PROJECT REPORT ON
Flight Price Predictor

AIML Six Week Course At NIELIT Ropar

BY:-

Akshat Saini
Saurabh Yadav
Abhijit Singh
Parneet Kaur
Sarthak Sharma

Abstract

The ever-changing nature of flight prices makes travel budgeting a challenge. This project aims to develop a model leveraging machine learning to predict flight fares with greater accuracy.

Project will:

- **Identify key factors influencing flight prices, including origin, destination, travel dates, airlines, and historical trends.**
- **Collect and pre-process flight data to prepare it for machine learning algorithms.**
- **Train and evaluate various machine learning models to predict flight prices effectively.**
- **(Optional) Develop a user-friendly interface for travelers to input flight details and receive predicted prices.**

By providing reliable predictions, this project empowers travelers to make informed decisions and potentially save money on their trips. Additionally, the insights gained from the model could be valuable for airlines in optimizing their pricing strategies.

Contents

List of Figures

List of Program Code

Chapter 1 :
AI Ecosystem

1.1. Introduction

An AI ecosystem is a network of people, organizations, and technologies that work together to advance the field of artificial intelligence (AI). This can include AI researchers and engineers, AI-focused companies, academic institutions, and government agencies. The goal of an AI ecosystem is to create an environment that fosters innovation, collaboration, and the sharing of ideas and resources.
An AI ecosystem typically includes a variety of different components, such as:

Data: AI algorithms need large amounts of data to learn and make predictions. An AI ecosystem includes organizations that collect and curate data, as well as tools and infrastructure for storing, accessing, and analyzing data.
Computation: AI algorithms require significant computational power to process data and make predictions. An AI ecosystem includes organizations that develop and provide access to high-performance computing resources, such as cloud-based AI platforms and specialized hardware like graphics processing units (GPUs).

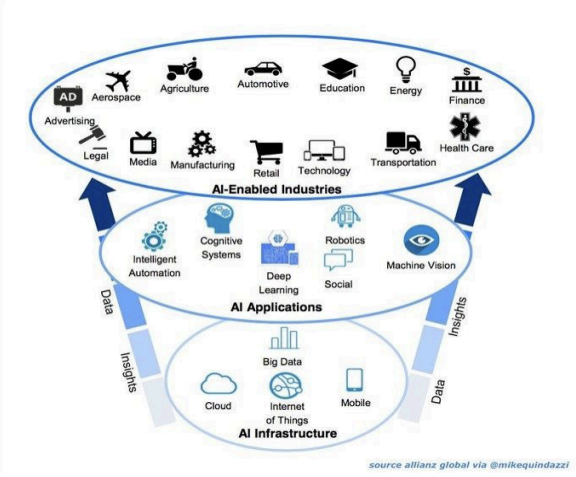


Figure 1:AI Ecosystem

Algorithms and models: AI algorithms are the building blocks of AI systems, and they are developed by researchers and engineers. An AI ecosystem includes organizations and individuals that develop and share new algorithms and models, as well as tools and frameworks for creating and deploying AI Systems.

Applications: **AI algorithms are used to solve a wide range of problems, from image recognition and natural language processing to autonomous vehicles and personalized medicine. An AI ecosystem includes organizations that develop and deploy AI-powered applications in various industries, as well as tools and platforms for creating and deploying AI applications. Overall, an AI ecosystem is a dynamic and evolving environment that plays a critical role in driving innovation and progress in the field of AI.**

1.2. Benefits of incorporating AI ecosystems

MSPs can gain many benefits from replacing time-intensive, manual tasks with AI-based solutions. When successfully implemented, AI-enabled solutions enhance cost savings, service delivery, network upkeep and security, customer support, scalability, and orchestration in the following ways:

- **Cost savings: Automating repetitive tasks frees up resources, allowing for better allocation of manpower, ultimately leading to cost savings.**
- **Faster and better service delivery: Automation of repetitive tasks results in improved response times and overall efficiency, leading to consistent service standards and better customer experiences.**
- **Network upkeep and security: AI-based maintenance and activities enhanced by detecting and addressing threats promptly, as well as identifying and updating outdated devices.**
- **Improved support quality: Automated processes are less prone to human error, ensuring higher quality support. Any errors can be identified and rectified during the initial testing phases.**
- **Enhanced scalability: AI-based processes can be efficiently scaled to various business areas, leading to faster growth and the ability to maintain service quality even with increased business volume.**
- **Orchestration and hyper automation: AI-based processes enable sequential task orchestration, streamlining entire processes. The integration of AI, analytics, and robotic process automation (RPA), known as hyper automation, further enhances efficiency and effectiveness.**

Integrating AI into vital business areas introduces transformative changes that benefit MSPs. By harnessing the power of automation, they can not only streamline their workflows and boost efficiency but also unlock greater potential for client satisfaction.

