FLIGHT PRICE PREDICTION USING MACHINE LEARNING

Project report submitted in partial fulfilment of the requirement for the degree of Bachelor of Technology

In

Computer Science and Engineering

By

Yadunandan Sood (181297)

UNDER THE SUPERVISION OF

Dr. Pradeep Kumar Gupta

to

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology, Waknaghat,

173234, Himachal Pradesh

CERTIFICATE

Candidate's Declaration

I hereby declare that the work presented in this report entitled **"Flight price prediction using machine learning "** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from August 2021 to May 2022 under the supervision of **Dr. Pradeep Kumar Gupta,** Associate Professor Department of Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Yadunandan Sood (181297)

This is to certify that the above statement made by the candidate is true to the best of my knowledge

Dr. Pradeep Kumar Gupta
Associate Professor
Computer Science & Engineering and Information Technology
Jaypee University of Information Technology, Waknaghat,

AKCNOWLEDGEMENT

Firstly, I express my heartiest thanks and gratefulness to almighty God for His divine blessing makes us possible to complete the project work successfully.

I really grateful and wish my profound my indebtedness to Supervisor **Dr Pradeep Kumar Gupta, Designation**, Department of CSE Jaypee University of Information Technology,Wakhnaghat. Deep Knowledge & keen interest of my supervisor in the field of **Computational and Machine Intelligence** to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stage have made it possible to complete this project.

I would like to express my heartiest gratitude to **Dr Pradeep Kumar Gupta,** Department of CSE, for his kind help to finish my project.

I would also generously welcome each one of those individuals who have helped me straight forwardly or in a roundabout way in making this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

Yadunandan Sood

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

- **EDA:** Exploratary Data Analysis
- **C:** Celsius
- **F:** Fahrenheit
- **ML:** Machine Learning
- **i.e:** that is
- **RAM:** Random Access Memory
- **CPU:** Central Processing Unit
- **GPU:** Graphics Processing Unit
- **KNN:** K-Nearest Neighbours
- **INR:** Indian Rupees

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Machine learning is a field of study that works with computer algorithms and data. It basically builds a tool on sample data and then predicts or take decisions about the same data without the need of specific programming. ML has its use in various industries including aviation industry where it is used to predict flight prices.

In today's world, flights have become a common mode of transportation for humans. It is faster way to reach destinations, thus saving a lot of time. But also not everyone can afford a plane ticket and fly high in the sky. The ticket prices for a plane journey are very high when compared to that of a train or a bus. But irrespective of that, nowadays the number of people using flights have increased massively. Thus it becomes very hectic to maintain the prices of tickets with changing conditions. The airline companies use various complex techniques to predict the various flight prices using various factors present at that time. These factors commonly include market related issues, financial issues and various social issues. So it becomes difficult for them to do so. Also the consumer has no such tool in hand to have an idea about the ticket prices, that could very helpful.

This project is designed and developed keeping in mind this problem and develops an algorithm that predicts various flight prices keeping in mind various factors that affects them. This can help the airline companies to check what prices they could maintain. Also it would help the customer to predict future flight ticket prices, which would help in planning the trip accordingly.

Chapter 1 INTRODUCTION

1.1 Introduction

Machine learning is the part of artificial intelligence that provides a computer the ability to automatically learn with experience without the need of external programming. It is one of the hot ,most in demand research topic in computer science engineering. It can provide intelligence to the machines with the help of various tools and techniques. ML uses programs that can access data and learn for themselves using the same. In today's world machine learning has become an important part of everything we are a part of. It helps various enterprises for the development of new products and also helps to have an idea about the customer trends. Many MNC's have made ML an important part of their organisation.

The learning process for an ML algorithm begins and ends with data. Therefore, data is the most important part of a model. This data has an effect on future predictions and decisions. The primary objective of ML is that a computer system learns automatically without human assistance.

Machine Learning is categorized mainly into 4 approaches based on how an algorithm goes through the process of learning.

- **Supervised machine learning**: Analysis of labelled data and then training on the same to predict future events. This system can provide output for new inputs if sufficient training is done.
- **Unsupervised machine learning:** The information used in training is not labelled and neither classified. This system is used to study different patterns and connections in data.
- **Semi-supervised machine learning**: Falls in between the above two as it uses small amount of labelled data and larger amount of the unlabelled data.
- **Reinforcement machine learning**: In this system the program learns to behave in a particular environment by performing actions and then seeing the results.

Decision tree is a powerful tool used for classification and prediction. It resembles the shape of a flowchart. They classify instances from the root node towards the leaf node. A condition is specified at each node(internal node) based on which the tree further divides into edges.

The end of the branch, i.e a branch where no further splits occur is called the leaf or the decision.

Root Nodes: First node at the start of the tree. Division starts here.

Decision Nodes: Nodes that we get after splitting of root nodes.

Sub-tree: It is basically a sub part of the whole tree.

Pruning: Cutting of some nodes to stop overfitting.

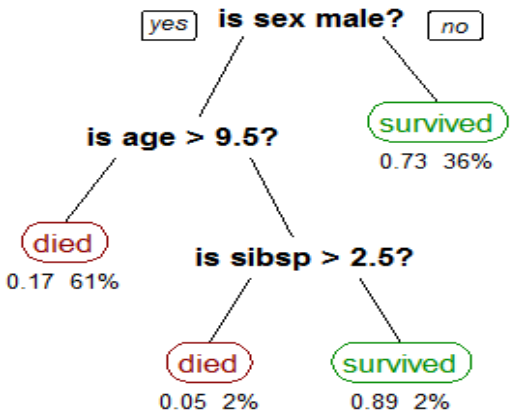

Figure 1.1: Depiction of decision tree

Random forest algorithm is a popular model in machine learning that comes under the supervised learning approach. This model can be used for both regression and classification problems. This model takes together a number of classification models or commonly known as classifiers to solve a problem and hence improve the accuracy of the same. Random forests contains a number of decision trees and predicts by taking the collective output of them ,i.e it takes the mean or average of the outputs from various trees. Also it can be inferred that by increasing the number of trees, increases accuracy. A main fundamental behind this classifier is that a large number of uncorrelated trees operating as a single unit have to be taken. This sense of least relationship between the trees would help to maximize the results. Example like in investments with low relatedness (like bonds and stocks) , form a portfolio that is much greater than individual sum of its parts.

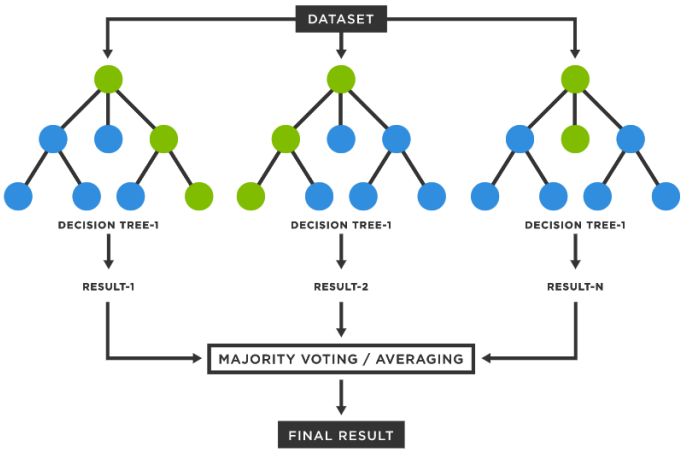

Figure 1.2: Random forest

1.2 Problem Statement

In today's world flight companies try to change flight prices according to their needs and present condition. In this way they can maximize their profits. Regression based solutions to this problem are common but they come with their own disadvantage. With the technological advancements and the load increase on these flight companies , regression does not give the desired results and offers low accuracy. Customers on the other hand do not have a tool to have an idea about the ticket prices that could help them in planning accordingly. Also many customers don't know the right time to book a ticket and sometimes have to pay more for a ticket. Consider an example, the least expensive ticket will change its value over a period and its value may be high or low depending on the time like summers ,winters etc or according to day ,night or evening. This happens because the underlying goal of the plane carrier is to maximize its income. But on the other side the purchaser try to get a ticket at a lower price so book them days before the actual takeoff. But sometimes the price of a particular ticket does not fluctuate and he consumer ends up paying more for the same.

So random forest machine learning model would be used for the same problem to get the prices of flight tickets..This was a brief into the problem which will be discussed below in the report and a possible solution to this problem will be provided.

1.3 Objectives

This project aims at developing a model that would predict the prices of the flight from the starting point to the point of destination and should be efficient in working. The price prediction would depend on various factors such as distance, flight time, number of stops etc. These factors help create a pattern to decide the price of flight ,and the machine learning models get trained on this pattern to make predictions in future. A model would be created by applying machine learning algorithms to the collected data or the historical data related to flights. This system would give people the idea about the trends that prices follow and provide a predicted value.

Nowadays the airline corporations are using complex methods to get airfare prices in a changing fashion. These strategies take into account several commercial, social and marketing factors. Due to the high variedness of the pricing models applied by the airlines, it is very difficult for a customer to buy a ticket in the lowest price possible. The model should be able to give an insight on the fare prices for the desired flight. Also it should be able to predict the prices for maximum number of airline companies, i.e it should be able to adopt to new changes or better say to new data. The accuracy offered by the model should be high so that it could be trusted and customers don't lose their interests in knowing the prices beforehand. Also the model should not have any unnecessary lags or stops in its working as the scenarios its dealing with does not allow any slip ups.

1.4 Methodology

The main methodology of this project revolves around using the random forest algorithm for classification and prediction. First and foremost we begin our project with collecting the required data and applying the exploratory data analysis. This is nothing but initial investigations on data to look for any patterns or to check if there is something wrong with it with the help of statistics and graphical ways. After this we move on to data preprocessing. Data preprocessing is the step where data is changed or encoded so that the machine could easily understand it. It also involves the process of dealing with missing values, duplicate values or inconsistent values. Then we move onto feature selection. A feature in ML is an attribute or characteristic under observation. For example for a flight power , fuel capacity,

number of stops, weather conditions can be considered as features. So in feature selection we pick out the best features or the features which when applied to the model gives results with high accuracy. Feature selection is an important of supervised learning because if we do not go ahead with the right features the results may vary a lot that would simply nullify the purpose of this flight price predictor.

The next step from here is to decide which model to apply to bear fruitful results. We experimented with models like linear regression, decision trees, multilayer perceptron etc. After experimenting with all these algorithms we move forward to give our approach for the problem at hand using random forests classifier.

1.5 Organization

In this part we will explain the organization of the chapter wise formatting of this whole report. In chapter one we have already seen the Introduction, Problem Statement. After that we saw the objectives and the methodology of the report. Now we stand in the Organization section that would explain the chapters and topics involved further down in this report.

Chapter 2 will be presenting a brief overview of the literature survey process. This will include some of the resources we used in the past to gain necessary knowledge in view of this project and other work we came across through the community that set us forward to reach to this stage where we were able to complete this project and develop a basework to solve these type of problems in the future.Then will follow Chapter 3 where we would be discussing about how our model is designed and developed. First we would discuss abouth the dataset that we used in making this report and see some basic features.We will discuss about the basic framework which help us lead to random forest algorithm that include decision tree and feature selection. We would take a look at the various models we applied to draw comparisons with the random forest and their respective accuracies. We have explained them a bit to give the reader a better understanding of them. After that we have also listed some of the mathematical formulas used in this report.

