

Relazione Progetto Big Data e Business Intelligence

Ferrara Luca Matricola: 312173

July 22, 2022

Contents

1	Introduzione	3
1.1	Dataset	3
2	Data Visualization	4
3	Data exploration	5
3.1	Feature Selection	6
3.2	Creazione Test e Training Set	6
3.3	Feature Scaling	6
4	Comparazione modelli	6
5	Random Forest Classifier	7
5.1	Fine Tuning	7
5.2	Testing Phase	8
6	Artificial Neural Network	9
6.1	Fine Tuning	9
6.2	Testing phase	9
7	Conclusioni	10

1 Introduzione

Il progetto assegnato prevede di gestire un modello di Machine Learning per fare un task di classificazione relativo al dataset: <https://www.kaggle.com/datasets/jimschacko/airlines-dataset-to-predict-a-delay>;

Il dataset riguarda un numero di voli americani e il task è di predire se il volo avrà un ritardo o no. Ecco come è composto il dataset:

1.1 Dataset

Il dataset è composto da 8 feature e 539383 istanze:

- id
Feature non utile che verrà rimossa, identificativo per il volo;
- Airline
Indica la compagnia aerea;
- Flight
Indica il tipo di volo;
- AirportFrom
Indica l'aeroporto da dove parte il volo;
- AirportTo
Indica l'aeroporto dove atterra il volo;
- DayOfWeek
Giorno della settimana;
- Time
Tempo del volo;
- Length
Lunghezza del volo;
- Delay
Indica se il volo è in ritardo (0 no, 1 si);

2 Data Visualization

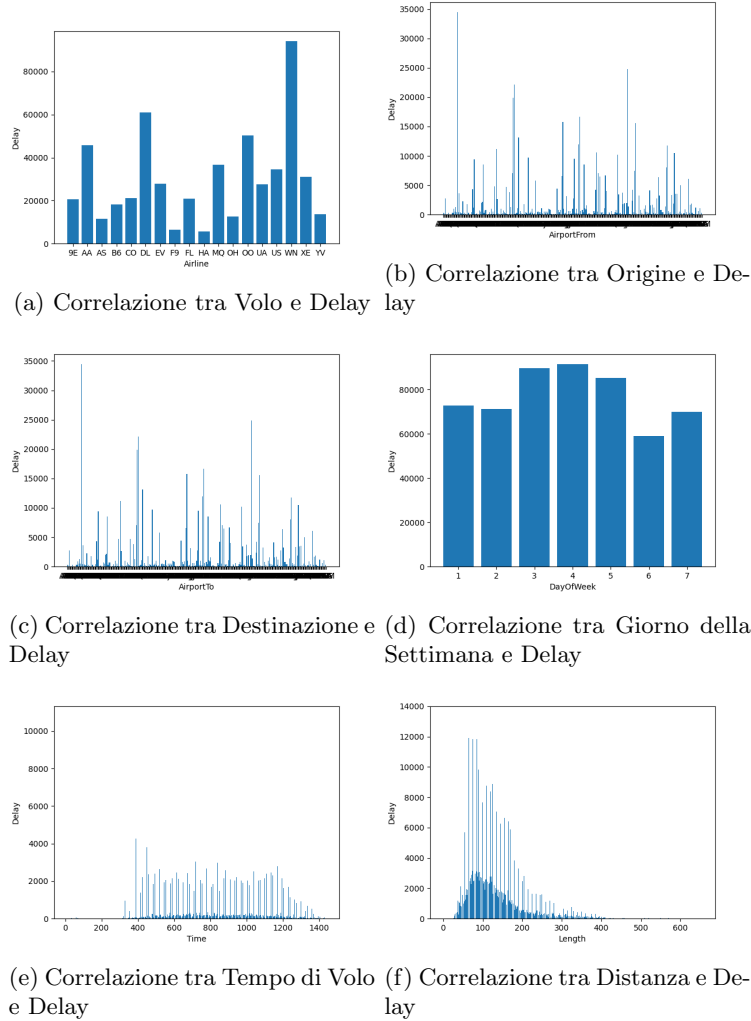


Figure 1: Alcuni grafici

Dopo aver visto studiato questi dati, andiamo a vedere la frequenza del label con un Pie Chart.

