Task 1: Predict Restaurant Ratings using Regression

Objective

Build a regression model to predict the **Aggregate Rating** of a restaurant based on various features like cuisine, location, cost, and other metadata.

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt
import seaborn as sns

# Mount Google Drive
from google.colab import drive
drive.mount('/content/drive')

# Load Dataset from Google Drive
file_path = '/content/drive/My Drive/ML_Internship/resturant_dataset.csv'
df = pd.read_csv(file_path)

# Preview first few rows
df.head()
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude	Cuisines	...	Currency	Tal book
0	6317637	Le Petit Souffle	162	Makati City	Third Floor, Century City Mall, Kalayaan Avenu...	Century City Mall, Poblacion, Makati City	Century City Mall, Poblacion, Makati City, Mak...	121.027535	14.565443	French, Japanese, Desserts	...	Botswana Pula(P)	.
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)	.
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)	.
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)	
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)	.

5 rows × 21 columns

Step 1: Data Preprocessing

- Handle missing values
- Encode categorical features
- Select relevant features for prediction

```
# Drop rows with missing cuisines
df = df.dropna(subset=['Cuisines'])

# Select relevant columns
data = df[['Average Cost for two', 'Has Table booking', 'Has Online delivery',
           'Price range', 'Votes', 'Cuisines', 'Aggregate rating']].copy()

# Encode binary columns
data['Has Table booking'] = data['Has Table booking'].map({'Yes': 1, 'No': 0})
data['Has Online delivery'] = data['Has Online delivery'].map({'Yes': 1, 'No': 0})

# Encode top 10 cuisines, rest as 'Other'
top_cuisines = data['Cuisines'].value_counts().nlargest(10).index
data['Cuisines'] = data['Cuisines'].apply(lambda x: x if x in top_cuisines else 'Other')
data = pd.get_dummies(data, columns=['Cuisines'], drop_first=True)

data.head()
```

	Average Cost for two	Has Table booking	Has Online delivery	Price range	Votes	Aggregate rating	Cuisines_Bakery, Desserts	Cuisines_Cafe	Cuisines_Chinese	Cuisines_Fast Food	Cuisines_North Indian
0	1100	1	0	3	314	4.8	False	False	False	False	False
1	1200	1	0	3	591	4.5	False	False	False	False	False
2	4000	1	0	4	270	4.4	False	False	False	False	False
3	1500	0	0	4	365	4.9	False	False	False	False	False
4	1500	1	0	4	229	4.8	False	False	False	False	False

Next steps: Generate code with data | View recommended plots | New interactive sheet

Step 2: Train-Test Split

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

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Step 3: Train and Evaluate Models

```
# Linear Regression
lr = LinearRegression()
lr.fit(X_train, y_train)
y_pred_lr = lr.predict(X_test)

print("Linear Regression R2 Score:", r2_score(y_test, y_pred_lr))
print("Linear Regression MSE:", mean_squared_error(y_test, y_pred_lr))
```

```
Linear Regression R2 Score: 0.30675107783544675
Linear Regression MSE: 1.5875623551619147
```

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

print("Decision Tree R2 Score:", r2_score(y_test, y_pred_dt))
print("Decision Tree MSE:", mean_squared_error(y_test, y_pred_dt))
```

```
Decision Tree R2 Score: 0.9057745050522386
Decision Tree MSE: 0.21577941759865893
```

Step 4: Feature Importance Analysis

```
importances = pd.Series(dt.feature_importances_, index=X.columns)
importances.sort_values().plot(kind='barh', figsize=(10,6))
plt.title('Feature Importances from Decision Tree')
plt.show()
```