

**Cardiovascular Risk Prediction**

**Debanjan Ganguly**

**Data Science Trainee**

**AlmaBetter, Bengaluru**

**CONTENTS**

**Page**

* **Introduction - - - 3**
* **Problem Statement - - - 3**
* **Dataset Analysis - - - 4**
* **Exploratory Data Analysis - - - 5**
* **Text Preprocessing - - - 8**
* **Text Normalization - - - 9**
* **Literature Survey - - - 10**
* **Experimental Analysis - - - 15**
* **Model Selection - - - 17**
* **Limitation - - - 20**
* **Scope of Improvement - - - 20**
* **Conclusion - - - 20**
* **References - - - 21**

**Introduction:**

Heart disease is the leading cause of death worldwide, accounting for one third of deaths in 2019. Heart disease cases nearly doubled over the period, from 271 million in 1990 to 523 million in 2019, and the number of heart disease deaths rose from 12.1 million to 18.6 million. Over three quarters of these deaths took place in low- and middle-income countries.

Of all heart diseases, coronary heart disease (aka heart attack) is by far the most common and the most fatal. The silver lining is that heart attacks are highly preventable and simple lifestyle modifications (such as reducing alcohol and tobacco use; eating healthily and exercising) coupled with early treatment greatly improves its prognosis. It is, however, difficult to identify high risk patients because of the multi-factorial nature of several contributory risk factors such as diabetes, high blood pressure, high cholesterol, et cetera. This is where machine learning and data mining come to the rescue.

This risk prediction analysis focuses on the supervised ML-based approach, which is computationally fast and exhibits promising classification results. Missing values have been handled, basic statistical analysis has been performed, resampled the imbalanced dataset by oversampling the positive cases, scaled the data using normalization, built six different machine learning models using pipeline, also the hyperparameters of these are tuned to best. Random forest model is selected to be the best amongst our trained models.

**Problem Statement:**

The dataset is from an ongoing cardiovascular study on residents of the town of Framingham, Massachusetts. The classification goal is to predict whether the patient has a 10-year risk of future coronary heart disease (CHD).

**Dataset Analysis:**

Dataset contains over 3000 records and 15 attributes. Each attribute is a potential risk factor. There are both demographic, behavioral, and medical risk factors.

Demographic:

* Sex: male or female ("M" or "F")
* Age: Age of the patient; (Continuous - Although the recorded ages have been truncated to whole numbers, the concept of age is continuous)

Education: No further information provided

Behavioral:

* is\_smoking: whether or not the patient is a current smoker ("YES" or "NO")
* Cigs Per Day: the number of cigarettes that the person smoked on average in one day. (Can be considered continuous as one can have any number of cigarettes, even half a cigarette.)

Medical (history):

* BP Meds: whether or not the patient was on blood pressure medication (Nominal)
* Prevalent Stroke: whether or not the patient had previously had a stroke (Nominal)
* Prevalent Hyp: whether or not the patient was hypertensive (Nominal)
* Diabetes: whether or not the patient had diabetes (Nominal) Medical(current)
* Tot Chol: total cholesterol level (Continuous)
* Sys BP: systolic blood pressure (Continuous)
* Dia BP: diastolic blood pressure (Continuous)
* BMI: Body Mass Index (Continuous)
* Heart Rate: heart rate (Continuous - In medical research, variables such as heart rate though in fact discrete, yet are considered continuous because of large number of possible values.)
* Glucose: glucose level (Continuous) Predict variable (desired target)
* 10-year risk of coronary heart disease CHD (binary: “1”, means “Yes”, “0” means “No”)

**Exploratory Data Analysis:**

Exploratory Data Analysis or EDA perform a key role to become acquainted with data to drive intuition and begin to formulate testable hypothesis. This process typically makes use of descriptive statistics and visualizations.

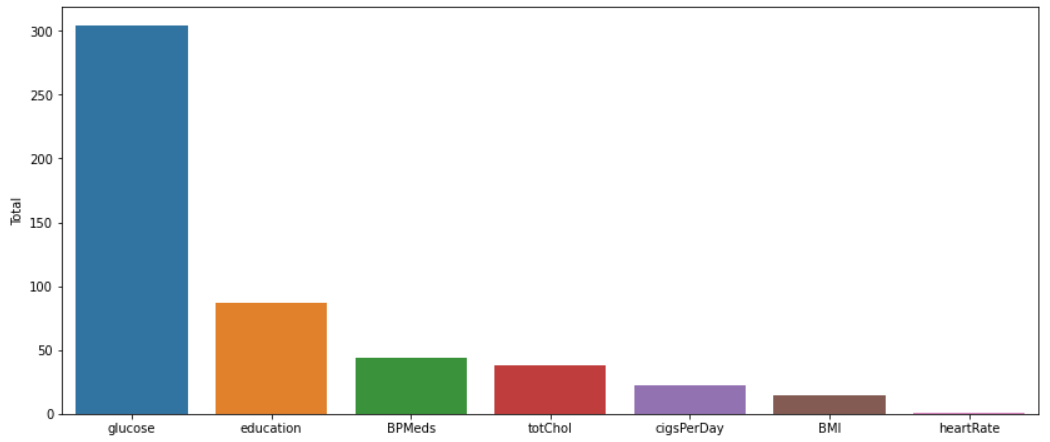
Out of 15 features, 7 features contain missing or null values.

Fig 1. Missing values count plot

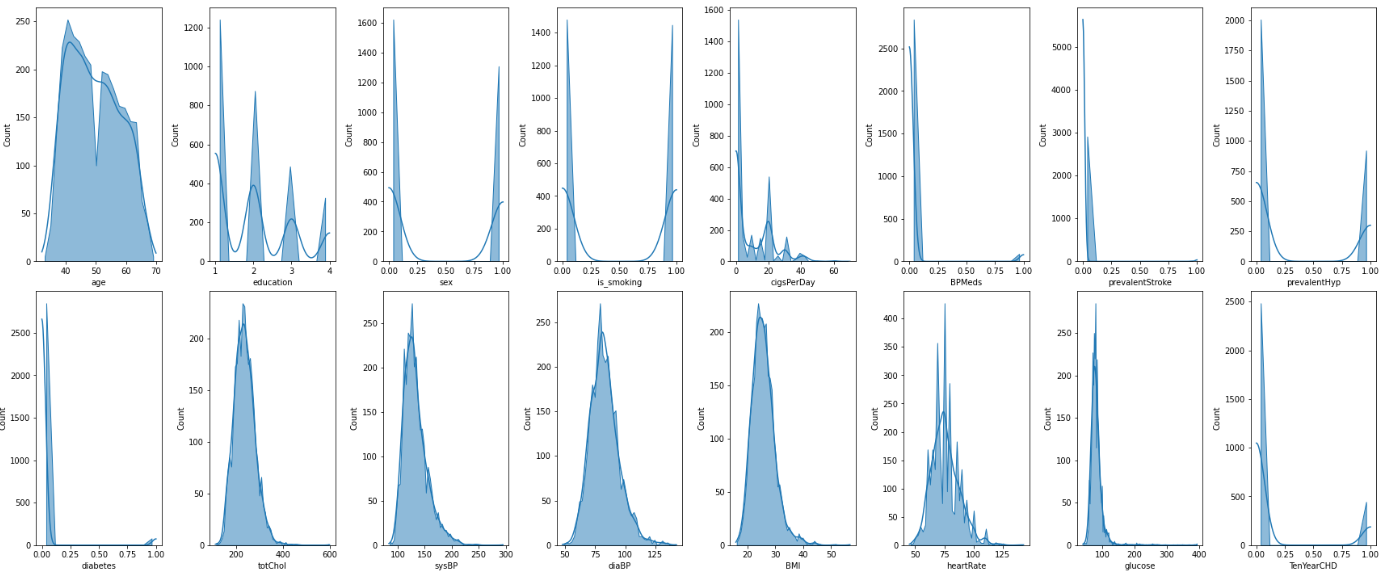
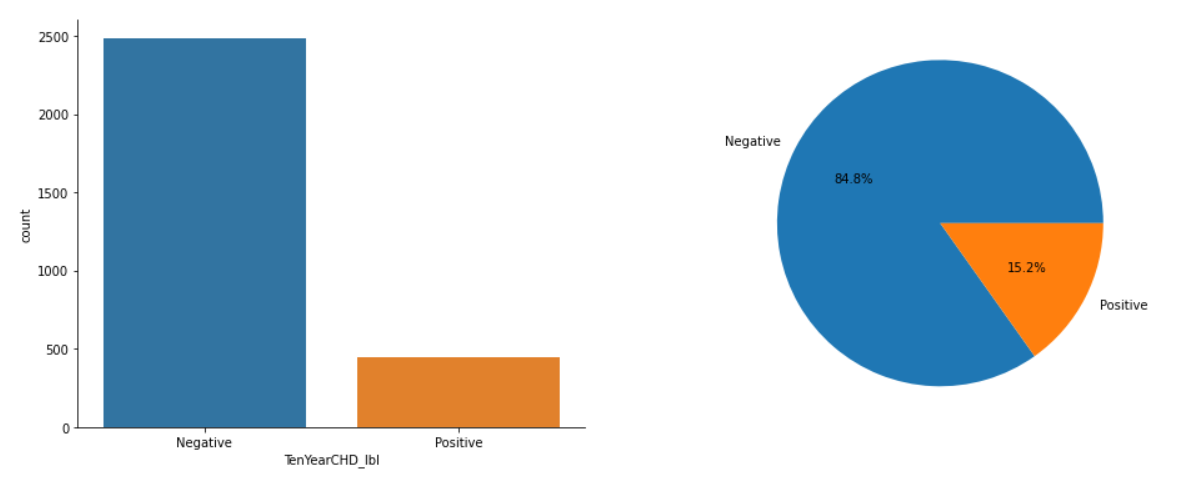
 Total percentage of missing data is 15%, of which the highest is for glucose that accounts for 9% of missing data.

