National Institute Of Technology, Calicut

Department of
Computer Science and Engineering

Faculty Incharge:

Mr. Bharath Narayanan

Group: 12

1. G. V. S. Rahul (B120715CS)
2. Abitha Thankaraj (B130177CS)
3. Amala Babu (B130463CS)
4. Amritha Anna Roy (B130407CS)

5.

Table Of Contents

1 INTRODUCTION

1.1 Project Overview

This project involves applying various data cleaning techniques using KNIME and thereby transform the data in such a way that it is fit for data mining. To address the problem statement and apply various mining techniques to get an appropriate result.

1.2 Project Deliverables

Data Pre-processing

- Noisy data
- Missing values
- Inconsistent data

Classification

- Decision Tree Model
- Naive Bayesian Classification Method
- Random Forests

Associations

- Apriori

Clustering

- K-means method

Correlation

- Linear Correlation

Attribute Subset Selection

- Correlation feature based Subset selection

Model Evaluation

Metrics for evaluating Classifier Performance

- Accuracy
- Precision
- Confusion Matrix

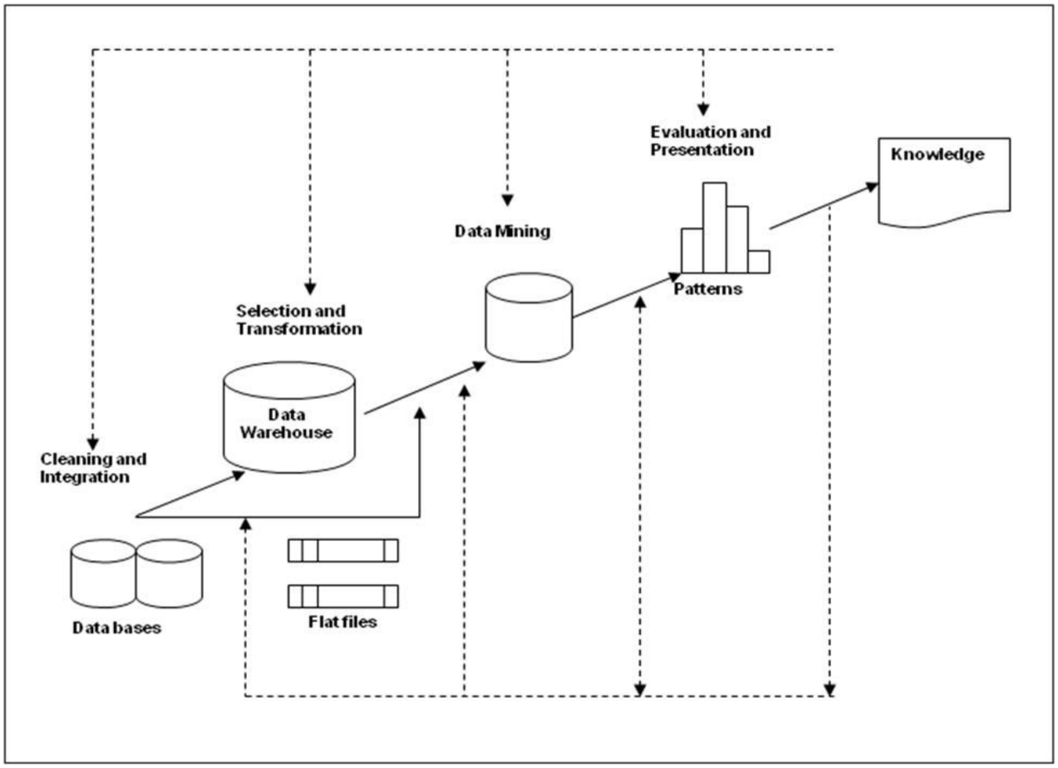
- Recall

Modifications to the dataset

- Training Set
- Test Set

2 PROCESS ORGANISATION

2.1 Process Model



2.2 Roles and Responsibilities

1. Amritha and Amala discovered the data set from UCI Machine Learning Repository.
2. Everyone is involved in cleaning the data using knime.
3. Tasks in Openrefine are completed by Amritha and Amala.
4. Mining Processes in Knime was done by Rahul, Amritha (Clustering), Abitha (Classification), Amala (Association).
5. Final report was made by Rahul and Abitha.

2.3 Tools and Techniques used

- **Data Cleaning Tool**
  - OpenRefine
  - Knime

3 PROJECT MANAGEMENT PLAN

3.1 DataSet Description

This data set contains weighted census data extracted from the 1994 and 1995 current population surveys conducted by the U.S. Census Bureau. The data contains demographic and employment related variables. There are 199,523 instances in the data file and 99,762 in the test file. The data was split into train/test in approximately 70:30 proportions. Number of attributes in data set is 40 (continuous : 7 nominal : 33). The *“instance weight”* attribute indicates the number of people in the population that each record represents due to stratified sampling.

Data Set Attributes
91 distinct values for attribute #0 (age) continuous
9 distinct values for attribute #1 (class of worker) nominal
52 distinct values for attribute #2 (detailed industry code) nominal
47 distinct values for attribute #3 (detailed occupation code) nominal
17 distinct values for attribute #4 (education) nominal
1240 distinct values for attribute #5 (wage per hour) continuous
3 distinct values for attribute #6 (enroll in edu inst last wk) nominal
7 distinct values for attribute #7 (marital stat) nominal

24 distinct values for attribute #8 (major industry code) nominal
5 distinct values for attribute #9 (major occupation code) nominal
5 distinct values for attribute #10 (race) nominal
0 distinct values for attribute #11 (hispanic origin) nominal
2 distinct values for attribute #12 (sex) nominal
3 distinct values for attribute #13 (member of a labor union) nominal
6 distinct values for attribute #14 (reason for unemployment) nominal
8 distinct values for attribute #15 (full or part time employment stat) nominal
132 distinct values for attribute #16 (capital gains) continuous
113 distinct values for attribute #17 (capital losses) continuous
1478 distinct values for attribute #18 (dividends from stocks) continuous
6 distinct values for attribute #19 (tax filer stat) nominal
6 distinct values for attribute #20 (region of previous residence) nominal
51 distinct values for attribute #21 (state of previous residence) nominal
38 distinct values for attribute #22 (detailed household and family stat) nominal
8 distinct values for attribute #23 (detailed household summary in household) nominal
10 distinct values for attribute #24 (migration code-change in msa) nominal
9 distinct values for attribute #25 (migration code-change in reg) nominal
10 distinct values for attribute #26 (migration code-move within reg) nominal
3 distinct values for attribute #27 (live in this house 1 year ago) nominal
4 distinct values for attribute #28 (migration prev res in sunbelt) nominal
7 distinct values for attribute #29 (num persons worked for employer) continuous
5 distinct values for attribute #30 (family members under 18) nominal
43 distinct values for attribute #31 (country of birth father) nominal
43 distinct values for attribute #32 (country of birth mother) nominal
43 distinct values for attribute #33 (country of birth self) nominal
5 distinct values for attribute #34 (citizenship) nominal
3 distinct values for attribute #35 (own business or self employed) nominal
3 distinct values for attribute #36 (fill inc questionnaire for veteran's admin) nominal
3 distinct values for attribute #37 (veterans benefits) nominal
53 distinct values for attribute #38 (weeks worked in year) continuous
2 distinct values for attribute #39 (year) nominal

3.2 Tasks

3.2.1 Pre-processing

- Data Selection
- Data Cleaning
- Data Integration and transformation
- Data reduction

- Data Discretization

3.3 Description

3.3.1 Pre-Processing

3.3.1.1 Data Cleaning

(i) Missing Values

The 3 migration code attributes namely, '*country of birth for self*' , '*country of birth for father*' and *'country of birth for mother'* had missing values denoted by '?' symbol. They were replaced with the value *'Unknown'*.

