31005 Advanced Data Analytics

Assignment 2: Practical Data Mining Project

Spring 2016

Table of Contents

1. Introduction

In this report, an existing data analytic package will be utilised to solve a data mining problem. The dataset will be analysed in detail and then multiple data analytical classifier techniques will be implemented to assist in solving the data mining problem.

The Cross Industry Standard Process for Data Mining, or CRIPS-DM model will be utilized to approach the data mining problem in this report. The six CRISP-DM phases which will be involved include understanding the business objectives, understanding the data, preparing the data, applying modelling techniques, evaluating the models and then the deployment of the knowledge and findings.

The dataset which will be used in this report is a census and income dataset. The dataset contains data from the 1994 to 1995 Current Population Surveys which was conducted by the United States Census Bureau (UCI Machine Learning Repository 2013). The business understanding and objectives of this project is to use is to use the data and create multiple classifiers to perform categorical predictions in order to determine whether a person's yearly salary is either greater than or less than $50,000. Census data which are categorized can be utilized for a range of situations, such as by the governments to track fraud and tax related crimes, all the way to allowing businesses and corporations to target specific demographics of the population.

The dataset will then be explored and analysed in detail. All the important and significant attributes within the dataset will be analysed and various statistics and interesting findings which are related will be mentioned or discussed. The next section will explain in detail the implementation of the various classifiers which will be utilized to solve this data mining problem. The implementation stages and configurations will be mentioned and various statistics and test results, will be discussed. After the classifiers are created and compared with each other, a recommendation on the most suitable classifier will be made and justified.

2. Exploration of the dataset

This section of the report will be exploring the data within the dataset. This section will cover the second phase and the third phases of the CRISP-DM model. The second phase is to assess and explore the dataset in order to understanding data and then the third phase involves preparation and the transformation of the dataset (Kennedy 2015). A detailed analysis of the dataset will be performed and various statistics from the census data will be presented as well as interesting and important findings will be explained. After the exploration of the data, pre-processing and cleaning of the data will be performed.

2.1. Analysis of the attributes

The most important attributes within the dataset will be explored and analysed in detail in this section of the report. Interesting findings such as correlations between attributes or any other important aspects within the data will also be mentioned and explained. Some of the attributes within this dataset will not be analysed, due to the insignificance of the data.

2.1.1. Salary Attribute

One of the most important attributes within the dataset is the salary. The reason for its importance is because this will be the attribute which the classifiers will be trained to predict and categorise. Salary is a nominal typed attribute which categorises into either less than $50,000 (<50,000) or greater than $50,000 (>50,000). These two categories are the only two distinct values for this attribute.

After analysis and calculations are made, it is determined that 6.21% of the total population of people in the census data earns greater than $50,000 a year, while the remaining 93.79% learns less than $50,000 a year. The diagram below (Figure 1 – Salary Histogram) visually shows the difference in proportion between the two categories. It should be noted that the population which earns less than $50,000 also include the children, students and retired population. Some of the people in these population groups would not be earning any money, and are grouped into earning less than 50,000 dollars a year category.

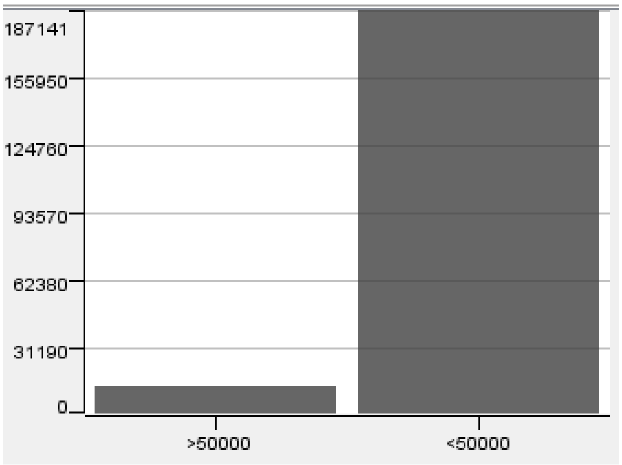


Figure 1 – Salary Histogram (x-axis shows the salary categories, y-axis shows the count)

2.1.2. Age Attribute

The first attribute of the census dataset is the “age”, which contains the ages of the people in the census. Age is a ratio typed attribute, as it is a measurement which has an origin at zero and the values are counted in years.

The age attribute contains 91 distinct values which range from 0 years old all the way to 90 years old, as shown in the box plot below (Figure 2 – Age Box Plot). These two ranges are also shown on the 10th and 90th percentile lines on the box plot, which also indicates that there are no other outliers greater than 90 and less than 0. Anyone with an age less than 0 should be non-existent as they are not even born yet and ages greater than 90 should be quite uncommon. The box plot diagram also shows the median, which is 33 years, and further calculations show that the mean or average age is 34.5 years. The greatest clustering of ages is between 15 and 50 years old, which is indicated by the 25th and 75th percentiles, meaning that at least half of the people in the census are aged between 15 and 50 years.

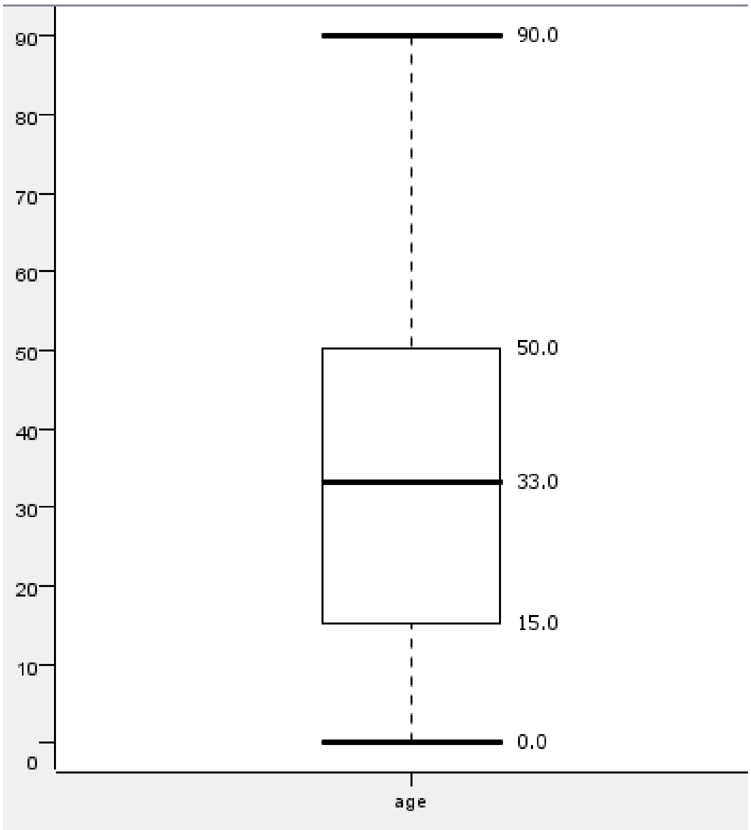


Figure 2 – Age Box Plot

Further analysis is performed to determine the frequency grouping of the ages. The age values are binned into a fixed amount of equal width bins, with the width of 10 years. The results of age binning process is shown in the diagram below (Figure 3 – Age (Binned) Histogram). This diagram further reinforces the findings from the box plot as the majority of the ages are grouped between 10 to 50 years old.

As the age increases, the number of population in that age group will start to decrease. This finding begins at around 50 years old and continues until 90 years old. This is a logical finding, as when people age, the mortality rate will increase, which reduces the total number of population in those age groups. This trend is also shown on the histogram below (Figure 3 – Age (Binned) Histogram).

