FACULTY OF COMPUTING

SEMESTER 2 /20222023

SECP2753-01 DATA MINING

GROUP PROJECT - KNIME

LECTURER: ROZILAWATI BINTI DOLLAH @ MD ZAIN

VIDEO PRESENTATION : https://www.youtube.com/watch?v=U-MUFKWjqXw

GROUP: 08

NAME	MATRICS ID
IZZAT HAQEEMI BIN HAIRUDIN	A21EC0033
MUHAMMAD FIKRI BIN SHARUNAZIM	A21EC0075
MUHAMMAD FARHAN BIN IBRAHIM	A21EC0072
MUHAMMAD ADAM FAHMI BIN MOHD TAUFIQ	A21EC0061

Table of content

1.0 Introduction

The data mining tasks can be classified generally into two types based on what a specific task tries to achieve. Those two categories are descriptive tasks and predictive tasks. The descriptive data mining tasks characterize the general properties of data whereas predictive data mining tasks perform inference on the available data set to predict how a new data set will behave.

There are a number of data mining tasks such as classification, prediction, association, clustering, summarization and many more. All these tasks are either predictive data mining tasks or descriptive data mining tasks. A data mining system can execute one or more of the above specified tasks as part of data mining. Predictive data mining tasks come up with a model from the available data set that is helpful in predicting unknown or future values of another data set of interest. A medical practitioner trying to diagnose a disease based on the medical test results of a patient can be considered as a predictive data mining task. Descriptive data mining tasks usually find data describing patterns and come up with new, significant information from the available data set. A retailer trying to identify products that are purchased together can be considered as a descriptive data mining task.

- Classification derives a model to determine the class of an object based on its attributes. A collection of records will be available, each record with a set of attributes. One of the attributes will be a class attribute and the goal of the classification task is assigning a class attribute to a new set of records as accurately as possible.
- Association discovers the association or connection among a set of items. Association identifies the relationships between objects. Association analysis is used for commodity management, advertising, catalog design, direct marketing etc. A retailer can identify the products that normally customers purchase together or even find the customers who respond to the promotion of the same kind of products. If a retailer finds that soda and cola are bought together mostly, he can put colas on sale to promote the sale of soda.
- Clustering is used to identify data objects that are similar to one another. The similarity can be decided based on a number of factors like purchase behavior, responsiveness to certain actions, geographical locations and so on. For example, an insurance company can cluster its customers based on age, residence, income etc. This group information will

be helpful to understand the customers better and hence provide better customized services.

- Prediction task predicts the possible values of missing or future data. Prediction involves developing a model based on the available data and this model is used in predicting future values of a new data set of interest. For example, a model can predict the income of an employee based on education, experience and other demographic factors like place of stay, gender and so on. Also prediction analysis is used in different areas including medical diagnosis, fraud detection and many more.

Explain dataset

First dataset: **bank-full.csv**

Input Variables:

1. Age: The age of the client (numeric).
2. Job: The type of job the client has (categorical).
3. Marital: The marital status of the client (categorical).
4. Education: The educational background of the client (categorical).
5. Default: Indicates whether the client has credit in default (categorical).
6. Housing: Indicates whether the client has a housing loan (categorical).
7. Loan: Indicates whether the client has a personal loan (categorical).
8. Contact: The communication type used to contact the client (categorical).
9. Month: The month of the last contact (categorical).
10. Day_of_week: The day of the week of the last contact (categorical).
11. Duration: The duration of the last contact in seconds (numeric).
12. Campaign: The number of contacts performed during the current campaign for this client (numeric).
13. Pdays: The number of days that passed since the client was last contacted from a previous campaign (numeric; 999 means the client was not previously contacted).
14. Previous: The number of contacts performed before the current campaign for this client (numeric).
15. Poutcome: The outcome of the previous marketing campaign (categorical).

16. Y: Indicates whether the client has subscribed to a term deposit (binary: 'yes', 'no').

The dataset provides a wide range of information about the clients, their background, previous interactions, and the economic context in which the marketing campaigns took place. These features can be used to build a predictive model to determine the likelihood of a client subscribing to a term deposit.

Second dataset: **groceriesItem.csv**

Input Variables:

1. Item(s) : The number of items present in a basket.
2. Item 1 - Item 32 : Indicates the name of the fruit

The dataset you have consists of 20 columns labeled as "item 1" to "item 32" and 9835 transactions. Each column represents the name of a fruit. The dataset focuses on the content of fruit baskets, which suggests that each row in the dataset represents a specific fruit basket.

The dataset likely contains information about the composition of fruit baskets, indicating which fruits are present in each basket. The values in the dataset's rows would indicate the presence or absence of specific fruits in each corresponding fruit basket.

With 32 columns representing different fruits, the dataset allows for a wide range of fruit combinations and possibilities. By analyzing the dataset, you can gain insights into the patterns, associations, or relationships between different fruits and potentially identify commonly occurring combinations of fruits in the fruit baskets.

Third dataset: **Boston.csv**

The dataset Boston.csv contains information about different houses in Boston. There are 506 samples and 13 feature variables in this dataset. This is the data dictionary for the dataset Boston.csv.

- **CRIM**: Per capita crime rate by town
- **ZN**: Proportion of residential land zoned for lots over 25,000 sq. ft
- **INDUS**: Proportion of non-retail business acres per town
- **CHAS**: Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
- **NOX**: Nitric oxide concentration (parts per 10 million)
- **RM**: Average number of rooms per dwelling
- **AGE**: Proportion of owner-occupied units built prior to 1940
- **DIS**: Weighted distances to five Boston employment centers
- **RAD**: Index of accessibility to radial highways
- **TAX**: Full-value property tax rate per $10,000
- **PTRATIO**: Pupil-teacher ratio by town
- **B**: $1000(Bk - 0.63)^2$, where Bk is the proportion of [people of African American descent] by town
- **LSTAT**: Percentage of lower status of the population
- **MEDV**: Median value of owner-occupied homes in $1000s

The prices of the house indicated by the variable **MEDV** is our target variable and the remaining are the feature variables based on which we will predict the value of the house.

1.1 Data Preprocessing

1.1.1 Dataset 1: bank-full.csv (Classification - Decision Tree)

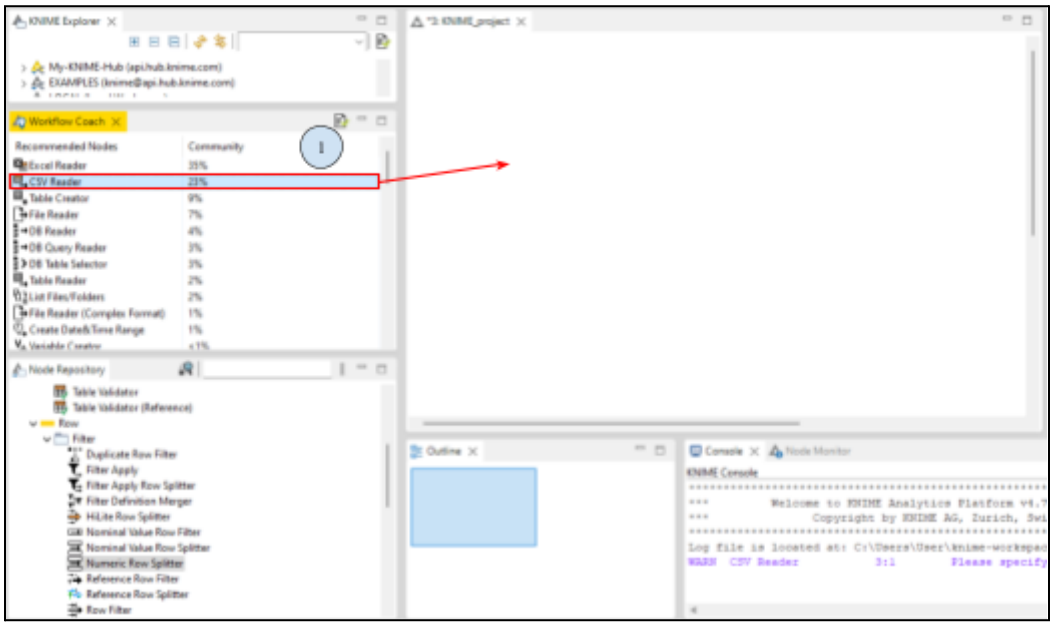


Figure 1.1.1.1

1. Drag the "CSV Reader" component at Workflow Coach to the work area to insert the dataset in the .csv type of file.

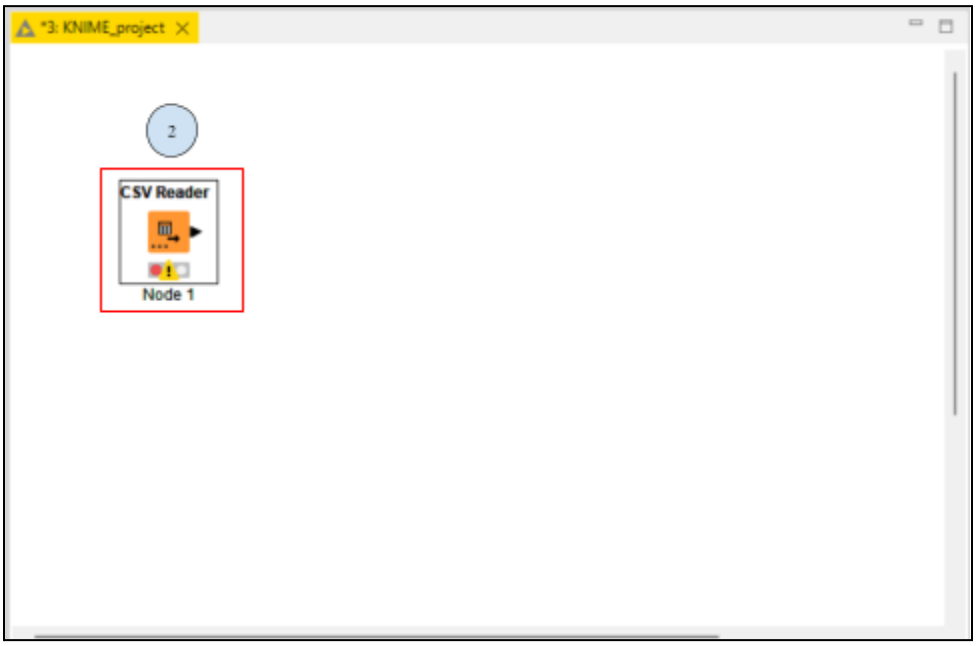


Figure 1.1.1.2

2. Double click on the CSV Reader node and then a new window will open automatically.

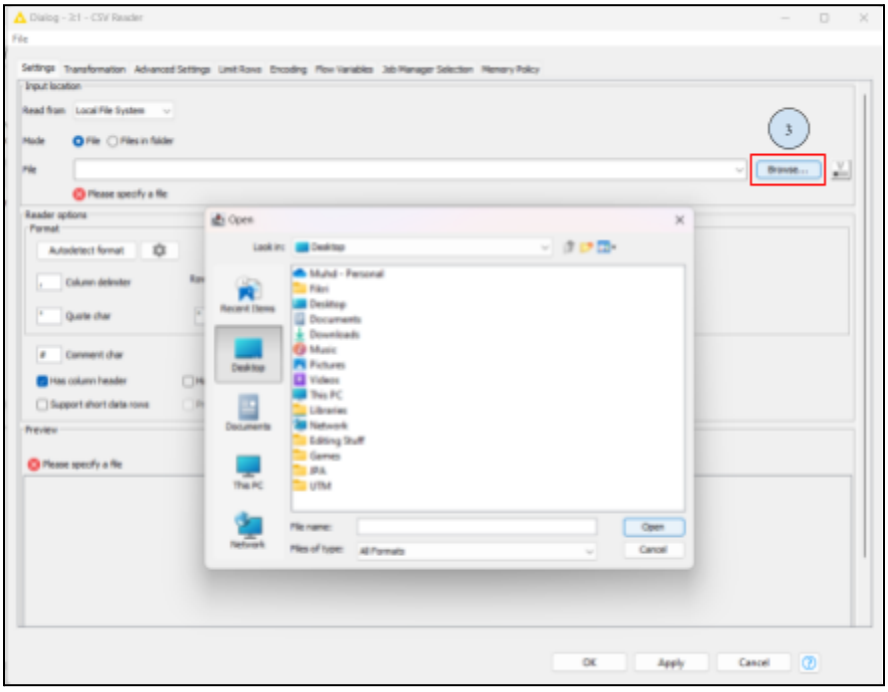


Figure 1.1.1.3

3. Click on “Browse” and then a new window will open to select the dataset file on your computer.

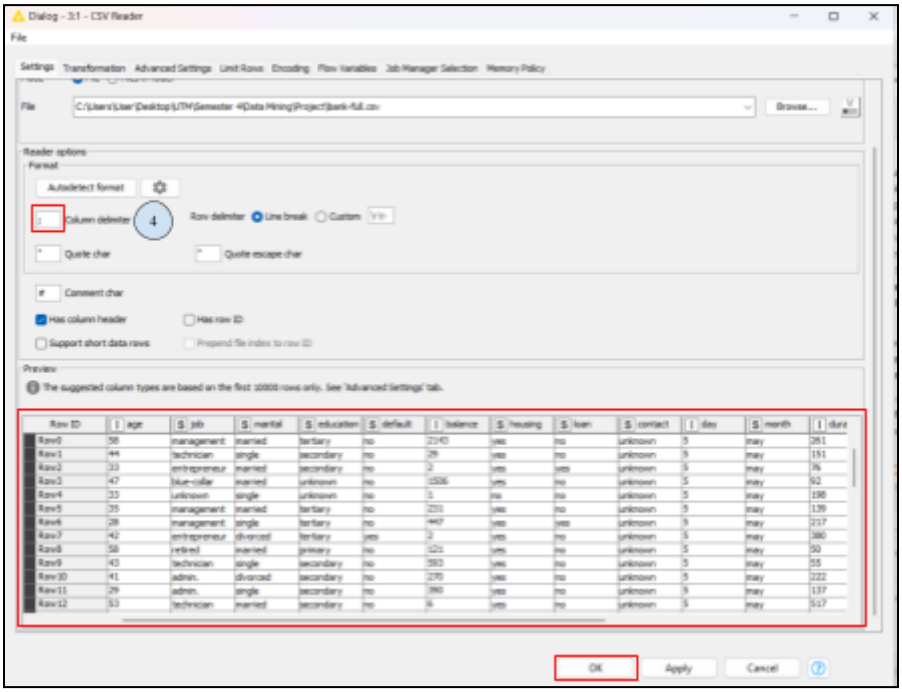


Figure 1.1.1.4

4. Set the “Column delimiter” to ‘;’ (Semicolon) and then there is a preview of the data below and click on the “OK” button.

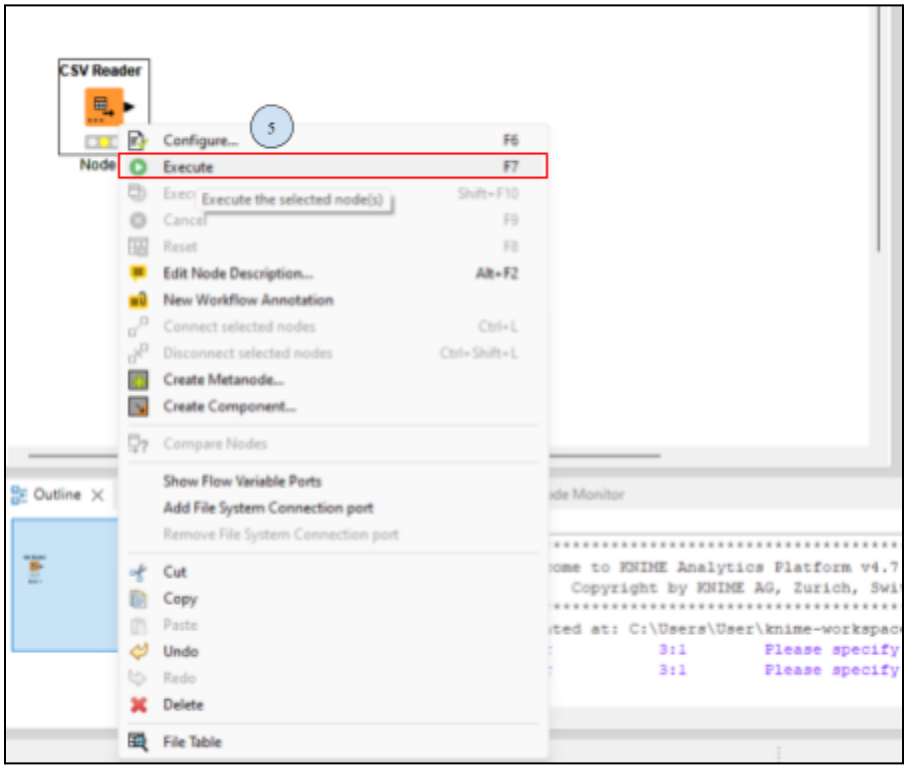


Figure 1.1.1.5

5. Right click on the “CSV Reader” node and then click on “Execute” to execute the selected node.

Data Cleaning

Now we are going to do Data Cleaning on our dataset. So, let us start with identifying outliers in the dataset.

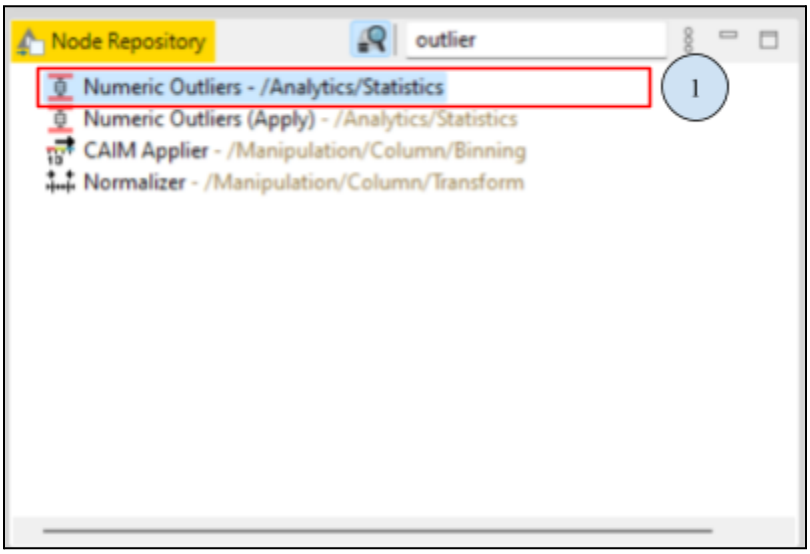


Figure 1.1.1.6

1. Type “outliers” on the Node Repository section and drag the one in the red-colored box to the work area.

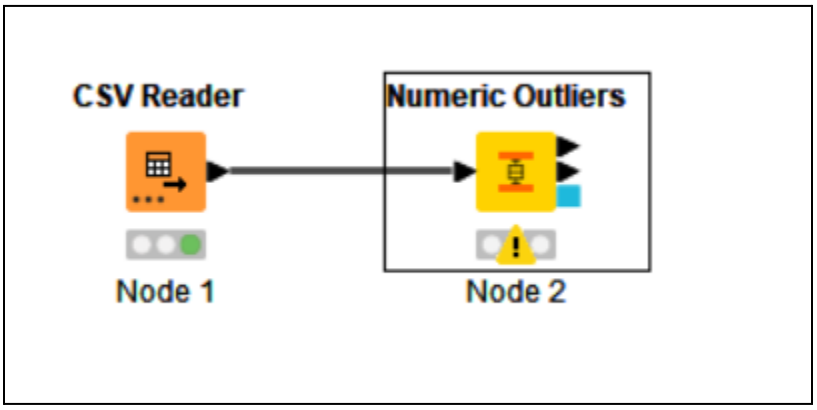


Figure 1.1.1.7

2. Make a connection from the “CSV Reader” node to the “” node and double click on the “Missing Value” node and then a new window will open.

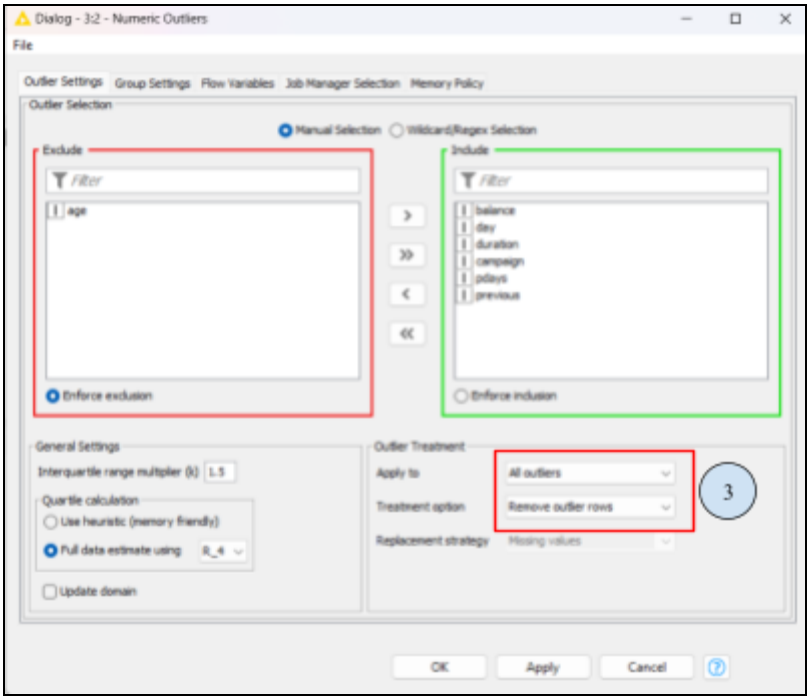


Figure 1.1.1.8

3. We are excluding the "age" attribute. Set "Apply to" to "All outliers" at Outlier Treatment section and "treatment option" to "Remove outlier rows" because this is the most prevalent method to deal with outliers.

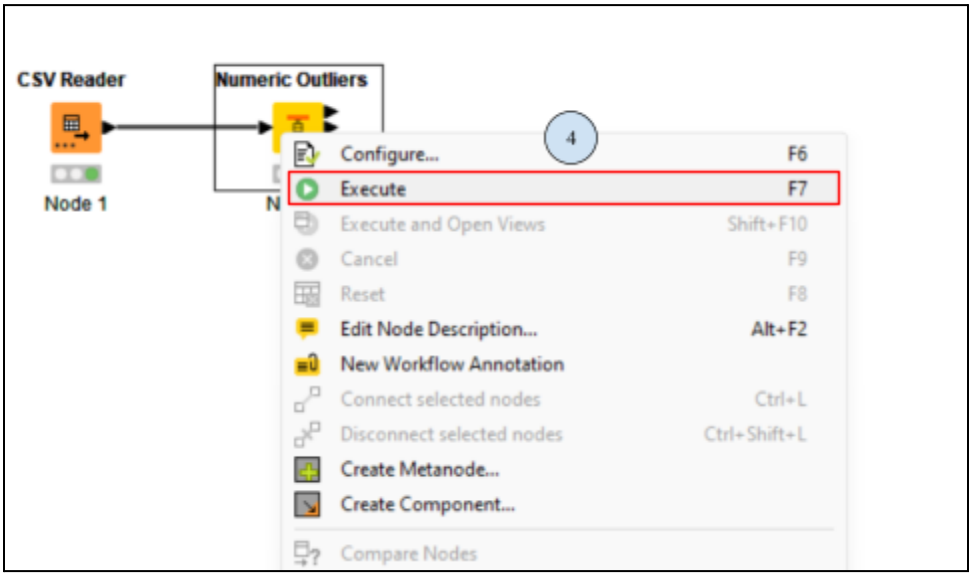


Figure 1.1.1.9

4. Right click on the "Numeric Outliers" node and then click on "Execute" to execute the selected node.

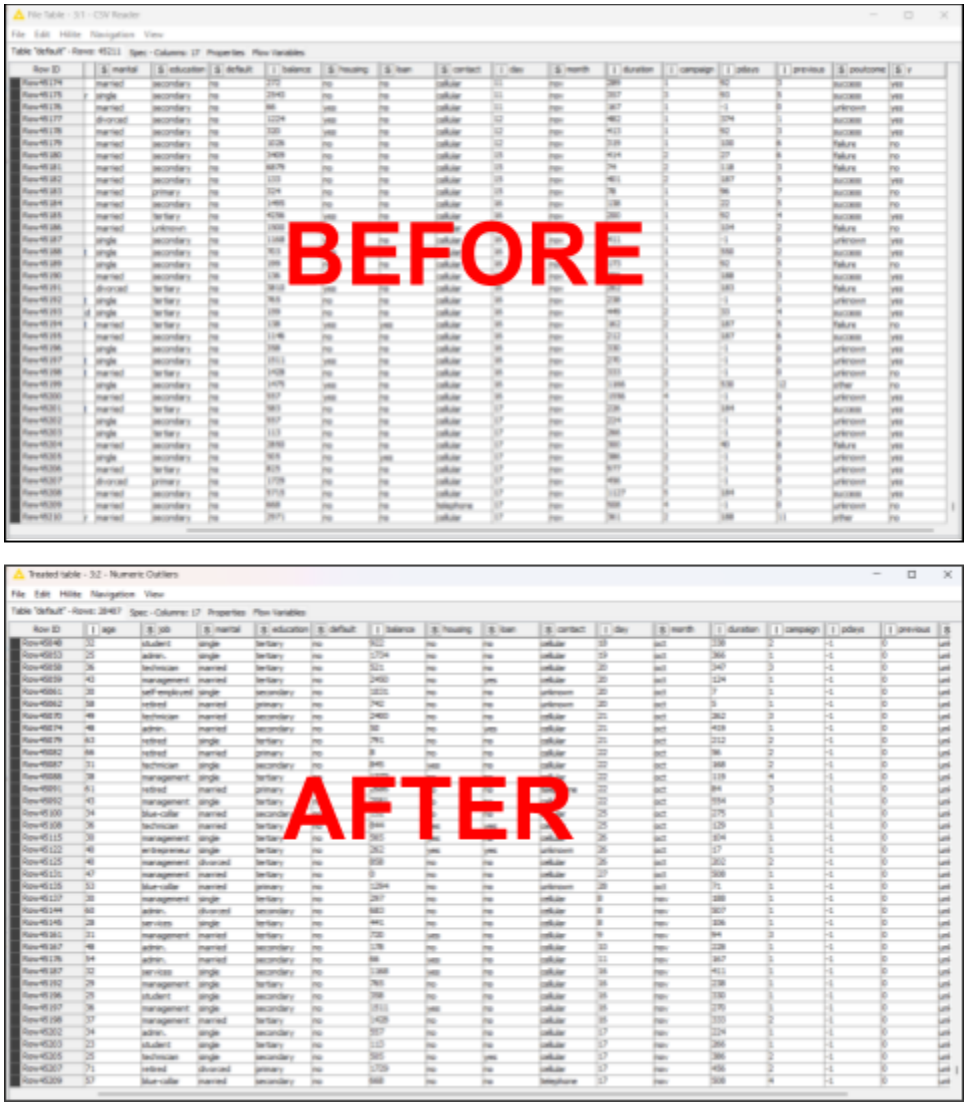


Figure 1.1.1.10

5. As you can see, there is a difference between the number of total rows produced. Before we performed the deletion of the outlier, it was a total of 45,210 numbers of rows. But after we performed the deletion of the outlier, the number of rows became 45,209 which means that there is an outlier on the dataset.

