A WEB APP FOR PREDICTION OF AIRLINE FARE

**A PROJECT REPORT**

###### ***Submitted by***

##### Naman Talwar (20BCE10184)

**Chandni Soni** (20BCE10224)

**Jenil Savalia** (20BCE10183)

**Pranjal Singhal** (20BCE10238)

*in partial fulfillment for the award of the degree*

*of*

**BACHELOR OF TECHNOLOGY**

*in*

# COMPUTER SCIENCE AND ENGINEERING

****

**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING**

**VIT BHOPAL UNIVERSITY**

**KOTHRIKALAN, SEHORE**

**MADHYA PRADESH - 466114**

##### DEC 2021

**VIT BHOPAL UNIVERSITY, KOTHRIKALAN, SEHORE**

**MADHYA PRADESH – 466114**

**BONAFIDE CERTIFICATE**

Certified that this project report titled **“A Web-App for prediction of airline fare”** is the bonafide work of “**NAMAN TALWAR (Register No: 20BCE10184), CHANDNI SONI (Register No.: 20BCE10224), JENIL SAVALIA (Register No.: 20BCE10183), PRANJAL SINGHAL (Register No.: 20BCE10238)”** who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported at this time does not form part of any other project/research work based on which a degree or award was conferred on an earlier occasion on this or any other candidate.

**PROGRAM CHAIR PROJECT GUIDE**

DR. SANDIP MAL, Program Chair BCE

Prof. Anand Motwani, Project Guide

School of Computer Science and Engineering

VIT BHOPAL UNIVERSITY

School of Computer Science and Engineering

VIT BHOPAL UNIVERSITY

The Project Exhibition I Examination is held on 22- DEC-2021

**ACKNOWLEDGEMENT**

First and foremost, I would like to thank the Lord Almighty for His presence and immense blessings throughout the project work.

I wish to express my heartfelt gratitude to **Dr. S. Poonkuntran** Head of the Department, School of Computer Science and Engineering for much of his valuable support encouragement in carrying out this work.

I would like to thank my internal guide, **Dr. Anand Motwani** Sir, for continually guiding and actively participating in my project, giving valuable suggestions to complete the project work.

I would like to thank all the technical and teaching staff of the School of Computer Science and Engineering, who extended directly or indirectly all support.

Last, but not least, I am deeply indebted to my parents who have been the greatest support while I worked day and night for the project to make it a success.

**LIST OF ABBREVIATIONS**

**Libraries:-** A Python library is a collection of related modules. It contains bundles of code that can be used repeatedly in different programs. It makes Python Programming simpler and convenient for the programmer.

**ML:-** ML stands for Machine Learning. Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so.

**EDA:-** EDA stands for Exploratory Data Analysis. It is an approach of analyzing data sets to summarize their main characteristics, often using statistical graphics and other data visualization methods.

**Web-App:-** It is a software program that provides interactive functionality and is accessed through a web browser and URL.

**ATF:-** It stands for Air Turbine Fuel. It is used by Airlines as fuel.

**R2 score:-** It is the proportion of the variance in the dependent variable that is predictable from the independent variable(s).

**LIST OF FIGURES AND GRAPHS**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO.**  1.  2.  3.  4. | **TITLE**  System Architecture Diagram  Year vs ATF imported by India.  ATF imported vs Fuel expenses per passenger.  ATF imported vs Fuel Prices | **PAGE NO.**  18  28  29  29 |

**Abstract**

In the project “Prediction of Airline Fares”, we take a dataset that contains the information of the airline name, source, destination, date and time of arrival and departure, and the number of stops.

On this data, we apply different models to predict the fares of the flight in accordance with the above-mentioned variables.

The models we are using are namely, Linear Regression, Random Forest, Decision Tree, Support Vector Regression, Lasso Regression.

We compare these models and pick the one which suits the most (highest r2 score), the one being **Random Forest**. The model is then saved using the library Pickle and loaded in the code for the web app using the library Flask.

Now, after training this model, we deploy the model in the form of a web that accepts inputs and provides us with suitable output. The web app is designed and put together using HTML and CSS.

In this report, we dive deeper into our project and find out about the applications, shortcomings, methodology, and structural architecture.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | List of Abbreviations  List of Figures and Graphs  Abstract | 4  5  6 |
| 1 | **CHAPTER-1:**  **PROJECT DESCRIPTION AND OUTLINE**   * 1. Introduction   2. Motivation for the project   3. Problem Statement   4. Objective of the work   5. Organization of the project   6. Summary | 10 |
| 2 | **CHAPTER-2:**  **RELATED WORK INVESTIGATION**  2.1 Introduction  2.2 Core idea of the project  2.3 Existing Approaches  2.5 Observations from the investigation  2.6 Summary | 12  .  .  . |
| 3 | **CHAPTER-3:**  **REQUIREMENT ARTIFACTS**  3.1 Introduction  3.2 Hardware and Software requirements  3.3 Specific Project requirements  3.3.1 Data requirement  3.3.2 Functions requirement  3.3.3 Performance and security requirement  3.3.4 Look and Feel Requirements  3.4 Summary | 14 |
| 4 | **CHAPTER-4:**  **DESIGN METHODOLOGY AND ITS NOVELTY**  4.1 Methodology and goal  4.2 Functional modules design and analysis  4.3 Software Architectural designs  4.4 Subsystem services  4.5 User Interface designs  4.6 Summary | 16 |
| 5 | **CHAPTER-5:**  **TECHNICAL IMPLEMENTATION & ANALYSIS**  5.1Outline  5.2 Technical coding and code solutions  5.3 Working Layout of Forms  5.4 Test and validation  5.5 Performance Analysis (Graphs/Charts)  5.6 Summary | 20 |
| 6 | **CHAPTER-6:**  **PROJECT OUTCOME AND APPLICABILITY**  . 6.1Outline  6.2 Key implementations outlines of the project  6.3 Significant project outcomes  6.4 Project applicability on Real-world applications | 28 |
| 7 | **CHAPTER-7:**  **CONCLUSIONS AND RECOMMENDATION**  7.1Outline  7.2 Limitation of the project  7.3 Future Enhancements | 32 |
|  | References | 33 |

**CHAPTER - 1**

**PROJECT DESCRIPTION AND OUTLINE**

**Introduction**

It will be a Machine Learning model in the form of a Web App through which we will be able to predict the airline fares based on a given Dataset and Input. The airline may increase the prices when the demand is to be expected to increase the capacity. 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. This paper gives the machine learning regression methods to predict the prices at the given time.

