```python
import pandas as pd
df = pd.read_csv("Dataset .csv")
print(df['Cuisines'].value_counts())
top_cuisines = df['Cuisines'].value_counts().nlargest(10).index
df = df[df['Cuisines'].isin(top_cuisines)]
df = df.drop(['Restaurant Name', 'Address', 'Locality Verbose'], axis=1)
from sklearn.preprocessing import LabelEncoder
label = LabelEncoder()
df['Cuisines_encoded'] = label.fit_transform(df['Cuisines'])
for col in df.columns:
    if df[col].dtype == 'object':
        df[col] = label.fit_transform(df[col])
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
X = df.drop(['Cuisines', 'Cuisines_encoded'], axis=1)
y = df['Cuisines_encoded']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
clf = RandomForestClassifier()
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
```

```python
import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize=(10, 6))
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt="d", cmap="Blues")
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()
```

What can I help you build?

```
Cuisines
North Indian                        936
North Indian, Chinese               511
Chinese                             354
Fast Food                           354
North Indian, Mughlai               334
                                    ...
World Cuisine, Patisserie, Cafe       1
Burger, Izgara                        1
Desserts, B�_rek                      1
Restaurant Cafe, Turkish, Desserts    1
Restaurant Cafe, Desserts             1
Name: count, Length: 1825, dtype: int64
Accuracy: 0.3148936170212766
Classification Report:
              precision    recall  f1-score   support

           0       0.22      0.07      0.11        56
           1       0.27      0.07      0.12        41
           2       0.44      0.48      0.46        56
           3       0.26      0.11      0.16        80
           4       0.28      0.28      0.28        64
           5       0.37      0.51      0.43       186
           6       0.24      0.35      0.29        91
           7       0.24      0.32      0.28        60
           8       0.11      0.05      0.07        38
           9       0.39      0.39      0.39        33

    accuracy                           0.31       705
   macro avg       0.28      0.26      0.26       705
weighted avg       0.30      0.31      0.29       705
```

Confusion Matrix