COMP 1942 Project Phase 3

Wijaya, Andrew Sebastian - 20994984

Kusnadi, Vicko Nicholas - 21022506

Group No. 61

I. Design Report

1. Introduction

In this report, we are going to design a report for a project that utilizes Classification with XLMiner. The project is designed to develop five distinct models, each offering unique parameters and prediction results that will be used in Phase 3 of the Project. The data used in this project comes from four sources. The first, referred to as “Data 1”, is raw data extracted from the training sheet of Phase 1. The second, “Data 2”, is a transformation of Data 1 into a numerical format, with the 'native-country' data attribute removed to streamline the analysis. The third one, called “Data 3” is raw data extracted from the test sheet of Phase 1. Finally, The fourth one is called “Data 4” which is a numerical form of “Data 3”. The third and fourth data will be used to predict the result of our model

The classification learning models used in this project include the Decision Tree, Naive Bayesian, and K-Nearest Neighbor models. Each of these models brings a different approach to classification, providing a comprehensive analysis of the data. Through this project, we aim to explore the capabilities of these models and to guide the reader through the steps of each model, using the power of XLMiner to drive our data analysis.

2. Data Preparation

To prepare the data we first duplicate the page “Training” twice and rename one of the duplicates into “Data 1” as shown on Fig 1.0. Then we split data, selecting the first 8000 entries as the training data and next 2000 entries as the testing data. This step is shown in Fig 1.1 and Fig 1.2.

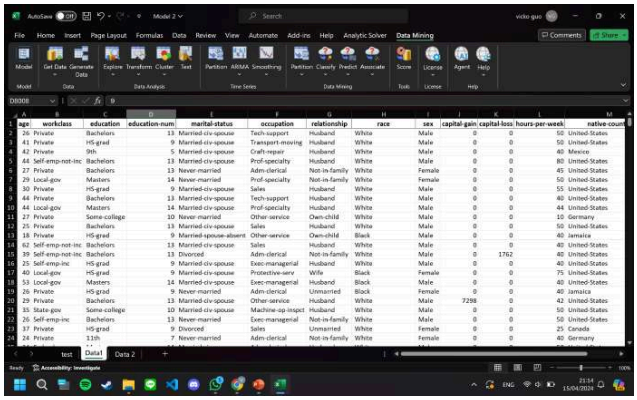


Fig 1.0

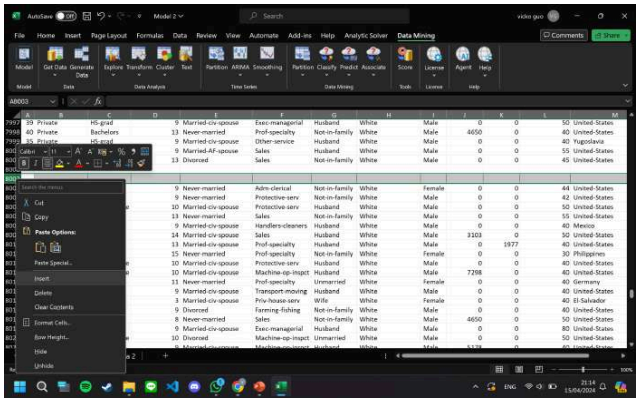


Fig 1.1

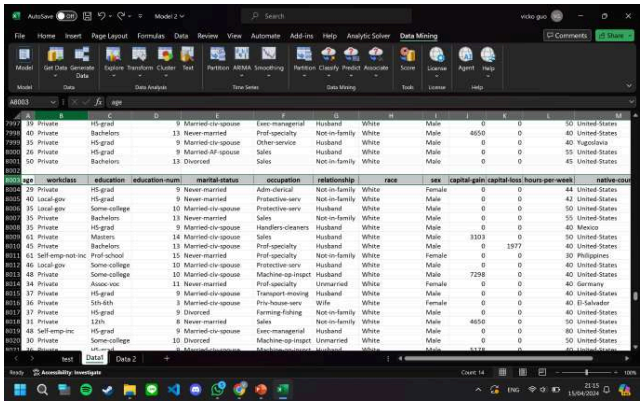


Fig 1.2

The next step is to take the second duplicate of the “Training” page from phase 1 and transform it into numerical data (as shown on Fig. 2.0.0). We select all data attributes, except “native-country” attributes and several other attributes that already have numerical value. We don’t use “native-country” attribute as it exceeds the limit of XLminer, and transform it into numerical data (as shown on Fig. 2.0.1). The resulting data is shown on Fig. 2.0.2.

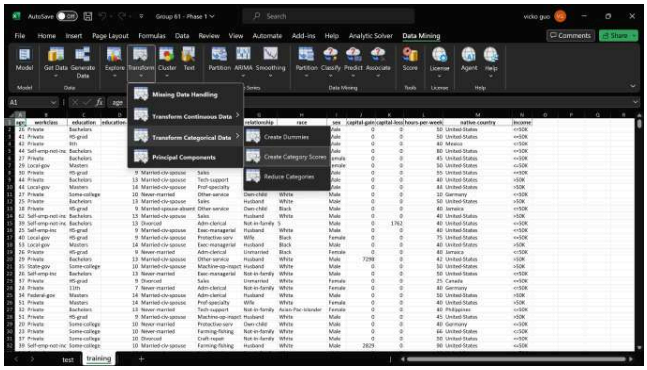


Fig. 2.0.0

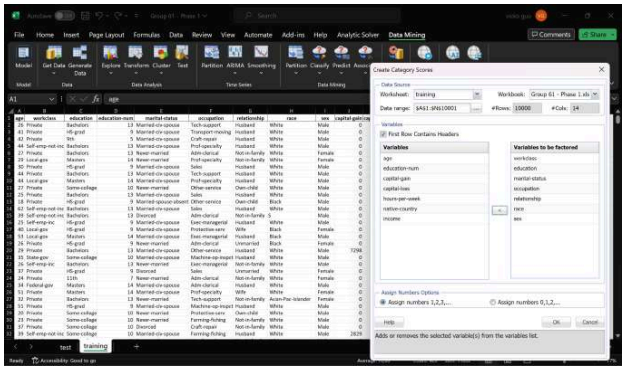


Fig. 2.0.1

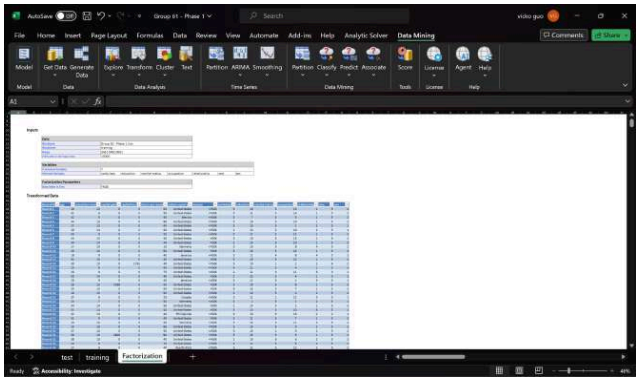


Fig. 2.0.2. Resulting numerical data from transformation .

The result of the numerical data page is renamed to “Data 2” (as shown on Fig. 2.1.0). We will now take this data and split the data for training and testing. In this design report, we will use the first 8000 entries for training and the next 2000 entries for testing. We split the data by inserting 2 new rows below the 8000th entry and copying the corresponding attribute names there (as shown in Fig. 2.1.1). The final result of our preparation for “Data 2” is shown in Fig. 2.1.2 and Fig. 2.1.3. That concludes the data preparation step needed to start generating our model. Our model will then use “Data 1” and “Data 2”. Next, to generate “Data 3”, we renamed the “test” sheet to “Data 3” (as shown on FIg. 2.1.4).

