



Project Report

Title: Predict Restaurant Ratings

Author : Pranjal Gupta

Date: 05.07.2024

Internship at: Cognifyz Technologies

Abstract:

The project aimed to develop a machine learning model to predict the aggregate rating of restaurants based on various features. By leveraging data science techniques, the model seeks to provide valuable insights to restaurant owners and stakeholders for enhancing customer satisfaction and business performance. This report outlines the methodology, implementation details, results, and analysis of the project.

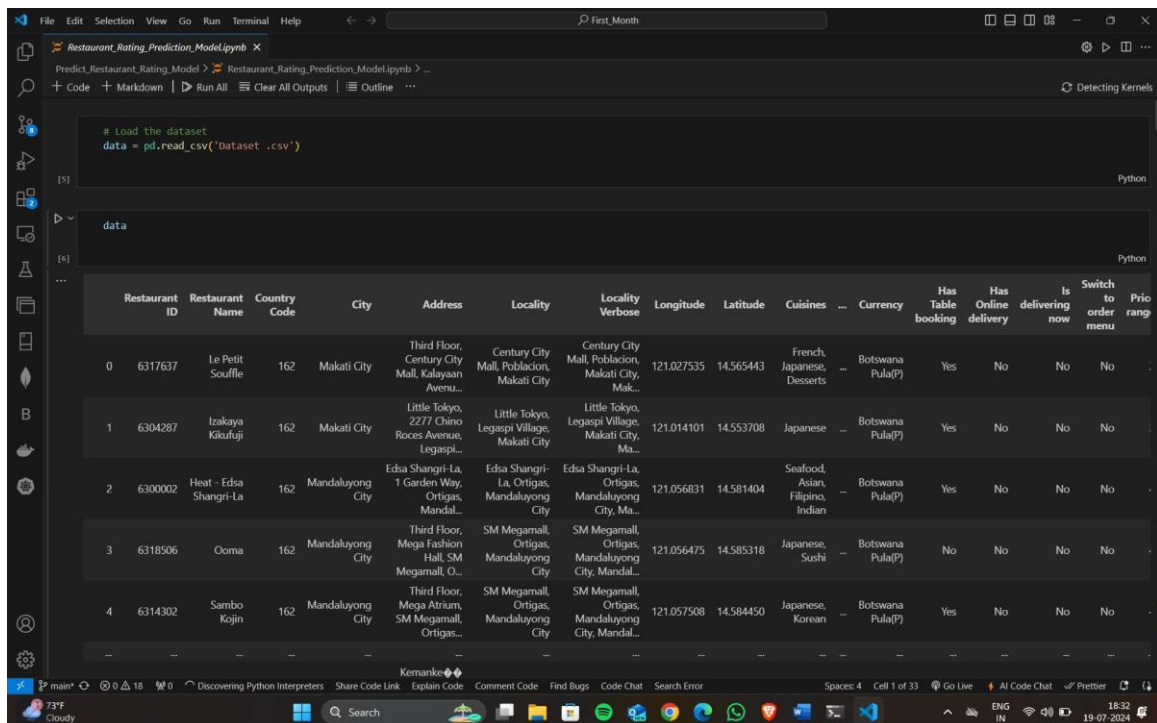
1. Introduction:

The introduction provides context for the project by discussing the growing significance of online reviews and ratings in shaping consumer decisions in the restaurant industry. It underscores the challenges faced by restaurant owners in maintaining high ratings amidst fierce competition and the increasing reliance on data driven approaches for strategic decision making. The introduction sets the stage for the project's objectives and highlights its potential impact on the restaurant industry ecosystem.

2. Data Preprocessing:

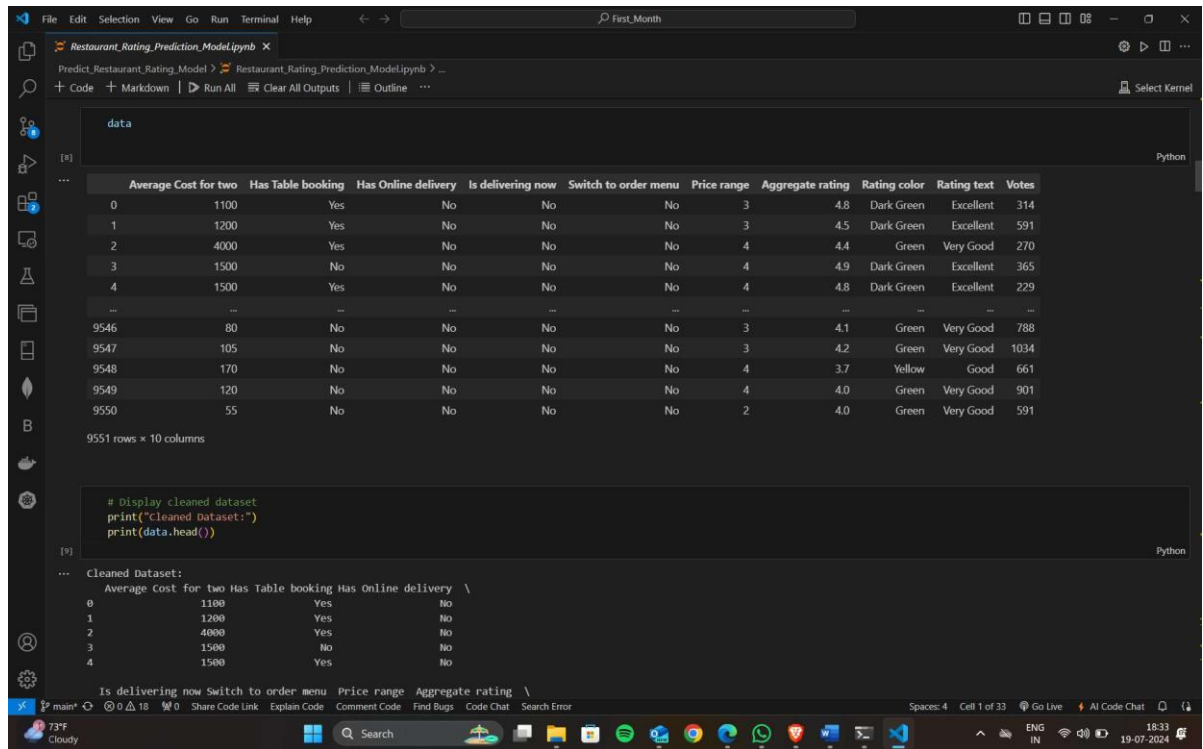
This section delves into the intricacies of data preprocessing, including data cleaning, feature engineering, and normalization. It discusses techniques for handling missing values, outliers, and categorical variables, ensuring the dataset is suitable for model training. The section also explores methods for addressing class imbalance and selecting relevant features to improve model performance.

