

Retrieving users results in :

The screenshot shows the KNIME interface with two main panes. The left pane is a 'Data View' showing a table titled 'ExampleSet (11 examples, 0 special attributes, 15 regular attributes)'. The table has columns: 'irs_Per...', 'Preferred_B...', 'Preferred_S...', 'Preferred_E...', 'Read_News', 'Online_Sh... ', 'Online_Ga...', 'Facebook', 'Twitter', and 'Other_Soci...'. The right pane is a 'Repository View' showing a tree structure of local and cloud repositories.

irs_Per...	Preferred_B...	Preferred_S...	Preferred_E...	Read_News	Online_Sh...	Online_Ga...	Facebook	Twitter	Other_Soci...
Firefox	Google	Yahoo	Y	N	N	Y	N	?	
Chrome	Google	Hotmail	Y	N	N	Y	N	?	
Firefox	Yahoo	Yahoo	Y	Y	?	Y	N	?	
Firefox	Google	Hotmail	N	Y	N	N	Y	?	
Internet Exp...	Bing	Hotmail	Y	Y	N	Y	N	?	
Internet Exp...	Google	Yahoo	Y	N	Y	N	N	?	
Firefox	Google	Yahoo	?	Y	?	Y	Y	LinkedIn	
Internet Exp...	Yahoo	Yahoo	Y	?	?	Y	Y	LinkedIn	
Firefox	Google	Gmail	N	Y	N	N	N	?	
Safari	Yahoo	Yahoo	Y	?	Y	Y	N	MySpace	
Chrome	Google	Gmail	Y	N	N	Y	N	Google+	

Repository

- Samples
- DB
- Local Repository (Nadia)
 - data (Nadia)
 - users (Nadia - v1, 6/1/18 10:48 PM - 1 kB)
 - processes (Nadia)
 - DM4 (Nadia - v1, 6/1/18 10:42 PM - 825 bytes)
- Cloud Repository (disconnected)

we see missing values in online shopping, online Gaming and the social network column.

then missing values in online gaming and social network column would be filled by KNN.

The screenshot shows the KNIME interface with the following components:

- Top Bar:** Views: Design, Results, Auto Model, Find data, operators...etc, All Studio.
- Result History:** Shows two tabs: ExampleSet (Impute Missing Values) and ExampleSet (/Local Repository/data/users).
- Data View:** Displays a table titled "ExampleSet (11 examples, 0 special attributes, 15 regular attributes)". The columns are: irs_Per..., Preferred_B..., Preferred_S..., Preferred_E..., Read_News, Online_Sh..., Online_Ga..., Facebook, Twitter, Other_Soci.... The rows show various combinations of browser (Firefox, Chrome, Internet Exp...) and search engines (Google, Bing, Yahoo, Hotmail, Gmail). Some cells contain Y, N, or ?.
- Repository View:** Shows the local repository structure:
 - Samples
 - DB
 - Local Repository (Nadia)
 - data (Nadia)
 - users (Nadia - v1, 6/1/18 10:48 PM - 1 kB)
 - processes (Nadia)
 - DM4 (Nadia - v1, 6/1/18 10:42 PM - 825 bytes)
 - Cloud Repository (disconnected)
- Left Sidebar:** Contains icons for Data, Statistics, Charts, Advanced Charts, and Annotations.

Filtering examples by Online_Shopping=. causes removing missing online shopping rows.

The screenshot shows the KNIME Analytics Platform interface. The top navigation bar includes icons for file operations, a play button, and a search bar labeled "Find data, operators...etc". The tabs "Design", "Results", and "Auto Model" are present, with "Results" being the active tab. The left sidebar contains icons for Data, Statistics, Charts, Advanced Charts, and Annotations, with "Data" currently selected. The main workspace displays two windows: "Result History" and "Repository".

Result History: This window shows a table titled "ExampleSet (9 examples, 0 special attributes, 15 regular attributes)". The columns are: hours_Per_Day, Preferred_Browser, Preferred_Social_Net, Preferred_Electronic_Mail, Read_News, Online_Shopping, Online_Gaming, Facebook, Twitter, and Other_Social_Net. The data consists of 9 rows, each representing a user profile. The last row is highlighted in yellow.

hours_Per_Day	Preferred_Browser	Preferred_Social_Net	Preferred_Electronic_Mail	Read_News	Online_Shopping	Online_Gaming	Facebook	Twitter	Other_Social_Net
Firefox	Google	Yahoo	Y	N	N	Y	N	Google+	
Chrome	Google	Hotmail	Y	N	N	Y	N	Google+	
Firefox	Yahoo	Yahoo	Y	Y	Y	Y	N	LinkedIn	
Firefox	Google	Hotmail	N	Y	N	N	Y	Google+	
Internet Exp...	Bing	Hotmail	Y	Y	N	Y	N	Google+	
Internet Exp...	Google	Yahoo	Y	N	Y	N	N	LinkedIn	
Firefox	Google	Yahoo	?	Y	Y	Y	Y	LinkedIn	
Firefox	Google	Gmail	N	Y	N	N	N	Google+	
Chrome	Google	Gmail	Y	N	N	Y	N	Google+	

Repository: This window shows the local repository structure. It includes sections for Samples, DB, Local Repository (Nadia), and Cloud Repository (disconnected). The Local Repository (Nadia) section contains a "data" folder which includes a "users" file (last modified 6/1/18 10:48 PM - 1 kB).

results data for Sampling 50% of data examples using sample operator in relative mode and sample ratio =0.5 are shown below :

The screenshot shows the KNIME Analytics Platform interface. The top navigation bar includes icons for file operations, a play button, and a search bar labeled "Find data, operators...etc". The tabs "Design", "Results", and "Auto Model" are present, with "Results" being the active tab. The main area displays two windows: "Result History" and "Repository".

