

Cover sheet for submission of work for assessment



UNIT DETAILS

Unit name	Data Science Principles	Class day/time	Monday	Office use only
Unit code	COS10022	Assignment no.	2	Due date
			19/11/2023	
Name of lecturer/teacher	Pham Thi Kim Dung			
Tutor/marker's name	Pham Thi Kim Dung			
	Faculty or school date stamp			

STUDENT(S)

Family Name(s)	Given Name(s)	Student ID Number(s)
(1) Le Xuan	Nhat	104169523
(2)		
(3)		
(4)		
(5)		
(6)		

DECLARATION AND STATEMENT OF AUTHORSHIP

- I/we have not impersonated, or allowed myself/ourselves to be impersonated by any person for the purposes of this assessment.
- 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.
- 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.
- I/we have not previously submitted this work for this or any other course/unit.
- I/we give permission for my/our assessment response to be reproduced, communicated, compared and archived for plagiarism detection, benchmarking or educational purposes.

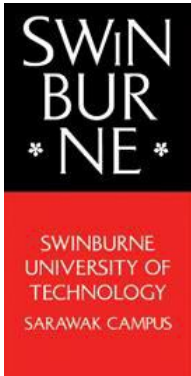
I/we understand that:

- 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)		(4)	
(2)		(5)	
(3)		(6)	



COS10022 Data Science Principles

Assignment 2 - Semester 1, 2023

Assessment Title: Data Cleaning and Analytics

Assessment Weighting: 30%

Due Date: Saturday, 14th May 2023 at 11.59 pm (AEDT)

Assessable Item:

- One (1) piece of a written report no more than 10-page long with the signed Assignment Cover Sheet.
- A unit peer must review your submission before it can be marked.

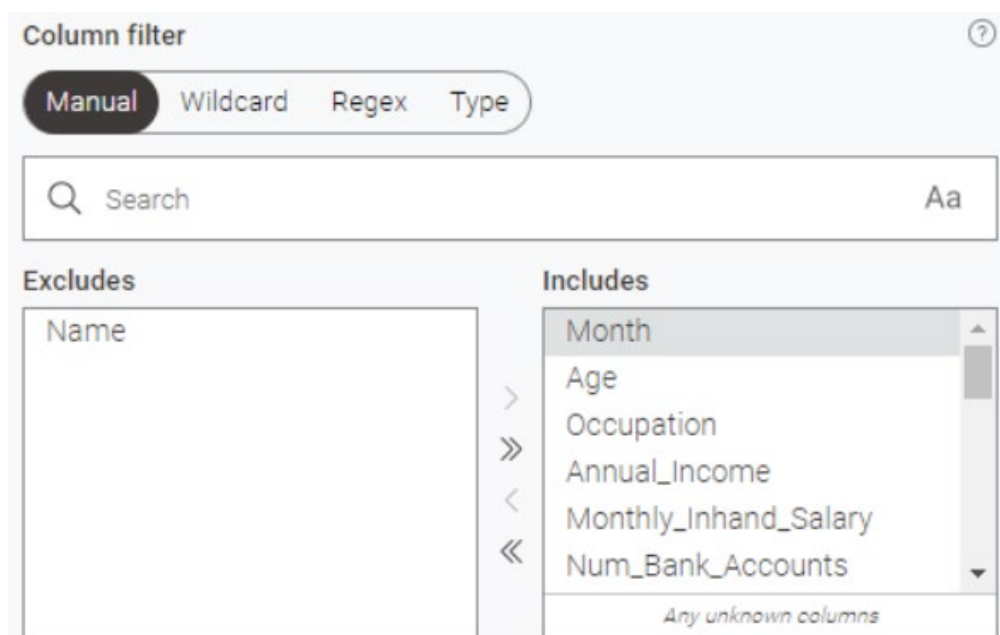
The submitted report should answer all questions listed in the assignment task section in sequence. You must include a digitally signed Assignment Cover Sheet with your submission.

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 **3122** to the seed to fix the random state. **[70 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. Exclude the attribute(s) and give a persuasive rationale for that. The excluded attribute(s) is(are)_____, and the reason(s) for removing it(them) is(are)_____. **[5 marks]**

Ans:

The excluded attribute is(are) "Name" because it is just an unique identifier that does not contribute to prediction models.

