Data Analytics Coursework

Louis Boursier
40404293@napier.ac.uk
Edinburgh Napier University - Module Title (SET09120)

Keywords – Louis, Boursier, Data analytics

1 Introduction

This document is a report for the second coursework in the data analytics module. For this coursework, we were asked to use OpenRefine and Weka to extract patterns from a given data set. We also had to use the lecture and the additional literature to make a better use of those tools. We had to understand the data, clean it and convert it for a Weka usage. Then, we had to use different algorithms to find patterns around the success of a bank loan for a costumer.

2 Data preparation

2.1 Data Cleaning

The data cleaning has been done thanks to OpenRefine. For each columns, I made sure that the each values has its good type. For example, the age should be a numeric attribute.

Then, I clustered the same values with different names. For example, "Radio/Tv" should be merged with "radio/tv". To do so, I used a facet to identify same values with different labels. I used the built-in key collision and nearest neighbor methods with different parameters to cluster all the common labels. Once the common labels identified, I merged them into the label I chose as the most appropriate.

After that, I tried to identify outliers. If I was not sure that those values were not due to an input mistake, I deleted them. To do so, I used numeric facet. This allowed me to correct some values. For example, I changed the age "-22" to "22" because the minus can only be a keyboard input error. But for the credit amount with a huge value, I deleted them, because I was not able to know if those values were a mistake or not. For example, I had some credit amount with values like 128000000. I did not know if the big bumber of zeroes were due to a different unit, or if it was a mistake, or even a good value.

Finally, I looked for wrong type of values or blank values in the cells, and deleted them. I also deleted nominal values for which their value was not corresponding to the possible value. For example, I deleted all value in the class column that were not either "good" or "bad".

2.2 Data conversion

For the future use of the data set, I wanted three different format: the original cleaned format, the format with only nominal values, and the format with only numeric values. This is because some algorithms like the association algorithm "apriori" only work with nominal attribute. The same is true for linear regression algorithm, which only work with numeric values. So I had to convert the data set into a nominal only and a numeric only format.

The nominal only format was done thanks to OpenRefine. Only two columns were numerical: the credit amount and the age. I used GREL expression to transform the numeric values into nominal. I splited the age into the following categories: less than 20, less than 25, less than 30, less than 40, less than 50, less than 60, less than 70 and more than 70. I did the same for the credit amount. I split it into those categories: less than 1000, less than 2000, less than 3000, less than 4000, less than 5000, less than 6000, less than 7000, less than 8000, less than 10000, less than 15000, and less than 20000. Because of the previous deletion, I had no credit amount over 20000.

I also used OpenRefine for numeric conversion. As explain in the lecture, I added a binary column for each values of a nominal column. Those added column contains a 1 in their cells if the row contained the nominal value this new binary column represents. For each non numeric column, I created a new binary column for each of the possible values. Then, I deleted the nominal column because it was not necessary anymore. At the end, I gained many more columns than the original data.

The final process was to convert those three CSV files to ARFF so that Weka could use them. For that purpose, I used the Weka tool called "ARFF viewer".

3 Data Analytics

3.1 Classification

For the analysis, I removed the first column called Casedno in Weka because it is only an index and it is not related to the values. For the classification, I used the J48 algorithm with the cleaned ARRF file. It gave me a tree that I simplified to displayed only 6 rules. I simplified it by saving the six leaves with the more instances. That way, the tree covers the maximum instances. Because there are 988 instances in the final data set, the coverage of each rules is its number of instances divided by 988. For example the first rule has a coverage of 67/988. The confidence is given by the ratio in parenthesis. Indeed, the second value in the parenthesis correspond to the misclassified instances.

We can see that the loan is more likely to be given if the checking status is good, or if the customer does not have any account in the bank. If the checking status is less or equal to zero, and the credit history is existing in another bank, the credit is also likely to be given. If the checking status is between zero and two hundred, the loan is likely to be given as long as the credit amount is equal or less than 9283.