Result History: This window shows a table titled "ExampleSet (Sample)" containing 4 examples. The columns are: "Irs_Per_...", "Preferred_B...", "Preferred_S...", "Preferred_E...", "Read_News", "Online_Sh... ", "Online_Ga... ", "Facebook", "Twitter", and "Other_Soci... ". The data is as follows:

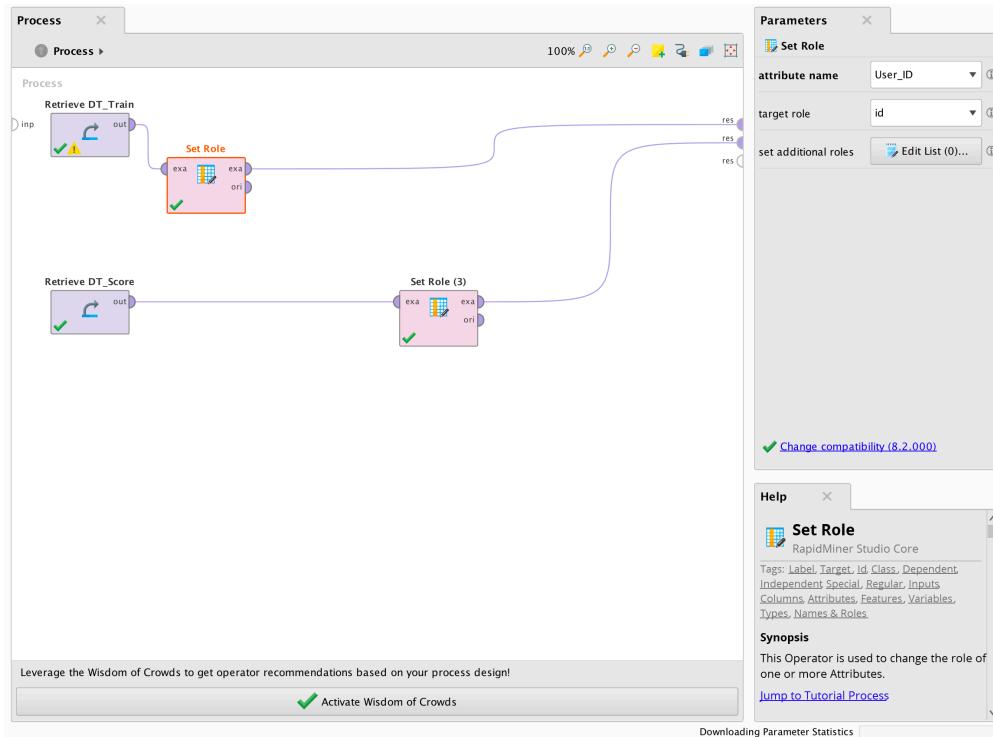
Irs_Per_...	Preferred_B...	Preferred_S...	Preferred_E...	Read_News	Online_Sh...	Online_Ga...	Facebook	Twitter	Other_Soci...
Firefox	Google	Yahoo	Y	N	N	Y	N	Google+	
Firefox	Google	Hotmail	N	Y	N	N	Y	Google+	
Internet Exp...	Bing	Hotmail	Y	Y	N	Y	N	Google+	
Chrome	Google	Gmail	Y	N	N	Y	N	Google+	

Repository: This window shows the local repository structure. It includes sections for "Samples", "DB", and "Local Repository (Nadia)". Under "Local Repository (Nadia)", there is a "data" folder containing a "users" file (6/1/18 10:48 PM - 1 kB) and a "processes" folder containing a "DM4" file (6/1/18 10:42 PM - 825 bytes). A "Cloud Repository" section is also listed as disconnected.

we had 9 items before and 50% sampling results in choose 4 of them.

Decision tree

adding set_Role to both DT_Train and DT_Test results in:



the user_ID column has shown as a special attribute in blue color. set_Role identifies user_ID as ID so decision tree doesn't determine this column.

ExampleSet (Set Role (3))																	ExampleSet (Set Role)																		
ExampleSet (661 examples, 1 special attribute, 10 regular attributes)																	ExampleSet (473 examples, 1 special attribute, 9 regular attributes)																		
Filter (661 / 661 examples): all																	Filter (473 / 473 examples): all																		
Row No.	User_ID	Gender	Age	Marital_Status	Website_Access	Browsed_Electronics	Bought_Electronics	Bought_Digital	Row No.	User_ID	Gender	Age	Marital_Status	Website_Access	Browsed_Electronics	Bought_Electronics	Bought_Digital	Row No.	User_ID	Gender	Age	Marital_Status	Website_Access	Browsed_Electronics	Bought_Electronics	Bought_Digital	Row No.	User_ID	Gender	Age	Marital_Status	Website_Access	Browsed_Electronics	Bought_Electronics	Bought_Digital
1	9552	M	61	M	Seldom	Yes	Yes	Yes	1	56031	M	57	S	Regular	Yes	Yes	Yes	8	48465	F	36	S	Frequent	Yes	No	Yes	14	77373	M	41	S	Seldom	Yes	Yes	Yes
2	6757	M	27	M	Regular	Yes	No	Yes	2	25913	F	51	M	Regular	Yes	Yes	No	9	19889	M	29	M	Regular	Yes	Yes	Yes	15	65685	M	61	M	Frequent	Yes	Yes	Yes
3	3599	F	29	M	Seldom	Yes	Yes	Yes	3	19396	M	41	M	Seldom	Yes	Yes	Yes	10	63570	M	47	S	Regular	Yes	Yes	Yes	16	72282	F	31	S	Seldom	Yes	No	Yes
4	6811	M	49	M	Seldom	Yes	Yes	Yes	4	93666	M	66	S	Regular	Yes	Yes	Yes	11	64466	M	68	M	Regular	Yes	Yes	Yes	17	7226	M	60	S	Seldom	Yes	Yes	Yes
5	4104	M	29	S	Seldom	Yes	Yes	Yes	5	72282	F	31	S	Seldom	Yes	No	Yes	12	76655	F	51	S	Seldom	Yes	No	No	18	4693	F	53	S	Regular	Yes	Yes	Yes
6	7226	M	60	S	Seldom	No	Yes	Yes	6	64466	M	68	M	Regular	Yes	Yes	Yes	13	48465	F	36	S	Frequent	Yes	No	Yes	19	7814	F	63	S	Seldom	Yes	No	No
7	7814	F	53	S	Regular	Yes	No	Yes	7	76655	F	51	S	Seldom	Yes	No	No	20	19889	M	29	M	Regular	Yes	Yes	Yes	21	4693	F	63	S	Seldom	Yes	No	Yes
8	4693	F	63	S	Seldom	Yes	Yes	Yes	8	48465	F	36	S	Frequent	Yes	No	Yes	22	19889	M	29	M	Regular	Yes	Yes	Yes	23	7226	M	46	M	Regular	Yes	Yes	Yes
9	7266	M	46	M	Regular	Yes	Yes	Yes	9	19889	M	29	M	Regular	Yes	Yes	Yes	24	63570	M	61	M	Frequent	Yes	No	Yes	25	5259	F	25	S	Seldom	Yes	Yes	Yes
10	5259	F	25	S	Seldom	Yes	Yes	Yes	10	63570	M	61	M	Frequent	Yes	Yes	No	26	64466	M	68	S	Regular	Yes	No	Yes	27	7417	F	40	M	Regular	Yes	No	Yes
11	7417	F	40	M	Regular	Yes	Yes	No	11	63239	M	47	S	Regular	Yes	No	Yes	28	76655	F	62	S	Regular	Yes	Yes	Yes	29	6795	M	29	S	Seldom	Yes	Yes	Yes
12	6795	M	29	S	Seldom	Yes	Yes	Yes	12	67603	F	47	S	Regular	Yes	Yes	Yes	30	77373	M	17	M	Seldom	Yes	No	Yes	31	8669	F	49	S	Seldom	Yes	Yes	Yes
13	8669	F	49	S	Seldom	Yes	Yes	Yes	13	65685	M	32	M	Regular	Yes	Yes	No	32	77373	M	41	M	Regular	Yes	No	Yes	33	9888	M	41	M	Regular	Yes	Yes	Yes
14	9888	M	41	M	Regular	Yes	Yes	Yes	14	77373	M	17	M	Seldom	Yes	No	Yes	34	77373	M	41	M	Regular	Yes	No	Yes	35	7226	M	60	S	Seldom	Yes	Yes	Yes

