SECP2753 – DATA MINING

SECTION 01

SEMESTER 2 2021/2022

GROUP PROJECT

DATASET B

Members:

1. **GROUP LEADER**: NAYLI NABIHAH JASNI (A20EC0105)
2. MIKHEL ADAM BIN MUHAMMAD EZRIN (A20EC0237)
3. MUHAMMAD DINIE HAZIM BIN AZALI (A20EC0084)
4. MADINA SURAYA BINTI ZHARIN (A20EC0203)

Lecturer's Name: Madam Rozilawati Binti Dollah @ Md. Zain

TABLE OF CONTENT

1.0 INTRODUCTION

In general, data mining is a method used to transform unstructured data into valuable information. In computer science, the practice of identifying interesting and practical patterns and relationships in vast amounts of data is known as data mining, often referred to as KDD or Knowledge Discovery in Databases (Clifton, 2022). Data mining is widely used in business, science research and government security (Clifton, 2022). There are a variety of techniques employed in data mining, including classification, clustering and association. We can transform the raw data into meaningful knowledge and information using a variety of methodologies, which may then be utilized to boost profits, lower expenses, strengthen customer relationships, lower risks, and many more.

For this group project, in a group of four students, we are required to use data mining tools to do some data preprocessing and data mining tasks based on the provided raw dataset. The purpose of this project is to ensure that the students can understand data mining by completing the assigned activities and to improve their ability to use data mining tools. There are two parts in this project. First part is we need to do data pre-processing on the raw dataset that has been provided by using data mining tools. Next, we need to do at least three data mining tasks consisting of classification, clustering or association.

In this project, we are given a raw dataset in a document file. The raw data contained a set of data related to the medical industry. The data provides disease names, disease number series, and a disease-related abstract. We are assigned to perform the data mining tasks using this dataset. For our project, our group received Dataset B. A total of 23 raw data concerning medical terminology were included in the collection.

2.0 DATA MINING TOOLS

In this group project, we were assigned to use RapidMiner as our data mining tool. RapidMiner is a robust data mining tool that supports model deployment, model operations, and data mining. All the data preparation and machine learning capabilities required to make a significant effect throughout your organization are provided by this end-to-end data science platform.

Figure 1: RapidMiner Logo

3.0 DATA PREPROCESSING

1. Convert dataset given in words into Excel file. This is because RAPIDMINER only accepts csv and excel files.

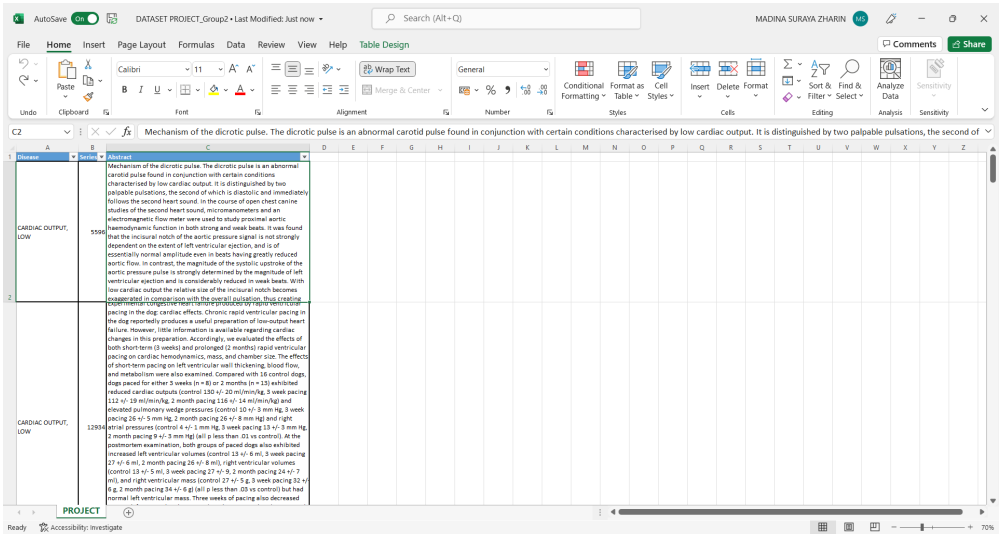


Figure 2: Dataset in Excel file

2. Open RAPIDMINER and start doing text processing. Excel that has been imported can be read by the **Read Excel** operator.

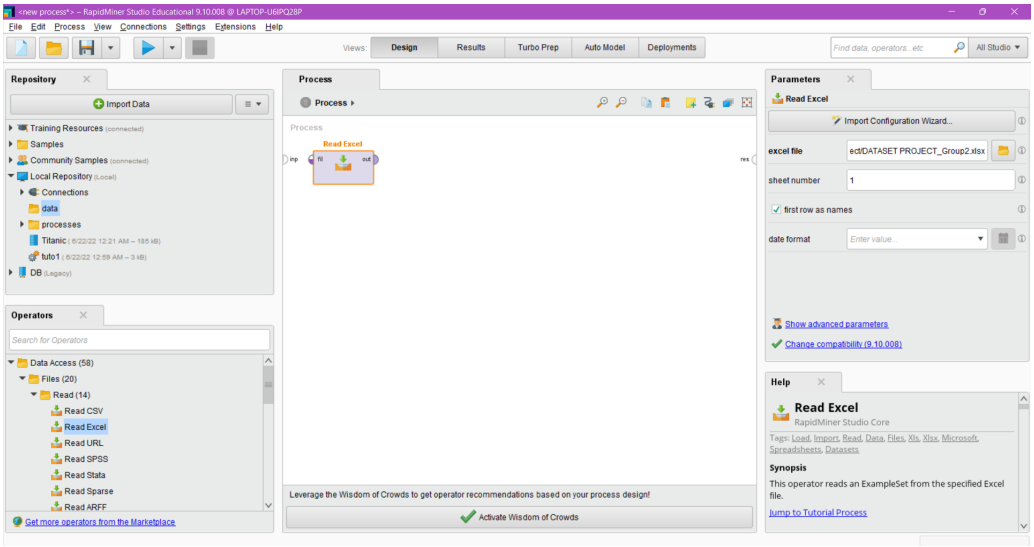


Figure 3: Read Excel operator

3. Change the document from nominal to text by inserting **Nominal to Text** operator. All the data will be converted into string value, which helps in the Data Transformation process.

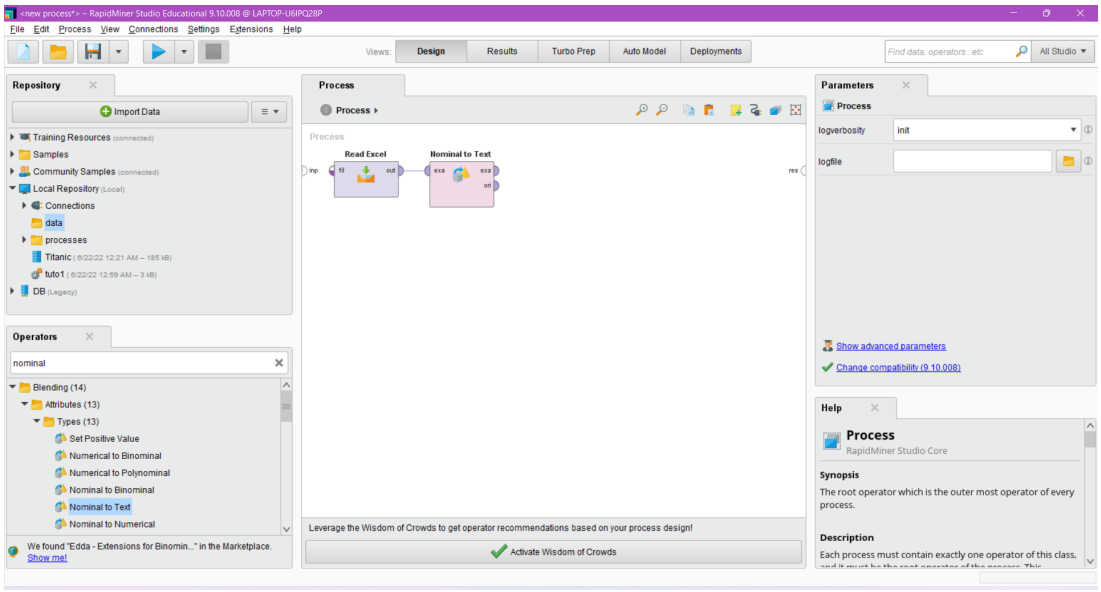


Figure 4: Nominal to Text operator

4. Add **Process Document from Data** operator for data transformation. Word vectors will be produced.

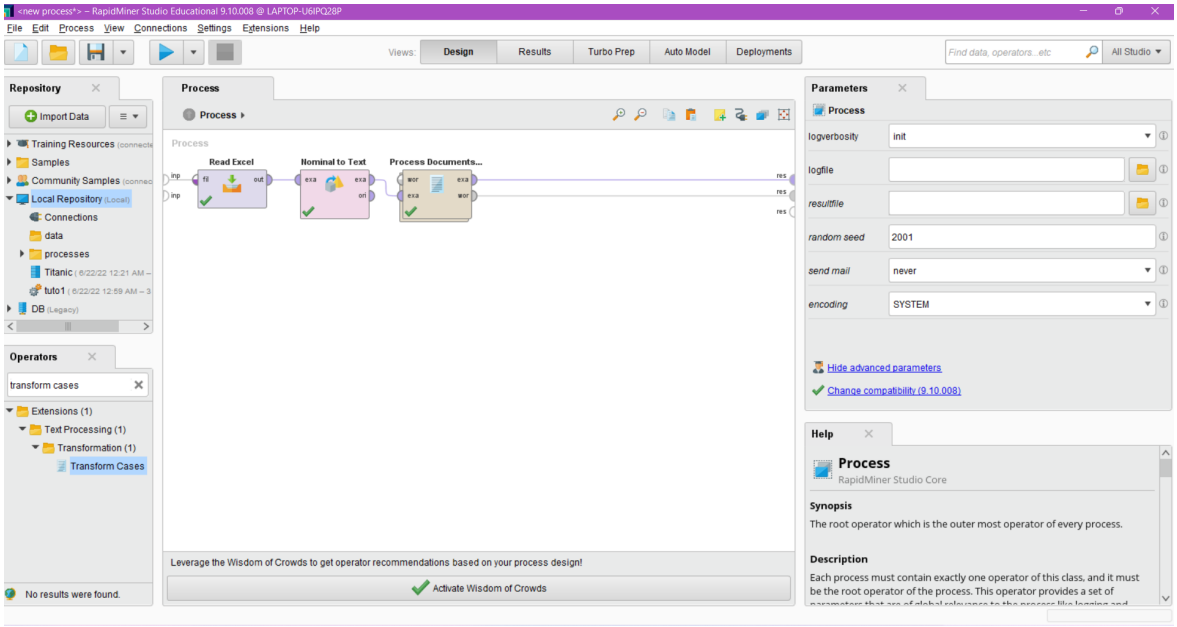


Figure 5: Process Document from Data operator

Double-clicking the operator and a new workspace will appear. We will start doing data cleaning and transformation from here.

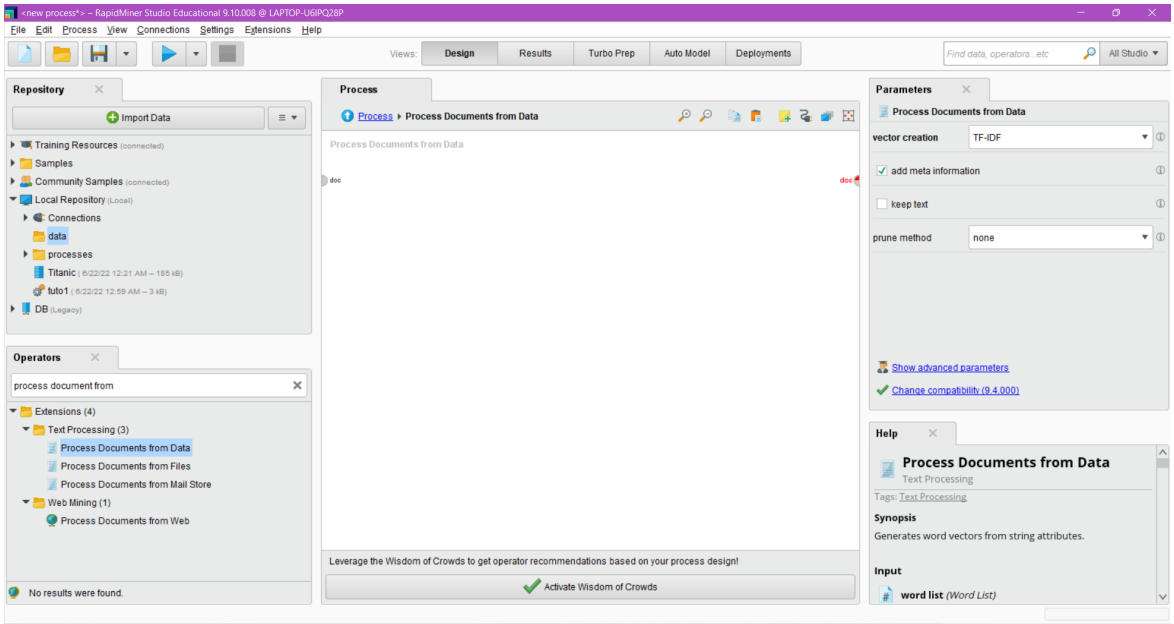


Figure 6: Workspace inside Process Document from Data operator

5. Start to do the process inside the workspace shown in figure 6. Data transformation starts with tokenization. Tokenize will split data into single words which in our dataset, it was

written in paragraphs. Thus, go to the operator and choose **Tokenize** inside the Tokenization file.

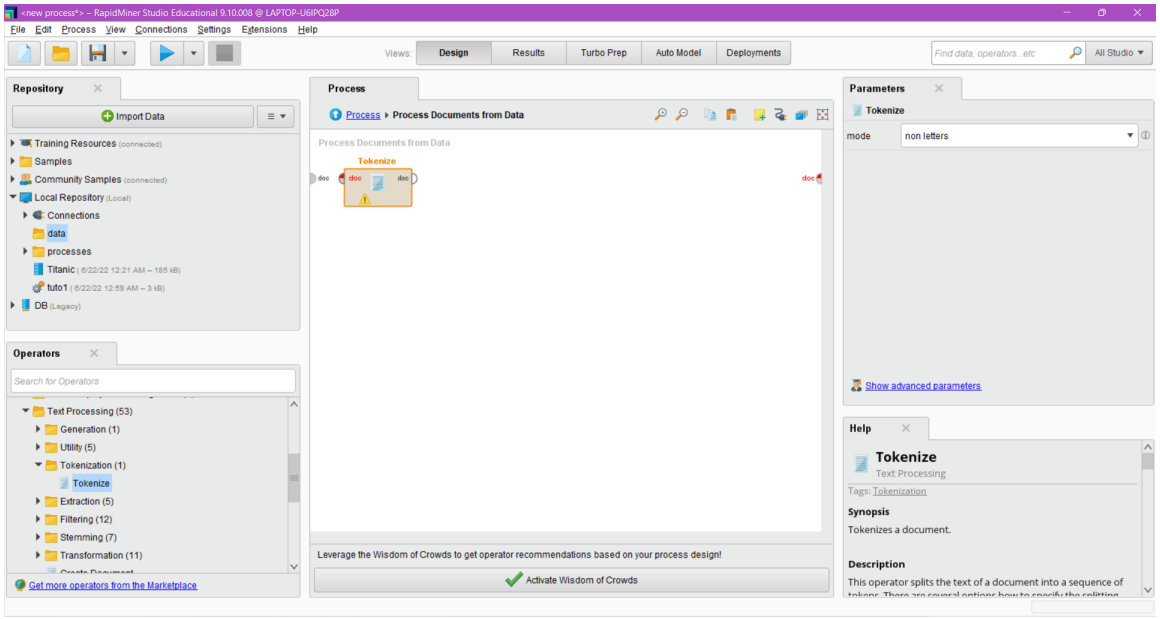


Figure 7: Tokenize operator

Word	Attribute Name	Total Occurences	Document Occurences
A	A	5	5
ABSTRACT	ABSTRACT	2	2
AT	AT	2	2
Actuarial	Actuarial	2	2
After	After	3	2
Association	Association	4	3
BRADYCARDIA	BRADYCARDIA	3	3
CARDIAC	CARDIAC	6	6
CORONARY	CORONARY	6	6
Cardiac	Cardiac	2	2
Class	Class	5	2
Coronary	Coronary	3	2
Ergonovine	Ergonovine	2	2
Functional	Functional	3	2
Heart	Heart	4	3
Hg	Hg	7	2
However	However	6	5
I	I	3	3

Figure 8: Output after tokenization

6. Transformed the data into lowercase to minimize the number of words which have the same meaning.

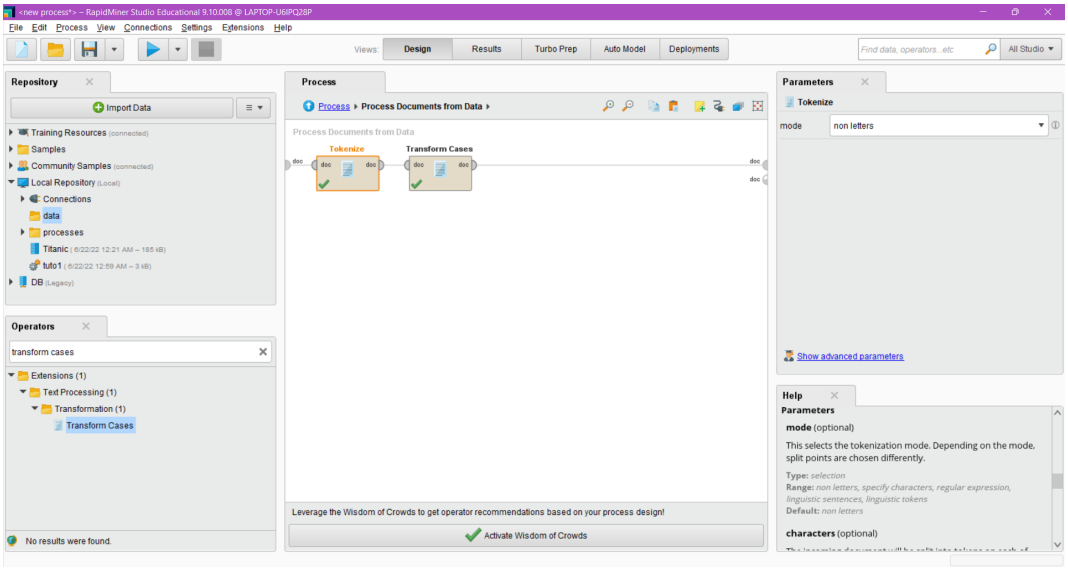


Figure 9: Transform Cases operator

Word	Attribut...	Total O...	Docum...
a	a	70	22
abnormal	abnormal	3	2
abnorma...	abnorma...	6	4
abstract	abstract	2	2
accurate	accurate	2	2
action	action	2	2
actuarial	actuarial	2	2
acute	acute	2	2
addition	addition	2	2
administ...	administ...	2	2
administ...	administ...	7	4
after	after	22	10
age	age	3	2
aged	aged	3	2
all	all	9	8
allow	allow	2	2
also	also	4	2
although	although	5	4

Figure 10: Output after transform cases

7. Based on figure 10, you can see that there are some unnecessary words. Filtering stop word operators is necessary to produce a useful wordlist. Choose the **Filter Stopwords** operator according to language used.

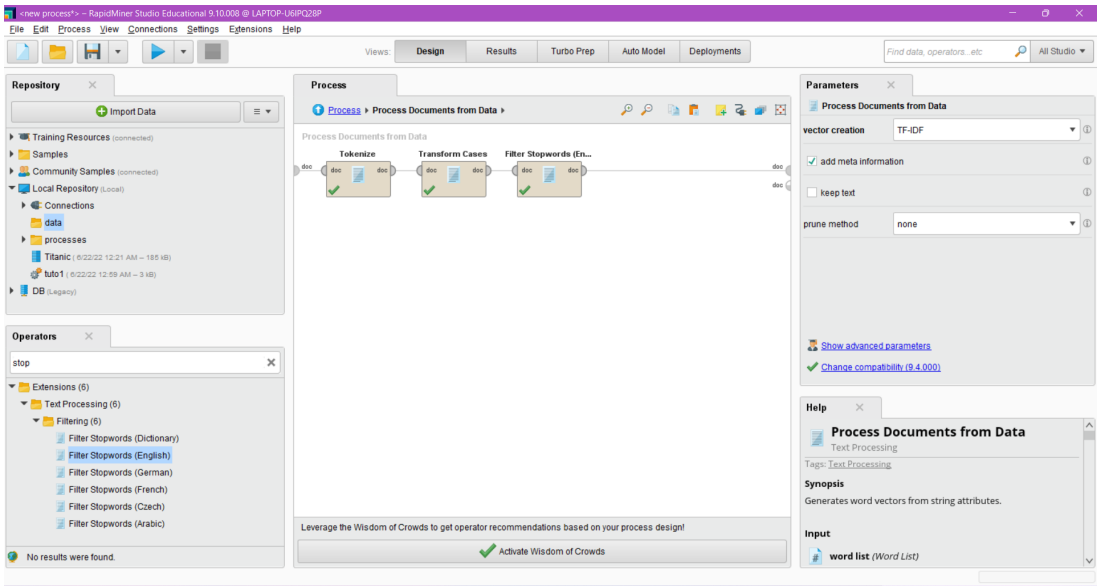


Figure 11: Filter Stopwords operator

Word	Attribut...	Total O...	Docum...
abnormal	abnormal	3	2
abnorma...	abnorma...	6	4
abstract	abstract	2	2
accurate	accurate	2	2
action	action	2	2
actuarial	actuarial	2	2
acute	acute	2	2
addition	addition	2	2
administ...	administ...	2	2
administ...	administ...	7	4
age	age	3	2
aged	aged	3	2
allow	allow	2	2
amplitude	amplitude	2	2
angina	angina	13	4
angiogra...	angiogra...	3	3
anterior	anterior	2	2
aorta	aorta	5	2

Figure 12: Output after filtering stopwords

8. Some of the words might have the same meaning. Thus, we must minimize the words by converting them into root words. Use the **Stem (Snowball)** operator for this.

