Data Mining and Decision Systems
600092
Assigned Coursework Report

Brian Davis
Student ID: 201707824

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**, although these stages have been considered and are featured in the report.

Business Understanding Phase

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).

Measurement of the success of the project will be defined by how well a model classifies correctly that patients are at risk. Multiple model results will be considered and will be featured in the evaluation, to see which results that have been produced meet the aim of the business the best. Training a single model will be bad, as comparisons between models will help towards the goal of finding the model best suited for the task.

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. In the case of this dataset, it closely matches its data description, with no real differences that are worth noting. Missing values are present, evidenced by differing column value maximums.

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 or float as it contains numeric values. This can be changed through code in the next phase to instead be a float through the **to numeric()** function through pandas.

Data Preparation Phase

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.

The data is cleaned in steps through 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. The data is 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, these values are dropped. 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. Contra is also made numeric.

Visualisation

Visualisation is a tool that can be used throughout the methodology to explore the data in a more visual sense. Being able to look at the data in this way helps to find links which would take longer to find by simply scrolling through the data.

Graphically representing the data allows us to see things we might not see otherwise. It is incredibly effective at finding and identifying relationships in the data. As Arrhythmia is a yes/no attribute in the data, its relationship towards a result is not noticeable other than by understanding the data description or the domain.

Visualisation is used to find trends in the data and clarify suspicions 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.

Data Formatting

Data Formatting is necessary to ensure the data is of the correct format that models can use later. The two techniques used 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.

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 and visualised within graphs.

Modelling Phase

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. A variety of different models were chosen, which are Naïve Bayes, LinearSVC, KNN, Logistic Regression, Decision Tree and Neural Network.

Due to working with text classifications problems in the past, I knew about Naïve Bayes as a model. This is the reason I chose it, but unfortunately it didn't perform well enough for this task. KNN is chosen as it is an algorithm that works to find similar things that exist in proximity i.e. its neighbours, and the model had pretty good results.
Logistic Regression is a model taught in the module and is a good model to run when the classification target is binary and was used due to the fact Linear Regression is not suitable for the task, whilst Logistic Regression is a similar model to it.

LinearSVC is a linear model based around SVM. It uses a kernel trick to transform the data and then based on these transformations it finds the optimal boundary between the possible outputs. I chose this model due to the rising popularity of SVM models. Decision Tree was chosen due to the ability to visually see how the model splits the data to train itself. This is important as you can see what the model considers important for splits and helps to see what is considers the most important aspects.

Neural Network is the last model I worked with, and it was chosen as this model can be tailored and adjusted towards achieving a higher accuracy. Being able to adjust the degree to which the model learns is useful, and parameters can be adjusted to get to the result quicker or slower, dependant on preference. It works like the human brain and is incredibly useful for solving problems.

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. Each model is trained on X, and then evaluated on Y, where X is the data, and Y is the target.

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 through train/test split. Cross validation aims to help find out if the split in the data is even and well representative of the overall data that you are using, and that the model is not suffering from overfitting.

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 through a classification report and by way of its confusion matrix. The process of these results can be found in the Results section of this 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, it is important to find out which performs the best, and in which areas one model beats another if the accuracy appears to be the same.
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 & Classification Reports as the metrics used to measure how well the models did, and which one would be the best choice out of all the options. The process went about as well as expected, there were small sections that required returning to, such as the visualisation section, but overall the project timeline went along the line of the CRISP-DM methodology.

A review of progress is completed within the evaluation section of this report, which summarises the process to see if there is anything that could skew the result that has been obtained, and all results are evaluated for each model.

Deployment Phase

The Neural Network model would be the model to deploy, due to its higher accuracy and better suitability towards the domain target. Deployment must take some things into consideration however, such as the need to retrain the model due to the change in data. Real world data is a lot more difficult and will require cleaning each time to remove erroneous data, and due to this, the model will also need to be retrained.

Deployment of the model would have to take into consideration real-world metrics and can’t be expected to perform as well as it has on the data it has been trained on for this project. Deployment would be expected to take place in incremental stages, and would be expected to be deployed within hospitals, mostly focused on working within the cardiovascular section. The model could be deployed in areas in medicine that are like the dataset that was worked on, but this data would need to go through the same process as the data within this project did before it can be safely passed through to deployment.

The overall strategy towards deploying this would have to go through a deployment plan, to make sure it is suitable. I would personally employ the model as a first opinion to help a doctor or nurse come to a suitable conclusion based on some other varying factors, I wouldn’t recommend the model taking place of doctors for the classification task all by itself, but rather be used as a tool to help get a second opinion and help a doctor come to a conclusion in a faster timeframe.

