Cardiovascular Disease Prediction Project Report

Kento Hopkins

Contents

1 Introduction and Motivation

Cardiovascular disease is one of the most significant health issues we face in modern society. According to the CDC (https://www.cdc.gov/heart-disease/data-research/facts-stats/index.html), heart disease is the leading cause of death across various groups of people, regardless of gender, race, or ethnicity. The same source states that one person dies from cardiovascular disease every 33 seconds. Although we face many health challenges, the high cost of healthcare and insurance makes access unaffordable for many people. This prediction project aims to analyze the health factors contributing to cardiovascular disease and provide suggestions for improvements to address health disparities caused by financial status.

As suggested by the study published in Circulation (https://www.ahajournals.org/doi/10.1161/CIRCULATIONAHA.124.0 predicting cardiovascular disease is challenging, even though we have various health measures, some of which have been proven to impact the risk of heart-related diseases, and we have observed correlations. Additionally, according to Harvard Health (https://www.health.harvard.edu/heart-health/a-closer-look-at-heart-disease-risk), heart disease prediction has never been an exact science. The same source mentions that, even with factors like smoking status, diabetes, and cholesterol levels, predicting heart disease remains difficult. Even though new approaches, such as the coronary artery calcium scan, have developped, many people cannot afford cutting-edge healthcare due to its high cost.

In this project, I used two main approaches, logistic regression and neural networks. Logistic regression returns a probability value between 0 and 1 based on the input of independent variables. This is beneficial

for our prediction task, as we are trying to classify whether a person is diagnosed with cardiovascular disease which is a binary outcome. My choice to use a neural network was driven by both my curiosity to gain hands-on experience and its ability to capture more complex data correlations. Given the complexity of predicting cardiovascular disease which is a major challenge even in the healthcare industry, I believe neural networks offer a powerful solution for this type of problem.

2 Data

For this project, I was looking for a dataset with more than 10,000 samples. I found the primary dataset on Kaggle(www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset.).

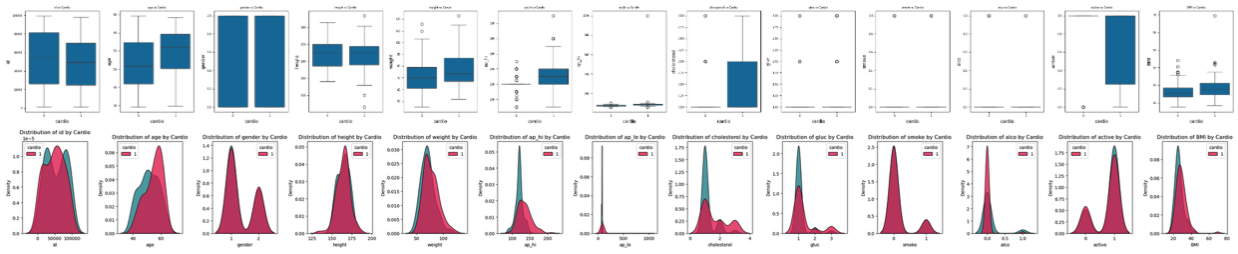
This cardiovascular disease dataset contains 70,000 data samples with 12 different features. These include: age (in days), height (in centimeters), weight (in kilograms), gender (1 indicates female, 2 indicates male), systolic blood pressure (ap_{hi}), diastolic blood pressure (ap_{lo}), cholesterol level, glucose level, smoking status, alcohol intake status, physical activity status, and the presence or absence of cardiovascular disease. According to the CDC (Heart Disease Risk Factors), key risk factors for heart disease include high blood pressure, high cholesterol, and smoking. Therefore, this dataset contains more than enough relevant features to explore correlations with the diagnosis outcome. To further explain some of the features: systolic blood pressure represents the pressure in your blood vessels when your heart beats and pumps blood, while diastolic blood pressure represents the pressure in your arteries when your heart is at rest between beats. Overall, this dataset provides a large number of samples and a wide variety of features, making it great for experimenting with different combinations of variables to enhance prediction performance.

Later in the project, I wanted to compare results from the primary dataset with another dataset that includes symptoms of heart disease as features. I found a secondary dataset from the following source: Cardiovascular_Disease_Dataset (2023, December 9). https://www.kaggle.com/datasets/jocelyndumlao/cardiovascular-disease-dataset.

It includes features such as Age, Gender, Resting Blood Pressure, Serum Cholesterol, and Fasting Blood Sugar which are similar to the primary dataset, but also contains unique features like Chest Pain Type, Resting Electrocardiogram Results, Maximum Heart Rate Achieved, Exercise-Induced Angina, Oldpeak, Slope of the Peak Exercise ST Segment, Number of Major Vessels, and a target classification. This dataset focuses on a different set of attributes for predicting cardiovascular disease, which may lead to different prediction results compared to the primary dataset.

Data

2.1 Visualization



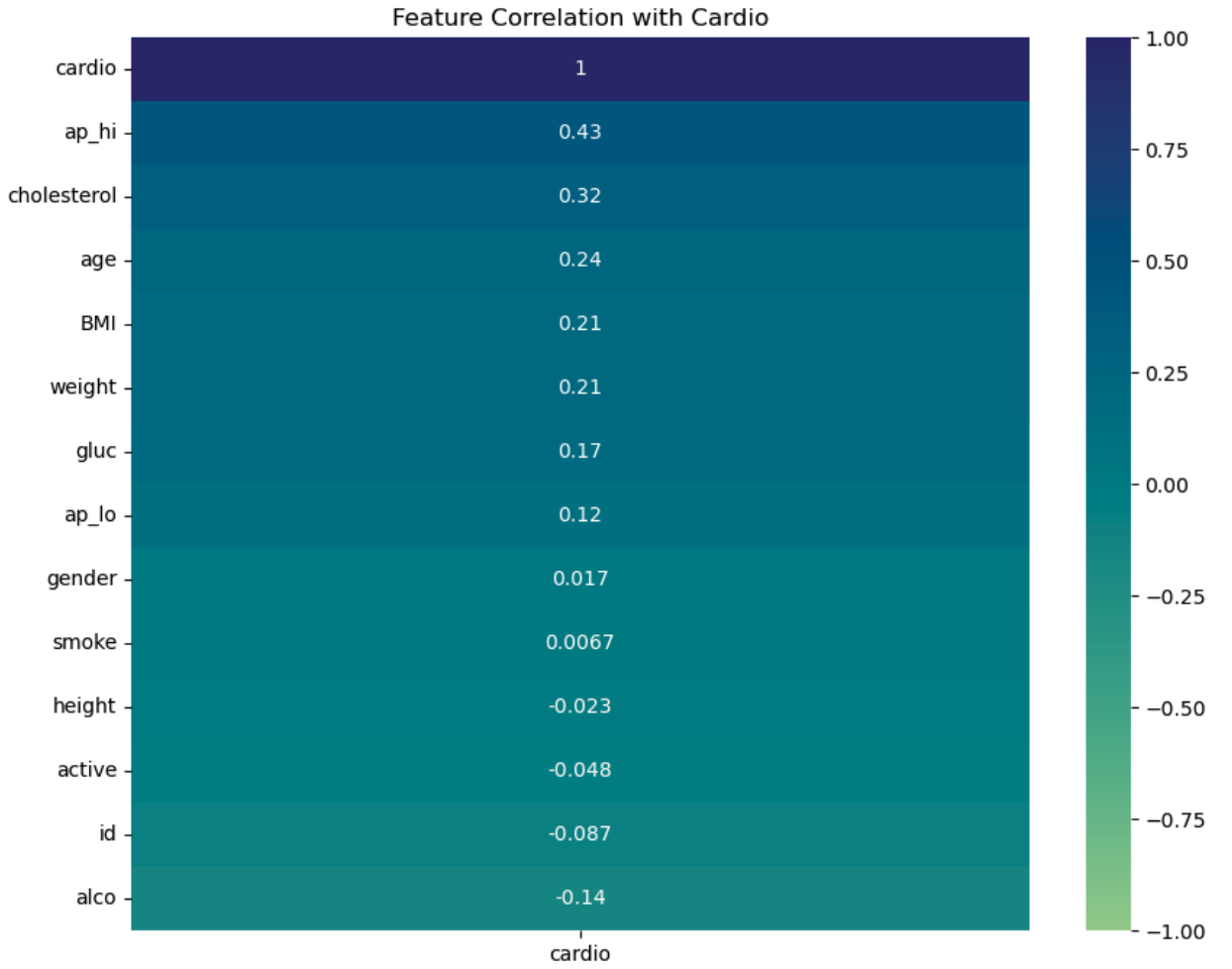
In these two types of visualizations, I wanted to see whether there are differences in the distribution of each feature between people diagnosed with cardiovascular disease and those who are not.