then we add another set_Role to identify eReader_Adoption as label so we can then classify data by decision tree.

The screenshot shows the KNIME environment with a process flow and a data table.

Process Flow:

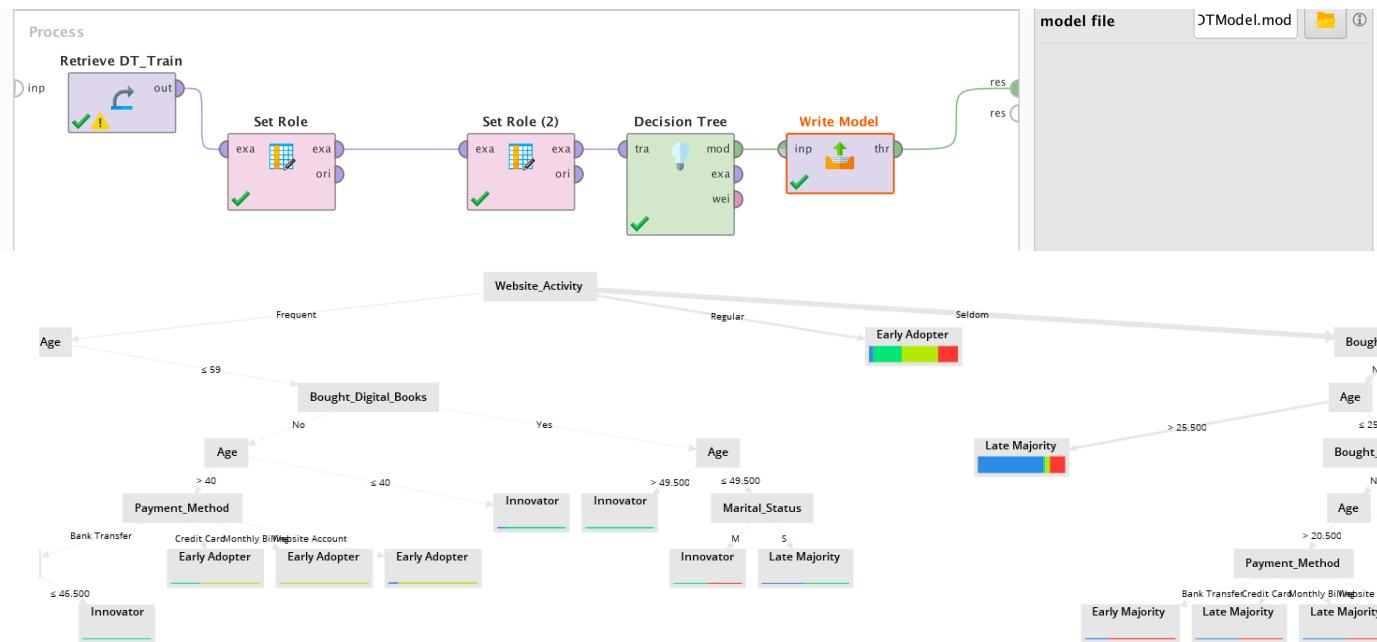
- A "Retrieve DT_Train" node (purple) has an "inp" port connected to the "Set Role" node (pink).
- The "Set Role" node has two outputs: one labeled "exa" and one labeled "ori".
- The "ori" output connects to a second "Set Role" node (pink).
- The second "Set Role" node also has two outputs: one labeled "exa" and one labeled "ori".
- The "ori" output from the second "Set Role" node connects to three separate output ports, each labeled "res".

Data Table:

The data table displays 661 examples across 10 columns:

Row No.	User_ID	eReader_A...	Gender	Age	Marital_Sta...	Website_Ac...	Browsed_El...	Bought_Ele...	...
1	9552	Late Majority	M	61	M	Seldom	Yes	Yes	...
2	6757	Innovator	M	27	M	Regular	Yes	No	...
3	3599	Early Adopter	F	29	M	Seldom	Yes	Yes	...
4	6811	Early Adopter	M	49	M	Seldom	Yes	Yes	...
5	4104	Late Majority	M	29	S	Seldom	Yes	Yes	...
6	7226	Early Majority	M	60	S	Seldom	No	Yes	...
7	7814	Early Adopter	F	53	S	Regular	Yes	No	...
8	4693	Late Majority	F	63	S	Seldom	Yes	Yes	...
9	7266	Innovator	M	46	M	Regular	Yes	Yes	...
10	5259	Early Majority	F	25	S	Seldom	Yes	Yes	...
11	7417	Innovator	F	40	M	Regular	Yes	Yes	...
12	6795	Early Adopter	M	29	S	Seldom	Yes	Yes	...
13	8669	Late Majority	F	49	S	Seldom	Yes	Yes	...
14	9888	Innovator	M	41	M	Regular	Yes	Yes	...
15	9181	Innovator	F	47	S	Regular	Yes	No	...
16	2792	Late Majority	F	43	M	Seldom	Yes	Yes	...
17	5174	Early Adopter	M	63	M	Seldom	Yes	Yes	...
18	8702	Early Adopter	F	37	M	Regular	No	Yes	...
19	4318	Early Majority	M	32	S	Seldom	Yes	Yes	...
20	6091	Late Majority	M	48	M	Seldom	No	No	...
21	8380	Early Adopter	F	57	M	Seldom	Yes	No	...
22	6892	Early Majority	M	36	M	Seldom	Yes	Yes	...
23	7976	Late Majority	F	55	S	Seldom	Yes	No	...

next step we should train a model using Decision tree and save it. Results shown below.



