Predicting Cardiovascular Disease

Adam M. Lang

IBM Data Science Professional Certificate - Coursera

Course: Applied Data Science Capstone

September 20, 2020

Introduction

Cardiovascular Disease (CVD) remains the leading cause of morbidity and mortality despite 4 decades of declining mortality rates in the United States. CVD remains the leading cause of morbidity and mortality for women in the United States (Cho et al. 2020). The age-adjusted death rate attributable to CVD, based on 2017 data, is 219.4 per 100,000. On average, someone dies of CVD every 37 seconds in the U.S. There are 2,353 deaths from CVD each day, based on 2017 data (AHA, 2020). Around 17.5 million people die each year from cardiovascular diseases (CVDs), an estimated 31% of all deaths worldwide. This statistic is expected to grow to more than 23.6 million by 2030 (Hajar, 2016).

CVDs are a group of disorders of the heart and blood vessels, and they include coronary heart disease, cerebrovascular disease, peripheral arterial disease, and rheumatic heart disease, congenital heart disease, deep vein thrombosis, and pulmonary embolism. The cause of CVD is usually the presence of a combination of risk factors, such as tobacco use, unhealthy diet and obesity, physical inactivity and harmful use of alcohol, hypertension, diabetes, and hyperlipidemia (Hajar, 2016). Collectively, CVDs and their risk factors are a gateway to other diseases. It is therefore vital to understand and research CVD risk factors and how they predict the development of heart disease.

Risk Factors for Heart disease are broken down into "Modifiable" and "Nonmodifiable". They are as follows:

Modifiable

- **Smoking or exposure to environmental tobacco smoke:** The most preventable risk factor. Smokers have more than twice the risk of developing cardiovascular disease. On average, smoking costs 13 years of life to a male smoker and 14 years to a female smoker. Exposure to smoke — secondhand smoking — increases the risk even for non-smokers.

- **Obesity:** One of the highest risk factors for CVD. Obesity is defined as having a body mass index (BMI) $\geq$30 kg/m^2. Obesity is often subdivided into classes (Class I: BMI=30.0–34.9, Class II: BMI=35.0–39.9, Class III: BMI $\geq$40.0) to further stratify health risk. Using BMI-based diagnostic criteria, 39.8% of the US population meets the definition of obesity with 7.7% having class III obesity or severe obesity, defined as a BMI$\geq$40.0 kg/m^2 Obesity is a strong independent predictor of CVD even in the absence of other risk factors, however, interestingly after onset of CVD the relationship between higher BMI and clinical outcomes is not linear. BMI and obesity should however be carefully considered with respect to an individuals' amount of lean mass and fat mass (Carbone et al. 2019).

- **Sedentary lifestyle (not enough physical activity):** Sedentary behavior and physical inactivity are among the leading modifiable risk factors worldwide for CVD and all-cause mortality. Although the American Heart Association, the American College of Cardiology, and the American College of Sports Medicine, among other leading organizations, have emphasized that sedentary behavior (SB) and physical inactivity (PI) are major modifiable cardiovascular disease (CVD) risk factors, a sizable percentage of the United States and

worldwide population still present with high levels of SB/PI and low levels of physical activity (PA). Recently, a major emphasis has been directed at making health promotion a priority, including the promotion of PA and exercise training (ET) and improving levels of cardiorespiratory fitness (CRF) in the United States and worldwide in efforts to prevent chronic diseases, especially CVD (Lavie et al. 2019).

- **Diabetes:** Whether Type 1 or Type 2 the risk for CVD is the same. A fasting plasma glucose above 125mg/dL is at risk for Diabetes. Diabetes and CVD share similar risk factors – high cholesterol, hypertension, and obesity.

- **High cholesterol or abnormal blood lipids (fats):** Controlling levels of LDL (bad cholesterol), HDL (good cholesterol), total cholesterol and triglycerides (most common body fat) will reduce risk of CVD. Total cholesterol should be less than 200mg/dL. Triglycerides should be less than 150mg/dL. HDL levels above 60mg/dL and LDL levels below 70mg/dL are ideal to prevent CVD.

- **Hypertension (high blood pressure):** For persons older than 50, systolic blood pressure is more important than diastolic blood pressure as a cardiovascular disease risk factor. Starting at 115/75 mmHg, cardiovascular disease risk doubles with each increment of 20/10 mmHg throughout the blood pressure range. Normal blood pressure is 120/80 and for each increment of systolic pressure over 20 and diastolic pressure over 10 it increases your stage of developing hypertension.

Nonmodifiable

- **Gender:** Women tend to develop CVD 10 years later in life than Men, but outcome is often worse.

- **Age older than 50 years:** 80% of people who die from CVD are 65 years or older.

- **Family history of heart disease:** A primary risk is a relative who developed heart disease before age 55.

One controversial risk factor not discussed above is alcohol intake. Some studies have shown that moderate alcohol intake can reduce your risk of developing and even dying from CVD. There is some evidence that moderate alcohol intake will increase good levels of HDL cholesterol. It is important to understand that “moderate drinking” is defined as an average of one drink per day for women and one or two for men. A drink might be less than you think: 12 ounces of beer, 4 ounces of wine or 1.5 ounces of 80-proof spirits. It is however when excessive alcohol intake can lead to hypertension, heart failure or even stroke. Excessive drinking can also lead to cardiomyopathy a dangerous disorder that enlarges the heart muscle leading to many issues among the CVDs (Johns Hopkins, 2020).

Business Problem

Heart disease continues to be studied worldwide. In the United States our understanding of heart disease is due to the work of the Framingham Heart Study (FHS). The study began in 1948 with 5209 adult subjects from Framingham and is now on its third generation of participants. Much of our appreciation of the pathophysiology of heart disease came from the results of studies from the FHS. It established the traditional risk factors, such as high blood pressure, diabetes, and cigarette smoking for coronary heart disease. Framingham also spearheaded the study of chronic noninfectious diseases in the USA and introduced preventive medicine.

It is obvious that CVD is not only a major cause of morbidity and mortality in our society but influences the development of other acute and chronic diseases and how we continue to evolve our approach to preventative medicine and lifestyle improvements. It is therefore prudent to continue to work with cardiovascular risk factor data and try to predict what leads to CVD developing as risk factors and people evolve. The goal of this project is to perform exploratory data analysis on cardiovascular risk factor data and use machine learning algorithms to try and predict which risk factors lead to the development of heart disease.

Data Understanding

This dataset is entitled the "Cardiovascular Disease dataset" was obtained from Kaggle and is ideal for performing data mining and machine learning activities to predict Cardiovascular Disease (CVD). The dataset contains 70,000 records, 11 features, and 1 target variable (0 having no CVD, 1 having CVD). There are 3 types of input features for each variable: Objective (factual information), Examination (results of medical exam), and Subjective (information given by the patient). All dataset values were entered after a patient's medical exam (Ulianova, 2018).

There 3 types of input features:

Objective: factual information;
Examination: results of medical examination;
Subjective: information given by the patient.

Features:

Age | Objective Feature | age | int (days)
Height | Objective Feature | height | int (cm) |
Weight | Objective Feature | weight | float (kg) |
Gender | Objective Feature | gender | categorical code |
Systolic blood pressure | Examination Feature | ap_hi | int |
Diastolic blood pressure | Examination Feature | ap_lo | int |
Cholesterol | Examination Feature | cholesterol | 1: normal, 2: above normal, 3: well above normal |
Glucose | Examination Feature | gluc | 1: normal, 2: above normal, 3: well above normal |
Smoking | Subjective Feature | smoke | binary |

Alcohol intake | Subjective Feature | alco | binary |
Physical activity | Subjective Feature | active | binary |

Presence or absence of cardiovascular disease | Target Variable | cardio | binary |

We can see there are 5 numeric or continuous variables: Age, Height, Weight. Systolic blood pressure and Diastolic blood pressure. There are 5 categorical variables that are labeled based on level (1-3) or presence of the factor or not: Cholesterol, Glucose, Smoking, Alcohol intake and Physical activity. We already discussed above the final variable is presence or absence of CVD. These variables fit in well with the aforementioned "Modifiable" and "Non-Modifiable" risk factors. The only variable not mentioned from those risk factors is BMI or Obesity and we will have to create that variable during exploratory analysis.

We will perform exploratory data analysis on the data and see what data pre-processing or cleaning needs to be done in addition to any interesting findings. We will then prepare the data for machine learning using appropriate techniques such as normalization and feature scaling. The goal is to see which continuous and/or categorical variables are better predictors of cardiovascular disease and how well we can predict the disease based on these variables.

