Cover sheet for submission of work for assessment

SWIN BUR NE

SWINBURNE UNIVERSITY OF TECHNOLOGY

UNIT DETAILS

Unit name	Data Science Principles		Class day/time	Friday	Office use only
Unit code	COS10022	Assignment no. 2	Due date	12/Nov/2023	
Name of lecturer/teacher	Huy Truong				
Tutor/marker's name	Huy Truong				Faculty or school date stamp

STUDENT(S)

	Family Name(s)	Given Name(s)	Student ID Number(s)
(1)	Nguyen Dinh	Nhat Minh	103802490

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)

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/**

Swinburne University of Technology Hawthorn Campus
Dept. of Computer Science and Software Engineering

COS10022 Data Science Principles

Assignment 2 - *Semester3 2023*

Assessment Title: Predictive Model Creation and Evaluation

Assessment Weighting: 30%

Due Date: Saturday, 12th November 2023 at 11.59 pm (GMT+7)

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]**

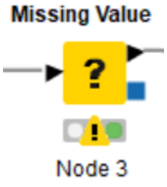
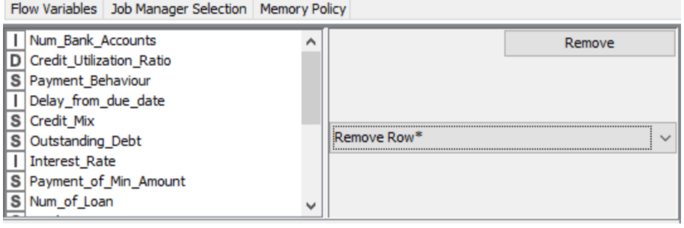
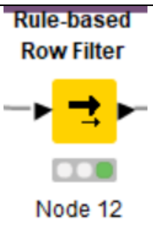
      Ans:

      Since these attributes are needed for later task, I would clean the raw data first and then remove the selected attributes just before partitioning and importing into the models.

Removed Attributes	Reason
Month	We do not rely on the period to determine our credit score, hence this information is redundant and should be remove.
Name	Since our primary focus is determining credit scores, a secondary identifier is unnecessary. Additionally, using personal information can pose privacy concerns.
Age	Using personal information to assess creditworthiness is often illegal.
Occupation	Although this attribute can be related to loan repayment ability and credit improvement, it is less significant compared to income or credit history age.
Type_of_Loan	Similar to occupation, while the types of loans may demonstrate one's ability to repay loans, is not necessarily more significant than, for example, the number of loans.
Payment_Behaviour	The number of delayed payments and payment due dates is a more decisive factor, making this attribute somewhat redundant

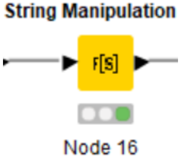
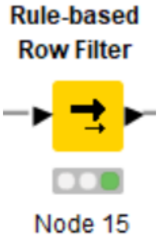
2) After removing the selected attribute(s), let's start to remove tuples containing missing values. Remove tuples only if any of the attributes listed below have missing values: "Month," "Age," "Occupation," "Annual_Income," "Num_Bank_Accounts," "Num_Credit_Card," "Interest_Rate," "Num_of_Loan," "Delay_from_due_date," "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:

No	Nodes	Command	Explanation
1		Flow Variables | Job Manager Selection | Memory Policy I Num_Bank_Accounts D Credit_Utilization_Ratio S Payment_Behaviour I Delay_from_due_date S Credit_Mix S Outstanding_Debt I Interest_Rate S Payment_of_Min_Amount S Num_of_Loan Remove Remove Row*	To make sure the defined rows that has missing value are removed, we use the 'Missing Value' nodes with the 'Remove Row' configuration.
3		$Monthly_Inhand_Salary$ **< 0** => **TRUE** $Num_Bank_Accounts$ **< 0** => **TRUE** Num_Credit_Card **< 0** => **TRUE** $Changed_Credit_Limit$ **MATCHES "[^0-9.-]"** => **TRUE** '*Exclude TRUE matches' option choose*	Using the defined condition, we are able to roughly clean the data and complete the task at hand.

