CS-4641 Proposal Heart Disease Classification

Cayman is a clean, responsive theme for GitHub Pages.

View on GitHub Download .zip Download .tar.gz

Proposal

Introduction/Problem Definition

Given that the leading cause of death continues to be cardiovascular related diseases, more work is needed to improve accuracy in diagnosing these diseases. Previous work done in the field used various machine learning algorithms [2] to identify heart disease risk based on factors based on chest pains, age, and gender. Neural networks [4] have also been utilized to detect early signs of heart disease. There is also literature detailing the preprocessing of data related to ischemic heart disease by pruning extreme outliers [6], using DBSCAN to remove outliers, utilizing gradient boosting [1], addressing data imbalance [3], and data imputing for missing values [5]. Given the necessity to more accurately diagnose cardiovascular disease risk, our team seeks to utilize some of these techniques and hopes to improve in accuracy by controlling hyperparameters.

The data set the team is using is from Kaggle, which features factors such as age, sex, fasting blood sugar, angina, and cholesterol levels. Some features are continuous or discrete. Each data point is also identified with a target, for if HD is present or not.

Dataset Link

The leading cause of death in the developed world is heart disease. Therefore, additional work needs to be done to identify those at risk, based on a variety of non-invasive characteristics. This would allow for identification and warning as a preventative measure for people to take action on. Using the dataset, prediction of which patients are most likely to suffer from heart disease will be derived using the assigned features, so that when a new data point (a patient) is introduced to the model, it can be classified accurately, allowing the patient to act.

Methods

Data Preprocessing

To enhance our data quality and improve future model accuracy, outliers were removed according to an Interquartile Range method. Outliers can skew model performance, and to reduce that effect, the IQR was calculated across all data. From this, if a data point fell below Q1 - 1.5 * IQR (where IQR = Q3 - Q1), and (Q3 + 1.5 * IQR). Once identified, these outliers were removed from the dataset [6].

This method was chosen as it provides a high resistance to outliers, which is critical for an algorithm like this as misclassification must be avoided when dealing with highly sensitive datapoints. Furthermore, IQR does not assume a distribution, meaning it can be applied when the distribution is unknown, with this leading to its simplicity in calculation. If the model were expanded to a much larger training set, a reduction in computational complexity would be advantageous.

Converting categorical data to numerical format was also used for model performance through the implementation of ordinal encoding. Specifically, the target of is heart disease was present or absent was converted to 1 for present or 0 for absent [5].

Without parsing through every single data point, being able to handle any missing data points is used to increase the robustness of the preprocessing step. To ensure completeness of data, the team employed the K-Nearest Neighbors algorithm to handle missing values, with K = 5 However, we did not have any missing values from our data set.

Each value was also further modified through a min-max scaling process, which led to minimum values being assigned 0, and maximum values being assigned a 1. Scaling features normalize the data distribution to help improve model accuracy [5].

Finally, our dataset was split into 80% training data and 20% testing data. The rationale behind this split is that we leave behind a substantial portion of the data for unbiased evaluation. This helped us get a better picture of model accuracy on unseen data points.

Models

Linear Kernel Support Vector Machine (SVM)

The first algorithm the team used was an SVM with a linear kernel, which is an ML algorithm that creates a decision boundary to class data points. This decision boundary is a hyperplane in a multidimensional space to maximize the distance between the closest data points from the two classifications. SVM’s are frequently used in Medical Diagnoses, and in cases where efficiency and simplicity are critical [7].

The benefits of using an SVM is that they are relatively easy to understand and interpret, with visualization of how the model is making decision is also easy to understand. Training SVM’s is relatively computationally efficient, easily expandable to higher number of features for datasets.

Because of its simplicity, the risk of overfitting is also reduced. Additionally, important features are quickly weighed more and provide insight into the most important features as they correspond to the classification problem [7].

Random Forest (RF)

The second algorithm the team used was random forest, which is an ensemble algorithm that uses several decision trees to classify data points with a majority decision. Combining many decision trees helps reduce overfitting and improves the model's robustness against noisy data. Random forests are known to have previously worked successfully when classifying heart disease in patients, which was another reason we selected the model [2].

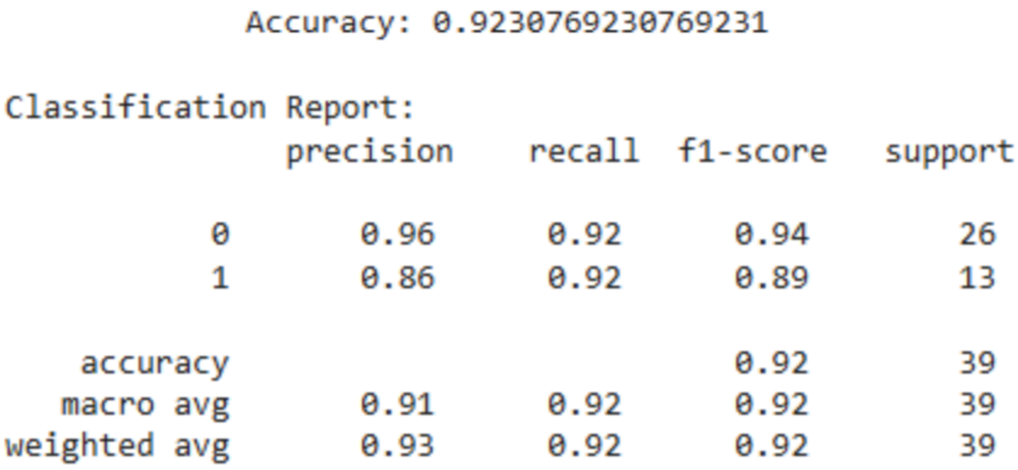
Multilayer Perceptron (MLP)

The third and final algorithm used by the team was a Multilayer Perceptron (MLP). An MLP is a machine learning algorithm that implements a feedforward neural network with nonlinear activation functions. When several perceptrons are not only densely packed together but also layered together, they can capture non-linearities in trends, with the final perceptron being used to identify and classify the results of the linear combination of outputs of these perceptrons [9]. It was determined that using a sigmoid function in the last layer to classify the results, ADAM as the optimizer, and binary cross entropy to evaluate loss. The sigmoid function scales and separates the resultant values between 0 and 1, which allows for easy classification and modification to the thresholds for classification. ADAM uses an adaptive learning algorithm, and while this is computationally more expensive, due to the size of the data set being used it was determined that this would be the most appropriate optimizer implementation. Finally, cross entropy loss is used over hinge loss because of its ability to be generalized, which is lacking in hinge loss as it cannot be used with differentiable methods. MLP was selected as a general model to see how it would capture complex non-linear relationships in the health data available.