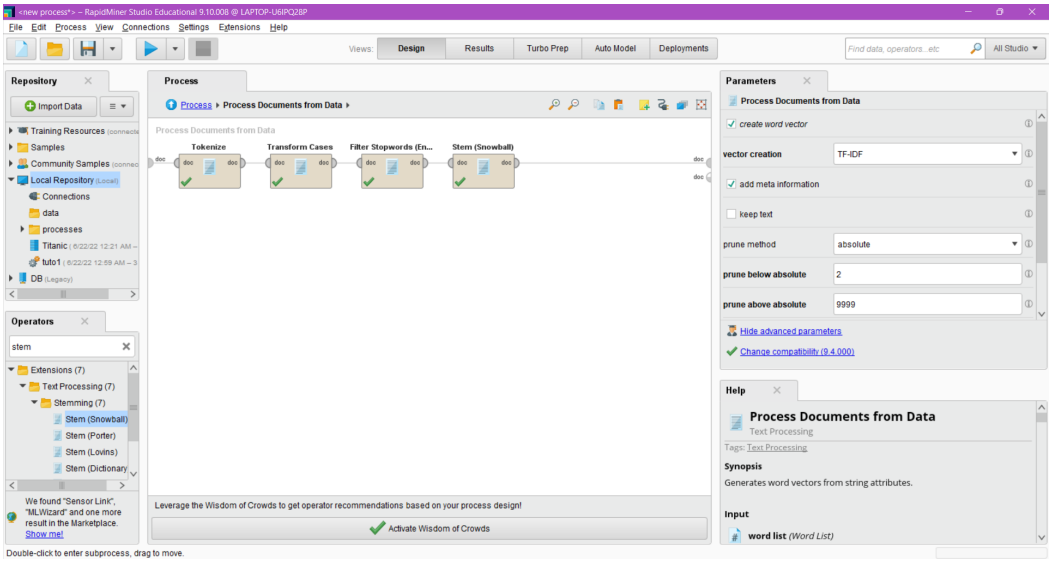


Figure 13: Stem(Snowball) operator

Word	Attribute ...	Total O...	Docum...
abnorm	abnorm	10	6
abstract	abstract	2	2
accept	accept	2	2
accur	accur	2	2
action	action	2	2
activ	activ	2	2
actuari	actuari	2	2
acut	acut	2	2
addit	addit	2	2
administ	administ	2	2
administr	administr	7	4
age	age	6	3
allow	allow	3	3
alter	alter	2	2
amplitud	amplitud	2	2
anaesthet	anaesthet	2	2
angina	angina	13	4
angiograph	angiograph	5	4

Figure 13: Output after converting all into root words

9. Since this is a medical dataset, we can identify those medical terms by **Generate n-grams** operator. The max length for this operator we set up as 2.

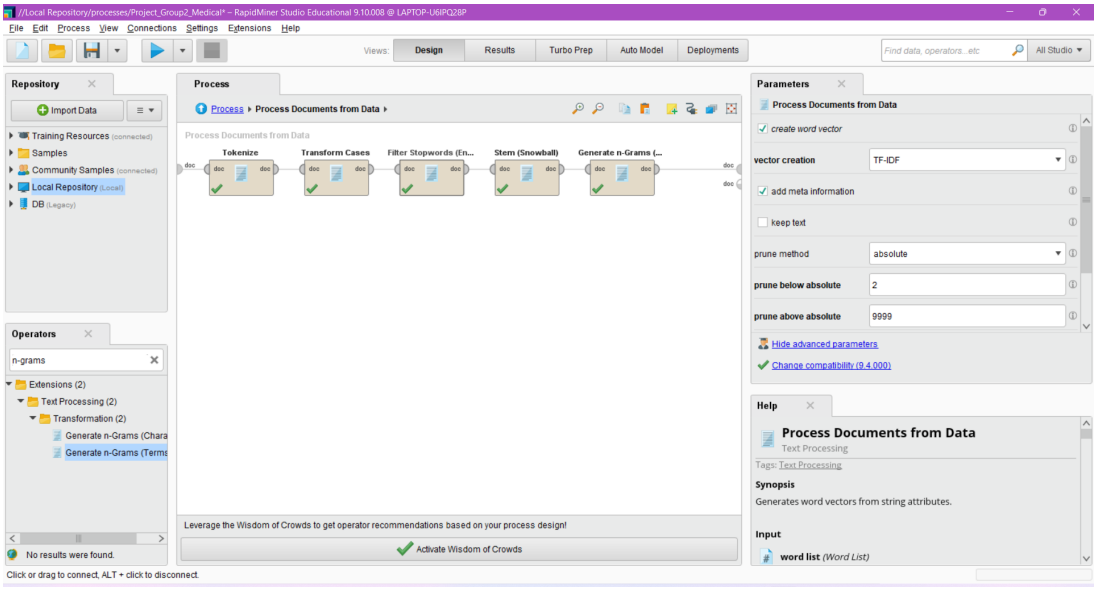


Figure 14: Generate n-grams operator

Word	Attribute Name	Total O...	Docum...
abnorm	abnorm	10	6
abstract	abstract	2	2
abstract_truncat	abstract_truncat	2	2
accept	accept	2	2
accur	accur	2	2
action	action	2	2
activ	activ	2	2
actuari	actuari	2	2
actuari_surviv	actuari_surviv	2	2
acut	acut	2	2
addit	addit	2	2
administ	administ	2	2
administr	administr	7	4
age	age	6	3
age_year	age_year	6	3
allow	allow	3	3
alter	alter	2	2
amplitud	amplitud	2	2

Figure 15: Output after generate n-grams operator

4.0 DATA MINING TASK

4.1 CLASSIFICATION (NAIVE BAYES)

Naive Bayes classifiers are a group of classification algorithms that are based on Bayes' Theorem used to solve classification problems. Naive Bayes is said to be one of the most common and effective supervised learning methods for quickly predicting machine learning models. It can even be used to build a model from a small data set and is less costlier than other algorithms. For this task, the preprocessing is done differently to what was previously mentioned. Below are the steps when involving Naive Bayes in an activity.

1. First, drag the "Read Excel" operator from the Operators panel into the Process field, then double click the operator in the Process field and search for the dataset to work on.

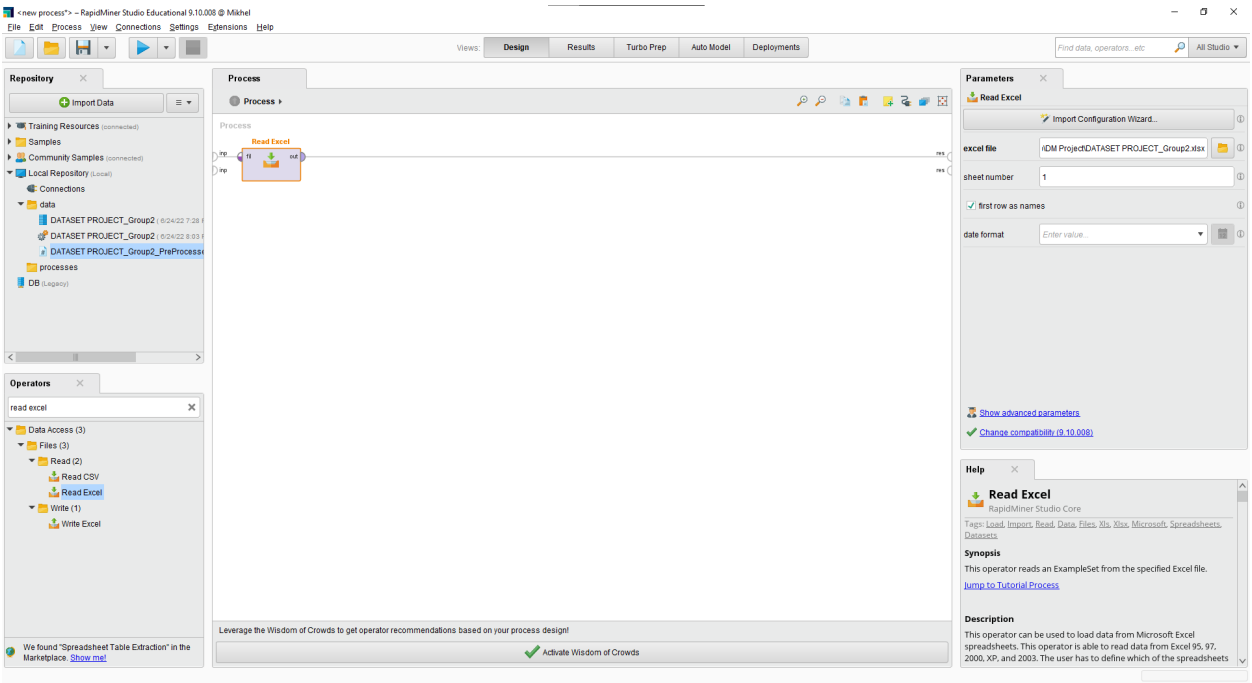


Figure 16: Adding "Read Excel" operator into Process field

2. After that, search for the "Numerical to Polynomial" operator in the Operators panel and drag it into the Process field and then in the Parameters panel, select "no_missing_values" in the attribute filter type dropdown menu. This is used to change

the attributes from numerical to polynomial type and also maps all values of these attributes to corresponding polynomial values.

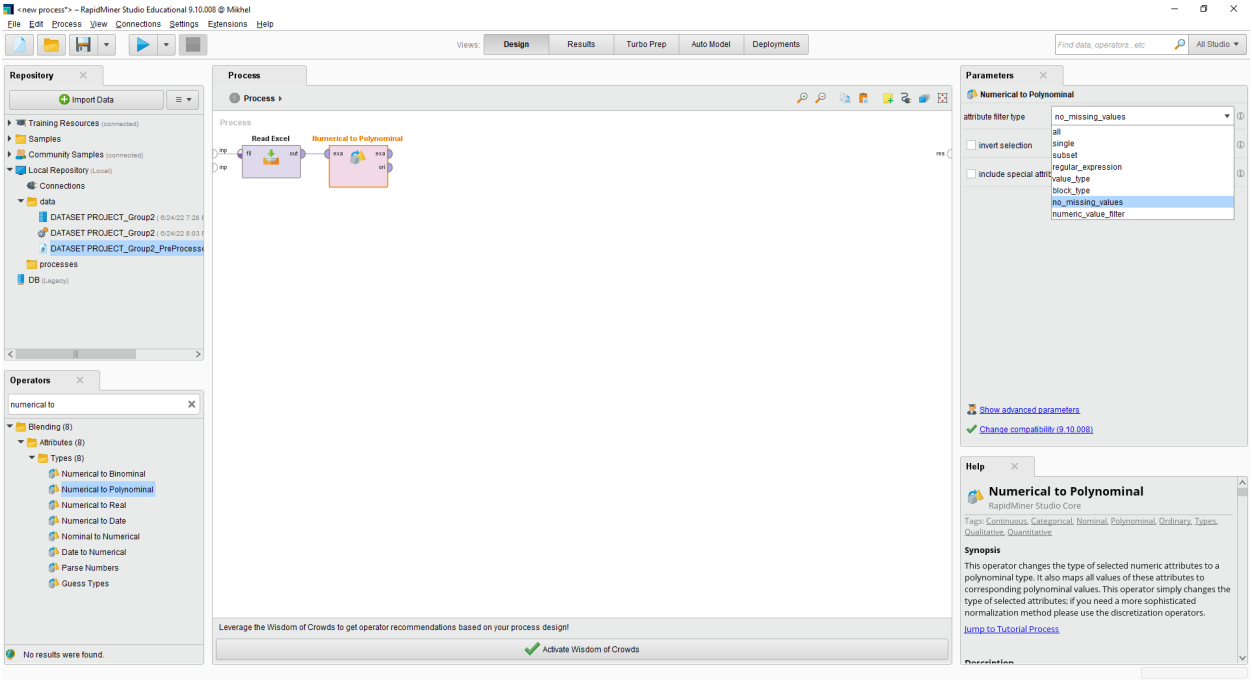


Figure 17: Adding “Numerical to Polynomial” operator to process field

3. Next, search and drag the “Set Role” operator to the Process field then in the Parameters panel, select “Disease” as the attribute name and “label” as the target role. Disease was chosen as the attribute name because it represents the main column in the dataset.

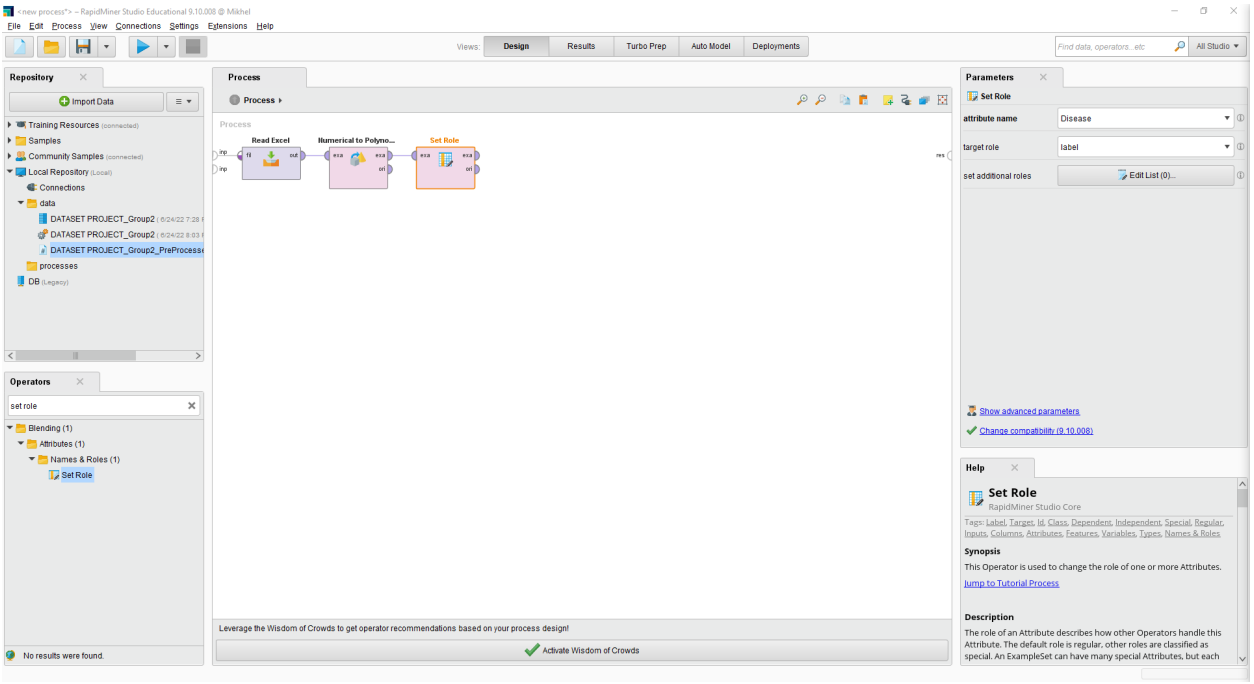


Figure 18: Adding “Set Role” operator to process field

4. The next step is to search and drag the operator “Select Attributes” into the Process field and select “no_missing_values” as the attribute filter type in the dropdown menu under the Parameters panel. This will be used to select the attributes that will be used in the predictions and remove the unnecessary ones such as duplicated attributes.

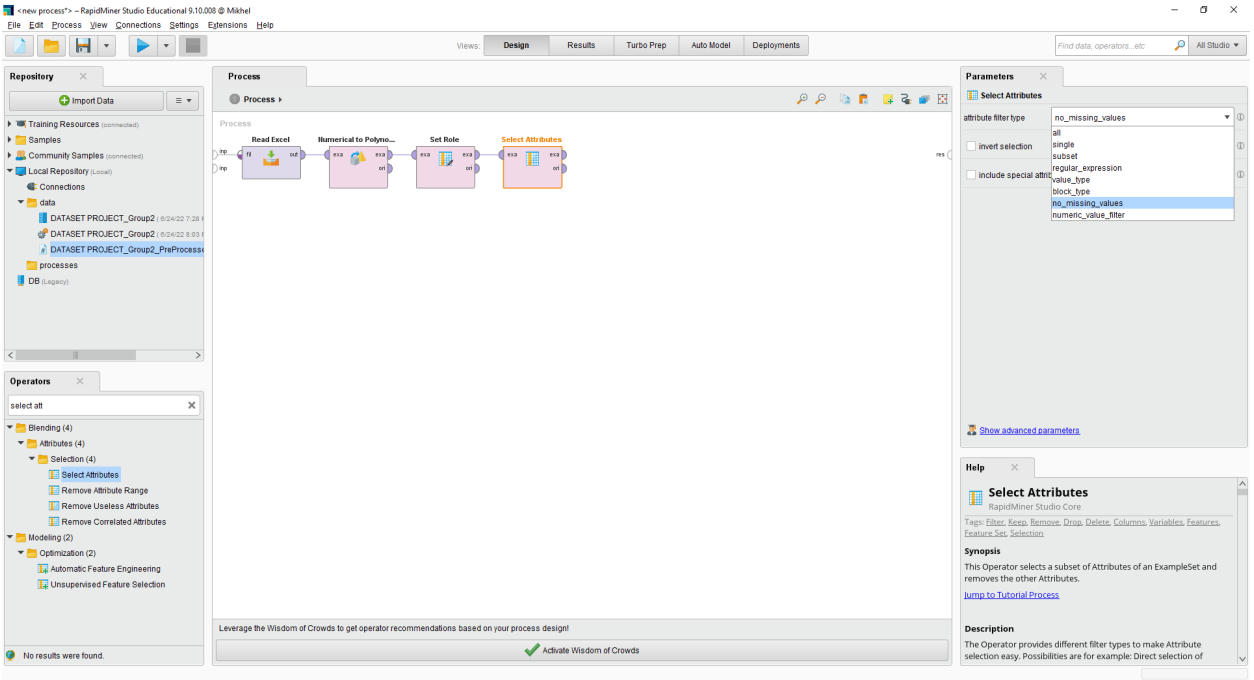


Figure 19: Adding “Select Attributes” operator

5. Now we need to add the “Cross Validation” operator into the Process field. This will perform cross validation to estimate the statistical performance of the built model. Then, double click the operator to enter the Cross Validation field.

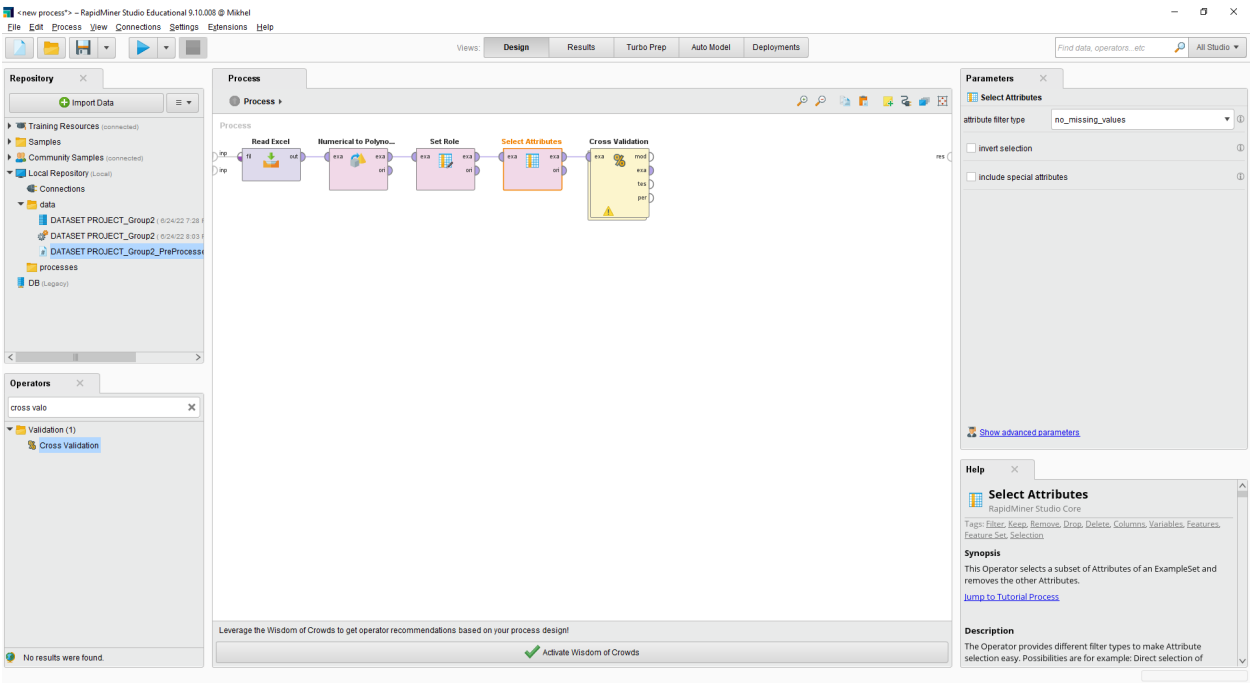


Figure 20: Adding “Cross Validation” operator

6. Next, add the “Naive Bayes” operator into the Training field (left side) found in the “Cross Validation” operator. This is where the classification process happens.

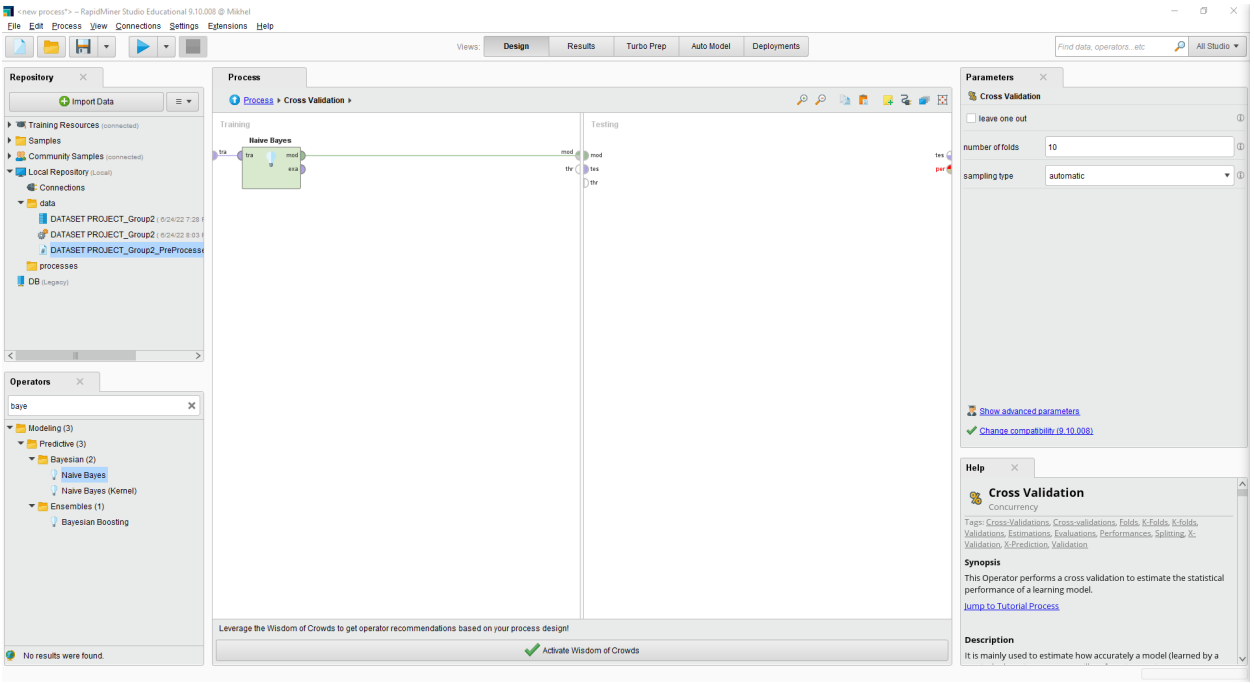


Figure 21: Adding “Naive Bayes” operator into training field of “Cross Validation” operator

7. After that, add the “Apply Model” operator into the Testing field (right side) found in the “Cross Validation” operator. This will apply the model that was built onto the dataset.

