Predicting Heart Attacks through Analysis of Features

Machine Learning - CS4641

Alex Jaegook Kim
alex.jg.kim@gatech.edu

Ishan Patel
ishan2397@gatech.edu

Introduction

Heart disease remains as the leading cause of death in the United States (*FastStats- Death and Mortality*, 2022). With hundreds of thousands of Americans dying from the disease annually, it is shocking how little the public knows about heart issues and the various health factors that influence their occurrence. A 2019 study published by the International Journal of Environmental Research and Public Health analyzes over 200,000 individuals' knowledge of the factors that influence heart attacks; results indicated that almost 20% of respondents had what was deemed a low "cardiovascular disease knowledge score," with males having lower scores than females and other factors such as "older age" and "lack of regular exercise" further contributing to a lower score (Han et. al, 2019). Acknowledging this, our team set out to create different types of machine learning models to most accurately predict heart attack occurrences. This project predicts the likelihood of suffering from a heart attack based on a set of 13 health factors and an output variable. We implemented three training models to analyze these health factors: a decision tree model with all 13 features, a decision tree model with the 5 highest correlated features, and a decision tree model with 2 highly predictive features. Altogether, this project aims to draw light to how accurately certain health factors can predict a heart attack given the correct learning model.

Data Source

The dataset chosen is licensed under the public domain and is annually updated (Rahman, 2021). The collection methodology is crawling; heart attack data was gathered via a crawler that gathered dataset points from a reputable source. This dataset was chosen because of its thoroughness; features are abundant and well-documented. The dataset is comprehensive with many data points to support the creation and training of machine learning models. The dataset was retrieved via the Kaggle dataset search engine and downloaded under the CC0 license; it was read via the *read_csv()* method.

Data Format

The data contains 13 unique features and an output target variable. The output target variable represents if the individual has suffered from a heart attack; the 13 unique features can be compared to the output variable to draw relationships and patterns.

The 13 features are defined (Jay, 2021):

- *Ex. Column description (Column name)*
- Age of person in years (age)
- Gender of person (sex)
  - 0 = female
  - 1 = male
- Chest pain type (cp)
  - 0 = asymptomatic
  - 1 = typical angina
  - 2 = atypical angina
  - 3 = non-anginal pain
- Resting blood pressure in mm Hg on admission to hospital (trtbps)
- Serum cholesterol in mg/dl (chol)
- Fasting blood sugar >120 mg/dl (fbs)
  - 0 = false
  - 1 = true
- Resting electrocardiograph results (restecg)
  - 0 = hypertrophy
  - 1 = normal
  - 2 = having ST-T wave abnormality
- Maximum heart rate achieved (thalach)
- Exercise induced angina (exng)
  - 0 = no
  - 1 = yes
- ST depression induced by exercise relative to rest (oldpeak)
- Slope of the peak exercise ST segment (slp)
  - 0 = downsloping
  - 1 = flat
  - 2 = upsloping
- Number of major vessels (caa)
  - 0 to 3 colored by fluoroscopy
- Thallium Stress Test Result (thall)
  - 1 = fixed defect
  - 2 = normal
  - 3 = reversible defect
- Diagnosis of heart disease from angiographic disease status (output)
  - 0 ≤ diameter narrowing
  - 1 ≥ 50% diameter narrowing

No data modifications or preprocessing methods were performed on the dataset before feature analysis or machine learning methods. The data was chosen to be kept as unfiltered as possible to ensure high applicability.

Method Documentation and Reasoning

Imports

The names of the libraries and imports used can be found in *Figure 1*.

