Coronary Heart Disease (CHD) Prediction

What causes Coronary Heart Disease?

ABSTRACT

The 'framingham' dataset provides information whether the participants developed Coronary Heart Disease (CHD) after ten years of the study. The objective of the project is to determine the risk factors for predicting the probability of developing CHD. The initial hypothesis of the project states that BMI is the biggest risk factor of CHD. To determine the importance of the features in the dataset, data cleaning will be conducted. This process not only helps filter the necessary features, but also determines whether the initial hypothesis stands or not. Afterward, various machine learning algorithms are used to test model accuracies.

Keywords: Coronary Heart Disease, data cleaning, machine learning

1 Introduction

Coronary heart disease (CHD) is one of the most common types of heart disease in America. That is, knowing what causes CHD is important. According to Adam Felman, an assistant editor for Medical News today, CHD is caused by a plaque in the arteries, which decreases blood and oxygen supply to the heart. [Felman 2019]. The symptoms of CHD include chest pain, shortness of breath, heart attack, etc. This brings another good question; What are the known factors to develop CHD? Gender, diets, and medical conditions are the broader categories followed by numerous sub-categories. Before moving on to analyzing the dataset, let's look at what other researchers have done with the 'framingham' dataset.

The 'framingham' dataset can be found from the open source website called Kaggle, which is available at the URL https://www.kaggle.com/amanajmera1/framingham-heart-study-dataset#framingham.csv. Many studies from Kaggle indicate that the Logistic Regression model for 'framingham' is a good machine learning algorithm to use as the target class is binary. Popular conclusions suggest that men are more susceptible to heart disease than women as well as increasing in age, number of cigarettes smoked per day, and systolic blood pressure also affect developing CHD. In the following paragraphs,

2 Data Collection

The dataset includes 4,240 samples with a total of 16 columns, consisting of eight categorical and eight numerical attributes. Categorical attributes include 'male', 'education', 'currentSmoker', 'BPmeds', 'prevalentStroke', 'prevalentHyp', 'diabetes', and 'TenYearCHD' with 'TenYearCHD' being the target class. Numerical attributes include 'age', 'cigPerDay', 'totChol', 'sysBP', 'diaBP', 'BMI', 'heartRate', and 'glucose'.
The dataset has some missing as shown in Figure 1. This indicates that imputation of missing data is necessary. How the missing values are handled will be explained in the following paragraph.

Column Name	Data Type	Total	Missing Value
male	Categorical	4240	0
age	Numerical	4240	0
education	Categorical	4135	105
currentSmoker	Categorical	4240	0
cigsPerDay	Numerical	4211	29
BPMeds	Categorical	4187	53
prevalentStroke	Categorical	4240	0
prevalentHyp	Categorical	4240	0
diabetes	Categorical	4240	0
totChol	Numerical	4190	50
sysBP	Numerical	4240	0
diaBP	Numerical	4240	0
BMI	Numerical	4221	19
heartRate	Numerical	4239	1
glucose	Numerical	3852	388
TenYearCHD	Target	4240	0

Figure 1: overview of 'framingham' dataset features

3 Data Processing

Data processing is defined as a series of operations on data, especially by a computer, to retrieve, transform, or classify information. Data processing is an important process before building models on the dataset. It helps improve the data quality and in doing so, increase overall productivity. The techniques implemented for this project are handling missing values, categorical data, and feature engineering to compare model

accuracies. As mentioned in the previous paragraph, data imputation is conducted to handle the missing values.

3.1 Handling Missing Values

The project handles missing values in multiple ways. First, figuring out how missing values should be handled is decided by examining the distributions of the features. For Binomial Distributions, NA values are replaced with mode (Figure 2) while NA values are replaced with the median value for asymmetrical distributions (Figure 3). As shown in Figure 1, seven columns contain missing values: education, cigsPerDay, BPMeds, totChol, BMI, heartRate and glucose. The glucose feature contains 388 missing values. Although 388 seems to be a large number, it is less than 10% of the data, thus it does not need to be removed. Among the seven features with missing values, there are two categorical features: education and BPMeds. Missing values of these two categorical features are replaced by "the most frequent values" within each column (Figure 2).