Methodology

a. Exploratory Data Analysis

I began exploratory data analysis and started to find some interesting things about the dataset features. I first saw that there are 70,000 rows and 13 columns. We can see the data frame below:

Exploratory Data Analysis

```
In [198]: #inspect dataset first 5 rows
          cardio.head(5)
```

Out[198]:

	id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active	cardio
0	0	18393	2	168	62.0	110	80	1	1	0	0	1	0
1	1	20228	1	156	85.0	140	90	3	1	0	0	1	1
2	2	18857	1	165	64.0	130	70	3	1	0	0	0	1
3	3	17623	2	169	82.0	150	100	1	1	0	0	1	1
4	4	17474	1	156	56.0	100	60	1	1	0	0	0	0

Fig.1. Exploratory Data Analysis

Right away we can tell that there are some interesting features that need to be dealt with from the dataset. The age is in number of days which will need to be changed to years to fully understand and work with. I am not too concerned about height which is in cm and weight which is in kg as this can be useful to calculate the body mass index (BMI) which is an important risk

factor that I will add as a separate column to the dataset for predicting CVD. The columns "api_hi" and "ap_lo" correspond to "systolic blood pressure" and "diastolic blood pressure." I am going to change these headings so we can better understand these values. The final variables are cholesterol, glucose, smoking, alcohol intake, activity, and the target variable cardio which 0 stands for no disease and 1 for having CVD.

The data types are seen below:

Examine Variables

```
In [204]: cardio.dtypes

Out[204]: Age              int64
          Gender           int64
          Height           int64
          Weight         float64
          Systolic_BP      int64
          Diastolic_BP     int64
          Cholesterol      int64
          Glucose          int64
          Smoke            int64
          Alcohol          int64
          Active           int64
          Cardio           int64
          dtype: object
```

Fig. 2. Data Types

What we can see is that all variables are numeric. However, not all variables are continuous as there are categorical variables present which have been encoded for analysis. The numeric variables are: Age, Height, Weight, Systolic_BP, and Diastolic_BP. The categorical variables are: Gender, Cholesterol, Glucose, Smoke, Alcohol, Active, and Cardio. Looking closer at these variables Cholesterol and Glucose are scaled from 1 to 3 with 1 being "Normal", 2 being "Above Normal" and 3 being "Well Above Normal". Smoke, Alcohol, Active, and Cardio are all 0 or 1 for not present vs. present. Since the variables are already encoded, we will not have to do much for preparing them for machine learning.

In terms of data integrity, there are no missing values but there were 24 duplicate records as seen in this data frame from my Jupyter notebook:

- Let's take a look at the duplicate variables first

```
In [207]: duplicated = cardio[cardio.duplicated(keep=False)]
          duplicated = duplicated.sort_values(by=['Age', "Gender", "Height"], ascending= False)
          # I sorted the values to see duplication clearly

          duplicated.head(24) # Show all duplicated values
```

Out[207]:

	Age	Gender	Height	Weight	Systolic_BP	Diastolic_BP	Cholesterol	Glucose	Smoke	Alcohol	Active	Cardio
2677	60.444773	1	175	69.0	120	80	1	1	0	0	1	1
45748	60.444773	1	175	69.0	120	80	1	1	0	0	1	1
1568	60.083369	1	165	60.0	120	80	1	1	0	0	1	0
48917	60.083369	1	165	60.0	120	80	1	1	0	0	1	0
40301	60.077893	1	165	65.0	120	80	1	1	0	0	1	1
52552	60.077893	1	165	65.0	120	80	1	1	0	0	1	1
8190	59.626139	1	160	58.0	120	80	1	1	0	0	1	0
65622	59.626139	1	160	58.0	120	80	1	1	0	0	1	0
21871	58.262661	1	165	65.0	120	80	1	1	0	0	1	0
45125	58.262661	1	165	65.0	120	80	1	1	0	0	1	0

Fig. 3. We can see 24 duplicate records above. I ended up dropping all 24 duplicate records from the dataset.

Next, I evaluated the variables. First, I looked at the Age variable.

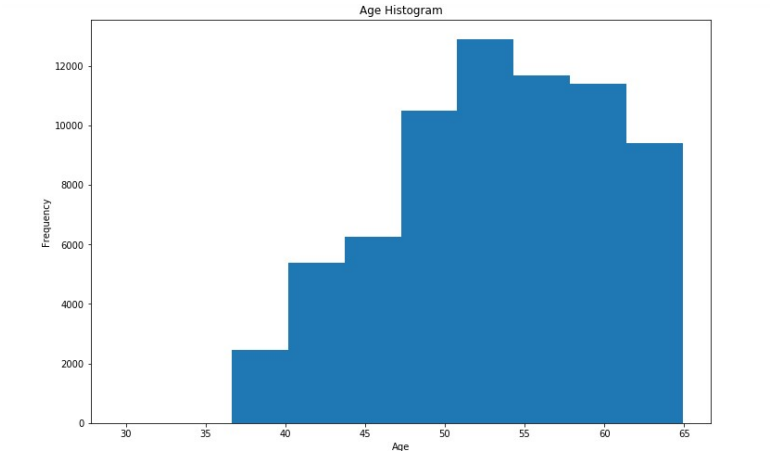


Fig. 4. We can see that the age distribution is predominantly around age 50-55 but all below age 65. I decided to bin the ages to make the data more manageable for analysis.

Now let's create a new column to describe the decade of age groups.

```
In [214]: cardio['Age_By_Decade'] = pd.cut(x=cardio['Age'], bins=[30, 39, 49, 59,69], labels=['30s', '40s', '50s','60s'])
          cardio.head(5)
```

Out[214]:

	Age	Gender	Height	Weight	Systolic_BP	Diastolic_BP	Cholesterol	Glucose	Smoke	Alcohol	Active	Cardio	Age_Bins	Age_By_Decade
0	50.358324	2	168	62.0	110	80	1	1	0	0	1	0	(49, 59]	50s
1	55.382383	1	156	85.0	140	90	3	1	0	0	1	1	(49, 59]	50s
2	51.628712	1	165	64.0	130	70	3	1	0	0	0	1	(49, 59]	50s
3	48.250135	2	169	82.0	150	100	1	1	0	0	1	1	(39, 49]	40s
4	47.842187	1	156	56.0	100	60	1	1	0	0	0	0	(39, 49]	40s

Fig. 5. Age Bins were created. Of note there were a few outlying individuals who were 29 years of age and these were included in the 30-39 age group as I rounded the values up.

This made it easier to see how many age groups we are working with.

```
Out[21]: (49, 59]    35370
         (39, 49]    17982
         (59, 69]    16620
         (29, 39]        4
         Name: Age_Bins, dtype: int64
```

```
[24]: cardio['Age_Bins'].value_counts().plot(kind='bar',figsize=(8,4))
      plt.title("Age Bins Distribution")
      plt.xlabel("Age Bins")
      plt.ylabel("Frequency of Age Groups")
      plt.show();
```

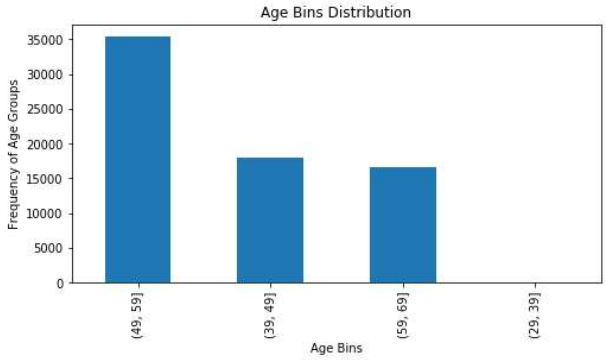


Fig. 6. We can see though that the age bins are not evenly distributed with almost twice the number of 50-59-year old's than 60-69-year old's. This could prove controversial as the sample sizes are not even for analysis.

The next thing I did was evaluate all categorical variables. It was interesting to me that there were more "Normal" (Group 1) findings for Glucose, Cholesterol; more smokers and non-drinkers; and more active individuals overall in the dataset. This makes it interesting to automatically assume we have patients with heart disease as those alone are high risk factors. I did however compare the dataset based on those with and without CVD and saw that those with CVD do have higher glucose and cholesterol levels as seen below:

I next decided to look at a regplot in seaborn to see if there were any interesting correlations early on that I could investigate further with statistical analysis. The most interesting correlation was this:

```
In [26]: sns.regplot(x="Weight",y="Height",data=cardio).set(title="Weight vs. Height");
```

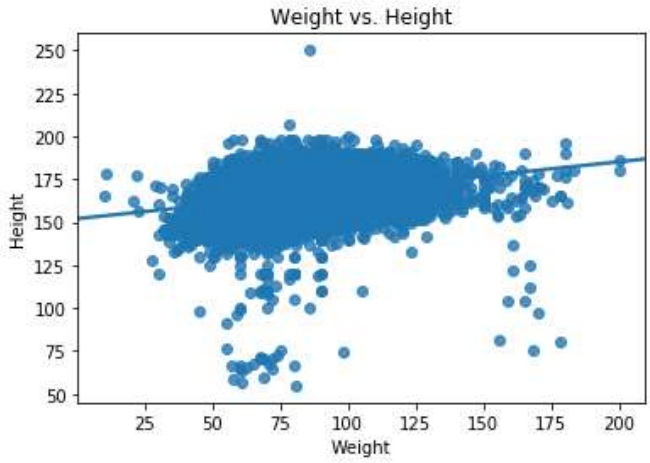


Fig. 7. Weight vs. Height regplot in seaborn. There appears to be a somewhat linear relationship between the 2 variables. I decided to investigate this further by making a body mass index category but will explain this later in this report.

b. Data Visualization – Categorical variables

Since the first part of this report is visualization of the numeric variables, we will now go over the unique features of the categorical variables. The first categorical variable was Glucose.

