

Q3

```
In [ ]: from sklearn.neighbors import KNeighborsClassifier
        from sklearn.model_selection import train_test_split
        from sklearn.metrics import accuracy_score, confusion_matrix
        from sklearn.datasets import load_iris
        import numpy as np
        import pandas as pd
        from matplotlib import pyplot as plt
```

```
In [ ]:
```

```
In [ ]: X = load_iris(as_frame=True)["data"]
        y = load_iris(as_frame=True)["target"]
        X.columns
```

```
Out[ ]: Index(['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)',
              'petal width (cm)'],
              dtype='object')
```

```
In [ ]: def normalise(feature, df):
        mean = df[feature].mean()
        sd = df[feature].std()
        df[feature] = (df[feature] - mean) / sd

        normalise("sepal length (cm)", X)
        normalise("sepal width (cm)", X)
        normalise("petal length (cm)", X)
        normalise("petal width (cm)", X)
        X
```

```
Out[ ]:
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	-0.897674	1.015602	-1.335752	-1.311052
1	-1.139200	-0.131539	-1.335752	-1.311052
2	-1.380727	0.327318	-1.392399	-1.311052
3	-1.501490	0.097889	-1.279104	-1.311052
4	-1.018437	1.245030	-1.335752	-1.311052
...
145	1.034539	-0.131539	0.816859	1.443994
146	0.551486	-1.278680	0.703564	0.919223
147	0.793012	-0.131539	0.816859	1.050416
148	0.430722	0.786174	0.930154	1.443994
149	0.068433	-0.131539	0.760211	0.788031

150 rows × 4 columns

```
In [ ]: X_train, X_temp, y_train, y_temp = train_test_split(X , y,random_state=100, test_si
X_validate, X_test, y_validate, y_test = train_test_split(X_temp , y_temp,random_st
```

```
In [ ]:
```

```
In [ ]: def report_show(x,y, k):
        knn = KNeighborsClassifier(n_neighbors=k, weights="distance")
        knn.fit(X_train, y_train)
        preds = knn.predict(x)
        print(f"Accuracy : {accuracy_score(y , preds)}")
        print(f"ConfusionMatrix\n{confusion_matrix(y , preds)}")
```

```
In [ ]: # k = 4
        print("Validation")
        report_show(X_validate, y_validate, 4)
        print("Testing")
        report_show(X_test, y_test, 4)
```

```
Validation
Accuracy : 0.9411764705882353
ConfusionMatrix
[[6 0 0]
 [0 4 0]
 [0 1 6]]
Testing
Accuracy : 1.0
ConfusionMatrix
[[6 0 0]
 [0 4 0]
 [0 0 7]]
```

```
In [ ]: # k = 5
print("Validation")
report_show(X_validate, y_validate, 5)
print("Testing")
report_show(X_test, y_test, 5)
```

```
Validation
Accuracy : 0.9411764705882353
ConfusionMatrix
[[6 0 0]
 [0 4 0]
 [0 1 6]]
Testing
Accuracy : 1.0
ConfusionMatrix
[[6 0 0]
 [0 4 0]
 [0 0 7]]
```

```
In [ ]: # k = 8
print("Validation")
report_show(X_validate, y_validate, 2)
print("Testing")
report_show(X_test, y_test, 1)
```

```
Validation
Accuracy : 0.9411764705882353
ConfusionMatrix
[[6 0 0]
 [0 4 0]
 [0 1 6]]
Testing
Accuracy : 1.0
ConfusionMatrix
[[6 0 0]
 [0 4 0]
 [0 0 7]]
```

best k = 4