Finally, we will also show the necessary steps to create “Data 4”, which is the numerical data of “Data 3”. Take the “Test” sheet page from phase 1 and transform it into numerical data similar with generating “Data 2”. We select all data attributes, except “native-country” attributes and several other attributes that already have numerical value. We don’t use “native-country” attribute as it exceeds the limit of XLminer, and transform it into numerical data. The result of the data is shown in Figure 2.2.7

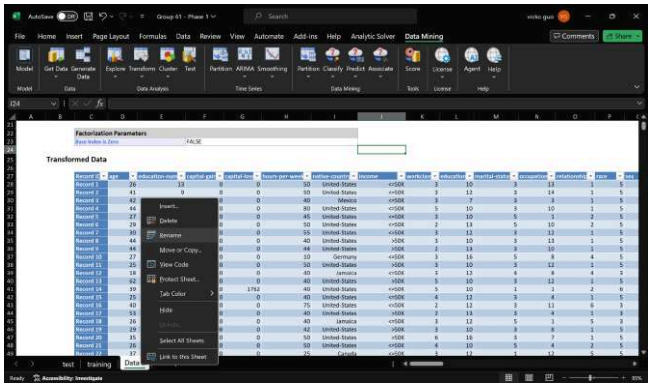


Fig. 2.1.0. Rename the data as "Data 2"

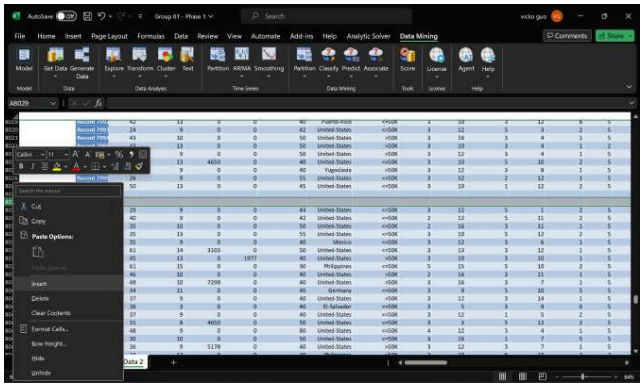


Fig. 2.1.1

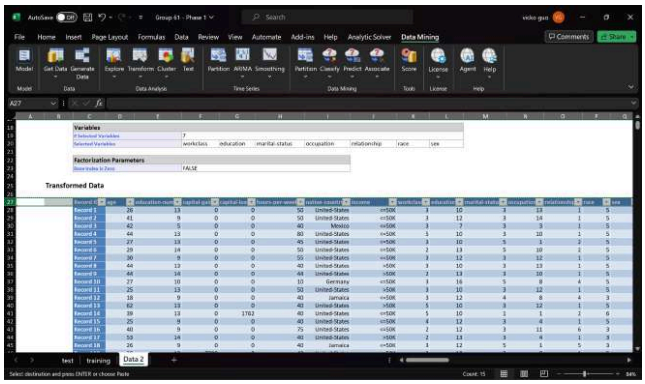


Fig. 2.1.2

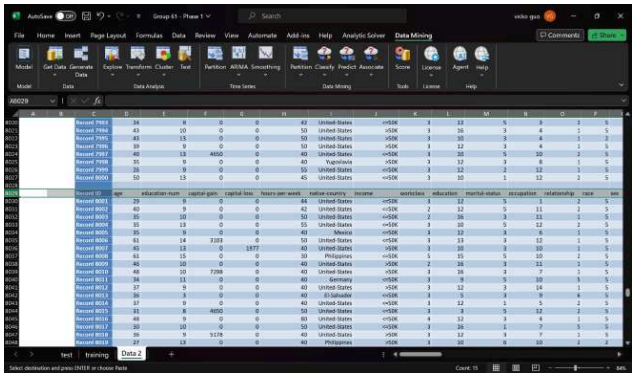


Fig 2.1.3

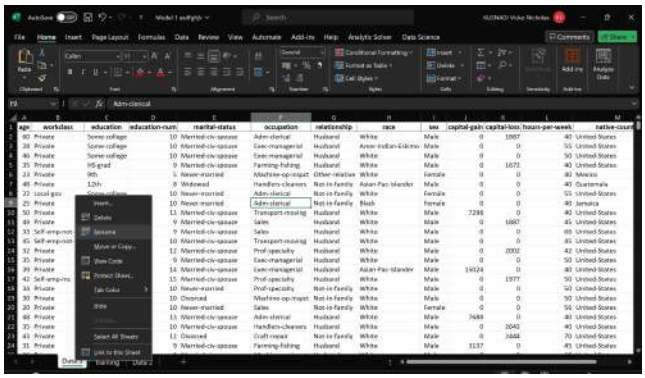


Fig 2.1.4

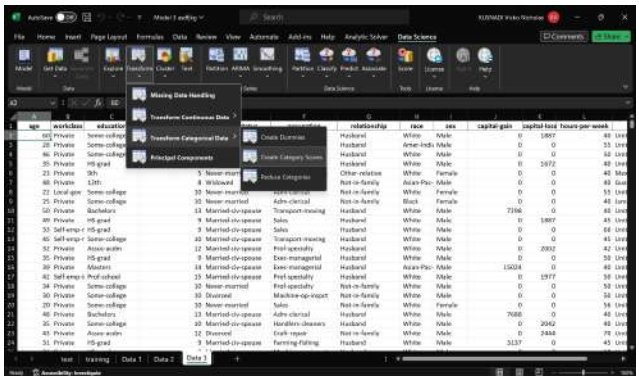


Fig 2.1.5

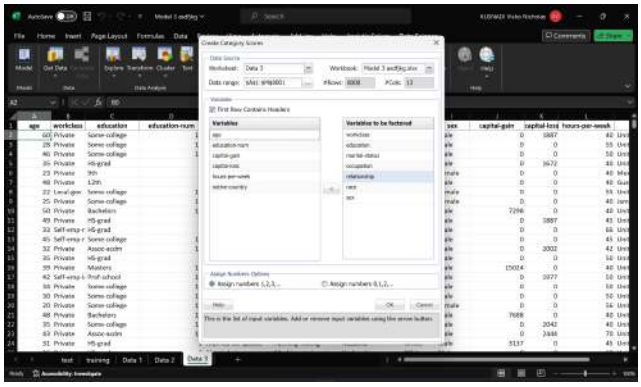


Fig 2.1.6

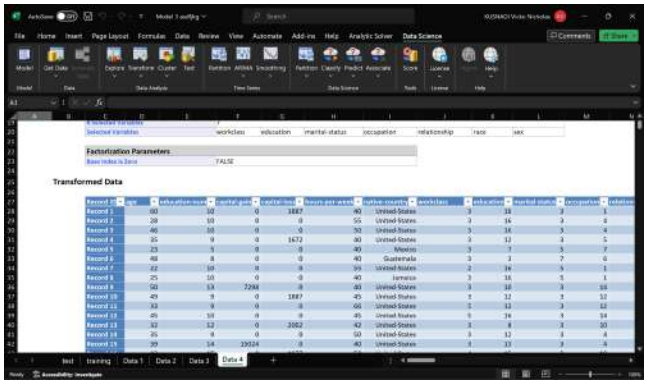


Fig 2.1.7

3. Model 1

- Data: Data 2
- Type of Model: Decision Tree
- Parameter used:
  - Success Class: >50K
  - Number of Classes: 2
  - Success probability cut-off: 0.5
  - Tree to display: Fully Grown
  - Records in Terminal Nodes: 1
  - Maximum Number of Leaves: 7
  - Set the Test set and Training set variables to Match by Name

The first model we are generating using Classification Tree algorithm. The parameters used for this model are setting the number of Classes to 2 by default, success probability cut-off to 0.5, set Tree to display to Fully Grown, set Records in Terminal Nodes to 1, and Maximum Number of Leaves: 7.In addition, we also set the Success Class to ">50K".

To generate, first select classify then classification tree in “Data 2” sheet (as shown in Fig. 3.0). Now we see the data tab, set the tab to “Data” tab, and set the data range to C27 until C8027, as we want to use the first 8000 entries as our training data (as shown on Fig 3.1). We set all the variables in “Variables in Input Data” to selected Variables except “Record ID”, “native-country”, and “Income”, set the “Ouput Variable” to “Income”. Finally, keep the success Probability Cutoff to 0.5, the Number of Classes to 2, and also the Success class to >50K.

