**A report on**

**INFO-H423 - Data Mining. Assignment – part 1.**

By

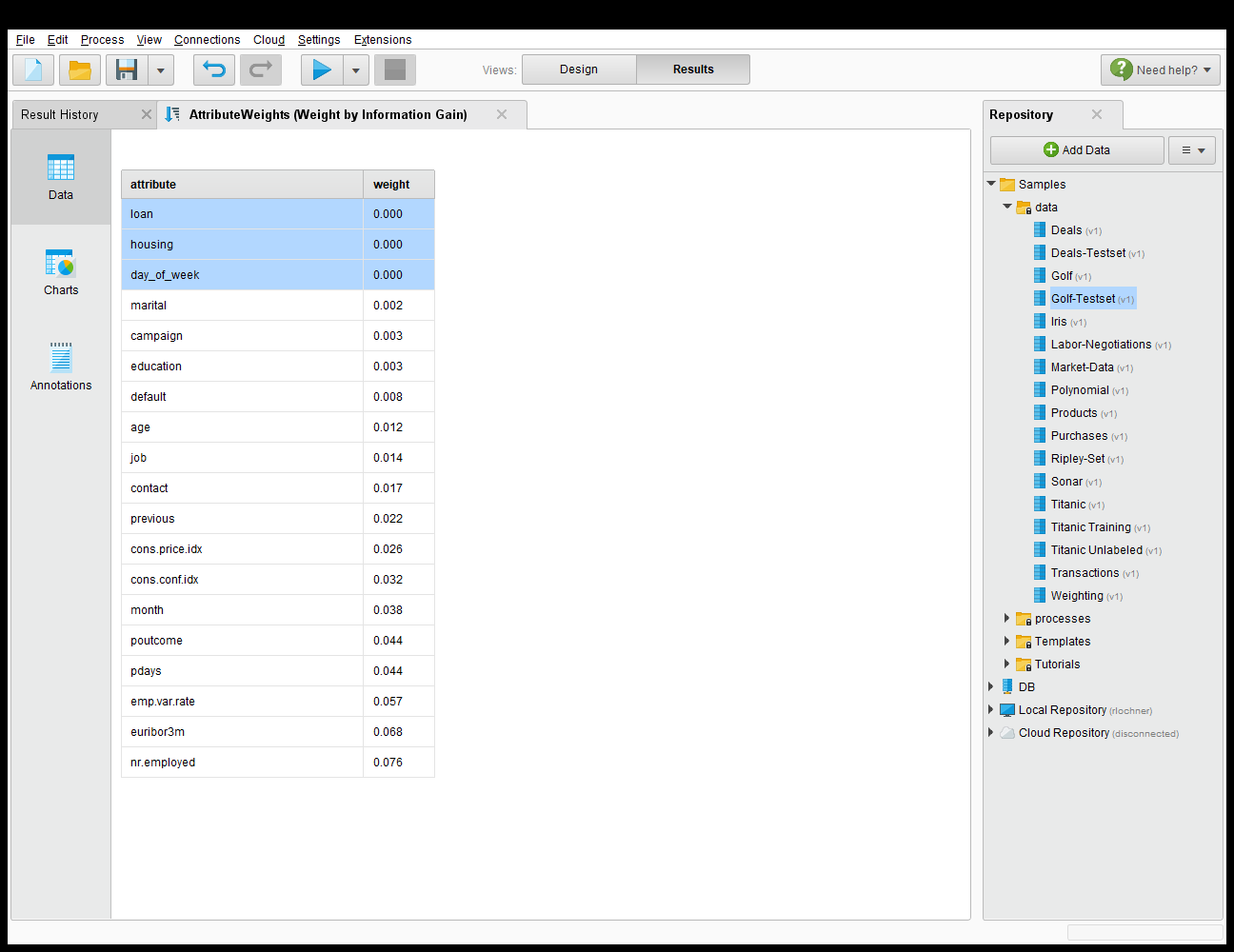
Raymond Lochner (???)

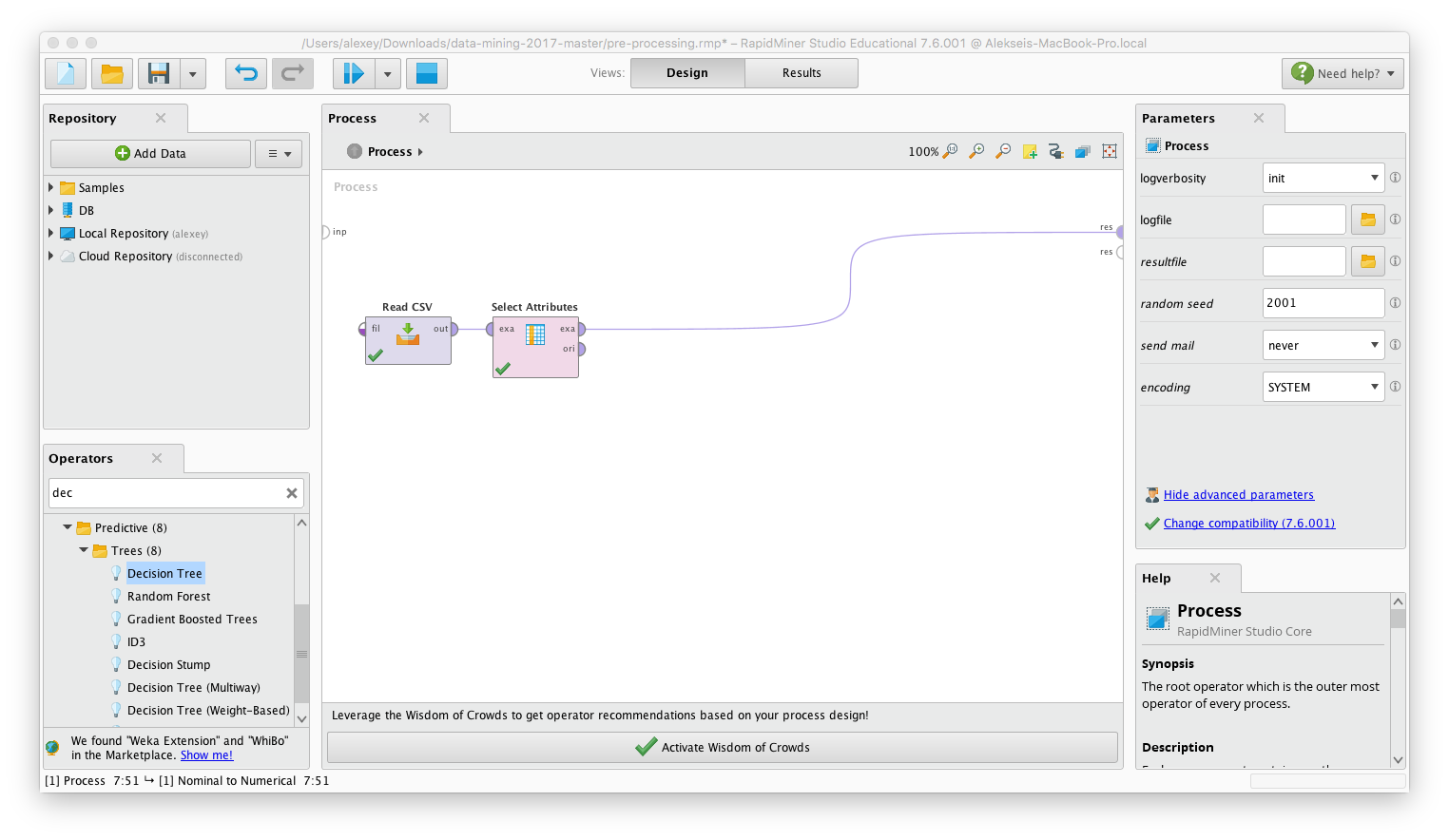
Aleksei Karetnkiov (000455065)