The first set of visualizations, which are boxplots, show clear differences in features such as age, weight, ap_{hi} (systolic blood pressure), cholesterol, active status, and BMI. Except for the "active" feature, individuals diagnosed with cardiovascular disease tend to have higher values in these features.

The second set of visualizations is used to observe distribution differences in continuous, numarical features such as age, weight, and blood pressure. In these plots, some features like gender, height, smoking status, and active status appear to have similar shapes and overlap significantly between the two groups. However, noticeable differences can still be seen in age, weight, blood pressure, cholesterol, and glucose levels.

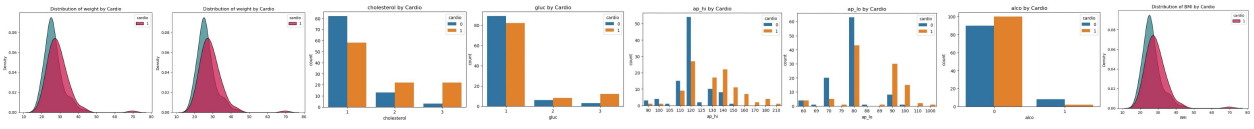
The glucose feature did not show a significant difference in the boxplot but does show some noticeable separation in the second visualization. Also, in the second set of visualizations, the “active” feature does not show a significant difference between those diagnosed with cardiovascular disease and those who are not, as was observed in the boxplot. This may be due to the y-axis scale in the boxplot being too precise for this feature.

From these two visualizations, we can conclude that features like ‘id’, ‘gender’, ‘height’, ‘smoke’, and ‘active’ may not contribute significantly to the risk of cardiovascular disease.



This visualization indicates the correlation between cardiovascular disease and each feature. From this, we can observe significant differences in correlation between the top eight features and the rest, such as gender, smoking, height, activity, and ID.

One interesting point I noticed is that although BMI is calculated using height which by itself does not have a strong correlation with cardiovascular disease, BMI shows a stronger correlation than weight alone. This suggests that creating new features based on existing ones can result in stronger correlations and may improve predictive performance, even if the original features used in the calculation do not individually have high correlation with the target variable.



When we look each visualization closely, we can observe a relatively clear difference in the disparity of

each feature, as these features have a higher correlation with the target value.

Based on these visualizations, I selected the top 8 and top 3 features based on their correlation values. I will use these selections to determine whether feature selection impacts the results in this project.

3 Methods

In this section I will use Logistic Regression. In class, we learned that logistic regression is a supervised machine learning algorithm used for classification. It is unique because it uses the sigmoid function, which returns a probability value between 0 and 1 based on the independent variable inputs. Since my label for Cardiovascular Disease is either 0 or 1, I thought this algorithm was the best fit for my project. Additionally, I use Neural Network as my second algorithm since I wanted to get more knowledge and practice of it but it can handle more complex, nonlinear relationship that logistic regression may not be able to handle.

3.1 Logistic Regression

Logistic regression is a supervised machine learning algorithm used for classification tasks. Its goal is to predict the probability of an instance belonging to a target class. Since this project aims to predict cardiovascular disease status, logistic regression matches well with the purpose of the project and is a good algorithm to use. It is a statistical method that predicts a binary outcome by analyzing the relationship between independent and dependent variables.

The key characteristic of logistic regression is the use of the sigmoid function. This function returns probability values ranging from 0 to 1 based on the input of independent variables. The equation is provided in the logistic regression notes and is as follows:

$$P(y \mid \mathbf{x}; \mathbf{w}) = f(\mathbf{x}; \mathbf{w}) = a = \frac{1}{1 + e^{-z}} \tag{1}$$

$$= \frac{1}{1 + e^{-\mathbf{w}^\top \mathbf{x}}} \tag{2}$$

$$P(y \mid \mathbf{x}; \mathbf{w}) = f(\mathbf{x}; \mathbf{w}) = a = \frac{e^{z}}{1 + e^{z}} \tag{3}$$

$$= \frac{e^{\mathbf{w}^\top \mathbf{x}}}{1 + e^{\mathbf{w}^\top \mathbf{x}}} \tag{4}$$

The difference between these two equations is that $\frac{e^z}{1+e^z}$ helps prevent overflow when z is a very large positive value, while $\frac{1}{1+e^z}$ helps prevent overflow when z is a very small negative value. The sigmoid function is also known as the activation function used in logistic regression.

In addition to the sigmoid function, we use the Binary Negative Log Likelihood (NLL) as the loss function. The formula is provided in the notes:

$$NLL(\mathbf{w}) = -\frac{1}{N} \sum_{n=1}^{N} y_n \log[f(\mathbf{x}_n; \mathbf{w})] + (1 - y_n) \log[1 - f(\mathbf{x}_n; \mathbf{w})]$$

Here, $f(x; w)$ indicates the predicted probability of the positive class and ranges between 0 and 1. A smaller predicted probability results in a larger loss value, while a higher predicted probability leads to a smaller loss value. One of the advantages of NLL is its compatibility with gradient descent, which makes it useful for updating model weights during training.

Logistic regression can also be extended to multi-class classification problems by using the softmax activation function. Softmax is a generalization of the sigmoid function for handling multiple classes. However, since I am predicting cardiovascular disease status, which is a binary classification task, I use logistic regression with the sigmoid function.

3.2 Neural Networks

The perceptron is a fundamental component of neural networks and is often considered similar to a biological neuron. It computes the dot product of input features and their corresponding weights. Then, an activation function is applied to convert the continuous value into discrete outputs such as [-1, 1], based on a defined threshold. The neuron in a neural network is very similar to a perceptron. Each input is assigned a weight, forming a linear combination, as described by the following equation from the notes:

$$Z = W^\top X + b$$

After computing this linear combination, an activation function will be applied, and the resulting value is passed as the input to the next layer.

A neural network consists of an input layer, one or more hidden layers, and an output layer.

- The input layer receives the input data.
- The hidden layers, which will be between the input and output layers, are where most of the learning occurs. You can choose the number of hidden layers to stack depending on the complexity of the problem. According to this Medium article, the number of neurons in the hidden layer should typically fall between the size of the input and output layers.
- The output layer contains neurons that produce the final prediction.

The non-linear activation function is one of the most important components of a neural network because it allows the network to model complex, non-linear relationships. Several activation functions exist, and in this class, we have covered sigmoid, tanh, and linear activation functions.

As explained in the logistic regression section, the sigmoid activation function outputs values in the range of 0 to 1. However, it does not perform well in neural networks for two main reasons:

- The mean of the sigmoid function is 0.5, while it is recommended to standardize data to have a mean of 0.
- The sigmoid function saturates near 0 and 1, which causes its derivatives to approach zero. This leads to the disappearing gradient problem during training.

