Data Mining and Decision Systems  
600092  
Assigned Coursework Report

Student ID: 201602395  
Date: 07 October 2019

## Due Date: 12 December 2019

CHECK WORDS

**Our, my, we,**

# Methodology

Provide details on the methodology applied towards the data mining analysis undertaken, providing rationale for these steps.

This should detail how you went from the raw data provided to the chosen model(s), choice of model, and how this methodology helps address the problem domain.

Evidence to support the following of this methodology should be presented, especially any cases which required moving backwards in the process to readdress issues.

A large set of legacy data from a domain of cardio-vascular medicine has been provided to be understood and handled using industry tooling.

CRISP-DM methodology has been used to complete this task by filtering, cleaning, and transforming the data as appropriate such that it can be used to produce optimal classification for patient risk of dying.

### Business Understanding

The first step completed was to gain a business understanding of the data. However due to the lack of a client within the domain, gaining an accurate business understanding proves difficult. Therefore, predictions were created using external knowledge of the domain and assumed as correct.

It is assumed that the main objective is to produce a system which can predict and aid medical professionals in determining a patient’s mortality risk based on entered patient data.

### Data Understanding

The next step was to generate an understanding of the data present. Usually, using the business understanding, only the appropriate data attributes will be collected and used for a system such as this. However, due to the lack of a client to provide any additional data or information, this project will rely fully on the provided data.

Below is a description of the data and sample can be seen in section 1.0 of the code.

The data provided consists of over 1500 patient records each containing 11 attributes describing factors of the patient’s health and numerical data such as ID. The first attribute ‘Random’ displays a unique randomly generated number. This is used to help in randomly sorting the data. Next is the ‘ID’ column which is a unique identifier for each patient. The third attribute ‘Indication’ is a nominal value expressing the cardiovascular event which triggered the hospitalisation. The values included in this attribute should only be one of the following: “A-F”, “ASX”, “CVA” or “TIA”.

There is then a series of Yes or No features depicting factors of each patient including whether the patient suffers from Diabetes, IHD (Ischemic Heart Disease or otherwise known as CHD (Coronary heart disease)), Hypertension, Arrhythmia or if the patient has a history of cardiovascular interventions.

The next two attributes ‘IPSI’ and ‘Contra’ both display integers indicating the percentage figure for cerebral ischemic lesions and contralateral cerebral ischemic lesions accordingly.

Finally, the last attribute ‘label’ indicates the patient’s mortality risk, indicated by “RISK” or “NORISK” only.

### Data Filtering and Cleaning

Once a reasonable understanding of the problem was created, the next step was to go through the entirety of the data, filter it and clean it. This involves removing any erroneous errors, transforming data from one format to another and removing irrelevant data. It was found that the best way of doing this was to go through each attribute individually, inspect its data members and compare it to the data understanding. This can be seen in section 2.0.

Before checking each attribute, the number of NaN values is counted to check how complete the data is. Within the data, only 17 NaN values were present, so it was decided that these records would be dropped as attempting to fill them could carry any present errors forwards.

From the data understanding it is known that the attribute ‘Random’ should contain a unique random number for each patient so that all patients can be ordered randomly. However, after inspecting the data in section 2.2 it can be seen that certain values of Random are repeated up to 4 times. This means the attribute does not perform as its supposed to and regards it useless. A possible reason of this could be that when a professional enters their patient’s data into the system, this record isn’t immediately shared with all other professionals, and therefore another doctor could generate the same random number.

After inspecting column ‘Id’ in section 2.4 it can be seen that no duplicates are present and that each value is truly unique. It can also be seen that there seems to be no order with the Id and therefore is essentially random further proving that the attribute ‘Random’ has no use in this system. From these findings it was decided that both attributes ‘Random’ and ‘Id’ are to be dropped from our data set as they have no impact toward the prediction of risk of mortality.

The next attribute to look at is the ‘Indication’ attribute. As seen in section 2.5 there are 5 different values present for Indication in the data. This does not match what is stated in data understanding. By looking at the values present we can see we have two repeating values, “ASx” and “Asx”. Using the data understand it can be assumed that these values are identical but have been entered into the system with different capitilisation. To solve this and prevent any similar error throughout the entire data set, every attribute containing a string was changed to fully uppercase.

Furthermore, to avoid any confusion or unfair weighting later on in any models made, One Hot Encoding with be carried out on this nominal attribute to split it into 4 separate Boolean values.

The next 5 attributes (Diabetes, IHD, Hypertension, Arrhythmia and History) all contained either “YES” or “NO” as their value. This is fine when a human is reading and understanding the data, however, models and algorithms will not know to take these values as Booleans and will instead treat them as nominal string values and give them order. To solve this, all these “YES” and “NO” values were changed to “True” or “False” and are now considered Boolean values.

