

A MINI PROJECT REPORT ON  
**AIRLINE FARE PREDICTION**

A dissertation submitted in partial fulfilment of the  
Requirements for the award of the degree of

**BACHELOR OF TECHNOLOGY**

in

**INFORMATION TECHNOLOGY**

*Submitted by*

**Kotha Sri Lakshmi (18B81A12A3)**

**D V S S Mihir (18B81A12B2)**

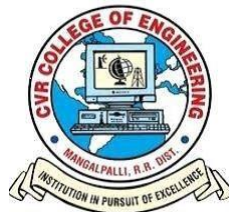
**Chitukula Vaishnavi(18B81A12B3)**

*Under the esteemed guidance of*

**Mrs.K.Revathi**

Assistant Professor, IT Department

CVR College of Engineering



**DEPARTMENT OF INFORMATION TECHNOLOGY**

**CVR COLLEGE OF ENGINEERING**

ACCREDITED BY NBA, AICTE & Affiliated to JNTU-H

Vastunagar, Mangalpally (V), Ibrahimpatnam (M), R.R. District, PIN-501510

2021-2022



Cherabuddi Education Society's  
**CVR COLLEGE OF ENGINEERING**

(An Autonomous Institution)

ACCREDITED BY NATIONAL BOARD OF ACCREDITATION, AICTE

(Approved by AICTE & Govt. of Telangana and Affiliated to JNT University)

Vastunagar, Mangalpalli (V), Ibrahimpatan (M), R.R. District, PIN - 501 510

Web : <http://cvr.ac.in>, email : [info@cvr.ac.in](mailto:info@cvr.ac.in)

Ph : 08414 - 252222, 252369, Office Telefax : 252396, Principal : 252396 (O)

---

**DEPARTMENT OF INFORMATION TECHNOLOGY**

**CERTIFICATE**

This is to certify that the Project Report entitled “**Airline-fare Prediction**” is a Bonafide work done and submitted by **Kotha Sri Lakshmi (18B81A12A3)** , **D V S S Mihir (18B81A12B2)**, **Chitukula Vaishnavi (18B81A12B3)** during the academic year 2021-2022, in partial fulfilment of requirement for the award of Bachelor of Technology degree in Information Technology from Jawaharlal Nehru Technological University Hyderabad, is a Bonafide record of work carried out by them under my guidance and supervision.

Certified further that to my best of the knowledge, the work in this dissertation has not been submitted to any other institution for the award of any degree or diploma.

**INTERNAL GUIDE**

**Mrs.K.Revathi**

Assistant Professor, IT Department

**HEAD OF THE DEPARTMENT**

**Dr. Bipin Bihari Jayasingh**

Professor, IT Department

**PROFESSOR INCHARGE**

**Dr. R. Seetharamaiah**

Professor, IT Department

**PROJECT COORDINATOR**

**G. Sunitha Rekha**

Assistant Professor, IT Department

**EXTERNAL EXAMINER**

---

City Office : # 201 & 202, Ashoka Scintilla, Opp. KFC, Himayatnagar, Hyderabad - 500 029, Telangana.

Phone : 040 - 42204001, 42204002, 9391000791, 9177887273

## **DECLARATION**

We hereby declare that the project report entitled “**Airline-fare prediction**” is an original work done and submitted to IT Department, CVR College of Engineering, affiliated to Jawaharlal Nehru Technological University Hyderabad, Hyderabad in partial fulfilment of the requirement for the award of Bachelor of Technology in **Information Technology** and it is a record of Bonafide project work carried out by us under the guidance of **Mrs.K.Revathi, Assistant Professor, Department of Information Technology.**

We further declare that the work reported in this project has not been submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other Institute or University.

**KOTHA SRI LAKSHMI**

**(18B81A12A3)**

**D V S S MIHIR**

**(18B81A12B2)**

**CHITUKULA VAISHNAVI**

**(18B81A12B3)**

## ACKNOWLEDGEMENT

The satisfaction of completing this project would be incomplete without mentioning our gratitude towards all the people who have supported us. Constant guidance and encouragement have been instrumental in the completion of this project.

First and foremost, we thank the Chairman, Principal, Vice Principal for availing infrastructural facilities to complete the mini project in time.

We offer our sincere gratitude to our internal guide **Mrs.K.Revathi**, Assistant Professor, IT Department, CVR College of Engineering for his immense support, timely co-operation and valuable advice throughout the course of our project work.

We would like to thank the Professor In-Charge of Projects, **Dr. R. Seetharamaiah**, Professor, Information Technology for his valuable suggestions in implementing the project.

We would like to thank the Head of Department, Professor **Dr. Bipin Bihari Jayasingh**, for his meticulous care and cooperation throughout the project work.

We are thankful to **G. Sunitha Rekha**, **Project** Coordinator, Assistant Professor, IT Department, CVR College of Engineering for his supportive guidelines and for having provided the necessary help for carrying forward this project without any obstacles and hindrances.

We also thank the **Project Review Committee Members** for their valuable suggestions.

## ABSTRACT

As domestic air travel is getting more and more popular these days in India with various air ticket booking channels coming up online, travelers are trying to understand how these airline companies make decisions regarding ticket prices over time.

Nowadays, airline corporations are using complex strategies and methods to assign airfare prices in a dynamic fashion. Due to the complexity of the pricing methods applied by the airlines, it is very difficult for a customer to purchase an air ticket at the lowest price, since the price changes dynamically.

To estimate the minimum airfare, data for a specific air route has been collected including the features like departure time, arrival time and airways over a specific period.

Features are extracted from the collected data to apply Machine Learning (ML) models. Remarkably, the trends of the prices are highly sensitive to the route, month of departure, day of departure, time of departure, whether the day of departure is a holiday and airline carrier. The data also validated the fact that, there are certain time-periods of the day where the prices are expected to be maximum.

The objectives of the project can broadly be laid down by the following questions:

**1. Flight Trends** - Do airfares change frequently?

**2. Best time to buy** -What is the best time to buy so that the consumer can save the most by taking the least risk?

**3. Verifying Myths** - Does price increase as we get near to departure date? Is Indigo cheaper than Jet Airways? Are morning flights expensive?

## LIST OF FIGURES

S.NO	Title	Pg.No
1.	Dynamic Pricing	15
2.	Architecture Diagram	17
3.	Use Case Diagram	18
4.	Data Sets	19
5.	Training Data Set	20
6.	Testing Data Set	21
7.	Importing Data Sets	24
8.	OneHotEncoding	25
9.	Label Encoding	26
10.	Visualization	27
11.	Random Forest	28
12.	HyperParameterTuning	29
13.	Sample Test Data	30
14.	Heat Map	31
15.	Metric Evaluation	32
16.	Python Version Selection	38
17.	Download executable Installer	39
18.	Run Executable Installer	39
19.	Verify Python was Installed	40
20.	Verify Pip was Installed	40
21.	Jupyter Installation	41

## TABLE OF CONTENTS

<b>S. No</b>	<b>Topic</b>	<b>Pg. No</b>
<b>1.</b>	<b>Introduction .....</b>	<b>8</b>
<b>2</b>	<b>Software Requirement Specifications.....</b>	<b>16</b>
<b>3</b>	<b>Design.....</b>	<b>17</b>
<b>4</b>	<b>Implementation.....</b>	<b>24</b>
<b>5</b>	<b>Testing.....</b>	<b>30</b>
	<b>Conclusion.....</b>	<b>34</b>
	<b>Future Enhancements.....</b>	<b>35</b>
	<b>References.....</b>	<b>36</b>
	<b>Appendix A - Abbreviations.....</b>	<b>37</b>
	<b>Appendix B - Software Installation Procedure.....</b>	<b>38</b>
	<b>Appendix C – Software Usage Process.....</b>	<b>42</b>

# CHAPTER 1

## INTRODUCTION

### 1.1. Literature Survey

#### 1.1.1. Customer side models

The studies performed on the customer side can be roughly categorized into two: those that try to predict optimal ticket purchase timing and those that proposed solution to predict exact value of ticket price.

##### Optimal ticket purchase timing

The authors in (T. Wohlfarth et al., 2011) proposed an optimal ticket purchase time optimizing model based on a special preprocessing step known as marked point processes (MPP), data mining techniques (clustering and classification) and statistical analysis techniques.

The MPP pre-processing technique was suggested to convert heterogeneous price series data such as international, national, long and short flights, different providers (low cost and regular) into an interpolated price series trajectory that can be fed to an unsupervised clustering algorithm.

Once the MPP step is completed, the model applies clustering followed by classification and statistical processing techniques on historical price data to develop price decrease event predictive rules. First, the price series trajectory is clustered into groups based on similar pricing behavior. Next, a price evolution model that estimates price change patterns up to departure date is defined for each cluster.

For a new test dataset, a tree-based classification algorithm is used to select the best matching cluster and then the corresponding price evolution model defined for that cluster is used to predict the price decreasing event.

The dataset used by this research is obtained from Liligo.com's historical price data collected for 28 days.

It covers data for 6 routes from 9 airlines. Unlike others, this paper also considers round-trips for 3, 7 and 14 days. The set of features in the analysis include: departure station, arrival station, departure date, return date, provider, day of week, day of month, day of year and demand. The authors claim that the model achieved 55% performance as compared to (Etzioni et al, 2003). However, no details of performance evaluation steps were presented.



The study by (Domínguez-Menchero et al., 2014) Suggested a model that predicts the optimal purchase timing based on non-parametric isotonic regression techniques for a specific route, time period and airlines. The model determines the maximum number of days users might wait before purchasing ticket without significant price increase and the daily money loss that comes from delaying the purchase.

