Heart Disease Data Analysis

Bryson Gullett Thomas Neuefeind Andy Walsh

I. Objective

The objective of this project is to analyze heart-related patient data in order to create machine learning models that predict whether or not a given person has any form of heart disease, predict the probability of the person having heart disease, and find the most important factors in heart disease prediction.

II. Data

A. Raw Data

The first raw data set we came across when searching for heart disease data was from the UCI ML repository [5]. There was plentiful data, but the format was messy to the point of being unusable given the time it would have taken to clean. Instead, we focused our efforts on finding data sets on kaggle.com, of which there were four [1]–[4]. Upon some basic exploration and cleaning however, it became clear that one of these data sets did not contain real patient data (since some values in the data set were not realistic), while the other three data sets all pulled data from the same location – the UCI ML repository we first came across.

```
first = pd.read_csv("./heart.csv")
second = pd.read_csv("./heart1.csv")
third = pd.read_csv("./heart2.csv")
print("shapes:")
print("first df ==", first.shape)
print("second df ==", second.shape)
print("third df == ", third.shape)
```

Fig. 1. Reading in the three remaining data sets, printing their shape

```
shapes:
first df == (918, 12)
second df == (303, 14)
third df ==  (1025, 14)
second and third merged == (1033, 14)
```

Fig. 2. Combining the second and third data set yielded only 8 unique patient records

While it became clear that we would not be able to combine these data sets for meaningful results, we did take the time to research these more thoroughly. In doing so, we discovered that the data set with 918 patient records had checked for duplicates in the data already. Hence we chose to do our analysis on this data set going forward [1]. Importantly, this data set had a nearly balanced positive vs. negative heart disease label column which we could use for predictions.

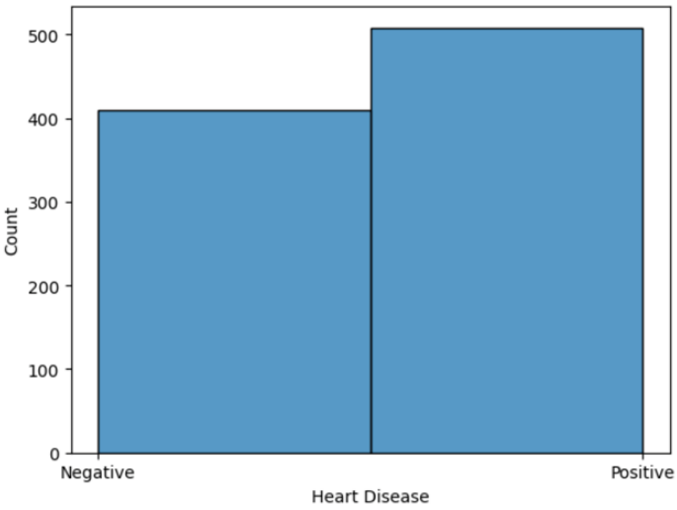


Fig. 3. The final data set we chose was nearly balanced – 55.38% positive labels, and 44.62% negative data

The remaining eleven features in the dataset were as follows:

- Age: age of the patient in years
- Sex: sex of the patient (M/F)
- ChestPainType: type of chest pain caused by exercise, four values possible:
  - TA: Typical Angina
  - ATA: Atypical Angina
  - NAP: Non-anginal pain
  - ASY: Asymptomatic
- RestingBP: resting blood pressure (mm Hg)
- Cholesterol: level of cholesterol (mm/dl)
- FastingBS: fasting blood sugar (binary value – 1 if greater than 120 mg/dl, 0 otherwise)
- RestingECG: results of resting electrocardiogram, three values possible
  - Normal: results were normal
  - ST: ST-T wave abnormanility
  - LVH: left ventricular hypertrophy
- MaxHR: maximum heartrate achieved – numeric value between 60 and 202
- ExerciseAngina: whether or not the patient has exercise enduced angina – 0 or 1
- Oldpeak: taken from the ST ECG graph – is a numeric value indicating the strength of the ST depression in the patient
- ST_Slope: the slope of the ST segment of the ST ECG graph, three values possible:

- Up
- Flat
- Down

B. Cleaning Data

In order to clean the data for machine learning model training, it was first split into features and a label. Because the project is focused on classifying heart disease, the data column HeartDisease was used as the label. Luckily, the label column already had its data in a binary format. All other eleven data columns in the data set (Age, Sex, ChestPainType, RestingBP, Cholesterol, FastingBS, RestingECG, MaxHR, ExerciseAngina, Oldpeak, and ST_Slope) were used as features for machine learning.

After splitting the data into features and labels, many feature columns had to be converted from string values into numeric values. Therefore, the Sex column was converted from 'F' and 'M' characters to binary values. Furthermore, the ChestPainType, RestingECG, ExerciseAngina, and ST_Slope data columns were each encoded into multiple "dummy variable" binary columns via the one-hot encoding technique.

In the final phase of data cleaning, all of the rows of data were randomly divided into a training data set and a test data set. The training data was used to train the machine learning models while the test data was used to test model performance in an unbiased manner.