Visualizations such as correlation matrices and distribution plots are used to illustrate preprocessing steps and insights gained from the data.



The screenshot displays a Jupyter Notebook interface with a Python script that loads a dataset from a CSV file. The dataset is visualized as a table with the following columns: Restaurant ID, Restaurant Name, Country Code, City, Address, Locality, Locality Verbose, Longitude, Latitude, Cuisines, Currency, Has Table booking, Has Online delivery, Is delivering now, Switch to order menu, and Price range. The table contains five rows of data, each representing a different restaurant.

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude	Cuisines	Currency	Has Table booking	Has Online delivery	Is delivering now	Switch to order menu	Price range
0	6317637	Le Petit Souffle	162	Makati City	Third Floor, Century City Mall, Kalayaan Avenue...	Century City Mall, Poblacion, Makati City	Century City Mall, Poblacion, Makati City, Mak...	121.027535	14.565443	French, Japanese, Desserts	Botswana Pula(P)	Yes	No	No	No	
1	6304287	Izakaya Kikufuji	162	Makati City	Little Tokyo, 2277 Chino Roces Avenue, Legaspi...	Little Tokyo, Legaspi Village, Makati City	Little Tokyo, Legaspi Village, Makati City, Ma...	121.014101	14.553708	Japanese	Botswana Pula(P)	Yes	No	No	No	
2	6300002	Heat - Edsa Shangri-La	162	Mandaluyong City	Edsa Shangri-La, 1 Garden Way, Ortigas, Mandal...	Edsa Shangri-La, Ortigas, Mandaluyong City	Edsa Shangri-La, Ortigas, Mandaluyong City, Ma...	121.056831	14.581404	Seafood, Asian, Filipino, Indian	Botswana Pula(P)	Yes	No	No	No	
3	6318506	Ooma	162	Mandaluyong City	Third Floor, Mega Fashion Hall, SM Megamall, O...	SM Megamall, Ortigas, Mandaluyong City	SM Megamall, Ortigas, Mandaluyong City, Mandal...	121.056475	14.585318	Japanese, Sushi	Botswana Pula(P)	No	No	No	No	
4	6314302	Sambo Kojin	162	Mandaluyong City	Third Floor, Mega Atrium, SM Megamall, Ortigas...	SM Megamall, Ortigas, Mandaluyong City	SM Megamall, Ortigas, Mandaluyong City, Mandal...	121.057508	14.584450	Japanese, Korean	Botswana Pula(P)	Yes	No	No	No	



```
data
```

	Average Cost for two	Has Table booking	Has Online delivery	Is delivering now	Switch to order menu	Price range	Aggregate rating	Rating color	Rating text	Votes
0	1100	Yes	No	No	No	3	4.8	Dark Green	Excellent	314
1	1200	Yes	No	No	No	3	4.5	Dark Green	Excellent	591
2	4000	Yes	No	No	No	4	4.4	Green	Very Good	270
3	1500	No	No	No	No	4	4.9	Dark Green	Excellent	365
4	1500	Yes	No	No	No	4	4.8	Dark Green	Excellent	229

9551 rows x 10 columns

```
# Display cleaned dataset
print("Cleaned Dataset:")
print(data.head())
```

