



## MSc in Business Analytics

### Business Analytics Practicum

Ass. Pr. Kotidis Ioannis

Name	Code
Aikaterini Kalaidopoulou	BAFT1509
Nikolaos Papakonstantinou	BAFT1510
Konstantinos Tsamis	BAFT1503
Konstantinos Chronis	BAFT1502

### Implementation of data mining technics

on Census data 1994-1995

using



Athens, 28/12/2015

## Table of Contents

Introduction .....	3
Data cleansing .....	3
Data Transformation .....	4
Data Mining Technics .....	7
Prediction of income level .....	7
Classification .....	7
Association Rules .....	16
Clustering .....	19
SVD .....	25
Additional technics .....	26
Results Discussion .....	29
Prediction of income level .....	29
Citizens' segmentation .....	30
References .....	31
Appendix .....	32
Full list of attributes .....	32
Clustering results .....	33
Complete XML code of the clustering analysis .....	41

## Introduction

This project is based on the Census dataset 1994-1995 with demographic data about the American citizens for these year. From the description of the dataset we found that it contains almost 300,000 records with about 40 attributes. This dataset contains weighted demographic data from the census of America for the years 1994 and 1995, conducted by the country's Census Bureau. The official name of the data is "Census-Income (KDD) Data Set" and it was downloaded from the Machine Learning Repository link:

<http://archive.ics.uci.edu/ml/datasets/Census-Income+%28KDD%29>

First, **we should point that we have not used or processed this dataset using techniques similar to those required and used in the following pages in another lesson of the Master in Business Analytics.**

Using this dataset we wanted to answer different interesting questions, such as what attributes lead to high income and what attributes to smaller income? In order to find the answer we implemented two classification technics, i.e. decision tree and Naïve Bayes, as well as association rules and logistic regression. Comparing the results from the 4 different approaches we will reach more solid conclusions about the factors that affect the income level.

Another interesting analysis, is the clustering analysis which aims to find different patterns of citizens in four different categories of data. Demographic, household composition, socio-economic and employment data assist four different analyses that may be interesting for different types of analysts, such as advertisers (demographic data), social analysts (socio-economic data), companies that look where to build new offices (employment data) etc.

In order to implement the above mentioned algorithms we also used different tools. Each one of them has its own characteristics, but let's us become familiar with multi-tool analyses as well as different needs in the data preparation phase. In any case, it is helpful to learn how to produce and evaluate results from different platforms, broadening thus our knowledge in the data mining area.

Finally, in the last section of this deliverable, we conclude with the most important and useful results from our analyses. At the end, we attach additional information in the appendix so as to accommodate analyses of other analysts.

## Data cleansing

Having downloaded the data, we had to proceed with cleansing processes. First of all, we tried to open the data via Microsoft Excel in order to see how the data are stored and make some first modifications. The .data file contained only the data without the headers, while the headers were only stored in the names file. So, we had to manually match each one of the column names with the respective columns. Making the first descriptive statistics about the dataset, we found that we have 299.285 rows with 41 attributes. Removing duplicate rows we saw that the unique rows are 292.550 (removed 2.25%) with 7 continuous and 34 categorical attributes (41 in total).

Proceeding with data cleansing, we looked for the columns that add value to the dataset. The attribute "instance weight" represents the number of citizens that have the characteristics described in the respective row. Then, the "predicted income" is an attribute with two different values '-50000' and '+50000' (which represent low and high income) and will be used for future classification technics, since we will identify the attributes that lead to high and low income.

Next step is to examine all the columns for the existence of empty fields, i.e. the appearance of the symbol "?", which symbolizes the empty cells. It is observed that the question marks appear only in 8 fields ([state of previous residence], [migration code-change in msa], [migration code-change in reg], [migration code-move within reg], [migration prev res in sunbelt ], [country of birth father], [country of birth mother], [country of birth self]). Therefore, we had to find how many records have blank values in at least one of these columns. The code that does this is shown in the following figure.

It appears that 153.247 rows have at least one space in these columns. Since they are so many we must decide whether to delete the rows with empty fields or the respective columns. In order to conclude to a decision, we take into account the percentage of records that have empty fields in each of the 8 columns separately. The results are as follows:

Attribute	% of missing values
state of previous residence	1034 / 292550 = 0,35%
migration code-change in msa	146141 / 292550 = 49,95%
migration code-change in reg	146141 / 292550 = 49,95%
migration code-move within reg	146141 / 292550 = 49,95%
migration prev res in sunbelt	146141 / 292550 = 49,95%
country of birth father	10120 / 292550 = 3,46%
country of birth mother	9169 / 292550 = 3,13%
country of birth self	5150 / 292550 = 1,76%

As a result, we chose to delete columns [migration code-change in msa], [migration code-change in reg], [migration code-move within reg] and [migration prev res in sunbelt], as for these columns half of rows are empty prices and do not provide us useful information. The remaining columns with empty values remain in the dataset, but the question mark is converted to "NA" to indicate that there is no value, while avoiding problems with algorithms that cannot handle empty values.

Next, it is important to determine meaning of the value "Not in universe". The "universe" is the population that may have an answer for a variable. In most cases, they are citizens to whom are the census questions addressed. For example, children are not asked about labor issues, while men are not asked about maternity issues. All these cases (citizens) that have not reason to give an answer for a variable belong to the category "Not in universe" and thus labeled respectively. Therefore, usage of this value has meaning in our dataset and is not subject to changes.

Last step of the cleansing process of the dataset is to check the type of data of each one of the attributes. Some attributes were identified by Excel right, but in some case we changed the type from General to Number. However, more changes will be done for the needs of each algorithm. The final clean dataset has 292.550 rows with 37 columns.

## Data Transformation

Here is the set of variables that we are going to use in order to do some of the data mining technics. The choice of these variables was made by common sense and of what we thought would be more relevant in order to create association rules about the income and clustering on different subsets.

- age – The age of the individual
- class of worker – Whether they are government, military, private, and so on.
- education – The highest level of education achieved for that individual
- marital – Marital status of the individual
- occupation – The occupation of the individual
- race – descriptions of the individuals race. Black, White, Eskimo, and so on
- sex – Biological Sex
- capital\_gain – Capital gains recorded
- capital\_loss – Capital Losses recorded
- country of birth self – Country of origin for person
- income – Whether or not the person makes more than \ \$50,000 per annum income.

Now, we begin with the data preparation. All the variables that were stored as text, and some that were stored as integers, should be converted to factors during the data import. Because we're going to be modifying the text directly, we need to convert them to character strings. We do this for all the text variables we intend to work with.

```
data$class.of.worker = as.character(data$class.of.worker)
data$major.occupation.code = as.character(data$major.occupation.code)
```

Because some groups in each variable represent a larger group we thought it would be a good idea to put together some groups in order to reduce the general number of groups in respect of the given information of the data.

In specific, the values of the variable “class of worker” that are “Never worked”, “Not in universe” and “Without-Pay” are both very small groups, and they are likely very similar. So we can combine them to form a “Not Working” more general category. In a similar vein, we can combine government employee categories, and self-employed categories. This allows us to significantly reduce the number of categories.

```
data$class.of.worker = gsub("^ Federal government", "Federal-
Govt", data$class.of.worker)
data$class.of.worker = gsub("^ Local government", "Other-
Govt", data$class.of.worker)
data$class.of.worker = gsub("^ State government", "Other-
Govt", data$class.of.worker)
data$class.of.worker = gsub("^ Private", "Private", data$class.of.worker)
```

On occupation, a simple way to group the categories would include blue collar versus white collar. We separate out service industry, and other occupations that we didn't see fitting well with the other groups into their own group. It's unfortunate that Armed Forces won't fit well with any of the other groups. In order to get it properly represented, we can try up-sampling it when we will train the model.

```
data$major.occupation.code = gsub("^ Adm support including
clerical", "Admin", data$major.occupation.code)
data$major.occupation.code = gsub("^ Armed
Forces", "Military", data$major.occupation.code)
```

```
data$major.occupation.code = gsub("^ Precision production craft &
repair", "Blue-Collar", data$major.occupation.code)
```

The variable country presents a small problem. Obviously, the United States represent the vast majority of observations, but some of the groups have such small numbers that their contributions to the model might not be significant. A way around this, would be to group the countries.

```
data$country.of.birth.self[data$country.of.birth.self==" Cambodia"] = "SE-
Asia"
data$country.of.birth.self[data$country.of.birth.self==" Canada"] = "British-
Commonwealth"
data$country.of.birth.self[data$country.of.birth.self==" China"] = "China"
```

We tried to use a combination of geographical location, political organization, and economic zones. “Euro\_1” includes countries within the Eurozone that we considered more affluent, and therefore people from these countries are probably going to be more affluent, too. “Euro\_2” includes countries within the Eurozone that we considered less affluent. These includes countries that are financially troubled like Spain and Portugal, but also the Slavic countries and those formerly influenced by the USSR like Poland. Formerly British holdings that are still closely economically aligned with Britain are included under the British-Commonwealth.

We should group the education variable, as well. Ultimately, the goal is to shave down the number of categories in the categorical variables. For some methods this vastly simplifies the calculations, as well as makes the output more readable. We choose to group all the dropouts together. Additionally, we group high school graduates and those that attended some college studies without receiving a degree as another group. Those college graduates who receive an associates are grouped together regardless of the type of associates. Those who graduated college with a Bachelors, and those who went on to graduate school without receiving a degree are grouped as another group. Mostly, everything thereafter is separated into its own group.

```
data$education = gsub("^ 10th grade", "Dropout", data$education)
data$education = gsub("^ 11th grade", "Dropout", data$education)
data$education = gsub("^ 12th grade no diploma", "Dropout", data$education)

data$marital.status[data$marital.status==" Never married"] = "Never-Married"
data$marital.status[data$marital.status==" Married-A F spouse present"] =
"Married"
data$marital.status[data$marital.status==" Married-civilian spouse present"] =
"Married"

data$race[data$race==" Amer Indian Aleut or Eskimo"] = "Amer-Indian"
data$race[data$race==" Asian or Pacific Islander"] = "Asian"
```

Some changes we made aimed simply to make the variable names more readable or easier to type. We chose to block “spouse absent”, “separated”, and “divorced” values all together as “not married”, since some initial data mining suggested they were similar in respect to income.

```
data[[ "capital.gains"]] = ordered(cut(data[[ "capital.gains"]], c(-
Inf, 0, median(data[[ "capital.gains"]][data[[ "capital.gains"]]>0]), Inf)),
labels = c("None", "Low", "High"))
```

```
data[[ "capital.losses"]] = ordered(cut(data[[ "capital.losses"]], c(-Inf,0,
median(data[[ "capital.losses"]][data[[ "capital.losses"]]>0]), Inf)), labels
= c("None", "Low", "High"))
```

Finally, we chose to block capital gains and losses together, rather than do a transformation. Both variables are heavily skewed to the point that we think a numerical transformation would not have been appropriate. So we choose to block them into “None”, “Low”, and “High”. For both variables, “None” means they aren’t present on the market. “Low” means they have some investments, while “high” means they have significant investments. Both gains and losses are positively associated with higher income, because if you have money in the market, odds are you have money to begin with.

## Data Mining Technics

### Prediction of income level

As mentioned before, the predicted income takes two values ‘-50000’ and ‘+50000’, which represent low and high income respectively. So, the first meaningful question we want to answer is what attributes lead to higher income and what to lower income? Alternatively, the main objective of the analysis is to find the characteristics of "rich people". To ensure the reliability of results we use both Classification technics and Association Rules, as described below. The findings are compared in order to identify the best model for our data.

### Classification

For classification technics, we used the Microsoft SQL Analysis Services, where we implemented the Decision Tree and Naïve Bayes algorithms. However, in order to mine the data in the Analysis services, first we need to import the data to Microsoft SQL Server. The first step is to save the clean (not the transformed) dataset as a .csv file. Then we create a new database named “Census” and using the import tool in SQL Server, we import the dataset as a table with the name [Data Mining]. We see that there is no column that be a primary key and thus we create a new column named id, which is unique for each row.

Having a table with the needed data in the database in SQL Server, we move to Analysis Services and create a new Analysis Services Multidimensional and Data Mining Project titled "Data Mining". In the new project we created a new "Data Source" (the Census database) and a new "Data Source View". In particular, as "Data Source View" we used the new table [data\_mining], we created earlier in SQL Server.

Next, we create a new Data Mining Structure, the Decision Tree and select the table which stores our data. As a next step, we need to choose the attributes which will be used in the model. In particular we define the "Predictable" attribute (what would be predicted by the model), the "Input" attributes of the model and the "Key" attribute (id). After selecting the "Predictable", Analysis Services enables proposals of attributes that are highly correlated with "Predictable". With the button "Suggest" window appears, as shown in the next image. The attributes that are correlated over 95% with the “Predictable” are automatically entered as "Input" in the model.

It is important to mention how attributes were selected:

- As a Key Attribute, the table id is selected.
- As Predictable Attribute we selected "predicted income" which takes the values '+50.000' and '-50000'. It shows the annual income of the citizen with the characteristics of the row.
- Finally, as Input we set all the features that have a high correlation with the "Predictable".
- The remaining attributes are imported to the model (option in the left column of the image below).

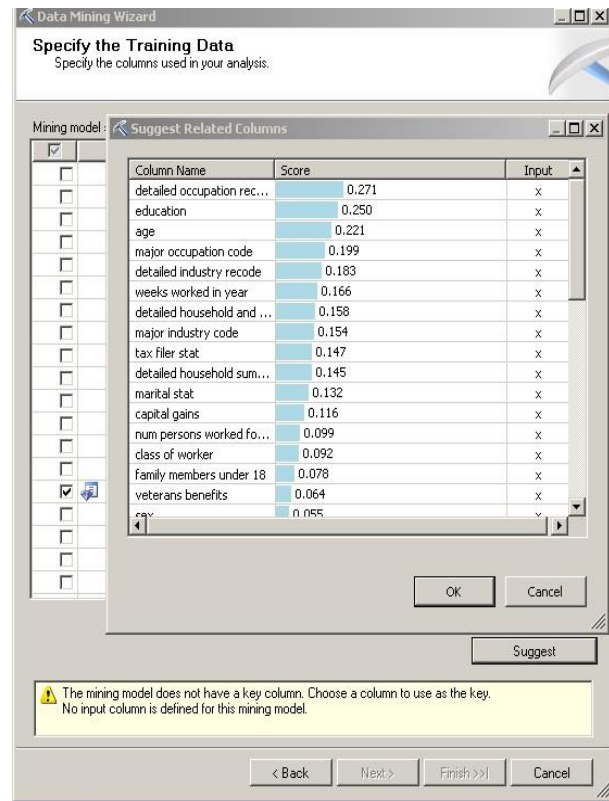


Figure 1 Suggest related columns

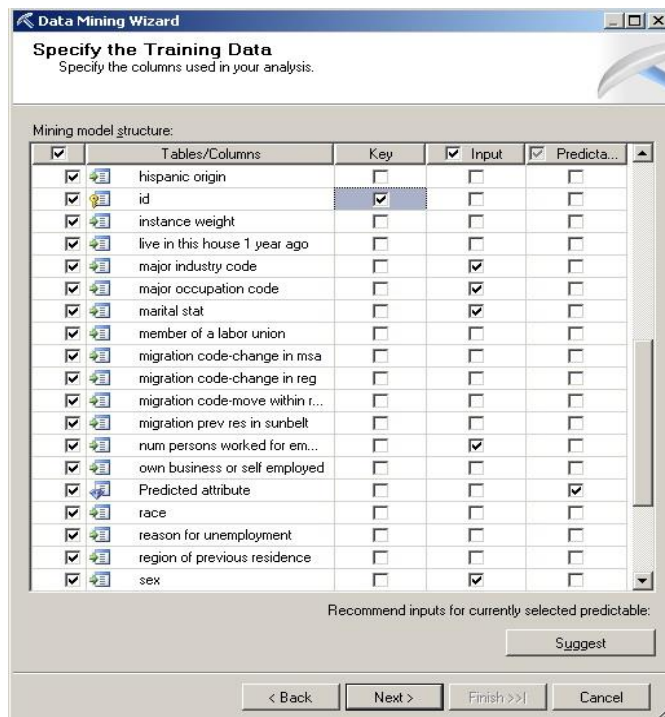
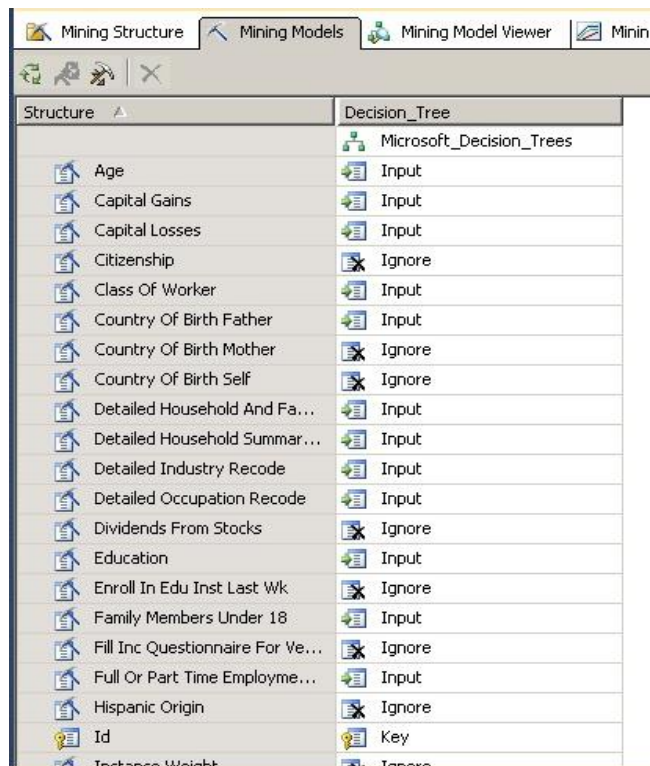


Figure 2 Specify Key, Input and Predictable



Then the Data Mining Wizard identifies the type of each attribute (whether it is continuous or discrete variable, and what type is). It also provides the Detect button which automatically identifies the discrete attributes, deriving this information from Microsoft SQL Server. Next, we define the ration of the Training and Testing Set. By default it is 70%/30%, which we do not change. Upon completion of the process, we name the Mining Structure, as well as the Mining Model and select the "Allow drill through" for future exploration of the results.

