Baruch COLLEGE | ZICKLIN SCHOOL OF BUSINESS

Course:

Data Mining for Business Analytics

CIS 9660 – S3DA [8685] – Summer 2021

Final Project Report

Binary Classification for Cardiovascular Disease Dataset

Professor:

Andrew Treadway

Authors:

Aarif Munwar Jahan (ammunwar.jahan@baruchmail.cuny.edu)

Jacob Bryer (jacob.bayer@baruchmail.cuny.edu)

John Makhijani (john.makhijani@baruchmail.cuny.edu)

Shawn Meng (smeng0428@gmail.com)

August 11th, 2021

Table of Contents

Executive Summary

This report covers documentation for all necessary deliverables for the final project of the CIS 9660 Group Project. The primary objective of this report is to provide documentation for all techniques used to predict binary classification on the cardiovascular disease dataset. The report is divided into five sections as follows:

1. **Section A:** Project Information
2. **Section B:** Data Cleaning and Exploration
3. **Section C:** Modeling
4. **Section D:** Model Evaluation
5. **Section E:** Improvements and Performance Modeling

Section A: Project Information

Background

Cardiovascular disease (CVD) includes heart disease, stroke, heart failure, cardiomyopathy, and other heart related issues. High blood pressure is estimated to account for approximately 13% of CVD deaths, while tobacco, diabetes, lack of exercise, and obesity are also major contributing factors. (Pekka & Norrving, 2011)

Cardiovascular disease accounted for 32.1% of global deaths in 2015, but McGill et al. (2008) estimate that up to 90% of cardiovascular disease may be preventable by improving on personal behavior that contributes to risk. Death by cardiovascular disease may also be avoidable if a patient receives treatment soon enough.

Our group's focus for the final project was to employ and showcase as many modeling techniques as we learned in this class. Specifically, we were interested in solving a classification problem after learning about the various supervised learning modeling techniques throughout the semester. We obtained a CVD dataset from Kaggle with about 700 votes to it. This dataset offers well-organized structured data without missing values that are optimum for running supervised learning algorithms to predict binary classification. In addition, the well-defined features and volume of the dataset also allowed us to apply unsupervised learning algorithms for detailed data exploration and segmentation purposes.

Problem Statement

How can we use known risk factors to build a machine learning model that identifies patients who might be more likely to have a cardiovascular disease?

Our project attempts to predict the presence or absence of cardiovascular disease in this dataset of 70000 patients using 11 predictor variables. These predictor variables include the leading risk factors for CVD, such as whether a patient smokes or exercises, and the blood pressure of the patient. Each observation in this dataset represents an individual patient. The data is already very clean, and we did not encounter many issues in cleaning it. We added the feature “BMI” for body mass index, a common ratio of height to weight.

We attempt to predict CVD classification using three statistical modeling techniques: logistic regression, random forest, and XGBoost. We will also use the K-means clustering technique to dissect the original dataset into groups to make inferences about potential performance improvements. We tune our models using cross validation and grid search for the optimal hyperparameters.

Section B: Data Cleaning and Exploration

Data Dictionary

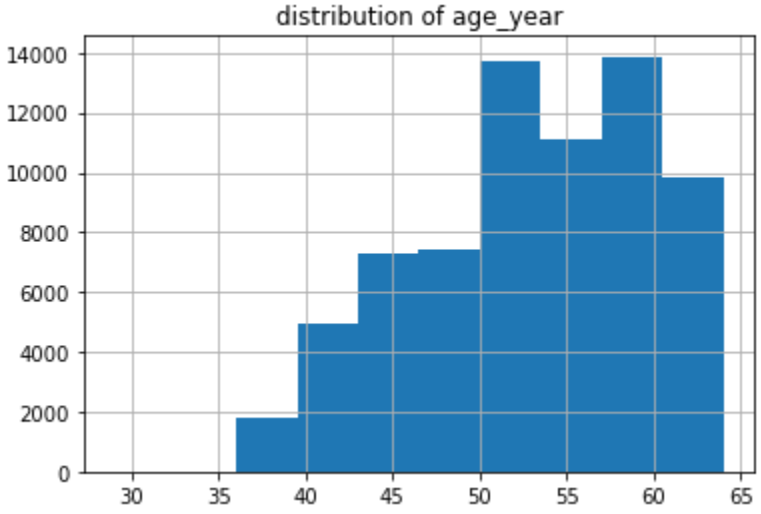
Name	Type	Variable Name	Format
Age	Objective Feature	age	int (days)
Height	Objective Feature	height	int (cm)
Weight	Objective Feature	weight	float (kg)
Gender	Objective Feature	gender	1 : male 2 : female
Systolic blood pressure	Examination Feature	ap_hi	int
Diastolic blood pressure	Examination Feature	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

Data Cleaning and Exploration

The label of this dataset is cardio, which indicates whether a person has cardiovascular disease. There are 35021 positive labels and 34979 negative labels. Thus, the label of this dataset is evenly distributed.

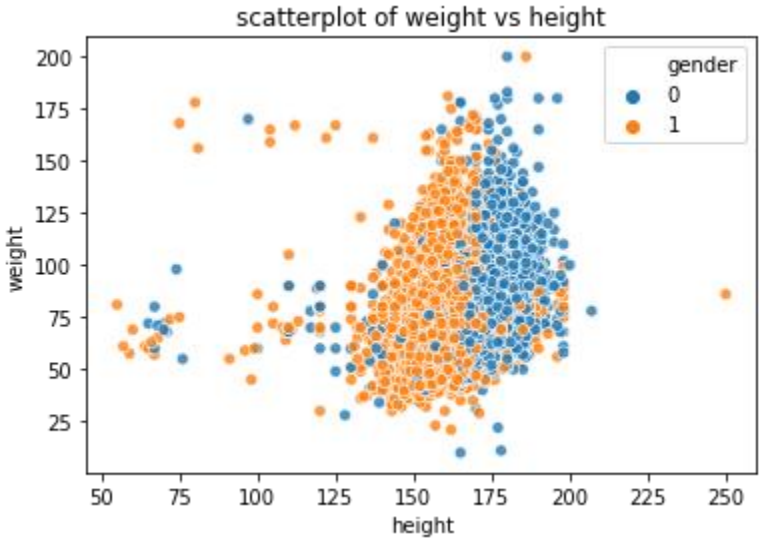
Even though there are no missing values for this dataset, there are several variables with abnormal data.

First, we looked at the age variable. According to the codebook, the age was represented in days when the diagnosis was performed. It is hard for us to understand this variable. So age_year was created, which indicated the age of year when the diagnosis was performed. From the summary statistics, we can see that the minimum age is 29, the maximum age is 64, and the median is 53. From the histogram plot, we can also see that the age is skewed to the left. So we can conclude that this dataset only contains adults with more data about older adults since we do not have details about the data source. This information could provide values reference for further analysis.



