Data Mining and Decision Systems
600092
Assigned Coursework Report

Student ID: 554423

Date: 07 October 2019

Due Date: 12 December 2019

Methodology

Introduction

To data mine is to analyse data and transform it with the purpose of discovering knowledge and giving insight into a problem. Data mining can allow business and organisations to analyse data from multiple sources to make informed decisions. It is a creative process requiring an array of skills in areas such as data cleaning and modelling. Conversely, considering the range of creative skills needed, there currently is no standard frame work in which to carry out a data mining project. Therefore, the success or failure of a project is highly dependant on the individuals undertaking the project. (Wirth and Hipp, 2019)

This project plans to explore the different creative processes one could use to analyse a set of data. Creative processes can include the various methods used to clean a data set. There are also the different classifiers that can be trained to interpret the data and the hyperparameters that can modify their functionality and accuracy for a particular data set. Appropriate data transformations and mining techniques will be explored, and their effectiveness evaluated.

The CRISP-DM model will be used to structure the progression of the project. CRISP-DM can be described as a hierarchical process model comprised of, Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation and Deployment. For the purposes of this project, Deployment will be omitted from the project as this simply a research project. Each section will detail characteristics of the project and describe the methodology used.

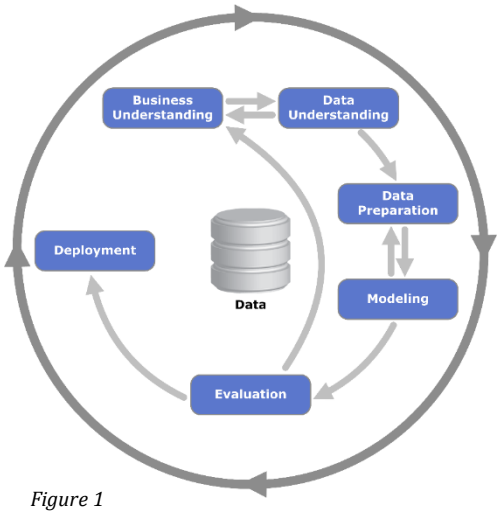

Figure 1

Business Understanding

The project's main objective is to clean, transform and manipulate a data set in various forms. This data set is comprised of patient information from the domain of cardio-vascular medicine. With this cleaned data, classifiers will be trained on the data set with the intention of being able to accurately deduce whether a patient is at risk or not based on patient diagnosis.

Regarding the business understanding of this project, a constraint may be the lack of domain knowledge. Despite the classifiers having the to identify trends and correlations, explaining them or attempting to rectify erroneous data proved difficult without domain knowledge on subject matter.

Data Understanding

Data Characteristics

Familiarity with a data set is key before any analysis of or training can be done. As the given data set is comprised of legacy data, inherently speaking, it is not optimised for data mining. The data set contains missing variables and is liable to be limited in depth. Duplicated, contradictory or corrupted data are all likely due to the characteristics of human error during data collection. This can pose risks regarding the consistency and accuracy of the data making data cleaning an all too necessary process when data mining.

Variable Types and Data Quality

The patient data contains 1520 records and 11 columns of which two are the ‘Random’ and ‘Id’ columns. The ‘Random’ and ‘Id’ columns contain numeric values that appear to uniquely identify each record. Albeit, whilst they seem to serve as a primary key, this function seems redundant as the index for the data frame serves the same function. On the other hand, assuming there may duplicates in each column, a combination of the two columns may result in no duplicate primary keys. A scenario like this may give insight into whether records could belong to the same person undergoing multiple examinations at different times. The number of duplicate cells in ‘Random’ and ‘Id’ may also suggest the possible time for when the data was collected. This may be important as whilst the given data is synthetic, it is based off a real-world data set. Nevertheless, the ‘Random’ and ‘Id’ columns do not appear to contain any useful information regarding identifying whether a patient is at risk. Therefore, these columns may be omitted from the final data frame before training.

