**Project Title: Heart Disease Prediction using Logistic Regression**

**Introduction:**

The project aimed to create a smart computer program that can predict if someone might have a heart problem using a method called logistic regression. Heart issues are a big problem, and finding them early can help a lot. We wanted to use the power of computers to learn from information and help doctors identify potential problems early on. This way, we hope to make a positive impact on people's health by catching and addressing heart-related issues before they become more serious.

**Objective:**

The primary objective was to create a logistic regression model capable of predicting the likelihood of heart disease based on relevant medical attributes. The model aimed to assist healthcare professionals in identifying high-risk individuals for timely intervention.

**Dataset:**

The dataset used for training and evaluation contained records of individuals with and without heart disease. Attributes included demographic information, lifestyle factors, and medical indicators such as cholesterol levels, blood pressure, and chest pain results.

data **=** pd**.**read\_csv('Heart\_Disease\_Prediction.csv')

Dataset Used: <https://www.kaggle.com/datasets/rishidamarla/heart-disease-prediction>

**Data Preprocessing:**

Data preprocessing comprises tasks such as adjusting data types to ensure uniformity, but without removing any columns deemed important. No missing values were found in the dataset, eliminating the need for column deletion.

df['HeartDisease'] **=** pd**.**to\_numeric(df['HeartDisease']**.**map({'Presence':1,'Absence':0}))

**Model Development:**

Logistic regression, a well-suited algorithm for binary classification problems, was chosen for its simplicity and interpretability. The dataset underwent a crucial step of splitting into training and testing sets, facilitating model training on the training set and subsequent evaluation on the test set. This splitting was accomplished using the following code snippet:

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

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

This code ensures that 80% of the data is allocated for training the logistic regression model, while the remaining 20% is reserved for evaluating its predictive performance. The random\_state parameter ensures reproducibility of the split, maintaining consistency in subsequent model evaluations. The logistic regression model then learns patterns from the training set, and its efficiency is assessed on the unseen test set.

**Evaluation Metrics:**

The model's performance was evaluated using various metrics, including accuracy, precision, recall, F1 score, and the confusion matrix. These metrics provided a comprehensive understanding of the model's predictive capabilities and its ability to correctly classify individuals with and without heart disease.

**Accuracy:**

**def** calculate\_accuracy(y\_true, y\_pred):

correct\_predictions **=** np**.**sum(y\_true **==** y\_pred)

total\_predictions **=** len(y\_true)

accuracy **=** (correct\_predictions **/** total\_predictions) **\*** 100

**return** accuracy

**Confusion Matrix:**

**def** calculate\_confusion\_matrix(y\_true, y\_pred):

tp **=** np**.**sum((y\_true **==** 1) **&** (y\_pred **==** 1))

tn **=** np**.**sum((y\_true **==** 0) **&** (y\_pred **==** 0))

fp **=** np**.**sum((y\_true **==** 0) **&** (y\_pred **==** 1))

fn **=** np**.**sum((y\_true **==** 1) **&** (y\_pred **==** 0))

confusion\_matrix **=** np**.**array([[tn, fp], [fn, tp]])

**return** confusion\_matrix

**F1 score:**

**def** calculate\_f1\_score(y\_true, y\_pred):

tp **=** np**.**sum((y\_true **==** 1) **&** (y\_pred **==** 1))

fp **=** np**.**sum((y\_true **==** 0) **&** (y\_pred **==** 1))

fn **=** np**.**sum((y\_true **==** 1) **&** (y\_pred **==** 0))

precision **=** tp **/** (tp **+** fp)

recall **=** tp **/** (tp **+** fn)

f1\_score **=** 2 **\*** (precision **\*** recall) **/** (precision **+** recall)

**return** f1\_score

**Recall:**

**def** calculate\_recall(y\_true, y\_pred):

tp **=** np**.**sum((y\_true **==** 1) **&** (y\_pred **==** 1))

fn **=** np**.**sum((y\_true **==** 1) **&** (y\_pred **==** 0))

recall **=** tp **/** (tp **+** fn)

**return** recall

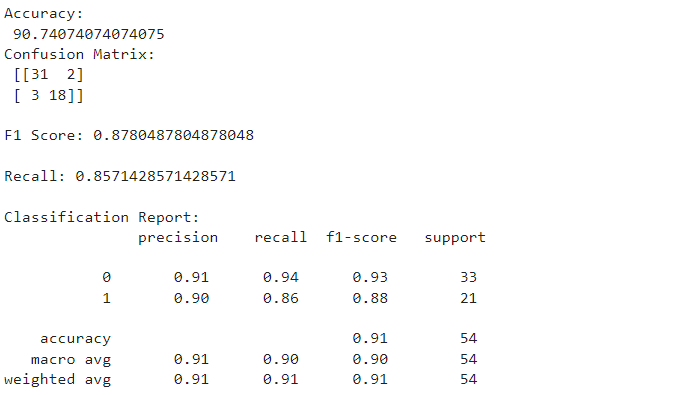
**Classification Report:**

**from** sklearn.metrics **import** classification\_report

classification\_rep **=** classification\_report(Y\_test**.**ravel(), Y\_Test\_Predict**.**ravel())

**Results:**

The logistic regression model achieved a commendable 90.74% accuracy in predicting heart disease. With a balanced F1 score of 0.88 and recall of 0.86%, the model demonstrated reliability in both precision and sensitivity, showcasing its potential for effective early diagnosis in healthcare applications.



**Conclusion:**

In conclusion, the developed logistic regression model for heart disease prediction exhibited robust performance with an accuracy of 90.74%. The balanced F1 score and recall indicate its reliability in identifying individuals at risk. The model holds promise for aiding healthcare professionals in early diagnosis, contributing to improved patient outcomes and preventive interventions. Further refinement and exploration of additional features could enhance its predictive capabilities. This project underscores the potential of machine learning in augmenting healthcare practices for proactive cardiac care.

**Future Work:**

Future work could involve expanding the dataset, incorporating advanced feature engineering techniques, and exploring other machine learning algorithms for comparison. Additionally, collaboration with healthcare professionals for domain-specific insights and continuous model monitoring and updates would contribute to the ongoing improvement of the predictive system.

In conclusion, the heart disease prediction project showcased the potential of logistic regression in the healthcare domain, contributing to the ongoing efforts to leverage machine learning for early disease detection and improved patient outcomes.