The tanh activation function is similar to the sigmoid but returns values in the range [-1, 1], which provides mean of 0. This makes it a better choice than sigmoid. However, it still suffers from the saturation problem. The equation is as follows:

$$g(z) = \tanh(z) \tag{5}$$

$$= \frac{e^z - e^{-z}}{e^z + e^{-z}} \tag{6}$$

$$g'(z) = 1 - \tanh^2(z) \tag{7}$$

The linear activation function is used when no activation is desired. In this case, the output is equal to the input, which can be useful for output layers in regression tasks.

The goal of the neural network is to compute the output from each layer effectively. According to the notebook, the prediction is computed as follows:

- Compute $Z^{[1]}$ and apply the tanh activation function g:

$$Z^{[1]} = XW^{[1]} + b^{[1]} \tag{8}$$

$$A^{[1]} = g(Z^{[1]}) \tag{9}$$

- Compute $Z^{[2]}$, the input to the output layer. Since we do not apply an activation function to the output neuron, we use a linear activation:

$$Z^{[2]} = A^{[1]}W^{[2]} + b^{[2]} \tag{10}$$

$$A^{[2]} = f(Z^{[2]}) \tag{11}$$

The final prediction $\hat{y}$ is equal to $A^{[2]}$.

4 Experiment

4.1 Hypothesis

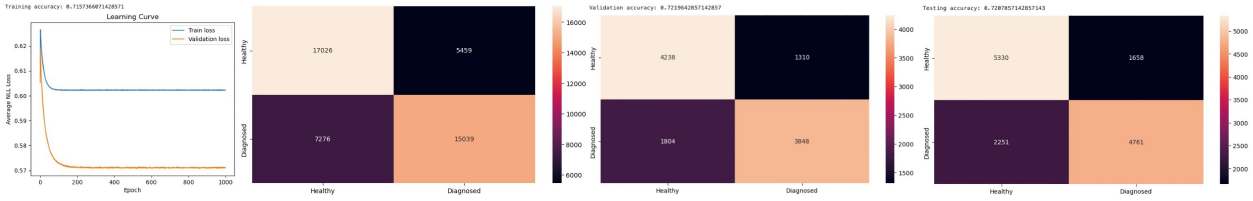
My hypothesis for both the logistic regression and neural network models is that as I narrow the features to those with higher correlation (from all features to the top 8, then to the top 3), model performance will improve. This is because selecting only the most correlated features helps remove noise from the dataset, leading to more accurate predictions. I was not certain which model would perform better, but since neural networks are generally more capable of capturing complex relationships in data, I expected the neural network model to achieve higher accuracy.

4.2 Experimental Setup

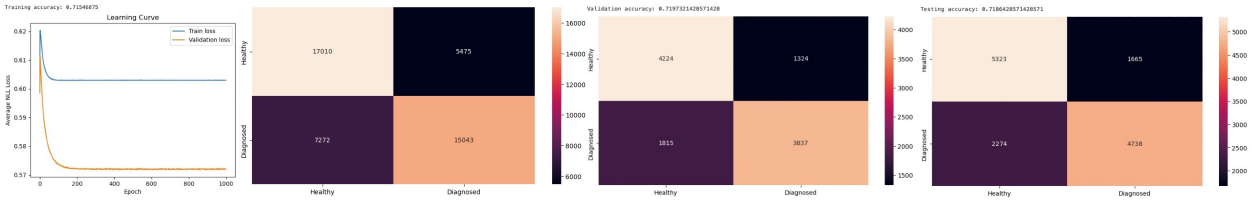
Initially, I ran the logistic regression model with three different processed datasets which are the entire dataset, a dataset with the top 8 features with the highest correlation values, and another with the top 3 features. I then applied the same approach to the neural network model. To compare performances, I also used a neural network model from TensorFlow to see if there was any difference between it and my custom neural network. For the TensorFlow model, I tested both the entire dataset and the dataset with the top 8 selected features to observe any performance differences. Next, I selected features based on the CDC article about heart disease risk factors and ran both logistic regression and neural network models using those features. I experimented with different numbers of neurons to see if that impacted performance and to determine the optimal number of neurons for the CDC-based feature set. Finally, I compared results using another dataset that had fewer samples but included symptoms, not just regular measured body statistics.

5 Results

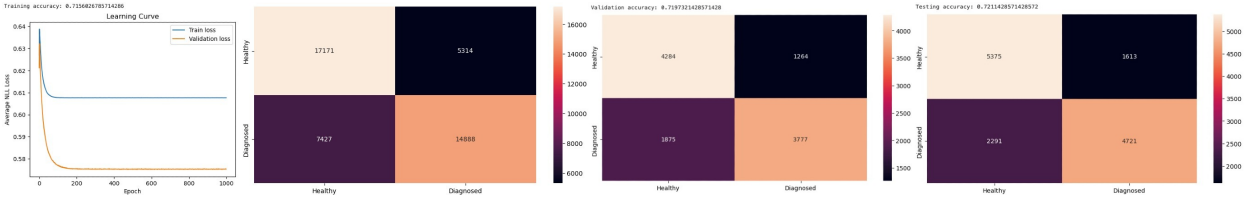
First, I ran logistic regression using the entire dataset. As a result, I obtained a training accuracy of 71.57%, validation accuracy of 72.19%, and testing accuracy of 72.07%. These results are not particularly high, but they were expected since some features had low correlation values.



Next, I trained the same model using the top 8 features with the highest correlation values. The results were a training accuracy of 71.97%, validation accuracy of 72.19%, and testing accuracy of 71.86%. This showed little to no improvement compared to using the full dataset.



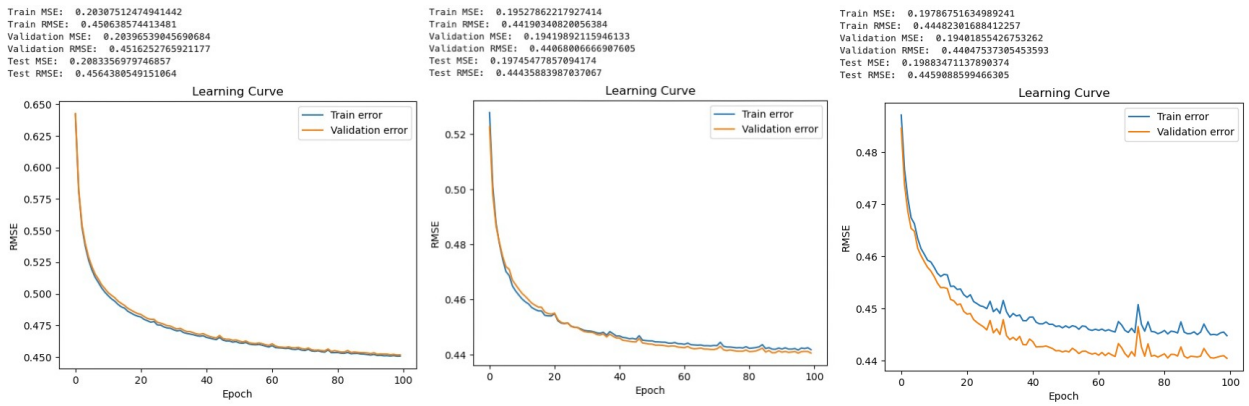
Lastly, I trained the model using only the top 3 selected features. I achieved a training accuracy of 71.56%, validation accuracy of 71.97%, and testing accuracy of 72.11%These results contradicted my hypothesis. According to Number Analytics, carefully selecting features can improve logistic regression accuracy, making the model more stable and easier to maintain. pred.