Two types of variables are considered for the prediction: price and date of purchase. The authors analyzed four routes for direct flights and one-stop flights based on a two-month period daily price information that is extracted 30 days prior to departure date.

They found that purchasing a ticket up to 18 days prior to departure incurs no significant economic loss. The authors claim that the isotonic method is advantageous in that this effect cannot be achieved with other types of regression techniques.

### **Ticket price prediction**

The authors in (Anastasia Lantseva et al., 2015) Proposed a ticket prediction model based on an empirical data-driven Regression Model. The model predicts the price per kilometer for a given flight within 90 days before departure date.

Two kinds of flights (local and international) were considered for the study based on data collected from two independent ticket price information aggregators (Avia Sales and Sabre) in spring 2015.

For local flights, they used flights from two Russian cities (Moscow and Saint-Petersburg) to 50 local Russian cities. Flights from the same two cities (Moscow and Saint-Petersburg) to 40 international destinations were considered for international flights with the domination of European cities.

The minimum price for each flight per day was collected over a period of 75 days for Avia Sales and 90 days for Saber. The features used for building the model include city of departure, destination, ticket purchase date, departure date, ticket options with the price. Based on the proposed model, the authors compared the effect of early ticket purchasing on the price of tickets for local and global flights.

It was found that early ticket purchasing has an advantage for international flights while local flight required additional investigations to reach concrete conclusions. The authors did not provide performance evaluations of the model. Moreover, the dataset used by (Anastasia

Lantseva et al., 2015) Was limited since it was collected over a short period and for specific routes.

### **1.1.2 Airlines side models**

Airlines side models represent studies targeting profit gained by airlines and OTAs. Two main categories of researches exist in the literature regarding this. The first group proposes demand prediction models while the second group focuses on price discrimination.

#### **Demand prediction**

The model outperforms previous models based on three performance metrics: Pearson Correlation Coefficient (CC),  $R^2$ , and Mean Absolute Error (MAE). The Correlation Coefficient was 0.95 for market share and 0.98 for demand as compared to 0.82 and 0.77 for previous models. However, the proposed model has higher time overheads in comparison with previous models because of the additional time for clustering and more advanced regression methods.

The same authors above provided the extension of their work in another article (Bo An et al., 2017) where they introduce two new concepts on the basic Frequency-Based Profit Maximization algorithms in order to capture the conservative nature of airlines in deciding flight frequencies: bounded frequency and long-term profits.

The tighter frequency bounds capture the scenario where airlines make only bounded changes to the frequencies even if it is more profitable to change the frequency by a large number.

The second case expresses the case where airlines try to “drive off” other competing airlines and gain potential future profits by maintaining high frequency numbers which might not bring profit currently. Consideration of these two factors gave better profit maximization and proved that airlines are conservative in changing their frequencies and more concerned about long-term profits.

The extension also investigated how optimal frequencies and profits can be calculated for the case where multiple airlines are strategic and independently change their frequencies (the original method assumes that only one airline is strategic in deciding its frequencies of a certain set of routes).

The decision of customers to buy a ticket for a given flight and route depends on various factors such as airlines’ market share, customer membership (loyalty), and travelers’ personal

preferences of popular cities for destination and popular airlines for travel etc. (Jie Liu et al., 2017a, Liu et al., 2017b).

The article proposed a probabilistic framework model that enables to model airline customer travel preferences and to predict personalized airline passenger demand i.e. the destination and the airline an individual customer will choose. This is among the first works that proposes personalized air travel demand prediction.

The approach utilizes Bayesian network-based topic model named Relational Travel Topic Model (RTTM) to model the preferences of customers and the characteristics of air routes and airline companies. Demand prediction is formulated using a Multiple Factor Travel Prediction (MFTP) framework that integrates multiple factors that influence the decision of customer's travel.

Experiments are performed based on a 2-year passenger travel records of two cities in China (Beijing and Guangzhou,) consisting of more than 50 million flight records from more than 3 million customers with a total of around 550 air routes and 60 carrier companies.

The data of the first year is used to train the models while the second-year data is used to test the models. The data is gained from airline reservation systems in the form of so-called passenger name records (PNRs). The PNRs contain the itinerary information of passengers and includes user-related information such as ID number, name, and gender, and flight-related information such as airline, origin and destination airport.

Experiment results indicated that the proposed method is effective in demand prediction. A closely related work to that of (Jie Liu et al., 2017a, Liu et al., 2017b) Is proposed by Han-Tao Yang and Xia Liu, 2018. The paper came up with a model that predicts the airline passenger volume based on the daily passenger data of the airline for the route from Beijing to Sanya for the period between 2010 and 2017.

The approach applied three types of prediction models: random forest, SVR and neural network. The result showed that the random forest prediction model achieved the highest accuracy with an MAPE of 4.18% followed by SVR: 6.87% and neural network: 12.38%.

## **Price discrimination**

As indicated by several previous research (Mantin Benny and Bonwoo Koo, 2010; Marco Alderighi et al., 2011, Puller and Taylor, 2012) airlines use various kinds of price discrimination mechanisms to charge customers different prices based on their willingness to pay for travel. However, most of the earlier studies are focused on testing a hypothesis to proof the existence of price discrimination and did not propose specific models or techniques for price discrimination.

Moreover, mainly day dependent price discrimination was considered. The authors in ([Steven L.Puller and Lisa M.Taylor, 2012](#)) conducted research to check if there exists price discrimination based on the day of the week in which the ticket was purchased.

A regression model is used to analyse ticket prices for the same flights purchased on different days of the week. The model considers controlling other factors which might affect the ticket price such as ticket restrictions (e.g. purchase deadline, travel restriction or duration of stay), factors that might influence the demand (e.g. the week of travel, time of the day) and the number of days before departure.

The model is tested based on ticket transaction data collected for 85 US domestic routes across six major airlines (American, Delta, United, Continental, USAir and Northwest) for the fourth quarter of 2004. However, the data considers only nonstop round-trips.

The information contained in the data included ticket price, the date of purchase, date of departure, the airline, route, flight number and service class. The test discovered that airlines charge 5% less price for similar tickets purchased on weekends as compared to tickets purchased on weekdays.

This finding is in line with other studies such as ([Mumbower et al., 2014](#)) which concluded that customers who prefer to purchase tickets on weekends as leisure customers are more price elastic than customers who choose to purchase tickets on week days as business customers. The paper further investigated whether the weekend purchase effect is consistent with price discrimination or not using cross-sectional variation in route characteristics i.e. by testing the weekend effect for various routes serving different volume of leisure and business travellers.

A closely related work to that of ([Steven L.Puller and Lisa M.Taylor, 2012](#)) was done by ([Mantin Benny and Bonwoo Koo, 2010](#)). The authors investigated whether day of week dependent price discrimination existed or not. The authors performed an empirical analysis to test the claim that “price dispersion during weekends is larger than that during weekdays while the average price stays constant over all days of the week”.

The variables included in the equations governing the hypothesis are the number of days prior to the departure date and the day of the week. The data used for the hypothesis test is collected from [Farecast.com](#) website. It consisted of the lowest daily airfare history for 6 departure dates (each Wednesday between February 27, 2008 and April 2, 2008, and returning 7 days later) spanning 90 days prior to the departure date.

The data was gathered for 1000 randomly selected routes across all airlines resulting in approximately 540,000 observations per day. The test result showed that a strong weekend effect exists in the dispersion of ticket prices, but not in the price level. Therefore, the study concluded that airlines implement day dependent dynamic pricing discrimination. Moreover, the study indicated that this weekend effect is likely driven by the different types of consumers who purchase tickets on different days of the week.

### **1.1.3 Other studies related to ticket price and Demand.**

Besides the customer side and demand side models discussed in the previous two sections, there are also several other researchers that have been mainly conducted to investigate the role of various factors affecting ticket prices and demand, the factors determining the price elasticity of demand such as economic, demographic and geographic determinants for airline passengers is analyzed in their research finding indicated that price elasticity increases with time and leisure customers are more price sensitive than business customers.

The authors in (Silke J. Forbes, 2008) Analyzed the effect of air traffic delays on airline prices and found that prices fall by \$1.42 on average for each additional minute of flight delay.

The influence of purchase date and flight duration on the dispersion of airline ticket prices is studied in (Tomasz Szopiński and Robert Nowacki, 2015). According to this paper, price dispersion increases closer to the departure date and longer flights cause less price dispersion. Another study (María-Encarnación Andrés Martínez et al., 2017) Examined the determinants of airfare pricing including presence of low-cost carriers in the market, market domination, market share, and type of destination and reached on a conclusion that market dominance and the presence of low-cost airlines have strong effect on ticket prices.

## **Airline industry and customers**

The airline industry is considered as one of the most sophisticated industry in using complex pricing strategies. Nowadays, ticket prices can vary dynamically and significantly for the same flight, even for nearby seats. The ticket price of a specific flight can change up to 7 times a day.

Customers are seeking to get the lowest price for their ticket, while airline companies are trying to keep their overall revenue as high as possible and maximize their profit. However, mismatches between available seats and passenger demand usually leads to either the customer paying more or the airlines company losing revenue.

Airlines companies are generally equipped with advanced tools and capabilities that enable them to control the pricing process. However, customers are also becoming more strategic with the development of various online tools to compare prices across various airline companies. In addition, competition between airlines makes the task of determining optimal pricing is hard for everyone.

#### **1.1.4 Need for price prediction**

The last two decades have seen steadily increasing research targeting both customers and airlines. Customer side research focus on saving money for the customer while airline side studies are aimed at increasing the revenue of the airlines.

Conducted research employ a variety of techniques ranging from statistical techniques such as regression to different kinds of advanced data mining techniques.

From the customer point of view, determining the minimum price or the best time to buy a ticket is the key issue.

