2021

Cardiovascular Disease Detection

BY:
MORGAN BACCUS
WEN WU

CPTS 475 | Project Report

Table of Contents

Abstract

For our final project, we implemented several machine learning algorithms to predict if a patient will develop cardiovascular disease based on the features in the dataset. We chose to use multiple algorithms so we could evaluate the performance of each model. After comparing various metrics for the five different models, we concluded that Random Forest performed the best while Gaussian Naïve Bayes performed the worst. A heat map of the dataset features revealed that age, cholesterol, and weight had the highest correlation with a patient having cardiovascular disease. Our findings confirmed our hypothesis of which model would perform best, but the feature correlation surprised us.

Introduction

Cardiovascular disease is the number one killer of adults in the United States. More than 616,000 people die of heart disease in the United States each year making up almost 25% of annual deaths. The ability to predict which patients are likely to develop cardiovascular disease could significantly decrease these statistics. This is why we chose to implement five different machine learning algorithms to find which model is the best at predicting if a patient will develop heart disease and what features in the dataset have the highest correlation to patients with the disease.

The problem here is simple: to predict if a patient will develop cardiovascular disease based on the features in the dataset. The features are all things that doctors have already established can affect cardiovascular disease onset in patients. Take age for example. We know that heart disease is more prevalent in older patients than younger ones. The question is, how correlated is age to cardiovascular disease and how do the other features affect that relationship? We were able to answer this for all the features in our dataset by creating a heatmap. The condensed Decision Tree visual we created from our model is also helpful in seeing what factors affect others.

Cardiovascular disease is relatively preventable if people take care of themselves by eating healthy, staying active, and managing their stress. That being said, certain uncontrollable factors, such as age and gender, make patients more susceptible to developing the disease. Our goal was to create models that can predict if a patient will have heart disease so that they can prevent it. Knowing that a patient is likely to contract cardiovascular disease can help the doctor to suggest certain lifestyle changes that will lower their risk.

Our approach was to use Python libraries to implement five machine learning algorithms as well as create visuals to help understand the results and compare the models. The main objective was to label if a patient had cardiovascular disease or not, so classification algorithms were the most appropriate for the task. We used multiple machine learning

algorithms to see which one performed the best. We were curious how different the results could be since they are all supervised classification algorithms. Our models were trained on the training set, a subset of the data, and then assessed on how they performed against the test set. Surprisingly, they all scored very differently for the metrics we chose to analyze.

Analyzing specific factors that are likely to indicate disease in patients is not a new idea. Doctors have been looking at this for years and will continue to do so for many more years. Our project differs from current techniques as it can analyze any number of patients at once and has the potential to be automated. Our program could be used in conjunction with another program that automatically reads patient information from their chart and then send an alert if that patient is at risk. This system could be used to check all the patients in a healthcare facility and significantly increase preventative care which would in turn lower the number of deaths caused by cardiovascular disease.

Overall, Random Forest Classifier scored the best for the metrics we analyzed. However, there are several measures that could be taken to improve the performance of the other models. The worst performing model was Gaussian Naïve Bayes. The three features in the dataset that had the highest correlation with cardiovascular disease being present were age, cholesterol, and weight respectively. We will discuss our findings in more detail later in this paper.

Problem Definition

The main problem that we wanted to address was predicting if a patient will develop cardiovascular disease. Since the desired result is a “yes” or “no”, a classification model was the most appropriate to use. But in order to implement classification models, we needed to know more about the dataset. We used a few different exploratory data analysis methods to get a better understanding of the dataset and the features it contains. After we did this, we were able to make more informed decisions about which classification models to use. This problem needed to be solved so that patients can preventatively treat cardiovascular disease to either delay the onset of the disease, possibly forever, or to lessen the severity of it.

Another problem that we wanted to explore was which model performed the best. To do this, we calculated and compared several metrics for each model. We also created confusion matrices as a more visual way to understand the accuracy of each model. Random Forest Classifier stood out as the best performing model, while Gaussian Naïve Bayes was the worst. This is an interesting problem since we are comparing five different models that are all supervised classification. We expected them to have similar results, yet they varied widely.

Additionally, we were interested in finding which features most strongly correlated with a patient having heart disease. Age, cholesterol, and weight were the three highest correlations

of all the features. Several features had negative correlations with cardiovascular disease that we expected to have a strong, positive correlation. We used a heat map to find and visualize the relationships between features. The correlations between features are important as it can help us to understand the most contributing factors to cardiovascular disease.

Models/Algorithms/Measures

Using the Sklearn Python Library, we implemented the five machine learning models we chose to analyze. These algorithms all utilize supervised learning which is the process of creating an algorithm that learns by mapping a particular input to a particular output. In the case of our project, we map the twelve features of the dataset to determine if a patient will have cardiovascular disease. Once we trained the models, we ran them against the test set to evaluate their performance. Since we are trying to predict a binary result, classification models were the most appropriate to use.