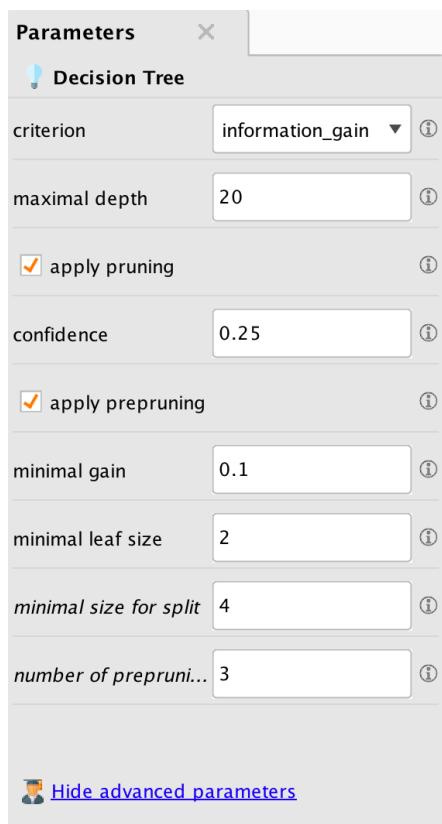
Tree

```

Website_Activity = Frequent
| Age > 59
| | Age > 63.500: Innovator {Late Majority=1, Innovator=2, Early Adopter=0, Early Majority=0}
| | Age ≤ 63.500
| | | Browsed_Electronics_12Mo = No: Early Majority {Late Majority=0, Innovator=1, Early Adopter=0, Early Majority=0}
| | | Browsed_Electronics_12Mo = Yes
| | | | Bought_Electronics_12Mo = No
| | | | | Gender = F
| | | | | | Age > 61.500: Early Majority {Late Majority=0, Innovator=0, Early Adopter=0, Early Majority=0}
| | | | | | Age ≤ 61.500: Early Adopter {Late Majority=0, Innovator=0, Early Adopter=2, Early Majority=0}
| | | | | | Gender = M: Early Majority {Late Majority=0, Innovator=0, Early Adopter=0, Early Majority=2}
| | | | | | Bought_Electronics_12Mo = Yes: Early Adopter {Late Majority=0, Innovator=0, Early Adopter=2, Early Majority=0}
| | | Age ≤ 59
| | | | Bought_Digital_Books = No
| | | | | Age > 40
| | | | | | Payment_Method = Bank Transfer
| | | | | | | Age > 46.500
| | | | | | | | Gender = F: Early Adopter {Late Majority=0, Innovator=0, Early Adopter=2, Early Majority=0}
| | | | | | | | Gender = M: Innovator {Late Majority=0, Innovator=2, Early Adopter=1, Early Majority=0}
| | | | | | | | Age ≤ 46.500: Innovator {Late Majority=0, Innovator=2, Early Adopter=0, Early Majority=0}
| | | | | | | | Payment_Method = Credit Card: Early Adopter {Late Majority=0, Innovator=1, Early Adopter=2, Early Majority=0}
| | | | | | | | Payment_Method = Monthly Billing: Early Adopter {Late Majority=0, Innovator=0, Early Adopter=2, Early Majority=0}
| | | | | | | | Payment_Method = Website Account: Early Adopter {Late Majority=1, Innovator=0, Early Adopter=8, Early Majority=0}
| | | | | | | | Age ≤ 40: Innovator {Late Majority=1, Innovator=7, Early Adopter=0, Early Majority=0}
| | | | | | | | Bought_Digital_Books = Yes
| | | | | | | | | Age > 49.500: Innovator {Late Majority=0, Innovator=6, Early Adopter=0, Early Majority=0}
| | | | | | | | | Age ≤ 49.500
| | | | | | | | | | Marital_Status = M: Innovator {Late Majority=0, Innovator=1, Early Adopter=0, Early Majority=1}
| | | | | | | | | | Marital_Status = S: Late Majority {Late Majority=1, Innovator=1, Early Adopter=0, Early Majority=0}
Website_Activity = Regular: Early Adopter {Late Majority=9, Innovator=58, Early Adopter=75, Early Majority=41}
Website_Activity = Seldom
| | Bought_Digital_Books = No
| | | Age > 25.500: Late Majority {Late Majority=142, Innovator=4, Early Adopter=10, Early Majority=33}
| | | Age ≤ 25.500
| | | | Bought_Electronics_12Mo = No
| | | | | Age > 20.500
| | | | | | Payment_Method = Bank Transfer: Early Majority {Late Majority=1, Innovator=0, Early Adopter=0, Early Majority=0}
| | | | | | Payment_Method = Credit Card: Late Majority {Late Majority=1, Innovator=0, Early Adopter=0, Early Majority=0}

```

decision tree has several parameters such as max-depth, minimum_leaf_size,



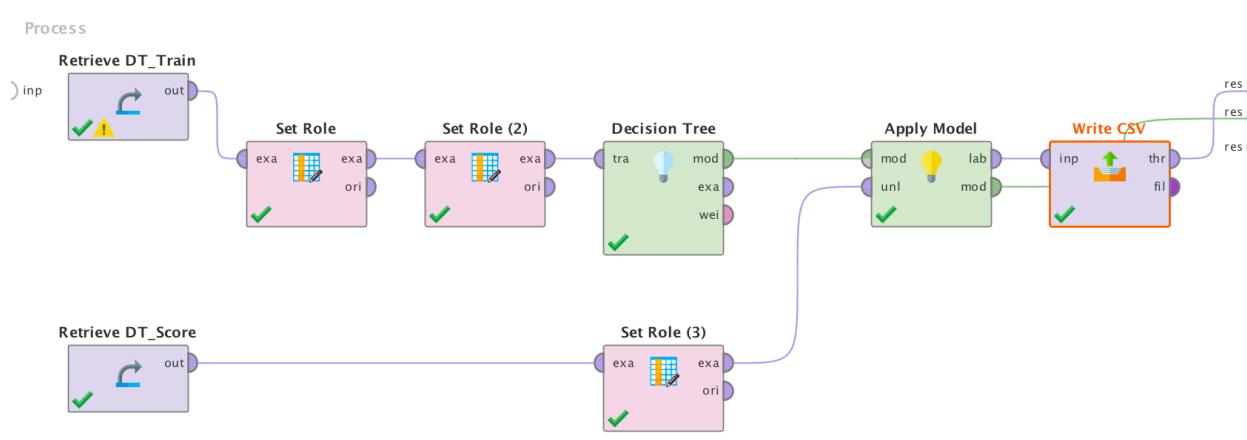
criterion is the method which decision tree decides about nodes arrangement based on it. we can try different methods like information_gain, gain_ratio,

maximal depth is used to restrict the depth of the decision tree.

apply pruning: The decision tree model can be pruned after generation. If checked, some branches are replaced by leaves according to the confidence parameter

confidence: This parameter specifies the confidence level used for the pessimistic error calculation of pruning.

then by using apply_model operation eReader_Adoption is predicted for test samples.



