Università degli Studi di Milano Bicocca

Scuola di Scienze

Dipartimento di Informatica, Sistemistica e Comunicazione

Corso di laurea in Informatica

Link dataset: https://www.kaggle.com/datasets/teejmahal20/airline-passenger-satisfaction

MACHINE LEARNING

Progetto d'esame

Riccardo Moschi 856243 r.moschi@campus.unimib.it
Luca Poli 852027 l.poli6@campus.unimib.it

Indice

1. Introduzione

1.1. Descrizione del dominio di riferimento e obiettivi

Il dominio di riferimento di questo elaborato sono i viaggi in aero; e in particolare, la soddisfacibilità del cliente in relazione con alcune metriche del volo.
La ricerca mirava a stabilire la relazione tra la soddisfazione del cliente e queste variabili di influenza, utilizzando un metodo di correlazione per verificarne l'effetto, col fine di riuscire a predire la soddisfazione dei futuri clienti.

1.2. Scelte di design per la creazione del data set, eventuali ipotesi o assunzioni

Si è deciso di analizzare questo dataset visto il dominio applicativo di nostra conoscenza.
Abbiamo effettuato una ristrutturazione del dataset, cambiando attributi stringhe in numerici, ed abbiamo effettuato uno scaling che ci permette di normalizzare il range di variazione delle caratteristiche (feature).
Per motivi progettuali è stato scelto di utilizzare solo un campione di dati nel dataset, visto l'enorme carico computazionale che avrebbe comportato all'elaboratore.

2. Dataset

2.1. Preprocessing

Dopo il caricamento in memoria del dataset, e prima della sua analisi bisogna effettuare le primissime operazioni di preprocessing. Visto l'uso "accademico" dei dati, e per velocizzare i tempi di calcolo e di elaborazione, abbiamo deciso di prelevare casualmente un campione di 5000 istanze (a partire dalle circa 100000).

La fase di preprocessing inizia rimuovendo le variabili inutili (come l'ulteriore indice e l'id); in seguito, si procede spostando e rinominando la variabile obbiettivo della classificazione, in modo da semplificarne l'accesso; successivamente si eliminano le righe nulle (visto che sono poche non è necessario imputarle); infine si effettua la conversione in factor delle variabili non numeriche.

2.2. Analisi esplorativa

L'analisi esplorativa inizia con l'analisi delle distribuzioni delle variabili, tramite box plot e density plot; successivamente procede con la valutazione delle distribuzioni in relazione al target, esaminando sia la frequenza delle due classi, sia quelle dei dati in rapporto alla classe di appartenenza.

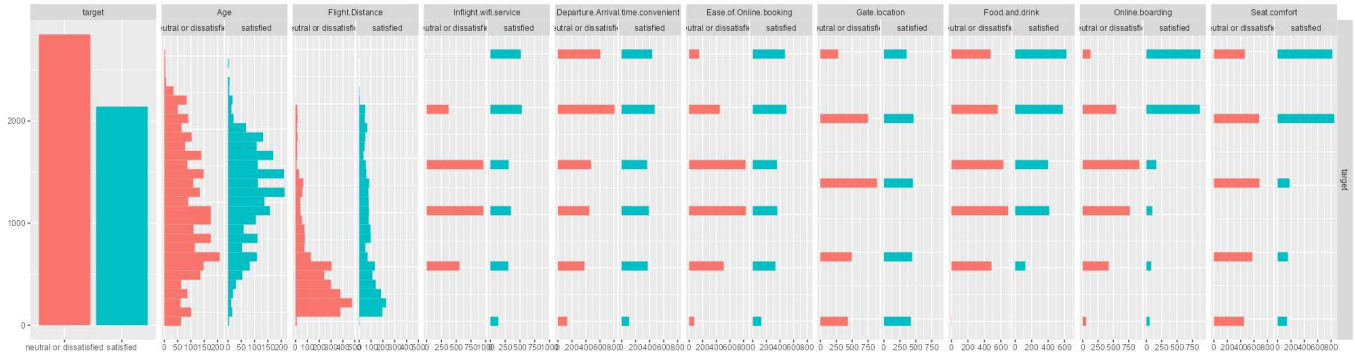

Figura 1: Grafo riassuntivo di gran parte delle variabili in relazione al target

Il grafico presenta la distribuzione delle feature in relazione con il target, cioè la soddisfazione o meno del cliente in base agli altri dati del dataset, sotto forma di istogrammi.

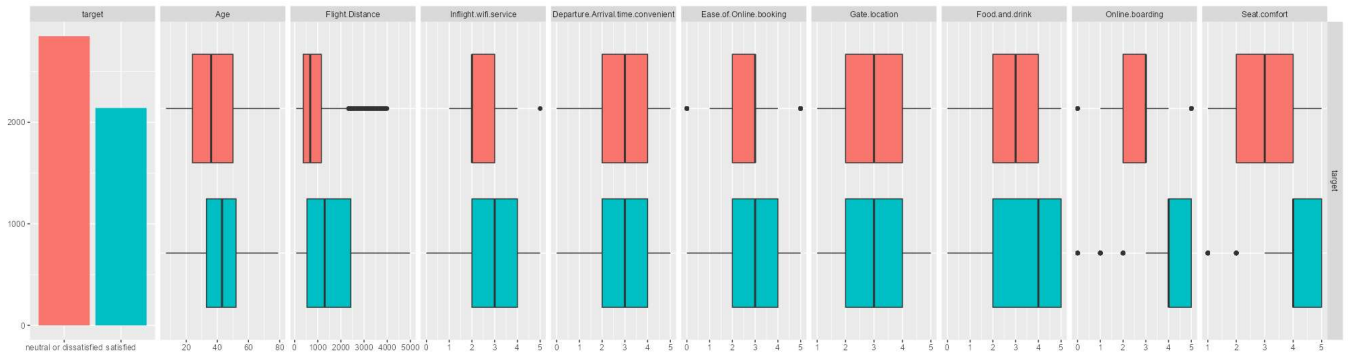

Figura 2: Grafo riassuntivo dei boxplot in relazione al target

Il grafico mostra la distribuzione delle feature in relazione col target, sotto forma di boxplot.
Nei seguenti boxplot è possibile osservare alcuni dati che si discostano significativamente dal resto dei dati (outliers).
Per esempio, è possibile osservare come l'attributo "Flight.distance" abbia distribuzione tendente a sinistra (come mostrato anche nella *figura 1*) e come ci aspettiamo, presenta diversi outliers in quanto è probabile possa assumere molteplici valori di distanza che si distacchino di molto dalla media.

