Report

Dan Nash
40217045@napier.ac.uk
Edinburgh Napier University - Module Title (SET09120)

1 Introduction

We have been tasked to clean, prep and analyse a large set of data. This is so we can then identify interesting patterns within the data and give the bank a data driven informed decision, on whom they should give a loan out to.

2 Data Cleaning

The Author used the program "OpenRefine" during the data cleaning stage. When the data has been cleaned, and the relevant data types have been declared either numerical or nominal, then it is ready to be prepped for Weka.

2.1 OpenRefine

When the data file (excel spread sheet) is first downloaded, it must be converted into a CSV file format, instead of it's standard xlsx format. Unfortunately by doing this, any numeric data type (column) is then transformed into a text value. The affected data types can be converted back it's original form within OpenRefine though. Before loading the CSV file into OpenRefine, make sure to place the appropriate headers over the correct columns of the spreadsheet. This is so the user can maintain an understanding of what the data in each column actually represents.

Start OpenRefine and start a new project. Select the CSV file and then click begin. The data will then present it's self in a very similar fashion as if it was Excel. Before we start transforming the relative data columns into it's respective data types; "num_dependents" to numerical. We must first clean the data in the affected columns.

In "**num_dependents**", select text facet and then you'll see a number of discrepancies located within the data set. The following discrepancies need to be "edited"; "1one" and "one" need to be edited to "1". "two" and "twotwo" need to be edited to "2". This column should now be cleaned and it can be "common transformed" into a numeric data type.

"**Purpose**" column needs to be text facet as well. The following discrepancies need to be fixed; "busness" and "busines" to "buisness". "Eduction" to "education". "ather" to "other". You will then see 10 choices left, but when looking at the assignment brief you'll notice that there are 11 entries. "Vacation" is the missing entry, but it's never marked in any of the "case_no" entries for "purpose". So you don't need to worry about it being missing.

In "**job**", select text facet and look at the discrepancy "yes". There are 2 instances of "yes", so select "yes" and then it will show you the data rows for both of those entries. The issue here is we must identify how "yes" is interpreted. The reason being is that there are 4 options for "job" so there are many ways we can identify "yes". After reviewing them, I decided to place them in "unskilled resident", the reason being was that one of them had "<100" savings and was relatively young "26". The other "case_no" was older "37" and had "<100" savings, but his "purpose" was education. Which gave me the impression that he was needing a loan to pay for education, which would lead him to be a "skilled" or "high qualif/self emp/mgmt". So edit "yes" to "unskilled resident".

Common Transform "**credit_amount**" into numeric, then do a "numeric facet" on the same column. There are 5 records that are over the amount of 10,000,000 which is an obscene amount and has to be a clerical error. I then proceeded to remove all of the zero's for each case. 4 of the 5 data entries will now be fixed and within an appropriate "credit_amount" range, but the 5th one (case_no 432) will still be too high. You must numeric facet the column again and set the search range from between 100,000 to 7,200,000. After doing so, 4 rows will appear, with one of them being "case_no 432" again. Ignore case "432" for now and remove three "0" on the other "credit_amount" cases. I say remove 3 zero's instead of 4 because if you look at their purpose then the amount they need will be above 1000. With case number 432, remove the first digit "1", and that will bring the credit value down to "11328", which would match his profile because his "purpose" is "other" and he's in the "high qualif/self emp/mgmt" for the "job" category.

In the "**existing_credits**" column, do a text facet, and then proceed to remove any last digit with numbers that contain 2 digits, i.e 11 = 1. Any entities with 3 digits i.e 333, then remove the last two digits. Any numbers that are represented like this; 0.1 , then remove the "0." from the entities. After these edits you should be left with four entities that go as follows: 1, 2, 3, 4.

