

ARCHITECTURE

FLIGHT FARE

ESTIMATOR

| | |
|-------------------|-------------|
| Written By | Suraj Joshi |
| Document Version | 1.0 |
| Last Revised Data | 25 Nov 2021 |

1 Document Version Control:

| Version | Date | Author | Comments |
|---------|------------|-------------|----------|
| 1.0 | 25/11/2021 | Suraj Joshi | |
| | | | |
| | | | |
| | | | |
| | | | |

Table of Contents

| | |
|--|----|
| 1 Document Version Control: | 2 |
| 1.Introduction..... | 4 |
| 1.1 What is Low-Level design document? | 4 |
| 1.2. Scope..... | 4 |
| 2. Architecture | 5 |
| 3. Dataset | 5 |
| 3.1 Dataset Overview | 5 |
| 4. Logging..... | 7 |
| 5. Database | 7 |
| 6. Deployment | 7 |
| 7. Proposed Solution | 7 |
| 8. Model Training/Validation workflow | 8 |
| 9. User I/O workflow | 9 |
| 10. Test Cases | 10 |

1.Introduction

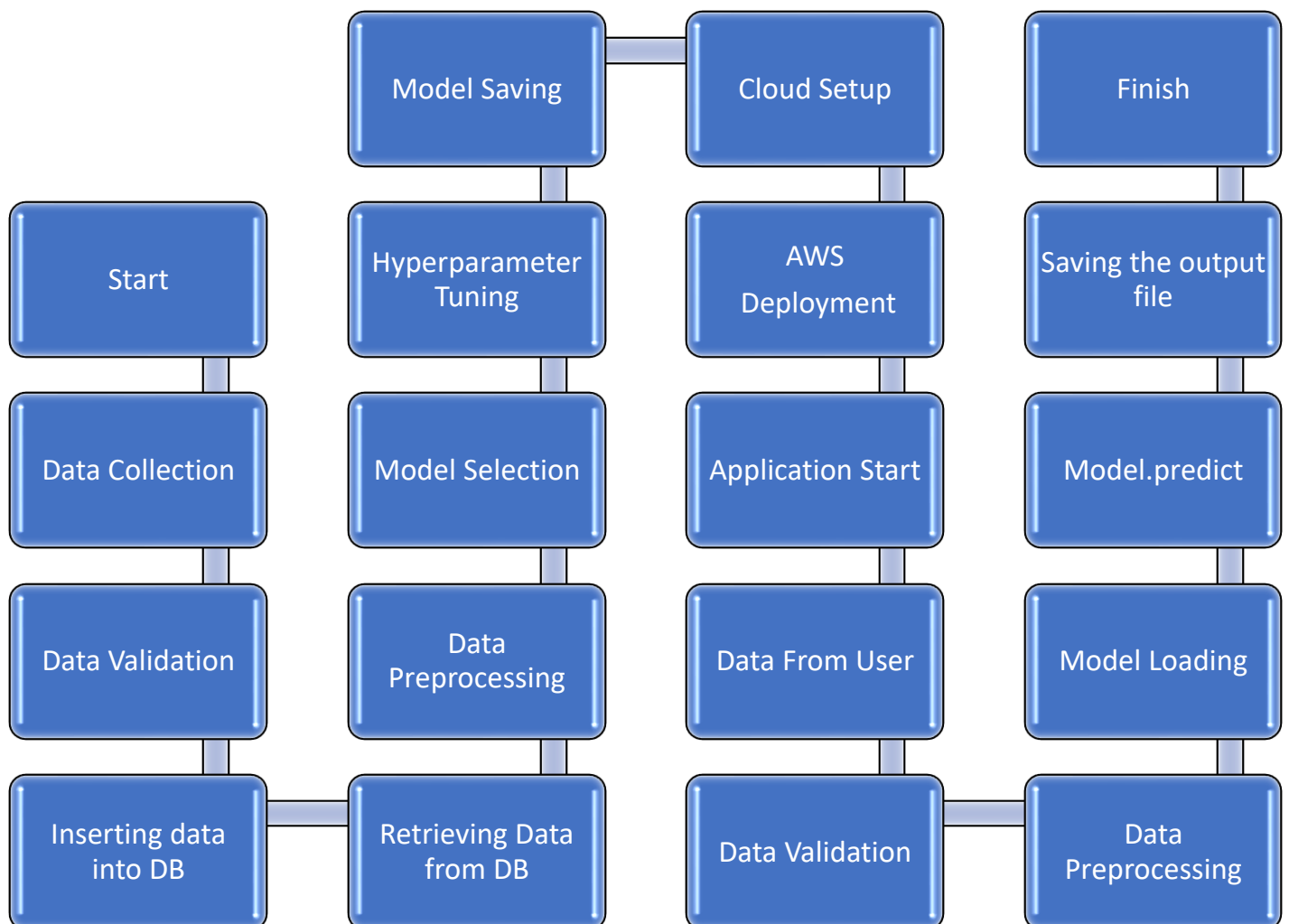
1.1 What is Low-Level design document?