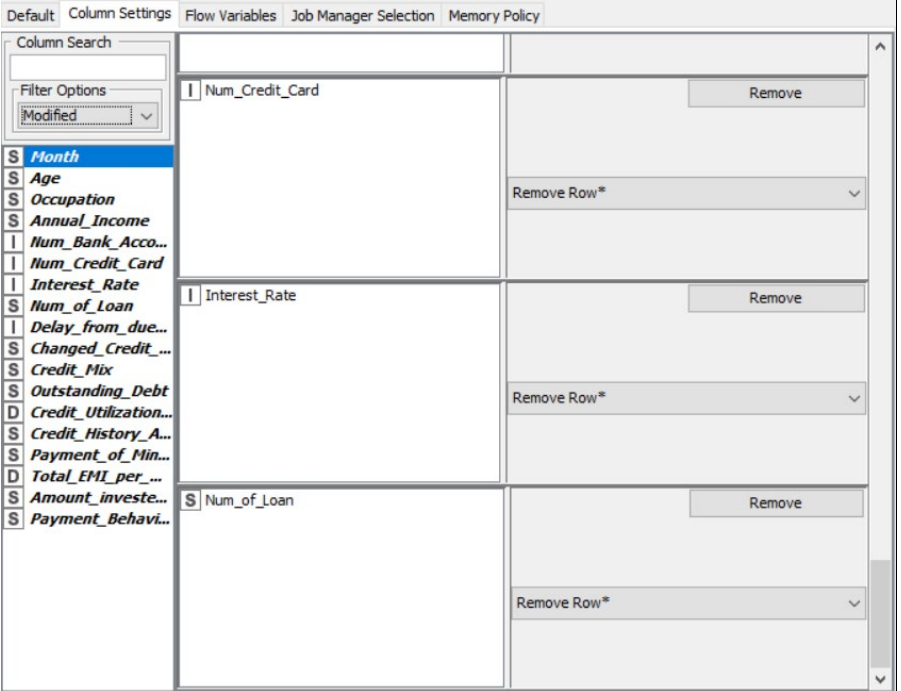


Using "Column Filter" to exclude "Name" attributes.

- 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]

Ans:

The used node are **"Missing Value"** and **"Rule-based Row Filter"**. The used commands are listed as follows:

Sequence	Node	Command
1	Missing Value	 <p>Filter Options "Modified" to show all changed columns.</p>
2	Rule-based Row Filter	<p>Expression</p> <pre> 1 // enter ordered set of rules, e.g.: 2 // \$double column name\$ > 5.0 => FALSE 3 // \$string column name\$ LIKE "*blue*" => FALSE 4 // TRUE => TRUE 5 \$Monthly_Inhand_Salary\$ < 0 => TRUE 6 \$Num_Bank_Accounts\$ < 0 => TRUE 7 \$Num_Credit_Card\$ < 0 => TRUE 8 \$Changed_Credit_Limit\$ MATCHES "[^0-9.-]" => TRUE </pre> <p><input type="radio"/> Include TRUE matches <input checked="" type="radio"/> Exclude TRUE matches</p> <p>Values that match the conditions are considered true and thus excluded from the table.</p>

- 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]

Ans:

Sequence	Node	Command
----------	------	---------

1	String Manipulation	<p>Expression</p> <pre>1 toInt(regexReplace(\$Age\$, "[^0-9.-]", ""))</pre> <p>We use regex to find non-numerical data and then replace it. [^0-9.-] means that all value that is NOT (^) from 0-9, or decimal point (.) or negative sign (-) are removed.</p>
2	Rule-based Row Filter	<p>Expression</p> <pre>1 // enter ordered set of rules, e.g.: 2 // \$double column name\$ > 5.0 => FALSE 3 // \$string column name\$ LIKE "*blue*" => FALSE 4 // TRUE => TRUE 5 \$Age\$ <= 0 => TRUE 6 \$Age\$ > 120 => TRUE</pre> <p><input type="radio"/> Include TRUE matches <input checked="" type="radio"/> Exclude TRUE matches</p> <p>We exclude any value that is lower or equal than 0 and greater than 120.</p>