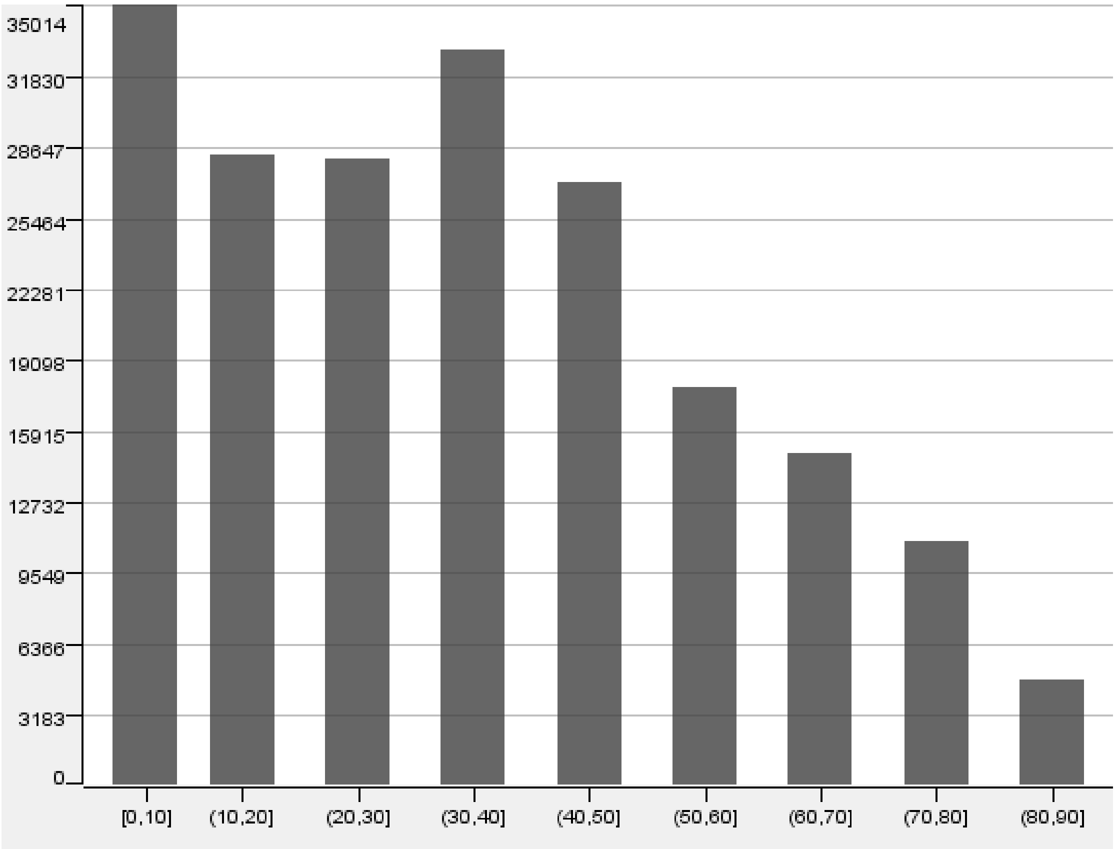


Figure 3 – Age (Binned) Histogram (x-axis shows age bins, y-axis shows the count)

2.1.3. Sex Attribute

The next attribute to be analysed is the "sex" of the people in the census data. This attribute only contains two different possible values, which are male and female. Through a brief analysis, it is calculated that 52.12% of all the people in the census data are female while the remaining 47.88% are males, which is show in the diagram below (Figure 4 – Sex Histogram). This shows that the census data is not specifically biased against a gender in particular, as the two sexes are roughly split with approximately half of the people in the census data being female and the other half being male. This almost equal split is a beneficial, as an equal range of data from both genders could allow the classifiers to be trained more accurately for both sexes instead of being more biased or more accurate towards one of the genders.

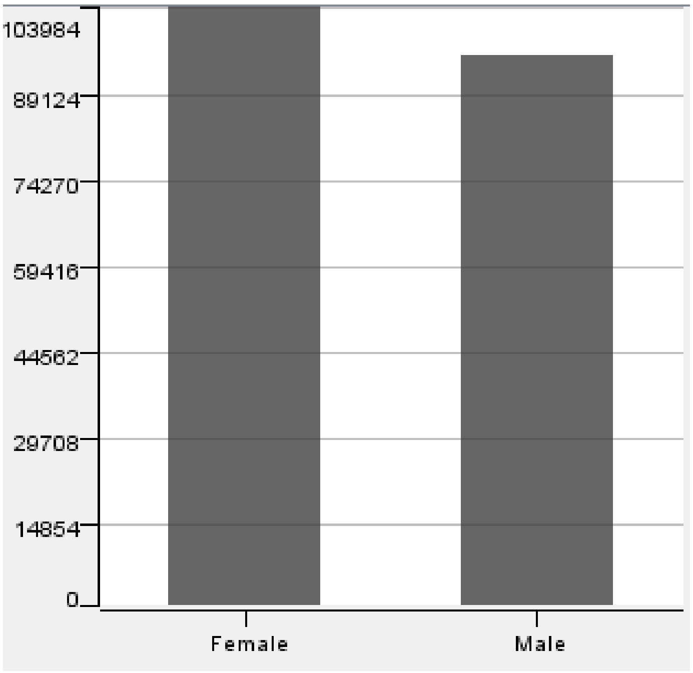


Figure 4 – Sex Histogram (x-axis shows the sex, y-axis shows the count)

2.1.4. Race Attribute

The “race” attribute contains the race of a particular person in the census. This categories in this attribute are nominal values and contains five distinct categories, which are White, Black, Asian or Pacific Islander, Amer Indian Aleut or Eskimo and Other. This analysis shows majority of the population or 83.88% of the population in the census are white, as show in the diagram below (Figure 5 – Race Pie Chart). The diagram also shows that the second largest race group, at 10.23% of the population in the United States, is black.

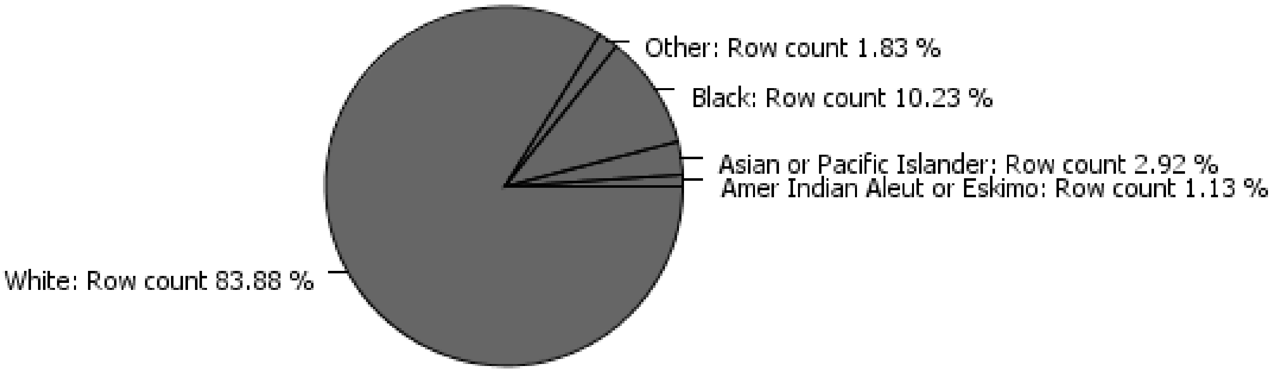


Figure 5 – Race Pie Chart

Comparing these statistics from the 1994 to 1995 US census dataset with the current 2015 and official statistics from the United States Census bureau, it is revealed that the current white population percentage is 77.1% and the black population is at 13.3% (United States Census Bureau 2016). These numbers are very similar to the calculated statistics, showing that the census dataset is not biased against a particular race. Since this dataset is only a small portion of the actual data which was obtained from the 1994 to 1995 census, this finding shows that this limited census dataset contains a relatively good representation of what the actual census data is like. This is a huge advantage as it allows the classifier to be trained with a good representation of real census data, and will allow the classifier to predict more accurately if it is used to classify real data.

