Qom University

Faculty of Engineering and Technology

Department of Computer Engineering and Information Technology

Project Report in Data Mining

Title:

Investigating the Body Mass Index Criterion on Blood Pressure

Student:

Ramin Badri

Professor:

Dr. Amirkhani

Contents

Abstract

Today, collecting extensive data about various diseases holds great importance in medical science. Medical centers gather this data for numerous purposes. Analyzing this data to derive useful insights and patterns regarding diseases is one of the key objectives of its collection. However, the vast amount of data and the resulting confusion present challenges that hinder the achievement of significant results.

Therefore, data mining is utilized to address this problem and uncover valuable relationships among risk factors in diseases.

In this study, we use the existing data and the decision tree classification technique, Bayes' theory, and the algorithm to predict the occurrence of high blood pressure in people with changing body mass index.

Using the software and with the help of clustering techniques and the most appropriate algorithm, we place our data in several clusters.

Keywords: K-medoids algorithm, K-means algorithm, KNN, Data mining, Decision tree, Bayes theory

Section 1: Introduction

High blood pressure is one of the risk factors for cardiovascular diseases. On the other hand, death from cardiovascular diseases accounts for the highest mortality rate in most industrialized countries, and its rate is also increasing in 17 (WHO) developing countries. According to the World Health Organization (WHO) classification, there is a main cause of mortality, of which cardiovascular diseases occupy the seventh place in the world, but in Iran, death from cardiovascular diseases ranks first.

In many cases, the main cause of hypertension is unknown. However, several factors have been indicated to aggravate this condition. Body Mass Index (BMI) is an anthropometric measure of weight divided by the square of height (kg /m^2) and is often widely used to assess overweight and obesity in large populations.

Section 2: General Research Objectives

The diagnosis of the disease is the responsibility of the doctor, who observes the symptoms and performs tests. Among the diseases that doctors face difficulties in diagnosing are blood pressure and heart diseases, which are of great importance due to the vital role of the heart in human health. For this reason, there is a need for systems to help make decisions. There are decision support systems with a different approach to this type of decision-making, and the basis of most of these approaches is the use of techniques that can discover the relationship between data. Three of these techniques are categorized. KNN decision tree classification, Bayes theory, and algorithm. In this report, we will first explain the database used and provide a brief explanation of the KNN data characteristics. Then we will refer to the decision tree classification technique, the Bayes algorithm and algorithm, and then we will implement these techniques on the data, and then we will cluster the data using the

RapidMiner software. Finally, the results obtained from the implementation of K-Medoids and K-Means using the algorithm can be seen.

Section 3: Data Description

3.1. Dataset

To apply data mining methods, data is needed first. The existing blood pressure index correlation database is from China, which was collected between 2006 and 2011. It examined the optimal index for predicting the occurrence of high blood pressure in people aged 18 to 65 years (BMI) and obesity.

3.2. Dataset Features

The existing database contains information on 3,253 individuals, each with 15 BMI and blood pressure characteristics.

Feature	Description	Data Type
Age	Candidates’ age	Numerical
Gender	Male/Female	Nominal
Smoke2006	Smoking status in 2006	Nominal
Drink2006	Drinking alcoholic drinks status in 2006	Nominal
Hypettension2011[1]	History of high blood pressure	Nominal
Hip Circumference 2006	Hip size in millimeters	Numerical
Waist Circumference2006	Waist size in millimeters	Numerical
Residence2006	Residency area location	Nominal
Height2006	Height in 2006, centimeters	Numerical

Weight2006	Weight in 2006, centimeters	Numerical
Height2011	Height in 2011 centimeters	Numerical
Weight2011	Height in 2011, centimeters	Numerical
Systol2006	Systolic blood pressure (mmHg)	Numerical
Diastol2006	Diastolic blood pressure (mmHg)	Numerical
Salt	Daily salt intake (1=less than 6 grams, 2=between 6 and 12 grams, 3=between 12 and 18 grams, 4=more than 18 grams)	Numerical

1: This feature is considered a target feature (label).

Section 4: Preprocessing phase

We dealt with the data in the most important research step (data preparation or data preprocessing). In the real world, data is not always complete, and this is always true for medical information. We used data processing to eliminate some of the inconsistencies and incomplete data associated with the data. Some fields, such as height and weight, which are not significant on their own, have been replaced; here we have used the BMI index calculated according to the following formula:

$$\text{BMI}=\text{Weight(kg)}/\text{Height(m}^2\text{)}$$

As a result, after filtering the data, we arrived at records with the specifications in the table below.

Feature	Description	Data Type
Age	Candidates' age	Numerical

Gender	Male/Female	Nominal
Smoke2006	Smoking status in 2006	Nominal
Drink2006	Drinking alcoholic drinks status in 2006	Nominal
Hypettension2011	History of high blood pressure	Nominal
BMI 2006	BMI in 2006	Numerical
Residence2006	Residency area location	Nominal
BMI 2011	BMI in 2011	Numerical
Systol2006	Systolic blood pressure (mmHg)	Numerical
Salt	Daily salt intake (1=less than 6 grams, 2=between 6 and 12 grams, 3=between 12 and 18 grams, 4=more than 18 grams)	Numerical

Section 5: Description of the utilized techniques

5.1. Classification methods

1. Decision Tree

The classification method based on the production of a decision tree is considered one of the machine learning methods, which is one of the supervised learning methods due to the use of an initial training set. To produce a decision tree, first, an initial set is considered, and its decision tree is built. If this tree does not respond to all cases, the tree is expanded by selecting another set. This process continues until the tree is completed to respond to all cases. The produced decision tree is a tree whose leaves represent different classes and intermediate nodes, their features, and different states.

2. Naive Bayes

Bayesian theory is a statistical classification method. In this method, different classes are considered, each in the form of a hypothesis with a probability. Each new training record increases or decreases the probability of the previous hypotheses being correct, and finally, the hypotheses with the highest probability are considered as a class and labeled. This technique combines Bayesian theory and the causal relationship between data to classify.

3. KNN Algorithm

The KNN algorithm selects the records from the set of training records that are the closest neighbors of a group consisting of K records and decides on the category of the test record based on the superiority of their corresponding label category. In simpler terms, this method selects the rows that have the largest number of records assigned to that category in the selected neighborhood. Therefore, the nearest neighbor is considered as the new record category in the K-rows that are most among all the categories. There are three considerations to tune a KNN, which are:

1. We must have a dataset.
2. We must also have a similarity calculation criterion.
3. Hyperparameter K; It must also be specified so that we can act on it.

5.2. Clustering methods

1. K-means algorithm

Despite its simplicity, this method is regarded as fundamental for many other clustering techniques, such as fuzzy clustering. It is considered an exclusive and flat approach. Various K-means algorithms have been described for this algorithm; however, they all follow an iterative procedure. In the algorithm, the number of clusters is randomly selected from among the k members, with n members assigned to the nearest cluster. After all members are assigned, the number of clusters is n-k. The cluster centers are then recalculated and assigned to the clusters according to the new centers, continuing this process until the cluster centers remain constant.

2. K-medoid algorithm

One of the clustering methods that is dependent on nodes is the k-medoid algorithm. Which means, in this method, cluster heads (known as medoids) are chosen from the nodes themselves. Therefore, it does not use the mean, which is used by k-means. In fact, the medoid of a cluster is the most central element of that cluster. The goal of this method is to reduce sensitivity to large values in the dataset. In this algorithm, each cluster is identified by one of the data points close to the center.

Section 6: Implementation

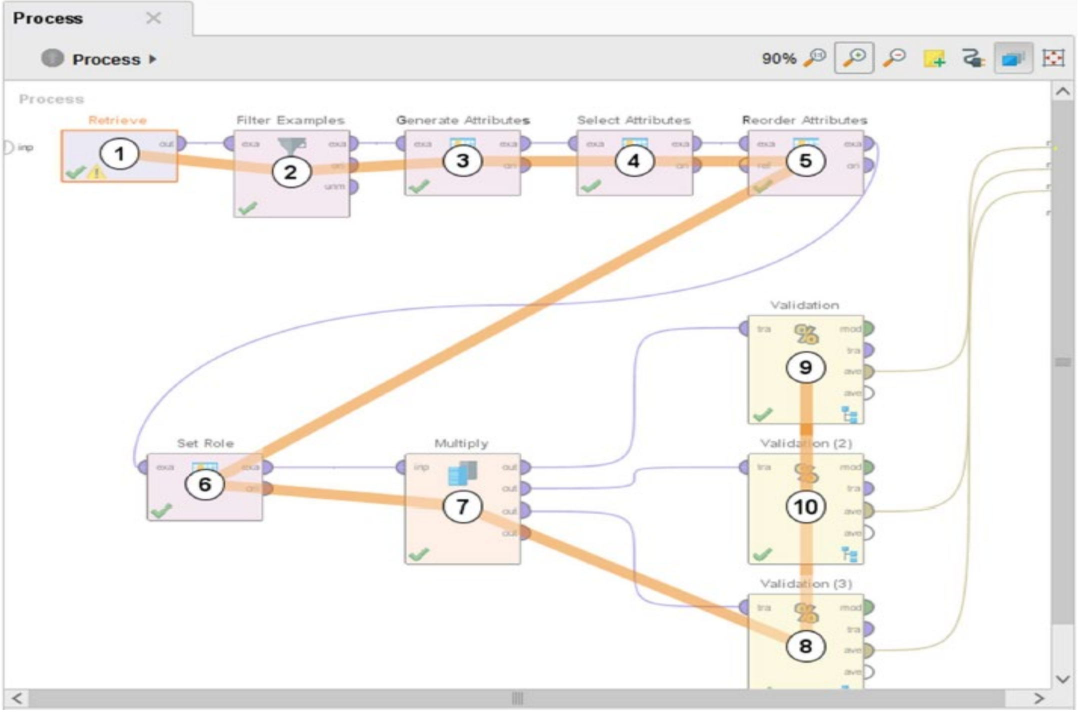
6.1. Classification

In this section, first, we want to know which of the three algorithms is better to use on our dataset, and then we will apply it.

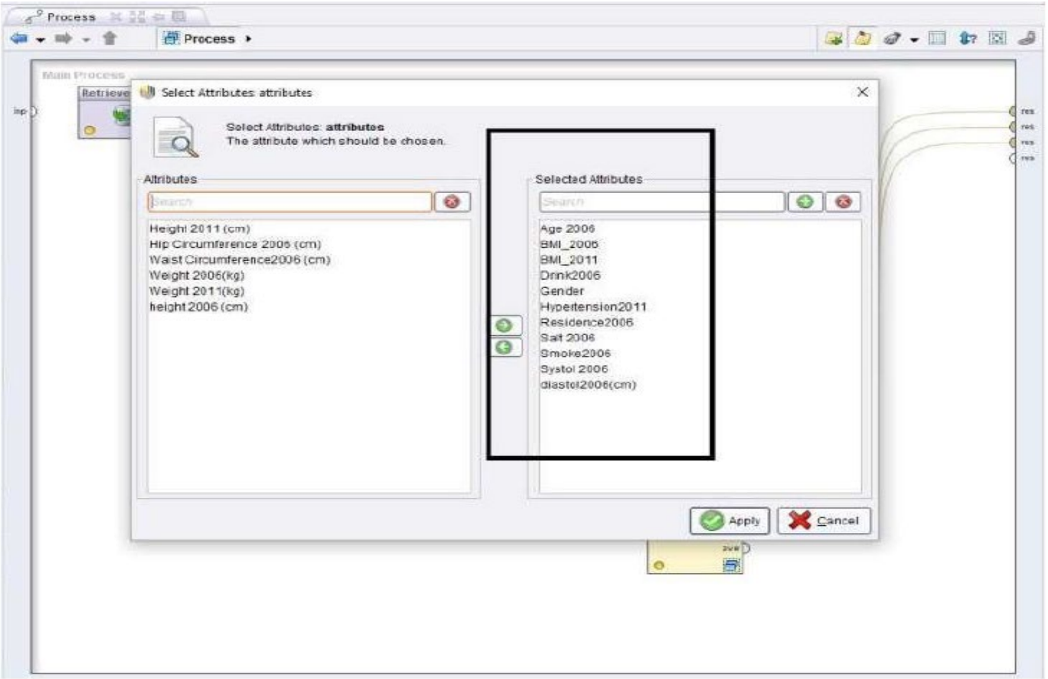
6.1.1. Choosing an appropriate model

We will classify the data using **RapidMiner software**. At first, according to the image below and operator number 1, the data has been added to the whole processing stage. Then, before doing anything, data preprocessing must be done using operators 2, 3, 4, 5, and 6, which are described as follows. We are going to apply the three classification methods we have covered before.

1. **Retrieve operator:** The first thing to do in preparing data is to prepare it for the operation that will be performed on it. This operator reads the data. To use this operator, you must have previously stored the data in the repository.
2. **Filter Example operator:** Handling missing values and null values is done via this operator. We will remove them using the Filter Example operator.
3. **Generate Attributes operator:** In many applications, you need a column that results from other columns with specific calculations. In these cases, this operator is a good choice. Here, we created two attributes, weight and height, in 2006 and 2011, using this operator.
4. **Select Attributes operator:** In many cases, you do not need all the attributes in the data set, and some of them are sufficient for your analysis. In this case, the data is filtered using this operator as shown in the image below. The attributes that you see in the box on the right side of the image are selected for our analysis (See image below).



5. **Reorder operator:** This operator is used to change columns.
6. **Set Role operator:** One of the important issues in working with data is assigning roles to specific attributes. With the help of this operator, we can assign a role to specific attributes. Here, we select the Label role for the Blood Pressure column.
7. **Multiply operator:** We used this operator to copy the output of the processed dataset; it will not change any other dataset features.



Select Attributes: attributes

Select Attributes: **attributes**
The attribute which should be chosen.

