The Impact of Classification Algorithms in Predicting Medical Conditions

Tucker Stantliff
tucker@stantliff.com

***Abstract*—With the rise of machine learning and artificial intelligence in the past decade, society has seen many use-cases from predicting the stock market, building generative AI, building marketing outreach prediction models; however, there is one industry that has the potential of saving lives daily - the medical field. This study examined how Neural Networks, Support Vector Machines, and K-Nearest Neighbor algorithms can be used to predict heart conditions and strokes in a wide variety of patients. When predicting for a stroke, the K-Nearest Neighbors algorithm was the most effective with a 2 percent false negative rate, an f-score of .92, and a Receiver Operating Characteristic (ROC) curve score of 0.97. On the other hand, when predicting for heart disease, the neural network algorithm was the most effective with a 7 percent false negative rate, an f-score of .90, and a ROC curve score of 0.94.**

***Index Terms*—machine learning, healthcare, heart disease, stroke**

I. Introduction

Using AI and machine learning to assist doctors and nurses in diagnosing difficult and life-threatening conditions must be at the forefront of research and experimentation. This empirical study aims to do exactly that by using synthetic data to predict a person's likelihood to have a stroke (dataset one) or a person's likelihood to have heart disease (dataset two). This will be done through a careful evaluation of common classification machine learning algorithms - Neural Networks, Support Vector Machines (SVM), and k-Nearest Neighbor (KNN).

This study will compare the algorithms against one another both within a dataset and between datasets to determine the most effective algorithm given these datasets. For the sake of this study, effectiveness is a holistic understanding of accuracy, precision, and usability within the industry - all to be thoroughly discussed later on.

A. Hypothesis

It is believed that the SVM algorithm will outperform the other algorithms in the study's definition of effectiveness due to its speed, accuracy, and adaptability. It is believed that the SVM algorithm is faster than Neural Networks, and less prone to over-fitting than the KNN algorithm and therefore the more effective method when diagnosing life-threatening conditions.

Identify applicable funding agency here. If none, delete this.

II. Methodology

A. Data Selection

The datasets collected are real-life, anonymized health data from a small group of national and international hospitals and health organizations. The data was subsequently published publicly on the *Kaggle* website.

B. Data Preparation

Each dataset went through a thorough preprocessing method with the goal of providing clean and usable data. During the preparation process, data was stored as a CSV file and imported into a python file as a Panda's DataFrame. The data was then visualized and experimented with to determine if there were any opportunities to engineer any features or remove any noise from the dataset prior to developing the machine learning models.

C. Model Development

All of the machine learning models used throughout this study were imported from pre-existing libraries. The neural network algorithm was imported from TensorFlow, the SVM and kNN Classifiers were both imported from Scikit Learn Library. The neural network with layer activation equation is demonstrated in equation 1.

$$a^{[l]} = \sigma(W^{[l]}a^{[l-1]} + b^{[l]}) \tag{1}$$

The kNN with Euclidean Distance is demonstrated in equation 2.

$$d(x, x_i) = \sqrt{\sum_{j=1}^{n}(x_j - x_{ij})^2} \tag{2}$$

The SVM equation for the linear decision boundary is demonstrated in equation 3.

$$f(x) = w^T x + b \tag{3}$$

The datasets were then split into training and testing sets, 80 and 20 percent respectively. The models were then prototyped in a Jupyter notebook for rapid development, evaluation, and tuning. Once the model tuning was complete, a python file was developed with an adjoining YAML config file.

D. Model Evaluation

Once a model was being prototypes, there was a standardized evaluation process for all models in order to evaluate the effectiveness of a particular model fairly. Each model was evaluated for its precision, recall, f1-score, confusion, ROC curve, wall clock time, and model complexity.

E. Hyperparameter Fine-Tuning

Each model went through a standardized and thorough hyperparameter tuning. For the neural networks, the layers and activation functions were tested and tuned to balance precision-recall, minimize type II errors, and correct over- and under-fitting. The Support Vector Machines went through a CV grid search algorithm to determine the best kernel function. The KNN models went through a random grid search algorithm to determine the optimal significant k value.

