DETERMINING THE REASON FOR UNEMPLOYMENT

ABSTRACT:

The Project that I have taken is related to something with the current issues. The main question that we are focussing in our project is why do engineer loose or doesn’t get their initial job yet. I used Data Mining in order to find the pattern using a large data set. Finally, I could analyse if the person/student has these qualification they are fit or they will get a job for sure and vice versa if he/she does not have such qualification.

INTRODUCTION:

This project analyses if the job searcher, the job loser, or the different categorical person that will be mentioned below will finally get a job or not. There are different criteria that will be analysed in order to find if the person with specific qualification can survive with a job or not. This project is totally about predicting. I have used five predicting algorithms each are better than other is been chosen to predict the data set that we have taken.

DATA SET:

The **UNEMPLOYMENT** Data set that is been used. The Data set is taken from the git hub repository. The attributes of the data set are as follows.

Description

A cross-section from 1993

Number of observations: 452

Observation: individuals

Country: United States

Usage

Data (Unemployment)

Format

A data frame containing:

Duration

Duration of first spell of unemployment, t, in weeks

Spell

1 if spell is complete

Race

One of non-white, white

Sex

One of male, female

Reason

Reason for unemployment, one of new (new entrant), lose (job loser), leave (job leaver), re-enter (labour force re-entrant)

Search

'yes' if (1) the unemployment spell is completed between the first and second surveys and number of methods used to search > average number of methods used across all records in the sample, or, (2) for individuals who remain unemployed for consecutive surveys, if the number of methods used is strictly no decreasing at all survey points, and is strictly increasing at least at one survey point.

Pubemp

'Yes' if an individual used a public employment agency to search for work at any survey points relating to the individuals first unemployment spell.

ftp1

1 if an individual is searching for full time work at survey 1.

ftp2

1 if an individual is searching for full time work at survey 2.

ftp3

1 if an individual is searching for full time work at survey 3.

ftp4

1 if an individual is searching for full time work at survey 4.

Nobs

Number of observations on the first spell of unemployment for the record.

ANALYSIS:

The Unemployment analysis, which will predict if he/she will get a job with the provided qualification and skills, they have.

ALGORITHMS:

- **Decision Tree**
- **Random Forest**
- **K means**
- **Naïve Bayes**
- **KNN**

DECISION TREE:

Decision tree learning uses a decision tree to go from observations about an item to conclusions about the item's target value. It is one of the predictive modelling approaches used in statistics, data mining and machine learning.

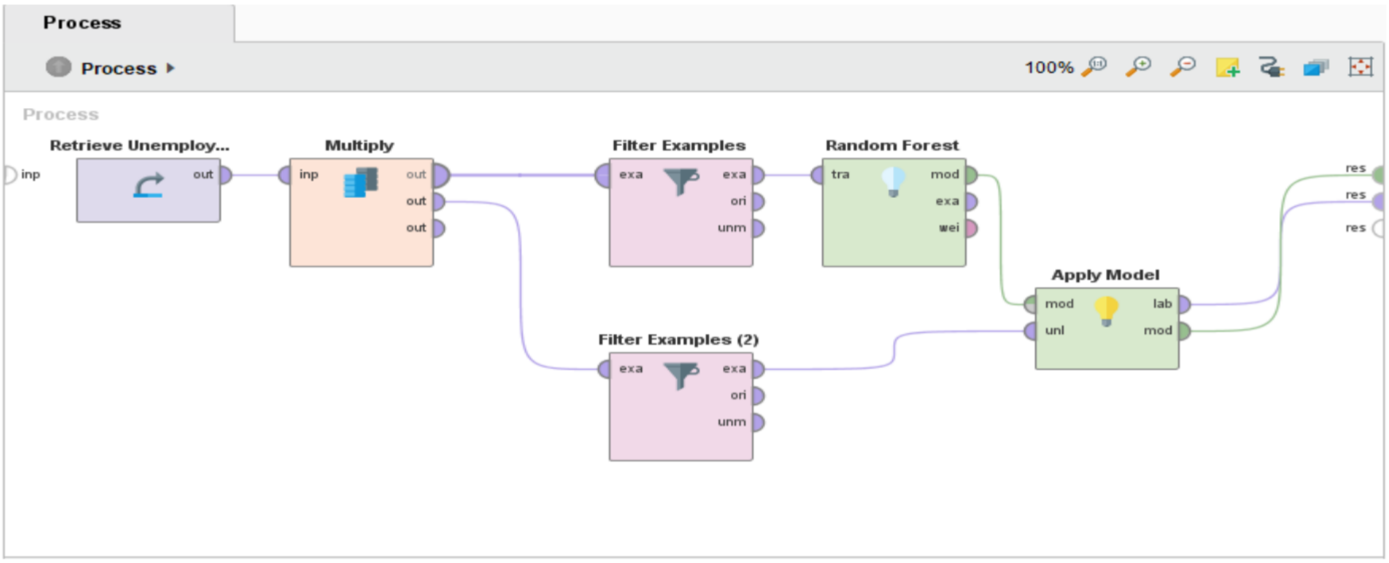
FORMULA:

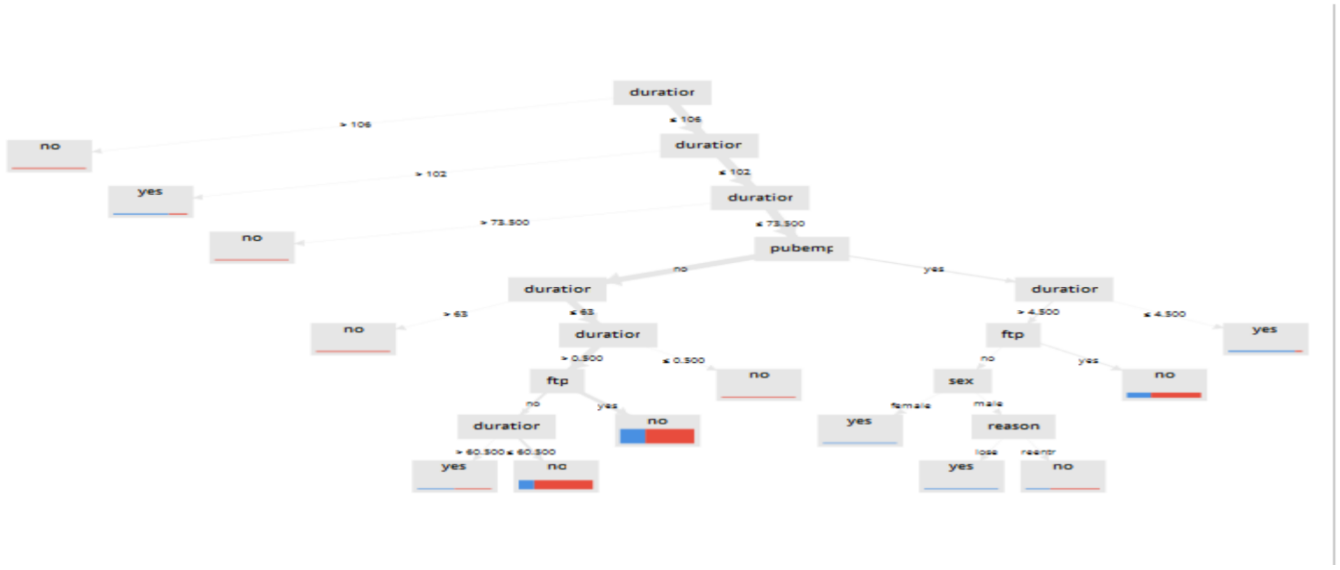
$$H(X) = \mathbb{E}_X[I(x)] = -\sum_{x\in\mathbb{X}} p(x)\log p(x).$$

PSEUDOCODE:

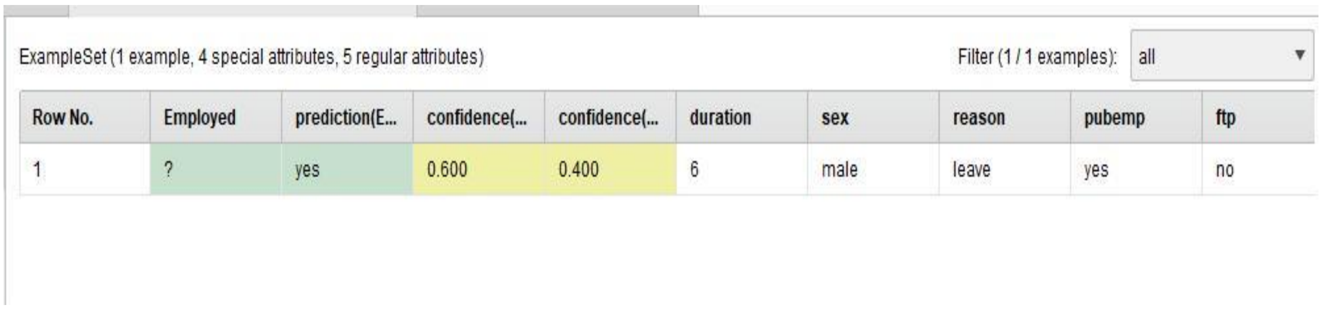
- Place the best attribute of the dataset at the root of the tree.
- Split the training set into subsets. Subsets should be made in such a way that each subset contains data with the same value for an attribute.
- Repeat step 1 and step 2 on each subset until you find leaf nodes in all the branches of the tree.