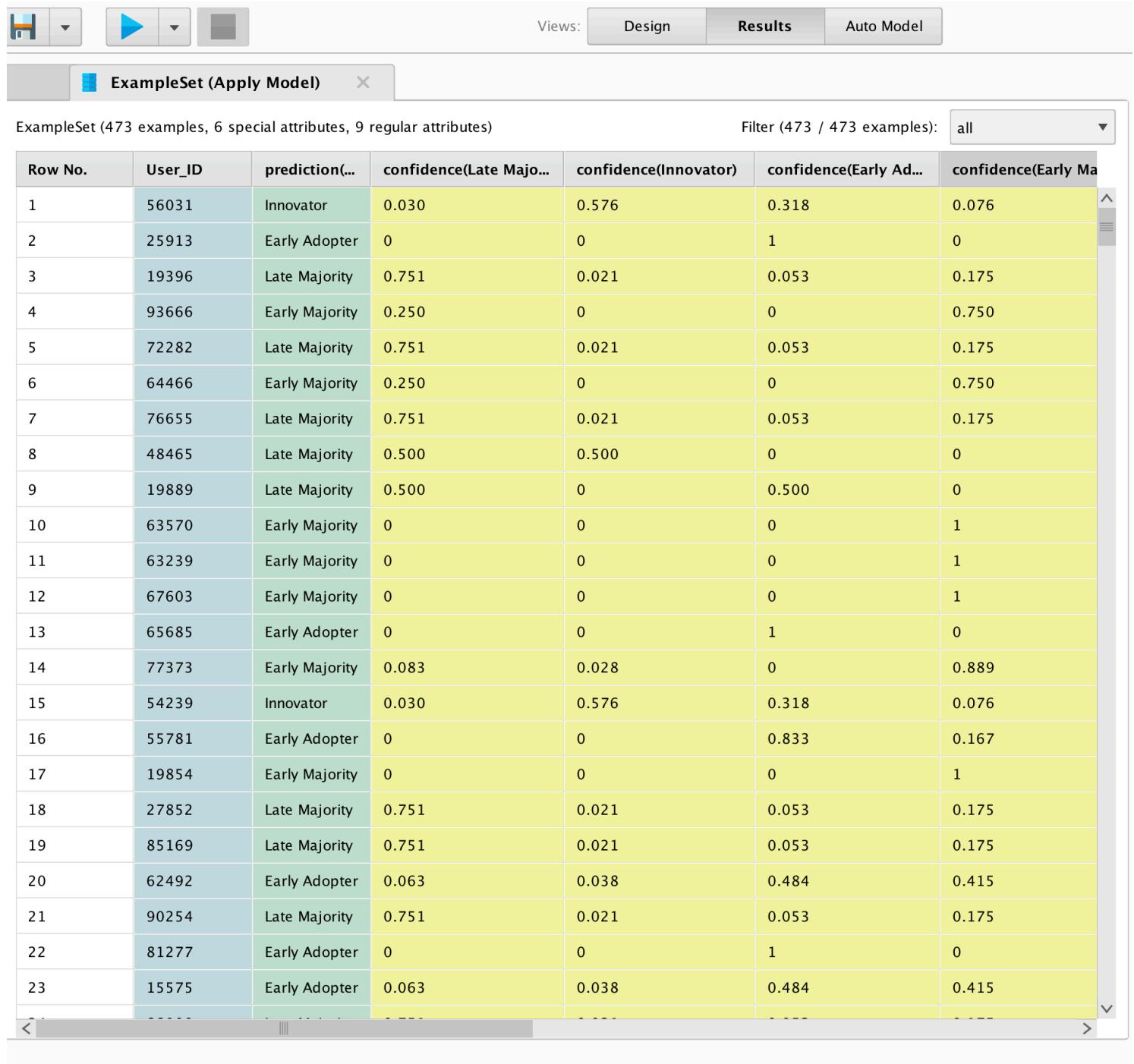
it calculates confidence for each attribute and then predicts eReader_Adoption with attribute which has the most confidence. for example confidence of early majority in 10th row is 1 while other values are zero so we could be sure eReader_Adoption is early_majority there.



The screenshot shows the RapidMiner interface with the 'ExampleSet (Apply Model)' tab selected. The data grid displays 473 examples across 9 attributes. The columns are labeled: Row No., User_ID, prediction(...), confidence(Late Maj...), confidence(Innovat...), confidence(Early Ad...), and confidence(Early Majori...). The data shows various adoption types (Early Adopter, Late Majority, Innovator) and their associated confidence values.

Row No.	User_ID	prediction(...)	confidence(Late Maj...)	confidence(Innovat...)	confidence(Early Ad...)	confidence(Early Majori...)
1	56031	Early Adopter	0.049	0.317	0.410	0.224
2	25913	Early Adopter	0.049	0.317	0.410	0.224
3	19396	Late Majority	0.751	0.021	0.053	0.175
4	93666	Early Adopter	0.049	0.317	0.410	0.224
5	72282	Late Majority	0.751	0.021	0.053	0.175
6	64466	Early Adopter	0.049	0.317	0.410	0.224
7	76655	Late Majority	0.751	0.021	0.053	0.175
8	48465	Innovator	0.125	0.875	0	0
9	19889	Early Adopter	0.049	0.317	0.410	0.224
10	63570	Early Majority	0	0	0	1
11	63239	Early Adopter	0.049	0.317	0.410	0.224
12	67603	Early Adopter	0.049	0.317	0.410	0.224
13	65685	Early Adopter	0.049	0.317	0.410	0.224
14	77373	Early Majority	0	0	0	1
15	54239	Early Adopter	0.049	0.317	0.410	0.224
16	55781	Early Adopter	0.049	0.317	0.410	0.224
17	19854	Early Adopter	0.049	0.317	0.410	0.224
18	27852	Late Majority	0.751	0.021	0.053	0.175
19	85169	Late Majority	0.751	0.021	0.053	0.175
20	62492	Early Adopter	0.070	0.060	0.508	0.362

if we choose gain_ratio for criterion it will separate classes better so we'll get better results as we see in the following :