Chapter 2: Python Programming

1. Introduction

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991. It is used for:

- **web development (server-side),**
- **software development,**
- **mathematics,**
- **system scripting. What can Python do?**
- **Python can be used on a server to create web applications.**
- **Python can be used alongside software to create workflows.**
- **Python can connect to database systems. It can also read and modify files.**
- **Python can be used to handle big data and perform complex mathematics.**
- **Python can be used for rapid prototyping, or for production-ready software development.**

2. Why Python?

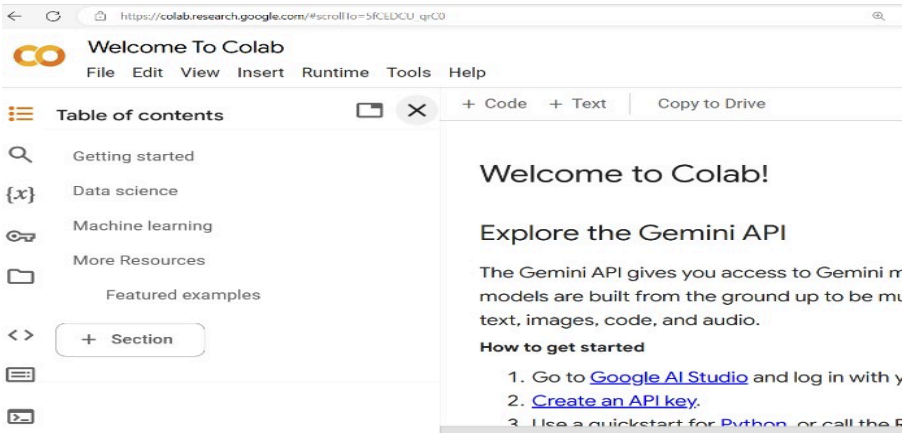
- **Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).**
- **Python has a simple syntax similar to the English language.**
- **Python has syntax that allows developers to write programs with fewer lines than some other programming languages.**
- **Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.**
- **Python can be treated in a procedural way, an object-oriented way or a functional way. Good to know**
- **The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.**
- **In this tutorial Python will be written in a text editor. It is possible to write Python in an Integrated Development Environment, such as Thonny, Pycharm, Netbeans or Eclipse which are particularly useful when managing large collections of Python files.**

Python Syntax compared to other programming languages

- **Python was designed for readability, and has some similarities to the English language with influence from mathematics.**
- **Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.**
- **Python relies on indentation, using whitespace, to define scope; such as the scope of loops, functions and classes. Other programming languages often use curly-brackets for this purpose.**

- **Python programming with Colab**

  Colab, or "Colaboratory", allows you to write and execute Python in your browser, with



- **Zero configuration required**
- **Access to GPUs free of charge**
- **Easy sharing ,Whether you're a student, a data scientist or an AI researcher, Colab can make your work easier.**

My first Colab code file

```
[ ]  df1.head()
```

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40

```
[ ]  gen_variation=df1.groupby("Gender")
```

```
[ ]  gen_variation["Annual Income (k$)"].count()
```

```
Gender
Female    112
Male       88
Name: Annual Income (k$), dtype: int64
```

```
[ ]  x = df1.loc[:, "Annual Income (k$)"]
     y= df1.loc[:, "Spending Score (1-100)"]
```

LINK:-

https://colab.research.google.com/drive/1c_t89KWSZbXAgJFPytx38DKHa9ay8Y-K?usp=sharing

Chapter 3 : Numpy

3.1 Introduction

NumPy: The Workhorse for Numerical Computing in Python

NumPy, short for Numerical Python, is a fundamental library for scientific computing in Python. It provides a powerful foundation for various data analysis tasks, especially those involving large datasets and complex calculations.

Here's a breakdown of what NumPy offers:

- **Multidimensional Arrays:** **The core of NumPy is the nd array object, a versatile data structure for storing multidimensional arrays of elements, all of the same data type. This allows efficient handling of matrices, vectors,and higher-dimensional data.**
- **Mathematical Functions:** **NumPy boasts a rich collection of mathematical functions that operate efficiently on entire arrays at once. These functions cover linear algebra operations, trigonometric calculations, random number generation, and more.**
- **Performance:** **Compared to using regular Python lists, NumPy arrays are optimized for numerical computations.They leverage C and Fortran code under the hood, resulting in significantly faster execution for array-based operations.**
- **Broadcasting:** **A powerful feature of NumPy, broadcasting allows performing operations on arrays of different shapes under certain conditions. This simplifies calculations involving arrays with compatible dimensions.**
- **Foundation for Other Libraries:** **NumPy serves as the bedrock for many popular scientific Python libraries like Pandas (data analysis) and SciPy (advanced scientific computing).**

Chapter 4 :

Exploratory Data Analysis (EDA) with Pandas

4.1 Introduction

Pandas shines in the realm of Exploratory Data Analysis (EDA). It offers a plethora of functions and methods to help you get a feel for your data, identify patterns and trends, and uncover potential issues before diving deeper into analysis or machine learning.

