Predicting the probability of patients suffering from heart disease passing away

Practical Data Science

Assignment 2: Data Modelling and Presentation

Shashwot Karki (s3841123) and Karan Pradhan(s3814520)

We certify that this is all our own original work. If we took any parts from elsewhere, then they were non-essential parts of the assignment, and they are clearly attributed in our submission. We will show we agree to this honor code by typing "Yes": Yes.

Table of Contents

Assignment 2: Data Exploration and Modelling

1. Abstract

This research document looks at numerous patients all of whom suffered heart failure and **aims to recognize distinct variables that would have greatly impacted their condition** but more importantly their lifespan/ mortality. Using a combination of machine learning algorithms and medical information we can obtain a concrete conclusion that can help us put a rest to the confusion. To help us get a solution to our problem we first acquired a dataset from the UCL Machine Learning Repository then came up with the following hypothesis to try and work with the dataset.

1) Age and gender would not be impactful factors in determining if a patient passed away after suffering from heart failure.

2) Patients with abnormal levels of Sodium Serum and Serum Creatinine are likely to have passed away

2. Dataset

The dataset was obtained from a research paper by "Survival analysis of heart failure patients: a case study" which can be extracted from the UCL Machine Learning Repository [UCI Machine Learning Repository: Heart failure clinical records Data Set]. The dataset contains the medical record of 299 patients who had heart failure which was collected during their follow-up period, where each patient profile has 13 clinical features which include:

- age: age of the patient (years)
- anaemia: decrease of red blood cells or hemoglobin (boolean)
- high blood pressure: if the patient has hypertension (boolean)
- creatinine phosphokinase (CPK): level of the CPK enzyme in the blood (mcg/L)
- diabetes: if the patient has diabetes (boolean)
- ejection fraction: percentage of blood leaving the heart at each contraction (percentage)
- platelets: platelets in the blood (kiloplatelets/mL)
- sex: woman or man (binary)
- serum creatinine: level of serum creatinine in the blood (mg/dL)
- serum sodium: level of serum sodium in the blood (mEq/L)
- smoking: if the patient smokes or not (boolean) - time: follow-up period (days) - [target] death event: if the patient deceased during the follow-up period (boolean)

3. Methodology

1. Retrieving Data

Before doing anything we must first import the dataset as well as the relevant packages required to process the data. We then proceed to use the read csv function in pandas to store the dataset into a variable df. We can then empoy the df.sampe() and df.describe() methods to check if our dataset has been imported properly or not.

2. **Data Cleaning**

Before doing modelling or exploration we must first check the data for errors as if we proceeded further without fist fixing them our results would not be accurate or reliable. First we checked if there were any invalid rows by using the isnull() method to check if any of the patients are missing any values. Then we use the df. drop_duplicates () to get rid of any duplicates. After looking at these we will check for errors for each individual patient. This includes things such as negative ages, typos, whitespaces etc.

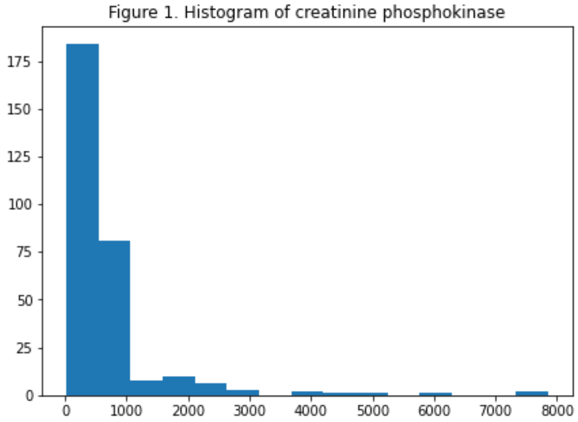
- We will check for negative values by using a for loop to go through the entire data to see if any of the values are lower than 0.
- We use df[name].unique() to check for any typos.

4. Data Exploration

- As there are numerous variables in the dataset it wouldn't make sense to look at all of them thus we will only analyze the variables that were left after performing hill climbing. The below displays the graphical analysis for individual variables as well as in conjunction with the target variable DEATH_EVENT.

Creatinine phosphokinase

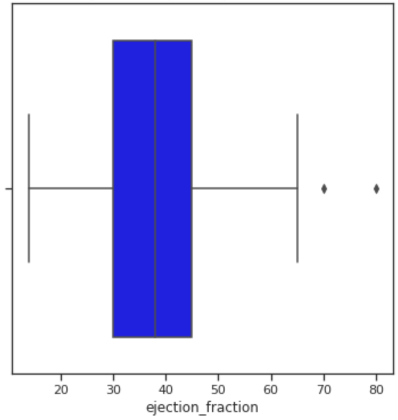
Figure 1. Histogram of creatinine phosphokinase



From the above histogram of creatinine phosphokinase we can see that for the majority of the sample proportion the value is between 0-500 mcg/L which is not very promising as a normal creatinine phosphokinase range I considered to be between 10-120 mcg/L.

