Tyler Anderson
DSC 680
9/15/2020

Abstract

This report will discuss Cardiovascular diseases (CVDs) as it's a silent killer as well as being one of the top causes of death globally. My aim is to see if high blood pressure is the top leading cause of heart disease and to predict to a reasonable accuracy the death event for a patient with high blood pressure. This assumes that high blood pressure is the leading cause, however if it is not the leading cause based on the data then I will use what is the best variable/symptom that gives the best results. I plan to use model fitting for multiple models and compare the results to determine the best fitting model for this data. After analyzing the data my hypothesis that high blood pressure was the leading cause of Cardiovascular diseases was incorrect and instead the Ejection Fraction feature had the most influence in deaths caused by CVDs.

This dataset includes features that are factors in determining the risk for CVD. It is the number 1 cause of death globally, taking an estimated 17.9 million lives each year, which accounts for 31% of all deaths worldwide. Heart failure is a common event caused by CVDs and this dataset contains 12 features that can be used to predict mortality by heart failure. Most cardiovascular diseases can be prevented by addressing behavioral risk factors such as tobacco use, unhealthy diet and obesity, physical inactivity and harmful use of alcohol using population-wide strategies. People with cardiovascular disease or who are at high cardiovascular risk (due to the presence of one or more risk factors such as hypertension, diabetes, hyperlipidemia or already established disease) need early detection and management where a machine learning model can be of great help.

Meaning of Variables

Age – The age of the individual

Anemia - Is 1 or 0 with 1 being the patient does have this condition. Anemia is a condition in which you lack enough healthy red blood cells to carry adequate oxygen to your body's tissues.

Creatinine Phosphokinase - Level of CPK enzyme in the blood

Diabetes - Is a 1 or 0 - whether the patient suffers from diabetes or not - like anemia

Ejection Fraction - Is a percentage (numerical between 0 to 100) Ejection fraction is a measurement of the percentage of blood leaving your heart each time it contracts.

High Blood Pressure - Is a 1 or 0 - whether patient suffers from high blood pressure

Platelets – Number of platelets in the blood

Serum Creatine - Level of Creatine produced from the kidneys in the blood

Serum Sodium - Level of Sodium in the blood

Sex - assuming 1 is male and 0 is female

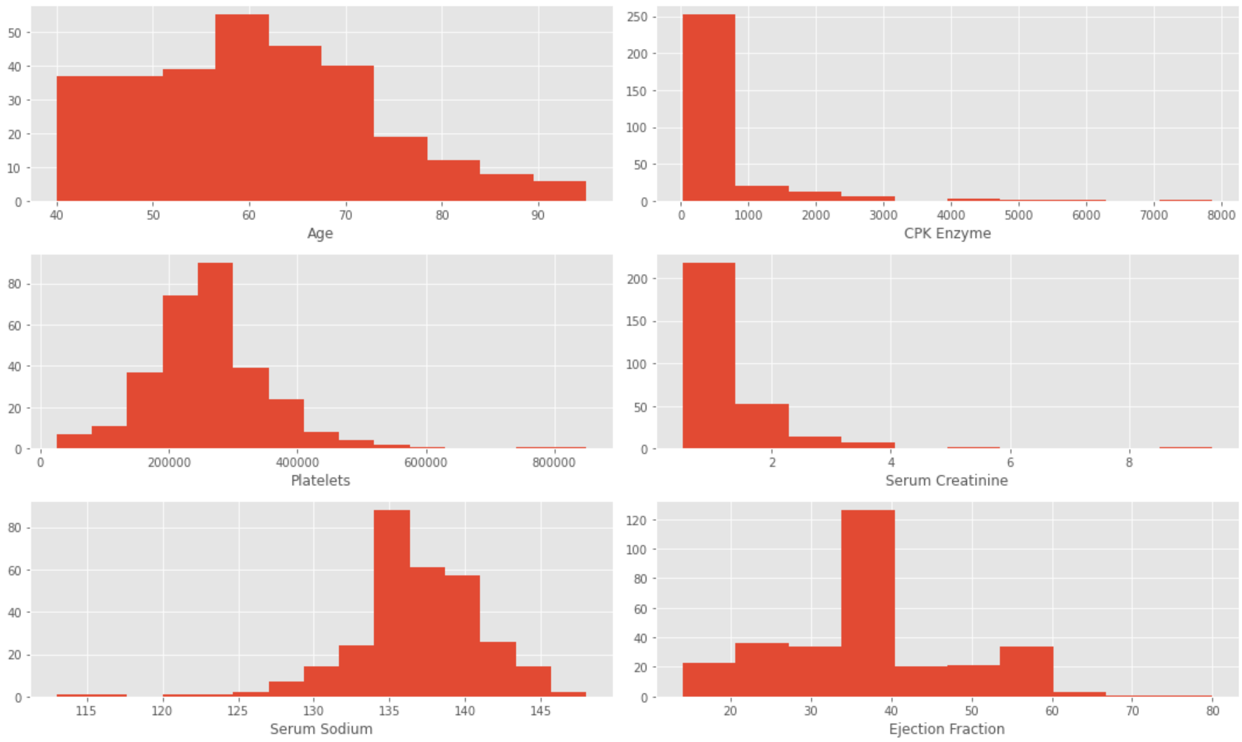
Smoking - assuming 1 is smokes and 2 is doesn't smoke

