MACHINE LEARNING IN BUSINESS

MIS710 – Assignment 2 – Part A: Technical Report

Enhancing Restaurant Performance on FoodieBay: Data-Driven Insights and Predictive Models

Class Group: Wednesday (15:00 – 16:30)

Tutor: XXX

Students: Cam Ha Nguyen

DEAKIN
BUSINESS
SCHOOL

Table of Contents

Executive Summary

This report aims to uncover key factors influencing restaurant ratings on the FoodieBay platform, enabling data-driven decisions for enhanced restaurant performance.

We conduct comprehensive data analysis, deploy three supervised machine learning models (K-Nearest Neighbors, Decision Tree, Random Forest), and employ unsupervised methods (K-Means Clustering) to extract actionable insights, predict restaurant ratings and group restaurants sharing similar characteristics.

Our analysis reveals the influence of votes, costs, table booking availability, and other factors on restaurant ratings. In addition to these insights, our K-Means Clustering analysis identifies four distinctive restaurant clusters, each exhibiting unique characteristics.

We recommend Random Forest as the model of choice, offering a balance of accuracy, efficiency, and scalability.

We propose the integration of the Random Forest model for real-time rating predictions on FoodieBay's platform. Additionally, we commit to providing ongoing support and maintenance for the deployed model, alongside the exploration of advanced data analytics techniques. We advocate for the incorporation of new variables to further enhance prediction accuracy.

While our model excels in accuracy and scalability, we emphasize the need for vigilance against model drift over time. To address this, we recommend time-based retraining, continuous monitoring, and the creation of comprehensive documentation for future enhancements.

1. Introduction

Business Context and Problem Statement

In the dynamic food industry, success for partner restaurants on the FoodieBay platform depends on several key factors. Our objective within restaurant analytics is to uncover these factors, particularly those influencing restaurant ratings. We aim to provide clear insights that can inform better decision-making for both FoodsAnalytics and FoodieBay regarding their restaurant partners.

Value Proposition

This project promises substantial value by:

- In-Depth Data Analysis: We conduct a comprehensive examination of the FoodieBay dataset, uncovering intricate relationships and patterns between restaurant characteristics and overall ratings. These insights support practical decision-making, including performance improvements for Indian restaurants and enhanced customer experiences.
- Predictive Power: Our machine learning model not only provides retrospective insights but also serves as a proactive tool for future-oriented decision-making. By predicting future restaurant ratings, FoodieBay can make proactive decisions, including initiatives to support and promote restaurants likely to perform well and strategies to assist those facing challenges.
- Technical Expertise: This report serves as a testament to our technical proficiency in data analysis and machine learning, extending beyond the immediate business problem.
- Business Alignment: Our analysis is closely aligned with the core business problem and context, ensuring that our findings directly contribute to FoodieBay's pursuit of enhanced restaurant ratings and improved customer experiences.

2. Approach

Our approach centers on supervised machine learning, specifically regression analysis, to predict restaurant ratings. We aim to develop accurate rating forecasting models for FoodieBay's platform and restaurant partners. Furthermore, we applied unsupervised machine learning, utilizing K-Means Clustering on the same dataset to segment restaurants. The process is outlined in Figure 1, with detailed explanations provided in Appendix A.

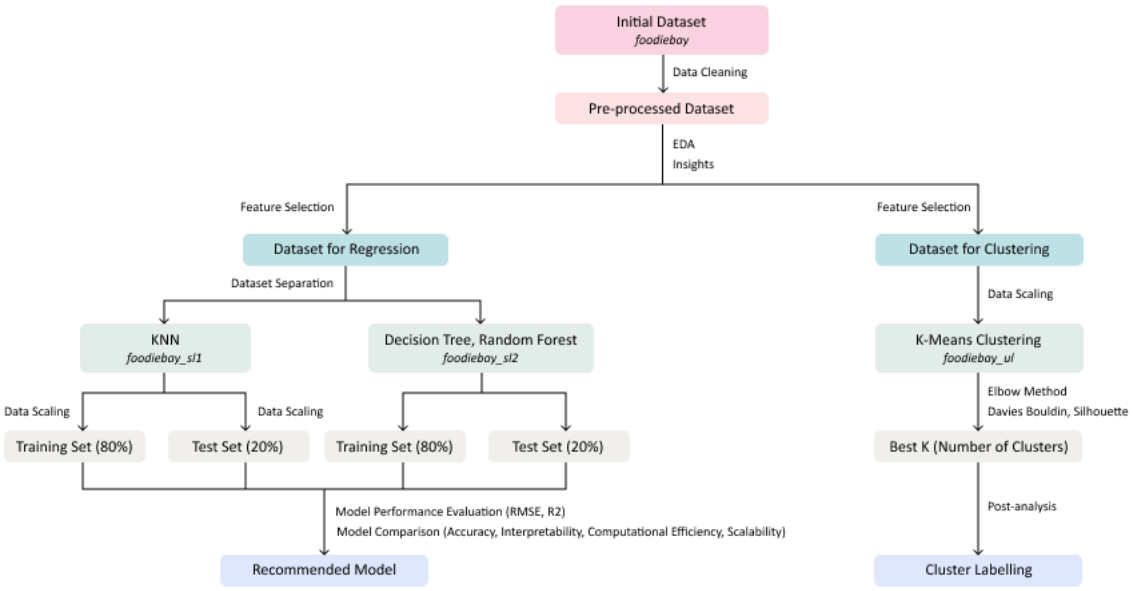

Figure 1 - Machine Learning Model Development Process

3. Data Understanding and Data Preparation

3.1. Data Understanding

The FoodieBay dataset consists of 40,130 rows and 17 columns, featuring a mix of numerical and categorical variables. Independent variables can be organized into distinct categories (Figure 2).

Restaurant Details	url, address, name, phone
Location Information	location, rest_type
Menu and Cuisine	cuisines, menu_item
Listing Specifics	listed_in_type, listed_in_city
Operational Aspects	online_order, book_table, ave_cost_for_two
Customer Reviews and Feedback	dish_liked, votes, ave_review_ranking

Figure 2 - Group of independent variables in FoodieBay dataset

Our primary objective is to explore these independent variables and identify potential predictors of restaurant ratings on FoodieBay ('rate' column).

3.2. Data Cleansing and Preparation

To ensure the accuracy and reliability of our analysis, we prepared the dataset through essential data-cleansing steps:

- **Column Selection:**
    - We excluded irrelevant columns like 'address' 'name' and 'phone' and dropped 'menu_item' due to its limited predictive power.

    - The 'url' column, unrelated to restaurant ratings, was retained temporarily for unique identification.
- **Missing Data:** We addressed missing data in two passes
    - Initially, we focused on the target column, 'rate,' removing 2.87% of rows, leaving 31,794 rows.
    - In the second pass, we set a 10% threshold (1,590 rows) and eliminated 1,100 observations (3.46%) with missing values in 'ave_review_ranking', 'ave_cost_for_two' and 'cuisines' totaling 3.46%.
- **Duplicate Entries:** No duplicate entries were found, allowing us to remove the 'url' column.
- **Outliers and Anomalies:** Using the IQR method, we identified 6,580 outliers, particularly in 'votes' and 'review ranking'. We decided not to remove or impute outliers considering their user-generated nature and substantial data loss.
- **Feature Engineering:** We introduced 'num_cuisines' and 'num_dishes_liked' columns to enhance the dataset's informativeness.

These steps resulted in our final dataset for EDA, comprising 30,699 rows and 14 columns, forming the foundation for subsequent analysis and modeling.

4. Exploratory Data Analysis and Insights Gained

'rate':

- This target variable has a mean rating of about 3.66 and a standard deviation of 0.43.
- As seen from Figure 3, most ratings are clustered around the central value. Ratings range from a minimum of 1.80 to a maximum of 4.90, with the interquartile range (IQR) falling between 3.40 and 4.00 (Figure 4).
- There are outliers with exceptionally low ratings.