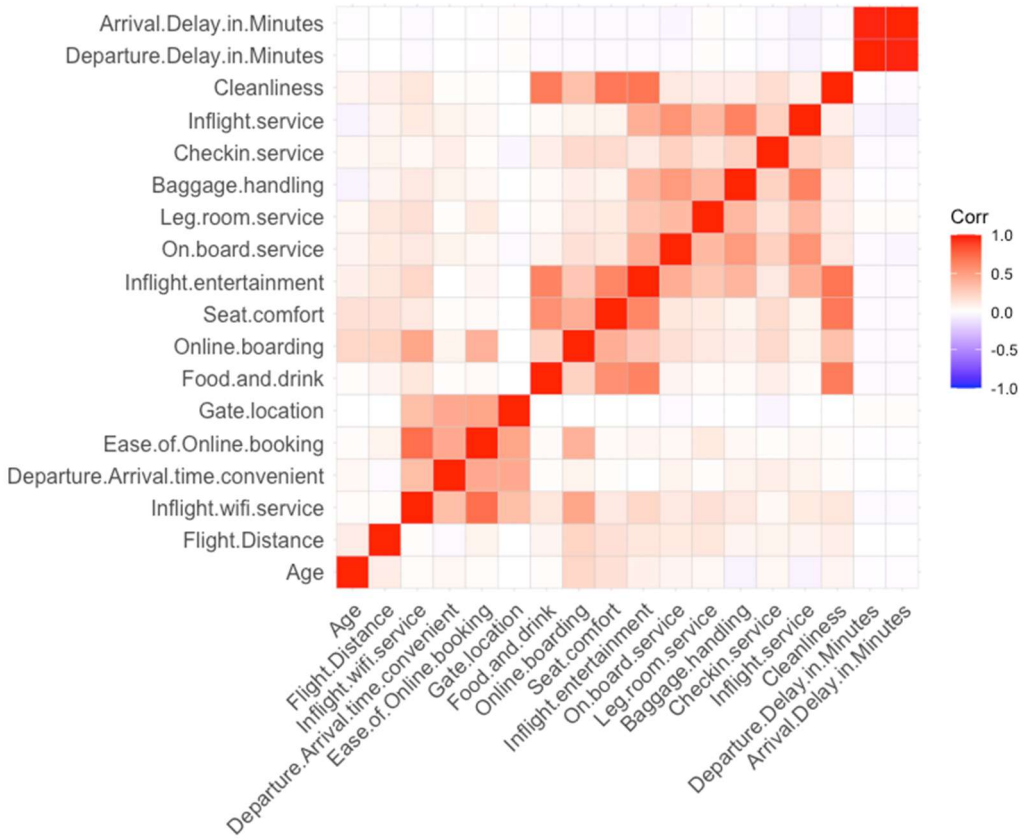

Figura 3: Matrice correlazione degli attributi

La correlation matrix, ovvero la matrice di correlazione indica se due variabili sono in qualche modo legate tra loro, in particolare è possibile osservare una buona correlazione tra le features "Cleanliness" e "food.and.drink", "Seat.comfort", "Inflight.entertainment", e soprattutto tra "Arrival.Delay.in.Minutes" e "Departure.Delay.in.Minutes" che presentano il massimo legame.

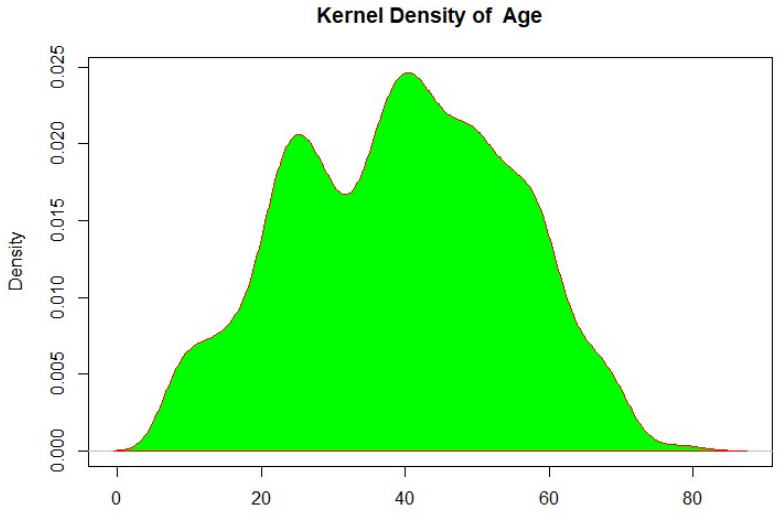

Figura 4: Kernel density dell'età dei passeggeri

È possibile osservare come l'età sia maggiormente addensata al centro, mostrando così una maggiore frequenza sull'età intorno ai 40 anni.

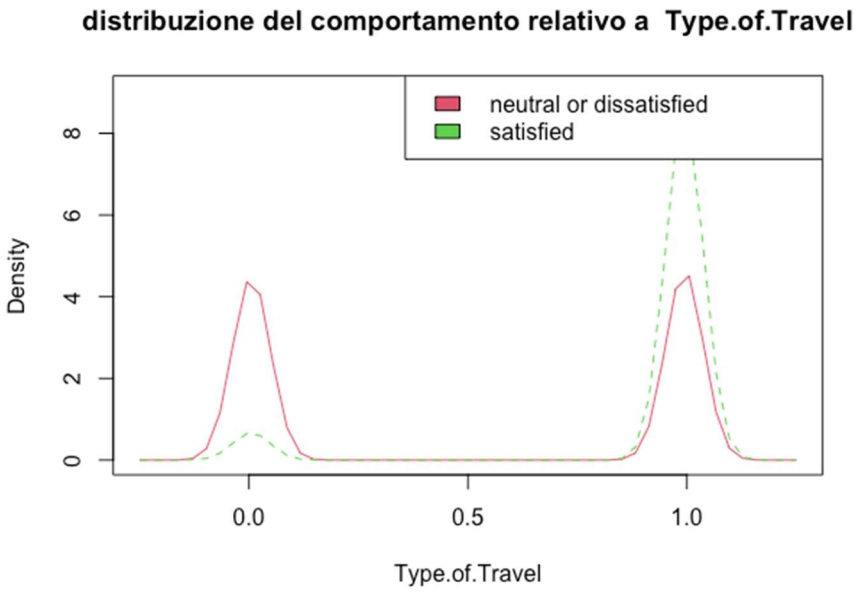

Figura 5: Distribuzione comportamento rispetto al tipo di viaggio di lavoro (se per svago o per lavoro)