The ‘Indication’ column contains four values known as A-F, ASX, CVA and TIA. These values describe what cardiovascular event triggered the hospitalisation of a patient. This may important when it comes to finding correlations between whether a patient is at risk or not. Correlations such as, what other aspects of a patient’s health may have led to a certain cardiovascular event, may also be identified. Furthermore, certain cardiovascular events may be statistically more or less likely to put a patient at risk compared to others. Domain knowledge in this area can be incredibly useful when explaining possible the trends + as well as any inconsistencies.

The ‘Diabetes’, ‘IHD’, ‘Hypertension’, ‘Arrythmia’, and ‘History’ columns all have yes or no values to indicate their respective status. On the other hand, ‘IPSI’ and ‘Contra’ use percentages. Lastly, there is the ‘label’ column responsible for conveying when a patient is at risk or not. If a patient is at risk, then their mortality is low; and vice versa. Although, the ‘label’ column does not give any indication as to threshold at which a patient would be considered to have a high mortality. Hypothetically, if the column illustrated patient mortality

using percentages to show chance, it may offer a better insight as to what may increase or decrease a patient's mortality.

A table description of the data can be found here detailing the variable types, number of values and a short description of what they represent. (*Figure 2)*

Data Preparation 1.0 (White Box Model Simple Clean)

Before the data can be used to train a classifier, its contents must be cleaned to reduce the number of incomplete records and potentially anomalous data. There are many advantages to using the white box model for data cleaning in juxtaposition to a black box model. A black box model uses a classifier to predict the values of null cells inside a data set. Consequently, the justifications and variables that influence the classifiers predictions are unknown to the individual. In addition to this, a trained classifier likely lacks the domain knowledge to account for anomalies and erroneous data. In contrast to black box modelling, white box is carried out buy the individual cell by cell. Erroneous and anomalous data can be cleaned with the individual having full knowledge of the justification that went into making a prediction. (GeeksforGeeks, 2019)

Identifying Unspecified Values

The first step to cleaning the data is identifying all the unspecified values in the data frame. In this set of data, such values came in the form of empty white spaces in a string variable for the 'Contra' attribute. All other columns excluding 'Random' and 'Id' contained null values in their cells. In addition to this, 'label' contained two 'Unknown' values in two of its cells. After identifying 20 records with an unspecified value each, progress could be made to clean the values in each attribute. *(See cell 9, Data Mining 600092, 554423)*

'label' Attribute

The description of the data provided stated that the 'label' attribute had only two values to determine a patient's mortality. Despite the 'Unknown' values signifying the data for that cell is empty, it is not entirely impossible for patients to remain undiagnosed for unspecified amounts of time during hospitalisation (Levetan et al., 2019). Therefore, it may reasonable to suggest that, at the time of data collection, this data was in fact intentionally placed to suggest as such. On the other hand, predicting the risk or no risk chance for the 'Unknown' values may offer a greater insight as to what a patient's mortality may in fact have been. For this reason, the 'Unknown' values will be kept in the data set to be predicted. *(See cell 9, Data Mining 600092, 554423)*

'Contra' Attribute

As the 'Contra' attribute does not technically contain any null values, instead containing a white space, searching for a null value will return zero entries. The same is true for 'label' as 'Unknown' is no considered a null entry. This explains the `patients.isnull().sum()` function only returning 3 null values for 'label' whilst ignoring the two 'Unknowns' whereas 'Contra' returns no null values. *(See cell 9, Data Mining 600092, 554423)*

Impute Methods

The process for imputing data involved searching for similar records to the records containing the null values. As an attribute containing a null value is likely to have similar characteristics to other records, statistical methods such as mean, and mode can be used to estimate what a null value might be. This subclass specific data will ensure that that search space only returns information relevant to that null value. However, the attributes containing no numeric values can only use mode as a method of prediction. Whilst a seemingly reasonable option, this may present an issue if there was no overwhelming majority between possible values. This is due the possibility of a classifier developing an unreliable bias when training if one simply cleaned the data based on the majority values. Thankfully, the issue never arose whilst cleaning. Although if this hadn't been the case, it would have not been entirely irrational to remove the record from the data set.

