Marist College

Predicting Death from Heart Failure using Machine Learning

Genevieve Anderson
Machine Learning DATA 440
Professor Lauria
December 7th 2022

Abstract

For my final project, I will attempt to use machine learning to predict instances of death from heart failure. I will use a dataset I found from Kaggle with 12 observations and one target variable. I will clean and familiarize myself with the data prior to creating any models. Then, I will create a Gaussian naive Bayes model, a logistic regression model, random forest model, XGBoost model. I will fine-tune the Random Forest and XGBoost models using randomized search. I will calculate predictive performance metrics for all models and compare results against one another. I hope to explore whether machine learning and data science could contribute to the healthcare field. Collecting and analyzing data from healthcare could lead to new discoveries that have lasting impacts on doctors and patients.

Introduce & Describe Topic

Cardiovascular diseases and heart failure is the leading cause of death globally. It accounts for approximately 31% of deaths worldwide. For my final project, I would like to predict the occurrence of death from heart failure. Since cardiovascular diseases are the leading cause of death so I am curious as to whether we can use machine learning and data science to predict instances of death. I will use several of the machine learning models we have discussed in class to predict the likelihood of death from heart failure I am hoping to gain insight as to whether people may be able to stop heart conditions early on based on the observations in this dataset. I would like to see if these specific observations are successful in predicting instances of death. Healthcare is something that I am passionate about and I am curious to combine data science with healthcare to see what kind of results I may discover.

The dataset is from Kaggle includes information on age, anaemia, creatinine, diabetes, blood ejection, blood pressure, platelets, serum creatinine, serum sodium, gender, smoking habits, time since last seen doctor, and death instances.

Now, I will describe each of the specific fields in the dataset. The field for age contains continuous data with unique numerical values. The field for anaemia contains boolean values as to whether or not there was a decrease in red blood cells or hemoglobin. The field for creatinine contains continuous data with unique numerical values pertaining to the level of the CPK enzyme in the blood measured in mcg/L. The field for diabetes contains boolean values as to whether or not the person had diabetes. The field for ejection fraction contains percent values pertaining to the percentage of blood leaving the heart at each contraction. The high blood pressure field contains boolean values pertaining to whether or not the person had high blood pressure. The platelets field contains continuous numerical values pertaining to platelets in the

blood measured in kiloplatelets/mL. The serum creatinine field contains continuous numerical data pertaining to the level of serum creatinine in the blood measured in mg/dL. The serum sodium field contains continuous numerical data pertaining to the level of serum sodium in the blood measured in mEq/L. The sex field contains boolean values pertaining to the person's sex. The smoking field contains boolean values pertaining to whether or not the person is a smoker. The time field contains numerical continuous data pertaining to the time since the person's last visit to the doctor measured in days. Then lastly the death even field contains boolean values pertaining to whether or not the person died from heart disease. This will be the target variable.

I plan to clean the data prior to any creation of models to remove null values and data we may not need. I will then split the data into train and test sets. Then I will create a naive Bayes model, a logistic regression model, random forest, and XGBoost models. I plan to use randomized search to fine-tune my random forest and XGBoost models.

Literature Review

Prior to beginning my project, I researched more about heart disease and its symptoms. I found that the leading risk factors for heart disease is dependent on blood pressure, high low-density lipoprotein (LDL) cholesterol, diabetes, smoking, obesity, diet, and physical activity. Our dataset contains information on blood pressure, diabetes, and smoking, so I will go in-depth on these symptoms.

Nearly 1 in 2 adults in the US have high blood pressure. When a person has high blood pressure, the lining of their arteries becomes damaged. This causes an increase of plaque (fat, cholesterol, calcium, and more) in the arteries which narrows the blood's pathway to the heart and brain. Sodium is also a large contributor to high blood pressure. Over 70% of sodium that Americans intake is added to food prior to purchasing it. Americans can avoid high sodium by making more home-cooked meals with simple ingredients rather than buying pre-mixed or processed foods from grocery stores. To avoid high blood pressure, a healthy diet and physical activity are recommended.