Time - Follow up days

Death Event - whether patient died during follow up period

Analysis

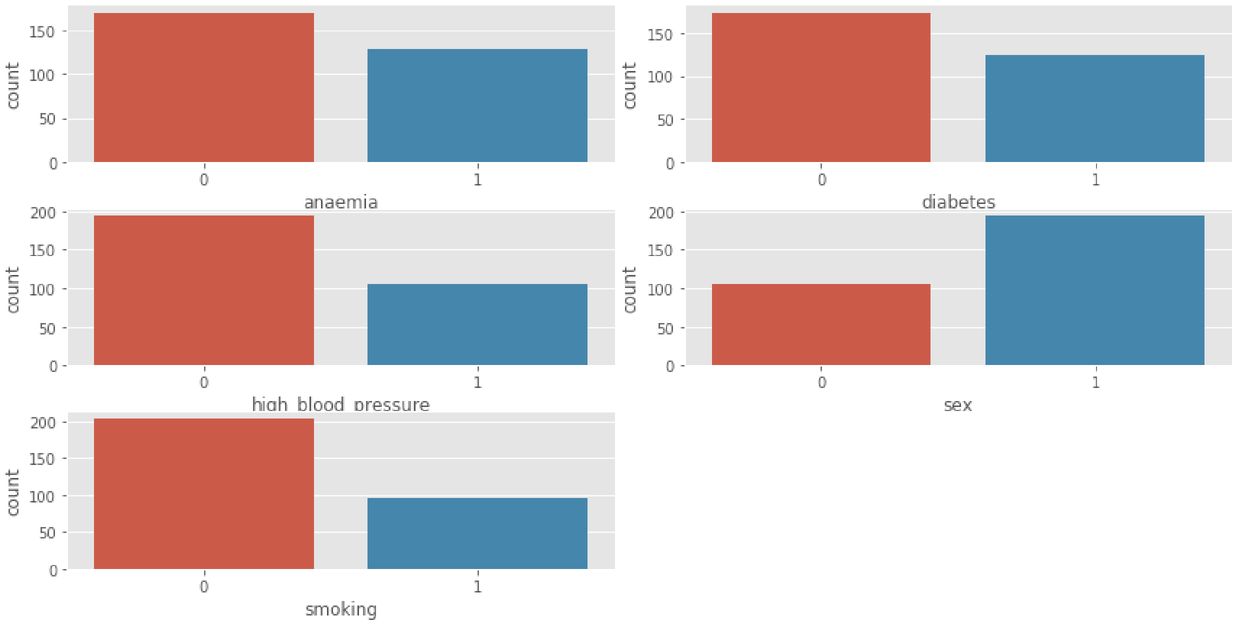
My first step was to clean the data and first looked at outliers for each variable. Based on my observations by using df.describe() Creatinine Phosphokinase (CPK) had outliers as the max value was way higher than expected at the 75% quartile. This is also true for Platelets and Serum Creatinine and ejection fraction. The remaining variables seem to have consistent data and have no outliers. Using a cool-warm heatmap we can see that we have no missing data. Looking at the data visually we can see the data more clearly and get some more information about this data.