III. Dataset One - Predicting Strokes

Dataset one is the stroke prediction dataset from Kaggle. This dataset has 5,110 observations with twelve features [1]. Other than rudimentary information (e.g. gender, age, etc), this dataset has heart health-specific measures such as diagnosis of hypertension, average glucose level, BMI, heart disease, and smoking status. These features are commonly associated with medical comorbidity and pathology [2]. The inclusion of these features will prove to be useful for the model's learning to predict strokes.

A. Data Processing

There were two major issues with the dataset that needed to be addressed prior to building the machine learning models. First, the BMI column had a lot of missing data. It was decided to use one-hot encoding to fill those data points with weight on the median BMI value. Second, the dataset was heavily imbalanced in favor of class 0 (no stroke), this would cause over-fitting and the model will appear highly effective in a controlled environment but operate poorly in the real world. It was decided to use a "Synthetic Minority Over-sampling Technique" (SMOTE) to balance the dataset. This is used to address class imbalance for classification problems. This is particularly useful when the positive class is underrepresented - as was in this case. There are many SMOTE methods, therefore, it was decided to run a test to determine which SMOTE method produced the best method in a logistic regression analysis.

As demonstrated in Figure 1, Borderline-SMOTE performed the best out of all the SMOTE methods. Borderline-SMOTE focuses on generating data near the decision boundary between majority and minority classes. Due to this SMOTE method being the most effective, there is a high likelihood that that there will be misclassifying between minority classes near the decision boundary; however, this balancing method will reduce false negatives which is critical in the purpose behind this experiment.

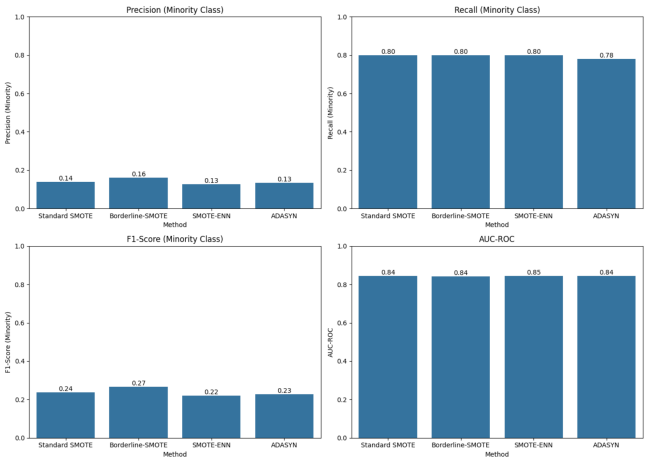

Fig. 1. SMOTE Metric Comparison.

B. Neural Network Results

The Neural Network model performed well in predicting strokes and reducing false negatives. The final architecture of the model consisted of three layers: 64 units with ReLU activation, 32 units with ReLU activation, and 1 unit with sigmoid activation for binary classification. The use of the Adam optimizer ensured efficient weight updates, while the binary cross-entropy loss function was appropriate for the binary nature of the problem.

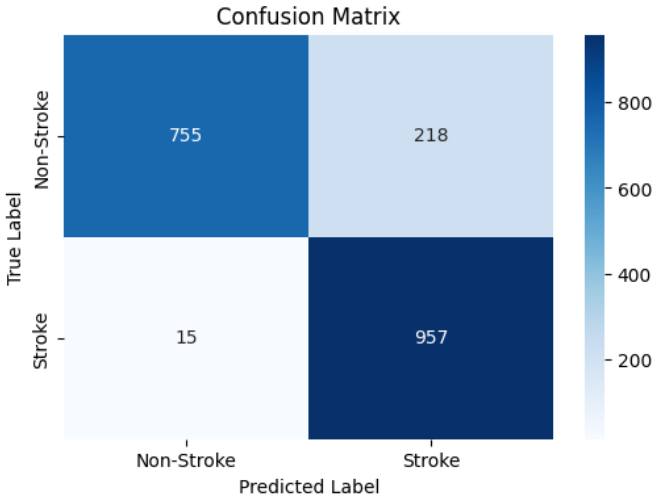

Fig. 2. Neural Network Confusion Matrix.

To emphasize the importance of predicting the stroke correctly, a class weight of three was given to the positive class, and the recall-based early stopping method was used to achieve optimal recall performance without overfilling. The confusion matrix in Figure 2 demonstrates that the class weight and early stopping helped reduce type II error, even if it slightly increased type I. A more thorough graphical representation of the effectiveness of the this model can be found in the README.txt file of the associated GitHub Repository [3].

