Data Mining Assignment

1.0 Introduction

The assignment brief that ‘Banks are currently having trouble with debt, also states that the banks would like to avoid lending money to people in the future, who are unable to to repay these loans. Using the data provided from 2000 previous loan customers, appropriate data mining techniques allow differentiation, and prediction of how likely a customer is to repay their loans. Predictions on the likelihood of a customer repaying future loans is determined by the given attributes, and outcomes recorded in the dataset for each individual.

The raw data contained 15 attributes of 2000 individuals (instances), a unique six-digit ID identified the individual customers.

Attributes:

Customer ID I Forename I Surname I Age I Gender I Years at Address I
Employment Status I Country I Current Debt I Postcode I Income I
Own Home I CCJs I Loan Amount I Outcome I

Terminology: Task / Scenario / Project all refer to the situation of which this report and data mining project has been presented. The terminology refers to the scenario of the financial establishment, and the end goal of predicting the probable chance of future loans being repaid by customers.

Financial Establishment refers to the bank in which the data saws supplied from.

Record(s) refer to the individual data variables for each row within the provided dataset.

Red text within tables (Data Summary section) indicated flagged issues, which may affect the data, or should be attended to.

Instances / Customers / Individuals / Person / Applicant all refer to an individual set of values for one unique person. One person has 15 attributes, and can be identified by their Customer ID.

Record(s) refer to the results of a particular customer or customers.

NB found in chapter 4 (table data) is shorthand for Naive Bayes.

In the Modelling section: **D** = Defaulted I **P** = Paid

Mean refers to the average of a set of numbers.

Standard Deviation refers to the quantity expressing by how much the members of a group differ from the mean value for the group.

Noise: a value that differs from the norm; is error or irrelevant data, it is not part of the data and should be removed.

Outliers: a value that significantly differs from the norm; it is not usually expected however it is part of the data, may be of importance.

2.0 Data Summary

2.1 Customer ID

Numeric, Unique

Inspection of the dataset found that the Customer ID attribute, a unique numeric value used by the financier establishment to independently identify a customer, had a number of duplicate values. The dataset had a 99% unique rate (1986 columns) for the Customer ID attribute, leaving a remainder of 14 values that were either incorrectly entered in to the system, or should be merged as the individual customer may have requested multiple loans.

Customer ID (Attribute)	Missing	Distinct	Unique
Value	0	1993	1986
Percentage	0%	99%	99%

It can be deduced from the unique rate for the Customer ID that 7 customers use the same Customer ID twice each, this may be in error or due to multiple loan requests.

However this is not the case, as filtering duplicate Customer ID's from the dataset, displayed different customers referencing the same unique Customer ID attribute.

Customer ID (Attribute)	Forename	Surname	Age
684733	Paul	Brown	44
684733	Mark	Bickertor	27

It is important to ensure each customer can be uniquely identified, as the merging of many customer accounts may effect data, especially if incorrect Customer ID entry is a regular occurrence. This also limits how effective other attributes are to finding a solution to the proposed loan repayment issue, as their reliability to the correct customer, and therefore other relevant attributes, is uncertain.

2.2 Forename/Surname

Nominal

The forename and surname of customers stored in the financial establishments database, and therefore this dataset, is not an attribute that can be used to provide any statistical information of value, and has very little relevance to providing probability results for customers repaying future loans.

This does not exempt the attributes from containing multiple variations of data, and some invalid entry types. Due to the nature and irrelevance of naming and gender, 'R Bundy' can be considered a valid entry, however it makes for 'messy' data.

Punctuation can be found within some Surname attributes, such as De,ath and O,sullivan (ideally O'Sullivan). It would not be a serious loss to remove these entries, due to the size of the dataset. The more worrying element is the mismatched gender attribute, as Peter Fewson has been entered as female. It is important to reiterate the lack of importance the forename and surname attributes contribute, to the overall outcome of predicting future loan repayments, as naming and gender provide no real usable statistics.

Customer ID (Attribute)	Forename	Surname	Gender
730480	Andrew	D Evil	F
1040020	David	@	M
993790	Ian	O,sullivan	F
979431	Michael	De,ath	M
986670	R	Bundy	M
1037772	Peter	Fewson	F

2.3 Age

Numeric

Age appears to be an attribute that has been entered correctly for every customer within the dataset, age values range from (Ascending) 17 to 89. There are a total of 73 distinct values. Statistical data provides a greater understanding of the data, displaying a Mean of 52.912 (average age 53 rounded). The standard deviation measuring the spread of data surrounding the Mean value, is 20.991, presenting a good spread of age values, rather than a more unreliable clustered spread.

Probable distributions can be applied to specific subsets of age values i.e. 17 to 20 (youth), or other formed subsets deemed fit to provide the most reliable data, i.e. deviations of 5 or 10 year age differences.

Age subsets could provide statistical data on the reliability of these specific age groups, without any external factors effecting the customer situation. Combining this data with external factors i.e. debt amount, valuable resulting data could be obtained to present the probability of these customers repaying future loans.

2.4 Gender

Nominal

Based on the majority of values for the Gender attribute in the provided dataset, the guidelines for gender

Post-analysis, the gender attribute displays a total of nine discrete variations which represent male and female. The majority of values are presented as ‘M’ and ‘F’ for their respective genders. The significance of the attribute values ‘H’, ‘D’, ’N' are unknown, however using the forename attribute, the most appropriate gender will be associated, and the gender attribute changed accordingly. These values have therefore been associated as noise, as the data has been entered incorrectly.

The remaining two attribute variations ‘Male’ and ‘Female’ could simply be altered to represent ‘M’ and ‘F’, however under further inspection, particularly concerning ‘Female’, forenames consisted of David, Simon, Stuart and Dan. Assuming data has been entered incorrectly, alterations to the gender for these four customers has been altered. Gender data would be entered in the form of a checkbox / radio button (easily misconfigured, and the forename attribute would be entered manually, it is more probable that the gender attribute is incorrect. Therefore the four values labelled ‘Female’ have been altered to ‘M’ to represent male, gender data now has two distinct values of ‘F’ and ‘M’, as other variations have been eliminated.

Gender (Attribute)	F	M
Count	969	1031
Percentage	48.45%	51.55%

Gender (Attribute)	F	M	Male	Female	H	D	N	1.0	0.0
Count	968	1017	3	4	2	1	2	2	1

The percentage of M and F values representing the gender attribute is extremely similar for the large dataset (3.1% differentiation). Considering the low percentage bias, the gender attribute could be used to accurately provide statistics, on the probability of the repayments of future loans.

However gender discrimination may be apparent, therefore rendering the gender of customers irrelevant. As previously mentioned, all data for the gender attribute may be kept, even if a presumed gender attribute for a customer cannot be entirely accurate. The role gender plays in the probability of customers repaying future loans can be determined (near-even split of data), however its usefulness is low or may even be disregarded entirely.

Gender is unreliable based on the Forename attribute shown in section 2.2, as Ian and Andrew are both considered to be female, forename which are generally presumed to be a male. As described above, the usefulness of the gender attribute is minimal, therefore it is not worthwhile the clean the data.

