Big Data Coursework 2019 – 2020A

Greg MacPherson

S1509595

**“I declare that all work submitted for this coursework is the work of Greg MacPherson alone unless stated otherwise.”**

Introduction

This report will outline the implementation of a supervised machine learning algorithm.

The selected dataset is available from the UCI machine learning repository and it contains data concerning diagnosis of heart disease. The machine learning algorithm will be trained using this data set with the aim of being able to predict a diagnosis of heart disease in a patient.

Once the data has been prepared, the SVM will be trained using a subset of the attributes and tuning of the algorithm will take place. Adjustment of the SVM parameters will impact the results of the classification, with the aim to find the highest possible degree of correct classification.

Data Set Exploration and Pre Processing

The heart disease dataset comes courtesy of UCI machine learning depository and accessed on Kaggle.com [1]. It is a csv file that will be imported and explored in the Spyder IDE.

Before any pre -processing has been undertaken the dataset contains 294 rows with 14 attributes.

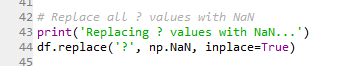
|  |  |  |
| --- | --- | --- |
| ***Attribute*** | ***Data Type*** | ***Scale*** |
| age – Age of patient | Numerical | Years |
| sex – Sex of patient | Nominal | 1 = male; 0 = female |
| cp – Chest Pain Type | Nominal | 1 = typical angina; 2 = atypical angina; 3 = non-anginal pain; 4 = asymptomatic |
| trestbps – Resting Blood Pressure | Numerical | mm/Hg |
| chol - Cholesterol | Numerical | serum cholestoral in mg/dl |
| fbs – Fasting Blood Sugar > 120 mg/dl | Nominal | 1 = true; 0 = false |
| restecg – resting electrocardiographic results | Nominal | 0 = normal; 1 = abnormality; 2 = showing probable or definite left ventricular hypertrophy |
| thalach - maximum heart rate achieved | Numerical | bpm |
| exang – exercise induced chest pain | Nominal | 1 = yes; 0 = no |
| oldpeak - ST depression (ST is a segment of a patient’s ecg waveform) induced by exercise | Numerical | Degrees |
| slope - the slope of the ST peak segment during exercise | Nominal | 1 = Upsloping; 2 = Flat; 3 = Downsloping |
| ca - number of major vessels (0-3) coloured by flourosopy | Numerical | Major vessels |
| thal – Thallium Stress Test Result | Nominal | 1 = fixed defect; 2 = normal; 7 = reversible defect |
| target | Nominal | 0 = disease; 1 = no disease |

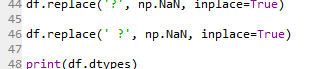
*Figure 1 – dataset contents*

All 14 attributes in the set either take numerical or nominal data represented with numbers. The purpose of the data is to record patients that are admitted to hospital with chest pain and are being treated for possible heart disease. Various test results are recorded and a diagnosis of heart disease is represented by the ‘target’ column. The machine learning algorithm used in this project will aim to predict the value of this column based on the supporting data.

Pre-processing must be carried out before the data can be used in our SVM - it is necessary to do this to ensure that the data is as accurate and suitable as possible.

Observing the heart disease dataset we can see that there are a large number of missing values. These are represented by a ‘?’ symbol. In particular, columns ’ca’, ‘thal’ and ‘slope’ are rife with missing values. As the data throughout the set is entirely numerical in nature, these values will be removed and replaced with NaN.

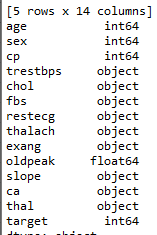




*Figure 2 –replacing missing values with NaN*

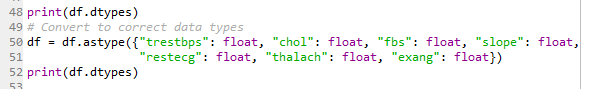
The .unique keyword is used on every column to ensure that no unexpected non-numerical values are present. Several missing values were found saved as **‘ ?’** as well as **‘?’** so care is taken to change all of these.

A duplicate row was found, however the attributes are fairly average and unlikely to affect the training in a negative way, so it will not be removed from the set.



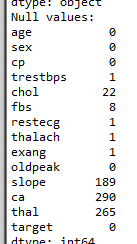
*Figure 3 - data types of dataset*

We can use dtypes keyword to output the data type for each column in our set. This shows that most of the columns are using object type, despite only holding numerical data (now that we have removed the ? characters). The columns are coerced to float type:



*Figure 4 – object types are changed to numerical type*

Null.sum is used to show the amount of missing values present in each column. Different strategies can be used to deal with missing values, like dropping rows or filling in the blanks with real data. The aim is to reduce the number of rows that need to be dropped from the dataset, and this shouldn’t be more than about 5% of rows. In the case of the heart disease dataset this would be about 14 rows.

