Report : Part 3

Data splitting :

We split the data to give 2/3 of the whole dataset to training and 1/3 of the data for the testing part. The reason behind this split is that the amount of data is very small, consequently we need to add a little bit more data than usual to the testing part.

Non numerical data Handling

We handled the non-numerical values using the Label Encoder in the scikit learn library, it helps us transform text features into numerical ones that can be understanded by our model. The label encoder assigns a number to each new text category, so the larger the category types are, the wider will be the range of the numbers encoded.

Dealing with underfitting and hyperparameters tuning:

At first, the model had a 0.35 accuracy score which is extremely low, thus, we must do some changes in order to make our model better.

1. Dropping the chd column:

   After visualizing the correlation between the features, we can notice that the chd columns doesn't correlate with any feature. Also, removing it from our dataset increases the accuracy of our model which is good.

2. Tuning the logistic regression hyperparameters:
   a. Solver: it's the algorithm to use in the optimization problem. The choices are {'newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'}. Here we choose newton-cg. Although it's computationally expensive because of the Hessian Matrix, but it is more accurate in our case.
   b. Max_iter: which refers to the maximum iterations allowed for our model. We choose 200 for a maximum training.

c. C: refers to Regularization strength works with the penalty to regulate overfitting. Smaller values specify stronger regularization and high value tells the model to give high weight to the training data.