For the five numerical features (cigsPerDay, totChol, BMI, glucose, HeartRate), the missing values are replaced with the median value since due to asymmetrical distributions in the histograms. For example, as shown by Figure 3, the histogram of CigsPerDay is a right-skewed asymmetrical, indicating that replacing the missing values with median is better than the mean.

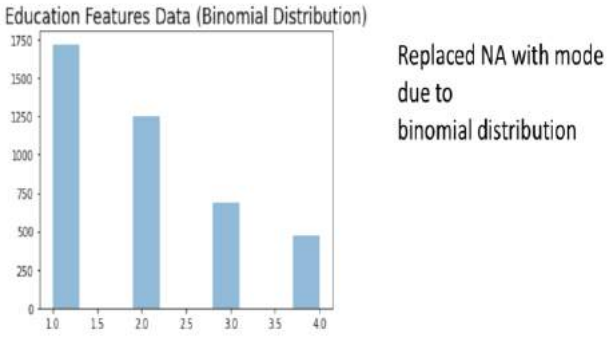

Figure 2: Histogram of Categorical Features

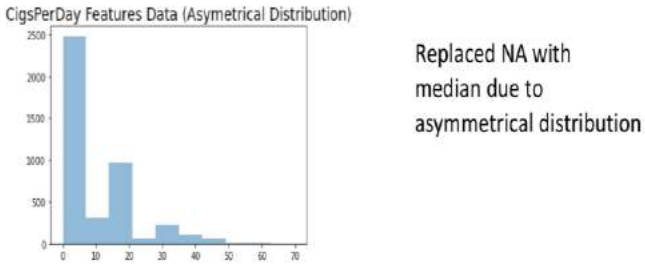

Figure 3: Histogram of Numerical Features

3.2 Categorical Data

There are seven categorical features in the dataset). The categorical features have been converted to numeric values (See Table 1) by OneHotEncoding, a machine learning technique for categorical data. Table 1 shows that the education column contains four education levels represented by integers from 1 to 4. Even though an ordinal relationship exists, it is acceptable as the education levels have a natural ordering with each other.

The TenYearCHD feature, which is the dependent variable in this project is a categorical data as well. However, just like other categorical features, it is already mapped into binary values of 0 and 1, representing "No" and "Yes" respectively. Therefore, there is no need to deal with this categorical variable.

Categorical Feature	Value
male	0 = Female 1= Male
education	1 = Some High School 2 = High School or GED 3 - Some College 4 = College
currentSmoker	0 = non smoker 1 = smoker
BPMeds	0 = not on blood pressure medication 1 = on blood pressure medication
prevalentStroke	0 = no 1 = yes
diabetes	0= patient is not diabetic 1 = patient is diabetic

Table 1: Categorical Features and Description

3.3 Feature Engineering

Feature engineering, also known as dimensionality reduction, is a vital process in machine learning. The definition of feature engineering is the process of using domain knowledge to extract features from raw data via data mining techniques. This technique can have a huge impact on the result of Machine Learning. According to Amit Shekhar, a co-founder and CEO of MindOrks & AfterAcademy, "If feature engineering is done correctly, it increases the predictive power of machine learning algorithms by creating features from raw data that help facilitate the machine learning process. Feature Engineering is an art" [Shekhar 2018]. Additionally, dimensionality reduction gives computational efficiency and improved model accuracy. In this project, dimensionality reduction is performed in two different ways: feature selection and feature extraction. By comparing model performances of just raw data, feature selection, and feature extraction, we can determine which technique gives the best outcome.

3.3.1 Feature Selection

Feature selection technique is often used in Machine Learning as a strategy to include variables from a dataset. The fundamental idea of feature selection is to fit the features that are statistically most relevant to the model. Three major techniques in feature selection are backward elimination, forward selection, and stepwise.