In chapter 4 we have shown how our system performed and also compared them with the other systems used earlier like linear regression ,decision tree etc. Then we begin to showcase the outputs at various stages of our project work that include various graphs and figures. This would conclude with the final output of our model.

Finally, in chapter 5 we started the conclusion part and explained about what we managed to achieve in the project and how this model was the perfect way for us to learn and explore in this field. After that we proposed some future plans that can be achieved through this project and made an extension to it.

Chapter 2 LITERATURE SURVEY

Travelling by the mode of airplanes from one place to another is perhaps the easiest and most convenient ways but selecting the right type of flight with minimum ticket expense and the perfect route is a cumbersome process for a customer. Hence sophisticated Artificial Intelligence technique called Machine Learning provides a suitable and convenient platform to the users for selecting the right type of flight with minimum price and suitable route. Flight ticket price prediction is a challenging task as the various factors/variables determining the cost can change constantly over time.[1]

Flight prices can vary drastically depending on whether the day was a festival, weekday, weekend, peak season etc. Traditional factor like the total distance between source and destination is not the sole factor which determines the price but has a broader aspect depending on a number of factors. Generally the flights that have departure during weekends have a higher price than those which departure during weekdays [5].Also flight prices during festive season are generally lower. The route and number of stops also play a major role in determining the price of the flights. Features like total number of stops, Duration hours, journey day are some of the most important features for the prediction purpose as the price of the flight tickets is highly dependent on these types of features. To find the most important features correlation heatmap is used which gives the correlation values and higher the correlation value between two features higher are the chances of one of them being removed.

Various Machine Learning algorithms have been used to predict the flight ticket price whether on international level or national level. A number of datasets have also been used for the same although the main essence of prediction is the same for all which deals with proper preprocessing and feature selection. Various algorithms like Linear Regression, Decision tree, Support Vector Machines (SVMs) and Neural network are commonly used for building this model. Among all the algorithms mentioned above, Random forest gives the best accuracy. SVMs fail to give a desired accuracy on regression analysis however it can be used for classifying the price as high or low.[3]

The following algorithms were tested for accuracy among which Decision tress proved to be the best:

1) Linear Regression

2) Decision tree Regressor

3) Random forest

Linear Regression however does not prove to be a suitable algorithm and only gives an accuracy of 61 percent. As the numbers of features are more in number and complex in nature, linear regression fails to give the expected accuracy. Decision tree however gives an accuracy of 72 percent which is quite good for a dataset of this size.[2]

Random forest is the most suitable algorithm as it uses multiple decision trees to get a final output and thus results in maximum accuracy among the above mentioned algorithms. It gives an accuracy of 79 percent and along with hyper-parameter tuning the accuracy jumps to almost 81 percent. Random forest has almost equivalent parameters as a decision tree.

In order to find the best features for fitting the model, feature selection is very important as it investigates the degree of impact of each feature on prediction output. Correlation between the features determines how closely the features are related to one another. One of the two highly correlated features are removed so as to decrease the size of features and decrease the complexity in fitting the model.[2]

ML Model	*Accuracy (%)*	*Execution Time (sec)*
Multilayer Perceptron	75.49	16.31
Generalized Regression Neural Network	66.25	0.14
Extreme Learning Machine	67.18	0.05
Random Forest Regression Tree	**79.49**	10.6
Regression Tree	78.76	0.06
Bagging Regression Tree	77.50	15.07
Regression SVM (Polynomial)	78.12	0.87
Regression SVM (Linear)	44.95	0.42
Linear Regression	57.19	0.23

The above table represents the various ML models along with their accuracies and execution time. This table was one of the motivating factors to go with Random forest due to its high accuracy.

Various evaluation metrics have been used to get the accuracy and model performance which helps us to improve the model by accordingly adjusting the parameters or algorithms. Depending on these metrics we find a suitable algorithm.[4]

The various evaluation metrics used are:

1) MAE (mean absolute error)

2) MSE (mean square error)

3) RMSE (root mean square error)

Chapter 3 SYSTEM DEVELOPMENT

3.1 Design and Algorithm

The main algorithm used in this report is random forest. But before explaining that we would explain little bits about the other stages of our model too, i.e exploratory data analysis , data preprocessing and feature selection. Also we would try and develop a comparison between decision trees and random forest and see why the latter is more efficient. Also we have decided to make a webapp for the same so that the model becomes more expressive and easy to use. We used HTML,CSS Bootstrap and Flask to create the same. These concepts will be elaborated further down in the report.

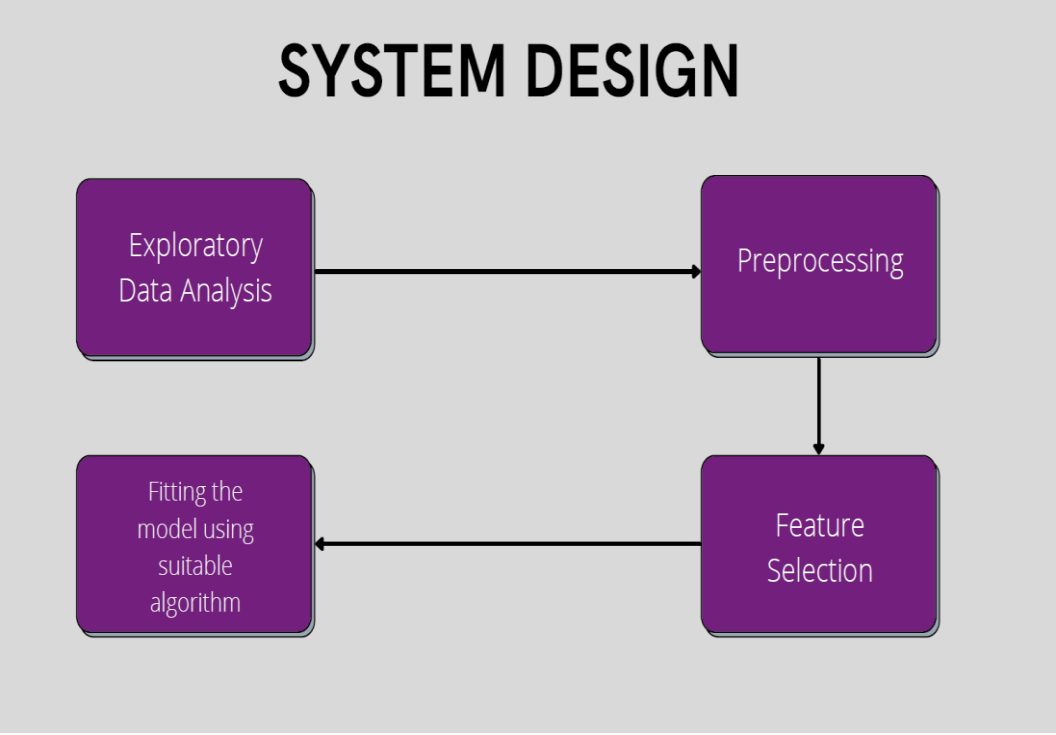

Figure 3.1: System Design

3.1.1 Exploratory Data Analysis

Exploratory data analysis is the process of critical inspection on data so that we can have an insight on the patterns of dataset, to check for any anomalies with the help of graphs and statistical analysis. It is a very great step to understand the dataset and gather as many information from it. It helps to determine how to manilpulate data to get the answers we need. EDA was originally developed by the mathematician John Turkey in the 1970's.

Some tools and techniques that can be used for the same are :

- Clustering and dimension reduction techniques: These help create graphical representations of data high in dimensionality.
- Univariate and bivariate analysis: These help to find a relation between the input variable and the target variable.
- K-means clustering: An unsupervised learning model where K groups are made based on distance from each group's centroid.
- Prediction models like regression, that use stats and data for outcomes.

We can use different functions available in the python pandas library to begin our EDA.

The ".head()" and the ".tail()" functions show the first five and the last five rows from the dataset respectively. It is a good practice to do so as it given an idea about the pattern of data.