Fig 2. Distribution of features

totChol, sysBP, diaBP, BMI has an almost normal but little right skewed distribution. BPMeds, prevalentStroke and diabetes is very poorly balanced, so is our target variable too. So it is an imbalanced dataset.

Fig 3. Proportion of TenYearCHD risk

Total Negative labeled count is 2483 and total positive labeled count is 444. As in, the number of negative cases outweigh the number of positive cases. This would lead to class imbalance problem while fitting our models. Therefore, this problem needs to be addressed and taken care of.

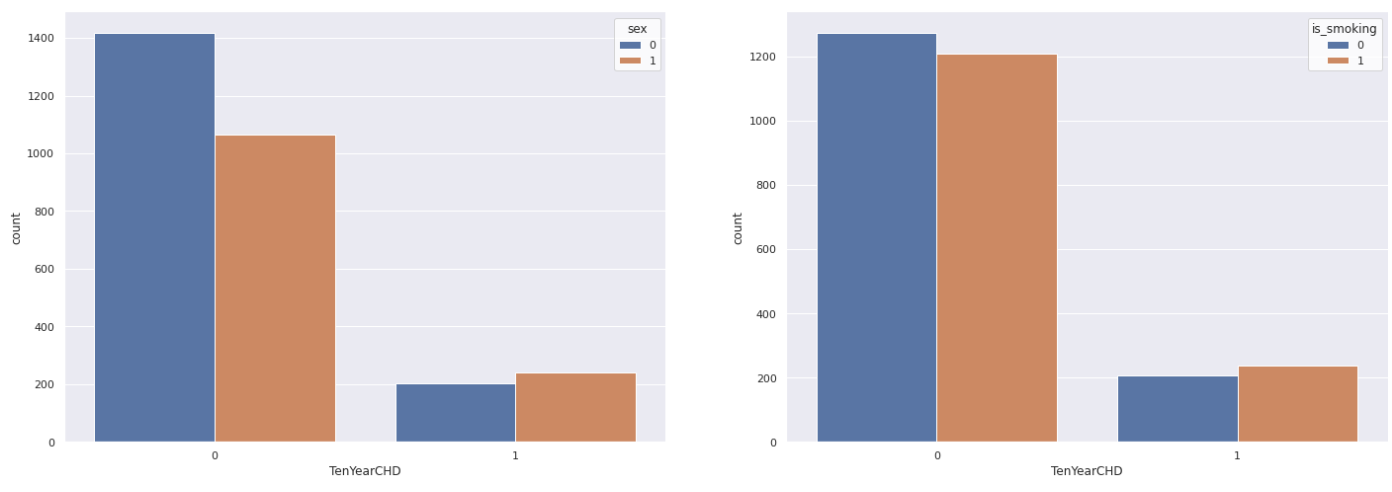
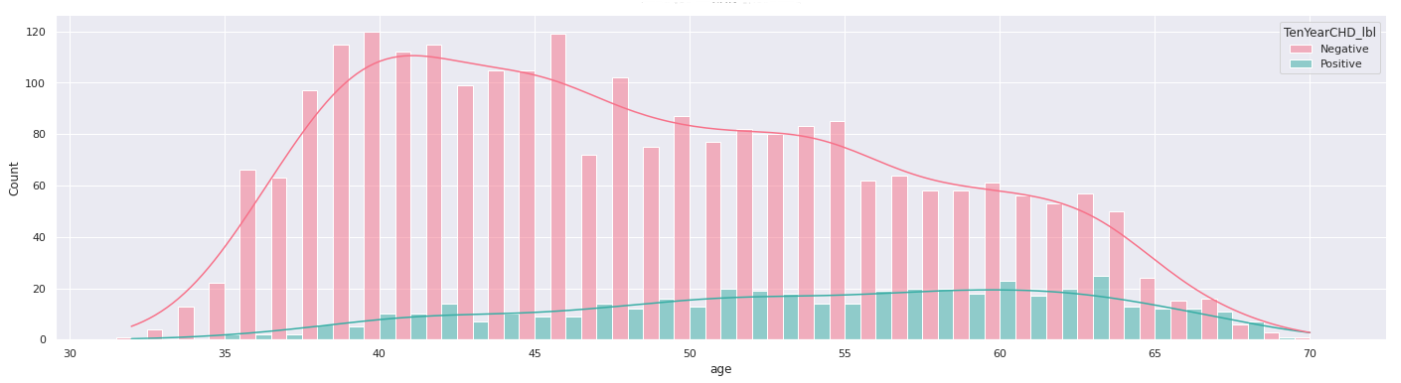


Fig 4. Chances of CHD over gender and smoking habits

Males has quite high risk of CHD than females, the similar observation found for smoking habits too. Also, males are normally more addicted to smoke then females, this clears the equation. Now, we will take age into count, and observe how age affects whether in smoking habits or chances of heart attack or CHD.

