**Detection of Cardio Vascular Disease**

ALY 6015 INTERMEDIATE ANALYTICS

Debpriyo Roy

Harshal Kumbhar

Northeastern University



Note:

Debpriyo Roy, Department of Enterprise Intelligence, College of Professional Studies, Northeastern University, Boston, MA 02115.

Contact: [roy.de@husky.neu.edu](mailto:roy.de@husky.neu.edu)

kumbhar.h@husky.neu.edu

**INTRODUCTION**

Cardiovascular Diseases (CVDs) are one of the most prominent reason for death globally in the past few decades. On an average of 38 seconds, someone dies due to CVDs which equals 2302 deaths per day as per the report form WHO, it also says that approximately 17 million people are dying each year due to cardiovascular diseases. Predicting CVDs among patients is tough job for medical practitioner as it involves around 14-15 different factors. We will be creating model in order to predict the presence of CVDs in a person by focusing on the key responsible factors. Our aim is to predict the pattern by applying different data mining and predictive analysis techniques which may lead to presence heart disease. To predict the pattern, we have referred the data set from Kaggle website which contains the data of cardiovascular disease patients consisting 70,000 observations with 11 different responsible factors. We wish to forecast the presence of CVDs in patient’s body with available parameters and further focusing on key factors with respect to our target variable which can help in making our model more efficient and successful. We will be using different classification techniques and analyzing method in order to do so. To start with models we are going to use Naïve Bayes which is simple probabilistic classifier based on Bayes algorithm, which results in conditional probability model, Random forest uses assembly of decision trees in order to predict the accuracy based on multiple parameters. Lastly XG Boost uses decision tree created on collective Machine Learning techniques which uses gradient boosting framework.

**Analysis**

The dataset taken in consideration for this project is based on Cardiovascular disease (CVD) detection, which contains 70,000 records of patient’s data in 12 features, such gender, systolic blood pressure, diastolic blood pressure, etc.  The target variable is denoted by ‘cardio’ which is binary as ‘0’ represents absence of the CVD and ‘1’ represents presence of CVD.

The input features are divided into three categories:

1. Objective: Describes factual information.
2. Examination: Describes the result of medical examination.
3. Subjective: Describes the information obtained by patients.

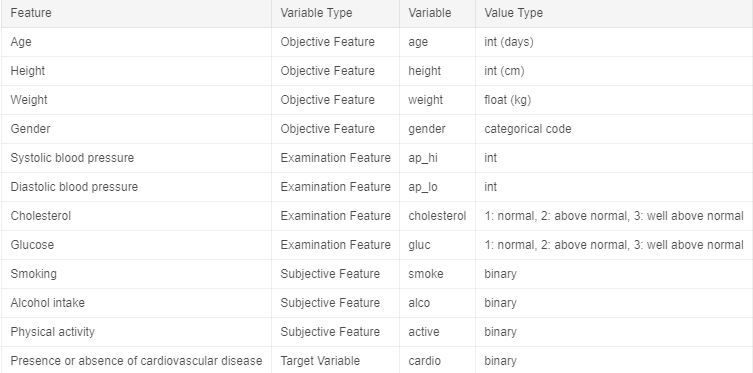


Figure1: Different categories of input features.

The preprocessing of the dataset is as follows:

1. Descriptive Analysis
2. The first step after importing the dataset is to understand the data, by viewing it and performing Descriptive analysis. This can be performed by using the describe () function in R.

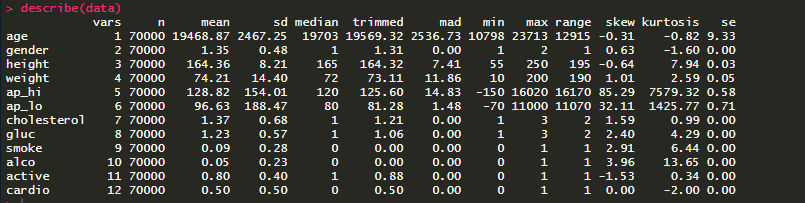


Figure 2: Descriptive Analysis.

It is evident from the above figure that the data is not scaled as it has varying mean and variance.

1. The next step is to check the data type of each feature, as the dataset should contain features of both numerical and categorical datatype. This step can be performed by using the str () function.

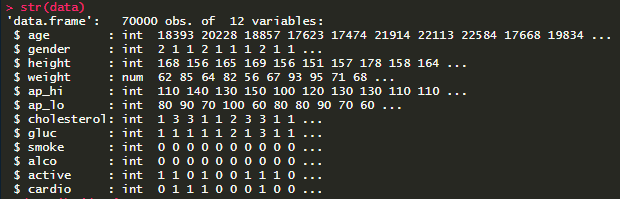


Figure 2: Class of the features.