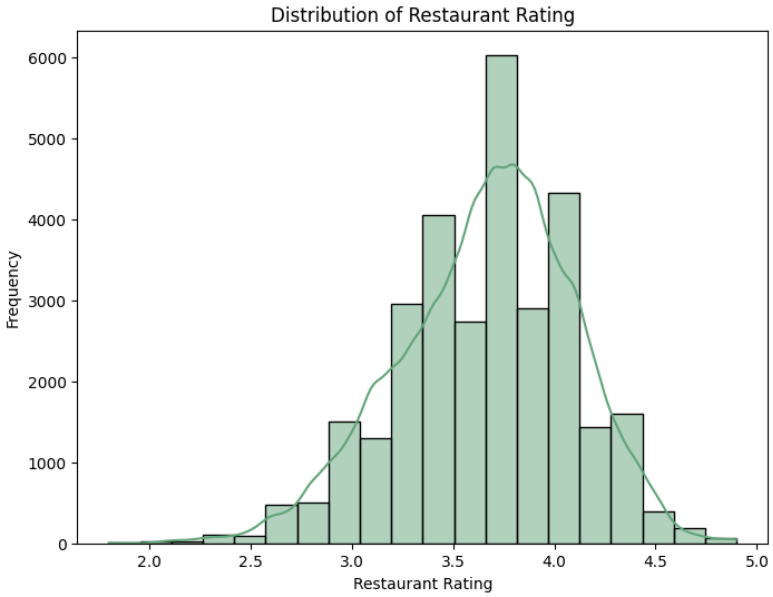

Figure 3 - Distribution of restaurant rating (Histogram)

Figure 4 - Distribution of restaurant rating (Boxplot)

'location' and 'listed_in_city':

- With 90 unique 'location' values and 30 unique 'listed_in_city' values, 'location' is dropped in favor of 'listed_in_city' for more general representation of restaurant locations and simplicity.
- However, the minimal difference between the median and mean ratings by these cities suggests that 'location' has limited influence on restaurant ratings (Figure 5).

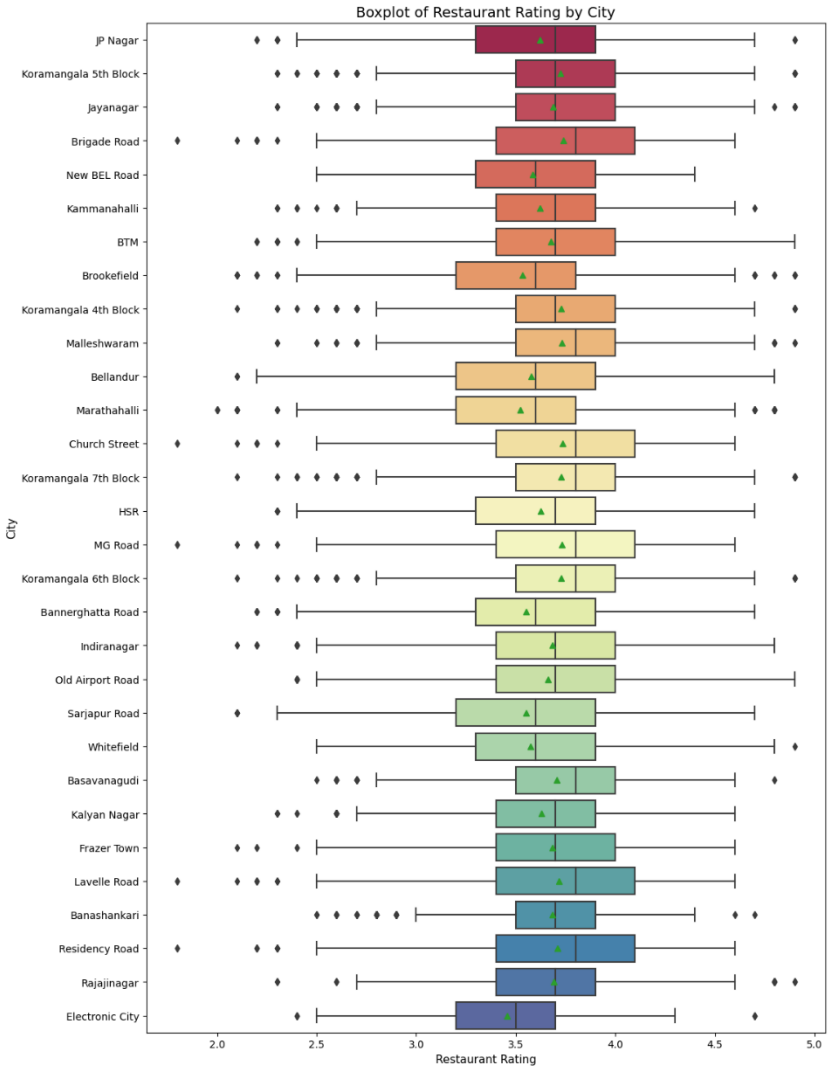

Figure 5 - Distribution of restaurant rating by Indian cities

'rest_type' and 'listed_in_type':

- Both variables have 7 unique values. 'rest_type' includes restaurants belonging to two types simultaneously, leading to potential ambiguity. In contrast, 'listed_in_type' offers clearer and more straightforward categorization without dual types. Therefore, 'rest_type' is dropped.
- Regarding service type, 'Buffet', 'Pubs and Bars', and 'Drinks and Nightlife' types tend to have higher ratings (Figure 6). Furthermore, the 'Dine-out' type exhibits the widest range of ratings, showcasing a diverse range of customer experiences within this category.

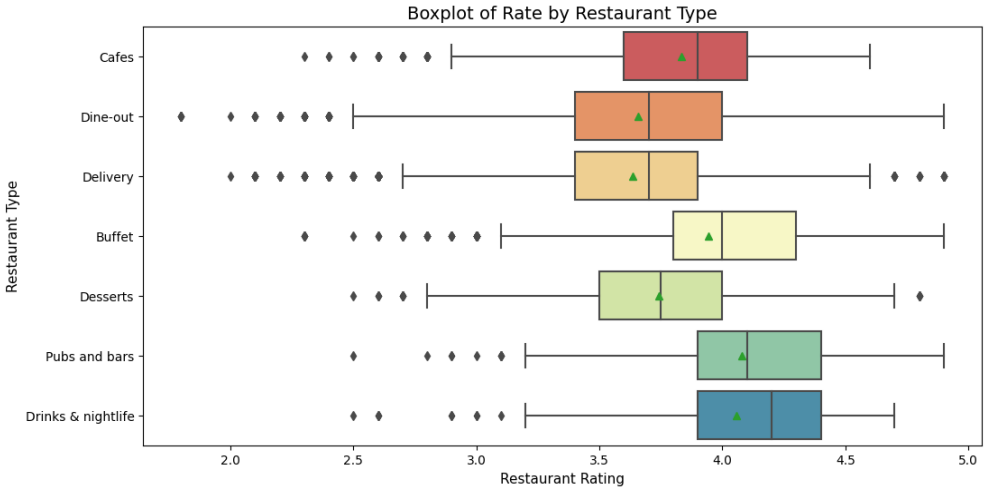

Figure 6 - Distribution of restaurant rating by restaurant service type

'rate' by 'num_cuisines', 'num_dishes_liked', 'votes' and 'ave_review_ranking':

- 'num_dishes_liked' has the strongest positive correlation with 'rate' (R-squared = 0.3299), indicating a significant impact on ratings as the number of liked dishes increases (Figure 7).
- 'votes' and 'ave_review_ranking' also moderately correlate with 'rate' (R-squared = 0.1941 and 0.2258, respectively), suggesting their influence on ratings (Figure 8-9)
- 'num_cuisines' exhibits a weaker correlation (R-squared = 0.0320) with 'rate,' indicating a minor impact on ratings (Figure 10).