Having diabetes greatly increases one’s chance of having heart disease. People that have diabetes tend to have high blood sugar. High blood sugar causes damage to blood vessels in the heart or even blocks blood vessels to the brain which in turn causes heart disease or stroke.

Smoking is another leading cause of heart disease. The CDC states that smoking is a cause of 1 in 4 deaths related to heart disease. Smoking raises fat content in the blood while also lowering HDL cholesterol (this is a “good” cholesterol). Smoking also increases the chance of having blood clots which block blood flow to the heart and brain. Smoking damages cells that

line blood vessels which can cause a buildup of plaque within the blood vessels. Smoking greatly thickens the blood and narrows blood vessels which often leads to heart disease.

Now, I will go on to discuss research on the machine learning techniques I will use in this project. I plan to use naive Bayes, logistic regression, random forest, and XGBoost.

Naive Bayes is a classification model based on Baye's theorem. Naive Bayes isn't just one algorithm, but a collection of several based on a central idea. Naive Baye's models are based on the idea that all variables that are being classified are independent of each other. In theory, every variable will be independent of each other and also have equal importance. Naive Bayes models are based on Baye's theorem which finds the probability of an event occurring based on the probability that another event has already happened. When adding the "naive" assumption to Baye's theorem we are integrating the idea that the features are all independent of each other. A popular implementation of a naive Bayes model is Gaussian naive Bayes, which I implement later in my project. Gaussian naive Bayes works best with continuous data where there is a normal distribution. Some other implementations of naive Baye's models are Bernoulli naive Bayes for binary data and multinomial naive Bayes for text data.

Logistic regression is another model typically used in prediction and classification. Logistic regression predicts the likelihood of an event occurring based on a set of independent variables. Logistic regression is used when predicting a categorical variable such as 0 or 1. When performing logistic regression, one of the important factors is the sample size. If the dataset is not large, then the model may not have the computational power to output an accurate prediction. When used for machine learning, logistic regression uses the process of gradient descent to find the global maximum. A downside of logistic regression is often prone to overfitting, especially when there are a hefty amount of predictors in the data set. Logistic regression is considered a

supervised learning model and a discriminative model. This means that it is able to differentiate between categories. This differs from naive Bayes, for example, because naive Bayes is not able to generate information on what it is trying to predict.

Random forest models use multiple decision trees to achieve an output. Random forest models have gained a lot of popularity due to the fact that they are able to handle both classification and regression problems. Decision trees are formed based on a series of questions about the data set. In the context of the heart disease dataset, consider did the patient have diabetes? Was the patient also a smoker? Each question asked leads to a decision made. Decision trees attempt to find the best way to subset the data. Alone, decision trees can often be prone to overfitting and bias. However, when put together in a random forest model, they are able to provide more accurate results. Especially when the separate trees are uncorrelated from each other.

XGBoost is one of the most popular machine learning algorithms today. XGBoost is an application of gradient-boosted decision trees. Here, decision trees are built in a sequential form. Each variable in the data will have a calculated "weight" prior to it being used in the decision trees. In the first decision tree, if a variable is incorrectly predicted, it will have a higher weight going into the next tree. This method leads to a more accurate and precise model.

I will not use neural networks, however, I was able to find a study that used neural networks in order to predict death from chronic disease. The study aimed to predict breast cancer, diabetes, heart attack, hepatitis, and kidney disease. The study showed that creating a neural network achieved 91.61% accuracy. Close behind was the random forest model with 91.19% accuracy. However, it is noted that the random forest model had a much higher processing time than the neural network. This study is unique because typically prediction

techniques are designed for use on only one type of disease at a time, however, this study tried to use the same neural network to predict several chronic illnesses. This resulted in several unimportant features being removed and high weights being implemented on the more important features. The neural network model for predicting chronic illness was overall successful.

