MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Project Report

on

Flight Fare Prediction Application

Submitted By:

Prateek Verma

0901CS191083

Faculty Mentor:

Mr. Mir Shahnawaz Ahmad

Assistant Professor, Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE
GWALIOR - 474005 (MP) est. 1957

MAY-JUNE 2022

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

Project Report

on

Flight Fare Prediction Application

A project report submitted in partial fulfilment of the requirement for the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Submitted by:

Prateek Verma

0901CS191083

Faculty Mentor:

Mr. Mir Shahnawaz Ahmad

Assistant Professor, Computer Science and Engineering

Submitted to:

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE
GWALIOR - 474005 (MP) est. 1957

MAY-JUNE 2022

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

CERTIFICATE

This is certified that **Prateek Verma** (0901CS191083) has submitted the project report titled **Flight Fare Prediction Application** under the mentorship of **Prof. Mr Mir Shahanawaz Ahmad,** in partial fulfilment of the requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering from Madhav Institute of Technology and Science, Gwalior.

09/05/21

Mr. Mir Shahnawaz Ahmad
Faculty Mentor
Assistant Professor
Computer Science and Engineering

09/05/2021

Dr. Manish Dixit
Professor and Head
Computer Science and Engineering

Dr. Manish Dixit
Professor & HOD
Department of CSE
M.I.T.S. Gwalior

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR

(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

DECLARATION

We hereby declare that the work being presented in this project report, for the partial fulfilment of requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering at Madhav Institute of Technology & Science, Gwalior is an authenticated and original record of my work under the mentorship of **Mr. Mir Shahnawaz Ahmad, Assistant Professor**, Computer Science and Engineering.

We declare that we have not submitted the matter embodied in this report for the award of any degree or diploma anywhere else.

Prateek Verma
0901CS191083
3rd Year
Computer Science and Engineering

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR
(A Govt. Aided UGC Autonomous & NAAC Accredited Institute Affiliated to RGPV, Bhopal)

ACKNOWLEDGEMENT

The full semester project has proved to be pivotal to my career. I am thankful to my institute, **Madhav Institute of Technology and Science,** for allowing me to continue my disciplinary project as a curriculum requirement, under the provisions of the Flexible Curriculum Scheme (based on the AICTE Model Curriculum 2018), approved by the Academic Council of the institute. I extend my gratitude to the Director of the institute, **Dr. R. K. Pandit** and Dean Academics, **Dr. Manjaree Pandit** for this.

I would sincerely like to thank my department, **Department of Computer Science and Engineering, for allowing** me to explore this project. I humbly thank **Dr. Manish Dixit**, Professor and Head, Department of Computer Science and Engineering, for his continued support during the course of this engagement, which eased the process and formalities involved.

I am sincerely thankful to my faculty mentors. I am grateful to the guidance of **Mr Mir Shahnawaz Ahmad**, **Assistant Professor**, Computer Science and Engineering for their continued support and guidance throughout the project. I am also very thankful to the faculty and staff of the department.

Prateek Verma
0901CS191083
3rd Year
Computer Science and Engineering

Abstract

Travelling through flights has become an integral part of today's lifestyle as more and more people are opting for faster travelling options. The flight ticket prices increase or decrease every now and then depending on various factors like timing of the flights, destination, duration of flights. various occasions such as vacations or festive season. Therefore, having some basic idea of the flight fares before planning the trip will surely help many people save money and time. In the proposed system a predictive model will be created by applying machine learning algorithms to the collected historical data of flights. This system will give people the idea about the trends that prices follow and also provide a predicted price value which they can refer to before booking their flight tickets to save money. This kind of system or service can be provided to the customers by flight booking companies which will help the customers to book their tickets accordingly.