The first algorithm that we implemented was a Decision Tree. Using information gain, Decision Trees determine which feature is the most indicative of the final result and makes that the root node. This process continues until each instance in the dataset is accounted for. Every branch of the tree corresponds to a feature. Decision Trees are one of the most straightforward classification models, so we decided to use it as a sort of “base” to compare the others to.

We also included the Sklearn K-Nearest Neighbors with k=5. K-Nearest Neighbor works by calculating the distance from an unknown point to the closest number of k known points. The algorithm then votes on how to classify the unknown point. The classification with the most points in the k number of points is how the unknown point will be labeled. For example, if we have an unknown point close to three “no” for cardio and only two “yes”, the point will be classified as a no.

The next algorithm we implemented was a Random Forest Classifier with n=100. Random Forest Classifier works by creating n number of decision tree classifiers on subsets of the data and uses averaging to increase the accuracy and reduce over-fitting. This was our best performing model overall out of the five we analyzed.

Gaussian Naïve Bayes is a classification model that uses statistics to sort and label the data. The algorithm got its name from Bayes Theorem which is used to calculate the probability that a patient will develop cardiovascular disease. Naïve Bayes assumes that the features of a dataset are all independent, hence why it is naïve. For example, height is going to affect weight, but Naïve Bayes will consider them independently. Naïve Bayes is one of the fastest and better performing classification algorithms, which is why we were interested to see how it compares against the other four.

Support Vector Machines are another classification algorithm that utilizes statistical learning. However, Support Vector Machines are based on the statistical learning theory of Vap Nik. By using smaller Kernal function, the algorithm creates a hyper-plane that is then divided into sections and each section is a classification. In the case of our project, the hyper-plane would be divided into sections for “yes” and “no” for developing cardiovascular disease. If a support vector falls in the “yes” section, then that patient will be classified as likely to contract the disease.

In order to compare the algorithms, we used the Sklearn metrics library. We calculated F1, accuracy, precision, recall, and cross-validation score for each model. We also created confusion matrices for each to be able to visualize the results more easily. F1 score is the harmonic mean of precision and recall. The best F1 score is 1 and the worst is 0. Accuracy is the computation of how well each model predicted outcomes correctly versus incorrectly. Precision is the ratio of true positives over the addition of true positives and false positives. It is the ability of the model to not mislabel a sample as positive when it is actually negative. Recall is similar to precision but is the ratio of true positives over the addition of true positives and false negatives. Recall represents the ability of the model to find all the positive samples. Cross-validation score is useful when assessing the effectiveness of a model, especially when over-fitting is a concern. We also calculated t-statistic and p-values for each model once all other calculations were complete. These are all the metrics that we analyzed to judge the performance of each model.

Implementation/Analysis

The first step that we took to create our project was to find and download the dataset “cardio_data.csv” from Kaggle. Then we set up a Google Colaboratory document so that we could write code simultaneously. We then read in the “cardio_data.csv” file using the Pandas library function read.csv. Next, we split the data by semicolon and put each feature into their own column. Once all the data was split, we named each column by the feature it holds.

Before we could begin implementing any algorithms, we needed to learn more about our dataset. We started by printing out a section of the dataset (fig. 1).

	id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active	cardio
0	988	22469	1	155	69.0	130	80	2	2	0	0	1	0
1	989	14648	1	163	71.0	110	70	1	1	0	0	1	1
2	990	21901	1	165	70.0	120	80	1	1	0	0	1	0
3	991	14549	2	165	85.0	120	80	1	1	1	1	1	0
4	992	23393	1	155	62.0	120	80	1	1	0	0	1	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
69296	99993	19240	2	168	76.0	120	80	1	1	1	0	1	0
69297	99995	22601	1	158	126.0	140	90	2	2	0	0	1	1
69298	99996	19066	2	183	105.0	180	90	3	1	0	1	0	1
69299	99998	22431	1	163	72.0	135	80	1	2	0	0	0	1
69300	99999	20540	1	170	72.0	120	80	2	1	0	0	1	0

Figure 1: Dataset

Now that we understood the format of the dataset, we could do exploratory data analysis. We found the total number of rows to be 69,301 and total columns to be 13. All of the data types were objects and there were no null entries. Next, we created a heatmap of all the features by finding the correlation matrix and then converting it to a heatmap using the Seaborn library function. The heatmap was very insightful into the correlation between features. We will discuss what the relationships are in the next section.

To continue with our exploratory data analysis, we created three bar charts. Each bar chart shows the distribution of three different features. We chose to use features that have continuous values and are relatively out of the patient's control. This was helpful in assessing the patient demographic and finding any outliers.