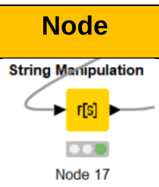
3) Check for the "Age" attribute to eliminate symbols that are not numbers to recover the data into the usual number format. Moreover, drop the tuples whose "Age" value is lower than or equal to 0 or greater than 120. List the node(s) (in sequence) and the corresponding command(s) used in this process.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

Ans:

No	Nodes	Command	Explanation
1		toInt(regexReplace(Age,**"[^0-9.-]"**,**""))**	Since we would be recovering the values that should be numeric but aren't. In this case, we would use regular expression to replace non-numerical characters with an empty character, effectively removing the non-numerical characters.
2		$Age_Cleaned$ **<= 0 AND** $Age_Cleaned$ **>=** 120 => **TRUE** '*Exclude TRUE matches' option choose*	Setting the parameters to eliminate the ages outside of 0 and 120.

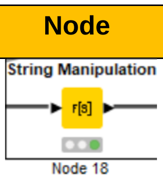
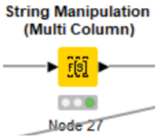
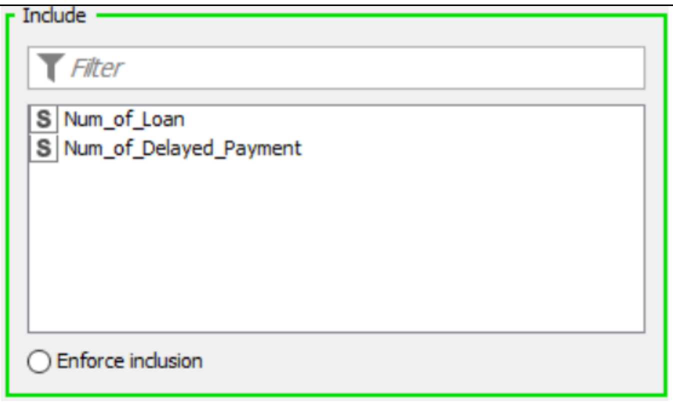
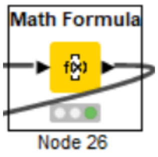
4) Remove the non-numerical symbol in the “Annual_Income” column and convert it to the double format. List the node(s) (in sequence) and the corresponding command(s) used in this process. **[5 marks]**

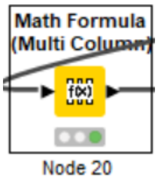
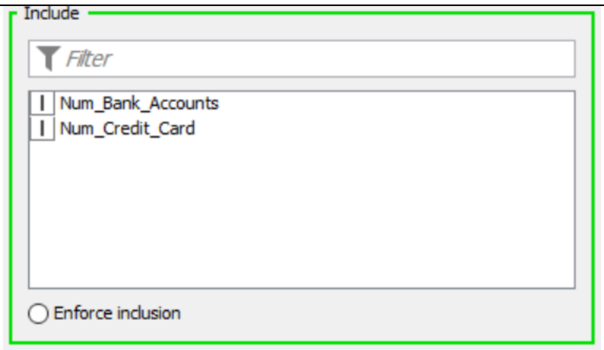
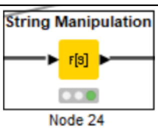
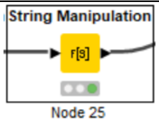
Ans:

No	Node	Command	Explanation
1		toDouble(regexReplace($Annual_Income$,"[^0-9.-]",""))	We would use regular expression to replace non-numerical characters with an empty character, effectively removing the non-numerical characters.

