Data Analytics- Coursework Report

Emma Parsley
40206111@live.napier.ac.uk
Edinburgh Napier University - Data Analytics (SET09120)

Abstract

This projects aim is to prepare, analyse and visualise the data historically recorded by a bank in Germany. The data will be analysed using algorithms such as classification and regression.

Keywords – data, preperation, mining, cleaning, visualisation

1 Introduction

The data analysed in this project will be the credits data set given to us for this poject. This data set contains 1000 rows each with 13 attributes. The data will need to be cleaned and prepared, before it can be analysed. OpenRefine will be used to clean and prepare the date, the program Weka will be used to analyse the data and it will be visualised using R.

2 Preparation

2.1 Cleaning

In order to clean the data the dataset was opened in OpenRefine. The most obvious problem with the data was that some of the fields under the attributes; checking_status, credit_history, saving_status, personal_status and job, had single quotes around them. This was easily rectified using the replace command in OpenRefine to transform all the cells in these coloums, replaceing the single quote with and empty string.

The purpose attribute contained many values that where the same except they were written or spelled differently. Any values that simply had single quotes around them were fixed in the previous step. A few values needed fixed so they were all spelt correctly and others just needed fixed so their use of case was consistent (e.g Education needed to become education).

From sorting the credit_amount attribute from highest to loweset it becomes clear that there are 8 rows of data with what seem like ridiculously high amounts of money for it's purpose, it seems likely that there were extra zeros at the end, however there is no way of knowing how many of the zeros were accidental so those rows were removed from the data set.

There were also many ages that were negative which was fixed by taking the absolute value of all the ages. There were also a few impossible ages, values less than 1 that were assumed to be correct without the "0." in front, for example 0.24 would become 24. After all this was sorted there were only 2 unlikely ages left 1 and 6, it seems likely that there was a number missed out when these were entered but as there as no way of knowing if these values should be 26 or 60 or something else these sets of data where removed.

The existing_credits attribute had many values that seemed incorrect. For example there shouldn't be able to be a value that is isn't in the set of natural numbers, so for example 1.1 was replaced with 1. It also seemed clear that someone shouldn't be able to a value that isn't a single digit, as anything greater than that seems excessive, so anything greater than 9 was replaced with it's greatest most common digit, for example 101 became 1.

Two entries have the job field filled out as "yes", this isn't a valid entry so these were replaced with the same job as other similar entries, for example the 37 year old single male that is using the loan for education, their job was replaced with "unskilled resident" as this is the same as someone with a similar entry and it does make sense that someone that wants money for education is likely to be unskilled.

Many entries in the num_dependants attribute where written out rather than entered as numbers, for example 1 was written as "One", these entries merely needed to be fixed to only be written as a number.

2.2 Preparation

To prepare the data to be used it needed to be converted into .arff format so it could be opened in Weka. OpenRefine has the ability to template how it will output, so the data can be exported straight from OpenRefine to a .arff compatible file. All numerical attributes can be changed from within Weka using the unsupervised Discretize filter to save a nominal only version of the database. This was done by running the data without the filter using J48 and checking the percent correctly classified and then applying the filter with a different amount of bins and running that with J48 until the percent correctly classified where close. The filter ending up having 2 bins with the original percentage correctly classified being 82.121% and the one after the filter being 81.616% with equal frequency being true.

3 Analysis

The class_no a attribute was removed for analysis as this is merely a unique identifier for each row of data and this wont be useful in the analysis.

3.1 Classification

The oneR method with 10 fold cross validation was used in this stage. Firstly ZeroR with 10 fold cross validation was run to get a base line percentage to test the rules created with oneR against. Both the nominal and the regular data set were used at this stage to see how the results differed. Rules were created based on the class attribute.

When using ZeroR are the nominal dataset gave a percentage of correctly classified instances as 70.292%. Running OneR without changing ant settings with this data set gave a percentage of 71.818%, which is above the baseline percentage given by zeroR which means that the rule is probably not just accurate on the training set. The rule given by this run looked at the credit history and stated, if no credits/all paid or allpaid then class is bad else class is good. This however seems like it should be the wrong way round as it is expected that someone who is all paid would be classed as good and someone who is has delayed previously would be bad.