After learning more about the data, we split our training and testing set to be 70/30 of the dataset. At this point, we looked at our data and decided we needed to use classification models as we wanted to classify each patient as either having or going to have cardiovascular disease or not. We discussed several different algorithms but couldn't decide which would be the best for the given situation, so we came up with the idea to use multiple classification algorithms and then compare the results. The algorithms we chose were Decision Tree, K-Nearest Neighbors with k=5, Random Forest Classifier with n=100, Gaussian Naïve Bayes, and Support Vector Machines. We then added the Sklearn library and the necessary extensions to implement the algorithms.

Our next step was to implement the algorithms. We started by creating the classifier variables for each model using their respective library calls. Next, we wrote the code needed to create confusion matrices and calculate F1, accuracy, precision, recall, and cross-validation

scores for each model. Then we printed out the best result for each score. This is also when we calculated the t-statistic and p-value for each model and printed those out as well.

We then created a visual representation of our Decision Tree model but limited the depth to three for readability. We also created a chart with all the metrics for each model. The worst score is highlighted in red, and the best score is highlighted in green for each metric. Our main basis of comparison between the models was this chart in conjunction with the confusion matrices.

We hypothesized that the models would perform in the following order from best to worst: Random Forest Classifier, Decision Tree, K-Nearest Neighbors, Support Vector Machines, and Gaussian Naïve Bayes. Our ranking had to do with how well the algorithm could handle large sets of data, how it addressed interdependencies, and its reputation for classifying data accurately. We will discuss if our hypothesis was supported by the results in the following section.

Results and Discussion

The results we produced revealed the features with the highest correlation to cardiovascular disease as well as some unexpected insights. The key information we were looking for in the exploratory data analysis phase was what kind of data types are there, if there were any outliers, and if there was a class imbalance. During the exploratory data analysis we learned the size of our dataset as well as that the only datatypes it held were objects. We also learned that there were no missing or unexpected entries, so we did not need to clean the data. See Figure 2.

```
69301 rows × 13 columns
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 69301 entries, 0 to 69300
Data columns (total 13 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   id           69301 non-null  object
 1   age          69301 non-null  object
 2   gender       69301 non-null  object
 3   height       69301 non-null  object
 4   weight       69301 non-null  object
 5   ap_hi        69301 non-null  object
 6   ap_lo        69301 non-null  object
 7   cholesterol  69301 non-null  object
 8   gluc         69301 non-null  object
 9   smoke        69301 non-null  object
 10  alco         69301 non-null  object
 11  active       69301 non-null  object
 12  cardio       69301 non-null  object
dtypes: object(13)
memory usage: 6.9+ MB
```

Figure 2: Dataset Information

From the bar charts we created, we learned that the weight distribution is slightly skewed to the left of 75 kilograms or 165.35 pounds (fig. 3). This makes sense as most adults are somewhere around 165 pounds depending on height. The age bar chart was slightly skewed to the right, but that was expected as the study only included adults and that patients with confirmed cardiovascular disease were most likely to be older (fig. 4). The height distribution bar chart was almost a bell-shaped curve centered at 162.5 centimeters (fig. 5). The center of the curve is almost exactly in between the average height for males and females, so the distribution seems reasonable. The bar charts revealed that the patient demographic is evenly distributed given the parameters of the dataset (only adults) and there are no outliers. We also learned through the exploratory data analysis that the data is approximately balanced, so we don't need to balance the dataset.