2.5 Years at Address

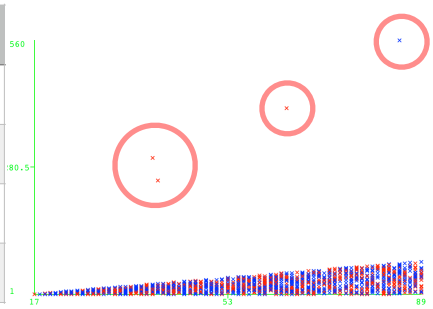
Numeric

The majority of values for the Years at Address attribute have been entered correctly, however there are four anomalies, indicating that customers have been living at the address stated for an unrealistic period of time.

Given the attribute values shown, it is not conclusive as to whether these four values were entered incorrectly (additional figure value), or that the figure is intended to represent number of months, rather than years.

The four values shown below have been interpreted as noise, as these values are considerably outside the normal standard range for the ‘years at address’ attribute, presumed to be invalid data.

Customer ID	Forename	Surname	Years at Address
1075394	Simon	Wallace	560
967482	Brian	Humphreys	410
678376	Steve	Hughes	300
722046	Chris	Greenbank	250



The four customers listed could be removed from the dataset, as the remaining 1996 customers would provide reliable data (the loss of four customers data is minimal). Alternatively the average Years at Address value could be taken from the 1996 values (excluding the noisy values), to replace the four values.

However, Years at Address could be regarded as an attribute that does not affect the end outcome for the probability of a customer to repay their loan, and therefore could remain within the dataset. It is unwise to assume what the provided figures are intended to be, and should not be changed.

The dataset in its original state has 74 distinct values ranging from minimum 1 year to maximum 560 years, presuming the above scenario of incorrectly input data, the maximum years would be 71. The following results are for the unedited dataset that was provided.

Years at Address	Unedited Dateset	Edited Dateset
Minimum Value	1	1
Maximum Value	560	71
Number of Values	1996	2000
Distinct Values	74	70
Mean	18.526	17.801
Standard Deviation	23.202	15.764

Subsets of 10 year intervals for this attribute could assist in the solution to the issue, of providing a reliable probability rate of repaying future loans, dependant on years lived at the current address.

2.6 Employment Status

Nominal

The discrete values for the Employment Status attribute in the provided dataset include 'Self Employed' which is the majority, it is unlikely that 1013 (50.65%) people of 2000 are self-employed, however there is no evidence to disprove this figure.

The remaining 3 of the 4 discrete values for this attribute, indicate that 642 (32.1%) people are Unemployed, 340 (17%) people are Employed and 5 (0.25%) people are Retired.

The default retirement age (formerly 65) has currently been phased out, now allowing people to work as long as desired. However for this dataset, the former retirement age of 65 has been used. Only 5 people from the entire dataset are retired, none of which are over the age of 65.

The 5 customers who have an Employment Status of retired are considered to be outliers, and should remain in the dataset. This data will be relevant to determine the percentage of retired people, who have previously paid debts, and who have previously defaulted. However, the data amount is considerably small, and should be treated as such (there are limitations on its reliability).

Employment Status	Self Employed	Unemployed	Employed	Retired
Count	1013	642	340	5
Percentage	50.65%	32.1%	17%	0.25%

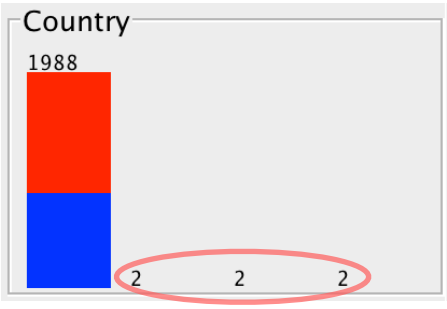
2.7 Country

Nominal

There are four distinct values for the Country attribute: Germany, France, Spain and UK. The majority of values (1994) are UK, leaving a very small amount of 6 value (2 per) relating to the other three remaining countries.

Due to the extremely small dataset size for countries excluding the UK, it is impossible to retrieve a reliable result for Germany, France and Spain. Therefore, the only possible use for the Country attribute would be to find a common trend for customers in the UK, however it is more than likely that the Country attribute will not be used.

Country	Germany	France	Spain	UK
Count	2	2	2	1994
Percentage	0.1%	0.1%	0.1%	99.7%



The histogram (left) indicates that the number of values for the country attribute that are non-UK, are so few, that a valid representation of paid and defaulted attributes cannot be shown. This is the justification for their removal in the final dataset, a reliable trend in results cannot be obtained.

2.8 Current Debt

Numeric

The Current Debt attribute has 788 distinct values with 15% (294) unique values. Current debt is continuous as there is no finite amount of debt a person can have. Current debt will make it harder for a person to take out new loans, and repay current loans, the higher the debt amount is.

Therefore it can be hypothesised that a person with a high current debt and an income which is less than their current debt, will not repay future loans, however this is still to be proven. This attribute will be useful in predicting the probability that a person will repay future loans, as it provides a statistical figure of past debts. Debt can be interpreted as a person who struggles with money, whereby a loan is needed in the first place.

Current Debt	Unedited Dataset
Minimum Value	0
Maximum Value	9980
Mean	3309.325
Standard Deviation	2980.629

2.9 Postcode

Nominal

The Postcode attribute has 1971 distinct values, and has 0% missing values. This confirms that multiple customers live in the same area (the same postcode) where trend may develop. However there are too few numbers to create any significant trends, and much like the gender attribute, discrimination based on living location is not permitted.

The usefulness of this attribute is therefore nil, and a solution should not regard this attribute for the probability rate of future loan repayments, based on postcode.

This attribute may be used in other circumstances outside the given scenario, i.e. marketing purposes.

2.10 Income

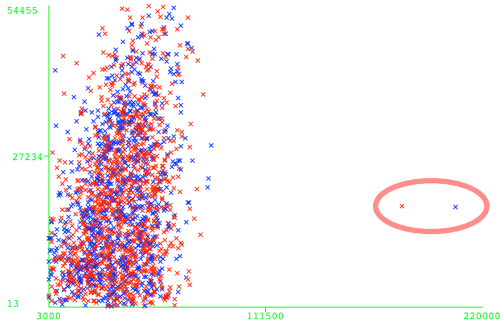
Numeric

The Income attribute has 100 distinct values with 7 unique values (rounded to 0%), this is a continuous amount as there is no finite amount of income a person can have, due to varying salaries, and other sources i.e. commission.

This attribute has great importance especially when evaluated against Current Debt. It can be hypothesised that a person with a low income i.e. 3000, who has a similar amount or higher in current debt, it is unlikely that person will repay future loans.

Trends may also be established using multiple attributes i.e. the higher a persons income, the less current debt they have, except for people with an income of £30,000 - £40,000. This is purely an example, and trends will be acknowledged in following chapters.

Income	Unedited Dataset
Minimum Value	3000
Maximum Value	220000
Mean	38319
Standard Deviation	12786.506



The scatter plot (right) represents loan amount against income, the 2 outliers are shown, as they significantly vary from the average loan amounts, however there are only 2 values whereby 1 has paid and 1 has defaulted.

2.11 Own Home

Nominal

The Own Home attribute has 3 distinct values and may provide valuable data in whether future loan repayments will be secured or not. This attribute in conjunction with another attribute i.e. Income may provide some solid statistics. Example: people who own their home, and have an income greater than £30,000, are likely to repay their loans, this is a hypothesis however, and will be examined in a later chapter.

The hypothesis stated above in conjunction with logical reasoning can determine that applicants who own their home with a comfortable income, can likely pay their mortgage and repay future loans with ease, whereas an applicant who rents their home may struggle, due to the outgoing payments each month.