**Problem Statement**

Research Analysis- Comparing multiple machine learning algorithms using the dataset for the prediction of airline fares.

We will also be analyzing the relationship between the amount of fuel imported by India and flight fare using two separate datasets.

**Organization of the project**

1. Import datasets and merge them.
2. **Data cleaning**- refers to identifying and correcting errors in the dataset that may negatively impact a predictive model. Data cleaning is used to refer to all kinds of tasks and activities to detect and repair errors in the data.
3. **Data Processing**- is the task of converting data from a given form to a much more usable and desired form. Using Machine Learning algorithms, mathematical modeling, and statistical knowledge, this entire process can be automated.
4. **Data visualization**- is defined as a graphical representation that contains the information and the data. By using visual elements like charts, graphs, and maps, data visualization techniques provide an accessible way to see and understand trends, outliers, and patterns in data.
5. Will implement four machine learning algorithms and compare them.
6. Finding out the best algorithm
7. Will make web-app.

**Summary**

This model can help airlines by predicting what prices they can maintain and which prices they need to update. It can also help customers to predict future flight prices and plan their journey accordingly. Construct and train the ML Model so that it can distinguish between expensive and cheap tickets.

**CHAPTER – 2**

**RELATED WORK INVESTIGATION**

**The core idea of the project**:

To maximize the utility of the prediction of the Airline fares.

**Existing Approaches:**

There are many apps in which different kinds of ML algorithms are used so the problem with this kind of apps is there are various kinds of variations in the prices.

The user gets confused about when to buy the ticket and from which source would he avail it in an easier way.

**Literature Review:**

The main idea of our project was to predict the lowest possible price of airline tickets. We agree there may be many who might have done the same task but we just gave it a try and implemented our machine learning knowledge. We worked on over two datasets and were successfully combining them. It is very difficult for the customer to purchase a flight ticket at the minimum price. For this several techniques are used to obtain the day at which the price of an air ticket will be minimum. Most of these techniques are using sophisticated artificial intelligence research is known as Machine Learning.

Utilizing AI models to acquire the greatest presentation to get the least cost of aircraft ticket buying, having 75.3% precision.

**CHAPTER – 3**

**REQUIREMENT ARTIFACTS**

**Hardware and Software requirements;**

**Logistic Regression:**

It is used in statistical software to understand the relationship between the dependent variable and one or more independent variables by estimating probabilities using a logistic regression equation.

**Random Forest:**

It uses bagging and feature randomness when building each individual tree to try to create an uncorrelated forest of trees whose prediction by committee is more accurate than that of any individual tree

**Decision Tree:**

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression.

The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features

**SVR Regression:**

Support Vector Regression is a supervised learning algorithm that is used to predict discrete values. Support Vector Regression uses the same principle as the SVMs.

**Lasso Regression:**

Lasso regression is a regularization technique. It is used over regression methods for a more accurate prediction. This model uses shrinkage. Shrinkage is where data values are shrunk towards a central point as the mean

**Specific Project Requirement**

* Programming language- Python 4.0
* IDE- Jupyter Notebook, Google Colab, Visual Studio Code
* Dataset from Kaggle and other Sources
* Python libraries- NumPy, Pandas, Matplotlib, Seaborn, Sckit-learn, Flask
* Machine learning algorithms
* HTML, CSS.

**Summary**

This project provides us the information about the highs and lows in the airfares according to the timings of the day that is morning, evening, and night.

As the pricing models of the company are developed, in such a way so that the company maximizes its revenues. So, to get results with maximum accuracy regression analysis is used.

**CHAPTER – 4**

**DESIGN METHODOLOGY AND ITS NOVELTY**

**Methodology and Architecture:**

1. In the analysis part of the project, we find out the change in airfare corresponding to the ATF (Airline Turbine Fuel) Prices and the amount ATF imported in India in recent years. For this, we merge two datasets – one with the amount of Petroproducts imported and the second which gives us the data for how much the airline fares cost over the years.
2. We import another dataset that provides us with the schedule and prices of singular flight tickets.
3. We categorize and pre-process the data.
4. We divide this data into test data and train data.
5. After this, feature selection is done on the data, and the variable which affects the price the most is found (Number of Stops).
6. We then introduce the machine learning algorithms and check which one suits our model the most, which turns out to be Random Forest.
7. After finding the most suitable algorithm, we perform Hyperparameter Tuning and train the model.
8. We check for the mean absolute, mean squared, and root means squared error.
9. After this, we save this model in the form of a pickle file to be accessed in our Web App implementation.
10. We load the file and the model using pickle and flask into the Web App.

* Pickle: Python pickle module is used for serializing and de-serializing a Python object structure. Any object in Python can be picked 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. The idea is that this character stream contains all the information necessary to reconstruct the object in another python script.

* Flask: Flask is a popular Python web framework, meaning it is a third-party Python library used for developing web applications.

1. We can then implement the Web App.

**Functional modules design and analysis:**

The project can be divided into separate parts:

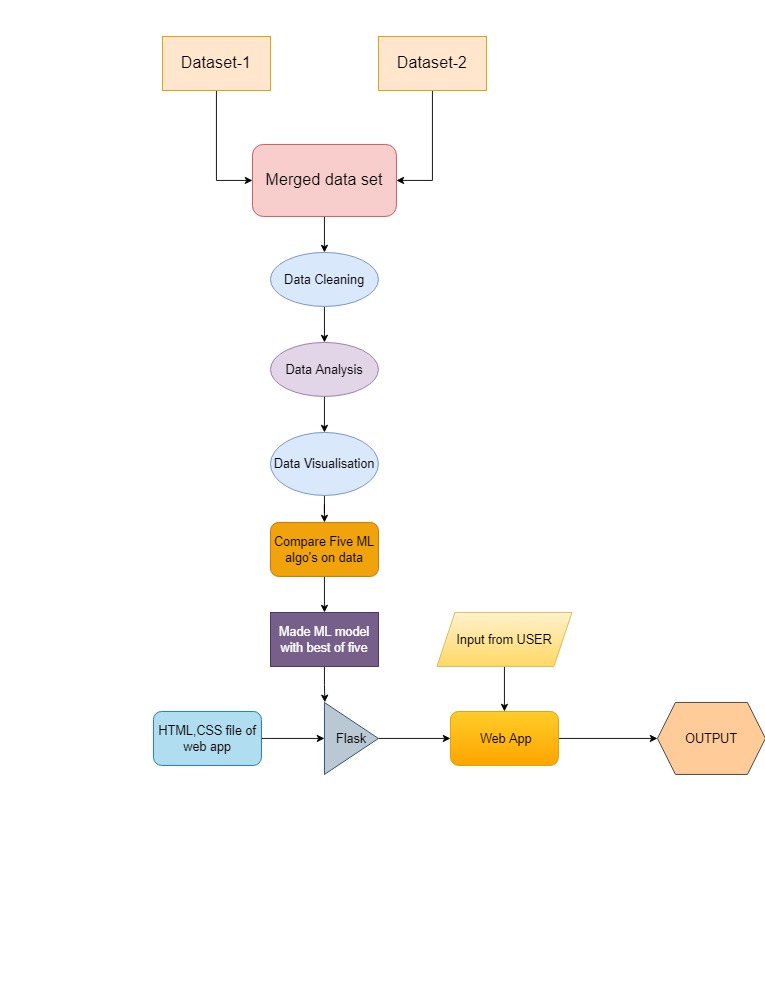
1. Analysis 1 - Analysis of the fuel prices with the amount of ATF imported after merging 2 datasets.
2. Analysis 2 – Analysis of the prices with the airline, number of stops, departure date, arrival date, source, and destination.
3. Testing – Testing five different models namely:

* Linear Regression
* Random Forest
* Decision Tree
* Support Vector Regression
* Lasso Regression

And finding out which is best suited for our data.

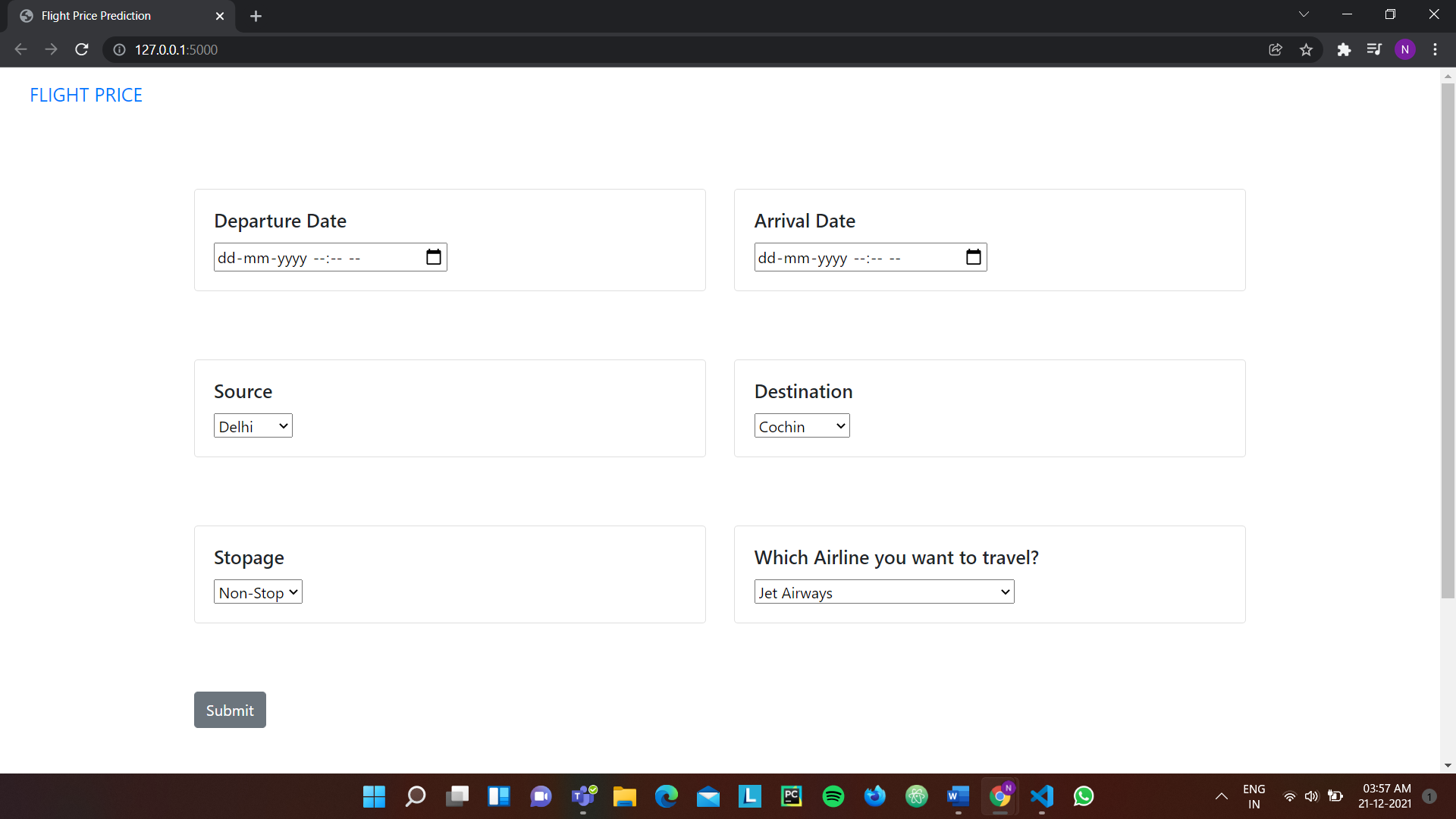
1. Tuning – Hyperparameter Tuning of the suitable model.
2. Implementation – Training the model, saving it to link it into the web app.
3. Deployment – Making the web app and running the model through it.