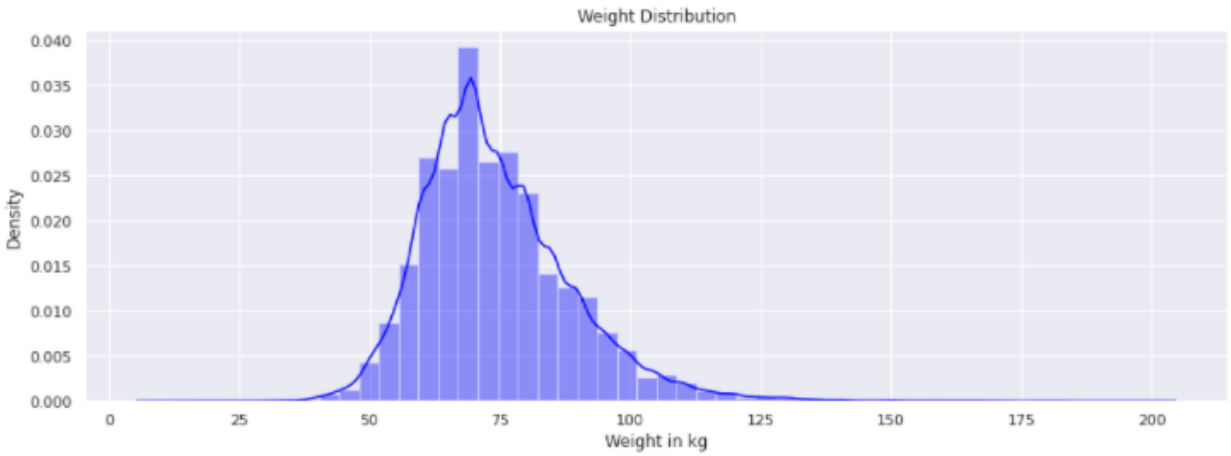

Figure 3: Weight Distribution

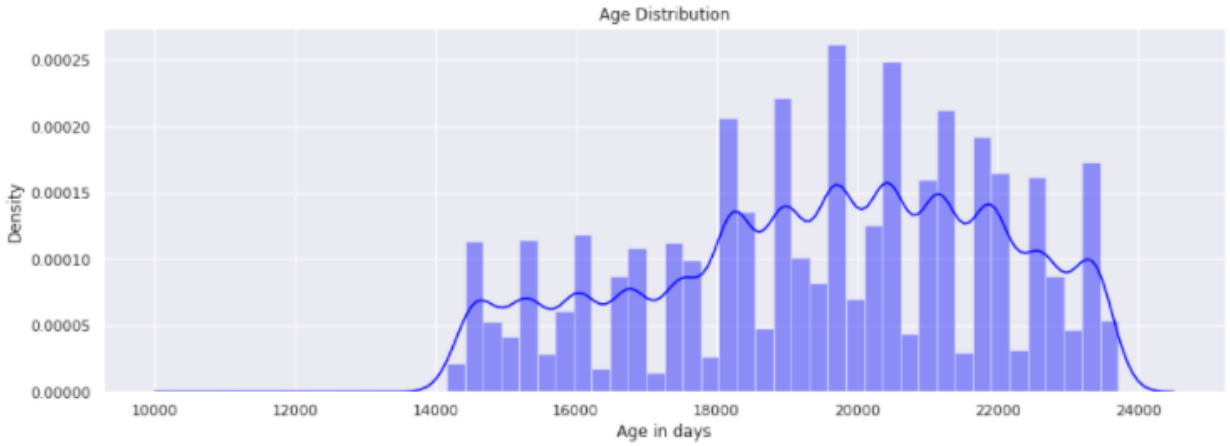

Figure 4: Age Distribution

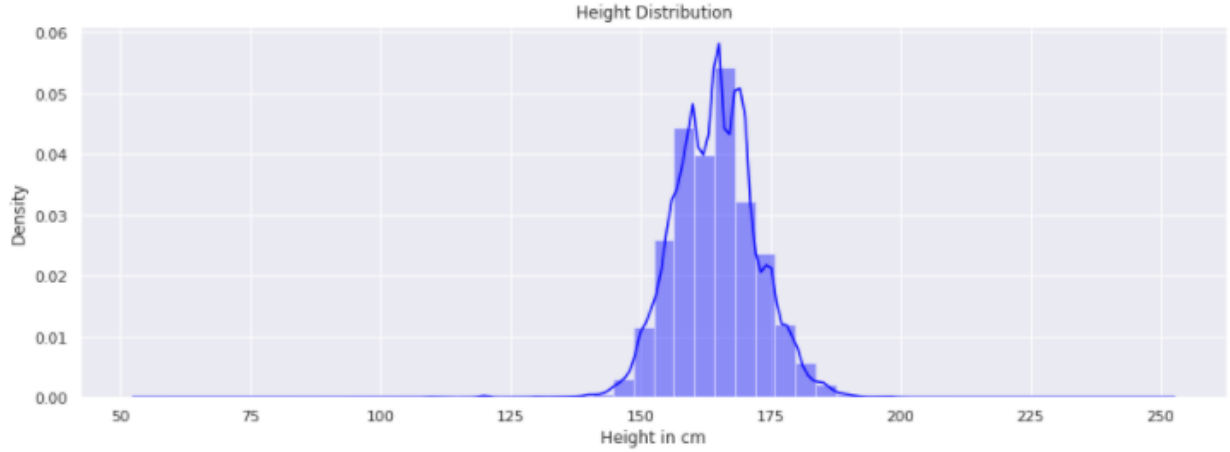

Figure 5: Height Distribution

We completed the exploratory data analysis phase by creating a heatmap that shows the correlations between all the features in the dataset (fig. 6). The most important part of the chart is the cardio row. The cardio feature is a binary variable that indicates the presence or absence of cardiovascular disease. The features that have the strongest correlation to cardio are the ones that are most predictive of a patient developing the disease. Age had the strongest correlation at 0.24, cholesterol was a close second at 0.22, and weight was the third highest at 0.18. Several features had negative or inverse correlations with cardio. For example, active, a binary variable that means a patient is either physically active or not, has a -0.036 correlation to cardio. This is to be expected as doctors have already established that physical exercise can help prevent cardiovascular disease. Smoke and alco, both binary variables that indicate if a patient partakes in smoking or alcohol consumption respectively, also have inverse correlations with cardio. This is a surprising insight as people are generally told that smoking and alcohol are bad for your health.

