Flight Price Prediction

Bikkavolu Prasanthi- B20CS010

INTRODUCTION

In this project, we will analyse the flight fare prediction using a Machine Learning dataset using essential exploratory data analysis techniques, and then draw some predictions about the price of the flight based on some features such as the type of airline, arrival time, departure time, flight duration, source, destination, and more using a variety of regression models. A CSV file comprising 10683 rows and 11 columns, including the Price column, is included in the data set. The data set includes the cost of tickets during the months of March and June.

To minimise the noise in features, feature selection approaches were utilised. Selected models have had their parameters fine-tuned to ensure that they work well.

METHODOLOGY

Overview

There are various regression algorithms present out of which we shall implement the following

- XGBoost Regressor
- Gradient Boost Regressor
- Random Forest Regressor

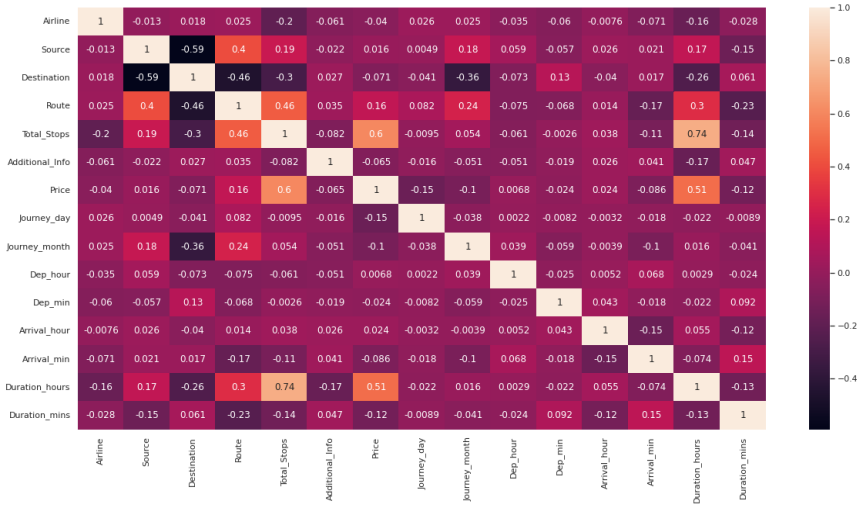
I also made use of GridSearchCV for hyperparameter tuning along with cross-validation. And Sequential Backward Selection was used for feature selection.

Data Pre-Processing

The following columns are included in the data set : 'Airline', 'Date_of_Journey','Source','Destination','Route','Dep_Time','Ar 'Duration', 'Total_Stops', 'Additional_Info'.

After dropping NA values, there were no missing values in the dataset.Date_of_Journey has been separated to Journey_day, Journey_month and then dropped.Dep_Time, Duration and Arrival_Time have been separated into hours and minutes in two columns and then dropped. Airline, Source, Route, Additional_Info and Destination have been labelled encoded. The data has been split into 80% train data and 20% test data.

Fig 1.1: Correlation Heatmap for the dataset



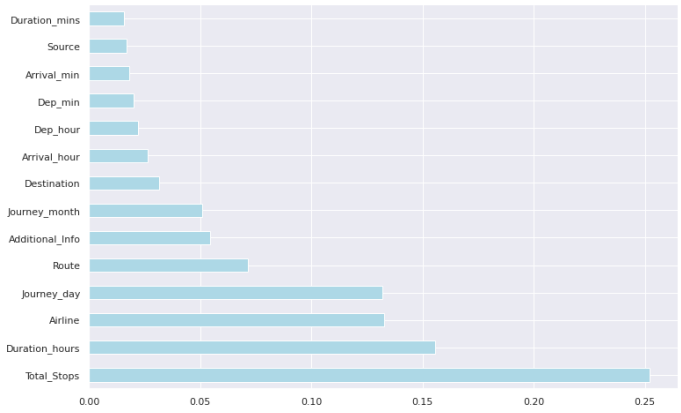
Exploratory data analysis

From Fig 1.1, we are able to say that Duration_hours, Total_stops and Price are related to each other. And Price is affected by the mentioned two features.

