Predicting Heart Attack Likelihood Using Machine Learning

Brett Geunes

CMSE 492

Michigan State University

Abstract

As of 2021, heart disease has remained the leading cause of death in America. While this has been a constant issue throughout human history, the discovery and application of machine learning techniques has provided strong new insight into treatment and prevention. The goal of this project was to develop a machine learning model that can effectively decide if someone is at an increased risk of heart attack such that they could get in contact with a doctor or decide what next steps to take. This model development used a full ML pipeline with data cleaning, feature selection, and model optimization. The final model worked with Scikit-Learn KBest and XGBoost decision tree gradient boosting to predict likelihood with an accuracy of 0.908. A model with this accuracy can provide a good free step to determine if consulting a doctor is necessary or can be used by doctors to determine the severity of someone's situation.

Predicting Heart Attack Likelihood Using Machine Learning

1. INTRODUCTION

1.1 Heart Attacks

According to the CDC, heart disease has been the leading cause of death in America since 1950. This type of heart disease, known officially as coronary heart disease manifests itself in many ways, however one of the most common and serious ways it appears is in the form of a myocardial infarction, more commonly known as a heart attack. A heart attack occurs when some form of problem or blockage prevents the heart from getting sufficient blood. Without proper understanding or treatment, this lack of blood flow can quickly lead to significant damage or death. Such a significant health concern that reaches a major audience made this stand out as the subject I want to contribute to using my data science knowledge and experience.

1.2 Machine Learning Applications

Techniques in the data science world have not only changed the tech industry, but they have also introduced new ways we analyze and predict issues in every medium we can collect data from. One of the largest jumps forward in this has been from our increased access to information and computational power combined with complex machine learning algorithms. In key longstanding research subjects like heart disease and heart attacks, machine learning innovations have redefined the way we study and introduced a new dimension to prediction and prevention. For this project, my goal was to take the problem of heart attacks and apply these robust machine learning techniques to attempt to predict likelihood of a heart attack, therefore

allowing whoever uses the model to take proper precautions along with necessary preventative measures.

1.3 Existing Research

As comes naturally with such an important and widespread topic, there have been many different research projects applying machine learning techniques to heart attacks along with more general heart disease issues. Reading through research publications on this subject has given me a deeper knowledge on the issue and what has been done already and what did not work, but it also allowed me to focus on things that have not been tried as widely. One dimension I did not see as often and wanted to introduce in my model was more of a spectrum of risk, as opposed to a binary no heart attack or heart attack option. Using a binary classifier like that could introduce some risk, like predicting no heart attack on a case even if it is a small 51% probability. For this reason and concern, I decided to use a binary classifier but display the probability involved with its classification to show users just how confident the model was. With something as serious as heart attacks, staying on the safer side of these decisions is extremely important.

2. OVERVIEW AND GOALS

2.1 Data

One immediate and key issue within this project was procuring data that was useful and readily available without any unique access or grant money. A natural challenge of trying to complete public medical projects is a lack of data access due to patient privacy, as it is the right of all medical patients to keep their data private. This means that there is much less readily available data than other fields, and especially less publicly online. Despite these challenges, I

was able to find a sufficient data set on the data science website Kaggle, called Heart Attack Analysis and Prediction Dataset, developed by Rashik Rahman. This dataset provided some positives as well as some negatives for this project. One good aspect of it was that it required very little cleaning, as there were no missing values anywhere important in the file. However, it also introduced some less desirable aspects. For example, the dataset had a good 14 features, but only had 303 rows. This could be much worse, but a larger dataset would naturally prove to make a more robust model and help avoid drawing any conclusions that do not reflect the larger population. Another unfortunate reality of getting the data from this source is that it is harder to identify where it came from originally, and whether it is from actual patients or synthetic. While it reflects real trends in heart disease regardless, such background would allow conclusions to be slightly deeper and more nuanced.

