PROJECT AIRLINE SATISFACTION

start2impact UNIVERSITY

AUTHOR:
GIACOMO ABRAMO

THE DATASET
(DESCRIPTION)

The Dataset contains 24 features which should help us to predict whether a passenger will be generally satisfied or neutral/dissatisfied about flight:

1) Unnamed: 0
2) Id
3) Gender
4) Customer Type
5) Age
6) Type of Travel
7) Class
8) Flight Distance
9) Inflight wifi service
10) Departure/Arrival time convenient
11) Ease of Online booking
12) Gate location
13) Food and drink
14) Online boarding
15) Seat comfort
16) Inflight entertainment
17) On-board service
18) Leg room service
19) Baggage handling
20) Checkin service
21) Inflight service
22) Cleanliness
23) Departure Delay in Minutes
24) Arrival Delay in Minutes

25) satisfaction

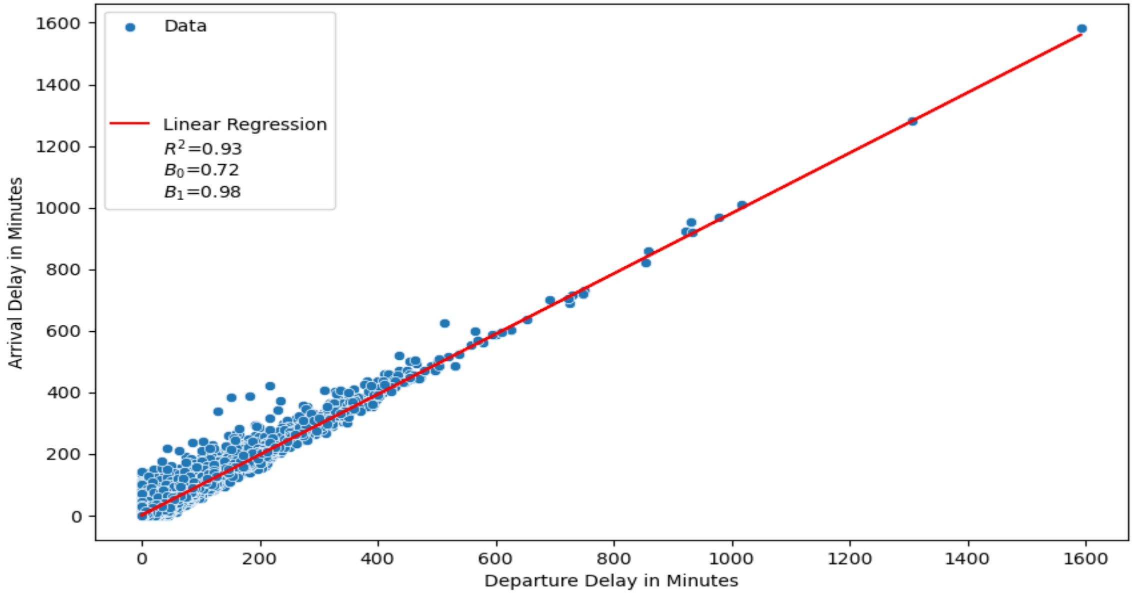
THE DATASET
(PRELIMINARY ACTIVITIES FOR BASELINE MODELS)

By carrying out an initial superficial analysis of the features available to us individually, I immediately realize that:

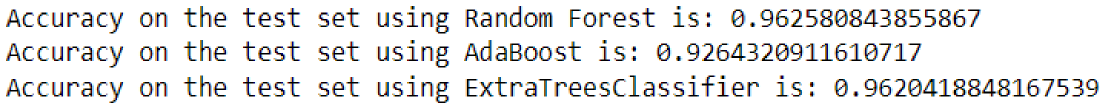
1) "Unnamed: 0" variable is a simple column with numerical values that repeats the row indexes that can be extracted from the dataset; it is an useless variable.

2) "id" variable is a simple passenger identifier. For this work it is a useless variable since in fact it has no predictive power.

3) "Arrival Delay in Minutes" variable contains missing value. Given that it is strongly linear correlated to the "Departure Delay in Minutes" variable, I used imputation based on linear regression (see graph below).