Basic Feature Analysis

Before implementing any machine learning methods, the team decided to conduct a basic feature analysis with features individuals generally associate with higher rates of heart attacks. Heart attacks are often associated with health factors such as older age, higher maximum heart rates, and higher cholesterol (*Understand your risks*, 2022). The basic feature analysis attempts to determine any similarities between the degree to which these specified health factors truly impact heart attack occurrence. The Gaussian Kernel Density Estimate (KDE) was added onto the graph to better analyze the probability distribution for each compared feature: age, maximum heart rate, and cholesterol. The Gaussian Kernel Density Estimate (KDE) visualizes the probability density function of the dataset by basing the function on kernels as weights (*seaborn.kdeplot*, 2021). To further visualize the data, the seaborn package was imported and the *distplot()* method was utilized. The *distplot()* method was chosen because of its straightforwardness in applying the KDE and plotting the histogram for the compared variable.

Decision Tree Classifier

A decision tree classifier was quickly identified as the ideal model to produce because of the output variable representing a binary value. Decision trees excel as algorithms when performing binary classifications; it is possible to calculate true positives, false positives, true negatives, and false negatives and deduce further conclusions from these values. Decision trees were also a valid machine learning model taught in lecture, further encouraging their use in this project.

Figure 2 shows how the dataset was defined. The *X* variable contains the 13 features and the *Y* variable contains the output. The dataset was split 80 to 20, where 80% of the dataset was used to train the model and 20% was used to test the model. The random state was determined through multiple tests to maximize the accuracy of the model. To use the *train_test_split()* method, it was imported from the library *sklearn.model_selection* as shown in *Figure 1*. After splitting the data, the *MinMaxScaler()* function is called to normalize the data and have all the data be from a range from 0 to 1. Using the *fit_transform* method that is built into the *MinMaxScaler* function, this allows the data to be fit and transformed respectively. This is done for the data stored in *X_train* and *X_test* as there are values that range out of 0 and 1 unlike the data stored in *Y_train* and *Y_test* which are all 0 or 1s.

Figure 3 shows how the model was created and trained. A *DecisionTreeClassifier()* was imported from the library *sklearn.tree*. The *random_state* was set to 10 as it resulted in the highest accuracy for the model. The *criterion* was set to *entropy*, an information theory which calculates the average level of "information" of a variable's possible outcomes (Gray, 2013).

In order to get the results to analyze the decision tree model, *precision_recall_fscore_support* and *accuracy_score* were imported from *sklearn.metrics* library to calculate the accuracy, mean square error, precision, recall, and f-score. The accuracy returns how accurate the model is (TP+TN)/(TP+FP+TN+FN) where T is True, F is False, P is Positive, and N is Negative. The mean square error measures the average square difference between the estimated value and actual value. Precision quantifies the number of positive class predictions that actually belong to the positive class: TP/(TP+TF). Recall is the ratio of correctly predicted to all observations in the actual class: TP/(TP + FP). And f-score is the weight average between precision and recall: *2 * (recall * precision) / (recall + precision)*. All these components are essential in knowing whether the model is working well as a model can be accurate but predict a lot of false positives or false negatives which will potentially be harmful if the model were to be used in real life.

In order to visualize the decision tree models, ROC curves were used. *plot_roc_curve* was imported from *sklearn.metrics* and the *plot_roc_curve()* function was used to display each graph. A curve representing a truly random classifier was also plotted on each graph using the *.plot()* function for further visualization.

