

Predicting Flight Price Using Machine Learning Models

Vigneshwar-B

vigneshbw2002@gmail.com

Abstract

With the recent global economic difficulties and tech layoffs, many people have been faced with financial difficulties and every dollar now matters more than ever. Many industries are forced to adjust the prices of their commodities quite rampantly and unpredictably, leaving consumers in dismay. The airline industry is one of the few domains with most sporadic pricing. This makes it very difficult for consumers who want to save money due to their current financial situation. In this project, I used machine learning to help airline customers predict the price of an air ticket within India based on features such as the source of the travel, the destination of the travel, the departure date, the arrival date, the type of airline and the number of stoppages if any. I used 5 different supervised machine learning models to make the prediction of the flight prices. The result of these 5 models are:

- ExtraTreesRegressor (79%)
- RandomForestRegressor(80%)
- CatBoostRegressor(83%)
- LGBMRegressor (80%)
- XGBRegressor (82%)

As a final step, after evaluating the performance of all these models, I proceeded with XGBoost which outperformed the other 4 models. In order to make the model more accessible for consumers, I deployed the model via the Heroku cloud platform.

Overview and Motivation

The motivation of the project comes from the current economic hardship making it difficult for people of India to keep pace with the financial crisis. Helping the individuals to save on their flight ticket booking was at the heart of this project.

Related Work

Airline ticketing congestion and tight delivery time windows force passengers to exorbitant prices which they wouldn't have paid under normal circumstances¹. This poses a lot of strains on consumers and in the past few years, machine learning have been applied into different time series problems. Airline companies use many different variables to determine the flight ticket prices: indicator whether the travel is during the holidays, the number of free seats in the plane etc. Such variables have been used to optimize the pricing and routing of airlines². Stefan Klein and co.³ found that during the early years of e-Commerce, the tourism sector had high expectations for online booking. All the major airlines invested huge sums of money, not only to make booking features available, but also to integrate them into attractive, easy-to-use Web offerings. Nevertheless, even after years of investment and improvements, the booking ratio for all but the no-frills airlines is still disappointing. In spite of the rapid growth in Internet purchasing of products and services in the world, the buying rate of online flight ticketing remains low. Prior research⁴ investigates the factors that influence online ticket purchasing through a survey of Internet consumers to determine the relationships between convenience, willingness to purchase, price and trust. They found that ticket pricing still remains the major factor influencing consumer purchasing of airline tickets.

Initial Questions

The main question while building this project is “Can machine learning help to predict the price of an air ticket within India based on features such as the source of the travel, the destination of the travel, the departure date, the arrival date, the type of airline and the number of stoppages if any.

In the course of time, several other questions also came to mind we can be explored in further projects :

- Based on the features of the dataset collected, can we predict the amount of time that customers will take to fly from say Kolkata to Bangalore using SpiceJect?
- Can we generate a visual analysis of the price distribution of the various airlines for each destination city?
- Based on the features of the dataset, can we predict the average time taken by Air India to reach Bangalore from Delhi during high passenger traffic hours?

Data

Octoparse scraping tools will be used to extract data from the website. An estimated total of 300261 distinct flight booking options will be extracted from the site. Data will be collected for 50 days, from February 11th to March 31st, 2022. Data source will be secondary data and collected from Ease my trip website.

The various features of the cleaned dataset are explained below:

- **Airline:** The name of the airline company is stored in the airline column. It is a categorical feature having 6 different airlines.
- **Flight:** Flight stores information regarding the plane's flight code. It is a categorical feature.
- **Source City:** City from which the flight takes off. It is a categorical feature having 6 unique cities.
- **Departure Time:** This is a derived categorical feature obtained by grouping time periods into bins. It stores information about the departure time and have 6 unique time labels.
- **Stops:** A categorical feature with 3 distinct values that stores the number of stops between the source and destination cities.
- **Arrival Time:** This is a derived categorical feature created by grouping time intervals into bins. It has six distinct time labels and keeps information about the arrival time.
- **Destination City:** City where the flight will land. It is a categorical feature having 6 unique cities.
- **Class:** A categorical feature that contains information on seat class; it has two distinct values: Business and Economy.
- **Duration:** A continuous feature that displays the overall amount of time it takes to travel between cities in hours.
- **Days Left:** This is a derived characteristic that is calculated by subtracting the trip date by the booking date.
- **Price:** Target variable stores information of the ticket price.