SCREENSHOTS:





INTERPRETATION:

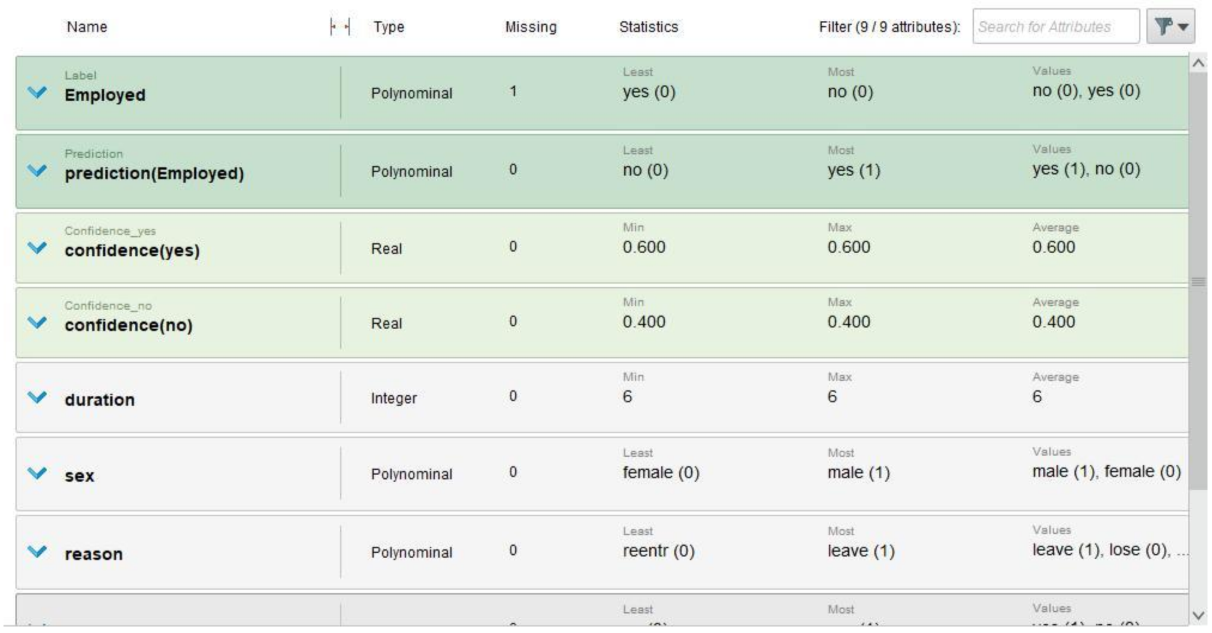


ExampleSet (1 example, 4 special attributes, 5 regular attributes)

Filter (1 / 1 examples): all

Row No.	Employed	prediction(E...	confidence(...	confidence(...	duration	sex	reason	pubemp	ftp
1	?	yes	0.600	0.400	6	male	leave	yes	no

Generally, Decision tree is a pattern in which everyone can easily interpret if they have basic software knowledge in order to make it much clearer I have taken a null data set and predicted the results as mentioned above.



Name	Type	Missing	Statistics		Filter (9 / 9 attributes): Search for Attributes
Label **Employed**	Polynominal	1	Least yes (0)	Most no (0)	Values no (0), yes (0)
Prediction **prediction(Employed)**	Polynominal	0	Least no (0)	Most yes (1)	Values yes (1), no (0)
Confidence_yes **confidence(yes)**	Real	0	Min 0.600	Max 0.600	Average 0.600
Confidence_no **confidence(no)**	Real	0	Min 0.400	Max 0.400	Average 0.400
duration	Integer	0	Min 6	Max 6	Average 6
sex	Polynominal	0	Least female (0)	Most male (1)	Values male (1), female (0)
reason	Polynominal	0	Least reentr (0)	Most leave (1)	Values leave (1), lose (0), ...
			Least	Most	Values

I have taken employed as the class label to predict. The above picture shows the missing value in the data set.

RANDOM FOREST:

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean prediction of the individual trees.

FORMULA:

There is no formula particularly mentioned for this Algorithm.

PSEUDOCODE:

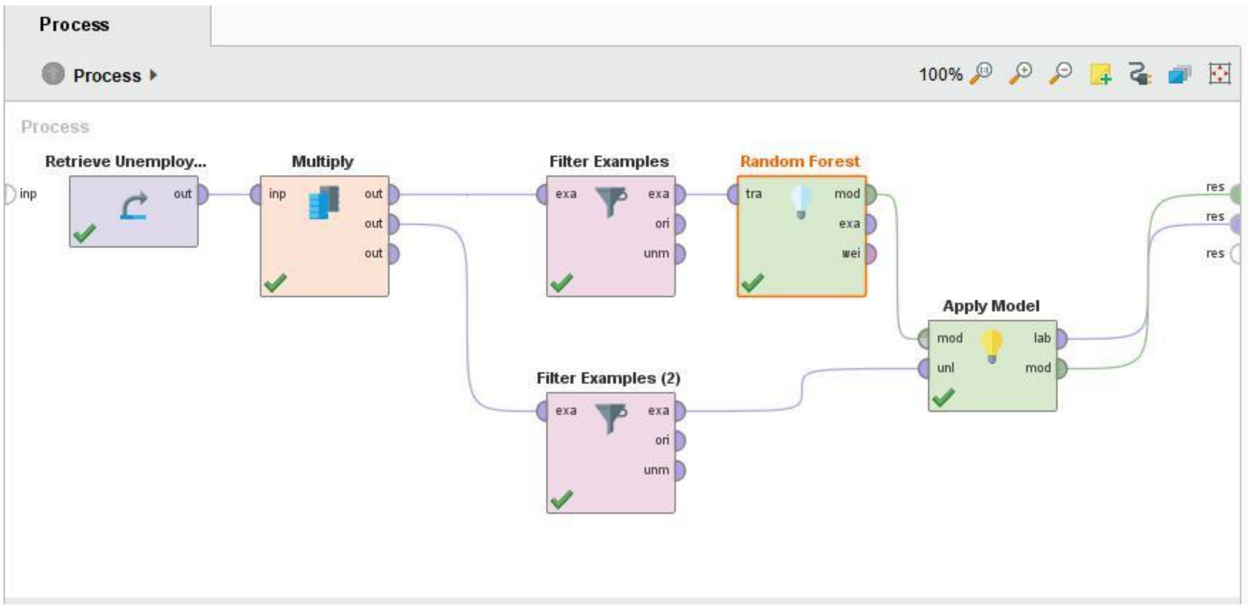
1. Randomly select “k” features from total “m” features.
2. Where k << m
3. Among the “k” features, calculate the node “d” using the best split point.
4. Split the node into daughter nodes using the best split.
5. Repeat 1 to 3 steps until “l” number of nodes has been reached.
6. Build forest by repeating steps 1 to 4 for “n” number times to create “n” number of trees.
7. Takes the test features and use the rules of each randomly created decision tree to predict the outcome and stores the predicted outcome (target)

8. Calculate the votes for each predicted target.

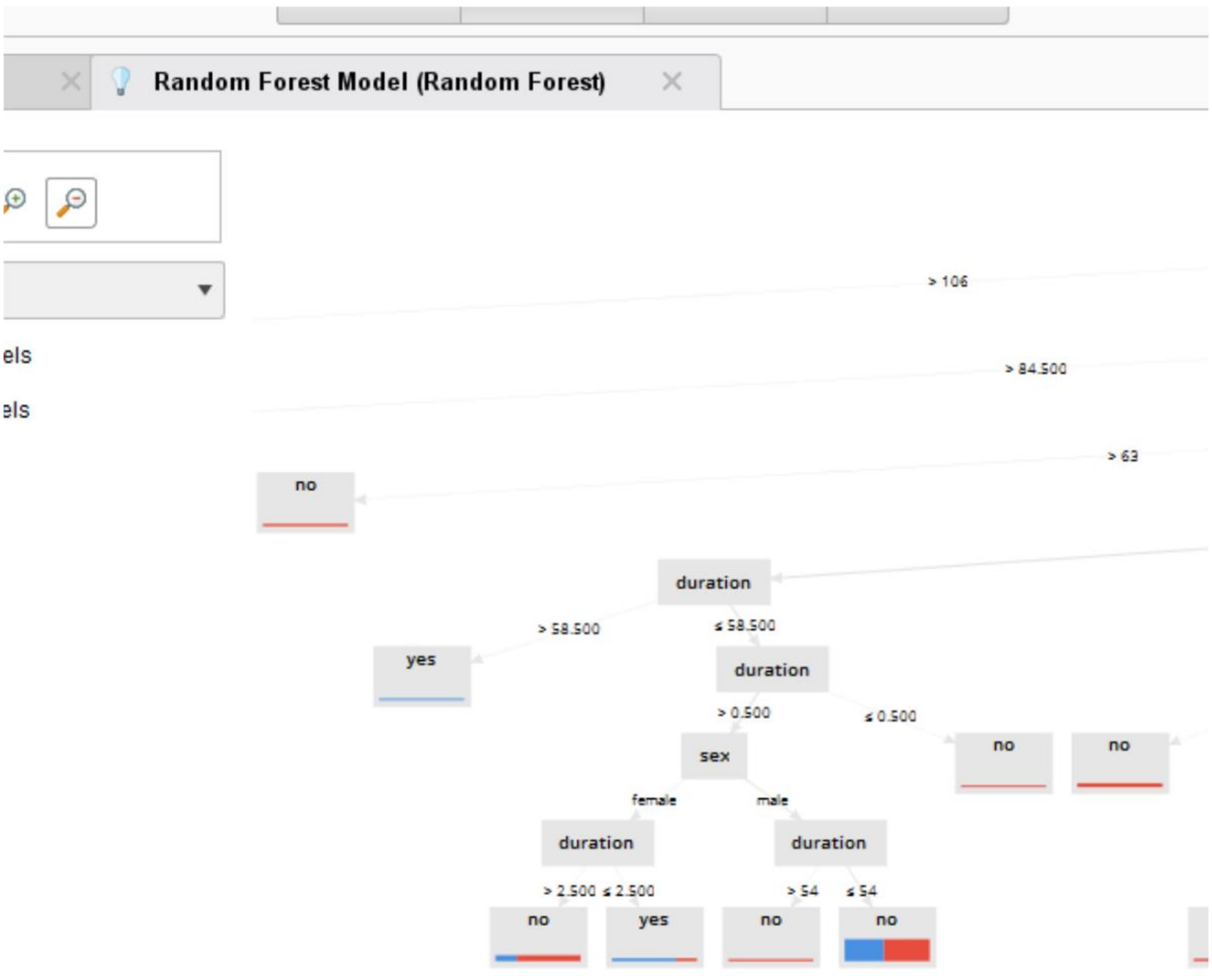
9. Consider the high voted predicted target as the final prediction from the random forest algorithm.