सार:

उड़ानों के माध्यम से यात्रा करना आज की जीवनशैली का एक अभिन्न अंग बन गया है क्योंकि अधिक से अधिक लोग तेजी से यात्रा के विकल्प चुन रहे हैं। उड़ान के समय, गंतव्य, उड़ानों की अवधि जैसे विभिन्न कारकों के आधार पर उड़ान टिकट की कीमतें हर बार बढ़ती या घटती हैं। छुट्टियों या त्योहारी सीजन जैसे विभिन्न अवसरों पर। इसलिए, यात्रा की योजना बनाने से पहले उड़ान के किराए के बारे में कुछ बुनियादी विचार रखने से निश्चित रूप से कई लोगों को पैसे और समय बचाने में मदद मिलेगी। प्रस्तावित प्रणाली में उड़ानों के एकत्रित ऐतिहासिक डेटा के लिए मशीन लर्निंग एल्गोरिदम लागू करके एक भविष्य कहनेवाला मॉडल बनाया जाएगा। यह प्रणाली लोगों को उन रुझानों के बारे में विचार देगी जो कीमतों का पालन करते हैं और एक अनुमानित मूल्य मूल्य भी प्रदान करते हैं जिसे वे पैसे बचाने के लिए अपनी उड़ान टिकट बुक करने से पहले संदर्भित कर सकते हैं। फ्लाइट बुकिंग कंपनियों द्वारा ग्राहकों को इस तरह की प्रणाली या सेवा प्रदान की जा सकती है जिससे ग्राहकों को अपने अनुसार टिकट बुक करने में मदद मिलेगी।

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1: PROJECT OVERVIEW

1.1 Introduction

Airline companies use complex algorithms to calculate flight prices given various conditions present at that particular time. These methods take financial, marketing, and various social factors into account to predict flight prices.

Nowadays, the number of people using flights has increased significantly. It is difficult for airlines to maintain prices since prices change dynamically due to different conditions. That's why we will try to use machine learning to solve this problem. This can help airlines by predicting what prices they can maintain. It can also help customers to predict future flight prices and plan their journey accordingly.

1.2 Aim

The aim is to predict flight prices given the various parameters. Data used in this is publicly available at Kaggle. This will be a regression problem since

the target or dependent variable is the price (continuous numeric value).

1.3 Data Used

Data was used from Kaggle which is a freely available platform for data scientists and machine learning enthusiasts.

Reference: ref. 1 in reference table

1.4 System Requirements

Windows Based Requirements:

Computers running Microsoft Windows must meet the following minimum hardware and software requirements.

Microsoft Windows: 7/8/10/11

4 GB RAM minimum, 8 GB RAM recommended

1GB of available disk space minimum

1280 * 800 minimum screen resolution

Software Requirement: Python 3.10.4

Hardware Requirement: Laptop/Computer

Internet Connectivity

CHAPTER 2: TOOLS AND TECHNOLOGY

2.1Python

Python is a high-level, interpreted, interactive, and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently whereas other languages use punctuation, and it has fewer syntactical constructions than other languages.

2.2Machine Learning

Machine learning is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention

2.3Jupyter Notebook

Jupyter Notebook is a web-based interactive computational environment for creating notebook documents. A Jupyter Notebook document is a browser-based REPL containing an ordered list of input/output cells which can contain code, text, mathematics, plots and rich media. Underneath the interface, a notebook is a JSON document, following a versioned schema, usually ending with the ".ipynb" extension.

2.4Visual Studio Code

Visual Studio Code is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

2.5Anaconda

Anaconda is a distribution of the Python and R programming languages for scientific computing that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS. We used anaconda to get access to Anaconda Navigator and Anaconda Prompt.

2.6 Numpy

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. NumPy is open-source software and has many contributors.

2.7Pandas

Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the three-clause BSD license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals.

2.8Seaborn

Seaborn is a data visualization library built on top of matplotlib and closely integrated with Pandas data structures in Python. Visualization is the central part of Seaborn which helps in exploration and understanding of data.

2.9Matplotlib

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK.

2.10 Scikit-Learn

Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

2.11 Pickle

Pickle module is used for serializing and de-serializing a Python object structure. Any object in Python can be pickled so that it can be saved on disk. What pickle does is that it “serializes” the object first before writing it to file. Pickling is a way to convert a python object into a character stream.

CHAPTER 3: DATA ANALYSIS

3.1 Data Importing

The procedure of extracting information from given raw data is called data analysis

We can import data for data training

Fig 3.1 Data Importing

3.2 Dataset variable

1**. Airline:** So this column will have all the types of airlines like Indigo, Jet Airways, Air India, and many more.

2. **Date_of_Journey:** This column will let us know about the date on which the passenger's journey will start.

3**. Source**: This column holds the name of the place from where the passenger's journey will start.