Exploratory Data Analysis

Exploratory data analysis(EDA) is one of the most important steps in machine learning. In order to get the data right and build robust models, I did a lot of EDA on the collected data. The following steps are what I used in arriving at the final dataset for building the machine learning models:

1. Automated Exploratory Data Analysis Using Pandas Profile Report

Dataset statistics		Variable types	
Number of variables	11	CAT	10
Number of observations	10683	NUM	1
Missing cells	2		
Missing cells (%)	< 0.1%		
Duplicate rows	220		
Duplicate rows (%)	2.1%		
Total size in memory	918.2 KiB		
Average record size in memory	88.0 B		

Fig. 1: Pandas profiling report showing the EDA of the extract data.

The above summary output shows the statistical EDA output of the variables in the dataset.

2. Manual Exploratory Data Analysis

A) First I check to see the data types of the features in the dataset, to make sure they are in the right type.

```
In [105]: df.dtypes #checking the data types
```

Airline	object
Date_of_Journey	object
Source	object
Destination	object
Route	object
Dep_Time	object
Arrival_Time	object
Duration	object
Total_Stops	object
Additional_Info	object
Price	int64
dtype:	object

Fig 2: checking the data types

B) Second I checked to see if there missing values present in my dataset:

```
In [106]: df.isna().sum() #Checking null values
```

Airline	0
Date_of_Journey	0
Source	0
Destination	0
Route	1
Dep_Time	0
Arrival_Time	0
Duration	0
Total_Stops	1
Additional_Info	0
Price	0
dtype:	int64

Fig 3: checking for missing values

C) Next I removed the missing values from the dataset

```
In [107]: df.dropna(how='any',inplace=True)
df.isnull().sum()

Out[107]: Airline      0
Date_of_Journey    0
Source            0
Destination       0
Route            0
Dep_Time         0
Arrival_Time     0
Duration         0
Total_Stops      0
Additional_Info   0
Price           0
dtype: int64
```

Fig 4: removing missing values

Feature Engineering

A) Again there was some feature engineering that needed to be done on the dataset in order to get the right data for the analysis. The first feature engineering to be done was to convert ***Date_of_Journey*** feature in the dataset to its appropriate format as datetime with regards to day and month.

```
Date_of_journey

In [109]: df['Date_of_Journey']=pd.to_datetime(df['Date_of_Journey'])
df['Day_of_Journey']=(df['Date_of_Journey']).dt.day
df['Month_of_Journey']=(df['Date_of_Journey']).dt.month

In [110]: df.head(3)

Out[110]:
```

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
0	IndiGo	2019-03-24	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3897
1	Air India	2019-01-05	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2 stops	No info	7662
2	Jet Airways	2019-09-06	Delhi	Cochin	DEL → LKO → BOM → COK	09:25	04:25 10 Jun	19h	2 stops	No info	13882

Fig 5: convert ***Date_of_Journey*** feature to datetime format

I repeated the above strategy for couple of features that needed to be engineered as shown below:

Arrival_time

```
In [114... df['Arrival_hr']=pd.to_datetime(df['Arrival_Time']).dt.hour
df['Arrival_min']=pd.to_datetime(df['Arrival_Time']).dt.minute
```

```
In [115... #we can now drop the 'Arrival_Time'

df.drop(["Arrival_Time"],axis=1,inplace=True)
```

Duration Time

```
In [116... duration=df['Duration'].str.split(' ',expand=True) #split duration datapoints based on space ' '
duration[1].fillna('00m',inplace=True) #fill all "NaN" with '00m'
df['duration_hr']=duration[0].apply(lambda x: x[:-1]) #select the item at index 0 and leave the last one (in
df['duration_min']=duration[1].apply(lambda x: x[:-1]) #select the item at index 1 and leave the last one (in
```

```
In [117... #we can now drop the 'Duration'

df.drop(["Duration"],axis=1,inplace=True)
```

```
In [118... df.head(3)
```

