SET09120 – Data Mining Coursework

Introduction

The aim of the data mining coursework is to present an adequate understanding of the utilisation of Weka and OpenRefine to appropriately explore a dataset through a variety of data mining techniques, which I will then compile a report on my findings. I used the software OpenRefine to prepare and clean the 'dirty' dataset provided and then use Weka to analyse, model and predict outcomes through applying algorithms on my dataset to produce visualisations and decision trees.

Data Preparation

Data Preparation consists of a few processes: Reformatting, Correction and Enrichment. Cleaning & Reformatting enables top-quality data to be used in your analysis, while enrichment allows you to combine sets of data within the dataset to allow for deeper insight.

Data Cleaning

Attribute	Pre-Clean	Post-Clean
purpose	ather	other
	busines	business
	busness	business
	Eduction	education
	Radio/Tv	radio/tv
	'used car'	used car
	'new car'	new car
	'domestic appliance'	domestic appliance
credit_amount	111328000	1328
	13580000	1358
	13860000	1386
	19280000	1928
	5180000	518
	5850000	585
	63610000	6361
	7190000	719
age	-29	29
	-34	34
	-35	35
	0.24	24
	0.35	35
	0.44	44
	1	19
	6	60
	222	22
	333	33
job	yes	skilled

Firstly, I started by adding column headers/attribute labels, making it easier for people to read. Then, I removed the case_number column, as that is the same as an ID field and provides no actual relevant data or provides relational information from the dataset.

The dataset provided was riddled with errors and was a rather 'dirty'. These had to be 100% rectified before I could begin the data analysis and mining process. For this, I had to use some toolsets that came preinstalled with OpenRefine called Facets and Transformations (see table above). There were also some blank cells, these were "filled down" with random data from the set available.

Data Conversion

With the dataset containing mostly nominal values, transforming it to fully nominal wasn't necessarily difficult. All I had to do was replace the continuous data (age (Fig 1) & credit_amount (Fig 2)) with sensible ranges of X, similar to how it is elsewhere in the dataset (similar to employment or checking_status).

I then went back and created a fully numeric dataset which took roughly 20 minutes of my time. This wasn't necessarily difficult either as I took the data in the specification, which was then transformed into a numerically encoded dataset.

Attribute	Original Value	New Value
case_no	a number	[removed]
checking_status	< 0	0
	0<=X<200	1
	X>200	2
	no checking	3
credit_history	no credits/all paid	0
	all paid	1
	existing paid	2
	delayed previously	3
	critical/other existing credit	4
purpose	new car	0
	used car	1
	furniture/equipment	2
	radio/tv	3
	domestic appliance	4
	repairs	5
	education	6
	retraining	7
	business	8
	other	9
	vacation	10
credit_amount	a number	
saving_status	<100	0
	100<=X<500	1
	500<=X<1000	2
	>=1000	3
	no known savings	4
employment	unemployed	0
	< 1	1
	1<=X<4	2
	4<=X<7	3
	>=7	4
personal_status	male div/sep	0
	female div/dep/mar	1
	male single	2

	male mar/wid	3
age	a number	
job	unemp/unskilled non res	0
	unskilled res	1
	skilled	2
	high qualif/self emp/mgmt	3
class	good	0
	bad	1

Data Analytics

Classification

Classification is a problem where we have a Machine Learning algorithm that attempts to correctly identify and classify an object based on a number of parameters. This is mostly achieved through a process of yes/no questions called Binary Classification[1]. Each of these observations made are assigned a value of 1 and -1 through a sign function – Often using 1 and 0 for ease of use and understanding –.

Classification tasks primarily require data. We can use a 50/50 split or use the most common convention of 1/3 to 2/3 split.

For the classification task of this assignment, I decided to use J48 as it creates a Decision Tree (Fig 3) from my nominal dataset, which is pruned, meaning it removes specific sections of the tree which are effectively useless making the tree easier to read for the data analyst.

Handily, by using J48 I can get a confusion matrix created (Fig 4), which gives me everything I need to know how correct my classifier is. In this case, the classification algorithm I used was 79.6% correct, which means false classifications sit at 20.4%, which isn't perfect, but it's certainly good!

Rule 1

IF checking_status = "<0" AND credit_history = "critical/other existing credit" THEN "good" (67/18) – This rule implies if the customer checking status is less than 0, but their history implies they have other debt at another bank, then they will be granted the loan. This rule has (67/18) accuracy, meaning every 67 items in 85 will be classified correctly (78.9%).

Rule 2

IF checking_status = "no checking" THEN "good" (394/46) – This rule implies if the customer has no existing current account, then they will be granted the loan. This rule has (394/46) accuracy, meaning every 394 items in 440 will be classified correctly (89.5%).

Rule 3

IF checking_status = "<0" AND credit_history = "existing paid" AND purpose = "radio/tv" AND age = "20<=X<30" AND credit_amount = "0<=X<2000" THEN "bad" (11/4) – This rule implies that if the customer checking status is less than 0, they have paid off all their other debts, they are purchasing a radio/tv, their age is between 20 [inclusive] and 30 and they have between 0 [inclusive] and 2000 in the bank, then their loan application will be declined. This rule has an accuracy of (11/4) meaning for every 11 in 15 items classified will be classified correctly (73.3%).

