FLIP ROBO

FLIGHT PRICE PREDICTION

Submitted by:

Satu Vinayak Pole

ACKNOWLEDGMENT

I would like to express my special gratitude to my SME Mr. Shwetank Mishra as well as “Flip Robo” team for letting me work for a project named “Flight Price Prediction” . Also thanks to my institute ‘Data Trained’ .

Also I would like to thank websites such as StackOverflow, geeksforgeeks and Youtube who has helped me in solving issues and errors.

[1] Introduction

{1.1} Business Problem Framing

Flights are one of the 3 modes of transport used frequently by person to travel around the world. Prices of flights are always expensive compared to all other forms of transport. The reason behind this high prices is higher cost of flights and their maintenance and taking into account all other premium features or services provided to customer with faster travel time.
There are various websites that are used to book flight tickets. It is often seen flight prices mostly increases as the booking date comes nearer, and in some instances during holiday seasons the flight prices are higher compared to other days.

Flight price prediction can help customers to get better deals/prices on booking via websites and/or applications that sell or help in booking flight tickets. Since the flight prices vary unexpectedly, a person booking ticket can be benefited from this model to book advance tickets for his/her journey as the app/website developed will tell customer when the prices will be going to rise or go down.

{1.2} Conceptual Background of the Domain Problem

Customers are seeking to get the lowest price for their ticket, while airline companies are trying to keep their overall revenue as high as possible and maximize their profit. However, mismatches between available seats and passenger demand usually leads to either the customer paying more or the airlines company loosing revenue. Airlines companies are generally equipped with advanced tools and capabilities that enable them to control the pricing process. However, customers are also becoming more strategic with the development of various online tools to compare prices across various airline companies .Conducted researches employ a variety of techniques ranging from statistical techniques such as regression to different kinds of advanced data mining techniques.

Using such models can be helpful in customers and airline point of view as this kind of models can be a benefit to both.

{1.3} Review of Literature

The factors that affect the flight price have to be studied and their impact on price has also to be modeled. An analysis of the past data is to be considered. Since many people travel via flight there is a need for such kind of model that will help airline companies to generate more revenue and also customers to plan and book their flights based on price changes.

Therefore, in this project report, we present various important features to use while predicting flight prices with good accuracy. We can use regression models, using various features to have lower Residual Sum of Squares error. While using features in a regression model some feature engineering is required for better prediction.

The primary aim of this report is to use these Machine Learning Techniques and provide ML models which can then serve the users

{1.4} Motivation for the Problem Undertaken

In this case I have scrapped flight data and saved in a csv file. This can be then used to determine prices of flights departing from airport and also can used by flight companies to analyze and generate additional revenue from this model.

The project was the first provided to me by Flip Robo Technologies as a part of the internship programme. The exposure to real world data and the opportunity to deploy my skillset in solving a real time problem has been the primary motivation

Early prediction of the demand along a given route could help an airline company pre-plan the flights and determine appropriate pricing for the route. In addition, competition between airlines makes the task of determining optimal pricing is hard for everyone. So prime motive is to build flight price predication system based on short range time frame (7- 14 days) data available prior to actual take-off date.

[2] Analytical Problem Framing

{2.1} Mathematical/ Analytical Modeling of the Problem

First phase of problem involves gathering data. This was done by scrapping different kinds of data that is related to Flight price prediction. All the data that is seen in dataset was scrapped from www.kayak.in website for a period from 26th January 2023 (Thursday) to 1st February 2023 (Wednesday). Data was scrapped for departure point as Goa Airport to Delhi Airport for all kinds of flight classes. Next phase was data cleaning and explaining data with the help of visualization plots and graphs. Further all pre- processing steps were done. Since our target variable is 'price', hence regression algorithm models were built. Further hyperparameter tuning was performed on the best model.

{2.2} Data Sources and their formats

Data is collected via scrapping from www.kayak.in website for 1 week ranging from 26th Jan 2023(Thursday) to 1st Feb 2020 (Wednesday) using selenium method and saved in csv file. The data was scrapped for flights departing from Goa Airport to Delhi Airport on above mentioned days. Approximately more than 2000 flights details were collected throughout the week.

The scrapped data contains 10 columns which includes features such as Date, flight name, flight arrival location, flight departure location, flight arrival time and flight departure time, flight journey duration, stops taken by flight before reaching destination, data source I.e. from this websites the data was taken by kayak.in and the most important target variable flight prices. All the columns had the datatype as 'object'.

{2.3} Data Preprocessing Done

- The scrapped dataset named 'Flight data.csv' was uploaded using pandas library.
- The dataset had 2117 rows and 12 columns of which it contained 2 non useful columns such as 'Unnamed:0' and 'Unnamed:0.1'. These columns were dropped as this columns just contained index values for scrapped data.
- Further the dataset was checked for any duplicate entries and it was found that there are 64 duplicates and it was later removed. The dataset did not have any null values.
- Further in arrival time column there were some rows which contains data such as '02:20+1' instead of 02:20. This +1 represented that flight will be landing next day. This data points were cleaned using str.replace method. Later the arrival and departure time columns were further separated into hours and minutes for departure and arrival.
- The duration data was scrapped in this -'2h 20 min' format, hence it had to be changed to 2:20. This was also done using a looped python function. This column was further divided into 2 columns as duration hour and duration minutes.

- Then later using visualization plots and graphs data analysis was explained.
- Later the data was label encoded, and the data which contained outliers and skewness were checked and removed.
- Further the data was split into features and label and features were statndardised using Standard scaler method.
- Further the model building was done and its performance was noted.