The conception of “tickets bought in advance are cheaper” is no longer working. It is possible that customers who bought a ticket earlier pay more than those who bought the same ticket later.

Moreover, early purchasing implies a risk of commitment to a specific schedule that may need to be changed usually for a fee. The ticket price may be affected by several factors thus may change continuously.

To address this, various studies were conducted to support the customer in determining an optimal ticket purchase time and ticket price prediction.

#### **1.1.5 Dynamic Pricing**

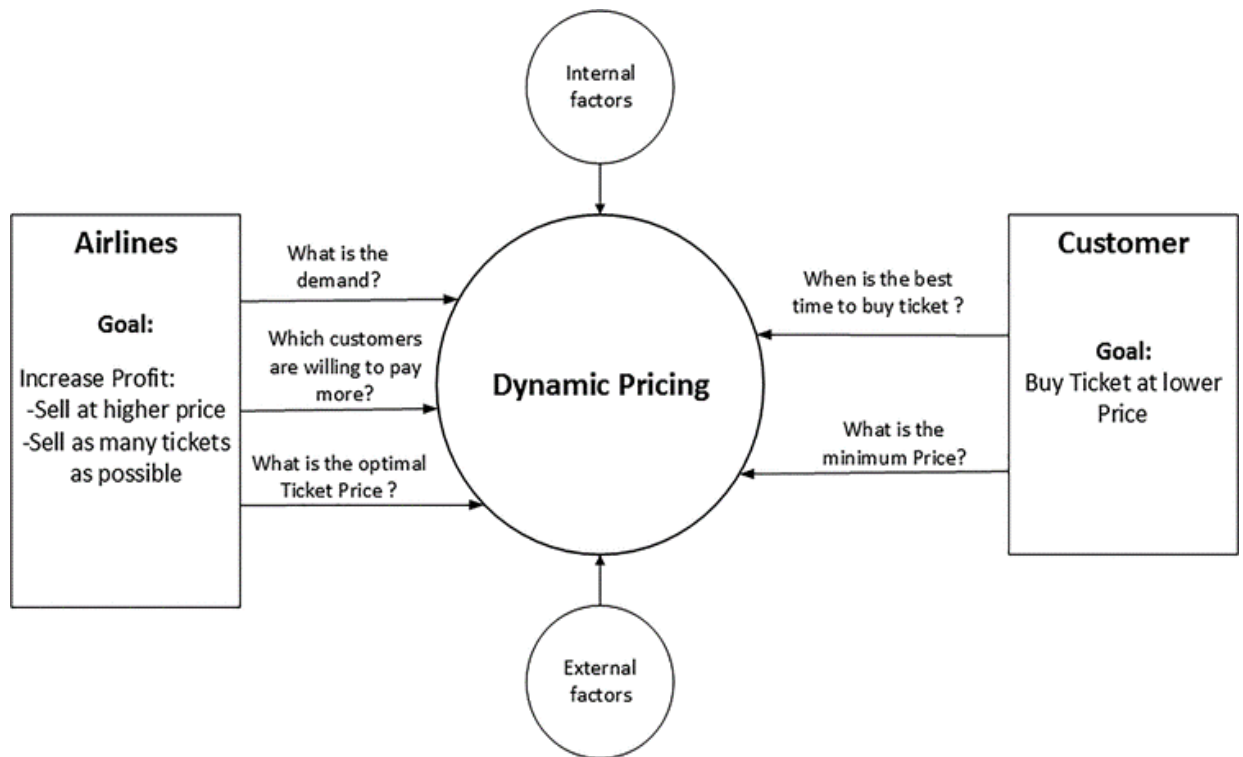
A significant number of research works exists that proposed prediction models for dynamic pricing in airlines which can be classified into two groups.

- demand prediction
- price discrimination.

Early prediction of the demand along a given route could help an airline company preplan the flights and determine appropriate pricing for the route. Existing demand prediction models generally try to predict passenger demand for a single flight/route and market share of an individual airline. Price discrimination allows an airline company to categorize customers based on their willingness to pay and thus charge them different prices.

Customers could be categorized into different groups based on various criteria such as business vs leisure, tourist vs normal traveler, profession etc. For example,

business customers are willing to pay more as compared to leisure customers as they rather focus on service quality than price.



**Fig 1. Dynamic Pricing**

## CHAPTER 2

### SOFTWARE REQUIREMENT SPECIFICATIONS

#### 2.1. Hardware and Software Requirements

##### Hardware

– Linux (Ubuntu 18.04 or equivalent) or Windows (7 or higher) machine with a minimum of 4 GB RAM and 50 GB disk space.

##### Software

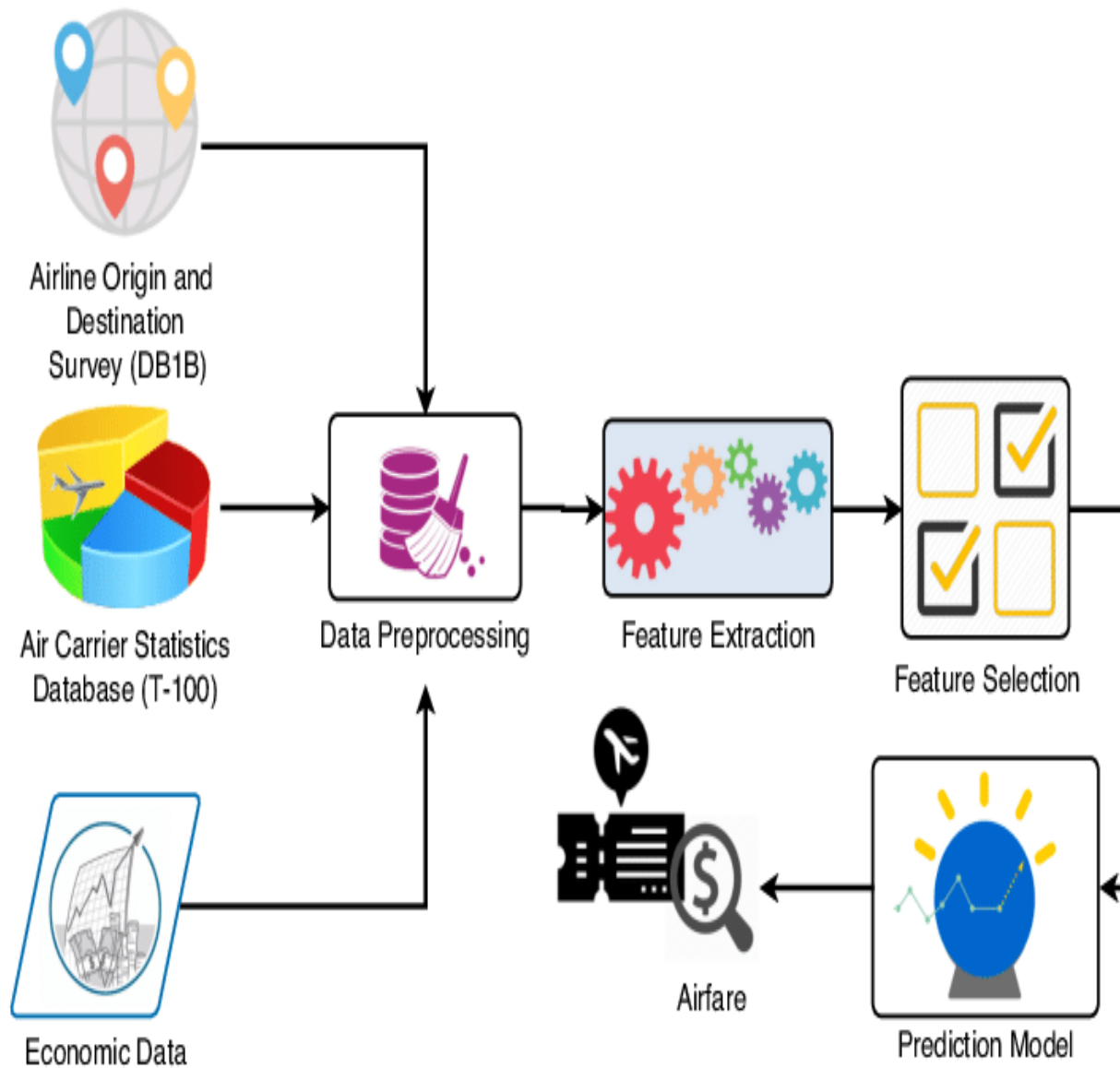
- Flask\_Cors 3.0.10
- Flask 2.0.1
- Pandas 1.2.4
- scikit\_learn 0.24.2
- Gunicorn 20.0.4
- Numpy 1.18.1
- python-dateutil 2.8.1
- Appdirs 1.4.3