**System Architecture Diagram: -**



**Fig-1**

**User Interface Design:**



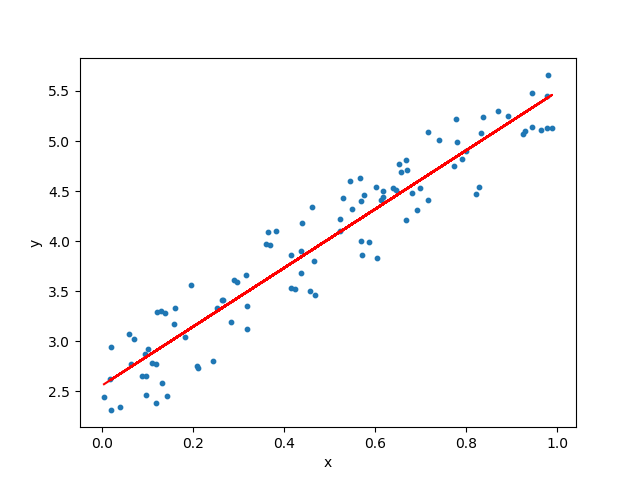
In this interface, we fill in the given fields and get the unique output for the specific case.

**Chapter – 5**

**TECHNICAL IMPLEMENTATION & ANALYSIS**

1. **Linear Regression**: Regression is a method of modelling a target value based on predictors that are independent. It is mostly based on the number of independent variables and the relationship between independent and dependent variables. linear regression is a type of analysis where the number of independent variables is one and the relationship between the dependent and independent variables vary linearly. The important concept to understand linear regressions are cost function and Gradient decent.

**y(pred) = b0+b1 x (4)**



**CODE:-**

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)

from sklearn import linear\_model

reg = linear\_model.LinearRegression()

reg.fit(X\_train, y\_train)

print("Coefficients: ",reg.coef\_)

print("Variance Score: ", reg.score(X\_test, y\_test))

y\_pred = reg.predict(X\_test)

reg.score(X\_train,y\_train)

reg.score(X\_test,y\_test)

from sklearn import metrics

print("Mean Absolute Error: ", metrics.mean\_absolute\_error(y\_test,y\_pred))

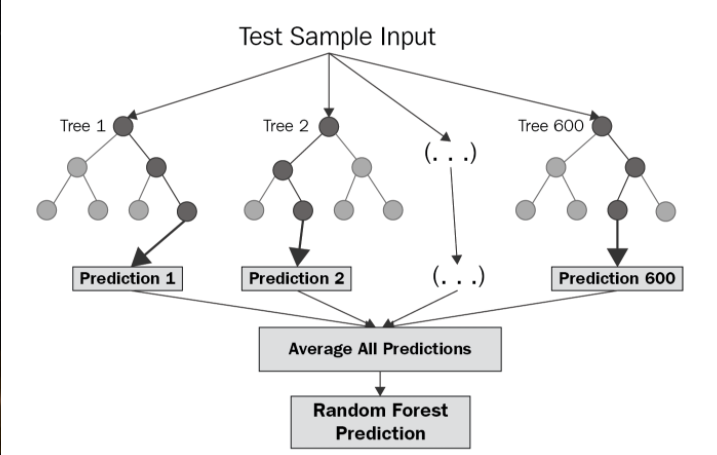
print("Mean Squared Error: ", metrics.mean\_squared\_error(y\_test,y\_pred))

print("Root Mean Squared Error: ", np.sqrt(metrics.mean\_squared\_error(y\_test,y\_pred)))

print("Score: ",metrics.r2\_score(y\_test,y\_pred))

1. **Random Forest:** It is a supervised learning algorithm. The benefit of the random forest is, it very well may be utilized for both characterization and relapse issue which structure most of current machine learning framework. Random forest forms numerous decision trees, whats more, adds them together to get an increasingly exact and stable expectation. Random Forest has nearly the equivalent parameters as a decision tree or a stowing classifier model. It is very simple to discover the significance of each element on the expectation when contrasted with others in this calculation.

The regular component in these techniques is, for the kth tree, a random vector theta k is produced, autonomous of the past random vectors theta 1, … , theta k-1 however with the equivalent distribution, while a tree is developed utilizing the preparation set and bringing about a classifier. x is an information vector. For a period, in stowing the random vector is created as the includes in N boxes where N is the number of models in the preparation set of information. In random split, choice includes various autonomous random whole numbers between 1 to K. The dimensionality and nature of theata rely upon its utilization in the development of a tree. After countless trees are created, they select the most famous class. These methodologies are called as random forests.



**CODE:**

from sklearn.ensemble import RandomForestRegressor

reg\_rf = RandomForestRegressor()

reg\_rf.fit(X\_train,y\_train)

y1\_pred = reg\_rf.predict(X\_test)

reg\_rf.score(X\_train,y\_train)

reg\_rf.score(X\_test,y\_test)

from sklearn import metrics

print("Mean Absolute Error: ", metrics.mean\_absolute\_error(y\_test,y1\_pred))

print("Mean Squared Error: ",metrics.mean\_squared\_error(y\_test,y1\_pred))

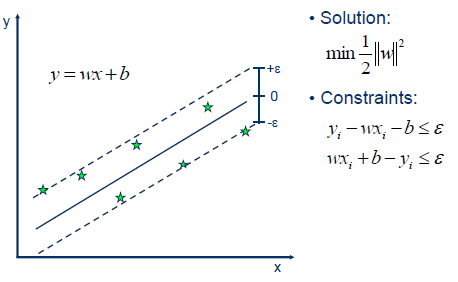
print("Root Mean Squared Error: ", np.sqrt(metrics.mean\_squared\_error(y\_test,y1\_pred)))

print("Score: ",metrics.r2\_score(y\_test,y1\_pred))

1. **Support Vector Regression:**

In the proposed paper Support Vector Machine used as regression analysis that relays on kernel function considered as non-parametric technique. The following kernels are used: Linear, Polynomial, Radial Basis Function.