As this rule didn't seem to work with common sense oneR was run again with different settings to try and find a more accurate result that appears to make more sense. However changing neither the batch size nor the minimum bucket size changed the results.

When using ZeroR on the regular dataset the percentage of correctly classified instances was 70.202%. Running OneR with default settings returned a percentage of 64.141% which is significantly lower than the baseline percentage calculated with ZeroR suggesting this result isn't particularly great. In order to try and get a better result the rule was looked at which had 26 different ways the result could be good or bad based on the credit amount, which seemed like it was probably overfitting the data so the minimum bucket size was increased.

At a minimum bucket size of 10 the correctly classified instances percentage exceeded that of the baseline by 0.707% creating the same rule about credit history that was found by the nominal set.

The minimum bucket size continued to be increased from 10 to find the maximum the correctly classified instance can be. At size 17 the percentage stopped increasing and it began to decrease at size 24. At size 17 the percentage was 71.818% which is the same as it was for the nominal data set. The rule created was also the same as that of the nominal data set so it seems likely that although it seems like it wouldn't make sense, this rule is mostly true for this data set.

3.2 Association

Apriori was used to associate this data. In order to use Apriori all data needs to be nominal so the nominal data that was prepared earlier was used. 20 rules were generated using Apriori with a confidence of at least 0.9. The 3 rules with the highest confidence value were as follows;

If the personal status is a divorced, separated or married female, and they are younger than 34 then they will have 1 dependant. This is the most confident rule with a 98% confidence rating.

The second rule is almost the same as the above. If the personal status is a divorced, separated or married female then they will have 1 dependant. This rule has a 95% confidence rating which is quite high, but if you wanted to know how many dependants a record is likely to have then it'd be better using the above rule as it's only one more attribute to check for 3% more accuracy.

If saving status is less than or equal to 100, and age is less than 34, and job is skilled then number of dependants is 1. This rule has a confidence of 0.95.

All the rules so far have said that if something then number of dependants is 1, which isn't all that surprising as 826 of the 990 records have 1 dependant. So in order to get more interesting results this attribute was removed. Running Apriori again with the same settings as before but with the number of dependants removed gives the following 6 rules the highest confidence;

If credit history is existing paid, and personal status is a divorced, separated or married female, and class is good then existing credits is 1. This rule has a confidence of 97% which, while lower than the highest rule that was obtained when including number of dependants, tells us more about the data as only 626 of the 990 records have 1 existing credit.

If checking status is no checking and credit amount is less than 2324 and age is greater than 33 then class is good. This rule has a confidence of 96%.

If credit history is existing paid, and employment is between 1 and 4 years and class is good then existing credits is 1.This rule has a confidence of 96%.

If checking status is less than 0, and credit history is existing paid and saving status is less than 100 then existing credits is 1. This rule has a confidence of 95%.

If checking status is less than 0, and credit history is existing paid then existing credits is 1. This rule has a confidence of 95%. This rule makes the rule above pointless, as they both have the same confidence rating and this rule is almost the same as the one above except it doesn't look at the saving status.

If checking status is no checking, and age is greater than 33, and job is skilled then class is good. This rule has a confidence of 94%.

All of the newly created rules either state that existing credits is 1 or that class is good, this makes sense as they are probably the values that appear most in the data set now that number of dependants has been removed, with 695 entries classed as good and 626 entries with 1 existing credit.

3.3 Clustering

Clustering is a descriptive way of viewing the data. Using a classes to cluster evaluation with both EM (expectation maximisation) and KMeans methods on the class attribute we can see how well people with good or bad credit ratings can be grouped.

Using the EM method to create 2 clusters of data we get 18% of the data in one cluster and 82% in the other with a log likelihood of -18.029. The classes to cluster evaluation assigns the first cluster to bad and the second to good. Which suggests to us that about 82% of the data sets have class good, however 375 instances(37.879%) where incorrectly clustered.

Using the KMeans method to create 2 clusters, 45% are in the first cluster and 55% percent are in the second with the clusters being assigned good and bad respectively. This seems like a much more even split, although this time bad has slightly more than good. However 435 instances(43.929%) where incorrectly clustered, 60 more than the EM method.

