
# FLIGHT PRICE PREDICTION

Submitted by:

UJJWAL PRATIK

# ACKNOWLEDGMENT

For this study, I sincerely thank faculties of Data Trained for their detailed teaching method. Topic covered like Machine Learning and Neural language with python has helped a lot. There are also some references taken from YouTube videos and google search.

# INTRODUCTION

- ## Business Problem Framing

  Optimal timing for airline ticket purchasing from the consumer's perspective is challenging principally because buyers have insufficient information for reasoning about future price movements. In this project we simulate various models for computing expected future prices and classifying whether this is the best time to buy the ticket.

- ## . 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).

- ## Review of Literature

- It is very difficult for the customer to purchase a flight ticket at the minimum price. For this several techniques are used to

- obtain the day at which the price of air ticket will be minimum. Most of these techniques are using sophisticated artificial intelligence (AI) research is known as Machine Learning.

- Utilizing AI models, [2] connected PLSR (Partial Least Square Regression) model to acquire the greatest presentation to get the least cost of aircraft ticket buying, having 75.3% precision. Janssen [3] presented a direct quantile blended relapse model to anticipate air ticket costs for cheap tickets numerous prior days take-off. Ren, Yuan, and Yang [4], contemplated the exhibition of Linear Regression (77.06% precision), Naive Bayes (73.06% exactness, SoftMax Regression (76.84% precision) and SVM (80.6% exactness) models in anticipating air ticket costs. Papadakis [5] anticipated that the cost of the ticket drops later on, by accepting the issue as a grouping issue with the assistance of Ripple Down Rule Learner (74.5 % exactness.), Logistic Regression with 69.9% precision and Linear SVM with the (69.4% exactness) Machine Learning models.

- Gini and Groves [2] took the Partial Least Square Regression (PLSR) for developing a model of predicting the best purchase time for flight tickets. The data was collected from major travel journey booking websites from 22 February 2011 to 23 June 2011. Additional data were also collected and are used to check the comparisons of the performances of the final model.

- Janssen [3] built up an expectation model utilizing the Linear Quantile Blended Regression strategy for San Francisco to New York course with existing every day airfares given by www.infare.com. The model utilized two highlights including the number of days left until the take-off date and whether the flight date is at the end of the week or weekday. The model predicts airfare well for the days that are a long way from the take-off date, anyway for a considerable length of time close the take-off date, the expectation isn't compelling.

- Wohlfarth [15] proposed a ticket buying time enhancement model dependent on an extraordinary pre-preparing step known as mocked point processors and information mining systems (arrangement and bunching) and measurable investigation strategy. This system is proposed to change over heterogeneous value arrangement information into added value arrangement direction that can be bolstered to unsupervised grouping calculation. The value direction is bunched into gathering dependent on comparative estimating conduct. Advancement model gauge the value change designs. A tree based order calculation used to choose the best coordinating group and afterward comparing the advancement model.

.

- ## Motivation for the Problem Undertaken

  Deciding whether a Flight is worth the posted price when you see listings online can be difficult. Several factors, including timings, duration, stops, dates, etc. can influence the actual fare of the price. From the perspective of a sites, it is also a dilemma to price a flight appropriately. Based on existing data, the aim is to use machine learning algorithms to develop models for predicting used flight prices.

# Analytical Problem Framing

- ## Mathematical/ Analytical Modeling of the Problem

  I have used various Mathematical and Statistical approach for making this project predict better. I have used Statistical summary for checking and analysing various stats like, number of values counts in dataset of each column, mean, median and mode for each column. Other than that, stats show outliers and skewness in overall dataset. After analysing the data through Statistical Summary, I have applied some Mathematical tools for removing outliers and skewness present in the dataset, Mathematical tools is also used for checking null values and at last making prediction.

- ## Data Sources and their formats

I have scrapped the data of cars from various websites like yatra.com, make my trip, paytm etc. The Datasets contains 2100 rows and 8 columns.

Dataset format using Pandas

| | Unnamed: 0 | Flight | Boarding City | Destination | Departure | Arrival | duration | Stops | Price |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 0 | Air Asia | New Delhi | Mumbai | 18:40 | 06:20\n+ 1 day | 11h 40m | 1 Stop | 5,953 |
| **1** | 1 | Go First | New Delhi | Mumbai | 18:50 | 20:45 | 1h 55m | 1 Stop | 5,954 |
| **2** | 2 | Go First | New Delhi | Mumbai | 09:05 | 11:05 | 2h 00m | 1 Stop | 5,954 |
| **3** | 3 | Go First | New Delhi | Mumbai | 06:15 | 08:20 | 2h 05m | 1 Stop | 5,954 |
| **4** | 4 | Go First | New Delhi | Mumbai | 14:20 | 16:25 | 2h 05m | 1 Stop | 5,954 |

- ## Data Preprocessing Done
  - Importing Libraries
  - Loading the dataset
  - Checking the shape of the data
  - Checking data types
  - Checking Null values
  - Analysing through Statistical Summary
  - Univariate Analysis
  - Bivariate Analysis
  - Multivariate Analysis
  - Splitting the independent and target variable
  - Checking and removing skewness and outliers
  - Building models

- ## Data Inputs- Logic- Output Relationships
  - I am finding partial correlation between each input with the output, also finding high value of coefficient.
  - By extracting weights of the trained model, used connection weights method the relative importance of each input variable on the outputs can be reduced.
  - I have got the validation accuracy almost equal to the training accuracy.