Reason: Statistical measures such as mean, median or mode cannot be considered to replace these values as no information to determine whether each person has migrated or not or their birth country since each tuple is unique in its own accord.

(ii) Inconsistent data

Tuples that contained inconsistent data were removed. Those tuples which satisfied the following conditions were removed:

(a) *detailed household and family stat* -> spouse of household
and
age -> 0-18

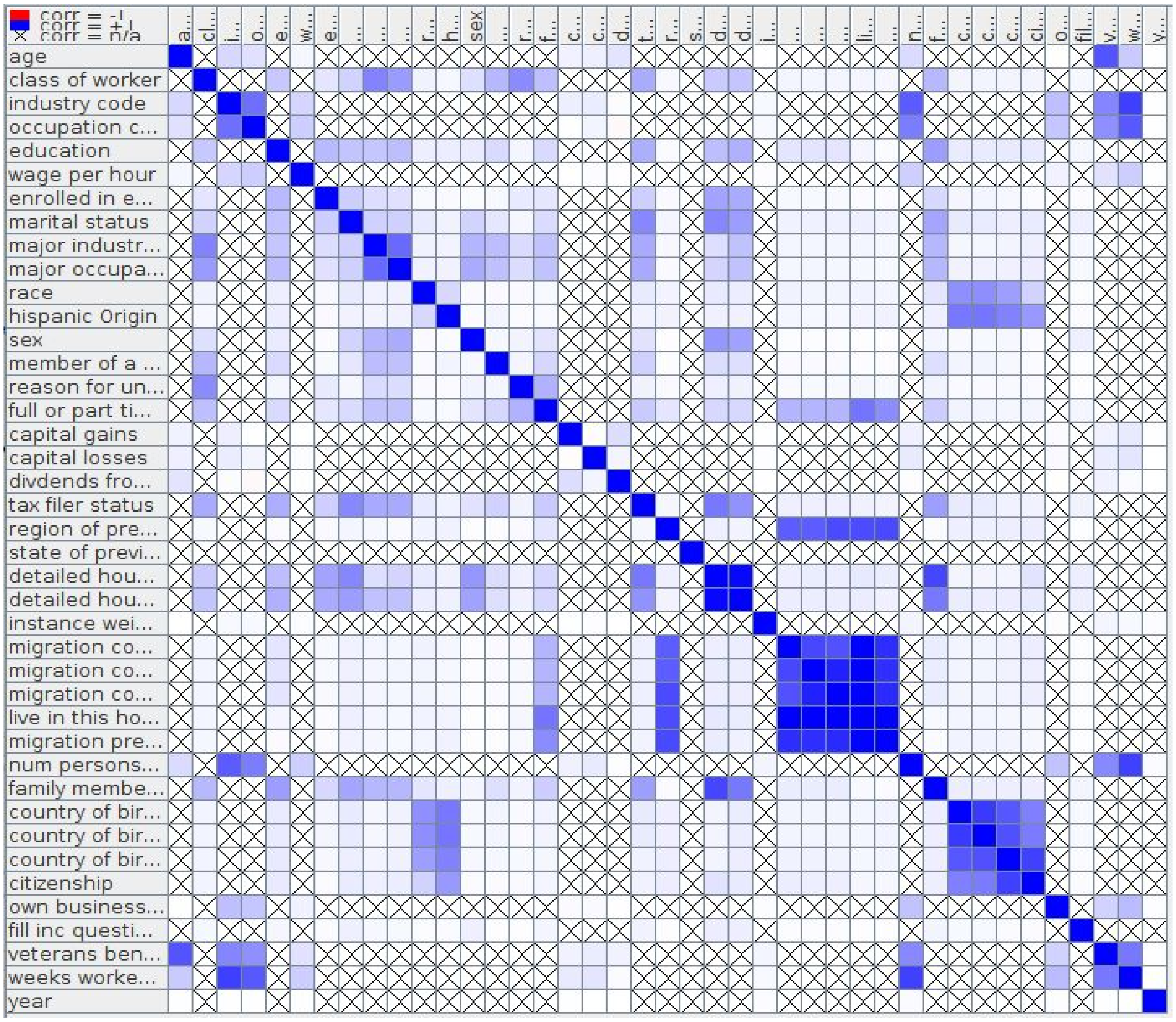
(b) *age* -> 0-18
and
marital status -> divorced , spouse absent , separated , married civilian spouse present , married A-F spouse present , widowed

3.3.1.2 Data Integration and Transformation

Integration

(i) Correlation Analysis

Four attributes which were found to be positively correlated with correlation coefficient 0.982 (on performing correlation analysis) were removed, namely *migration code - change in msa, migration code - change in reg, migration code - move within reg and migration prev res in sunbelt.*



Correlation Matrix

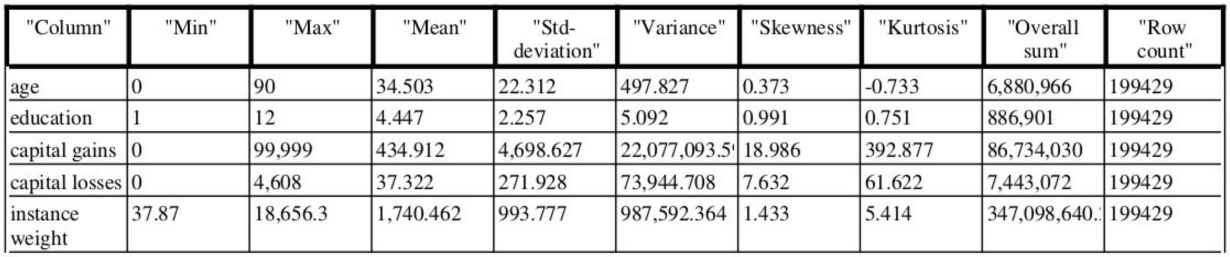
Transformation

(i) Attribute construction

The product of the attributes, *wage per hour* and *no of weeks worked per year* was found and multiplied by 40 (assuming 40 hours of work per individual for a week) to form a new attribute, '*total income*'. (Further analysis on this attribute construction mentioned in the '*dimensionality reduction'*)

(ii) Aggregation

Summary or aggregation operations were applied to the data using statistics node in knime. A summary containing the mean , median and other values related to the data was calculated.
The standard deviations in the data indicates that there is a significant quantity of values in all attributes, particularly in the case of the instance weight attribute and the capital gains attribute.



