In [1]:
```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, classification_report

iris =
load_iris() X =
iris.data y =
iris.target

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_

lr =
LogisticRegression(max_iter=200)
lr.fit(X_train, y_train) lr_pred =
lr.predict(X_test)

print("Logistic Regression:")
print("Accuracy:", accuracy_score(y_test, lr_pred))
print(classification_report(y_test, lr_pred))

knn =
KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train, y_train) knn_pred =
knn.predict(X_test)

print("\nK-Nearest Neighbors:")
print("Accuracy:", accuracy_score(y_test, knn_pred))
print(classification_report(y_test, knn_pred))
```

```
Logistic Regression:
Accuracy: 0.9777777777777777
              precision    recall  f1-score
support

0       1.00      1.00      1.00        16
1       1.00      0.94      0.97        18
        2       0.92      1.00      0.96        11

    accuracy                           0.98
45    macro avg       0.97      0.98      0.98
45 weighted avg       0.98      0.98      0.98
45


K-Nearest Neighbors:
Accuracy: 0.9777777777777777
              precision    recall  f1-score
support

0       1.00      1.00      1.00        16
1       1.00      0.94      0.97        18
        2       0.92      1.00      0.96        11

    accuracy                           0.98
45    macro avg       0.97      0.98      0.98
45 weighted avg       0.98      0.98      0.98
45
```

In [ ]:

7/8/25

Week-3

Aim: To compare different classification algorithms using a open-source dataset based on accuracy, precision, recall and F1-score.

Procedure:

1) Import the required libraries.
2) Load the iris dataset (built in).
3) Split the dataset into training and test sets.
4) Train two classifiers:
   - → Logistic Regression
   - → K-Nearest Neighbours.
5) Use both models to make predictions.
6) Evaluate each model using:
   - → Accuracy
   - → Classification Report.
7) Print and compare the results.

Code:

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import kNeighborsClassifier
from sklearn.metrics import accuracy_score, classification_report

iris = load_iris()
X = iris.data
Y = iris.target
X_train, X_test, Y_train, Y_test = train_test_split
                    (X, Y, test_size=0.3, random_state=0)

lr = LogisticRegression(max_iter=200)
lr.fit(X_train, Y_train)
lr_pred = lr.predict(X_test)
```

```
print ("Logistic Regression :")
print ("Accuracy :", accuracy_score (y_test, lr_pred))
print (classification_report (y_test, lr_pred))

knn = KNeighborsClassifier (n_neighbors=3)
knn.fit (x_train, y_train)
knn_pred = knn.predict (x_test)
print ("\n k-Nearest Neighbors:")
print ("Accuracy :", accuracy_score (y_test, knn_pred))
print (classification_report (y_test, knn_pred))
```

output :-

Logistic Regression :

Accuracy :- 0.977

	precision	recall	F1-score	support
0	1.00	1.00	1.00	16
1	1.00	0.94	0.97	18
2	0.92	1.00	0.96	11
accuracy			0.98	45
macro avg	0.97	0.98	0.98	45

K - Nearest Neighbors :

Accuracy : 0.9777

	precision	recall	F1-score	support
0	1.00	1.00	1.00	16
1	1.00	0.94	0.97	18
2	0.92	1.00	0.96	11
accuracy			0.98	45
macro avg	0.97	0.98	0.98	45

Accuracy = $\frac{\text{Total positive + Total Negative}}{TP + TN + FP + FN}$

precision = $\frac{TP}{TP + FP}$ (How many are actual positive

Recall = $\frac{TP}{TP + FN}$ (How many actual positives

F1 score = Harmonic mean of precision and Recall.

Result :- In this experiment I have successfully implemented the comparing of different classification algorithms using open source dataset.