Results

SVM

After using sklearn's train_test_split() function, the data was concisely split into a ratio of 80% training and 20% testing. Then, the team used the SVC class from sklearn to generate a linear kernel SVM classification model. After fitting training data, a comparison between the provided results and the testing data was made, which yielded an accuracy of approximately 92% and a macro average recall score of 0.92. The accuracy metric provides a sense of how the model performs in correctly identifying cases of heart disease and no heart disease. The recall metric is examined here because one of the team’s objectives is to reduce the occurrence of false negatives, as classifying a patient who has heart disease as non-disease could have severe negative repercussions on the patients’ health.



```
Accuracy: 0.9230769230769231

Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.92      0.94        26
           1       0.86      0.92      0.89        13

    accuracy                           0.92        39
   macro avg       0.91      0.92      0.92        39
weighted avg       0.93      0.92      0.92        39
```

Figure 1: Accuracy and Classification Reports from the SVM Model.

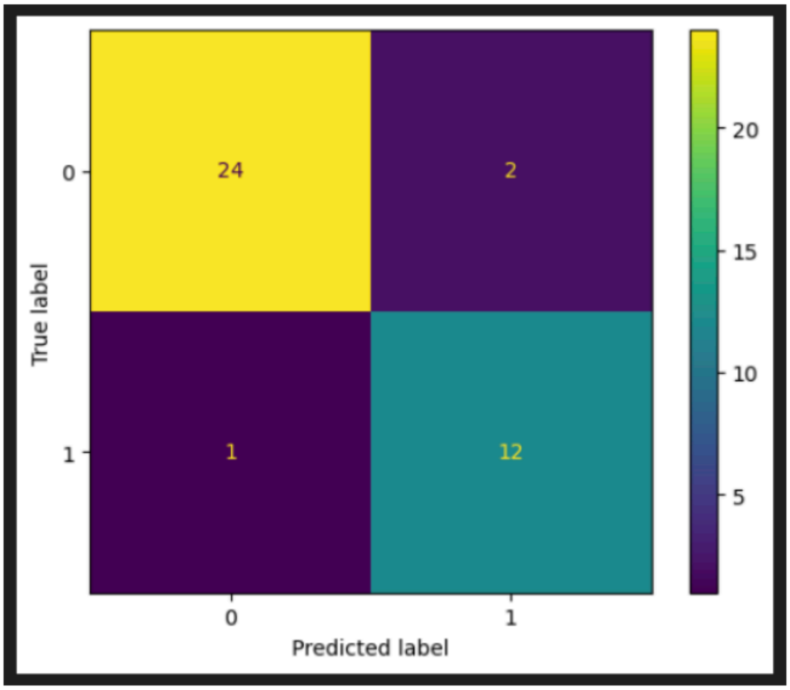


Figure 2: Confusion Matrix Results from the SVM Model.

Considering that accuracy and recall score were greater than 90%, the SVM model did well in drawing the decision hyperplane in predicting patients who did and did not have heart disease based on the given features.

A feature importance graph, from the Linear SVM model used, was also generated, allowing for better visualization of what parameters had the biggest impact on the heart disease being present. This graph can help both with understanding the current dataset, but also expanding into future models when approaching them with different preprocessing methods, or with using larger, more complex datasets. The most important data sets came from the number of vessels identified as calcifying from a fluoroscopy, as well as thallium levels. The least important data sets seemed to come from blood pressure and fasting blood sugar levels.