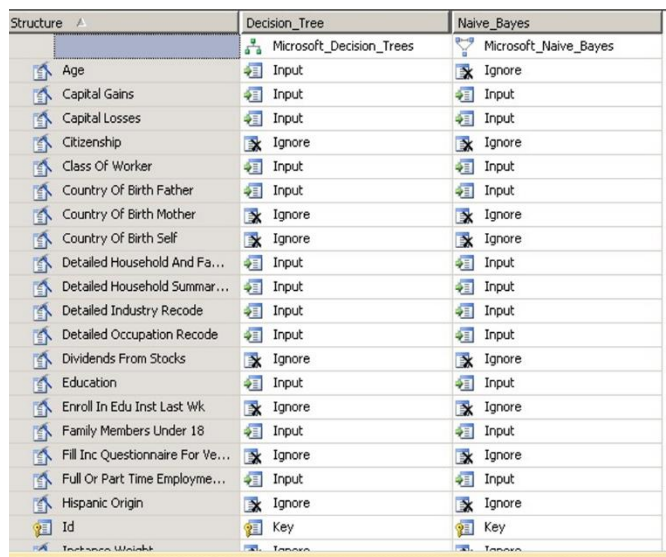
So, we have created the mining model as shown on the picture on the right. Then, we add a new mining model, Naïve Bayes, which informs us that it cannot process continuous data and thus it will remove them from the model. At the end, we have one mining structure with two mining models, as shown below.



Structure	Decision_Tree
	Microsoft_Decision_Trees
Age	Input
Capital Gains	Input
Capital Losses	Input
Citizenship	Ignore
Class Of Worker	Input
Country Of Birth Father	Input
Country Of Birth Mother	Ignore
Country Of Birth Self	Ignore
Detailed Household And Fa...	Input
Detailed Household Summar...	Input
Detailed Industry Recode	Input
Detailed Occupation Recode	Input
Dividends From Stocks	Ignore
Education	Input
Enroll In Edu Inst Last Wk	Ignore
Family Members Under 18	Input
Fill Inc Questionnaire For Ve...	Ignore
Full Or Part Time Employme...	Input
Hispanic Origin	Ignore
Id	Key
Income Weight	Ignore

Figure 3 Mining Models tab

Finally, we deploy the model and proceed the results' examination.



Structure	Decision_Tree	Naive_Bayes
	Microsoft_Decision_Trees	Microsoft_Naive_Bayes
Age	Input	Ignore
Capital Gains	Input	Input
Capital Losses	Input	Input
Citizenship	Ignore	Ignore
Class Of Worker	Input	Input
Country Of Birth Father	Input	Input
Country Of Birth Mother	Ignore	Ignore
Country Of Birth Self	Ignore	Ignore
Detailed Household And Fa...	Input	Input
Detailed Household Summar...	Input	Input
Detailed Industry Recode	Input	Input
Detailed Occupation Recode	Input	Input
Dividends From Stocks	Ignore	Ignore
Education	Input	Input
Enroll In Edu Inst Last Wk	Ignore	Ignore
Family Members Under 18	Input	Input
Fill Inc Questionnaire For Ve...	Ignore	Ignore
Full Or Part Time Employme...	Input	Input
Hispanic Origin	Ignore	Ignore
Id	Key	Key
Income Weight	Ignore	Ignore

Figure 4 Final "Mining Models" tab

## Decision Tree

The Decision tree consists of several branches separated in different nodes. All branches start from the central node in the left part of the tree that contains all data. Studying the tree from left to right, data divide themselves because some features take different values. The first split in the tree root is the most

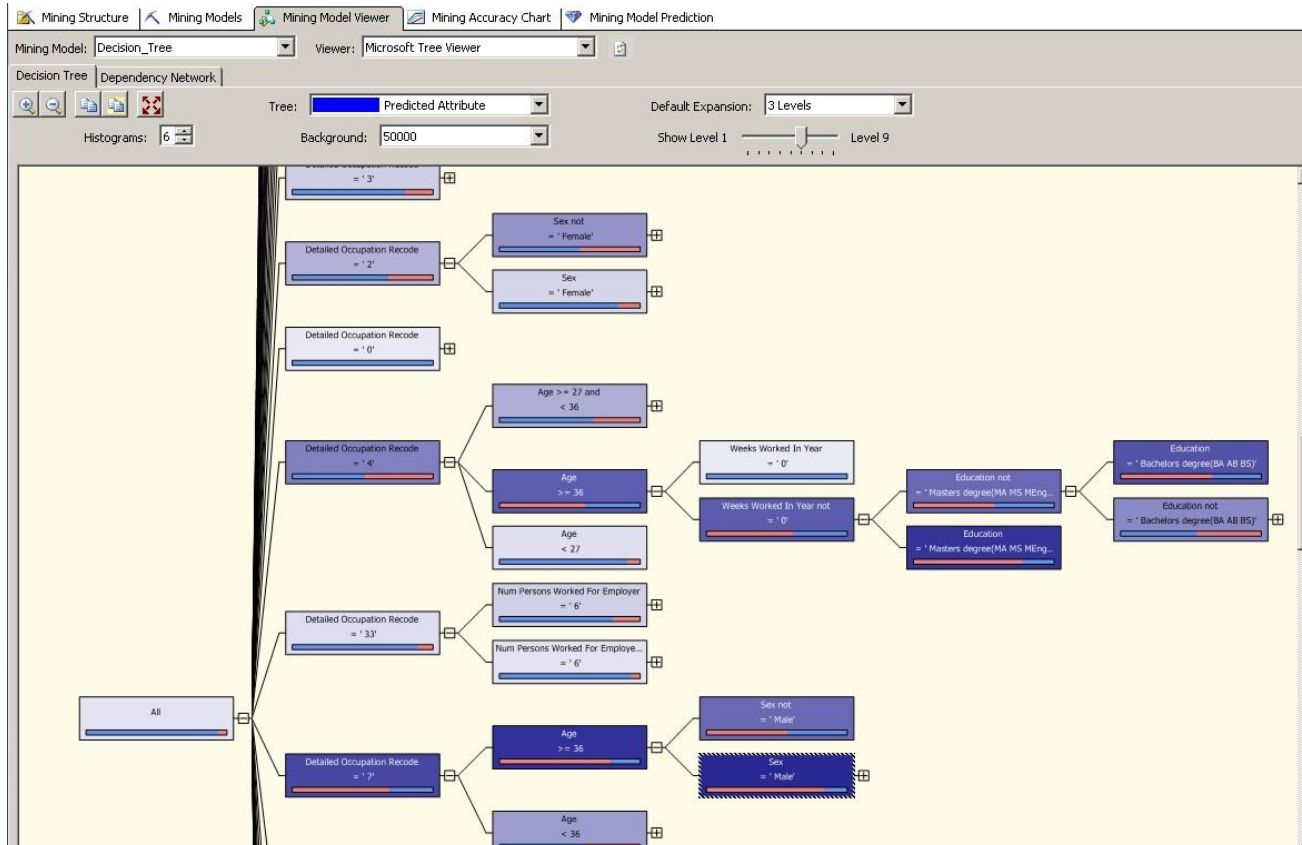


Figure 5 Part of the Decision Tree

important as it split has the biggest information gain (i.e. gives the most valuable information about the data).

The study of data can the analyst "group" or "open" a branch in order to hide or display information respectively. Also you can choose how many levels wants to "open" the tree. In this analysis it was chosen that the tree has initially three levels of splits, and then some specific branches "opened" in order to find more specific information. Since the tree that was built is extremely large, we show only a part of it, which collects though very interesting information.

As shown in the above decision tree (figure 5) it was chosen to highlight the branches that lead to people with incomes greater than \$ 50.000. We can conclude that people have large incomes when they work in positions "Detailed Occupation Recode" = '4', "Age" >= '36' and work at least one week a year. Out of these citizens, most of them have master's degree, while those who do not, have at least an undergraduate degree. So, this is the profile of the "rich" citizens.

Moreover, income of over \$ 50.000 have citizens working in positions "Detailed Occupation Recode" = '7', aged over 36 and that are men. This racial difference shows wage discrimination between the two sexes

that prevailed at that time (1994, 1995). This means that workers in the same location and with the same age had different salaries because of their gender.

Similarly are the other branches read, which are not visible in the figure. Of course the other branches show obvious relations, suggesting that people with "Capital gains" other than zero have higher annual incomes.

Since during construction of the structure it was chosen to enable drill through in the models, we can see the exact percentages of data distribution in each node. For example, selecting the node with "Detailed Occupation Recode" = '7', age over 36 and sex = 'Male' we can see a small window with the following rates.

As the picture on the right shows, 83.96% people with these characteristics have annual earnings of more than \$ 50.000 and the remaining 16.04% under \$ 50.000. It seems that there are no empty values for this attribute as the missing values equal 0%. In the same way one can see the exact percentages of each node and create histograms, pies etc. to visualize the data.

However, this analysis aims to find the main characteristics of citizens with high incomes. Until now we noticed that job, age, educational level and gender significantly affect income. But we can better illustrate the correlation between the characteristics and the target attribute through the Dependency Network.

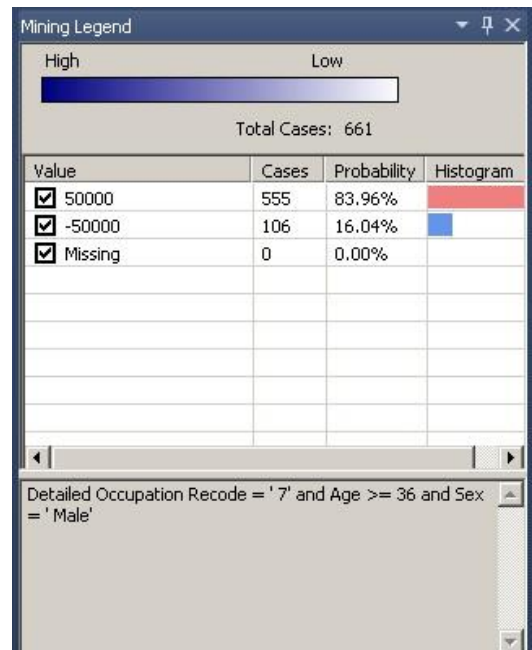


Figure 6 Drill through in a node

The Dependency Network displays the dependencies between the attributes that are the input of the model and the predictable attribute (here the annual income level named "Predicted income"). To the left of the network is a filter where the analyst may specify how many connections he/she wants to portray. Starting from the most powerful connection we can visualize all connections to the weakest.

The network of the following figure is set to show the 5 most powerful outfits. The 5 most important attributes that can predict the annual income is in the order of importance:

- Detailed occupation recode
- Education
- Sex
- Age
- Number of Persons worked for Employer

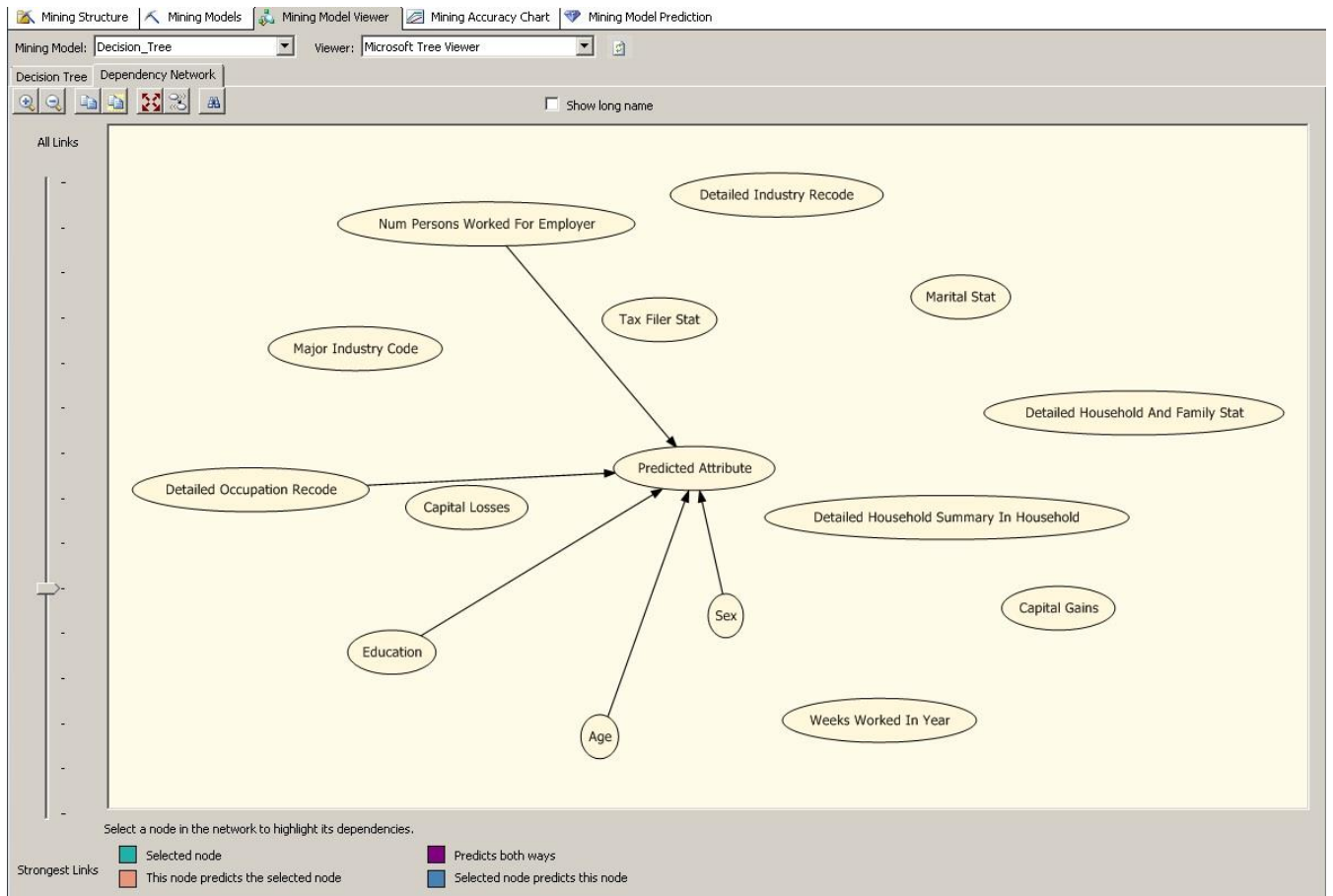


Figure 7 Dependency Network

We note that job position, education level, age and gender are once associated with the predicted attribute (income). Next, we study the results of the Naïve Bayes algorithm in order to compare them with the previous ones and extract more reliable conclusions.'

## Naïve Bayes

The first results tab of the algorithm "Naïve Bayes" is the Dependency Network, which was also analyzed in the Decision Tree. The new network the algorithm extracted is as follows in figure 8.

The most important attributes in order of importance are:

- Detailed occupation recode
- Education
- Major occupation recode
- Weeks worked in Year
- Detailed industry recode

The next tab is "Attribute Profiles" which shows how many times each attribute appears in the population and in each value of the "Predicted income" (i.e. when "Predicted income" equals +\$50.000 or -\$50.000).