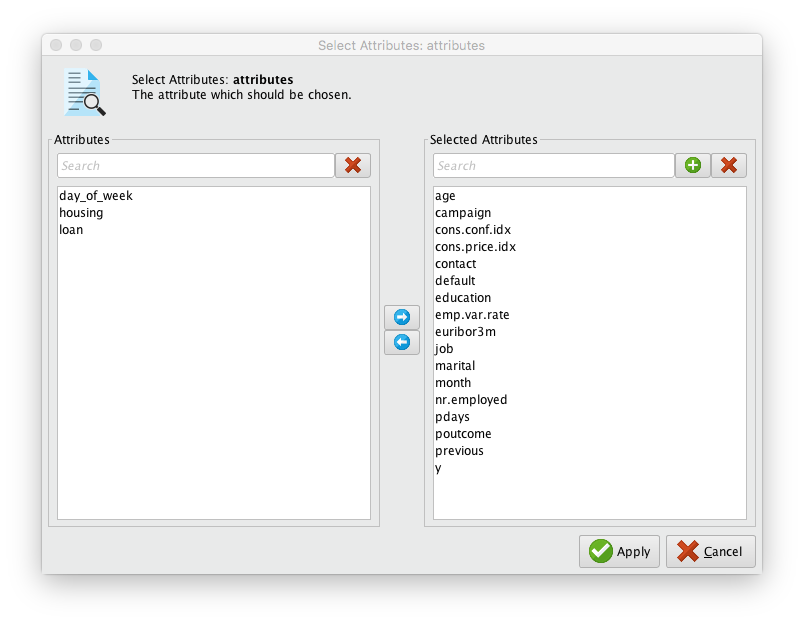
Aldar Saranov (000435170)

# Preprocessing Stage

The input file has been successfully loaded and the appropriate roles have been assigned. In order to detect and remove attributes which do not contribute to the prediction, we measure the information gain for each attribute. We find that certain attributes have negligible information gain (load, housing and day\_of\_week).







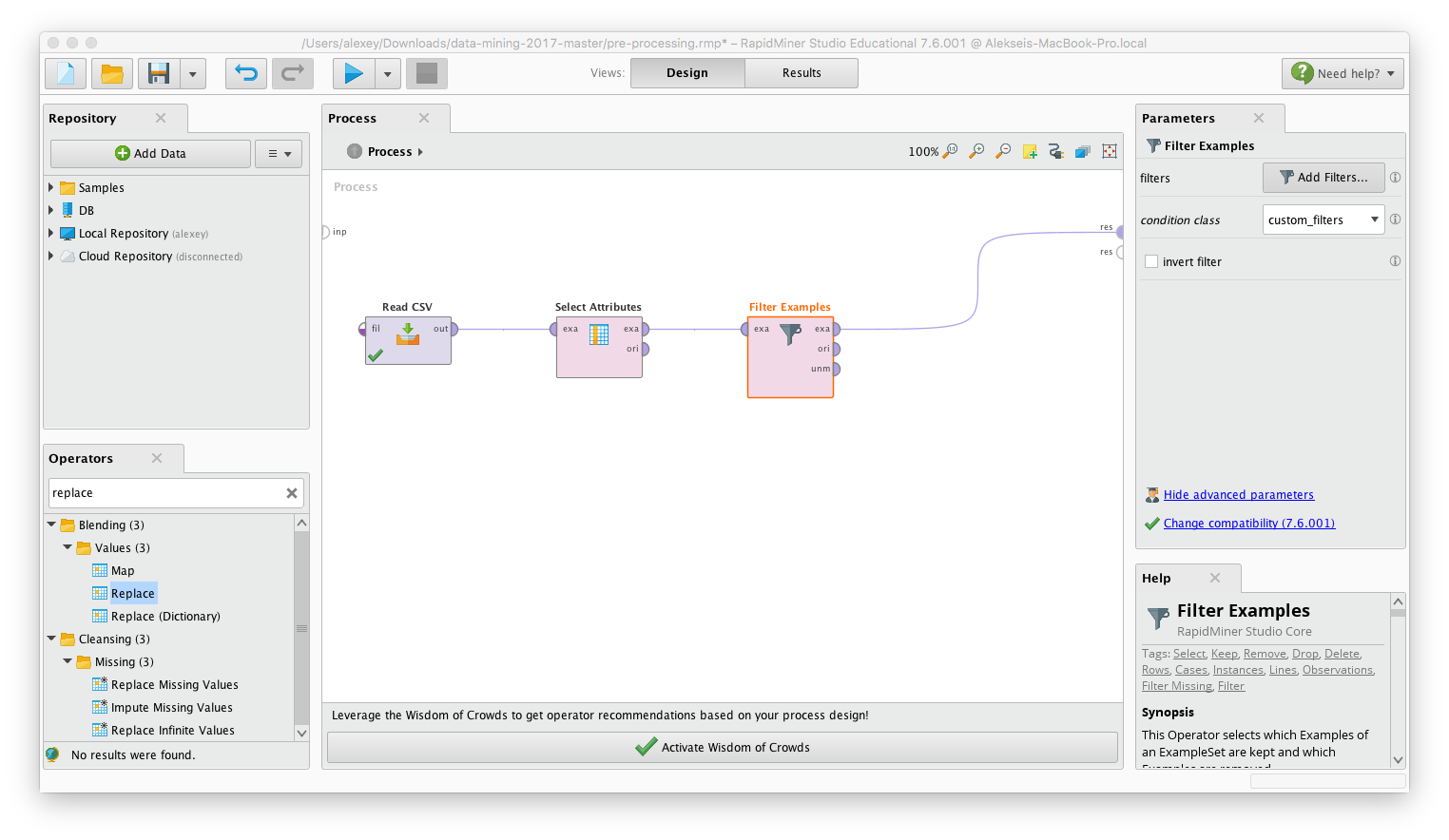
Then, we need to consider the “unknown” values. On condition that our dataset has 41.188 entries we suppose that it is enough to have a 5% significance level, which is about 2.000 entries. It is necessary to optimize the task without any significant influence on the result of the whole research. So, we can remove the entries, which has smaller number of “unknown” values and recovery the others. To recover the damaged values we will use the further method of the interpolation, which based on the correlation of values with values of other attributes:

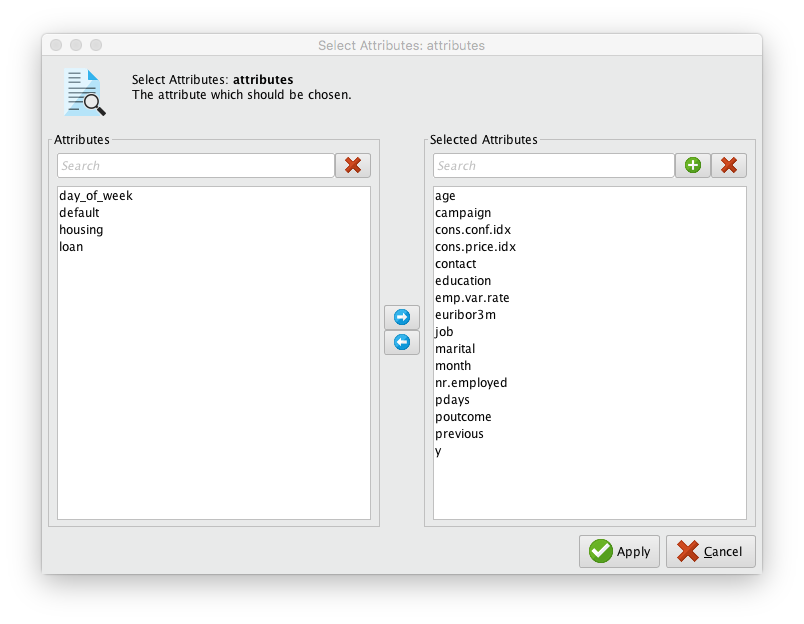
1. We need to find the maximal absolute value of the correlation with other attributes;
2. Consider the sign of the correlation coefficient;
3. Find a pattern;
4. Replace these “unknown” values by the found pattern.

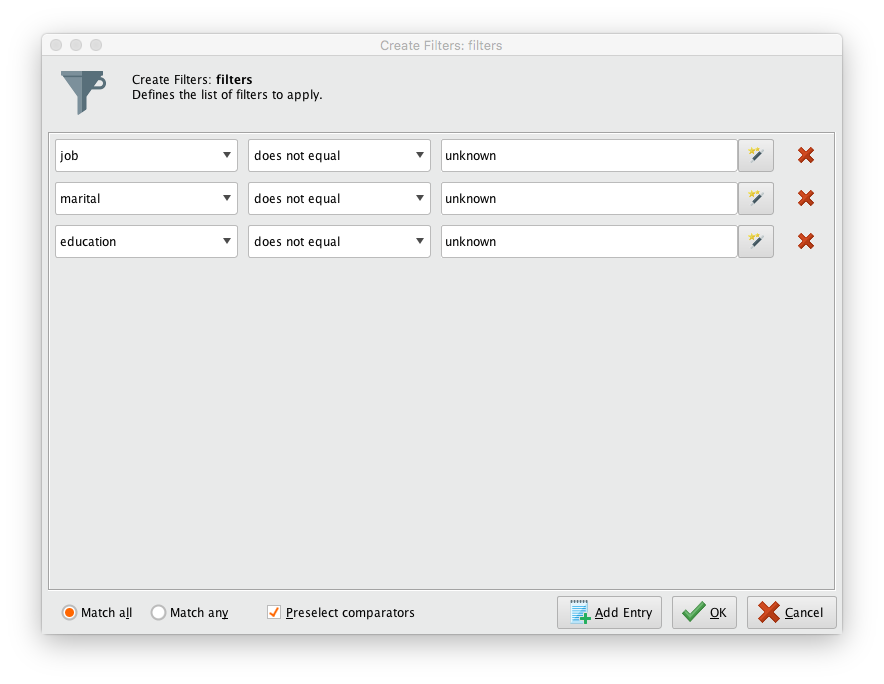
We need to recognize all the attributes with the possible “unknown” values, considering that attributes “load”, “housing” and “day\_of\_week” excluded of the research on the previous step.

|  |  |  |
| --- | --- | --- |
| Attribute | “Unknown” values | Ratio of the “unknown” values |
| Job | 330 | 0.008 |
| Marital | 80 | 0.002 |
| Education | 1731 | 0.042 |
| Default | 8597 | 0.209 |