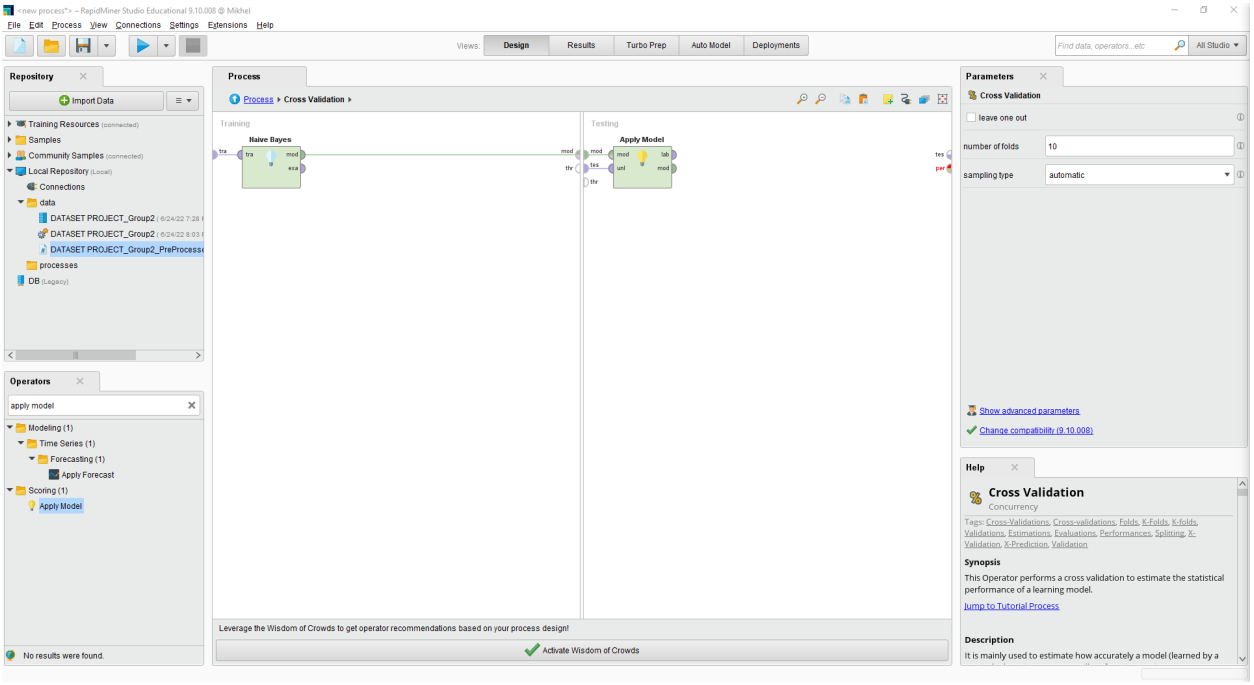


Figure 22: Add “Apply Model” into Training field

8. Now add the “Performance (Classification)” into the Testing field (right side). This is used to assess the statistical performance of a classification task and display the list of performance criteria values of the tasks.

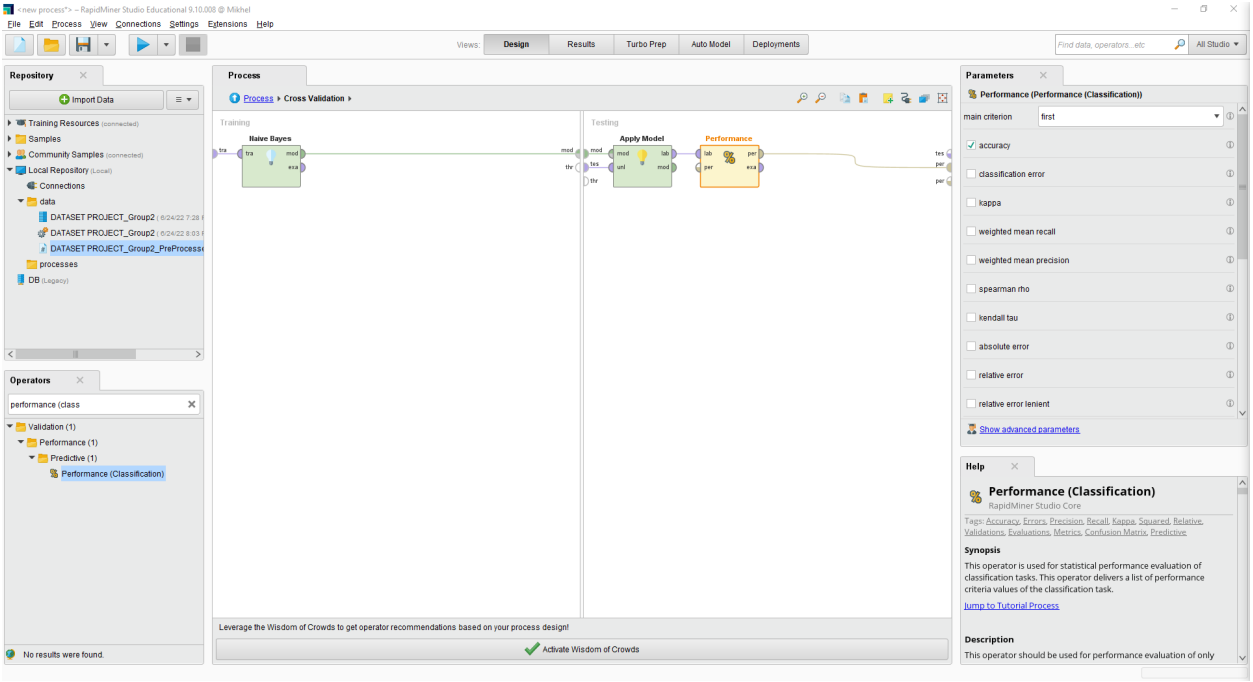


Figure 23: Add “Performance (Classification)” operator to Testing field

9. Finally, connect all the operators accordingly and click “Run” to display the results of the classification task.

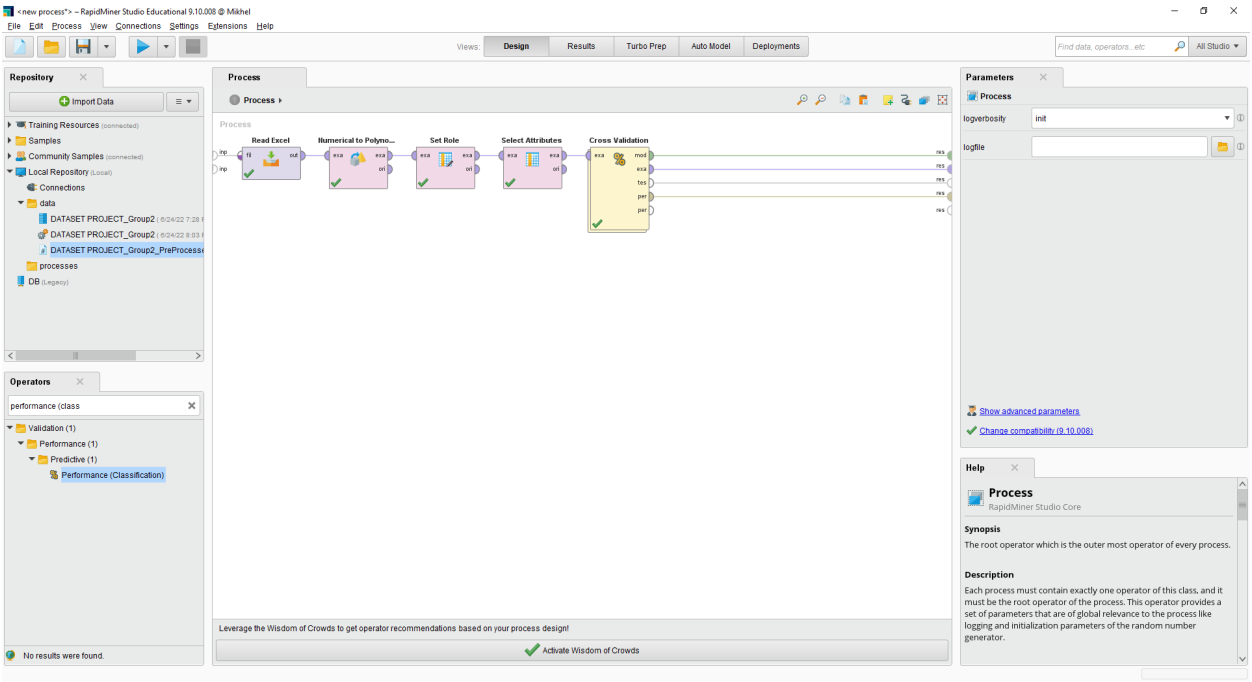


Figure 24: Making the connections between operators

Below are the results of the task:

accuracy: 40.00% +/- 22.50% (micro average: 39.13%)

	true CARDIAC OUTPUT, LOW	true CORONARY VASOSPASM	true MARFAN SYNDROME	true PERICARDITIS	true BRADYCARDIA	class precision
pred. CARDIAC OUTPUT, LOW	0	0	0	0	0	0.00%
pred. CORONARY VASOSPASM	0	0	0	0	0	0.00%
pred. MARFAN SYNDROME	3	0	4	0	0	57.14%
pred. PERICARDITIS	0	2	0	2	0	50.00%
pred. BRADYCARDIA	3	4	0	2	3	25.00%
class recall	0.00%	0.00%	100.00%	50.00%	100.00%	

Figure 25: Performance Vector accuracy results

PerformanceVector

```
PerformanceVector:
accuracy: 40.00% +/- 22.50% (micro average: 39.13%)
ConfusionMatrix:
True:    CARDIAC OUTPUT, LOW     CORONARY VASOSPASM      MARFAN SYNDROME PERICARDITIS    BRADYCARDIA
CARDIAC OUTPUT, LOW:    0       0       0       0
CORONARY VASOSPASM:     0       0       0       0       0
MARFAN SYNDROME:        3       0       4       0       0
PERICARDITIS:   0       2       0       2       0
BRADYCARDIA:    3       4       0       2       3
```

Figure 26: Performance Vector description

SimpleDistribution

```
Distribution model for label attribute Disease


Class CARDIAC OUTPUT, LOW (0.261)
2 distributions

Class CORONARY VASOSPASM (0.261)
2 distributions

Class MARFAN SYNDROME (0.174)
2 distributions

Class PERICARDITIS (0.174)
2 distributions

Class BRADYCARDIA (0.130)
2 distributions
```

Figure 27: Simple Distribution

4.2 REGRESSION AND PREDICTION (SUPPORT VECTOR MACHINE)

SVMs are a set of related supervised learning methods used for classification and regression. They belong to a family of generalized linear classification. A special property of SVM is , SVM simultaneously minimizes the empirical classification error and maximizes the geometric margin (Bhavsar et al.,2012).

The data mining technique of regression is frequently used to forecast a variety of continuous values (sometimes referred to as "numeric values") in a dataset. Each piece of data will be plotted onto a hyperplane multidimensional plane by the model, with each feature's value representing the value of a certain coordinate in the SVM method. The hyperplane that clearly separates the two classes is then located, and classification or regression is then performed. The two types of SVM are linear and nonlinear, respectively.

1. Retrieve imported data from data repository by drag and drop it from repository to process canvas.

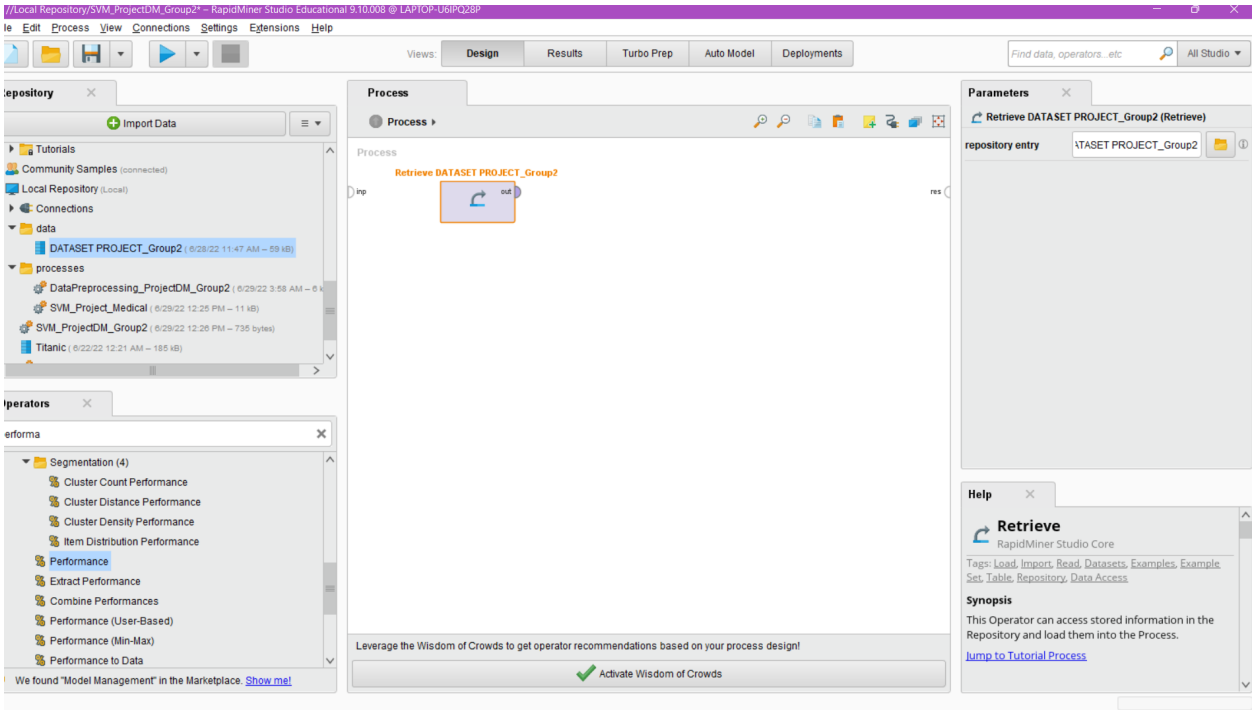


Figure 28: Retrieve imported data

2. Search and drag “Select Attributes” operator from operators to process. This operator will select a subset of attributes of an ExampleSet and remove the other attributes. Choose all for ‘attribute filter type’.

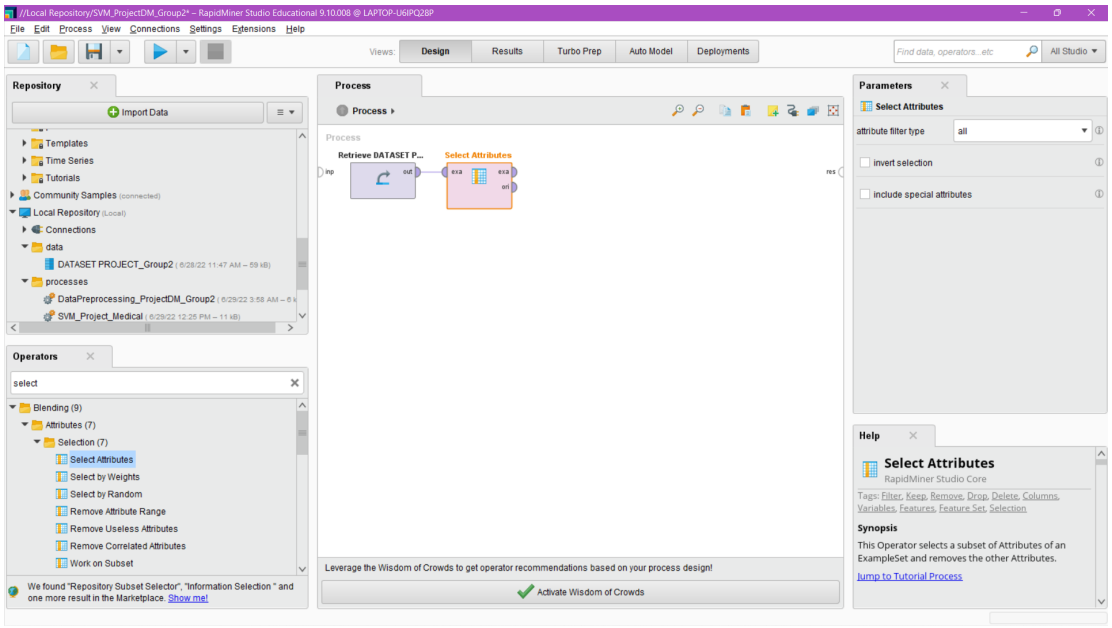


Figure 29: Use Select Attributes operator

3. Search and drag “Set Role” operator from operators to process. This operator can change the role of one or more attributes. Choose attribute name to “attribute name” and target role as ‘label’.

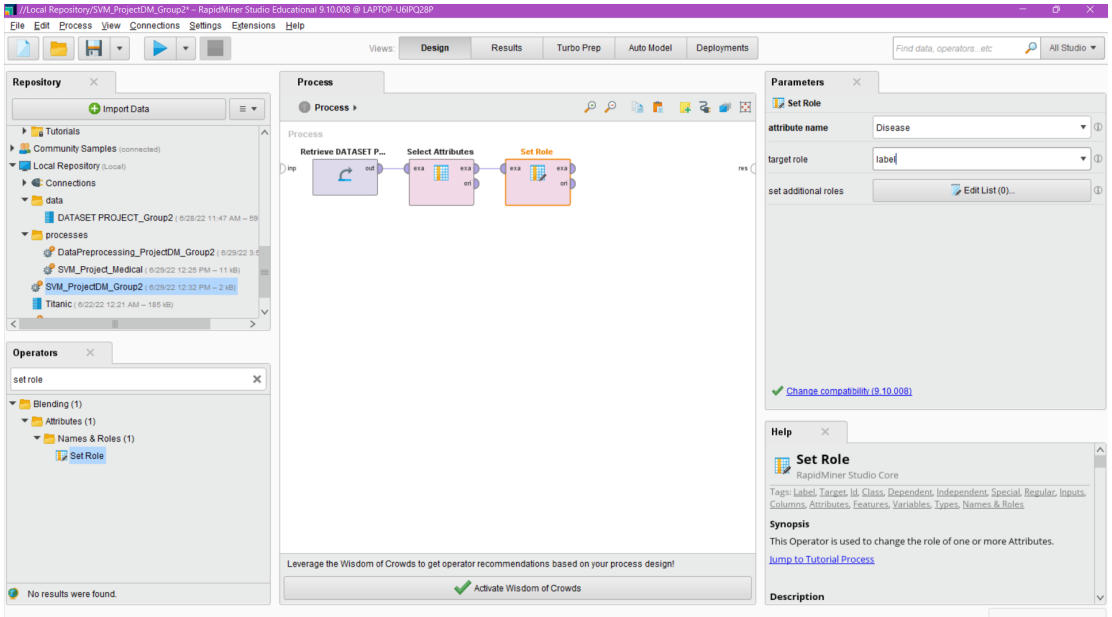


Figure 30: Use Set Role operator

4. Search and drag “Nominal to Text” operator from operators to process. Using this operator, specified nominal properties are converted to text. Additionally, it converts each value of these properties into a string value.

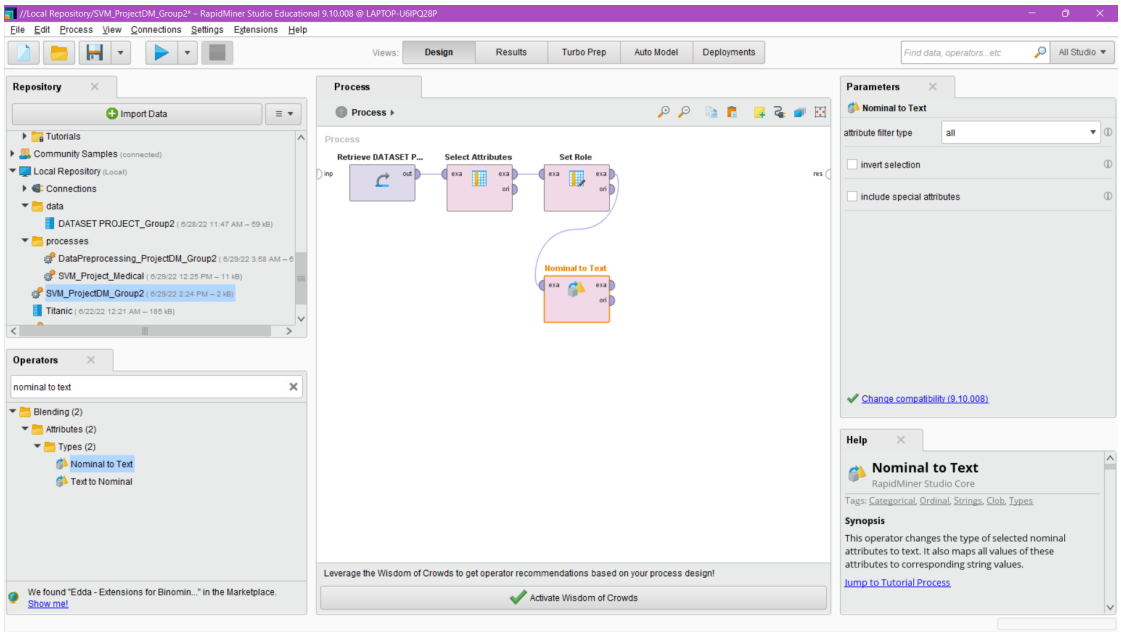


Figure 31: Use Nominal to Text operator

5. Search and drag “Process Document from Data” operator from operators to process. This will generate word vectors from string attributes.