Results

Naïve Bayes *Accuracy: 89%* *Predicted*

		Positive	Negative
Observed	Positive	296	4
	Negative	46	104

Naïve Bayes Classification Report	*Precision Sensitivity*	*Recall Specificity*	*F1 Score*
Risk	96%	69%	80%
NoRisk	87%	99%	92%

LinearSVC *Accuracy: 97%* *Predicted*

		Positive	Negative
Observed	Positive	294	6
	Negative	8	142

Linear SVC Classification Report	*Precision Sensitivity*	*Recall Specificity*	*F1 Score*
Risk	96%	95%	95%
NoRisk	97%	98%	98%

KNearestNeighbours *Accuracy: 96%* *Predicted*

		Positive	Negative
Observed	Positive	291	9
	Negative	9	141

K Nearest Neighbours Classification Report	*Precision Sensitivity*	*Recall Specificity*	*F1 Score*
Risk	94%	94%	94%
NoRisk	97%	97%	97%

LogisticRegression *Accuracy: 97%* *Predicted*

		Positive	Negative
Observed	Positive	294	6
	Negative	8	142

Logistic Regression Classification Report	*Precision Sensitivity*	*Recall Specificity*	*F1 Score*
Risk	93%	95%	94%
NoRisk	97%	97%	97%

Decision Tree	*Accuracy: 97%*	*Predicted*	
		Positive	Negative
Observed	Positive	291	9
	Negative	5	145

Decision Tree Classification Report	*Precision Sensitivity*	*Recall Specificity*	*F1 Score*
Risk	94%	97%	95%
NoRisk	98%	97%	98%

Neural Network	*Accuracy: 98%*	*Predicted*	
		Positive	Negative
Observed	Positive	295	5
	Negative	3	147

Neural Network Classification Report	*Precision Sensitivity*	*Recall Specificity*	*F1 Score*
Risk	97%	98%	97%
NoRisk	99%	98%	99%

Evaluation & Discussion

Accuracy is a good overall identifier at how well a model classifies the data and reaches a positive and accurate result, but the accuracy metric isn't necessarily the most important aspect. 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, this is due to treatment not being offered to a False Negative patient, which could lead to serious implications, and in some cases, the death of a patient.

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.

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 accurately the model predicts. The problem is that this doesn't tell you where the model fails. It is more effective to generate and evaluate based on a classification report and confusion matrix for each model as these give a good indication as to how well a model is doing predicting positive and negative outcomes.

The models trained for the purposes of this report all produced a result quite quickly, most in the space of a few seconds. The only model that took longer was the Neural Network, which took about a minute on powerful hardware to reach a result, slightly longer for its cross-validation result. This model is quite big compared to the other models trained towards this solution, and probably wouldn't be a good model to put on something such as a smart phone but might be reasonable on something like an Arduino, which is a pretty low energy device.

Found in the results section above, it is possible to see in more depth where the models fall short in their classification. The most important metric for this project is Sensitivity, which is the metric concerning how well the model predicted successfully those who are at Risk. The variety of models trained here all come up with similar accuracies in the range of 96-97%, which is where Sensitivity score starts to become a more important metric to separate these models.

The Sensitivity score, or the Precision, is the metric that tells you how well a model can predict the % of at-risk people correctly. Most of the models have good Sensitivity scores, whilst it is no surprise that the best performing model also has the best Precision scores too. Where this metric is of more use, is where the models share the same accuracy, which is the case regarding some of the other models.

Each model used has its own style of algorithm and difference in execution. Another metric that was used to discover model results alongside its use to compare models with a similar accuracy is the Confusion Matrices. Using this metric alongside the classification report helps to understand in more depth how the models do towards the task.

Confusion Matrices proved to be great when it came to the evaluation of the models. Before hyper-parameters were really tested in more depth, Decision Tree not only matched the accuracy of the Neural Network model, it also fared well when compared to the Precision and Recall scores as well. This shows the importance of how much the parameters can be explored for the Neural Network, in that without these being explored in the depth that they have been, then the choice of model would have been a close choice and may have led to multiple models being deployed, or at least being in contention.

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 sensitivity score 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 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 that 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 the Neural Network is considered the best for this domain.

The Neural Network achieves an accuracy of rounded 98% (99% Cross-Validation), with a 97% precision accuracy towards a Risk classification, and only gets the Type I error wrong in 3 of the 450 cases presented in the test dataset. The final decision for this comes down to not only the scores presented, but also the flexibility of hyper-parameter tuning that is possible with a Neural Network.