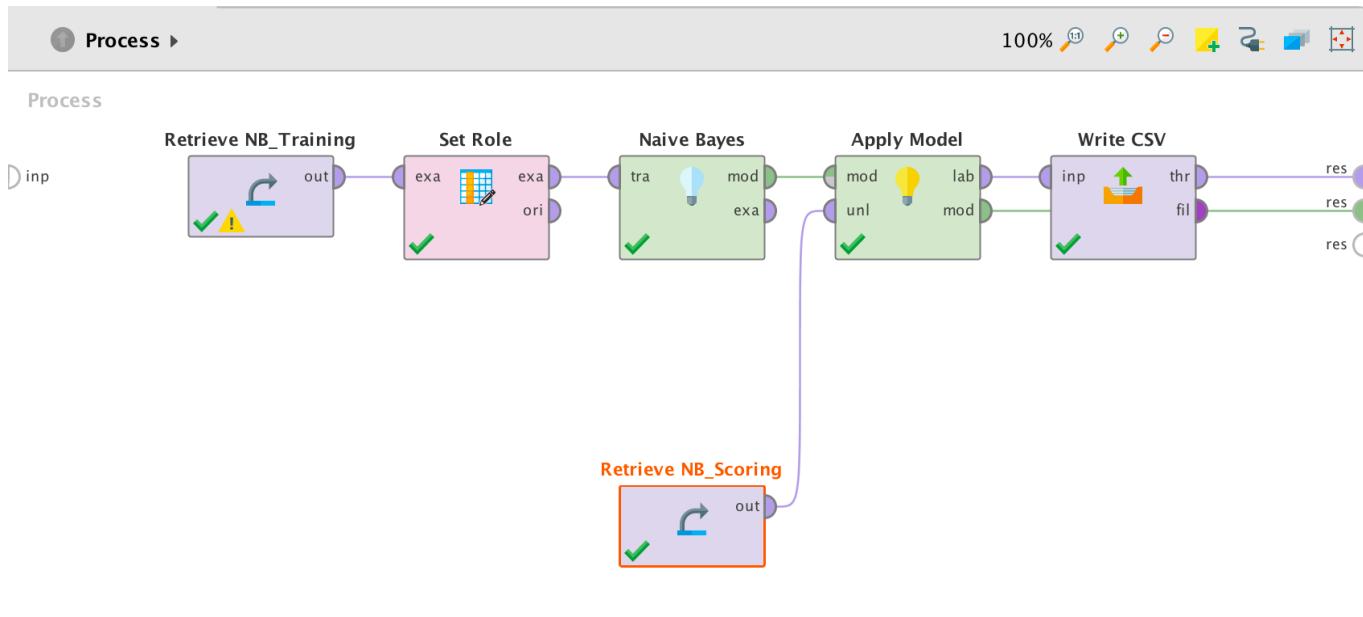


The screenshot shows the KNIME environment with the 'ExampleSet (Apply Model)' node open. The table displays 473 examples across 9 columns. The columns are: Row No., User_ID, prediction(...), confidence(Late Majo...), confidence(Innovator), confidence(Early Ado...), and confidence(Early Ma...). The data shows various user IDs and their predicted categories along with their corresponding confidence scores.

Row No.	User_ID	prediction(...)	confidence(Late Majo...)	confidence(Innovator)	confidence(Early Ado...)	confidence(Early Ma...)
1	56031	Innovator	0.030	0.576	0.318	0.076
2	25913	Early Adopter	0	0	1	0
3	19396	Late Majority	0.751	0.021	0.053	0.175
4	93666	Early Majority	0.250	0	0	0.750
5	72282	Late Majority	0.751	0.021	0.053	0.175
6	64466	Early Majority	0.250	0	0	0.750
7	76655	Late Majority	0.751	0.021	0.053	0.175
8	48465	Late Majority	0.500	0.500	0	0
9	19889	Late Majority	0.500	0	0.500	0
10	63570	Early Majority	0	0	0	1
11	63239	Early Majority	0	0	0	1
12	67603	Early Majority	0	0	0	1
13	65685	Early Adopter	0	0	1	0
14	77373	Early Majority	0.083	0.028	0	0.889
15	54239	Innovator	0.030	0.576	0.318	0.076
16	55781	Early Adopter	0	0	0.833	0.167
17	19854	Early Majority	0	0	0	1
18	27852	Late Majority	0.751	0.021	0.053	0.175
19	85169	Late Majority	0.751	0.021	0.053	0.175
20	62492	Early Adopter	0.063	0.038	0.484	0.415
21	90254	Late Majority	0.751	0.021	0.053	0.175
22	81277	Early Adopter	0	0	1	0
23	15575	Early Adopter	0.063	0.038	0.484	0.415

Naive Bayes

we applied this process to NB dataset :



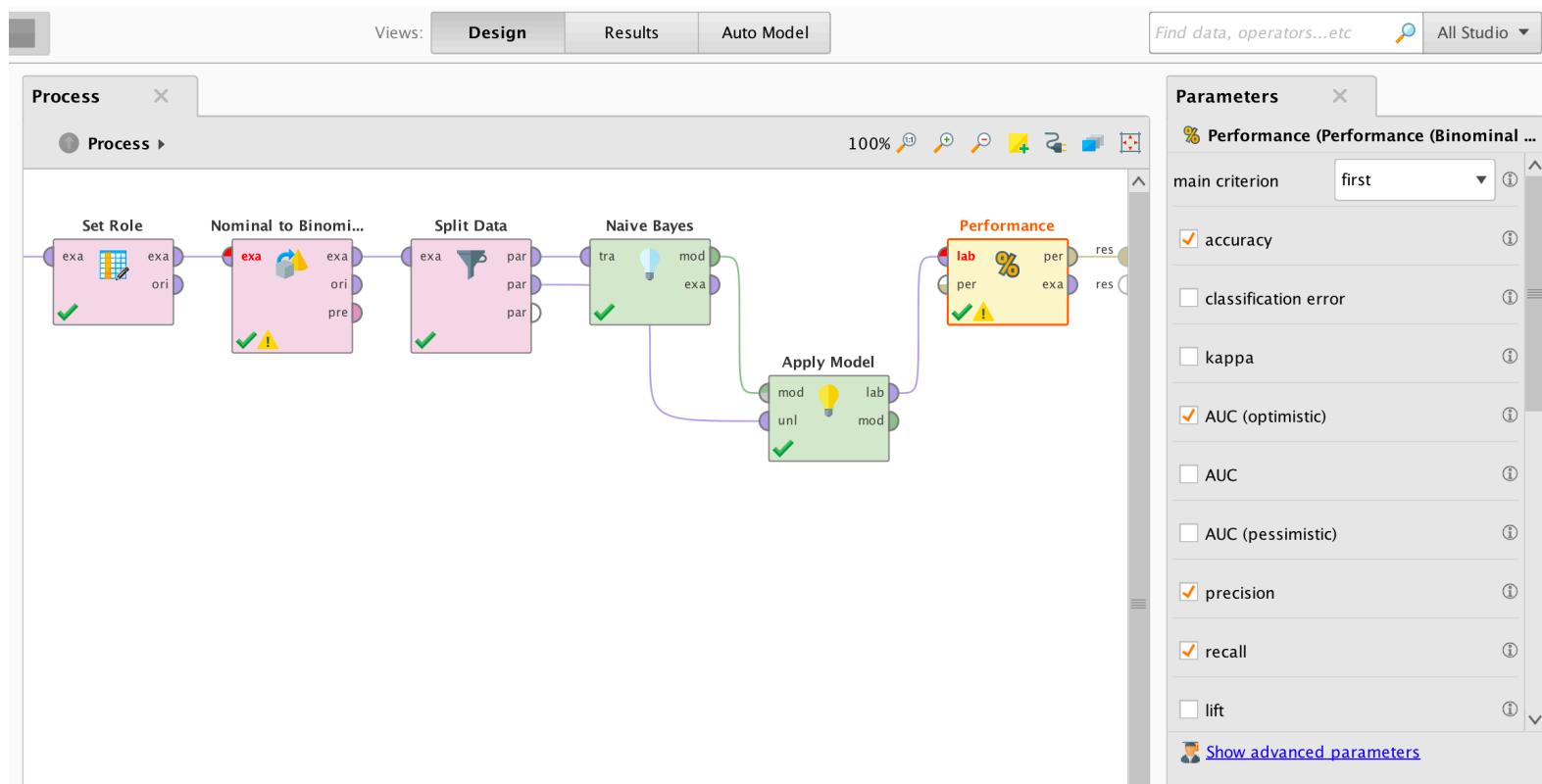
set Role determines 2nd_Heart_Attack as label and it will predicted for NB_Scoring dataset.