From the above figure, it can be seen that numerical features are correctly categorized as ‘int’ but the categorical values such as gender, smoke are classified as numerical, whereas in order to build a model, these features need to be categorized as ‘factor’ datatype.

1. After conducting basic analysis, the next step is to visualize the data and get more information to understand the data in hand.

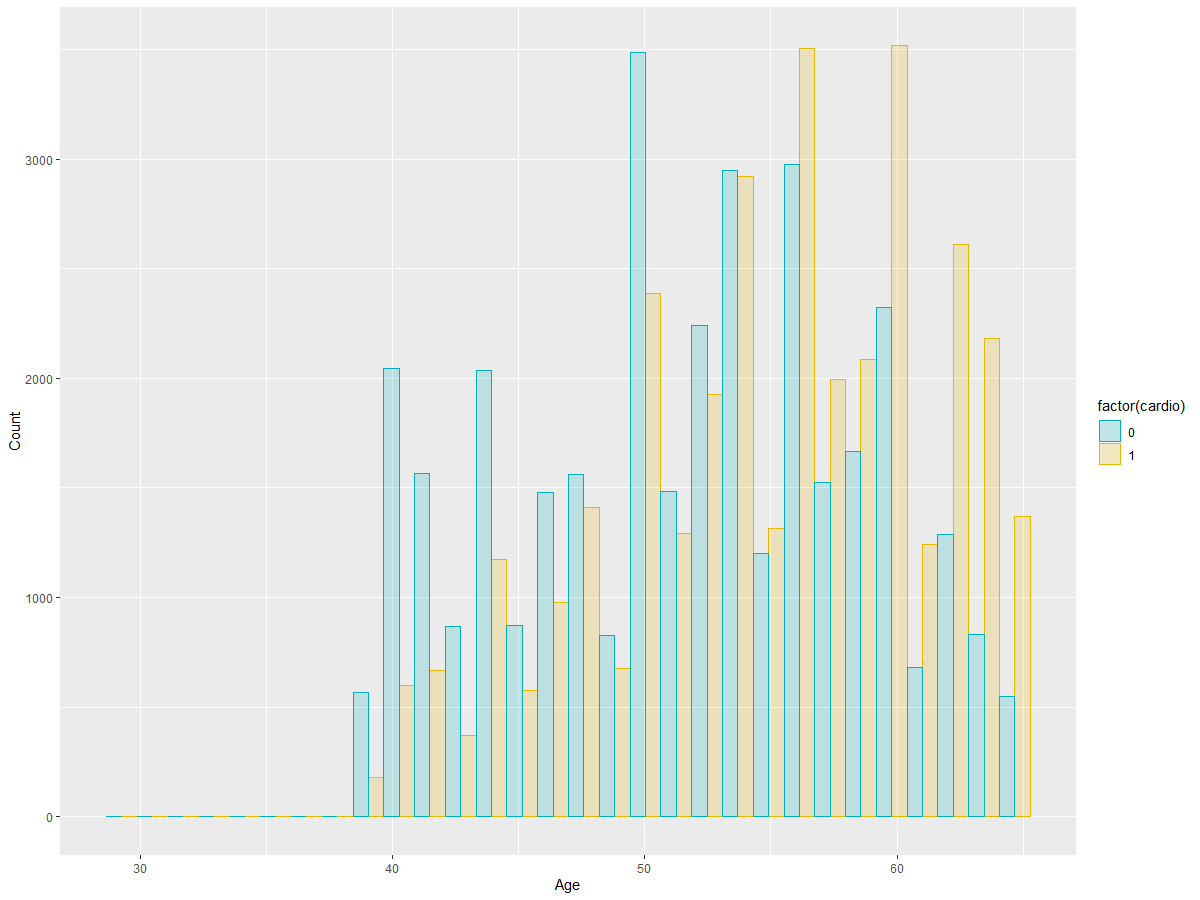


Figure 3: Age affected by CVD

From the above plot, it is clear that age plays a crucial role in the presence of CVD. Patients above the age of 55 are prone to CVD than others.