But calling the Neural Network the best model straight up is not the whole story. The beauty of the model comes from how well you can adjust the hyper parameters of the model. I run through a few different hyper parameters to see how adjustments in this area adjust the overall performance and final metrics of the model. All these hyper parameter adjustments can be seen in the code, alongside any of the other models that had their parameters tuned where possible.

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 this means 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.

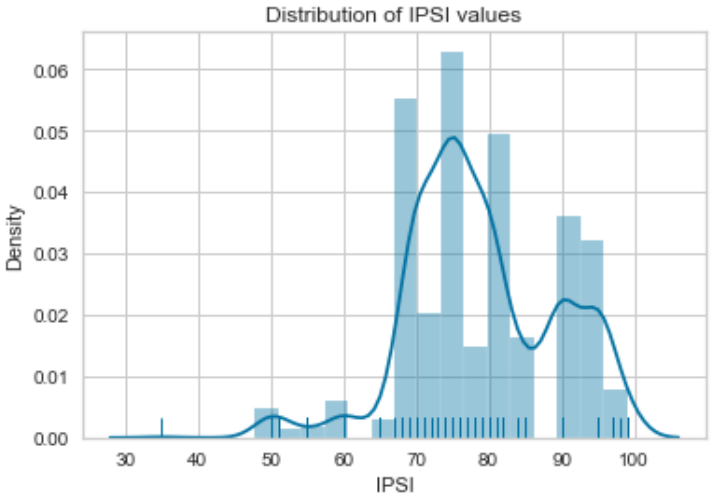
The biggest issue however with CRISP-DM, is that is not updated to deal with the most common issue in the world today, Big Data. Whilst CRISP-DM maintains the highest popularity amongst the methodologies at 43% (Stirrup, 2017), the importance and scale of data since its creation is very different. Business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

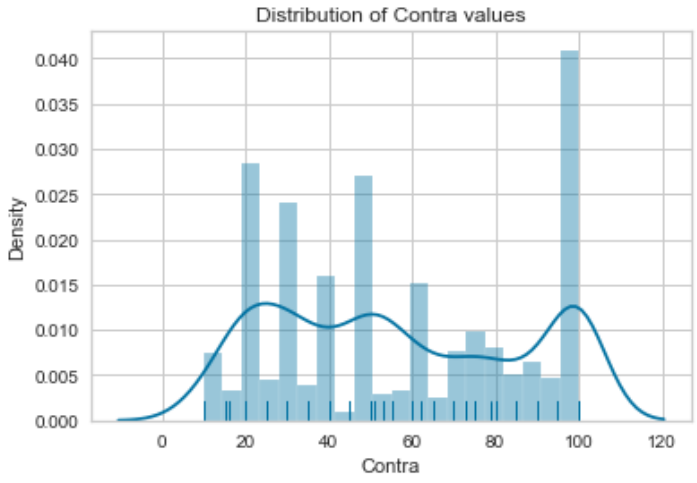
Where the stages of Modelling, Data Preparation and Evaluation are clear and concise, it is the areas of Understanding and Deployment where the methodology starts to show age. It does not adapt to Big Data, and this leads to issues for a business, where a business cannot understand the complexities of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

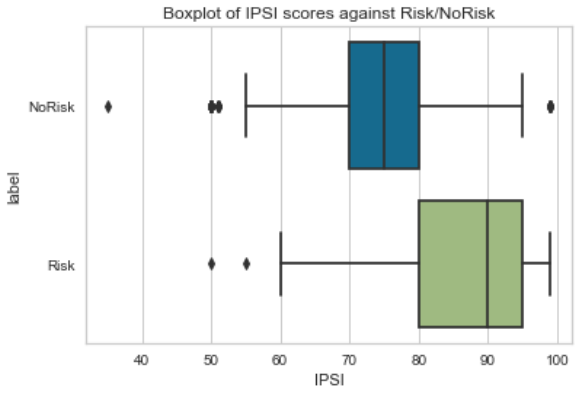
References

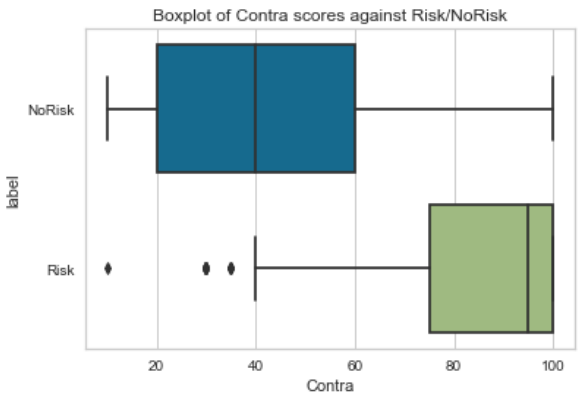
Stirrup, J., 2017. *jenstirrup.com.* [Online]
Available at: https://jenstirrup.com/2017/07/01/whats-wrong-with-crisp-dm-and-is-there-an-alternative/
[Accessed 23 11 2019].
Taylor, J., 2017. *KDNuggets.* [Online]
Available at: https://www.kdnuggets.com/2017/01/four-problems-crisp-dm-fix.html
[Accessed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