Backward elimination is decided on a significant level for a variable to be retained in the model using a p-value (Figure 4). First, a model is fitted with all independent variables. Then the variable with the highest p-values is removed before the model is fitted again with the removed variable. The process continues until no further steps are necessary. Forward selection, on the other hand, is the opposite process of backward elimination A model is created with each independent variable. Then the variable with the lowest p-value is selected to create a new model. A new variable with the lowest p-value is added until no further steps are necessary. Lastly, there is stepwise technique. Stepwise technique is a hybrid between backward elimination and forward selection. The selection process starts with the forward selection process, then onto the backward elimination until no new variables can be added or need to be dropped from the model. After the feature selection process, the following features are selected.

```
Only Independant Variables (X1-X6) have impact on predicting Y:
        Column 1(x0) = column of 1s
        Column 2(X1) = male
        Column 3(x2) = age
        Column 4(x3) = cigsPerDay
        Column 5(X4) = prevalentStroke
        Column 6(x5) = sysBP
        Column 7(x6) = glucose
```

Figure 4: features after feature selection

3.3.2 Feature Extraction

Feature extraction is a process of dimensionality reduction by which an initial set of raw data is reduced to more manageable groups for data processing. In other words, feature extraction allows the data to be projected into a new lower dimensional space while minimizing the loss of information. For this project, three approaches were taken into consideration; Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and Kernel Principal Component Analysis (KPCA).

PCA is a method that rotates the dataset in a way such that the rotated features are statistically uncorrelated. That is, PCA rotates the data in the direction of the largest variance. This rotation is often followed by selecting only a subset of the new features, according to how important they are for explaining the data. The goal of performing PCA is to capture as much variability as possible. One disadvantage of PCA is interpretation of data can be difficult.

The basic idea of LDA is similar to that of PCA. The major difference between PCA and LDA is that PCA is unsupervised whereas LDA is supervised. While PCA attempts to find the orthogonal component axes of maximum variance in a dataset, LDA is to find the feature subspace that optimizes class separability.

The two techniques above deal with linear problems. However, dealing with nonlinear problems happens more often in reality. This is when we use KPCA. As in the name of KPCA, KPCA is a kernelized version of PCA. KPCA transforms the data that is not linearly separable onto a new lower dimensional space for linear classifiers. In other words, first, a nonlinear mapping is performed to transform the data onto a higher dimensional space. After this, PCA is performed in this higher dimensional space to project the data back onto a lower dimensional space where the data points can be separated by a linear classifier.

4 Model Training

For this project, seven different models are used: Logistic Regression, SVM (linear and RBF kernel), Naïve Bayes, Decision Tree, Random Forest, and KNN. For each model, the first step is to train each model by splitting the dataset into 70% training dataset and 30% testing dataset. That is, 70% of data is assigned for the training purpose while the rest 30% of data is assigned for the validation. Each machine learning model is explained in the following paragraphs.

4.1 Logistic Regression

Despite having "regression" in its name, logistic regression is one of the most widely used techniques in machine learning to solve classification problems. It can be used when the dependent variable

is categorical. Logistic function is also called Sigmoid function, in which a linear model $(\beta 0 + \beta 1x)$ is included in $\frac{1}{1+e-z}$:

$$P(yi\ = 1|X) = \frac{1}{1 + e-(\beta 0\ +\ \beta 1x)}$$

In the equation above, X is the training data, and $P(yi\ = 1|X)$ is the probability of the *i*th observation's target value. The idea is that if the P-value,$P(yi\ = 1|X)$, is greater than or equal to 0.5, class 1 is predicted. Otherwise, class 0 is predicted. After training the dataset, the logistic regression model produces 86.08% accuracy (Figure 5).

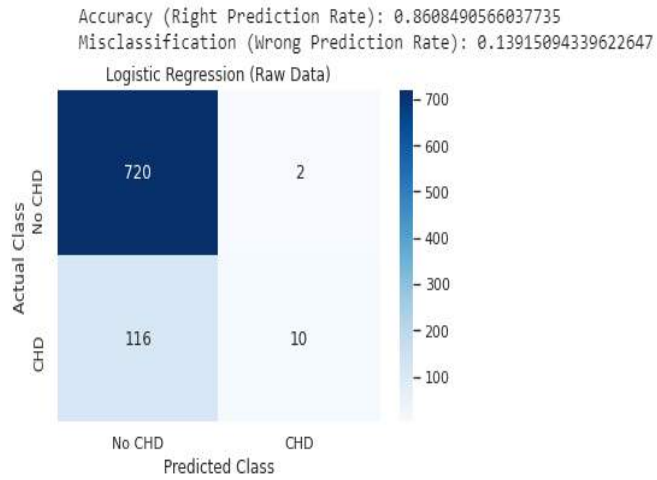

Figure 5: Logistic Regression Confusion Matrix

4.2 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a machine learning algorithm for classification, regression, and other learning tasks. The idea of SVM is to make the classifier to have the greatest possible margin from the decision boundary as the greatest possible margin indicates more confidence about the prediction made. There are some advantages and disadvantages of SVM.