Ejection Fraction

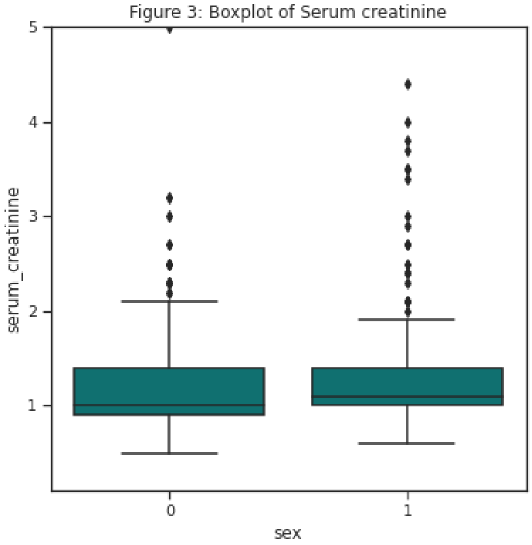
Figure 2: Boxplot of Ejection Fraction



From the above box plot of ejection fraction we can see that the median ejection fraction for the patients is around 38 percent which is much lower than the normal range which is between 50 to 70 percent

Serum creatinine:

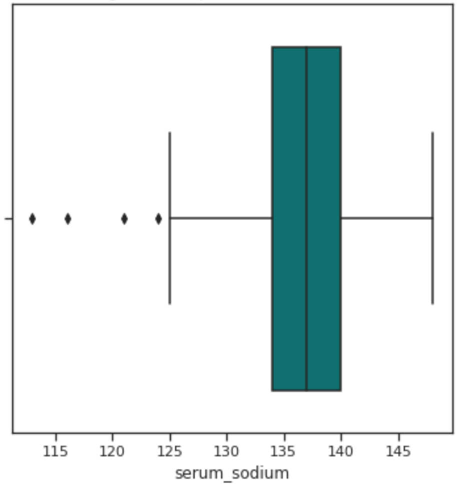
Figure 3: Boxplot of Serum creatinine



From the above box plot of serum creatinine we can see that the average level of serum creatinine for women was around 1 while for men it was closer to 1.2. Both of these values are fine as the normal range of serum creatinine for women is 0.59 to 1.04 while for men it is between 0.75 to 1.34. But from the box plot we can also see that both datasets have numerous outliers which are likely going to correspond with DEATH_EVENT.

Serum Sodium

Figure 4: Boxplot of Serum sodium

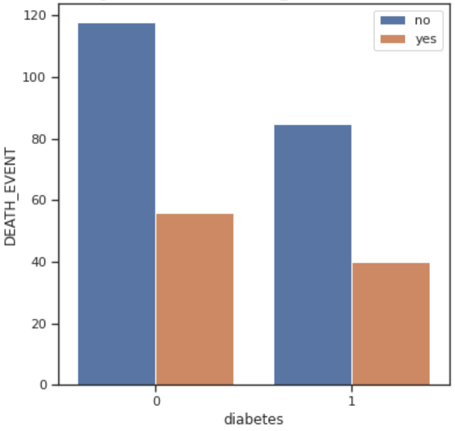


From the above box plot we can see that the mean level of serum_sodium for the patients was around 137.5 which falls right in the normal range which is between 135 and 145.

Relationship between target variable and other variables:

Diabetes and DEATH_EVENT

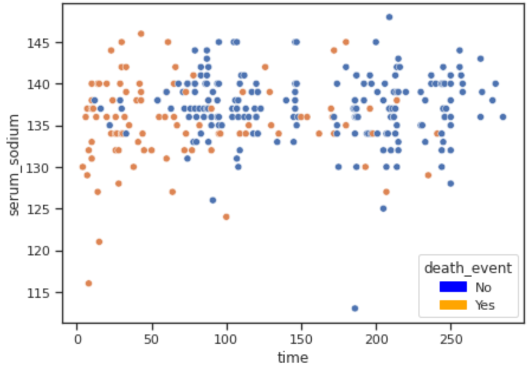
Figure 5: Box Plot of DEATH_EVENT vs diabetes



Diabetes is one of the more important variables and this is displayed in the bar chart. As we can see in the graph of the patients that suffered from diabetes the mortality rate(DEATH_EVENT) was around 33% while for the others it was closer to 25%.