In order to use the algorithms (Random Forest, AdaBoost and ExtraTrees) I used label encoding and ordinal encoding for the variables that assumes string type values. Given that the target variable is balanced I used the accuracy as evaluation metric.

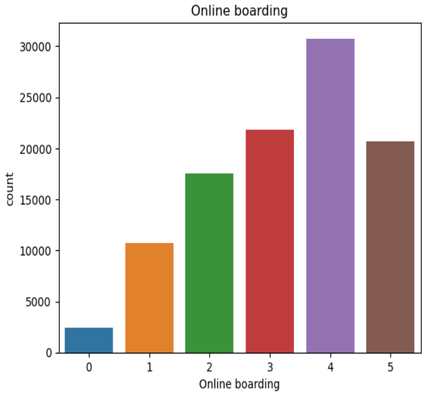


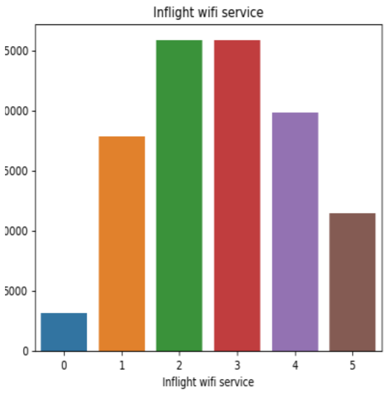
```
Accuracy on the test set using Random Forest is: 0.962580843855867
Accuracy on the test set using AdaBoost is: 0.9264320911610717
Accuracy on the test set using ExtraTreesClassifier is: 0.9620418848167539
```

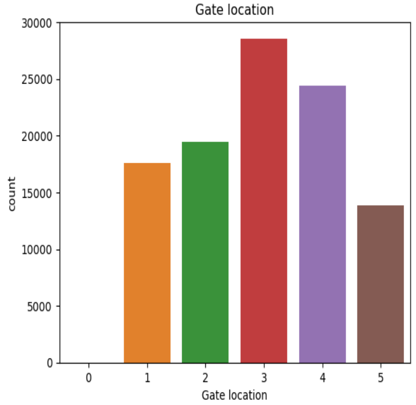
EXPLORATORY DATA ANALYSIS
(UNIVARIATE AND BIVARIATE WITH FOCUS ON "SATISFACTION ")

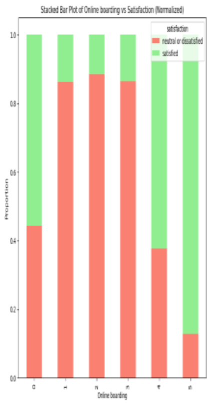
By carrying out EDA from an univariate point of view I was able to better examine the distribution of the variables. The most important insight I noticed was the presence of "strange" values not so much in continuous variables (since some values could actually make sense, although they could be considered outliers from a statistical point of view) but in almost all categorical variables associated with a satisfaction context; they have some (few, and this could be a clue) "0" values (see the first three graphs). In the absence of information on this matter we could deduce that it is a coding for the absence of information and therefore a missing value (NaN).

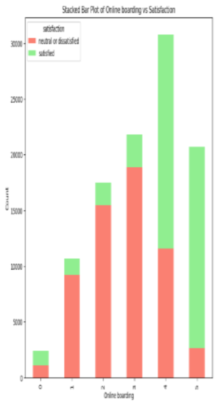
By carrying out EDA from a bivariate point of view with focus on "satisfaction" I was able to better examine the relationships between features and satisfaction. For some variables (such as Gender, Customer Type and Class) the relationship was as expected; in relation to the categorical variables associated with a satisfaction context, the bivariate EDA seems to confirm the doubts expressed in the univariate EDA (see the last three graphs) as the satisfied and neutral/dissatisfied proportion is not what would be expected from passengers if they had given a strongly negative evaluation to a specific flight service.