Attributes

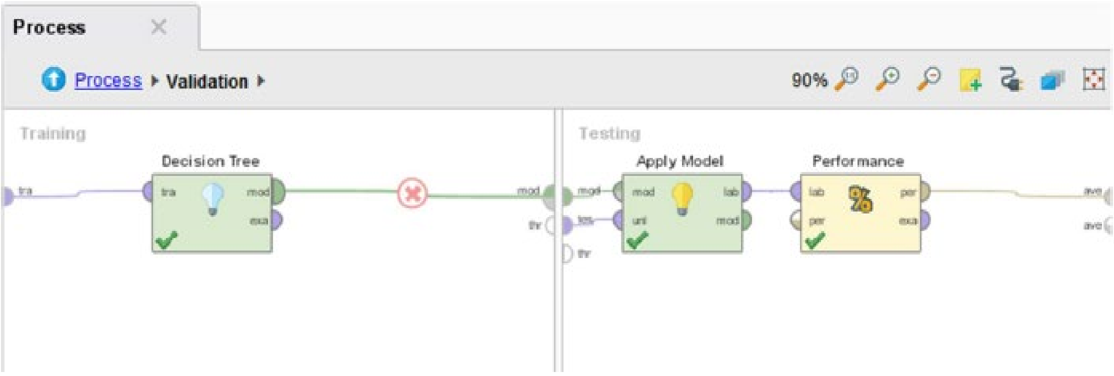
Heighl 2011 (cm)
Hip Circumference 2006 (cm)
Waist Circumference2006 (cm)
Weight 2006(kg)
Weight 2011(kg)
height 2006 (cm)

Selected Attributes

Age 2006
BMI_2006
BMI_2011
Drink2006
Gender
Hypertension2011
Residence2006
Salt 2006
Smoke2006
Systol 2006
diastol2006(cm)

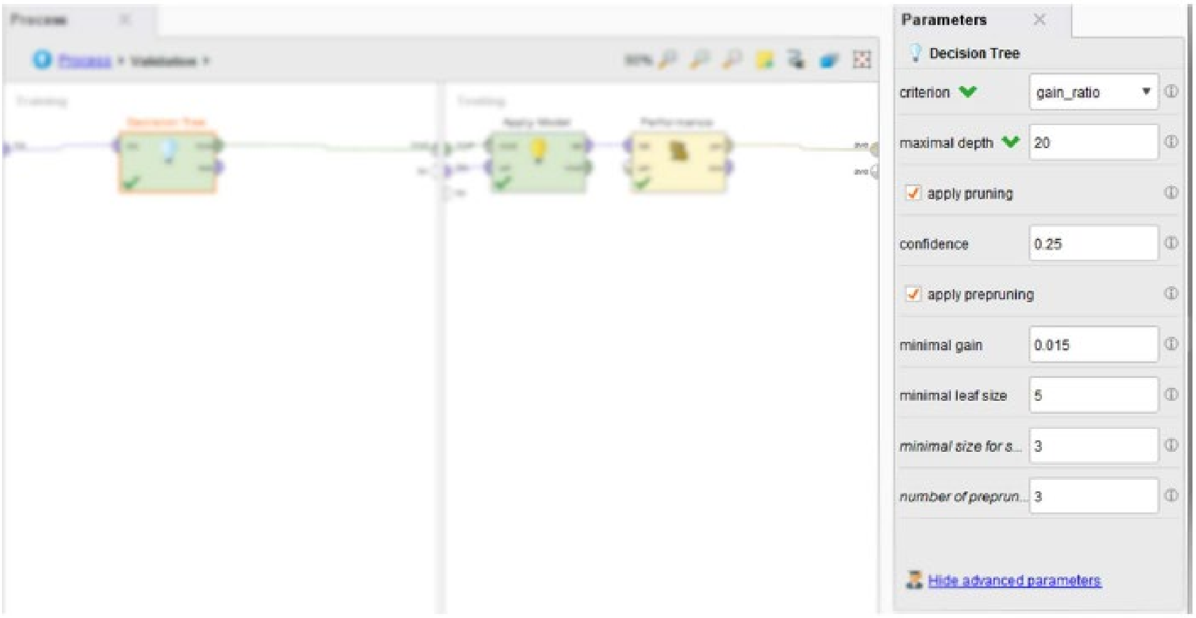
After going through steps 1 to 7, our data is ready for processing. We use three algorithms on this dataset, namely, Decision Tree, KNN, and Bayesian algorithm, and then evaluate the results of each algorithm. In the final part, we conclude which algorithm has the best performance on this dataset.

8. Here, we need to discuss more about **the validation operator (numbers 8,9, and 10)**. To validate the model we have trained or compare several models together, this operator is used. One of the common and widely used techniques in application programs to evaluate the model, especially the classification, is ***K-fold cross-validation***. Here we used this technique, and the value of k is 10. By taking 10 for K, the dataset is separated into 10 equal segments. Then, one part is saved for the training phase, and the remaining 9 segments are used in the test phase; this cycle repeats for 10 rounds (as the total segments is 10). Accuracy and mean error rate are then inferred from these 10 rounds. As we can see in the figure below, the operator has internal subprocesses. By double-clicking on this operator, we are directed to the X-validation subprocess, and we can place the appropriate operators inside it.



There are two main subprocesses in the image above:

1. **Training part**: This part is used to train a model. In this section, we use the **decision tree operator** and set the values of its parameters as shown in the image below.
2. **Testing part**: This part is used to test the trained model. We used the *Apply model* and *Performance* to apply the trained model to the dataset and then evaluate the results, respectively.



Parameters

Decision Tree

criterion: gain_ratio

maximal depth: 20

☑ apply pruning

confidence: 0.25

☑ apply prepruning

minimal gain: 0.015

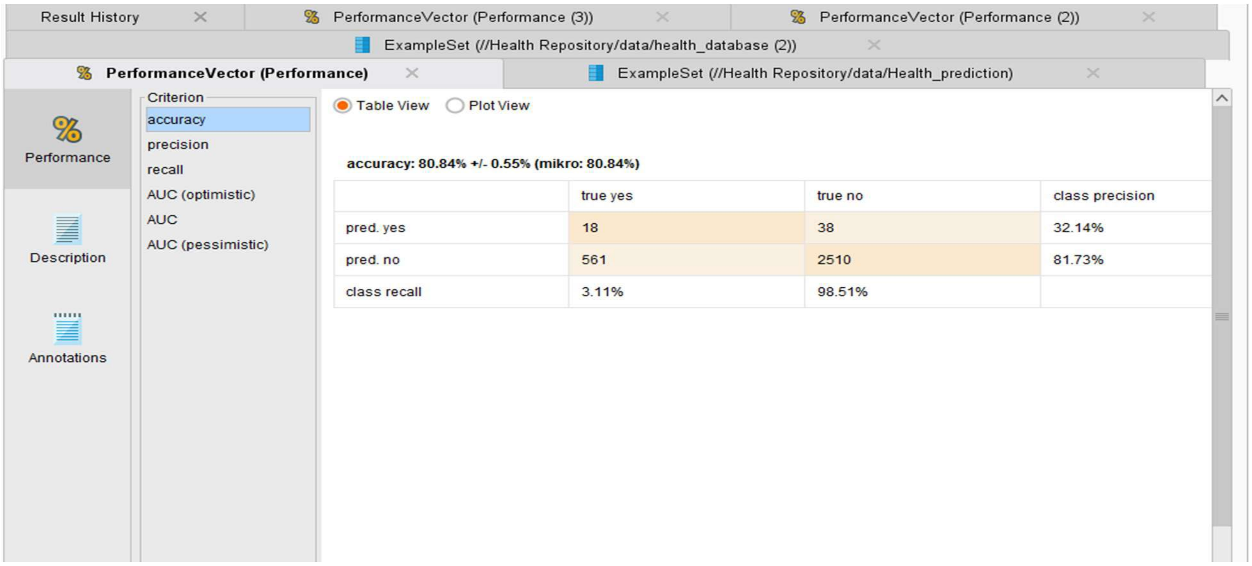
minimal leaf size: 5

minimal size for s...: 3

number of preprun...: 3

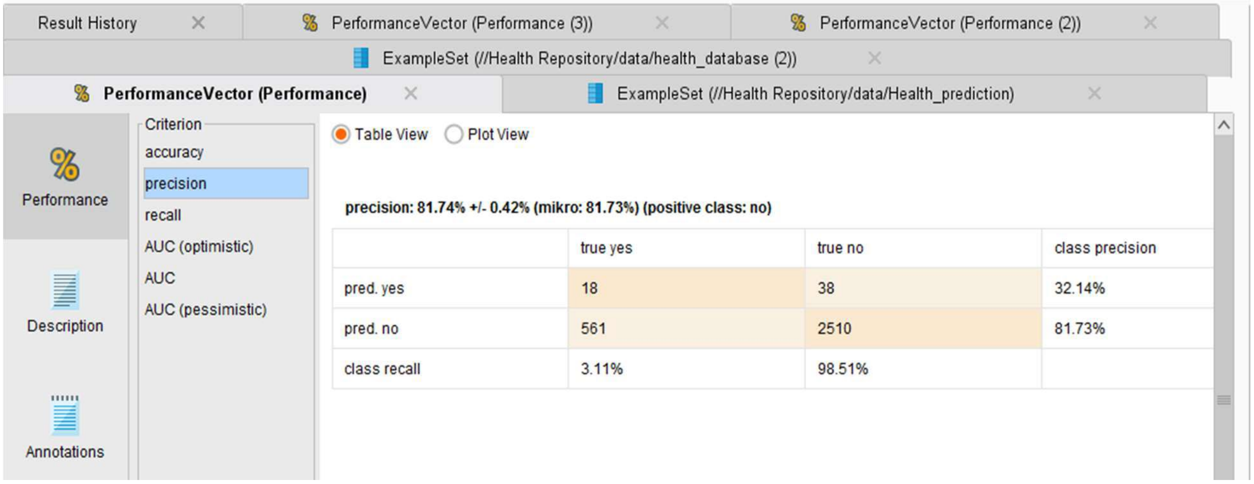
Hide advanced parameters

The following figures provide useful information about the model evaluation. The accuracy of the model is estimated to be about 80.84 percent, and other accuracies can be achieved by changing the parameters. On the left side of the image, the criteria we requested are shown, and by clicking on each one, we can see its value. For example, we chose to show **the accuracy**, **precision, and recall** tabs so the **R1-score** can be inferred easily. The table in the middle of the image shows more precise information about the estimation of the class label of the samples.


Result History
PerformanceVector (Performance (3))
PerformanceVector (Performance (2))
ExampleSet (//Health Repository/data/health_database (2))
PerformanceVector (Performance)
ExampleSet (//Health Repository/data/Health_prediction)
Performance
Description
Annotations
Criterion
accuracy
precision
recall
AUC (optimistic)
AUC
AUC (pessimistic)
● Table View
○ Plot View

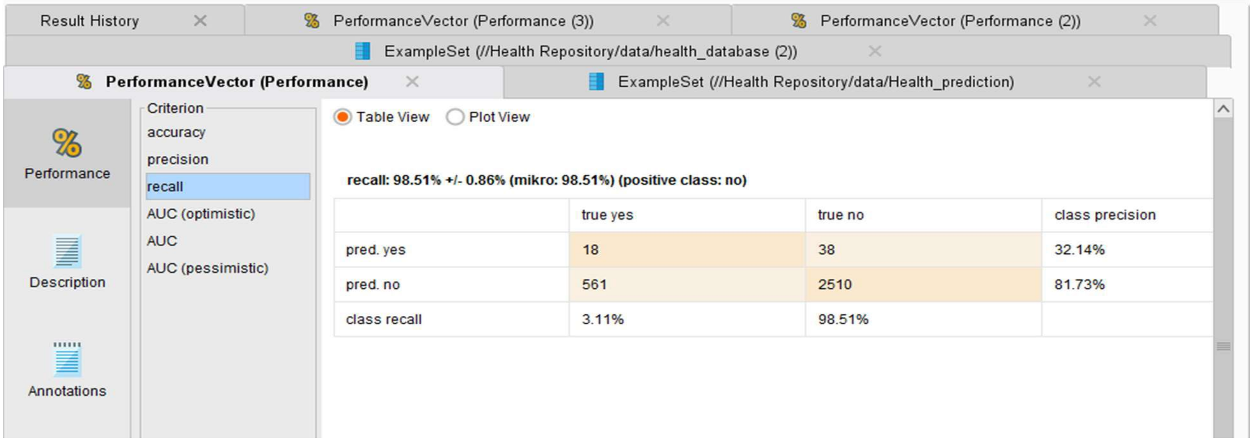
accuracy: 80.84% +/- 0.55% (mikro: 80.84%)

	true yes	true no	class precision
pred. yes	18	38	32.14%
pred. no	561	2510	81.73%
class recall	3.11%	98.51%	


Result History
PerformanceVector (Performance (3))
PerformanceVector (Performance (2))
ExampleSet (//Health Repository/data/health_database (2))
PerformanceVector (Performance)
ExampleSet (//Health Repository/data/Health_prediction)
Performance
Description
Annotations
Criterion
accuracy
precision
recall
AUC (optimistic)
AUC
AUC (pessimistic)
● Table View ○ Plot View

precision: 81.74% +/- 0.42% (mikro: 81.73%) (positive class: no)

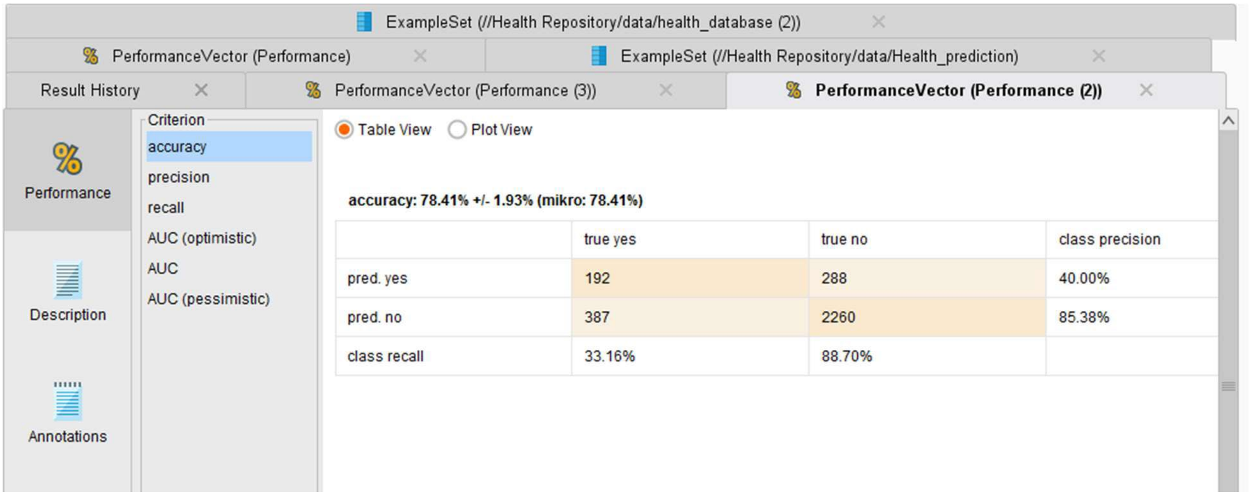
	true yes	true no	class precision
pred. yes	18	38	32.14%
pred. no	561	2510	81.73%
class recall	3.11%	98.51%	


Result History
PerformanceVector (Performance (3))
PerformanceVector (Performance (2))
ExampleSet (//Health Repository/data/health_database (2))
PerformanceVector (Performance)
ExampleSet (//Health Repository/data/Health_prediction)
Performance
Description
Annotations
Criterion
accuracy
precision
recall
AUC (optimistic)
AUC
AUC (pessimistic)
● Table View ○ Plot View

recall: 98.51% +/- 0.86% (mikro: 98.51%) (positive class: no)