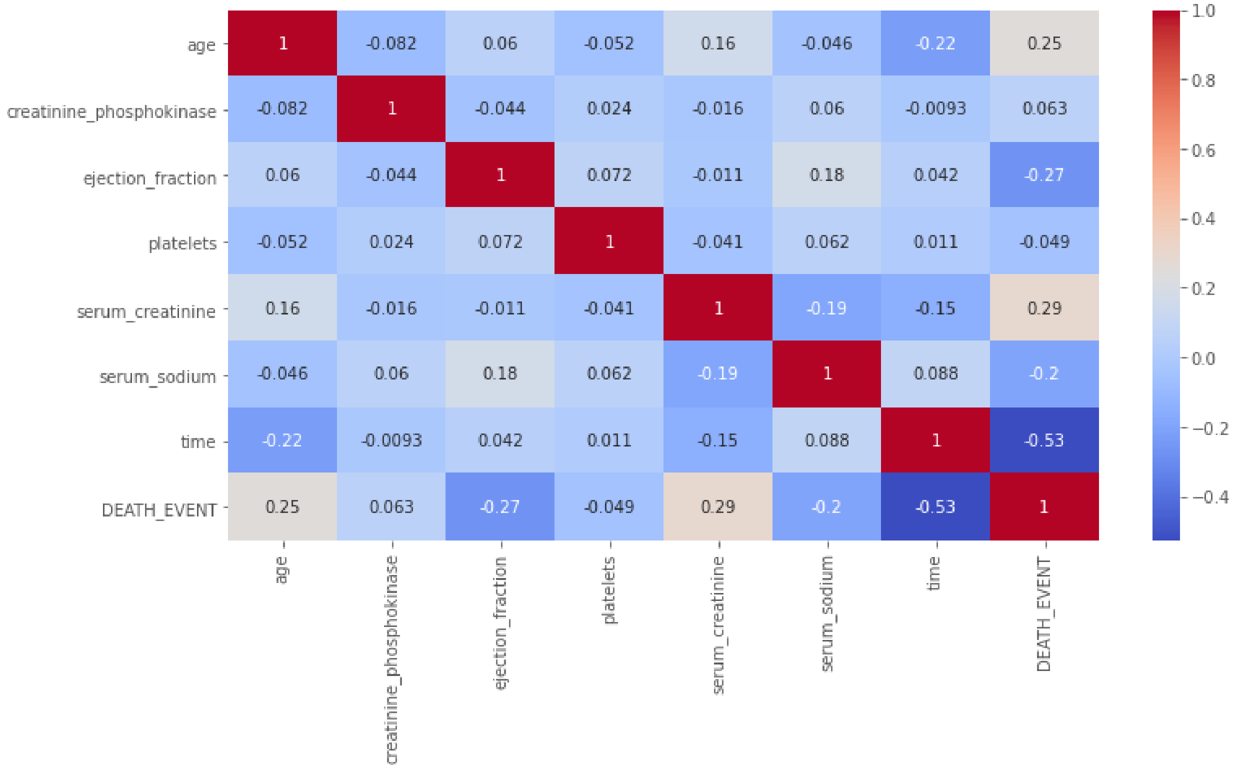
We can see that the majority of patients are around 50-65 with around one third of them above 65. When looking at CPK we can confirm the outliers based on this histogram as well as the other features, ejection fraction, serum sodium, platelet counts and creatinine. Most of the patients CPK values are below 1,000. Platelets were between 150k-400k. Serum Creatinine values were below 2 for 75% of patients. Sodium values were between 132-142.Ejection

Fraction was varied as some patients were low at around 15% or high at 55% when excluding outliers.