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 estimation 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.

1.2. 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



3. Dataset

3.1 Dataset Overview

The training dataset consists of 12 columns and every column datatype is string except the last one that is “Price” which is a integer datatype.

| A | B | C | D | E | F | G | H | I | J | K | L | M |
|-------|-------------------|-----------------|----------|-------------|-----------------------|----------|------------------|----------|-------------|-----------------|-------|---|
| Id | Airline | Date_of_Journey | Source | Destination | Route | Dep_Time | Arrival_Time | Duration | Total_Stops | Additional_Info | Price | |
| 4317 | Multiple carriers | 18-05-2019 | Delhi | Cochin | DEL → BOM → COK | 08:30 | 19:15 | 10h 45m | 1 stop | No info | 7887 | |
| 3372 | Air Asia | 21-05-2019 | Kolkata | Banglore | CCU → BBI → BLR | 19:55 | 23:30 | 3h 35m | 1 stop | No info | 5162 | |
| 1584 | Jet Airways | 06-06-2019 | Delhi | Cochin | DEL → BOM → COK | 19:15 | 07-06-2021 19:00 | 23h 45m | 1 stop | No info | 14714 | |
| 7034 | IndiGo | 21-05-2019 | Delhi | Cochin | DEL → PNQ → COK | 23:30 | 22-05-2021 04:35 | 5h 5m | 1 stop | No info | 8745 | |
| 9892 | Air India | 03-06-2019 | Delhi | Cochin | DEL → GOI → BOM → COK | 10:55 | 19:15 | 8h 20m | 2 stops | No info | 12173 | |
| 9640 | Multiple carriers | 09-06-2019 | Delhi | Cochin | DEL → BOM → COK | 16:00 | 10-06-2021 01:30 | 9h 30m | 1 stop | No info | 13587 | |
| 9067 | IndiGo | 01-06-2019 | Delhi | Cochin | DEL → MAA → COK | 02:00 | 07:25 | 5h 25m | 1 stop | No info | 5636 | |
| 4830 | IndiGo | 24-03-2019 | Kolkata | Banglore | CCU → HYD → BLR | 15:10 | 19:50 | 4h 40m | 1 stop | No info | 7476 | |
| 2731 | Multiple carriers | 09-05-2019 | Delhi | Cochin | DEL → BOM → COK | 10:20 | 19:00 | 8h 40m | 1 stop | No info | 9794 | |
| 10113 | Jet Airways | 01-03-2019 | Banglore | New Delhi | BLR → BOM → DEL | 07:00 | 13:15 | 6h 15m | 1 stop | No info | 26890 | |
| 5056 | Air India | 18-05-2019 | Delhi | Cochin | DEL → BLR → COK | 09:45 | 23:00 | 13h 15m | 1 stop | No info | 8907 | |
| 6428 | Vistara | 12-06-2019 | Chennai | Kolkata | MAA → CCU | 17:45 | 20:05 | 2h 20m | non-stop | No info | 11982 | |

Testing Data consists of only 10 columns because there will not be two columns first is Id and second is Price. The data type inside the test day for every column is a string.