1. Now, to check for others features which share a pattern towards contributing to the presence of CVD.

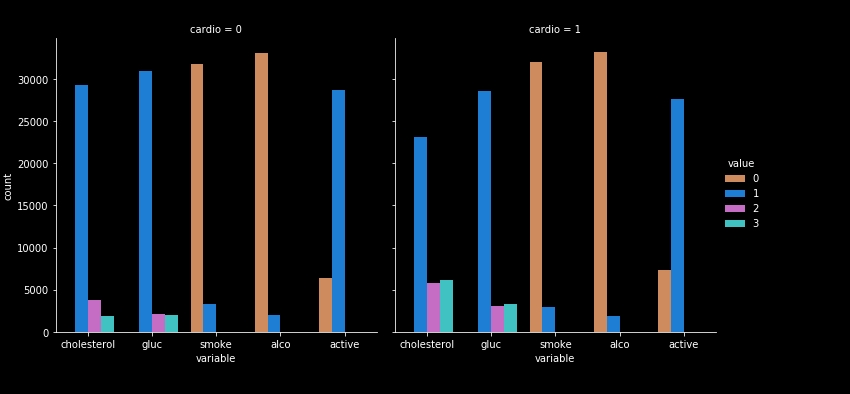


Figure 4: Effect of features on CVD.

From the above plot it can be concluded that people with CVD have high cholesterol, glucose level and do less physical activity.

1. The next step is to check if the data has any outliers.

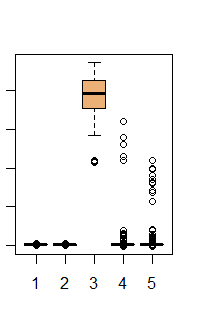


Figure 5: Boxplot

As it can be seen from the figure, there are many outliers in the data which if not taken care would result in skewness in the data.

1. Data Cleaning

Data cleaning is a very important aspect before building any model. Data cleaning is the process of removing the data which is not useful or might result in skewness. It is performed as follows:

1. Removing missing data in the dataset.



Figure 6: Missing Data

After a thorough check of the dataset, it can be seen there are no missing data in the dataset.

1. Removing duplicate data in the dataset.



Figure 7: Duplicates

As we can see from the above figure, there are 24 duplicate items which need to be removed before proceeding.

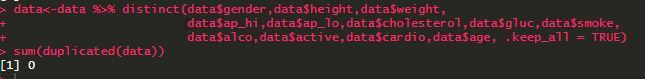


Figure 8: Duplicates removed

The duplicates in the data have been successfully removed.

1. Removing outliers which can skew the data.

As from figure 4, there are outliers present in the data which needs dealing. Outliers are the data points which don’t contribute towards the predictive analysis.

In our data, it can be seen that minimum height is 55 cm and minimum weight is 10 kg which are erroneous data, since the minimum age is 29 years and, in some cases, diastolic blood pressure is higher than systolic blood pressure and also blood pressure is denoted by negative values which surely are errors.

These outliers are treated by removing data pertaining to the above-mentioned features which lie below 2.5% quantile range and above 97.5% quantile range.

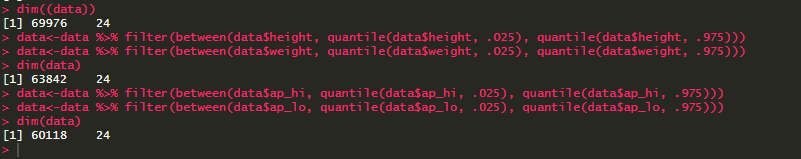


Figure 9: Outliers treatment

From the above figure, the outliers treatment has been implemented. Now to check the presence of any more outliers a box plot can be used.

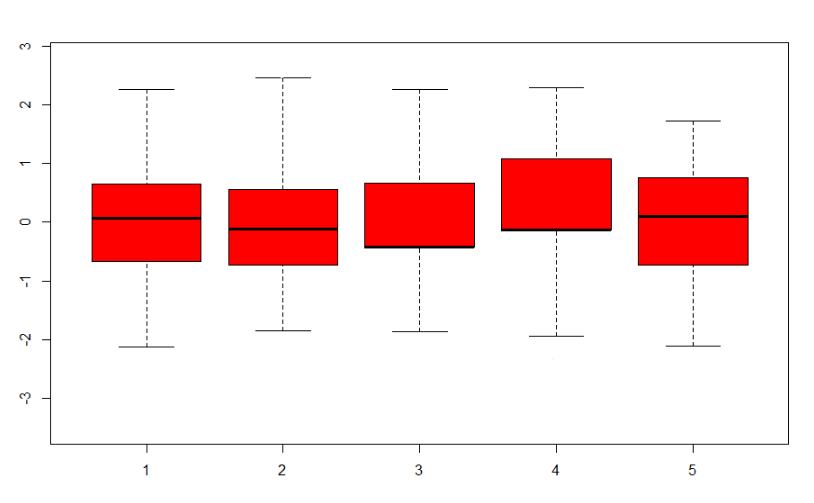


Figure 10: Box- Plot

The above figure shows that, all the outliers present in the data have been removed.