4. **Destination**: This column holds the name of the place to where passengers wanted to travel.

5. **Route**: Here we can know about that what is the route through which passengers have opted to travel from his/her source to their destination.

6. **Arrival_Time:** Arrival time is when the passenger will reach his/her destination.

7**. Duration:** Duration is the whole period that a flight will take to complete its journey from source to destination.

8. **Total_Stops**: This will let us know in how many places flights will stop there for the flight in the whole journey.

9. **Additional_Info**: In this column, we will get information about food, kind of food, and other amenities.

10**. Price**: Price of the flight for a complete journey including all the expenses before onboarding.

3.3 Reading Training Data

```
train_data = pd.read_excel("Data_Train.xlsx")
train_data.head(10)
```

Output:

	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
5	SpiceJet	24/06/2019	Kolkata	Banglore	CCU → BLR	09:00	11:25	2h 25m	non-stop	No info	3873
6	Jet Airways	12/03/2019	Banglore	New Delhi	BLR → BOM → DEL	18:55	10:25 13 Mar	15h 30m	1 stop	In-flight meal not included	11087
7	Jet Airways	01/03/2019	Banglore	New Delhi	BLR → BOM → DEL	08:00	05:05 02 Mar	21h 5m	1 stop	No info	22270
8	Jet Airways	12/03/2019	Banglore	New Delhi	BLR → BOM → DEL	08:55	10:25 13 Mar	25h 30m	1 stop	In-flight meal not included	11087
9	Multiple carriers	27/05/2019	Delhi	Cochin	DEL → BOM → COK	11:25	19:15	7h 50m	1 stop	No info	8625

Fig. 3.3 Reading Data

3.4 Exploratory Data Analysis

Now here we will be looking at the kind of columns our dataset has.

```
train_data.columns
```

Output:

```
Index(['Airline', 'Date_of_Journey', 'Source', 'Destination', 'Route',
       'Dep_Time', 'Arrival_Time', 'Duration', 'Total_Stops',
       'Additional_Info', 'Price'],
      dtype='object')
```

Here we can get more information about our dataset

```
train_data.info()
```

Output:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10683 entries, 0 to 10682
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Airline          10683 non-null  object
 1   Date_of_Journey  10683 non-null  object
 2   Source           10683 non-null  object
 3   Destination      10683 non-null  object
 4   Route            10682 non-null  object
 5   Dep_Time         10683 non-null  object
 6   Arrival_Time     10683 non-null  object
 7   Duration         10683 non-null  object
 8   Total_Stops      10682 non-null  object
 9   Additional_Info  10683 non-null  object
 10  Price            10683 non-null  int64
dtypes: int64(1), object(10)
memory usage: 918.2+ KB
```

Fig. 3.4.1 Information Of Dataset

To know more about the dataset

```
train_data.describe()
```

Output:

	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

Fig. 3.4.2 Statistics Of Dataset

Now while using the IsNull function we will gonna see the number of null values in our dataset

```
train_data.isnull().head()
```

Output:

Airline	Source	Destination	Route	Duration	Total_Stops	Additional_Info	Price	Journey_day	Journey_month	Dep_hour	Dep_min	Arrival_hour	Arrival_min
False	False	False	False	False	False	False	False	False	False	False	False	False	False
False	False	False	False	False	False	False	False	False	False	False	False	False	False
False	False	False	False	False	False	False	False	False	False	False	False	False	False
False	False	False	False	False	False	False	False	False	False	False	False	False	False
False	False	False	False	False	False	False	False	False	False	False	False	False	False

Fig. 3.4.3 NAN Values

Now while using the IsNull function and sum function we will gonna see the number of null values in our dataset

```
train_data.isnull().sum()
```

Output:

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

Dropping NAN values

```
train_data.dropna(inplace = True)
```

Checking the Additional_info column and having the count of unique types of values.