'Indication' Attribute

Whilst cleaning 'Indication', an issue with the integrity of the data was discovered. There was an inconsistency with the naming of one of the values. Some of the records for 'ASx' were represented as 'Asx'. The lack in uniformity would prove to be an inconvenience whilst cleaning as both values would need to be searched for. To rectify this, both values were replaced with 'ASX'. *(See cell 20, Data Mining 600092, 554423)*

Dropping

Whilst cleaning the 'label' attribute, the record 475 appeared to have no similar records. Without any similar records to refer to, changing the search criteria to reduce the similarity requirements may impair the likely hood of obtaining an accurate result. For this reason, record 475 was dropped from the data set. *(See cell 69, Data Mining 600092, 554423)*

As mentioned in Data Understanding, the 'Random' and 'Id' columns do not appear to hold any data relating to the patient's mortality. Henceforth, the data will be omitted from the official data set.

Modelling 1.0 (White Box Model)

The classifiers used in this project are logistic regression, k-nearest neighbours (KNN), multi-layer perceptions (MLP) and decision trees. Before the classifiers are trained, the data set is shuffled once to avoid the classifiers from developing any bias. By shuffling the data, it skews any patterns that may influence the classifier's decision-making process. Then, when testing the trained classifier, the bias inherited from the pattern may leave it less able to adopt new trends, reducing the accuracy of its predictions. To maintain a fair test, the data set is only shuffled once to ensure the other classifiers are trained on the same, transformed data.

Logistic Regression

As the ‘Diabetes’, ‘IHD’, ‘Hypertension’, ‘Arrhythmia’, ‘History’ and ‘label’ all contain dichotomous data, logistic regression seems an ideal option. The classifier as excels at describing data and explaining the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables (Statistics Solutions, 2019). This will work to the data set’s advantage as the ‘IPSI’ and ‘Contra’ attributes both contain interval data types.

For the classifier to be able to interpret the binary data from the relevant attributes, the values must be mapped to a number. Positive values a represented as ‘1’ and negative values as ‘0’. The values in ‘Indication’, are also mapped to enumerate the data for the classifier. *(See cell 84, Data Mining 600092, 554423)*

To keep the training time to a minimum whilst maintaining a high accuracy, the max number of iterations was set to 50000. The solver used is ‘newton-cg’. *(See cell 91, Data Mining 600092, 554423)*

MLP

MLP by far is the longest classifier to train in this project. Increasing the number of hidden layers increases the accuracy of the classifier. However, after a certain number of hidden layers, improvements to the accuracy start to suffer from diminishing results. This depends on the complexity and the size of the data set. *(See cell 97, Data Mining 600092, 554423)*

KNN

KNN is essentially an algorithm that searches for the number (k) nearest neighbours of the current object. The hyperparameter to change the value of k can be used to increase the search space. If increased, the algorithm is liable to have a reduced accuracy. However, if this value is too low, the algorithm is known to suffer from overfitting. This is because reducing the search space and limiting the number of nearest neighbours forces the classifier to fit too closely to a limited set of data points. The problem arises when the classifier begins to priorities random errors in data rather than the relationships. (Frost, 2019)

To avoid the risk of overfitting, the k value was set to 5. *(See cell 102, Data Mining 600092, 554423)*

Decision Tree

For decision trees, the limit of the $\mathrm{max\_depth}$ hyperparameter was set to 9 after considering the limited size of the data set’s 1520 records. An advantage to this is that the classifier can maintain its ability to search for complex patterns in the data. Additionally, the chance of the classifier falling victim to overfitting decreases and prevents it from focusing on finding rules for erroneous data. *(See cell 108, Data Mining 600092, 554423)*

Data Preparation 2.0 (Black Box Model Using KNN)

The methodology for the second data set was created to contrast the white box model by training a classifier to clean the data. By training a classifier to predict the null values of a data set, it is possible that the classifier may be able to generate more accurate predictions compared to a data set cleaned by a human. The KNN classifiers k value was left at 5 to mitigate the chances of overfitting.