Second, we looked at the height and weight variables. From the summary statistics, we can see that the minimum height is 55cm, maximum height is 250cm. For the weight, the minimum of

10kg, and the maximum weight is 200kg. Is it possible for a person to measure at 250cm? According to Wikipedia, the tallest living man is 252cm. But according to our dataset, this person with 250cm is female, which is impossible. Therefore, we excluded this datapoint because it is unrealistic.



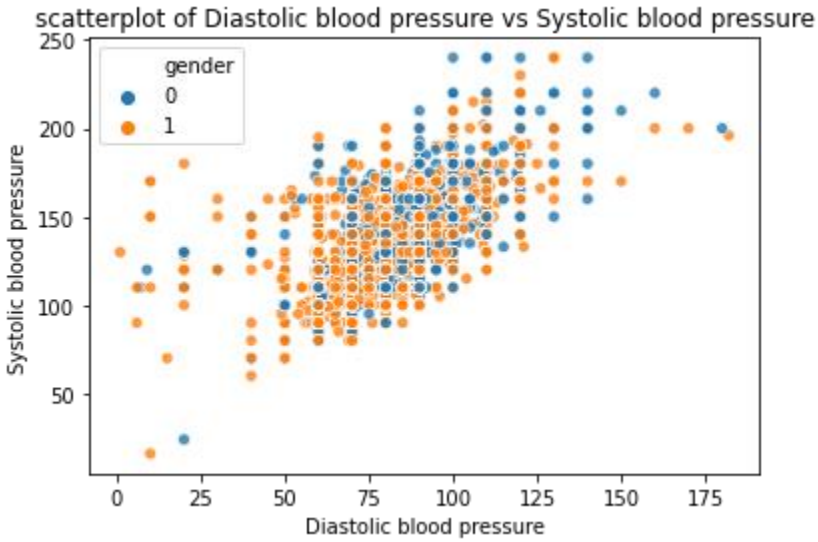
From this weight vs. height, we can also see most data clustered together. Also, males tend to be heavier and taller, which is in line with our common knowledge. But several pieces of data stand out. From height or weight alone, it is hard to exclude these data because people with certain conditions, like dwarfism, have smaller stature.

The most common way to measure a person's body type is BMI. We created this variable with the formula BMI = weight / (height/100)^2. From the summary statistics of BMI and the box plot of BMI, we can see several outliers. Therefore, we used the interquartile rule to exclude the outlier. This step filtered out about 3% of the data.

Then we looked at the ap_hi and ap_lo, which is about blood pressure. First, it is impossible to have negative blood pressure. Furthermore, we choose 300 as the ceiling for systolic blood pressure. Some medical professionals suggest systolic blood pressure higher than 180 or diastolic blood pressure greater than 100 requires immediate medical attention. The data

source is ambiguous about where the data was collected; we cannot rule out the possibility that the data was collected from the hospital. Giraffes have the highest systolic blood pressure with 300 among all species of mammal. Therefore, we will use it for the upper limit of systolic blood pressure. Also, Systolic blood pressure is always higher than diastolic blood pressure. Consequently, we exclude the data that diastolic blood pressure is higher than systolic blood pressure.

After cleaning abnormal blood pressure data, the systolic and diastolic blood pressure falls into narrow bands, which makes more sense. We suspect some anomalous blood pressure data was caused by different units being used because those data are about 100 fold higher than the normal range. However, we could not confirm it with the data source. Further investigation is recommended.

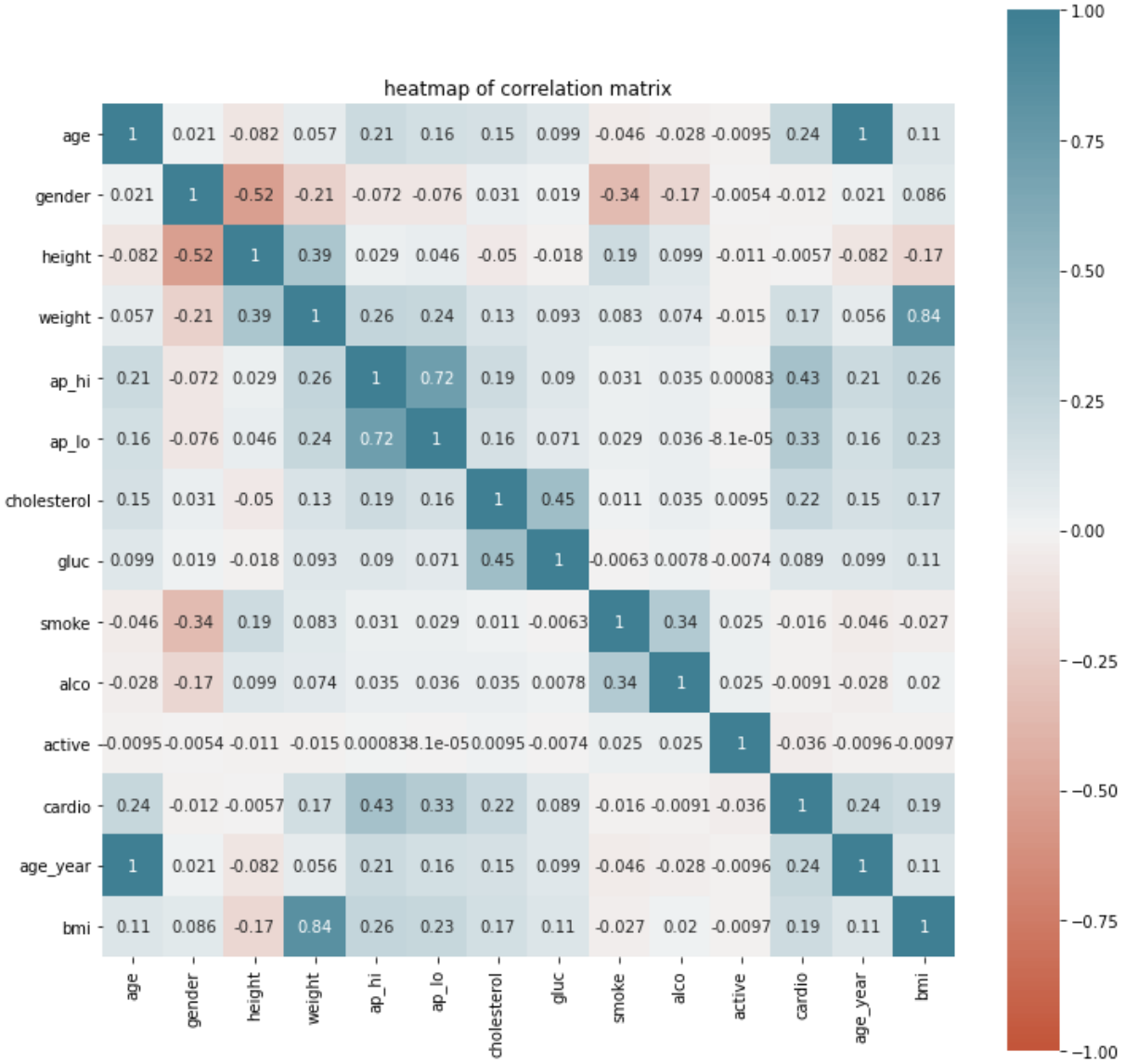


Lastly, the cleaned data is written into a new CSV file for modeling.

