RELAZIONE CASO DI STUDIO
INGEGNERIA DELLA CONOSCENZA

Utilizzo di tecniche di apprendimento supervisionato, apprendimento non supervisionato (clustering) e apprendimento stocastico su un dominio turistico

Gruppo di lavoro:

- Angela Di Staso, 722581, a.distaso16@studenti.uniba.it
- Daniele Latini, 718150, d.latini1@studenti.uniba.it

https://github.com/0xBush1do/Icon22_23

AA 2022-2023

Introduzione

Gli obiettivi del caso di studio sono:

- Confrontare diversi modelli di classificazione e regressione al fine di trovare il modello migliore (il più accurato) che effettui la predizione della cancellazione di una prenotazione (classificazione) e il numero di giorni che precedono la cancellazione della stessa (regressione).
- Raggruppare in cluster le diverse tipologie di clienti al fine di ottenere una suddivisione automatica della clientela.
- Utilizzare un modello di apprendimento stocastico (Gaussian Naive Bayes) per fornire un modello di classificazione alternativo a quelli testati in precedenza. Lo scopo della classificazione rimane invariato.

Il dataset utilizzato è disponibile sulla piattaforma *Kaggle* al seguente link:
https://www.kaggle.com/datasets/jessemostipak/hotel-booking-demand

Elenco argomenti di interesse

- Paragrafo 7 per apprendimento supervisionato:
  - Sotto-paragrafo 7.2.2: Tipi di errore.
  - Sotto-paragrafo 7.3: Modelli base per l'apprendimento supervisionato.
  - Sotto-paragrafo 7.4.2: Regolarizzazione.
  - Sotto-paragrafo 7.4.3: Cross Validation.
  - Sotto-paragrafo 7.6: Modelli composti.
- Paragrafo 10 per l'apprendimento non supervisionato:
  - Sotto-paragrafo 10.2: Apprendimento non supervisionato.
- Paragrafo 10 per l'apprendimento probabilistico:
  - Sotto-paragrafo 10.1.2: Modelli probabilistici di classificazione.

Fase Preliminare: Preprocessing del dataset.

Sommario

Per il nostro caso di studio abbiamo fatto riferimento ad un dataset contenente un singolo file che confronta varie informazioni di prenotazione tra due hotel: un hotel di città e un hotel resort. È costituito da 32 features che includono informazioni quali la data in cui è stata effettuata la prenotazione, la durata del soggiorno, il tipo di stanza riservata. Osservando le diverse features si nota che alcune di esse sono di tipo categorico e altre di tipo numerico quindi il dataset può essere considerato ibrido. Proprio per questa ragione è stato necessario un lavoro di preprocessing preliminare.

Strumenti utilizzati

Per effettuare il preprocessing del dataset in questione sono state utilizzate diverse librerie quali:

- *Pandas*: Utilizzato per la gestione dei dataframes.
- *Sklearn.preprocessing:* Libreria di sklearn utilizzato per effettuare l'encoding delle features categoriche in numeriche.

Decisioni di Progetto

Le modifiche e le trasformazioni che sono state effettuate sono le seguenti:

- Creazione di una nuova feature "weekend_or_weekday" in modo da accorpare le features "stays_in_weekend_nights" e "stays_in_week_nights" già presenti e ottenere, in generale, il periodo di soggiorno dei clienti nella struttura.
- Conversione dei mesi dell'anno in numeri interi che vanno da 1 a 12 in modo da semplificare il loro successivo utilizzo.
- Fusione delle features "children" e "babies" nella singola feature "all_children" in modo da includere i minori in una sola fascia di età fino ai 10 anni.
- Compilazione dei Not a Number (NaN) con il valore 0 tramite la funzione *fillna()* (della libreria *pandas*) per evitare che questi influenzino negativamente i risultati dell'analisi dei dati.
- Per la gestione delle feature categoriche è stato usato un *LabelEncoder*. Con questa tecnica, a ciascuna feature viene assegnato un numero intero univoco in base all'ordine alfabetico.
- Eliminazione di alcune features ridondanti o che sono risultate essere poco utili per il nostro scopo.

Al termine del preprocessing, il nuovo dataset ottenuto è costituito da 23 features, il numero minimale considerato da noi utile, tutte di tipo numerico.

Prima Fase: Classificazione e Regressione

Sommario

Nella prima fase sono stati creati e testati diversi modelli di classificazione e regressione al fine di predire l'eventuale cancellazione di una prenotazione e il numero di giorni che precedono la stessa cancellazione. I modelli creati sono stati infine confrontati gli uni con gli altri per trovarne uno quanto più preciso possibile e che si adatti meglio ai dati.

