Exp. No. 8. Write a program to implement k-Nearest Neighbour algorithm to classify the iris data set. Print both correct and wrong predictions. Java/Python ML library classes can be used for this problem.

#Python Program to Implement the k-Nearest Neighbour Algorithm

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

names = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'Class']

# Read dataset to pandas dataframe
dataset = pd.read_csv("8-dataset.csv", names=names)
X = dataset.iloc[:, :-1]
y = dataset.iloc[:, -1]
print(X.head())
Xtrain, Xtest, ytrain, ytest = train_test_split(X, y, test_size=0.10)

classifier = KNeighborsClassifier(n_neighbors=5).fit(Xtrain, ytrain)

ypred = classifier.predict(Xtest)

i = 0
print ("\n-------------------------------------------------------------------")
print ('%-25s %-25s %-25s' % ('Original Label', 'Predicted Label', 'Correct/Wrong'))
print ("-------------------------------------------------------------------")
for label in ytest:
    print ('%-25s %-25s' % (label, ypred[i]), end="")
    if (label == ypred[i]):
        print (' %-25s' % ('Correct'))
    else:
        print (' %-25s' % ('Wrong'))
    i = i + 1
print ("-------------------------------------------------------------------")
print("\nConfusion Matrix:\n",metrics.confusion_matrix(ytest, ypred))
print ("-------------------------------------------------------------------")
print("\nClassification Report:\n",metrics.classification_report(ytest, ypred))
print ("-------------------------------------------------------------------")
print('Accuracy of the classifer is %0.2f' % metrics.accuracy_score(ytest,ypred))
print ("-------------------------------------------------------------------")
```

OUTPUT:

	sepal-length	sepal-width	petal-length	petal-width
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

```
-------------------------------------------------------------------
Original Label          Predicted Label         Correct/Wrong
-------------------------------------------------------------------
Iris-virginica          Iris-virginica          Correct
Iris-setosa             Iris-setosa             Correct
Iris-virginica          Iris-virginica          Correct
Iris-versicolor         Iris-versicolor         Correct
```

```
Iris-virginica      Iris-virginica      Correct
Iris-versicolor     Iris-versicolor     Correct
Iris-virginica      Iris-virginica      Correct
Iris-versicolor     Iris-versicolor     Correct
Iris-versicolor     Iris-versicolor     Correct
Iris-virginica      Iris-virginica      Correct
Iris-versicolor     Iris-versicolor     Correct
Iris-setosa         Iris-setosa         Correct
Iris-virginica      Iris-virginica      Correct
Iris-versicolor     Iris-virginica      Wrong
Iris-virginica      Iris-virginica      Correct
----------------------------------------------------------------------

Confusion Matrix:
 [[2 0 0]
 [0 5 1]
 [0 0 7]]
----------------------------------------------------------------------

Classification Report:
                  precision    recall  f1-score   support

     Iris-setosa       1.00      1.00      1.00         2
 Iris-versicolor       1.00      0.83      0.91         6
  Iris-virginica       0.88      1.00      0.93         7

        accuracy                           0.93        15
       macro avg       0.96      0.94      0.95        15
    weighted avg       0.94      0.93      0.93        15

----------------------------------------------------------------------
Accuracy of the classifer is 0.93
```

Refer the video: https://www.youtube.com/watch?v=hBiywaFK-xQ

OR