Additionally, there was an equal correlation between the patient's height and gender. This is to be expected as males are generally taller than females. The next highest correlating features were cholesterol and glucose. High glucose can result in high cholesterol, so you would expect to see a strong correlation between the two. What's interesting is that cardio has a strong correlation with cholesterol, but a low correlation with glucose. We expected to see a strong correlation between both features and cardio if high glucose truly does cause high cholesterol. However, this is not the case and is likely due to a variety of factors affecting cholesterol levels. Alcohol consumption and smoking also had a strong correlation. Typically, these two activities go hand-in-hand, so it is no surprise to see a correlation

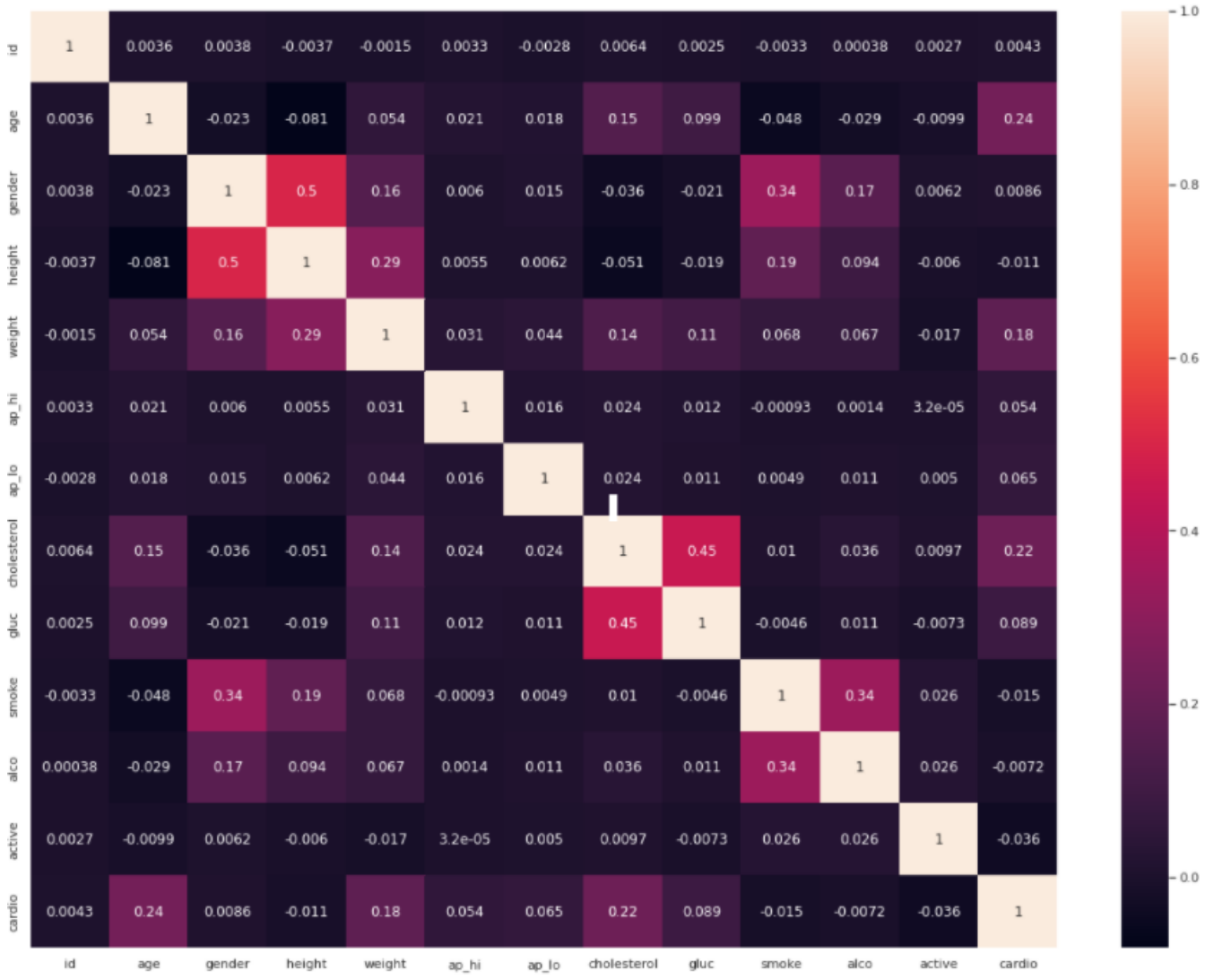

Figure 6: Heatmap

We created a confusion matrix for each model to compare and evaluate them. Overall, Random Forest Classifier was able to predict the true label most accurately. Decision Tree was a

close second. On the other hand, Support Vector Machines performed the worst at predicting the true labels.

