Cover sheet for submission of work for assessment

SWINBURNE UNIVERSITY OF TECHNOLOGY

UNIT DETAILS

Unit name	Data Science Principles			Class day/time	Tuesday
Unit code	COS10022	Assignment no.	2	Due date	10/11/2024
Name of lecturer/teacher	Mr. Minh Hoang				
Tutor/marker's name	Mr. Minh Hoang				

Office use only

Faculty or school date stamp

STUDENT(S)

	Family Name(s)	Given Name(s)	Student ID Number(s)
(1)	Huynh	Trung Chien	104848770

DECLARATION AND STATEMENT OF AUTHORSHIP

1. I/we have not impersonated or allowed myself/ourselves to be impersonated by any person for the purposes of this assessment.
2. This assessment is my/our original work and no part of it has been copied from any other source except where due acknowledgement is made.
3. No part of this assessment has been written for me/us by any other person except where such collaboration has been authorised by the lecturer/teacher concerned.
4. I/we have not previously submitted this work for this or any other course/unit.
5. I/we give permission for my/our assessment response to be reproduced, communicated, compared and archived for plagiarism detection, benchmarking or educational purposes.

I/we understand that:

6. Plagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a form of cheating and is a very serious academic offence that may lead to exclusion from the University. Plagiarised material can be drawn from, and presented in, written, graphic and visual form, including electronic data and oral presentations. Plagiarism occurs when the origin of the material used is not appropriately cited.

Student signature/s

I/we declare that I/we have read and understood the declaration and statement of authorship.

(1) HUYNH TRUNG CHIEN Digitally signed by HUYNH TRUNG CHIEN Date: 2024.11.10 10:51:23 +07'00'

Further information relating to the penalties for plagiarism, which range from a formal caution to expulsion from the University is contained on the Current Students website at **www.swin.edu.au/student/**

Copies of this form can be downloaded from the Student Forms web page at **www.swinburne.edu.au/studentforms/**

burne University of Technology Hawthorn Campus
of Computing Technologies

COS10022 Data Science Principles

Assignment 2 - *Semester 1, 2024*

Assessment Title: Data Cleaning and Analytics

Assessment Weighting: 30%

Due Date: Sunday, 12th May 2023 at 11.59 pm (AEDT)

Assessable Item:

- One (1) piece of a written report no more than 10 pages along with the signed Assignment Cover Sheet.
- The submitted report must be checked by Turnitin, and the similarity from **not the template part** should be less than 12%.
- A KNIME workflow in Assessment 2.1.

The submitted report should answer all questions listed in the assignment task section in sequence.

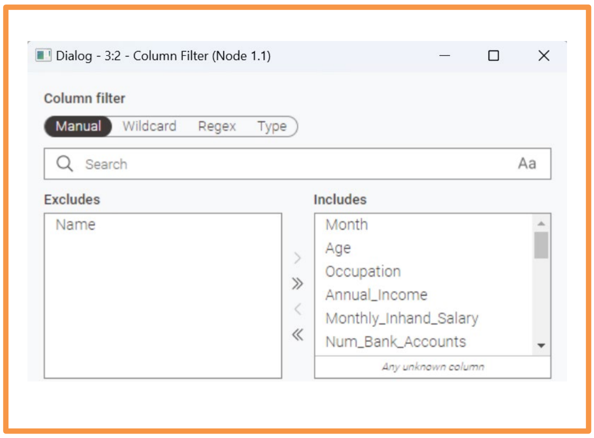
You must include a digitally signed Assignment Cover Sheet with your submission.

---

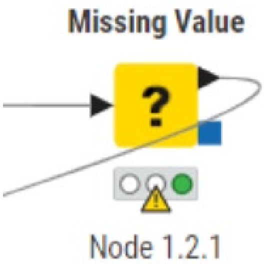
1. Follow the instructions to clean the data and answer questions. If any of the nodes you used in the workflow has a random seed, set **9214** to the seed to fix the random state. **[65 marks in total]**

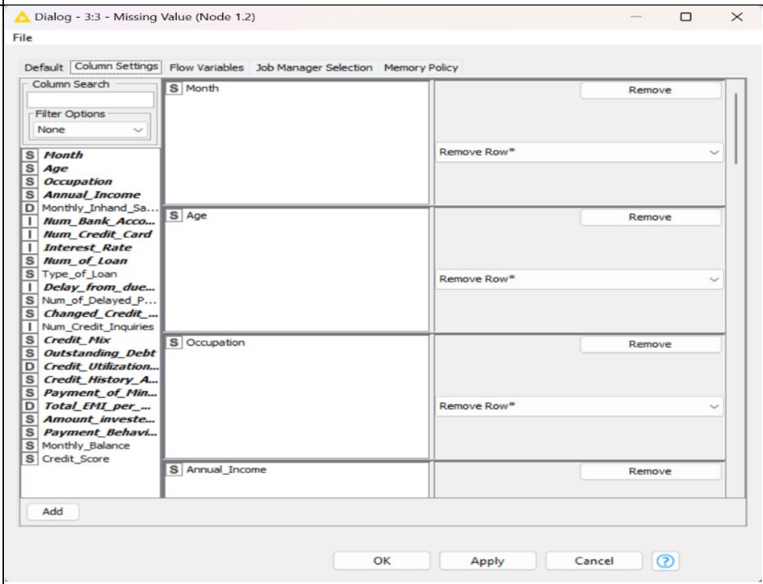
   1) Our goal is to predict the credit score from the given data. There is/are one (or multiple) attribute(s) which is/are significantly irrelevant to the goal. Pick the most irrelevant attribute and give a persuasive rationale for that. The excluded attribute(s) is __________, and the reason for removing it is _________. **[2.5 marks]**

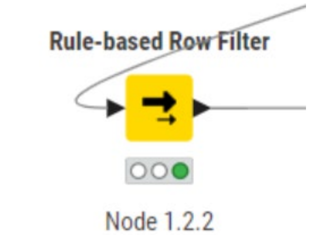
      - In my opinion, the most irrelevant and should be excluded attribute is “Name”. This is because names do not provide any useful information in terms of financial behaviours, creditworthiness or risk assessment. Therefore, this attribute does not contribute to the process of predicting the credit scores.

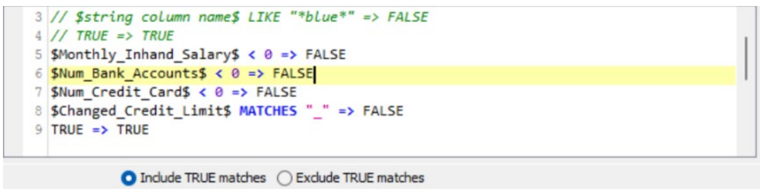


2) After removing the selected attribute(s), let’s start to remove tuples containing missing values. Remove tuples only if any of the attributes listed below have missing values: “Month,” “Age,” “Occupation,” “Annual_Income,” “Num_Bank_Accounts,” “Num_Credit_Card,” “Interest_Rate,” “Num_of_Loan,” “Delay_from_due_date,” “Changed_Credit_Limit,” “Credit_Mix,” “Outstanding_debt,” “Credit_Utilization_Ratio,” “Credit_History_Age,” “Payment_of_Min_Amount,” “Total_EMI_per_month,” “Amount_invested_monthly,” and “Payment_Behaviour.” Moreover, some tuples with infeasible values in the attributes, such as “Monthly_Inhand_Salary” < 0, “Num_Bank_Accounts” < 0, “Num_Credit_Card” < 0, and “Changed_Credit_Limit” contains “_”, should also be removed. List the node(s) (in sequence) and the corresponding command(s) used in this process. **[5 marks]**

<table>
<tr><th>Sequence</th><th>Node</th><th>Command</th></tr>
<tr><td>1</td><td>