We can also know abouth the total of rows and columns in any dataset using the ".shape" function in pandas. We can also use ".info()" function which help to know the columns and their data types, and also helps to find out whether any column contains null values or not. The "describe()" is like the most handy one among all these. It returns various information about the data like count, mean, minimum and maximum values ,standard deviation etc.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 458 entries, 0 to 457
Data columns (total 9 columns):
Name        object
Team        object
Number      float64
Position    object
Age         float64
Height      object
Weight      float64
College     object
Salary      float64
dtypes: float64(4), object(5)
memory usage: 32.3+ KB
```

Figure 3.2: Example of Info function

DESCRIBE

	attr1	attr2	attr3	attr4	attr5 \
count	4.393476e+06	4.393476e+06	4.393476e+06	4.393476e+06	4.393476e+06
mean	2.669727e-02	2.907766e-02	4.297700e-02	5.193186e-02	1.914401e-01
std	1.068498e-02	1.207357e-02	1.529949e-02	2.227283e-02	1.194076e-01
min	5.700000e-03	6.500000e-03	8.200000e-03	5.300000e-03	8.700000e-03
25%	1.940000e-02	2.030000e-02	3.060000e-02	3.350000e-02	1.084000e-01
50%	2.430000e-02	2.600000e-02	4.020000e-02	4.990000e-02	1.565000e-01
75%	3.410000e-02	3.780000e-02	5.380000e-02	7.010000e-02	2.354000e-01
max	8.930000e-02	8.770000e-02	1.028000e-01	1.000000e-01	6.818000e-01

Figure 3.3: Example of Describe function

Data visualization also comes under the EDA as it helps the user to get a graphical insight into the data. This is basically divided into three categories:

- Univariate analysis: This displays all the observations in data around a single data variable. Ex: line plot, histogram etc
- Bivariate analysis: Reveals relationship between two data variables. Ex: heat maps, box plots etc
- Multivariate analysis: reveals relationship between more than two sets of variables. Ex: violin plots, box plots, histograms etc.

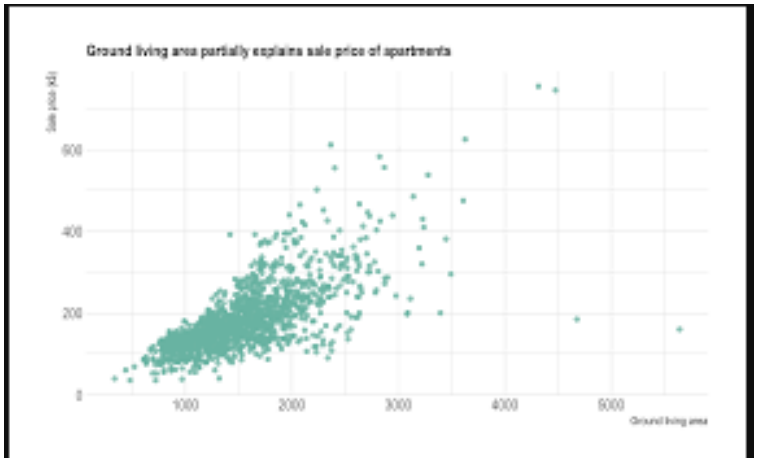

Figure 3.4: Scatter plot

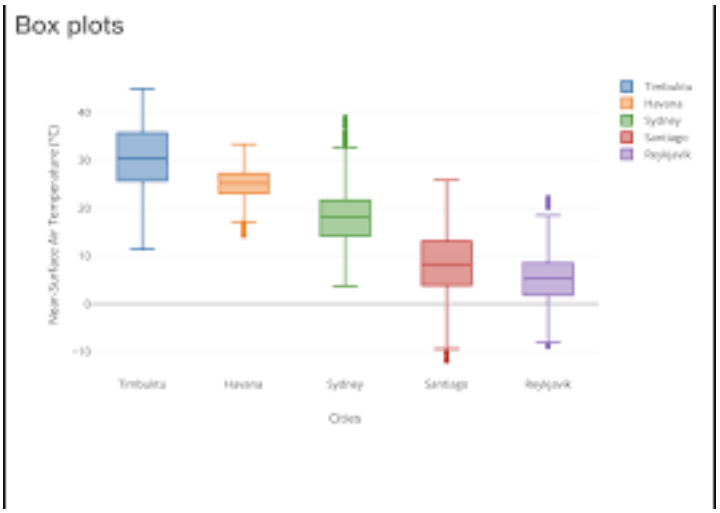

Figure 3.5: Box plots

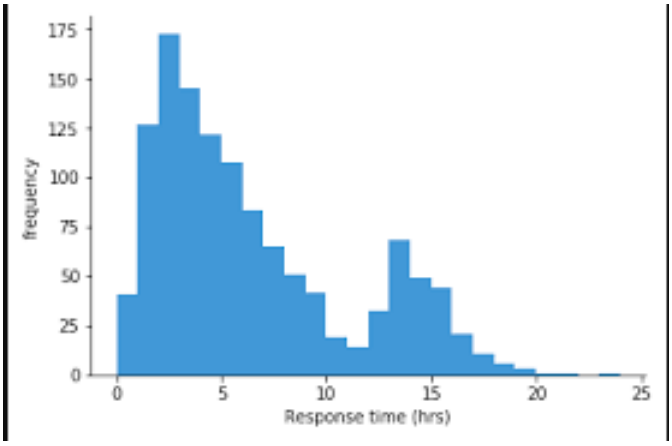

Figure 3.6: Histograms

3.1.2 Data Preprocessing:

Whenever we think of datasets , a large number of rows and columns pop into our heads. But this is not the case always because data can be in structured ,unstructured, tables, images, audio files, videos etc. Now as we all know our computers do not understand anything other than the language of 1's and 0's. So it will be utterly make no sense to pass set of videos or images to our model. Also a dataset may contain missing values, duplicates that are of no use to the model or can cause errors. So here comes the part of preprocessing. EDA and preprocessing are closely related terms and sometimes used in the same way.

In any ML procedure, data preprocessing is the step where our data is transformed or in complex terms encoded ,to bring it to a state so that the computer program can understand it. The columns or the attributes of the data can be consumed or understood by the algorithm after preprocessing. Also it helps in cleaning the data i.e getting rid of the various null or missing values and it helps in increasing the accuracy and efficiency of the model.

This whole process can be divided into the following sub categories:

- **Getting the dataset:** The collection of data for a particular use is called the dataset. As a machine learning model completely depends on data so we have to make sure we acquire the best possible data out there.
- **Importing libraries**: Some python libraries that we need in our project foe some specific jobs have to be integrated. This is called importing libraries. Mainly there is use of three python libraries

  Numpy: Used for mathematical calculations in the code.

  Pandas: Most common and useful library. Used for data manipulation and analysis of dataset.

  Matplotlib: Used for plotting various charts using python. Works with the help of a sub library pyplot.
- **Importing datasets:** The step where we import required datasets for the model
- **Handling missing/inconsistent/duplicate data:** It is a common thing to have missing values in a dataset. The cause to it can be anything ,whether it was by default in the dataset of happened during data collection. But it is our responsibility to handle these missing values.

  1. Eliminate the rows having missing data: It is a simple and effective strategy. This could also fail if many objects have missing values.

Sometimes the feature has to be eliminated if its has mostly missing values.

2. Estimation of the missing values: If only some values are missing then various interpolation methods can be used to fill up these missing values. But in general dealing with these missing values is done by filling them with median ,mode or mean from the respective column.

	Name	State	Gender	Score
0	George	Arizona	M	63
1	Andrea	Georgia	F	48
2	micheal	Newyork	M	56
3	maggie	Indiana	F	75
4	Ravi	Florida	M	NaN
5	Xien	California	M	77
6	Jalpa	NaN	NaN	NaN
7	NaN	NaN	NaN	NaN

Figure 3.7: Illustration of missing values

3. Duplicate values: Deduplication is an often used term that refers to the process of dealing with duplicate values. Duplication of values is also another problem with data. In real world example it can happen when a person fills in his student details twice when asked by the company. The reason behind removing of duplicates is that it should not give a particular data object an edge over the other or in simple words a bias while learning algorithms.

4. Inconsistent values: Like duplicate values, finding inconsistent data in a dataset is highly common. For example the flight number in some rows could be sometimes written under the passenger phone number. The cause to this could simply be a human error or the information was not read correctly at the time of filling the entries.

- **Encoding categorical data:** Categorical data is that whose values are taken from a pre defined set of values. A simple example can be months of a year :{January, February, March……}

  as its values is always taken form this set. Now as an algorithm works on number and maths it is necessary to convert these into suitable format and can be done using various methods like LabelEncoder(), OneHotEncoder etc.

- **Splitting of data set into training and test set:** If we train our model on a dataset but when it comes to testing we use a completely different dataset we would create difficulties for the model. Doing this would also decrease the performance of our model. So we make a model that does well on the training and on the test set. So the test set can be defined as a subset to the training set to test our model on. Validation data is also another term used in ML. We use this data to improve hyperparameter tuning. The model does not learn on this validation data set.

  Split Ratio- It is largely dependent on the model we our building and on the dataset. If a lot of training is required then we use a larger portion of the data for training purposes ,example image data as it contains millions of features.

Figure 3.8: Split ratio

3.1.3 Feature Selection

Before jumping onto feature selection first lets take a look at what do we mean by the term features. A dataset is a collection of various objects that are defines by a number of features, which simply tell us about the different characteristics of our data. Features are also called attributes, fields or variables. Like for example the mass, height of a person can be features of a dataset regarding lifestyles of various people. A feature is an individual characteristic of a phenomenon. We have to choose unbiased and independent features for ML models so that we can have accurate regression and classification. Features are predominantly

numerical but in some cases they can be graphical or strings. For example, for a scooter, colour, weight, mileage cane be observed as features. In a speech recognition system noise,sounds,power can be used a features. In algorithms using spam detection certain email headers, structures, frequency of words can be used as features. Also in character recognition, the length of a particular letter, the boldness of the letters, the spaces between words are used as or can be used as features.

Now features can be:

1. **Categorical**: These are the features whose values are chosen from a predefined set of values. Example dates in a month {1,2,3,4,5,……..} is a categorical value as it is chosen always from this set. The Boolean set is also an example of categorical data.
    i. Nominal data: These are the values that don't have any specific order. Example a new scooter comes in three color- green ,blue, red.
    ii. Ordinal data: These are the values that have a natural order within them but the difference on scale is not defined. Example when we go for buying clothes the sizes have a natural order small<medium<large but this does not ensure that the difference among them would also be the same.

2. **Numerical**: These are the features whose values are either discrete or continuous. They represented in the form of numbers and mostly posses their properties. Example speed of a plane or the number of steps we walk in a day.
    i. Interval data: This contains a defined unit of measurement and the difference between the data values is meaningful. Example include dates of calender ,temperature in C or F.
    ii. Ratio : This also contains defined unit of measurement but both differences and the ratio are meaningful. Examples Age, mass , height etc.

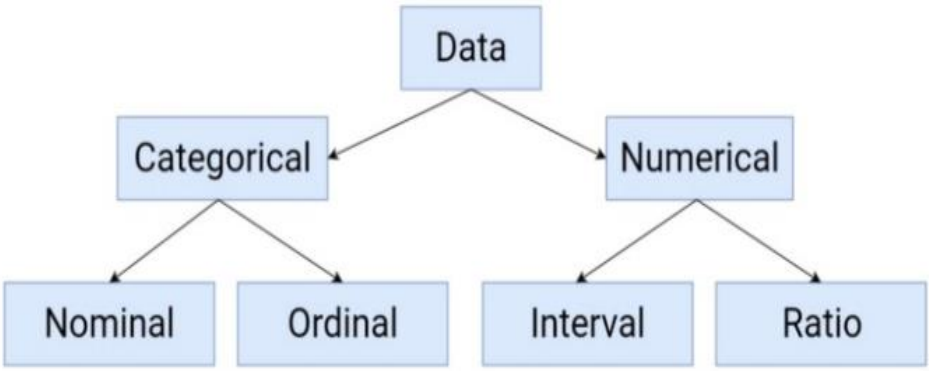

Figure 3.9: Types of features

Features can also be represented as feature vectors. In machine learning a feature vector is an n dimensional vector that represent some object. Many ML models require numerical equivalent of an object to perform statistical processing. Example in order to represent images we need to have numerical representations and that is provided y feature vector ,which contain the pixels of on image. Feature vectors usually combine with a set of weights and use dot product to produce a linear predictor function in order to determine a score for making a prediction. The space vector associated with these are called feature space. Also we can develop new features or higher level features from the already available features and adding it to the feature vector. This is called feature construction.

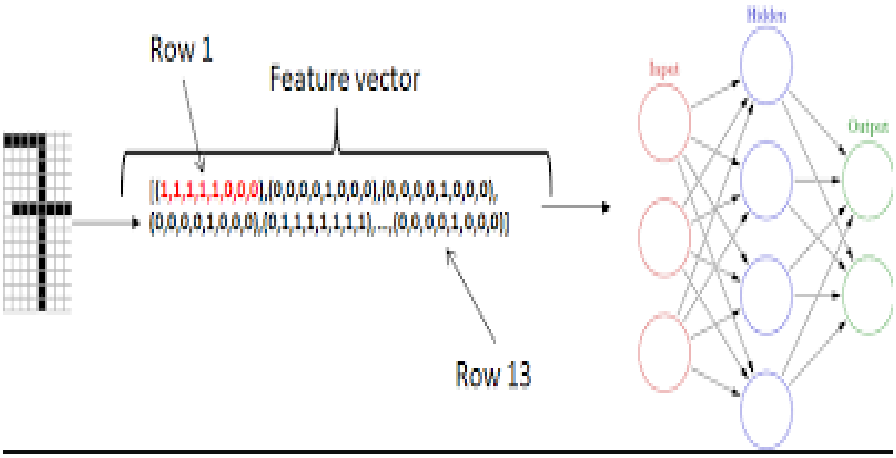

Figure 3.10: Feature vector

In many scenarios using all the features of a dataset is not an ideal approach as some features do not contribute in increasing the efficiency of a program and also the data size would become very large. The three main goals of feature selection are to

- Reduce the cost of computation
- To improve the overall accuracy of the model
- Producing a more understandable model

Feature selection can be performed both before or after training ,it can be performed manually or by automated methods. Below are some methods.

Correlation plot: It is one of the manual techniques to perform feature selection. This creates a visualization that plots the correlation measure for every feature in the data. We can then observe which features are closely related and therefore can remove some of those , and also some variables might have a low correlation with the output variable , so it is recommended to remove some of those.

Below mentioned two techniques are also used by us in development of our model.

Feature importance: It is a technique that gives a score value to input features based on the fact that how useful they are in predicting a target variable. This plays an important role in selecting the best features for our model and increasing the efficiency as it acts as a good indicator of the features that have an impact on the model and we can remove those that do not have a significant one.

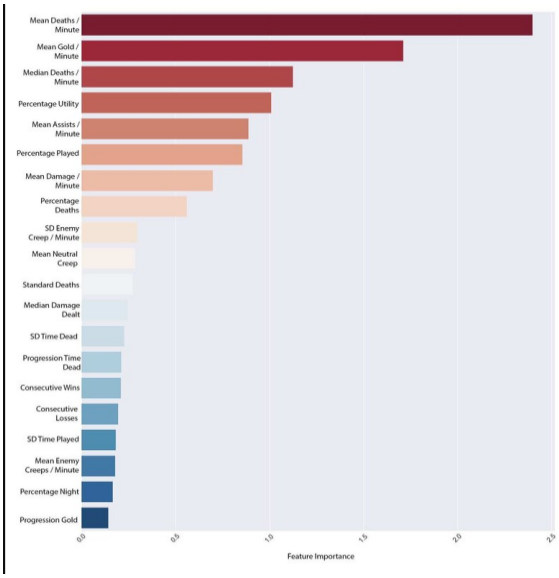

Figure 3.11: Feature importance

Heat maps: Heat maps are good technique to find the correlation between the dependent and the independent variables. These are a form of matrix plots. We can gain useful insights from this method.

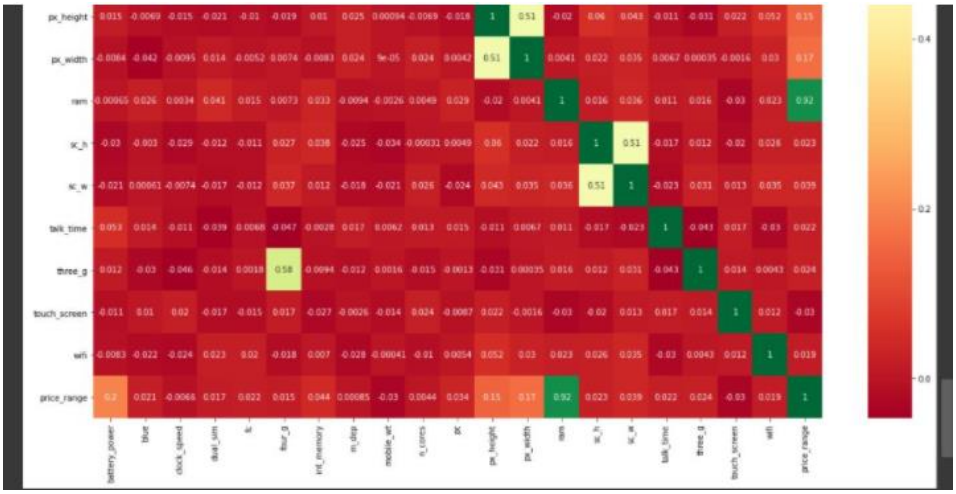

Figure 3.12: Heat map

3.1.4 Random Forest Algorithm

Before studying random forests we should know about the building blocks and that is decision trees. Decision tree forms a tree like structure and supports decision. In simple language decision trees are an advanced version of if-else statement. A decision tree consists of three components –

Root nodes- Nodes present at start of tree.

Decision nodes- Nodes after splitting.

Leaf Nodes – Nodes where no further split happens.

A decision tree simply takes the training data and then splits it till we reach the point of leaf nodes. The leaf node cannot be further splitted. The nodes of a tree are the features that are used for predicting the results of the model. Decision trees are often used in performance research and project management. If, in fact, decisions have to be made online without having to reconsider under incomplete information, the decision tree should be compared to the opportunity model as the best selection model or the online selection algorithm. Making a decision tree involves the process where we have to choose the features to act on and the condition on which a split would occur.

Splitting cost: It differs in problems of regression and classification. But the common thing in both the cases is that the cost functions always try to find branches of the tree with groups that have similar responses. Below are given the general formula used for splitting.

$$\text{Regression} : \text{sum}(y - \text{prediction})^2$$

$$\text{Classification} : G = \text{sum}(pk * (1 - pk))$$

Now the next question arises is that how do we know when to stop splitting in a decision tree. When a dataset contains large number of features , it would result in larger number of splits and can lead to overfitting. So there are many ways to stop this from happening by hyperparameter tuning, example setting the max depth of the tree , or setting the value after which a split would occur.

Entropy and information gain are the two main bricks that build a decision tree.

Entropy: In simple terms it refers to the measure of noise or randomness in the dataset. In a decision tree mostly the output is either yes or no. So there has to be a clear distinction. Example if a certain decision gets 5 upvotes and 4 downvotes thsis would to lead to disorderness as there are almost equal number of votes for both the ways it becomes difficult to decide. It measures the impurity of a node i.e, the randomness. A pure split occurs if we get all the answers of a split from the positive class or from the negative class. When the purity of a leaf node reaches 100 percent we make that node as leaf node. Higher the entropy, higher will be the impurity. By means of entropy we can only know about the impurity of a particular node and have no idea abouth the entropy change in the parent or any particular node. So information gain steps in here. The formula for entropy is

$$E(S) = -p_{(+)} \log p_{(+)} - p_{(-)} \log p_{(-)}$$

where p_+ refers to probability of positive class

p_- refers to probability of negative class

S is the subset of the training data.

Information Gain: Plays an important part to decide whether a node is to be used as a decision node or the root node. It also measures the reduction in randomness according to a given feature. It can be called the entropy for a full dataset. The formula for information gain is

Information Gain= E(Y)- E(Y|X)

Now we proceed with the random forest algorithm. As obvious from the name random forests consists of large number of singular trees that operate as a single unit . Each tree in the forest makes a class prediction and the class which gathers most of the votes becomes the final prediction.

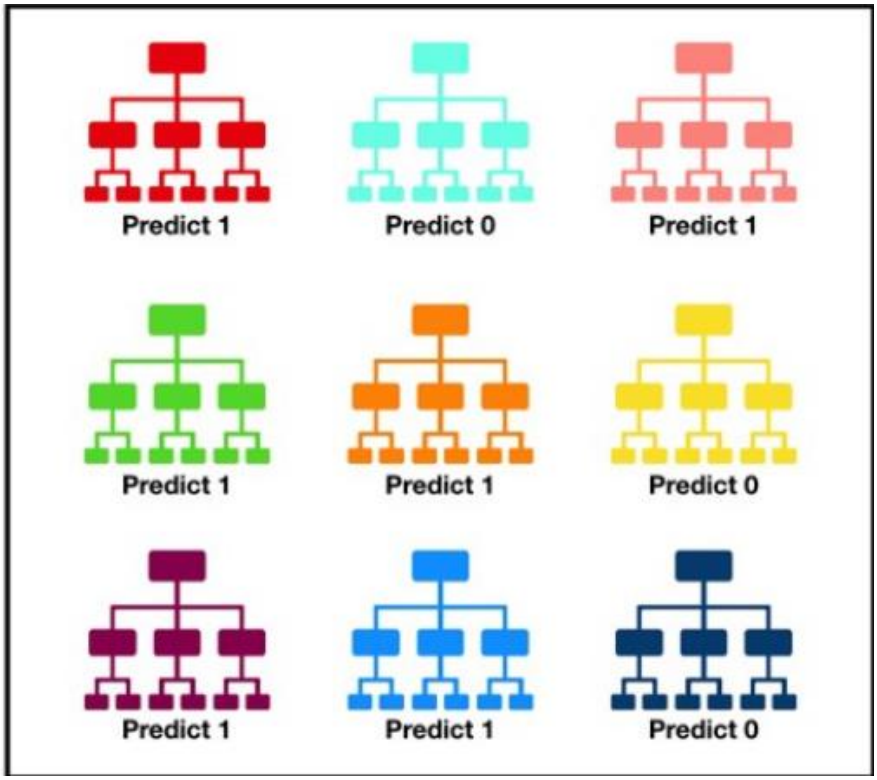

Figure 3.13: Simple illustration of random forest

Also we have to keep in mind the basic fundamental behind this is that tress with low correlation are to be used. A greater number of least correlated trees operating as a single unit would result in high efficiency. The simple reason behind this is that the trees would be safe from each others individual errors.

Random forest works on the ensemble technique which simply means combining multiple models. Ensemble uses two types of methodologies-

- Bagging: Also known as bootstrap aggregation. Decision trees are very sensitive on the training data, i.e even the small changes to it can have a big impact on the tree

structures. It makes a new training subset from sample training data with the help of replacement and the final prediction is based on majority voting. Therefore each model is generated from the Bootstrap samples provided by the data via replacement and this is called row sampling. Now this is called bootstrap, i.e the process of row sampling with replacement. Example if we have a training data with n features and goes something like this [5,6,7,8,9], we can give one of the trees the following set [5,5,6,7,7] i.e values from the same set but provided randomly. Now we have separate models generating their own results and the final outcome is based on the maximum voting. The step where all the results are combined and the output is generated is called aggregation.

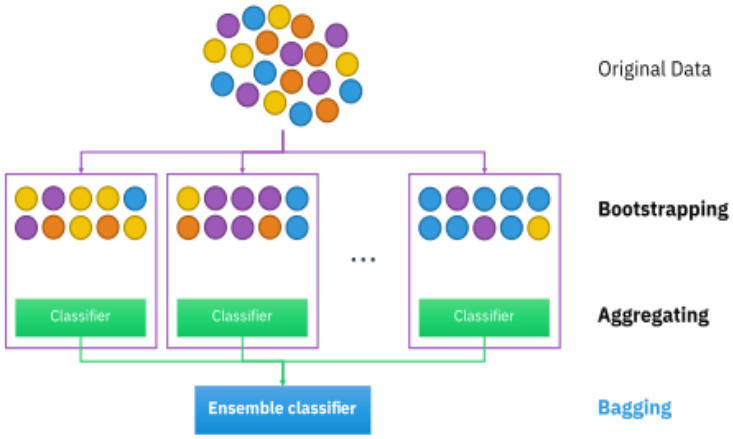

Figure 3.14: Bagging process

- Boosting: It simply combines the weak trainers into strong trainers by creating sequential models such the final model results is higher efficiency. We are not going to elaborate on this as this technique does not find its use in random forest.

Feature randomness- When we talk about a normal decision tree, the time when we have to do a split at the node, we have to consider all the features but ultimately have to pick one that would give us the most amount of separation between the right node and the left node. But in a random forest , it only has random subset of features to its disposal which forces even more separation among the individual trees and leads to less correlation among them and therefore more efficiency.

Steps involved in the random forest algorithm:

i. First we have to take k number of random records from a set that has a total of n records.
ii. Now for each sample data a decision tree is made.
iii. In the next step, the individual trees would generate an output.
iv. The final output is based on average in the case of regression and and majority votes for the case of classification.

So ultimately in random forest we are left with trees that are trained on various sets of data (bagging) but also use completely random features to make decisions.

Important characteristic of random forests-

- Variety- not all attributes/variables/capabilities are considered even as making an character tree, every tree is distinct.
- Proof against the curse of dimensionality- since every tree does now not bear in mind all the features, the characteristic area is reduced.
- Parallelization-each tree is created independently out of various statistics and attributes. this means that we will make full use of the CPU to build random forests.
- The brake up in train and test data- In a random forest we don't should segregate the facts for train and test set data as there'll usually be 30% of the information which is not visible by using the selection tree.
- Stability -balance arises because the end result is based totally on majority vote casting/ averaging.

So now lets us try to draw a comparison between the two and see as to why random forest just outperforms the decision tree and why we chose it to employ in our project. A decision tree might be simple to understand and employ but sometimes it can lead to over complex trees that do not come to good use. A decision tree many times suffer from the problem of overfitting as the user sometimes does not controls it, but in the case of random forest the overfitting is handled quite well as model is trained on subsets of data and final result comes from average or voting. A random forest suffers via slow computational power as it works on many trees simultaneously and hence the decision tree comes up first in this aspect. Also they go through higher training periods as compared to decision trees. In random forest, the data set is not set up on any rules or formuals , i.e random features are

selected on their own, but in the case of a decision tree some predefined rules are set to do the prediction. The main disadvantage of a decision tree is that the slightest of change in the data can cause disruption to the tree structure.Decision tree novices create biased trees if some instructions dominate.
Itis consequently advocated to balance the information set prior to fitting with the decision tree.In an ideal world we would want to reduce the variance and bias based error and random forests are just perfect for that. They offer more resilience than decision trees. Based on all these factors we can say that random forest is one of the best and highly efficient techniques and why we chose to go along with it in our model too.

3.1.5 HTML and CSS

HTML is short for Hyper Text Markup Language and is used to create webpages and webapps. Tim Berners-Lee is called father of HTML. In 1991 Tim , proposed the first available description of HTML called HTML tags. Now let us divide into sections.

Hyper Text: This in simple words means text within text. A link is hiddenly stored within the text. Whenever someone clicks on a link that takes you to a new webpage, this means you have interacted with a hypertext. It is way to link one webpage to another.

Markup language: It can be defined as a language used to apply layout and formatting to text document. It makes the text more visually appealing and interactive. Text can be converted to links , tables and images etc.

Webpage: It is usually written in HTML and then a web browser translates it. URL is used to identify a web page. Webpages are of two types static or dynamic . But with HTML we can create static web pages.

HTML is made up of different HTML tags and content is different for different tags. Below given is the figure of a simple webpage code written in HTML.