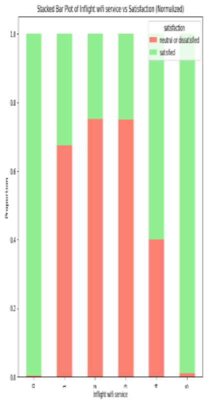


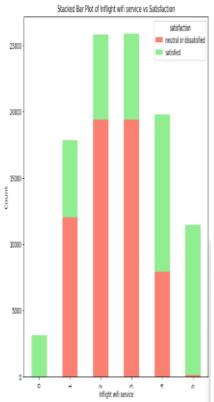


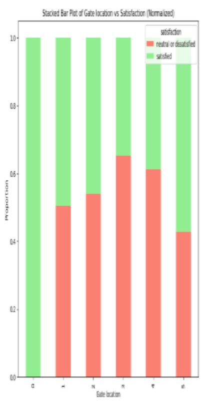


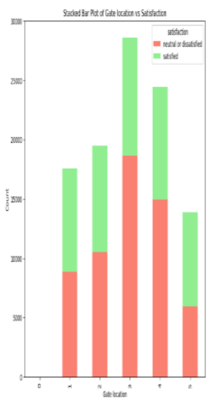










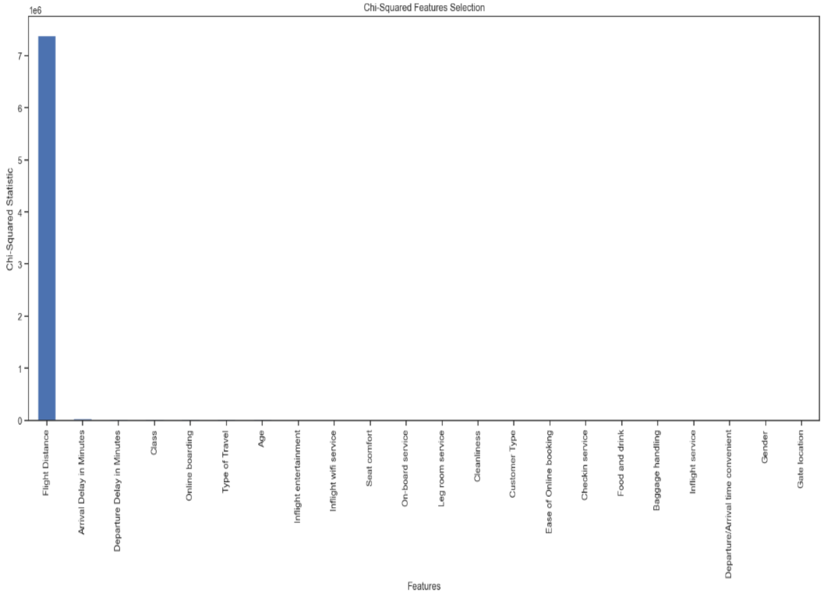


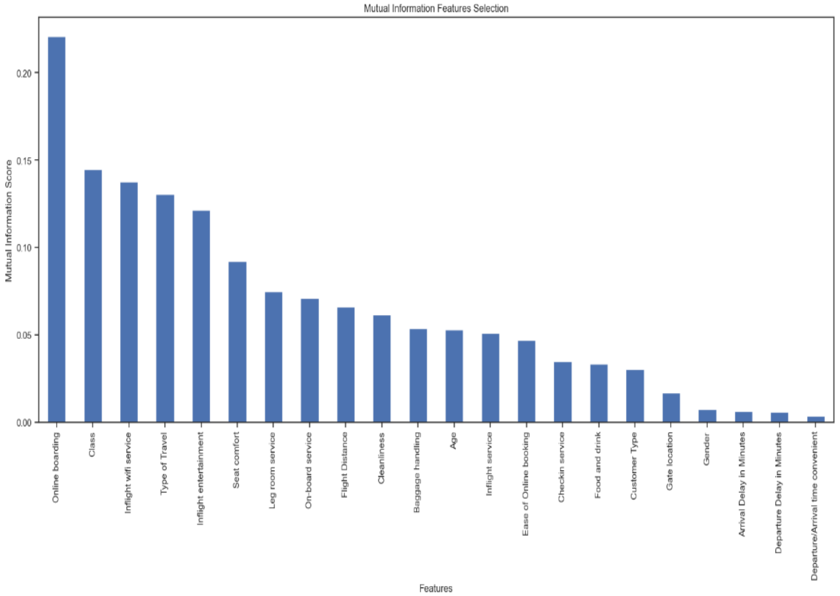
BEST FEATURES
(CHI-SQUARED STATS / ANOVA F-VALUE / MUTUAL INFORMATION)