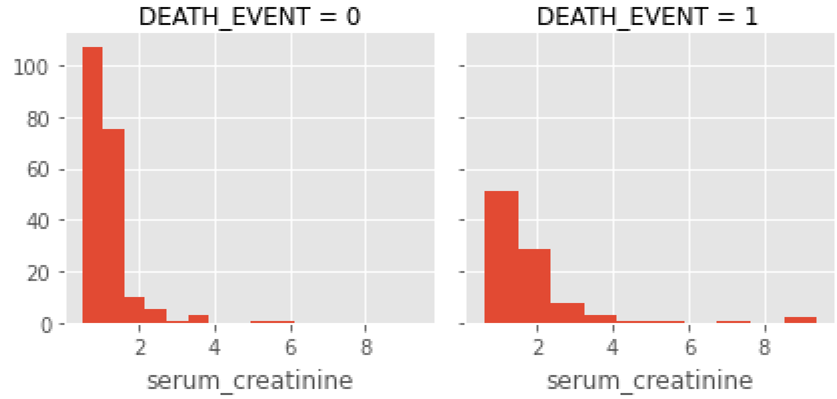
Next, we can see the distributions of the categorical variables. I looked at anemia, diabetes, high blood pressure, sex and smoking as they were binary so it would be easy to spot some differences and gain some insight. The majority of patients were male about 65% with 1 being Male. About a third of patients suffered from high blood pressure, smoking or both at once. Around 40% of patients had diabetes, anemia or both as well. Based on the analysis so far, we can come to the assumptions that the variables do not need to be completed meaning the variables do not need data removed or cleaned beyond adjusting for outliers. So far it seems like women have a higher chance of survival than men do and those who had less time between follow ups had a higher chance of survival and patients who were younger than 60 had a higher chance of survival.



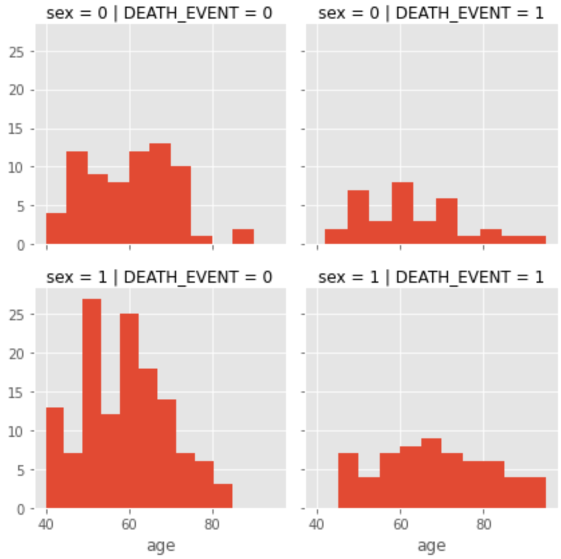
I also wanted to see the relationship between the came variables but against the death event variable to take a look at sex, smoking, anemia, diabetes and high blood pressure to see if there were strong correlations between them and the death event. After evaluation I notice there is not any obvious correlation between their sex or whether they smoked against whether they died or not. My previous assumption that none of the variables need to be changed was wrong as these variables may not be used in our final conclusions. However, before we discard these variables I need to see if its possible to create new features out of these before I discard them. I could create a new categorical feature denoting whether the patient has an underlying condition or not like diabetes, anemia, high blood pressure.



After looking at the heatmap we can see that the highest correlation is the serum creatinine variable and after looking at it further we can see there is a small correlation with this one.



Based on all the plots and visualizations we can see that individuals especially over 70 had higher chances of dying and those under 50 had higher chances of survival suggesting the need to band ages. Patients with a larger amount of follow up days tended to survive the most while those with greater than 50 days ended up with a higher mortality. Another interesting thing was the ejection fraction, those with high percentages had lower deaths and those with lower percentages of about less than 30% had more deaths. Two features, Platelets and CPK, seem to not have any effect with survival. So we have four numerical variables that can be used for our future model building. I also want to see any relationships between the categorical variables and numerical variables.



