* Classification—Deciding if something belongs to one category or another
* Finding relations—Finding correlations or potential causes of effects seen in the data
* Characterization—Very general plotting and report generation from data
* Distribution – Too narrow Vs. Too wide, good predictor if wide, varies. You should rely on information about the problem domain to judge if the data range is narrow, but a rough rule of thumb is the ratio of the standard deviation to the mean. If that ratio is very small, then the data isn’t varying much.

**Exploratory Analysis**

**Feature Selection**

Most of the feature selection methods can be categorized in two categories, which are, wrapper methods and filter methods [1]. Wrapper methods evaluate a model by plugging different sets of features in order to find out the optimal subset for which the performance is maximum. Wrapper methods are, indeed, search algorithms that take features as inputs and output the optimal subset of features. There are various wrapper methods available, for example, recursive feature elimination, genetic algorithms, simulated annealing, etc. On the other hand, filter methods find out the relevance of the features before modelling the data and models the data subsequently only with important features. In other words, only features with important relationship are retained for training.

However, there are both advantages and disadvantages of both methods. Filter methods are less computationally demanding task than its counterpart, but it does not directly justify the performance of the model. As this method evaluates each feature separately, important interactions between features is not quantified. In contrast, wrapper methods are computationally intensive tasks, but there is no risk of overfitting.

In this study, a wrapper method Recursive Feature Elimination (RFE) was applied because during the exploratory analysis no feature was found to have significant predictive power over the target. So, we trained each model with different subsets of feature to identify the most effective ones.

**Model Selection**

Our aim is to predict whether a person is at risk of developing cardiovascular disease, which is a classification problem. Because we are classifying patients into two groups, that is positive and negative, this problem is a binary classification problem. There are many algorithms available for binary classification problems. For example, Naïve Bayes, Decision Tree, Logistic Regression, Support Vector Machine, etc. For this project, we chose to use all of the aforementioned algorithms to select the best one.

A. Logistic Regression

B. Decision Tree

C. Support Vector Machine

SVM is a non-parametric model and makes less assumptions about the data. For this reason, even if the real-world data do not follow the training data distributions in future, it will still give a fair result.

D. Naïve Bayes

In contrast, Naïve Bayes is a parametric model and has several assumptions about the data, for example, it assumes that the features are independent of each other.

**Model Evaluation**

For model evaluation, we, first, established the null model, which is the lower bound of the model. As it is a classification problem, we selected null model to be the most common of all target classes. Then we calculated the Bayes rate which is the upper bound of the model. We also constructed the best single variable model possible and compared it against our final models. For performance measurement, we constructed confusion matrices and calculated accuracy, precision, recall, f1 score, specificity, and sensitivity for all the models. However, in this case, misclassification of someone who is not at risk of developing disease into at risk or positive would not be much of a problem because taking preventive measures are not discourageable. In contrast, if we classify somone who is indeed at risk into negative, it would be a problem. So, we wanted the precision or sensitivity to be as high as possible.

**Model Validation**

Standard procedures were maintained for model validation. The data was split into three groups for training, testing, and calibration. K-fold cross validation was applied during modeling. Significance tests were performed on the models and their p-values were compared.

**Documentation**

Steps were taken to ensure proper documentation and flawless reproducibility of the project from the start. For documentation, the codes were written in R Markdown file using the knitr package, which is basically a stream of texts and codes. This ensured that the documentation was synchronized with codes, results were in sync with data, and it provided efficient means of collaboration between the project contributors. The project history was tracked from the beginning using Git, the most popular version control system, and GitHub, the most popular code sharing platform based on Git. All the data were saved in non-proprietary format to ensure reproducibility.

**Reference**

[1] G. John, R. Kohavi, and K. Pfleger, “IrreleJohn, G., Kohavi, R., & Pfleger, K. (1994). Irrelevant Features and the Subset Selection Problem. Icml, 121–129. Retrieved from http://machine-learning.martinsewell.com/feature-selection/JohnKohaviPfleger1994.pdfvant Features and the Subset Selectio,” *Icml*, pp. 121–129, 1994.