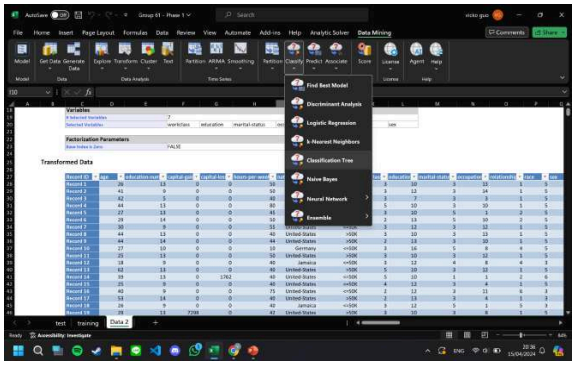


Fig. 3.0. Choose the classification Tree from XLMiner

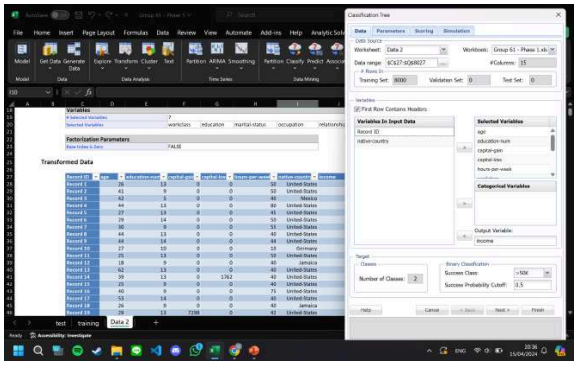


Fig. 3.1. Data tab. set the data range for the training set (C27:C8027)

In the Parameters Tab, we make changes to certain settings. We enable the inclusion of records in terminal nodes and set it to 1. We also limit the maximum number of levels to 7, which is the maximum allowed. Clicking on Trees to Display allows us to choose the Fully Grown option for displaying the complete classification tree. The settings can be seen on Fig.3.2 and Fig. 3.3.

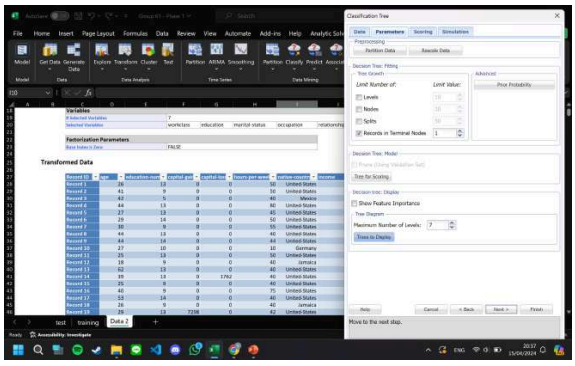


Fig 3.2. Parameters Tab

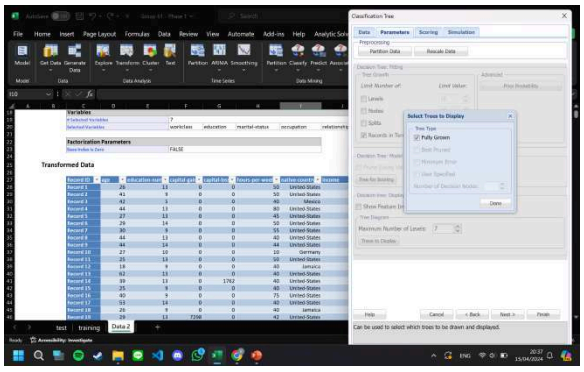


Fig 3.3. Select Tree to Display type to Fully Grown

Moving to the Scoring Tab, we evaluate the performance of the model on the training data by selecting all the relevant checkboxes which are “Detailed Report”, “Summary Report”, “Lift Charts”, and “Frequency Chart” (as shown on Fig. 3.4). In the Score New Data section, we choose the option to score data within the same worksheet as the training set. This leads to the creation of a new tab called the New Data (WS) Tab, where we set up the test data. The test data range is specified as C8029 to Q10029, with a total of 2000 data points (as shown on Fig. 3.5). We use the “Match By Name” option to match variables with the training set and then click Finish to complete the setup of the test data.

The result obtained from XLMiner may show some limitations or unexpected outcomes, which require careful analysis.

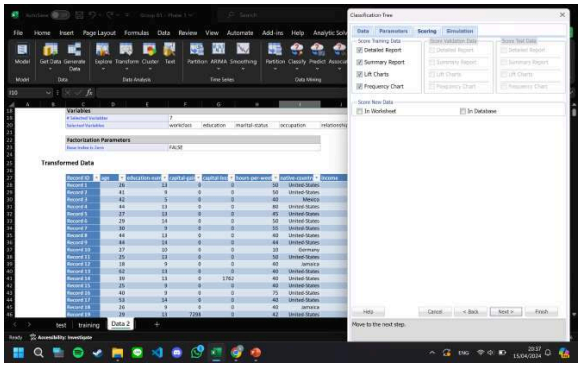


Fig. 3.4. Scoring tab

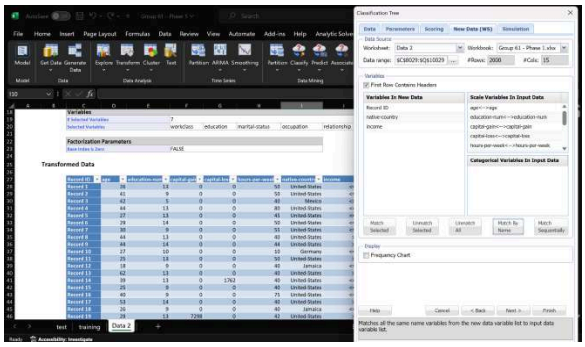


Fig. 3.5. New Data (WS) Tab. Set the data range for test set (C8029:C10029); Press “match by name” to match the training set attributes and test set attributes by name

4. Model 2

- Data: Data 1
- Type of Model: Naive Bayesian
- Parameter used:
  - Success Class: >50K
  - Number of Classes: 2
  - Success probability cut-off: 0.5
  - Laplace Smoothing: False
  - Prior Probability Method: Empirical
  - Show Prior Conditional Probability: True
  - Show Log Density: True
  - Set the Test set and Training set variables to Match by Name

In the second model, we will generate the training data using the Naive Bayesian algorithm. The parameter used for the second model is similar to the first model with a bit of modification. The parameters are set the Number of Classes to 2 and the success probability cut-off to 0.5. In addition, we also disable the Laplace smoothing, use the empirical method for the Prior Probability Method, Show Prior Conditional Probability, Show Log Density, and assign “>50K” as the Success Class.

The first step to generate the data is to select "Classify" and then "Naive Bayes" in the "Data 1" sheet (as shown in Fig. 4.0). In the data tab, set the data range to A1 until A8001, as we want to use the first 8000 entries as our training data (as shown in Fig 4.1). Then, set the success probability cutoff to 0.5, the number of classes to 2, and the success class to ">50K". Next, press the "Next" button.

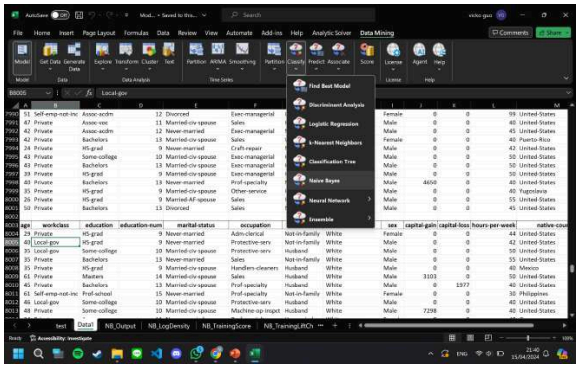


Fig.4.0. Naive Bayesian Classifier Selection

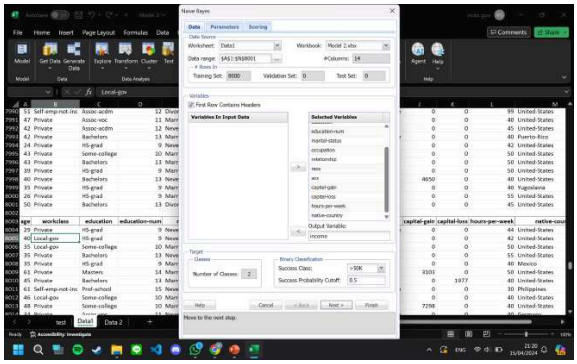


Fig.4.1. Data. TabSelected Variable and the Output Variable Selection and Data range selection from A1 to A8001

In the parameters tab, disable Laplace Smoothing and select "Prior Probability" (as shown in Fig 4.2). In the pop-up, select the "Empirical" option for the Prior Probability Method (as shown in Fig 4.3). In the "Display Options", check all the boxes in the "Naive Bayes: Display" section, and then press the "Next" button again to move to the "Scoring Tab" (as shown in Fig 4.4).