Fig 6: feature engineer on 2 features in the dataset

Data Visualization

The next activity was to perform data visualization in order to have a better visual understanding of my dataset. I then started with visualizing the various airlines and their pricing components and realised that Jet Airways Business has the highest price with Trujet having the lowest

Airline vs Price

```
In [121]: Airprices=df.groupby('Airline')['Price'].mean().sort_values(ascending=False)
plt.figure(figsize=(15,10))
sns.barplot(Airprices.index,Airprices.values)
plt.xticks(rotation=270)
```

```
Out[121]: (array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11]),
<a list of 12 Text major ticklabel objects>)
```

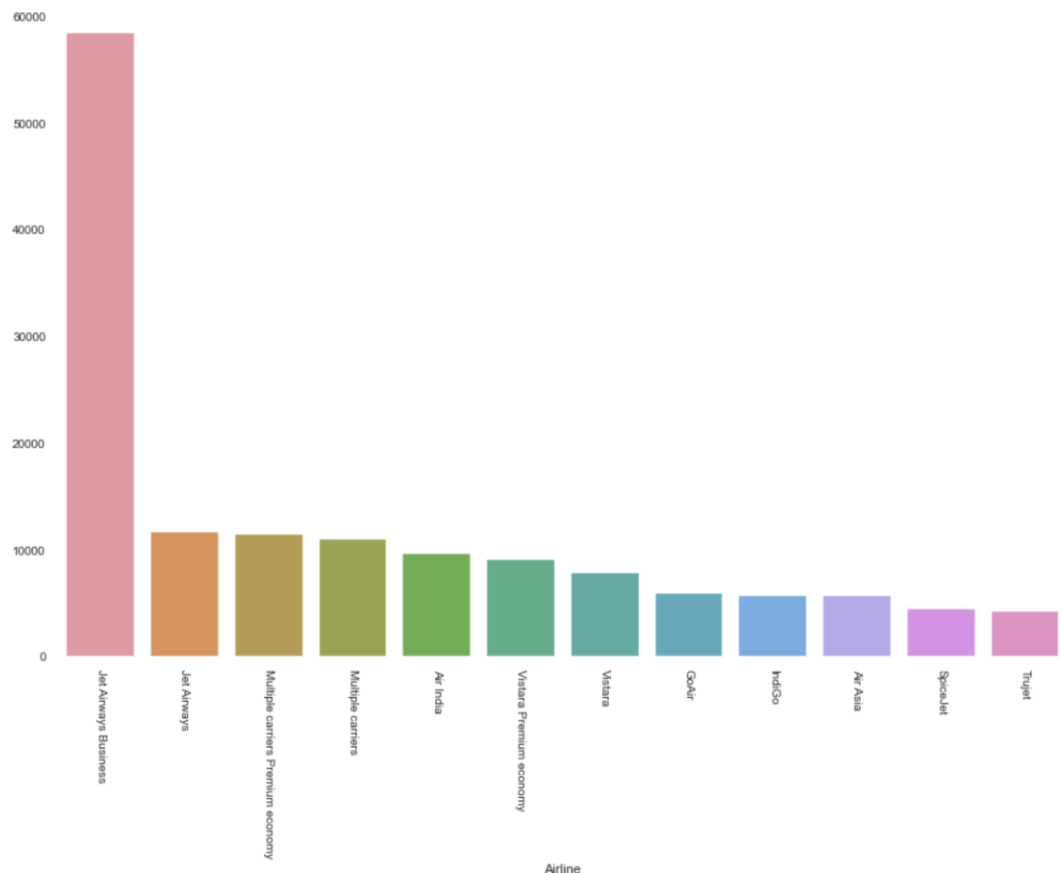


Fig 7: visualization of Airling Vs Pricing

Again based on the number of stops of the airlines:

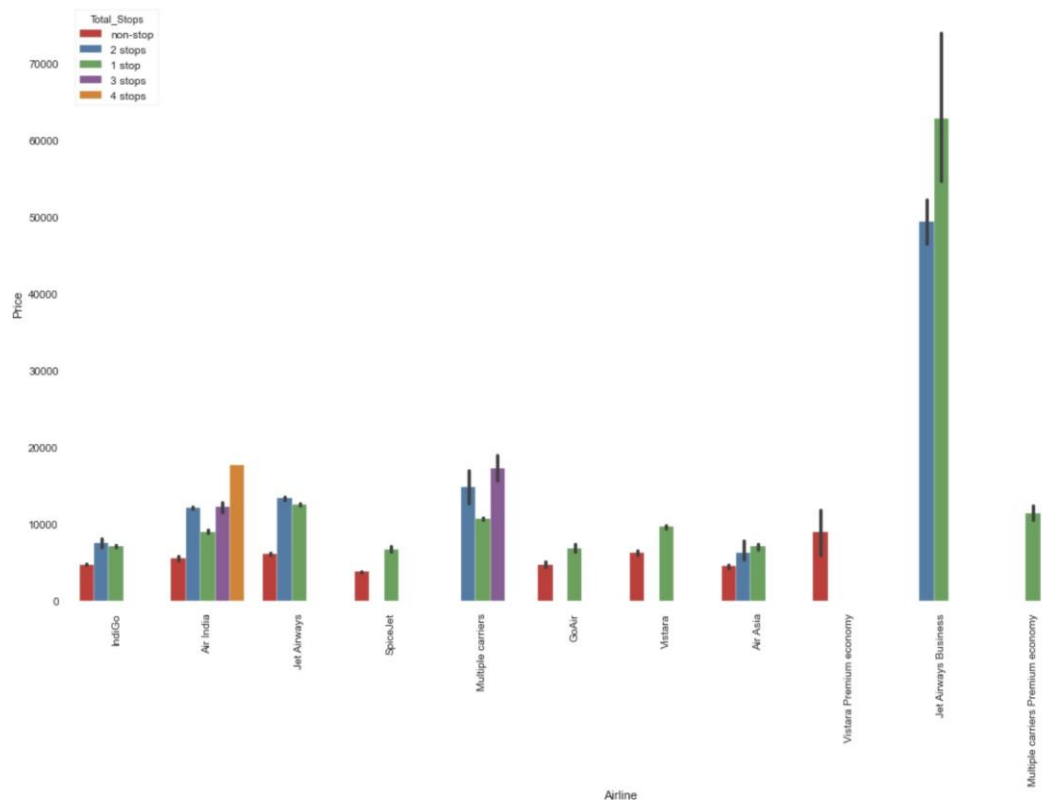
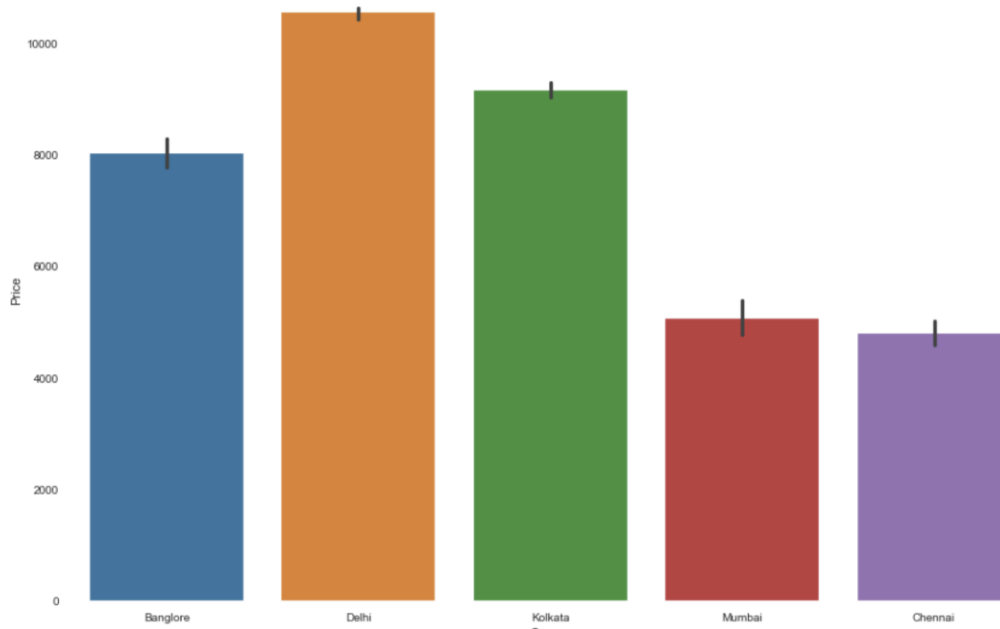


