Importing Libraries

- **TASK-1 -->PREDICT RESTAURANT RATING**
- **Importing Libraries**

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score,accuracy_score
from sklearn.preprocessing import LabelEncoder
```

Uploading Dataset

```
df= pd.read_csv('/content/Dataset .csv')
```

```
df.head()
```

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude	Cuisines	...	Currency
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

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9551 entries, 0 to 9550
Data columns (total 21 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   Restaurant ID         9551 non-null   int64
 1   Restaurant Name       9551 non-null   object
 2   Country Code          9551 non-null   int64
 3   City                  9551 non-null   object
 4   Address               9551 non-null   object
 5   Locality              9551 non-null   object
```

```
 6   Locality Verbose      9551 non-null   object
 7   Longitude             9551 non-null   float64
 8   Latitude              9551 non-null   float64
 9   Cuisines              9542 non-null   object
 10  Average Cost for two  9551 non-null   int64
 11  Currency              9551 non-null   object
 12  Has Table booking     9551 non-null   object
 13  Has Online delivery   9551 non-null   object
 14  Is delivering now     9551 non-null   object
 15  Switch to order menu  9551 non-null   object
 16  Price range           9551 non-null   int64
 17  Aggregate rating      9551 non-null   float64
 18  Rating color          9551 non-null   object
 19  Rating text           9551 non-null   object
 20  Votes                 9551 non-null   int64
dtypes: float64(3), int64(5), object(13)
memory usage: 1.5+ MB
```

Null Value Inspection

```
df.isnull().sum()
```

	0
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

Duplicate Value Inspection

```
df.duplicated().sum()
```

```
np.int64(0)
```

```
desc = df.dtypes
```

Converting Categorical Text Labels Into Numerical Values

```
le=LabelEncoder()
```

```
for col in df.columns:
  if df[col].dtype=='object':
    df[col]=le.fit_transform(df[col])
```

```
df
```

	Restaurant ID	Restaurant Name	Country Code	City	Address	Locality	Locality Verbose	Longitude	Latitude	Cuisines	...	Currency	Has Table booking	C del
0	6317637	3748	162	73	8685	171	172	121.027535	14.565443	920	...	0	1	
1	6304287	3172	162	73	6055	593	601	121.014101	14.553708	1111	...	0	1	
2	6300002	2896	162	75	4684	308	314	121.056831	14.581404	1671	...	0	1	
3	6318506	4707	162	75	8690	862	875	121.056475	14.585318	1126	...	0	0	
4	6314302	5523	162	75	8689	862	875	121.057508	14.584450	1122	...	0	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
9546	5915730	4443	208	140	5926	517	523	28.977392	41.022793	1813	...	11	0	
9547	5908749	1310	208	140	5962	552	558	29.041297	41.009847	1824	...	11	0	
9548	5915807	3068	208	140	5966	554	561	29.034640	41.055817	1110	...	11	0	
9549	5916112	512	208	140	5967	554	561	29.036019	41.057979	1657	...	11	0	
9550	5927402	7240	208	140	4258	670	681	29.026016	40.984776	331	...	11	0	

9551 rows × 21 columns

Removing Unwanted Data

```
df = df.drop(columns=['Restaurant ID', 'Restaurant Name', 'Address', 'Rating color', 'Rating text', 'Locality Verbose'
])
```

```
df['Cuisines'].fillna('Unknown', inplace=True)
```

```
/tmp/ipython-input-12-863532844.py:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col

  df['Cuisines'].fillna('Unknown', inplace=True)
```

```
df
```

	Country Code	City	Locality	Longitude	Latitude	Cuisines	Average Cost for two	Currency	Has Table booking	Has Online delivery	Is delivering now	Switch to order menu	Price range	Aggre ra
0	162	73	171	121.027535	14.565443	920	1100	0	1	0	0	0	3	
1	162	73	593	121.014101	14.553708	1111	1200	0	1	0	0	0	3	
2	162	75	308	121.056831	14.581404	1671	4000	0	1	0	0	0	4	
3	162	75	862	121.056475	14.585318	1126	1500	0	0	0	0	0	4	
4	162	75	862	121.057508	14.584450	1122	1500	0	1	0	0	0	4	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
9546	208	140	517	28.977392	41.022793	1813	80	11	0	0	0	0	3	
9547	208	140	552	29.041297	41.009847	1824	105	11	0	0	0	0	3	
9548	208	140	554	29.034640	41.055817	1110	170	11	0	0	0	0	4	
9549	208	140	554	29.036019	41.057979	1657	120	11	0	0	0	0	4	
9550	208	140	670	29.026016	40.984776	331	55	11	0	0	0	0	2	

9551 rows × 15 columns

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

Data Separation

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

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

	Country Code	City	Locality	Longitude	Latitude	Cuisines	Average Cost for two	Currency	Has Table booking	Has Online delivery	Is delivering now	Switch to order menu	Price range	Votes
4731	1	88	540	77.128443	28.651778	1306	350	4	0	0	0	0	1	54
1468	1	50	362	77.095432	28.460444	1284	700	4	0	1	0	0	2	84
9037	1	89	1014	77.340449	28.585474	1514	550	4	0	0	0	0	2	36
7866	1	88	1149	77.201128	28.692000	828	200	4	0	1	0	0	1	163
5570	1	88	687	77.216130	28.712062	1514	400	4	0	0	0	0	1	14
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8149	1	89	478	77.334786	28.575916	1520	600	4	0	0	0	0	2	44
5849	1	88	706	77.251202	28.547266	1306	400	4	0	0	0	0	1	1
9019	1	89	965	77.381743	28.520004	518	500	4	0	0	0	0	2	10
742	1	14	848	77.608179	12.972532	625	1200	4	0	0	0	0	3	334
180	216	23	466	-91.534100	41.661000	1022	40	2	0	0	0	0	3	428