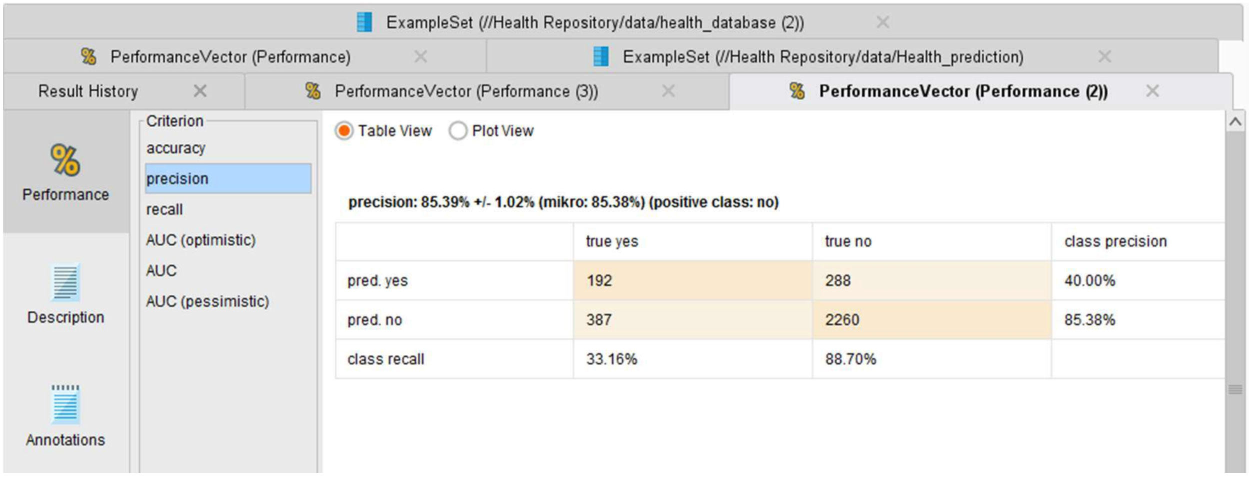
	true yes	true no	class precision
pred. yes	18	38	32.14%
pred. no	561	2510	81.73%
class recall	3.11%	98.51%	

We did the same process for **Naïve Bayes**; as we can we below, this time the accuracy was 78.41 percent.


ExampleSet (//Health Repository/data/health_database (2))
PerformanceVector (Performance)
ExampleSet (//Health Repository/data/Health_prediction)
Result History
PerformanceVector (Performance (3))
PerformanceVector (Performance (2))
Performance
Description
Annotations
Criterion
accuracy
precision
recall
AUC (optimistic)
AUC
AUC (pessimistic)
● Table View ○ Plot View

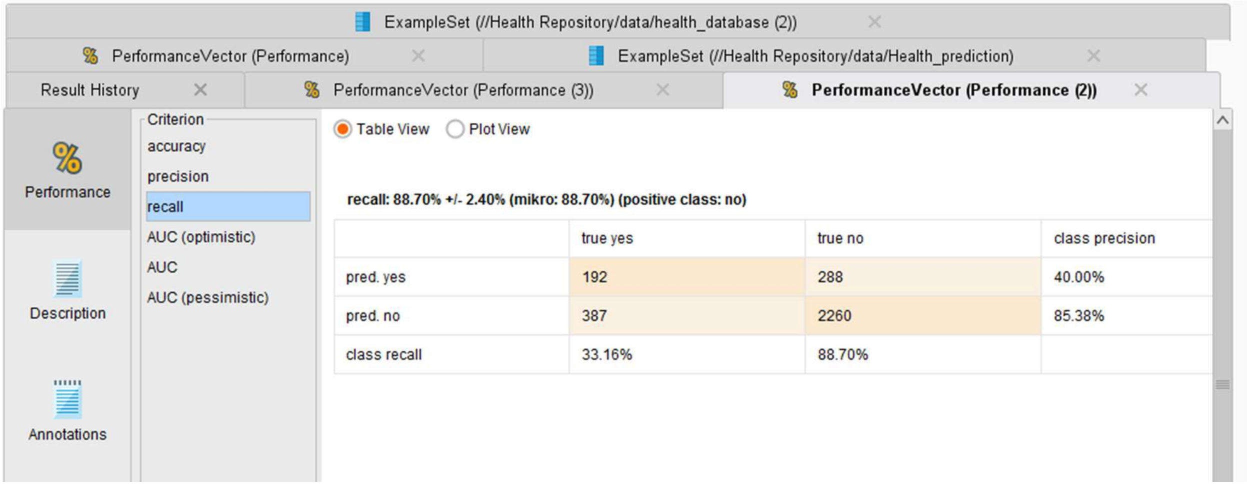
accuracy: 78.41% +/- 1.93% (mikro: 78.41%)

	true yes	true no	class precision
pred. yes	192	288	40.00%
pred. no	387	2260	85.38%
class recall	33.16%	88.70%	


ExampleSet (//Health Repository/data/health_database (2))
PerformanceVector (Performance)
ExampleSet (//Health Repository/data/Health_prediction)
Result History
PerformanceVector (Performance (3))
PerformanceVector (Performance (2))
Performance
Description
Annotations
Criterion
accuracy
precision
recall
AUC (optimistic)
AUC
AUC (pessimistic)
Table View
Plot View

precision: 85.39% +/- 1.02% (mikro: 85.38%) (positive class: no)

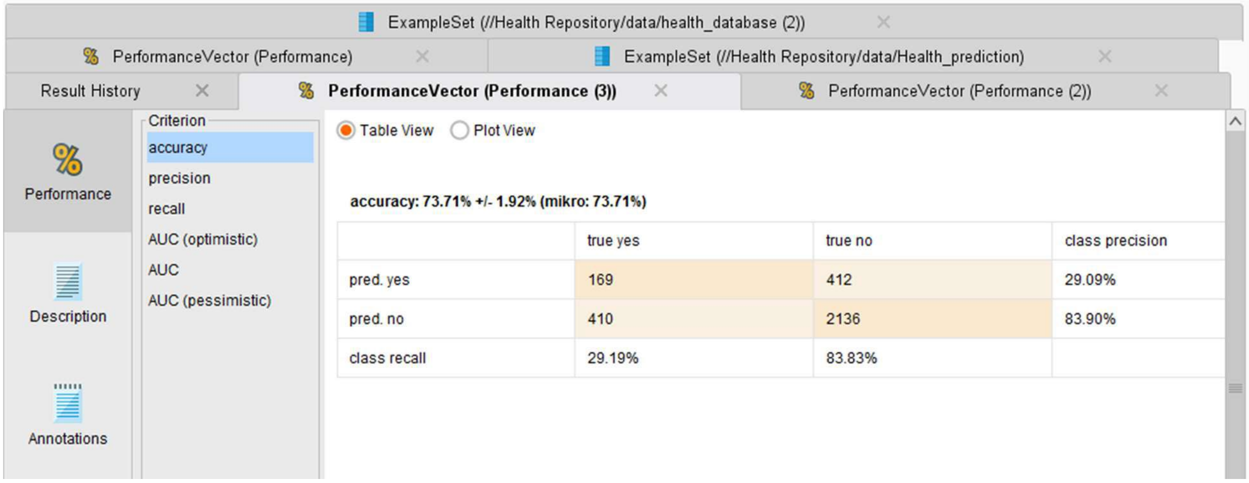
	true yes	true no	class precision
pred. yes	192	288	40.00%
pred. no	387	2260	85.38%
class recall	33.16%	88.70%	


ExampleSet (//Health Repository/data/health_database (2))
PerformanceVector (Performance)
ExampleSet (//Health Repository/data/Health_prediction)
Result History
PerformanceVector (Performance (3))
PerformanceVector (Performance (2))
Performance
Description
Annotations
Criterion
accuracy
precision
recall
AUC (optimistic)
AUC
AUC (pessimistic)
Table View
Plot View

recall: 88.70% +/- 2.40% (mikro: 88.70%) (positive class: no)

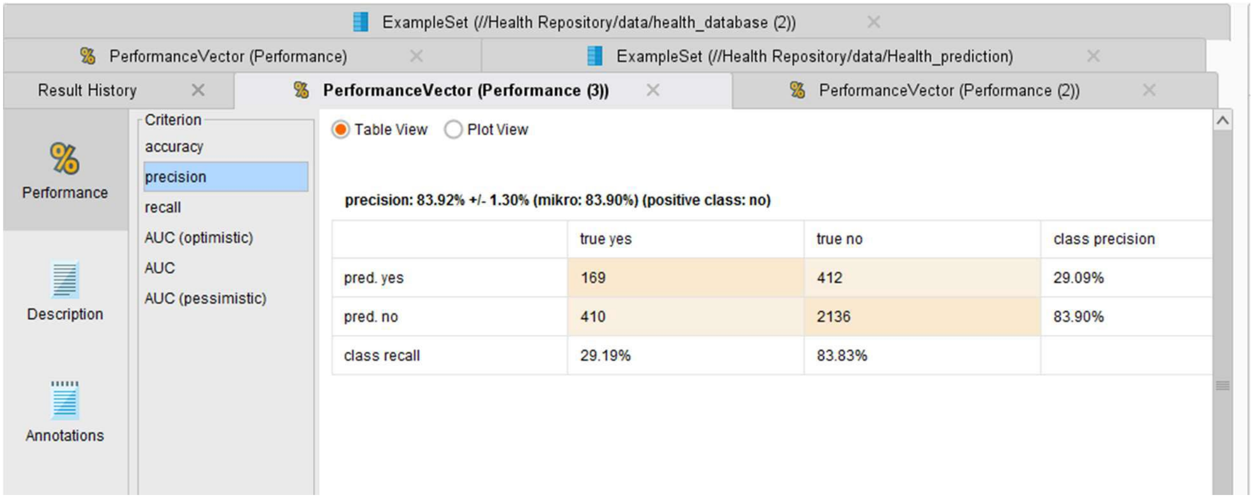
	true yes	true no	class precision
pred. yes	192	288	40.00%
pred. no	387	2260	85.38%
class recall	33.16%	88.70%	

So, as we aimed to compare 3 algorithms in the classification phase, the **KNN algorithm** was the third one. We can see the results below. With parameters set: *K = 4*, and distance measurement: *Euclidean distance*


ExampleSet (//Health Repository/data/health_database (2))
PerformanceVector (Performance)
ExampleSet (//Health Repository/data/Health_prediction)
Result History
PerformanceVector (Performance (3))
PerformanceVector (Performance (2))
Performance
Description
Annotations
Criterion
accuracy
precision
recall
AUC (optimistic)
AUC
AUC (pessimistic)
Table View
Plot View

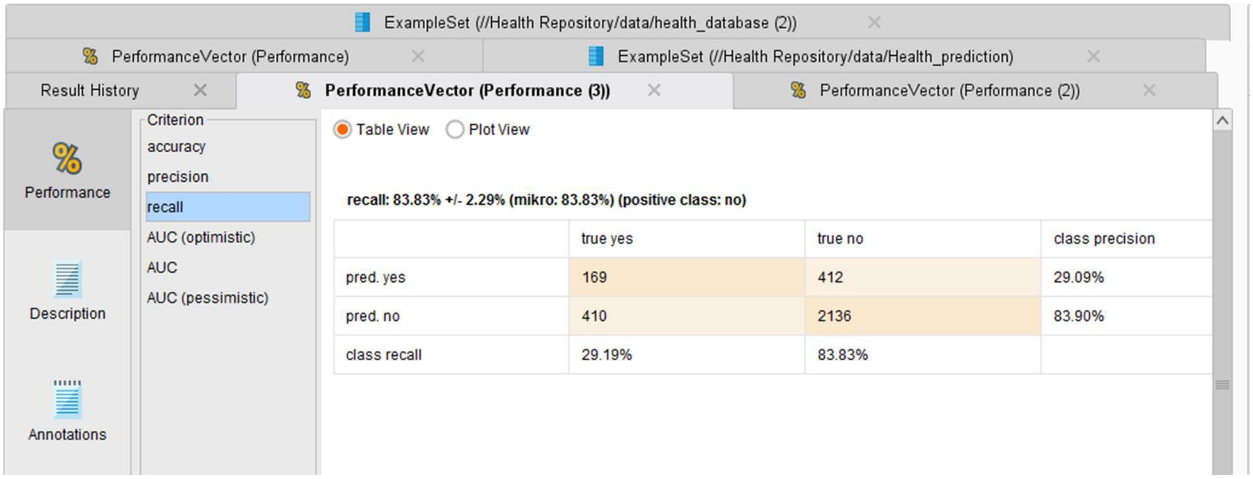
accuracy: 73.71% +/- 1.92% (mikro: 73.71%)

	true yes	true no	class precision
pred. yes	169	412	29.09%
pred. no	410	2136	83.90%
class recall	29.19%	83.83%	


ExampleSet (//Health Repository/data/health_database (2))
PerformanceVector (Performance)
ExampleSet (//Health Repository/data/Health_prediction)
Result History
PerformanceVector (Performance (3))
PerformanceVector (Performance (2))
Performance
Description
Annotations
Criterion
accuracy
precision
recall
AUC (optimistic)
AUC
AUC (pessimistic)
● Table View
○ Plot View

precision: 83.92% +/- 1.30% (mikro: 83.90%) (positive class: no)

	true yes	true no	class precision
pred. yes	169	412	29.09%
pred. no	410	2136	83.90%
class recall	29.19%	83.83%	


ExampleSet (//Health Repository/data/health_database (2))
PerformanceVector (Performance)
ExampleSet (//Health Repository/data/Health_prediction)
Result History
PerformanceVector (Performance (3))
PerformanceVector (Performance (2))
Performance
Description
Annotations
Criterion
accuracy
precision
recall
AUC (optimistic)
AUC
AUC (pessimistic)
● Table View
○ Plot View

recall: 83.83% +/- 2.29% (mikro: 83.83%) (positive class: no)

	true yes	true no	class precision
pred. yes	169	412	29.09%
pred. no	410	2136	83.90%
class recall	29.19%	83.83%	

