High Level Design

Flight Fare Prediction Model

Ricky Mehra

Abstract

Flight fare prediction models are machine learning-based systems that aim to predict the cost of a flight based on a set of input features. These models can be used to help users make informed decisions about their travel plans by providing accurate and up-to-date information about flight prices.

The primary goal of a flight fare prediction model is to provide a tool that can help users save money on their travel expenses by identifying the best deals and cheapest flights. In this model, we propose a high-level design of a flight fare prediction model that utilizes historical flight data to train a machine learning regression model. The high-level design includes data ingestion, data validation, data pre-processing, model builder, model evaluation, deployment and user interface design. The proposed model will be evaluated using metrics such as mean-absolute-error (MAE).

The deployment plan includes infrastructure and process for updating the model with new data. This model can be a valuable tool for individual travellers looking for the best flight deals within India as per the given data.

Introduction

Why this HLD document?

The main purpose of this HLD document is to feature the required overview of the project and provide the outline of the different stages of machine learning end-to-end project along with its deployment.

The HLD describe:
- The design aspects and define them in detailed.
- The user interface being implemented.
- The hardware and software interfaces.
- The performance requirements.
- Include design features and architectural design of the project.
- List and describe the non - functional attributes like:
  1. Security
  2. Reliability
  3. Maintainability
  4. Portability
  5. Reusability
  6. Resource
  7. Utilization

Scope

The HLD documentation presents the structure of the project, such as architectural design, application flow and database design. The HLD uses non-technical terms to technical terms that can be understood to the administrator of the system.

Definitions

Terms	Descriptions
FFP	Flight Fare Prediction
MongoDB	MongoDB is used to store and retrieve dataset.
VS Code	VS Code stands for Visual Studio code and it is an IDE for the development of the python based project.
AWS	AWS is Amazon Web Services that enables developers to build, run, and operate applications entirely in the cloud.

General Description

Problem Statement

The flight fare prediction problem involves using historical flight data to predict the future cost of a flight. This data can include factors such as the date of the flight, the departure and arrival airports, the airline, and the class of service. The goal of the prediction is to provide accurate information on flight costs to consumers in order to assist with flight planning and pricing decisions.

Proposed Solution

We are using the machine learning regression model to predict the price of flights for the given set of cities. Implementing the ML model using modular approach and saving the latest robust model with for the current dataset. The proposed model will predict the prices using batch file.

Further Improvements

We have used various regression algorithms with their hyperparameter to find out the robust model as per the given dataset moreover in the training pipeline the code will evaluate the performance of the current model with the previously saved model, if available.

Technical Requirements

As technical requirements, we don't need any specialized hardware for virtualization of the application. The user should have a device that has the access to the web and the fundamental understanding of providing the input. For the backend, we need a AWS to run all the required packages to process the input and predict the desired output.

Data Requirements

The data requirements totally supported the matter statement and also the dataset is accessible on the Kaggle within the file format of (.xlsx). Because the main theme of the project is to induce the expertise of real time issues, we have a tendency to transform the information into the MongoDB database and the csv format.
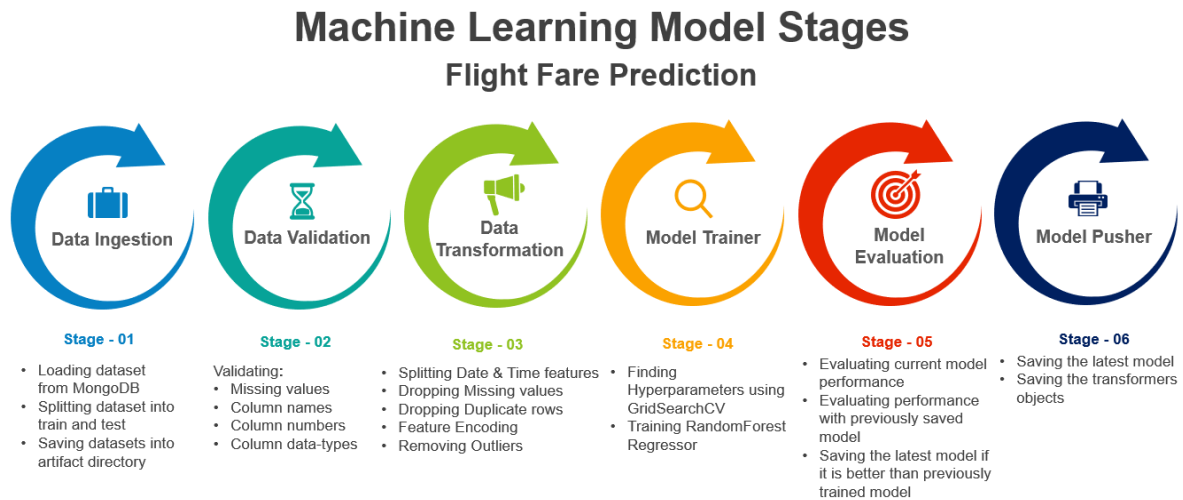
Tools Used

- Python 3.8 is used while creating the environment and libraries like NumPy, Pandas, Scikit-learn, Pymongo, Exceptions and logger are used for developing the model.
- VS code is used to development the modular model.
- MongoDB is used to store and retrieve the data.
- GitHub is employed for continuous deployment.
- AWS is used for the deployment.

Assumptions

The objective of the project is to implement the regression model to predict the price of the new dataset. It assumed that all aspects of this project have the ability to work together in the way it is supposed to be.

Design Flow

Machine Learning stages:

Machine Learning Model Stages
Flight Fare Prediction

Stage - 01	Stage - 02	Stage - 03	Stage - 04	Stage - 05	Stage - 06
Data Ingestion	Data Validation	Data Transformation	Model Trainer	Model Evaluation	Model Pusher
• Loading dataset from MongoDB • Splitting dataset into train and test • Saving datasets into artifact directory	Validating: • Missing values • Column names • Column numbers • Column data-types	• Splitting Date & Time features • Dropping Missing values • Dropping Duplicate rows • Feature Encoding • Removing Outliers	• Finding Hyperparameters using GridSearchCV • Training RandomForest Regressor	• Evaluating current model performance • Evaluating performance with previously saved model • Saving the latest model if it is better than previously trained model	• Saving the latest model • Saving the transformers objects

Logging

All events are logged into the log file as per the execution flow is created and it also captures the error or exceptions with reason and timestamp. This helps the developer to debug the system bugs and rectify the error.

Exception Handling

In logging, each time an error or an exception occurs, the event is logged into the log file with reason and timestamp.

Performance Evaluation

Reusability

The code has been written as per the industry standard and the components used have the ability to be reused with ease.

Application Compatibility

The different parts of the system are communicating or using Python as an interface between them. All the components have its own tasks to perform and it is the job of a

Deployment

The model has been deployed using Amazon web services (AWS) components like ECR, EC2 and S3 bucket.