2.2 Model Goal

The goal of this project and model is to predict a patient's risk of heart attack based on various health factors. The model in question should not only tell whether someone is at increased risk for a heart attack but give the probability score with which it made that decision and its overall accuracy to give whoever uses it the most educated information possible. These results will be achieved using a binary classification algorithm that gives insights into its decision. Naturally, such important health decisions should be made with medical professionals, but this model can still have a home within this world. For example, consulting a doctor can mean very large medical bills, so some people may prefer to wait for more concrete evidence to schedule appointments. This tool could give people a low-price accessible way to gather more information before making that decision. This model could also prove to be a useful aspect of

such medical professionals' toolkits when analyzing patients and deciding what next step to take in their treatment. Overall, there will be many practical instances where this model can be used to identify someone's risk of a heart attack.

3. MACHINE LEARNING WORKFLOW

3.1 Data Cleaning

As mentioned in section 2.1, this data came from Kaggle and was fairly simple to prepare for machine learning. The data came in the form of a CSV (comma separated value) file, and was read into a Python Jupyter Notebook using the Pandas package. Pandas was then used to display the full data frame and search for any missing data or data types that would prove to be a challenge within the model. This was a short process, as there were no missing values and the data was all numerical from the start, with even the ordinal data being already encoded into integers.

3.2 Feature Selection

3.2.1 Initial Exploratory Data Analysis

Once the data was sufficient and ready, exploratory data analysis began. The first step was some very simple exploration of how the data related to each other using a Pair Plot from Python's Seaborn package in conjunction with the Matplotlib visualization package. This allowed for the direct comparison of all features in the data against each other to look for trends.