È possibile osservare la distribuzione del comportamento (soddisfatto o insoddisfatto), in relazione al tipo di viaggio (0= "Personal Travel",1 =" Business.travel") e contrariamente a quanto potremmo aspettarci, tendenzialmente i viaggi maggiormente graditi sono quelli per Business.

2.3. Feature Reduction

Visto il grande numero di feature abbiamo deciso di applicare due strategie (Boruta e PCA) per provare a ridurle.

2.3.1. Feature Selection: Boruta

Boruta è un algoritmo che estendendo i modelli Random Forest, permette di selezionare le feature migliori da mantenere nella successiva fase.

L'algoritmo Random Forest applica le tecniche di bagging agli alberi di decisione; cioè estrae dei campioni casuali dal train set, con istanze e feature diverse; successivamente, su ogni campione applica un Decision Tree e a seconda dello split ottenuto attribuisce una stima di importanza.

Boruta applica un principio simile ai Random Forest: prima, a partire da ogni feature, ne genera una versione "sintetica" detta Shadow; poi applica il Random Forest sul dataset esteso ottenendo la stima d'importanza; successivamente va a rimuovere tutte le feature meno importanti rispetto alla shadow più importante; ripete questi due step fino ad avere solo quelle più importanti (o fino ad un limite massimo).

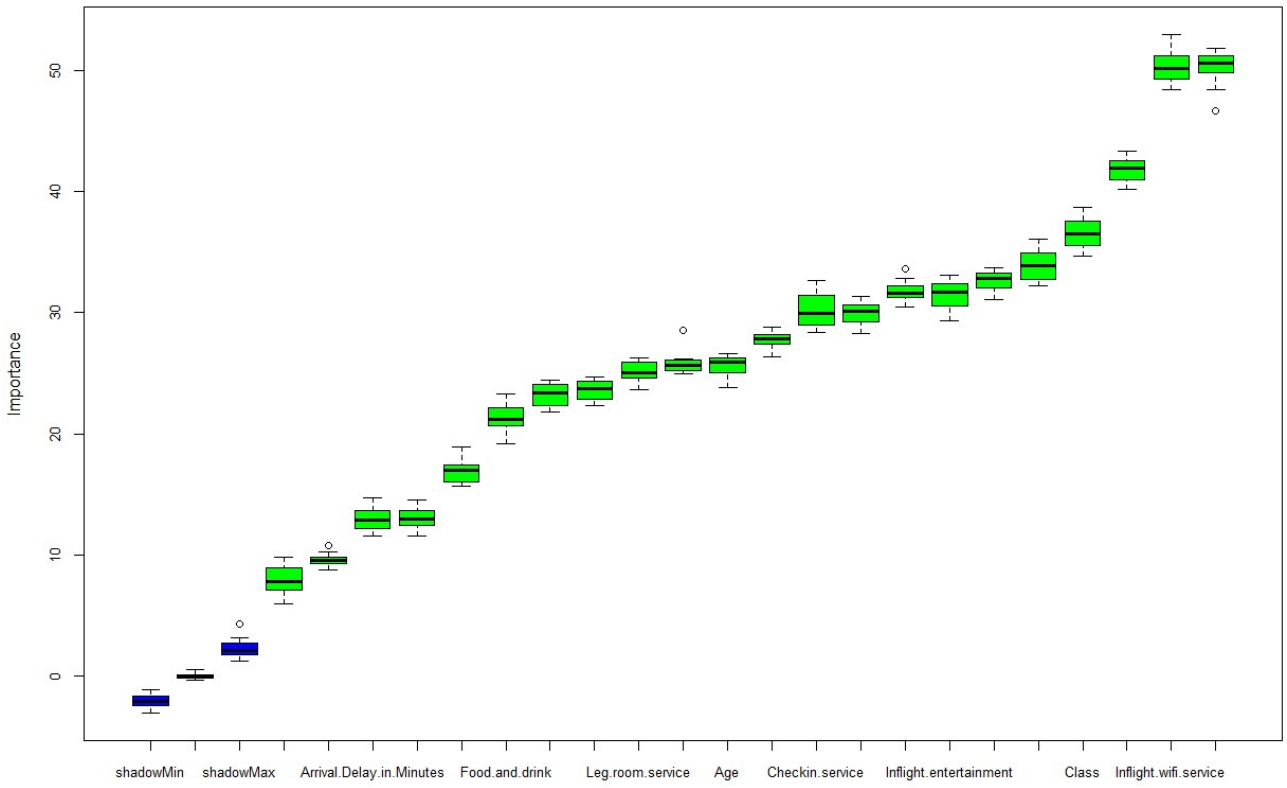

Figure 1: Stime d'importanza per ogni feature

Inflight.wifi.service	50,270
Online.boarding	50,223
Type.of.Travel	41,790
Class	36,640
Seat.comfort	33,890

Figure 2: Tabella delle feature ordinata per importanza media

Come si nota dalla tabella e dal grafico (che indica la distribuzione dell'importanza per ogni feature nei vari cicli), l'algoritmo Boruta non individua feature superflue (infatti tranne le shadow, in blu, ogni feature è in verde).