I then applied a similar approach using a neural network model. First, I trained the model on the entire dataset and got a training RMSE of 0.4506, validation RMSE of 0.4516, and testing RMSE of 0.4564. These results suggest that the model performs slightly better than random guessing (since the label is binary and the mean value is 0.5).

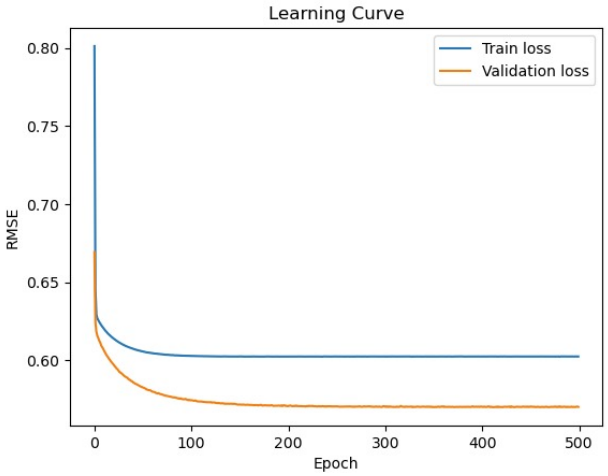
Next, I trained the neural network using the top 8 selected features. This gave me a training RMSE of 0.4419, validation RMSE of 0.4407, and testing RMSE of 0.4444.

Then, I used only the top 3 features for training. The results were a training RMSE of 0.4448, validation RMSE of 0.4405, and testing RMSE of 0.4459. This behavior was very similar to what I observed with logistic regression: selecting features with higher correlation did not significantly improve performance.

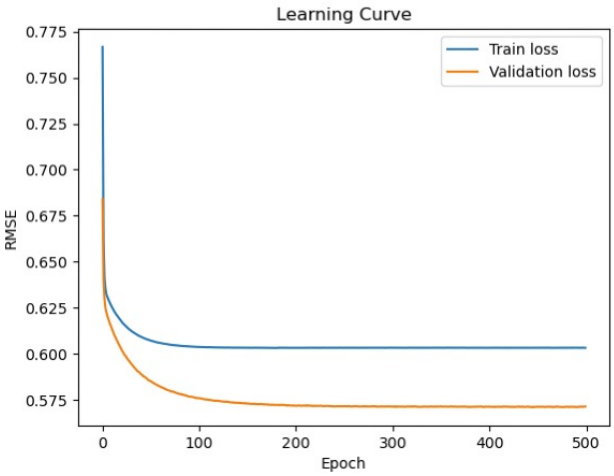


To compare models, I also tested a neural network built with TensorFlow. Using the entire dataset, I got a training, validation, and testing RMSE of 0.4404. Using the top 8 selected features, I got a training, validation, and testing RMSE of 0.4408. Interestingly, the TensorFlow model returned identical RMSEs for all sets, and the values were very similar to those from my custom neural network model. Therefore, there was no significant difference in accuracy between the TensorFlow model and my implementation in this prediction project.

```
1400/1400 ━━━━━━━━━━━━━━━━━━━━ 0s 135us/step
350/350 ━━━━━━━━━━━━━━━━━━━━ 0s 136us/step
438/438 ━━━━━━━━━━━━━━━━━━━━ 0s 146us/step
Train MSE:  0.193953774837823
Train RMSE:  0.44040183337245886
Validation MSE:  0.19094541736921639
Validation RMSE:  0.44040183337245886
Test MSE:  0.19251167787778384
Test RMSE:  0.44040183337245886
```



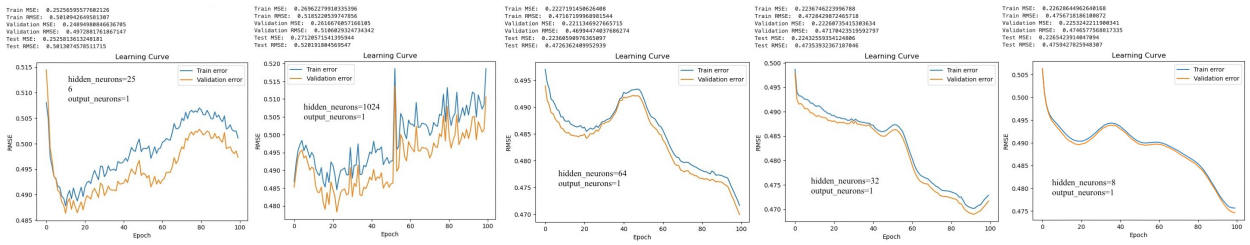

```
1400/1400 ━━━━━━━━━━━━━━━━━━━━ 0s 136us/step
350/350 ━━━━━━━━━━━━━━━━━━━━ 0s 132us/step
438/438 ━━━━━━━━━━━━━━━━━━━━ 0s 146us/step
Train MSE:  0.19428632750940317
Train RMSE:  0.4407792276292103
Validation MSE:  0.19137459002766813
Validation RMSE:  0.4407792276292103
Test MSE:  0.19326684637813696
Test RMSE:  0.4407792276292103
```



Now, instead of selecting features based on correlation values, I referred to the Centers for Disease Control and Prevention (CDC). According to the CDC (https://www.cdc.gov/heart-disease/risk-factors/index.html), heart disease risk factors include high blood pressure, high cholesterol, and smoking. I preprocessed the data to select only ap_{hi}, ap_{lo}, cholesterol and smoking status as features to predict cardiovascular disease.

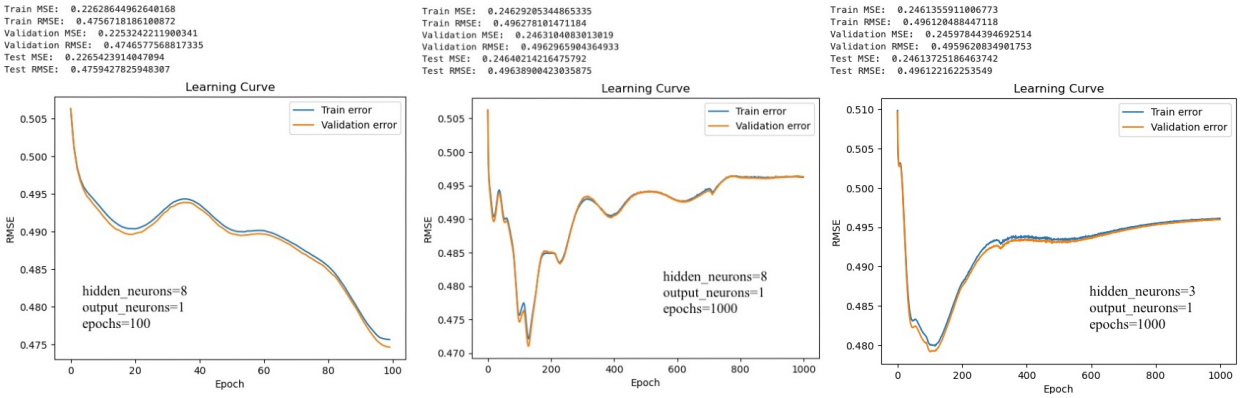
As a result, I obtained a training accuracy of 71.18%, validation accuracy of 72.12%, and testing accuracy of 71.39%. These results are very similar to the previous ones and do not indicate strong predictive performance.

Since I was not able to get satisfactry result with any feature selection, I will change number of hiddel layers. According to Medium article, the number of neurons in the hidden layer should typically fall between the size of the input and output layers. Therefore, in this case, reducing the number of hidden layers may lead to an increase in model accuracy.