- Hardware and Software Requirements and Tools Used
  - Hardware Requirement:
    - Intel core i5
    - 8 GB Ram
  - Software Requirement:
    - Python 3.x with packages
    - Pandas: Data analysis and manipulation tool
    - NumPy: Provide support for mathematical functions, random number etc
    - Selenium: Web scrapping tool
    - Chrome driver: automated Google Chrome
    - Matplotlib: is a low-level graph plotting library in python that serves as a visualization.
    - Seaborn: is a library mostly used for statistical plotting in python.
    - Scikit-Learn: is an open-source Python library that has powerful tools for data
    - analysis and data mining.

# Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)
  - R2 score: is used to evaluate the performance of a linear regression mode.
  - Linear Regression: Logistic regression is fast and relatively uncomplicated, and it is convenient for you to interpret the results.
  - Lasso: The Lasso is a linear model that estimates sparse coefficients with l1 regularization.
  - Ridge: Ridge regression is an extension of linear regression where the loss function is modified to minimize the complexity of the model.
  - Elastic Net: is a linear regression model trained with both l1 and l2 -norm regularization of the coefficients.
  - Cross-Validation-Score: a model that would just repeat the labels of the samples that it has just seen would have a perfect score but would fail to predict anything useful on yet-unseen data.
  - Grid Search CV: This function helps to loop through predefined hyperparameters and fit your estimator (model) on your training set.
  - Mean Squared Error: this metric gives an indication of how good a model fits a given dataset.
  - Root Mean Squared error: is a frequently used measure of the differences between values (sample or population values) predicted by a model or an estimator and the values observed.
  - Label Encoder: Label Encoding refers to converting the labels into numeric form.
  - Standard Scaler: Standard Scaler. Standard Scaler helps to get standardized distribution, with a zero mean and standard deviation of one (unit variance).
  - Random Forest Regressor: A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting

- Testing of Identified Approaches (Algorithms)
  - Train Test Split
  - Linear Regression
  - Lasso Regularization
  - Ridge Regularization
  - Elastic Net Regularization
  - Grid Search CV
  - Cross Validation
  - Random Forest Regressor

- Run and Evaluate selected models
  - Linear Regression

```
r2 score 1.0
error
mean absolute error 8.51805398702882e-15
mean squared error 1.0824674211832703e-28
root mean squared error 1.0404169458362691e-14
```

  - Lasso Regularization

```
r2 score 0.9896507987856674
error
mean absolute error 0.7716643245023824
mean squared error 1.0077197922734296
root mean squared error 1.00385247535354
```

  - Ridge Regularization

```
r2 score 0.9999996133335763
error
mean absolute error 0.004704199082138801
mean squared error 3.7650384801730695e-05
root mean squared error 0.006135990938856632
```

  - Elastic Net Regularization

```
r2 score 0.8675862507418696
error
mean absolute error 2.7792173447488633
mean squared error 12.89335796387398
root mean squared error 3.590732232271571
```

  - Random Forest Regressor

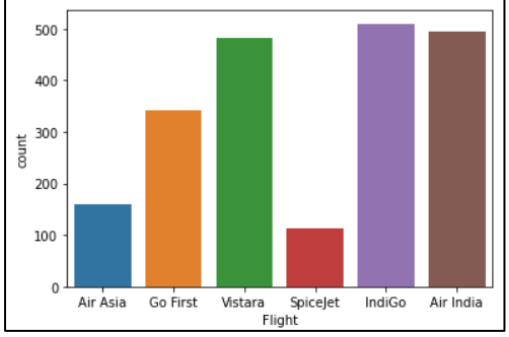
```
r2 score 99.96922109940547
cv score 99.96708725481447
```

  - Lasso Regularization (Hyper parameter tuning)

```
r2 score: 98.96507987856674
cross val score: 98.96951672769555
```

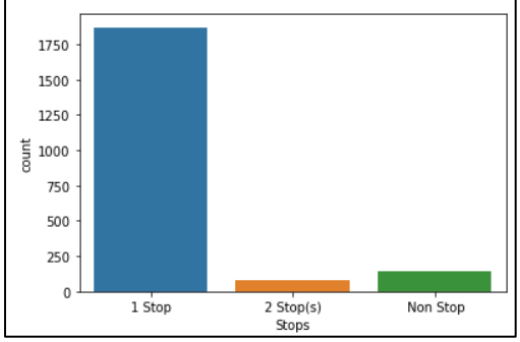
- Key Metrics for success in solving problem under consideration
  - Analysed data for any outliers
  - Analysed data for any skewness
  - Done Label Encoding for converting string value to numerical form.
  - Cross Validation for cross validates the accuracy-score from overfitting.
  - Hyper parameter tuning using Grid Search Cv for making the prediction better
  -
- Visualizations
  - Flight

```
IndiGo        510
Air India     494
Vistara       481
Go First      343
Air Asia      160
SpiceJet      112
Name: Flight, dtype: int64
```