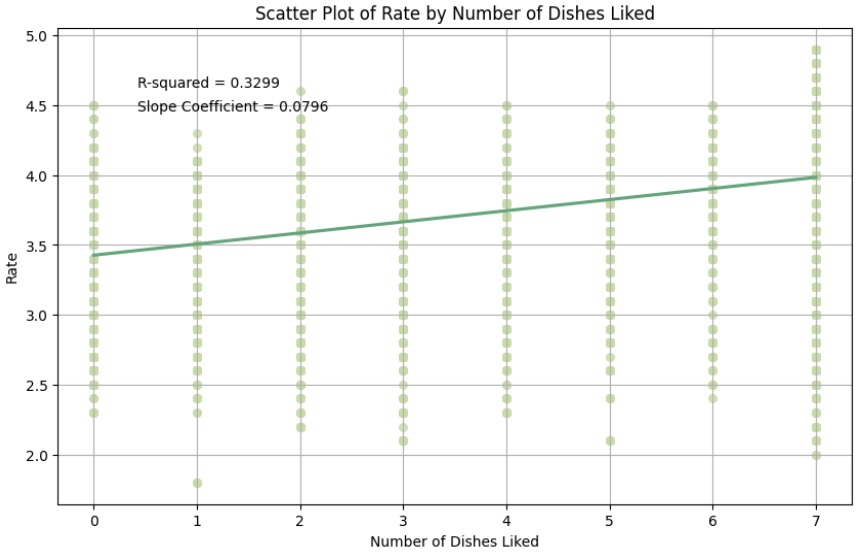

Figure 7 - Scatter plot of rate vs. number of dishes liked

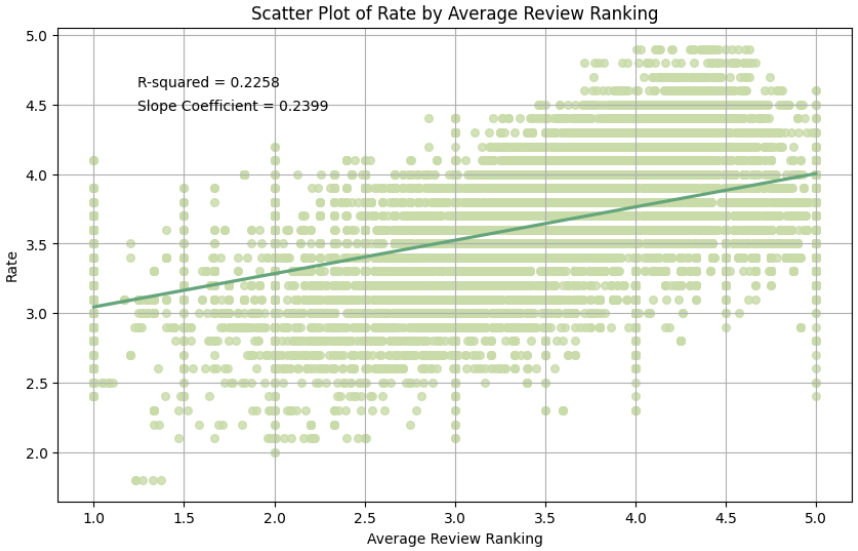

Figure 8 - Scatter plot of rate vs. average review ranking

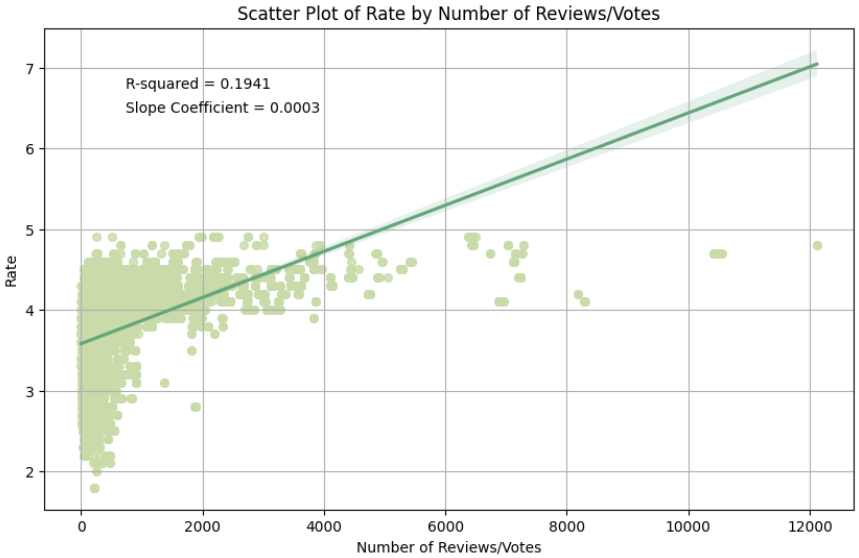

Figure 9 - Scatter plot of rate vs. number of reviews/votes

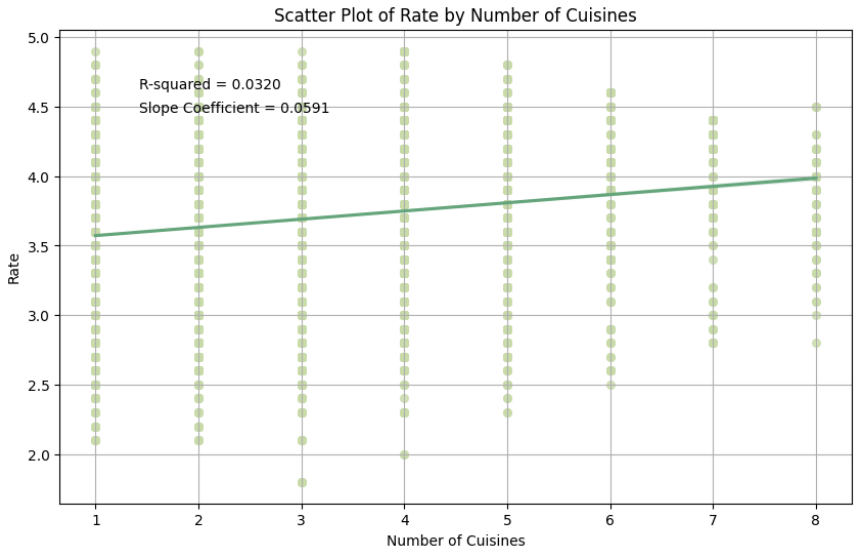

Figure 10 - Scatter plot of rate vs. number of cuisines

'rate' by 'book_table' and 'ave_cost_for_two':

- Restaurants offering table booking services tend to have higher mean and median ratings, indicating a potential predictor of restaurant ratings (Figure 11).

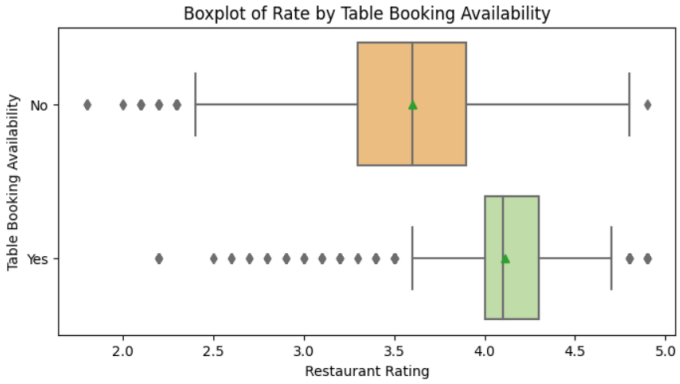

Figure 11 - Restaurant rating distribution by table booking availability

- The relationship between restaurant ratings and average cost for two is statistically significant (R-squared = 0.1479). The positive slope coefficient of 0.0005 suggests that restaurants with higher average costs tend to receive slightly higher ratings (Figure 12).

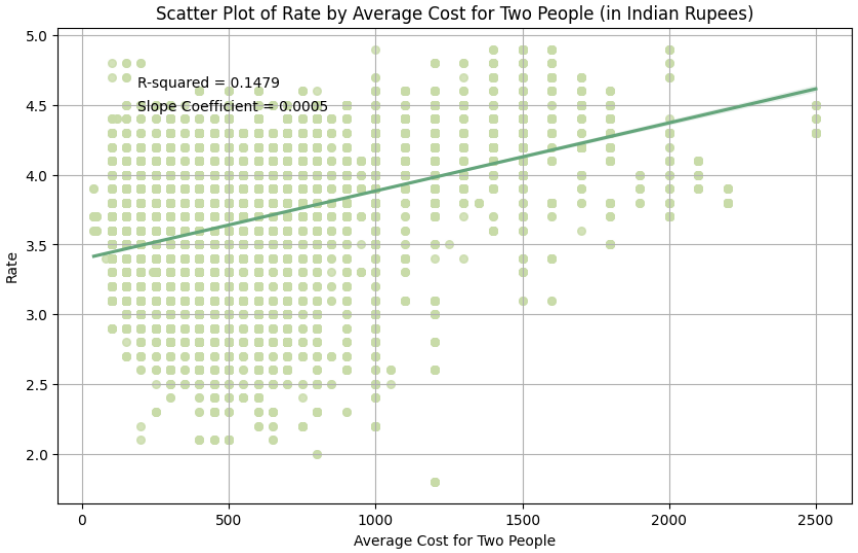

Figure 12 - Scatter plot of rate vs. average cost for two

- When there is no table booking available, many restaurants receive ratings ranging from approximately 2.5 to 4.5 when the average cost for two is below INR 1000. However, when the average cost for two exceeds INR 1500, most ratings range from around 3.5 to 5.0. In contrast, when table booking is offered, most restaurants have an average cost for two ranging from INR 1000 to INR 2000, with the majority receiving ratings above 3.5 (Figure 13).

