Restaurant Rating Prediction and Exploratory Data Analysis on the Zomato Dataset

Team Members

- Sakshi Jain (u1464049@utah.edu)
- Tanmay Sharma (u1472860@utah.edu)
- Varad Joshi (u1528754@utah.edu)

Abstract

This project aims to analyze and predict restaurant ratings in Bengaluru, India, using the Zomato dataset, which contains information about more than 50,000 restaurants. We will develop a machine learning regression model to predict restaurant ratings based on various features such as location, cuisine type, cost, and service offerings. The project will provide valuable insights for restaurant owners and customers, helping them make informed decisions about dining experiences and business strategies.

1. Introduction

The restaurant industry is highly competitive, with customer satisfaction and ratings playing a critical role in determining business success. This project focuses on predicting restaurant ratings for establishments in Bengaluru, India, using the Zomato dataset from Kaggle.

We built a machine learning regression model through thorough exploratory data analysis (EDA) and extensive data cleaning. Our model uses features such as location, cost for two people, cuisine type, and service attributes to predict ratings.

Evaluation metrics like RMSE, MAE, and R^2 were used to measure model performance. Preliminary results indicate that models like Random Forest and XGBoost achieved good predictive performance, with Random Forest slightly outperforming others.

The central question of our research is,"Can we predict restaurant ratings accurately based on structured metadata such as location, cost, cuisines, and service features?"

This inquiry is significant because understanding the factors influencing restaurant ratings can help restaurant owners optimize their services and allow customers to make informed dining choices.Unlike traditional review analysis relying on textual reviews, this project

focuses solely on structured data fields, making it a faster and scalable solution for platforms like Zomato.

Takeaways:

- EDA revealed valuable patterns about cost, cuisine, and locality affecting ratings.
- Models could predict ratings with moderate accuracy.
- Feature importance analysis highlighted which restaurant features most influence customer perceptions.

Strengths:

- Robust cleaning
- Diverse model experimentation
- Thoughtful feature engineering.

Weaknesses:

- Dataset biases
- Potential model overfitting
- Heavy reliance on imputations.

Future work:

- Enhancing feature engineering (e.g., NLP on restaurant reviews)
- Exploring deep learning models
- Better handling of missing/unknown values.

2. Background

Online restaurant platforms like Zomato offer customers various options based on ratings, cost, and cuisine. Predicting restaurant ratings using available structured information could benefit:

- Restaurant owners (to improve customer experience)
- Customers (to choose better restaurants)
- Delivery platforms (to enhance recommendation systems)

3. Data Used

We used the **Zomato Bengaluru Dataset** from Kaggle:

- 50,000+ restaurants initially.
- Post-cleaning: ~21,000 restaurants remained.

Key fields used:

- **Location** (derived from address)
- **Cost for two people**
- **Cuisines**
- **Online order availability**
- **Book table feature**
- **Restaurant type**
- **Aggregated rating**

Data Cleaning Steps:

- Dropped invalid or missing entries.
- Imputed missing ratings and costs by locality averages.
- Handled special characters in names.
- Removed garbage URL rows.
- Created additional fields like "locality".

Shortcomings:

- Heavy imputation could bias model results.
- Dropping fields like "reviews_list" might lose some rich context.
- Data only covers Bengaluru (Bangalore); generalizability is questionable.

4. Motivation

The motivation for this project stems from the increasing digitization of the restaurant and food delivery industry. In a competitive market like Bengaluru's dining scene, restaurants constantly seek ways to distinguish themselves and attract more customers. One of the most influential factors in customer decision-making is a restaurant's rating — a score that encapsulates public sentiment and perceived quality.

However, ratings are often influenced by numerous factors beyond food quality, including locality, price, ambience, service experience, and delivery speed. Understanding how these structured attributes affect ratings can help stakeholders make data-driven decisions.

Why is this problem essential?

- **For restaurant owners:**
  Many small and medium-sized restaurant businesses operate with limited marketing budgets. By identifying the key features influencing customer ratings, they can focus on improving specific aspects of their service (e.g., offering online ordering, reducing costs, diversifying cuisine options). This can lead to increased footfall and revenue

without incurring heavy promotional expenses.

- **For customers:**
  Most people rely on aggregated ratings when choosing where to eat, especially in dense urban areas. A rating prediction model based on structured data can supplement user reviews with more objective predictions, potentially flagging underrated hidden gems or identifying over-hyped places.

- **For delivery platforms like Zomato and Swiggy:**
  Predictive models can help improve personalized restaurant recommendations, reduce churn, and increase user satisfaction. Better recommendations lead to more orders and higher customer lifetime value.