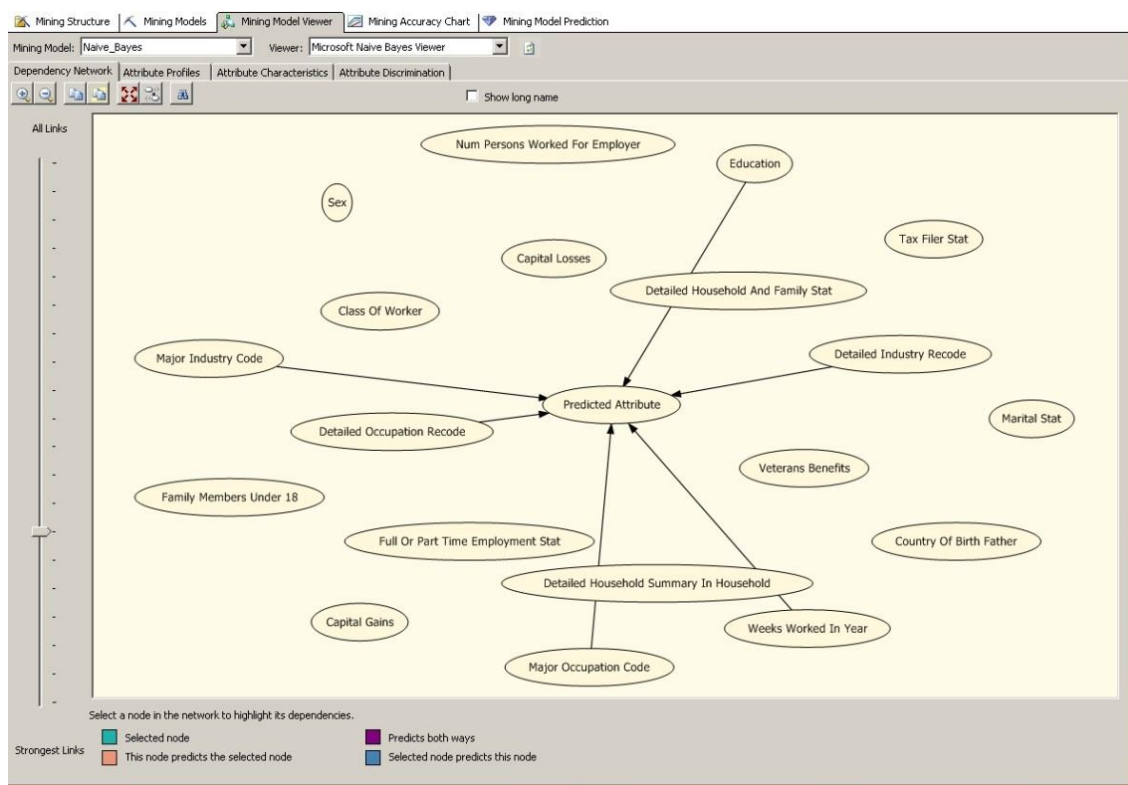


Figure 8 Dependency Network

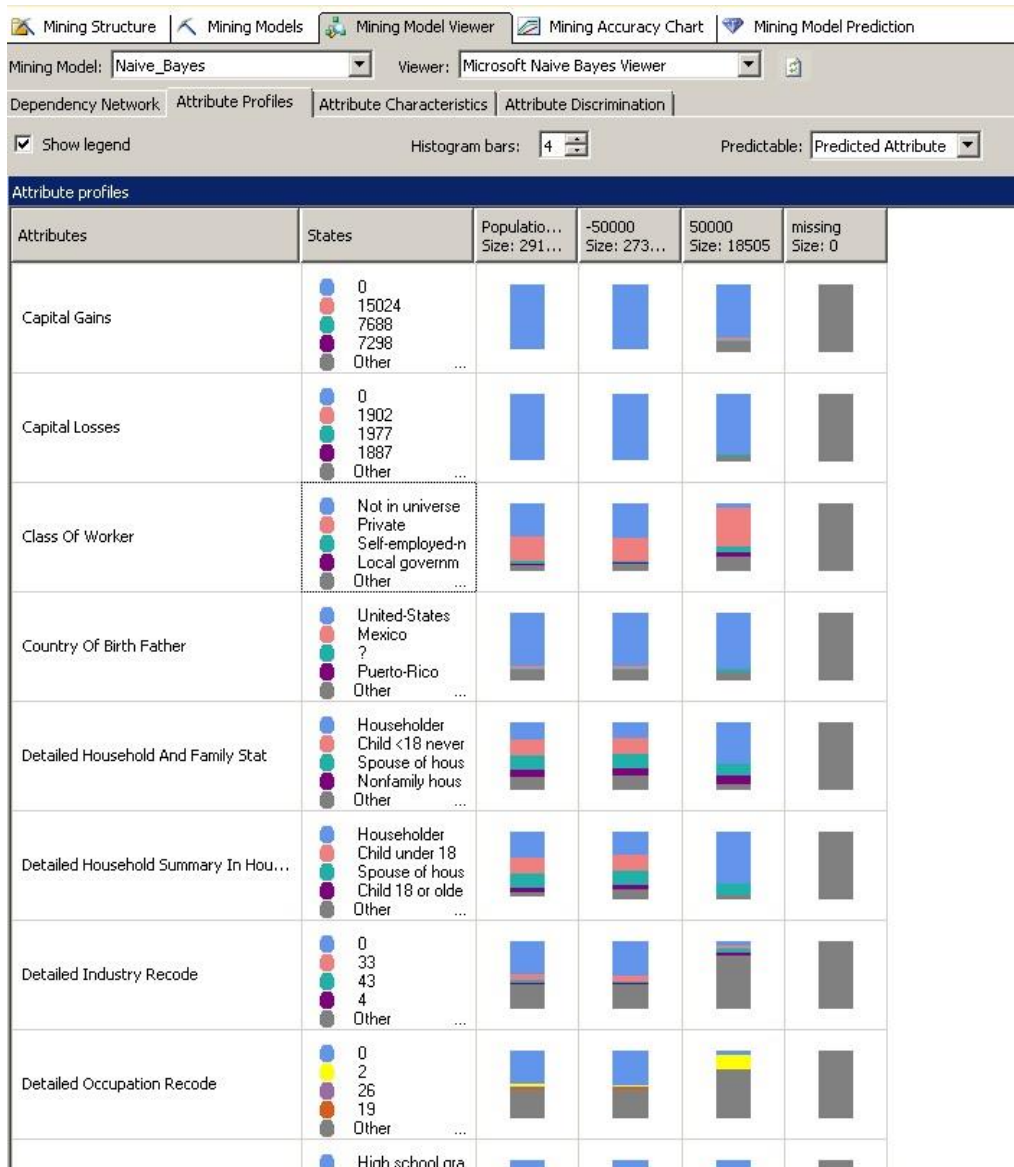


Figure 9 Attribute Profiles

The next tab is "Attribute Characteristics" which drills down to each alternative of the "Predicted income" attribute. We may select the value that we want to examine and the tab presents the features that are most relevant to it in order of importance. Comparing the results for different values of "Predicted income" we can see what results in every situation. The next two figures show the results for "Predicted income" = 50,000 and "Predicted income" = -50,000.



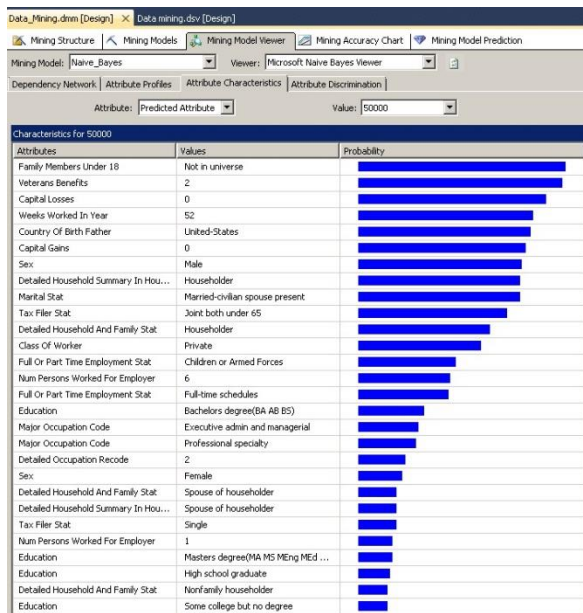


Figure 10 Predicted income = 50.000

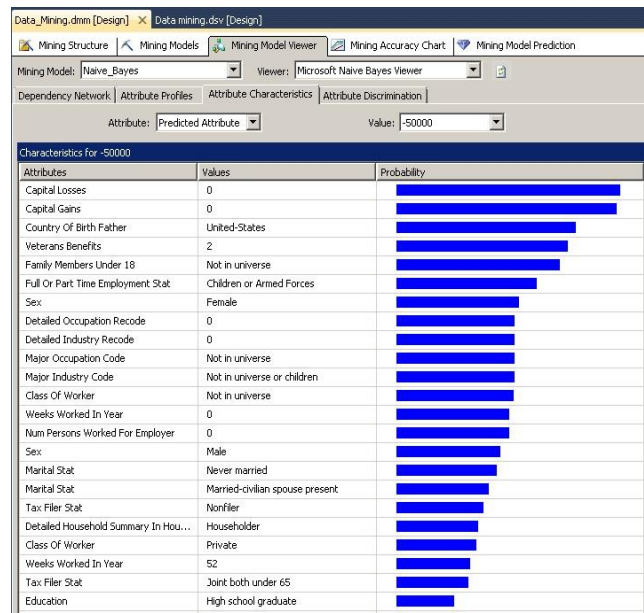


Figure 11 Predicted income = -50.000

However, the comparison becomes more understandable via the tab "Attribute Discrimination". In "Value 1" we choose to view features for the price of +50.000 and in "Value 2" the features for the price -50.000. Then we may see how attributes are associated with the two values (in order of importance), which attribute is associated with which value, while the size of the bar to show how big the correlation.

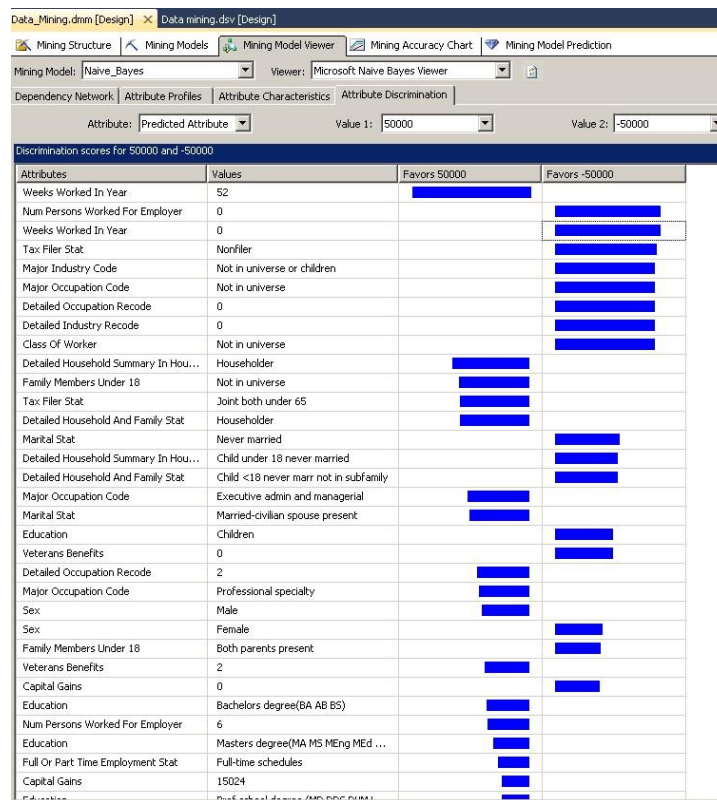


Figure 12 Attributes Discrimination

The above figure shows that citizens have annual earnings above \$ 50.000 when they work 52 weeks a year, are the main holders of the household, have no members in the family under 18 and all the members are registered in the tax records. On the other hand, earnings are less than \$ 50.000 when there are no working members in the household, the person does not complete tax report and does not hold a professional job position. So, it is concluded that how much time someone devotes working affects significantly the salary level, but the other attributes of the dependency network are not high in the ranking.

### Association Rules

On the contrary to Classification, we chose to implement Association rules through the free software R. Before we are ready to start data mining in our dataset we need to convert the variables age and income in order to be able to calculate rules. Income is a variable with only two values : <50000 and >50000. So we will convert it to small and large income respectively. The “Age” has a range 1:100 and thus we are going to break it in groups so it would be easier to identify the rules later.

```
data[["age"]] = ordered(cut(data[["age"]], c(-1,25,45,65,100)), labels =
c("Young", "Middle-aged", "Senior", "Old"))

data[["predicted_income"]]= ordered(ifelse(data[["predicted_income"]]== "
50000+.",50000,-50000))

levels(data$predicted_income)[1] = "small"
levels(data$predicted_income)[2] = "large"
```

Now that the data are ready, we need to transform the whole dataset in a format that is going to be used to generate rules. For this purpose we are going to use the “arules” package. Next, we apply the R function *apriori*. This function calculates and screens the support, the confidence and the lift of items. Consequents in *apriori* are single items (columns), but antecedents can be either single items or groups of items (item sets). Here we use the default specification for two of *apriori*’s parameters, and specify minlen (an integer value for the minimal number of items per item set) as 1 and maxlen (an integer value for the maximal number of items per item set) as 10. This means that *apriori* searches over all item sets for which the sum of the number of items in the LHS item set and the (single) item on the RHS is between 1 and 10. This involves quite a lot of computations and queries. In this particular application we consider only those LHS and RHS item sets for which the support is at least 0.005 and the confidence is at least 0.60. Finally, we search over a subset of all rules (with support greater than 0.01 and confidence greater than 0.60, as specified previously) that have small income on the RHS and achieve a lift larger than 1.2. The combinations that satisfy these conditions are the ones that are able to predict small income earners from the coded explanatory information. Similarly, we search and list all rules that identify high income earners. Next, we present two graphs with the association rules for each income category.

```
new_data = as(new_data,"transactions")
new_data :
transactions in sparse format with
 292550 transactions (rows) and
 56 items (columns)

rulesIncomeSmall <- subset(rules, subset = rhs %in% "predicted_income=small" &
lift > 1.05)
rulesIncomeLarge <- subset(rules, subset = rhs %in% "predicted_income=large" &
lift > 1.2)
```



Graph for 25 rules

size: support (0.005 - 0.011)  
color: lift (9.733 - 11.164)

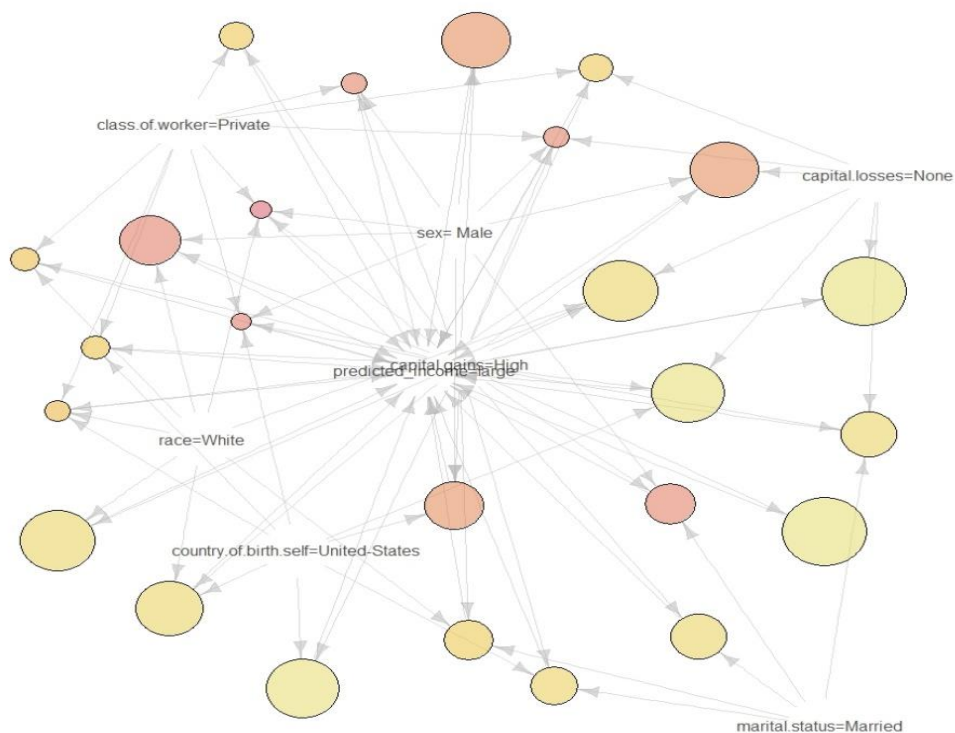


Figure 13 Association rules for High Income

Graph for 25 rules

size: support (0.005 - 0.489)  
color: lift (1.051 - 1.067)