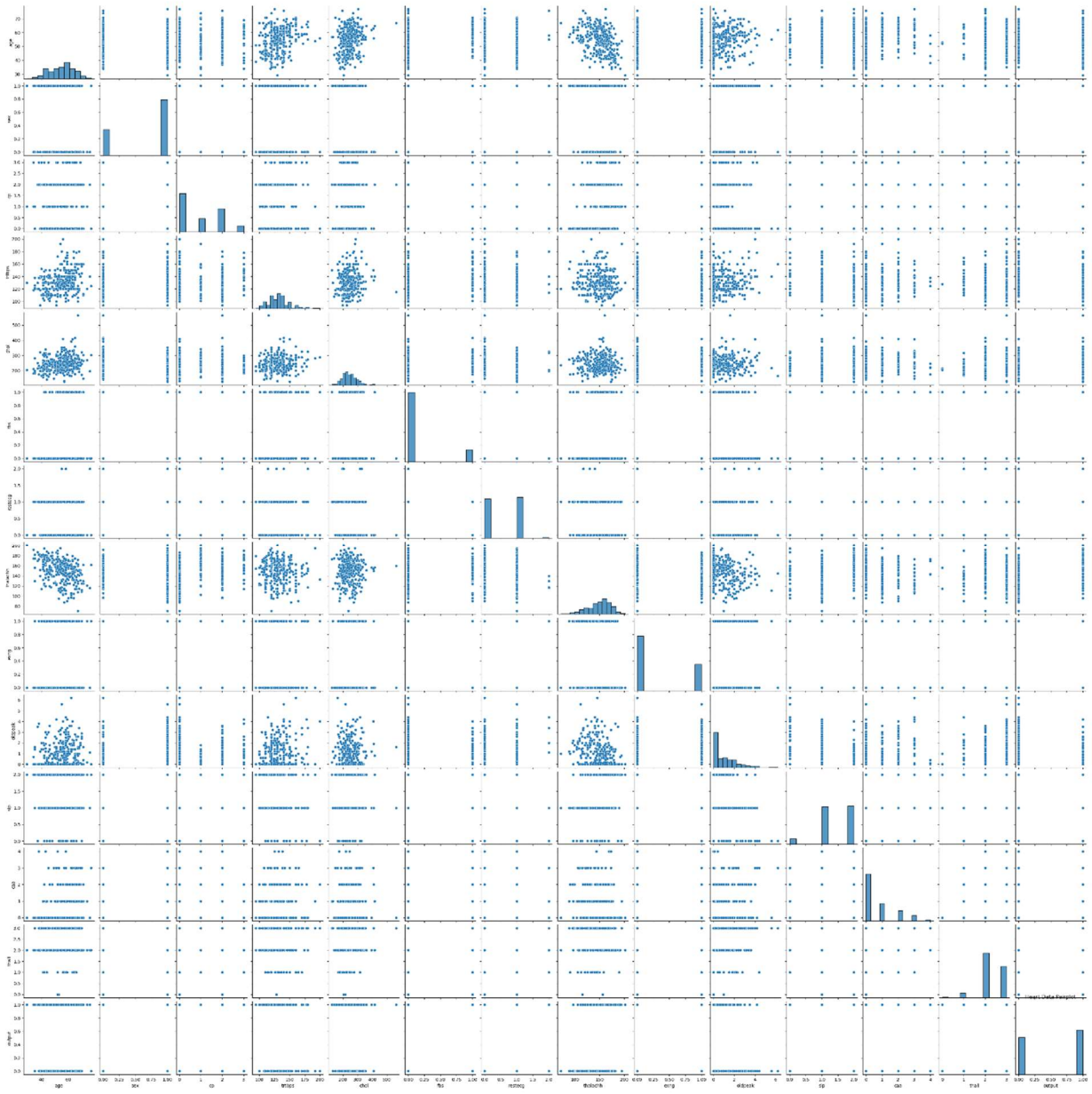

Figure 1: Massive dataset pairplot

From here, individual data was displayed on Matplotlib scatter plots and points were colored based on presence of heart attack risk, to allow for a clear look at how these trends affect heart attack likelihood. These exploratory methods provided new insights but did not give the clear

feature selection answers I was looking for, so more complex dimensionality reduction methods were implemented.

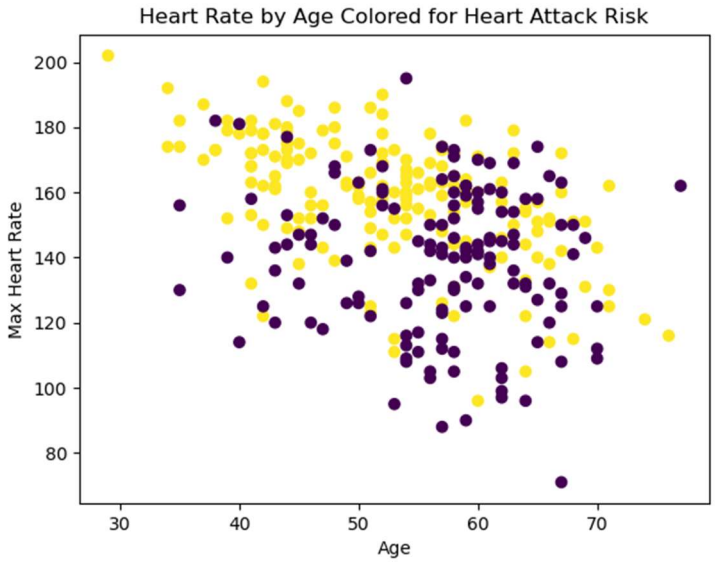

Figure 2: Individual feature scatterplot shaded for risk

3.2.2 Principal Component Analysis

In an effort to reduce dimensionality in a way that gave clearer insight and created stronger hyperparameters, principal component analysis was applied. Principal component analysis, or PCA, is a method that computes complex linear algebra methods on the covariance matrix of a dataset to create vectors called principal components, and then selects the best *n* principal components to be used as hyperparameters in the machine learning model. For this problem, I used three principal components. This allowed me to minimize the amount of information lost through dimensionality reduction while also being able to visualize the data in a way that was easy to understand and analyze for all audiences. To perform principal component analysis, I used the PCA function in the Python Scikit-Learn package. Upon getting my three components, I plotted them all against each other in a three-dimensional scatter plot. When

coloring for heart attacks similarly to 3.2.1, the data showed some linear separability when viewed from the right angle. This insight led to some exploratory applications of a Support Vector Machine model, but this combination was later abandoned when compared to the baseline model described in section 3.3.1.