Here's a breakdown of how Pandas aids in EDA:

Understanding Data Structure:

- **`head()` and `tail()`: These functions provide a quick glimpse at the beginning and end portions of your data,respectively.**
- **`info()`: This method offers a summary of the data, including data types, number of non-null values, and memory usage for each column.**
- **`shape`: This attribute reveals the dimensions of your data (number of rows and columns).**

Data Cleaning and Preprocessing:

- **`isnull().sum()`: This helps identify columns with missing values and their counts.**
- **`fillna()`: This function allows you to impute missing values with a specific value (e.g., mean, median) or a strategy.**
- **`dtypes`: This attribute displays the data types of each column, enabling you to convert them if necessary for analysis.**

Descriptive Statistics:

- **`describe()`: This workhorse function provides summary statistics for numerical columns, including mean,standard deviation, quartiles, minimum, and maximum values.**
- **`value_counts()`: This function calculates the frequency of each unique value in a column, useful for categorical data exploration.**

Data Visualization (often with Matplotlib or Seaborn):

- **GroupBy and Aggregation: Pandas allows grouping data by specific columns and performing aggregations like `mean()`, `sum()`, or `count()`. These aggregations can then be visualized using Matplotlib or Seaborn to reveal trends across different categories.**
- **Histograms, Scatter Plots, and Box Plots: These visualizations can be created using libraries like Matplotlib or Seaborn based on Pandas DataFrames, helping to uncover relationships between variables and identify outliers.**

Chapter 5: Machine Learning

5.1 Introduction

Machine learning (ML) is a fascinating subfield of Artificial Intelligence (AI) that empowers computers to learn without explicit programming. Imagine a system that can improve its performance on a task by analyzing data, just like humans learn from experience. That's the core idea behind machine learning!

Here's a breakdown of key concepts in ML:

- **Learning from Data: ML algorithms are trained on datasets that contain examples (data points) and corresponding labels (desired outputs). The algorithm analyzes the patterns within this data to learn a model that can then be used to make predictions or classifications on new, unseen data.**
- **Types of Machine Learning: There are several categories of machine learning, each suited for different tasks:**
  - **Supervised Learning: Involves training a model with labeled data, where each example has a corresponding output value. The model learns to map inputs to desired outputs and can then predict outputs for new data.(Ex: Spam filtering)**
  - **Unsupervised Learning: Deals with unlabeled data, where the model seeks to uncover hidden patterns or structures within the data itself. This can be used for tasks like customer segmentation or anomaly detection.(Ex: Recommender systems)**
  - **Reinforcement Learning: Involves an agent interacting with an environment and learning through trial and error. The agent receives rewards for desired actions and penalties for undesired ones, enabling it to refine its behavior over time. (Ex: Training a robot to navigate a maze)**
- **Machine Learning Models: These are the core components that embody the learned knowledge. Common examples include:**
  - **Linear Regression: A model that learns to fit a linear equation to the data, used for continuous predictions.**
  - **Decision Trees: These tree-like structures classify data points based on a series of yes/no questions, leading to a final classification.**
  - **Artificial Neural Networks: Inspired by the structure of the human brain, these complex networks learn intricate relationships within data and are powerful for various tasks, including image recognition and natural language processing.**

Applications of Machine Learning: Machine learning finds applications in a vast array of fields, including:

- **Image and Speech Recognition:** **Facial recognition software, self-driving cars, and virtual assistants all leverage machine learning for these tasks.**
- **Natural Language Processing:** **Machine translation, sentiment analysis, and spam filtering are powered by ML models that understand and process human language.**
- **Recommender Systems:** **The suggestions you see on e-commerce platforms or streaming services are often driven by machine learning algorithms that analyze your past behavior and preferences.**

Machine learning is a rapidly evolving field with immense potential. As you delve deeper, you'll discover a multitude of algorithms, techniques, and applications that are shaping the future!

Chapter 6:

Flight Price Prediction System

6.1 About Project:

The Flight Price Prediction system is a machine learning-based application designed to predict the prices of flights based on various user inputs. Utilizing a Random Forest Regressor, this model takes into account several factors that influence flight prices, including the airline, source, destination, number of intermediate stops, date of travel, duration etc. By analyzing these inputs, the system provides users with an estimated flight price, helping them make informed decisions when planning their travel.

This project aims to demonstrate the application of machine learning techniques in real-world scenarios, offering users a practical tool to estimate flight costs. The system's ability to analyze multiple variables and provide accurate predictions showcases the power and utility of machine learning in everyday applications.

6.2 Dataset:

The dataset used for the Flight Price Prediction system was sourced from Kaggle. This comprehensive dataset includes detailed information about various flights, which is crucial for building an accurate prediction model.

The CSV dataset contains the following columns:

- **Airline: The name of the airline operating the flight.**
- **Source: The departure location of the flight.**
- **Destination: The arrival location of the flight.**
- **Total_Stops: The number of stops between the source and destination.**
- **Price: The price of the flight ticket, which serves as the target variable for the prediction model.**
- **Date: The day of the month the flight is scheduled to depart.**
- **Month: The month the flight is scheduled to depart.**
- **Year: The year the flight is scheduled to depart.**
- **Dep_hours: The hour of the day the flight is scheduled to depart.**
- **Dep_min: The minute of the hour the flight is scheduled to depart.**
- **Arrival_hours: The hour of the day the flight is scheduled to arrive.**
- **Arrival_min: The minute of the hour the flight is scheduled to arrive.**
- **Duration_hours: The duration of the flight in hours.**

- **Duration_min: The duration of the flight in minutes.**

The dataset has a shape of 10683 rows and 14 columns, providing a robust foundation for analyzing and predicting flight prices. By leveraging this rich dataset, the project aims to capture the underlying patterns that influence flight prices and use this information to develop a reliable prediction model.