C. Support Vector Machine Results

The SVM model did not do as well as reaching the effectiveness goals of predicting a stroke. Although it did well in not over fitting the results, the false negative rate was high even after attempting to optimize the early stop function. Figure 3 represents the confusion matrix for the optimal SVM model.

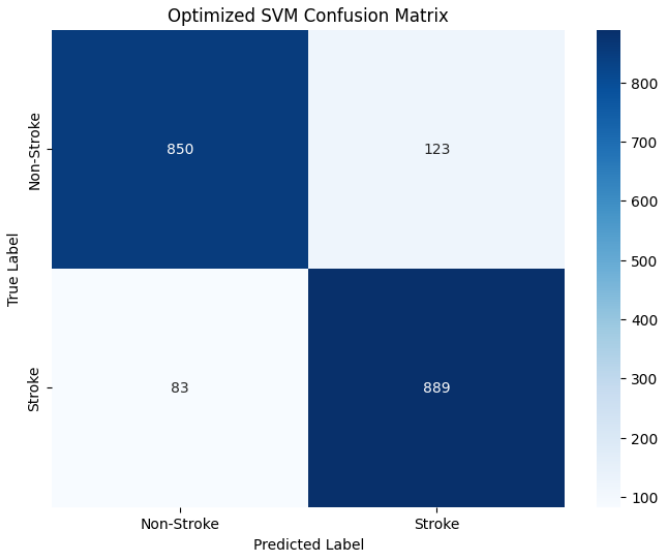

Fig. 3. SVM Confusion Matrix.

The less effective results demonstrates the impact of the data's non-linear relationships between features and SVM does not capture as well. The class imbalanced - although mitigated - may have also played a role in the reduced effectiveness. Perhaps a custom kernel with higher complexity may have produced a better result. A more thorough graphical representation of the effectiveness of the this model can be found in the README.txt file of the associated GitHub Repository [3].

D. k-Nearest Neighbor Results

The results for the kNN model was phenomenal. It greatly outperformed both the Neural Network and SVM models with a precision of .86, recall of .98, f1-score of .92, and a wall clock time of 0.0110 seconds, this model is fast, reliabile, and robust. Figures 4-6 demonstrates the effectiveness of this model.

The confusion matrix in Figure 4 demonstrates that only two percent of non-stroke predictions are false negatives. This is an outstanding achievement. It should be noted that fifteen percent of the stroke predictions were false positives. This could cause undue stress and burden on the patients; however, with further tuning and proper use of the model by doctors the effect may be able to be mitigated.

The ROC curve and the classification report is useful in demonstrating that the kNN for K=15 is an effective model that has not overfitted. It's 86 percent precision is benefited with an astonishing .98 percent recall and .92 f1-score.

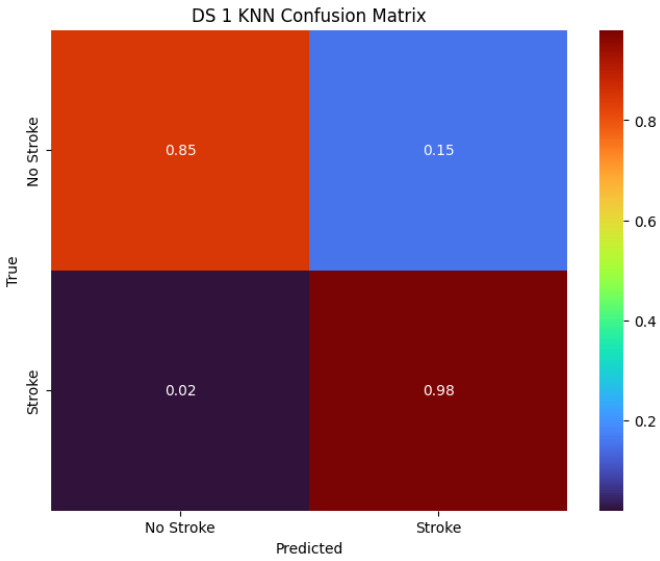

Fig. 4. KNN Confusion Matrix.

