Data Mining and Decision Systems  
600092  
Assigned Coursework Report

Student ID: 201707824  
Date: 15th November 2019

## Due Date: 12 December 2019

**Report must be within 8 page maximum. Strict page limits will be enforced. Any extra pages will be ignored and no marks awarded for any work on these. Exclusions to this limit are the front page, the references section, and any appendices. Please keep to the given section headings and format; subsections are permitted.**

# **Methodology**

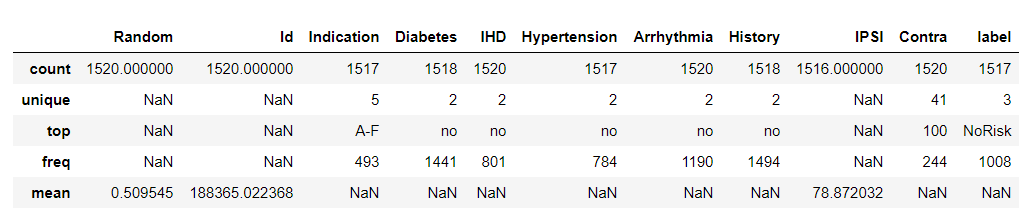
The methodology followed for this report will be a slightly modified version of the CRISP-DM methodology. In this instance, we do not have the first and last stages of the methodology, which include the **business understanding phase** and the **deployment phase**, so these will be included in a hypothetical sense rather than a more general sense.

The task is to create a classification model with the dataset supplied. The data is provided via the domain of Cardio-Vascular medicine. The aim of this is to identify if a patient is at risk or not at risk. The model will be of a binary nature as we are working with an if or else classification, also known as a 0 or 1 classification mode (binary classifier).

### **Business Understanding Phase**

### **Deployment Phase**

### **Data Understanding Phase**

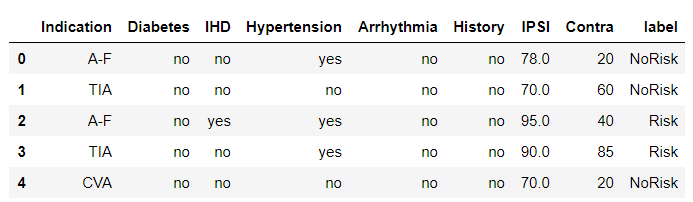
The process of Data Understanding is to Describe and Explore the data. This is done by observing the differences between what is in the Data Description, and then what is present within the data. These observed differences are then compared and adjustments are made. The adjustments must be justified however, as it is important to maintain data integrity throughout this process.

Early exploration shows that the total values for the dataset equate to 1520, with mostly categorical attributes. Thankfully in the case of this dataset, it closely matches its data description, with no real differences that are worth noting. Missing values can be seen to be present, evidenced by differing column value maximums and will be the focus of the data cleaning from the offset. Calling the head function of pandas allows initial exploration of the data to clarify everything discussed so far is accurate.

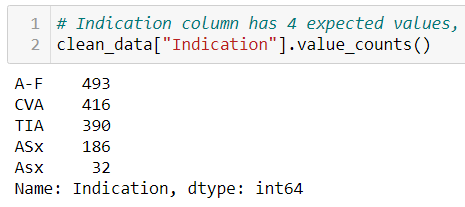
In the data given, most values are noticeably similar, most of the attributes with value type nominal are classed as objects within the data frame. Random is a float and ID is an integer, which correlate correctly. IPSI is a float while expected to be an int which is fine, but Contra is an object, this attribute should be an int, as it contains numeric values. This can be changed through code in the next phase to instead be a float through the pandas **to numeric** function.

### **Data Preparation Phase**

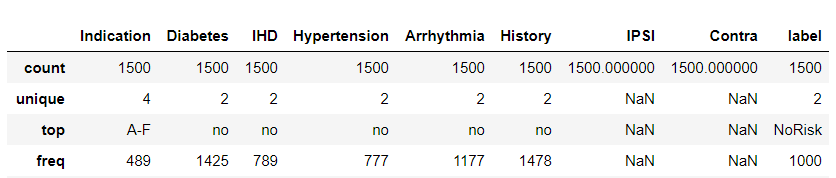
The first thing to do is to select the data to use, and the data to exclude.

For this task, all features will be used besides Random and ID. This is because, while useful for indicating individual patients and their scores, the attributes themselves don’t have suitability towards the end goal and therefore are the features for exclusion. As the model will be of a binary nature, these columns indicate high numerical values and may cause incorrect classifications later.