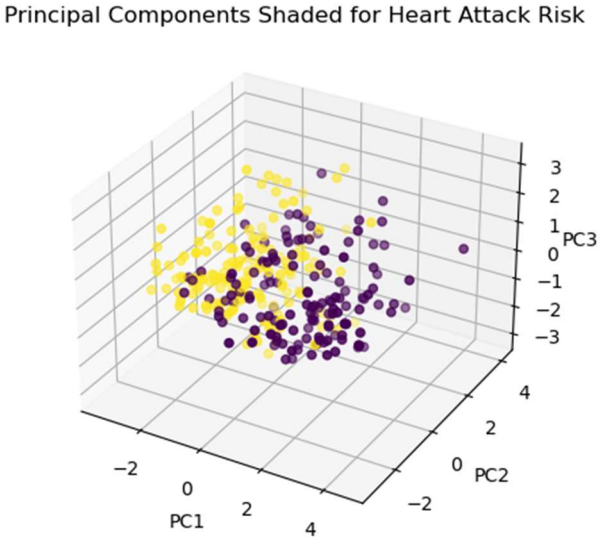

Figure 3: Initial view of 3D PCA data

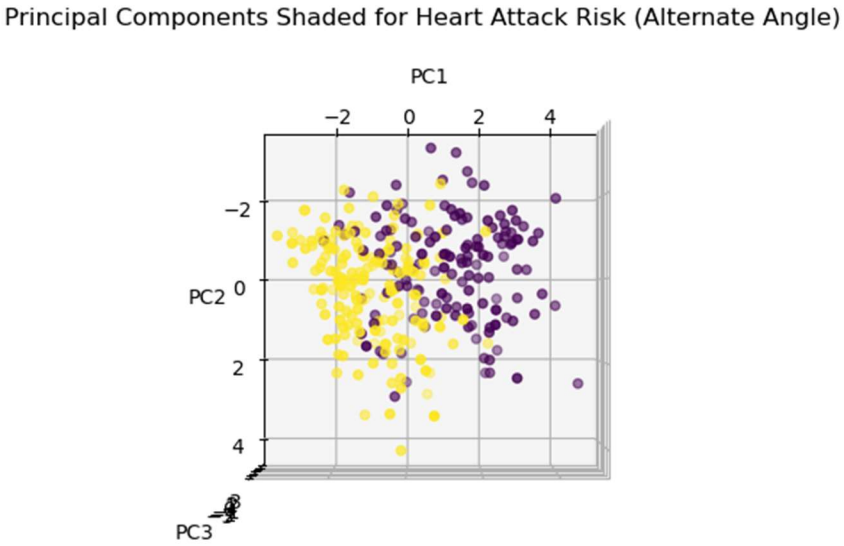

Figure 4: Rotated PCA data to show separability

3.2.3 Select K Best

The final and most successful feature reduction method applied was Scikit-Learn's Select K Best algorithm. This is a supervised feature selection algorithm that computes the ANOVA F-value for each feature column and decides the *k* best columns to use based on highest score. This proved to be the most successful way to focus on only the strongest features without losing any necessary data. Once my model and parameters were decided on, (3.3.3) I optimized this feature by iterating through all possible *k* values and finding which resulted in the strongest accuracy. In the final product, my initial guess of using eight features turned out to be the best instinct, as that the highest accuracy of all the *k* values. Once those eight features were selected, the smaller, more focused data was ready for testing with the algorithms.

3.3 Model Selection

3.3.1 Baseline Model

The first goal of my model selection was to find a baseline model and accuracy to benchmark my explorations against. An extremely efficient and effective way to set up a baseline model is through automated machine learning packages, commonly known as AutoML. The packages automatically create and run many machine learning pipelines on data in an effort to find the strongest option and accuracy score. While these pipelines tend to not be as strong as you can achieve with human working and tuning, they can provide many initial insights. For example, an AutoML score in the 70's lets the researcher know that they should be looking for accuracy scores slightly above that, but there may not be a model that gives a score in the 90's. Similarly, you can look at what model your automated pipeline is using for inspiration when selecting your own models. For this project I used TPOT, a python AutoML package that focuses on genetic programming, or machine learning models based on evolutionary and biological

processes. I ran the TPOT model with 10 generations, meaning that it would try ten different pipelines and learn from them as it developed. The TPOT model gave me an accuracy score of 0.875 that I could compare my further exploration to, and did so using gradient boosting, which I later incorporated into my final model.

