CITS3401 Final Project Submission

Pattern Discovery and Predictive Analytics

Gregory Edmonds 21487148

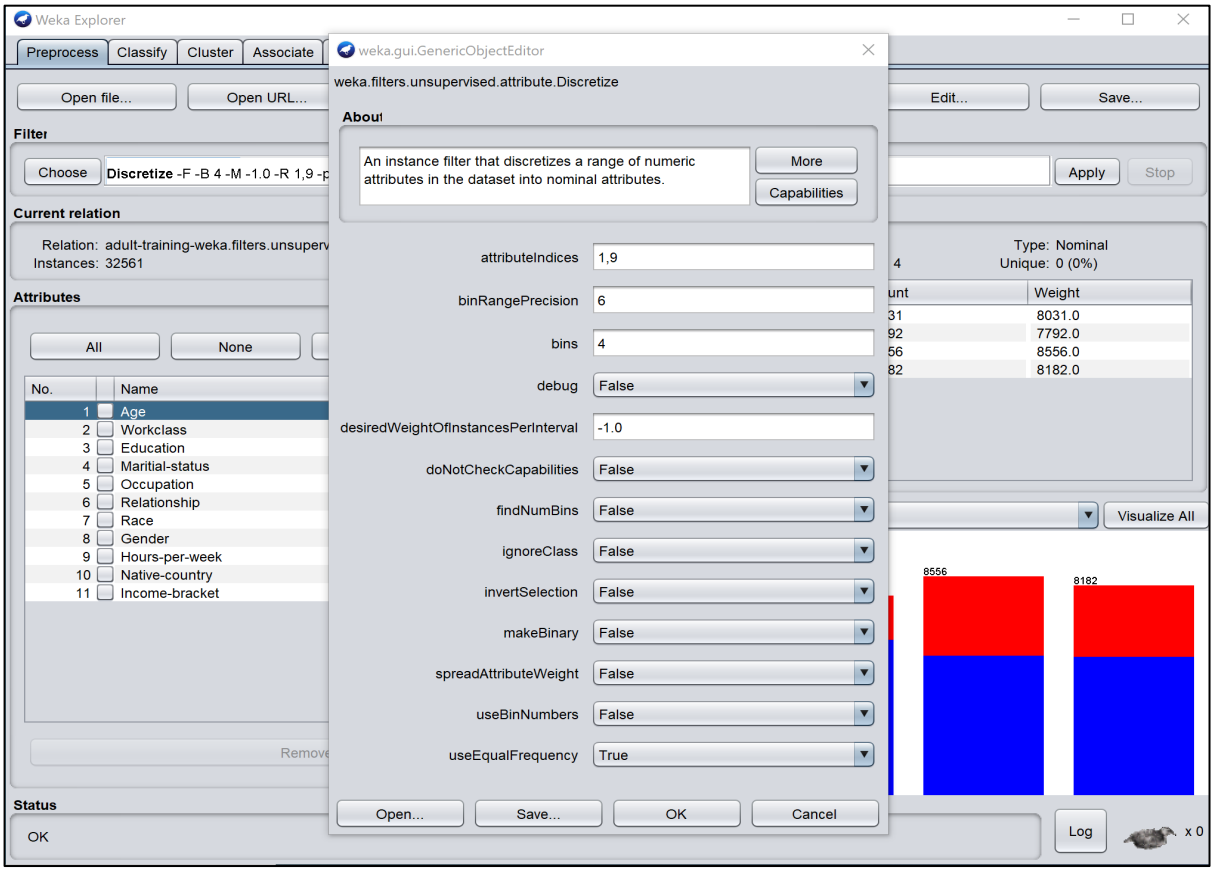
Association Rule Mining

Choosing Attributes

When deciding which attributes to mine, I started with the original dataset and removed the attributes Fnlwgt, Education-num, Capital-gain and Capital-loss. I removed Fnlwgt because it is a count of people that are predicted to belong to each row, but I believe the spreadsheet has enough data to analyse without this. I removed Education-num (the highest level of education achieved in numerical form) because I also believe that Education contains enough information to be interesting by itself. I also removed Capital-gain and Capital-loss as there were to many instances which had no value for either of these attributes. Keeping these would make the data more sparse and it could also increase the processing load of the dataset.

Pre-processing in Weka

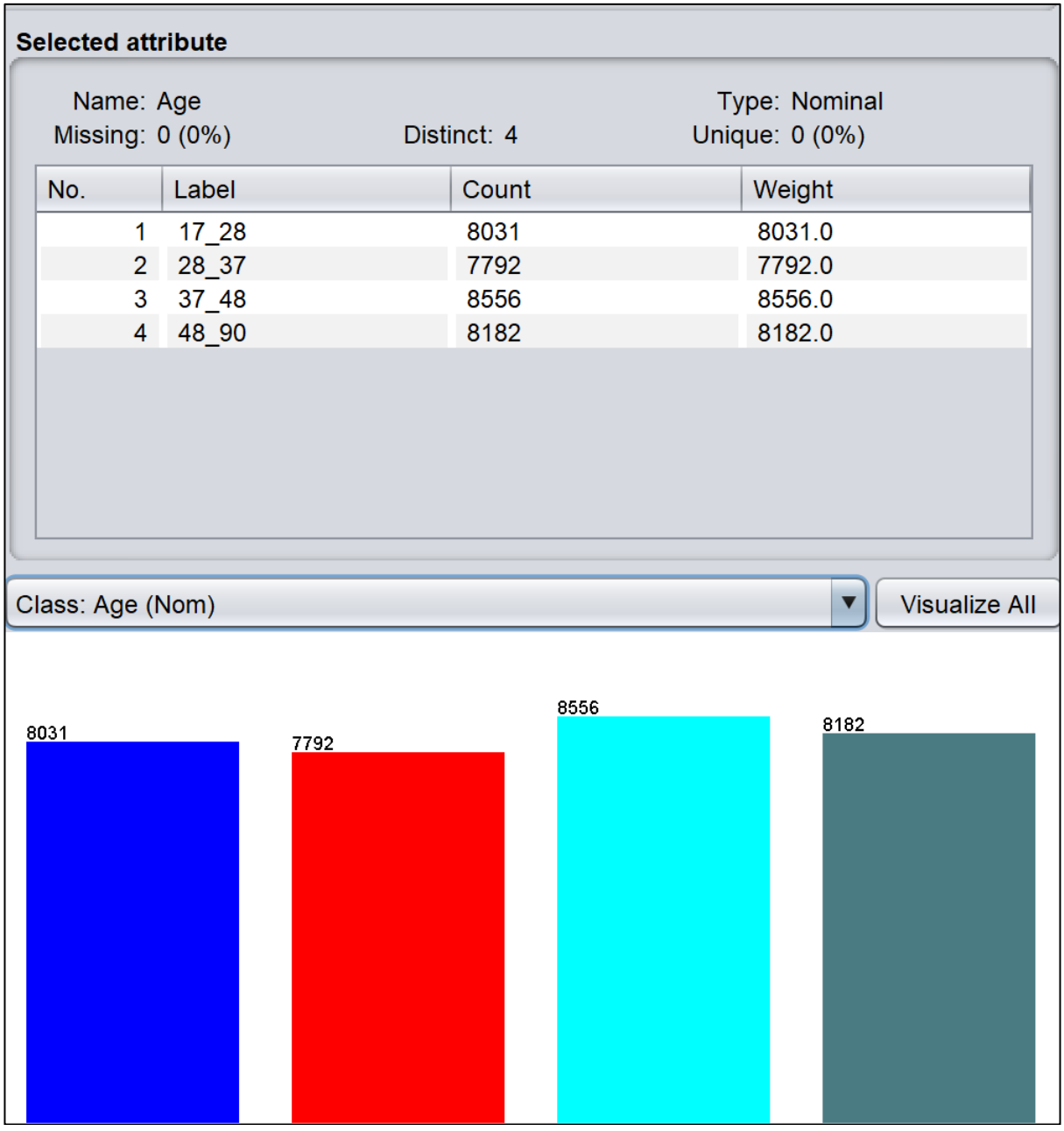
To be able to mine association rules, the attributes Age and Hours-per-week had to be discretized to nominal attributes. After experimenting with different options in Weka, I found that discretizing both attributes using 4 bins and equal frequency allowed them to be sorted into groups that provided a good output for both attributes.