</td><td>


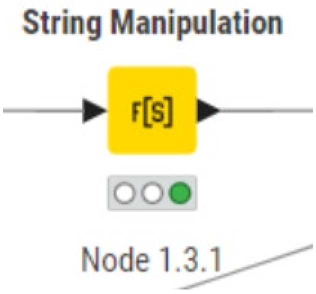
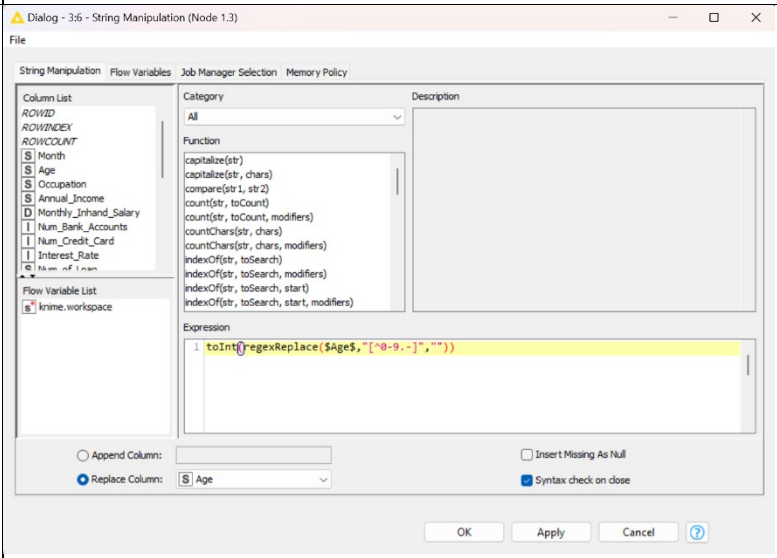
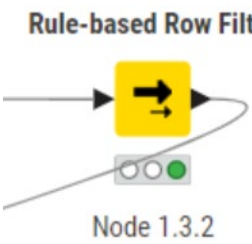
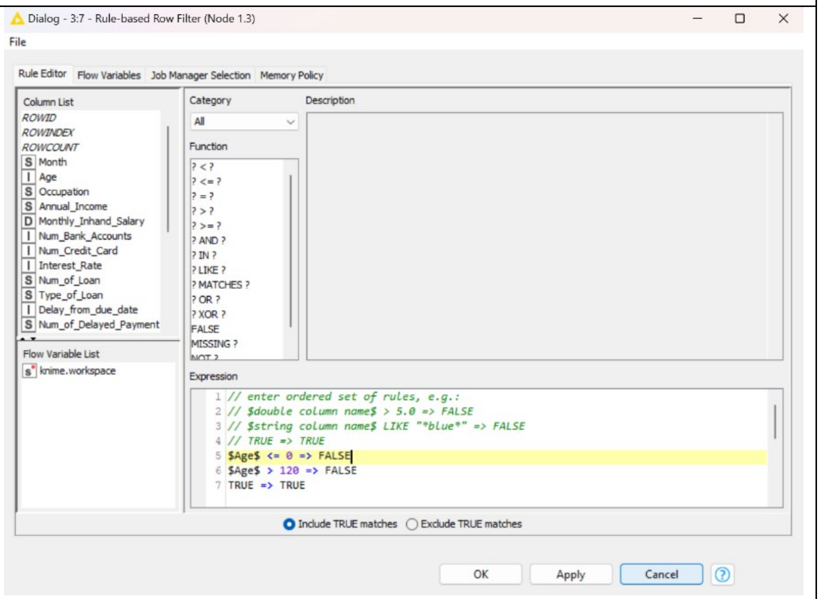
Command:
To remove tuples containing missing values, I use the Missing Value node. Then I use “Remove Row*” for specific attributes that contain missing values in the Column Settings.
</td></tr>
<tr><td>2</td><td>


</td><td>


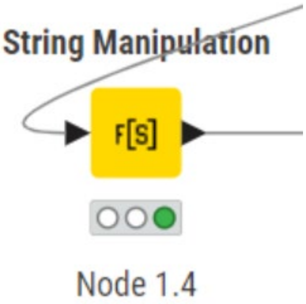
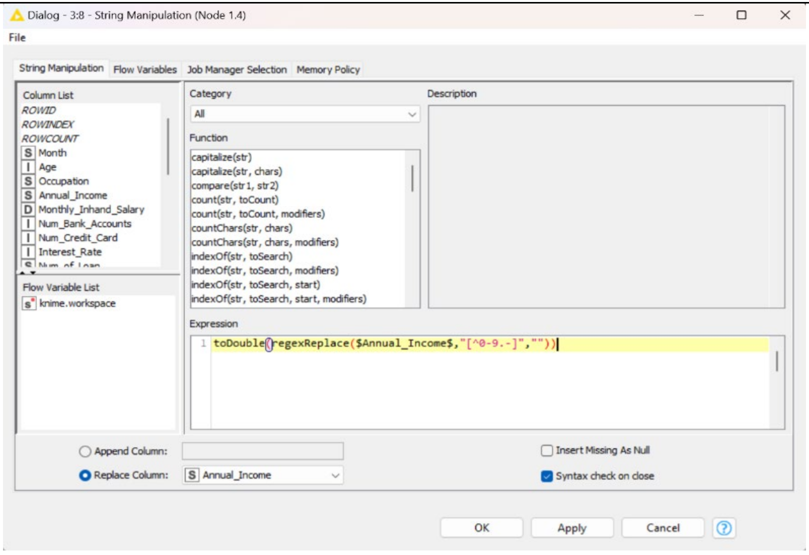
```
// $string column name$ LIKE "*blue*" => FALSE
// TRUE => TRUE
$Monthly_Inhand_Salary$ < 0 => FALSE
$Num_Bank_Accounts$ < 0 => FALSE
$Num_Credit_Card$ < 0 => FALSE
$Changed_Credit_Limit$ MATCHES "_" => FALSE
TRUE => TRUE
```


Command:
$Monthly_Inhand_Salary$ < 0 => FALSE
$Num_Bank_Accounts$ < 0 => FALSE
Num_Credit_Card < 0 => FALSE
$Changed_Credit_Limit$ MATCHES "_" => FALSE
TRUE => TRUE
</td></tr>
</table>

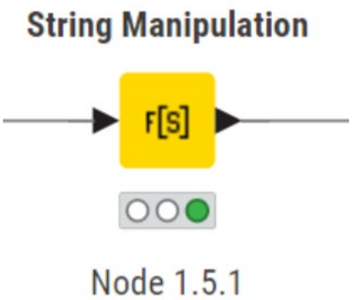
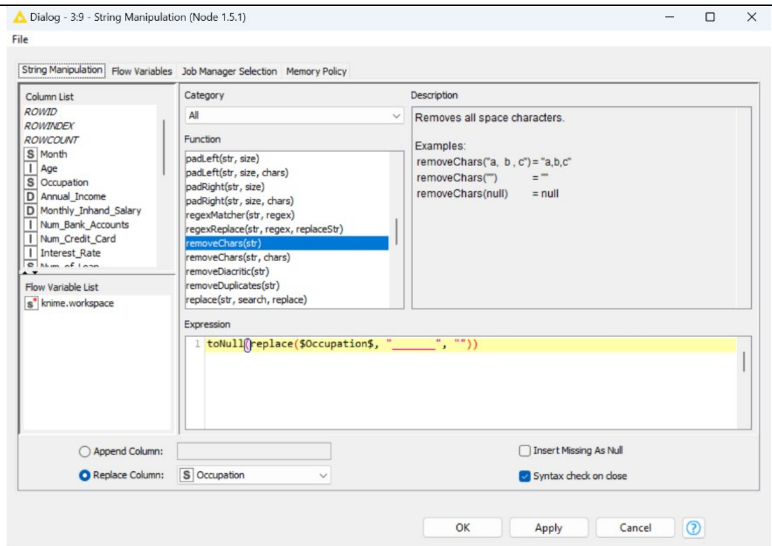
3) Check for the “Age” attribute to eliminate symbols that are not numbers to recover the data into the usual number format. Moreover, drop the tuples whose “Age” value is lower than or equal to 0 or greater than 120. List the node(s) (in sequence) and the corresponding command(s) used in this process. **[5 marks]**

