PREDICTING CARDIOVASCULAR DISEASE

Angelina Amato, Rebecca Breland, Charlie Cryer, Braedyn Edwards, Magdeline Ng

CSSE/MA415 Machine Learning

19 May 2023

ABSTRACT

This paper examines the ability to predict Cardiovascular Disease (CVD), using machine learning techniques. K-Nearest Neighbors, Logistic Regression, Decision Trees, Random Forest, and Support Vector Machines were all analyzed on a dataset including 70,000 data records with features pertaining to an individual's health. Many preprocessing techniques as well as feature engineering and selection were utilized on the dataset in order to obtain the best possible accuracy scores. After examining both accuracy scores and false negative rates, Random Forest was determined to be the best classifier for this problem as it maximized testing accuracy while minimizing false negative rate. The most important features when predicting if a person has CVD are systolic and diastolic blood pressure, age, smoking, alcohol consumption, weight, and MAP.

1. INTRODUCTION

Around the globe, approximately 17.9 million individuals die from cardiovascular disease (CVD) each year [1]. In the United States alone, an individual dies from CVD every 34 seconds [2]. According to the World Health Organization (WHO), one of the most important things to do to prevent these deaths is to provide early detection to the affected individuals [1], which is why many health organizations have been attempting to train sufficient machine learning (ML) models to predict if an individual has CVD.

As a result, we decided to attempt to determine which ML model provides the best accuracy and minimizes false negative predictions. Additionally, we sought out to determine which features were most predictive of CVD in the hope of being able to provide guidance to individuals who want to avoid contracting it. Though there are many known factors that are correlated to CVD as seen in Figure 1, we also wanted to compare the feature importances of the various ML models that we ran to the features that are known to be highly correlated.

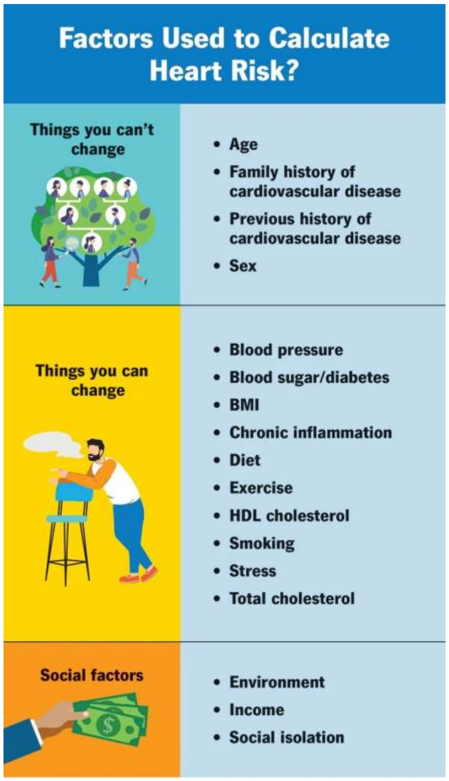


Figure 1: Known correlating factors to CVD [3]

Since CVD affects so many individuals and relies on many subjective factors like family history of CVD or stress, highly accurate predictions are hard to come by when performing a binary classification from a set of subjective and objective numerical factors. Additionally, CVD affects age groups differently, i.e. older age groups, thus outliers in relatively unaffected age ranges become harder to properly predict. The optimization of hyper-parameters and training of multiple classifiers to determine the most optimal estimator takes a large amount of computation time; however, if done properly, the results of this study could be built upon for creating a CVD screening tool in future work.

Identifying the most accurate classifier for systems like screening tools or other devices is crucial to the health of patients since we want to avoid false negative results. Testing many different classifiers allowed us to compare both the accuracy and the number of false negatives that were produced. This allowed us to have justification for our best classifier based on both of these parameters.

2. LITERATURE REVIEW

Before beginning our project, we did an extensive literature review to determine techniques we could utilize. One of the papers we pulled heaviest from was “Effective Heart Disease Prediction Using Machine Learning Techniques” by Chintan M. Bhatt et al. This was a study that describes the preprocessing and results of multiple estimators on a 70,000-record dataset. From this study, we used the dataset and multiple preprocessing techniques that worked well on that dataset. Our original dataset was around 1,000 records so this one provided a much larger pool of records to train on. We used their data cleaning method in reference to removing outliers and used similar engineered features. For instance, we engineered mean arterial pressure (MAP) and body mass index (BMI) since they are able to be calculated from features present in the dataset and MAP is highly correlated to CVD [4]. Utilizing the information from this source allowed us to add these features to our dataset.