- 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]**

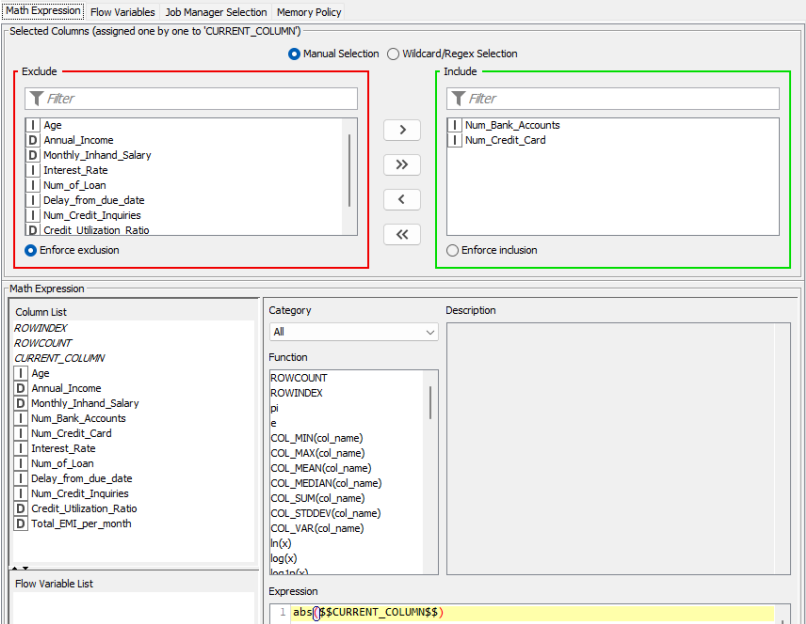
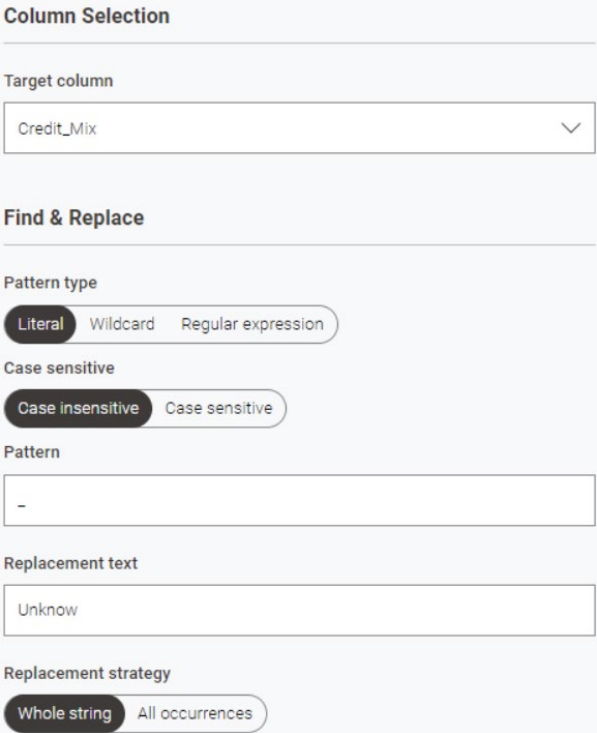
Ans:

Sequence	Node	Command
1	String Manipulation	<p>Expression</p> <pre>1 toDouble(regexReplace(\$Annual_Income\$, "[^0-9.-]", ""))</pre> <p>We remove non-numerical characters and convert to double.</p>

- 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]**

Ans:

Sequence	Node	Command
1	String Manipulation	<p>Expression</p> <pre>1 toNull(replace(\$Occupation\$, "_____", ""))</pre> <p>We replace “_____” with empty values, and then change them to null (missing) values.</p>
2	String Manipulation	<p>Expression</p> <pre>1 toInt(regexReplace(\$Num_of_Loan\$, "[^0-9.-]", ""))</pre> <p>We remove non-numerical values, then convert them from string to integers.</p>

3	Math Formula (Multi-Column)	 <p>Using this node, we include “Num_Bank_Accounts” and “Num_Credit_Card” and then convert them to absolute value.</p>
4	Math Formula	<p>Expression</p> <pre>1 if(\$Num_of_Loan\$ < 0, 0,\$Num_of_Loan\$)</pre> <p>We use the “if” condition here to convert any value that is below 0 to 0. If not, the value remains the same.</p>
5	String Manipulation	<p>Expression</p> <pre>1 toInt(regexReplace(\$Num_of_Delayed_Payment\$,"[^0-9.-]", ""))</pre> <p>Again, we use regex replace to remove non-numerical values, and then convert them to integers.</p>
6	String Replacer	 <p>Replace any instance of “_” to “Unknow” (according to the assignment requirements)</p>
7	String Manipulation	<p>Expression</p> <pre>1 toDouble(regexReplace(\$Outstanding_Debt\$,"[^0-9.-]", ""))</pre> <p>Remove non-numerical values and then convert to double.</p>