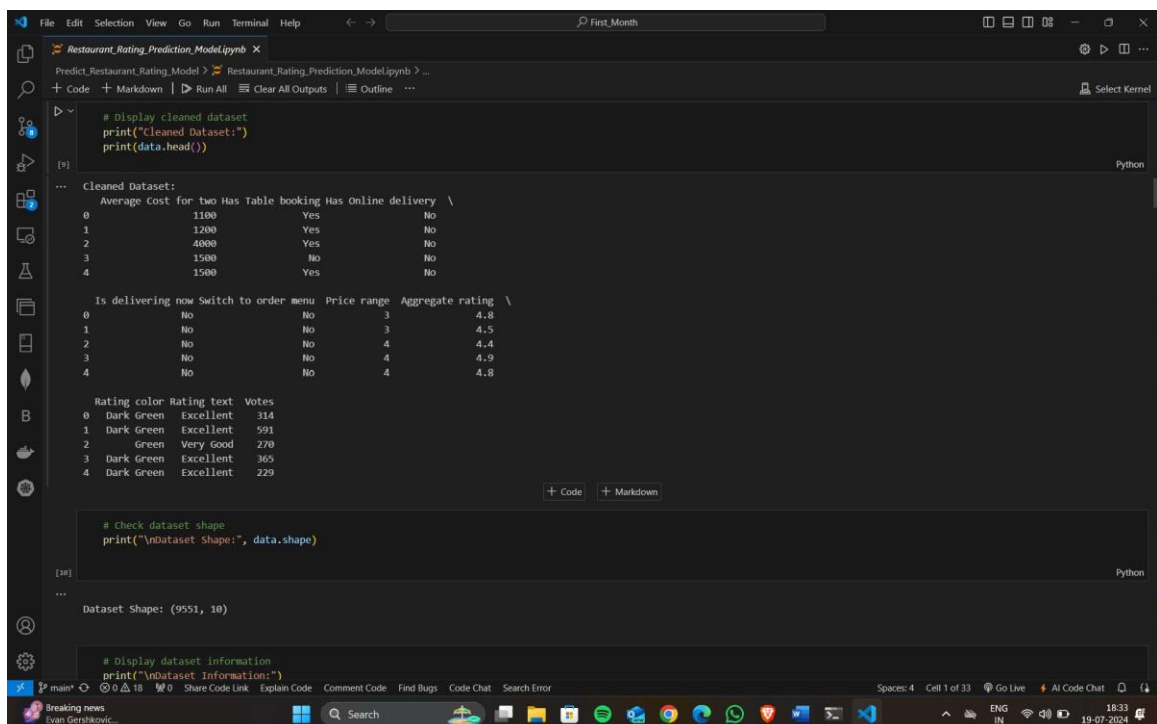
```
Cleaned Dataset:
Average Cost for two  Has Table booking  Has Online delivery  \
0                    1100                Yes                No
1                    1200                Yes                No
2                    4000                Yes                No
3                    1500                No                 No
4                    1500                Yes                No

Is delivering now  Switch to order menu  Price range  Aggregate rating  \
0                No                 No           3             4.8
1                No                 No           3             4.5
2                No                 No           4             4.4
3                No                 No           4             4.9
4                No                 No           4             4.8

Rating color  Rating text  Votes
0  Dark Green  Excellent    314
1  Dark Green  Excellent    591
2    Green    Very Good    270
3  Dark Green  Excellent    365
4  Dark Green  Excellent    229
```

3. Exploratory Data Analysis (EDA):

EDA involves a deep dive into the dataset to uncover hidden patterns, trends, and relationships between variables. It explores factors influencing restaurant ratings such as cuisine type, location, price range, and customer preferences. EDA techniques include univariate and bivariate analysis, hypothesis testing, and data visualization. Insights gleaned from EDA inform feature selection, model development, and interpretation of results.



```
# Display cleaned dataset
print("Cleaned Dataset:")
print(data.head())
```

```
Cleaned Dataset:
Average Cost for two  Has Table booking  Has Online delivery  \
0                    1100                Yes                No
1                    1200                Yes                No
2                    4000                Yes                No
3                    1500                No                 No
4                    1500                Yes                No

Is delivering now  Switch to order menu  Price range  Aggregate rating  \
0                No                 No           3             4.8
1                No                 No           3             4.5
2                No                 No           4             4.4
3                No                 No           4             4.9
4                No                 No           4             4.8

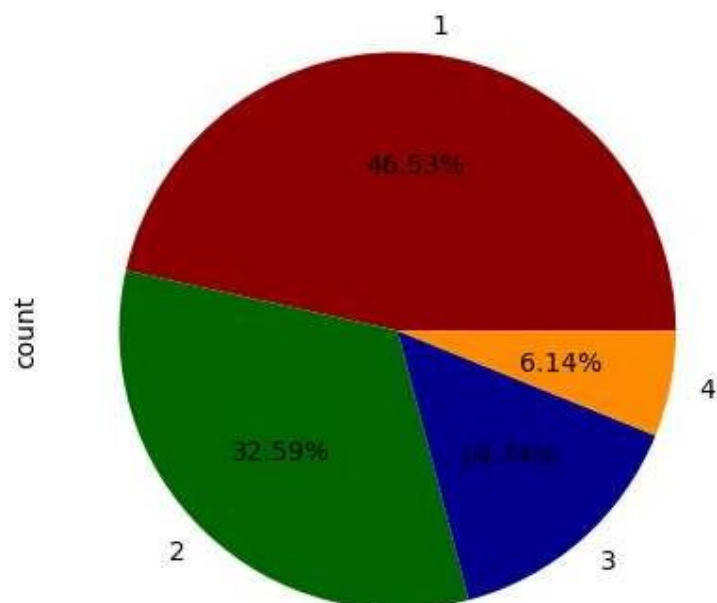
Rating color  Rating text  Votes
0  Dark Green  Excellent    314
1  Dark Green  Excellent    591
2    Green    Very Good    270
3  Dark Green  Excellent    365
4  Dark Green  Excellent    229
```

```
# Check dataset shape
print("Dataset Shape:", data.shape)
```

