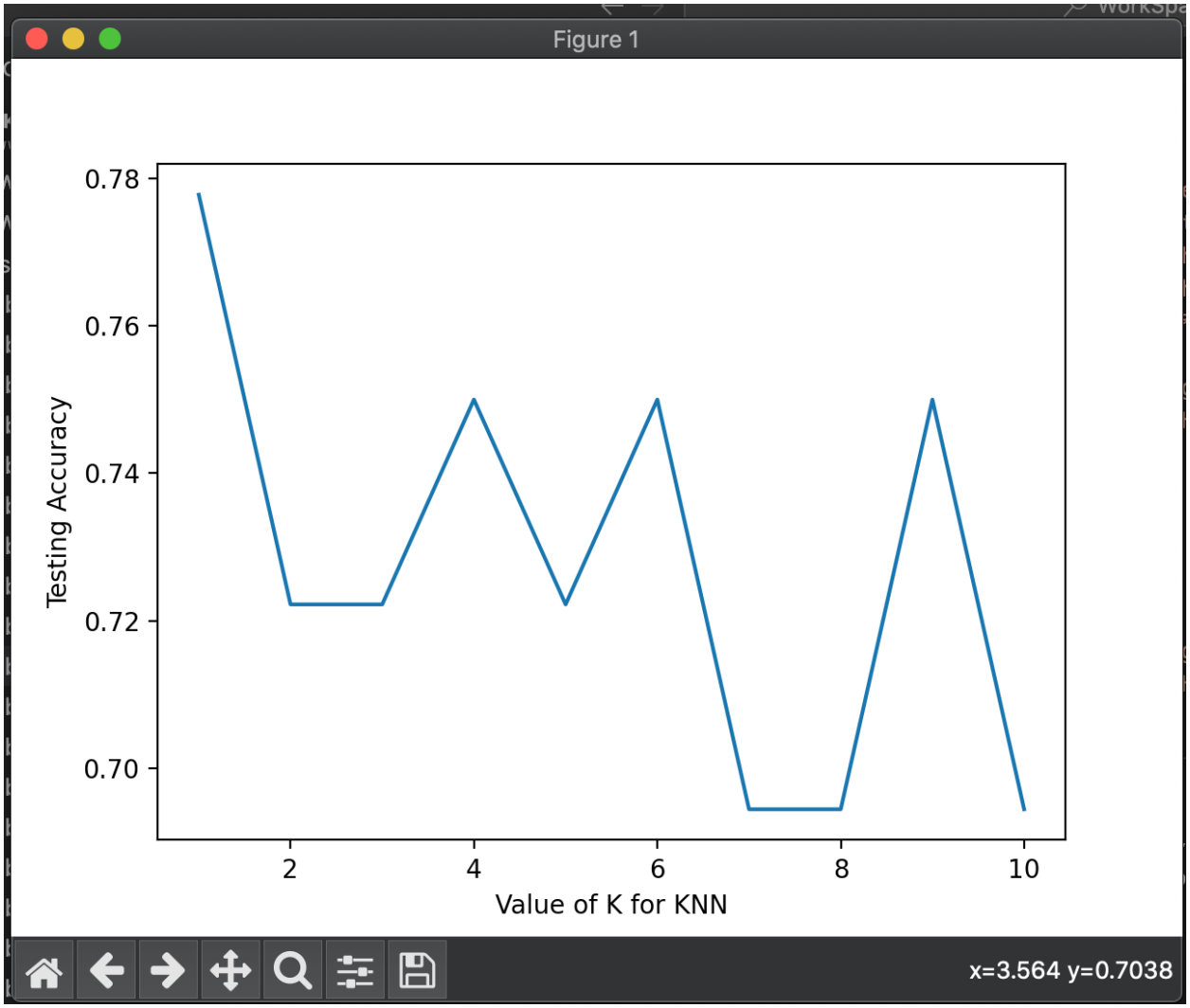
Raul Rodriguez

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.neighbors import KNeighborsClassifier
names = ['class', 'Alcohol','Malic Acid','Ash','Acadlinity','Magnisium','Total Phenols','Flavanoids',
'NonFlavanoid Phenols', 'Proanthocyanins', 'Color Intensity', 'Hue', 'OD280/OD315', 'Proline' ]
df = pd.read_csv('wine.data.csv', names=names)
#print(df)
X = np.array(df.loc[:, 'Alcohol' : 'Proline'])
y = np.array(df['class'])
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)
scores=[]
K_range=range(1,11)
print(K_range)
for K in K_range:
    knn=KNeighborsClassifier(n_neighbors=K)
    knn.fit(X_train,y_train)
    y_pred=knn.predict(X_test)
    scores.append(accuracy_score(y_test,y_pred))
plt.plot(K_range,scores)
plt.xlabel("Value of K for KNN")
plt.ylabel("Testing Accuracy")
plt.show()
```




```python
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.neighbors import KNeighborsClassifier
names = ['Sample code number', 'Clump Thickness','Uniformity of Cell Size','Uniformity of Cell Shape','Marginal Adhesion',
'Single Epithelial Cell Size','Bare Nuclei','Bland Chromatin','Normal Nucleoli','Mitoses','Class' ]
df = pd.read_csv('breast-cancer-wisconsin.data.csv', names=names)
df=df.replace('?',np.nan)
df=df.dropna()
df=df.drop("Sample code number",axis=1)
X = np.array(df.loc[:, 'Clump Thickness' : 'Mitoses'])
y = np.array(df['Class'])
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30)
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)
pred = knn.predict(X_test)
print('Model accuracy score: ', accuracy_score(y_test, pred))
print('Index\tPredicted\tActual')
for i in range(len(pred)):
    if pred[i] != y_test[i]:
        print(i, '\t', pred[i], '\t\t', y_test[i], '***')
print(f'\nConfusion Matrix: \n{confusion_matrix(y_test, pred)}')
#print(df)
```

(variable) pred: ndarray[Any, dtype[float64]]

```
raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython % /usr/local/
Model accuracy score:  0.9658536585365853
Index   Predicted       Actual
31       2               4 ***
96       2               4 ***
113      2               4 ***
142      2               4 ***
144      2               4 ***
145      2               4 ***
154      4               2 ***

Confusion Matrix:
[[127   1]
 [  6  71]]
raulrodriguez@Rauls-MacBook-Air WorkSpaceVSPython %
```