SCDM Lab - 2

Faculty-in-charge: Dr. D.K. Shaw, Dr. Ashok Kumar Mehta

Data Mining Lab Assignment 1:

Data : 04/02/2022

Name- Ravi Kumar

Reg- 2020pgcaca72

In [ ]:

```
```

Que 1. Load ‘spambase.csv’ file in jupyter notebook.

In [4]:

```
import numpy as np
import pandas as pd
```

In [5]:

```
df=pd.read_csv('spambase.csv')
```

In [6]:

```
df
```

Out[6]:

	0	0.64	0.64.1	0.1	0.32	0.2	0.3	0.4	0.5	0.6	...	0.40	0.41	0.42	0.778
0	0.21	0.28	0.50	0.0	0.14	0.28	0.21	0.07	0.00	0.94	...	0.000	0.132	0.0	0.372
1	0.06	0.00	0.71	0.0	1.23	0.19	0.19	0.12	0.64	0.25	...	0.010	0.143	0.0	0.276
2	0.00	0.00	0.00	0.0	0.63	0.00	0.31	0.63	0.31	0.63	...	0.000	0.137	0.0	0.137
3	0.00	0.00	0.00	0.0	0.63	0.00	0.31	0.63	0.31	0.63	...	0.000	0.135	0.0	0.135
4	0.00	0.00	0.00	0.0	1.85	0.00	0.00	1.85	0.00	0.00	...	0.000	0.223	0.0	0.000
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4595	0.31	0.00	0.62	0.0	0.00	0.31	0.00	0.00	0.00	0.00	...	0.000	0.232	0.0	0.000
4596	0.00	0.00	0.00	0.0	0.00	0.00	0.00	0.00	0.00	0.00	...	0.000	0.000	0.0	0.353
4597	0.30	0.00	0.30	0.0	0.00	0.00	0.00	0.00	0.00	0.00	...	0.102	0.718	0.0	0.000
4598	0.96	0.00	0.00	0.0	0.32	0.00	0.00	0.00	0.00	0.00	...	0.000	0.057	0.0	0.000
4599	0.00	0.00	0.65	0.0	0.00	0.00	0.00	0.00	0.00	0.00	...	0.000	0.000	0.0	0.125

4600 rows × 58 columns

In [ ]:

```

```

Que 2. Print the top 5 and last 5 data of the above dataset.

In [33]:

```
print("Top 5 data of data set are Below.\n")
df.head(5)
```

```
Top 5 data of data set are Below.
```

Out[33]:

	0	0.64	0.64.1	0.1	0.32	0.2	0.3	0.4	0.5	0.6	...	0.40	0.41	0.42	0.778	0.43
0	0.21	0.28	0.50	0.0	0.14	0.28	0.21	0.07	0.00	0.94	...	0.00	0.132	0.0	0.372	0.180
1	0.06	0.00	0.71	0.0	1.23	0.19	0.19	0.12	0.64	0.25	...	0.01	0.143	0.0	0.276	0.184
2	0.00	0.00	0.00	0.0	0.63	0.00	0.31	0.63	0.31	0.63	...	0.00	0.137	0.0	0.137	0.000
3	0.00	0.00	0.00	0.0	0.63	0.00	0.31	0.63	0.31	0.63	...	0.00	0.135	0.0	0.135	0.000
4	0.00	0.00	0.00	0.0	1.85	0.00	0.00	1.85	0.00	0.00	...	0.00	0.223	0.0	0.000	0.000

5 rows × 58 columns

In [25]:

```
print("\n\nLast 5 data of data set are Below. \n")
df.tail(5)
```

```
Last 5 data of data set are Below.
```

Out[25]:

	0	0.64	0.64.1	0.1	0.32	0.2	0.3	0.4	0.5	0.6	...	0.40	0.41	0.42	0.778	0.43
4595	0.31	0.0	0.62	0.0	0.00	0.31	0.0	0.0	0.0	0.0	...	0.000	0.232	0.0	0.000	0.0
4596	0.00	0.0	0.00	0.0	0.00	0.00	0.0	0.0	0.0	0.0	...	0.000	0.000	0.0	0.353	0.0
4597	0.30	0.0	0.30	0.0	0.00	0.00	0.0	0.0	0.0	0.0	...	0.102	0.718	0.0	0.000	0.0
4598	0.96	0.0	0.00	0.0	0.32	0.00	0.0	0.0	0.0	0.0	...	0.000	0.057	0.0	0.000	0.0
4599	0.00	0.0	0.65	0.0	0.00	0.00	0.0	0.0	0.0	0.0	...	0.000	0.000	0.0	0.125	0.0

5 rows × 58 columns

In [ ]:

```

```

Que: 3. Print the total instances per class of the above dataset.

