High Level Design (HLD)

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 HLD	Siridinath

Abstract

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 accordingly.

1 Introduction

1.1 Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) document is to add the necessary detail to the project description to represent a suitable model and coding for application. This document is also intended to help detect contradictions before coding and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

- Present all of the design aspects and define them in detail
- Describe the user interface is implemented
- Describe the hardware and software interfaces
- Describe the performance requirements
- Include design features and the architecture of the project
- List and describe the non-functional attributes like:
    - Security
    - Reliability
    - Maintainability
    - Portability
    - Reusability
    - Application compatibility
    - Resource utilization
    - Serviceability

1.2 Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology stack. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

2 General Description

2.1 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.

2.2 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

2.3 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.

3 Design Details

3.1 Process flow

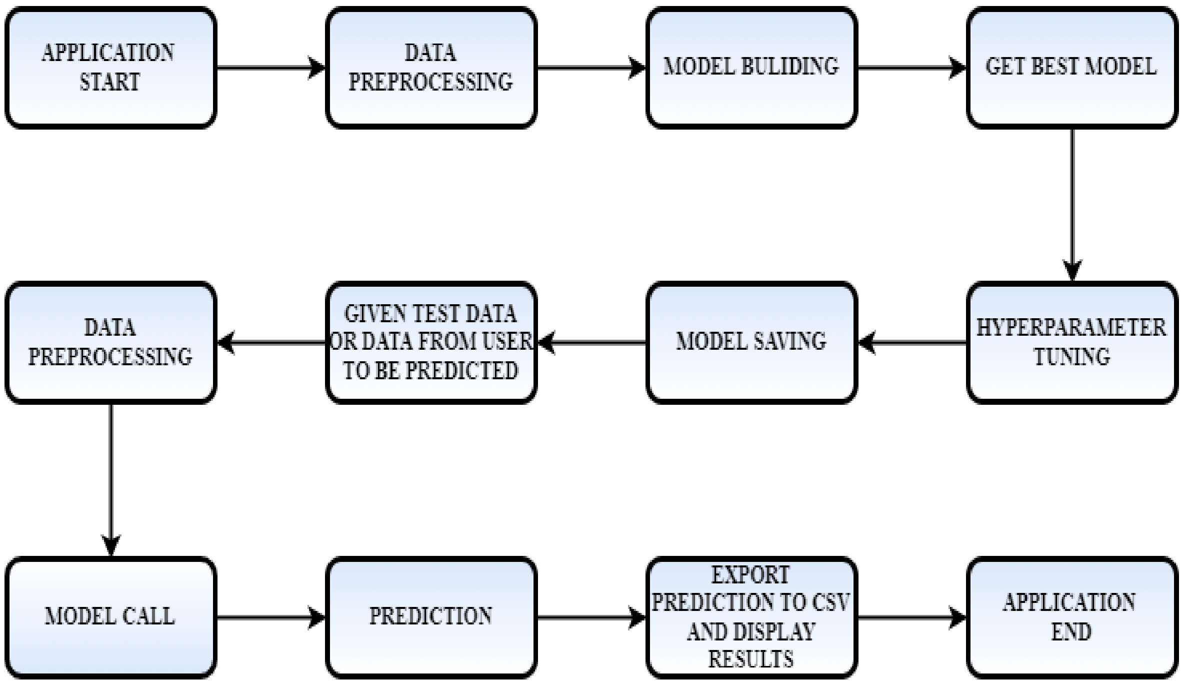

Figure 1: Process flow

3.2 Event log

The system should log every event so that the track of every detail will be known and what process is running currently could be seen.

Initial Step-By-Step Description:

1. The System identifies at what step logging is required.
2. The System should be able to log each and every system flow.
3. Developer can choose logging method. You can choose database logging/ File logging as well.
4. System should not hang even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

3.3 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.

3.4 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

3.5 Reusability

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

3.6 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.

3.7 Deployment

heroku

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