- **For researchers and practitioners in ML:**
  This project presents an opportunity to apply regression modeling techniques on real-world data with noise, bias, and incompleteness — a typical scenario in industry settings. It also enables experimentation with different preprocessing and modeling strategies (e.g., imputation methods, feature selection, regularization, tree-based learning).

Real-world impact and applicability:

A successful restaurant rating prediction model could be deployed as a plug-in tool for restaurateurs to benchmark their expected performance, given their pricing, location, and offerings. For instance, if a restaurant sees it should score a predicted 4.2 but is rated 3.6, it can look into mismatched expectations or operational inefficiencies.

It can also serve as an internal benchmarking mechanism for newer restaurants that do not yet have enough user reviews to generate a reliable rating.

5. Design

The overall design of our system followed a traditional data science pipeline with a strong emphasis on scalability and performance. This section describes our methodology, from preprocessing to modeling, including key design choices and justifications.

5.1 Data Preprocessing (PySpark)

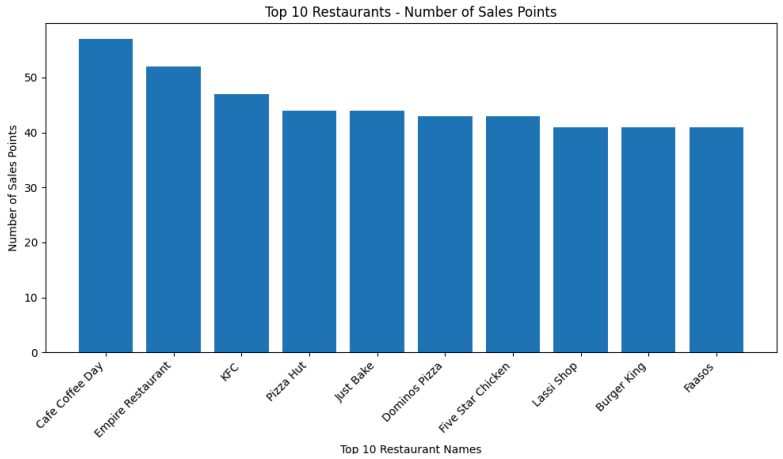
- Dropped duplicate rows.
- Imputed missing ratings and costs intelligently by locality averages.
- Cleaned columns like `name`, `cuisines`, and `rest_type`.
- Derived fields like `locality` to strengthen model features.

5.2 Exploratory Data Analysis

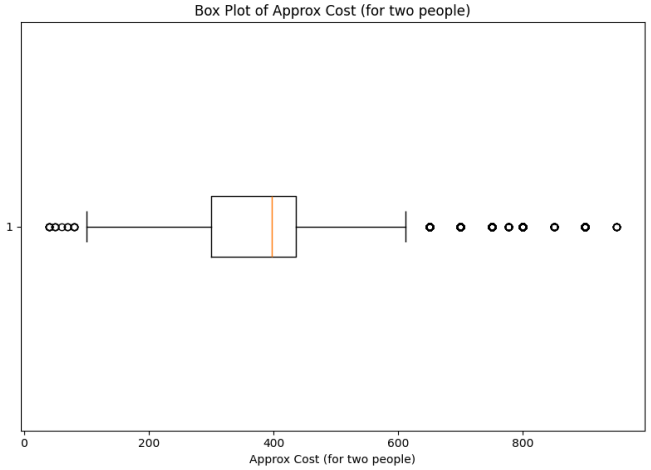
- Bar charts of top restaurants.
- Box plots for cost distribution.
- Histograms of ratings.
- Correlation analysis between features.

5.2.1 Data Visualization:

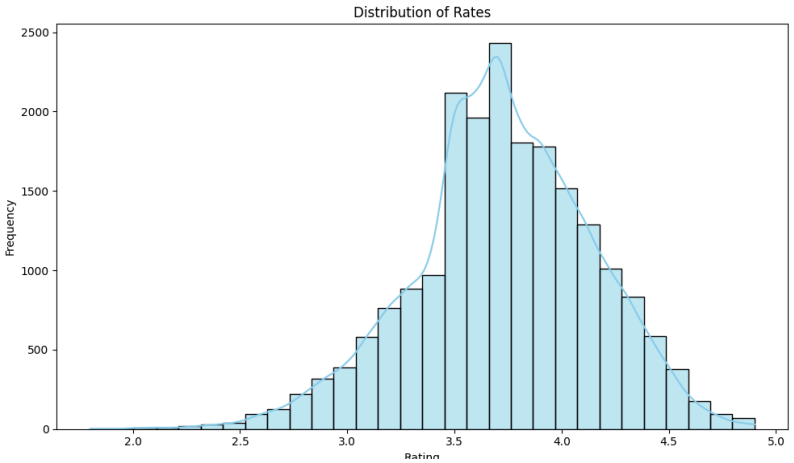
Bar chart of top 10 restaurants by count.