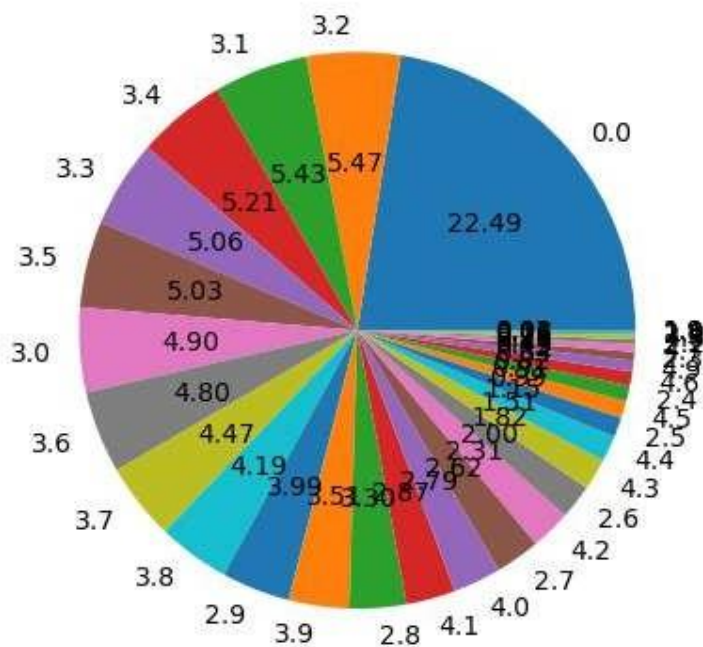
```
Dataset Shape: (9551, 10)
```

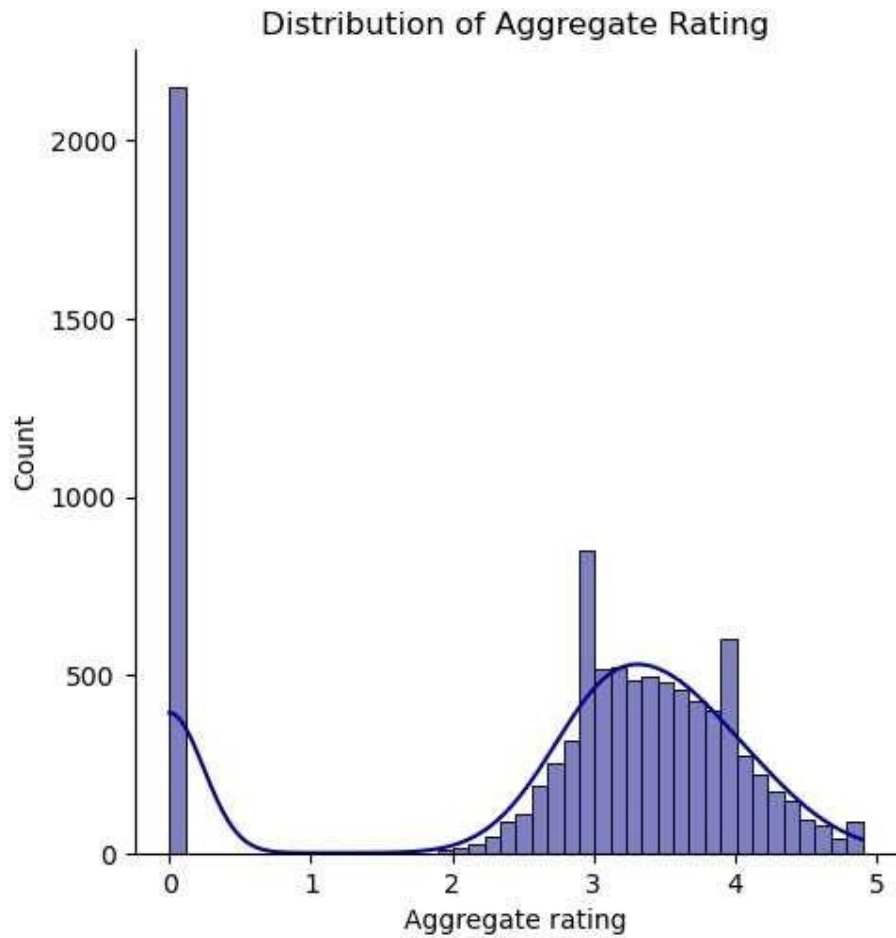
```
# Display dataset information
print("Dataset Information:")
```