1.1.2 Dataset 2: groceriesItem.csv (Association Rule - Apriori)

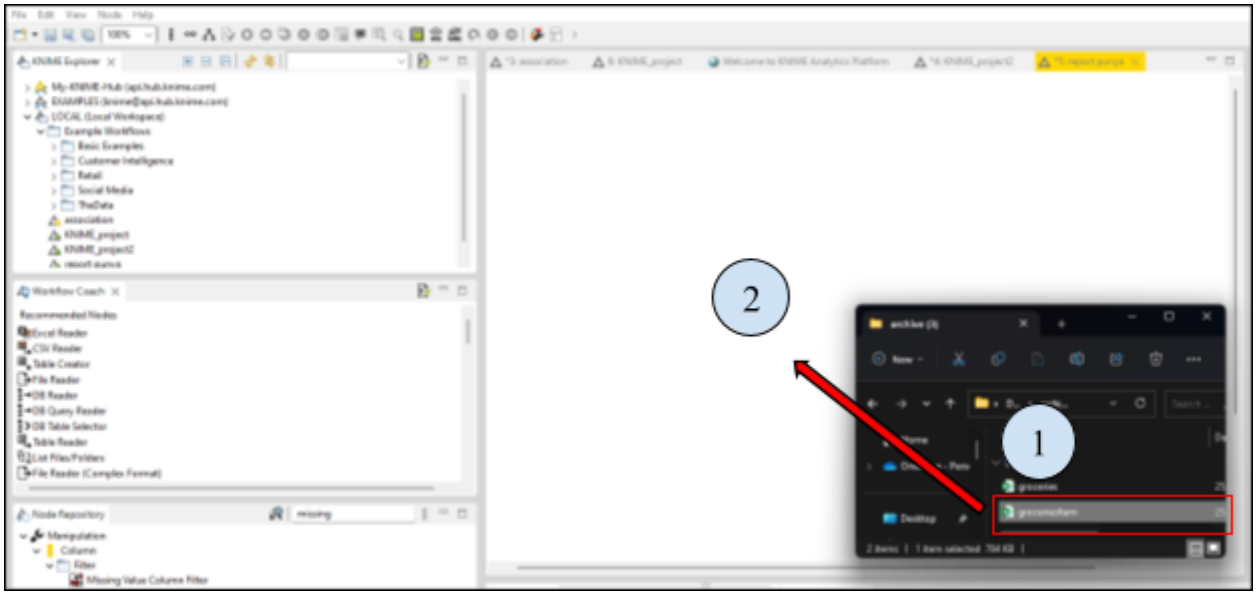


Figure 1.1.2.1

1. To import the data , drag your file from file manager and drop it into KNIME canvas.

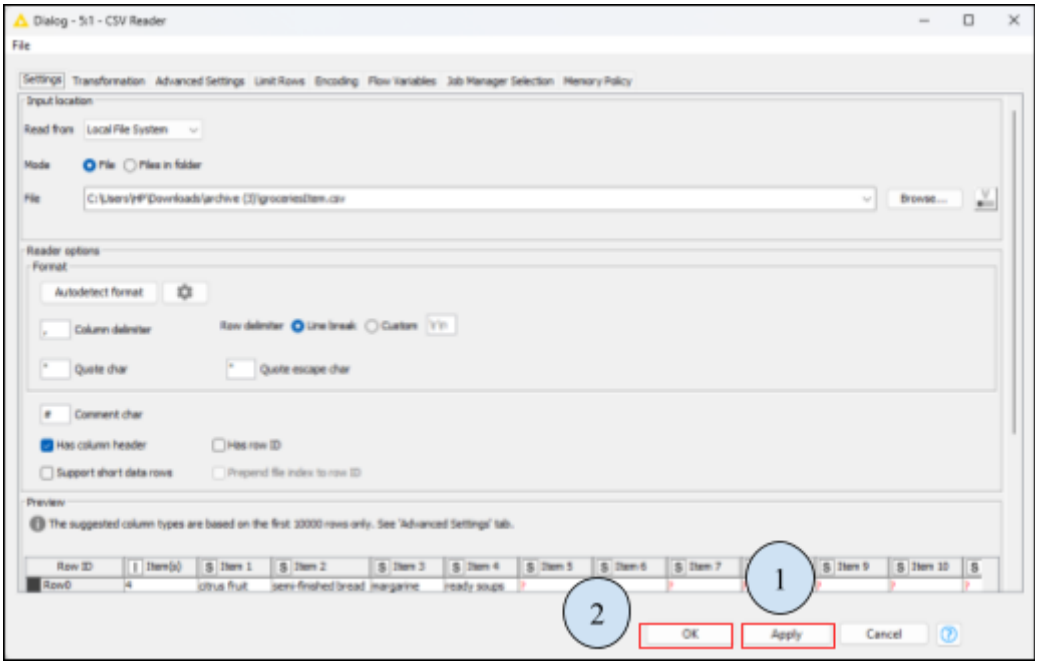


Figure 1.1.2.2

2. A window will pop up, click Apply and Ok.

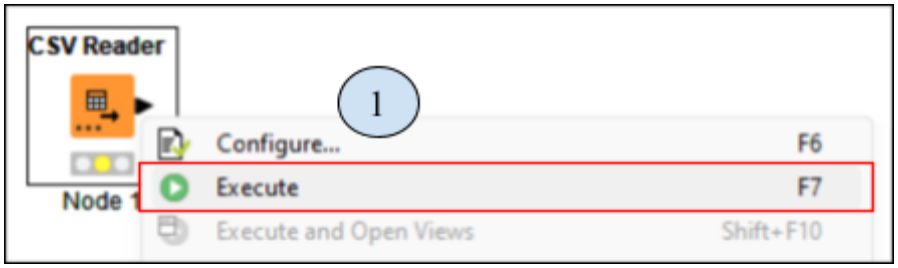


Figure 1.1.2.3

3. Double click the CSV Reader and click Execute.

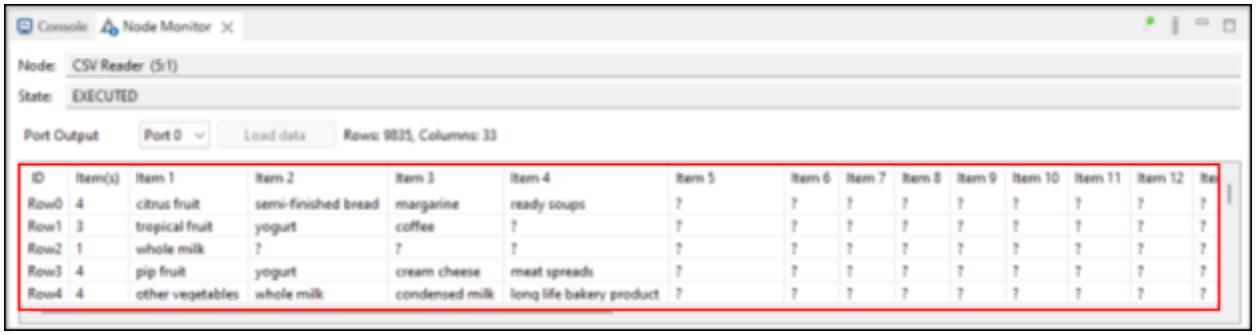


Figure 1.1.2.4

4. A table will appear at the bottom indicating that the dataset is imported.

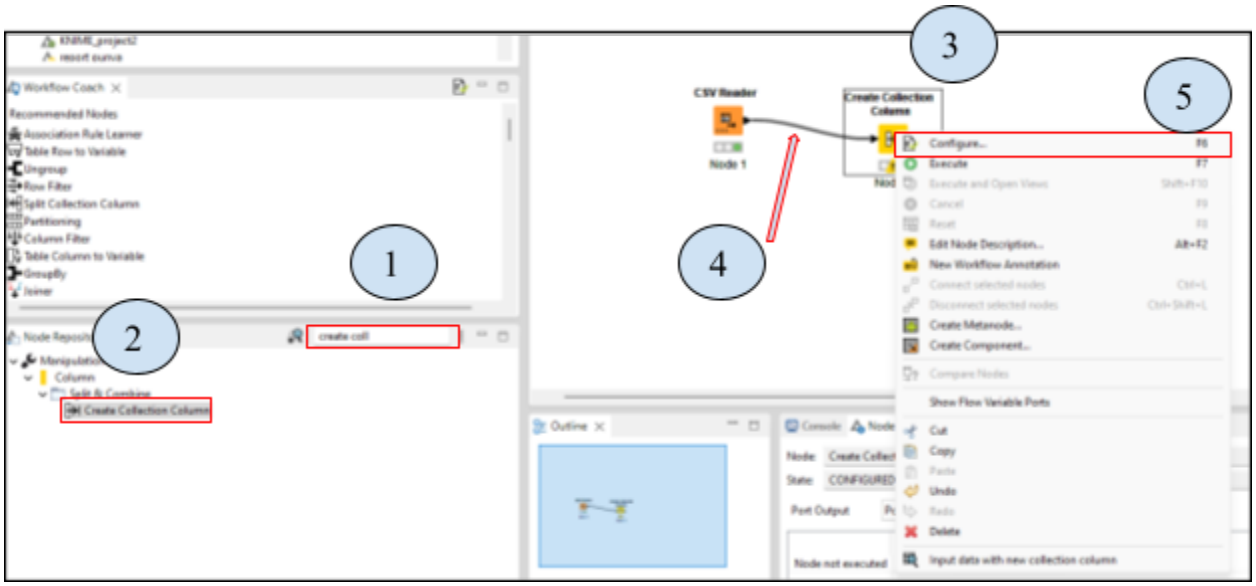


Figure 1.1.2.5

5. Search Create Collection Column and drag into the canvas
6. Connect the output of the CSV Reader to the new node added.
7. Double click the added node and click configuration.

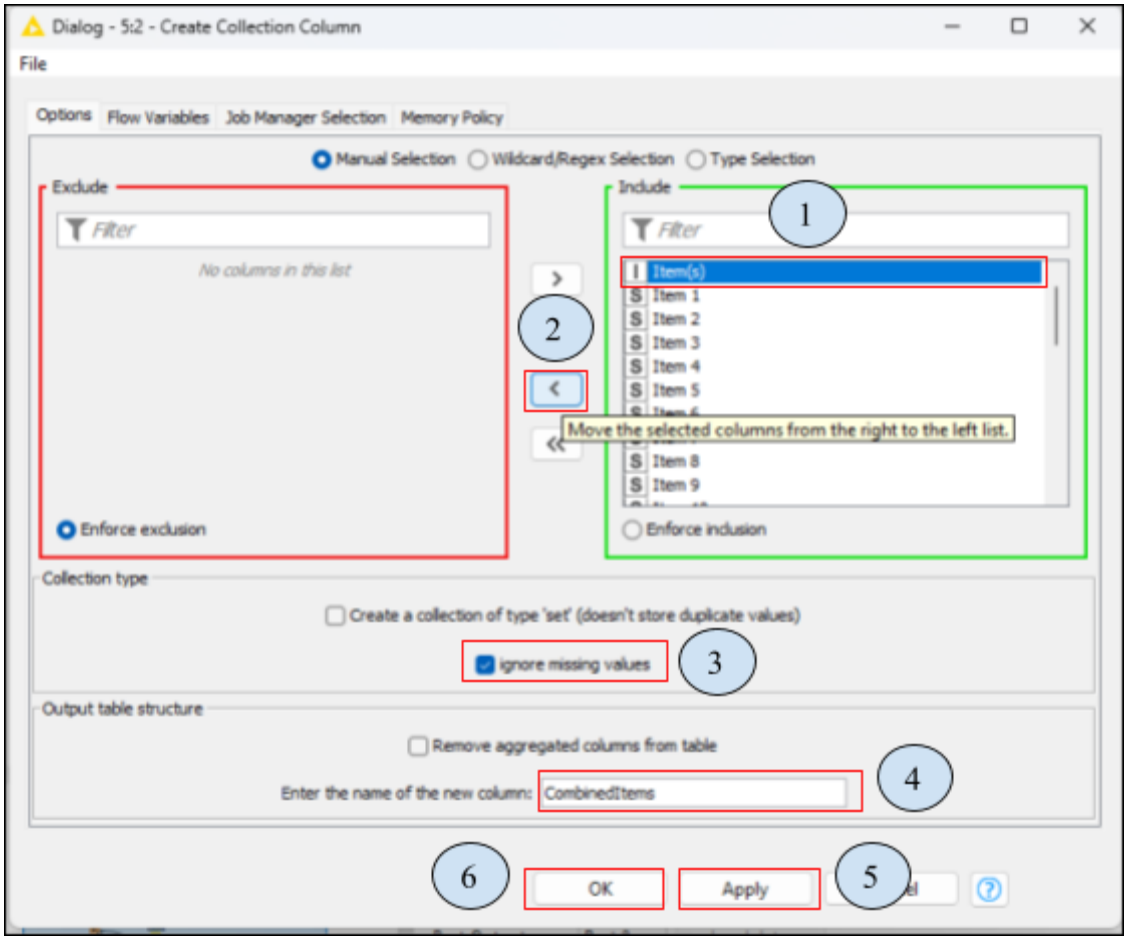


Figure 1.1.2.6

8. A new window will popup, select the first attribute and click on the left arrow.
9. This is to filter out the selected which is the 1st column (Item(s)) and create a new column with the values of combined columns selected on the include box.
10. Check the “ignore missing values” box.
11. Rename the new column at the box indicated.
12. Click Apply and OK.