Imputing with a classifier was done buy training the classifier on the attribute the null value resides in. *(See cell 128, 142, 151, 163, 172, 187, 193, Data Mining 600092, 554423)*

After training, the record containing the null value has the attribute for the null value removed. The remaining record is then passed into the df.predict() function. This function returns a prediction of what the null value should be based on the other values in the record. *(See cell 133, 134, 135, Data Mining 600092, 554423)*

Modelling 2.0 (Black Box Model Using KNN)

All the parameters for the classifiers remained the same with the intention of keeping them as control variable. This ensures that the only independent variables are the two data sets using the simple white box model and the classifier cleaned black box model. *(See cell 220 – 243, Data Mining 600092, 554423)*

Results

	TP	TN	FP	FN	Accuracy %	Sens%	Spec%
Logistic Regression	100	194	7	3	96.710526	97.087379	96.517413
MLP	104	192	3	5	97.368421	95.412844	98.461538
KNN	99	185	8	12	93.421053	89.189189	95.854922
Decision Tree	104	192	3	5	97.368421	95.412844	98.461538
Logistic Regression (KNN)	100	193	7	4	96.381579	96.153846	96.500000
MLP (KNN)	100	196	7	1	97.368421	99.009901	96.551724
KNN (KNN)	96	190	11	7	94.078947	93.203883	94.527363
Decision Tree (KNN)	106	196	1	1	99.342105	99.065421	99.492386

Figure 3 Truth table 80% 20% split

	TP	TN	FP	FN	Accuracy %	Sens%	Spec%
Logistic Regression	141	299	8	8	96.491228	94.630872	97.394137
MLP	140	302	9	5	96.929825	96.551724	97.106109
KNN	134	294	15	13	93.859649	91.156463	95.145631
Decision Tree	143	304	6	3	98.026316	97.945205	98.064516
Logistic Regression (KNN)	138	300	9	9	96.052632	93.877551	97.087379
MLP (KNN)	136	304	11	5	96.491228	96.453901	96.507937
KNN (KNN)	134	294	13	15	93.859649	89.932886	95.765472
Decision Tree (KNN)	144	304	3	5	98.245614	96.644295	99.022801

Figure 4 70% 30% split

Evaluation & Discussion

After training each model, to assess the reliability and accuracy of each model, a truth table containing the classifiers used with both data sets was created. It details the number and percentages of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN). It also details the accuracy, sensibility and the specificity *(Figure 3).* For a model to have a high sensibility and specificity, it signifies that the model is accurately predicting a patient's mortality.

One peculiarity that was discovered was that occasionally, each classifier would return with an accuracy identical to another classifier using a different data set. In addition to this, the occurrence was not limited to classifiers only sharing the same results with the same classifier. It is probable that due to the train test split being limited to an 80% train and a 20% test, the test size may not be large or complex enough to create any meaning full evidence *(Figure 5).* This assumption was further validated by the overall accuracy of the models. Often, the accuracy of each classifier would be within 0.5-1.5% of the same classifier using a different data set.

In an attempt to obtain some conclusive evidence, a 70% 30% split was tested across all models. This attempt to gain some clear trend however, failed to yield any contrasting accuracies between the data sets. *(Figure 4) (Figure 6)*

The clear lack of contrasting performance appears to convey that the difference between the two sets of data were similar enough for the classifiers to obtain a similar result. In this case, there was no clear difference in accuracy. Despite this, if the data set had been larger or required a more significant amount of data cleaning, perhaps a clearer disparity in performance would have been present. As a larger percentage of the data set would have needed to be cleaned, it is likely that differences in their predictions used would be more prevalent as classifiers would be more likely to discover unique trends.

