```python
# Basic libraries
import pandas as pd
import numpy as np

# For model and preprocessing
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Optional for visualization
import matplotlib.pyplot as plt
import seaborn as sns

# Replace this with your actual file name
df = pd.read_csv("C:/Users/91832/Downloads/Dataset .csv")

# Show first 5 rows to understand data
df.head()
```

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

                                              Address  \
0  Third Floor, Century City Mall, Kalayaan Avenu...
1  Little Tokyo, 2277 Chino Roces Avenue, Legaspi...
2  Edsa Shangri-La, 1 Garden Way, Ortigas, Mandal...
3  Third Floor, Mega Fashion Hall, SM Megamall, O...
4  Third Floor, Mega Atrium, SM Megamall, Ortigas...

                                    Locality  \
0   Century City Mall, Poblacion, Makati City
1  Little Tokyo, Legaspi Village, Makati City
2  Edsa Shangri-La, Ortigas, Mandaluyong City
3      SM Megamall, Ortigas, Mandaluyong City
4      SM Megamall, Ortigas, Mandaluyong City

                                   Locality Verbose   Longitude
Latitude  \
0  Century City Mall, Poblacion, Makati City, Mak...  121.027535
14.565443
```

```
1  Little Tokyo, Legaspi Village, Makati City, Ma...  121.014101
14.553708
2  Edsa Shangri-La, Ortigas, Mandaluyong City, Ma...  121.056831
14.581404
3  SM Megamall, Ortigas, Mandaluyong City, Mandal...  121.056475
14.585318
4  SM Megamall, Ortigas, Mandaluyong City, Mandal...  121.057508
14.584450

                            Cuisines  ...          Currency Has Table
booking  \
0        French, Japanese, Desserts  ...  Botswana Pula(P)
Yes
1                          Japanese  ...  Botswana Pula(P)
Yes
2  Seafood, Asian, Filipino, Indian  ...  Botswana Pula(P)
Yes
3                   Japanese, Sushi  ...  Botswana Pula(P)
No
4                  Japanese, Korean  ...  Botswana Pula(P)
Yes

  Has Online delivery Is delivering now Switch to order menu Price
range  \
0                  No                 No                       No
3
1                  No                 No                       No
3
2                  No                 No                       No
4
3                  No                 No                       No
4
4                  No                 No                       No
4

   Aggregate rating  Rating color Rating text Votes
0               4.8    Dark Green   Excellent   314
1               4.5    Dark Green   Excellent   591
2               4.4         Green   Very Good   270
3               4.9    Dark Green   Excellent   365
4               4.8    Dark Green   Excellent   229

[5 rows x 21 columns]
```

```python
# Fill missing numeric values with the mean
df.fillna(df.mean(numeric_only=True), inplace=True)

# Fill missing text/categorical values with mode
for col in df.select_dtypes(include='object').columns:
    df[col].fillna(df[col].mode()[0], inplace=True)
```

```
# Convert text columns to numbers using LabelEncoder
label_encoder = LabelEncoder()
for col in df.select_dtypes(include='object').columns:
    df[col] = label_encoder.fit_transform(df[col])

# Check actual column names
print(df.columns)
```

```
Index(['Restaurant ID', 'Restaurant Name', 'Country Code', 'City',
'Address',
       'Locality', 'Locality Verbose', 'Longitude', 'Latitude',
'Cuisines',
       'Average Cost for two', 'Currency', 'Has Table booking',
       'Has Online delivery', 'Is delivering now', 'Switch to order
menu',
       'Price range', 'Aggregate rating', 'Rating color', 'Rating
text',
       'Votes'],
      dtype='object')
```

```
# Use correct column name for rating (update this if needed)
target_column = "Aggregate rating"

# change if it's like "Aggregate rating"

X = df.drop("Aggregate rating", axis=1)
y = df["Aggregate rating"]

# Split into Train and Test
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Train a Simple Model
model = LinearRegression()
model.fit(X_train, y_train)
```

```
LinearRegression()
```

```
# Predict and Evaluate
y_pred = model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print("Mean Squared Error (MSE):", round(mse, 2))
print("R² Score:", round(r2, 2))
```

```
Mean Squared Error (MSE): 1.22
R² Score: 0.46
```

```
# Plot feature importance based on coefficients
coeffs = pd.Series(model.coef_, index=X.columns)
```

```
coeffs = coeffs.sort_values()

plt.figure(figsize=(10,6))
coeffs.plot(kind='barh', color='skyblue')
plt.title("Feature Influence on Rating (Linear Regression
Coefficients)")
plt.xlabel("Coefficient Value")
plt.ylabel("Feature")
plt.grid(True)
plt.tight_layout()
plt.show()
```