## CHAPTER 3

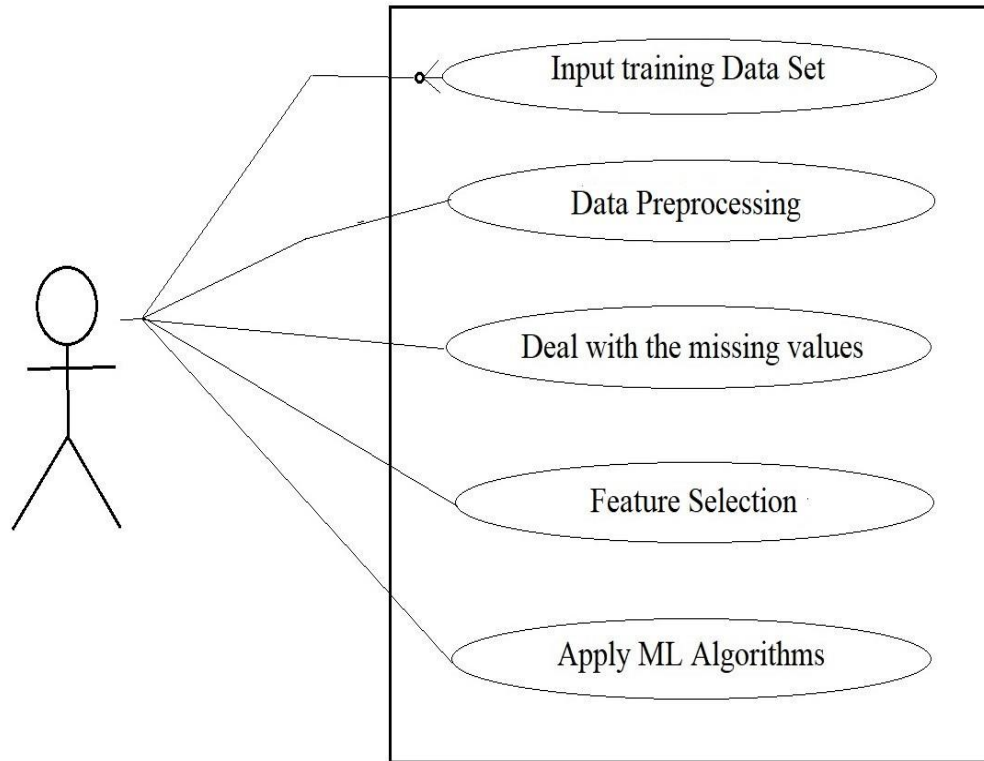
### DESIGN

#### 3.1 Architecture Diagram



**Fig.2.Architecture Diagram**

### 3.2. Use Case Diagram



**Fig.3. Use case Diagram**

#### 3.1.1. Input Training Data Set:

The training data is an initial set of data used to help a program understand how to apply technologies.

Training data is used to train the algorithms, so that it can accurately predict the outcome.

#### 3.1.2. Data Pre-processing:

Data pre-processing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model.

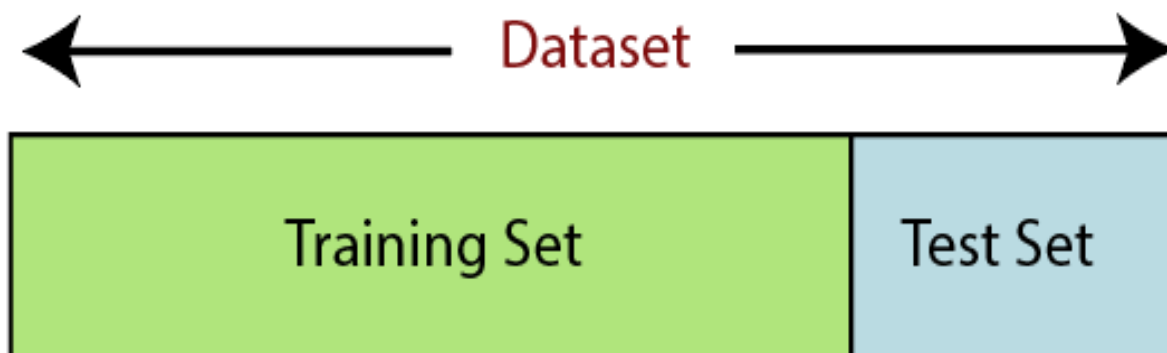
When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So, for this, we use data pre-processing task.

A real-world data contains noises, missing values, and in an unusable format which cannot be directly used for machine learning models. Data pre-processing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

In machine learning data pre-processing, we divide our dataset into a training set and test set. This is one of the crucial steps of data pre-processing as by doing this, we can enhance the performance of our machine learning model.

Suppose if we have given training to our machine learning model by a dataset and we test it by a completely different dataset. Then, it will create difficulties for our model to understand the correlations between the models.

If we train our model very well and its training accuracy is also extremely high, but we provide a new dataset to it, then it will decrease the performance. So, we always try to make a machine learning model which performs well with the training set and with the test dataset. Here, we can define these datasets as:



**Fig.4. Data sets**

**Training Set:** The training dataset is used to prepare a model, to train it. We pretend the test dataset is new data where the output values are withheld from the algorithm. We gather predictions from the trained model on the inputs from the test dataset and compare them to the withheld output values of the test set.

	A	B	C	D	E	F	G	H	I	J	K
1	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	Price
2	IndiGo	24/03/2019	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 Mar	2h 50m	non-stop	No info	3897
3	Air India	1/05/2019	Kolkata	Banglore	CCU → IXR → BBI →	05:50	13:15	7h 25m	2 stops	No info	7662
4	Jet Airwa	9/06/2019	Delhi	Cochin	DEL → LKO → BOM	09:25	04:25 10 Jun	19h	2 stops	No info	13882
5	IndiGo	12/05/2019	Kolkata	Banglore	CCU → NAG → BLR	18:05	23:30	5h 25m	1 stop	No info	6218
6	IndiGo	01/03/2019	Banglore	New Delhi	BLR → NAG → DEL	16:50	21:35	4h 45m	1 stop	No info	13302
7	SpiceJet	24/06/2019	Kolkata	Banglore	CCU → BLR	09:00	11:25	2h 25m	non-stop	No info	3873
8	Jet Airwa	12/03/2019	Banglore	New Delhi	BLR → BOM → DEL	18:55	10:25 13 Mar	15h 30m	1 stop	In-flight meal nc	11087
9	Jet Airwa	01/03/2019	Banglore	New Delhi	BLR → BOM → DEL	08:00	05:05 02 Mar	21h 5m	1 stop	No info	22270
10	Jet Airwa	12/03/2019	Banglore	New Delhi	BLR → BOM → DEL	08:55	10:25 13 Mar	25h 30m	1 stop	In-flight meal nc	11087
11	Multiple	27/05/2019	Delhi	Cochin	DEL → BOM → COK	11:25	19:15	7h 50m	1 stop	No info	8625
12	Air India	1/06/2019	Delhi	Cochin	DEL → BLR → COK	09:45	23:00	13h 15m	1 stop	No info	8907
13	IndiGo	18/04/2019	Kolkata	Banglore	CCU → BLR	20:20	22:55	2h 35m	non-stop	No info	4174
14	Air India	24/06/2019	Chennai	Kolkata	MAA → CCU	11:40	13:55	2h 15m	non-stop	No info	4667
15	Jet Airwa	9/05/2019	Kolkata	Banglore	CCU → BOM → BLR	21:10	09:20 10 May	12h 10m	1 stop	In-flight meal nc	9663
16	IndiGo	24/04/2019	Kolkata	Banglore	CCU → BLR	17:15	19:50	2h 35m	non-stop	No info	4804
17	Air India	3/03/2019	Delhi	Cochin	DEL → AMD → BOM	16:40	19:15 04 Mar	26h 35m	2 stops	No info	14011
18	SpiceJet	15/04/2019	Delhi	Cochin	DEL → PNQ → COK	08:45	13:15	4h 30m	1 stop	No info	5830
19	Jet Airwa	12/06/2019	Delhi	Cochin	DEL → BOM → COK	14:00	12:35 13 Jun	22h 35m	1 stop	In-flight meal nc	10262
20	Air India	12/06/2019	Delhi	Cochin	DEL → CCU → BOM	20:15	19:15 13 Jun	23h	2 stops	No info	13381
21	Jet Airwa	27/05/2019	Delhi	Cochin	DEL → BOM → COK	16:00	12:35 28 May	20h 35m	1 stop	In-flight meal nc	12898
22	GoAir	6/03/2019	Delhi	Cochin	DEL → BOM → COK	14:10	19:20	5h 10m	1 stop	No info	19495
23	Air India	21/03/2019	Banglore	New Delhi	BLR → COK → DEL	22:00	13:20 19 Mar	15h 20m	1 stop	No info	6955
24	IndiGo	3/04/2019	Banglore	Delhi	BLR → DEL	04:00	06:50	2h 50m	non-stop	No info	3943

**Fig.5.Training Data Set**

**Test set:** A subset of dataset to test the machine learning model, and by using the test set, model predicts the output.

