JOURNAL OF
EMERGING INVESTIGATORS

Article

Study of neural network parameters in detecting heart disease

Maxim Sean Malkevich[1], Violeta Prisacari[1]
[1] Newton North High School, Newton, Massachusetts

SUMMARY

Heart disease is a major health problem worldwide, and in the United States it causes over half a million deaths per year. Machine learning algorithms are being used more widely in medicine, leading to the possibility of using machine learning to identify heart disease with greater accuracy before it presents. Machine learning would also allow for more patients to be screened, helping patients and doctors make an informed decision before any damage is done. We examined the accuracy of different machine learning models in predicting heart disease from a set of 301 patients. Specifically, we aimed to identify the optimal parameters for the accurate detection of heart disease. We hypothesized that increasing the value of the L2-regularization constant and the number of layers would increase accuracy; however, our results showed lower values of the L2-regularization constant for error regularization and lower numbers of rectified linear unit (ReLU) layers to be beneficial for the function of the model. The average accuracy after 600 epochs of models without error regularization was 73.75%, while the average final accuracy for models that utilized any non-zero value of the L2-regularization constant was consistently less than or equal to 66.02%. Of all the models studied, only two had a higher accuracy (86.64%) than a logistic regression (85.53%). As such, we found that error regularization and large models may be poorly suited for detecting heart disease, and models with one or no hidden layers may perform better than models with greater amounts of ReLU nodes. This may be significant when designing fast, accurate, and life-saving diagnostic models.

INTRODUCTION

The leading cause of death worldwide is ischemic heart disease, which caused 16% of all deaths in 2019 (1). Within the United States, 695,000 deaths were caused by any heart disease, 375,476 of which were from ischemic heart disease (2). Even as the number of deaths per 100,000 has been trending downwards for more than 50 years, heart disease remains the top cause of death in the United States (3). However, heart disease is often not recognized by patients and medical professionals until it is too late (4). For example, between 16.1% and 68.5% of heart failure cases are misdiagnosed as chronic obstructive pulmonary disease (4). Without heart disease being detected, the individual is left at greater risk of irreversible damage. As such, the ability to accurately determine whether an individual is at risk of being affected by heart disease could help reduce mortality by alerting physicians earlier.

Artificial intelligence (AI) systems are becoming more relevant in the medical field due to their ability to perform tasks normally requiring human capabilities, such as decision-making or problem-solving, and machine learning has been put to use for tasks ranging from the prediction of post-stroke pneumonia to robotically-assisted surgeries (5). Diagnostic models are able to achieve increased speed and accuracy and thus could help doctors provide more effective treatment at the right time (5). These advancements enable the development of potentially life-saving models capable of alerting patients and doctors to probable heart disease. Accurate diagnostic tools could help decrease the number of heart disease cases and deaths year by year by helping identify risk while action can still be taken. However, the exact parameters of a helpful model for this purpose are not easily intuited, due to both the large number of relevant factors in a machine learning model and the uncertainty of how they will work together to affect a model's performance. As such, our goal was to identify the parameters of the most accurate neural network for the diagnosis of heart disease.

In the present study, we investigated how regularization and the depth of a multilayer perceptron (MLP) model affected its accuracy. MLP models are artificial neural networks consisting of several layers of nodes, which are the fundamental units of machine learning models that take inputs and process them to produce outputs for other nodes, with the outputs of each layer providing the inputs for the next. The depth of a model is its number of hidden layers (all layers except for the input and output layers), and regularization of an MLP adds a value proportional to the sum of the squares of the weights (coefficients that the outputs of one node are multiplied by to make them into inputs for the next node) used by the model to the loss (which is a measure of the difference between the model's predictions and the actual data, which is minimized by the MLP by updating its weights) so as to prevent the weights from growing to a large value and prevent overfitting. We evaluated 240 models on a dataset of 301 patients with 13 hypothesized predicting factors for heart disease, including resting blood pressure, cholesterol, fasting blood sugar, chest pain type, and maximum achieved heart rate.

We initially hypothesized that increasing regularization and increasing the total number of layers would enhance model accuracy. Our rationales were that increasing the regularization constant value would help protect against overfitting and adding layers would make for a more complex model that can potentially detect more patterns useful to

https://doi.org/10.59720/24-281

making a final decision. However, in the present study, we found that higher values of the regularization constant and high amounts of layers may be detrimental to model accuracy. As such, less complex models might be better suited for the detection of heart disease when trained on the dataset used in the present study.