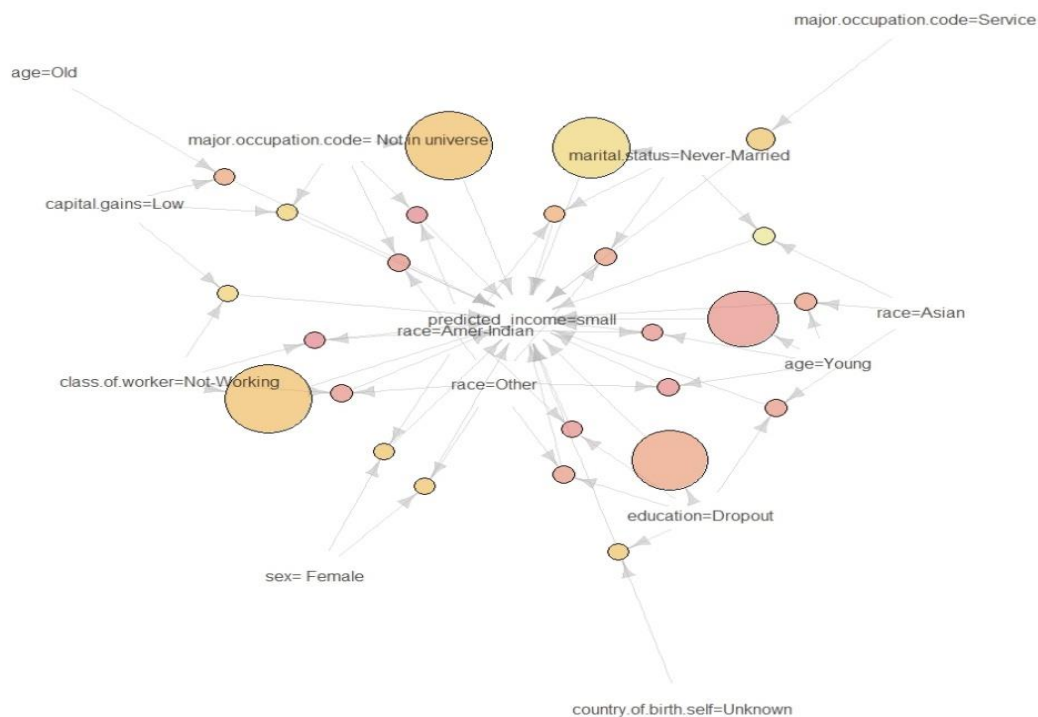


Figure 14 Association rules for Small Income

By looking at the rules that were generated from the R commands and at the plots that are shown above, we infer some very interesting insights. Associations for the large income are much easier because the rules are fewer. In other words, in order to be rich you have to belong in a very specific pattern. This is also confirmed by the lift of the rules which is almost 11 (i.e. if you have these specific characteristics you are 11 times more possible to be rich than a random person). Below we can see the top 10 association rules about the large income:

Left hand side	Right hand side	Support	Confidence	Lift
{class of worker=Private, race=White, sex= Male, capital gains=High}	{predicted_income=large}	0.0052	0.7086	11.1639
{class of worker=Private, race=White, sex= Male, capital gains=High, capital losses=None}	{predicted_income=large}	0.0052	0.7086	11.1639
{marital status=Married, race=White, sex= Male, capital gains=High, country of birth self=United-States}	{predicted_income=large}	0.0070	0.7042	11.0945
{marital status=Married, race=White, sex= Male, capital gains=High, capital losses=None, country of birth self=United-States}	{predicted_income=large}	0.0070	0.7042	11.0945
{marital status=Married, race=White, sex= Male, capital gains=High}	{predicted_income=large}	0.0075	0.7032	11.0794
{marital status=Married, race=White, sex= Male, capital gains=High, capital losses=None}	{predicted_income=large}	0.0075	0.7032	11.0794
{class of worker=Private, sex= Male, capital gains=High, country of birth self=United-States}	{predicted_income=large}	0.0051	0.6961	10.9682
{class of worker=Private, sex= Male, capital gains=High, capital losses=None, country of birth self=United-States}	{predicted_income=large}	0.0051	0.6961	10.9682
{race=White, sex= Male, capital gains=High}	{predicted_income=large}	0.0091	0.6954	10.9570
{race=White, sex= Male, capital gains=High, capital losses=None}	{predicted_income=large}	0.0091	0.6954	10.9570

If we want to create a profile about the citizens with large income, we can conclude to a person who is white, male, working in the private section, with high capital gains and that comes from United States. According to the high lifts, such persons are more probable to have a large income as derived from the available data.

On the other hand, when we are talking about small income the situation is different. We can see that the maximum lift is very close to 1. This means that given the conditions of the rule, it does not necessarily mean that the citizen is more possible to be poor than a random person. And in real life this is the truth. A certain pattern can give clues about small income but we cannot be sure that this will be correct. Below we can see the top 10 association rules about the small income:

Left hand side	Right hand side	Support	Confidence	Lift
{education=Dropout, marital status=Never-Married, race=Amer-Indian}	{predicted_income=small}	0.0051	1.0000	1.0678
{education=Dropout, major occupation code= Not in universe, race=Amer-Indian}	{predicted_income=small}	0.0056	1.0000	1.0678
{class of worker=Not-Working, education=Dropout, race=Amer-Indian}	{predicted_income=small}	0.0056	1.0000	1.0678
{marital status=Never-Married, major occupation code= Not in universe, race=Amer-Indian}	{predicted_income=small}	0.0050	1.0000	1.0678
{class of worker=Not-Working, marital status=Never-Married, race=Amer-Indian}	{predicted_income=small}	0.0050	1.0000	1.0678

{education=Dropout, marital status=Never-Married, race=Other}	{predicted_income=small}	<b>0.0088</b>	1.0000	1.0678
{education=Dropout, major occupation code= Not in universe, race=Other}	{predicted_income=small}	<b>0.0093</b>	1.0000	1.0678
{class of worker=Not-Working, education=Dropout, race=Other}	{predicted_income=small}	0.00927021	1	1.06777088
{age=Young, education=Dropout, major occupation code= Not in universe}	{predicted_income=small}	0.25990087	1	1.06777088
{age=Young, class of worker=Not-Working, education=Dropout}	{predicted_income=small}	0.26003760	1	1.06777088

As we did with the high income, we may create a profile of the citizens with small income, too. However, here the things are quite different. Many rules have the same lift with different attributes for each rule. Additionally, this is also explained from real life because many things can lead to a small income but having those does not necessarily mean that you would have also small income. For our investigation, people who haven't got any education degree, are American-Indian and never got married are more probable to have small income (i.e. young people or foreigners etc.). Furthermore, attributes such as the age, are very reasonable to be related with small income as young people are obviously not working, and thus have small income now. However, this does not mean that they will remain with the same small income in the future, too.

## Clustering

The next data mining technic we will use is the Clustering, which will be implemented using RapidMiner. Clustering is a data mining method that analyzes a given data set and organizes it based on similar attributes. Clustering can be performed with pretty much any type of organized or semi-organized data set, including text, documents, number sets, census or demographic data, etc. The core concept is the cluster, which is a grouping of similar objects. Clusters can be any size – theoretically, a cluster can have zero objects within it, or the entire data set may be so similar that every object falls into the same cluster. This would be rare: most often, objects will naturally cluster due to mathematic and statistical similarities. In the case of text analytics, objects will often cluster due to keywords, phrasing, and subject/context.

RapidMiner (formerly "Yale") is an environment for machine learning and data mining experiments. It allows experiments to be made up of a large number of arbitrarily nestable operators, described in XML files (PMML) which are created with RapidMiner's graphical user interface. RapidMiner is used for both research and real-world data mining tasks. The initial version has been developed by the Artificial Intelligence Unit of University of Dortmund since 2001. The Community Edition of RapidMiner is distributed under the AGPL license. RapidMiner provides more than 500 operators for all main machine learning procedures, including input, output, data preprocessing and visualization. It is written in Java and therefore can work on all popular operating systems. It also integrates learning schemes and attributes evaluators of the Weka learning environment.

Census data is a fundamental source of information in numerous research areas. Because of this, algorithms used by geographers for clustering, should be capable of dealing with specific problems associated with the use of census data. For the purpose of this work we would like to highlight the problems which result from dealing with high dimensional datasets and which may have measurement errors. Additionally, in census datasets one should expect variations in size and homogeneity in the

geographical units and also non-stationary in the relations between variables, which are bound to change across regions. All these problems concur to the complexity which is involved in clustering census data. Emphasis should be put on the importance of using robust clustering algorithms, algorithms which, as much as possible, should be insensitive to the presence of outliers. Closely related with robustness is the capability of modeling locally, preserving the impact of errors and inaccuracies in data within local structures of the clustering, rather than allowing these problems to have a global impact on the results. The idea is to find algorithms which degrade progressively in the presence of outliers instead of abruptly disrupting the clustering structure. Improvements in clustering algorithms will yield benefits in all research areas which use census data as part of their analysis process. Although the performance of the clustering methods in itself is not enough to solve all the problems related with the quality of census based on clustering, it is definitely a relevant issue.

The common procedure of clustering census data includes the following 5 steps:

1. Definition of the clustering objective;  
The clustering objective for this assignment is to find out interesting facts about the census data.
2. Careful choice of the variables to use;  
We had to exclude some variables in each of the four clustering processes in order to be able to draw meaningful results.
3. Normalizing the data;  
We had to group some variables (such as age, country) with many levels in order to achieve findings describing certain groups of the data. Apart from that we transformed the variables kept to binary form for better result interpretation.
4. Clustering the data;  
The clustering process with the Rapidminer tool.
5. Labeling, interpretation and evaluation;  
We presented the results in table format and some interesting data were presented.

Here we concentrate on step 4, the clustering algorithm that should be used to achieve the desired data reduction. In this assignment, the clustering algorithm we will use is the K-means algorithm. The k-means is widely known and used and thus only a brief outline of the algorithm is presented. K-means is an iterative procedure, to place cluster centers, which quickly converges to a local minimum of its objective function. This objective function is sum of the squared Euclidean distance (L2) between each data point and its nearest cluster center this is also known as “square error distortion”. It has been shown that k-means is basically a gradient algorithm which justifies the convergence properties of the algorithm.

The workflow process for each one of the 4 clustering processes is:

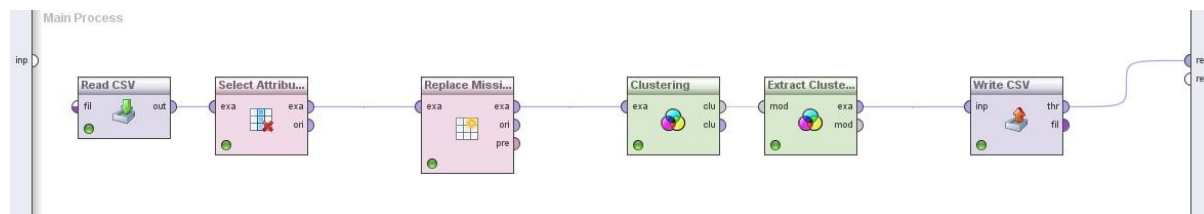


Figure 15 Clustering with k-means

The full results of the clustering processes (as xml code) are attached in the Appendix.

In order to complete meaningful analyses we split the data (39 Census variable) into 4 categories and then performed clustering for each one of them. For each clustering attempt we present the most important results about the clusters that are interesting for further examination. The complete tables with the results about the four clustering attempts are attached in the appendix. In general, different sub-datasets may be interesting for pattern discovering from advertisers, social analysts, etc.

### *Demographic attributes*

*(Includes - age, race, sex, country of birth self, citizenship, education, capital gains)*

The first clustering, implemented on the demographic data makes us observe many interesting things. For starters, we got 10 different clusters with unique characteristics among them. In specific, Cluster 0 contains mainly middle-aged white men who are born in the United States of America with a US citizenship and High School graduates with none capital gains.

Other interesting conclusions risen in the results include the ones related to cluster 1 and cluster 2. In cluster 1 we identify women born in the USA with a US citizenship who have mostly dropped their educations and have none capital gain. Additionally, in cluster 2 we can see white middle-aged women born in the USA with a US citizenship who also are high school graduates but have also zero capital gains. Furthermore, in cluster 6 we notice that there are black, middle-aged women born in the USA with a US citizenship who are also high school graduates, but earn no money. Apart from that one other cluster (specifically cluster 9) contains white, middle-aged women, born mainly in Latin America who have dropped education and have none capital gain.

At this point, we should mention that, as shown above, in all clusters people have a high tendency to zero capital gains, despite their other characteristics. This could be explained by the fact that either the dataset is not filled properly in this variable section or the public dataset does not offer full information or something else

From the 10 clusters we attach as an example the results of Cluster 1, with the top values of the centroid.

CLUSTER	CLUSTER 1
<b>Sex = Female</b>	1.00
<b>Country of birth =United States</b>	0.99
<b>Citizenship= Native-Born in US</b>	0.99
<b>Capital gains=None</b>	0.99
<b>Education=Dropout</b>	0.97

### *Household composition*

*(Includes - Marital status, detailed household and family stat, family members under 18)*

Regarding the second clustering within the Household-related attributes, some interesting facts were raised. First, we got 5 clusters as the best split of citizens. Examining them, in cluster 0 we can see the widowed or not-married householders, living alone or with nonrelatives (i.e. roommates) and having no children under 18. At this point, we should notice that all but one clusters have the value “Not in universe” for the attribute “Family-members under 18”, which is one way to identify families with children.

Cluster 1 puts together married householders with no child under 18, while cluster 2 the householders that have never been married, live alone or with nonrelatives and have no children under 18. Interestingly, cluster 2 contains also citizens with at least one child over 18 who is never married. Additionally, cluster 3 can be explicitly described as it contains only one type of citizens. In specific, citizens in this cluster are married, with their spouses as the householders and have no children under 18. Finally, cluster 4 represents citizens that have never been married, but have under-aged children. In some cases, there are both parents, while in others one of them may have passed away or just not present.

As an example, we attach the results for cluster 2.

CLUSTER	CLUSTER 2
Family members under 18= Not in universe	0.94
Household and family stat= Child 18+, never married and not in a subfamily	0.36
Household and family stat= Nonfamily householder	0.25
Household and family stat= Secondary individual	0.13

#### *Education and Employment Traits*

*(Includes - Class of worker, education, own business or self-employed)*

As another approach to the data we did clustering on the socio-economic related attributes. There we got 3 clusters in total, which provided us with some interesting results. For start, in Cluster 0 we got all the High school graduates of the data. Most of them work in the Private sector, while others have their own businesses (self-employed) or do not work at all. Cluster 1 puts together education dropouts who are not working anywhere, while cluster 2 shows employees that have either dropped education or got one Bachelor's or Master's degree and are mainly working in the Private sector.

As an example, we attach the results for cluster 1.

CLUSTER	CLUSTER 1
Class of worker=Not-Working	0.96
Education=Dropout	0.90
Education= Bachelor's Degree(BA AB BS)	0.05
Education=Associates	0.02

#### *Employment attributes*

*(Includes - Detailed occupation, wage per hour, major industry code, full/part time employment stat, capital gains, fax filter stat)*

Finally, for the fourth clustering we used all the attributes related to Employment, which got us 8 clusters. However, only five are worth mentioning in this report. Firstly, in cluster 0 almost all citizens are lining on their own, have a nonfamily householder, meaning they live on rent, and fill the tax status as single. Additionally, they are either children, belong to the army or work full-time.

Next, cluster 1 represents citizens that own their home, fill the tax status with someone else and are both under 65. Regarding their employment status, they work in the Armed forces or as full time employees, mainly in the manufacturing industry or retail/trade industry but have none capital gains (this supports our theory that having no capital gains may be related to data issues).

Additionally, Cluster 3, on the other hand, contains the citizens that do not complete their tax obligations. We suppose that this cluster regards children with no tax obligation, who do not work and have no value about their major industry code.

At the end, cluster 7 contains the people who fill the tax status as single, work in the Armed forces or as full time employees. They have none capital gain and a child over 18 who is not living with the family anymore. As an example, we attach the results for cluster 0.

CLUSTER	CLUSTER 0
Tax filer status= Single	0.98
Capital gains=None	0.93
Full or part time employment stat= Children or Armed Forces	0.48
Full or part time employment stat= Full-time schedules	0.35

### Hierarchical Approach

We also tried out hierarchical procedure for clustering. For hierarchical there are two kinds of clustering procedures the agglomerative and the divisive. The first, is a "bottom up" approach where each observation starts in its own cluster and pairs of clusters are merged as one moves up the hierarchy. The second, is a "top down" approach where all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy. In general, the merges and splits are determined in a greedy manner. We continue with a "top down" approach firstly with the whole dataset and then with 10000 rows of this just to see if we will get better results but the results were misunderstanding both times. For hierarchical clustering we used max depth = 6, max leaf size = 5, k = 5, max runs = 10 and all the others left them by default.

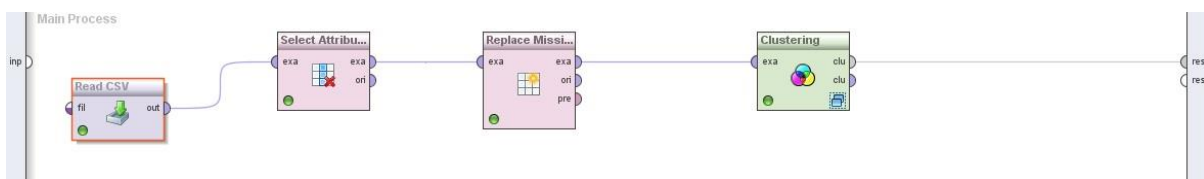


Figure 16 Hierarchical clustering

From the 10000 rows of data and the parameters that we mentioned before, hierarchical clustering returned us the following divisions of clusters. As we can see, there are very 5 distinct groups of clusters, which are different from one another. An interesting observation is that one group has only one node, which seems to be an outlier. The other four groups are split in smaller clusters in order to create smaller but more homogeneous groups of nodes. Even if this perspective is very insightful about the whole picture and to show big patterns, it cannot help us understand the differences between the nodes and thus between the smaller clusters.