1. After removing the outliers, convert the class of categorical values to factors.

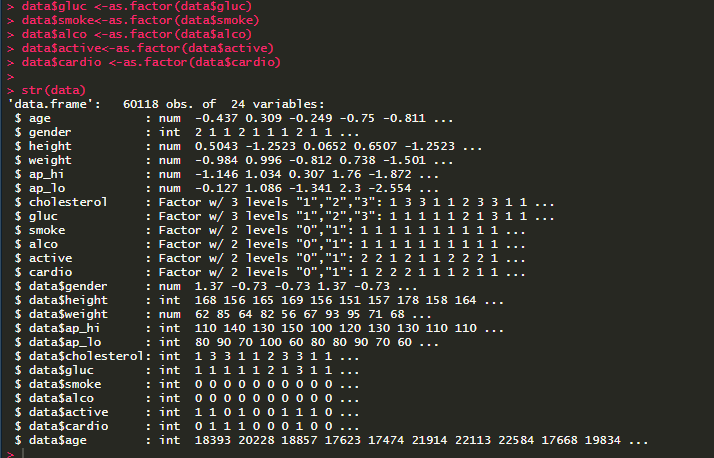


Figure 11: Converting categorical values

1. Feature Engineering

Feature engineering is the science of selecting features which are contributing more towards prediction analysis than others and creating more features using the existing features.

The correlation of the features can be visualized by using a heatmap.

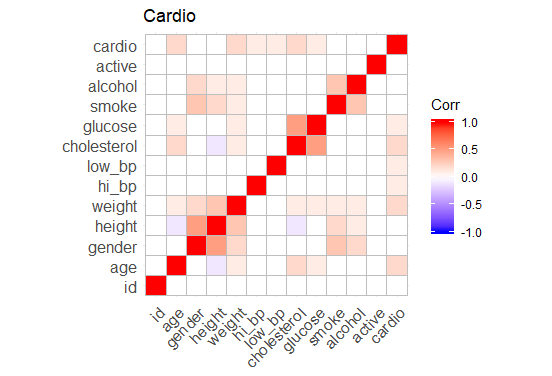


Figure 12: Correlation plot

From the correlation plot it is evident that glucose and age level share a strong relationship,

But are not highly correlated to the target class ‘cardio’.

The next part deals with creating more features using the existing features. The feature Body Mass Index ratio (BMI) can be calculated using the attributes ‘height’ and ‘weight’ featuring the patient.

Data in the dataset is now clean, but it needs to be standardized to get more accurate and precise results. Data can be scaled by using the scale () function in R. The scaling of the data needs to be conducted only on numerical data and leaving out the categorical data.

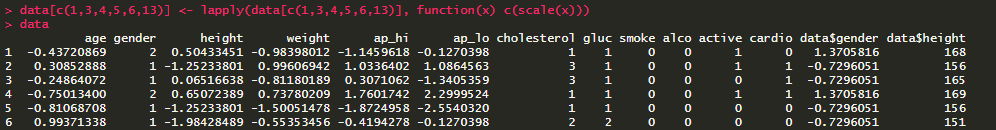


Figure 13: Scaled data

Now, the dataset is ready for modelling. The dataset has to be split in two parts namely for training and testing. The ratio picked for training is 75% of the dataset and training is 25% of the dataset. The data can be split as follows:

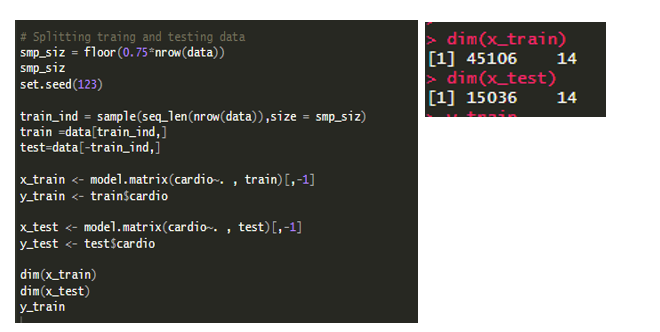


Figure 14: Splitting the dataset.

1. Model Implementation.

With a clean dataset, model can be implemented. The precision, accuracy, sensitivity and f1scores are calculated using the following formulas.

Actual

Predicted

|  |  |  |
| --- | --- | --- |
| **Confusion Matrix** | NEGATIVE | POSITIVE |
| NEGATIVE | True Negative | False Positive |
| POSITIVE | False Negative | True Positive |

The models taken for predictive analysis are as follows:

1. Naïve Bayes

Naïve Bayes algorithm can be implemented by calling the naiveBayes () function, after importing the library e1071.