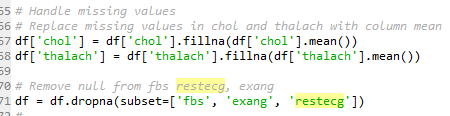


*Figure 5 – Null values for every column in the set*

The thal, ca and slope columns are much less valuable than the others – they have more missing data than valid data. With this volume of missing values, these columns can be dropped entirely from the set without having much of an effect on the results of the SVM.

Dropping these columns from the dataset gives a new shape of (293, 11) – only one row dropped.

The other 34 missing values are spread across six other columns. 22 are from the cholesterol column alone. The cholesterol column has numerical data which can be easily substituted for with a mean value. This is the preferable way to fix these missing values as removal would result in > 5% of the dataset being lost. The rest of the values will be dropped from the set, giving us a full dataset with no missing values.



*Figure 6 – removing null values from columns, replacing with the mean for chol and thal*

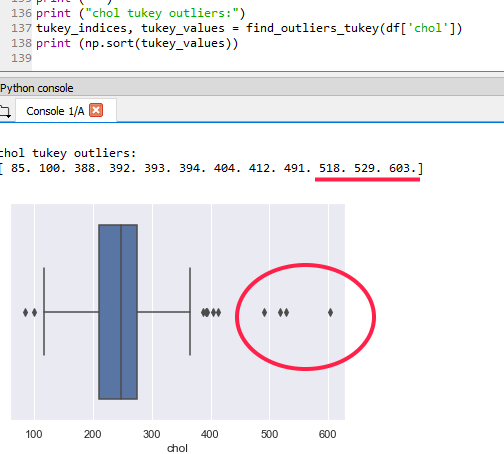
The new shape of the dataset is (283,11) meaning only 11 values have been dropped in total.

Outliers

It is now necessary to identify any outliers in the data and decide if these values should be removed or not. There are a few methods for finding outliers

The Tukey method uses the interquartile range of the data to highlight extreme outlying values.

A boxplot or scatter plot can also be used to visualise data and help identify outlying data. A combination of tukey analysis and a boxplot was used for every numerical column, and significant outliers were found for the cholesterol column:



*Figure 7 – tukey analysis results and a boxplot illustrating the outliers for the cholesterol column*

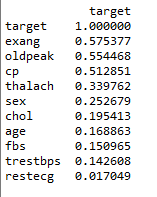
Healthy cholesterol levels are under 200 mg/dl, meaning that three instances in the dataset have recorded cholesterol levels at very high levels. The average value in the cholesterol column is 276 making the value of 603 an extreme outlier to our set. Some research on cholesterol levels indicates that a value of more than 500 mg/dl is rare but not impossible so it is likely these values are correct. [2][3] The values will be removed however as they will skew our data and make the algorithm’s predictions less accurate.

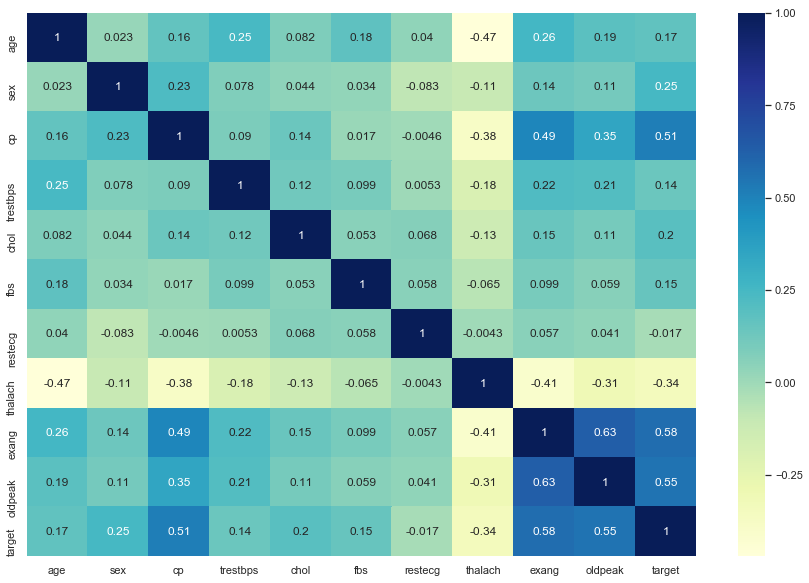
The values in the cholesterol column that were filled in with the mean value will need to be recalculated for the new mean value. With this done we are left with a new shape of (280, 11).