Section C: Data Modeling

Logistic Regression

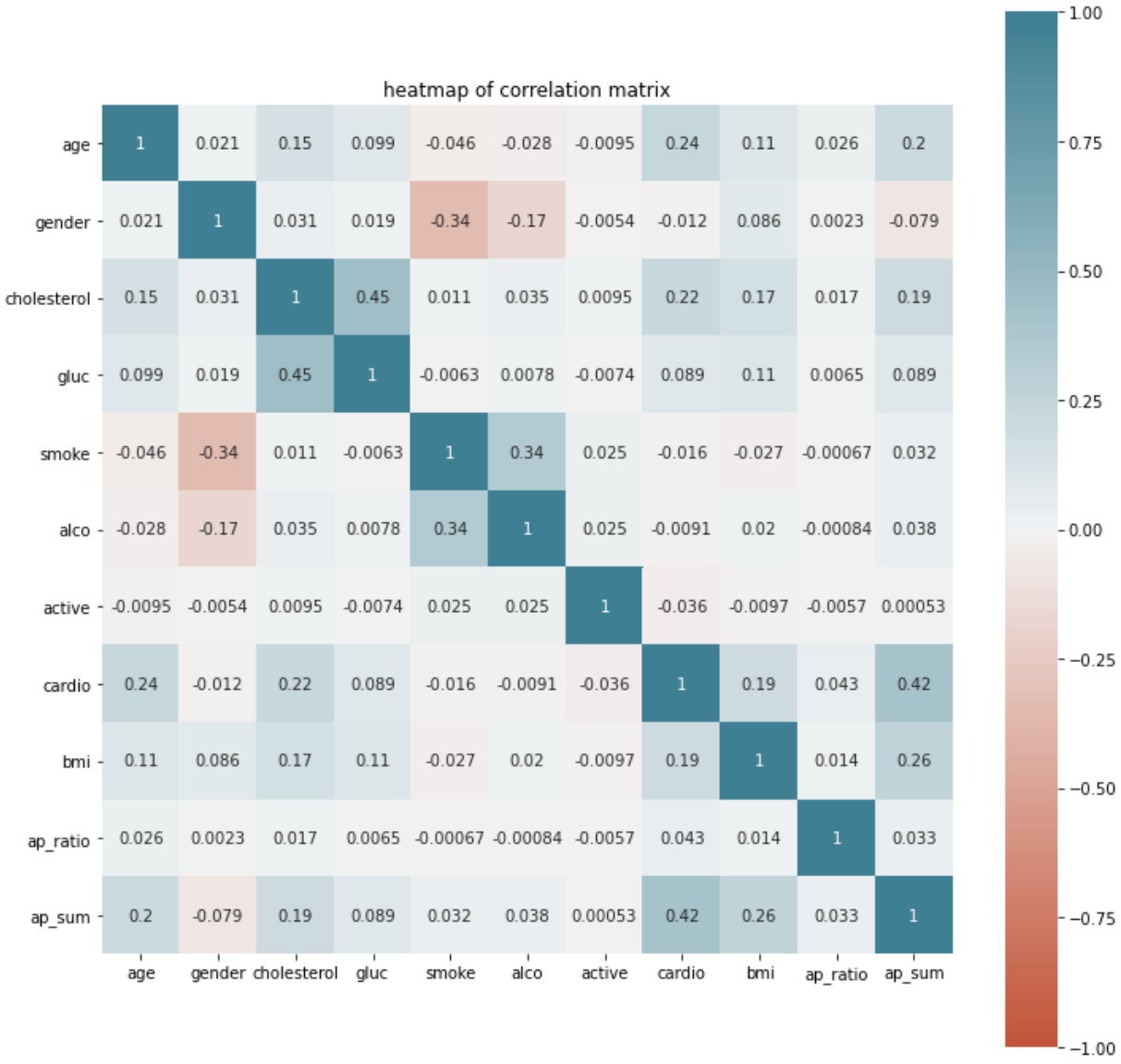
We will start by using logistic regression as a baseline model, since it is not computationally intensive, and it is easy to implement and interpret. Logistic regression requires that the features not be highly correlated, so the cleaned data will require some additional feature engineering to eliminate highly correlated variables (using a cutoff of r-squared = 0.5).



Looking at the above correlation matrix, it is apparent that age and age_year are highly correlated (r-squared = 1), since they represent the exact same information. BMI and weight are correlated (r-squared = 0.84), since BMI is calculated based on weight, ap_lo and ap_hi are

highly correlated (r-squared = 0.72), and height is negatively correlated with gender (r-squared = -0.52). These relationships will have to be addressed before training a logistic regression model on these data.

Since BMI is a function of two other variables (height and weight), it would be best to use either BMI or height and weight as features. Since gender appears to be strongly negatively correlated with height, but not BMI, I will choose to retain BMI. I will also eliminate age_years and use age instead. Finally, I will attempt to use the ratio of ap_hi to ap_lo (ap_hi/ap_lo) and the sum of these values (ap_hi + ap_lo) instead of using the two variables on their own.



After feature engineering, there are no longer any variables with correlations over 0.5 or below -0.5.

I've chosen to standardize the data from 0 to 1, rather than using the z-score. I then split the standardized data into a training and validation set using a 70%/30% split in favor of the training data, and fit a logistic regression model on the training data using scikit learn. Using 10 fold cross validation and the L1 regularization penalty, this model took two minutes to train and converged in less than 250 iterations. The optimal value of C, regularization strength, was determined to be 120 through cross validation. The estimated coefficients are printed as follows.

'age': 1.763,
'gender': -0.0255,
'cholesterol': 1.0258,
'gluc': -0.198
'smoke': -0.169,
'alco': -0.213,
'active': -0.227,
'bmi': 1.113,
'ap_ratio': 104.472,
'ap_sum': 14.0588

Despite using the L1 regularization penalty, none of the coefficients were reduced to zero. This means that all of the coefficients are useful in predicting the presence or absence of cardiovascular disease. This conclusion is supported by the literature. (Pekka & Norrving, 2011) (McGill et al., 2008)

This model achieved an AUC of 0.79. To attempt to improve this, I performed one-hot encoding on the cholesterol and glucose variables. These variables have three levels, where 1 indicates low, 2 indicates medium, and 3 indicates high. After creating dummy variables out of this

information, the model training time was reduced to 1 minute and the regularization strength was reduced to 90, indicating stronger regularization. The estimated coefficients are as follows:

'age': 1.754
'gender': -0.02385
'smoke': -0.167
'alco': -0.212
'active': -0.228
'bmi': 1.107
'ap_ratio': 104.751
'ap_sum': 14.056
'cholesterol_1': -3.356
'cholesterol_2': -2.981
'cholesterol_3': -2.219
'gluc_1': -3.042
'gluc_2': -3.012
'gluc_3': -3.342

Again, no coefficients were reduced to zero through regularization.