| A | B | C | D | E | F | G | H | I | J |
|-------------|-----------------|----------|-------------|-----------------------|----------|------------------|----------|-------------|-----------------------------|
| Airline | Date_of_Journey | Source | Destination | Route | Dep_Time | Arrival_Time | Duration | Total_Stops | Additional_Info |
| Jet Airway | 06-06-2019 | Delhi | Cochin | DEL → BOM → COK | 17:30 | 07-06-2021 04:25 | 10h 55m | 1 stop | No info |
| IndiGo | 12-05-2019 | Kolkata | Banglore | CCU → MAA → BLR | 06:20 | 10:20 | 4h | 1 stop | No info |
| Jet Airway | 21-05-2019 | Delhi | Cochin | DEL → BOM → COK | 19:15 | 22-05-2021 19:00 | 23h 45m | 1 stop | In-flight meal not included |
| Multiple ca | 21-05-2019 | Delhi | Cochin | DEL → BOM → COK | 08:00 | 21:00 | 13h | 1 stop | No info |
| Air Asia | 24-06-2019 | Banglore | Delhi | BLR → DEL | 23:55 | 25-06-2021 02:45 | 2h 50m | non-stop | No info |
| Jet Airway | 12-06-2019 | Delhi | Cochin | DEL → BOM → COK | 18:15 | 13-06-2021 12:35 | 18h 20m | 1 stop | In-flight meal not included |
| Air India | 12-03-2019 | Banglore | New Delhi | BLR → TRV → DEL | 07:30 | 22:35 | 15h 5m | 1 stop | No info |
| IndiGo | 01-05-2019 | Kolkata | Banglore | CCU → HYD → BLR | 15:15 | 20:30 | 5h 15m | 1 stop | No info |
| IndiGo | 15-03-2019 | Kolkata | Banglore | CCU → BLR | 10:10 | 12:55 | 2h 45m | non-stop | No info |
| Jet Airway | 18-05-2019 | Kolkata | Banglore | CCU → BOM → BLR | 16:30 | 22:35 | 6h 5m | 1 stop | No info |
| Jet Airway | 21-03-2019 | Delhi | Cochin | DEL → MAA → BOM → COK | 13:55 | 22-03-2021 18:50 | 28h 55m | 2 stops | In-flight meal not included |
| IndiGo | 15-06-2019 | Delhi | Cochin | DEL → HYD → COK | 06:50 | 16:10 | 9h 20m | 1 stop | No info |
| Multiple ca | 15-05-2019 | Delhi | Cochin | DEL → BOM → COK | 09:00 | 19:15 | 10h 15m | 1 stop | No info |
| Jet Airway | 12-03-2019 | Banglore | New Delhi | BLR → BOM → DEL | 05:45 | 10:25 | 4h 40m | 1 stop | No info |
| Jet Airway | 03-06-2019 | Delhi | Cochin | DEL → BOM → COK | 19:15 | 04-06-2021 12:35 | 17h 20m | 1 stop | In-flight meal not included |
| Jet Airway | 06-03-2019 | Banglore | New Delhi | BLR → BOM → DEL | 21:25 | 07-03-2021 08:15 | 10h 50m | 1 stop | No info |
| Multiple ca | 06-06-2019 | Delhi | Cochin | DEL → HYD → COK | 13:15 | 22:30 | 9h 15m | 1 stop | No info |
| Vistara | 24-03-2019 | Kolkata | Banglore | CCU → DEL → BLR | 09:55 | 22:10 | 12h 15m | 1 stop | No info |
| Jet Airway | 12-06-2019 | Delhi | Cochin | DEL → BOM → COK | 19:15 | 13-06-2021 04:25 | 9h 10m | 1 stop | In-flight meal not included |
| Jet Airway | 12-03-2019 | Banglore | New Delhi | BLR → BOM → DEL | 22:55 | 13-03-2021 08:15 | 9h 20m | 1 stop | No info |
| IndiGo | 06-03-2019 | Delhi | Cochin | DEL → BOM → COK | 10:45 | 07-03-2021 01:35 | 14h 50m | 1 stop | No info |
| Jet Airway | 09-05-2019 | Kolkata | Banglore | CCU → BOM → BLR | 20:00 | 10-05-2021 10:05 | 14h 5m | 1 stop | In-flight meal not included |
| Jet Airway | 18-03-2019 | Banglore | New Delhi | BLR → BOM → DEL | 21:25 | 16-03-2021 09:00 | 11h 35m | 1 stop | In-flight meal not included |

4. Logging

Logging is very important to keep track of the activities performed by our application. I have used logging module to do so. All the logs either it is train or test, both will be present inside All_logs folder. Logging helps us in debugging process also so it is mandatory to do.

5. Database

The database I am using is Cassandra db. System needs to store every request into the database and we need to store it in such a way that it is easy to retrain the model as well. The system stores each and every data given by the user or received on request to the database.

6. Deployment

Deployment is done in AWS and it's a production server.