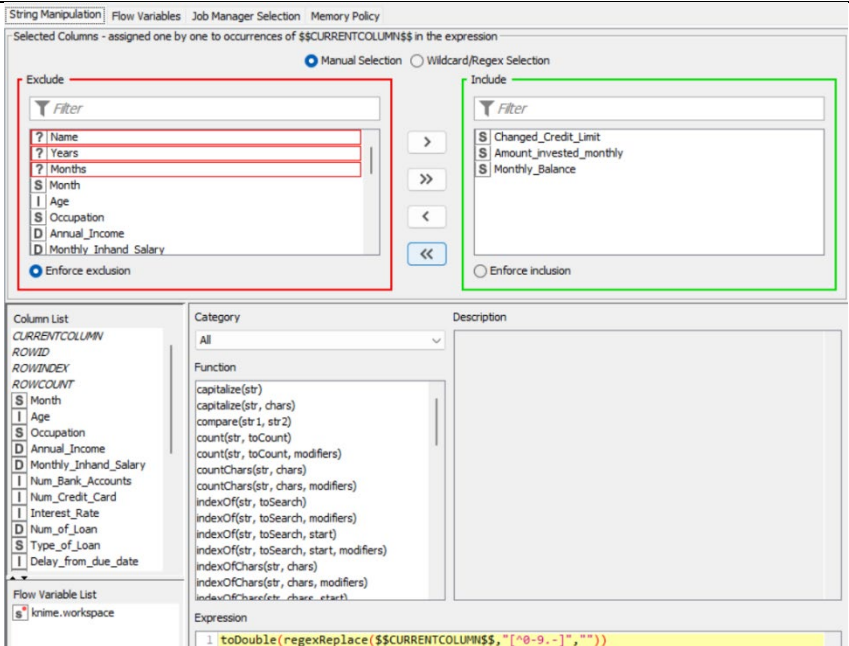
- 6) Convert the “Credit_History_Age” to the count of months and store it in the integer format. For example, if the original value from a tuple is “22 Years and 1 Months”, the value will be 265 after the conversion ($22 * 12 + 1 = 265$). Store the converted result in a new attribute called “Total_CHA.”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

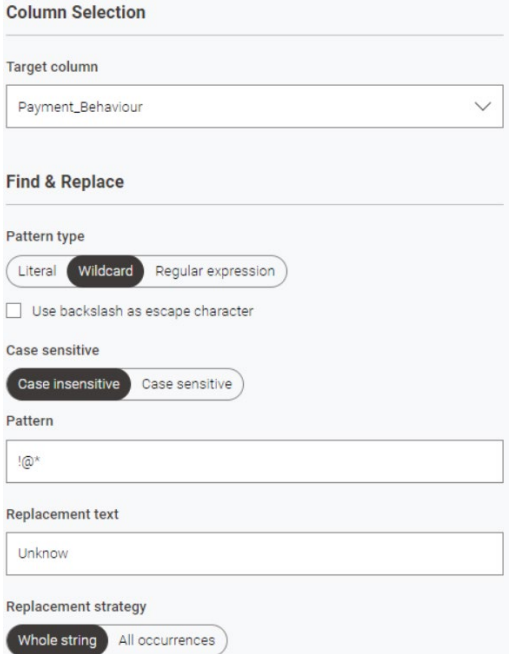
Ans:

Sequence	Node	Command
1	String Manipulation	<pre>toInt(strip(substr(\$Credit_History_Age\$,0,2)))*12 + toInt(strip(substr(\$Credit_History_Age\$,indexOf(\$Credit_History_Age\$,"d")+1,3)))</pre> <p>I use a singular node to extract the year value, convert to integer, multiply by 12 and then add the month value. First, I use the “substr” command to extract the year value. The year value is either 1 or 2 digits at the start of the string (for example: 22 years and 1 months). That is why in “substr” clause we have 0, 2 as starting from position zero (from the beginning) and takes 2 characters. However, sometimes the year number is a single-digit number “6 Years and 8 Months” so I add “strip” to remove whitespace for a successful conversion to integer. Take that value and times 12 and we have the years converted to months. To extract the months value, I also use “strip” like before, but “substr” is a little different. Because the length of the string varies, in this case I must use “indexOf” to find the index of the character “d” in the word “and” plus one (so that the function does not extract the letter “d”) and takes 3 characters after that to account for both single and double digit numbers.</p>