III. Models and Algorithms

A. Machine Learning Classification Models

A variety of different machine learning models and algorithms were used for heart disease classification in order to compare performance across different algorithms and to get a better idea of what features were the most significant in heart disease prediction. The following machine learning algorithms were used to classify heart disease in this project:

- k-Nearest Neighbors (KNN)
- Support Vector Machine (SVM)
- Gaussian Naive Bayes
- Random Forest
- AdaBoost
- Multi-Layer Perceptron (MLP)
- Logistic Regression

We used scikit-learn's Python packages in order to implement these machine learning algorithms.

B. Feature Importance Ranking

In order to rank which features were the most important in predicting heart disease, we chose to use permutation feature importance for each machine learning model. The permutation feature importance algorithm is given below (derived from scikit-learn [6]):

Fit machine learning model m using tabular data set D.
Find reference classification accuracy a of m on D.
for each feature f in D **do**
  for each repetition k in 1,2,3...,K **do**
    Randomly shuffle values of f to obtain a corrupted data set D_{kf}.
    Find classification accuracy a_{kf} of m on D_{kf}.
  end for
  Calculate the importance i_{fm} of f:

$$i_{fm} = a - \frac{1}{K}\sum_{k=1}^{K} a_{kf}$$

end for

Utilizing permutation feature importance provided us knowledge of how important each feature is for each model. However, we also wanted to obtain a ranking of which features were the overall most significant in predicting heart disease. Consequently, we also took a weighted average of each feature's importance for each model, where we weighted each importance value based on the classification accuracy for the corresponding model on the test data. This weighted average formula used for each feature, where M is the number of machine learning models, is shown below:

$$i_f = \frac{1}{\sum_{m=1}^{M} a_m}\sum_{m=1}^{M} i_{fm}a_m$$

C. ML Probabilistic Interpretations

Hypotheses were formed on heart disease through our probabilistic model, logistic regression. Regression gives us the likelihood of a data instance falling into a class. We analyzed the average of each feature, across all data entries, at certain levels of confidence up to 100 to see if there were notable trends in the data that might lend themselves to forming hypotheses in the vein of indicating heart disease, its severity, and general heart health.

```
entryPercentiles = dict()
for i in range(0,100+1, 5):
    entryPercentiles.update({i: []})
for i in range(len(lgrProbs)):
    entryPercentiles[floor(5 * round((100 * lgrProbs[i][1]) / 5))].append(i)

def plotFeatureTrend(featureName, featureIndex):
    xAxis = []
    yAxis = []
    for i in range(0,101,5):
        if(len(entryPercentiles[i]) == 0):
            continue
        featureSum = 0
        for k in entryPercentiles[i]:
            featureSum += X_train[k][featureIndex]
        xAxis.append(i)
        yAxis.append(featureSum / len(entryPercentiles[i]))
    plt.plot(xAxis, yAxis)
    plt.xlabel("Percentage Chance of Heart Disease")
    plt.ylabel("Average feature value")
    plt.title(f'{featureName} as it trends towards 100% indication of heart disease')
    plt.show()

for i in range(len(feature_cols_clean)):
    plotFeatureTrend(feature_cols_clean[i], i)
```

Fig. 4. Algorithm for Probabilistic Visualizations

The algorithm utilized to accomplish this was hand-written. It groups entries in the data by their malignant classification

probability to the nearest percentile in multiples of 5, then for each feature makes a plot of the average feature value at each level of confidence. To visualize this, we made a plot for each feature that showed the average value of that feature and the percentage chance of the entries at that value being malignant. The chance of malignancy was calculated using our logistic regression model. Many features when evaluated in this manner did not reflect a consistent pattern conducive to extracting greater meaning, but the following were noteworthy.

IV. Results

A. Data Correlation

In order to guide our exploration using the different machine learning models, we first explored how strongly the data correlates with each other. Consequently, one of the first forms of data analysis that we did was creating a correlation matrix and visualizing it with seaborn's heat map functionality. The correlation matrix is depicted below:

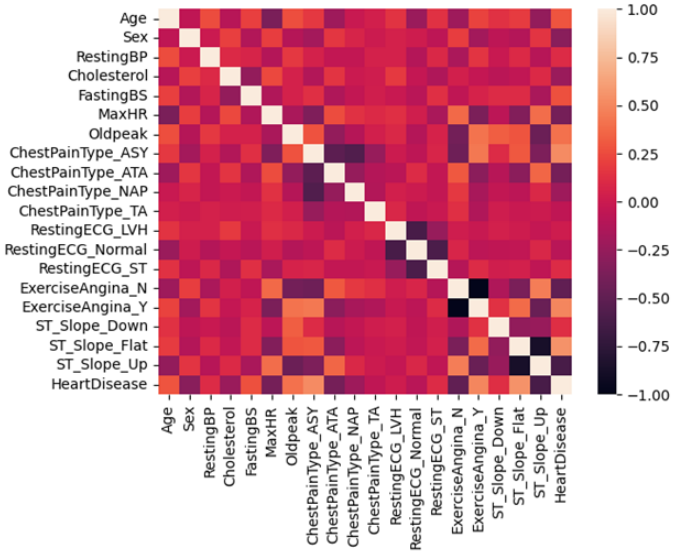


Fig. 5. Correlation matrix: Lighter colors mean features are more positively correlated, while darker colors mean features are more negatively correlated

From this correlation matrix, the features most correlated with the HeartDisease label are:

- ChestPainType
- ST_Slope
- ExerciseAngina
- Oldpeak
- MaxHR

All of these features that highly correlate with heart disease are reasonable and expected. Chest pain (represented by the ChestPainType and ExerciseAngina variables) is consistently associated by the masses with both heart attacks and heart disease. It also makes sense that heart rate (represented by the MaxHR) is related to heart disease because heart diseases significantly change heart beat patterns. Finally, the ST ECG graph (represented by the ST_Slope and Oldpeak variables) has been proven to indicate heart disease as well.

