# *RAJALAKSHMI EDUVERSE*

**DATA SCIENCE FROM PYTHON SCRATCH**

**PROJECT BY**


## NAME: R. KALAISELVAN

## YEAR: 2023 BATCH


## CAPSTONE PROJECT: HEART DISEASE PREDICTION

**Machine Learning Project Problem Statement: Predicting Heart Disease**

**Problem Statement:**

The objective of this project is to predict the presence of heart disease in patients based on various medical parameters. The dataset used for this project is the heart disease dataset, which contains 14 attributes and 303 instances. The attributes include age, sex, chest pain type, resting blood pressure, serum cholesterol, fasting blood sugar, resting electrocardiographic results, maximum heart rate achieved, exercise-induced angina, ST depression induced by exercise relative to rest, slope of the peak exercise ST segment, number of major vessels colored by fluoroscopy, and thalassemia. The target variable is the presence of heart disease, which is represented by a binary variable (0 = no heart disease, 1 = heart disease).

To solve this problem, we will use a classification algorithm to predict the presence of heart disease in patients. We will first preprocess the data by performing exploratory data analysis, handling missing values, and scaling the data. We will then train and evaluate several classification models, including logistic regression, decision trees, and random forests. We will select the best model based on its performance on the test set and use it to predict the presence of heart disease in new patients.

## Objective:

The objective of this project is to develop a machine learning model that can predict the likelihood of heart disease in individuals based on their medical and clinical attributes. The model should classify individuals into two categories: those with heart disease and those without.

## Dataset:

The dataset used for this project is "heart_disease.csv," which contains information on various health-related attributes and a binary target variable indicating the presence or absence of heart disease.

## Steps to Achieve the Objective:

**IMPORTING LIBRARIES:**

**1.Data Acquisition:**

```
import pandas as pd
```

- Download the heart disease dataset from a reliable source such as Kaggle Machine Learning Repository.
- Load the "heart disease.csv" dataset, which contains historical patient data

**2.Data Cleaning:**

- Remove any duplicates, handle missing values, and remove any irrelevant columns.

**3.Exploratory Data Analysis (EDA):**

```
import seaborn as sns  # For data visualization

import matplotlib.pyplot as plt  # For data visualization
```

- Explore and visualize the dataset to gain insights into the relationships between different features and the presence of heart disease.
- Identify any correlations or patterns that may aid in classification.

**4.Feature Engineering:**

**from sklearn.feature_selection import SelectKBest**

**from sklearn.decomposition import PCA**

- Engineer new features or preprocess existing ones if needed
- Perform one-hot encoding or label encoding for categorical variables.

**5.Data Splitting:**

**from sklearn.model_selection import train_test_split # For data splitting**

- Split the data into training and testing sets.

**6.Model Selection:**

**from sklearn.ensemble import RandomForestClassifier**

- Select the appropriate classification algorithm(s) for the problem. Some popular algorithms for binary classification include logistic regression, decision trees, random forests, and support vector machines (SVMs).
- For building a random forest model
- You can also import other models like Logistic Regression, SVM, etc.

**7.Model Training:**

**from sklearn.model_selection import train_test_split**
**from sklearn.ensemble import RandomForestClassifier**

- Model training involves fitting the model on the training data. Here's an example of training the model.

**8.Model Prediction:**

**from sklearn.model_selection import train_test_split**
**from sklearn.ensemble import RandomForestClassifier**

- Model prediction involves making predictions on the test data. Here's an example of making predictions.

**9.Model Evaluation:**

**from sklearn.metrics import accuracy_score, classification_report**

- Evaluate the performance of the model(s) using metrics such as accuracy, precision, recall, and F1 score. And model evaluation

**10.Model Optimization:**

**from sklearn.model_selection import GridSearchCV**

- Fine-tune the model(s) by adjusting the hyperparameters and/or selecting different features.
- For hyperparameter tuning

**# Optional libraries for data visualization**

**import numpy as np**

- NumPy is used for numerical operations, and it's often used in conjunction with Pandas to handle arrays and matrices of data.

**# Additional libraries for general data manipulation and preprocessing**

**from sklearn.preprocessing import StandardScaler**

- For feature scaling
- You might need to use other preprocessing techniques like one-hot encoding or label encoding for categorical variables.