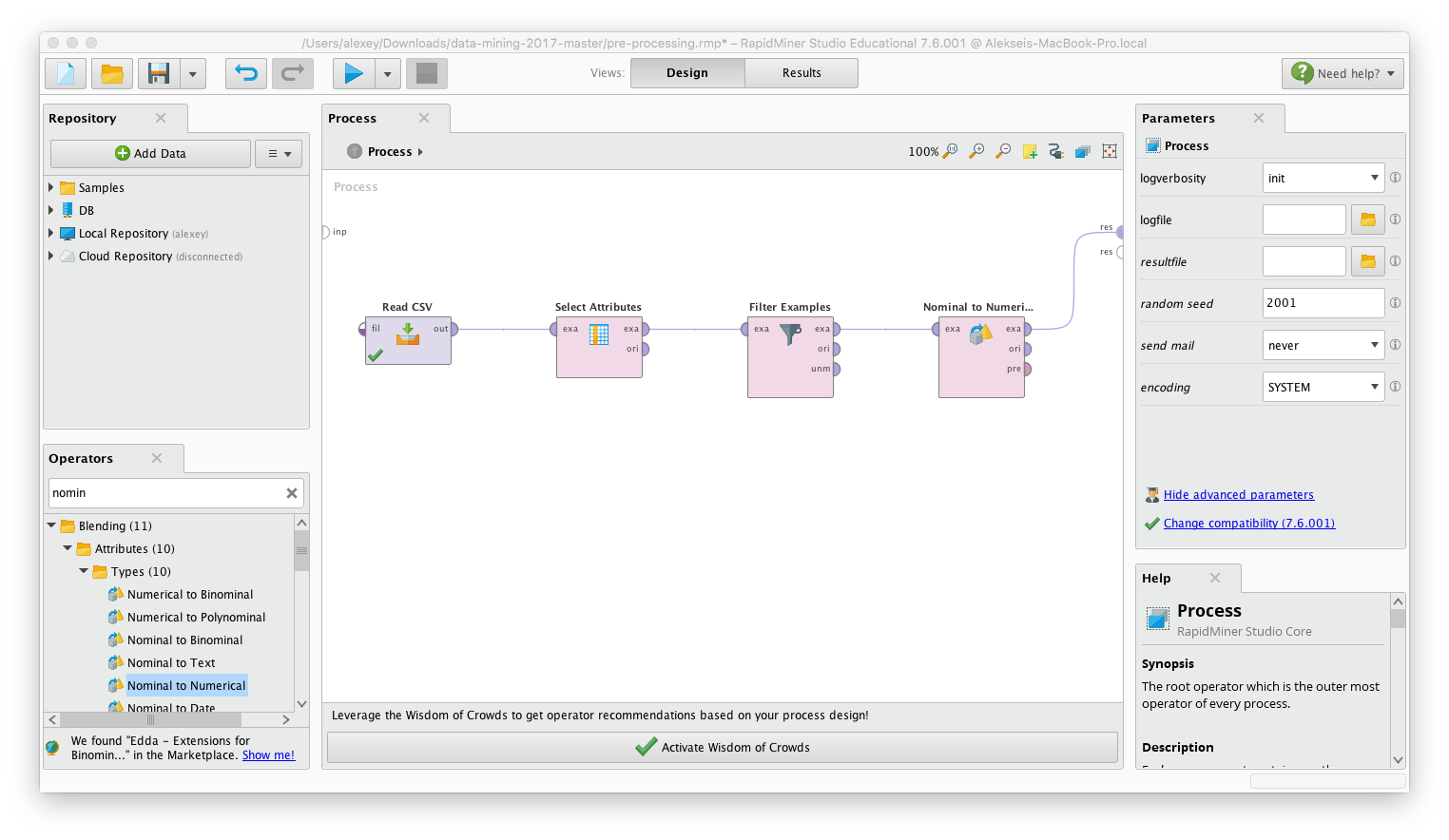
Regarding the results, we can see the we need to work only with “default” attribute and remove other entries to optimize the task. Regarding this attribute, we can see that we have 32.588 “no” (79.25% of the all entries) and 3 “yes” (0.007%) values. Unfortunately, this set is not enough for building any model. So, we can assume that other “unknown” values could be “no”. Since, more than 99% of the “default” values are the same, we could also exclude this attribute, because it will be useless.







Finally, we have to replace all the nominal values by corresponding numerical to optimize the dataset for the further analysis. We will save the resulting preprocessed dataset to the local repository to simplify access to it.



# Data Mining Stage

## K-medoids

The k-medoids methods provides us the opportunity to make a clusterisation of the dataset with randomly selected its centers. According to the task, we have to take 70% of the prepared data as “training set” and 30% as a “test set”. So, we need to use “Split Data” operator. Then, we have to remove the “y” attribute of the “test set” for further applying the trained model to this subset. We have to reduce number of steps for the operators’ runs and optimization steps because of the high computation costs. So, we will use 3 runs and 3 optimization steps. It is possible to increase number of steps for better accuracy in the future. In comparison with k-means method, the k-medoid provide us more balanced and distributed result.

Regarding the process, we will start from the previously prepared dataset, which had been divided into 2 subsets: “training set” and “test set”. So, we need to run the following operators:

1. “K-Medoids” clustering with the following parameters (all the parameters have the default value if it is not mentioned):
   * k=2 (number of the resulting clusters)
   * max runs = 3 (as already mentioned, we need to optimize our calculations)
   * max optimization=3 (as already mentioned, we need to optimize our calculations)
   * Measure type=NumericalMeasures (because recently we have converted all the Nominal values to Numerical)
   * Mixed measure=MixedEuclideanDistance.
2. “Select Attributes” operator as a parallel action to remove the “y” attribute from the “test set”. After that, two output streams are going to the inputs of the “Apply model” and “Rename” operators to apply the calculated model for this dataset.
   1. Then we have to “Apply model” to the whole dataset.
      1. After that it is necessary to set role of the cluster with target role “regular”;
      2. Then we need to apply the “Map” operator;
      3. After that we changed the role of the cluster to “prediction”;
      4. Finally, we set new name of the predicted value from “cluster” to “prediction(y)”;
   2. As a parallel process, we need to rename the left attribute;
      1. Then we changed all the numerical values back to nominal;
      2. Finally, we set role of the “y” attribute as “label”;
3. Inner Join operator, which is aimed to join the predicted and original attributes;
4. Finally, we are recognizing the performance of the applied k-medoids method to estimate the accuracy of final data.