Figure 17 Presenting the derived clusters via graph

In order to understand better how the nodes are split into groups we use a different perspective, the tree of clusters. From the image below, we can see very fast part of the bigger tree. It is important to say that the tree is very big and thus not presentable. However, we want to show the idea of how this tree may be useful for analysis and so we made a more user-friendly part of the whole tree. Below we show 4 levels of clusters from two branches. In specific, we can see two clusters split into smaller ones. If we need to derive knowledge from it, we can start either from the top or from the bottom of the tree.

As the divisive method that we used is top-down, we will read the tree top-down as well.

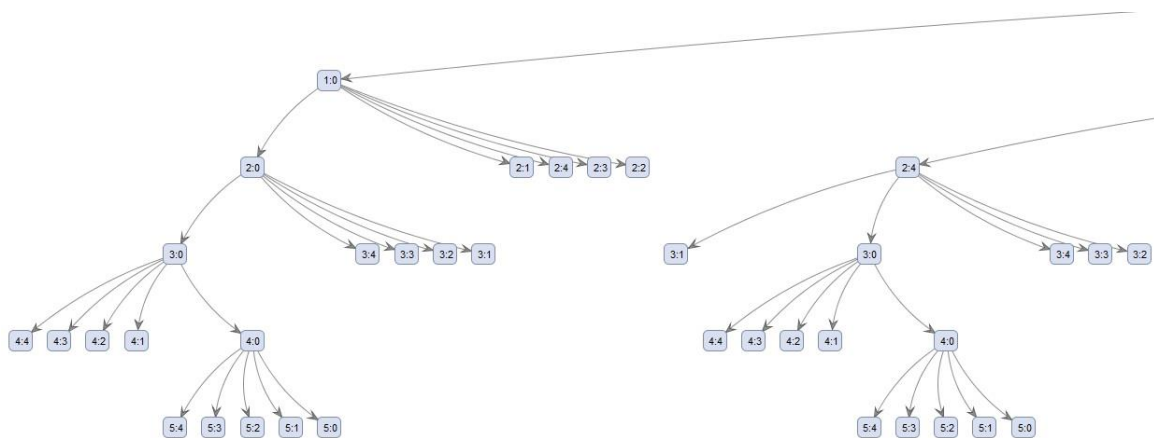


Figure 18 Presenting part of the clusters via tree



As we can see here we have many paths. From the left we can see the basics node 1, 2, 3, 4 which have their connections. The right part of the picture shows other connections between clusters 2, 3 and 4.

## SVD

Singular Value Decomposition (SVD) can be used to better understand an ExampleSet by showing the number of important dimensions. It can also be used to simplify the ExampleSet by reducing the number of attributes of the ExampleSet. This reduction removes unnecessary attributes that are linearly dependent in the point of view of Linear Algebra. It is useful when you have obtained data on a number of attributes (possibly a large number of attributes), and believe that there is some redundancy in those attributes. In this case, redundancy means that some of the attributes are correlated with one another, possibly because they are measuring the same construct. Because of this redundancy, you believe that it should be possible to reduce the observed attributes into a smaller number of components (artificial attributes) that will account for most of the variance in the observed attributes. For example, imagine an ExampleSet which contains an attribute that stores the water's temperature on several samples and another that stores its state (solid, liquid or gas). It is easy to see that the second attribute is dependent on the first attribute and, therefore, SVD could easily show us that it is not important for the analysis.

RapidMiner provides various dimensionality reduction operators e.g. the Principal Component Analysis operator. The Principal Component Analysis technique is a specific case of SVD. It is a mathematical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated attributes into a set of values of uncorrelated attributes called principal components. The number of principal components is less than or equal to the number of original attributes. This transformation is defined in such a way that the first principal component's variance is as high as possible (accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it should be orthogonal to (uncorrelated with) the preceding components.

Firstly, we tried to use only the “read CSV” and “SVD” operators in the whole dataset (numeric) and RapidMiner with the help of a computer(CPU=4cores and RAM=16GB) couldn't get us a result.

We tried out two different procedures with SVD (fixed number and dimensions=2) and SVD (fixed number and dimensions=10) just like the photo below shows.

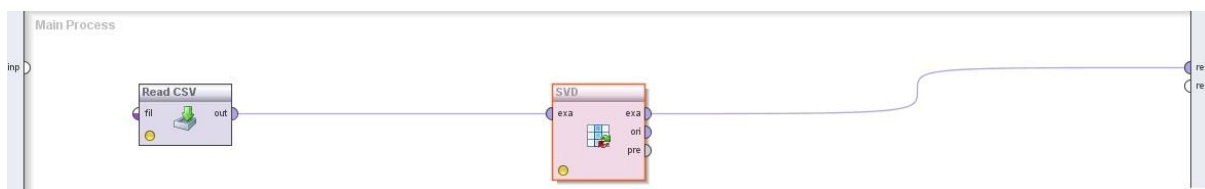


Figure 19 RapidMiner process with SVD

Secondly, we read again the dataset (numeric) with the help of “Read CSV”, then we use split data operator which produces the desired number of subsets of the given dataset. The dataset is partitioned into subsets according to the specified relative sizes. After that we use SVD operator (fixed number and dimensions=2). We faced a problem (in specific we got the error message “Input example set must have special attribute ‘label’”) when we run the procedure with the “Naïve Bayes” operator and we couldn't

get any results. We achieved to fix this problem, but this could not lead to any results, as the process was working for a very long time with top CPU usage which always ended with RapidMiner not responding.

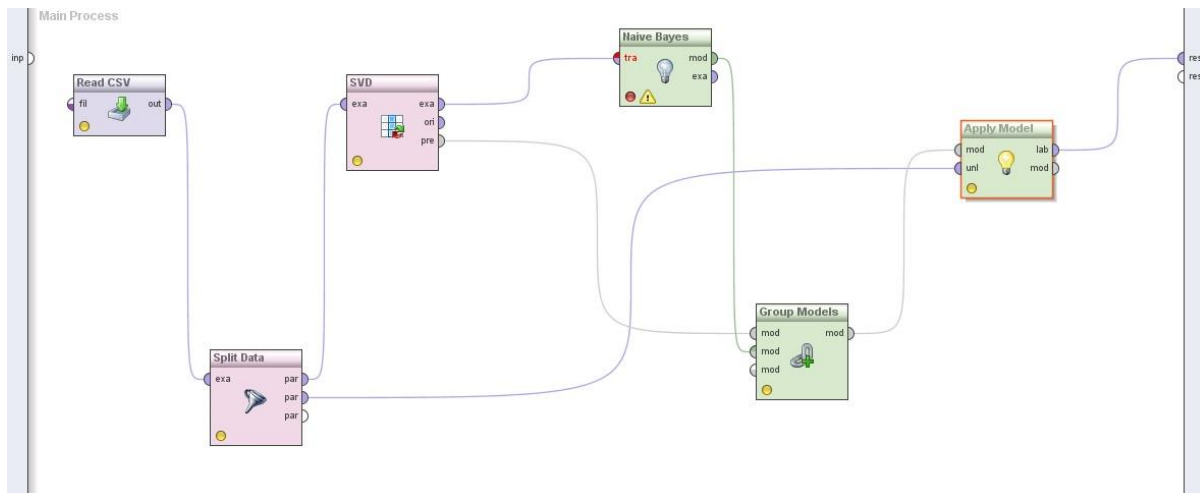


Figure 20 More complicate RapidMiner process with SVD

## Additional technics

### Logistics Regression

Logistic Regression is a type of classification model. In classification models, we attempt to predict the outcome of categorical dependent variables, using one or more independent variables. The independent variables can be either categorical or numerical. Logistic regression is based on the logistic function, which always takes values between 0 and 1. Replacing the dependent variable of the logistic function with a linear combination of dependent variables that we intend to use for regression, we conclude with the formula for logistic regression.

For this method we are going to transform the two variables again, age and income in a different way this time.

```
data$income = as.factor(ifelse(data$income==data$income[1],0,1))
data$age = scale(data$age)
```

When comparing classifiers, it's important to have a constant sample left out on which the classifier will not be trained, in order to see how well the trained classifier will generalize to new data. Here, we are randomly sampling approximately thirty percent of the data into a validation set that we will then use to check the accuracy of my model.

```
sample = rbinom(dim(data)[1],1,.3)
trainset = data[sample==0,]
valset = data[sample==1,]
```

To fit the logistic regression model, we will use the *glm* method.

```
fit1 = glm(predicted_income ~ ., family = binomial(logit), data = trainset)
```

Here for the formula, we are fitting income versus all other variables in the data frame. *glm* expects a family, which asks for the family of the error function, and the type of link function. Here we are using Binomial errors, and the link function is logit, for logistic regression.

```
fit1sw = step(fit1) # Keeps all variables
Start: AIC=57303.99
predicted_income ~ age + class.of.worker + education + marital.status +
  major.occupation.code + race + sex + capital.gains + capital.losses +
  country.of.birth.self
```

	Df	Deviance	AIC
<none>		57218	57304
race	4	57242	57320
country of birth self	11	57286	57350
class of worker	4	57523	57601
marital status	3	57781	57861
age	1	58084	58168
capital losses	2	58340	58422
major occupation code	8	59282	59352
sex	1	59682	59766
capital gains	2	60194	60276
education	6	61238	61312

Figure 21 Step function results

The `step()` function iteratively tries to remove predictor variables from the model in an attempt to delete variables that do not significantly add to the fit.

It is interesting to note here that by AIC, the stepwise model selection schema kept all variables in the model. We will want to choose another criteria to check the variables included. A good choice is the variable inflation factor. The variable inflation factor in the *car* package provides a useful means of checking multicollinearity. Multicollinearity is what happens when a given predictor in the model can be approximated by a linear combination of the other predictors in the model. This means that your inputs to the model are not independent of each other. This has the effect of increasing the variance of model coefficients. We want to decrease the variance to make our models more significant, more robust. That's why it's a good idea to omit variables from the model which exhibit a high degree of multicollinearity.

In this case, since we're using logistic regression, the *car* package implements a slightly different VIF calculation called GVIF, or generalized VIF. The interpretation is similar. On normal VIF in multiple regression, we attempt to eliminate variables with a VIF higher than 5.

```
vif(fit1)
```

We see from the output of the VIF function, that the variables `class.of.worker`, and `major.occupation.code` are heavily collinear. We will eliminate one of them from the model. We believe that `major.occupation.code` is less informative about our dataset. We will now fit the new model using the *glm()* function.

	GVIF	Df	GVIF <sup>1/(2*Df)</sup>
age	1.554.172	1	1.246.664
class of worker	506.436.132	4	2.178.039
education	1.651.121	6	1.042.673
marital status	1.335.684	3	1.049.423
major occupation code	710.522.334	8	1.507.383
race	2.105.301	4	1.097.525
sex	1.135.188	1	1.065.452
capital gains	1.018.103	2	1.004.495
capital losses	1.016.462	2	1.004.090
country of birth self	2.143.556	11	1.035.265

Figure 22 Vif function results

```
fit2 = glm(predicted_income ~.-major.occupation.code ,family = binomial(logit),
data = trainset)
```

```
vif(fit2)
```

	GVIF	Df	GVIF <sup>1/(2*Df)</sup>
age	1.282.282	1	1.132.379
class of worker	1.457.199	4	1.048.190
education	1.107.265	6	1.008.527
marital status	1.349.202	3	1.051.186
race	2.060.100	4	1.094.551
sex	1.044.269	1	1.021.895
capital gains	1.018.076	2	1.004.489
capital losses	1.015.184	2	1.003.775
country of birth self	2.080.083	11	1.033.852

VIF for fit2 finds no variables with a high VIF. Therefore the model passes the multicollinearity test.

Next, we predict based on the validation dataset, and use those predictions to build a [ROC curve](#) to assess the performance of our model.

```
fitpreds = predict(fit2,newdata=valset,type="response")
fitpred = prediction(fitpreds,valset$predicted_income)
fitperf = performance(fitpred,"tpr","fpr")
plot(fitperf,col="blue",lwd=2,main="ROC Curve for Logistic: Adult")
abline(a=0,b=1,lwd=2,lty=2,col="gray")
```

Finally, the figure below shows a ROC curve through which we can see that the performance of the model rises well above the diagonal line, indicating we are doing much better than random guess.

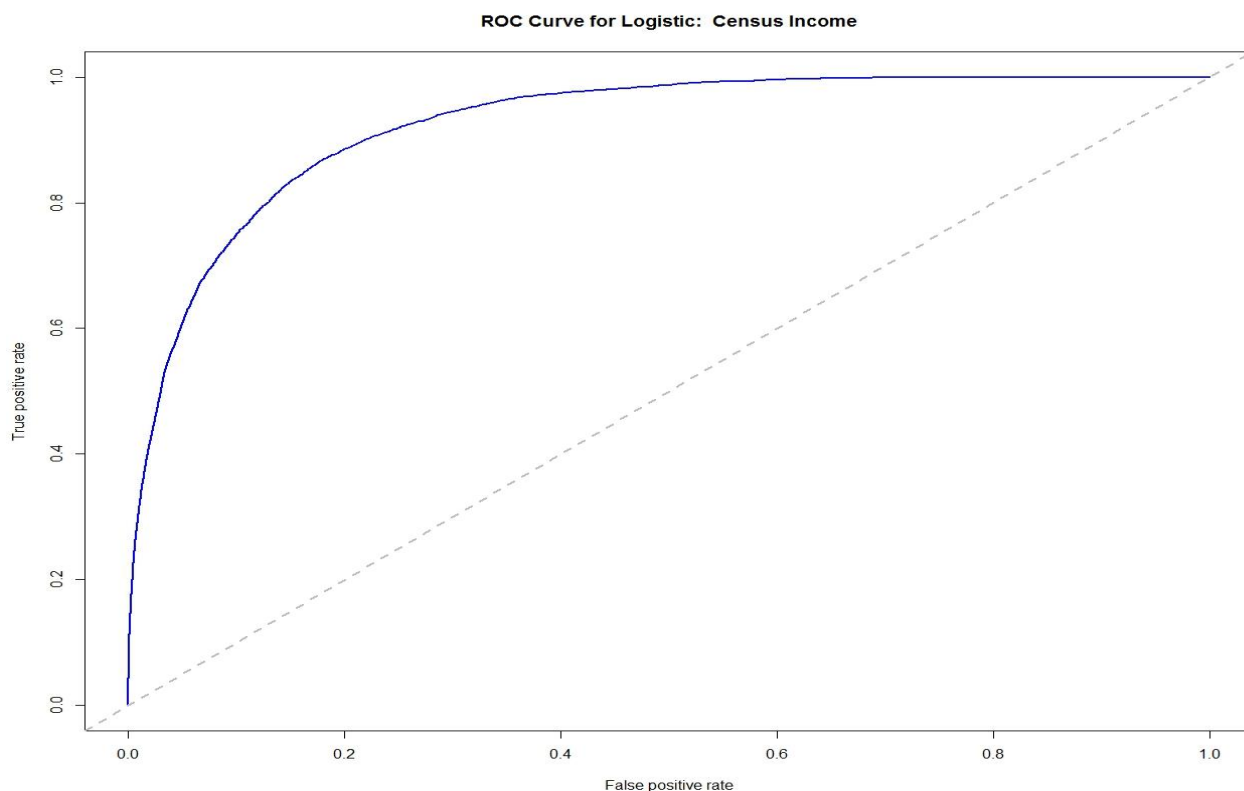


Figure 24 ROC curve

## Results Discussion

### Prediction of income level

The three models used in the analysis section aimed to categorize citizens into two categories (classification) depending on their characteristics and find the most important associations between the input attributes and the predicted income (association rules). In the next table we compare the five most important characteristics that are associated with the predicted attribute, according to each model.

Attribute number	Decision tree	Naïve Bayes	Association rules	Logistic Regression
1	Detailed occupation recode	Detailed occupation recode	Class of worker	Age
2	Education	Education	Race	Class of worker
3	Sex	Major occupation recode	Sex	Education
4	Age	Weeks worked in Year	Capital gains	Marital status
5	Number of Persons worked for Employer	Detailed industry recode	Country of birth self	Race

The above table shows that the labor market position of the individual is significantly affecting the level of income in both classification models, while also important in logistic regression. Similarly, the level of education is also an important factor since it seems that people with master's degree have larger annual salaries. The other characteristics are different, but in general we can combine the results and conclude that in the USA of 1994-1995 white people, especially men, that had specific working industries (e.g. private sector), of certain age ranges (especially older people) and specific education levels (Bachelor's degree and higher) are more possible to have higher income than the other citizens.

### Citizens' segmentation

Clustering results are attached in the appendix, while the most interesting and useful ones are described in the previous section. In general, clustering is a useful method for patterns discovery per target of analysis. For example, advertisers may want to find out what patterns of consumers they will reach through a generic advertisement in television (for targeted campaigns they should also have an identification attribute so as to relate each consumer with a cluster).