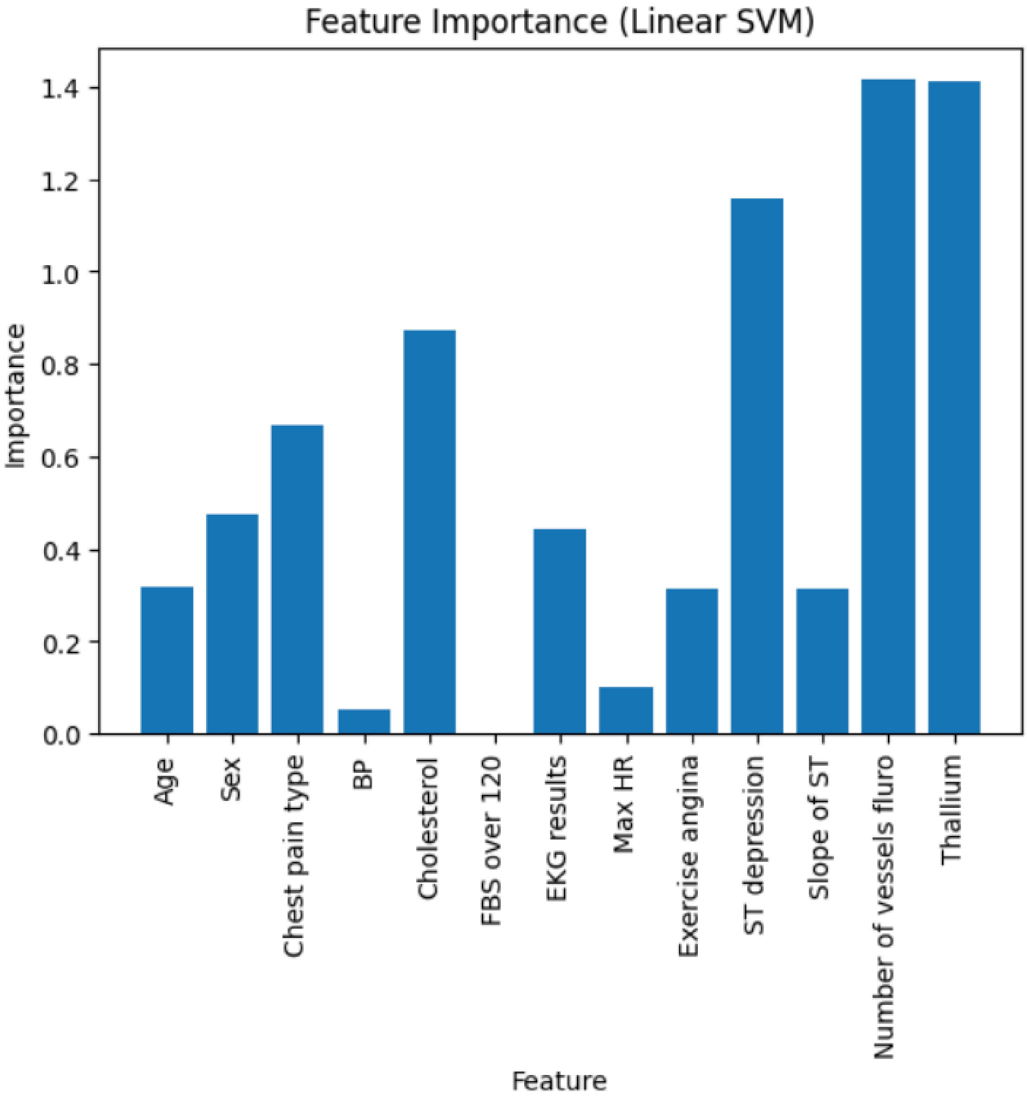
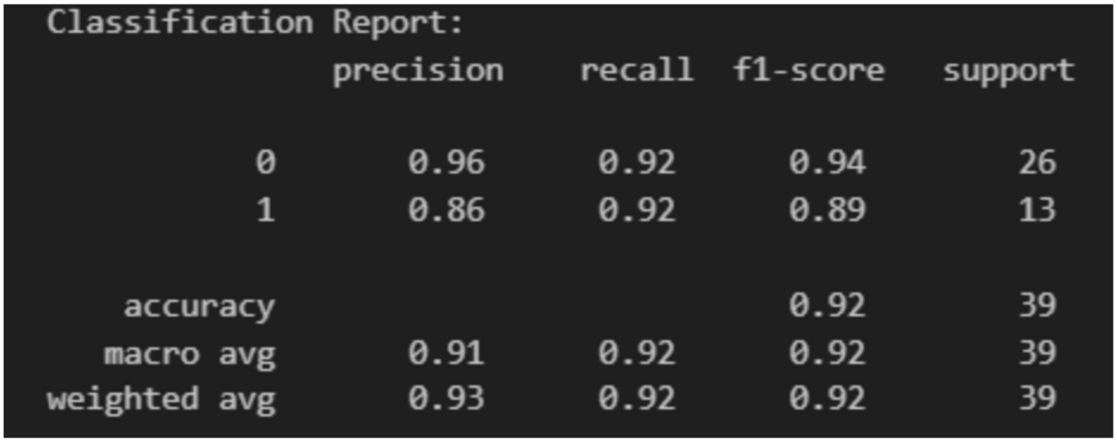


Figure 3: Linear SVM feature importance graph.

The team also examined hyperparameters that may impact our model metrics. Since our model is based on a linear kernel, we only have the regularization hyperparameter C to interact with, as the gamma hyperparameter is involved when the kernel is nonlinear. After running hyperparameter optimization using Bayesian optimization [8] on the accuracy metric from scikit-optimize, the result was that the least a sub-optimal C value of 0.3729944224017752 searching through the range (0.01, 100). Fitting the training data again with the resulting C value, the same accuracy and recall was achieved, reinforcing the fact that the previously used hyperplane was already classifying patients with and without heart disease well as over/underfitting.



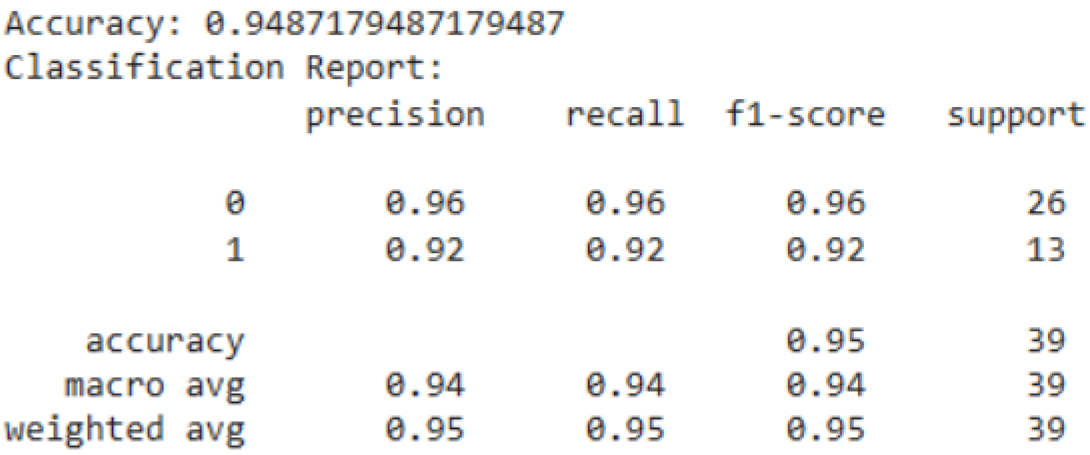
```
Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.92      0.94        26
           1       0.86      0.92      0.89        13

    accuracy                           0.92        39
   macro avg       0.91      0.92      0.92        39
weighted avg       0.93      0.92      0.92        39
```

Figure 4: The modified classification report when evaluating separate hyperparameters.

The team does note: these results may be in part because our data set was rather limited. In the future, we could look at replicating these metrics with a much larger dataset.

RF

The team used 100 estimators and a max tree depth of 3 to construct the random forest model. This relatively large number of estimators and low tree depth helped ensure that there was no overfitting. As mentioned previously, we focused on accuracy and recall metrics because we want to avoid false negatives, while classifying heart disease correctly.



```
Accuracy: 0.9487179487179487
Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.96      0.96        26
           1       0.92      0.92      0.92        13

    accuracy                           0.95        39
   macro avg       0.94      0.94      0.94        39
weighted avg       0.95      0.95      0.95        39
```

Figure 5: Accuracy and Classification Reports from the Random Forest Model.

