**Detailed Project Report (DPR)**

**Flight Fare Prediction**

Created On **– 02/08/2022**

**Akshit Singhal**

**Contents**

Abstract

Introduction

Importance of Detailed Project Report (DPR)?

1. Description
   1. Problem Perspective
   2. Problem Statement
   3. Proposed Solution
   4. Solution Improvement
2. Technical Requirements
   1. Tools used
3. Data Requirements
   1. Data Collection
   2. Data Description
   3. Importing data from the database
   4. Exporting data to the databases
4. Data Preprocessing
5. Project Workflow
   1. Data cleansing and data preprocessing
   2. Model building
   3. UI Integration
   4. Deployment
6. Data from user
7. Data Validation
8. Rendering the results
9. Deployment
10. Conclusion

Abstract

For years, the introduction and growth of platforms to be used by end customers and passengers has been based on transparency about cost, routs, global alternatives, and required travel time. This was a prerequisite and starting point to increase in the growth rate of low-cost carriers and for questioning the traditional business model of legacy airlines. The revolution in over in that segment, new business models will come but we find ourselves in the post revolution era.

However, there are good opportunities to take ideas and learnings from industry 4.0 and use them in aircraft production. In particular, the use of tools for customer-specific options could be reduced by software driven solutions and intelligent implementation of laser pointing mechanisms. When it comes to the implementation of industry 3.0 i.e., automation and industry 4.0 that is digitalization the level of maturity is not homogenous throughout the supply chain.

INTRODUCTION

Importance of DPR Documentation?

The main purpose of this DPR documentation is to add the necessary details of the project and provide the description of the machine learning model and written code. This also provides the detailed description on how the entire project has been designed end to end.

Key Points:

* Describes the Design flow
* Implementation
* Software requirements
* Architecture of project
* Non-functional attributes like:
  + Reusability
  + Portability
  + Resource utilization

1. Description
   1. Problem Perspective

The flight fare prediction is the machine learning model which helps us to predict the cost of the flight ticket and helps the users to know the cost of the flight ticket this is very beneficial to users when it comes to the festive season or vacations because they can know the flight fare beforehand and prepare their plans accordingly.

* 1. Problem Statement

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, and 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. The main goal is to predict the fares of the flights based on different factors available in the provided dataset.

* 1. Proposed Solution

The solution proposed to take the required inputs from the user through the web interface created by us, pass this requirement to our machine learning model, and based on these inputs we have to print that ‘’XYZ” is the predicted cost of flight.

* 1. Solution Improvements

Cost of ticket can be predicted considering some other features like festive season, weekdays, weekends, discounts given by the airlines owners and some other factors may also be considered. But as of now we are not considering all of this because discounts and all these features depends on company and their profits.

1. Technical Requirements

There are no hardware requirements required as such. The user using our applications must have an active internet connection and user should have the basic understanding of adding some information on our webpage as an input. Also, the backend part of it must satisfy all the technical requirements and should be able to display the result on the screen.

* 1. Tools Used
* Python version 3.7 is used in this project.
* Some python libraries like NumPy, pandas, matplotlib, seaborn and scikit learn are used for implementation of machine learning algorithms.
* Jupyter notebook and Visual studio code is used as IDE for writing the code.
* Cassandra database is used as the database for this project.
* HTML and CSS are used for developing the front end of our web application.
* Flask is used for the backend development.
* Github is used as the version control system.
* Heroku is used for deployment of model.

1. Data Requirement

Whenever we are working on any project the data is completely dependent on requirement of the problem statement. For this project that is flight fare prediction data is available on the kaggle.com website. The data available over here is in the form of excel file(.xlsx) format. As we have to take experience of real-world use case, we are importing the data from the Cassandra database and importing it to the .csv file format.

* 1. Data Collection from the main source

The data which is used in this project was given by the client itself in the excel format and available on the kaggle.com website.

Dataset link:<https://www.kaggle.com/nikhilmittal/flight-fare-prediction-mh>

* 1. Data Description

The dataset contains more that 10 thousand lines of dataset having the features like airline name, date of journey, source, destination, route, arrival time, departure time, duration, total stops, additional information, and the final target column that is price.

* 1. Importing Data from the database

Created the API for uploading the data into the Cassandra database

* Step 1: Making connection with the database
* Step 2: Database creation with the name flight fare
* Step 3: Cqlsh command is written for creating data table with the required features.
* Step 4: Finally, Cqlsh is used for uploading the dataset into data table for bulk insertion.
  1. Exporting data to the database

While creating the API which is mentioned in the above step the importing URL is also created in parallel so that it would download the data in the .csv format.

1. Data Preprocessing

* Checking the basic profile of the dataset.
* Checking for the null values, there are some null values so in this case we are dropping it because they are very less in numbers.
* There are some features like departure time, arrival time, duration which our machine learning algorithm is unable to understand so we need to extract basic information like hours and minutes which we are storing in new columns.
* Similarly, we must extract the journey date and journey column from date of journey column.
* There are some categorical features like number of stops, source, destination all this feature must be encoded this process is called as one hot encoding.