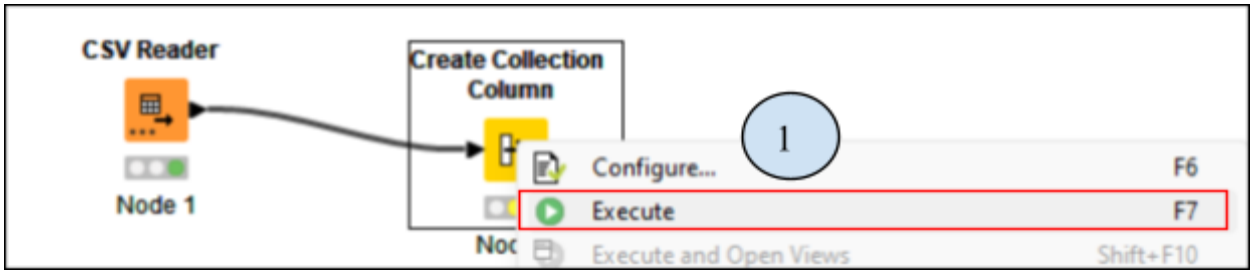


Figure 1.1.2.7

13. Double click the node and click on Execute.

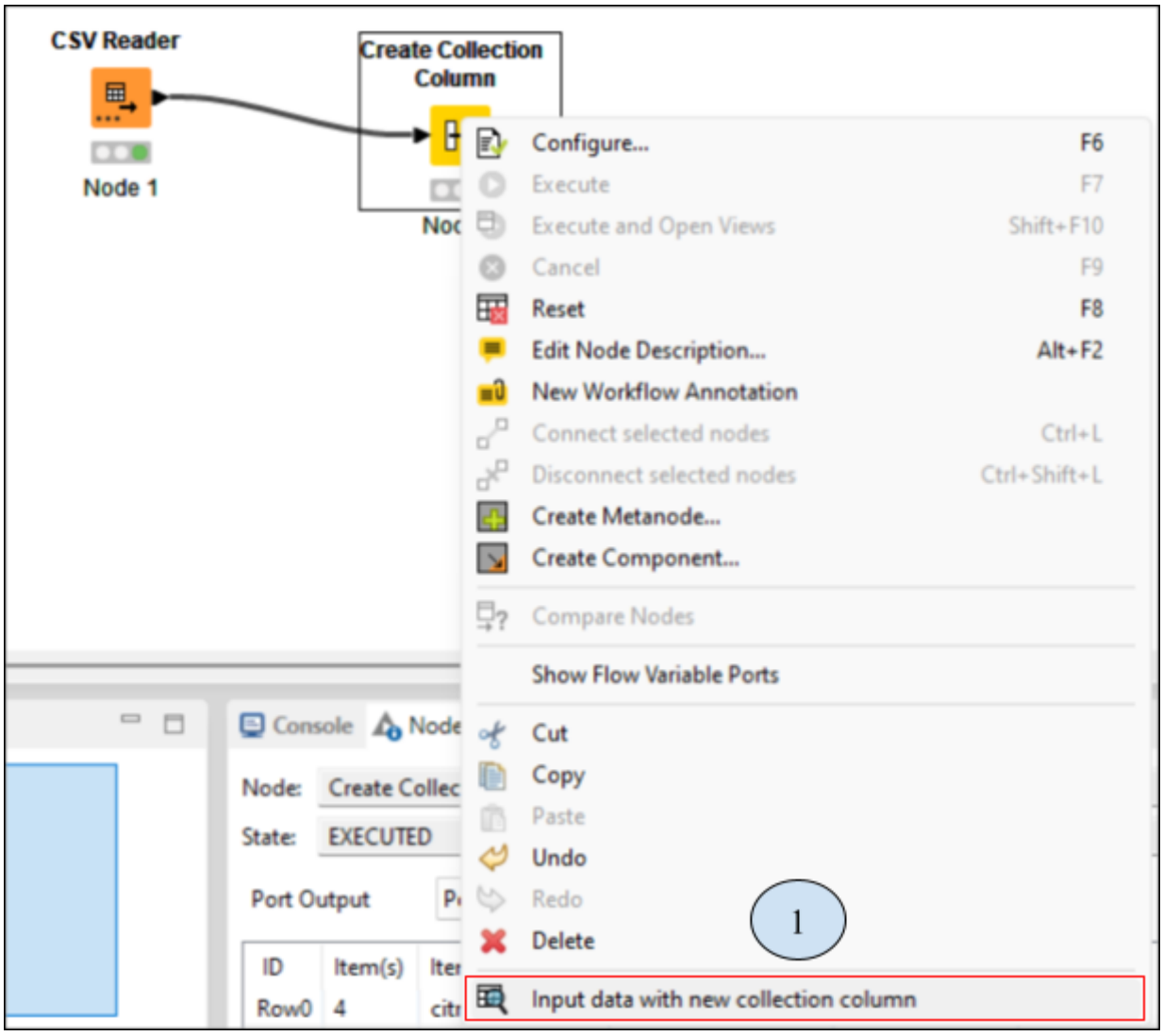


Figure 1.1.2.8

14. To view the output, double click the node and click Input data with the new collection column.

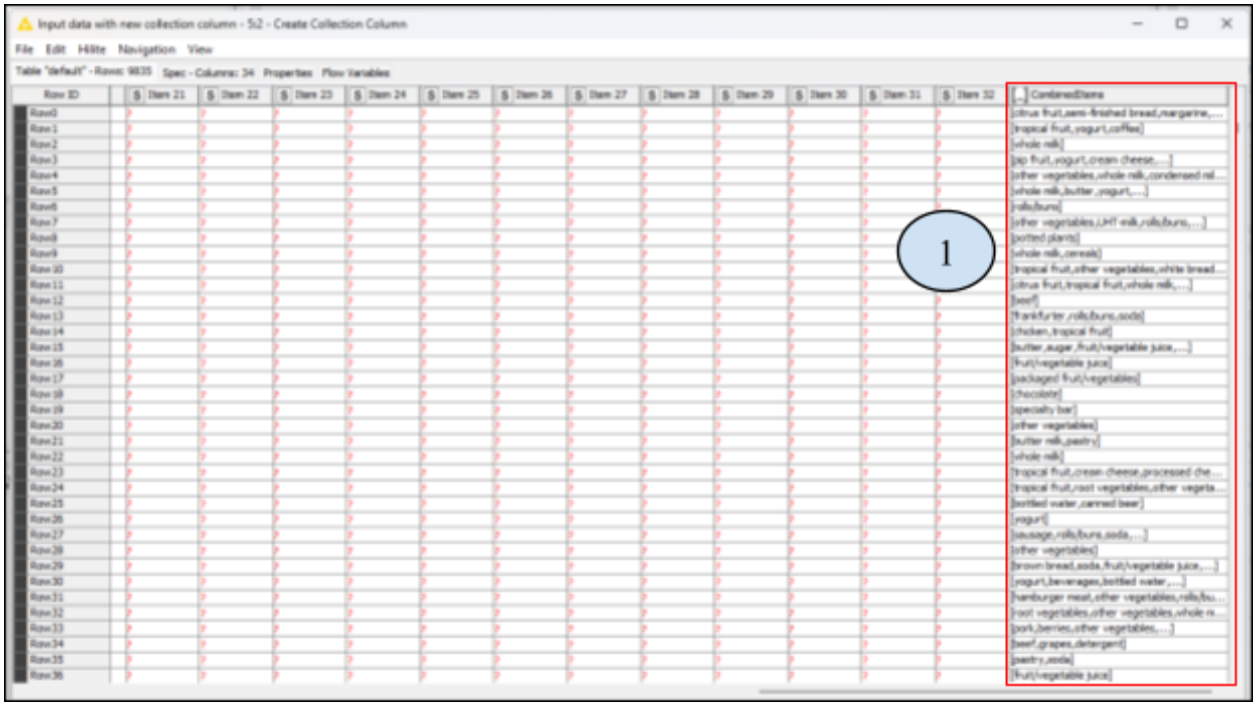


Figure 1.1.2.9

15. We can see that a new column is added.

Figure 1.1.2.10

16. Search for column filters, and drag into the canvas.
17. Connect the Create Column node to the Column Filter node.

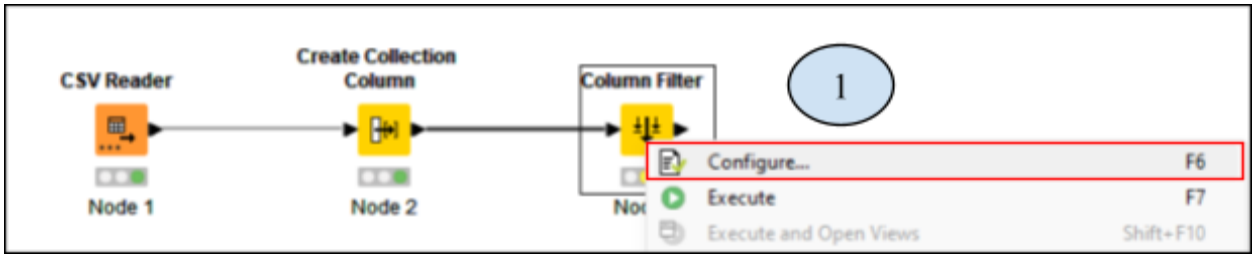


Figure 1.1.2.11

18. Configure it by double clicking the node and click configure.

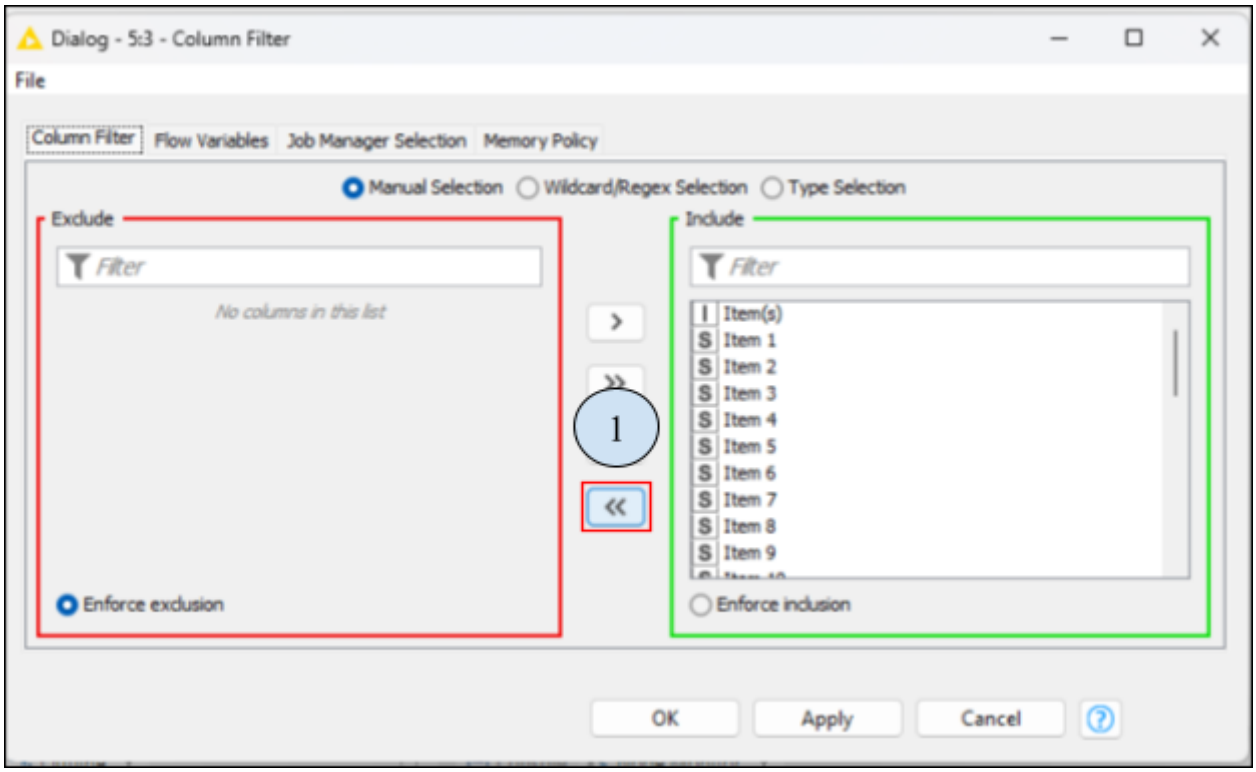


Figure 1.1.2.12

19. Click on the double left arrow to move all the attributes to the left.

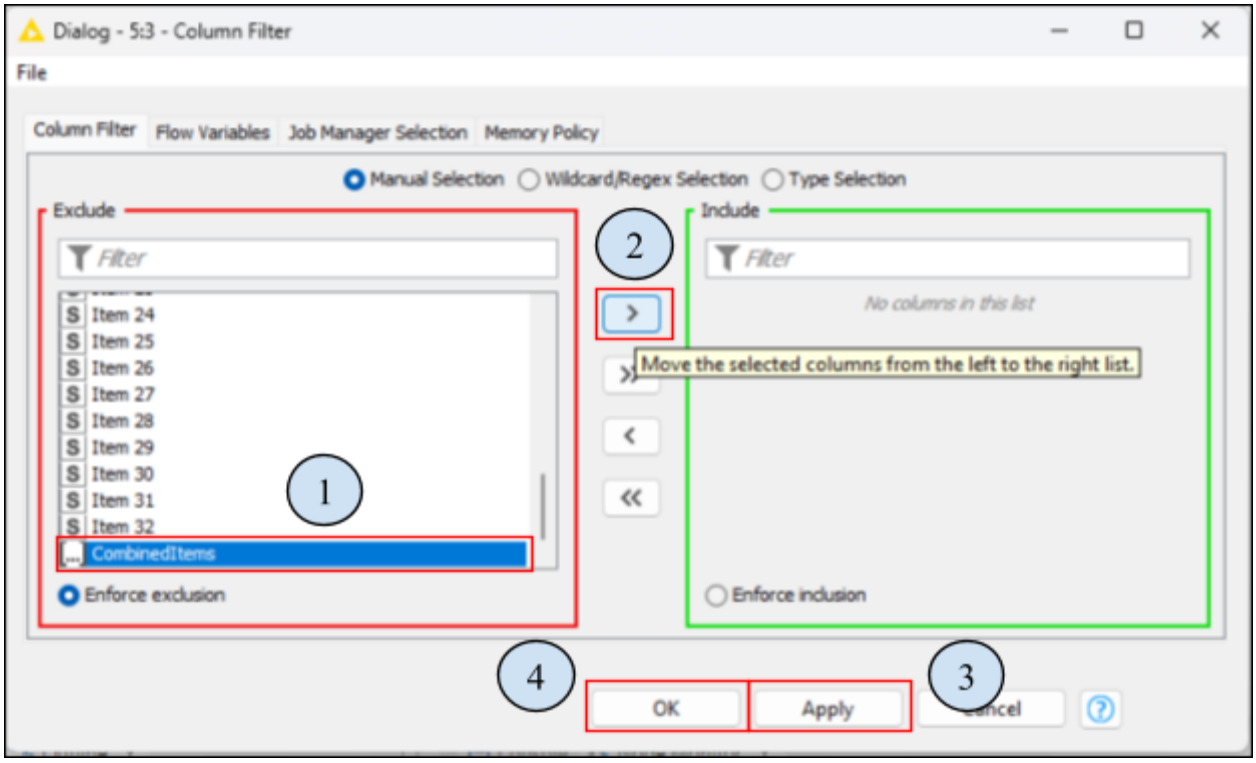


Figure 1.1.2.13

20. Select the new column added and move it to the right by clicking the right arrow.
21. Click Apply and OK.

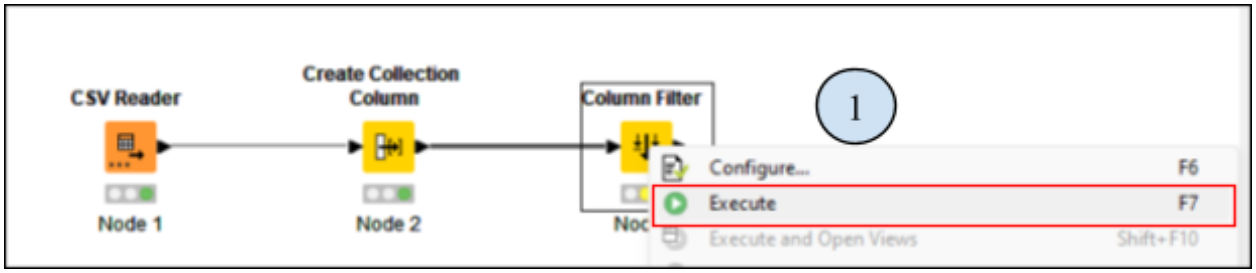


Figure 1.1.2.14

22. Double click on the added node and click execute.

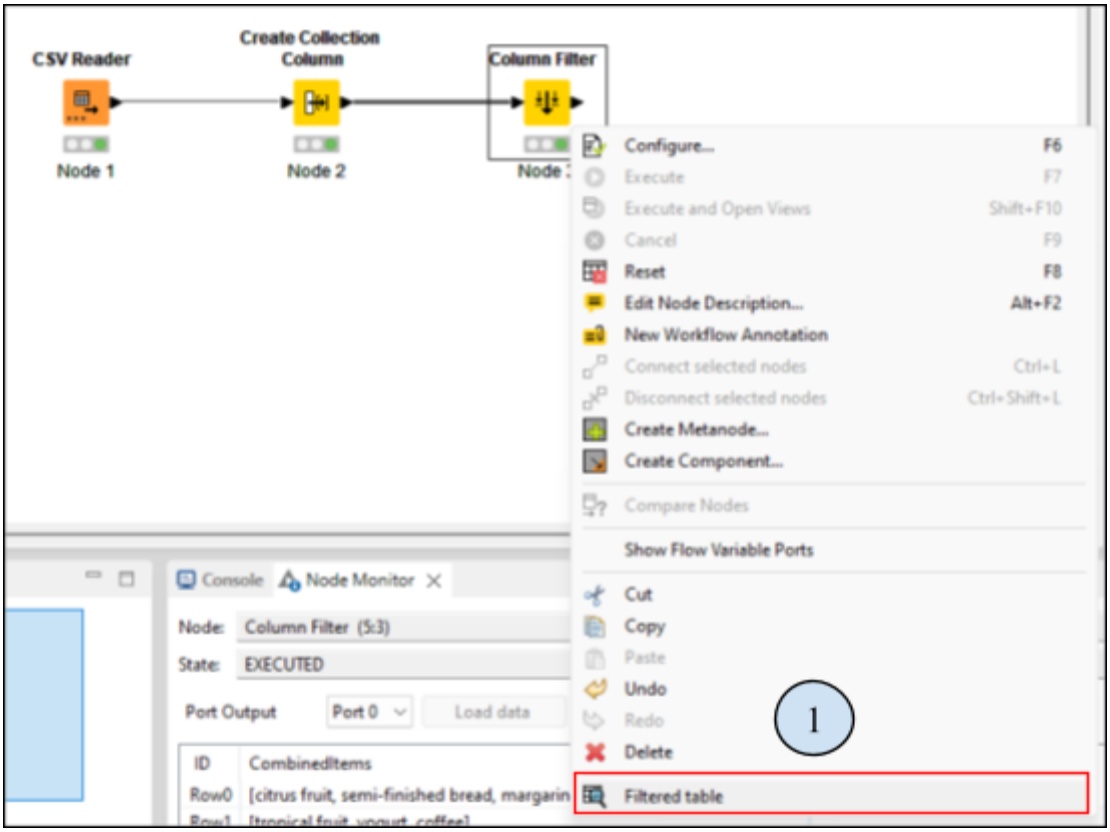


Figure 1.1.2.15

23. To see the executed content, double click on the added node.
24. Click the Filtered table.

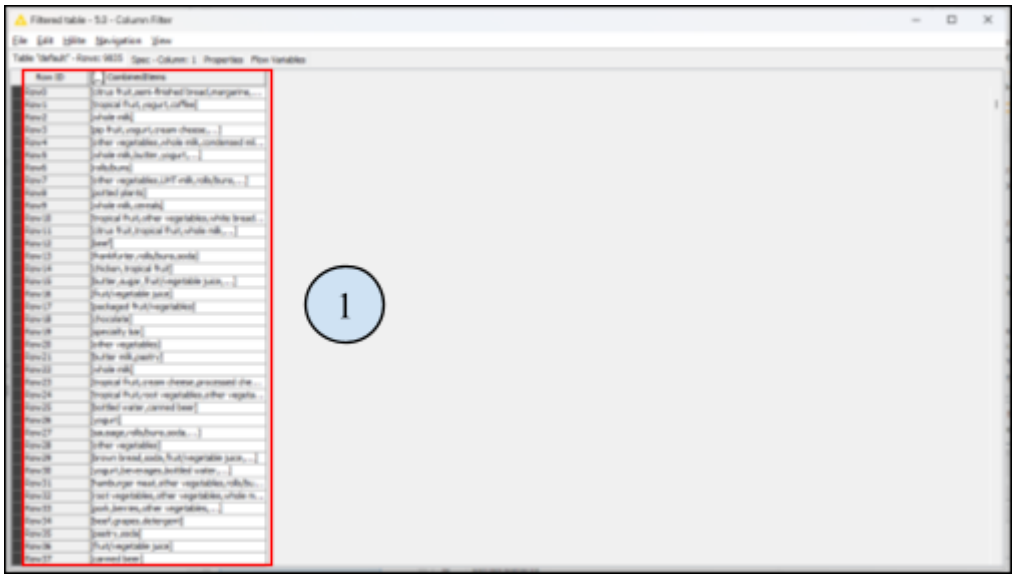


Figure 1.1.2.16

25. Filtered table is shown on a new window.

1.1.3 Dataset 3: Boston.csv (Regression Rule - Linear Regression)

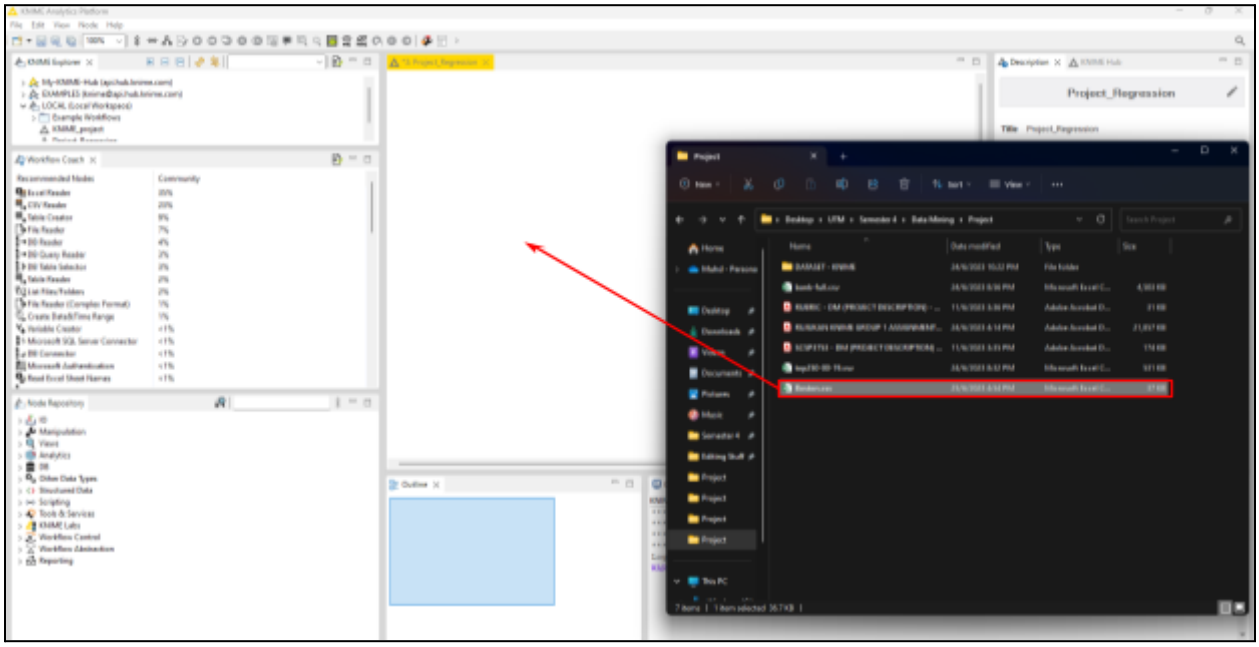


Figure 1.1.3.1

1. Drag the dataset Boston.csv file from the computer to the KNIME work area canvas.

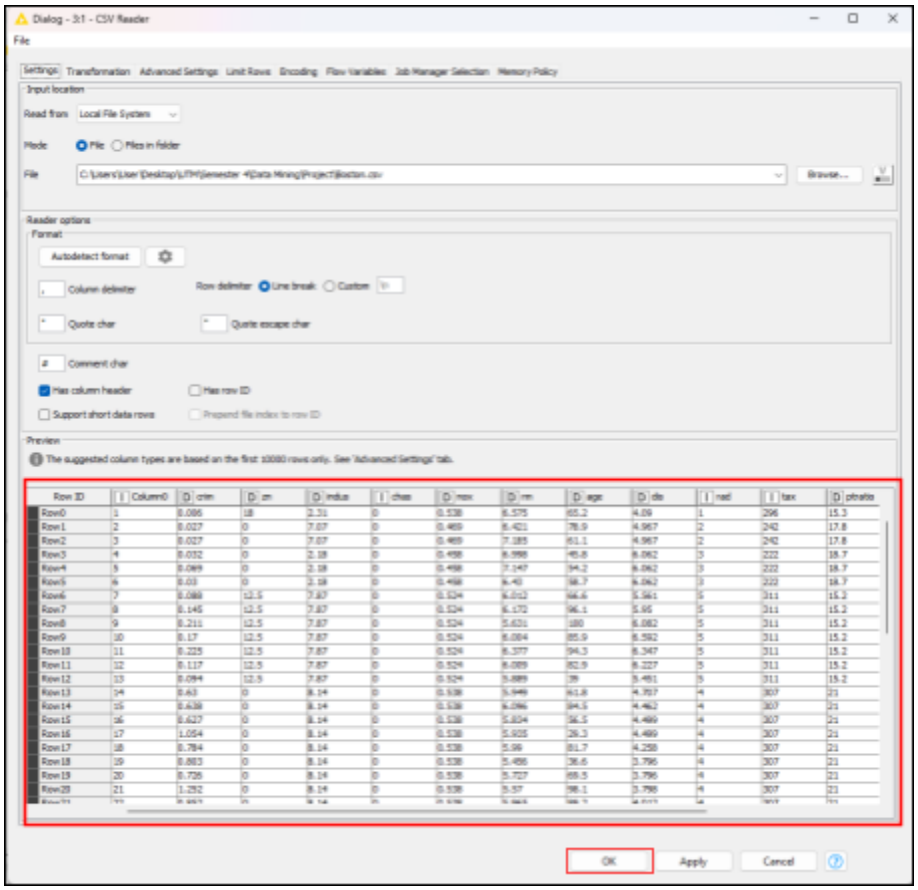


Figure 1.1.3.2

2. A new window will open. From the window, you can see the preview of the data that has been loaded to the canvas. And then click on the “OK” button.

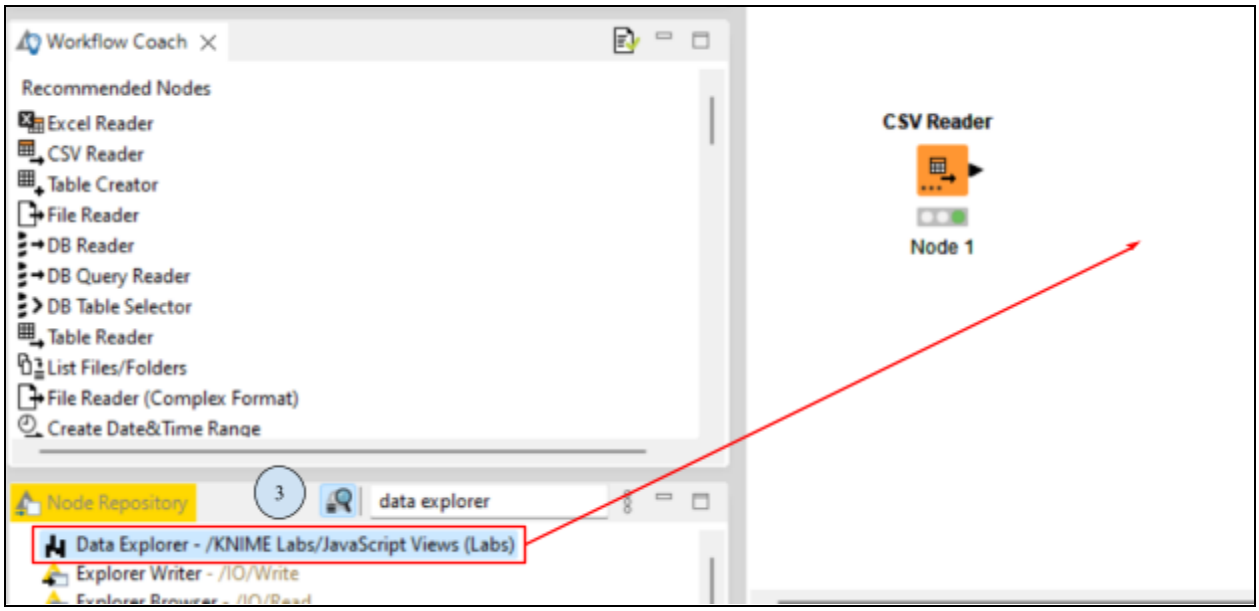


Figure 1.1.3.3

3. Next, we are going to insert a “Data Explorer” node to the work area. Please note that if you cannot find the “Data Explorer” node from the node repository, Please install the "KNIME Labs Extensions". Go to File -> Install KNIME Extension, then type that in.

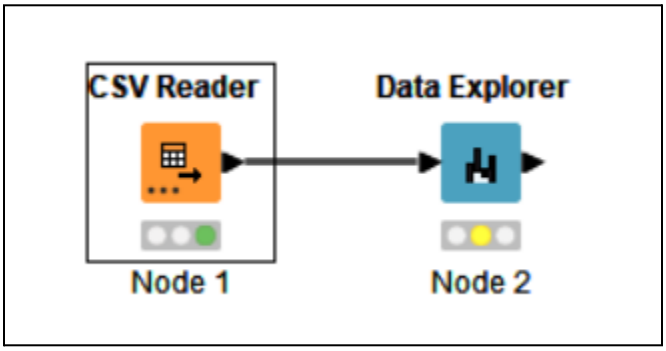


Figure 1.1.3.4

4. Connect the “CSV Reader” node by dragging the arrow from the node to the “Data Explorer” node.

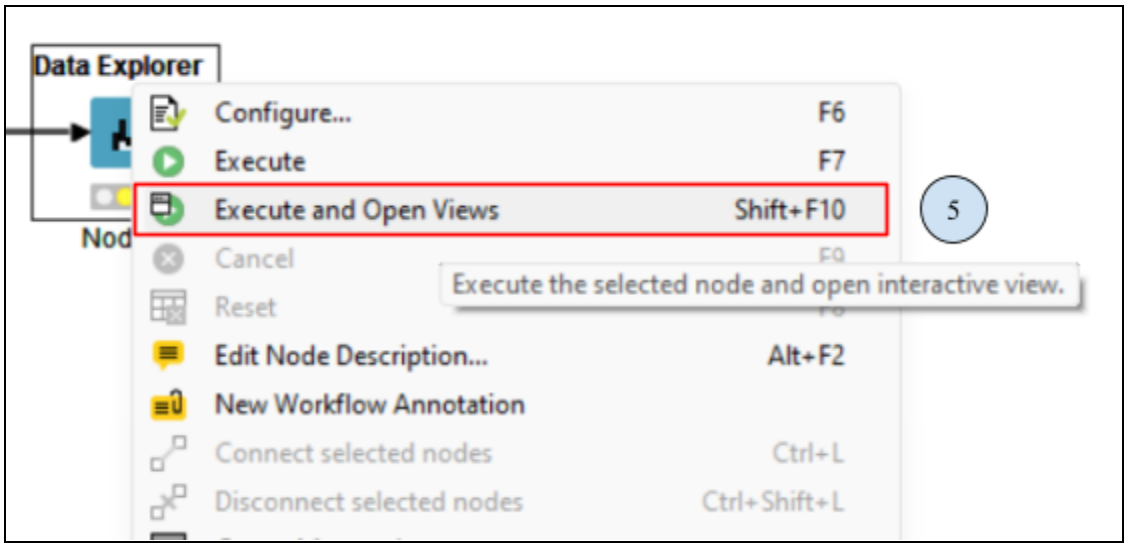


Figure 1.1.3.5

5. Right click on the "Data Explorer" node and then click on the "Execute and Open Views". A new window will open right after that.

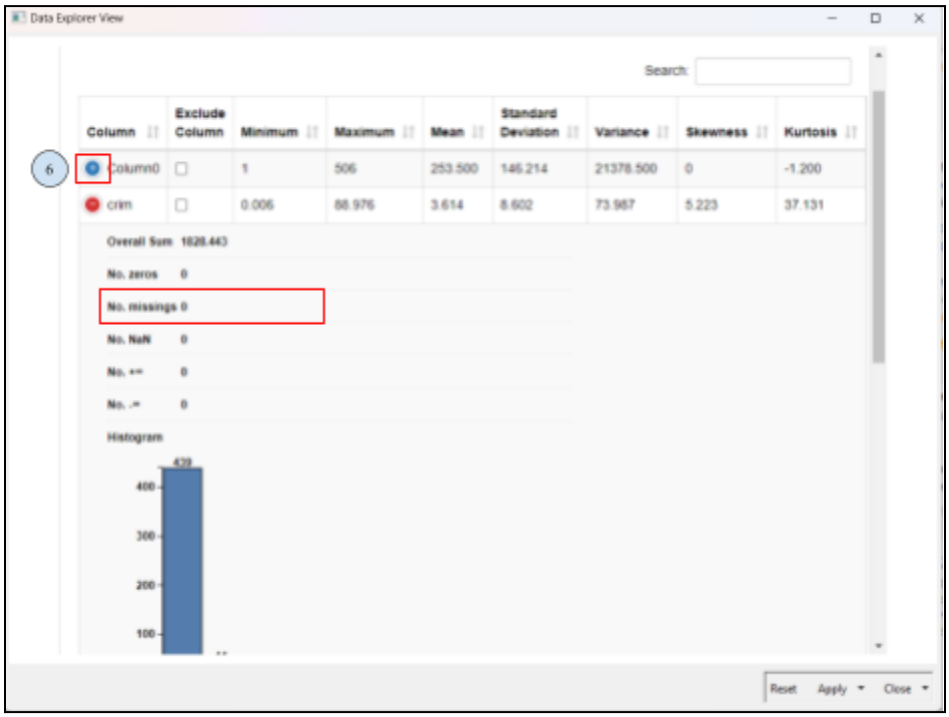


Figure 1.1.3.6

6. By clicking on the "+" button on the left side, we can see the details of each column in our dataset. This is to check whether there are any missing values on that column because we need to deal with the missing value first before we can proceed with the data

mining task. Next, repeat these steps on every column to see if there are any missing values on that column. The node also allows us to see histogram visualization according to that column.

7. Since there were no missing values on that data. So, we can proceed to the next step.

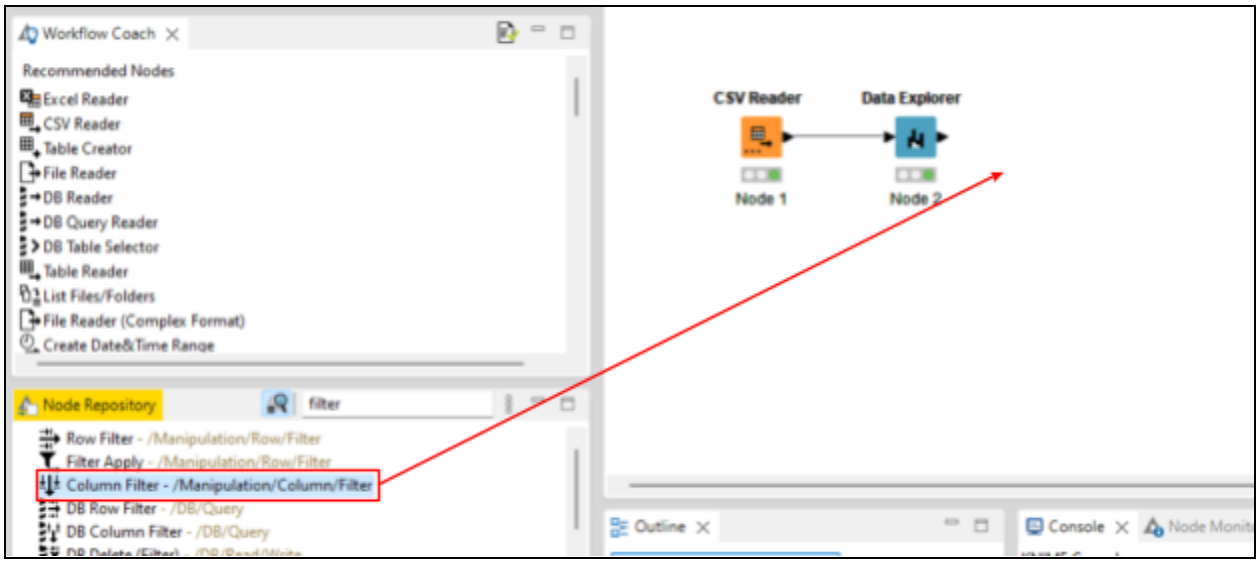


Figure 1.1.3.7

8. For this step, I am going to delete a column called “Column0” because I think this column is unnecessary for the task as it just repeats the row ID provided by KNIME. Therefore, we need to find the “Column Filter” node and drag the node to the work area and connect the “Data Explorer” node to the “Column Filter” node by dragging the arrow from “Data Explorer” node to the “Column Filter” node.

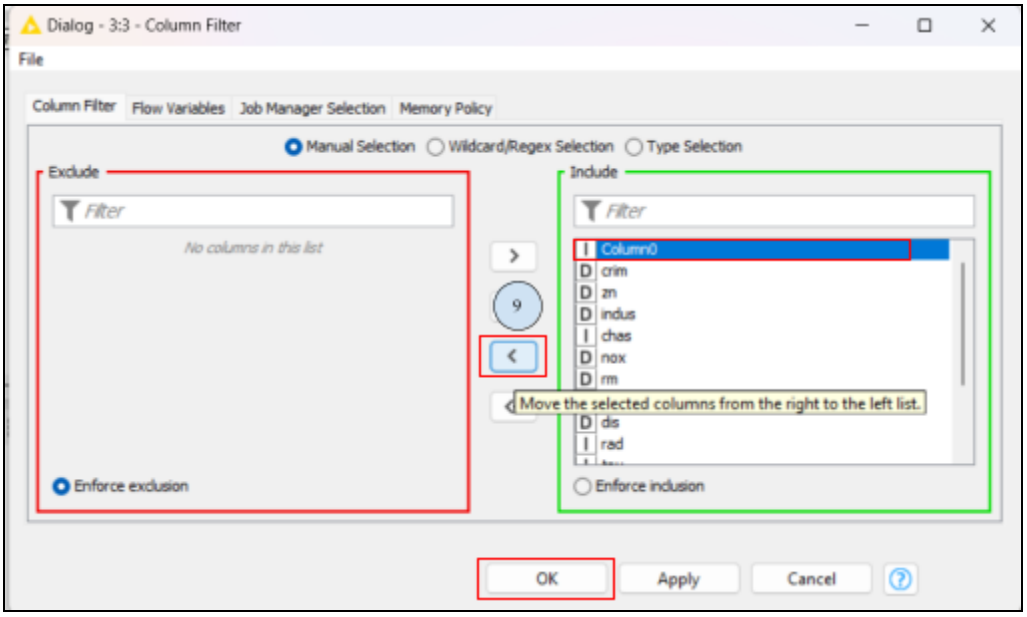
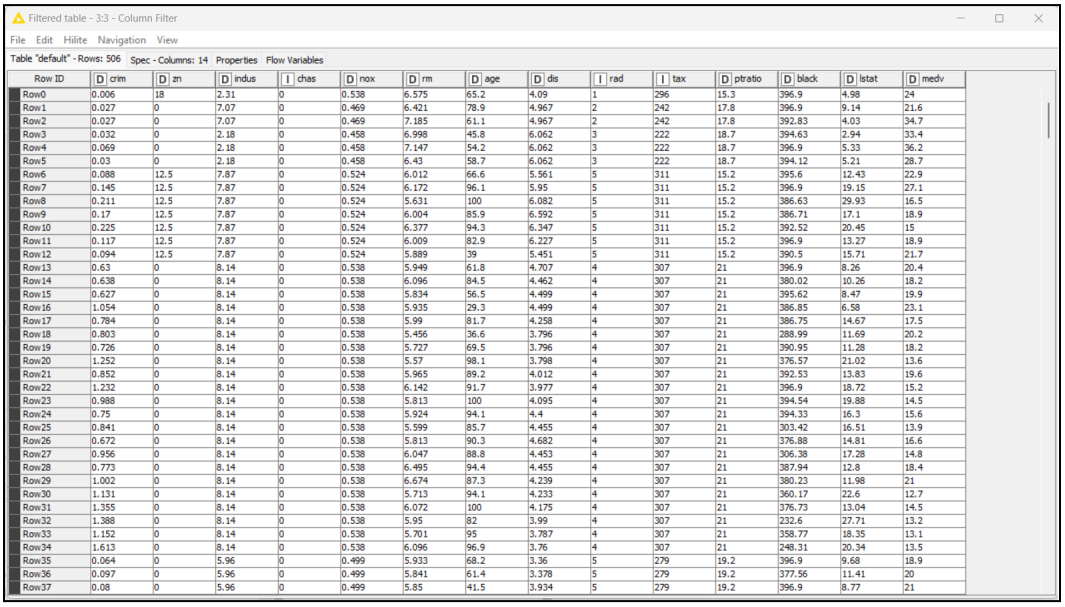


Figure 1.1.3.8

9. Double click on the "Column Filter" node and a new window will open right after that. Select the "Column0" column on the right side and click on the "<" button. After that, click on the "OK" button. Then, execute the node.



Filtered table - 3:3 - Column Filter

File Edit Hilite Navigation View

Table "default" - Rows: 506 Spec - Columns: 14 Properties Flow Variables

Row ID	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv
Row0	0.006	18	2.31	0	0.538	6.575	65.2	4.09	1	296	15.3	396.9	4.98	24
Row1	0.027	0	7.07	0	0.469	6.421	78.9	4.967	2	242	17.8	396.9	9.14	21.6
Row2	0.027	0	7.07	0	0.469	7.185	61.1	4.967	2	242	17.8	392.83	4.03	34.7
Row3	0.032	0	2.18	0	0.458	6.998	45.8	6.062	3	222	18.7	394.63	2.94	33.4
Row4	0.069	0	2.18	0	0.458	7.147	54.2	6.062	3	222	18.7	396.9	5.33	36.2
Row5	0.03	0	2.18	0	0.458	6.43	58.7	6.062	3	222	18.7	394.12	5.21	28.7
Row6	0.088	12.5	7.87	0	0.524	6.012	66.6	5.561	5	311	15.2	395.6	12.43	22.9
Row7	0.145	12.5	7.87	0	0.524	6.172	96.1	5.95	5	311	15.2	396.9	19.15	27.1
Row8	0.211	12.5	7.87	0	0.524	5.631	100	6.082	5	311	15.2	386.63	29.93	16.5
Row9	0.17	12.5	7.87	0	0.524	6.004	85.9	6.592	5	311	15.2	386.71	17.1	18.9
Row10	0.225	12.5	7.87	0	0.524	6.377	94.3	6.347	5	311	15.2	392.52	20.45	15
Row11	0.117	12.5	7.87	0	0.524	6.009	82.9	6.227	5	311	15.2	396.9	13.27	18.9
Row12	0.094	12.5	7.87	0	0.524	5.889	39	5.451	5	311	15.2	390.5	15.71	21.7
Row13	0.63	0	8.14	0	0.538	5.949	61.8	4.707	4	307	21	396.9	8.26	20.4
Row14	0.638	0	8.14	0	0.538	6.096	84.5	4.462	4	307	21	380.02	10.26	18.2
Row15	0.627	0	8.14	0	0.538	5.834	56.5	4.499	4	307	21	395.62	8.47	19.9
Row16	1.054	0	8.14	0	0.538	5.935	29.3	4.499	4	307	21	386.85	6.58	23.1
Row17	0.784	0	8.14	0	0.538	5.99	81.7	4.258	4	307	21	386.75	14.67	17.5
Row18	0.803	0	8.14	0	0.538	5.456	36.6	3.796	4	307	21	288.99	11.69	20.2
Row19	0.726	0	8.14	0	0.538	5.727	69.5	3.796	4	307	21	390.95	11.28	18.2
Row20	1.252	0	8.14	0	0.538	5.57	98.1	3.798	4	307	21	376.57	21.02	13.6
Row21	0.852	0	8.14	0	0.538	5.965	89.2	4.012	4	307	21	392.53	13.83	19.6
Row22	1.232	0	8.14	0	0.538	6.142	91.7	3.977	4	307	21	396.9	18.72	15.2
Row23	0.988	0	8.14	0	0.538	5.813	100	4.095	4	307	21	394.54	19.88	14.5
Row24	0.75	0	8.14	0	0.538	5.924	94.1	4.4	4	307	21	394.33	16.3	15.6
Row25	0.841	0	8.14	0	0.538	5.599	85.7	4.455	4	307	21	303.42	16.51	13.9
Row26	0.672	0	8.14	0	0.538	5.813	90.3	4.682	4	307	21	376.88	14.81	16.6
Row27	0.956	0	8.14	0	0.538	6.047	88.8	4.453	4	307	21	306.38	17.28	14.8
Row28	0.773	0	8.14	0	0.538	6.495	94.4	4.455	4	307	21	387.94	12.8	18.4
Row29	1.002	0	8.14	0	0.538	6.674	87.3	4.239	4	307	21	380.23	11.98	21
Row30	1.131	0	8.14	0	0.538	5.713	94.1	4.233	4	307	21	360.17	22.6	12.7
Row31	1.355	0	8.14	0	0.538	6.072	100	4.175	4	307	21	376.73	13.04	14.5
Row32	1.388	0	8.14	0	0.538	5.95	82	3.99	4	307	21	232.6	27.71	13.2
Row33	1.152	0	8.14	0	0.538	5.701	95	3.787	4	307	21	358.77	18.35	13.1
Row34	1.613	0	8.14	0	0.538	6.096	96.9	3.76	4	307	21	248.31	20.34	13.5
Row35	0.064	0	5.96	0	0.499	5.933	68.2	3.36	5	279	19.2	396.9	9.68	18.9
Row36	0.097	0	5.96	0	0.499	5.841	61.4	3.378	5	279	19.2	377.56	11.41	20
Row37	0.08	0	5.96	0	0.499	5.85	41.5	3.934	5	279	19.2	396.9	8.77	21

Figure 1.1.3.9

10. So, this is what the table looks like after we remove the column "Column0". Now, let's see some virtualizations based on the data that have been preprocessed.

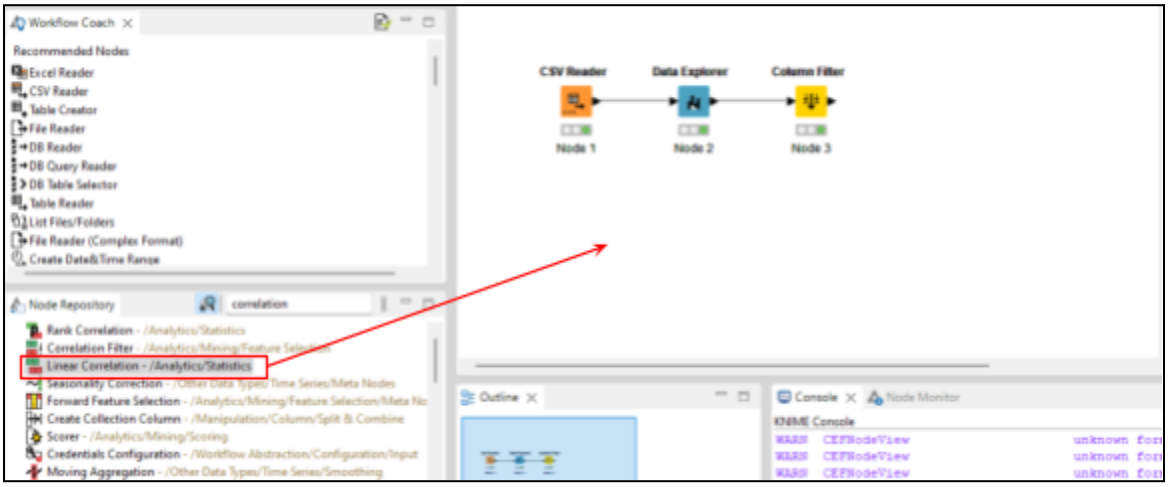


Figure 1.1.3.10

11. Add the "Linear Correlation" node to the work area. This is to measure the linear relationships between the variables.