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]**

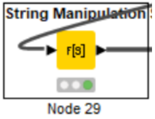
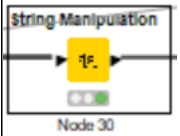
Ans:

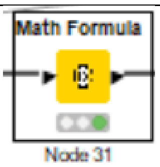
No	Node	Command	Explanation
1		toNull(replace($Occupation$, "________",""))	Replace the defined attribute to an empty string, then use toNull function to replace empty string with null value.
2		 Include Filter S Num_of_Loan S Num_of_Delayed_Payment ○ Enforce inclusion toInt(regexReplace($$CURRENTCOLUMN$$,"[^0-9.-]",""))	Replace non-numerical character with empty string, then convert the type of the selected column to integer. And since both “Num_of_Loan” and “Num_of_Delayed_Payment” has the same requirement, we use he multi column option of ‘String Manipulator’ to do our bidding.
3		if(Num_of_Loan<0,0,Num_of_Loan)	Replace the integer with value < 0 with a 0 after converting ‘Num_of_Loan’ to integer.

4		Include Filter Num_Bank_Accounts Num_Credit_Card ○ Enforce inclusion abs($$CURRENT_COLUMN$$)	Since these 2 column has the same requirement, we use the 'Math Formular (Multi Column)' to get the absolute value of both of them at the same time.
5		replace($Credit_Mix$, "_", "Unknown")	Replace the defined parameter with "Unknown"
6		toDouble(regexReplace($Outstanding_Debt$,"[^0-9.-]",""))	Replace non-numerical character with empty string, then convert the type of the selected column to double

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]**

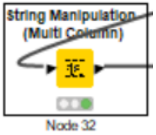
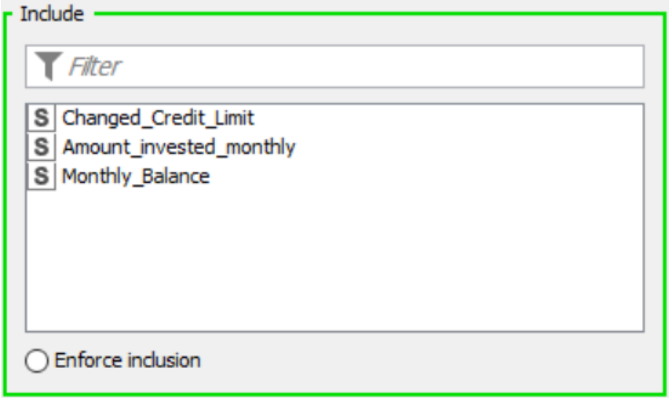
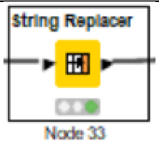
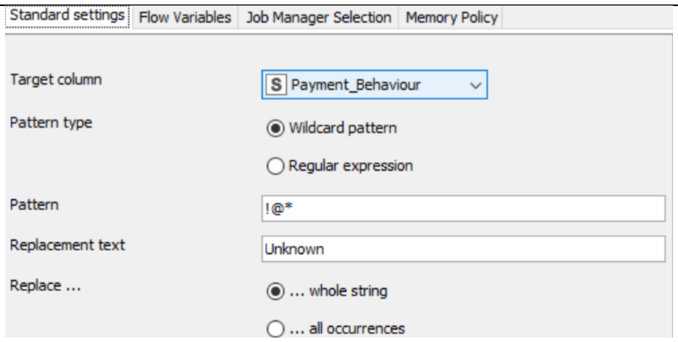
Ans:

No	Node	Command	Explanation
1		toInt(strip(substr($Credit_History_Age$,0,2)))	This node handles the "Credit_History_Age" by first getting the first 2 digit in the string. Then trim all the trailing space where applicable. Finally convert the value to Integer. We append the new column named "CHA_Year" to store the value of this operation
2		toInt(strip(substr($Credit_History_Age$,indexOf($Credit_History_Age$, "d") + 1,3)))	The same treatment is used for this node, with the addition of getting the position after the letter "d" to act as the starting point of the substring. We use +1 because we want to not get the letter "d" into our result. We append the new column named "CHA_Month" to

			store the value of this operation
3		CHA_Year * 12 + CHA_Month	Multiply the year by 12 and add the month to get the total CHA.