One of the major advantages is that SVM is effective in higher dimensional spaces when the number of features is greater than the number of observations [Chang 2019]. Also, SVM is versatile as different kernel functions can be used for each specific decision function. On the other hand, SVM can be easily led toward overfit if the number of features is much greater than the number of observations. In this case, it is important to carefully choose the kernel functions and regularization term.

In this project, linear, RBF (Radial Basis Function) and Poly kernels are used for SVM. The linear, polynomial and RBF or Gaussian kernel are simply different in case of making the hyperplane decision boundary between the classes. The kernel functions are used to map the original dataset (linear/nonlinear) into a higher dimensional space with view to making it linear dataset. Linear Kernel is used when the data is Linearly separable, that is, it can be separated using a single Line. RBF kernel is a function whose value depends on the distance from the origin or from some point. The polynomial kernel is to calculate the dot product by increasing the power of the kernel. Figure 6, 7, and 9 show the accuracy of each kernel SVM respectively.

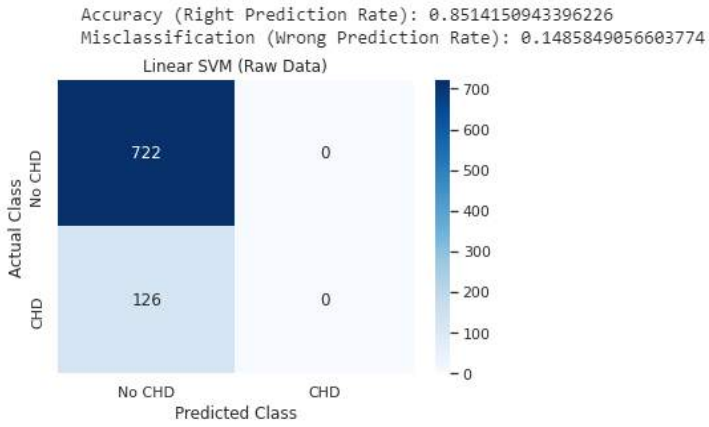

Figure 6: Linear SVM Confusion Matrix

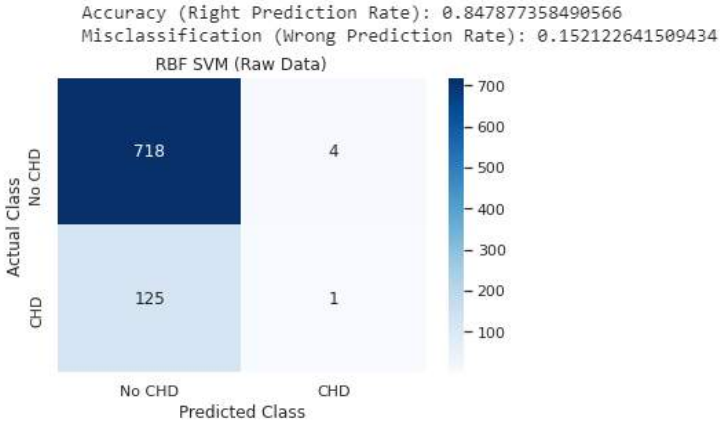

Figure 7: RBF SVM Confusion Matrix

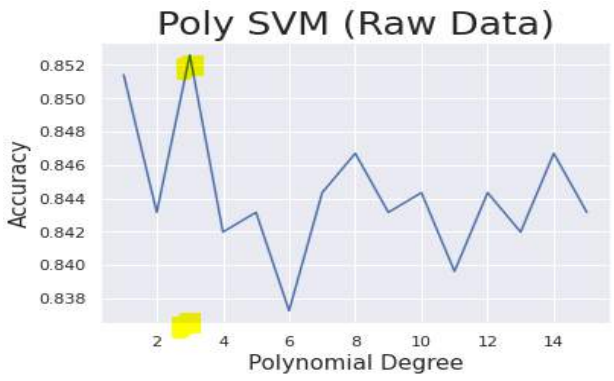

Figure 8: Poly SVM Optimal Polynomial Degree =3

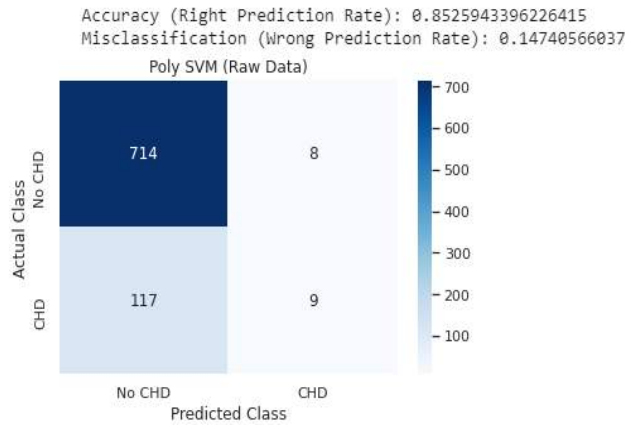

Figure 9: Poly SVM Confusion Matrix

4.3 Naïve Bayes

Naïve Bayes is a supervised machine learning algorithm that is widely used in classification problem Naïve Bayes. Naïve Bayes comes from Bayes' Theorem. The word Naïve is added for Naïve assumption that the features are statistically independent. As suggested by Rish, "Naïve Bayes works well for certain nearly functional feature dependencies, thus reaching its best performance in two opposite cases: completely independent features (as expected) and functionally dependent features (which is surprising)" [Rish 2001]. The following is how the Naïve Bayes formula is constructed.