B. ML Classification Results

We used the binary label of HeartDisease in our data set to train and test the different machine learning models. The best test accuracies for the models are displayed below:

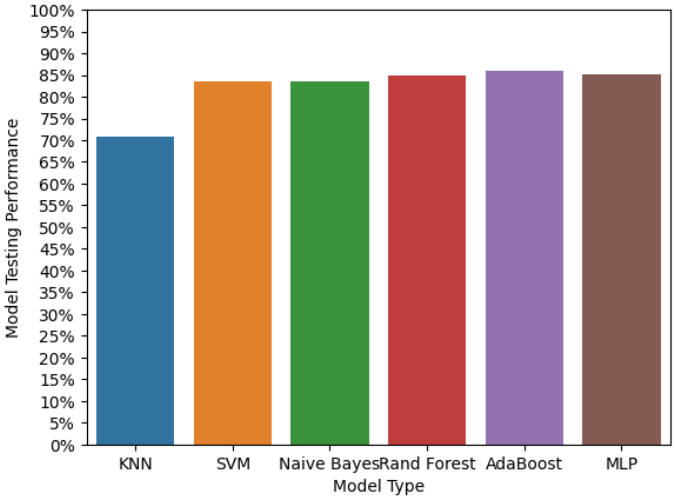


Fig. 6. Test accuracies for the various machine learning models employed. KNN had the worst test accuracy with just over 70%, while AdaBoost had the best test accuracy with 86%.

It can be seen from the bar chart that all the models except k-Nearest Neighbors had roughly 85% test accuracy. The relative under-performance of the k-Nearest Neighbors approach may be due to the data simply not being amenable to distance metrics, though further work would be required to test this idea.

Since all the test accuracies for all the other models were very similar, it is quite likely that further performance increases for the remaining models would need to come from a change in the data. In particular, a larger and more diverse data set would help the models employed here pick up on more trends in the data.

In addition to using test accuracy to evaluate the models' performance we also constructed ROC curves.

