ITM 703
Artificial Intelligence in Business

Final Course Project
Heart Disease Classification Using Neural Networks

Group Members
Jacqueline Chung
(Group Members' names are excluded for privacy)

Prepared For:
Professor Dr. Morteza Zihayat Kermani
School of Information Technology Management
Ted Rogers School of Management

April 12th, 2020
Ryerson University

Table of Contents

Introduction

The learning objective of this project is to gain hands-on experience using Python to build models from real-world datasets. To achieve this, different testing parameters of the network will then measure them against performance metrics of test loss.

A predictive neural network is created by using heart patient data to predict if a patient has heart disease. The goal is to achieve the highest possible accuracy, minimizing test loss. Factors of run time and other performance measurements such as Precision, F-Score are considered. Initially, the patient data is explored to assess any skewed data points and initial trends from the raw data. Correlations are assessed to determine any outstanding relationships between data variables. Following data exploration, the neural network can begin to be built. In building the neural network, tests help to determine an optimized network. A parameter sensitivity analysis provides an understanding of the variance in the impact that the network's parameters have on the accuracy of the model. These parameters are ranked based on their accuracy rate. Modified Parameters include activation functions used, optimizer, epochs, learning rate, number of hidden/input layers, and number of neurons. After performing several tests and sensitivity analysis, the model will be looked at, and a discussion will be provided outlining key findings. Based on this discussion, a conclusion on the neural network will highlight the outcome of the network and recommendations for future use or development of this neural network.

This report will be presenting the following:

1. Data exploration including correlation analysis
2. TensorBoard with Hyperparameters searching or tuning
3. Evaluation of each change in Hyperparameter
4. Discussion of the evaluations
5. Conclusion and summary of the main results

Data Exploration

Data Cleaning

Before exploring the data, the dataset was reviewed, and it was determined that no cleaning was necessary. All values had a float or int value, and there were no null values identified.

Methodology

Visual diagrams were used to examine any skewing of the data with outliers and distribution. Bar Charts, Scatter Plots, and Box Plots are used to see the volume of each variable. Heat Maps and Probability Analysis were used to assess the correlation between the variables and the target value. The target value is whether an individual has heart disease indicated by a binary value of 0 or 1, 1 representing that the patient has been diagnosed with heart disease. The data was not normalized.

Bar Chart (Appendix 1)

The variables that showed a high level of skew are listed below. Other variables and data points did not demonstrate significant skewing.

Age was assessed to range between 30 and 75 years old. The majority of representation being between 45 and 65.

The rate of Thalassemia was determined primarily in the range of fixed defect and reversible defect. A small number of total patients were showing 'normal', meaning that the majority of patients had or currently suffer from a blood disorder.

Oldpeak examines the change in ST wave induced by exercise relative to rest. This is a measurement from the ECG of the patient. The data shows that patients primarily have normal and ST-T wave abnormality.

Chest pain (CP) experienced skews towards typical angina chest pain. This means that the pain associated with their chest is typical of heart-related issues. Atypical angina has strong representation, but atypical include chest pain not related to the heart.

Gender revealed that a large majority of patients are male.

Scatter Plots (Appendix 2)

Cholesterol was measured in comparison to the maximum heart rate achieved (thalach). There is a clustering of data that shows a cholesterol level of 200-350mg/dl is strong in a blood pressure above 120 mm Hg. These are common risk indicators for heart disease risk (Story, 2020).

The resting heart rate was compared against age. Patients had a resting heart above 100 beats per minute and had a long tail up to 180 beats per minute. The age is above 40 years old, where many individuals begin to be at risk for heart disease. These are extremely high values that are also indications of heart disease risk (Sloan, 2019). Both of these skewed data points come as no surprise as those reporting heart disease will likely have higher cholesterol, blood pressure, and resting heart rate. All of these factors are often key attributes of heart disease patients.

Box Plot (Appendix 3)

Once again, cholesterol showed to have many strong outliers that ranked high above the averages of the data. This was compared with the target value of having heart disease number of major vessels, and resting heart rate. Cholesterol showed many influential outliers and skewed on each of these comparison metrics.

Heat Mapping (Appendix 4)

Heat mapping determined that there are no strong correlations between variables. No two variables achieved a correlation higher than 0.3. However, when a heat map was created with target values, chest pain, slope and maximum resting heart rate all showed correlations above 0.35. These are variables that we can target when designing our neural network and weights.

It is shown that the correlation between the "target" variable and other variables is low, below around 0.8. This means that the target variable may not be the best target class for this dataset.

On the other side, for the second heatmap, it seems that variables have correlations less than 0.3, which shows that there is no redundancy.

Probability Analysis (Appendix 6)

The probability plot shows us whether the data fits on a normal distribution for age and target. This shows that both age and target both fit on a normal distribution.

Target Value Analysis (Appendix 5,7)

From the exploration exercises that target value expressed a great deal as for the percentage of patients who are positive with heart disease. The target variable demonstrates the certainty of which corresponding variables lead to a positive or negative diagnosis. This helps in training as it defines the yes or no answer.