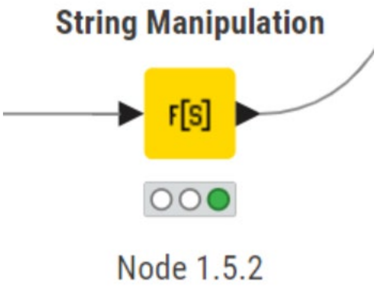
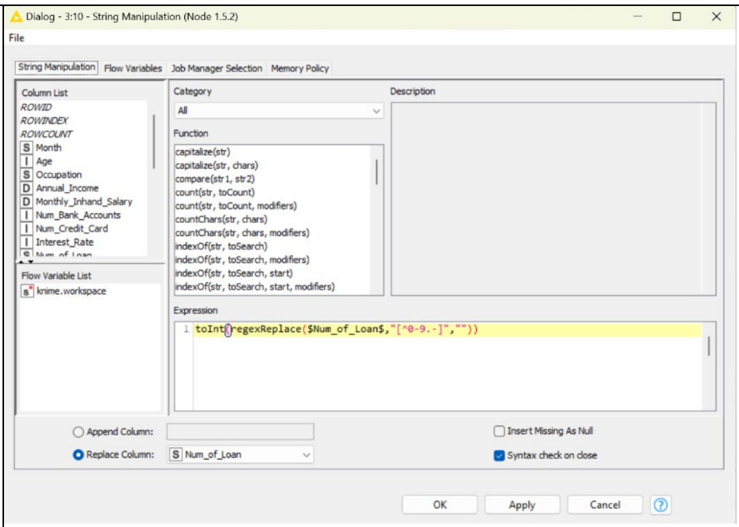
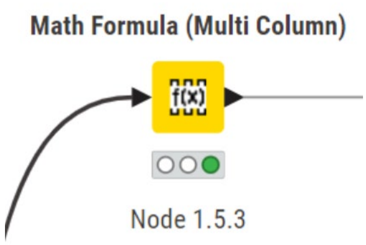
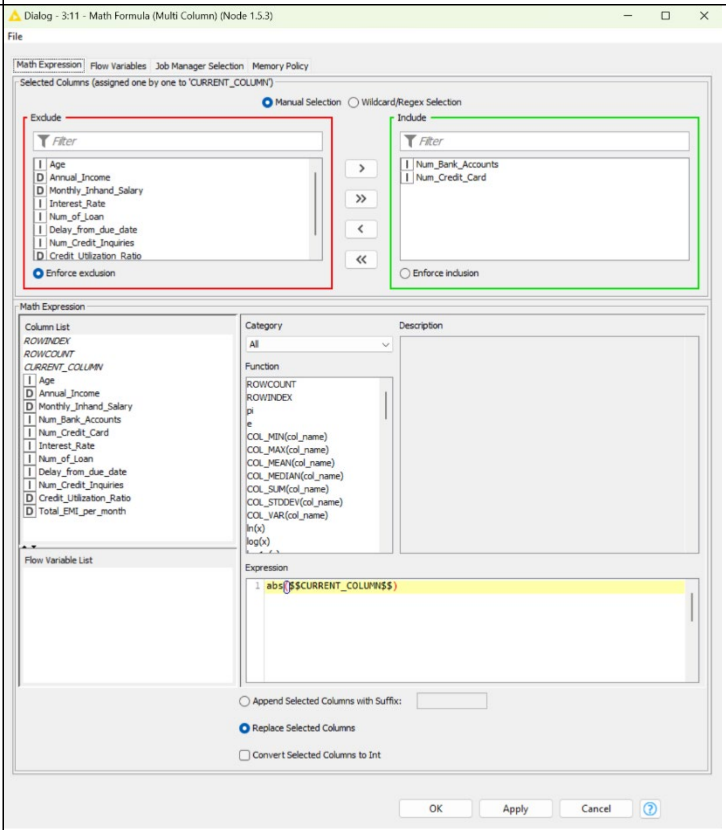
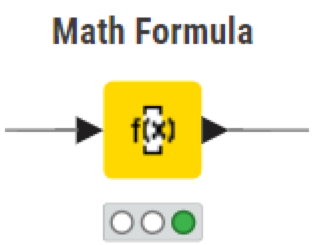
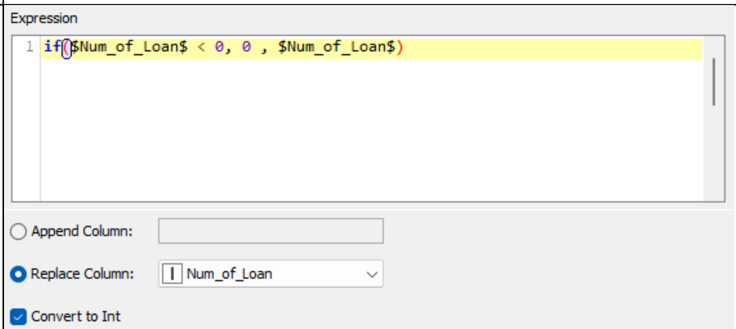
Sequence	Node	Command
1		 **Command:** toInt(regexReplace(Age,"[^0-9.-]",""))
2		 **Command:** Age <= 0 => FALSE Age > 120 => FALSE TRUE => TRUE

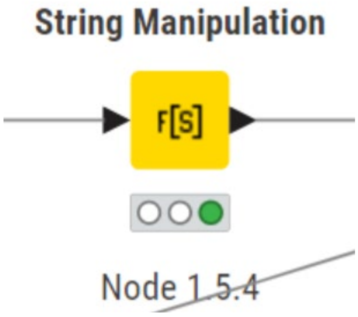
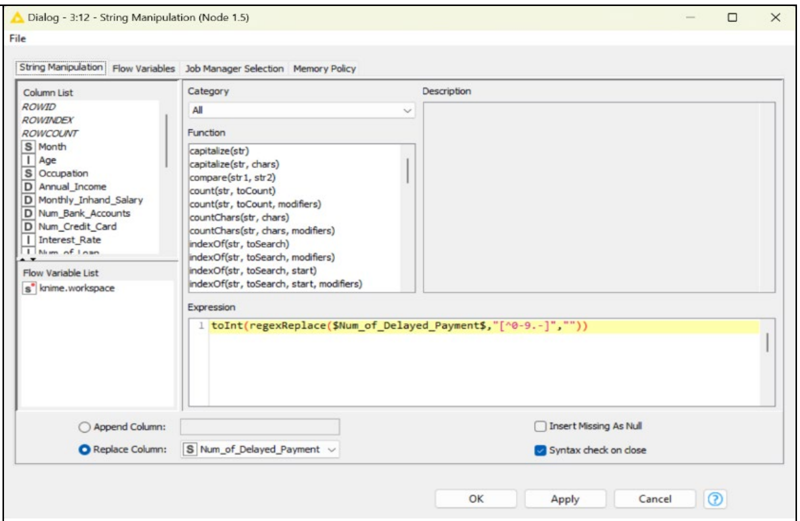
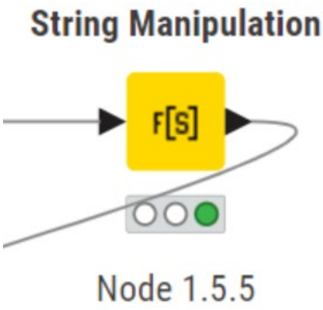
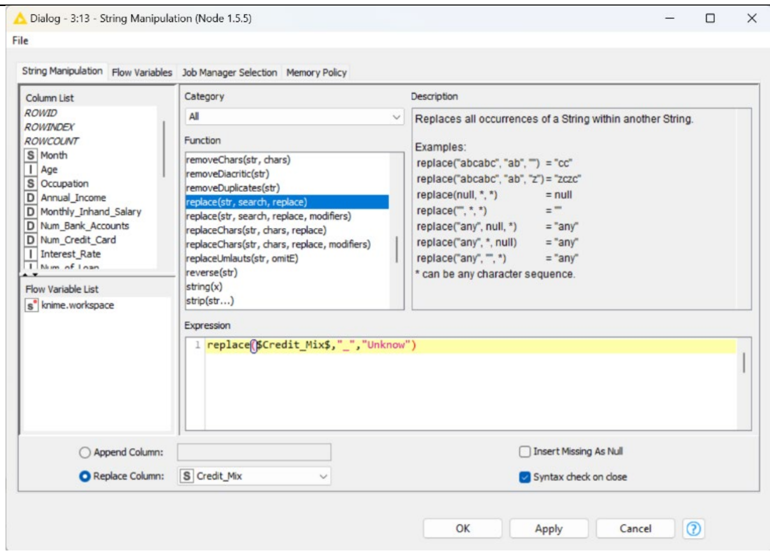
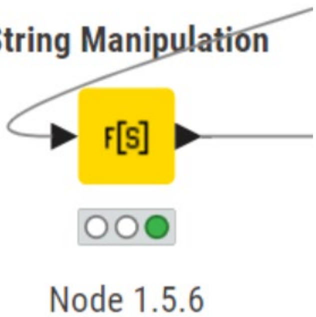
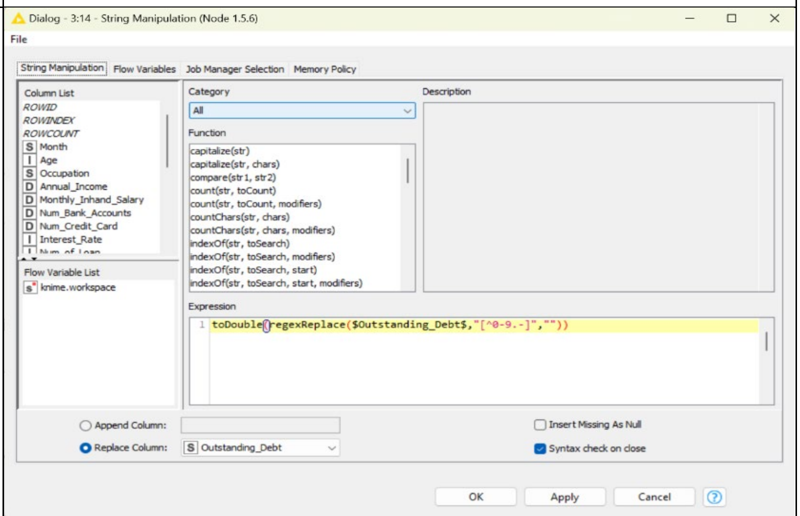
4) Remove the non-numerical symbol in the “Annual_Income” column and convert it to the double format. List the node(s) (in sequence) and the corresponding command(s) used in this process. **[5 marks]**

Sequence	Node	Command
1		 **Command:** toDouble(regexReplace($Annual_Income$,"[^0-9.-]",""))

