Flight Price Prediction Using Machine Learning

Submitted by:

Yash Bhardwaj

ACKNOWLEDGMENT

I whole heartedly thank our SME Sapna Verma, flip robo technologies for their support towards me to complete this project.

I also thank my family for supporting me.

Yash Bhardwaj

Contents

INTRODUCTION

Business Problem Framing

With ever increasing air route connectivity throughout the world, air travel has become a common, integral and faster way to travel. Predicting fares for airlines is an important as well as challenging task since a constant fluctuation in fares is observed and it is known to be dependent on varied set factors. With tremendous study in area, it is observed that using Machine Learning, Artificial Intelligence and Deep Learning techniques an estimation of flight fares at a given time can be obtained within seconds .So, we have to work on a project where you collect data of flight fares with other features and work to make a model to predict fares of flights.

Conceptual Background of the Domain Problem

Anyone who has booked a flight ticket knows how unexpectedly the prices vary. The cheapest available ticket on a given flight gets more and less expensive over time. This usually happens as an attempt to maximize revenue based on -

1. Time of purchase patterns (making sure last-minute purchases are expensive)

2. Keeping the flight as full as they want it (raising prices on a flight which is filling up in order to reduce sales and hold back inventory for those expensive last-minute expensive purchases)

In this paper, we use different Machine Learning Regression approach to predict flight fare by providing basic details of departure date and time, arrival time, source, destination, number of stops and name of the airline and selecting the best approach.

Review of Literature

Flight Fare Prediction: Machine learning project by
https://medium.com/@tejashree-nawale

Here, they have been provided with prices of flight tickets for various airlines between the months of March and June of 2019 and between various cities, using which they had aim to build a model which predicts the prices of the flights using various input features.

They have 2 datasets here — training set and test set.

The training set contains the features, along with the prices of the flights. It contains 10683 records, 10 input features and 1 output column — 'Price'.

The test set contains 2671 records and 10 input features. The output 'Price' column needs to be predicted in this set. We will use Regression techniques here, since the predicted output will be a continuous value.

Following is the features available in the dataset – Airline, Date_of_Journey, Source, Destination, Route, Dep_Time, Arrival_Time ,Duration, Total_Stops, Additional_Info, Price.

The author performed the process in following order:

- Exploratory Data Analysis
- Feature Engineering
- Feature selection
- Model Deployment

The author have used Random Forest model giving out accuracy score of around 95% and 80% for training dataset and test dataset respectively.

Motivation for the Problem Undertaken

The objective or motivation is to model the price of flights with the available (scraped) independent variables. This model will then be used by the clients to understand how exactly the prices vary with the variables. They can accordingly manipulate the strategy of the firm and concentrate on areas that will yield high returns. Further, the model will be a good way for the clients to understand the pricing dynamics of a new market. Moreover, this might give the management team an insight of the real time world.

Analytical Problem Framing

Mathematical/ Analytical Modelling of the Problem

First phase of problem modelling involves data scraping of flights from internet. For that purpose, flight data is scrap from www.yatra.com for timeframe of 04 August 2022 to 06 August 2022 and 01 September 2022 to 04 September. Data is scrape for flights on different routes like New Delhi to Mumbai, Mumbai to Bengaluru etc. Data is scrap for Economy class flights. Next phase is data cleaning & pre-processing for building ML Model. Our objective is to predict flight prices which can be resolve by use of regression-based algorithm. Further Hyperparameter tuning performed to build more accurate model out of best model.