Strumenti utilizzati

L'apprendimento supervisionato prevede due tipologie di task: la classificazione e la regressione. La prima è utilizzata per prevedere la categoria o la classe a cui appartiene un dato, mentre la seconda è utilizzata per prevedere un valore numerico continuo. In entrambi i casi, il modello di apprendimento supervisionato è addestrato su un insieme di dati di esempio con etichette note per fare previsioni su dati nuovi e non etichettati. Nel caso specifico, al fine di trovare un modello che fosse migliore rispetto agli altri, sono stati costruiti e testati quattro modelli di classificazione (Decision Tree, Random Forest Classification, Gradient Boosting e Adaboost) al fine di predire l'eventuale cancellazione di una prenotazione, e tre modelli di regressione (Linear Regression, Ridge Regression e Adaboost Regression, Random Forest Regression) per predire il numero di cancellazioni effettuate da un utente in precedenza.

Di seguito sono riportati i fondamenti teorici dei modelli quali Gradient Boosting, Adaboost e Ridge regression.

Gradient Boosting

Il Gradient boosting è una tecnica di ensemble learning che consiste nella creazione di alcuni "weak learners", ognuno con un peso associato, che viene ricalibrato in modo da minimizzare la funzione di errore (log loss) del modello tramite la discesa di gradiente. Quest'ultimo è un algoritmo di ottimizzazione iterativa del primo ordine utile per trovare un minimo locale in una funzione differenziabile.

Ada Boost

L'algoritmo AdaBoost, abbreviazione di Adaptive Boosting, è una tecnica di Boosting utilizzata come metodo Ensemble nel Machine Learning. Si chiama Adaptive Boosting perché i pesi vengono riassegnati a ogni istanza, con pesi più alti assegnati alle istanze classificate in modo errato. Il boosting viene utilizzato per ridurre i bias e la varianza nell'apprendimento supervisionato. Funziona secondo il principio della crescita sequenziale dei learners. Ad eccezione del primo, ogni learner viene fatto crescere a partire dagli learners cresciuti in precedenza. In parole povere, i weak learners vengono convertiti in quelli forti.

Ridge Regression

La Ridge Regression è una versione regolarizzata (secondo la norma L2) della Linear Regression in quanto alla funzione di costo originale della LR si aggiunge un termine regolarizzato che costringe l'algoritmo di apprendimento a adattarsi ai dati e aiuta a

mantenere i pesi più bassi possibile. Il termine regolarizzato ha il parametro "alfa" che controlla la regolarizzazione del modello, ossia aiuta a ridurre la varianza delle stime.

Decisioni di Progetto

Ogni modello è stato testato attraverso una k-fold cross validation, che consiste nella suddivisione del training set in k partizioni di cui, in maniera iterativa, se ne utilizzano k-1 per l'addestramento e la restante parte per il test. Facendo ciò, è possibile valutare la bontà di un modello anche qualora non fossero disponibili nuovi dati di test. Nel caso specifico è stato utilizzato un valore di k pari a cinque.

Al fine di agevolare l'addestramento di ogni modello, il training set e il test set sono stati scalati tramite il metodo *StandardScaler()* presente nella libreria *preprocessing* di *sklearn*.

È stata utilizzata la tecnica del *hyperparameter tuning* con *Grid Search* al fine di trovare i migliori parametri per ogni modello (classificazione e regressione). Quest'ultimo è un metodo di regolazione degli iperparametri "a forza bruta". In particolare, si crea una griglia di possibili valori discreti per gli stessi iperparametri e si adatta il modello con ogni possibile combinazione. Infine, si registrano le prestazioni del modello per ogni serie e si seleziona la combinazione che ha prodotto le migliori prestazioni. La classe utilizzata per tale scopo è *GridSearchCV* presente nella libreria *sklearn.model_selection.*

Esempio di output di GridSearchCV per il modello Decision Tree:

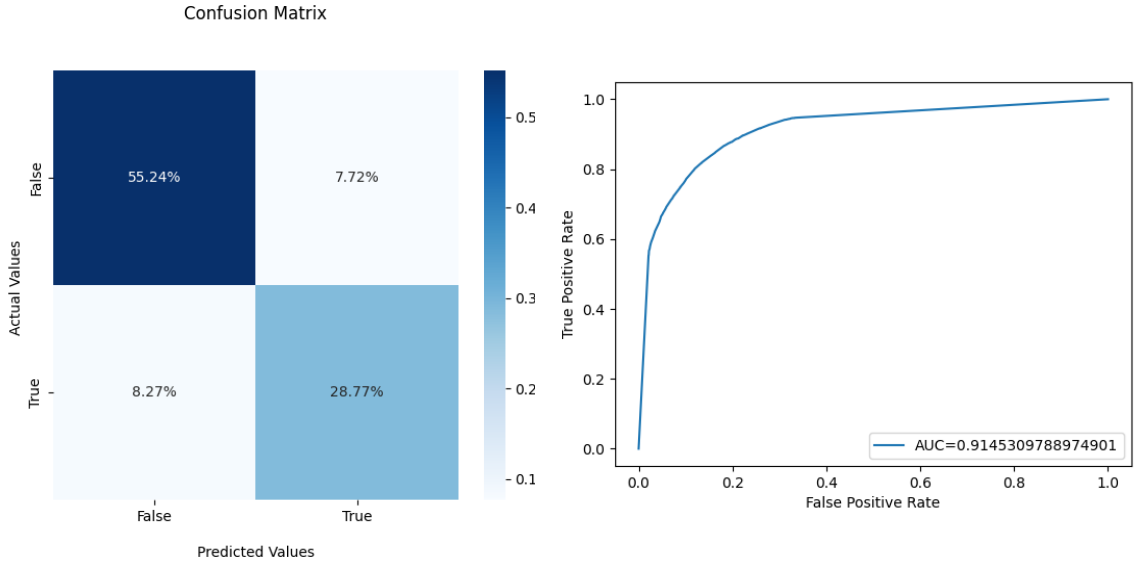
Parametri	***Valori***
Criterion	Entropy
Max_features	Sqrt
Min_samples_leaf	1
Min_samples_split	6