2.3.2. Feature Extraction: PCA

Dato il numero elevato di features, è stata implementata una strategia di feature extraction, la PCA, in modo da estrarre dai dati numerici le componenti principali che influiscono maggiormente sulla varianza dei dati.
Abbiamo deciso di scegliere le prime 7 componenti per raggiungere il nostro obiettivo, in quanto volevamo ottenere circa il 74% di varianza cumulata.

	eigenvalue	variance.percent	cumulative.variance.percent
Dim.1	3.83158079	21.2865599	21.28656
Dim.2	2.38775786	13.2653215	34.55188
Dim.3	2.19625476	12.2014153	46.75330
Dim.4	1.96684907	10.9269393	57.68024
Dim.5	1.23864985	6.8813880	64.56162
Dim.6	0.95753884	5.3196602	69.88128
Dim.7	0.91050903	5.0583835	74.93967

Figura 6 Parametri ottenuti dalla PCA

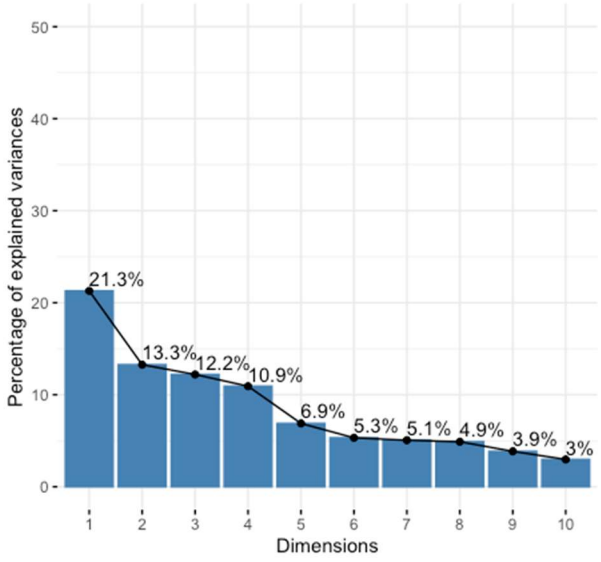

Figura 7 Dimensioni PCA in grafico

Notando il risultato prenderò i primi 7 valori in quanto ottengo circa il 74% di varianza cumulativa e sono soddisfatto con quei dati.

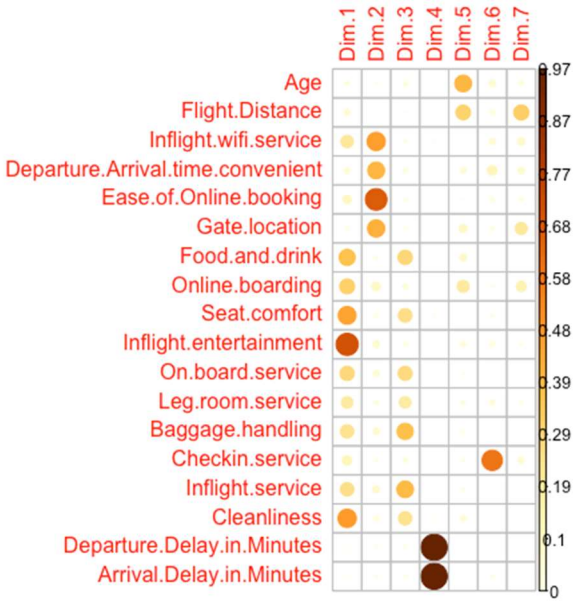

Figura 8 Cos2 di tutte le feature in corrispondenza della dimensione

Grazie a questo grafico è possibile visualizzare il cos2 delle variabili su tutte le dimensioni, in particolare è possibile osservare nella dimensione 1 e 2 gli attributi con valore maggiore, in quanto seguiranno successive analisi.

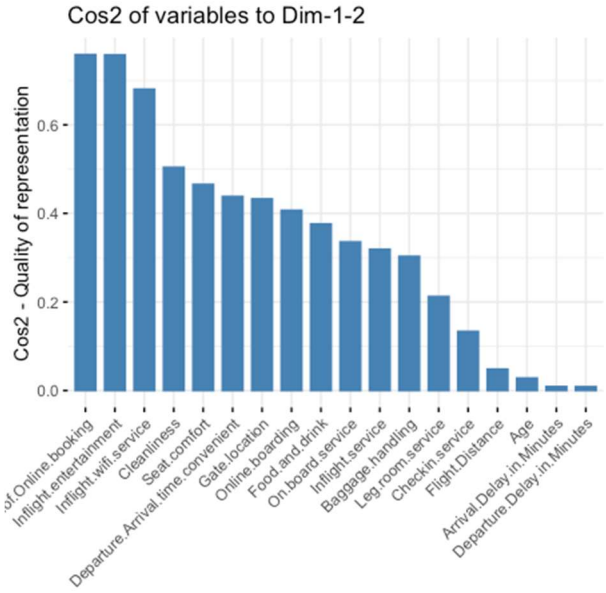

Figura 9 Cos2 delle features rispetto alla dimensione 1-2 della PCA

È anche possibile creare un grafico a barre sul cos2 delle variabili, come è possibile notare "Ease.of.Online.booking" risulta la feature che contribuisce maggiormente con la varianza, rispetto al totale nella dimensione 1-2, quindi che si distacca maggiormente dalla media. Infatti, osservando la *figura 9*, gli attributi in corrispondenza della colonna "Dim.1" e "Dim.2" con più alta varianza (di colore arancione) sono proprio gli stessi che presentano valori più importanti nell'istogramma.

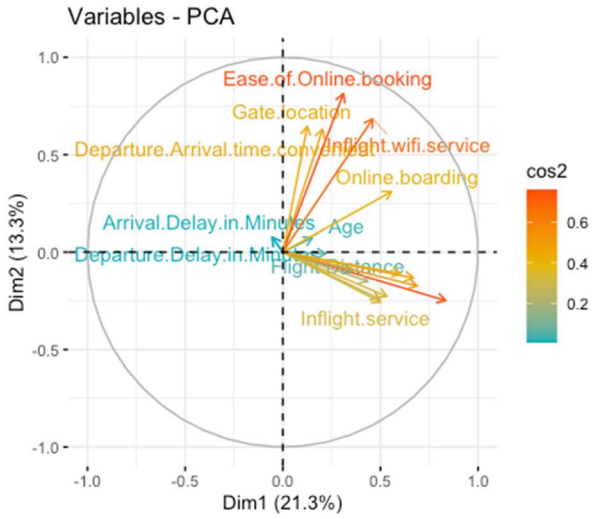

Figura 10 Cos2 rappresentata nel grafo

I medesimi attributi, con varianza più rilevante, è possibile osservarli anche in questo grafo, in particolare “Ease.of.Online.booking”, “Inflight.wifi.service” e “Inflight.entertainment” risultano i più rossastri, ovvero con cos2 maggiore.