{2.4} Data Inputs- Logic- Output Relationships

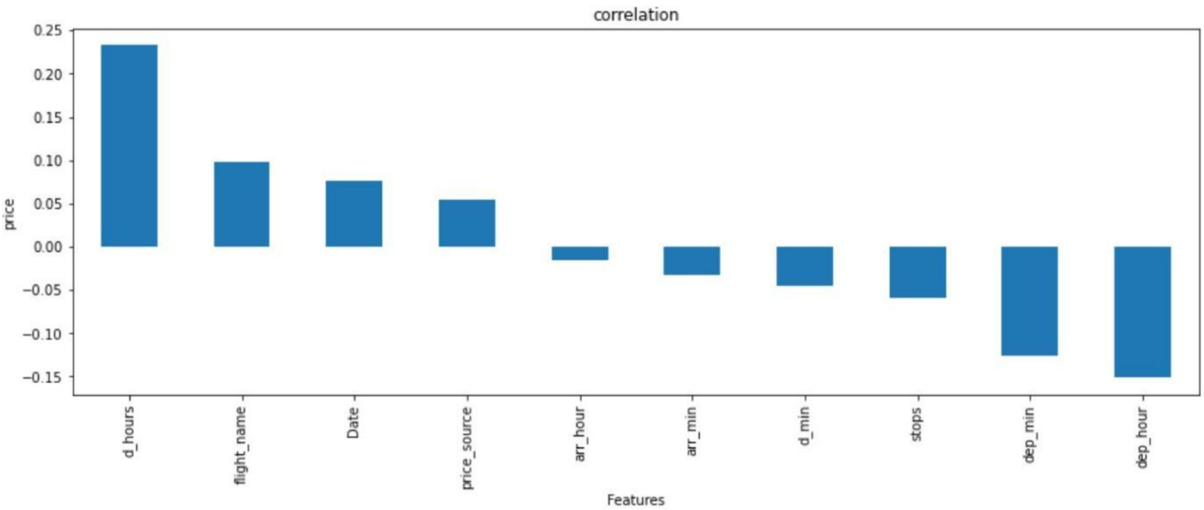


FIGURE 2.1

As per figure 2.1, it can be seen that duration hours (denoted as d_hours) is the most positively related feature to predict flight prices and departure hours (denoted by dep_ hour) is the most negatively related feature to predict flight prices. Arrival hour(arr_hour) is the least related feature to label.

Also it is seen that no features show multicollinearity problem. This was confirmed after plotting heatmap and also using VIF (Variance Inflation Factor).

It is seen from visualization plots and graphs that the flight which depart from evening to night or in very early mornings, it is seen that flight prices are higher at this time compared to flights departing some other time.

Also plots suggests that the flights that takes more time to reach their destination, they are expensive compared to short duration flights.

{2.5} Hardware and Software Requirements and Tools Used

Hardware used to complete the analysis and model building:
Model: ASUS TUF A15
Processor: AMD RYZEN 5 4600H OCTA CORE
RAM: 8GB
ROM:500 GB SSD

Software used to complete the analysis and model building is:

Jupyter Notebook (via Anaconda Navigator) - used for data scrapping, data cleaning, all the analysis and model building.

[3] Model/s Development and Evaluation

{3.1} Identification of possible problem-solving approaches (methods)

I have used both statistical and analytical approach in model building. I have used plots and graphs which shows the relation within features and with respect to target variable. Further data encoding was done using label encoder. Later the dataset was checked if its continuous columns except target variable has any outliers and skewness present or not. It was found that 1 column had outliers, hence it was treated using z-score method. Date column had skewness present but since the data scrapped is for only 7 days that is the reason skewness was seen, hence it was kept untreated.

The data checked if it had any multicollinearity within features, but it had none. Before model building part started dataset was divided into features and label and after that feature data was standardized for model building purpose. Since the target variable is flight price, which is a continuous kind of data, hence regression algorithms were used. Four algorithms were trained and tested on dataset and best model was selected. Later hyperparameter tuning was done on the best model based on its R2 score and RMSE value. In this case Random Forest regressor model was chosen as it had high R2 score and low RMSE value. Later on best model hyperparameter tuning was performed. The best model was saved from all 4models based on good R2 score and lower RMSE values.

{3.2} Testing of Identified Approaches (Algorithms)

The price being a continuous type column, it was know that it is going to be a regression kind of problem. Hence after standardizing (scaling) feature data the data was then split into train test split. Further Random Forest regressor model was used to find best random state possible which gave best training and testing R2 score for model. Folllowing is the list of algorithms that were used in model building:

1. Random Forest regressor
2. AdaBoost regressor
3. Decision Tree regressor
4. Gradient Boosting regressor

{3.3} Run and Evaluate selected models

```
rf=RandomForestRegressor()
ab=AdaBoostRegressor()
dt=DecisionTreeRegressor()
gbdt=GradientBoostingRegressor()
```

FIGURE 3.1

The above figure 3.1 shows the variable names given to regression models that were mentioned earlier.

1) Random Forest Regressor model

```
rf.fit(x_train,y_train)
```

```
RandomForestRegressor()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
# since thge model is already trained, below code will help to predict based on train and test data

y_pred=rf.predict(x_train)

pred=rf.predict(x_test)

#printing r2 score for testing and training models.
#r2 score give value of how good the model has studied and learnt the data

print(f'training R2 score:{r2_score(y_train,y_pred)*100:.2f}%')
print(f'testing R2 score:{r2_score(y_test,pred)*100:.2f}%')
```

```
training R2 score:98.39%
testing R2 score:92.42%
```

FIGURE 3.2

```
#cross validation score
print('Cross Validation Score for Random Forest regressor model :- ',((cross_val_score(rf,x_scaled,y,cv=7).mean())*100))
```

```
Cross Validation Score for Random Forest regressor model :-  76.54836093095
```

```
#finding mean absolute error() for above model(MAE)
print('mean absolute error',mean_absolute_error(y_test,pred))

#finding root mean_squared_error(RMSE)
print('root mean squared error',np.sqrt(mean_squared_error(y_test,pred)))
```

```
mean absolute error 2158.3140160579796
root mean squared error 3736.390765967458
```

FIGURE 3.3

Figures 3.2 shows the training and testing R2 score for Random Forest model while Figure 3.3 shows the cross validation score (CV), Mean Absolute error (MAE) and Root Mean Squared Error (RMSE). The training and testing R2 score for above model as shown is 98.39% and 92.42% respectively while CV score is 76.54. The MAE and RMSE values are 2158.31 and 3736.39 respectively.