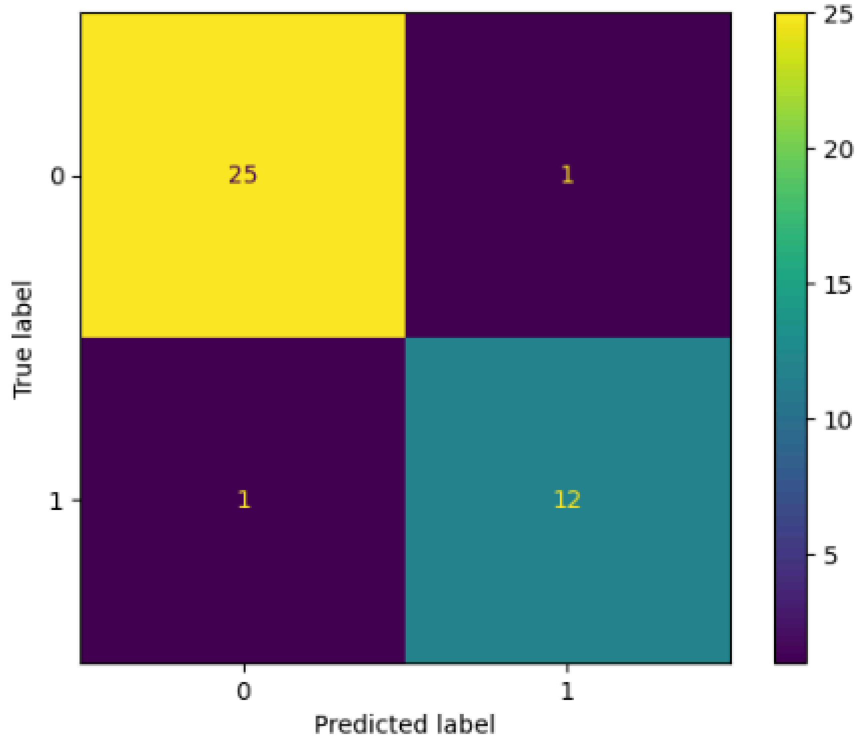


Figure 6: Confusion Matrix Results from the Random Forest Model.

The random forest algorithm performed quite well, with an accuracy of approximately 95% and a macro average recall score of 94%.

The team also plotted the first decision tree in the forest to better understand the model. Although decision trees are easy to interpret, one flaw of random forests is that it is hard to interpret all of the decision trees together. However, the visual still gave us an idea of the structure of each tree and the types of decisions it was making to classify each data point.

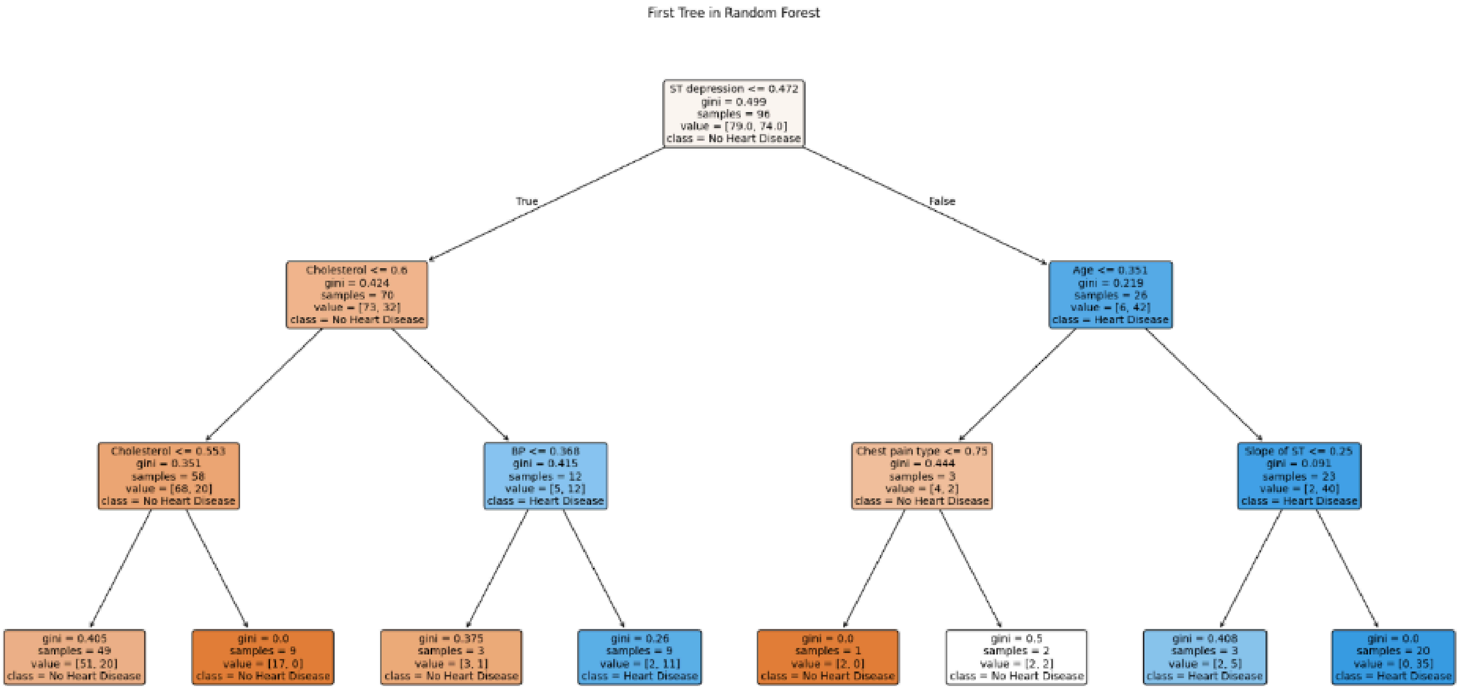
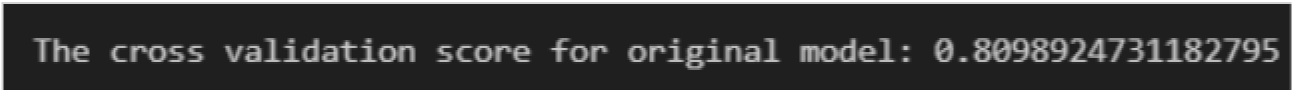


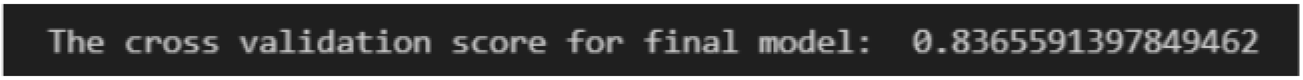
Figure 7: First decision tree in random forest.

