**CA 614 – Data Mining and Analytics**

**Assignment – III**

**Classification and Prediction**

1. **Read the seeds.csv file**

import pandas as pd

df = pd.read\_csv("C:\\Users\\student\\Downloads\\seeds.csv")

print(df)

df.info()

df.info()

print(df)

df.describe()

df.isnull()

1. **Perform all data pre-processing techniques.**

df.info()

df.isnull()

df.isnull().sum()

df.isnull().sum()

1. **Perform one-hot encoding and normalization.**

#Task : To categorise the countries using socio-economic and health factors

#that determine the overall development of the country.

import numpy as np

import pandas as pd

import statsmodels.api as sm

import random as rd

import matplotlib.pyplot as plt

import seaborn as sns

sns.set()

data = pd.read\_csv('C:\\Users\\student\\Downloads\\seeds.csv')

data

#Apply one hot encoding to categorical column

data1 = pd.get\_dummies(data, columns = ['Type'])

data1

#Apply normalization to the dataframe as k-means is distance-based algorithm

from sklearn import preprocessing

data2 = preprocessing.normalize(data1)

data2

1. **Apply decision tree algorithm and display classification report.**

import numpy as np

import pandas as pd

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

from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import accuracy\_score

from sklearn import tree

seed\_data = pd.read\_csv('C:\\Users\\student\\Downloads\\seeds.csv', sep= ',', header= None)

print(seed\_data)

print("Dataset Lenght:: ", len(seed\_data))

print("Dataset Shape:: ", seed\_data.shape)

print("Dataset:: ")

seed\_data.head()

X = seed\_data.values[1:, 0:6]

Y = seed\_data.values[1:,7]

print(Y)

print(X)

X\_train, X\_test, y\_train, y\_test = train\_test\_split( X, Y, test\_size = 0.3, random\_state = 100)

clf\_gini = DecisionTreeClassifier(criterion = "gini", random\_state = 100,

                               max\_depth=3, min\_samples\_leaf=5)

clf\_gini.fit(X\_train, y\_train)

y\_pred = clf\_gini.predict(X\_test)

y\_pred

print("Accuracy is ", accuracy\_score(y\_test,y\_pred)\*100)

from sklearn.metrics import confusion\_matrix

print(confusion\_matrix(y\_test, y\_pred))

from sklearn.metrics import classification\_report

print(classification\_report(y\_test, y\_pred))

clf\_entropy = DecisionTreeClassifier(criterion = "entropy", random\_state = 100,

 max\_depth=3, min\_samples\_leaf=5)

clf\_entropy.fit(X\_train, y\_train)

y\_pred = clf\_entropy.predict(X\_test)

y\_pred

print("Accuracy is ", accuracy\_score(y\_test,y\_pred)\*100)

from sklearn.metrics import confusion\_matrix

print(confusion\_matrix(y\_test, y\_pred))

from sklearn.metrics import classification\_report

print(classification\_report(y\_test, y\_pred))

1. **Apply naïve-byes algorithm and display classification report.**

import numpy as np

import pandas as pd

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

from sklearn.metrics import accuracy\_score

from sklearn.naive\_bayes import GaussianNB

seed\_data = pd.read\_csv('C:\\Users\\91972\\Downloads\\seeds.csv', sep= ',', header= None)

X = seed\_data.values[1:, 0:6]

Y = seed\_data.values[1:,7]

X\_train, X\_test, y\_train, y\_test = train\_test\_split( X, Y, test\_size = 0.3, random\_state = 100)

from sklearn.naive\_bayes import GaussianNB

# Build a Gaussian Classifier

model = GaussianNB()

# Model training

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

y\_pred

from sklearn.metrics import (

    accuracy\_score,

    confusion\_matrix,

    ConfusionMatrixDisplay,

    f1\_score,

)

print("Accuracy is ", accuracy\_score(y\_test,y\_pred)\*100)

f1 = f1\_score(y\_pred, y\_test, average="weighted")

print("F1 Score:", f1)

from sklearn.metrics import confusion\_matrix

print(confusion\_matrix(y\_test, y\_pred))

labels = ['1','2','3']

cm = confusion\_matrix(y\_test, y\_pred, labels=labels)

disp = ConfusionMatrixDisplay(confusion\_matrix=cm, display\_labels=labels)

disp.plot();

from sklearn.metrics import classification\_report

print(classification\_report(y\_test, y\_pred))