Low Level Design Document (LLD)

Flight Fare Prediction

An airplane flying in the sky

Description automatically generated

Project Member: Rohit Atnoor

Mentor: Ineuron

Latest Version: 1.7

Last Data of Revision: 19/09/2023

**Document Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Version | Description | Author |
| 15/07/2023 | 1.0 | Initial Project setup, data extraction setup. | Rohit Atnoor |
| 25/07/2023 | 1.1 | Setting up the data ingestion pipeline | Rohit Atnoor |
| 30/07/2023 | 1.2 | Creating of Data Transformation pipeline. | Rohit Atnoor |
| 03/08/2023 | 1.3 | Creation of prediction pipeline. | Rohit Atnoor |
| 08/08/2023 | 1.4 | Initial End to End prediction pipeline. | Rohit Atnoor |
| 12/08/2023 | 1.5 | Deploying of HTML Files. | Rohit Atnoor |
| 20/08/2023 | 1.6 | Data Addition | Rohit Atnoor |
| 10/09/2023 | 1.7 | Final End to End Deployable code. | Rohit Atnoor |

**Contents**

1. Introduction

1.1. What is Low-Level design document?

1.2. Scope

2. Architecture

3. Architecture Description

3.1. Data Description

3.2. Data Transformation

3.3. Data Pre-processing

3.4. Model Selection

3.5. Model Retraining

3.6. Prediction

3.7. Deployment

4. Unit Test Cases

1. **Introduction** 
   1. **What is Low Level Design Document**

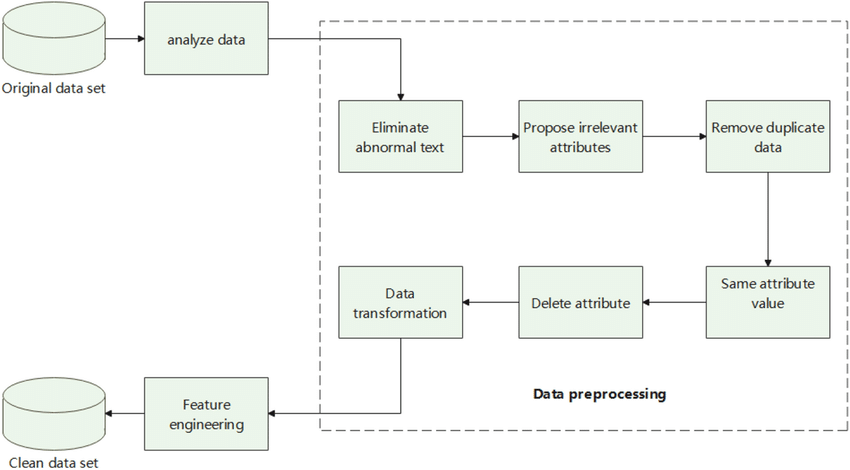
The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for Flight Fare Prediction System. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly code the program from the document

* 1. **Scope**

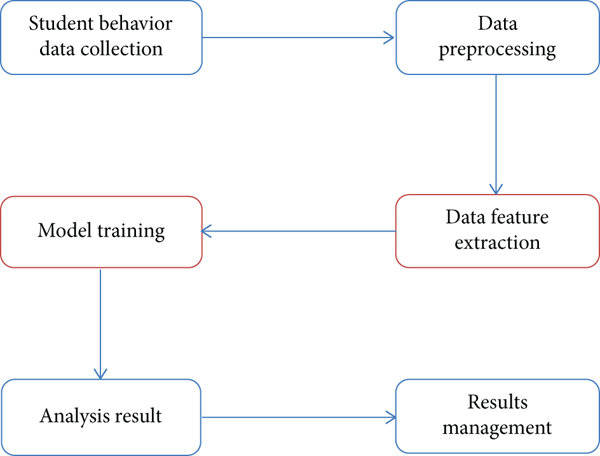
Low-level design (LLD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement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