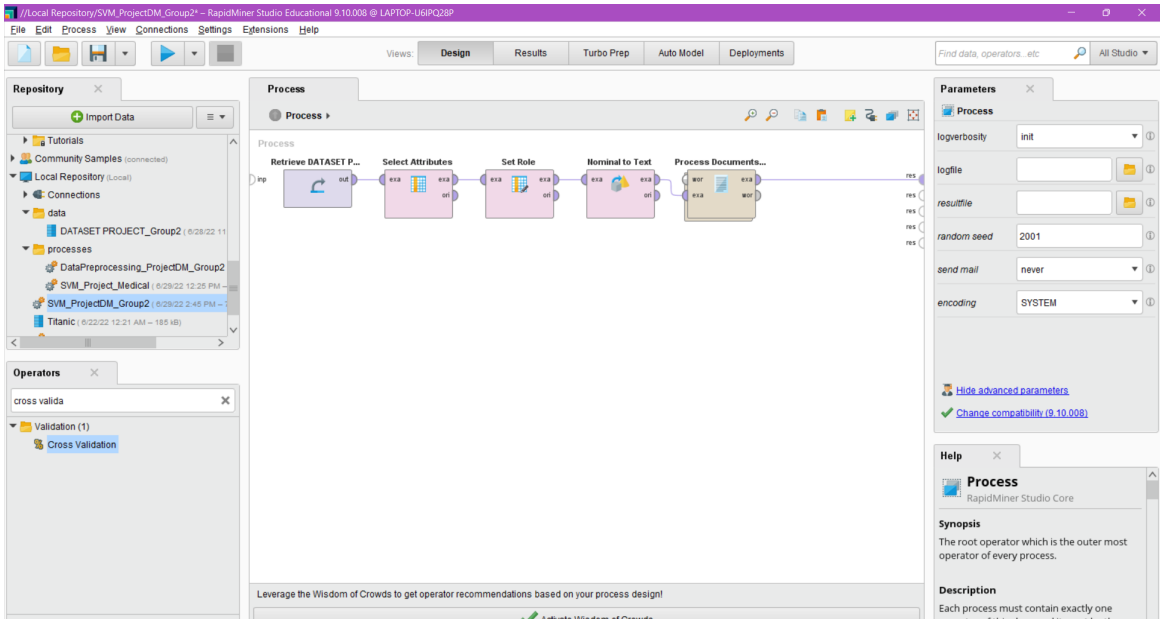


Figure 32: Use Process Document from Data operator

6. Double click ‘Process Document from Data’ operator to start adding more operators there. Operators needed in here are as follows:
    - **Tokenize** - Tokenize the document (split into words)
    - **Filter Stopwords (English)** - Removes English stopwords from a document.
    - **Transform Cases** - Transforms cases of characters in a document. (Lower case)
    - **Filter Tokens (by Length)** - Filters tokens based on their length (i.e. the number of characters they contain).
    - **Generate n-Grams (Terms)** - n-Gram is defined as a series of consecutive tokens of length n.
    - **Stem (Snowball)** - Stems words by applying stemming algorithms written for the Snowball language. It translates a Snowball script (a . sbl file) into a program.

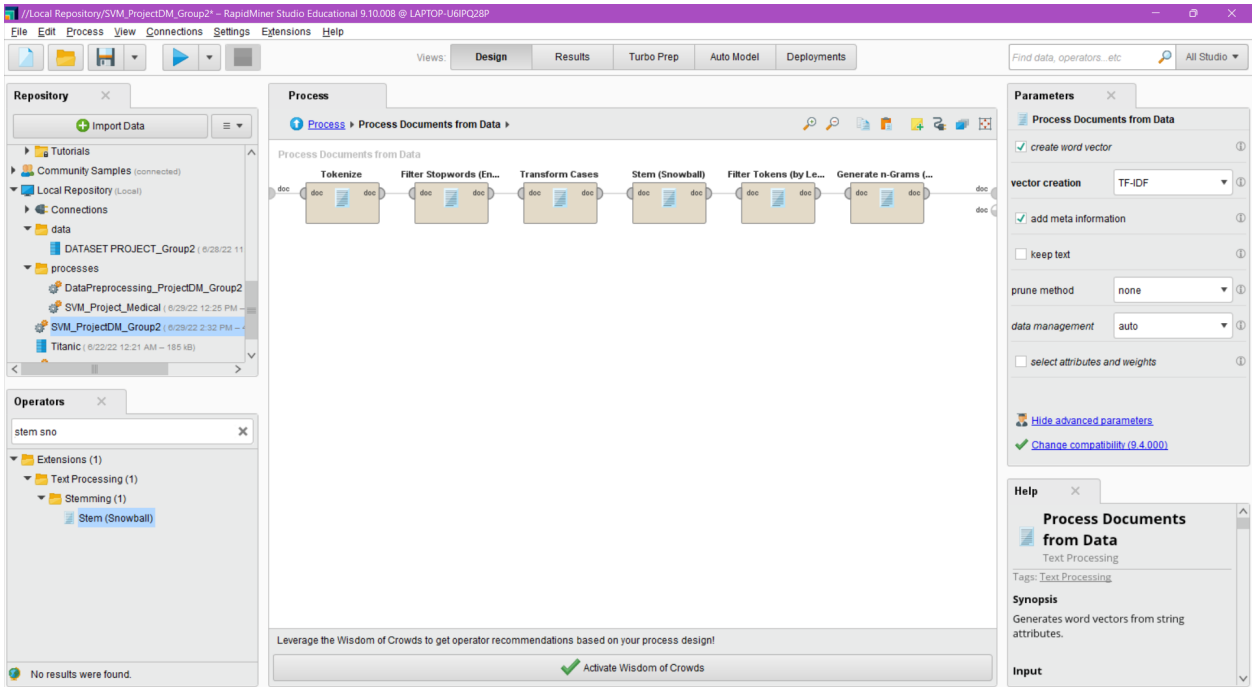


Figure 33: Connect few operators inside Process Document from Data operator

7. Search and drag “Cross Validation” operator from operators to process. It is mostly used to predict how well a model (trained by a specific learning Operator) would function in a real-world situation.The nested operator is the cross-validation operator. A Training subprocess and a Testing subprocess are its two subprocesses. A model is trained using the Training subprocess. The Testing subprocess then uses the learned model. During the Testing phase, the model's effectiveness is evaluated.

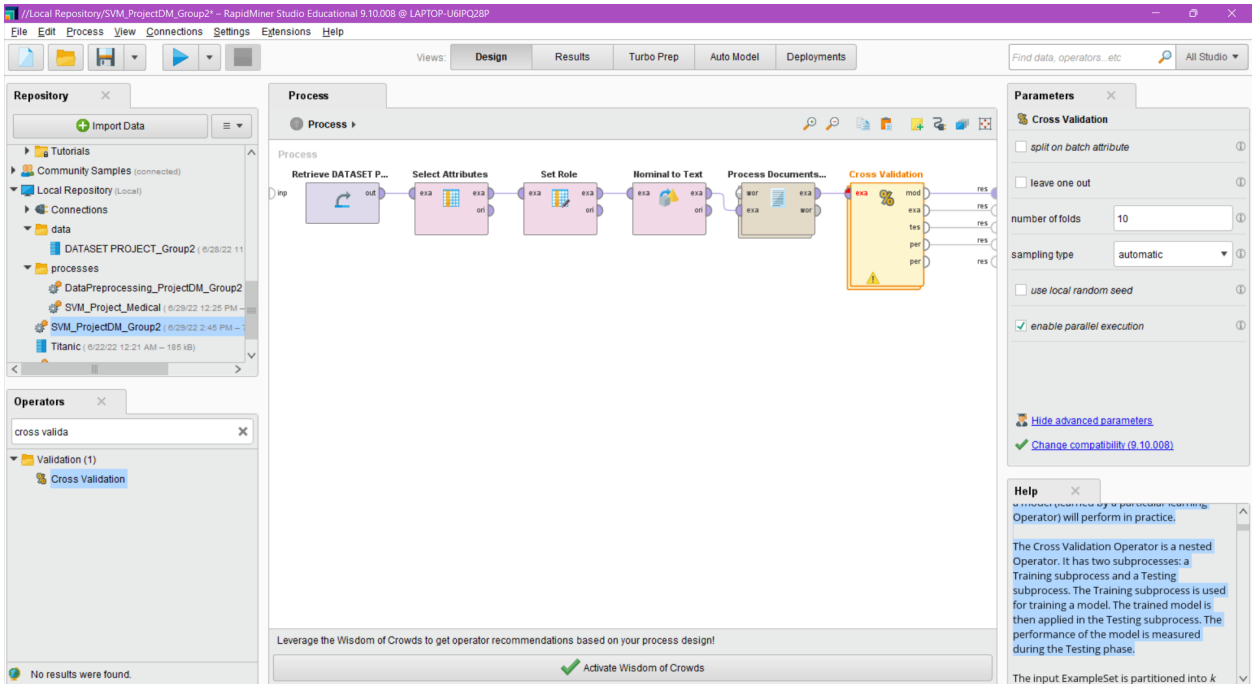


Figure 34: Use Cross Validation

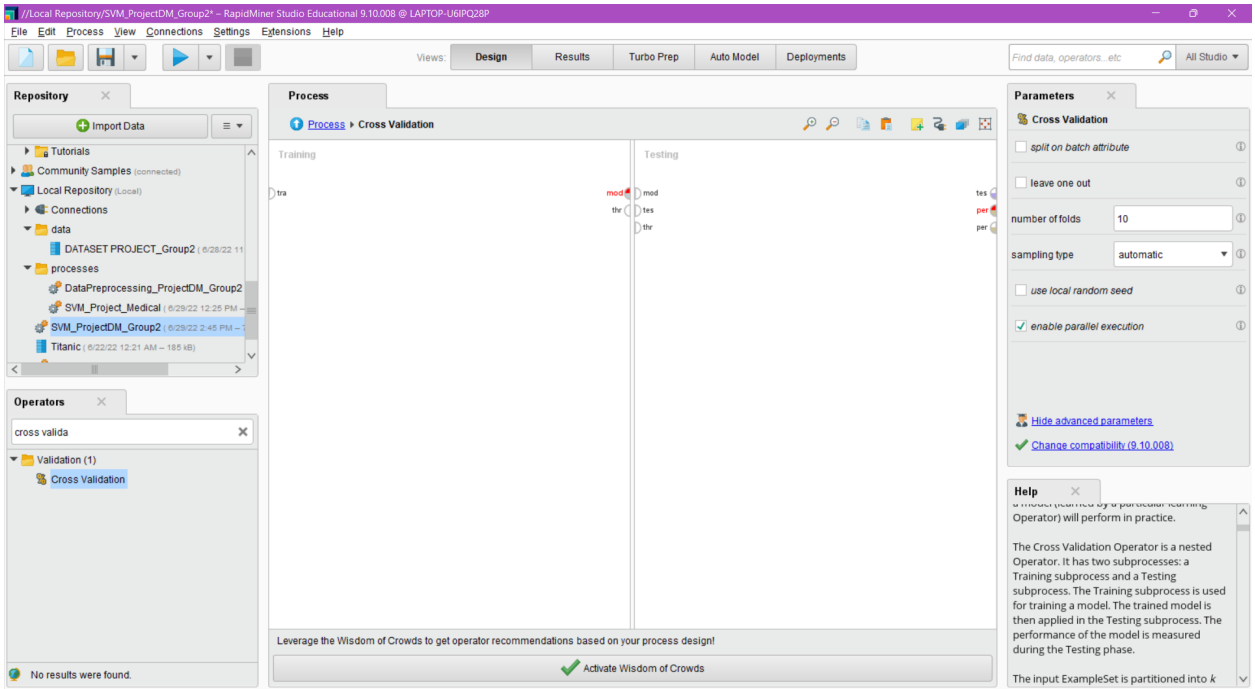


Figure 35: Inside Cross Validation operator

8. This step is inside the Cross Validation operator. Search and drag “Support Vector Machine (libSVM)” operator from operators to training process. It is a SVM (Support vector machine) Learner. It is based on the Java libSVM. The standard SVM takes a set

of input data and predicts, for each given input, which of two possible classes comprises the input, making the SVM a non-probabilistic binary linear classifier.

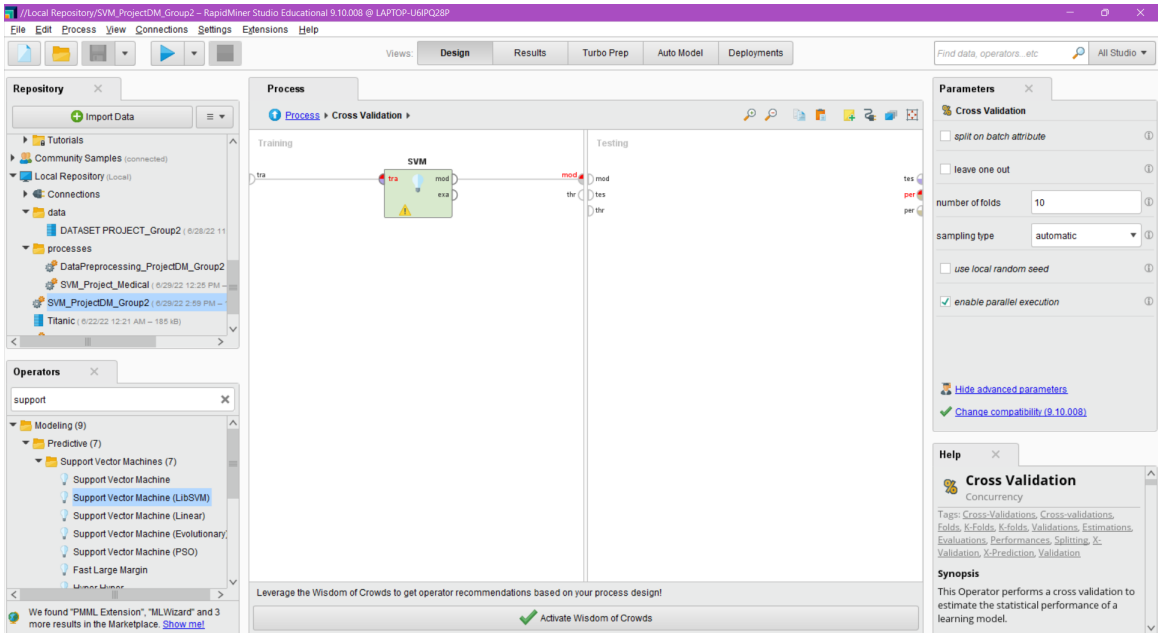


Figure 36: Use SVM in training process inside Cross Validation operator

Search and drag “Apply Model” operator from operators to testing process. This will apply a model on an ExampleSet.The goal is to get a prediction on unseen data or to transform data by applying a preprocessing model.

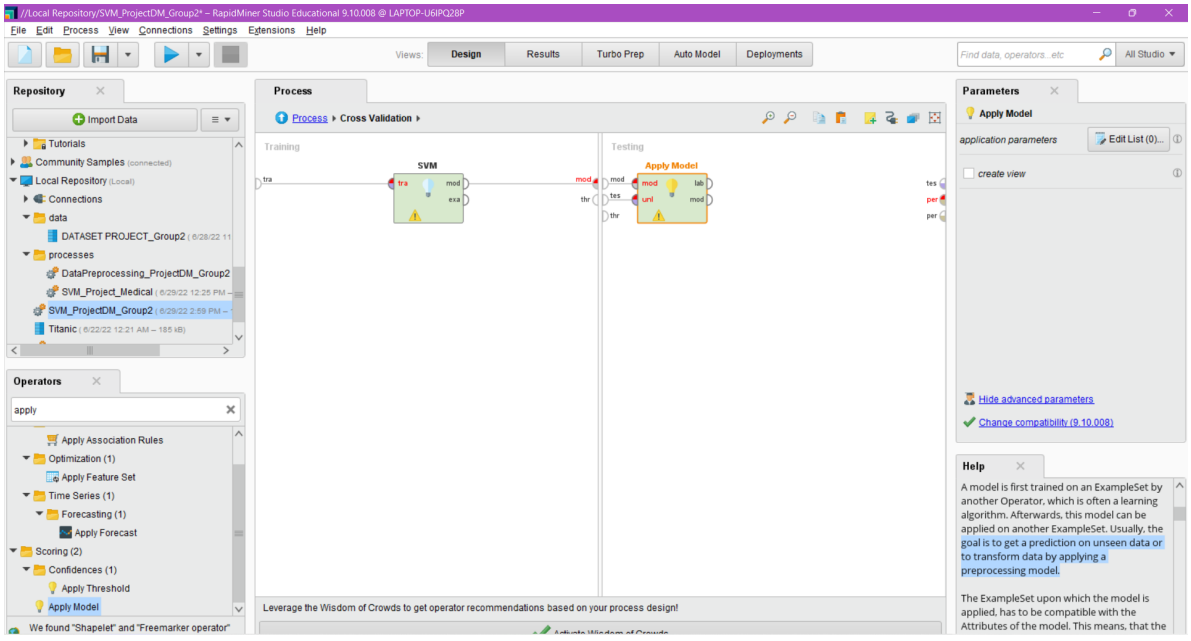


Figure 37: Use Apply Model operator in testing process inside Cross Validation operator

Search and drag “Performance” operator from operators to testing process. It is used for performance evaluation. It delivers a list of performance criteria values. These performance criteria are automatically determined in order to fit the learning task type

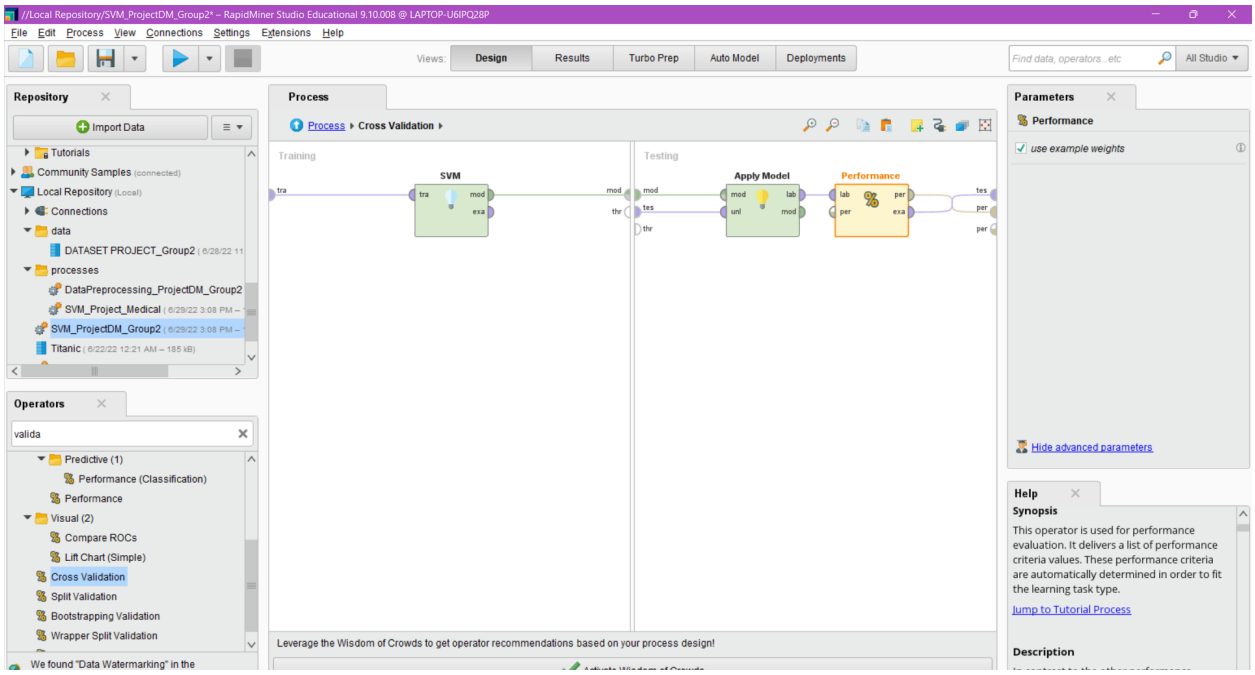
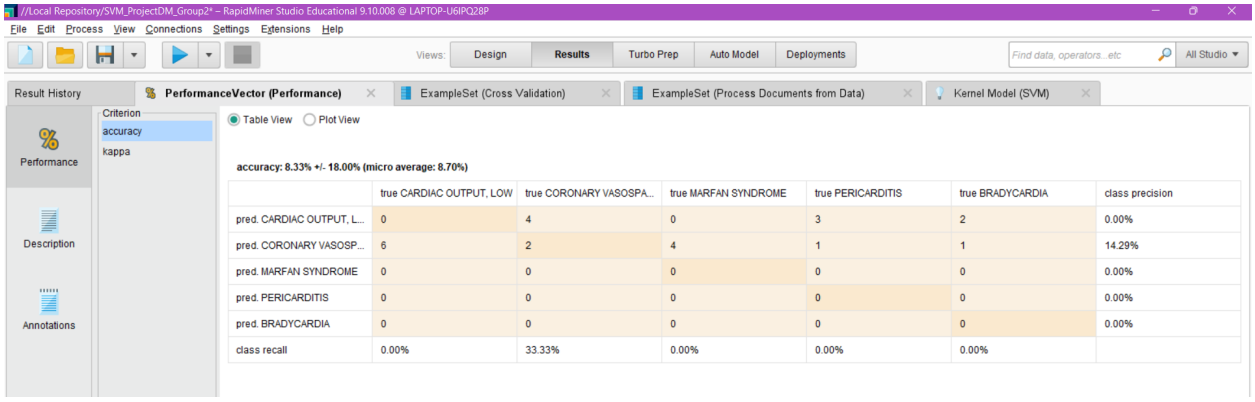


Figure 38: Use Performance operator in testing process inside Cross Validation operator

9. Go back to the main process and start executing it.

Results



	true CARDIAC OUTPUT, LOW	true CORONARY VASOSPA...	true MARFAN SYNDROME	true PERICARDITIS	true BRADYCARDIA	class precision
pred. CARDIAC OUTPUT, L...	0	4	0	3	2	0.00%
pred. CORONARY VASOSP...	6	2	4	1	1	14.29%
pred. MARFAN SYNDROME	0	0	0	0	0	0.00%
pred. PERICARDITIS	0	0	0	0	0	0.00%
pred. BRADYCARDIA	0	0	0	0	0	0.00%
class recall	0.00%	33.33%	0.00%	0.00%	0.00%	

Figure 39: Results based on Performance Vector (Table View)