The data is cleaned in steps, looking at the patterns inspected from the earlier phase and putting these into practice. This can be seen in more detail through the code found within the accompanying notebook, specifically the Exploratory Data Analysis section, which covers these. The data will be changed in places where it is deemed necessary, starting with the label category. EDA, or Exploratory Data Analysis is an approach to analysing datasets to summarise their main characteristics, often through visual methods.

Any attributes found to have values that don’t match the data description are investigated, and anything found foul is removed. Seen in the code, the extra value for label is replaced with nan, and the 5th unique value of the attribute Indication is checked and corrected. The value within label must be replaced to a numpy value of nan rather than simply being replaced by nan because the simple replacement led to the value not being dropped later.

There is always the argument of imputing missing data or simply removing it. This depends on the amount of data loss, however. In code block [15] the null data is explored and is found to equal 20 values. Considering we have 1520 expected values and only 20 contain a null value, I decided to drop these values. There is an argument for imputation here, but due to this data being legacy, and the percentage of values missing being low (code block [16]), the data was removed instead.

Once these duplicates have been removed, its important to conclude by making sure all values are now of the correct data type. Contra which was discovered earlier to be in the wrong format, is changed to be of a numerical type float64 using pandas at code block [14]. Simply changing Contra to be numeric caused an error however, which was resolved by applying errors=” coerce” to the code. This meant that there was an empty value that wasn’t being picked up and the conversion was failing due to this.

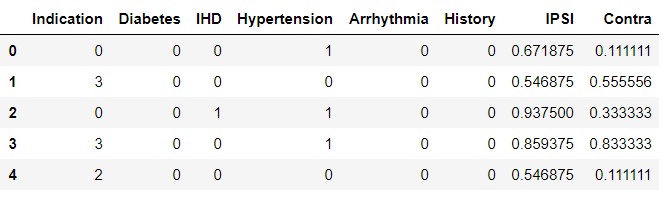
#### **Visualisation**

Visualisation is a tool that can be used throughout the methodology to explore the data in a more visual sense. CRISP\_DM is great because it is flexible and allows back and forth movement. Where Visualisation is seen as an area of Data Understanding, the methodology allows for the ability to be able to revisit older phases to explore the data to a further degree of detail. Being able to look at the data in this way helps to find links which would take longer to find simply scrolling through the data.

Visualisation is used to find trends in the data and clarify suspicions I have within, regarding which attributes are of more use to the end goal than others. This can be seen in the code section Visualising the data. I believe from an early state that the attribute Indication, alongside Contra are of particular use towards the end goal. Visualisation is used to clarify this.

#### **Data Formatting**

Data Formatting is necessary to ensure the data is of the correct format that models can use later. The two techniques I chose to use for data formatting were techniques called Normalisation and One-hot Encoding.



Normalisation is also known as feature scaling, where all numerical values are changed to equal between 0 and 1. This is done as to change the values to a more common scale, without distorting their values.

This can be seen in the notebook in the section titled Normalising the data.

After this, the categorical values of the dataset need to be pre-processed, which is done by way of a sklearn library function called label encoder. This is done to make sure all categorical values equate to a numeric value. A lot of the data within the dataset is categorical values of yes and no, which will become 0 and 1, equalling a more common scale that can be assessed by machine learning models.

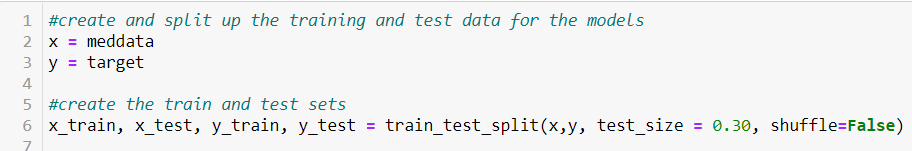
### **Modelling Phase**

Once the data is pre-processed and normalised, it is split from the label into its own dataset.

The aim of modelling is to use the data and train a model to be able to predict the results.

There are many model’s worth using for this phase. I chose to go with a variety of different models. The models I chose to use are Naïve Bayes, LinearSVC, KNN, Logistic Regression, Decision Tree and Neural Network.

Each model uses the same inputs and outputs. The training for each model is done by using the train/test split method, and then further training is done using cross validation. The data is split using a method known as the train/test split, which is a common approach where data is separated into X and Y, and then split 70/30. This refers to the percentile split of the data. X is all data in the dataset excluding the target, and Y is the target. The model is trained on X, and then evaluated on Y.



To test the data later, I opted to run cross validation at 10 folds towards analysis of the accuracy metric, to make sure the accuracy achieved is satisfactory. This is then compared to the accuracy of cross validation. Cross validation aims to help find out if the split in the data is even and well representative of the data that you are using.

