Data Analytics (SET09120)

Coursework II

Valentin Kisimov

Edinburgh Napier University
40439132@live.napier.ac.uk

Abstract. The aim of this report is to showcase patterns and relationships in a database which contains historic observations for 1000 applications for credits. This is achieved by preparing and cleaning the data, analyzing the result and summary of the findings.

Keywords: database, analytics, credit, score.

1 Introduction

This report would explain and show the methods used to analyze a raw database. In order to find patterns and relationships in a data which possibly have errors, we should start with preprocessing the whole database, which includes converting the data to useful format, converting and changing values, removing spelling errors, fixing outliers and improving the overall data consistency. Tools used: **OpenRefine**, **Weka**;

2 Data Processing

2.1 Data Cleaning

This step aims to correct inconsistencies, typos and improve readability and usability of the data. This pre-processing step is crucial for working efficiently with the data.

Firstly, the approach used when encountering missing data was replacing it with keyword “missing” to indicate that there is a problem with the current row but not completely delete the record, which ensures that the rest of the information of the case is available. Changing the value to a mean value one can result in unexpected results, bad decision based on wrong data etc.

After testing with several algorithms, I have decided that the more efficient and simple way is simply to delete the fields which include values that are not easily predictable.

"Id" column is not deleted originally, it can be considered as a column that is not providing real value to the database but having an additional id can be helpful in some cases. When dataset is used in Weka "id" column is removed, as this improves the accuracy of the model.

Steps taken in pre-processing:

1. Adding a header as it was not present in the dataset provided. Renaming a few columns for better readability. *See Table 1*.
2. Correcting errors, outliers and cleaning the data. Deleting rows which cannot be used properly for the training. *See Table 2*.
3. Converting the data to nominal values and simplifying it for better training and accuracy. *See Data conversion and simplification.*

Table 1. Attribute name changes:

Coursework Specification attribute name	Row 1 values	Changed attribute name
Case_no	1.0	id
checking_status	'<0'	account_status
credit_history	'critical/other existing credit'	debt_status
purpose	radio/tv	credit_reason
credit_amount	1169.0	credit_amount
saving_status	'no known savings'	savings_amount
employment	'>=7'	employment
personal_status	male single	gender:family
age	67.0	age
job	skilled	job_status
class	good	loan_status

Table 2. Data correction:

All columns	Removed single quotes
credit_reason	fixed typos and punctuation consistency
gender:family	fixed typo
age	Floating point remove and fixed outliers caused by typos
job_status	3 instances of wrong value converted to 'missing'
loan_status	3 instances of '1' converted to missing1; 3 of '2' to missing2

2.2 Data conversion and simplification

Column "**debth_status"** attributes "all paid", " no credits/all paid" and "existing_paid" combined with "all paid".

Column **"credit_reason"** attributes "used car" and "new car" to car;
attributes "domenestic appliance" and "furniture/equipment" to "furniture";
attribute "retraining" to "other";

Column "**credit_amount**" modified with python code to nominal values using the same logic as employment – "1K<=X<2K" meaning the credit amount is from 1000 (incl.) to 2000 (excl.). Not using raw numbers, because they can be hard to read when visualizing a tree graph. 1K step until 5K, then 5<=X<10K and X>10K.

Column "**age**" modified with python code to nominal values using the same logic as employment – Split to below 25, 25 to 30 (example in database 25<=X<30), 30 to 40, 50 to 50, above 50.
Column "**job_status**" attributes "high qualif/self emp/mgmt." to highly skilled;
"unemp/unskilled non res" and "unskilled resident" to "unskilled".