On the other hand, social scientists may want to find segments of citizens with different gender, racial or educational characteristics through the years in order to examine changes in social structures and/or the evolution of the society. In this case, they would use data from different years and for a big time period. However, here we got data only from two consecutive years and thus no such analysis was possible.

Additionally, a foreign company could use some attributes of the data so as to understand what types of employees it could reach. Knowing the patterns of employment-related attributes could help it understand the culture and the standards of a new area and act accordingly. This could help decide the offices new headquarters or identify the most potential area for growth.

Concluding, we attempted four different clustering for four different sub-datasets. However, having a more specific question to answer would lead us to more specific results and conclusions.

## References

<https://thevellum.files.wordpress.com/2012/03/datamining-paper-03152012-v01-morecomplete-2chapters.pdf>

[http://vlabs.ihu.edu.gr/fileadmin/labsfiles/decision\\_support\\_systems/lessons/CITrees\\_AssocRules/CITrees\\_AssocRules-IHU.pdf](http://vlabs.ihu.edu.gr/fileadmin/labsfiles/decision_support_systems/lessons/CITrees_AssocRules/CITrees_AssocRules-IHU.pdf)

[http://www.novaims.unl.pt/ensino/docentes/fbacao/bacao\\_kdnet04.pdf](http://www.novaims.unl.pt/ensino/docentes/fbacao/bacao_kdnet04.pdf)

[http://docs.rapidminer.com/studio/operators/data\\_transformation/attribute\\_space\\_transformation/transformation/singular\\_value\\_decomposition.html](http://docs.rapidminer.com/studio/operators/data_transformation/attribute_space_transformation/transformation/singular_value_decomposition.html)

## Appendix

### Full list of attributes

age  
class of worker  
industry code  
occupation code  
education  
wage per hour  
enrolled in edu inst last wk  
marital status  
major industry code  
major occupation code  
hispanic Origin  
sex  
member of a labor union  
reason for unemployment  
full or part time employment stat  
capital gains  
capital losses  
divdends from stocks  
tax filer status  
region of previous residence  
state of previous residence  
detailed household and family stat  
detailed household summary in household  
instance weight  
migration code-change in msa (removed)  
migration code-change in reg (removed)  
migration code-move within reg (removed)  
live in this house 1 year ago  
migration prev res in sunbelt (removed)  
num persons worked for employer  
family members under 18  
country of birth father  
country of birth mother  
country of birth self  
citizenship  
own business or self employed  
fill inc questionnaire for veteran's admin  
veterans benefits  
weeks worked in year  
year  
predicted income



## Clustering results

### Demographic attributes

Cluster	cluster_0	cluster_1	cluster_2	cluster_3	cluster_4	cluster_5	cluster_6	cluster_7	cluster_8	cluster_9
age=Middle-aged	1.000	0.081	1.000	0.436	0.000	0.000	0.497	0.000	0.000	0.416
age=Old	0.000	0.093	0.000	0.088	0.347	0.112	0.102	0.000	0.356	0.124
age=Senior	0.000	0.084	0.000	0.204	0.653	0.068	0.254	0.000	0.644	0.223
age=Young	0.000	0.742	0.000	0.272	0.000	0.821	0.148	1.000	0.000	0.238
capital.gains=High	0.047	0.001	0.014	0.025	0.058	0.001	0.016	0.001	0.015	0.007
capital.gains=Low	0.037	0.004	0.013	0.027	0.047	0.007	0.024	0.006	0.023	0.011
capital.gains=None	0.917	0.995	0.973	0.947	0.895	0.992	0.960	0.993	0.962	0.983
citizenship= Foreign born- Not a citizen of U S	0.000	0.000	0.000	0.604	0.000	0.000	0.000	0.000	0.000	0.583
citizenship= Foreign born- U S citizen by naturalization	0.000	0.000	0.000	0.251	0.000	0.000	0.000	0.000	0.000	0.272
citizenship= Native- Born abroad of American Parent(s)	0.000	0.003	0.000	0.084	0.000	0.000	0.000	0.000	0.000	0.074
citizenship= Native- Born in Puerto Rico or U S Outlying	0.000	0.002	0.000	0.061	0.000	0.000	0.000	0.000	0.000	0.071
citizenship= Native- Born in the United States	1.000	0.996	1.000	0.000	1.000	1.000	1.000	1.000	1.000	0.000
country.of.birth.self= Panama	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.002
country.of.birth.self= South Korea	0.000	0.000	0.000	0.018	0.000	0.000	0.000	0.000	0.000	0.023
country.of.birth.self=British-Commonwealth	0.000	0.000	0.000	0.077	0.000	0.000	0.000	0.000	0.000	0.080
country.of.birth.self=China	0.000	0.000	0.000	0.035	0.000	0.000	0.000	0.000	0.000	0.033
country.of.birth.self=Euro_1	0.000	0.002	0.000	0.054	0.000	0.000	0.000	0.000	0.000	0.070
country.of.birth.self=Euro_2	0.000	0.000	0.000	0.040	0.000	0.000	0.000	0.000	0.000	0.035
country.of.birth.self=Latin-America	0.000	0.002	0.000	0.422	0.000	0.000	0.000	0.000	0.000	0.403
country.of.birth.self=Other	0.000	0.000	0.000	0.058	0.000	0.000	0.000	0.000	0.000	0.060
country.of.birth.self=SE-Asia	0.000	0.000	0.000	0.066	0.000	0.000	0.000	0.000	0.000	0.073
country.of.birth.self=South-America	0.000	0.000	0.000	0.070	0.000	0.000	0.000	0.000	0.000	0.075
country.of.birth.self=United-States	1.000	0.996	1.000	0.000	1.000	1.000	1.000	1.000	1.000	0.000
country.of.birth.self=Unknown	0.000	0.000	0.000	0.158	0.000	0.000	0.000	0.000	0.000	0.147
education= Bachelor's degree	0.199	0.012	0.192	0.109	0.135	0.027	0.092	0.022	0.124	0.108
education= Doctorate degree	0.010	0.000	0.005	0.019	0.020	0.000	0.004	0.000	0.007	0.005
education= Master degree	0.056	0.001	0.058	0.044	0.066	0.002	0.031	0.001	0.059	0.027

education= Prof school degree	0.022	0.000	0.010	0.018	0.025	0.000	0.005	0.000	0.005	0.010
education=Associates	0.087	0.008	0.106	0.037	0.048	0.014	0.064	0.011	0.075	0.048
education=Dropout	0.087	0.979	0.073	0.452	0.225	0.794	0.118	0.811	0.000	0.449
education=HS-Graduate	0.539	0.000	0.556	0.322	0.480	0.164	0.686	0.155	0.729	0.353
race=Amer-Indian	0.015	0.087	0.012	0.004	0.008	0.003	0.000	0.017	0.007	0.002
race=Asian	0.007	0.086	0.007	0.166	0.005	0.003	0.000	0.022	0.006	0.171
race=Black	0.000	0.728	0.000	0.061	0.000	0.000	1.000	0.111	0.000	0.058
race=Other	0.008	0.099	0.008	0.077	0.003	0.003	0.000	0.022	0.003	0.070
race=White	0.970	0.000	0.974	0.693	0.984	0.992	0.000	0.828	0.985	0.698
sex= Female	0.000	1.000	1.000	0.000	0.000	1.000	0.514	0.000	1.000	1.000
sex= Male	1.000	0.000	0.000	1.000	1.000	0.000	0.486	1.000	0.000	0.000

### Household composition

Cluster	cluster_0	cluster_1	cluster_2	cluster_3	cluster_4
marital.status=Never-Married	0.000	0.000	1.000	0.000	1.000
detailed.household.and.family.stat= Child <18 never marr not in subfamily	0.000	0.000	0.000	0.000	0.940
family.members.under.18= Both parents present	0.000	0.000	0.000	0.000	0.715
family.members.under.18= Mother only present	0.001	0.000	0.000	0.000	0.252
detailed.household.and.family.stat= Grandchild <18 never marr child of subfamily RP	0.000	0.000	0.004	0.000	0.034
family.members.under.18= Father only present	0.000	0.000	0.008	0.000	0.033
detailed.household.and.family.stat= Child under 18 of RP of unrel subfamily	0.000	0.000	0.002	0.000	0.013
detailed.household.and.family.stat= Other Rel <18 never marr child of subfamily RP	0.000	0.000	0.001	0.000	0.012
detailed.household.and.family.stat= In group quarters	0.001	0.000	0.003	0.000	0.000
marital.status=Widowed	0.373	0.000	0.000	0.000	0.000
detailed.household.and.family.stat= Child <18 ever marr not in subfamily	0.001	0.000	0.000	0.000	0.000
family.members.under.18= Not in universe	0.999	1.000	0.944	1.000	0.000
marital.status=Married	0.000	1.000	0.000	1.000	0.000
detailed.household.and.family.stat= Spouse of householder	0.000	0.000	0.000	1.000	0.000
detailed.household.and.family.stat= Child 18+ never marr Not in a subfamily	0.000	0.000	0.365	0.000	0.000

detailed.household.and.family.stat= Nonfamily householder	0.495	0.000	0.251	0.000	0.000
detailed.household.and.family.stat= Secondary individual	0.065	0.000	0.130	0.000	0.000
detailed.household.and.family.stat= Householder	0.299	0.960	0.099	0.000	0.000
detailed.household.and.family.stat= Other Rel 18+ never marr not in subfamily	0.000	0.000	0.051	0.000	0.000
family.members.under.18= Neither parent present	0.000	0.000	0.049	0.000	0.000
detailed.household.and.family.stat= Grandchild <18 never marr not in subfamily	0.000	0.000	0.031	0.000	0.000
detailed.household.and.family.stat= Child 18+ never marr RP of subfamily	0.000	0.000	0.018	0.000	0.000
detailed.household.and.family.stat= Other Rel <18 never marr not in subfamily	0.000	0.000	0.018	0.000	0.000
detailed.household.and.family.stat= Grandchild 18+ never marr not in subfamily	0.000	0.000	0.012	0.000	0.000
detailed.household.and.family.stat= RP of unrelated subfamily	0.013	0.001	0.008	0.000	0.000
detailed.household.and.family.stat= Other Rel 18+ never marr RP of subfamily	0.000	0.000	0.003	0.000	0.000
detailed.household.and.family.stat= Child <18 never marr RP of subfamily	0.000	0.000	0.002	0.000	0.000
detailed.household.and.family.stat= Grandchild 18+ never marr RP of subfamily	0.000	0.000	0.000	0.000	0.000
detailed.household.and.family.stat= Other Rel <18 never married RP of subfamily	0.000	0.000	0.000	0.000	0.000
detailed.household.and.family.stat= Grandchild <18 never marr RP of subfamily	0.000	0.000	0.000	0.000	0.000
detailed.household.and.family.stat= Other Rel 18+ spouse of subfamily RP	0.000	0.015	0.000	0.000	0.000
detailed.household.and.family.stat= Other Rel 18+ ever marr RP of subfamily	0.006	0.011	0.000	0.000	0.000
detailed.household.and.family.stat= Child 18+ ever marr RP of subfamily	0.013	0.007	0.000	0.000	0.000
detailed.household.and.family.stat= Child 18+ spouse of subfamily RP	0.000	0.003	0.000	0.000	0.000

detailed.household.and.family.stat= Spouse of RP of unrelated subfamily	0.000	0.001	0.000	0.000	0.000
detailed.household.and.family.stat= Grandchild 18+ spouse of subfamily RP	0.000	0.000	0.000	0.000	0.000
detailed.household.and.family.stat= Child <18 ever marr RP of subfamily	0.000	0.000	0.000	0.000	0.000
detailed.household.and.family.stat= Grandchild 18+ ever marr RP of subfamily	0.000	0.000	0.000	0.000	0.000
detailed.household.and.family.stat= Other Rel <18 ever marr RP of subfamily	0.000	0.000	0.000	0.000	0.000
detailed.household.and.family.stat= Other Rel <18 spouse of subfamily RP	0.000	0.000	0.000	0.000	0.000
detailed.household.and.family.stat= Child <18 spouse of subfamily RP	0.000	0.000	0.000	0.000	0.000
marital.status=Not-Married	0.627	0.000	0.000	0.000	0.000
detailed.household.and.family.stat= Other Rel 18+ ever marr not in subfamily	0.070	0.000	0.000	0.000	0.000
detailed.household.and.family.stat= Child 18+ ever marr Not in a subfamily	0.035	0.000	0.000	0.000	0.000
detailed.household.and.family.stat= Grandchild 18+ ever marr not in subfamily	0.001	0.000	0.000	0.000	0.000
detailed.household.and.family.stat= Other Rel <18 ever marr not in subfamily	0.000	0.000	0.000	0.000	0.000
detailed.household.and.family.stat= Grandchild <18 ever marr not in subfamily	0.000	0.000	0.000	0.000	0.000

#### *Education and Employment Traits*

Cluster	cluster_0	cluster_1	cluster_2
class.of.worker=Federal-Govt	0.020666	0.001629	0.02707
class.of.worker=Not-Working	0.312069	0.969011	0
class.of.worker=Other-Govt	0.066857	0.010505	0.139293
class.of.worker=Private	0.520771	0	0.736359
class.of.worker=Self-Employed	0.079637	0.018855	0.097278
education= Bachelor's degree	0	0.050432	0.364905
education= Doctorate degree	0	0.002751	0.023857
education= Master's degree	0	0.015624	0.122636
education= Prof school degree	0	0.003525	0.034452
education=Associates	0	0.026387	0.17467
education=Dropout	0	0.901282	0.27948
education=HS-Graduate	1	0	0

Employment attributes

Cluster	cluster_0	cluster_1	cluster_2	cluster_3	cluster_4	cluster_5	cluster_6	cluster_7
capital.gains=High	0.026	0.070	0.007	0.000	0.000	0.036	0.000	0.004
capital.gains=Low	0.044	0.044	0.017	0.000	0.001	0.050	0.001	0.021
capital.gains=None	0.930	0.886	0.976	1.000	0.999	0.915	0.999	0.975
detailed.household.and.family.stat=Child <18 ever marr not in subfamily	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000
detailed.household.and.family.stat=Child <18 ever marr RP of subfamily	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
detailed.household.and.family.stat=Child <18 never marr not in subfamily	0.044	0.000	0.110	0.746	0.000	0.000	0.000	0.000
detailed.household.and.family.stat=Child <18 never marr RP of subfamily	0.000	0.000	0.001	0.001	0.000	0.001	0.000	0.000
detailed.household.and.family.stat=Child <18 spouse of subfamily RP	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
detailed.household.and.family.stat=Child 18+ ever marr Not in a subfamily	0.026	0.000	0.006	0.002	0.000	0.000	0.001	0.000
detailed.household.and.family.stat=Child 18+ ever marr RP of subfamily	0.000	0.001	0.005	0.001	0.006	0.020	0.007	0.000
detailed.household.and.family.stat=Child 18+ never marr Not in a subfamily	0.000	0.000	0.050	0.021	0.000	0.000	0.000	1.000
detailed.household.and.family.stat=Child 18+ never marr RP of subfamily	0.000	0.000	0.005	0.002	0.000	0.025	0.001	0.000
detailed.household.and.family.stat=Child 18+ spouse of subfamily RP	0.000	0.000	0.001	0.000	0.004	0.000	0.002	0.000
detailed.household.and.family.stat=Child under 18 of RP of unrel subfamily	0.001	0.000	0.001	0.012	0.000	0.000	0.000	0.000
detailed.household.and.family.stat=Grandchild <18 ever marr not in subfamily	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
detailed.household.and.family.stat=Grandchild <18 never marr child of subfamily RP	0.000	0.000	0.002	0.031	0.000	0.000	0.000	0.000
detailed.household.and.family.stat=Grandchild <18 never marr not in subfamily	0.001	0.000	0.003	0.017	0.000	0.000	0.000	0.000

detailed.household.and.family.stat= Grandchild <18 never marr RP of subfamily	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
detailed.household.and.family.stat= Grandchild 18+ ever marr not in subfamily	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000
detailed.household.and.family.stat= Grandchild 18+ ever marr RP of subfamily	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
detailed.household.and.family.stat= Grandchild 18+ never marr not in subfamily	0.008	0.000	0.003	0.001	0.000	0.000	0.001	0.000
detailed.household.and.family.stat= Grandchild 18+ never marr RP of subfamily	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
detailed.household.and.family.stat= Grandchild 18+ spouse of subfamily RP	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
detailed.household.and.family.stat= Householder	0.025	0.997	0.299	0.045	0.000	0.914	0.000	0.000
detailed.household.and.family.stat= In group quarters	0.002	0.000	0.001	0.001	0.000	0.000	0.000	0.000
detailed.household.and.family.stat= Nonfamily householder	0.646	0.000	0.106	0.043	0.000	0.000	0.000	0.000
detailed.household.and.family.stat= Other Rel <18 ever marr not in subfamily	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
detailed.household.and.family.stat= Other Rel <18 ever marr RP of subfamily	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
detailed.household.and.family.stat= Other Rel <18 never marr child of subfamily RP	0.000	0.000	0.002	0.010	0.000	0.000	0.000	0.000
detailed.household.and.family.stat= Other Rel <18 never marr not in subfamily	0.001	0.000	0.002	0.009	0.000	0.000	0.001	0.000
detailed.household.and.family.stat= Other Rel <18 never married RP of subfamily	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
detailed.household.and.family.stat= Other Rel <18 spouse of subfamily RP	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
detailed.household.and.family.stat= Other Rel 18+ ever marr not in subfamily	0.025	0.000	0.033	0.010	0.000	0.000	0.001	0.000