Models are trained through many iterations. Some models chosen didn’t have many hyper-parameters that could be adjusted, but where adjustable different approaches are used and the final best result is kept, and then ran through cross-validation using the same hyper-parameters. These hyper-parameters including things such as the learning rate, what solver the model will use, the max iterations the model should run, and in the case of a neural network, it is possible to make adjustments to the amount of hidden layers and the batch size. A decision tree can be adjusted by specifying its max depth.

Each model is assessed by using a classification report and by way of confusion matrices. The process of these results can be found in the Results section of the report. Whilst the code showcases not only the result, it also shows the hyper-parameters that were tried out so that the accuracies can be compared.

### **Evaluation Phase**

All results are brought together at this stage and evaluated. As 6 models were created and trained with the same train/test split, now it is time to find out which performs the best.

The business approach for these models is all about achieving a high accuracy, alongside a low False Negative, as a False Negative indicates where a patient is classified as not being at risk when in fact, they are, which is very bad for the medical domain.

I used the combination of Confusion Matrices, Mean Squared Error valuations and accuracy scores as the metrics used to measure how well the models did, and which one would be the best choice out of all the options. This is done through a review of progress, which summarises the process to see if anything that could skew the result that has been obtained.

Once these reviews are done, the model would be deployed in the domain, to which the steps towards this distribution need to be reviewed. This will be done in a hypothetical sense next since this model won’t be deployed.

# **Results**

This section is used to showcase the Confusion Matrices These help to show how the model performed in classifying True Positive (top left), False Positive (top right), False Negative (bottom left) and True Negative (bottom right) for each classification algorithm.

We can use these matrices to calculate the accuracy of each model using the formulae;

TP + TN / (TP + TN + FP + FN). Classification reports can be found in the appendix section B.

|  |  |  |  |
| --- | --- | --- | --- |
| Naïve Bayes | Accuracy = 90% | Predicted | |
|  |  | Positive | Negative |
| Observed | Positive | 296 | 4 |
|  | Negative | 43 | 107 |

|  |  |  |  |
| --- | --- | --- | --- |
| LinearSVC | Accuracy = 96% | Predicted | |
|  |  | Positive | Negative |
| Observed | Positive | 292 | 8 |
|  | Negative | 9 | 141 |

|  |  |  |  |
| --- | --- | --- | --- |
| KNearestNeighbour | Accuracy = 96% | Predicted | |
|  |  | Positive | Negative |
| Observed | Positive | 291 | 9 |
|  | Negative | 9 | 141 |

|  |  |  |  |
| --- | --- | --- | --- |
| LogisticRegression | Accuracy = 96% | Predicted | |
|  |  | Positive | Negative |
| Observed | Positive | 290 | 10 |
|  | Negative | 8 | 142 |

|  |  |  |  |
| --- | --- | --- | --- |
| Decision Tree | Accuracy = 97% | Predicted | |
|  |  | Positive | Negative |
| Observed | Positive | 291 | 9 |
|  | Negative | 5 | 145 |

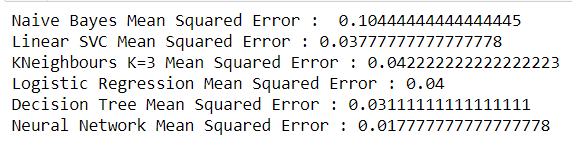
|  |  |  |  |
| --- | --- | --- | --- |
| Neural Network | Accuracy = 98% | Predicted | |
|  |  | Positive | Negative |
| Observed | Positive | 297 | 3 |
|  | Negative | 5 | 145 |

# **Evaluation & Discussion**

For the task, 6 models were created, trained on the data and evaluated on the accuracy of their predictions. The results of these can be found above. In regard to the medical domain, it was necessary to find a model that would be able to predict the best in regards to achieving the lowest False Negative result, as the idea of someone being found to be at risk when they are not, is better than someone going away with the idea they are not at risk, when the reality is that they are.

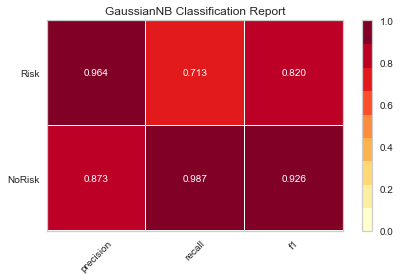
The problem with only determining how well a model predicts through its accuracy metric is that the accuracy doesn’t give you much to look at, other than how well the model predicts. The problem is this doesn’t tell you where the model fails. To do that, I looked at the classification report and confusion matrix for each model after it had been trained, as each gives a good indication as to how well a model is doing predicting positive and negative outcomes.