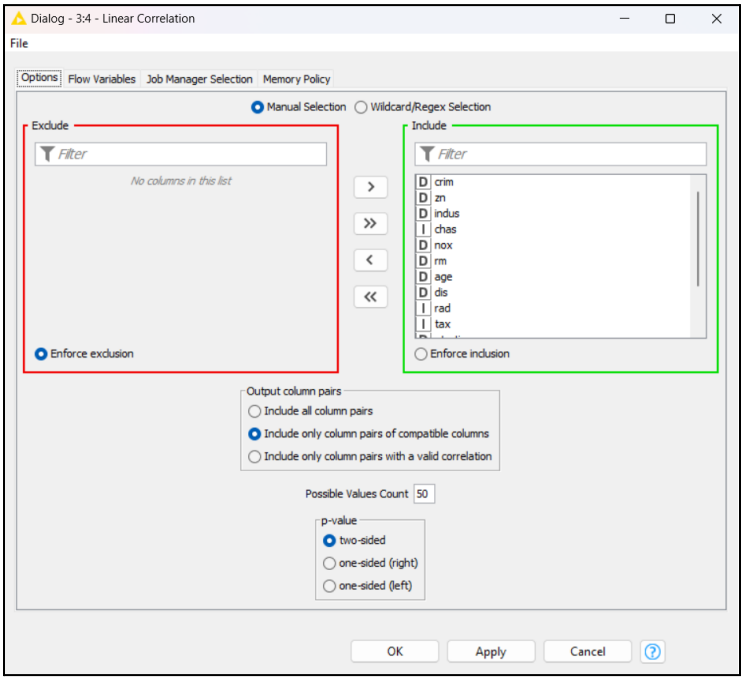


Figure 1.1.3.11

12. Double click on the "Linear Correlation" node. A new window will open. After that, click on the "OK" button. Then, execute the node.

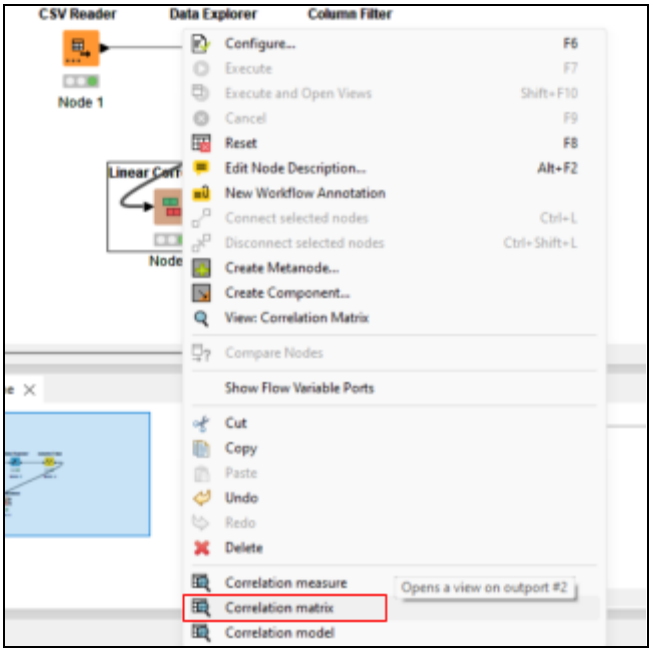
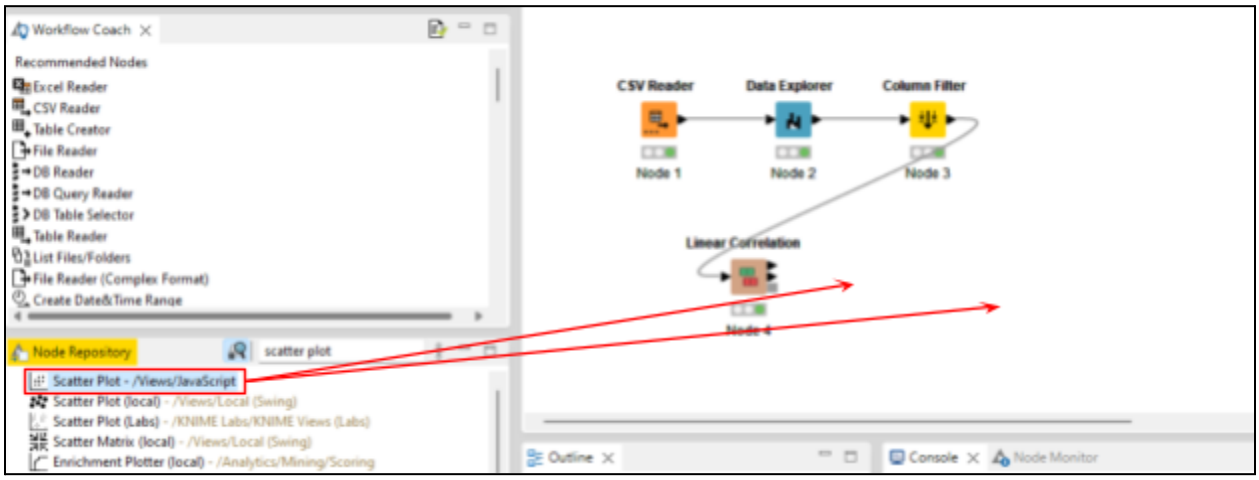


Figure 1.1.3.12

13. Right click on the node and click on “Correlation matrix”. The correlation coefficient ranges from -1 to 1. If the value is close to 1, it means that there is a strong positive correlation between the two variables. When it is close to -1, the variables have a strong negative correlation. We can see that “rm” has a strong positive correlation with “medv” (0.7) whereas “lstat” has a high negative correlation with “medv” (-0.74).



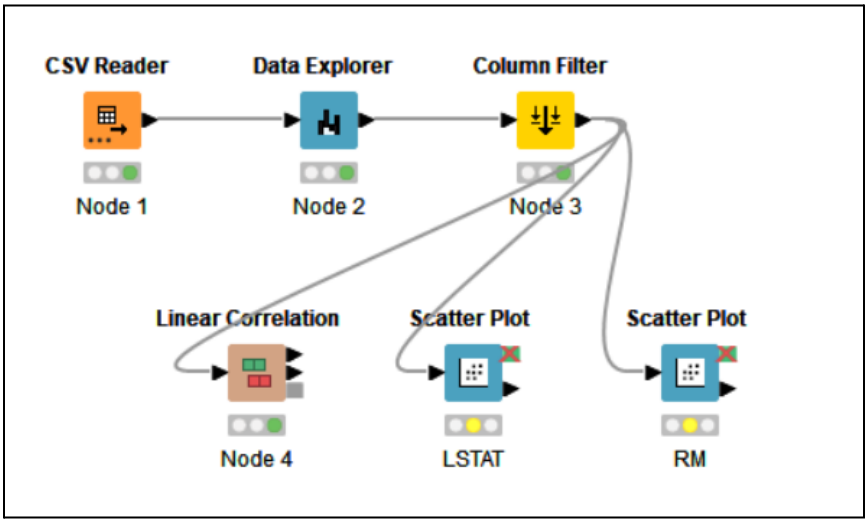


Figure 1.1.3.13

14. Add two “Scatter Plot” nodes to the work area to see how “rm” and “lstat” features vary with “medv”. I am going to rename each node to “LSTAT” and “RM” respectively and connect both nodes with the “Column Filter” node.

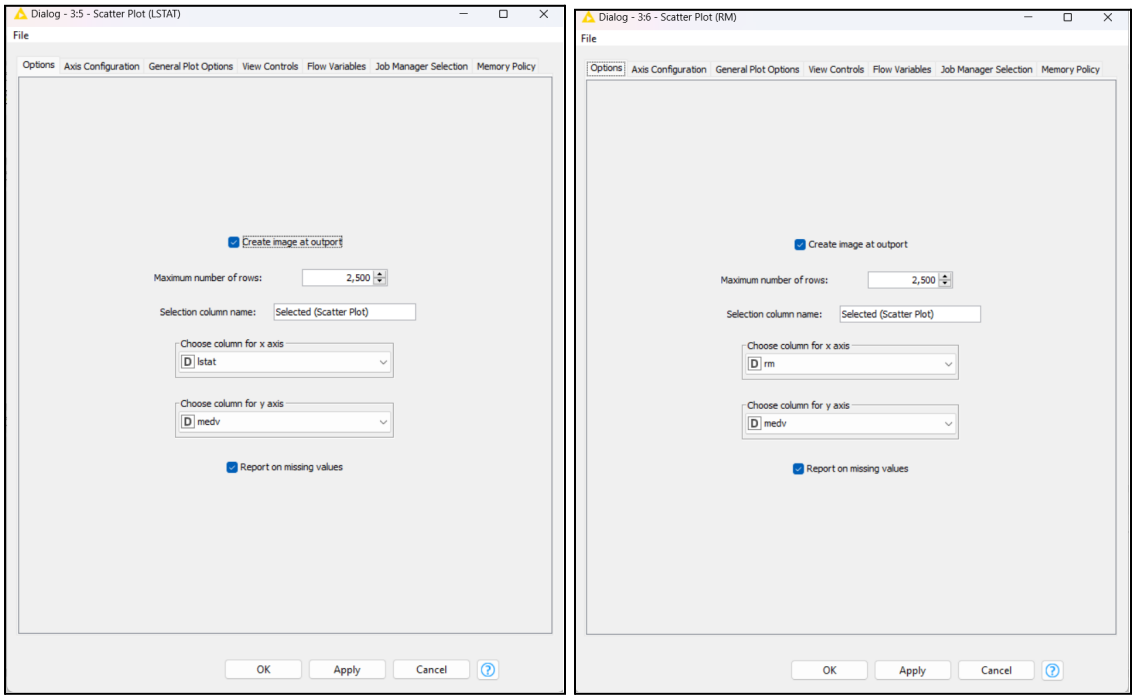
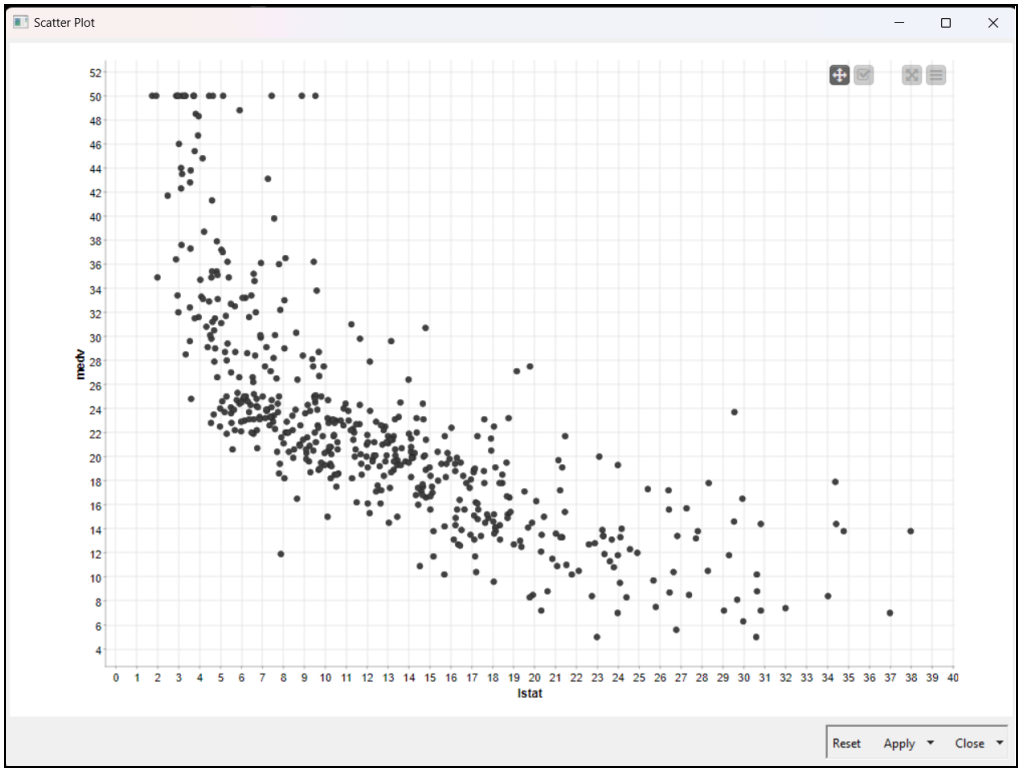


Figure 1.1.3.14

15. Now double click on each scatter plot node and set the column for x-axis and y-axis with “lstat” against “medv” and “rm” against “medv” respectively. After that, click the “OK” button and execute both nodes.



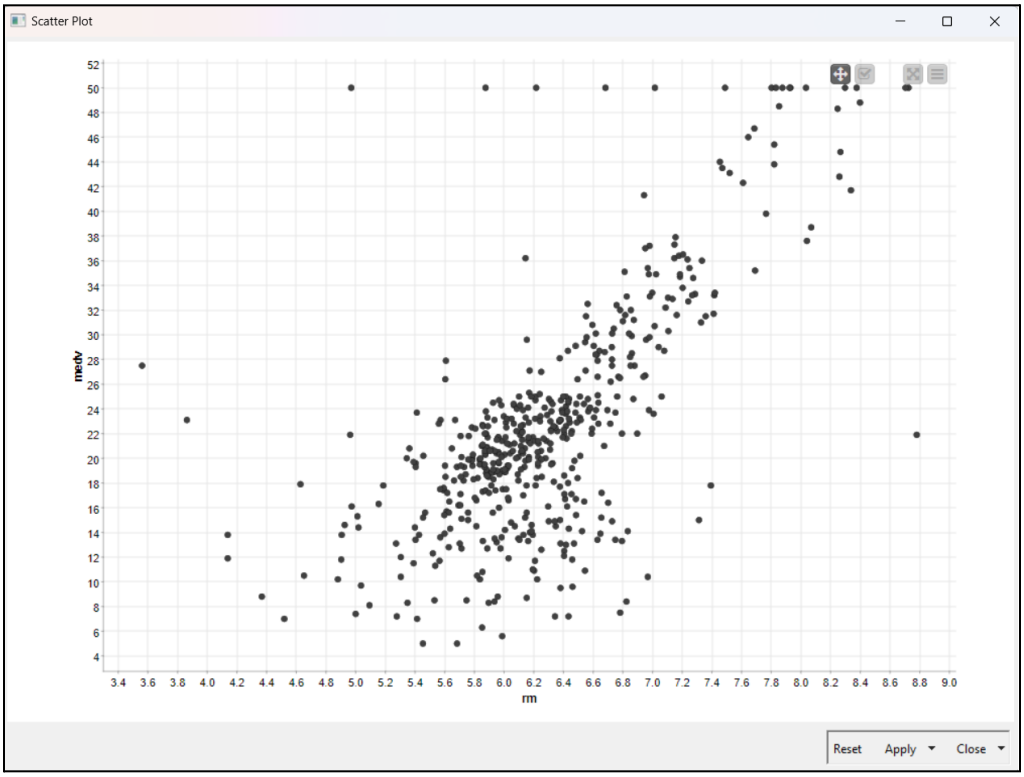


Figure 1.1.3.15

16. So, these are the scatter plots for both “lstat” against “medv” and “rm” against “medv”. Based on my observations, The price of the house (medv) increases as the value of RM increases linearly. There are a few outliers and the data seems to be capped at 50. The price of the house tend to decrease with an increase in “lstat”. Though it does not look to be following exactly a linear line.

1.1.4 Dataset 3: Boston.csv (Clustering Rule - K-means)

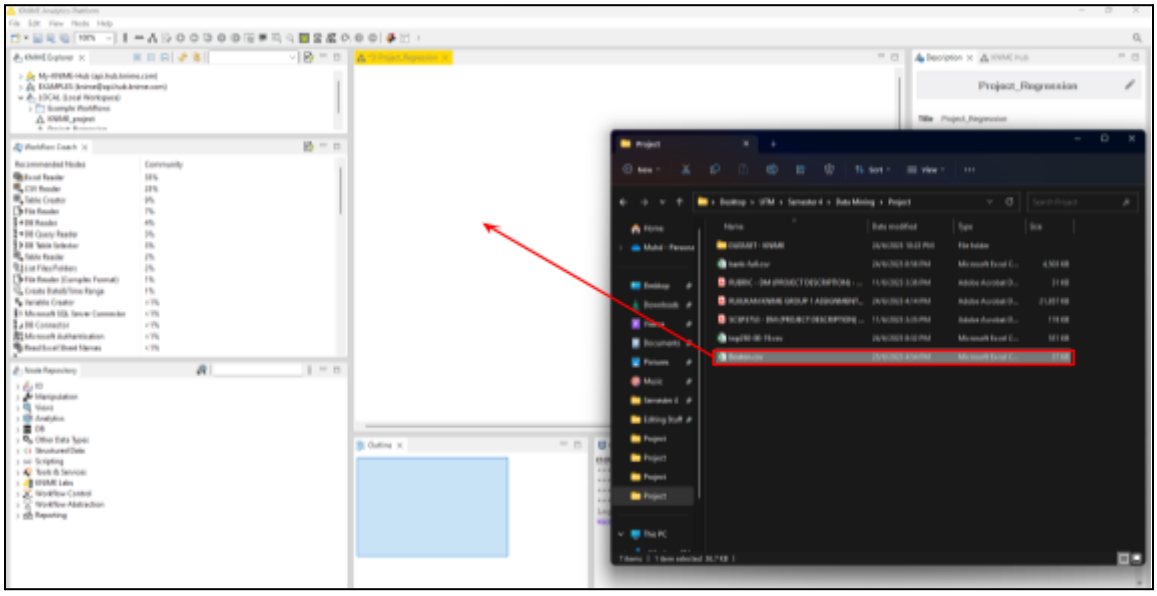


Figure 1.1.4.1

1. Drag the dataset Boston.csv file from the computer to the KNIME work area canvas.

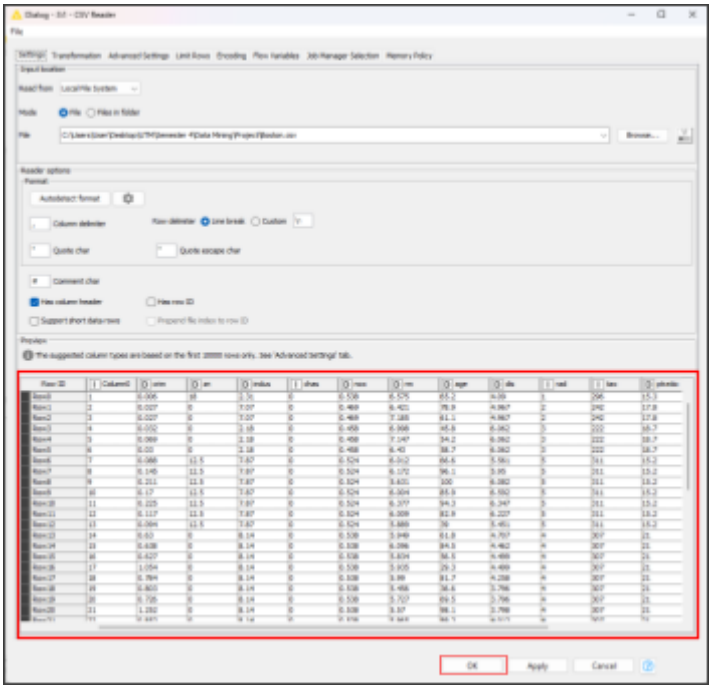


Figure 1.1.4.2

2. A new window will open. From the window, you can see the preview of the data that has been loaded to the canvas. Click “OK”.

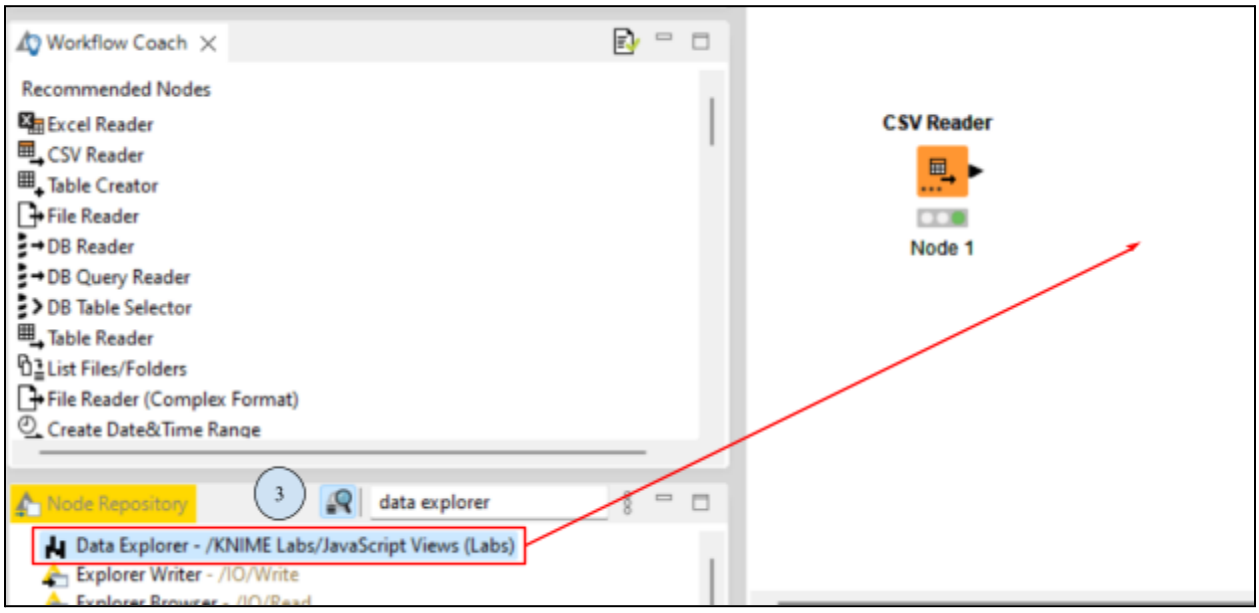


Figure 1.1.4.3

3. Insert the “Data Explorer” node to the work area. Please note that if you cannot find the “Data Explorer” node from the node repository, Please install the "KNIME Labs Extensions". Go to File -> Install KNIME Extension, then type that in.

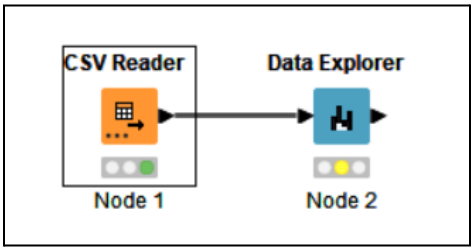


Figure 1.1.4.4

4. Connect the “CSV Reader” node by dragging the arrow from the node to the “Data Explorer” node.

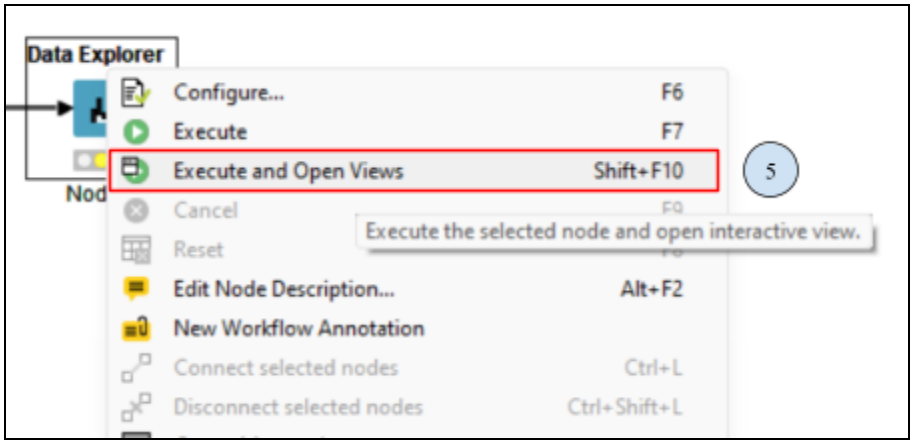


Figure 1.1.4.5

5. Right click on the "Data Explorer" node and then click on the "Execute and Open Views". A new window will open right after that.

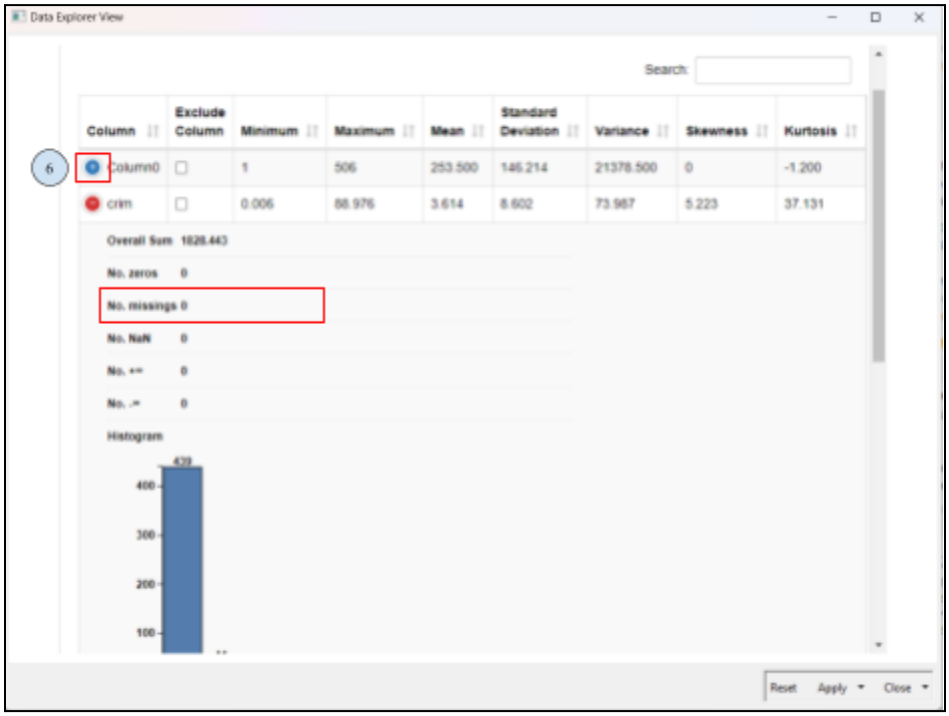


Figure 1.1.4.6

6. Click on the "+" button on the left side for all variables in the column, it shows the details of each column in our dataset and histogram visualization according to that column. We need to check for missing values in that column before proceeding with the data mining task.

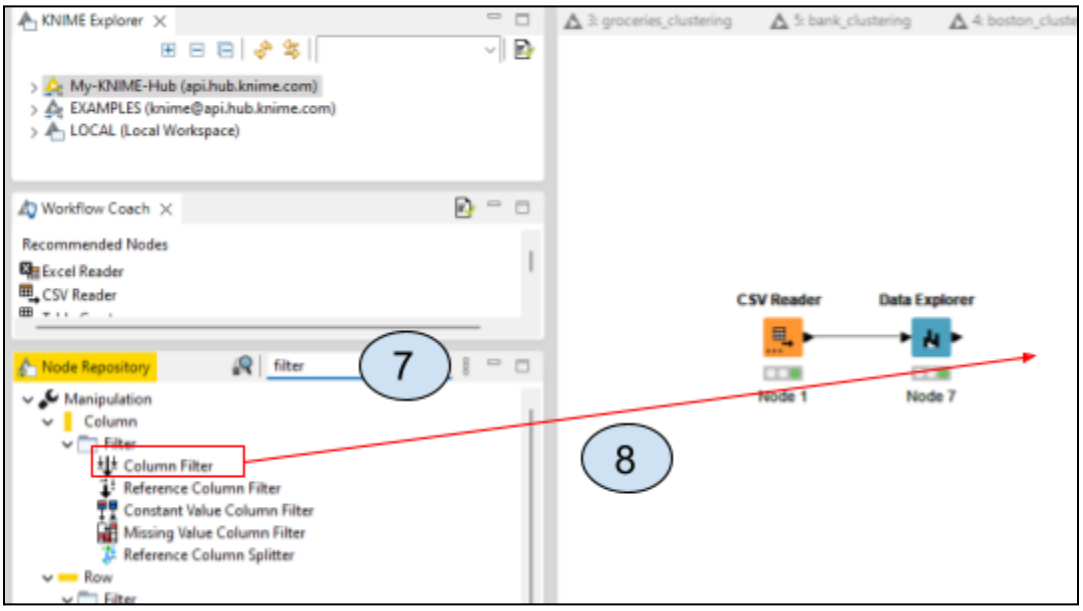


Figure 1.1.4.7

7. Search “column filter” in the Node Repository to filter specific attributes for clustering.
8. Drag the “column filter” put it in the canvas.