RESULTS

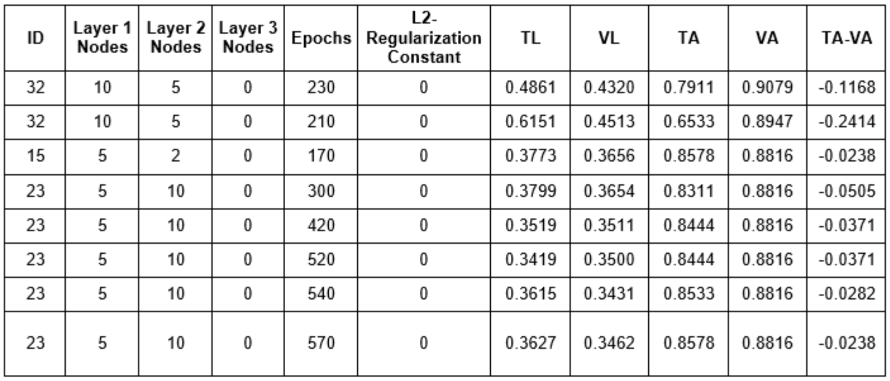
To examine which model parameters would lead to increased accuracy and better performance and determine whether increasing regularization and layer count would enhance model accuracy in detecting heart disease, we trained a large amount of MLP models with varying parameters. We generated 240 different models based on a set of possible parameters for hidden (non-input and non-output) layer width (number of nodes in a layer), depth (number of hidden layers: from 0 to 3, inclusive), and values of the L2-regularization constant (six equally spaced values from 0 to 0.25, inclusive), and tracked their performance for 600 epochs (i.e., 600 cycles of training models on the data) by recording their validation accuracy. By examining the model performance with data it was not trained on and analyzing the potential impacts of each different parameter on the results, we could determine what would help or harm the model performance.

Model #32, which has 10 nodes in the first hidden layer, 5 nodes in the second, and no error regularization, gave the highest achieved accuracy with the validation data. After 230 epochs of training, the model achieved a validation accuracy of 0.9079 (**Table 1**). However, we hesitated to designate this model as the best model, for the following two reasons: first, after 220 and 240 epochs of training, this model had a validation accuracy of 0.5658 and 0.75, respectively, suggesting that training this model with the same parameters for a similar amount of epochs will give inconsistent results. Second, the training accuracy for 230 epochs is 0.7911, which is lower than the validation accuracy by 0.1168. This discrepancy may suggest that the model did not learn from the training data and that its performance with the validation data was coincidental.

The next highest validation accuracy (0.8816) comes from model #15 at 170 epochs, which has 5 nodes in the first hidden layer and 2 in the second, with no error regularization. This model has a training accuracy of 0.8578, which is much closer to the validation accuracy of the model than model #32's training accuracy was to its validation accuracy (**Table 1**). We considered any difference between the validation and training accuracy of a model greater than 0.03 to be a large enough difference to indicate that the model likely either overfitted or did not learn sufficiently from the training data. Since model #15's training and validation accuracies are within 0.03 of each other, we considered that the difference was not large enough to warrant these concerns. However, one problem remains, as the validation accuracies at 160 and 180 epochs are 0.75 and 0.4605, respectively. Model #23, which has 5 nodes in the first hidden layer, 10 in the second, and no error regularization, achieves the same validation accuracy as model #15 after 300, 420, 520, 540, and 570 epochs, while the training accuracy in each of these epochs is within 0.051 of the validation accuracy (**Table 1**). Except for epochs 310 and 580, the validation accuracy does not change drastically and remains not far away from the high validation accuracy of 0.8816 at 0.8553 by the end of 600 epochs.

We found the models with the highest validation accuracies after the full 600 epochs of training out of those that had their validation and training accuracies within 0.03 of each other. Since validation accuracy is a measure of the model's ability to accurately predict data that it was not trained on, a high validation accuracy was used as an indicator of the model's overall suitability for the problem. We restricted the models we looked at to those with validation and training accuracies within range of each other for reasons elaborated on earlier, and we looked at the results after the full 600 epochs since these models would be the most trained. We found the best-performing models trained after 600 epochs to be model #1 (2 nodes in its single hidden layer and no error regularization) and model #37 (10 nodes in the first and second hidden layers, 2 in the third, and no error regularization), with each one having a validation accuracy of 0.8684 (**Table 2**). The next best performing models were model #0, which was a logistic regression; model #17, with hidden layers of width 5, 2, and 5; model #23, with widths 5 and 10; and model #33, with widths 10, 5, and 2. All of these used no error regularization, and they all achieved a final validation accuracy of 0.8553 (**Table 2**).