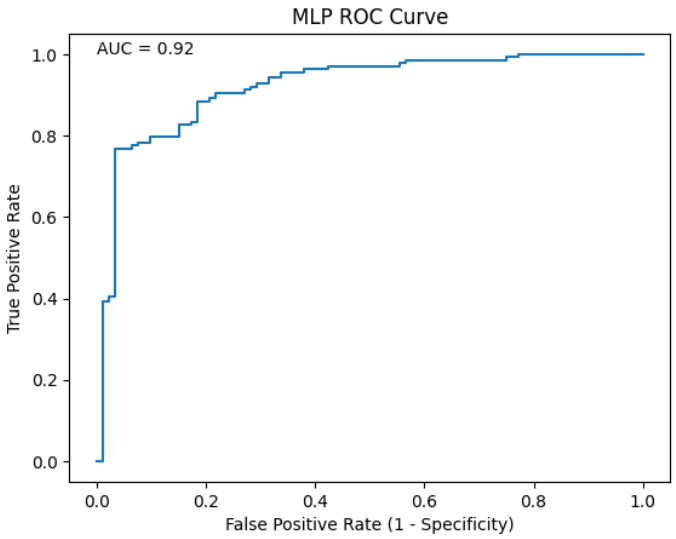


Fig. 7. ROC curve for the multi-layer perceptron model

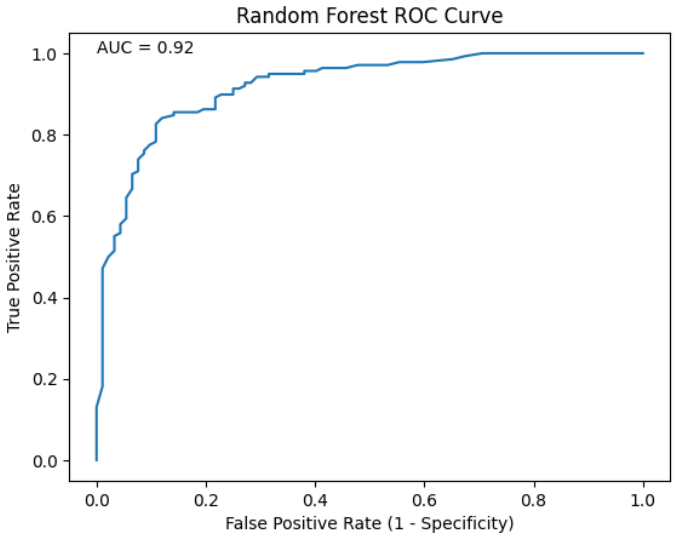
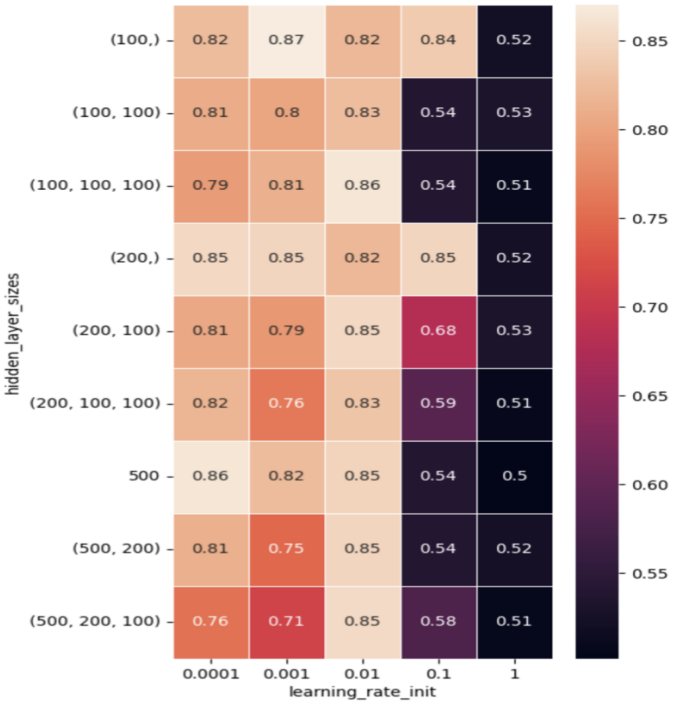


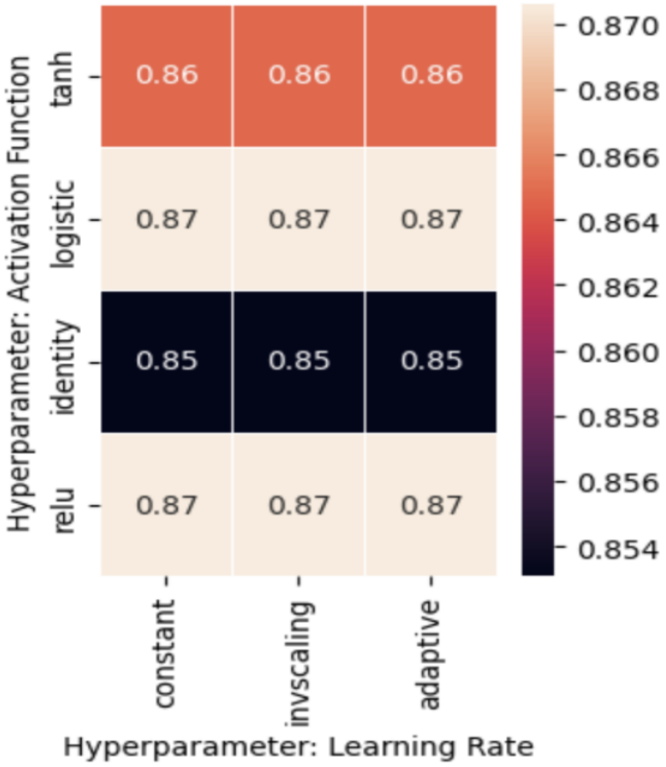
Fig. 8. ROC curve for the random forest model

The AUC (area under the curve) values are another indicator of model performance. More specifically, they describe the ratio of the true positive rate to the false positive rate. By this metric, the random forest and multi-level perceptron were the best performers, while all but the k-nearest neighbors model had an AUC value of 0.90 or above. More work could be done here in order to further evaluate specificity vs. sensitivity, since these are important metrics when dealing with medical data.

C. Hyperparameter Optimization

A thorough hyperparameter search was conducted on neural networks. This was our model of choice, due to the fact that they were our highest performing model on sklearn's default parameters outside of AdaBoost. In order to optimize computational cost, the hyperparameter search was divided into two stages: initial learning rate and neural architecture, then learning rate and activation function. Pictured are the heatmaps of these two independent searches.





Our highest accuracy combination of hyperparameters included a neural architecture of (100,), an initial learning rate of

0.001, a constant learning rate, and the RELU activation function. These parameters are, surprisingly, the default parameters of the MLP classifier. It happened to be the case that, given our search space, the best performing model was sklearn's default. The training accuracy of our model with these parameters was 88.23%, and the testing accuracy was 85.22%. Possible permutations of the aforementioned hyperparameters were evaluated by K-fold cross validation, with 10 folds specifically.

D. ML Probabilistic Model Results

Visualizations derived from the previously described algorithm enabled analysis of feature trends as confidence, an analog for likelihood, of heart disease increased. Many features when evaluated in this manner did not reflect a consistent pattern conducive to extracting greater meaning, but the following were noteworthy.