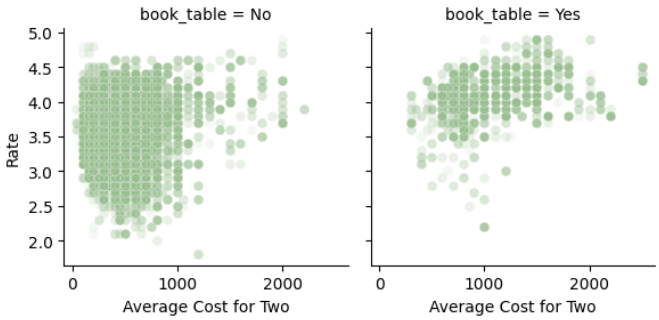

Figure 13 - Scatter plot of rate vs. average cost for two by table booking availability

Correlation matrix:

- The categorical variable 'listed_in_type' was transformed into dummy variables, resulting in seven new columns, each representing a different restaurant service type. 'online_order' and 'book_table' were simplified into binary representations.
- A correlation heatmap (Figure 14) was generated to visualize the relationships between numeric variables, providing a clear and intuitive representation of the degree of association between different factors influencing restaurant ratings. It includes correlation coefficients ranging from -1 to 1, guiding feature selection by identifying influential factors for restaurant rating.

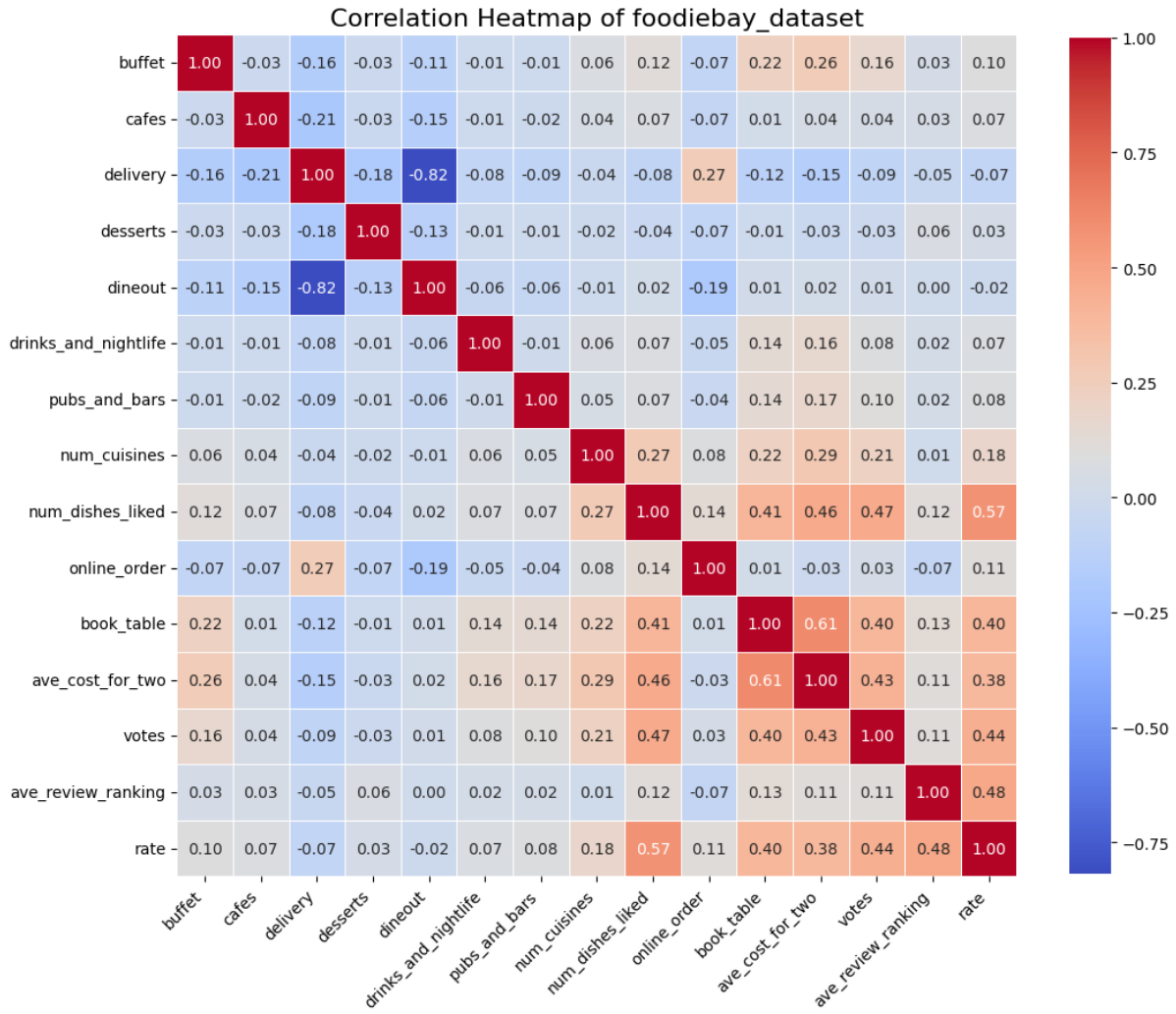

Figure 14 - Correlation matrix for FoodieBay dataset

5. Model Development and Evaluation

5.1. Supervised Machine Learning

5.1.1. Methodology

In the model development phase, we employed regression as our machine learning algorithm as it is well-suited for predicting continuous numerical values, which aligns with our goal of predicting restaurant ratings on FoodieBay. For this analysis, we selected three machine learning models: K-Nearest Neighbors (KNN), Decision Tree, and Random Forest.

5.1.2. Feature Selection

To ensure that our models are focused on the most relevant variables, we identified the top five independent variables with the highest absolute correlation scores with our target variable, 'rate' (Figure 15). These variables include 'num_dishes_liked', 'ave_review_ranking', 'votes', 'book_table', and 'ave_cost_for_two', which all positively correlate with the restaurant rating.

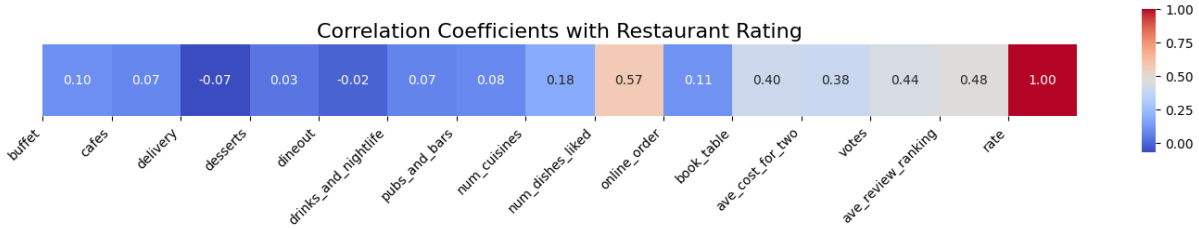

Figure 15 - Correlation coefficients between restaurant rating and other features

5.1.3. Model Development and Evaluation

We initiated our model development process by splitting the dataset into a training set (80%) and a test set (20%) as recommended by Joseph (2022). This division allowed us to develop and fine-tune our models with a clear separation between training and evaluation.

We began with baseline models for KNN, Decision Tree, and Random Forest, optimizing them through hyperparameter tuning for better predictive performance. Model evaluation employed Root Mean Square Error (RMSE), a measure of prediction accuracy, and R-squared (R^2), an indicator of how well the model explains the variance in the data (Chicco et al., 2021).

Here are some specific details related to our model development process:

- For KNN, we applied the Min-Max scaling to ensure that all features contribute equally to the model, preventing potential biases introduced by variables with larger numerical ranges. However, we did not perform scaling for Decision Tree and Random Forest due to their insensitivity to feature scaling as tree-based models.
- We conducted two rounds of cross-validation: one after creating base models and another after hyperparameter tuning, exclusively on the training set to prevent data leakage and maintain model integrity.

5.1.3.1. K-Nearest Neighbors

We initiated our KNN model development by starting with k = 5. To identify the optimal value of k, we assessed the model's performance using RMSE and R^2 values. Upon evaluation, we observed that the lowest RMSE and the highest R^2 were both achieved when k = 1 (Figure 16).