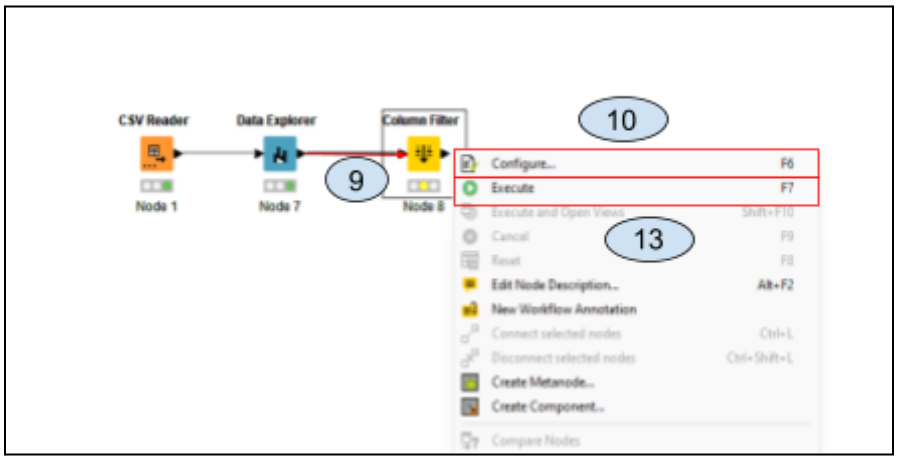


Figure 1.1.4.8

9. Connect the nodes
10. Configure the node column filter

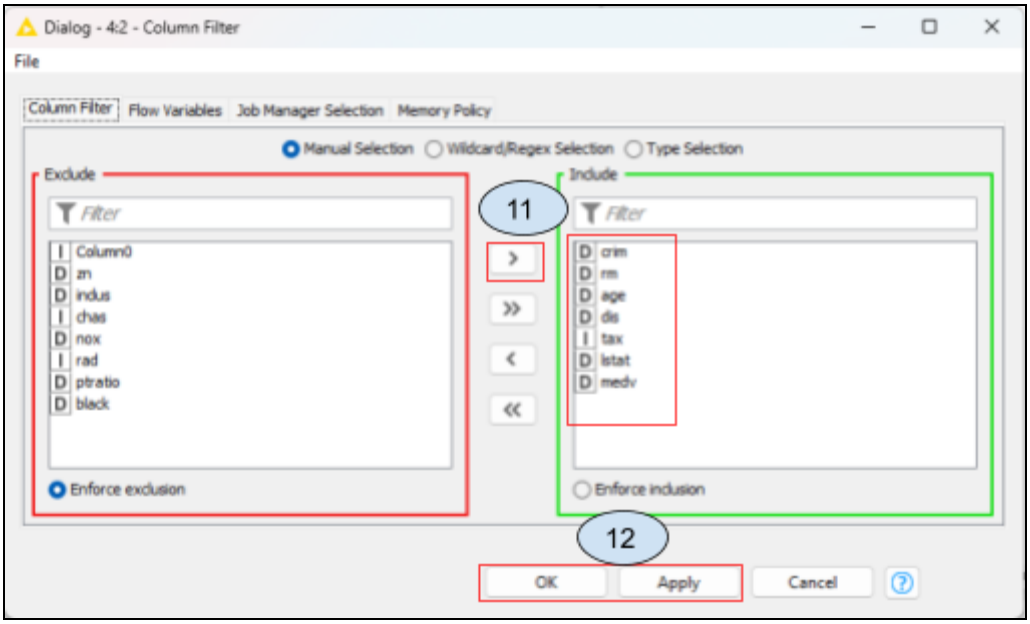


Figure 1.1.4.9

11. In the column filter configuration, we select a certain attributes that suitable for our clustering algorithm and put it into the include section *the green section*.
12. Click Apply and OK.
13. Execute after done filtering the column.

1.2 Data Mining Task

1.2.1 Association Rule - Apriori

- **Dataset 2: groceriesItem.csv**

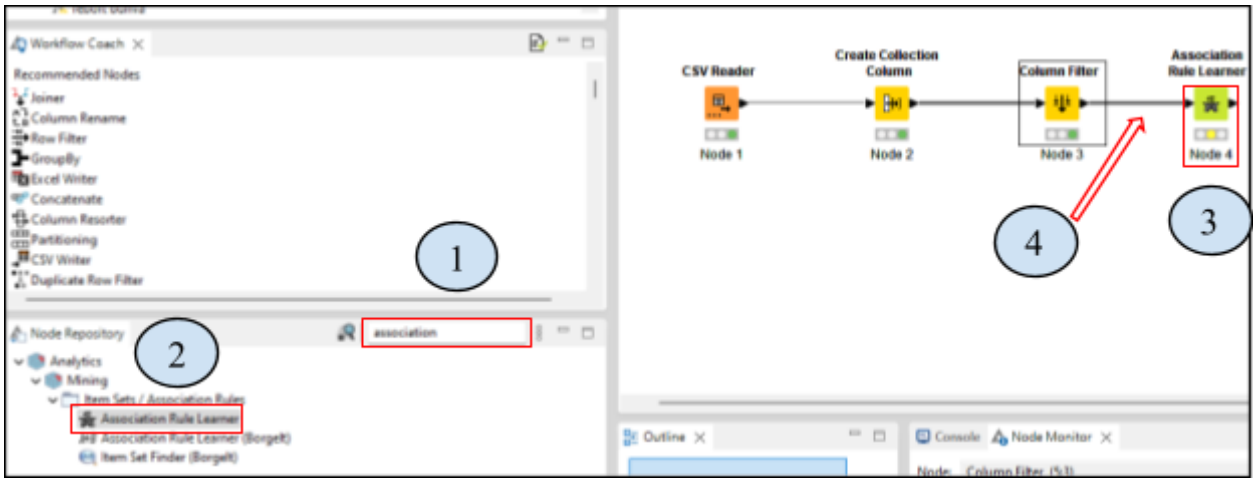


Figure 1.2.1.1

1. Search for Association Rule Learner and drag into the canvas
2. Connect the Column Filter node to the added node.

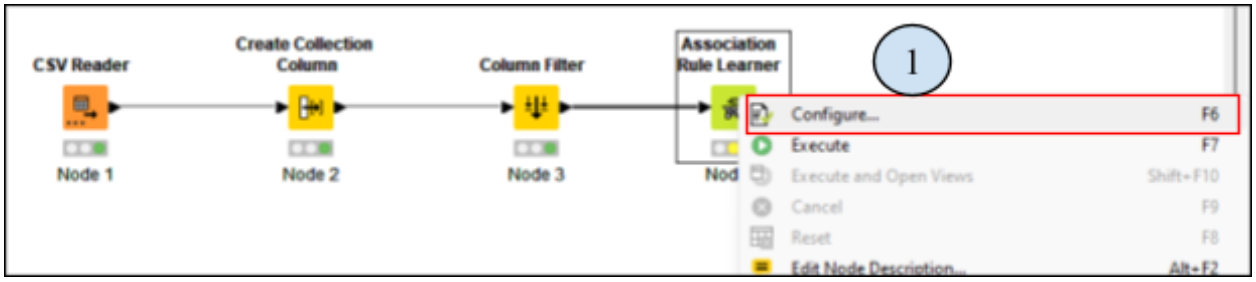


Figure 1.2.1.2

3. Double click on the added node and click configure.

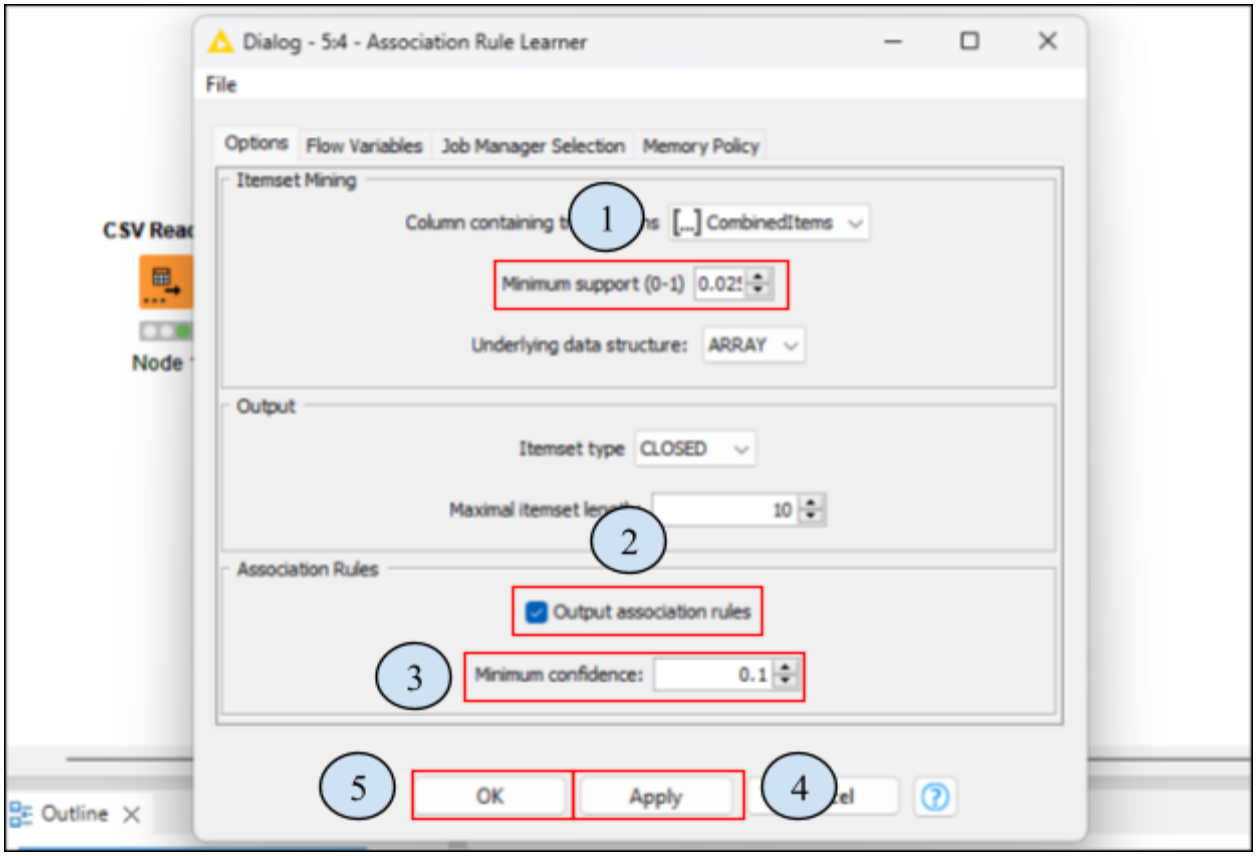


Figure 1.2.1.3

4. A new window will pop up.
5. Set the support to 0.025.
6. Check the "Output association rules box.
7. Set the confidence to 0.1.
8. Click Apply and OK.

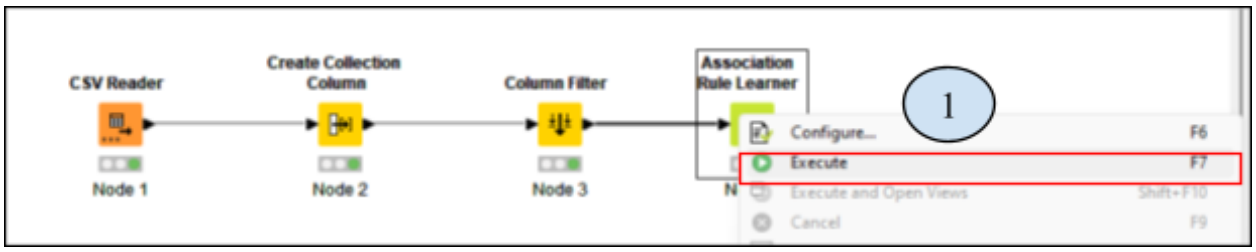


Figure 1.2.1.4

9. Double click on the Association Rule node.
10. Click execute.

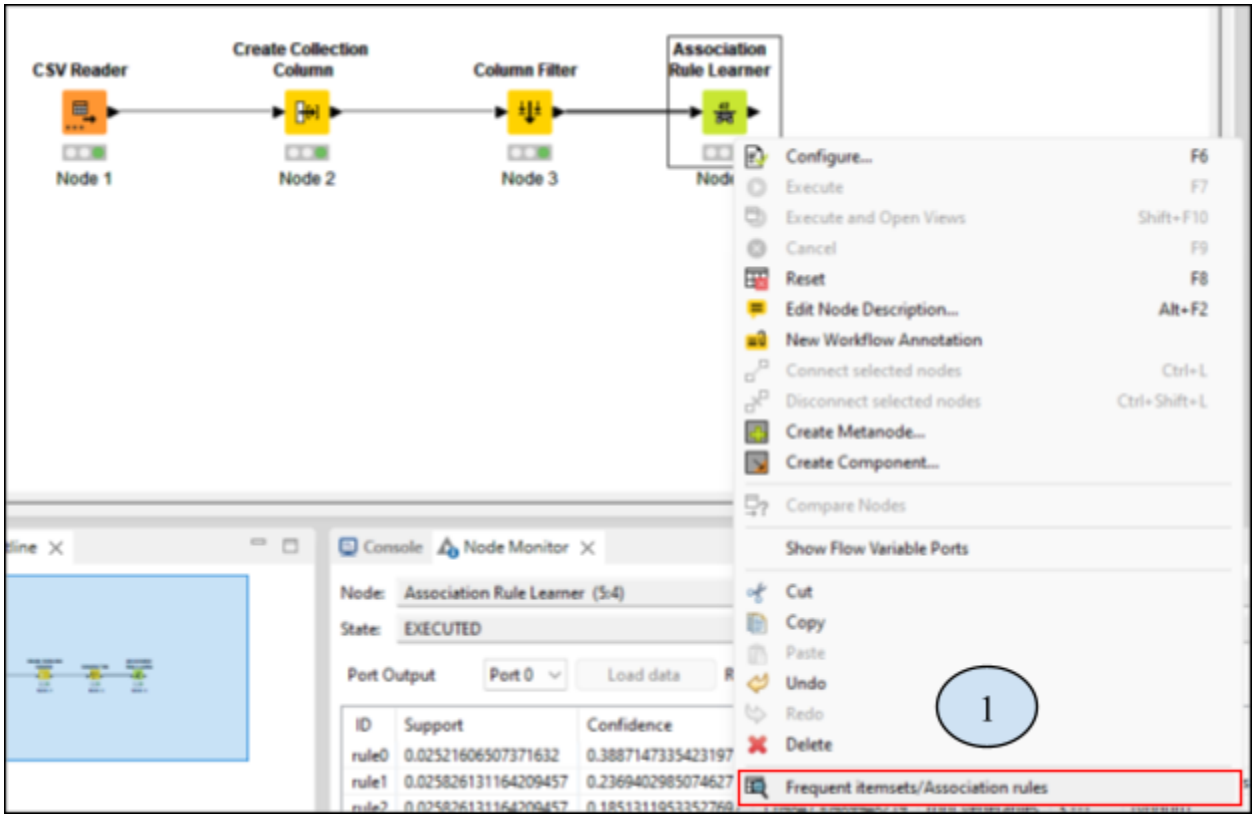


Figure 1.2.1.5

11. To view the associated output, right click on the node.
12. Click the “Frequent Itemsets”.

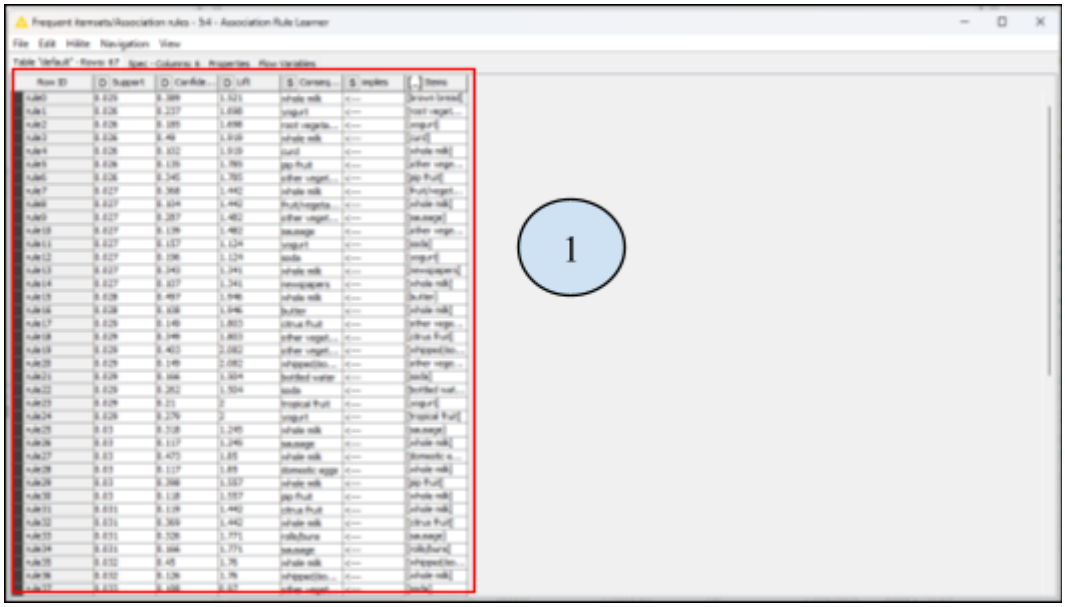


Figure 1.2.1.6

13. Now we have the output executed with the Association Rule Learner.

Figure 1.2.1.7

14. Search for Top K Selector, drag the node into the canvas.
15. Connect the output of Association Rule node to the added node.
16. Double click the added node.
17. Click configure.

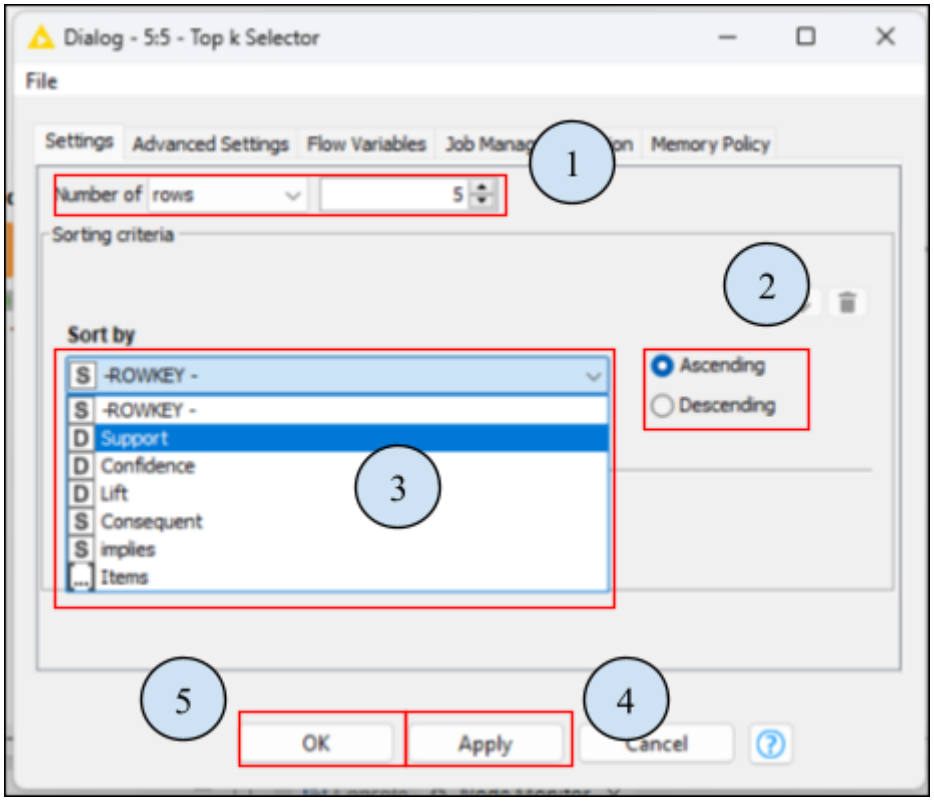


Figure 1.2.1.8

18. A new window will pop up.
19. This node serves as a visualization to display certain values in a certain way either in ascending or descending order.

20. Click the number of rows to determine the number of rows to be displayed.
21. Choose whether to display in which order.
22. Lastly choose which type of attribute to be displayed, then click Apply and OK.

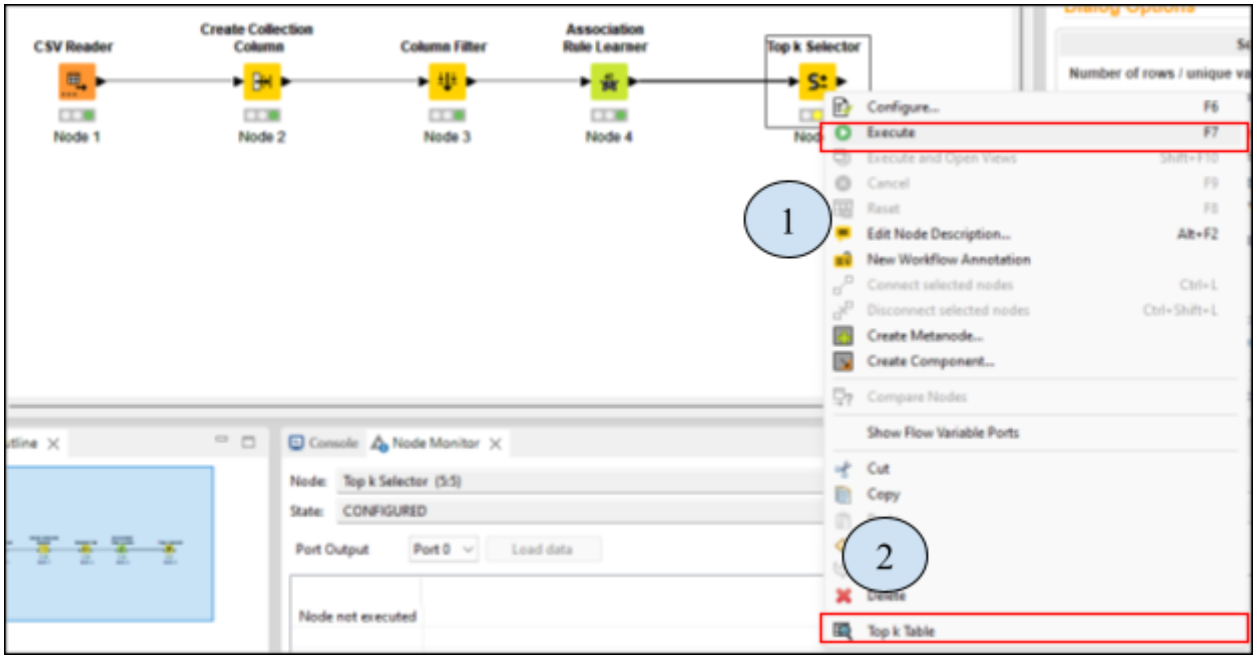
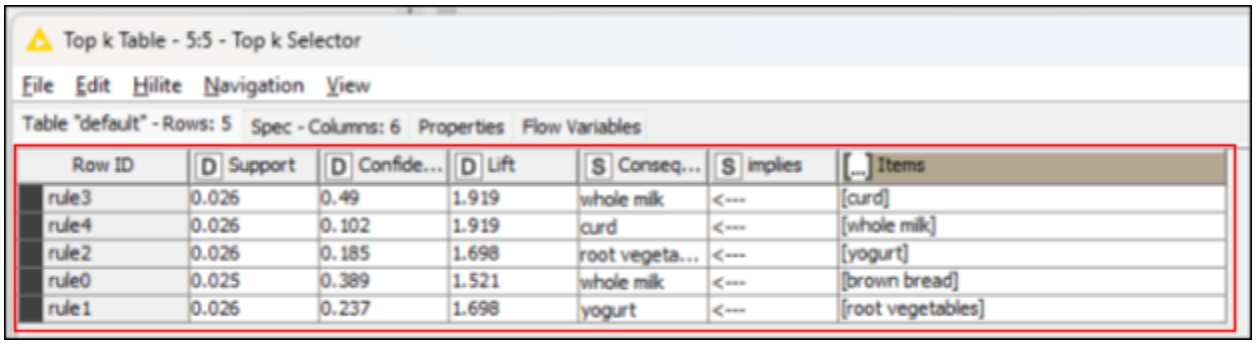


Figure 1.2.1.9

23. Double click on Top K Selector node and click execute.
24. Click the Top K table to see the result.



Row ID	Support	Confide...	Lift	Conseq...	implies	Items
rule3	0.026	0.49	1.919	whole milk	<---	[curd]
rule4	0.026	0.102	1.919	curd	<---	[whole milk]
rule2	0.026	0.185	1.698	root vegeta...	<---	[yogurt]
rule0	0.025	0.389	1.521	whole milk	<---	[brown bread]
rule1	0.026	0.237	1.698	yogurt	<---	[root vegetables]

Figure 1.2.1.10

25. A new window popped up showing the output with the desired settings.
26. This data shows that item curd

Result Discussion

Dataset : groceriesItem.csv (e.g. curd -> whole milk)

Items: The association rule consists of two items: "curd" and "whole milk." This indicates that there is an association or relationship between these two items in the dataset.

Lift: The lift value represents the measure of how much more likely the items "curd" and "whole milk" are to appear together compared to if they were independent of each other. A lift value of 1.919 suggests that the occurrence of "curd" is about 1.919 times more likely when "whole milk" is present compared to its expected frequency if the two items were unrelated.

Confidence: The confidence value indicates the conditional probability of finding the consequent item ("whole milk") given the antecedent item ("curd"). In this case, a confidence value of 0.49 means that "whole milk" is selected in approximately 49% of the transactions where "curd" is present.

Support: The support value represents the frequency or proportion of transactions that contain both "curd" and "whole milk." A support value of 0.026 indicates that about 2.6% of the transactions in the dataset include both items "curd" and "whole milk."

In summary, the association rule suggests that there is a moderate association between "curd" and "whole milk" in the dataset. The lift value indicates that the occurrence of "curd" is positively influenced by the presence of "whole milk." The confidence value tells us that when "curd" is present, there is a 49% chance of "whole milk" being present as well. The support value indicates that around 2.6% of the transactions contain both "curd" and "whole milk."

1.2.2 Classification - Decision Tree

- **Dataset 1: bank-full.csv**

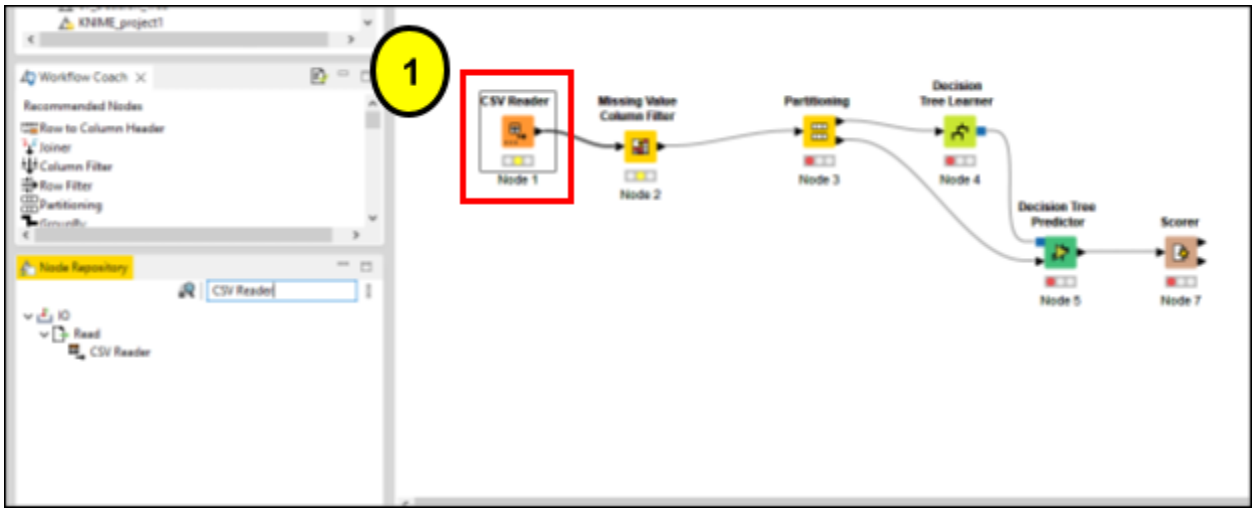


Figure 1.2.2.1

1. Create a new workflow and search CSV Reader in Node Repository. Drag the CSV Reader into the canvas.

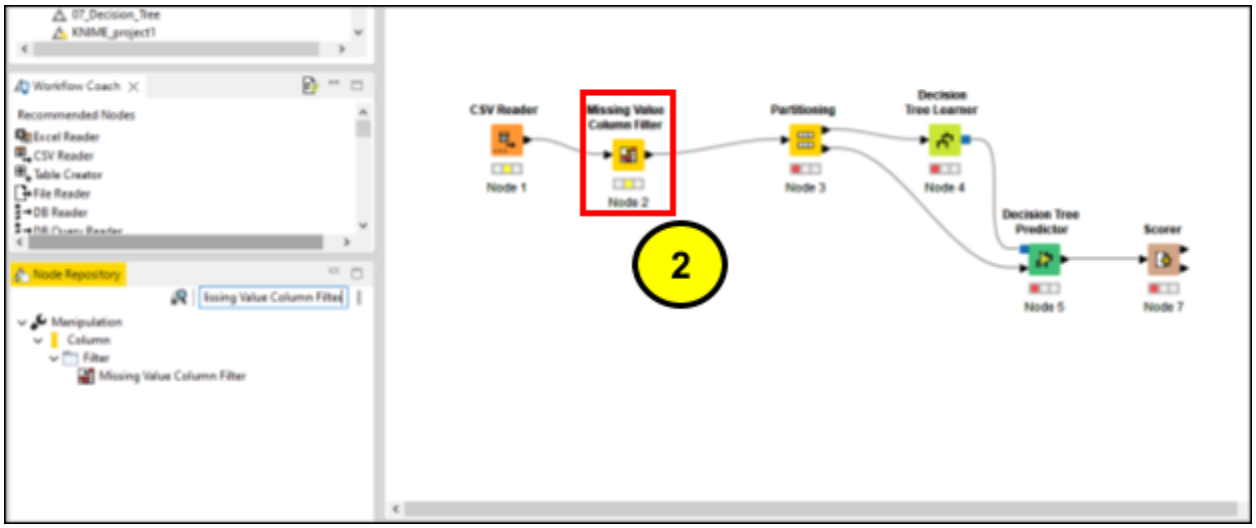


Figure 1.2.2.2

2. Search for Missing Value Column Filter in the Node Repository and drag it into the canvas. Then, create a connection between CSV Reader node and Missing Value Column

Filter node. Missing Value Column Filter node require to removes all columns from the input table which contain more missing values than a certain percentage.

