Low Level Design (LLD)

FLIGHT FARE PREDICTION

Document Version: 0.3

Last Revised Date: 26/06/2023

VIKAS SHARMA

Document Version Control:

Version	Date	Author	Description
0.1	22/06/2023	Vikas Sharma	Introduction Architecture, Document Content, Version Control and Unit Test cases to be added
0.2	23/06/2023	Vikas Sharma	Data Pre-processing, Model Building
0.3	26/06/2023	Vikas Sharma	Deployment

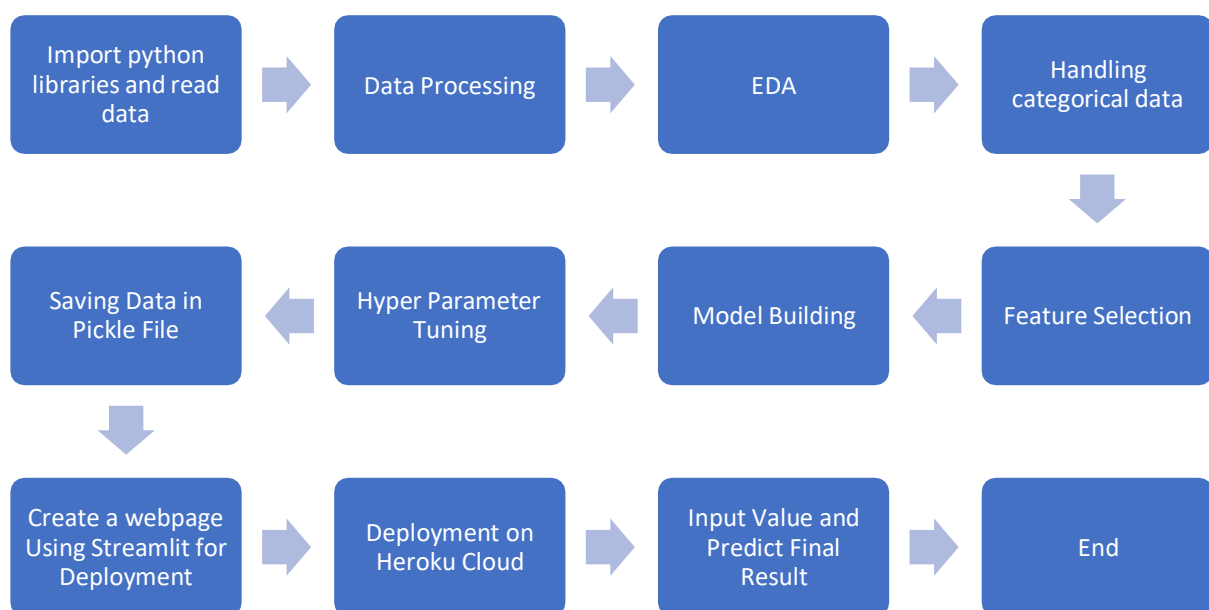# Contents

1. Introduction

1.1.   What is Low-Level design document?

The main purpose of this LLD documentation is to feature the required details of the project and supply the outline of the machine learning model and also the written code. This additionally provides the careful description on however the complete project has been designed end-to-end.

1.2. Scope

Low-level design (LLD) is a component-level design process that follows a step-bystep 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

Import python libraries and read data	→	Data Processing	→	EDA	→	Handling categorical data

↓

Saving Data in Pickle File	←	Hyper Parameter Tuning	←	Model Building	←	Feature Selection

↓

Create a webpage Using Streamlit for Deployment	→	Deployment on Heroku Cloud	→	Input Value and Predict Final Result	→	End

3. Architecture Description

This project is to make associate interface for the user to grasp their approximate flight price ticket worth, additionally to the present, in would like of obtaining the important time project expertise we have a tendency to square measure mercantilism the gathered information into our own information then begin the project from the scratch.

3.1. Data Gathering

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

3.2. 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. A glance of the dataset is shown below.