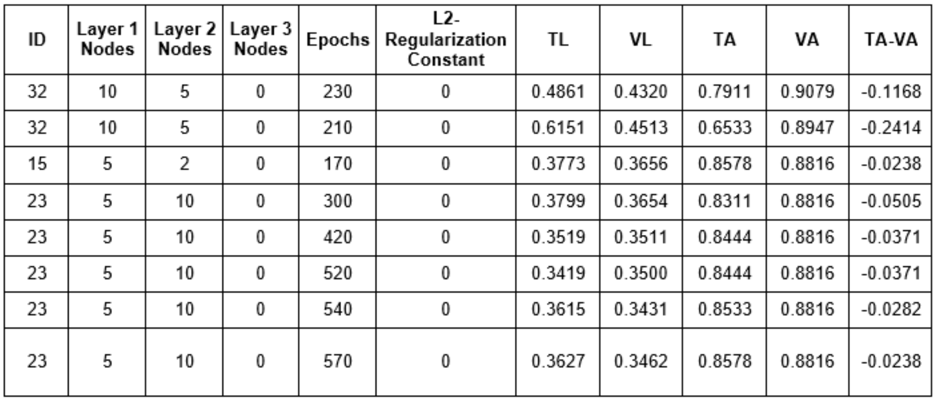
Lower values of the L2-regularization constant resulted in more accurate models (**Figure 1**). In no model does



ID	Layer 1 Nodes	Layer 2 Nodes	Layer 3 Nodes	Epochs	L2-Regularization Constant	TL	VL	TA	VA	TA-VA
32	10	5	0	230	0	0.4861	0.4320	0.7911	0.9079	-0.1168
32	10	5	0	210	0	0.6151	0.4513	0.6533	0.8947	-0.2414
15	5	2	0	170	0	0.3773	0.3656	0.8578	0.8816	-0.0238
23	5	10	0	300	0	0.3799	0.3654	0.8311	0.8816	-0.0505
23	5	10	0	420	0	0.3519	0.3511	0.8444	0.8816	-0.0371
23	5	10	0	520	0	0.3419	0.3500	0.8444	0.8816	-0.0371
23	5	10	0	540	0	0.3615	0.3431	0.8533	0.8816	-0.0282
23	5	10	0	570	0	0.3627	0.3462	0.8578	0.8816	-0.0238

Table 1. Models with highest validation accuracies. Top eight rows by validation data accuracy showing the model ID, the number of nodes in each layer, the epoch at which data was recorded, and the model's value of the L2-regularization constant. Training loss (TL), validation loss (VL), training accuracy (TA), validation accuracy (VA), and the difference in training accuracy and validation accuracy (TA-VA) are also given. Data taken from performance of automatically generated models.

https://doi.org/10.59720/24-281



ID	Layer 1 Nodes	Layer 2 Nodes	Layer 3 Nodes	Epochs	L2-Regularization Constant	TL	VL	TA	VA	TA-VA
32	10	5	0	230	0	0.4861	0.4320	0.7911	0.9079	-0.1168
32	10	5	0	210	0	0.6151	0.4513	0.6533	0.8947	-0.2414
15	5	2	0	170	0	0.3773	0.3656	0.8578	0.8816	-0.0238
23	5	10	0	300	0	0.3799	0.3654	0.8311	0.8816	-0.0505
23	5	10	0	420	0	0.3519	0.3511	0.8444	0.8816	-0.0371
23	5	10	0	520	0	0.3419	0.3500	0.8444	0.8816	-0.0371
23	5	10	0	540	0	0.3615	0.3431	0.8533	0.8816	-0.0282
23	5	10	0	570	0	0.3627	0.3462	0.8578	0.8816	-0.0238

Table 2. Models with highest final validation accuracies. Table of top eight rows by validation data accuracy at 600 epochs, showing the model ID, the number of nodes in each layer, and the model's L2-regularization constant, along with the training loss (TL), validation loss (VL), training accuracy (TA), validation accuracy (VA), and the difference in training accuracy and validation accuracy (TA-VA). Data taken from performance of automatically generated models.