Within the dataset, the target variable provided a positive bias, with 54% of patients diagnosed with heart disease. With a positive bias, it is easier to view what corresponding variables correlate and provide a positive test. Specifically, what types of traits do patients present that correlates with being diagnosed with heart disease. The heat map is a great source of information when searching for correlated variables. Cp, thalach are all in string correlation, while age and sex no not have a great with the target variable. Furthermore, it can be said that heart disease does not discriminate between age or sex and relies on other factors in order for a positive test.

Neural Network Classifier

Network Design Process

Neural networks will make predictions for the classification. Deep learning is a neural network with one or more hidden layers. Artificial neural networks are a computational model based on simple neural units. A neural network uses an iterative process, which is repeatedly improving the answer gradually.

While training the neural network classifier, the dataset is randomly split, where 70% goes to training, and 30% goes to testing. Then, it is built with the following characteristics: Layers (Input, Hidden, Output), Number of Neurons, Optimizer, Activation Function, Learning Rate, Loss Function, and Metrics. Each layer has its activation functions and shape (number of neurons). The output has only two neurons because it would be 0 or 1. Afterward, the model is compiled by calculating the appropriate metrics and loss functions. Finally, the model is fitted and trained using the training data. This will save the metrics into the model’s history, which will be evaluated and validated with the test data. In this case, performance metrics used to support the accuracy are recall, precision, and F1 score.

Hyperparameter (Hparam) method is used to assess the variance in each parameter. Each parameter's sensitivity is assessed based on the change in accuracy and loss function. Each parameter was changed individually from the parameters of the initial test. (Appendix 5)

Activation Function

Different activation functions can change how the network trains, and it’s a scale of correction. Changing different activation functions should be assessed. Common ones being Sigmoid, ReLU, and Tanh. Linear often overcorrects data and will be omitted from this sensitivity test.

Learning Rate

The learning rate determines at what rate the network corrects itself with each iteration. Higher learning rates overcorrect the data creating major inaccuracies in the model. Learning rates that are too low take too long to optimize. The role of a data scientist would be to determine the correct learning rate somewhere in $0.03 < x < 0.1$. However, there is no specific number that is perfect and may vary across data types and activation functions.

Number of Hidden Layers in the Neural Network

Additional layers can be added to a network depending on the complexity of the data set and the accuracy required. Traditional neural networks have been able to operate with a single hidden layer. However, denser models have been built that can take on more layers. Additional layers are commonly used for datasets with time-series data points.

Number of Neurons in Each Hidden Layer

Using too few neurons can cause underfitting, while too many neurons can cause overfitting. Adjusting the number of neurons can help build a more effective accurate network. A few general rules for determining the number of neurons are: number of hidden neurons should be between the size of input and output layer, number of hidden neurons should be 2/3 size of the input layer plus the size of the output layer, number of hidden neurons should be less than twice the size of the input layer.

Number of Neurons in Each Input Layer

With respect to the number of neurons comprising this layer, this parameter is completely and uniquely determined once you know the shape of your training data. Specifically, *the* number of neurons comprising that layer is equal to the number of features (columns) in the dataset. Some NN configurations add one additional node for a bias term.

Number of Epochs

An epoch is one forward and one backward pass through the network. With backpropagation, the weights are updated. With more epochs, the network should optimize more. However, this can be faulty logic and the marginal optimization for epochs may plateau. The learning rate and the activation function have massive impacts on how many epochs it takes to optimize.

Optimizer

Optimizers update weights to minimize the loss function of a network. There are several different variations of optimizers. The most common group type is SGD and Adam.

Running Time

Run time is how long it takes for the network to become optimized. This is usually measured in “epochs”, or iterations. As the iterations increase, the neural network is optimized and strives for a lower test/training loss rate. If a neural network takes less time to optimize data, it can save lots of time and resources to run, which shows an efficient neural network. Some networks may achieve ‘fast’ optimization but may overcorrect the model. This can cause serious issues for overfitting the data, which means the network may not be applicable outside of this dataset.

Accuracy

When conducting a sensitivity analysis, the variance is measured on the change in input parameters and its effect on an outcome. For accuracy, this measures the percentage of outputs that are True Positive (TP) and True Negative (TN) over the total outcomes of the network. This accuracy can be measured on how the network classifies training or test data. Test data is the primary indicator for network accuracy, but through optimization training data, accuracy data is often assessed.

The regular accuracy function, Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) were used to measure accuracy. MAE measures the average magnitude of the errors in a group of predictions. It’s the average over the test sample of the absolute differences between prediction and actual results where all individual differences have equal weight. On the other hand, RMSE is a quadratic scoring rule that measures the average magnitude of the error. It calculates the square root of the average of squared differences between prediction and actual results. From an interpretation standpoint, MAE is a better measurement, while RMSE does not describe average error. However, one distinct advantage of RMSE over MAE is that RMSE avoids the use of taking the absolute value, which is undesirable in mathematical calculations.