"Column"	"Min"	"Max"	"Mean"	"Std-deviation"	"Variance"	"Skewness"	"Kurtosis"	"Overall sum"	"Row count"
age	0	90	34.503	22.312	497.827	0.373	-0.733	6,880,966	199429
education	1	12	4.447	2.257	5.092	0.991	0.751	886,901	199429
capital gains	0	99,999	434.912	4,698.627	22,077,093.5	18.986	392.877	86,734,030	199429
capital losses	0	4,608	37.322	271.928	73,944.708	7.632	61.622	7,443,072	199429
instance weight	37.87	18,656.3	1,740.462	993.777	987,592.364	1.433	5.414	347,098,640.	199429

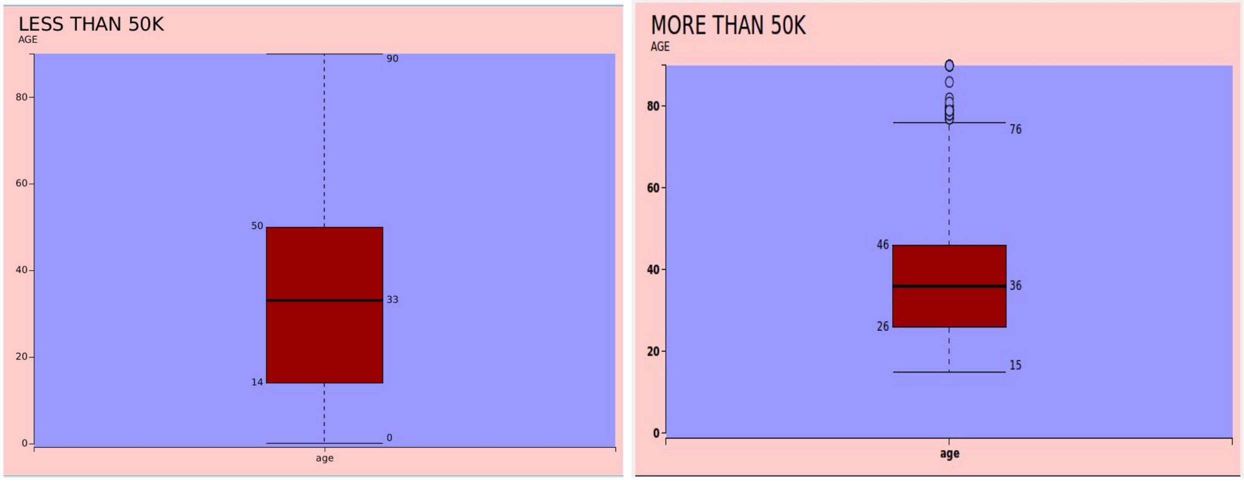
3.3.1.3 Data Reduction

(i) Dimensionality Reduction

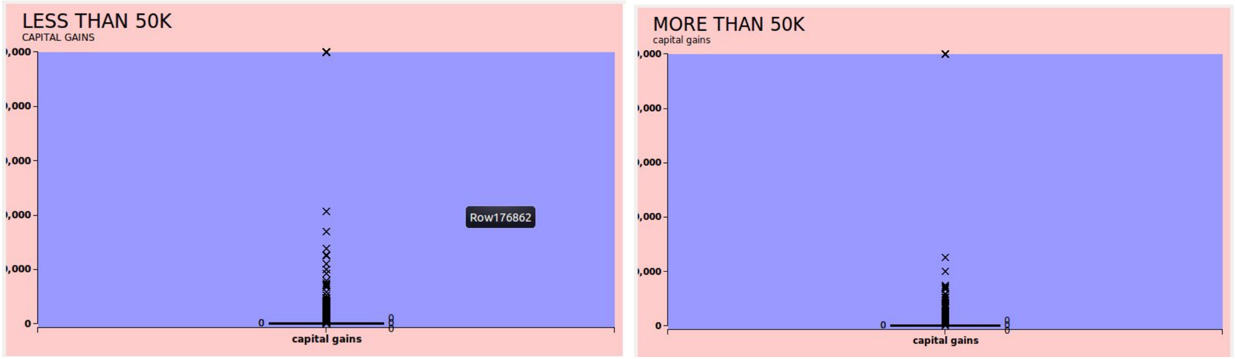
The two attributes, *wages per hour* and *no: of weeks worked per year* were removed after the attribute construction, as they were felt to be redundant and did not contribute to the analysis of the data later on.

[This dimensionality reduction was discarded during the mining process, as the constructed attribute was found to be inconsistent due to the possibility of self - employed individuals and other individuals who has other means of income and hence would have '*wages per hour'* value to be 0]

Boxplots were created for the data split according to who earned below 50k and above 50k , and we came to the conclusion that some of the attributes like the *capital gain*, *capital loss* and *dividend from stocks* have no effect on our further analysis.
From the box plots it can be seen that the range of attribute values for *age* and *education* is slightly higher in >50K class instances indicating that these attributes may have predictive significance. [Box plots are given below]