When inspecting the attributes ‘IPSI’ and ‘Contra’ in section 2.7 and 2.8, it is shown that IPSI displays an array of valid float values. However, Contra displays an array of integers with one empty value. Upon further inspection, by looking in section 1.0, it is discovered that due to this empty value, the entire Contra attribute is considered an object rather than an integer. To solve this the empty value’s entire record was removed from the data set and the Contra attribute was changed to int64.

The last attribute to clean is the ‘label’ attribute. Inspecting the values within ‘label’ in section 2.9 it is shown that it contains 3 different values: “RISK”, “NORISK” and “UNKNOWN”. This is invalid data when compared to the data understanding that states that label should only contain “RISK” and “NORISK”. Methods or models could be implemented to fill these “UNKNOWN” values however in the prevention of carrying any error forward, these “UNKNOWN” records are changed to NaN and dropped.

### Data Modeling

Data Mining Models are virtual structures used to represent predictive analysis on a set of data. Multiple models have been created that learn from our cleaned data and are able to predict a patient’s risk of mortality. However, the data still needs to be slightly manipulated in order to pass to any models.

First the data is split into X and Y values as seen in section 3.1. X being the dataset that contains all attributes of our cleaned data apart from label, and Y being the values in label as this is the value that is being predicted.

Using these X and Y datasets, training and testing data can then be generated. Since no client is present to request additional data from to test our models, the dataset is all that is available. Therefore, using X and Y, training and testing data is created at a 70%:30% split.

#### K-Nearest-Neighbors Model

The first model used is the K-Nearest-Neighbors Model (KNN). The KNN model is a simple algorithm that when given a piece of test data, will work out the distance between this test value and all the training values, collects the specified number of values (K) closest to the test data and conducts a majority vote of these values to determine the test data’s Y value. The KNN model has been chosen due to its ease of interpreting its output and its low calculation time.

The first step to implementing a KNN model is to determine a value for K. This has been done by simply implementing a KNN model with every value of K up to a certain point and compare the accuracies of each. Another popular method of determining K is to set it equal to the square root of n, n being the number of datapoints in your dataset. In this datasets case, using this method, K would equal 32.

As shown by figure 1, the KNN model is most accurate when K is set to 1. However, using small values of K is ill-advice as it makes your result susceptible to outliers. On the other hand, using larger values of K will have smoother decision boundaries which will result in a lower variance and an increase bias. It is also best if K is an odd number to prevent a tie when the majority vote is conducted. Taking all of this into consideration a K value of 31 was selected for the KNN model.

#### Decision Tree Model

The second model used is the Decision Tree Model. A decision tree is a flowchart like structure in which each node is our dataset split according to a certain parameter. The Decision Tree Model has been chosen as the steps the model has taken to produce an outcome can be followed by following the tree nodes and leaves.

#### Multi-Layer Perceptron

The third model used is the Multi-Layer Perceptron (MLP).

We can test a model using K-fold and give a more accurate representation of how model will perform on unseen data.

USE FOR IMPROVEMENT OF MODEL INSTEAD.

The second method implemented is to again implement a KNN model with every value of K up to a certain point, however, instead of using the training and testing data created earlier, we use k-Fold Cross-Validation. (KFCV) KFCV randomly splits our data into F number of groups, with one of the groups being the test set and the rest are used for training. The model is trained on the training data and scored on the test data, then the process is repeated so that each group is used for the test data once. Using this method provides a more accurate representation of how our model will perform on unseen data.

ALL MODELS COMPLETE, RE LOOK AT DATA UNDERSTANDING

HEATMAP, VARIABLES ARE NOT DIRECTLY COORELATED

DROP AND RE RUN MODELS.

# Results

|  |  |
| --- | --- |
| Figure 1. | Figure 2. |
|  |  |
| Figure 3. | |
|  | |
|  |  |

Results should include tables showing model performance with appropriately selected metrics. No rationale should be provided for this section - simply results of evaluative processes.

If using modified variants of the dataset, these should be clearly identified in the tables with appropriate naming. The justification and description of modification is not for this section.

Additional figures may be used as appropriate, in support of discussion points in the Evaluation & Discussion section, or as evidence for methodology following above.

# Evaluation & Discussion

Evaluation methodology used for generating the results provided in the previous section. How were these evaluated? Why was this selected? What metrics were used and why?

Discussion of the results should be presented with appropriate evidence and rationale. E.g Which is the best model, and why?

Consider each stage in the methodology, and reflect on any improvements which could have been made. Could any techniques have been used which may have improved performance? Why?

No client, so Business understanding may not have been accurate.

Could have normalised percentage values in data

# References

Any references used throughout the report should be included here in Hull Harvard Style. If no references used, remove this section.