**We want to predict the probability of death from heart disease basedon three risk factors: age, gender, and blood cholesterol level. What isthe most appropriate algorithm for this case?**

**mauzum shamil a m**

**mohammed arshath**

**INTRODUCTION**

Heart disease is the major cause of morbidity and mortality globally: it accounts for more deaths annually than any other cause. According to the [WHO](https://www.who.int/health-topics/cardiovascular-diseases/#tab=tab_1), an estimated 17.9 million people died from heart disease in 2016, representing 31% of all global deaths. 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. In the United States, for example, it is estimated that someone has a heart attack every 40 seconds and about **805,000 Americans**have a heart attack every year ([CDC 2019](https://www.cdc.gov/heartdisease/facts.htm)).

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.

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.

In this article, I will be giving you a walk through on the development of a screening tool for predicting whether a patient has 10-year risk of developing coronary heart disease(CHD) using different Machine Learning techniques on the [Framingham dataset](http://biolincc.nhlbi.nih.gov/studies/framcohort/).

**2. DATA SET DESCRIPTION**

The data set is publicly available on the [Kaggle](https://www.kaggle.com/amanajmera1/framingham-heart-study-dataset." \t "_blank)website, and it is from an ongoing cardiovascular study on residents of the town of Framingham, Massachusetts. The classification goal is to predict whether the patient has 10-year risk of future coronary heart disease (CHD). The data set provides the patients’ information. It includes over 4,000 records and 15 attributes. Each attribute is a potential risk factor. There are both demographic, behavioral and medical risk factors.

**Attributes:**

1. **Demographic**:

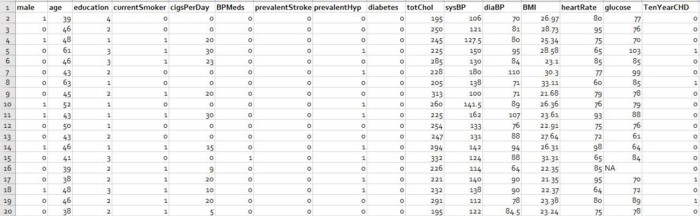
* Sex: male or female(Nominal)
* Age: Age of the patient;(Continuous — Although the recorded ages have been truncated to whole numbers, the concept of age is continuous)

**TOOL DEVELOPMENT**

The full code for this article can be found [here](https://github.com/amayomode/heart-disease-prediction/tree/master/Heart%20Disease%20Prediction). It is implemented in Python and different classification algorithms are used. Below is a brief description of the general approach that I employed:

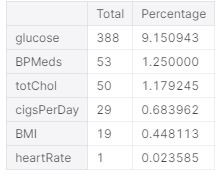
1. **Data cleaning and pre-processing**: Here I checked and dealt with missing and duplicate variables from the data set as these can grossly affect the performance of different machine learning algorithms (many algorithms do not tolerate missing data).
2. **Exploratory Data Analysis**: Here I wanted to gain important statistical insights from the data and the things that I checked for were the distributions of the different attributes, correlations of the attributes with each other and the target variable and I calculated important odds and proportions for the categorical attributes.
3. **Feature Selection**: Since having irrelevant features in a data set can decrease the accuracy of the models applied, I used the Boruta Feature Selection technique to select the most important features which were later used to build different models.
4. **Model development and comparison**: I used four classification models, i.e., Logistic Regression, K-Nearest Neighbors, Decision Trees and Support Vector Machine, After which I compared the performance of the models using their accuracy and F1 scores. I then settled with the best performing model.

**3.1 Data cleaning and pre-processing**

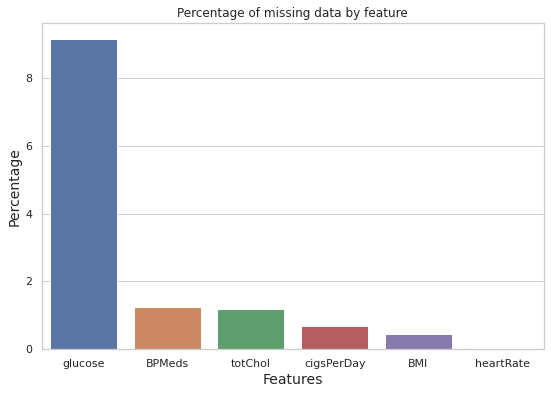


First 20 records of the data set

There were no duplicate entries in the data set but some had missing values and the table below gives a summary of these:



percentage of missing values per feature

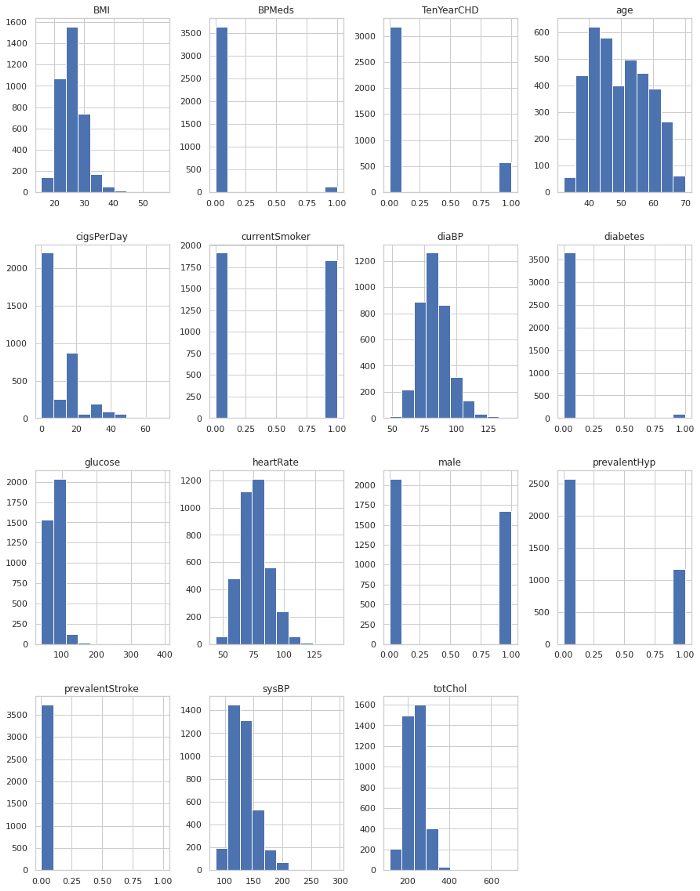


At 9.15%, the blood glucose entry had the highest percentage of missing data. The other features have very few missing entries.

The missing entries accounted for only 12% of the total data and could, therefore, be dropped without losing a lot of data.

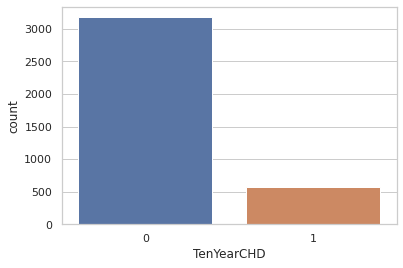
**3.2 Exploratory Data Analysis**

The first step was to check the distribution different attributes and this was best visualized by histograms