2.1.5. Country of Birth and Citizenship Attributes

A group of closely related attributes are “country of birth father”, “country of birth mother”, and “Hispanic origin”. These attributes are all nominal typed and the country of birth attributes contains a range of 43 different country categories. The Hispanic origin attribute contains 10 different values which are either a Hispanic country, which means the person has a Hispanic origin or contains “not in universe”, indicating that the person does not have a Hispanic origin.

The attributes “country of birth father” and “country of birth mother” have correlation with the attribute “Hispanic origin.” This correlation means that there is a relationship between these three attributes and two of the attributes can determine the attribute of the other. An interesting finding is in most of cases in the dataset, if either one or both of the parents are from a Hispanic country, then the child will have very likely have a Hispanic origin. This discovery is logical, as if a person’s parents are from a certain heritage or background, then the child of the parents will be from the same heritage or background.

Another group of very closely related attributes is “country of birth self” and “citizenship”. These two attributes are also nominal typed and “country of birth self” contains a range of 43 different countries which the person is born in. The citizenship attribute in the census dataset contains five distinct categories which indicate whether the person is a US citizen by birth or by other means or not a citizen.

These two attributes have a correlation with each other, as in all of the cases within this dataset, all of the people born in the United States have an American citizenship, which is indicated in the dataset by “Native – Born in the United States”. This finding in the data is also very logical, as most people who are born in a country will have citizenship to that particular country.

2.1.6. Education and Enrolment Attributes

The next attribute to be analysed is “education”. This attribute shows a person’s education levels which is determined by the high school or institution grade. The lowest grade start at “children”, which indicates that either the person is either too young to attend school or they are in primary school. A finding from analysing the dataset is at all the people aged 0 to 14 are in the "children" category, meaning that they are either too young for education or they are in primary school. It is found that almost a quarter of the census population have a high school graduate as their highest level of education, which is shown in the diagram below (Figure 6 – Education Pie Chart).

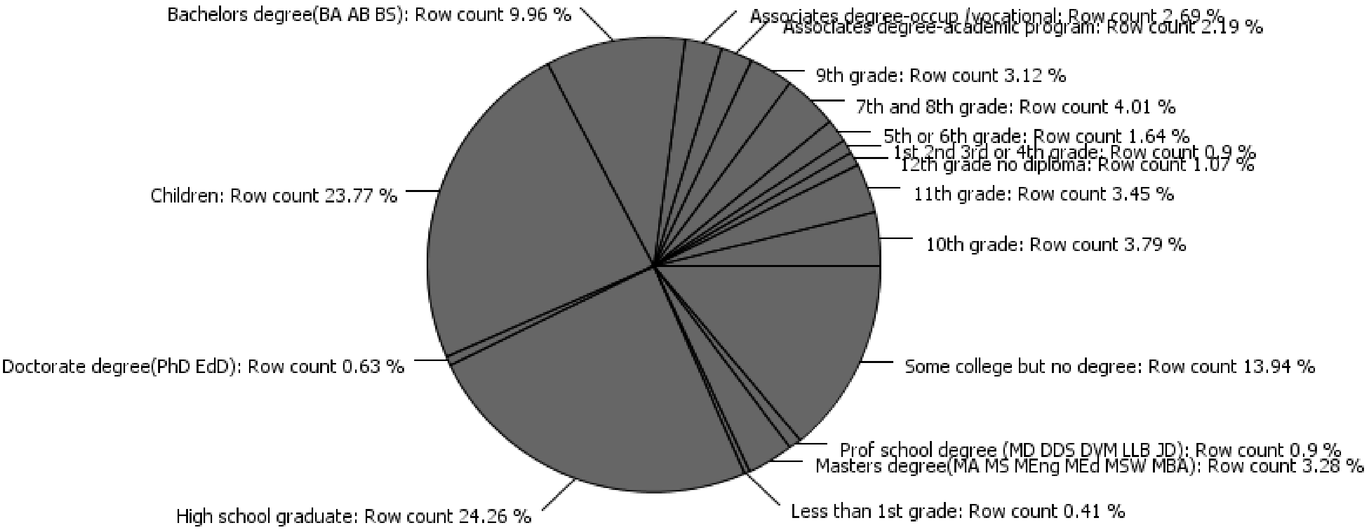


Figure 6 – Education Pie Chart

The "enrolled in edu inst last wk" attribute is closely related to the "education" attribute, as this attribute records if a person is currently enrolled in one of the education institutions within the previous week. This attribute will determine if a person is currently studying or not. Analysis of the data in this column shows that only 6.3% of the population in the census are currently attending high school, college or university. The population who are categorised as "children" in the "education" attribute are currently shown in this attribute as "not enrolled or attending an educational institute". Children who are aged 5 years and over should start to be enrolled and attending school, as they are old enough. But because of the classification which the dataset used, people who are attending primary school are labelled as "not enrolled".

2.1.7. Weeks Worked in a Year Attribute and Full or Part Time Employment Status Attributes

The "weeks worked a year" attribute indicates the number of weeks which a person from the census would work in a year. The minimum weeks worked in a year is 0, and the maximum is 52 weeks. 52 weeks worked in a year is the largest possible number, as there is only 52 weeks in a year and any number larger will be considered an error in the dataset. The minimum weeks worked in a year can only be 0, which indicated that the person has not worked within the year. The diagram below (Figure 7 – Weeks worked in a Year Binned) Histogram) shows the distribution of the amount of weeks worked. The weeks which are shown on the diagram are binned into 13 equal width bins.

Majority of the population are grouped in working for 0 to 4 weeks and 48 to 52 weeks a year. The population which are group in the 0 to 4 weeks category also include children and adolescents who attend primary school and high school.