2) AdaBoost Regressor model

AdaBoost Regressor

```
ab.fit(x_train,y_train)
```

```
AdaBoostRegressor()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
# since thge model is already trained, below code will help to predict based on train and test data

y_pred=ab.predict(x_train)

pred=ab.predict(x_test)

#printing r2 score for testing and training models.
#r2 score give value of how good the model has studied and learnt the data

print(f'training R2 score:{r2_score(y_train,y_pred)*100:.2f}%')
print(f'testing R2 score:{r2_score(y_test,pred)*100:.2f}%')
```

```
training R2 score:59.10%
testing R2 score:60.69%
```

```
#cross validation score
print('Cross Validation Score for AdaBoost regressor model :- ',((cross_val_score(ab,x_scaled,y,cv=7).mean())*100))
```

```
Cross Validation Score for AdaBoost regressor model :-  46.37939322833099
```

```
#finding mean absolute error() for above model(MAE)
print('mean absolute error',mean_absolute_error(y_test,pred))

#finding root mean_squared_error(RMSE)
print('root mean squared error',np.sqrt(mean_squared_error(y_test,pred)))
```

```
mean absolute error 7251.717576197027
root mean squared error 8510.144047949527
```

FIGURE 3.4

Figures 3.4 shows the training and testing R2 score, cross validation score (CV), Mean Absolute error (MAE) and Root Mean Squared Error (RMSE) for AdaBoost model. The training and testing R2 score for above model as shown is 59.10 % and 60.69 % respectively while CV score is 46.37. The MAE and RMSE values are 7251.71 and 8510.14 respectively.

3) Decision Tree Regressor model:

Decision Tree Regressor

```
dt.fit(x_train,y_train)
```

```
DecisionTreeRegressor()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
# since thge model is already trained, below code will help to predict based on train and test data

y_pred=dt.predict(x_train)

pred=dt.predict(x_test)

#printing r2 score for testing and training models.
#r2 score give value of how good the model has studied and learnt the data

print(f'training R2 score:{r2_score(y_train,y_pred)*100:.2f}%')
print(f'testing R2 score:{r2_score(y_test,pred)*100:.2f}%')
```

```
training R2 score:99.98%
testing R2 score:72.93%
```

```
#cross validation score
print('Cross Validation Score for Decision Tree regressor model :- ',((cross_val_score(dt,x_scaled,y,cv=7).mean())*100))
```

```
Cross Validation Score for Decision Tree regressor model :-  65.11548776017077
```

```
#finding mean absolute error() for above model(MAE)
print('mean absolute error',mean_absolute_error(y_test,pred))

#finding root mean_squared_error(RMSE)
print('root mean squared error',np.sqrt(mean_squared_error(y_test,pred)))
```

```
mean absolute error 2231.4608610567516
root mean squared error 7061.738364002469
```

FIGURE 3.5

Figures 3.5 shows the training and testing R2 score, cross validation score (CV), Mean Absolute error (MAE) and Root Mean Squared Error (RMSE) for Decision Tree model. The training and testing R2 score for above model as shown is 99.98 % and 72.93 % respectively while CV score is 65.11. The MAE and RMSE values are 2231.46 and 7061.73 respectively.

4) Gradient Boosting Regressor model:

Gradient Boosting Regressor

```
gbdt.fit(x_train,y_train)
```

```
GradientBoostingRegressor()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
# since thge model is already trained, below code will help to predict based on train and test data

y_pred=gbdt.predict(x_train)

pred=gbdt.predict(x_test)

#printing r2 score for testing and training models.
#r2 score give value of how good the model has studied and learnt the data

print(f'training R2 score:{r2_score(y_train,y_pred)*100:.2f}%')
print(f'testing R2 score:{r2_score(y_test,pred)*100:.2f}%')
```

```
training R2 score:89.34%
testing R2 score:88.68%
```

```
#cross validation score
print('Cross Validation Score for Gradient Boosting regressor model :- ',((cross_val_score(gbdt,x_scaled,y,cv=7).mean())*100
```

```
Cross Validation Score for Gradient Boosting regressor model :-  75.43646729671168
```

```
#finding mean absolute error() for above model(MAE)
print('mean absolute error',mean_absolute_error(y_test,pred))

#finding root mean_squared_error(RMSE)
print('root mean squared error',np.sqrt(mean_squared_error(y_test,pred)))
```

```
mean absolute error 3139.730027530608
root mean squared error 4566.112050211043
```

FIGURE 3.6

Figures 3.6 shows the training and testing R2 score, cross validation score (CV), Mean Absolute error (MAE) and Root Mean Squared Error (RMSE) for Gradient Boosting model. The training and testing R2 score for above model as shown is 89.34 % and 88.68 % respectively while CV score is 75.43. The MAE and RMSE values are 3139.73 and 4566.11 respectively.

From above model outputs I selected Random Forest as the best model. As according to me:
- It has seen most of the data than other models hence its training R2 score is more than other models.
- Also its testing R2 score is more than other models.
- It has lower RMSE value than other models.

The following figure will show hyperparameter tuning performed on Random Forest model using GridSearchCV.

```python
#importing GridSearchCV library
from sklearn.model_selection import GridSearchCV
```

```python
#setting parameters for tuning
grid_param=[{'criterion':["squared_error", "absolute_error", "poisson"],
            'min_samples_split':[2,2.5],
             'max_depth':[1,2],
             'bootstrap':[True,False]}]
```

```python
grid=GridSearchCV(rf,param_grid=grid_param)
```

```python
grid.fit(x_train,y_train)
```

```
GridSearchCV(estimator=RandomForestRegressor(),
             param_grid=[{'bootstrap': [True, False],
                          'criterion': ['squared_error', 'absolute_error',
                                        'poisson'],
                          'max_depth': [1, 2], 'min_samples_split': [2, 2.5]}])
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```python
#best parameters for tuning
grid.best_params_
```

```
{'bootstrap': True,
 'criterion': 'poisson',
 'max_depth': 2,
 'min_samples_split': 2}
```

```python
#using best parameters to train

rf1=RandomForestRegressor(criterion='poisson'
                          ,min_samples_split=2,
                          max_depth=2,
                          bootstrap=True)
```

```python
rf1.fit(x_train,y_train)
```

