**ML3**

**import** pandas **as** pd

**import** numpy **as** np

**import** seaborn **as** sns

**import** matplotlib.pyplot **as** plt

**%**matplotlib inline

**import** warnings

warnings**.**filterwarnings('ignore')

**from** sklearn.model\_selection **import** train\_test\_split

**from** sklearn.svm **import** SVC

**from** sklearn **import** metrics

df**=**pd**.**read\_csv('diabetes.csv')

df**.**columns

df**.**isnull()**.**sum()

X **=** df**.**drop('Outcome',axis **=** 1)

y **=** df['Outcome']

**from** sklearn.preprocessing **import** scale

X **=** scale(X)

*# split into train and test*

X\_train, X\_test, y\_train, y\_test **=** train\_test\_split(X, y, test\_size **=** 0.3, random\_state **=** 42)

**from** sklearn.neighbors **import** KNeighborsClassifier

knn **=** KNeighborsClassifier(n\_neighbors**=**7)

knn**.**fit(X\_train, y\_train)

y\_pred **=** knn**.**predict(X\_test)

print("Confusion matrix: ")

cs **=** metrics**.**confusion\_matrix(y\_test,y\_pred)

print(cs)

print("Acccuracy ",metrics**.**accuracy\_score(y\_test,y\_pred))

total\_misclassified **=** cs[0,1] **+** cs[1,0]

print(total\_misclassified)

total\_examples **=** cs[0,0]**+**cs[0,1]**+**cs[1,0]**+**cs[1,1]

print(total\_examples)

print("Error rate",total\_misclassified**/**total\_examples)

print("Error rate ",1**-**metrics**.**accuracy\_score(y\_test,y\_pred))

print("Precision score",metrics**.**precision\_score(y\_test,y\_pred))

print("Recall score ",metrics**.**recall\_score(y\_test,y\_pred))

print("Classification report ",metrics**.**classification\_report(y\_test,y\_pred))