6.3 Pre-Processing Data:

Data preprocessing is a crucial step in building a reliable machine learning model. It involves cleaning and transforming the raw data into a suitable format for analysis. For the Flight Price Prediction system, the following preprocessing steps were applied to the dataset:

- ❖ **Handling Missing Values:**
  - ➢ **Checked for any missing values in the dataset and addressed them appropriately. Missing values in numerical columns were filled with the median, and categorical columns with missing values were filled with the mode.**
- ❖ **Date and Time Feature Extraction:**
  - ➢ **Extracted day, month, and year from the 'Date' column.**
  - ➢ **Extracted hour and minute from the 'Dep_hours' and 'Dep_min' columns for departure time.**
  - ➢ **Extracted hour and minute from the 'Arrival_hours' and 'Arrival_min' columns for arrival time.**
  - ➢ **Converted the 'Duration_hours' and 'Duration_min' columns into a single 'Duration' column expressed in minutes.**
- ❖ **Categorical Encoding:**
  - ➢ **Converted categorical columns such as 'Airline', 'Source', 'Destination', and 'Total_Stops' into numerical values using label encoding and one-hot encoding techniques.**
- ❖ **Feature Scaling:**
  - ➢ **Applied feature scaling to normalize the numerical features. This step ensures that all features contribute equally to the model performance, preventing any single feature from dominating due to its scale.**
- ❖ **Outlier Detection and Removal:**
  - ➢ **Identified and removed outliers in the 'Price' column to improve the model's accuracy and robustness.**
- ❖ **Splitting the Dataset:**
  - ➢ **Split the dataset into training and testing sets, typically using an 80-20 or 70-30 ratio. This split allows the model to be trained on one portion of the data and tested on another to evaluate its performance.**

By applying these preprocessing steps, the dataset was transformed into a clean and structured format, ready for building and training the flight price prediction model. Effective data preprocessing ensures the model's accuracy, reliability, and generalization to new data.

6.3.1 Import the packages:

To build and deploy the Flight Price Prediction system, several Python packages and libraries were utilized. These packages are essential for data manipulation, preprocessing, model building, and evaluation. Below are the key packages imported for this project:

- **Pandas: Used for data manipulation and analysis, Pandas provides data structures like DataFrames, which are highly efficient for handling structured data.**
- **NumPy: A fundamental package for numerical computations in Python, NumPy provides support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions.**
- **Matplotlib and Seaborn: These are data visualization libraries. Matplotlib provides a wide range of plotting functions, while Seaborn is built on top of Matplotlib and offers additional features for creating attractive and informative statistical graphics.**
- **Scikit-learn: A machine learning library in Python, Scikit-learn provides simple and efficient tools for data mining and data analysis. It includes various classification, regression, and clustering algorithms, along with tools for model evaluation and selection.**

6.3.2 Reading the CSV file:

The next step in the data preprocessing pipeline is to read the dataset from the CSV file into a Pandas DataFrame. This dataset contains detailed information about various flights, which is essential for training the flight price prediction model.

By loading the CSV file into a DataFrame, we can easily manipulate and analyze the data using Pandas' powerful data handling capabilities. The DataFrame structure allows for efficient data operations, such as filtering, grouping, and statistical analysis, which are crucial for preparing the data for the machine learning model.

Once the dataset is loaded into a DataFrame, we can proceed with the necessary preprocessing steps, including handling missing values, feature extraction, categorical encoding, and feature scaling. This initial step of reading the CSV file sets the foundation for all subsequent data processing and analysis tasks.

6.3.3 Cleaning the data - Numpy/Pandas:

Cleaning the data is a vital step in ensuring the accuracy and reliability of the machine learning model. For the Flight Price Prediction system, the following data cleaning steps were performed using NumPy and Pandas:

1. **Handling Missing Values:**
    - **Checked for any missing (NA or null) values in the dataset.**
    - **Addressed missing values appropriately to prevent them from affecting the model's performance. For numerical columns, missing values were filled with the median of the respective column. For categorical columns, missing values were filled with the mode of the respective column.**
2. **Label Encoding:**
    - **Applied label encoding to convert categorical columns into numerical values. Specifically, the 'Airlines', 'Source', and 'Destination' columns were label encoded. This process involves assigning each unique category a corresponding integer value, which allows the model to process categorical data effectively.**

These cleaning steps ensured that the data was in a consistent and usable format, eliminating potential issues that could arise from missing or categorical data. With clean data, the machine learning model can accurately learn from the dataset and make reliable flight price predictions.