Parameters chosen for discretize filter in Weka

When doing this, Age was sorted into groups of 17 to 28, 28 to 37, 37 to 48 and 48 to 90. I believe these groups to be relevant because:

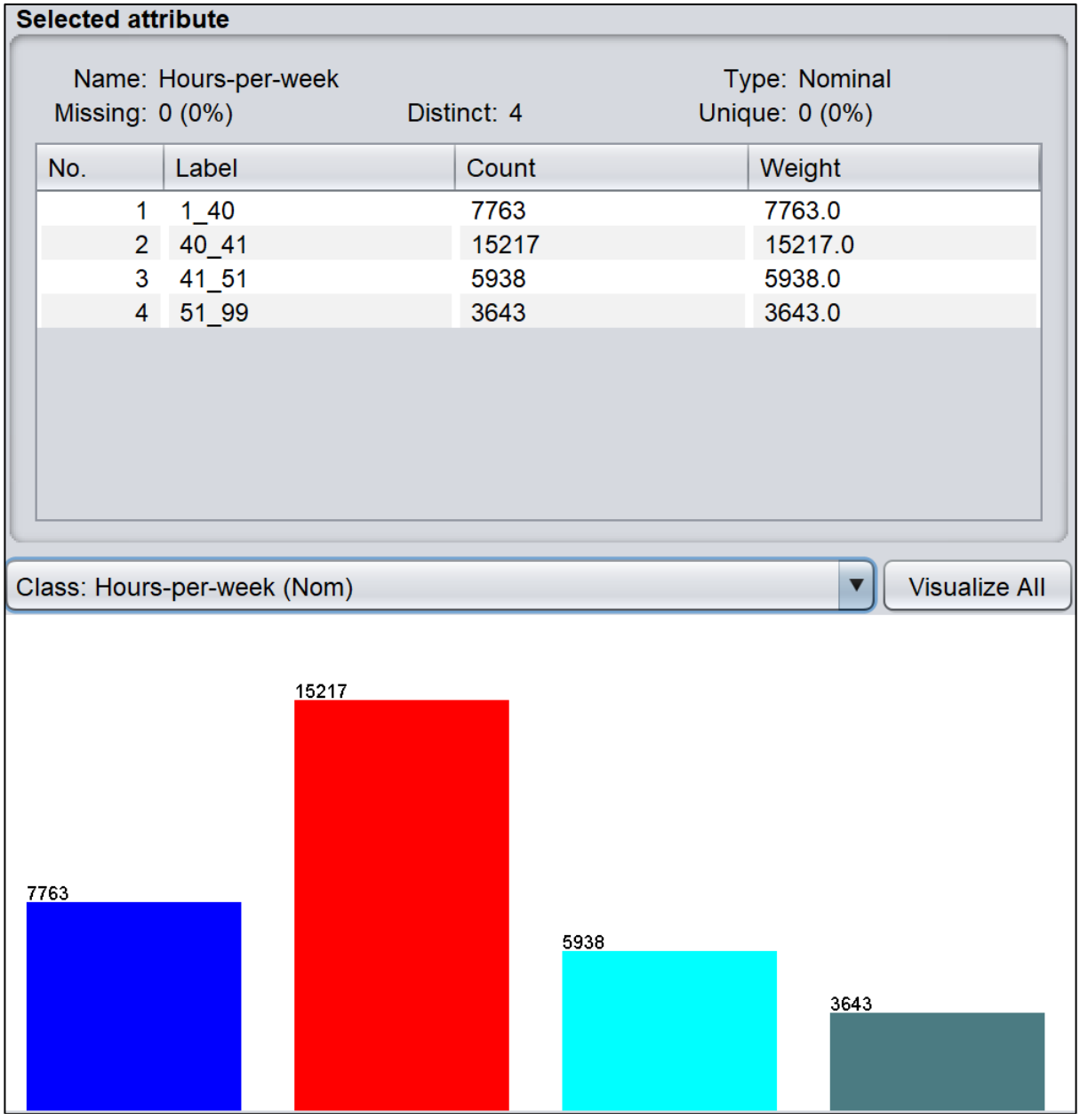
- The age group 28 to 37 is the age when most people would have settled in a career for the rest of their lives, meaning that their current occupation would be a good indicator of their financial situation.
- The age group 37 to 48 is when I believe most people would be earning the greatest income, especially for people doing more physical work, as their income could possibly decrease with age after this time.
- The equal frequency option allowed a relatively equal number of instances in each group.



Age attribute after discretzation in Weka

Similarly, Hours-per-week was sorted into the groups 1 to 40, 40 to 41, 41 to 51 and 51 to 99. This was done because:

- The purpose of analysing the dataset is to predict which factors can earn a salary of over 50k, so it follows that people who work more will generally have a higher income, so more attention is given to those who work at least 40 hours per week.
- The data is heavily populated with those who work exactly 40 hours, so having a group to contain only these people (40 to 41) is ideal.



Hours-per-week attribute after discretzation in Weka

As shown above, the labels for each group and all their associated instances in the data were also renamed using a text editor to be more readable. I now used this data set as my original, discretized data for future use.

Mining Association Rules

Using the discretized data to mine association rules, I found some problems with the rules initially generated. Firstly, because the data was heavily populated with people who earned less than 50k, the strongest rules generated were all for this group of people. Also, the data was heavily populated with certain groups for other attributes. For example, the number of those with United States as their native country heavily outweighed those with other countries. This makes sense for the dataset but results in some of the top rules having only United States as the native country where native country is present. This bias is also notably present in Race => White and Workclass => Private among a few others. Below are the rules generated using a confidence of 0.5 as the metric, with the minimum support adjusted by Weka to be 0.4 (lowerBoundMinSupport set to 0.1).

