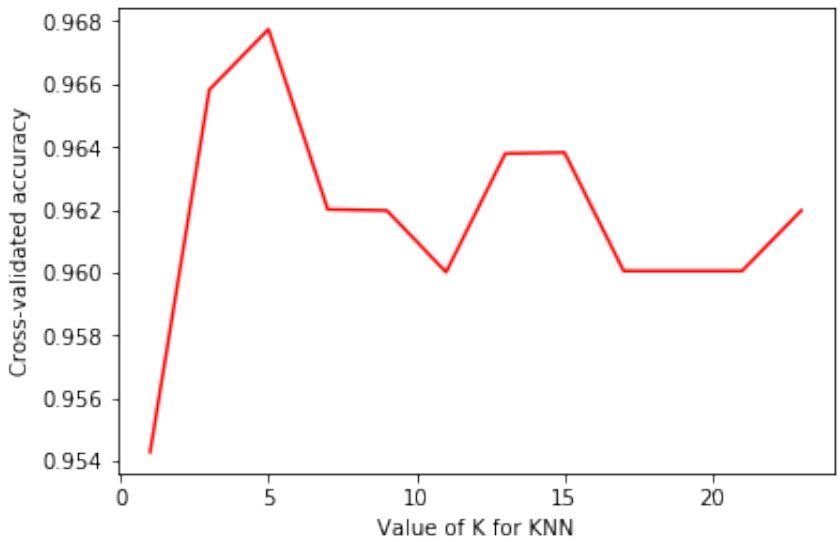
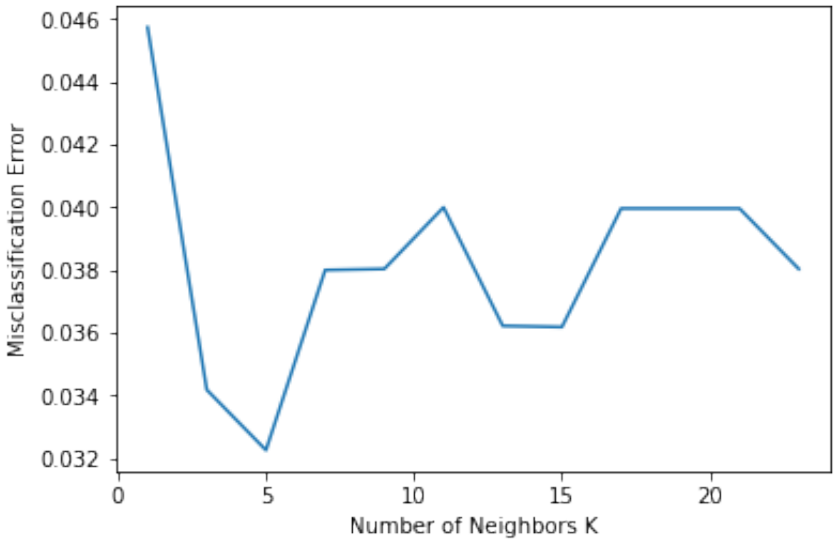
In [2]:
```
# ML Assignment 1 K-Nearest Neighbors (K-NN)

# Importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import os

path="/Users/jayendra/Desktop/python"
os.chdir(path)
data=pd.read_csv('breastCancer.csv')
data.replace('?', 0, inplace=True)
data = data.applymap(np.int64)
x = data.iloc[:, 1:-1].values
y = data.iloc[:, -1].values
# Splitting the dataset into the Training set and Test set

from sklearn.cross_validation import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size
= 0.25, random_state = 0)

# Applying Knn Classifier
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors = 5, metric = 'minkow
ski', p = 2)
classifier.fit(x_train, y_train)

#Predicting the values from the classifier
y_pred = classifier.predict(x_test)

#Finding
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print(cm)
result=  (cm[0][0]+ cm[1][1])/(cm[0][0]+cm[1][1]+ cm[0][1]+cm[1][0]
)*100
print ("The accuracy of the model %f" % result)
# creating odd list of K for KNN
kList = list(range(1,25))
# subsetting just the odd ones
kodd = list(filter(lambda x: x % 2 != 0, kList))
# empty list that will hold cv scores
cv_scores = []

# perform 10-fold cross validation
from sklearn.model_selection import cross_val_score
for k in kodd:
    knn = KNeighborsClassifier(n_neighbors=k)
    scores = cross_val_score(knn, x_train, y_train, cv=10, scoring=
'accuracy')
    cv_scores.append(scores.mean())
```