Fig 5. Chances of CHD over ages

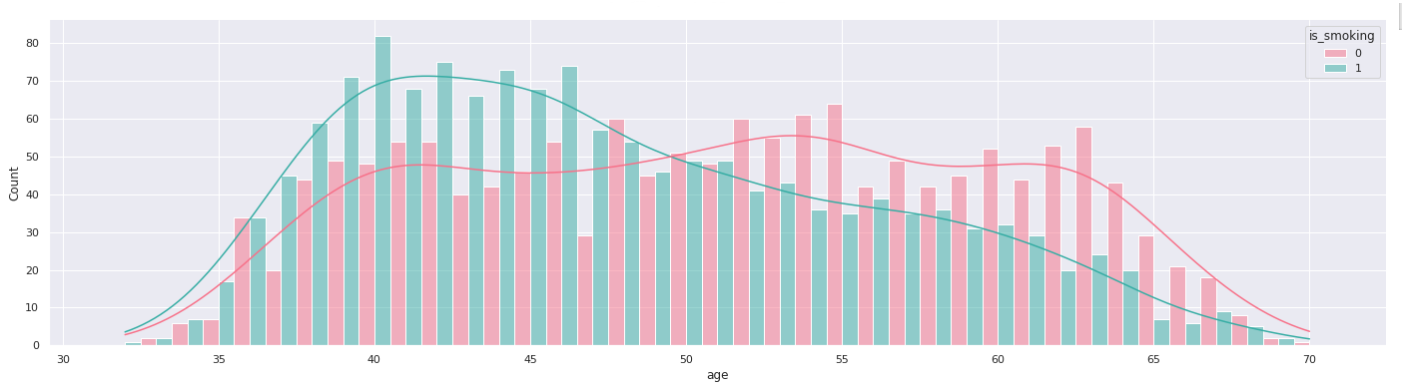
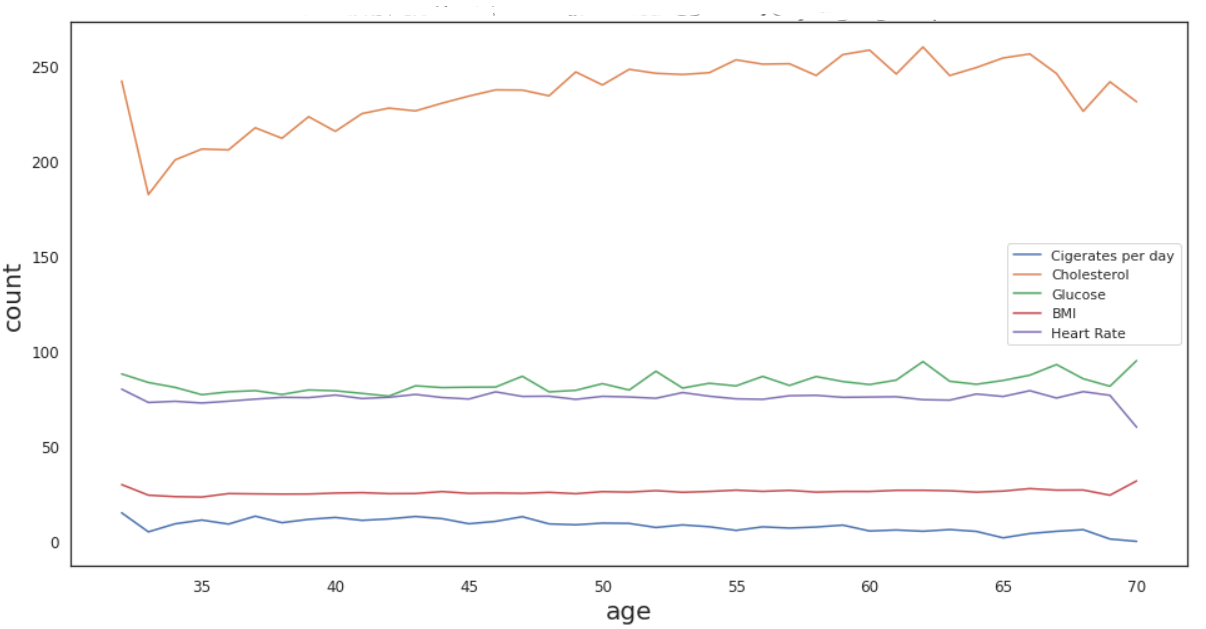
 The chances for CHD are more than avg within age range of 51 to 64 in a slowly increasing manner.

Fig 6. Smokers count with respect to ages

As per our dataset, mid age groups ranging from 37 to 46 has more smokers. As the age group passes 50, numbers of non-smokers gradually started increasing.

Fig 7. Distribution of Medical features in every age group

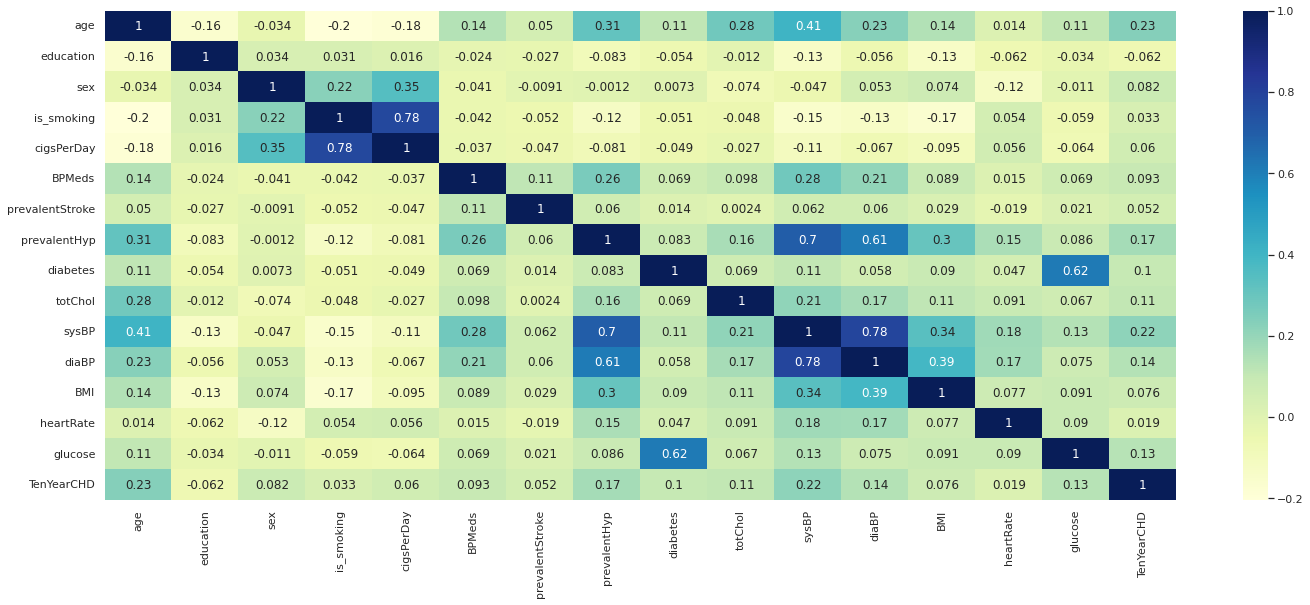
 There is some minor relation between cholesterol and glucose. Heart Rate and cigarettes per day has a strong relation. Glucose has an inverse relation with heart rate in later old age. Cigarettes per day has a fairly parallel relationship with age.

Fig 8. Heatmap of Correlation Matrix

There are no features with more than 0.3 correlation with the Ten-year risk of developing CHD and this shows that the features a poor predictor. The correlation coefficient between education and target variable TenYearCHD is very low and actually negative.

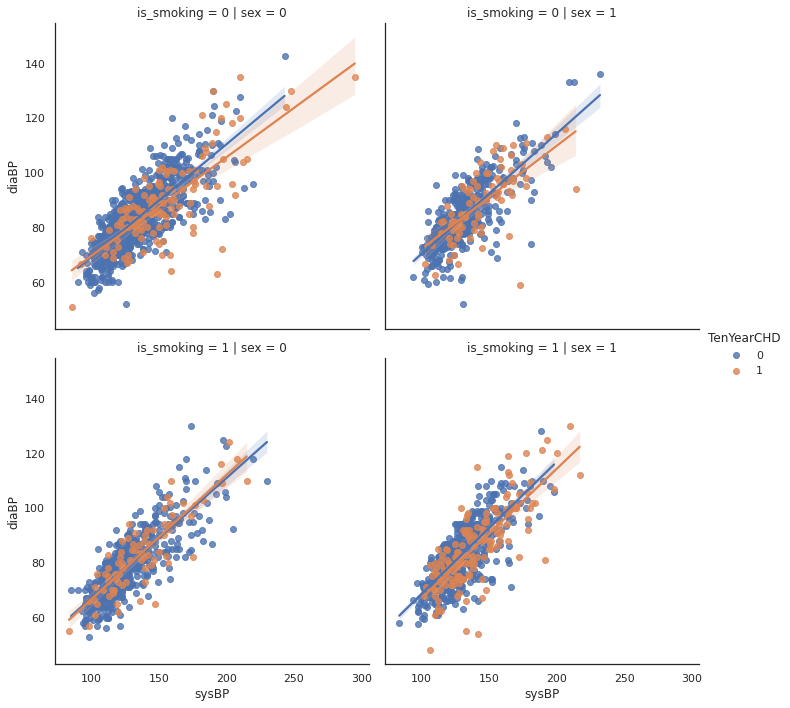


Fig 9. sysBP & diaBP based on gender and smoking habits

The above graph plots the relationship between systolic blood pressure and diastolic blood pressure for patients based on their gender and whether they are current smokers or not and plots the best fit line

**Feature Selection:**

Two different methods have been tried to select best features BorutaPy and SelectKBest.