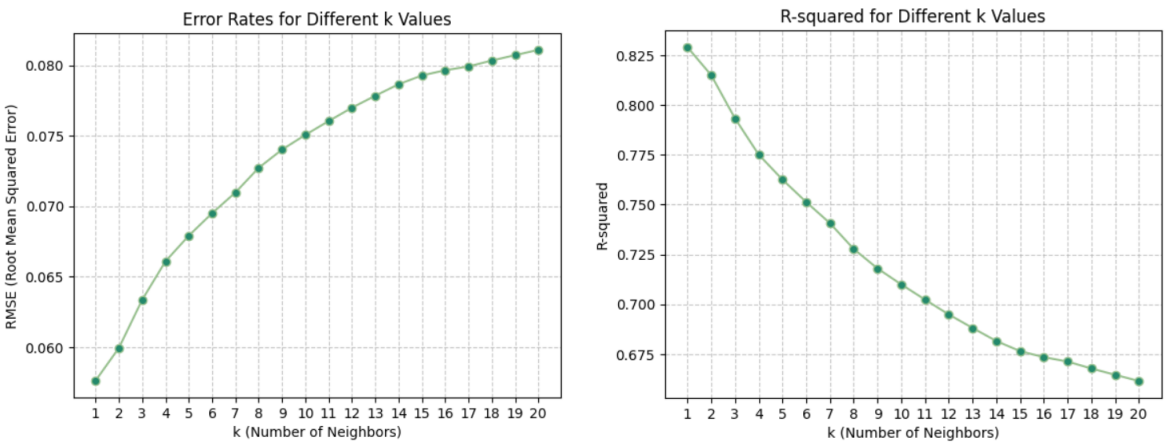

Figure 16 - RMSE and R-squared for different k values in KNN model

While the adjustment substantially improved the model's predictive performance (Figure 17), it introduced the risk of overfitting and loss of smoothing due to the small k value.

KNN	Base Model		k-optimized Model	
	Test Set	Cross-Validation	Test Set	Cross-Validation
Root Mean Squared Error (RMSE)	0.211	0.068 - 0.070	0.179	0.061 - 0.064
R-squared (R2)	0.763	0.733 - 0.746	0.829	0.782 - 0.787

Figure 17 - Summary of KNN model performance

5.1.3.2. Decision Tree

In the Decision Tree approach, we began with a base model consisting of a fully grown tree, which showed impressive training set performance marked by an extremely low RMSE of 0.002 and a high R^2 of 0.989 (Figure 18). However, these characteristics raised concerns about potential overfitting.

Decision Tree	Base Decision Tree			Post-pruned Decision Tree		
	Training Set	Test Set	Cross-Validation	Training Set	Test Set	Cross-Validation
Root Mean Squared Error (RMSE)	0.002	0.178	0.188 - 0.197	0.126	0.195	0.201 - 0.209
R-squared (R2)	0.989	0.830	0.781 - 0.795	0.916	0.796	0.758 - 0.770

Figure 18 - Summary of Decision Tree model performance

To mitigate overfitting, we applied post-pruning method using ccp_alpha, a regularization parameter that controls the complexity of the tree. The post-pruned model, although displaying slightly higher cross-validation RMSE and a somewhat lower cross-validation R^2, strikes a better balance performance between training and test sets. This adjustment ensures more reliable predictions on unseen data.

5.1.3.3. Random Forest

This ensemble learning method, employing fully grown trees, outperformed previous models with low RMSE (0.152 to 0.160) and high R^2 (0.852 to 0.865) (Figure 19), while showing greater resistance to overfitting when compared to individual decision trees.

Random Forest	Base Random Forest			Pre-pruned Random Forest		
	Training Set	Test Set	Cross-Validation	Training Set	Test Set	Cross-Validation
Root Mean Squared Error (RMSE)	0.069	0.147	0.152 - 0.160	0.067	0.141	0.146 - 0.152
R-squared (R2)	0.975	0.885	0.852 - 0.865	0.976	0.894	0.864 - 0.874

Figure 19 - Summary of Random Forest model performance

For this model, we utilized GridSearchCV for comprehensive hyperparameter optimization, testing 162 combinations (Appendix B). After reviewing the results, models with 200 and 100 trees in the forest resulted in R^2 values of 0.8708 and 0.8693, respectively (Figure 20). Given the minimal performance difference, we opted for 'n_estimators' as 100 to accelerate the training process, a valuable choice when dealing with large datasets and memory constraints during deployment.

param_max_depth	param_max_features	param_min_samples_leaf	param_min_samples_split	param_n_estimators	mean_test_r2	rank_test_r2	mean_test_neg_mean_absolute_error	rank_test_neg_mean_absolute_error
40	sqrt	1	2	200	0.8708	1	-0.0829	1
40	sqrt	1	2	100	0.8693	2	-0.0833	2

Figure 20 - Snapshot of Random Forest model performance during hyperparameter tuning

5.2. Unsupervised Machine Learning

5.2.1. Methodology

We utilized K-Means Clustering, an unsupervised machine learning method, to uncover patterns and groupings within the FoodieBay dataset. Our analysis focused on numeric variables, and we performed feature scaling before applying the clustering algorithm. The selected features for K-Means Clustering included 'num_cuisines', 'online_order', 'book_table', 'ave_cost_for_two', 'votes', 'ave_review_ranking' and 'rate'. Our goal was to identify the optimal number of clusters (K) for meaningful segmentation using evaluation methods like the Elbow Method, Davies Bouldin Index, and Silhouette Score.

5.2.2. Model Evaluation

5.2.2.1. Optimal K Value

By combining insights from these evaluation methods (Figure 21), we determined that K = 4 offers the most meaningful and balanced clustering solution:

- Elbow Method: At K = 4, there is a significant change in the rate of decrease in WCSS, indicating an elbow point.
- Davies Bouldin Score: The lowest score was at K = 4 (approximately 0.60), indicating well-separated and compact clusters.

- Silhouette Score: The highest score occurred at K = 4, slightly above 0.6, indicating fairly good cluster cohesion and separation.

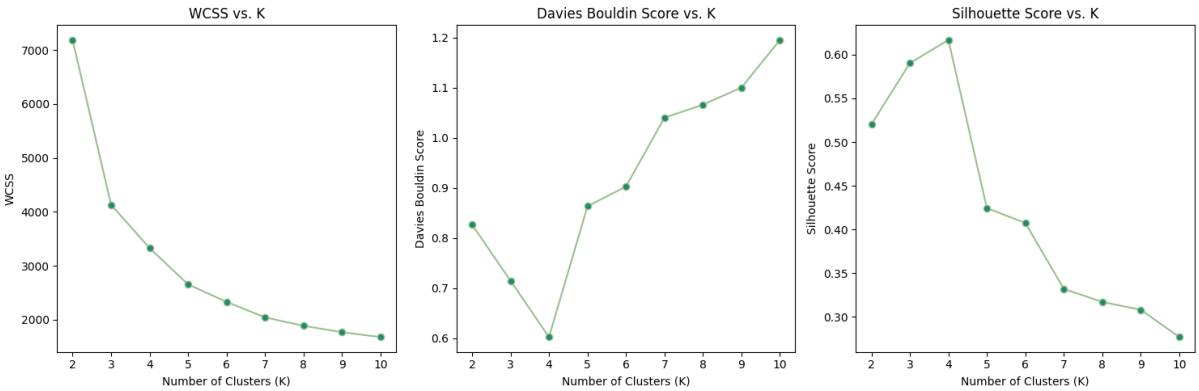

Figure 21 - K-Means Clustering performance across different K values

From Figure 22, the average Silhouette coefficient consistently hovers around 0.62, with no clusters falling below this average score. No negative values are observed, implying correct cluster assignments for all restaurants.

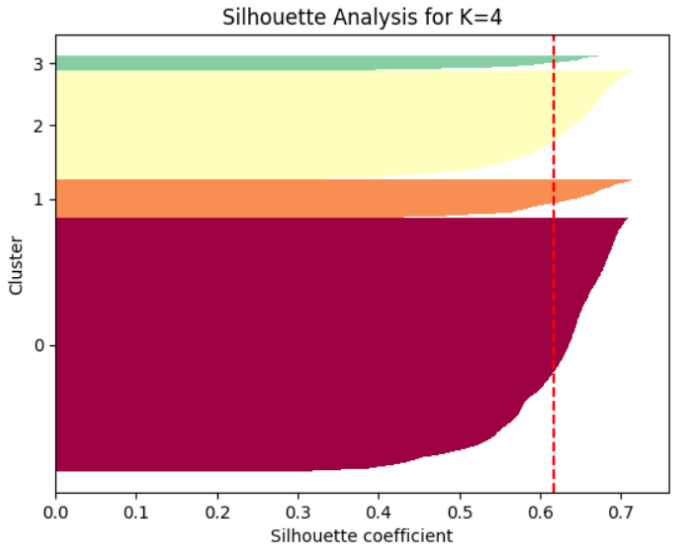

Figure 22 - Silhouette plot for K-Means Clustering with K = 4

5.2.2.2. Cluster Analysis

Cluster Size

Cluster 0 is the largest (18,711 data points), followed by Cluster 2 (8,033). Cluster 1 and Cluster 3 are smaller, comprising 2,846 and 1,109 data points, respectively. Figure 23 visually represents the percentage distribution of data points across these clusters, offering insights into their distribution.