- Criterion = La funzione per misurare la qualità di una suddivisione. I criteri supportati sono "gini" per l'impurità di Gini ed "entropia" per l'information gain.
- Max_features = Il numero di caratteristiche da considerare quando si cerca la migliore suddivisione.
- Min_sample_leaf = Il numero minimo di campioni richiesto in un nodo foglia. Un punto di divisione a qualsiasi profondità sarà considerato solo se lascia almeno ``min_samples_leaf`` campioni in ciascuno dei rami sinistro e destro.
- Min_samples_split = Il numero minimo di campioni necessari per dividere un nodo interno.

Visualizzazione delle prestazioni dei modelli di classificazione

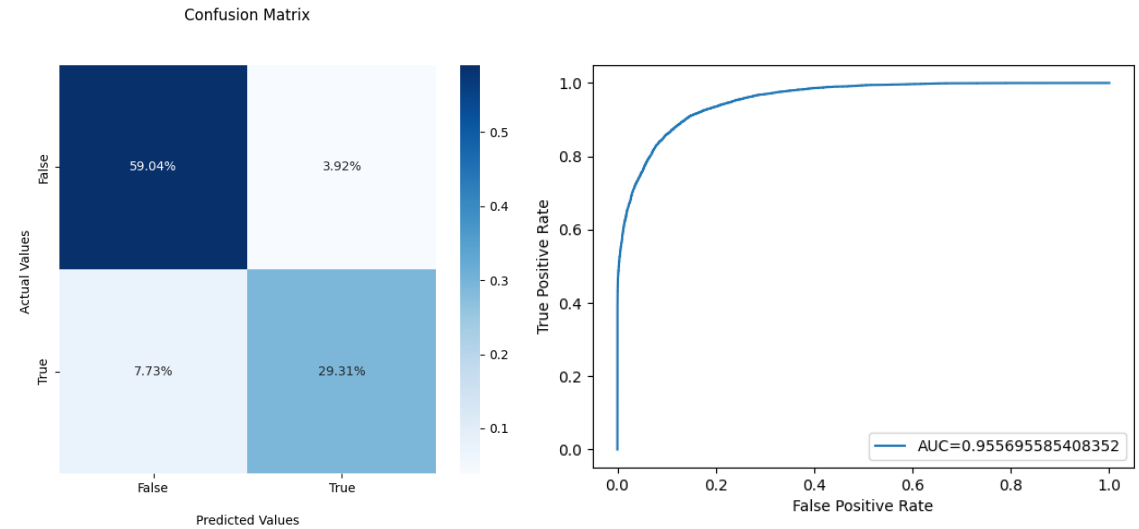
DecisionTree Classification Report, Matrice di confusione, curva ROC:

Output	Precision	Recall	F1-Measure	Accuracy
0	0.87	0.88	0.87	0.85
1	0.79	0.78	0.78	



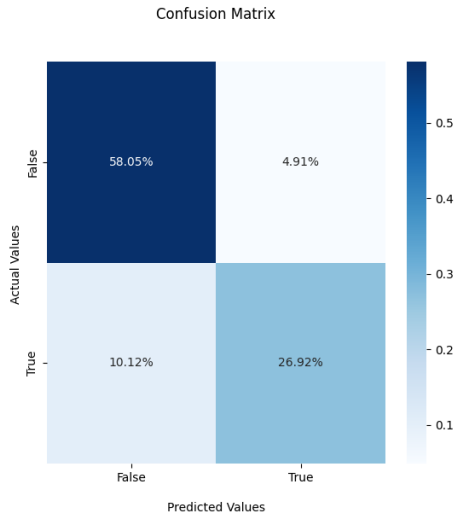
Random Forest Report, Matrice di confusione, curva ROC:

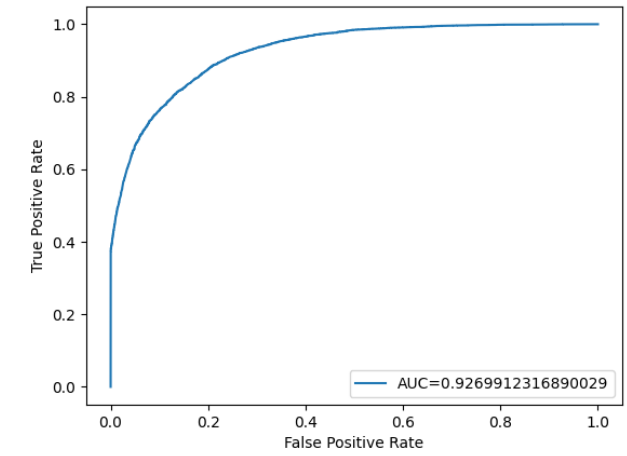
Output	Precision	Recall	F1-Measure	Accuracy
0	0.88	0.94	0.91	0.88
1	0.88	0.79	0.83	