Visualize price range distribution

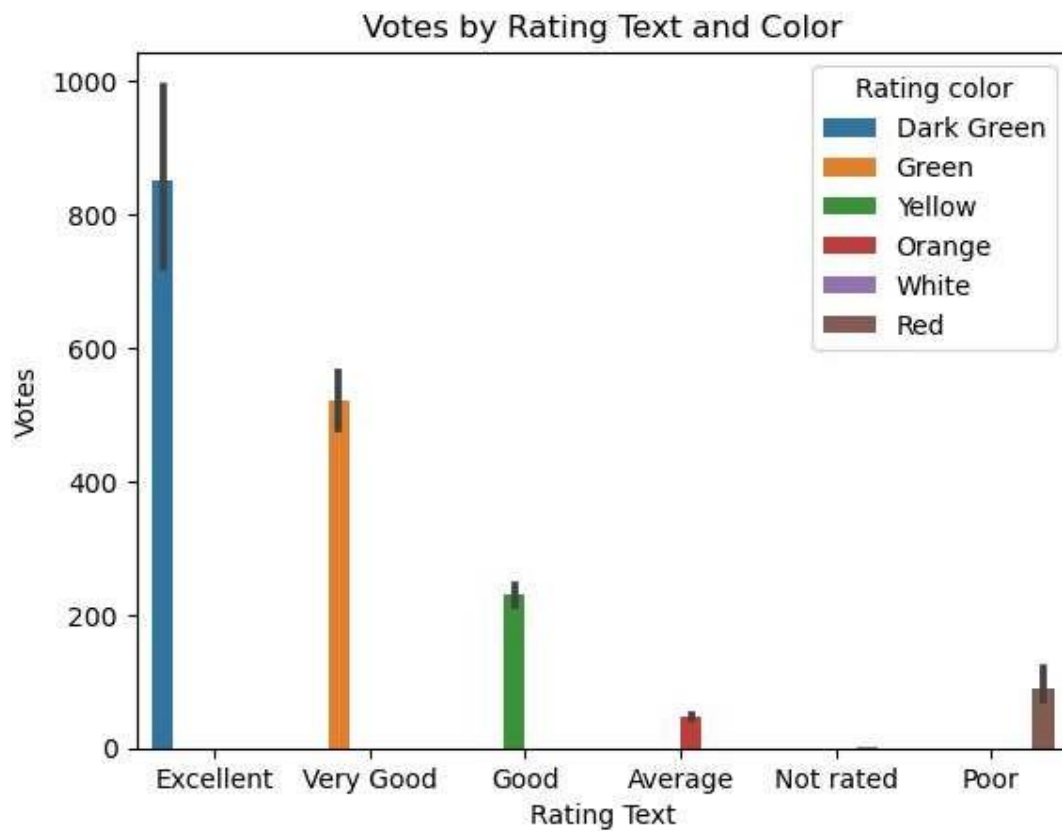


Aggregate Rating Distribution (Dark Rainbow Colors)