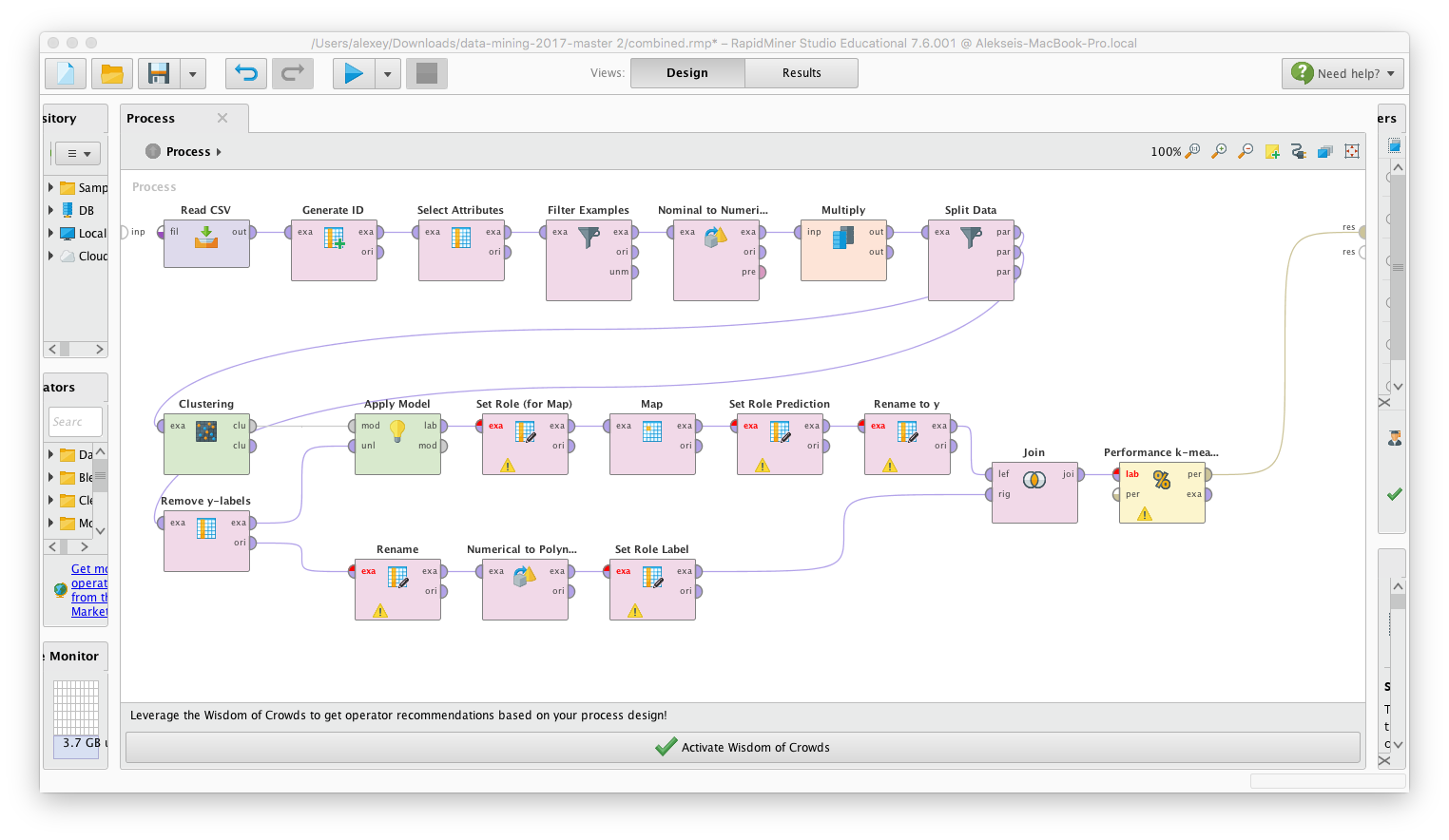


Figure The whole data-mining process

# Models evaluation

## K-medoids

After the calculations, we have received two cluster, which were selected by k-medoids method and received the confusion matrix after the performance estimation. To estimate the data, we will use the next metrics (we have no prescribed values, so we will estimate the method by comparing its results with the results of other recognized method):

* Accuracy;
* Precision coefficient;
* Recall coefficient;
* Calculation time;

After 2 runs with 2 optimizations we have received only 10.24% of accuracy, which means that it is necessary to increase number of runs to receive better result. It happened because of the k-medoids method principals, which are based on the randomly selected medoids.

