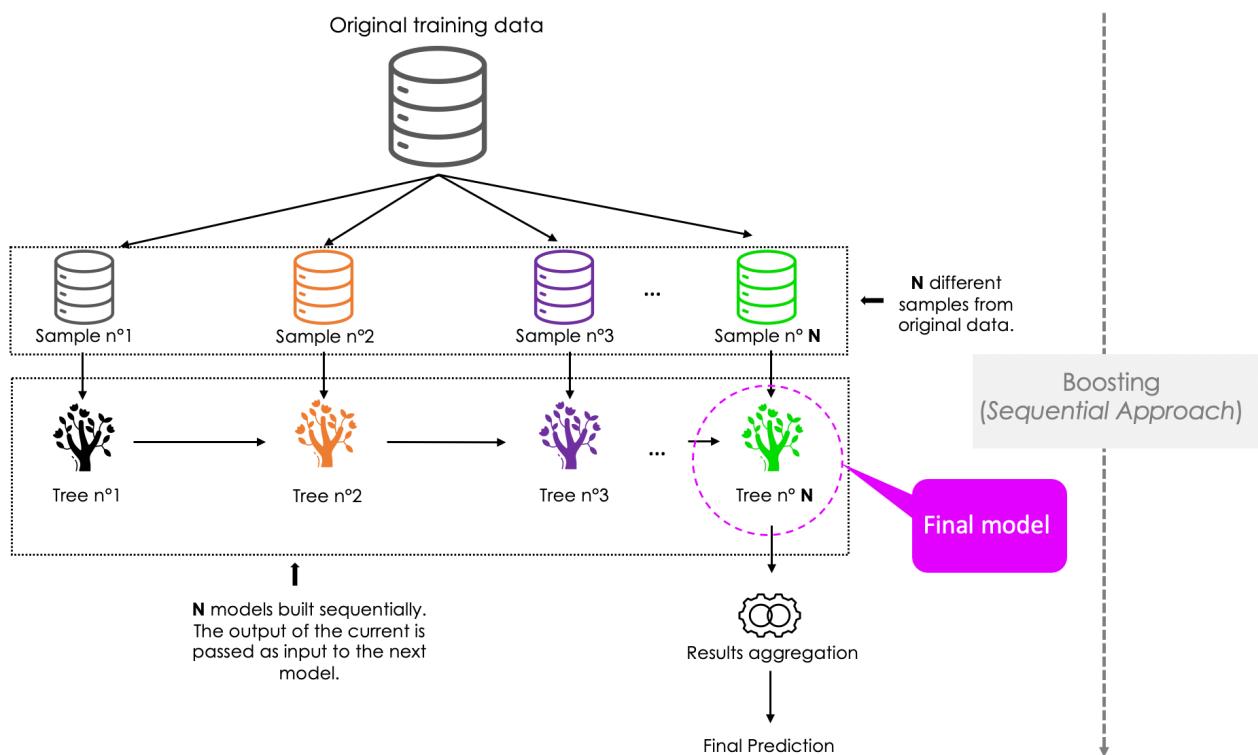
Lo step successivo dell'analisi è stato quindi, quello di migliorare i risultati ottenuti abbandonando l'idea dei dati simulati derivanti dal resampling. Si è deciso di optare per l'applicazione della variabile target pesata sul rapporto di sbilanciamento, selezionando inoltre le migliori feature categoriche attraverso il **Chi Square Test**. Anche in questo caso i due modelli bilanciati (Random Forest e Logistic Regression) non hanno apportato benefici in termini di performance.

L'ultima tecnica applicata concerne il modello di machine learning noto come **Ensemble**, il quale ha l'obiettivo di combinare più algoritmi di classificazione al fine di ottenere performance predittive migliori rispetto quelle date dall'applicazione del singolo modello.

Sono stati applicati due modelli specifici di Ensemble: il **Bagging Ensemble** e il **Boosting Ensemble**.



Il Bagging, conosciuto anche come aggregazione bootstrap, prevede l'estrazione di N sottoinsiemi dal dataframe di riferimento. I modelli di classificazione saranno successivamente addestrati sui sottoinsiemi estratti e la predizione finale sarà data dall'aggregazione delle singole previsioni ottenute dai classificatori.



Il Boosting segue un approccio simile al Bagging per quanto riguardo la creazione dei sottoinsiemi, ma si differenzia da esso per l'esecuzione dei modelli di classificazione. Infatti, mentre nel Bagging si avrà un'esecuzione parallela dei diversi classificatori, con il Boosting si avrà un'esecuzione sequenziale degli algoritmi di classificazione, comportando un miglior apprendimento dei modelli successivi che riceveranno in input eventuali previsioni errate dai precedenti.

I risultati ottenuti dal Bagging e Boosting sono stati pressoché analoghi. Si è assistito ad un lieve miglioramento della Precision, che è passata da un valore di 0,13 allo 0,25, a seconda dei parametri utilizzati in fase di Hypertuning(scelta dei migliori parametri per modello).

CONCLUSIONS

Dati i risultati ottenuti è possibile infine affermare che, nonostante siano state applicate differenti metodologie con lo scopo di risolvere il problema di sbilanciamento di cui la variabile target è affetta, i modelli di classificazione implementati hanno presentato un basso adattamento ai dati e non sono dunque riusciti a spiegare in maniera ottimale il fenomeno in oggetto.

Per una corretta classificazione sarebbe opportuno effettuare un ulteriore implementazione delle osservazioni totali, concatenando ad esempio dataset simili laddove disponibili sul web o in caso contrario, effettuare un lavoro di *web scrapping* per il reperimento "granulare" dei dati sui singoli pazienti.

BIBLIOGRAPHY

Siti visionati:



kaggle

seaborn



jupyter

Repository Progetto:



Dataset utilizzato:

[Stroke Prediction Dataset](#)

Altro materiale:

[Outliers detection](#)

[Resampling Imbalanced Data](#)

[Classification Models](#)

[Ensemble Modeling](#)