Predictive Data Analysis

CS5812

Bartosz Napieralski - 1713314

# Research Question

Our primary research question is to investigate the probability of a person suffering from heart failure based on the conditions of their circulatory system, age, and lifestyle. As such the aim of this is to use exploratory data analysis, machine learning and deep learning to identify the most important factors to be able to predict heart failure.

Our primary research question is, what conditions in the circulatory systems, affect the possibility of dying due to a heart attack and to what extent?  
Furthermore, an additional feature would be to predict the possibility of a patient is to suffer from heart failure in the upcoming future based on his current condition.

# Data

The data that we are using consists of 12 clinical features that can be used for predicting death events due to heart failure. This dataset can be found on Kaggle with the following link: <https://www.kaggle.com/datasets/andrewmvd/heart-failure-clinical-data>

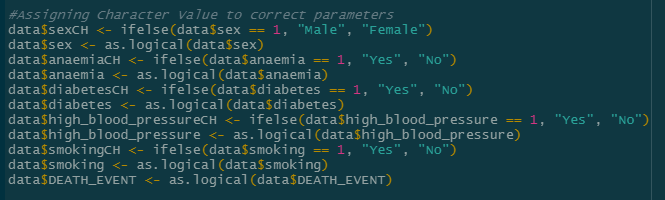
One major aspect of using my data was understanding the metadata and the specifics behind it. This required me to for the correct definitions of each variable and gather some understanding of what would be considered within the normal range for specific parameters. As such I had used external sources for variables that contained continuous values, to understand the data better and provide a point of reference when analysing the data. The metadata of the variables can be found in the RMD file with an explanation of what it is.

The breakdown of the Metadata can be seen here:

* Age – Age
* Anaemia - Decrease of red blood cells or haemoglobin (0 = No, 1 = Yes)
* creatinine\_phosphokinase - Level of the CPK enzyme in the blood
  + Normal Values range between 10 to 120 mcg/L
* diabetes - If the patient has diabetes (0 = No, 1 = Yes)
* ejection\_fraction - Percentage of blood leaving the heart at each contraction (percentage)
  + Normal Values have a percentage of 40% EF or higher, abnormal is below 40
* high\_blood\_pressure - If the patient has hypertension (0 = No, 1 = Yes)
* platelets - Platelets in the blood (kiloplatelets/mL)
  + Normal number of platelets is between 150,000 and 450,000 kiloplatelets/mL
* serum\_creatinine - Level of serum creatinine in the blood (mg/dL)
  + For men it is 0.74-1.35mg/dL
  + For women it is 0.59-1.04mg/dL
* serum\_sodium - Level of serum sodium in the blood (mEq/L)
  + Normal levels are between 135-145 mEq/L
* sex - Woman (0) or Man (1)
* smoking - If the patient smokes or not (0 = No, 1 = Yes)
* time - Follow up period (days)
* DEATH\_EVENT - If the patient deceased during the follow-up period (0 = No, 1 = Yes)

## Data Preparation and Cleaning

Due to the cleanliness of the data, no necessary work has been required. However, for me to appropriately identify the data types I had to alter the data to the desired data type for specific processes, this is primarily evident in figure 1:



This required me to assign to correct Boolean value for each variable. This required me to cross-reference with the metadata to ensure that I am assigning the correct variable to the data type, allowing me to perform certain evaluatory methods on the data.

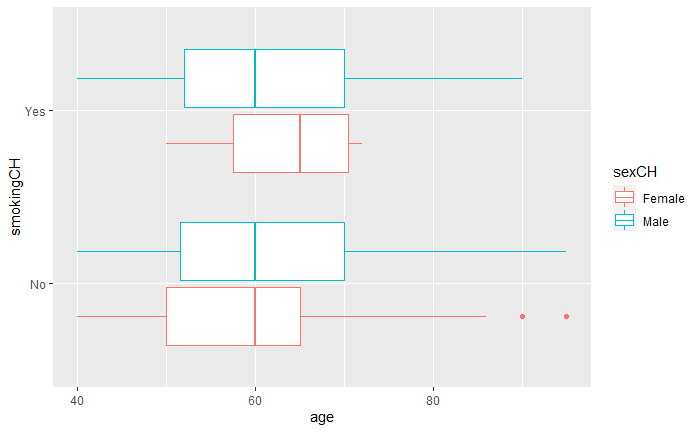
Another example of this is the preparation of data for the neural network which it had required me to scale the data to an appropriate size using normalization allowing me to maximize the accuracy of the deep learning implementation. This is evident below:

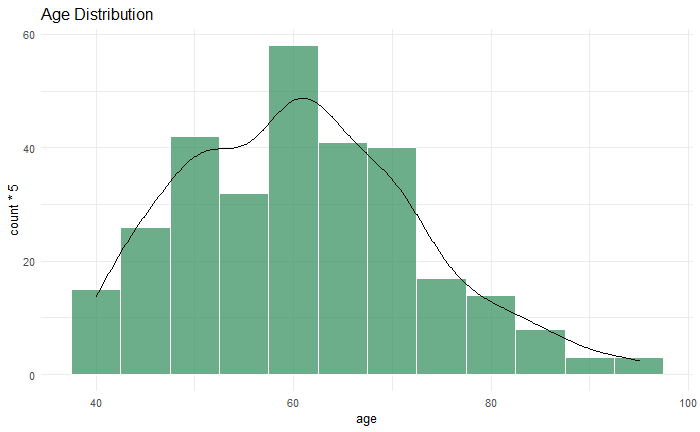
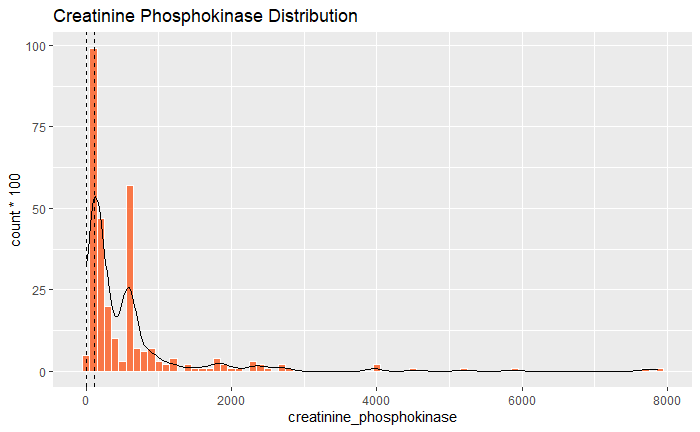
Text

Description automatically generated

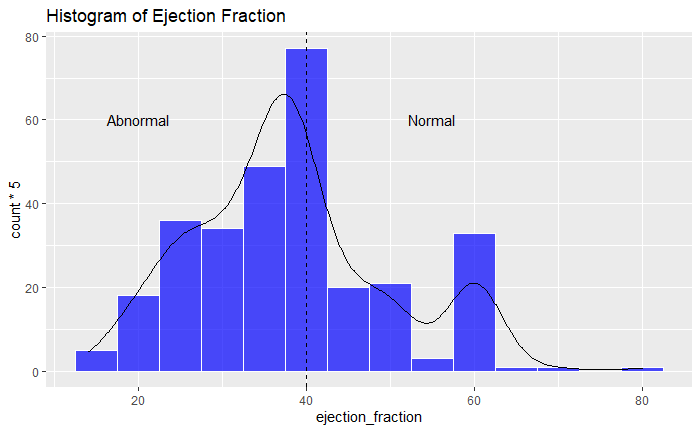
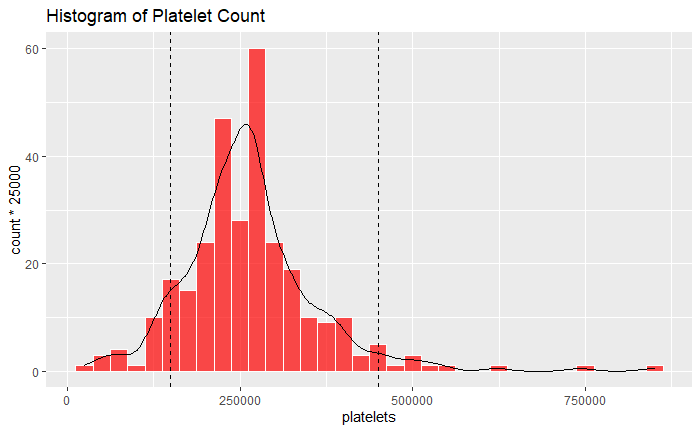
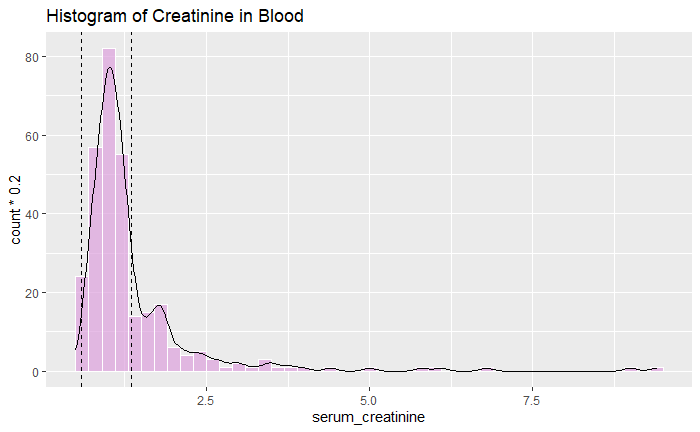
## Exploratory Data Analysis