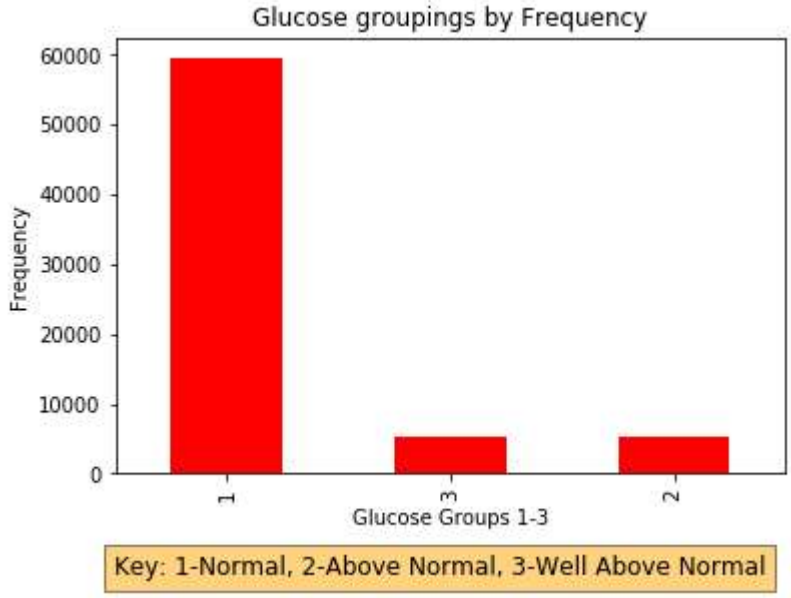


Fig. 8. Glucose categorical variable.

It was rather interesting that this data set contained majority of people with normal glucose levels and are not diabetic. Considering that high glucose levels and diabetes are a major risk factor for CVD not having a lot of patients in this dataset with diabetes will make prediction more difficult.

I did the same thing with Cholesterol, Smoking, Alcohol and Activity level. It was interesting to me that there were more "Normal" (Group 1) findings for Glucose, Cholesterol; more smokers and non-drinkers; and more active individuals overall in the dataset. This makes it interesting to automatically assume we have patients with heart disease as those alone are high risk factors. I did however compare the dataset based on those with and without CVD and saw that **those with CVD do have higher glucose and cholesterol levels as seen below:**

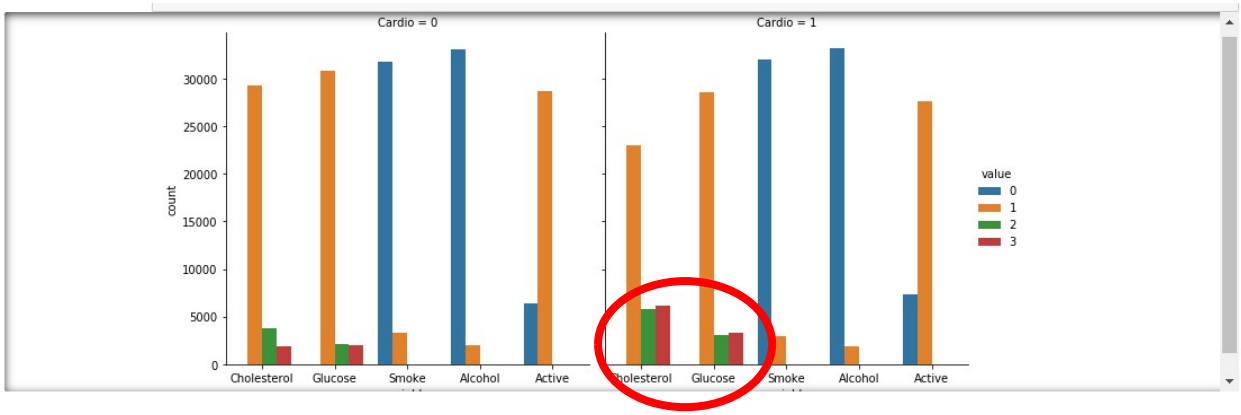


Fig. 9. Higher Cholesterol and Glucose levels are seen in those with CVD within the dataset. This proves my earlier statement wrong and this may make the prediction more robust now.

As we can see the dataset does reveal there are more people with cardiovascular disease that have higher glucose and cholesterol levels as compared to those without CVD. Every other variable is virtually the same and normal.

c. Outlier Detection

Another major issue with this dataset is there are significant outliers. We can see this in the data frame below:

	Age	Gender	Height	Weight	Systolic_BP	Diastolic_BP	6
count	69976.000000	69976.000000	69976.000000	69976.000000	69976.000000	69976.000000	
mean	53.304175	1.349648	164.359152	74.208519	128.820453	96.636261	
std	6.755442	0.476862	8.211218	14.397211	154.037729	188.504581	
min	29.563920	1.000000	55.000000	10.000000	-150.000000	-70.000000	
25%	48.362389	1.000000	159.000000	65.000000	120.000000	80.000000	
50%	53.944982	1.000000	165.000000	72.000000	120.000000	80.000000	
75%	58.391343	2.000000	170.000000	82.000000	140.000000	90.000000	
max	64.923989	2.000000	250.000000	200.000000	16020.000000	11000.000000	

Fig. 10. We can see the outliers in the red box in the data frame above.

The values outlined in red are maximums for the respective columns. Most concerning is that the highest Systolic blood pressure is 1,6020 and the highest Diastolic blood pressure is 11,000 both of which are not possible as the highest recorded blood pressure on planet earth was in an exercise study at 370/360. A doctor will diagnose you with hypertension if your blood pressure is above 120/80 and is considered a “hypertensive crisis” if it is above 200/120mm/Hg (Narloch et al. 1995 and Unger et al. 2020).

To deal with these extreme outliers I had to perform data normalization to better quantify the extremeness of the values. I did this by performing scaling of the data. I was then able to quantify appropriate quartiles for the upper levels of systolic and diastolic blood pressure as 250 and 200, respectively.

As for the weight outlier this is 200kg and corresponds to about 440lbs which may be an outlier but is still important consideration and reasonable to consider when diagnosing obesity which is a major risk factor for CVD so I left it as is. I did however remove the height of 250cm which corresponds to over 8 feet tall which is a definite outlier.

d. Descriptive/Inferential Statistics

The next thing I did to the data was create specific columns for obesity classification and BMI. This can be seen below:

nder	Height	Weight	Systolic_BP	Diastolic_BP	Cholesterol	Glucose	Smoke	Alcohol	Active	Cardio	Age_Bins	Age_By_Decade	BMI	Obesity
2	168	62.0	110	80	1	1	0	0	1	0	(49, 59]	50s	21.967120	Healthy
1	156	85.0	140	90	3	1	0	0	1	1	(49, 59]	50s	34.927679	Obesity
1	165	64.0	130	70	3	1	0	0	0	1	(49, 59]	50s	23.507805	Healthy
2	169	82.0	150	100	1	1	0	0	1	1	(39, 49]	40s	28.710479	Overweight
1	156	56.0	100	60	1	1	0	0	0	0	(39, 49]	40s	23.011177	Healthy

Fig. 11. BMI and Obesity columns created.

You can also see that I created columns for Age bins and Age classification by decade. This made it easier to visualize the data and classify each group. I was then able to quantify this with a bar graph.

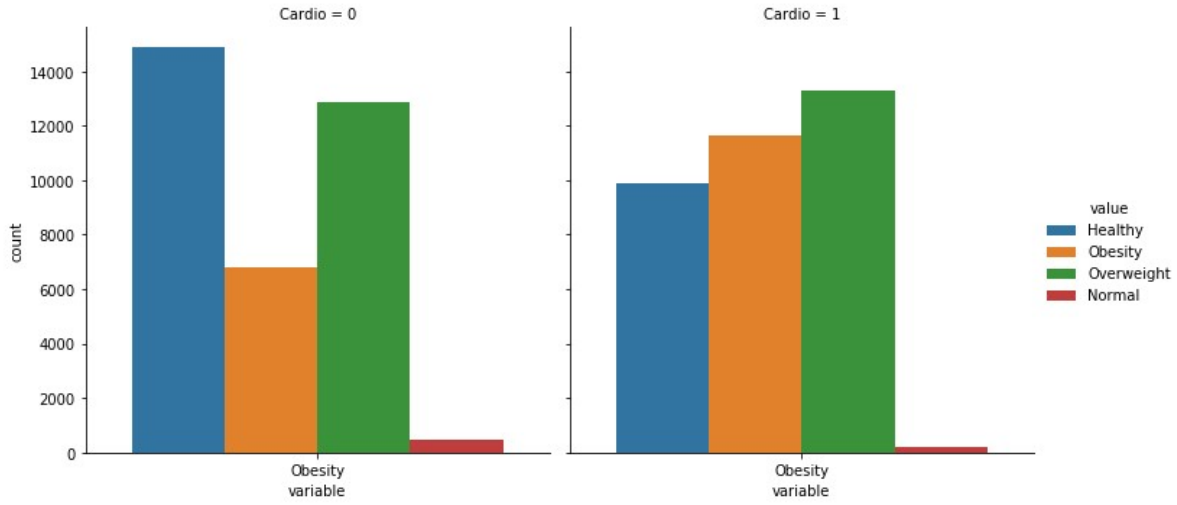


Fig. 12. We can now see that Obesity had more patients in the dataset that have CVD (1) than do not.