AGE



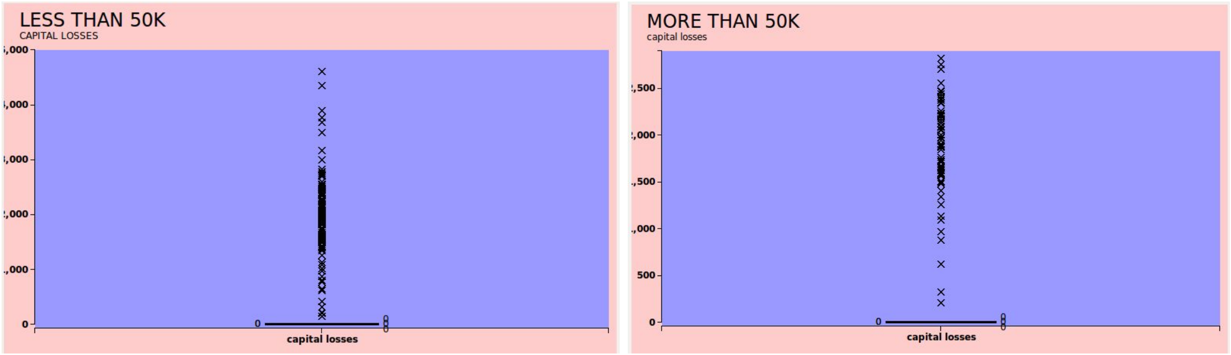
LESS THAN 50K

CAPITAL GAINS

MORE THAN 50K

capital gains

Capital Gains



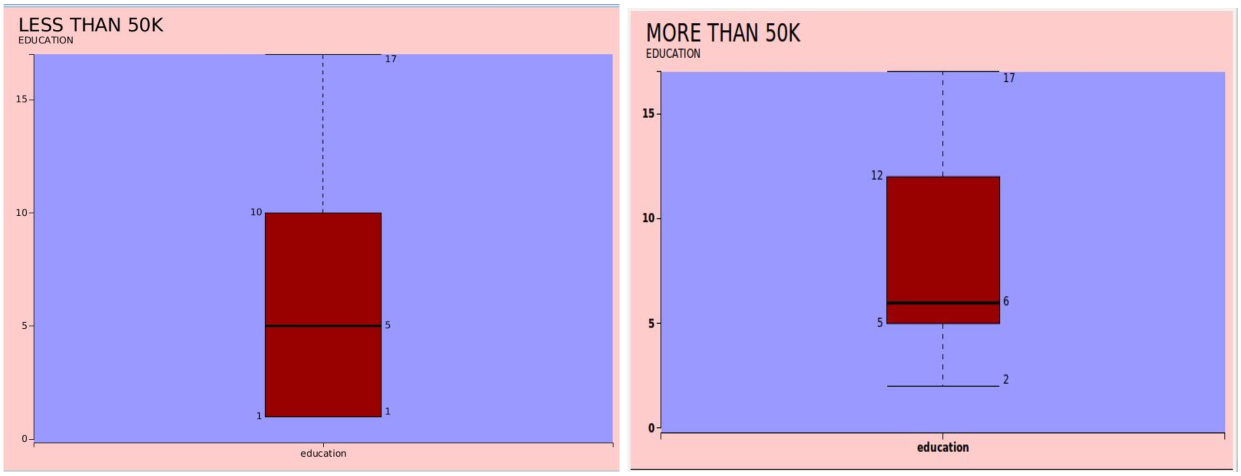
LESS THAN 50K

CAPITAL LOSSES

MORE THAN 50K

capital losses

Capital Losses



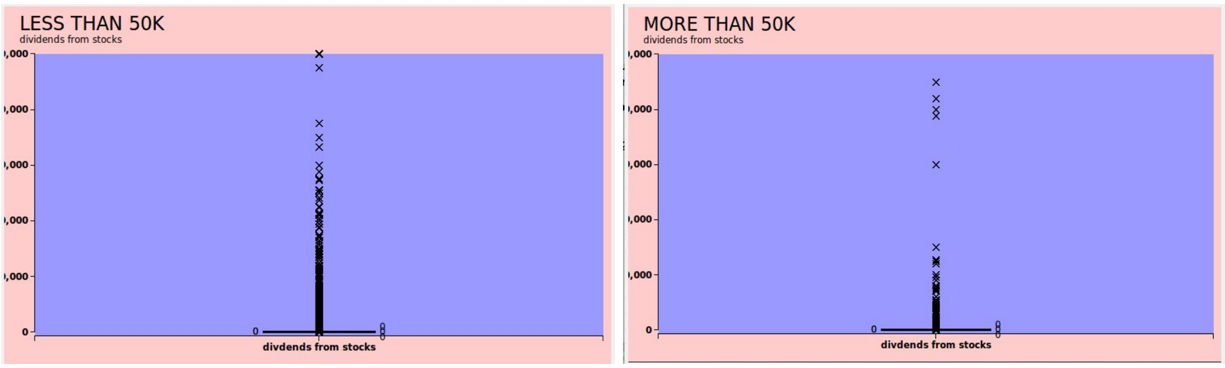
LESS THAN 50K

EDUCATION

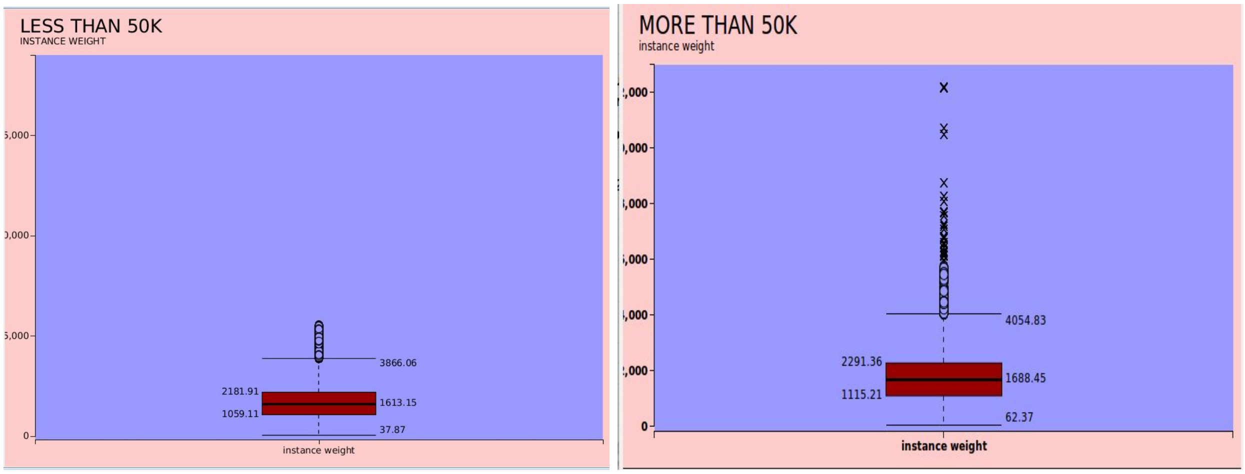
MORE THAN 50K

EDUCATION

Education



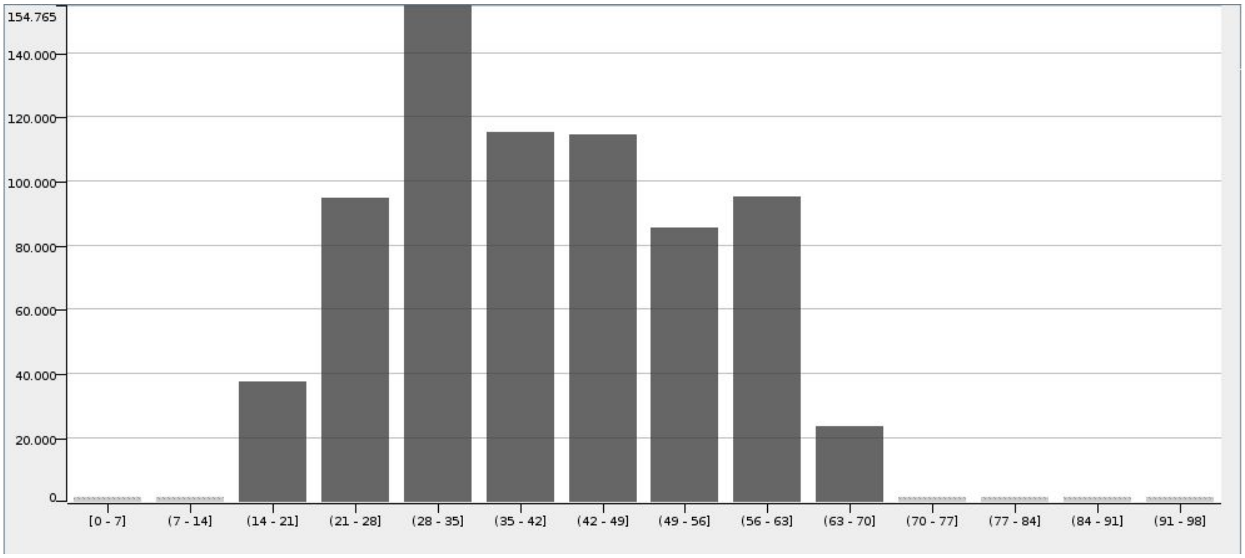
Dividends from Stock



Instance Weight

(ii) Numerosity Reduction

Binning was used for the age attribute.*Age* was split into disjoint buckets/bins.This was done with the help of a histogram, we experimented with various number of bins and concluded from the final histogram which binning intervals should be chosen.
On the y axis : wages per hour was plotted and the x axis depicted age intervals.



3.3.1.4 Data Discretization

(i) The following changes have been applied to the education column,

(a) The columns *'10th grade', '11th grade', '12th grade no diploma*' have been grouped as 'below 12th grade'.
(b) Children between 4 to 14 age who were given 'Children' tag were put into 'below 12th grade'
(c) Children with 0-3 age are given 'Children' tag
(d) The value 'less than 1st grade' has not been included in the 'below 12th grade' group because it consisted of individuals whose ages were between 15 and 90 which points to the fact that they were relatively uneducated.

(ii) We have seven continuous attributes in the dataset and one generated by applying mathematical formula , namely :
age, wage per hour, capital gains, capital losses, dividends from stocks, instance weight and *weeks worked in year , total income* (generated attribute).

age: We have applied the binning technique and hence applied discretization on this attribute.
wage per hour, weeks worked in year: These two attributes are combined to form the new attribute, *total income*. (The resulting attribute was found to be inconsistent due to the above mentioned reasons .)
capital gains, capital losses, dividend from stocks, instance weight: Using the box plots, we have come to the conclusion that these attributes have minimal impact on the further analysis and hence no discretization methods were applied to it.

3.4 Mining

3.4.1 Classification

Classification is done to know exactly how the data is being classified.The classification tools in Knime operate on some classification algorithm and the algorithm which has the highest accuracy for the data-set is selected.
We applied the following classification techniques to the preprocessed data-set:
(i)Decision Tree classification
(ii)Naive Bayes classification
(iii)Random Forest

Modification to the Data

The following modifications were applied to the data-set:
The data-set initially contained 1,99,523 tuples of which only 12,382 tuples had data of individuals earning greater than 50,000 dollars (class : +50000). As the data was skewed ,stratified sampling was done on 1,87,141 tuples (class : -50000), and we obtained 13,000 tuples. A data-set containing 12,382 positive (class : +50000) and 13000 negative (class : -50000) values were obtained. Since the *"instance weight"* attribute indicates the number of people in the population that each record represents , we oversampled the data , so that we could represent the data proportionately .
OverSampling: The data was binned according to the instance weight and oversampled accordingly.