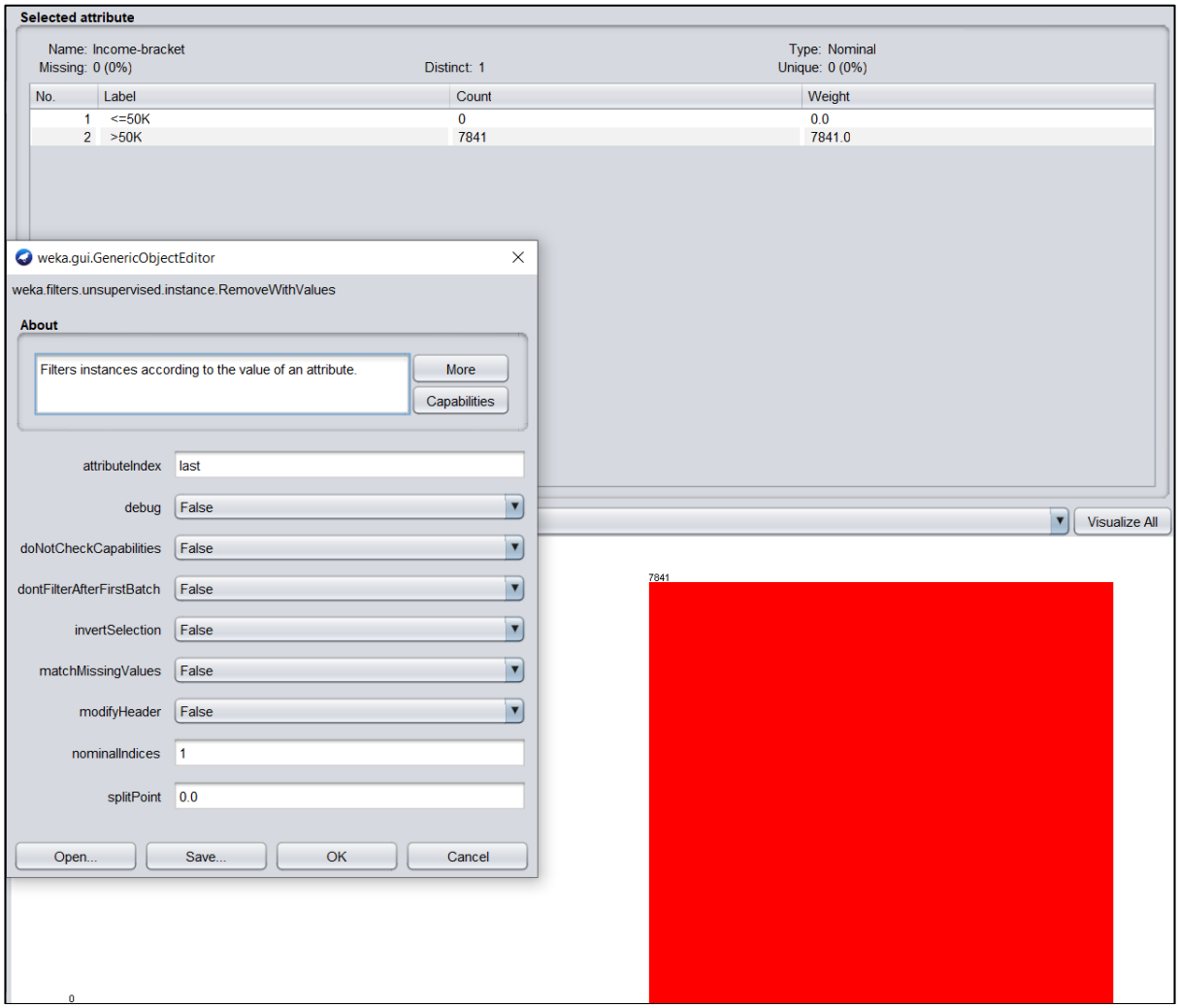
Associator output

```
=== Run information ===

Scheme:       weka.associations.Apriori -N 10 -T 0 -C 0.5 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -A -c -1
Relation:     adult-training-weka.filters.unsupervised.attribute.Discretize-F-B4-M-1.0-R1,9-precision6
Instances:    32561
Attributes:   11
              Age
              Workclass
              Education
              Maritial-status
              Occupation
              Relationship
              Race
              Gender
              Hours-per-week
              Native-country
              Income-bracket
=== Associator model (full training set) ===


Apriori
=======

Minimum support: 0.4 (13024 instances)
Minimum metric <confidence>: 0.5
Number of cycles performed: 12

Generated sets of large itemsets:

Size of set of large itemsets L(1): 4

Size of set of large itemsets L(2): 5

Size of set of large itemsets L(3): 1

Best rules found:

 1. Workclass= Private 22696 ==> Income-bracket= <=50K 17733    conf:(0.78)
 2. Workclass= Private Native-country= United-States 20135 ==> Income-bracket= <=50K 15594    conf:(0.77)
 3. Workclass= Private Race= White 19404 ==> Income-bracket= <=50K 14872    conf:(0.77)
 4. Workclass= Private Race= White Native-country= United-States 17728 ==> Income-bracket= <=50K 13452    conf:(0.76)
 5. Native-country= United-States 29170 ==> Income-bracket= <=50K 21999    conf:(0.75)
 6. Race= White 27816 ==> Income-bracket= <=50K 20699    conf:(0.74)
 7. Race= White Native-country= United-States 25621 ==> Income-bracket= <=50K 18917    conf:(0.74)
 8. Gender= Male 21790 ==> Income-bracket= <=50K 15128    conf:(0.69)
 9. Gender= Male Native-country= United-States 19488 ==> Income-bracket= <=50K 13389    conf:(0.69)
10. Race= White Gender= Male 19174 ==> Income-bracket= <=50K 13085    conf:(0.68)
```

Initial association rules in Weka using confidence

I found that further data reduction was needed to improve the results and make it more consistent with the goal: providing a recommendation for a person who wants to improve their income. For this, the attributes would have to be those that a person could achieve or those in relation to their lifestyle, rather than something that is hereditary. Because of this, I further reduced the data to remove Age, Native-country, Race and Gender, and also removed all instances of those who earned less than 50k to keep it in line with the goal.

To do this in Weka, I simply selected the four attributes and clicked the 'Remove' button. To remove the instances for income less than 50k, I used the RemoveWithValues filter which left only the instances of those earning over 50k in the data.



Filter to remove instances with specific values (<=50k) and its corresponding output in Weka

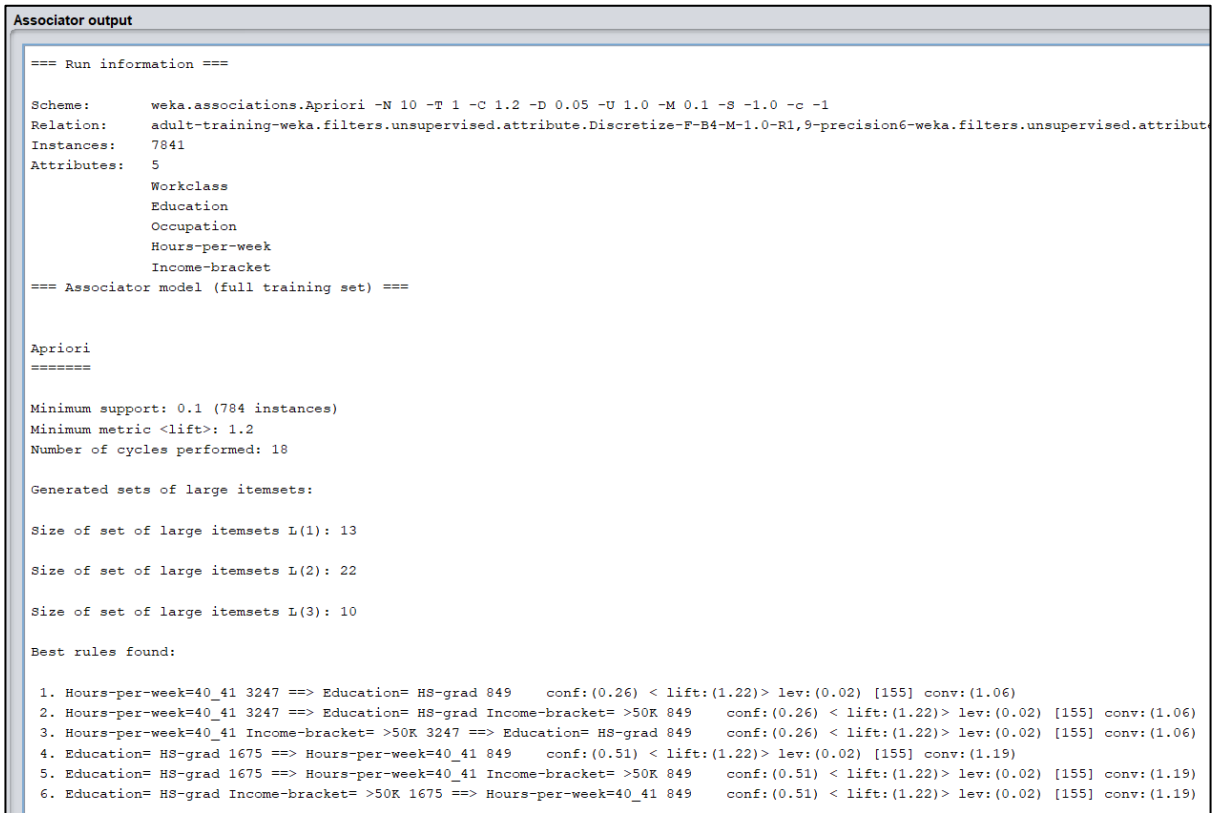
After this reduction, the generated rules using the same properties as before (with the minimum support now adjusted by Weka to be 0.3) were now more relevant.

