Edinburgh Napier UNIVERSITY

Data Analytics Coursework 2

Marcos Ramirez Aceves
40284094@live.napier.ac.uk
Edinburgh Napier University - Data Analytics (SET09120)

1 Introduction

The aim of this project is to perform an exploratory data mining of the data provided. This will be achieved by using different software tools like OpenRefine and Weka in combination with the different techniques and approaches seen in the lectures and practicals. The following sections in the report cover in detail the complete process with the exception of the first point as the data is already provided. The data mining process consist in the following steps:

- Data gathering.
- Data preparation and cleaning.
- Pattern extraction and discovery.
- Data visualitation.
- Evaluation and Analysis of results.

After reading the metadata and having a clear understanding of the different attributes and their values data preparation can be started.

2 Data preparation

2.1 Data Cleaning

Data cleaning is the first step in the process, it basically consists in detect and correct errors and inconsistencies to ensure the quality of the data. For this task, OpenRefine is going to be used. The approach followed is to use different facets for the different columns to check in which ones there are inconsistencies or errors, identify the errors type and deal with them.

The first thing that can be noticed is that the field names are attribute values, which is making things imprecise and harder to understand. Therefore, the first correction implemented was the modification of the initial column names by the corresponding attribute names in the metadata.

In the "Loan Purpose" column, by using a text facet, some misspelling errors can be noticed. To solve this, clustering was used in order to merge those values that were referring to the same thing but that were written differently.

Another very simple error can be noticed in the "Case Number" column, as this values start from two instead of one. This error was corrected by extracting one from all the values in the column.

In the "Job Status" column, by using a text facet, a domain error that consist in two values classified as "yes" can be noticed. The correction of this error was done by replacing the "yes" values by the most popular one, in this case being "Skilled".

The last errors in the data are outliers, which can be found in the "Age" and "Credit Amount" columns.

About the "Age" column, for the values that are greater than eighty it can be assumed that the last digit was an input error and therefore it was removed. For the ages less than nineteen, in the case of the negative values and fractions, it can be assumed that those are input errors, as before, which were solved by removing the "0." and the negative sign. However, the main issue comes with two outliers of value six and one, as for this ones the last approach can not be used. For that reason, the solution implemented was to change those two values by the most popular one, which is twenty seven.

In relation to the "Credit Amount" column, there are eight records that are above twenty thousand and by checking the rest of the attribute values of those records, it can be assumed that for seven of them the last four ceros are input errors and therefore those digits were removed. Nonetheless, there is still one value which is above one hundred million, in this case it is difficult to know if that value is the result of an input error, therefore the correction implemented was to change it by the mean which was calculated using Excel.

For the rest of the colunmns, no errors were detected.

2.2 Data Conversion

The first conversion implemented in the data was to remove the "Case Number" column, that column does not really gives any relevant information about the problem (wheter is a good decision or not to give the credit to a customer) and therefore, it would have made things harder to understand independently of the algorithm used to analyse the data.

Considering the different data mining methods and algorithms that are going to be used, two variations of the data provided are needed. These variations are data with only nominal attributes and data with nominal and numerical attributes.

The data with both types of attributes is the one obtained after performing the cleaning and therefore there is no conversion needed to obtain it.

About the conversion to obtain data with only nominal attributes, the approach followed to deal with the numerical attributes ("Credit Amount" and "Age") was to create several nominal groups that included different ranges of the values. For "Credit Amount" the groups made are: "<2000", "2000<=X<4000" and ">=4000". In the case of "Age" the groups made are: "<30", "30<=X<40" and ">=40". The grouping was made taking into account a minimum of 200 instances per group.

The final data conversion implemented was to convert the csv files to arff files.

3 Data Analytics

3.1 Classification

3.1.1 OneR

The first Classification algorithm used was OneR (One-attribute-Rule) and it was used in the data with nominal and numerical attributes. The basic idea of OneR is to find an attribute that makes the fewest predictions errors. In the data provided, the attribute selected by OneR was "Credit Amount". As "Credit Amount" has a big numerical range no relevant patterns can be found in the output produced by Weka.