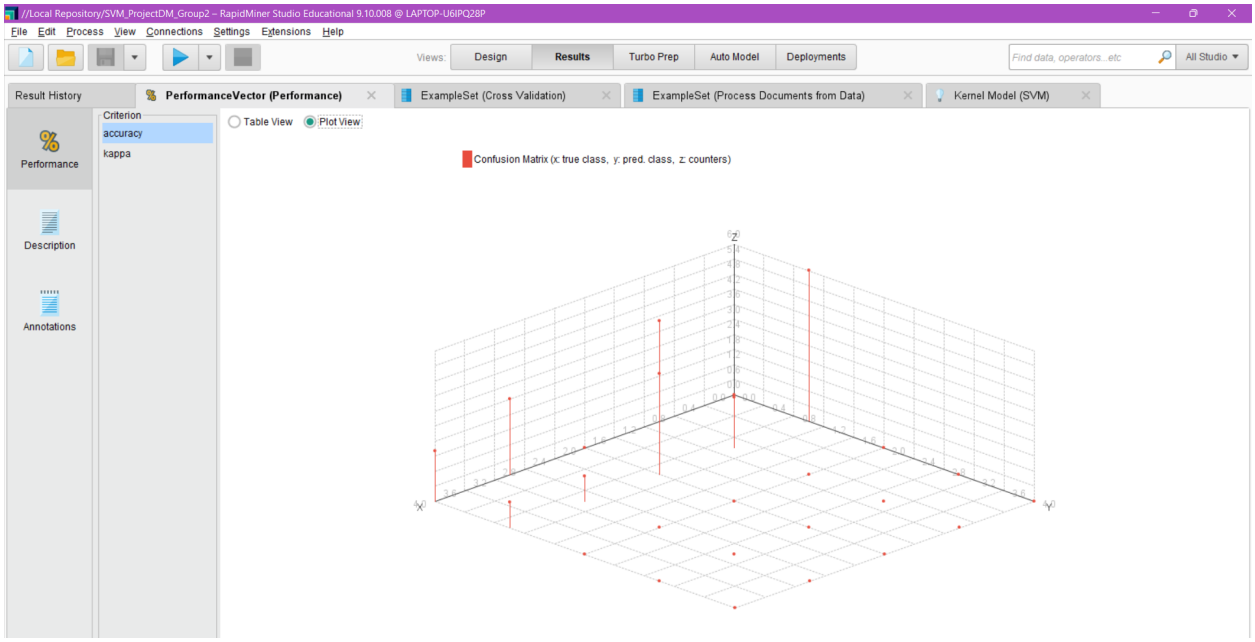


Figure 40: Results based on Performance Vector (Plot View)

Kernel Model

```
Total number of Support Vectors: 23
Bias (offset): -0.000

Feature weight calculation only possible for two class learning problems.
Please use the operator SVMWeighting instead.

number of classes: 5
number of support vectors for class CARDIAC OUTPUT, LOW: 6
number of support vectors for class CORONARY VASOSPASM: 6
number of support vectors for class MARFAN SYNDROME: 4
number of support vectors for class PERICARDITIS: 4
number of support vectors for class BRADYCARDIA: 3
```

Figure 41: Results in Kernel Mode (SVM) by description

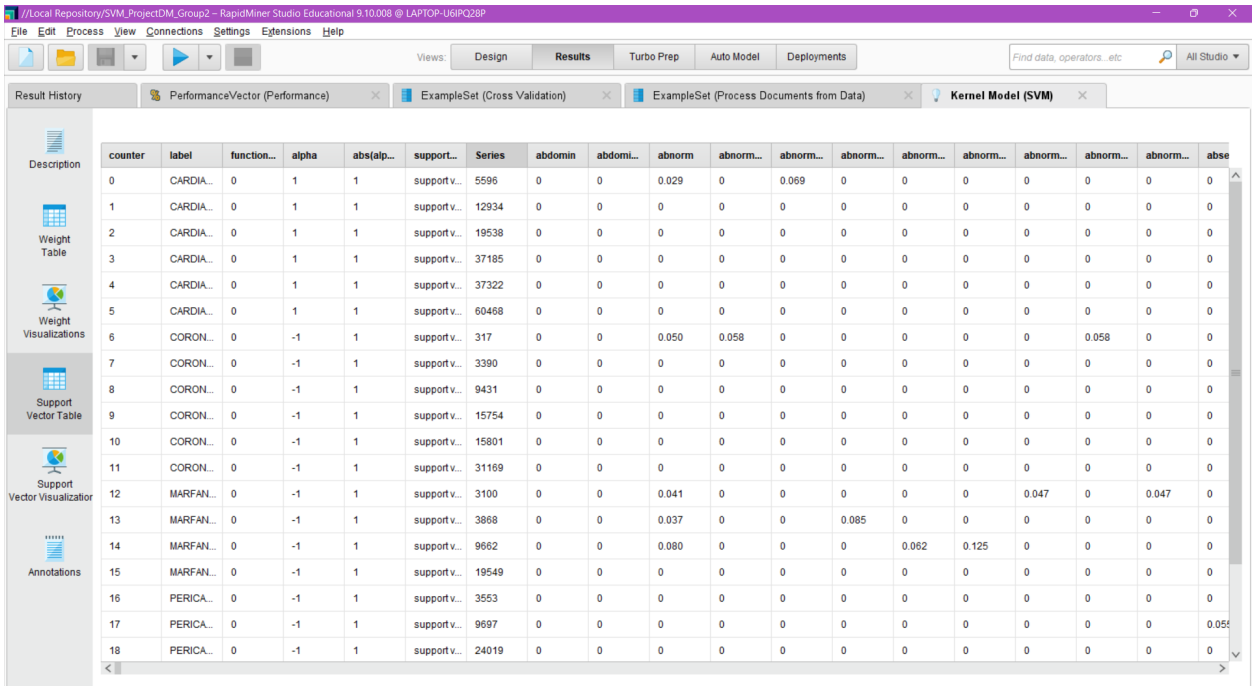


Figure 42: Results in Kernel Mode (SVM) by Support Vector Table

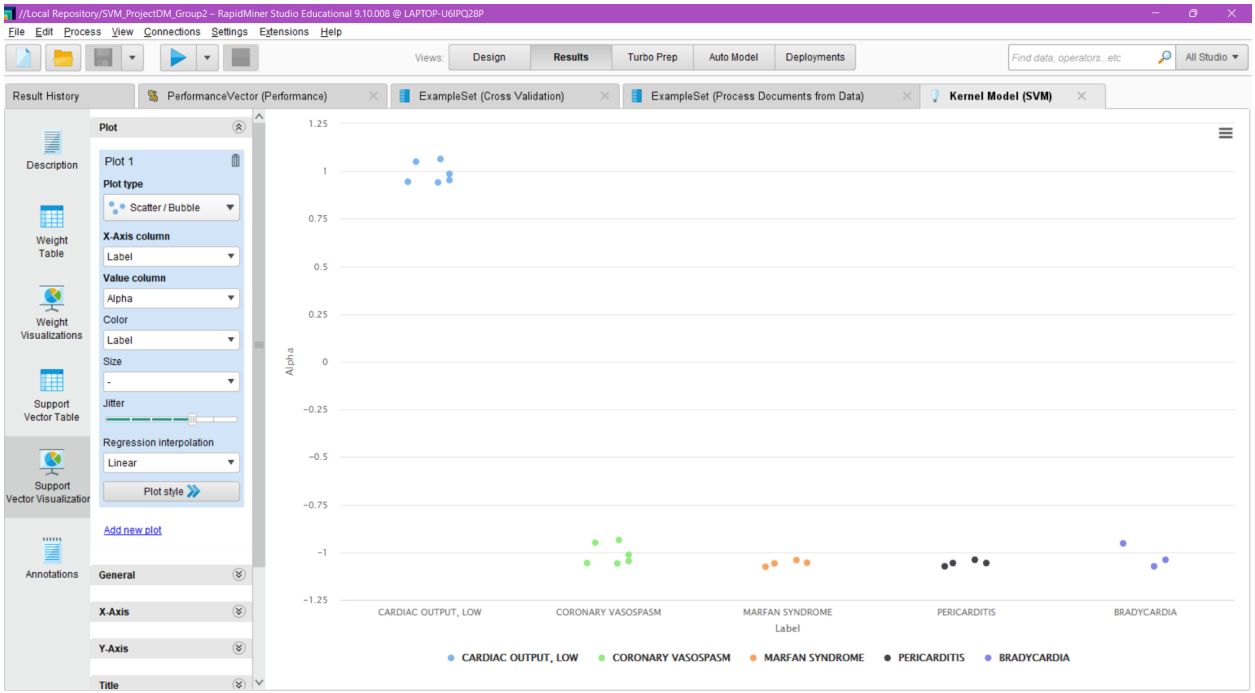


Figure 43: Results in Kernel Mode (SVM) by Support Vector Visualization

4.3 CLUSTERING (K-MEDOIDS)

Data mining techniques can be varied and distinguished from one another. One of the common techniques is clustering. According to Ester et al. (1996), class identification is strongly related to clustering algorithms. In this project, the clustering algorithm used is k-medoids algorithm. One of the examples of clustering algorithms that is unaffected by outliers or other extreme variables is the K-medoids algorithm (Atmaja, 2019). Therefore, that is why this algorithm is selected to be used in this project. Below are the steps on how we implement clustering for the given data set in RapidMiner.

1. Since the dataset has been saved inside the storage, operator “Retrieve” will be used to read the dataset.

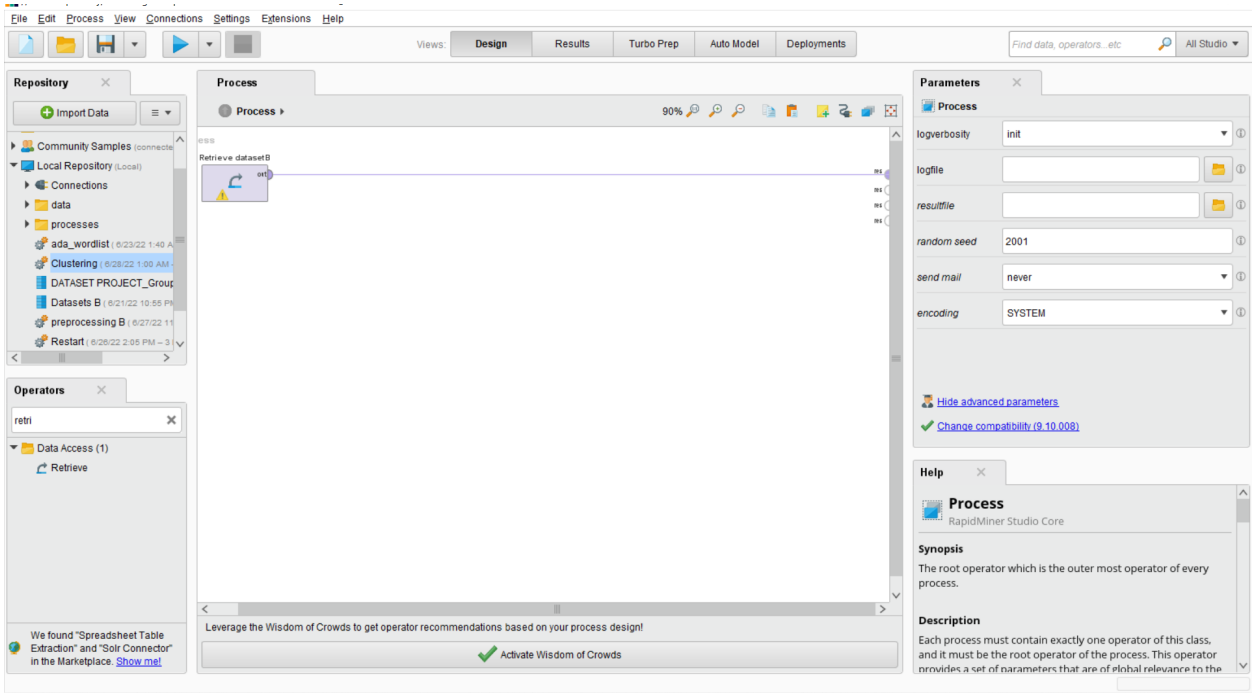


Figure 44: Retrieve DatasetB

2. Use the “Select Attribute” operator to discard the unwanted attribute.

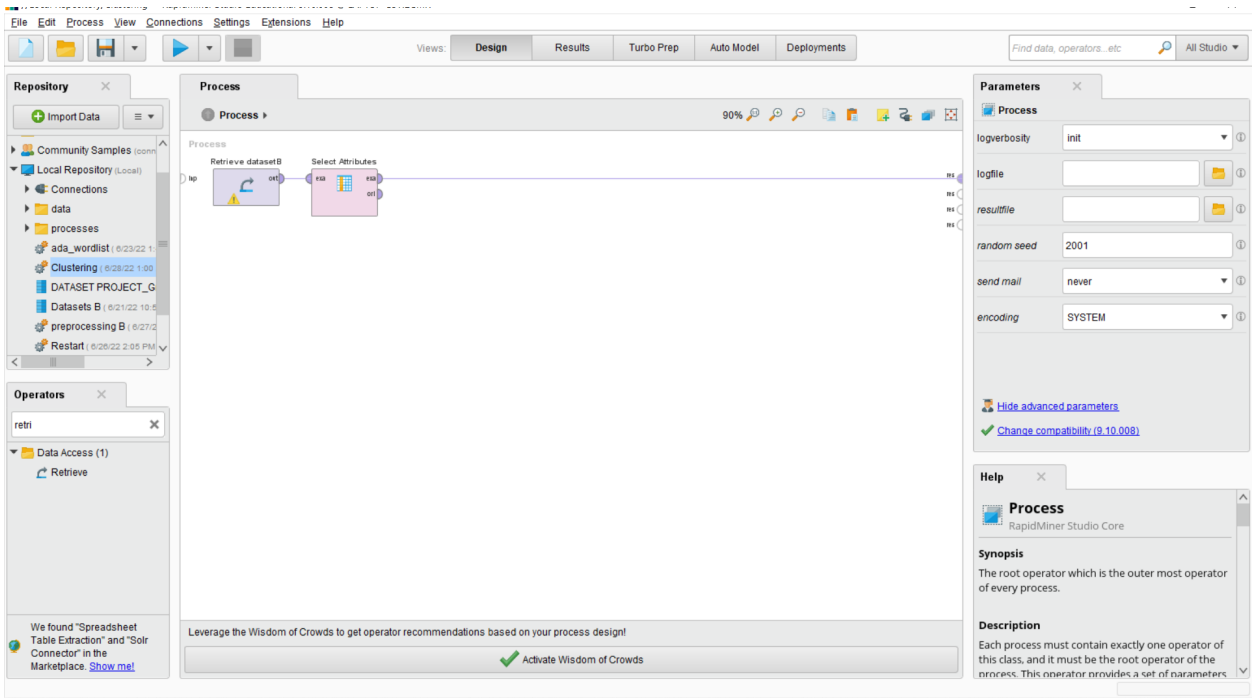


Figure 45: Select Attribute Operator

Click on the “Select Attribute” operator after dragging it into the workspace to see the parameter. Make sure that the parameters are set as shown below.

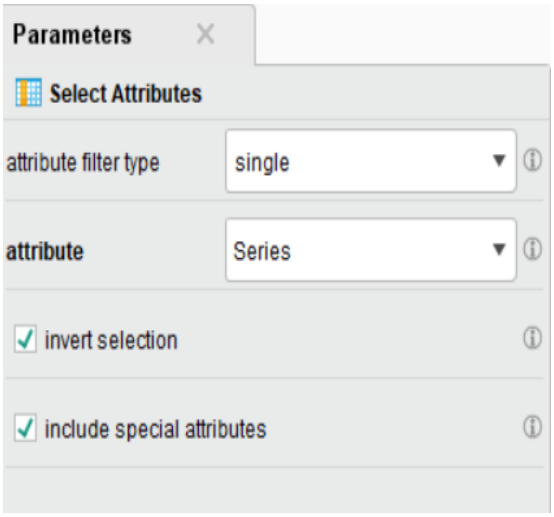


Figure 46: Select Attribute Parameter

3. Apply the “Nominal to Text” operator to change the data into string value.

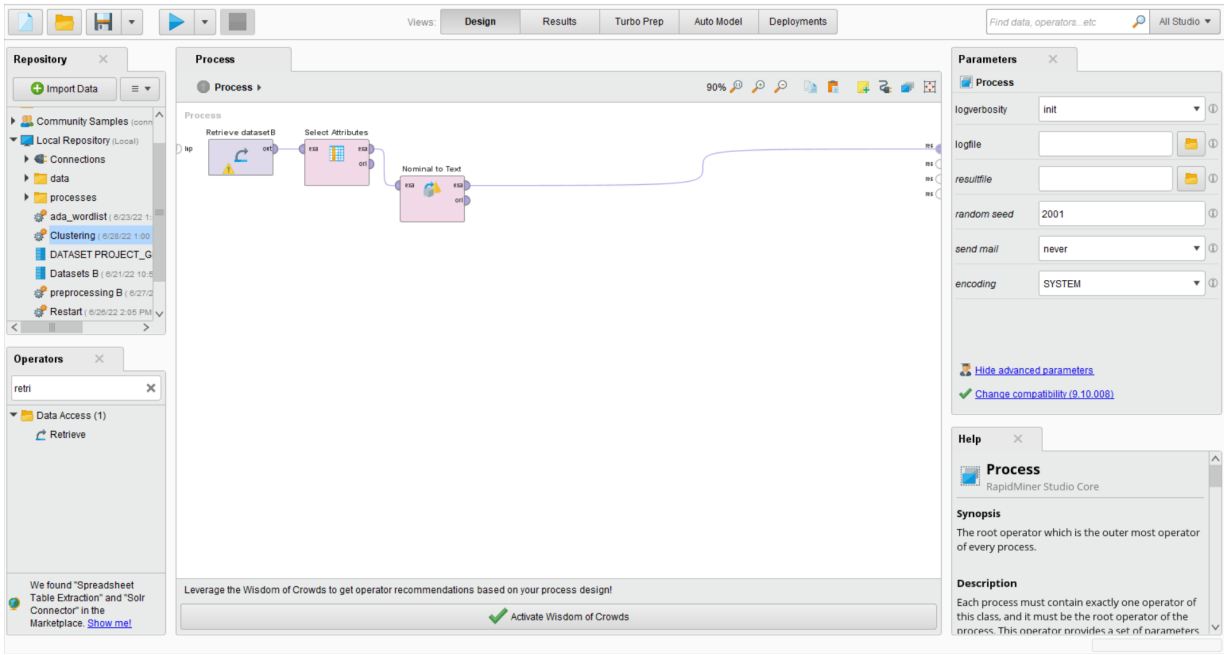


Figure 47: Apply Nominal to Text Operator

4. Add the “Process Documents from Data” operator to optimize the data used before performing clustering.

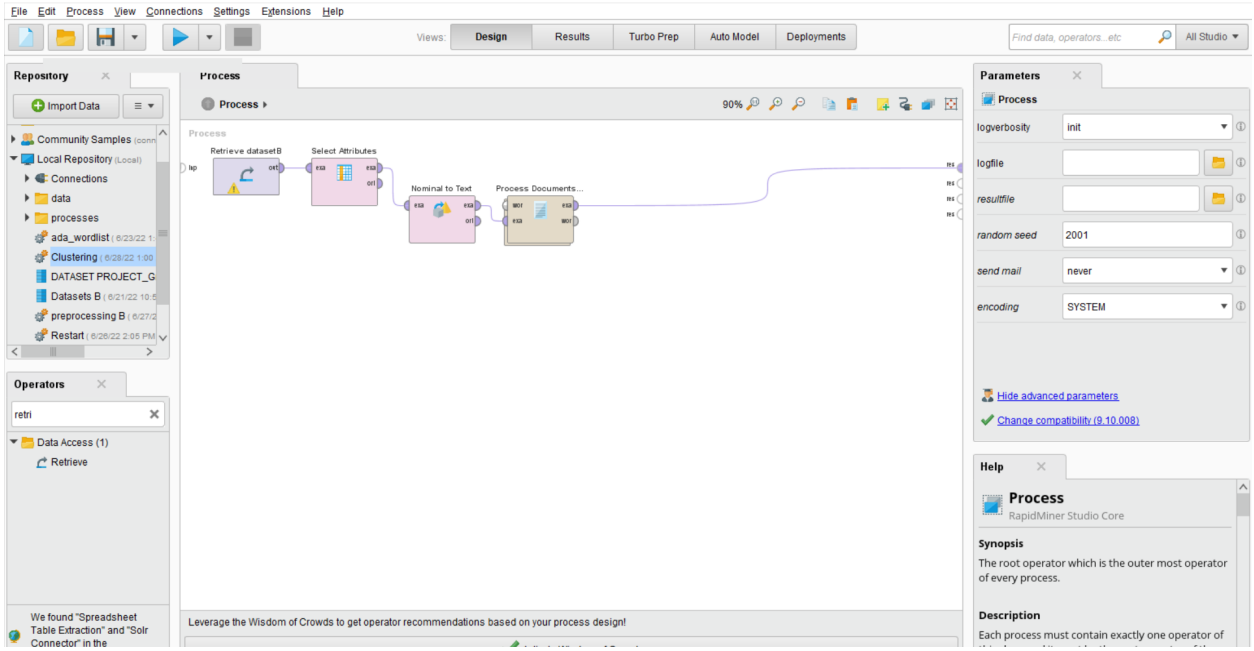


Figure 48: Add Process Documents from Data Operator

5. By double-clicking the “Process Documents from Data” operator, to optimize the data, we have added more operators inside it. There are six additional operators included inside which are ‘Tokenization’, ‘Filter Stopword (English)’, ‘Generate n-Grams (Terms)’, ‘Transform Cases’, ‘Filter Token (by Length)’ and ‘Stem (Lovins)’. Each operator has its own significance in optimizing the data. In example, ‘Tokenization’ is used to break down the text documents into each word. Removing unimportant words are necessary in this process and to do so, operator “Filter Stopword (English)’ is used. By using the ‘Generate n-Grams (Terms)’ operator, the pieces of words can be grouped as n successive items in the document that might include punctuations, words, numbers and symbols. ‘Transform Cases’ operator is used to standardize the spellings of the words. Then, to remove the excessive unwanted words from the documents, operators ‘Filter Token (by Length)’ and ‘Stem (Lovins)’ are applied in the workspace. Overall, the workspace inside the “Process Documents from Data” operator should look like below.

