

Low Level Design

Flight Fare Prediction Model

Ricky Mehra

Introduction

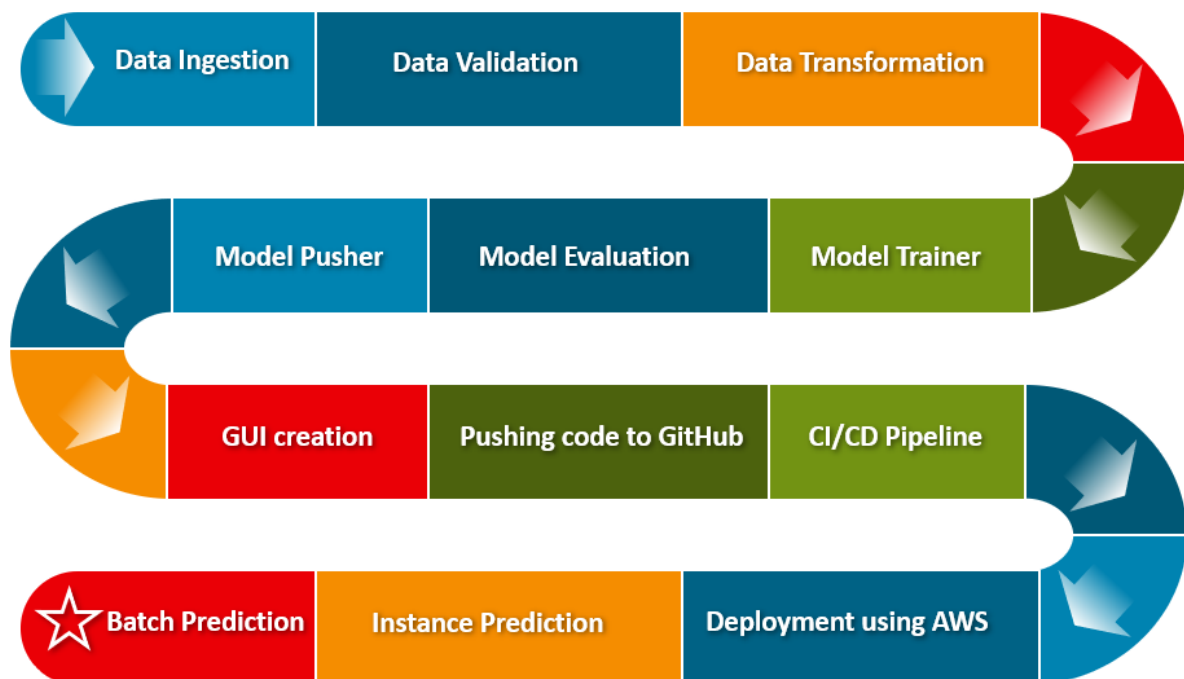
What is Low-Level design document

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

Scope

Low-level design (LLD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work.

Project Architecture:



Architecture Description

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.