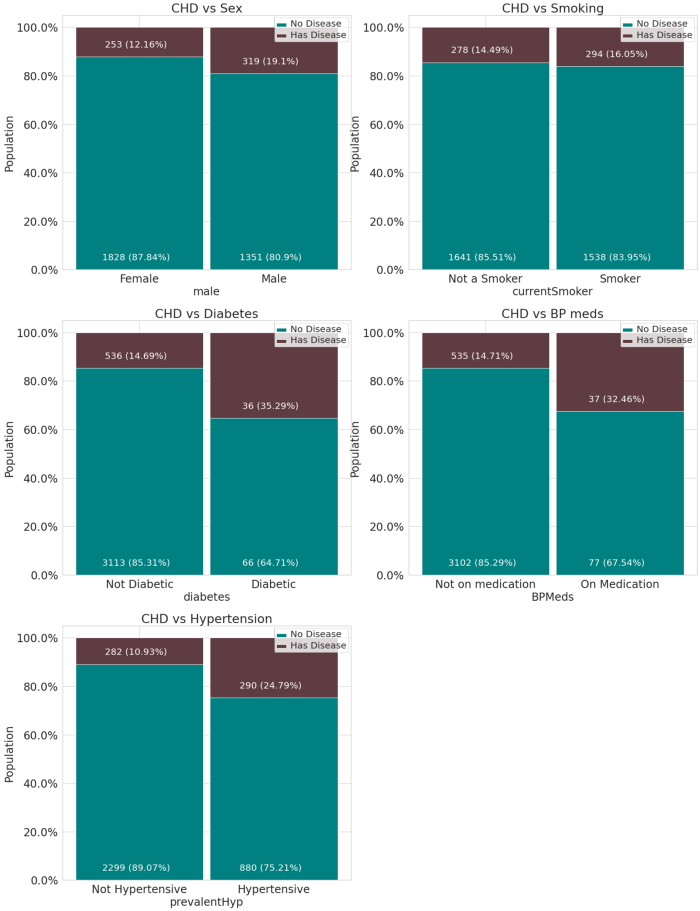
Data distribution

It is easy to pick out the categorical and continuous variables from the distribution plots. Also, it can be seen that none of the respondents had prevalent stroke and very few were diabetic, on blood pressure medication or hypertensive. These distributions also raised the suspicion that the data set might not be properly balanced and to confirm this I compared the number of positive and negative cases and true to my suspicions there were 3179 respondents without CHD and 572 patients with CHD.



Imbalanced data set

To gain more insight into the data I checked the proportions of positive and negative cases in each category.

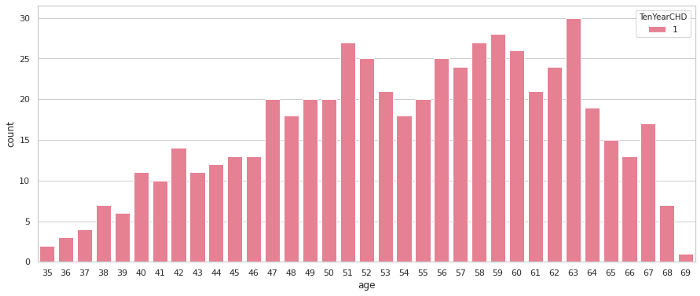


Categorical variable proportions

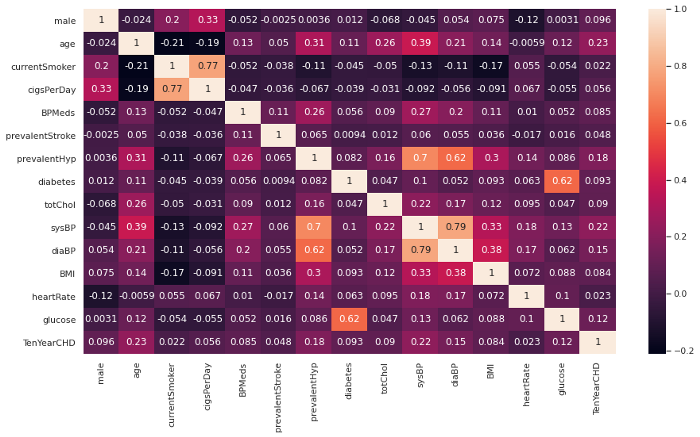
Due to the imbalanced nature of the data set it was difficult to make conclusions but based on what is observed but these are the conclusions that could be drawn:

* Slightly more males are suffering from CHD than females.
* The percentage of people who have CHD is almost equal between smokers and non smokers.
* The percentage of people who have CHD is higher among the diabetic, and those with prevalent hypertension as compared to those who don’t have similar morbidities.
* A larger percentage of the people who have CHD are on blood pressure medication.

Another interesting trend I checked for was the distribution of the ages of the people who had CHD and the number of the sick generally increased with age with the peak being at 63 years old.



The final step was to check the correlation of the different features with the target variable and with each other as this would not only give a good estimate of the strength of the features as predictors of coronary heart disease but also reveal any [co-linearity](http://faculty.cas.usf.edu/mbrannick/regression/Collinearity.html) among the features.



Correlation matrix

From the matrix, there are no features with a correlation of more than 0.5 with the 10-year risk of developing CHD and this shows that the features are poor predictors. However, the features with the highest correlation are age, prevalent hypertension and systolic blood pressure.