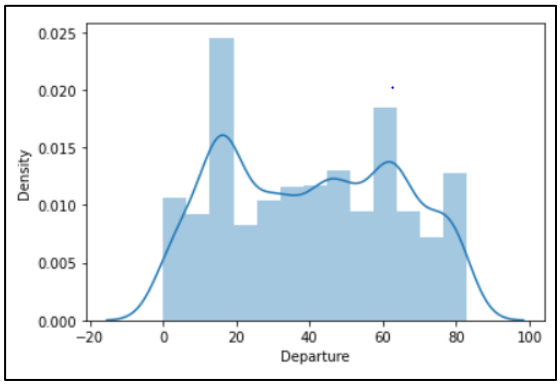



  - Stops

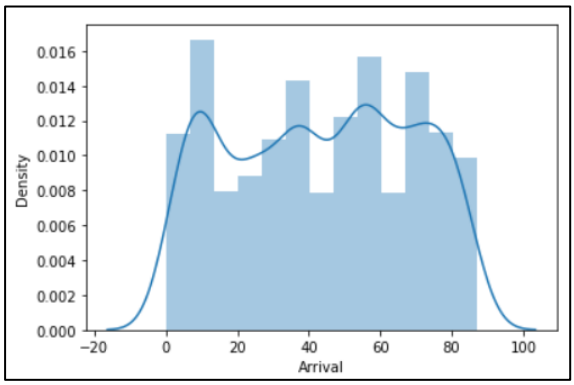
```
1 Stop        1871
 Non Stop      145
2 Stop(s)       84
Name: Stops, dtype: int64
```



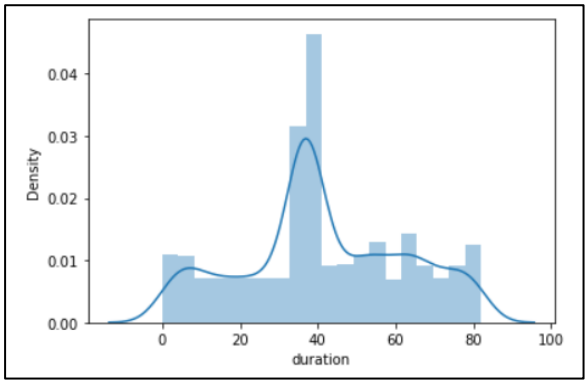
- Departure



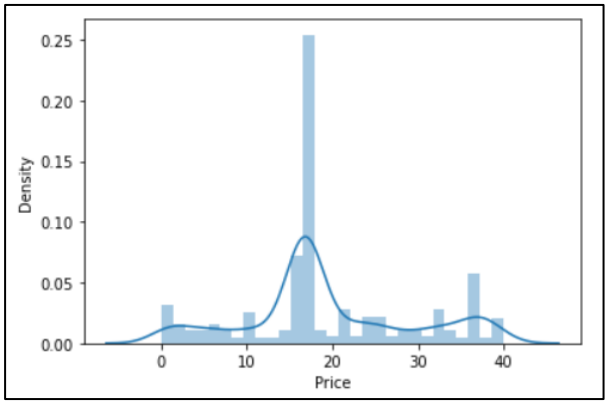

- Arrival




- Duration




- Price



- Interpretation of the Results

  Given these considerations, we are able to predict if the price of a flight with 98% of accuracy after cross validation and hyper parameter tuning. This means our model can predict the price of a flight so that traveller can focus their area in very right directions to get the best out of it.

# CONCLUSION

- ## Key Findings and Conclusions of the Study

  During my analysis I got the highest accuracy with Lasso Regularization that is 98% and Ridge regularization that is also 99%, but it can be due to overfitting also. With hyper parameter tuning I got minimum difference in accuracy and cross validation is for Lasso Regularization.

- ## Learning Outcomes of the Study in respect of Data Science

  This study gives me opportunity for lots of learning starting from various types of plotting like histograms, boxplot, scatterplot, line chart and many more graphs. These graphs helped me to analyse different aspects of data like outlier, skewness, correlation etc. It also helped me to learn how to apply various model techniques on data and enable predications

- ## Limitations of this work and Scope for Future Work

  For better performance, we plan to judiciously design deep learning network structures, use adaptive learning rates and train on clusters of data rather than the whole dataset. To correct for overfitting in Random Forest, different selections of features and number of trees will be tested to check for change in performance.