These is no noise for this attribute, as there are only 3 possibilities for home ownership, and there is 0% missing values.

Own Home	Rent	Own	Mortgage
Count	292	1031	677
Percentage	14.6%	51.55%	33.85%

2.12 CCJs

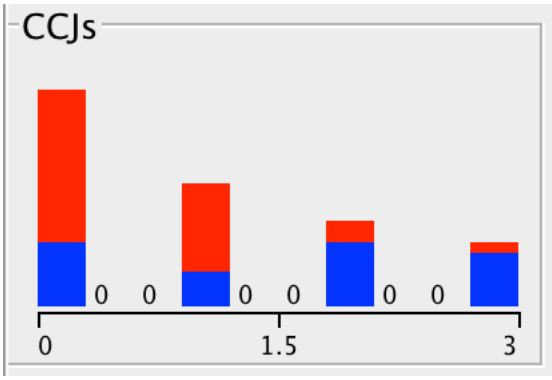
Numeric

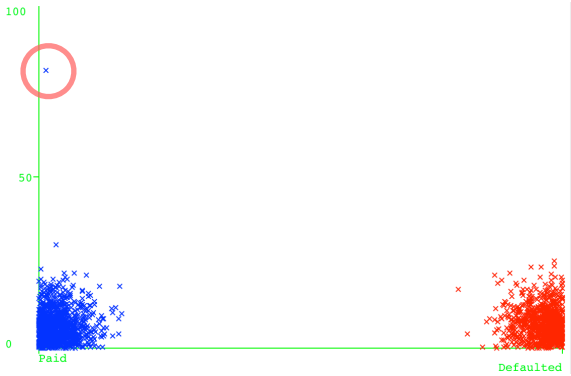
The CCJs (County Court Judgement) attribute for this dataset has 6 distinct values, ranging from 0 to 100. A CCJ may be issued if court action has been taken against a person, claiming the person owe money. A CCJ will be issued if the court has formally decided a person owes money, and will explain how much money is owed, and the deadline for payment.

CCJs suggest a person is unreliable in their repayments of debts, whereby a higher number indicates a greater risk for loan repayments, making it difficult for a person to secure loans or credit. Therefore this attribute will be of high value in providing a result for the probability of future loans being repaid by customers.

This attribute has 6 distinct values shown below, the record listing a customer with 100 CCJ's is most likely to be the result of a data error (noise). The likelihood of a person aged 27 with 100 CCJs is low. This record could therefore be altered to be 1 or 10, or deleted as the actual value cannot be determined for definite, and the removal of 1 value is insignificant. The same can be said for the person with 10 CCJs, as this is the only other unique CCJ value, as the remaining CCJ values have many instances.

CCJs	0	1	2	3	10	100
Count	886	497	347	268	1	1
Percentage	44.3%	24.85%	17.35%	13.4%	0.05%	0.05%





The histogram (left) shows that customers with lower CCJs (0 - 1) mostly defaulted (RED), whereas with an increase to the number of CCJs (2 - 3), the majority of customers actually paid (BLUE).

The scatter plot (right) represents the number of CCJs against the outcome of either paid or defaulted. The noise can clearly be seen here, the customer who had 100 CCJs paid, which is clearly an error in data entry; 100 CCJs is near impossible.

As the '100' value significantly deviates from the other set numerical values, consistently alternating between 1 - 3, it may be classed as an outlier if the scenario was different i.e. the IQ of a scientifically recognised figure, against a group of primary school children. However as the scenario relates solely to the amount of CCJs, the probability of the 27 year old customer, having 100 CCJs, is extremely unlikely, and has been addressed as noise.

2.13 Loan Amount

Numeric

Loan Amount is a continuous amount as there is no finite amount of loans a person can have, unless they are declined loans by the bank or other financial establishment.

The minimum value present is 13, followed by numerous similar values i.e. 35, 45, 50. These values are considered to be noise as it is unlikely a bank would loan small amounts of £13. If the value of 13 is an input error, the actual amount is uncertain, as the value could have been £1,300 or £13,000.

The cut-off line for the removal of these values can only be estimated, due to the financial standards whereby most established bank loans begin at £1,000. However the alternative is that people may take out payday loans from third-party websites, such as Wonga and Kwik Cash, whereby these loans are valid amounts.

Loan Amount	Value
Minimum Value	13
Maximum Value	54455
Mean	18929.628
Standard Deviation	12853.189

2.14 Outcome

Nominal

The Outcome attribute has 2 distinct values of either ‘Paid’ or ‘Defaulted’, which provides credibility on the ability to repay future loans. The outcome attribute is based on whether the person paid or defaulted on the loan they had, at the collection time of data, concerning their income, loan amount and current debt.

The amount of people who paid and defaulted, in both the unedited dataset, and dataset after manipulation (removal of noise), can be seen in the table below.

Outcome	Unedited Dataset	Edited Dataset
Paid	1118	1115
Defaulted	882	879

With the manipulated dataset, both the amount of people who paid and defaulted have been reduced by 3, leaving consistent outcome values, with a cleaner dataset.

3.0 Data Mining Technique Selection

3.1 Decision Trees

General

A decision tree is represented by a leaf and node structure, each attribute is represented by a node, which then branches multiple times in to further attributes, until a final decision has been made (a leaf). A common method of producing decision trees is to use TDIDT.

First Phase

The TDIDT method (top-down induction of decision trees) has two phases, the first of which is to build or 'grow' the decision tree. This is done by determining the 'best' attribute in the provided dataset, which will become the root node of the decision tree. The best attribute is the one that partitions the class attribute the most purely. Ideally an attribute that can present a pure node will be used, however if this is not possible, the most pure attribute will be used. A pure node indicates that all samples of that node have the same class label, and there is no need to further split the node in to additional branches.

Second Phase

The second phase is to 'prune' the tree, which is achieved using a bottom-up method, alternating from the top-down method in the first phase. Any data that is deemed to be noise (this may be uncertain but logical deduction should be applied here), should be pruned, this would remove the node from the decision tree, creating a leaf. Essentially pruning reduces classification errors in the presence of noisy data.

Ross Quinlan developed the first classification decision tree known as ID3 (Iterative Dichotomiser 3) in 1986, which has been developed in to a more general algorithm called C4.5 (1993), which can use continuous attributes in addition to discrete.

C4.5 Algorithm

The C4.5 algorithm is used to generate decision trees used for classification problems from training sets, which can deal with both continuous and discrete attributes, in addition to missing values. The decision tree is pruned after creation, replacing irrelevant branches with leaf nodes.

The dataset for the given financial establishment includes both continuous and discrete attributes, which this algorithm now caters for. This method works well with noisy data which the dataset contains, and may therefore not have to be removed due to the lack of effect they would have upon the end data result.

This algorithm can be enhanced post-completion using the pruning method, described above, and differentiating test options to 'train the data' also provide varying outlooks on the dataset i.e. cross-validation, percentage split and training sets. A training set is a section of a dataset used on a specific model, to discover potentially predictive relationships throughout known values of a dataset, for future unknown data [1].

Pros and Cons of Decision Trees

Decision trees are usually easy to interpret visually which is advantageous, however upon large sets of data, decision trees can become large, even after pruning has been completed. Therefore over-complex decision trees may be generated, making data interpretation difficult and ungeneralised.