predicted results by NaiveBayes are shown below:

The screenshot shows the KNIME interface with the 'SimpleDistribution (Naive Bayes)' node selected. The title bar displays 'SimpleDistribution (Naive Bayes)' and 'ExampleSet (/)'. The main content area is titled 'SimpleDistribution' and describes it as a 'Distribution model for label attribute 2nd_Heart_Attack'. It lists two classes: 'Class Yes (0.493)' with '7 distributions' and 'Class No (0.507)' with '7 distributions'.

ExampleSet (690 examples, 3 special attributes, 7 regular attributes)									Filter (690 / 690 examples): all	
Row No.	prediction(...)	confidence(Yes)	confidence(No)	Age	Marital_Status	Gender	Weight_Cat...	Cholester...		
1	No	0.000	1.000	61	0	1	1	139		
2	Yes	0.963	0.037	55	2	1	2	163		
3	No	0.163	0.837	53	1	1	1	172		
4	Yes	0.999	0.001	58	1	1	2	206		
5	No	0.139	0.861	62	2	1	1	148		
6	No	0.020	0.980	70	1	0	0	172		
7	No	0.000	1.000	52	1	0	0	171		
8	No	0.109	0.891	50	1	1	1	172		
9	Yes	0.954	0.046	67	2	1	1	172		
10	No	0.040	0.960	62	1	1	1	166		
11	Yes	1.000	0.000	66	2	1	2	220		
12	No	0.007	0.993	56	2	1	0	141		
13	Yes	1.000	0.000	77	2	1	2	181		
14	Yes	0.931	0.069	64	2	1	1	174		

we can use performance operator for classification results analysis. I have implemented it for the naive bayes algorithm on NB_training dataset. we should split it to train and test to measure accuracy of it so I used Split Data operator too.