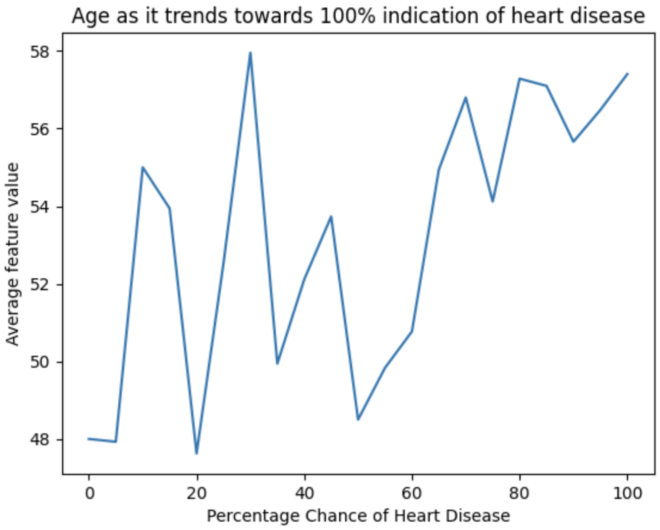


Fig. 9. Those who had greater than 50% chance of heart disease spiked and trended up in average age, which indicates it is a strong factor in heart health

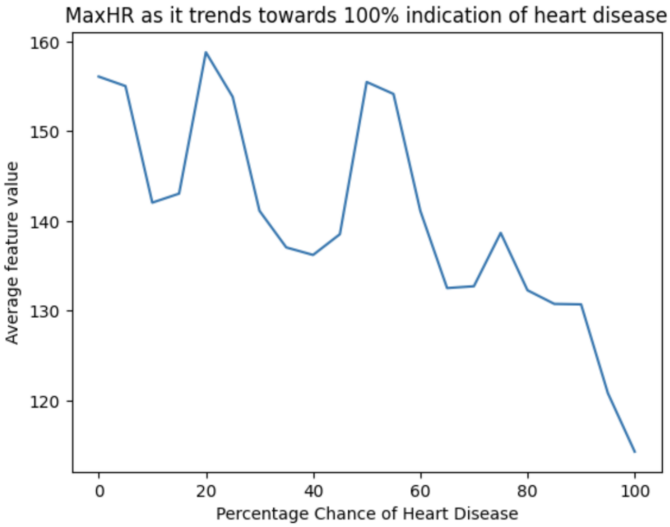


Fig. 10. Lower max heart rate appeared to indicate increased disease severity

Out of all the features, age, max heart rate, fasting blood sugar, atypical chest pain, asymptomatic chest pain, and exercise induced chest pain showed the most discernible trends. Further, the heatmap of our dataset above reflects that these features are positively correlated with one another. Individually and collectively, on average, these values seem to be a good indicator for risk factor in heart disease, and may have some relationship with disease severity and heart health.

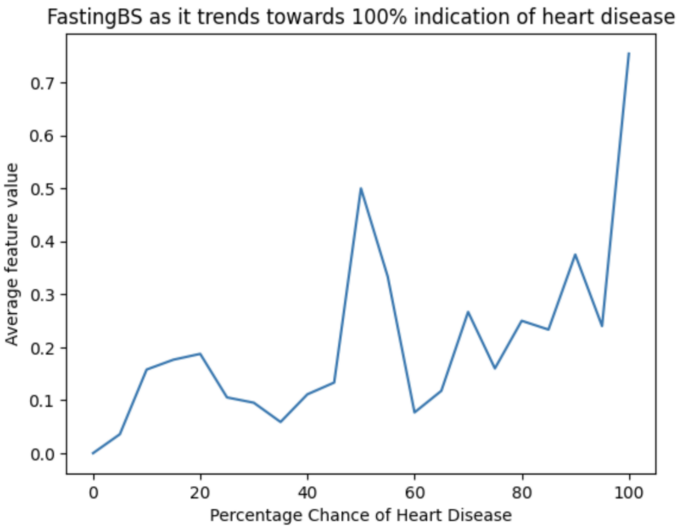


Fig. 11. Blood sugar showed an almost exponential pattern, reflecting that higher blood sugar likely indicates worsening heart health, particularly on the extreme end of the spectrum