the average accuracy increase from its value when the regularization constant is 0 and drops of up to 30 percentage points occur as the value of the regularization constant increases (**Table 3**). These results were observed to hold for all depths studied, from zero (for logistic regressions) to three. Furthermore, a lower depth led to higher mean and median final validation accuracies. Logistic regressions had the highest mean (0.7610) and median (0.8092) validation accuracies after 600 epochs (**Table 4**). They are followed by models with two hidden layers (with a mean accuracy of $\bar{x}$=0.6513 and median accuracy of M=0.6447), one hidden layer ($\bar{x}$=0.6374, M=0.6184), and three hidden layers ($\bar{x}$=0.6320, M=0.5789) (**Table 4**). However, the most accurate models had one (model #1) or three hidden layers (model #37), with a final validation accuracy of 0.8684 (**Table 4**). This is slightly more than the accuracy of 0.8553 for logistic regressions and networks with two hidden layers (**Table 4**).

DISCUSSION

We hypothesized that error regularization and increased layer count would enhance model performance when detecting heart disease. We reasoned that the former would prevent overfitting, and the latter could allow for more complex models due to the greater number of nodes connected to each other, both of which we believed would make for a more accurate model. However, on average, error regularization decreased the final accuracy of the model. The highest validation accuracy regardless of model size was for the group of models where the value of the L2-regularization constant was equal to zero, and none of the eight most accurate models used error regularization (**Figure 1**, **Table 2**). Error regularization adjusts the loss calculation (which was made using binary crossentropy) by adding **Equation 1** to the loss.

$$\frac{\lambda}{n}\Sigma\left(w_{ij}^{[l]}\right)^2 \qquad \textbf{(Eqn. 1)}$$

λ is the L2-regularization constant, *n* is the number of examples, and *w* is the weight that the output of the i^{th} node of the *l-1th* layer is multiplied by to get an input for the *jth* node of the *lth* hidden layer. This combats overfitting by preventing the model's weights from growing excessively, resulting in smaller weights that are less sensitive to minor deviations in the data. However, in this case, adding to the error based on the size of the weights has likely placed undue focus on keeping the weights manageable rather than making them appropriate to the data, thus forcing the model to underfit. Keeping the relatively small size of the dataset in mind, it is possible that this alone would have already made neural networks less likely to form trends from the data, before adding this enforced underfitting.

Furthermore, smaller models tended to perform more accurately on the data. The average and median of the final accuracies of logistic regressions with any value of the L2-regularization constant were greater than those of neural networks with any depth other than zero (**Table 4**). This may be because of models with a greater number depth being too complicated for the dataset used, and thus being more likely to overfit. Other contributing factors may be the learning rate (which was set at the default value of 0.001) being too high, preventing the model from reaching the optimum parameters, or the use of the activation function ReLU which zeroes intermediate values and may have hindered the model's ability to learn from them.

Limiting factors that may have influenced the results of the present study include the range of values of the L2-regularization constant and depths. It is possible that a greater range of possible values for both may have allowed for a more complete presentation of how exactly these factors influence model performance. As such, future experiments may focus on examining models constructed from a wider range of depths and values of the L2-regularization constant. Furthermore, another potential factor of influence is the dataset itself. The data used here consisted of 301 non-null entries from a dataset of 303 patients, which may have limited the extent to which models could learn from the data, since larger datasets may be more representative of the general population and show patterns, which allow for easier determination of whether or not the patient has heart disease. This dataset was sourced from patients undergoing angiography at the Cleveland Clinic in Cleveland, Ohio, which may make it even more beneficial to collect data from a wider range of hospitals and geographical regions to avoid overrepresenting a particular location or set of environmental conditions. More replicates of the experiment could also be