```
<!DOCTYPE>
<html>
<head>
<title>Web page title</title>
</head>
<body>
<h1>Write Your First Heading</h1>
<p>Write Your First Paragraph.</p>
</body>
</html>
```

Figure 3.15: Basic Html Code

<!DOCTYPE> : It simply tells the web browser about the version of HTML.

<html>: It tells the browser that the current document is of type HTML. All things written between html tag describes the web document. It also acts as a container for the other elements except <!DOCTYPE>.

<head>: First tag inside the html tag. Contains metadata. Should be closed before the body tag.

<title>: It is optional. It basically is the name displayed on the tab of the web browser window. Should be placed in the head tag and closed immediately.

<body>: Defines the content visible to the user.

<h1>: Top heading level

<p>: Stands for the paragraph in the webpage.

CSS: Stands for cascading style sheets and allows you to create beautiful looking web pages. With the use of CSS you control how elements of HTML will look in the browser according to your design. CSS can perform basic actions like change of color and size etc, but can also do things like animation. In CSS we have to define rules containing styles and that should be applied to particular elements in the HTML. There are three ways to add CSS to a document.

- Inline CSS: applies style to a single element. Add the <style> attribute to the required element.
- Internal CSS: <style> has to be added inside the <head> tag.

- External CSS: In this the html document should have a link or a reference to the external CSS file. The reference has to be provided inside the <link> tag inside the head section. The external document should be saved with a .css extension. Also it should be free of any HTML tags.

  <link rel=”stylesheet” href=”styles.css”>

Below is a simple example.