For that reason, the attribute was removed and the next one selected by OneR was "Debt History". After removing this one no more clear patterns can be found. The rules obtained from OneR are:

- If the client "Debt History" is "credits/all paid" then is a bad decision to give money.
- If the client "Debt History" is "all paid" then is a bad decision to give money.
- If the client "Debt History" is "existing paid" then is a good decision to give money.
- If the client "Debt History" is "delayed previously" then is a good decision to give money.
- If the client "Debt History" is "critical/other existing credit" then is a good decision to give money.

In the rules generated can be observed that the ones which are up to date with their payments have bad decisions while those that have payments to do have good decisions, this is something logically strange which might mean that having a clean "Debt History" does not really help when it comes to receiving loans. This algorithm is implemented with the training set as test option in Weka. Apart from that, 716 of the 999 instances are classified correctly.

3.1.2 J48

The second algorithm used was J48 or C4.5 and it was used with the same data variation as for OneR. J48 is an improved version of ID3, it works in the same way and produces a decision tree in the same way ID3 does (which basic idea is to use the entropy of each attribute in relation to the target attribute, in this case "decision"), however it introduces some advantages like being able to deal with nominal and numerical data, handle missing attributes or perform tree pruning. Some of the rules obtained by the output produced by Weka are:

- If the client "Account Status" is "<0" and "Debt History" is "existing paid" and "Loan Purpose" is "new car" then is a bad decision to give money (42 classified correctly, 15 misclassified).
- If the client "Account Status" is "<0" and "Debt History" is "existing paid" and "Loan Purpose" is "furniture/equipment" then is a good decision to give money (45 classified correctly, 17 misclassified).
- If the client "Account Status" is "0<=X<=200" and "Credit Amount" is <= 9283 then is a good decision to give money (248 classified correctly, 88 misclassified).
- If the client "Account Status" is "0<=X<=200" and "Credit Amount" is > 9283 then is a bad decision to give money (21 classified correctly, 4 misclassified).
- If the client "Account Status" is ">=200" then is a good decision to give money (63 classified correctly, 14 misclassified).
- If the client "Account Status" is "no checking" then is a good decision to give money (394 classified correctly, 46 misclassified).

From the rules generated can be observed that the clients who hold a positive balance in their current account or that do not have a current account at all are more likely to receive a loan, while those clients who hold a negative balance in their current account have more difficulties to get one. This algorithm is implemented with the training data set as test option in Weka and with a confidence factor of 0.1.

3.2 Regression

3.2.1 Logistic Regression

The algorithm used for Regression is Logistic Regression. In comparison with Linear Regression, this algorithm can handle nominal and numerical values, therefore, the data used for the classification algorithms can be used for this one as well. Two class classification problems can also be seen as regression problems, in this case "Decision" has two values which means that we can use Logistic Regression to find patterns in

the data. The basic idea of the two class problem is to find a description of the probability function which in this case it is described by the logistic function. Some of the patterns found were the following:

- If the client "Loan Purpose" is "used car" then there is a high probability of having a good decision to give money (Odd ratio of 2.5968).
- If the client "Loan Purpose" is "retraining" then there is a very high probability of having a good decision to give money (Odd ratio of 3.3964).
- If the client "Loan Purpose" is "new car" then there is a low probability of having a good decision to give money (Odd ratio of 0.5909).
- If the client "Loan Purpose" is "education" then there is a low probability of having a good decision to give money (Odd ratio of 0.3849).
- The client "Credit Amount" does not seem to affect the probability of having a good decision to give money (Odd ratio of 0.9999).
- If the client "Savings" is "<100" then there is a low probability of having a good decision to give money (Odd ratio of 0.6944).
- If the client "Savings" is "100<=X<500" then there is a low probability of having a good decision to give money (Odd ratio of 0.8239).
- If the client "Savings" is ">=1000" then there is a high probability of having a good decision to give money (Odd ratio of 2.1063).