6.3.4 Visualize the clean data - matplotlib:

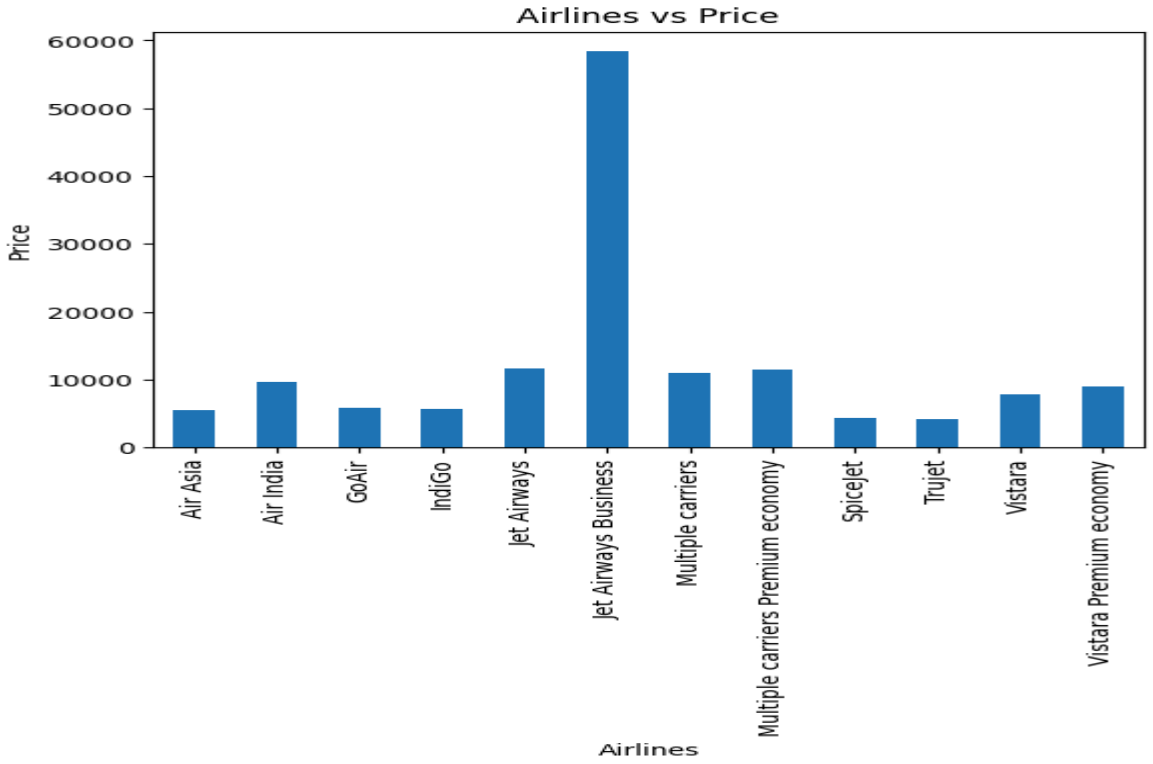
Visualization plays a crucial role in understanding the underlying patterns and relationships within the dataset. Using Matplotlib, we can create clear and informative visualizations that help in interpreting and communicating insights from the cleaned data. Using Matplotlib, we can create various types of plots to visualize different aspects of the cleaned data, such as:

- **Histograms: Visualize the distribution of numerical features, such as flight prices or durations, to understand their spread and central tendencies.**
- **Scatter Plots: Explore relationships between numerical variables, such as the relationship between flight price and duration, to identify any correlations.**
- **Count Plots: Visualize the distribution of categorical variables, such as the number of flights originating from different sources or airlines.**
- **Line Plots: Track trends over time, such as changes in flight prices across different months or years.**

Here's how we visualized the relationship between airlines and flight prices using a bar graph:

Bar Graph for Airlines vs Price:

Created a bar graph to visualize the average flight prices for different airlines.The x-axis represents the airlines, and the y-axis represents the average flight price.This visualization provides a comparative view of how prices vary across different airlines, helping to identify which airlines tend to have higher or lower ticket prices.By visualizing the cleaned data in this manner, we gain insights into the distribution and trends within the dataset, which are essential for further analysis and model building in the Flight Price Prediction system.



6.3.5 Machine learning:

In the context of the Flight Price Prediction system, machine learning techniques are employed to build a predictive model that estimates flight prices based on various input variables. Here's how the machine learning process was applied:

Data Preparation:

- **Split the dataset into features (X) and the target variable (y). Here, X contains all columns except "Price", which represents the flight prices, and y contains the "Price" column.**

Model Selection and Training:

- **Choose the Random Forest Regressor model, a powerful ensemble learning technique suitable for regression tasks.**
- **Initialized the model and trained it on the prepared data.**

Model Evaluation:

- **Evaluated the model's performance using appropriate metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R2) score.**
- **Adjusted the model parameters as needed to optimize performance and ensure accurate predictions.**

By leveraging machine learning algorithms like Random Forest Regressor, the Flight Price Prediction system can effectively learn from the dataset and provide reliable estimates of flight prices based on input variables such as airline, source, destination, and travel date. This approach transforms raw data into actionable insights, enhancing decision-making processes related to flight bookings.

6.4 Front End:

Frontend development involves creating the user interface and experience (UI/UX) of a web application. It focuses on designing and implementing elements that users interact with directly, such as input forms, navigation menus, and content presentation. Using HTML, CSS, and possibly JavaScript, frontend developers ensure that the application is visually appealing, responsive across different devices, and intuitive to use.

6.4.1 : HTML / CSS/ Javascript:

For the Flight Price Prediction system, creating a user-friendly frontend is crucial for users to interact with the prediction model effectively. Here's an overview of the frontend development process using Glitch.com and HTML:

- **Glitch.com Platform:**
  - **Utilized Glitch.com as the development platform for creating and hosting the frontend web application.**
  - **Glitch.com provides a collaborative environment with built-in tools for HTML, CSS, and JavaScript development, making it ideal for rapid prototyping and deployment.**
- **HTML Pages:**
  - **index.html: Serves as the welcome or home page of the web application. It provides an introduction to the Flight Price Prediction**

system and may include instructions or information about the model's functionality.