As you can see, when the number of layers decreases, the graph becomes more downward-sloping and smoother. However, the accuracy still remains around 47% to 52 %. Based on these results, we cannot expect significant improvement by optimizing the number of hidden layers.

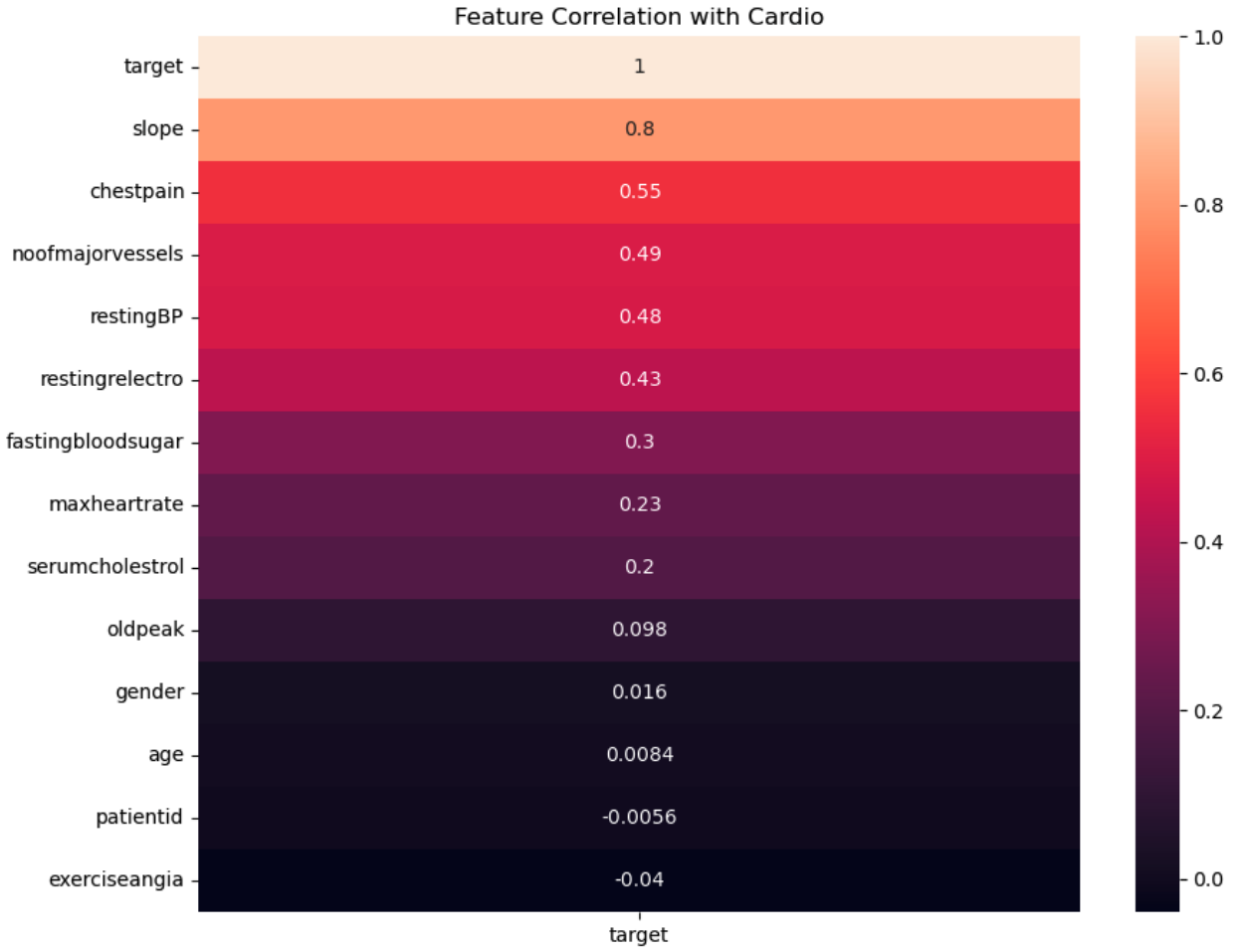
Now, I changed the number of epochs. Since my dataset has various features and a relatively high complexity in the relationship between each feature and the target value, I wanted to observe the effect of increasing the number of epochs. Additionally, since I previously experimented to determine the best number of hidden layers, I also wanted to see how changing the number of output layers would affect the results.



As a result, increasing the number of epochs to 1000 caused the model to become less accurate. From the middle graph, the optimal number of epochs for this model appears to be around 160. However, even at that point, the RMSE value remains above 0.470, which is not ideal. Increasing the number of output layers helped smooth the curve, but the general trend and values remained very similar to those in the middle graph. Therefore, I cannot expect significant improvement from changing the number of epochs or output layers.

5.1 Compare with Another Data

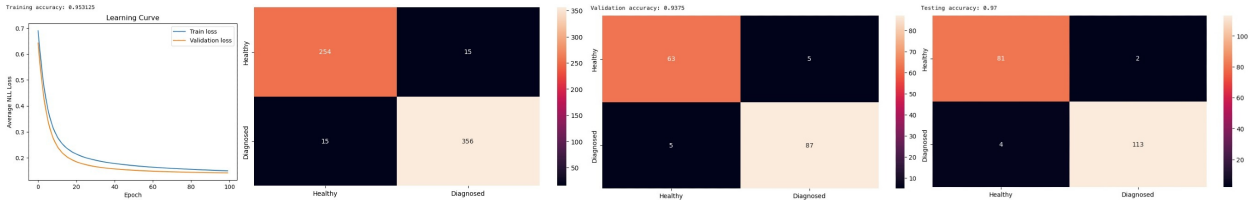
As I mention in Data section, I want to compare results from the primary dataset with another dataset that includes symptoms of heart disease as features.



This visualization shows the correlation between cardiovascular disease and each feature in the secondary dataset. From this, we can observe that the features in the secondary dataset have relatively higher correlations with the target value compared to the primary dataset.