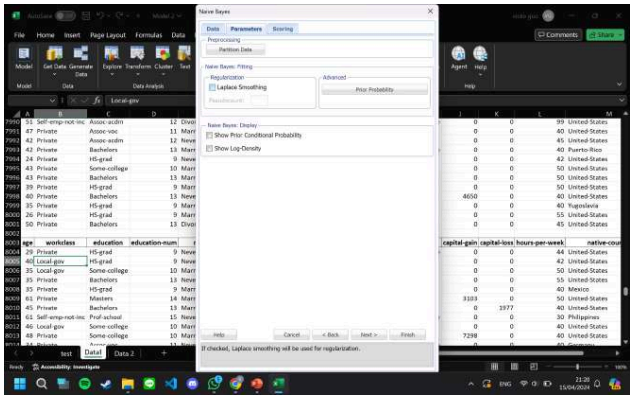


Fig.4.2. Parameter Tabs

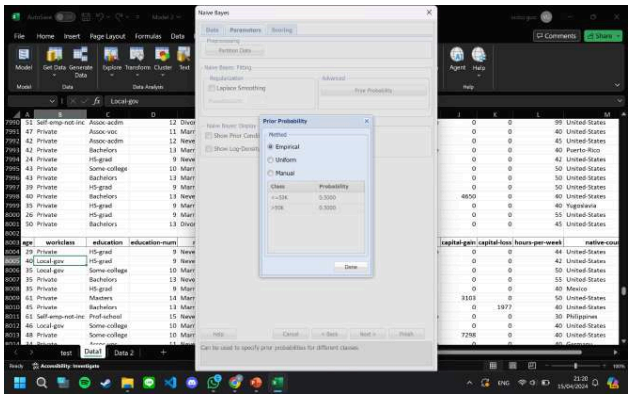


Fig.4.2.Prior Probability Tab

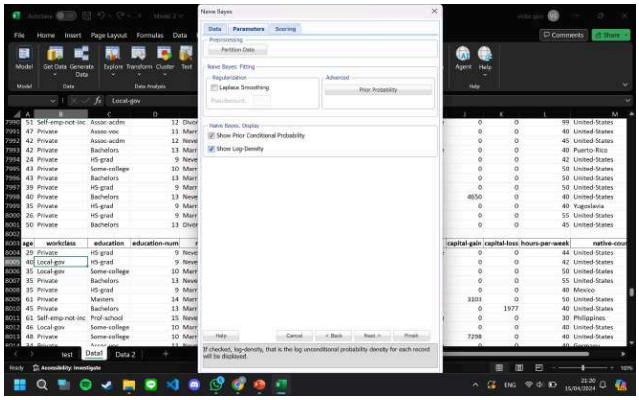


Fig.4.4. Parameter Tabs

In the "Scoring Tab", check all the boxes in the "Score training data" section, namely "Detailed Report", "Summary Report", "Lift Charts", and "Frequency Chart" (as shown in Fig 4.5). In addition, in the "Score New Data" area, select the "In Worksheet" option. After selecting the "In Worksheet" option, the "New Data (WS)" tab will appear (as shown in Fig 4.6).

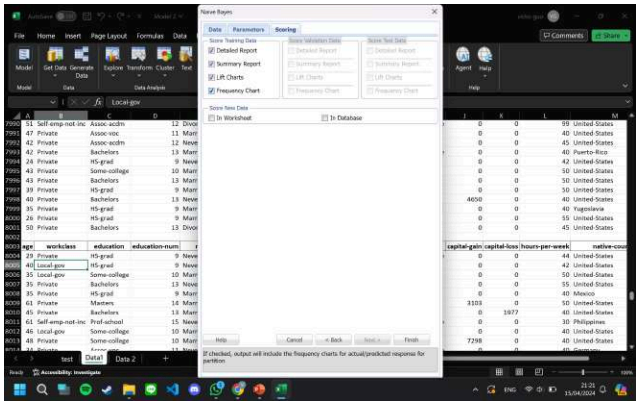


Fig.4.5. Scoring Tab

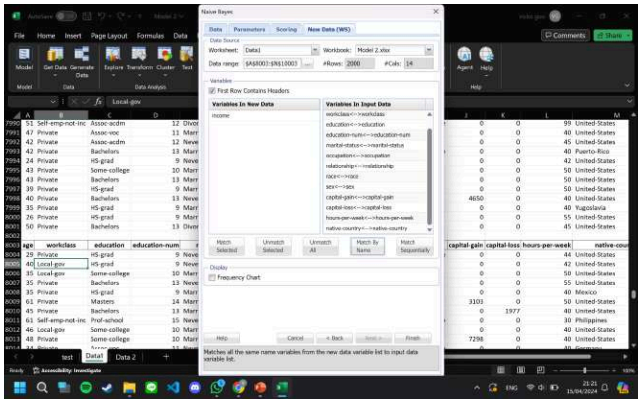


Fig.4.6. New Data(WS) Tab. Set the data range for the test set (A8003:N10003); Press "match by name" to match the training set attributes and test set attributes by name

The "New Data (WS)" tab is used to set the test set. Set the data range from A8003 to N10003, as we are using 2000 data points for testing. Finally, click the 'Match By Name' button to match the attributes of the training data with the test data set, and click 'Finish' to generate the results by XLMiner (as shown in Fig 4.6).

5. Model 3

- Data: Data 2
- Type of Model: K Nearest Neighbor
- Parameter used:
  - Success Class: >50K
  - Number of Classes: 2
  - Success probability cut-off: 0.5
  - Fixed K with K = 3
  - Prior Probability to Empirical
  - Set the Test set and Training set variables to Match by Name

Model 3, based on K Nearest Neighbor, was applied to the Data 2 dataset with specific parameters. The analysis involved classifying the data into two classes based on the "Income" variable, with the success class defined as ">50K." A success probability cutoff of 0.5 was used to determine the class assignment. The K Nearest Neighbor algorithm was configured with a fixed value of K=3. Prior probability estimation was performed using the empirical method.

In the Data 2 Sheet, select Data mining tab on Excel, and then select Classify, and choose K Nearest Neighbor as the Algorithm type for this model (as shown in Fig 5.0). In the Data Tab, set the data range from C27 to C8029, since 8000 data points are for training sets. All variables except "Record ID," "native-country," and "Income" were selected as the Selected Variables. The "Income" variable was designated as the Output Variable (as shown in Fig 5.1). After that, press the Next button to go to the Parameter Tab

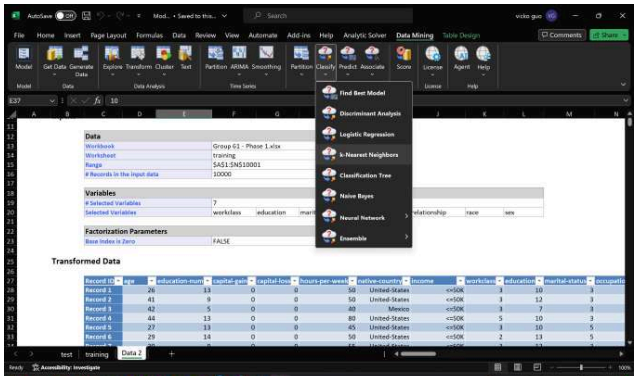


Fig.5.0. K Nearest Neighbor Classifier Selection

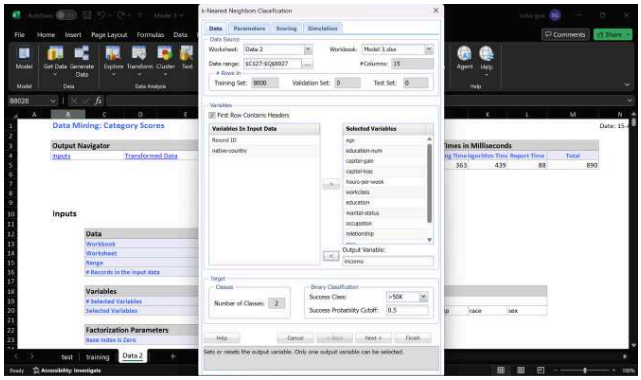


Fig.5.1. Data Tab. Selected Variable and the Output Variable Selection and Data range selection from C27 to C8027