https://doi.org/10.59720/24-281

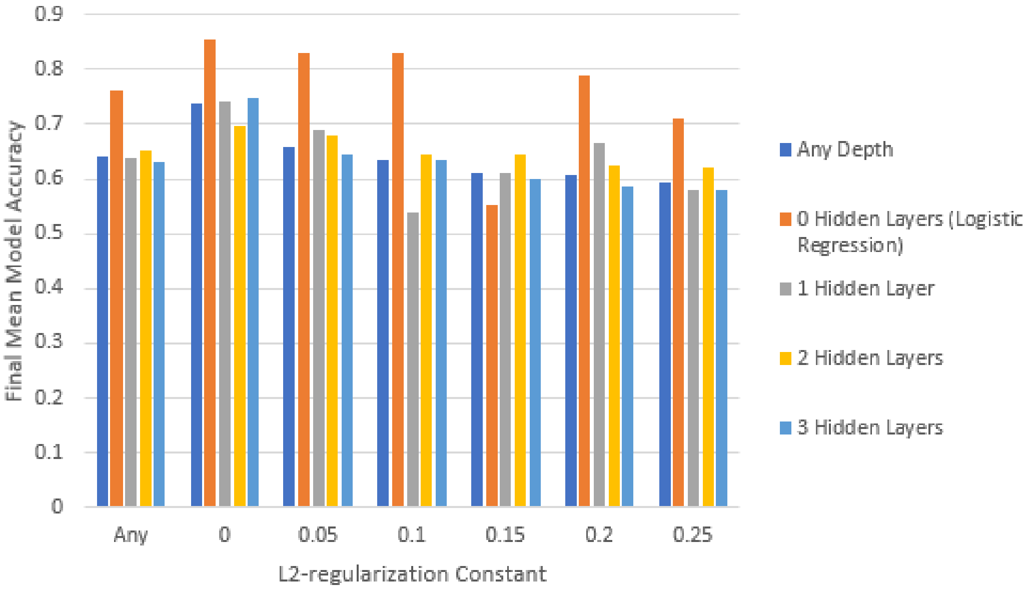


Figure 1. Final model accuracy by L2-regularization constant value. Mean model validation accuracy at 600 epochs, grouped together by the value of the L2-regularization constant for models with any depth (dark blue, n = 240), zero hidden layers (logistic regression, orange, n = 6), one hidden layer (gray, n=18), models with two hidden layers (yellow, n=54), and models with three hidden layers (light blue, n=162). Data taken from performance of automatically generated models.

done by generating the models with the same parameters and determining whether the training process has led to the same results as before or whether there is some variation in model performance.

Further subjects of study may be whether the activation functions used by the model impact its final performance, how the order in which different functions are used affects accuracy, and whether networks of constant, increasing, or decreasing widths are more accurate with these data. One of our criteria for accurate models (that the validation accuracy and the training accuracy had to be within 0.03 of each other) may also be important to consider for future experiments. While we trained each model once and compared validation data against training data once, we would expect that if the model were indeed learning from the data, it would have a high training accuracy. A high training accuracy indicates that the model learned from the data it was trained on, and a high validation accuracy indicates that the patterns it learned are not overly specific to the data it was trained on but rather can be applied to other data it will encounter. Thus, if the training accuracy and validation accuracy were very different from each other, the model was not considered to have performed well; either the training accuracy was high and the validation accuracy was low, in which case the model overfit and could not reliably classify data outside the set it was trained on, or the training accuracy was low and the validation accuracy was high, in which case it would not make sense to say that the patterns that the model learned could accurately classify data outside the training set because the model did not learn patterns from the training data. Future experiments could focus on how the difference between the validation and training accuracy of a model affects the model's training and validation accuracy after further training.

Overall, the results suggest that smaller models with lower values of the L2-regularization constant may be better suited for predicting the presence of heart disease. Simpler models such as logistic regressions may thus be helpful in making the determination whether or not a patient has heart disease. Considering the potential for machine learning models to help make faster and more accurate diagnoses, understanding what parameters result in a model well-suited for detecting heart disease would be highly beneficial. If more

		Value of the L2-regularization Constant						
		0	0.05	0.10	0.15	0.20	0.25	Any
Depth	0	0.8553	0.8289	0.8289	0.5526	0.7895	0.7105	0.7610
	1	0.7412	0.6886	0.5395	0.6096	0.6667	0.5789	0.6374
	2	0.6959	0.6798	0.6433	0.6433	0.6243	0.6213	0.6513
	3	0.7466	0.6442	0.6360	0.5994	0.5867	0.5789	0.6320
	Any	0.7375	0.6602	0.6352	0.6089	0.6063	0.5918	0.6400

Table 3. Final model accuracies by L2-regularization constant and depth. Mean model accuracy at 600 epochs, grouped by depth and L2-regularization constant value. Data taken from performance of automatically generated models.

		Model Depth			
		0*	1	2	3
Validation Accuracy	Mean	0.761	0.6374	0.6513	0.632
	Median	0.8092	0.6184	0.6447	0.5789
	Maximum	0.8553	0.8684	0.8553	0.8684

Table 4. Final training model mean, median, and maximum accuracies by depth. Table of mean, median model accuracy at 600 epochs, grouped by depth. Data taken from performance of automatically generated models.
*Represents a logistic regression model

https://doi.org/10.59720/24-281

cases are identified early and fewer are left undiagnosed or misdiagnosed, patients might be able to receive effective treatment earlier and there could be a reduction in mortality from heart disease.