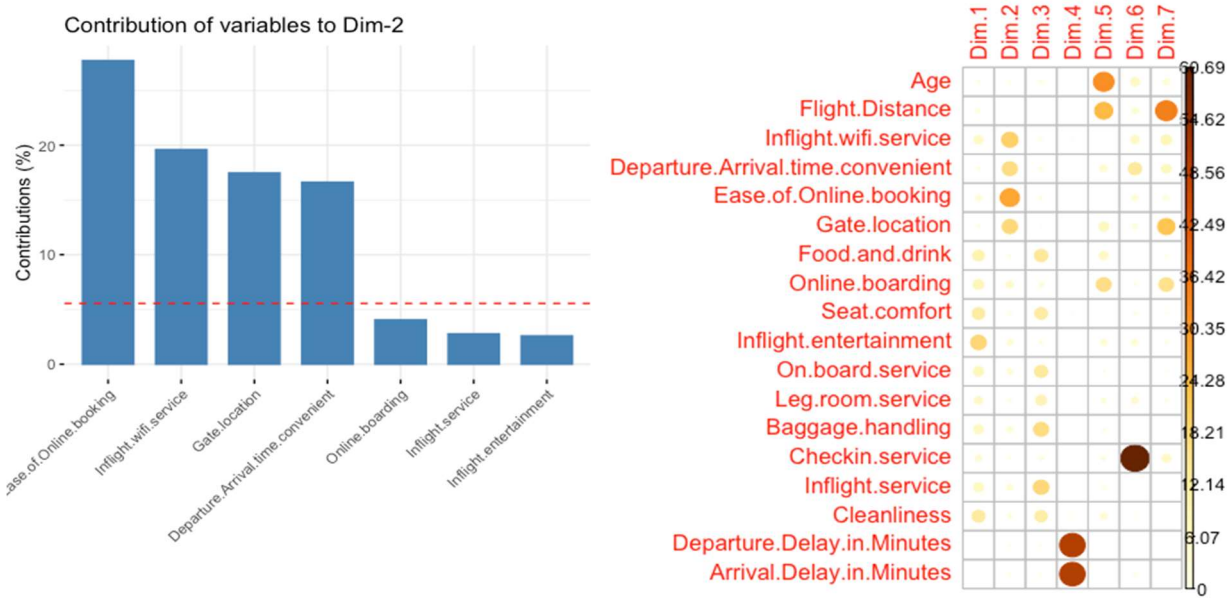

Figura 11 Contributo delle singole variabili nella dimensione 2 della PCA

A partire dal grafo a sinistra si può esaminare il contributo della variabile originale a una determinata componente principale, in questo caso nella dimensione 2, che come è possibile analizzare nell’immagine a destra la feature con contributo maggiore risulta essere “Ease.of.Online.booking”.

3. Modelli

Si è deciso di utilizzare 2 modelli di Machine Learning: Support Vector Machine e Neural Networks; sono stati scelti per la loro potenzialità e versatilità, successivamente addestrati tramite K-Fold Cross Validation con Grid Search (con k pari a 5).

3.1. Rete neurale

La rete neurale è un modello di ML, basato sulla composizione di più neuroni; ognuno dei quali effettua la sommatoria pesata dei propri input e ci applica una funzione. La rete impiegata ha una struttura di tipo deep: cioè con strati nascosti. Il primo strato (non computazionale) immette in input nella rete le feature, ed ha un numero di neuroni pari a quello delle variabili; il secondo strato, detto "hidden", effettua le prime computazioni tramite la funzione non lineare logistic; il terzo strato, composto da 2 neuroni con logistic, effettua le ultime computazioni e restituisce le probabilità di appartenenza alla classe binaria ("soddisfatto" o "non soddisfatto"). È stata scelta la funzione logistic sia perché restituisce delle probabilità, e quindi ideale per problemi di classificazione; sia perché, essendo non lineare, permette di separare le due classi anche in spazi non linearmente separabili.