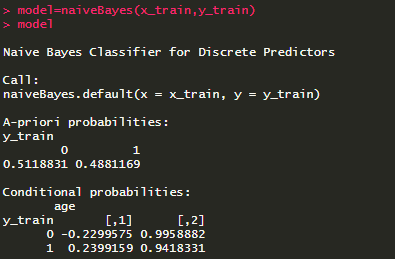


Figure 15: Naïve Bayes implementation

Now, testing against data which has not been seen by the model i.e. the test dataset and calculating the accuracy.

|  |  |  |  |
| --- | --- | --- | --- |
| Precision | Accuracy | Sensitivity | F1 score |
| 60% | 70% | 74% | 66% |

1. Random Forest

Random Forest can be implemented by using the randomForest () function and importing the randomForest and caTools library.

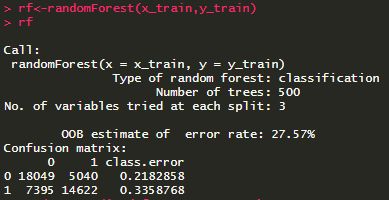


Figure 16: Random Forest implementation

The evaluation metrics are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| Precision | Accuracy | Sensitivity | F1 score |
| 61% | 73% | 75% | 67% |

1. Xgboost

The algorithm can be implemented as follows:

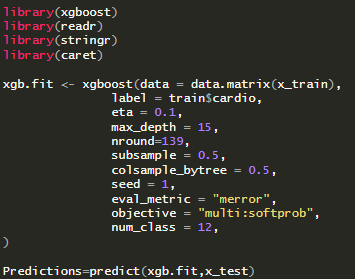


Figure 17: XGBoost implementation

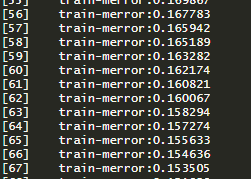


Figure 18: XgBoost output

The accuracy delivered by XgBoost is around 85%. Top sum it all up, below which shows the comparison for accuracy among the models.

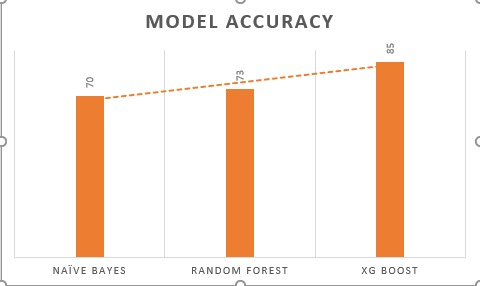


Figure 19: Comparison of accuracies

**Conclusion**

In this project, we started with data selection with respect to CVD. Further we have preprocessed data in order to be more precise and accurate while running through model. We started with creating data dictionary and storing the data into same. In next step we performed data cleaning by removing the unwanted, NA and duplicate data values, which finally resulted in removal of approximately 12,000 rows out of total 70,000. After performing data cleaning, we performed various method like descriptive and inferential statistics to check if the data set is standardized or not. Moving on we scaled the data by bring down the mean to 0 and standard deviation to 1 and normalization which resulted the range of values between 0 and 1.  As an add-on we have performed data featuring by adding additional feature of BMI in order to enhance the accuracy of our models. For predicting the presence or absence of CVD we have implemented 3 different models namely Naïve Bayes, Random forest and XGBoost. We were able to achieve 70% of accuracy after performing naïve Bayes and 73% with Random Forest which gave the output in the form of confusion matrix. After implementing and running 3rd model which is XGBoost we were able to achieve the maximum accuracy of 80%. Finally, on an average we have implemented our project with average accuracy of 70%. This project can be used to provide early notifications based on the data collected from patients and other monitoring devices used for medical examinations. By using the model, we can detect the presence of CVD among the patients in less time and this can lead to improved medical support and further required treatment. For future scope our model results can be improved by using cross-validation for testing model and performing hyper tuning the parameter of models by using different methods like grid random search.

**References**

Srivastava, T., & Srivastava, T. (2019, June 25). Use XGBoost in R: A Complete tutorial with easy steps. Retrieved from <https://www.analyticsvidhya.com/blog/2016/01/xgboost-algorithm-easy-steps/>

Maklin, C. (2019, July 30). Random Forest In R. Retrieved from <https://towardsdatascience.com/random-forest-in-r-f66adf80ec9>

Datasets. (n.d.). Retrieved from <https://www.kaggle.com/datasets>

Naive Bayes Classifiers. (2019, January 14). Retrieved from <https://www.geeksforgeeks.org/naive-bayes-classifiers/>