As per the previous studies Random Forest and the gradient boosting gives the maximum accuracy.



from sklearn.svm import SVR

from sklearn.pipeline import make\_pipeline

from sklearn.preprocessing import StandardScaler

reg\_svr = make\_pipeline(StandardScaler(), SVR(C= 1, epsilon = 0.2))

reg\_svr.fit(X\_train,y\_train)

y3\_pred = reg\_svr.predict(X\_test)

reg\_svr.score(X\_train,y\_train)

reg\_svr.score(X\_test,y\_test)

from sklearn import metrics

print("Mean Absolute Error: ",metrics.mean\_absolute\_error(y\_test,y3\_pred))

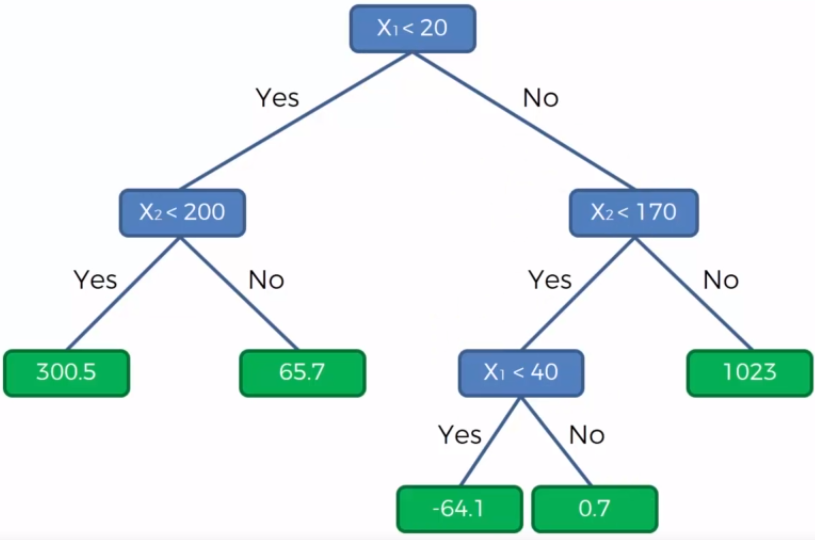
print("Mean Squared Error: ",metrics.mean\_squared\_error(y\_test,y3\_pred))

print("Root Mean Squared Error: ", np.sqrt(metrics.mean\_squared\_error(y\_test,y3\_pred)))

print("Score: ", metrics.r2\_score(y\_test,y3\_pred))

1. **Decision Tree:**

The Decision tree calculation separates the informational collection into small subsets, at a similar same time it creates gradually. The last outcomes are the tree with the decision nodes, what’s more, the leaf nodes. A decision hub may have at least two branches. In the beginning, consider the entire informational collection as root. Highlight esteems are wanted to be downright. On the off chance that the qualities are constant then they are discretized before structure the model. Based on characteristic qualities records are dispersed recursively. There are two primary characteristics in the decision tree calculation. One is Information Gain and another is the Gini index. Information Gain is the proportion of Change in entropy. Higher the entropy more the instructive substance, where the entropy is a proportion of vulnerability of arbitrary variable. Gini Index is a component that measures how frequently an arbitrarily picked. Component would be mistakenly distinguished. It implies a characteristic with a lower Gini index ought to be liked.



**CODE:**

from sklearn.tree import DecisionTreeRegressor

reg\_dt = DecisionTreeRegressor()

reg\_dt.fit(X\_train,y\_train)

y2\_pred = reg\_dt.predict(X\_test)

reg\_dt.score(X\_train,y\_train)

reg\_dt.score(X\_test,y\_test)

from sklearn import metrics

print("Mean Absolute Error: ",metrics.mean\_absolute\_error(y\_test,y2\_pred))

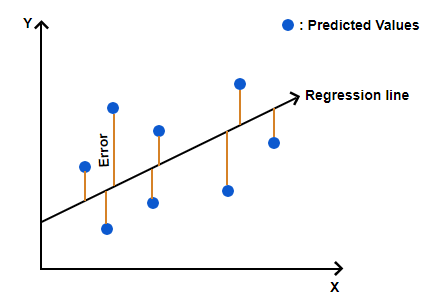
print("Mean Squared Error: ",metrics.mean\_squared\_error(y\_test,y2\_pred))

print("Root Mean Squared Error: ",np.sqrt(metrics.mean\_squared\_error(y\_test,y2\_pred)))

print("Score: ",metrics.r2\_score(y\_test,y2\_pred))

1. **Lasso Regression: Lasso regression** is a type of linear regressionthat uses [shrinkage](https://www.statisticshowto.com/shrinkage-estimator/). Shrinkage is where data values are shrunk towards a central point, like the [mean](https://www.statisticshowto.com/mean/). The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters). This particular type of regression is well-suited for models showing high levels of multicollinearity or when you want to automate certain parts of model selection, like variable selection/parameter elimination.

The acronym “LASSO” stands for **L**east **A**bsolute **S**hrinkage and **S**election **O**perator.



**CODE:**

from sklearn.tree import DecisionTreeRegressor

reg\_dt= DecisionTreeRegressor()

reg\_dt.fit(X\_train,y\_train)

y2\_pred=reg\_dt.predict(X\_test)

reg\_dt.score(X\_train,y\_train)

reg\_dt.score(X\_test,y\_test)

from sklearn import metrics

print("Mean Absolute Error: ",metrics.mean\_absolute\_error(y\_test,y2\_pred))

