Zachary Weinfeld and Frank Assumma

July 28 2023

# CSC 466 Project Report

We are interested in predicting a patient's risk of having a stroke. A stroke is a medical emergency characterized by a lack of blood supply to parts of the brain, leading to brain damage or cell death[1]. According to the National Institute of Health, someone who has had a stroke can become paralyzed on one side of their body, experience muscle weakness, have trouble speaking, and more[2]. In severe cases, strokes can be lethal. We are interested in applying various machine learning methods to build a model to predict stroke risk. Such models could be used to potentially save lives by identifying members of society who are susceptible to having a stroke and taking preventative measures before a stroke occurs. We will implement two separate models from scratch, and then implement them through SciKit Learn's library. We are interested in comparing the performance of the paired models, in terms of efficiency and accuracy. Our code is stored in this repository[3].

The dataset[4] we utilized contains 5,110 subjects with ten different potential stroke indicators and one binary column describing whether the subject has had a stroke or not. Among the indicators were gender, age, hypertension status, work type, average glucose level, heart disease status, and Body Mass Index (BMI). Around 4% of the observations did not have a BMI value present, so this data was not included in the prediction. Additionally, 30% of the observations had "unknown" smoking status. We made the choice to drop observations with missing data. A few of the columns were binary and, thus, simple numerical conversions while others, like smoking status, were a spectrum that was converted to a 0 to *n* score, where *n* is the number of possible responses to the inquiry. Another column, work type, was not ordinal and

[1] https://www.cdc.gov/stroke/about.htm
[2] https://www.nia.nih.gov/health/stroke
[3] https://github.com/Zweinfeld001/CSC466Project
[4] https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset

therefore presented an undesirable trait in our analysis, leading us to drop it from our models. Ultimately, this left us with 3,026 individuals, nine different potential stroke indicators to analyze, and one binary column denoting our target variable of stroke presence.

Two different models were used to train and predict stroke occurrence. These were decision trees and neural networks. Under each, we compared our own implementation of the model with Python's SciKit Learn library implementation. In the decision tree models, a numeric decision tree was utilized with entropy as the criteria, a test set size of 20% of the dataset, a training set size of 80% of the dataset, and a minimum split count of five. Gain and gain ratios were calculated for use in generating the rules of the decision tree and producing predictions based upon specified rules.

In the neural network models, the preprocessed data was converted into z-scores of the original data using SciKit Learn's Standard Scaler. The neural network implemented by our team was a single-layer neural network opposed with the 100-layer network designed using SciKit Learn, each with the logistic sigmoid activation function. In both cases, 200 epochs were utilized with a test set size of 30% of the dataset and a training set size of 70% of the dataset. Our team's neural network utilized a fixed learning rate of 0.5 while the SciKit Learn neural network involved a more complex optimizer that involved inverse scaling the learning rate with each epoch.

In our analysis, we will utilize the F1 score as the measure of efficacy. Our decision tree predicted stroke absence with an F1 score of 96.3% and stroke presence with an F1 score of 7.5%, highlighted by a significant amount of false negative results. SciKit Learn's decision tree performed worse when predicting stroke absence with an F1 score of 94.5%, but performed significantly better when predicting stroke occurrence with an F1 score of 22.2%. Ultimately, the

SciKit Learn implementation had a tendency to predict more overall stroke occurrences than the team's implementation (46 to 9 on average).

Our neural network predicted stroke absence with an F1 score of 84.4% and stroke presence with an F1 score of 2.1% with a significantly large amount of false positive results at 155. The SciKit Learn neural network was more accurate, predicting stroke absence with an F1 score of 96.0% and stroke presence with an F1 score of 7.0%. False positive and false negative counts were comparable in the SciKit Learn model at around 25 in the testing data. Overall, the efficacy of the SciKit Learn model tested higher while performing more efficiently and effectively than our team's implementations, though our implementations were still comparable.

From our data, we will ask and investigate three specific questions. First, we will investigate which variables have the highest correlation with having a stroke. Second, we are interested in analyzing the effect of a subject having missing data, as many subjects don't have instances for BMI and smoking status. Finally, we want to determine if any natural clusters form in our data.

To investigate our specific aims, we took a variety of approaches. To answer our first question, we found the correlation between each variable with the target variable. We did this by creating a correlation matrix and examining the "stroke" column. We found that "age" has the highest correlation (0.23) with "stroke," followed by "hypertension" (0.14). While these correlations are relatively weak, they are still informative, since we would expect a correlation of nearly zero if the variables were truly independent. In addition to the positive correlations, "ever_married" had a weak negative correlation (-0.11) with "stroke." The remaining variables had a negligible correlation with "stroke."

Zachary Weinfeld and Frank Assumma

July 28 2023

Our second question involved analyzing subjects with missing data. When creating our neural network and decision tree, we simply dropped observations with missing data. However, further investigation has shed light onto the fact that this may have introduced bias to our model. In the entire original dataset of 5,110 subjects, only 4.9% were classified as having a stroke. We examined the 201 subjects who had missing values for BMI and found that in this group, 20.0% were classified as having a stroke, significantly juxtaposing the expected percentage of 4.9%. This could be explained by a systematic difference in ability to get BMI measured with susceptibility of having a stroke, or some other lurking variable. Additionally, these results tell us that dropping these subjects from our training data likely introduced bias into our model. Somewhat unexpectedly, when examining subjects with missing data for "smoking status," we found that only 3.0% of these patients were classified as having a stroke, opposing our findings for those with missing BMI data.

Our final question relates to clustering; we are interested in determining if we can cluster our dataset into *k* clusters using KMeans clustering. We implemented this using SciKit Learn's KMeans clustering algorithm, and used the silhouette score to compare different clusterings. After testing various values of n_clusters, we determined that 2 clusters had the highest silhouette score, and proceeded with *k* = 2 clusters. We clustered based on all variables excluding "stroke." We found that the data naturally partitioned into two sets; one of the sets was primarily people who had a stroke, and the other was primarily people who did not have a stroke. In fact, the cluster assignment matched with the "stroke" column with an accuracy of 82.5%.

Overall, for both the decision tree and neural network, our implementations produced similar results with SciKit Learn's implementation—except for significant variations in the amount of true negatives, where each of SciKit Learn's models had far fewer true negatives.

Despite this, our decision tree was significantly slower than SciKit Learn's tree. From our results, it would be interesting to fine-tune our models—particularly our neural network—to see what performance improvements we could generate. It would also be interesting to see the effect of handling missing data differently rather than simply dropping subjects with missing data. Analysis of the clusters we produced using KMeans clustering could also yield promising results to guide another classification model. Hopefully, more analysis of this data can pave the way to more accurate models, and ultimately save lives by aiding with stroke prevention across the world.