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

Model **Selection**

```
model = RandomForestRegressor(random_state=42)
```

Data Fitting Process

```
model.fit(X_train, y_train)
```

```
RandomForestRegressor
RandomForestRegressor(random_state=42)
```

Predicting Values

```
predictions = model.predict(X_test)
```

```
check_1 = np.array([88, 540, 77.128443, 28.651778, 1306, 350, 4, 0, 0, 0, 0, 1, 1, 54]).reshape(1,-1)
check_2=np.array([50, 362, 77.095432, 28.460444, 1284, 700, 4, 0, 1, 0, 0, 2, 2,84]).reshape(1,-1)
print("Rating=",model.predict(check_1),"/5")
```

```
Rating= [3.711] /5
/usr/local/lib/python3.11/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but Rand
  warnings.warn(
```

```
print("Rating=",model.predict(check_2),"/5")
```

```
Rating= [3.648] /5
/usr/local/lib/python3.11/dist-packages/sklearn/utils/validation.py:2739: UserWarning: X does not have valid feature names, but Rand
  warnings.warn(
```

Performance Metrics

```
mse = mean_squared_error(y_test, predictions)*100
r2 = r2_score(y_test, predictions)*100
print("Mean Squared Error:", mse)
print("R² Score:", r2)
```

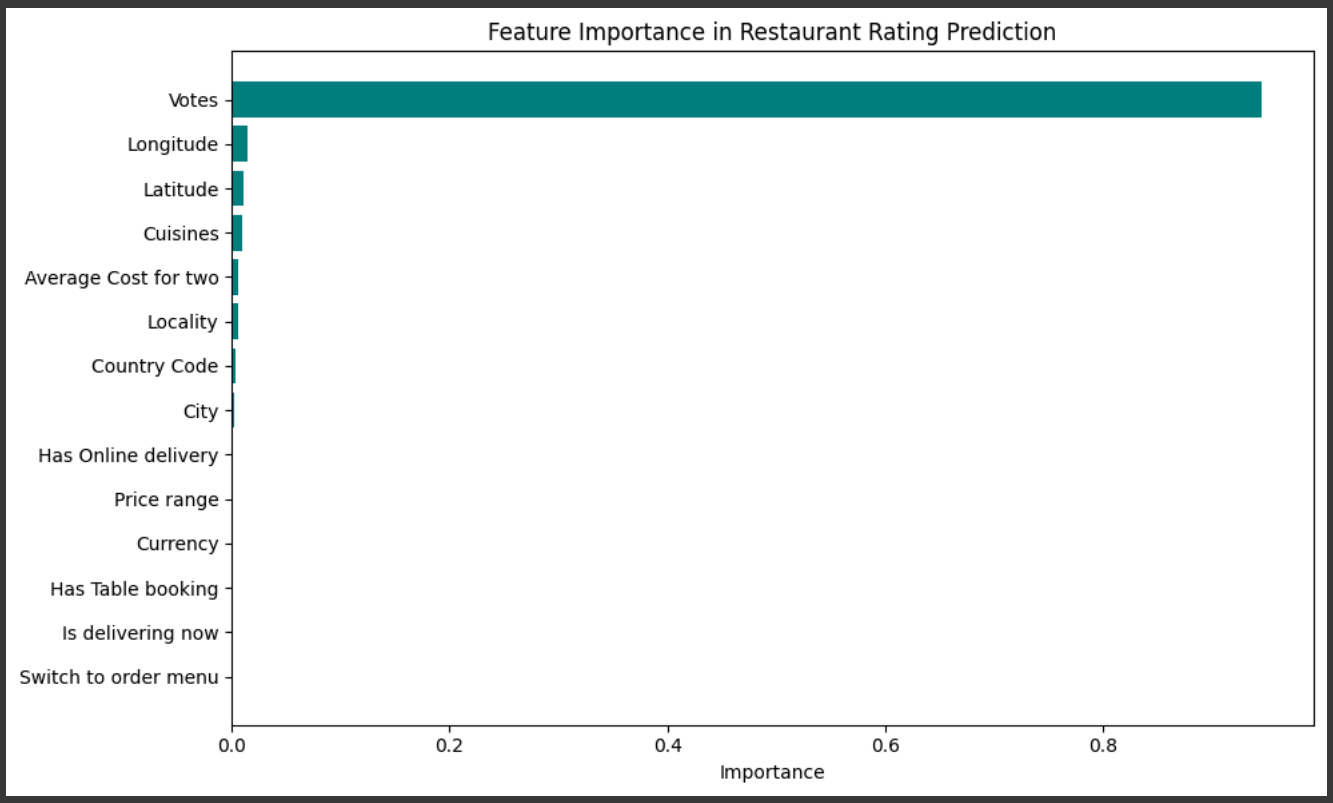
```
Mean Squared Error: 8.665819623233906
R² Score: 96.1927044530255
```

Analysis

```
importances = model.feature_importances_
feature_names = X.columns
```

```
feat_imp_df = pd.DataFrame({'Feature': feature_names,'Importance': importances})
feat_imp_df = feat_imp_df.sort_values(by='Importance', ascending=True)
```

```
plt.figure(figsize=(10, 6))
plt.barh(feat_imp_df['Feature'], feat_imp_df['Importance'], color='teal')
plt.xlabel('Importance')
plt.title('Feature Importance in Restaurant Rating Prediction')
plt.tight_layout()
plt.show()
```



Start coding or generate with AI.