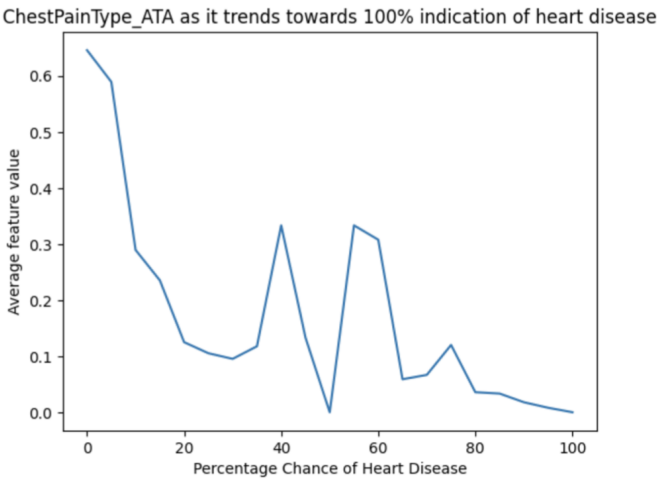


Fig. 12. ATA showed clear trends in predicting heart disease negatively

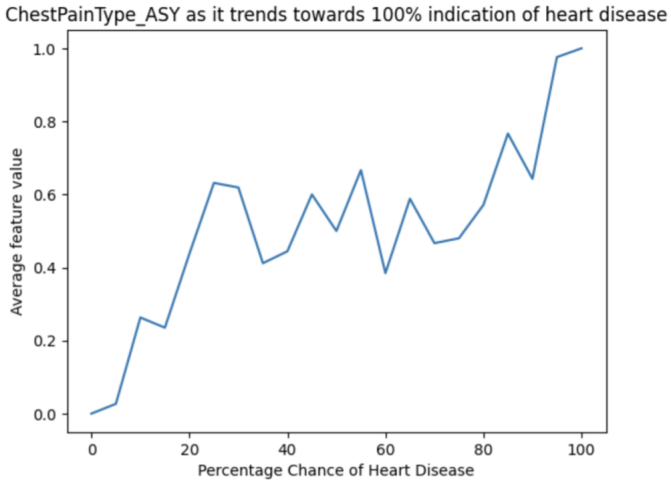


Fig. 13. ASA showed clear trends in predicting heart disease positively

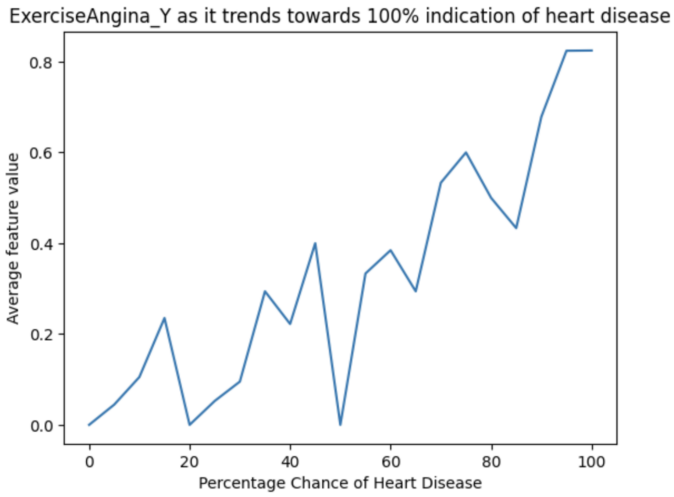


Fig. 14. Chest pain induced by exercise similarly indicated poor heart health

E. Feature Ranking Results

Figure 17 in the Appendix section provides a table of feature importance values for each feature and for each machine learning model. These feature importance values were generated using the feature permutation importance algorithm detailed in Section III-B. In Figure 17, the top three features with the highest feature importance values are highlighted in gold, silver, and bronze, respectively. Figure 17 demonstrates that the ST_Slope feature is the most important feature in predicting heart disease for five of the seven different machine learning models. This is sensible because the correlation matrix visualization (Figure 5) demonstrates that ST_Slope has a strong correlation to the HeartDisease label. Therefore, it is reasonable that ST_Slope's high correlation to HeartDisease leads to ST_Slope's high importance in predicting HeartDisease.

After applying the weighted average algorithm described in Section III-B to the per-model feature importance values, we obtained the following overall feature importance rankings:

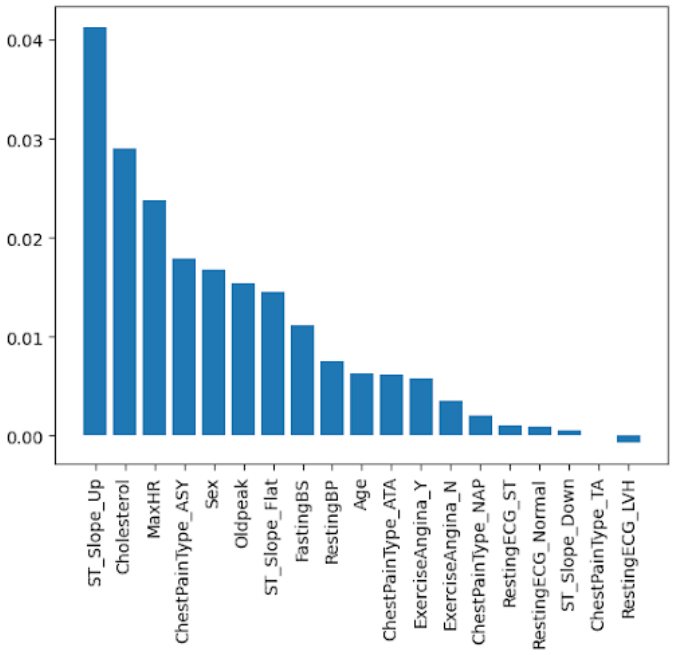


Fig. 15. Overall feature importance rankings

From this chart, we can conclude that the following were the most significant in allowing our machine learning models to classify heart disease:

1) ST_Slope
2) Cholesterol
3) MaxHR
4) ChestPainType
5) Sex

Three of these five most important overall features (ST_Slope, MaxHR, and ChestPainType) are among the most highly correlated features with HeartDisease from the correlation matrix visualization (Figure 5). Once again, it is sensible that the variables most highly correlated with HeartDisease would be the features that are the most important in predicting it.