The drawback is that accuracy rates are not scalable with different datasets if there is an imbalance that varies from the training/test data to other datasets. If a neural network is to be deemed ‘effective’, it must be able to take in formatted data and provide accurate output. However, if the balance of TP and TN are different in each dataset, the measurement of ‘accuracy’ can vary between networks.

Precision

Measurement of TP over TP + FP. This is a measurement of how correct the positive prediction is in comparison to false positives. This can help examine if the model does give the correct prediction and if it is overfitting. Since the focus is only on TP this does not assess how correct the model is at discarding data that does not qualify. A more refined and narrow view of the network's predictability.

Recall

Measurement of TP over TP + FN. This measures all relevant data and how correct it is in assessing TPs. Since the focus is only on TP this does not assess how correct the model is at discarding data that does not qualify. A more refined and narrow view of the network’s predictability.

F1-Score

This measures the balance between Precision and Recall. This takes an accurate view of the predictability of the network regarding identifying the classifying variable. Since the focus is only on TP this does not assess how correct the model is at discarding data that does not qualify. A more refined and narrow view of the networks’ predictability.

Unmodified Parameters

Through prior research, it was determined that particular parameters did not need to change. For this research, the learning rate was left unmodified at 0.1 for each test. For the parameters that were modified, many were iterated through 40 times.

Ranking the Parameters

An adapted feedforward neural network topology is designed to enable the pairwise analysis of input parameter weights. This enables the ranking of input parameters in terms of importance to output accuracy, without the need to train numerous models (Lotz, 2019). The AUC and Accuracy are the primary measurements for parameter sensitivity. The parameters should be ranked based on their individual effect accuracy and AUC rate. Precision, Recall and F1-Score should also be used as performance measurements, while running time is a secondary measure in ranking the parameters.

Parameters modified and assessed in these tests include:

- **Neurons in Input Layer**: A range of 5 to 1000 were used
- **Neurons in Hidden Layer**: 1 to 3 hidden layers were added to the neural network.
- **Optimizers:** SGD and Adam were used
- **Activation:** Tanh, Sigmoid ReLU were used
- **Epoch:** A range of 5 to 100 were used

Optimized Model

Unmodified Parameters		**Modified Testing Parameters**				
Learning Rate	*Epochs*	*Input Layer Neurons*	*Hidden Layer 1 Neurons*	*Hidden Layer 2 Neurons*	*Optimizer*	*Activation Function*
0.1	5	1000	100	16	SGD	ReLU

Accuracy: 89.87%
As the primary optimizing metric this was the highest achieved accuracy for the testing parameters, while still having strong supporting performance metrics. **Recall** 100%, **Precision** 61%, **F-Score** 77%.

Discussion

Optimal Parameters

(Accuracy was above 80%)

Unmodified Parameters	**Modified Testing Parameters**						
Learning Rate	*Epochs*	*Input Layer*	*Hidden Layer 1*	*Hidden Layer 2*	*Optimizer*	*Activation Function*	*Loss Function*

		Neurons	*Neurons*	*Neurons*			
0.1	5 10	500 750 1000	8 16 100	50 16	Adam SGD	Tanh ReLU	Hinge Squared Hinge

Number of Neurons

500, 750 and 1000 input neurons were optimal to achieve high accuracy; 750 has the lowest loss. Based on our assessment, hidden layers should have fewer neurons than the input layer. The optimal amount of hidden neurons should be 8,16,50,100. When measuring the overall performance, the best would be, with two hidden layers, the first hidden layer is 8, 16 or 100 , then the second hidden layer would be 16 or 50 since it has the lowest loss.

Number of Hidden Layers

First, having 1 hidden layer had good accuracy. Having 2 hidden layers offered a higher accuracy with the data. Adding a 3rd hidden layer, made no significant difference in the performance. The best for this model is 2 hidden layers, as the accuracy is higher than 1 hidden layer. Some models achieved accuracy above 90%, but had each of the other performance metrics at less than 30%, showing it is not an optimized network.

Optimizer

SGD was determined to have the highest accuracy with a smaller number of neurons. Adam performed better with a more significant number of neurons. Overall, Adam and SGD both performed well with 1 and 2 hidden layers, while Adam performed better with 3 hidden layers.

Activation Function

Tanh was determined to be the most effective activation function across different parameters. It worked well with larger amounts of neurons and longer epochs, while consistently performing well with the least amount of loss. Then, ReLU was the second most consistent activation function. Lastly, Sigmoid is the third-best; it achieved optimization but required over 500 epochs to achieve the same accuracy as Tanh or ReLU achieved in 50. Softmax was the worst-performing, as the model overcorrected quickly and did not continue to optimize with continuing epochs, showing nominal improvement with each iteration.

Loss Function

The binary classification loss functions are best for this problem. Hinge and Squared Hinge minimizes the amount of loss.