	A	B	C	D	E	F	G	H	I	J	K
1	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info	
2	Jet Airways	6/06/2019	Delhi	Cochin	DEL → BOM → COK	17:30	04:25 07 Jun	10h 55m	1 stop	No info	
3	IndiGo	12/05/2019	Kolkata	Banglore	CCU → MAA → BLR	06:20	10:20	4h	1 stop	No info	
4	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	
5	Multiple car	21/05/2019	Delhi	Cochin	DEL → BOM → COK	08:00	21:00	13h	1 stop	No info	
6	Air Asia	24/06/2019	Banglore	Delhi	BLR → DEL	23:55	02:45 25 Jun	2h 50m	non-stop	No info	
7	Jet Airways	12/06/2019	Delhi	Cochin	DEL → BOM → COK	18:15	12:35 13 Jun	18h 20m	1 stop	In-flight meal not included	
8	Air India	12/03/2019	Banglore	New Delhi	BLR → TRV → DEL	07:30	22:35	15h 5m	1 stop	No info	
9	IndiGo	1/05/2019	Kolkata	Banglore	CCU → HYD → BLR	15:15	20:30	5h 15m	1 stop	No info	
10	IndiGo	15/03/2019	Kolkata	Banglore	CCU → BLR	10:10	12:55	2h 45m	non-stop	No info	
11	Jet Airways	18/05/2019	Kolkata	Banglore	CCU → BOM → BLR	16:30	22:35	6h 5m	1 stop	No info	
12	Jet Airways	21/03/2019	Delhi	Cochin	DEL → MAA → BOM	13:55	18:50 22 Mar	28h 55m	2 stops	In-flight meal not included	
13	IndiGo	15/06/2019	Delhi	Cochin	DEL → HYD → COK	06:50	16:10	9h 20m	1 stop	No info	
14	Multiple car	15/05/2019	Delhi	Cochin	DEL → BOM → COK	09:00	19:15	10h 15m	1 stop	No info	
15	Jet Airways	12/03/2019	Banglore	New Delhi	BLR → BOM → DEL	05:45	10:25	4h 40m	1 stop	No info	
16	Jet Airways	3/06/2019	Delhi	Cochin	DEL → BOM → COK	19:15	12:35 04 Jun	17h 20m	1 stop	In-flight meal not included	
17	Jet Airways	06/03/2019	Banglore	New Delhi	BLR → BOM → DEL	21:25	08:15 07 Mar	10h 50m	1 stop	No info	
18	Multiple car	6/06/2019	Delhi	Cochin	DEL → HYD → COK	13:15	22:30	9h 15m	1 stop	No info	
19	Vistara	24/03/2019	Kolkata	Banglore	CCU → DEL → BLR	09:55	22:10	12h 15m	1 stop	No info	
20	Jet Airways	12/06/2019	Delhi	Cochin	DEL → BOM → COK	19:15	04:25 13 Jun	9h 10m	1 stop	In-flight meal not included	
21	Jet Airways	12/03/2019	Banglore	New Delhi	BLR → BOM → DEL	22:55	08:15 13 Mar	9h 20m	1 stop	No info	
22	IndiGo	6/03/2019	Delhi	Cochin	DEL → BOM → COK	10:45	01:35 07 Mar	14h 50m	1 stop	No info	
23	Jet Airways	9/05/2019	Kolkata	Banglore	CCU → BOM → BLR	20:00	10:05 10 May	14h 5m	1 stop	In-flight meal not included	
24	Jet Airways	18/03/2019	Banglore	New Delhi	BLR → BOM → DEL	21:25	09:00 16 Mar	11h 35m	1 stop	In-flight meal not included	

**Fig.6. Testing Data Set**

### 3.1.3 Deal with missing values:

Data can have missing values for several reasons such as observations that were not recorded and data correction. Complete removal of data with missing values results in robust and highly accurate model.

There are two options to deal with the missing values:

1. Drop columns with missing values
2. Imputation

### 3.1.4. Feature Selection:

Feature selection in machine learning refers to the process of choosing the most relevant features in our data to give to our model. By limiting the number of features, we use (rather than just feeding the model the unmodified data), we can often speed up training and improve accuracy, or both.

Finding out the best feature which will contribute and have good relation with target variable. Following are some of the feature selection methods,

**1. HeatMap:** Correlation states how the features are related to each other or the target variable. Correlation can be positive (increase in one value of feature increases the value of the target variable) or negative (increase in one value of feature decreases the value of the target variable).

Heatmap makes it easy to identify which features are most related to the target variable, we will plot heatmap of correlated features using the seaborn library.

**2. Feature importance:** You can get the feature importance of each feature of your dataset by using the feature importance property of the model. Feature importance gives you a score for each feature of your data, the higher the score more important or relevant is the feature towards your output variable.

Feature importance is an inbuilt class that comes with Tree Based Classifiers, we will be using Extra Tree Classifier for extracting the top 10 features for the dataset.

3.**SelectKBest**: Select features according to the k highest scores. Function taking two arrays X and y, and returning a pair of arrays (scores, pvalues) or a single array with scores.

### 3.1.5. Apply ML Algorithms:

At its most basic, machine learning uses programmed algorithms that receive and analyse input data to predict output values within an acceptable range. As new data is fed to these algorithms, they learn and optimise their operations to improve performance, developing intelligence over time.



## CHAPTER 4

### IMPLEMENTATION

#### 4.1. Data Collection & Processing of Data:

The data is collected from the Kaggle. This data set consists of 10683 records with 13 columns that explain about the flight in Indian Airlines. Then, the pre-processing of data includes the cleaning of data to separate important data and error data that brings bias in developing the flight price prediction that we want to build by eliminating the null values. Pandas, Seaborn, Numpy and matplotlib are the libraries used to visualize and analyse the dataset.

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

sns.set()
```

#### Importing dataset

1. Since data is in form of excel file we have to use pandas read\_excel to load the data
2. After loading it is important to check the complete information of data as it can indicate many of the hidden information such as null values in a column or a row
3. Check whether any null values are there or not. If it is present then following can be done,
  - A. Imputing data using Imputation method in sklearn
  - B. Filling NaN values with mean, median and mode using fillna() method
4. Describe data --> which can give statistical analysis

**Fig.7. Importing Dataset**

#### Imports

**1.NumPy:** It is used for working with arrays and stands for Numerical Python. Pandas: It is most popular python library which is used for data analysis

**2.Matplotlib:** It is a comprehensive library for creating static, animated, interactive visualizations in python.

**2.a.Matplotlib.pyplot** - It is a collection of functions that makes matplotlib work like MATLAB.

**3.Seaborn:** It is a python data visualisation library based on matplotlib. This library provides high level interface



**4.Sklearn:** Scikit-learn is probably the most useful library for machine learning in Python. The sklearn library contains a lot of efficient tools for machine learning and statistical modelling including classification, regression, clustering and dimensionality reduction.

## 4.2. Exploratory Data Analysis (EDA)

The exploratory Data Analysis plays a major role in the prediction of ticket price. It produces output for the model that created by the Machine learning techniques. The important part in the Exploratory Data Analysis (EDA) is handling categorical data, which helps to label the idea.

### Handling Categorical Data:

**1.Nominal Data:** The data which are not in any order. OneHotEncoding is used in this case to handle the data. In OneHotEncoding integer encoded variable is removed and one new binary variable is added for each unique integer value in the variable.

```
In [26]: # As Source is Nominal Categorical data we will perform OneHotEncoding
Source = train_data[["Source"]]
Source = pd.get_dummies(Source, drop_first= True)
Source.head()
```

```
Out[26]:
```

	Source_Chennai	Source_Delhi	Source_Kolkata	Source_Mumbai
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	0

Fig.8. OneHotEncoding

**2.Ordinal Data:** The data which are in order. Label Encoder is used in this case to handle the data. Label Encoding refers to converting the labels into numeric form to convert it into

the machine-readable form. Machine learning algorithms can then decide in a better way on how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning.

```
In [31]: train_data["Total_Stops"].value_counts()
```

```
Out[31]: 1 stop      5625
non-stop   3491
2 stops    1520
3 stops     45
4 stops     1
Name: Total_Stops, dtype: int64
```

```
In [32]: # As this is case of Ordinal Categorical type we perform LabelEncoder
# Here Values are assigned with corresponding keys

train_data.replace({"non-stop": 0, "1 stop": 1, "2 stops": 2, "3 stops": 3, "4 stops": 4}, inplace = True)
```

```
In [33]: train_data.head()
```

```
Out[33]:
```

	Airline	Source	Destination	Total_Stops	Price	Journey_day	Journey_month	Dep_hour	Dep_min	Arrival_hour	Arrival_min	Duration_hours	Duration
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

**Fig.9. Label Encoding**

## Outcomes of this module

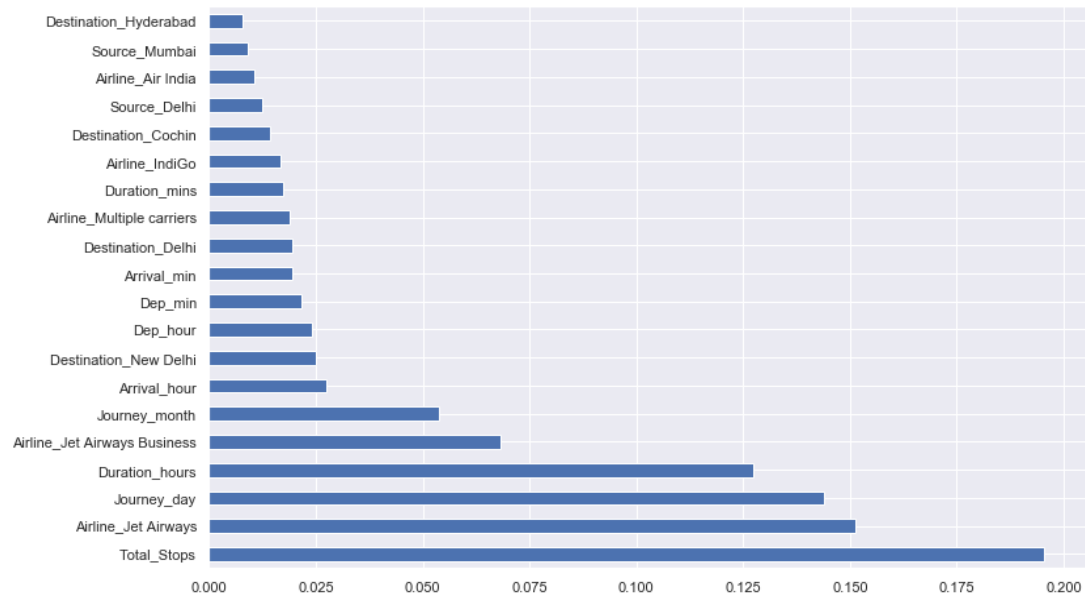
- Understanding the given dataset and helps clean up the given dataset.
- It gives you a clear picture of the features and the relationships between them.
- Providing guidelines for essential variables and leaving behind/removing non-essential variables.
- Handling Missing values or human error.
- Identifying outliers.
- EDA process would be maximizing insights of a dataset.

### 4.3. Test Data:

The testing of the data is done to avoid data leakage. Then, it helps in the prediction of price and to select the best feature, which has good relationship with the target variable.

