In [8]:
```
import sklearn.datasets

data = sklearn.datasets.load_wine()
data
```

Out[8]:
```
{'data': array([[1.423e+01, 1.710e+00, 2.430e+00, ..., 1.040e+00, 3.920e+00,
         1.065e+03],
        [1.320e+01, 1.780e+00, 2.140e+00, ..., 1.050e+00, 3.400e+00,
         1.050e+03],
        [1.316e+01, 2.360e+00, 2.670e+00, ..., 1.030e+00, 3.170e+00,
         1.185e+03],
        ...,
        [1.327e+01, 4.280e+00, 2.260e+00, ..., 5.900e-01, 1.560e+00,
         8.350e+02],
        [1.317e+01, 2.590e+00, 2.370e+00, ..., 6.000e-01, 1.620e+00,
         8.400e+02],
        [1.413e+01, 4.100e+00, 2.740e+00, ..., 6.100e-01, 1.600e+00,
         5.600e+02]]),
 'target': array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2,
        2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
        2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
        2, 2]),
 'target_names': array(['class_0', 'class_1', 'class_2'], dtype='<U7'),
 'DESCR': 'Wine Data Database\n====================\n\nNotes\n-----\nData Set Characteristics:\n    :Number of Instances: 178 (50 in each of three c
lasses)\n    :Number of Attributes: 13 numeric, predictive attributes and the class\n    :Attribute Information:\n \t\t- 1) Alcohol\n \t\t- 2) Malic
acid\n \t\t- 3) Ash\n\t\t- 4) Alcalinity of ash  \n \t\t- 5) Magnesium\n\t\t- 6) Total phenols\n \t\t- 7) Flavanoids\n \t\t- 8) Nonflavanoid phenols
\n \t\t- 9) Proanthocyanins\n\t\t- 10)Color intensity\n \t\t- 11)Hue\n \t\t- 12)OD280/OD315 of diluted wines\n \t\t- 13)Proline\n        \t- clas
s:\n                - class_0\n                - class_1\n                - class_2\n\t\t\n    :Summary Statistics:\n    \n    ====================
======== ==== ===== ======= =====\n                                   Min   Max   Mean     SD\n    ============================ ==== ===== =======
=====\n    Alcohol:                      11.0  14.8    13.0   0.8\n    Malic Acid:                   0.74  5.80    2.34  1.12\n    Ash:
             1.36  3.23    2.36  0.27\n    Alcalinity of Ash:            10.6  30.0    19.5   3.3\n    Magnesium:                   70.0 162.0    9
9.7  14.3\n    Total Phenols:                0.98  3.88    2.29  0.63\n    Flavanoids:                   0.34  5.08    2.03  1.00\n    Nonflavanoid
Phenols:         0.13  0.66    0.36  0.12\n    Proanthocyanins:              0.41  3.58    1.59  0.57\n    Colour Intensity:              1.3  13.0
    5.1   2.3\n    Hue:                          0.48  1.71    0.96  0.23\n    OD280/OD315 of diluted wines: 1.27  4.00    2.61  0.71\n    Proline:
                         278  1680     746   315\n    ============================ ==== ===== ======= =====\n\n    :Missing Attribute Values: None\n
  :Class Distribution: class_0 (59), class_1 (71), class_2 (48)\n    :Creator: R.A. Fisher\n    :Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.g
ov)\n    :Date: July, 1988\n\nThis is a copy of UCI ML Wine recognition datasets.\nhttps://archive.ics.uci.edu/ml/machine-learning-databases/wine/wi
ne.data\n\nThe data is the results of a chemical analysis of wines grown in the same\nregion in Italy by three different cultivators. There are thir
teen different\nmeasurements taken for different constituents found in the three types of\nwine.\n\nOriginal Owners: \n\nForina, M. et al, PARVUS -
\nAn Extendible Package for Data Exploration, Classification and Correlation. \nInstitute of Pharmaceutical and Food Analysis and Technologies,\nVia
Brigata Salerno, 16147 Genoa, Italy.\n\nCitation:\n\nLichman, M. (2013). UCI Machine Learning Repository\n[http://archive.ics.uci.edu/ml]. Irvine, C
A: University of California,\nSchool of Information and Computer Science. \n\nReferences\n----------\n(1) \nS. Aeberhard, D. Coomans and O. de Vel,
\nComparison of Classifiers in High Dimensional Settings, \nTech. Rep. no. 92-02, (1992), Dept. of Computer Science and Dept. of \nMathematics and S
tatistics, James Cook University of North Queensland. \n(Also submitted to Technometrics). \n\nThe data was used with many others for comparing vari
ous \nclassifiers. The classes are separable, though only RDA \nhas achieved 100% correct classification. \n(RDA : 100%, QDA 99.4%, LDA 98.9%, 1NN 9
6.1% (z-transformed data)) \n(All results using the leave-one-out technique) \n\n(2) \nS. Aeberhard, D. Coomans and O. de Vel, \n"THE CLASSIFICATION
PERFORMANCE OF RDA" \nTech. Rep. no. 92-01, (1992), Dept. of Computer Science and Dept. of \nMathematics and Statistics, James Cook University of No
rth Queensland. \n(Also submitted to Journal of Chemometrics). \n',
 'feature_names': ['alcohol',
  'malic_acid',
  'ash',
  'alcalinity_of_ash',
  'magnesium',
  'total_phenols',
  'flavanoids',
  'nonflavanoid_phenols',
  'proanthocyanins',
  'color_intensity',
  'hue',
  'od280/od315_of_diluted_wines',
  'proline']}
```