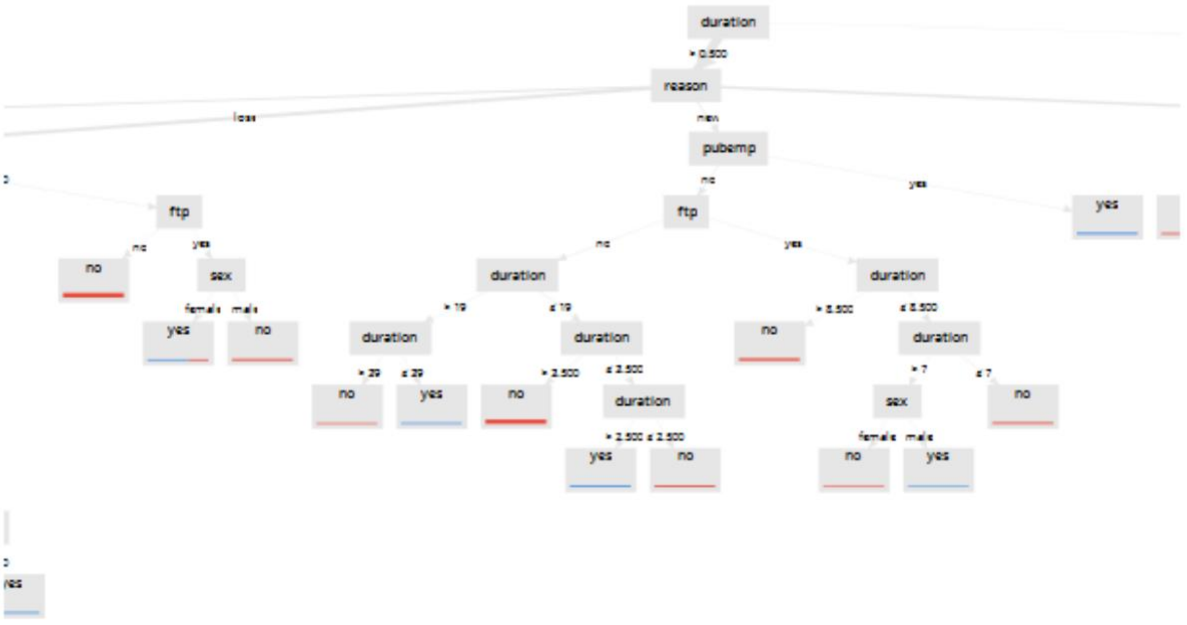
SCREEENSHOTS:

Assuming k value as 100. Attaching 2 to 3 trees for the report purpose.

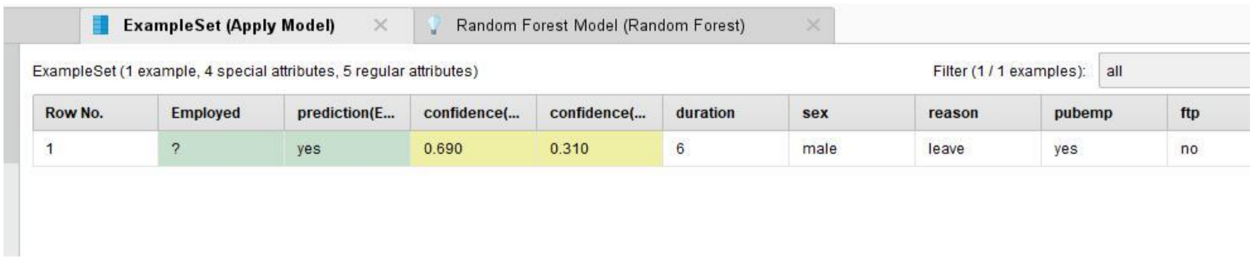


As mentioned in the pseudocode comparing all the 100 trees each gets a vote and the prediction becomes stronger than decision tree.


Random Forest Model (Random Forest)
els
els
> 106
> 84.500
> 63
no
duration
> 58.500
≤ 58.500
yes
duration
> 0.500
≤ 0.500
no
no
sex
female
male
duration
duration
> 2.500 ≤ 2.500
> 54
≤ 54
no
yes
no
no


duration
> 0.500
reason
loss
new
pubemp
no
yes
yes
ftp
no
yes
no
sex
female
male
yes
no
ftp
no
yes
duration
> 19
≤ 19
duration
> 39
≤ 39
no
yes
duration
> 2.500
≤ 2.500
no
duration
> 2.500 ≤ 2.500
yes
no
duration
> 8.500
≤ 8.500
no
duration
> 7
≤ 7
sex
female
male
no
yes
no
yes

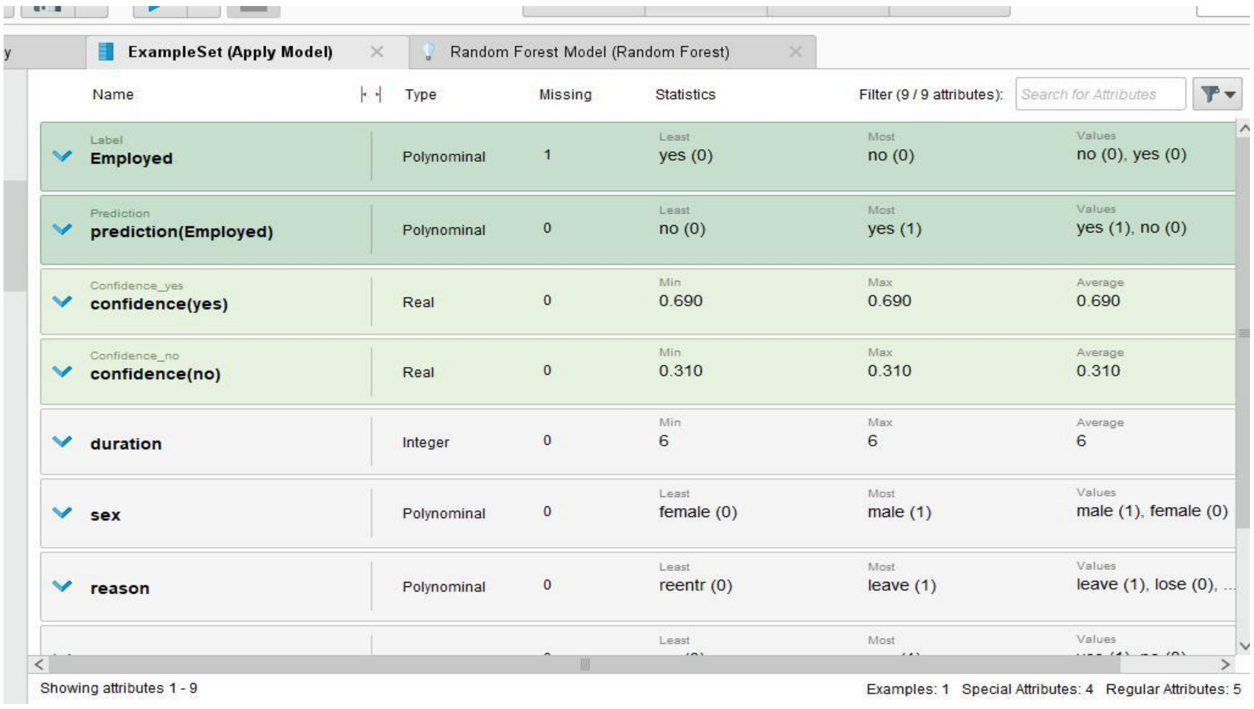
INTERPRETATION:


ExampleSet (Apply Model) | Random Forest Model (Random Forest)

ExampleSet (1 example, 4 special attributes, 5 regular attributes) Filter (1 / 1 examples): all

Row No.	Employed	prediction(E...	confidence(...	confidence(...	duration	sex	reason	pubemp	ftp
1	?	yes	0.690	0.310	6	male	leave	yes	no

Similarly, test data set for the random forest also has been taken to predict.


ExampleSet (Apply Model) | Random Forest Model (Random Forest)

Filter (9 / 9 attributes): Search for Attributes

Name	Type	Missing	Statistics		
Label Employed	Polynominal	1	Least yes (0)	Most no (0)	Values no (0), yes (0)
Prediction prediction(Employed)	Polynominal	0	Least no (0)	Most yes (1)	Values yes (1), no (0)
Confidence_yes confidence(yes)	Real	0	Min 0.690	Max 0.690	Average 0.690
Confidence_no confidence(no)	Real	0	Min 0.310	Max 0.310	Average 0.310
duration	Integer	0	Min 6	Max 6	Average 6
sex	Polynominal	0	Least female (0)	Most male (1)	Values male (1), female (0)
reason	Polynominal	0	Least reentr (0)	Most leave (1)	Values leave (1), lose (0), ...
			Least	Most	Values

Showing attributes 1 - 9 Examples: 1 Special Attributes: 4 Regular Attributes: 5

NAÏVE BAYES:

Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. There is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 10 cm in diameter. A naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of any possible correlations between the colour, roundness, and diameter features. An advantage of naive Bayes is that it only requires a small number of training data to estimate the parameters necessary for classification. It is a classification technique based on Bayes theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as ‘Naive’. Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods. Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). I have chosen naïve Bayes for this dataset because this dataset is basically a predictive one. This classifier works well for prediction. This algorithm gives a better result for the dataset when we compared to other algorithm. It is easy and fast to predict class of test data set. It also performs well in multi class prediction.