While one-hot encoding reduced model training time by 30 seconds, the prediction performance is nearly the same:

Without one-hot encoding:

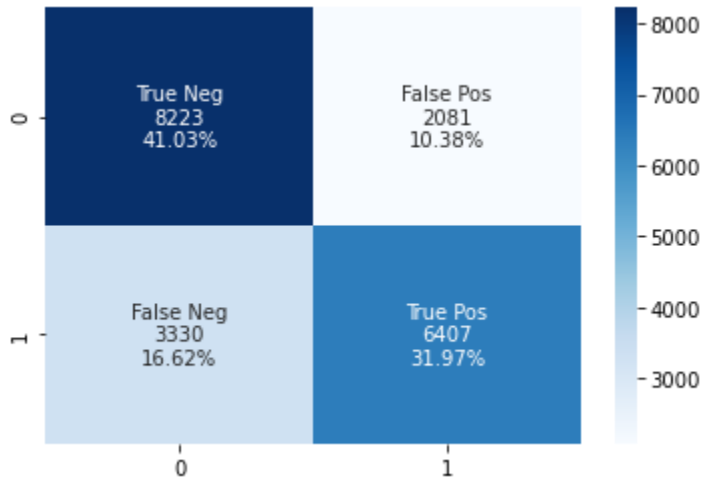
	train	val
accuracy	0.72618	0.72940
precision	0.75118	0.75269
recall	0.66081	0.65985
f1	0.70310	0.70322
auc	0.78893	0.79249

With one-hot encoding:

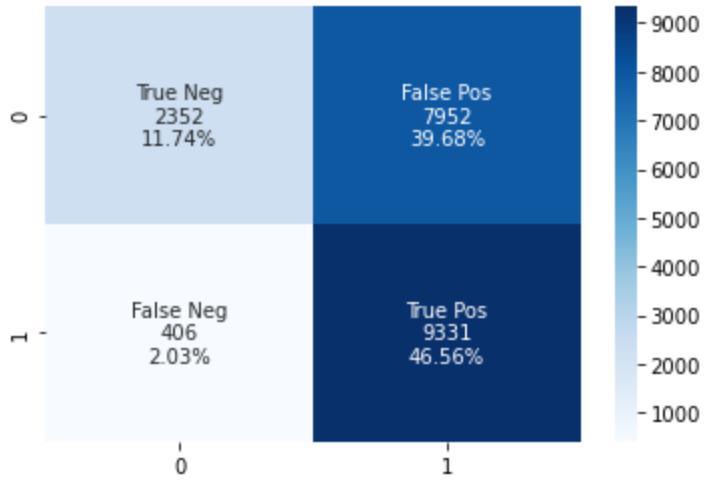
	train	val
accuracy	0.72725	0.73000
precision	0.75377	0.75483
recall	0.65955	0.65801
f1	0.70352	0.70310
auc	0.78951	0.79289

One-hot encoding made a very small improvement in AUC, accuracy, and precision with a slight decrease in recall and F1. The results are generally the same, however. I will conclude that one-hot encoding these variables is a good performance enhancing strategy.

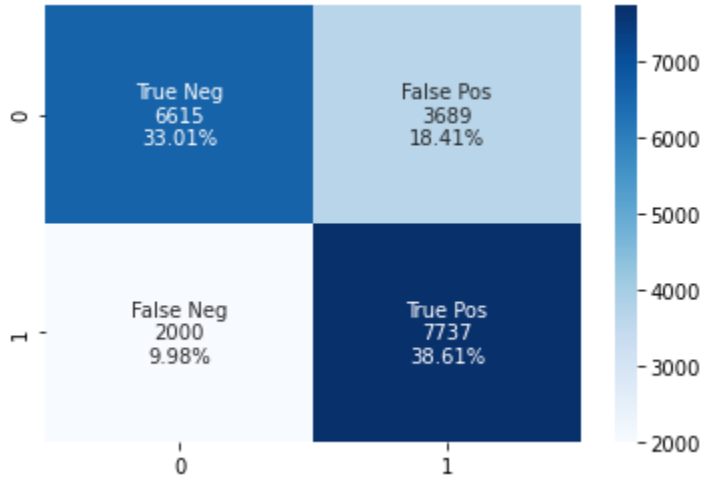
Most important, however, in assessing the performance of this model is the confusion matrix. Here we are interested in the number of false negatives. These are patients whose cardiovascular disease has potentially gone undetected. This is much more serious than a false positive, a patient who may be at risk for cardiovascular disease but, upon further testing, was found not to have cardiovascular disease. These patients, despite not having cardiovascular disease, are still highly at risk and should take steps to improve their lifestyle. Therefore the optimal probability threshold should not be based on making the model as accurate as possible, but should be based on reducing the number of false negatives to be as low as possible.



The confusion matrix indicates that 3330 patients with cardiovascular disease have gone undetected. By choosing a lower probability threshold for the logistic regression predictions, this number can be lowered. It might be reasonable to lower the probability threshold to around 20%, in which case only 2% of the positive observations in the validation data are missed. If this model were to be put into production in a medical setting, however, there could be thousands of missed cases.



If the consequences of a false negative are not as serious, for example in some other business context, we might use this logistic regression model with an F1 optimized probability threshold. This leaves us with about 10% of observations classified as false negatives. This would still be more helpful than no model at all, or random guessing. This might be useful to deploy in a context where a quick calculation is needed, since this model takes only a minute to train. It is probably more useful, however, to train a more complex model that does not misclassify so many positive patients. This will be discussed in the sections below.



Random Forest

For the Random Forest Model, the cleaned dataset was split into training and validation sets using a 70%/30% ratio. Then the label was split out of the predictors. The age variable was dropped because it is essentially the same thing as the age_years. Then, a Random Forest model was trained without any hyperparameter tuning. The training process just took 1 second to train the model.

The feature importance was as follows:

	var	imp
0	ap_hi	0.419838
1	ap_lo	0.278120
2	age_year	0.108915
3	cholesterol	0.104939
4	bmi	0.056121
5	weight	0.021451
6	gluc	0.007042
7	active	0.001607
8	height	0.001044
9	gender	0.000423
10	smoke	0.000329
11	alco	0.000171

With the default probability of 0.5, the AUC for the training set is 0.78908 and the AUC for the validation set is 0.78661.

