RESTAURANT RATING PREDICTION USING ML

By-Manav Shah

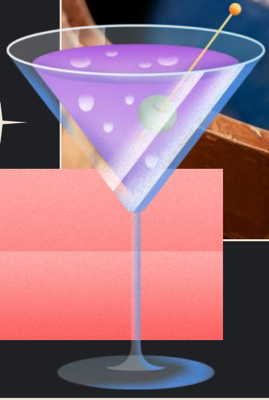

TABLE OF CONTENTS

01

PROBLEM STATEMENT

PROBLEM STATEMENT

Develop a machine learning model to predict the aggregate rating of a restaurant based on various features related to the restaurant's characteristics and operations.

OBJECTIVE

Predicting Restaurant Ratings

02

DATASET OVERVIEW

DATASET OVERVIEW

The dataset contains the following columns:

1. Restaurant ID: Unique identifier for each restaurant.
2. Restaurant Name: Name of the restaurant.
3. Country Code: Numeric code representing the country where the restaurant is located.
4. City: Name of the city where the restaurant is situated.
5. Address: Physical address of the restaurant.
6. Locality: Locality or neighborhood where the restaurant is located.
7. Locality Verbose: Detailed description of the locality.
8. Longitude: Geographical longitude of the restaurant's location.
9. Latitude: Geographical latitude of the restaurant's location.
10. Cuisines: Types of cuisines offered by the restaurant.
11. Average Cost for Two: Average cost for a meal for two people.
12. Currency: Currency used for transactions in the restaurant.

DATASET OVERVIEW

13. Has Table Booking: Indicator of whether the restaurant accepts table bookings.
14. Has Online Delivery: Indicator of whether the restaurant offers online delivery.
15. Is Delivering Now: Indicator of whether the restaurant is currently delivering.
16. Switch to Order Menu: Indicator of whether the restaurant has switched to an order menu.
17. Price Range: Price range category of the restaurant.
18. Aggregate Rating: Overall rating of the restaurant.
19. Rating Color: Color code representing the rating.
20. Rating Text: Text description of the rating.
21. Votes: Number of votes received by the restaurant.

03

IMPORTING DATA

```
[1] import pandas as pd
    import numpy as np
    from sklearn.model_selection import train_test_split
    from sklearn.preprocessing import StandardScaler, LabelEncoder
    from sklearn.ensemble import RandomForestRegressor
    from sklearn.linear_model import LinearRegression
    from sklearn.svm import SVR
    from sklearn.tree import DecisionTreeRegressor
    from sklearn.metrics import mean_squared_error, r2_score
```

```
[2] data = pd.read_csv('/content/Restaurant Rating Dataset.csv')
```

```
0s  # Display first few rows
    print(data.head())
```

```
   Restaurant ID         Restaurant Name  Country Code
0        6317637        Le Petit Souffle           162        Makat
1        6304287        Izakaya Kikufuji           162        Makat
2        6300002  Heat - Edsa Shangri-La           162  Mandaluyon
3        6318506                    Ooma           162  Mandaluyon
4        6314302             Sambo Kojin           162  Mandaluyon
```

04

DATA PREPROCESSING

CHECKING FOR NULL VALUES AND HANDELLING :

```python
data['Cuisines'].fillna(data['Cuisines'].mode()[0], inplace=True)
```

```python
# Check for missing values
print(data.isnull().sum())
```

```
Restaurant ID           0
Restaurant Name         0
Country Code            0
City                    0
Address                 0
Locality                0
Locality Verbose        0
Longitude               0
Latitude                0
Cuisines                9
Average Cost for two    0
Currency                0
Has Table booking       0
Has Online delivery     0
Is delivering now       0
Switch to order menu    0
Price range             0
Aggregate rating        0
Rating color            0
Rating text             0
Votes                   0
dtype: int64
```

LABEL ENCODING FOR CATEGORICAL COLUMNS

```
# Label Encoding for categorical columns
le = LabelEncoder()
data['Cuisines'] = le.fit_transform(data['Cuisines'])
data['City'] = le.fit_transform(data['City'])
data['Currency'] = le.fit_transform(data['Currency'])
```

REMOVING UNNECESSARY COLUMNS AND FEATURE SCALING

```
[6] # Drop unnecessary columns
    data.drop(['Restaurant ID', 'Restaurant Name', 'Address', 'Locality', 'Locality Verbose',
```

```
#feature scaling
scaler = StandardScaler()
numerical_features = ['Average Cost for two', 'Latitude', 'Longitude', 'Votes']
data[numerical_features] = scaler.fit_transform(data[numerical_features])
```

05

MODEL IMPLEMENTATION

DEFINING FEATURES AND TARGETS AND SPLIT INTO TRAIN TEST DATA

```
[8]  # Define features and target
     X = data.drop('Aggregate rating', axis=1)
     y = data['Aggregate rating']

[9]  # Split the data into train and test sets
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

IMPLEMENTING DECISION TREE REGRESSOR

0s

```python
# Decision Tree Regressor
dt = DecisionTreeRegressor(random_state=42)
dt.fit(X_train, y_train)

# Predictions
y_pred_dt = dt.predict(X_test)

# Evaluation
mse_dt = mean_squared_error(y_test, y_pred_dt)
r2_dt = r2_score(y_test, y_pred_dt)
print(f'Decision Tree - MSE: {mse_dt}, R2: {r2_dt}')
```

```
Decision Tree - MSE: 0.17420198848770277, R2: 0.9234650057491243
```

IMPLEMENTING RANDOM FOREST REGRESSOR

```python
rf = RandomForestRegressor(random_state=42, n_estimators=100)
rf.fit(X_train, y_train)

# Predictions
y_pred_rf = rf.predict(X_test)

# Evaluation
mse_rf = mean_squared_error(y_test, y_pred_rf)
r2_rf = r2_score(y_test, y_pred_rf)
print(f'Random Forest - MSE: {mse_rf}, R2: {r2_rf}')
```

```
Random Forest - MSE: 0.08747064207221349, R2: 0.9615700994791466
```

06

CONCLUSION

CONCLUSION

The **Random Forest Regressor** outperforms the **Decision Tree Regressor** in both metrics:

- The **MSE** is lower for the Random Forest Regressor (0.0875 vs. 0.1742), indicating that the model's predictions are, on average, closer to the actual values.
- The **R^2** score is higher for the Random Forest Regressor (0.9616 vs. 0.9235), which means that it explains a larger proportion of the variance in the target variable.

RECOMMENDATION

Given these results, **the Random Forest Regressor is the better model for predicting restaurant ratings**. It provides higher accuracy and better generalization to the test data, making it the preferred choice for the prediction task.

THANKS!

Do you have any questions?

manavshah1704@gmail.com