Feature Selection

With most data sets, a large number of attributes will often not correlate strongly with the target variable. If this is the case, it is wise to only include the columns that return a high correlation when training the SVM. This helps to reduce the complexity of the model and increase the speed at which the training process can be completed.

To help identify the features that correlate the most, a rank of the columns can be generated along with a heatmap:





*Figure 8 – Ranking of attributes by correlation to target variable, along with heatmap*

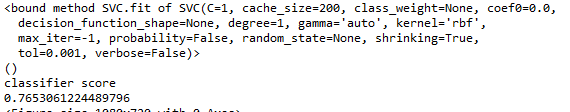
As our pre processed dataset only has 11 columns total – feature selection may not be necessary. As well as this, the correlations are not very strong even for the attributes that correlate higher.

SVM

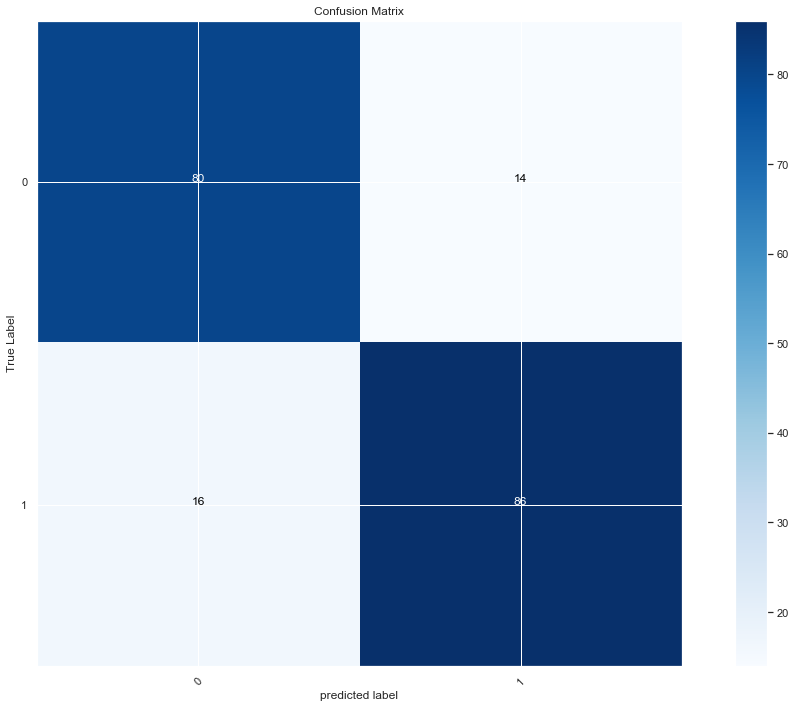
It is now time to use a machine learning algorithm on our data. First the dataset will be split into a training set and a testing set

An SVM was chosen to analyse the dataset. It is implemented with scikit-learn. The SVM will analyse the test data and attempt to predict the value of the target column for the remaining data. SVM was chosen as they can be highly accurate even with smaller data sets, however they need to be tuned correctly for the best results.

The SVM has parameters that can be changed to affect the algorithm’s results. First it will be run with default parameters:



A confusion matrix will also be generated to give a visualisation of the classifier score.



*Figure 9 - Confusion matrix showing 30 values incorrectly sorted*

Different values were used to find the optimal setting for each parameter, until the classification score was as high as possible:

|  |  |
| --- | --- |
| **Kernel** | Classifier Result |
| RBF (default) | 0.7551020408163265 |
| Linear | 0.8573469387755102 |
| Poly | 0.8418367346938775 |
| Sigmoid | 0.5204081632653061 |

Linear Kernel will be used as it gives the highest classification score. Other parameters will now be tried with the kernel set to linear.

|  |  |
| --- | --- |
| **C** | Classifier Result |
| 0.1 | 0.8418367346938775 |
| 1 (default) | 0.8573469387755102 |
| 10 | 0.8469387755102041 |
| 100 | 0.8671428571428571 |
| 1000 | 0.8671428571428571 |

C values between 100 and 1000 were found to be the most accurate. Some identical scores were outputted even with different values for C.

