High-Level Design(HLD)

Restaurant Rating Prediction

M V S K Karthik

Document Version Control

Date issued	Version	Description	Author
	1	Initial-HLD-V1	M V S K Karthik

Content

Abstract

Whenever we try to order some food online the first thing most of us do is check the ratings and reviews of the restaurant to confirm whether the restaurant serves quality food in time. To know whether the restaurant can provide quality food or not, one first looks at the restaurant rating and reviews about the restaurant and quality of food. Bengaluru is one the top cities in India. Most of the people here are dependent mainly on restaurant food as they might be busy in their own works. With such an overwhelming demand for restaurants, studying the demography of a location becomes important. In the world of rising new technology and innovation, the industry is advancing with the role of Artificial Intelligence. Machine learning algorithms can help us to simplify the tasks by helping us to predict and forecast the fututre. This study demonstrates how different Regression algorithms can forecast the rating of restaurants so that one can make a decision whether to buy food(online) from a particular restaurant or not based on ratings and reviews. Different regression algorithms such as Decision Tree, Random forest, XGboost, have been tested and compared to predict the ratings and the algorithm which performed better was chosen.

1. Introduction

1.1 Why this High-Level Document?

The purpose of this High-Level Design (HLD) Document is to add necessary details to the current project description to represent a suitable model for coding. This model 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 the design aspects and define them in detail

- Describe the user interface being implemented

- Describe the hardware and software interfaces

- Describe the performance and 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, application flow (Navigations), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrator of the system.

1.3 Definitions

Terms	DESCRIPTION
Database	Collection of all the information monitored by this system
IDE	Integrated Development Environment
AWS	Amazon Web Services

2. General Description

2.1 Product Perspective

The Restaurant Rating Prediction is a machine learning-based model which will help us to predict the rating of the restaurant in Bangalore. The dataset also contains reviews for each of the restaurants which will help in finding the overall rating for the place.

2.2 Problem Statement

The main goal of this project is to perform extensiv exploratory data analysis and later predict the rating of the restaurant.

2.3 Proposed Solution

The proposed solution for this project is Machine learning algorithms that can be implemented to predict the rating of the restaurant. Considering various features like online order, book table, location,votes, rest type, cuisines,approximate cost for two, listed_in(type) as inputs from the web app, the implemented regression model which will predict the rating of the restaurant.
Here we tried different algorithms such as Decision trees, Random forest,XGboost, etc.
The final model with the highest accuracy(93%) turns out to be an Random forest regressor.

2.4. Data Requirements

The dataset consists of a table with 56351 records and 17 features. The given features are

- **url**: contains the URL of the restaurant in the zomato website.
- **address**: contains the address of the restaurant in Bengaluru.
- **name**: contains the name of the restaurant.
- **online_order**: whether online ordering is available in the restaurant or not.
- **book_table**: table book option available or not.
- **rate**: contains the overall rating of the restaurant out of 5.
- **votes**: contains the total number of ratings for the restaurant as of the above-mentioned date.
- **phone**: contains the phone number of the restaurant.
- **location**: contains the neighborhood in which the restaurant is located.
- **rest_type**: restaurant type
- **dished_liked**: dishes people liked in the restaurant.
- **cuisines**: food styles, separated by comma
- **approx._cost(for two people)**: contains the approximate cost for a meal for two people.
- **reviews**: list of tuples containing reviews for the restaurant, each tuple consists of two values, rating, and review by the customer.
- **menu_item**: contains the list of menus available in the restaurant
- **listed_in(type)**: type of meal.
- **listed_in(city)**: contains the neighborhood in which the restaurant is listed

2.5. Tools used

Python programming language and frameworks such as NumPy, Pandas, Scikit-learn, Flask, AWS, Git.