<b>detailed.household.and.family.stat= Other Rel 18+ ever marr RP of subfamily</b>	0.000	0.001	0.006	0.001	0.009	0.008	0.007	0.000
<b>detailed.household.and.family.stat= Other Rel 18+ never marr not in subfamily</b>	0.041	0.000	0.011	0.005	0.000	0.001	0.003	0.000
<b>detailed.household.and.family.stat= Other Rel 18+ never marr RP of subfamily</b>	0.000	0.000	0.001	0.000	0.000	0.004	0.000	0.000
<b>detailed.household.and.family.stat= Other Rel 18+ spouse of subfamily RP</b>	0.000	0.001	0.005	0.001	0.012	0.001	0.013	0.000
<b>detailed.household.and.family.stat= RP of unrelated subfamily</b>	0.004	0.000	0.003	0.003	0.001	0.025	0.002	0.000
<b>detailed.household.and.family.stat= Secondary individual</b>	0.174	0.000	0.026	0.016	0.000	0.000	0.001	0.000
<b>detailed.household.and.family.stat= Spouse of householder</b>	0.000	0.000	0.315	0.021	0.966	0.000	0.958	0.000
<b>detailed.household.and.family.stat= Spouse of RP of unrelated subfamily</b>	0.000	0.000	0.000	0.000	0.001	0.000	0.001	0.000
<b>full.or.part.time.employment.stat= Children or Armed Forces</b>	0.489	0.532	0.000	0.989	1.000	0.644	0.000	0.510
<b>full.or.part.time.employment.stat= Full-time schedules</b>	0.353	0.399	0.000	0.007	0.000	0.271	0.837	0.338
<b>full.or.part.time.employment.stat= Not in labor force</b>	0.090	0.003	0.996	0.000	0.000	0.004	0.011	0.067
<b>full.or.part.time.employment.stat= PT for econ reasons usually FT</b>	0.005	0.005	0.000	0.000	0.000	0.004	0.009	0.005
<b>full.or.part.time.employment.stat= PT for econ reasons usually PT</b>	0.010	0.012	0.000	0.000	0.000	0.009	0.027	0.006
<b>full.or.part.time.employment.stat= PT for non-econ reasons usually FT</b>	0.028	0.029	0.000	0.000	0.000	0.031	0.073	0.026
<b>full.or.part.time.employment.stat= Unemployed full-time</b>	0.020	0.017	0.003	0.001	0.000	0.029	0.030	0.034
<b>full.or.part.time.employment.stat= Unemployed part- time</b>	0.006	0.003	0.001	0.003	0.000	0.007	0.014	0.013
<b>major.industry.code= Agriculture</b>	0.021	0.033	0.001	0.002	0.016	0.029	0.024	0.026
<b>major.industry.code= Armed Forces</b>	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
<b>major.industry.code= Business and repair services</b>	0.052	0.056	0.001	0.002	0.032	0.046	0.050	0.058
<b>major.industry.code= Communications</b>	0.011	0.013	0.000	0.000	0.007	0.008	0.012	0.007
<b>major.industry.code= Construction</b>	0.050	0.090	0.001	0.001	0.020	0.031	0.035	0.051
<b>major.industry.code= Education</b>	0.056	0.057	0.000	0.001	0.087	0.057	0.126	0.072
<b>major.industry.code= Entertainment</b>	0.018	0.011	0.001	0.001	0.009	0.010	0.012	0.028

major.industry.code= Finance insurance and real estate	0.050	0.048	0.000	0.000	0.052	0.049	0.077	0.046
major.industry.code= Forestry and fisheries	0.002	0.003	0.000	0.000	0.001	0.001	0.001	0.002
major.industry.code= Hospital services	0.030	0.023	0.000	0.000	0.040	0.039	0.058	0.021
major.industry.code= Manufacturing-durable goods	0.067	0.128	0.000	0.001	0.044	0.055	0.067	0.056
major.industry.code= Manufacturing-nondurable goods	0.053	0.077	0.000	0.001	0.043	0.054	0.064	0.047
major.industry.code= Medical except hospital	0.034	0.024	0.000	0.001	0.048	0.054	0.074	0.026
major.industry.code= Mining	0.003	0.012	0.000	0.000	0.002	0.003	0.003	0.003
major.industry.code= Not in universe or children	0.194	0.065	0.992	0.971	0.330	0.237	0.000	0.110
major.industry.code= Other professional services	0.037	0.047	0.000	0.001	0.031	0.031	0.050	0.026
major.industry.code= Personal services except private HH	0.027	0.017	0.000	0.001	0.023	0.033	0.032	0.026
major.industry.code= Private household services	0.011	0.002	0.000	0.002	0.006	0.012	0.009	0.007
major.industry.code= Public administration	0.036	0.053	0.000	0.000	0.028	0.037	0.044	0.023
major.industry.code= Retail trade	0.162	0.107	0.002	0.011	0.108	0.131	0.155	0.282
major.industry.code= Social services	0.019	0.010	0.000	0.001	0.026	0.026	0.036	0.023
major.industry.code= Transportation	0.033	0.057	0.000	0.000	0.021	0.027	0.035	0.030
major.industry.code= Utilities and sanitary services	0.007	0.020	0.000	0.000	0.006	0.006	0.008	0.006
major.industry.code= Wholesale trade	0.027	0.048	0.000	0.000	0.018	0.025	0.029	0.024
tax.filer.status= Head of household	0.000	0.000	0.017	0.000	0.000	0.550	0.000	0.000
tax.filer.status= Joint both 65+	0.000	0.000	0.150	0.000	0.104	0.192	0.018	0.000
tax.filer.status= Joint both under 65	0.000	1.000	0.240	0.000	0.845	0.000	0.934	0.000
tax.filer.status= Joint one under 65 & one 65+	0.000	0.000	0.048	0.000	0.047	0.106	0.031	0.000
tax.filer.status= Nonfiler	0.012	0.000	0.504	0.996	0.005	0.061	0.017	0.106
tax.filer.status= Single	0.988	0.000	0.041	0.003	0.000	0.090	0.000	0.894



## Complete XML code of the clustering analysis

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<process version="6.5.002">
  <operator activated="true" class="read_csv" compatibility="6.5.002" expanded="true"
height="60" name="Read CSV" width="90" x="112" y="165">
    <parameter key="csv_file" value="C:\Users\Kostas\Desktop\binary files\New
Binary\teliko_2.txt"/>
    <parameter key="column_separators" value="&#9;"/>
    <parameter key="trim_lines" value="false"/>
    <parameter key="use_quotes" value="true"/>
    <parameter key="quotes_character" value="&quot;"/>
    <parameter key="escape_character" value="\"/>
    <parameter key="skip_comments" value="false"/>
    <parameter key="comment_characters" value="#">
    <parameter key="parse_numbers" value="true"/>
    <parameter key="decimal_character" value=".">
    <parameter key="grouped_digits" value="false"/>
    <parameter key="grouping_character" value=",">
    <parameter key="date_format" value=""/>
    <parameter key="first_row_as_names" value="false"/>
    <list key="annotations">
      <parameter key="0" value="Name"/>
    </list>
    <parameter key="time_zone" value="SYSTEM"/>
    <parameter key="locale" value="English (United States)"/>
    <parameter key="encoding" value="windows-1253"/>
    <list key="data_set_meta_data_information">
      <parameter key="0" value="class\.of\.worker=Federal-
Govt.true.integer.attribute"/>
      <parameter key="1" value="class\.of\.worker=Not-
Working.true.integer.attribute"/>
      <parameter key="2" value="class\.of\.worker=Other-Govt.true.integer.attribute"/>
      <parameter key="3" value="class\.of\.worker=Private.true.integer.attribute"/>
      <parameter key="4" value="class\.of\.worker=Self-
Employed.true.integer.attribute"/>
      <parameter key="5" value="education= Bachelors degree (BA AB
BS).true.integer.attribute"/>
      <parameter key="6" value="education= Doctorate degree (PhD
EdD).true.integer.attribute"/>
      <parameter key="7" value="education= Masters degree (MA MS MEng MEd MSW
MBA).true.integer.attribute"/>
      <parameter key="8" value="education= Prof school degree (MD DDS DVM LLB
JD).true.integer.attribute"/>
      <parameter key="9" value="education=Associates.true.integer.attribute"/>
      <parameter key="10" value="education=Dropout.true.integer.attribute"/>
      <parameter key="11" value="education=HS-Graduate.true.integer.attribute"/>
      <parameter key="12" value="marital\.status=Married.true.integer.attribute"/>
      <parameter key="13" value="marital\.status=Never-
Married.true.integer.attribute"/>
      <parameter key="14" value="marital\.status=Not-Married.true.integer.attribute"/>
      <parameter key="15" value="marital\.status=Widowed.true.integer.attribute"/>
      <parameter key="16" value="major\.industry\.code=
Agriculture.true.integer.attribute"/>
      <parameter key="17" value="major\.industry\.code= Armed
Forces.true.integer.attribute"/>
      <parameter key="18" value="major\.industry\.code= Business and repair
services.true.integer.attribute"/>
      <parameter key="19" value="major\.industry\.code=
Communications.true.integer.attribute"/>
      <parameter key="20" value="major\.industry\.code=
Construction.true.integer.attribute"/>
    </list>
  </operator>
</process>
```

```

    <parameter key="21" value="major\industry\code=
Education.true.integer.attribute"/>
    <parameter key="22" value="major\industry\code=
Entertainment.true.integer.attribute"/>
    <parameter key="23" value="major\industry\code= Finance insurance and real
estate.true.integer.attribute"/>
    <parameter key="24" value="major\industry\code= Forestry and
fisheries.true.integer.attribute"/>
    <parameter key="25" value="major\industry\code= Hospital
services.true.integer.attribute"/>
    <parameter key="26" value="major\industry\code= Manufacturing-durable
goods.true.integer.attribute"/>
    <parameter key="27" value="major\industry\code= Manufacturing-nondurable
goods.true.integer.attribute"/>
    <parameter key="28" value="major\industry\code= Medical except
hospital.true.integer.attribute"/>
    <parameter key="29" value="major\industry\code=
Mining.true.integer.attribute"/>
    <parameter key="30" value="major\industry\code= Not in universe or
children.true.integer.attribute"/>
    <parameter key="31" value="major\industry\code= Other professional
services.true.integer.attribute"/>
    <parameter key="32" value="major\industry\code= Personal services except
private HH.true.integer.attribute"/>
    <parameter key="33" value="major\industry\code= Private household
services.true.integer.attribute"/>
    <parameter key="34" value="major\industry\code= Public
administration.true.integer.attribute"/>
    <parameter key="35" value="major\industry\code= Retail
trade.true.integer.attribute"/>
    <parameter key="36" value="major\industry\code= Social
services.true.integer.attribute"/>
    <parameter key="37" value="major\industry\code=
Transportation.true.integer.attribute"/>
    <parameter key="38" value="major\industry\code= Utilities and sanitary
services.true.integer.attribute"/>
    <parameter key="39" value="major\industry\code= Wholesale
trade.true.integer.attribute"/>
    <parameter key="40" value="race=Amer-Indian.true.integer.attribute"/>
    <parameter key="41" value="race=Asian.true.integer.attribute"/>
    <parameter key="42" value="race=Black.true.integer.attribute"/>
    <parameter key="43" value="race=Other.true.integer.attribute"/>
    <parameter key="44" value="race=White.true.integer.attribute"/>
    <parameter key="45" value="sex= Female.true.integer.attribute"/>
    <parameter key="46" value="sex= Male.true.integer.attribute"/>
    <parameter key="47" value="full\or\part\time\employment\stat= Children or
Armed Forces.true.integer.attribute"/>
    <parameter key="48" value="full\or\part\time\employment\stat= Full-time
schedules.true.integer.attribute"/>
    <parameter key="49" value="full\or\part\time\employment\stat= Not in labor
force.true.integer.attribute"/>
    <parameter key="50" value="full\or\part\time\employment\stat= PT for econ
reasons usually FT.true.integer.attribute"/>
    <parameter key="51" value="full\or\part\time\employment\stat= PT for econ
reasons usually PT.true.integer.attribute"/>
    <parameter key="52" value="full\or\part\time\employment\stat= PT for non-
econ reasons usually FT.true.integer.attribute"/>
    <parameter key="53" value="full\or\part\time\employment\stat= Unemployed
full-time.true.integer.attribute"/>
    <parameter key="54" value="full\or\part\time\employment\stat= Unemployed
part- time.true.integer.attribute"/>
    <parameter key="55" value="tax\filer\status= Head of
household.true.integer.attribute"/>

```

```

        <parameter key="56" value="tax\filer.status= Joint both
65+.true.integer.attribute"/>
        <parameter key="57" value="tax\filer.status= Joint both under
65.true.integer.attribute"/>
        <parameter key="58" value="tax\filer.status= Joint one under 65 & one
65+.true.integer.attribute"/>
        <parameter key="59" value="tax\filer.status=
Nonfiler.true.integer.attribute"/>
        <parameter key="60" value="tax\filer.status= Single.true.integer.attribute"/>
        <parameter key="61" value="state\of\previous.residence=
Abroad.true.integer.attribute"/>
        <parameter key="62" value="state\of\previous.residence=
Alabama.true.integer.attribute"/>
        <parameter key="63" value="state\of\previous.residence=
Alaska.true.integer.attribute"/>
        <parameter key="64" value="state\of\previous.residence=
Arizona.true.integer.attribute"/>
        <parameter key="65" value="state\of\previous.residence=
Arkansas.true.integer.attribute"/>
        <parameter key="66" value="state\of\previous.residence=
California.true.integer.attribute"/>
        <parameter key="67" value="state\of\previous.residence=
Colorado.true.integer.attribute"/>
        <parameter key="68" value="state\of\previous.residence=
Connecticut.true.integer.attribute"/>
        <parameter key="69" value="state\of\previous.residence=
Delaware.true.integer.attribute"/>
        <parameter key="70" value="state\of\previous.residence= District of
Columbia.true.integer.attribute"/>
        <parameter key="71" value="state\of\previous.residence=
Florida.true.integer.attribute"/>
        <parameter key="72" value="state\of\previous.residence=
Georgia.true.integer.attribute"/>
        <parameter key="73" value="state\of\previous.residence=
Idaho.true.integer.attribute"/>
        <parameter key="74" value="state\of\previous.residence=
Illinois.true.integer.attribute"/>
        <parameter key="75" value="state\of\previous.residence=
Indiana.true.integer.attribute"/>
        <parameter key="76" value="state\of\previous.residence=
Iowa.true.integer.attribute"/>
        <parameter key="77" value="state\of\previous.residence=
Kansas.true.integer.attribute"/>
        <parameter key="78" value="state\of\previous.residence=
Kentucky.true.integer.attribute"/>
        <parameter key="79" value="state\of\previous.residence=
Louisiana.true.integer.attribute"/>
        <parameter key="80" value="state\of\previous.residence=
Maine.true.integer.attribute"/>
        <parameter key="81" value="state\of\previous.residence=
Maryland.true.integer.attribute"/>
        <parameter key="82" value="state\of\previous.residence=
Massachusetts.true.integer.attribute"/>
        <parameter key="83" value="state\of\previous.residence=
Michigan.true.integer.attribute"/>
        <parameter key="84" value="state\of\previous.residence=
Minnesota.true.integer.attribute"/>
        <parameter key="85" value="state\of\previous.residence=
Mississippi.true.integer.attribute"/>
        <parameter key="86" value="state\of\previous.residence=
Missouri.true.integer.attribute"/>
        <parameter key="87" value="state\of\previous.residence=
Montana.true.integer.attribute"/>

```