In [14]:
```
print(data.target)
print(data.target_names)
print(data.data.shape)
print(data.target.shape)
```

```
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2]
['class_0' 'class_1' 'class_2']
(178, 13)
(178,)
```

In [17]:
```
#Import scikit libraries
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics
```

In [19]:
```
X = data.data
y = data.target
```

In [23]:

```
#Split into train and test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state =4) # 70% training and 30% test

# Create a k-NN classifier with 5 neighbors
knn = KNeighborsClassifier(n_neighbors=8)

# Fit the classifier to the training data
knn.fit(X, y)

#Predict the response for test dataset
y_pred = knn.predict(X_test)

print(y_pred)
```

```
[2 2 0 0 2 0 0 0 0 1 1 0 2 2 0 1 0 1 1 1 1 2 1 2 0 2 1 1 1 2 0 2 0 1 1 2 0
 2 0 1 1 0 2 1 1 0 1 2 0 2 1 1 1 0 0 1 1 1 1 1 0 2 1 1 1 0 2 1 2 0 2 2]
```

In [24]:

```
print (metrics.accuracy_score(y_test, y_pred))
```

```
0.7777777777777778
```

In [21]:

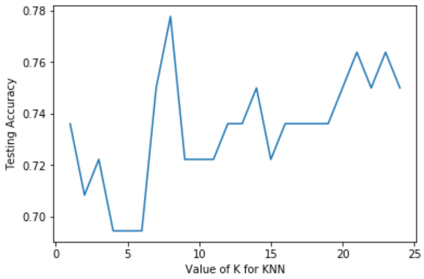
```
#Finding a the best value for K
#For to to calculate K from 1 through 25 and record testing accuracy

k_range = range(1,25)
scores = []
for k in k_range:
    knn = KNeighborsClassifier(n_neighbors = k)
    knn.fit(X_train, y_train)
    y_pred = knn.predict(X_test)
    scores.append(metrics.accuracy_score(y_test,y_pred))
```

In [22]:

```
# Plot relationship between K and testing accuracy

import matplotlib.pyplot as plt

%matplotlib inline

plt.plot(k_range, scores)
plt.xlabel('Value of K for KNN')
plt.ylabel('Testing Accuracy')
```

Out[22]:

```
Text(0,0.5,'Testing Accuracy')
```



In [ ]: