**Experiment – 10: Apply logistic regression algorithm for a classification problem.**

**Source 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, classification\_report, confusion\_matrix

# Load the dataset (replace 'your\_dataset.csv' with your actual file path)

# Ensure your dataset has a column for the target variable (e.g., 'Class' for breast\_cancer dataset)

data = pd.read\_csv('/content/breast\_cancer.csv')

# Define features (X) and target variable (y)

# Assuming 'Class' is the target variable for breast\_cancer dataset

X = data.drop('Class', axis=1)  # Features (all columns except 'Class')

y = data['Class']  # Target variable ('Class')

# Split 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)  # 80% train, 20% test

# Initialize and train a Logistic Regression model

logreg\_model = LogisticRegression(max\_iter=1000)  # You can adjust parameters as needed

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

# Make predictions on the test set

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

# Evaluate the model

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

print(f"Accuracy: {accuracy}")

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

# Confusion Matrix

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

print("Confusion Matrix:\n", cm)

**Output:**

Accuracy: 0.9562043795620438

precision recall f1-score support

2 0.94 0.99 0.96 79

4 0.98 0.91 0.95 58

accuracy 0.96 137

macro avg 0.96 0.95 0.95 137

weighted avg 0.96 0.96 0.96 137

Confusion Matrix:

[[78 1]

[ 5 53]]