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]**

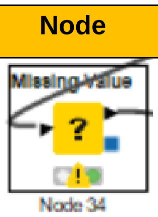
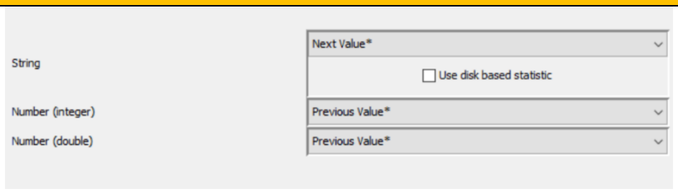
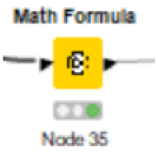
Ans:

No	Node	Command	Explanation
1		Include Filter S Changed_Credit_Limit S Amount_invested_monthly S Monthly_Balance ○ Enforce inclusion toDouble(regexReplace($$CURRENTCOLUMN$$,"[^0-9.-]",""))	Since “Amount_invested_monthly”, “Payment_Behaviour”, and “Changed_Credit_Limit” has similar condition, we would use one multi Column string manipulator to handles them. Even though “Changed_Credit_Limit” does not need non-numerical character removed. We would include it to make the most out of the to Double function, beside it add an extra layer of cleansing.
2		 Standard settings Flow Variables Job Manager Selection Memory PolicyTarget column: S Payment_Behaviour Pattern type: ● Wildcard pattern ○ Regular expression Pattern: !@* Replacement text: Unknown Replace ...: ● ... whole string ○ ... all occurrences	We use ‘String Replacer’ node to change all occurrence of “!@” in the “Payment_Behaviour” column. And since we want to replace the starting pattern, we put an asterisk at the end to get the value after the defined pattern.

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

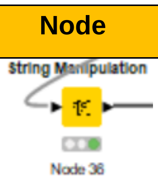
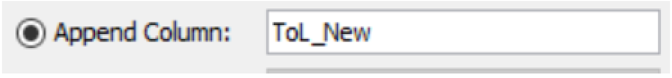
settings. **[5 marks]**

Ans:

No	Node	Command	Explanation
1		String: Next Value* ☐ Use disk based statistic Number (integer): Previous Value* Number (double): Previous Value*	Configured according to the task requirement.
2		if($Monthly_Balance$<0,0,$Monthly_Balance$)	Replace the number with value < 0 with a 0. If the condition false, the value remain the same.

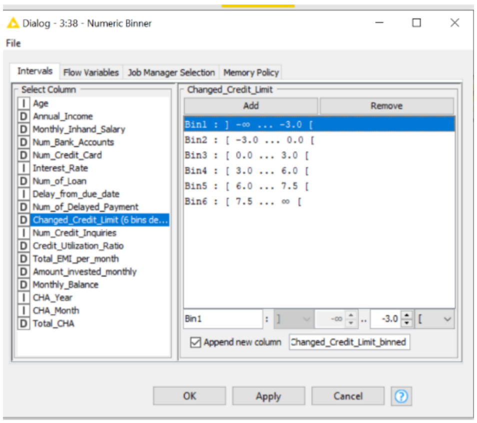
9) Simplify the “Type_of_Loan” attribute. If the original content has more than one type separated by a comma, keep only the first part. Otherwise, keep the full description if there is no comma included. For example, “Auto Loan, Credit-Builder Loan, Personal Loan, and Home Equity Loan” will become “Auto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

No	Node	Command	Explanation
1		substr($Type_of_Loan$, 0, indexOf($Type_of_Loan$, ",")) ● Append Column: ToL_New	In this command, we get the position of the “,” first, this will also act as our string length. Then we get the subtring starting from the 0 index position to get the first description of the “Type_of_Loan”. We store the handled value under a new column named “ToL_New”. Empty value can be present after this step, which is why we would need to use another node.
2		ToL_New LIKE "" => $Type_of_Loan$ TRUE => ToL_New	This node replace the empty string in the “ToL_New” column with the value in the original “Type_of_Loan” column. If the string is not empty, it remains the new value.