Also, there are a couple of features that are highly correlated with one another and it makes no sense using both of them to building a machine learning model. These include: blood glucose and diabetes (obviously); systolic and diastolic blood pressures; cigarette smoking and the number of cigarettes smoked per day.

**3.3 Feature Selection**

The results from the correlation matrix prompt the need for feature selection. To do this I employed the [Boruta Feature Selection](https://www.datasciencecentral.com/profiles/blogs/select-important-variables-using-boruta-algorithm" \t "_blank) algorithm which is a wrapper method built around the [random forest classification algorithm](https://en.wikipedia.org/wiki/Random_forest). It tries to capture all the important, interesting features in a data set with respect to an outcome variable.

It works in the following way:

* First, it adds randomness to the given data set by creating shuffled copies of all features (which are called shadow features).
* Then, it trains a random forest classifier on the extended data set and applies a feature importance measure (the default is[Mean Decrease Accuracy](https://www.thelearningmachine.ai/mda)) to evaluate the importance of each feature where a higher score means a more important feature.
* At every iteration, it checks whether a real feature has a higher importance than the best of its shadow features (i.e. whether the feature has a higher Z-score than the maximum Z-score of its shadow features) and constantly removes features which are deemed highly unimportant.
* Finally, the algorithm stops either when all features get confirmed or rejected or it reaches a specified limit of random forest runs.

Check the full description [here](http://danielhomola.com/2015/05/08/borutapy-an-all-relevant-feature-selection-method/)

After running the algorithm for 100 iterations the top selected features were: Age, total cholesterol, systolic blood pressure, diastolic blood pressure, BMI, heart rate and blood glucose.

I then calculated the odds ratio of the top features and the ten year risk of developing CHD and these were the results:

CI 5% CI 95% Odds Ratio  
age 1.011381 1.033813 1.022536  
totChol 0.994963 0.999184 0.997071  
sysBP 1.018236 1.031493 1.024843  
diaBP 0.962258 0.984627 0.973378  
BMI 0.929304 0.973798 0.951291  
heartRate 0.963690 0.977730 0.970685  
glucose 1.001074 1.007518 1.004291

Holding all other features constant, the odds of getting diagnosed with heart disease increases with about 2% for every increase in age an systolic blood pressure.

The other factors show no significant positive odds.

**3.4 Model Development And Comparison**

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.

In our case, the number of negative cases (3179) greatly exceeds the number of positive cases(572). If we, for example, train a model that always predicts the negative classes, it will achieve high accuracy of 84.75 %(3179/(3179+572) x 100) but have a sensitivity of 0% (0/(0+572) x 100) because it never predicts a positive case.

To address this problem. I balanced the data set using the Synthetic Minority Oversampling Technique (SMOTE). This is how it works:

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. — Page 47, Imbalanced Learning: Foundations, Algorithms, and Applications, 2013.

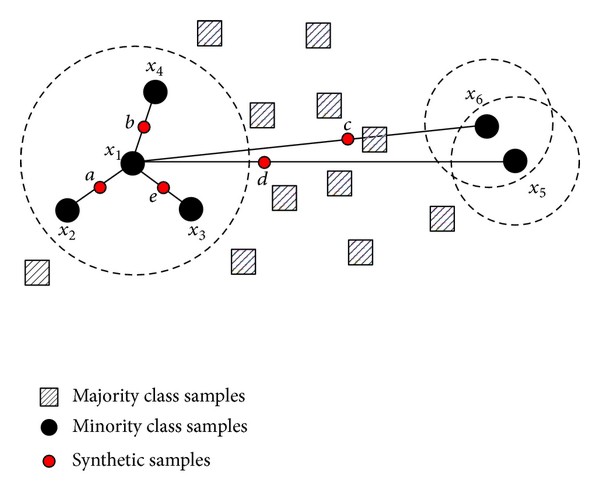
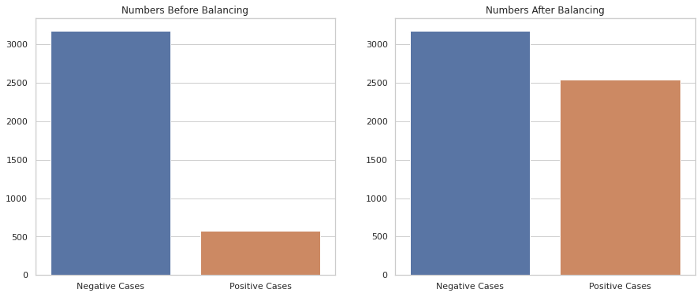


