Assignment – 8

# B.Rithwik

# 2303A52330

# Batch – 35

**Question - 1**

import pandas as pd

from google.colab import drive

import matplotlib.pyplot as plt

import seaborn as sns

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

from sklearn.linear\_model import LogisticRegression

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

file\_path = ("/content/drive/MyDrive/SML Dataset/pass\_fail\_data.csv")

df = pd.read\_csv(file\_path)

print("Data Preview:")

print(df.head())

print("\nData Shape:")

print(df.shape)

print("\nData Description:")

print(df.describe())

print("\nPass/Fail Value Counts:")

print(df['Pass/Fail'].value\_counts())

sns.pairplot(df, hue='Pass/Fail')

plt.show()

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

y = df['Pass/Fail']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model = LogisticRegression()

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

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

print("\nModel Accuracy:", accuracy\_score(y\_test, y\_pred))

print("\nClassification Report:")

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

**OUTPUT –**

Data Preview:

X1 X2 X3 X4 X5 Pass/Fail

0 10 90 85 1 100 1

1 5 60 70 0 80 0

2 15 95 90 1 100 1

3 2 30 50 0 40 0

4 12 85 88 1 90 1

Data Shape:

(10, 6)

Data Description:

X1 X2 X3 X4 X5 Pass/Fail

count 10.000000 10.000000 10.000000 10.000000 10.00000 10.000000

mean 8.600000 73.000000 73.600000 0.500000 79.50000 0.500000

std 4.526465 21.390548 15.629033 0.527046 19.06859 0.527046

min 2.000000 30.000000 50.000000 0.000000 40.00000 0.000000

25% 5.250000 61.250000 61.250000 0.000000 71.25000 0.000000

50% 9.000000 80.000000 75.000000 0.500000 82.50000 0.500000

75% 11.750000 89.500000 87.250000 1.000000 93.75000 1.000000

max 15.000000 95.000000 93.000000 1.000000 100.00000 1.000000

Pass/Fail Value Counts:

Pass/Fail

1 5

0 5

Name: count, dtype: int64

**A screenshot of a graph

Description automatically generated**

Model Accuracy: 1.0

Classification Report:

precision recall f1-score support

0 1.00 1.00 1.00 1

1 1.00 1.00 1.00 1

accuracy 1.00 2

macro avg 1.00 1.00 1.00 2

weighted avg 1.00 1.00 1.00 2

**Question – 2**

from google.colab import drive

drive.mount('/content/drive')

import pandas as pd

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

from sklearn.preprocessing import StandardScaler, LabelEncoder

from sklearn.linear\_model import LogisticRegression

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)

logistic\_model = LogisticRegression(max\_iter=1000, random\_state=42)

logistic\_model.fit(X\_train, y\_train)

y\_pred = logistic\_model.predict(X\_test)

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

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

**OUTPUT -**

Model accuracy: 93.59%