Hyperparameter tuning took 7 minutes and 47 seconds and resulted in the following optimized parameters:

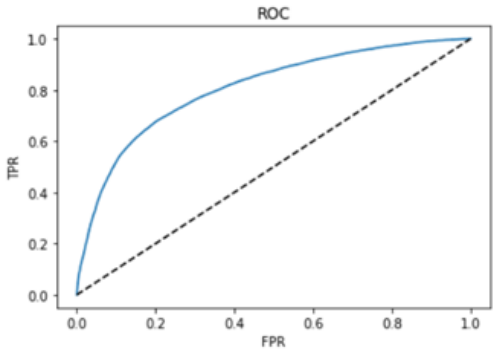
```
clf.best_params_
```

```
{'n_estimators': 300,
 'min_samples_split': 70,
 'min_samples_leaf': 30,
 'max_samples': 0.35,
 'max_features': 6,
 'max_depth': 7}
```

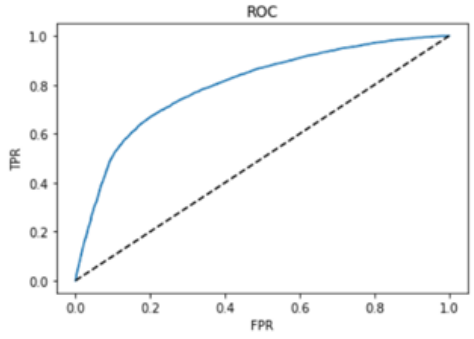
Feature importance changed after hypertuning

	var	imp
0	ap_hi	0.555829
1	ap_lo	0.163096
2	age_year	0.127071
3	cholesterol	0.083080
4	bmi	0.030142
5	weight	0.017646
6	height	0.009094
7	gluc	0.006009
8	active	0.004901
9	gender	0.001356
10	smoke	0.001070
11	alco	0.000706

AUC on Training set improved to 0.80683

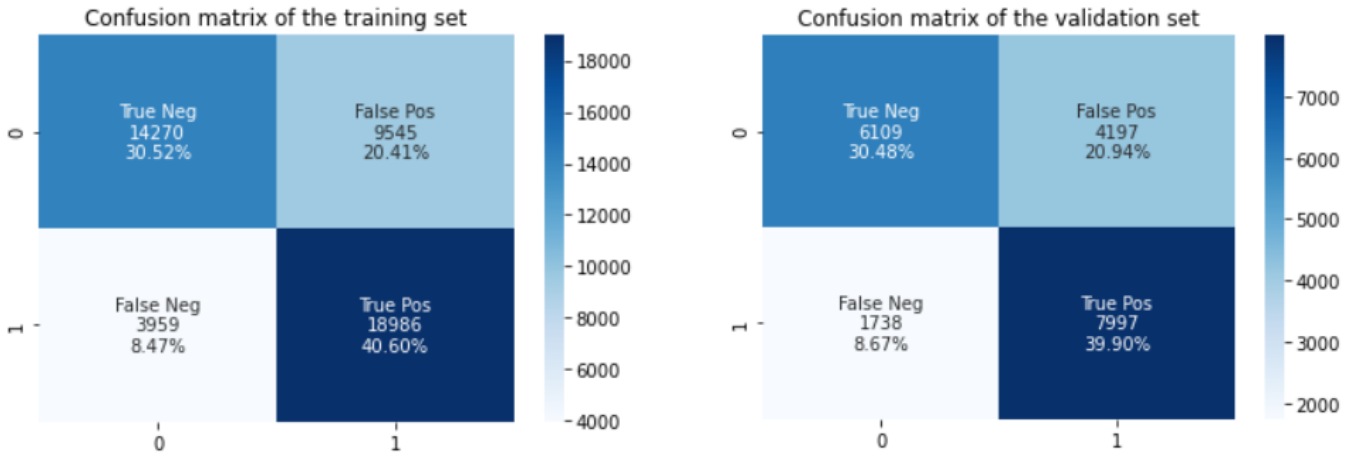


AUC on Validation set improved to 0.79827



Based on the best F1 score of the training set, the optimal threshold is 0.36.

Here is the confusion matrix for the training and validation sets using the optimal probability threshold of 0.36.




```
Validation precision:  0.6558143349188126
Validation recall:  0.8214689265536723
Validation accuracy:  0.7038570929594332
Validation F1-Score:  0.7293538237037713
```

Overall, based on the confusion matrix, the training and validation set have similar results indicating there is not an overfitting issue. False negatives of under 10% is good but not great. False positives is higher at around 20% but at least a false positive is on the conservative side and further tests could reveal the false positive. Factoring in time and results, the base Random Forest without hypertuning appears to be a better choice given the results are very close but the time to run the model is significantly less.

XGBoost

First, the cleaned dataset was split into the training and validation set using a 70%/30% ratio. Then the label was split out of the predictors. The age_year variable was also dropped because it is essentially the same thing as the age.

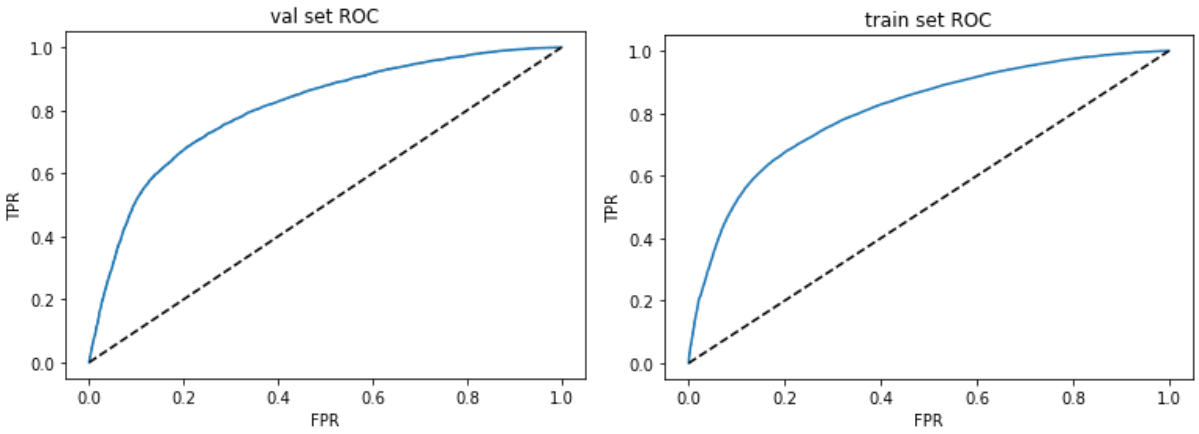
Then, an XGBoost model was made with randomized search cross-validation. The training process takes about 1 hour to train the model.

From the feature importance, we can see that blood pressures and cholesterol are the most important features followed by age, active, bmi, glucose, weight, smoke, alcohol, gender, and height.