Another paper that was utilized was “Significance of Visible Non-Invasive Risk Attributes for the Initial Prediction of Heart Disease Using Different Machine Learning Techniques” by Syed Immamul Ansarullah, et al. This paper studied the accuracy and error rate of many different classifiers in predicting CVD and the authors were able to conclude that a Random Forest is the best at predicting CVD [5]. We drew inspiration from this study by using the same classifiers, and some more, to see if we could reproduce, or at least corroborate, their results. To do this, we utilized a K-Nearest Neighbors Classifier, Decision Tree Classifier, Random Forest Classifier, and a Support Vector Machine Classifier just as Ansarullah et al. did in their study.

3. PROCESS

3.1 Data Source

For our data source, we chose to use a Cardiovascular Disease dataset from Kaggle [6] that has both numerical and categorical features pertaining to an individual’s health and information on the presence of CVD. This data was used in a binary classification problem, determining whether or not an individual has CVD.

The data set contained 70,000 records, of which 35,021 did not have CVD and 34,979 did, resulting in evenly balanced targets. The data set also contained 11 different features to classify someone with CVD: age, height, weight, gender, systolic blood pressure (BP), diastolic BP, cholesterol, glucose, smoking, alcohol intake, and physical activity. Of these features, age, height, weight, systolic BP, and diastolic BP are continuous, whereas the others are all categorical. Smoking, alcohol intake, and physical activity all represent two binary categories: yes or no. Cholesterol and glucose, however, are divided into 3 separate categories representing the differing levels of the compounds found in an individual’s body: normal, above normal, and well above normal.

3.2 Preprocessing

Considering the large size of our dataset with more than 70,000 records, *.dtype()* was used to check if Pandas correctly classified each variable as numerical or categorical. If it incorrectly classified them, appropriate adjustments were made. In such a large dataset with a large number of records, it was essential that the target variable, cardio, is balanced, with approximately equal numbers for each value. A heatmap was plotted for the correlation matrix for all variables to ensure no two variables are highly correlated since this creates problems related to multicollinearity and overfitting of the model, the heatmap can be seen in Figure A.1. For this dataset, all variables have a correlation value of <0.1 except height and weight, which is expected to be slightly more correlated. *.describe()* was used to observe the minimum and maximum values for numerical features, outliers that fell outside of the Interquartile Range (IQR) were removed. Additionally, rows with illogical values were removed, specifically removing negative ap_hi and ap_lo values, entries where ap_hi < ap_lo, and ap_hi values above 350. We finished our preprocessing after removing a total of 3511 rows, leaving 66489 entries in our final dataset.

3.3 Feature engineering

The first form of feature engineering we performed was one-hot encoding of the categorical features. We did this for all the algorithms because it was necessary to properly predict and interpret the results of our classifiers. Depending on the demands of the algorithm, there were several other methods of feature engineering employed. For instance, we used second degree polynomial feature engineering and created standardized and nonstandardized versions of the engineered dataset, in order to cater to different needs of the classifiers. The polynomial feature engineering produced 231 different features; however, in an attempt to shorten computation time, we removed some unnecessary features that were created due to conflicting categorical features. In the end, we had 217 features to train our classifiers to predict CVD on .

Additionally, based on the paper by C.M Bhatt et al. we engineered two features: Body Mass Index (BMI), a function of height and weight, and Mean Arterial Pressure (MAP) a function of systolic and diastolic blood pressure. These two features are highly predictive of cardiovascular disease [4]. To create the BMI feature, we utilized equation 1, and for the MAP feature we used equation 2 since we had the underlying data to properly engineer these features.

$$BMI \ = \frac{weight}{(height/100)^2} \tag{1}$$

$$MAP \ = diastolic\ BP \ + \frac{systolic\ BP - diastolic\ BP}{3} \tag{2}$$

After running into some issues with run time, we reviewed the feature engineering process and made variant datasets with fewer features. Since the majority of the features were combinations of one-hot-encoded features, we cut down the total number of features by not including interactions between categorical features. In total, we made a version with everything except a few contradictory features (like both male and female), a version without interactions involving categorical features (continuous interactions only), and finally a version of the clean dataset with no feature engineering applied. These data sets had 218, 38, and 22 features respectively, including the target.

3.4 Classifiers and tuning

K-Nearest Neighbors

One of the first estimators we experimented with was the K-Nearest Neighbors Classifier. This estimator is a supervised method that classifies data points based on a certain number of their closest neighbors. This is a similarity model and works by comparing data based on a distance calculation. Before running the estimator data was standardized to ensure all features contribute equally and the engineered dataset with feature interactions was used. Since this estimator takes a long time to process, Principal Component Analysis (PCA) was used to eliminate some of the features. A total of 37 principal components were used and these explained 95% of the variance within the dataset.

There are three different hyper-parameters that we chose to vary to determine the best fit: number of neighbors, weighting, and distance calculation. We completed a grid search with the number of neighbors being varied from 1-201, weights of distance and uniform, and distance calculations of city-block (p=1) and Euclidean (p=2). GridSearchCV was utilized with a 5-fold cross validation to find optimal parameters in our grid while also cross-validating. The result of this search can be seen in Figure 2. The optimal parameters for our dataset were found to be 121 neighbors using uniform weighting and city-block distance calculation.