To visualise other aspects of a model's performance, its number of TNs, TPs, FNs and FPs can be used to calculate a model's sensibility and specificity. Sensibility describes how often

a model generates a positive result for cases that are indeed positive, whereas specificity describes how correctly a negative output is outputted. Sensitivity is the variable that needs to remain as high as possible. If it were ever too low, a patient may walk away from the hospital without being diagnosed and assumed to not be at risk. For the majority of the tests, decision trees and MLP using the 'adam' hyperparameter for the solver= function, were able to maintain the highest sensitivity regardless of the data set model used. If ever deployed, these classifiers would be the statistically safer option to aid in patient mortality prediction. *(Figure 3) (Figure 4) (Figure 5) (Figure 6)*

After the data cleaning process and converting the objects type attributes to enumerated values, visual metrics could be used to gather conclusive evidence and give insight into possible correlations between attributes. A heat map of the diagram visually displays the correlations between attributes and displays values from 0-1 to signify the strength. Regarding a patient's mortality, 'Arrhythmia' had the highest correlation with 'label' at 0.71. 'IPSI' and 'Contra' also have relatively high correlations with 'label' at 0.49 and 0.65. *(Figure 6)*

This data seemed to signify that having a high 'Contra', 'IPSI' and being positive for 'Arrhythmia' increased decreased a patient's mortality, putting them at risk. To help visualise this correlation, a scatter plot details what the relationship would look like. *(Figure 9)*

Conclusion

In conclusion, an improvement that could be made to avoid inconclusive results would be to manipulate the feature set of the training data. Not every aspect of the patient's diagnosis would have the same impact on a patient's mortality. Training classifiers on different aspects of a patient's medical situation would allow one to identify what factors are most impactful to the outcome of a diagnosis. Additionally, with enough testing, data less significant to the search criteria can be identified and omitted from the search criteria.

Instead of attempting to predict empty values, it may be worth attempting to create models with the null values removed and test it against a data set with imputed data. Doing this may give insight into how beneficial imputed data can be or if it improves the accuracy, specificity or sensitivity of a classifier.

References

Scikit-learn.org. (2019). *1.10. Decision Trees — scikit-learn 0.22 documentation*. [online] Available at: https://scikit-learn.org/stable/modules/tree.html [Accessed 12 Dec. 2019].
GeeksforGeeks. (2019). *Differences between Black Box Testing vs White Box Testing - GeeksforGeeks*. [online] Available at: https://www.geeksforgeeks.org/differences-between-black-box-testing-vs-white-box-testing/ [Accessed 12 Dec. 2019].

Frost, J. (2019). *Overfitting Regression Models: Problems, Detection, and Avoidance - Statistics By Jim*. [online] Statistics By Jim. Available at: https://statisticsbyjim.com/regression/overfitting-regression-models/ [Accessed 12 Dec. 2019].

Levetan, C., Passaro, M., Jablonski, K., Kass, M. and Ratner, R. (2019). *Diabetes CareUnrecognized Diabetes Among Hospitalized Patients*. [online] Diabetes Care. Available at: https://care.diabetesjournals.org/content/21/2/246 [Accessed 11 Dec. 2019].
Statistics Solutions. (2019). *What is Logistic Regression? - Statistics Solutions*. [online] Available at: https://www.statisticssolutions.com/what-is-logistic-regression/ [Accessed 12 Dec. 2019].

Scikit-learn.org. (2019). *sklearn.linear_model.LogisticRegression — scikit-learn 0.22 documentation*. [online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html [Accessed 12 Dec. 2019].

Scikit-learn.org. (2019). *sklearn.neighbors.KNeighborsClassifier — scikit-learn 0.22 documentation*. [online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html [Accessed 12 Dec. 2019].

Scikit-learn.org. (2019). *sklearn.neural_network.MLPClassifier — scikit-learn 0.22 documentation*. [online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html [Accessed 12 Dec. 2019].

Wirth, R. and Hipp, J. (2019). *CRISP-DM: Towards a Standard Process Model for DataMining*. [online] Citeseerx.ist.psu.edu. Available at: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.198.5133&rep=rep1&type=pdf&fbclid=IwAR0zJ3bPGG2b5EQ6JMroTShooGDv7TB20fewuTCTXgdXQ-5F6SwjXvdkRcY [Accessed 10 Dec. 2019].