Associator output

```
=== Run information ===

Scheme:       weka.associations.Apriori -N 10 -T 0 -C 0.5 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -A -c -1
Relation:     adult-training-weka.filters.unsupervised.attribute.Discretize-F-B4-M-1.0-R1,9-precision6-weka.filters.unsupervise
Instances:    7841
Attributes:   7
              Workclass
              Education
              Maritial-status
              Occupation
              Relationship
              Hours-per-week
              Income-bracket
=== Associator model (full training set) ===


Apriori
=======

Minimum support: 0.3 (2352 instances)
Minimum metric <confidence>: 0.5
Number of cycles performed: 14

Generated sets of large itemsets:

Size of set of large itemsets L(1): 4

Size of set of large itemsets L(2): 5

Size of set of large itemsets L(3): 2

Best rules found:

 1. Maritial-status= Married-civ-spouse 6692 ==> Income-bracket= >50K 6692    conf:(1)
 2. Relationship= Husband 5918 ==> Income-bracket= >50K 5918    conf:(1)
 3. Maritial-status= Married-civ-spouse Relationship= Husband 5914 ==> Income-bracket= >50K 5914    conf:(1)
 4. Workclass= Private 4963 ==> Income-bracket= >50K 4963    conf:(1)
 5. Workclass= Private Maritial-status= Married-civ-spouse 4220 ==> Income-bracket= >50K 4220    conf:(1)
 6. Workclass= Private Relationship= Husband 3753 ==> Income-bracket= >50K 3753    conf:(1)
 7. Workclass= Private Maritial-status= Married-civ-spouse Relationship= Husband 3751 ==> Income-bracket= >50K 3751    conf:(1)
 8. Hours-per-week=40_41 3247 ==> Income-bracket= >50K 3247    conf:(1)
 9. Maritial-status= Married-civ-spouse Hours-per-week=40_41 2852 ==> Income-bracket= >50K 2852    conf:(1)
10. Relationship= Husband Hours-per-week=40_41 2504 ==> Income-bracket= >50K 2504    conf:(1)
```

Updated association rules in Weka using confidence and reduced data

Because confidence alone is not a good measure to discover interesting association rules, I also used the lift measure to obtain more rules. In doing this, I found most of the lift association to be between Relationship and Marital-status, in the sense that a Relationship of Husband is highly correlated with a Marital-status of Married-civ-spouse. This, of course, is trivial and explains nothing new or interesting about the data. Despite this, the association rules using confidence alone show a correlation between Marital status/Relationship and high income, so I didn't want to disregard these attributes entirely. I decided to remove both Martial-status and Relationship from the data for the lift associations only and see if there were any other patterns. This resulted in rules centred around Hours-per-week and Education.



Association rules in Weka using lift

The updated lift output only provided four rules relating to an income greater than 50k out of six in total. This was done with a fairly low lift threshold of 1.2. Of these four, only two displayed income on the right side (a correlated occurrence of the left side). These were rules 2 and 5, which suggest that working a full 40-hour work week and having an education of at least a high school graduate have a positive effect on income. From these and from the confidence rules, I now had four distinct rules that related to having an income over 50k. Since I couldn't mine any more rules using lift and a minimum support of 0.1, I went back to the confidence method and set the lowerBoundMinSupport to 0.1 to find new and distinct rules (using a Weka adjusted minimum support of 0.15). From doing this, a few new associated attributes where presented, with the highest rated being rule number 4 of having a bachelor's education.

```
Best rules found:

 1. Workclass= Private 4963 ==> Income-bracket= >50K 4963    conf:(1)
 2. Hours-per-week=40_41 3247 ==> Income-bracket= >50K 3247    conf:(1)
 3. Hours-per-week=41_51 2352 ==> Income-bracket= >50K 2352    conf:(1)
 4. Education= Bachelors 2221 ==> Income-bracket= >50K 2221    conf:(1)
 5. Workclass= Private Hours-per-week=40_41 2129 ==> Income-bracket= >50K 2129    conf:(1)
 6. Occupation= Exec-managerial 1968 ==> Income-bracket= >50K 1968    conf:(1)
 7. Occupation= Prof-specialty 1859 ==> Income-bracket= >50K 1859    conf:(1)
 8. Education= HS-grad 1675 ==> Income-bracket= >50K 1675    conf:(1)
 9. Workclass= Private Hours-per-week=41_51 1624 ==> Income-bracket= >50K 1624    conf:(1)
10. Hours-per-week=51_99 1504 ==> Income-bracket= >50K 1504    conf:(1)
```

Association rules in Weka using confidence and a minimum support of 0.15

Now, using a mix of rules from both confidence and lift methods, I settled on 5 rules:

1. Marital-status= Married-civ-spouse 6692 ==> Income-bracket= >50K 6692 conf:(1)
2. Workclass= Private 4963 ==> Income-bracket= >50K 4963 conf:(1)
3. Hours-per-week=40_41 3247 ==> Income-bracket= >50K 3247 conf:(1)
4. Education= HS-grad 1675 ==> Hours-per-week=40_41 Income-bracket= >50K 849 conf:(0.51) < lift:(1.22)> lev:(0.02) [155] conv:(1.19)
5. Education= Bachelors 2221 ==> Income-bracket= >50K 2221 conf:(1)

Recommendation

Although a relationship status of Husband was one of the strongest correlations I obtained when using the reduced data, I specifically chose the five rules above so that they could be achieved by anyone regardless of their gender or any other hereditary attributes. These five attributes were able to be achieved by anyone over the course of their life, making the recommendations more suitable for a wider audience.

Overall, based on these association rule mining methods, to improve your income you should:

- Be in a married relationship
- Work in the private industry
- Work around 40 hours per week
- Have an education of at least high school, preferably at least bachelors

Practically, these recommendations carry over well into the real world. For example, if a person is married, they are more often in a good financial position than not due to having a spouse to support and also support them. They are also more likely to have children then those who have never been married, providing further responsibility for themselves. Also, for a given occupation, working in the private sector generally means you have a higher income than if you are working in the public sector. In the same way, working 40 hours per week means you likely have full-time employment, and working more hours per week unsurprisingly results in more money earned. Also, having a bachelors or high school education indicates that a person has some level of aptitude and likely comes from a household that values education and people with a higher education tend to obtain higher paying jobs in general.