#### **Error Metrics**

The first thing I thought was worth looking at for each model are error metrics. While error metrics are mostly used for the evaluation of regression models, I still considered it in evaluation. I used the Mean Squared Error through sklearn metrics to see how much error each model was having. This metric revolves heavily on the accuracy of the model and is the opposite of the accuracy score.

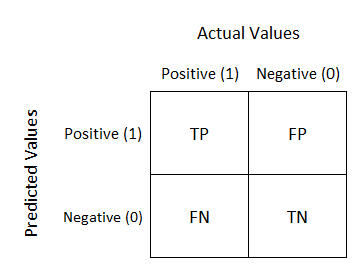
Within the 6 models, it is possible to see which is achieving good accuracy as well as achieving low error. Naïve Bayes as a model scored the worst accuracy of the models, whilst also scoring the highest MSE. This leads me to believe that although the metric seems useful, it is simply another accuracy metric and is therefore not very suited to distinguishing each model from another anymore than the accuracy metric already does.

#### **Classification Report**

This report is an essential way to measure the quality of predictions of a classification algorithm. The report itself shows the main classification metrics of Precision, Recall and F1 score, which are worked out through calculations found through use of True and False Positives, and True and False Negatives.

These metrics give you a percentile valuation of the numbers that can be found within a confusion matrix. I used a classification report for each model, as to get a more visual idea of how well each model was doing, as I felt that just using a confusion matrix wasn’t the best way to consider a model better than another when number in the matrix were close.

#### **Confusion Matrix**

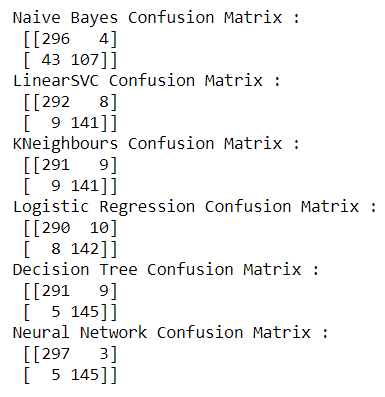
Each model used has its own style of algorithm and difference in execution, but they all present results that are worth evaluation. The main evaluation metric I used to distinguish which model was best for the task, was through a confusion matrix.

A Confusion matrix acts as a visual representation in way of a matrix, that shows the results of a binary test in an easy to understand way. The reason this type of evaluation was selected over all others, was its visual nature. It is easy to see where each model is getting the accuracy presented through the accuracy metric. The results of a CM can also be seen in a percentile format if the results are normalised. The notebook and appendix section B contain normalised confusion matrices for each trained model.

The most important metrics of the Confusion Matrix for the medical domain is False Positive and, in this task, False Negative. A False Negative is also known as a Type I error, these are deemed to be dangerous, as this is the misclassification that causes a patient to be thought to be safe, when they may require treatment. A False Positive is known as a Type II error, because it is deemed to be not as dangerous as Type I, and further medical analysis can resolve this sort of error.

#### **Deciding which Model was best**

To determine the best model, I took into consideration the accuracy score of each model, how well each model did in terms of Cross Validation, the mean squared error of each model and the Confusion Matrix of each model. The only model I considered bad for the task, was Naïve Bayes, which was solely because of its accuracy and its weakness in classification of the Type I error, which could be a disaster if the model was being used in a live setting later down the line.

Looking at the results, it is not entirely clear which model is better if the entire aim is Type I error classification. What can be seen from looking at all 6 confusion matrices, there are 2 models are do incredibly well in this respect.

When taking into considerations model accuracies, the models are very closely matched, there are mostly accuracies at the range 96% and up. Calling any of the models with these accuracies bad is a mistake. The downside is that only one model can really be implemented later, so there must be a choice, which is why Decision Tree and Neural Network are considered the top 2.

Decision Tree and Neural Network both do very well, scoring 5 in terms of incorrect Risk classification, which is a great result. The thing that makes Neural Network the best model for this task, however, is how well each of these models classifies Type II error.

What can be seen from the results for this, is that Neural Network scores 3, whilst Decision Tree scores 9. It was determined that Type I error is the biggest thing to consider when choosing a correct model, but due to both models doing very well here, Type II error must be considered too.

#### **Discussing the Methodology, what could be better**

The CRISP-DM methodology is flexible and makes the task of preprocessing, cleaning and evaluating this data easy and straightforward. It is easy to see where you are, what you should do next, and if you need to go backwards, the path to do so is clear and easy to follow.