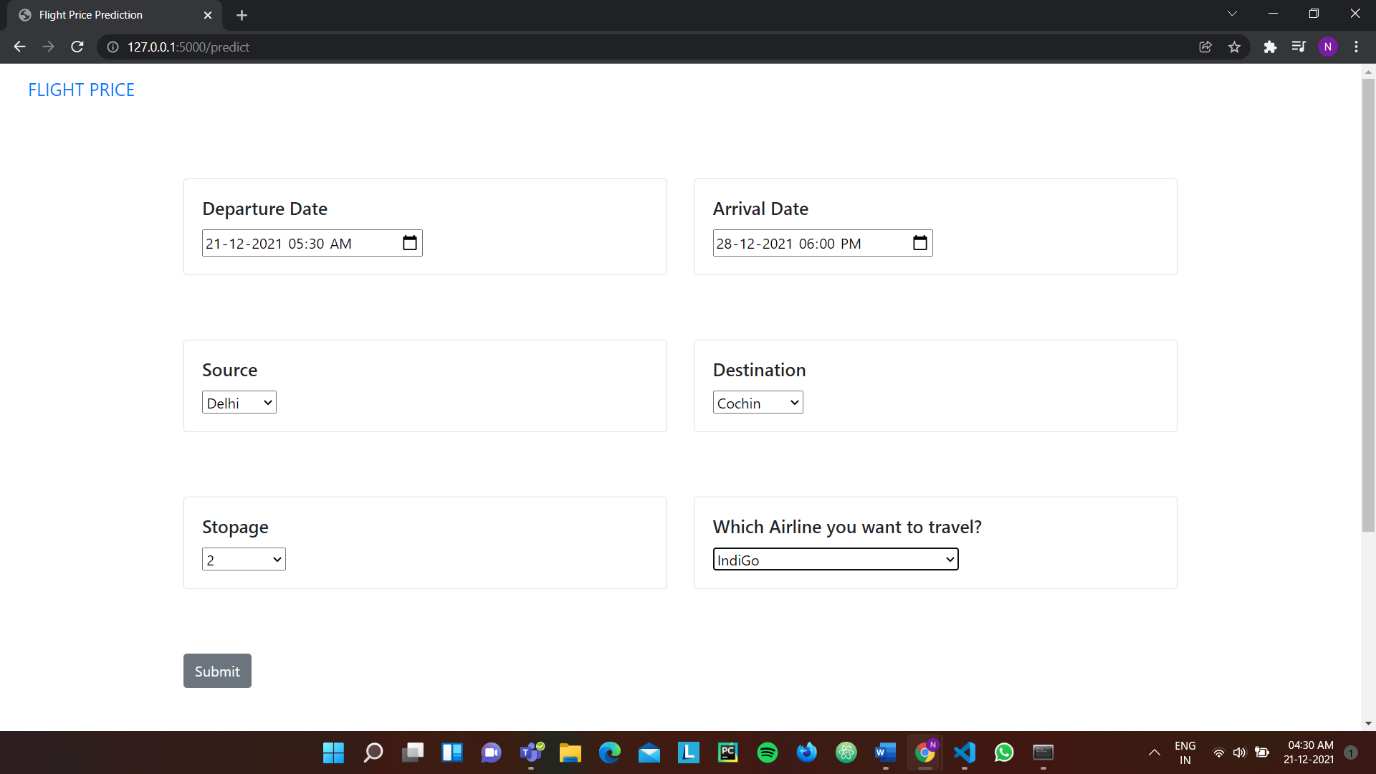
print("Mean Squared Error: ",metrics.mean\_squared\_error(y\_test,y2\_pred))

print("Root Mean Squared Error: ",np.sqrt(metrics.mean\_squared\_error(y\_test,y2\_pred)))

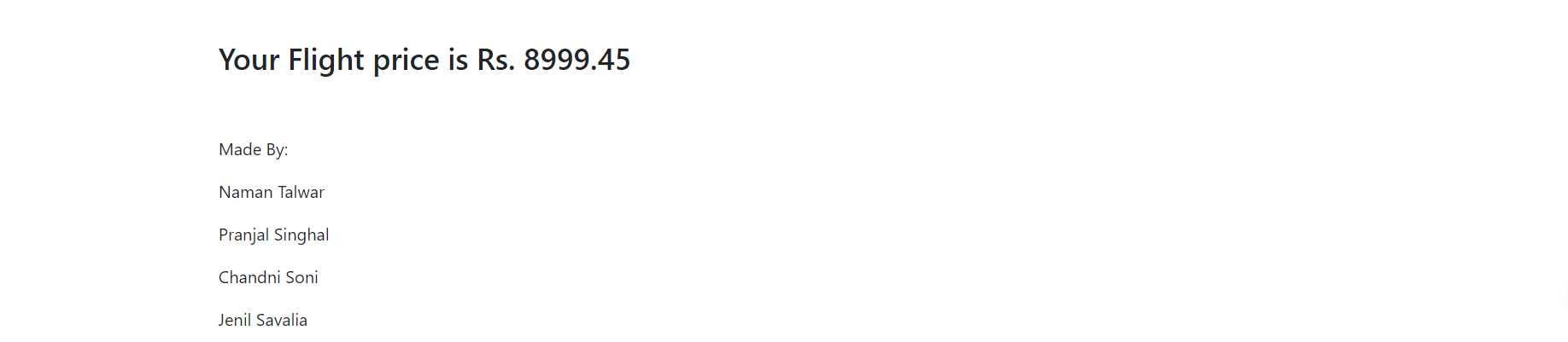
print("Score: ",metrics.r2\_score(y\_test,y2\_pred))

**Test** –

**Input:**



**Output:**



**CHAPTER – 6**

**PROJECT OUTCOME AND APPLICABILITY**

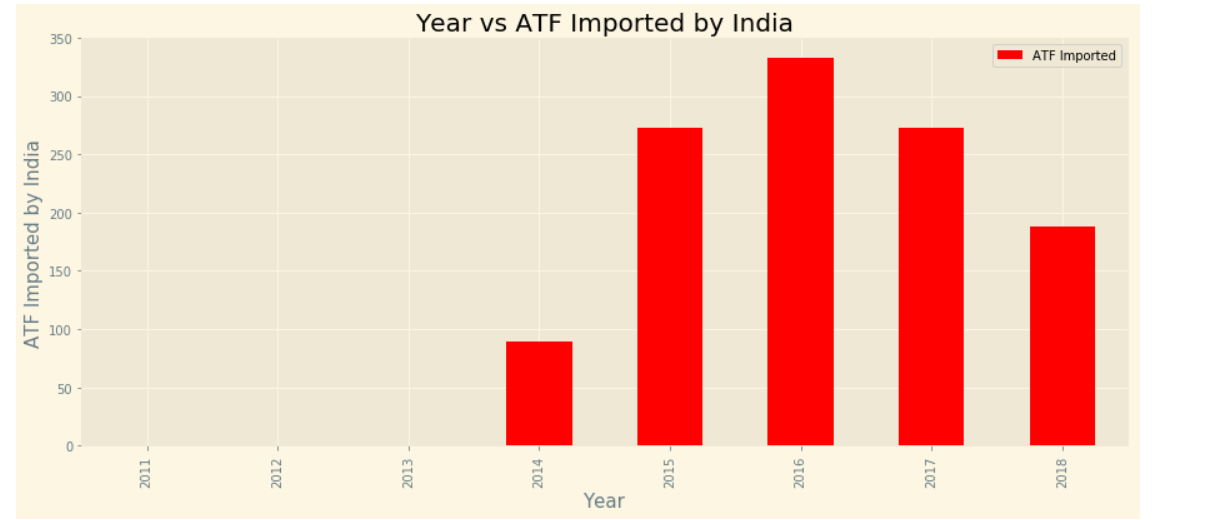
**Outline:**

In this chapter, we are going to discuss the significant project outcomes and the project’s applicability in the real world.