FIGURE 3.7

```python
# since thge model is already trained, below code will help to predict based on train and test data

y_pred=rf1.predict(x_train)

pred=rf1.predict(x_test)

#printing r2 score for testing and training models.
#r2 score give value of how good the model has studied and learnt the data

print(f'training R2 score:{r2_score(y_train,y_pred)*100:.2f}%')
print(f'testing R2 score:{r2_score(y_test,pred)*100:.2f}%')
```

```
training R2 score:32.16%
testing R2 score:28.11%
```

```python
#cross validation score
print('Cross Validation Score for Random forest tuned model :- ',((cross_val_score(rf1,x_scaled,y,cv=7).mean())*100))
```

```
Cross Validation Score for Random forest tuned model :-  23.425152745131015
```

```python
#finding mean absolute error() for above model(MAE)
print('mean absolute error',mean_absolute_error(y_test,pred))

#finding root mean_squared_error(RMSE)
print('root mean squared error',np.sqrt(mean_squared_error(y_test,pred)))
```

```
mean absolute error 7441.417286466785
root mean squared error 11508.489611944318
```

FIGURE 3.8

Figure 3.7 shows the snapshot of commands used for hyperparameter tuning. In this figure it can be seen that parameters list is feed in the Grid Search CV. This parameters are then applied on Random Forest model in permutation and combinations and best parameters are provided that will give maximum accuracy to the model.

In Figure 3.8 it is shown that Random Forest model is again trained with new parameters and its training and testing R2 score alongwith CV, RMSE and MAE scores are printed.

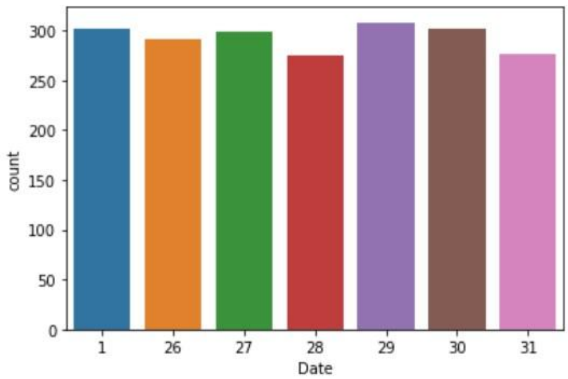
It is seen that the tuned model does not give good accuracy hence the original Random Forest model is saved as the best model.

{3.4} Key Metrics for success in solving problem under consideration

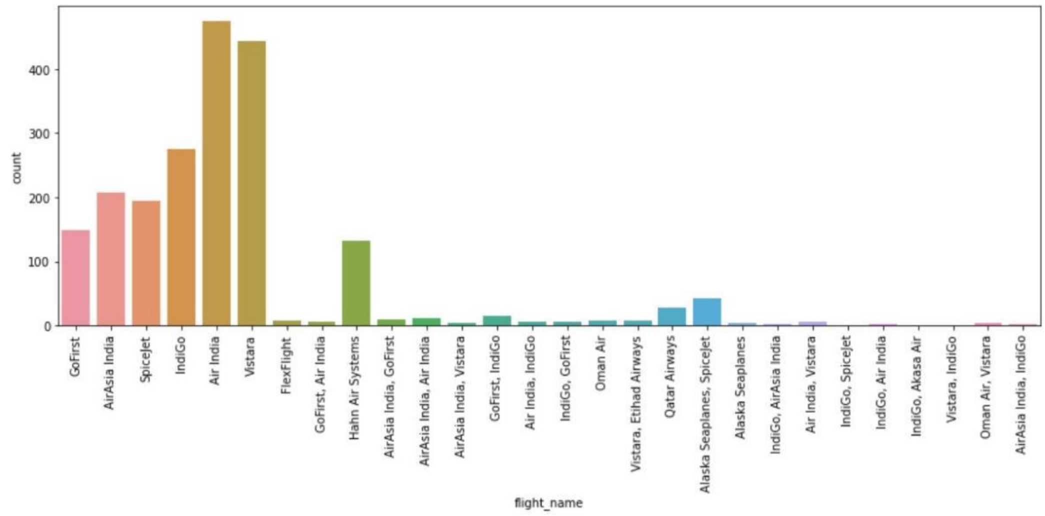
The key metrics used in this models are:
1) Cross validation (CV) score: This is used on models to check if the model is overfitting or not. If the testing R2 score and CV score is equal or near equal then model is said to be best fitted model or else overfit model.
2) MAE (Mean Absolute error): Represents average error i.e. error for every single data point and takes its average. This error is lesser the better the model. Zero MAE means best model.
3) RMSE(Root Mean Squared Error): This is similar to MAE but noise is exaggerated and larger error are punished. This is the main metric used in interpreting model. RMSE close or equal to zero means best model.

{3.5} Visualizations

Univariate analysis:

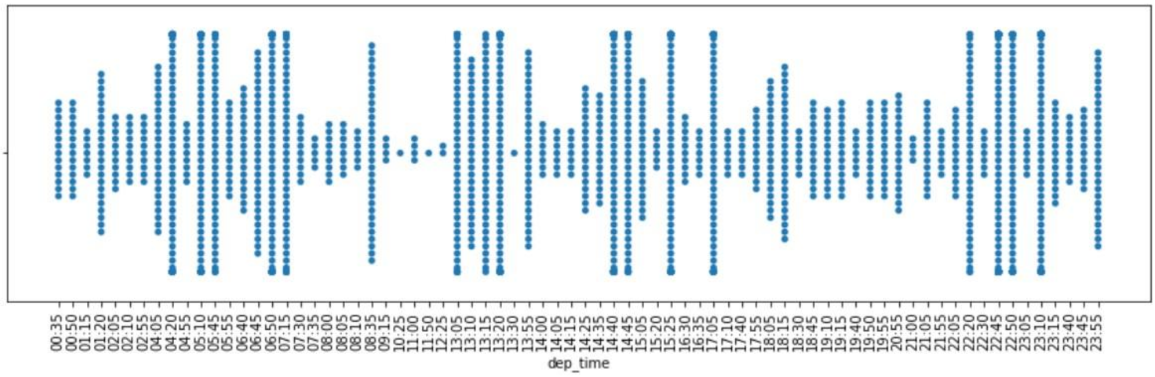


Observation:
- On average approx 290 flights operate from Goa to Delhi airport.
- Flights compared to other days are low on 28th January.