FORMULA:

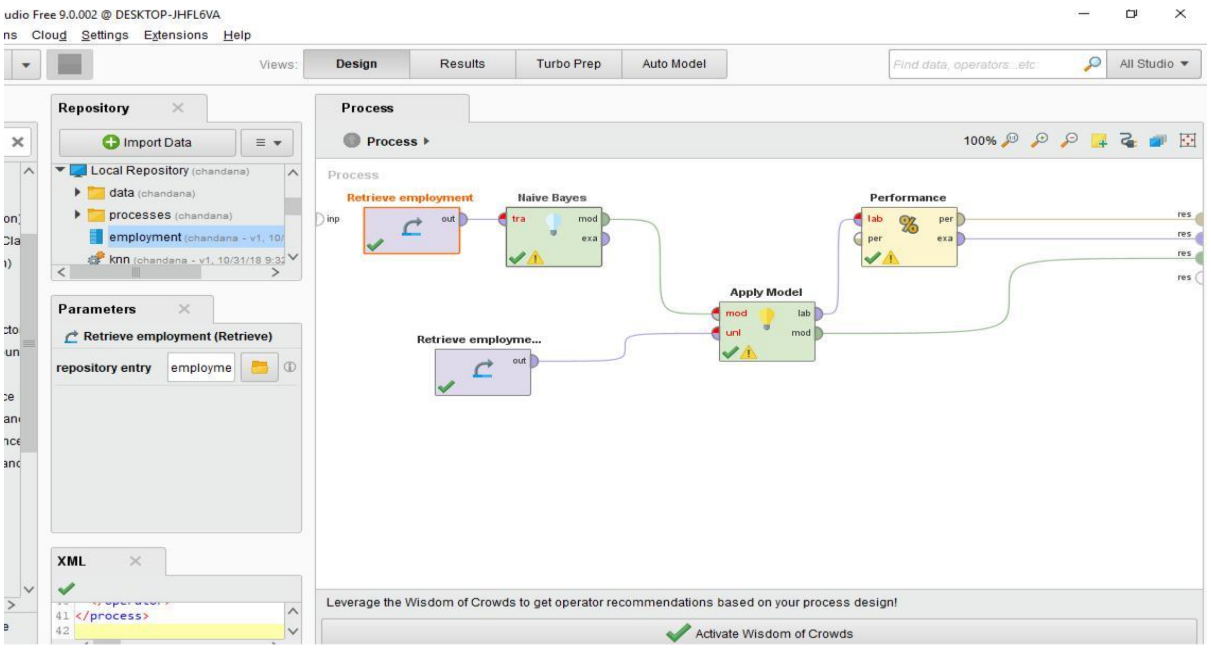
$$\underbrace{P(c \mid x)}_{\text{Posterior Probability}} = \frac{\overbrace{P(x \mid c)}^{\text{Likelihood}}\,\overbrace{P(c)}^{\text{Class Prior Probability}}}{\underbrace{P(x)}_{\text{Predictor Prior Probability}}}$$

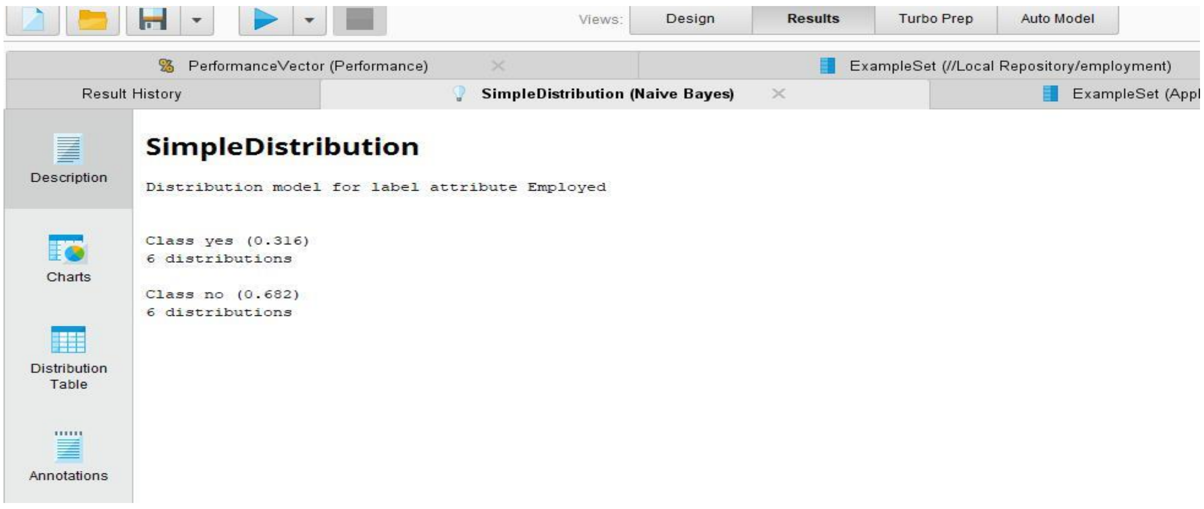
$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

PSEUDOCODE:

```
1.  Read the training dataset T;
2.  Calculate the mean and standard deviation of the
    predictor variables in each class;
3.  Repeat

        Calculate the probability of $f_i$ using the gauss
        density equation in each class;

    Until the probability of all predictor variables ($f_1$, $f_2$,
    $f_3$,.., $f_n$) has been calculated.
4.  Calculate the likelihood for each class;
5.  Get the greatest likelihood;
```

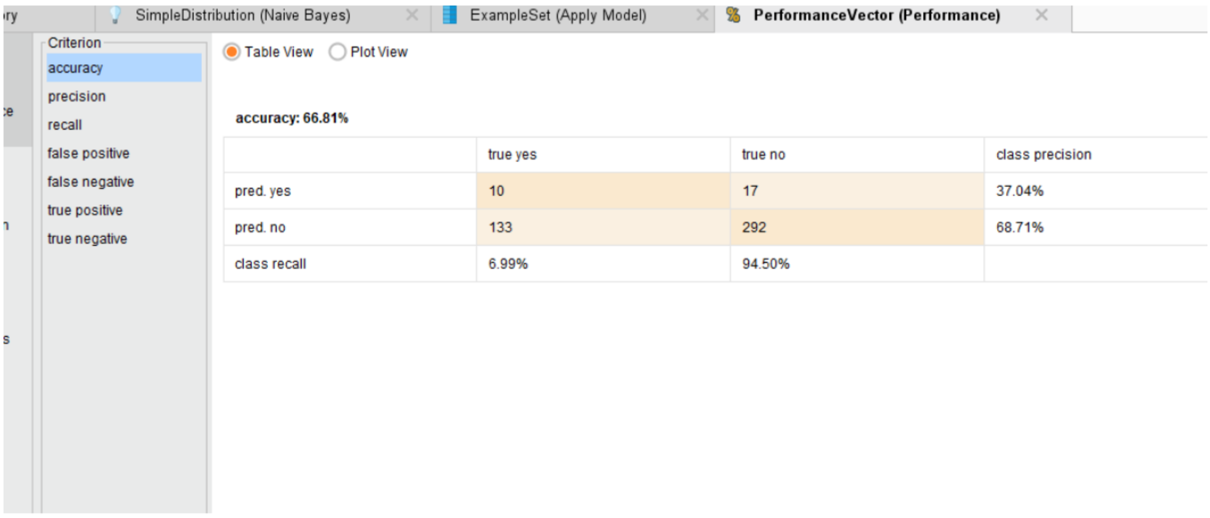
SCREENSHOTS:


udio Free 9.0.002 @ DESKTOP-JHFL6VA
Cloud
Settings
Extensions
Help
Views:
Design
Results
Turbo Prep
Auto Model
Find data, operators...etc
All Studio
Repository
Import Data
Local Repository (chandana)
data (chandana)
processes (chandana)
employment (chandana - v1, 10/
knn (chandana - v1, 10/31/18 9:3
Parameters
Retrieve employment (Retrieve)
repository entry
employme
XML
</process>
Process
Process
100%
Retrieve employment
Naive Bayes
Performance
Apply Model
Retrieve employme...
Leverage the Wisdom of Crowds to get operator recommendations based on your process design!
Activate Wisdom of Crowds



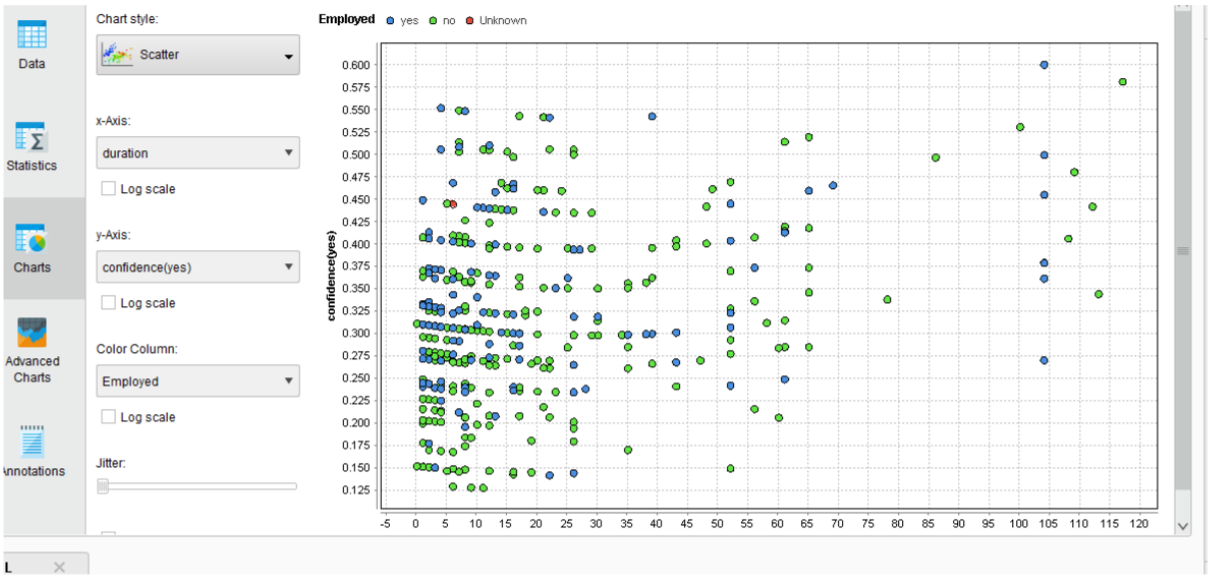
SimpleDistribution

```
Distribution model for label attribute Employed

Class yes (0.316)
6 distributions

Class no (0.682)
6 distributions
```