Interestingly and unexpectedly, ExerciseAngina, which was highly correlated to HeartDisease in the correlation matrix, was not even in the top half of the final overall feature importance rankings. Furthermore, it is also odd that ST_Slope_Up is overwhelmingly the most important feature in predicting heart disease for our models while ST_Slope_Down is the third least important. However, looking back at the data, ST_Slope equalling "Down" appears in very few patients, which reasonably explains why the machine learning models do not depend on the ST_Slope_Down feature for heart disease prediction.

V. Issues

The main issue we ran into in this project was the data from which we could make predictions. Though we searched through many readily available sources of data, they all drew from a single source – the UCI Machine Learning Repository

[5]. We also looked into using the data directly from this repository, but the format was too inconsistent to be able to make usable for the scope of this project. This limitation led to using a single, mostly pre-cleaned data set from kaggle.com [1]. While this made using and analyzing the data easier, it limited the statistical power of our analysis.

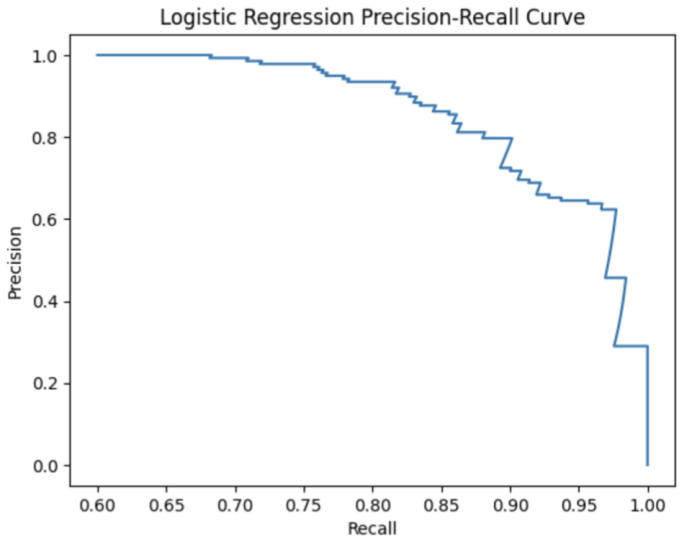


Fig. 16.

Our hypothetical analysis was conducted with probabilities from our logistic regression model, which was only 82% accurate. Therefore, given an 18% error rate in classifications, our confidence values utilized to plot feature trends as certainty of heart disease approaches 100% is not perfectly reflective of these trends. In general, to form these hypotheses more concretely, we need more data with more labels from which to draw conclusions.

VI. Future Work

There are a number of key points that could be addressed with future work. We could work on improving our current models with further hyperparameter optimization, improve the data set, improve the robustness of our results analysis, and apply different types of machine learning algorithms for a more varied approach to learning trends in heart disease prediction.

A. Improve Current Models

Our best performing model using the current data was based on AdaBoost. Since this method can use various machine learning models as its base estimator, a hyperparamter search should be carried out to determine which is best at learning from the current data. This search space however will be computationally intensive, as base estimators have hyperparameter spaces of their own. Apart from further hyperparameter optimization, we could also employ dimensionality reduction or feature pruning.

B. Improving the Data Set

Perhaps the most meaningful way to improve and better generalize model performances would be to get a larger and more diverse data set.

Currently, the data set employs only eleven features to predict on a binary label. Given more features, we would likely be able to get better test accuracy results. Given a different label type, e.g. type of heart disease would allow us to see if these models could be used for multi-class classification as well. Further, given real-valued labels (e.g. between 0 and 1) would make the problem more amenable to regression-based approaches, allowing us to better predict the severity of heart disease. All these factors would likely improve the model performance.

Lastly, in order to better generalize from these predictions, a data set that draws from more diverse sources is required. As mentioned above, this data came from hospitals in Europe and North America, which may have different important features than Asian, South American, or African countries.

C. Improving Analysis

There are still several techniques that could be employed to improve our analysis of the data. Importantly, since we are dealing with medical data, a more thorough evaluation of specificity vs. sensitivity should be conducted, building on the ROC curves constructed already. This could be used to help set a threshold of likelihood that a patient has heart disease, which in turn would make it much more viable to employ these models in the real world.

Additionally, mapping a line of best fit to individual features along with an R-squared value of the strength of the fit, would supplement our understanding of how strongly these features correlate with our label.

D. Using Unsupervised Learning

An alternative approach to learning from our data is to use unsupervised learning, such as a k-means clustering approach. Using such an approach may help us identify further trends in the data, such as previously unrecognized relationships between features.

VII. Project Organization

A. Team Member Responsibilities

- **Bryson Gullett:**
  Bryson was in charge of implementing the machine learning algorithms. He researched which approaches would be best for our data. He used the data cleaned and combined by Thomas and used it to train various machine learning models that classified and provided a probability of a given individual having heart disease. Lastly, Bryson provided model outputs and accuracy data to Andy for analysis.
- **Thomas Neuefeind:**
  Thomas was in charge of cleaning and evaluating the data sets that we used for this project. Therefore, Thomas was responsible for removing erroneous data, and identifying

common features across data sets. He then provided the final cleaned data set to Bryson and Andy. Thomas also performed the ROC curve and precision/accuracy curve analyses.