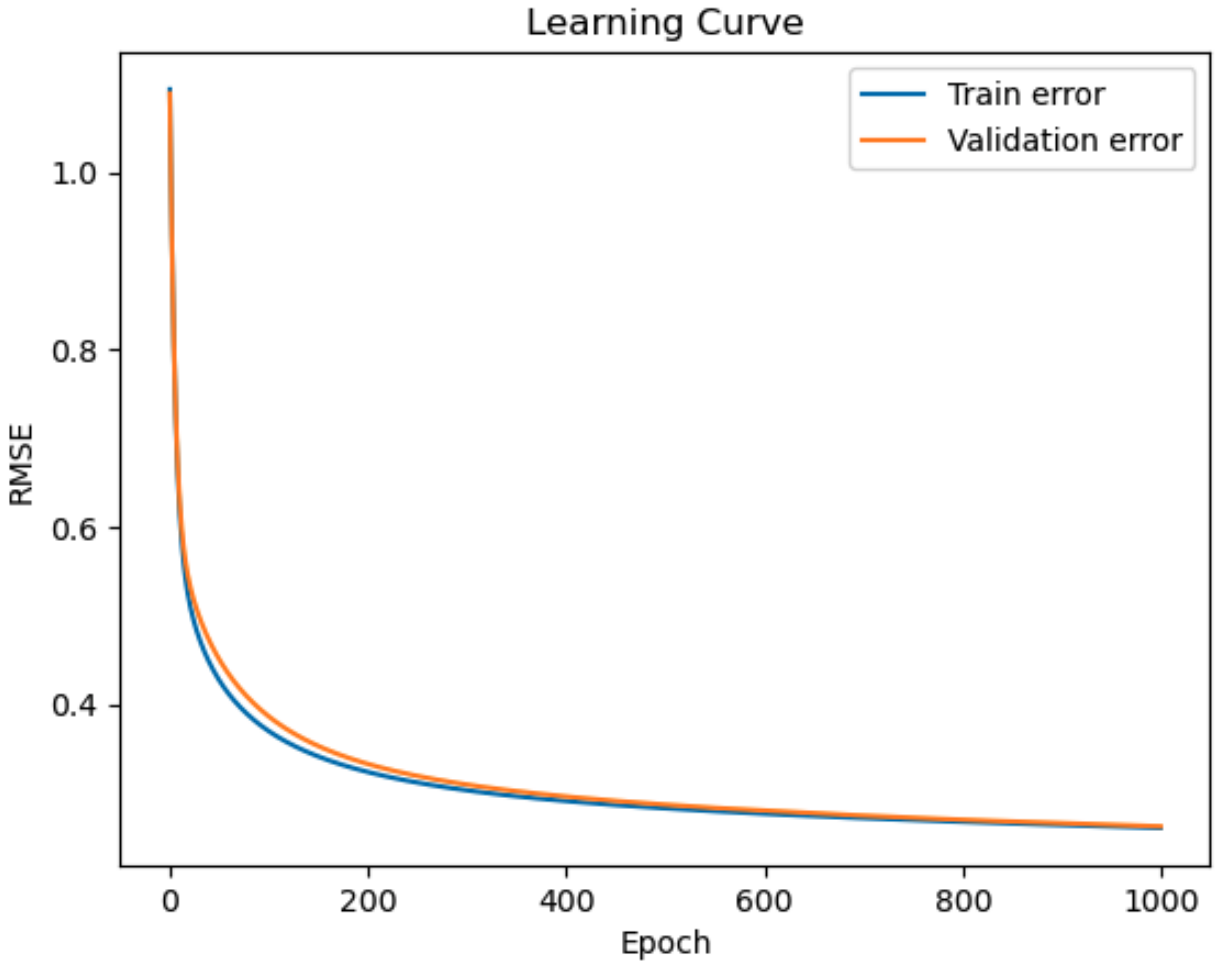
In the primary dataset, only 5 features had a correlation value higher than 0.2, and none had a correlation above 0.5. However, in the secondary dataset, 2 features ('slope' and 'chestpain') have correlation values over 0.5, and 8 features have correlation values higher than 0.2, which is a significant improvement compared to the first dataset.

First, I ran logistic regression using the entire dataset. As a result, I obtained a training accuracy of 95.31%, validation accuracy of 93.75%, and testing accuracy of 97%. These results are excellent and show a significant improvement over the previous training results.



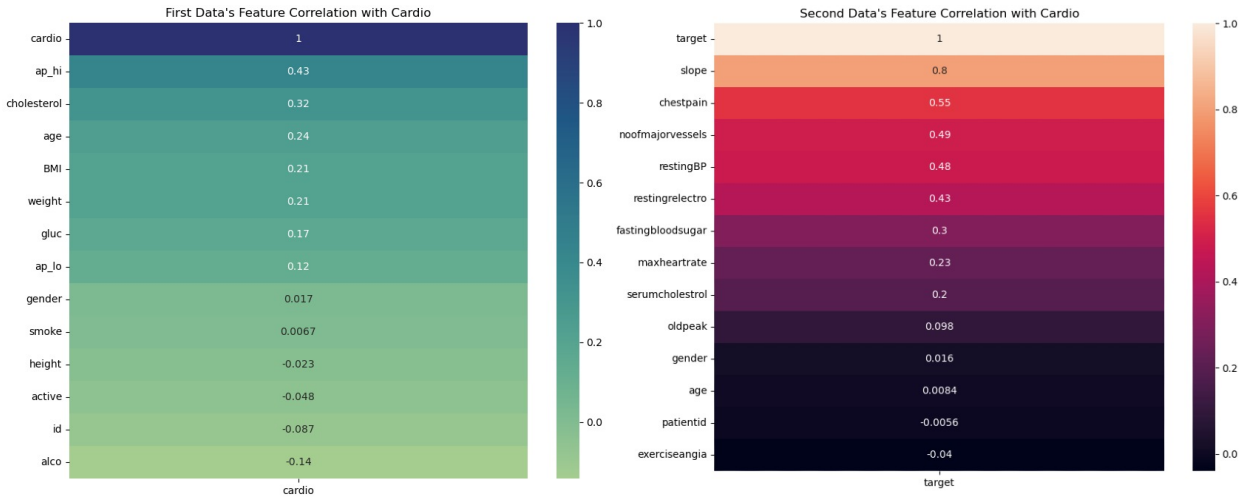
I also trained a neural network model using the secondary dataset. As a result, I achieved a training RMSE of 0.2609, validation RMSE of 0.2627, and testing RMSE of 0.2585, which also indicate excellent performance and a significant improvement over the previous neural network results.

```
Train MSE:  0.0680588405538947
Train RMSE:  0.26088089342436466
Validation MSE:  0.06899376063525196
Validation RMSE:  0.26266663403495305
Test MSE:  0.06679668579776919
Test RMSE:  0.258450548070166
```



Even though predictions using the secondary dataset showed higher accuracy (above 90

As I hypothesized, the lower prediction accuracy of the primary dataset is likely due to the generally weaker correlation values between its features and the target variable. In contrast, the secondary dataset includes some symptom-based features, which tend to have stronger correlations with the target.



As noted in this AHA journal article, predicting cardiovascular disease remains a complex task, even when various health measures are available and some are known to influence heart disease risk.

Therefore, we cannot determine which dataset is better. However, combining the features from both datasets may provide a more accurate prediction model for cardiovascular disease.

6 Conclusions and Future Work

Cardiovascular disease prediction remains a significant challenge due to the complex relationships between various health indicators and the diagnosis of the disease. As we observed, symptom-based features tend to have higher correlation values than general health measurements. However, relying solely on symptom-based features may contradict the goal of early prediction, which aims to prevent cardiovascular disease before symptoms appear.

However, incorporating both symptom-based features and general health measurements can improve the effectiveness of the diagnostic process. Furthermore, extending this prediction task to a multi-class classification model using logistic regression or a neural network with a softmax function could allow us to assess the severity of cardiovascular disease and the necessity for surgery. This advancement would benefit the healthcare industry and help a wide range of individuals by providing more accurate predictions.

As I explained in the Method section, the use of logistic regression and neural networks was appropriate for this binary classification task. The sigmoid activation function for logistic regression and the tanh activation function for neural networks generally performed well for this project's purpose. Additionally, this project allowed me to gain hands-on experience with two of the algorithms I'm most interested in, and TensorFlow which made it a valuable learning experience.

Most of the models and code I used were from class assignments, so the implementation itself was not the most challenging part. The real challenge was identifying the cause of the lower accuracy when using the primary dataset. To address this issue, I applied various approaches, as described in the Results section. Another difficulty was the long runtime of the model training process, which made hyperparameter tuning more time-consuming and difficult.