KMeans was run a few more times using different seeds as different seeds in KMeans can quite drastically change the results. The lowest amount of incorrectly clustered instances that was found doing this was 349(35.253%), less than with EM, with 77% being clustered as good and 23% being clustered as bad.

Allowing the EM method to decide for itself how many clusters to create causes it to create 4 clusters, no class (11%), bad (19%), good (49%), no class (21%). However this incorrectly clusters 575 instances(58.081%), worse than the other results.

The results from these clusterings suggest that it isn't obvious from any particular row of data weather it will be classed as good or bad, however it is more likely to be classed as good.

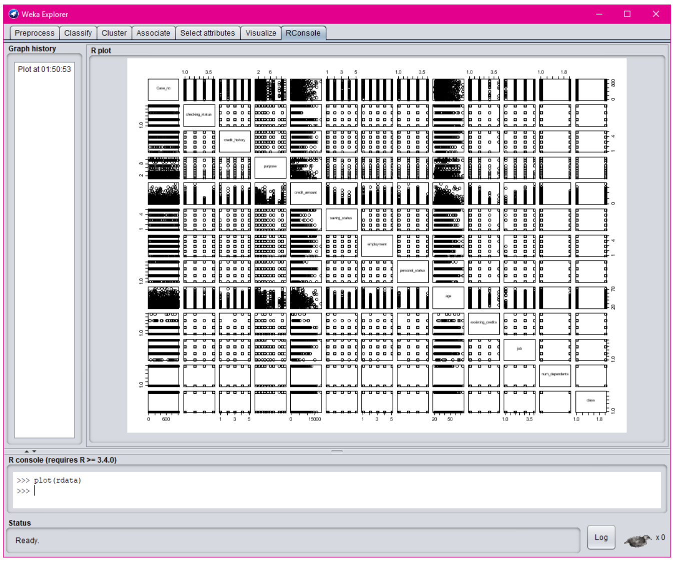
The EM clustering method was run a few more times ignoring certain attributes, such as ignoring everything that probably shouldn't have anything to do with their class, job, personal status, age, or ignoring everything but those, but the incorrectly clustered instances were never below 30%. The most accurate clusterings that were found were from only using the credit_history, credit_amount, saving_status and class attributes, 25% bad, 75% good with 336 (33.94%) incorrectly clustered.

4 Visualisation

R will be used to visualise this data set. In visualising this dataset the question, "How do the other attributes affect the class attribute in this data set?" as this is what has been explored for the most part in our analysis up to this stage.

A package was installed in weka to enable R to be used within Weka. See figure 1

Firstly credit history was created as a bar graph showing the amount of good and bad for each value in this attribute see figure 2. Looking at this graph it becomes clear that the rule generated earlier using OneR is correct, as most of the bars are mainly good except no credits/ all paid and all paid which are mainly bad. With this bar graph now though it is clear that there are a lot less records in no credits/ all paid and all paid compared to the rest and that the distribution within them while slightly skewed to bad is closer to 50, 50.



```
>>> plot(rdata)
>>> |
```

Ready.

Figure 1: **R in Weka** - This is the data set plotted within Weka

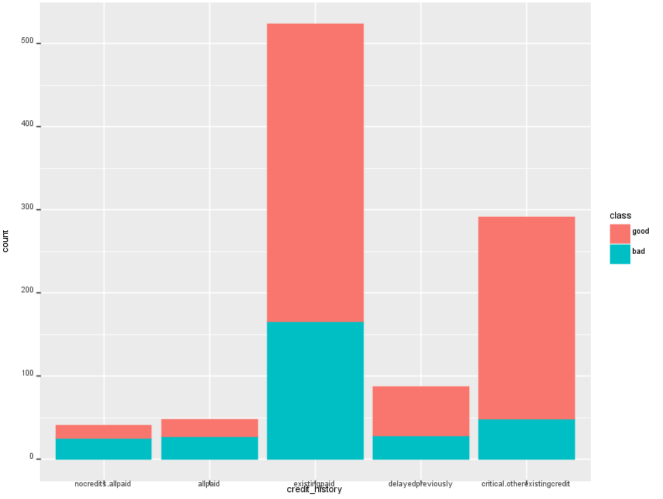


Figure 2: **Credit History Bar Graph** - This the amount of each value in credit history split by class

In attempt to find some more relationships with class the numerical attribute age was used to plot graphs that might give information on a correlation between age and class.