However, it is not without its downfalls. One of the things that CRISP-DM suffers from is a lack of clarify. Defined by the model diagram itself. We assume that we can revisit the business understanding phase after evaluation, but is this really the case? The model lacks clarity. What I mean is that once you understand the business objective, you won’t really revisit it. Analysing and working on data is way more interesting, and most people using the methodology, will not revisit business understanding.

The biggest issue however with CRISP-DM, is that is not updated to deal with the most common issue in the world today, Big Data. Whilst CRISP-dm maintains the highest popularity amongst the methodologies at 43% (Stirrup, 2017), the importance and scale of data since its creation is very different. Business needs are different, and the methodology hasn’t really adapted to the times.

Where the stages of Modelling, Data Preparation and Evaluation are clear and concise, it is the areas of Understanding and Deployment where the methodology starts to show age. It does not adapt to Big Data, and this leads to issues for a business, where a business cannot understand the complexities of the data they have, how can you understand what the business wants from this data if they don’t understand themselves.

# **References**

Stirrup, J., 2017. *jenstirrup.com.* [Online]   
Available at: https://jenstirrup.com/2017/07/01/whats-wrong-with-crisp-dm-and-is-there-an-alternative/  
[Accessed 23 11 2019].

Taylor, J., 2017. *KDNuggets.* [Online]   
Available at: https://www.kdnuggets.com/2017/01/four-problems-crisp-dm-fix.html  
[Accessed 23 11 2019].

Wang, R. R., 2012. *softwareinsider.org.* [Online]   
Available at: https://blog.softwareinsider.org/2012/02/27/mondays-musings-beyond-the-three-vs-of-big-data-viscosity-and-virality/  
[Accessed 23 11 2019].

# 

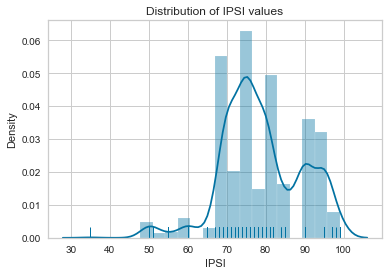
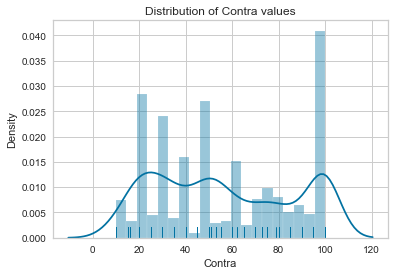
SAVE FOR ADDITION TO MODEL PHASE

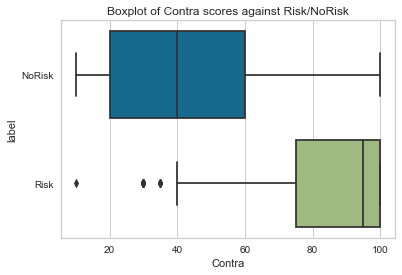
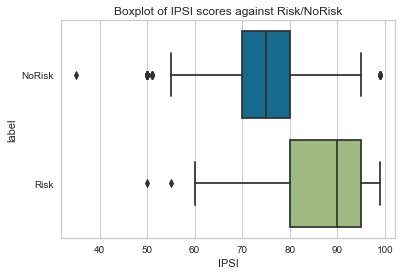
The first thing to do, is discuss why each model was chosen, and the results to which that model achieved. Where parameters were adjusted to try to obtain better results, this will be discussed as well.

#### **Naïve Bayes**

# **Appendix**

##### Section A – Graphs & Boxplots





#### Section B – Classification Reports for each trained model

***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.***

***RESULTS***

***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?***

In the case of the Naïve Bayes model in the report screenshot above, this model has an approximate 87.3% accuracy when it tried to predict correct NoRisk classifications. In the case of the medical domain and the task at hand, the main target is to get a high precision in this area. In this area, the best models were the Decision Tree and Neural Network, each with a 98.3% Precision score.

Considering the Naïve Bayes model again for this example, this model has the lowest Recall score amongst all models trained for Risk, with a score of 71.3%. The Decision Tree and Neural Network shared the highest score when it came to Recall of Risk classification of 96.7%.

Precision is the sum of TP / (TP+FP). Precision tells you what percentage of predictions by the model were correct.

Recall is the percentage of positive cases that the model managed to catch. This is done by taking TP / (TP+FN). It is the ability of the classifier to find all positive instances.

F1 score is the last important metric used within a classification report, which is a harmonic of the mean of precision and recall. This means the best score is 1.0, and the worst is 0.0. F1 score tells us what percent of positive predictions were correct. In the case of this task, I didn’t use the F1 score too much when determining which model was performing better.