Common Transform "**age**" to numeric, then do a numeric facet. Set the range from -40 to 0, after doing so, proceed to edit out any of the "-" values. Do another numeric facet, set the range from 0 to it's maximum, from here you'll see values with decimal points, i.e 0.1. Remove any decimal points and then do another numeric facet setting the range from the first instance "6" to it's maximum range. There will be two instances which seem our of place. These instances are "6" (case_no 26) and "11" (case_no 54). Instead of removing them, I investigated into what their possible age bracket could be. I looked at their "employment" and "credit_history"; both had employment in the "1<=X<4" range and had "existing_paid" within the "credit_history" category. Going by this I believed it was fair to place a "2" in front of their original values, which would place them within the "20's" bracket. Finally do another numeric facet with the range of 80 to 340. Any entity that's beyond the age of "100", then remove the final digit to place them under the age of "100".

The data cleaning should now be complete once going through all these steps.

3 Data Preparation

Now it's time to prep the data into the correct formats so that Weka can work with them. First off we need to create a standard data set which will contain both numeric and nominal data. So, make sure to common transform the following data columns (if have not previously been done by now); "**case_no**", "**credit_amount**", "**age**", "**existing_credits**" and "**num_dependent**". Then proceed to common transform all the other data columns into text, this is purely just for insurance purposes. There is also no need to text facet or numeric facet any of the other data columns that have not been inspected already. They have already been checked and there is no discrepancies.

Export the project to a Excel document in CSV format, and then proceed to open it in a "Word" document. Proceed to save the document and then open it in a text editor of your choice, I stuck with Notepad. From here you have to reposition the data and add particular value, so that when it's turned into a ".arff" file, Weka can then read it with no errors occurring. Review **appendix 1** to understand how the txt file should look. The main points to understand is that any previous columns is now designated with a "@attribute" and that any "attributes" that deal with "nominal" data must have a their data objects i.e "tv/radio", "0<=X200"; stated in the curly brcaes, which are located in their relevant attribute fields. Review **appendix 2** to understand. Any data type that's numeric, simply put "real" right after stating the attribute name. Review **appendix 3** to understand. Once this is done convert it into an arff file and Weka will now be able to work with it.

3.1 Numeric to Nominal

We need a ".arff" file with mostly nominal data. Certain algorithms like "Apriori" within "Association" can only work with nominal data. So, go back to the OpenRefine project and start to turn any numeric data into nominal.

"**num_dependents**" and "**existing_credit**" can simply just be turned into nominal data by doing Common Transform > To Text.

To turn larger data sets like "**age**" and "**credit_amount**" into nominal data, we must start applying the data into clusters, to which we can then express with an appropriate nominal value.

Select "**age**" and use a numeric facet, select the range 19 to 26 then Edit Cell > Transform the selected range to "18<=X<26". From here on, repeat the previous steps but work with range increments of 10. For example, 26 to 36 ("26<=X<36"), 36 to 46 ("26<=X<46"). Do this till you get to the end of the range "76". By doing this, the data will be turned into "text" data and will be identifiable clusters of ranged data that we can work with.

Select "**credit_amount**", and do a numeric facet. We follow almost the same steps that we did for **age** but we break down the range to 2000 increments. I.e. 0 to 2000 will be "0<=X=2000". We do this all the way up to 20,000. Also, there are no records of data within the range of 16,000 to 18,000. So, do not worry about trying to create this range. You will be left with 9 instance categories of "**credit_amount**", which can be seen in **appendix 4**.

All data (apart from **case_no**) has been turned into nominal data. Export the file into a excel ".xlsx" file and then open it in word. Save the the file and then open it in your preferred text editor. Similar to what we did previously in (**appendix 1**), we now have to state which "attributes" are "real" (numeric) and which attributes are nominal ("26<=X<36", "26<=X<46"). **case_no** should be the only "real" attribute there. Look at **appendix 5** to see how the ".txt" file should look. Now save thew file and convert it into a ".arff" file. You now have a mostly nominal ".arff" file which can be interacted within Weka.