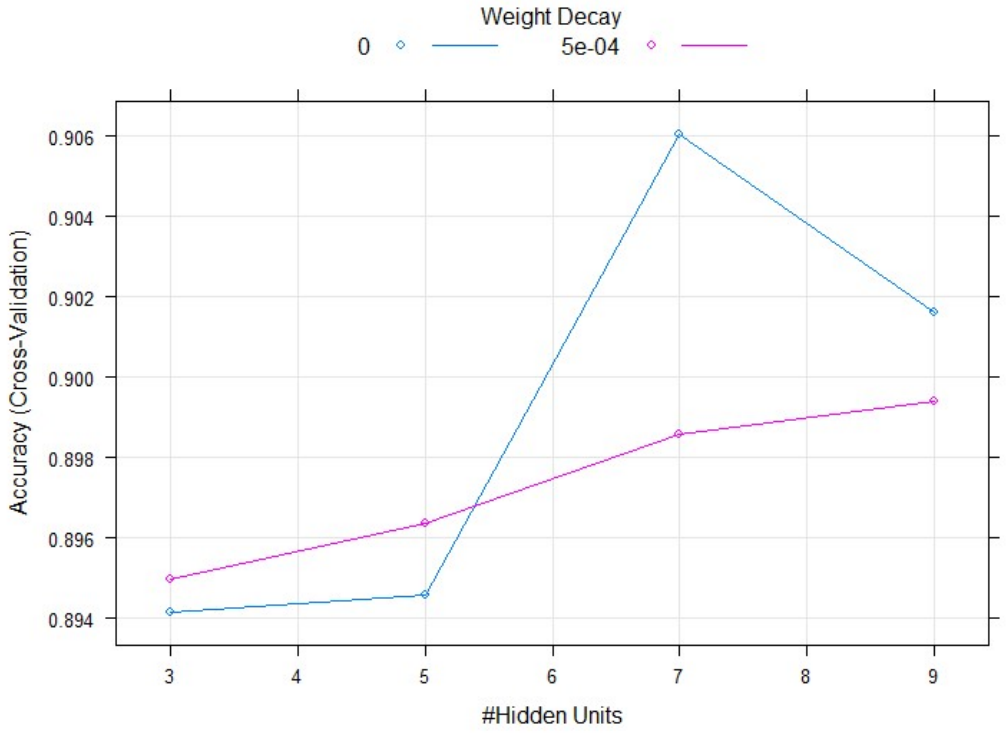

Figura 12 Grafo di Cross-validation di Neural Networks

La rete è stata addestrata con un approccio di K-Fold Cross Validation con Grid Search, con k pari a 5; questo permette di ottenere metriche di valutazione più affidabili, perché ottenute dalla media delle metriche sui k-fold; e ci consente di variare alcuni iperparametri per individuare i migliori (effettuando il cosiddetto Tuning). Gli iperparametri testati sono: il numero di neuroni nello strato nascosto, che variano da 3, 5, 7 e 9; e il decadimento dei

pesi, che varia tra 0 e 0.0005. Come si può vedere dal grafico la precisione maggiore si ottiene con 7 neuroni e 0 di Weight Decay; dunque, tale modello sarà quello ad essere poi adottato nella successiva fase di testing.

3.2. SVM

Svm è un tipo di modello di classificazione che utilizza i vettori di supporto per costruire una decisione di separazione della superficie tra le diverse classi. La scelta della superficie di separazione viene effettuata in modo da massimizzare la distanza tra le classi, il che significa che il modello cerca di trovare la decisione di separazione che ha il maggior margine possibile. Uno dei vantaggi è la sua capacità di effettuare una separazione non lineare usando un kernel appropriato.

Un parametro importante che abbiamo deciso di variare è il costo che ci consente di penalizzare le istanze interne al margine, migliorando la robustezza del modello rispetto al rumore nei dati.
Mentre per il parametro kernel, abbiamo scelto di usare il Radial Basis Fuction (RBF); in quanto, oltre ad essere uno dei più usati ed efficienti, permette di effettuare classificazioni con diversi gradi di separabilità.

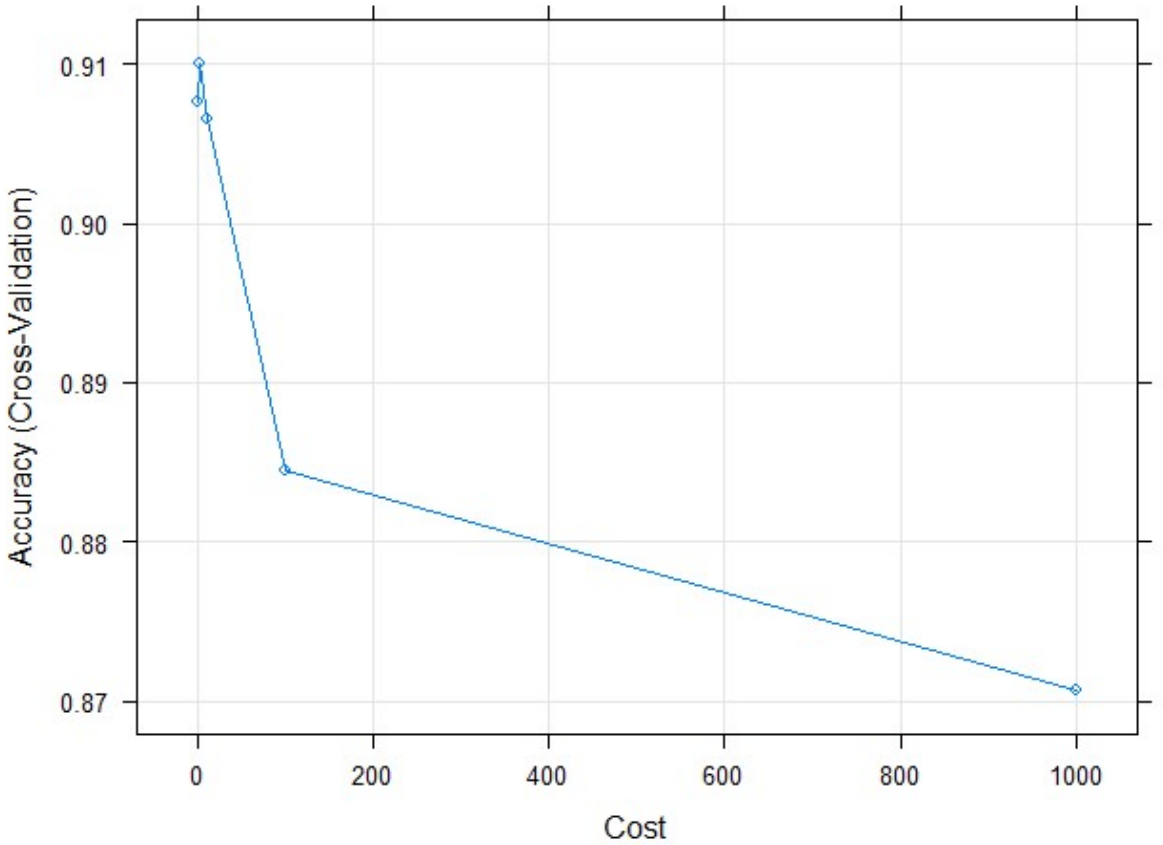

Figura 13 Grafo di Cross-validation di SVM

Questo grafo descrive la K-Fold Cross Validation SVM che varia sul costo (da 1 a 1000)

Si è deciso di fissare alcuni punti: 0.5, 1, 10, 100, 1000; evidenziati da dei pallini nel grafico. È possibile notare come all'aumentare del costo diminuisce l'accuratezza, infatti evitando di considerare dei dati, si andrebbero a sbilanciare le classi, così diminuendo l'accuracy.
In particolare, dal grafico si analizza che il valore di costo = 1 ha portato dei risultati particolarmente buoni, ovvero il valore ottimale, dopodiché l'accuratezza diminuisce.