6.1.2. Conclusion

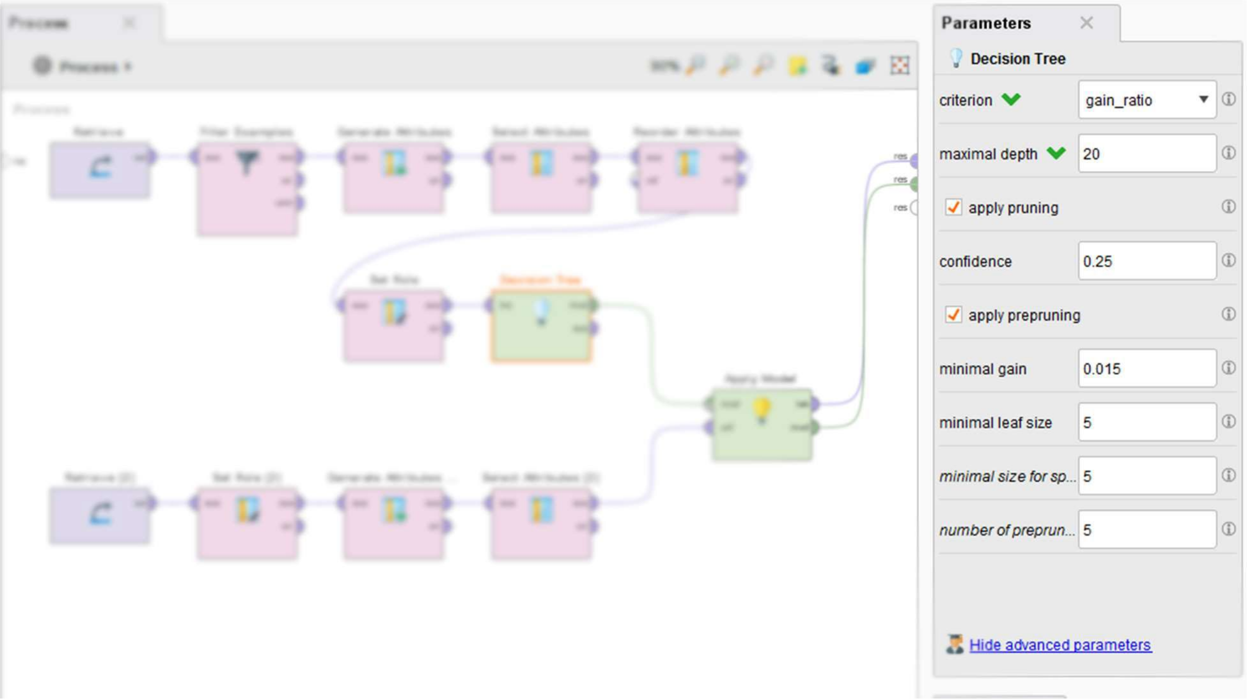
Below, we can see the three classification algorithms' results:

	DECISION TREE	KNN	NAÏVE BAYES
ACCURACY (%)	80.84	73.71	78.41
PRECISION (%)	81.74	83.92	85.39
RECALL (%)	98.51	87.70	83.83
AREA UNDER THE CURVE (AUC)	0.638	0.500	0.631

As can be seen, the decision tree was slightly better than the other two classifiers, so we chose it in the next part of the project to forecast some new samples.

6.1.3. Label prediction using the pre-trained decision tree model

Here we have a dataset of 100 samples whose blood pressure attribute value is unknown. We want to predict the value of this attribute for this dataset using the built decision tree model. The values available for the attributes of the test data must be within the range of the attribute values in the training data. We will remove the out-of-range test data. If it is not, use the Filter Example operator as explained in the previous section. The ***Set Role*** and ***Reorder Attributes***, ***Select Attributes***, and ***Generate Attributes*** operators are used to preprocess the data. Then the decision tree operator is applied to the training dataset and its parameters are determined as shown in the image below.



Parameters

Decision Tree

criterion: gain_ratio

maximal depth: 20

☑ apply pruning

confidence: 0.25

☑ apply prepruning

minimal gain: 0.015

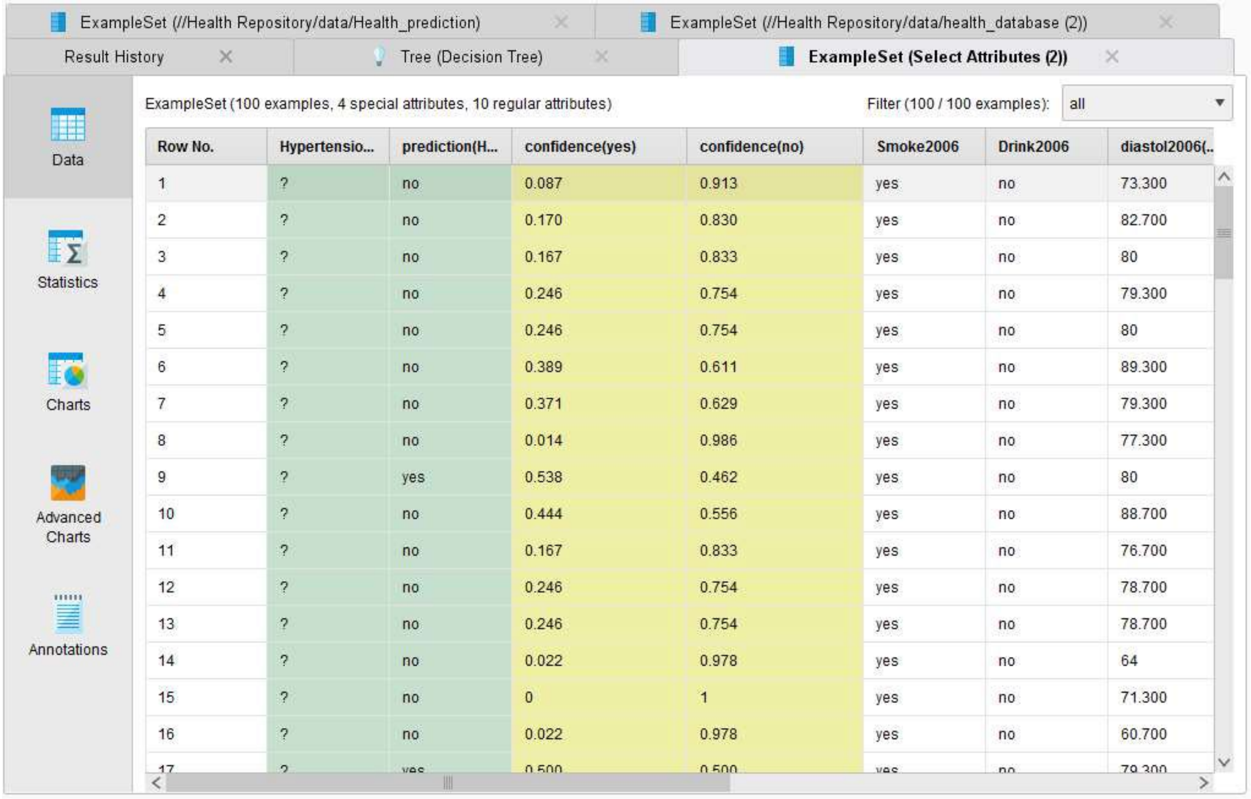
minimal leaf size: 5

minimal size for sp...: 5

number of preprun...: 5

Then, the output of the decision tree operator and the output from the preprocessing of the training data are applied to the operator to predict the label of the experimental data. We will then observe the output from this processing in the Model.

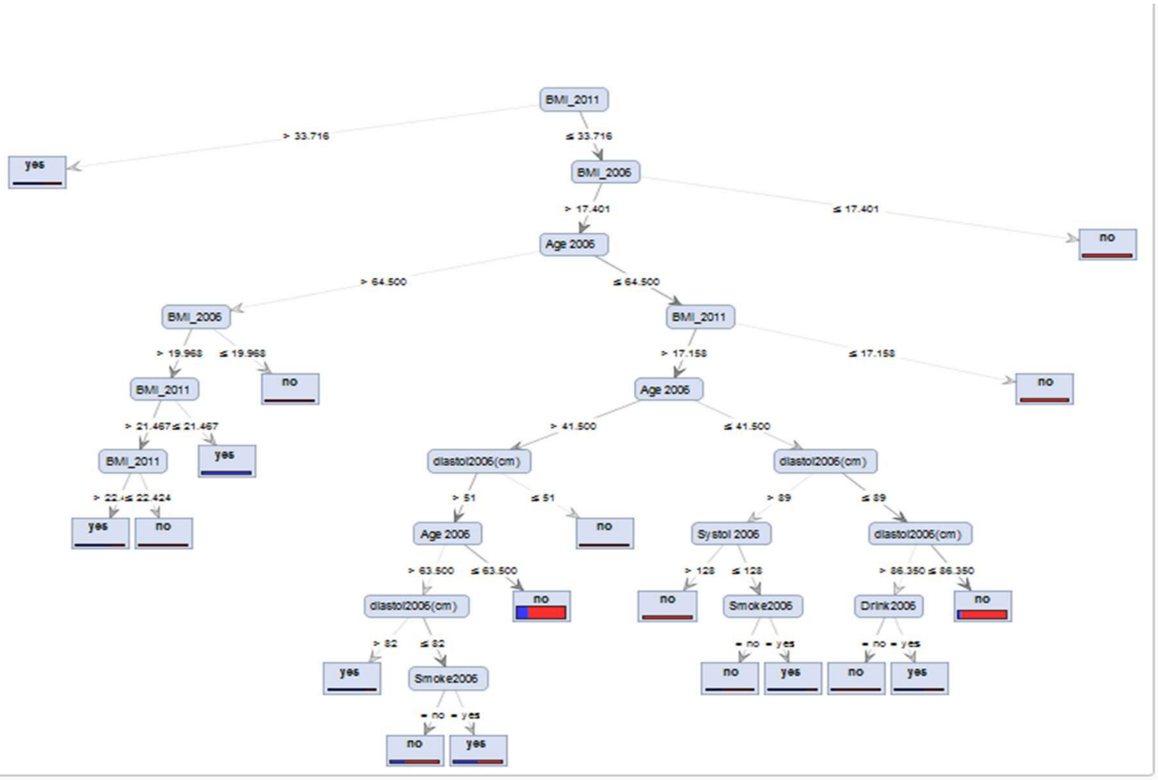
Looking at the images below, we see that body mass index(BMI) is at the first level. The most information is obtained through this index and we get to the label faster through this index. Therefore, the body mass index feature has the greatest impact on the class label and the most division on the tree is done by this index. Of the 17 branches on the tree, 7 are related to body mass index, which again shows the importance of this index.


ExampleSet (//Health Repository/data/Health_prediction)
ExampleSet (//Health Repository/data/health_database (2))
Result History
Tree (Decision Tree)
ExampleSet (Select Attributes (2))
Data
Statistics
Charts
Advanced Charts
Annotations

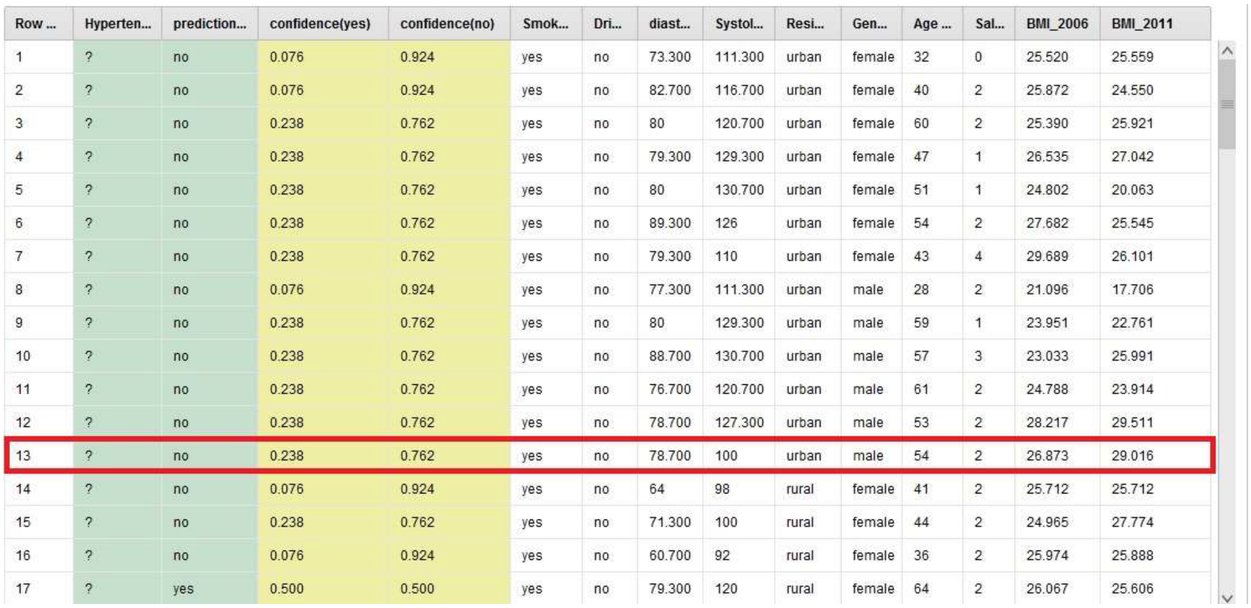
ExampleSet (100 examples, 4 special attributes, 10 regular attributes)

Filter (100 / 100 examples): all

Row No.	Hypertensio...	prediction(H...	confidence(yes)	confidence(no)	Smoke2006	Drink2006	diastol2006(...
1	?	no	0.087	0.913	yes	no	73.300
2	?	no	0.170	0.830	yes	no	82.700
3	?	no	0.167	0.833	yes	no	80
4	?	no	0.246	0.754	yes	no	79.300
5	?	no	0.246	0.754	yes	no	80
6	?	no	0.389	0.611	yes	no	89.300
7	?	no	0.371	0.629	yes	no	79.300
8	?	no	0.014	0.986	yes	no	77.300
9	?	yes	0.538	0.462	yes	no	80
10	?	no	0.444	0.556	yes	no	88.700
11	?	no	0.167	0.833	yes	no	76.700
12	?	no	0.246	0.754	yes	no	78.700
13	?	no	0.246	0.754	yes	no	78.700
14	?	no	0.022	0.978	yes	no	64
15	?	no	0	1	yes	no	71.300
16	?	no	0.022	0.978	yes	no	60.700
17	?	yes	0.500	0.500	yes	no	79.300