Feature Selection

Sequential Backward Selection has been used. The sequential backward selection algorithm aims to reduce the dimensionality of the initial feature subspace from N to K-features with a minimum reduction in the model performance to improve computational efficiency and reduce generalization error.

	feature_idx	cv_scores	avg_score
14	(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13)	[0.8191536289089849, 0.7837272313641752, 0.827...	0.814913
13	(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13)	[0.8234804312170582, 0.7969273576282301, 0.819...	0.818871
12	(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 13)	[0.8238108817032517, 0.7951317463415072, 0.821...	0.819737
11	(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10)	[0.8287208922798059, 0.7935901436945191, 0.826...	0.820752
10	(0, 2, 3, 4, 5, 6, 7, 8, 9, 10)	[0.8309508940407025, 0.797244819731802, 0.8190...	0.819789



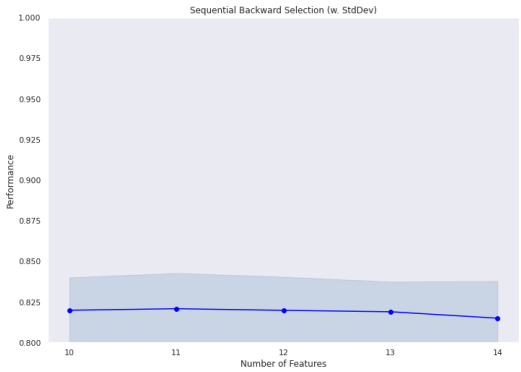


Fig 1.2: metric data frame of SBS, Feature importance plot, No. of features through SBS

The sequential backward search has been performed with a number of desired features as 10. The cross-validation scores of no. of features ranging from 10-14 have been obtained. From the feature importance and SBS results, *Duration_mins* has been dropped as its importance is low and without it, the avg scores are high relatively.

Implementation of Regression algorithms

- *XGBoost Regressor*

XGBoost is a distributed gradient boosting library that is optimised for efficiency, flexibility, and portability. It uses the Gradient Boosting framework to create machine learning algorithms. XGBoost is a parallel tree boosting algorithm that addresses a wide range of data science issues quickly and accurately.

3 types of XGBoost Regressors were made:

  - *XGBoost Regressor*
  - *XGBoost Regressor with GridSearchCV*
  - *XGBoost Regressor with GridSearchCV and SBS*

- *Gradient Boost Regressor:*

Gradient boosting is a machine learning approach for classification and regression problems. It returns a prediction model in the form of an ensemble of weak prediction models, most often decision trees. Gradient-boosted trees is the consequence of a decision tree that is a poor learner; it generally outperforms random forest. A gradient-boosted trees model is constructed in the same stage-wise manner as other boosting methods, but it differs in that it allows optimization of any differentiable loss function.

2 types of Gradient Boost Regressors were made:

  - *Gradient Boost Regressor with SBS*
  - *Gradient Boost Regressor with GridSearchCv and SBS*

- *Random Forest Regressor:*

Random forests, also known as random decision forests, are an ensemble learning approach for classification, regression, and other problems that work by training a large number of decision trees. The mean or average forecast of the individual trees is returned for regression tasks.

2 types of Random Forest Regressors are made:

- *Random forest with SBS*
- *Random Forest with GridSearchCv and SBS*

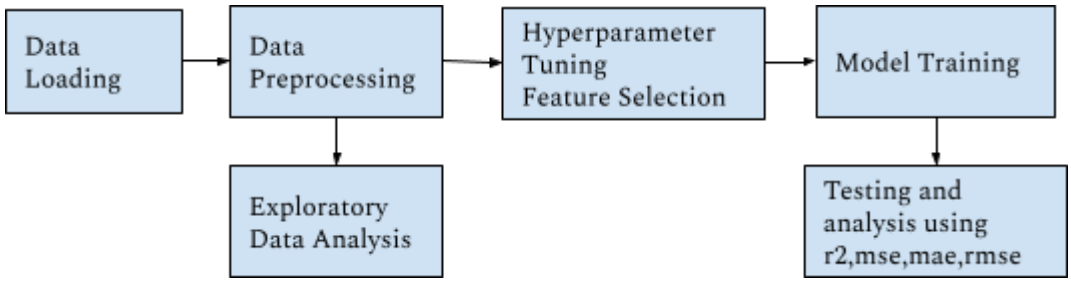
Hyperparameter Tuning with Cross-Validation: *GridSearchCV*

Feature Selection: *Sequential Backward Selection*

Hyperparameter tuning

Hyperparameter tuning along with Cross-Validation has been done using GridSearchCV on all 3 models. Parameters like learning rate,max_depth and n_estimators have been varied.