4 Weka

The three areas we'll be analysing the data from our newly created ".arff" files are "**Classification**" with the **48** Algorithm, "**Association**", with the **Apriori** Algorithm and finally "**Clusters**" with the **simpleKmeans** algorithm.

4.1 Classification

Load the nominal values only ".arff" file into weka. When using the classification rule set, having a mix of numerics and nominals can muddle the data a bit. Once the file has been loaded, go to the "Select Attribute" tab located at the top right of the Weka interface. Then choose "CorrelationAttributeEval" (evaluates the worth of an attribute by measuring the correlation between it and the class) from the "Attribute Evaluator" terminal. Make sure "(Nom) class" is chosen and then click start. The following results will then be produced; Review Appendix 6

From here, we identify the ones which are ranked poorly (highest **entropy**), "**age**", "**num_dependents**", "**case_no**" and "**job**". This is so we can then remove them from our data set before proceeding to the **Classification** stage. Go back to the "Preprocess" tab and proceed to remove the previously mentioned attributes from the data set.

Go to the "**Classify**" tab on the Weka interface and choose the "**J48**" algorithm from "trees" directory within the "Classifier" interface. Click on "J48" and a new menu will appear, look for "MinNumObj" then change the value from 2 to 10 (**Appendix 7**). After the changes, exit that sub menu and make sure "(Nom) Class" is selected, then click "Start" to begin the calculations.

From output screen there are a number of factors to consider, they are the "**Correctly and Incorrectly Classified Instances**" and the "**Confusion Matrix**". From looking at the correctly and incorrectly classified instances (**Appendix 8**), you can see that 73% instances have been calculated correctly and 27% incorrectly. This percentage ratio is not ideal, but even if you decrease (reducing the "MinNumObj" number) the number of leaves and size of the tree to below the 30's (currently at Leaves:

40, size of Tree: 46). These percentages only changes from 1 to 3%. From looking at the "**Confusion Matrix**" (**Appendix 9**), you can see that 623 instances has been predicted as good and a 107 instances have been predicted classified as bad. But upon further inspection it is stated that in fact 77 instances were identified as bad when in fact they were good, but more alarmingly, there is 193 instances which are stated as good, when they are actually bad. Obviously the assessment of incorrectly identified "good" instances takes more precedence than the incorrectly predicted "bad" instances due to the larger size difference.

Right click on the "Result List" and select visualize tree, from here (**Appendix 10**) you should be able to identify some common rules within the data presented.

Rule 1: **If** Checking Status = No Checking, **Then** Class = good. This rule has a coverage of 394 with an accuracy of 88%

Rule 2: **If** Checking Status = 0<=X<200 **And** Credit Amount = 0<=X<2000, **Then** Class = good. This rule has a coverage of 106 and an accuracy of 62%.

Rule 3: **If** Checking Status = <0 **And** Credit History = Existing Paid **And** Purpose = New Car, **Then** class = bad. This rule has a coverage of 42 with an accuracy of 64%.

Rule 4: **If** Checking Status = 0<=X<200 **And** Credit Amount = 2000<=X<4000, **Then** Class = good. This rule has a coverage of 77 with an accuracy of 72/

Rule 5: **If** Checking Status = >=200, **Then** Class = good. This rule has a coverage of 63 with an accuracy of 77%.

Rule 6: **If** Checking Status = <0 **And** Credit History = Critical/Other Existing Credit, **Then** Class = good. This rule has a coverage of 67 and an accuracy of 73%.

What can be taken from looking at these rules is that having "No Checking" greatly increases your chance to receive a loan. This is probably due to the fact that if the bank does not already have you as a customer, then they want to make you a customer.