	Airline	e_of_Jour	Source	Destination	Route	Dep_Time	rrival_Tim	Duration	Total_Stop	ditional_Ir	Price
2	IndiGo	24/03/201	Banglore	New Delhi	BLR → DEL	22:20	01:10 22 N	2h 50m	non-stop	No info	3897
3	Air India	1/05/2019	Kolkata	Banglore	CCU → IXF	05:50	13:15	7h 25m	2 stops	No info	7662
4	Jet Airway	9/06/2019	Delhi	Cochin	DEL → LKC	09:25	04:25 10 J	19h	2 stops	No info	13882
5	IndiGo	12/05/201	Kolkata	Banglore	CCU → NA	18:05	23:30	5h 25m	1 stop	No info	6218
6	IndiGo	01/03/201	New Delhi	BLR → NA	16:50	21:35	4h 45m	1 stop	No info	13302
7	SpiceJet	24/06/201	Kolkata	Banglore	CCU → BLI	09:00	11:25	2h 25m	non-stop	No info	3873
8	Jet Airway	12/03/201	Banglore	New Delhi	BLR → BOI	18:55	10:25 13 N	15h 30m	1 stop	In-flight m	11087
9	Jet Airway	01/03/201	Banglore	New Delhi	BLR → BOI	08:00	05:05 02 N	21h 5m	1 stop	No info	22270
10	Jet Airway	12/03/201	Banglore	New Delhi	BLR → BOI	08:55	10:25 13 N	25h 30m	1 stop	In-flight m	11087
11	Multiple c	27/05/201	Delhi	Cochin	DEL → BOI	11:25	19:15	7h 50m	1 stop	No info	8625
12	Air India	1/06/2019	Delhi	Cochin	DEL → BLF	09:45	23:00	13h 15m	1 stop	No info	8907
13	IndiGo	18/04/201	Kolkata	Banglore	CCU → BLI	20:20	22:55	2h 35m	non-stop	No info	4174
14	Air India	24/06/201	Chennai	Kolkata	MAA → C(	11:40	13:55	2h 15m	non-stop	No info	4667
15	Jet Airway	9/05/2019	Kolkata	Banglore	CCU → BO	21:10	09:20 10 N	12h 10m	1 stop	In-flight m	9663
16	IndiGo	24/04/201	Kolkata	Banglore	CCU → BLI	17:15	19:50	2h 35m	non-stop	No info	4804
17	Air India	3/03/2019	Delhi	Cochin	DEL → AM	16:40	19:15 04 N	26h 35m	2 stops	No info	14011
18	SpiceJet	15/04/201	Delhi	Cochin	DEL → PN(	08:45	13:15	4h 30m	1 stop	No info	5830
19	Jet Airway	12/06/201	Delhi	Cochin	DEL → BOI	14:00	12:35 13 J	22h 35m	1 stop	In-flight m	10262

3.3. Tool Used

- Python 3.9 is employed because the programming language and frame works like numpy, pandas, sklearn and alternative modules for building the model.

- PyCharm is employed as IDE.

- For visualizations seaborn and components of matplotlib are getting used

- For information assortment prophetess info is getting used version management.

- Jupyter Notebook is used for EDA and for model building.

3.4. Data Pre-processing

Steps performed in pre-processing are:

- First the info sorts square measure being checked and located solely the value column is of sort number.

- Checked for null values as there square measure few null values, those rows square measure born.

- Converted all the desired column into the date time format.

- Performed one-hot cryptography for the desired columns.

- Scaling is performed for needed information.

- And, the info is prepared for passing to the machine learning formula

3.4. Model Building

The pre-processed information is then envisioned and every one the specified insights are being drawn. though from the drawn insights, the info is at random unfold however still modelling is performed with completely different machine learning algorithms to form positive we tend to cowl all the chances. and eventually, for sure random forest regression performed well and any hyperparameter calibration is finished to extend the model's accuracy.

3.5. Data from User

The data from the user is retrieved from the created HTML web page.

3.6. Data Validation

The data provided by the user is then being processed by app.py file and validated. The validated data is then sent for the prediction.

3.7. Rendering Result

The data sent for the prediction is then rendered to the web page.

3.8. Deployment

The tested model is then deployed to cloud platform. So, users can access the project from any internet devices.

3.9. Unit Tests

Test Case Description	Pre-Requisites	Expected Results
Verify whether the User Interface URL is accessible to the user.	1. User Interface URL should be defined.	User Interface URL should be accessible to the user.
Verify whether the User Interface loads completely for the user when the URL is accessed.	1. User Interface URL is accessible. 2. User Interface is deployed.	The User Interface should load completely for the user when the URL is accessed.
Verify whether user is able to edit all input fields.	1. User Interface is accessible.	User should be able to edit all input fields.