Classification techniques were applied on this data-set. 70% of the tuples were taken for training the classifier (training set) and the remaining 30% were used for testing (testing set).

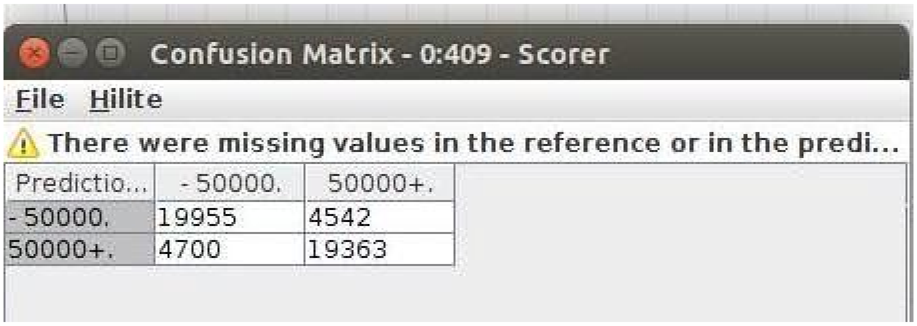
Model Evaluation

Based on data mining techniques described above, all the developed models are evaluated in terms of following error measures –

Accuracy: Is a percentage of samples that are classified correctly.
Accuracy = (TP + TN) / (P + N)
Sensitivity = TP/ (TP+FN)
Specificity = TN/ (TN +FP)

Where TP, TN, FP and FN are true positive, true negative, false positive and false negative respectively.
The following accuracy statistics were obtained for the different classification models:

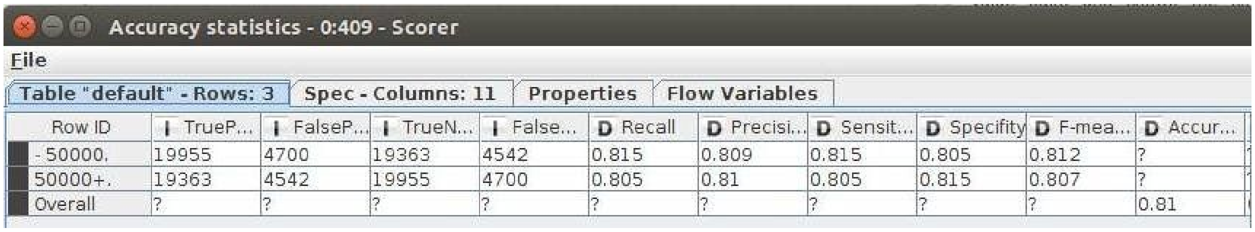
3.4.1.1 Decision Tree classification



⚠ There were missing values in the reference or in the predi...

Predictio...	- 50000.	50000+.
- 50000.	19955	4542
50000+.	4700	19363

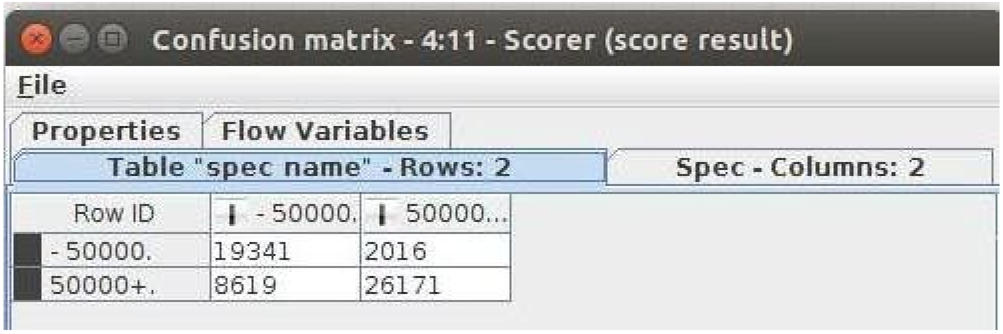
Confusion Matrix


Accuracy statistics - 0:409 - Scorer
File
Table "default" - Rows: 3
Spec - Columns: 11
Properties
Flow Variables

Row ID	I TrueP...	I FalseP...	I TrueN...	I False...	D Recall	D Precisi...	D Sensit...	D Specifity	D F-mea...	D Accur...
- 50000.	19955	4700	19363	4542	0.815	0.809	0.815	0.805	0.812	?
50000+.	19363	4542	19955	4700	0.805	0.81	0.805	0.815	0.807	?
Overall	?	?	?	?	?	?	?	?	?	0.81

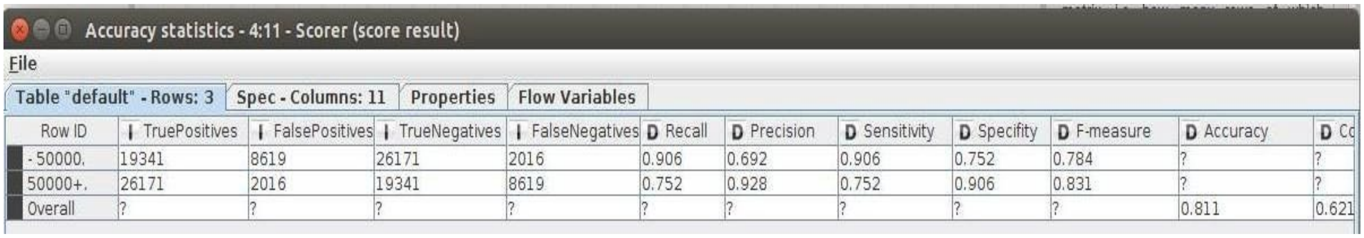
Accuracy Statistics

3.4.1.2 Naive Bayes classification


Confusion matrix - 4:11 - Scorer (score result)
File
Properties
Flow Variables
Table "spec name" - Rows: 2
Spec - Columns: 2

Row ID	I - 50000.	I 50000...
- 50000.	19341	2016
50000+.	8619	26171

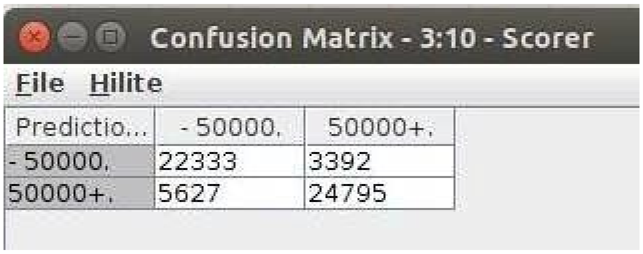
Confusion Matrix


Accuracy statistics - 4:11 - Scorer (score result)
File
Table "default" - Rows: 3
Spec - Columns: 11
Properties
Flow Variables

Row ID	I TruePositives	I FalsePositives	I TrueNegatives	I FalseNegatives	D Recall	D Precision	D Sensitivity	D Specifity	D F-measure	D Accuracy	D Co
- 50000.	19341	8619	26171	2016	0.906	0.692	0.906	0.752	0.784	?	?
50000+.	26171	2016	19341	8619	0.752	0.928	0.752	0.906	0.831	?	?
Overall	?	?	?	?	?	?	?	?	?	0.811	0.621