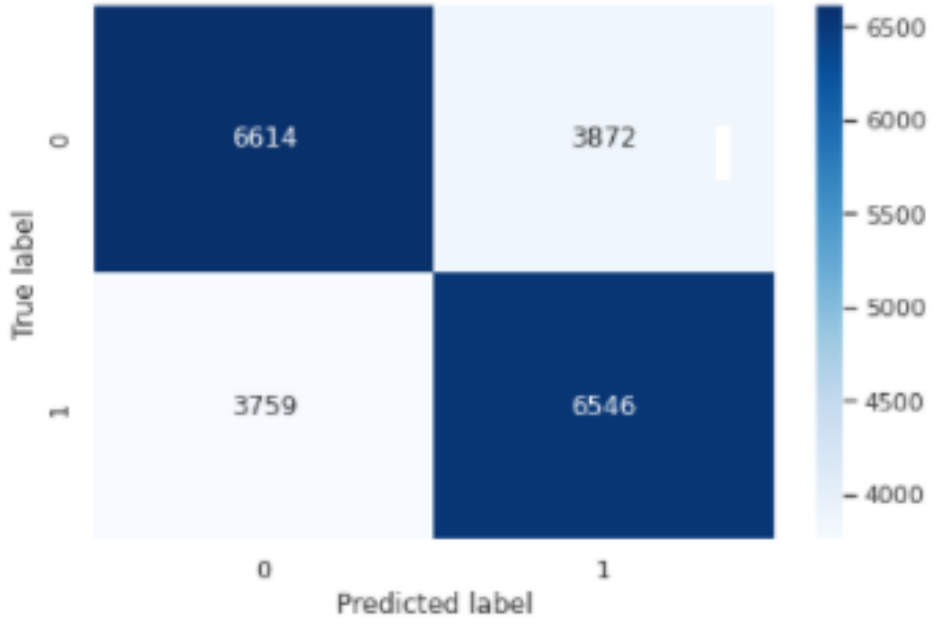

Figure 7: Decision Tree Confusion Matrix

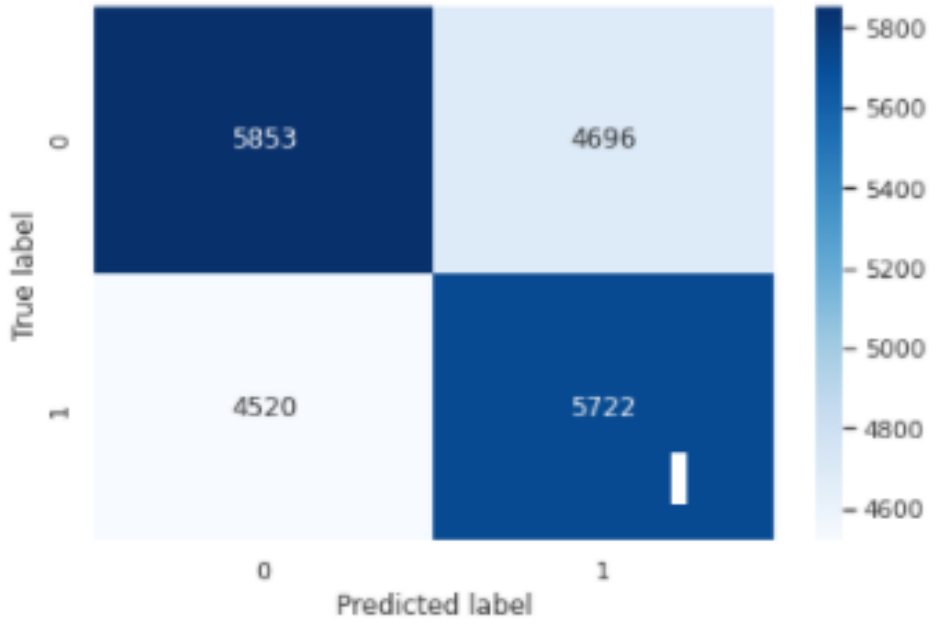

Figure 8: K-Nearest Neighbors Confusion Matrix

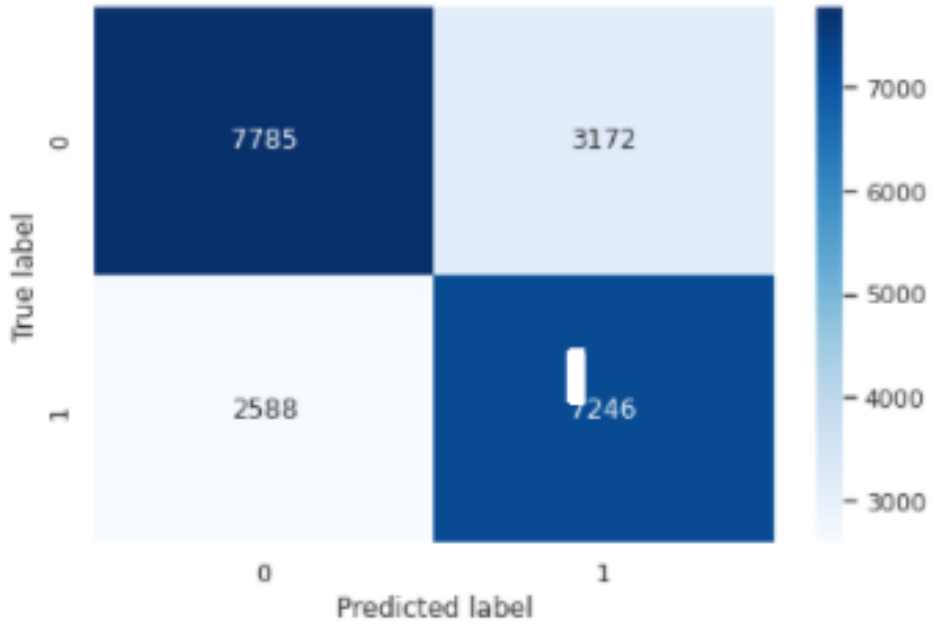

Figure 9: Random Forest Classification Confusion Matrix

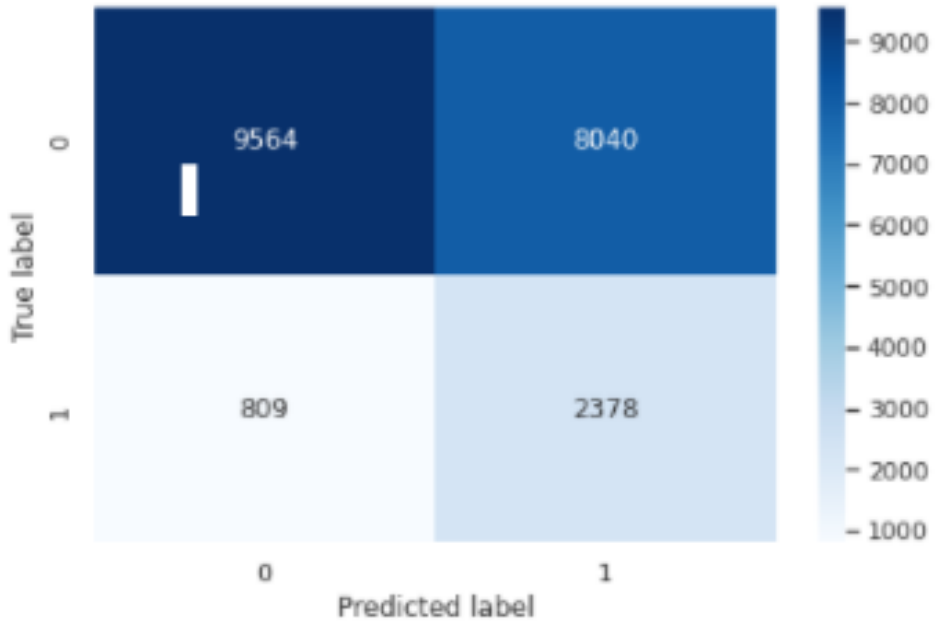

Figure 10: Gaussian Naïve Bayes Confusion Matrix

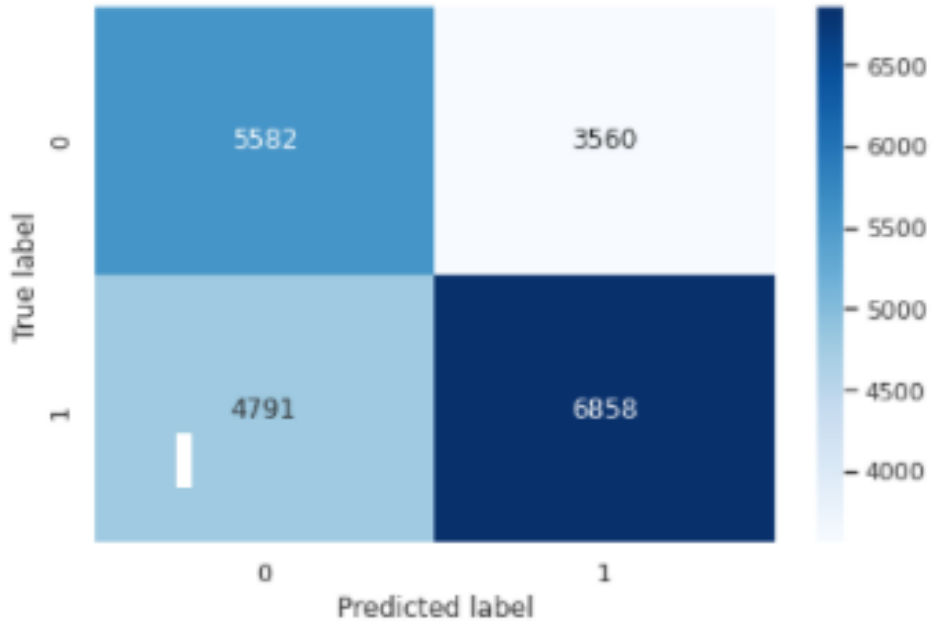

Figure 11: Support Vector Machines Confusion Matrix

Additionally, we calculated several metrics to compare the performance of the five algorithms. Those metrics are F1, accuracy, precision, recall, and cross-validation score (fig. 12). Random Forest Classification again performs the best using this evaluation. K-Nearest Neighbors and Support Vector Machines both have the worst result in F1 score and recall respectively. Gaussian Naïve Bayes scores worst for accuracy, precision, and cross-validation score, but best for recall. Decision Tree doesn't score worst or best for any metric; however it did produce the second-best confusion matrix.