**2. Architecture**

Initial Data Cleaning Flow Chart



Medel Training and Prediction Flow

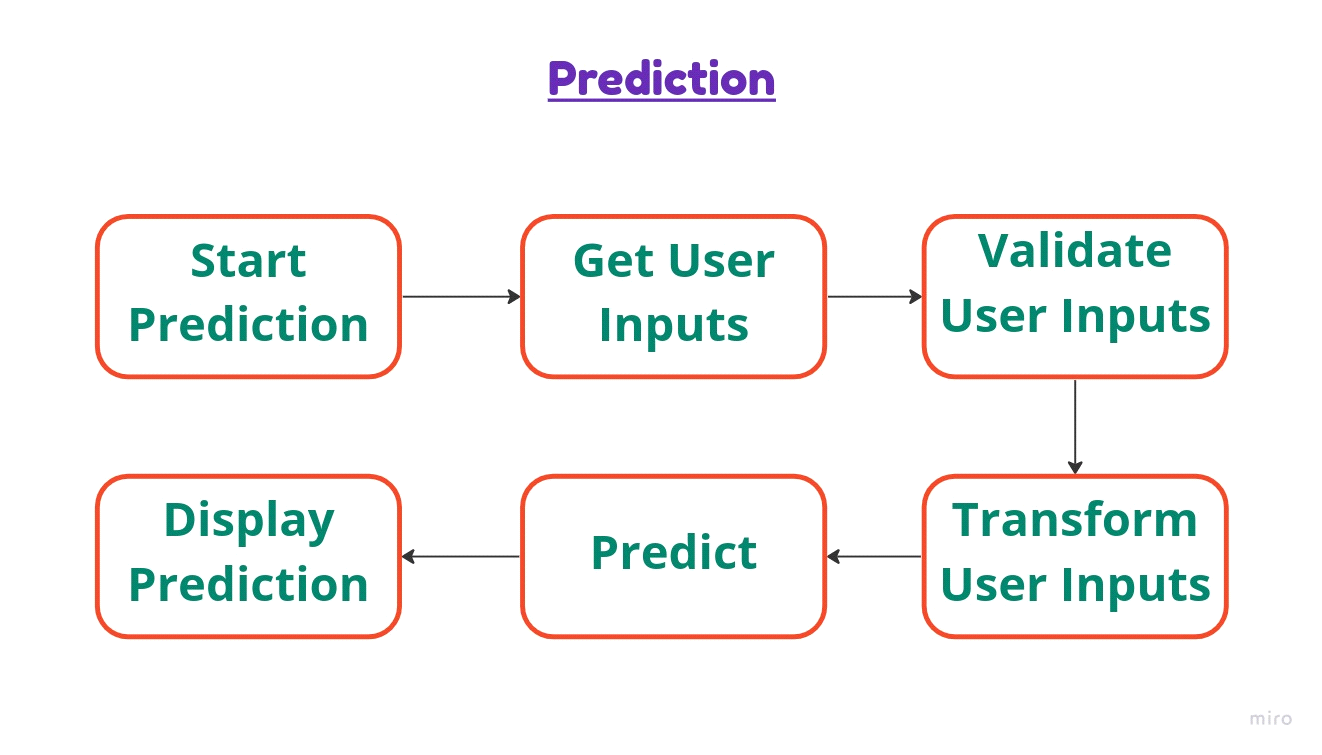


Model Deployment

Best Model Selection

Clean Data Set

Prediction Flow



Display Prediction value

Predict the Flight price

Transform User input

Data validation of user input

Web Page User UI

User Inputs

**3. Architecture Description**

**3.1 Data Description**

Price is the Dependent column for this project because we are predicting the price of the flight.

1. Airline : Company Name.

2. Date\_of\_Journey : Date of travelling.

3. Source : Starting Point or City.

4. Destination : Travelling destination city.

5. Route : Route include the path or all the city between source and destination.

First city is source, last city is destination, other cities between route.

6. Dep\_Time : The time of departure in 24 hours time format.

7. Arrival\_Time : Reaching time to destination in 24 hours format.

8. Duration : Total time taken to reach the destination from source.

9. Total\_Stops : Number of Stops between source and destination.

10. Additional\_Info : Extra Information about the flight.

11. Price : The total amount for travelling from source to destination in Rupees.

**3.2 Data Transformation**

In the Transformation Process, we will convert our original dataset which is in JSON format to CSV Format. And will merge it with the Scrapped dataset.

**3.3 Data Pre-Processing**

Data Pre-processing steps we could use are Null value handling, stop words removal, punctuation removal, Tokenization, Lemmatization, TFIDF, Imbalanced data set handling, Handling columns with standard deviation zero or below a threshold, etc.

**3.4 Model Selection**

After model training we will find the best model for each cluster. For each cluster, algorithms will be passed with the best parameters derived from Grid-Search. We will calculate the AUC scores for models and select the model with the best score. Similarly, the models will be selected for each cluster. All the models for every cluster will be saved for use in Recommendation.

**3.5 Model Retraining**

Whenever a new or updated data is available in the database, we can initiate our Model Retaraining pipeline manually. At the end of this process, new models will get stored in the same directory but in a new folder indicating a new versions of models. As soon as we commit these changes to Github, the new model will get deployed automatically.

**3.6 Prediction**

This is the frontend of this project. Any user can use the service with the deplyment url. Once the user provides all the required inputs in the

**3.7 Deployment**

We will be deploying the model to Azure Web Services. This is a workflow diagram for the Recipe Recommendation.

**4 Unit Test Cases**

|  |  |  |
| --- | --- | --- |
| **Test case Description** | **Pre-Requisite** | **Expected Result** |
| Verify whether the Application URL is  accessible to the user | 1. Application URL  should be defined | Application URL should be  accessible to the user |
| Verify whether the Application loads  completely for the user when the URL  is accessed | 1. Application URL  is accessible  2. Application is  deployed | The Application should load  completely for the user when the  URL is accessed |
| Verify whether user is able to see input  fields on logging in | 1. Application is  accessible  2. User is signed up  to the application 3. User is logged in  to the application | User should be able to see input  fields on logging in |
| Verify whether user is able to edit all  input fields | 1. Application is  accessible  2. User is signed up  to the application 3. User is logged in  to the application | User should be able to edit all input  fields |
| Verify whether user gets Submit  button to submit the inputs | 1. Application is  accessible  2. User is signed up  to the application 3. User is logged in  to the application | User should get Submit button to  submit the inputs |
| Verify whether user is presented with  recommended results on clicking  submit | 1. Application is  accessible  2. User is signed up  to the application 3. User is logged in  to the application | User should be presented with  recommended results on clicking  submit |