Gradient Boosting Report, Matrice di confusione, curva ROC:

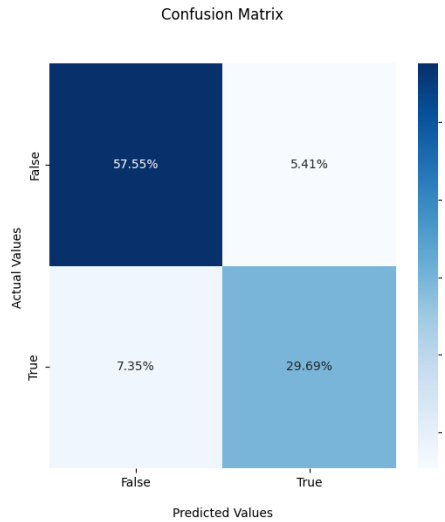
Output	Precision	Recall	F1-Measure	Accuracy
0	0.85	0.92	0.89	0.85
1	0.85	0.73	0.78	

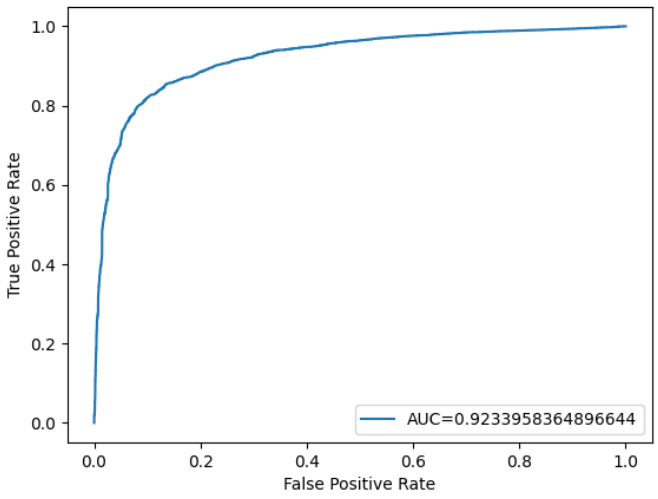




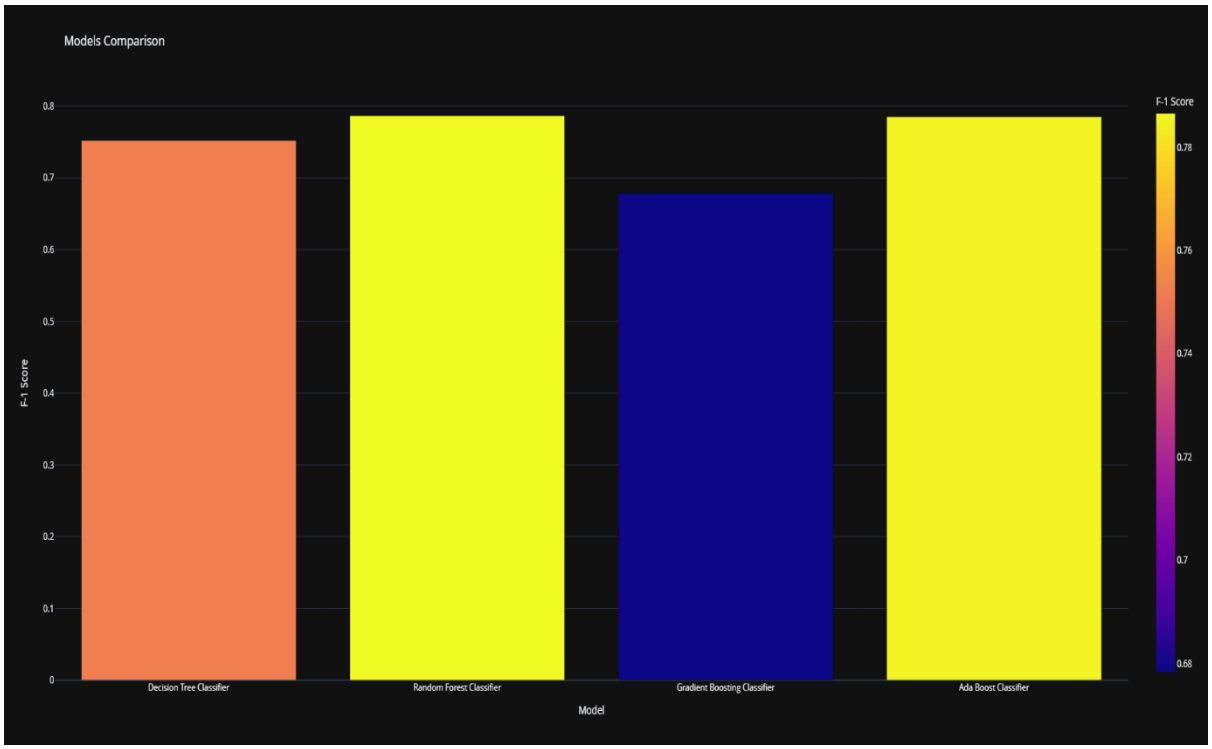
Adaboost Report, Matrice di confusione, curva ROC:

Output	Precision	Recall	F1-Measure	Accuracy
0	0.89	0.91	0.90	0.87
1	0.85	0.80	0.82	





Modelli di classificazione a confronto:



Considerazioni sui modelli di classificazione utilizzati

In conclusione, si può affermare che i modelli che meglio si sono adattati ai dati sono: *Random Forest* e *Adaboost.* Entrambi hanno riportato una accuratezza prossima al 90% (rispettivamente 88% e 87%) e uno score F-1 rispettivamente di 0.79 e 0.78. Per valutare la bontà di ogni modello sono state utilizzare le matrici di confusione (utili per visualizzare le percentuali di falsi positivi, falsi negativi, veri positivi, veri negativi) e la curva ROC con AUC.

Per il confronto tra i modelli è stato deciso di utilizzare la metrica F-1 in quanto il dataset risulta essere sbilanciato (sono presenti per la feature “is_canceled” più esempi di classe *0* che di classe *1*).

Distribuzione degli esempi per la feature “is_canceled”:

La curva ROC viene creata tracciando il valore del *True Positive Rate* (TPR, frazione di veri positivi) rispetto al *False Positive Rate* (FPR, frazione di falsi positivi), mentre AUC è l’area sottesa alla curva ROC e può assumere un valore compreso tra 0 e 1 (un buon modello presenta un AUC prossimo ad 1). Nel caso specifico l’AUC di Random Forest e di Adaboost sono rispettivamente 0.95 e 0.92.

Visualizzazione delle prestazioni dei modelli di regressione

Linear Regression Report ed Errori:

MAE	MSE	RMSE	R^2
66.4	7623.3	87.3	0.33

Ridge Regression Report ed Errori:

MAE	MSE	RMSE	R^2
66.6	7625.2	87.3	0.31

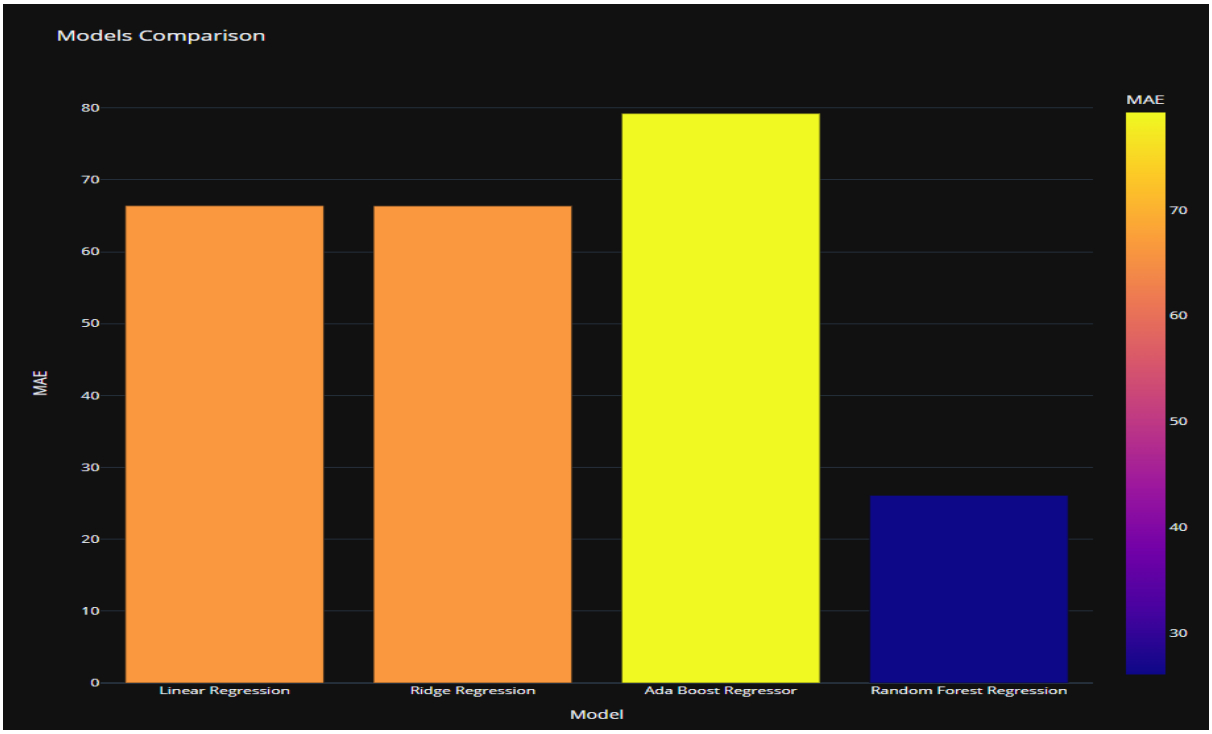
Adaboost Regression Report ed Errori:

MAE	MSE	RMSE	R^2
79.2	8658.9	93	0.24

Random Forest Regression Report ed Errori:

MAE	MSE	RMSE	R^2
26	1876.4	43.3	0.84

Modelli di regressione a confronto:



Al fine di valutare ogni modello di regressione, sono state calcolate diverse misure di errore:

- Errore Medio Assoluto (MAE): $\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} = \frac{\sum_{i=1}^{n} |e_i|}{n}$
- Errore Quadratico Medio (MSE): $\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n} \left(Y_i - \hat{Y}_i\right)^2$.
- Radice dell'Errore Quadratico Medio (RMSE): $RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(f_i - o_i)^2}$

È stato utilizzato lo score R^2 per valutare la bontà di ogni modello generato:

- R^2 score, noto anche come coefficiente di determinazione, è la differenza tra i campioni nel dataset e le previsioni effettuate dal modello. Un modello che più si adatta ai dati ha valori di R^2 prossimi ad 1.

A seguito delle valutazioni sui quattro modelli testati, si può concludere che il *Random Forest Regressor* risulta essere il modello che meglio si adatta ai dati e che meglio predice il valore di "lead_time" (i giorni che precedono la cancellazione di una prenotazione). Ciò si evince dalle basse misure di errore (MAE, MSE, RMSE) e dal valore R^2 molto vicino ad 1 (0.85).

Seconda fase: Apprendimento non supervisionato

Sommario

Per cercare di organizzare e classificare i dati in maniera efficiente e mirata al nostro caso di studio è stato utilizzato un metodo di apprendimento non supervisionato, il clustering.

Il clustering consiste nel raggruppare dati in classi omogenee chiamate cluster.

L'obiettivo è quello di ottenere dei cluster dove è minimizzata la distanza tra dati dello stesso cluster (intra-cluster) e massimizzata quella tra cluster diversi (inter-cluster).

Abbiamo due tipi di clustering:

- *Hard clustering*: Ogni dato viene assegnato a una classe precisa.
- *Soft clustering*: Si ha una distribuzione di appartenenza dei dati alle varie classi.

In particolare, tra i vari algoritmi di hard clustering, sono stati presi in considerazione il k-means e il k-modes.

Strumenti utilizzati

Il K-means identifica un numero k di centroidi ovvero i punti che rappresentano il centro del cluster, e assegna ogni dato al cluster il cui centroide risulta più vicino (tipicamente si utilizza

la distanza euclidea). Esso inizia con un primo gruppo di centroidi selezionati in modo casuale, che vengono utilizzati come punti di partenza per ogni cluster, successivamente esegue calcoli iterativi per ottimizzare le posizioni dei centroidi. K-means termina la creazione e l'ottimizzazione dei cluster quando i centroidi si sono stabilizzati e non vi è alcun cambiamento nei valori oppure quando il numero massimo di iterazioni è stato raggiunto.

Il k-modes è un'estensione dell'algoritmo K-means e usa la moda al posto della media come parametro per scoprire i centroidi del cluster ed è utilizzato per stabilire la dissimilarità tra attributi categorici.

La differenza sostanziale che porta a preferire l'uno rispetto all'altro è che il primo viene applicato solitamente a dati numerici mentre il secondo a dati categorici.

Decisioni di Progetto

Ottimizzazione della misura d'errore

Il k-Means migliora iterativamente l'errore quadratico, ovvero lo scostamento che c'è tra il valore reale dei dati per ognuna delle features di input e la classe del dato, assegnata dall'algoritmo. Sommando l'errore quadratico per i dati di ogni feature osservata, si ottiene il Within-Cluster-Sum of Squared Errors (WSS).

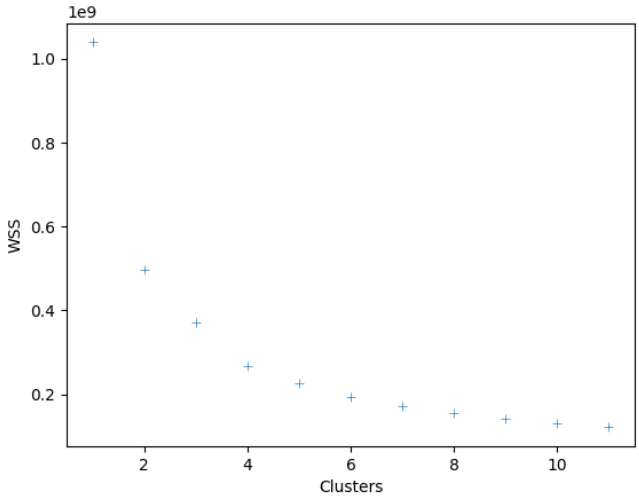
K-Elbow

Per determinare il numero ottimale di cluster in cui i dati possono essere raggruppati è stato utilizzato il metodo *Elbow***.**

Esso consiste nell'interpretazione di un grafico a linee con una forma a gomito.