We also tuned the hyperparameters m, the number of randomly chosen attributes, and B, the number of decision trees in the model. By using cross validation GridSearch, we were able to get a m of 1 among all of the usual values (1, square root of number of features, 10). We also got a B value of 10 by adding trees until the training error began to plateau locally for a significant tolerance count.

While we got a lower accuracy (1 extra case of false positives), we were able to increase the mean 5-fold cross validation score. For comparison, the two are shown below:



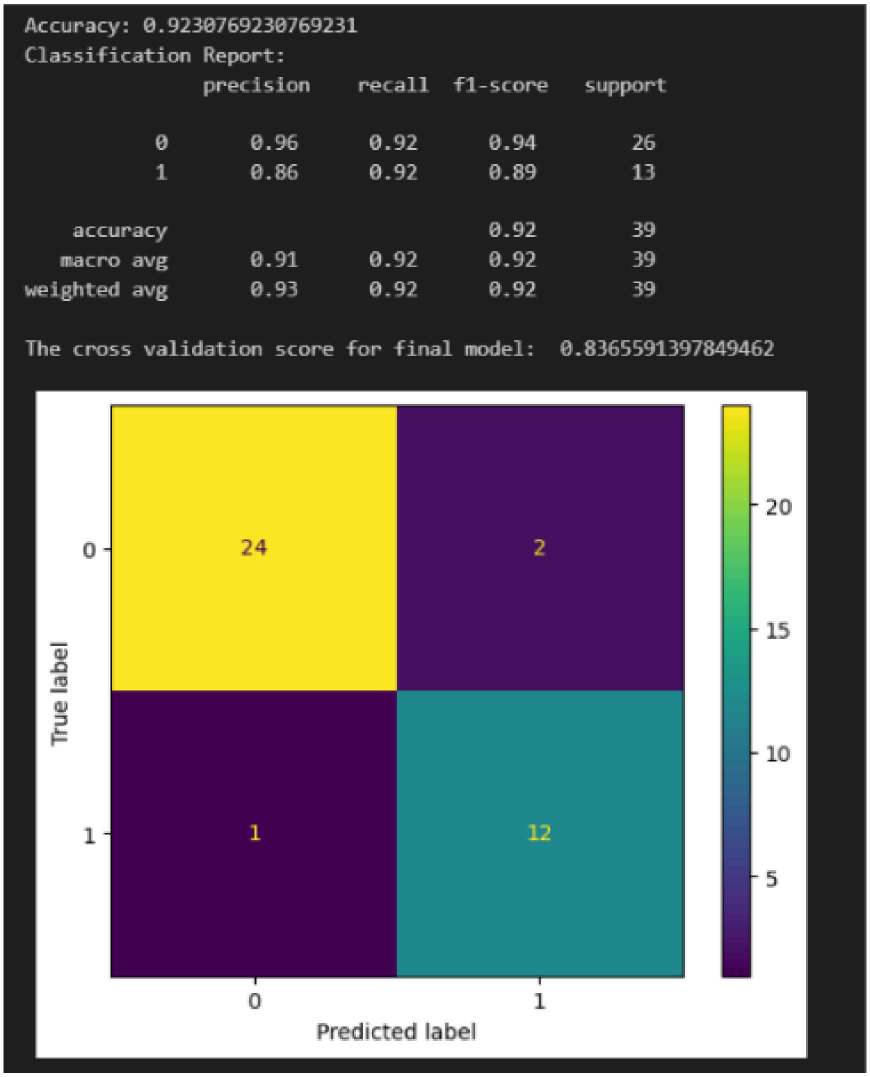
```
The cross validation score for original model: 0.8098924731182795
```



```
The cross validation score for final model:  0.8365591397849462
```

This is an indication that the default hyperparameters may have overfitted the model.

Our post tuning results are shown below:



```
Accuracy: 0.9230769230769231
Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.92      0.94        26
           1       0.86      0.92      0.89        13

    accuracy                           0.92        39
   macro avg       0.91      0.92      0.92        39
weighted avg       0.93      0.92      0.92        39

The cross validation score for final model:  0.8365591397849462
```

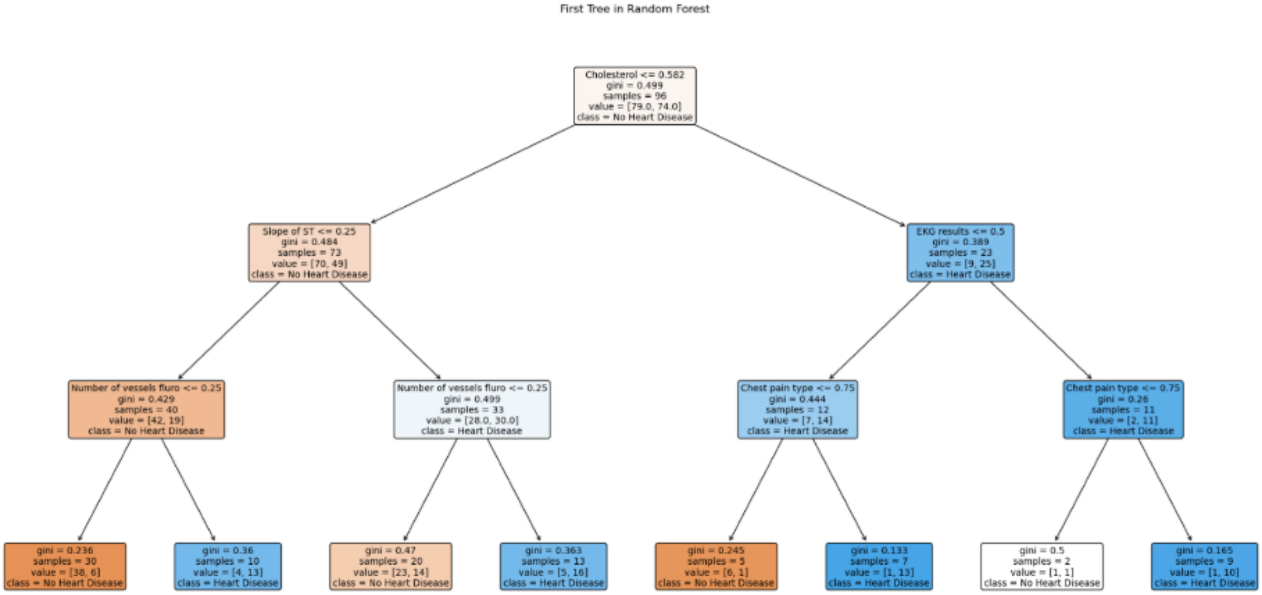


Figure 8: Confusion Matrix and First decision tree in Random Forest after hyperparameter tuning