Moving to the Parameters Tab, the K Nearest Neighbor algorithm was configured with a value of K=3, indicating that the three nearest neighbors were considered for classification. The nearest neighbor search method was set to "Fixed K.". After that, press the prior probability button, and the Prior Probability Tab will appear, set the Prior Probability Empirical. Then, press the next button again to move to the Scoring tab

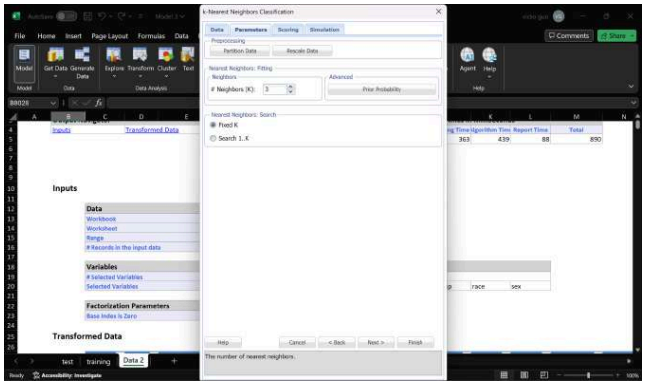


Fig.5.2. Parameter Tabs

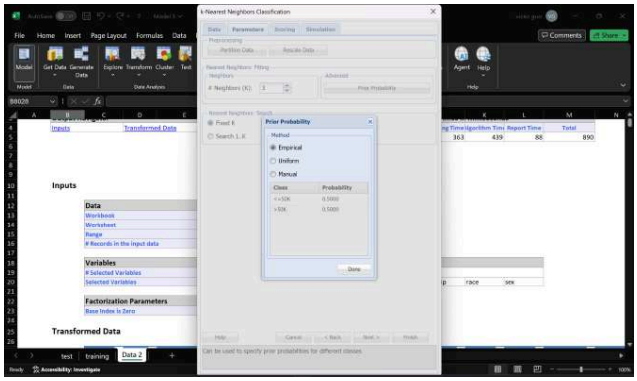


Fig.5.3.Prior Probability Tab

In the Scoring Tab, all the checkboxes in the Score Training data section were selected to evaluate the model's performance on the training set (as shown in Fig.5.4). Next, in the scoring new data section, enable the In Worksheet option, to indicate the test data was present in the same worksheet as the training set.

After enabling the In Worksheet option, a new tab w, the New Data (WS) Tab, will appear to set up the test data (as shown in Fig.5.5). The test data range was specified as C8029 to Q10029, comprising

2000 data points. The variables in the test data were matched with the training set using the Match By Name function. Upon completion of the test data setup, the analysis was conducted. Finally, press the Finish button, and the result will be generated by XLMiner.

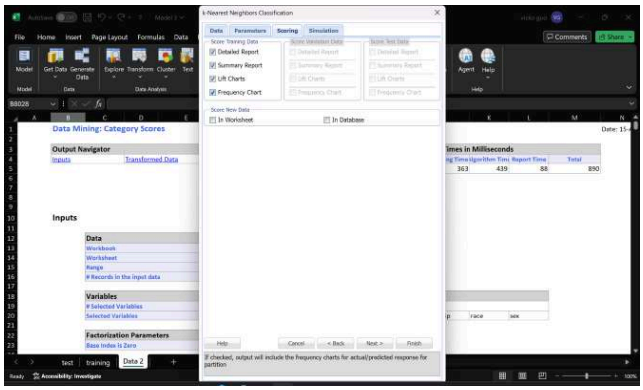


Fig.5.4. Scoring Tabs

Fig.5.5. New Data (WS) Tab. Set the data range for the test set (C8029:C10029); Press "match by name" to match the training set attributes and test set attributes by name

6. Model 4

- Data: Data 2
- Type of Model: K Nearest Neighbor
- Parameter used:
  - Success Class: >50K
  - Number of Classes: 2
  - Success probability cut-off: 0.5
  - Fixed K with K = 5
  - Prior Probability to Empirical
  - Set the Test set and Training set variables to Match by Name

Similar with Model 3, in Model 4, we use the K Nearest Neighbor and Data 2 dataset with different parameters from Model 3. The parameter used for this Model is classifying the data into two classes based on the "Income" variable, with the success class defined as ">50K". and also a success probability cutoff of 0.5 to determine the class assignment. In addition, The K Nearest Neighbor algorithm was configured with a fixed value of K=5 and the prior probability estimation was performed using the empirical method.

In the Data 2 Sheet, select the Data mining tab on Excel, and then select Classify, and choose K Nearest Neighbor as the Algorithm type for this model (as shown in Fig 5.0). In the Data Tab, set the data range from C27 to C8029, since 8000 data points are for training sets. All variables except "Record ID," "native-country," and "Income" were selected as the Selected Variables. The "Income" variable was designated as the Output Variable (as shown in Fig 5.1). After that, press the Next button to go to the Parameter Tab

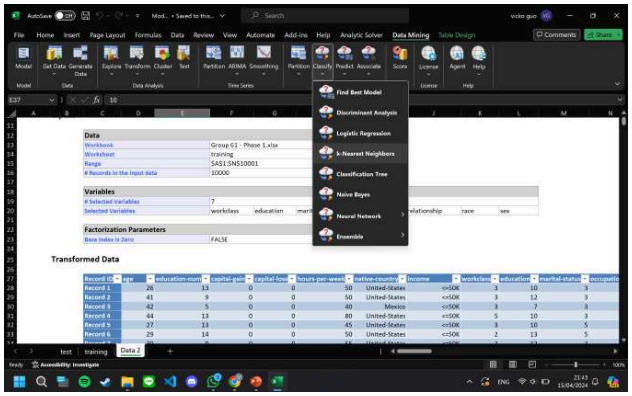


Fig.6.0. K Nearest Neighbor Classifier Selection

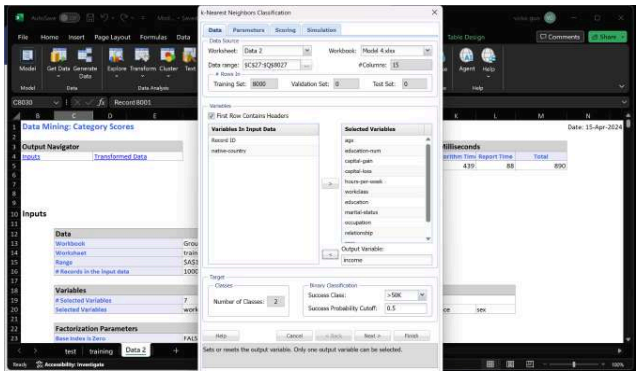


Fig.6.1. Data Tab. Selected Variable and the Output Variable Selection and Data range selection from C27 to C8027

Moving to the Parameters Tab, the K Nearest Neighbor algorithm was configured with a value of K=5, indicating that the three nearest neighbors were considered for classification. The nearest neighbor search method was set to "Fixed K.". After that, press the prior probability button, and the Prior Probability Tab will appear, set the Prior Probability Empirical. Then, press the next button again to move to the Scoring tab

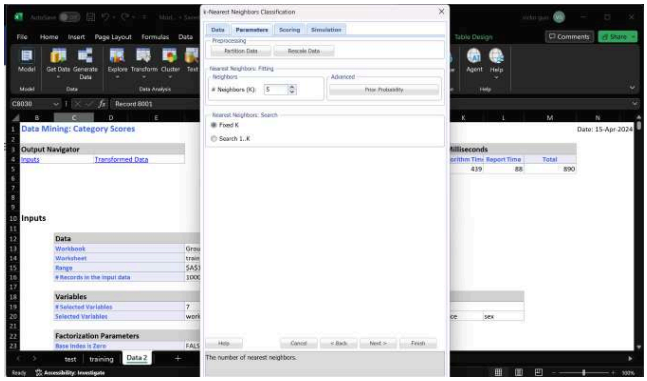


Fig.6.2. Parameter Tabs

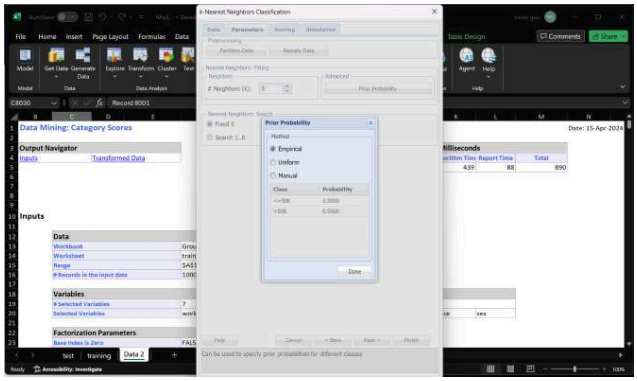


Fig.6.3. Prior Probability Tab

In the Scoring Tab, all the checkboxes in the Score Training data section were selected to evaluate the model's performance on the training set (as shown in Fig.5.4). Next, in the scoring new data section, enable the In Worksheet option, to indicate the test data was present in the same worksheet as the training set.