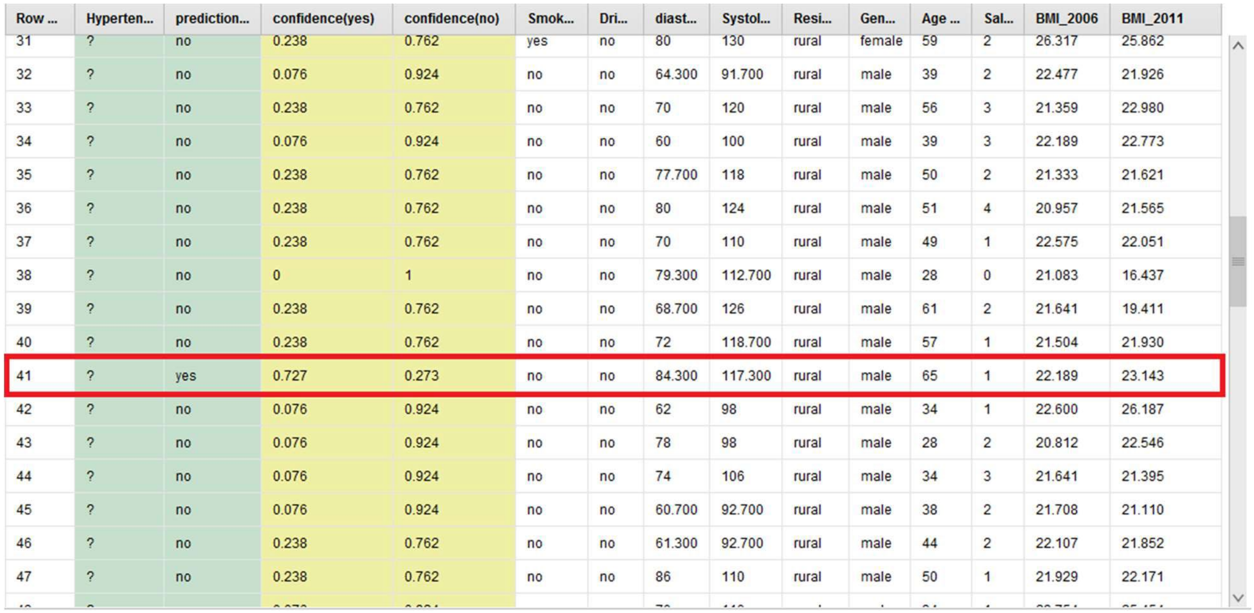
Label prediction for sample 13:



Row ...	Hyperten...	prediction...	confidence(yes)	confidence(no)	Smok...	Dri...	diast...	Systol...	Resi...	Gen...	Age ...	Sal...	BMI_2006	BMI_2011
1	?	no	0.076	0.924	yes	no	73.300	111.300	urban	female	32	0	25.520	25.559
2	?	no	0.076	0.924	yes	no	82.700	116.700	urban	female	40	2	25.872	24.550
3	?	no	0.238	0.762	yes	no	80	120.700	urban	female	60	2	25.390	25.921
4	?	no	0.238	0.762	yes	no	79.300	129.300	urban	female	47	1	26.535	27.042
5	?	no	0.238	0.762	yes	no	80	130.700	urban	female	51	1	24.802	20.063
6	?	no	0.238	0.762	yes	no	89.300	126	urban	female	54	2	27.682	25.545
7	?	no	0.238	0.762	yes	no	79.300	110	urban	female	43	4	29.689	26.101
8	?	no	0.076	0.924	yes	no	77.300	111.300	urban	male	28	2	21.096	17.706
9	?	no	0.238	0.762	yes	no	80	129.300	urban	male	59	1	23.951	22.761
10	?	no	0.238	0.762	yes	no	88.700	130.700	urban	male	57	3	23.033	25.991
11	?	no	0.238	0.762	yes	no	76.700	120.700	urban	male	61	2	24.788	23.914
12	?	no	0.238	0.762	yes	no	78.700	127.300	urban	male	53	2	28.217	29.511
13	?	no	0.238	0.762	yes	no	78.700	100	urban	male	54	2	26.873	29.016
14	?	no	0.076	0.924	yes	no	64	98	rural	female	41	2	25.712	25.712
15	?	no	0.238	0.762	yes	no	71.300	100	rural	female	44	2	24.965	27.774
16	?	no	0.076	0.924	yes	no	60.700	92	rural	female	36	2	25.974	25.888
17	?	yes	0.500	0.500	yes	no	79.300	120	rural	female	64	2	26.067	25.606

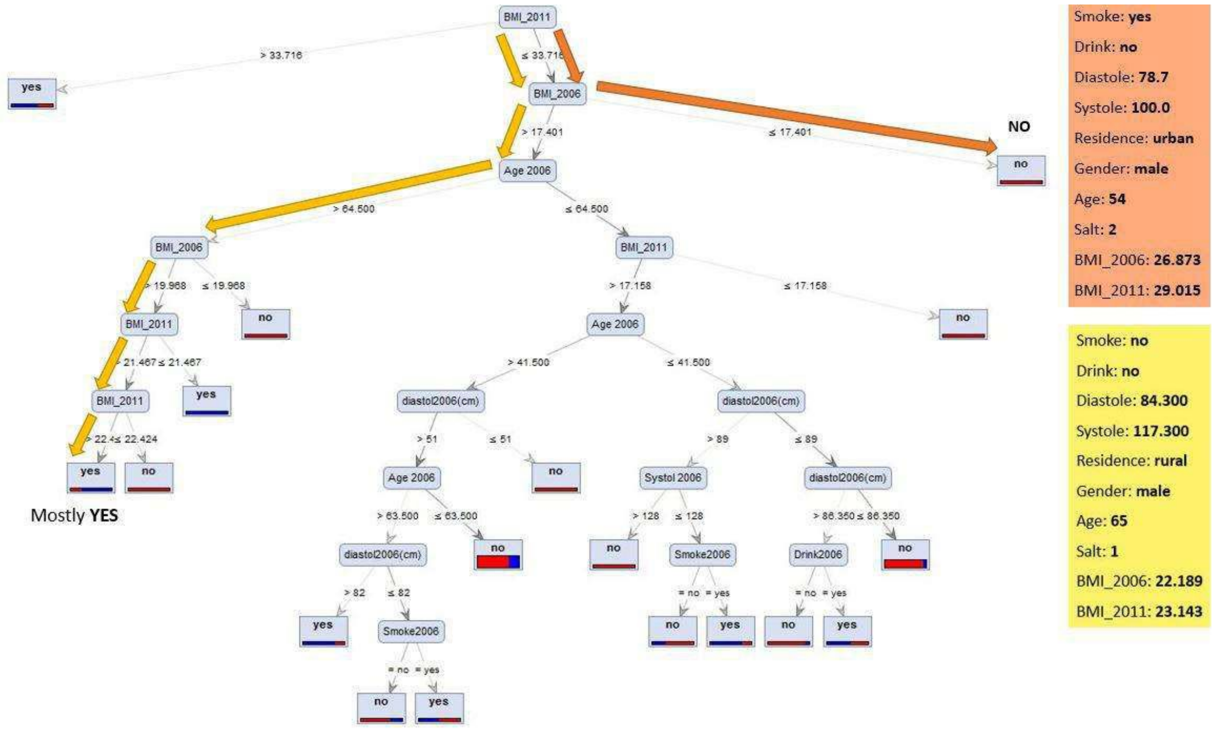
As you can see in the image above, a 54-year-old man living in an urban area with a blood pressure of 10 over 7 and a smoker is selected for prediction analysis. His body mass index changed from 26.87 in 2006 to 29.01 in 2011. He has a probability of 0.72 of not having high blood pressure and a probability of 0.23 of having high blood pressure, which in **this sample is predicted to be non-hypertensive.**

Label prediction for sample 42:



Row ...	Hyperten...	prediction...	confidence(yes)	confidence(no)	Smok...	Dri...	diast...	Systol...	Resi...	Gen...	Age ...	Sal...	BMI_2006	BMI_2011
31	?	no	0.238	0.762	yes	no	80	130	rural	female	59	2	26.317	25.862
32	?	no	0.076	0.924	no	no	64.300	91.700	rural	male	39	2	22.477	21.926
33	?	no	0.238	0.762	no	no	70	120	rural	male	56	3	21.359	22.980
34	?	no	0.076	0.924	no	no	60	100	rural	male	39	3	22.189	22.773
35	?	no	0.238	0.762	no	no	77.700	118	rural	male	50	2	21.333	21.621
36	?	no	0.238	0.762	no	no	80	124	rural	male	51	4	20.957	21.565
37	?	no	0.238	0.762	no	no	70	110	rural	male	49	1	22.575	22.051
38	?	no	0	1	no	no	79.300	112.700	rural	male	28	0	21.083	16.437
39	?	no	0.238	0.762	no	no	68.700	126	rural	male	61	2	21.641	19.411
40	?	no	0.238	0.762	no	no	72	118.700	rural	male	57	1	21.504	21.930
41	?	yes	0.727	0.273	no	no	84.300	117.300	rural	male	65	1	22.189	23.143
42	?	no	0.076	0.924	no	no	62	98	rural	male	34	1	22.600	26.187
43	?	no	0.076	0.924	no	no	78	98	rural	male	28	2	20.812	22.546
44	?	no	0.076	0.924	no	no	74	106	rural	male	34	3	21.641	21.395
45	?	no	0.076	0.924	no	no	60.700	92.700	rural	male	38	2	21.708	21.110
46	?	no	0.238	0.762	no	no	61.300	92.700	rural	male	44	2	22.107	21.852
47	?	no	0.238	0.762	no	no	86	110	rural	male	50	1	21.929	22.171

In this example, you can see that a 65-year-old man living in a rural area with a blood pressure of 11 over 8 and a BMI change of 22.18 to 23.14 has a probability of 0.27 that he does not have hypertension and a probability of 0.72 that he does have **hypertension**, which is **predicted** here. You can see the path of the two examples above on the tree in the image below.



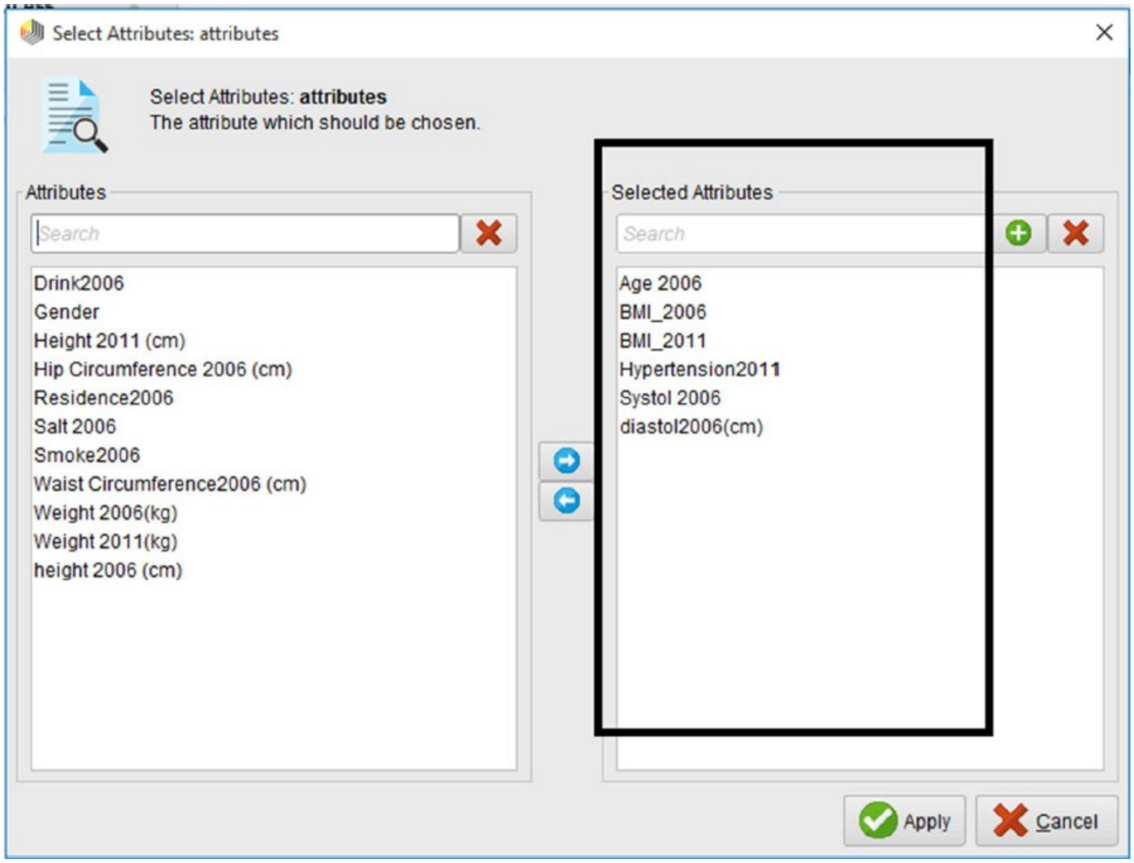
Smoke: **yes**
Drink: **no**
Diastole: **78.7**
Systole: **100.0**
Residence: **urban**
Gender: **male**
Age: **54**
Salt: **2**
BMI_2006: **26.873**
BMI_2011: **29.015**

Smoke: **no**
Drink: **no**
Diastole: **84.300**
Systole: **117.300**
Residence: **rural**
Gender: **male**
Age: **65**
Salt: **1**
BMI_2006: **22.189**
BMI_2011: **23.143**

6.1.4. Correlation and Confusion matrix

In this part, we intend to obtain the correlation of the features with respect to each other. The primitive operators are applied to the dataset to perform preprocessing and data preparation according to the ***select attribute operator***. The operator shown in the image below is used to specify the features whose correlation we want to obtain with each other.

In the next step, the output of the above is connected to the ***correlation matrix operator*** as the input. This process is shown in the second image below.


Select Attributes: attributes

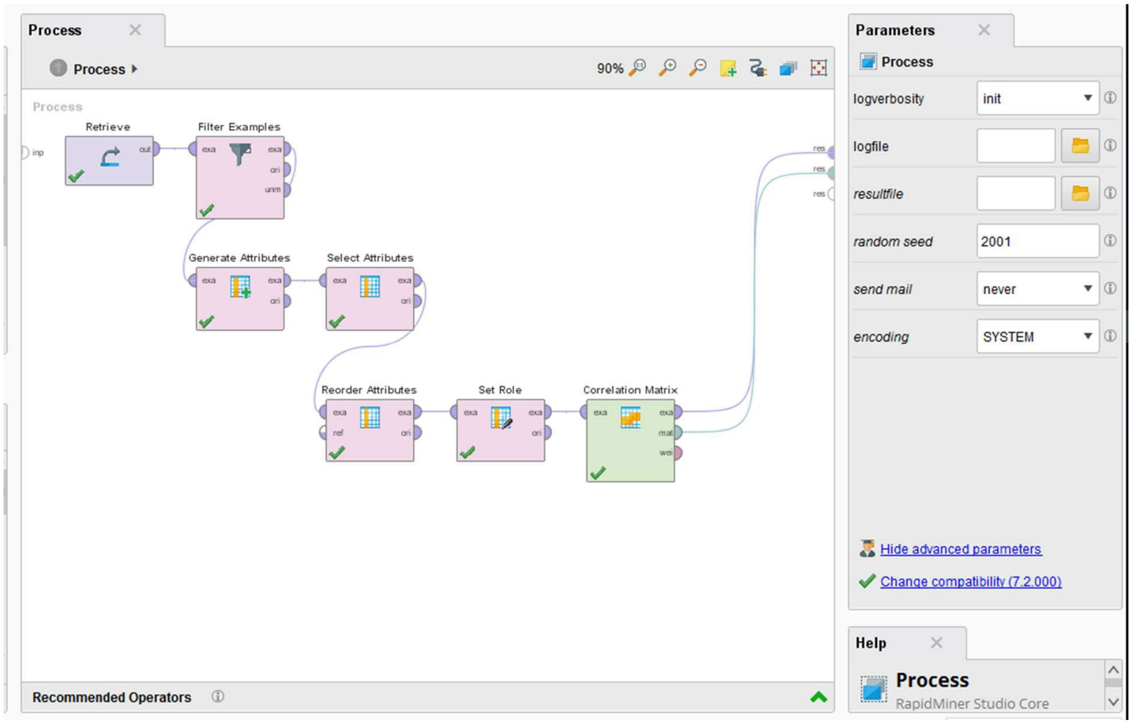
Select Attributes: **attributes**
The attribute which should be chosen.

