


Excited to share my latest machine learning project: [Cardiovascular disease \(CVDs\)](#)

- Cardiovascular disease (CVDs) is the number one cause of death globally, taking an estimated 17.9 million lives each year, which accounts for 31% of all deaths worldwide. Four out of five CVD deaths are due to heart attacks and strokes, and one-third of these deaths occur prematurely in people under 70 years of age. Heart failure is a common event caused by CVDs.

- This dataset contains 11 features that can be used to predict possible heart disease.

 About the Project:

I built a machine learning model(Logistic Regression& DecisionTree) to help diagnose cardiovascular disease :

- 1) load the dataset
- 2) Data preprocessing:
involves transforming raw data into a format suitable for analysis or machine learning algorithms. It typically includes various techniques such as cleaning, transforming, and encoding the data.
- 3) One-hot encoding using Pandas: I removed the binary variables, because fast encoding would do nothing for them. To achieve this, we will only count the number of different values present in each class variable and only consider variables with 3 or more values.

- data: DataFrame to be used

- prefix: A list with prefixes, so we know which value we are dealing with

- columns: the list of columns that will be one-hot encoded. 'prefix' and 'columns' must have the same length.

4)Splitting the Dataset: I split my dataset into training and testing datasets. Using the train_test_split

function from Scikit-learn.

5)Building the Models: Employed various machine learning algorithms

1) Decision Tree

-The hyperparameters I used are:

- min_samples_split: The minimum number of samples required to split the internal node.

-It may help reduce overfitting

- max_degree: maximum depth of the tree.

-It may help reduce overfitting

2)Random forest


Random Forest algorithm was also used, using Scikit-learn implementation.

- All hyperparameters present in the decision tree model will also be present in this algorithm, since a random forest is a collection of many decision trees.

3) XGBoost

Boosting methods train many trees, but rather than being unrelated to each other, The model has the same parameters as the decision tree, plus the learning rate.

- Learning rate is the step size in the Gradient Descent method that XGBoost uses internally to reduce the error in each train step

 Accuracy Validation: both Random Forest and XGBoost had similar performance (test accuracy).

- 1) load the dataset
- 2) Data preprocessing:
involves transforming raw data into a format suitable for analysis or machine learning algorithms. It typically includes various techniques such as cleaning, transforming, and encoding the data.

3) Visualizing the data: Data visualization is the process of representing data visually to gain insights and communicate patterns or trends effectively.

Some common visualization techniques used:

a. Histograms:

B. Scattered plots:

C. Bar charts:

D. Line plots:

Visualization libraries used such as Matplotlib, Seaborn, Plotly in Python

1.4 Split the data into training and test sets:

- Split the data into features (X) and target (y)

- we plot the confusion matrix using `sn.heatmap`.

5)  Accuracy Validation:

- I calculated the accuracy in the training set & test set and Plot the output

- The accuracy score achieved using Logistic Regression is: 84.78 %