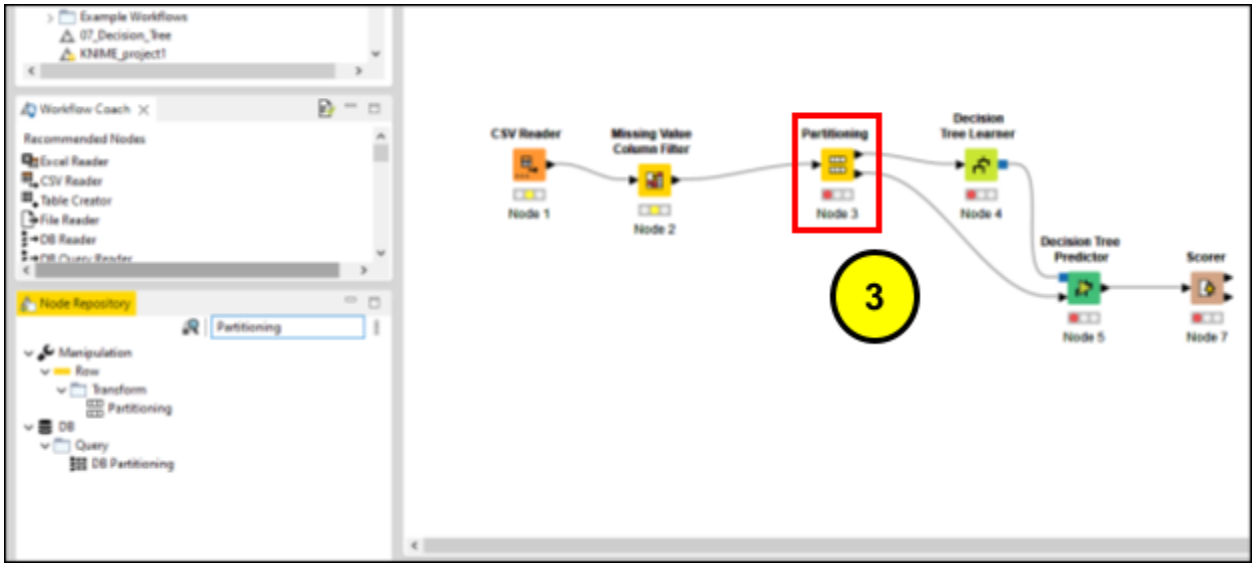


Figure 1.2.2.3

3. Search for Partitioning in the Node Repository and drag it into the canvas. Create a connection between Missing Value Column Filter node and Partitioning. This is where the input table is split into two partitions which are train data and test data.

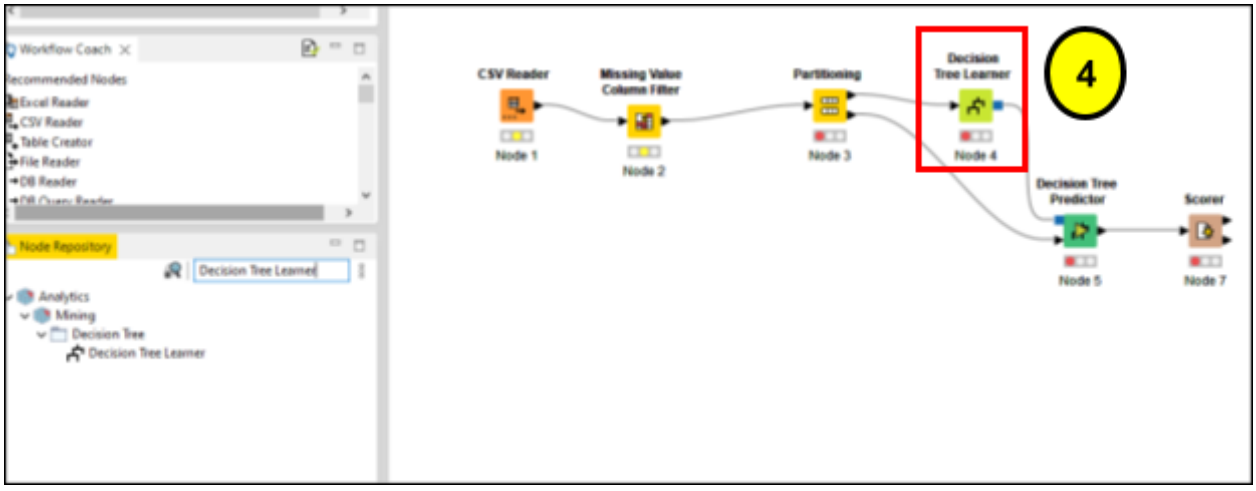


Figure 1.2.2.4

4. Search for Decision Tree Learner in the Node Repository and drag it into the canvas. Create a connection by using Partitioning 0 output port (First partition) with Decision Tree Learner input port.

Figure 1.2.2.5

5. Search for Decision Tree Predictor in the Node Repository and drag it into the canvas. Create a connection by using Partitioning 1 output port (Second partition) with Decision Tree Predictor 1 input port. Create a connection also between Decision Tree Learner and Decision Tree Predictor.

Figure 1.2.2.6

6. Search for Scorer in the Node Repository and drag it into the canvas. Create a connection between Decision Tree Predictor and Scorer.

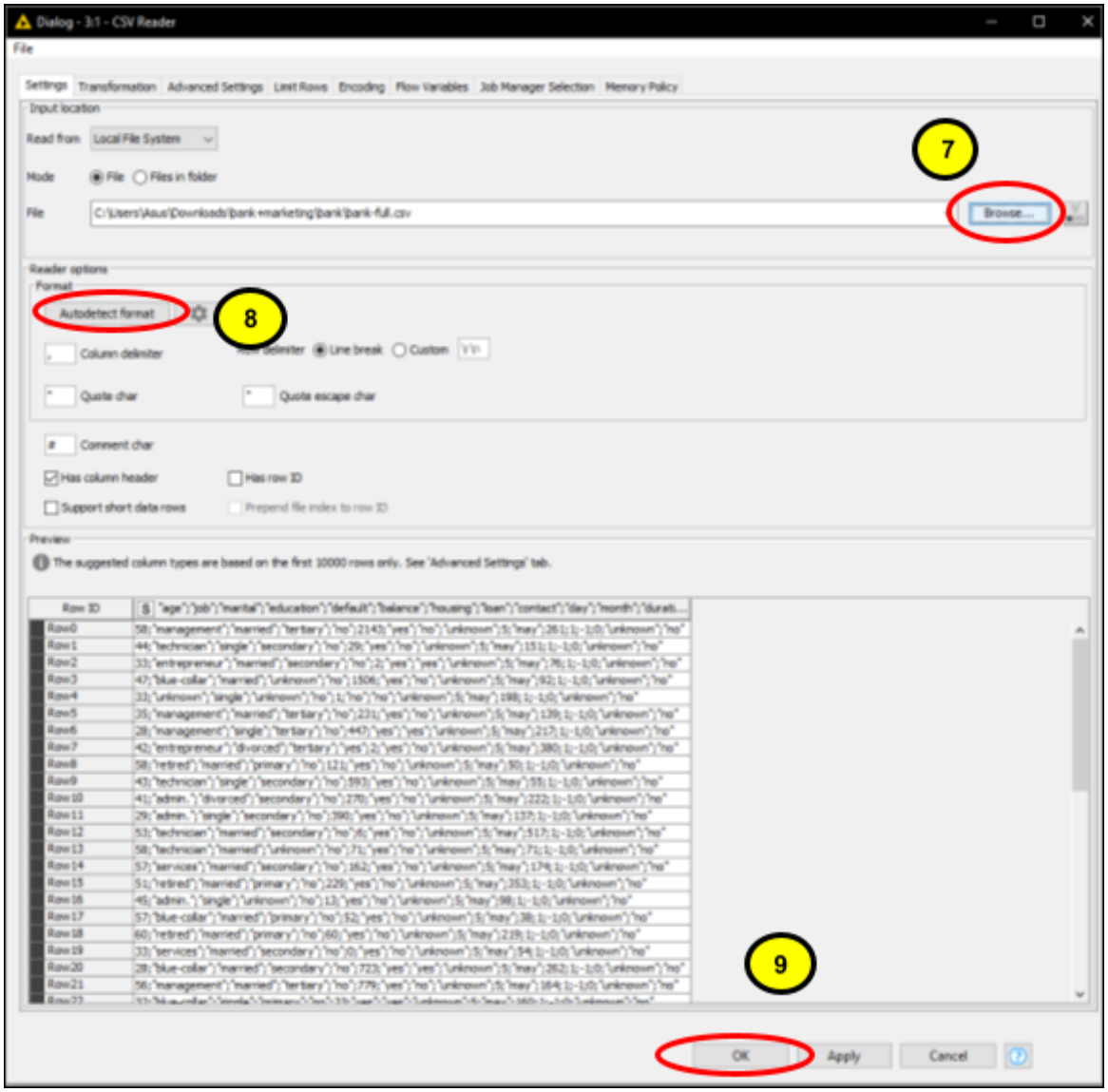


Figure 1.2.2.7

7. Then, you right click on the CSV Reader node and click Configure. A windows shown as above will pop up and click on Browse and select .csv file. Next, click Autodetect format and click OK.

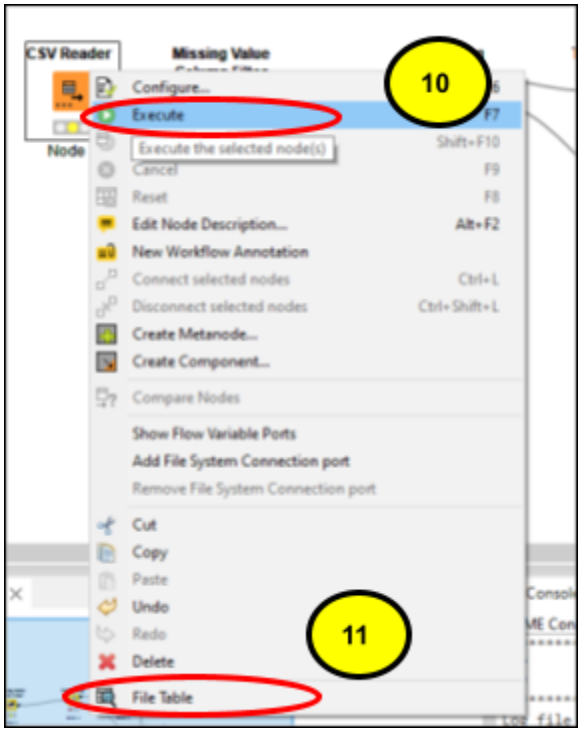


Figure 1.2.2.8

File Table - 4:1 - CSV Reader

File Edit Hilite Navigation View

Table "default" - Rows: 45211 | Spec - Columns: 17 | Properties | Flow Variables

Row ID	age	job	marital	education	default	balance	housing	l
Row0	58	management	married	tertiary	no	2143	yes	no
Row1	44	technician	single	secondary	no	29	yes	no
Row2	33	entrepreneur	married	secondary	no	2	yes	yes
Row3	47	blue-collar	married	unknown	no	1506	yes	no
Row4	33	unknown	single	unknown	no	1	no	no
Row5	35	management	married	tertiary	no	231	yes	no
Row6	28	management	single	tertiary	no	447	yes	yes
Row7	42	entrepreneur	divorced	tertiary	yes	2	yes	no
Row8	58	retired	married	primary	no	121	yes	no
Row9	43	technician	single	secondary	no	593	yes	no
Row10	41	admin.	divorced	secondary	no	270	yes	no
Row11	29	admin.	single	secondary	no	390	yes	no
Row12	53	technician	married	secondary	no	6	yes	no
Row13	58	technician	married	unknown	no	71	yes	no
Row14	57	services	married	secondary	no	162	yes	no
Row15	51	retired	married	primary	no	229	yes	no
Row16	45	admin.	single	unknown	no	13	yes	no
Row17	57	blue-collar	married	primary	no	52	yes	no
Row18	60	retired	married	primary	no	60	yes	no
Row19	33	services	married	secondary	no	0	yes	no
Row20	28	blue-collar	married	secondary	no	723	yes	yes
Row21	56	management	married	tertiary	no	779	yes	no
Row22	32	blue-collar	single	primary	no	23	yes	yes
Row23	25	services	married	secondary	no	50	yes	no
Row24	40	retired	married	primary	no	0	yes	yes
Row25	44	admin.	married	secondary	no	-372	yes	no
Row26	39	management	single	tertiary	no	255	yes	no

Figure 1.2.2.9

8. After finish configure, right click again on CSV Reader node and click execute. After finish execute, you can check your input table by right click on CSV Reader node and click File table.

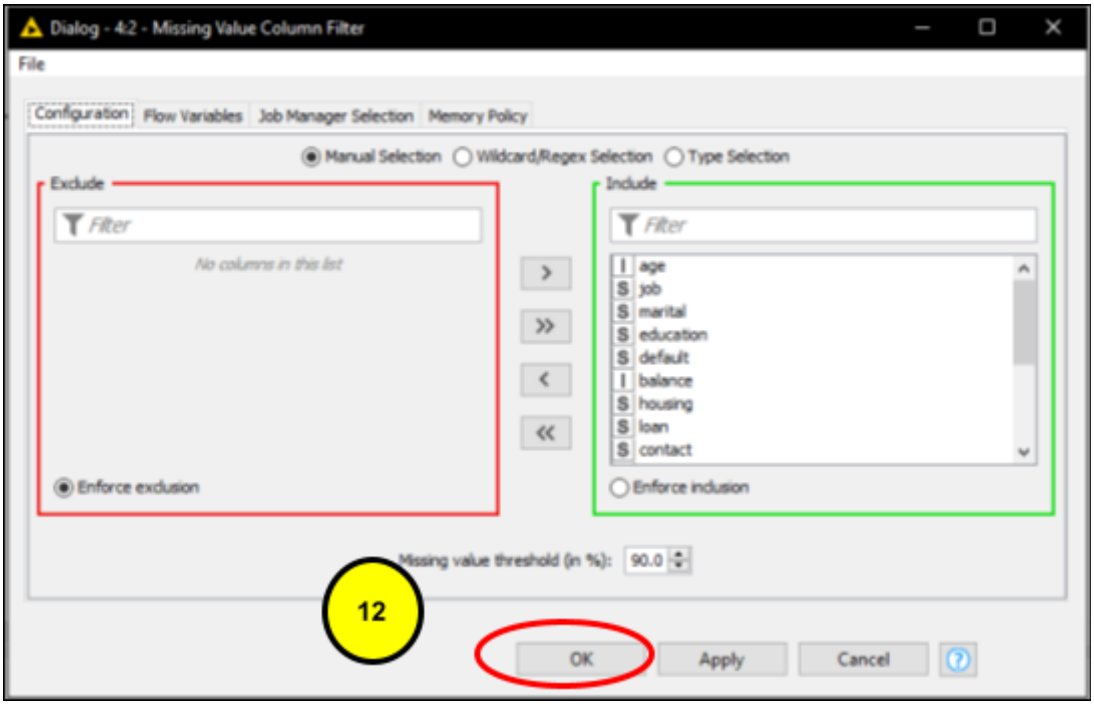


Figure 1.2.2.10

9. Right click on Missing Value Column Filter node and click Configure. Then, windows shown as above will pop up and click OK. Ensure that Missing value threshold is 90%.

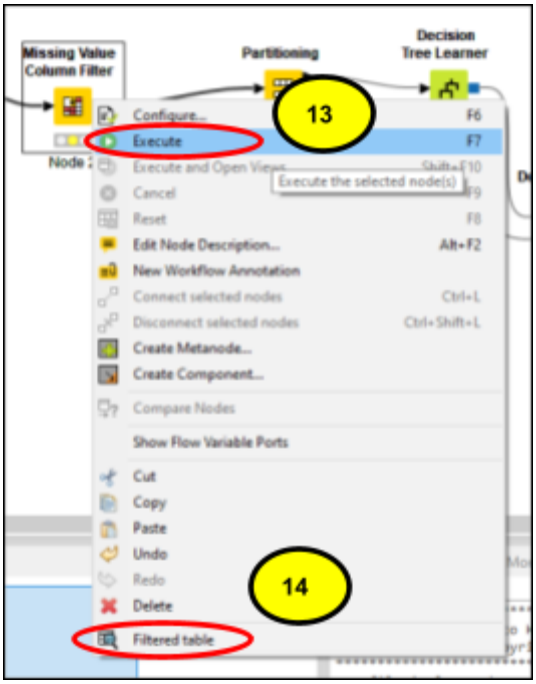


Figure 1.2.2.11

10. After finish configure, right click again on Missing Value Column Filter node and click execute. After finish execute, you can check your input table by right click on Missing Value Column Filter node and click Filtered table.

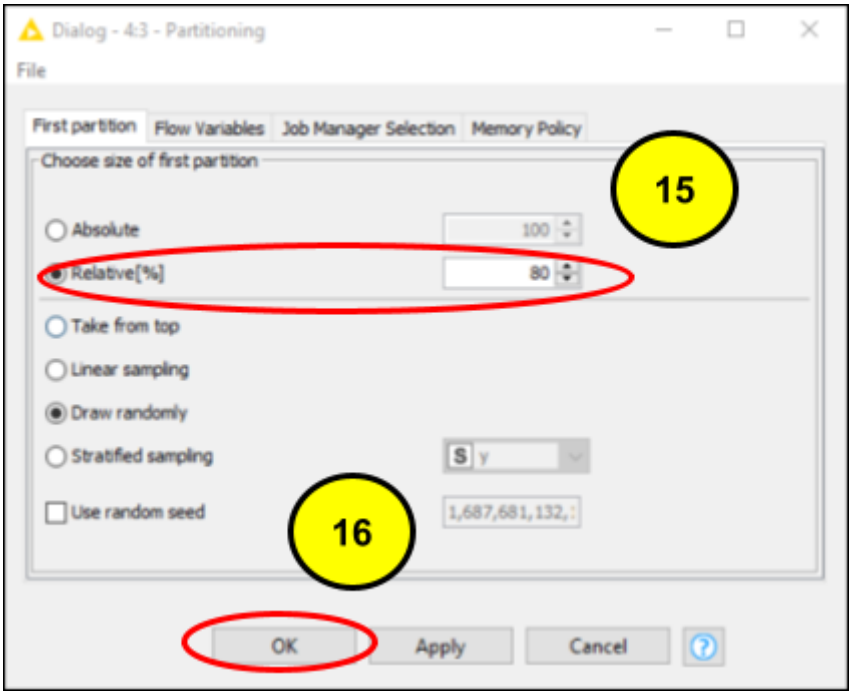


Figure 1.2.2.12

11. Next, right click on Partitioning node and click Configure. Partitioning node will split input table into two partitions. In this case, you just set the Relative[%] as 80 which means 80% for the first partition and 20% for the second partition. Then, click OK.

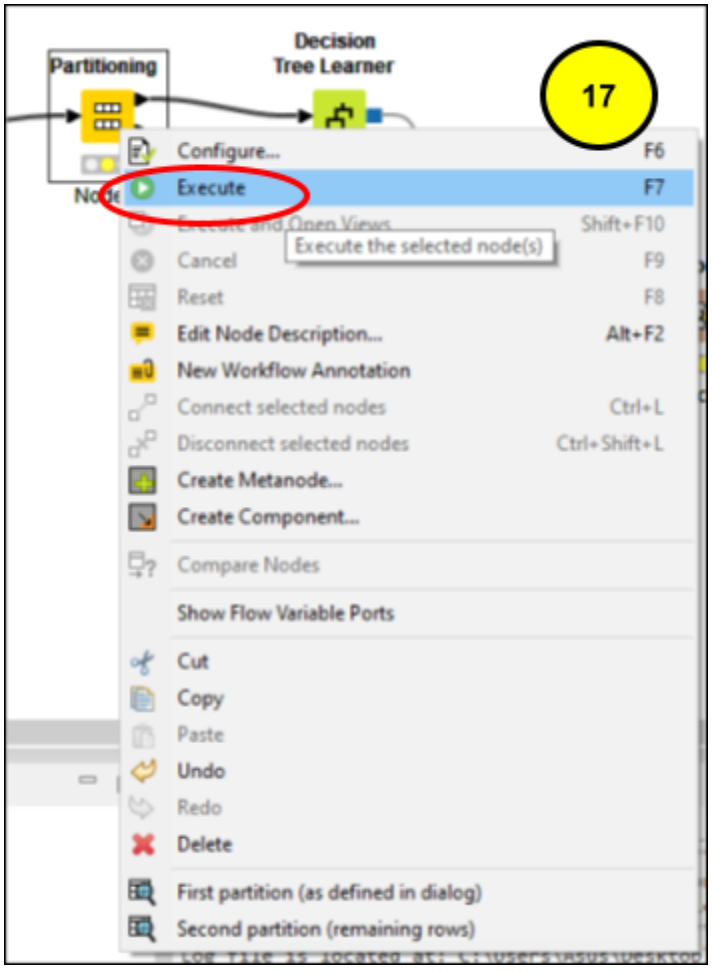


Figure 1.2.2.13

12. Right click on Partitioning node and click execute. If you want to check the partitions, you can right click on Partitioning node and click First partition and second partition.

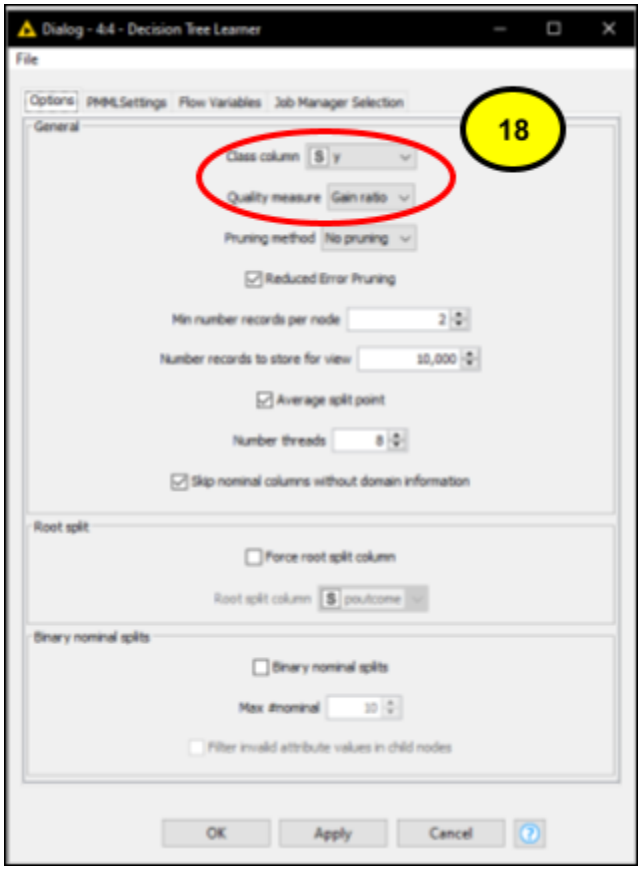


Figure 1.2.2.14

13. Right click on Decision Tree Learner node and click configure. Then, for the class column choose y which is the column that we will predict. Set the quality measure as Gain ratio. Next, click OK.

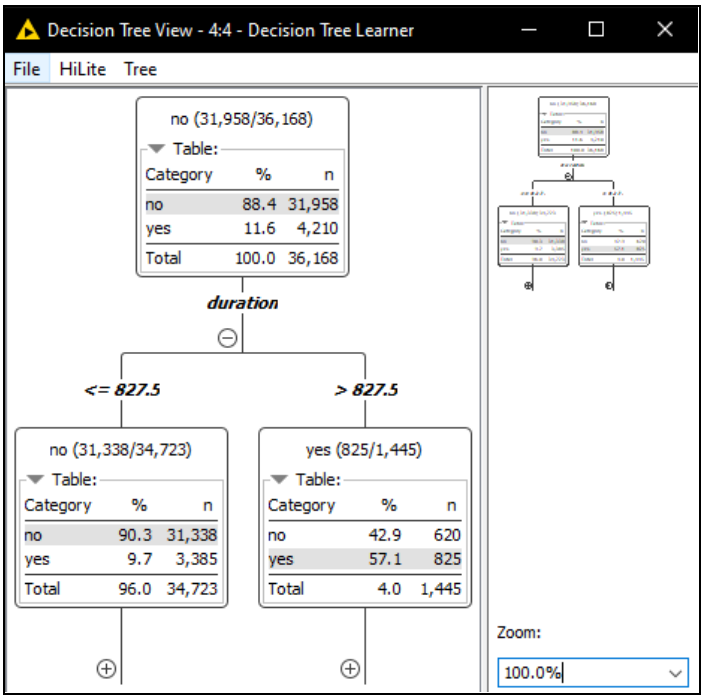


Figure 1.2.2.15

14. Right click on Decision Tree Learner and click execution. If you want to see Decision Tree you can right click on the node and click Decision Tree Learner.

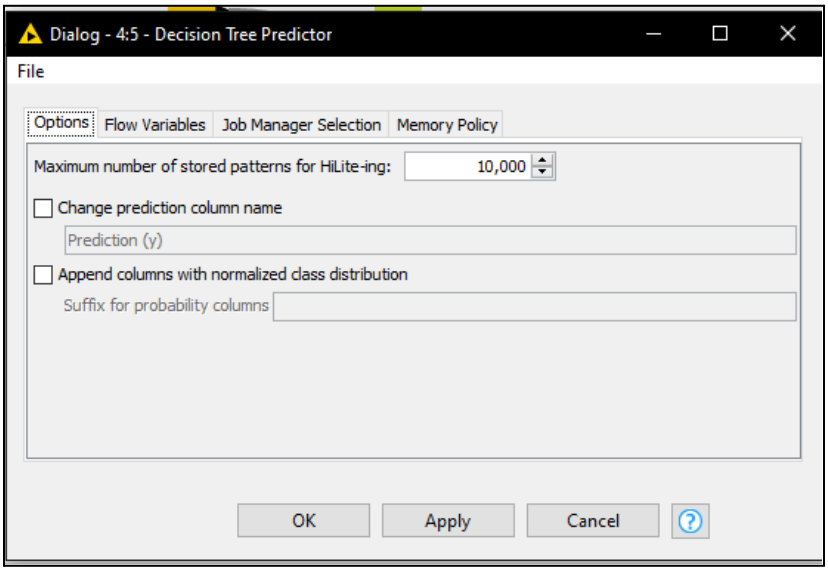
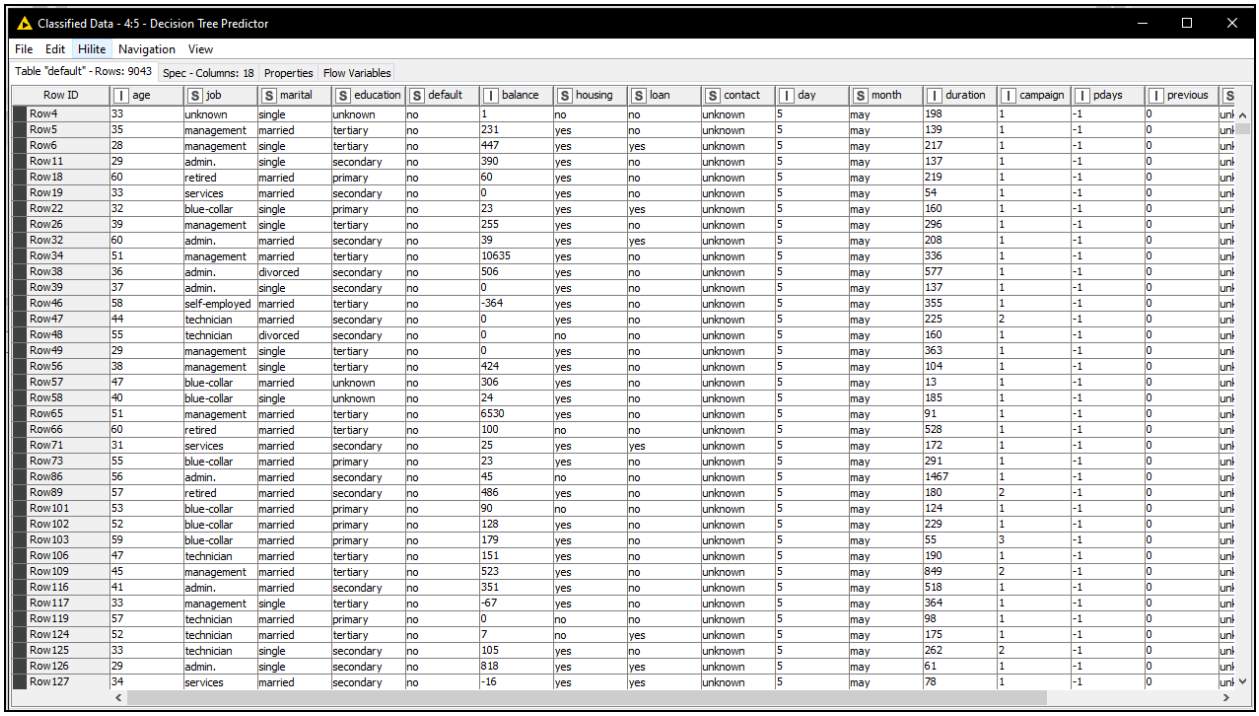


Figure 1.2.2.16

15. Right click on Decision Tree Predictor node and click configure. Then, click OK.



Row ID	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	
Row4	33	unknown	single	unknown	no	1	no	no	unknown	5	may	198	1	-1	0	uni
Row5	35	management	married	tertiary	no	231	yes	no	unknown	5	may	139	1	-1	0	uni
Row6	28	management	single	tertiary	no	447	yes	yes	unknown	5	may	217	1	-1	0	uni
Row11	29	admin.	single	secondary	no	390	yes	no	unknown	5	may	137	1	-1	0	uni
Row18	60	retired	married	primary	no	60	yes	no	unknown	5	may	219	1	-1	0	uni
Row19	33	services	married	secondary	no	0	yes	no	unknown	5	may	54	1	-1	0	uni
Row22	32	blue-collar	single	primary	no	23	yes	yes	unknown	5	may	160	1	-1	0	uni
Row26	39	management	single	tertiary	no	255	yes	no	unknown	5	may	296	1	-1	0	uni
Row32	60	admin.	married	secondary	no	39	yes	yes	unknown	5	may	208	1	-1	0	uni
Row34	51	management	married	tertiary	no	10635	yes	no	unknown	5	may	336	1	-1	0	uni
Row38	36	admin.	divorced	secondary	no	506	yes	no	unknown	5	may	577	1	-1	0	uni
Row39	37	admin.	single	secondary	no	0	yes	no	unknown	5	may	137	1	-1	0	uni
Row46	58	self-employed	married	tertiary	no	-364	yes	no	unknown	5	may	355	1	-1	0	uni
Row47	44	technician	married	secondary	no	0	yes	no	unknown	5	may	225	2	-1	0	uni
Row48	55	technician	divorced	secondary	no	0	no	no	unknown	5	may	160	1	-1	0	uni
Row49	29	management	single	tertiary	no	0	yes	no	unknown	5	may	363	1	-1	0	uni
Row56	38	management	single	tertiary	no	424	yes	no	unknown	5	may	104	1	-1	0	uni
Row57	47	blue-collar	married	unknown	no	306	yes	no	unknown	5	may	13	1	-1	0	uni
Row58	40	blue-collar	single	unknown	no	24	yes	no	unknown	5	may	185	1	-1	0	uni
Row65	51	management	married	tertiary	no	6530	yes	no	unknown	5	may	91	1	-1	0	uni
Row66	60	retired	married	tertiary	no	100	no	no	unknown	5	may	528	1	-1	0	uni
Row71	31	services	married	secondary	no	25	yes	yes	unknown	5	may	172	1	-1	0	uni
Row73	55	blue-collar	married	primary	no	23	yes	no	unknown	5	may	291	1	-1	0	uni
Row86	56	admin.	married	secondary	no	45	no	no	unknown	5	may	1467	1	-1	0	uni
Row89	57	retired	married	secondary	no	486	yes	no	unknown	5	may	180	2	-1	0	uni
Row101	53	blue-collar	married	primary	no	90	no	no	unknown	5	may	124	1	-1	0	uni
Row102	52	blue-collar	married	primary	no	128	yes	no	unknown	5	may	229	1	-1	0	uni
Row103	59	blue-collar	married	primary	no	179	yes	no	unknown	5	may	55	3	-1	0	uni
Row106	47	technician	married	tertiary	no	151	yes	no	unknown	5	may	190	1	-1	0	uni
Row109	45	management	married	tertiary	no	523	yes	no	unknown	5	may	849	2	-1	0	uni
Row116	41	admin.	married	secondary	no	351	yes	no	unknown	5	may	518	1	-1	0	uni
Row117	33	management	single	tertiary	no	-67	yes	no	unknown	5	may	364	1	-1	0	uni
Row119	57	technician	married	primary	no	0	no	no	unknown	5	may	98	1	-1	0	uni
Row124	52	technician	married	tertiary	no	7	no	yes	unknown	5	may	175	1	-1	0	uni
Row125	33	technician	single	secondary	no	105	yes	no	unknown	5	may	262	2	-1	0	uni
Row126	29	admin.	single	secondary	no	818	yes	yes	unknown	5	may	61	1	-1	0	uni
Row127	34	services	married	secondary	no	-16	yes	yes	unknown	5	may	78	1	-1	0	uni

Figure 1.2.2.17

16. To have a look table as shown above you need to right click on the Decision Tree Predictor node and click execution. After finish execute, you can click Classified Data. You can notice that at the last column there will be a new column named as prediction that you just made.