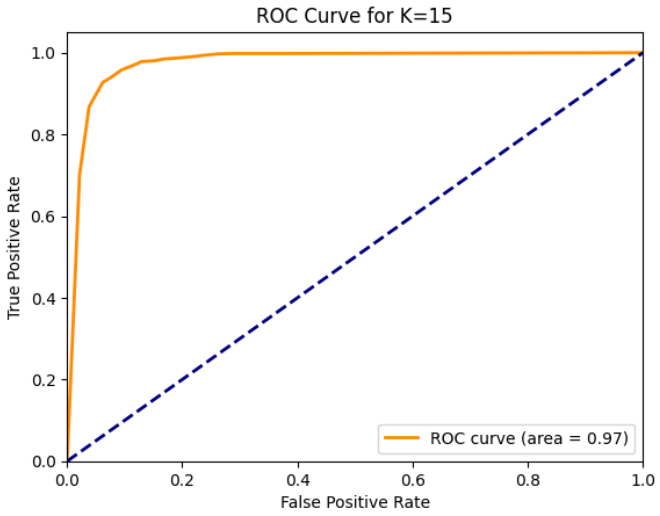

Fig. 5. KNN ROC.

KNN Training Time: 0.0110 seconds

	precision	recall	f1-score	support
0	0.98	0.85	0.91	973
1	0.86	0.98	0.92	972
accuracy			0.91	1945
macro avg	0.92	0.91	0.91	1945
weighted avg	0.92	0.91	0.91	1945

Fig. 6. KNN Classification Report.

E. Dataset Conclusion

Overall, although the neural network did very well for this dataset, the k-Nearest Neighbor algorithm outperformed it in terms of precision and recall, demonstrating that this is an effective method for fast and reliable predictions. While the kNN delivered excellent recall, the increased number in false positives suggest limitations.

F. Dataset Limitations

There are a couple of limitations in this study. They must be examined thoroughly and addressed in future studies:

- **Small Dataset:** The dataset contains only 5,110 observations, which limits the model's ability to generalize to larger and more diverse populations.
- **Class Imbalance:** Class imbalance remains a challenge - despite the use of Borderline-SMOTE.
- **Missing Data:** The BMI feature had missing values, which were filled using one-hot encoding based on the median. This might have introduced inaccuracies because it is not live data.
- **Simple Feature Set:** The dataset only contains 12 features, this may not sufficient to capture the nuanced relationships necessary for accurately predicting strokes.
- **Potential Overfitting:** Although steps were taken to avoid overfitting, the small sample size and the complexity of the network structure could cause overfitting issues.

IV. Dataset Two - Predicting Heart Disease

Dataset one is the heart failure prediction dataset from Kaggle. This data was curated from Cleveland Hospital, Long Beach VA, the Stalog Dataset, and from Hungry and Switzerland. This dataset has 918 observations with eleven features [4]. Other than rudimentary information (e.g. gender, age, etc), this dataset has heart health specific measures such as diagnosis of Chest Pain information, resting blood pressure, cholesterol, blood sugar levels, heart rate, exercise-induced angina indicator, ST segment slope, and ECG results. These useful features are commonly associated with medical comorbidity and pathology for heart disease [2]. The inclusion of these features will prove to be useful for the model's learning to predict strokes.

A. Data Processing

There was one major issue with the dataset that needed to be addressed prior to building the machine learning models. First, the features needed to be encoded. To do this, sklearn's Standard Scaler was used to fit, scale, and transform these features. The Pandas library was used to get dummy values for non numerical values using the .getdummies method. The dataset was balanced and there were no missing values that needed to be encoded, therefore, the data was more useful off the bat than dataset once.

B. Neural Network Results

The Neural Network model performed great in predicting heart disease, outperforming the SVM and kNN models. The model was configured with the following architecture:

- **Input Layer**: 16 input features.
- **Hidden Layers**:
  - First hidden layer: 16 units with ReLU activation and a 0.5 dropout rate.
  - Second hidden layer: 8 units with ReLU activation and a 0.5 dropout rate.
- **Output Layer**: 1 unit with sigmoid activation for binary classification.
- **Optimizer**: Adam with a learning rate of 0.001.
- **Loss Function**: Binary cross-entropy for the binary classification task.
- **Training Process**: The model was trained over 50 epochs with a batch size of 16. Early stopping was used to monitor the validation loss with a patience of 5 epochs, restoring the best weights to avoid overfitting.

The neural network's 7 percent of false negatives are significantly effective because false negatives can result in missed diagnosis possibly leading to loss of life. The model was 88 percent accurate and 93 percent recall. It was better at detecting the positive case as demonstrated in Figure 8. Figure 7 shows the breakdown of the type I and type II errors.