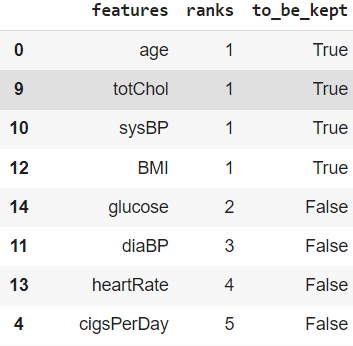
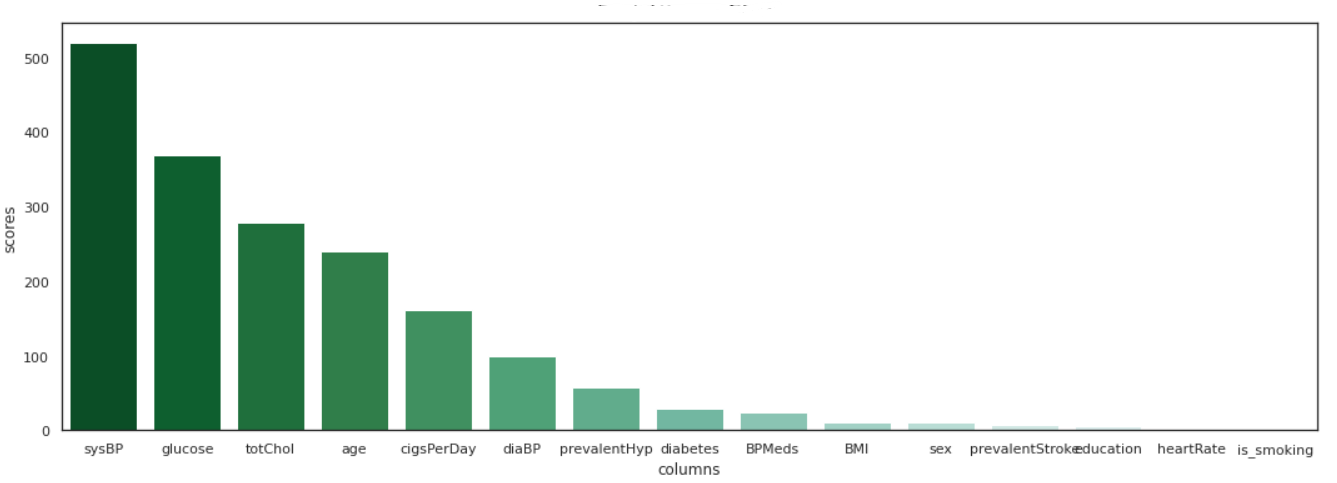
 **Boruta** is a random forest-based method, so it works for tree models like Random Forest or XGBoost, but is also valid with other classification models like Logistic Regression or SVM. Boruta iteratively removes features that are statistically less relevant than a random probe (artificial noise variables introduced by the Boruta algorithm). In each iteration, rejected variables are removed from consideration in the next iteration. It generally ends up with a good global optimization for feature selection. Initially, Boruta recommends to keep only 4 features but we’re considering features which has a rank less than or equal to 5. Also, we’ve made some trial and error manually before selecting these 8 features.

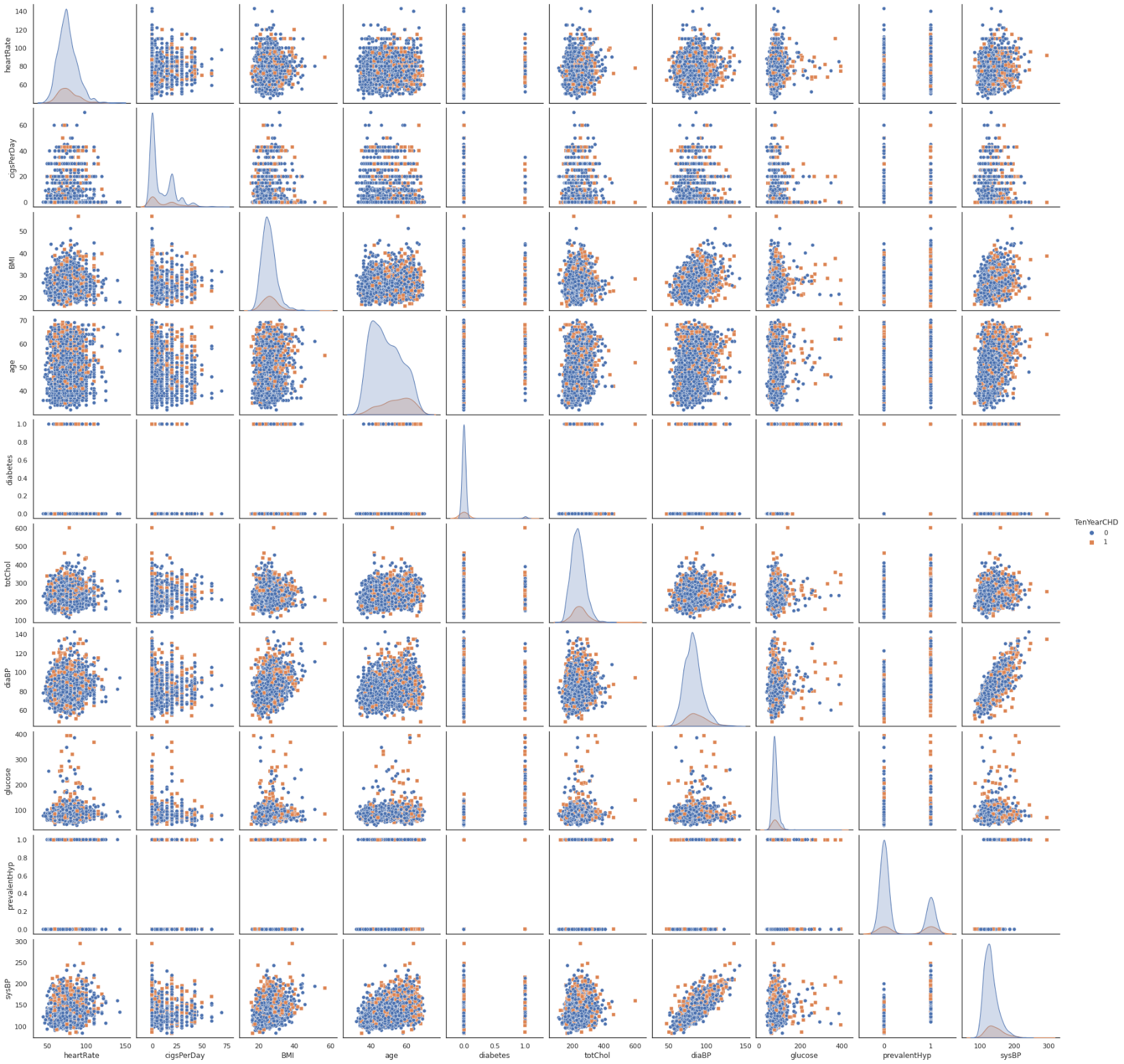
Fig 10. Recommended features by Boruta

Univariate Feature Selection is a feature selection method based on the univariate statistical test, e.g.: chi2, Pearson-correlation, and many more. The premise with **SelectKBest** is combining the univariate statistical test with selecting the K-number of features based on the statistical result between the X and y. The score function we’ve used is chi2. Below is the scores allotted to features,

Fig 11. Feature weightage score provided by SelectKBest

Similarly, from SelectKBest method, we’ve chosen top 8 features i.e., to diabetes.