Pruning presents the factor that very little data preparation is required. Noise can generally be dealt with via pruning, and is not critical to the entire accuracy of the decision tree generated. A white box model is used as the data is known to the tester, and can be explained using result logic.

Decision trees are usually a very fast method of data mining, which can classify unknown records with little effort [9].

Entropy and Information Gain

Entropy is a common way to measure impurity, a perfect 50-50 split provides good training data. Information gain is based on the decrease of entropy (or impurity) after a dataset has been split on an attribute. This would then find the most homogenous node, providing the most pure results. The formula below measures the reduction of entropy achieved from a given split [3].

$$GAIN_{split} = Entropy(p) - \left(\sum_{i=1}^{k} \frac{n_i}{n} Entropy(i) \right)$$

Pseudo-Code for C4.5

The pseudo-code for the C4.5 algorithm is shown below, whereby the algorithm consists of a set S of examples, the pseudo-code has been referenced [6].

Check **for** base **case**.

Find the attribute with the highest
informational gain (A_best)

Partition S into S1,S2,S3... according to the
values of A_best

Repeat the steps **for** S1,S2,S3...

Overfitting

Decision Trees grown until each leaf node has the lowest impurity possible may overfit, whereby the algorithm reduces the training set error at the cost of an increased test set error. Overfitting generally occurs in complex models with many parameters, noise may be described instead of legitimate relationships. Post-pruning effectively solves this issue by allowing the tree to classify the training set, and then the nodes are pruned (post) [8, 9].

3.2 Naïve Bayes

General

Bayesian refers to Reverend Thomas Bayes and his solution to the theory of probability, known as Bayes' Theorem published in 1763. Naive Bayes can be used to understand how the probability that a theory is true, is affected by another piece of evidence [2].

The Naive Bayes model has been applied in real world scenarios i.e. the averaging to predict Alzheimer's disease from genome-wide data [5]. A significant number of records (1411) made use of the classifiers strengths dealing with large amounts of data.

Naive Bayes assume conditional independence:

> *'Naive Bayes makes the assumption that each predictor is conditionally independent of the others. For a given target value, the distribution of each predictor is independent of the other predictors. In practice, this assumption of independence, even when violated, does not degrade the model's predictive accuracy significantly, and makes the difference between a fast, computationally feasible algorithm and an intractable one.'* [4]

The goal of Naive Bayes is to work out whether a new example is in a class given that it has a certain combination of attribute values. We work out the likelihood of the example being in each class given the evidence (its attribute values), and take the highest likelihood as the classification [3].

The conditional probability of an event B in relationship to an event A, is the probability that event B occurs given that event A has already occurred.

$$P(B|A) = \frac{P(A \text{ and } B)}{P(A)}$$

The formula (right) is for conditional probability, P(BIA) read as the probability of B given A.

To calculate the probability of B given A, the algorithm counts the number of cases where A and B occur together and divides it by the number of cases where A occurs alone [4].

Pros and Cons of Naive Bayes

Naive Bayes' assumption that variable are all equally important and independent of one another is often untrue. Redundant attributes that are included also hinder Naive Bayes. Strong dependencies i.e. a person with high income has an expensive house, unfairly multiplies the effect of having low income. Attribute values will be assigned a zero probability if they are not present in the data. Small positive values which estimate the so-called 'prior probabilities' are often used to correct this.

Naive Bayes often provide practical results, and the algorithm is somewhat robust to noise and irrelevant attributes. A large number of records must be present to achieve the best results from using this classifier [10].

The "zero-frequency problem"

The solution is to never allow zero probability. An attribute value that does not occur with every class value e.g. ('Humidity = high' for class 'yes'). Pr[Humidity = high I yes] = 0. The probability will be zero independent of how likely other values are.

The fix is to add 1 to the count for every attribute value-class combination (Laplace estimator), this keeps a consistent set of values, whilst never allowing a zero value.

4.0 Data Exploration and Preparation

4.1 Relevant Data

As discussed in chapter 1 the attributes that will be used are Age, Years at Address, Employment Status, Current Debt, Income, Own Home, CCJs, Loan Amount and Outcome.

Therefore the attributes that have not been included in data preparation include Customer ID, Forename, Surname, Gender, Country and Postcode.

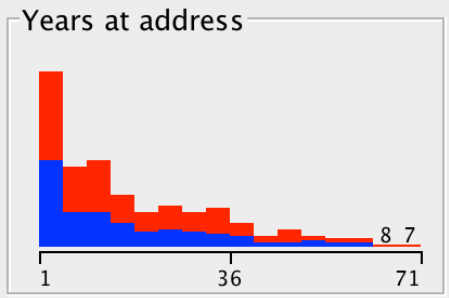
Previous discussions in chapter 1 describe the discrimination surrounding the gender of a person, and their home location, whereby these attributes are no longer used to evaluate whether an applicant should be eligible for future loans or not. This effectively relates to postcode, as the country attribute has very few records outside the UK, that reliable trends cannot be established.

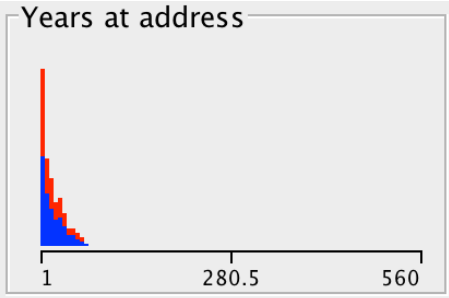
4.2 Noise

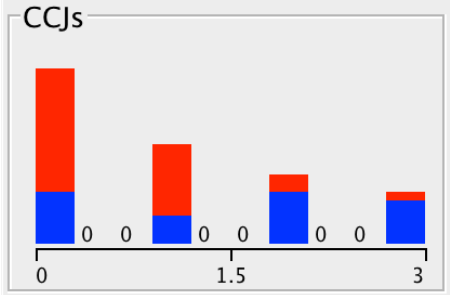
The total amount of records post elimination of noisy data, returns a total of 1994, 6 records have been deleted from the dataset.

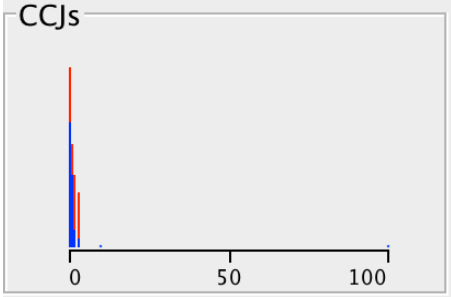
Years at Address and CCJs

The 4 records that have been removed are due to 2 assumed incorrect attributes; CCJs and Years at Address. The values recorded within these attributes are almost certainly noise, i.e. 560 for Years at Address and 100 for CCJs. The Years at Address figures could easily be misinterpreted and entered as months instead of years, and CCJs should not be extensive figures, therefore it was easier to label these records as noise, and remove them, as the large dataset will accommodate for very few records being removed.





