Flight Rate Prediction

Regression Problem

[#]*DONE BY: VEDASAMHITHA CHALLAPALLI*

B20CS078

***Abstract*— The paper reports my experience and efforts in dealing the dataset of flight rate prediction which is basically a machine learning regression problem. I have basically used 3 regression models along with other essential Machine Learning concepts. The ultimate goal of taking up the project is to suggest the best model which predicts the flight rates given various inputs about the travel.**

The paper contains following subparts — **Introduction (Dataset), Methodology (models used, concepts used), Exploring and cleaning the dataset, Implementation of regression algorithms, evaluation of models, results and analysis, conclusion, references**

I. INTRODUCTION

One of the applications of Machine Learning lies in the Aviation Industry and various different principles can be used to solve the problem of "Flight Rate Prediction" using many regression techniques. Thus, I found the project very interesting and started to work on the dataset….

A. *DATASET:*

The dataset consists of 11 columns, the last one being the "Price" column which is the target column here. The other columns include "Airline", "Date_of_journey", "Source", "Destination", "Route", "Dep_time", "Arrival_time", "Duration", "Total_Stops", "Additional_Info" and "Price"

II. METHODOLOGY

A. *MODELS USED:*

- Random Forest Regressor
- XGBoost
- Logistic Regression

B. *CONCEPTS USED:*

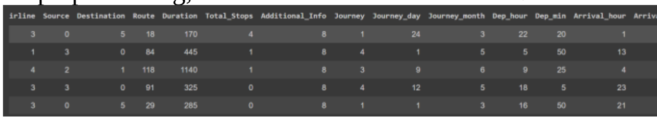
- Preprocessing
- Feature Engineering
- Feature Scaling
- Exploratory Data Analysis
- Model Training
- Hyperparameter Tuning
- Sequential Feature Selection
- Performance measures

III. EXPLORING AND CLEANING THE DATASET

The dataset has 10863 rows and 11 columns. Out of all the columns, 9 columns are of object data type and 1 column is of integer data type which is the target column: "Price".

A. *PREPROCESSING:*

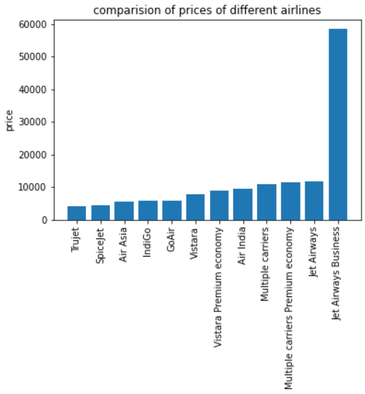
- Empty data has been replaced by mean of the column.
- Duplicate data has been removed.
- "Date_of_Journey" column has been modified such that the date and month of the date are separately added to the data as integers.
- The arrival and departure times have been changed to hours and minutes and are separately added to the data as integers.
- Duration time has also been changed to minutes and is added to the data
- Label encoding has been done to the categorical columns.
- After preprocessing, the head of the data looks like:



irline	Source	Destination	Route	Duration	Total_Stops	Additional_Info	Journey	Journey_day	Journey_month	Dep_hour	Dep_min	Arrival_hour	Arriv
3	0	5	18	170	4	8	1	24	3	22	20	1	
1	3	0	84	445	1	8	4	1	5	5	50	13	
4	2	1	118	1140	1	8	3	9	6	9	25	4	
3	3	0	91	325	0	8	4	12	5	18	5	23	
3	0	5	29	285	0	8	1	1	3	16	50	21	

B. *EXPLORATORY DATA ANALYSIS:*

(all graphs are there in the colab file)

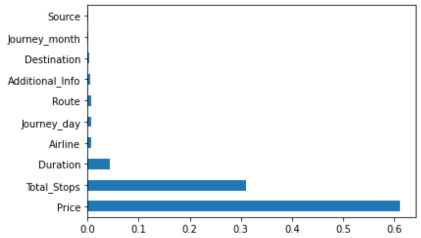


CONCLUSIONS BASED ON EDA GRAPHS:

- ➔Jet Airways Business has the costliest flight ticket
- ➔Number of stops cannot decide the flight ticket price alone
- ➔Ticket price is high when the source is Bangalore.
- ➔Generally, flights arriving to New Delhi are costlier
- ➔More the duration, more is the ticket price

➔From the heatmap, independent features are not correlated with each other.

C. *FEATURE SELECTION*:



Conclusion: All the features are important.

IV. IMPLEMENTATION OF REGRESSION ALGORITHMS

Various regressions algorithms have been applied:

- RANDOM FOREST REGRESSOR:

Random Forest Regressor is a regressor machine learning model which trains various Decision Trees on a random sampling basis based on the random state given. The final prediction turns out to be the average of all the predictions of various trees. The advantage of this model is that an error occurred by one model is overshadowed by other models. Thus the performance increases. This is the reason why I chose this model.

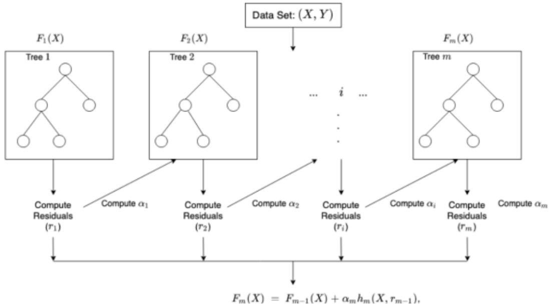
✓ HYPERPARAMETER TUNING:
Grid search CV has been used to tune the hyper parameters, it basically takes in the different possible hyper parameters that we should give as input, it then choses the best parameters based on Grid Search CV.