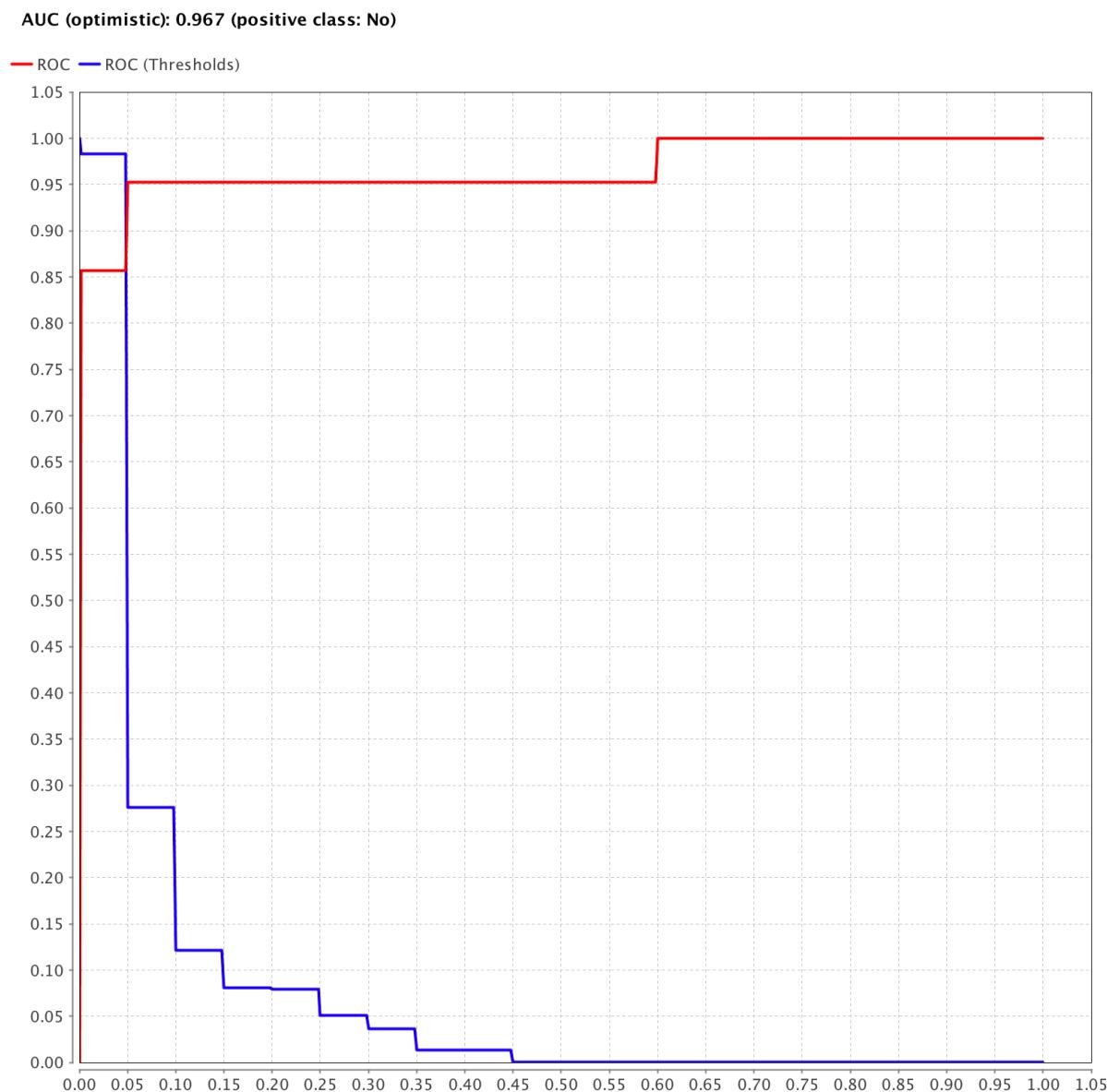


Results shown below :

accuracy: 92.68%

	true Yes	true No	class precision
pred. Yes	19	2	90.48%
pred. No	1	19	95.00%
class recall	95.00%	90.48%	

we can measure AUC too. It has been reported 0.967 :



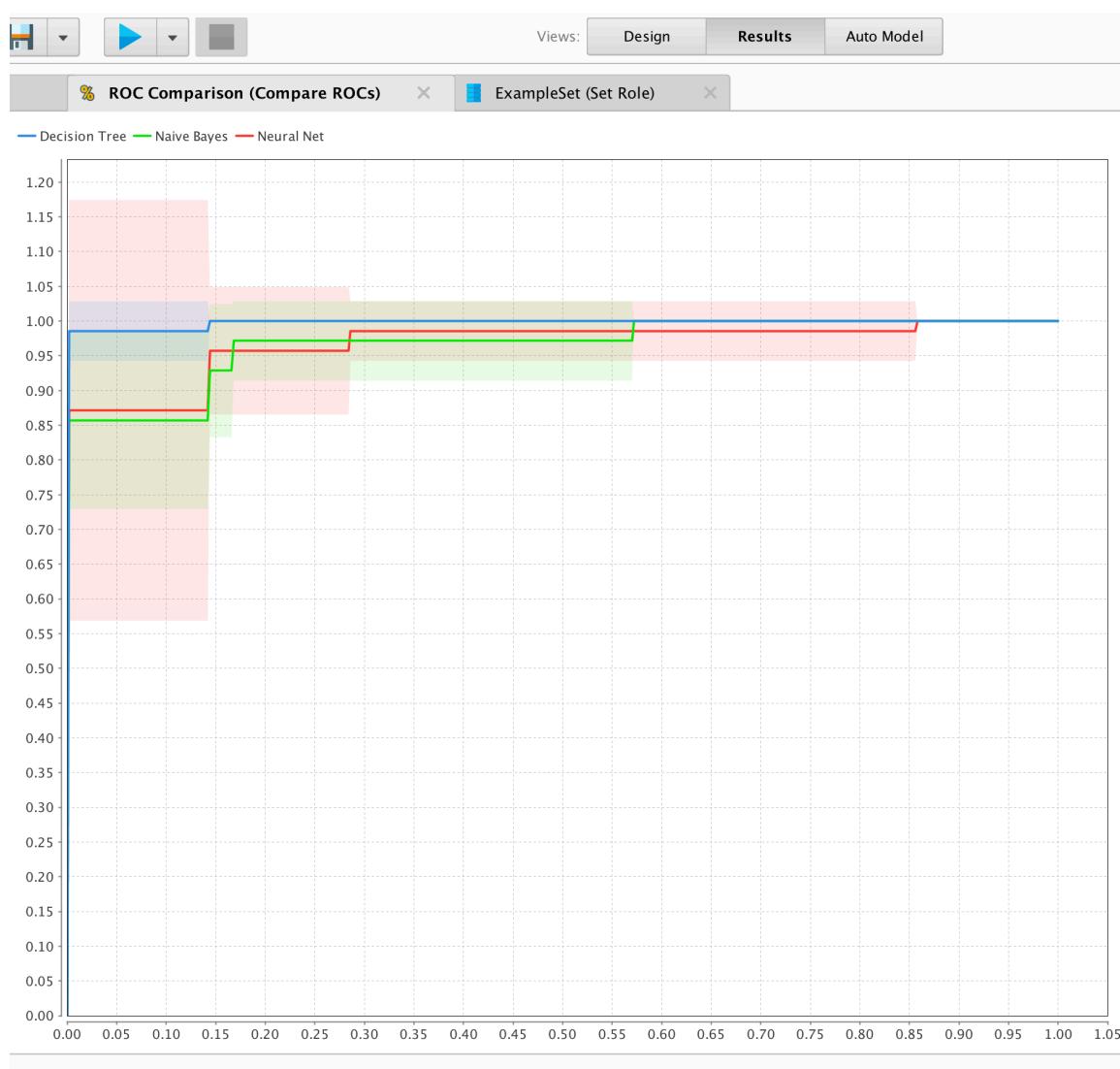
then I replaced naive bayes with decision tree and the accuracy is 95.12 which is more than Naive bayes accuracy (92.68):

accuracy: 95.12%

	true Yes	true No	class precision
pred. Yes	19	1	95.00%
pred. No	1	20	95.24%
class recall	95.00%	95.24%	

ROC

An ROC curve plots sensitivity or TPR(True Positive Rate) vs. FPR(False Positive Rate) at different classification thresholds. For a perfect classifier the ROC curve will go straight up the Y axis and then along the X axis. A classifier with no power will sit on the diagonal, whilst most classifiers fall somewhere in between.



we see that Decision Tree is better for this problem.

the Confidence column in Naive Bayes shows how sure predicted value is true. for example 0.9 for true shows that it's 90 percent true positive. The ROC curve plots TPR vs FPR at different confidences so

AUC measures the entire two-dimensional area underneath the entire ROC curve. One way of interpreting AUC is as the probability that the model ranks a random positive example more highly than a random negative example. more AUC represents a better classification method. for example in upper image Decision Tree has more AUC so it's a better method for classification.

Compare ROC operator has many parameters like number of folds, split ratio, sampling Type, . . . it uses k-fold cross validation and we can adjust k by number of folds parameter.

Several types of sampling can be used for building the subsets. for example Linear sampling simply divides the ExampleSet into partitions without changing the order of the examples but Shuffled sampling builds random subsets of the ExampleSet. in this problem shuffle sampling result is better.