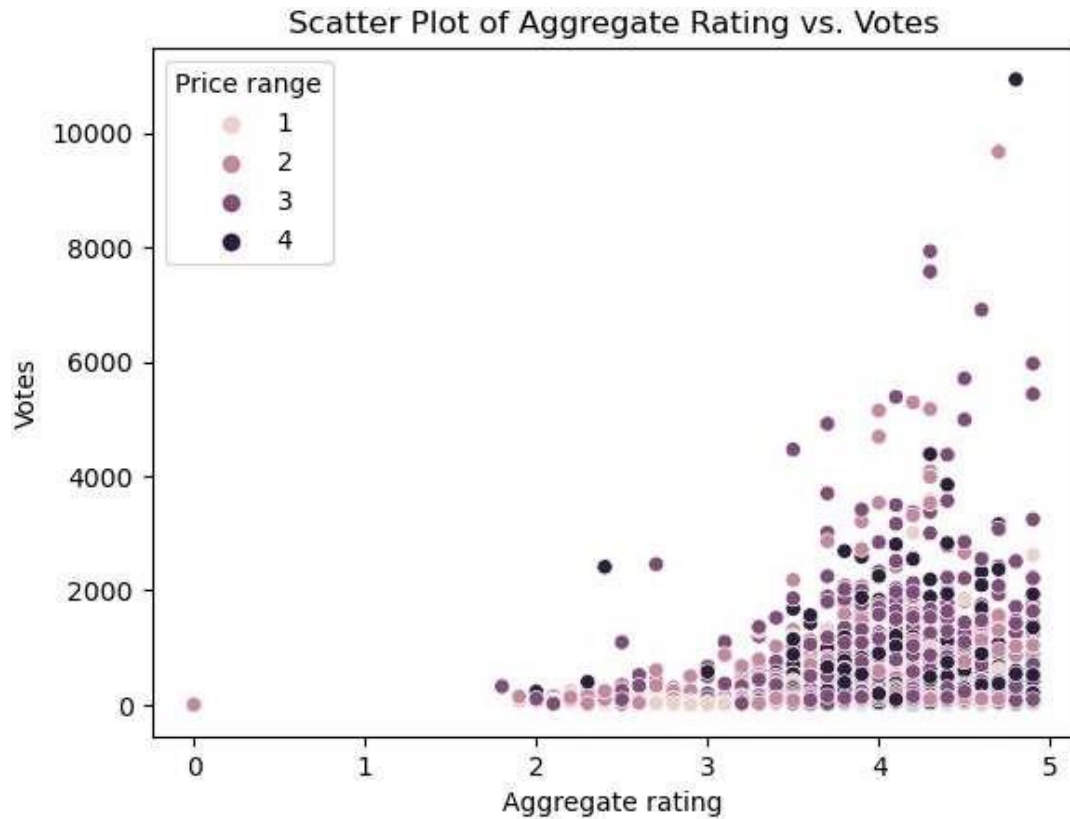


Bar plot of votes by rating text and color



4. Model Implementation:

The model implementation section details the selection, training, and evaluation of machine learning algorithms for predicting restaurant ratings. It discusses the rationale behind choosing regression models such as linear regression, decision tree regression, and ensemble methods. Model hyperparameters are fine tuned using techniques like cross validation and grid search to optimize performance. Model robustness and generalization are assessed through rigorous testing on unseen data.



```
File Edit Selection View Go Run Terminal Help
Restaurant_Rating_Prediction_Model.ipynb
Predict_Restaurant_Rating_Model > Restaurant_Rating_Prediction_Model.ipynb > ...
+ Code + Markdown | ▶ Run All | Clear All Outputs | Outline ...
Select Kernel

# Regression analysis

# Linear Regression
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import LabelEncoder

label_encoder = LabelEncoder()
data['Has Table booking'] = label_encoder.fit_transform(data['Has Table booking'])
data['Has Online delivery'] = label_encoder.fit_transform(data['Has Online delivery'])
data['Is delivering now'] = label_encoder.fit_transform(data['Is delivering now'])
data['Switch to order menu'] = label_encoder.fit_transform(data['Switch to order menu'])
data['Rating color'] = label_encoder.fit_transform(data['Rating color'])
data['Rating text'] = label_encoder.fit_transform(data['Rating text'])

# Display encoded dataset
print("\nEncoded Dataset:")
print(data.head())

Encoded Dataset:
Average Cost for two  Has Table booking  Has Online delivery \
0      1100             1             0
1      1200             1             0
2      4000             1             0
3      1500             0             0
4      1500             1             0

Is delivering now  Switch to order menu  Price range  Aggregate rating \
0      0             0             3             4.8
1      0             0             3             4.5
2      0             0             4             4.4
3      0             0             4             4.9
4      0             0             4             4.8

Rating color  Rating text  Votes
0      0             1      314
1      0             1      501
2      1             5      770
3      0             1      365
4      0             1      229

# correlation heatmap
correlation_matrix = data.corr()
plt.figure(figsize=(10, 8))
sb.heatmap(correlation_matrix, annot=True, cmap='viridis', fmt=".2f", linewidths=0.5)
plt.title('Correlation Heatmap')
plt.show()
```

```
File Edit Selection View Go Run Terminal Help
First_Month
Restaurant_Rating_Prediction_Model.ipynb
Predict_Restaurant_Rating_Model > Restaurant_Rating_Prediction_Model.ipynb > ...
+ Code + Markdown | ▶ Run All | Clear All Outputs | Outline ...
Select Kernel

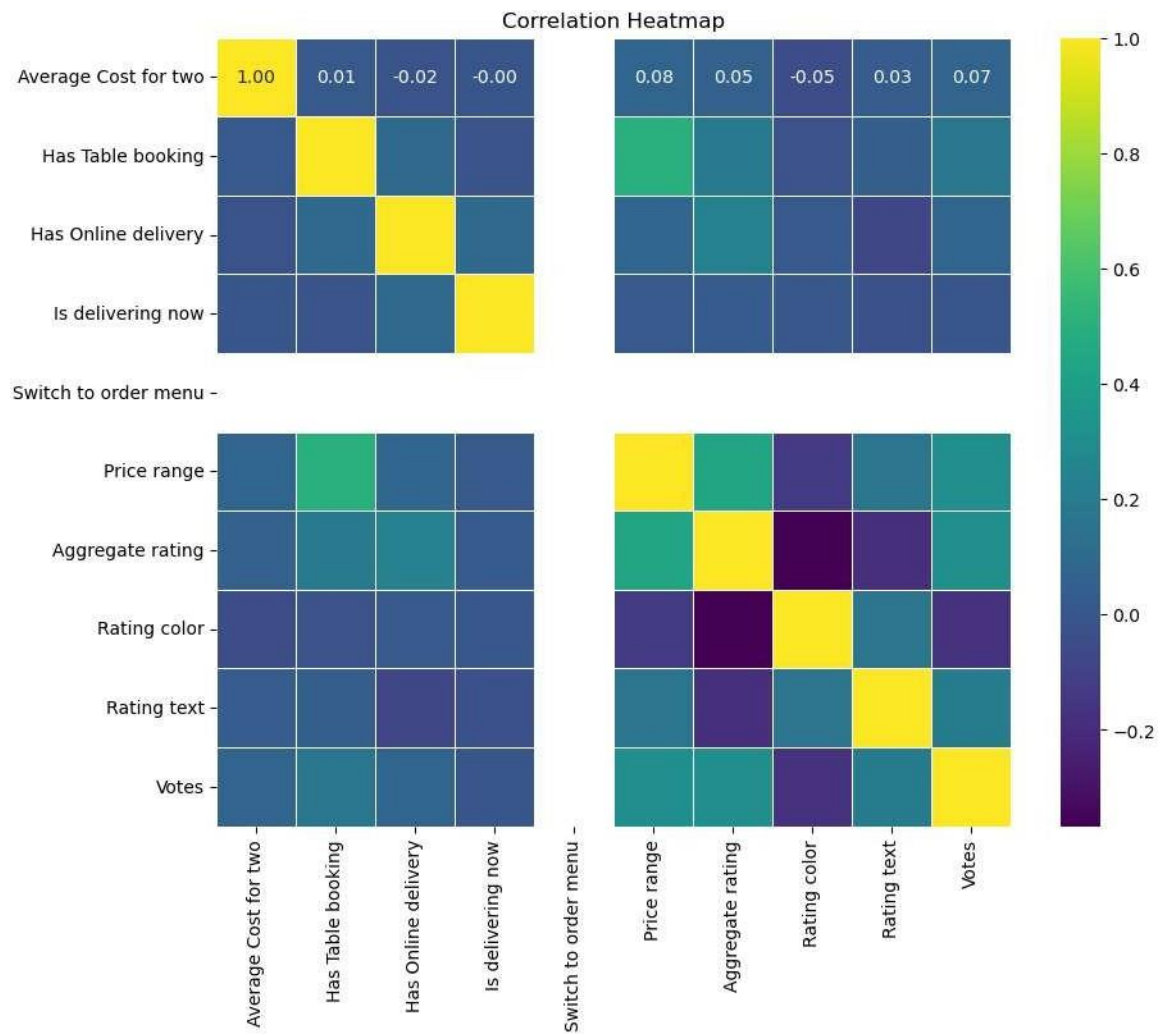
# Display encoded dataset
print("\nEncoded Dataset:")
print(data.head())

Encoded Dataset:
Average Cost for two  Has Table booking  Has Online delivery \
0      1100             1             0
1      1200             1             0
2      4000             1             0
3      1500             0             0
4      1500             1             0

Is delivering now  Switch to order menu  Price range  Aggregate rating \
0      0             0             3             4.8
1      0             0             3             4.5
2      0             0             4             4.4
3      0             0             4             4.9
4      0             0             4             4.8

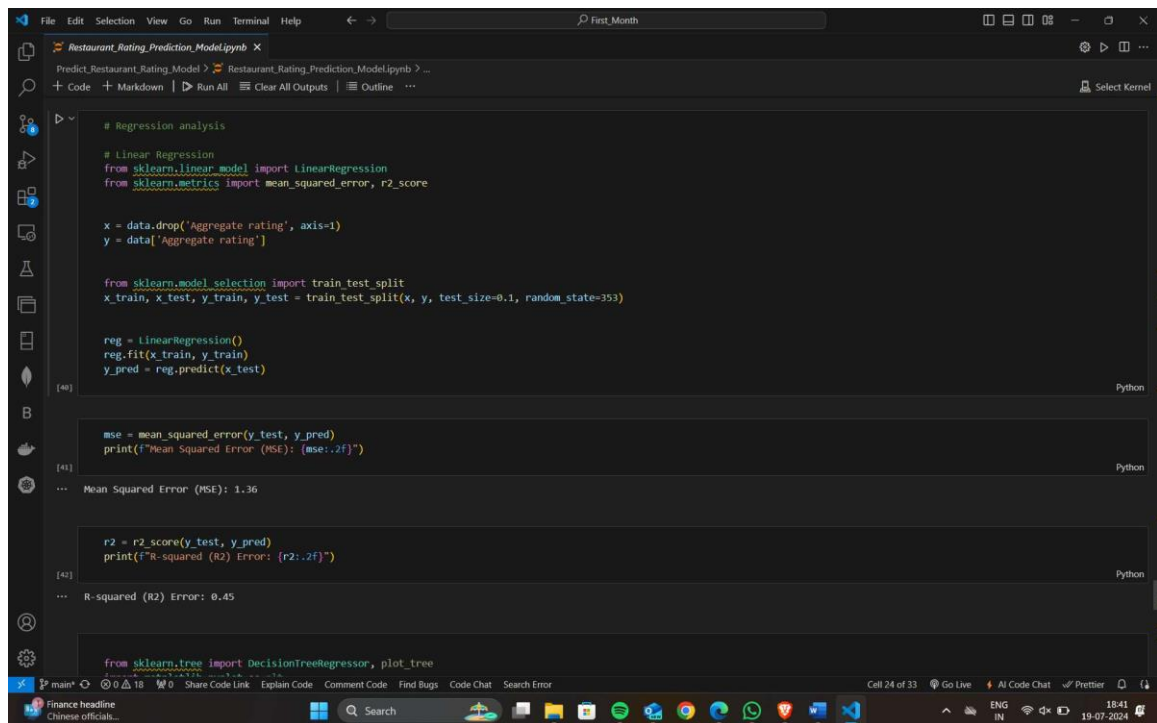
Rating color  Rating text  Votes
0      0             1      314
1      0             1      501
2      1             5      770
3      0             1      365
4      0             1      229

# correlation heatmap
correlation_matrix = data.corr()
plt.figure(figsize=(10, 8))
sb.heatmap(correlation_matrix, annot=True, cmap='viridis', fmt=".2f", linewidths=0.5)
plt.title('Correlation Heatmap')
plt.show()
```