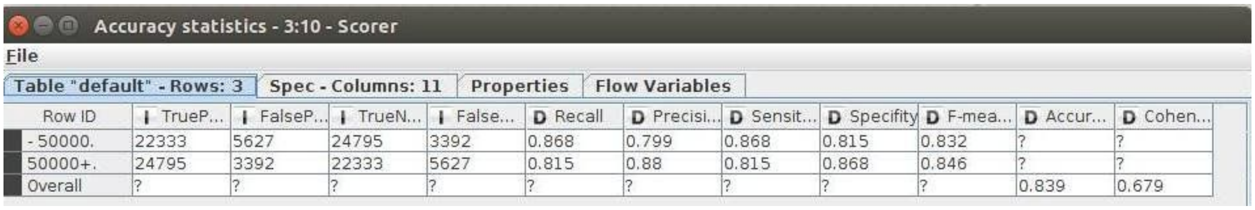
Accuracy statistics

3.4.1.3 Random Forest classification



Predictio...	- 50000.	50000+.
- 50000.	22333	3392
50000+.	5627	24795

Confusion Matrix

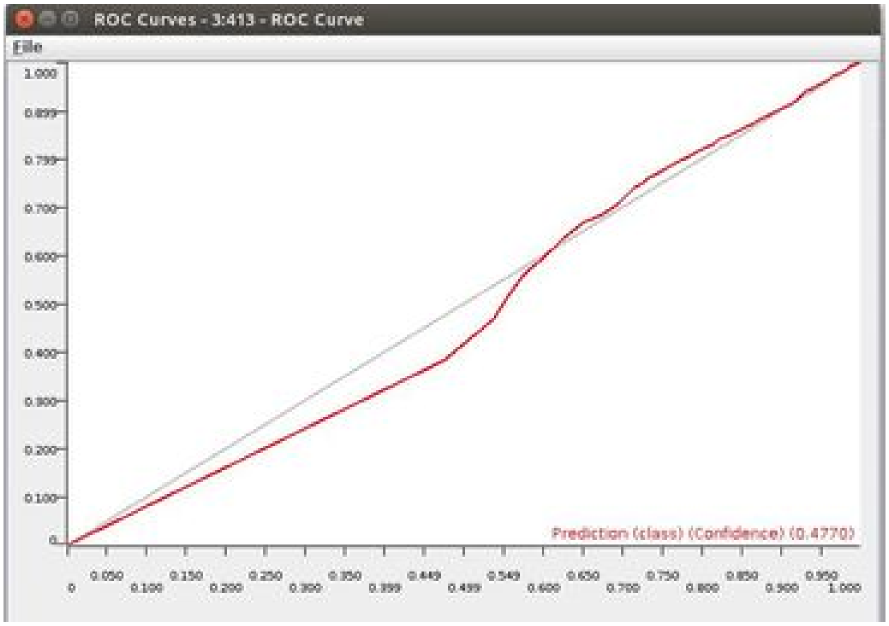

Accuracy statistics - 3:10 - Scorer
File
Table "default" - Rows: 3
Spec - Columns: 11
Properties
Flow Variables

Row ID	I TrueP...	I FalseP...	I TrueN...	I False...	D Recall	D Precisi...	D Sensit...	D Specifity	D F-mea...	D Accur...	D Cohen...
- 50000.	22333	5627	24795	3392	0.868	0.799	0.868	0.815	0.832	?	?
50000+.	24795	3392	22333	5627	0.815	0.88	0.815	0.868	0.846	?	?
Overall	?	?	?	?	?	?	?	?	?	0.839	0.679

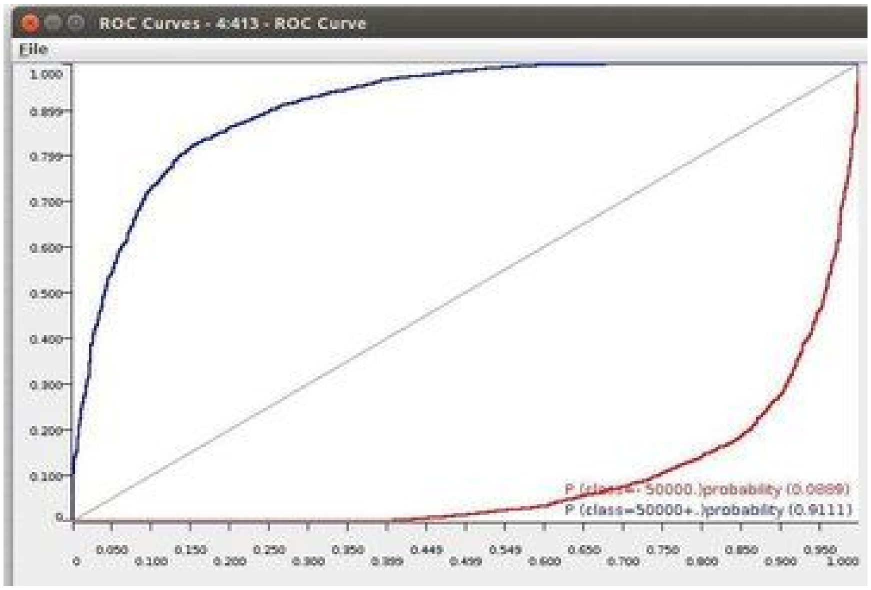
Accuracy statistics

Receiver Operating Characteristics (ROC) curve

In order to compare the performance of the classifiers, a ROC curve was generated for multiple classifiers including Naïve Bayes and Random forest.