- PyCharm is used as IDE.
- For visualization of the plots, Matplotlib and Seaborn are used.
- Streamlit is used for the deployment of the model.
- Frontend development is done using Streamlit
- Python Streamlit is used for backend development.
- GitHub is used as a version control system

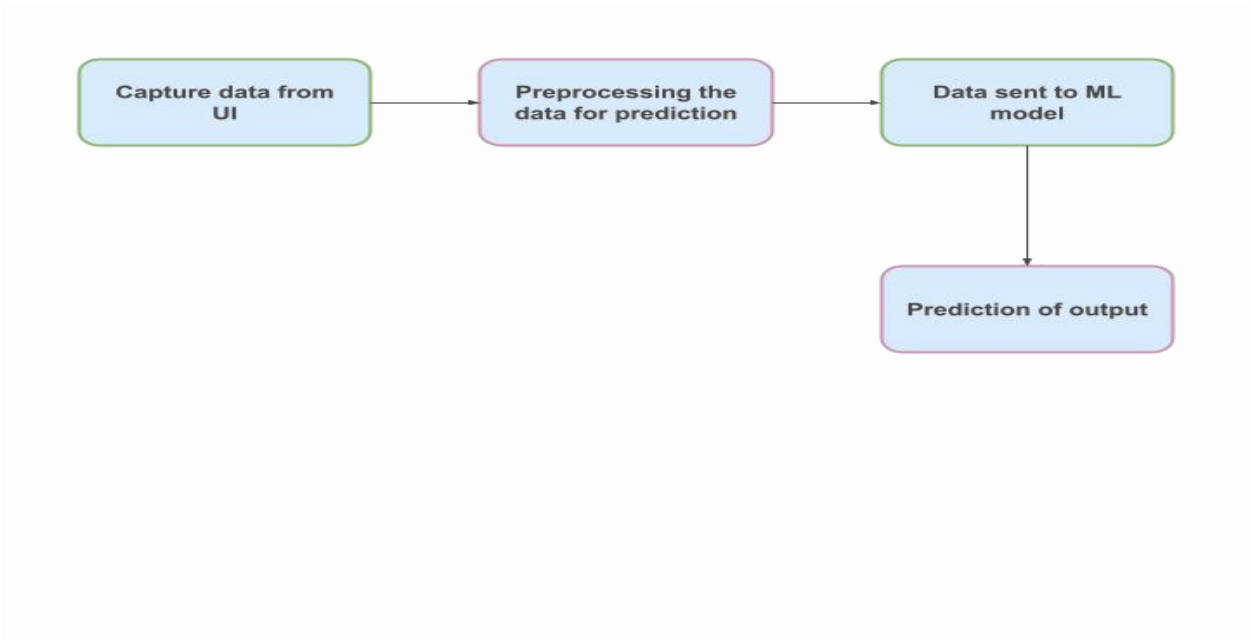
2.6 Constraints

The restaurant rating prediction application must be user-friendly, as automated as possible and users should not be required to know any of the workings.