Report

Now, I will discuss my code for the final project. First, I uploaded my data into a pandas data frame. I cleaned my data by checking for null values and the type of data for each variable. I was fortunate that the dataset I choose had no null values and the variable types would all cooperate with my models. Then, I created my target variable for the death event. I checked the distribution of the target and noticed that in about ⅓ of the instances there was a death (*figure 1*). So, my target was not perfectly normally distributed, which I thought may cause some issues going forward. Instead of removing some of the instances where the person lived to create a normal distribution, I decided to leave it alone because my data only had 299 rows which in retrospect is not very many. I split the data into train and test sets using a 70/30 split to prepare for the creation of my models.

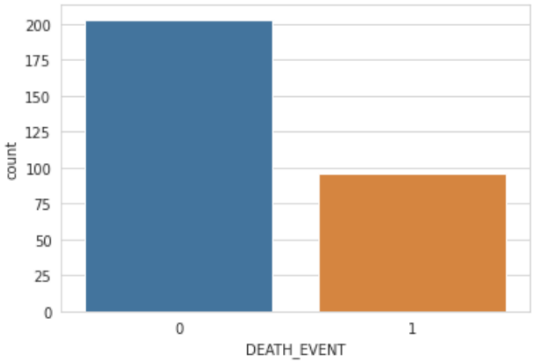

figure 1

First, I created my Gaussian naive Bayes model. I proceeded to then print the parameters of each variable such as the mean, standard deviation, and conditional density. Then, I went on to make my predictions for the model. In the classification report, I was able to achieve overall

accuracy of 77% The recall for prediction life instances was 94% whereas the recall for predicting death instances was 51% (*figure 2*).

	precision	recall	f1-score	support
0	0.74	0.94	0.83	53
1	0.86	0.51	0.64	37
accuracy			0.77	90
macro avg	0.80	0.73	0.74	90
weighted avg	0.79	0.77	0.75	90

figure 2

Then, I created my logistic regression model. After running my model, I made predictions and calculated the probabilities of the predictions. This model performed slightly worse with 74% overall accuracy. The recall for life instances was 92% and for death instances was 49% (*figure 3*).

	precision	recall	f1-score	support
0	0.72	0.92	0.81	53
1	0.82	0.49	0.61	37
accuracy			0.74	90
macro avg	0.77	0.71	0.71	90
weighted avg	0.76	0.74	0.73	90

figure 3

Next, I created my random forest model. After creating my model I decided to tune the model using a randomized search. The randomized search uses a “fit” and “score” method for tuning the model. I printed the classification report for both the non-tuned and tuned model. It was interesting to see that the non-tuned model has 78% accuracy with 87% recall meanwhile the tuned model had 77% accuracy with 81% recall (*figure 4*).

```
No tuning:
              precision    recall  f1-score   support

           0       0.94      0.75      0.83        67
           1       0.54      0.87      0.67        23

    accuracy                           0.78        90
   macro avg       0.74      0.81      0.75        90
weighted avg       0.84      0.78      0.79        90


[[50 17]
 [ 3 20]]


Randomized:
              precision    recall  f1-score   support

           0       0.91      0.75      0.82        64
           1       0.57      0.81      0.67        26

    accuracy                           0.77        90
   macro avg       0.74      0.78      0.74        90
weighted avg       0.81      0.77      0.78        90


[[48 16]
 [ 5 21]]
```

figure 4

Lastly, I created an XGBoost model. After running my model, I also tuned it using randomized search. Again, I achieved 77% accuracy with 56% recall on predicting death instances (*figure 5*).

```
Accuracy.........: 76.6667
Precision........: 80.7692
Recall...........: 56.7568
FP Rate..........:9.4340
ROC AUC (probs)..: 0.881693
Confusion matrix.:
 [[48  5]
 [16 21]]
```

figure 5

After achieving the best of 78% accuracy on my models, I decided to see if I could do better. Based on my research I discovered that smoking, diabetes, and blood pressure were the

main causes of heart disease. So I modified my dataset to only have these observations. At this point, all of my data was binary so I used a Bernoulli naive Bayes model to run this data. Unfortunately, these results were even less desirable with 59% accuracy and recall (*figure 6*).