```
train_data["Additional_Info"].value_counts()
```

Output:

```
No info                         8182
In-flight meal not included     1926
No check-in baggage included     318
1 Long layover                    19
Change airports                    7
Business class                     4
No Info                            3
1 Short layover                    1
2 Long layover                     1
Red-eye flight                     1
Name: Additional_Info, dtype: int64
```

Now let's look at our testing dataset

```
test_data = pd.read_excel("Test_set.xlsx")
test_data.head(10)
```

Output:

	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
5	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
6	Air India	12/03/2019	Banglore	New Delhi	BLR → TRV → DEL	07:30	22:35	15h 5m	1 stop	No info
7	IndiGo	1/05/2019	Kolkata	Banglore	CCU → HYD → BLR	15:15	20:30	5h 15m	1 stop	No info
8	IndiGo	15/03/2019	Kolkata	Banglore	CCU → BLR	10:10	12:55	2h 45m	non-stop	No info
9	Jet Airways	18/05/2019	Kolkata	Banglore	CCU → BOM → BLR	16:30	22:35	6h 5m	1 stop	No info

Fig. 3.4.4 Test_set Dataset Information

Now here we will be looking at the kind of columns our testing data has.

```
test_data.columns
```

Output:

```
Index(['Airline', 'Date_of_Journey', 'Source', 'Destination', 'Route',
       'Dep_Time', 'Arrival_Time', 'Duration', 'Total_Stops',
       'Additional_Info'],
      dtype='object')
```

Information about the dataset

```
test_data.info()
```

Output:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2671 entries, 0 to 2670
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   Airline          2671 non-null   object
 1   Date_of_Journey  2671 non-null   object
 2   Source           2671 non-null   object
 3   Destination      2671 non-null   object
 4   Route            2671 non-null   object
 5   Dep_Time         2671 non-null   object
 6   Arrival_Time     2671 non-null   object
 7   Duration         2671 non-null   object
 8   Total_Stops      2671 non-null   object
 9   Additional_Info  2671 non-null   object
dtypes: object(10)
memory usage: 208.8+ KB
```

Fig. 3.4.5 Test_data Information

To know more about the testing dataset

```
test_data.describe()
```

Output:

	Airline	Date_of_Journey	Source	Destination	Route	Dep_Time	Arrival_Time	Duration	Total_Stops	Additional_Info
count	2671	2671	2671	2671	2671	2671	2671	2671	2671	2671
unique	11	44	5	6	100	199	704	320	5	6
top	Jet Airways	9/05/2019	Delhi	Cochin	DEL → BOM → COK	10:00	19:00	2h 50m	1 stop	No info
freq	897	144	1145	1145	624	62	113	122	1431	2148

Fig. 3.4.6 Statistic Information of Test_data

Now while using the IsNull function and sum function we will gonna see the number of null values in our testing data

```
test_data.isnull().sum()
```

Output:

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

CHAPTER 4: DATA VISUALIZATION

4.1 Plotting Price vs Airline Plot:

```
catplot(y = "Price", x = "Airline", data = train_data.sort_values("Price", ascending = False), kind="boxen",
height = 6, aspect = 3)

plt.show()
```

Output:

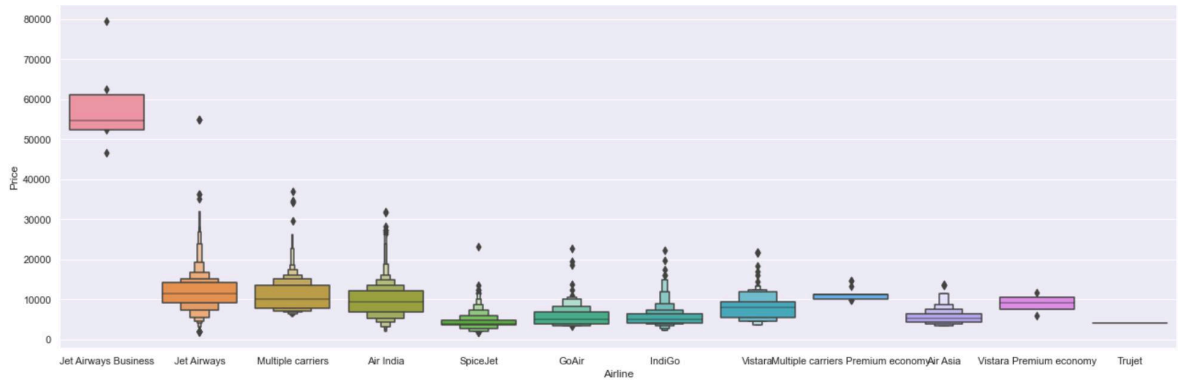

Fig. 4.1.1 Plotting Price vs Airline Plot

Inference: Here with the help of the cat plot we are trying to plot the boxplot between the price of the flight and airline and we can conclude that **Jet Airways has the most outliers in terms of price**.