- **➢ form.html: This page serves as the interface where users input data required for flight price prediction. It typically includes input fields for variables such as airline, source, destination, date, and number of stops.**

- **❖ Styling and Layout:**
  - **➢ Implemented CSS (Cascading Style Sheets) to enhance the visual appeal and usability of the web pages.**
  - **➢ Ensured a responsive design that adapts well to different screen sizes and devices, optimizing the user experience.**
- **❖ Integration with Backend:**
  - **➢ Linked the frontend HTML pages with backend Python code responsible for handling user inputs, executing the prediction model, and displaying results.**
  - **➢ Used appropriate HTTP methods (e.g., POST requests) to send user input data from form.html to the backend for processing.**
- **❖ Deployment and Hosting:**
  - **➢ Hosted the frontend application on Glitch.com, allowing seamless access for users to interact with the Flight Price Prediction system online.**
  - **➢ Tested the functionality and user interface to ensure smooth operation and responsiveness.**

By using Glitch.com and HTML, the frontend of the Flight Price Prediction system provides an intuitive interface for users to input travel details and receive estimated flight prices based on machine learning predictions. This integration of frontend and backend components facilitates a streamlined user experience for accessing and utilizing predictive insights.Here is image of our project's frontend html pages.

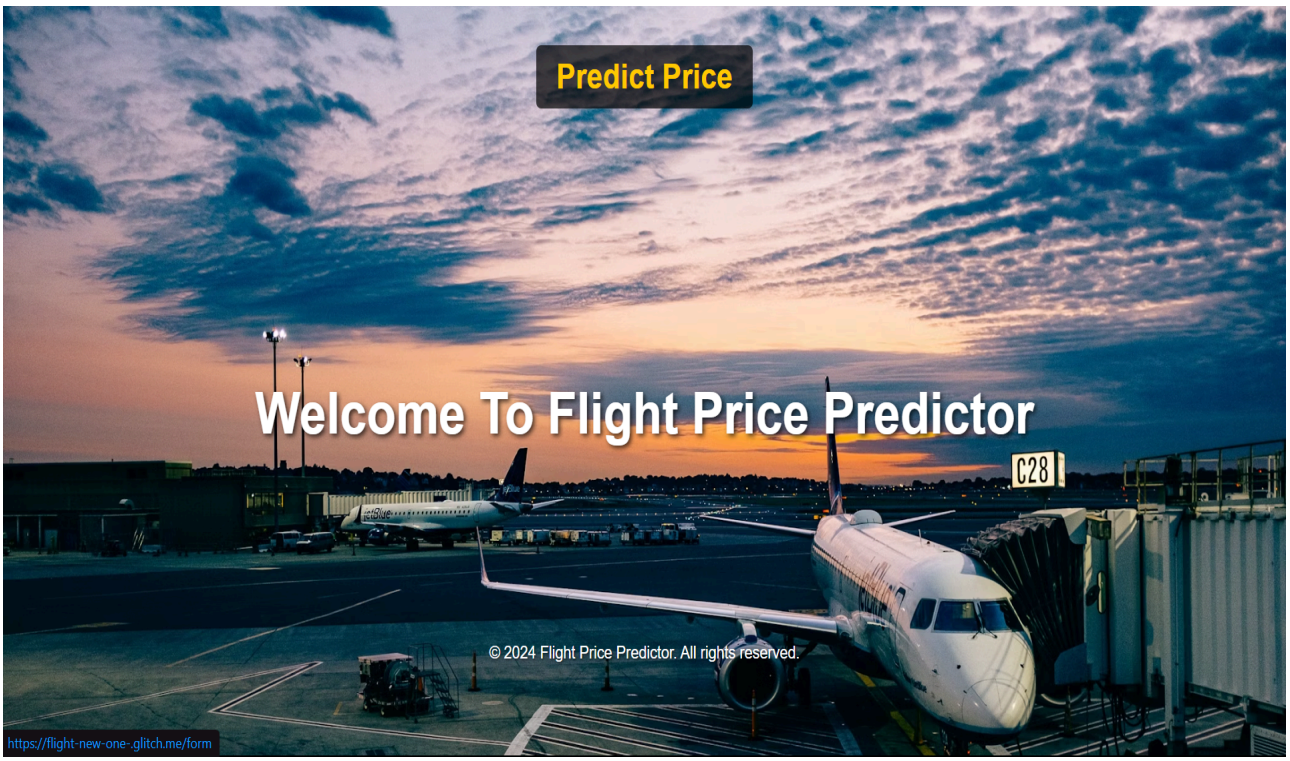


Image = index.html