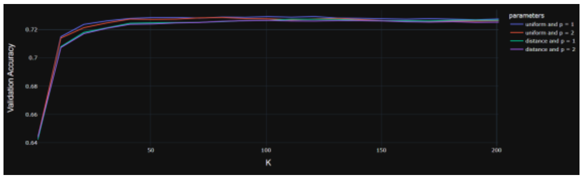


Figure 2: KNN validation accuracy for different hyper-parameters.

Logistic Regressor

The Logistic Regressor is a classifier that utilizes multiple linear regressors, one for each class in your target. It is a way of fitting a linear model to classification problems and utilizes probability based on the linear models. The standardized dataset with feature interactions was used in order to evaluate this estimator since computation time was not an issue.

The data was first split using train_test_split and then cross validated using 5-fold cross-validation in order to determine the mean training and validation accuracies. Then the classifier was tested on the portion reserved by the train-test-split to obtain the testing accuracy. For this estimator, the importance of each of the features is able to be determined by the coefficients that they have in each of the linear regressors.

Decision Trees

We also used Decision Trees classifiers to predict cardiovascular disease. Besides feature engineering, there were two more steps we performed: feature selection and grid search. Overall, we made two attempts at making an optimal tree; however, the first one had a better result than the second. There are three differences between the two attempts, one for each of the two preprocessing steps and one in the data preparation.

For the first attempt, we had not yet finalized the datasets that all the classifiers were using. As a result, the first difference is that this attempt used a unique version of the dataset created in its own Jupyter notebook, while the second one used our fully feature engineered dataset. The second difference is that feature selection on the first attempt was performed with a logistic regressor, whereas feature selection in the second attempt used a Decision Tree. Both were performed by a variety of training models with incrementally larger feature sets. We would select the model that has the highest improvement to accuracy. The third and final difference was that the grid search on the first attempt only tried to optimize values for max_depth. However, the second attempt expanded the optimization to max_depth, min_samples_leaf, and min_samples_split. From this grid search, we could see a classic example of overfitting: as the depth of the trees and the train accuracy goes up, the test accuracy goes down. As a result, we determined that the optimal tree depth was 6, and the feature selection process identified 74 relevant features from the first attempt.

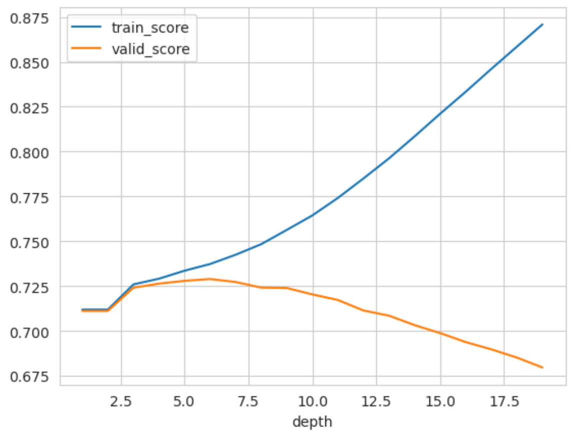


Figure 3: Grid search on max depth of Decision Tree, training and validation accuracies.

Random Forest

The next estimator that we experimented with was the Random Forest Classifier. This classifier applies bagging to a large number of Decision Trees to create predictions with low variance. It does this by averaging the results of the various Decision Trees to create a new prediction since averaging reduces variance with independent quantities. With a large data set, the classifier runs relatively quickly if the hyper-parameters have been tuned; however, with a large number of features, we found it best for time efficiency to utilize the 37 principal components produced by PCA. Though, for best accuracy we found it was best to use the engineered feature set with optimized hyper-parameters.

To tune a Random Forest Classifier, we needed to optimize the number of Decision Tree estimators, as well as the maximum depth of the Decision Trees using a grid search and 5-fold cross validation. If we were training with the engineered feature set, the grid contained a range of 10 to 2,000 different estimators and a range of 1 to 20 differing maximum depths. The grid search determined 210 estimators with a maximum depth of 10 provides the best results as seen in Figure 4.