Il valore di k da selezionare sarà quello che determina il "gomito", cioè il punto dopo il quale la distorsione inizia a diminuire in modo lineare.

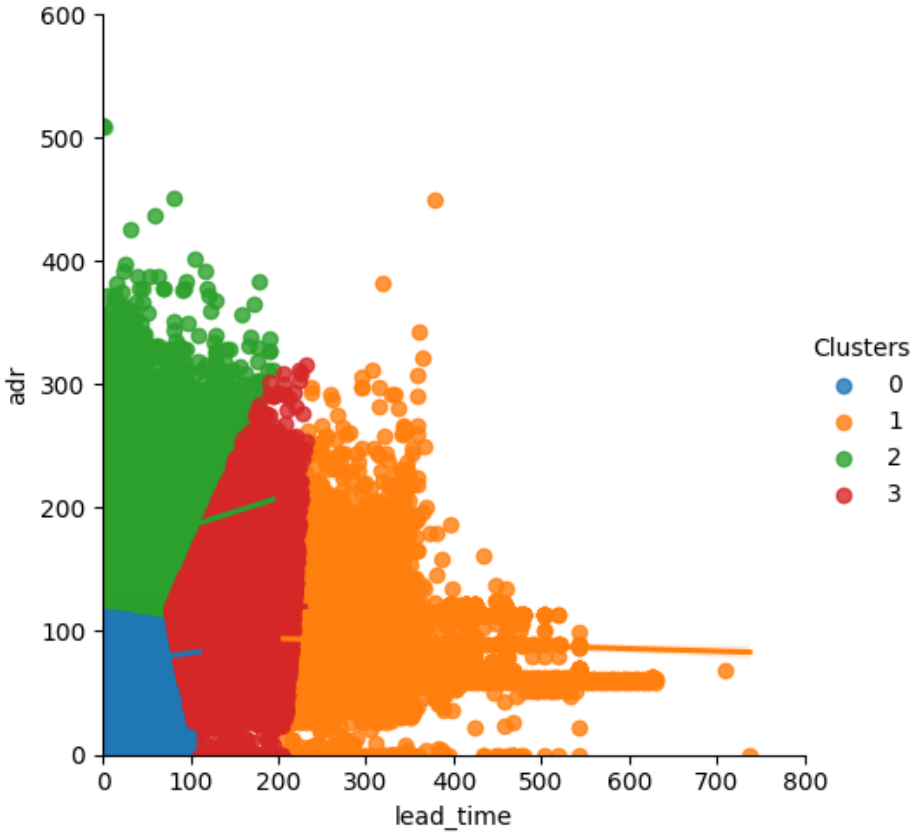
Dopo aver plottato i cluster sull'asse delle ascisse e la somma degli errori al quadrato (WWS) sull'asse delle ordinate, *la curva del gomito* è la seguente:



Pertanto, per i dati a disposizione, concludiamo che il numero ottimale di cluster è 4.

Visualizzazione dei Cluster

Dopo aver plottato il tutto su un grafico con la feature “lead_time” sull’asse delle ascisse e quella “adr” sull’asse delle ordinate, il *risultato del clustering* è il seguente:



Terza fase: Apprendimento probabilistico

Sommario

Il bisogno di utilizzare un modello probabilistico nasce dalla possibile mancanza dei dati utili a risolvere un determinato problema. Una conoscenza limitata, dunque, implica una forma di incertezza chiamata epistemologica (che riguarda le credenze sullo stato del mondo). I modelli probabilistici sono particolarmente utili per i problemi di classificazione e regressione, in cui l'obiettivo è prevedere una categoria o un valore numerico in base a un insieme di input, data una assunzione di incertezza e di indipendenza tra le features. I modelli probabilistici più comuni sono: le reti bayesiane e i modelli di Markov. Essi sono utilizzati in una vasta gamma di applicazioni, tra cui il riconoscimento del linguaggio naturale, la previsione delle vendite e la diagnostica medica. Nel caso specifico, è stato utilizzato l’algoritmo Gaussian Naive Bayes per fornire un ulteriore modello predittivo di classificazione per la feature “is_canceled” che indica la cancellazione (o meno) di una prenotazione.