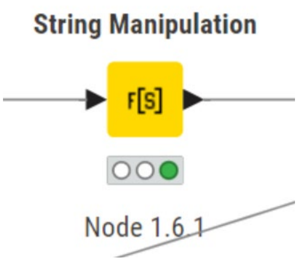
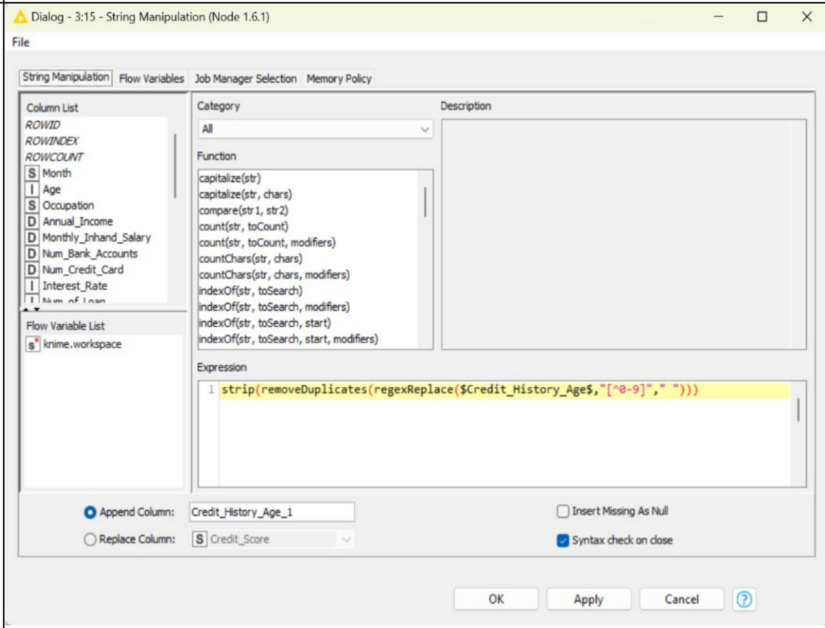
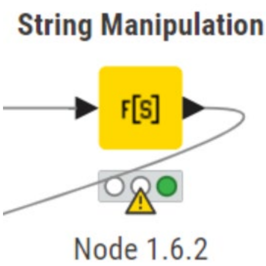
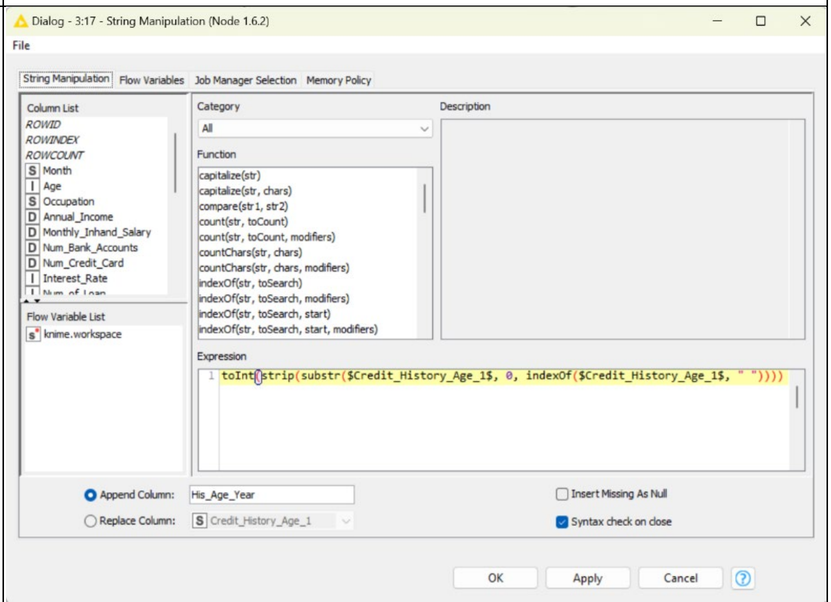
5) Convert the “_______” in the “Occupation” attribute to Null. Please note that Null is different from an empty string. Remove the non-numerical symbol in “Num_of_Loan” and convert it to integer data type. Take absolute values of attributes “Num_Bank_Accounts” and “Num_Credit_Card.” Set values to 0 for the “Num_of_Loan” attribute if the original values are negative. Remove the non-numerical symbol in “Num_of_Delayed_payment” and convert it into integer format. Set the “Credit_Mix” value to “Unknow” if the original value is “_”.Remove the non-numerical symbol in “Outstanding_Debt” and convert it into the double format. List the node(s) (in sequence) and the corresponding command(s) used in this process. **[10 marks]**

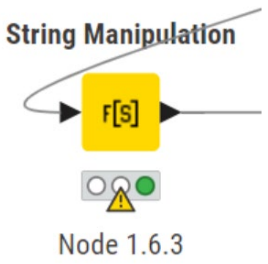
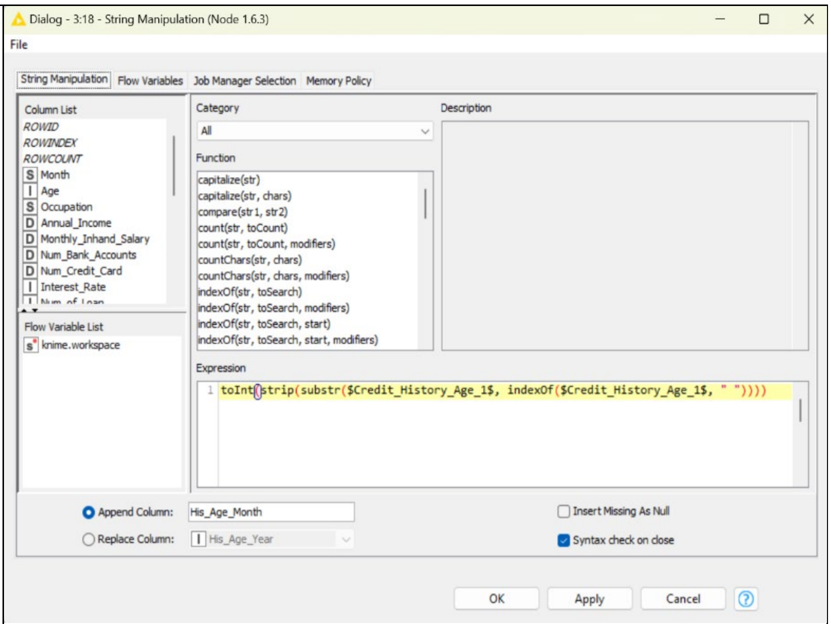
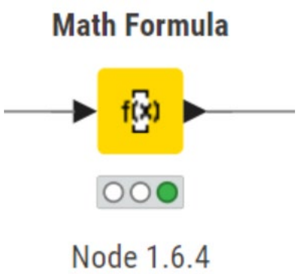
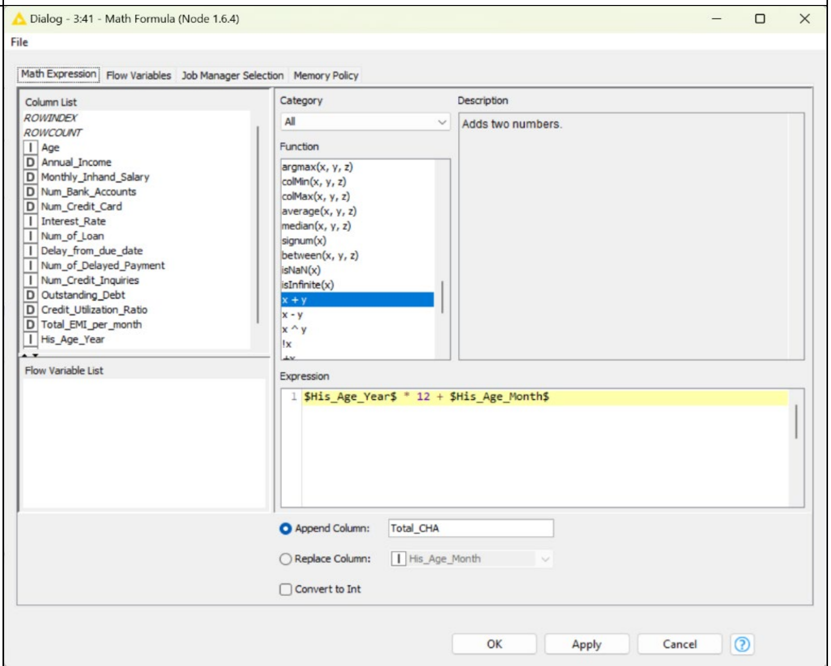
Sequence	Node	Command
1		 **Command:** toNull(replace($Occupation$, "_______", ""))

2		 **Command:** toInt(regexReplace(Num_of_Loan,"[^0-9.-]",""))
3		 **Command:** abs($$CURRENT_COLUMN$$)
4		 **Command:** if(Num_of_Loan < 0, 0 , Num_of_Loan)