Let's see our processed data first

```
train_data.head()
```

Output:

	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

Fig. 4.1.2 Updated Test_data Head

4.2 Time & Date Setting:

Here first we are dividing the features and labels and then converting the hours in minutes.

```
In [19]: # Time taken by plane to reach destination is called Duration
         # It is the differnce betwwen Departure Time and Arrival time


         # Assigning and converting Duration column into list
         duration = list(train_data["Duration"])

         for i in range(len(duration)):
             if len(duration[i].split()) != 2:    # Check if duration contains only hour or mins
                 if "h" in duration[i]:
                     duration[i] = duration[i].strip() + " 0m"   # Adds 0 minute
                 else:
                     duration[i] = "0h " + duration[i]           # Adds 0 hour

         duration_hours = []
         duration_mins = []
         for i in range(len(duration)):
             duration_hours.append(int(duration[i].split(sep = "h")[0]))    # Extract hours from duration
             duration_mins.append(int(duration[i].split(sep = "m")[0].split()[-1]))   # Ext
```

Fig. 4.2.1 Time Conversion

Date_of_Journey: Here we are organizing the format of the date of journey in our dataset for better preprocessing in the model stage.

```
train_data["Journey_day"] = pd.to_datetime(train_data.Date_of_Journey, format="%d/%m/%Y").dt.day

train_data["Journey_month"] = pd.to_datetime(train_data["Date_of_Journey"], format
="%d/%m/%Y").dt.month

train_data.drop(["Date_of_Journey"], axis = 1, inplace = True)
```

Dep_Time: Here we are converting departure time into hours and minutes

```
train_data["Dep_hour"] = pd.to_datetime(train_data["Dep_Time"]).dt.hour

train_data["Dep_min"] = pd.to_datetime(train_data["Dep_Time"]).dt.minute

train_df.drop(["Dep_Time"], axis = 1, inplace = True)
```

Arrival_Time: Similarly we are converting the arrival time into hours and minute

```
train_data["Arrival_hour"] = pd.to_datetime(train_data.Arrival_Time).dt.hour

train_data["Arrival_min"] = pd.to_datetime(train_data.Arrival_Time).dt.minute

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

Now after final preprocessing let's see our dataset

```
train_data.head()
```

Output:

Airline	Source	Destination	Route	Duration	Total_Stops	Additional_Info	Price	Journey_day	Journey_month	Dep_hour	Dep_min	Arrival_hour	Arrival_min
IndiGo	Banglore	New Delhi	BLR → DEL	170	non-stop	No info	3897	24	3	22	20	1	10
Air India	Kolkata	Banglore	CCU → IXR → BBI → BLR	445	2 stops	No info	7662	1	5	5	50	13	15
Jet Airways	Delhi	Cochin	DEL → LKO → BOM → COK	1140	2 stops	No info	13882	9	6	9	25	4	25
IndiGo	Kolkata	Banglore	CCU → NAG → BLR	325	1 stop	No info	6218	12	5	18	5	23	30
IndiGo	Banglore	New Delhi	BLR → NAG → DEL	285	1 stop	No info	13302	1	3	16	50	21	35

Fig. 4.2.2 Updated train_data

4.3 Correlation Between All Features:

Plotting Correlation