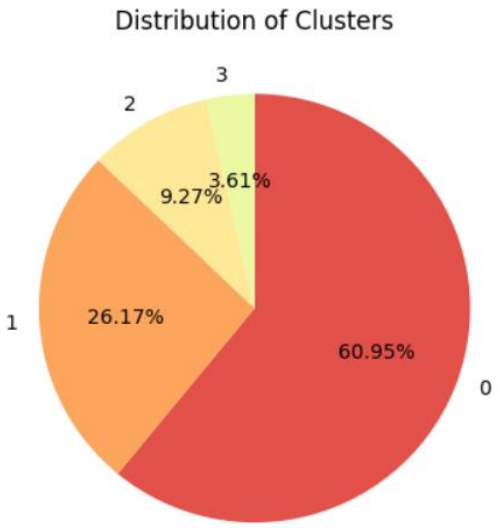

Figure 23 - Cluster distribution

Feature Importance for Clusters

Key factors distinguishing restaurant clusters include online order availability, table booking, and the number of votes, as indicated by their high mutual information scores (Figure 24). These insights are valuable for understanding the defining characteristics of different restaurant segments within the dataset.

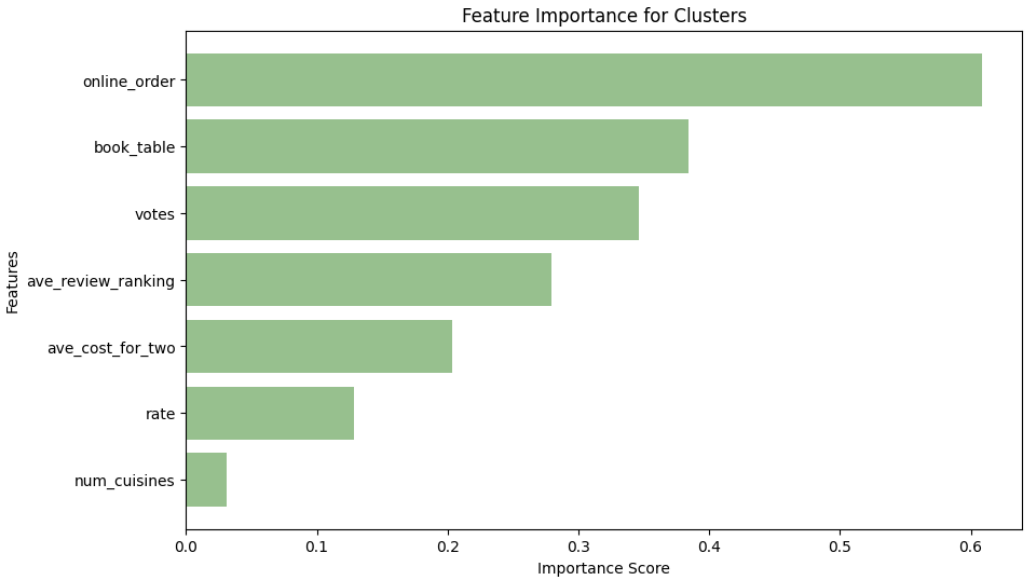

Figure 24 - Importance score for all features in K-Means Clustering model

Our feature selection for cluster analysis is also guided by Figure 25 to 29:

- Figures 25 to 27 demonstrate that Clusters 0 and 2 exhibit similar distributions in 'ave_cost_for_two' and 'votes' while Clusters 1 and 3 share similar distributions within these two variables. Clusters 0 and 2 have lower mean and median values in 'ave_cost_for_two', 'votes' and 'rate', whereas Clusters 1 and 3 show higher mean and median values for these features.

- Figure 30 emphasizes that Clusters 2 and 3 do not offer online orders, while Clusters 0 and 1 do. Figure 31 illustrates that Clusters 0 and 2 lack table booking, whereas Clusters 1 and 3 have this feature.