- **Andy Walsh:**
  Andy was in charge of analyzing and interpreting data as well as the outputs of the machine learning algorithms. Informed by Bryson's findings on model performance, Andy conducted the hyperparameter search on neural networks. Andy also made many of the visual aids, in particular with regards to the regression-based approaches. Andy was also instrumental in guiding the direction of the project in general.

B. Project Timeline

- **October 13th:** Finished basic data analysis and interpretation on each of the originally selected data sets.
- **October 27th:** Finalized data set selection. Applied the first machine learning algorithm to the finalized data set.
- **November 10th:** Began preliminary data analysis. Started search for best type of machine learning algorithm.
- **November 23rd:** Finished applying all additional machine learning techniques. Finished analysis of results of the machine learning model outputs.
- **December 10th:** The project is finished. We refined the visual aids from various analyses during this time. We also ensured that all machine learning approaches were properly applied to the data, and laid out tasks for future work in this area.

REFERENCES

[1] https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction
[2] https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset
[3] https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset
[4] https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset
[5] https://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/
[6] https://scikit-learn.org/stable/modules/permutation_importance.html
[7] https://scikit-learn.org/stable/index.html

APPENDIX

Ranking of each feature by model

Attribute Name	KNN	SVM	Naive Bayes	Rand Forest	AdaBoost	MLP	Log Reg.
Age	0.012461873638344185	-0.0011764705882353322	-0.0007843137254901889	0.023246187363834415	0.006906318082788643	0.004553376906318123	-0.0006535947712418055
Sex	0.0	0.022440087145969456	0.011263616557734207	0.025816993464052276	0.02045751633986924	0.01644880174291943	0.01840958605664491
RestingBP	0.01647058823529408	-0.0015686274509804266	0.0010457516339869243	0.02030501089324618	0.011481481481481443	0.008148148148148187	-0.0020697167755990976
Cholesterol	0.07777777777777774	0.006078431372548978	0.013289760348583883	0.037211328976034845	0.018758169934640478	0.047625272331154725	0.00960784313725493
FastingBS	0.0	0.018126361655773384	0.012592592592592593	0.01206971677559911	0.016274509803921533	0.0023093681917211706	0.015664488017429215
MaxHR	0.10938997821350759	-0.0004793028322440462	0.0034640522875816915	0.0313725490196078 3	0.015795206971677515	0.01954248366013075	0.0007407407407407706
Oldpeak	0.0007189542483659773	0.006274509803921 5215	0.01607843137254903	0.031612200435729834	0.022875816993464016	0.017952069716775633	0.009324618736383461
ChestPainType_ASY	-0.0005664488017429559	0.00999999999999996	0.009302832244008716	0.039760348583877995	0.03570806100217861	0.009629629629629662	0.018061002178649265
ChestPainType_ATA	-0.00028322440087147794	0.0008278867102396048	0.016688453159041397	0.005642701525054449 5	0.002483660130718921	0.008845315904139466	0.007908496732026166
ChestPainType_NAP	0.0	-0.002461873638344261	0.0031590413943355044	0.006318082788671003	0.0	0.005795206971677602	0.0008278867102396892
ChestPainType_TA	0.0	-0.0002396514161220 1536	0.0013071895424836533	0.0	-0.0007843137254902311	0.0001960784313725749 8	-0.0001089324618736342 6
RestingECG_LVH	0.0	-0.003050108932461899	-0.0026579520697167713	0.00511982570806099	-0.0032244008714597205	-0.00098039215686270 6	-0.0001307189542483322 5
RestingECG_Normal	0.0	-0.0003703703703703942	-0.0001960784313725527 5	0.003660130718954237 4	0.002941176470588198 4	0.000675381263616592 4	-0.0005010893246187176
RestingECG_ST	0.0	-2.178649237476904e-05	0.002113289760348582 2	0.005272331154684082	0.0	0.001285403050108970 9	-0.0013943355119825295
ExerciseAngina_N	0.0	0.0042483660130718534	0.005424836601307183 5	0.008366013071895 41	0.0	0.004901960784313761	0.0016557734204793294
ExerciseAngina_Y	0.0	0.0042483660130718534	0.005424836601307183 5	0.009019607843137243	0.007211328976034832	0.01017429193899785	0.0038344226579520 9
ST_Slope_Down	0.0	0.0	0.001067538126361651 2	0.000130718954248361 12	0.003137254901960748 6	-0.00030501089324615 82	-0.0002396514161219931 3
ST_Slope_Flat	-0.0005664488017429559	0.024945533769063135	0.016688453159041 41	0.007233115468409 57	0.0	0.021111111111111146	0.030893246187363 87
ST_Slope_Up	-0.0004357298474945814 5	0.025599128540304966	0.022309368191721 13	0.063856209150326 79	0.120740740740740 73	0.016187363834422 69	0.031895424836601 33

Fig. 17. Feature importance rankings for each model