

```
In [1]: # =====  
# LEVEL 3 - ALL TASKS  
# =====
```

```
In [2]: import pandas as pd  
import matplotlib.pyplot as plt  
from sklearn.model_selection import train_test_split  
from sklearn.linear_model import LinearRegression  
from sklearn.tree import DecisionTreeRegressor  
from sklearn.ensemble import RandomForestRegressor  
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
```

```
In [3]: df = pd.read_csv("C:/Users/jadha/Downloads/Dataset .csv")  
df['Cuisines'] = df['Cuisines'].fillna("Not Specified")
```

```
In [4]: # Feature Engineering  
df['Name_Length'] = df['Restaurant Name'].apply(len)  
df['Has_Table_Booking'] = df['Has Table booking'].map({'Yes':1, 'No':0})  
df['Has_Online_Delivery'] = df['Has Online delivery'].map({'Yes':1, 'No':0})
```

```
In [5]: # Predictive Modeling  
features = ['Average Cost for two', 'Price range', 'Votes',  
           'Has_Table_Booking', 'Has_Online_Delivery', 'Name_Length']  
  
X = df[features]  
y = df['Aggregate rating']  
  
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)  
  
models = {  
    "Linear Regression": LinearRegression(),  
    "Decision Tree": DecisionTreeRegressor(),  
    "Random Forest": RandomForestRegressor()  
}  
  
for name, model in models.items():  
    model.fit(X_train, y_train)  
    predictions = model.predict(X_test)  
    print(f"\n{name} Results:")  
    print("MAE:", mean_absolute_error(y_test, predictions))  
    print("MSE:", mean_squared_error(y_test, predictions))  
    print("R2 Score:", r2_score(y_test, predictions))
```

Linear Regression Results:

MAE: 1.0737694001468152

MSE: 1.6736131796875675

R2 Score: 0.2647042884093149

Decision Tree Results:

MAE: 0.3118358625501482

MSE: 0.22809694022908308

R2 Score: 0.8997864596114667

Random Forest Results:

MAE: 0.2315463921754738

MSE: 0.1235066780161419

R2 Score: 0.9457378014225283

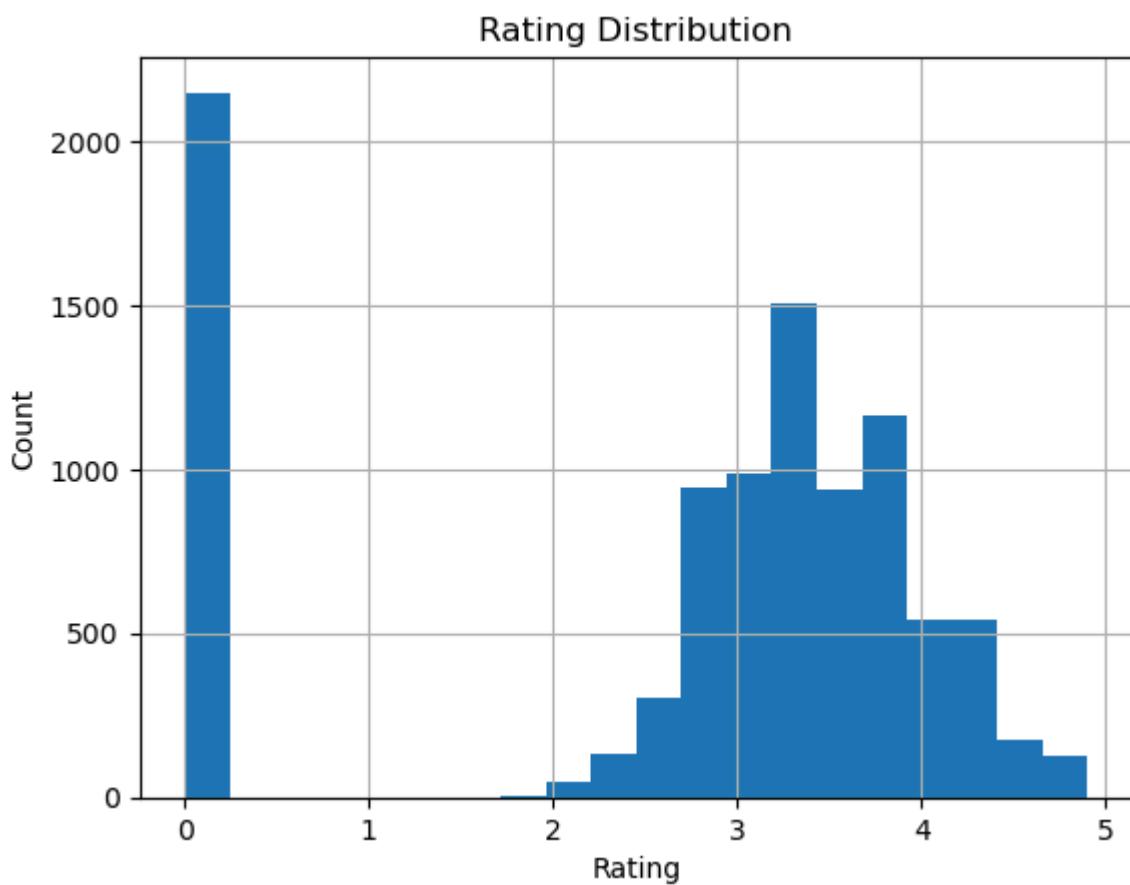
```
In [6]: # Customer Preference
print("\nTop Cuisines by Votes:\n", df.groupby('Cuisines')[ 'Votes' ].sum().sort_v
```

Top Cuisines by Votes:

Cuisines	Votes
North Indian, Mughlai	53747
North Indian	46241
North Indian, Chinese	42012
Cafe	30657
Chinese	21925
North Indian, Mughlai, Chinese	20115
Fast Food	17852
South Indian	16433
Mughlai, North Indian	15275
Italian	14799

Name: Votes, dtype: int64

```
In [7]: plt.figure()
df['Aggregate rating'].hist(bins=20)
plt.title("Rating Distribution")
plt.xlabel("Rating")
plt.ylabel("Count")
plt.show()
```



```
In [ ]:
```