Data Sources and their formats

Data is collected from www.yatra.com for timeframe of 04 August 2022 to 06 August 2022 and 01 September 2022 to 04 September using selenium and saved in CSV file. Data is scrape for flights on routes like New Delhi to Mumbai, Mumbai to Bengaluru etc. Data is scrap for Economy class. Around 3500 flights details are collected for this project.

```
#Loading the data
df = pd.read_excel("flightsfaredata.xlsx")
print(df.shape)
```

```
(3509, 10)
```

```
df.sample(10)
```

	Unnamed: 0	Airline	Source	Departure	Destination	Arrival	Duration	Stop(s)	Date	Fare
2648	2648	IndiGo	Bangalore	11:10	Chennai	12:10	1h 00m	Non Stop	Fri, 5 Aug	6,004
3355	3355	IndiGo	Ahmedabad	04:30	Hyderabad	11:25	6h 55m	1 Stop	Sat, 3 Sep	5,975
811	811	IndiGo	Bangalore	15:25	Kolkata	18:00	2h 35m	Non Stop	Sat, 6 Aug	9,049
1856	1856	Air India	New Delhi	20:00	Kolkata	08:50\n+ 1 day	12h 50m	1 Stop	NaN	10,848
1398	1398	IndiGo	New Delhi	16:20	Kolkata	21:35	5h 15m	1 Stop	Sat, 6 Aug	8,579
160	160	Go First	New Delhi	09:30	Mumbai	18:35	9h 05m	1 Stop	Sat, 6 Aug	8,579
1683	1683	SpiceJet	New Delhi	13:05	Kolkata	18:10	5h 05m	1 Stop	Sun, 4 Sep	7,383
2857	2857	IndiGo	Bangalore	14:50	Chennai	15:55	1h 05m	Non Stop	Sat, 3 Sep	3,694
1929	1929	IndiGo	Mumbai	18:05	Bangalore	23:40	5h 35m	1 Stop	Thu, 4 Aug	12,033
2594	2594	Air India	Mumbai	18:35	Bangalore	10:30\n+ 1 day	15h 55m	2 Stop(s)	NaN	11,819

Unnecessary column of index name as ‘Unnamed: 0’ is drop out. There are 9 features in dataset including target feature ‘Fare’. The data types of different features are as shown below:

```
#checking for datatypes

df.dtypes
```

```
Airline        object
Source         object
Departure      object
Destination    object
Arrival        object
Duration       object
Stop(s)        object
Date           object
Fare           object
dtype: object
```

Data Pre-processing

The dataset is large and it may contain some data error. In order to reach clean, error free data some data cleaning & data pre-processing performed data.

Data Integrity check –

The dataset have around 80 duplicated rows. So we drop these duplicated data from the dataset:

```
#checking shape of dataset
df.shape
```

```
(3509, 9)
```

```
#removing duplicated rows from the dataset
df.drop_duplicates(inplace=True)

#again checking for shape of the now data
df.shape
```

```
(3429, 9)
```

Conversion of Duration column from hour and minutes format into Minutes –

By default, Duration of flights are given in format of [(h) hours, (m) minute] which need to convert into uniform unit of time. Here is the code to convert duration in terms of minute:

```
#firstly removing the leading zero with empty string as it would give us error while converting
df['Duration'] = df['Duration'].str.replace(' 0',' ')

# and now converting the duration into minutes only

df['Duration_New']=df['Duration'].str.replace("h", '*60').str.replace(' ','+').str.replace('m','*1').apply(eval)
```

Creation of four new variables from Departure and Arrival variables –

By default, the data format for departure and arrival time of flights is in 24-hour time and has object datatype as well. So, the below code creates two new variable from each variable i.e. departure hour and departure minute from departure and same for arrival:

```
df['Dep_hour'] = pd.to_datetime(df['Departure']).dt.hour
df['Dep_min'] = pd.to_datetime(df['Departure']).dt.minute
```

