Zomato Restaurant Analysis And Predict Their Ratings

12.06.2024

—

Lavish Gangwani

Document Control

Date	Version	Description	Author
12/06/2024	1.0	Abstract , Introduction ,General Description	Lavish Gangwani
14/06/2024	1.1	Design Detail , API Deployment	Lavish Gangwani
17/06/2024	1.2	Final Revision	Lavish Gangwani

High Level Design (HLD)

TABLE OF CONTENT

High Level Design (HLD)

Abstract

Title: Predicting Restaurant Ratings Using Machine Learning: An Analysis of Zomato Data

Objective: The primary objective of this project is to perform extensive Exploratory Data Analysis (EDA) on the Zomato dataset and build a predictive Machine Learning model to estimate restaurant ratings based on specific features. This will assist Zomato restaurants in anticipating their ratings, thereby enabling them to take strategic actions to enhance customer satisfaction and overall performance.

Approach: The project follows a structured approach encompassing several classical machine learning tasks:

1. **Data Exploration**: Detailed analysis of the dataset to understand its structure, identify patterns, and uncover insights.
2. **Data Cleaning**: Handling missing values, correcting inconsistencies, and preparing the dataset for modeling.
3. **Feature Engineering**: Transforming raw data into meaningful features that enhance model performance.
4. **Model Building**: Training various machine learning models on the processed data.
5. **Hyperparameter Tuning**: Grid Search and Randomized Search methods were used to optimize model parameters.
6. **Model Testing**: Evaluating the performance of the models using appropriate metrics to determine the best fit for predicting restaurant ratings.

Results: The models demonstrated varying degrees of predictive accuracy, with Random Forest Regressors performing the best, achieving the highest R-squared scores on the test data. This model was able to predict restaurant ratings with reasonable accuracy based on the provided features.

High Level Design (HLD)

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 current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to 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 being 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:
    1. Security
    2. Reliability
    3. Maintainability
    4. Portability
    5. Application compatibility
    6. Resource utilization

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 architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

High Level Design (HLD)

2. General Description

2.1 Product Perspective & Problem Statement

The goal of this project is to analyze the restaurants in Bangalore and predict their ratings. To achieve the goal, we used a data set that is formed by the information of 70-80 thousand restaurants. The problem is based on the given Information about each individual. We have to calculate the ratings of individual restaurants.

2.2 Tools Used

jupyter

NumPy

seaborn

3. Design Detail

3.2 Optimization

1. **Your data strategy drives performance**
    - Minimize the number of fields.
    - Minimize the number of records.
    - Optimize extracts to speed up future queries by materializing
    - calculation removing columns and the use of accelerated views

2. **Reduce the marks (data points) in your view**
    - Practice guided analytics. There's no need to fit everything you plan to show in a single view. Compile related views and connect them with action filters to travel from overview to highly-granular views at the speed of thought.
    - Remove unneeded dimensions from the detail shelf.
    - Explore. Try displaying your data in different types of views.

3. **Limit your filters by number and type**
    - Reduce the number of filters in use. Excessive filters on a view will create a more complex query, which takes longer to return results. Double-check your filters and remove any that aren't necessary.

- Use an include filter. Exclude filters load the entire domain of a dimension while including filters do not. An include filter runs much faster than an exclude filter, especially for dimensions with many members.
- Use a continuous date filter. Continuous date filters (relative and range-of-date filters) can take advantage of the indexing properties in your database and are faster than discrete data filters
- Use Boolean or numeric filters. Computers process integers and Booleans (t/f) much faster than strings.
- Use parameters and action filters. These reduce the query load (and work across data sources).

4. Optimize and materialize your calculations

- Perform calculations in the database
- Reduce the number of nested calculations.
- Reduce the granularity of LOD or table calculations in the view. The more granular the calculation, the longer it takes.
- Where possible, use MIN or MAX instead of AVG. AVG requires more processing than MIN or MAX. Often rows will be duplicated and display the same result with MIN, MAX, or AVG.
- Make groups with calculations. Like include filters, calculated groups load only named members of the domain, whereas Tableau's group function loads the entire domain.

High Level Design (HLD)

- Use Booleans or numeric calculations instead of string
- calculations. Computers can process integers and Booleans (t/f) much faster than strings.
- Boolean>Int>Float>Date>DateTime>String.

4. KPI

API will be implemented to display and indicate certain KPIs and relevant indicators for the disease.

4.1 KPIs (Key Performance Indicators)

Key indicators displaying a summary of the restaurants rating and its relationship with different metrics.

- Percentage of People book tables online or offline.
- Location of restaurants.
- Neighborhood in which the restaurants are listed.
- Restaurants accept online orders or not.
- Most liked dish of the restaurants.
- Cuisines of the respective restaurants.

High Level Design (HLD)

5. Deployment

Prioritizing data and analytics couldn't come at a better time. Your company, no matter what size, is already collecting data and most likely analyzing just a portion of it to solve business problems, gain competitive advantages, and drive enterprise transformation. With the explosive growth of enterprise data, database technologies, and the high demand for analytical skills, FastAPI is a modern, fast (high-performance), web framework for building APIs with Python 3.7+ based on standard Python type hints.

- Fast: Very high performance, on par with NodeJS and Go (thanks to Starlette and Pydantic). One of the fastest Python frameworks available.
- Fast to code: Increase the speed to develop features by about **200% to 300%.**
- Fewer bugs: Reduce about **40%** of human (developer) induced errors.
- Intuitive: Great editor support. Completion everywhere. Less time debugging.
- Easy: Designed to be easy to use and learn. Less time reading docs.
- Short: Minimize code duplication. Multiple features from each parameter declaration. Fewer bugs.