

Detail Project Report

Flight Fare Prediction Model

Ricky Mehra

Problem Statement:

The goal of the machine learning project is to create a programme that predicts the fare of the flight by taking bound input from the user like date and time of journey, source and destination, airline company, number stops between the journey etc.

Proposed Solution:

The Airline Flight Fare Prediction is a Flask web application to predict airline flight fares across the Indian cities. The dataset for the project is taken from Kaggle, and it is a time-stamped dataset so, while building the model, extensive pre-processing was done on the dataset especially on the date-time columns to finally come up with a ML model which could effectively predict airline fares across various Indian Cities. The dataset had many features which had to be pre-processed and transformed into new parameters for a cleaner and simple web application layout to predict the fares.

The various independent features in the dataset were:

Airline: The name of the airline.

Date_of_Journey: The date of the journey

Source: The source from which the service begins.

Destination: The destination where the service ends.

Route: The route taken by the flight to reach the destination.

Dep_Time: The time when the journey starts from the source.

Arrival_Time: Time of arrival at the destination.

Duration: Total duration of the flight.

Total_Stops: Total stops between the source and destination.

Additional_Info: Additional information about the flight

Price: The price of the ticket

We have used the regression algorithms to find out the expected price of the flight for the given set of input. We are using the sample dataset from the Kaggle to develop the model and created both instance and batch prediction.

In instance prediction, user can provide input on the HTML webpage and on click of the predict button, the user will get to know the predicted fare of the flight for the given set of inputs.

In batch prediction, user can save .xlsx file in the Input_folder and can find the predicted prices in the Output_folder.

Technical Requirements:

The user can access the webpage and should have the fundamental understanding of providing the input. And rest AWS itself will take care the backend requirements to run all the package that are needed for the process the provided information and to show the results.

Data Gathering:

The data for the current project is taken from Kaggle dataset, the link to the data is:
<https://www.kaggle.com/nikhilmittal/flight-fare-prediction-mh>

Data Description:

There are about 10k+ records of flight information such as airlines, data of journey, source, destination, departure time, arrival time, duration, total stops, additional information, and price.

Tool Used:

- Python 3.8 is used while creating the environment.
- VS Code is used as IDE.
- AWS is used for deployment.
- HTML is used for developing the webpage for the instance prediction.
- GitHub is used as code repository.

Data Pre-processing:

- Initiating the pr-processing by removing the missing rows from the data.
- Removing the duplicate records from the data
- Splitting date, time and duration features and converting them into integers.
- Encoding the categorical data into integers using respective dictionaries.
- Saving the dictionaries to encode the input values during prediction.

Model trainer:

Pre-processed data has been passed to various machine learning regression models along with their hyper-parameters provided by GridSearchCV. The RandomForest Regressor outperform among the given regression algorithms.

Model Evaluation:

Currently trained model is evaluated with the previously saved trained model, if available. If currently trained model is better than the previously trained model then the current model and transformer objects are pushed and saved for the future use.

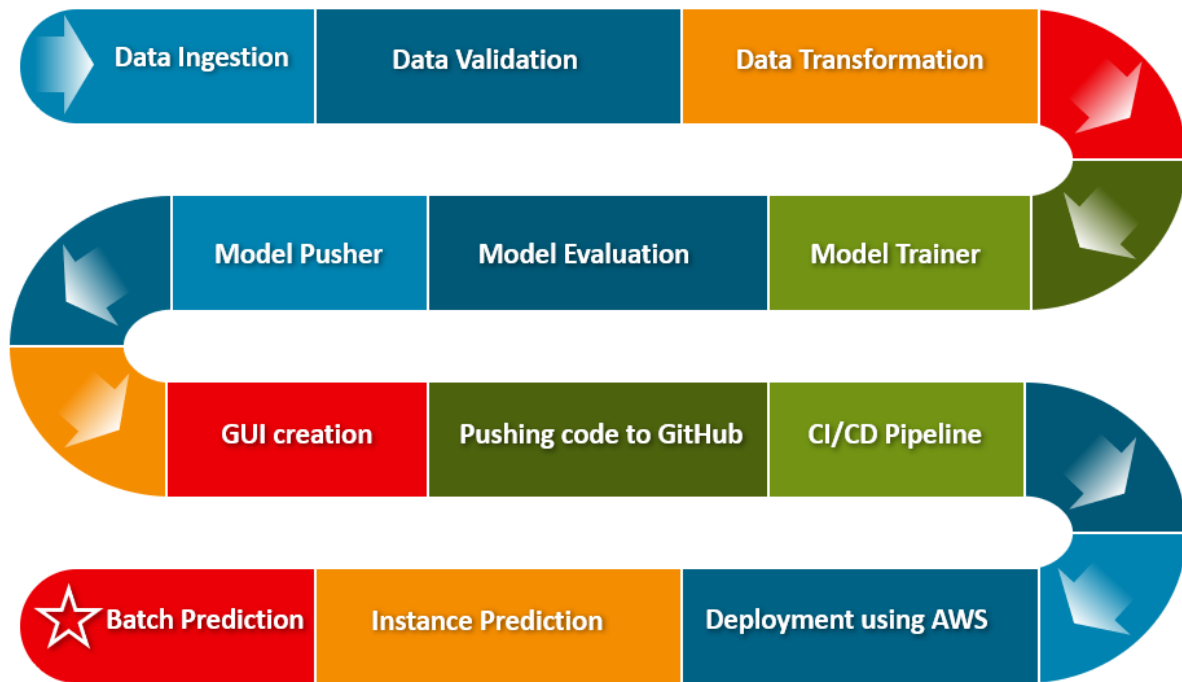
Prediction:

Both the instance and batch prediction can be performed using the code. The app.py file can be used for instance prediction by taking input values from the user through HTML page and the main.py file can be used batch prediction or training the model as per the requirement.

Deployment:

AWS's EC2, ECR and S3 are used to help us to deploy both the instance and batch prediction models.

Modelling and Deployment Process:



Machine Learning Model Stages Flight Fare Prediction