$$P(y|(x1,\ldots,xj\ ) = \frac{P(x1,\ldots,xj|\ y)\ P(y)}{P(x1,\ldots,xj)}$$

where $P(y|(x1,\ldots,xj\ )$ is the probability of A given the observation value for j features, x1,...xj. This is called the posterior probability; **P(x1, ..., xj|y)** is the probability of x1...xj given class y. **P(y)** is the probability of y. This is called the prior probability of y. **P(x1, …, xj)** is the probability of x1, …, xj.

There are three types of Naïve Bayes classifiers which are based on different distributions: Gaussian, multinomial and Bernoulli distributions. How to choose one classifier versus the others depends on the nature of the target feature (binary or continuous). As our data is binary classification, Gaussian classifier is used. Shown in Figure 10, the model generates 82.43% accuracy.

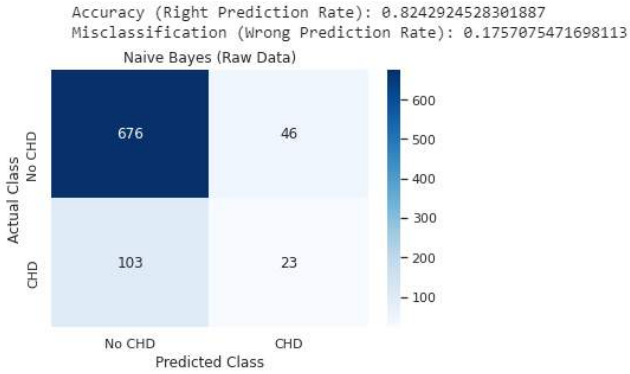

Figure 10: Naive Bayes Confusion Matrix

4.4 Decision Tree

Decision tree is one of the predictive modeling methods used in both regression and classification learning tasks. It breaks down the dataset into smaller subsets by selecting certain features. The final result is a tree with internal decision nodes and leaf nodes. A decision node has two or more branches, each branch represents an outcome of the tested feature. A leaf node represents a class label. For each node, one feature is chosen to split the training dataset into distinct classes as much as possible. The criterion for feature selection is choosing the feature that results in greatest information gain, and the entropy function shown below gives the result in bits.

$$\text{entropy}(p_1, p_2, \cdots, p_n) = -p_1 log p_1 - p_2 log p_2 - \cdots - p_n log p_n$$

The gain is the information before split minus the information after split. The feature with the greatest gain is chosen. The decision tree model generates 74.29% accuracy as shown in Figure 11.