Classification

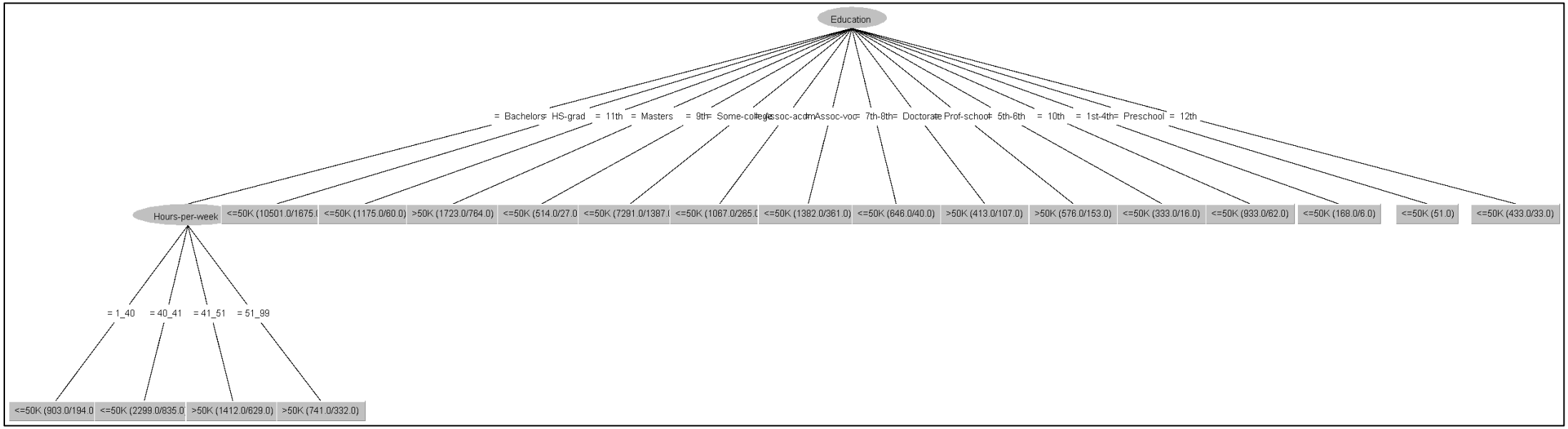
Decision Tree (1) – Selected Attributes

For the decision trees, I decided that each tree would have four attributes each to keep the trees uniform and concise. For the decision tree to be based on my understanding, I chose the attributes Workclass, Education, Occupation and Hours-per-week. These are the same attributes I selected when mining association rules using lift. I chose them because they had already been reduced to remove factors that were hereditary and remove the Relationship: Husband and Marital-status: Married-civ-spouse bias. These attributes were also present in my final top five association rules, so based on my understanding of how the rules were generated, I believed they would produce relevant and interesting decision trees also.

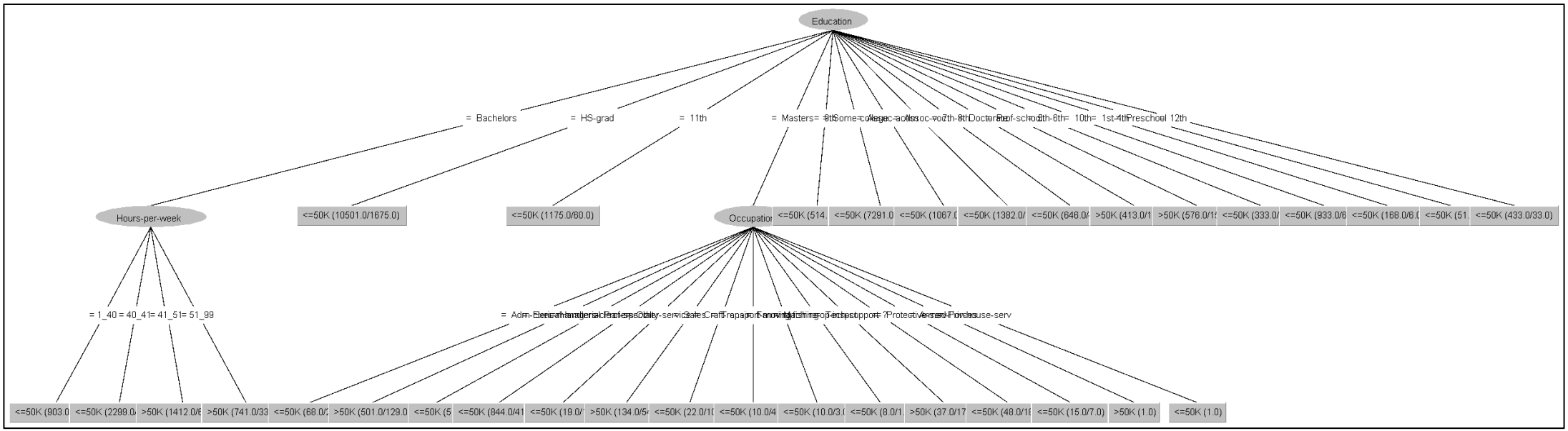
When creating the tree, I realised there were too many branches to provide a clear interpretation, so I experimented with changing the minimum number of instances per leaf. I found that using values of 500 and 1000 allowed for a better looking tree. The detailed output for both of these trees were the same.

```
Time taken to build model: 0.13 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       25547               78.4589 %
Incorrectly Classified Instances      7014               21.5411 %
Kappa statistic                          0.3187
Mean absolute error                      0.3113
Root mean squared error                  0.395
Relative absolute error                 85.1331 %
Root relative squared error             92.3749 %
Total Number of Instances            32561

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.921    0.645    0.818      0.921    0.867      0.334    0.717     0.872      <=50K
                 0.355    0.079    0.587      0.355    0.442      0.334    0.717     0.473      >50K
Weighted Avg.    0.785    0.509    0.763      0.785    0.764      0.334    0.717     0.776

=== Confusion Matrix ===

     a     b   <-- classified as
 22767  1953 |     a =  <=50K
  5061  2780 |     b =  >50K
```

Classifier output using J48 for tree (1) in Weka



Tree view for tree (1) in Weka using minimum number of instances per leaf = 1000



Tree view for tree (1) in Weka using minimum number of instances per leaf = 500

Decision Tree (2) – Information Gain Ranked Attributes

For the second tree, the attributes were chosen using the original, discretized data with information gain ranking, which ranks the attributes which have more unique values (more possible answers per attribute) higher. A larger information gain suggests a lower entropy group or groups of samples, and hence less surprise. This was done in Weka in the 'Select attributes' tab by using InfoGainAttributeEval as the Attribute Evaluator. The top 4 selected attributes outputted were Relationship, Marital-status, Education and Occupation.

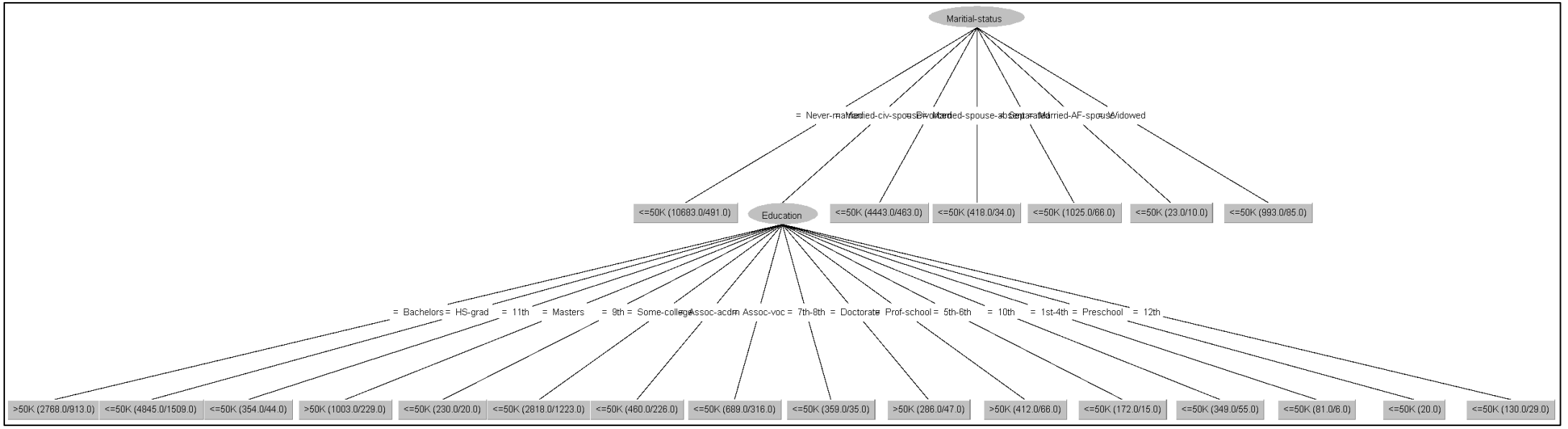
```
Ranked attributes:
 0.16537    6 Relationship
 0.15653    4 Maritial-status
 0.09359    3 Education
 0.09292    5 Occupation
 0.08379    1 Age
 0.05277    9 Hours-per-week
 0.03717    8 Gender
 0.02157    2 Workclass
 0.0087    10 Native-country
 0.00838    7 Race
```