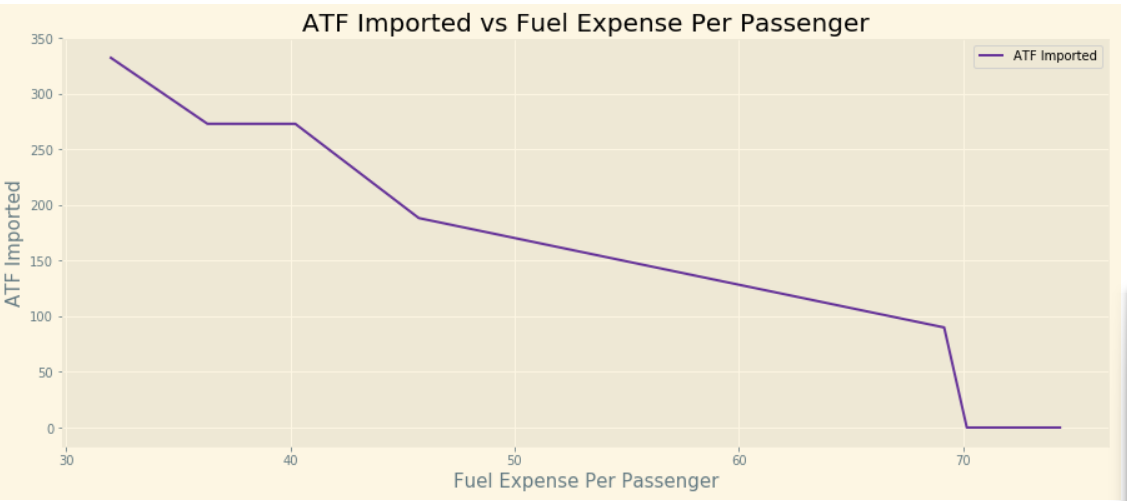
**Significant Project Outcomes:**

In our project, we have done three tasks for which there are three types of outcomes.

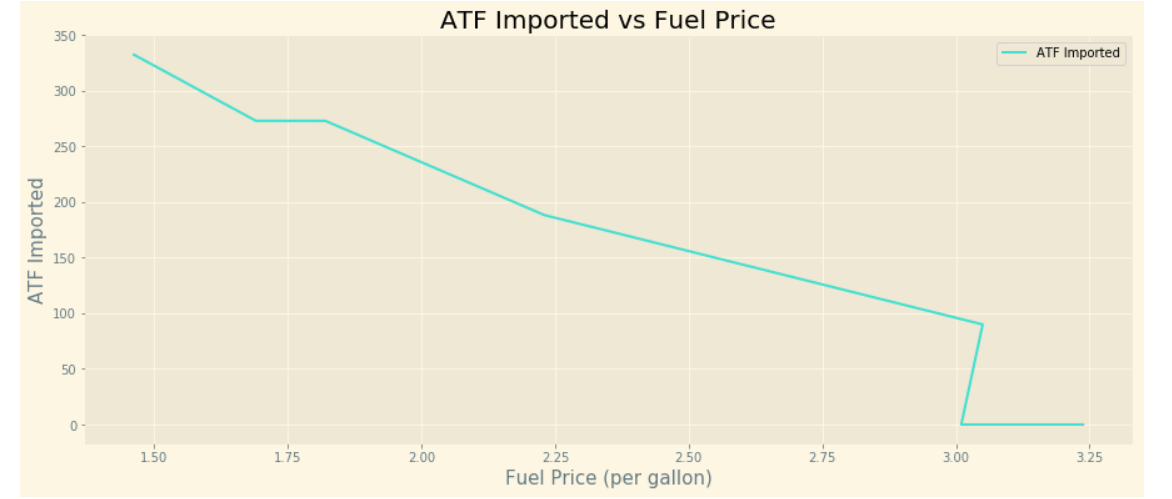
1. **Data Analyzing Outcomes:** In the project, we have merged two datasets and after analyzing, visualizing the data then we have got the following outcomes.