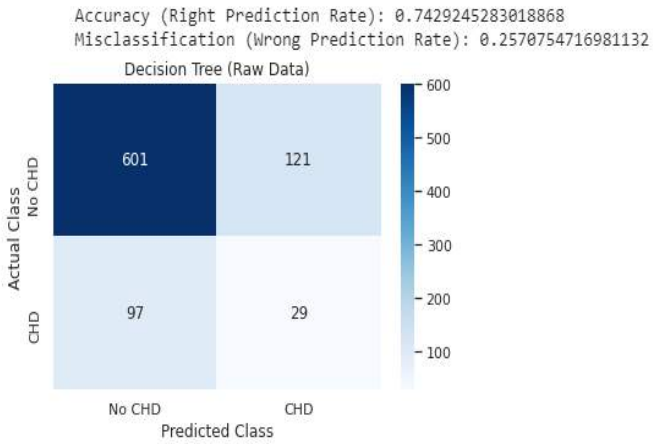

Figure 11: Decision Tree Confusion Matrix

4.5 Random Forest

As seen in Figure 11, 74.29% accuracy is not great compared to the previous models. To achieve a better result utilizing the decision tree, random forest model is used next.

If a decision tree is built on the entire dataset, a random forest Instead of building just one decision tree, random forest randomly selects observations and features to build multiple decision trees from the dataset and averages the result.

First step is to decide the number of trees to build, then pick k instances from the training set randomly each time we build a decision tree. The class that is predicted most frequently among all the trees is the final class prediction.

The diagram below shows that generating 8 trees gives the highest accuracy (Figure 12). After selecting 8 random decision trees to achieve the best accuracy, the random forest model successfully achieves 84.08% accuracy on the test, which is significantly a improved result compared to the result of one decision tree (Figure 13).

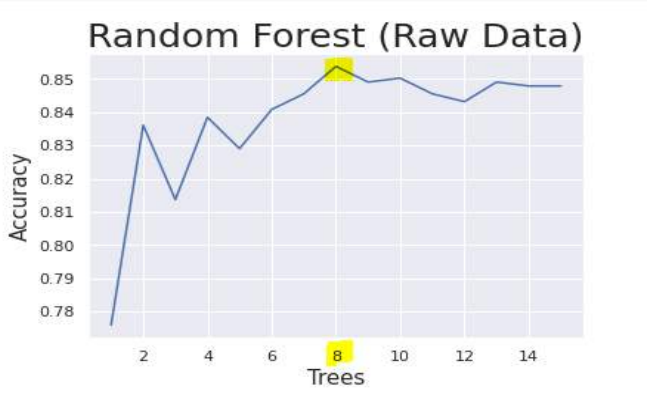

Figure 12: Random Forest Optimal # of Trees = 8

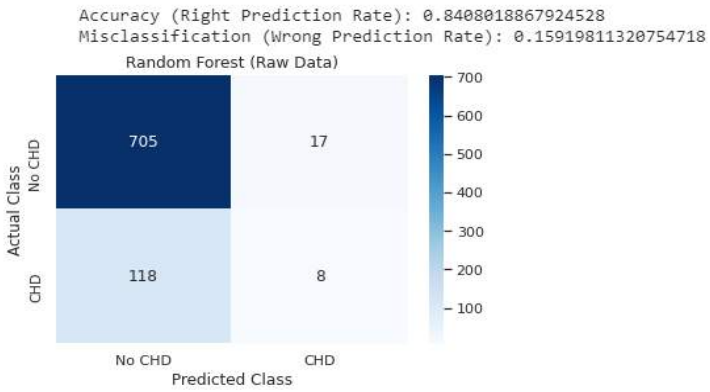

Figure 13: Random Forest Confusion Matrix

4.6 KNN

The k-nearest neighbors (k-NN) algorithm is a non-parametric method used for both classification and regression learning tasks. For k-NN classification, the input consists of k closest training instances and the output are class memberships. The algorithm finds the closest data points in the training dataset – its nearest neighbors.

The first step of the algorithm is to pick a value of k (usually small), then calculate the distance of the new instance from all the existing instances. Select the k instances in the training dataset that are closest to the new instance and predict the class for the new instance based on the majority class among the k neighbors.