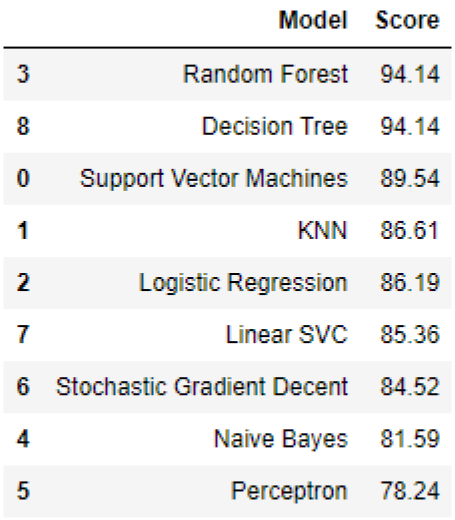
Looking this I can see that we don’t see much of a relationship for smoking against age even when we split into whether a patient smokes or not, the same can be said for sex so we can be confident in dropping these two variables in this analysis. Overall, the most useful features currently are age, time, serum creatinine and the ejection fraction. However, at the same time we can create a new column indicating whether a patient has any underlying condition whether it be anemia, high blood pressure or diabetes. After creating a new column and comparing it to the death event it still does not give us any correlation so we can also drop this variable.

After removing outliers, we can create our training and test sets. Since this is a classification and a regression problem, we can predict an output based off other independent features, additionally as this is a supervised problem as we will have a training dataset to train our model against. Here are the models I will be using as they all can be used similarly to one another for this project.

- Logistic Regression
- KNN or k-Nearest Neighbors
- Support Vector Machines
- Naive Bayes classifier
- Decision Tree
- Random Forrest
- Perceptron
- Artificial neural network
- RVM or Relevance Vector Machine

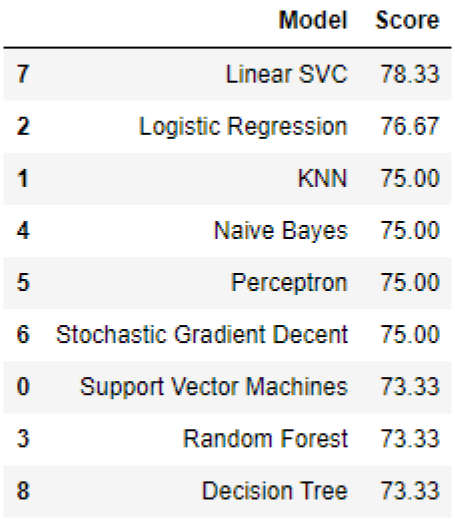
After getting the results we have test scores for our training set and test set. Out training set unsurprisingly had a high percentage with the highest being 94% and our test set with 78%.

The training set



	Model	Score
3	Random Forest	94.14
8	Decision Tree	94.14
0	Support Vector Machines	89.54
1	KNN	86.61
2	Logistic Regression	86.19
7	Linear SVC	85.36
6	Stochastic Gradient Decent	84.52
4	Naive Bayes	81.59
5	Perceptron	78.24

The test set



	Model	Score
7	Linear SVC	78.33
2	Logistic Regression	76.67
1	KNN	75.00
4	Naive Bayes	75.00
5	Perceptron	75.00
6	Stochastic Gradient Decent	75.00
0	Support Vector Machines	73.33
3	Random Forest	73.33
8	Decision Tree	73.33

Conclusions

Based on our results it would seem that the Linear SVC model produces our best results of 78.3% accuracy on the test set with the best model under the training set being the Random Forest with 94% accuracy. I think for this project there is clear room for improvement if we can adjust hyperparameters using cross-validation I’m unsure how to do this yet, however. I think we got some interesting results and while my methods could be more efficient I think we got decent scores and was a fun project to work on.

Questions

1. How can I adjust hyperparameters using cross-validation on this project?
2. What other methods are there that are better than the ones I used
3. Was my reasoning for dropping certain features sound?
4. What kind of improvements could I make to improve this project?
5. Did I make any glaring errors or mistakes in my code?
6. If there are better models out there what are they and how can they be used in my project?
7. Where their variables I should have used but didn't?
8. Did I use the models correctly?
9. Was my analysis sound with the visuals I presented?
10. Was I correct in recognizing my hypothesis to be wrong?