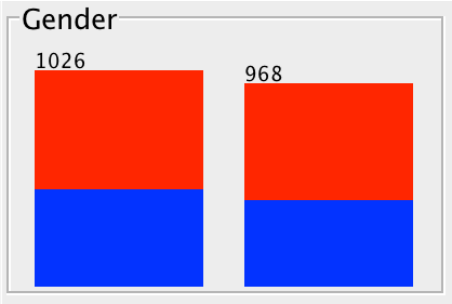


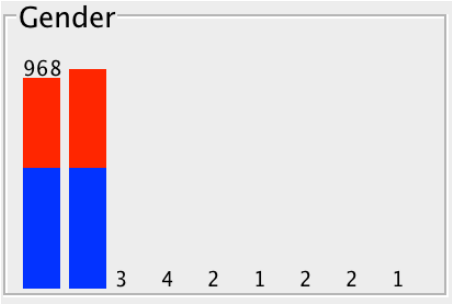


The histograms above are created from the edited dataset (removal of noise) on the left, and the unedited dataset on the right. It is apparent that the CCJ and Years at Address attributes that have been removed are noise and not outliers, which hold no significance.

Gender

Initial cleanup of data removed noise values to represent valid F and M genders. The edited dataset histogram is on the left, the unedited dataset is on the right. The removed





values have been labeled as noise as they hold very little significance, 1 and 0 may represent M and F however these were uncertain, logical deduction has categorised these uncertain values as M and F respectively, based on the Forename of customers, as this is the only attribute that may identify the relevant gender.

There are consistent errors within the Forename, Surname and Gender attributes, as different formats are used i.e. first initial and complete surname, complete name or irrelevant punctuation found within names. Gender also suggests irrelevance, as David is classed as ‘F’ for female.

4.3 Outliers

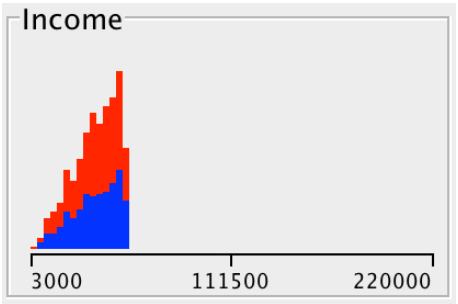
Interesting Attributes (Income and Country)

The Income attribute may hold some interesting information as Outliers include potentially include values of 180000 and 220000. It may be possible to deduce the possibility of whether a customer will default or pay future loans with this attribute.

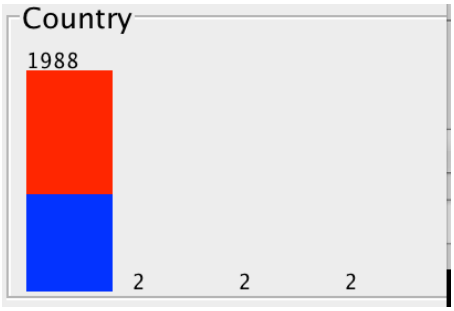
Age: 17 | Income: 220000 | CCJs: 1 | Outcome: Defaulted

Age: 42 | Income: 180000 | CCJs: 0 | Outcome: Paid

Assumptions may include that a young person with a high income and previous CCJs is likely to default on future loan repayments, whereas an older person with a high income and no previous CCJs is likely to repay future loans.



The histograms to the right represent unchanged data in the edited dataset, as the outliers have not been deleted, however the country attribute has lost 6 records due to removal of noise in the CCJ and Years at Address attributes.



The other 6 country values will not be used, as there are not enough records to represent any accurate trends, with differentiating country dependencies.

5.0 Modelling

Experiment One

Investigate Training and Testing Strategies

Aim: to investigate whether 50:50 or cross-validation gives the best classification accuracy when using Naive Bayes and J48.

Methodologies

The two data mining techniques I have chosen in chapter 3 will be modified by individual parameters to produce to purest nodes, and therefore the most accurate data (where the percentage of correctly classified instances is the highest). WEKA's implementation of the c4.5 decision tree algorithm is called J48, which will be tested in addition to Naive Bayes.

The first initial experiment will determine whether a 50:50 split or cross validation training method will produce the highest percentage of correctly classified instances.

50:50 Percentage Split

This testing methods splits the dataset in to sections based on the percentage chosen, whereby the algorithm applies rules learnt from the training data, then applies the rules to the testing data, which will then present the results to be used.

A 50:50 split seems to be the most reliable as the entire dataset is evenly split, without bias. A 90:10 split produces a slightly higher classification accuracy, however the sample size is too small to form a valid opinion based on reliable data.

The 50:50 split (although producing a slightly lower classification accuracy) has a reliable sample size, and evenly distributes (splits) training and testing data.

Cross-Validation (10 fold)

This testing method splits the dataset in to (my desired number of folds) 10 equal sections, the algorithm learns from the first section or fold, and applies the 'rules' its learnt to the other remaining folds.

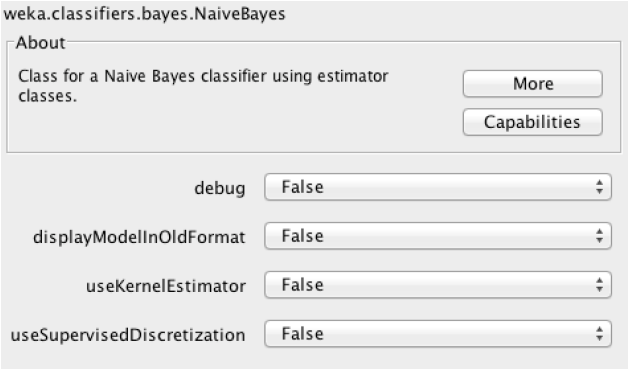
> *'Extensive tests on numerous different datasets, with different learning techniques, have shown that 10 is about the right number of folds to get the best estimate of error.'* [7]

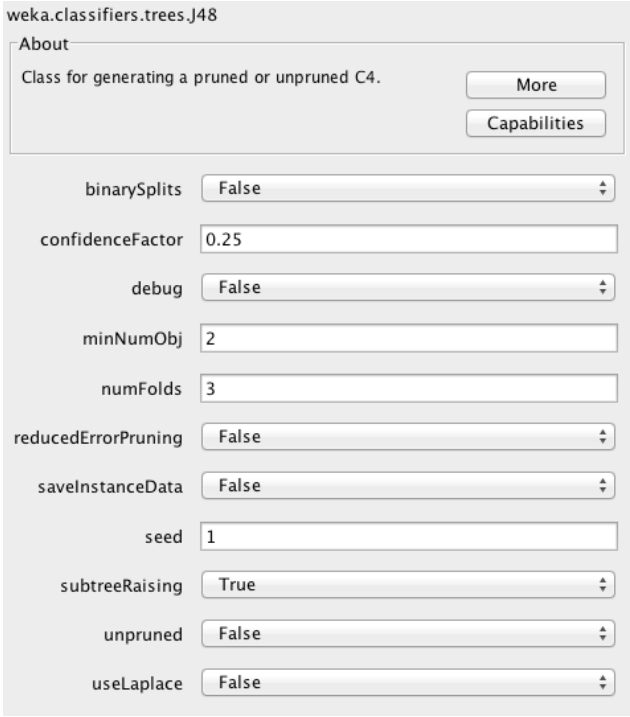
If the dataset were the total 2000 records, of 10 folds, each fold is divided into two groups; 1800 records used for training and 200 records used for testing.

A classifier is produced with an algorithm from the 1800 records, and applied to the 200 records for each fold. Classifiers are produced for each fold, whereby the average number of cases predicted incorrectly is used, which produces the classification accuracy percentage and confusion matrix. The higher the correctly classified instance percentage, the more accurate the test is.

The results from the 2 testing options for each methodology will be compared for both J48 and Naive Bayes. I will then continue to use the 2 methodologies with the testing option that provided the highest correctly classified instance percentage, for the remainder of the experiments i.e. Naive Bayes may be more accurate using crossfold-validation whereas J48 may be more accurate using percentage split.