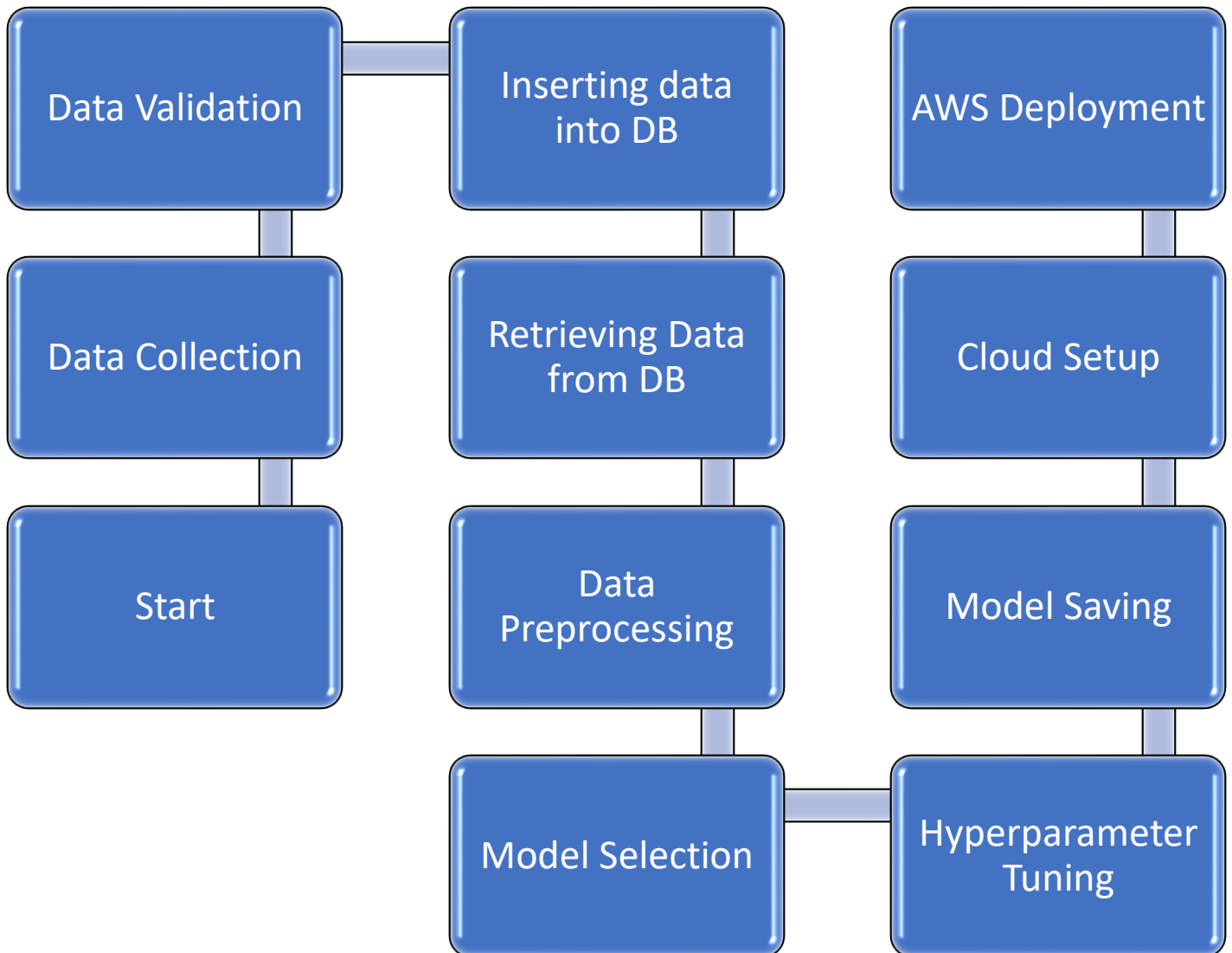


Link: <http://ec2-3-141-18-169.us-east-2.compute.amazonaws.com:5000/>

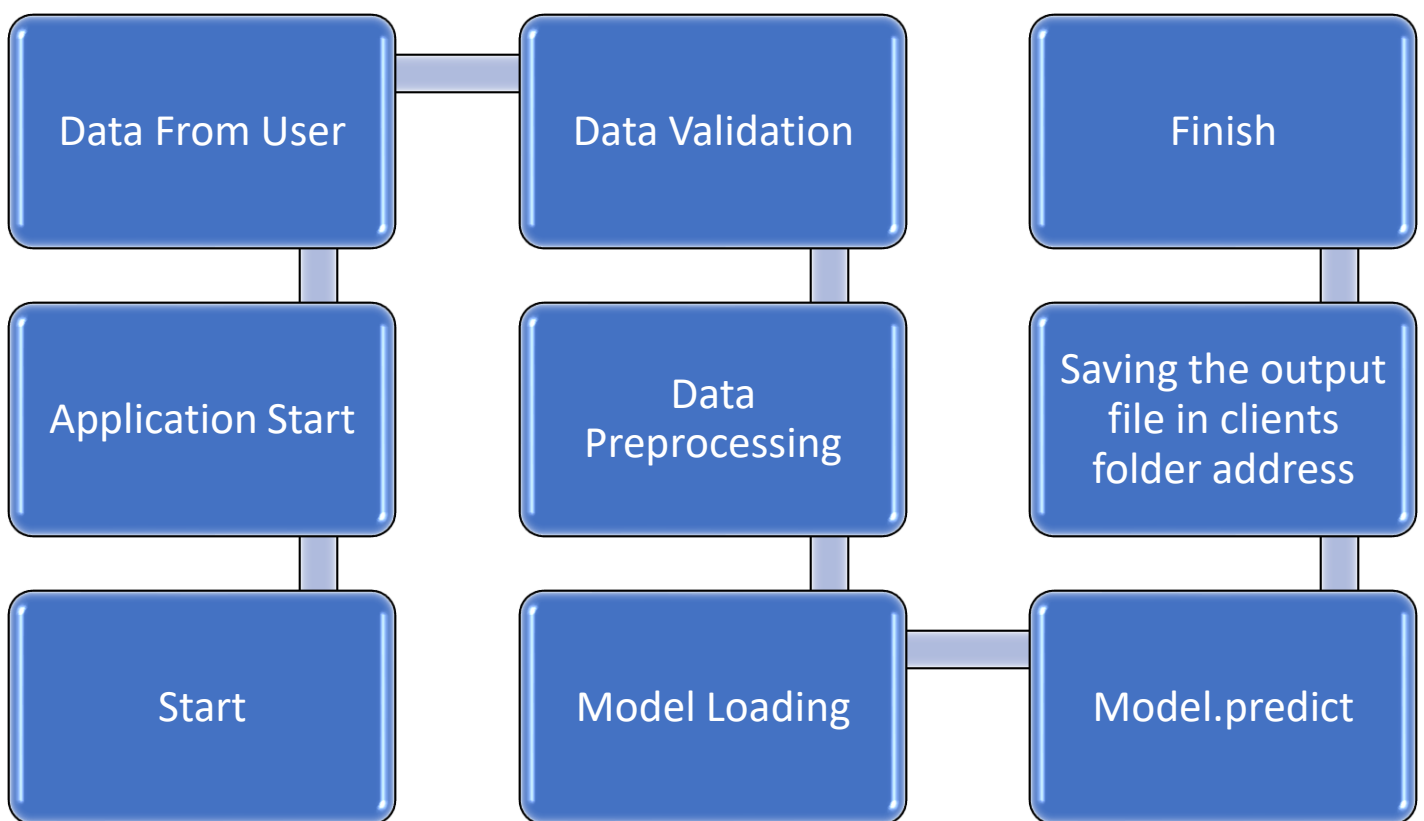
7. Proposed Solution

Solution is very simple here. I am going to build a simple ML model which will be able to predict the flight fare based on the data given. Doing some EDA on the dataset I got to know that xgboost, Random Forest will be the best.

8. Model Training/Validation workflow



9. User I/O workflow



10. Test Cases

Test cases are given below

| 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 Response time of url from backend model. | 1. Application is accessible | The latency and accessibility of application is very faster we got in AWS service. |
| Verify whether user is giving standard input. | 1. Handeled test cases at backends. | User should be able to see successfully valid results. |
| Verify whether user is able to see input fields on logging in | 1. Application is accessible 2. 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 logged in to the application | User should be able to edit all input fields |
| Verify whether user gets Custom File Predict, Default File Predict button to submit the inputs | 1. Application is accessible 2. User is logged in to the application | User should get both buttons to submit the inputs |

LOW LEVEL

| | | |
|--|--|---|
| Verify whether user is presented with recommended results on clicking submit | <ol style="list-style-type: none"> 1. Application is accessible 2. User is logged in to the application | User should be presented with recommended results on clicking submit |
| Verify whether the recommended results are in accordance to the selections user made | <ol style="list-style-type: none"> 1. Application is accessible 2. User is logged in to the application and database | The recommended results should be in accordance to the selections user made |