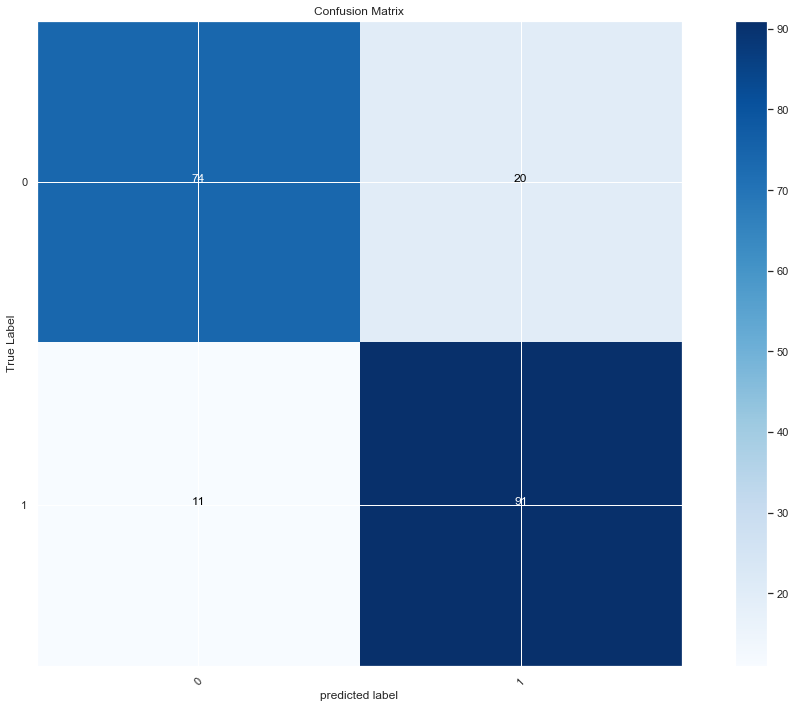
|  |  |
| --- | --- |
| **Gamma** |  |
| Auto (default) | 0.8573469387755102 |
| 1 | 0.8575510204081632 |
| 100 | 0.8524489795918368 |
| 1000 | 0.8775510204081632 |
| 10,000 | 0.8782448979591837 |

|  |  |
| --- | --- |
| Decision Function Shape |  |
| None (default) | 0.8573469387755102 |
| ovr | 0.8622448979591837 |
| ovo | 0.8469387755102041 |

Gamma of 1000 -10000 was found to give the highest classification score, along with a decision function shape ovr. Further changes of the test / train size gave different results. Altering the size of the test/train groups will affect the results of the classification:

|  |  |
| --- | --- |
| **Test / Train group size** |  |
| 40 / 60 | 0.8529411764705882 |
| 30 / 70 | 0.8313253012048193 |
| 20 / 80 | 0.8940677966101694 |
| 10 / 90 | 0.8783739837398373 |

A split of around 20/80 yielded the highest classifier score. With the final parameters decided through testing, the optimised classification can now be visualized with a confusion matrix:



*Figire 10 - Final confusion matrix giving a score of 0.8418367346938775*

DISCUSSION

The SVM was able to correctly classify a large proportion of the data correctly given a small dataset. The classification score of however is still low for the purposes of heart disease diagnoses – a subject where there is no room for error.

The split between training and testing was found to have a great impact on the results of the SVM, as was changing the kernel type. Further experimentation with the train/test split could have found a more accurate setup.

The pre-processing stage was very important given the amount of missing values in the dataset. Many missing values and even entire columns had to be removed from the set which could have provided valuable data to our model. The actions undertaken in the pre-processing stage should hopefully have improved the accuracy of the SVM and the value of the dataset. Without altering the values beforehand the results could have been quite far off and inaccurate. Abiding by the 5% rule meant that only a select number of rows could be dropped from the set. With a larger dataset, the data could hopefully be dropped with less of an impact on the set.

Tuning of the SVM parameters proved to be very valuable and helped to produce a more accurate classification. Even small changes to the parameters could yield dramatic differences in classification score. The compound effects of these parameters could have been further explored with a more rigorous testing procedure – testing each parameter option against each other option. However this would have been very time consuming and with a small dataset the results may not have differed a great deal.

Other methods for constructing a model could have been tested. An SVM was chosen as they perform well in cases where there are a low number of samples as well as being memory efficient. However another method such as a Multilayer Perception classifier could have been used. MLP is a feed forward neural network that has much quicker training time than SVM.S

### References

1 Heart Disease Data Set - https://www.kaggle.com/ronitf/heart-disease-uci

2 KOHLI, P. ‘The Recommended Cholesterol Levels by Age’ (2019) accessed 05/12 at [https://www.healthline.com/health/high-cholesterol/levels-by-age]

3 LAWES et AL. ‘High Cholesterol’ from ‘**Rethinking the “Diseases of Affluence” Paradigm: Global Patterns of Nutritional Risks in Relation to Economic Development’ (2005) accessed 05/12 at [**www.who.int%2Fpublications%2Fcra%2Fchapters%2Fvolume1%2F0391-0496.pdf&usg=AOvVaw0h4p6bCed9YamlWrgrFaOO**]**

Appendices

Appendix 1 – Main SVM Script