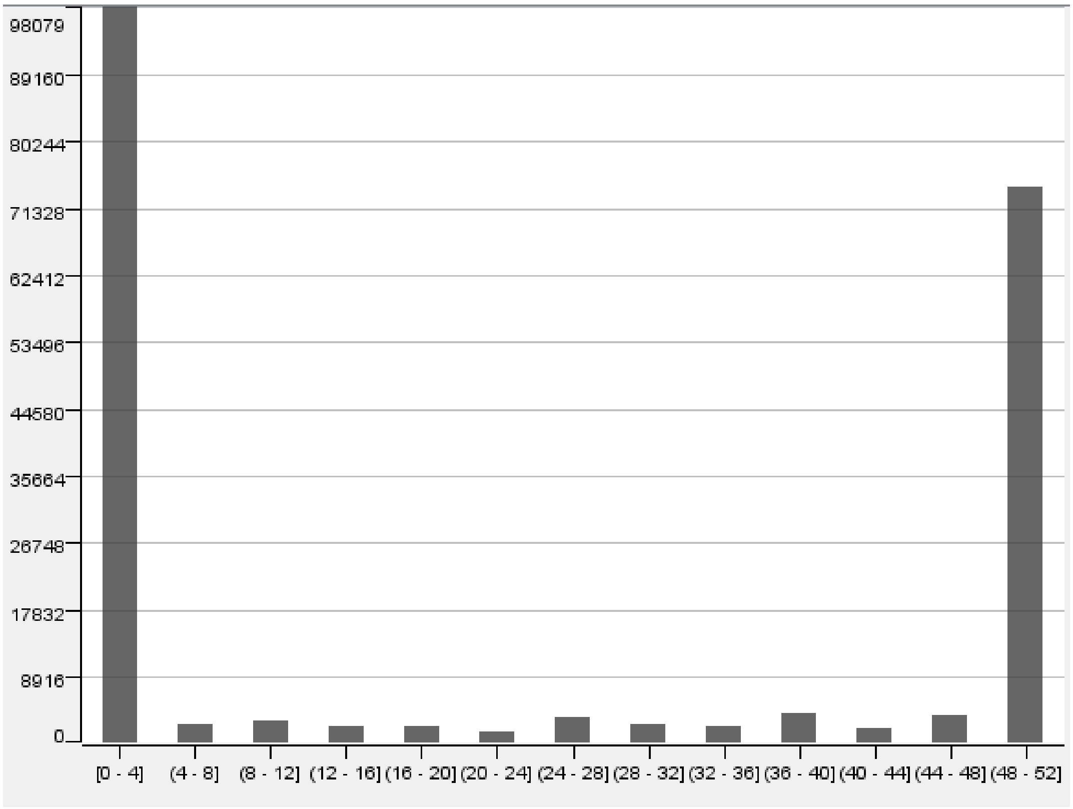


Figure 7– Weeks worked in a Year (Binned) Histogram (x-axis shows the hour bins, y-axis shows the count)

The “full time or part time employment stat” attribute contains values of the employment status of the population. This attribute contains 8 distinct different categories ranging from full time, not in labour force, children or armed forces and a range of other employment categories. It is found that 20.42% of the population are working full time, 13.44% of the population are not in the labour force, or unemployed and 62.03% of the population are either children or armed forces.

From the graph from the previous sections (Figure 3 – Age (Binned) Histogram), it is estimated that 31.5% of the population are between the ages of 0 to 20. Since the population of people from the census data contain up to 31.5% children, this means that at least half of the children or armed forces employment group are children, while the remaining half are people employed in the armed forces

2.2. Dataset Pre-Processing

After the understanding and exploration of the data phase is completed, the next phase in the CRISP-DM model is the data preparation stage, which involves with the transformation and cleaning of the data. It is clear that the data required data preparation, which includes cleaning the dataset, after the exploration of the data. One of the reasons for this is that there are many missing values throughout the dataset. Some of the remaining data are not very useful in the current state that it is in, and will be removed before the further analysis of the dataset. The removal of some data is performed manually on excel such as deleting rows. Some of the pre-processing is performed later on using the column filter node on KNIME.

After pre-processing, a brief analysis of the dataset indicates that there are 40 different usable attributes in the dataset and there are 199,523 instances of records.

3. Implementation of Classifiers

After the data exploration and pre-processing, the various classifiers which will be used to solve this data mining problem can be implemented. This is the modelling stage of the CRISP-DM, where the classifier techniques will be implemented and the parameters and settings will be adjusted in order to create the most suitable classifier (Kennedy 2015).

The KNIME Analytics Platform and WEKA data analytical packages will be utilized to solve the data analytical problem of classifying whether a person earns more or less than $50,000 a year. Classifiers which will be utilized and implemented to solve this data mining and business problem will include decision trees, random forest, naïve Bayes and support vector machines.

3.1. Decision Tree

The first classifier which will be implemented in KNIME is the decision tree. The decision tree is a supervised classifier technique which constructs a tree order to classify and predict a categorical outcome. The tree will have many nodes which branches into other nodes, with each of these nodes having a set of conditions which is to be tested. The tree will traverse through these nodes and test each of the conditions until it reaches the leaf node at the end of the decision tree. This leaf node will contain the final prediction or outcome for a specific test.

An advantage of using decision trees is the simplicity of the classifier technique and the ease of understanding of the internal operations of this classifier. Decision trees use a simple concept of having nodes with a conditions and corresponding branches which leads to more nodes, which build up the tree. The data is fed into the first node and then will traverse through the appropriate nodes in the tree and this process will continue until it reaches the leaf node. The decision tree can easily be visualised, by creating a diagram tree and the nodes. This allows people with minimal knowledge on data analytical techniques to easy understand and visualise the internal processes of a decision tree.

A disadvantage of using decision tree is that it has higher chances of overfitting the classifier, when compared to other classification techniques. If the classifier is to be overtrained, then that will lead to inaccuracies when the classifier is used to predict real data. Overtrained is when the classifier has “memorised” the training data and learnt every feature of the training data which might not be present on the actual data.

The dataset which is in a comma separated value (CSV) format is first loaded into KNIME through the CSV reader node. Some of the columns and data are removed using the column filter node. The reason for this is because during experimenting and testing the classifier, by remove irrelevant columns and only keeping with the most relevant columns, the accuracy actually increase. Some of the columns which was removed because they caused inaccuracies includes “industry” and “occupations” codes, “migration codes” and parent’s “countries of birth”.

The training data and test data is then obtained by splitting the original census dataset by partitioning 66% of the data for training and the remaining 33% is utilized for testing. Splitting the dataset is performed by the partitioning node. It is found from various testing that a 2:1 ratio spit of the dataset is the most optimal, as it allows for sufficient data to train the classifiers accurately and has enough remaining data to test the trained classifier. It also reduces the chances of overfitting or over training the classifier with the test data.

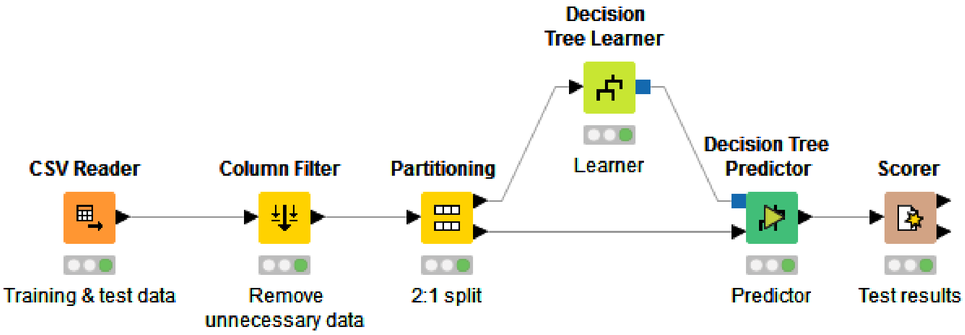
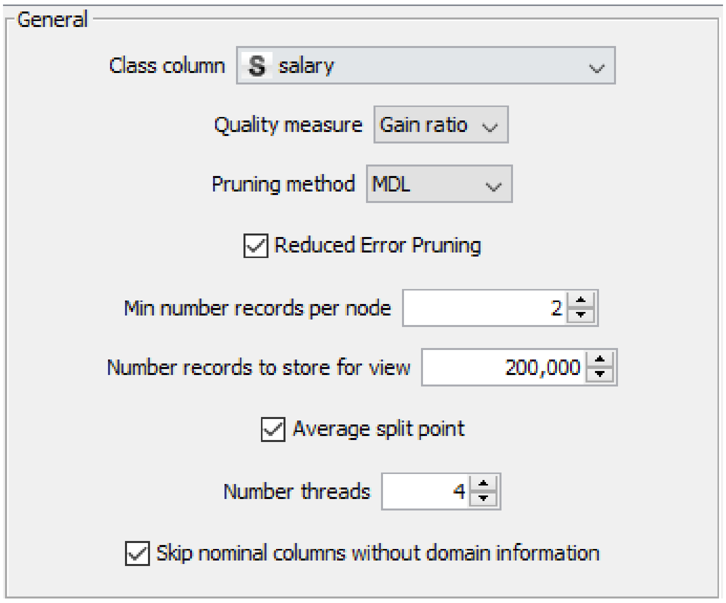