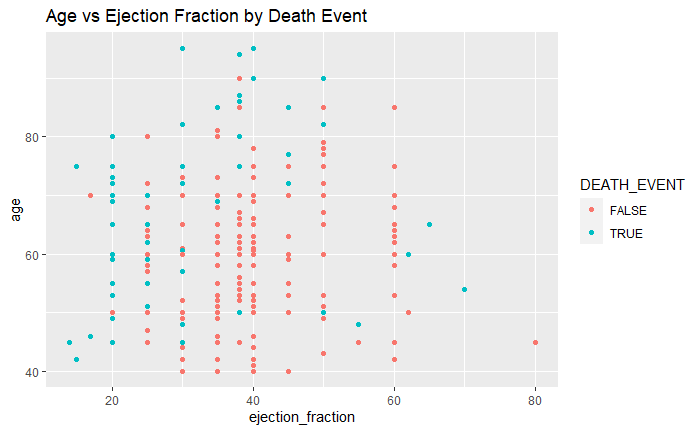
Figure 1
For me to understand the data in greater detail, thorough data analysis would be required. As such I am going to explore the data to be able to discover the potential causes of death due to heart attack and its relation to the circulatory system. As a result. I am going to use different techniques to help me understand the distribution of my dataset. To begin, I am going to look at different aspects of the population. As such I had initially prepared some basic count graphs to display some basic statistics:

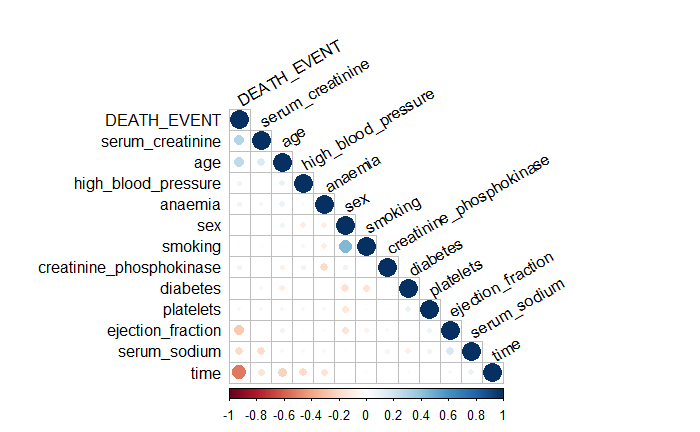
When looking at this chart we can see that eh distribution of the population is skewed in several directions, for example, the number of male cases overshadows the woman by almost a ratio of 2:1. This indicates that there may be some bias when training the model as it could cause issues with the system favouring results more towards males as well as those who do not smoke. One such example of this can be seen in the output box plot identifying the difference between men and women who smoke and their age. Due to the lack of dissimilarity between the two different demographics, we can identify that age and smoking do not seem to present much of any outliers. This had caused me to look at other continuous variables, this can be seen next.

Looking at the distribution of age, we can see that we have a fairly normal distribution with a bell curve formatting with the peak at 62 years. The skew is towards the left meaning that more extreme cases occur towards the latter ages, but due to the nature of the study those would are older are more prone to heart failure and thus should be included within the dataset as an example of edge cases that could occur. Removing these cases could prove costly as it would reduce the capacity of the model and its ability to adapt to the general population. When comparing this distribution to the number of creatinine phosphokinase within the blood, the data is skewed much towards the left side of the data. The two vertical dashed lines indicate normal levels of creatinine phosphokinase within the blood. There seem to be many outliers with the dataset, however, those could be explained by many different conditions with the data set such as obese patients or those suffering from 4rhabdomyolysis.