Fig 8: number of stops

One stop and two stops Jet Airways Business is having the highest price

Pricing Vs Destination from the Delhi International Airport:

Fig 9:



Pricing Vs Destination

Checking for correlation in the various features in the dataset:

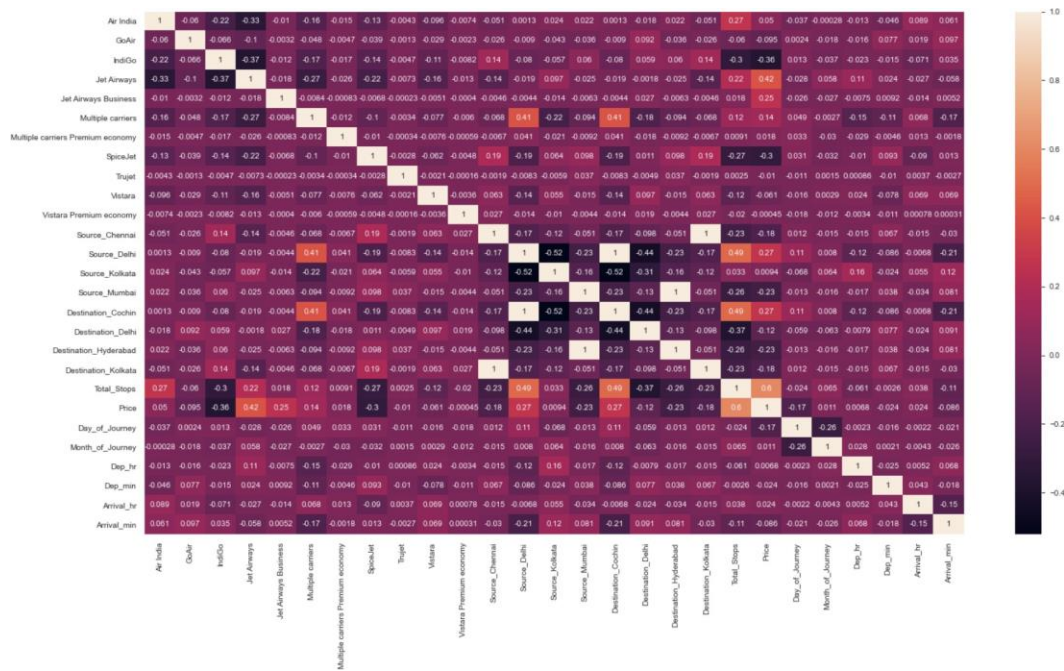


Fig 9: Pricing Vs Destination

Model Building and Hyperparameter Tuning

After the data preprocessing stage and doing all the necessary exploratory data analysis(EDA), I moved to the model building part. At this stage, Since my problem is a regression problem, I choose 5 regression algorithms in building and optimizing my machine learning model. Among the 5 models built, XGBoost quite outperformed the other algorithms with an accuracy of 82% hence I choose that as my base model.

Feature Importance.