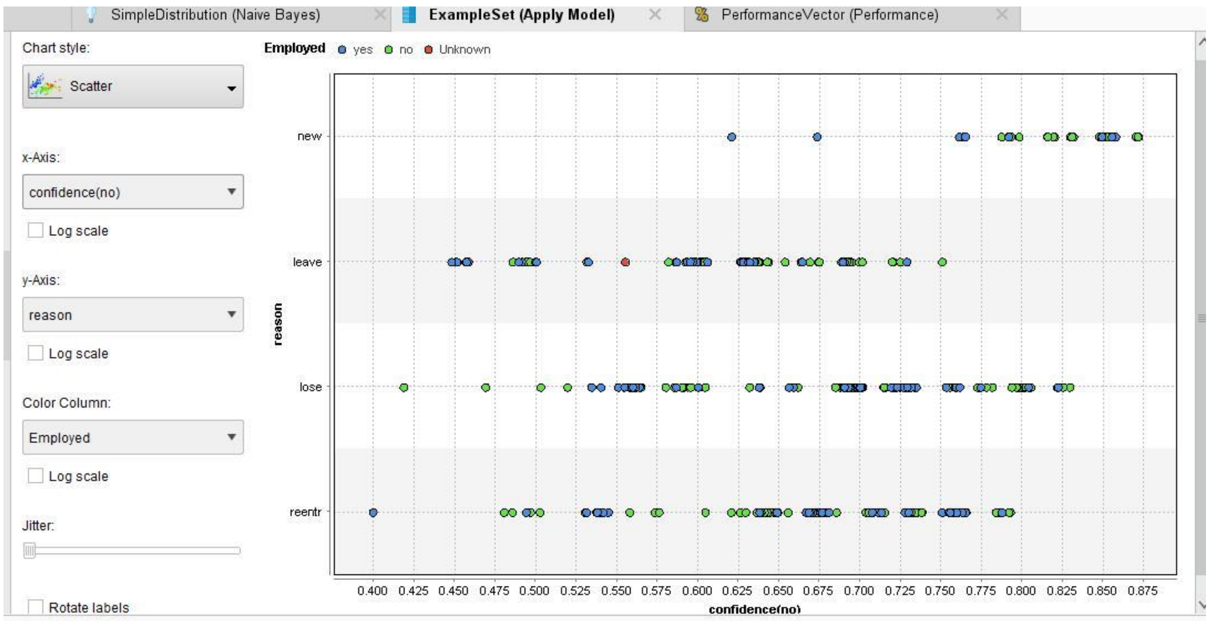


accuracy: 66.81%

	true yes	true no	class precision
pred. yes	10	17	37.04%
pred. no	133	292	68.71%
class recall	6.99%	94.50%	



INTERPRETATION:



I have taken confidence (no) on x axis and reason on y axis. Blue dot indicates yes green indicated no and red indicates unknown. From the graph we can infer that from 0.426-0.451 we have only blue dots i.e. yes for the people who have taken leave and it is the reason for their unemployment. At 0.485 we have green i.e. no that implies leave is not the reason .At 0.555 we have red i.e. unknown that implies the reason for the unemployment is unknown. It is suitable for all the attributes.

KNN:

KNN can be used for both classification and regression predictive problems. However, it is more widely used in classification problems in the industry. To evaluate any technique we generally look at 3 important aspects:

1. Ease to interpret output
2. Calculation time
3. Predictive Power

Another algorithm which I chose is knn (k nearest neighbour) which is used to find the nearest neighbour for the k. It is the one and only algorithm to understand quickly and get the results for the dataset. It has more accuracy when we compared to other algorithm.

PSEUDOCODE:

1. Calculate “d(x, xi)” i =1, 2,...,**n**; where **d** denotes the Euclidean distance between the points.

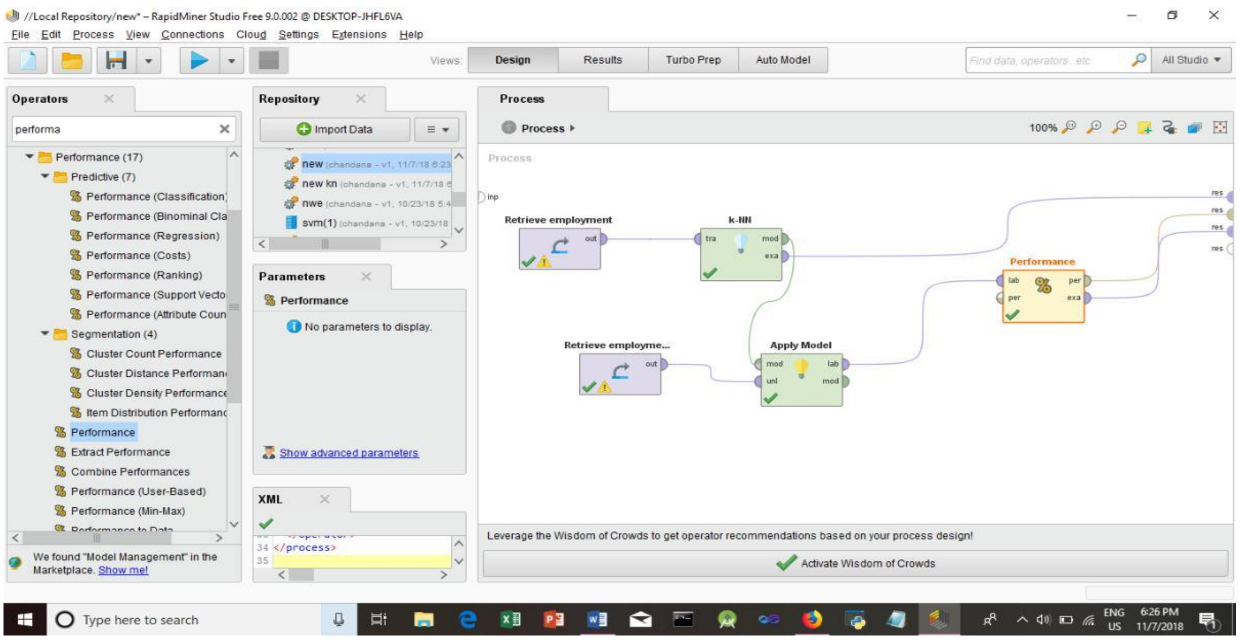
2. Arrange the calculated **n** Euclidean distances in non-decreasing order.

3. Let **k** be a +ve integer, take the first **k** distances from this sorted list.

4. Find those **k**-points corresponding to these **k**-distances.

5. Let **ki** denotes the number of points belonging to the ith class among **k** points i.e. $k \geq 0$

SCREENSHOTS:



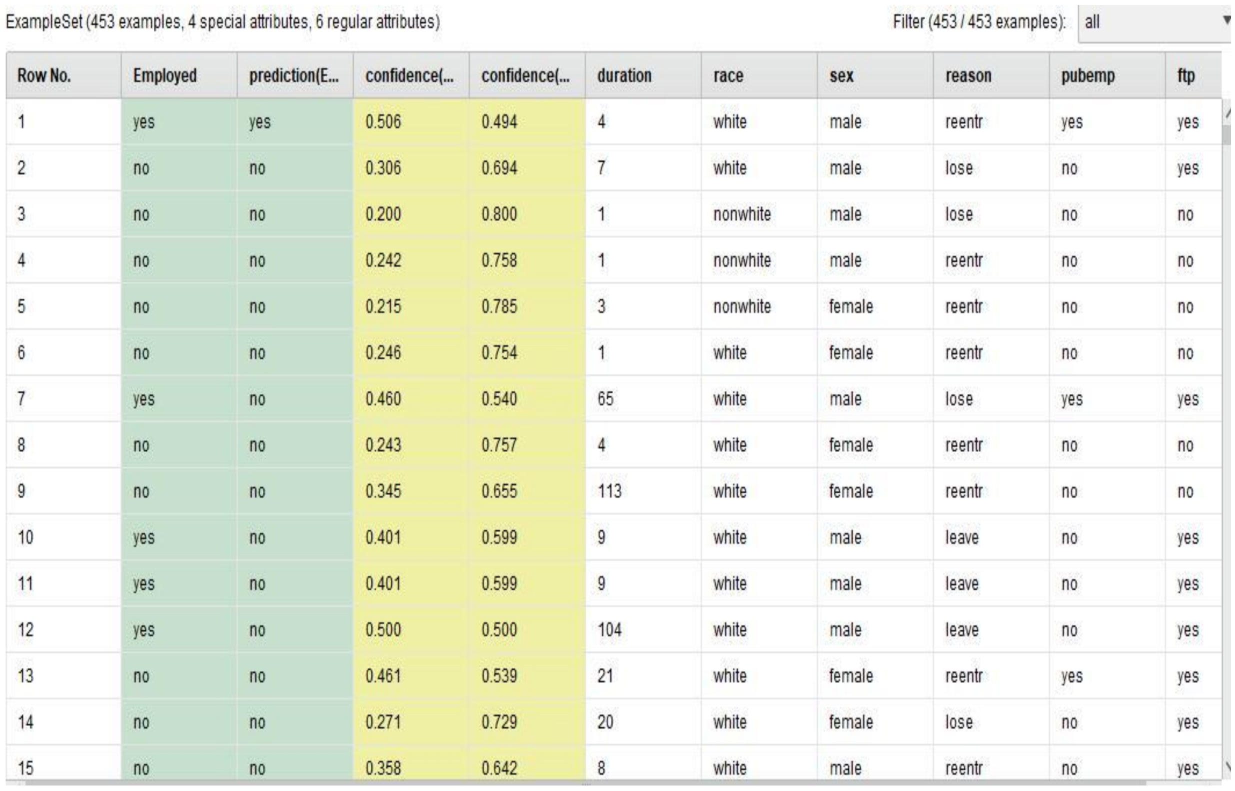
//Local Repository/new* – RapidMiner Studio Free 9.0.002 @ DESKTOP-JHFL6VA
File Edit Process View Connections Cloud Settings Extensions Help
Views: Design Results Turbo Prep Auto Model
Operators
performa
Performance (17)
Predictive (7)
Performance (Classification)
Performance (Binominal Cla
Performance (Regression)
Performance (Costs)
Performance (Ranking)
Performance (Support Vecto
Performance (Attribute Coun
Segmentation (4)
Cluster Count Performance
Cluster Distance Performan
Cluster Density Performance
Item Distribution Performanc
Performance
Extract Performance
Combine Performances
Performance (User-Based)
Performance (Min-Max)
Repository
Import Data
new (chandana - v1, 11/7/18 6:23
new kn (chandana - v1, 11/7/18
nwe (chandana - v1, 10/23/18
svm(1) (chandana - v1, 10/23/18
Parameters
Performance
No parameters to display.
Show advanced parameters
XML
Process
Retrieve employment
k-NN
Retrieve employme...
Apply Model
Performance

We found "Model Management" in the Marketplace. Show me!

```
34 </process>
```

Leverage the Wisdom of Crowds to get operator recommendations based on your process design!