Figure 8 – Decision Tree Node Configurations

The configurations for the decision tree which was constructed on KNIME is shown in the diagram above (Figure 8 – Decision Tree Node Configurations). After partitioning the dataset, the training data is fed straight into the decision tree learner node, which then processes the data and trains the classifier. The decision tree configurations are shown on the configuration diagram below (Figure 9 – Decision Tree Learner Configuration). The pruning of the decision tree, which reduces the size of the tree, is configured to MDL method and the quality measure is set to gain ratio. After the classifier is trained, the trained classifier data is fed into the predictor node which predicts the remaining 33% of the data from the census dataset and the results of the predictions are then scored by the scorer.



General

Class column: S salary

Quality measure: Gain ratio

Pruning method: MDL

☑ Reduced Error Pruning

Min number records per node: 2

Number records to store for view: 200,000

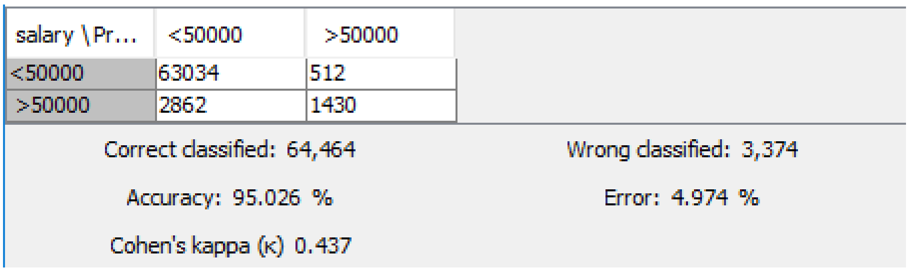
☑ Average split point

Number threads: 4

☑ Skip nominal columns without domain information

Figure 9 – Decision Tree Learner Configuration

It is found that this configuration, the decision tree classifier manages to predict with a 95% accuracy. The figure below (Figure 10 – Decision Accuracy Statistics) shows the various statistics which are used to measure the accuracy of the decision tree classifier.



salary \Pr...	<50000	>50000
<50000	63034	512
>50000	2862	1430

Correct classified: 64,464

Wrong classified: 3,374

Accuracy: 95.026 %

Error: 4.974 %

Cohen's kappa (κ) 0.437

Figure 10 – Decision Tree Accuracy Statistics

3.2. Random Forest

The random forest classifier technique will be the next data analytical method which is implemented to solve this business problem. Random forest is a supervised classifier which combines a multitude of decision trees to build up a “forest”. Each tree in the random forest classifier is similar to the decision tree which is explained in the previous section. Random attributes from the data are selected by the algorithm during the training stage which builds up large number of decision trees. These decision trees are then combined to allow the classifier to perform a categorical predict as accurately as possible.

An advantage which random forest classifier technique has over other techniques is the ability to handle large amounts of missing attributes and the ability to handle missing data. In this case, the census dataset contains up to 40 different attributes with many of the attributes containing missing values. This advantage means that less time on pre-processing and cleaning of the dataset is necessary which can save a lot of time, since cleaning of the dataset usually takes the largest amount of time. Spending less time on pre-processing means the classifier can be trained and implemented into the business to solve the business problem earlier, which results in a quicker return on investment.

One of the disadvantages is it does not handle irrelevant attributes in the dataset very well. Irrelevant attributes can cause the classifier to predict much less accurately, which means that during the implementation of the classifier, all of the unnecessary and irrelevant attributes will need to be removed.

Similar to the decision tree which is implemented previously, the census dataset was loaded through the CSV reader node and then all the irrelevant, unnecessary and missing attributes and data are removed using the column filter node. The removal of unnecessary attributes and columns are required to ensure that they do not cause the final classifier model to be less accurate and also to ensure that the classifier model does not overfit the census training data. When compared to the decision tree which is implemented previously, the random forest classifier require even more columns or attributes to be removed in order to get an equivalent prediction accuracy level. The partitioning node then splits the census dataset into 66% for training and the remaining 33% for testing. The training portion of the dataset is then fed into the random forest learner and then testing portion is fed into the random forest predictor node. After the prediction, the scorer determines the accuracy of the random forest classifier. The following configurations are shown in the diagram below (Figure 11 – Random Forest Node Configurations).

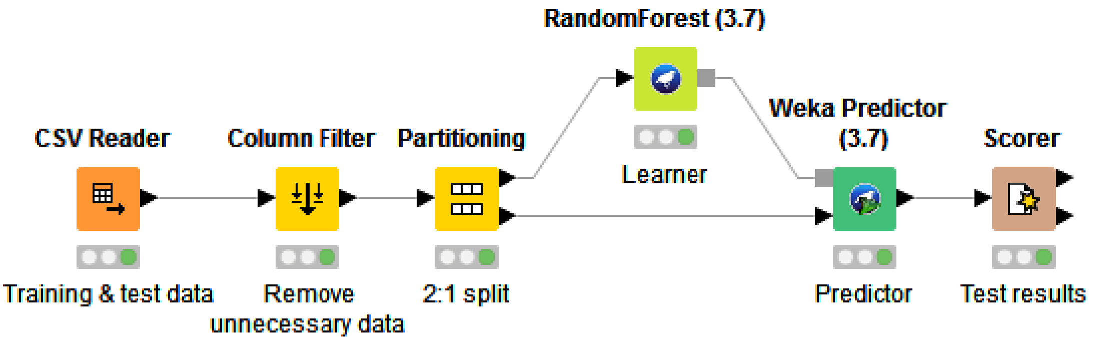
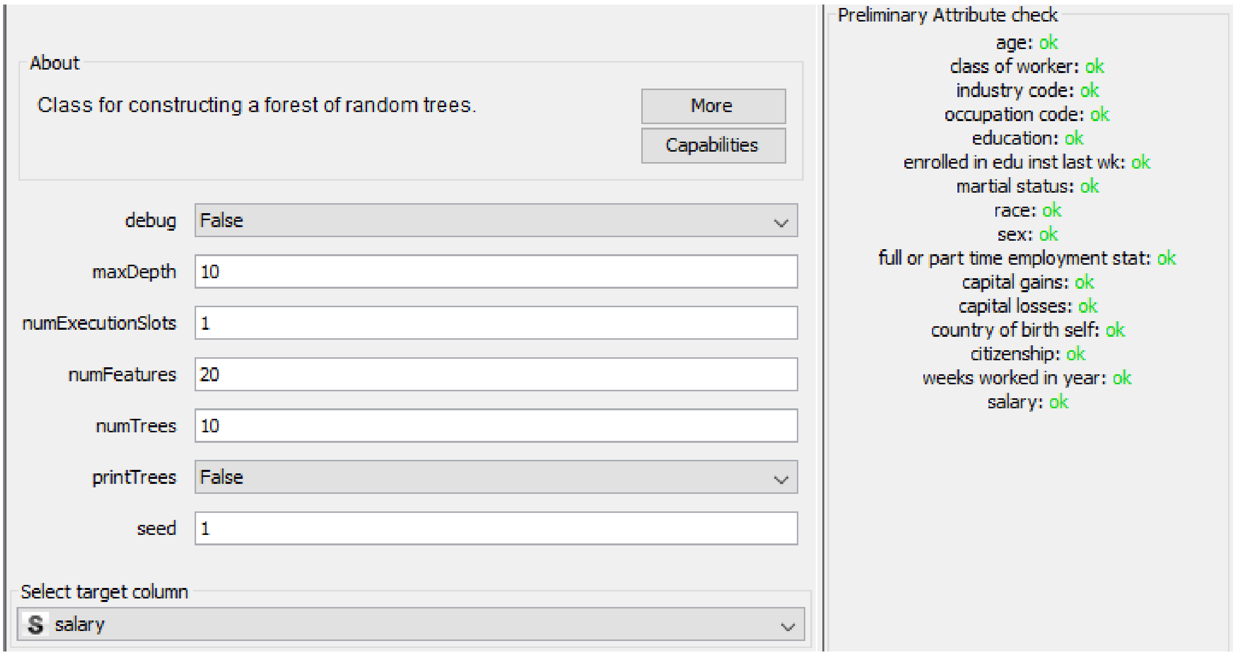