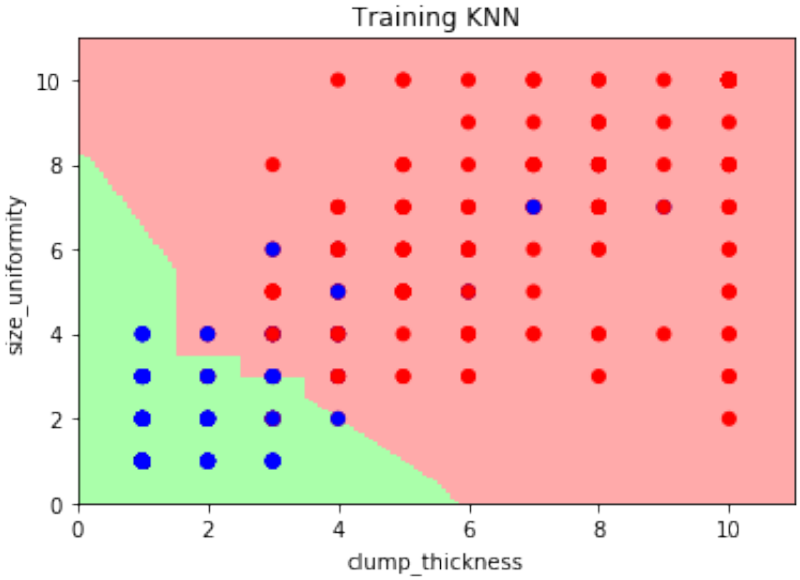
```
print('Length of list', len(cv_scores))
print('Max of list', max(cv_scores))
MSE = [1 - x for x in cv_scores]

# determining best k
optimal_k = kodd[MSE.index(min(MSE))]
print ("The optimal number of neighbors is %d" % optimal_k)
plt.plot(kodd, cv_scores,color="red")
plt.xlabel('Value of K for KNN')
plt.ylabel('Cross-validated accuracy')
plt.show()


# plot misclassification error vs k
plt.plot(kodd, MSE)
plt.xlabel('Number of Neighbors K')
plt.ylabel('Misclassification Error')
plt.show()

#training Plot
kn=KNeighborsClassifier(n_neighbors=5)
kn.fit(x_train[:,1:3], y_train)
from matplotlib.colors import ListedColormap
cmap_light = ListedColormap(['#AAFFAA','#FFAAAA'])
cmap_bold = ListedColormap(['#0000FF','#FF0000'])
# creating a meshgrid
x_min, x_max = x[:, 1].min() - 1, x[:, 1].max() + 1
y_min, y_max = x[:, 2].min() - 1, x[:, 2].max() + 1
h=0.07
xx, yy = np.meshgrid(np.arange(x_min, x_max, h),np.arange(y_min, y_
max, h))
xy_mesh=np.c_[xx.ravel(), yy.ravel()]
Z = kn.predict(xy_mesh)
Z = Z.reshape(xx.shape)
plt.figure()
plt.pcolormesh(xx, yy, Z, cmap=cmap_light)
ax=plt.scatter(x[:, 1], x[:, 2], c=y, cmap=cmap_bold)
plt.xlim(xx.min(), xx.max())
plt.ylim(yy.min(), yy.max())
plt.xlabel('clump_thickness')
plt.ylabel('size_uniformity')
plt.title('Training KNN')
plt.show()

# Testing Plot
kn=KNeighborsClassifier(n_neighbors=5)
kn.fit(x_test, y_test)
# for display purposes, we fit the model on the first two component
s i.e. PC1, and PC2
kn.fit(x_test[:,1:3], y_test)
from matplotlib.colors import ListedColormap
cmap_light = ListedColormap(['#AAFFAA','#FFAAAA'])
cmap_bold = ListedColormap(['#0000FF','#FF0000'])
```

