

```
In [1]: from scipy.io import arff
import pandas as pd
import math
import matplotlib.pyplot as plt
import operator
```

```
In [2]: col=["c_type", "lifestyle", "vacation", "credit", "salary", "property_value", "cl
```

```
In [3]: train_data=pd.read_csv("trainProdSelection.arff", delimiter="\t")
test_data=pd.read_csv("testProdSelection.arff", delimiter="\t")
```

```
In [4]: train_data.columns=col
```

```
In [5]: train_data.head()
train_data.shape
```

Out[5]: (185, 7)

```
In [6]: test_data.columns=col
test_data.head()
```

Out[6]:

	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 [7]: train_data.describe()
```

Out[7]:

	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 [8]: test_data.head()
```

```
Out[8]:
```

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

```
Out[9]:
```

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

```
Out[10]:
```

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

```
Out[11]:
```

	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 [12]: dtest_norm = (dtest_numeric-dtrain_numeric.min())/(dtrain_numeric.max()-dtrain_numeric.min())
dtest_norm.head()
```

Out[12]:

	vacation	credit	salary	property_value
0	0.444444	0.020349	0.330534	0.138584
1	0.428571	0.165698	0.299126	0.066076
2	0.222222	0.110465	0.548670	0.080036
3	0.015873	0.017442	0.482442	0.038246
4	0.095238	0.017442	0.180799	0.081990

```
In [13]: train_data.head()
```

Out[13]:

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

Out[14]:

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

Out[15]:

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

Out[16]:

	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 [17]: # 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 [18]: dtest['cclass']=dtest['class']
dtest=dtest.drop("class",axis=1)
dtest.head()
```

Out[18]:

	c_type	lifestyle	vacation	credit	salary	property_value	cclass
0	student	spend>>saving	0.444444	0.020349	0.330534	0.138584	C1
1	student	spend<<saving	0.428571	0.165698	0.299126	0.066076	C1
2	engineer	spend>saving	0.222222	0.110465	0.548670	0.080036	C1
3	librarian	spend<saving	0.015873	0.017442	0.482442	0.038246	C1
4	librarian	spend>saving	0.095238	0.017442	0.180799	0.081990	C1

FROM SCRATCH

```

In [19]: 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=True)
        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: 20.0% with k= 1

Accuracy: 10.0% with k= 3

Accuracy: 10.0% with k= 5

```

Accuracy: 15.0% with k= 7
Accuracy: 10.0% with k= 9
Accuracy: 10.0% with k= 11
Accuracy: 10.0% with k= 13
Accuracy: 10.0% with k= 15
Accuracy: 10.0% with k= 17
Accuracy: 10.0% with k= 19
Accuracy: 10.0% with k= 21
Accuracy: 10.0% with k= 23
Accuracy: 10.0% with k= 25

```

Using KNN Classifier

```

In [20]: from sklearn.model_selection import train_test_split
         from sklearn.neighbors import KNeighborsClassifier
         from sklearn.metrics import accuracy_score

```

```

In [21]: x_train= dtrain.loc[:, 'c_type': 'property_value']
         y_train= dtrain.loc[:, ['cclass']]

```

```

In [22]: x_train.head()

```

Out[22]:

	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 [23]: from sklearn.preprocessing import LabelEncoder
         lb_make = LabelEncoder()
         x_train["type"] = lb_make.fit_transform(x_train["c_type"])
         x_train["type"].value_counts()

```

Out[23]:

1	49
3	39
4	37
0	37
2	23

Name: type, dtype: int64

```
In [24]: x_train["style"] = lb_make.fit_transform(x_train["lifestyle"])
x_train["style"].value_counts()
```

```
Out[24]: 3    86
1     41
2     38
0     20
Name: style, dtype: int64
```

```
In [25]: x_train.head()
```

```
Out[25]:
```

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

```
Out[26]:
```

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

```
Out[27]:
```

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

Out[28]:

	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 [29]: y_train.head()
```

Out[29]:

	cclass
0	C1
1	C1
2	C1
3	C1
4	C1

```
In [30]: x_test=dtest.loc[:, 'c_type': 'property_value']
y_test=dtest.loc[:, ['cclass']]
```

```
In [31]: x_test["type"] = lb_make.fit_transform(x_test["c_type"])
x_test["type"].value_counts()
```

Out[31]:

4	6
3	5
2	3
1	3
0	3

Name: type, dtype: int64

```
In [32]: x_test["style"] = lb_make.fit_transform(x_test["lifestyle"])
x_test["style"].value_counts()
```

Out[32]:

3	8
2	7
1	4
0	1

Name: style, dtype: int64


```
In [33]: x_test=x_test[["type","style","vacation","credit","salary","property_value"]]
x_test.head()
```

Out[33]:

	type	style	vacation	credit	salary	property_value
0	4	2	0.444444	0.020349	0.330534	0.138584
1	4	0	0.428571	0.165698	0.299126	0.066076
2	1	3	0.222222	0.110465	0.548670	0.080036
3	2	1	0.015873	0.017442	0.482442	0.038246
4	2	3	0.095238	0.017442	0.180799	0.081990

```
In [34]: x_test["type"] = x_test['type'].astype(float)
x_test.head()
```

Out[34]:

	type	style	vacation	credit	salary	property_value
0	4.0	2	0.444444	0.020349	0.330534	0.138584
1	4.0	0	0.428571	0.165698	0.299126	0.066076
2	1.0	3	0.222222	0.110465	0.548670	0.080036
3	2.0	1	0.015873	0.017442	0.482442	0.038246
4	2.0	3	0.095238	0.017442	0.180799	0.081990

```
In [35]: x_test["style"] = x_test['style'].astype(float)
x_test.head()
```

Out[35]:

	type	style	vacation	credit	salary	property_value
0	4.0	2.0	0.444444	0.020349	0.330534	0.138584
1	4.0	0.0	0.428571	0.165698	0.299126	0.066076
2	1.0	3.0	0.222222	0.110465	0.548670	0.080036
3	2.0	1.0	0.015873	0.017442	0.482442	0.038246
4	2.0	3.0	0.095238	0.017442	0.180799	0.081990

```
In [36]: for K in range(20):
          K_value = K+1
          neigh = KNeighborsClassifier(n_neighbors = K_value, weights='uniform', algorithm='brute')
          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  15.0 % 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  15.0 % 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  15.0 % for K-Value: 15
Accuracy is  10.0 % for K-Value: 16
Accuracy is  10.0 % for K-Value: 17
Accuracy is  15.0 % for K-Value: 18
Accuracy is  10.0 % for K-Value: 19
Accuracy is  15.0 % 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 shape 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 shape 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 shape 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 shape 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 shape 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 shape 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 shape 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 shape 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 shape 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 shape 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 shape 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 shape 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 shape 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 shape 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 shape 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 shape 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 shape 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 shape of y to (n_samples, ), for example using ravel().
    after removing the cwd from sys.path.
```