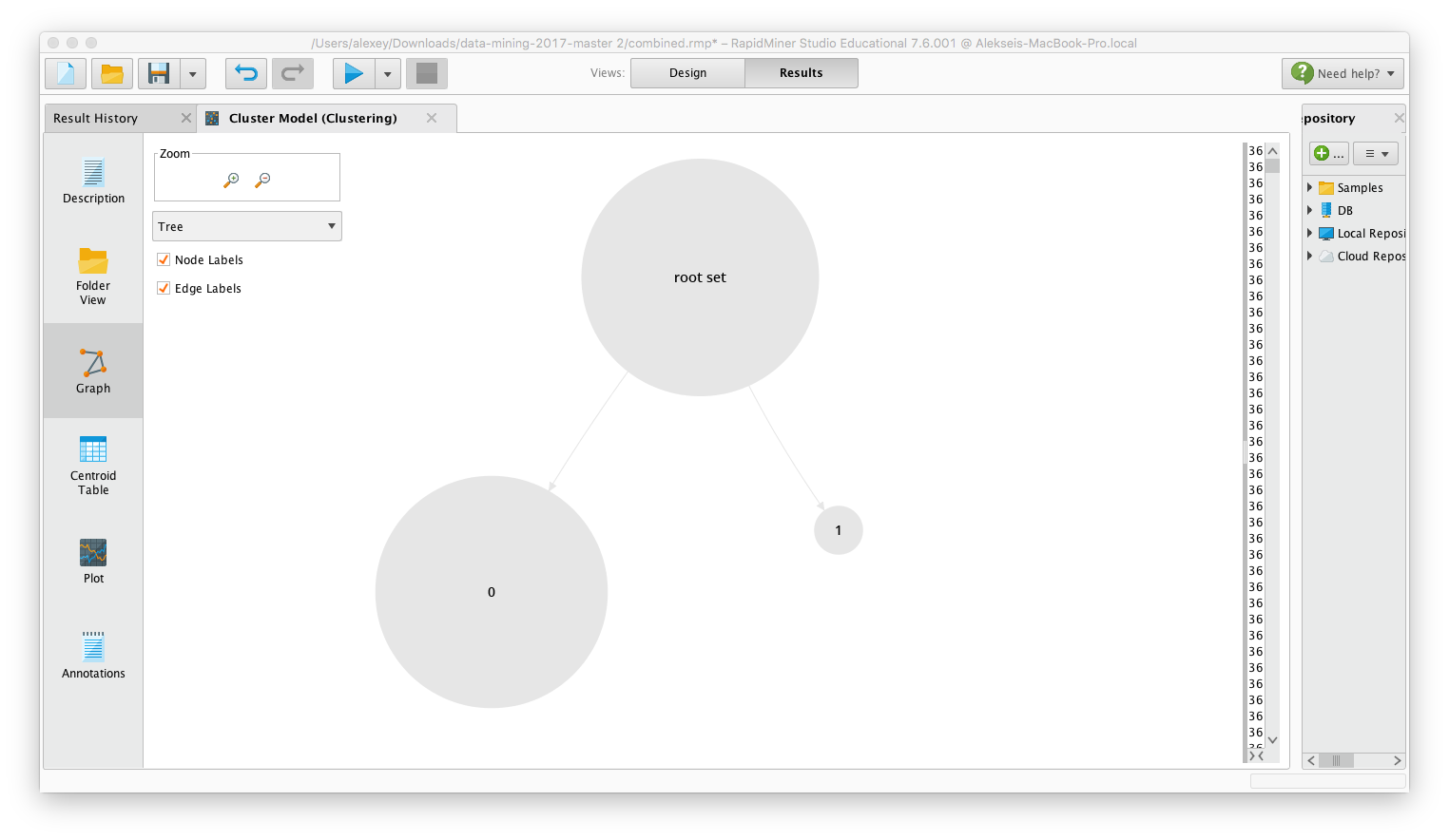


Figure . Clusters visualisation

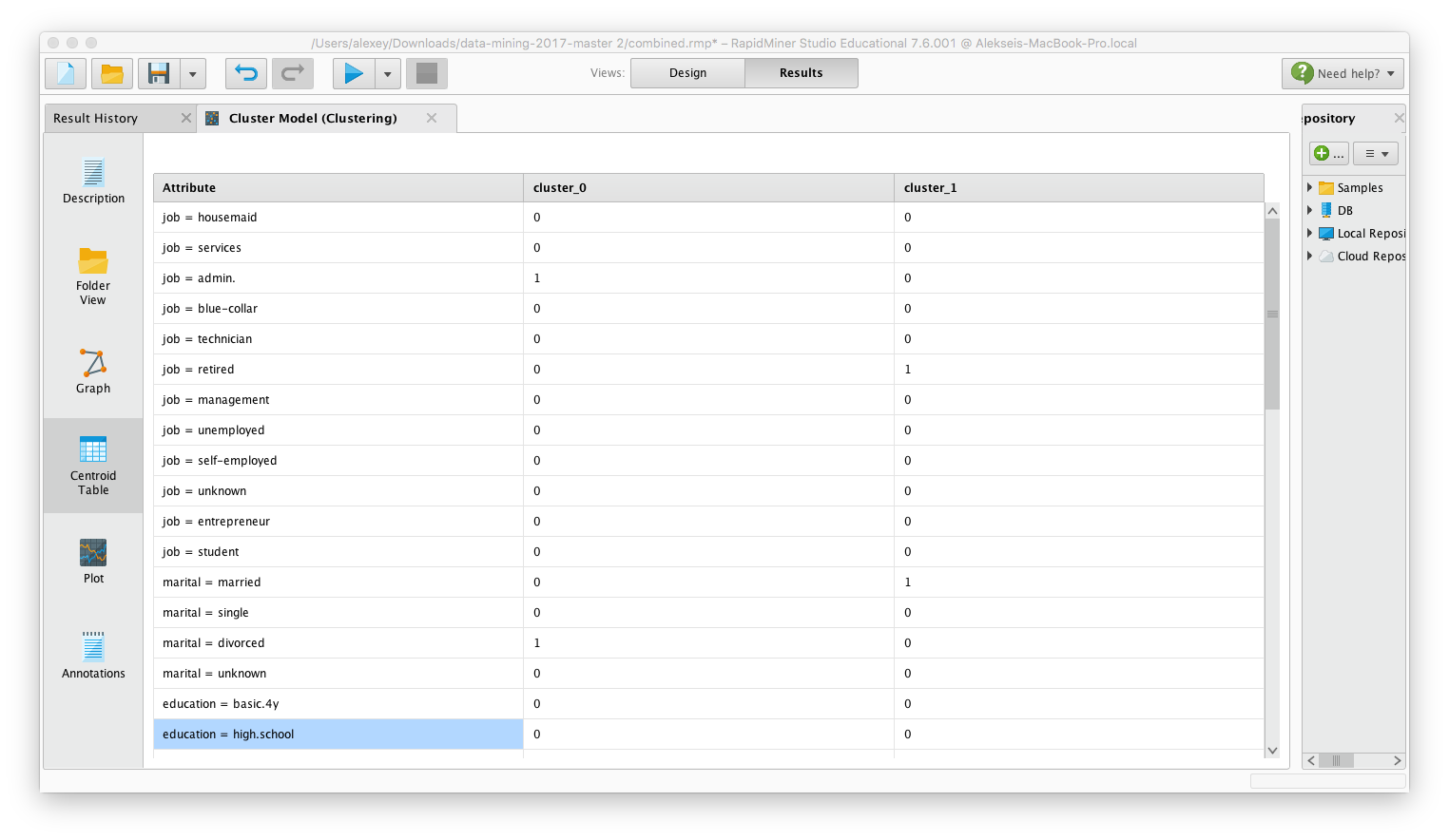


Figure Predicted conditions of data for two clusters

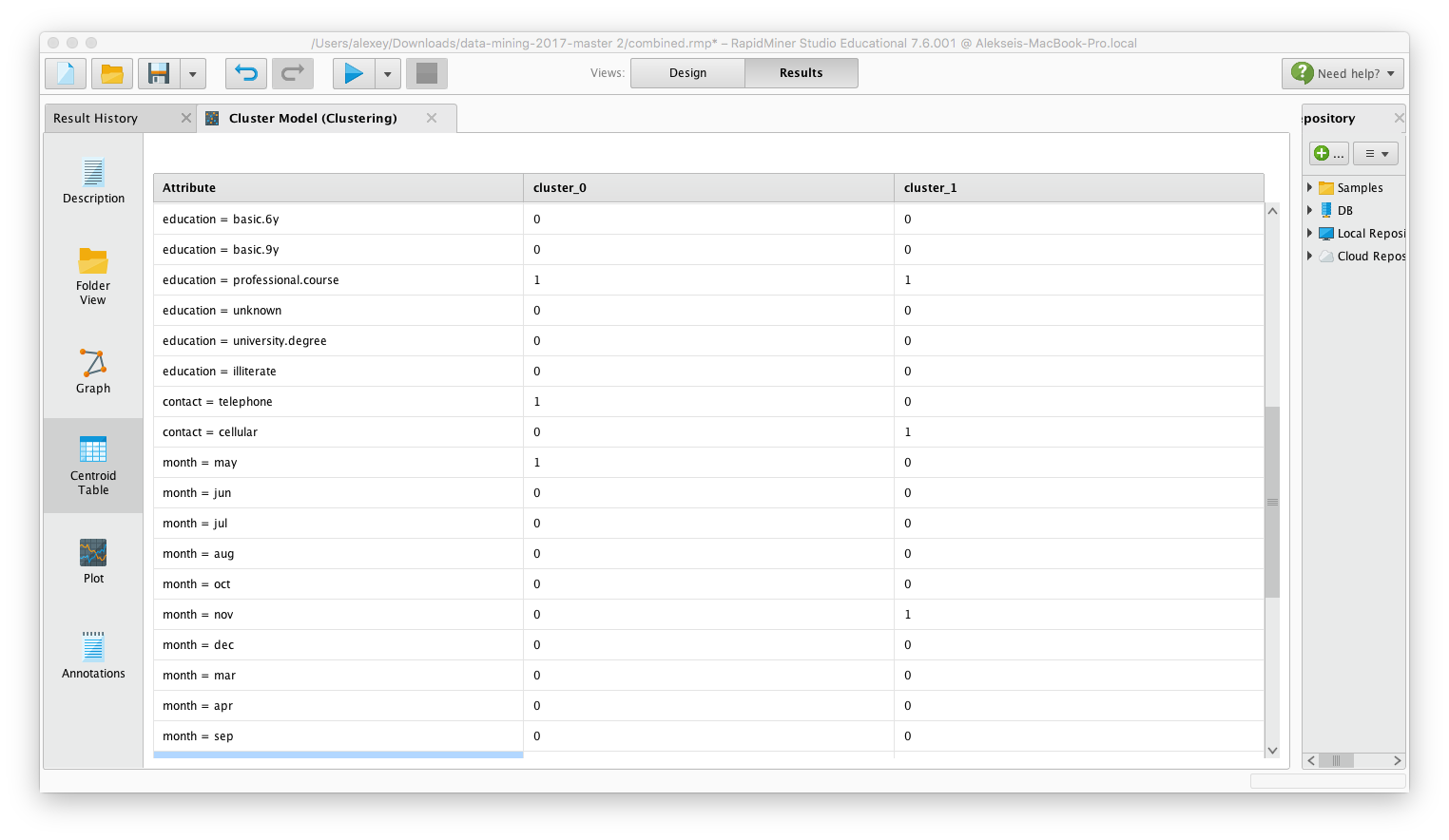


Figure Predicted conditions of data for two clusters

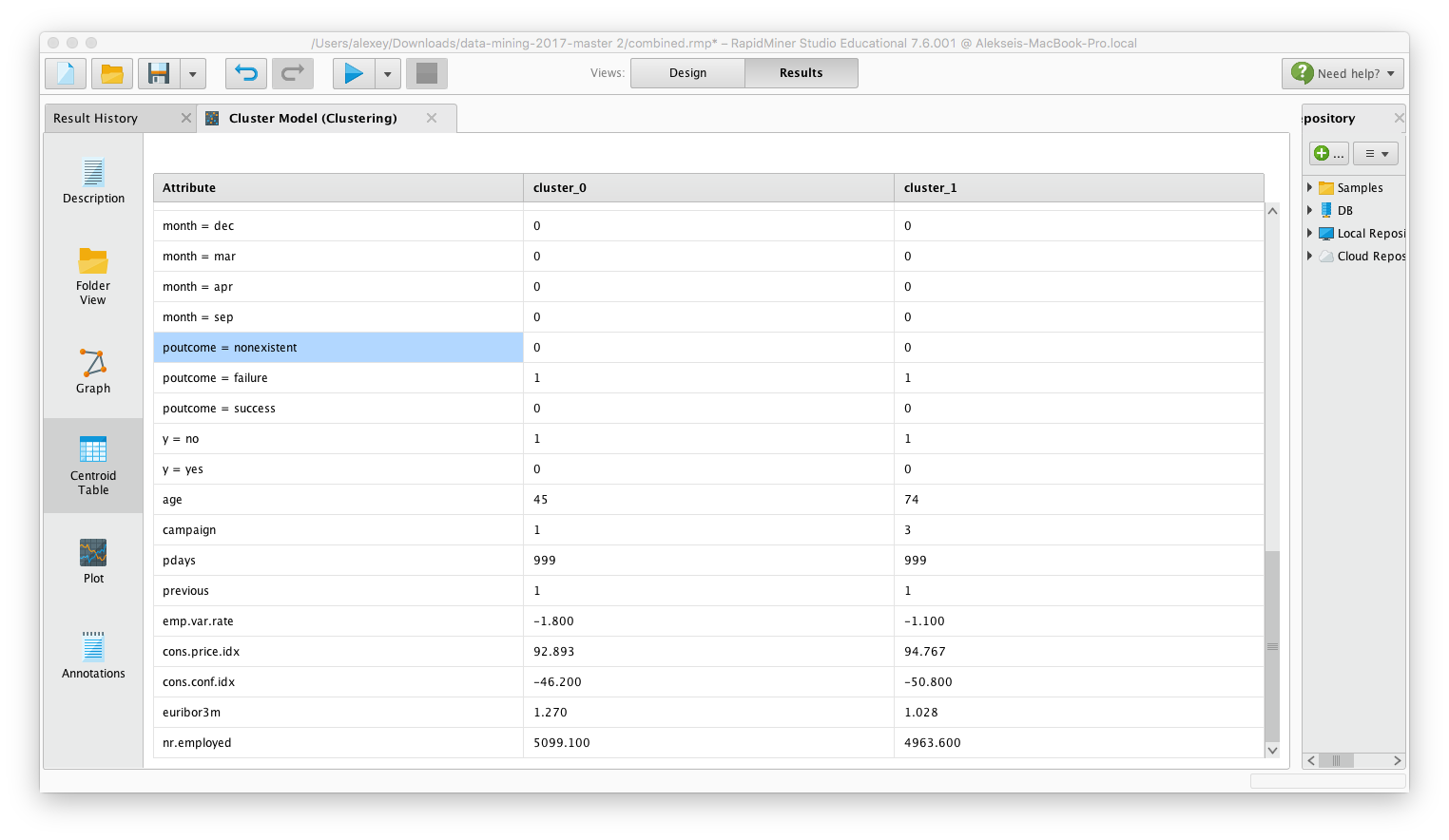


Figure Predicted conditions of data for two clusters

According to the results, we can see that from the whole population of “test set”, which includes 11.575 records, the system predicted that value of our recognized attribute “y” as “no” in 153 records, while “yes” in 10310 records. Similarly, we can see that in 243 cases the system predicted the “y” answer as “n”, and 1051 other records. We can see that the result is better in comparison with the negative answers. So, we can see the cost-sensitive measures of our model. We have received 1.46% of class recall coefficient for the negative answers and 81.22% for positive ones respectively. Also, we can see the precision evaluation of the selected method, which has 38.64% of precision for all the predicted negative answers and 9.25% for positive ones.

The whole data mining process has taken 3 hours for this stage only, so, it is a poor result in comparison with