Strumenti utilizzati

Gaussian Naive Bayes

Il modello Naive Bayes può essere esteso ad attributi con valore reale, più comunemente assumendo una distribuzione gaussiana, questa estensione è chiamata Gaussian Naive Bayes. È possibile utilizzare altre funzioni per stimare la distribuzione dei dati, ma la distribuzione gaussiana (o normale) è la più semplice da utilizzare perché è sufficiente stimare la media e la deviazione standard dai dati di addestramento. Quando si lavora con dati continui, spesso si assume che i valori associati a ciascuna classe siano distribuiti secondo una distribuzione normale (o gaussiana). Si assume quindi, che la verosimiglianza delle caratteristiche sia la seguente:

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

Dove:

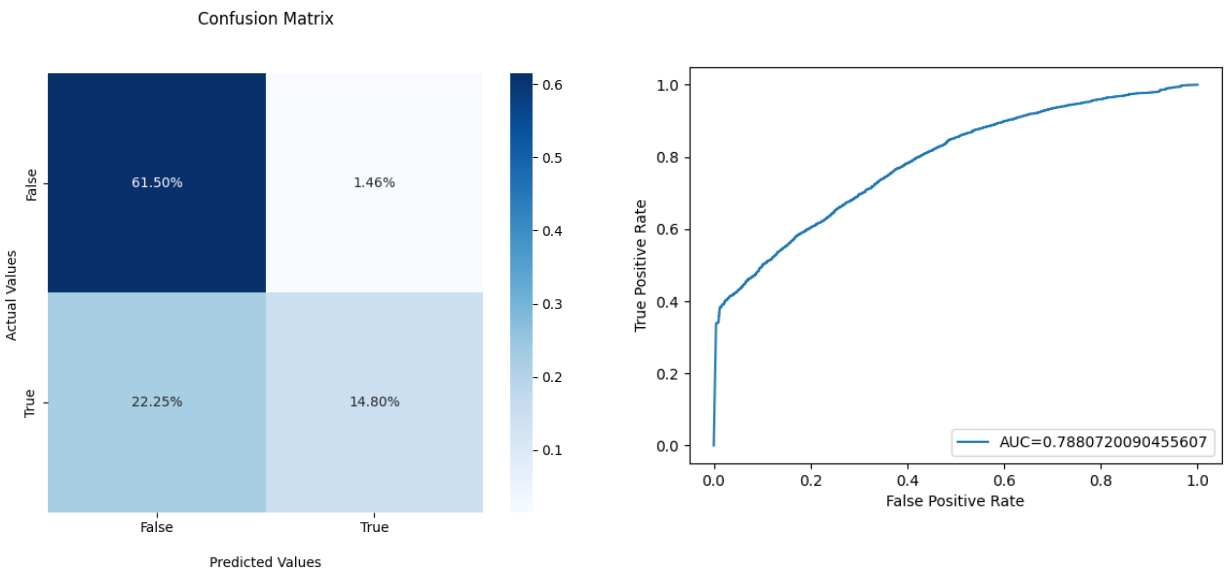
- σ^2_y: Varianza.
- μ: Media delle variabili continue X per una data classe y di Y

Un approccio per creare un modello semplice consiste nell'assumere che i dati siano descritti da una distribuzione gaussiana (detta anche distribuzione normale) senza co-varianza (dimensioni indipendenti) tra le dimensioni. Questo modello può essere adattato semplicemente trovando la media e la deviazione standard dei punti all'interno di ogni etichetta, il che è tutto ciò che serve per definire tale distribuzione.

Visualizzazione delle prestazioni di Gaussian Naive Bayes

Gaussian Naive Bayes Report, Matrice di confusione, curva ROC:

Output	Precision	Recall	F1-Measure	Accuracy
0	0.73	0.98	0.84	0.76
1	0.91	0.40	0.56	



Considerazioni sul modello

Al fine di fornire un modello di classificazione alternativo, sono stati testati precedentemente diversi algoritmi di classificazione stocastica, tra cui *Bernoulli Naive Bayes* e *Multinomial Naive Bayes*, ma il modello che meglio si adatta al dataset preso in esame risulta essere il Gaussian Naive Bayes con una accuratezza del 76%, uno score F-1 di 0.55 e un valore di AUC di 0.79. Si precisa che le prestazioni riportate sono state incrementate notevolmente grazie all'utilizzo di alcune tecniche quali:

- k-fold cross validation e regolarizzazione dei dati di training.
- Hyperparameters tuning con GridSearch al fine di utilizzare i migliori parametri per il modello.

Conclusioni

Il caso di studio ci ha permesso di comprendere le possibili applicazioni dei modelli di apprendimento supervisionato e non supervisionato in un contesto reale.

L'apprendimento automatico può essere utile per effettuare la predizione che un evento si verifichi o meno, come ad esempio la cancellazione di una prenotazione, oppure il numero di giorni che la precedono. Un modello preciso, dunque, può essere utilizzato come strumento a supporto dell'uomo al fine di agevolarlo nei compiti che altrimenti risulterebbero impossibili poiché richiederebbero la considerazione di un considerevole numero di variabili.

Alcune possibili estensioni potrebbero essere:

- La creazione di una base di conoscenza, al fine di permettere la costruzione di un sistema che interagisca in maniera intelligente con l'utente.
- L'integrazione di una ontologia, al fine di garantire l'interoperabilità sintattica e semantica, ossia la capacità di diverse fonti di conoscenza di collaborare sul piano sintattico e semantico.
- L'utilizzo di modelli probabilistici sequenziali, come le catene di Markov.