Findings Recording and Findings Analysis

Basic Feature Analysis Results

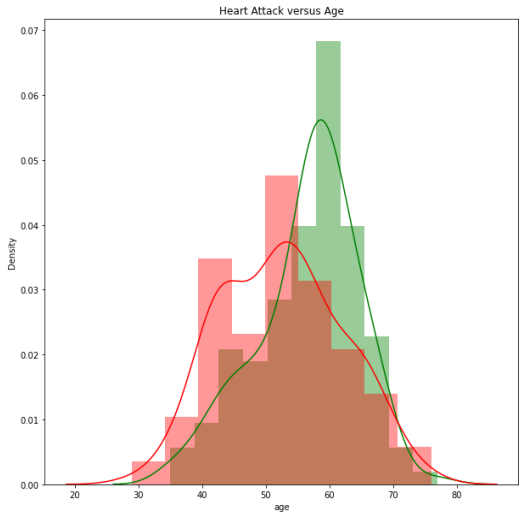

Figure 4

From our basic feature analysis, we had various hypotheses depending on which variable we were comparing with the output. For all three graphs, the green represents the people who are less likely to get heart attacks and the red represents the people who are more likely to get a heart attack, with respect to the variable being analyzed. For *Figure 4*, we hypothesized that older people are more likely to get heart attacks. From the graph, there is a weak correlation between age and likelihood of getting a heart attack. We did not expect, however, the dataset to show a high density of people who are less likely to get a heart attack from ages between 60 to 63. This counters the hypothesis as there seems to be a less likelihood of heart attacks for older people.

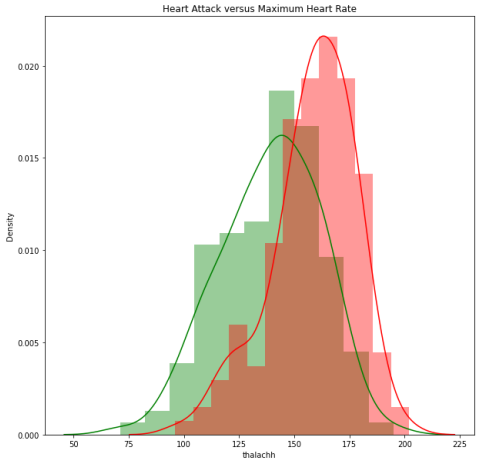

Figure 5

For *Figure 5*, we compare the likelihood of heart attack to the maximum heart rate recorded for each person. We hypothesized that the higher the maximum heart rate, the more likely the person will have a heart attack. The graph supports the hypothesis as the red histogram has an average of around 165 whereas the green histogram has an average of about 140. It was expected and seems conclusive that a maximum heart rate plays a significant factor in the likelihood of getting a heart attack.

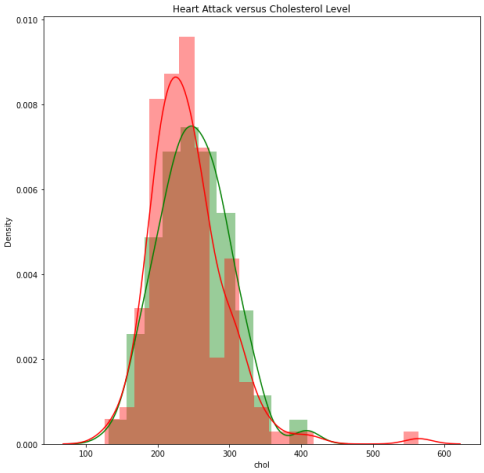

Figure 6

Figure 6 shows the relation between cholesterol level and the output. We hypothesized that a person with higher cholesterol levels will more likely have a heart attack. The analysis from the dataset shows otherwise that cholesterol levels are not a determining factor to likelihood of heart attacks as the two histograms overlap one another.

Decision Tree Results

```
Execution time: 0.00427 seconds
Accuracy: 88.52459%
Mean Square Error: 0.11475
Precision: 0.88889
Recall: 0.91429
F-Score: 0.90141
```

Figure 7

Figure 7 reveals the execution results for Model 1. Given that this model incorporated all 13 features in defining and training the model, it is logical that the execution time is the longest. Accuracy was the highest out of all three models created; this is also logical because this analysis is the most thorough because it contains every feature in the dataset. Mean square error remains verifiable with the accuracy value (1 - accuracy). Precision, recall, and f-score are all extremely high, indicating a high number of positive class predictions and a high proportion of those predictions being accurate.

```
age       -0.225439
sex       -0.280937
cp         0.433798
trtbps    -0.144931
chol      -0.085239
fbs       -0.028046
restecg    0.137230
thalachh   0.421741
exng      -0.436757
oldpeak   -0.430696
slp        0.345877
caa       -0.391724
thall     -0.344029
```