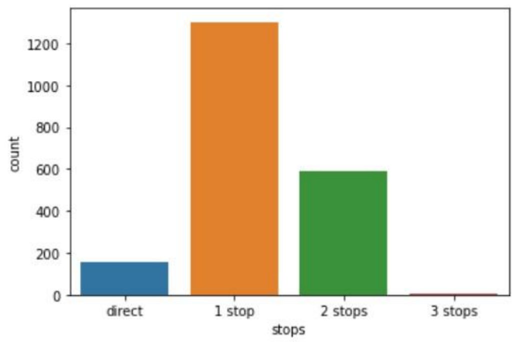
Observation:

- Air India and Vistara operates more flights from Goa to Delhi.



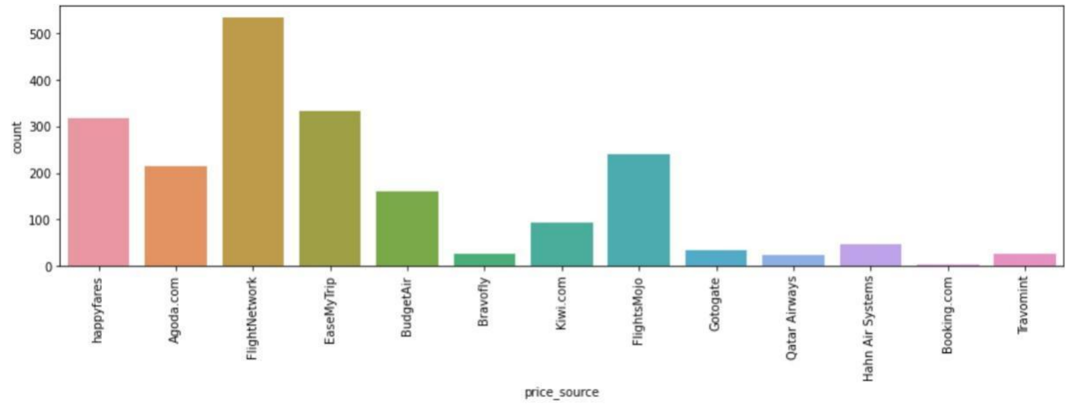
Observation:

- The above plot shows that most flights depart from Goa mostly in evening from 1pm to 3pm and then at late night from 10.30 to very early morning.



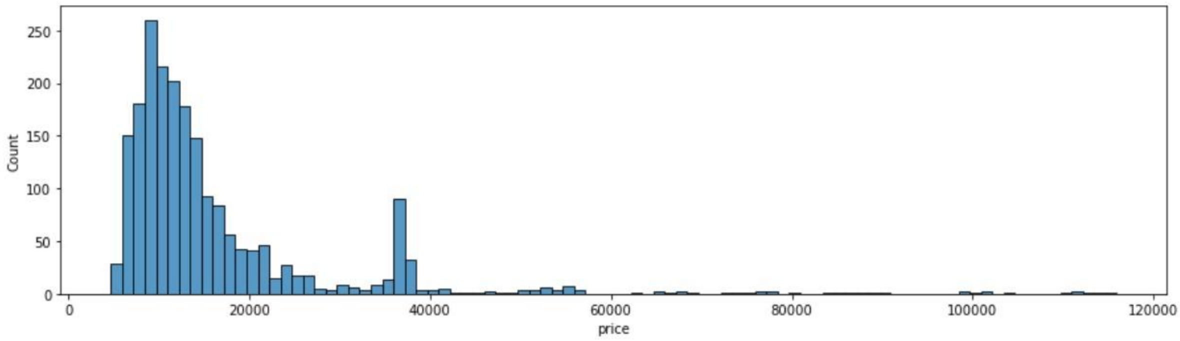
Observation:

- The above plot shows that most flights take 1 stop to reach Delhi. While almost 200 flights are direct flight departing from Goa. Much less flights present that take >2 stops to reach Delhi.



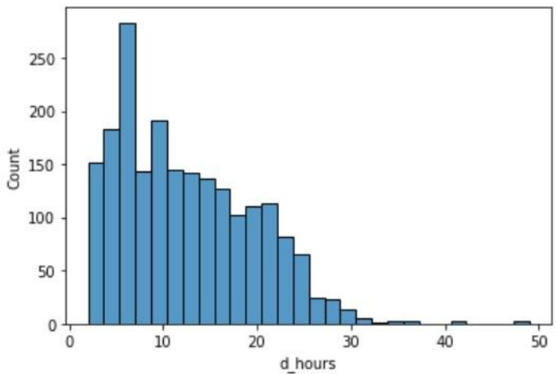
Observation:

- The above plot shows the source websites that shows cheapest prices of flights departing from Goa to Delhi on particular day.
- It can be seen that in most cases source is FlightNetwork which shows that FlightNetwork source provides cheapest prices for number of flights.
- Very less number of flights are shown from booking.com



Observation:

- To travel from Goa to delhi most flights charge from approx.INR 5000 to 20000. Over 250 flights charge approx. INR 10000 per trip.
- The prices from Goa to Delhi can go as high upto 115000 approx. This can be because some flight taken big timely stops before reaching destination.

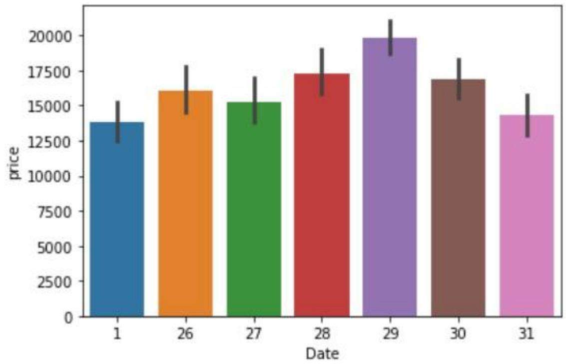


Observation:

Most flights mentioned in dataset take around approx.4-6 hours to reach Delhi from Goa.