Overall, the Random Forest model showed similar metric performance to SVM both pre and post tuning.

MLP

The MLP model as applied to the prediction of heart disease, performs relatively well compared to other models, with a lot more hyperparameters necessary to tune from cross validation. From the 14 input parameters, there are enough non-linearities to justify the use of an MLP. Many different architectures were tested with varying success, but the overall shape that tended to draw the most success was descending the amount of perceptron in each subsequent hidden layer, using either ReLU or PreLU activation functions in the hidden layers, and having a final activation function of a sigmoid for classification. The number of perceptron that tended to draw the most success was a 16,10,6,1 structure with the first two layers using ReLU and the penultimate layer using PreLU. As seen in the confusion matrix below in Figure 8, when further adjusting the threshold for classification to 0.48, only 2 values were misclassified, with 28 correctly identified as true negatives and 19 identified as true positives. The overall accuracy of the model tended to hover around 0.88 - 0.92, but with a high level of uncertainty due to model inconsistency in performance. The only area for some level of misalignment was the false positives, but as the team previously justified, it is much safer to have an additional number of false positives, and to reduce as much as possible the number of false negatives while still maintaining a relatively accurate classifier.

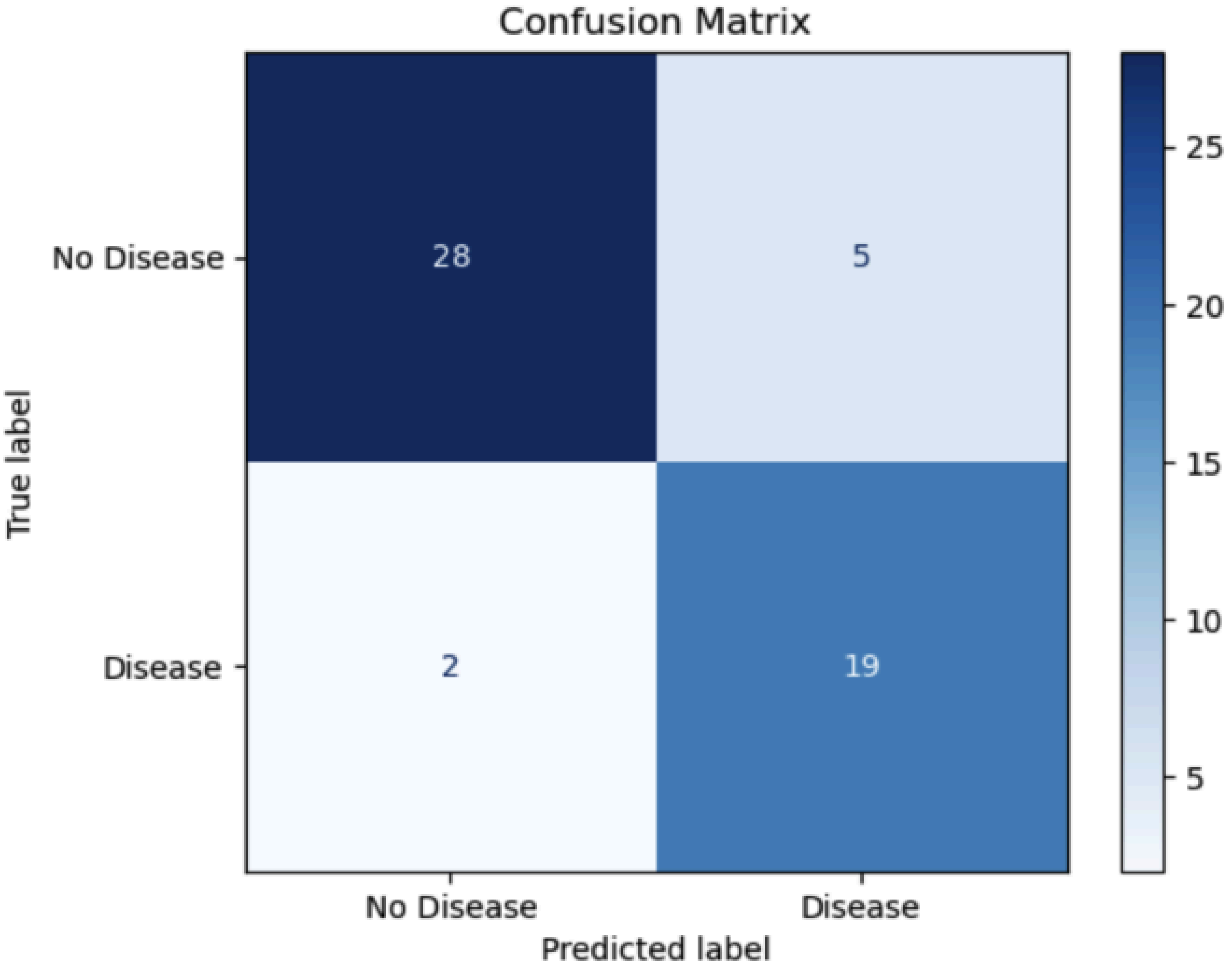


Figure 8: Confusion Matrix for MLP

The largest issue with the MLP is due to the number of hyperparameters there are to tune, and from this comes a lot of variation between different test runs. For example, when trying to decipher the best architecture, some models would perform with an accuracy of 0.944 in one run, and 0.815. This caused high levels of uncertainty when assessing which architectures were the most appropriate for the test cases being evaluated. With more time, a more rigorous testing apparatus could be devised, allowing for a systemic way to optimize all hyperparameters so that the model being used would always be the most optimal for the given type of data.