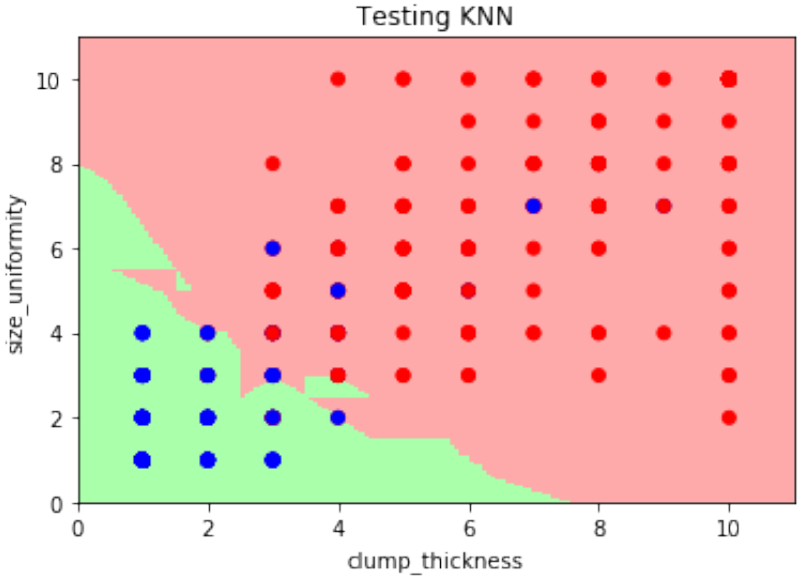
```
# creating a meshgrid
x_min, x_max = x[:, 1].min() - 1, x[:, 1].max() + 1
y_min, y_max = x[:, 2].min() - 1, x[:, 2].max() + 1
h=0.07
xx, yy = np.meshgrid(np.arange(x_min, x_max, h),np.arange(y_min, y_
max, h))
xy_mesh=np.c_[xx.ravel(), yy.ravel()]
Z = kn.predict(xy_mesh)
Z = Z.reshape(xx.shape)
plt.figure()
plt.pcolormesh(xx, yy, Z, cmap=cmap_light)
ax=plt.scatter(x[:, 1], x[:, 2], c=y, cmap=cmap_bold)
plt.xlim(xx.min(), xx.max())
plt.ylim(yy.min(), yy.max())
plt.xlabel('clump_thickness')
plt.ylabel('size_uniformity')
plt.title('Testing KNN')
plt.show()
```

```
/anaconda3/lib/python3.6/site-packages/sklearn/cross_validation.py
:41: DeprecationWarning: This module was deprecated in version 0.1
8 in favor of the model_selection module into which all the refact
ored classes and functions are moved. Also note that the interface
of the new CV iterators are different from that of this module. Th
is module will be removed in 0.20.
  "This module will be removed in 0.20.", DeprecationWarning)

[[110   2]
 [  2  61]]
The accuracy of the model 97.714286
Length of list 12
Max of list 0.96774168303
The optimal number of neighbors is 5
```




Misclassification Error
Number of Neighbors K
0.046
0.044
0.042
0.040
0.038
0.036
0.034
0.032
0
5
10
15
20


Training KNN
size_uniformity
dump_thickness


Testing KNN
size_uniformity
dump_thickness

In [ ]:
```
#Conclusion:
#On applying k Nearest Neighbors on the dataset Breast Cancer, the
prediction accuracy of the model comes out be 97.71% using confusio
n matrix.
#After that for getting the Optimal number of neighbor ,I have take
n the set of the odd values of k and applied 10 fold CV. The optima
l number of neighbors comes out to be 5 with the prediction accurac
y of 96.77% and same can be infer from the Cross Validation accurac
y Graph.
#We can infer the same from the misclassification error Graph, the
error is least at neighbor count 5.
#By Ploting the Visual Graph between "clump_thickness Vs size_unifo
rmity" for Training and Testing data explains the decision boundary
using KNN classifier which is non linear and the points predicted b
y the classifier scattered along the boundary and 2D space.
```