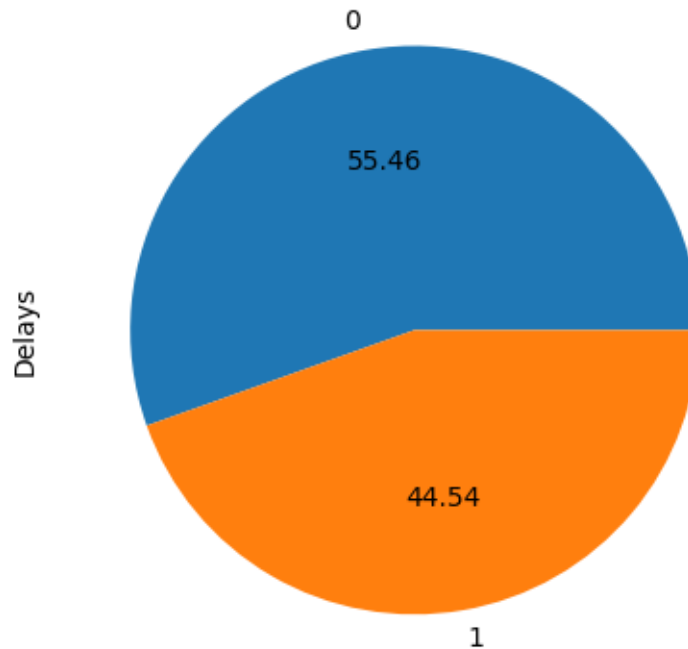


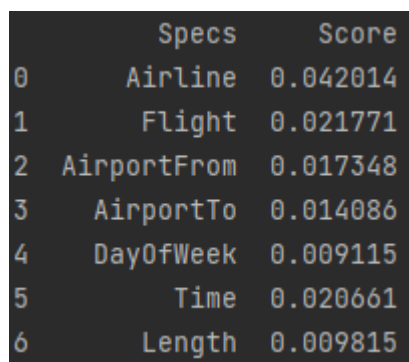
Figure 2: Frequenza del Label delay

Vediamo quindi, che il dataset è sbilanciato; andremo a modificarlo tramite l'utilizzo di oversampling.

3 Data exploration

Dopo aver visualizzato graficamente il dataset, andremo a fargli delle modifiche per renderlo più accessibile, e utilizzabile dai modelli di Machine Learning. Prima di tutto rimuoviamo la feature 'id', non utile per il task; andiamo poi a fare l'encoding delle feature categoriche, trasformando le colonne di esse in colonne di interi ordinali, tramite l'utilizzo di OrdinalEncoder. Non ho bisogno

di fare label encoding perchè è già corretto così; tramite l'utilizzo dello z-score, andiamo a rimuovere gli outliers (più o meno 9000, quelli con z-score > 3). Con l'uso di Mutual Information andiamo a verificare la correlazione tra le feature e il label:



	Specs	Score
0	Airline	0.042014
1	Flight	0.021771
2	AirportFrom	0.017348
3	AirportTo	0.014086
4	DayOfWeek	0.009115
5	Time	0.020661
6	Length	0.009815

Figure 3: Risultati MutualInformation con SelectKBest

3.1 Feature Selection

Con la funzione appena usata, andiamo a rimuovere le due feature con score molto basso, DayOfWeek e Length

3.2 Creazione Test e Training Set

Andiamo poi a fare lo splitting del Dataset in Training e Test set (rispettivamente 80% e 20%); dopo aver fatto ciò è necessario fare oversampling sul training set per bilanciarlo, utilizzando SMOTE

3.3 Feature Scaling

Andiamo in fine a fare feature scaling, per aiutare la discesa del gradiente, con l'utilizzo della normalizzazione Min Max (chiaramente con riferimento al training set, ed applicata successivamente al test set)

4 Comparazione modelli

Successivamente al preprocessing, è necessario dover scegliere un modello per eseguire il task di classificazione; andrò ad utilizzare i modelli visti a lezione:

- Logistic Regression
- Decision Tree Classifier

- Random Forest Classifier
- AdaBoost Classifier
- Gradient Boosting Classifier
- XGB Classifier