Activate Wisdom of Crowds

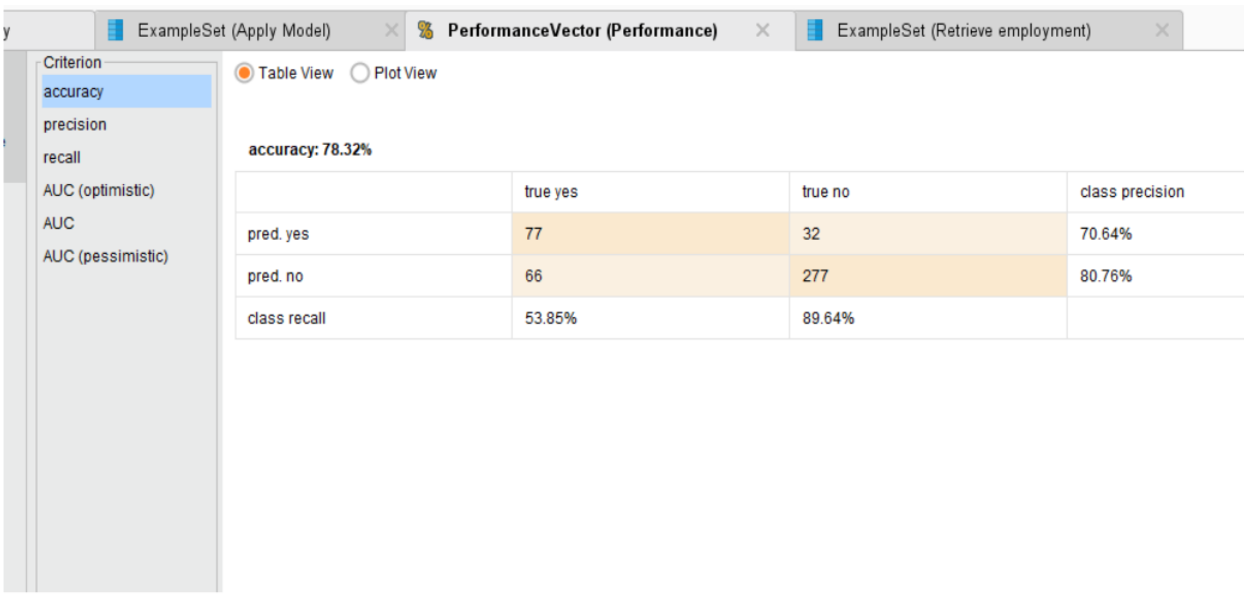


ExampleSet (453 examples, 4 special attributes, 6 regular attributes)

Filter (453 / 453 examples): all

Row No.	Employed	prediction(E...	confidence(...	confidence(...	duration	race	sex	reason	pubemp	ftp
1	yes	yes	0.506	0.494	4	white	male	reentr	yes	yes
2	no	no	0.306	0.694	7	white	male	lose	no	yes
3	no	no	0.200	0.800	1	nonwhite	male	lose	no	no
4	no	no	0.242	0.758	1	nonwhite	male	reentr	no	no
5	no	no	0.215	0.785	3	nonwhite	female	reentr	no	no
6	no	no	0.246	0.754	1	white	female	reentr	no	no
7	yes	no	0.460	0.540	65	white	male	lose	yes	yes
8	no	no	0.243	0.757	4	white	female	reentr	no	no
9	no	no	0.345	0.655	113	white	female	reentr	no	no
10	yes	no	0.401	0.599	9	white	male	leave	no	yes
11	yes	no	0.401	0.599	9	white	male	leave	no	yes
12	yes	no	0.500	0.500	104	white	male	leave	no	yes
13	no	no	0.461	0.539	21	white	female	reentr	yes	yes
14	no	no	0.271	0.729	20	white	female	lose	no	yes
15	no	no	0.358	0.642	8	white	male	reentr	no	yes

INTERPRETATION:



y | ExampleSet (Apply Model) | PerformanceVector (Performance) | ExampleSet (Retrieve employment)

Criterion: accuracy, precision, recall, AUC (optimistic), AUC, AUC (pessimistic)

[✓ Table View] Plot View

accuracy: 78.32%

	true yes	true no	class precision
pred. yes	77	32	70.64%
pred. no	66	277	80.76%
class recall	53.85%	89.64%	

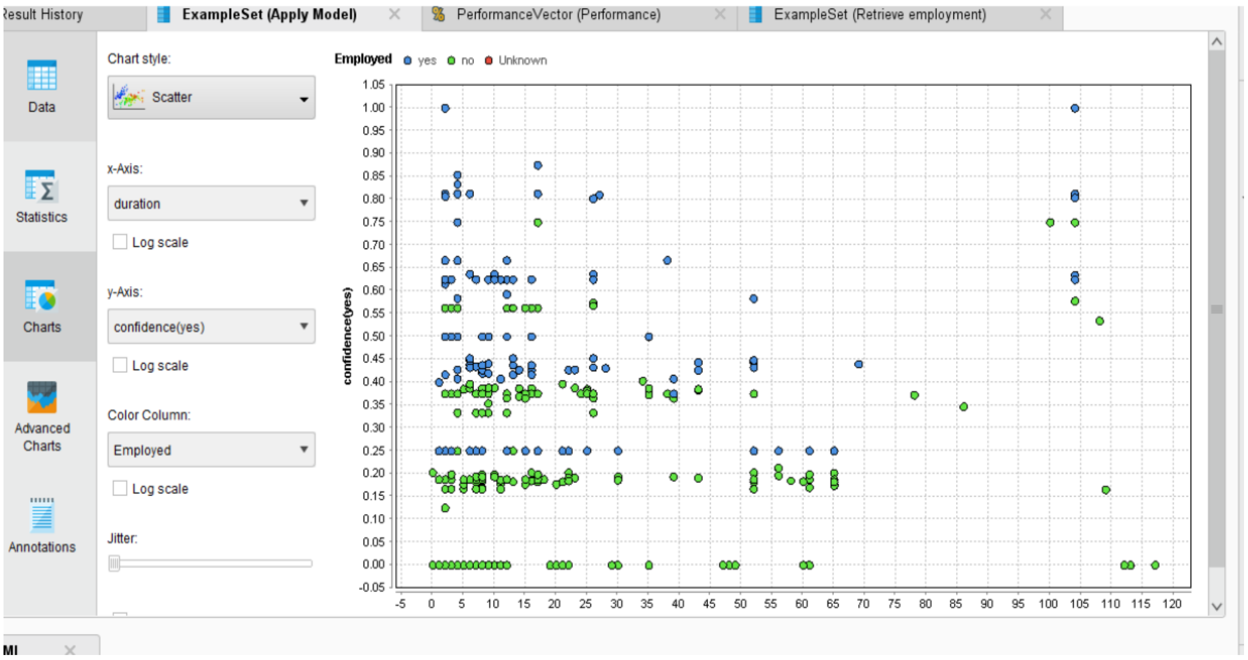


Chart style: Scatter

x-Axis: duration

y-Axis: confidence(yes)

Color Column: Employed

We have to declare the value of k. We have to find the distance between the example instances with the original dataset. Then rank the distance if the distance value is less than or equal to the k value then that attributes only be taken. Then we have to count the each instance in the class label. If the count of the instance is greater than another instance that instance will be taken as the class label for the example instance. In original dataset we have 309 no and 143 yes after applying he knn algorithm we got the increase in the value of no 344 and 104 yes So I have taken no as the instance for the employed class label. As k value increases accuracy also changes. So that I can infer that the instance count is high then that instance will be taken as the attribute for that particular instance.

K – MEANS:

K-Means clustering intends to partition *n* objects into *k* clusters in which each object belongs to the cluster with the nearest mean. This method produces exactly *k* different clusters of greatest possible distinction. The best number of clusters *k* leading to the greatest separation (distance) is not known as a priori and must be computed from the data. The objective of K-Means clustering is to minimize total intra-cluster variance, or, the squared error function.

FORMULA:

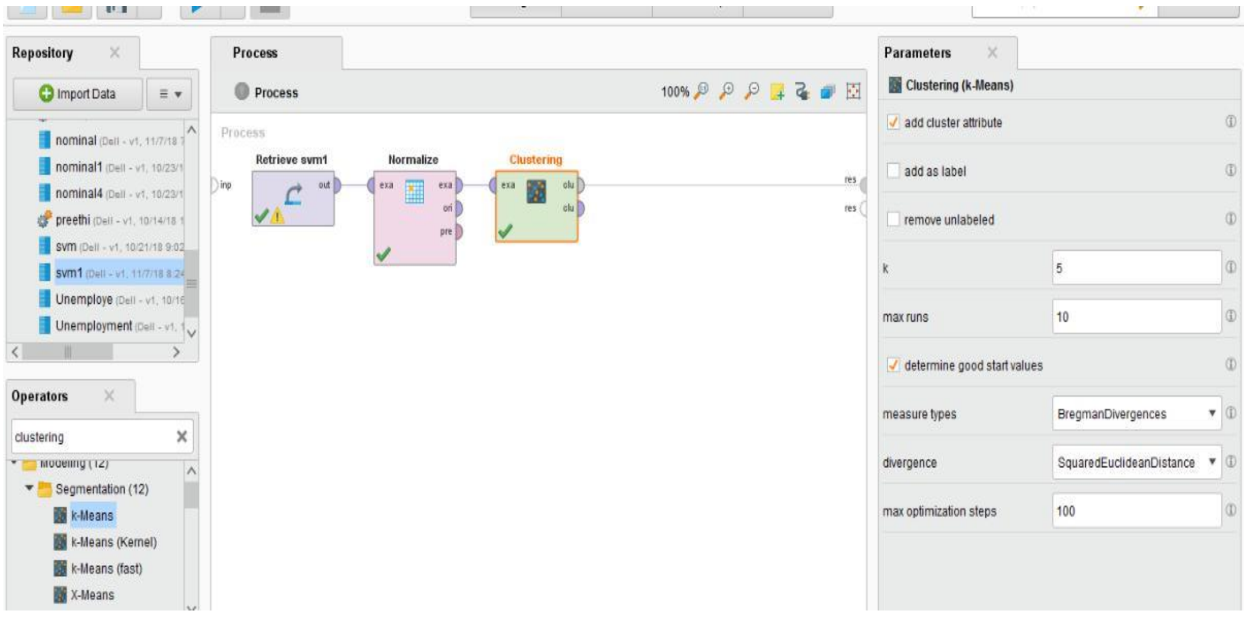
Euclidean distance :

Root of (x2-x1)2 + (y2-y1)2.