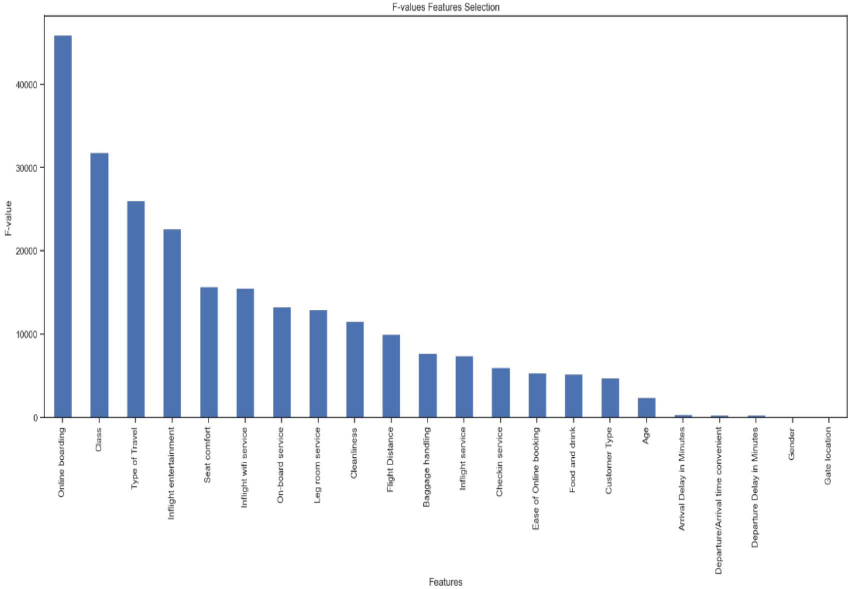
By carrying out analysis in order to discover the best features help us to predict satisfaction I used three function of the module “feature_selection” of “sklearn” library (see the three graphs).

The last two methods have similar results; the first one presents different results.

I took into account the 4 most important variables (“Type of Travel”, “Class”, “Online boarding”, “Inflight entertainment”) to check whether the models perform better considering all the features or only the most important ones.