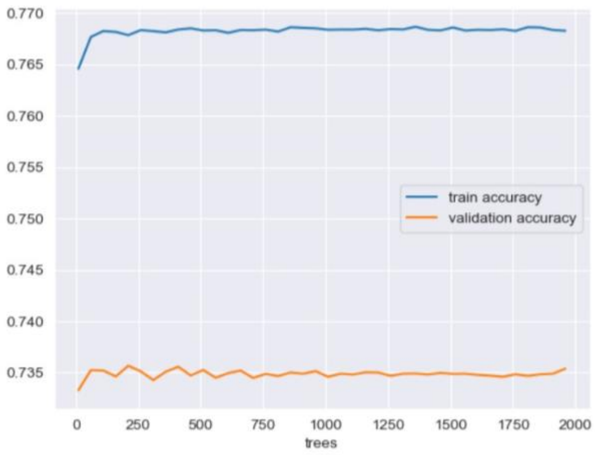


Figure 4: Random Forest grid search with the engineered feature set

Support Vector Machines

The biggest advantage of support vector machines is that the more complex the data, the more accurate the predictor will become. As the number of dimensions increases, other classification algorithms might suffer from the curse of dimensionality where the data becomes sparse and the classifier's performance degrades. Our feature engineered dataset has a large number of features, so a SVM can use a small subset of training instances (support vectors) to define the decision boundaries, and the classification is determined by a relatively small number of support vectors rather than the entire dataset.

There are three hyperparameter values we had to set, namely the Regularization Parameter C, the Kernel Function and the gamma parameter. GridSearchCV was utilized with three-fold cross validation to find optimal parameters in our grid and was run on the standardized feature engineered dataset without category interactions. The grid search determined a C of 100 and a Gamma of 0.0001 obtain the highest validation accuracy and are therefore the optimal parameters.

4. EXPERIMENTAL SETUP AND RESULTS

After processing the dataset, we were left with 66,489 total data records each with 217 features. Of the 66,489 data records, 33,927 do not have CVD and 32,562 have CVD. The dataset is equally balanced for our target leading to a baseline accuracy of 0.510. This data was split into training and testing sets using a train test split with 20% reserved for testing. We stratified on y and used a random state equal to 0 to ensure the dataset split the same way each time. In total, the training set had 53,191 records with 27,141 not having CVD and 26,050 having CVD. The testing set had 13,298 data records with 6,786 not having CVD and 6,512 having CVD. Each set was evenly balanced over the target. The baseline accuracy was beaten by all of our estimators, the results of which can be seen in Table 1.

Table 1: Training, validation, and testing accuracies for predicting CVD on different estimators.

Estimator	**Training Accuracy**	**Validation Accuracy**	**Testing Accuracy**
KNN	0.732	0.729	0.725
Logistic Regressor	0.730	0.730	0.730
Decision Trees	0.737	0.729	0.735
Random Forest	0.768	0.736	0.732
SVM	0.744	0.721	0.728

Since all of the estimators had similar testing accuracies it was hard to determine with confidence the best estimator. To truly decide we also needed to look at the false negative rates for each estimator which are listed in Table 2. The false negative rate was calculated by dividing the number of false negatives by the total number of mispredictions. Since this is a medical application we would rather see false positives. If we have too many false negatives, we would be telling people they are not at risk when they are. A false negative would lead individuals to not seek a doctor and possibly not determine they have CVD until it is too late.

Estimator	**False Negative Rate**
KNN	0.634
Logistic Regressor	0.600
Decision Trees	0.636
Random Forest	0.597
SVM	0.588

Table 2: False negative rates for each of the estimators tested

5. DISCUSSION

K-Nearest Neighbors

KNN performed relatively similarly to our other estimators, as seen in Table 1. To determine the strengths or downfalls of this classifier comparatively, we have to examine the results more closely. A confusion matrix was created of both the training and the testing predictions, the results of the testing data are shown in Figure 5. Based on this confusion matrix we can see that the estimator is predicting more false negatives than false positives. Due to the nature of this project, we would want false negatives to be minimized. This is not ideally what we would want to see when analyzing how our estimators operate.

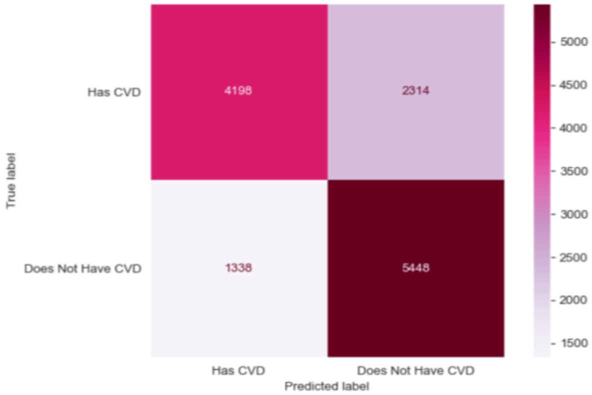


Figure 5: Confusion matrix for test set predictions on the KNN classifier.