Appendices

Attribute	Value Type	NumberOfValues	Values	Comment
Random	Real	Number of Records	Unique	Real number of help in randomly sorting the data records
Id	Integer	Max of Number of Records	Unique to patient	Anonymous patient record identifier: Should be unique values unless patient has multiple sessions
Indication	Nominal	Four	{a-f, asx, cva, tia}	What type of Cardiovascular event triggered the hospitalisation?
Diabetes	Nominal	Two	{no, yes}	Does the patient suffer from Diabetes?
IHD	Nominal	Two	{no, yes}	Does the patient suffer from Coronary artery disease (CAD), also known as ischemic heart disease (IHD)?
Hypertension	Nominal	Two	{no, yes}	Does the patient suffer from Hypertension?
Arrhythmia	Nominal	Two	{no, yes}	Does the patient suffer from Arrhythmia (i.e. erratic heart beat)?
History	Nominal	Two	{no, yes}	Has the patient a history of Cardiovascular interventions?
IPSI	Integer	Potentially 101	[0, 100]	Percentage figure for cerebral ischemic lesions defined as ipsilateral
Contra	Integer	Potentially 101	[0, 100]	Percentage figure for contralateral cerebral ischemic lesions
Label	Nominal	Two	{risk, norisk}	Is the patient at risk (Mortality)?

Figure 2

	TP	TN	FP	FN	Accuracy %	Sens%	Spec%
Logistic Regression	94	196	10	4	95.394737	95.918367	95.145631
MLP	94	198	10	2	96.052632	97.916667	95.192308
KNN	92	196	12	4	94.736842	95.833333	94.230769
Decision Tree	99	199	5	1	98.026316	99.000000	97.549020
Logistic Regression (KNN)	108	184	7	5	96.052632	95.575221	96.335079
MLP (KNN)	104	187	11	2	95.723684	98.113208	94.444444
KNN (KNN)	104	186	11	3	95.394737	97.196262	94.416244
Decision Tree (KNN)	109	186	6	3	97.039474	97.321429	96.875000

Figure 5 Example of the same accuracy with different classifiers to the 6th decimal place 80% 20% split

	TP	TN	FP	FN	Accuracy %	Sens%	Spec%
Logistic Regression	143	294	18	1	95.833333	99.305556	94.230769
MLP	158	289	3	6	98.026316	96.341463	98.972603
KNN	141	286	20	9	93.640351	94.000000	93.464052
Decision Tree	155	286	6	9	96.710526	94.512195	97.945205
Logistic Regression (KNN)	146	297	8	5	97.149123	96.688742	97.377049
MLP (KNN)	149	300	5	2	98.464912	98.675497	98.360656
KNN (KNN)	139	294	15	8	94.956140	94.557823	95.145631
Decision Tree (KNN)	150	297	4	5	98.026316	96.774194	98.671096