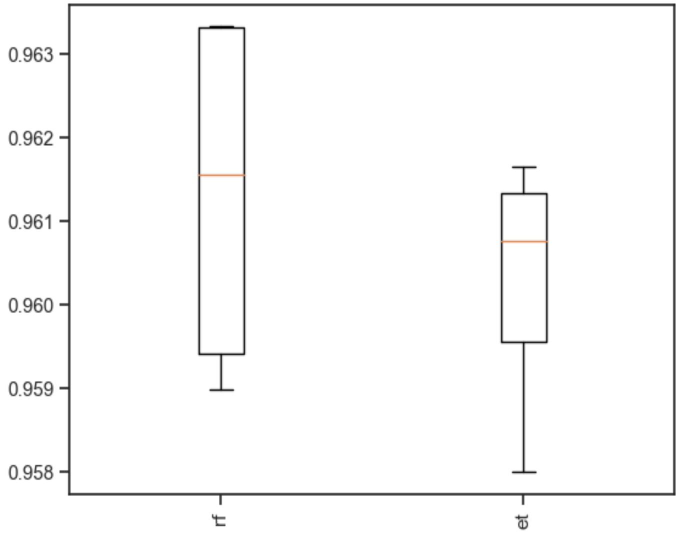


SPOT-CHECK AND HYPERPARAMETER TUNING

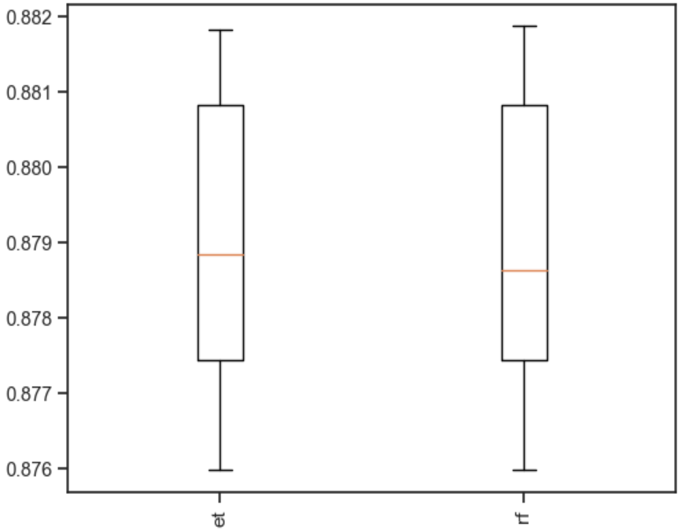
By carrying out a spot-check I verified, both considering all the features and only the four most important ones (“Type of Travel”, “Class”, “Online boarding”, “Inflight entertainment”), which were the two models (with the default values of the hyperparameters) best among the three used. In both scenarios the best models were ExtraTrees and Random Forest (see the first two graphs).

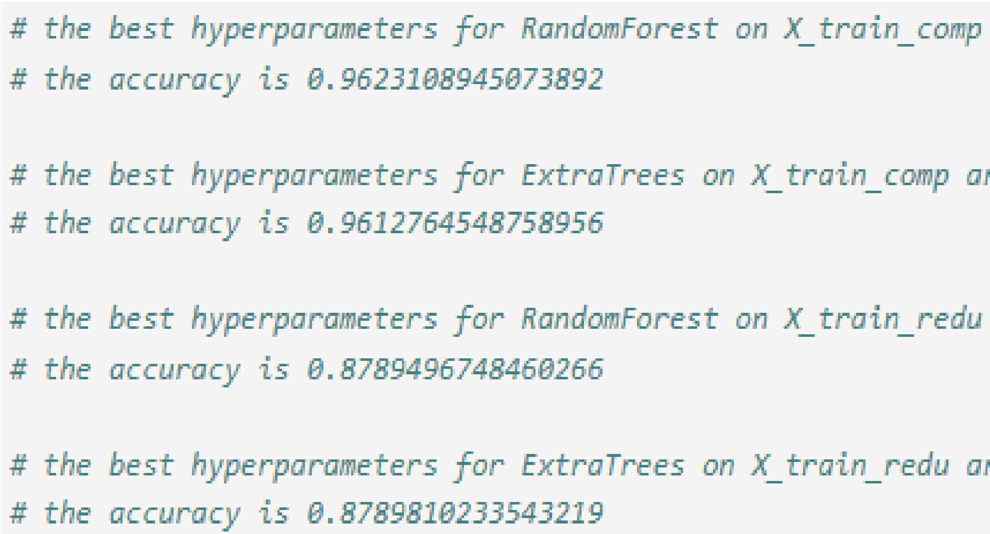
By carrying out hyperparameter tuning on the four models (ExtraTrees and Random Forest considering all the features and only the four most important ones), the best model was the Random Forest in combination with all the features (see the last gragh).

```
Rank=1, Name=rf, Score=0.961 (+/- 0.002)
Rank=2, Name=et, Score=0.960 (+/- 0.001)
```



```
Rank=1, Name=et, Score=0.879 (+/- 0.002)
Rank=2, Name=rf, Score=0.879 (+/- 0.002)
```