Since KNN uses standardized data and simple distance calculations, we cannot determine feature importance for this classifier. Though, since this was completed by using PCA we can analyze the principal components and determine what was most important in creating those. When a pair plot was completed of all the principal components only one component was truly distinguishable over our target. PC1, which accounted for 21.7% of the variance in the data is pictured in Figure 6. Blue represents the individual not having CVD and orange represents the individual having CVD. From this plot, it is clear that the two target classes are fairly distinguishable which is good. The highest loadings for this principal component are combinations of MAP, no alcohol, no smoking, and both systolic and diastolic blood pressure. These features and combinations of them are what divide the dataset well over our target. Having higher values of these leads an individual to be more likely to have CVD. A large amount of the variance in the data was explained by these features leading them to be rather significant.

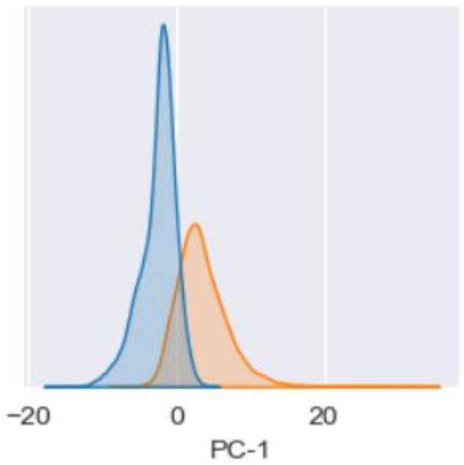


Figure 6: Plot of PC-1 colored with orange representing individuals with CVD and blue representing individuals not having CVD.

When looking at specific examples where the KNN estimator incorrectly predicted testing data, 75% of their PC1 values lay between -2.07 and 1.39. This is compared to the entire training dataset where 75% lay between -2.39 and 2.36. It is clear that this principal component was influential and helped to properly predict points where they were distinguishable. Examples at or around zero for this feature were commonly mispredicted.

Logistic Regressor

In order to determine the most significant features for the Logistic Regressor, the coefficients of the linear regressors must be examined. The results of this examination can be seen in Figure 7. The most significant features were weight, high cholesterol, and smoking which is understandable considering the context of our problem. Since the categorical features were one-hot-encoded, the diagram presents smoke_1 which means the patient smokes and cholesterol_3 which means the patient has well above normal cholesterol. These are typically the top risk factors when discussing CVD in medical practice and therefore the information the classifier gave makes sense.

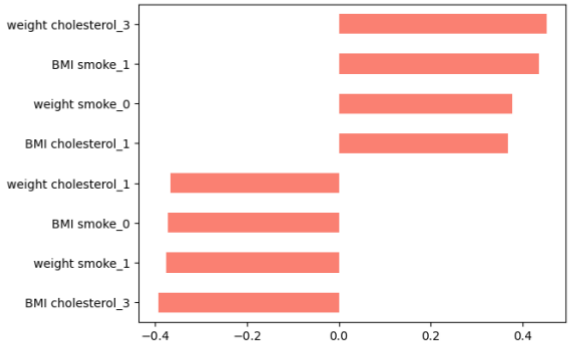


Figure 7: Significant features for predicting CVD from the Logistic Regressor.

A confusion matrix was also used on the testing set of the data used for the Logistic Regressor in order to compare this estimator against the others. Figure 8 below shows that the false negative rate is 60%. This means that the classifier is getting more false negatives than false positives and as mentioned before, this is not ideal.

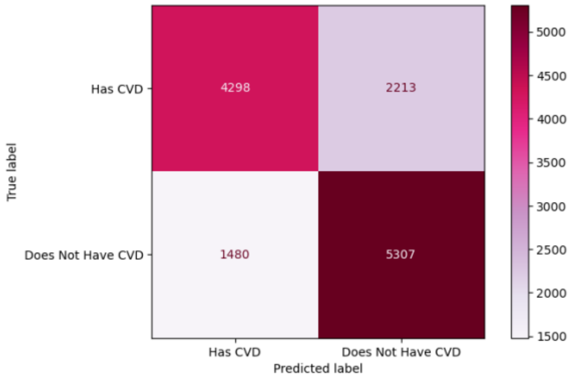


Figure 8: Confusion matrix for the Logistic Regressor.

Decision Trees

Overall, the best Decision Tree estimator had an accuracy of 0.735. Taking a look at the optimal tree, the most relevant features were systolic blood pressure, age, cholesterol, MAP, and BMI. It turns out that the changes made to the second attempt, like feature selection with Decision Trees and grid search with more parameters didn't help, and the first attempt was better overall. It's worth noting that the feature selection process for the second attempt selected strange polynomial features, like being both male and female as important. With the setup of our dataset, that column would always be zero, making it not very useful for predictions. This says to us that there may be something flawed in the second attempt's feature selection process. Returning to the first attempt, we also generated a confusion matrix for the best Decision Tree classifier, and found that it had some shortcomings in its false negative rate. This gave us an important view on how the classifier was performing even with the relatively high accuracy score.