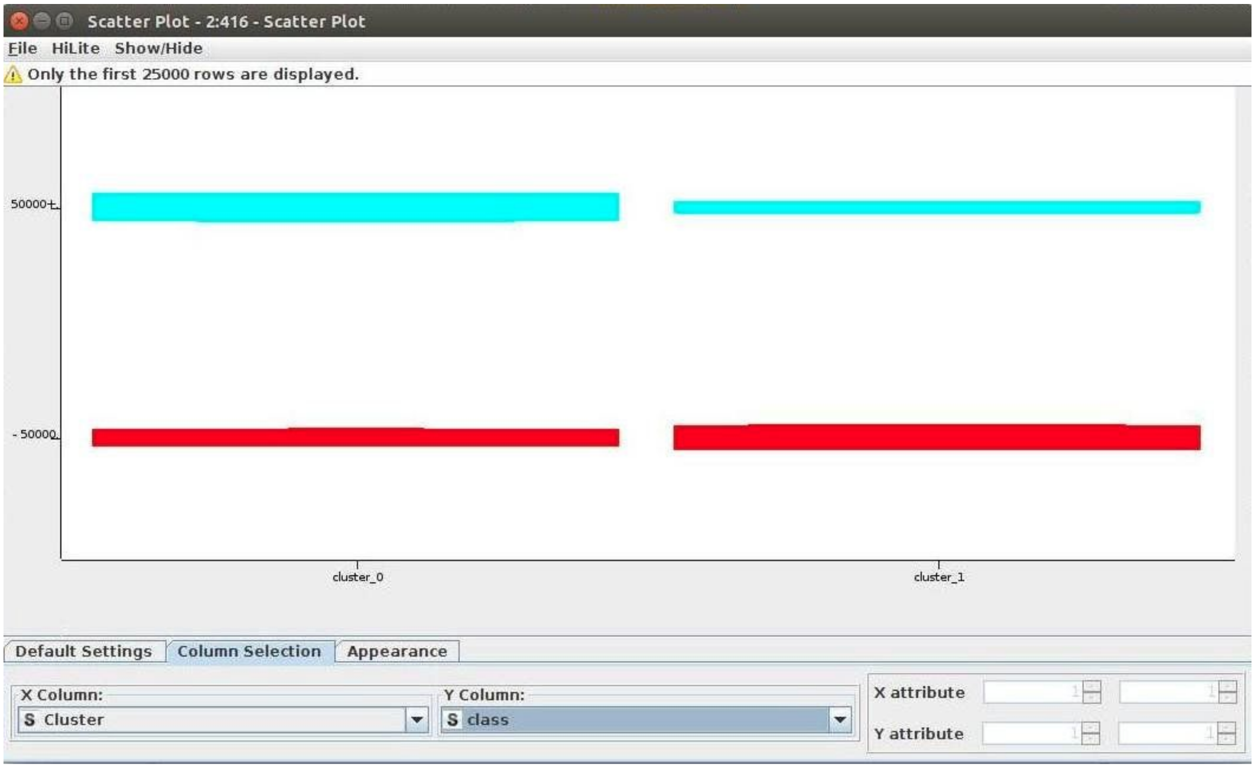
ROC curve for Random forest



ROC curve for Naive Bayes classifier

3.4.2 Clustering

Clustering techniques are used to find and group data points with natural similarities. And this is done as a part of Unsupervised data mining, which, does not focus on predetermined attributes, nor does it predict a target value.



⚠ Only the first 25000 rows are displayed.

X Column: S Cluster

Y Column: S class

X attribute 1 1

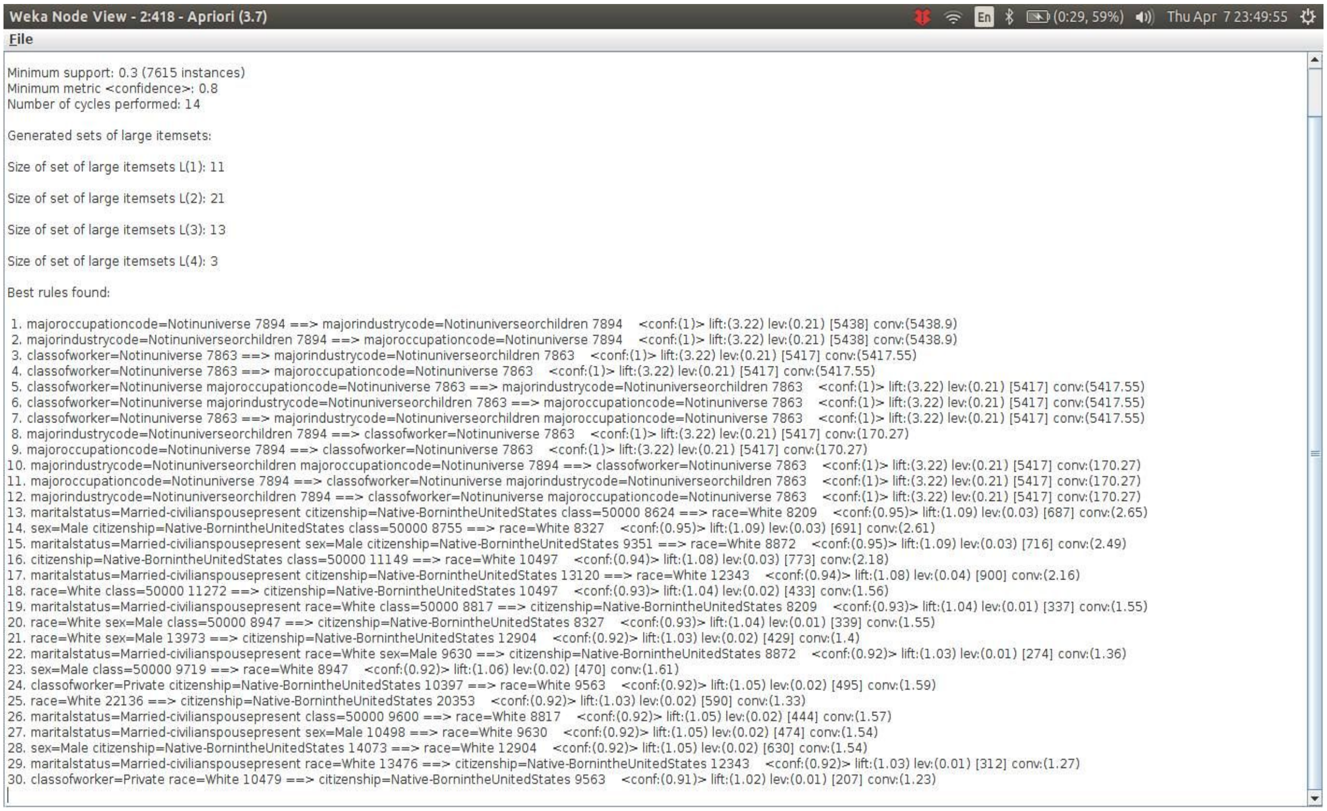
Y attribute 1 1

Scatter plot for K-Means clustering

But since, Clustering only finds hidden structure and relation among data, and since we are not attempting to find appropriate 'categories' for our data, Clustering is not an appropriate technique for this data mining process.