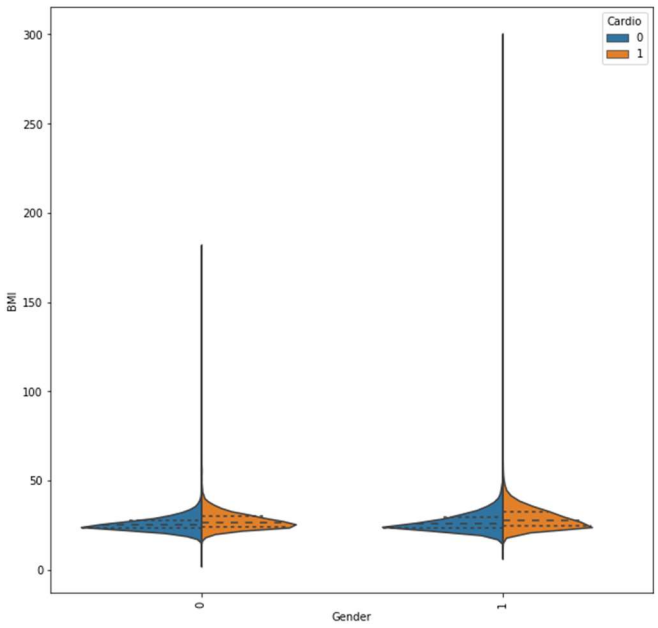


Fig. 13. Next, I created a violin plot to examine the quartiles for Male vs. Female and their BMI. We can clearly see that the BMI median and quartiles are higher and more robust for those individuals with cardiovascular disease (1) and that are Male (1).

I then decided to look closer at the numerical variables and see what correlations there are. I ran a correlation function which would be like a Pearson correlation and turned this into a heatmap.

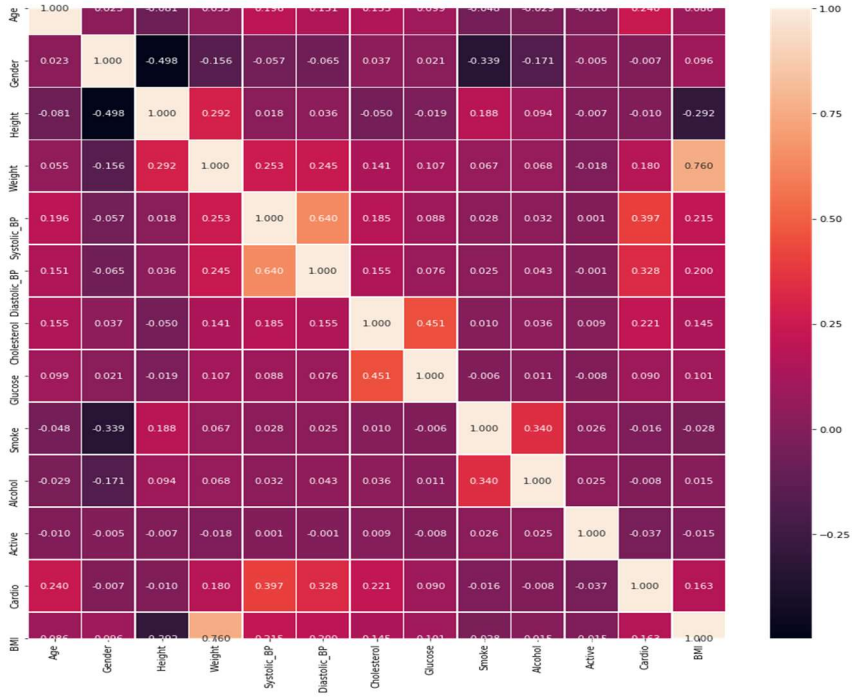


Fig. 14. Correlations on the heatmap.

We can see that the strongest correlations appear to be:

- Weight and BMI (0.760)
- Systolic_BP and Diastolic_BP (0.640)
- Glucose and Cholesterol (0.451)

Since I had these correlations, I did not see it necessary to perform any further testing. I did attempt Pearson correlation testing with P values, but the P values were 0.0 and I tried to fit a regplot for each variable. All this did was prove the strongest correlation is seen between weight and BMI which is already proven by the 0.760 value being closest to 1.0.

e. Data Pre-Processing

Data Pre-processing was performed next and this was primarily to perform label encoding for the newly created categorical variables: Obesity, Age_By_Decade, and Age_Bins. I think it is debatable as to whether you can say the categories for each group are nominal or ordinal. I decided to go with they are ordinal as each category represents a higher level on the spectrum. This again is highly debatable and much has been written about the importance of label encoding vs. One-hot encoding and how label encoding can create a 'hierarchy' effect which can skew predictive modeling (Srinidhi, 2018). I started off by trying the label encoding. This is how I encoded the variables:

1. Obesity: 1-Normal, 2-Healthy, 3-Overweight, 4-Obesity
2. Age_By_Decade: 1-30s, 2-40s, 3-50s, 4-60s

The decision was made to drop the Age_Bins column as this was not necessary to perform predictive modeling and was only for purposes of exploratory analysis.

f. Feature Selection

The features that were selected for machine learning were basically all categorical and continuous variables. I did consider performing advanced feature selection using Lasso Regression which is a form of penalized regression that performs regularization of the coefficients of each feature shrinking some coefficients to zero and thus selecting the optimal features for the model (Dubey, 2019). However, since I was going to be using a Random Forest Model, I figured that would select the most important features for me anyways. I also considered that each variable did have its importance to predicting CVD and thus should be tested in the models.

g. Model Training

For model training I used the standard train/test split imported from sci-kit learn. I used the test size as 0.30 or 30% so we would have 70% of the data used for training and 30% used for testing. I then established the size of the x training set and the y training set as seen below.

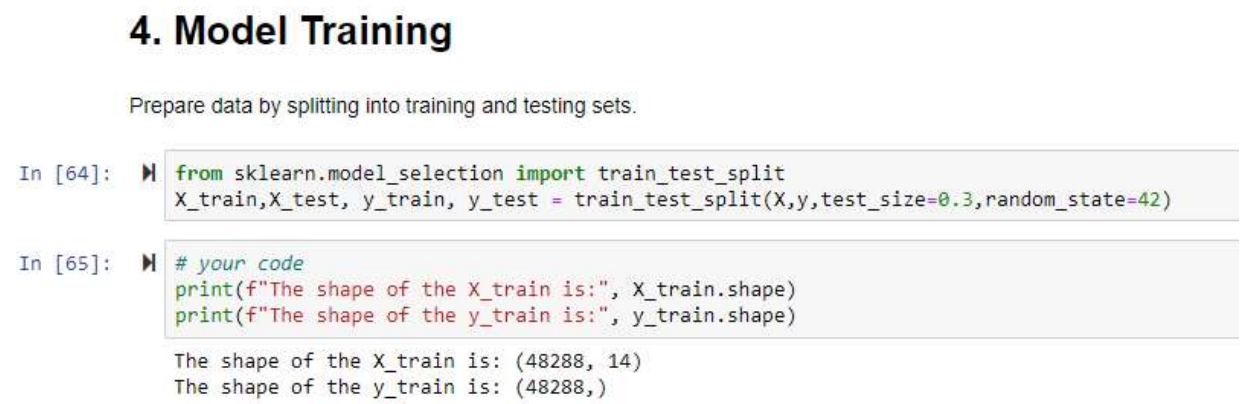


Fig. 15. We can see train test split was initiated.

The most important part of model training was feature scaling. I used this to normalize the data and better prepare the diverse data for the machine learning algorithms. Standardization is a transformation that **centers the data by removing the mean value of each feature and then scale it by dividing (non-constant) features by their standard deviation**. After standardizing data, the mean will be zero and the standard deviation one.