Later on, we’ve combined selected features obtained from both methods, and chosen the common from both of these. Below is the pair plot for all combined features given by both methods.

Fig 12. Pairplot for selected combined features

From the above pairplot, we found no such pattern that splits the data well. The common selected features that we’ve finalized are sysBP, glucose, totChol, age, cigsPerDay, diaBP.

**Literature Survey:**

Cardiovascular risk factors are associated with each other and have a multiplicative rather than additive effect on health. Even slight elevations in two or more risk factors noticeably increase the risk for CHD (Smith, 2007). Thus, assessment of the overall effect of multiple risk factors on the development of CHD is crucial for accurately identifying peoples at high CHD risk. It is estimated that someone has a heart attack every 40 seconds.

It is, however, difficult to identify high risk patients because of the multi-factorial nature of several contributory risk factors such as diabetes, high blood pressure, high cholesterol, et cetera. This is where machine learning and data mining come to the rescue. Doctors and scientists alike have turned to machine learning (ML) techniques to develop screening tools and this is because of their superiority in pattern recognition and classification as compared to other traditional statistical approaches.

Being CHD a highly sensitive field, we need to take care of recall more sensitively. In our cases false negatives are costlier than false positives. Below is the algorithms that we’ve used while training and evaluating our models.

* **Naïve Bayes:** Naive Bayes falls under the umbrella of supervised machine learning algorithms that are primarily used for classification. This algorithm is called naïve because the classifier assumes that the input features that go into the model are independent of each other. Hence, changing one input feature won’t affect any of the others. It's therefore naive in the sense that this assumption may or may not be true, and it most probably isn't. Bayes Rule revolves around the concept of deriving a hypothesis (H) from the given evidence (E). It relates two notions: the probability of the hypothesis before getting the evidence, P(H), and the probability of the hypothesis after getting the evidence, P(H|E). In general, it’s given by the following equation:

**P(H|E) = (P(E|H) \* P(H)) / P(E)**

From equations,

**P(H|E)** How often H happens given that E happens

**P(E|H)** How often E happens given that H happens

**P(H)** How likely H happens on its own

**P(E)**  How likely E happens on its own

In simple terms, it provides a way to calculate the probability of a hypothesis given the evidence. The Bayes Rule provides the formula to compute the probability of output (Y) given the input (X). In real-world problems, unlike the hypothetical assumption of having a single input feature, we have multiple X variables. When we can assume the features are independent of each other, we extend the Bayes Rule to what is called Naive Bayes. Consider a case where there are multiple inputs (X1, X2, X3 ... Xn). We predict the outcome (Y) using the Naive Bayes equation as follows:

**P(Y=k | X1...Xn) = ( P(X1 | Y=k) \* P(X2 | Y=k) \* P(X3 | Y=k) \* ....\* P(Xn | Y=k) ) \* P(Y=k) / P(X1)\*P(X2)\*P(X3)\*P(Xn)**

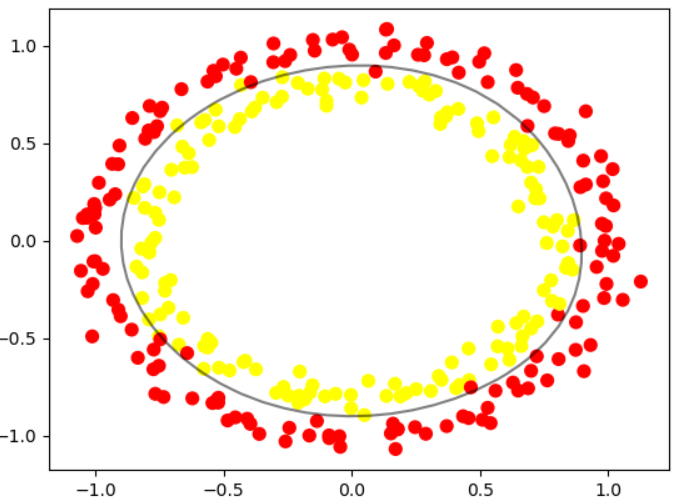
**P(Y=k | X1...Xn)** is called the Posterior Probability, which is the probability of an outcome given the evidence.

**P(X1 | Y=k) \* P(X2 | Y=k) \* ... P(Xn | Y=k)** is the probability of the likelihood of evidence.

**P(Y=k)** is the Prior Probability.

**P(X1)\*P(X2)\*P(Xn)** is the probability of the evidence.

Multinomial Naïve Bayes is used to predict multi-class classification. It considers a feature vector where a given term represents the number of times it appears or very often i.e., frequency. It has low computation cost, can work effectively on relatively large datasets. Also, for small sample sizes, it can outperform the most powerful alternatives. But it is very difficult to get the set of independent predictors for developing model.

* **Support Vector Machine:** Support vector machines are a set of supervised learning methods used for classification, regression, and outliers’ detection. All of these are common tasks in machine learning. A simple linear SVM classifier works by making a straight line between two classes. That means all of the data points on one side of the line will represent a category and the data points on the other side of the line will be put into a different category. This means there can be an infinite number of lines to choose from. The linear SVM algorithm better than some of the other algorithms, like k-nearest neighbors, is that it chooses the best line to classify your data points. It chooses the line that separates the data and is the furthest away from the closet data points as possible.

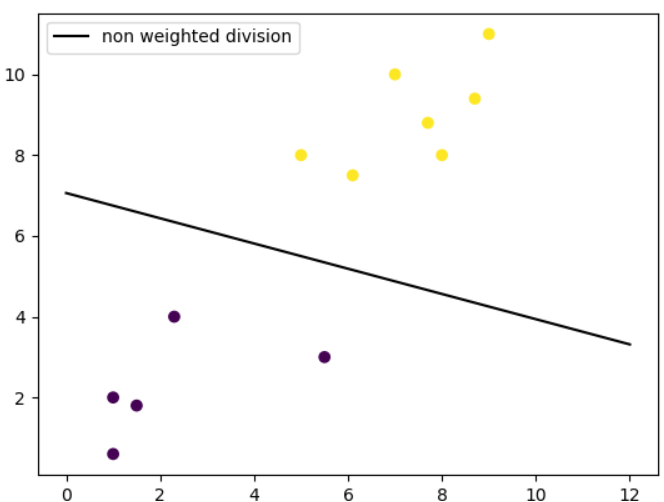


Fig 13. Linear SVM Fig 14. Non-linear SVM with RVF Kernel

SVM is Effective on datasets with multiple features or in cases where number of features is greater than number of data points

Train and test set has been standardized using StandardScaler from sklearn, and target variable has been log transformed. Let us plot the graphs of train and test set. It uses a subset of training points in the decision function called support vectors which makes it memory efficient. It has different kernel functions can be specified for the decision function. But SVMs don't directly provide probability estimates. Those are calculated using an expensive five-fold cross-validation.

Linear Kernel is commonly recommended for text classification because most of these types of classification problems are linearly separable. The linear kernel works really well when there are a lot of features, and text classification problems have a lot of features. Linear kernel functions are faster than most of the others and you have fewer parameters to optimize. The LinearSVC module from sklearn library provides support for linear support vector machine. The function for linear kernel is as below,

**f(X) = wT \* X + b**

* **Decision Tree:** A decision tree is a tree-like graph with nodes representing the place where we pick an attribute and ask a question; edges represent the answers the to the question; and the leaves represent the actual output or class label. They are used in non-linear decision making with simple linear decision surface. Decision trees classify the examples by sorting them down the tree from the root to some leaf node, with the leaf node providing the classification to the example. Each node in the tree acts as a test case for some attribute, and each edge descending from that node corresponds to one of the possible answers to the test case. This process is recursive in nature and is repeated for every subtree rooted at the new nodes. A general algorithm for a decision tree can be described as follows:
* Pick the best attribute/feature. The best attribute is one which best splits or separates the data.
* Ask the relevant question.
* Follow the answer path.
* Go to step 1 until you arrive to the answer.