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:



Dialog - 3:38 - Numeric Binner

File

Intervals | Flow Variables | Job Manager Selection | Memory Policy

Select Column:

```
I Age
D Annual_Income
D Monthly_Inhand_Salary
D Num_Bank_Accounts
D Num_Credit_Card
I Interest_Rate
D Num_of_Loan
I Delay_from_due_date
D Num_of_Delayed_Payment
D Changed_Credit_Limit (6 bins de...
I Num_Credit_Inquiries
D Credit_Utilization_Ratio
D Total_EMI_per_month
D Amount_invested_monthly
D Monthly_Balance
I CHA_Year
I CHA_Month
D Total_CHA
```

Changed_Credit_Limit

Add | Remove

```
Bin1 : ] -∞ ... -3.0 [
Bin2 : [ -3.0 ... 0.0 [
Bin3 : [ 0.0 ... 3.0 [
Bin4 : [ 3.0 ... 6.0 [
Bin5 : [ 6.0 ... 7.5 [
Bin6 : [ 7.5 ... ∞ [
```

Bin1 :] -∞ .. -3.0 [

☑ Append new column: Changed_Credit_Limit_binned

OK | Apply | Cancel | ?

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]**

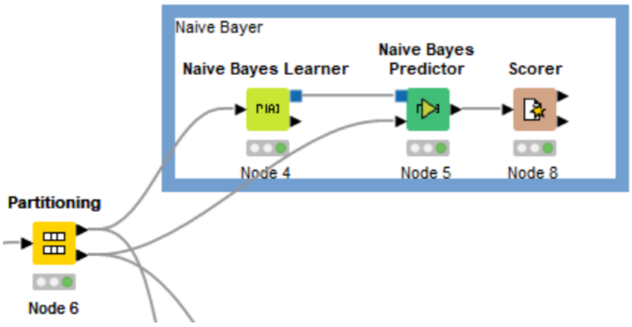
Ans:

There are **68196 tuples and 10 attributes** in training set at the end.

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]**

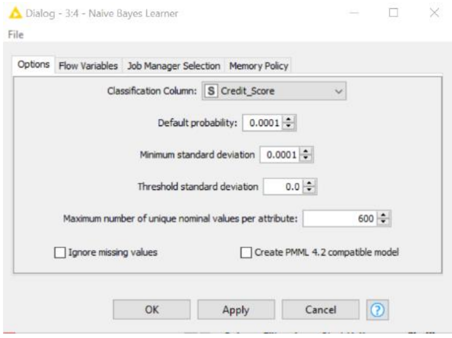
1) Give a screenshot of the Naïve Bayes classifier in the KNIME workflow. You can take the screenshot starting from the portioning node output to the end of the Naïve Bayes classifier part scorer. **[2.5 marks]**

Ans:



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:



Dialog - 3:4 - Naive Bayes Learner

File

Options | Flow Variables | Job Manager Selection | Memory Policy

Classification Column: S Credit_Score

Default probability: 0.0001

Minimum standard deviation 0.0001

Threshold standard deviation 0.0

Maximum number of unique nominal values per attribute: 600

□ Ignore missing values

□ Create PMML 4.2 compatible model

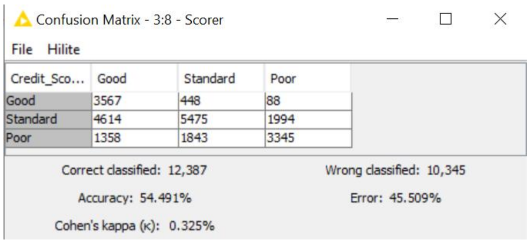
OK | Apply | Cancel | ?