Accuracy Function

MAE: Having a higher number of epochs gave less error, which is important to note. Although accuracy is generally higher with 5 epochs, 50 epochs are better with error optimization. Having a fewer number of input neurons (less than 100), but input neurons still have to be larger than hidden neurons.
RMSE: There can be more input neurons than in MAE, but still has to be around 100 to 500. The SGD optimizer and short epochs have the lowest errors.

Number of Epochs

5 and 10 epochs performed the best, as it would have a lower run-time. It works well in combination with the Tanh and ReLU activation functions and SGD and Adam optimizers.

Final Results

When testing the parameters, it was essential to ensure that these adjustments do not cause overfitting in the model. While a parameter can have a significant impact on increasing accuracy, this can cause issues when using test data or test data, not from this respective set.

After running over 40 different tests with modified parameters, the neural network design presented was the most optimized. However, with additional testing in time, human resources, further refinement could be made.

Additional adjustments do run the risk of overfitting the data. As data scientists / AI developers, it is their responsibility as humans to determine when the network suffices in providing optimal outputs.

When rerunning models, there were variances, inaccuracies and performance metrics with each run. These are likely due to the random sampling of the dataset. Models are run several times to find an ‘average’ performance rate. As well, specific performance measures worked more consistently than others.

Thus, this shows that even if there is a neural network with well-performing accuracy or other metrics, the loss can still be high.

As well with parameters, when running the data through the network, different measurements were recorded. Due to the data being randomly split. This means the parameters can be modified within a margin of performance and discretion of the network designer to achieve a similarly high performing network classification.

Accuracy was the most valuable performance metric as it simply assesses how accurate the network is in classifying the data.

Other metrics were helpful to give supporting evidence and monitor how the parameters impacted the network’s ability to classify the data. It is difficult to optimize a classification model for other performance metrics such as Recall, Precision, F1 as their optimization comes at the cost of others. Instead, these metrics are monitored for comparative analysis.

Conclusion

Based on the research conducted and the actual model created, it was determined which parameters have a strong impact on the accuracy of training a neural network.

Number of neurons in the layers (hidden/input), number of layers, optimizer, Activation function, epochs. Adjusting these parameters had the most significant impact on the test loss.

It is critical that parameters be measured beyond accuracy and ranked performance measurements are incorporated. Defining the critical performance metric and then supporting it with secondary performance measurements.

Parameters have a high degree of variance, meaning there is no one perfect set of parameters for this data set. Parameters can be used in variance, and they will give a similar degree of accuracy as the model used in this report.

Improvements for Future Classification Models

Additional Data exploration can be done with more software to calculate the correlation and IG for all features in the data set. Following the data could be cleaned further for outliers.

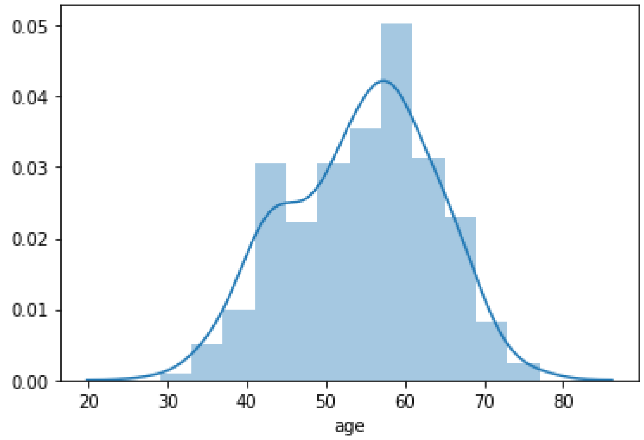
AUC (area under the curve) and ROC (receiving operating characteristics) as a metric and the score to understand what parameters.

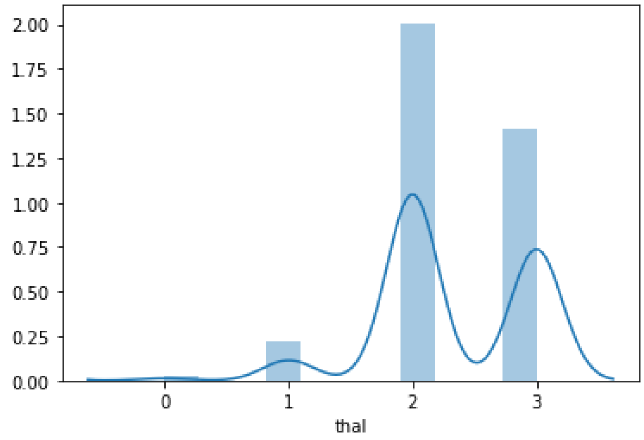
Various parameters like dropout ratio, regularization weight penalties, learning rate, early stopping can be changed while training neural network models.