The best split is one which separates two different labels into two sets. The best attribute is the one with the highest information gain. Information gain is a statistical property that measures how well a given attribute separates the training examples according to their target classification. To define information gain precisely, we need to define a measure commonly used in information theory called entropy that measures the level of impurity in a group of examples. It is defined as,



 Now, given entropy as a measure of the impurity in a sample of training examples, information gain can be defined as a measure of the effectiveness of an attribute in classifying the training data. Information gain, Gain (S, A) of an attribute A, relative to a sample of examples S, is defined as:

Decision Tree is easy to use and understand, can handle both categorical and numerical data, is resistant to outliers, hence require little data preprocessing. It can be used to build larger classifiers by using ensemble techniques. But it is very prone to overfitting and hyper parameter tuning is very sensitive to metrics. In case of high imbalanced dataset, it can create biased trees which is also a form of overfitting.

* **Logistic Regression:** Logistic Regression is the appropriate regression analysis to conduct when the dependent variable is dichotomous (binary). Like all regression analyses, the logistic regression is a predictive analysis. Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

Fig 15. Logistics Regression with sigmoid function

A logistic regression model predicts a dependent data variable by analyzing the relationship between one or more existing independent variables. For example, a logistic regression could be used to predict whether a political candidate will win or lose an election or whether a high school student will be admitted or not to a particular college. These binary outcomes allow straightforward decisions between two alternatives. It assumes that,

* The dependent variable is binary or dichotomous—i.e. It fits into one of two clear-cut categories.
* There should be no, or very little, multicollinearity between the predictor variables—in other words, the predictor variables (or the independent variables) should be independent of each other.
* The independent variables should be linearly related to the log odds.
* Logistic regression requires fairly large sample sizes—the larger the sample size, the more reliable (and powerful) you can expect the results of your analysis to be.
* **Random Forest:** Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.



Fig 16. Mechanism of Random Forest Algorithm

Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. The assumptions on which random forest works are,

* There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.
* The predictions from each tree must have very low correlations.

Random Forest is capable of performing both Classification and Regression tasks. It is capable of handling large datasets with high dimensionality. It enhances the accuracy of the model and prevents the overfitting issue.

* **K Nearest Neighbour:** K-Nearest Neighbor is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. K-NN algorithm stores all the available data and classifies a new data point based on the similarity. K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset. KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.



Fig 16. Mechanism of K Nearest Neighbour

**Experimental Analysis:**

In our analysis NumPy, pandas, scikit-learn, imblearn, matplotlib, seaborn, shap, eli5, boruta libraries were used for experiments. Matplotlib, seaborn is used to plot all the graphs. NumPy, pandas are used to process and store dataset. All the machine learning algorithms are from scikit-learn library, the **StandardScaler** i.e., used to normalize data are also from scikit-learn. **Boruta** is used for selection of features. Shap and eli5 are the other two libraries used for model explainability.

Initially, missing values were removed from dataset. Then, data is split with a 0.2 ratio of test size. We’ve used **pipelines** to train our models with cross validation. Pipeline has two steps, first one is to normalize the data, second one is to train our model using specified algorithms. Each algorithm was used two times, one with default imbalanced dataset and on next turn, with oversampled dataset of positive class (minority class) using **SMOTE**. This is done to compare the metrics of two different scenarios. Within pipelines, the training data is again split due to cross validation. These is done using **ShuffleSplit**, which will randomly sample entire training dataset during each iteration to divide it further into train and test, which are to be used for cross validation using **HalvingGridSearch** supported by scikit-learn. Halving grid search over specified parameter values with successive halving. The returned metrics is almost similar to grid search but with less time and computation power. The search strategy starts evaluating all the candidates with a small number of resources and iteratively selects the best candidates, using more and more resources, thus results in most similar result with impressive lesser time than its closest GridSearchCV.

Different metrics were used to evaluate the performance of our pipelines.

* **Accuracy:** Accuracy is the quintessential classification metric. Accuracy is the proportion of true results among the total number of cases examined. It is easily suited for binary as well as a multiclass classification problem. Accuracy is a valid choice of evaluation for classification problems which are well balanced and not skewed or no/less class imbalance.

**Accuracy = (TP + TN) / (TP + TN + FP + FN)**

* **Precision (Macro):** Precision answers what proportion of predicted positives is truly positives. Precision is a valid choice of evaluation metric when we want to be very sure of our prediction.

**Precision Macro = (Sum of Precision for each individual class) / (No. of Classes)**

* **Recall (Macro):** Recall answers what proportion of actual Positives is correctly classified. Recall is a valid choice of evaluation metric when we want to capture as many positives as possible.

**Recall Macro = (Sum of Recall for each individual class) / (No. of Classes)**

* **F1 Score:** The F1 score is a number between 0 and 1 and is the harmonic mean of precision and recall. F1 score sort of maintains a balance between the precision and recall for classifier. If precision is low, the F1 is low and if the recall is low again F1 score is low. The F1 score manages the tradeoff.

**F1 Score = 2 \* (Precision \* Recall) / (Precision + Recall)**

* **AUC:** AUC is the area under the ROC curve. AUC ROC indicates how well the probabilities from the positive classes are separated from the negative classes. AUC is scale-invariant. It measures how well predictions are ranked, rather than their absolute values.

As specified earlier, pipelines were given two different datasets. Let us look through those briefly.

1. **Pipeline created for with no resampling done:**

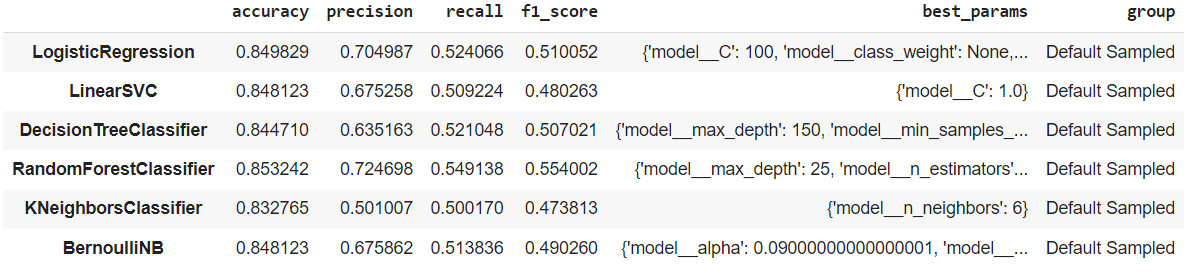
Steps involved are, normalization (StandardScaler) -> cross validation (shuffle split with 10 folds, halving grid search cv). 6 different algorithms have been evaluated in this pipeline. The results of classification are described as below,

Fig 17. Metrics score for Imbalanced dataset

The performance is low for every algorithm. Random forest classifier tops the list with a f1 score of 0.554 and recall score of 0.549. In the second position, it is logistic regression with f1 score of 0.51.

1. **Pipeline created with SMOTE:**

It is not advised to train a classifier on an imbalanced data set as it may be biased towards one class thus achieve high accuracy but have poor sensitivity or specificity. To address this problem, we’ve balanced the data using **SMOTE** or **Synthetic Minority Oversampling Technique**.