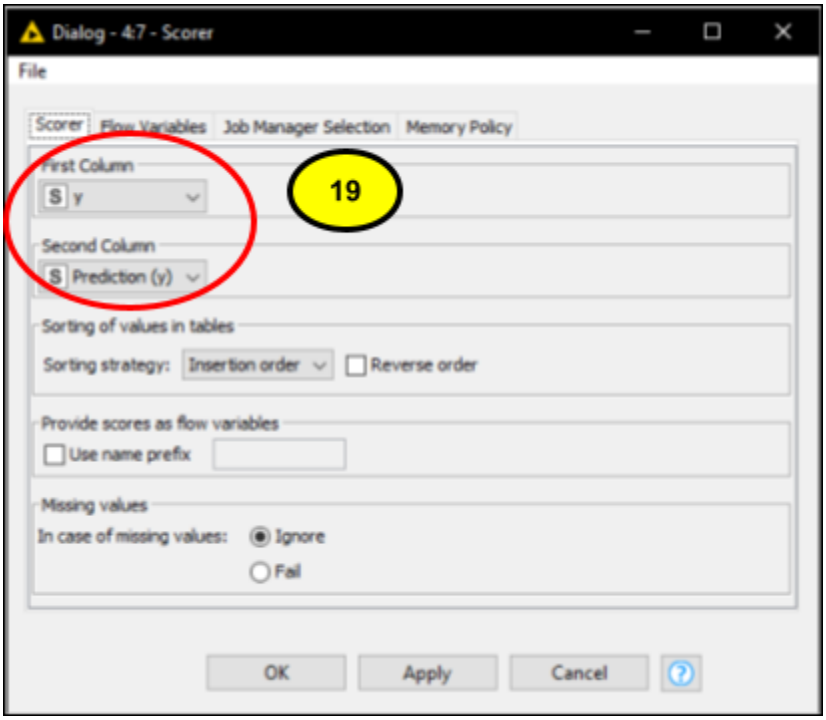
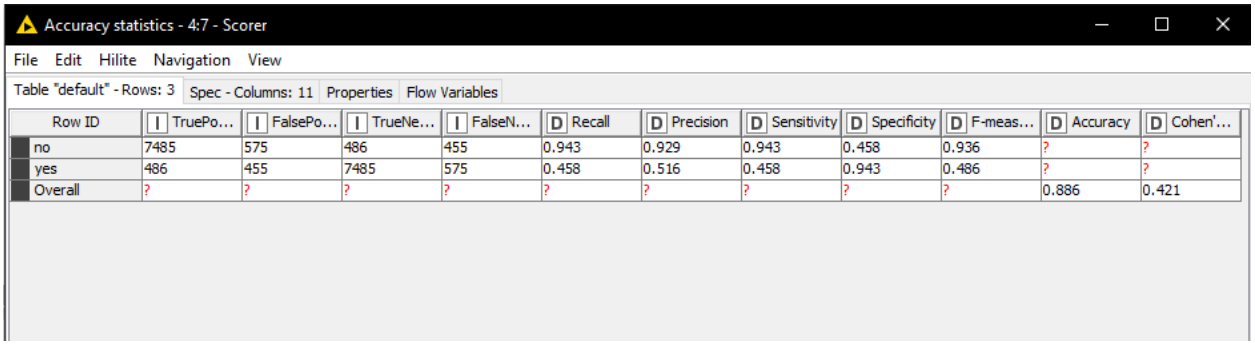


Figure 1.2.2.18

17. Lastly, you right click on Scorer node and click Configure. Then, you set the first column as represents the real classes of data which is column y. Set second column as prediction(y) as represents the predicted classes of the data. After that, click OK.



Row ID	TruePo...	FalsePo...	TrueNe...	FalseN...	Recall	Precision	Sensitivity	Specificity	F-meas...	Accuracy	Cohen'...
no	7485	575	486	455	0.943	0.929	0.943	0.458	0.936	?	?
yes	486	455	7485	575	0.458	0.516	0.458	0.943	0.486	?	?
Overall	?	?	?	?	?	?	?	?	?	0.886	0.421

Figure 1.2.2.19

18. Right click on Scorer node and click execute. Next, right click again on the Scorer node and click Accuracy statistics to check accuracy statistics table.

Result Discussion

Dataset :bank-full.csv

The decision tree analysis was performed on a dataset with a total of 36,168 instances. The majority class in the dataset was "no," accounting for 88.4% of the instances, while the "yes" class represented 11.6% of the instances.

The decision tree split the data based on the "duration" feature, with a threshold of 827.5. Instances with a duration less than or equal to 827.5 were assigned to the left branch, while instances with a duration greater than 827.5 were assigned to the right branch.

In the left branch, the majority class remained "no," with a count of 31,338 out of 34,723 instances. This indicates that instances with a lower duration were more likely to be classified as "no" according to the decision tree. In the right branch, the majority class was "yes," but with a count of 825 out of 1,445 instances. This suggests that instances with a higher duration, above the threshold of 871.5, were more likely to be classified as "yes."

The accuracy statistics provide additional insights into the performance of the model. The "no" class achieved a high recall (sensitivity) of 0.943, indicating that it correctly identified a large proportion of the instances belonging to the "no" class. The specificity for the "no" class was also 0.458, indicating a low proportion of true negatives identified.

However, the "yes" class had a lower recall (sensitivity) of 0.458, indicating that the model had more difficulty correctly identifying instances belonging to the "yes" class. The specificity for the "yes" class was 0.943, suggesting a high proportion of true negatives identified.

The overall accuracy of the model was calculated to be 0.886, indicating that it correctly predicted the class for approximately 87.9% of the instances in the dataset. The Cohen's kappa coefficient of 0.421 suggests a fair agreement between the predicted and actual classes beyond what would be expected by chance.

1.2.3 Regression Rule - Linear Regression

- **Dataset 3: Boston.csv**

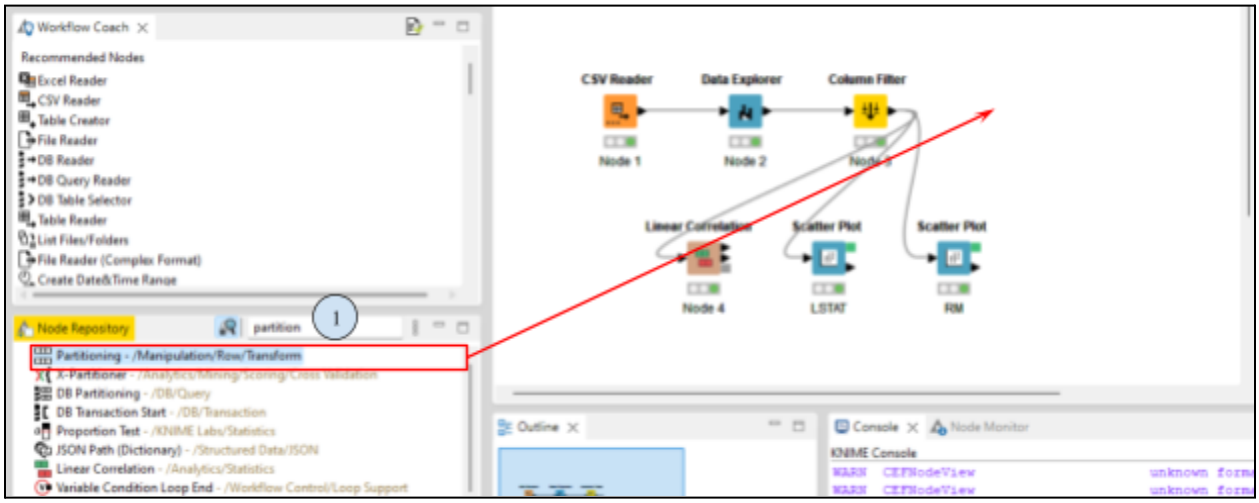


Figure 1.2.3.1

1. Search for the "Partitioning" node in the Node Repository and drag the node to the work area and connect the "Column Filter" node to the "Partitioning" by dragging the arrow of the "Column Filter" node.

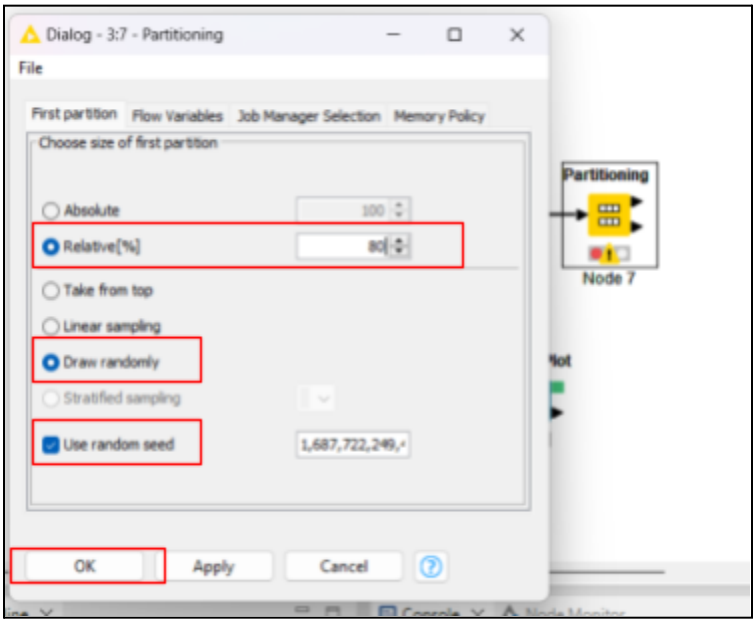


Figure 1.2.3.2

2. Double click on the “Partitioning” node and then set the size of the portion as shown in figure above and then click the “OK” button. This is to split the data into training and testing sets. Then, right click the node and select “Execute” to execute the node.

Figure 1.2.3.3

3. Search for the “Linear Regression Learner” node in the Node Repository and drag the node to the work area and connect the training set node to the “Linear Regression Learner” by dragging the arrow of the training set node. Repeat this step for the test set too.

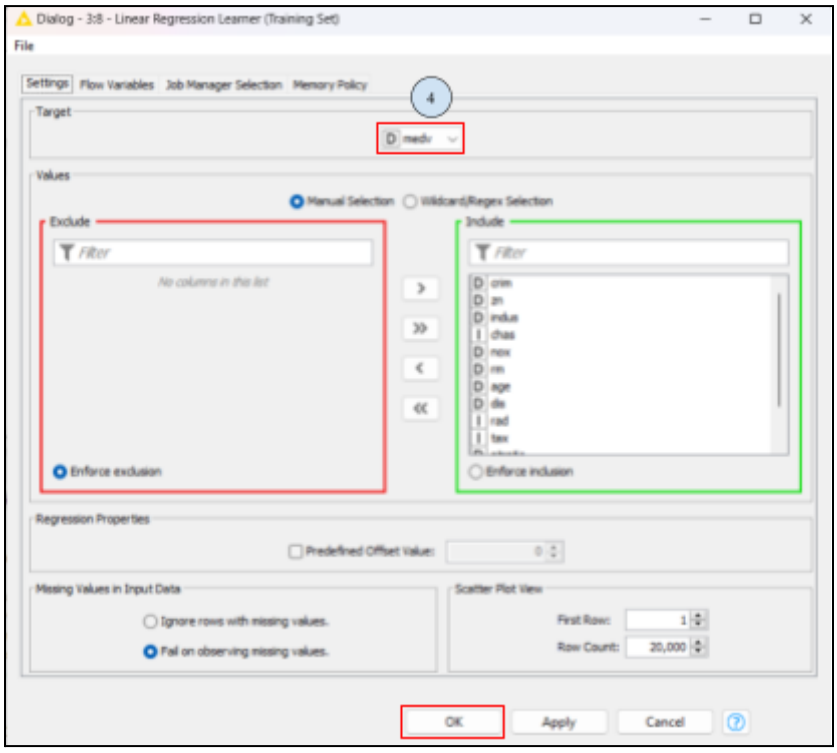


Figure 1.2.3.4

4. Double click on the “Linear Regression Learner” node to configure the node and a new window will open right after that. Set the target to the “medv” attribute because “medv” is our target attribute which indicates the price of the house. After that, click on the “OK” button. Then, execute the node.

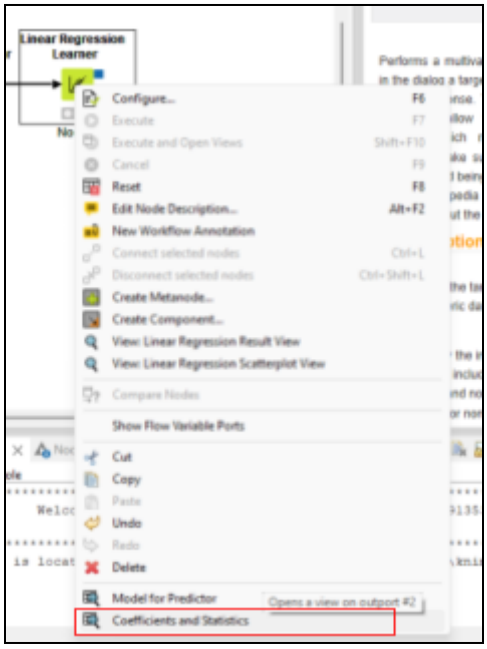


Figure 1.2.3.5

5. Right click on the “Linear Regression Learner” and then select “Coefficient and Statistics”.

Coefficients and Statistics - 3:8 - Linear Regression Learner (Training Set)

File Edit Hilite Navigation View

Table "Coefficients and Statistics" - Rows: 14 Spec - Columns: 5 Properties Flow Variables

Row ID	S Variable	D Coeff.	D Std. Err.	D t-value	D P>|t|
Row1	crim	-0.097	0.038	-2.56	0.011
Row2	zn	0.04	0.015	2.584	0.01
Row3	indus	0.015	0.072	0.205	0.838
Row4	chas	3.293	1.048	3.141	0.002
Row5	nox	-15.95	4.474	-3.565	0
Row6	rm	3.527	0.469	7.52	0
Row7	age	0.001	0.015	0.064	0.949
Row8	dis	-1.393	0.221	-6.301	0
Row9	rad	0.291	0.077	3.796	0
Row10	tax	-0.013	0.004	-2.901	0.004
Row11	ptratio	-0.95	0.153	-6.193	0
Row12	black	0.008	0.003	2.521	0.012
Row13	lstat	-0.531	0.058	-9.221	0
Row14	Intercept	38.086	5.859	6.501	0

Coefficients and Statistics - 3:9 - Linear Regression Learner (Test Set)

File Edit Hilite Navigation View

Table "Coefficients and Statistics" - Rows: 14 Spec - Columns: 5 Properties Flow Variables

Row ID	S Variable	D Coeff.	D Std. Err.	D t-value	D P>|t|
Row1	crim	-0.173	0.063	-2.725	0.008
Row2	zn	0.06	0.03	1.989	0.05
Row3	indus	0.052	0.111	0.469	0.64
Row4	chas	1.952	1.433	1.363	0.176
Row5	nox	-18.506	7.07	-2.617	0.01
Row6	rm	6.073	0.977	6.216	0
Row7	age	-0.027	0.027	-1.026	0.308
Row8	dis	-1.982	0.479	-4.135	0
Row9	rad	0.362	0.124	2.933	0.004
Row10	tax	-0.013	0.007	-1.937	0.056
Row11	ptratio	-1	0.243	-4.114	0
Row12	black	0.02	0.005	3.807	0
Row13	lstat	-0.419	0.108	-3.884	0
Row14	Intercept	20.576	10.464	1.966	0.052

Figure 1.2.3.6

6. As you can see, that is the coefficients and statistics table for our both training and test sets.

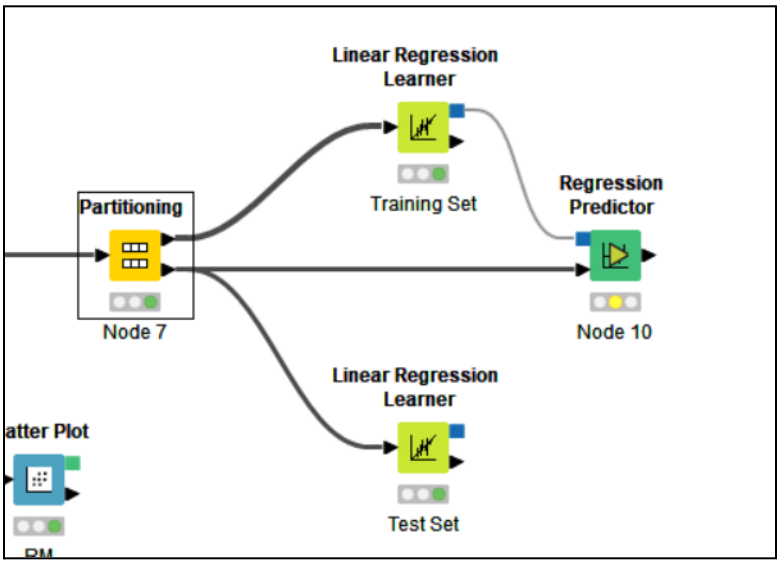
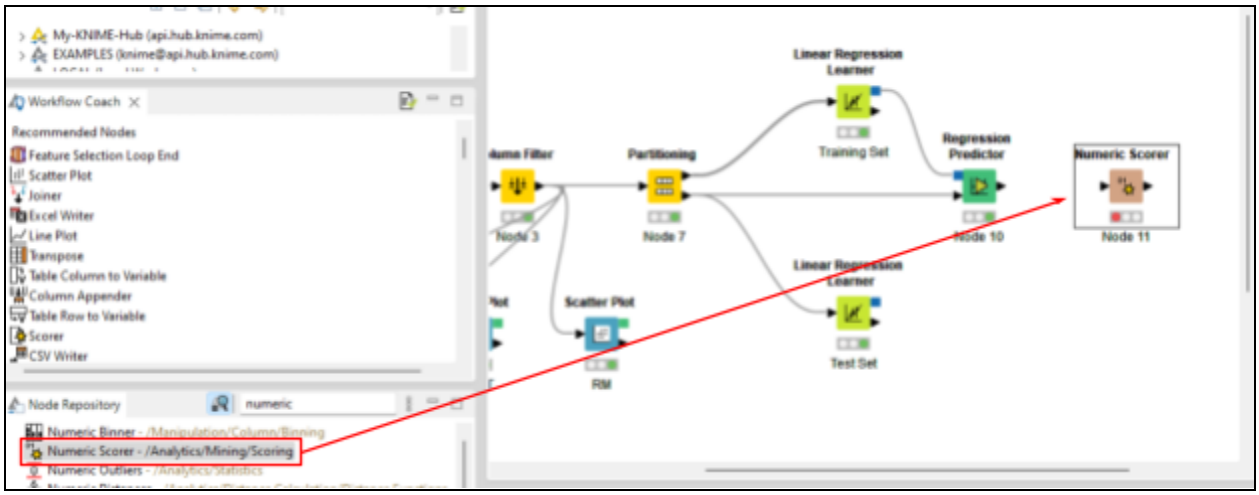


Figure 1.2.3.7

7. Search for the "Regression Predictor" node in the Node Repository and drag the node to the work area and connect the model for the predictor from training set "Linear Regression Learner" node by dragging the regression model of "Regression Predictor" node. Also drag the arrow of the test set of the "Partitioning" node to "Regression Predictor" node as shown in figure above. Then, execute the "Regression Predictor" node.



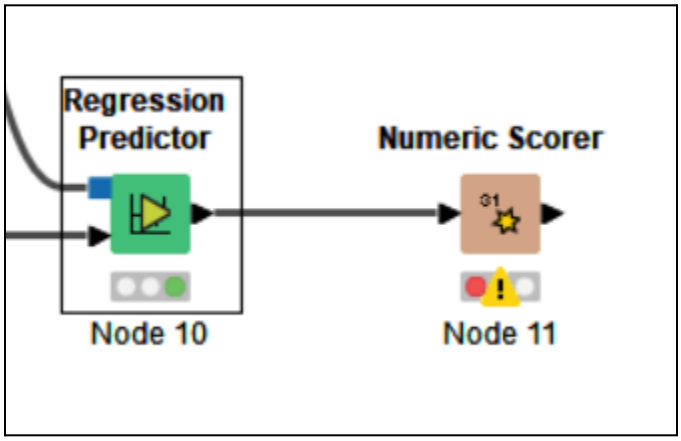


Figure 1.2.3.8

8. Search for the "Numeric Scorer" node in the Node Repository and drag the node to the work area and connect the "Regression Predictor" node by dragging the arrow to the "Numeric Scorer" node.

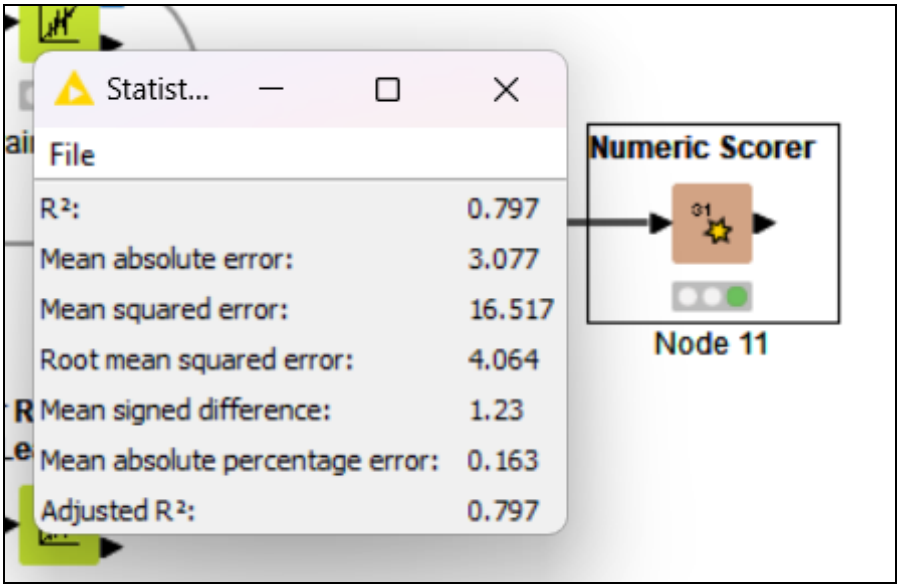


Figure 1.2.3.9

9. Right click on the "Numeric Scorer" and click on "Execute" to execute the node. Figure above shows the result of the calculations made from the prediction.

Result Discussion

Dataset : Boston.csv

The main thing we want to figure out is how much these houses are worth, which is represented by the median value of owner-occupied homes. To start off, we need to get the data ready for analysis. We can use a tool like KNIME to handle any missing values, scale or normalize the features, and take care of any categorical variables. This helps us ensure that our results are accurate and reliable.

Once the data is all prepped, we split it into two sets: training and test. The training set is where we teach our models how to estimate house prices based on the given features. Finally, the test set allows us to see how well our chosen model performs on unseen data.

With Linear Regression, we train our models on the training set using algorithms. The goal is to find the best combination of coefficients that minimize the difference between our predicted house prices and the actual prices in the training data.

After training, we need to evaluate our models to see how well they perform. We use metrics like Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared for this. MSE and RMSE measure the average difference between our predicted prices and the actual prices. Lower values of MSE and RMSE mean that our models are doing a better job of estimating house prices. On the other hand, R-squared tells us how much of the variation in house prices our models can explain. Higher R-squared values indicate a better fit to the data.

It's also important to check if our models meet certain assumptions of Linear Regression. For example, we assume that there is a linear relationship between the features and the target variable, and that the residuals (the differences between predicted and actual prices) are normally distributed and independent.

In a nutshell, by using Linear Regression on the Boston Housing Dataset, we can estimate house prices based on the given features. The evaluation of our models using metrics like MSE, RMSE, and R-squared helps us choose the best one. Ultimately, this analysis gives us insights into the factors that influence house prices in the Boston area and helps us make more informed decisions in the real estate market.

1.2.4 Clustering Rule - K-means

- **Dataset 3: Boston.csv**

Figure 1.2.4.1

1. Search “k-Means” in the Node Repository to do the clustering algorithm K-Means.
2. Drag the “k-Means” into the canvas and connect the node.