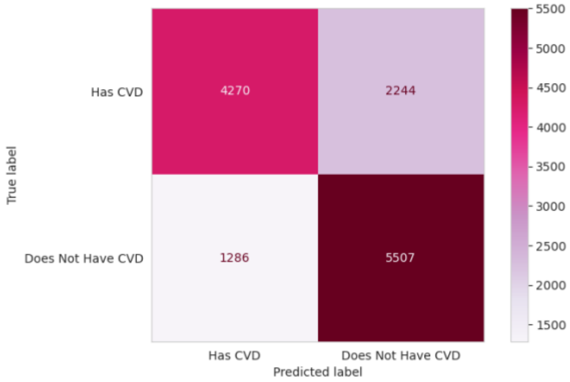


Figure 9: Confusion matrix for best Decision Tree

Random Forest

Since the Random Forest classifier performs its own form of feature selection, its most impactful features differ from those of the Logistic Regressor. For instance, as seen in Figure 10, systolic and diastolic blood pressure, ap_hi and ap_lo respectively, are the two most important features with age and MAP also being important. This makes sense in terms of predicting CVD considering that individuals diagnosed with CVD are generally older and have higher BP resulting in a higher MAP score. These feature importances also match those of Ansarullah et al.'s study, which claim that systolic BP, diastolic BP, and age are the most predictive features [5]. Even though Ansarullah et al.'s study did not include the MAP feature, it makes sense for it to be included as one of our most important features considering it is made of both systolic and diastolic BP.

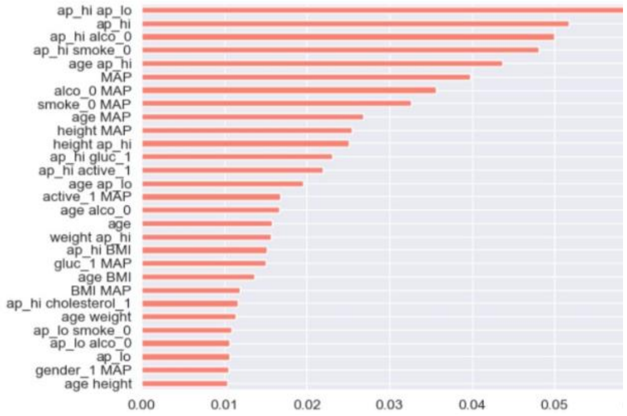


Figure 10: Significant features from the Random Forest

When analyzing the results of the Random Forest, we discovered that the Random Forest tended to predict false negatives with its mispredictions. 59.66% of the incorrect predictions made turned out to be false negatives, while 40.34% were false positives as seen in Figure 11. It is not ideal that our Random Forest tends to make false negative predictions compared to false positives due to the medical nature of this problem. As a result, we tried to determine the root cause of the false positive. It is believed to occur when an individual's systolic and diastolic BP is within a normal range but still have CVD, whereas a 'typical' individual with CVD has higher systolic and diastolic BP. Due to the high importance of both systolic and diastolic BP in the classifier, we believe these normal BPs are what is causing the false negative prediction. For the false positives, we believe the inverse is occurring where individuals without CVD have high systolic and diastolic BP are getting classified as having CVD.

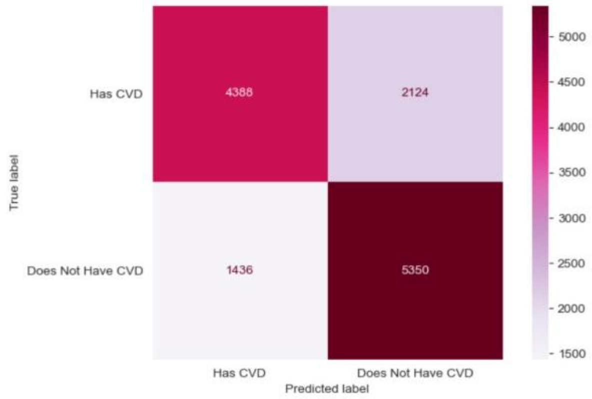


Figure 11: Confusion matrix for the Random Forest

Support Vector Machines

For the Support Vector Machines (SVM) classifier, the optimal hyperparameters were previously determined using grid search and cross-validation. These hyperparameters were then used to predict the target in the test set produced by train-test split, producing the following confusion matrix.

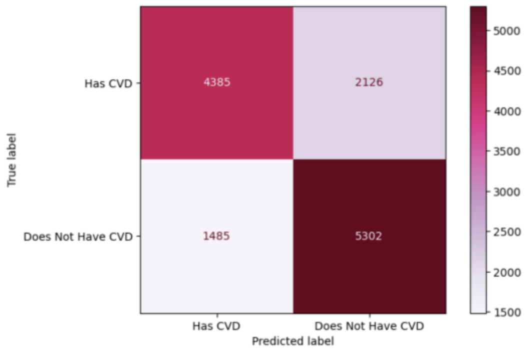


Figure 12: Confusion matrix for the SVM

SVM's test accuracy performed similarly to our other estimators, as seen in Table 1. However, it produced notable results by achieving the lowest false negative rate, which is desired for our use case of predicting the risk of CVD. The SVM was performed on the dataset without feature interaction and three-fold cross-validation due to its computational intensiveness. With better computational capacity, the SVM has the potential to minimize its false negatives even further with higher prediction accuracy. Since this is a comparison based model, feature importance is not interpretable.