PSEUDOCODE:

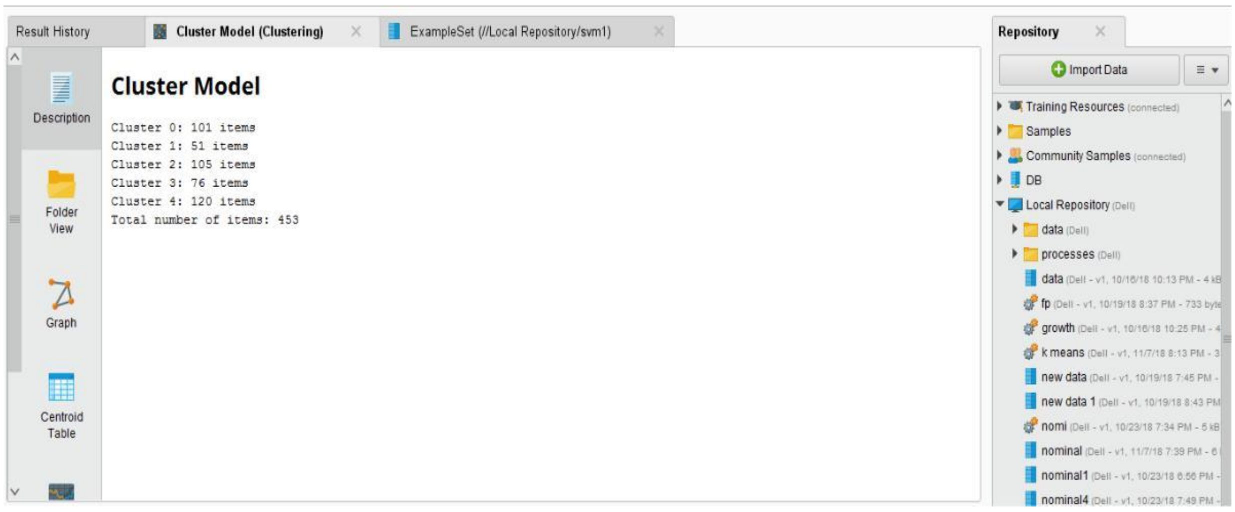
```
1. Clusters the data into k groups where k is predefined.
2. Select k points at random as cluster centres.
3. Assign objects to their closest cluster centre according to the Euclidean distance function.
4. Calculate the centroid or mean of all objects in each cluster.
5. Repeat steps 2, 3 and 4 until the same points are assigned to each cluster in consecutive
   rounds.
```

SCREENSHOTS:



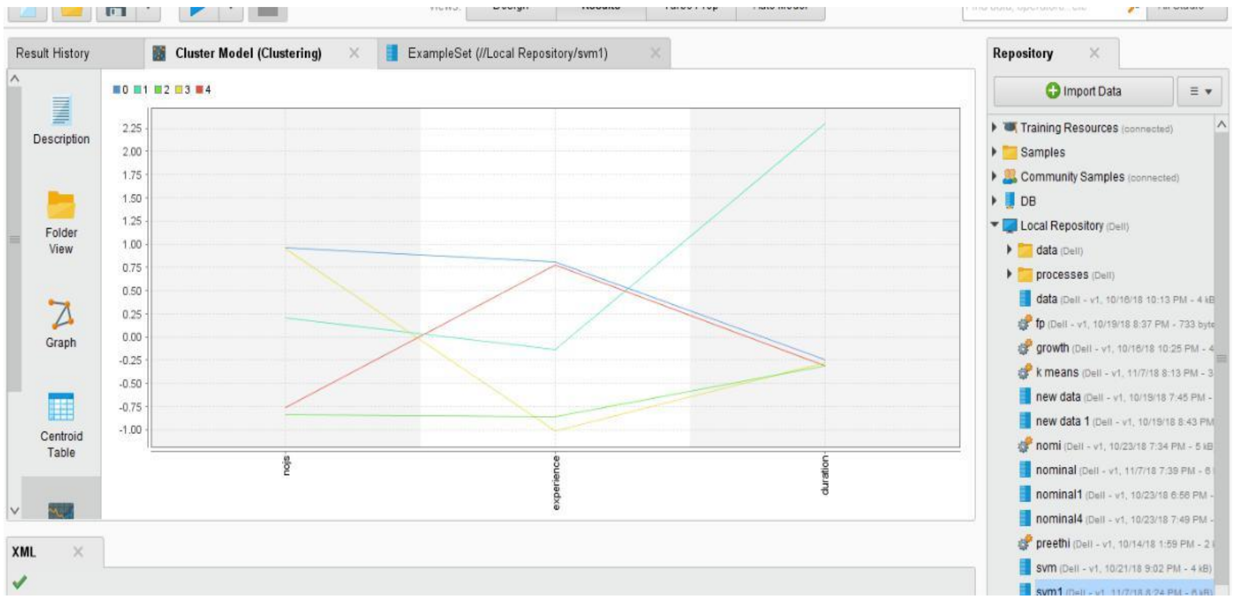
Clustering (k-Means)

[✓] add cluster attribute	
[ ] add as label	
[ ] remove unlabeled	
k	5
max runs	10
[✓] determine good start values	
measure types	BregmanDivergences
divergence	SquaredEuclideanDistance
max optimization steps	100



Cluster Model

```
Cluster 0: 101 items
Cluster 1: 51 items
Cluster 2: 105 items
Cluster 3: 76 items
Cluster 4: 120 items
Total number of items: 453
```



INTERPRETATION:

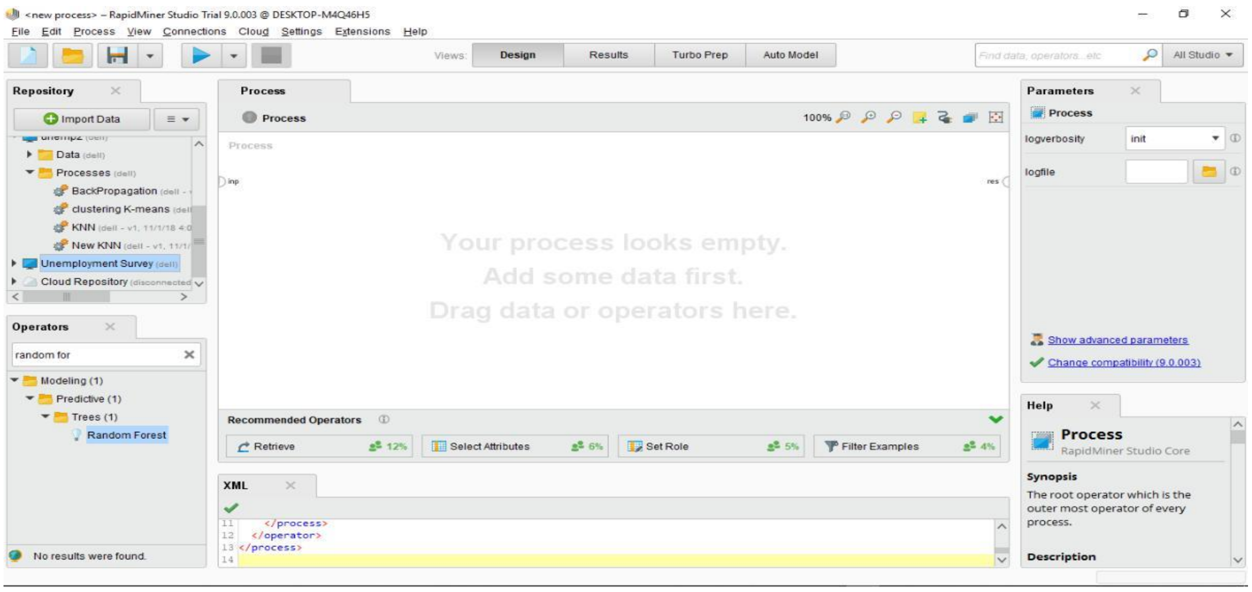
From the algorithm k means clustering k value is 5 (0,1,2,3,4) . Data sets have nojs (number of job searched), experience (experience of each candidates) and duration (working period) .Each cluster has different items like cluster 0 = 101 items, cluster 1 = 51 items , etc… In the graph each cluster represents different colours. For the cluster 0 (blue colour) the nojs is high, experience is moderate and duration is low. For the cluster 3 (yellow colour) the nojs is high, experience is very low when compared to all clusters and duration is moderate. Similarly the graphs represent the other clusters.

TECHNOLOGY:

Rapid Miner:

The tool I used is Rapid miner. It is licenced software. They provide trial version to experience the tool. The tools performance is as good as expected. Rapid Miner is a data science software platform developed by the company of the same name that provides an integrated environment for data preparation, machine learning, deep learning, text mining, and predictive analytics.

TOOL INTERFACE:



Your process looks empty.
Add some data first.
Drag data or operators here.

```
    </process>
  </operator>
</process>
```

The root operator which is the outer most operator of every process.

CONCLUSION AND FUTURE ENHANCEMENTS:

The project will definitely help students to improve their skills if they are not fit to get the job. They can survey themselves and train them even better to get a job. I will be trying to integrate with android studio as well as rapid miner with visual studio (ADO.NET OR ASP.NET) to create a user interface to make it work and understand easily. The future enhancement will generally be the user interface for the user and stake holders.