Sodium Serum, Time and Death Event

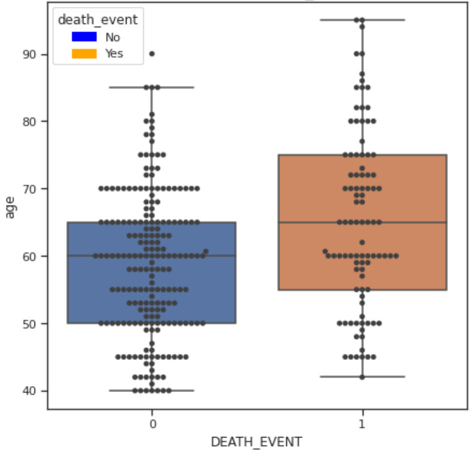
Figure 6: Box Plot of serum_sodium vs time



The graph shows that there is quite a strong correlation between time and DETH_EVENT as we can see that a majority of the patients that died fall between 0-50 days. There is a strong negative correlation between time and DEATH_EVENT as the lower the time the more likely it is that the patient passed away. But for the Serum Sodium we can observe that there is no real correlation as the data points are all over the place. The normal range of serum sodium in the bloodstream is between 135 and 145 mEq/L this further disproves our hypothesis as the graph shows that even for a majority of the patients that passed away they all had a serum sodium level that was normal.

Age and DEATH_EVENT

Figure 7: Boxplot of DEATH_EVENT vs Age



From the graph it is clearly visible that for the patients that did pass away on average were older. We can see that the median age for the people that passed away was around 65 while for the others it was 65. This shows that there is indeed a correlation between age and death event proving that our hypothesis of the age not impathing a patients mortality to be false.

Sex

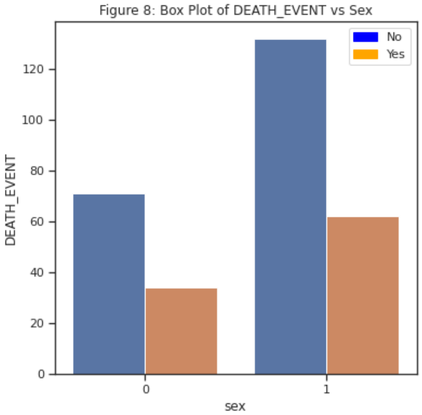


Figure 8: Box Plot of DEATH_EVENT vs Sex

From the graph we can see that there is no real correlation between sex and death event as the percentage of patients that passed away from each category is almost equal.

Serum Creatinine and DEATH_EVENT

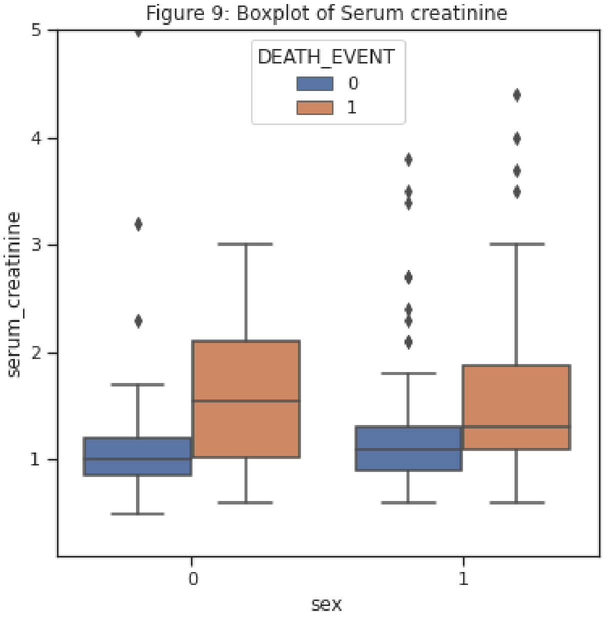


Figure 9: Boxplot of Serum creatinine

As previously discussed the normal range of serum creatinine for males and females differ thus they must be seperated by sex. For the females that passed away we can observe that their median serum creatinine level was around 1.6 mg/dL which is much higher than the normal range of 0.59 to 1.04 mg/dL. While for the males that passed away their median was around 1.3 mg/dL which falls in the normal range of 0.74 to 1.35 mg/dL. Although this result does not support our hypothesis it also does not necessarily disprove it. As the graph displays that abnormal levels of serum creatinine resulted in death for females while for males this was not ture. Thus, no reasonable conclusion can be reached.