MATERIALS AND METHODS

Dataset

The dataset used to train and test the models was sourced from the UC Irvine Machine Learning Repository and was gathered by Andras Janosi, William Streinbrunn, Matthias Pfisterer, and Robert Detrano. It contains 303 entries of patient data from the Cleveland Clinic in Cleveland, Ohio that included symptoms, vital measurements, and the presence or absence of heart disease (6).

Code

A program was written to process the dataset, generate the models used in the present study, train them, and analyze their performance. Please find the code used in the present study at the following link: https://github.com/malkevich-maxim/neural-net-heart-disease

Data processing and training-validation split

One-hot encoding, which is the conversion of categorical data into multiple columns of binary categorical data, was performed upon the "cp" (chest pain type), "rest_ecg" (resting cardiograph abnormalities), "slp" (ST (slope type) segment (the usually flat portion of an ECG that follows the QRS complex and indicates ventricle repolarization)), and "thall" (thallium stress test results) data columns, creating three different features for each one other than "cp", from which four features were created. This was done so that the categories would not be treated like numerical data; for instance, without one-hot encoding, the model could treat the values of 0, 1, 2, and 3 for chest pain as numbers which are higher or lower than each other, whereas they really just represent different types of chest pain with no way of ranking them hierarchically (and even if it were a case of "no pain," "mild pain," "severe pain," etc, there still would not be a "correct" numerical scale those could be placed on). Data entries with null values were then removed (which applied for two patients). Afterwards, the data were split between validation (n=76) and training data (n=225), both of which were scaled. The training data were scaled by subtracting the mean values for each column and dividing by the standard deviation to obtain a mean of 0 and a standard deviation of 1, and the validation data were scaled by subtracting the mean of the training data and dividing by the standard deviation of the training data.

Model generation and training

Four nested loops were then used to automatically generate models; the first cycled through six different values for the L2-regularization constant (0, 0.05, 0.1, 0.15, 0.2, and 0.25) while the second through fourth cycled through hidden layer widths (0, 2, 5, and 10). If the resulting model was not invalid (due to a hidden layer having a nonzero width when a prior layer had zero width and was thus absent), it would then be created with the specified hidden layers and an output layer of one node using the sigmoid function, compiled, and trained with the data. All nodes except for the output node (which used the sigmoid function) used ReLU as the activation function.

Model loss minimization

The loss for each model was calculated with **Equation 2**, the binary crossentropy function.

$$J = -\frac{1}{n}\sum(y_i \ln(\hat{y}_i) + (1-y_i)\ln(1-\hat{y}_i)) \qquad \textbf{(Eqn. 2)}$$

J is the error, *n* is the number of examples, y_i is the real output value for the *ith* example, and *ŷi* is the predicted output value for the *ith* example. Loss was minimized by making the model's predictions closer to the actual outputs since the *ith* term in the summation goes to 0 as *ŷi* approaches the value of y_i and goes to infinity as it goes further away from the real value.

Model tracking

For each of the 240 models created, loss and accuracy for both training and validation sets were recorded from every 10 epochs up until 600, along with a unique model ID for ease of reference and the results from the first epoch for ease of comparison.

Received: August 28, 2024
Accepted: December 19, 2024
Published: September 7, 2025

REFERENCES

1. "The top 10 causes of death." *World Health Organization*, 9 Dec. 2020, www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death. Accessed 13 Aug. 2024.
2. "Heart Disease Facts." *Center for Disease Control*, 15 May 2023, www.cdc.gov/heartdisease/facts.htm. Accessed 13 Aug. 2024.
3. "U.S. Trends in Heart Disease, Cancer, and Stroke." *Population Reference Bureau*, 1 Dec. 2002, www.prb.org/resources/u-s-trends-in-heart-disease-cancer-and-stroke/. Accessed 13 Aug. 2024.
4. Wong, Chun Wai et al. "Misdiagnosis of Heart Failure: A Systematic Review of the Literature." *Journal of Cardiac Failure*, vol. 27, no. 9, 25 May 2021, pp. 925-933. https://doi.org/10.1016/j.cardfail.2021.05.014.
5. Habehh, Hafsa, and Suril Gohel. "Machine Learning in Healthcare." *Current Genomics* vol. 22,4 (2021): 291-300. https://doi.org/10.2174/1389202922666210705124359.
6. Janosi, Andras et al. "Heart Disease." *UCI Machine Learning Repository*. https://doi.org/10.24432/C52P4X.