Furthermore, a large percentage of the population falls outside the normal levels. This can be attributed to the lack of general health which usually is the primary cause of heart failure.

When inspecting the next set of graphs we can see that the population tends to keep within the expected results, however, for ejection fraction, a significant amount is below the normal threshold of 40%. This puts them at risk of heart failure as the heart is not performing in optimal conditions. This is could be investigated further to check how it compares with death\_event.

When looking at Age vs Ejection Fraction we cannot identify a clear correlation. This is further supported by the scatterplot graph in figure 8.

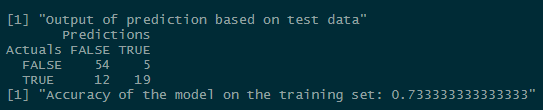
When looking at the correlation matrix, we can see that multiple different correlations stand out, both positive and negative. This can be identified through the hues ranging from a strong positive to negative correlation. From this many major correlations arise, especially that of age and serum creatinine, these seem to have the highest correlation with death\_event, as such high values in those two variables increase the odds of suffering from heart failure. However, a high level of sodium and a good ejection fraction reduces the risk of a death event occurring. The most evident correlation can be seen between death\_event and time. This indicates that the longer the patient went without a follow-up check-up, the more likely he was to not suffer from heart failure.

# Predictions

Bartosz has used a training split of 70% and 30% for training and testing respectively  
Tracy has used a training split of 80% and 20%. This disparity in the training levels can affect the fitness of the model due to the model being able to adapt to a wider array of results. This would lead to increased accuracy.

## Machine Learning Prediction

### Bartosz Napieralski

For my implementation of machine learning, I had decided to implement the usage of regression through the usage of a Generalized Linear Model. I had initially created a model that was then simplified using the steps() available in R. When using the GLM to predict the test data and compare it to the actual set, it had managed to score with an accuracy of 73.33%. This is further supported by the confusion matrix that displays the following results:

The GLM model was able to successfully identify 54 true negatives and 19 true positives, however, the model has identified 12 false negatives and 5 false positives. This is a worrying amount due to the model making a mistake 25% of the time.

To achieve my implementation, I had created a formula that utilized all of the possible variables within the data, the model would then be tested and simplified to remove noise and less impactful variables with weaker correlations. Graphical user interface, text

Description automatically generated

The usage of step() in R allows me to create a more robust formula for calculating the death\_event. This works by utilising the add1, drop 1 method and works by using the Akaike Information Criterion in an attempt to obtain the lowest possible AIC which indicates that the model is more parsimonious. This places importance on fewer parameters within the model. The final output can be found here:

Text

Description automatically generated

### Tracy Kimani

Text

Description automatically generatedOn the other hand, Tracy had decided to implement her machine learning using python. Her model also used Logistics Regression to create a model. However, Tracy’s model was more successful in obtaining a higher accuracy score as it was able to obtain an accuracy of 80% and 83.26% on the test and train sets respectively.

This could be indicative of the issues that a larger train set would have enabled a higher accuracy scoring when using the model to predict the dataset.

We can look at the classification report of the model here. This gives a better insight into how the model works, how we can compare it against my implementation later:

A screen shot of a computer

Description automatically generated with low confidence

## Deep Learning Prediction

### Bartosz Napieralski

For my implementation of deep learning, I had decided to implement an adaptation of an Adversarial Neural network. My primary goal was to develop a neural network with an appropriate number of input layers, and hidden layers that would lead to a singular output node for death\_event. This goal was achieved by using the Keras toolkit and SciKit-Learn. This model uses 3 layers with 10 nodes each. It also has 13 input nodes as it takes in all 13 different parameters that are available for each case.

Text

Description automatically generatedThis model is trained for 300 epochs/iterations to obtain the best possible accuracy whilst minimizing the loss. This is done to ensure that the best possible model is created that can cover the widest possible amount of combinations within the dataset. However, due to this being a NN, we first needed to scale the data to eh correct size to ensure that the Keras toolkit and NumPy would be able to process them correctly.

