Restaurant Rating Prediction System Technical Documents

by ***Ravi Kumar M***

Abstract

- The Restaurant Rating Prediction System_ is a web-based application designed to predict restaurant ratings based on user inputs such as online orders, table bookings, location, restaurant type, cuisines, and cost for two people. This system utilises machine learning to analyse these factors and make accurate predictions, providing insights for both restaurant owners and customers. The application is built using Flask for the web interface and a pre-trained machine learning model for predictions.

Table of Contents

1. Introduction

Project Overview:

- The Restaurant Rating Prediction System aims to provide an automated tool to predict restaurant ratings based on various features such as location, cost, and restaurant type. This tool helps both restaurant managers to understand the factors impacting their ratings and customers to gauge restaurant quality before visiting.

Problem Statement:

- Many factors contribute to restaurant ratings, and it’s often difficult to assess how each factor influences the final rating. The goal is to build a system that can predict the rating of a restaurant based on features like location, cuisines, cost, and other key metrics using a machine learning model.

2. Objectives

- **Accurate Prediction: Predict restaurant ratings based on features such as cost, location, and restaurant type.**
- **User-Friendly Interface: Develop an easy-to-use web interface for users to input restaurant data and obtain rating predictions.**
- **Integration of Machine Learning: Utilize machine learning algorithms to train the model and deliver accurate predictions.**

3. System Architecture*

Architecture Overview:

The system architecture consists of three main components:

1.Frontend: A web-based user interface built with HTML and CSS for data input and prediction display.

2. Backend: A Flask-based application that processes user input, passes it through the machine learning model, and returns predictions.

3. Machine Learning Model: A trained model that uses historical restaurant data to predict ratings.

4. Dataset and Preprocessing

Dataset Description:

The dataset consists of restaurant attributes such as:

- **Online Order: Whether the restaurant offers online orders (Yes/No).**
- **Book Table: Whether the restaurant allows table bookings.**
- **Votes: The number of votes the restaurant has received.**
- **Location: The location of the restaurant.**

4. Dataset and Preprocessing

Dataset Description:

- **Restaurant Type: The type of restaurant (e.g., cafe, fast food).**
- **Cuisines: The types of cuisines offered.**
- **Cost for Two People: Approximate cost for a meal for two people.**
- **Type: The general category of the restaurant (e.g., dine-in, delivery).**

4. Dataset and Preprocessing

Preprocessing Steps:

- **Label Encoding: Categorical features such as location, restaurant type, and cuisines are encoded using Label Encoders.**
- **Numeric Conversion: Features like votes and cost for two people are converted to numeric data types for model training.**
- **Missing Values Handling: Imputation techniques are applied to handle missing values in the dataset.**

5. Model Selection and Training

Machine Learning Model :

- **A decision tree-based model, **XGBoost**, was selected for its accuracy and performance on the dataset.**

5. Model Selection and Training

Model Training Process:

1. Feature Selection: Features like online orders, table booking, location, restaurant type, etc., are chosen based on their importance.

2. Data Split: The dataset is split into training (70%) and testing (30%) sets.

3. Hyperparameter Tuning: Techniques like grid search are used to fine-tune model hyperparameters.

4. Model Evaluation: The model achieved an accuracy of 89% on the test data.

6. Web Application Implementation

Flask Web Framework:

The web interface was implemented using Flask. It consists of:

- Frontend: HTML forms for user inputs.

- Backend: Flask routes to process form data and call the trained machine learning model for prediction.

6. Web Application Implementation

Example of a Example of a **predictor form** in the app:

```html
<form method="POST" action="/predict">
    <label for="location">Location:</label>
    <select id="location" name="location">
        <option value="Koramangala">Koramangala</option>
        <option value="Indiranagar">Indiranagar</option>
        <!-- More options -->
    </select>
    <input type="submit" value="Predict">
</form>
```

predictor form in the app:

```html
<form method="POST" action="/predict">
    <label for="location">Location:</label>
    <select id="location" name="location">
        <option value="Koramangala">Koramangala</option>
```

7. Testing and Results

Testing Methodology

- Functional Testing: Each component of the system was tested to ensure it works as expected.

- Performance Testing: The model was tested on unseen data to measure prediction accuracy.

Results:

- The system was able to predict restaurant ratings with an accuracy of 89%, and predictions are displayed to the user in real-time on the web interface.

8. Conclusion

- The _Restaurant Rating Prediction System_ demonstrates the ability to utilize machine learning for predicting restaurant ratings based on key features such as location, cost, and type. The web interface ensures that users can easily input data and obtain ratings instantly. Future improvements could include refining the model and adding more advanced features like real-time data updates.

9. Future Enhancements

- **Integration with a Live Database: Connecting the system to a live database for real-time predictions.**

- **User Authentication: Adding login functionality for restaurant owners to access detailed analytics.**

- **More Features: Including features like reviews, service quality, and ambiance for more accurate predictions.**

10. References

1. Flask Documentation. Available at: https://flask.palletsprojects.com

2. XGBoost Documentation. Available at: https://xgboost.readthedocs.io/

3. Pandas Library Documentation. Available at: https://pandas.pydata.org/

End of Document

- **This template includes headings with bold formatting and underlined sections for emphasis, along with code snippets, a placeholder for an architecture diagram, and detailed sections about your project from start to finish. You can further customize it by adding more technical details specific to your project and model, such as the exact accuracy numbers, diagrams, and additional implementation details.**

Thank You