Ex.No: 5.a **DIABETES CLASSIFICATION** Date: 24-Jan-2025

**Aim:-**

To train a logistic regression model to accurately predict diabetes based on health metrics.

**Program Code:-**

import pandas as pd

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

from sklearn.linear\_model import LogisticRegression

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

# Load the dataset

data = pd.read\_csv('Diabetes.csv')

#Preview the dataset

print("Preview the data")

print(data.head())

# Select features and target variable

X = data.drop('Outcome', axis=1)

y = data['Outcome']

# Split the data into training and testing sets

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

# Initialize and train the logistic regression model

model = LogisticRegression(max\_iter=1000)

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

# Make predictions on the test set

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

# Evaluate the model

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

conf\_matrix = confusion\_matrix(y\_test, y\_pred)

class\_report = classification\_report(y\_test, y\_pred)

print(f'Accuracy: {accuracy:.2f}')

print('Confusion Matrix:')

print(conf\_matrix)

print('Classification Report:')

print(class\_report)

**Output:-**

Preview the data

Pregnancies Glucose BloodPressure SkinThickness Insulin BMI \

0 6 148 72 35 0 33.6

1 1 85 66 29 0 26.6

2 8 183 64 0 0 23.3

3 1 89 66 23 94 28.1

4 0 137 40 35 168 43.1

DiabetesPedigreeFunction Age Outcome

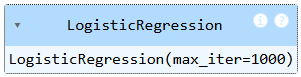
0 0.627 50 1

1 0.351 31 0

2 0.672 32 1

3 0.167 21 0

4 2.288 33 1



Accuracy: 1.00

Confusion Matrix:

[[1]]

Classification Report:

precision recall f1-score support

0 1.00 1.00 1.00 1

accuracy 1.00 1

macro avg 1.00 1.00 1.00 1

weighted avg 1.00 1.00 1.00 1

**Result:-**

Thus, the program was successfully executed.

Ex.No: 5.b **CREDIT CARD DEFAULT PREDICTIONS** Date: 24-Jan-2025

**Aim:-**

To train a logistic regression model to accurately predict credit card default using customer data.

**Program Code:-**

import pandas as pd

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

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score, confusion\_matrix

#Load the data

data=pd.read\_csv('Creditcard.csv')

#Preview the data

print("Preview the dataset")

print(data.head())

# Select features and target variable

X = data.drop('Default', axis=1)

y = data['Default']

# Split the data into training and testing sets

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

# Initialize and train the logistic regression model

model = LogisticRegression(max\_iter=1000)

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

# Make predictions on the test set

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

# Evaluate the model

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

conf\_matrix = confusion\_matrix(y\_test, y\_pred)

# Print results

print(f'Accuracy: {accuracy:.2f}')

print('Confusion Matrix:')

print(conf\_matrix)

# Print predictions

predictions = pd.DataFrame({'CreditScore': X\_test['CreditScore'], 'Actual': y\_test, 'Predicted': y\_pred})

print(predictions)

**Output:-**

Preview the dataset

CreditScore Age Income LoanAmount Default

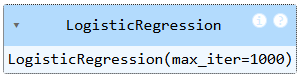
0 700 34 50000 20000 0

1 600 45 45000 15000 1

2 650 29 30000 12000 0

3 720 41 60000 25000 0

4 580 36 32000 10000 1



Accuracy: 1.00

Confusion Matrix:

[[1]]

CreditScore Actual Predicted

1 600 1 1

**Result:-**

Thus, the program was successfully executed.

Ex.No: 5.c **HEART DISEASE CLASSIFICATION** Date: 24-Jan-2025

**Aim:-**

To train a logistic regression model to accurately classify heart disease based on various health indicators.

**Program Code:-**

import pandas as pd

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

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score, confusion\_matrix

# Load the dataset

data = pd.read\_csv('Heartdisease.csv')

#Preview the dataset

print(f"Preview the dataset")

print(data.head())

# Select features and target variable

X = data.drop('HeartDisease', axis=1)

y = data['HeartDisease']

# Split the data into training and testing sets

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

# Initialize and train the logistic regression model

model = LogisticRegression(max\_iter=1000)

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

# Make predictions on the test set

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

# Evaluate the model

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

conf\_matrix = confusion\_matrix(y\_test, y\_pred)

# Print results

print(f'Accuracy: {accuracy:.2f}')

print('Confusion Matrix:')

print(conf\_matrix)

# Print predictions

predictions = pd.DataFrame({'Age': X\_test['Age'], 'Actual': y\_test, 'Predicted': y\_pred})

print(predictions)

**Output:-**

Preview the dataset

Age Sex ChestPainType RestingBP Cholesterol FastingBS RestingECG \

0 63 1 3 145 233 1 0

1 37 1 2 130 250 0 1

2 41 0 1 130 204 0 0

3 56 1 1 120 236 0 1

4 57 0 0 120 354 0 1

MaxHR ExerciseAngina Oldpeak ST\_Slope HeartDisease

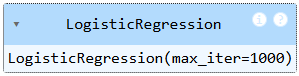
0 150 0 2.3 0 1

1 187 0 3.5 1 1

2 172 0 1.4 2 1

3 178 0 0.8 2 1

4 163 1 0.6 2 0



Accuracy: 1.00

Confusion Matrix:

[[1]]

Age Actual Predicted

1 37 1 1

**Result:-**

Thus, the program was executed successfully.