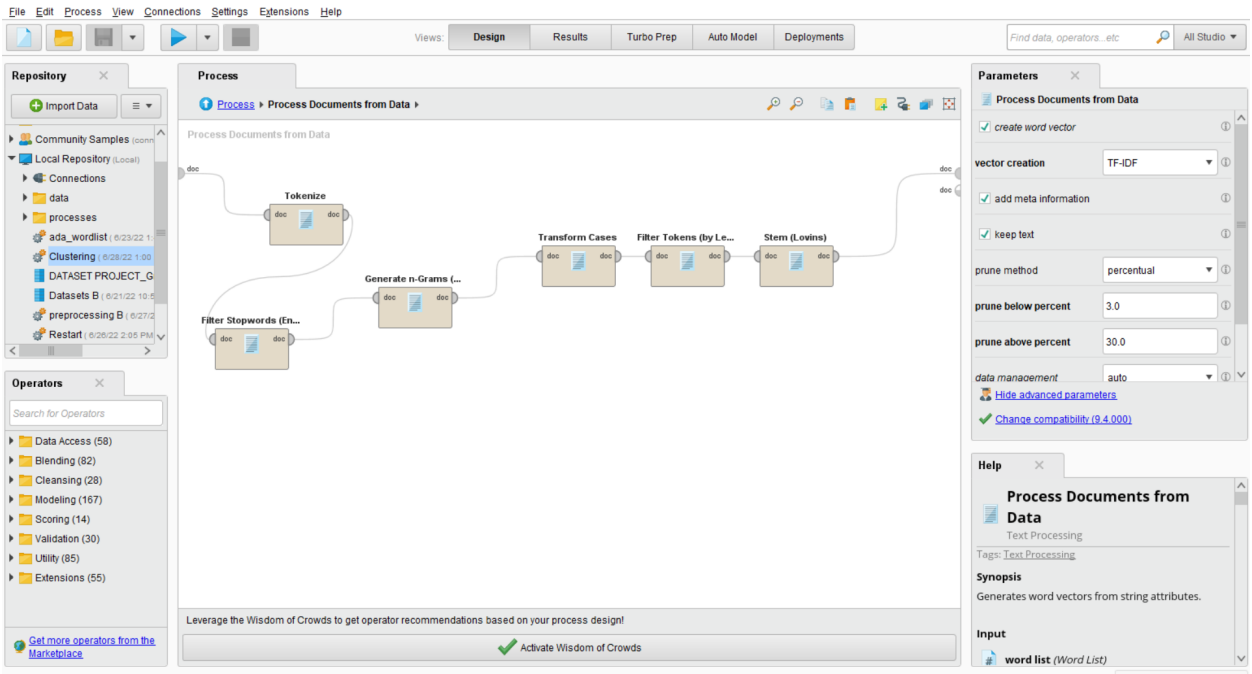


Figure 49: Process Documents from Data Operator Workspace

6. Next, we will put the **K-Medoids** operator from the "Segmentation" file to perform the clustering task. The parameters are set as shown below as well:

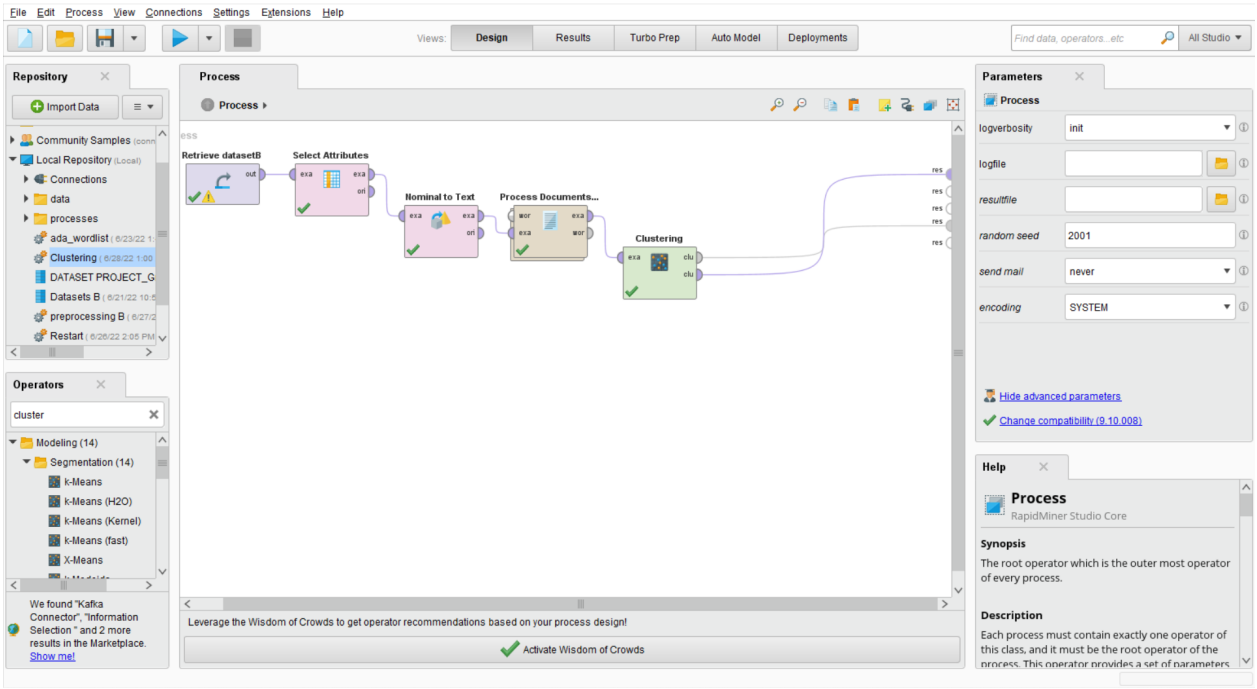


Figure 50: Add K-Medoids Operator

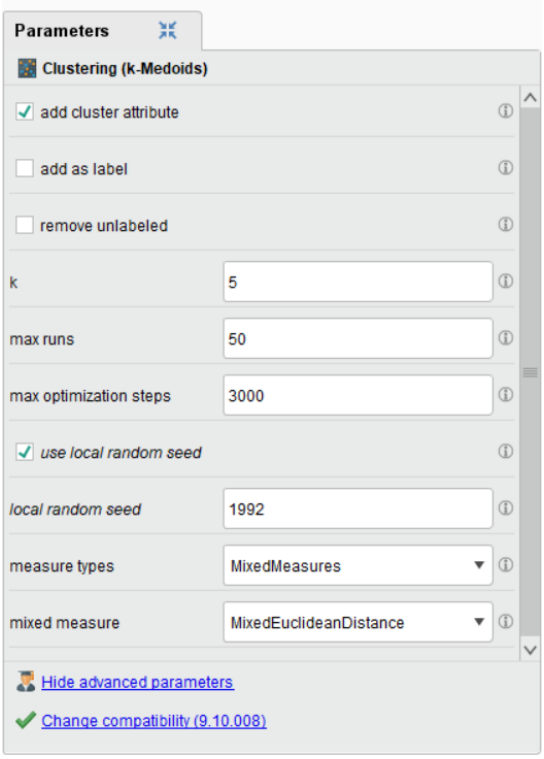


Figure 51: Set K-Medoids Operator Parameters

7. Then, add the “Multiply” operator to copy the of the clustering objects.

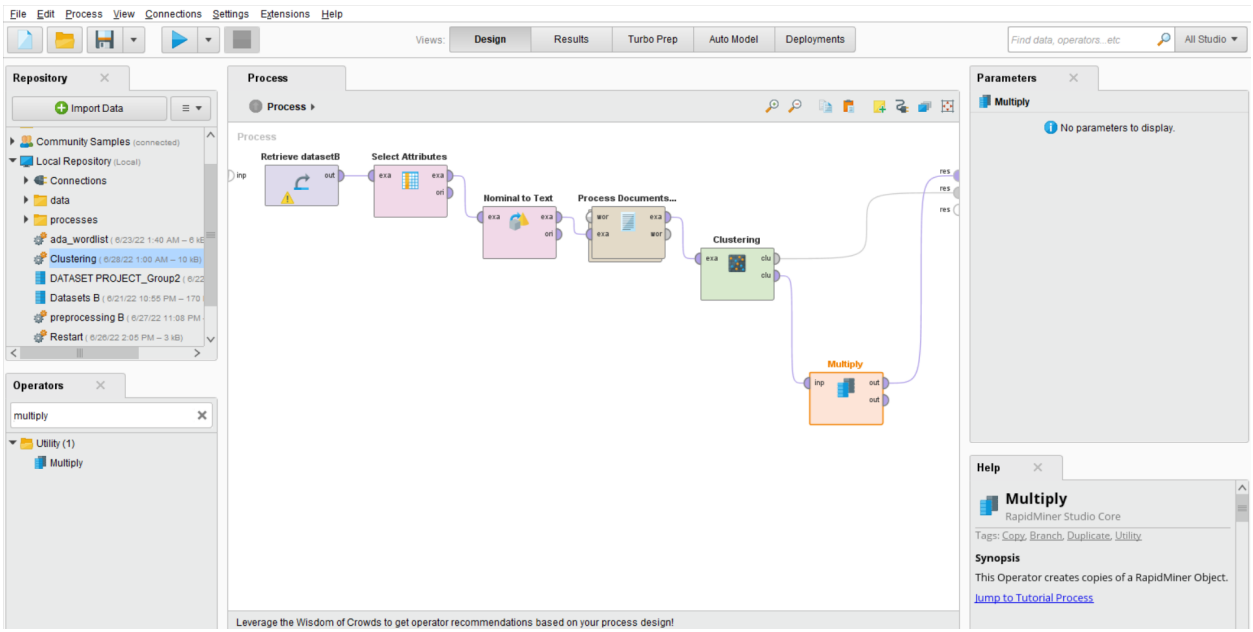


Figure 52: Add Multiply Operator

8. To get the visualization of the cluster made, we will add the “Cluster Model Visualizer” operator.

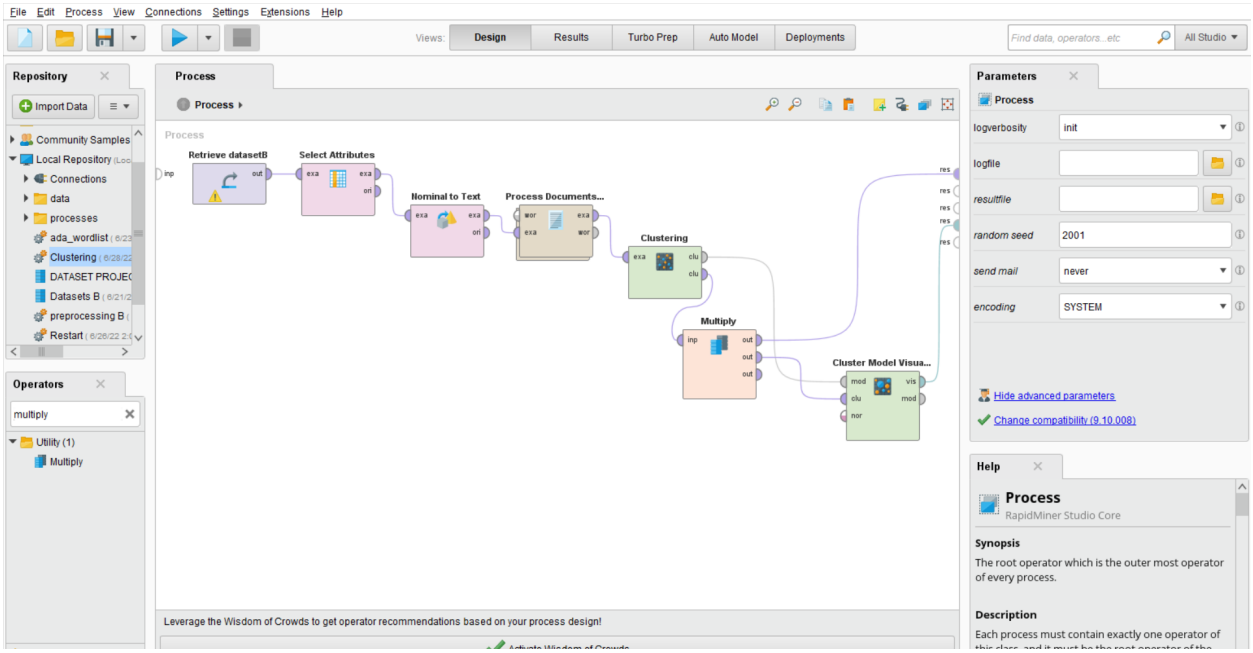


Figure 53: Add Cluster Model Visualizer Operator

9. Lastly, add the last operator which is the “Correlation Matrix” operator to correlations between all attributes and use these correlations to create a weights vector. A statistical method called correlation may be used to determine if and how strongly two pairs of attributes are connected.

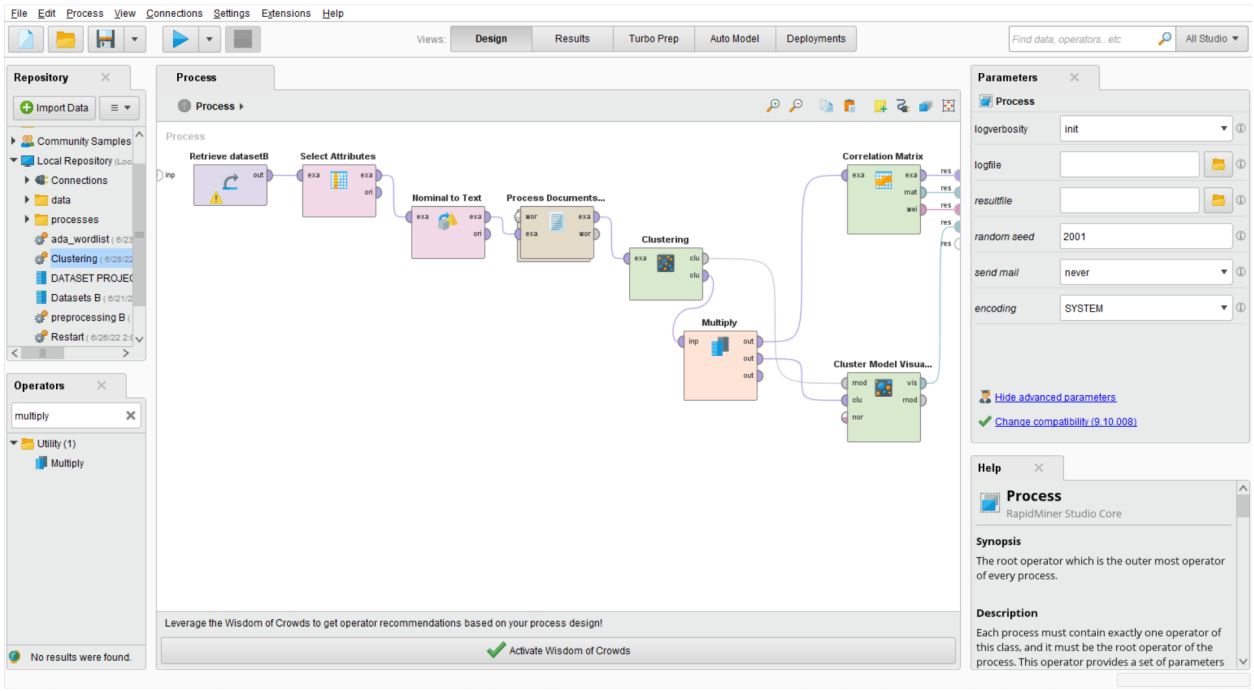


Figure 54: Add Correlation Matrix Operator

Based on these steps, below are the results that can be obtained and collected:

Open in Turbo Prep Auto Model Filter (23 / 23 examples): all

Row No.	id	cluster	text	abdomin	abdominal_...	abnorm	abnormal_a...	abnormal_c...	abnormal_s...	abnormalitie...	abnc
2	2	cluster_3	cardiac cardi...	0	0	0	0	0	0	0	0
3	3	cluster_3	cardiac cardi...	0	0	0	0	0	0	0	0
4	4	cluster_4	cardiac cardi...	0	0	0	0	0	0	0	0
5	5	cluster_4	cardiac cardi...	0	0	0	0	0	0	0	0
6	6	cluster_2	cardiac cardi...	0	0	0	0	0	0	0	0
7	7	cluster_2	coron corona...	0	0	0.051	0.059	0	0.059	0	0
8	8	cluster_2	coron corona...	0	0	0	0	0	0	0	0
9	9	cluster_2	coron corona...	0	0	0	0	0	0	0	0
10	10	cluster_2	coron corona...	0	0	0	0	0	0	0	0
11	11	cluster_3	coron corona...	0	0	0	0	0	0	0	0
12	12	cluster_2	coron corona...	0	0	0	0	0	0	0	0
13	13	cluster_3	marfan marfa...	0	0	0.041	0	0	0	0	0
14	14	cluster_3	marfan marfa...	0	0	0.036	0	0	0	0	0.08
15	15	cluster_3	marfan marfa...	0	0	0.079	0	0	0	0.061	0
16	16	cluster_1	marfan marfa...	0	0	0	0	0	0	0	0
17	17	cluster_0	pericardit peri...	0	0	0	0	0	0	0	0
18	18	cluster_0	pericardit dial...	0	0	0	0	0	0	0	0
19	19	cluster_4	pericardit valv...	0	0	0	0	0	0	0	0