3.4.3 Association

A certain number of association rules were found using the Apriori algorithm . The measure called "confidence" was used to extract the most significant rules.
The confidence of an association rule *A→C* with antecedent *A* and consequent *C* is defined to be the ratio **P(*A* ∩ *C*)/P(*C*)**. The higher the ratio the more confidence we have in the rule.



```
Minimum support: 0.3 (7615 instances)
Minimum metric <confidence>: 0.8
Number of cycles performed: 14

Generated sets of large itemsets:

Size of set of large itemsets L(1): 11

Size of set of large itemsets L(2): 21

Size of set of large itemsets L(3): 13

Size of set of large itemsets L(4): 3

Best rules found:

 1. majoroccupationcode=Notinuniverse 7894 ==> majorindustrycode=Notinuniverseorchildren 7894    <conf:(1)> lift:(3.22) lev:(0.21) [5438] conv:(5438.9)
 2. majorindustrycode=Notinuniverseorchildren 7894 ==> majoroccupationcode=Notinuniverse 7894    <conf:(1)> lift:(3.22) lev:(0.21) [5438] conv:(5438.9)
 3. classofworker=Notinuniverse 7863 ==> majorindustrycode=Notinuniverseorchildren 7863    <conf:(1)> lift:(3.22) lev:(0.21) [5417] conv:(5417.55)
 4. classofworker=Notinuniverse 7863 ==> majoroccupationcode=Notinuniverse 7863    <conf:(1)> lift:(3.22) lev:(0.21) [5417] conv:(5417.55)
 5. classofworker=Notinuniverse majoroccupationcode=Notinuniverse 7863 ==> majorindustrycode=Notinuniverseorchildren 7863    <conf:(1)> lift:(3.22) lev:(0.21) [5417] conv:(5417.55)
 6. classofworker=Notinuniverse majorindustrycode=Notinuniverseorchildren 7863 ==> majoroccupationcode=Notinuniverse 7863    <conf:(1)> lift:(3.22) lev:(0.21) [5417] conv:(5417.55)
 7. classofworker=Notinuniverse 7863 ==> majorindustrycode=Notinuniverseorchildren majoroccupationcode=Notinuniverse 7863    <conf:(1)> lift:(3.22) lev:(0.21) [5417] conv:(5417.55)
 8. majorindustrycode=Notinuniverseorchildren 7894 ==> classofworker=Notinuniverse 7863    <conf:(1)> lift:(3.22) lev:(0.21) [5417] conv:(170.27)
 9. majoroccupationcode=Notinuniverse 7894 ==> classofworker=Notinuniverse 7863    <conf:(1)> lift:(3.22) lev:(0.21) [5417] conv:(170.27)
10. majorindustrycode=Notinuniverseorchildren majoroccupationcode=Notinuniverse 7894 ==> classofworker=Notinuniverse 7863    <conf:(1)> lift:(3.22) lev:(0.21) [5417] conv:(170.27)
11. majoroccupationcode=Notinuniverse 7894 ==> classofworker=Notinuniverse majorindustrycode=Notinuniverseorchildren 7863    <conf:(1)> lift:(3.22) lev:(0.21) [5417] conv:(170.27)
12. majorindustrycode=Notinuniverseorchildren 7894 ==> classofworker=Notinuniverse majoroccupationcode=Notinuniverse 7863    <conf:(1)> lift:(3.22) lev:(0.21) [5417] conv:(170.27)
13. maritalstatus=Married-civilianspousepresent citizenship=Native-BornintheUnitedStates class=50000 8624 ==> race=White 8209    <conf:(0.95)> lift:(1.09) lev:(0.03) [687] conv:(2.65)
14. sex=Male citizenship=Native-BornintheUnitedStates class=50000 8755 ==> race=White 8327    <conf:(0.95)> lift:(1.09) lev:(0.03) [691] conv:(2.61)
15. maritalstatus=Married-civilianspousepresent sex=Male citizenship=Native-BornintheUnitedStates 9351 ==> race=White 8872    <conf:(0.95)> lift:(1.09) lev:(0.03) [716] conv:(2.49)
16. citizenship=Native-BornintheUnitedStates class=50000 11149 ==> race=White 10497    <conf:(0.94)> lift:(1.08) lev:(0.03) [773] conv:(2.18)
17. maritalstatus=Married-civilianspousepresent citizenship=Native-BornintheUnitedStates 13120 ==> race=White 12343    <conf:(0.94)> lift:(1.08) lev:(0.04) [900] conv:(2.16)
18. race=White class=50000 11272 ==> citizenship=Native-BornintheUnitedStates 10497    <conf:(0.93)> lift:(1.04) lev:(0.02) [433] conv:(1.56)
19. maritalstatus=Married-civilianspousepresent race=White class=50000 8817 ==> citizenship=Native-BornintheUnitedStates 8209    <conf:(0.93)> lift:(1.04) lev:(0.01) [337] conv:(1.55)
20. race=White sex=Male class=50000 8947 ==> citizenship=Native-BornintheUnitedStates 8327    <conf:(0.93)> lift:(1.04) lev:(0.01) [339] conv:(1.55)
21. race=White sex=Male 13973 ==> citizenship=Native-BornintheUnitedStates 12904    <conf:(0.92)> lift:(1.03) lev:(0.02) [429] conv:(1.4)
22. maritalstatus=Married-civilianspousepresent race=White sex=Male 9630 ==> citizenship=Native-BornintheUnitedStates 8872    <conf:(0.92)> lift:(1.03) lev:(0.01) [274] conv:(1.36)
23. sex=Male class=50000 9719 ==> race=White 8947    <conf:(0.92)> lift:(1.06) lev:(0.02) [470] conv:(1.61)
24. classofworker=Private citizenship=Native-BornintheUnitedStates 10397 ==> race=White 9563    <conf:(0.92)> lift:(1.05) lev:(0.02) [495] conv:(1.59)
25. race=White 22136 ==> citizenship=Native-BornintheUnitedStates 20353    <conf:(0.92)> lift:(1.03) lev:(0.02) [590] conv:(1.33)
26. maritalstatus=Married-civilianspousepresent class=50000 9600 ==> race=White 8817    <conf:(0.92)> lift:(1.05) lev:(0.02) [444] conv:(1.57)
27. maritalstatus=Married-civilianspousepresent sex=Male 10498 ==> race=White 9630    <conf:(0.92)> lift:(1.05) lev:(0.02) [474] conv:(1.54)
28. sex=Male citizenship=Native-BornintheUnitedStates 14073 ==> race=White 12904    <conf:(0.92)> lift:(1.05) lev:(0.02) [630] conv:(1.54)
29. maritalstatus=Married-civilianspousepresent race=White 13476 ==> citizenship=Native-BornintheUnitedStates 12343    <conf:(0.92)> lift:(1.03) lev:(0.01) [312] conv:(1.27)
30. classofworker=Private race=White 10479 ==> citizenship=Native-BornintheUnitedStates 9563    <conf:(0.91)> lift:(1.02) lev:(0.01) [207] conv:(1.23)
```

3.5 Limitations of the Tool

KNIME

- It is easier to handle missing values in OpenRefine when compared to KNIME .
- The boxplots were difficult to create due to the limited options present.

- Certain tools like the boxplot and histogram in KNIME have limited options to work with. This consumed much time and brought down the quality of the work produced.

OpenRefine

- Use of text facet in a field with a large number of unique values can lead to “too many to display” error.

3.6 Problem Statement

Prediction task is to determine the income level for the person represented by the record . Demographic information such as the level of education , age , current employment type and similar attributes have been used to identify individuals whose salary exceeds a specified value.The goal field of this data, was drawn primarily from our class label.

On the basis of the mining process done on this dataset, Random Forest Classifier was chosen as the best among the three classifiers tested , due to the following reasons :

- It is unexcelled in accuracy among the other classification techniques used here.
- It runs efficiently on large data bases.
- It can handle thousands of input variables without variable deletion.

3.7 Dependencies and other constraints

Before we could start off with our classification task, we had to first clean the data set which involved getting rid of missing values, noisy data. Only after doing this, we could start the classification task.

4 ABOUT THE SOFTWARE

4.1 Introduction

KNIME, the Konstanz Information Miner, is an open source data analytics, reporting and integration platform. KNIME integrates various components for machine learning and data mining through its modular data pipelining concept. A graphical user interface allows assembly of nodes for data preprocessing (ETL: Extraction, Transformation, Loading), for modeling and data analysis and visualization.

We worked on our data using this tool and it enabled us to manipulate and use the data effectively.

OpenRefine : OpenRefine is an open source application with a web UI, used to clean data and convert to other formats. It was developed by Metaweb Technologies Inc. as Freebase Gridworks in 2010, later acquired by Google and renamed to Google Refine. In 2012, Google stopped providing support to Google Refine, which since then, has become an open source project under the name OpenRefine.

We found it easier to work with OpenRefine to deal with the missing values and the inconsistent data which was very much prevalent in our dataset.

4.2 Reliability

KNIME provides a user with various metrics that can be used to accurately and reliably make prediction and results.We were able to work with several nodes and had the option to choose which method to use and thus were able to choose the most effective method to deal with our data.

OpenRefine is a highly reliable software with options for Undo/Redo that facilitates recovery and thus fault tolerance.

4.3 Availability

KNIME is released under the GNU General Public License making it freely available for users.

OpenRefine is freely available for download at its website.

4.4 Security

KNIME is secure as it does not communicate with over the internet without permission.

OpenRefine ensures security by listening only to TCP. Also, Open Refine, although it is a web application, runs on the local machine, and thus no data ever leaves the local site.

4.5 Maintainability

KNIME is highly maintainable as it reuses many classes and packages, and also contains documentation of every nodes available.

OpenRefine is a free open source tool and is well documented which makes it easy to maintain.

4.6 Portability

KNIME is fully portable and runs on any system that has a working implementation of the Java Runtime Environment. We can export and import workflows easily which makes it easier for a team to work from their respective consoles and still work together.

OpenRefine is portable as it is a web application and can run in any of the modern supported browsers.

4.7 Performance

KNIME supports multithreading which greatly improves performance on multicore CPUs.
OpenRefine is quite a high performance system, provided the RAM needs are available.