The accuracy of this model is far greater than the ML model that was created by me previously as it can score a 10% higher than before. This is evident below.

Text

Description automatically generated

### Tracy Kimani

For Tracy’s implementation of the deep learning method, she chose to use a K Nearest Neighbour classifier. This works by approximating the two possible events of death\_even being 1 and 0. It would then evaluate the position of the different cases and assign them to the closest node. This would determine the output. In Tracy’s implementation, she used between 1 and 30 centroids for the tests.

Text

Description automatically generatedThe recursion allows for the model to attempt many different outputs to identify the best solution and use that solution for the final output. This has the added advantage of producing better results at the cost of time.

Text

Description automatically generated with medium confidence

As seen here the accuracy of the model is 0.96. This is an excellent outcome as it only has 3 false negatives that occur within the entire test set.

Chart, line chart

Description automatically generated

The usage of recursion allows it to obtain a minimum error rate of 0.02666. This allows for minimizing the number of mistakes in the classifier. This can be seen in the chart on the left.

# Performance Evaluation and Comparison

Due to the limited number of implementations within our group, I will be mostly focusing on the comparison between ML predictions and DL predictions and identifying the advantages and drawbacks of each model.

To begin with, Tracy’s implement and mine both used linear Regression as the primary method. However when comparing the outputs. Mine had scored with an accuracy of 0.73 whereas Tracy’s implementation has a 0.8 accuracy. This can be attributed to many things but a potential one could be through different data splits where I had used a 70/30 split and Tracy had decided upon an 80/20 split. This difference in decision making has allowed her models to train on a larger dataset before being tested. This pattern continues in the DL implementations as Tracy’s implantation has scored an impressive accuracy of 0.96 whereas mine had managed 0.86. This difference is substant enough to warrant further investigation.

Another aspect that can be investigated is the F1 score of a model as this is the primary means of comparing how good two models are to one another. Going from best to worst:

* Tracy’s KNN implementation
* Bartosz’s ANN implementation
* Tracy’s Linear Regression implementation
* Bartosz’s GLM implementation

The F1 score is a combination of the precision and recall using a simple formula of . This measures how relevant the items that are retrieved through precision and recall measures how many relevant items are retrieved.

Another metric that the models can be evaluated upon is the execution time of the model. In this case. ANN implementation takes the longest due to having 300 epochs, this requires a significant amount of time to process in comparison to easier ML systems but it does offer higher accuracy in turn. However, due to the issue of this being run on multiple different environments, including Google Colab and local machines. This metric is also hard to compare due to the utilisation of different programming languages of R and python.

# Appendix

## Data Management Plan

The data will be obtained from online sources that are readily available for processing, this will most likely take the form of a ‘CSV’ file. The data is going to logical, ordinal, text and continuous data about cases regarding the aspect of the circulatory system and the occurrence of a heart failure. This data will most likely originate from a dataset sharing website like Kaggle as it is easily accessible and does not require many specialized storing requirements due to not possessing personal or sensitive information.

For the data to be understandable to others, I will be producing visualisation/graphics to help illustrate my ideas and point. The additional metadata will assist others in understanding what each variable means and its information for the event. The metadata will be included as annotations within the markdown file. This allows for easy access and referencing in case anything occurs. The naming scheme will follow standard procedures and the data will be stored locally and on the cloud to allow for both local processing and through the use of external platforms like Google Colab.  
To process the data, we will be utilizing both RStudio and Python3 with the person’s IDE of choice.

It will be our group's responsibility to manage the data whilst we are using it, however, if any issues occur, the data is available to be re-obtained from the original hosting website. The finding will not be posted in a journal or be published as they would require external verification. The data does not infringe on any patents or licensed technology.

## Author Contribution Statement

My contribution to this project involves the formulation of the question, performing an Exploratory Data Analysis that can be seen in this document and the creation of a Generalized Linear Model (GLM) as well as implementing of a Domain Adversarial Neural Network (DANN). I have also contributed to the evaluation of the different implementations provided by my group.

The only other member to have provided implementations of machine learning and deep learning implementations is Tracy Kimani (1605622) within this report.

The remaining members have not been able to provide an implementation of either method before 25/04/2022. The members who have not submitted their implementations for comparison are:

* Benyamin Jameeimoghadam (2049527)
* Shenene Jess (2049009)