6. CONCLUSIONS AND FUTURE WORK

After reviewing both the testing accuracy and the false negative rates for each of our estimators, it was concluded that Random Forest was the best estimator. Even though it had lower accuracy than Decision Trees, it had a lower false negative rate making it more desirable. It had a slightly higher false negative rate than SVM, but the accuracy of SVM is much lower than Decision Trees. Random Forest was found to be optimal for both of the parameters we were reviewing. When looking at feature importance both the Random Forest and Decision Trees came to similar conclusions with systolic and diastolic blood pressures, age, smoking, alcohol, and MAP being some of the most important features. The linear regressor came to slightly different conclusions with weight being highly important. Overall, all of these features listed were concluded as the most important when predicting whether an individual has CVD or not.

If given more time to complete this project, it would be beneficial to experiment with more classifiers such as XGBoost. Implementing more advanced techniques deep learning techniques, such as neural networks, could also potentially lead to more accurate predictions due to their ability to process data through hidden layers between the network's input and output nodes. Due to the complex nature of predicting CVD from both subjective and objective information, we believe that it would be best to gather more features. For instance, additional features such as income, environment, or family history could be added to our data set and analyzed for their feature importance.

Additionally, using AI image models for ECG or MRI scans could also help in making predictions since it is subjectively easier to predict whether an individual has CVD compared to numerical and categorical data. New features and images such as these combined with deep learning techniques is crucial to achieve the most accurate predictions. Since predicting whether an individual has CVD needs to be as accurate as possible to prevent a false diagnosis, we believe we would pursue these next steps. Once the truly most accurate classifier is found, we would like to then start developing this into a screening tool that health organizations could potentially use for early detection of CVD.

7. KEY CHALLENGES AND LESSONS LEARNED

Throughout this process, we encountered many obstacles that we had to work with, with the main one being the size of the dataset and processing time. We had 66,489 data records after processing, each with 217 features. With this large of a dataset, a lot of time was needed to train our estimators. To work through this challenge we implemented items like feature selection and Principal Component Analysis (PCA) to lower the number of features. In most cases, this worked well and allowed us to run estimators that were taking too long beforehand. Some estimators such as Gradient Boosting Trees (GBT) were still unable to complete in a 12-hour time frame; therefore, became infeasible for the scope of our project. This obstacle taught us to utilize the different tools available to make estimators run more efficiently, but that it might not always be possible to run every estimator on such a large dataset.

Another issue we ran into when completing this project were the features available to us. Typically when diagnosing CVD in a medical setting, the doctor would look at blood work panels or ECG and MRI results. None of those things were present in our dataset. Additionally, features such as family history of CVD or social factors such as income were not included. These are other features that are often used to calculate risk for CVD and therefore could have helped in our estimators' accuracy [features]. To work around this issue, we made sure to use and modify all available features, even going as far as to add features such as MAP and BMI that could be calculated from the features we did have.

The final major challenge we encountered was similarity between estimators. Though this does not seem to be a huge challenge, it was harder to make conclusions at the end of our project. As seen in section 4, most of our estimators have very similar accuracies, with decision trees being the highest by a small margin. If we were to compare on just accuracy alone, we would not have been able to be confident in our results. To work through this challenge, we decided to also look at false negative results. Since this is a screening tool we want to eliminate false negatives. With this additional parameter it was easier to determine a best performing estimator.

8. REFERENCES

[1] "Cardiovascular diseases," World Health Organization,https://www.who.int/health-topics/cardiovascular-diseases#tab=tab_1 (accessed May 17, 2023).

[2] "Heart disease facts," Centers for Disease Control and Prevention, https://www.cdc.gov/heartdisease/facts.htm (accessed May 17, 2023).

[3] "Cardiac risk calculator and assessment," Cleveland Clinic, https://my.clevelandclinic.org/health/diagnostics/17085-heart-risk-factor-calculators (accessed May 16, 2023).

