Low Level Design (LLD)

FLIGHT PRICE PREDICTION

Revision Number: 1.0

Last date of revision: 13TH JAN 2023

SIRIDI NATH PEDDINA

Contents

Document Version Control

Date Issued	Version	Description	Author
13-01-2023	1.0	First Version of Complete LLD	Siridi nath

1 Introduction

1.1 Why this Low-Level Design Document?

The goal of the Low-level design document (LLDD) is to give the internal logic design of the actual program code for the Food Sales Analysis dashboard. LLDD describes the class diagrams with the methods and relations between classes and programs specs. It describes the modules so that the programmer can directly code the program from the document.

1.2 Scope

Low-level design (LLD) is a component-level design process that follows a step- by-step refinement process. The 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.

1.3 Project Introduction

Travelling through flights has become an integral part of today's lifestyle as more and more people are opting for faster traveling options. The flight ticket prices increase or decrease every now and then depending on various factors like timing of the flights, destination, and duration of flights. Various occasions such as vacations or festive seasons. Therefore, having some basic idea of the flight fares before planning the trip will surely help many people save money and time. In the proposed system a predictive model will be created by applying machine learning algorithms to the collected historical data of flights. This system will give people the idea about the trends that prices follow and also provide a predicted price value which they can refer to before booking their flight tickets to save money. This kind of system or service can be provided to the customers by flight booking companies which will help the customers to book their tickets according.

2. General Description

1.4 Problem Statement & Product Perspective

Flight ticket prices can be something hard to guess, today we might see a price, check out the price of the same flight tomorrow, it will be a different story. We might have often heard travelers saying that flight ticket prices are so unpredictable. As data scientists, we are goanna prove that given the right data anything can be predicted. Here you will be provided with prices of flight tickets for various airlines between the months of March and June of 2019 and between various cities.

1.5 Tools used

- Python programming language.
- Libraries such as Pandas, Numpy, Scikit-Learn, Matplotlib
- Framework: Flask
- IDE: Jupyter Notebook, PyCharm.
- Cloud service: Heroku
- CI/CD pipeline : Heroku
- Git and Github.

- Pycharm and Jupyter notebook is used as IDE.
- For Visualization of the plots Matplotlib is used.
- Heroku is used for deployment of the model.
- Front-end development is done using HTML/CSS.
- Python Flask is used for backend development.
- Heroku is used for CI/CD pipeline.
- GitHub is used for version control system

1.6 Constraints

MLOPs on the cloud must be fully automated in consideration of continuous integration, continuous deployment with retraining approach of model, and archiving the data over time. Users can easily use the application and not needed to know any of the workings.

2 Dataset Information

- The dataset consists of 10683 rows and 11 columns.
- Dateset – https://www.kaggle.com/datasets/nikhilmittal/flight-fare-prediction-mh/