5. Data Modeling:

Abstract:

Following the data exploration we now move onto the modeling where we have employed machine learning algorithms to conduct a k-Nearest Neighbour Classification as well as a Decision Tree classification. All of the modeling/ classification will be conducted with the target variable ‘DEATH_EVENT’ as the focal point.

1.k-Nearest Neighbour Classification.

2.Decision Tree Classification.

We have done two k-Nearest Neighbour classifications: initially with our base model where we just took all of the given features without filtering anything. Then we employed a hill climbing algorithm for feature selection which resulted in a better precision for our prediction model.

k-Nearest Neighbour:

We started off our modeling by implementing the k-Nearest Neighbour classification as it is a simple and widely used classification technique with numerous applications.

For the parameters in the k-Neighbour classifier we initially loaded some default parameters then proceeded to tune it changing the values of n_neighbors to 12 and then the weight to ‘distance’ and then the leaf to 50 and so on. The k value is very optimum as we found it by Hyperparameter tuning as explained later.As of the other parameters I have used its default values.

The parameters were tuned as adjusting the parameters increases the accuracy for our model which results in high quality results.

Decision Tree:

Decision tree is a supervised classification which much like k-Nearest Neighbour is another commonly used machine classification model. It is a flowchart like structure where the path from the root to the leaf are the classification rules

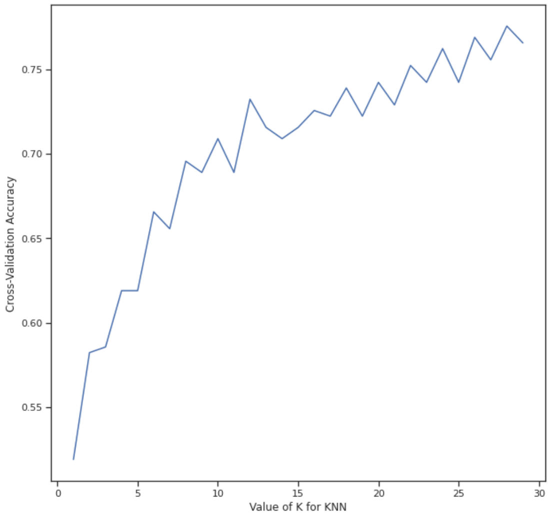
Unlike the k-Nearest Neighbour the decision tree is more difficult to deal with as the number of parameters we deal with are very large and additionally the range of possible values are also very ambiguous. For our model we employed only a select few parameters such as the criterion which accepts string values ‘gini and ‘entropy’. We used the default value ‘gini’ as the gini index executes binary splits in the tree; thus, the more the gini index the higher the homogeneity. The other parameters employed were max_depth which indicates the depth of the tree as well as random_state, split etc.…

Hill Climbing Algorithm:

Initially we did a classification where we used all of the given features without using any sort of algorithm for feature selection. Then we proceed to use the Hill Climbing algorithm where we selected features and only worked with around 9 features for better accuracy. This showed us that some of the features given were not very significant for our model and neglecting those features results in a more efficient model.

After the algorithm has been employed we then select the new features obtained through hill climbing to make a new dataset set. In our case we had a total of 12 features excluding the target and after selection we were left with 9.

Figure 10: Cross Validation accuracy vs value of K



Hyperparameter tuning:

Hyperparameter tuning is a technique where we loop the value of k from 0 to a certain number. For us we used a value 30 and then applied the cross validation technique to get the best value of k for the model. We could observe that the best k value was k=12 .

K-Fold Cross Validation:

We then used another technique that is the k-fold cross validation which splits the data into **k** folds (parts) and uses each part as a test set while the rest is used as a train set. This gives us a more detailed evaluation of our model as it uses all the data in our data set for both training and testing. We then get the results for each fold at the end of the process.

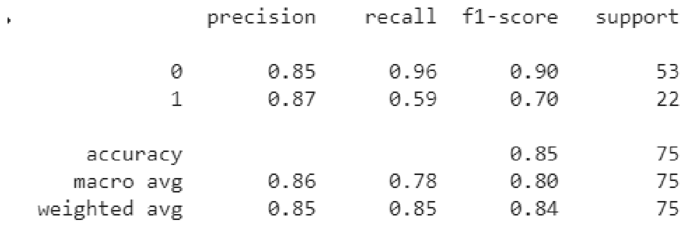
Results:

Train and test data:

As mentioned earlier we have two models, and for both of them our train is 75% and the test is 25% respectively.

k-Nearest Neighbour:

For our 75-25 split which had selected features we achieved an accuracy of 85% model. While the base model had an accuracy of 56% this clearly shows us that the hill climbing and parameter tuning has made a huge difference in the overall quality/ accuracy of a model.



```
              precision    recall  f1-score   support

           0       0.85      0.96      0.90        53
           1       0.87      0.59      0.70        22

    accuracy                           0.85        75
   macro avg       0.86      0.78      0.80        75
weighted avg       0.85      0.85      0.84        75
```

Figure 11: classification report of the main KNN model.