```
plt.figure(figsize = (18,18))
sns.heatmap(train_data.corr(), annot = True, cmap = "RdYlGn")
plt.show()
```

Output:

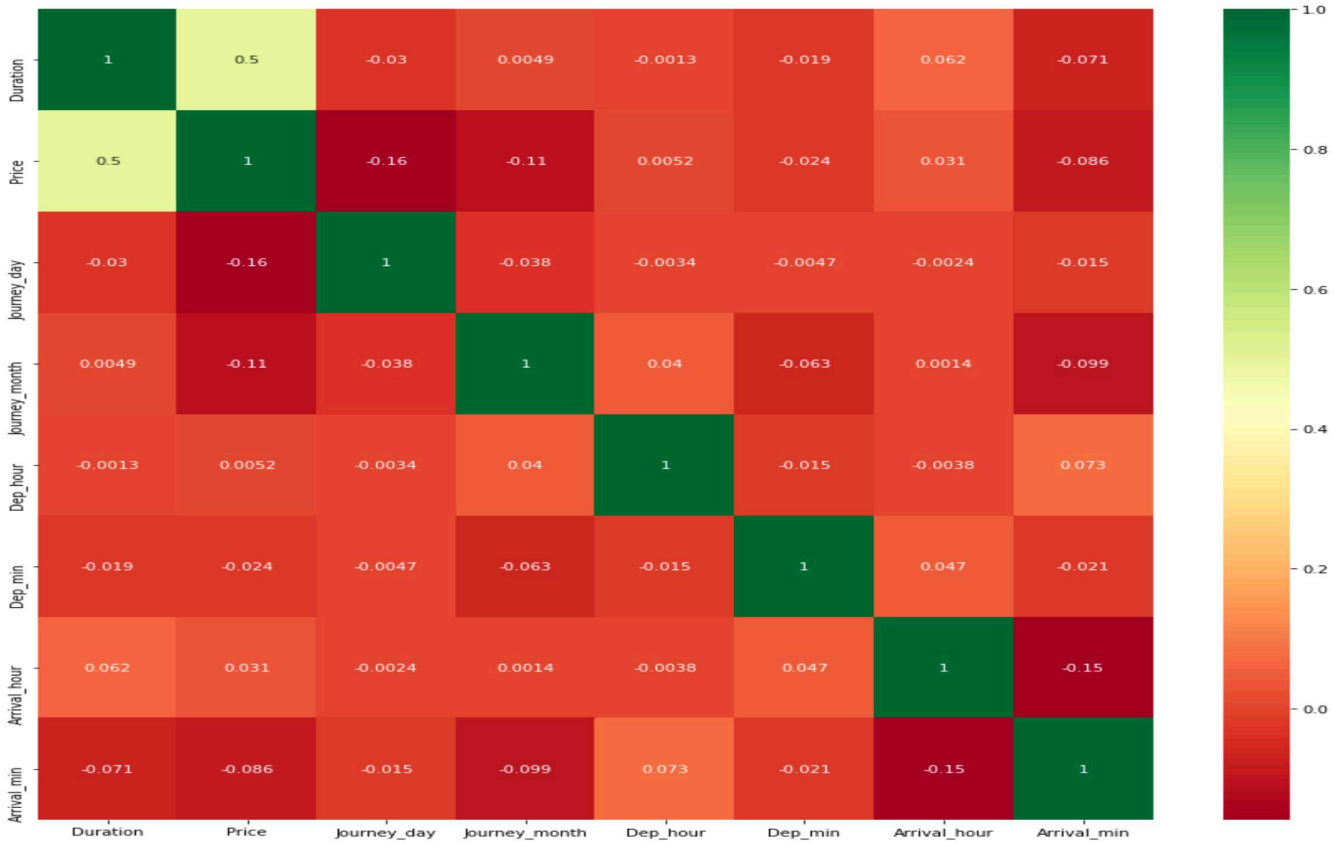

Fig. 4.3 Correlation Between all Features

4.4 Fitting Model Using Random Forest:

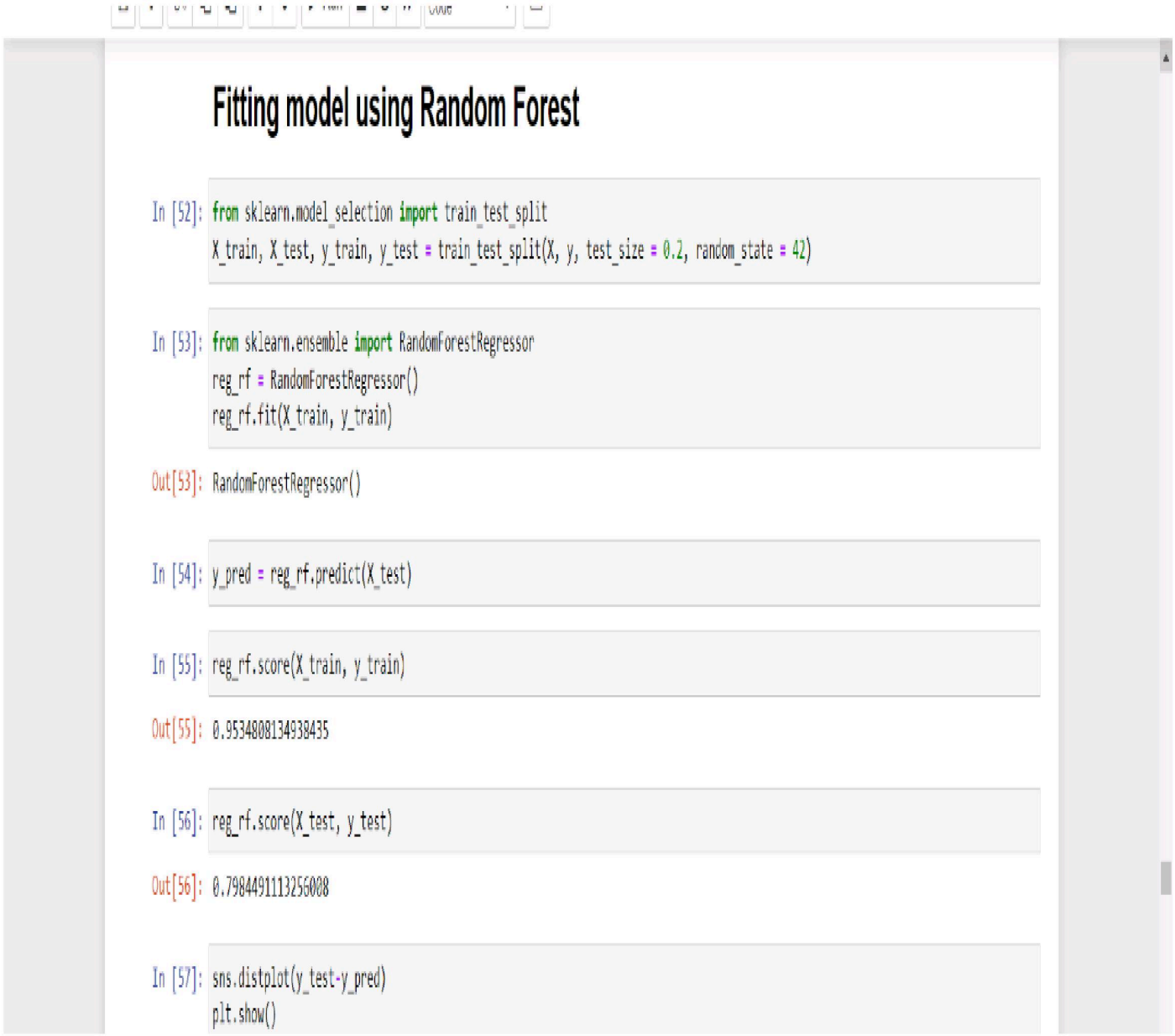

Fig. 4.4 Fitting Model Using Random Forest

Dropping the Price column as it is of no use

```
data = train_data.drop(["Price"], axis=1
```

CHAPTER 5: CONCLUSION

5.1 Conclusion:

In the proposed paper the overall survey for the dynamic price changes in the flight tickets is presented. this gives the information about the highs and lows in the airfares according to the days, weekend and time of the day that is morning, evening and night. also the machine learning models in the computational intelligence feild that are evaluated before on different datasets are studied. their accuracy and performances are evaluated and compared in order to get better result. For the prediction of the ticket prices perfectly differnt prediction models are tested for the better prediction accuracy. As the pricing models of the company are developed in order to maximize the revenue management. So to get result with maximum accuracy regression analysis is used. From the studies , the feature that influences the prices of the ticket are to be considered. In future the details about number of availble seats can improve the performance of the model.

References

1. Dataset Reference: https://www.kaggle.com/nikhilmittal/flight-fare-prediction-mh

2. https://www.geeksforgeeks.org/short-note-on-data-visualization/

3. https://www.youtube.com/playlist?list=PL_1pt6K-CLoDMEbYy2PcZuITWEjqMfyoA.

4. William Groves and Maria Gini "An agent for optimizing airline ticket purchasing" in proceedings of the 2013 international conference on autonomous agents and multi-agent systems.