References

Bento, C. (2020, July 6). *Support Vector Machines explained with Python examples.* Retrieved from towardsdatascience: https://towardsdatascience.com/support-vector-machines-explained-with-python-examples-cb65e8172c85

Deb, D. (2020, June 8). *Decision tree for classification and regression using Python.* Retrieved from dibyendudeb: https://dibyendudeb.com/decision-tree-using-python/

K, D. (2020, Mar 17). *Naive Bayes Classifier in Python Using Scikit-learn.* Retrieved from heartbeat.fritz.ai: https://heartbeat.fritz.ai/naive-bayes-classifier-in-python-using-scikit-learn-13c4deb83bcf

Koehrsen, W. (2017, Dec 27). *Random Forest in Python.* Retrieved from towardsdatascience: https://towardsdatascience.com/random-forest-in-python-24d0893d51c0

Kumar, N. (2019, Feb 19). *Implementing The Perceptron Algorithm From Scratch In Python.* Retrieved from hackernoon: https://hackernoon.com/implementing-the-perceptron-algorithm-from-scratch-in-python-48be2d07b1c0

Li, S. (2017, Sep 28). *Building A Logistic Regression in Python, Step by Step.* Retrieved from towardsdatascience: https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8

McCullum, N. (2020, July 3). *How to Build and Train K-Nearest Neighbors and K-Means Clustering ML Models in Python.* Retrieved from freecodecamp: https://www.freecodecamp.org/news/how-to-build-and-train-k-nearest-neighbors-ml-models-in-python/

Navlani, A. (2019, Dec 27). *Support Vector Machines with Scikit-learn.* Retrieved from datacamp: https://www.datacamp.com/community/tutorials/svm-classification-scikit-learn-python

randerson112358. (2019, Aug 10). *Build Your Own Artificial Neural Network Using Python.* Retrieved from medium: https://medium.com/@randerson112358/build-your-own-artificial-neural-network-using-python-f37d16be06bf

Galarnyk, M. (2019, Jul 31). *Understanding Decision Trees for Classification (Python).* Retrieved from towardsdatascience: https://towardsdatascience.com/understanding-decision-trees-for-classification-python-9663d683c952#:~:text=Understanding%20Decision%20Trees%20for%20Classification%20%28Python%29%201%20Classification,that%20they%20are%20relatively%20easy%20to%20interpret.%

Jaroli, H. (2019, April 8). *K-Nearest Neighbors (KNN) with Python.* Retrieved from datascienceplus: https://datascienceplus.com/k-nearest-neighbors-knn-with-python/#:~:text=%20K-Nearest%20Neighbors%20%28KNN%29%20with%20Python%20%201,whether%20someone%20will%20TARGET%20CLASS%20or...%20More%20

Li, S. (2017, Sep 28). *Building A Logistic Regression in Python, Step by Step.* Retrieved from towardsdatascience: https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-stepbecd4d56c9c8#:~:text=Building%20A%20Logistic%20Regression%20in%20Python%2C%20Step%20by,%28yes%2C%20success%2C%20etc.%29%20or%200%20%28no%2C%20failure%2C%20etc.%29.

Chicco, D., Jurman, G. Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone. BMC Med Inform Decis Mak 20, 16 (2020). https://doi.org/10.1186/s12911-020-1023-5

Ahmad T, Munir A, Bhatti SH, Aftab M, Ali Raza M. Survival analysis of heart failure patients: a case study. Dataset. https://plos.figshare.com/articles/Survival_analysis_of_heart_failure_patients_A_case_study/5227684/1. Accessed 25 Jan 2019.

National Heart Lung and Blood Institute (NHLBI). Heart failure. https://www.nhlbi.nih.gov/health-topics/heart-failure.

The Guardian. UK heart disease fatalities on the rise for first time in 50 years. https://www.theguardian.com/society/2019/may/13/heart-circulatory-disease-fatalities-on-rise-in-uk.

World Health Organization, World Heart Day. https://www.who.int/cardiovascular_diseases/world-heart-day/en/.

https://github.com/davidechicco/cardiovascular_heart_disease