Standardization can drastically improve the performance of our models. For instance, many elements used in the objective function of a learning algorithm assume that all features are centered around zero and have variance in the same order. If a feature has a variance that is orders of magnitude larger than others, it might dominate the objective function and make the estimator unable to learn from other features correctly as expected (Van Dorpe, 2018). Although I removed the more drastic **outliers**, there are still some that remain such as in the weight column, so scaling using the mean and standard deviation of the data is likely to not work very well. In these cases,

you can use the *RobustScaler* is a good idea. **It removes the median and scales the data according to the quantile range**. By default, the scaler uses the Inter Quartile Range (IQR), which is the range between the 1st quartile and the 3rd quartile (Van Dorpe, 2018). I also thought that having an even mix of categorical and continuous variables would make the robust scaler more appropriate to handle the data.

```
In [69]: X_train.head()
```

Out[69]:

	Age	Gender	Height	Weight	Systolic_BP	Diastolic_BP	Cholesterol	Glucose	Smoke	Alcohol	Active	Age_By_Decade	BMI	Obesity
0	1.051394	-1.0	0.818182	-0.294118	0.0	0.0	0.0	2.0	1.0	0.0	0.0	1.0	-0.675643	-0.5
1	0.394204	0.0	-0.636364	-0.588235	-1.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.242062	-0.5
2	-0.837343	0.0	-0.636364	-1.000000	0.0	0.0	0.0	0.0	1.0	1.0	0.0	-1.0	-0.691348	-0.5
3	-0.533898	0.0	0.454545	0.000000	-0.5	-1.0	0.0	0.0	0.0	0.0	0.0	-1.0	-0.229607	0.0
4	-0.435210	0.0	0.454545	-0.411765	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	-0.617703	-0.5

Fig. 16. Data after Robust Scaling was performed.

h. Machine Learning Models

The decision was made to use multiple machine learning models. The models that I selected included: Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Random Forest (RF) and Naïve Bayes (NB). Using more than one model is called "Ensemble Methods". The idea is to combine the outputs of several models to optimize prediction performance with the best model to get the best results (Singh, 2018). Although there are multiple ensembling techniques such as max voting, averaging, weighted average, stacking, blending, bagging, boosting, and many more complex algorithms based on bagging and boosting (i.e. XGB), we will use the max voting technique. Essentially, we take the model(s) with the best performance and use those for our results evaluation.

Machine learning involves the learning of hidden patterns within a dataset (data mining) and then using these patterns to classify or predict an outcome or event. This problem of predicting whether a patient does or does not have cardiovascular disease is a dichotomous categorical dependent variable and thus a classification problem – the person either has it or they do not. We will be using "Supervised Learning" which involves predetermined output attributes besides the use of input attributes. The algorithms attempt to predict and classify the predetermined attribute, and their accuracies and misclassification alongside other performance measures is dependent on the counts of the predetermined attribute correctly predicted or classified or otherwise. It is also important to note the learning process stops when the algorithm achieves an acceptable level of performance (Kotsiantis, 2007).

So why are we going to use the 4 algorithms I discussed above? Here is why:

Support Vector Machines: the objective of an SVM algorithm is to find a hyperplane in an N-dimensional space (N – the number of features) that distinctly classifies the patient has having CVD or not (Gandhi, 2018). The algorithm's objective is to find the plane that has the maximal

margin (maximum distance between both classes). By maximizing the margin distance there is re-enforcement, so all future data points are classified with confidence. SVM works well when there is a clearly defined margin of separation between classes which in theory there should be here, each person either has CVD or they do not. As with most algorithms SVM does not perform well on exceptionally large datasets or with noise such as heavily overlapping classes which I believe we will not have here and so this should be a highly effective classifier.

K-Nearest Neighbors: this model classifies data points based on points that are most similar. It uses test data to make an "educated guess" on what each unclassified point should be classified as. The great thing about KNN is that it does not make assumptions about the data which will be great for classifying CVD based on the number of features that we have selected. The only problems with KNN are that accuracy depends on data quality, it must find an optimal k value (number of nearest neighbors) to work, and it is poor at classifying data points where the boundary is poorly defined (Schott, 2019). I do think this algorithm will work well as there is a good mix of categorical and continuous data points and many different "profiles" of each person that could potentially have CVD and it should be able to find the most similar groupings.

Random Forest: this classifier uses multiple decision trees and performs "bagging" or random sampling at each node of the decision trees to yield classification results. We are also able to obtain "variable importance' with this algorithm which will tell us which variables are weighted more important than the others for predicting CVD (Williams, 2011). A random forest can overfit a dataset that is too noisy, so this is something we need to watch out for.

Naïve Bayes: this is based on Bayes' theorem. There are two assumptions, one is that we consider all predictors to be independent of each other, and the second is that all predictors have an equal effect on the outcome. A Gaussian Naïve Bayes will be used assuming a normal distribution with the continuous data (Gandhi, 2018). Although a Multinomial Naïve Bayes algorithm is usually used for categorical data, it is used most often with text data and we have all numerical values. I believe the NB classifier will be excellent for this prediction task as we can assume each risk factor is independent of one another for predicting CVD.

There are 2 major disadvantages of the NB classifier. The first is that we cannot always assume all predictors are independent of one another, this is awfully hard to come by. The second is called "Zero Frequency". This is when a categorical variable has a category in test data set, which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction (Gandhi, 2018).

i. Cross Validation

We will use Cross Validation to test the best machine learning models on data it has not seen before. Why do we use cross validation? The bias variance trade-off is why we use cross validation. Usually, complex models have small bias and large variance, while simple models have large bias and small variance. We are looking for practically useful trade-offs between bias and variance. It is difficult to get an adequate tradeoff between bias and variance and assess model accuracy using the training and test set since it is split 70/30. We would need exceptionally large training and test sets to get a good enough trade off. Therefore, we use cross

validation which creates k-folds, in this case 10 folds, so the model has time to see an adequate amount of data in a fair manner. If we just relied on our test set of the original model that is not enough data to make a fair outcome assessment (Krstajic et al. 2014).

j. Grid Search

Grid search is the process of performing hyper parameter tuning to determine the optimal values for a given model. This is significant as the performance of the entire model is based on the hyper parameter values specified. A machine learning model has multiple parameters that are not trained by the training set. These parameters control the actual accuracy of the model. Therefore, hyperparameters are important. For example, in our best model so far, the SVM, there are several hyperparameters that we did not specify in the original model which should be considered. They include:

- **C:** This is a regularization parameter
- **Kernel:** We can set the kernel parameter to linear, poly, rbf, sigmoid, precomputed or provide our own callable.
- **Degree:** We can pass in a custom degree to support the poly kernel parameter.
- **Gamma:** This is the coefficient for rbf, poly and sigmoid kernel parameter.
- **Max_Iter:** It is the maximum number of iterations for the solver.

There is a library within sklearn called **"GridSearchCV"** that we can use to tune hyperparameters on multiple models. Let us give it a try to optimize our best model(s).

Results

The most accurate model was the SVM and the least accurate model was the NB. However, as we can see in the chart below all accuracy scores were within 3 points from min to max.

	Accuracy Score
SVM	0.736120
KNN	0.731336
Random forest	0.715680
Naive bayes	0.705098

Fig. 17. Accuracy scores for all models.

The Jaccard Index and F1-scores were run to further test model efficiency. The Jaccard Index, also known as the Jaccard similarity coefficient, is a statistic used in understanding the similarities between sample sets. The measurement emphasizes similarity between finite sample sets and is formally defined as the size of the intersection divided by the size of the union of the sample sets (Glen, 2016). The F-score, also called the F1-score, is a measure of a model's accuracy on a dataset. It is used to evaluate binary classification systems, which classify examples into 'positive' or 'negative'. The F-score is a way of combining the precision and

recall of the model, and it is defined as the harmonic mean of the model's precision and recall. The results were as follows:

	Jaccard Index	F1-score
SVM	0.74	0.74
KNN	0.73	0.73
Random Forest	0.72	0.72
Naive Bayes	0.71	0.70

Fig. 18. Jaccard Index and F1-scores.

We an see again that the SVM model has the best score for both tests. This implies that there is moderate similarity between the "sets" or groupings of having CVD or not as the score was 74%. The F1-score of 0.74 shows that the accuracy was similar to the accuracy score and has a moderate harmonic mean between the precision and recall. Again, clarifying the model had decent accuracy.

We will now go into an analysis of each machine learning model results and what this tells us overall for the classification task of this project.

a. Random Forest

The variable importance is based upon the Gini Impurity Index. Each Decision Tree is a set of internal nodes and leaves. In the internal node, the selected feature is used to make decision how to divide the data set into two separate sets with similar responses within. The features for internal nodes are selected with some criterion, which for classification tasks can be Gini impurity or Information gain, and for regression is variance reduction. We can measure how each feature decrease the impurity of the split (the feature with highest decrease is selected for internal node). For each feature we can collect how on average it decreases the impurity. The average over all trees in the forest is the measure of the feature importance. The drawbacks of the method are the tendency to prefer (select as important) numerical features and categorical features with high cardinality. What is more, in the case of correlated features it can select one of the features and neglect the importance of the second one (which can lead to wrong conclusions) (Ptonski, 2020).