To know which of the 28 features are contributing the most in making a good prediction of the flight prices, I checked the feature importance of all the features and the results is shown in fig 10 below:

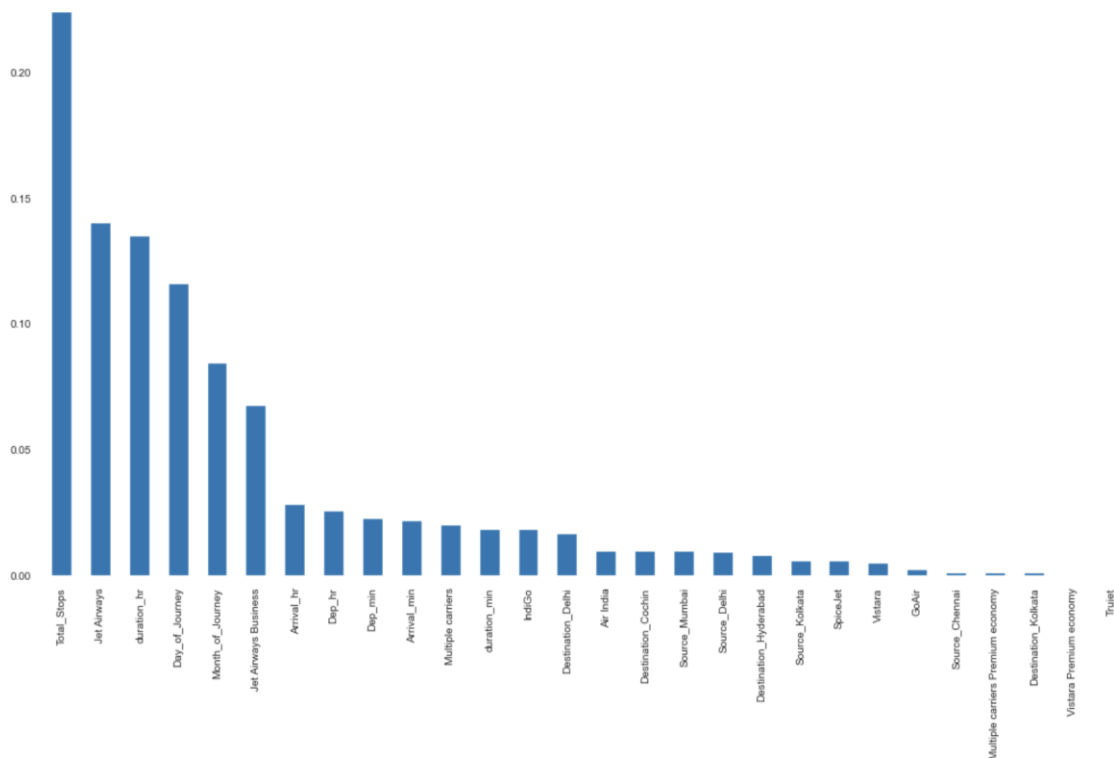


Fig 10: plot of feature importance

From the above plot, I used the top 10 features to continue building the model.