PROJECT PIPELINE



EVALUATION OF MODELS

The models implemented were evaluated using techniques like - **r2 score, mean absolute error, mean squared error, and root mean squared error.**

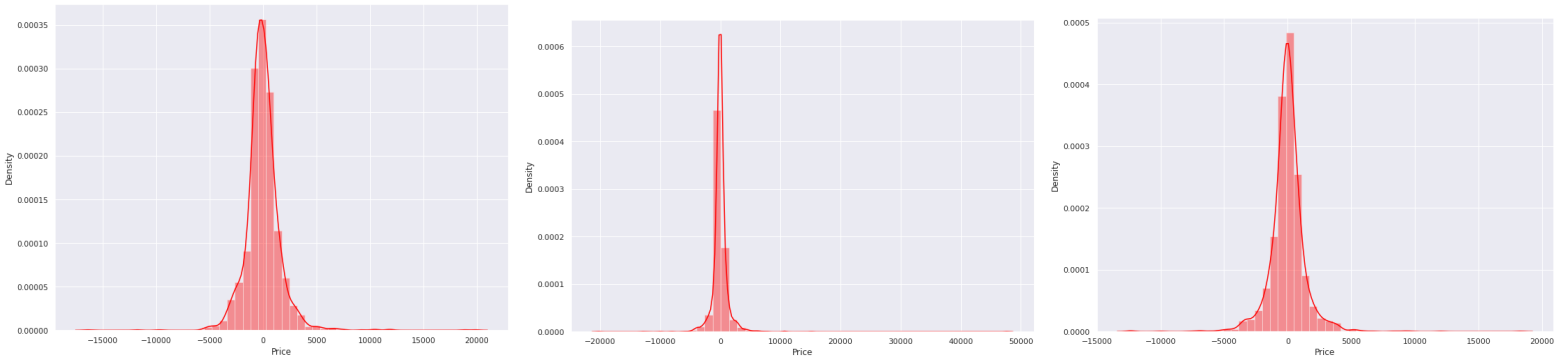
Fig1.3 The comparison plots of the models:

Table 1.1: The R2 scores of the models through the pipeline

Regressor ->	**Random Forest**	**Gradient Boost**	**XGBoost**
Normal	--	--	0.8450983536079149
After feature selection	0.8608424121291189	0.8425859634881302	--
After hyperparameter tuning and feature selection	0.8606031543771293	0.8520997448744265	0.8926387336920154

Fig 1.4: The Residual plots for all 3 models after parameter tuning and feature selection

Gradient Boost | Random Forest | XG Boost



CONCLUSION

XGBoost has the best accuracy compared to Random Forest and Gradient Boost. While XGBoost steadily performed well after hyperparameter tuning and feature selection, not much change can be seen in the other two. XG Boost has a lesser run time compared to Random Forest and Gradient Boost. Gradient Boost Regressor performs the least well among the three while XGBoost performs the best. Though feature selection enhances the performance of the model, it takes much run time. XGBoost is a good option for unbalanced datasets. XGBoost needs only a very low number of initial hyperparameters (depth of the tree, number of trees) when compared with the Random forest. XG Boost might have chances of overfitting as well and can rely only on when testing data is provided. Therefore for this model ensemble learning techniques perform better than other regression models out of which XG Boost has the best performance.

REFERENCES

1. K. Tziridis, T. Kalampokas, G. A. Papakostas and K. I. Diamantaras, "Airfare prices prediction using machine learning techniques," 2017 25th European Signal Processing Conference (EUSIPCO), 2017, pp. 1036-1039, doi: 10.23919/EUSIPCO.2017.8081365.

2. https://en.wikipedia.org/wiki/XGBoost

3. https://xgboost.readthedocs.io/en/stable/get_started.html

4.https://en.wikipedia.org/wiki/Gradient_boosting

5. https://en.wikipedia.org/wiki/Random_forest

6. http://rasbt.github.io/mlxtend/user_guide/feature_selection/SequentialFeatureSelector/

7. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html