Attributes

Drink2006
Gender
Height 2011 (cm)
Hip Circumference 2006 (cm)
Residence2006
Salt 2006
Smoke2006
Waist Circumference2006 (cm)
Weight 2006(kg)
Weight 2011(kg)
height 2006 (cm)

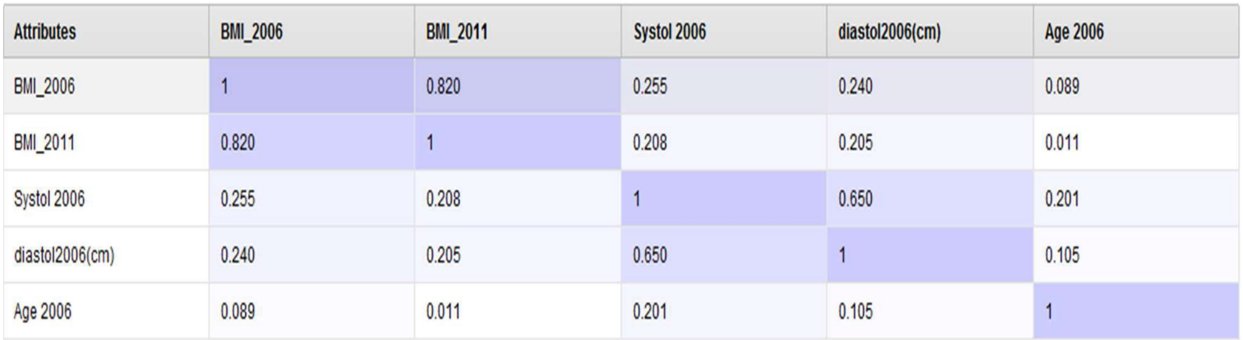
Selected Attributes

Age 2006
BMI_2006
BMI_2011
Hypertension2011
Systol 2006
diastol2006(cm)



logverbosity	init
logfile	
resultfile	
random seed	2001
send mail	never
encoding	SYSTEM

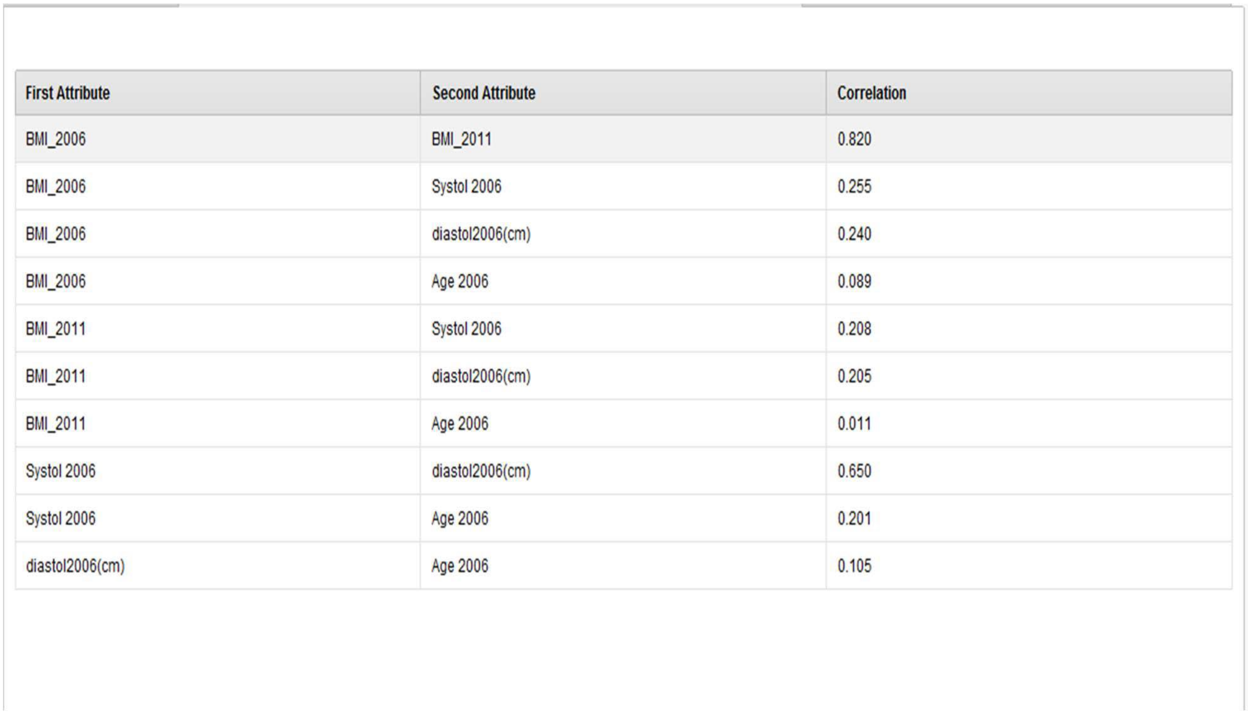
The correlation matrix is now calculated as shown in the following image.



Attributes	BMI_2006	BMI_2011	Systol 2006	diastol2006(cm)	Age 2006
BMI_2006	1	0.820	0.255	0.240	0.089
BMI_2011	0.820	1	0.208	0.205	0.011
Systol 2006	0.255	0.208	1	0.650	0.201
diastol2006(cm)	0.240	0.205	0.650	1	0.105
Age 2006	0.089	0.011	0.201	0.105	1

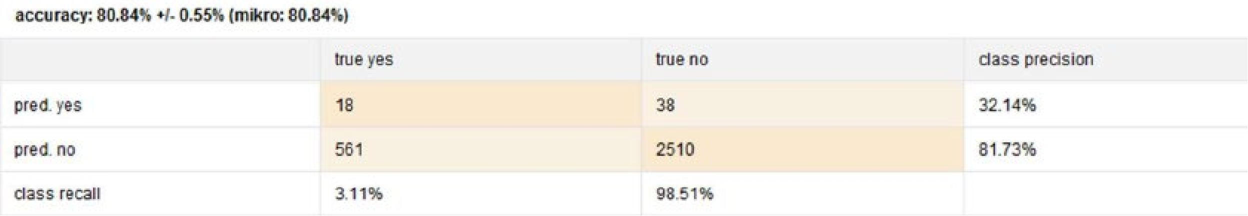
In the image above, we can see that body mass index (BMI) has a correlation of about 0.2 with systolic and diastolic blood pressure, which is a **relatively good correlation**, and as body mass index increases or decreases, systolic and diastolic blood pressure also increase or decrease. The **age** characteristic also has a **relatively good correlation** with **systolic** and **diastolic** blood pressure, and as age increases, systolic and diastolic blood pressure also increase. A similar analysis can be performed for each of the characteristics using this matrix.

The image below shows the pairwise correlation between the features.



First Attribute	Second Attribute	Correlation
BMI_2006	BMI_2011	0.820
BMI_2006	Systol 2006	0.255
BMI_2006	diastol2006(cm)	0.240
BMI_2006	Age 2006	0.089
BMI_2011	Systol 2006	0.208
BMI_2011	diastol2006(cm)	0.205
BMI_2011	Age 2006	0.011
Systol 2006	diastol2006(cm)	0.650
Systol 2006	Age 2006	0.201
diastol2006(cm)	Age 2006	0.105

After running the classification algorithms on the data set, the results are stored in vectors that are the same dimension as the label vector of the experimental data and can be compared. The image below shows this comparison for the decision tree model.



accuracy: 80.84% +/- 0.55% (mikro: 80.84%)

	true yes	true no	class precision
pred. yes	18	38	32.14%
pred. no	561	2510	81.73%
class recall	3.11%	98.51%	

As can be seen, 18 people had high blood pressure, which the decision tree correctly identified. Also, 2510 people did not have high blood pressure, which was correctly identified. However, 38 people did not have high blood pressure, which was incorrectly identified as high blood pressure, and 561 people had high blood pressure, which was incorrectly identified as low blood pressure. Therefore, we reached an accuracy of 80.84% for the decision tree. Note that this reduction is negligible due to the application of the Precision class 'Yes' to the decision tree and the removal of extra branches from the tree as a result of **pruning**.

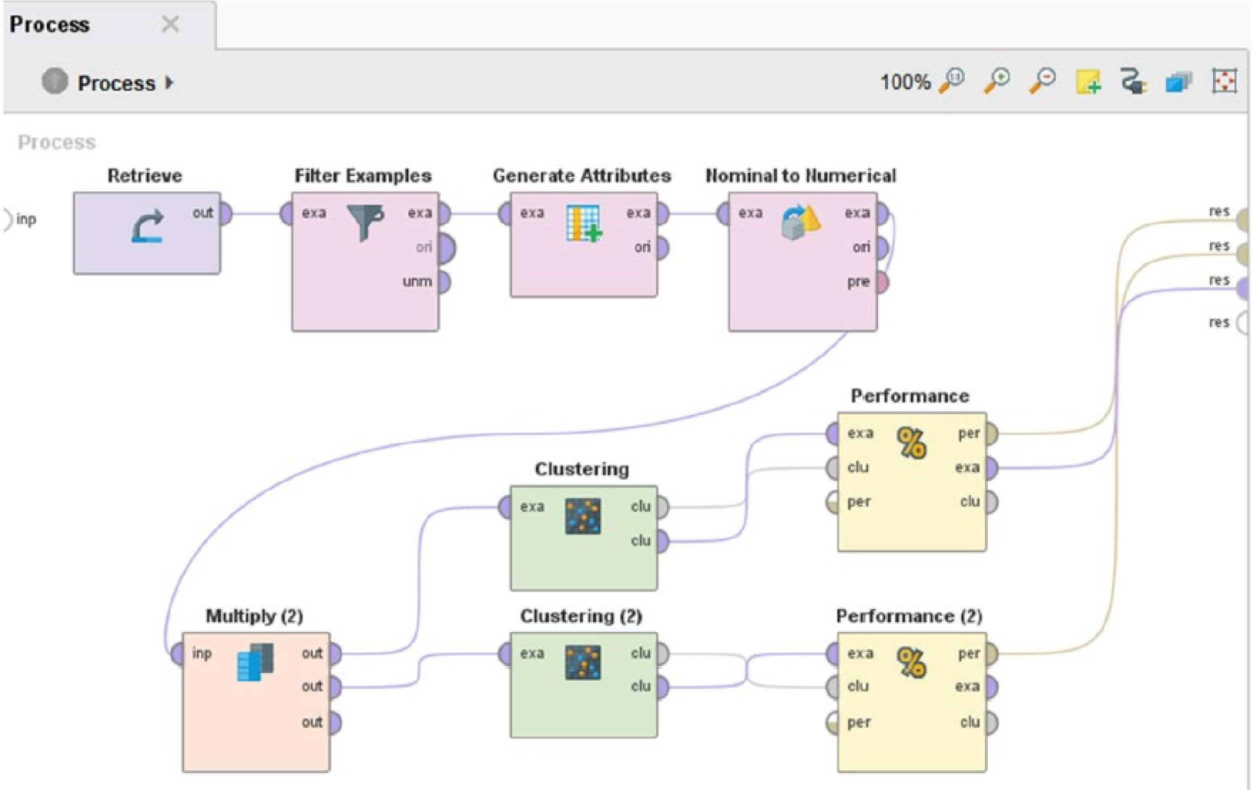
6.2. Clustering

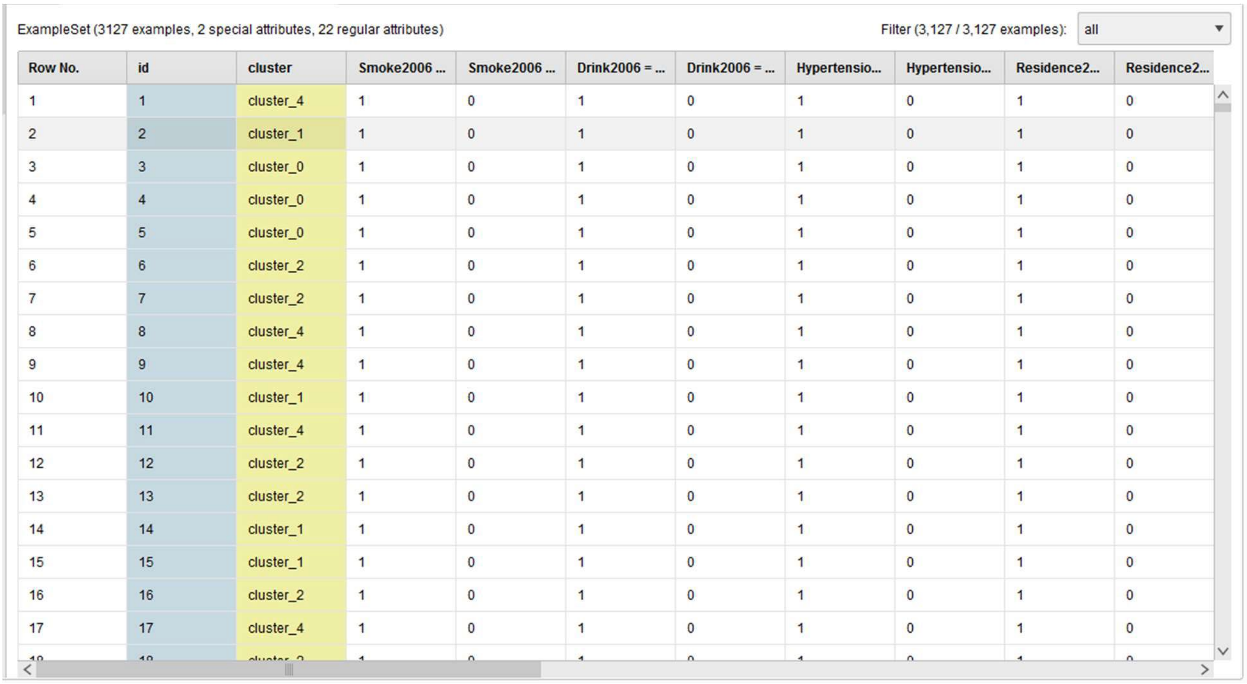
In this part, records are placed in multiple clusters based on their similarity to each other and their dissimilarity to other records. In this technique, there are no training and testing stages, and at the end, a model is created that is practically the same as determining the clusters and is presented as an output along with its efficiency. In this step, we have used **K-medoid** and **K-means** algorithms and evaluated the results of these algorithms, and the most suitable algorithm is selected.

