NAME:PAREENITA A.SHIRSATH PRN:221101062 ROLL.NO:57 T.E.A.I.&.D.S.

**ML EXPERIMENT NO : 4**

import pandas as pd

import matplotlib.pyplot as plt

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

from sklearn.linear\_model import LogisticRegression # Importing LogisticRegression

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

# Load data

data = pd.read\_csv("temperatures.csv")

# Check if the data is loaded correctly and inspect the column names

print("Data loaded successfully. Columns:", data.columns)

# Ensure the column "YEAR" exists in the dataframe

if "YEAR" in data.columns:

x = data["YEAR"].values # No reshape here, just the raw values

else:

raise ValueError("Column 'YEAR' not found in the CSV file")

y = data["ANNUAL"]

# Convert 'ANNUAL' to binary values (e.g., classify temperatures above a certain threshold)

threshold = y.mean() # Using the mean as a threshold for classification

y\_binary = (y > threshold).astype(int) # Convert to 1 if above threshold, else 0

# Split the data into training and test sets

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x.reshape(-1, 1), y\_binary, test\_size=0.2, random\_state=0)

# Create the Logistic Regression model and fit it to the training data

model = LogisticRegression()

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

# Make predictions on the test set

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

# Print accuracy and confusion matrix

print("Accuracy of Logistic Regression model:", accuracy\_score(y\_test, y\_pred))

print("Confusion Matrix:\n", confusion\_matrix(y\_test, y\_pred))

# Plotting the decision boundary and data points

plt.figure(figsize=(10, 6)) # Optional: Specify figure size

plt.title("Logistic Regression: Temperature Classification")

plt.xlabel("Year")

plt.ylabel("Temperature (Binary Classification)")

# Scatter plot for the original data

plt.scatter(x, y\_binary, color='blue', label='Full Data', alpha=0.5)

# Plot the logistic regression curve

import numpy as np

x\_range = np.linspace(x.min(), x.max(), 100).reshape(-1, 1)

y\_range = model.predict\_proba(x\_range)[:, 1] # Get probabilities for the positive class

plt.plot(x\_range, y\_range, color='green', label='Logistic Regression Curve')

# Show the legend to distinguish the points

plt.legend()

# Display the plot

plt.show()