For the initial test all parameters for both J48 and Naive Bayes were fixed to their defaults as stated in WEKA, the screenshots below indicate the parameter defaults. The parameters will be changed individually in later experiments to determine which is the optimal classifier, by enhancing (increasing) the correctly classified instance percentage.





The table below indicated the training and testing methods used on the J48 and Naive Bayes, and their classification accuracy. The percentage of records for the paid and defaulted attributes are also shown.

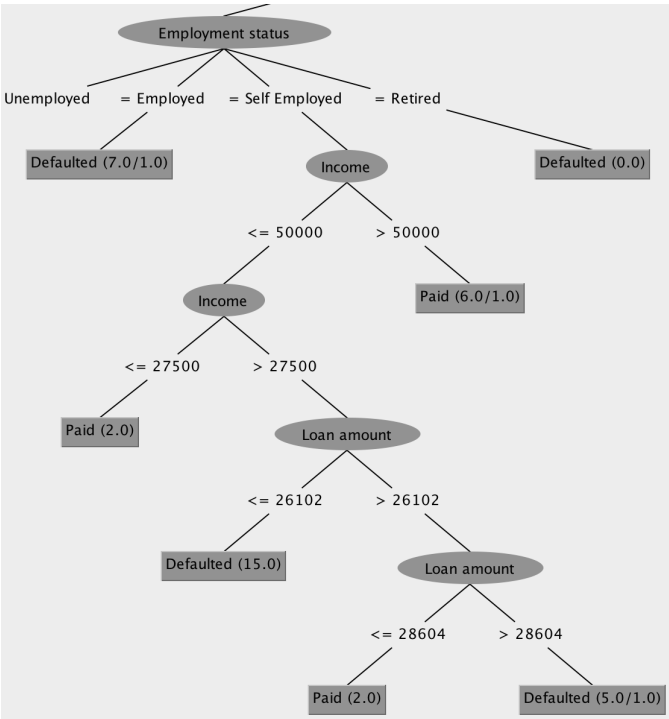
Green / red cells indicate the training and testing methods that will be used and ignored (respectively) for other experiments.

Model	Training Testing	Paid (%)	Default (%)	Classification Accuracy (%)
J48	50:50	88.13	70.04	80.44
J48	CV	84.39	72.01	78.93
NB	50:50	90.99	64.85	76.62
NB	CV	83.58	63.48	74.72

For the following experiments concerning J48, only the 50:50 percentage split training method will be used, as this provided the highest classification accuracy percentage.

Similarly, for Naive Bayes, experiments will continue to use the 50:50 percentage split training method, as this too provided the highest classification accuracy percentage.

Decision Tree



The decision tree in its entirety was too large to show on screen, the image to the left shows a section of the decision tree.

The decision tree shown related to the J48 algorithm, using the percentage split training method.

Confusion Matrix

The confusion Matrices below are labelled accordingly to the data mining techniques chosen, and training methods used. The confusion Matrices (right) generate the following percentages of correctly classified paid and defaulted records (indicated by P and D respectively):

J48 PS [D = 70%] [P = 88%]
J48 CV [D = 72%] [P = 84%]

NB PS [D = 64%] [P = 85%]
NB CV [D = 63%] [P = 83%]

On average J48 (Percentage Split) has the highest average accuracy of correctly classified instances, of paid and defaulted records, with a total figure of 79%. These figures are reliable as it is extremely close to the classification accuracy of the model (shown on the previous page).

J48 (Percentage Split)

```
=== Confusion Matrix ===

   a   b   <-- classified as
 297 127 |   a = Defaulted
  68 505 |   b = Paid
```

Naive Bayes (Percentage Split)

```
=== Confusion Matrix ===

   a   b   <-- classified as
 275 149 |   a = Defaulted
  84 489 |   b = Paid
```

J48 (Cross-validation)

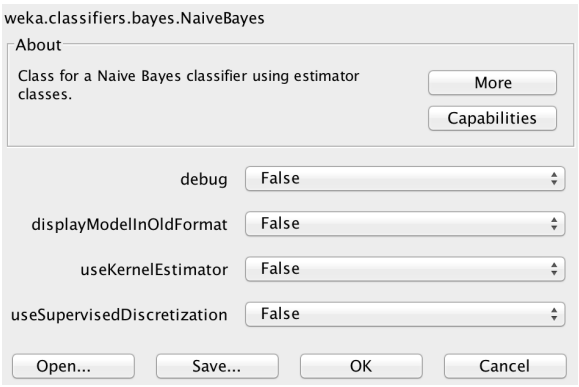
```
=== Confusion Matrix ===

   a   b   <-- classified as
 633 246 |   a = Defaulted
 174 941 |   b = Paid
```

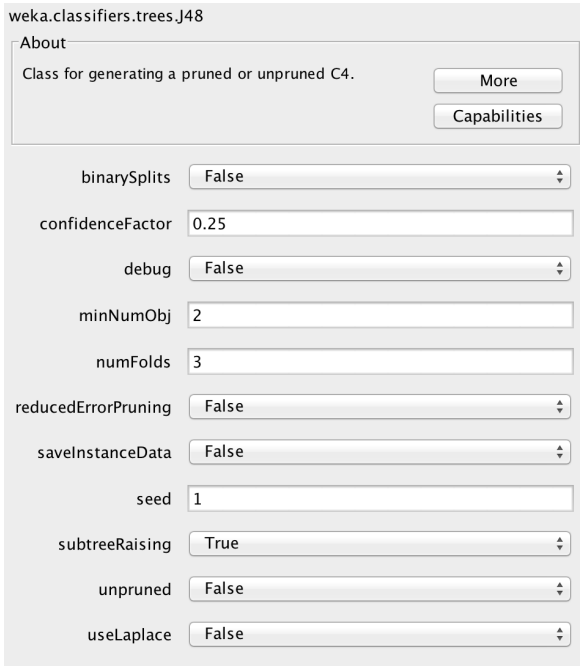
Naive Bayes (Cross-validation)

```
=== Confusion Matrix ===

   a   b   <-- classified as
 558 321 |   a = Defaulted
 183 932 |   b = Paid
```

Parameters

The two images indicate the parameters that can be changed (one at a time) for each classifier, that will be performed in experiments 2, 3, 4 and 5. J48 (right) will have parameters binarySplits and unpruned changed. Naive Bayes (left) will have parameters useKernelEstimator and useSupervisedDiscretisation changed.





Parameters were changed one at a time, the confusion matrix and correctly classified instances percentage were noted. The parameter was then changed back to its default setting, allowing for another parameter to be changed. This ensured greater reliability, as if multiple parameters were changed at the same time, it would be difficult to determine which parameter had the most noticeable effect on the overall outcome.

Experiment Two (J48)

Investigate the Effectiveness of Pruning

Aim: to investigate whether using pruning gives a better classification accuracy.