Rule 4

IF checking_status = "<0" AND credit_history = "existing paid" AND purpose = "radio/tv" AND age = "30<=X<40" THEN "bad" (12/2) – This rule implies that if the customer checking status is less than 0, their other debts have been paid off, they are purchasing a radio/tv and their age is between 30 [inclusive] to 40 then their loan application will be

declined. This rule has an accuracy of (12/2) meaning for every 12 in 14 applications classified will be classified correctly (85.7%).

Rule 5

IF checking_status = "0<=X<200" AND credit_amount = 0<=X<2000 AND purpose = "radio/tv" THEN "good" (40/9) – This rule implies that if their checking status is between 0 [inclusive] and 200, their credit amount is between 0 [inclusive] and 2000 and the purpose is for a radio/tv then their loan application will be accepted. For every 49 items being classified, 40 will be classified correctly each time. This has an accuracy of 81.6%.

Rule 6

IF checking_status = "<0" AND credit_history = "all paid" THEN "bad" (22/6) – This rule implies that if their checking status is less than 0, their credit and their previous credit has been paid back duly, their loan application will get rejected. For every 28 items being classified, 22 will be classified correctly each time. This gives an accuracy of 78.6%.

Association

For these rules, I allowed the Apriori algorithm to handle this process using my nominal dataset I created in the section above. I used the default parameters for this algorithm with the exception of the rules generated, I changed that to 6.

The way association works, is by "discovering the probability of the co-occurrence of items in a collection. The relationships between co-occurring items are expressed as association rules"[3].

Rule 1

checking_status=no checking purpose=radio/tv 127 ==> class=good 120 <conf:(0.94)> – This rule means if they have no current account with the bank and they are purchasing a radio/tv, they will likely obtain this loan. This is true for 94% of the objects classified under this rule as the confidence of this rule is 0.94.

Rule 2

checking_status=no checking credit_history=critical/other existing credit 153 ==> class=good 143 <conf:(0.93)> – This rule means if they have no current account at the bank and their credit history is critical (they have other outstanding debt at another bank) then they will also likely have this loan granted. This rule is true for 93% of objects identified under this rule as the confidence value is 93%.

Rule 3

checking_status=no checking employment=>=7 115 ==> class=good 107 <conf:(0.93)> - This rule implies that if they have no current account at the bank and they have been employed for 7 years or more, then they are likely to be granted this loan. This is true for 93% of all objects identified under this rule as it has a confidence value of 0.93.

Rule 4

checking_status=no checking personal_status=male single job=skilled 150 ==> class=good 139 <conf:(0.93)> – This rule implies that if you have no current checking account at the bank, they are also a single male while being in a skilled/official job, they will be likely to receive the loan. This rule has a 93% accuracy as the confidence value for it is 0.93.

Rule 5

checking_status=no checking credit_amount=0<=X<2000 job=skilled 115 ==> class=good 106 <conf:(0.92)> – This rule means that if you have no current account at the bank, your credit amount is between 0 [inclusive] and 2000 and you're in a skilled job, then you would be granted the loan. This is true for 92% of all identified data objects as the confidence value is 0.92.

Rule 6

checking_status=no checking credit_amount=2000<=X<4000 131 ==> class=good 120 <conf:(0.92)> – This rule imples that if you have no current account at the bank and your credit amount is between 2000 [inclusive] and 4000, then you will likely receive the grant for the loan. This is true for 92% of all data objects identified like this as the confidence value for this rule is 0.92.

Clustering

Clustering is the act of grouping a set of related data together to visualise/make patterns which are easier to recognise through the creation of a table structure. Usually, we use clustering to determine the number of outputs we require (which in this case would be two) but seeing I require roughly 6 rules for this section, I will generate 6 outputs from the clustering algorithm with the input being my nominal dataset.

I used the algorithm SimpleKMeans as it's a really nice algorithm and quite popular in industry. "It makes use of vector quantisation, that aims to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest mean (cluster centres or cluster centroid)" [2]. This method of clustering will also utilise the Euclidean Distance as its distance function.

	Cluster 1 (21%)	Cluster 2 (13%)	Cluster 3 (17%)	Cluster 4 (18%)	Cluster 5 (13%)	Cluster 6 (19%)
checking_status	no checking	<0	no checking	no checking	<0	0<=X<200
credit_history	critical/other existing credit	critical/other existing credit	existing paid	existing paid	existing paid	existing paid
purpose	new car	used car	radio/tv	radio/tv	new car	radio/tv
credit_amount	0<=X<2000	2000<=X<4000	0<=X<2000	0<=X<2000	2000<=X<4000	0<=X<2000
saving_status	<100	<100	no known savings	<100	<100	<100
employment	1<=X<4	unemployed	>=7	>=7	>=7	1<=X<4
personal_status	female div/dep/mar	male single	male single	male single	male single	female div/dep/mar
age	20<=X<30	30<=X<40	40<=X<50	30<=X<40	30<=X<40	20<=X<30
job	skilled	high qualif/self emp/mgmt	unskilled resident	skilled	skilled	skilled
class	bad	good	good	good	bad	good