5 Appendix

```
@relation germanfinancedata

@attribute case_no real
@attribute checking_status {'<0', '0<=X<200', '>=200', 'no checking'}
@attribute credit_history {'no credits/all paid', 'all paid', 'existing paid', 'delayed previously', 'critical/othe
@attribute purpose {'new car', 'used car', furniture/equipment, radio/tv, 'domestic appliance', repairs, education,
@attribute credit_amount real
@attribute saving_status {'<100', '100<=X<500', '500<=X<1000', '>=1000', 'no known savings'}
@attribute employment {unemployed, '<1', '1<=X<4', '4<=X<7', '>=7'}
@attribute personal_status {'male div/sep', 'female div/dep/mar', 'male single', 'male mar/wid', 'female single'}
@attribute age real
@attribute existing_credits real
@attribute job {'unemp/unskilled non res', 'unskilled resident', skilled, 'high qualif/self emp/mgmt'}
@attribute num_dependents real
@attribute class {good, bad}
```

Figure 1: **Appendix 1** - Data Prep, Nominal and Numeric

```
@attribute checking_status {'<0', '0<=X<200', '>=200', 'no checking'}
```

Figure 2: **Appendix 2** - Data Prep, Nominal attribute value

```
@attribute age real
```

Figure 3: **Appendix 3** - Data Prep, Numeric attribute value

```
credit_amount
9 choices  Sort by:  name  count
'0<=X<2000' 429
'10000<=X<12000' 19
'12000<=X<14000' 9
'14000<=X<16000' 11
'18000<=X<20000' 1
'2000<=X<4000' 322
'4000<=X<6000' 99
'6000<=X<8000' 80
'8000<=X<10000' 30
Facet by choice counts
```

Figure 4: **Appendix 4** - Data Prep, Credit Amount instances

```
@relation germanfinancenominaltonumeric

@attribute case_no real
@attribute checking_status {'<0', '0<=X<200', '>=200', 'no checking'}
@attribute credit_history {'no credits/all paid', 'all paid', 'existing paid', 'delayed previously', 'critical/othe
@attribute purpose {'new car', 'used car', furniture/equipment, radio/tv, 'domestic appliance', repairs, education,
@attribute credit_amount {'0<=X<2000', '2000<=X<4000', '4000<=X<6000', '6000<=X<8000', '8000<=X<10000', '10000<=X<1
@attribute saving_status {'<100', '100<=X<500', '500<=X<1000', '>=1000', 'no known savings'}
@attribute employment {unemployed, '<1', '1<=X<4', '4<=X<7', '>=7'}
@attribute personal_status {'male div/sep', 'female div/dep/mar', 'male single', 'male mar/wid', 'female single'}
@attribute age {'18<=X<26', '26<=X<36', '36<=X<46', '46<=X<56', '56<=X<66', '66<=X<76'}
@attribute existing_credits {1, 2, 3, 4}
@attribute job {'unemp/unskilled non res', 'unskilled resident', skilled, 'high qualif/self emp/mgmt'}
@attribute num_dependents {1, 2}
@attribute class {good, bad}

@data
```

Figure 5: **Appendix 5** - Data Prep, Nominal txt file

```
Ranked attributes:
 0.23276    2 checking_status
 0.13162    6 saving_status
 0.08988    3 credit_history
 0.07494    4 purpose
 0.07192    8 personal_status
 0.06506    5 credit_amount
 0.0527     7 employment
 0.04746    9 age
 0.04205   10 existing_credits
 0.03461    1 case_no
 0.01973   11 job
 0.00301   12 num_dependents
```

Figure 6: **Appendix 6** - Classification, Ranked Attributes

```
minNumObj   10
```

Figure 7: **Appendix 7** - Classification, MinNumObj

```
Correctly Classified Instances         730               73      %
Incorrectly Classified Instances       270               27      %
```

Figure 8: **Appendix 8** - Classification, Instances

```
=== Confusion Matrix ===

   a   b   <-- classified as
 621  79 |   a = good
 200 100 |   b = bad
```

Figure 9: **Appendix 9** - Classification, Confusion Matrix