	precision	recall	f1-score	support
0	0.59	1.00	0.74	53
1	0.00	0.00	0.00	37
accuracy			0.59	90
macro avg	0.29	0.50	0.37	90
weighted avg	0.35	0.59	0.44	90

figure 6

Discussion & Conclusions

In conclusion, random forest had the best statistics with XGBoost and naive Bayes closely behind. For this data, logistic regression performed the worst. The non-tuned random forest had the best accuracy with 78% and recall of 87%. The recall in this model was vastly better than any of the other models which are promising. Although 78% isn't a great accuracy, in the context of life or death it may not be that bad. I thought it was interesting to see that variables other than smoking, diabetes, and blood pressure played a big role in even achieving the 78% accuracy. Going forward, it would be interesting to see how important these “other” variables are in the prediction of death from heart failure. I would also like to explore the use of a neural network on this type of data. From my literature review, it is clear that this method would likely be successful.

Somewhere I could have improved was my data set. I had three main issues with my dataset; size, distribution, and field categories. My dataset only had 299 records which is quite small. If my models had more data to train themselves on I am confident they would have performed better, especially logistic regression. As mentioned previously, my target variable was also not normally distributed. If the target was normally distributed, the models would have had an easier time predicting death instances. The recall percentage would have been much higher for predicting death instances if there were more death records. This is promising because most of my models had recall percentages of over 90% for predicting life instances which had more records than death instances. In my research, I found that physical activity and cholesterol play a large factor in whether one has heart disease. Perhaps if there were fields containing this information the models would have also performed better.

Sources

Centers for Disease Control and Prevention. (2022, September 8). *Heart disease and stroke*. Centers for Disease Control and Prevention. Retrieved December 6, 2022, from https://www.cdc.gov/chronicdisease/resources/publications/factsheets/heart-disease-stroke.htm#:~:text=Leading%20risk%20factors%20for%20heart,unhealthy%20diet%2C%20and%20physical%20inactivity

GeeksforGeeks. (2022, July 11). *XGBoost*. GeeksforGeeks. Retrieved December 6, 2022, from https://www.geeksforgeeks.org/xgboost/

GeeksforGeeks. (2022, August 24). *Naive Bayes classifiers*. GeeksforGeeks. Retrieved December 6, 2022, from https://www.geeksforgeeks.org/naive-bayes-classifiers/

GeeksforGeeks. (2020, February 10). *ML: Linear Regression vs logistic regression*. GeeksforGeeks. Retrieved December 6, 2022, from https://www.geeksforgeeks.org/ml-linear-regression-vs-logistic-regression/

IBM Cloud Education. (2020, December 7). *What is Random Forest?* IBM. Retrieved December 6, 2022, from https://www.ibm.com/cloud/learn/random-forest

IBM Cloud Education. (n.d.). *What is logistic regression?* IBM. Retrieved December 6, 2022, from https://www.ibm.com/topics/logistic-regression

Mutha, N. (2020, May 30). *Bernoulli naive Bayes*. OpenGenus IQ: Computing Expertise & Legacy. Retrieved December 6, 2022, from https://iq.opengenus.org/bernoulli-naive-bayes/

Rashid, J., Batool, S., Kim, J., Wasif Nisar, M., Hussain, A., Juneja, S., & Kushwaha, R. (2022, February 22). *An augmented artificial intelligence approach for chronic diseases prediction*. Frontiers. Retrieved December 6, 2022, from https://www.frontiersin.org/articles/10.3389/fpubh.2022.860396/full#:~:text=A%20feature%20selection%2Dbased%20machine,applied%20to%20anticipate%20disease%20presence.