3 Architecture

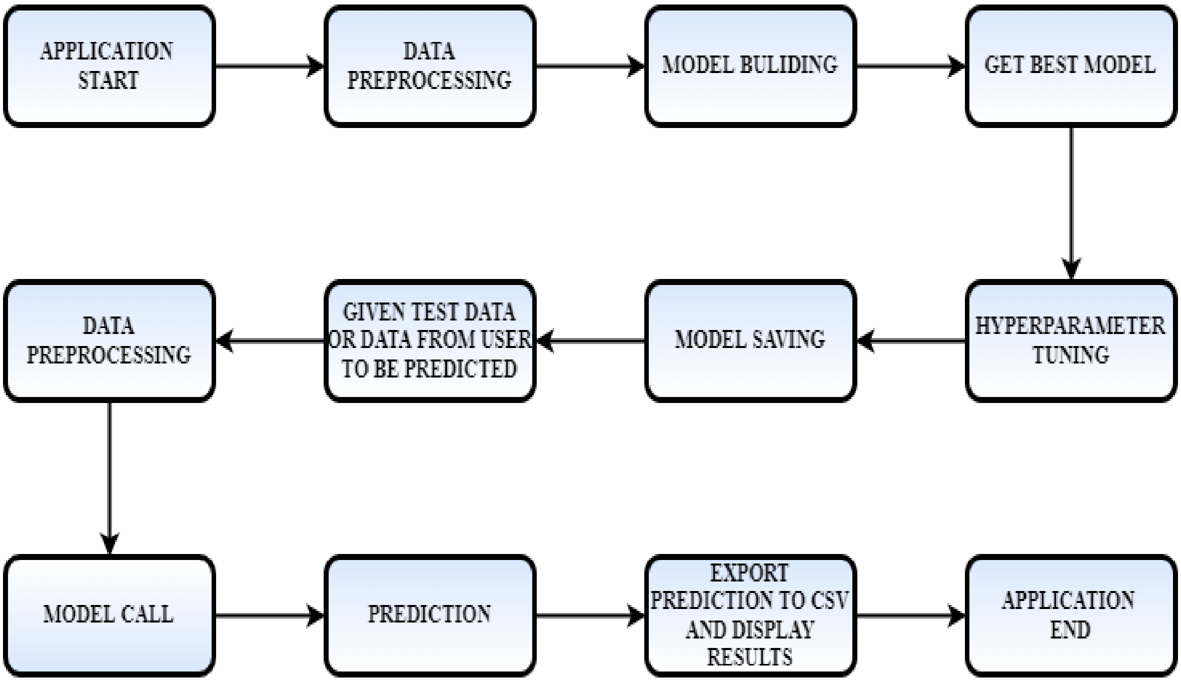

Figure 1: Process flow

3.1 Architecture Description

1. Raw Data Collection

The Data set was taken from kaggle Provided Project Description Document.

2. Data Pre-Processing

Before building any model, it is crucial to perform data pre-processing to feed the correct data to the model to learn and predict. Model performance depends on the quality of data fed to the model to train. This Process includes.

a) Handling Null/Missing Values
b) Outliers Detection and Removal

3. Data Cleaning

Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a data set.

a) Remove duplicate or irrelevant observations

b) Filter unwanted outliers

c) Renaming required attributes

4. Exploratory Data Analysis (EDA)

Exploratory Data Analysis refers to the critical process of performing initial investigations on data to discover patterns, spot anomalies, test hypothesis and to check assumptions with the help of summary statistics and graphical representations.

5. Reporting

Reporting is a most important and underrated skill of a data analytic field. Because being a Data Analyst you should be good in easy and self-explanatory report because your model will be used by many stakeholders who are not from technical background. a) High Level Design Document (HLD)

b) Low Level Design Document (LLD)

c) Architecture

d) Wireframe

e) Detailed Project Report

f) Power Point Presentation

6. Modelling

Data Modelling is the process of analyzing the data objects and their relationship to the other objects. It is used to analyze the data requirements that are required for the business processes. The data models are created for the data to be stored in a database. The Data Model's main focus is on what data is needed and how we have to organize data rather than what operations we have to perform.

7. Deployment

The final model is deployed on Heroku by github.

3.2 Error Handling

The system should identify the errors encountered, an explanation will be displayed as to what went wrong? An error will be defined as anything that falls outside the normal and intended usage.

3.3 Optimization

Data strategy derives performance:

1. Filling missing values.
2. Replacing outliers.
3. Creating new features from datetime feature.
4. Removing correlated features by checking VIF score.
5. Validating score.
6. Hyperparameter tuning
7. Validating score again

3.4 Reusability

The code written and the components used should have the ability to be reused with no problems.

3.5 Application compatibility

The different components for this project will be using Python as an interface between them. Each component will have its task to perform, and it is the job of Python to ensure the proper transfer of information.

3.6 Deployment

4 Conclusion

In this projects, The flight fare prediction will predict the worth of the trained knowledge set within the rule. Therefore, the user will recognize the approximate value for his or her journey.