	Rank	Variable
0	9	Age
1	8	Gender
2	13	Height
3	10	Weight
4	1	Systolic_BP
5	7	Diastolic_BP
6	11	Cholesterol
7	6	Glucose
8	5	Smoke
9	2	Alcohol
10	3	Active
11	12	Age_By_Decade
12	4	BMI
13	0	Obesity

Fig. 19. Random Forest Variable Importance. The higher the Gini Index number the better. So, in this case the topmost important features are:

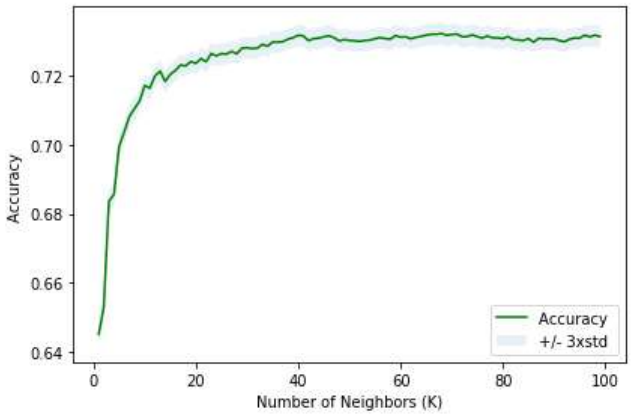
- Height
- Age_By_Decade
- Cholesterol
- Weight
- Age
- Gender

This makes sense as all are top modifiable risk factors for cardiovascular disease.

The next result of the Random Forest that is important to report is the "Out of Bag Error Estimate". This tells us how many trees we need to produce before the model becomes effective (Williams, 2011). **The OOB score was 0.7097 which was remarkably like the score obtained from the testing set which was 0.7156.** We see that the accuracy measured by OOB is like that obtained with the testing set. It thus follows through the theory that the OOB accuracy is a better metric by which to evaluate the performance of your model rather than just the score. This is a consequence of bagging models and cannot be done with other types of classifiers.

b. K-Nearest Neighbors

This model was awfully close in accuracy to the SVM. However, one thing we did not do was select the exact correct number of K or neighbors. So, I tested the number of neighbors I used in the original model which was 100. This is what I found:




```
In [84]: print( "The best accuracy was with", mean_acc.max(), "with k=", mean_acc.argmax()+1)
         The best accuracy was with 0.7321575259724571 with k= 68
```

Fig. 20. The most accurate KNN model has a k of 68. This achieved an accuracy score of 0.732 or 73.2% which was remarkably close to the SVM model which was 73.6%.

c. Naïve Bayes Classifier

For this model I calculated the class probabilities of 1-having CVD vs. 0-not having CVD. Here are the first 10 class probabilities:

	Prob of NO CVD	Prob of Having CVD
0	0.902758	0.097242
1	0.690058	0.309942
2	0.706952	0.293048
3	0.979760	0.020240
4	0.007875	0.992125
5	0.907321	0.092679
6	0.938670	0.061330
7	0.662660	0.337340
8	0.010223	0.989777
9	0.798357	0.201643

Fig. 21. The established threshold was 0.5 for having CVD or not.

I then made a histogram of the predicted probabilities for having CVD based on the testing data set. Here it is:

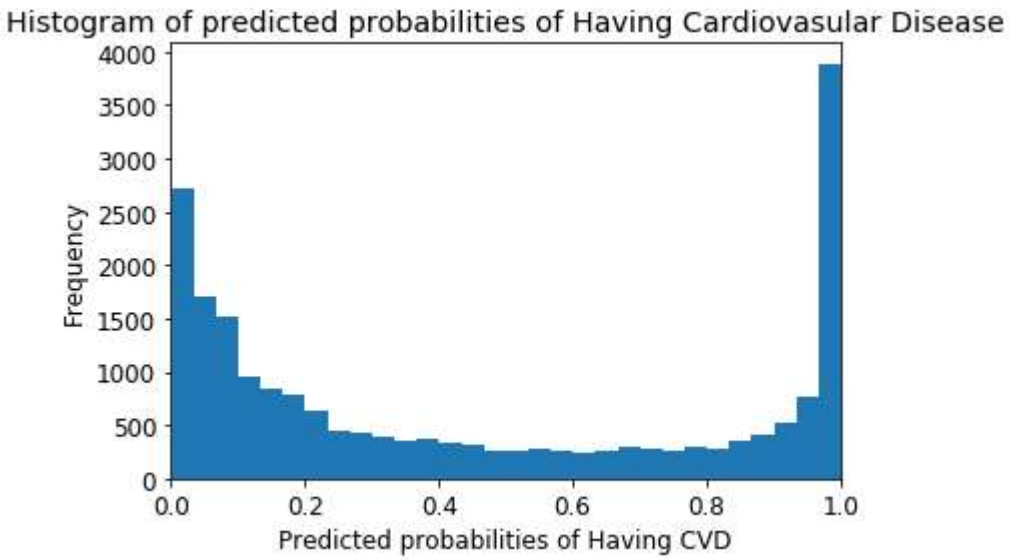


Fig. 22. Predicted probabilities of having CVD

- We can see that the above histogram is highly positive skewed.
- The far-right column tells us that there are approximately 4500 observations with probability between 0.8 and 1.0 with probability of having CVD.
- There are relatively small number of observations with probability <0.5 and more with probability between 0.0 and 0.2.
- So, these small number of observations predict that this population will more than likely have CVD.
- **Majority of observations predict that the population will have CVD which corresponds with the best accuracy prediction of 74% by the SVM which says three-quarters of the population of the dataset having these risk factors have CVD.**

d. Support Vector Machine

This was the best model, so I performed evaluation testing on this model by computing the confusion matrix, plotting the AUC-ROC curve, and calculating the Precision and Recall.

1. Confusion Matrix

A confusion matrix is a tool for summarizing the performance of a classification algorithm. A confusion matrix will give us a clear picture of classification model performance and the types of errors produced by the model. It gives us a summary of correct and incorrect predictions broken down by each category. The summary is represented in a tabular form. Four outcomes are possible: True Positives, True Negatives, False Positives (Type 1 error), and False Negatives (Type II error). The confusion matrix is as follows:

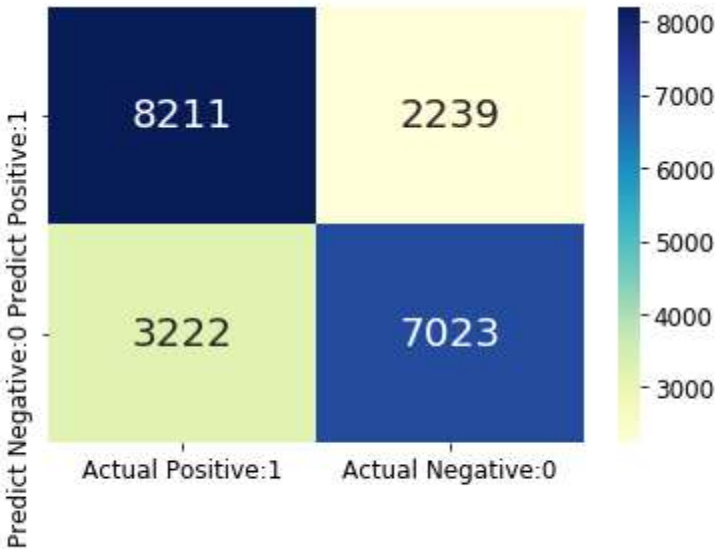


Fig. 23. Confusion Matrix for SVM best model. The confusion matrix shows 8211 + 7023 = 15234 correct predictions and 2239 + 3222 = 5461 incorrect predictions.

In this case, we have:

- True Positives (Actual Positive:1 and Predict Positive:1) - 8211
- True Negatives (Actual Negative:0 and Predict Negative:0) - 7023
- False Positives (Actual Negative:0 but Predict Positive:1) - 2239 (Type I error)
- False Negatives (Actual Positive:1 but Predict Negative:0) - 3222 (Type II error)

2. ROC-Curve

ROC Curve stands for Receiver Operating Characteristic Curve. An ROC Curve is a plot which shows the performance of a classification model at various classification threshold levels. The ROC Curve plots the True Positive Rate (TPR - also called Recall) against the False Positive Rate (FPR) at various threshold levels. The ROC Curve plots TPR vs FPR at different classification threshold levels. If we lower the threshold levels, it may result in more items being classified as positive. It will increase both True Positives (TP) and False Positives (FP).