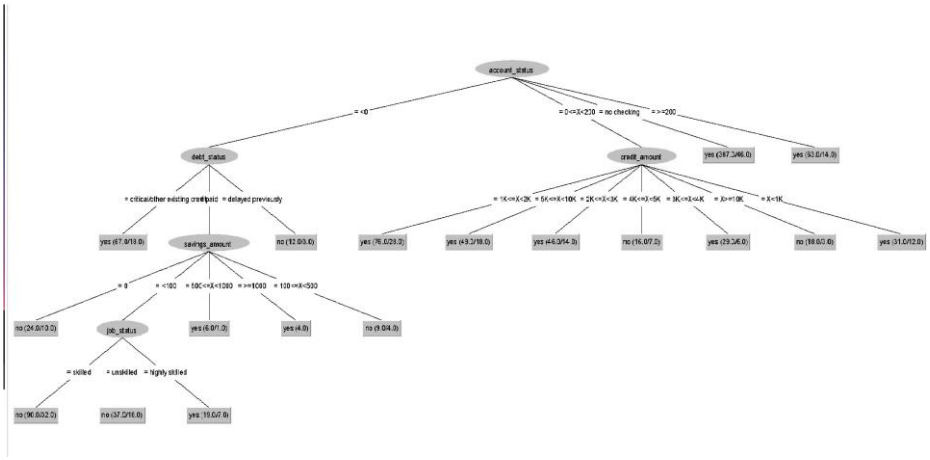
After all the preprocessing is completed the dataset is saved in csv format and converted using Weka to .arff format.

3 Data Analytics

3.1 Classification

J48 algorithm is used as it provides high accuracy, tree visualization and a lot of parameters to choose and tweak. This can result in highly accurate model with tree structure which is easy to read, understand and extract rules from.

After much experimenting the parameters of J48 algorithm and highly pre-processed dataset I was able to achieve an accuracy of 99.49%. This was achieved without pruning and minimum instances per leaf = 1. This results in highly complex tree with 755 leaves. Unfortunately, this tree cannot be properly visualized and analyzed due to its size and impossible readability.



The complexity of the tree can be easily controlled by two parameters “confidenceFactor” and “minNumObj”. For simpler, easier to read tree which can be analyzed and visualized the parameters (settings) changed are:

minNumObj	15	Higher number for less nodes with more instances
confidenceFactor	0.5	Lower numbers introduce more pruning and low accuracy

This results in much simpler, less accurate tree, but it is properly displayed. I have tried to find the balance between complexity and readability, but I could not get a perfect result.

Rule 1:

If “account_status = no checking” then loan_status = “yes” (387/46)

If the client’s account status is “no checking” then the loan would be most likely given. 387 instances apply to this rule, where 46 are incorrectly classified.

Rule 2:

If “account status = 0<=X<200” and credit amount “X>10K” then “no” (18/3)

If the client’s account status is between 0 and 200 and he wants a credit above 10K then he would be rejected. 3 of 18 cases has been wrongly classified by the model.

Rule 3:

If “account status = 1K<X<2K” and credit amount is “1K<=X<2K” then “yes” (76/28)

If the client’s account status is between 1000 and 2000 and he wants a credit between 1000 and 2000 then its highly likely he would be given this credit.

Rule 4:

If “account status = <0” and “debth_status = delayed previously” then “no” (12/3)

If the client delayed his credit payment, he would not be given a credit.

Rule 5:

If “account status = <0” and “debth_status = critical/other” then yes (67/18)
If the clients debth status is critical or other he would be given a credit

Rule 6,7,8:

If “account status = <0” and “debth status = paid” then:
If “savings amount = > 1000” then “yes” (4/0)
If the client has savings above 1000 then he would be given a credit.
If “savings amount = 500<=X<1000” then yes (6/1)
The client has savings between 500 and 1000 he would be given a credit.
If “saving amount < 100” and “job status = highly skilled” then yes (19/7)
The client has savings below 100 but is highly skilled he would be given a credit.

3.2 Association

Algorithm Used - Apriori
Settings used – Rules – 6

Rule 1:

account_status=no checking credit_reason=radio/tv 124 ==> loan_status=yes 117 <conf:(0.94)> lift:(1.34) lev:(0.03) [29] conv:(4.62)

If the client does not have a bank account with the current bank, his credit reason is radio/tv and his loan status is yes he have a 94% chance to get a loan.

Rule 2:

account_status=no checking debt_status=critical/other existing credit 152 ==> loan_status=yes 142 <conf:(0.93)> lift:(1.33) lev:(0.04) [35] conv:(4.12)

If the client has existing credit with other bank but not with the current he is 93% and his loan status is yes, he is 93% likely to get a credit

Rule 3:

account_status=no checking employment=>=7 113 ==> loan_status=yes 105 <conf:(0.93)> lift:(1.32) lev:(0.03) [25] conv:(3.74)

If the client has no current bank account with the current bank, he is employed for more than 7 years and his loan status is yes then his chance for a loan is 93%.