Ranked attributes for decision tree (2) in Weka

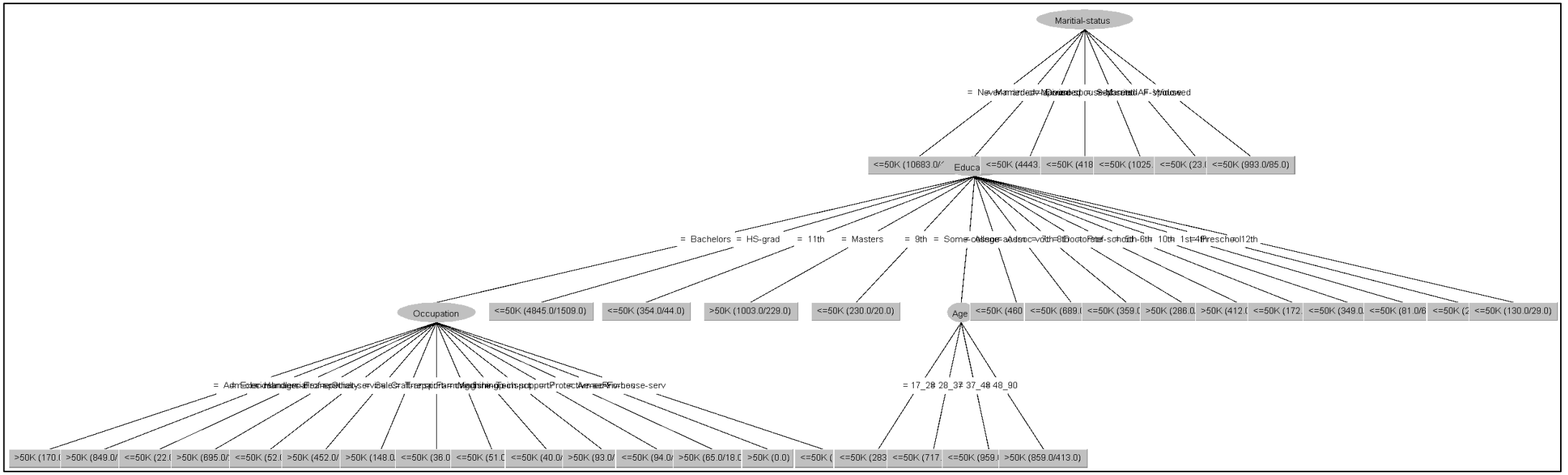
Comparing this to my chosen values for decision tree (1), the tree attributes both contain Education and Occupation, but tree (2) has Relationship and Martial-status instead of Workclass and Hours-per-week. The tree for this and its corresponding details were created in the same way as tree (1), with the same minimum number of instances per leaf so that they could be easily compared.

```
Time taken to build model: 0.14 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances       26670               81.9078 %
Incorrectly Classified Instances      5891               18.0922 %
Kappa statistic                          0.4213
Mean absolute error                      0.2543
Root mean squared error                  0.3567
Relative absolute error                 69.5349 %
Root relative squared error             83.4175 %
Total Number of Instances            32561

=== Detailed Accuracy By Class ===

                 TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                 0.948     0.588     0.836      0.948    0.888      0.446    0.832     0.924     <=50K
                 0.412     0.052     0.716      0.412    0.523      0.446    0.832     0.618     >50K
Weighted Avg.    0.819     0.459     0.807      0.819    0.800      0.446    0.832     0.850

=== Confusion Matrix ===

     a     b   <-- classified as
 23439  1281 |     a =  <=50K
  4610  3231 |     b =  >50K
```

Classifier output using J48 for tree (2) in Weka



Tree view for tree (2) in Weka using minimum number of instances per leaf = 1000



Tree view for tree (2) in Weka using minimum number of instances per leaf = 500

Comparing the Trees

Looking at the tree (1), we can first look at the version having a minimum number of instances per leaf set to 1000 to see patterns that are more present in the data. The diagram suggests that education is a major contributor to whether someone's income is less than or greater than 50k. Having a master's degree, doctorate, or professional schooling all indicate that a person earns over 50k. In the case of having a bachelor's degree, those who work 40 or more hours per week are able to earn over 50k while those that work less hours tend to earn less than 50k per year. Looking more closely at the tree (with minimum number of instances set to 500) we see similar results, but this time the masters degree attribute is broken down further into occupation, suggesting that those who work in the fields exec-managerial, sales, tech support and armed forces are able to earn more than 50k while the other occupations tend to earn less than 50k. Overall, tree (1) suggests patterns that I believe can easily relate to the real world. It focuses on Education, which can be a good indicator of a person's income, and typically, the higher a person's level of education the more they can potentially earn. For the lowest type of education out of all types earning over 50k i.e. bachelor's degree, it makes sense that the number of hours work per week is more crucial to their income for the same reason.

Looking at tree (2), the results in terms of education and occupation are very similar, however they are now bounded by the martial-status attribute one level above in the tree. It essentially shows that people who are currently married have a greater chance of earning over 50k, given that they have an education of at least bachelor's degree as well. This is likely highly skewed by the database having many more instances of those who are married than those that are otherwise. Because of this large bias it is harder to say whether this tree will reflect real life, but as stated in the association rule mining section, being in a married relationship can have a positive effect on how much one can earn.

In terms of the confusion matrix for both trees, for tree (1), the precision is 81.8%, which is slightly less than the 83.6% precision of tree (2). This means that tree (2) contains more true positives and can be said to be more accurate in that regard. The recall for tree (2) is also slightly higher at 94.8% compared to 92.1% for tree (1). Since recall can be thought of as a model's ability to find all the data points of interest in a dataset, tree (2) is also better in this regard. Both trees also contain an F-measure of 86.7% and 88.8% respectively, showing that both models are somewhat accurate in terms of the data provided. Although tree (2) is seen to be slightly more accurate across measures, I believe that tree (1) suits the analysis better, as it uses the same data as the lift association rule, which was heavily reduced to lessen bias in the data.

Clustering

K-Means Clustering

For clustering, I decided to use simple k-means clustering with Euclidean distance on the original, discretized data since this data is already nominal and contains the most attributes, so it could potentially show interesting, hidden patterns between attributes. I believe that clustering would be better suited to finding interesting patterns throughout all the data, instead of only seeing which attributes would correlate to a higher income. This is because clustering is able to display traits of those earning over 50k alongside traits of those earning less than 50k in the same view. After experimenting with different numbers of clusters, I decided to use only two to reduce the amount of incorrectly clustered instances.