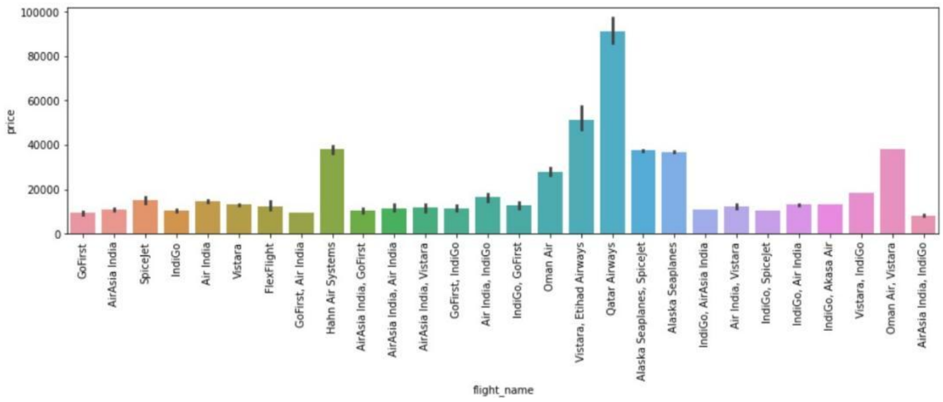
- The number of flights are very less who take more time to reach destination 'delhi'

Bivariate analysis:



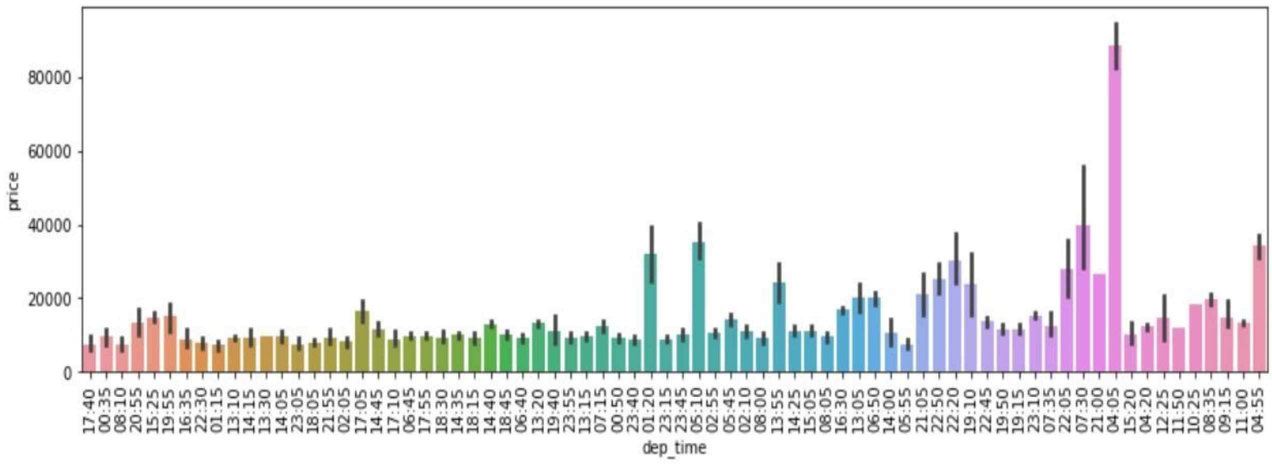
Observation:

- As seen above fare prices are highest and keeps on increasing from 26-29 and then fare reduces from 30,31 till 1st date.



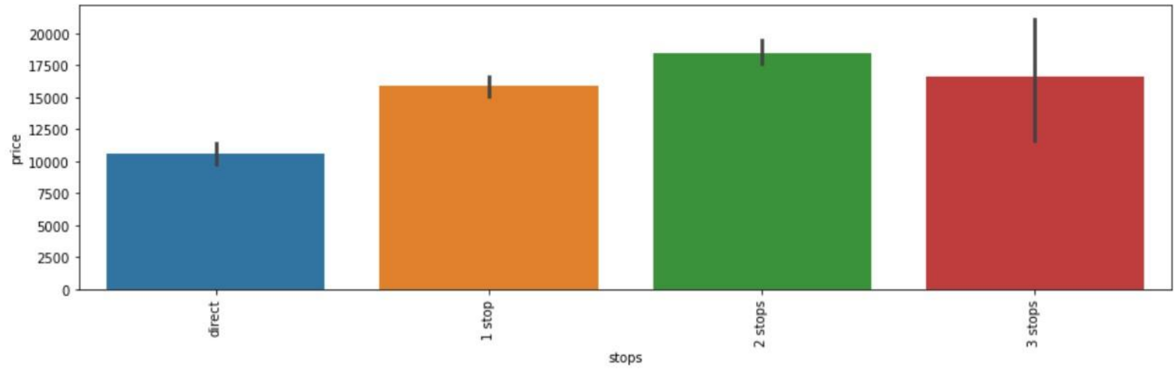
Observation:

- The flight company that charges highest fare to go from Goa to delhi is Qatar airways.
- The least fare from Goa to Delhi is charged by IndiGo,GoFirst,AirIndia,Vistara and Spicejet flights.



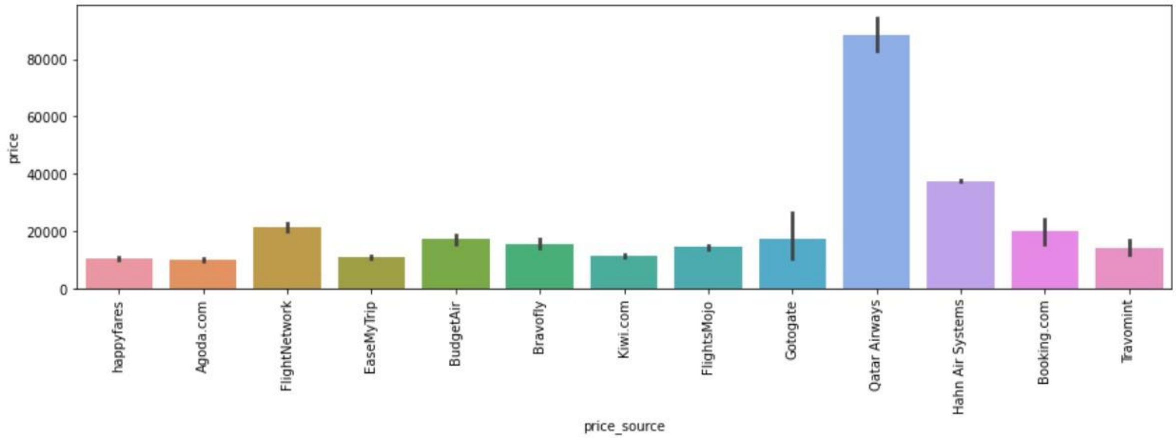
Observation:

It can be seen from the above plot that people travelling in evening to early morning flights, will have to pay more fare compared to other departure timings.