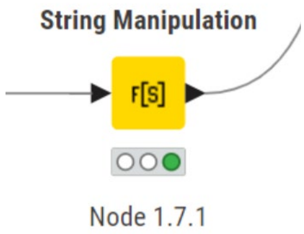
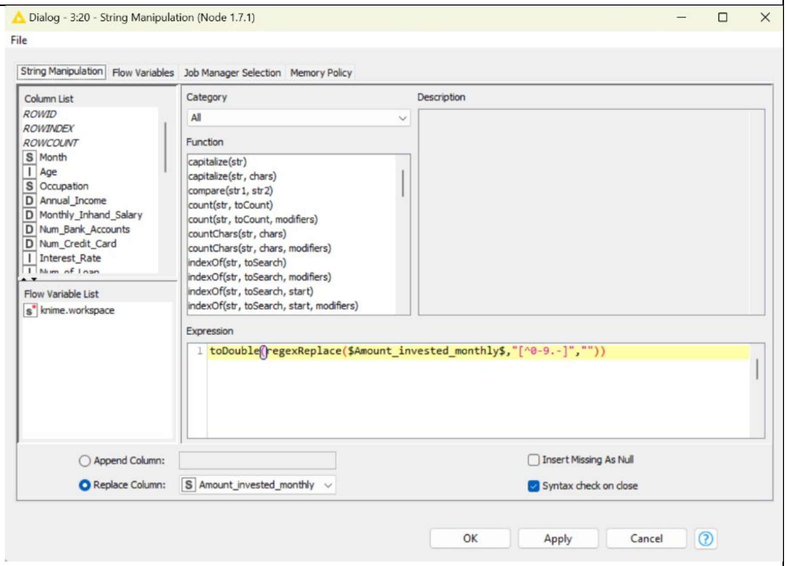
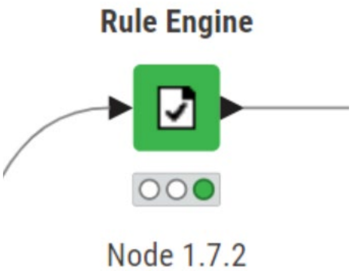
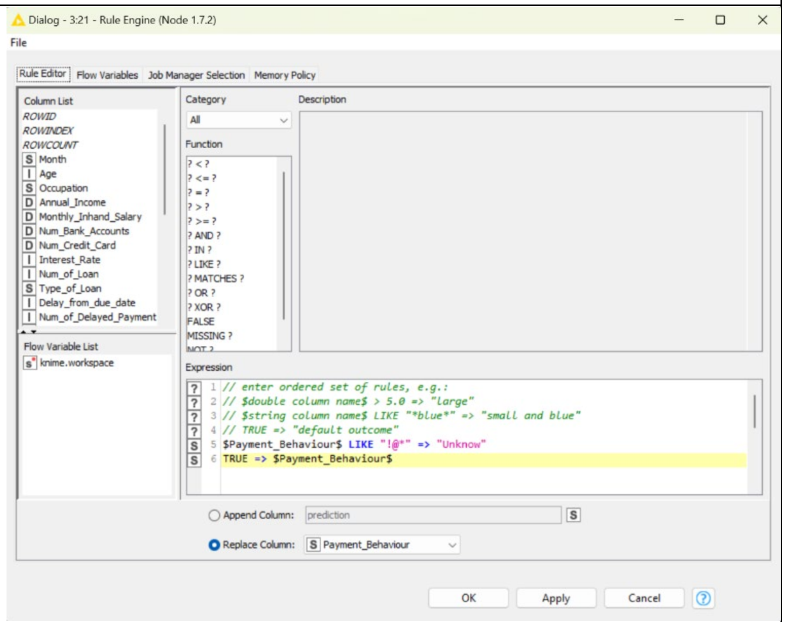
5		 **Command:** toInt(regexReplace($Num_of_Delayed_Payment$,"[^ 0-9.-]",""))
6		 **Command:** replace($Credit_Mix$,"_","Unknow")
7		 **Command:** toDouble(regexReplace($Outstanding_Debt$,"[^0-9.-]",""))

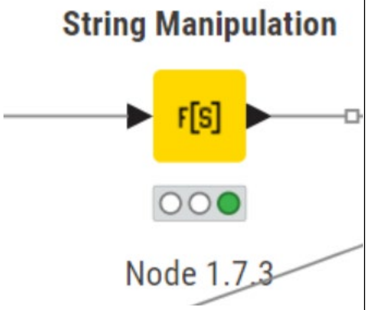
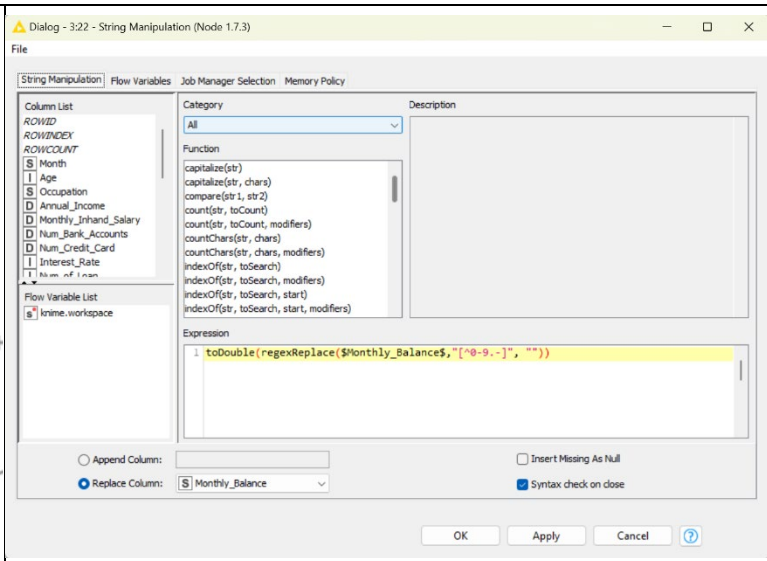
6) Convert the “Credit_History_Age” to the count of months and store it in the integer format. For example, if the original value from a tuple is “22 Years and 1 Months”, the value will be 265 after the conversion (22 * 12 + 1 = 265). Store the converted result in a new attribute called “Total_CHA.” List the node(s) (in sequence) and the corresponding command(s) used in this process. **[10 marks]**

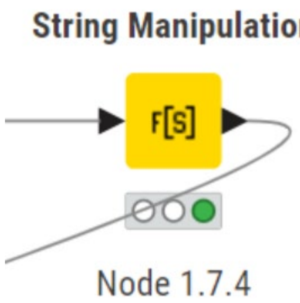
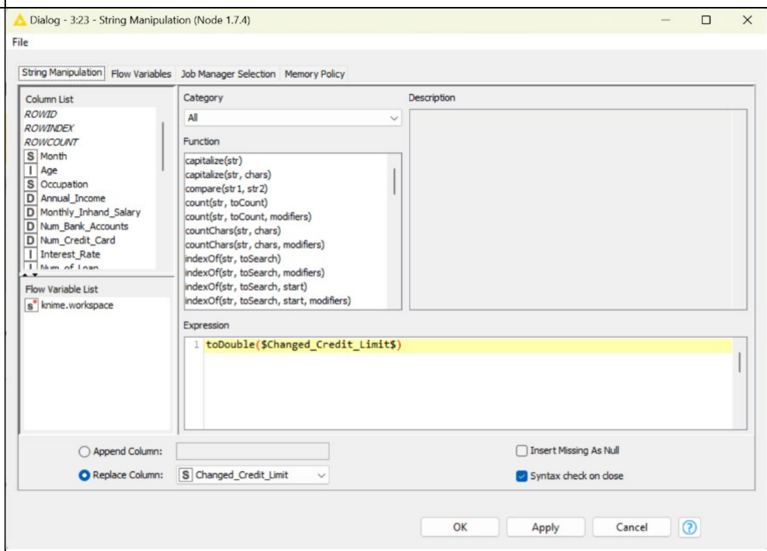
Sequence	Node	Command
1		 **Command:** strip(removeDuplicates(regexReplace($Credit_History_Age$,"[^0-9]"," ")))
2		 **Command:** toInt(strip(substr($Credit_History_Age_1$, 0, indexOf($Credit_History_Age_1$, " "))))

3		 **Command:** toInt(strip(substr($Credit_History_Age_1$, indexOf($Credit_History_Age_1$, " "))))
4		 **Command:** His_Age_Year * 12 + His_Age_Month

7) Remove the non-numerical symbol in “Amount_invested_monthly” and convert it to the double format. Set the value to “Unknow” if the original value in “Payment_Behaviour” attribute starts with “!@”. Remove the non-numerical symbol in “Monthly_Balance” and convert it to the double format. Convert “Changed_Credit_Limit” into the double format. List the node(s) (in sequence) and the corresponding command(s) used in this process. **[5 marks]**