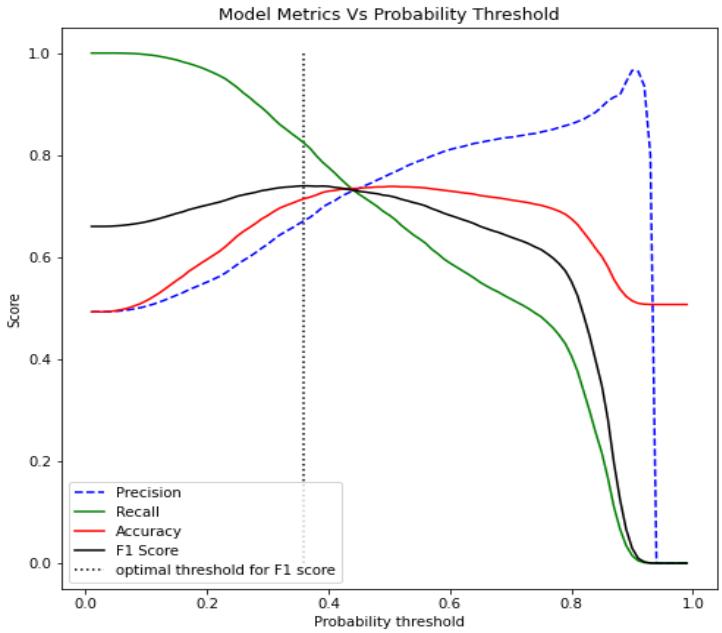
The best hyperparameters selected were:

'subsample': 0.8,
'n_estimators': 300,
'max_depth': 4,
'learning_rate': 0.05,
'lambda': 0.75,
'gamma': 5,
'colsample_bytree': 0.7,
'colsample_bynode': 0.5

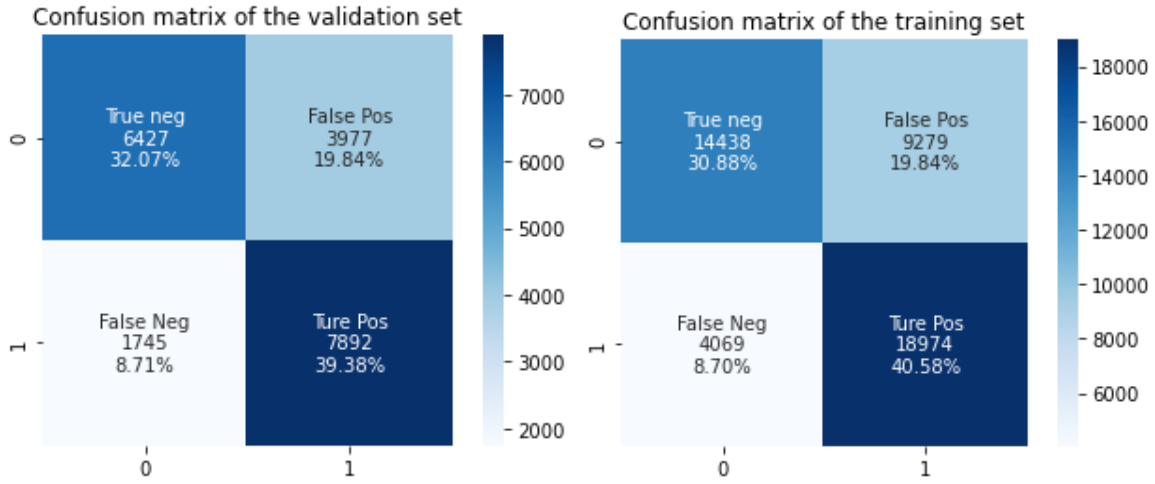
With the default probability of 0.5, the AUC for the training set is 0.80782 and the AUC for the validation set is 0.80543.



Because the label of this dataset is about balanced, we decided to optimize for F1 score. The optimal probability threshold is 0.36.



Here is the confusion matrix for the training and validation set using the optimal probability threshold of 0.36.



K-means Clustering

In this project, we solve a binary classification problem of whether a person has a cardiovascular disease based on certain predictor variables. These classification solutions are usually achieved by using supervised machine learning algorithms such as Logistic Regression, Random Forest, XGBoost, etc. On the other hand, clustering is a set of unsupervised machine learning algorithms that helps group subsets of data into homogenous groups. The primary function of clustering is to form groups and not classify. Thus, clustering in this project will not help directly compared to supervised learning algorithms as mentioned above. However, we can still use this powerful clustering algorithm to gather insights about the dataset, identify strong and weak cluster performance, and apply targeted work to improve model performance in the future.

Before starting clustering, we considered using PCA for dimensionality reduction. However, we decided against doing that. Even though PCA will help identify principal components that define most of the variance in data leading to better clusters, PCA will also create new aggregated features that will be hard to interpret after the segmentation. Besides, there are only thirteen features in this dataset which is quite manageable for a K-means clustering algorithm. PCA would have a significant impact if there were many dimensions where distance with K-means clustering loses its value. In fact, we attempted doing PCA, and it took seven principal components to explain 95% of the variance in the data. After comparing seven principal components to the original thirteen features, performing PCA before clustering was ill-advised.

Following high variable correlation concerns in the previous modeling techniques, the age, weight, height, ap_hi, ap_lo variables were dropped from the dataset. Instead, we used the engineered age_in_years, BMI and ap_ratio features to represent all the dropped variables.

We first started by feature engineering an optimized cluster predictor variable using k-means clustering. However, performance metric analysis showed zero to negligible performance improvement using this new feature on a logistic regression model.

The team then shifted focus to data segmentation, where after optimized clustering, we will train individual logistic regression models for each cluster to identify strong and weak performing clusters. The information can then be used to perform targeted work to improve the performance of lower-performing clusters by collecting more data concerning the given cluster and identifying other reasons that might have triggered the low performance.

The steps below were followed to achieve the goals stated above. The data used was already cleaned and explored. The steps are as follows:

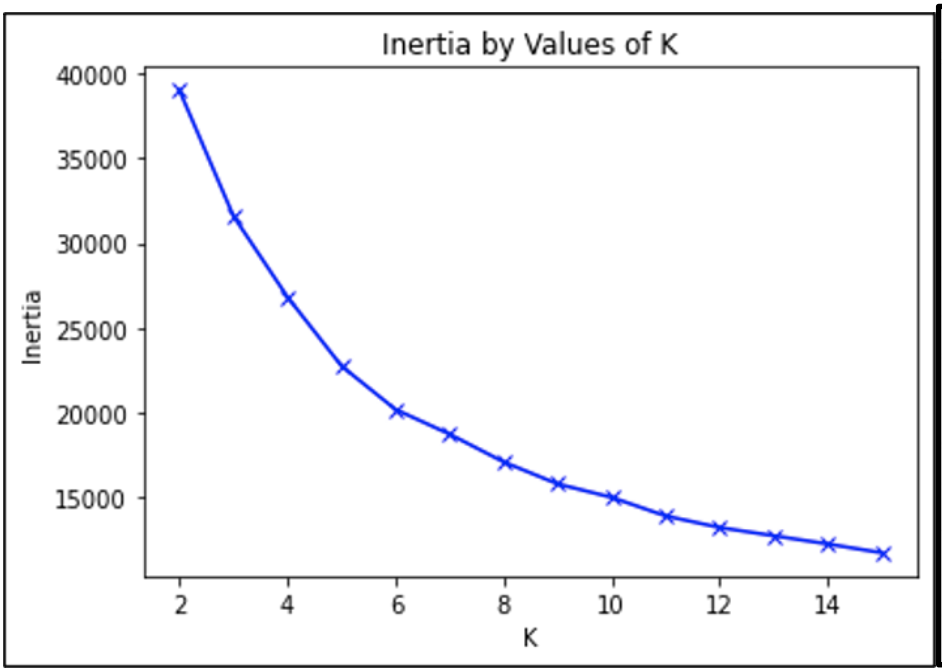
Step 1: Train a logistic regression model on the cleaned dataset and evaluate the performance
Step 2: Use k-means cluster to create segmentation on two important variables - age and BMI
Step 3: For data in each cluster, retrain logistic regression models
Step 4: Evaluate performance in each cluster and compare it with performance before clustering

In Step 1, a Logistic Regression model with optimized regularization parameter was used to get performance metrics (Precision, Recall, Accuracy, F1-Score) and AUC on the raw, cleaned data before clustering. The data used was split into training and validation sets using the standard 70/30 rule. The results are as follows:

	Training	Validation
Accuracy	0.72615	0.72935
Precision	0.75094	0.75272
Recall	0.66116	0.65965
F1-Score	0.70320	0.70312
AUC	0.78895	0.79257

The results perform well without any overfitting. The goal would be to break this dataset into multiple clusters and re-create this table for each cluster to see targeted performance.