5. Results and Analysis:

Results from the trained models are presented, showcasing performance metrics such as MSE, RMSE, and R2 score. The section provides a comprehensive analysis of model predictions, highlighting areas of strengths and limitations. Insights derived from feature importance analysis and model interpretations shed light on the factors driving restaurant ratings and actionable recommendations for stakeholders. Visualization techniques such as SHAP (SHapley Additive exPlanations) values and partial dependence plots are employed to explain model predictions and validate domain knowledge.



The screenshot displays a Jupyter Notebook interface with a dark theme. The notebook is titled "Restaurant_Rating_Prediction_Model.ipynb". The code in the first cell performs a linear regression analysis. It imports necessary libraries, drops the 'Aggregate rating' column from the data, and splits the data into training and testing sets. A linear regression model is trained on the training data and used to predict the ratings for the test set. The Mean Squared Error (MSE) is calculated and printed as 1.36. The R-squared (R2) score is also calculated and printed as 0.45. The second cell shows the start of a decision tree model implementation.

```
# Regression analysis

# Linear Regression
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

x = data.drop('Aggregate rating', axis=1)
y = data['Aggregate rating']

from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.1, random_state=353)

reg = LinearRegression()
reg.fit(x_train, y_train)
y_pred = reg.predict(x_test)

mse = mean_squared_error(y_test, y_pred)
print(f"Mean Squared Error (MSE): {mse:.2f}")

r2 = r2_score(y_test, y_pred)
print(f"R-squared (R2) Error: {r2:.2f}")

from sklearn.tree import DecisionTreeRegressor, plot_tree
```

Mean Squared Error (MSE): 1.36

R-squared (R2) Error: 0.45

Restaurant_Rating_Prediction_Model.ipynb X

Predict_Restaurant_Rating_Model > Restaurant_Rating_Prediction_Model.ipynb > ...

+ Code + Markdown | Run All | Clear All Outputs | Outline ...