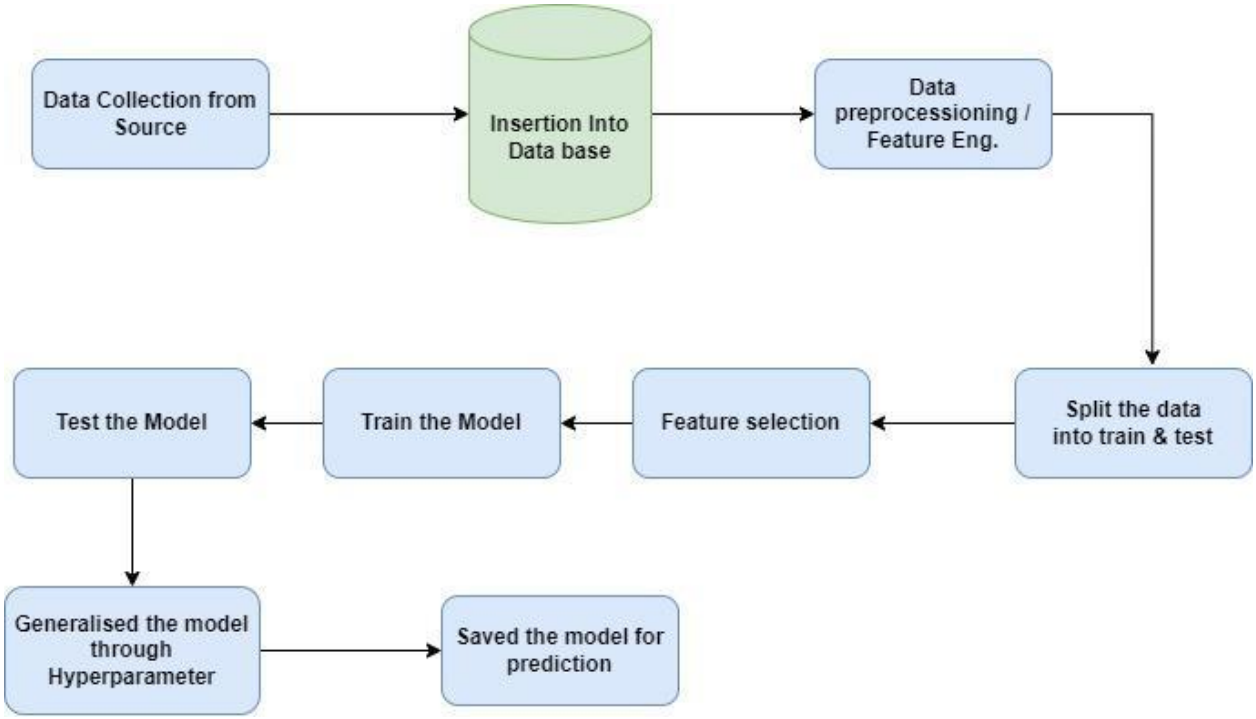
3. Design Details

3.1 Process Flow For predicting the rating of the restaurant, we will use a regression model. Below is the process flow diagram is as shown below.

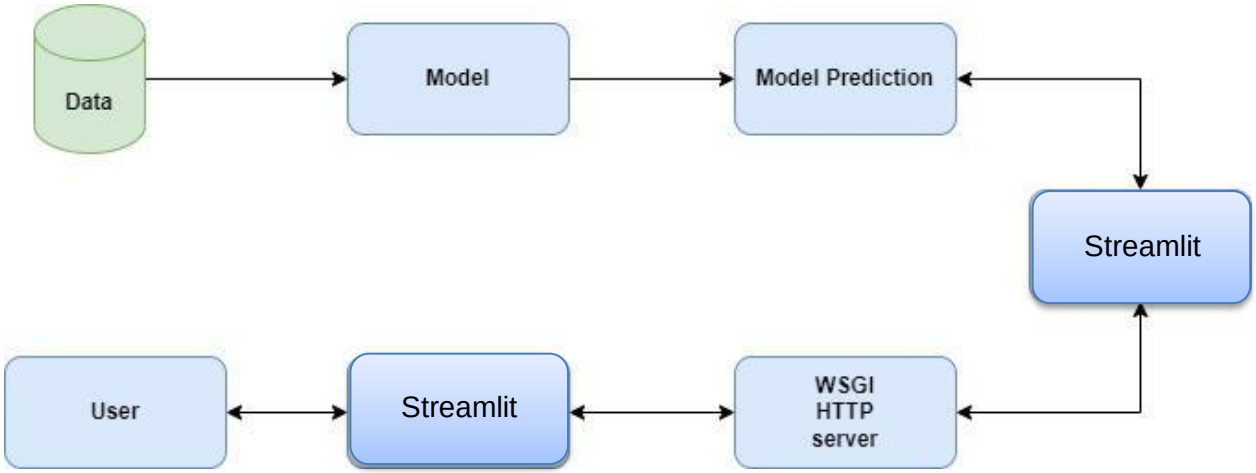
Proposed Methodology



3.1.1 Model Training and Evaluation



3.1.2 Deployment Process



3.2 Error Handling

The error should be encountered well, 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.

4. Performance

We can observe that the accuracy of the predicted output was seen at 93% using an Random Forest regressor. Other classification models such as Decision trees and Xgboost have given good accuracy above 90% and 74% respectively.

4.1 Reusability

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

4.2 Application Compatibility

The different components for this project will be used 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.

4.3 Resource Utilization

When any task is performed, it will likely use all the processing power available until that function is finished.

4.4 Deployment

Streamlit

5. Conclusion

In this project, EDA was performed showing various analytical results. About three machine learning models were built and each of the models shows different accuracies. The best among these models was the Random forest regressor model which shows an accuracy of 93%.