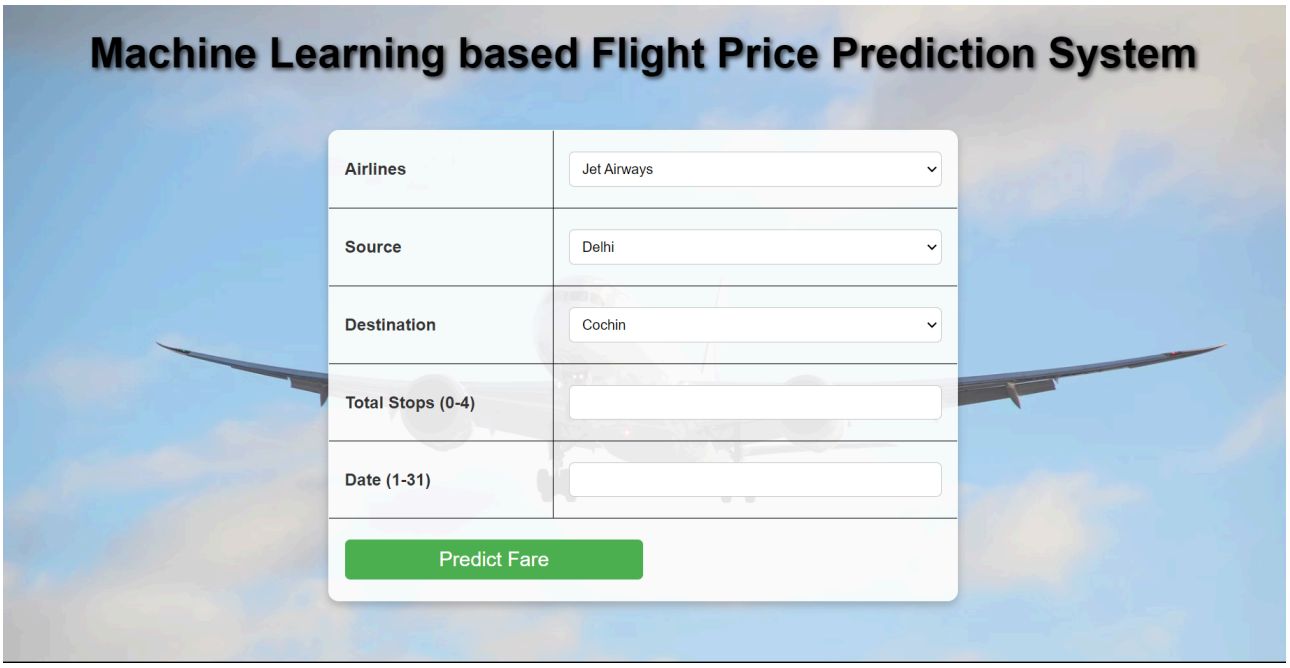


Image = form.html

Flight Price Prediction System webpage Link :-

https://flight-new-one-.glitch.me/

6.4.2 Web server Development using Flask:

APP.PY File

Overview:

- **Developed a Flask web application (app.py) for predicting flight prices based on user input.**
- **Utilized Flask routes to handle different pages and functionalities (`index`, `form`, `fpredict`).**
- **Integrated a `RandomForestRegressor` model trained on flight data for price prediction.**
- **Implemented data preprocessing using `pandas` and `LabelEncoder` from `scikit-learn`.**

Requirements.txt File

Purpose:

- **Creation: Defined a `requirements.txt` file listing necessary Python packages (`Flask`, `numpy`, `pandas`, `scikit-learn`) to ensure consistent environment setup.**
- **Functionality: Ensures seamless deployment and execution by specifying dependencies essential for data handling, model training, and web server development.**

Start.sh File

Objective:

- **Development: Designed a `start.sh` shell script to initialize and execute the Flask application.**
- **Content: Included commands to set environment variables (`FLASK_APP`, `FLASK_ENV`) and install dependencies from `requirements.txt`.**
- **Execution: Facilitates straightforward deployment and operational consistency across various deployment environments.**

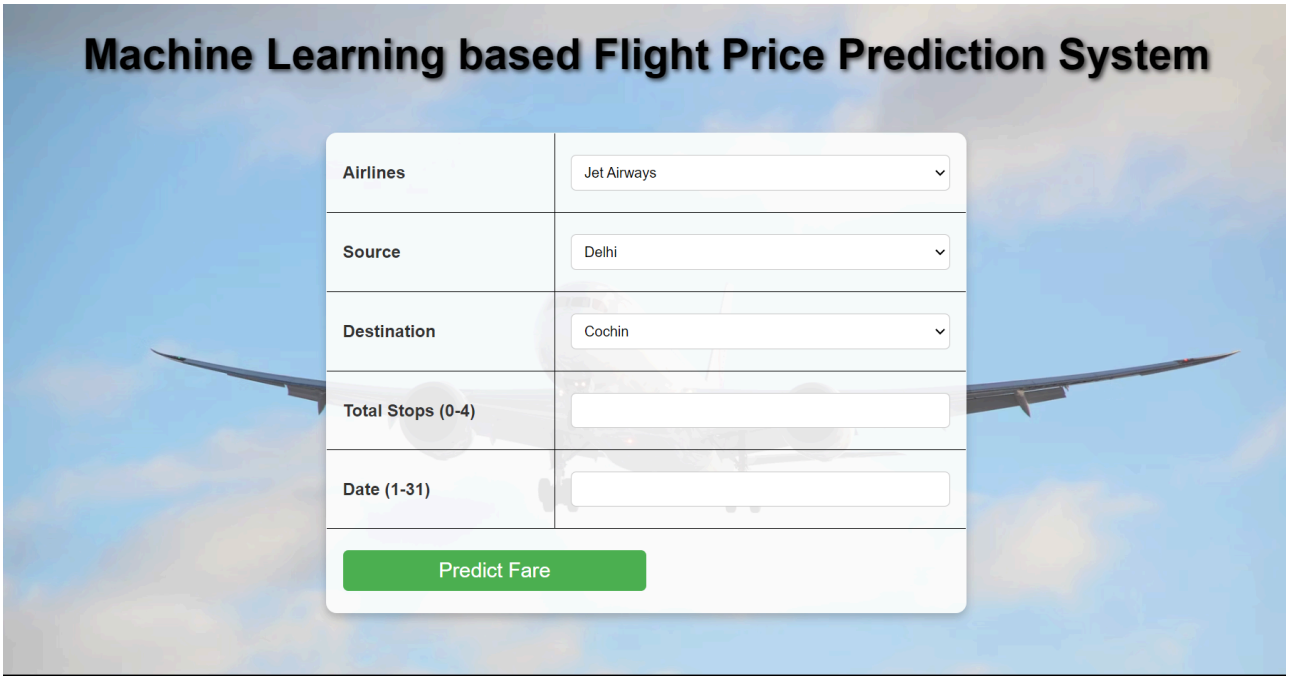
These components collectively facilitate the development, deployment, and execution of the Flight Price Prediction system, ensuring robust functionality and ease of use for end-users interacting with the web application.