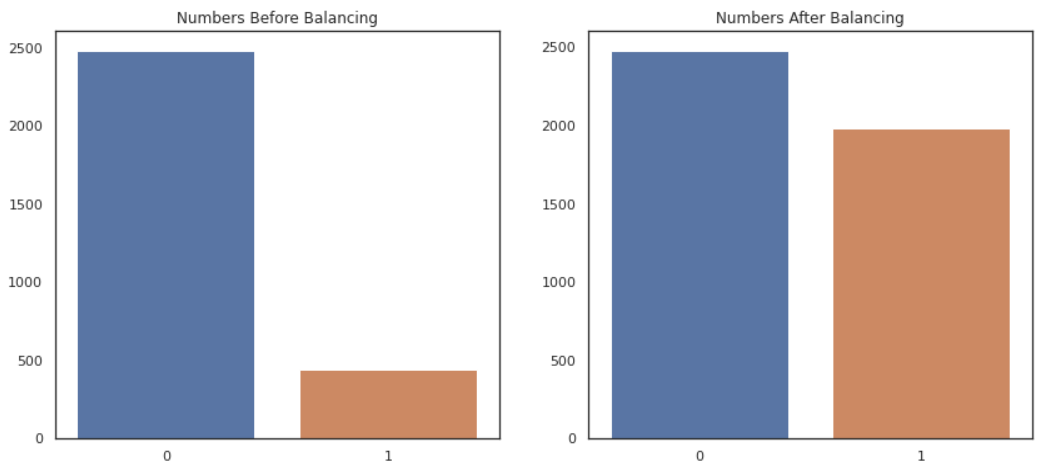
 SMOTE first selects a minority class instance x¹ at random and finds its k nearest minority class neighbors. The synthetic instance is then created by choosing one of the k nearest neighbors x² at random and connecting x¹ and x² to form a line segment in the feature space. The synthetic instances are generated as a convex combination of the two chosen instances x² and x¹. This procedure suggests first using random under sampling to trim the number of examples in the majority class, then use SMOTE to oversample the minority class to balance the class distribution. We’ve used SMOTE to oversample with 0.8 ratio. In default dataset, the number of positive records is **444**, after oversampling, number increased to **1986**.

Fig 18. Comparison of class counts in dataset

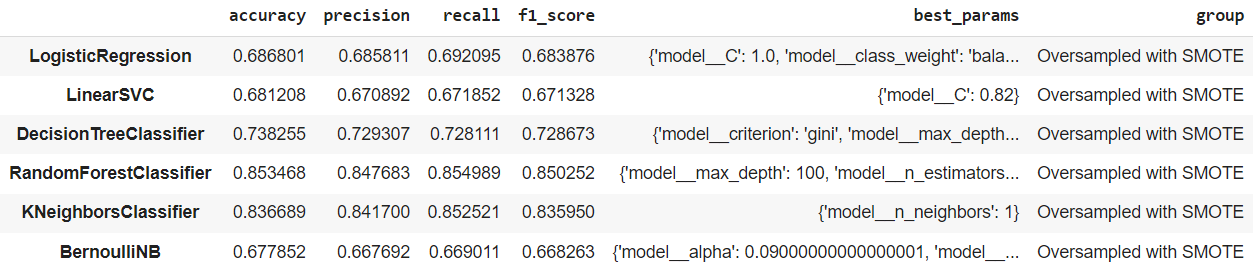
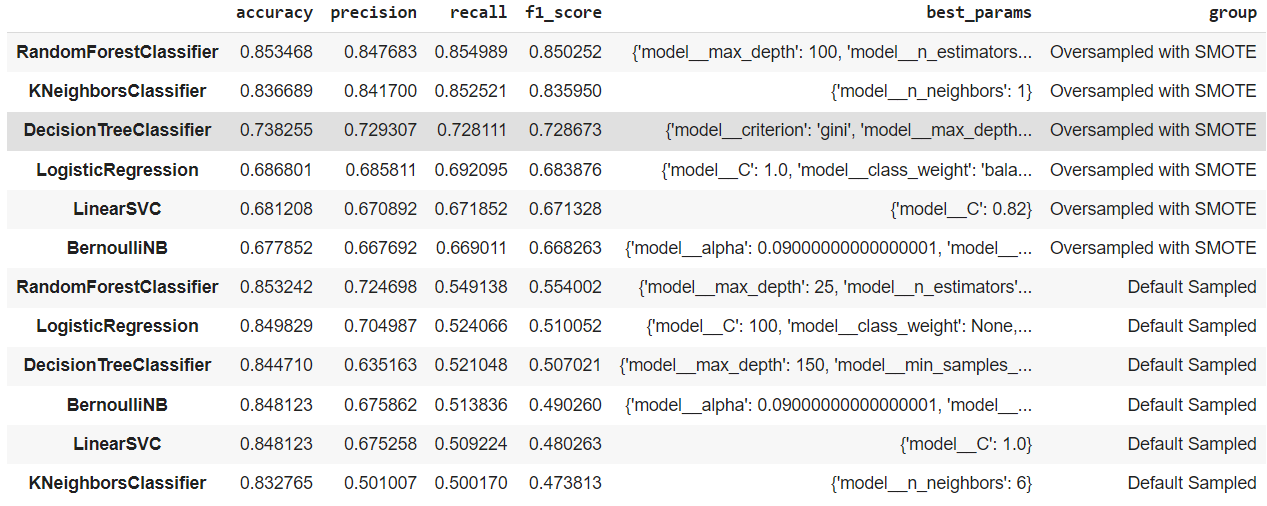
****After balancing, the pipeline steps are similar with same settings. The results of classification are described as be

Fig 19. Metrics score for over sampled dataset

We’ve observed an immense increase in performance for all algos’ after oversampling. Still, the random forest achieved highest performance with a f1 score of 0.85 and recall of 0.855. In the second spots, its k nearest neighbor with a f1 score of 0.836 and a recall of 0.8525.

We’ve trained a total of 12 models includes no resampled and over sampled dataset. Below is the list of models’ performance ascending by their performance on the scale of f1 score.

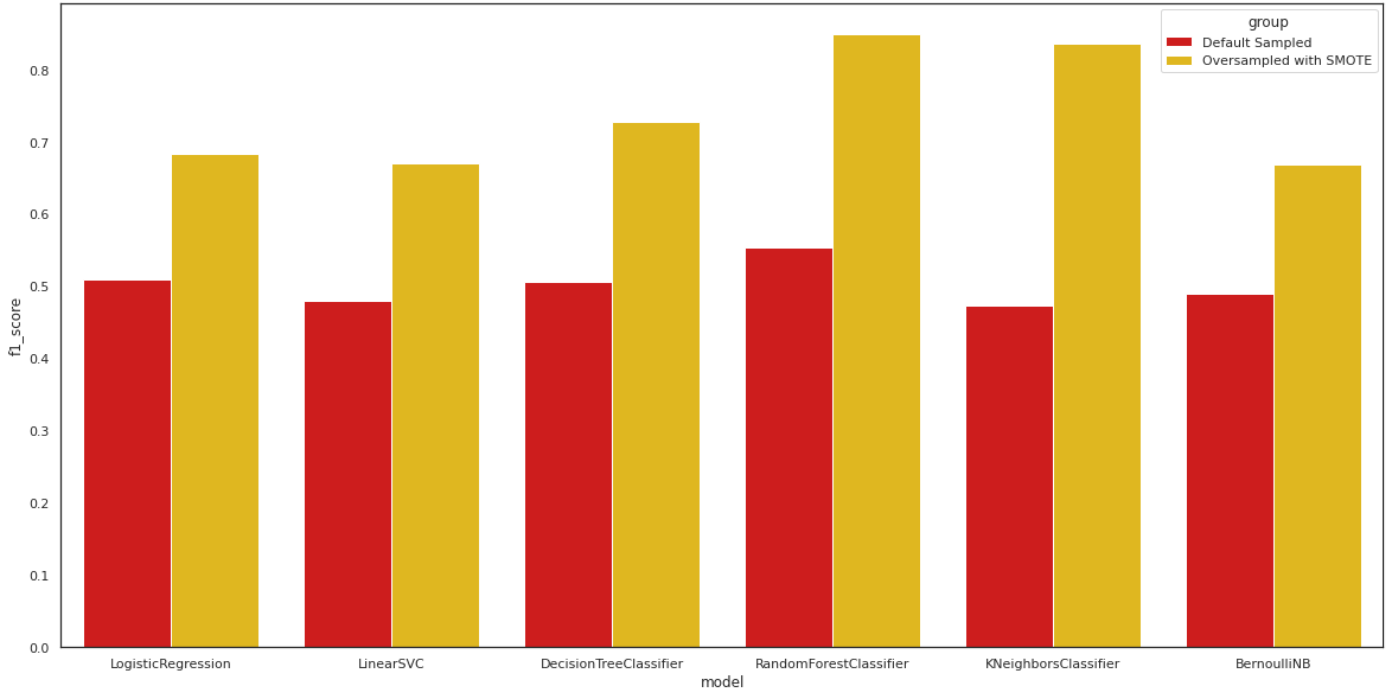
Fig 20. Metrics scores for all algorithms trained

Fig 21. Comparison of F1 scores

A huge performance increase can be observed with oversampled dataset for all algorithms. Random forest tops the table, whereas k nearest neighbor achieved highest performance increase with the change of sampling type.