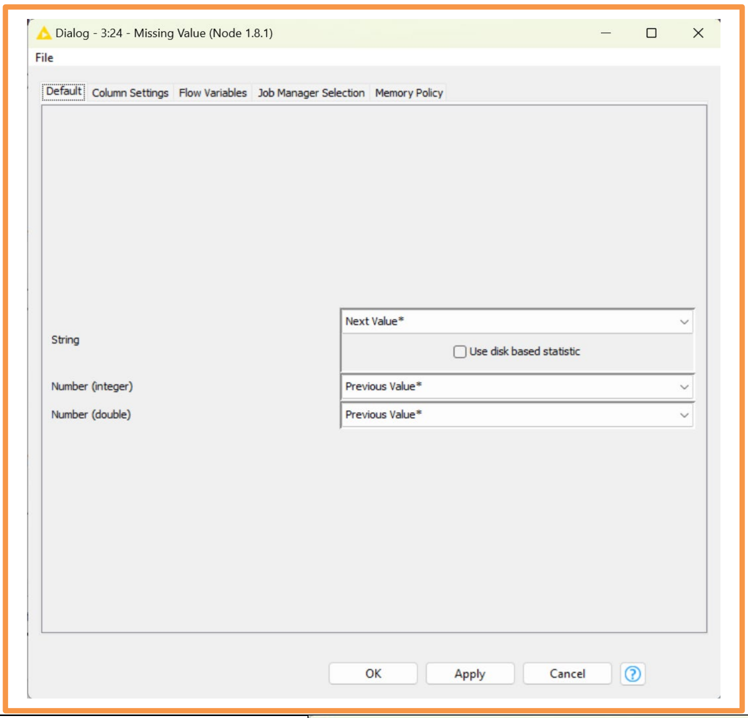
Sequence	Node	Command
1		 **Command:** toDouble(regexReplace($Amount_invested_monthly$,"[^0-9.-]",""))
2		 **Command:** $Payment_Behaviour$ LIKE “!@*” => “Unknow” TRUE => $Payment_Behaviour$

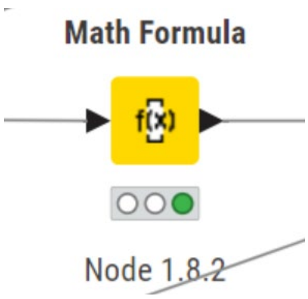
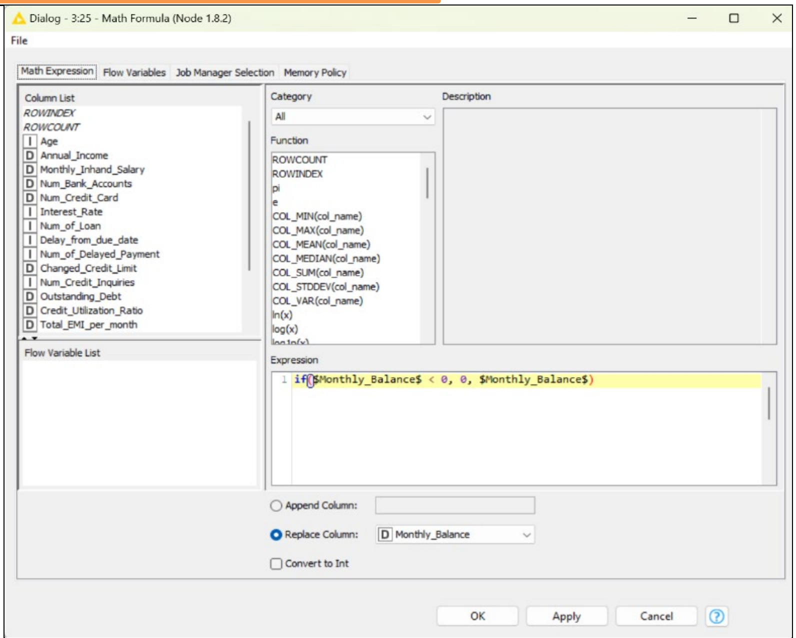
<table>
<tr>
<td>3</td>
<td>
</td>
<td>

Command:
toDouble(regexReplace($Monthly_Balance$,"[^0-9.-]", ""))</td>
</tr>
<tr>
<td>4</td>
<td>
</td>
<td>

Command:
toDouble($Changed_Credit_Limit$)</td>
</tr>
</table>

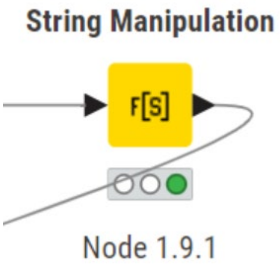
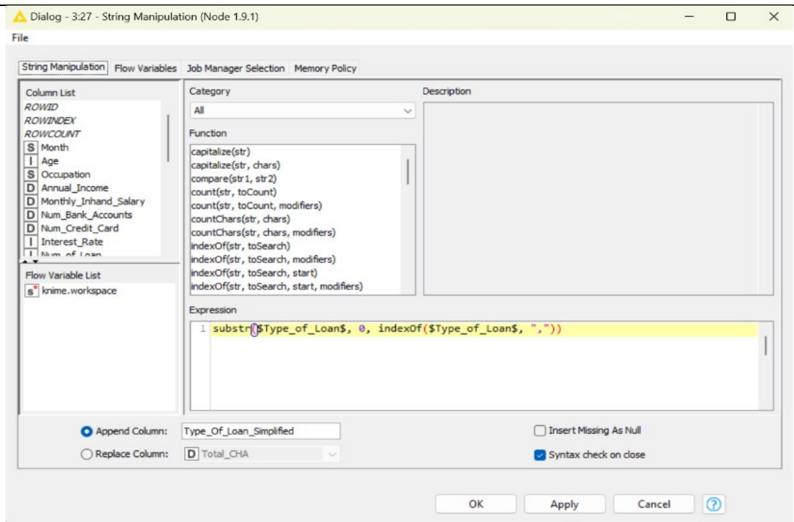
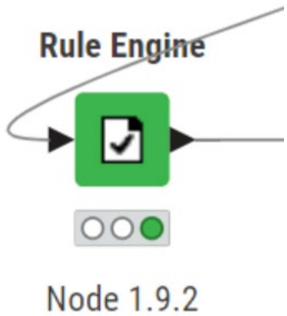
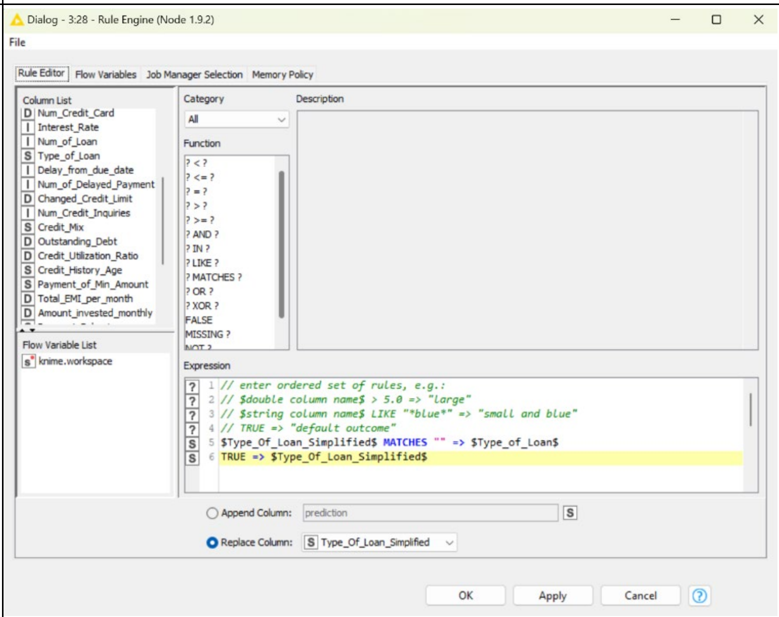
8) Use the “Missing Value” node and use the “Next Value*” to replace missing values in all string type attributes. Use the “Previous Value*” in the same node to replace missing values in any numerical format. If the value of “Monthly_Balance” is negative, replace the value with 0. Screenshot the pop-up window with the correct settings. **[5 marks]**