6.2.1. Operators Description

Looking at the image below, we skip the first three operators as we described them in the previous section.

1. **Nominal to Numerical operator:** It is used to convert nominal data to numerical data and must be preceded by the K-medoids, K-means Clustering operators.
2. **Multiply operator**: The Multiply operator does not make any changes to the data and is suitable for situations where: You want to view the output of multiple processes on the same data.
3. **Clustering operator**: This is the operator we used to apply clustering methods to the dataset. Here we used **K-means** with parameter **k=5**. We can see the result of this clustering in the second image below. As it is shown, sample numbers, **2,10,14, and 15** are clustered in **cluster_1**. We can see that clusters are numbered like **cluster_0** to **cluster_4**
4. **Clustering (2) operator**: Just like the previous clustering operator, with only one difference that here we used **K-medoids** as the clustering method.





ExampleSet (3127 examples, 2 special attributes, 22 regular attributes)

Filter (3,127 / 3,127 examples): all

Row No.	id	cluster	Smoke2006 ...	Smoke2006 ...	Drink2006 = ...	Drink2006 = ...	Hypertensio...	Hypertensio...	Residence2...	Residence2...
1	1	cluster_4	1	0	1	0	1	0	1	0
2	2	cluster_1	1	0	1	0	1	0	1	0
3	3	cluster_0	1	0	1	0	1	0	1	0
4	4	cluster_0	1	0	1	0	1	0	1	0
5	5	cluster_0	1	0	1	0	1	0	1	0
6	6	cluster_2	1	0	1	0	1	0	1	0
7	7	cluster_2	1	0	1	0	1	0	1	0
8	8	cluster_4	1	0	1	0	1	0	1	0
9	9	cluster_4	1	0	1	0	1	0	1	0
10	10	cluster_1	1	0	1	0	1	0	1	0
11	11	cluster_4	1	0	1	0	1	0	1	0
12	12	cluster_2	1	0	1	0	1	0	1	0
13	13	cluster_2	1	0	1	0	1	0	1	0
14	14	cluster_1	1	0	1	0	1	0	1	0
15	15	cluster_1	1	0	1	0	1	0	1	0
16	16	cluster_2	1	0	1	0	1	0	1	0
17	17	cluster_4	1	0	1	0	1	0	1	0
18	18	cluster_2	1	0	1	0	1	0	1	0

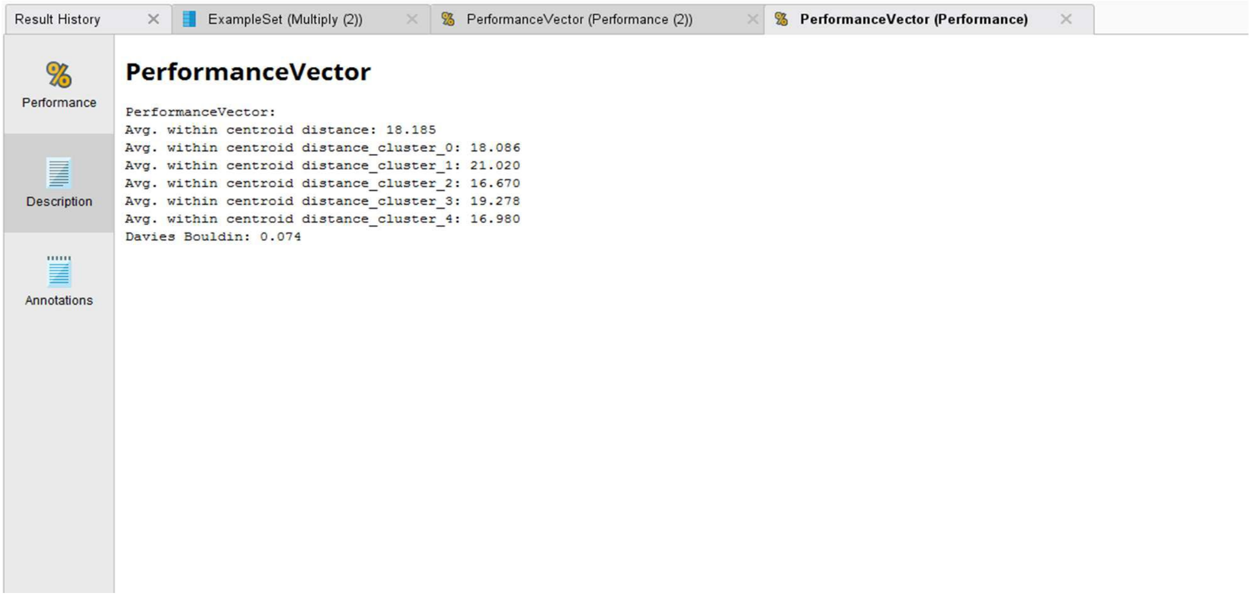
5. **Performance operators:** Both of these operators evaluate the output and performance of the clustering methods (k-means and k-medoids)

6.2.2. Result analysis

After running the process in the previous part, we can see the result of the two performance operators (one for each clustering method) below. The **performance** tab is related to **k-means,** and the **performance (2)** tab is inferred from the **k-medoids** algorithm.

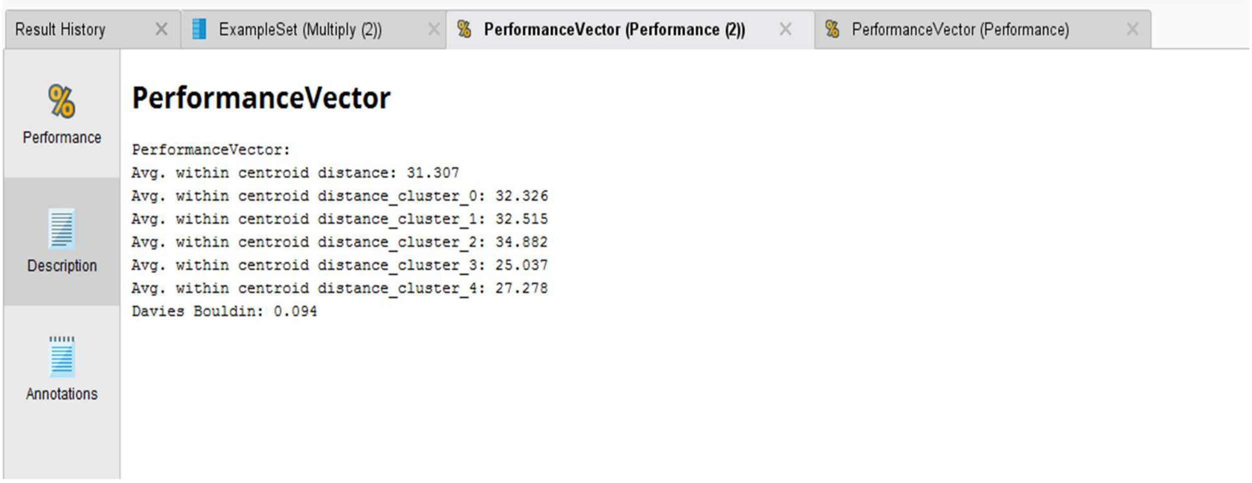
As we can see, there are two performance metrics used to evaluate the results, which are:

1. **Average inside centroid distance**: In this metric, the distance of each sample to the element is specified as the intra-cluster similarity. The average of these values is the intra-cluster similarity or centroid, and the lower this value, the greater and more appropriate the intra-cluster similarity.
2. **Davis Bouldin (DB)**: Using a combination of the two previous criteria (Inside clusters and between clusters) based on a specific formula, a criterion called Davies Bouldin is obtained; the lower the criterion, the more appropriate it is.


Result History
ExampleSet (Multiply (2))
PerformanceVector (Performance (2))
PerformanceVector (Performance)
Performance
Description
Annotations

PerformanceVector

```
PerformanceVector:
Avg. within centroid distance: 18.185
Avg. within centroid distance_cluster_0: 18.086
Avg. within centroid distance_cluster_1: 21.020
Avg. within centroid distance_cluster_2: 16.670
Avg. within centroid distance_cluster_3: 19.278
Avg. within centroid distance_cluster_4: 16.980
Davies Bouldin: 0.074
```


Result History
ExampleSet (Multiply (2))
PerformanceVector (Performance (2))
PerformanceVector (Performance)
Performance
Description
Annotations

PerformanceVector

```
PerformanceVector:
Avg. within centroid distance: 31.307
Avg. within centroid distance_cluster_0: 32.326
Avg. within centroid distance_cluster_1: 32.515
Avg. within centroid distance_cluster_2: 34.882
Avg. within centroid distance_cluster_3: 25.037
Avg. within centroid distance_cluster_4: 27.278
Davies Bouldin: 0.094
```

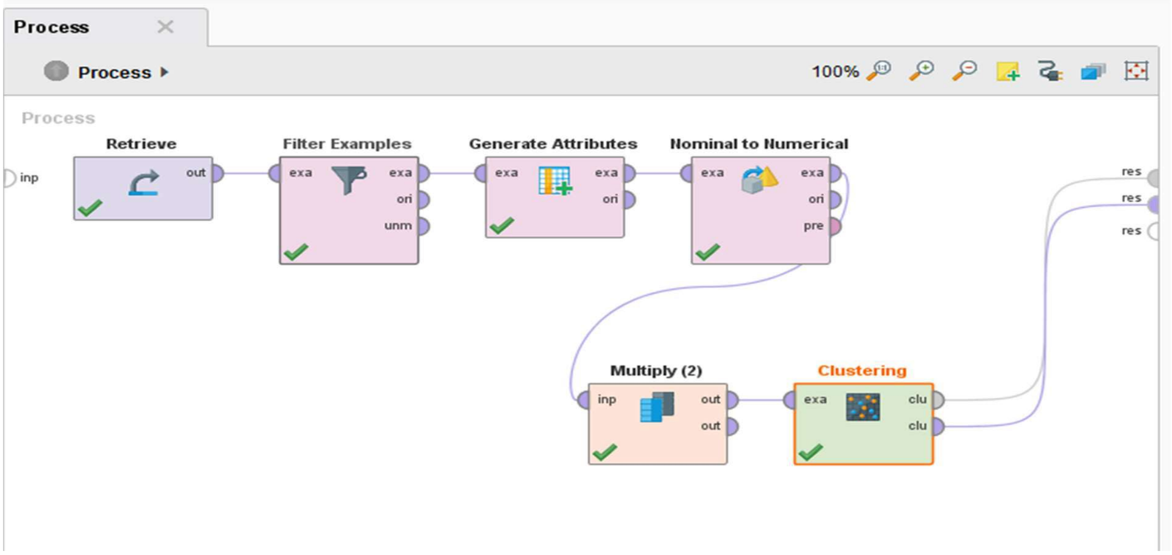
6.2.3. Conclusion

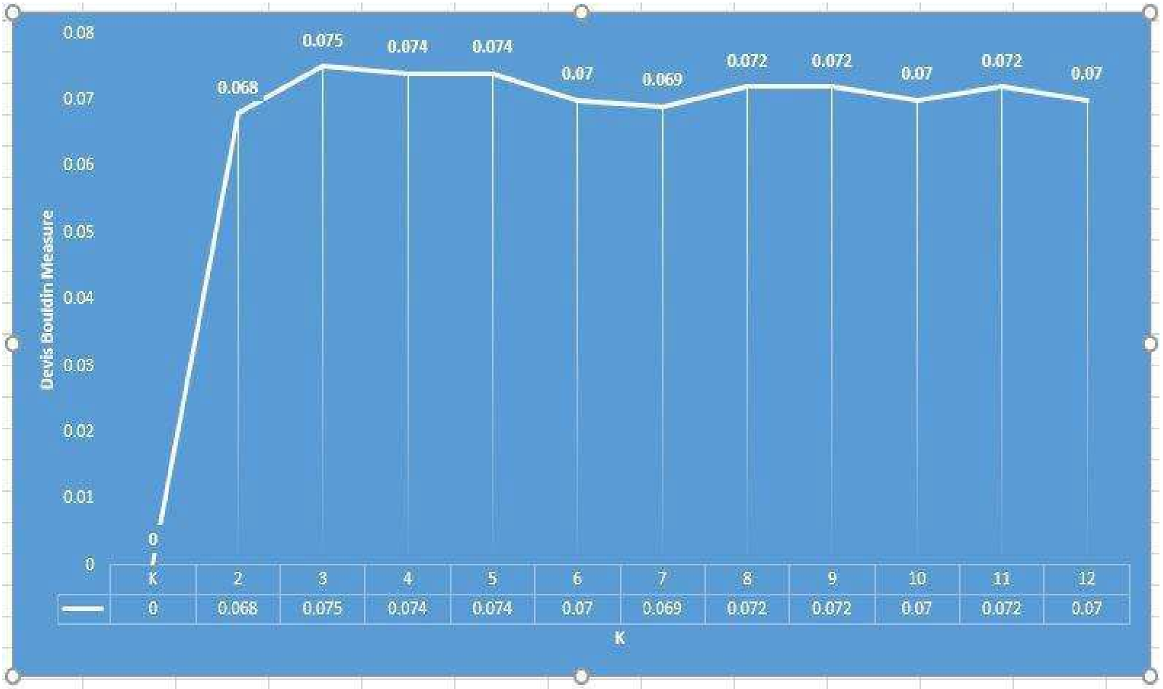
With the criteria discussed above, the following result is inferred. We compared the clustering algorithms used in this process, and we concluded that the **K-means** has a better intra-cluster similarity measure than the K-medoids algorithm. In the K-medoids algorithm, the values are higher than those of the K-means algorithm, so the intra-cluster similarity in the K-means algorithm has a better performance than the other algorithm. **Based on the Davis-Bouldin** criterion, the K-means algorithm also has a higher execution speed, so the performance of the K-means clustering algorithm and also the K-means algorithm is more suitable than the other algorithms on this set of data. We can see the result in the image below.