Figure 6 70% 30% split

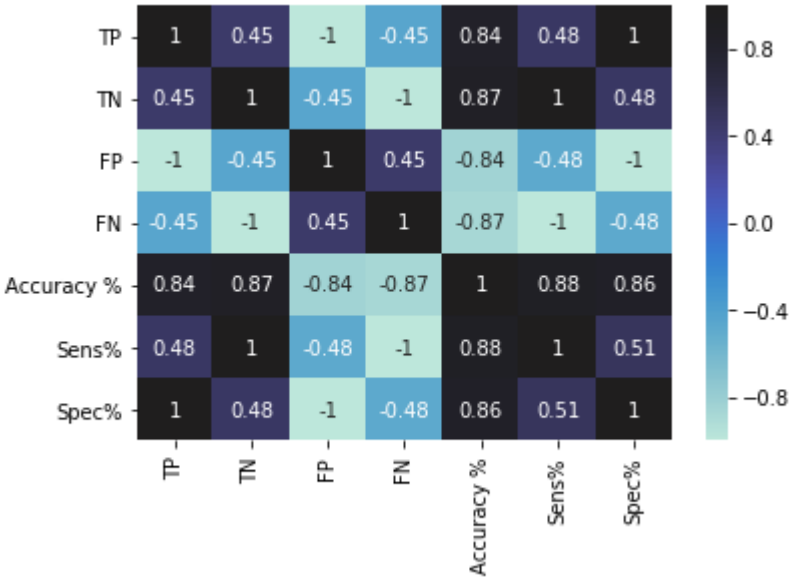

Figure 7

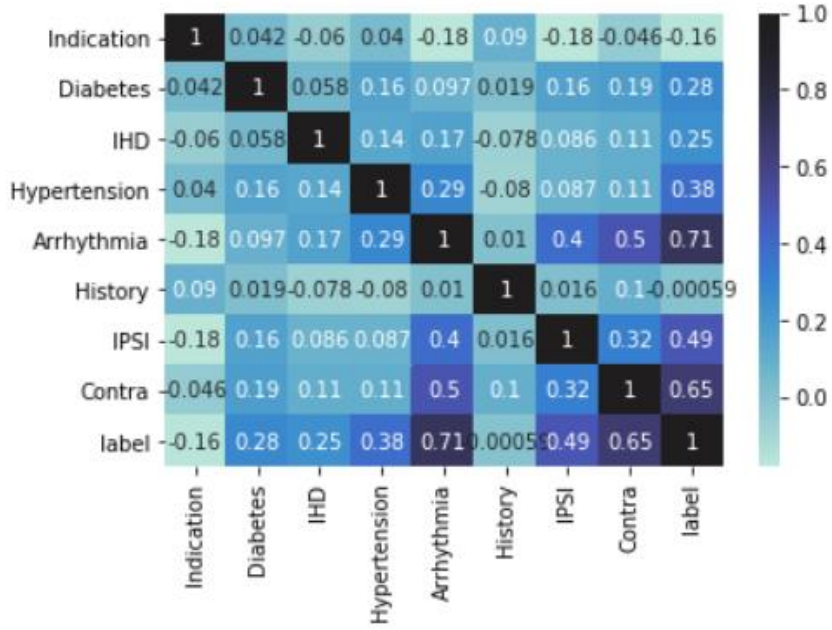

Figure 8 Correlations

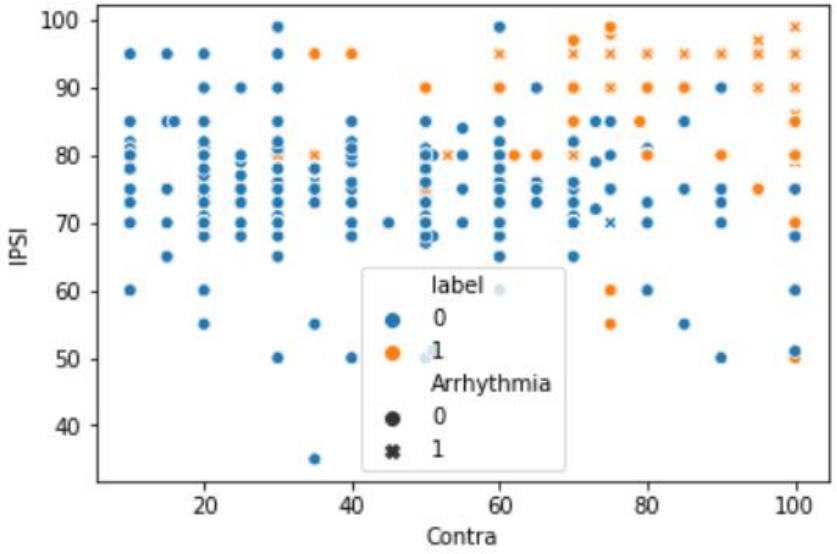

Figure 9 Scatter Plot For Arrhythmia, lebel, Contra and IPSI correlatoin