**PRACTICAL 6**

**Aim: Implementation and evaluate any Classification Algorithm using Python / R /Scilab.**

# Naive Bayes

import pandas as pd

from sklearn.naive\_bayes import GaussianNB

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

from sklearn.preprocessing import OneHotEncoder

from sklearn.compose import ColumnTransformer

from sklearn.metrics import accuracy\_score, classification\_report

# Load the CSV file into a Pandas dataframe

data = pd.read\_csv('XyzNaive.csv')

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

    data[['age', 'income', 'student', 'credit\_rating']],

    data['buys\_computer'], test\_size=0.3, random\_state=42)

# One-hot encode the categorical features

categorical\_cols = ['age', 'income', 'student', 'credit\_rating']

ct = ColumnTransformer([('encoder', OneHotEncoder(), categorical\_cols)], remainder='passthrough')

X\_train = ct.fit\_transform(X\_train)

X\_test = ct.transform(X\_test)

# Create a Naive Bayes classifier object

clf = GaussianNB()

# Train the classifier on the training data

clf.fit(X\_train, y\_train)

# Use the trained classifier to make predictions on the test data

y\_pred = clf.predict(X\_test)

# Evaluate the accuracy of the classifier

accuracy = accuracy\_score(y\_test, y\_pred)

# Calculate precision, recall, specificity, and sensitivity

report = classification\_report(y\_test, y\_pred, target\_names=['no', 'yes'], output\_dict=True)

precision = report['yes']['precision']

recall = report['yes']['recall']

specificity = report['no']['recall']

sensitivity = recall

# Print the results

print(f'Accuracy: {accuracy:.2f}')

print(f'Precision: {precision:.2f}')

print(f'Recall: {recall:.2f}')

print(f'Specificity: {specificity:.2f}')

print(f'Sensitivity: {sensitivity:.2f}')

# decision tree

import pandas as pd

from sklearn.tree import DecisionTreeClassifier

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

from sklearn.preprocessing import OneHotEncoder

from sklearn.compose import ColumnTransformer

from sklearn.metrics import accuracy\_score, classification\_report

# Load the CSV file into a Pandas dataframe

data = pd.read\_csv('XyzNaive.csv')

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(

    data[['age', 'income', 'student', 'credit\_rating']],

    data['buys\_computer'], test\_size=0.3, random\_state=42)

# One-hot encode the categorical features

categorical\_cols = ['age', 'income', 'student', 'credit\_rating']

ct = ColumnTransformer([('encoder', OneHotEncoder(), categorical\_cols)], remainder='passthrough')

X\_train = ct.fit\_transform(X\_train)

X\_test = ct.transform(X\_test)

# Create a decision tree classifier object

clf = DecisionTreeClassifier(random\_state=42)

# Train the classifier on the training data

clf.fit(X\_train, y\_train)

# Use the trained classifier to make predictions on the test data

y\_pred = clf.predict(X\_test)

# Evaluate the accuracy of the classifier

accuracy = accuracy\_score(y\_test, y\_pred)

# Calculate precision, recall, specificity, and sensitivity

report = classification\_report(y\_test, y\_pred, target\_names=['no', 'yes'], output\_dict=True)

precision = report['yes']['precision']

recall = report['yes']['recall']

specificity = report['no']['recall']

sensitivity = recall

# Print the results

print(f'Accuracy: {accuracy:.2f}')

print(f'Precision: {precision:.2f}')

print(f'Recall: {recall:.2f}')

print(f'Specificity: {specificity:.2f}')

print(f'Sensitivity: {sensitivity:.2f}')