	F1	*Accuracy*	*Precision*	*Recall*	*Cross-Validation*
Decision Tree	0.63386	0.63359	0.63287	0.63486	0.636796536797
KNN (k=5)	0.56327	0.56024	0.55897	0.56763	0.55436
Random Forest	0.72769	0.72193	0.70650	0.75019	0.72381
Gaussian Naïve Bayes	0.68040	0.55086	0.52785	0.95697	0.553989
Support Vector Machines	0.57272	0.60026	0.61452	0.53625	0.59540

Figure 12: Metric Comparison

As purely a visual aid, we created a diagram of our Decision Tree model (fig. 12). In order for it to be legible, we had to set the max depth to three but it originally printed out with approximately 50 levels. The most interesting part of the tree is that the root node uses the ap_hi feature, an integer variable that represents the patients systolic blood pressure. Blood pressure is very indicative of cardiovascular disease, but it was surprising as our heatmap did not show a correlation.

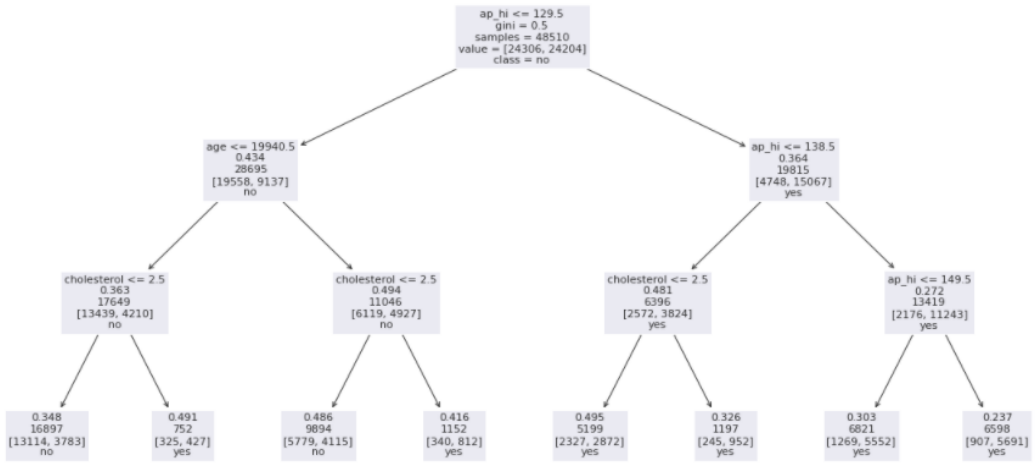

Figure 13: Decision Tree Visual

Related Work

As the dataset was found on Kaggle, many other people have attempted to create machine learning models to predict if a patient will have cardiovascular disease. As far as we are aware, our project is the only that compared several algorithms and assessed their performance. We tried to avoid looking at them as we wanted to create a project that was completely thought of by ourselves.

Conclusion

In conclusion, we found that the three highest correlating features to a patient having cardiovascular disease were age, cholesterol, and weight. The heatmap shows that there are also inverse correlations with cardio among which are smoke and alcohol use. The reason behind this is unknown but is certainly something we would like to look into.

The best performing algorithm was Random Forest Classification by quite a bit. This model produced the best confusion matrix and had the highest scores for the metrices. K-Nearest Neighbors has the potential to perform better if a different k-value was use. Gaussian Naïve Bayes might have performed better if we used a different type of Naïve Bayes algorithm that could handle continuous values better.

Overall, we created several models that are good at predicting if a patient will develop cardiovascular disease and found the features with the strongest correlation.

Bibliography

Bhadaneeraj. “Cardio Vascular Disease Detection.” *Kaggle*, 1 June 2020, https://www.kaggle.com/bhadaneeraj/cardio-vascular-disease-detection.

This is where we found the dataset and downloaded it from.

“How Common Is Heart Disease?: Heart Disease.” *Sharecare*, https://www.sharecare.com/health/heart-disease/how-common-is-heart-disease#:~:text=Heart%20disease%20is%20the%20leading%20cause%20of%20death,leading%20cause%20of%20disability%20in%20the%20United%20States.

We used this website to learn more about cardiovascular disease.

“SciKit Learn.” *Scikit*, https://scikit-learn.org/stable/index.html.

We used this website to understand how to implement various SciKit Learn (Sklearn) library functions.

Appendix

Link to cardio_data.csv

Link to our project code