As shown in Figure 14, having 6 neighbors gives the model the highest accuracy of 84.90% (Figure 15).

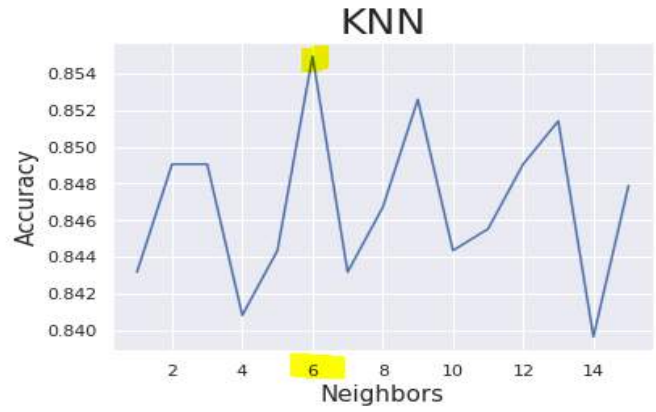

Figure 14: KNN Optimal # of Neighbors = 6

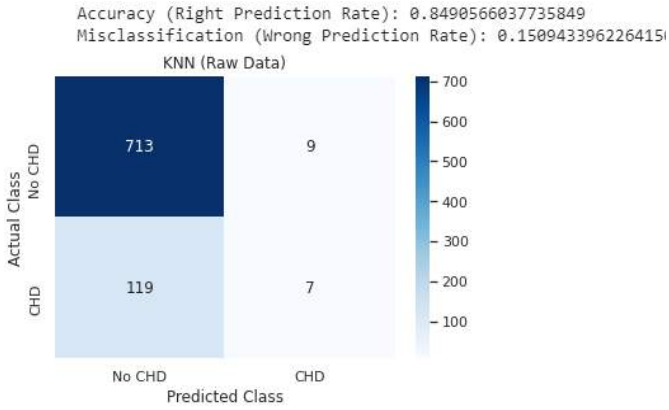

Figure 15: KNN Confusion Matrix

5 Model Testing

Testing each model is the next step to model training. The model results are derived from two different methods: feature selection and feature extraction. Figure 16 shows the testing results after feature selection for all supervised classification models. The blue bar indicates backward elimination, the orange bar indicates forward selection, and the green bar indicates stepwise. It is obvious that the Log Regression model shows the highest accuracy with the bar chart whereas the decision tree model is by far the least accurate model.

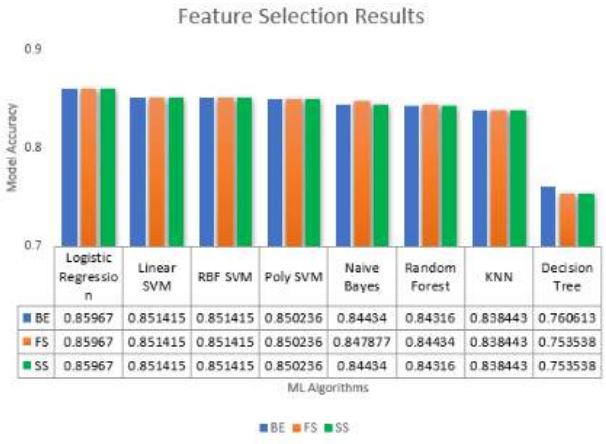

Figure 16: Feature Selection Results

Additionally, Figure 17 shows the testing results of raw data and the results of feature extraction. As a result, the winning model based on PCA is Logistic Regression (85.96 accuracy) while the winning model based on LDA and KPCA is RBF SVM (85.7% accuracy).

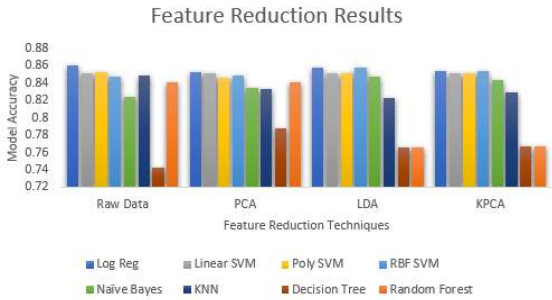

Figure 17: Feature Reduction Results

6 Model Tuning

One of the key steps in building a machine learning model is to estimate its performance on data that the model has not seen before. While the testing results are satisfactory, the results can always improve by some tuning. Model tuning is the process of tweaking hyperparameters in order to increase a model accuracy without creating overfit. In this project, K-Fold cross-validation (K-Fold CV) and Grid Search are used for hyper-parameter optimization.