Figure 55: Cluster Set List Result

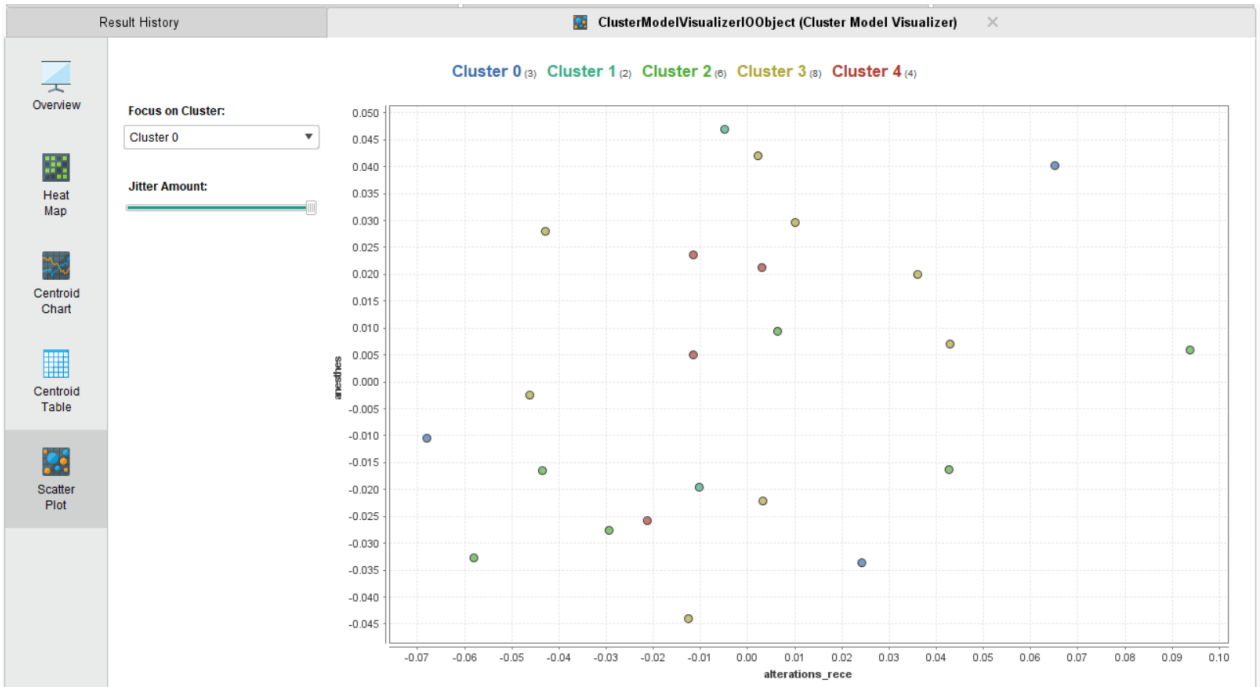


Figure 56: Scatter Plot from Cluster Model Visualizer

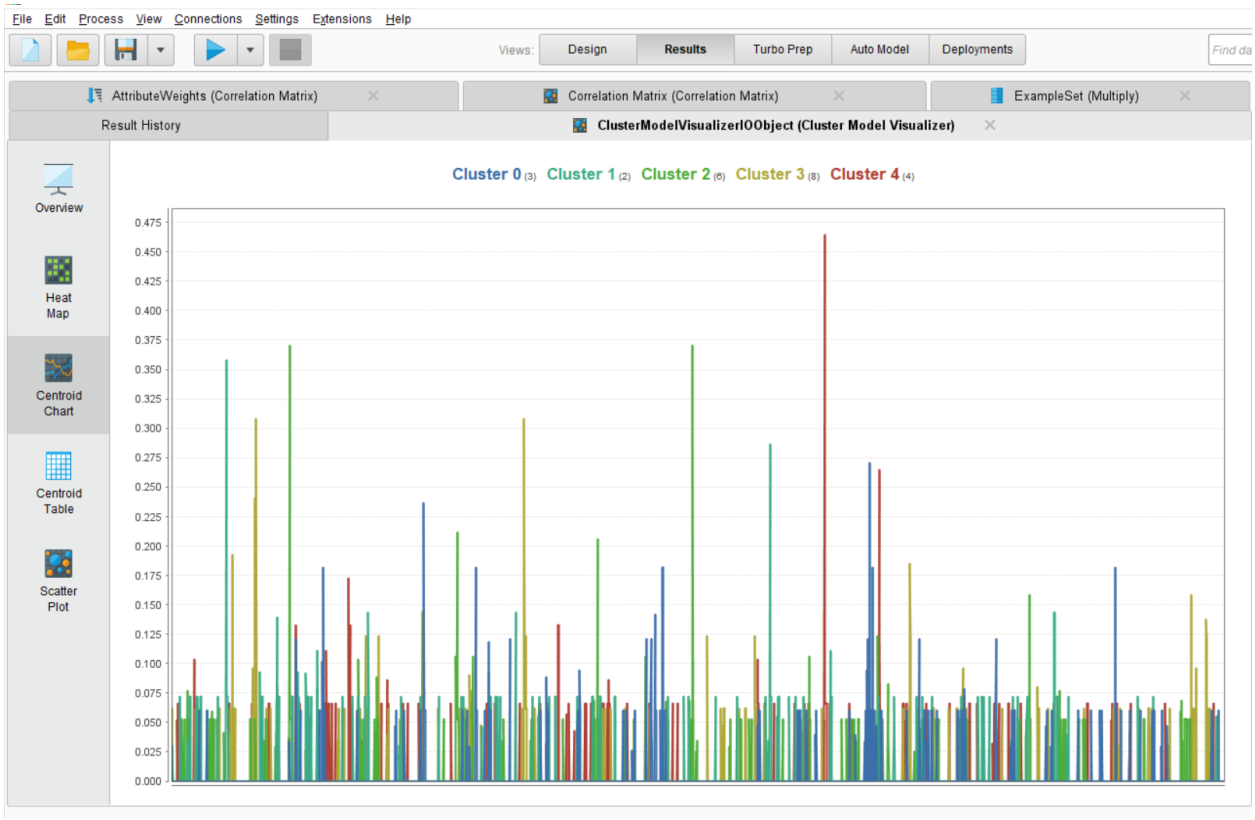
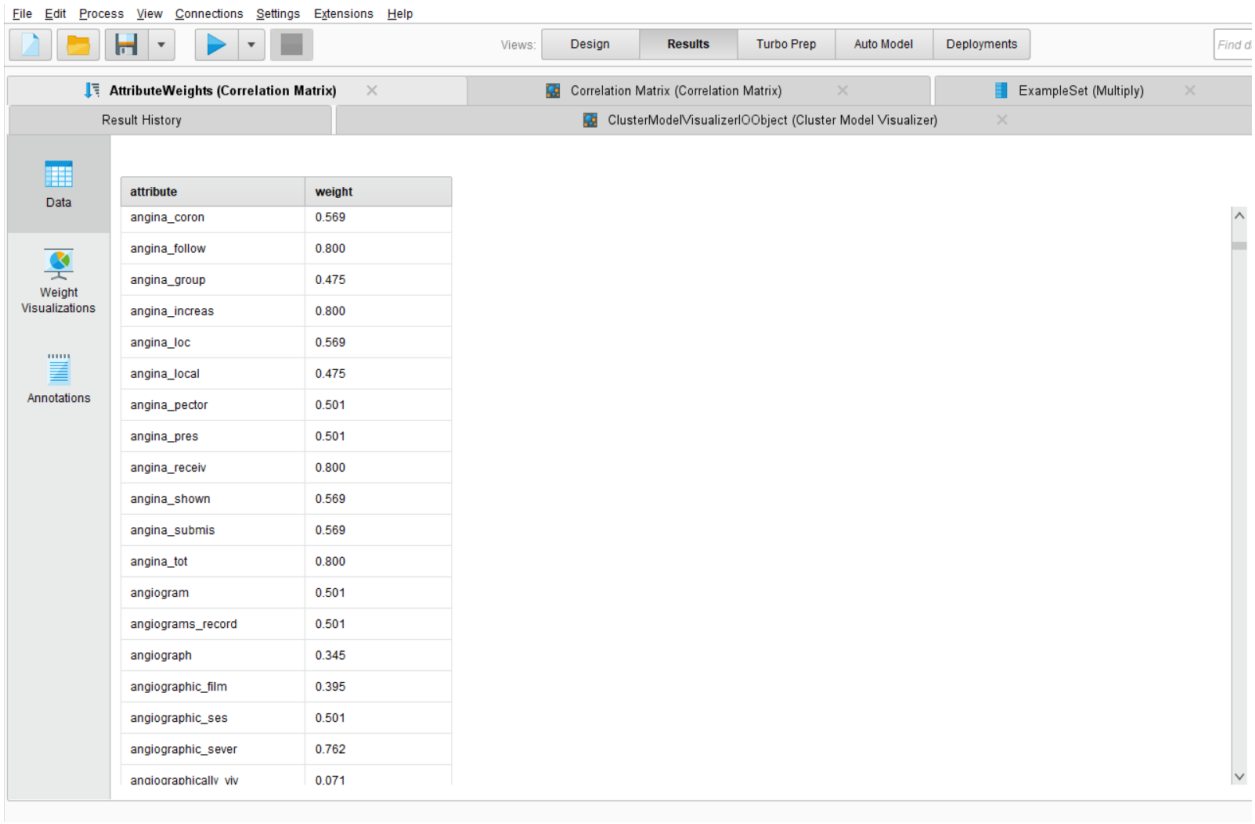
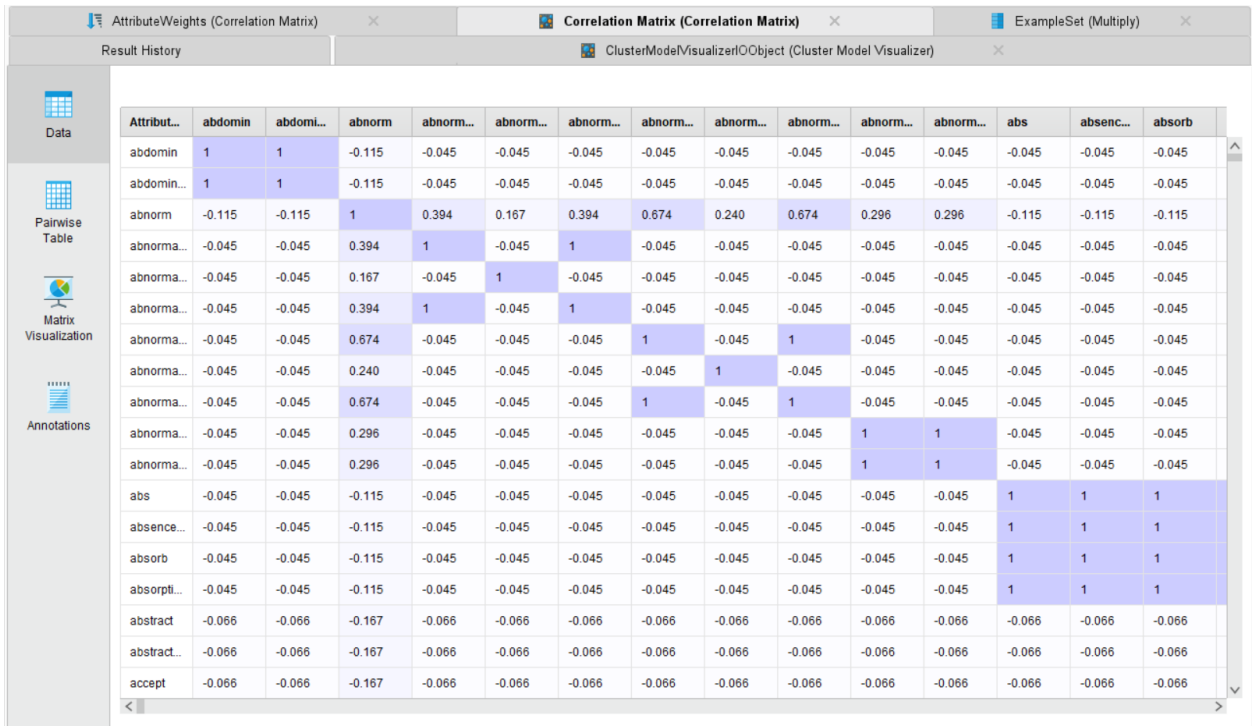


Figure 57: Centroid Chart from Cluster Model Visualizer



attribute	weight
angina_coron	0.569
angina_follow	0.800
angina_group	0.475
angina_increas	0.800
angina_loc	0.569
angina_local	0.475
angina_pector	0.501
angina_pres	0.501
angina_receiv	0.800
angina_shown	0.569
angina_submis	0.569
angina_tot	0.800
angiogram	0.501
angiograms_record	0.501
angiograph	0.345
angiographic_film	0.395
angiographic_ses	0.501
angiographic_sever	0.762
angiographically_viv	0.071

Figure 58: Attribute Weight (Correlation Matrix)



Attribut...	abdomin	abdomi...	abnorm	abnorm...	abnorm...	abnorm...	abnorm...	abnorm...	abnorm...	abnorm...	abnorm...	abs	absenc...	absorb
abdomin	1	1	-0.115	-0.045	-0.045	-0.045	-0.045	-0.045	-0.045	-0.045	-0.045	-0.045	-0.045	-0.045
abdomin...	1	1	-0.115	-0.045	-0.045	-0.045	-0.045	-0.045	-0.045	-0.045	-0.045	-0.045	-0.045	-0.045
abnorm	-0.115	-0.115	1	0.394	0.167	0.394	0.674	0.240	0.674	0.296	0.296	-0.115	-0.115	-0.115
abnorma...	-0.045	-0.045	0.394	1	-0.045	1	-0.045	-0.045	-0.045	-0.045	-0.045	-0.045	-0.045	-0.045
abnorma...	-0.045	-0.045	0.167	-0.045	1	-0.045	-0.045	-0.045	-0.045	-0.045	-0.045	-0.045	-0.045	-0.045
abnorma...	-0.045	-0.045	0.394	1	-0.045	1	-0.045	-0.045	-0.045	-0.045	-0.045	-0.045	-0.045	-0.045
abnorma...	-0.045	-0.045	0.674	-0.045	-0.045	-0.045	1	-0.045	1	-0.045	-0.045	-0.045	-0.045	-0.045
abnorma...	-0.045	-0.045	0.240	-0.045	-0.045	-0.045	-0.045	1	-0.045	-0.045	-0.045	-0.045	-0.045	-0.045
abnorma...	-0.045	-0.045	0.674	-0.045	-0.045	-0.045	1	-0.045	1	-0.045	-0.045	-0.045	-0.045	-0.045
abnorma...	-0.045	-0.045	0.296	-0.045	-0.045	-0.045	-0.045	-0.045	-0.045	1	1	-0.045	-0.045	-0.045
abnorma...	-0.045	-0.045	0.296	-0.045	-0.045	-0.045	-0.045	-0.045	-0.045	1	1	-0.045	-0.045	-0.045
abs	-0.045	-0.045	-0.115	-0.045	-0.045	-0.045	-0.045	-0.045	-0.045	-0.045	-0.045	1	1	1
absence...	-0.045	-0.045	-0.115	-0.045	-0.045	-0.045	-0.045	-0.045	-0.045	-0.045	-0.045	1	1	1
absorb	-0.045	-0.045	-0.115	-0.045	-0.045	-0.045	-0.045	-0.045	-0.045	-0.045	-0.045	1	1	1
absorpti...	-0.045	-0.045	-0.115	-0.045	-0.045	-0.045	-0.045	-0.045	-0.045	-0.045	-0.045	1	1	1
abstract	-0.066	-0.066	-0.167	-0.066	-0.066	-0.066	-0.066	-0.066	-0.066	-0.066	-0.066	-0.066	-0.066	-0.066
abstract...	-0.066	-0.066	-0.167	-0.066	-0.066	-0.066	-0.066	-0.066	-0.066	-0.066	-0.066	-0.066	-0.066	-0.066
accept	-0.066	-0.066	-0.167	-0.066	-0.066	-0.066	-0.066	-0.066	-0.066	-0.066	-0.066	-0.066	-0.066	-0.066

Figure 59: Correlation Matrix

4.4 ASSOCIATION (MARKET BASKET ANALYSIS)

Association is among the well-known data mining techniques, it discovers patterns based on the relationship between variables in the same transaction. It is also known as relation technique because it uses the relationship between items and discovers the frequent occurrence of different items that appear with the highest frequencies within the data set. Association rules use the if-then statements in order to show the probability of relationships between data items or variables within large data sets in various types of databases. Association rules have a number of applications and are widely used to help discover sales correlations in transactional data or in medical datasets. Association is widely used by retailers because it helps to understand the customer purchase behaviors (Osman, 2019). Below are the steps of how we applied association techniques into our project:

1. Read the excel file by using the "Read Excel" operator.

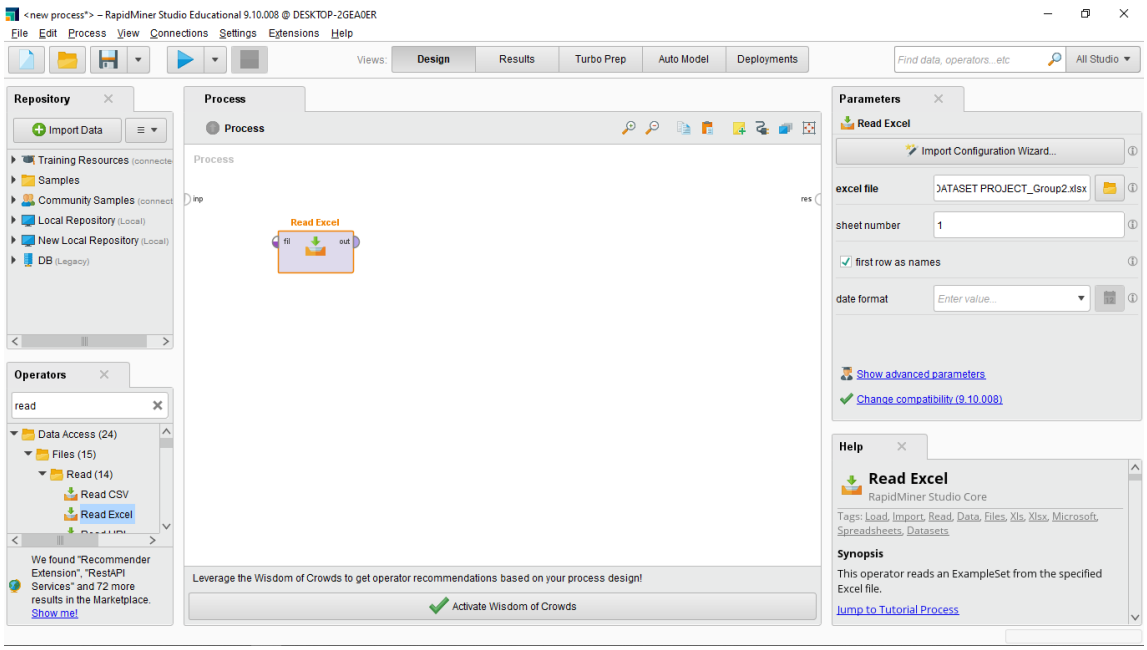


Figure 60: Read Excel Operator

2. Use the “Nominal to Text” operator in order to convert all of the data into string value. The process of data transforming can be more efficient by doing this process.

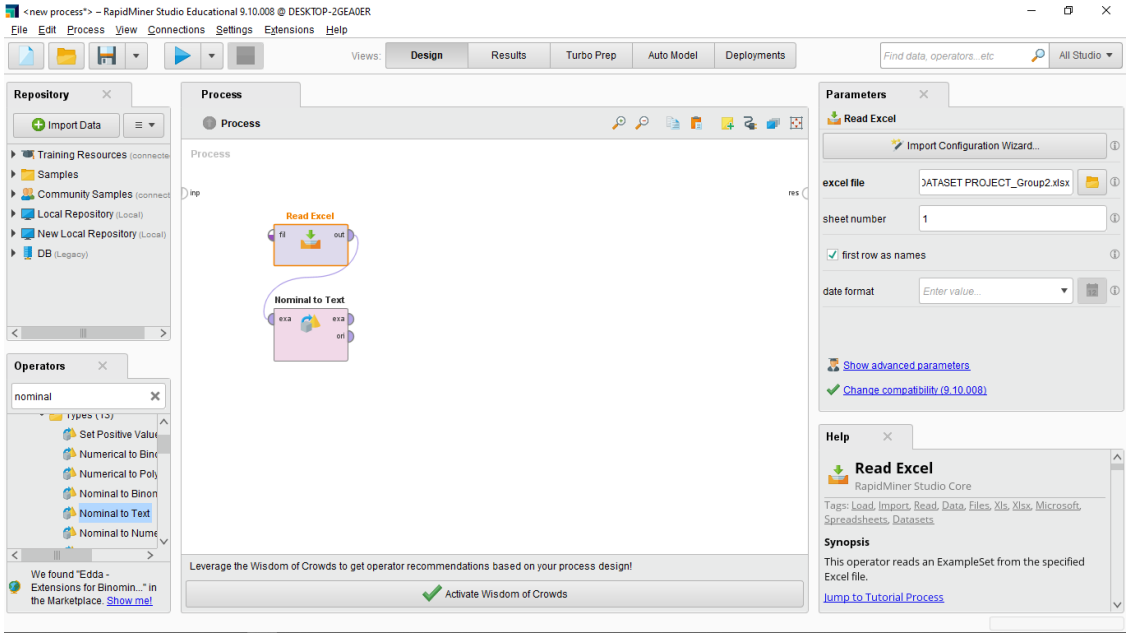


Figure 61: Nominal to Text Operator

3. Next, we add “Process Documents from Data”. The reason why we need to have this operator is because we want to process word vectors into string attributes based on the dataset. At the parameters, we need to set the vector creation into “Binary Term Occurrences” in order to get the value in binary (1 = exists and 0 = not exists) for us to understand it easily. By clicking on the “Process Documents from Data” operator, we can put any suitable operators inside it for the process to work well. Operators that we put inside it are “Extract Content” that help extracts textual content from a given document, “Tokenize” to splits the text of a document into a sequence of tokens, “Transform Cases” to transforms cases of characters in a document to lowercase, “Filter Stopwords (English)” to removes English stopwords from a document, and “Filter Tokens (by Length)” to filter tokens based on their length.

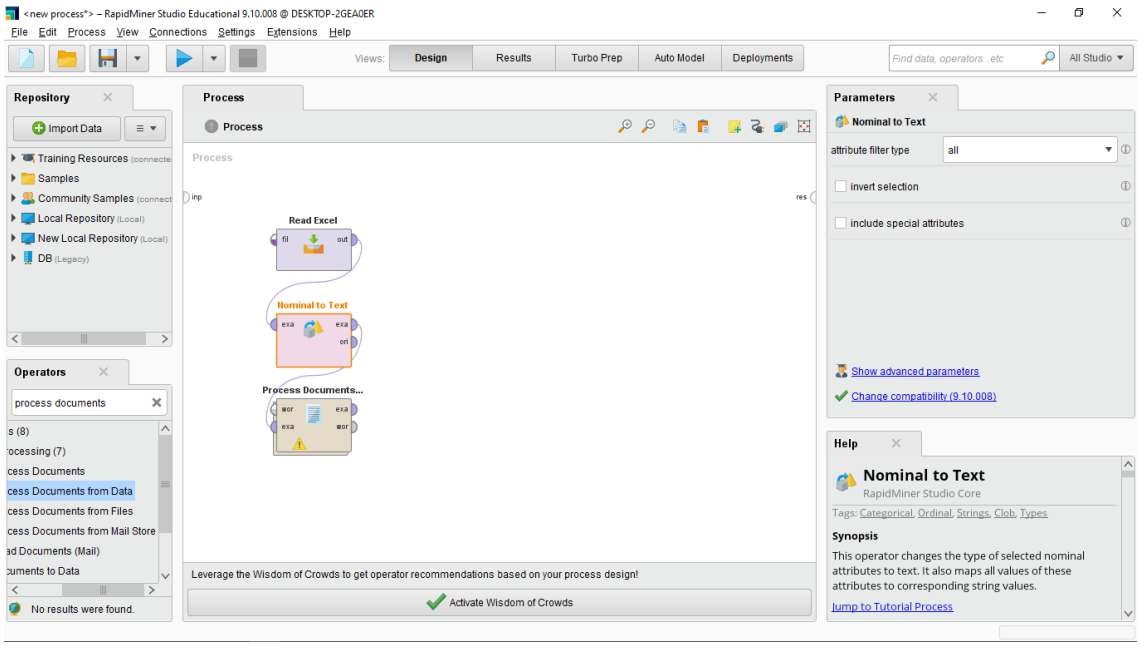


Figure 62: Process Documents from Data operator

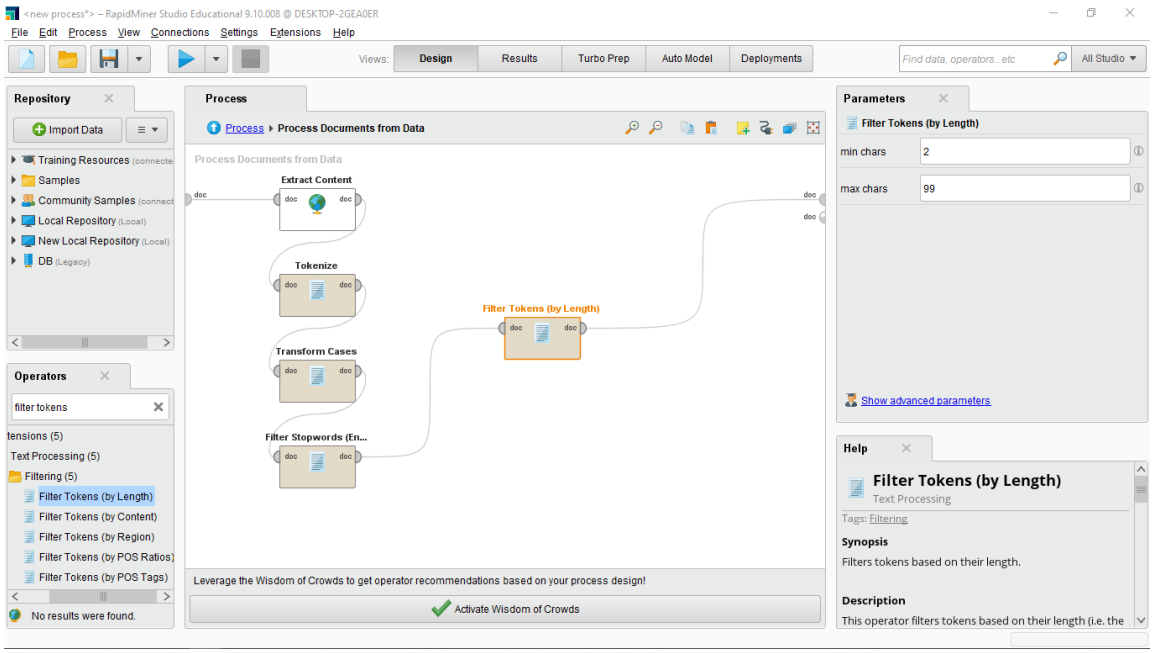


Figure 63: Operators used in Process Documents from Data operator

4. After that, we insert the "Numerical to Binomial" operator. This operator changes the type of the selected numeric attributes to a binomial type (1 = True and 0 = False).