Figure 8

Figure *8* displays the correlations between each feature and its related output variable. These correlations were obtained using the *.corr()* method. The features with the five most significant correlations (exng, cp, oldpeak, thalach, and caa) are used in Model 2.

```
Execution time: 0.00178 seconds
Accuracy: 85.2459%
Mean Square Error: 0.14754
Precision: 0.88889
Recall: 0.91429
F-Score: 0.90141
```

Figure 9

Figure 9 reveals the execution results for Model 2. It is logical that this model produces an accuracy closest to the decision tree with all 13 features despite only having 5 features. These 5 features are most correlated with the output, and thus it is logical for a high accuracy to still be achieved despite a lower number of features. Once again, mean square error is verifiable with the accuracy value (1 - accuracy). Precision, recall, and f-score are all exactly the same as Model 1, indicating the same number of positive class predictions and the same proportion of those predictions being accurate.

```
Execution time: 0.002 seconds
Accuracy: 80.32787%
Mean Square Error: 0.19672
Precision: 0.87097
Recall: 0.77143
F-Score: 0.81818
```

Figure 10

Figure *10* reveals the execution results for Model 3. This model produces the lowest accuracy of the three presented; however, it is extremely remarkable for a model with only two features to be only 8% less accurate in its predictions than a model with 11 more features than it. Model 3 uses only sex and chest pain as its features during its defining and training; these two features alone can be a whopping 80.32787% accurate. Though this model is the least accurate, it can be considered the most significant simply due to how insightful it may be in predicting heart attack occurrences given a limited amount of features. Mean-square error checks out with the accuracy again (1 - accuracy = mean-square error). Precision, recall, and f-score are all noticeably lower than in the other models; but this is understandable given the lower accuracy.

Model Visualizations

Figures *11*, *12*, and *13* display a receiver operating characteristic curve (ROC) for each decision tree model along with an area under the curve (AUC) calculation for each. An ROC curve reveals the performance of a classification model at all classification thresholds and plots two parameters: true positive rate and false positive rate (Google, 2022). The dotted red line on these graphs reveal the expected performance of a truly random classifier. AUC values are displayed in the legend for each visualization.

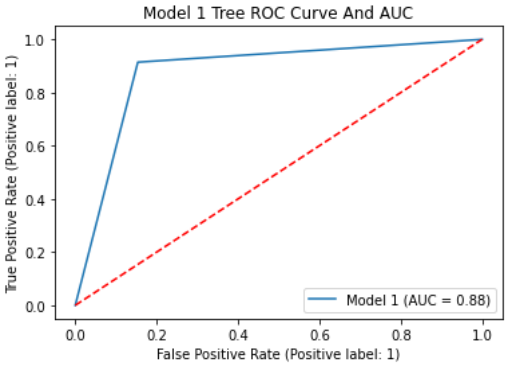

Figure 11

Figure 11 shows the ROC curve and associated AUC value for Model 1

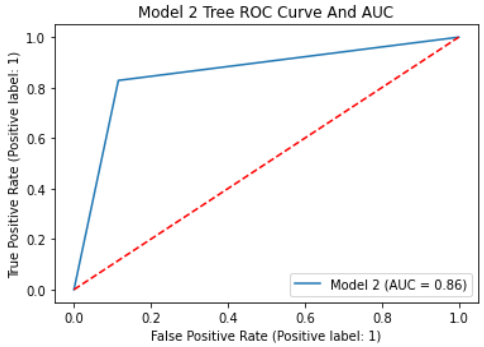

Figure 12

Figure 12 shows the ROC curve and associated AUC value for Model 2.

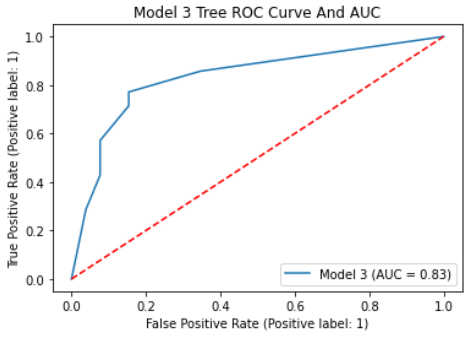

Figure 13

Figure *13* shows the ROC curve and associated AUC value for Model 3.

