**Implementation of SVM**

import pandas as pd

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

from sklearn.svm import SVC

from sklearn.metrics import accuracy\_score, precision\_score, recall\_score, f1\_score

df = pd.read\_csv('/content/heart.csv')

# Define the features (independent variables)

X = df.drop(columns=['HeartDisease']) # Replace 'target' with the actual name of your target column

# Define the target variable (dependent variable)

y = df['HeartDisease'] # Replace 'target' with the actual name of your target column

# Split the dataset 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)

# Create an SVM classifier

svm\_classifier = SVC(kernel='linear', C=2.0)

# Train the classifier

svm\_classifier.fit(X\_train, y\_train)

# Make predictions on the test set

y\_pred = svm\_classifier.predict(X\_test) # Make predictions on the test set

y\_pred = svm\_classifier.predict(X\_test) # Evaluate the performance

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

precision = precision\_score(y\_test, y\_pred, average='weighted')

recall = recall\_score(y\_test, y\_pred, average='weighted')

f1 = f1\_score(y\_test, y\_pred, average='weighted')

print(f"Accuracy: {accuracy}")

print(f"Precision: {precision}")

print(f"Recall: {recall}")

print(f"F1 Score: {f1}")

regression

import numpy as np

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

from sklearn.svm import SVR

from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error, r2\_score

from scipy.stats import pearsonr

# Load data set

import pandas as pd

df = pd.read\_csv('/content/heart.csv')

df

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

y = df['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 SVR model

svm\_regressor = SVR(kernel='linear')

# Train the model

svm\_regressor.fit(X\_train, y\_train)

# Make predictions on the test set

y\_pred = svm\_regressor.predict(X\_test)

# Evaluate the model

mae = mean\_absolute\_error(y\_test, y\_pred)

mse = mean\_squared\_error(y\_test, y\_pred)

rmse = np.sqrt(mse)

r2 = r2\_score(y\_test, y\_pred)

correlation, \_ = pearsonr(y\_test, y\_pred)

# Display the evaluation metrics

print(f'Mean Absolute Error (MAE): {mae:.4f}')

print(f'Mean Squared Error (MSE): {mse:.4f}')

print(f'Root Mean Squared Error (RMSE): {rmse:.4f}')

print(f'R-squared (R²): {r2:.4f}')

print(f'Correlation: {correlation:.4f}')