```
h1 {
    color: red;
    font-size: 5em;
}
```

It opens with a selector, in this case h1 which tells the doc that we have to style h1 elements. Followed by curly braces and then property-value pairs and ends with the colon. We can only apply values to a property that are allowed by CSS.

3.1.6 Bootstrap

It is a open source and free CSS framework, used to create responsive, mobile friendly websites. It is a framework for front end development. It is much easier and faster than other available platforms. It comprises of CSS and HTML designed templates to do various stuff like creating forms, tables , carousels, images , text etc. It also supports js plug ins. It was made at twitter and offered to help a solution to the inconsistencies between the different frameworks used for front end development.

Bootstrap styles the html elements and can make the webpage uniform. It has predefined components like tables , forms, cards etc which can be used directly in the project and customized with the help of CSS classes, according to the developer or user needs.

The most awesome feature of Bootstrap is the layout management or the grid system that simply brings uniformity to the webpage. The basic is called the ‘container’ and every other element is placed inside it. Containers can be fluid or fixed. A fluid one is flexible or responsive and fills the entire width of the webpage. But the fixed one uses fixed widths

depending on the screen size as defined by bootstrap. Inside the container bootstrap implements a row and column structure called CSS Flexbox.

3.1.7 Flask Framework

It is a python written micro web framework. It does not essentially require the particular tools or technologies and is therefore called micro framework. It does not have any form validation or a data base abstraction layer where some useful functions could be provided by the existing third party libraries. But with this flask has its support for extensions that add app features and looks like if they were written and implemented in it. Extensions exist for form validation, mappers and various related tools.

It has various components:

Jinja: It is an engine for templates for python. It handles templates.

MarkupSafe: It is the string library for the python language. It marks the strings as safe and when used with regular strings automatically helps in escaping unmarked strings.

ItsDangerous: It is the data serialization library. Its main use is to store the session of a Flask application inside of a cookie and prohibits users to tamper with the contents of the session.

Werkzeug: It is designed for python and is a utility library for Web Server Gateway Interface apps.

SAMPLE CODE:

```
From flask import Flask

app1 = Flask(__name__)

@app1.route("/")