K-Fold CV method splits the training dataset into K number of folds, where K-1 folds are used for the model training, and one fold is used for performance evaluation. The procedure is repeated K times for K number of models to obtain performance estimates. A good standard value for K is 10 based on empirical evidence. Thus, K is set to be 10 for this project as well. Grid Search method looks at all possible combinations of values specified for hyper-parameters and gives the best combination.

Figure 18 illustrates the hyper-parameter tuning results. Based on the hyper-parameter tuning, the winning models from the KFolds CV are Linear SVM and RBF SVM which both yielded 84.9% accuracy based on 10 folds.

On the other hand, the winning model for Grid Search are Linear SVM and RBF SVM which both yielded 85.49% accuracy based on 10 folds.

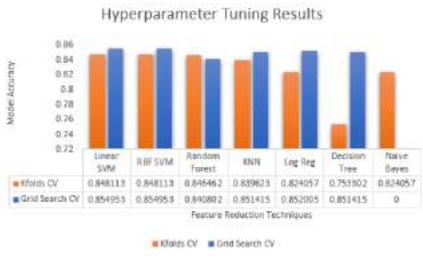

Figure 18: Hyperparameter Tuning Results

7 Conclusion

Coronary heart disease (CHD) is one of the most common types of heart disease in America. This brings some important questions: Why is it one of the most common diseases, and what are the main risk factors for developing CHD? With these questions in mind, the main objective of this project is to find the most significant risk factors for developing CHD. The initial hypothesis is that BMI is the biggest risk factor for CHD. However, the testing results have suggested otherwise.

The model testing presents different results based on the two approaches, feature selection and feature extraction. With feature selection, gender (Male), age, cigarette consumption per day (cigsPerDay), history of stroke (prevalentStroke), increased systolic blood pressure (sysBP), and high glucose level (glucose) are suggested to be prominent risk factors for CHD. With these features, the logistic regression gives the highest model accuracy of 85.97%. However, feature extraction techniques show that the RBF SVM model provides the highest accuracy with LDA and KPCA while the logistic regression provides the highest accuracy with PCA. Therefore, we reject the initial hypothesis and conclude that gender, cigsPerDay, prevalentStroke, sysBP, and glucose are the most significant risk factors for developing CHD. Furthermore, the logistic regression provides the highest model accuracy for developing CHD by feature selection and PCA, while RBF SVM model provides the highest model accuracy based on LDA and KPCA.

REFERENCES

[1] Felman, A., 2019. What to know about coronary heart disease. *MedicalNewsToday*.

[2] Shekhar, A., 2018. What is Feature Engineering for Machine Learning?. *MindOrks.*

[3] Chang, C, 2019. A Library for Support Vector Machines.

[4] Rish, Irina, 2001. An empirical study of the naive bayes classifier. *IJCAI 2001 workshop on empirical methods in artificial intelligence*, (Aug, 2001), 41-46. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.330.2788

[5] Prediction of Coronary Heart Disease using Supervised Machine Learning Algorithms. (March 2001). Retrieved March,26 2020 from https://www.ukessays.com/essays/computer-science/prediction-of-coronary-heart-disease-using-supervised-machine-learning-algorithms.php

*How to reference in ACM format
https://www.citethisforme.com/guides/association-for-computing-machinery/how-to-cite-a-website

DATA USED

framingham.csv

PYTHON SOURCE CODE

1. Model Studies (Raw Data).pdf

2.1 Model Studies (After BE).pdf

2.2 Model Studies (After-FS).pdf

2.3 Model Studies (After-SWS).pdf

3.1 Model Studies (After PCA).pdf

3.2 Model Studies (After LDA).pdf

3.3 Model Studies (After KPCA).pdf

4.1 Model Tuning (Kfolds CV).pdf

4.2 Model Tuning (Grid Search CV).pdf