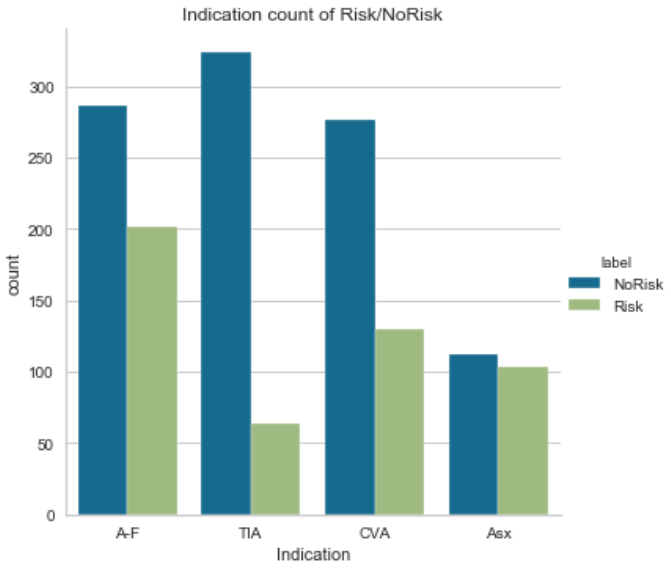
Appendix

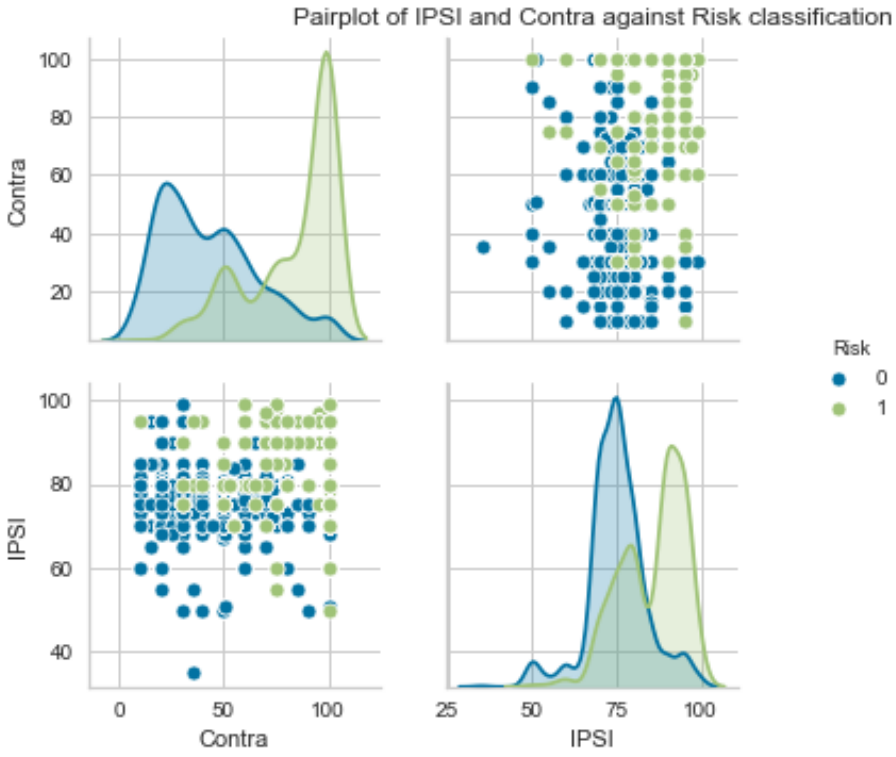


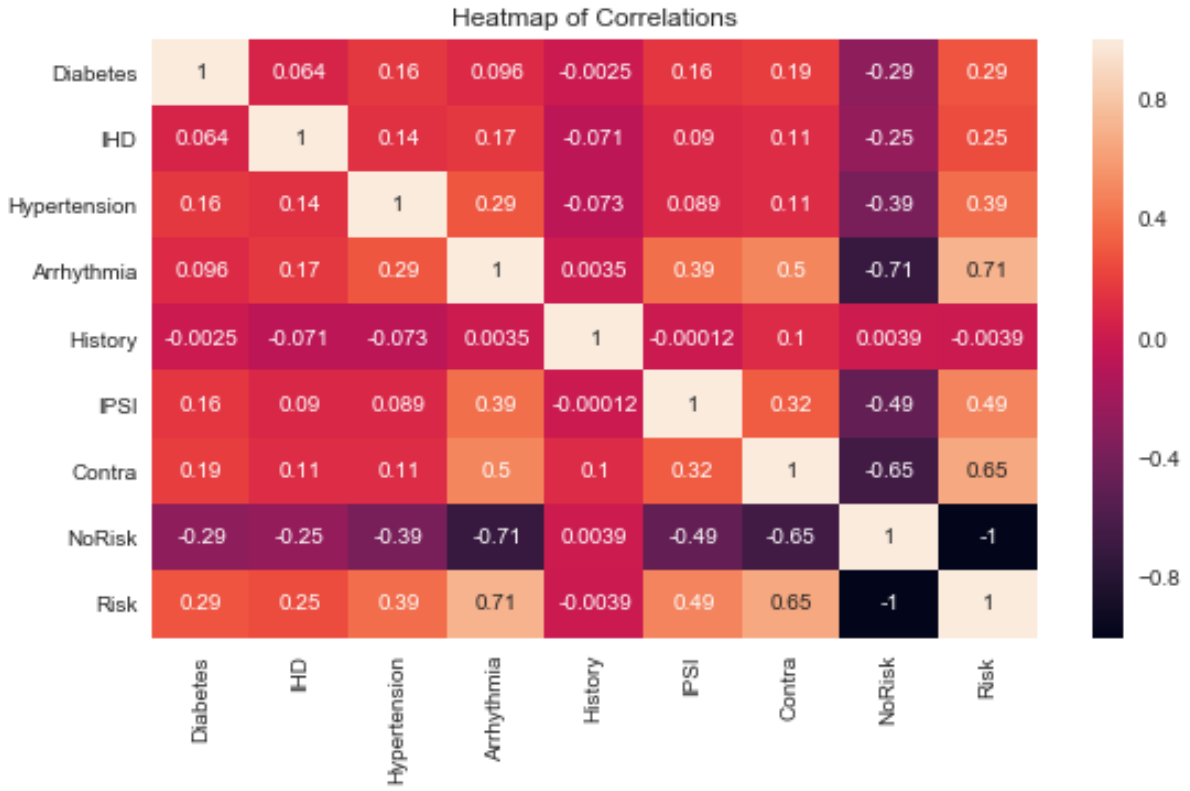


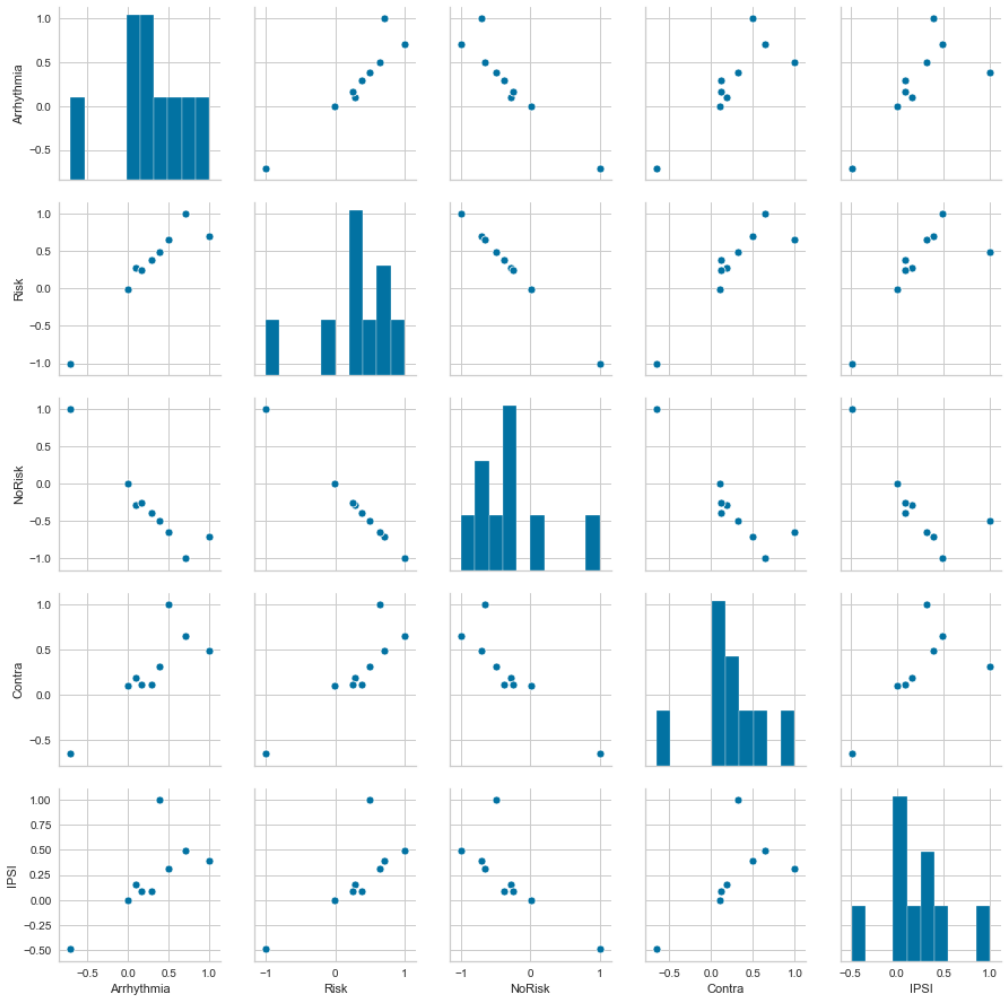

Boxplot of IPSI scores against Risk/NoRisk
label
NoRisk
Risk