Methodology

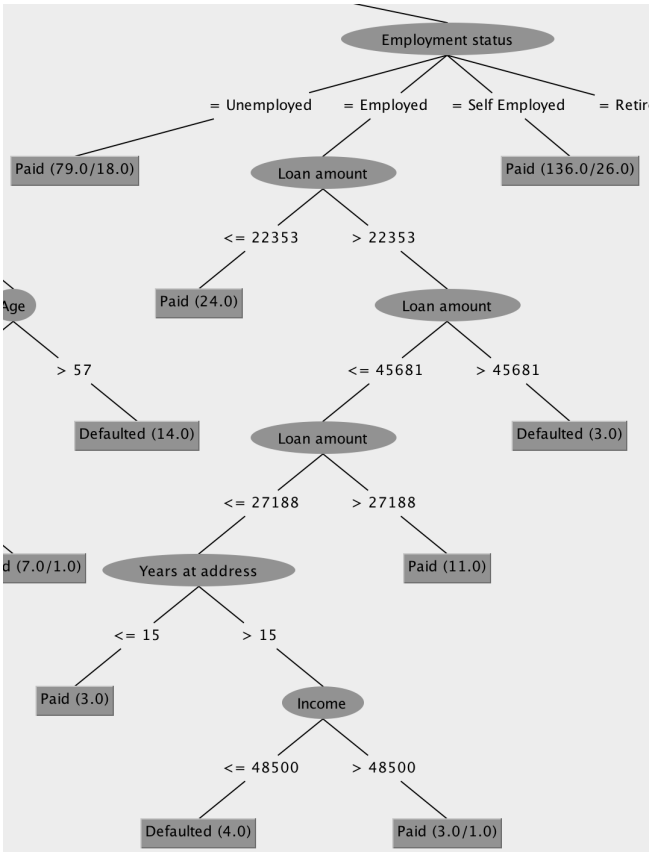
Change Unpruned to true, to determine how the correct classification percentage changes.

Parameter

By default, the 'Unpruned' parameter is set to false, indicating the decision tree has been pruned. This experiment will change the parameters value to true, to investigating whether an unpruned tree will give a higher classification accuracy percentage.

```
Correctly Classified Instances         776               77.8335 %
Incorrectly Classified Instances       221               22.1665 %
Kappa statistic                          0.5353
Mean absolute error                      0.306
Root mean squared error                  0.4418
Relative absolute error                 61.9997 %
Root relative squared error             89.1798 %
Total Number of Instances              997

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure
                 0.646     0.124       0.794    0.646      0.713
                 0.876     0.354       0.77     0.876      0.82
Weighted Avg.    0.778     0.256       0.78     0.778      0.774

=== Confusion Matrix ===

   a   b   <-- classified as
 274 150 |   a = Defaulted
  71 502 |   b = Paid
```

The decision tree generated using the J48 algorithm, 50:50 percentage split training method, has a higher classification accuracy when pruning is present. Un-pruning the tree drops the accuracy by 2.61%, to a final figure of 77.83, from 80.44.



The confusion matrix above is highly accurate if the company significantly cared about customers who have paid their loans, however the correctly classified accuracy percentage for defaulted records is quite low.

D = 64% P = 87%

The unpruned decision tree increases its size significantly, shown (left), whereby Employment Status shows each of the 4 possible attribute values.

This form of generated decision tree will have greater value if the company decides to drill down in to specific attributes held by customers, for marketing purposes.

I.e. if the company wishes to view records of customers who are retired, and have a certain loan amount greater than a set figure, an unpruned tree is the best option.

Experiment Three (J48)

Investigate the Effectiveness of Binary Splits

Aim: to investigate whether using binary splits gives a better classification accuracy.

Methodology

Change Binary Splits to true, to determine the correct classification percentage change.

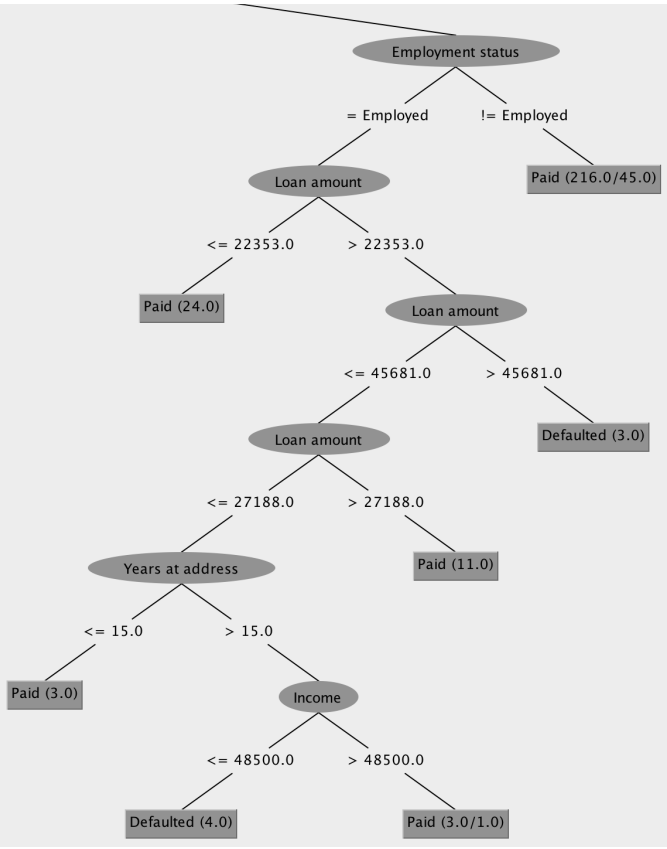
Parameter

By default the Binary Splits parameter is set to false, turning this parameter to true will use binary splits on nominal attributes when building the decision tree.

```
Correctly Classified Instances         795               79.7392 %
Incorrectly Classified Instances       202               20.2608 %
Kappa statistic                          0.5785
Mean absolute error                      0.3114
Root mean squared error                  0.4085
Relative absolute error                 63.1018 %
Root relative squared error             82.4718 %
Total Number of Instances              997

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure
                 0.698     0.129      0.8       0.698     0.746
                 0.871     0.302      0.796     0.871     0.832
Weighted Avg.    0.797     0.228      0.798     0.797     0.795

=== Confusion Matrix ===

   a   b   <-- classified as
 296 128 |   a = Defaulted
  74 499 |   b = Paid
```

The decision tree generated using the J48 algorithm, 50:50 percentage split training method, has a higher classification accuracy when binary splits are not used. Using binary splits drops the accuracy by 0.71%, to a final figure of 79.73, from 80.44.

Binary Splits also generates a high accuracy percentage for the paid attribute, however the defaulted accuracy is still quite low, higher than the pruning parameter yet lower than the default parameter settings.



D = 69% P = 87%

The tree has been pruned as the unpruned tree generated a lower correctly classified instance percentage.

As the accuracy was less reliable, reverting the pruning parameter, then changing a second parameter; enabling binary splits, generated a higher correctly classified instance percentage, however it was still less than the complete default set of parameters.

However when binary splits are enabled the decision tree is a lot more readable. The specifics of attributes are not as visible i.e. retired and self employed attribute, as only employed and not employed are visible.

Experiment Four (Naive Bayes)

Investigate the Effectiveness of Kernel Estimator

Aim: to investigate whether using kernel estimator gives a better classification accuracy.

Methodology

Change using Kernel Estimator to true, to determine how the correctly classified instances percentage changes, if the change positively effects the accuracy, it will be kept, however a negative effect will ensure the parameter change is reverted.