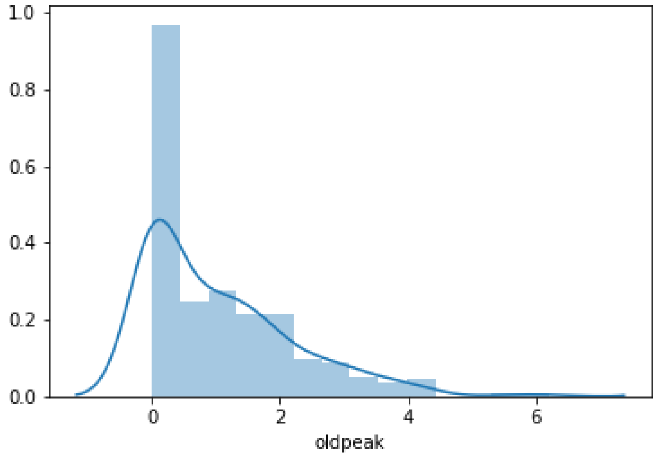
The model (overfitted) could be improved by adding regularization methods. Although it is not the scope of this article, further research showed that using an adjusted support vector machine has a higher performance than a regular neural network.

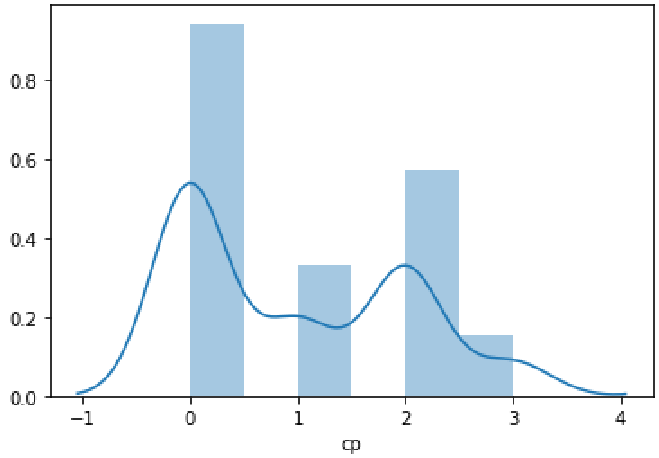
Appendix

Bar Charts (Appendix 1)