def hello():
        return "<p>Hello</p>"
```

We started by importing the Flask class. Our WSGI application will be an instance of this class. After that, we make an instance of the class. The name of the application's module or package is the first argument. __name__ is a handy shortcut for this that works well in most cases. Flask needs this to know where to look for resources like static files and templates.

The route() decorator is then used to notify Flask which URL should be used for calling our function. The message we want to show in the user's browser is returned by the function. Because HTML is the default content type, the browser will render HTML in the string.

Routing: To assist consumers, modern web apps employ sensible URLs. Users are more likely to like and return to a page if the URL is memorable and can be used to get directly to the page. Have to use the route decorator to bind any function to the URL. If we write route (" / ") this simply binds with the home page. But if we use something like this route("/hello") then the webpage is bounded with the hello URL.

The variable name> tag can be used to add variables to a URL. The variable name> keyword argument is subsequently passed to your function. Simple example such as app1.route('/user/<username>') show that the variable username will be passed to the function whenever the URL is called. We can also specify the type of argument for the function with the help of a converter, like this <converter:variable_names>. The various converters accepted by the route method are given below in the table.

Table 3.1: Types of converter variables

string	Accepts any text without a slash
path	Same as string but also accepts slashes
uuid	Accepts UUID strings
int	Accepts positive integers
float	Accepts positive floating point values

HTTP methods: When visiting URLs, web applications use various HTTP methods. As you work with Flask, you should become familiar with the HTTP methods. A route only responds to GET queries by default. The route() decorator's methods argument can be used to handle various HTTP methods. If GET method is in use flask automatically supports the OPTIONS variable to be used by a HTML form. The POST method is basically a tool that sends the data collected in a form to the server or the URL. Basically it means that the data we collect in a form is sent to the required URL for the required function when we mention the POST method in the route decorator.

Static Files: If you want to create a dynamic web application then you would need static files. That's usually where we find use of the CSS and javascript. When we create a folder named static in the project it will be available as /static in the application. For generating URL's for static files use the following code.

url_for('static' , filename='style.css') . The file has to be named as static/style.css .

Rendering templates: It's neither enjoyable or easy to generate HTML from Python because you have to do all of the HTML escaping yourself to keep the application secure. As a result, Flask automatically configures the Jinja2 template engine for you.

The render template() method can be used to render a template. All you have to do is supply the template engine the name of the template and the variables you want to pass as keyword arguments.

def hello(name=None)

return render_template('hello.html', name=name).

Flask would look into the /templates folder for the required templates used in the program.

Pickle Module: For serialising and de-serializing a Python object structure, the Python pickle package is utilised. Pickling an object in Python allows it to be saved on disc. Pickle "serialises" the object before writing it to the file. Pickling is a Python function that converts a dict, list or other Python object into a stream of characters. The concept is that this character stream provides all of the data required to recreate the object in another Python function. The pickle data format is specific to Python This bears the advantage that the external standards such as XDP and JSON impose no restrictions. Usually pickle uses a compact binary representation. In our case we had to serialize our model which was in the form of a jupyter notebook or ipynb file format. So we have to use the pickle.dump() function to load or serialize the model in use. When deserializing the model or when we have to use the model in our code again the pickle.load() function comes to use. Basically this function reads the pickle representation of the object and returns the reconstituted object.

3.1.8 Heroku

It is a cloud based platform that has gained in popularity in recent years. Heroku is a popular solution for many projects since it is so simple to utilise. It offers straightforward application creation and deployment, with a special focus on facilitating customer-focused apps.

Businesses that use the Heroku platform may focus on refining their apps because the platform takes care of the servers and platforms. Not to mention the infrastructure that keeps them running. More time is dedicated to ensure that users have the best possible experiences.

3.2 Model Development

3.2.1 Dataset

The dataset was collected from Kaggle.com. Kaggle is a google owned subsidiary, web community of information scientist and ML practitioners. Kaggle lets customers to find and upload record units, discover and construct models via internet based record science surroundings, engaging with other data scientists and also participate in competitions to solve any data science problems. We are assured that the dataset is original and authentic because kaggle is a trusted website that is used by over a million users worldwide.

Coming to the dataset we used. The name of the dataset is “Flight fare prediction dataset by machine hack”. The dataset consists of a total entries of 10683. The total number of columns or features in this are 11. Also when we take a closer look we find that out of the 11 variables 10 are categorical type of features and only 1 is of the numerical type. The numerical variable is the price. As we have to predict the various prices so we will take the price as the dependent variable and other 10 could be used as independent variables. Below mentioned table shows the features along with their datatype and count.

Table 3.1 Dataset Information

#	Feature	Non Null Count	Data type
0	Airline	10683	Object
1	Date_of_Journey	10683	Object
2	Source	10683	Object
3	Destination	10683	Object
4	Route	10682	Object
5	Dep_Time	10683	Object

6	Arrival_Time	10683	Object
7	Duration	10683	Object
8	Total_Stops	10682	Object
9	Additional_Info	10683	Object
10	Price	10683	Int64

The “Airline” field contains all the names of the various flights. Some of these are also repeated but come out as unique due to the different date of journeys etc. The “source” and “destination” field as expected contains the starting point and the ending point for the plane. The “dep_time” and the “arrival_time” contains the time stamp at which the plane leaves the airport and lands at the desired airport. The “duration” field contains the time for which the plane travels. The “total_stops” field includes the number of stops a flight takes between the starting point and the ending point.

Also it is clear from the above stats that there are 2 Null values in the dataset. Those null values are included in the column “Route” and the column “Total_Stops”. As it is our responsibility to deal with those they were duly handled with.

Also during analysis we found that “Date_of_Journey” was of object data type so had to be converted to timestamp so that it can be used correctly for prediction. Similarly for the ease of use we extracted information from “Dep_Time” , “Arrival_Time”, and “Duration” columns. Further doing the EDA we handled all the categorical data with the help of encoders i.e OnehotEncoder for all the nominal data and LabelEncoder for the ordinal data. The features that were handled here were “Airline”, “Source”, “Destination”, “Route”, “Total_Stops”.

Below mentioned is the general information about the price feature of the dataset.

	Price
count	10683.000000
mean	9087.064121
std	4611.359167
min	1759.000000
25%	5277.000000
50%	8372.000000
75%	12373.000000
max	79512.000000

Figure 3.16: Description of target variable

In this dataset the test set was given alongside the training dataset. It was a subset of the same dataset. The test data set had a total of 2671 entries and 10 columns. The price was columns was not there because as understood we have to predict that. The test set did not have any NULL values. Again with the test set we performed all the work that we did with the training data so that it was inline with the data we were training on.

3.2.2 Computational

For this project work we used the machine with the following specs at the time of training.

CPU: The computer we used had the following specs:

Table 3.2 CPU specifications

Parameter	Specifications
CPU Model name	Intel(R)Core(TM)
CPU frequency	2.3 Ghz
No of CPU cores	4
Available Ram	7.86 GB
Disk Space	400 GB

These are the CPU specs of the machine we used to do computations. Most of the computational work mostly happens on the GPU. But CPU takes care of most of the preprocessing. The large amount of RAM did not put loads of pressure and made it easier for the whole dataset to be loaded in time and we did not have to worry about any system

crashes occurring. The clock speed of the CPU mentioned 2.3 Ghz is the baseic clock speed which if needed can go upto 5 Ghz. But no overclocking was needed as the system was able to do the work in its normal 4 cores.

Table 3.3: GPU Specifications

Parameter	Specifications
GPU	NVIDIA GeForce GTX 1060
GPU Memory	10 GB
GPU Memory Clock	1.40 Ghz
GPU Release Year	2016
Cores	2
Available RAM	6 GB
Disk Space	400 GB

Tools Used: We used the following tools in making our model.

- Python
- Matplotlib
- Seaborn
- Jupyter Notebook
- Pandas
- Plotly
- Numpy
- Scikit Learn
- Bootstrap
- HTML

- CSS
- Flask
- Heroku

These packages mentioned above were used in their latest upto date editions. The code works properly and would not cause any issue until any further updates in them.

3.2.3 Experimental