4. Risultati

I risultati ottenuti sono stati calcolati sui i dati di test. I quali sono stati estratti dal relativo dataset, prendendo un campione casuale di 500, per poi essere processati in modo simile ai dati di train.

4.1. Confusion matrix

Dalle matrici di confusione è possibile osservare una visualizzazione dei risultati ottenuti dai modelli, in particolare analizzeremo *Precision, Recall* e *F-measure*.

4.1.1. Rete Neurale

```
                Reference
Prediction     satisfied dissatisfied
  satisfied          199           21
  dissatisfied        24          255

               Accuracy : 0.9098
                 95% CI : (0.8812, 0.9335)
    No Information Rate : 0.5531
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.8173

 Mcnemar's Test P-Value : 0.7656

            Sensitivity : 0.8924
            Specificity : 0.9239
         Pos Pred Value : 0.9045
         Neg Pred Value : 0.9140
              Precision : 0.9045
                 Recall : 0.8924
                     F1 : 0.8984
             Prevalence : 0.4469
         Detection Rate : 0.3988
   Detection Prevalence : 0.4409
      Balanced Accuracy : 0.9081

       'Positive' Class : satisfied
```

Figure 3: Confusion Matrix, per la rete neurale, delle previsioni sul test set

Come si può notare l'accuracy ottenuta (del 90%) è ottima rispetto al No Information Rate del 55%; cioè, significa che il modello è decisamente più preciso rispetto alla predizione Naive (che, per ogni istanza, si limita a restituire la classe più frequente, in quanto più probabile).

Il risultato di decision è molto soddisfacente, il modello è riuscito a classificare giustamente le osservazioni di quella classe quando ha detto che appartenevano a quella classe; infatti, ha una precisione del 90% per la classe "satified", cioè ha classificato correttamente l'90% delle osservazioni che ha detto appartenere alla classe "satified".

Anche il risultato di recall risulta molto buono, il modello ha un recall del 89% per la classe "satisfied", cioè significa che ha classificato correttamente l'89% delle osservazioni che appartengono effettivamente alla classe "satisfied".

4.1.2. SVM

```
                Reference
Prediction     satisfied dissatisfied
  satisfied          200           15
  dissatisfied        23          261

               Accuracy : 0.9238
                 95% CI : (0.897, 0.9455)
    No Information Rate : 0.5531
    P-Value [Acc > NIR] : <2e-16

                  Kappa : 0.8454

 Mcnemar's Test P-Value : 0.2561

            Sensitivity : 0.8969
            Specificity : 0.9457
         Pos Pred Value : 0.9302
         Neg Pred Value : 0.9190
              Precision : 0.9302
                 Recall : 0.8969
                     F1 : 0.9132
             Prevalence : 0.4469
         Detection Rate : 0.4008
   Detection Prevalence : 0.4309
      Balanced Accuracy : 0.9213

       'Positive' Class : satisfied
```

Figure 4: Confusion Matrix, per SVM, delle previsioni sul test set

Come si può notare l'accuracy ottenuta (del 92%) è ottima rispetto al No Information Rate del 55%; cioè, significa che il modello è decisamente più preciso rispetto alla predizione Naive (che, per ogni istanza, si limita a restituire la classe più frequente, in quanto più probabile).

Il risultato di decision è molto soddisfacente, il modello è riuscito a classificare giustamente le osservazioni di quella classe quando ha detto che appartenevano a quella classe; infatti, ha una precisione del 93% per la classe "satified", cioè ha classificato correttamente l'93% delle osservazioni che ha detto appartenere alla classe "satified".
Anche il risultato di recall risulta molto buono, il modello ha un recall del 89% per la classe "satisfied", cioè significa che ha classificato correttamente l'89% delle osservazioni che appartengono effettivamente alla classe "satisfied".
È interessante confrontare i risultati rispetto all'altro modello di machine learning scelto, ovvero Neural Network, i dati risultano veramente molto simili, questo indica aver scelto un buon dataset di dati e le features più importanti per l'addestramento e la predizione.

4.2. Roc Auc

Una curva ROC traccia TPR (Tasso di positività reale) rispetto a FPR (tasso di falsi positivi) a soglie di classificazione diverse. Abbassando la soglia di classificazione, vengono classificati più elementi positivi, aumentando così i falsi positivi e i veri positivi.

