

```
In [ ]: # assignment 5
```

```
In [ ]: # spambase
```

```
In [7]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
In [8]: df = pd.read_csv('spambase.csv')
```

```
In [9]: df
```

Out[9]:

	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	0.4
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	0.18
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	0.18
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	0.00
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	0.00
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	0.00
...
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	0.00
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	0.00
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	0.00
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	0.00
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	0.00

4600 rows × 58 columns



```
In [10]: [row, col] = df.shape
Data = df.iloc[0 : row , 0 : (col - 1)]
Label = df.iloc[0 : row , (col - 1)]
```

```
In [ ]: # DT
```

```
In [15]: import warnings
warnings.filterwarnings("ignore")
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import precision_score, recall_score, f1_score, accuracy_score
```

```
In [16]: X_train, X_test, y_train, y_test = train_test_split(Data, Label, test_size = 0.1,
```

```
In [17]: dt = DecisionTreeClassifier(random_state = 0, max_depth = 2)
dt.fit(X_train, y_train)
y_pred = dt.predict(X_test)
```

```
In [18]: print('Accuracy: %.4f' % accuracy_score(y_test, y_pred))
```

Accuracy: 0.8435

```
In [ ]: # kNN
```

```
In [19]: from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
X = Data
y = Label
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))
```

0.8076086956521739

```
In [ ]: # k-Means
```

```
In [20]: from sklearn.model_selection import train_test_split
from sklearn.cluster import KMeans
X = Data
y = Label
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state = 0)
kmns = KMeans(n_clusters = 2, random_state = 0)
kmns.fit(X_train, y_train)
```

```
Out[20]: KMeans(n_clusters=2, random_state=0)
```

```
In [ ]: # heart failure clinical records dataset
```

```
In [21]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
In [22]: df = pd.read_csv('heart_failure_clinical_records_dataset.csv')
```

```
In [23]: df
```

```
Out[23]:
```

	age	anaemia	creatinine_phosphokinase	diabetes	ejection_fraction	high_blood_pressure	p
0	75.0	0	582	0	20	1	26
1	55.0	0	7861	0	38	0	26
2	65.0	0	146	0	20	0	16
3	50.0	1	111	0	20	0	21
4	65.0	1	160	1	20	0	32
...
294	62.0	0	61	1	38	1	15
295	55.0	0	1820	0	38	0	27
296	45.0	0	2060	1	60	0	74
297	45.0	0	2413	0	38	0	14
298	50.0	0	196	0	45	0	39

299 rows × 13 columns



```
In [24]: [row, col] = df.shape
Data = df.iloc[0 : row , 0 : (col - 1)]
Label = df.iloc[0 : row , (col - 1)]
```

```
In [ ]: #DT
```

```
In [25]: import warnings
warnings.filterwarnings("ignore")
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import precision_score, recall_score, f1_score, accuracy_score
```

```
In [26]: X_train, X_test, y_train, y_test = train_test_split(Data, Label, test_size = 0.1,
```

```
In [27]: dt = DecisionTreeClassifier(random_state = 0, max_depth = 2)
dt.fit(X_train, y_train)
y_pred = dt.predict(X_test)
```

```
In [28]: print('Accuracy: %.4f' % accuracy_score(y_test, y_pred))
```

Accuracy: 0.8000

```
In [ ]: # kNN
```

```
In [29]: from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
X = Data
y = Label
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))

0.7
```

```
In [ ]: # k-Means
```

```
In [31]: from sklearn.model_selection import train_test_split
from sklearn.cluster import KMeans
X = Data
y = Label
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20, random_state = 0)
kmns = KMeans(n_clusters = 2, random_state = 0)
kmns.fit(X_train, y_train)
```

```
Out[31]: KMeans(n_clusters=2, random_state=0)
```

```
In [ ]:
```