Another issue with the MLP is the case of overfitting. Due to the number of parameters, it can be easy for the model to overfit to the data. Even with a level of cross validation, the dataset being analyzed is still relatively limited, and this could lead to an ungeneralizable model. Below in Figure 9, the learning and accuracy curves are given, to show the speed at which the model converges with the given architecture and hyperparameters.

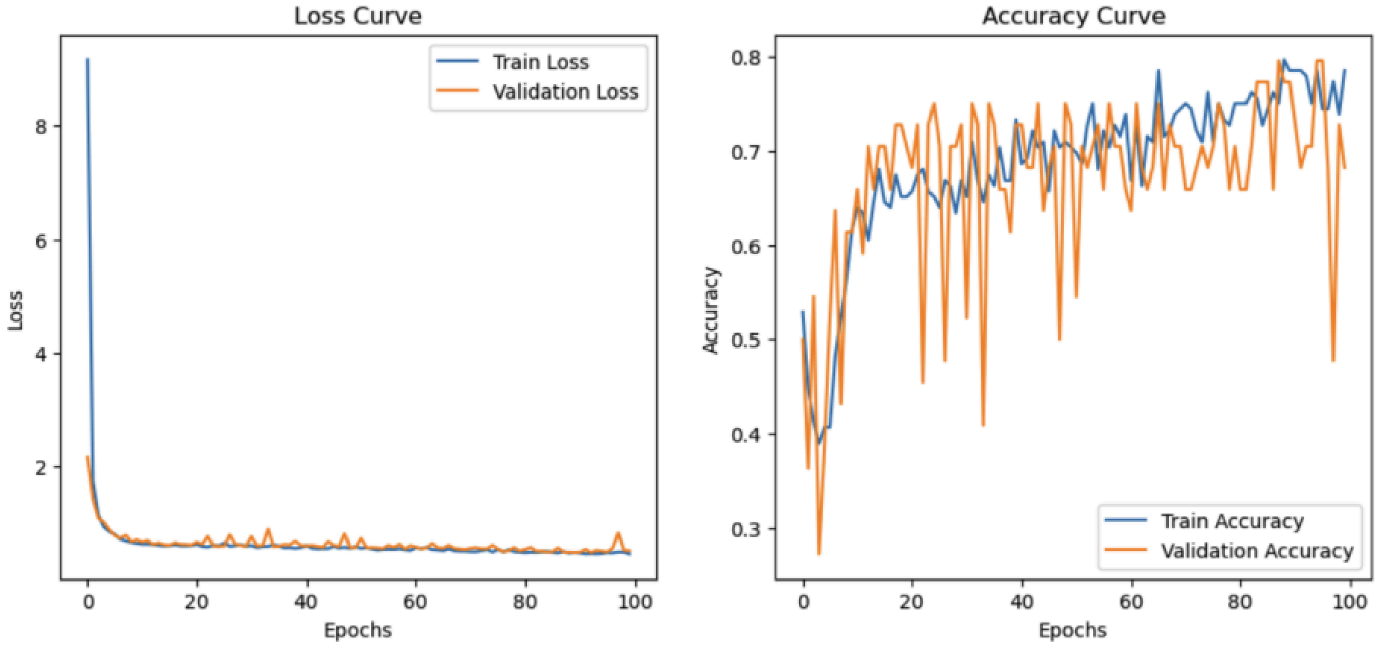
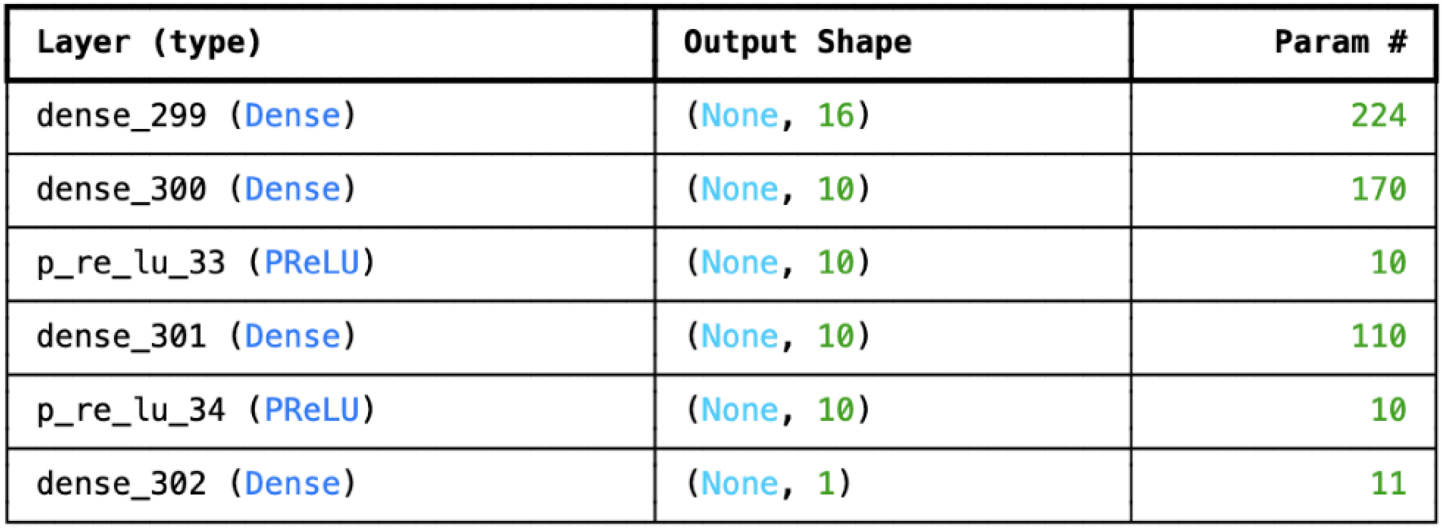


Figure 9: Loss and Accuracy Curves for a Sample MLP Model.