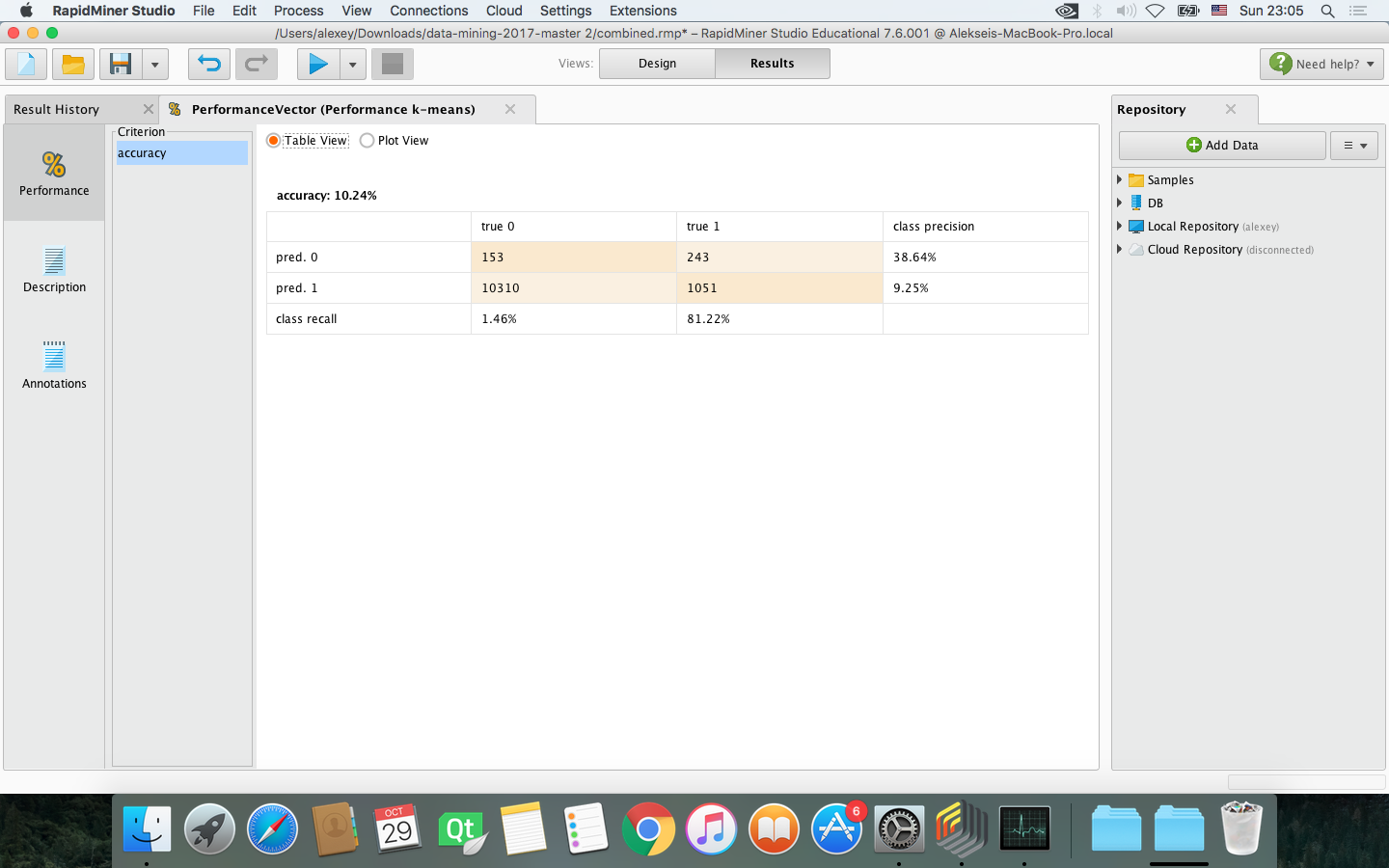


Figure Estimation of the process

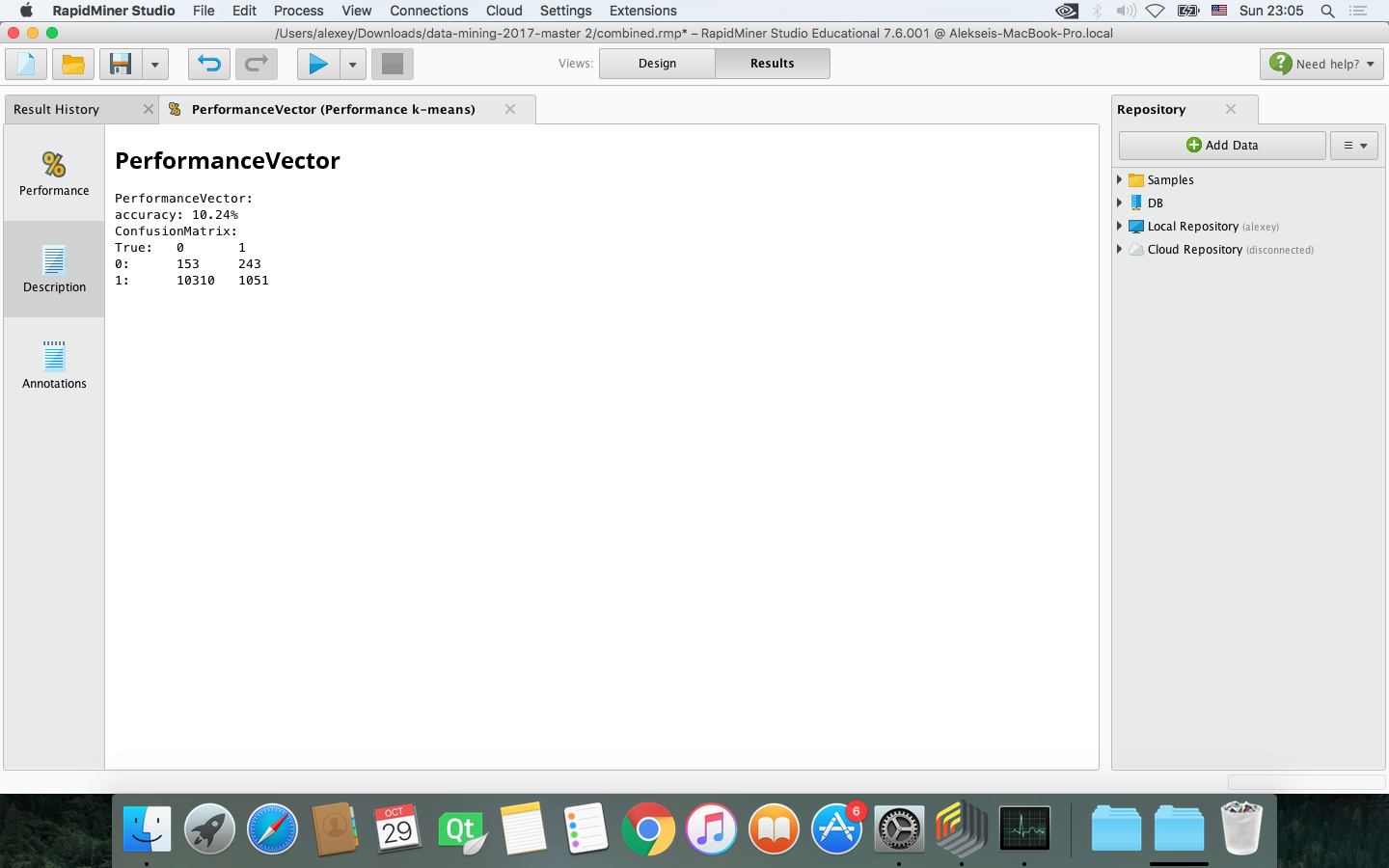


Figure Estimation of the process

Finally, this method provides us an opportunity to divide the data into two clusters (in this case) which are more normalized in comparison with k-means. Unfortunately, this method takes significantly more time to calculate the result, so, it is cannot be applied in time-dependent tasks. We have estimated this method with 2 runs and 2 optimization steps, which is not enough (according to the result). As a result, we received non-acceptable value of accuracy. Moreover, it is important that this method takes significantly more time to calculate than others, so, it is not the best method for given task.