40
50
60
70
80
90
100
IPSI

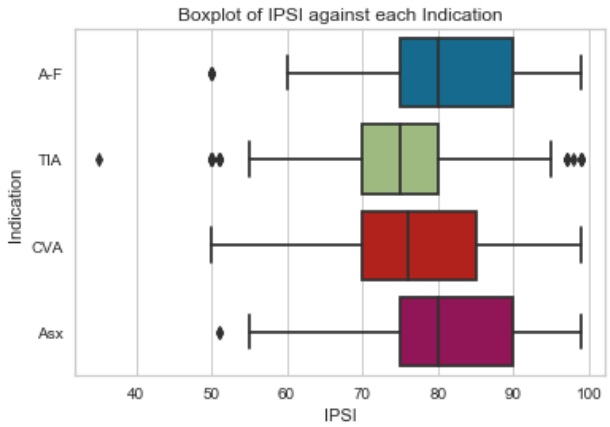

Boxplot of Contra scores against Risk/NoRisk
label
NoRisk
Risk
20
40
60
80
100
Contra

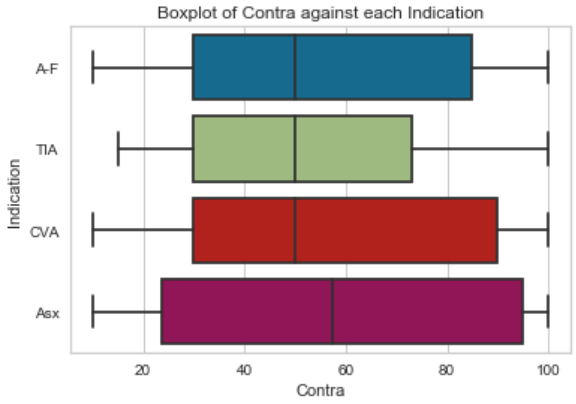

Indication count of Risk/NoRisk
count
0
50
100
150
200
250
300
A-F
TIA
CVA
Asx
Indication
label
NoRisk
Risk

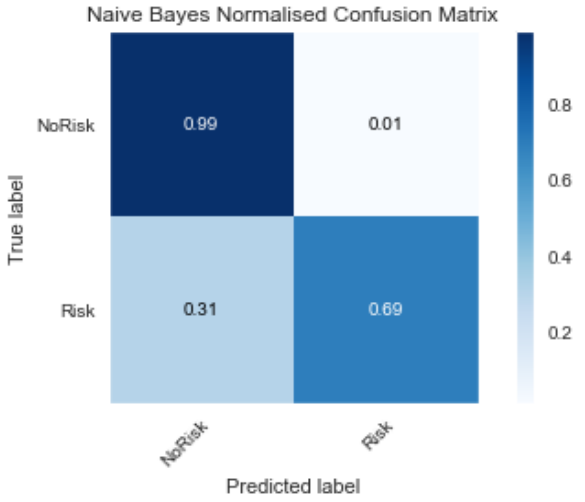

Pairplot of IPSI and Contra against Risk classification
Contra
IPSI
20
40
60
80
100
0
50
100
25
50
75
100
Contra
IPSI
Risk
0
1