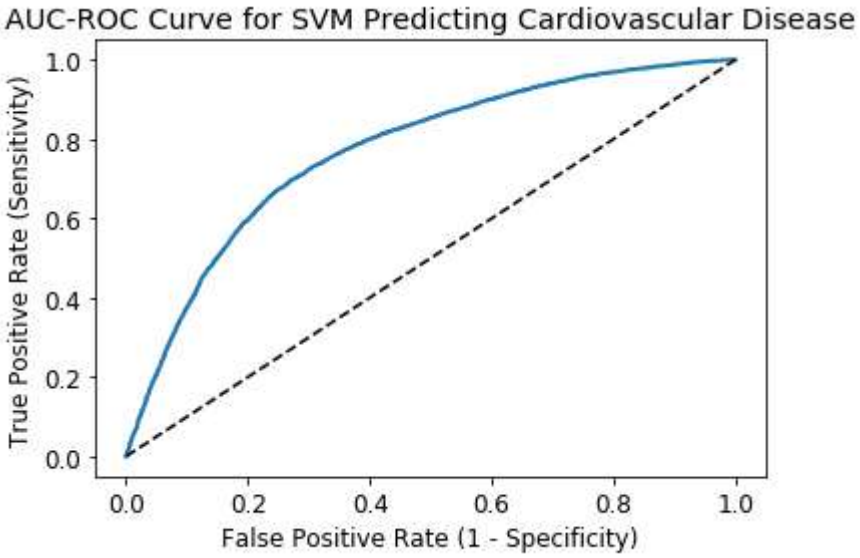


Fig. 24. AUC-ROC Curve. **The AUC (area under the curve value was 0.768) which shows the prediction rate is consistent and just below 80%.**

3. Precision and Recall

Precision: can be defined as the percentage of correctly predicted positive outcomes out of all the predicted positive outcomes. It can be given as the ratio of true positives (TP) to the sum of true and false positives (TP + FP). So, Precision identifies the proportion of correctly predicted positive outcome. It is more concerned with the positive class than the negative class. Mathematically, precision can be defined as the ratio of TP to (TP + FP).

```
In [101]:
#set up code to calculate
TP = cm[0,0]
TN = cm[1,1]
FP = cm[0,1]
FN = cm[1,0]

In [102]:
# print precision score

precision = TP / float(TP + FP)


print('Precision : {0:0.4f}'.format(precision))

Precision : 0.7857
```

Fig. 25. Precision Score is 0.7857. The ratio of true positives is consistent with our results so far.

Recall: can be defined as the percentage of correctly predicted positive outcomes out of all the actual positive outcomes. It can be given as the ratio of true positives (TP) to the sum of true positives and false negatives (TP + FN). Recall is also called Sensitivity.

Recall identifies the proportion of correctly predicted actual positives.

Mathematically, recall can be given as the ratio of TP to (TP + FN).

```
In [103]: recall = TP / float(TP + FN)

          print('Recall or Sensitivity : {0:0.4f}'.format(recall))

          Recall or Sensitivity : 0.7182
```

True Positive Rate is synonymous with Recall.

```
In [104]: true_positive_rate = TP / float(TP + FN)


          print('True Positive Rate : {0:0.4f}'.format(true_positive_rate))

          True Positive Rate : 0.7182
```

False Positive Rate

```
In [105]: false_positive_rate = FP / float(FP + TN)


          print('False Positive Rate : {0:0.4f}'.format(false_positive_rate))

          False Positive Rate : 0.2417
```

Fig. 26. We can see the Recall is 0.7182. This is the proportion of correctly predicted positives. The true positive rate is the same which backs up this finding. The false positive rate was low at 0.2417.

Results for Cross Validation

A 10 folds cross validation was run for the 2 best models: SVM and KNN. The results were as follows:

SVM Model: The accuracy was virtually the same as the original model which was 0.736. The standard deviation was 0.004 which shows the model is consistent overall.

KNN Model: The accuracy was again virtually the same as the original KNN model which was 0.731. The standard deviation was 0.004 which shows the model is again very consistent overall.

Results for Grid Search

The SVM model was selected as it was the best performing overall. The following parameters were tested:

```
from sklearn.model_selection import GridSearchCV
```

```
parameters = [{
    'kernel': ['linear','poly','rbf','sigmoid'],
    'C': [1,2,3,300,500],
    'max_iter': [1000,100000]
}]
```

Fig. 27. Parameters tested for SVM Model.

In [112]:
```
#find best parameters
SVM_cv = GridSearchCV(svm, parameters, cv=3) # GridSearchCV
SVM_cv.fit(X_train,y_train)# Fit

# Print hyperparameter
print("Tuned hyperparameters: {}".format(SVM_cv.best_params_))
print("Best score: {}".format(SVM_cv.best_score_))
```

```
Tuned hyperparameters: {'C': 3, 'kernel': 'rbf', 'max_iter': 100000}
Best score: 0.7319831013916501
```

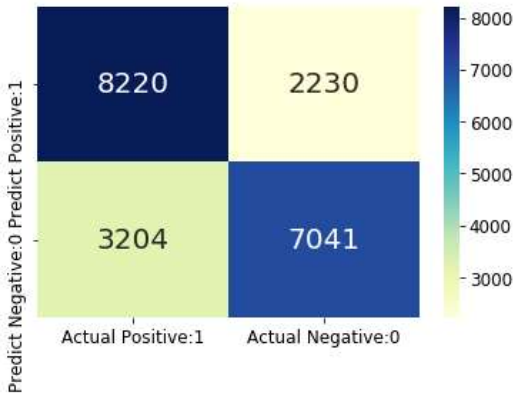
In [113]:
```
#run the model with best hyperparameters
SVM_best = SVC(C=3,kernel='rbf',max_iter=100000)
SVM_best.fit(X_train, y_train)
print("Test accuracy: ",SVM_best.score(X_test, y_test))
```

```
Test accuracy:  0.7374244986711767
```

Fig. 28. The hyperparameters identified were a C of 3, kernel of rbf and max_iter of 100,000. These were then run through the SVM model again and a better accuracy score was obtained of 0.737.

Final Optimized Model

Using the optimized hyperparameters from the grid search testing was then performed to evaluate the best model results and better results were seen for the confusion matrix but not for precision and recall.



The confusion matrix shows 8211 + 7023 = 15261 correct predictions (74%) and 2239 + 3222 = 5434 incorrect predictions (26%).

Fig. 29. New Confusion matrix for optimized SVM model. All aspects of the matrix improved in prediction capacity.

	Precision	Recall	F1 Score
Results	0.759465	0.687262	0.721562

Fig. 30. Final Precision, Recall and F1 Score for the optimized SVM model. Surprisingly, these values were worse with the new model which is a product of the "bias variance trade off".

So, from the Grid Search hyperparameter optimization technique we can say:

1. Precision decreased from 0.78 to 0.72
2. Recall decreased from 0.72 to 0.68
3. F1 score decreased from 0.74 to 0.72
4. The Jaccard Index remained the same at 0.74
5. **The number of True Positives went up from 8211 to 8220**
6. **The number of True Negatives went up from 7023 to 7041**
7. **The number of False Positives (Type 1 error) went down from 2239 to 2230**
8. **The number of False Negatives (Type 2 error) went down from 3222 to 3204**
9. **Accuracy increased from 0.7361 to 0.7374**

In summary, the best SVM model overall showed a 74% prediction rate for CVD.

Discussion

Overall, the best model was the SVM with a prediction rate of 74% for CVD. It had the least number of Type 1 and Type 2 errors at 26% total. The best Precision, Recall, and F1 score was obtained from the original SVM model at 0.78, 0.72, and 0.74, respectively. The Jaccard Index remained the same at 0.74.

The reason that the Precision, Recall, and F1 score went down is due to the “bias variance trade off”. Bias is the difference between the average prediction of our model and the correct value which we are trying to predict. Models with high bias pay little attention to the training data and oversimplifies the model. It always leads to high error on training and test data. Variance is the variability of model prediction for a given data point or a value which tells us spread of our data. Models with high variance pay a lot of attention to training data and does not generalize on the data which it has not seen before. As a result, such models perform very well on training data but has high error rates on test data. If our model is too simple and has very few parameters, then it may have high bias and low variance. On the other hand, if our model has large number of parameters then it is going to have high variance and low bias. So, we need to find the right/good balance without overfitting and underfitting the data. To build a good model we must find an even trade off which minimizes the total error (Singh, 2018). I believe we have done that with this SVM model.

While the other model’s accuracy performance was less than the SVM, overall if we considered the "averaging technique" from "ensemble modeling" the average accuracy was 72.2%. This is a great overall prediction effort considering that the accuracy rate for CVD prediction in machine learning has previously been reported at 69.11% for Naive Bayes, and as high as 85% using ensemble modeling techniques including Random forests (Latha et al. 2019). It has been reported that KNN has obtained an accuracy of 99.65% and SVM as much as 85%. Every algorithm has its intrinsic capacity to outperform other algorithms depending upon the situation. For example, a Random Forest performs much better with a large dataset, while Support Vector Machine performs better with smaller data sets (Khourdifi et al. 2019). The size of the dataset does play a role in the prediction capacity as well as the data being used. Our dataset had 70,000 patients which is average size compared to some studies that have been done.