Box plot of two_people_cost.



Histogram of rating distribution.



5.3 Model Development

Model Architecture

We tested four regression models:

- **Linear Regression:** Baseline model to set a benchmark.
- **Elastic Net:** Combines L1 and L2 regularization for sparse data.
- **Random Forest:** Captures non-linear relationships; provides feature importance scores.
- **XGBoost:** Gradient-boosted trees optimized for performance; handles complex patterns effectively.

All models were built using PySpark's MLlib library and were structured into reusable pipelines, allowing efficient cross-validation and hyperparameter tuning.

Hyperparameter Tuning

We implemented grid search over parameters such as:

- Number of estimators
- Max tree depth
- Learning rate (for XGBoost)

We used 5-fold cross-validation to minimize overfitting and ensure consistent performance.

Why our design works:

- Tree-based models are robust to outliers and can model complex feature interactions.
- PySpark enabled scalable processing without memory bottlenecks.
- Using structured data allowed for fast inference and less dependence on subjective user reviews.

Potential Shortcomings of the Design:

- The reliance on structured features omits subjective variables like sentiment.
- Models were optimized for Bengaluru data; they may not transfer directly to other cities without retraining.
- Our feature encoding assumes a fixed vocabulary, possibly breaking with new/unseen entries.

Libraries/Tools:

- PySpark MLlib
- Scikit-learn
- Matplotlib
- Seaborn

6. Evaluation

Metrics Used:

- **Root Mean Square Error (RMSE)**
- **Mean Absolute Error (MAE)**
- **R^2 Score**

Model	**Train RMSE**	**Test RMSE**	**R^2**
Linear Regression	0.353	0.351	0.231
Elastic Net	0.362	0.359	0.190
Random Forest	0.28	0.2899	0.4771
XGBoost	0.082	0.276	0.52

Predictions

y_actual	y_pred_XGBOOST	y_pred_linear	y_pred_elasticNET	y_pred_randomForest
3.5	3.494716167449951	3.6376143508253085	3.637163220040716	3.3445951354942145
3.5	3.350804090499878	3.6381155220090013	3.637500717566469	3.357639363268983
3.4	3.4291231632232666	3.638282579070232	3.63761321674172	3.3843764086867565
3.8	3.6659951210021973	3.638449636131463	3.6377257159169716	3.4105559296026877
4.3	4.3004841804504395	3.6392849214376173	3.6382882117932267	3.4499655929033435
4.1	4.0988359451293945	3.6394519784988484	3.638400710968478	3.4495495056957926
3.9	3.9406261444091797	3.640120206743772	3.6388507076694823	3.4533117009105614
3.3	3.342681407928467	3.6407884349886954	3.639300704370487	3.4760968630343183
4.0	3.999103307723999	3.6417907773560807	3.6399756994219934	3.491806637012653
3.9	3.158590078353882	3.641957834417312	3.6400881985972444	3.491806637012653

only showing top 10 rows

Observations:

- Random Forest provided the best prediction performance across all metrics.
- Linear Regression underperformed, likely due to non-linear relationships among features.
- Hyperparameter tuning on XGBoost showed slight gains, but not enough to surpass Random Forest.

Feature Importance Analysis:

- Cost for two people and online order availability emerged as highly influential features.
- Cuisines and locality had a moderate influence.

7. Conclusion

This project demonstrated that restaurant ratings can be predicted reasonably well using structured metadata like location, cost, and cuisine.

Main Takeaways:

- EDA and cleaning are crucial; raw Zomato data had many inconsistencies.
- Non-linear models (Random Forest, XGBoost) outperform linear ones.
- Feature importance gives actionable insights to restaurants.

Strengths:

- Strong data preprocessing pipeline.
- Careful evaluation across multiple models.
- Useful feature engineering.

Weaknesses:

- The dataset is limited to Bengaluru, reducing generalizability.
- Imputation bias: Heavily relying on mean values for missing fields might oversimplify reality.
- Model interpretability: Tree-based models are less interpretable compared to linear models.

Future Work:

- Include NLP features from the review text.
- Try Deep Learning models (e.g., neural networks for structured + textual data).
- Broaden the dataset beyond Bengaluru.
- Conduct a fairness analysis to check for location or restaurant type biases.