Hyperparameter Tuning

In order to get the best hyperparameters for the model, I performed hyperparameter tuning on the various algorithms.

```
In [178]: from sklearn.model_selection import RandomizedSearchCV

n_estimators = [int(x) for x in np.linspace(start = 80, stop = 1500, num = 10)]
max_features = ['auto', 'sqrt']
max_depth = [int(x) for x in np.linspace(6, 45, num = 5)]
min_samples_split = [2, 5, 10, 15, 100]
min_samples_leaf = [1, 2, 5, 10]

# create random grid
rand_grid={ 'n_estimators': n_estimators,
            'max_features': max_features,
            'max_depth': max_depth,
            'min_samples_split': min_samples_split,
            'min_samples_leaf': min_samples_leaf}

rf=RandomForestRegressor()

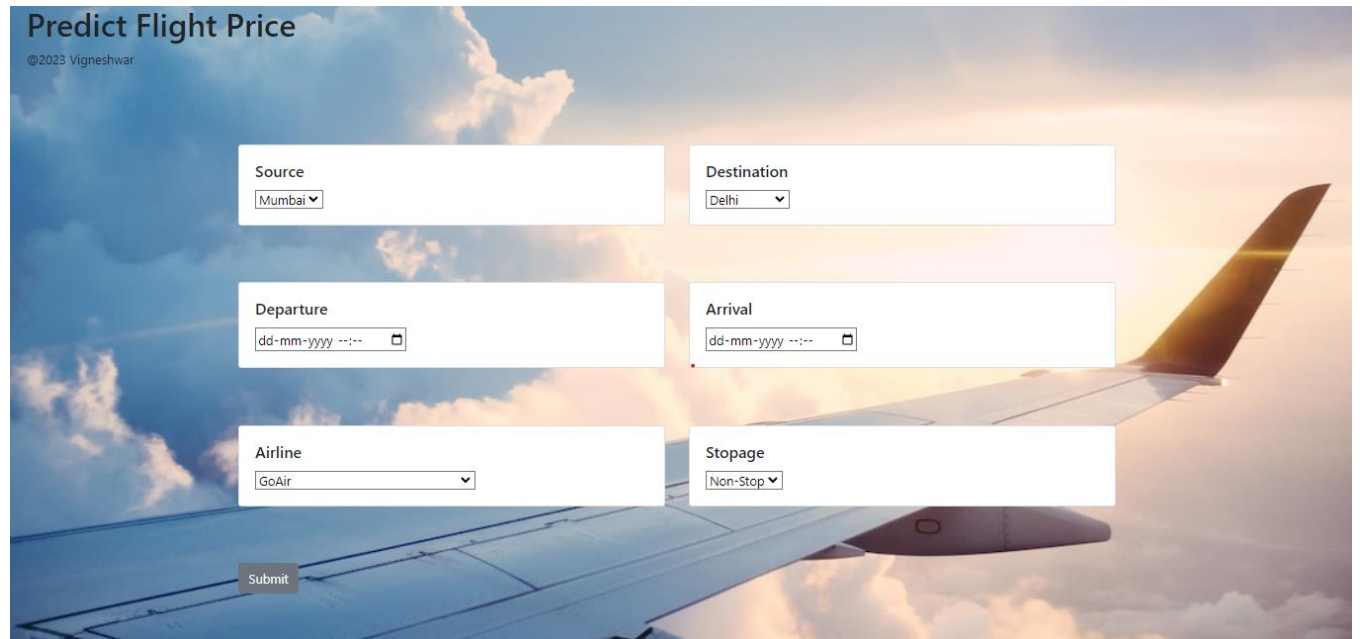
rCV=RandomizedSearchCV(estimator=rf,param_distributions=rand_grid,scoring='neg_mean_squared_error',n_iter=10,
```

Full Analysis

This project was quite involving from collecting the data through web scraping to processing it via exploratory data analysis. Initially I used the MakeMyTrip API to extract the data from the platform, however, I realized the incoming data was not consistent. I then reached out to the team to get premium access to the data on the platform. The results from the premium version of the API was very much what I wanted although it required a lot of cleaning as the data came in the form of json file. I needed to extract every single data point into a comma separated file format in order to work with it.

The results of the analysis and the model proves the robustness of my data preparation. I realized spending time to get the right dataset was as important as spending time to understand the business problem of a machine learning project. After my machine learning model is done, I asked myself, how will I make this model accessible for my stakeholders pan India to be able to use it.

I did a couple of research and found that although I can use cloud platforms like AWS, Microsoft Azure, or Google Cloud, I chose Heroku which gave me free access to the platform to deploy the model.



Predict Flight Price
©2023 Vigneshwar

Source: Mumbai ▼

Destination: Delhi ▼

Departure: dd-mm-yyyy --:-- ☞

Arrival: dd-mm-yyyy --:-- ☞

Airline: GoAir ▼

Stopage: Non-Stop ▼

Submit

After deploying the model, I tested it by allowing students of WPI to interact with it which proved to be robust. The screenshot below gives an overview of the model deployment:

Github: <https://github.com/Vigneshwar-B>

Conclusion:

This project is an eye opener and helping Indian citizens to better make decisions was a great achievement. Overall the data analysis and machine learning model building was a great lesson to go through. The data extraction part was the most part that I learnt the most and hope to learn more in further projects.