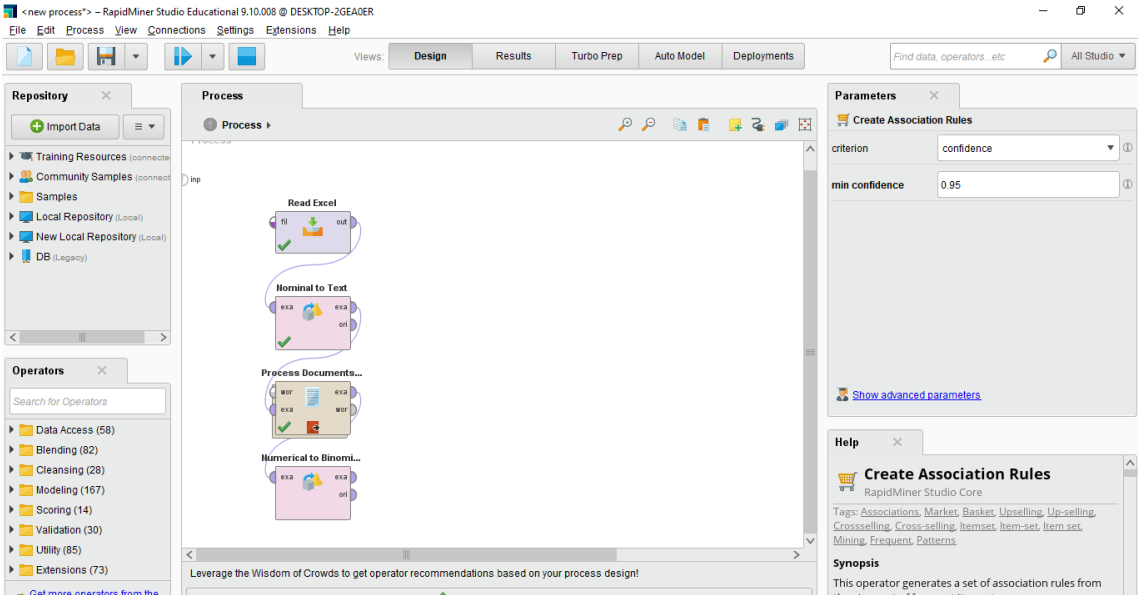


Figure 64: Numerical to Binomial operator

5. The next process is to insert the "FP-Growth" operator. This Operator efficiently calculates all frequently-occurring itemsets in a dataset, using the FP-tree data structure. In the parameters, the min support we set to 0.95.

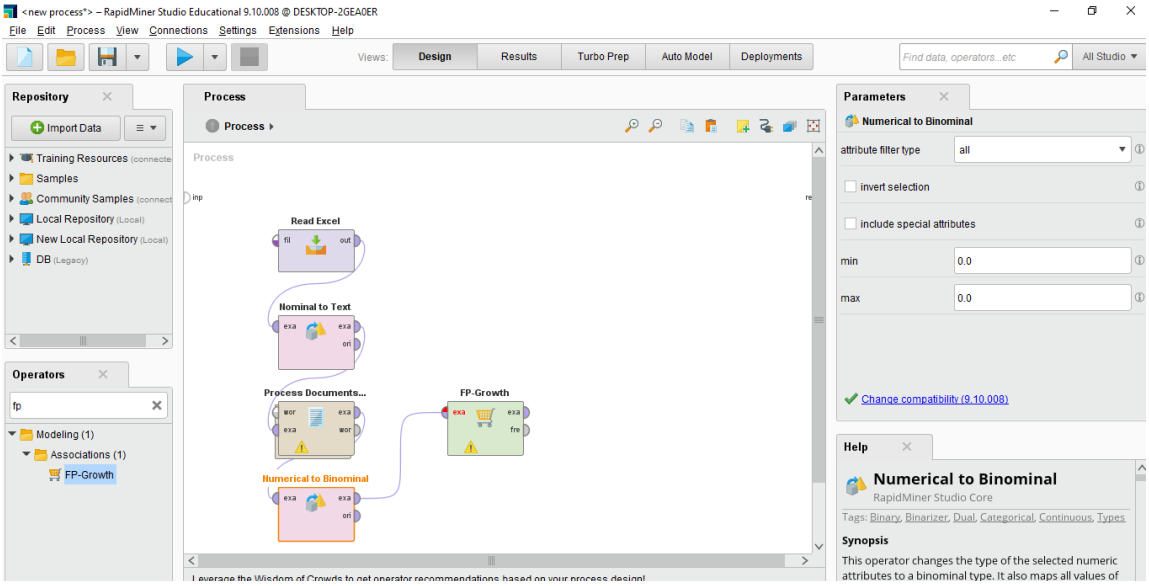


Figure 65: FP-Growth operator

6. The last operator used was "Create Association Rules". This operator generates a set of association rules from the given set of frequent itemsets. We set the min confidence equal to 0.95 in the parameters. This will be the last step before we run the design process algorithm.

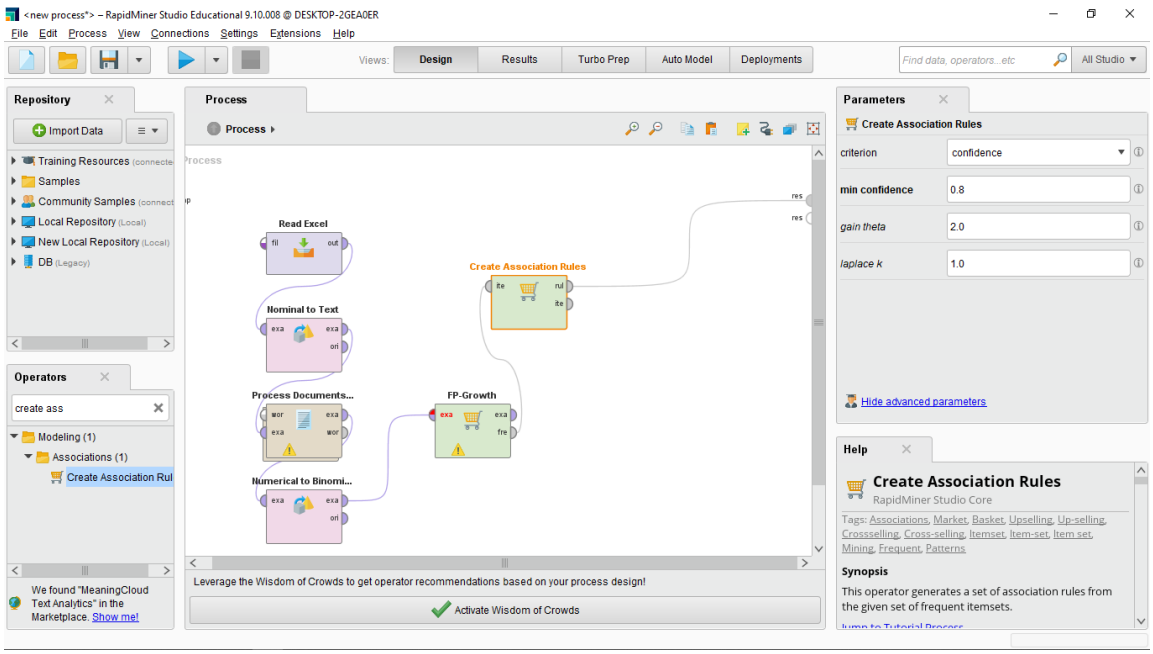


Figure 66: Create Association Rules operator

Below are the result for association task:

Row No.	text	abnormal	abnormalities	abstract	accurate	action	actuarial	acute	addition	administered	administrat
1	cardiac outpu...	1	0	0	0	0	0	0	0	0	0
ne data	cardiac outpu...	0	0	0	0	0	0	0	0	0	0
3	cardiac outpu...	0	0	0	1	0	0	0	0	0	0
4	cardiac outpu...	0	0	0	0	0	0	0	0	0	0
5	cardiac outpu...	0	0	0	0	0	1	0	0	0	0
6	cardiac outpu...	0	0	0	0	0	0	0	0	0	0
7	coronary vas...	1	0	0	0	1	0	0	0	1	0
8	coronary vas...	0	0	1	0	0	0	0	0	0	0
9	coronary vas...	0	0	0	0	1	0	0	0	1	1
10	coronary vas...	0	0	1	0	0	0	0	0	0	1
11	coronary vas...	0	0	0	0	0	0	0	0	0	0
12	coronary vas...	0	0	0	0	0	0	0	1	0	1
13	marfan syndr...	0	1	0	1	0	0	0	0	0	0
14	marfan syndr...	0	1	0	0	0	0	0	1	0	0
15	marfan syndr...	0	1	0	0	0	0	0	0	0	0
16	marfan syndr...	0	0	0	0	0	0	1	0	0	0

Figure 67: Result after run until "Process Documents from Data" operator

Row No.	text	abnormal	abnormalities	abstract	accurate	action	actuarial	acute	addition	administered	administrat
1	cardiac outpu...	true	false	false	false	false	false	false	false	false	false
2	cardiac outpu...	false	false	false	false	false	false	false	false	false	false
3	cardiac outpu...	false	false	false	true	false	false	false	false	false	false
4	cardiac outpu...	false	false	false	false	false	false	false	false	false	false
5	cardiac outpu...	false	false	false	false	false	true	false	false	false	false
6	cardiac outpu...	false	false	false	false	false	false	false	false	false	false
7	coronary vas...	true	false	false	false	true	false	false	false	true	false
8	coronary vas...	false	false	true	false	false	false	false	false	false	false
9	coronary vas...	false	false	false	false	true	false	false	false	true	true
10	coronary vas...	false	false	true	false	false	false	false	false	false	true
11	coronary vas...	false	false	false	false	false	false	false	false	false	false
12	coronary vas...	false	false	false	false	false	false	false	true	false	true
13	marfan syndr...	false	true	false	true	false	false	false	false	false	false
14	marfan syndr...	false	true	false	false	false	false	false	true	false	false
15	marfan syndr...	false	true	false	false	false	false	false	false	false	false
16	marfan syndr...	false	false	false	false	false	false	true	false	false	false

Figure 68: Result after “Numerical to Binomial” operator was added

No. of Sets: 61

Total Max. Size: 3

Min. Size: 1

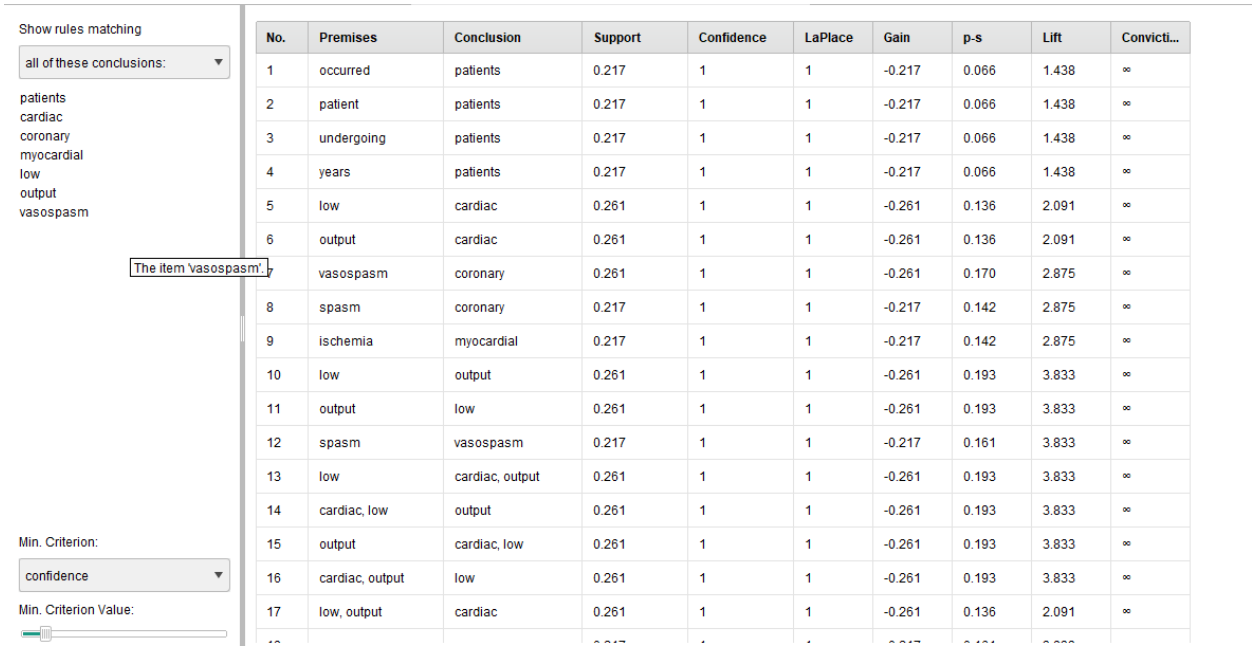
Max. Size: 3

Contains Item:

Update View

Size	Support	Item 1	Item 2	Item 3
1	0.696	patients		
1	0.478	cardiac		
1	0.435	normal		
1	0.348	coronary		
1	0.348	myocardial		
1	0.348	significant		
1	0.304	aortic		
1	0.304	heart		
1	0.304	percent		
1	0.261	control		
1	0.261	data		
1	0.261	low		
1	0.261	mean		
1	0.261	output		
1	0.261	time		
1	0.261	vasospasm		
1	0.261	ventricular		

Figure 69: Frequent Item Sets



No.	Premises	Conclusion	Support	Confidence	LaPlace	Gain	p-s	Lift	Convicti...
1	occurred	patients	0.217	1	1	-0.217	0.066	1.438	∞
2	patient	patients	0.217	1	1	-0.217	0.066	1.438	∞
3	undergoing	patients	0.217	1	1	-0.217	0.066	1.438	∞
4	years	patients	0.217	1	1	-0.217	0.066	1.438	∞
5	low	cardiac	0.261	1	1	-0.261	0.136	2.091	∞
6	output	cardiac	0.261	1	1	-0.261	0.136	2.091	∞
7	vasospasm	coronary	0.261	1	1	-0.261	0.170	2.875	∞
8	spasm	coronary	0.217	1	1	-0.217	0.142	2.875	∞
9	ischemia	myocardial	0.217	1	1	-0.217	0.142	2.875	∞
10	low	output	0.261	1	1	-0.261	0.193	3.833	∞
11	output	low	0.261	1	1	-0.261	0.193	3.833	∞
12	spasm	vasospasm	0.217	1	1	-0.217	0.161	3.833	∞
13	low	cardiac, output	0.261	1	1	-0.261	0.193	3.833	∞
14	cardiac, low	output	0.261	1	1	-0.261	0.193	3.833	∞
15	output	cardiac, low	0.261	1	1	-0.261	0.193	3.833	∞
16	cardiac, output	low	0.261	1	1	-0.261	0.193	3.833	∞
17	low, output	cardiac	0.261	1	1	-0.261	0.136	2.091	∞

Figure 70: Result of the Association Rules

AssociationRules

```
Association Rules
[occurred] --> [patients] (confidence: 1.000)
[patient] --> [patients] (confidence: 1.000)
[undergoing] --> [patients] (confidence: 1.000)
[years] --> [patients] (confidence: 1.000)
[low] --> [cardiac] (confidence: 1.000)
[output] --> [cardiac] (confidence: 1.000)
[vasospasm] --> [coronary] (confidence: 1.000)
[spasm] --> [coronary] (confidence: 1.000)
[ischemia] --> [myocardial] (confidence: 1.000)
[low] --> [output] (confidence: 1.000)
[output] --> [low] (confidence: 1.000)
[spasm] --> [vasospasm] (confidence: 1.000)
[low] --> [cardiac, output] (confidence: 1.000)
[cardiac, low] --> [output] (confidence: 1.000)
[output] --> [cardiac, low] (confidence: 1.000)
[cardiac, output] --> [low] (confidence: 1.000)
[low, output] --> [cardiac] (confidence: 1.000)
[spasm] --> [coronary, vasospasm] (confidence: 1.000)
[coronary, spasm] --> [vasospasm] (confidence: 1.000)
[vasospasm, spasm] --> [coronary] (confidence: 1.000)
```

Figure 71: Association Rules with confidence value

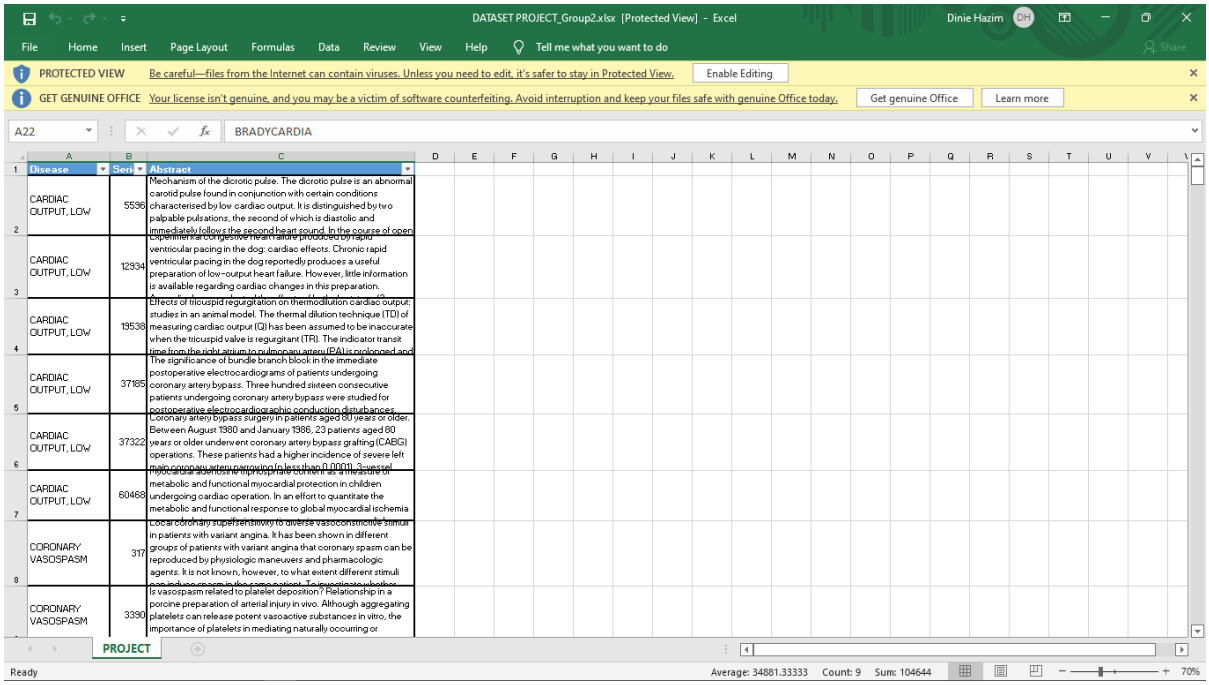
5.0 CONCLUSION

Based on the project we had conducted, in conclusion, we have gained extra knowledge on how to handle the real data mining tasks as we used the dataset given to us. Not only had we learned on the topic of data mining itself, we also had some experience on how to work with RapidMiner software that was assigned to us by the lecturer. We have implemented four data mining tasks in this project which includes classification, regression and prediction analysis, clustering analysis and association analysis. For classification, we chose to use the Naive Bayes Algorithm and we have obtained 39.13% of accuracy when performing the classification. For regression and prediction analysis, we used the Support Vector Machine (SVM) algorithm for the given dataset. Then, we used the K-Medoids algorithm for clustering analysis so that we can get better observation of the data inside the clusters formed and FP-Growth algorithm for association analysis to find the patterns based on the relationships.

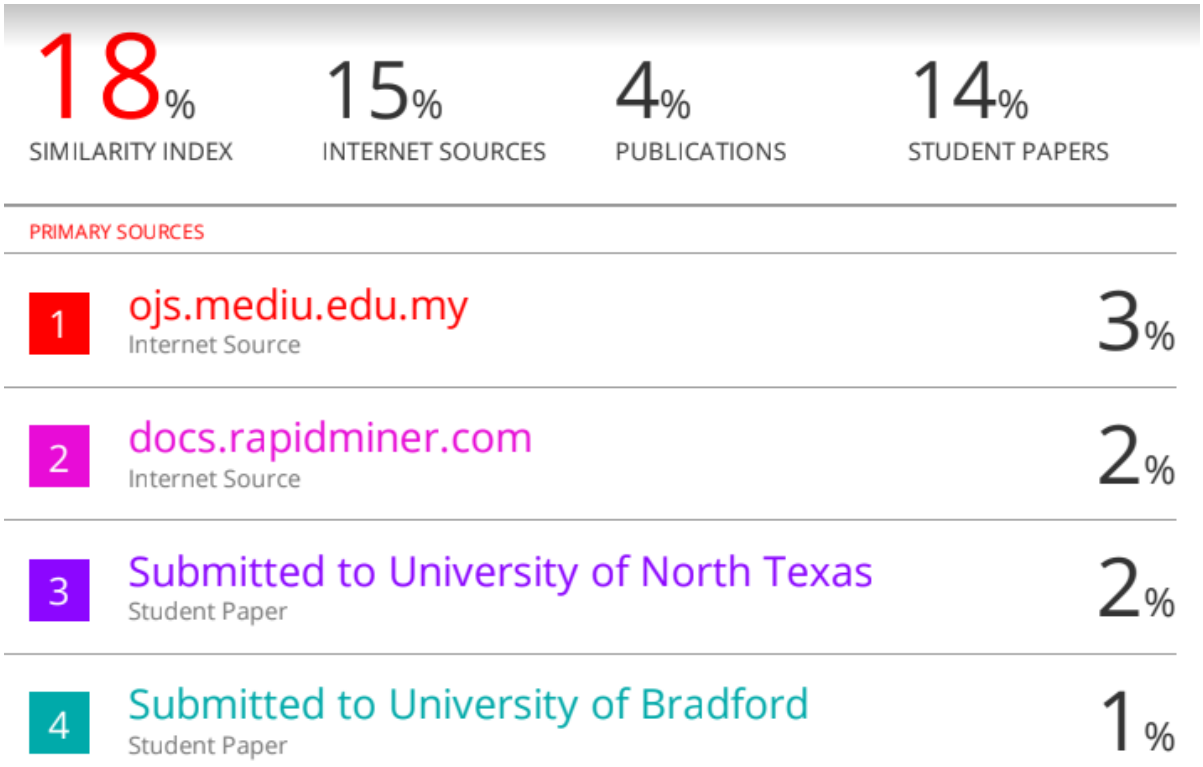
Since this project requires us to be involved with the data mining tool itself, we know that all of the new knowledge we learned during the progress of this project will be useful to us in the future as well as it will increase our ability to think critically as we analyze the data. These knowledge can also be applied by us during our industrial training in the future which makes the knowledge learnt precious and valuable. Through this project, we were able to improve our data mining abilities while also getting to know the software that can be utilized for data mining, which will come in handy in the future.

6.0 APPENDIX

Appendix 1: Dataset B in Excel File



Appendix 2: Plagiarism checker



7.0 REFERENCES

Association rule analysis | text mining | RapidMiner Studio. (2019, January 31). RapidMiner Academy. Retrieved June 27, 2022, from https://academy.rapidminer.com/learn/video/text-association-rules

Atmaja, E. H. S. (2019). Implementation of k-Medoids Clustering Algorithm to Cluster Crime Patterns in Yogyakarta. International Journal of Applied Sciences and Smart Technologies, 1(1), 33-44.

Clifton, C. (2022, February 14). *data mining | computer science | Britannica*. Encyclopedia Britannica. Retrieved June 22, 2022, from https://www.britannica.com/technology/data-mining

Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In kdd (Vol. 96, No. 34, pp. 226-231).

Osman, A. S. (2019). Data mining techniques.

Raval, K. M. (2012). Data mining techniques. International Journal of Advanced Research in Computer Science and Software Engineering, 2(10).