Parameter

By default the use of Kernel Estimator parameter is set to false, turning this parameter to true will use kernel estimator for numeric attributes rather than a normal distribution.

```
Time taken to build model: 0.02 seconds

=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances         756               75.8275 %
Incorrectly Classified Instances       241               24.1725 %
Kappa statistic                          0.4961
Mean absolute error                      0.3759
Root mean squared error                  0.4279
Relative absolute error                 76.1701 %
Root relative squared error             86.3902 %
Total Number of Instances              997

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   RO
                 0.644     0.157      0.752     0.644     0.694
                 0.843     0.356      0.762     0.843     0.8
Weighted Avg.    0.758     0.271      0.758     0.758     0.755

=== Confusion Matrix ===

   a   b   <-- classified as
 273 151 |   a = Defaulted
  90 483 |   b = Paid
```

The Naive Bayes classifier using 50:50 percentage split training method, has a higher classification accuracy when kernel estimator is not used. Using kernel estimator drops the accuracy by 0.8%, to a final figure of 75.82, from 76.62.

The Kernel Estimator compensates for attributes which in the dataset are not normalised, attempting to provide smoothing to the data.

Using Kernel Estimator lowered the classifier accuracy percentage for both paid and defaulted values, and therefore should be avoided, as enabling this parameter provided no beneficial factors.

D = 64% P = 84%

Workings

Defaulted: 273 / (273 + 151) = 0.64

Paid: 483 / (483 + 90) = 0.84

Using the sum of the two numbers generated:

(0.64 + 0.84) / 2 = 0.74 * 100 = 74 **(74 is the percentage value)**

This number is significantly close to the correctly classified instance accuracy of 75.82%

Experiment Five (Naive Bayes)

Investigate the Effectiveness of Supervised Discretisation

Aim: to investigate whether using supervised discretisation gives a better classification accuracy.

Methodology

Change the Supervised Discretisation parameter to be true, to determine whether this positively effects the correctly classified instance percentage, if it does, the parameter change will be kept, if it negatively effects the accuracy, the change will be reverted.

Parameter

By default the Supervised Discretisation parameter is set to false, turning this parameter to true will use supervised discretisation to convert numeric attributes to nominal ones.

```
Time taken to build model: 0.03 seconds

=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances         776               77.8335 %
Incorrectly Classified Instances       221               22.1665 %
Kappa statistic                          0.5364
Mean absolute error                      0.3682
Root mean squared error                  0.4183
Relative absolute error                 74.6027 %
Root relative squared error             84.439  %
Total Number of Instances              997

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC
                 0.656     0.131      0.788     0.656     0.716      0
                 0.869     0.344      0.773     0.869     0.818      0
Weighted Avg.    0.778     0.254      0.779     0.778     0.775      0

=== Confusion Matrix ===

   a   b   <-- classified as
 278 146 |   a = Defaulted
  75 498 |   b = Paid
```

The Naive Bayes classifier using 50:50 percentage split training method, has a higher classification accuracy when supervised discretisation is not used. Using supervised discretisation drops the accuracy by 2.61%, to a final figure of 77.83, from 76.62.

Enabling Supervised Discretisation is the only experiment that positively impacted the classification accuracy of the percentage for both the Paid and Defaulted attributes. The following percentages are Naive Bayes with all default parameters, with exception to Supervised Discretisation.

D = 65% P = 86%

Enabling the parameter saw an increase of 1% accuracy for both paid and defaulted attributes up from 64% and 85% respectively.

6.0 Conclusion

The table below conclusively shows that J48, 50:50 percentage split, with default parameters, correctly classified the most instances of paid and defaulted attributes.

J48 is the best classifier all round as no matter what parameters were changed, the outcome also favoured the J48 classifier over Naive Bayes in terms of accuracy.

The only experiment, and change in parameters that positively effected either of the classifiers, was the enabling of supervised discretisation for Naive Bayes. Enabling this parameter increased the accuracy of both paid and defaulted instances by 1%.

However this still lacked behind the J48 classifier, which correctly classified paid instances 2% more often, and defaulted instances 5% more often.

6.1 Final Results

Experiments	J48 (PS) Paid %	Naive Bayes (PS) Paid %	J48 (PS) Defaulted %	Naive Bayes (PS) Defaulted %
Classifier Default	88	85	70	64
Pruning	87	-	64	-
Binary Splits	87	-	69	-
Kernel Estimator	-	84	-	64
Supervised Discretisation	-	86	-	65

If the financial establishment were interested in specific information for marketing purposes, or specialised events that may only concern a niche group of customers, J48 multiway splits may be the most beneficial, as binary splits generate a more readable (small-scale) decision tree, that categorises attributes more aggressively.

The highest Paid and Defaulted classification accuracy percentages have both been achieved by using the J48 classifier, 50:50 percentage split training method, without the default parameters. Parameter alternations only negatively effected the correct classification accuracy, and therefore should not be used.

If the results that the financial establishment was interested in solely concerned defaulted customers, the highest accuracy percentage would concern the defaulted attribute, unconditional of how low the accuracy percentage was concerning the paid attribute i.e D = 84% P = 62%.

However if the financial establishment wished to achieve reliable figures for both paid and defaulted attributes, a more evenly matched set of figures should be aimed for i.e. D = 75% P = 78%.

In either situation, the J48 classifier with default parameters provides the most reliable overall results.

7.0 References

[1] Statistics (2014)
[Online] Available from: http://www.statistics.com/glossary&term_id=864
[Accessed 09 March 2014]

[2] Trinity (2014)
[Online] Available from: http://www.trinity.edu/cbrown/bayesweb/
[Accessed 09 March 2014]

[3] Gerber, L (2014)
[Online] Available from: MMU Data Engineering (6G6Z1006_1314_9Z6)
[Accessed 09 March 2014]

[4] Oracle Docs (2008)
[Online] Available from: http://docs.oracle.com/cd/B28359_01/datamine.111/b28129/algo_nb.htm#BGBEEFIA
[Accessed 09 March 2014]

[5] Wei. W, Visweswaran. S, Cooper. G (2011)
[Online] Available from: http://jamia.bmj.com/content/18/4/370.abstract
[Accessed 10 March 2014]

[6] Octaviansima (2011)
[Online] Available from: http://octaviansima.wordpress.com/2011/03/25/decision-trees-c4-5/
[Accessed 10 March 2014]

[7] Witten. H, Elbe. F, Hall. M (2011) Data Mining: Practical Machine Learning Tools and Techniques : Practical Machine Learning Tools and Techniques
[Online] Available from: http://lib.myilibrary.com/Open.aspx?id=295388&src=0
[Accessed 10 March 2014]

[8] CSE.MSU.EDU
[Online] Available from: http://www.cse.msu.edu/~cse802/DecisionTrees.pdf
[Accessed 10 March 2014]

[8] Sayad. S (2012) An Introduction to Data Mining
[Online] Available from: http://www.saedsayad.com/decision_tree_overfitting.htm
[Accessed 10 March 2014]

[8] Sayad. S (2012) An Introduction to Data Mining
[Online] Available from: http://www.saedsayad.com/decision_tree_overfitting.htm
[Accessed 10 March 2014]

[9] Pedregosa (2012) Scikit-learn: Machine Learning in Python
[Online] Available from: http://scikit-learn.org/stable/modules/tree.html
[Citation] http://scikit-learn.org/stable/about.html#citing-scikit-learn
[Accessed 10 March 2014]

[10] Safari Books Online
[Online] Available from: http://my.safaribooksonline.com/book/databases/business-intelligence/9780470526828/naive-bayes/advantages_and_shortcomings_of_the_naive
[Accessed 12 March 2014]