****

**Fig-2**

****

**Fig-3**

****

**Fig-4**

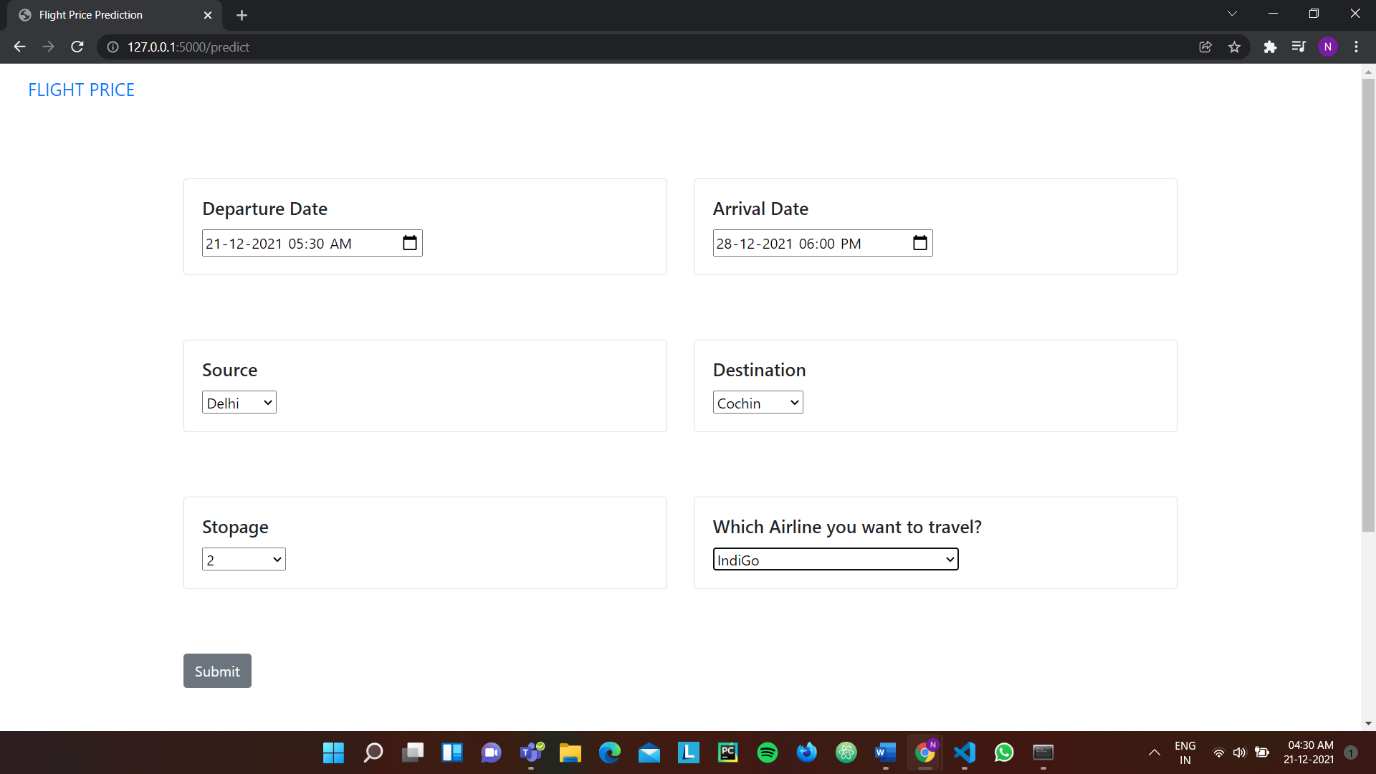
* In Fig-2, we can see, India has imported ATF in a max amount in 2016, after that its importing is continuously decreasing
* In Fig-3, we can easily analyze that, as import of ATD decreases, Fuel expense per passenger increases.
* From Fig-4, we can observe that as the import of ATF decreases, Fuel Prices increase.

1. **ML algorithm comparison:** In this project, we have compared five algorithms on the dataset and found the best one. The used algorithm and its R2 score are as follows.
2. Linear Regression - 0.61959437290701
3. Random Forest - 0.7970013354474519
4. Decision Tree - 0.7325160576230549
5. Support Vector Regression - 0.055960321034518756
6. Lasso Regression - 0.6190467615317858

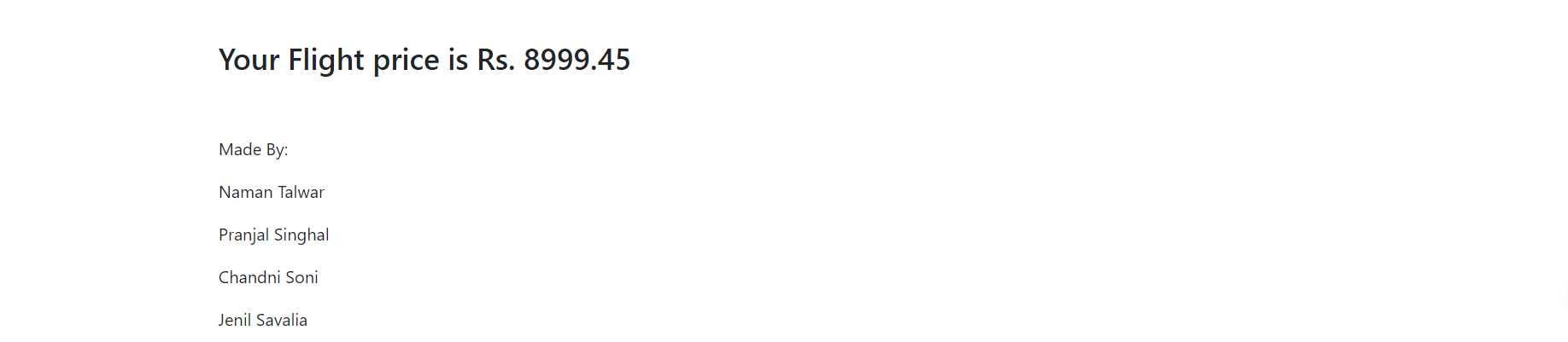
Here, we can see that the best algorithm for our dataset is the **Random Forest Algorithm** with an R2 score of **0.7970013354474519**.

1. **Input-Output Outcome:** When some user gives the required input our ML model will process it and give the required output on the screen of the web app.

Input:



Output:



**Real-World Application:** We have analyzed the data and found the relation between ATF prices and fuel expense per passenger, this data analysis can be used by various agencies to control the airline fare prices.

Through Our web app, someone can manage their travel expenses according to their budget and availability.

**Chapter – 7**

**CONCLUSIONS AND RECOMMENDATION**

1. **Outline-**

Here, we are going to talk about the limitations of our project after that, future enhancements, and conclusions.

1. **Limitation and Constraint of the System-**

In our project, we have compared four machine algorithms and then later on we have got the best R2 score of the Random-Forest Machine Learning Algorithm as approx. 0.8. We can also increase the same by cleaning the data more precisely.

In this project, due to insufficient time and knowledge, we have not made the web page so attractive. We will try to make it more attractive and user-friendly in the upcoming future.

1. **Future Enhancements-**

In the future, we are going to gather more and more datasets and will try to clean them more precisely so that we will get an amazing accuracy after applying a machine learning algorithm.

For now, our web app is working but it is not looking so attractive and good so, in future, we are going to make it more attractive.

We will also add many other functionalities to our web app like flight booking and many more. We will try our best to make this web app so attractive, convenient and user friendly so people will use our web-app for various purposes.

**REFERENCES**

**DATASETS: -**

* <https://www.kaggle.com/nikhilmittal/flight-fare-prediction-mh>
* <https://www.kaggle.com/xan3011/airline-data-project-mit-1995-2019>
* <https://www.kaggle.com/susree64/import-petro-products>