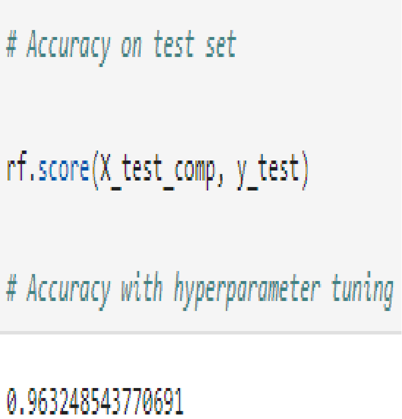
```
# the best hyperparameters for RandomForest on X_train_comp
# the accuracy is 0.9623108945073892

# the best hyperparameters for ExtraTrees on X_train_comp a
# the accuracy is 0.9612764548758956

# the best hyperparameters for RandomForest on X_train_redu
# the accuracy is 0.8789496748460266

# the best hyperparameters for ExtraTrees on X_train_redu a
# the accuracy is 0.8789810233543219
```

BEST MODEL / FEATURE IMPORTANCE / ERROR ANALYSIS

Once I found the best model I performed the fit (with the best hyperparameters) on the entire train set and evaluated it on the test set (see the first graph).

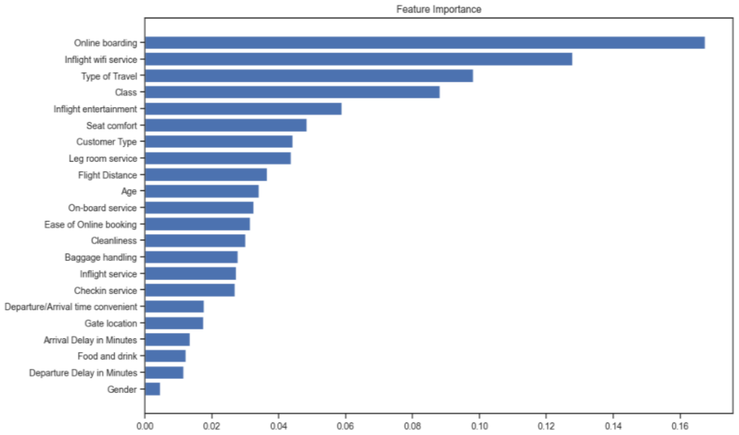
At the end of the fit phase it was possible to exploit the “feature_importances_” attribute to check again which variables had the most impact. The results are very similar to those obtained from previous analyses (see the second graph).

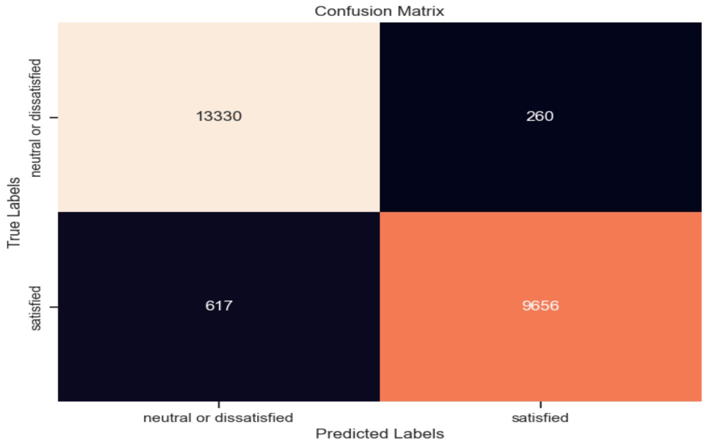
I subsequently carried out an error analysis by plotting the confusion matrix and the classification report (see the last two graphs).

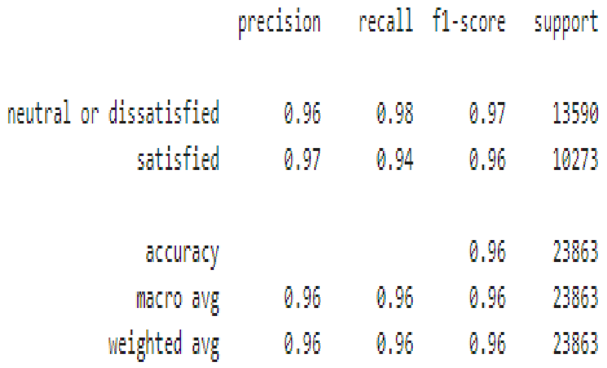


```
# Accuracy on test set

rf.score(X_test_comp, y_test)

# Accuracy with hyperparameter tuning
```

```
0.963248543770691
```