	K-means ✓	K-medoid
Avg. within centroid distance	**18.185**	**31.307**
Avg. within centroid distance_cluster_0	**18.086**	**32.326**
Avg. within centroid distance_cluster_1	**21.020**	**32.515**
Avg. within centroid distance_cluster_2	**16.670**	**34.882**
Avg. within centroid distance_cluster_3	**19.278**	**25.037**
Avg. within centroid distance_cluster_4	**16.980**	**27.278**
Davies Bouldin	**0.074**	**0.094**

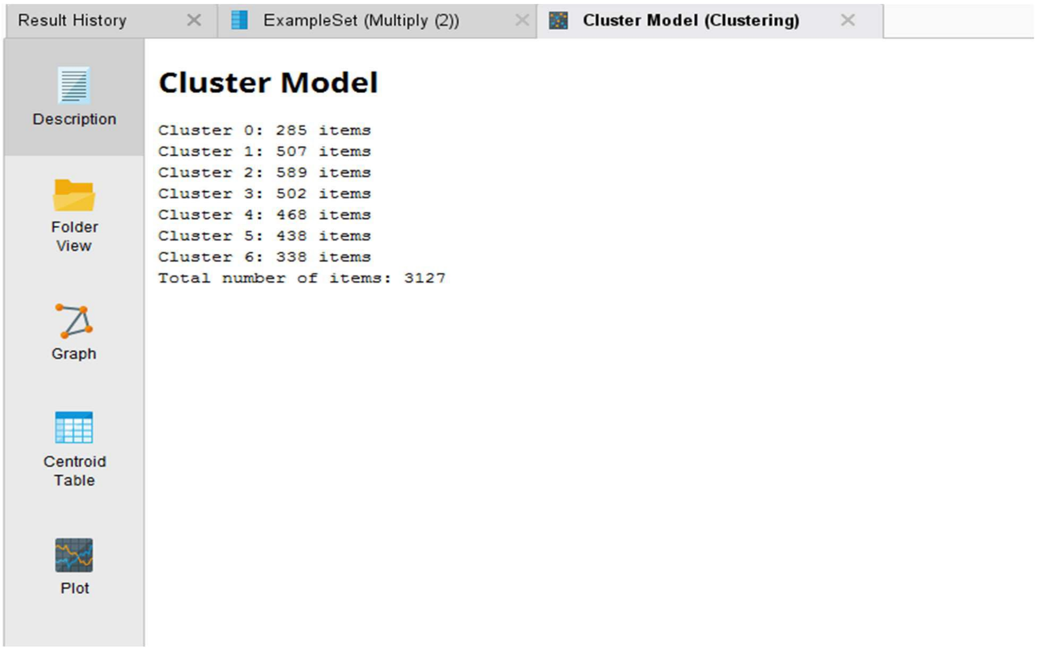
6.2.4. Cluster the dataset using the selected algorithm

Based on the operators specified in the image, we perform data preprocessing steps to arrive at the **clustering** operator. In this operator, we must choose a value for the parameter ***K,*** which represents the number of clusters. By using the Davis Bouldin chart, we can observe ups and downs as the number of K increases. As we can see in the second image below (the blue chart), the values of 6 and 7 are the most appropriate for our algorithm. So, we chose 7.





After executing the processes, the following outputs are displayed:



Cluster Model

```
Cluster 0: 285 items
Cluster 1: 507 items
Cluster 2: 589 items
Cluster 3: 502 items
Cluster 4: 468 items
Cluster 5: 438 items
Cluster 6: 338 items
Total number of items: 3127
```

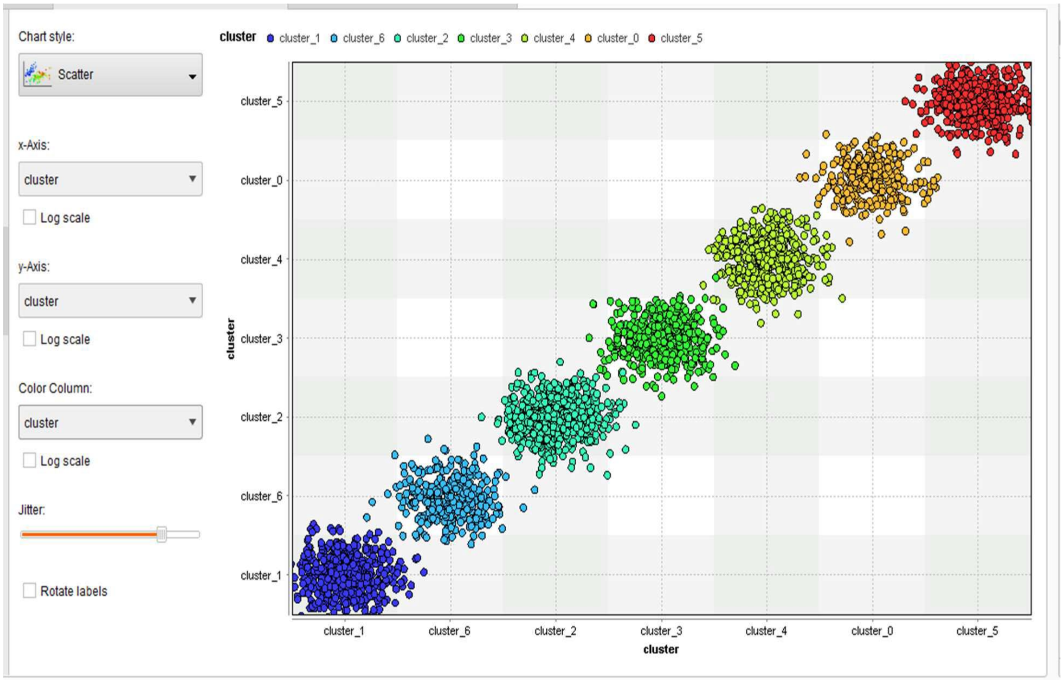


Chart style: Scatter

x-Axis: cluster

☐ Log scale

y-Axis: cluster

☐ Log scale

Color Column: cluster

☐ Log scale

Jitter:

☐ Rotate labels

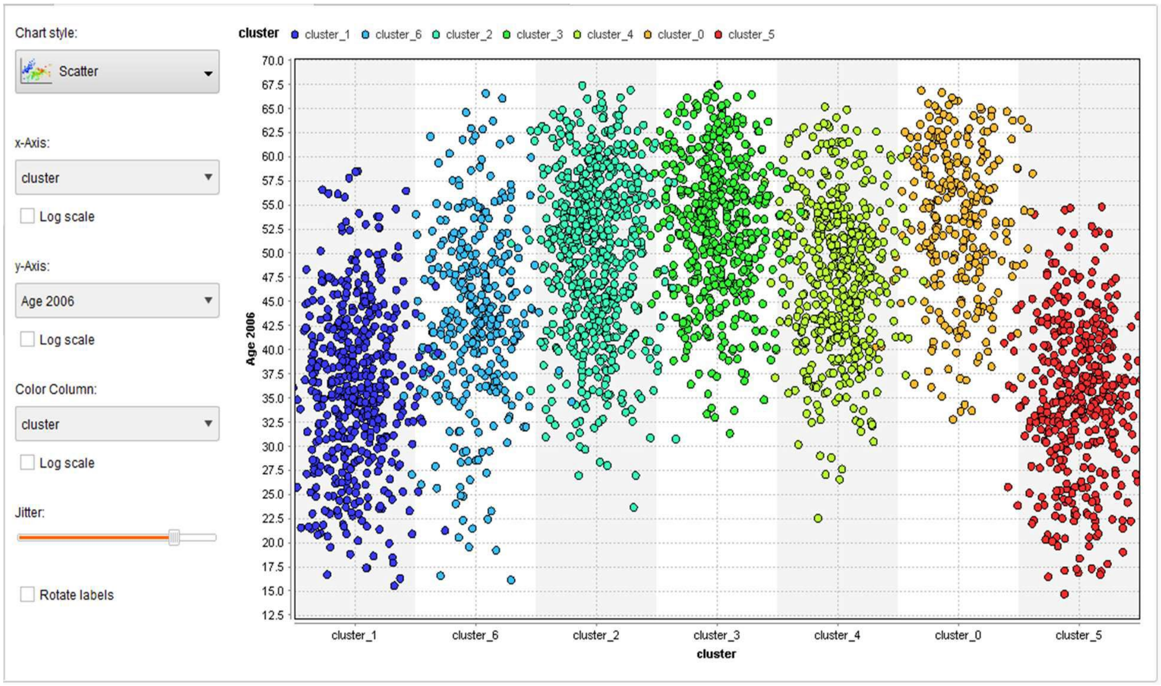


Chart style: Scatter

x-Axis: cluster

☐ Log scale

y-Axis: Age 2006

☐ Log scale

Color Column: cluster

☐ Log scale

Jitter:

☐ Rotate labels

In the image above, we see that the **age in the third cluster** is higher than 32, and in the **fifth cluster**, the age of the **samples is lower**. Now, according to the images below, we see that in the third cluster, which had a higher average age than the fifth cluster, it had **higher blood pressure** and **body mass index**. There are different analyses for each of the clusters according to different criteria.

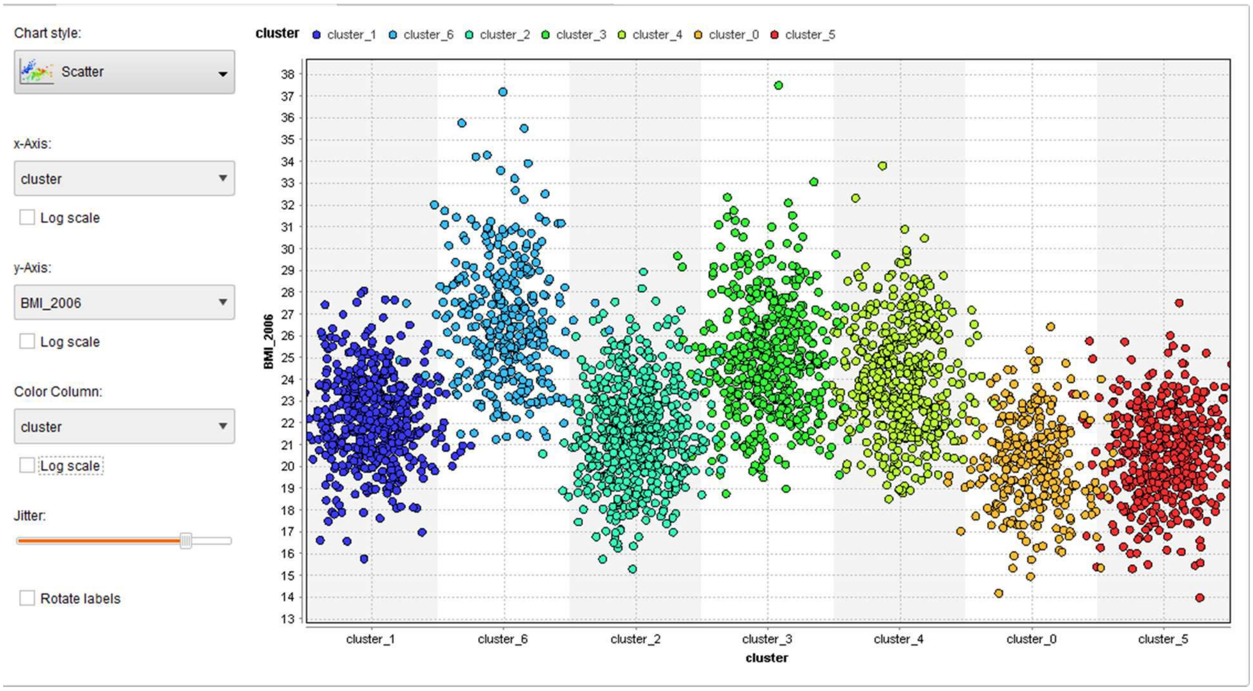


Chart style: Scatter

x-Axis: cluster

□ Log scale

y-Axis: BMI_2006

□ Log scale

Color Column: cluster

□ Log scale

Jitter:

□ Rotate labels

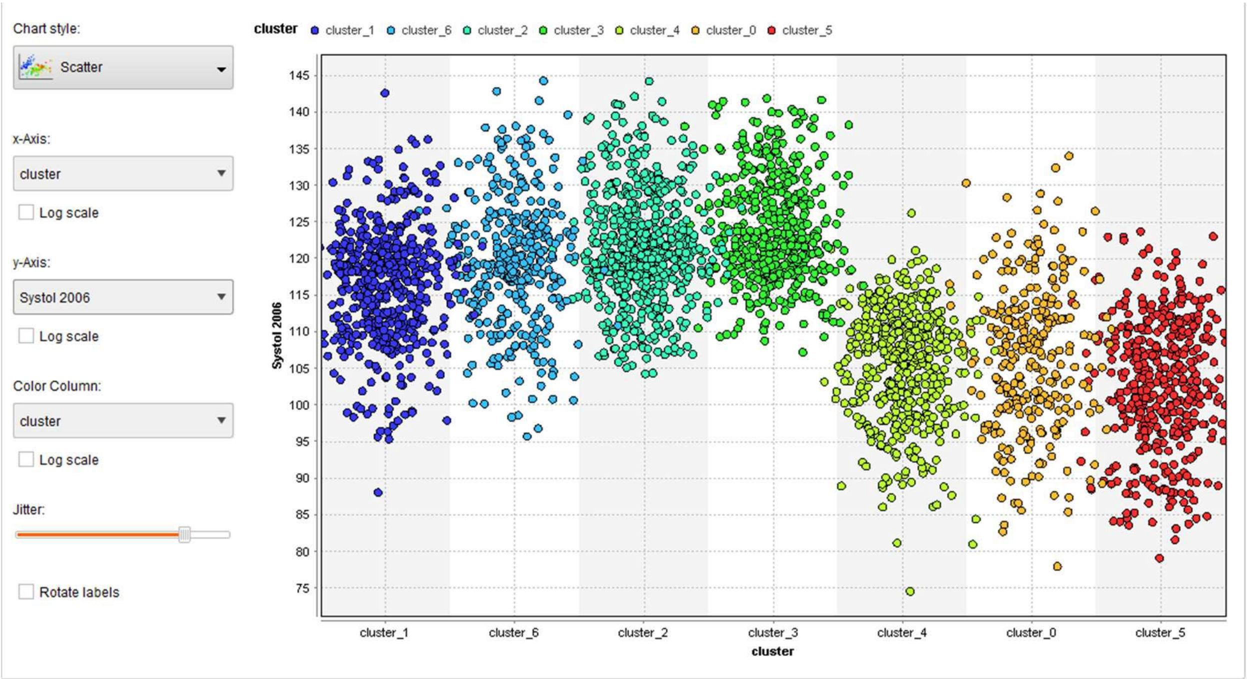


Chart style: Scatter

x-Axis: cluster

□ Log scale

y-Axis: Systol 2006

□ Log scale

Color Column: cluster

□ Log scale

Jitter:

□ Rotate labels

Section 7: Research Conclusion

Diagnosing hypertension is an important step in the medical field. In this report, we first explained this information by having a database of people's body mass index and blood pressure history, and then we reached some results by using **classification** and **clustering** techniques. With the help of a **decision tree**, we achieved a **decision support system**. This system helps doctors use this system in addition to their knowledge and expertise and ultimately make a more accurate diagnosis of the disease. Using clustering algorithms, samples that are more similar to each other were collected in several clusters. Using the similarities of the samples in each cluster, separate treatment measures can be taken.