**Model Selection:**

Final model has been built on Random Forest, with hyper parameter tuning done. The parameters are,

**max\_depth** = **100**; **n\_estimators** = **500**;

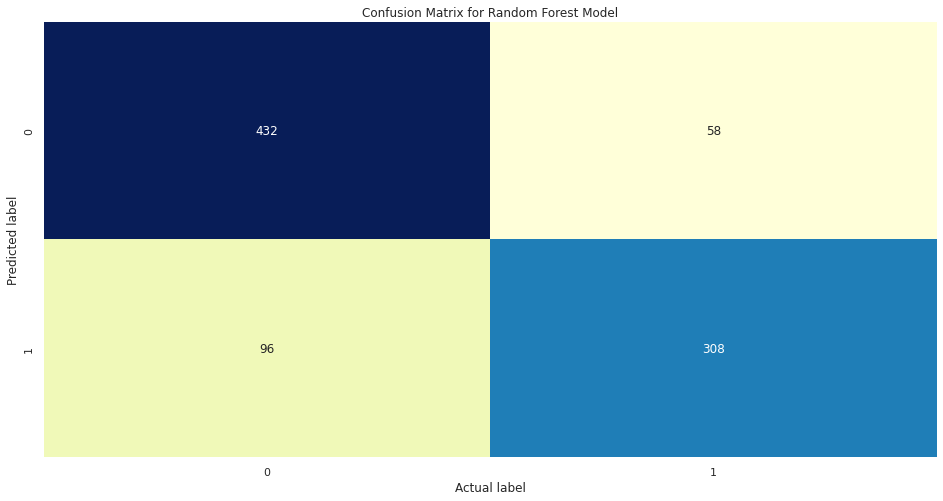
**Confusion Matrix:**

Fig 22. Confusion Matrix

**Classification Report:**

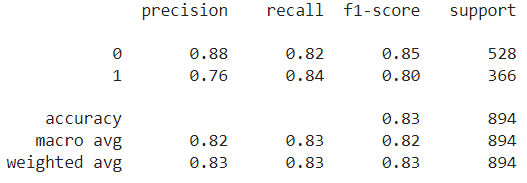


Fig 23. Classification Report

**Balanced Accuracy Score: 0.8298**

**Average Weighted F1 Score: 0.8287**

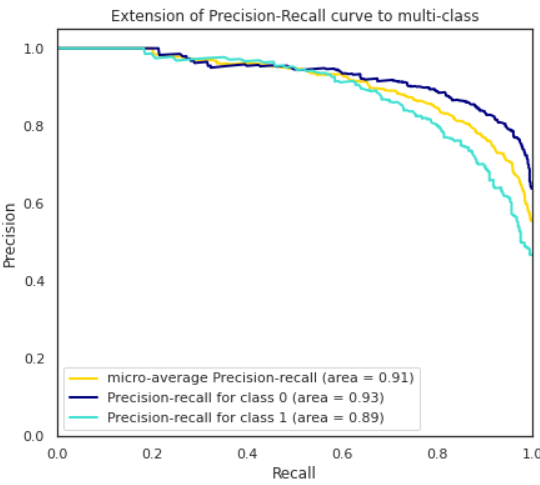
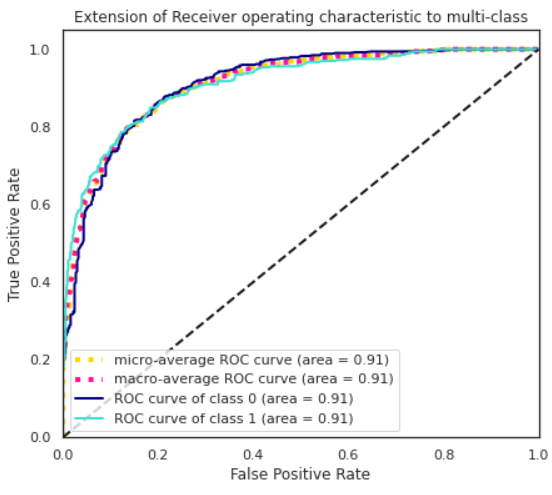
**ROC AUC Graph:**

Fig 24. Extension of ROC to multi-class Fig 25. Extension of Precision-Recall

**Feature Importance:**

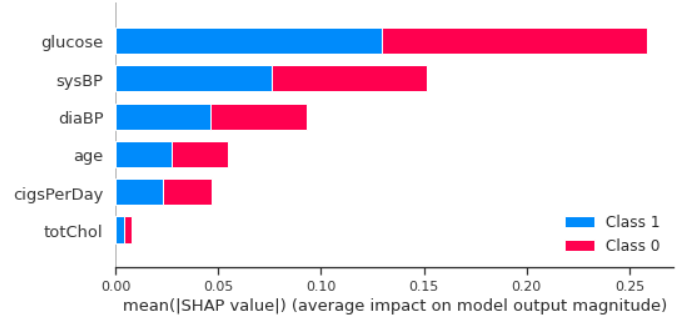
****

Fig 26. Feature importance using SHAP values

Glucose & sysBP contribute more to the model thus have a high predictive power.

**Limitation:**

* Most of the data on the positive cases were artificially synthesized using SMOTE and as such they may not be a true representation of the actual population.
* The interpretation of its results is limited to the risk for myocardial infarction or coronary death in the next **10** years only.

**Scope of Improvement:**

* In general, we’ve huge scope of improvements. We could try to add more data to our set to generalize, especially the data from actual positive cases as most of the data we’re working is artificially synthesized data.
* We can try some other normalization method like MinMaxScaler etc. Also, other strong ensemble learning method like XGBoost, CatBoost, AdaBoost can increase performance effectively but with the cost of more computation.

**Conclusion:**

* The most important features in prediction are glucose and sysBp.
* Random forest is the best performed learning algorithm. The recall is ~83% which is somewhat satisfactory but not in sensitive cases like these. It simply means, out of 100 every 13 peoples with the risk of CHD leave untreated without knowing anything.
* The area under roc curve is 0.91 which is satisfactory.
* Balancing the dataset using SMOTE helped tremendously in improving the model performance.
* Males seem to be more susceptible to heart disease than females. Increase in Age, number of cigarettes smoked per day and systolic Blood Pressure also show increasing odds of having heart disease.
* Overall model could be improved with more records of actual positive classes.

**References:**

* <https://towardsdatascience.com/simple-example-using-boruta-feature-selection-in-python-8b96925d5d7a>
* <https://christophm.github.io/interpretable-ml-book/shap.html#shap-feature-importance>
* <https://towardsdatascience.com/5-feature-selection-method-from-scikit-learn-you-should-know-ed4d116e4172>
* <https://towardsdatascience.com/having-an-imbalanced-dataset-here-is-how-you-can-solve-it-1640568947eb>

THE END