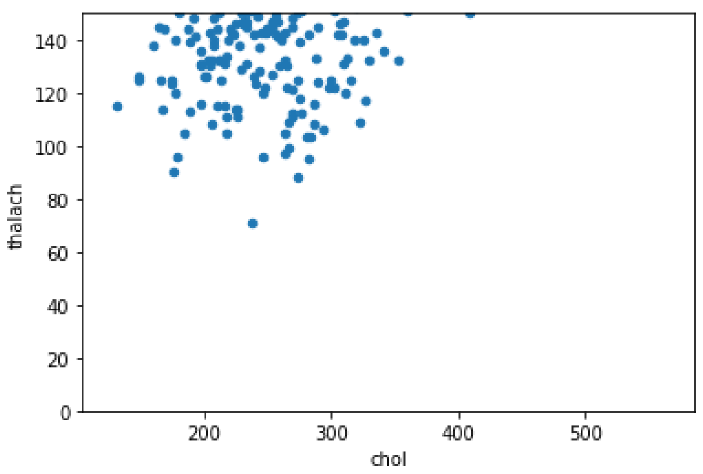


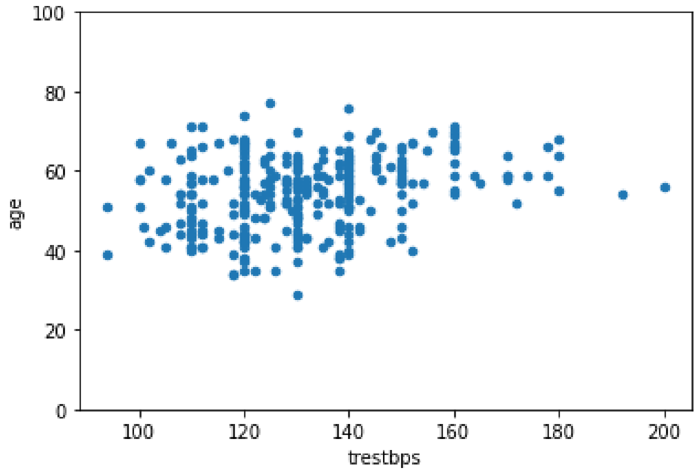




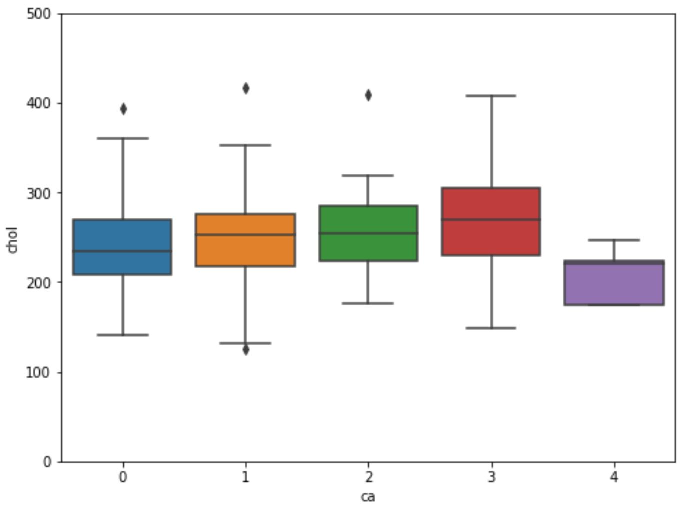


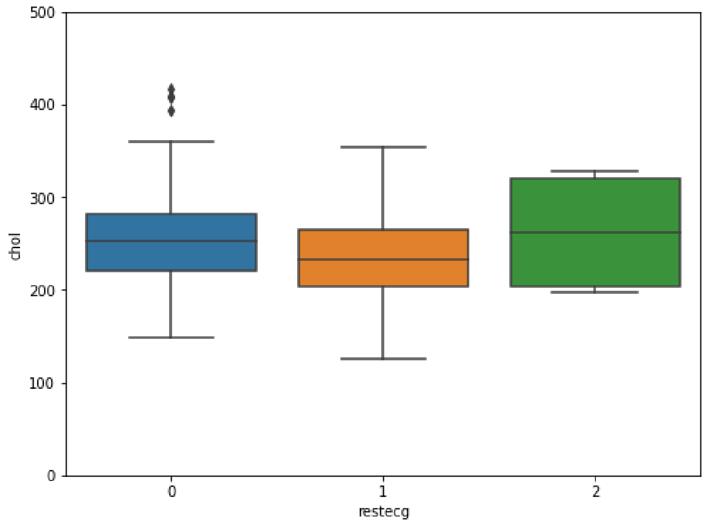
Scatter Plots (Appendix 2)

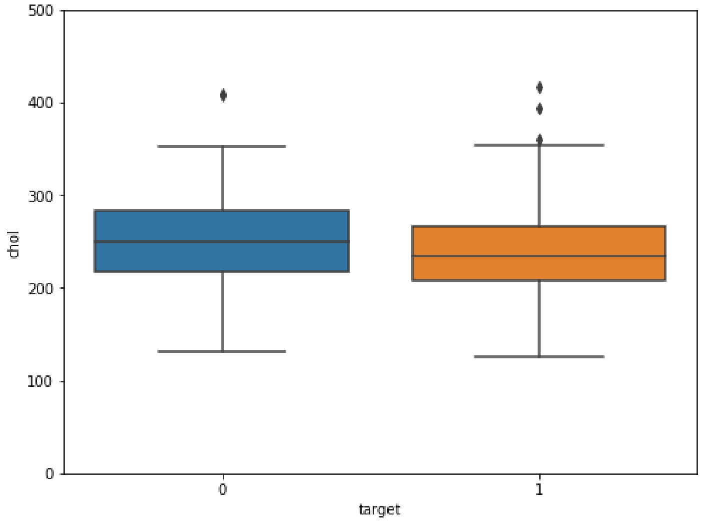




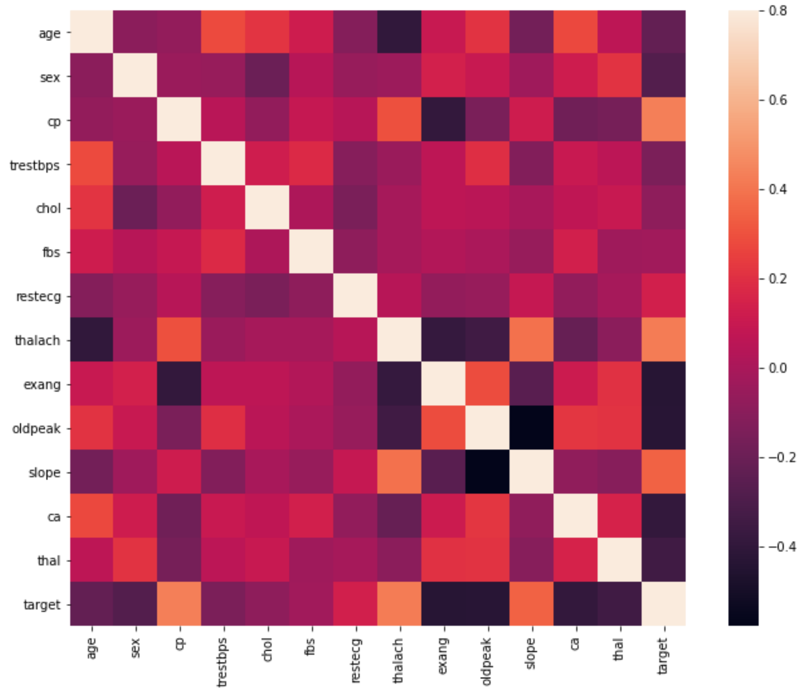
Box plots (Appendix 3)