Referring to Figure 3 of the appendix, you can see from the table I have created above from the clustering task, you can see that if your account has a "no checking" checking status, your credit history exists and is fully paid, you're purchasing a radio/tv, your credit amount sits between 0 and 2000, you've been employed for 7 or more years, you're a single male and your age is roughly between 30 & 40, your application for a loan will be approved. Conversely, if you've got existing credit which isn't paid, you're buying a new car, your age is roughly between 20 and 40 and you're a skilled resident, then your loan application will be declined. It appears that the clustering algorithm doesn't particularly take into account the saving status and checking status as a singular deciding factor, as the values for these attributes are fairly similar across all clusters.

Another thing to note, is that the personal status of the applicant is questionable, as 75% of males who are single and apply for the loan are granted this loan, but then 50% of the loan applications that come from females who are either divorced, separated or married are granted. This doesn't particularly show anything like a pattern from the dataset, as we'd require more data returning the "bad" class to determine anything meaningful from it.

Conclusion

Generally, I have found that the most effective method for finding and generating the more realistic rules between properties is without doubt association. As mentioned above, it's the act of finding patterns in data probabilistically with co-occurrence in mind.

Alternatively, I could have used classification for this task, but it didn't provide as deep an insight to the dataset as I initially thought as some of the rules produced were questionable – to say the least.

Although either method may be used for a prediction task and be used for supervised learning (where we know what we want the output of the data to be) we can use association for an unsupervised task. This gives the edge of being able to be used in real-time to aid where there are mass flows of data and our output from the machine is potentially unknown.

Appendix

Figure 1. age displayed as nominal

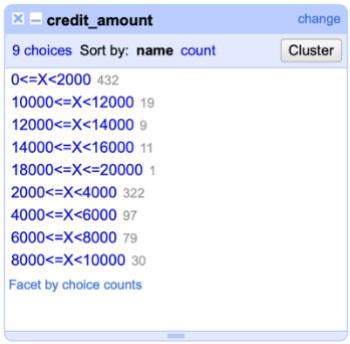

Figure 2. credit_amount displayed as nominal

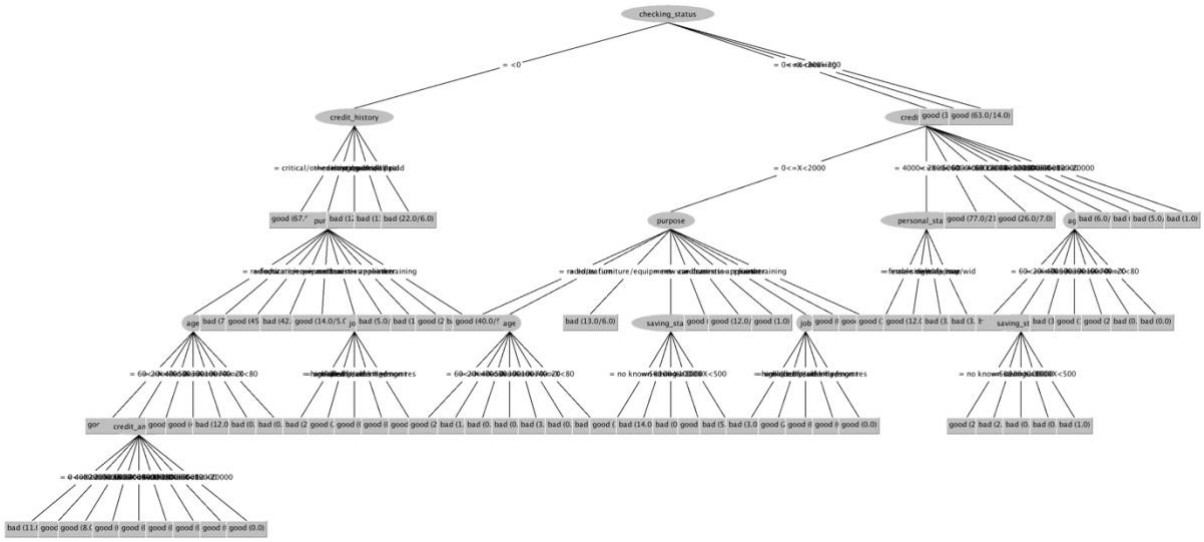

Figure 3. The Decision Tree created by the J48 Algorithm.

```
=== Confusion Matrix ===

   a   b   <-- classified as
 649  51 |   a = good
 153 147 |   b = bad
```

Figure 4. The Confusion Matrix generated from the J48 Decision Tree

References

1. Classification – Data Science for Everyone! https://matthew-brett.github.io/dsfe/chapters/09/classification
2. Clustering – k-means clustering https://en.wikipedia.org/wiki/K-means_clustering
3. Association - https://docs.oracle.com/cd/B28359_01/datamine.111/b28129/market_basket.htm#BABFDDCG