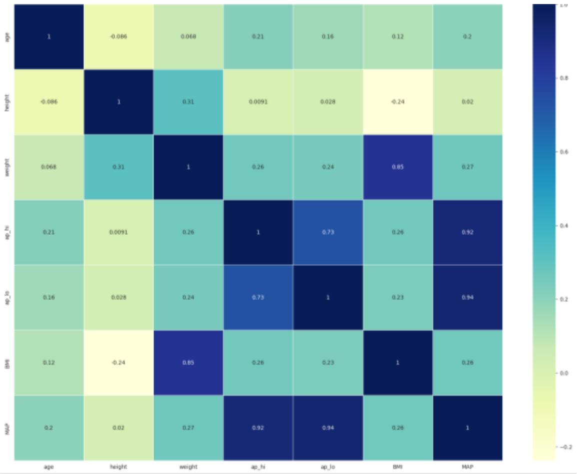
[4] C. M. Bhatt, P. Patel, T. Ghetia, and P. L. Mazzeo, "Effective Heart Disease Prediction Using Machine Learning Techniques," Algorithms, vol. 16, no. 2, p. 88, Feb. 2023, doi: 10.3390/a16020088. [Online]. Available: http://dx.doi.org/10.3390/a16 020088

[5] S. I. Ansarullah, S. M. Saif, P. Kumar, and M. M. Kirmani, "Significance of visible non-invasive risk

attributes for the initial prediction of heart disease using different machine learning techniques," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–12, 2022. doi:10.1155/2022/9580896. [Online]. Available: http://dx.doi.org/10.1155/2022/9580896

[6] S. Ulianova, "Cardiovascular disease dataset," Kaggle, https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset (accessed May 16, 2023).

9. APPENDIX



	age	height	weight	ap_hi	ap_lo	BMI	MAP
age	1	-0.086	0.068	0.21	0.16	0.12	0.2
height	-0.086	1	0.31	0.0091	0.028	-0.24	0.02
weight	0.068	0.31	1	0.26	0.24	0.85	0.27
ap_hi	0.21	0.0091	0.26	1	0.73	0.26	0.92
ap_lo	0.16	0.028	0.24	0.73	1	0.23	0.94
BMI	0.12	-0.24	0.85	0.26	0.23	1	0.26
MAP	0.2	0.02	0.27	0.92	0.94	0.26	1

Figure A.1: Heatmap representing correlation values between continuous features.

{More instructions:

The first and foremost rule of professional writing is not to skimp on detail. Leaving it until the last minute is not an excuse to rush through a report.

You will be evaluated on:

Writing mechanics: Is it free of typos and errors in writing mechanics: spelling, grammar and punctuation?

Proofread your paper slowly. It helps to have your partner proofread what you wrote and vice-versa. It should take 20 minutes to read carefully and note any errors – anything less and you are just skimming.

For any section written by non-native English speakers, we suggest having a native speaker help with this final step.

Organization: If you followed or adapted this standard template, you should be OK.

Clarity, Conciseness, Professionalism: Writing should be professional, clear and unambiguous, without exaggeration, and not unnecessarily wordy. Informal expressions and slang (e.g., "a lot", second-person "you") not used. Passive voice not used excessively. Specifically:

- *Don't* use contractions. (I hope you appreciate the irony of this statement.)
- You are allowed to write in the first person, but write factually, not emotionally. Saying a result is "weird" or that you are happy it works is valid (and I welcome you to communicate that to me offline) but not appropriate for a formal paper.
- Know when to use amount vs. number, less vs. fewer, and much vs. many.

A good, get- to-the-point resource for writing well is the Handbook for Technical Writers and Editors (http://www.sti.nasa.gov/publish/sp7084.pdf).

Effort expended: This is a bit subjective, but one one hand, if we think you worked hard on a challenging problem, you will get full credit here. On the other hand, if it ended up being a trivially easy problem and you did little work, you will not.

Experimental code: Make sure your final code submission is submitted and documented. If there are multiple files or any special way to set up the data to run your code, include a README to explain how we should do this and where we should start.

Aesthetics: Make sure all figures, tables, and equations are large enough and colors clear enough to read easily. Each should have a number and be referenced by number in the text, even if it seems obvious which figure you would mean without the number. See the next paragraph. Spacing consistent. If you need a wide diagram that spans multiple columns, they look best on the top or bottom of a page; you will probably need to insert a section break before and after it so that you can set the number of columns differently on each section to make it happen.

Always refer to figures, equations, and tables by number. For example, "Figure 6 shows the foo-ness of the bars. The resulting accuracy of the detector is greater than 90% (Table 2). We figured that out using Equation 1" And of course, include numbers in each caption. Equations do not need captions, just numbers to the right, as in:

$$A = \pi r^2 \quad (1)$$

When you are ready to submit it:

1. Convert to MS Word format. (Google docs are good for collaborative writing but Word is a better archival format, and we prefer it to pdf since we can easily add final comments and send them back to you at the end of the term.
2. Scan through it quickly looking for obvious formatting errors (a header on the bottom of a page, a figure or table that was split across columns or pages). Fix them. Then delete this final set of instructions from your report and submit it!

} % end instructions