In [ ]:

```
```

4. Split the data part and label part of the above dataset.

In [11]:

```
df.shape
```

Out[11]:

```
(4600, 58)
```

In [12]:

```
[row,col]=df.shape
```

In [13]:

```
Data=df.iloc[0 : row , 0 : (col-1)]
Label=df.iloc[0 : row , (col-1)]
```

In [15]:

```
Data.shape
```

Out[15]:

```
(4600, 57)
```

In [31]:

```
Label.value_counts()
```

Out[31]:

```
0    2788
1    1812
Name: 1, dtype: int64
```

In [37]:

```
Data
```

Out[37]:

	0	0.64	0.64.1	0.1	0.32	0.2	0.3	0.4	0.5	0.6	...	0.39	0.40	0.41	0.42	0
0	0.21	0.28	0.50	0.0	0.14	0.28	0.21	0.07	0.00	0.94	...	0.0	0.000	0.132	0.0	0
1	0.06	0.00	0.71	0.0	1.23	0.19	0.19	0.12	0.64	0.25	...	0.0	0.010	0.143	0.0	0
2	0.00	0.00	0.00	0.0	0.63	0.00	0.31	0.63	0.31	0.63	...	0.0	0.000	0.137	0.0	0
3	0.00	0.00	0.00	0.0	0.63	0.00	0.31	0.63	0.31	0.63	...	0.0	0.000	0.135	0.0	0
4	0.00	0.00	0.00	0.0	1.85	0.00	0.00	1.85	0.00	0.00	...	0.0	0.000	0.223	0.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
4595	0.31	0.00	0.62	0.0	0.00	0.31	0.00	0.00	0.00	0.00	...	0.0	0.000	0.232	0.0	0
4596	0.00	0.00	0.00	0.0	0.00	0.00	0.00	0.00	0.00	0.00	...	0.0	0.000	0.000	0.0	0
4597	0.30	0.00	0.30	0.0	0.00	0.00	0.00	0.00	0.00	0.00	...	0.0	0.102	0.718	0.0	0
4598	0.96	0.00	0.00	0.0	0.32	0.00	0.00	0.00	0.00	0.00	...	0.0	0.000	0.057	0.0	0
4599	0.00	0.00	0.65	0.0	0.00	0.00	0.00	0.00	0.00	0.00	...	0.0	0.000	0.000	0.0	0

4600 rows × 57 columns

In [38]:

```
Data.describe()
```

Out[38]:

	0	0.64	0.64.1	0.1	0.32	0.2	
count	4600.000000	4600.000000	4600.000000	4600.000000	4600.000000	4600.000000	4600.000
mean	0.104576	0.212922	0.280578	0.065439	0.312222	0.095922	0.114
std	0.305387	1.290700	0.504170	1.395303	0.672586	0.273850	0.391
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000
75%	0.000000	0.000000	0.420000	0.000000	0.382500	0.000000	0.000
max	4.540000	14.280000	5.100000	42.810000	10.000000	5.880000	7.270

8 rows × 57 columns

5. Find the accuracy of this using kNN algorithm.

In [73]:

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
import numpy as np

X = Data
y = Label
# Split into training and test set

X_train, X_test, y_train, y_test = train_test_split(
X, y, test_size = 0.2)

knn = KNeighborsClassifier(n_neighbors= 4)
knn.fit(X_train, y_train)
print(knn.score(X_test, y_test))
```

```
0.8217391304347826
```

In [74]:

```
y_test.value_counts()
```

Out[74]:

```
0    549
1    371
Name: 1, dtype: int64
```

In [60]:

```
y_train.value_counts()
```

Out[60]:

```
0    2245
1    1435
Name: 1, dtype: int64
```

In [70]:

```
y_train.value_counts()+y_test.value_counts()
```

Out[70]:

```
0    2788
1    1812
Name: 1, dtype: int64
```

In [67]:

```
Label.count()
```

Out[67]:

```
4600
```

In [ ]:

```

```