4.2.1. Rete neurale

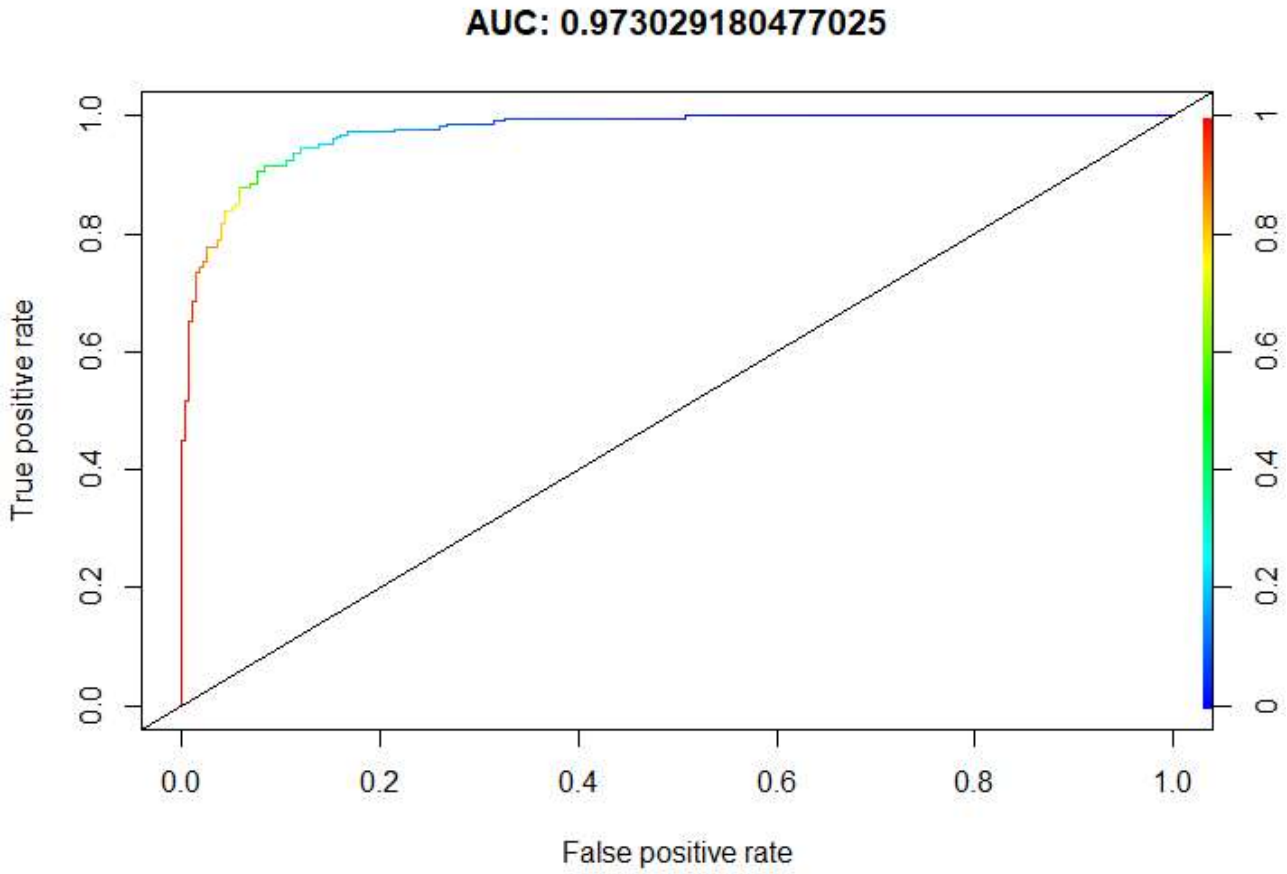

Figure 5: Grafico che mostra le performance della Rete Neurale sul test set

Il grafico presentato sopra permette di mostrare la capacità di classificazione del modello Neural Network di fare distinzioni tra le due classi. In questo caso il valore è molto vicino ad 1 e risulta soddisfacente.

4.2.2. SVM

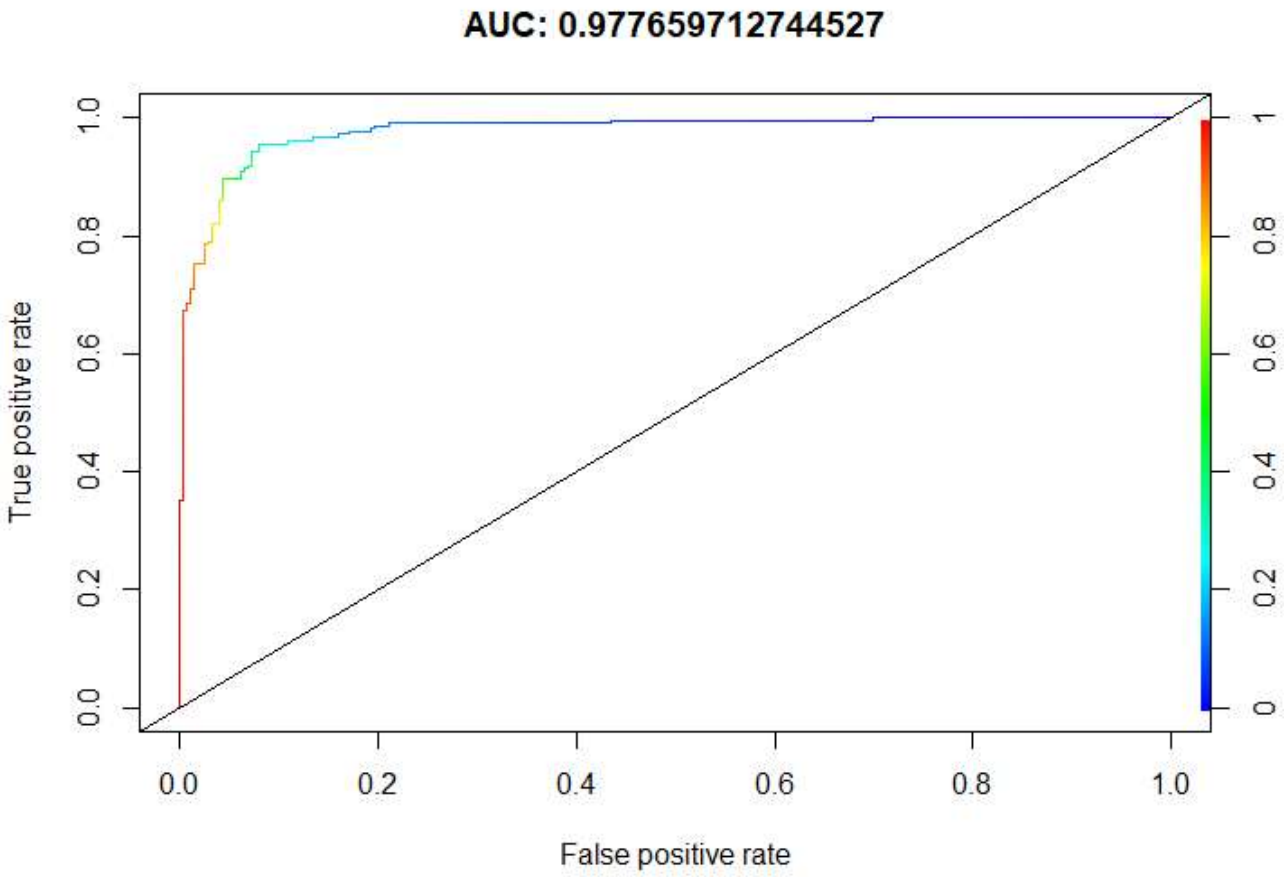

Figure 6: Grafico che mostra le performance di SVM sul test set

Il grafico presentato sopra permette di mostrare la capacità di classificazione del modello SVM di fare distinzioni tra le due classi. In questo caso il valore è molto vicino ad 1 e risulta soddisfacente.

Indice delle figure