SAMPLE CODE

Attaching the XML Code of all algorithms

Decision Tree:

```
<?xml version="1.0" encoding="UTF-8"?><process
version="9.0.003">

  <context>

    <input/>

    <output/>

    <macros/>

  </context>

  <operator activated="true" class="process" compatibility="9.0.003"
expanded="true" name="Process">

    <process expanded="true">

      <operator activated="true" class="retrieve" compatibility="9.0.003"
expanded="true" height="68" name="Retrieve Unemployment" width="90"
x="45" y="34">

        <parameter key="repository_entry"
value="../Data/Unemployment"/>

      </operator>

      <operator activated="true" class="multiply" compatibility="9.0.003"
expanded="true" height="103" name="Multiply" width="90" x="179"
y="34"/>

       <operator activated="true" class="filter_examples"
compatibility="9.0.003" expanded="true" height="103" name="Filter
Examples (2)" width="90" x="380" y="187">

        <list key="filters_list">

          <parameter key="filters_entry_key">
```

```
value="Employed.is_missing."/>
      </list>
    </operator>
    <operator          activated="true"          class="filter_examples"
compatibility="9.0.003"   expanded="true"   height="103"   name="Filter
Examples" width="90" x="380" y="34">
      <list key="filters_list">
        <parameter key="filters_entry_key"
value="Employed.is_not_missing."/> </list>
    </operator>
    <operator activated="true"
class="concurrency:parallel_decision_tree"
compatibility="9.0.003" expanded="true" height="103"
name="Decision Tree" width="90" x="514" y="34"/>
    <operator activated="true" class="apply_model"
compatibility="9.0.003" expanded="true" height="82" name="Apply
Model" width="90" x="648" y="136">
      <list key="application_parameters"/>
    </operator>
    <connect from_op="Retrieve Unemployment"
from_port="output" to_op="Multiply" to_port="input"/>
        <connect from_op="Multiply" from_port="output 1" to_op="Filter
  Examples" to_port="example set input"/> <connect from_op="Multiply"
                                                      from_port="output 2"
to_op="Filter Examples (2)" to_port="example set input"/> <connect
    from_op="Filter Examples (2)"
from_port="example set output" to_op="Apply Model"
```

to_port="unlabelled data"/>

<connect from_op="Filter Examples" from_port="example set output" to_op="Decision Tree" to_port="training set"/>

<connect from_op="Decision Tree" from_port="model" to_op="Apply Model" to_port="model"/>

<connect from_op="Apply Model" from_port="labelled data" to_port="result 2"/>

<connect from_op="Apply Model" from_port="model" to_port="result 1"/>

<portSpacing port="source_input 1" spacing="0"/> <portSpacing port="sink_result 1" spacing="0"/> <portSpacing port="sink_result 2" spacing="0"/> <portSpacing port="sink_result 3" spacing="0"/>

</process>

</operator>

</process>

Random Forest:

<?xml version="1.0" encoding="UTF-8"?><process version="9.0.003">

<context>

<input/>

<output/>

<macros/>

</context>

<operator activated="true" class="process" compatibility="9.0.003" expanded="true" name="Process">

<process expanded="true">

```
        <operator activated="true" class="retrieve" compatibility="9.0.003"
    expanded="true" height="68" name="Retrieve Unemployment" width="90"
    x="45" y="34">

       <parameter key="repository_entry" value="../Data/Unemployment"/>

       </operator>

    <operator activated="true" class="multiply" compatibility="9.0.003"
    expanded="true" height="103" name="Multiply" width="90" x="179"
    y="34"/>

        <operator activated="true" class="filter_examples"
    compatibility="9.0.003" expanded="true" height="103" name="Filter
    Examples (2)" width="90" x="380" y="187">


<list key="filters_list">

            <parameter key="filters_entry_key"

    value="Employed.is_missing."/>

          </list>

        </operator>

        <operator          activated="true"          class="filter_examples"
    compatibility="9.0.003"   expanded="true"   height="103"   name="Filter
    Examples" width="90" x="380" y="34">

          <list key="filters_list">

            <parameter key="filters_entry_key" value="Employed.is_not_missing."/>
            </list>

        </operator>

        <operator activated="true" class="concurrency:parallel_random_forest"
        compatibility="9.0.003" expanded="true" height="103" name="Random Forest"
        width="90" x="514" y="34"/>

      <operator activated="true" class="apply_model" compatibility="9.0.003"
    expanded=  "true" height="82" name="Apply Model" width="90"

    x="648" y="136">
```

```
          <list key="application_parameters"/>
        </operator>

        <connect from_op="Retrieve Unemployment"
from_port="output" to_op="Multiply" to_port="input"/>

        <connect from_op="Multiply" from_port="output 1" to_op="Filter
Examples" to_port="example set input"/>

        <connect from_op="Multiply" from_port="output 2" to_op="Filter
Examples (2)" to_port="example set input"/>

        <connect from_op="Filter Examples (2)"
from_port="example set output" to_op="Apply Model"
to_port="unlabelled data"/>

        <connect from_op="Filter Examples" from_port="example set output"
to_op="Random Forest" to_port="training set"/>

        <connect from_op="Random Forest" from_port="model" to_op="Apply
Model" to_port="model"/>

        <connect from_op="Apply Model" from_port="labelled data"
to_port="result 2"/>

        <connect from_op="Apply Model" from_port="model" to_port="result
1"/>

        <portSpacing port="source_input 1" spacing="0"/> <portSpacing
        port="sink_result 1" spacing="0"/> <portSpacing port="sink_result
        2" spacing="0"/> <portSpacing port="sink_result 3" spacing="0"/>

      </process>
    </operator>
</process>
```

Naïve Bayes

```
<?xml version="1.0" encoding="UTF-8"?><process
version="9.0.003">

  <context>

    <input/>

    <output/>

    <macros/>

  </context>

  <operator activated="true" class="process" compatibility="9.0.003"
expanded="true" name="Process">

    <process expanded="true">

      <operator activated="true" class="retrieve" compatibility="9.0.003"
expanded="true" height="68" name="Retrieve Unemployment" width="90" x="112"
y="34">

        <parameter key="repository_entry" value="//Unemployment
Survey/Data/Unemployment"/>

      </operator>

      <operator activated="true" class="select_attributes"
compatibility="9.0.003" expanded="true" height="82" name="Select
Attributes" width="90" x="246" y="34">

        <parameter key="attribute_filter_type" value="subset"/> <parameter
        key="attributes" value="reason|employed"/>

      </operator>

      <operator activated="true" class="naive_bayes" compatibility="9.0.003"
      expanded="true" height="82" name="Naive Bayes" width="90" x="380"
      y="34"/>

      <operator activated="true" class="retrieve" compatibility="9.0.003"
expanded="true" height="68" name="Retrieve Unemployment (2)"
width="90" x="380" y="136">
```

```
            <parameter key="repository_entry" value="//Unemployment
        Survey/Data/Unemployment"/>

           </operator>
           <operator activated="true" class="apply_model"
        compatibility="9.0.003" expanded="true" height="82" name="Apply
        Model" width="90" x="514" y="34">

            <list key="application_parameters"/>
           </operator>

           <connect from_op="Retrieve Unemployment"
        from_port="output" to_op="Select Attributes" to_port="example
        set input"/>

           <connect from_op="Select Attributes" from_port="example set output"
        to_op="Naive Bayes" to_port="training set"/>

<connect from_op="Naive Bayes" from_port="model" to_op="Apply Model"
to_port="model"/>

           <connect from_op="Retrieve Unemployment (2)"

        from_port="output" to_op="Apply Model" to_port="unlabelled

        data"/>
           <connect from_op="Apply Model" from_port="labelled data"
        to_port="result 2"/>

           <connect from_op="Apply Model" from_port="model" to_port="result
        1"/>

           <portSpacing port="source_input 1" spacing="0"/> <portSpacing
           port="sink_result 1" spacing="0"/> <portSpacing port="sink_result
           2" spacing="0"/> <portSpacing port="sink_result 3" spacing="0"/>

          </process>
```

```
  </operator>
</process>
```

KNN

```
 <?xml version="1.0" encoding="UTF-8"?><process version="9.0.003">
  <context>
    <input/>
    <output/>
    <macros/>
  </context>
  <operator activated="true" class="process" compatibility="9.0.003"
  expanded="true" name="Process">
    <process expanded="true">
      <operator activated="true" class="retrieve"compatibility="9.0.003"
      expanded="true" height="68" name="Retrieve svm" width="90" x="45"
      y="34">
        <parameter key="repository_entry" value="//unemp2/Data/svm"/>
      </operator>
      <operator activated="true" class="k_nn" compatibility="9.0.003"
      expanded="true" height="82" name="k- NN" width="90" x="246" y="34"/>
      <operator activated="true" class="retrieve"compatibility="9.0.003"
      expanded="true" height="68" name="Retrieve svm (2)" width="90" x="45"
      y="136">

        <parameter key="repository_entry" value="//unemp2/Data/svm"/>
      </operator>
      <operator activated="true" class="apply_model" compatibility="9.0.003"
      expanded="true" height="82" name="Apply Model" width="90" x="380"
      y="136">
```

```
          <list key="application_parameters"/>
        </operator>

        <operator activated="true" class="performance" compatibility="9.0.003"
        expanded="true" height="82" name="Performance" width="90" x="514"
        y="136"/>

        <connect from_op="Retrieve svm" from_port="output" to_op="k-NN"
to_port="training set"/>


        <connect from_op="k-NN" from_port="model" to_op="Apply

Model" to_port="model"/>

     <connect from_op="k-NN" from_port="exampleSet" to_port="result 1"/>

       <connect from_op="Retrieve svm (2)" from_port="output" to_op="Apply
Model" to_port="unlabelled data"/>

       <connect from_op="Apply Model" from_port="labelled data"
to_op="Performance" to_port="labelled data"/>

       <connect from_op="Performance" from_port="performance" to_port="result
2"/>

       <connect from_op="Performance" from_port="example set" to_port="result 3"/>

       <portSpacing port="source_input 1" spacing="0"/>
       <portSpacing port="sink_result 1" spacing="0"/>
       <portSpacing port="sink_result 2" spacing="0"/>
       <portSpacing port="sink_result 3" spacing="0"/>
       <portSpacing port="sink_result 4" spacing="0"/>

     </process>


   </operator>

</process>
```

K means:

```
<?xml version="1.0" encoding="UTF-8"?><process
version="9.0.003">

    <context>

      <input/>

   <output/>

   <macros/>

 </context>

 <operator activated="true" class="process" compatibility="9.0.003"
expanded="true" name="Process">

   <process expanded="true">

    <operator activated="true" class="retrieve"
compatibility="9.0.003" expanded="true" height="68" name="Retrieve
svm" width="90" x="45" y="34">

      <parameter key="repository_entry" value="//unemp2/Data/svm"/>

    </operator>

    <operator activated="true" class="normalize"
compatibility="9.0.003" expanded="true" height="103"
name="Normalize" width="90" x="179" y="34"/>


    <operator activated="true" class="concurrency:k_means"
compatibility="9.0.003" expanded="true" height="82" name="Clustering"
width="90" x="313" y="34"/>

    <connect from_op="Retrieve svm" from_port="output"
to_op="Normalize" to_port="example set input"/>


    <connect from_op="Normalize" from_port="example set output"
to_op="Clustering" to_port="example set"/>
```

```
        <connect from_op="Clustering" from_port="cluster model" to_port="result
1"/>

                    <portSpacing port="source_input 1" spacing="0"/>

        <portSpacing port="sink_result 1" spacing="0"/> <portSpacing
        port="sink_result 2" spacing="0"/>


      </process>

   </operator>

</process>
```