After enabling the In Worksheet option, a new tab w, the New Data (WS) Tab, will appear to set up the test data (as shown in Fig.5.5). The test data range was specified as C8029 to Q10029, comprising 2000 data points. The variables in the test data were matched with the training set using the Match By Name function. Upon completion of the test data setup, the analysis was conducted. Finally, press the Finish button, and the result will be generated by XLMiner.

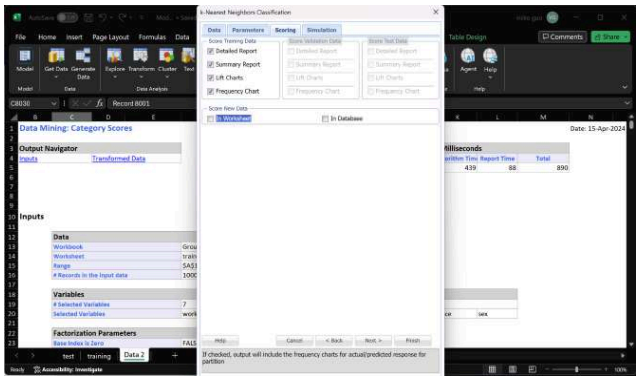


Fig.6.4. Scoring Tabs

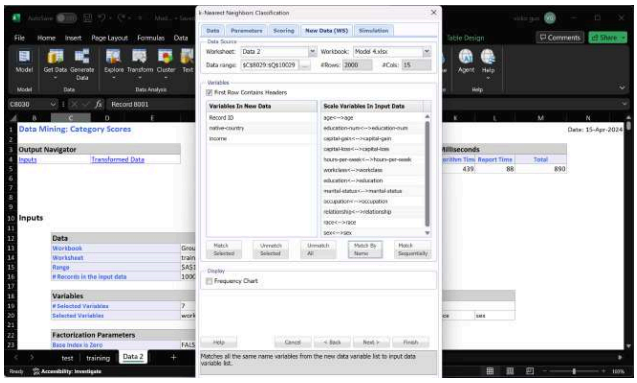


Fig.6.5. New Data (WS) Tab. Set the data range for the test set (C8029:C10029); Press "match by name" to match the training set attributes and test set attributes by name

7. Model 5

- Data: Data 2
- Type of Model: K Nearest Neighbor
- Parameter used:
  - Success Class: >50K
  - Number of Classes: 2
  - Success probability cut-off: 0.5
  - Fixed K with K = 10
  - Prior Probability to Empirical
  - Set the Test set and Training set variables to Match by Name

For Model 5, we are using “Data 2” and the model is K nearest neighbor, similar to models 3 and 4. We will be changing the parameters to hopefully achieve more accurate results. The parameter used for this Model is classifying the data into two classes based on the "Income" variable, with the success class defined as ">50K". We open the sheet and select “Predict” then select “K Nearest Neighbour” (as shown on Fig. 7.0). The success probability cutoff is set to 0.5 to determine the class assignment. We now set the K Nearest Neighbor algorithm which is configured with a fixed value of K=10 and the prior probability estimation was performed using the empirical method.

In the Data 2 Sheet, select Data mining tab on Excel, and then select Classify, and choose K Nearest Neighbor as the Algorithm type for this model (as shown in Fig 7.1). In the Data Tab, set the data range from C27 to C8029, since 8000 data points are for training sets. All variables except "Record ID," "native-country," and "Income" were selected as the Selected Variables. The "Income" variable was designated as the Output Variable (as shown in Fig 7.1). After that, press the Next button to go to the Parameter Tab

Moving to the Parameters Tab, the K Nearest Neighbor algorithm was configured with a value of K=5, indicating that the three nearest neighbors were considered for classification. The nearest neighbor search method was set to "Fixed K."(as shown in Fig 7.2). After that, press the prior probability button, and the Prior Probability Tab will appear, set the Prior Probability Empirical(as shown in Fig 7.3). Then, press the next button again to move to the Scoring tab

In the Scoring Tab, all the checkboxes in the Score Training data section were selected to evaluate the model's performance on the training set (as shown in Fig.7.4). Next, in the scoring new data

section, enable the In Worksheet option, to indicate the test data was present in the same worksheet as the training set.

After enabling the In Worksheet option, a new tab w, the New Data (WS) Tab, will appear to set up the test data (as shown in Fig.7.5). The test data range was specified as C8029 to Q10029, comprising 2000 data points. The variables in the test data were matched with the training set using the Match By Name function. Upon completion of the test data setup, the analysis was conducted. Finally, press the Finish button, and the result will be generated by XLMiner.

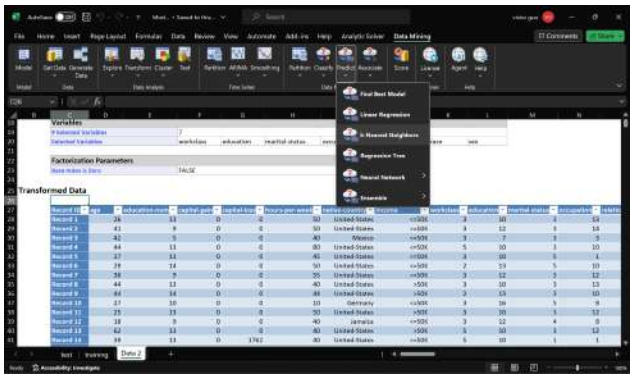


Fig.7.0. K Nearest Neighbor Classifier Selection

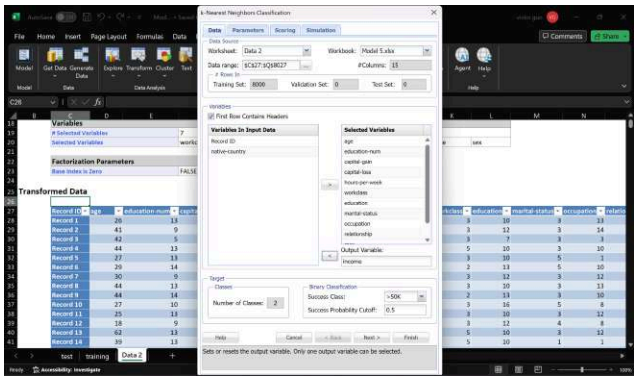


Fig.7.1. Data Tab. Selected Variable and the Output Variable Selection and Data range selection from C27 to C8027