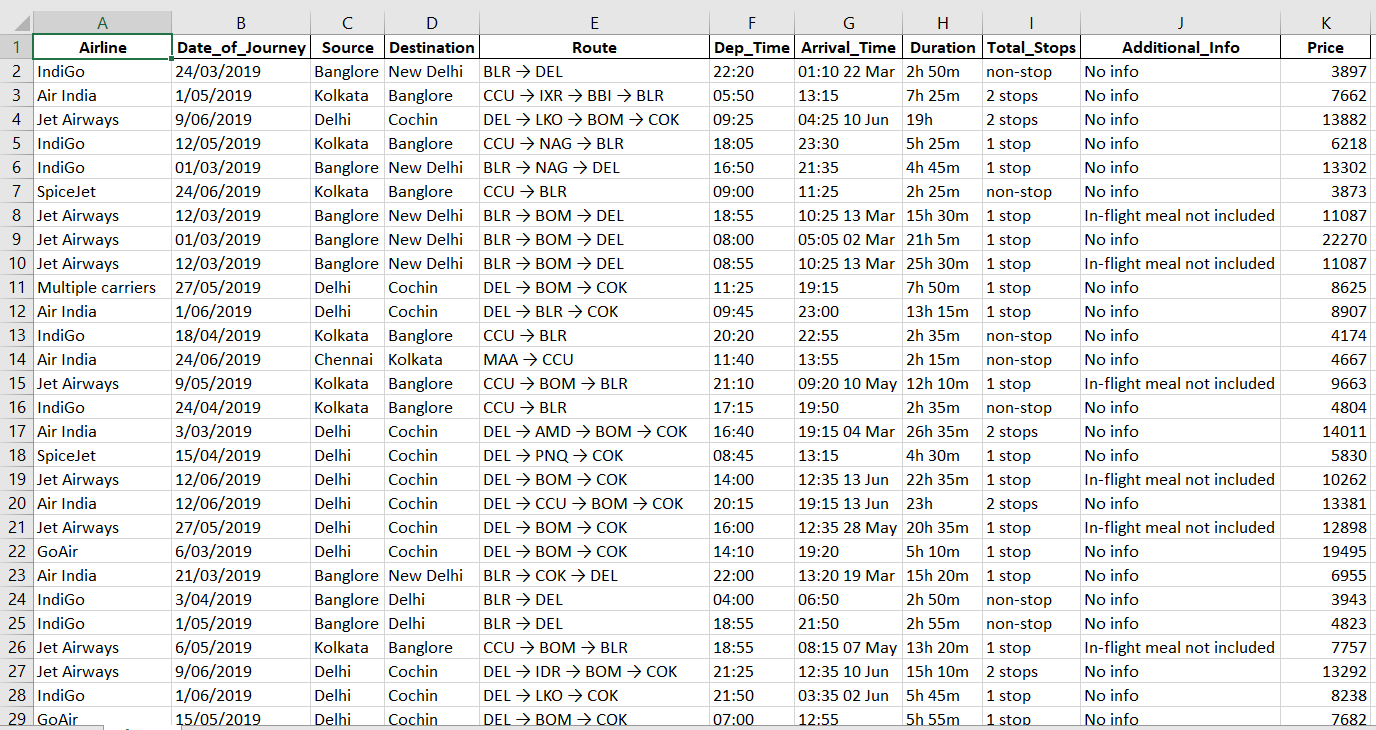
Now, data is ready to passed to machine learning algorithm.

1. Project Workflow

Project workflow gives you the flow of the project from start to the end.

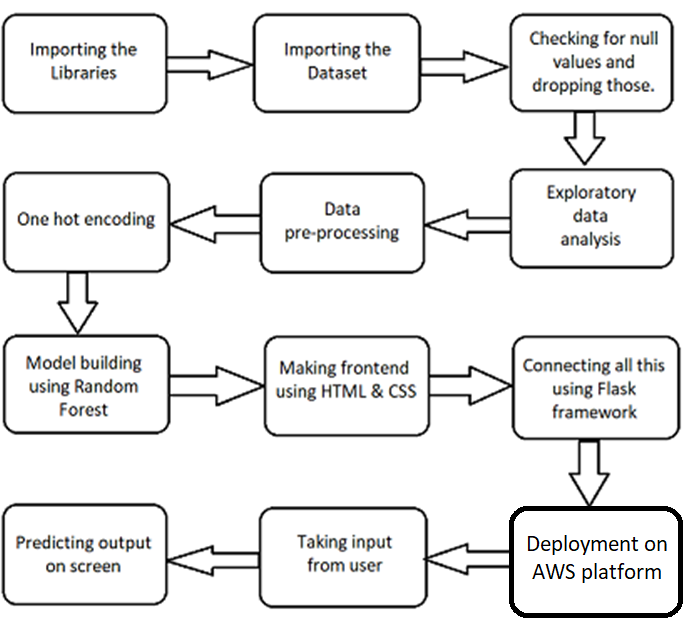
* 1. Data collection and data analysis

As discussed in the data preprocessing process our data is been filtered and is made ready in such a way that a computer system can understand it.



* 1. Modelling and Building

Almost all the regression machine learning algorithms we have tried on the same dataset in order to get the maximum accuracy we have found out that random forest can give the maximum accuracy with the hyperparameter tuning that is 83.26% on testing dataset.



* 1. UI Integration

HTML and CSS is being used to create the front-end interface of the webpage of our web app.

* 1. Deployment

This model is deployed on Heroku which is the cloud platform by Heroku.

1. Data from the user

The data from the user is being retrieved from the front-end web app which is created using HTML and CSS.

1. Data Validation

The data is taken form the frontend web app then sent to the machine learning model but, before that data which is entered by user is been validated by app.py which is the flask application.

1. Rendering the result

The data which is taken as the input from user is sent to the machine learning model and then the result is rendered on the front-end HTML page.

1. Deployment

This model is deployed on AWS Ec2 instance. The following are the steps to deploy the model on the AWS platform:

* Create an AWS account
* Create an EC2 instance
* Edit security group
* Download the keygen (.pem file)
* Download and install Putty and WinSCP
* Install packages on EC2 using putty
* Finally run app.py command

Conclusion:

We have successfully built the end-to-end web application using machine learning which can predict the fare of the flights using the features which is entered by the user.