Rule 4:

account_status=no checking gender:family=male single job_status=skilled 149 ==> loan_status=yes 137 <conf:(0.92)> lift:(1.31) lev:(0.03) [32] conv:(3.42)

if the client has no current bank account with the current bank and he is a single male with skilled job status and his loan_status is yes then he is 92% likely to get a loan

Rule 5:
account_status=no checking age=30<=X<40 145 ==> loan_status=yes 132 <conf:(0.91)> lift:(1.3) lev:(0.03) [30] conv:(3.09)

If the client does not have a bank account with the current bank, he is between 30 and 40 years old and his loan status is yes, then he is 91% likely to get a loan.

Rule 6:
account_status=no checking credit_amount=1K<=X<2K 126 ==> loan_status=yes 114 <conf:(0.9)> lift:(1.29) lev:(0.03) [25] conv:(2.89)

If the client has no current bank account with the current bank, his credit amount is between 1000 and 2000 and his loan status is yes then he is 90% likely to get a loan.

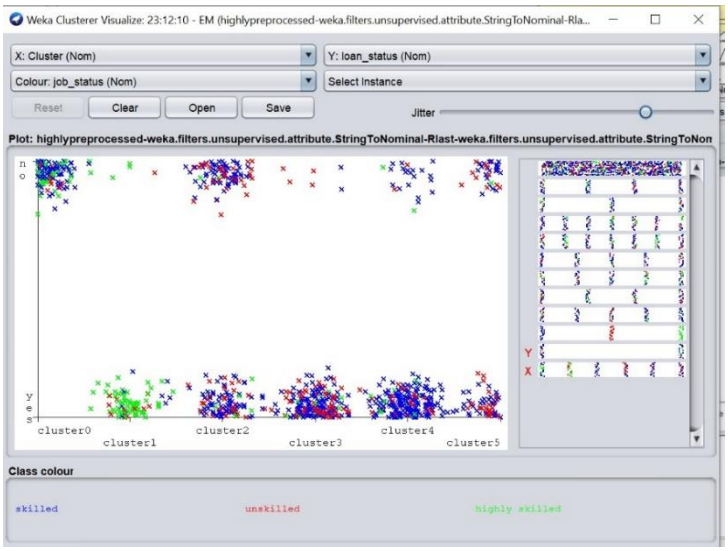
3.3 Clustering

Algorithm Used - simplekmeans
Settings Used – Default with Clusters set to 6

account_status	<0	no checking	<0	no checking	no checking	0<=X<200
debt_status	Paid	critical/other existing credit	paid	critical/other existing credit	paid	paid
credit_reason	Car	car	furniture	radio/tv	car	radio/tv
credit_amount	5K<=X<10K	5K<=X<10K	1K<=X<2K	1K<=X<2K	1K<=X<2K	X<1K
savings_amount	<100	<100	<100	<100	<100	<100
employment	>=7	>=7	1<=X<4	>=7	1<=X<4	1<=X<4
gender:family	male single	male single	female div/sep/mar	male single	male single	female div/sep/mar
age	30<=X<40	30<=X<40	25<=X<30	X>=50	30<=X<40	X<25
job_status	skilled	highly skilled	skilled	skilled	skilled	unskilled
loan_status	no	yes	yes	yes	yes	yes

Clustered Instances:

0 129 (13%)
1 87 (9%)
2 212 (22%)
3 206 (21%)
4 255 (26%)
5 94 (10%)



4 Summary

After a lot of experiments, it seems that the higher the quality the data is more accurate the machine learning algorithms work. Heavy pre-processing of the data is crutial if an accurate prediction is wanted.

For simpler relationships and explanations, it is a good practice to use simpler settings for the algorithms, especially for my personal favorite Classification J48. It can be used to accurately predict an outcome given the values. It's usefulness, flexibility and visualizations make it superior to the other methods. It seems that credit_amount and acount_status are the main influencers of the final credit decision.

Association Apiori algorithm is highly efficient in producing general rules based on the data and it have pretty wide applicable way of function, meaning that it can be used in different datasets efficiently producing accurate rules without too much changing of its settings.

Clustering is maybe more useful for different kind of dataset, but still it produced semi accurate results, which can be used for summarizing the data and quickly identifying trends and relationships.