✓ The best hyper parameters were found to be max_depth of 14 and n_estimators as 101.

- XGBOOST:

XGBoost is an implementation of gradient boosted trees algorithm. Gradient Boosting is a part of supervised learning where in weak classifiers and strong classifiers are used together such that strong classifiers can support weak classifiers and thus the model performs well.

Diagramatically,


Data Set: (X, Y)
$F_1(X)$
$F_2(X)$
$F_m(X)$
Tree 1
Tree 2
Tree m
... i ...
Compute Residuals (r_1)
Compute α_1
Compute Residuals (r_2)
Compute α_2
Compute Residuals (r_i)
Compute α_i
Compute Residuals (r_m)
Compute α_m
$F_m(X) = F_{m-1}(X) + \alpha_m h_m(X, r_{m-1}),$

✓ HYPERPARAMETER TUNING:
Grid search CV has been used to tune the hyper parameters, it basically takes in the different possible hyper parameters that we should give as input, it then choses the best parameters based on Grid Search CV.

✓ The best hyper parameters were found to be max_depth of 6 and n_estimators as 226.

- DECISION TREE REGRESSOR:

Decision Tree regressor is the most widely used Regression Model is due to its ability to break complex data of high complexity into smaller branches thus reducing complexity. Hence, I chose this model

✓ HYPERPARAMETER TUNING:
Grid search CV has been used to tune the hyper parameters, it basically takes in the different possible hyper parameters that we should give as input, it then choses the best parameters based on Grid Search CV.

✓ The best parameters have been found as max_depth as 14, and min_samples_split as 10

- SEQUENTIAL FEATURE SELECTION:

Out of all the above models, I got XGBoost as the best model because it gave the best r2_score, now considering XGBoost as the best regression classifier, I have applied SSFS variant of the Sequential Feature Selection to improve the performance. This Sequential Feature Selector adds (forward selection) or removes (backward selection) features to form a feature subset in a greedy fashion. At each stage, this estimator chooses the best feature to add or remove based on the cross-validation score of an estimator. I have taken forward as True and floating as True for SFFS, and I chose scoring as 'r2' because we are dealing with regression problem, apart from these, I have taken k_features as 10 and cv as 4.

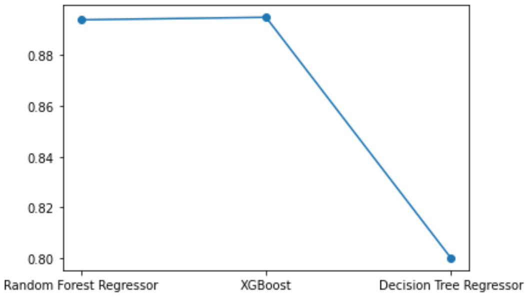
✓ Based on this, final score has been calculated.

V. EVALUATION OF MODELS

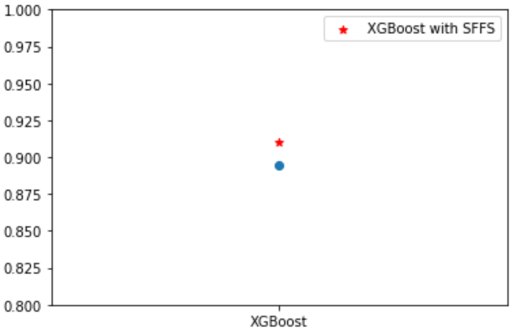
A. *PERFOMANCE MEASURES*:

MODEL	RFR	XGBoost	DTR	XGBoost with SFFS
R2 SCORE	0.894	0.895	0.80	0.91

B. *GRAPH*:

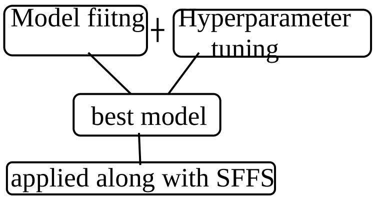


C. *XGBoost with SFFS*



VI. ANALYSIS AND CONCLUSIONS

- ✓ What has been done after preprocessing, cleaning and observing the data…



- ✓ Random Forest Regressor got an R2 score of around 0.894 upon hyperparamter tuning
- ✓ XGBoost got an R2 score of around 0.895 upon hyperparameter tuning
- ✓ Decision Tree Regressor got an r2 score of around 0.80 upon hyperparameter tuning.
- ✓ XGboost gave the best performance
- ✓ Upon doing Sequential Feature Selection for XGBoost, score increased to 0.91, and the ultimate performance increased.
- ✓ So, upon doing sequential feature selection, performance increases because computational efficiency is increased
- ✓ Performance comparison of all three models: XGBoost with SFFS > XGBoost > Random Forest Regressor > Decision tree Regressor
- ✓ XGBoost has the highest perfomance better than other two models because gradient boosting is a part of supervised learning
- ✓ Then it is Random Forest Regressor which is better than Decision Tree Regressor because RFR is nothing but combination of multiple decision trees and hence it has an added advantage because it compensates the low performance of each decision tree

VII. References

[1] https://www.analyticsvidhya.com/blog/2022/01/flight-fare-prediction-using-machine-learning/

[2] https://www.kaggle.com/code/unajtheb/flight-price-analysis-and-prediction

[3] https://www.kaggle.com/code/mohitdhapodkar/flight-fare-prediction-eda-and-modeling

[4] https://www.analyticsvidhya.com/blog/2022/01/flight-fare-prediction-using-machine-learning/

[5] https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SequentialFeatureSelector.html

[6] https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html

[7] https://www.datacamp.com/community/tutorials/xgboost-in-python