```
In [50]: #plot graph of feature importances for better visualization
```

```
plt.figure(figsize = (12,8))  
feat_importances = pd.Series(selection.feature_importances_, index=X.columns)  
feat_importances.nlargest(20).plot(kind='barh')  
plt.show()
```



**Fig.10. Visualization**

### 4.4. Applying the ML Algorithms:

Various Machine Learning algorithms can be used but random forest is the suitable algorithm to get more accuracy rate. The Random Forest is used for the analysis by splitting the data using scikitlearn. The accuracy may increase by implementing hyperparameter tuning, which consists of RandomizedSearchCV and GridSearchCV and then we can plot and predict the result.

## Random Forest Algorithm:

The random forest combines hundreds or thousands of decision trees, trains each one on a slightly different set of the observations, splitting nodes in each tree considering a limited number of the features. The final predictions of the random forest are made by averaging the predictions of each individual tree.

### Fitting model using Random Forest

1. Split dataset into train and test set in order to prediction w.r.t  $X_{test}$
2. If needed do scaling of data
  - Scaling is not done in Random forest
3. Import model
4. Fit the data
5. Predict w.r.t  $X_{test}$
6. In regression check **RSME** Score
7. Plot graph

```
In [51]: from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 42)
```

```
In [52]: from sklearn.ensemble import RandomForestRegressor
reg_rf = RandomForestRegressor()
reg_rf.fit(X_train, y_train)
```

```
Out[52]: RandomForestRegressor(bootstrap=True, ccp_alpha=0.0, criterion='mse',
                                max_depth=None, max_features='auto', max_leaf_nodes=None,
                                max_samples=None, min_impurity_decrease=0.0,
                                min_impurity_split=None, min_samples_leaf=1,
                                min_samples_split=2, min_weight_fraction_leaf=0.0,
                                n_estimators=100, n_jobs=None, oob_score=False,
                                random_state=None, verbose=0, warm_start=False)
```

```
In [53]: y_pred = reg_rf.predict(X_test)
```

```
In [54]: reg_rf.score(X_train, y_train)
```

```
Out[54]: 0.9539164511170628
```

```
In [55]: reg_rf.score(X_test, y_test)
```

```
Out[55]: 0.798383043987616
```

```
In [56]: sns.distplot(y_test-y_pred)
plt.show()
```

**Fig.11. Random Forest**

## Tuning:

### Hyperparameter Tuning

- Choose following method for hyperparameter tuning
  1. **RandomizedSearchCV** --> Fast
  2. **GridSearchCV**
- Assign hyperparameters in form of dictionary
- Fit the model
- Check best parameters and best score

```
In [62]: from sklearn.model_selection import RandomizedSearchCV
```

```
In [63]: #Randomized Search CV

# Number of trees in random forest
n_estimators = [int(x) for x in np.linspace(start = 100, stop = 1200, num = 12)]
# Number of features to consider at every split
max_features = ['auto', 'sqrt']
# Maximum number of levels in tree
max_depth = [int(x) for x in np.linspace(5, 30, num = 6)]
# Minimum number of samples required to split a node
min_samples_split = [2, 5, 10, 15, 100]
# Minimum number of samples required at each leaf node
min_samples_leaf = [1, 2, 5, 10]
```

```
In [64]: # Create the random grid

random_grid = {'n_estimators': n_estimators,
               'max_features': max_features,
               'max_depth': max_depth,
               'min_samples_split': min_samples_split,
               'min_samples_leaf': min_samples_leaf}
```

```
In [65]: # Random search of parameters, using 5 fold cross validation,
# search across 100 different combinations
rf_random = RandomizedSearchCV(estimator = reg_rf, param_distributions = random_grid, scoring='neg_mean_squared_error', n_iter = 10, cv = 5, verbose=2, random_state=42, n_jobs = 1)
```

**Fig.12. Hyperparameter tuning**

## CHAPTER 5

### TESTING

#### 5.1. Test dataset

##### Test set

```
In [39]: test_data = pd.read_excel(r"E:\MachineLearning\EDA\Flight_Price\Test_set.xlsx")
```

```
In [40]: test_data.head()
```

```
Out[40]:
```

	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

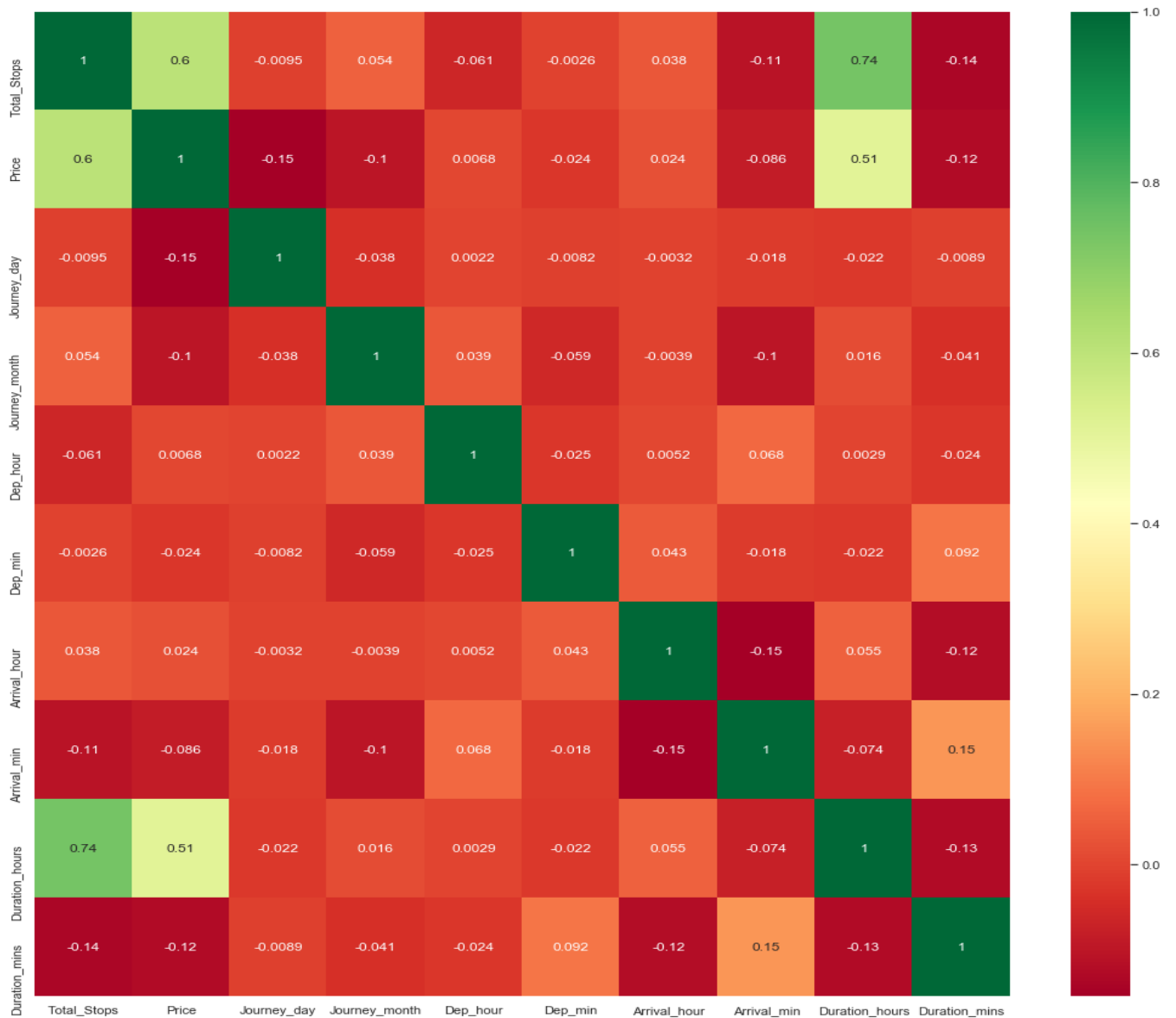
**Fig.13. Sample test data**

#### 5.2. Finding correlation between Independent and dependent attribute:

This shows how strongly the attributes and the target variables are correlated.

##### HeatMap:

A heat map (or heatmap) is a graphical representation of data where values are depicted by color. Heat maps make it easy to visualize complex data and understand it at a glance: THE DATA ON THE LEFT IS THE SAME AS THAT ON THE RIGHT—BUT ONE IS MUCH EASIER TO UNDERSTAND.



**Fig.14. Heatmap**

### 5.3. Regression Model Evaluation:

MSE, RMSE, R-squared are used to evaluate the prediction error and performance of the model is tested using these metrics.