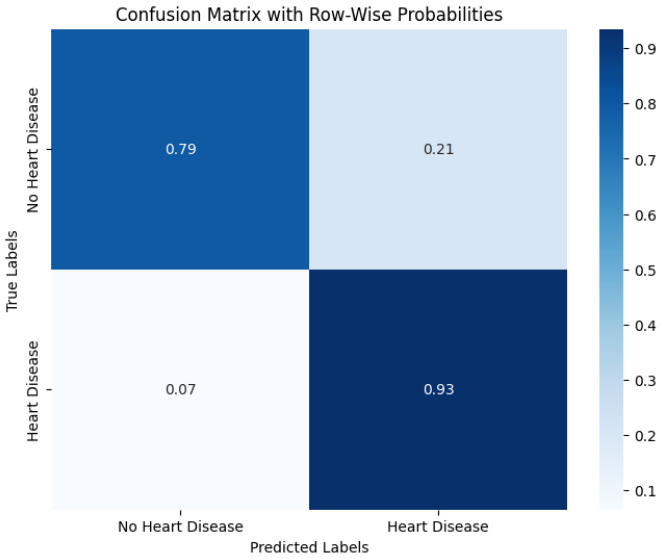

Fig. 7. Neural Network Confusion Matrix.

In addition to the classification report and the confusion matrix, the ROC curve in Figure 9 shows strong performance with an AUC of 0.94. This AUC indicates that the model has a high discriminatory ability between the classes. This ROC suggests that the hyperparameters are fine tuned.

C. Support Vector Machine Results

As with dataset one, the SVM model did not perform as well in reaching the effectiveness goals of predicting heart disease.

	precision	recall	f1-score	support
0	0.90	0.79	0.84	77
1	0.86	0.93	0.90	107
accuracy			0.88	184
macro avg	0.88	0.86	0.87	184
weighted avg	0.88	0.88	0.87	184

Fig. 8. Neural Network Classification Report.

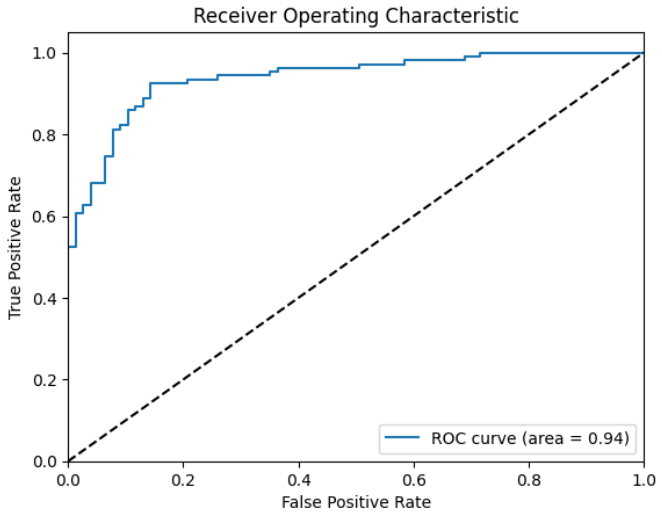

Fig. 9. Neural Network ROC Curve.

Although the linear kernel did well at not over-fitting, the false negative rate was high even after attempting to optimize the early stop function. Three kernels were tested while fine tuning the SVM: Linear, Polynomial (Poly), and Radial Basis Function (RBF) as the regularization parameter C. As seen in Figure 10, the linear kernel shows a much smaller gap between training and validation which demonstrates that this kernel is less prone to overfitting as C increases. This stability locks the complexity to capture non-linear relationships which is why it is outperformed by other models.