A density plot of age coloured by class, see figure 3, shows that there are more people aproximately between the ages of 20 and 30, that are classed as bad than at any other age.

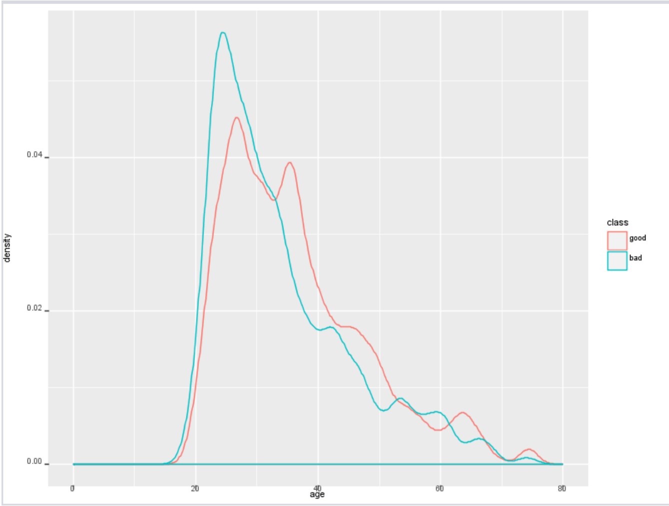


Figure 3: **Density of age by class** - A density plot of age with lines coloured by class

After creating a few more graphs, see figure 4, using age as the x attribute and changing the variable on the y axis. From these there seemed to be little correlation between age and anything else. but from looking at the points generated on the graph of age and job coloured by class, it looks like unskilled residents are more likely to be classed as good than more skilled workers, see figure 5.

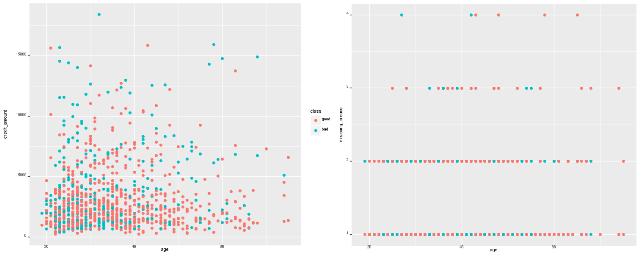


Figure 4: **Other graphs** - Some of the other graphs created while looking for patterns in the data

Before answering the question it shall be refined to "How does job type affect class?". Three different possible solutions were sketched out, a bar graph, a point graph and a density graph. See figure reffig:sketches

A bar graph would show all the data as clear as the bar graph of credit history earlier. It clearly groups the data so it's is very visually easy to see.

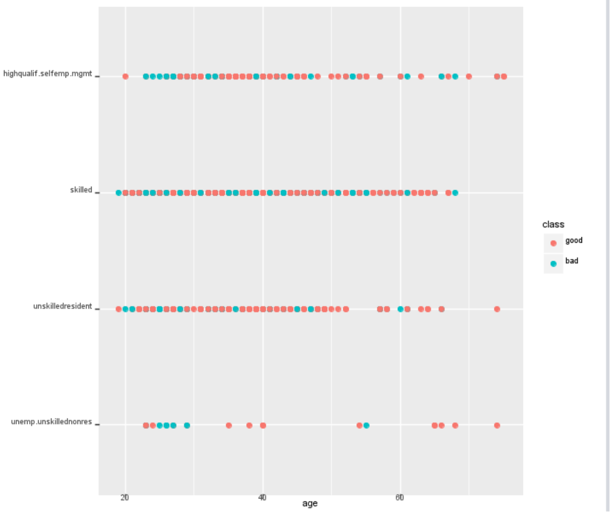


Figure 5: **jobs against age graph** - a graph with age on the x axis, job type on the y and is coloured based on class