Overall, the use of an MLP model, while maybe more computationally complex than necessary for the given dataset, should likely yield more accurate results overall. Further analysis should be done with a more complex dataset in this area to optimize the MLP set-up and allow for a generalizable model which could be used for the team’s end goal of applying the model to any datapoint a patient may have. In Figure 10, a few more visualizers to describe the model are used for understanding.



Layer (type)	Output Shape	Param #
dense_299 (Dense)	(None, 16)	224
dense_300 (Dense)	(None, 10)	170
p_re_lu_33 (PReLU)	(None, 10)	10
dense_301 (Dense)	(None, 10)	110
p_re_lu_34 (PReLU)	(None, 10)	10
dense_302 (Dense)	(None, 1)	11

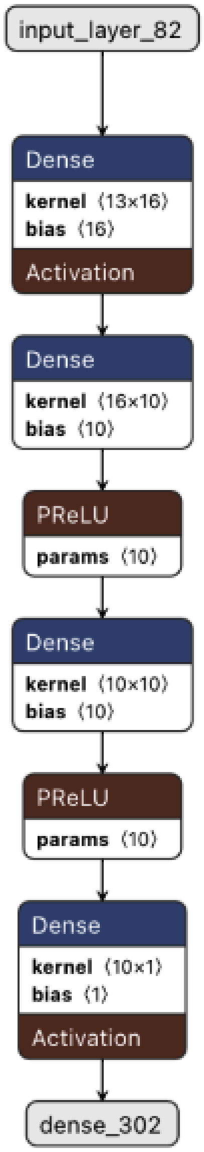


Figure 10: Model Architecture Visualizers.

Comparison

From a preliminary analysis of the results, it was determined that the most consistently weak model was the MLP model. This was likely caused not only by the lack of optimization of the hyper-parameters, but also because of the over-complexity problem. MLPs have a huge number of hyper-parameters, and with a relatively limited dataset, such as the one being worked on here, it can cause overfitting and an overall lack of ability to generalize the model in general. While a preliminary attempt was made to make-up a degree of this difference, overall, the MLP was not satisfactory within expectations.

Overall, the RF and SVM models performed equally. We achieved accuracy and recall of 0.92 and 0.92 respectively for both RF and SVM.

One unique benefit of SVM is its interpretability. It is easier to interpret the SVM results because we can see which features contributed the most to the classification. In contrast, it is hard to interpret how all the decision trees are working together in random forest. Due to this advantage, we would select SVM as the best model.

Next Steps

While our models above do show positive results, there is always plenty of room for improvement. Based on the dataset we are using, our model may be applicable to the region around where it was trained, but may not generalize well to a wider population. In addition, our data would likely benefit from better outlier detection. IQR is a valid strategy, but other ideas like the local outlier factor and DBSCAN may improve the performance of our models. We could also introduce new relationships in the data that may not be as directly stated, such as the relationship between Cholesterol and Age. Taking advantage of these relationships may allow us to draw new conclusions from our models. Outside of our dataset, there is further work that could be done with our models outside of tweaking. We could implement a 2/3 system in our analysis that only predicts heart disease if 2/3 of the models agree. However, this may go contrary to our goal of avoiding false negatives. Further research is needed here. Patient medical history could also be analyzed, as it may have unique traits specific to that dataset that can act as stronger indicators. This, however, could lead to biases forming as well. Finding our next steps is a careful balance between more data, new analysis, and careful consideration. We have found positive work in our results so far, now we want to enhance them.

References

[1] U. Nagavelli, D. Samanta, and P. Chakraborty, “Machine learning technology-based heart disease detection models,” Journal of healthcare engineering, https://pubmed.ncbi.nlm.nih.gov/35265303/

(accessed Sep. 27, 2024).

[2] H. Jindal, S. Agrawal, R. Khera, R. Jain, and P. Nagrath, “Heart disease prediction using machine learning algorithms,” IOP Conference Series: Materials Science and Engineering, https://iopscience.iop.org/article/10.1088/1757-899X/1022/1/012072 (accessed Sep. 27, 2024).

[3] H. A. Al-Alshaikh et al., “Comprehensive evaluation and performance analysis of machine learning in heart disease prediction,” Nature News, https://www.nature.com/articles/s41598-024-58489-7 (accessed Sep. 27, 2024).

[4] M. M. Ahsan and Z. Siddique, “Machine learning-based heart disease diagnosis: A systematic literature review,” Artificial intelligence in medicine, https://pubmed.ncbi.nlm.nih.gov/35534143/ (accessed Sep. 27, 2024).

[5] O. Sami, Y. Elsheikh, and F. Almasalha, “The Role of Data Pre-processing Techniques in Improving Machine Learning Accuracy for Predicting Coronary Heart Disease,” Semantic Scholar, https://www.researchgate.net/publication/353081808_The_Role_of_Data_Pre-processing_Techniques_in_Improving_Machine_Learning_Accuracy_for_Predicting_Coronary_Heart_Disease (accessed Sep. 28, 2024).

[6] C. Boukhatem, H. Y. Youssef, and A. B. Nassif, “Heart Disease Prediction Using Machine Learning,” IEEE Xplore, https://ieeexplore.ieee.org/document/9734880 (accessed Sep. 28, 2024).

[7] A. Patle and D. S. Chouhan, “SVM kernel functions for classification,” 2013 International Conference on Advances in Technology and Engineering (ICATE), Mumbai, India, 2013, pp. 1-9, doi: 10.1109/ICAdTE.2013.6524743.

[8] W. M. Czarnecki, S. Podlewska, and A. J. Bojarski, “Robust optimization of SVM hyperparameters in the classification of bioactive compounds,” Journal of Cheminformatics, vol. 7, no. 1, Aug. 2015, doi: https://doi.org/10.1186/s13321-015-0088-0.

[9] M. W. Gardner and S. R. Dorling, “Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences,” Atmospheric Environment, vol. 32, no. 14–15, pp. 2627–2636, Aug. 1998, doi: https://doi.org/10.1016/s1352-2310(97)00447-0.

Gantt Chart

Contribution Table

Name	Proposal Contributions
David	GitHub Pages/Notes
Anthony	Report Writing/RF Coding
Jacques	Report Writing/Synthesis/MLP Coding
Devan	Report Writing
Andy	Report/Model Coding

cs-4641-heart-disease-repository is maintained by athe27.
This page was generated by GitHub Pages.