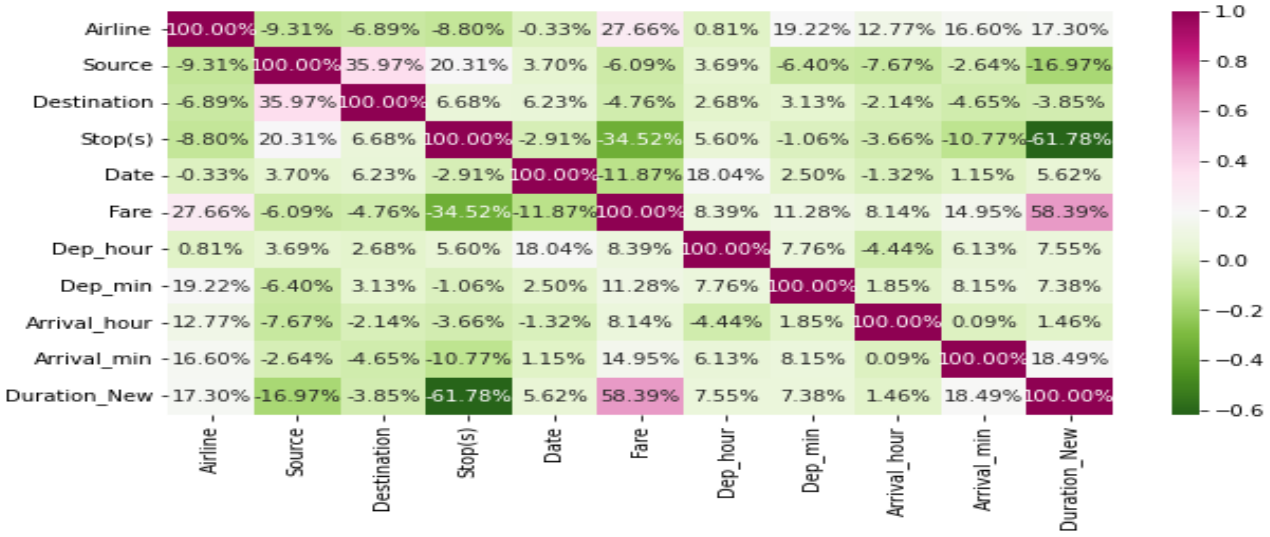
```
df['Arrival'] = df['Arrival'].apply(lambda x:x[:5])      #applying lambda function to remove the useless strings

df['Arrival_hour'] = pd.to_datetime(df['Arrival']).dt.hour  #using to_datetime and extracting hours only
df['Arrival_min'] = pd.to_datetime(df['Arrival']).dt.minute #same to extract minutes only
```

Conversion of Fare i.e. the target variable into numerical datatype –

By default, the target variable is in object datatype, the fare variable has prices in thousand only, so we have replaced the thousand place period with a decimal for a better understanding. Below is the code that replaces the period with decimal and convert the variable into numeric one:

```
#replacing the period with a decimal and rounding of the number
df['Fare'] = (pd.to_numeric(df['Fare'].apply(lambda x : x.replace(',','.')))).apply(lambda x : round(x,2))
```

Data Inputs- Logic- Output Relationships



Correlation heatmap is plotted to gain understanding of relationship between target features & independent features. We can see that Duration feature is correlated for more than 0.55 with target variable Fare. Stop(s) and Airline features also have good correlation with the target variable. Remaining feature are poorly correlated with target variable price.

Hardware and Software Requirements and Tools Used

Hardware Used –

1. Processor — AMD A9-9425 RADEON R5, 5 COMPUTE CORES 2C+3G 3.10 GHz

2. RAM — 4 GB

3. GPU — AMD Radeon(TM) R5 Graphics

Software utilised –

1. Anaconda – Jupyter Notebook

2. Selenium – Web scraping

Libraries Used –

General library for data wrangling & visualisation:

```
#Linear algebra

import numpy as np

#data processing

import pandas as pd

#data visualization

import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

Libraries used for web scraping data from e-commerce website:

```
#Importing required libraries
import pandas as pd
import selenium
from selenium import webdriver
from selenium.common.exceptions import NoSuchElementException , StaleElementReferenceException
import time
import warnings
warnings.filterwarnings('ignore')
```

Libraries used for machine learning model building and evaluation metrics:

```
#algorithms

from sklearn.linear_model import LinearRegression,Lasso,Ridge
from sklearn.tree import DecisionTreeRegressor
from sklearn.neighbors import KNeighborsRegressor
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestRegressor , AdaBoostRegressor , GradientBoostingRegressor
from xgboost import XGBRegressor

#importing metrics

from sklearn.model_selection import train_test_split,cross_val_score
from sklearn.metrics import r2_score,mean_squared_error,mean_absolute_error,mean_squared_log_error
```

Model/s Development and Evaluation

Identification of possible problem-solving approaches (methods)

First part of problem solving is to scrap data from www.yatra.com website which we already done. Next part of problem solving is building machine learning model to predict flight price. This problem can be solve using regression-based machine learning algorithm like linear regression. For that purpose, first task is to convert categorical variable into numerical features. Once data encoding is done then data is scaled using standard scalar. Final model is built over this scaled data. For building ML model before implementing regression algorithm, data is split in training & test data using train_test_split from model_selection module of sklearn library. After that model is train with various regression algorithm and 10-fold cross validation is performed. Further Hyperparameter tuning performed to build a model that is more accurate.

Testing of Identified Approaches (Algorithms)

Phase 1 - Web Scraping Strategy employed in this project as follow:

1. Selenium will be used for web scraping data from www.yatra.com

2. Flights on different routes on date period 04 August 2022 to 06 August 2022 and 01 September 2022 to 04 September 2022.

3. Selecting features to be scrap from website.

4. In next part web scraping code executed for above mention details.

5. Exporting final data in Excel file.

Phase 2 - The different regression algorithm used in this project to build ML model are as below:

- Linear Regression

- ❖ Lasso Regression
- ❖ Ridge Regression
- ❖ K-Neighbors Regressor
- ❖ Decision Tree Regressor
- ❖ Random Forest Regressor
- ❖ Ada Boost Regressor
- ❖ Gradient Boosting Regressor
- ❖ XGB Regressor

Run and Evaluate selected models

Best Model?

	Mean Squared Error	Mean Absolute Error	Accuracy Score
XGBRegressor	0.027089	0.111625	0.888436
RandomForestRegressor	0.030730	0.113621	0.873440
GradientBoostingRegressor	0.038497	0.147136	0.841454
DecisionTreeRegressor	0.051111	0.129498	0.789502
AdaBoostRegressor	0.075844	0.228749	0.687640
KNeighborsRegressor	0.115765	0.248948	0.523232
Lasso	0.143883	0.306700	0.407429
LinearRegression	0.143928	0.305841	0.407244
Ridge	0.143928	0.305841	0.407244

We can see that XGB Regressor gives maximum R2 score of 88.8436. Among all model we will select XGB Regressor as final model and we will perform hyper parameter tuning over this model to enhance its R2 Score.

```
from sklearn.experimental import enable_halving_search_cv
from sklearn.model_selection import HalvingGridSearchCV              #importing HalvingGridSearch
```

```
xgb = XGBRegressor()
x_train,x_test,y_train,y_test=train_test_split(x_sca,y,random_state=76,test_size=.33)
```

```
parameter = {'n_estimators':[400,500],'gamma':np.arange(0,0.2,0.1),
             'booster' : ['gbtree','dart','gblinear'], 'max_depth':[6,8],
             'eta' : [0.01, 0.1] }

search = HalvingGridSearchCV(xgb, parameter, verbose= 10).fit(x_train, y_train)
search.best_params_
```

Final model with optimum values for hyperparameters:

```
Final_mod=XGBRegressor(booster='dart', max_depth=8, eta=0.1,
                                  gamma=0.1, n_estimators=400)