From the patterns above can be observed that the "Loan Purpose" choice affects the decision of whether to give money or not as some of the attribute values provide the client with a higher chance of having a good decision than others. On the other hand "Credit Amount" does not seem to really matter when it comes to the decision of giving money or not. And finally, can also be seen that the higher the "Savings" the higher the probability of the client to get the loan. This algorithm was implemented with the training data set as test option in Weka.

3.3 Association

3.3.1 Apriori

The data mining algorithm used for Association was Apriori and it was implemented with the data with only nominal attributes as it can not handle numerical values. This algorithm works in the following way: first, a minimum item set support is set, after that, attribute associations at different levels are found until there are no more associations with a support higher than the minimum or a maximum number of attributes is reached and finally, rules that satisfy a minimum confidence, previously set, are generated from the associations. Some of the best associations rules found by Weka are:

- If the client "Personal Status" is "male single" and "Age" is "30<=X<40" and "Job Status" is "skilled" in the antecedent then the consequent is good decision(Confidence of 0.78).
- If the client "Years Employed" is ">=7" and "Job Status" is "skilled" in the antecedent then the consequent is good decision (Confidence of 0.78).
- If the client "Age" is ">=40" and "Job Status" is "skilled" in the antecedent then the consequent is good decision (Confidence of 0.75).
- If the client "Years Employed" is ">=7" and "Personal Status" is "male single" in the antecedent then the consequent is good decision (Confidence of 0.75).
- If the client "Age" is "30<=X<40" and "Job Status" is "skilled" in the antecedent then the consequent is good decision (Confidence of 0.74).

- If the client "Years Employed" is ">=7" and "Age" is ">=40" in the antecedent then the consequent is good decision (Confidence of 0.73).
- If the client "Personal Status" is "male single" and "Age" is ">=40" in the antecedent then the consequent is good decision (Confidence of 0.72).

From the associations rules can be observed that almost half of the rules indicate an amount of "Years Employed" greater than seven and almost all rules indicate an amount of "Age" greater than 30, these two patterns might mean that the clients who have a stable job or are older than 30 are more likely to get a loan. In addition, the "Personal Status" "male single" and the "Job Status" "skilled" appear multiple times in the antecedent of the rules, therefore, can be observed that being "male single" or being "skilled" has a positive influence when it comes to receiving loans. This algorithm is implemented with "car" set to true, "classIndex" set to 10, a minimum support of 0.1 and a minimum confidence of 0.7. Apart from that, 61 rules are generated and 18 cycles are performed.

4 Conclusion

Summarizing the findings stated in the last section can be said that having a clean "Debt History" does not really help to get loans, clients holding a positive balance in the "Account Status" are more likely to get loans, the "Loan Purpose" choice affects the decision of getting a loan, "Credit Amount" does not seem to matter when it comes to receiving money, the higher the "Savings" the higher the chance of getting a loan and finally, having an stable job, being older than 30, being "male single" or being "skilled" have a positive influence when it comes to receiving money.

About the algorithms used, in the case of OneR, the rules were clear to understand however they were too general in comparison with the rules obtained using the other algorithms. About Apriori, the main issue with this approach is that it generates lots of rules and some of them are very similar which makes the selection harder as well as the analysis of the patterns. On the other hand, J48 generates an appropriate amount of rules with an appropriate amount of detail which makes things easier to analyze and draw conclusions about the patterns, and in the case of Logistic Regression, by using the probability of the different attribute values it is also easier to find explanations for the patterns. For those reasons the most effective algorithms used were J48 and Logistic Regression.

References

[1] Ian H. Witten, Elbe Frank, Mark A. Hall. *Data Mining Practical Machine Learning Tools and Techniques Third Edition*. Chapter 11.2 Pages 429-430, Chapter 11.4 Pages 467-468, Chapter 11.7 Pages 485-486.

[2] OpenRefine Wiki: https://github.com/OpenRefine/OpenRefine/wiki

[3] Deal with outliers: https://conversionxl.com/blog/outliers/

[4] Logistic Regression (Odds and probability): https://www.theanalysisfactor.com/understanding-odds-and-probability/

[5] Apriori: https://machinelearningmastery.com/market-basket-analysis-with-association-rule-learning/