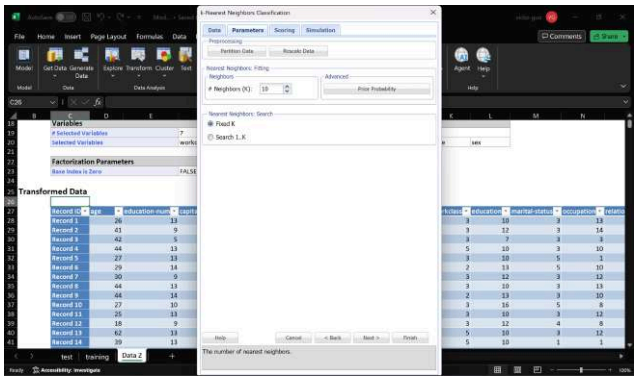


Fig.7.2. Parameter Tabs

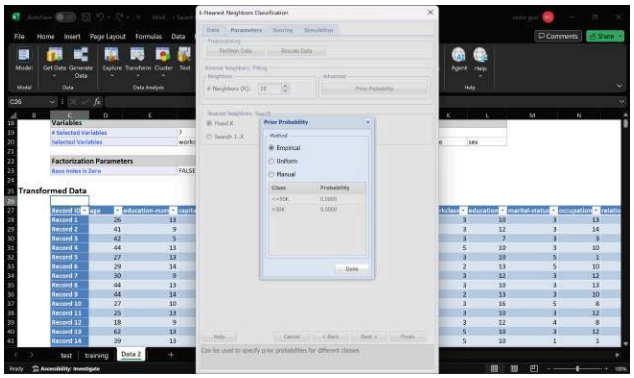


Fig.7.3. Prior Probability Tab

Fig.7.4. Scoring Tabs

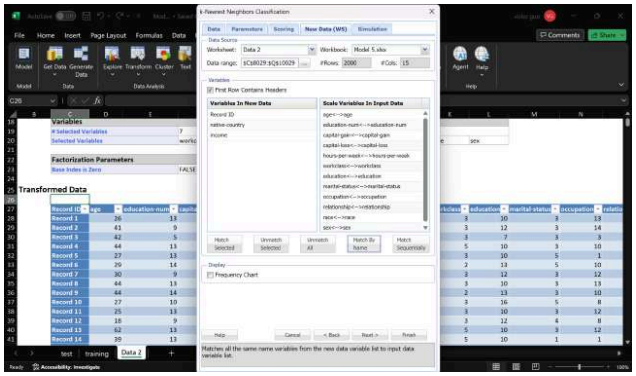
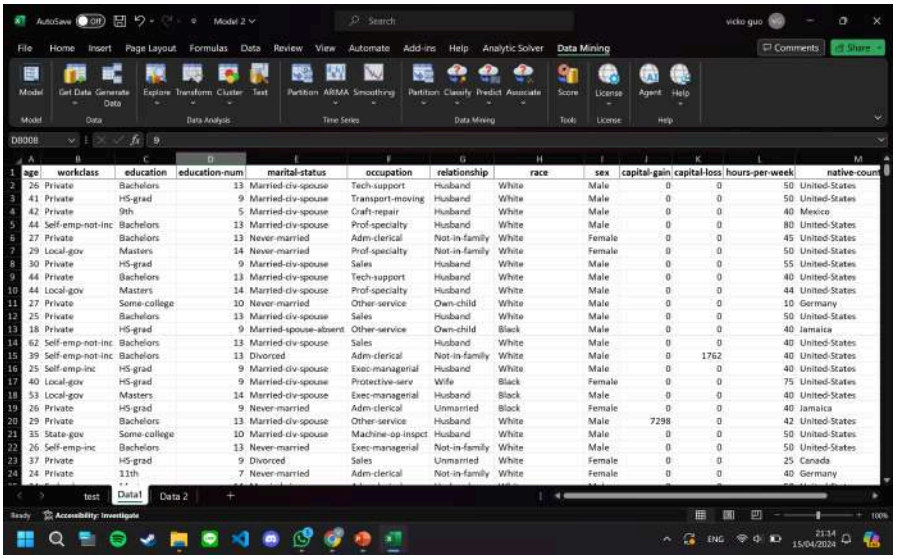


Fig.7.5. New Data (WS) Tab. Set the data range for the test set (C8029:C10029); Press "match by name" to match the training set attributes and test set attributes by name

8. Images for Data Model



age	workclass	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country
26	Private	Bachelors	13	Married-civ-spouse	Tech-support	Husband	White	Male	0	0	50	United-States
41	Private	HS-grad	9	Married-civ-spouse	Transport-moving	Husband	White	Male	0	0	50	United-States
42	Private	9th	5	Married-civ-spouse	Craft-repair	Husband	White	Male	0	0	40	Mexico
44	Self-emp-not-inc	Bachelors	13	Married-civ-spouse	Prof-specialty	Husband	White	Male	0	0	80	United-States
27	Private	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Female	0	0	45	United-States
29	Local-gov	Masters	14	Never-married	Prof-specialty	Not-in-family	White	Female	0	0	50	United-States
30	Private	HS-grad	9	Married-civ-spouse	Sales	Husband	White	Male	0	0	55	United-States
44	Private	Bachelors	13	Married-civ-spouse	Tech-support	Husband	White	Male	0	0	40	United-States
44	Local-gov	Masters	14	Married-civ-spouse	Prof-specialty	Husband	White	Male	0	0	44	United-States
27	Private	Some-college	10	Never-married	Other-service	Own-child	White	Male	0	0	10	Germany
25	Private	Bachelors	13	Married-civ-spouse	Sales	Husband	White	Male	0	0	50	United-States
18	Private	HS-grad	9	Married-spouse-absent	Other-service	Own-child	Black	Male	0	0	40	Jamaica
62	Self-emp-not-inc	Bachelors	13	Married-civ-spouse	Sales	Husband	White	Male	0	0	40	United-States
39	Self-emp-not-inc	Bachelors	13	Divorced	Adm-clerical	Not-in-family	White	Male	0	1762	40	United-States
25	Self-emp-inc	HS-grad	9	Married-civ-spouse	Exec-managerial	Husband	White	Male	0	0	40	United-States
40	Local-gov	HS-grad	9	Married-civ-spouse	Protective-serv	Wife	Black	Female	0	0	75	United-States
53	Local-gov	Masters	14	Married-civ-spouse	Exec-managerial	Husband	Black	Male	0	0	40	United-States
26	Private	HS-grad	9	Never-married	Adm-clerical	Unmarried	Black	Female	0	0	40	Jamaica
29	Private	Bachelors	13	Married-civ-spouse	Other-service	Husband	White	Male	7298	0	42	United-States
35	State-gov	Some-college	10	Married-civ-spouse	Machine-op-inspct	Husband	White	Male	0	0	50	United-States
26	Self-emp-inc	Bachelors	13	Never-married	Exec-managerial	Not-in-family	White	Male	0	0	50	United-States
37	Private	HS-grad	9	Divorced	Sales	Unmarried	White	Female	0	0	25	Canada
24	Private	11th	7	Never-married	Adm-clerical	Not-in-family	White	Female	0	0	40	Germany

