In [58]:
```
from scipy.io import arff
import pandas as pd
import math
import matplotlib.pyplot as plt
import operator
```

In [59]:
```
col=["c_type", "lifestyle", "vacation", "credit", "salary", "property_value", "cl
```

In [60]:
```
train_data=pd.read_csv("trainProdSelection.arff", delimiter="\t")
test_data=pd.read_csv("testProdSelection.arff",delimiter="\t")
```

In [61]:
```
train_data.columns=col
```

In [62]:
```
train_data.head()
train_data.shape
```

Out[62]:
```
(185, 7)
```

In [63]:
```
test_data.columns=col
test_data.head()
```

Out[63]:

	c_type	lifestyle	vacation	credit	salary	property_value	class
0	student	spend>>saving	29	10	16.1900	2.4839	C1
1	student	spend<<saving	28	60	15.4600	1.1885	C1
2	engineer	spend>saving	15	41	21.2600	1.4379	C1
3	librarian	spend<saving	2	9	19.7207	0.6913	C1
4	librarian	spend>saving	7	9	12.7098	1.4728	C1

In [64]:
```
train_data.describe()
```

Out[64]:

	vacation	credit	salary	property_value
count	185.000000	185.000000	185.000000	185.000000
mean	27.691892	62.783784	20.702852	4.146650
std	18.572630	69.120537	4.244655	3.775707
min	1.000000	3.000000	8.507600	0.008000
25%	9.000000	15.000000	18.594400	1.644700
50%	26.000000	45.000000	20.390000	2.897200
75%	48.000000	72.000000	22.790000	4.838800
max	64.000000	347.000000	31.750000	17.873700

In [67]:
```
```

In [68]:
```
```

In [69]:
```
test_data.head()
```

Out[69]:

	c_type	lifestyle	vacation	credit	salary	property_value	class
0	student	spend>>saving	29	10	16.1900	2.4839	C1
1	student	spend<<saving	28	60	15.4600	1.1885	C1
2	engineer	spend>saving	15	41	21.2600	1.4379	C1
3	librarian	spend<saving	2	9	19.7207	0.6913	C1
4	librarian	spend>saving	7	9	12.7098	1.4728	C1

In [70]:
```
dtrain_numeric=train_data[['vacation','credit','salary','property_value']]
dtrain_numeric.head()
```

Out[70]:

	vacation	credit	salary	property_value
0	11	21	15.32	2.0232
1	7	64	16.55	3.1202
2	3	47	15.71	3.4022
3	15	10	16.96	2.2825
4	6	80	15.50	3.7338

In [71]:
```
dtest_numeric=test_data[['vacation','credit','salary','property_value']]
dtest_numeric.head()
```

Out[71]:

	vacation	credit	salary	property_value
0	29	10	16.1900	2.4839
1	28	60	15.4600	1.1885
2	15	41	21.2600	1.4379
3	2	9	19.7207	0.6913
4	7	9	12.7098	1.4728

In [72]:
```
dtrain_norm = (dtrain_numeric-dtrain_numeric.min())/(dtrain_numeric.max()-dtrain_
dtrain_norm.head()
```

Out[72]:

	vacation	credit	salary	property_value
0	0.158730	0.052326	0.293102	0.112797
1	0.095238	0.177326	0.346023	0.174200
2	0.031746	0.127907	0.309882	0.189984
3	0.222222	0.020349	0.363663	0.127311
4	0.079365	0.223837	0.300847	0.208545

In [73]:
```
dtest_norm = (dtest_numeric-dtest_numeric.min())/(dtest_numeric.max()-dtest_numer
dtest_norm.head()
```

Out[73]:

	vacation	credit	salary	property_value
0	0.54	0.021008	0.175059	0.243041
1	0.52	0.231092	0.138339	0.085992
2	0.26	0.151261	0.430086	0.116229
3	0.00	0.016807	0.352657	0.025714
4	0.10	0.016807	0.000000	0.120460

In [74]:
```
train_data.head()
```

Out[74]:

	c_type	lifestyle	vacation	credit	salary	property_value	class
0	student	spend>saving	11	21	15.32	2.0232	C1
1	student	spend>saving	7	64	16.55	3.1202	C1
2	student	spend>saving	3	47	15.71	3.4022	C1
3	student	spend>saving	15	10	16.96	2.2825	C1
4	student	spend>saving	6	80	15.50	3.7338	C1

In [75]:
```
train_data=train_data.drop(['vacation','credit','salary','property_value'],axis=1
dtrain=pd.concat([train_data,dtrain_norm],axis=1)
dtrain.head()
```

Out[75]:

	c_type	lifestyle	class	vacation	credit	salary	property_value
0	student	spend>saving	C1	0.158730	0.052326	0.293102	0.112797
1	student	spend>saving	C1	0.095238	0.177326	0.346023	0.174200
2	student	spend>saving	C1	0.031746	0.127907	0.309882	0.189984
3	student	spend>saving	C1	0.222222	0.020349	0.363663	0.127311
4	student	spend>saving	C1	0.079365	0.223837	0.300847	0.208545

In [76]:
```
# dtrain=dtrain.drop(['lifestyle','c_type'],axis=1)
dtrain['cclass']=dtrain['class']
dtrain.head()
```

Out[76]:

	c_type	lifestyle	class	vacation	credit	salary	property_value	cclass
0	student	spend>saving	C1	0.158730	0.052326	0.293102	0.112797	C1
1	student	spend>saving	C1	0.095238	0.177326	0.346023	0.174200	C1
2	student	spend>saving	C1	0.031746	0.127907	0.309882	0.189984	C1
3	student	spend>saving	C1	0.222222	0.020349	0.363663	0.127311	C1
4	student	spend>saving	C1	0.079365	0.223837	0.300847	0.208545	C1

In [77]:
```
dtrain=dtrain.drop('class',axis=1)
dtrain.head()
```

Out[77]:

	c_type	lifestyle	vacation	credit	salary	property_value	cclass
0	student	spend>saving	0.158730	0.052326	0.293102	0.112797	C1
1	student	spend>saving	0.095238	0.177326	0.346023	0.174200	C1
2	student	spend>saving	0.031746	0.127907	0.309882	0.189984	C1
3	student	spend>saving	0.222222	0.020349	0.363663	0.127311	C1
4	student	spend>saving	0.079365	0.223837	0.300847	0.208545	C1

In [78]:
```
# dtest=pd.concat([test_data,tectype,telife],axis=1)
# dtest=dtest.drop(['lifestyle','c_type'],axis=1)
# dtest['cclass']=dtest['class']
# dtest.head()
test_data=test_data.drop(['vacation','credit','salary','property_value'],axis=1)
dtest=pd.concat([test_data,dtest_norm],axis=1)
```

In [79]:
```
dtest['cclass']=dtest['class']
dtest=dtest.drop("class",axis=1)
dtest.head()
```

Out[79]:

	c_type	lifestyle	vacation	credit	salary	property_value	cclass
0	student	spend>>saving	0.54	0.021008	0.175059	0.243041	C1
1	student	spend<<saving	0.52	0.231092	0.138339	0.085992	C1
2	engineer	spend>saving	0.26	0.151261	0.430086	0.116229	C1
3	librarian	spend<saving	0.00	0.016807	0.352657	0.025714	C1
4	librarian	spend>saving	0.10	0.016807	0.000000	0.120460	C1

FROM SCRATCH

In [80]:
```
k_list=[]
acc_list=[]
for k in range(1,26,2):
    k_list.append(k)
    predict=[]
    def euc_distance(testrow,trainrow,length):
        distance=0
        for i in range(1,3):
            if(testrow[i]==trainrow[i]):
                distance+=1
#         for i in range(2):
#             if(testrow[i]==trainrow[i]):
#                 distance+=1
        for i in range(3,length-1):
            distance+=pow((testrow[i]-trainrow[i]),2)
        return math.sqrt(distance)
    def getNeighbours(traindata,testRow,k):
        distance_with_train=[]
        length=len(testRow)
        for x in range(len(traindata)):
            dist=euc_distance(testRow,traindata[x],length)
            distance_with_train.append((traindata[x],dist))
        distance_with_train.sort(key=operator.itemgetter(1))
        neighbors = []
        for x in range(k):
            neighbors.append(distance_with_train[x][0])
        return neighbors
    def getResponse(neighbors):
        votes = {}
        for x in range(len(neighbors)):
            response = neighbors[x][-1]
            if response in votes:
                votes[response] += 1
            else:
                votes[response] = 1
        sortedVotes = sorted(votes.items(), key=operator.itemgetter(1), reverse=T
        return sortedVotes[0][0]
    def getAccuracy(dtest, predict):
        correct = 0
        for x in range(len(dtest)):
            if dtest[x][-1] == predict[x]:
                correct += 1
        return (correct/float(len(dtest))) * 100.0
    for i in range(len(dtest)):
        neighbour=getNeighbours(dtrain.values,dtest.values[i],k)
    #     print(neighbour)
        result = getResponse(neighbour)
        predict.append(result)
#         print('> predicted=' + repr(result) + ', actual=' + repr(xtest.values[i
    accuracy = getAccuracy(dtest.values, predict)
    acc_list.append(accuracy)
    print('Accuracy: ' + repr(accuracy) + '%','with k=',k)
```

```
Accuracy: 40.0% with k= 1
Accuracy: 35.0% with k= 3
Accuracy: 35.0% with k= 5
```

```
Accuracy: 35.0% with k= 7
Accuracy: 35.0% with k= 9
Accuracy: 35.0% with k= 11
Accuracy: 30.0% with k= 13
Accuracy: 30.0% with k= 15
Accuracy: 30.0% with k= 17
Accuracy: 25.0% with k= 19
Accuracy: 25.0% with k= 21
Accuracy: 30.0% with k= 23
Accuracy: 20.0% with k= 25
```

Using KNN Classifier

In [149]:
```
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
```

In [158]:
```
x_train= dtrain.loc[:,'c_type':'property_value']
y_train= dtrain.loc[:,['cclass']]
```

In [159]:
```
x_train.head()
```

Out[159]:

	c_type	lifestyle	vacation	credit	salary	property_value
0	student	spend>saving	0.158730	0.052326	0.293102	0.112797
1	student	spend>saving	0.095238	0.177326	0.346023	0.174200
2	student	spend>saving	0.031746	0.127907	0.309882	0.189984
3	student	spend>saving	0.222222	0.020349	0.363663	0.127311
4	student	spend>saving	0.079365	0.223837	0.300847	0.208545

In [160]:
```
x_train["type"] = lb_make.fit_transform(x_train["c_type"])
x_train["type"].value_counts()
```

Out[160]:
```
1    49
3    39
4    37
0    37
2    23
Name: type, dtype: int64
```

In [161]:
```
x_train["style"] = lb_make.fit_transform(x_train["lifestyle"])
x_train["style"].value_counts()
```

Out[161]:
```
3    86
1    41
2    38
0    20
Name: style, dtype: int64
```

In [162]:
```
x_train.head()
```

Out[162]:

	c_type	lifestyle	vacation	credit	salary	property_value	type	style
0	student	spend>saving	0.158730	0.052326	0.293102	0.112797	4	3
1	student	spend>saving	0.095238	0.177326	0.346023	0.174200	4	3
2	student	spend>saving	0.031746	0.127907	0.309882	0.189984	4	3
3	student	spend>saving	0.222222	0.020349	0.363663	0.127311	4	3
4	student	spend>saving	0.079365	0.223837	0.300847	0.208545	4	3

In [166]:
```
x_train=x_train[["type","style","vacation","credit","salary","property_value"]]
x_train.head()
```

Out[166]:

	type	style	vacation	credit	salary	property_value
0	4	3	0.158730	0.052326	0.293102	0.112797
1	4	3	0.095238	0.177326	0.346023	0.174200
2	4	3	0.031746	0.127907	0.309882	0.189984
3	4	3	0.222222	0.020349	0.363663	0.127311
4	4	3	0.079365	0.223837	0.300847	0.208545

In [171]:
```
x_train["type"] = x_train['type'].astype(float)
x_train.head()
```

Out[171]:

	type	style	vacation	credit	salary	property_value
0	4.0	3	0.158730	0.052326	0.293102	0.112797
1	4.0	3	0.095238	0.177326	0.346023	0.174200
2	4.0	3	0.031746	0.127907	0.309882	0.189984
3	4.0	3	0.222222	0.020349	0.363663	0.127311
4	4.0	3	0.079365	0.223837	0.300847	0.208545

In [173]:
```
x_train["style"] = x_train['style'].astype(float)
x_train.head()
```

Out[173]:

	type	style	vacation	credit	salary	property_value
0	4.0	3.0	0.158730	0.052326	0.293102	0.112797
1	4.0	3.0	0.095238	0.177326	0.346023	0.174200
2	4.0	3.0	0.031746	0.127907	0.309882	0.189984
3	4.0	3.0	0.222222	0.020349	0.363663	0.127311
4	4.0	3.0	0.079365	0.223837	0.300847	0.208545

In [174]:
```
y_train.head()
```

Out[174]:

	cclass
0	C1
1	C1
2	C1
3	C1
4	C1

In [175]:
```
x_test=dtest.loc[:,'c_type':'property_value']
y_test=dtest.loc[:,['cclass']]
```

In [176]:
```
x_test["type"] = lb_make.fit_transform(x_test["c_type"])
x_test["type"].value_counts()
```

Out[176]:
```
4    6
3    5
2    3
1    3
0    3
Name: type, dtype: int64
```

In [177]:
```
x_test["style"] = lb_make.fit_transform(x_test["lifestyle"])
x_test["style"].value_counts()
```

Out[177]:
```
3    8
2    7
1    4
0    1
Name: style, dtype: int64
```

In [181]:
```
x_test=x_test[["type","style","vacation","credit","salary","property_value"]]
x_test.head()
```

Out[181]:

	type	style	vacation	credit	salary	property_value
0	4	2	0.54	0.021008	0.175059	0.243041
1	4	0	0.52	0.231092	0.138339	0.085992
2	1	3	0.26	0.151261	0.430086	0.116229
3	2	1	0.00	0.016807	0.352657	0.025714
4	2	3	0.10	0.016807	0.000000	0.120460

In [182]:
```
x_test["type"] = x_test['type'].astype(float)
x_test.head()
```

Out[182]:

	type	style	vacation	credit	salary	property_value
0	4.0	2	0.54	0.021008	0.175059	0.243041
1	4.0	0	0.52	0.231092	0.138339	0.085992
2	1.0	3	0.26	0.151261	0.430086	0.116229
3	2.0	1	0.00	0.016807	0.352657	0.025714
4	2.0	3	0.10	0.016807	0.000000	0.120460

In [184]:
```
x_test["style"] = x_test['style'].astype(float)
x_test.head()
```

Out[184]:

	type	style	vacation	credit	salary	property_value
0	4.0	2.0	0.54	0.021008	0.175059	0.243041
1	4.0	0.0	0.52	0.231092	0.138339	0.085992
2	1.0	3.0	0.26	0.151261	0.430086	0.116229
3	2.0	1.0	0.00	0.016807	0.352657	0.025714
4	2.0	3.0	0.10	0.016807	0.000000	0.120460

