# KNN Manual Implementation

Mohammad Shakirul Islam | 151-15-311 | 23 feb 2018

In [80]:
```
# import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from scipy.spatial import distance
from sklearn.metrics import accuracy_score
from sklearn.neighbors import KNeighborsClassifier
```

In [81]:
```
# read data file as dataframe
data = pd.read_csv('iris.csv')
```

In [82]:
```
#take four numeric features as X input
X = data.values[:, :4]
```

In [83]:
```
#create an array of length 150 named y
y = np.zeros(150)
```

In [84]:
```
#encoding classes to numbers
for i in range(len(y)):
    if data.values[i, 4]=='setosa':
        y[i] = 0
    elif data.values[i, 4]=='versicolor':
        y[i] = 1
    elif data.values[i, 4]=='virginica':
        y[i] = 2
```

In [85]:
```
#randomly shuffle the whole dataset and create train-test partition
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_
```

In [86]:
```
# Distance matrix
d=distance.cdist(X_test,X_train,'euclidean')
print(d.shape)
```

```
(50, 100)
```

In [87]:
```
for i in range(50):
    y1=np.argsort(d,axis=1)

#print(y)
k=int(3)
y_pred=[]
j=int(0)

for j in range(0,50):
    y3=np.zeros(3)
    for i in range(0,3):
        ind=int(y1[j][i])
        val=int(y_train[ind])
        y3[val]+=1

    y_pred.append(np.argmax(y3, axis=0))

accuracy_score(y_test, y_pred)
```

Out[87]: 0.97999999999999998

## Plot K VS Accuracy graph (Line Plot) for K = 1, 3, 5, 7, 9, 11, 13, 15

In [88]:
```
k3=np.array([1, 3, 5, 7, 9, 11, 13, 15])
r=int(0)
rs=[]
for k2 in k3:
    y_pred=[]
    for j in range(0,50):
        y3=np.zeros(3)
        for i in range (0,k2):
            ind1=int(yy[j][i])
            val=int(y_train[ind1])
            y3[val]+=1

        y_pred.append(np.argmax(y3,axis=0))
    rs.append(accuracy_score(y_test,y_pred))

rst=np.array(rs)
plt.plot(rst,k3)
plt.show()
```