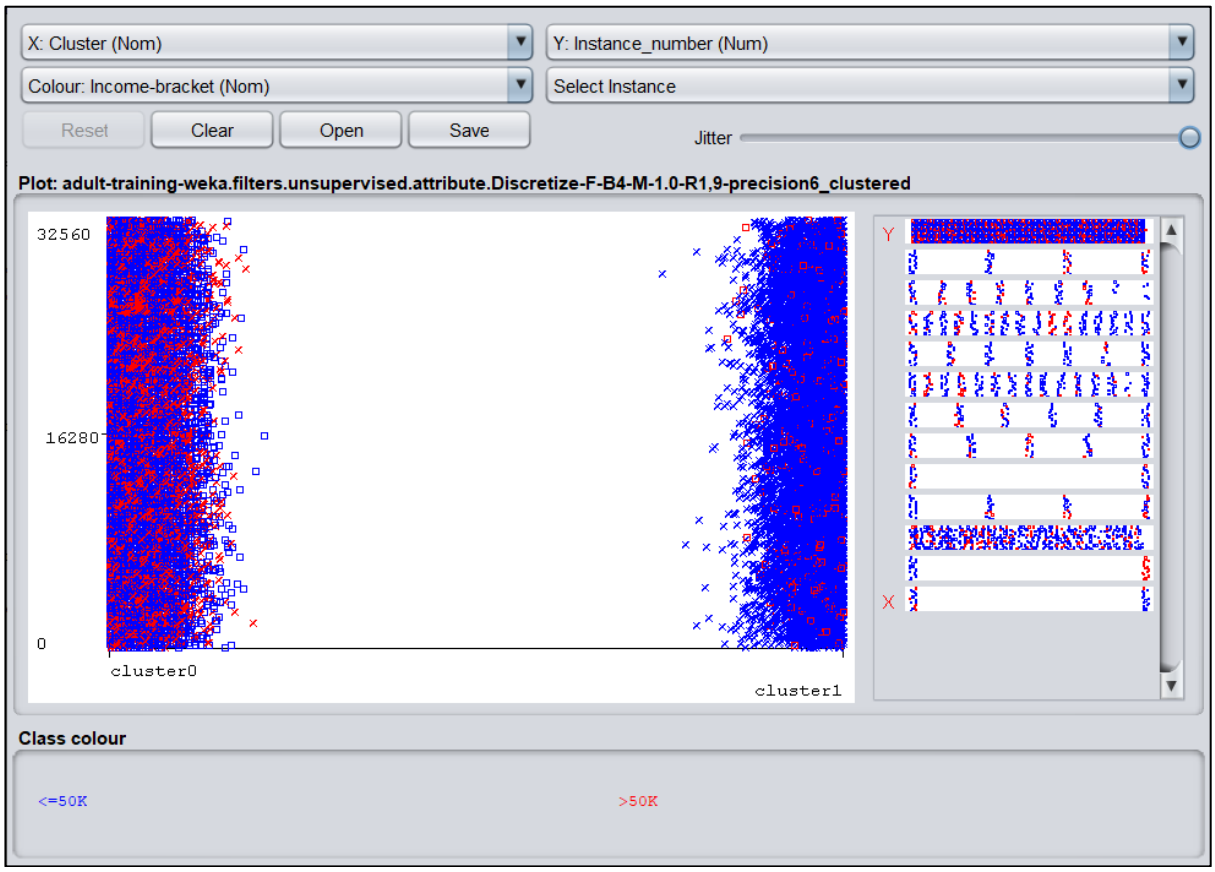
```
Cluster 0: 48_90,' Local-gov',' Doctorate',' Married-civ-spouse',' Prof-specialty',' Wife',' White',' Female',40_41,' United-States'
Cluster 1: 17_28,' Private',' 11th',' Never-married',' Other-service',' Own-child',' White',' Male',1_40,' United-States'

Missing values globally replaced with mean/mode

Final cluster centroids:
                                                        Cluster#
Attribute                                  Full Data            0                1
                                           (32561.0)     (20660.0)        (11901.0)
=============================================================================================
Age                                            37_48         48_90            17_28
Workclass                                    Private       Private          Private
Education                                    HS-grad       HS-grad          HS-grad
Maritial-status                 Married-civ-spouse  Married-civ-spouse  Never-married
Occupation                          Prof-specialty  Prof-specialty  Other-service
Relationship                               Husband       Husband  Not-in-family
Race                                         White         White            White
Gender                                        Male          Male             Male
Hours-per-week                               40_41         40_41            1_40
Native-country                       United-States  United-States  United-States




Time taken to build model (full training data) : 0.16 seconds

=== Model and evaluation on training set ===

Clustered Instances

0      20660 ( 63%)
1      11901 ( 37%)


Class attribute: Income-bracket
Classes to Clusters:

     0     1  <-- assigned to cluster
 13382 11338 |  <=50K
  7278   563 |  >50K

Cluster 0 <--  >50K
Cluster 1 <--  <=50K

Incorrectly clustered instances :       13945.0  42.8273 %
```

Simple K-Means Clustering in Weka on the original discretized data

The output shows the two greatest clusters (0: >50k and 1: <=50k) belonged to groups of people who were very similar in some attributes and very different in others. This was interesting because the common attributes for cluster 0 seemed to be as expected; they were the attributes that were overly represented in the data. These were Workclass => Private, Marital-status => Married-civ-spouse, Race => White, Gender => Male, etc.

The area that was more interesting to me, however, was of the attributes of those in cluster 1. These were instances where the person was never married, in an other-service occupation, not in a family and working less than 40 hours per week. These were different to attributes found using other analysis methods as those were only looking for factors correlating to a higher income, not a lower income.

In order to try and reduce the high amount of incorrectly clustered instances (42.83%), I decided to convert the data to binary using the NominalToBinary filter. While this did lower the percentage to 28.22%, it came at the cost of interpretability of the clusters, so I decided to stay with the original instance and use this for visualisation. The visualisation clearly displays the dissimilarity between the clusters in the form of the distance between them.

