**Final Report**

**Team or Group Members:**

Kimaya Havle, Sifat Naseem, and Ifunanya Ezeumeh.

**Title of the research project:**

Prediction of Cardiovascular Disease Risk (CVD) using multiple Machine Learning algorithms

## **Project Problem:**

To develop a Python program using multiple machine learning algorithms to predict the risk of CVD.

**GitHub** - <https://github.com/kimayahavle/Cardiovascular-Disease-Prediction-using-ML>

<https://github.com/Sifatnaseem/Cardiovascular-Disease-Prediction-using-ML>

<https://github.com/ieezeume/Cardiovascular-Disease-Prediction-using-ML>

**Dataset -** <https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset>

## **Data Preprocessing:**

The dataset contains a population between the ages of 29 and 64 consisting of 70,000 records of patient data with the target variable (Cardiovascular disease) describing the presence or absence of cardiovascular disease using 12 features.

The following steps are done for data preprocessing:

1. Identifying and removing missing entries: Null entries, if any, are removed from the dataset. The dataset from Kaggle did not contain any null entries.
2. Identifying and removing identifiers.
3. Identifying and removing duplicates: 24 duplicate data points were identified and resolved.
4. Renaming attributes with easier to understand names: We improved the data interpretability by renaming certain columns.
5. Identifying and removing outliers: We found a few outliers in the dataset while performing exploratory data analysis. For example, the minimum weight in the dataset is 10 kg (and the population age ranges from 29 to 64). An individual >= 29 age should not weigh 10 kg..

Similarly, the minimum value in the systolic column was -150, and the maximum value was 16020, which is clearly wrong. The minimum value of the diastolic column was -70, and the maximum value was 11000, which is also wrong.

To deal with the outliers, we visualized the data and removed the data points below the 2.5 percentile and above the 97.5 percentile. We performed this outlier removal for the height, weight, and systolic and diastolic columns.

1. Visualizing the dataset distribution and correlation between features: We visualized the data distribution and feature correlation using correlation matrix and pair plots. There were no strongly correlated features so we did not reduce the dimension of the features.
2. Feature engineering: We transformed the age in days into age in years.

## **Disease Prediction Results:**

To perform the prediction, we have chosen three models –

1. Gaussian Naive Bayes

2. Support Vector Machines

3. AdaBoost

**Gaussian Naive Bayes( Ifunanya Ezeumeh)**

* The dataset was preprocessed using the steps above.
* All relevant libraries and modules were imported.
* The preprocessed dataset was then imported into the notebook.
* The Gaussian Naive Bayes classifier was initialized
* Stratified KFold with n\_split =10 was implemented to split the dataset while preserving the sample proportion.
* After splitting the dataset, the training set was used to train the model by fitting it to the classifier.
* Finally, the test set was used to make a prediction.
* All performance metrics were outputted.

The performance of my model on the preprocessed data is as follows:

**Accuracy: 71%**

**Precision: 75.5%**

**Recall: 70.7%**

**Support Vector Machine (Sifat Naseem)**

After all the preprocessing of data as explained above, I used the support Vector Machine classifier (SVM) on the dataset to make the prediction. I used Numpy and pandas to load the data then I split it into the training and test sets. For tuning of hyperparameters, I used the radial basis function (RBF) kernel as the given dataset is non-linear, and for regularization, I have used parameter c = 5. I tested the model with the c in the range of 1 to 5 and got the best performance with the c value as 5. After training the model, I evaluated it on the test data and finally used the trained model to make the prediction on new data.

The performance of my model on the preprocessed data is as follows:

**Accuracy: 71%**

**Precision: 77%**

**Recall: 58%**

**AdaBoost (Kimaya Havle)**

We have trained the dataset using the AdaBoost classifier, a class of ensemble machine learning algorithms, using the SKLearn Package.

We have used Pandas to load the preprocessed dataset and split it into training and testing data, where 75% is chosen for training and 25% for testing.

The AdaBoost classifier is trained on the training dataset and tested using the test dataset. The initial accuracy we got was 72.28%. We have further tuned the hyperparameters using SKLearn’s GridSearch to increase the accuracy.

The hyperparameters used for tuning are ‘n\_estimators’ and ‘learning\_rate.’ The values used for ‘n\_estimators’ are [250,500,750] and for ‘learning\_rate’ are [0.01, 0.1, 1]. ‘n\_estimators’ are the number of weak learners we want to use. ‘learning\_rate’ is provided to shrink the contribution of each classifier.

After tuning the hyperparameters, we got 72.37% as the best accuracy using ‘n\_estimator’ = 500 and ‘learning\_rate’ = 1.

The performance of my model on the preprocessed data is as follows:

**Accuracy: 72.37%**

**Precision: 75.28%**

**Recall: 63.63%**

**Conclusion**

Of the three models, Adaboost had the highest accuracy, SVM had the highest precision and Gaussian Naive Bayes had the highest recall.