- 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]**

Ans:

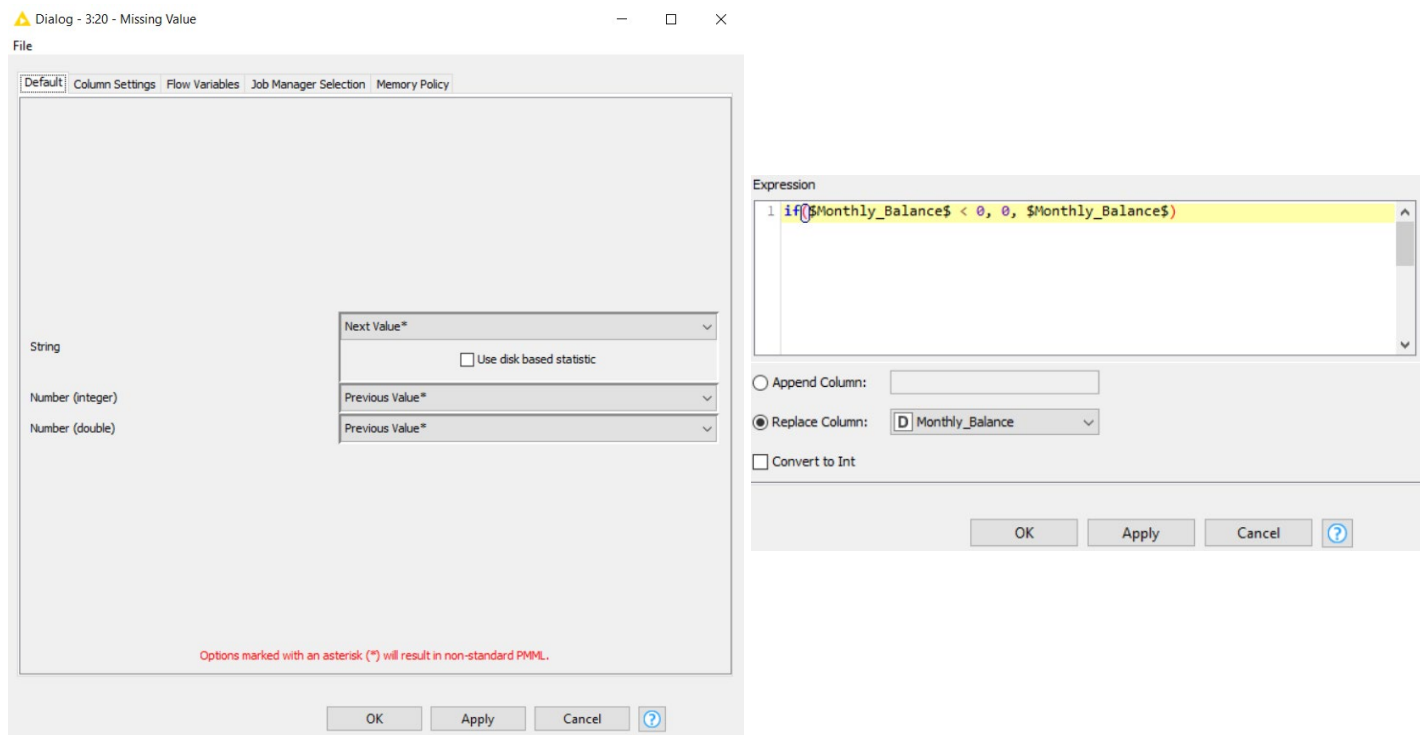
Sequence	Node	Command
1	String Manipulation (Multi-Column)	 <p>We use this node to clean the data and convert to integer.</p>

2	String Replacer	 <p>In this node, we use wildcard pattern to find and replace any string starting with “!@” and replace.</p>
---	-----------------	--

- 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]**

Ans:

These are the screenshots of nodes “Missing Value” and “Math Formula”. In the “Math Formula” node we use this if clause to determine if the value is below 0 or not. If the value is negative, we convert to 0, else the value stays the same.