Final_mod.fit(x_train,y_train)
pred=Final_mod.predict(x_test)
print('R2_Score:',r2_score(y_test,pred)*100)
print('mean_squared_error:',mean_squared_error(y_test,pred))
print('mean_absolute_error:',mean_absolute_error(y_test,pred))
print("RMSE value:",np.sqrt(mean_squared_error(y_test, pred)))
```

```
R2_Score: 87.93981513662095
mean_squared_error: 0.029283439888165185
mean_absolute_error: 0.11799616207025317
RMSE value: 0.1711240482461924
```

Saving the model using pickle:

```
#saving the best model

xgb = XGBRegressor(booster='dart', max_depth=8, eta=0.1,
                                  gamma=0.1, n_estimators=400)
import pickle
pickle.dump( xgb ,open('flightsfareprediction.pkl','wb'))
```

Key Metrics for success in solving problem under consideration

Following metrics used for evaluation:

1. Mean absolute error which gives magnitude of difference between the prediction of an observation and the true value of that observation.

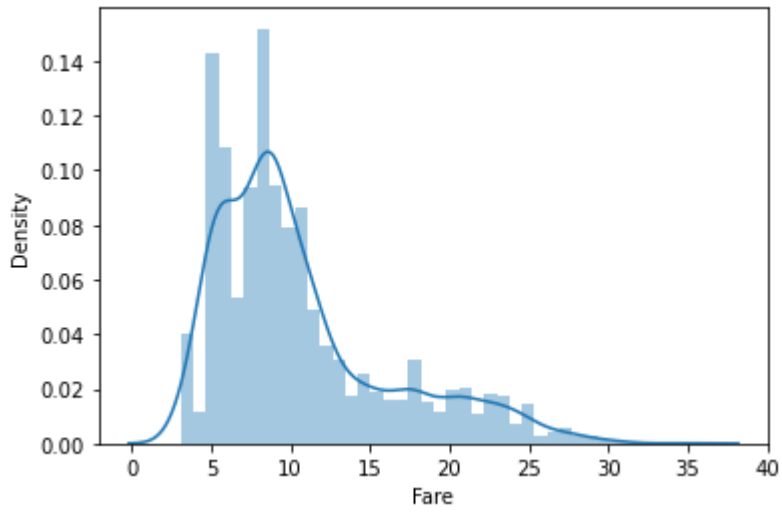
2. Root mean square error is one of the most commonly used measures for evaluating the quality of predictions.

3. R2 score which tells us how accurate our model predict result, is going to important evaluation criteria along with Cross validation score.

Visualizations

Now, let's check out the key findings from the EDA:

1. **The target variable 'Fare':**

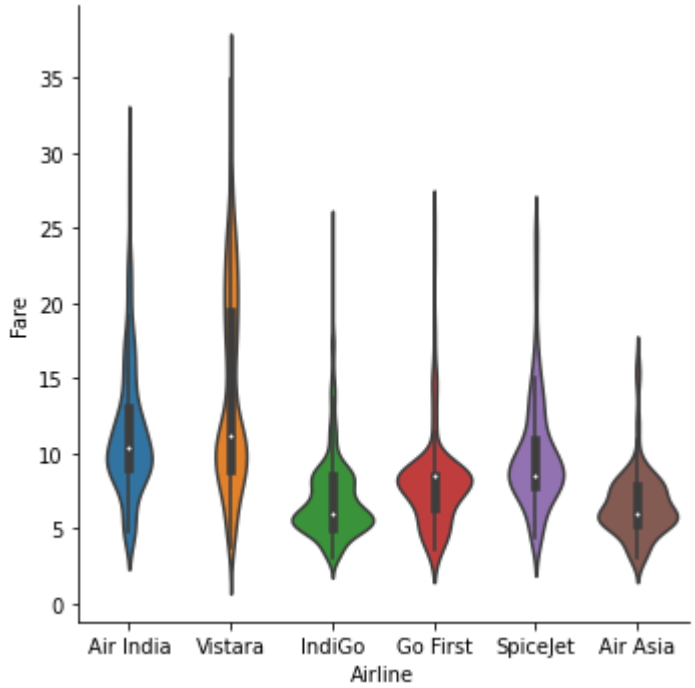


Observation:

- Skewness can be seen in the data for flight fares
- A few outliers can also be seen

- Most of the flight fares are in range five to fifteen thousand

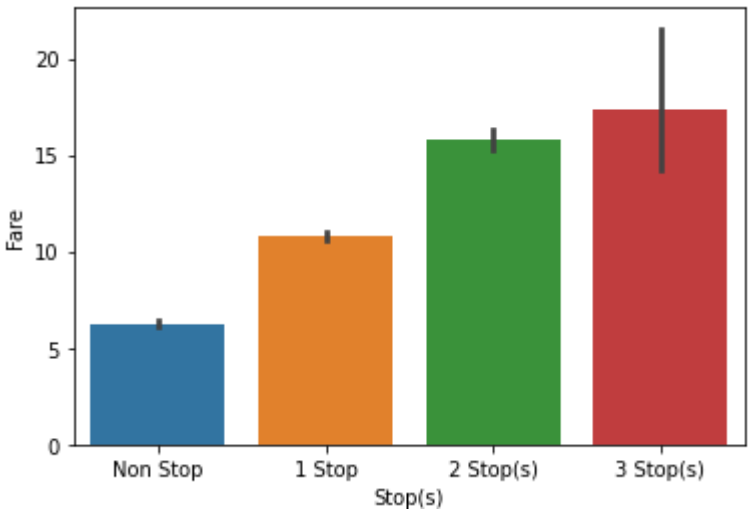
2. **Airline and Fare:**



Observation:

- Vistara airline has the highest average flight fares following by Air India airline
- Air Asia has the cheapest average flight fares

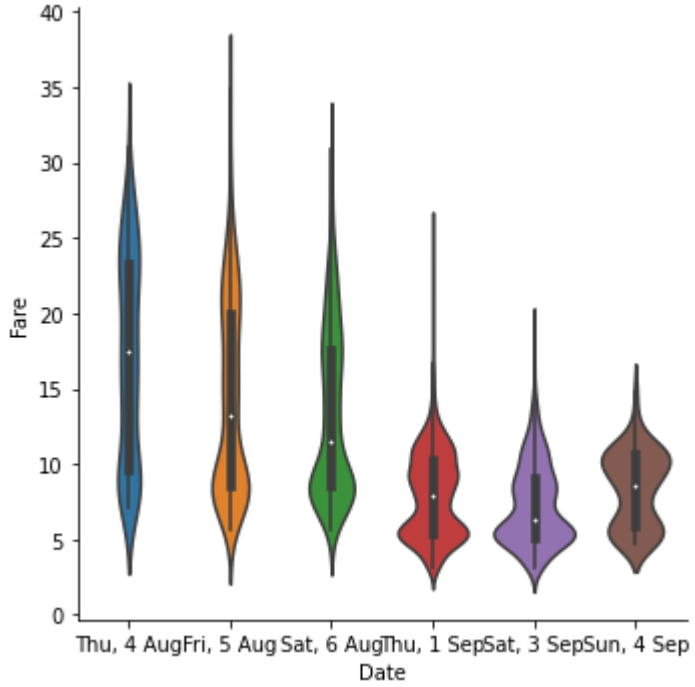
3. **Stops(s) and Fare:**



Observation:

- The flight fares are increasing with increasing in number of stops
- More the stops higher the price
- Non-stop flights have cheapest average flight fares

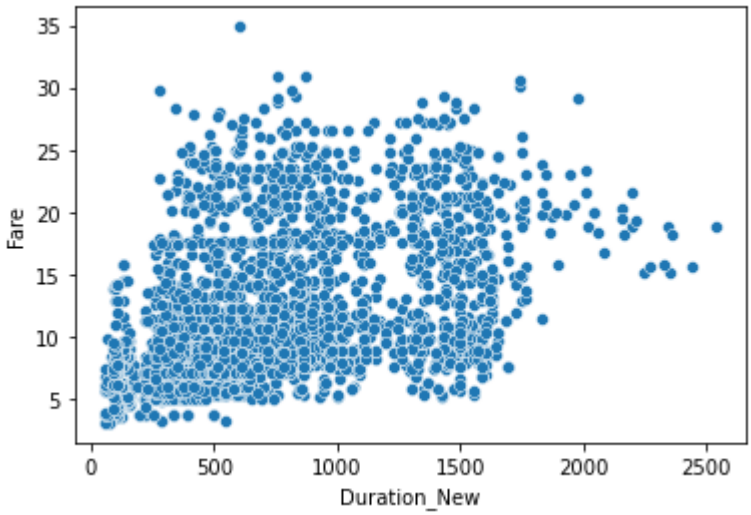
4. **Date and Fare:**



Observation:

- It can be seen that current flight fares are much higher than the flight fares after a month
- A person should book tickets at least a month before the date of booking for cheaper flight fares

5. **Duration and Fare:**



Observation:

- The data isn’t distributed uniformly this is why skewness might be there
- It can be seen that long duration flights have high fare as well

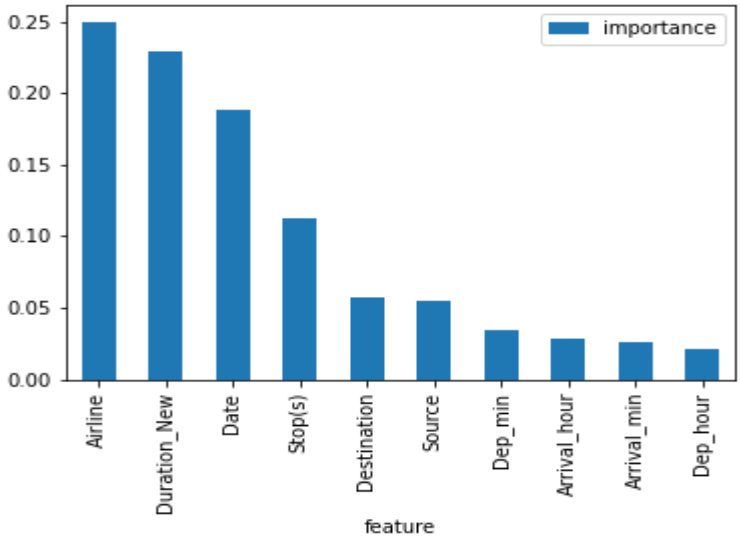
CONCLUSION

Key Findings and Conclusions of the Study

XGB Regressor giving us maximum R2 Score, so XGB Regressor is selected as best model.

After hyper parameter tuning Final Model is giving us R2 Score of 87% which must be slightly more accurate compare to earlier R2 score of 89%.

For XGB Regressor below is the feature importance variable:



Limitations of this work and Scope for Future Work

- In this study we focus on flights on a few routes only, more route can incorporate in this project to extend it beyond present investigation.
- This investigation focus on short timeframe which is 3 days from current booking date to 3 days of next month which can be extended variation over larger period.
- Time series analysis can be performed over this model.
- A much more extensive hyperparameter tuning and feature engineering can also produce better results.

References

- https://medium.com/geekculture/flight-fare-prediction-93da3958eb95
- https://www.ijraset.com/research-paper/aircraft-ticket-price-prediction-using-machine-learning
- https://www.analyticsvidhya.com/blog/2022/01/flight-fare-prediction-using-machine-learning/
- https://www.kaggle.com/general/93016
- https://www.investopedia.com/ask/answers/060315/what-difference-between-linear-regression-and-multiple-regression.asp#:~:text=Regression%20analysis%20is%20a%20common,regressions%20with%20multiple%20explanatory%20variables