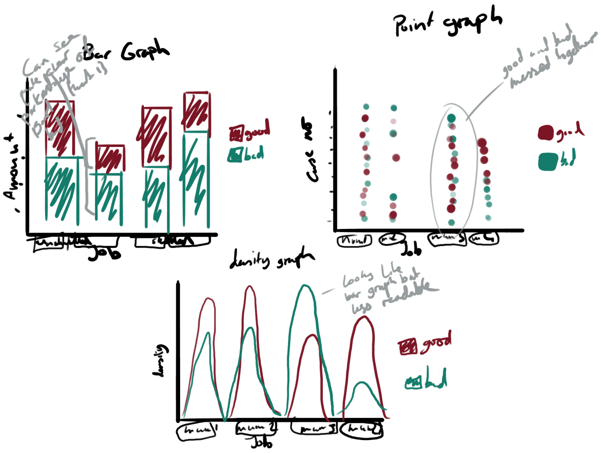


Figure 6: **Sketches** - Sketches for solution visualisation

A point graph is difficult to do because what should be in the y axis. The case number could be used here but it will still have all the good and bad cases mixed together so it will be hard to visually see the answer to the query.

A density graph shows pretty much what we want to see, but to see it clearly it will need to be adjusted for each section to be quite small, which will end up with it having peaks in the same places a bar graph would show the data as just bars.

It is clear from this that a bar graph is the best solution as it will display the data in the most clear and readable format. It should have a key for what colour the classes are on the side, the nominal job values written along the bottom and the amount indicated on the left.

The final visualisation, see figure 7, shows that a higher percentage of all the unskilled people are classed as good out of all of the unskilled people, than the percentage of skilled people classed as good out of all the skilled people. This answers the question, how does job type affect class?, as we can see that while all job types are mostly good, a higher percentage of the unskilled residents are good, so the most likely job type for someone to have if they are classed as bad is skilled.

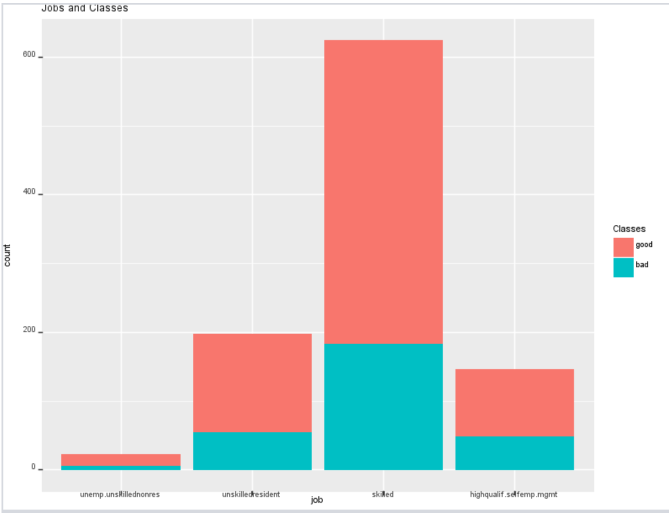


Figure 7: **Final Visualisation** - Visualisation from which the question can be answered

5 Analysis Conclusion

This analysis has shown that there while there is a lot of attribute values you may think would cause a record to have a class attribute of bad there is really no strong connection between any attribute and class.

The OneR rule where the most instances are correctly classified creates a rule the complete opposite of what most people would expect given a relationship between credit history and class.

Apriori gave the most convincing rules between attributes and class, however those rules where exclusively equated to class being good and never bad and it is the more likely scenario in general for any random record to have a class of good rather than bad as there are more good records.

Clustering is where it became really obvious how weak the link between class and the other attributes are, as although it did manage to group the data together into good and bad clusters, the percentage error was quite high and didn’t show a strong link.

Overall clustering was the most helpful analysis technique for this data set as the correlation that was being looked for was simply not there. This combined with the other techniques gave the clear overall conclusion that there is no strong link between the class and the other attributes. Visualisation made us able to see exactly how week the correlations were as we could see how difficult it was to see any of these correlations visually.