It was interesting that we did see from the Random Forest Classifier that the most important variable predictors of CVD were: Height, Age_By_Decade, Cholesterol, Weight, Age, and Gender. The least important predictors were Obesity, Systolic_BP, Alcohol, Activity and BMI. While this does align with the standard disease predictors, it has been reported in different orders depending upon the dataset.

The Naive Bayes Classifier did show us that the prediction there are approximately 4500 observations with probability between 0.8 and 1.0 with probability of having CVD. There are relatively small number of observations with probability <0.5 and more with probability between 0.0 and 0.2. This is based on the probability threshold of 0.5 or greater having CVD.

The K-Nearest Neighbors Classifier was closest in performance to the SVM classifier. The algorithm appeared to optimize at k=68 and this made the accuracy virtually equal to the SVM classifier.

Cross Validation and Grid Search helped optimize the models and prove that SVM had the best overall accuracy of 74%. It was an interesting finding that the ‘rbf’ was the most optimal SVM model. SVM algorithms use a set of mathematical functions that are defined as the kernel. The function of kernel is to take data as input and transform it into the required form. Different SVM algorithms use different types of kernel functions. These functions can be different types. For example, *linear, nonlinear, polynomial, radial basis function (RBF), and sigmoid.* The most

used type of kernel function is **RBF.** Because it has localized and finite response along the entire x-axis (Dataflair, 2018).

There were some weaknesses of this project. The outliers were removed from the start which could have skewed the data although the intention was not to. There were 3 people identified as age 29 and they were included in the 30-39 age group. Again, this could have skewed the outcome. Some of the more common risk factors such as smoking, alcohol and activity level were more normal in those that had CVD which is usually not common in the general population. Also, the age groups in this study were all between age 30 and 65. We know that CVD prevalence is: ~40% age 40-59, ~75% age 60-79, and ~86% age 80 and over (Rodgers et al. 2019). The dominant age group in this study was 50-59, followed by 40-49, and lastly 60-69. So, the sample size of one of the most dominant groups (60-69) was much smaller at almost half of the most dominant age group in this study. Age groups were not even in sample size which is not fair to the predictions we make.

The data pre-processing used a robust scaler which is sensitive to outliers and perhaps this could have again skewed the results and a different type of scaler should have been used. Distance algorithms like KNN, K-Means, and SVM are most affected by the range of features. This is because behind the scenes **they are using distances between data points to determine their similarity. Therefore, scaling is used to reduce the chance that higher weight is given to features with higher magnitude. Tree based algorithms like Random Forests are insensitive to feature scaling (Bhandari, 2020).**

In the end whether to standardize (scale) or normalize the data depends on if there is a Gaussian distribution. Normalization is used when there is not a Gaussian distribution, Standardization is. In the end it is dependent upon the algorithms and the data. I do believe that feature scaling was appropriate to normalize the data as there were some high-level features and an even mix of continuous and categorical data points, so we had to "level the playing field".

Future directions for this prediction project could focus on more ensemble modeling and perhaps using more robust models such as XGBoost and AdaBoost. It has been shown in more recent studies that bagging and boosting techniques improve the accuracy of models significantly as the random sampling techniques are more robust (Khourdifi et al. 2019).

Lastly, not much was done with the inferential statistical analysis that was performed in this project. While it did show there seemed to be the strongest correlation between Body Mass Index (BMI) and Weight, not much was done in terms of predictive modeling for CVD. A Linear Regression model or Polynomial regression could be performed to further assess this numeric data and correlations to predicting CVD especially since it is known that BMI and Obesity are more common predictors.

Conclusion

In conclusion we were able to predict cardiovascular disease with an incidence of 74%. The error rate was low and the best algorithm for prediction was the Support Vector Machine. Grid Search and Cross Validation helped optimize the models and prove the prediction accuracy of the SVM. The Random Forest helped identify the most important predictors of CVD. Future work should be done with more robust machine learning models and perhaps consider using deep

learning techniques to improve the prediction accuracy using the most common risk factors for CVD. This work continues to be an important public health concern.

References

AHA (2020) Heart Disease and Stroke Statistics – 2020 Update. Retrieved from: https://professional.heart.org.

Bhandari, A. (2020) Feature Scaling for Machine Learning: Understanding the Difference Between Normalization vs. Standardization. Retrieved from: https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/

Carbone et al. (2019) Obesity paradox in cardiovascular disease: where do we stand? Vasc Health Risk Manag. Vol. 15:89-100.

Cho et al. (2020) Summary of Updated Recommendations for Primary Prevention of Cardiovascular Disease in Women. JACC State of the Art Review. Vol. 75(20): DOI: 10.1016/j.jacc.2020.03.060

Dataflair (2018) Kernel Functions – Introduction to SVM Kernel & Examples. Retrieved from: https://data-flair.training/blogs/svm-kernel-functions/

Dubey, A. (2019) Feature Selection Using Regularization. Retrieved from: https://towardsdatascience.com/feature-selection-using-regularisation-a3678b71e499

Gandhi, R. (2018) Naïve Bayes Classifier. Retrieved from: https://towardsdatascience.com/naive-bayes-classifier-81d512f50a7c

Gandhi, R. (2018) Support Vector Machine – Introduction to Machine Learning Algorithms. Retrieved from: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

Glen, S. (2016) Jaccard Index/Similarity Coefficient. Retrieved from: StatisticsHowTo.com.

Hajar, R. (2016) Framingham Contribution to Cardiovascular Disease. Heart Views. Vol. 17(2):78-81.

Johns Hopkins (2020) Alcohol and Heart Health: Separating Fact from Fiction. Retrieved from: hopkinsmedicine.org.

Khourdifi et al. (2019) Heart Disease Prediction and Classification Using Machine Learning Algorithms Optimized by Particle Swarm Optimization and Ant Colony Optimization. International Journal of Intelligent Engineering & Systems. Vol. 12(1):242-252.

Kotsiantis, S (2007). Supervised machine learning: A review of classification techniques. Informatica, 31, 249–268.

Krstajic et al. (2014) Cross-validation pitfalls when selecting and assessing regression and classification models. Journal of Chemoinformatics. Vol. 6(10):1-15.

Latha et al. (2019) Improving the accuracy of prediction of heart disease risk based on ensemble classification techniques. Informatics in Medicine Unlocked. Vol. 16.

Lavie et al. (2019) Sedentary Behavior, Exercise, and Cardiovascular Health. Circulation Research. Vol. 124(5). Retrieved from: https://ahajournals.org.

MIT (n.d.) Categorical Data. Retrieved from: http://www.mit.edu/~6.s085/notes/lecture6.pdf

Narloch et al. (1995) Influence of breathing technique on arterial blood pressure during heavy weightlifting. Arch Phys Med Rehabil. Vol. 76(5): 457-62.

Ptonski, P. (2020) Random Forest Feature Importance Computed in 3 Ways with Python. Retrieved from: https://mljar.com/blog/feature-importance-in-random-forest/

Ray, S. (2017) 6 Easy Steps to Learn Naive Bayes Algorithm with codes in Python and R. Retrieved from: https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/

Rodgers et al. (2019) Cardiovascular Risks Associated with Gender and Aging. Journal of Cardiovascular Development and Disease. Vol. 6(19): 1-18.

Roy, B. (2020) All about Feature Scaling. Retrieved from: https://towardsdatascience.com/all-about-feature-scaling-bcc0ad75cb35

Schott, M. (2019) K-Nearest Neighbors (KNN) Algorithm for Machine Learning. Retrieved from: Medium.com.

Singh, A. (2018) A Comprehensive Guide to Ensemble Learning (with Python codes). Retrieved from: https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-for-ensemble-models/

Singh, S. (2018) Understanding the Bias-Variance Tradeoff. Retrieved from: https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229

Srinidhi, S. (2018) Label Encoder vs. One Hot Encoder in Machine Learning. Retrieved from: https://medium.com/@contactsunny/label-encoder-vs-one-hot-encoder-in-machine-learning-3fc273365621#:~:text=What%20one%20hot%20encoding%20does,which%20column%20has%20what%20value.&text=So%2C%20that's%20the%20difference%20between%20Label%20Encoding%20and%20One%20Hot%20Encoding.

UCSF Health (2020) Understanding Your Risk for Heart Disease. Retrieved from: ucsfhealth.org.

Ulianova, S. (2018) Cardiovascular Disease dataset. Retrieved from: https://www.kaggle.com/sulianova/cardiovascular-disease-dataset

Unger et al. (2020) 2020 International Society of Hypertension Global Hypertension Practice Guidelines. Hypertension. Vol. 75(6):1334-1357.

Torpy et al. (2003) Risk Factors for Heart Disease. JAMA. Vol. 290(7):980.

Van Dorpe, S. (2018) Preprocessing with sklearn: a complete and comprehensive guide. Retrieved from: https://towardsdatascience.com/preprocessing-with-sklearn-a-complete-and-comprehensive-guide-670cb98fcfb9

Williams, G. (2011) Data Mining with Rattle and R. NY, NY: Springer Science.