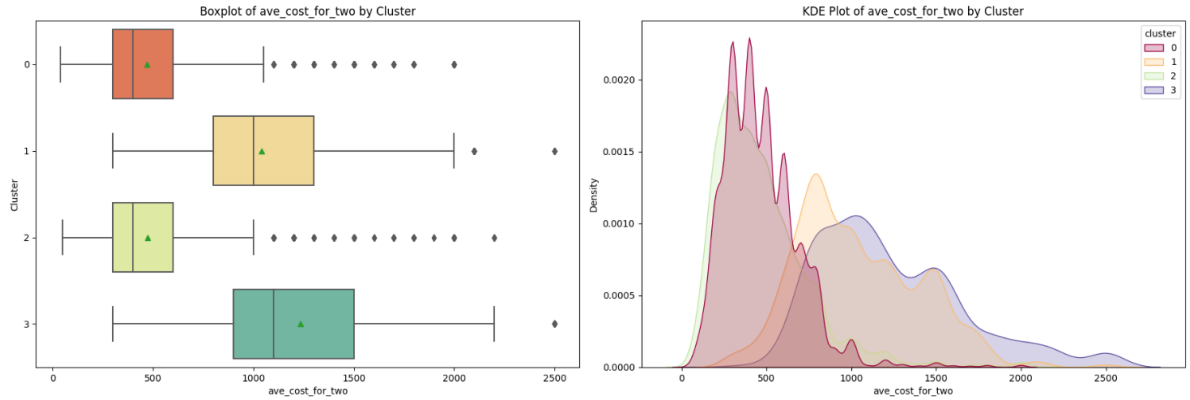

Figure 25 - Distribution of average cost for two by clusters

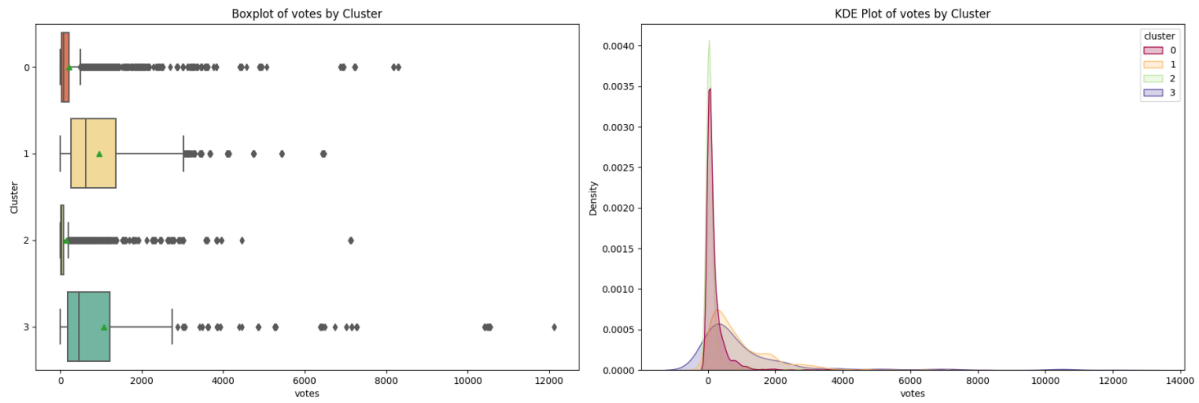

Figure 26 - Distribution of number of votes/reviews by clusters

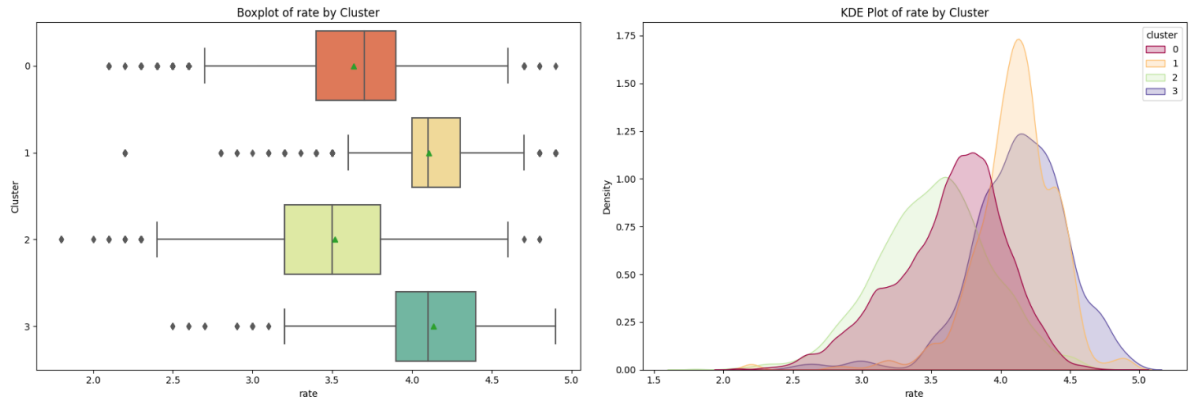

Figure 27 - Distribution of restaurant rating by clusters

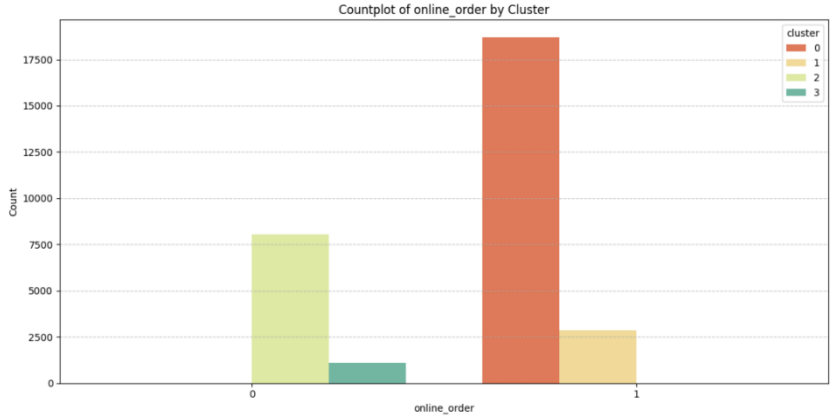

Figure 28 - Countplot for online ordering availability by clusters

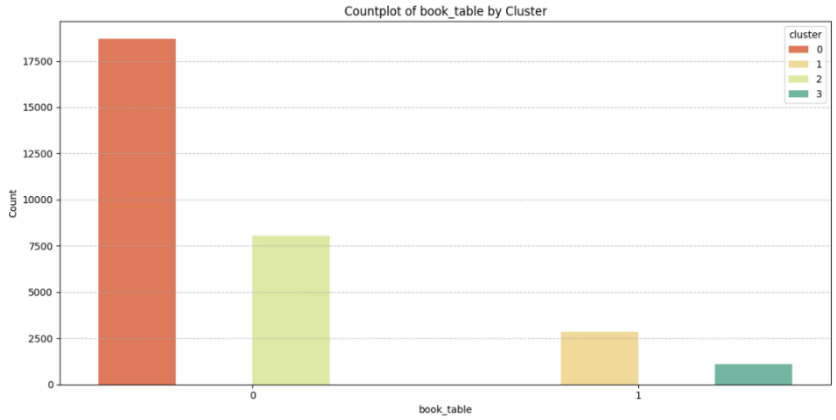

Figure 29 - Countplot for table booking availability by clusters

5.2.2.3. Cluster Labelling

Summarizing the discussed variables, we created a consolidated summary of cluster characteristics in Figure 30. Notably, regardless of whether restaurants offer online ordering, the absence of table booking hinders them from achieving high ratings, as evident in the characteristics of Clusters 0 and 2.

Cluster	Variables/Characteristics				
	book_table	online_order	votes	ave_cost_for_two	rate
Cluster 0	no	yes	moderate	moderate	moderate
Cluster 1	yes	yes	high	high	high
Cluster 2	no	no	low	moderate	low
Cluster 3	yes	no	low	high	high

Figure 30 - Summary table for cluster characteristics

By focusing on the remaining four key characteristics, we were able to assign technical and business labels to each cluster, as depicted in Figure 31. A more detailed explanation of the business labels can be found in Appendix C.

Cluster	Count	Label	
		Technical	Business
Cluster 0	18711	No Table Booking, Moderate Votes, Moderate Cost, Moderate Rating	Convenience Eateries
Cluster 1	8033	Table Booking Available, High Votes, High Cost, High Rating	Fine Dining Destinations
Cluster 2	2846	No Table Booking, Low Votes, Moderate Cost, Low Rating	Local Favorites
Cluster 3	1109	Table Booking Available, Low Votes, High Cost, High Rating	Gourmet Experiences

Figure 31 - Cluster label assignment

However, it is worth acknowledging that K-means clustering has limitations related to outliers and the sensitivity to initial centroid selection. Outliers can distort cluster boundaries, while the choice of initial centroids may lead to different local minima, impacting the stability and quality of clustering outcomes (Hassan et al., 2019, Singh et al., 2011).

6. Solution Recommendation

6.1. Model Recommendation

As FoodieBay is keen on uncovering influential factors behind restaurant ratings to enhance its platform's restaurant performance, our primary focus is on assessing and recommending an appropriate supervised machine learning model for regression analysis. This approach is designed to provide precise insights into rating predictions, rather than clustering restaurants into similar groups, as it aligns more closely with FoodieBay's specific goals and requirements.

In evaluating the performance of the three supervised machine learning models above, we will take these criteria into consideration: accuracy, interpretability, computational efficiency, and scalability.

Accuracy:

Based on the cross-validation results in Figure 32:

- Random Forest achieved the highest accuracy among the three models, with the lowest RMSE and the highest R^2. This indicates that it can make more precise predictions of restaurant ratings.
- KNN also performed well, achieving an R^2 slightly below that of Random Forest.
- Decision Tree had the lowest accuracy, indicating potential limitations in capturing data patterns.

Model	Performance Metrics	
	Root Mean Squared Error (RMSE)	R-squared (R2)
K-Nearest Neighbors	0.063 +/- 0.002	0.784 +/- 0.002
Decision Tree	0.205 +/- 0.004	0.764 +/- 0.006
Random Forest	0.149 +/- 0.003	0.869 +/- 0.005

Figure 32 - Model performance comparison

Interpretability:

All three models face inherent challenges due to the nature of predicting continuous numeric values for restaurant ratings.

Computational Efficiency:

- KNN can be computationally expensive, especially with large datasets, as it calculates distances for each data point.
- Decision Tree is typically faster and more computationally efficient compared to KNN.
- Random Forest gives moderate computational efficiency. While it involves multiple trees, it can be parallelized to some extent, making it more efficient than KNN for large datasets.

Scalability:

- KNN can struggle with scalability, particularly with a high number of data points, as the distance calculations grow.
- Decision Tree scales relatively well and is suitable for medium-sized datasets.
- Random Forest is moderately scalable and can handle larger datasets compared to KNN.

Considering all criteria, Random Forest emerges as the recommended model for FoodieBay. It not only achieved the highest accuracy among the models but also offers a balance between computational efficiency and scalability. This makes Random Forest a robust choice for making precise predictions of restaurant ratings.

6.2. Future Engagements with Clients

In our ongoing partnership with FoodieBay client, we foresee the following areas of collaboration:

- **Model Deployment:** Collaborating with our client to integrate the recommended Random Forest model into their platform, allowing for real-time restaurant rating predictions.
- **Monitoring and Maintenance:** Providing ongoing support and maintenance for the deployed model to ensure its continued effectiveness and accuracy.
- **Additional Insights:** Exploring advanced techniques like sentiment analysis to gain deeper customer insights and enhance user experience.
- **Expanded Data Collection:** Considering the inclusion of more restaurant-related data, such as parking availability and outdoor seating options, to enrich analysis and model development.

7. Technical Recommendation

7.1. Development and Testing Environment

To effectively implement our approach, we leveraged the following technical components:

- **Software Libraries:** We employed a range of software libraries for data manipulation, visualization, machine learning, and clustering (Figure 33).

- **Programming Language:** Our analysis was conducted using Python 3 as the primary programming language.
- **Computing Environment:** We executed our analysis within a Google Colab, a hosted Jupyter Notebook service, harnessing the advantages of cloud computing, which eliminates the need for powerful local hardware and easy collaboration among team members.