D. k-Nearest Neighbor Results

The kNN model for dataset two did not perform nearly as well as for dataset one. There were 11 percent false negatives and 15.5 percent of false positives. The AUC of the ROC is .93 and the precision, recall, and f1-scores are .87 on average. The kNN model is heavily dependent on the local data structure and the distribution of instances in the feature space. This suggests that the complex distance between points led to sub optimal performance. The curse of dimensionality may have also played a role. Although there aren't many features, the kNN model relies on Euclidean distance

$$d(x, x_i) = \sqrt{\sum_{j=1}^{n} (x_j - x_{ij})^2}$$

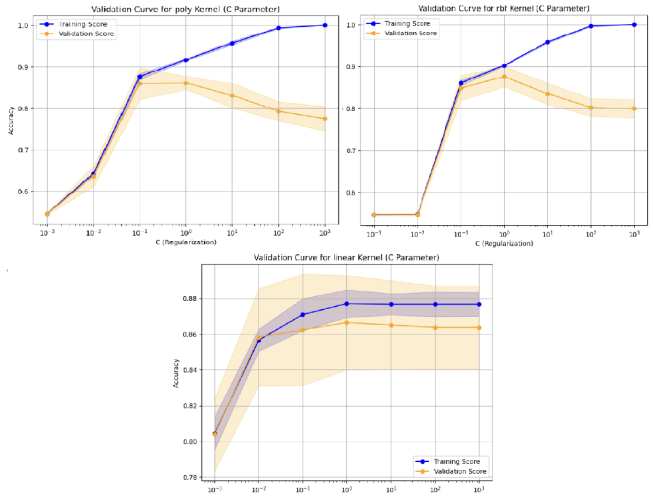

Fig. 10. RBF, Linear, and Poly Kernel for regularization parameter C.

which becomes minimally impactful for higher dimensions. There are opportunities to fine tune the model even further and where time complexity is a must, the lazy learner method of this model is useful for large datasets with a moderately high precision and recall.

E. Dataset Conclusion

Overall, although the k-Nearest Neighbor did moderately okay for this dataset, the neural network implementation outperformed it in terms of precision and recall. It is important to note that although the neural network was very accurate, the time complexity is far larger than the kNN model. Therefore, where time is a consideration, one should consider fine-tuning and implementing the kNN model instead. The neural network also lacks in explainability over the kNN model.

F. Dataset Limitations

There are a couple of limitations in this study. They must be examined thoroughly and addressed in future studies:

- **Small Dataset:** The dataset contains only 918 observations, which limits the model's ability to generalize to larger and more diverse populations.
- **Simple Feature Set:** The dataset only contains 12 features, this may not sufficient to capture the nuanced relationships necessary for accurately predicting heart disease.

V. Conclusion

The hypothesis in which the Support Vector Machine would outperform the neural network and kNN models in both datasets was demonstrated as flawed. SVM models are useful for small to medium-sized datasets. The datasets surely fell into that category; however, it appears that the linear or non-linear nature of the data could not be as effectively separated by a hyperplace with the help of the chosen kernels. Perhaps SVM would have done better with well-structured and higher dimension data with outliers. The decision boundaries were

not as clear, and therefore, kNN and Neural Networks were able to get a boost over SVM. Refer to the Jupyter Notebooks in the GitHub Rebo [6] to see the decision boundaries in the linear regression PCA analysis for dataset 1 as an example.

Overall, the Neural Network performed well in both datasets and therefore is found to be the more effective model across stroke and heart disease predictions. There is however, a need to conduct further research into the limitations of this study. It is recommended that additional experimentations are done on larger datasets where nrow ¿ 100,000, with more features (n¿120), and higher dimensionality. By testing into these limitations, the machine learning community can continue down the path of life-saving artificial intelligence implementation.

REFERENCES

[1] Fedesoriano, "Stroke Prediction Dataset," Kaggle, Feb. 2021. [Online]. Available: https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset.

[2] M. L. Ackermann, M. Edelstein, S. Narayan, J. Brazier, and M. Roberts, "Impact of elevated BMI and types of comorbid conditions on health-related quality of life in a nationally representative US sample," Public Health Nutrition, vol. 24, no. 3, pp. 487–497, Mar. 2021. [Online]. Available: https://www.cambridge.org/core/journals/public-health-nutrition/article/impact-of-elevated-bmi-and-types-of-comorbid-conditions-on-healthrelated-quality-of-life-in-a-nationally-representative-us-sample/14CA649399E1A8339D4DB2ABD42BC661.

[3] T. Stantliff, "CS7641 Assignment 1 Repository," GitHub, Sep. 2023. [Online]. Available: https://github.gatech.edu/tstantliff3/CS7641-A1-tstantliff3.

[4] fedesoriano. (September 2021). Heart Failure Prediction Dataset. Retrieved [Date Retrieved] from https://www.kaggle.com/fedesoriano/heart-failure-prediction.