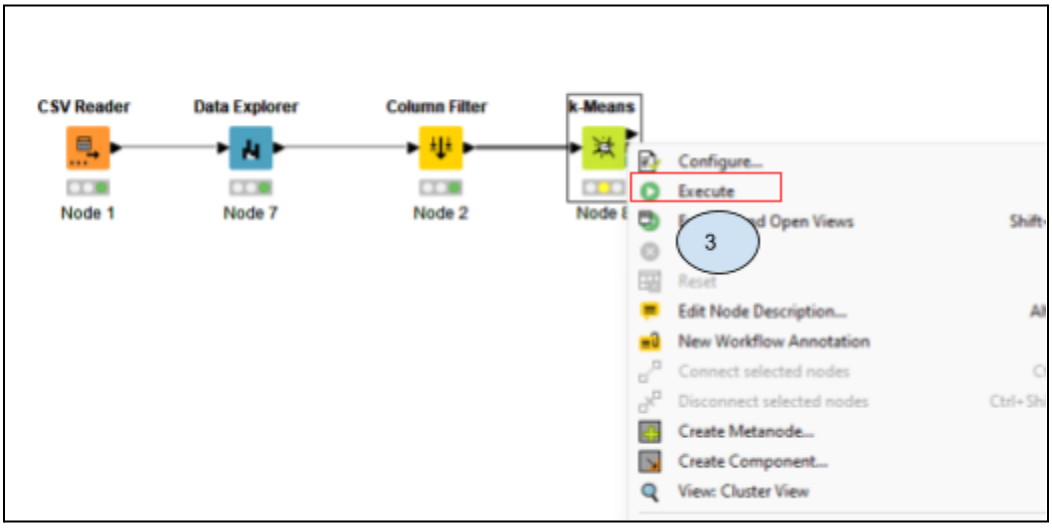


Figure 1.2.4.2

3. Execute the k-Means node.

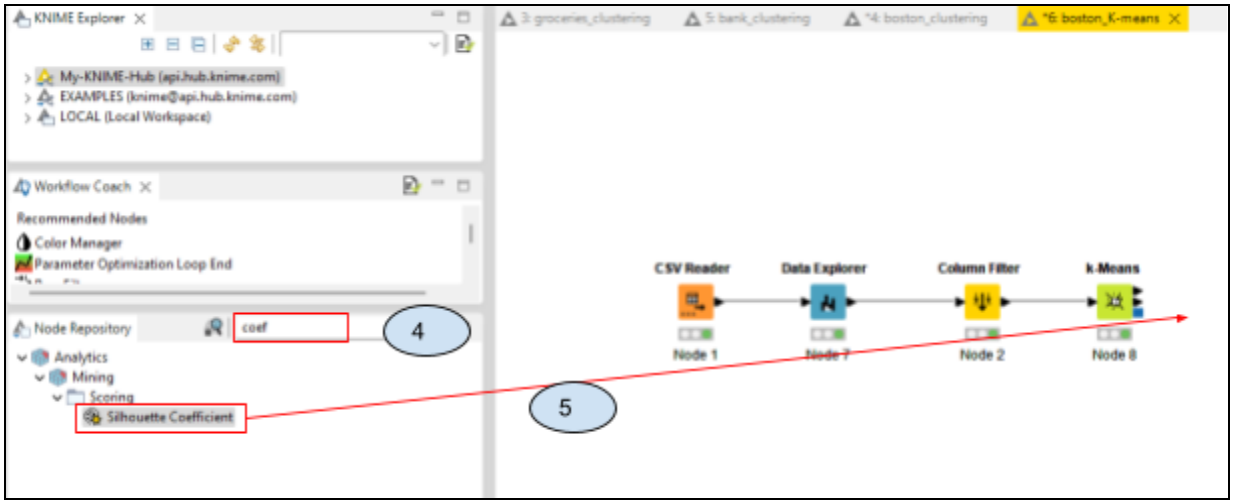


Figure 1.2.4.3

4. Search "Silhouette Coefficient" in the Node Repository to evaluate the clustering calculation coefficient.
5. Drag the "Silhouette Coefficient" into the canvas and connect the node.

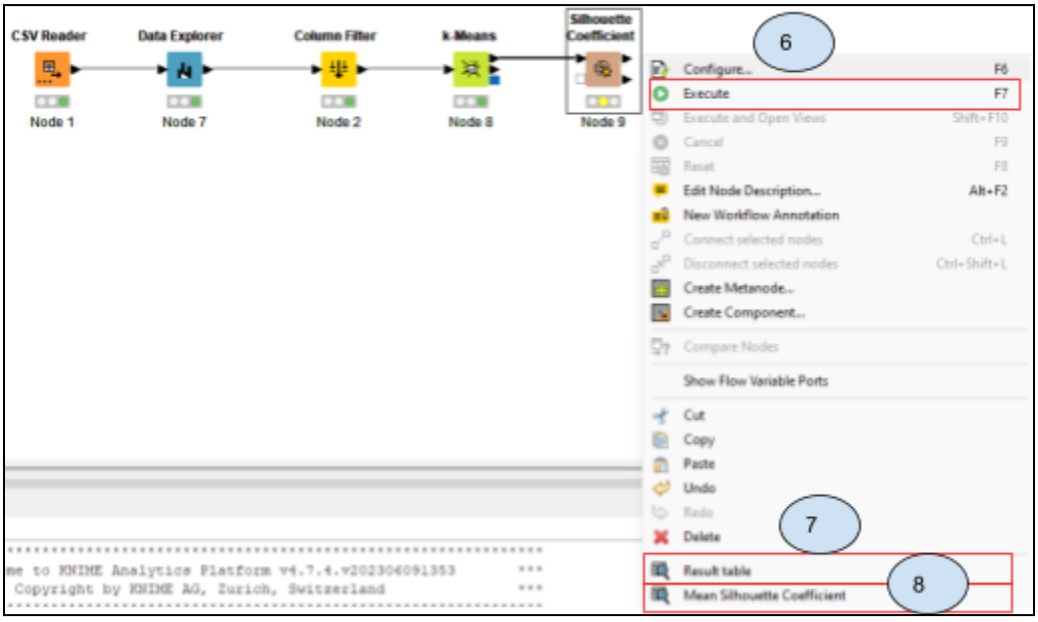
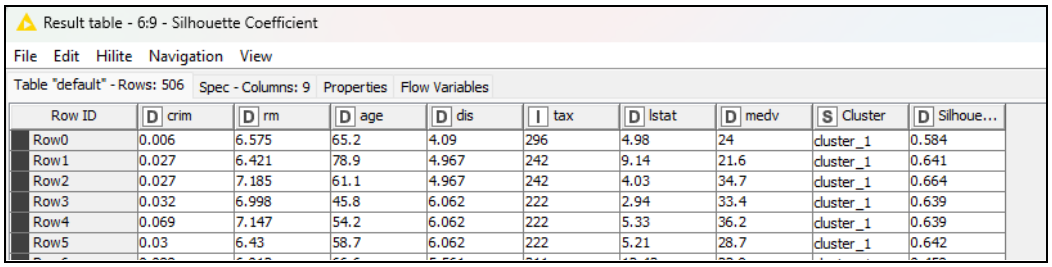


Figure 1.2.4.4

6. Execute the node.
7. Go the result table section to see the coefficient for each row.



Result table - 6:9 - Silhouette Coefficient

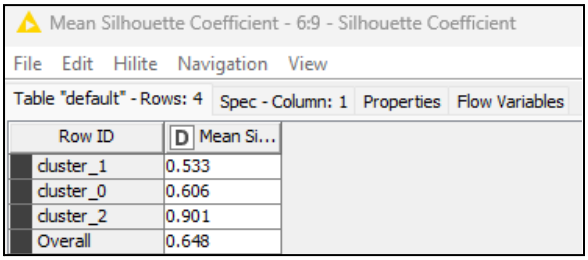
File Edit Hilite Navigation View

Table "default" - Rows: 506 Spec - Columns: 9 Properties Flow Variables

Row ID	crim	rm	age	dis	tax	lstat	medv	Cluster	Silhoue...
Row0	0.006	6.575	65.2	4.09	296	4.98	24	cluster_1	0.584
Row1	0.027	6.421	78.9	4.967	242	9.14	21.6	cluster_1	0.641
Row2	0.027	7.185	61.1	4.967	242	4.03	34.7	cluster_1	0.664
Row3	0.032	6.998	45.8	6.062	222	2.94	33.4	cluster_1	0.639
Row4	0.069	7.147	54.2	6.062	222	5.33	36.2	cluster_1	0.639
Row5	0.03	6.43	58.7	6.062	222	5.21	28.7	cluster_1	0.642

Figure 1.2.4.5

8. Go the Mean Silhouette Coefficient section to see the mean coefficient for each clusters and the overall. The closer the overall value to 1, indicate better-defined and well-separated clusters.



Mean Silhouette Coefficient - 6:9 - Silhouette Coefficient

File Edit Hilite Navigation View

Table "default" - Rows: 4 Spec - Column: 1 Properties Flow Variables

Row ID	Mean Si...
cluster_1	0.533
cluster_0	0.606
cluster_2	0.901
Overall	0.648

Figure 1.2.4.6

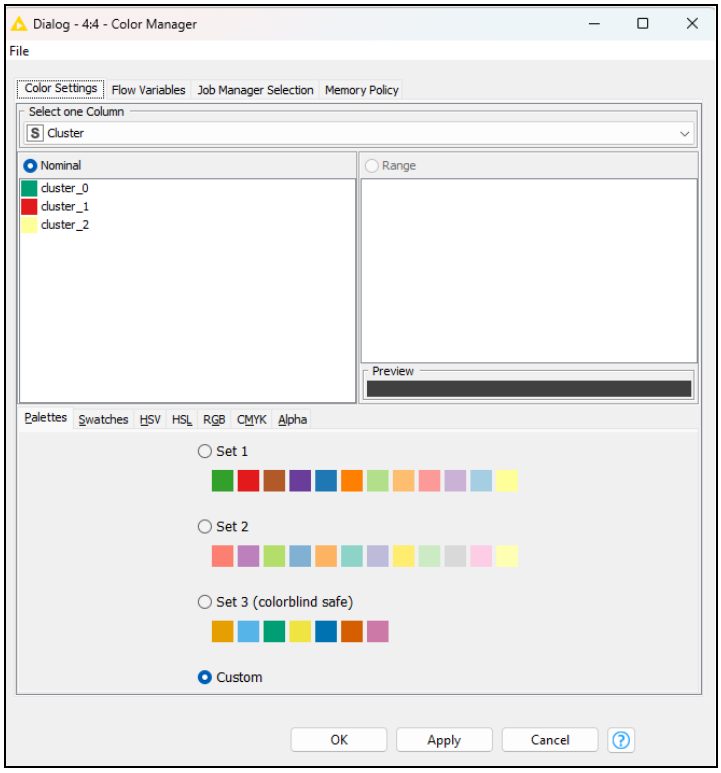


Figure 1.2.4.7

9. Use the Color Manager node and choose a suitable color for each of the clusters, apply and execute the node.

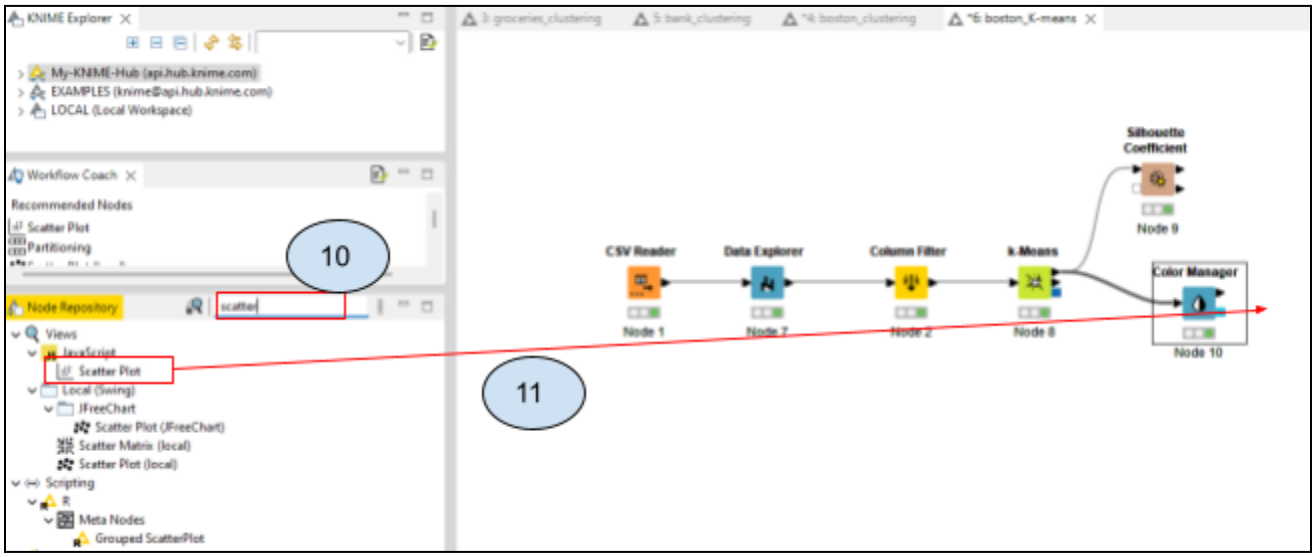


Figure 1.2.4.8

10. Search “Scatter Plot” in the Node Repository for the graph.
11. Drag the “Scatter Plot” into the canvas and connect the node.
12. Configure the nodes x-axis with RM (Average number of rooms per dwelling) and y-axis with CRIM (Per capita crime rate by town) for the clusters show a linear patterns, then click apply and OK.

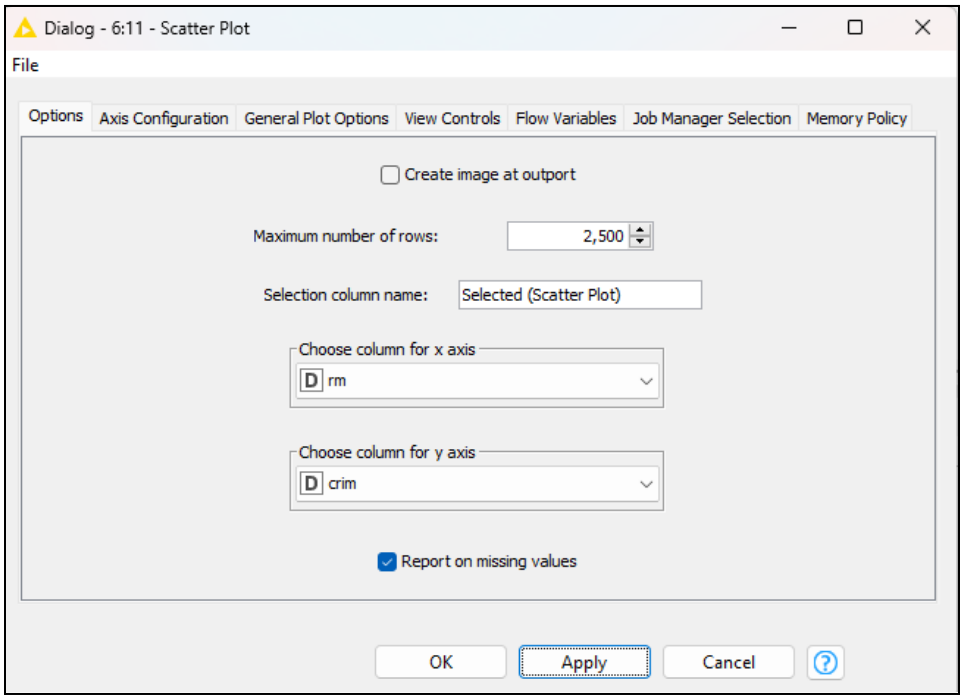


Figure 1.2.4.9

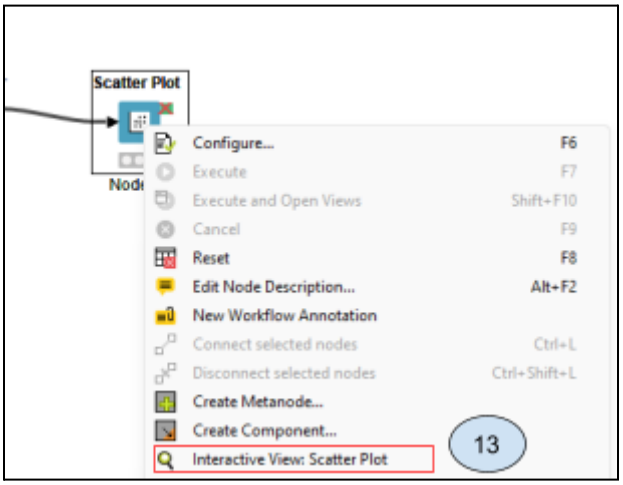


Figure 1.2.4.10

13. Right click the Scatter Plot node and click “Interactive View: Scatter Plot” to see the scatter plot graph.

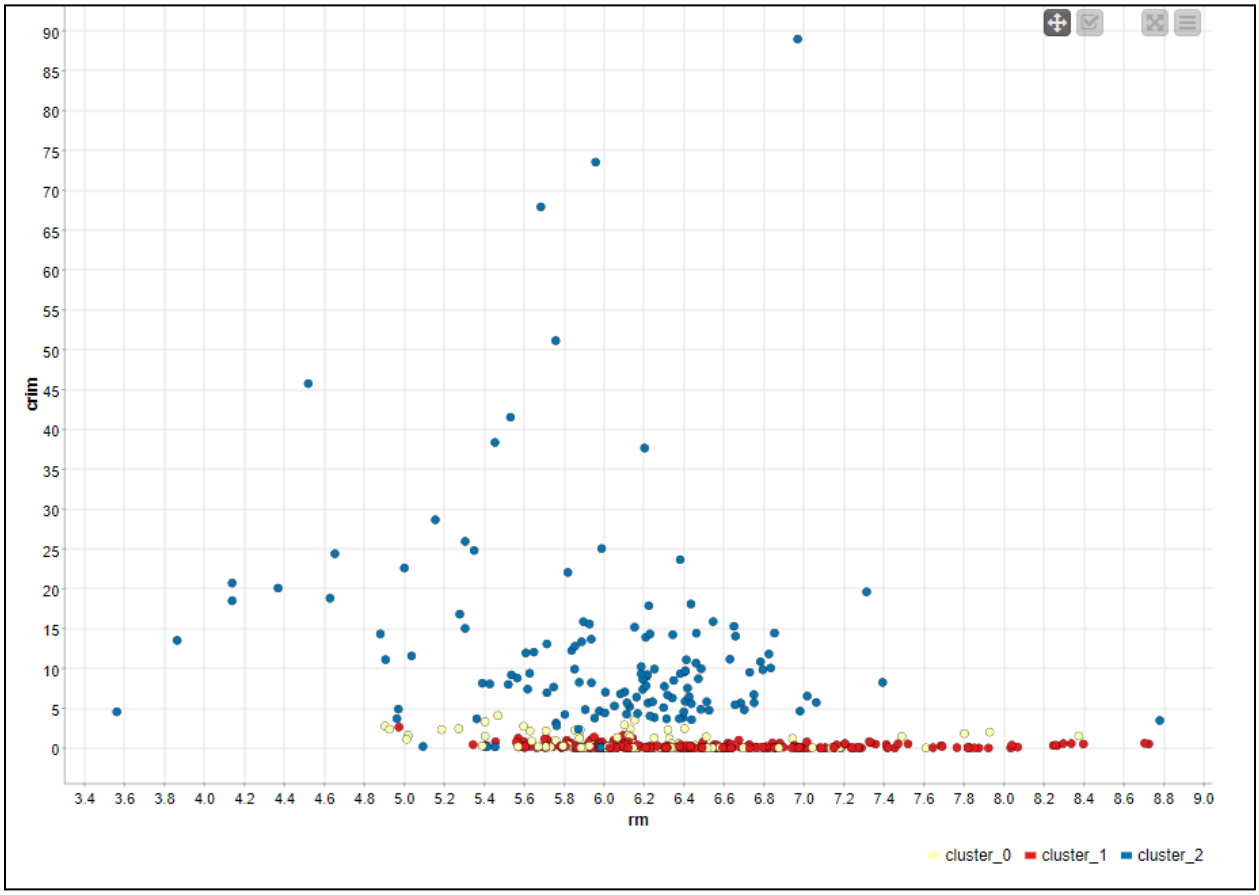


Figure 1.2.4.11

14. The scatter plot of graph CRIM vs RM will be display, the clusters can be seen almost separated from each other. As stated below in the legend of the scatter plot K-means, the yellow is cluster 0, red is cluster 1 and blue is cluster 2.

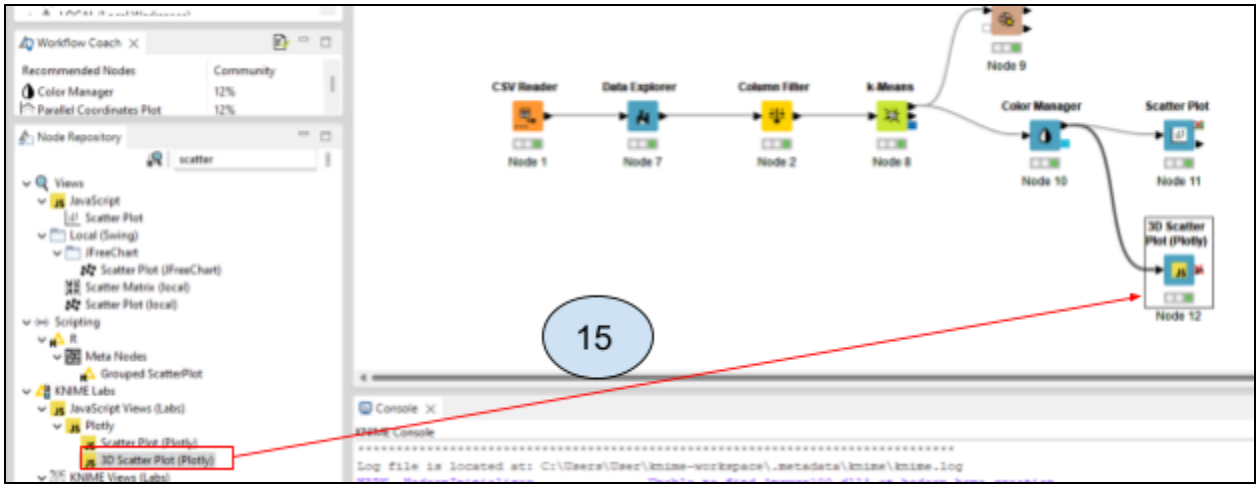


Figure 1.2.4.12

15. For an option of 3D scatter plot, use "3D Scatter Plot (Plotly)" node hence, add to the pallete

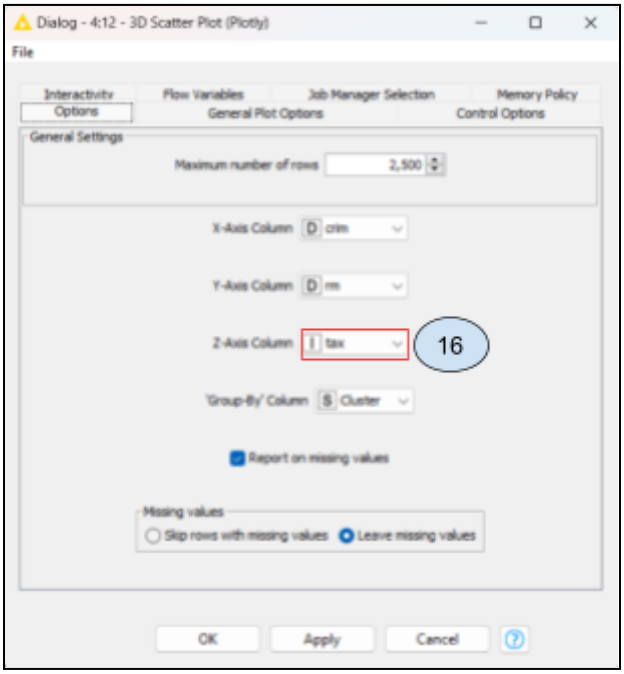


Figure 1.2.4.13

16. Configure the "3D Scatter Plot (Plotly)" node as above. Note that we add the tax variable to the existing 2D scatter plot.

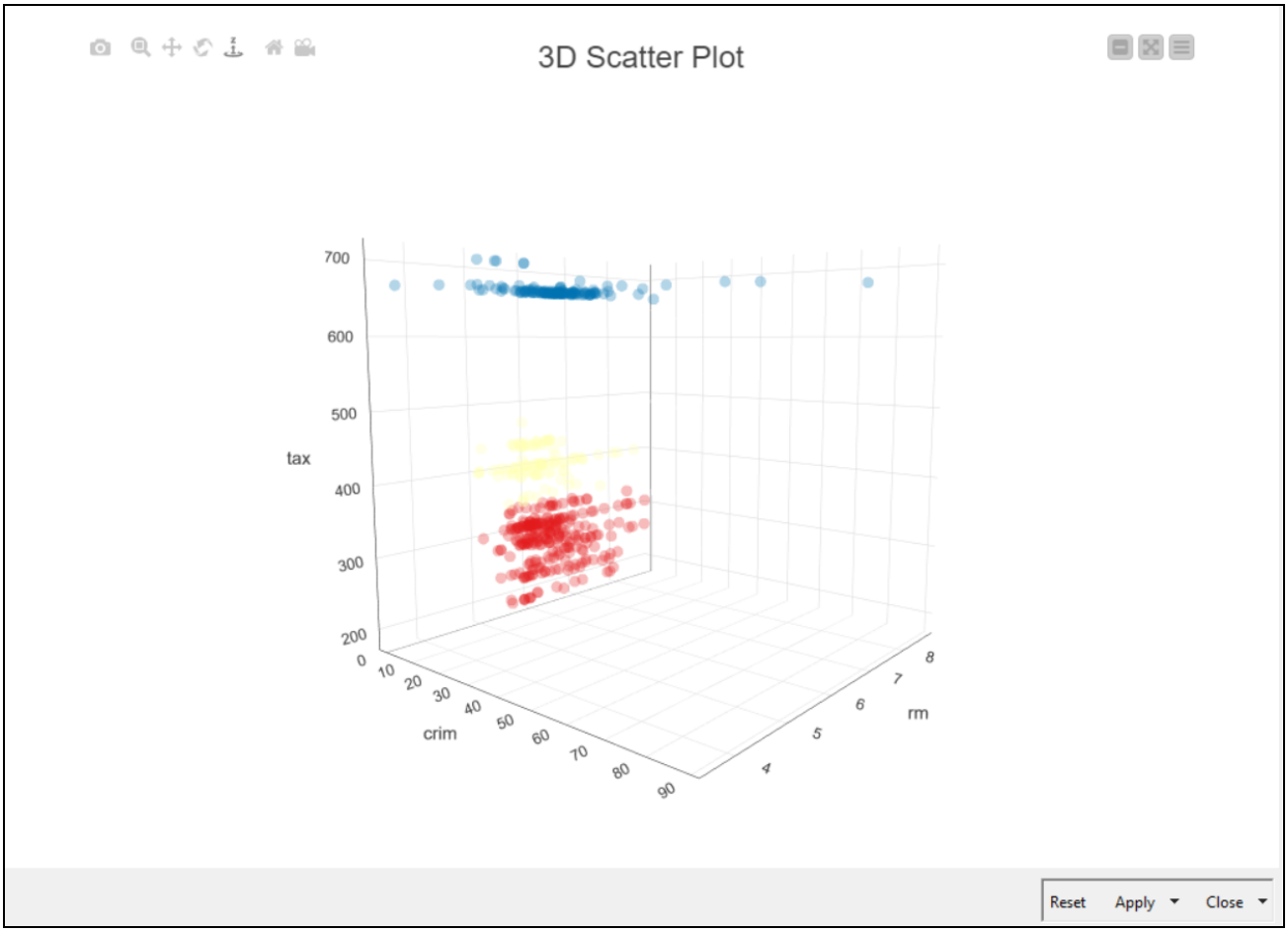


Figure 1.2.4.14

17. Execute the node, hence the 3D scatter plot of graph CRIM vs RM vs Tax will be display, the yellow is cluster 0, red is cluster 1 and blue is cluster 2. Notice how separated and distinct the clusters are compared to the 2D scatter plot.

Result Discussion

Dataset : Boston.csv

As we can conclude on the relationship between Room Count (RM) and Crime Rate (CRIM), by clustering the data based on RM and CRIM, we can examine the relationship between the number of rooms in a dwelling and the corresponding crime rates. We may observe that areas with a higher number of rooms tend to have lower crime rates, indicating a potential negative correlation between room count and crime.

For the Relationship between Room Count (RM) and Crime Rate (CRIM), and Property Tax in the 3D scatter plot, we can observe the relationships between room count, crime rate, and property tax. For instance, we might find that higher room counts tend to be associated with lower crime rates and moderate property tax rates, indicating larger and potentially more expensive homes in safer neighborhoods.

Clustering can help identify distinct segments or types of neighborhoods based on the combination of Room Count and Crime Rate. For example, we may find clusters representing high-room, low-crime neighborhoods or low-room, high-crime neighborhoods. This segmentation can provide insights into the different characteristics of neighborhoods within the dataset.

The purpose on using Clustering is to help identify outliers in the scatter plot. Outliers are data points that deviate significantly from the overall patterns. By examining the clusters, we can identify any observations that fall outside the general clustering structure, which may represent unique or exceptional cases in terms of Room Count and Crime Rate.

In assessing the degree of separation between clusters in the scatter plot can help evaluate the effectiveness of the clustering algorithm. If the clusters are well-separated and distinct, it suggests that the algorithm successfully grouped similar data points together based on their Room Count and Crime Rate. Based on the scatter plot we get above, we can see there is some distinct and separated that suggested the clustering is good.

1.3 Conclusion

In this data mining project, we leveraged the versatile tool KNIME to perform comprehensive data preprocessing and a range of data tasks, including association, classification, clustering, and regression. The project involved a systematic approach to extract valuable insights and patterns from the dataset, ultimately contributing to better decision-making and predictive modeling.

Data preprocessing played a crucial role in ensuring the data quality and reliability. We applied various techniques such as data cleaning, normalization, and handling missing values to ensure the dataset was accurate, complete, and suitable for analysis. By addressing data inconsistencies and preparing the data appropriately, we laid a solid foundation for subsequent tasks.

Through association analysis, we uncovered hidden relationships and dependencies between variables. Using algorithms such as Apriori or FP-Growth, we identified frequent itemsets and association rules, revealing valuable patterns that could contribute to strategic decision-making, marketing campaigns, or product recommendations.

Classification was a key task in this project, where we aimed to build models capable of categorizing new instances accurately. By employing decision trees, random forests, or support vector machines, we developed robust classifiers that learned from the data's patterns and features. These models can now be used to predict the class or category of unseen data, enabling automated decision-making or aiding in risk assessment.

Clustering techniques allowed us to group similar instances together, uncovering inherent structures within the data. Through algorithms such as k-means or hierarchical clustering, we identified natural clusters, segments, or patterns, enabling targeted marketing strategies, customer segmentation, or anomaly detection.

Regression analysis enabled us to understand and model the relationships between variables, specifically when predicting continuous numerical outcomes. By employing regression algorithms such as linear regression or polynomial regression, we developed models that could estimate or forecast values based on input variables, contributing to better resource allocation, demand forecasting, or performance optimization.

Overall, this project demonstrated the power of KNIME as a comprehensive data mining tool. Through effective data preprocessing and the application of various data tasks such as association, classification, clustering, and regression, we gained valuable insights, predictive models, and patterns that can inform decision-making, improve operational efficiency, and drive business growth. The outcomes of this project provide a solid foundation for further analysis, research, and data-driven strategies across various industries and domains.