```

In [58]: from sklearn import metrics

In [59]: print('MAE:', metrics.mean_absolute_error(y_test, y_pred))
          print('MSE:', metrics.mean_squared_error(y_test, y_pred))
          print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))

MAE: 1172.5455945373583
MSE: 4347276.1614450775
RMSE: 2085.0122688955757

In [60]: # RMSE/(max(DV)-min(DV))
          2090.5509/(max(y)-min(y))

Out[60]: 0.026887077025966846

In [61]: metrics.r2_score(y_test, y_pred)

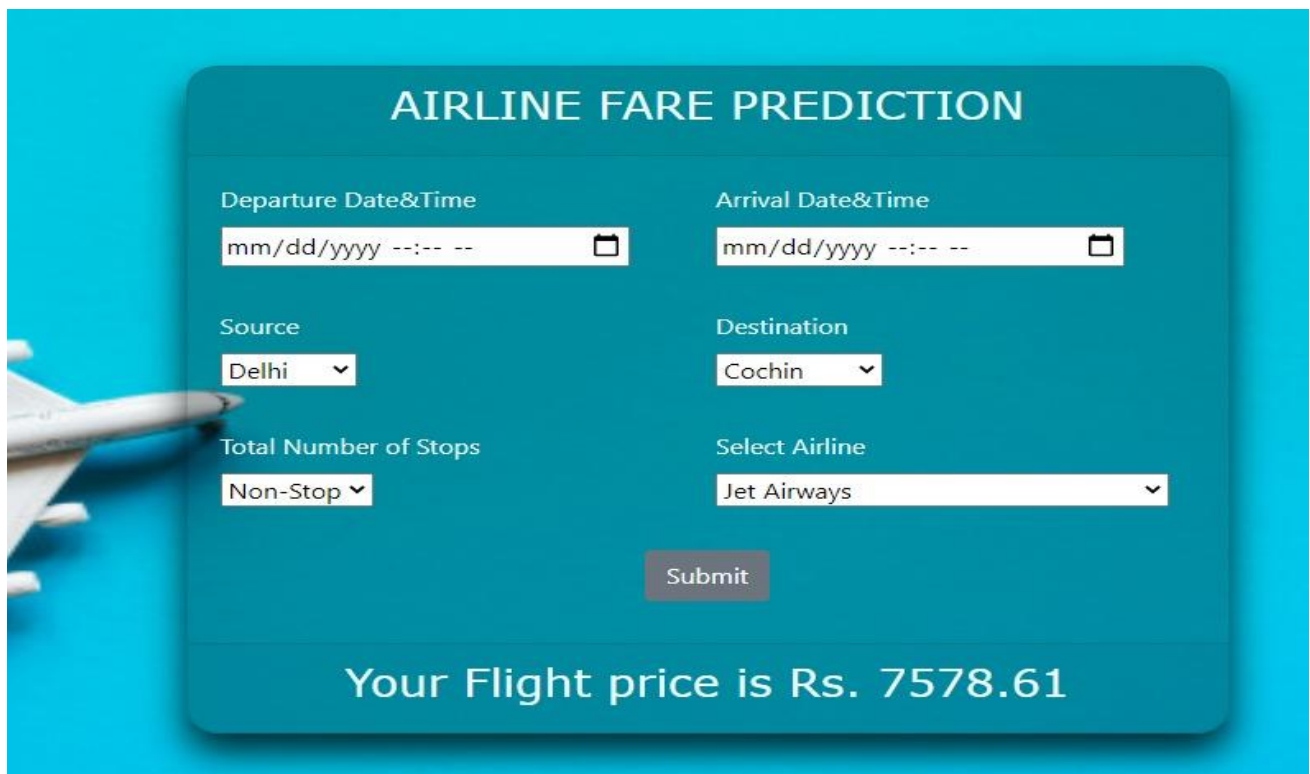
Out[61]: 0.7983830439876158

```

**Fig.15. Metrics evaluation**

The project is tested upon various test cases whose details and corresponding results are described in this chapter.

### Case 1: Long Distance Flight between two cities



**AIRLINE FARE PREDICTION**

Departure Date&Time: mm/dd/yyyy --:-- --

Arrival Date&Time: mm/dd/yyyy --:-- --

Source: Delhi

Destination: Cochin

Total Number of Stops: Non-Stop


Select Airline: Jet Airways

**Submit**

**Your Flight price is Rs. 7578.61**



## Case 2: Major Routes in the country



**AIRLINE FARE PREDICTION**

Departure Date&Time mm/dd/yyyy --:-- --	Arrival Date&Time mm/dd/yyyy --:-- --
Source Delhi	Destination Kolkata
Total Number of Stops Non-Stop	Select Airline Jet Airways
Submit	

**Your Flight price is Rs. 4760.46**

## **CONCLUSION**

After the feature selection the model will be hyper tuned with the help of Random Forest algorithm. Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions; it predicts the final output. In future if more data could be accessed such as the current availability of seats the predicted results will be more accurate.

## **FUTURE ENHANCEMENTS**

- One of the future directions that has great potential to improve the ticket price and demand prediction is to use the latest and advanced machine learning techniques (like deep learning) in conjunction with valuable social media-based data.
- Airline ticket prices/demand could be influenced by several dynamic factors which can be captured from social media.
- This project can be upgraded to handle large number of routes and for international travel.
- The ticket booking system and details about number of available seats can improve the performance of the model.

## REFERENCES

1. "Do we need hundreds of classifiers to solve real world classification problems".By,Amorim D.G.Barri,S.Cernadas&DelgadoMF(2014)<https://jmlr.org/papers/volume15/delgado14a/delgado14a.pdf>
2. W. Groves and M. Gini, —An agent for optimizing airline ticket purchasing, || 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013), St. Paul, MN, May 06 - 10, 2013, pp. 1341-1342.
3. T. Janssen, —A linear quantile mixed regression model for prediction of airline ticket prices, || Bachelor Thesis, Radboud University, 2014.
4. Viet Hoang Vu, Quang Tran Minh and Phu H. Phung,||An Airfare Prediction Model for Developing Markets||, IEEE paper 2018.
5. S.B. Kotsiantis, —Decision trees: a recent overview, || Artificial Intelligence Review, vol. 39, no. 4, pp. 261-283, 2013. [6] L. Breiman, —Random forests, || Machine Learning, vol. 45, pp. 5-32, 2001.
6. S. Haykin, Neural Networks – A Comprehensive Foundation. Prentice Hall, 2nd Edition, 1999.

## **Appendix A – Abbreviations**

**EDA** – Exploratory Data Analysis

**RMSE**- Root Mean Squared error

**MAE** – Mean Absolute error

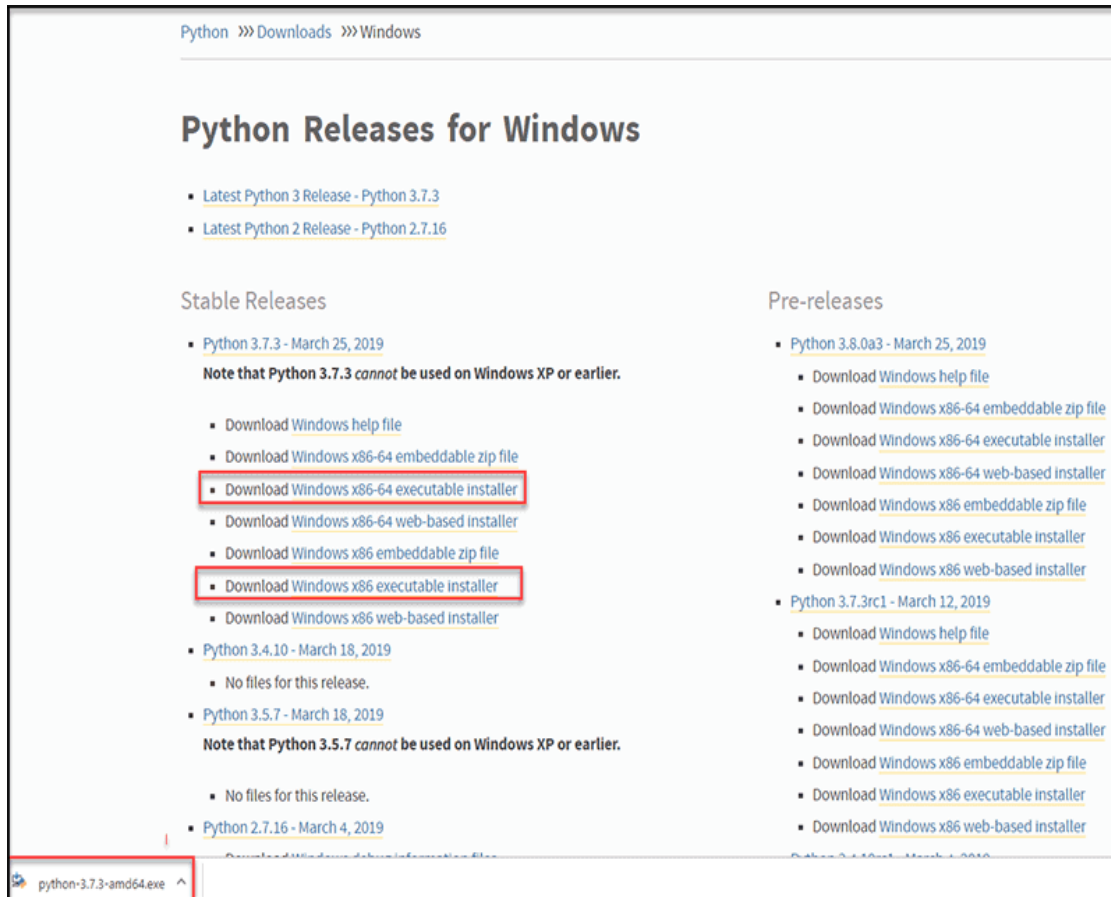
**MSE** – Mean squared error

**NaN**-Not a Number

## Appendix B – Software Installation Procedure

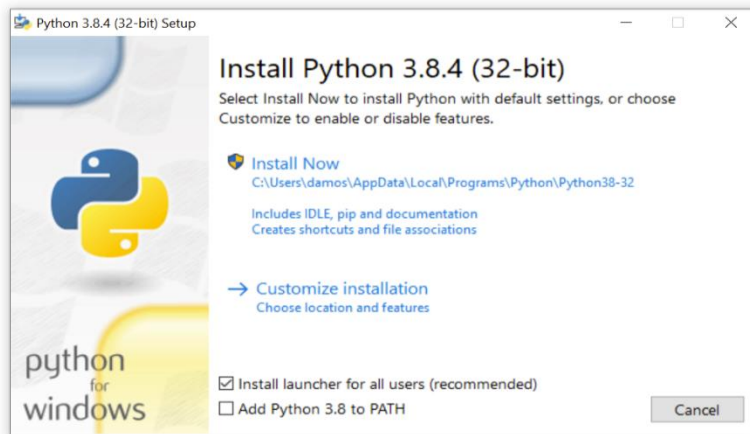
### 1. Python3 installation

**Step1:** Select Version of Python to Install



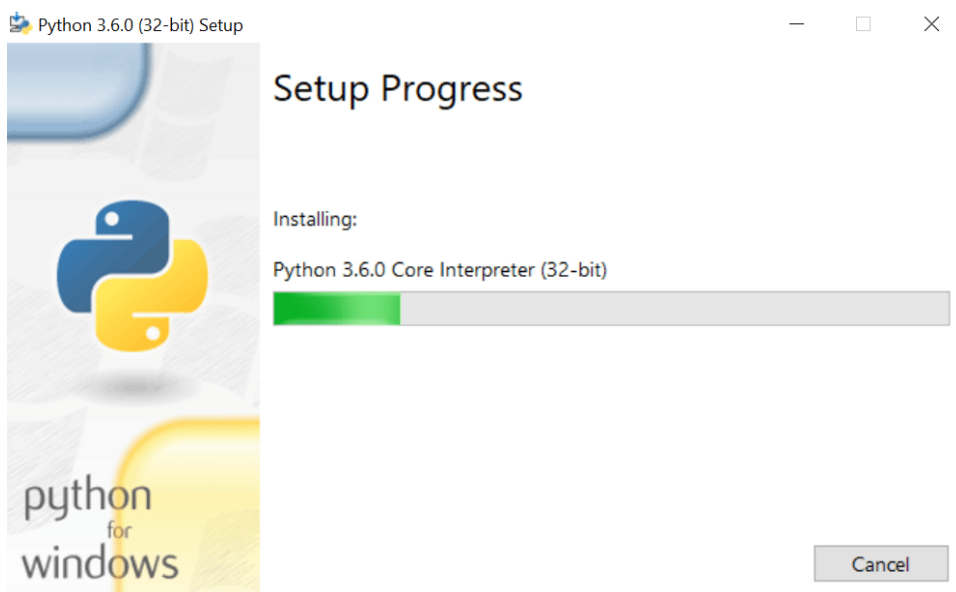
**Fig 16.**

**Step 2:** Download Python Executable Installer.



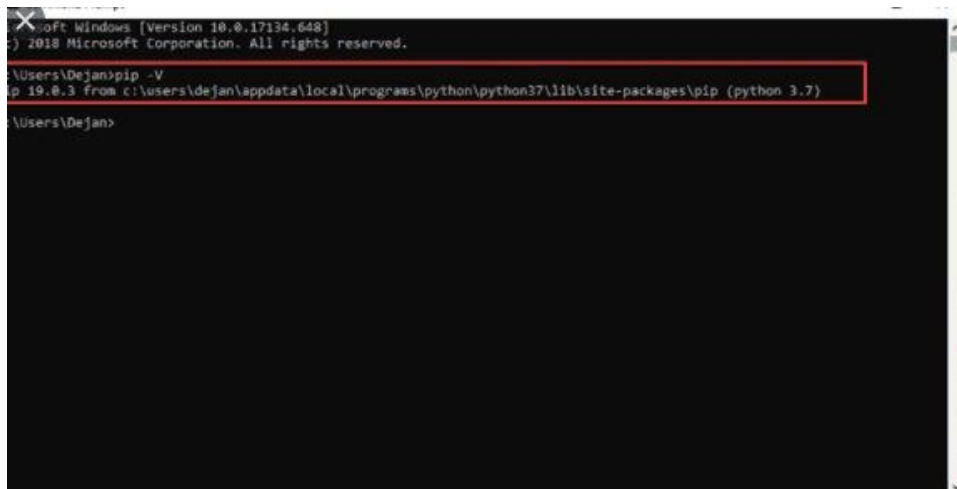
**Fig 17.**

**Step 3:** Run Executable Installer.



**Fig 18.**

**Step 4: Verify Python Was Installed on Windows.**



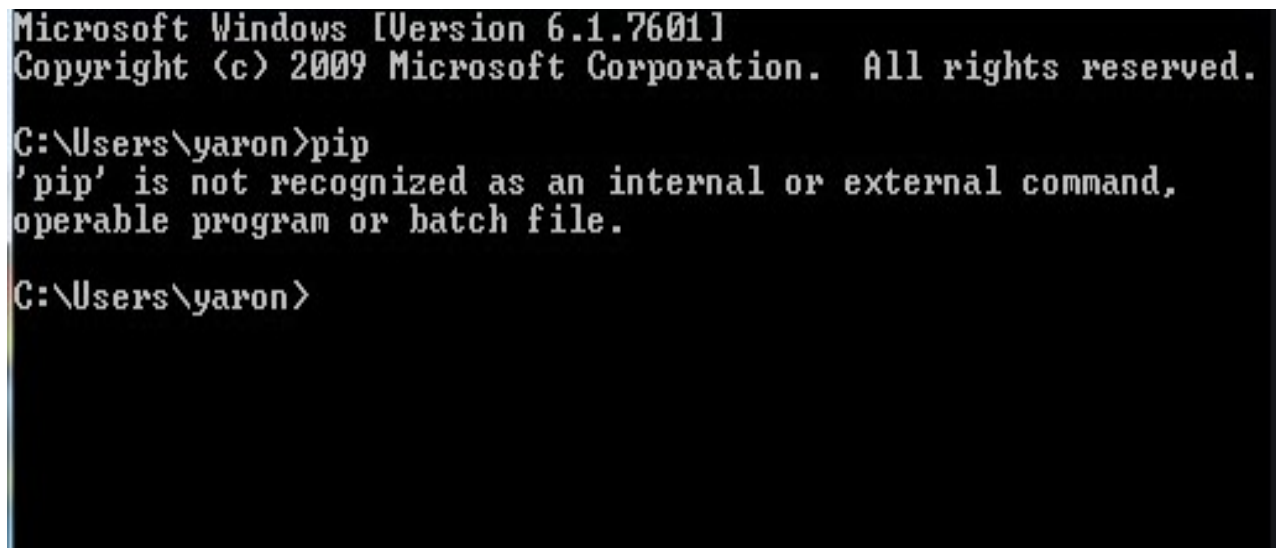
```
Microsoft Windows [Version 10.0.17134.648]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Dejan>pip -V
pip 19.0.3 from c:\users\dejan\appdata\local\programs\python\python37\lib\site-packages\pip (python 3.7)