Dialog - 3:20 - Missing Value

File

Default | Column Settings | Flow Variables | Job Manager Selection | Memory Policy

String

Number (integer)

Number (double)

Next Value*

Previous Value*

Previous Value*

Options marked with an asterisk (*) will result in non-standard PMML.

OK | Apply | Cancel | ?

Expression

1 if(\$Monthly_Balance < 0, 0, \$Monthly_Balance)

Append Column:

Replace Column: Monthly_Balance

Convert to Int

OK | Apply | Cancel | ?

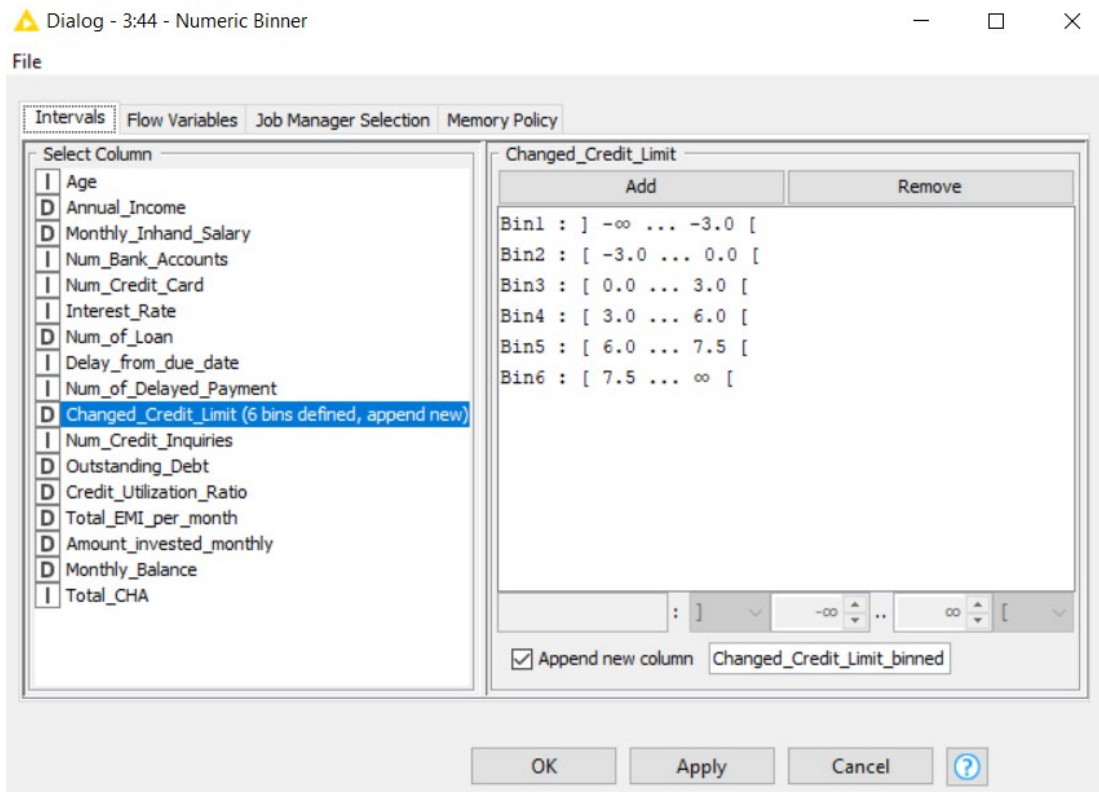
- 9) Simplify the “Type_of_Loan” attribute. If the original content has more than one type separated by a comma, keep only the first part. Otherwise, keep the full description if there is no comma included. For example, “Auto Loan, Credit-Builder Loan, Personal Loan, and Home Equity Loan” will become “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]**

Ans:

Sequence	Node	Command
1	String Manipulation	
2	Rule Engine	

- 10) Bin the “Changed_Credit_Limit” attribute with six bins of ranges: $[-\infty, -3.0)$, $[-3.0, 0)$, $[0, 3.0)$, $[3.0, 6.0)$, $[6.0, 7.5)$, 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]**