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]**

Ans:

Confusion matrix:



Confusion Matrix - 3:8 - Scorer

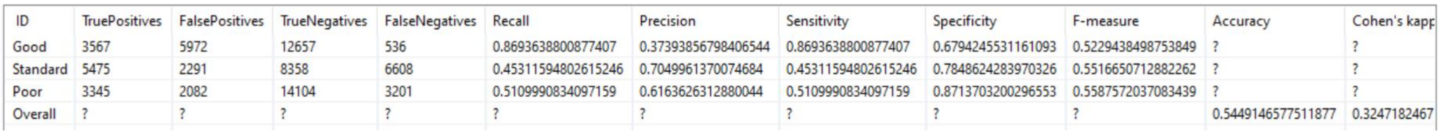
Credit_Sco...	Good	Standard	Poor
Good	3567	448	88
Standard	4614	5475	1994
Poor	1358	1843	3345

Correct classified: 12,387 Wrong classified: 10,345

Accuracy: 54.491% Error: 45.509%

Cohen's kappa (κ): 0.325%

Accuracy statistic:



ID	TruePositives	FalsePositives	TrueNegatives	FalseNegatives	Recall	Precision	Sensitivity	Specificity	F-measure	Accuracy	Cohen's kapp
Good	3567	5972	12657	536	0.8693638800877407	0.37393856798406544	0.8693638800877407	0.6794245531161093	0.5229438498753849	?	?
Standard	5475	2291	8358	6608	0.45311594802615246	0.7049961370074684	0.45311594802615246	0.7848624283970326	0.5516650712882262	?	?
Poor	3345	2082	14104	3201	0.5109990834097159	0.6163626312880044	0.5109990834097159	0.8713703200296553	0.5587572037083439	?	?
Overall	?	?	?	?	?	?	?	?	?	0.5449146577511877	0.3247182467

From the result of the Naïve Bayes model, we can say that the classifier does not perform very well when it comes to identifying good credit since the precision for identifying “Good” credit is rather low. Hence, we can’t say that it satisfies the criteria of minimising lending risk for banks.

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

Ans:

In this model, our primary focus is on the "Good" class because it serves as the most reliable determinant for identifying eligible borrowers with minimal risk for the bank. While the precision rate for this class may be relatively low, it remains the key factor in assessing who can safely borrow money.

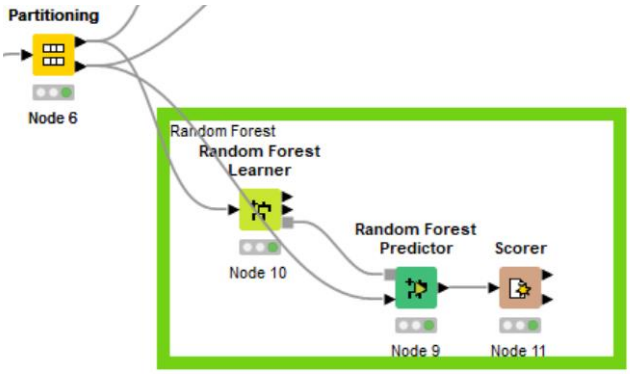
This approach is reasonable because individuals with "Standard" or "Poor" credit are less likely to repay the money to the bank on time or even at all. Lending to such individuals increases the risk of the bank losing money. On the other hand, selecting accounts with "Good" credit credibility implies a higher likelihood of timely repayment, thereby reducing the risk of financial loss for the bank. By accurately identifying and approving accounts with a "Good" credit status, the bank can significantly enhance its safety and confidence in minimizing the lending risk.

On the other hand, even though the recall rate for the “Good” class is high, we cannot rely on it as it dismissed all the case of “Standard” or “Poor” that are wrongly determined as “Good”. As we want to have an accurate vision, relying on potentially misleading numbers can prove disastrous. Regarding F-measure, it is a combination of both recall and precision rate, hence, it still confines the risk of generalizing the data and provide inaccurate calculations to determine lending risks.

