DPR

FLIGHT FARE P REDIC TION

Antima Nakhat

Abstract

Recent global events have had a significant impact on the aviation sector for various reasons. This impact can be viewed from two main perspectives:

1. **Business Perspective**
2. **Customer Perspective**

Safety has been the primary reason for these disruptions. In response, governments across the world have implemented new regulations for their respective airline companies. These restrictions have affected flight availability and passenger capacity. As a result, flight ticket prices have increased and now vary significantly depending on the location.

Flight bookings are now typically made through two methods: online and offline. Each of these methods has different criteria influencing ticket prices — for example, server load and the number of concurrent booking requests can affect costs in online booking.

In this machine learning implementation, we aim to analyse the various factors that affect flight ticket pricing and predict the most appropriate cost of a ticket based on given inputs.

INTRODUCTION

Why this DPR Documentation?

The main purpose of this Detailed Project Report (DPR) is to outline the specifics of the project, explain the machine learning model used, and describe the supporting code. This document provides a detailed overview of how the project was designed and implemented end-to-end.

Key Points Covered

- Design flow
- Implementations
- Software requirements
- Project architecture
- Non-functional attributes such as:
  - Reusability
  - Portability
  - Resource utilization

1 Description

1.1 Problem Perspective

Flight fare prediction is a machine learning application designed to estimate the cost of flight tickets, helping users make informed travel decisions.

1.2 Problem Statement

The main objective of this project is to build a user interface that predicts the flight ticket cost based on inputs provided by the user, such as journey date, departure location, and destination.

1.3 Proposed Solution

The proposed solution involves capturing the user's input through a custom interface, processing this data using a trained machine learning model, and finally displaying the predicted ticket price.

1.4 Solution Improvements

Future enhancements could include predicting prices based on weekdays, holiday seasons, or other social trends. However, from a business standpoint, incorporating discounted ticket prediction could result in financial losses for airline companies. Therefore, this approach is currently not considered.

2 Technical Requirements

There are no specific hardware requirements for using this application. Users simply need an internet-enabled interactive device and basic knowledge of data input. On the backend, the server should run all necessary software required to process the data and return predictions.

2.1 Tools Used

- Python 3.9 is used as the programming language and frame works like NumPy, pandas, sklearn and other modules for building the model.
- vscode is used as IDE.
- For visualizations seaborn and parts of matplotlib are being used.
- For data collection csv file from kaggle is being used.
- Front end development is done using HTML/CSS.
- Flask is used for both data and backend deployment .

3 Data Requirements

The data requirement is completely based on the problem statement. And the data set is available on the Kaggle in the form of excel sheet(.xlsx). As the main theme of the project is to get the experience of real time problems

3.1 Data Gathering from Main Source

The data for the current project is being gathered from Kaggle dataset

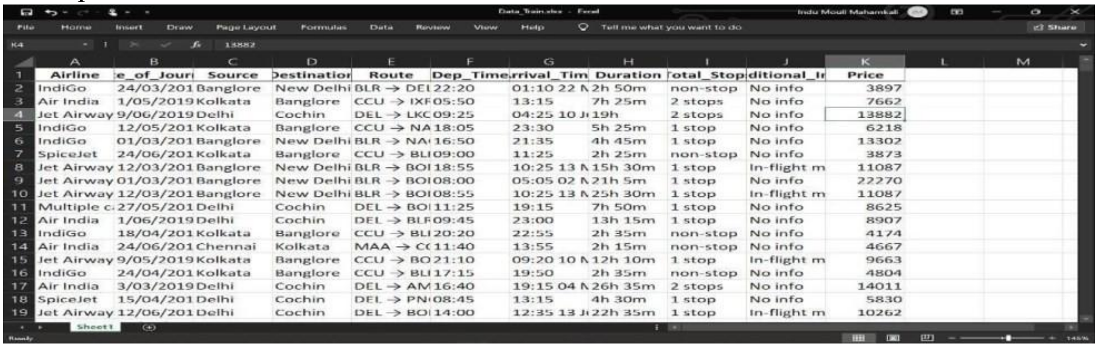
3.2 Data Description

The dataset contains over 10,000 records of flight-related information including:

- Airline
- Date of journey
- Source
- Destination
- Departure time
- Arrival time
- Duration
- Total stops

- Additional information
- Price

A sample of the dataset is shown below:



Airline	e_of_Jouri	Source	Destinatior	Route	Dep_Time	rrival_Tim	Duration	Total_Stop	ditional_Ir	Price
IndiGo	24/03/201	Banglore	New Delhi	BLR → DEI	22:20	01:10 22 N	2h 50m	non-stop	No info	3897
Air India	1/05/2019	Kolkata	Banglore	CCU → IXF	05:50	13:15	7h 25m	2 stops	No info	7662
Jet Airway	9/06/2019	Delhi	Cochin	DEL → LKC	09:25	04:25 10 Ji	19h	2 stops	No info	13882
IndiGo	12/05/201	Kolkata	Banglore	CCU → NA	18:05	23:30	5h 25m	1 stop	No info	6218
IndiGo	01/03/201	Banglore	New Delhi	BLR → NA	16:50	21:35	4h 45m	1 stop	No info	13302
SpiceJet	24/06/201	Kolkata	Banglore	CCU → BLI	09:00	11:25	2h 25m	non-stop	No info	3873
Jet Airway	12/03/201	Banglore	New Delhi	BLR → BOI	18:55	10:25 13 N	15h 30m	1 stop	In-flight m	11087
Jet Airway	01/03/201	Banglore	New Delhi	BLR → BOI	08:00	05:05 02 N	21h 5m	1 stop	No info	22270
Jet Airway	12/03/201	Banglore	New Delhi	BLR → BOI	08:55	10:25 13 N	25h 30m	1 stop	In-flight m	11087
Multiple c	27/05/201	Delhi	Cochin	DEL → BOI	11:25	19:15	7h 50m	1 stop	No info	8625
Air India	1/06/2019	Delhi	Cochin	DEL → BLF	09:45	23:00	13h 15m	1 stop	No info	8907
IndiGo	18/04/201	Kolkata	Banglore	CCU → BLI	20:20	22:55	2h 35m	non-stop	No info	4174
Air India	24/06/201	Chennai	Kolkata	MAA → CC	11:40	13:55	2h 15m	non-stop	No info	4667
Jet Airway	9/05/2019	Kolkata	Banglore	CCU → BO	21:10	09:20 10 N	12h 10m	1 stop	In-flight m	9663
IndiGo	24/04/201	Kolkata	Banglore	CCU → BLI	17:15	19:50	2h 35m	non-stop	No info	4804
Air India	3/03/2019	Delhi	Cochin	DEL → AM	16:40	19:15 04 N	26h 35m	2 stops	No info	14011
SpiceJet	15/04/201	Delhi	Cochin	DEL → PN	08:45	13:15	4h 30m	1 stop	No info	5830
Jet Airway	12/06/201	Delhi	Cochin	DEL → BOI	14:00	12:35 13 Ji	22h 35m	1 stop	In-flight m	10262

4 Data Pre-Processing

Steps performed in pre-processing are:

- First, the data types of each column were checked. It was found that only the **Price** column was of type integer.
- Null values were identified in a few rows, which were subsequently dropped.
- All necessary columns were converted to **datetime** format where applicable.
- One-hot encoding was applied to categorical columns to prepare the data for machine learning.

At this stage, the data is ready to be passed into a machine learning algorithm.

5 Design Flow

5.1 Modelling

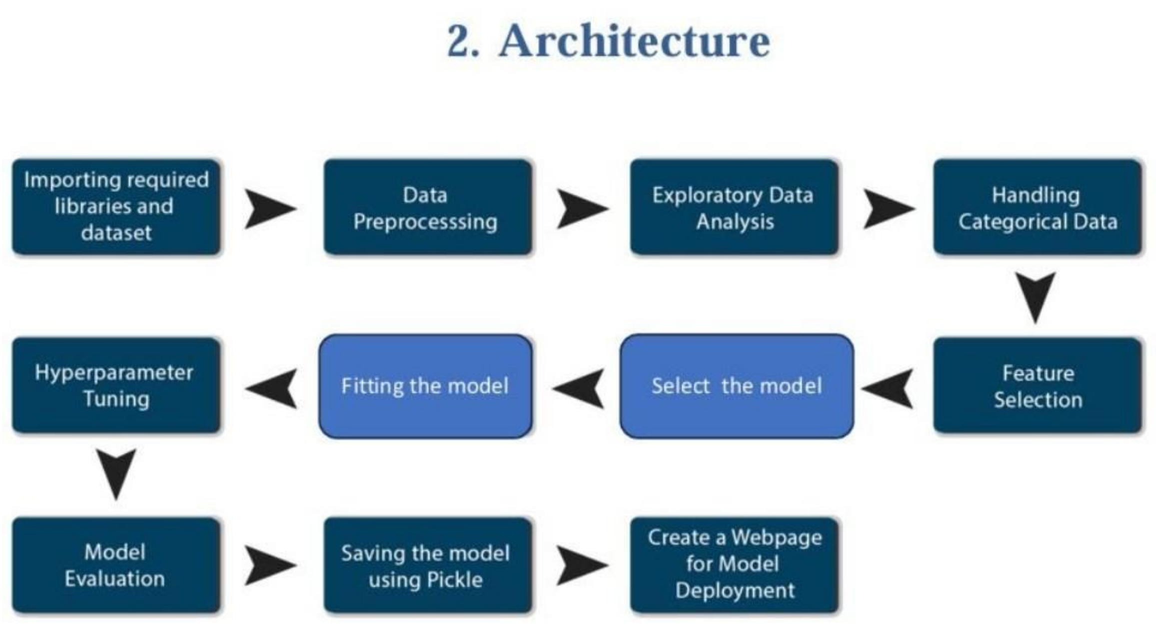
The pre-processed data was visualized, and key insights were extracted. Although the data appeared randomly distributed, multiple machine learning algorithms were tested to ensure comprehensive evaluation.

Among them, Random Forest Regression performed the best. To further improve accuracy, hyperparameter tuning was applied.

5.2 UI Integration

HTML and CSS files were created and integrated with the machine learning model. These front-end components were then linked to the app.py file, which served as the application backend. The complete system was successfully tested in a local environment.

5.3 Deployment Process



6 Data from User

The data from the user is retrieved from the created HTML web page.

7. Data Validation

The data provided by the user is processed by the app.py file and validated. Once validation is complete, the data is passed on for prediction.

---

8. Rendering the Results

The predicted result is rendered and displayed on the web page for the user to view.

---

9. Deployment

The trained and tested model is deployed to render, allowing users to access the application from any internet-enabled device.

Conclusion

The flight fare prediction model estimates the cost of a flight based on the input provided by the user and the data the model has been trained on. This allows users to get an approximate idea of ticket prices for their journey.