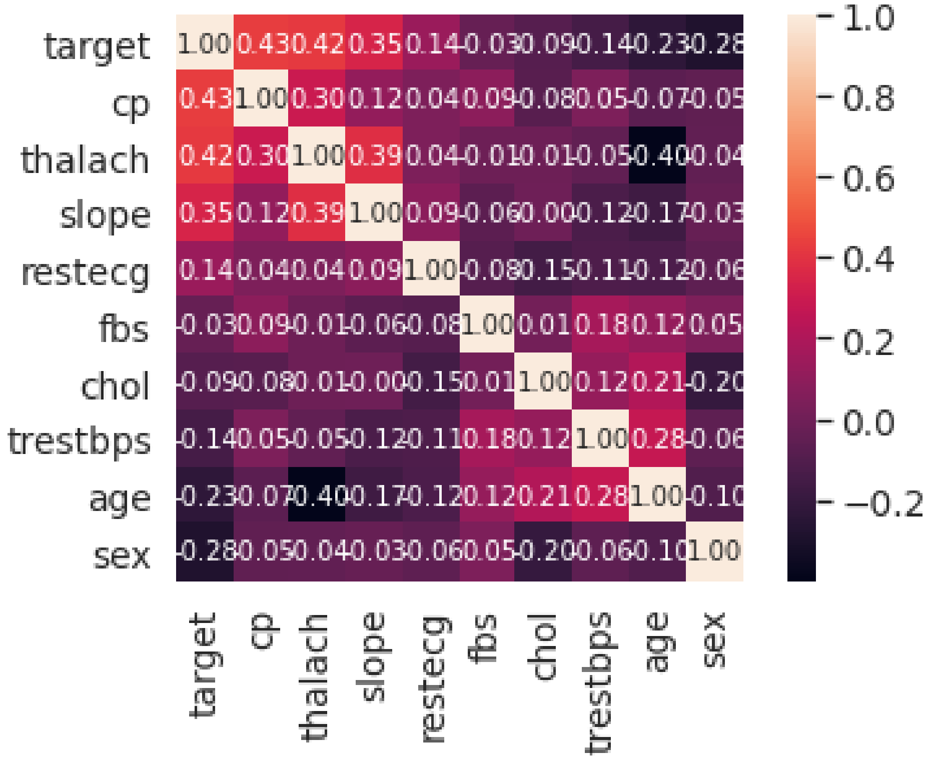




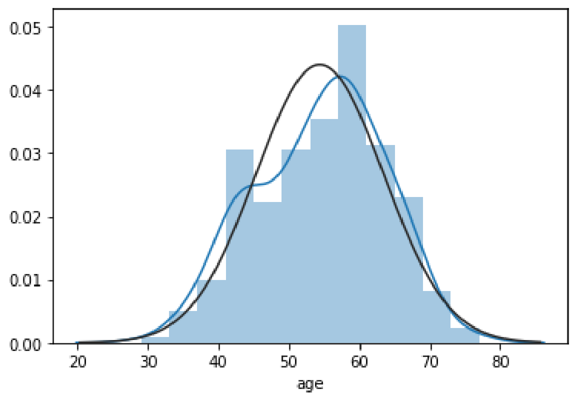
Heat Mapping (Appendix 4)

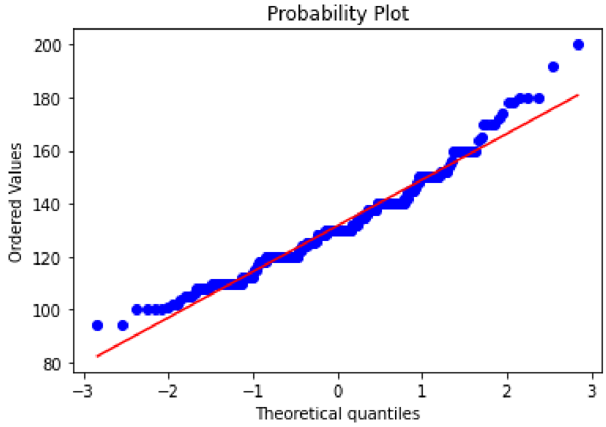


Target Mapping (Appendix 5)