Listing 1: J48 results

```
1 checking_status = <0
2 |   credit_history  = critical/other existing credit: good (67.0/18.0)
3 checking_status = 0<=X<200
4 |   credit_amount  <= 9283: good (244.0/86.0)
5 |   credit_amount  > 9283: bad (21.0/4.0)
6 checking_status = no checking: good (388.0/43.0)
7 checking_status = >=200: good (63.0/14.0)
```

3.2 Association

For the association algorithm, I used the Apriori algorithm with the nominal data set. The default configuration with a 90 percent confidence gave me 9 rules. I chose the four first ones because their confidence are the best, and I also selected the last two because their coverage are bigger for just 2 percent less confidence. This gave me the following six rules:

Listing 2: Apriori results

```
Minimum support: 0.1 (99 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 18

Chosen rules:
1. checking_status=no checking purpose=radio/tv 125 ==> class=good 118
<conf:(0.94)> lift:(1.34) lev:(0.03) [30] conv:(4.65)
2. checking_status=no checking employment=>=7 114 ==> class=good 107
<conf:(0.94)> lift:(1.34) lev:(0.03) [26] conv:(4.24)
3. checking_status=no checking credit_history=critical/other existing credit 153 ==> class=good 143
<conf:(0.93)> lift:(1.33) lev:(0.04) [35] conv:(4.14)
4. checking_status=no checking personal_status=male single job=skilled 149 ==> class=good 138
<conf:(0.93)> lift:(1.32) lev:(0.03) [33] conv:(3.69)
5. checking_status=no checking job=skilled 261 ==> class=good 236
<conf:(0.9)> lift:(1.29) lev:(0.05) [52] conv:(2.99)
6. checking_status=no checking personal_status=male single 229 ==> class=good 207
<conf:(0.9)> lift:(1.29) lev:(0.05) [46] conv:(2.96)
```

Here is how to interpret the first line: if the checking status is no checking (meaning that the customer has no current account in the bank) and the purpose of the loan is a radio/tv, 125 instances correspond to this case and 118 of them have an accepted loan (the class value is "good"). Because the number of instances is 988, the coverage for the first rule is 128/988 and the confidence us 118/125. We can do the same . We can do the same calculation with the other lines.

3.3 Clustering

For the clustering algorithm, I chose the density base clustering method with the model and evaluation on the training set. The normal cleaned data file was used.

Listing 3: Clustering results

```
Cluster 0 (good): 'no checking','critical/other existing credit','new car',credit_amount_less_2000,>=1000,>=7,'male single',age_less_70,skilled
Cluster 1 (bad): 0<=X<200,'all paid',business,credit_amount_less_4000,100<=X<500,4<=X<7,'male single',age_less_40,skilled

=== Model and evaluation on training set ===

Clustered Instances

0      784 ( 79 percent)
1      204 ( 21 percent)


Log likelihood: -12.65365


Class attribute: class (loan is given or not)
Classes to Clusters:

   0   1  <-- assigned to cluster
 582 112 | good
 202  92 | bad

Cluster 0 <-- good
Cluster 1 <-- bad

Incorrectly clustered instances : 314.0  31.78 percent
```

The clustering has been made around the class attribute. It has identified some values has more likely to appear with a successful loan, and other more likely to appear for a rejected loan. We can see that the following features contribute to have its loan accepted: not being a member of the bank, having an existing credit in another bank, asking a loan for a new car, having a credit amount between 1000 and 2000, having a saving status greater or equal to 1000, having an employment for seven years or more, being a single male, being aged between 60 and 70 years old, and being

a skilled worker. We can see that the following features contribute to have its loan rejected: having a checking status between 0 and 200, having no debt taken or all debts paid back duly, having a business purpose for the loan, having a credit amount between 3000 and 4000, having a saving status between 100 and 5000, being employed for four to seven years, being a single male, being aged between 30 and 40 years old, and being skilled.

Although these values may seem contradictory, hindsight can help us to better understand them. For example, more people being skilled are likely to ask for a loan, that is why it appears that they are also more likely to see their loan rejected. If being a single male is more likely to be true when the loan is accepted and rejected, it is because the majority of loan request are asked by single males.

4 Conclusion

With the coursework, I have been able to see what tools using to interpret data, and how to use them. I also applied the knowledge I gain in the data analysis thanks to the lecture to choose the appropriate cleaning method and data analysis algorithm. I have leveraged different kind of algorithms to extract relationships in the data. I have understood that even with the best tools, understanding the data is crucial to be able to interpret the results.