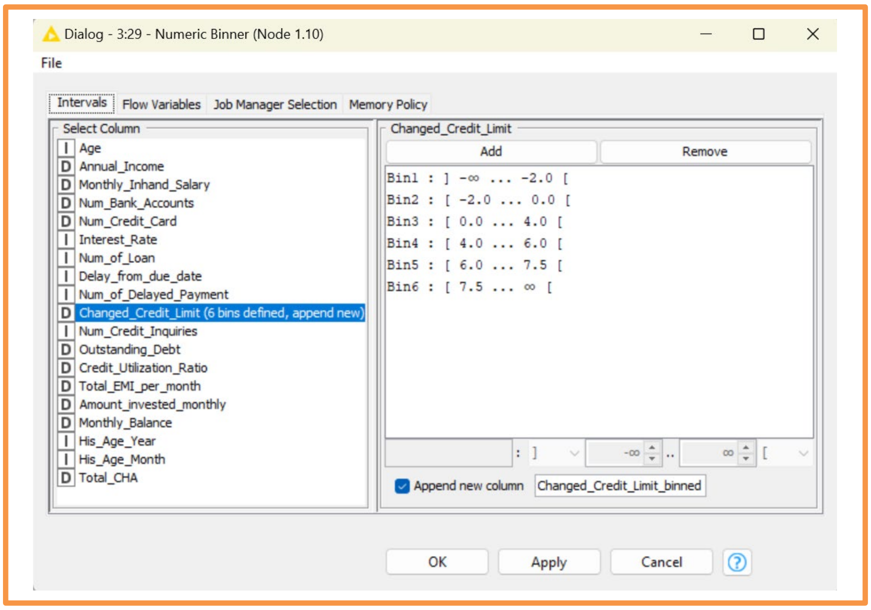
<table>
<tr>
<td>1</td>
<td>
</td>
<td>

Command:
if($Monthly_Balance$ < 0, 0, $Monthly_Balance$)</td>
</tr>
</table>

9) Simplify the “Type_of_Loan” attribute. If the original content has more than one type separated by a comma, keep only the first part. Otherwise, keep the full description if there is no comma included. For example, “Auto Loan, Credit-Builder Loan, Personal Loan, and Home Equity Loan” will become “Auto Loan”, “Credit-Builder Loan” will still be “Credit-Builder Loan”, and “Not Specified, Auto Loan, and Student Loan” will become “Not Specified” after the process. List the node(s) (in sequence) and the corresponding command(s) used in this process. **[10 marks]**

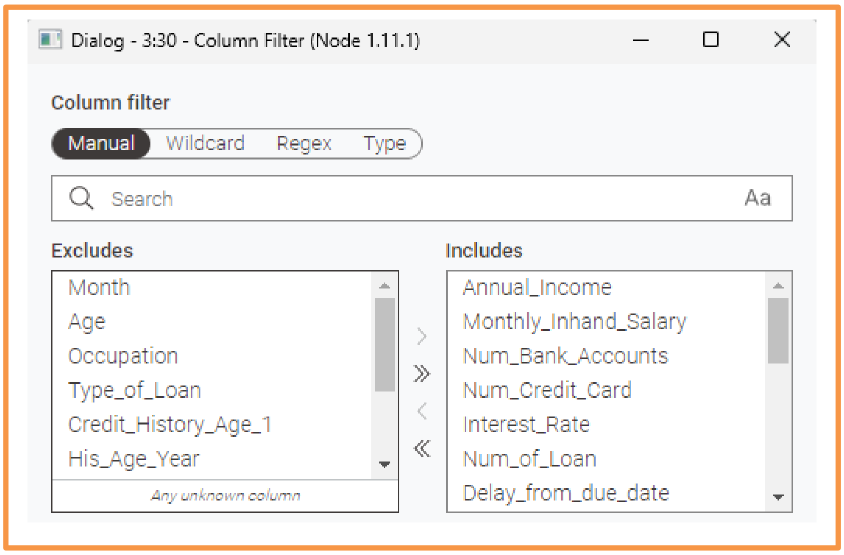
Sequence	Node	Command
1		 **Command:** substr($Type_of_Loan$, 0, indexOf($Type_of_Loan$, ","))
2		 **Command:** $Type_Of_Loan_Simplified$ MATCHES "" => $Type_of_Loan$ TRUE => $Type_Of_Loan_Simplified$

10)Bin the “Changed_Credit_Limit” attribute with six bins of ranges: $[-\infty, -2.0)$, $[-2.0, 0)$, $[0, 4.0)$, $[4.0, 6.0)$, $[6.0, 7.5)$, and $[7.5, \infty)$ and put the result into a new attribute called “Changed_Credit_Limit_binned”. Screenshot the pop-up window with the correct settings of your binner. **[5 marks]**

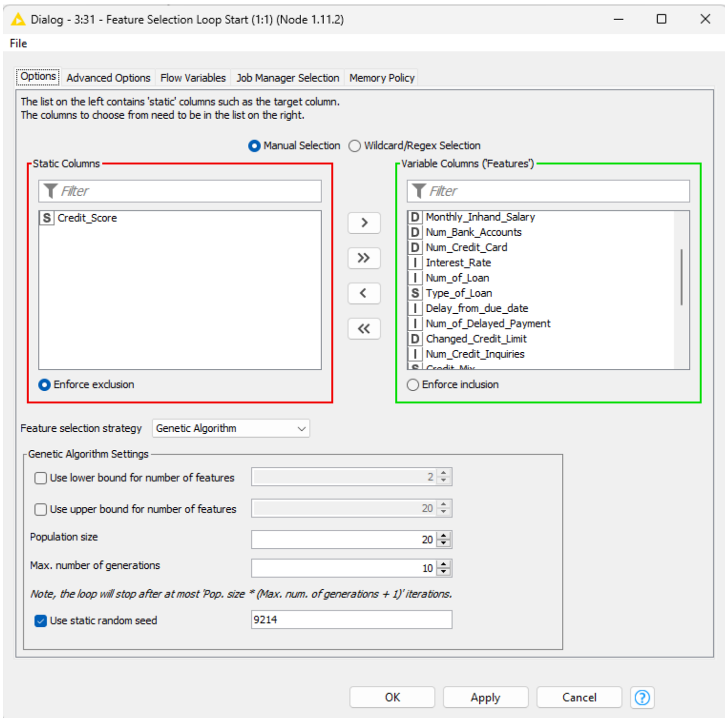


11)Remove all temporarily created or useless attributes. Use the “Feature Selection Loop Start (1:1)” node to select the feature. The class label should be excluded from the features in the feature selection node. The Genetic Algorithm is specified to be the feature selection strategy with default population size and the maximum number of generations. Again, **9214** should be used as the static random seed. After selecting features, shuffle the data with seed **9214**. The data should be partitioned by “Linear sampling”, with 80% data in the training set and 20% in the test set. How many tuples and attributes (excluding the class label) are in the training set at the end? **[5 marks]**

- **Remove all temporarily created or useless attributes:**

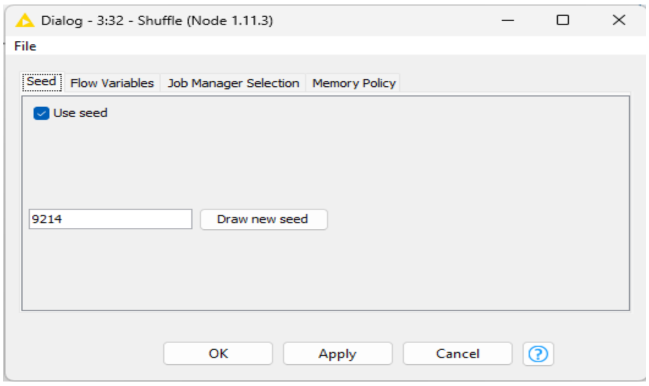


- **Feature Selection Loop Start (1:1):**

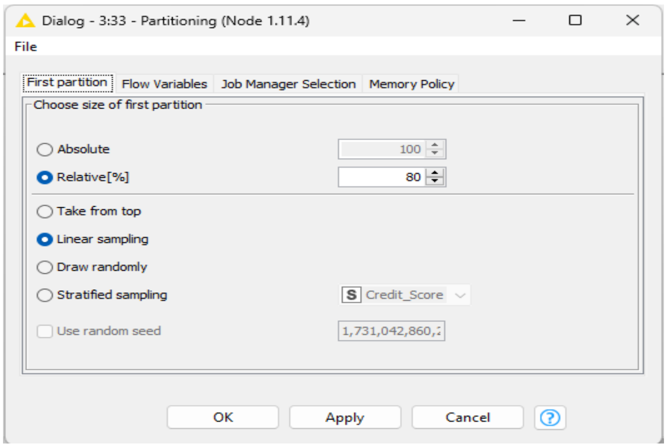


- **The number tuples and attributes are:**

SHUFFLE



PARTITIONING

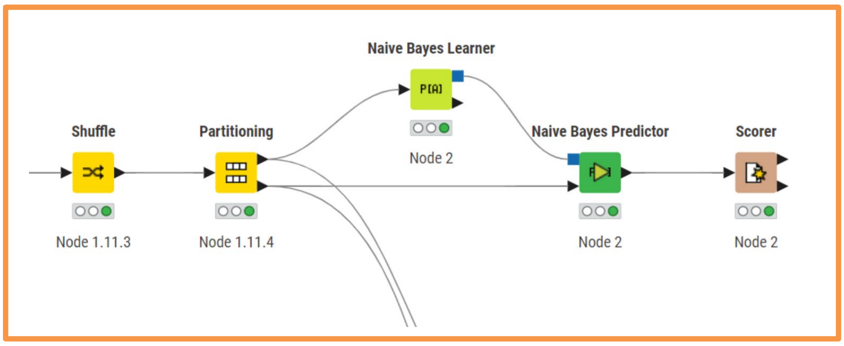


RESULT

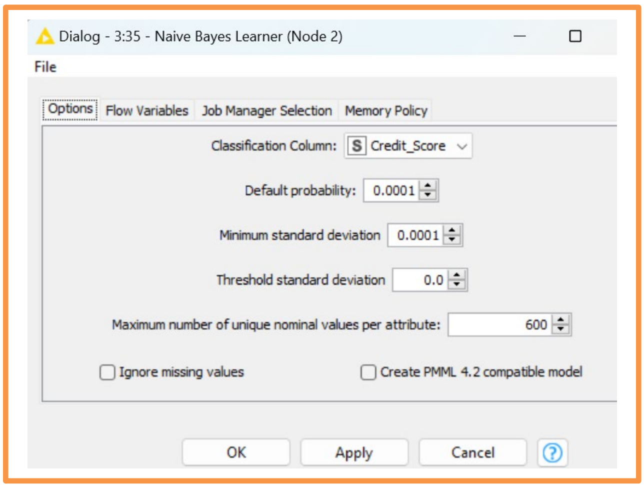
Training set: 72742
Test set: 18186

2. Build a Naïve Bayes classifier using the training and test sets created in the previous task. Answer the following questions after completing the model training and test. **[15 marks in total]**