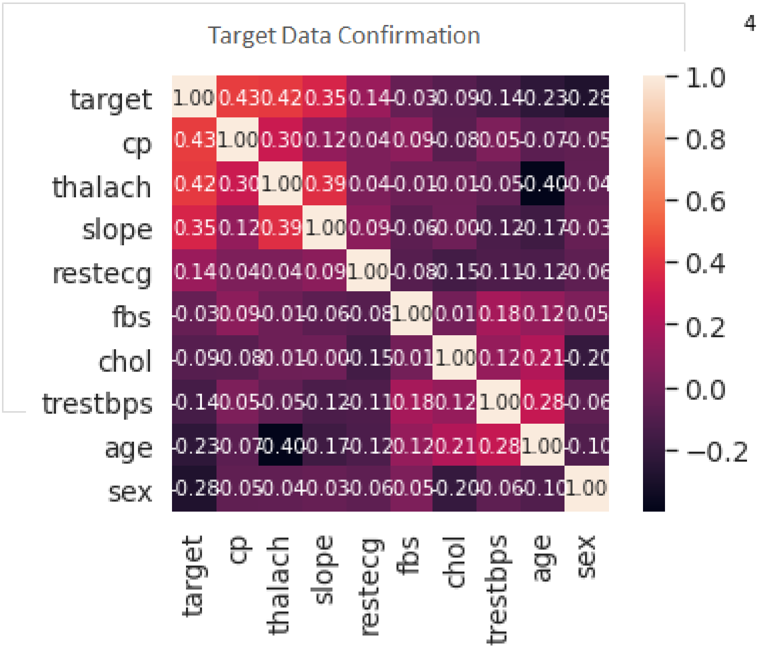


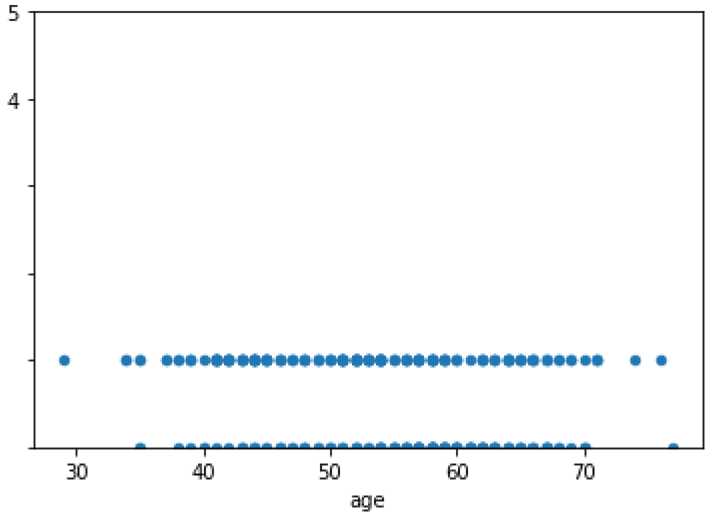
Probability Analysis (Appendix 6)

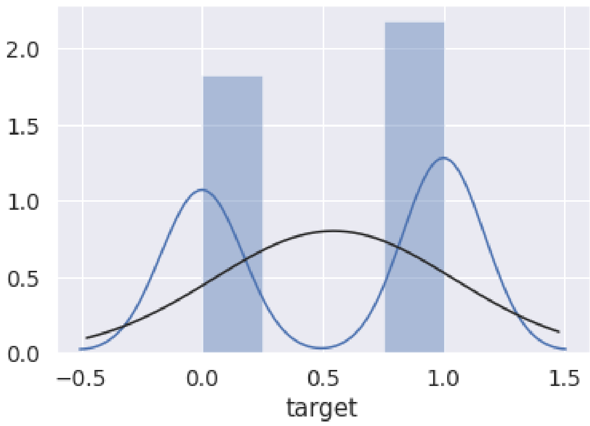




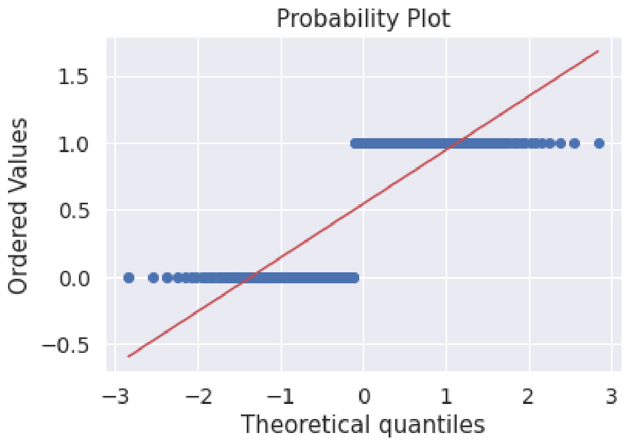
Target Data Analysis (Appendix 7)







Shareable Links:
[Final Code] (dataset is shareable)
[Additional Code](Can't be run)
[Research & Code Experimentation]
[Data Exploration]



References

Story, C. (2020, March 29). What's the Connection Between High Blood Cholesterol and Hypertension? Retrieved April 5, 2020, from https://www.healthline.com/health/high-cholesterol/treating-with-statins/hypertension#1

Solan, M. (2019, February 6). Your resting heart rate can reflect your current - and future - health. Retrieved April 6, 2020, from https://www.health.harvard.edu/blog/resting-heart-rate-can-reflect-current-future-health-201606179806)

Goldberger, A. (2019, March 12). Training Neural Networks for binary classification: Identifying types of breast cancer (Keras in R). Retrieved from https://medium.com/duke-ai-society-blog/training-neural-networks-for-binary-classification-identifying-types-of-breast-cancer-keras-in-r-b38fb26a500c

JJ. (2016, March 26) MAE and RMSE — Which Metric is Better?: Mean Absolute Error versus Root Mean Squared Error. Retrieved from: https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d