Future Works

Going forward, there are multiple potential improvements and expansions to the project. Firstly, none of the models were able to reach an accuracy of >95%; producing any type of machine learning model, decision tree or not, that can achieve an accuracy of >95% is the first step to improvement and can have massive implications for the healthcare industry if put into practical use. Alternatively, more decision tree models can be produced to determine

if there is another pair of features other than sex and chest pain that can produce a highly accurate training model. Determining more pairs of features that can predict heart attack occurrences extremely accurately could have extreme implications in the medical field; physicians may be able to motivate their patients to be wary of their risk for heart attack if patients only had to track two health factors as opposed to 13. Further machine learning models can be created for the same purpose of predicting heart attack occurrences: k-nearest models and various boosting algorithms are all valid models that can likely produce accuracies significantly greater than what a standard decision tree can produce. If given months to work on this project, we would attempt to incorporate all of these ideas: more types of machine learning models and more decision trees with highly specialized pairs of features. Altogether, this project reveals that this machine learning model has massive implications in the medical world and many more machine learning models are yet to be made from this dataset.

Figures

```
#Imports
import pandas as pd
import numpy as np
from sklearn.metrics import precision_recall_fscore_support as precision_score
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import plot_roc_curve
import time
import matplotlib.pyplot as plt
import seaborn as sns
```

Figure 1

```
#Code spliting and normalizing the dataset
X = heart_attack_data.iloc[:,:13]
Y = heart_attack_data["output"]

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=102)

X_train = MinMaxScaler().fit_transform(X_train)
X_test = MinMaxScaler().fit_transform(X_test)
```

Figure 2

```
#Code training model
start = time.time()

model_tree = DecisionTreeClassifier(random_state=10,criterion="entropy")
model_tree.fit(X_train,Y_train)
Y_pred = model_tree.predict(X_test)

finish = time.time()

model_time = round((finish - start), 5)

print("Execution time:", model_time, "seconds")
```

Figure 3

References

Centers for Disease Control and Prevention. (2022, January 13). *FastStats- Deaths and Mortality*. Centers for Disease Control and Prevention. Retrieved July 17, 2022, from https://www.cdc.gov/nchs/fastats/deaths.htm#:~:text=Number%20of%20deaths%20for%20leading,Accidents%20(unintentional%20injuries)%3A%20200%2C955

Google Developers. (2022). *Classification: Roc curve and AUC | machine learning |; google developers*. Google Developers. Retrieved July 22, 2022, from https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc#:~:text=An%20ROC%20curve%20

Gray, R. G. (2013). *Entropy and Information Theory (1st ed.)*. Electrical Engineering Department Stanford University.

Han, C. H., Kim, H., Lee, S., & Chung, J. H. (2019). Knowledge and Poor Understanding Factors of Stroke and Heart Attack Symptoms. *International Journal of Environmental Research and Public Health*, 16(19). https://doi.org/10.3390/ijerph16193665

Jay, J. (2021). *Heart Attack Analysis & Prediction Dataset* [Dataset]. Kaggle. https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset/discussion/234843?datasetId=1226038&sortBy=voteCount&select=heart.csv

Rahman, R. (2021, March 22). Heart attack analysis & prediction dataset. Kaggle. Retrieved July 23, 2022, from https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset?resource=download

Understand your risks to prevent a heart attack. American Heart Association. (2022, June 29). Retrieved July 22, 2022, from https://www.heart.org/en/health-topics/heart-attack/understand-your-risks-to-prevent-a-heart-attack

seaborn.kdeplot — seaborn 0.11.2 documentation. (2021). Seaborn. Retrieved July 24, 2022, from https://seaborn.pydata.org/generated/seaborn.kdeplot.html#seaborn.kdeplot