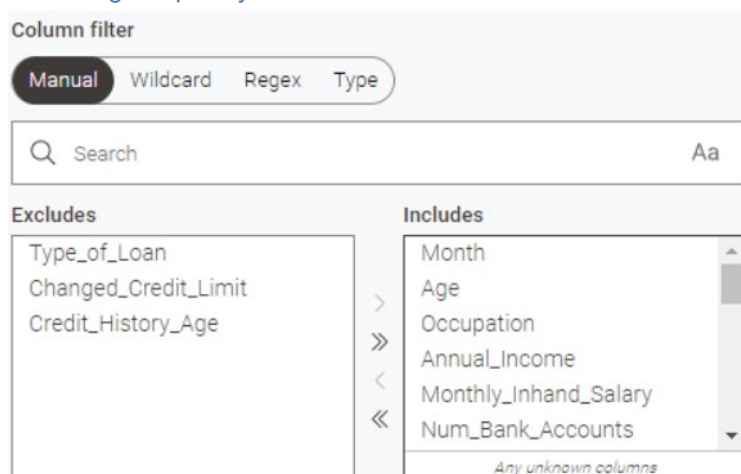
Ans:



- 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, **3122** should be used as the static random seed. After selecting features, shuffle the data with seed **3122**. The data should be partitioned by “Linear sampling”, with 75% data in the training set and 25% in the test set. How many tuples and attributes (excluding the class label) are in the training set at the end? **[5 marks]**

Ans:

Removing temporary classes with Column Filter:



Removing attribute “Credit Score” in the Feature Selection Loop Start (1:1) because it is a labeling attribute:

The list on the left contains 'static' columns such as the target column. The columns to choose from need to be in the list on the right.

☒ Manual Selection ☐ Wildcard/Regex Selection

Static Columns

Filter

S Credit_Score

☒ Enforce exclusion

Variable Columns (Features)

Filter

S Month
I Age
S Occupation
D Annual_Income
D Monthly_Inhand_Salary
I Num_Bank_Accounts
I Num_Credit_Card
I Interest_Rate
D Num_of_Loan
I Delay_from_due_date
I Num_of_Delivered_Payment

☐ Enforce inclusion

Feature selection strategy: Genetic Algorithm

Genetic Algorithm Settings

☐ Use lower bound for number of features: 2

☐ Use upper bound for number of features: 20

Population size: 20

Max. number of generations: 10

Note, the loop will stop after at most 'Pop. size * (Max. num. of generations + 1)' iterations.

☒ Use static random seed: 3122

There are **68196 tuples and 13 attributes** in training set at the end.

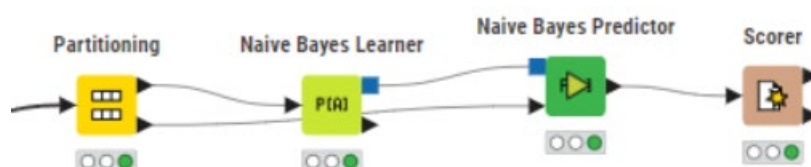
► 1: First partition (as defined in dialog)

Rows: 68196 | Columns: 13

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 partitioning node output to the end of the Naïve Bayes classifier part scorer. **[2.5 marks]**

Ans:

Naive Bayes Classifier



- 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]**

Ans:

- 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]**

<input type="checkbox"/>	RowID	Good Number (integer)	Standard Number (integer)	Poor Number (integer)
<input type="checkbox"/>	Good	3376	620	107
<input type="checkbox"/>	Standard	3069	6651	2363
<input type="checkbox"/>	Poor	1034	1636	3876

<input type="checkbox"/>	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)
<input type="checkbox"/>	Good	3376	4103	14526	727	0.823	0.451	0.823	0.78	0.583		
<input type="checkbox"/>	Standard	6651	2256	8393	5432	0.55	0.747	0.55	0.788	0.634		
<input type="checkbox"/>	Poor	3876	2470	13716	2670	0.592	0.611	0.592	0.847	0.601		
<input type="checkbox"/>	Overall										0.612	0.404

Ans:

Rationales:

If the “Good” class is the major target for us to determine the performance of the classifier, then based on the result the Naïve Bayes model performs poorly due to the fact that precision is very low, with a score of 0.451.

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

Ans:

To assess the classifier's performance in minimizing the risk of lending money to customers, with a focus on correctly identifying "Good" customers, the key metric to consider is Precision for the "Good" class:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

This metric is crucial because it measures the accuracy of the classifier when it predicts a customer as "Good." A high precision indicates that the classifier is effective in avoiding false positives, reducing the risk of wrongly classifying customers as "Good" when they are actually “Standard” or “Poor” Therefore, to determine if the classifier performs satisfactorily in the context of minimizing lending risk, evaluating precision for the "Good" class is essential.