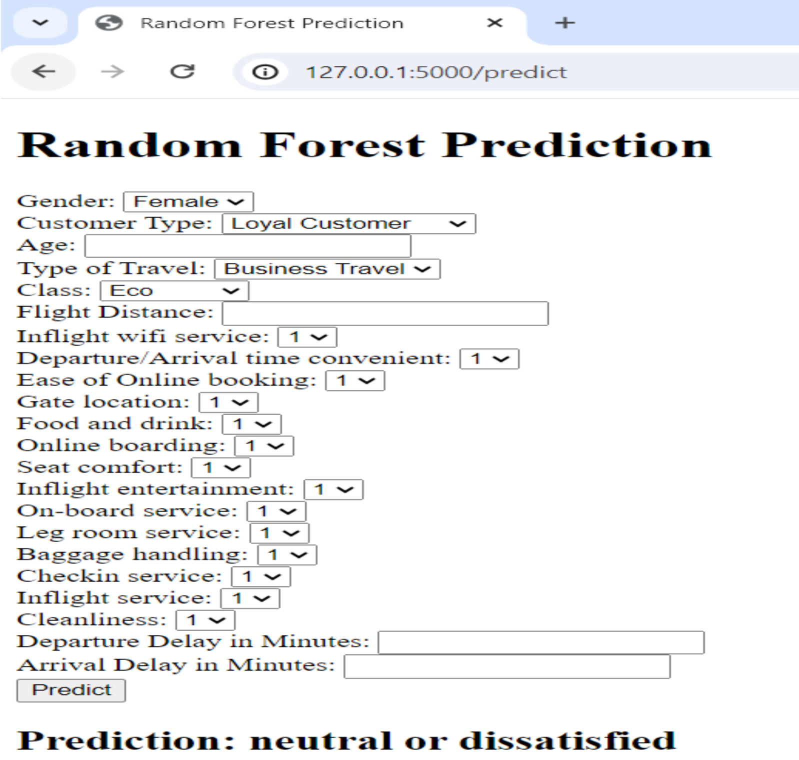


```
                         precision    recall  f1-score   support

neutral or dissatisfied       0.96      0.98      0.97     13590
              satisfied       0.97      0.94      0.96     10273

               accuracy                           0.96     23863
              macro avg       0.96      0.96      0.96     23863
           weighted avg       0.96      0.96      0.96     23863
```

WEB SERVICE FOR PREDICTION AND CONCLUSIONS

I created a web service via flask useful for making predictions given specific values for features.

Following the analyzes carried out we can say that the features available proved to be rather good for the creation of a predictive model; a small (?) flaw that I think is present in the data is instead linked to the target variable; this variable takes only the values "satisfied" and "neutral or dissatisfied".



Random Forest Prediction

Gender: Female
Customer Type: Loyal Customer
Age:
Type of Travel: Business Travel
Class: Eco
Flight Distance:
Inflight wifi service: 1
Departure/Arrival time convenient: 1
Ease of Online booking: 1
Gate location: 1
Food and drink: 1
Online boarding: 1
Seat comfort: 1
Inflight entertainment: 1
On-board service: 1
Leg room service: 1
Baggage handling: 1
Checkin service: 1
Inflight service: 1
Cleanliness: 1
Departure Delay in Minutes:
Arrival Delay in Minutes:
Predict

Prediction: neutral or dissatisfied

I think it would have been more appropriate, during the data collection phase, to distinguish between "satisfied", "neutral" and "dissatisfied" in order to have 3 distinct classes in the training phase. In my opinion, this subdivision would have improved data-driven decisions depending on the decisions to be made.

All the algorithms in general, even from an empirical point of view, seem quite powerful and efficient.
The data + algorithm combination has had excellent results since the baseline models. The EDA and Hyperparameter tuning part (the pre-processing phase has in fact remained unchanged compared to the baseline models) led more to a waste of computation and time as the improvements seem to have been quite imperceptible.