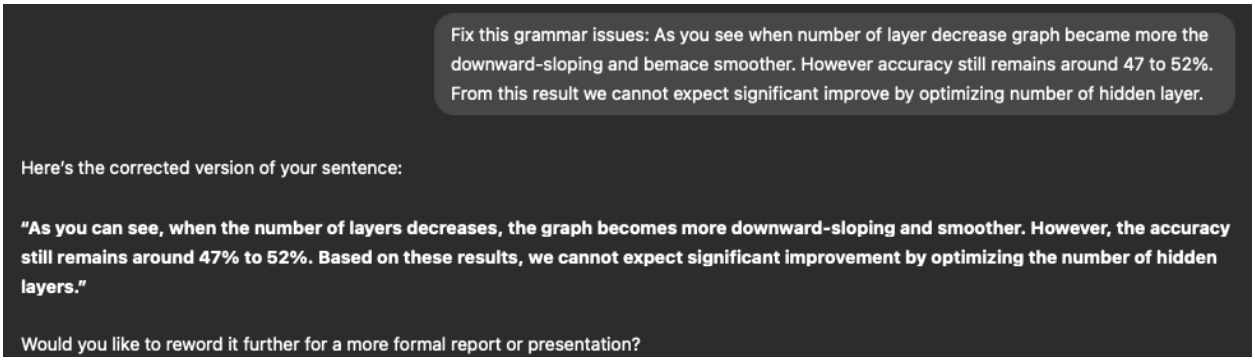
As mentioned earlier, I am considering extending this project to a multi-class classification model that can assess the severity level of each individual's cardiovascular condition. This would be based on a combination of symptom-based features and general health metrics. Since the logistic regression and neural network models I used are adaptable, they can be extended with a new dataset that includes labels indicating the severity of cardiovascular disease. I believe this model is highly scalable. Many of the health metrics used in this project are common to other diseases as well, so with appropriate datasets, this model could be repurposed to predict other health conditions too.

7 Acknowledgments

I, Kento Hopkins, acknowledge the use of AI in completing this assignment and would like to provide a brief explanation of how I utilized AI, specifically ChatGPT, as a tool to support my work. For this assignment, I used ChatGPT to help me fix my grammar mistakes, understand instructions properly and debug my code while properly constructing my training loop and assure that my plots ran smoothly and matched up with the assignment instructions.

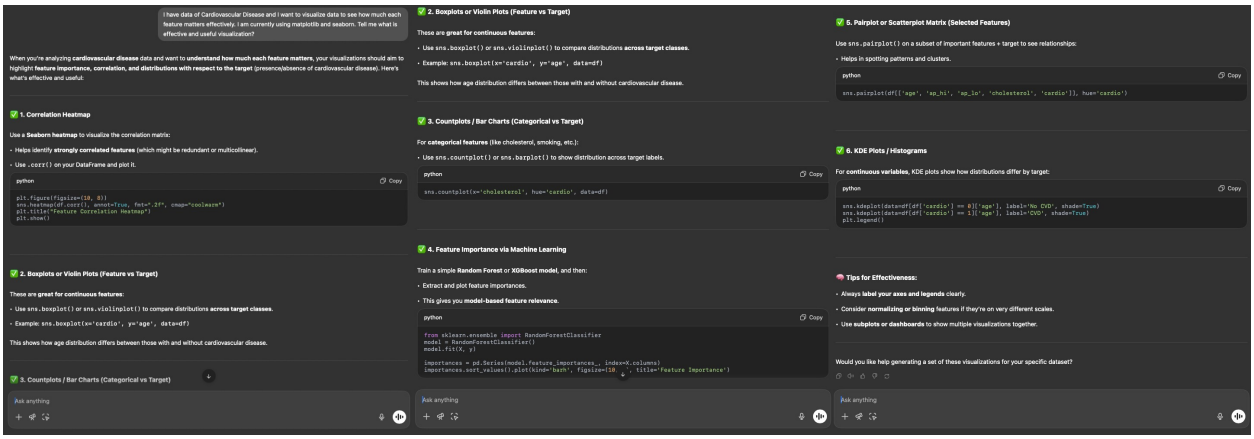
In general, including in this report, I use ChatGPT to correct my grammar. At the beginning of the chat, I told ChatGPT, "I will ask you to fix my grammar multiple times. Just fix grammar issues."

I used the prompt: "Fix these grammar mistakes: [sentences I wrote]." The following is an example.



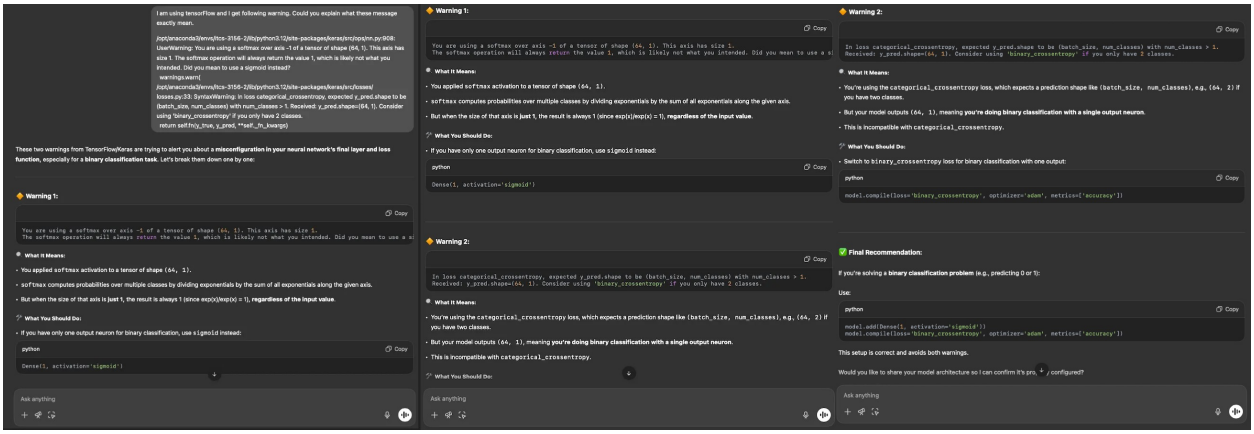
In the Data section, specifically when I was working on data visualization, I wanted to try a new visualization method that shows the disparities of each feature in relation to the target value to determine which features affect the diagnosis of cardiovascular disease. I also wanted to calculate the correlation values between each feature and the target. However, I was unsure about what options were available and didn't know exactly where or how to start searching. Out of curiosity, I asked ChatGPT, and I received several helpful and satisfying suggestions.

- Prompt
  - "I have data of Cardiovascular Disease and I want to visualize data to see how much each feature matters effectively. I am currently using matplotlib and seaborn. Tell me what is effective and useful visualization?"
- Output
  - ChatGPT provided useful visualization options, including correlation heatmaps, boxplots or violin plots, countplots/bar charts, feature importance via machine learning, pair plots or scatter matrix, and KDE plots/histograms.
- Takeaway
  - I was able to learn what options are available for data visualization using Matplotlib and Seaborn. I used correlation heatmaps, boxplots, bar charts, and KDE plots in this project.



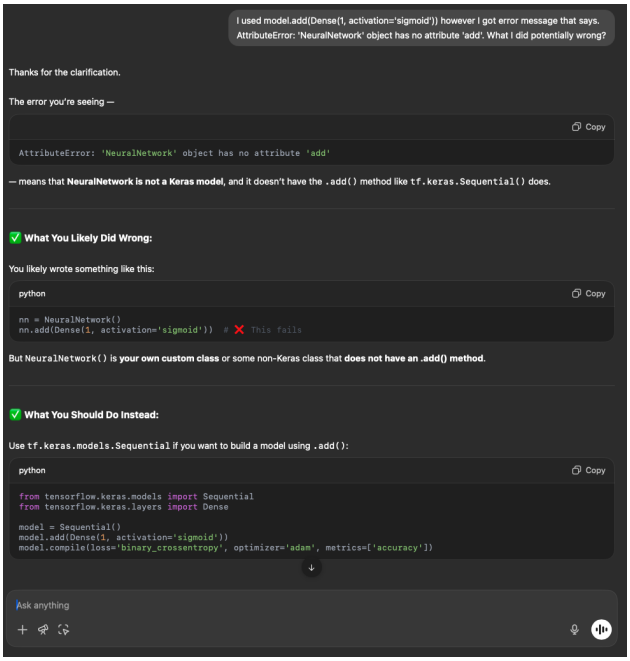
In this project, I used TensorFlow. However, since I was not very familiar with it, I encountered errors and wasn't entirely sure what they meant. Therefore, I asked ChatGPT to explain the error messages so I could understand the issues.