Now we tried a few algorithms just to check them in comparison to what we used.

i. Linear regression
ii. Decision trees
iii. KNN

We are also providing with some basic definitions of the above algorithms so as to explain easily and look why those algorithms did not perform the way desired.

i. Linear regression:

It is one of the well-known, understood and most used algorithm in ML and statistical learning.

It is based on the concept of supervised learning. It performs the work to predict a target variable (y) based on the independent variable (x), hence finds a linear relationship between the two. The hypothesis function for linear regression is:

$$y = \theta_1 + \theta_2 x$$

Where Θ_1 is the intercept and Θ_2 is the coefficient of x.

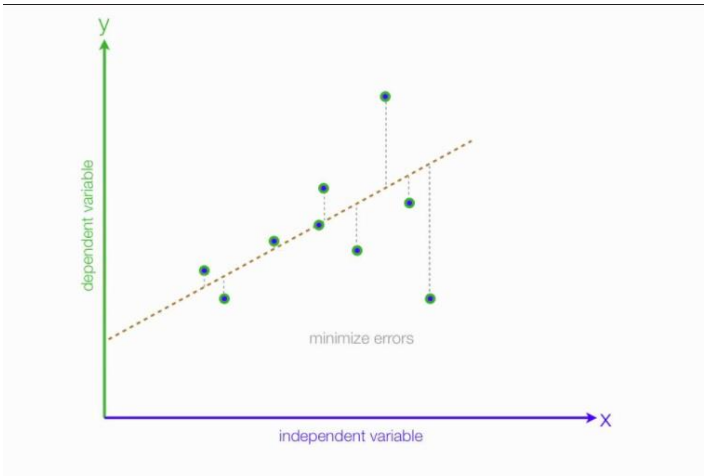

Figure 3.16: Linear Regression

Once we find the best theta values we draw the best fit line and we would check for our prediction. Our role is to minimize the cost function(J) which is given below

$$J = 1\frac{1}{n}\sum_{i=1}^{n}(pred_i - y_i)^2$$

$$Accuracy = 61\%$$

ii. Decision trees:

This is an algorithm we already talked about . It uses a tree approach to get to the results. In this algorithm we calculate entropy of every feature and the one with the least is selected as the root node and we keep on splitting for the remaining columns. The reason behind the failure of this algorithm is that he model works on data in a sequential manner and the feature that has more domination outlasted the other feature.

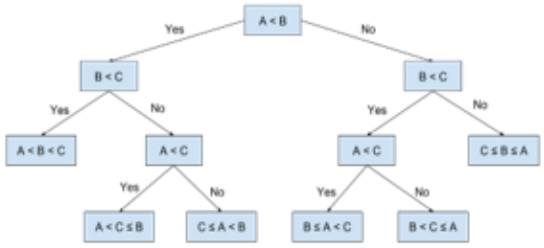

Figure 3.17: Decision Tree

$$Accuracy = 72\%$$

iii. KNN:

KNN i.e K nearest neighbours is one of the algorithm that uses the information of its neighbourhood to calculate results. This works on the concept that it checks similarity between the new data and the available data and puts this new input into the category which is most like to the other available categories. This algorithm works for both regression and classification based problems. It is a non parametric algorithm , which in simple terms mean that it does not make any assumptions on the

given data. It does not learn from the data quickly but does the action on the dataset at the time of need.

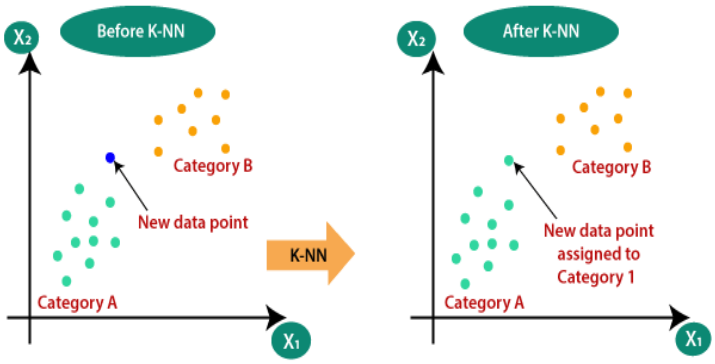

Figure 3.18: Depiction of KNN

$$Accuracy = 68\%$$

3.2.4 Mathematical

Down below we have listed the formulas used by us in the report.

Linear regression: $y = c + mx$

$$J = \frac{1}{n}\sum_{j=1}^{n}(pred_j - y_j)^2$$

Entropy: $E(S) = -p_{(+)} \log p_{(+)} - p_{(-)} \log p_{(-)}$

Information Gain: $E(Y) - E(Y|X)$

3.2.5 Statistical

As we have performed EDA on our dataset therefore we could infer some statistical insights from that. We would be listing some of the graphs not all, that we visualized down below.

Also the below mentioned table contains some stats and information about the mean , min etc of the numerical feature of our dataset.

Table 3.4 Dataset Stats

	Price
Count	10683.0000
Mean	9087.064121
Std	4611.359167
Min	1759.0000
25%	5277.0000
50%	8372.0000
75%	12373.0000
max	79512.0000

As seen we can infer that the mean price for a flight ticket is around 9000 INR and the prices can go as low as 1800 INR. So you can see the application of our model as it would be of great use to an average customer as he/she can afford the ticket at a lower price than paying higher amounts.

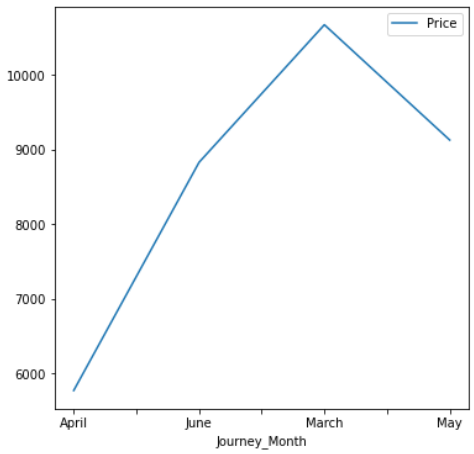

Figure 3.19: Journey_Month vs Price

We can see from the above graph that the fare price is fairly high in the month of March. This could be due to the prior booking that people do for flights which means like for

travelling in may or june bookings can be done in march which leads to higher demands and therefore higher prices.

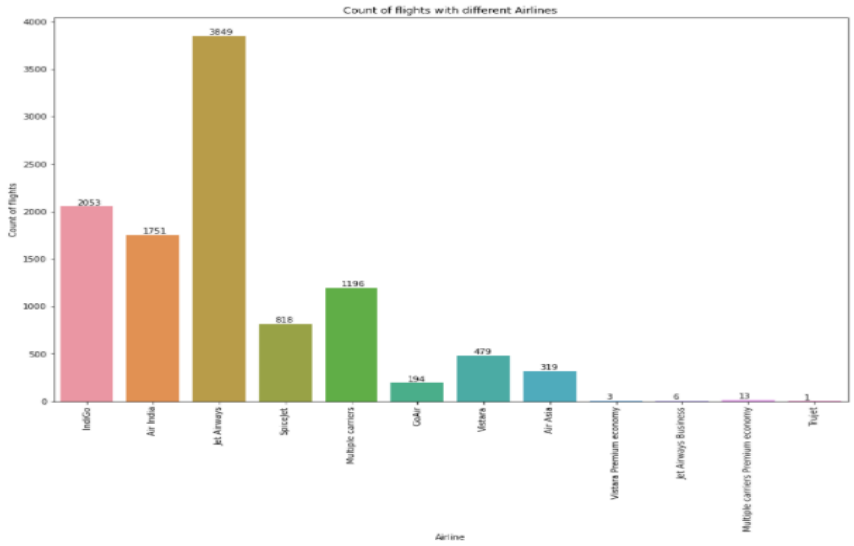

Figure 3.20: Airline vs price

Here is another example where we can see that “jet airways” have the price at the highest and other airways have a price almost at the same price bracket. These are a few examples of the data visualization that can help in understanding the data far better.

Chapter 4 PERFORMANCE ANALYSIS

4.1 Analysis of System

Analysis of the system is the thing where we compare our system with the previous implementations and check the performance of the system.In today's world flight companies try to change flight prices according to their needs and present condition. In this way they can maximize their profits. Regression based solutions to this problem are common but they come with their own disadvantage. With the technological advancements and the load increase on these flight companies , regression does not give the desired results and offers low accuracy. Customers on the other hand do not have a tool to have an idea about the ticket prices that could help them in planning accordingly. Also many customers don't know the right time to book a ticket and sometimes have to pay more for a ticket. Consider an example, the least expensive ticket will change its value over a period and its value may be high or low depending on the time like summers ,winters etc or according to day ,night or evening. This happens because the underlying goal of the plane carrier is to maximize its income. But on the other side the purchaser try to get a ticket at a lower price so book them days before the actual takeoff. But sometimes the price of a particular ticket does not fluctuate and he consumer ends up paying more for the same. Also while working on the model the accuracy score of our system on the training set was about 0.953 and on the test set was 0.805 which indicates that our model was trained good and overfitting did not occur.

Given below are the results of different models used on this task before which includes the models used by us.

Table 4.1: Models and precison

Model Name	Accuracy
Linear Regression	0.619
Decision Tree	0.729
K-Nearest Neighbours	0.689
Random Forest (initial)	0.798

Random Forest (after tuning)	0.813

Below are shown some of the output at various stage of this project.

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
0	IndiGo	24/03/2019	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3897
1	Air India	1/05/2019	Kolkata	Banglore	CCU → IXR → BBI → BLR	05:50	13:15	7h 25m	2 stops	No info	7662
2	Jet Airways	9/06/2019	Delhi	Cochin	DEL → LKO → BOM → COK	09:25	04:25 10 Jun	19h	2 stops	No info	13882
3	IndiGo	12/05/2019	Kolkata	Banglore	CCU → NAG → BLR	18:05	23:30	5h 25m	1 stop	No info	6218
4	IndiGo	01/03/2019	Banglore	New Delhi	BLR → NAG → DEL	16:50	21:35	4h 45m	1 stop	No info	13302

Figure 4.1 : Original dataset

EDA

	Airline	Source	Destination	Route	Arrival_Time	Duration	Total_Stops	Additional_Info	Price	Journey_day	Journey_month	Dep_hour	Dep_min
0	IndiGo	Banglore	New Delhi	BLR → DEL	01:10 22 Mar	2h 50m	non-stop	No info	3897	24	3	22	20
1	Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	13:15	7h 25m	2 stops	No info	7662	1	5	5	50
2	Jet Airways	Delhi	Cochin	DEL → LKO → BOM → COK	04:25 10 Jun	19h	2 stops	No info	13882	9	6	9	25
3	IndiGo	Kolkata	Banglore	CCU → NAG → BLR	23:30	5h 25m	1 stop	No info	6218	12	5	18	5
4	IndiGo	Banglore	New Delhi	BLR → NAG → DEL	21:35	4h 45m	1 stop	No info	13302	1	3	16	50

Figure 4.2: Departure time converted to single units

	Airline	Source	Destination	Route	Duration	Total_Stops	Additional_Info	Price	Journey_day	Journey_month	Dep_hour	Dep_min	Arrival_hour	Arrival_min
0	IndiGo	Banglore	New Delhi	BLR → DEL	2h 50m	non-stop	No info	3897	24	3	22	20	1	10
1	Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	7h 25m	2 stops	No info	7662	1	5	5	50	13	15
2	Jet Airways	Delhi	Cochin	DEL → LKO → BOM → COK	19h	2 stops	No info	13882	9	6	9	25	4	25
3	IndiGo	Kolkata	Banglore	CCU → NAG → BLR	5h 25m	1 stop	No info	6218	12	5	18	5	23	30
4	IndiGo	Banglore	New Delhi	BLR → NAG → DEL	4h 45m	1 stop	No info	13302	1	3	16	50	21	35

Figure 4.3: Arrival time converted to single units

Same was done for the duration field.

Data Preprocessing and visualization: Here we converted all the categorical data and did some vizualizations.

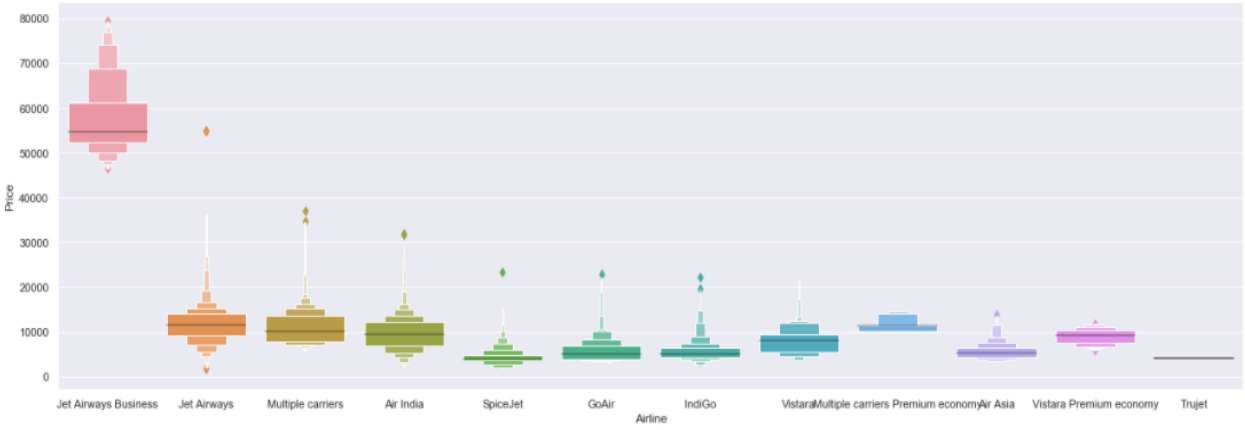

Figure 4.4: Airline vs price

	Airline_Air India	Airline_GoAir	Airline_IndiGo	Airline_Jet Airways	Airline_Jet Airways Business	Airline_Multiple carriers	Airline_Multiple carriers Premium economy	Airline_SpiceJet	Airline_Trujet	Airline_Vistara	Airline_Vistara Premium economy
0	0	0	1	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	1	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0	0	0
4	0	0	1	0	0	0	0	0	0	0	0

Figure 4.5: one hot encoding performed on the airline field

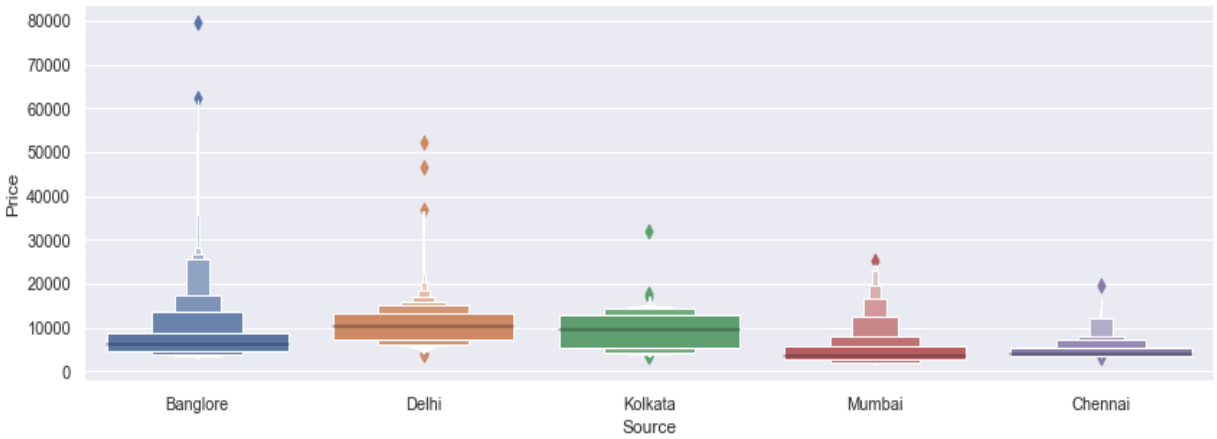

Figure 4.6: source vs price

	Airline	Source	Destination	Total_Stops	Price	Journey_day	Journey_month	Dep_hour	Dep_min	Arrival_hour	Arrival_min	Duration_hours	Duration_mins
0	IndiGo	Banglore	New Delhi	0	3897	24	3	22	20	1	10	2	50
1	Air India	Kolkata	Banglore	2	7662	1	5	5	50	13	15	7	25
2	Jet Airways	Delhi	Cochin	2	13882	9	6	9	25	4	25	19	0
3	IndiGo	Kolkata	Banglore	1	6218	12	5	18	5	23	30	5	25
4	IndiGo	Banglore	New Delhi	1	13302	1	3	16	50	21	35	4	45

Figure 4.7: Label encoding performed on the “total_stops” field

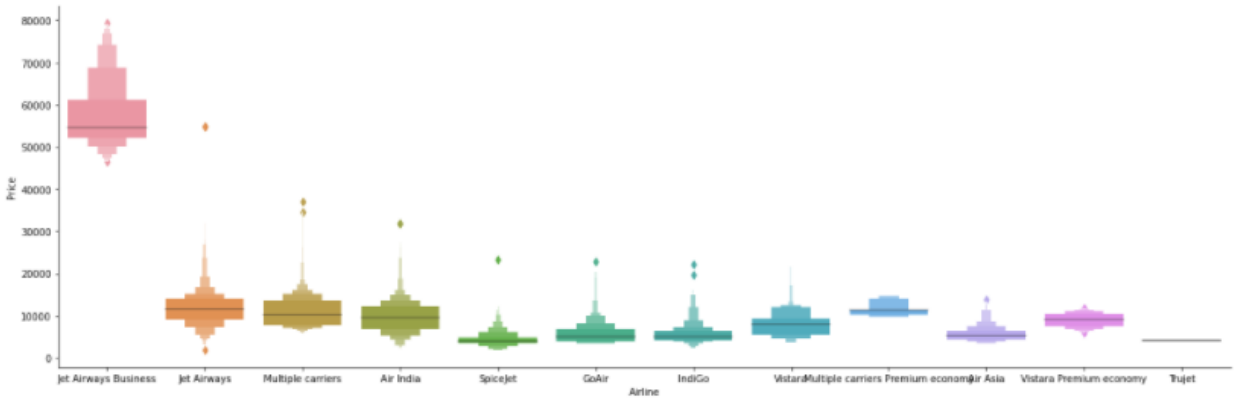

Figure 4.8: airline vs average price

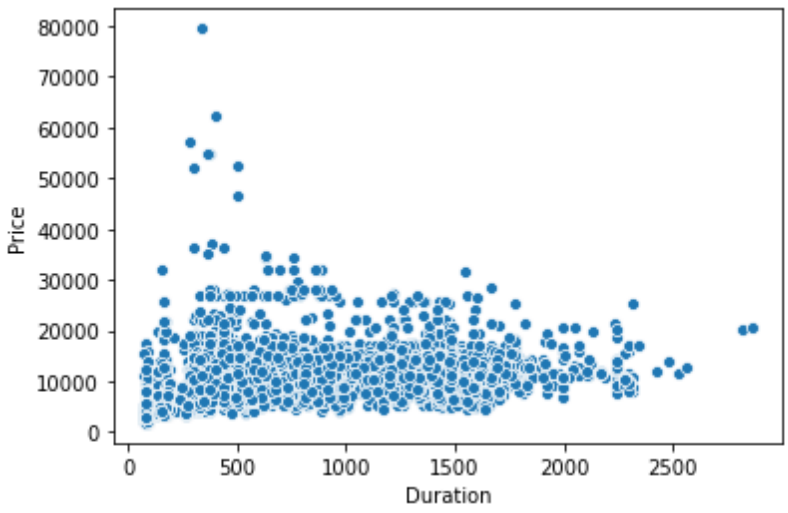

Figure 4.9: Scatter plot of duration vs average price

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info
0	Jet Airways	6/06/2019	Delhi	Cochin	DEL → BOM → COK	17:30	04:25 07 Jun	10h 55m	1 stop	No info
1	IndiGo	12/05/2019	Kolkata	Banglore	CCU → MAA → BLR	06:20	10:20	4h	1 stop	No info
2	Jet Airways	21/05/2019	Delhi	Cochin	DEL → BOM → COK	19:15	19:00 22 May	23h 45m	1 stop	In-flight meal not included
3	Multiple carriers	21/05/2019	Delhi	Cochin	DEL → BOM → COK	08:00	21:00	13h	1 stop	No info
4	Air Asia	24/06/2019	Banglore	Delhi	BLR → DEL	23:55	02:45 25 Jun	2h 50m	non-stop	No info

Figure 4.10: The original test set

All the actions that were performed on the training set were done on the test set.

Feature selection: To know which feature will have a better relationship with the target variable

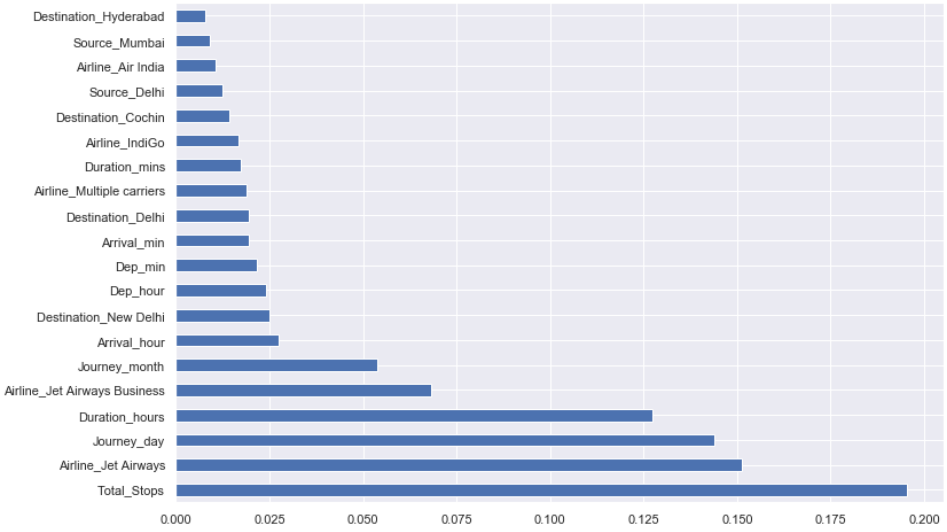

Figure 4.11: Graph of feature importance

Final output:

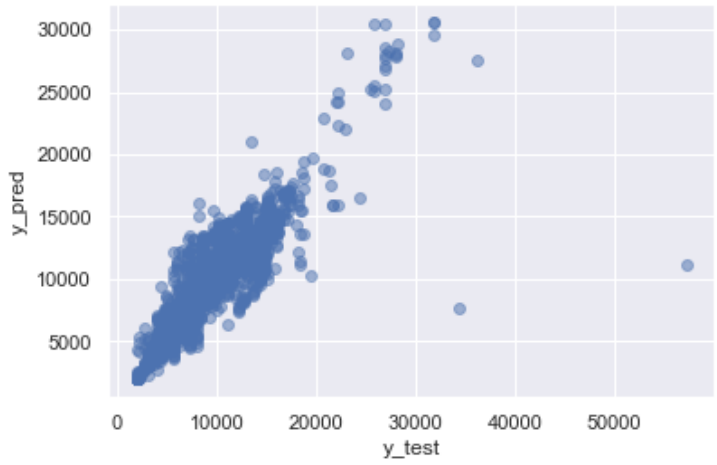

Figure 4.12: Scatter plot between the test and predicted values