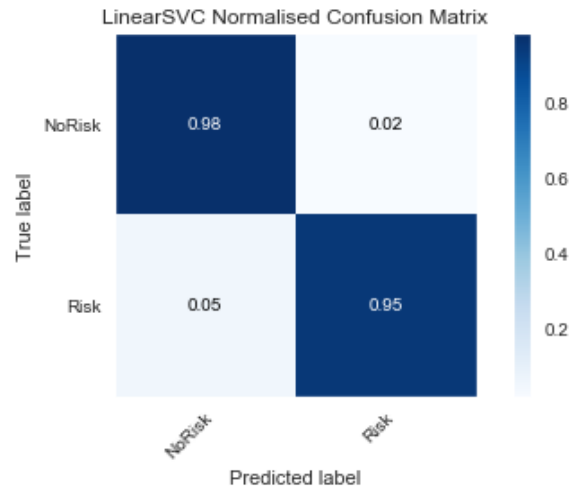

Heatmap of Correlations
Diabetes 1 0.064 0.16 0.096 -0.0025 0.16 0.19 -0.29 0.29
IHD 0.064 1 0.14 0.17 -0.071 0.09 0.11 -0.25 0.25
Hypertension 0.16 0.14 1 0.29 -0.073 0.089 0.11 -0.39 0.39
Arrhythmia 0.096 0.17 0.29 1 0.0035 0.39 0.5 -0.71 0.71
History -0.0025 -0.071 -0.073 0.0035 1 -0.00012 0.1 0.0039 -0.0039
IPSI 0.16 0.09 0.089 0.39 -0.00012 1 0.32 -0.49 0.49
Contra 0.19 0.11 0.11 0.5 0.1 0.32 1 -0.65 0.65
NoRisk -0.29 -0.25 -0.39 -0.71 0.0039 -0.49 -0.65 1 -1
Risk 0.29 0.25 0.39 0.71 -0.0039 0.49 0.65 -1 1
0.8
0.4
0.0
−0.4
−0.8
Diabetes
IHD
Hypertension
Arrhythmia
History
IPSI
Contra
NoRisk
Risk

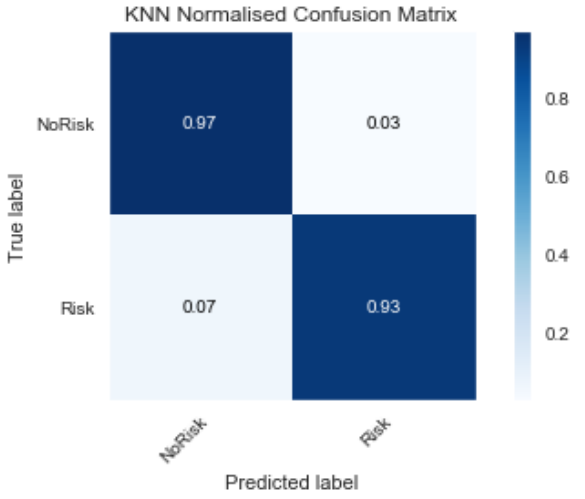

Arrhythmia
Risk
NoRisk
Contra
IPSI

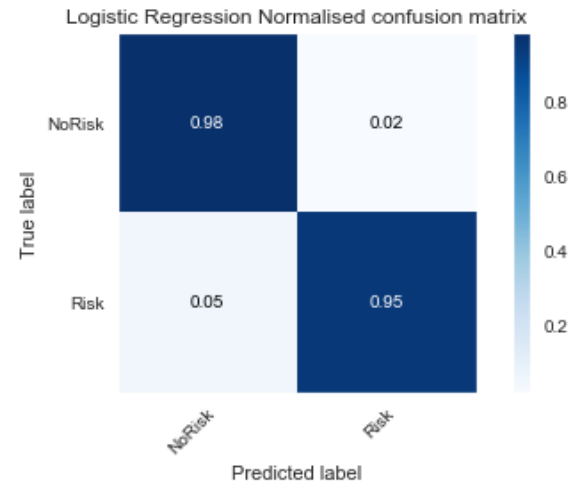

Boxplot of IPSI against each Indication
A-F
TIA
CVA
Asx
Indication
40
50
60
70
80
90
100
IPSI