C:\Users\Dejan>
```

**Fig 19.**

**Step 5: Verify Pip Was Installed.**



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\yaron>pip
'pip' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\yaron>
```

**Fig 20.**



## Step 6: Add Python Path to Environment Variables (Optional)

### 2. Jupyter notebook installation

1. If you use pip, you can install it with:

#### pip install jupyterlab

If installing using `pip install --user`, you must add the user-level bin directory to your PATH environment variable in order to launch jupyter lab. If you are using a Unix derivative (FreeBSD, GNU / Linux, OS X), you can achieve this by using `export PATH="$HOME/.local/bin:$PATH"` command.

```
1: !python --version
Python 3.5.2

2: import os
os.sys.path

3: [*,
"/usr/local/bin/kernel-launchers/python/scripts",
"/usr/lib/python3.5.zip",
"/usr/lib/python3.5",
"/usr/lib/python3.5/plat-x86_64-linux-gnu",
"/usr/lib/python3.5/lib-dynload",
"/usr/local/lib/python3.5/dist-packages",
"/usr/lib/python3/dist-packages",
"/usr/local/lib/python3.5/dist-packages/IPython/extensions",
"/home/jovyan/.ipython"]

4: ! ls -la /home/jovyan/.local/lib/python3.5/site-packages
total 16
drwx----- 4 jovyan users 4896 Apr  1 17:51 .
drwx----- 3 jovyan users 4896 Apr  1 17:51 ..
drwxr-xr-x 5 jovyan users 4896 Apr  1 17:51 cv2
drwxr-xr-x 2 jovyan users 4896 Apr  1 17:51 opencv_python-4.8.0.21.dist-info

5: !pip install --user --no-cache opencv-python
Collecting opencv-python
  Downloading https://files.pythonhosted.org/packages/39/de/288f66a8f57a8b32536c5f7ca5e883cb15ddae8032164ea192fa103d50f6/open
p35e-manylinux1_x86_64.whl (25.4MB)
  100% |#####| 25.4MB 22.8MB/s
Requirement already satisfied: numpy>=1.11.1 in /usr/local/lib/python3.5/dist-packages (from opencv-python) (1.15.4)
Installing collected packages: opencv-python
Successfully installed opencv-python
You are using pip version 10.1, however version 19.0.3 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.

6: import cv2
from PIL import Image
import numpy as np
import matplotlib
from matplotlib import pyplot as plt
import requests

ImportError                                Traceback (most recent call last)
/usr/local/bin/kernel-launchers/python/scripts/launch_ipynb.py in <module>
----> 1 import cv2
      2 from PIL import Image
```

Fig 21.

### Run JupyterLab

Once installed, launch JupyterLab with:

jupyterlab

## Appendix C – Software Usage Process

### How to use Python in Machine Learning:

1. Installing the **Python** and SciPy platform.
2. Loading the dataset.
3. Summarizing the dataset.
4. Visualizing the dataset.
5. Evaluating some algorithms.
6. Making some predictions.

### How To use Jupyter Notebook:

- 1.Explore raw data. Use a code cell to import the required Python libraries.
2. Feature and target columns.
3. Training and testing data sets.
4. Model training.
5. Save the model.
6. Inference with the model.