```
[43] from sklearn.ensemble import RandomForestRegressor
     reg_rf = RandomForestRegressor()
     reg_rf.fit(X_train, y_train)

     RandomForestRegressor()

[44] y_pred = reg_rf.predict(X_test)

[45] reg_rf.score(X_train, y_train)

     0.9538668924496992

[46] reg_rf.score(X_test, y_test)

     0.7969222173448429
```

Figure 4.13: Accuracy before hyperparameter tuning

```
[ ]  metrics.r2_score(y_test, prediction)

     0.8122412234615818
```

Figure 4.14 Accuracy after tuning

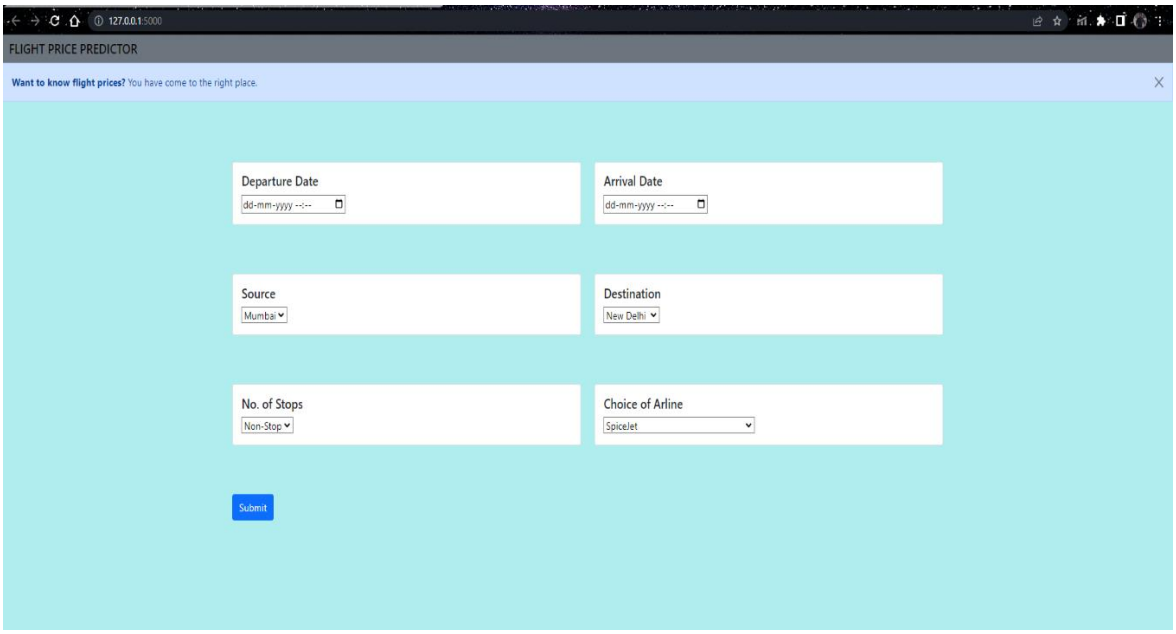

Figure 4.15 Flask App

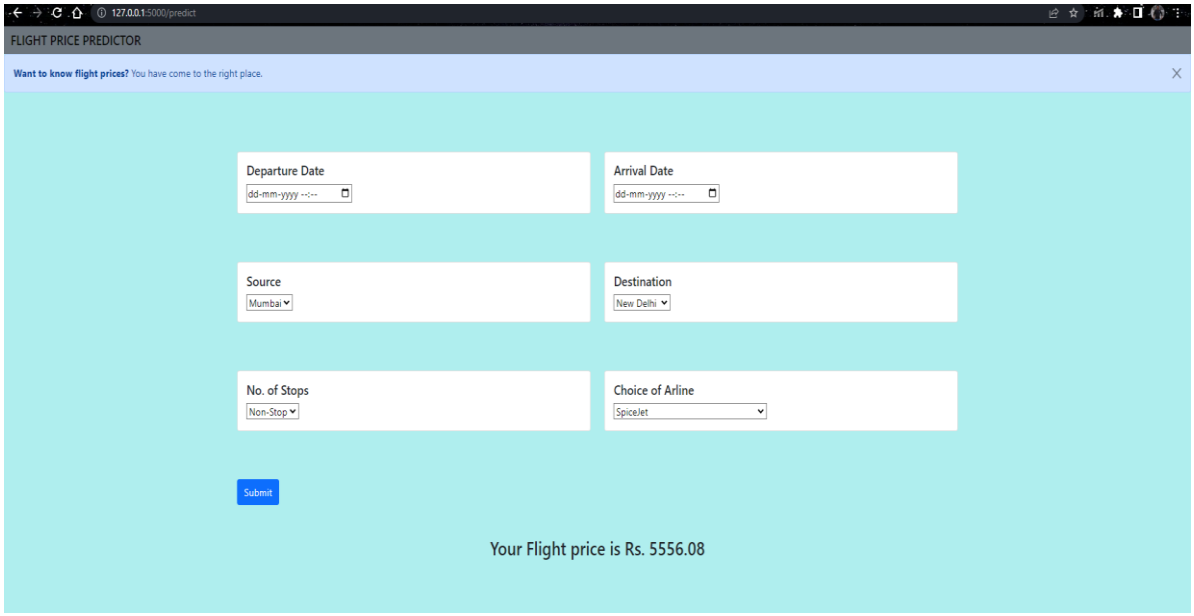

Figure 4.16 Final Output

Chapter 5 CONCLUSIONS

5.1 Conclusions

Predicting air fare prices has been extensively studied and various methods and features have been proposed for it to be performed. We gathered the flight price data from the web and clearly showed that it is very much possible and feasible to predict the prices based on some historical data. This report further shows that, ML predictors(models) are a more than satisfactory option to know the air ticket prices. To the level of our understanding , maximum of the preceding research work on the air plane price prediction concept focused on traditional statistical procedures, which have their own obstacles of estimating and prediction. Also we came to know that proper data and feature extraction and selection are an important part of this process and helped us to come out with some helpful insights. Many features were extracted form the data to make air travel segment easy to understand. We from this project we can get information about the lows and highs of the ticket prices depending on the time of day, the current day or even the weekends. We can also now with full conviction draw a conclusion that if the model implemented in a proper manner can be of great use in saving money of many people by providing them with the information about the air ticket trends and given them an idea about the price which would help them to decide whether to book a ticket right now or in the near future.

Also various ML models are studied and their efficiency and performance are compared so as to get better results. This can also be attributed to the fact that the pricing models used by different airlines are such that to maximize the revenue. Now with help of all this our model can predict the values of the prices with and MSE.

5.2 Future Scope

The future scope can be extended to the fact that it can be used for predicting the prices of the entire plane map of the airway service. More experiments on a larger air fare dataset is an absolute requirement but this initial study depicts the potential of the ML models that can be of use to both the user and the airline service. In the future our project can be prolonged to encompass air ticket transaction stats, that can offer more detail on a selected itinerary , comprising of time and date of arrival/departure, seat area and much more. By aggregating such records with the existing market section and the present day data , it is possible to build an extra powerful and comprehensive price prediction system that can perform on a day by day or even in hourly level. We can also use more better data ,trying with different features to be able to get more efficient results. Additionally we can also look into various advanced ML models, such as Deep learning models , even while running to enhance the present models by hyperparameter tuning to attain a high quality structure.

REFERENCES

[1] T.Wang et al., “A Framework for Airfare Price Prediction: A Machine Learning Approach,” 2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI), 2019.

[2] 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.

[3] S Rajankar, N Sakharkar, “A Survey on Flight Pricing Prediction using Machine Learning”, International Journal of Engineering, 2019.

[4] Vinod Kimbhuane, Harshil Donga, Asutosh Trivedi, Sonam Mahajan and Viraj Mahajan, “Flight Fare Prediction System”, May 2021

[5] Prashant Kapri, Shubham Patane, Noopur Thanvi, Rashmi Thakur, “Predictive Model for Airlines Flight Delay and Pricing”