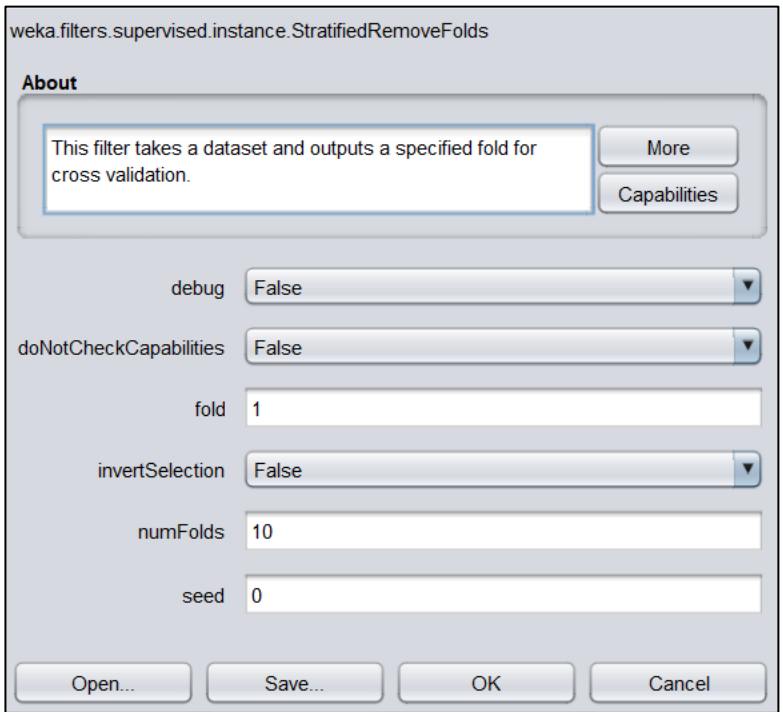


Visualization of clusters in Weka using simple k-means method

Data Reduction

Numerosity Reduction Using Stratified Folds

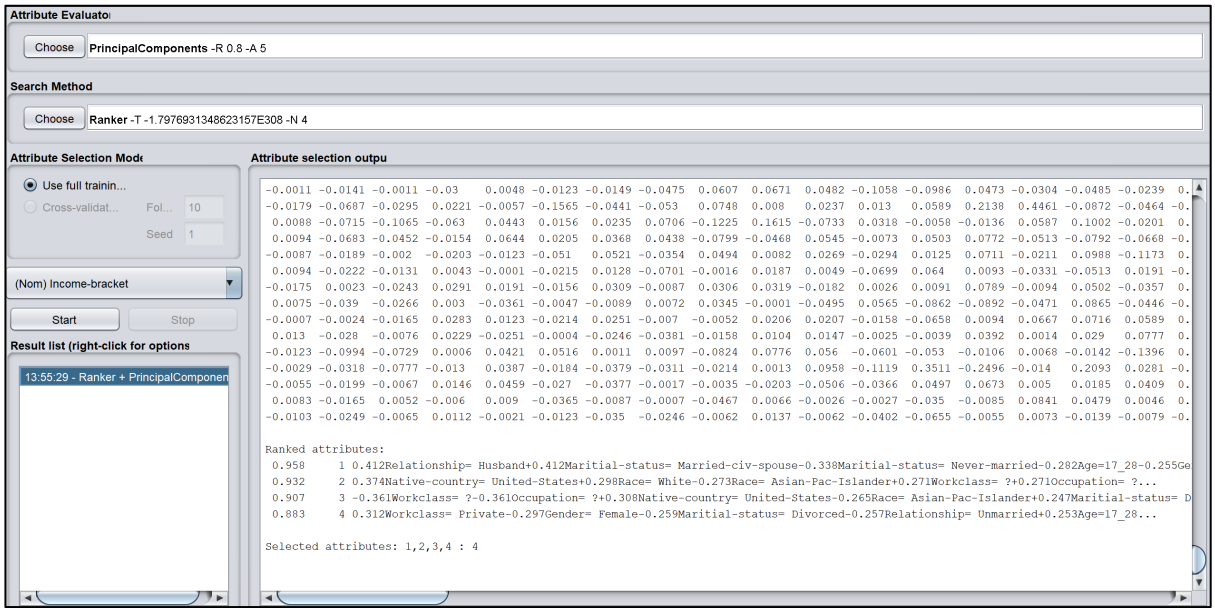
To reduce the data, I started with the original, discretized data and performed numerosity reduction in the form of stratified sampling. This allowed each attribute to be stratified such that their total number of instances approximately equalled the ratio of those in the initial data, therefore preserving the actual patterns present. This was done in Weka using the StratifiedRemoveFolds filter and reduced the total number of instances in the dataset from 32561 to 3257. Although numerosity reduction has little to no effect on the actual, it allows for faster processing and simplification if further manipulation is going to be carried out on the data set.



Stratified remove folds filter in Weka

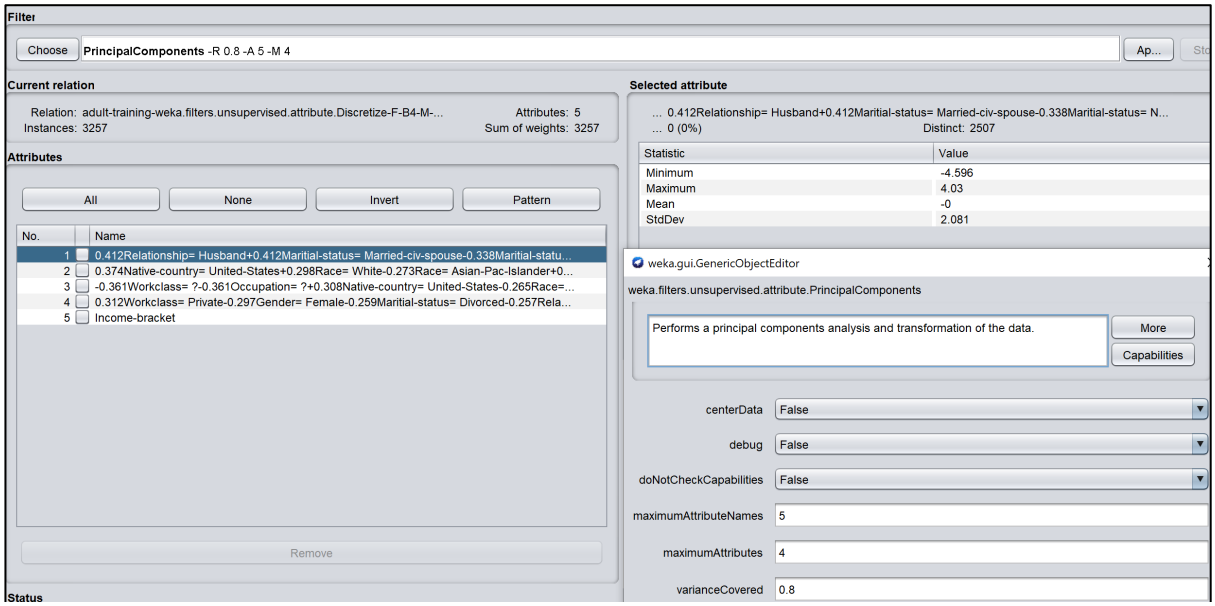
Feature Reduction Using PCA

Principal component analysis (PCA) can be used as a method to find out how much of the data is useful in order to reduce the data and keep only what is necessary. Using the newly stratified data, I decided to use PCA with a variance of 80% and only select the top 4 ranked features from this, as I was going to be modelling a decision tree based on this, and it would be simple to compare it to my previous decision trees since they each utilise 4 attributes also. To do this in Weka, I used the 'Select attributes' tab and the PrincipalComponents attribute evaluator (with a varianceCovered set to 0.8) and Ranker as the search method (with numToSelect set to 4). This allowed for the top four features which account for at least 80% of the variance to be outputted. They are also ranked according to their variance.



PCA output from select attributes tab in Weka

To see a similar list of features in the 'Preprocess' tab, I used the PrincipalComponents filter on the same data using the same properties for variance and number of attributes. Using this view showed the variance of each feature in the 'Selected attribute' output in the form of StdDev. These generated features involved most of the same attributes generated through other learning techniques, and since the PCA was performed on the original dataset, it is subject to the same overpopulation bias for certain attributes as mentioned previously.



PCA output from preprocess tab in Weka

Using this numerosity and feature reduced data, I went on to build a decision tree, tree (3), to compare it to previous trees created in the same way with more abundant data. Since tree (3) used stratified data, I used a minNumObj of 500 to compare it with previous tree with a minNumObj of 1000. The tree itself was largely incomprehensible, but the details generated were able to easily be compared to those of previous trees.

```
Time taken to build model: 0.05 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        2558               78.5385 %
Incorrectly Classified Instances       699               21.4615 %
Kappa statistic                          0.3876
Mean absolute error                      0.2938
Root mean squared error                  0.3845
Relative absolute error                 80.2709 %
Root relative squared error             89.8957 %
Total Number of Instances             3257

=== Detailed Accuracy By Class ===

                 TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                 0.878     0.507     0.845       0.878   0.861      0.389    0.766     0.894     <=50K
                 0.493     0.122     0.563       0.493   0.525      0.389    0.766     0.485     >50K
Weighted Avg.    0.785     0.414     0.777       0.785   0.780      0.389    0.766     0.796

=== Confusion Matrix ===

    a    b   <-- classified as
 2171  301 |    a =  <=50K
  398  387 |    b =  >50K
```

Classifier output using J48 for tree (3) in Weka

Looking at the time taken to build this model: 0.05 seconds, we can see that it much shorter than the times taken to build the models for tree (1) and tree (2) (0.13 and 0.14 seconds respectively). This is one major advantage of using reduced data, as it is quicker to process simply due to less data and features present. In terms of the precision value for each tree (81.8%, 83.6% and 84.5%) we can see that tree (3) has the most precision, but actually has the smallest recall (87.8%) and f-measure (86.1%) of the three trees. These differences can be attributed to changes across attribute selection, database size, sampling bias, and features used.

Comparing the Types of Learning

Putting it all together, I believe that in terms of the US Adult Income dataset, using a database with the most amount of data and discarding only attributes that are not important to the goal of the analysis is an ideal way to interpret the data. This would be different for something such as a data warehouse to keep track of transactions, as processing speed and simplicity would be much more important factors. In terms of the different types of learning used in this analysis, I believe that the association rule mining and classification techniques provided the best insight into understanding which factors (not including hereditary attributes) provided the best recommendation of how any person could improve their income. The association rules mined were easy to understand and the visual decision trees were simple to interpret compared to the models generated using clustering and feature reduced data. If the data reduction techniques were instead performed on an attribute reduced dataset without hereditary attributes, the models created would be more aligned with the goal of the analysis, but the slight reduction in numerosity accuracy would make these models generated slightly less accurate also.