Analysis of Heart Disease Data

Project Report

presented by
Club der toten Dichten (Team 12)
Finn Hülsbuch, 1913864
Thilo Dieing, 1692328
Lasse Lemke, 1914420
Eric Jacquomé, 1903834
Timotheus Gumpp, 1913876

submitted to the
Data and Web Science Group
Prof. Dr. Heiko Paulheim
University of Mannheim

December 2022

Contents

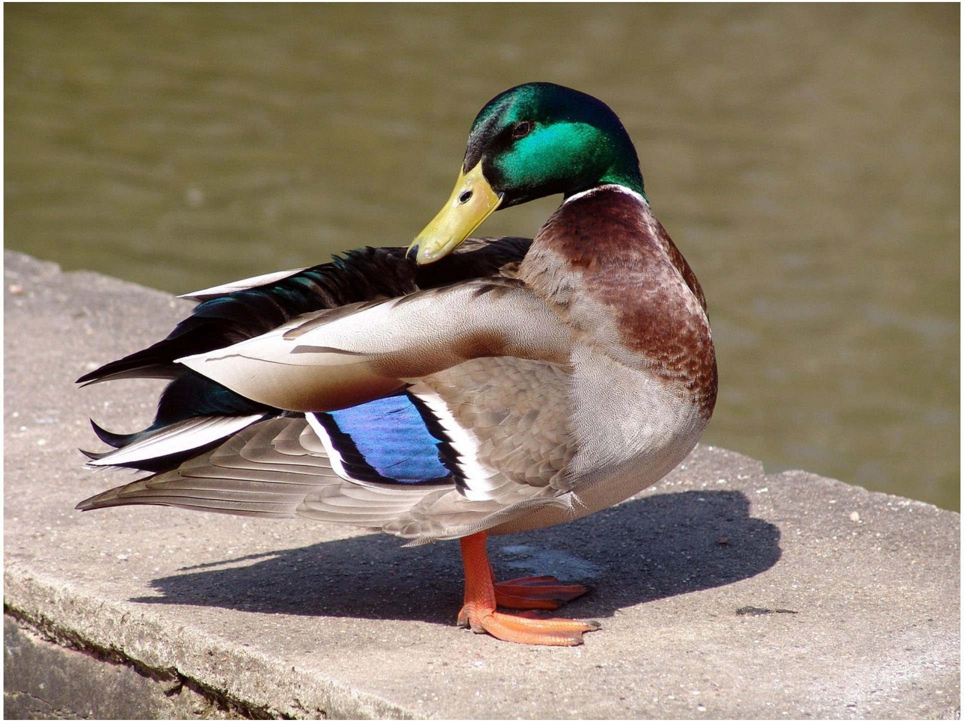
1 Application Area and Goals

Heart disease is currently still one of the highest causes of mortality on earth (**nahar2013**; **kavitha2016**; **statistischesbundesamt2020**). Given the successful application of data mining in other sectors e.g. banking and finance or marketing (**keles2017**) possible applications in the medical industry are plentiful. Yet the healthcare sector is information rich but knowledge poor (**soni2011**). According to **soni2011** medical data sets provide great potential for data mining to be used in clinical diagnosis.

This aim of this project was the application of data mining methods, more specifically classification methods, to predict whether or not a patient could suffer from a heart disease. The successful application could help doctors and medical staff with diagnosing patients by automatically analysing historical test result data of the patient and give a prediction when a higher potential of heart problems arise. By doing this analysis patients flagged for potential heart disease could possibly be prioritised. Due to the immense amount of stress and long working hours medical personal are facing, having an additional instance looking at the data could be beneficial. In the past such approaches have already been tested and proven to be a good diagnostic option (**usharani2011**). **jabbar2013** state that data mining techniques answer several important and critical questions related to healthcare and that they can improve the provision of quality services to patients.

This project report is based on the “Heart Disease Data Set” (**janosi1988**) which, despite its age is still relevant given the fact that it consists of results of medical tests. In addition to that the validity is assumed because it is frequently used in contemporary research (see **usharani2011**; **aha1988**; **nahar2013**).

Hallo Lasse. So kannst du Bilder in Latex anzeigen.



Da du aber wahrscheinlich ja eher Abbildungen machen willst versuche es hiermit.

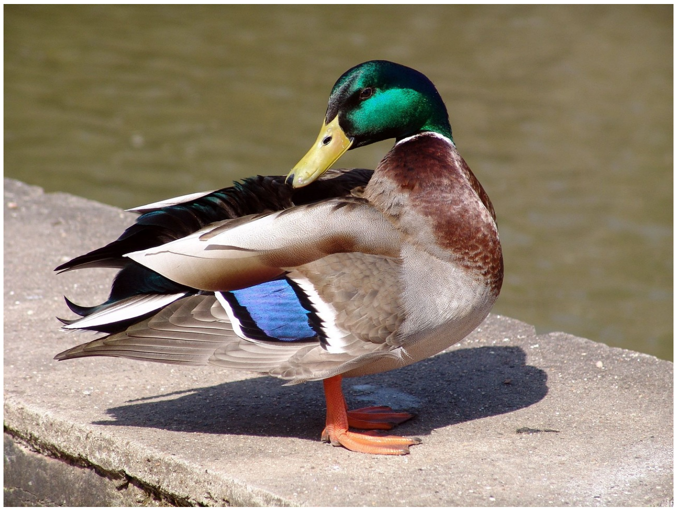


Figure 1: This is a smaller duck

2 Structure and Size of the Dataset

2.1 Structure of the Dataset

2.2 Size of the Dataset

3 Preprocessing

We approach preprocessing in two steps. In the first step we clean the data set based on knowledge we obtained in the Data Understanding section and further analysis. Secondly, we try many different approaches to transform the data, where we can not be certain about the best method.