Figure 11 –Random Forest Node Configurations

The configurations which was used to have the most accurate prediction on random forest is shown on the diagram below (Figure 12 – Random Forest Learner Configuration). The max depth which is configured to be 10, is the number of levels which the nodes in the trees in the classifier will go down to. Number of features, which is configured to be 20, determines how many features in the data will be used when creating the various trees. And then number of trees determines how many decision trees are to be created in the random forest, in this case 10 trees will be generated. The configuration diagram below also lists the attributes which are used by the learner when training the random forest classifier model.



Class for constructing a forest of random trees.

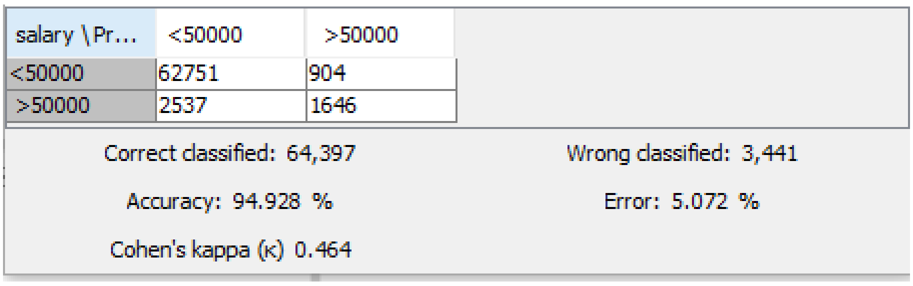
debug: False
maxDepth: 10
numExecutionSlots: 1
numFeatures: 20
numTrees: 10
printTrees: False
seed: 1

Select target column
S salary

Preliminary Attribute check
age: ok
class of worker: ok
industry code: ok
occupation code: ok
education: ok
enrolled in edu inst last wk: ok
martial status: ok
race: ok
sex: ok
full or part time employment stat: ok
capital gains: ok
capital losses: ok
country of birth self: ok
citizenship: ok
weeks worked in year: ok
salary: ok

Figure 12 –Random Forest Learner Configuration

After a range of tests with the random forest classifier, it is determined that the highest level of accuracy which this classifier can achieve is 94.9%. The rest of the results statistics is shown on the diagram below (Figure 13 – Random Forest Accuracy Statistics).



salary \ Pr...	<50000	>50000
<50000	62751	904
>50000	2537	1646

Correct classified: 64,397 Wrong classified: 3,441

Accuracy: 94.928 % Error: 5.072 %

Cohen's kappa (κ) 0.464

Figure 13 –Random Forest Accuracy Statistics

3.3. Naïve Bayes

3.3.1 KNIME Naïve Bayes Implementation

The next data analytical classifier to be implemented is Naïve Bayes. This is a supervised classifier which applies Bayes' theorem in order build a classifier model which is based on probability in order to categorize and predict values.

An advantage of naïve Bayes classifier is that it can create a working classification model very quickly and with small amount of training and testing data compared to other techniques. This can allow the classifier to be put into testing much more rapidly compared to other predictor models, which can result in a faster return on investment, from a business perspective. Being able to create a working classifier with a small amount of data is also an advantage as it allows a classifier to be created in situations where data is not available or when there is only a limit of data available.

A disadvantage of naïve Bayes is that the classifier will assume that all the features are independent, which can potentially result in a less accurate predictor model in some cases. This classifier is naïve, as it does not concern with attributes which have a strong correlation with each other, and only assumes that every attribute in the dataset is independent.

The implementation of the naïve Bayes classifier on KNIME is show on the diagram below (Figure 14 – Naïve Bayes Node Configurations). The implementation is also very similar to the previous classifier implementations. The data which is in a CSV format is first loaded into KNIME and then the column filter removes most of the columns with missing data and other unnecessary attributes. This is so the learner can focus on the more essential attributes and not overfit the classifier with the other irrelevant data. The dataset is then partition in a 2 to 1 ratio split, where the larger 2 parts of data will be used to train the classifier and the remaining 1 part will be utilized for testing the predictor model. The reason for using 66% of the data and not any more to train is to prevent the classifier model from being over fitted.

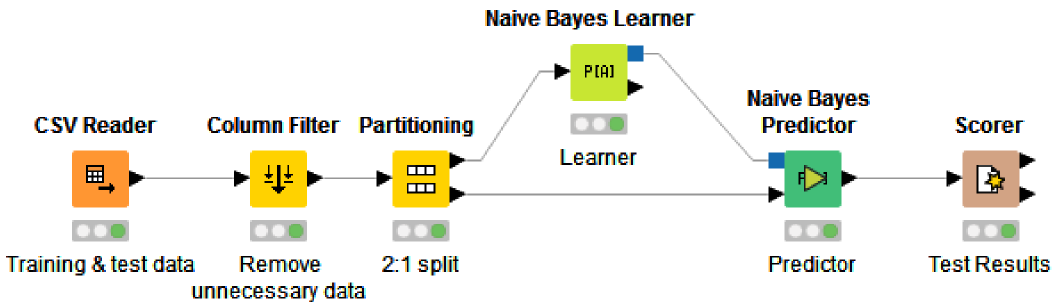
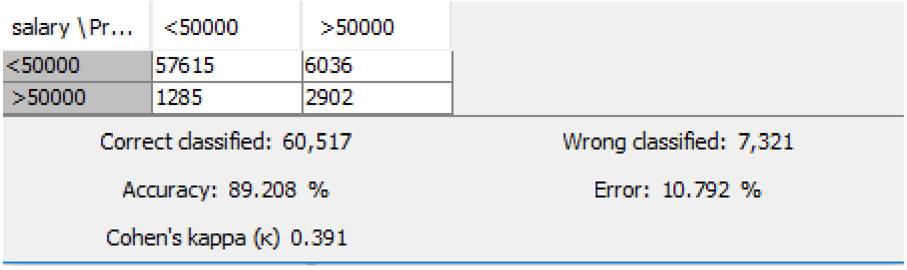