Con l'utilizzo della funzione `cross_val_score` andiamo quindi a provare ogni modello con `StratifiedKFoldValidation` con $K = 10$; la funzione ritorna come parametro di comparazione l'accuracy, sarà quindi quella che andremo ad usare. Ecco i risultati (prima senza rimozione feature, dopo con):

```

Regression logistica -> accuracy: 0.5814851312109315
Decision tree -> accuracy: 0.6262354514733708
Random Forest -> accuracy: 0.6422597219004753
AdaBoost -> accuracy: 0.6266432156850903
GradientBoosting -> accuracy: 0.6362879446030302
XGB -> accuracy: 0.6588574556292891

```

(a) Prima di rimuovere `DayOfWeek` e `Length`

```

Regression logistica -> accuracy: 0.5758185503249127
Decision tree -> accuracy: 0.6636097703671201
Random Forest -> accuracy: 0.6665475668108403
AdaBoost -> accuracy: 0.62400175515577
GradientBoosting -> accuracy: 0.6321318874192943
XGB -> accuracy: 0.6529764516600849

```

(b) Dopo averli rimossi

Figure 4: Risultati Valutazione

Il modello utilizzato sarà quindi `RandomForestClassifier`.

5 Random Forest Classifier

5.1 Fine Tuning

Per andare a fare Fine Tuning degli hyperparametri di Random Forest andremo a usare per prima cosa una funzione di Random Search con `RandomizedSearchCV`, e poi `GridSearch` sui parametri nel range dei parametri trovati da Random Search, con l'uso di `GridSearchCV`. Purtroppo l'operazione di Grid Search fatta bene è troppo onerosa per la mia postazione, ma il codice è pronto per eseguirla.

```
Best Params {'n_estimators': 100, 'min_samples_split': 7, 'min_samples_leaf': 5, 'max_features': 7, 'max_depth': 19, 'bootstrap': True}
Best Score: 0.6606134687310642
```

(a) Risultati RandomSearch

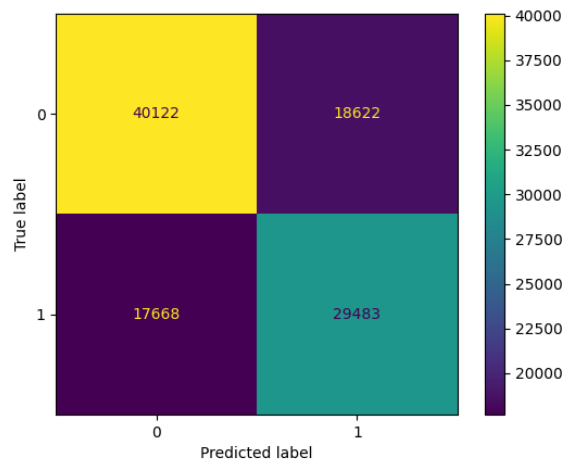
Figure 5

5.2 Testing Phase

Andiamo quindi a testare il modello con i parametri ottenuti, ecco di seguito i risultati:

```
Accuracy: 0.6573020444780207
Precision: 0.6128884731316911
F1: 0.6190266229948769
[[40122 18622]
 [17668 29483]]
```

(a)



(b)

Figure 6: Risultati

Con un GridSearch più approfondito probabilmente i risultati sarebbero migliorati.

6 Artificial Neural Network

Visto che a lezione abbiamo introdotto le reti neurali, ho deciso di creare un piccolo modello, per fare la stessa task di classificazione sullo stesso dataset (già preprocessato), con l'uso di Keras.

6.1 Fine Tuning

Anche qui andremo a fare Fine Tuning, facendo RandomSearch e GridSearch, con l'ausilio di scikeras, libreria che permette di usare gli strumenti di sklearn con Keras. Ecco i risultati del RandomizedSearch:

```
batch_size=34, .epochs=18, .optimizer=RMSprop;, .score=0.600
```

(a) Risultati RandomSearch

Figure 7

6.2 Testing phase

Vado infine a testare la rete con i parametri trovati:

```
loss: 0.6653 - accuracy: 0.5898
```

(a) Risultati RandomSearch

Figure 8

7 Conclusioni

Nonostante un'attenta fase di studio del dataset, e il suo preprocessing, il modello da me scelto e configurato appositamente per l'utilizzo, da dei risultati che non mi soddisfano pienamente: probabilmente la non esecuzione del Grid Search e il basso score di Mutual Information sono alcune delle ragioni del risultato.