There are other measurements to consider here. That is Recall (or Sensitivity), Specificity, and F-measure.

1. Recall (Sensitivity):

Recall is a metric that focuses on the ability of the classifier to correctly identify all instances of a specific class. In the context of minimizing the risk of lending money, it is relevant for capturing as many "Good"

customers as possible. However, it has a limitation as it does not account for false positives, which are crucial to minimize in this task.

2. Specificity:

Specificity measures the ability of the classifier to correctly identify all instances of the negative class. While it is relevant for avoiding false positives, it may not be the most suitable metric for lending scenarios, as it emphasizes the accurate identification of "Standard"/"Poor" customers without addressing the need to identify all "Good" customers.

3. F-measure (or F1 Score):

The F-measure, or F1 Score, is the harmonic mean of precision and recall, providing a balance between the two metrics. While it is a good overall metric, it may not be the most suitable if there is a significant imbalance between the importance of false positives and false negatives. In the context of lending, precision for the "Good" class becomes more crucial, as it directly aligns with the goal of minimizing the risk of lending money by accurately identifying trustworthy customers.

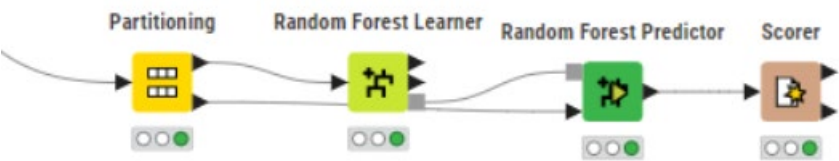
Therefore, the best measurement for this scenario is "precision".

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 3122 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]

Ans:

Random Forest Classifier



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

Ans:

<input type="checkbox"/> RowID	Good <small>Number (integer)</small>	Standard <small>Number (integer)</small>	Poor <small>Number (integer)</small>
<input type="checkbox"/> Good	2854	1168	81
<input type="checkbox"/> Standard	1055	9475	1553
<input type="checkbox"/> Poor	223	1559	4764

<input type="checkbox"/> RowID	TruePositives <small>Number (integer)</small>	FalsePositives <small>Number (integer)</small>	TrueNegatives <small>Number (integer)</small>	FalseNegatives <small>Number (integer)</small>	Recall <small>Number (double)</small>	Precision <small>Number (double)</small>	Sensitivity <small>Number (double)</small>	Specificity <small>Number (double)</small>	F-measure <small>Number (double)</small>	Accuracy <small>Number (double)</small>	Cohen's kappa <small>Number (double)</small>
<input type="checkbox"/> Good	2854	1278	17351	1249	0.696	0.691	0.696	0.931	0.693		
<input type="checkbox"/> Standard	9475	2727	7922	2608	0.784	0.777	0.784	0.744	0.78		
<input type="checkbox"/> Poor	4764	1634	14552	1782	0.728	0.745	0.728	0.899	0.736		
<input type="checkbox"/> Overall										0.752	0.587

- 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]

Ans:

	Naïve Bayes model	Random Forest model
Precision (Good)	0.612	0.752
Accuracy	0.451	0.691

The Random Forest model demonstrates a substantial improvement over the Naïve Bayes model, evident in both overall accuracy and precision. As previously discussed, precision is a crucial metric for this comparison, as it specifically evaluates the model's ability to correctly classify "Good" customers. The emphasis on precision aligns with the objective of minimizing the risk of lending money. Additionally, considering accuracy provides a comprehensive overview of the model's overall performance, making it a valuable metric for the comparison between the two models.

- 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]**

Ans:

The top three measurements to look at are precision, recall, and F1 score when comparing the performance of classes.

	Recall	Precision	F1 score
Good	0.696	0.691	0.693
Standard	0.784	0.777	0.78
Poor	0.728	0.745	0.736

Precision is to measure how many times a model correctly predicts a specific class out of all predictions for that class. Recall is the frequency with which the model correctly predicts a specific class out of all instances of that class in the dataset. The F1 score is a mixture of precision and recall that provides an estimation of the model's overall effectiveness for that class. However, in this context, we must prioritize "precision" due to the fact that in this context, the cost of false positives (accidentally lending money to "poor" customers) is higher than false negatives (missing some of the customers who would be capable of paying back the loan). After all due consideration, "Standard" is the best-performing class because not only it is the class with the highest precision rate, but also recall and F1 score as well.

----- End of Submission -----