Image by [Feng Hu](https://www.hindawi.com/journals/mpe/2013/694809/" \t "_blank) via [Hindawi](https://www.hindawi.com/journals/mpe/2013/694809/" \t "_blank) (CC0)

This procedure can be used to create as many synthetic examples for the minority class as are required. It suggests first using random undersampling to trim the number of examples in the majority class, then use SMOTE to oversample the minority class to balance the class distribution.

After using this technique, the resultant data set was much more balanced with 3178 negative cases and 2543 positive cases

Numbers before {positive: 3179, negative: 572}   
Numbers after {positive: 3178, negative: 2543}



Data set balancing by SMOTE

After balancing the data set, I scaled the features to speed up the training of the classifiers and then split the data into a training and test set at a ratio of 0.8 to 0.2 respectively

Using the training set, I trained four classifiers, i.e.,:

1. **Logistic regression**: Which models the probability of a data point belonging to a particular class and assigns this point the appropriate label based on a chosen threshold.

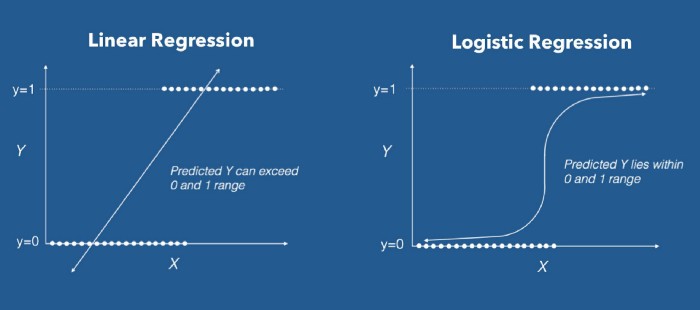


Image by [James Le](https://www.datacamp.com/profile/khanhle1013) via [Datacamp](https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.datacamp.com%2Fcommunity%2Ftutorials%2Flogistic-regression-R&psig=AOvVaw1tu3giJ7dvIrJdvld1_TXt&ust=1588091566496000&source=images&cd=vfe&ved=0CAMQjB1qFwoTCIj5_LaEiekCFQAAAAAdAAAAABAI" \t "_blank) (CC0)

**2. K-nearest neighbors**: Which attempts to determine what group a data point is in by looking at the data points around it. For example, given a data point C, if the majority of the points around it are in group A, then it is likely that the data point in question will belong to group A rather than B, and vice versa.



Image by [Chaitanya Reddy Patlolla](https://medium.com/@chaitanyareddypatlolla?source=post_page-----6851280d4c93----------------------) via [Medium](https://medium.com/datadriveninvestor/machine-learning-getting-started-with-k-nearest-neighbours-6851280d4c93) (CC BY-NC-ND 2.0)

**3. Decision trees**: Which is based on 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. 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 sub-tree rooted at the new nodes.