In [185]:
```
for K in range(20):
 K_value = K+1
 neigh = KNeighborsClassifier(n_neighbors = K_value, weights='uniform', algorithm
 neigh.fit(x_train, y_train)
 y_pred = neigh.predict(x_test)
 print("Accuracy is ", accuracy_score(y_test,y_pred)*100,"% for K-Value:",K_value
```

```
Accuracy is  20.0 % for K-Value: 1
Accuracy is  25.0 % for K-Value: 2
Accuracy is  25.0 % for K-Value: 3
Accuracy is  25.0 % for K-Value: 4
Accuracy is  25.0 % for K-Value: 5
Accuracy is  25.0 % for K-Value: 6
Accuracy is  25.0 % for K-Value: 7
Accuracy is  25.0 % for K-Value: 8
Accuracy is  25.0 % for K-Value: 9
Accuracy is  25.0 % for K-Value: 10
Accuracy is  25.0 % for K-Value: 11
Accuracy is  15.0 % for K-Value: 12
Accuracy is  15.0 % for K-Value: 13
Accuracy is  15.0 % for K-Value: 14
Accuracy is  10.0 % for K-Value: 15
Accuracy is  15.0 % for K-Value: 16
Accuracy is  10.0 % for K-Value: 17
Accuracy is  10.0 % for K-Value: 18
Accuracy is  5.0 % for K-Value: 19
Accuracy is  10.0 % for K-Value: 20

G:\Anaconda\lib\site-packages\ipykernel_launcher.py:4: DataConversionWarning: A
column-vector y was passed when a 1d array was expected. Please change the shap
e of y to (n_samples, ), for example using ravel().
  after removing the cwd from sys.path.
G:\Anaconda\lib\site-packages\ipykernel_launcher.py:4: DataConversionWarning: A
column-vector y was passed when a 1d array was expected. Please change the shap
e of y to (n_samples, ), for example using ravel().
  after removing the cwd from sys.path.
G:\Anaconda\lib\site-packages\ipykernel_launcher.py:4: DataConversionWarning: A
column-vector y was passed when a 1d array was expected. Please change the shap
e of y to (n_samples, ), for example using ravel().
  after removing the cwd from sys.path.
G:\Anaconda\lib\site-packages\ipykernel_launcher.py:4: DataConversionWarning: A
column-vector y was passed when a 1d array was expected. Please change the shap
e of y to (n_samples, ), for example using ravel().
  after removing the cwd from sys.path.
G:\Anaconda\lib\site-packages\ipykernel_launcher.py:4: DataConversionWarning: A
column-vector y was passed when a 1d array was expected. Please change the shap
e of y to (n_samples, ), for example using ravel().
  after removing the cwd from sys.path.
G:\Anaconda\lib\site-packages\ipykernel_launcher.py:4: DataConversionWarning: A
column-vector y was passed when a 1d array was expected. Please change the shap
e of y to (n_samples, ), for example using ravel().
  after removing the cwd from sys.path.
G:\Anaconda\lib\site-packages\ipykernel_launcher.py:4: DataConversionWarning: A
column-vector y was passed when a 1d array was expected. Please change the shap
e of y to (n_samples, ), for example using ravel().
  after removing the cwd from sys.path.
G:\Anaconda\lib\site-packages\ipykernel_launcher.py:4: DataConversionWarning: A
```

```
column-vector y was passed when a 1d array was expected. Please change the shap
e of y to (n_samples, ), for example using ravel().
  after removing the cwd from sys.path.
G:\Anaconda\lib\site-packages\ipykernel_launcher.py:4: DataConversionWarning: A
column-vector y was passed when a 1d array was expected. Please change the shap
e of y to (n_samples, ), for example using ravel().
  after removing the cwd from sys.path.
G:\Anaconda\lib\site-packages\ipykernel_launcher.py:4: DataConversionWarning: A
column-vector y was passed when a 1d array was expected. Please change the shap
e of y to (n_samples, ), for example using ravel().
  after removing the cwd from sys.path.
G:\Anaconda\lib\site-packages\ipykernel_launcher.py:4: DataConversionWarning: A
column-vector y was passed when a 1d array was expected. Please change the shap
e of y to (n_samples, ), for example using ravel().
  after removing the cwd from sys.path.
G:\Anaconda\lib\site-packages\ipykernel_launcher.py:4: DataConversionWarning: A
column-vector y was passed when a 1d array was expected. Please change the shap
e of y to (n_samples, ), for example using ravel().
  after removing the cwd from sys.path.
G:\Anaconda\lib\site-packages\ipykernel_launcher.py:4: DataConversionWarning: A
column-vector y was passed when a 1d array was expected. Please change the shap
e of y to (n_samples, ), for example using ravel().
  after removing the cwd from sys.path.
G:\Anaconda\lib\site-packages\ipykernel_launcher.py:4: DataConversionWarning: A
column-vector y was passed when a 1d array was expected. Please change the shap
e of y to (n_samples, ), for example using ravel().
  after removing the cwd from sys.path.
G:\Anaconda\lib\site-packages\ipykernel_launcher.py:4: DataConversionWarning: A
column-vector y was passed when a 1d array was expected. Please change the shap
e of y to (n_samples, ), for example using ravel().
  after removing the cwd from sys.path.
G:\Anaconda\lib\site-packages\ipykernel_launcher.py:4: DataConversionWarning: A
column-vector y was passed when a 1d array was expected. Please change the shap
e of y to (n_samples, ), for example using ravel().
  after removing the cwd from sys.path.
G:\Anaconda\lib\site-packages\ipykernel_launcher.py:4: DataConversionWarning: A
column-vector y was passed when a 1d array was expected. Please change the shap
e of y to (n_samples, ), for example using ravel().
  after removing the cwd from sys.path.
G:\Anaconda\lib\site-packages\ipykernel_launcher.py:4: DataConversionWarning: A
column-vector y was passed when a 1d array was expected. Please change the shap
e of y to (n_samples, ), for example using ravel().
  after removing the cwd from sys.path.
G:\Anaconda\lib\site-packages\ipykernel_launcher.py:4: DataConversionWarning: A
column-vector y was passed when a 1d array was expected. Please change the shap
e of y to (n_samples, ), for example using ravel().
  after removing the cwd from sys.path.
G:\Anaconda\lib\site-packages\ipykernel_launcher.py:4: DataConversionWarning: A
column-vector y was passed when a 1d array was expected. Please change the shap
e of y to (n_samples, ), for example using ravel().
  after removing the cwd from sys.path.
```