- Prompt
  - "I am using tensorFlow and I get following warning. Could you explain what these message exactly mean. /opt/anaconda3/envs/itcs-3156-2/lib/python3.12/site-packages/keras/src/ops/nn.py:908: UserWarning: You are using a softmax over axis -1 of a tensor of shape (64, 1). This axis has size 1. The softmax operation will always return the value 1, which is likely not what you intended. Did you mean to use a sigmoid instead? warnings.warn(/opt/anaconda3/envs/itcs-3156 2/lib/python3.12/site-packages/keras/src/losses/ losses.py:33: SyntaxWarning: In loss categorical_crossentropy, expected y_pred.shape to be (batch_size, num_classes) with num_classes ¿ 1. Received: y_pred.shape=(64, 1). Consider using 'binary_crossentropy' if you only have 2 classes. return self.fn(y_true, y_pred, **self._fn_kwargs)"
- Output
  - ChatGPT provided explanations for each warning and recommended that I use Dense(1, activation='sigmoid')..
- Takeaway
  - I was able to learn what Dense(1, activation='sigmoid') is and how to utilize it, which was a great option since my project was a binary classification task, and I used the sigmoid activation function in my logistic regression model..



Based on the results above, I used model.add(Dense(1, activation='sigmoid')). Although I had read the documentation and believed I understood it well, I encountered an error during implementation and struggled to resolve it. Since I originally learned about using model.add(Dense(1, activation='sigmoid')) from ChatGPT, I reached out to ChatGPT for assistance.

- Prompt
  - "I used model.add (Dense(1, activation='sigmoid')) however I got error message that says. AttributeError: 'NeuralNetwork' object has no attribute 'add'. What I did potentially wrong?"
- Output
  - ChatGPT provided an explanation of what I did wrong and recommended using Sequential() to resolve the error.
- Takeaway
  - I was able to learn what Sequential() is and applied the solution provided by ChatGPT from the above prompts.



For this project, I referred to the provided notebooks: "note-perceptron", "note-logistic-regression", "note-neural-networks", and "note-multilayer-neural-networks". I used these resources in the Methods section to deepen my understanding and improve my explanations. Some of the math equations are from these notes. Additionally, I integrated code from "note-multilayer-neural-networks" to implement the model using TensorFlow. I also drew upon my previous coursework and homework assignments to enhance my coding skills and reinforce my understanding of the algorithms used in this project.

8 References

Szabo, Bibor. "How to Create a Seaborn Correlation Heatmap in Python?" Medium, 14 Dec. 2021, medium.com/@szabo.bibor/how-to-create-a-seaborn-correlation-heatmap-in-python-834c0686b88e.

Seaborn.Boxplot — Seaborn 0.13.2 Documentation. seaborn.pydata.org/generated/seaborn.boxplot.html.

Choosing Color Palettes — Seaborn 0.13.2 Documentation. seaborn.pydata.org/tutorial/color_palettes.html.

Al-Alshaikh, Halah A., et al. "Comprehensive Evaluation and Performance Analysis of Machine Learning in Heart Disease Prediction." Scientific Reports, vol. 14, no. 1, Apr. 2024, https://doi.org/10.1038/s41598-024-58489-7.

Harvard Health. "A Closer Look at Heart Disease Risk." Harvard Health, 21 Sept. 2021, www.health.harvard.edu/heart-health/a-closer-look-at-heart-disease-risk.

National Library of Medicine. "High Blood Pressure." Hypertension — MedlinePlus, medlineplus.gov/highbloodpressure.html

"Understanding Blood Pressure Readings." www.heart.org, 4 Mar. 2025, www.heart.org/en/health-topics/high-blood-pressure/understanding-blood-pressure-readings.

GeeksforGeeks. "Applying PCA to Logistic Regression to Remove Multicollinearity." GeeksforGeeks, 5 June 2024, www.geeksforgeeks.org/applying-pca-to-logistic-regression-to-remove-multicollinearity.

"Everything You Need to Know About Logistic Regression." Spiceworks Inc, 10 Mar. 2025, www.spiceworks.com/tech/artificia intelligence/articles/what-is-logistic-regression.

GeeksforGeeks. "Logistic Regression in Machine Learning." GeeksforGeeks, 3 Feb. 2025, www.geeksforgeeks.org/understandin logistic-regression.

GeeksforGeeks. "What Is a Neural Network?" GeeksforGeeks, 3 Apr. 2025, www.geeksforgeeks.org/neural-networks-a-beginners-guide.

Krishnan, Sandhya. "How to Determine the Number of Layers and Neurons in the Hidden Layer? — by Sandhya Krishnan — Geek Culture — Medium — Geek Culture." Medium, 6 Aug. 2022, medium.com/geekculture/introduction-to-neural-network-2f8b8221fbd3.

Lee, Sarah. 5 Key Factors Boosting Logistic Regression Accuracy by 20%. www.numberanalytics.com/blog/logistic-regression-accuracy-5-factors.

Cardiovascular_Disease_Dataset. 9 Dec. 2023, www.kaggle.com/datasets/jocelyndumlao/cardiovascular-disease-dataset.

Cardiovascular Disease Dataset. 20 Jan. 2019, www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset.

Moran, Andrew E., and Lee Goldman. "Predicting the Future Prevalence of Cardiovascular Disease: The Good, the Bad, the Known, and the Unknown." Circulation, vol. 150, no. 4, June 2024, pp. 253–54. https://doi.org/10.1161/circulationaha.124.070265.

IBM. "What Is Principal Component Analysis?" IBM, https://www.ibm.com/think/topics/principal-component-analysis.

IBM. Logistic Regression. IBM, https://www.ibm.com/think/topics/logistic-regression.

IBM. Neural Networks. IBM, https://www.ibm.com/think/topics/neural-networks.

GeeksforGeeks. "Advantages and Disadvantages of Logistic Regression." GeeksforGeeks, 2 Jan. 2025, www.geeksforgeeks.org/advantages-and-disadvantages-of-logistic-regression.

GeeksforGeeks. “Logistic Regression in Machine Learning.” GeeksforGeeks, 3 Feb. 2025, www.geeksforgeeks.org/understandin logistic-regression.

Kleimann, Sebastián Gerard Aguilar. “Logistic Regression for Binary Classification.” Towards Data Science, 5 Mar. 2025, towardsdatascience.com/logistic-regression-for-binary-classification-56a2402e62e6.

“Causes and Risk Factors — NHLBI, NIH.” NHLBI, NIH, 24 Mar. 2022, www.nhlbi.nih.gov/health/heart-attack/causes.

“Logistic Regression for Binary Classification With Core APIs.” TensorFlow, www.tensorflow.org/guide/core/logistic_regressio

”tf.keras.Sequential. TensorFlow.” TensorFlow, https://www.tensorflow.org/api_docs/python/tf/keras/Sequential.

“Heart Disease Risk Factors.” Heart Disease, 2 Dec. 2024, www.cdc.gov/heart-disease/risk-factors/index.html.

GeeksforGeeks. “What Is Perceptron — the Simplest Artificial Neural Network.” GeeksforGeeks, 21 Oct. 2024, www.geeksforgeeks.org/what-is-perceptron-the-simplest-artificial-neural-network.

9 Source Code

The complete source code for this project is available on GitHub:https://github.com/kentoHopkins/Cardiovascular-Disease-Predictio