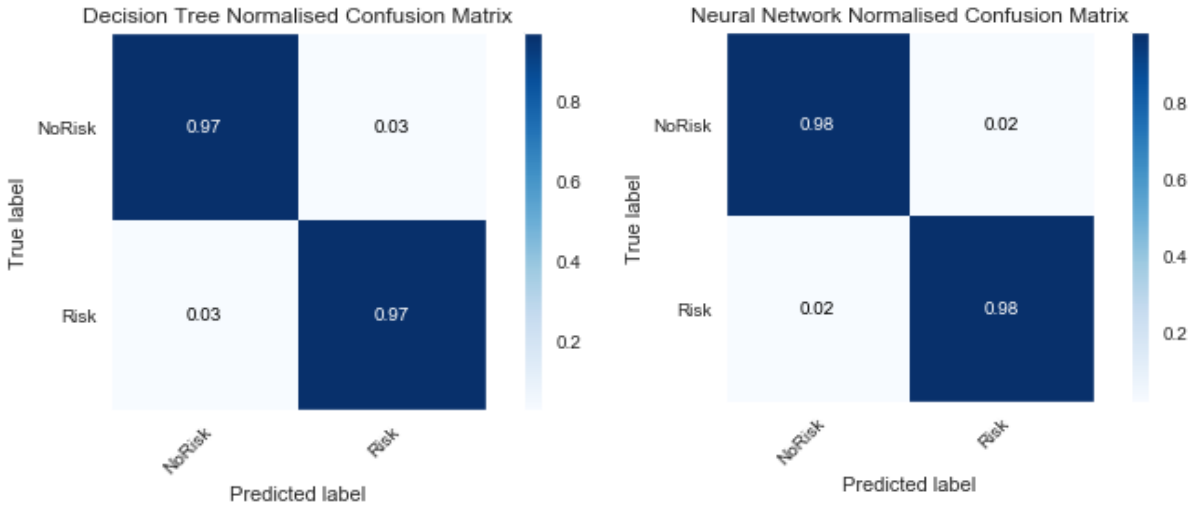

Boxplot of Contra against each Indication
A-F
TIA
CVA
Asx
Indication
20
40
60
80
100
Contra

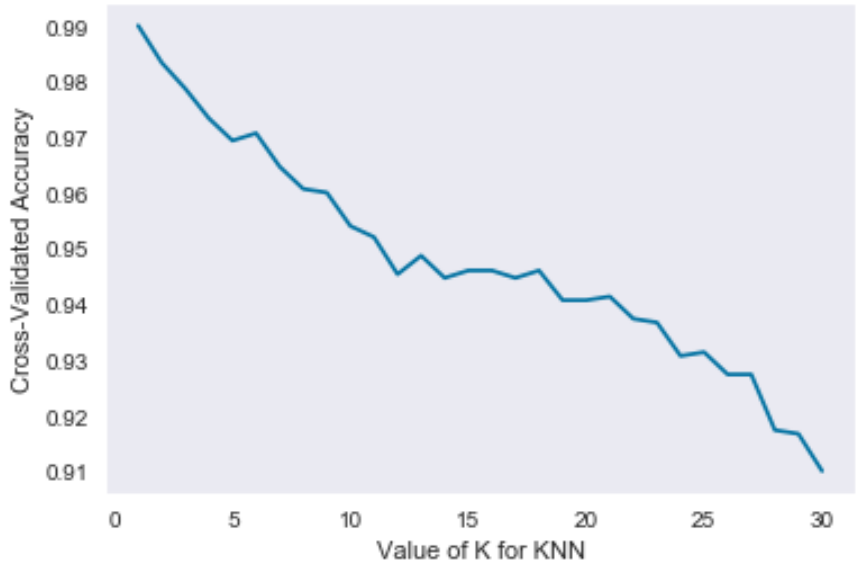

Naive Bayes Normalised Confusion Matrix
NoRisk
Risk
True label
0.99
0.01
0.31
0.69
0.8
0.6
0.4
0.2
NoRisk
Risk
Predicted label


LinearSVC Normalised Confusion Matrix
NoRisk
Risk
True label
0.98
0.02
0.05
0.95
0.8
0.6
0.4
0.2
NoRisk
Risk
Predicted label


KNN Normalised Confusion Matrix
NoRisk
Risk
True label
0.97
0.03
0.07
0.93
0.8
0.6
0.4
0.2
NoRisk
Risk
Predicted label


Logistic Regression Normalised confusion matrix
NoRisk
Risk
True label
0.98
0.02
0.05
0.95
0.8
0.6
0.4
0.2
NoRisk
Risk
Predicted label


Decision Tree Normalised Confusion Matrix
NoRisk
Risk
True label
0.97
0.03
0.03
0.97
0.8
0.6
0.4
0.2
NoRisk
Risk
Predicted label
Neural Network Normalised Confusion Matrix
NoRisk
Risk
True label
0.98
0.02
0.02
0.98
0.8
0.6
0.4
0.2
NoRisk
Risk
Predicted label


Cross-Validated Accuracy
0.99
0.98
0.97
0.96
0.95
0.94
0.93
0.92
0.91
0
5
10
15
20
25
30
Value of K for KNN