3.3.2 Model Exploration

After seeing the methods and results from the TPOT pipeline, combined with the lower accuracy of the support vector machine I trained on the principal components, I pivoted my focus to be on decision tree-based models. Decision trees are algorithms that branch out from decisions made at nodes to classify data. Given the relative weakness of an individual decision tree, many models use a large ensemble of trees to make more robust classifications. The first model I attempted was a random forest classifier. As the name implies, this model creates a randomized "forest" of decision trees, which it uses to try to find an optimal result. For this, I used Scikit-Learn's Random Forest Classifier function. This model resulted in an accuracy score of 0.842, however, so I abandoned it as I did not want to downgrade from the TPOT model. After that, I attempted to use an extra-trees classifier. Extra-trees classifiers are very similar to random forests, but they do not use bootstrapping and decide their splits randomly instead of based on the optimal decision. Similarly to the random forest model, I used the Extra Trees Classifier function from Scikit-Learn. This model returned an accuracy score of 0.829, so I similarly proceeded with a different model.

3.3.3 Gradient Boost Model

After finding that the other models were less successful than what TPOT had found, I decided to instead see if I could optimize their gradient boost model to be even more successful on my own. The package I used was called XGBoost, and it used an ensemble of weaker decision trees, however it used gradient descent methods to navigate them until every point is classified based on a tree that is best fit for its individual case. I attempted a grid search cross validation to find the best parameters for this model, however the parameters as chosen by the TPOT model proved to be the strongest. Combining this model with the Select K Best filtered data (3.2.3), then boosted this model's accuracy by about 3.5%. With a sufficiently accurate model, I then used XGBoost's predict_proba function to show the probability it was classifying each point with. This allowed for a model that gave a prediction and showed how confident it was in that prediction to try to avoid the binary nature of this classification problem.

4. RESULTS

4.1 Model Accuracy

The complete and final machine learning pipeline for this problem consisted of reading data into Python and cleaning with Pandas, splitting for training and testing using Scikit-Learn, selecting the eight strongest features using Select K Best, and applying a gradient boost classification model. This model predicted heart attack risk with an accuracy score of 0.910, stronger than any other found throughout this project. Along with predicting risk, the model also gave a unique probability score for its decision to provide more information about the strength of that individual classification.

4.2 Classification Report

Below is a report of various aspects of the model's accuracy with regard to each individual classification and overall success.