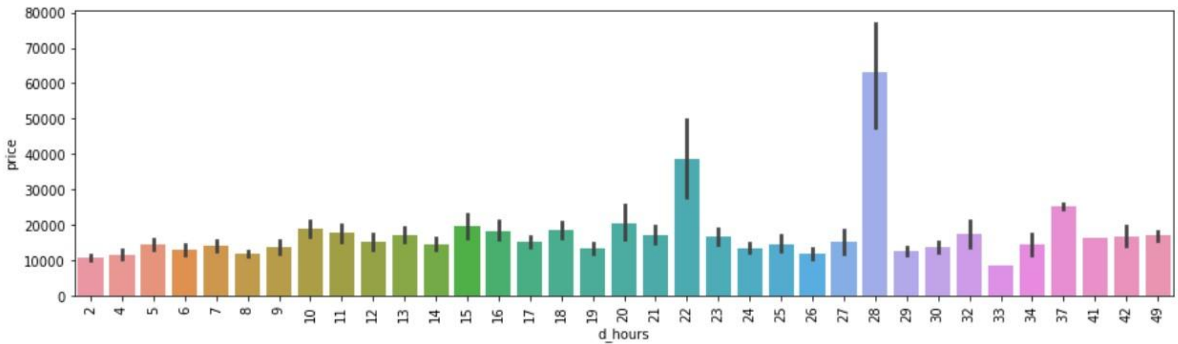
Observation:

- It can be seen from the plot that for direct flights price is less and gradually increases until flight takes 2 stops.
- It is also seen that flights that take 3 stops to reach destination charges lower than flights that take 2 stops.



Observations:

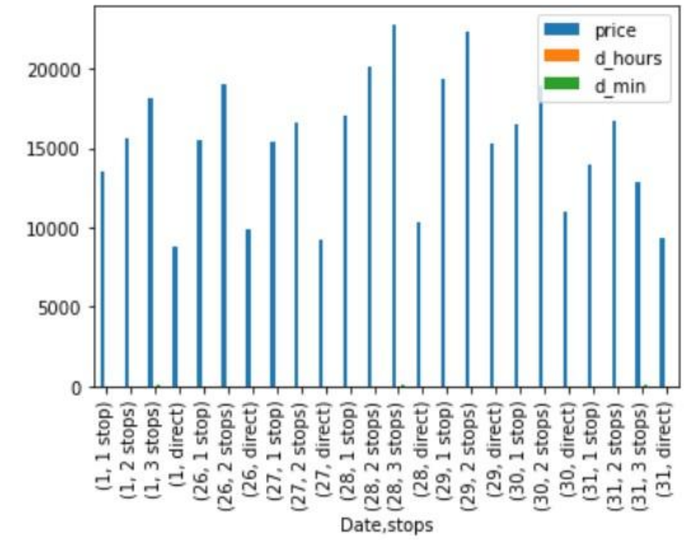
- The price source vs price plot shows that the prices source from Qatar airways is higher compared to other flight bpoking portals.
- It is also seen that happyfares and Agoda.com are the portals that can be trusted to book cheaper tickets for this journey.



Observation:

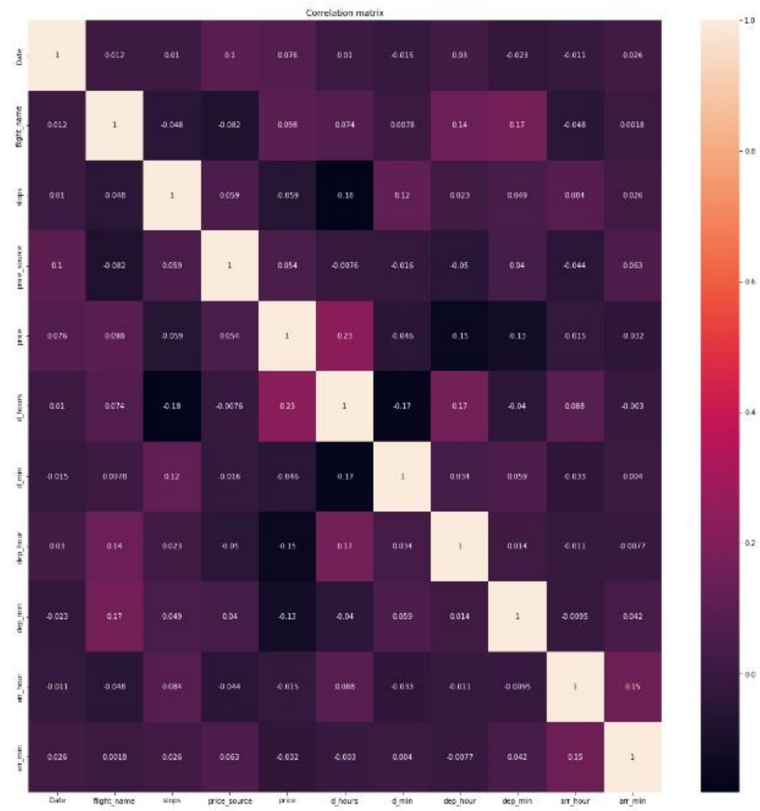
- The above plot shows that flights that take 22 and 28 hours to reach destination, their fare prices are higher. For other flights the prices are almost similar.