1) Give a screenshot of the Naïve Bayes classifier in the KNIME workflow. You can take the screenshot starting from the portioning node output to the end of the Naïve Bayes classifier part scorer. **[2.5 marks]**



2) The default probability should be 0.0001, the minimum standard deviation is 0.0001, the threshold standard deviation is 0, and the maximum number of unique nominal values per attribute should be set to 600 in the classifier. Screenshot the setting dialogue of your Naïve Bayes Learner. **[2.5 marks]**



3) Screenshot the confusion matrix and the Accuracy statistics of the test result. If the bank wants to minimise the risk of lending money to customers, the "Good" in "Credit_Score" should be the major target. Based on the current result, does the classifier perform satisfactorily? **[5 marks]**

Confusion Matrix

Confusion matrix (Table)

Rows: 3 | Columns: 3

#	RowID	Good Number (integer)	Standard Number (integer)	Poor Number (integer)
1	Good	2603	612	70
2	Stan...	2010	5873	1750
3	Poor	757	1745	2766

Accuracy Statistics

#	RowID	TruePositives Number (integer)	FalsePositives Number (integer)	TrueNegatives Number (integer)	FalseNegativ... Number (integer)	Recall Number (double)	Precision Number (double)	Sensitivity Number (double)	Specificity Number (double)	F-measure Number (double)	Accuracy Number (double)	Cohen's kappa Number (double)
1	Good	2603	2767	12134	682	0.792	0.485	0.792	0.814	0.602	?	?
2	Stan...	5873	2357	6196	3760	0.61	0.714	0.61	0.724	0.658	?	?
3	Poor	2766	1820	11098	2502	0.525	0.603	0.525	0.859	0.561	?	?
4	Overall	?	?	?	?	?	?	?	?	?	0.618	0.398

- The classifier does not perform satisfactorily based on the current result, this is because the Precision for “Good” is only 0.485, which is relatively low.

4) Which measurement should we look at to interpret your conclusion in this case? **[5 marks]**

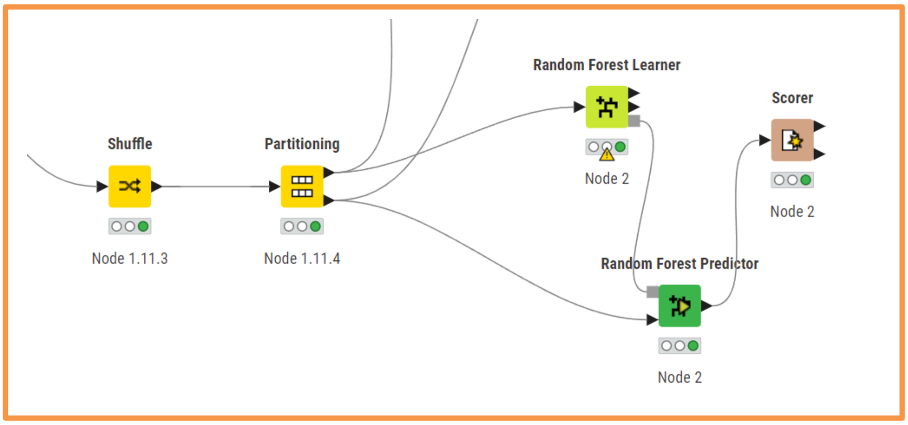
- If the bank wants to minimise the risk of lending money to customers, we must look at the Positive Predictive Value (PPV) – Precision.
- The formula for precision:

$$\text{Precision (PPV)} = \frac{TP\ (True\ Positive)}{TP\ (True\ Positive) + FP\ (False\ Positive)}$$

- In this case:
  + True Positive = True Good Customers (good customers that are correctly identified).
  + False Positive = False Good Customers (standard or poor customers but are incorrectly identified as good customers).
- If the bank wants to minimise the risk of lending money to customers, the number of False Positive cases must be low and the True Positive cases mut be high, which means the Precision (PPV) must be high. However, the value for Precision in this case is only 0.485, which is quite low.

3. Build a random forest classifier using the training and test sets created in the previous task. Answer the following questions after completing the model training and test. Use the information gain ratio as the split criterion and **9214** as the static random seed to build the random forest model. **[15 marks in total]**

1) Give a screenshot of the random forest classifier in the KNIME workflow. You can take the screenshot starting from the portioning node output to the end of the Naïve Bayes classifier part scorer. **[2.5 marks]**



2) Screenshot the confusion matrix and the Accuracy statistics of the test result. **[2.5 marks]**

Confusion Matrix

Rows: 3 | Columns: 3

#	RowID	Good Number (integer)	Standard Number (integer)	Poor Number (integer)
1	Good	2256	977	52
2	Stan…	853	7495	1285
3	Poor	169	1225	3874

Accuracy Statistics

#	RowID	TruePositives Number (integer)	FalsePositives Number (integer)	TrueNegatives Number (integer)	FalseNegativ… Number (integer)	Recall Number (double)	Precision Number (double)	Sensitivity Number (double)	Specificity Number (double)	F-measure Number (double)	Accuracy Number (double)	Cohen's kappa Number (double)
1	Good	2256	1022	13879	1029	0.687	0.688	0.687	0.931	0.687	?	?
2	Stan…	7495	2202	6351	2138	0.778	0.773	0.778	0.743	0.775	?	?
3	Poor	3874	1337	11581	1394	0.735	0.743	0.735	0.897	0.739	?	?
4	Overall	?	?	?	?	?	?	?	?	?	0.749	0.583

3) If the bank wants to minimise the risk of lending money to customers, the "Good" in "Credit_Score" should be the major target. Compare the measurements between random forest results and Naïve Bayes results. Which model presents a more suitable result? Which measure should be used to make the comparison? **[5 marks]**

- **Comparison for "Good" class:**

	Naïve Bayes	**Random Forest**
Precision	0.485	0.688

- In order to minimize the risk of lending money to customers, the major target is the "Good" in "Credit_Score". In the table above, the Random Forest model's Precision is higher than the Naïve Bayes model's (0.688 > 0.485). This shows that the Random Forest model has more correct predictions in terms of the Good customers, which will help the bank to achieve its goal. Therefore, the Random Forest model is more suitable in this case.

4) Which class does the built random forest model perform the best? What measurement(s) should we look at to find the answer? **[5 marks]**

Accuracy Statistics

Rows: 4 | Columns: 11

Table | Statistics

#	RowID	TruePosit... Number (integ...	FalsePosi... Number (integ...	TrueNega... Number (integ...	FalseNeg... Number (integ...	Recall Number (doub...	Precision Number (doub...	Sensitivity Number (doub...	Specificity Number (doub...	F-measure Number (doub...	Accuracy Number (doub...	Cohen's k... Number (doub...
1	Good	2256	1022	13879	1029	0.687	0.688	0.687	0.931	0.687	?	?
2	Stan...	7495	2202	6351	2138	0.778	0.773	0.778	0.743	0.775	?	?
3	Poor	3874	1337	11581	1394	0.735	0.743	0.735	0.897	0.739	?	?
4	Overall	?	?	?	?	?	?	?	?	?	0.749	0.583

- **Comparison between Classes:**

	Recall	**Precision**	**F-measure**
Good	0.687	0.688	0.687
Standard	0.778	0.773	0.775
Poor	0.735	0.743	0.739

- The measurements that we should look at to find the answer is the "Recall", "Precision", and "F-measure".
  + **Recall** – This measures the true positives out of all the total actual positives, therefore it is useful in understanding the model effectiveness.
  + **Precision** – This measures the proportion of true positives out of all predicted positives, which tells us how good the model in predicting values.
  + **F-measure** – This calculates the mean of the Precision and Recall, which is useful in case that there is an imbalance between recall and precision.
- Based on the summary table above, the values of Recall, Precision and F-measure for Standard class is higher than the others. Therefore, the Random Forest model performs best in the Standard class.

-- End of Submission --