Figure 14 – Naïve Bayes Node Configurations

The naïve Bayes learner didn't require a lot of configurations in order to create a working classifier model. It also had very limited amount of configuration options and most of the available options are irrelevant in assisting to creating a more accurate prediction. Through some experimentation and testing, it is found that the KNIME naïve Bayes classifier is capable of achieving 89.2% accuracy with an error rate of just under 11%. These accuracy values and various other statistics are shown on the diagram below (Figure 15 – Naïve Bayes Accuracy Statistics).



salary \Pr...	<50000	>50000
<50000	57615	6036
>50000	1285	2902

Correct classified: 60,517

Wrong classified: 7,321

Accuracy: 89.208 %

Error: 10.792 %

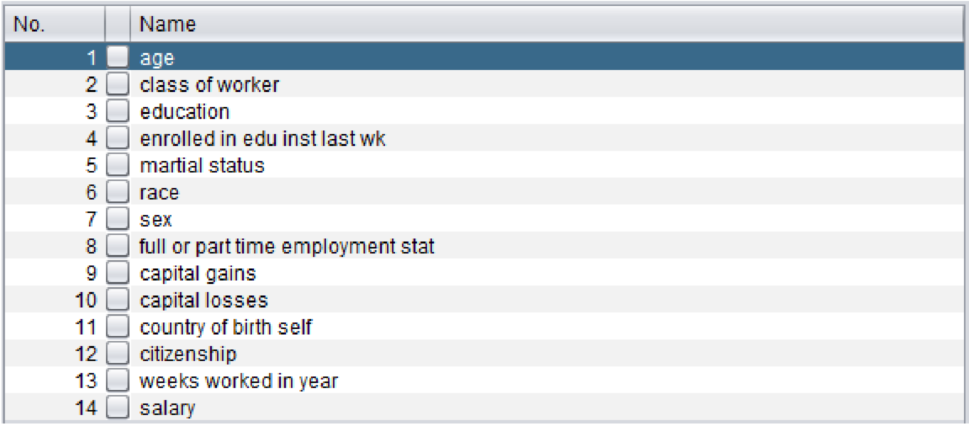
Cohen's kappa (κ) 0.391

Figure 15 –Naïve Bayes Accuracy Statistics

3.3.2 WEKA Naïve Bayes Implementation

Naïve Bayes classifier is implemented again in this project, but this time utilizing the WEKA data analytical package. This is done to allow a comparison between the two data analytical packages which are used for this project. The implementation of the naïve bays classifier here is done as similarly as possible to the KNIME implementation. All the dataset attributes used and the parameter settings are attempted to match as closely as possible. This is to reduce other variables from the comparison between the naïve Bayes classifier on KNIME and WEKA.

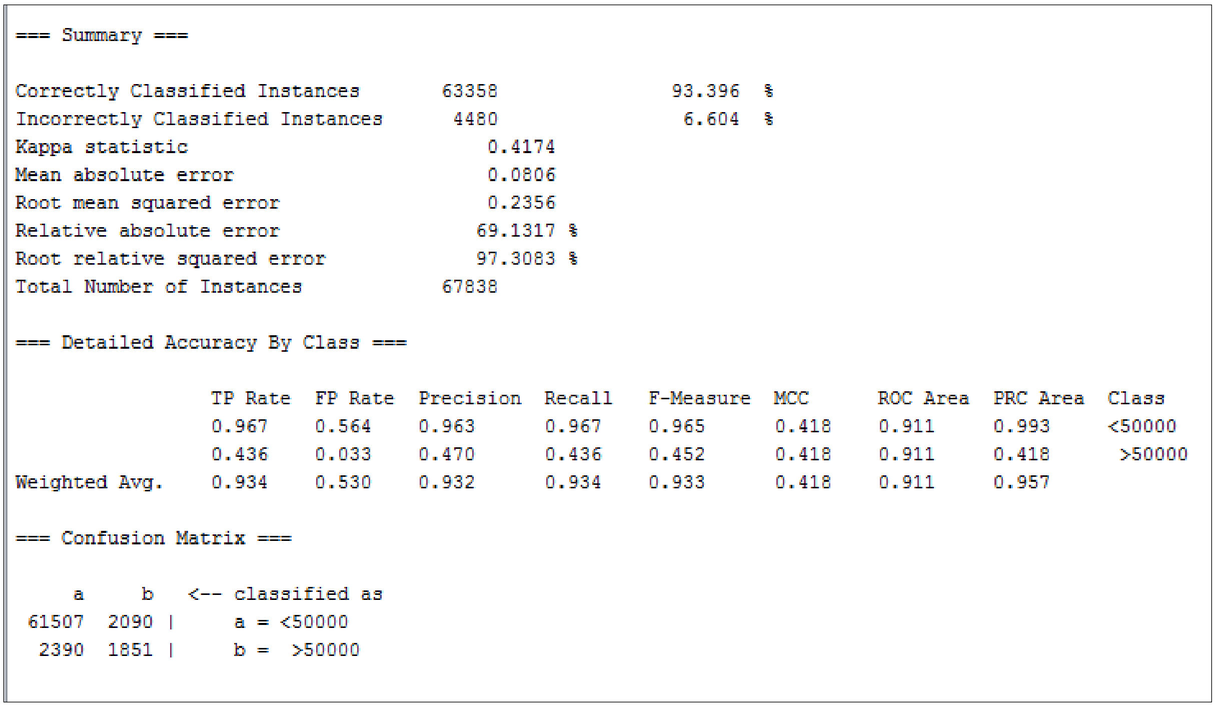
The census dataset is first loaded into WEKA, and then the unnecessary columns are removed to prevent the classifier from over fitting the data or from reducing the overall accuracy and effectiveness of the classifier model. The attributes which are not removed and will be utilized for the training are shown on the table diagram below (Figure 16 – Naïve Bayes Utilized Attributes).



No.		Name
1	□	age
2	□	class of worker
3	□	education
4	□	enrolled in edu inst last wk
5	□	martial status
6	□	race
7	□	sex
8	□	full or part time employment stat
9	□	capital gains
10	□	capital losses
11	□	country of birth self
12	□	citizenship
13	□	weeks worked in year
14	□	salary

Figure 16 – Naïve Bayes Utilized Attributes

After loading and removing some of the data, the classifier implementation can begin. The remaining data is partitioned into 66% of the data from training and the remained 33% will be used for testing the classifier model. This 2:1 ratio partition is used to reduce the chances of overfitting the classifier and also leaving enough data to effectively test the newly implemented prediction model. After testing, the WEKA naïve Bayes classifier model managed to score a 93.4% accuracy prediction level, which is noticeably better than the previous naïve Bayes classifier implemented on KNIME. The various test statistics are shown on the console output below (Figure 17 – WEKA Naïve Bayes Output & Statistics).



```
=== Summary ===

Correctly Classified Instances       63358               93.396  %
Incorrectly Classified Instances      4480                6.604  %
Kappa statistic                          0.4174
Mean absolute error                      0.0806
Root mean squared error                  0.2356
Relative absolute error                 69.1317 %
Root relative squared error             97.3083 %
Total Number of Instances            67838

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.967    0.564    0.963      0.967    0.965      0.418    0.911     0.993     <50000
                 0.436    0.033    0.470      0.436    0.452      0.418    0.911     0.418      >50000
Weighted Avg.    0.934    0.530    0.932      0.934    0.933      0.418    0.911     0.957

=== Confusion Matrix ===

     a     b   <-- classified as
 61507  2090 |     a = <50000
  2390  1851 |     b =  >50000
```

Figure 17 – WEKA Naïve Bayes Output & Statistics

3.4. Support Vector Machine

The final classifier to be implemented in this project to solve the business problem will be the support vector machine (SVM) classifier. Support vector machines are binary linear classifiers. This means that SVMs can only classify an element into one of two categorical groups, and the groups are split by a boundary or hyperplane which is created using a linear functions.