Select Kernel

```
from sklearn.tree import DecisionTreeRegressor, plot_tree
import matplotlib.pyplot as plt

def train_and_evaluate_decision_tree(x, y, test_size=0.1, random_state=105, min_samples_leaf=0.0001):
    x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=test_size, random_state=random_state)
    Dtree = DecisionTreeRegressor(min_samples_leaf=min_samples_leaf)
    Dtree.fit(x_train, y_train)
    y_predict = Dtree.predict(x_test)
    r2 = r2_score(y_test, y_predict)
    print("R-squared score:", r2)
    train_and_evaluate_decision_tree(x, y)
```

[40] ... R-squared score: 0.977264671353274

```
# Decision Tree Regressor
from sklearn.tree import DecisionTreeRegressor
```

[41] ...

```
Dtree = DecisionTreeRegressor(min_samples_leaf=0.0001)
Dtree.fit(x_train, y_train)
y_predict = Dtree.predict(x_test)
```

[42] ...

```
mse = mean_squared_error(y_test, y_predict)
print(f"Mean Squared Error (MSE): {mse:.2f}")
```

[43] ... Mean Squared Error (MSE): 0.06

```
r2 = r2_score(y_test, y_predict)
```

main* 0 18 0 Share Code Link Explain Code Comment Code Find Bugs Code Chat Search Error Cell 24 of 33 Go Live AI Code Chat 18:42

Finance headline Chinese officials...

Restaurant_Rating_Prediction_Model.ipynb X

Predict_Restaurant_Rating_Model > Restaurant_Rating_Prediction_Model.ipynb > ...

+ Code + Markdown | Run All | Clear All Outputs | Outline ...

Select Kernel

```
r2 = r2_score(y_test, y_predict)
print(f"R-squared (R2) Error: {r2:.2f}")
```

[47] ... R-squared (R2) Error: 0.98

```
# Conclusion
print("\nConclusion: Decision Tree Regressor model is performing exceptionally well on the test data.")
```

[48] ...

Conclusion: Decision Tree Regressor model is performing exceptionally well on the test data.

main* 0 18 0 Share Code Link Explain Code Comment Code Find Bugs Code Chat Search Error Cell 24 of 33 Go Live AI Code Chat 18:42

Finance headline Chinese officials...

Timeline:

■ Week 1: Project Setup and Data Collection

- ✓ Day 1: Understand project requirements and objectives.
- ✓ Day 2 3: Gather relevant datasets on restaurant ratings and related features.
- ✓ Day 4 5: Explore available datasets and select the most suitable one for the project.
- ✓ Day 6 7: Clean and preprocess the dataset, handling missing values and encoding categorical variables.

■ Week 2: Exploratory Data Analysis (EDA) and Feature Engineering

- ✓ Day 8: Perform exploratory data analysis (EDA) to understand the distribution and relationships between variables.
- ✓ Day 9 10: Visualize data using histograms, scatter plots, and correlation matrices.
- ✓ Day 11 12: Engineer new features and select relevant features for model training.
- ✓ Day 13 14: Conduct further data cleaning and preprocessing if necessary.

■ Week 3: Model Development and Evaluation

- ✓ Day 15: Select appropriate regression algorithms for predicting restaurant ratings (e.g., linear regression, decision tree regression).
- ✓ Day 16 17: Split the dataset into training and testing sets.
- ✓ Day 18 19: Train baseline models and evaluate their performance using metrics such as mean squared error (MSE) and R squared (R²) score.
- ✓ Day 20 21: Fine tune hyperparameters and optimize model performance using techniques like cross validation and grid search.

■ Week 4: Model Interpretation and Documentation

- ✓ Day 22: Interpret model results and analyze feature importance.
- ✓ Day 23 24: Document findings, insights, and recommendations in the project report.
- ✓ Day 25 26: Prepare visualizations and figures for the project report.
- ✓ Day 27 28: Review and finalize the project report, ensuring clarity and coherence.
- ✓ Day 29 30: Prepare for presentation or demonstration of the project to stakeholders.

6. Conclusion:

The conclusion summarizes the key findings and contributions of the project, emphasizing its implications for the restaurant industry. It underscores the significance of leveraging data analytics and machine learning techniques for enhancing decision making processes and improving business outcomes. The conclusion also discusses avenues for future research, such as exploring advanced modeling techniques, integrating real time data streams, and deploying the model in production environments.

7. Future Work:

This section outlines potential directions for future work and research extensions. It discusses opportunities for enhancing model performance, such as incorporating user generated content from social media platforms, integrating sentiment analysis and natural language processing techniques, and developing personalized recommendation systems. Collaboration with industry partners and stakeholders is encouraged to further validate and deploy the model in real world settings.

8. References:

- Garg, N., & Chaurasia, V. (2019). Predicting Restaurant Rating Using Machine Learning Algorithms. *International Journal of Computer Science and Mobile Computing*, 8 (5), 40 48.
- Aggarwal, A., Gupta, S., & Singla, A. (2020). Machine Learning Approach for Predicting Restaurant Ratings. *International Journal of Innovative Technology and Exploring Engineering*, 9(1), 2635 2639.
- Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12 , 2825 2830.
- McKinney, W., & others. (2010). Data Structures for Statistical Computing in Python. *Proceedings of the 9th Python in Science Conference* , 51 56.
- Seabold, S., & Perktold, J. (2010). Statsmodels: Econometric and Statistical Modeling with Python. *Proceedings of the 9th Python in Science Conference* , 92.

Appendices:

Supplementary information, code snippets, and additional analyses that complement the main report are included in the appendices. This may include detailed methodology descriptions, model implementation code, experimental results, and sensitivity analyses. Appendices serve as a valuable resource for readers interested in replicating or extending the project.

