**STATISTICAL MACHINE LEARNING**

**B Rithwik**

**2303A52330**

**ASSIGINMENT 6**

***Question 1***

**Code –**

**from google.colab import drive**

**drive.mount('/content/drive')**

import pandas as pd

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

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score

data = pd.read\_csv("/content/drive/MyDrive/SML Dataset/Assignment\_6.csv")

X = data.drop('Pass/Fail', axis=1)

y = data['Pass/Fail']

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

clf = RandomForestClassifier(random\_state=42)

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

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

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

print(f"Model Accuracy: {accuracy \* 100:.2f}%")

**OUTPUT –**

Model Accuracy: 100.00%

***Question 2***

***Code –***

import pandas as pd

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

from sklearn.preprocessing import StandardScaler, LabelEncoder

from sklearn.linear\_model import Perceptron

from sklearn.metrics import accuracy\_score

df = pd.read\_csv("/content/drive/MyDrive/SML Dataset/diabetes\_data\_upload.csv")

df = df.dropna()

label\_encoder = LabelEncoder()

df['Gender'] = label\_encoder.fit\_transform(df['Gender'])

for column in df.columns:

if df[column].isin(['Yes', 'No']).any():

df[column] = label\_encoder.fit\_transform(df[column])

X = df.drop('class', axis=1)

y = df['class']

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

scaler = StandardScaler()

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

X\_test = scaler.transform(X\_test)

perceptron = Perceptron(max\_iter=1000, tol=1e-3, random\_state=42)

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

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

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

print(f"Model accuracy: {accuracy \* 100:.2f}%")

OUTPUT –

Model accuracy: 89.74%