6.5 Project walkthrough:

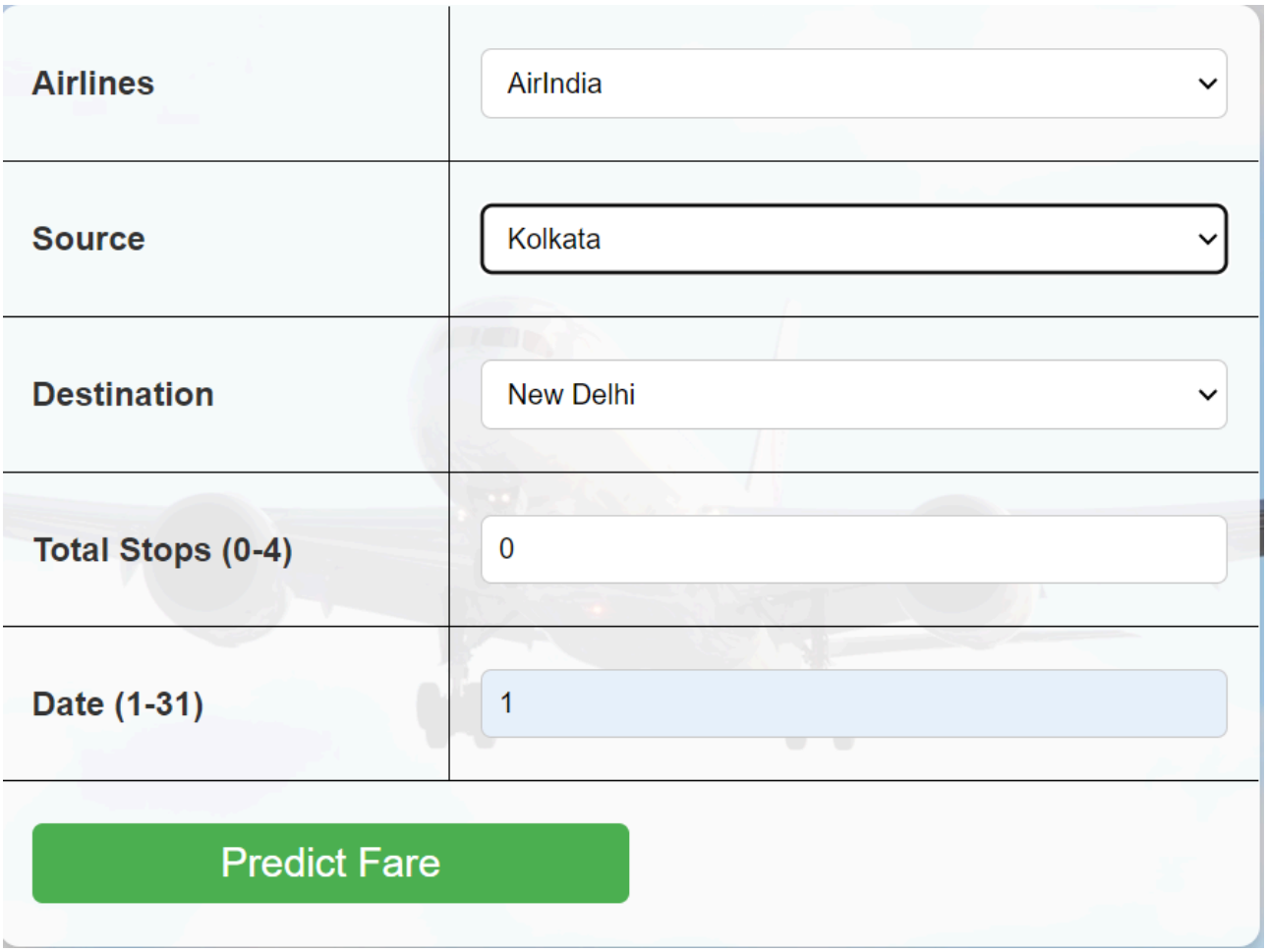
1.Go to our Project website .

https://flight-new-one-.glitch.me/

2.Click on Predict Price.



3.Fill the form and Click on Predict Fare.



4. Check the Approx. Predicted Fare.

5.Fill other inputs if you want to check other predicted fares.

References / Bibliography:

W3schools
Glitch
Colab
Kaggle
Github
NIELIT Notes,Slides
Books and blog available online
Google