An advantage of support vector machines is the classifier is not as susceptible to being overtrained compared to other classifier methods, as the boundary requires a large amount of data to change and a small amount of data and over fitting does not cause the boundary to have significant changes.

A disadvantage of support vector machines is the limitation in speed when it comes to training and testing the classifier. Support vector machines train and test much slower compared to the other classification methods which was implemented in this report. If a large amount of data is used for learning by the support machine, a lot of computational processing power and time will need to be allocated to create an accurate and working model.

Since support vector machines cannot accept nominal values which are stored as strings, only the integers the attributes in the census dataset with numerical values can be used to train this classifier. The dataset once again is loaded through the CSV reader and then all unnecessary attributes and attributes stored in as strings, apart from salary is removed by the column filter node. Through many tests, it is found that a significant amount of time and computational resources is required to train and test the support vector machine classifier. Ultimately, compromises had to be made to the amount of training data and the number of attributes to use to allow the classifier to complete the training and testing stages in an appropriate and reasonable amount of time. The first compromise which was made is that only 5% of the training data is used for the support vector machine to train and the remaining 95% of the data will be used to test the predictor model. The second compromise which was made is only two attributes are used, which are the "weeks worked in a year" and "salary" attributes. It is found when the support vector machine classifier used the RBF kernel with a sigma parameter of 0.1, it predicted more accurately compared to the other kernels types.

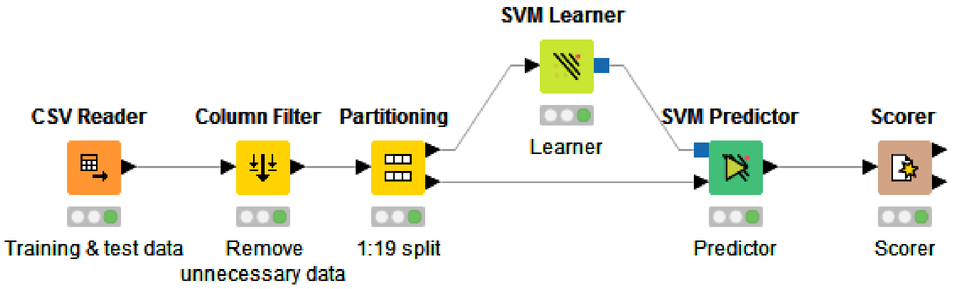
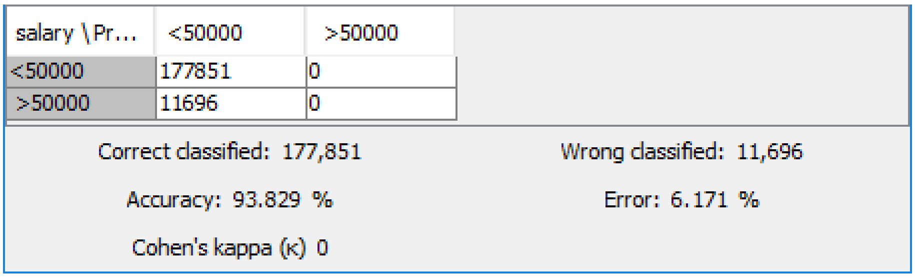


Figure 18 – Support Vector Machine Node Configurations



salary \ Pr…	<50000	>50000
<50000	177851	0
>50000	11696	0

Correct classified: 177,851 Wrong classified: 11,696

Accuracy: 93.829 % Error: 6.171 %

Cohen's kappa (κ) 0

Figure 19 – Support Vector Machine Accuracy Statistics

After testings, it was determined that the SVM model managed to score an accuracy of 93.8%, which is shown on the diagram above (Figure 19 – Support Vector Machine Accuracy Statistics). This is a very good score, considering that many extreme compromises where made. Using only 5% of the dataset and only 2 attributes for training, it managed to predict the remaining 95% of the dataset extremely accurately, with a 93.8% accuracy.

4. Classifier recommendation

The classifier recommendation section of the report will cover the evaluation and the deployment stages of the CRISP-DM model. The evaluation stage involves assessing the various classifier models which was created to the context of the business objects and the deployment stage is to present the knowledge gained back to the business (Kennedy 2015). The classifier models which are created are assessed based on the business objects and the most suitable one will be recommended.

Through this report, four different data classification methods are utilized and five classification models are implemented and tested. Each of the different methods are relatively different to the others and they all have their own advantages and disadvantages.

Classification Method	Accuracy (%)	Training & Test Data Ratio
Decision Tree	95.026	2:1
Random Forest	94.928	2:1
Naïve Bayes (KNIME)	89.208	2:1
Naïve Bayes (WEKA)	93.396	2:1
Support Vector Machine	93.829	1:19

Figure 20 – Accuracy Comparison Table

The table above (Figure 20 – Accuracy Comparison Table) shows the comparison of the accuracy between the various classifiers which was implemented. This table shows that the most accurate classifier prediction model is the decision tree and the most inaccurate model is the Naïve Bayes which was implemented on KNIME.

The table also shows the amount of data which the classifier models used as training and testing data. Most of the classifier used a 2 to 1 ratio of training and test data. This means that 66% of the census dataset was partitioned for training and the remaining was utilized for testing the classifier after the learning stages. The support vector machine classifier used only 5% of the data as training and the remaining 95% for testing. This is because of the compromise which had to me made, due to the limitations in time and computation power.

However, even with the compromises to the amount of training data which it can used, the support vector machine classifier managed to score 93.829% accuracy. This is very impressive, considering that only 2 attributes, weeks worked in a year and salary, are utilized to train the classifier and this shows the high quality of the classifier. These compromises currently hold back the full potential of the support vector machine, but in a business environment, much more time, resources and computational power will be available. This means that more training data can be used and significantly more attributes in the dataset can be used to train an even more accurate classifier. While the other classification methods are already at their limits in terms of accuracy, the support vector machine still has much more potential. Because of these reasons, the support vector machine classifier model will be the classifier which is recommended to be applied to solve the business and data mining problem in this report.

5. Conclusion

Creating a data analytical classifier which solves the data mining problem of determining whether a person earn greater or less than $50,000 per year from their census data is a very useful and important in many situations and for many businesses and government corporations. A data analytical classifier such as this can be used in a wide variety of things from track down criminals who performed tax frauds and not pay their taxes all the way to allowing businesses and large corporations to target specific sectors of the general population. Approaching this business problem by analysing the dataset in detail and then implementing various classifiers, a comparison can be done to determine the most suitable classifier to recommend for such a task. The support vector machine classifier is recommended to solve this data mining problem as this is a very high quality classifier and it still has a lot of potential, due to the compromises which had to be done for this report.

6. References

Kennedy, P. 2015, ‘Week 1: Introduction to Data Analytics Recap and Decision Trees’, *UTS Online Subject 31005*, lecture notes, UTS, Sydney, viewed 15 September 2016, <https://online.uts.edu.au/bbcswebdav/pid-990843-dt-content-rid-5499008_1/courses/31005M01/week1.pdf>.

UCI Machine Learning Repository 2013, *Census-Income (KDD) Data Set*, California, viewed 13 September 2016, <http://archive.ics.uci.edu/ml/datasets/Census-Income+%28KDD%29>.

United States Census Bureau 2016, *United States Quick Facts from the US Census Bureau*, Maryland, viewed 21 September 2016, <https://www.census.gov/quickfacts/table/PST045215/00>.