3.1 Cleaning

Firstly, we implement a custom loading function to transform the 4 datasets into a csv format, so we can use it further on.

We remove the false predictors lmt, ladprox, laddist, diag, cxmain, ramus, om1, om2, rcaprox and rcadist, as our target variable num is a combination of these according to the UCI.

Furthermore, the features restckm, exerckm, thalsev, thalpul, lvx1, lvx2, lvx3, lvx4, lvf, dummy and junk are considered irrelevant by the UCI, so we also drop them. Other irrelevant attributes we remove are IDs(id,ccf), constants(name, earlobe) and dates of medical examinations (ekgmo, ekgday, ekgyr, cmo, cday, cyr). We consider these dates irrelevant because we assume that the date of an examination does not affect its outcome.

We drop the feature pncaden because it is the sum of painlox, painexer and relrest and therefore contains no additional information.

The features cp, restecg, slope, ca and restwm were oneHotEncoded as they represent categorical values.

When checking for inconsistencies between features, we detected that thaltime is sometimes lower than thaldur. As thaltime describes the moment ST depression is noted within the exercise, it has to be lower than the duration of the exercise thaldur. We replace thaltime by NaN in all 17 instances that do not satisfy this criterion.

Also, the maximum heart rate (thalach) was replaced with NaN if it was lower than the heart rate at rest (thalrest).

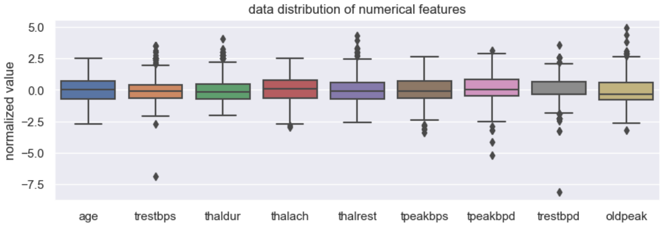


Figure 2: data distribution of all numeric elements

As shown in **??** we created a normalized box plot of all numeric features to check for outliers. The features trestbps and trestbpd show extreme outliers with a value of 0. These are incorrectly specified `NaN`s and are therefore replaced by `NaN`. All other outliers are not as extreme and come in groups. As the data contains sick persons, values diverging from the norm are expected. For these reasons we decided to keep these outliers as they can be a strong indication of a heart disease.

The remaining features were analysed regarding their pearson correlation. Only two pairs of features with substantive amount of data (less than 75% `NaN`s) have a very strong correlation (80%).

These are cp_4↔painexer and rldv5↔rldv5e. The highest correlation is between cp_4 and painexer. The feature cp_4, which was oneHotEncoded from the categorical variable cp, describes whether the patient has no chest pain at rest. Painexer describes whether the patient only has pain when exercising.

Concluding from the high correlation between the EKG amplitude when resting (rldv5) and the EKG amplitude when exercising (rldv5e), we decided to create a new feature(rldv5_diff) by using the difference between these. We did the same with resting heart rate and maximum heart rate (thal_diff = thalach - thalrest).

Furthermore, we enrich the feature smoke using years and cigs. Hereby, we reduce the number of `NaN`s from 74% to 43%.

3.2 Hyperparameter Tuning

For hyperparameters and methods where we could not be certain, we try out multiple different combinations.

Firstly, we try binning the feature age. We choose either 2 or 5 bins or no binning at all. We decided to use equal width binning so that the age groups are simpler and more intuitive to a doctor.

Diagram **??** shows the number of features, that have less than X% missing data and how many cells we would need to impute if we included them. It becomes apparent that there are certain steps where the number of features goes up a lot. To decide when a feature is included based on the number of missing values, we try the steps 0, 4, 8, 20, 35, 60, 75 and 100 % in the model. They are shown as vertical lines in the graph. Additionally, we decided to drop features based on their correlation. For this we decided to use the steps [X,X,X,X].

To impute the missing data we use a simple imputer. Missing values are replaced by the mean, median or mode of the feature. We decided against using a KNN imputer, because it is computationally much more expensive. The iterative imputer is not used, as it is still experimental and therefore subject to change.

To account for the different ranges of the features different scalers are tried out. We only use scalers that are applicable for all integers as some features contain neg-

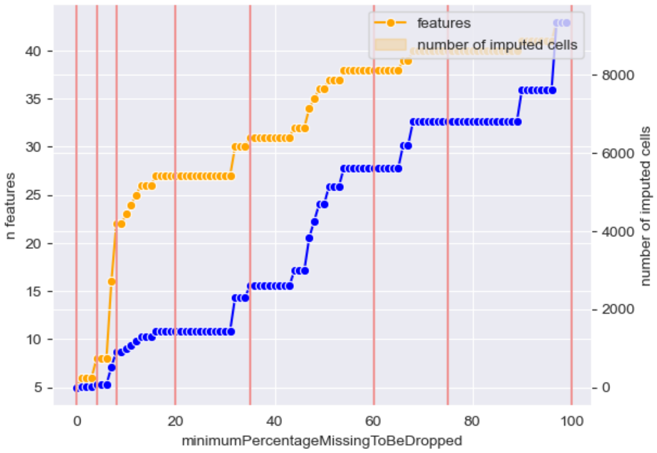


Figure 3: Number of features and values to be imputed by number of `NaNs`

ative values. We compare the MaxAbsScaler, MinMaxScaler, PowerTransformer, RobustScaler, Standardscaler and Normalizer. As the hyperparameters of most scalers turn on or off core functionalities of the scaler, we decided to only tune the hyperparameter norm of the Normalizer with the norms l1, l2 and max.

To account for the different amounts of healthy and unhealthy patients we try oversampling and undersampling in the training data in comparison to passing the values through.

4 Datamining

5 Results