From the above classification table we see that our precision for the death event to occur is 79% while the death event not to occur is 88% with an average precision for both between 83 to 85 percent.

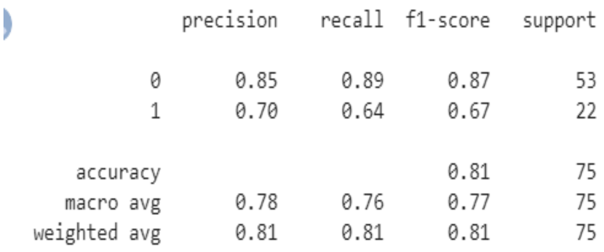
As we recall there is 68% accuracy for the person to die and 92% that the person will not die with an average of 80- 85 % accuracy for both.

As for the F-1 score it tells us that the chances of death occurring is 73% while the chances of it not occurring is 90%with an average of 85% for both occurrences.

Decision tree:

For the decision tree the gini index was used to split the node into the most corresponding sub-nodes. Parameters were also provided to prevent overfitting of the given data.

For our decision tree we had almost the same accuracy for both the base model and our main algorithmic model, although the base model for the decision tree did have a slightly better accuracy of 83% compared to 81% of the main model which is not that significant to be taken into consideration.



```
              precision    recall  f1-score   support

           0       0.85      0.89      0.87        53
           1       0.70      0.64      0.67        22

    accuracy                           0.81        75
   macro avg       0.78      0.76      0.77        75
weighted avg       0.81      0.81      0.81        75
```

Figure 12: Classification report of the Decision tree of the main model.

For precision, 70% of people are likely to die while 85% of people will not die.

As for recall we see that 64% people are likely to die while 89% people will not die.

For F-1 we see that 67% death event is likely to occur and 87% the death event will not occur.

Figure 12: Decision tree graph

Discussion:

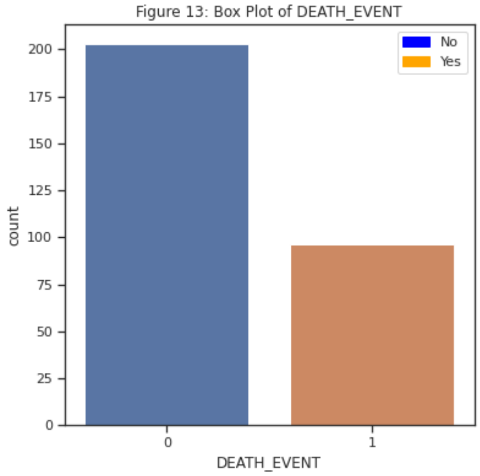


Figure 13: Box Plot of DEATH_EVENT

Even though all the modeling and classification has been done you cannot help but realize that our results are not that accurate due to a bias in the dataset. This can be observed in the boxplot of DEATH_EVENT below. We can see that a majority of the patients did not pass away due to which a lot of our results will be in favour of the patients that are still alive. This also means that the conclusion we drew would be biased and not very accurate. We realised this at the later stages of the project after a majority of it was completed thus we couldn’t implement changes. If we were to do the project again we would have undersampled to achieve a more accurate result.

Conclusion:

In the project as we only employed two classification algorithms both of which were supervised classification there is only a limited amount of conclusions we can draw about the quality of the models. But we can definitely conclude that the better classification algorithm was KNN with an optimal k value of 12 as it produced better results as well as high precision/ accuracy. Additionally in terms of our two hypotheses:

Age and gender would not be impactful factors in determining if a patient passed away after suffering from heart failure.

- The exploration phase showed us that age was a highly impactful variable in determining if a patient would pass away or not after suffering heart failure. While gender on the other hand wasn't that impactful at all.

Patients with abnormal levels of sodium serum and serum creatinine are likely to have passed away.

- The exploration phase also showed us that there was no real correlation between sodium serum and DEATH_EVENT. As for serum creatinine no definitive conclusion could be reached since we learnt that normal levels of serum creatinine is dependent on sex. And as male and females showed different results no real conclusion could be reached.

References

- **Practical Data Science Notes – by Prof. Yongli Ren**
- **https://pandas.pydata.org/**
- **https://archive.ics.uci.edu/ml/datasets/Heart+failure+clinical+records**
- **https://matplotlib.org/**