```

    <parameter key="88" value="state\\.of\\.previous\\.residence=
Nebraska.true.integer.attribute"/>
    <parameter key="89" value="state\\.of\\.previous\\.residence=
Nevada.true.integer.attribute"/>
    <parameter key="90" value="state\\.of\\.previous\\.residence= New
Hampshire.true.integer.attribute"/>
    <parameter key="91" value="state\\.of\\.previous\\.residence= New
Jersey.true.integer.attribute"/>
    <parameter key="92" value="state\\.of\\.previous\\.residence= New
Mexico.true.integer.attribute"/>
    <parameter key="93" value="state\\.of\\.previous\\.residence= New
York.true.integer.attribute"/>
    <parameter key="94" value="state\\.of\\.previous\\.residence= North
Carolina.true.integer.attribute"/>
    <parameter key="95" value="state\\.of\\.previous\\.residence= North
Dakota.true.integer.attribute"/>
    <parameter key="96" value="state\\.of\\.previous\\.residence= Not in
universe.true.integer.attribute"/>
    <parameter key="97" value="state\\.of\\.previous\\.residence=
Ohio.true.integer.attribute"/>
    <parameter key="98" value="state\\.of\\.previous\\.residence=
Oklahoma.true.integer.attribute"/>
    <parameter key="99" value="state\\.of\\.previous\\.residence=
Oregon.true.integer.attribute"/>
    <parameter key="100" value="state\\.of\\.previous\\.residence=
Pennsylvania.true.integer.attribute"/>
    <parameter key="101" value="state\\.of\\.previous\\.residence= South
Carolina.true.integer.attribute"/>
    <parameter key="102" value="state\\.of\\.previous\\.residence= South
Dakota.true.integer.attribute"/>
    <parameter key="103" value="state\\.of\\.previous\\.residence=
Tennessee.true.integer.attribute"/>
    <parameter key="104" value="state\\.of\\.previous\\.residence=
Texas.true.integer.attribute"/>
    <parameter key="105" value="state\\.of\\.previous\\.residence=
Utah.true.integer.attribute"/>
    <parameter key="106" value="state\\.of\\.previous\\.residence=
Vermont.true.integer.attribute"/>
    <parameter key="107" value="state\\.of\\.previous\\.residence=
Virginia.true.integer.attribute"/>
    <parameter key="108" value="state\\.of\\.previous\\.residence= West
Virginia.true.integer.attribute"/>
    <parameter key="109" value="state\\.of\\.previous\\.residence=
Wisconsin.true.integer.attribute"/>
    <parameter key="110" value="state\\.of\\.previous\\.residence=
Wyoming.true.integer.attribute"/>
    <parameter key="111" value="detailed\\.household\\.and\\.family\\.stat= Child &lt;18
ever marr not in subfamily.true.integer.attribute"/>
    <parameter key="112" value="detailed\\.household\\.and\\.family\\.stat= Child &lt;18
ever marr RP of subfamily.true.integer.attribute"/>
    <parameter key="113" value="detailed\\.household\\.and\\.family\\.stat= Child &lt;18
never marr not in subfamily.true.integer.attribute"/>
    <parameter key="114" value="detailed\\.household\\.and\\.family\\.stat= Child &lt;18
never marr RP of subfamily.true.integer.attribute"/>
    <parameter key="115" value="detailed\\.household\\.and\\.family\\.stat= Child &lt;18
spouse of subfamily RP.true.integer.attribute"/>
    <parameter key="116" value="detailed\\.household\\.and\\.family\\.stat= Child 18+
ever marr Not in a subfamily.true.integer.attribute"/>
    <parameter key="117" value="detailed\\.household\\.and\\.family\\.stat= Child 18+
ever marr RP of subfamily.true.integer.attribute"/>
    <parameter key="118" value="detailed\\.household\\.and\\.family\\.stat= Child 18+
never marr Not in a subfamily.true.integer.attribute"/>

```

```

    <parameter key="119" value="detailed\household\and\family\stat= Child 18+
never marr RP of subfamily.true.integer.attribute"/>
    <parameter key="120" value="detailed\household\and\family\stat= Child 18+
spouse of subfamily RP.true.integer.attribute"/>
    <parameter key="121" value="detailed\household\and\family\stat= Child under
18 of RP of unrel subfamily.true.integer.attribute"/>
    <parameter key="122" value="detailed\household\and\family\stat= Grandchild
&lt;18 ever marr not in subfamily.true.integer.attribute"/>
    <parameter key="123" value="detailed\household\and\family\stat= Grandchild
&lt;18 never marr child of subfamily RP.true.integer.attribute"/>
    <parameter key="124" value="detailed\household\and\family\stat= Grandchild
&lt;18 never marr not in subfamily.true.integer.attribute"/>
    <parameter key="125" value="detailed\household\and\family\stat= Grandchild
&lt;18 never marr RP of subfamily.true.integer.attribute"/>
    <parameter key="126" value="detailed\household\and\family\stat= Grandchild
18+ ever marr not in subfamily.true.integer.attribute"/>
    <parameter key="127" value="detailed\household\and\family\stat= Grandchild
18+ ever marr RP of subfamily.true.integer.attribute"/>
    <parameter key="128" value="detailed\household\and\family\stat= Grandchild
18+ never marr not in subfamily.true.integer.attribute"/>
    <parameter key="129" value="detailed\household\and\family\stat= Grandchild
18+ never marr RP of subfamily.true.integer.attribute"/>
    <parameter key="130" value="detailed\household\and\family\stat= Grandchild
18+ spouse of subfamily RP.true.integer.attribute"/>
    <parameter key="131" value="detailed\household\and\family\stat=
Householder.true.integer.attribute"/>
    <parameter key="132" value="detailed\household\and\family\stat= In group
quarters.true.integer.attribute"/>
    <parameter key="133" value="detailed\household\and\family\stat= Nonfamily
householder.true.integer.attribute"/>
    <parameter key="134" value="detailed\household\and\family\stat= Other Rel
&lt;18 ever marr not in subfamily.true.integer.attribute"/>
    <parameter key="135" value="detailed\household\and\family\stat= Other Rel
&lt;18 ever marr RP of subfamily.true.integer.attribute"/>
    <parameter key="136" value="detailed\household\and\family\stat= Other Rel
&lt;18 never marr child of subfamily RP.true.integer.attribute"/>
    <parameter key="137" value="detailed\household\and\family\stat= Other Rel
&lt;18 never marr not in subfamily.true.integer.attribute"/>
    <parameter key="138" value="detailed\household\and\family\stat= Other Rel
&lt;18 never married RP of subfamily.true.integer.attribute"/>
    <parameter key="139" value="detailed\household\and\family\stat= Other Rel
&lt;18 spouse of subfamily RP.true.integer.attribute"/>
    <parameter key="140" value="detailed\household\and\family\stat= Other Rel
18+ ever marr not in subfamily.true.integer.attribute"/>
    <parameter key="141" value="detailed\household\and\family\stat= Other Rel
18+ ever marr RP of subfamily.true.integer.attribute"/>
    <parameter key="142" value="detailed\household\and\family\stat= Other Rel
18+ never marr not in subfamily.true.integer.attribute"/>
    <parameter key="143" value="detailed\household\and\family\stat= Other Rel
18+ never marr RP of subfamily.true.integer.attribute"/>
    <parameter key="144" value="detailed\household\and\family\stat= Other Rel
18+ spouse of subfamily RP.true.integer.attribute"/>
    <parameter key="145" value="detailed\household\and\family\stat= RP of
unrelated subfamily.true.integer.attribute"/>
    <parameter key="146" value="detailed\household\and\family\stat= Secondary
individual.true.integer.attribute"/>
    <parameter key="147" value="detailed\household\and\family\stat= Spouse of
householder.true.integer.attribute"/>
    <parameter key="148" value="detailed\household\and\family\stat= Spouse of RP
of unrelated subfamily.true.integer.attribute"/>
    <parameter key="149" value="family\members\and\family\stat= Both parents
present.true.integer.attribute"/>

```

```

        <parameter key="150" value="family\members\.under\18= Father only
present.true.integer.attribute"/>
        <parameter key="151" value="family\members\.under\18= Mother only
present.true.integer.attribute"/>
        <parameter key="152" value="family\members\.under\18= Neither parent
present.true.integer.attribute"/>
        <parameter key="153" value="family\members\.under\18= Not in
universe.true.integer.attribute"/>
        <parameter key="154" value="country\.of\.birth\.self=
Panama.true.integer.attribute"/>
        <parameter key="155" value="country\.of\.birth\.self= South
Korea.true.integer.attribute"/>
        <parameter key="156" value="country\.of\.birth\.self=British-
Commonwealth.true.integer.attribute"/>
        <parameter key="157"
value="country\.of\.birth\.self=China.true.integer.attribute"/>
        <parameter key="158"
value="country\.of\.birth\.self=Euro_1.true.integer.attribute"/>
        <parameter key="159"
value="country\.of\.birth\.self=Euro_2.true.integer.attribute"/>
        <parameter key="160" value="country\.of\.birth\.self=Latin-
America.true.integer.attribute"/>
        <parameter key="161"
value="country\.of\.birth\.self=Other.true.integer.attribute"/>
        <parameter key="162" value="country\.of\.birth\.self=SE-
Asia.true.integer.attribute"/>
        <parameter key="163" value="country\.of\.birth\.self=South-
America.true.integer.attribute"/>
        <parameter key="164" value="country\.of\.birth\.self=United-
States.true.integer.attribute"/>
        <parameter key="165"
value="country\.of\.birth\.self=Unknown.true.integer.attribute"/>
        <parameter key="166" value="citizenship= Foreign born- Not a citizen of U S
.true.integer.attribute"/>
        <parameter key="167" value="citizenship= Foreign born- U S citizen by
naturalization.true.integer.attribute"/>
        <parameter key="168" value="citizenship= Native- Born abroad of American
Parent(s).true.integer.attribute"/>
        <parameter key="169" value="citizenship= Native- Born in Puerto Rico or U S
Outlying.true.integer.attribute"/>
        <parameter key="170" value="citizenship= Native- Born in the United
States.true.integer.attribute"/>
        <parameter key="171" value="age=Young.true.integer.attribute"/>
        <parameter key="172" value="age=Middle-aged.true.integer.attribute"/>
        <parameter key="173" value="age=Senior.true.integer.attribute"/>
        <parameter key="174" value="age=Old.true.integer.attribute"/>
        <parameter key="175" value="capital\gains=None.true.integer.attribute"/>
        <parameter key="176" value="capital\gains=Low.true.integer.attribute"/>
        <parameter key="177" value="capital\gains=High.true.integer.attribute"/>
    </list>
    <parameter key="read_not_matching_values_as_missings" value="true"/>
    <parameter key="datamanagement" value="double_array"/>
</operator>
</process>
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<process version="6.5.002">
    <operator activated="true" class="select_attributes" compatibility="6.5.002"
expanded="true" height="76" name="Select Attributes" width="90" x="246" y="165">
        <parameter key="attribute_filter_type" value="subset"/>
        <parameter key="attribute" value=""/>
        <parameter key="attributes" value="age=Middle-
aged|age=Old|age=Senior|age=Young|capital.gains=High|capital.gains=Low|capital.gains=N
one|citizenship= Foreign born- Not a citizen of U S|citizenship= Foreign born- U S

```

```

citizen by naturalization|citizenship= Native- Born abroad of American
Parent(s)|citizenship= Native- Born in Puerto Rico or U S Outlying|citizenship=
Native- Born in the United States|country.of.birth.self= Panama|country.of.birth.self=
South Korea|country.of.birth.self=British-
Commonwealth|country.of.birth.self=China|country.of.birth.self=Euro_1|country.of.birth
.self=Euro_2|country.of.birth.self=Latin-
America|country.of.birth.self=Other|country.of.birth.self=SE-
Asia|country.of.birth.self=South-America|country.of.birth.self=United-
States|country.of.birth.self=Unknown|education= Bachelors degree(BA AB BS)|education=
Doctorate degree(PhD EdD)|education= Masters degree(MA MS MEng MEd MSW MBA)|education=
Prof school degree (MD DDS DVM LLB
JD)|education=Associates|education=Dropout|education=HS-Graduate|race=Amer-
Indian|race=Asian|race=Black|race=Other|race=White|sex= Female|sex= Male|race=
White|race= Other|race= Black|race= Asian or Pacific Islander|race= Amer Indian Aleut
or Eskimo|country.of.birth.self= Yugoslavia|country.of.birth.self=
Vietnam|country.of.birth.self= United-States|country.of.birth.self=
Trinidad&Tobago|country.of.birth.self= Thailand|country.of.birth.self=
Taiwan|country.of.birth.self= Scotland|country.of.birth.self= Puerto-
Rico|country.of.birth.self= Portugal|country.of.birth.self=
Poland|country.of.birth.self= Philippines|country.of.birth.self=
Peru|country.of.birth.self= Outlying-U S (Guam USVI etc)|country.of.birth.self=
Nicaragua|country.of.birth.self= Mexico|country.of.birth.self=
Laos|country.of.birth.self= Japan|country.of.birth.self=
Jamaica|country.of.birth.self= Italy|country.of.birth.self=
Ireland|country.of.birth.self= Iran|country.of.birth.self=
India|country.of.birth.self= Hungary|country.of.birth.self= Hong
Kong|country.of.birth.self= Honduras|country.of.birth.self= Holand-
Netherlands|country.of.birth.self= Haiti|country.of.birth.self=
Guatemala|country.of.birth.self= Greece|country.of.birth.self=
Germany|country.of.birth.self= France|country.of.birth.self=
England|country.of.birth.self= El-Salvador|country.of.birth.self=
Ecuador|country.of.birth.self= Dominican-Republic|country.of.birth.self=
Cuba|country.of.birth.self= Columbia|country.of.birth.self=
China|country.of.birth.self= Canada|country.of.birth.self= Cambodia|education= Some
college but no degree|education= Less than 1st grade|education= High school
graduate|education= Children|education= Associates degree-occup /vocational|education=
Associates degree-academic program|education= 9th grade|education= 7th and 8th
grade|education= 5th or 6th grade|education= 1st 2nd 3rd or 4th grade|education= 12th
grade no diploma|education= 11th grade|education= 10th grade"/>
    <parameter key="use_except_expression" value="false"/>
    <parameter key="value_type" value="attribute_value"/>
    <parameter key="use_value_type_exception" value="false"/>
    <parameter key="except_value_type" value="time"/>
    <parameter key="block_type" value="attribute_block"/>
    <parameter key="use_block_type_exception" value="false"/>
    <parameter key="except_block_type" value="value_matrix_row_start"/>
    <parameter key="invert_selection" value="false"/>
    <parameter key="include_special_attributes" value="false"/>
</operator>
</process>
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<process version="6.5.002">
    <operator activated="true" class="replace_missing_values" compatibility="6.5.002"
expanded="true" height="94" name="Replace Missing Values" width="90" x="447" y="165">
        <parameter key="return_preprocessing_model" value="false"/>
        <parameter key="create_view" value="false"/>
        <parameter key="attribute_filter_type" value="all"/>
        <parameter key="attribute" value=""/>
        <parameter key="attributes" value=""/>
        <parameter key="use_except_expression" value="false"/>
        <parameter key="value_type" value="attribute_value"/>
        <parameter key="use_value_type_exception" value="false"/>
        <parameter key="except_value_type" value="time"/>

```

```

    <parameter key="block_type" value="attribute_block"/>
    <parameter key="use_block_type_exception" value="false"/>
    <parameter key="except_block_type" value="value_matrix_row_start"/>
    <parameter key="invert_selection" value="false"/>
    <parameter key="include_special_attributes" value="false"/>
    <parameter key="default" value="zero"/>
    <list key="columns"/>
  </operator>
</process>
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<process version="6.5.002">
  <operator activated="true" class="k_means" compatibility="6.5.002" expanded="true"
height="76" name="Clustering" width="90" x="581" y="165">
    <parameter key="add_cluster_attribute" value="true"/>
    <parameter key="add_as_label" value="false"/>
    <parameter key="remove_unlabeled" value="false"/>
    <parameter key="k" value="10"/>
    <parameter key="max_runs" value="10"/>
    <parameter key="determine_good_start_values" value="false"/>
    <parameter key="measure_types" value="BregmanDivergences"/>
    <parameter key="mixed_measure" value="MixedEuclideanDistance"/>
    <parameter key="nominal_measure" value="NominalDistance"/>
    <parameter key="numerical_measure" value="EuclideanDistance"/>
    <parameter key="divergence" value="SquaredEuclideanDistance"/>
    <parameter key="kernel_type" value="radial"/>
    <parameter key="kernel_gamma" value="1.0"/>
    <parameter key="kernel_sigma1" value="1.0"/>
    <parameter key="kernel_sigma2" value="0.0"/>
    <parameter key="kernel_sigma3" value="2.0"/>
    <parameter key="kernel_degree" value="3.0"/>
    <parameter key="kernel_shift" value="1.0"/>
    <parameter key="kernel_a" value="1.0"/>
    <parameter key="kernel_b" value="0.0"/>
    <parameter key="max_optimization_steps" value="100"/>
    <parameter key="use_local_random_seed" value="false"/>
    <parameter key="local_random_seed" value="1992"/>
  </operator>
</process>
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<process version="6.5.002">
  <operator activated="true" class="extract_prototypes" compatibility="6.5.002"
expanded="true" height="76" name="Extract Cluster Prototypes" width="90" x="782"
y="210"/>
</process>
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<process version="6.5.002">
  <operator activated="true" class="write_csv" compatibility="6.5.002" expanded="true"
height="76" name="Write CSV" width="90" x="983" y="210">
    <parameter key="csv_file" value="C:\Users\Kostas\Desktop\11.csv"/>
    <parameter key="column_separator" value=";" />
    <parameter key="write_attribute_names" value="true"/>
    <parameter key="quote_nominal_values" value="true"/>
    <parameter key="format_date_attributes" value="true"/>
    <parameter key="append_to_file" value="false"/>
    <parameter key="encoding" value="SYSTEM"/>
  </operator>
</process>

```