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]**

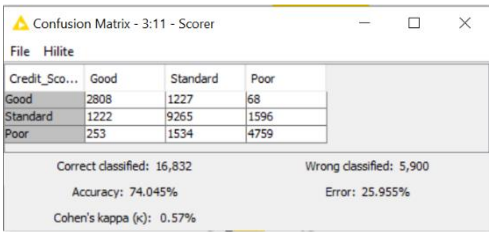
Ans:



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

Ans:

Confusion matrix:



Confusion Matrix - 3:11 - Scorer

Credit_Sco...	Good	Standard	Poor
Good	2808	1227	68
Standard	1222	9265	1596
Poor	253	1534	4759

Correct classified: 16,832

Wrong classified: 5,900

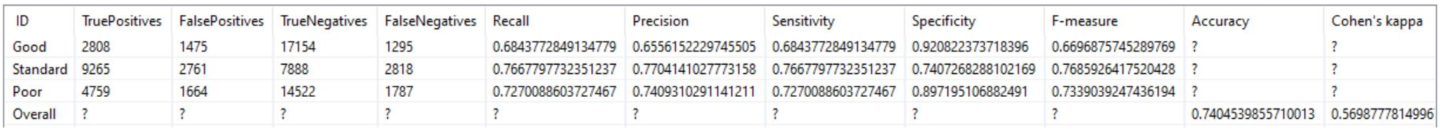
Accuracy: 74.045%

Error: 25.955%

Cohen's kappa (κ): 0.57%

Accuracy statistic:

It's evident that the Random Forest model outperforms the Naïve Bayes model in this comparison. The Random Forest model demonstrates a higher overall accuracy of over 70%. In contrast, the Naïve Bayes model achieves just over 50% accuracy, meaning it correctly predicts a lower percentage of the classes.



ID	TruePositives	FalsePositives	TrueNegatives	FalseNegatives	Recall	Precision	Sensitivity	Specificity	F-measure	Accuracy	Cohen's kappa
Good	2808	1475	17154	1295	0.6843772849134779	0.6556152229745505	0.6843772849134779	0.920822373718396	0.6696875745289769	?	?
Standard	9265	2761	7888	2818	0.7667797732351237	0.7704141027773158	0.7667797732351237	0.7407268288102169	0.7685926417520428	?	?
Poor	4759	1664	14522	1787	0.7270088603727467	0.7409310291141211	0.7270088603727467	0.897195106882491	0.7339039247436194	?	?
Overall	?	?	?	?	?	?	?	?	?	0.7404539855710013	0.5698777814996

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:

As stated in task 2.4 , we would rely on the precision rate, as the recall and F-measures only give an overall view and not the accurate result banks would want in the decision making process of lending money.

Naïve Bayes	Random Forest	Measurement
0.3739	0.6556	Precision

Comparing the precision rate between the model side by side, we can clearly see that the Random Forest model performs much better than Naïve Bayes, this is due to the fact that Random Forest is generally superior to Naïve Bayes in determining account credibility due to its ability to model complex relationships, handle irrelevant features, manage imbalanced data, mitigate overfitting, and provide higher accuracy, making it a more suitable choice for tasks like credit scoring where precision is crucial for minimizing risk.

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:

Performance in this case would be measured by the high precision of the model in picture.

Class	Random Forest	Measurement
Good	0.6556	Precision
Standard	0.7704	
Poor	0.7409	

Oberserving the precision of the Random Forest model, we can see that the "Standard" Class has the highest precision rate between the 3 classes. Although the standard class wouldn't be considered in minimizing lending risk, its high precision still proves that for complex model, Random Forest is the best for determining accuracy, but it does come at the cost of being rather slow and resource-intensive.

--- End of Submission ---