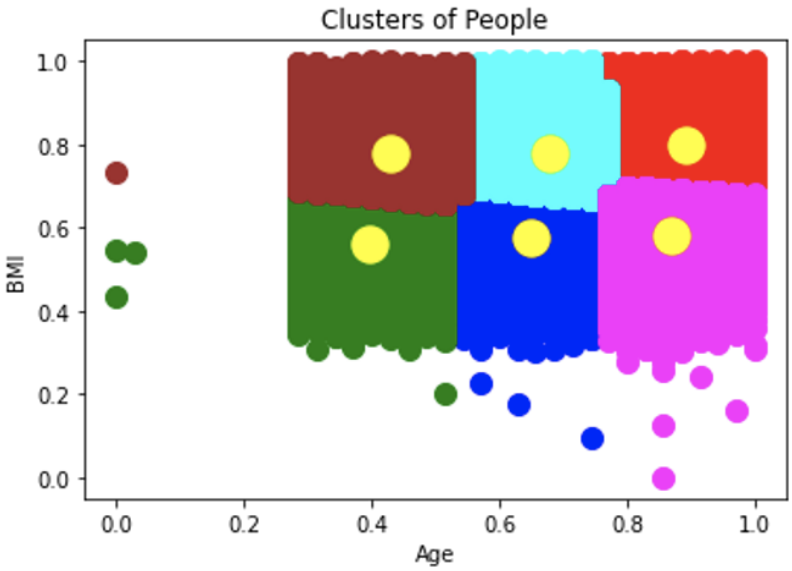
In Step 2, the k-means clustering technique was applied in a two-dimensional matrix of age vs. BMI for data segmentation. The idea is to create clusters based on two engineered features from the dataset: BMI and age in years. With the new clusters, new logistic regression models will be trained to check whether specific clusters perform better than the rest. The analysis can then be used to do targeted data collection or analysis for people in particular clusters. The number of clusters was optimized based on the Elbow Method (inertia) and the Silhouette Score table by running for loops for k-values in the range 2 to 15. The figures below show results from the two methods, respectively.



k_value	silhouette score
2	0.380191
3	0.423035
4	0.434172
5	0.461107
6	0.498472
7	0.470753
8	0.480255
9	0.492862
10	0.398074
11	0.410796
12	0.417804
13	0.360992
14	0.376300
15	0.384321

Both methods indicated an optimal k-value of 6. Since there is no difference in the optimal k value reported, the k-means cluster algorithm was reapplied on the data using k=6. If there had been a difference between the two methods, we would evaluate a trade-off between business judgement and data science.

The clusters created in the two-dimensional space using k=6 can be effectively viewed using the scatter plot function in the matplotlib package. The results are as follows:



Cluster definitions:

Cluster 0 - **Red** - Older Population with High BMI

Cluster 1 - **Blue** - Middle-Aged Population with Low BMI

Cluster 2 - **Green** - Younger Population with Low BMI

Cluster 3 - **Cyan** - Middle-Aged Population with High BMI

Cluster 4 - **Magenta** - Older Population with Low BMI

Cluster 5 - **Brown** - Younger Population with High BMI

We verify the clusters by concatenating the new cluster variable to the original dataset then grouping the dataset by clusters and looking at the mean for each column as follows:

kmeans_cluster	gender	cholesterol	gluc	smoke	alco	active	cardio	age_year	bmi	ap_ratio	ap_sum
0	0.71549	1.64652	1.38695	0.06141	0.05014	0.79285	0.69270	60.21974	32.48671	1.59287	219.14010
1	0.63801	1.26894	1.17637	0.09100	0.04973	0.80258	0.40471	51.69355	24.35195	1.55664	203.45311
2	0.59114	1.17492	1.11776	0.11114	0.05697	0.81175	0.27702	42.91207	23.89623	1.54997	196.89576
3	0.71064	1.44839	1.27378	0.08400	0.06095	0.80820	0.56740	52.76326	31.79878	1.56944	214.25819
4	0.61142	1.37554	1.22575	0.08312	0.04526	0.80404	0.56846	59.47655	24.62744	1.57844	208.37657
5	0.65852	1.32693	1.19612	0.10748	0.07315	0.80724	0.47751	44.01400	31.62868	1.54718	208.54656

From this table, we can take a closer look at the mean breakdown for the two features in question – age_year and bmi:

```
kmeans_cluster     kmeans_cluster
0   60.21974       0   32.48671
1   51.69355       1   24.35195
2   42.91207       2   23.89623
3   52.76326       3   31.79878
4   59.47655       4   24.62744
5   44.01400       5   31.62868
Name: age_year,    Name: bmi,
```

The clustering makes sense since we see the expected age and BMI mean values from the graph. For example, Cluster 0 was older people with high BMI, and we can how the mean for both age and BMI for Cluster 0 is the maximum value. Similarly, Cluster 2 was younger people with low BMI, and the mean table also shows the minimum mean values for both age and BMI for cluster 2.

Next, we explored the target variable breakdown for each cluster to get some preliminary inferences. The first of the two tables below show the count of the target variable (cardio disease) positive class (yes) and negative class (no) for each cluster. The second table shows the percentage of cardio disease positivity in each cluster.

HasCardioDisease	0	1
Cluster		
0	2562	5775
1	9839	6689
2	7754	2971
3	4053	5316
4	7113	9370
5	2800	2559

```
Cluster
0   69.26952
1   40.47072
2   27.70163
3   56.74031
4   56.84645
5   47.75145
dtype: float64
```

We notice clusters 1 and 4 having the most observations in them while cluster 5 has the least. Moreover, close to 70% of the data in cluster 0, that is, an older population with high BMI reported positive for cardio disease while only 28% of the data in cluster 2, that is, younger population with low BMI reported positive for the cardio disease. These inferences align with what we would generally expect for the combination of age and BMI features concerning the cardio disease.

In Step 3, we divide the dataset into six segments divided by the cluster variable. For data in each cluster, we retrain logistic regression models using optimized regularization parameters, split the data into training and validation datasets, and finally, calculate performance metrics (Precision, Recall, Accuracy, F1-Score) and AUC for each cluster dataset.