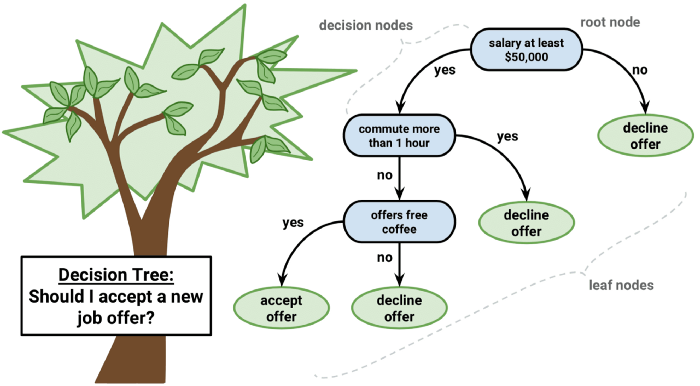


Image by [Rahul Saxena](https://dataaspirant.com/author/rahul-saxena/" \t "_blank) via [Dataaspirant](https://www.google.com/url?sa=i&url=https%3A%2F%2Fdataaspirant.com%2F2017%2F01%2F30%2Fhow-decision-tree-algorithm-works%2F&psig=AOvVaw2edoz1eGedERpV3gamwnwj&ust=1588092099583000&source=images&cd=vfe&ved=0CAMQjB1qFwoTCOi5k5-GiekCFQAAAAAdAAAAABAU" \t "_blank) (CCo)

**4. Support vector machine**: Which is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data, the algorithm outputs an optimal hyperplane which categorizes new examples based on which side they lie in relation to it. In a two dimensional space this hyperplane is a line dividing a plane in two parts where in each class lies on either side.

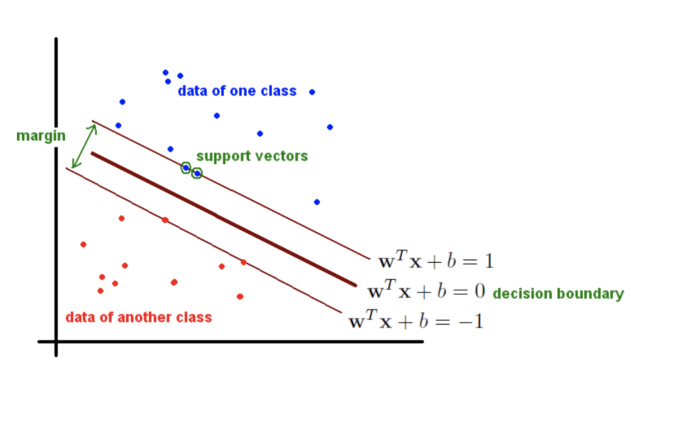


Image by [François de Ryckel](https://www.google.com/url?sa=i&url=https%3A%2F%2Ffderyckel.github.io%2Fmachinelearningwithr%2Fsvm.html&psig=AOvVaw3AyyyBipnHNdVZPkmu-3JG&ust=1588092217619000&source=images&cd=vfe&ved=0CAMQjB1qFwoTCMDqzOCGiekCFQAAAAAdAAAAABAJ) via [Github](https://fderyckel.github.io/machinelearningwithr/svm.html" \t "_blank) (CC BY-NC-ND 2.0)

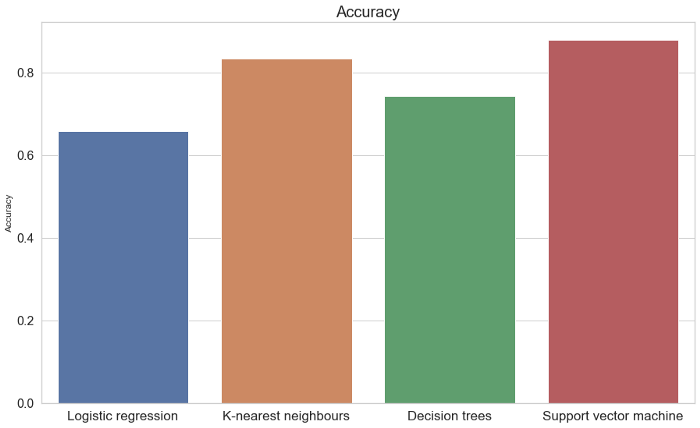
After training each model and tuning their [hyper-parameters](https://en.wikipedia.org/wiki/Hyperparameter_(machine_learning)) using [grid search](https://scikit-learn.org/stable/modules/grid_search.html), I evaluated and compared their performance using the following metrics:

1. **The accuracy score:** which is the ratio of the number of correct predictions to the total number of input samples. It measures the tendency of an algorithm to classify data correctly.
2. **The F1 Score**: Which is defined as the weighted [harmonic mean](https://deepai.org/machine-learning-glossary-and-terms/harmonic-mean) of the test’s [precision and recall](https://deepai.org/machine-learning-glossary-and-terms/precision-and-recall). By using both precision and recall its gives a more realistic measure of a test’s performance. (Precision, also called the positive predictive value, is the proportion of positive results that truly are positive. Recall, also called sensitivity, is the ability of a test to correctly identify positive results to get the true positive rate).
3. **The Area under the**[**ROC**](https://en.wikipedia.org/wiki/Receiver_operating_characteristic)**Curve (AUC):** Which provides an aggregate measure of performance across all possible classification thresholds. It gives the probability that the model ranks a random positive example more highly than a random negative example

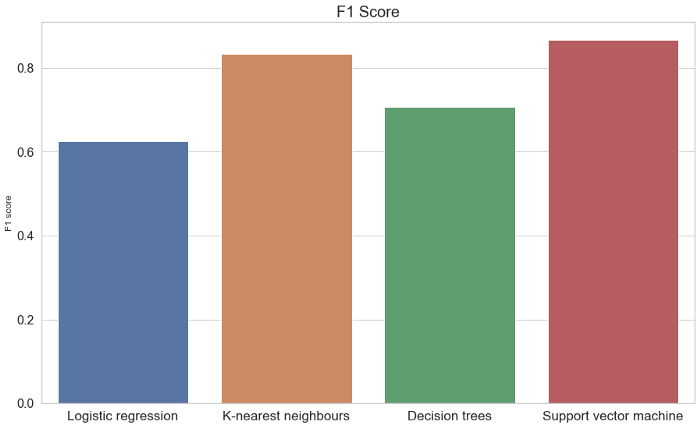
Here are the results:



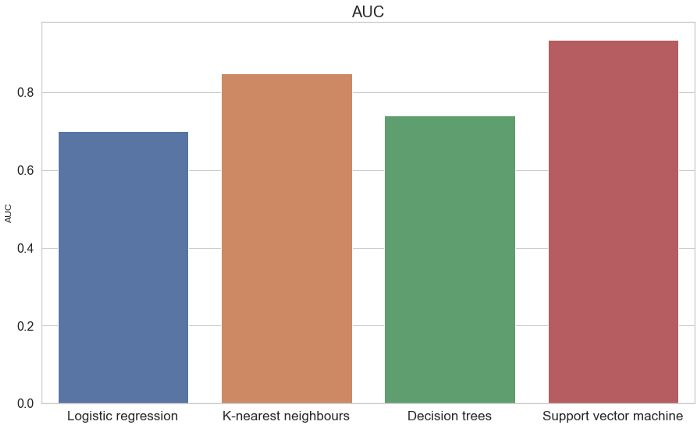
Performance scores of the different models



Comparison of accuracies



Comparison of F1 scores



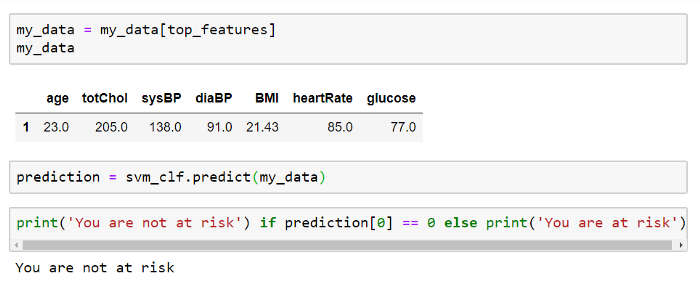
Comparison of AUCs

The support vector machine was the best performing model across all metrics. It’s best parameters were a radial kernel, a C value of 10 and a gamma value of 1. Its high AUC and F1 score also show that the model has a high true positive rate and is thus sensitive to predict if one has a high risk of developing CHD , i.e., getting a heart attack within 10 years.

**4.CONCLUSIONS**

This model can then be used as a simple screening tool and all that we need to do is to input ones: age, BMI, systolic and diastolic blood pressures, heart rate and blood glucose levels after which the model can be run and it outputs a prediction.

Like all good scientists, I decided to test the tool on myself by feeding it with my personal data and here are the results:



Personal predictions

I can say, with 84% confidence, that I don’t have a risk of developing CHD within the next 10 years . I trust these predictions because I do work out, at least once in a while.