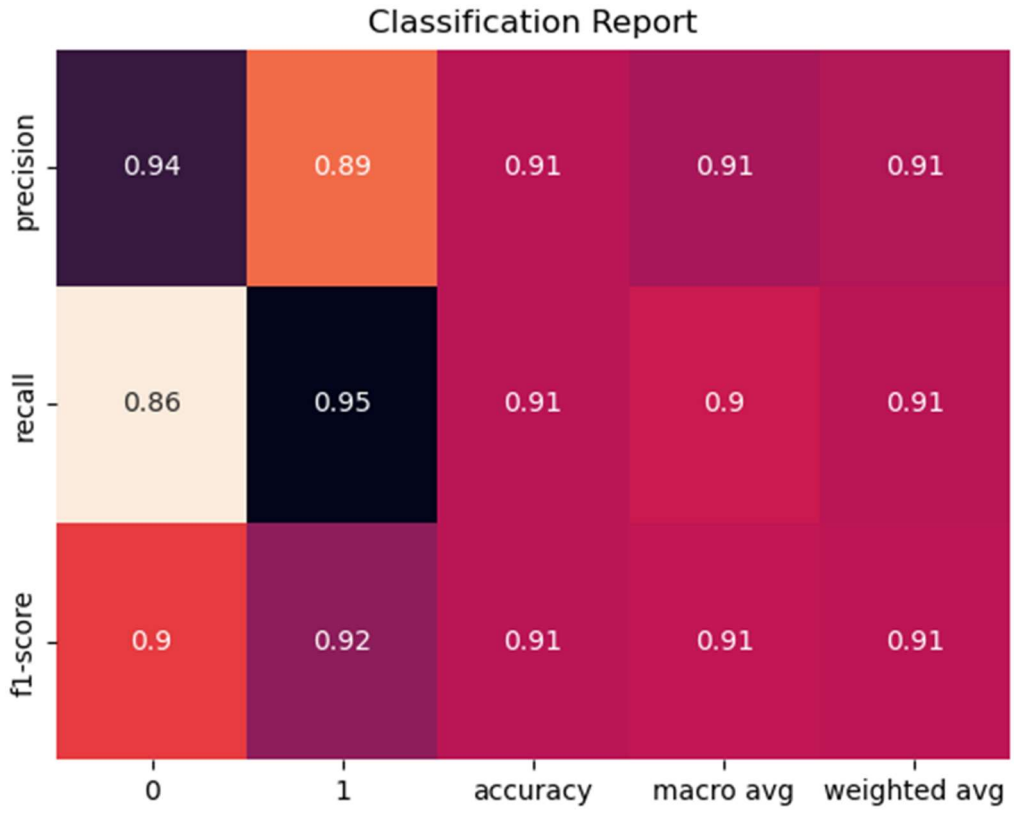

Figure 5: Resulting classification report

5. CONCLUSION

5.1 Analysis

The model created in this project can predict heart attack risk in the testing data correctly 91% of the time, all while providing a useful confidence score with its decision. This can be implemented at all levels of the real world when concerned about heart disease. For example, one can consult this model with their own information to get a read on whether they are at risk and to what extent they should be concerned. Similarly, a medical professional can implement this model as a key part of their analysis when dealing with patients who have heart disease

related concerns. While all major medical decisions should be made with professionals and using equipment that has been heavily tested and approved, this model can still be a helpful supplement for those looking for a quick answer before booking an appointment and could have potential as a professional tool with proper approvals.

5.2 Future Work

While this project was very successful at completing its goal, there are many potential opportunities to expand upon the work done if given more time and resources, like devoted data or funding. This model would greatly benefit from access to a more professional, large-scale medical dataset that it could train better on. It achieved significant results on this smaller dataset, but a 1,000+ row dataset that could be easily traced back to real patients would immediately make it much more verifiable and more likely to be implemented in real medical settings. A different way this project could be expanded upon with time would be in the form of a user-friendly interface for regular people who simply want to check in on their own personal heart attack risk. A useful application with easy inputs of your own health information that returned a prediction with a probability score would greatly increase accessibility for this model.

6. APPENDIX

6.1 Github

Link: geunesbr/492FinalProject (github.com)

References

Chauhan, G. (2021). Xgboost in Python – Guide for Gradient Boosting. *Machine Learning HD*. https://machinelearninghd.com/xgboost-in-python-guide-for-gradient-boosting/

Heart Attack Analysis & Prediction Dataset. (2021, March 22). Kaggle. https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset

Heart Attack: Symptoms and Treatment. (n.d.). Cleveland Clinic. https://my.clevelandclinic.org/health/diseases/16818-heart-attack-myocardial-infarction

Heart Attack Symptoms, Risk Factors, and Recovery | cdc.gov. (2022, July 12). Centers for Disease Control and Prevention. https://www.cdc.gov/heartdisease/heart_attack.htm

Heart disease deaths - Health, United States. (n.d.). https://www.cdc.gov/nchs/hus/topics/heart-disease-deaths.htm#featured-charts

Jaadi, Z. (2021). A Step-by-Step Explanation of Principal Component Analysis (PCA). *Built In*. https://builtin.com/data-science/step-step-explanation-principal-component-analysis

Mordecai, A., II. (2021, December 14). Heart Attack Risk Prediction Using Machine Learning. *Medium*. https://towardsdatascience.com/heart-disease-risk-assessment-using-machine-learning-83335d077dad

Olson, R. S. (n.d.). *TPOT*. http://epistasislab.github.io/tpot/

User guide: contents. (n.d.). Scikit-learn. https://scikit-learn.org/stable/user_guide.html

Gupta, Suraj & Shrivastava, Aditya & Upadhyay, Satya & Chaurasia, Pawan. (2021). A Machine Learning Approach for Heart Attack Prediction. International Journal of Engineering and Advanced Technology. 10. 124-134. 10.35940/ijeat.F3043.0810621.