Multivariate analysis:



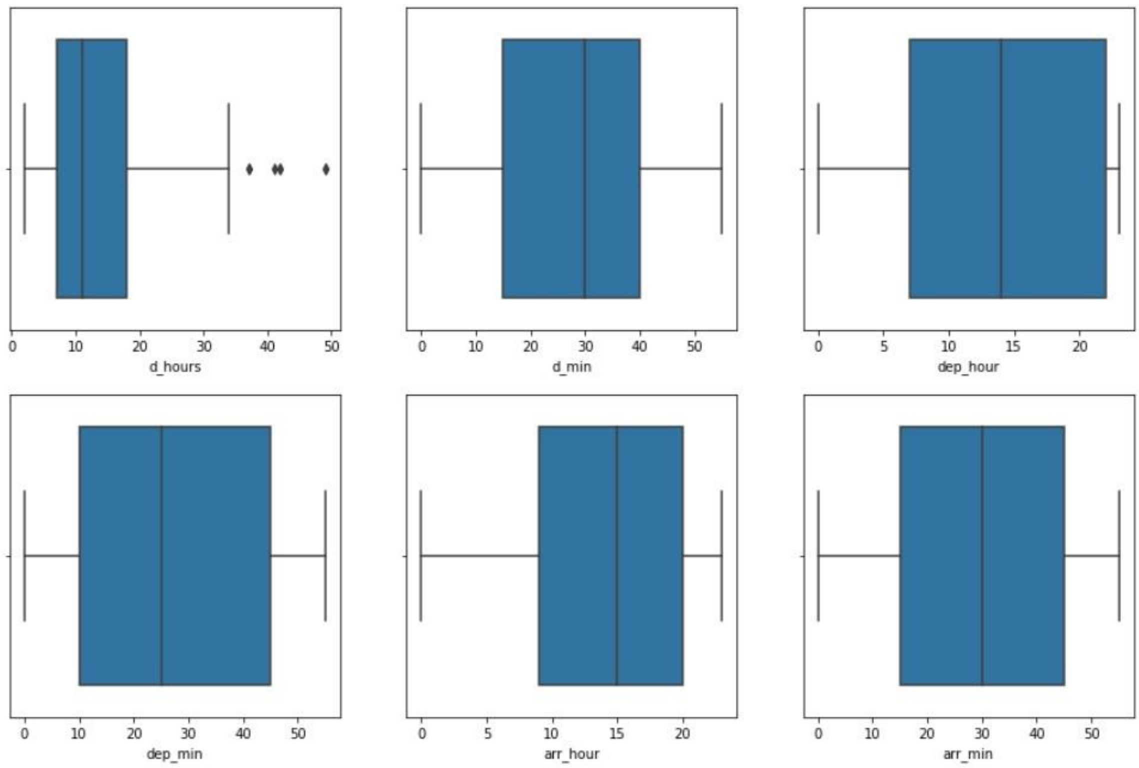
Observation:

- On 1st Feb, prices are low compared to other days, this includes all kinds of flights i.e flights that either takes stops or direct. The prices of flights for all stops kind increases from 27 to 28, and peaking highest on 29th. After that the prices slowly decreases with least prices in 1st date.

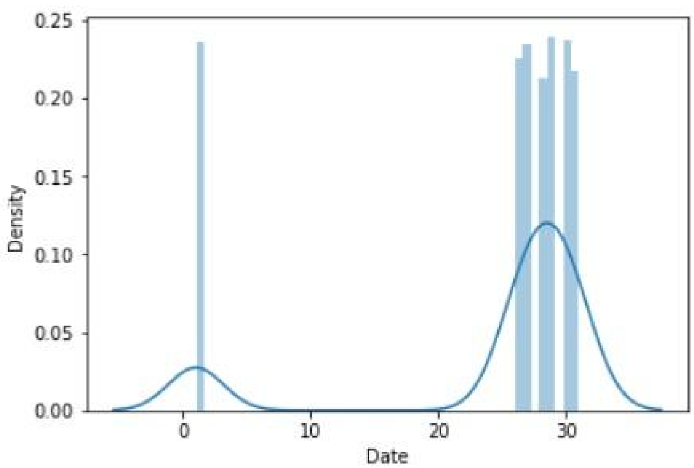


Observation:

The above heatmap shows that there is no multicollinearity seen within features.



Observation:
Only duration hours column has outliers present in it.4



In case of date column, though date is considered continuous data but here since only 7 dates are present. The dates starting from 26-31st Jan shows good normal distribution as seen in plot but since i scrapped data for 1st feb there is one more curve seen. If the data was scrapped for more days than data could have been normally distributed.

	vif_values	feature_names
0	1.013719	Date
1	1.072082	flight_name
2	1.072641	stops
3	1.034352	price_source
4	1.117187	d_hours
5	1.041147	d_min
6	1.059072	dep_hour
7	1.042887	dep_min
8	1.053232	arr_hour
9	1.032006	arr_min

Observation:

The above dataframe consists of VIF values with respect to feature names. There is no multicollinearity seen in above data.

[4] CONCLUSION

{4.1} Key Findings and Conclusions of the Study

In this project report, we have used machine learning algorithms to predict the Flight prices. We have mentioned the step by step procedure to analyze the dataset and finding the correlation between the features. The feature set was then given as an input to 4 algorithms .Hence we calculated the performance of each model using different performance metrics and compared them based on these metrics. Then the best model was selected and tuned called hyperparameter tuning. And the best model which gives best score alongwith lowest RMSE score was selected and saved for future use.

The main conclusions that can be obtained from this study is:

- Flight prices increases during the time of weeked/holidays as it was seen that on Saturday and Sunday flight prices were higher and prices went down on Monday. This means that flight prices tend to go up and down based on customer influx.
- It is seen that Air India and Vistara are the flight companies that operates most on this route.
- Flight prices of Qatar Airways is much higher compared to other flights.
- Flight prices increases if the flight takes more than 1 stop to reach its destination.
- It is also seen people travelling in early morning flights have to pay more.
-

{4.2} Limitations of this work and Scope for Future Work

- This dataset contained very less data I.e. nearly 2000 data points. This can be increased by scrapping or getting more data on flights, so model knowledge will increase subsequently it performing well.
- Also this data is only limited to only 1 departure and arrival. If the data was scrapped for multiple locations then it would have been better analyzing data and learning about how flights work for other locations.
- In this dataset the only time price is showing variation when the weekend comes nearer. This is because data was limited to only 7 days. Hence we can add data for months to better analyzing it. Also we should also take into account seasonal/ festival holidays as this times flight prices tend to increase.
- In this dataset the columns were only 10-12, hence analysing was not upto mark. This additional day may include columns like flight route, kind of food served in flights, leg room provided or not, window and aisle seating, etc as this will help customers to book their flights based on the services they are being offered. This data can also be used by flight companies

to see what kind of services are actually required by customers and customize their flight experiences as per customers needs resulting in higher customer influx in turn good profits.