Category	Key Libraries Used
Data Manipulation and Analysis	Pandas, NumPy, SciPy, Scipy.stats
Data Visualization	Matplotlib, Seaborn, pydotplus
Machine Learning	Scikit-Learn (sklearn)
Clustering	Scikit-Learn (sklearn), Matplotlib

Figure 33 - Main software libraries used in Python

7.2. Model Deployment and Data Preprocessing

To facilitate model deployment, we created a machine learning process diagram outlining the key steps, from data preprocessing to model training and validation (Figure 34). This diagram will serve as a valuable reference for deploying and maintaining the Random Forest model.

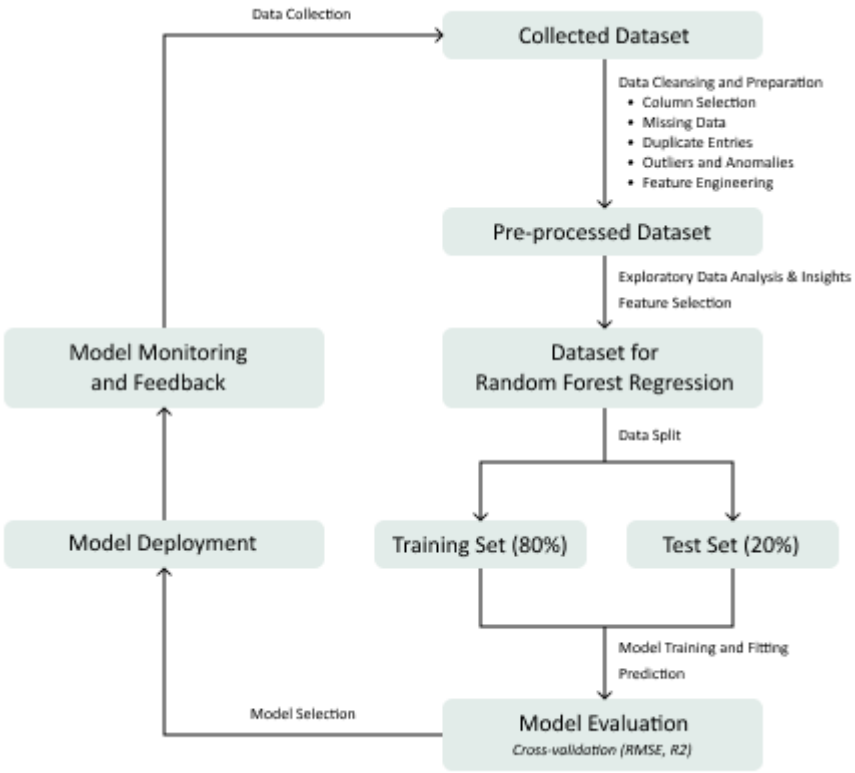

Figure 34 - Machine learning process diagram for Random Forest model

Regarding data preprocessing, we recommend implementing regular data audits and updates, addressing any missing or erroneous data. Additionally, feature engineering techniques may be explored to enhance the model's predictive power.

7.3. Maintenance for Accuracy and Relevance

Regarding maintenance for long-term accuracy, we need to be aware of the potential for model drift, where changing variables and environments can lead to a decline in accuracy over time. To mitigate this, we can consider the measures suggested by Appen (2021):

- Time-Based Retraining: Schedule regular model updates to adapt to evolving trends (Appen, 2021).
- Continuous Retraining: Monitor key performance indicators, triggering retraining when needed (Appen, 2021).

Regardless of the approach we choose, we still need human expertise for efficient issue detection and resolution (Appen, 2021). Furthermore, the development team should maintain meaningful and comprehensive model documentation to facilitate transparency, understanding, and efficient troubleshooting throughout the model's lifecycle (Bhat et al., 2023).

Future additions of new features or data sources should be considered to enhance predictive accuracy and relevance.

References

Appen (2021, February 26). AI Model Maintenance: A Guide to Managing a Model Post-Production. *Appen*. https://appen.com/blog/ai-model-maintenance-guide-to-managing-model/

Bhat, A., Coursey, A., Hu, G., Li, S., Nahar, N., Zhou, S., ... & Guo, J. L. (2023, April). Aspirations and Practice of ML Model Documentation: Moving the Needle with Nudging and Traceability. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (pp. 1-17).

Chicco, D., Warrens, M. J., & Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Computer Science*, *7*, e623. DOI:10.7717/peerj-cs.623

Hassan, A. A. H., Shah, W., Husein, A. M., Talib, M. S., Mohammed, A. A. J., & Iskandar, M. (2019). Clustering approach in wireless sensor networks based on K-means: Limitations and recommendations. *Int. J. Recent Technol. Eng*, *7*(6), 119-126.

Joseph, V. R. (2022). Optimal ratio for data splitting. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, *15*(4), 531-538. https://doi.org/10.1002/sam.11583

Singh, K., Malik, D., & Sharma, N. (2011). Evolving limitations in K-means algorithm in data mining and their removal. *International Journal of Computational Engineering & Management*, *12*(1), 105-109.

Appendices

Appendix A

Machine Learning Model Development Process

- **Data Sourcing:** Our dataset was provided by FoodieBay, ensuring its relevance and suitability for our analysis.
- **Data Preprocessing:** Data preprocessing was a critical step to clean, format, and prepare the dataset for analysis. This involved handling missing values, removing duplicated observations, addressing outliers, and ensuring data consistency.
- **Exploratory Data Analysis (EDA):** EDA played a pivotal role in gaining a comprehensive understanding of the dataset. Through statistical analysis and visualizations, we explored data distribution, relationships, and potential patterns.
- **Feature Selection:** Feature selection was guided by insights gained during EDA, helping us identify the most relevant variables for our machine learning models.
- **Dataset Preparation and Dataset Separation:** To facilitate our machine learning model development, we divided our dataset into two distinct subsets, each tailored for specific tasks:
    - Dataset for Supervised Machine Learning: We created two separate datasets, namely "foodiebay_sl1" and "foodiebay_sl2." These datasets were used for training and evaluating various supervised machine learning models, including k-Nearest Neighbors (KNN), Decision Trees (DT), and Random Forest (RF). Separating the data into these subsets allowed us to experiment with different algorithms and assess their performance.
    - Dataset for Unsupervised Machine Learning: For unsupervised machine learning, we established a dataset named "foodiebay_ul." This dataset was employed for K-Means Clustering analysis. By isolating a dedicated dataset for clustering, we aimed to uncover meaningful patterns and group restaurants based on similarity.
- **Model Development and Evaluation**
    - Supervised Machine Learning: Our focus was on three key supervised machine learning algorithms: KNN, DT, and RF. We followed a systematic approach that included both base and optimized models.
        - Model Comparison: To evaluate the performance of these models, we utilized various performance metrics, including Root Mean Squared Error and R-squared. This comparison allowed us to identify the most suitable model for the given problem.

- Unsupervised Machine Learning: Our unsupervised machine learning approach involved clustering restaurants into meaningful categories, providing valuable insights into restaurant groupings.

Appendix B

param_grid for GridSearchCV

- Number of trees in the forest: 3

  'n_estimators': [50, 100, 200]
- Maximum depth of each tree: 3

  'max_depth': [10, 20, 40]
- Minimum samples required to split a node: 3

  'min_samples_split': [2, 4, 6]
- Minimum samples required at each leaf node: 3

  'min_samples_leaf': [1, 2, 4]
- Number of features to consider when looking for the best split: 2

  'max_features': ['auto', 'sqrt']

Total number of combinations: $3 \times 3 \times 3 \times 3 \times 2 = 162$

Optimal combination: {'max_depth': 40, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200}

Appendix C

Cluster label assignment with description

Cluster	Technical	Characteristics (Technical Label)	Description
Cluster 0	No Table Booking, Moderate Votes, Moderate Cost, Moderate Rating	Convenience Eateries	Restaurants that offer convenience with moderate popularity and pricing
Cluster 1	Table Booking Available, High Votes, High Cost, High Rating	Fine Dining Destinations	Upscale dining establishments known for their high popularity, premium pricing, and excellent ratings
Cluster 2	No Table Booking, Low Votes, Moderate Cost, Low Rating	Local Favorites	Neighborhood eateries with lower popularity, affordable pricing, and moderate ratings
Cluster 3	Table Booking Available, Low Votes, High Cost, High Rating	Gourmet Experiences	Exclusive dining venues offering gourmet experiences, albeit with lower popularity and premium pricing