Finally, in step 4, we evaluate performance for data in each of the clusters and compare the cluster data performance with pre-clustering data performance. As a result, the final summary table containing performance metrics and AUC for training and validation set for each cluster looks like this:

Cluster	AUC_Train	AUC_Val	Accuracy_Train	Accuracy_Val	Precision_Train	Precision_Val	Recall_Train	Recall_Val	F1_Train	F1_Val
Pre-clustering	0.788952	0.792573	0.726155	0.729355	0.750941	0.752725	0.661160	0.659649	0.703196	0.703120
0	0.697057	0.718295	0.711054	0.711031	0.724328	0.724967	0.940856	0.939446	0.818515	0.818387
1	0.759158	0.750890	0.736969	0.741077	0.754294	0.768293	0.523080	0.507042	0.617762	0.610909
2	0.807314	0.791620	0.822299	0.812617	0.803661	0.791188	0.468023	0.455347	0.591549	0.578027
3	0.765339	0.763086	0.713327	0.710779	0.749065	0.728477	0.748665	0.769720	0.748865	0.748531
4	0.724294	0.744964	0.668747	0.677856	0.696775	0.701583	0.742097	0.746595	0.718722	0.723389
5	0.812038	0.834840	0.757131	0.777363	0.783163	0.790462	0.682601	0.719737	0.729433	0.753444

Performance metrics definitions:

1. AUC - provides an aggregate measure of performance across all possible classification thresholds
2. Accuracy Score - measures the ratio of correct predictions over all predictions

3. Precision Score - measures the ratio of true positives over all predicted positives
4. Recall Score - measures the ratio of true positives over all actual positives
5. F1 Score - measures an aggregated score based on precision and recall i.e. (2 * Prec. * Rec.) / (Prec. + Rec.)

From the summary table above, we see varying AUC and metric scores for the different clusters compared to the data pre-clustering. This was more or less, expected. The purpose of this analysis was to create segments on the data with two of the significant variables (age and BMI), then observe how well a logistic regression model performs for each of these segments or clusters, and finally identify which clusters contribute to the performance issues.

Let's focus on AUC and Accuracy first.

Based on the analysis, the logistic regression model within clusters 1, 2, and 5 shows improved performance compared with the pre-clustering scores. Conversely, clusters 0, 3, and 4 show slightly lower performance compared with the pre-clustering dataset.

The outcome of this analysis can lead the team to focus on cluster 4 and understand why the accuracy score is the lowest among all the other clusters. This can lead to more data collection, further data cleaning, more outlier reduction, etc.

If we look at the precision score, we can see cluster 2 has the best performance compared with others where generally there is similar performance compared to pre-clustering.

Both recall and F1 score show interesting results. Recall scores for cluster 0 are very high while clusters 1 and 2 show low recall scores. This means the ratio for true positives over all actual positives is the highest in cluster 0 and clusters 1 and 2 have the lowest ratio. This means with the default 50% probability threshold, cluster 0 sees more positives that agree with the actual

positive class. Since the recall score is drastically different for these clusters, the F1 score follows the same trend since F1 score is a function of recall and precision scores.

As an improvement, we can optimize the probability threshold for accuracy since the label variable is split in half for the original dataset; we would need to consider the cost of a false positive vs. false negative as well. However, for the purpose of clustering, we decided not to optimize the probability threshold since the idea is not to classify using clustering but to segment the raw data and understand performance differences for each cluster. The probability threshold will be optimized when we run models like Logistic Regression, Random Forest and XGBoost to do binary classification.

This analysis was done using a two-dimensional approach of age and BMI; we can repeat the analysis by clustering on other combinations of predictor variables and analyzing the corresponding results.

Section D: Model Evaluation

Model	Validation AUC	Validation False Negatives (%)
Logistic Regression	0.793	9.98%
Random Forest	0.798	8.67%
XGBoost	0.805	8.71%
Logistic Regression (20% threshold)	0.793	2.03%

The AUC of each model is virtually the same, but random forest classifies the smallest proportion of observations as false negatives. This model could be used to identify patients who are highly at risk and should be the target of additional follow up from the doctor, though

with the understanding that there is a small, but significant, chance that patients with cardiovascular disease are not identified by the model. We may want to use a lower probability threshold in this case, such as 20%, to reduce the number of false negatives. In the 20% logistic regression model, results can be obtained in about two and half minutes, and only 2% of positive patients are misclassified as negative. The negative patients who are misclassified as positive are still disproportionately likely to face cardiovascular disease than their negative counterparts who were correctly classified.

Since the models generally have the same performance, we are interested in putting the logistic regression model with a 20% threshold into production so that we can very rapidly obtain predictions. We expect that after a nurse collects this data from the patient, the model has generated a prediction quickly enough for a doctor to speak with the patient about their high risk for cardiovascular disease. Producing a quick calculation is important because the amount of time that a doctor can spend with each patient is very limited.

This model could potentially be deployed on a cloud-based system, where it could generate predictions for doctors and nurses treating patients in remote locations. These locations might not have the resources to provide every patient with treatment for cardiovascular disease, but a positive prediction from this model might justify the patient traveling to a larger healthcare facility where follow up testing can be performed. This way, they can receive preventive healthcare soon enough to potentially save their life or reduce the chance of hospitalization.

Section E: Improvements and Performance Monitoring

We should continue to monitor for false negatives, since these patients are at high risk for cardiovascular disease and should be identified. When a patient's data is entered into this system it should be stored for a long period of time so that we can follow up with patients and ask if they have undergone any testing for cardiovascular disease. If they have, their data can be labeled and used to train further models. This is the best way to improve the model.

For the first few years in the deployment of this model, we will certainly want to follow up with patients who received a positive prediction, since these patients were likely tested for cardiovascular disease soon after receiving their positive prediction. We should also follow up with negative predicted patients annually. Based on the results of these tests, the original prediction can be labeled and used as training data, as well as used to calculate performance metrics. Over time, a picture will start to emerge of performance. If the model begins misclassifying large numbers of positive patients as negative, the model may have to be re-evaluated.

Finally, we can use the K-means clustering technique for periodic data segmentation based on evolving feature importances. The clusters will allow us to plan targeted data mining activities to improve the model's overall accuracy, decreasing the number of false-negative predictions.

References

Pekka P, Norrving B (2011). Global Atlas on Cardiovascular Disease Prevention and Control (PDF). World Health Organization in collaboration with the World Heart Federation and the World Stroke Organization. pp. 3–18. ISBN 978-92-4-156437-3. Archived (PDF) from the original on 2014-08-17.

McGill HC, McMahan CA, Gidding SS (March 2008). "Preventing heart disease in the 21st century: implications of the Pathobiological Determinants of Atherosclerosis in Youth (PDAY) study". Circulation. 117 (9): 1216–27. doi:10.1161/CIRCULATIONAHA.107.717033. PMID 18316498.

Ulianova, S. (2019, January 20). *Cardiovascular disease dataset*. Kaggle. https://www.kaggle.com/sulianova/cardiovascular-disease-dataset