Fig 1.1

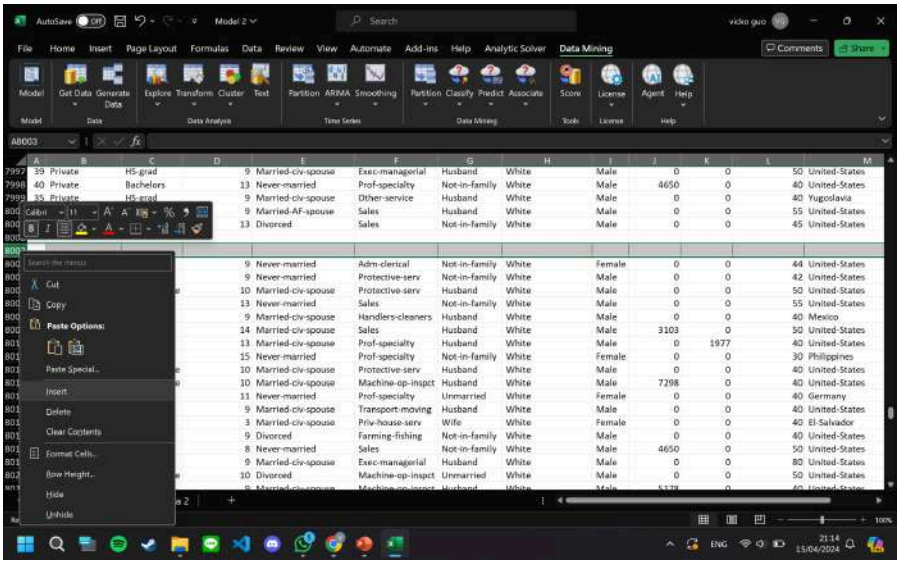


Fig 1.0

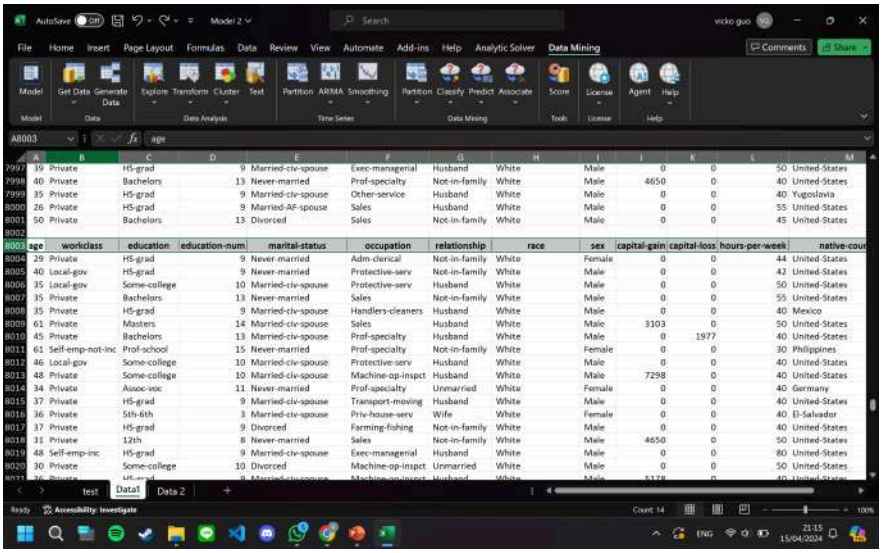


Fig 1.2

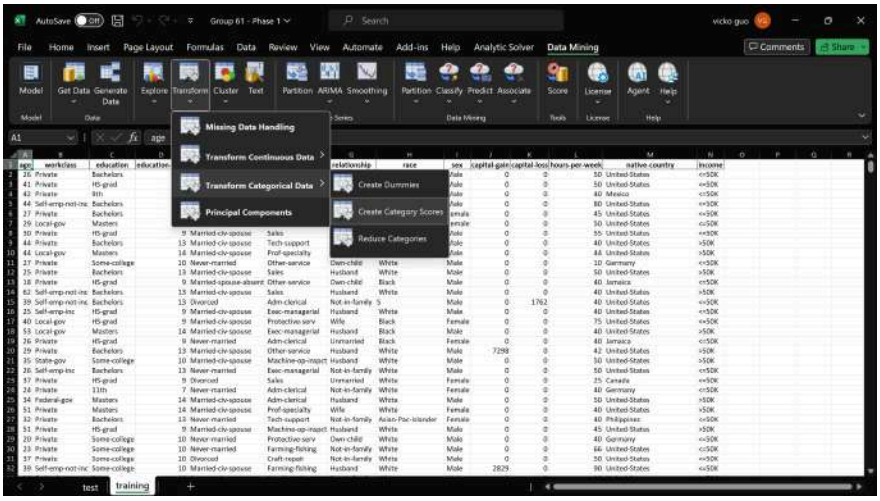


Fig. 2.0.0

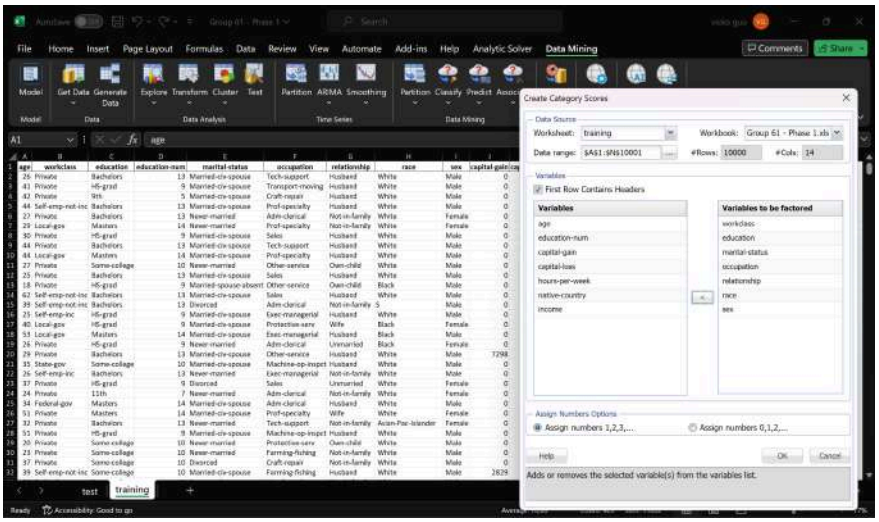


Fig. 2.0.1

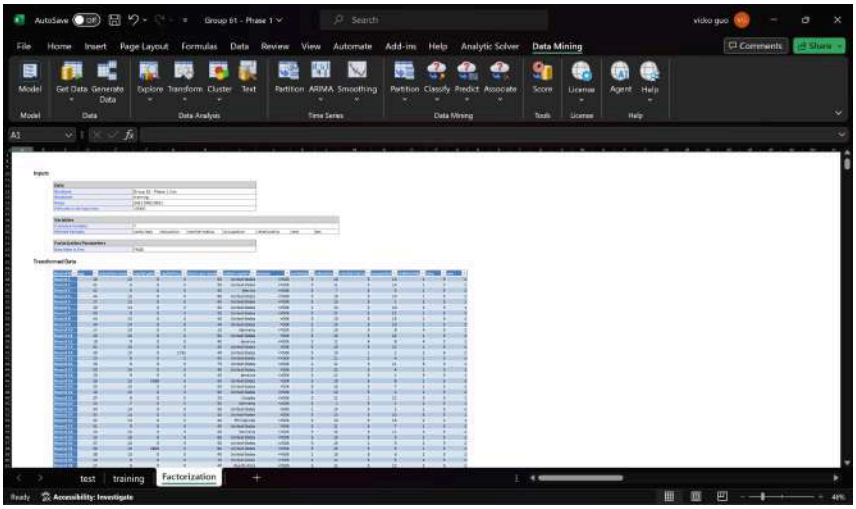


Fig. 2.0.2. Resulting numerical data from transformation

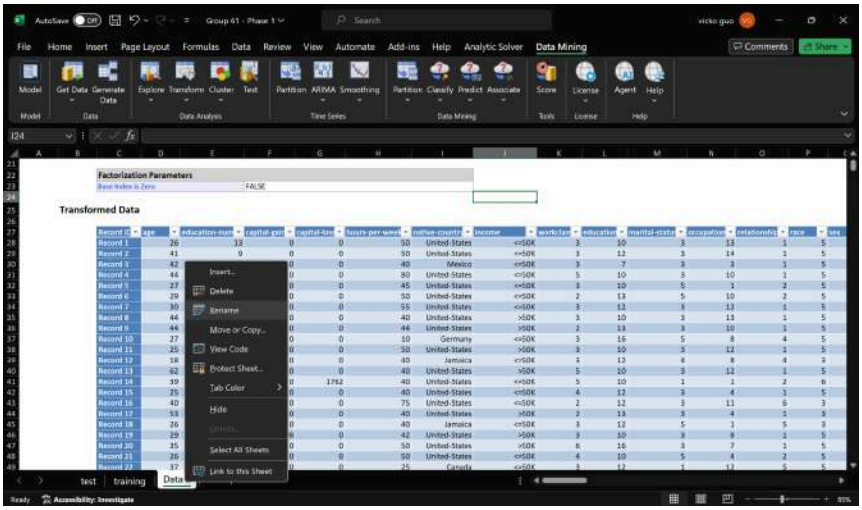


Fig. 2.1.1

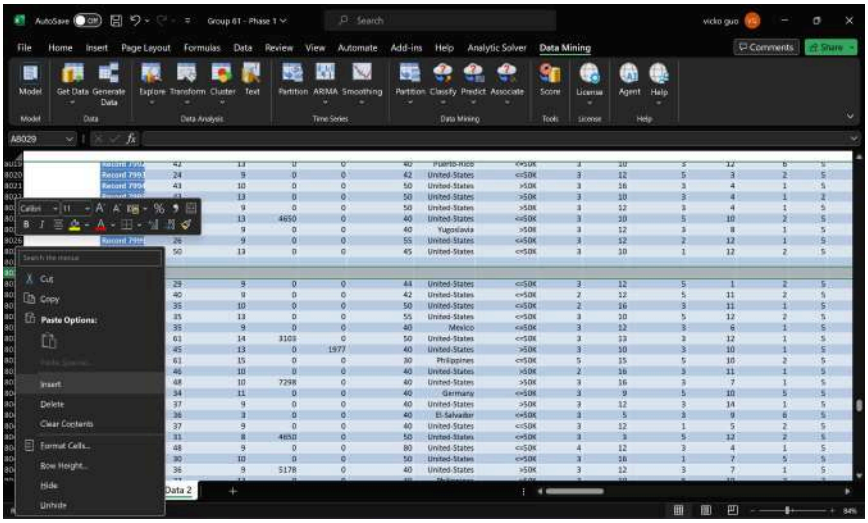


Fig. 2.1.0. Rename the data as "Data 2"

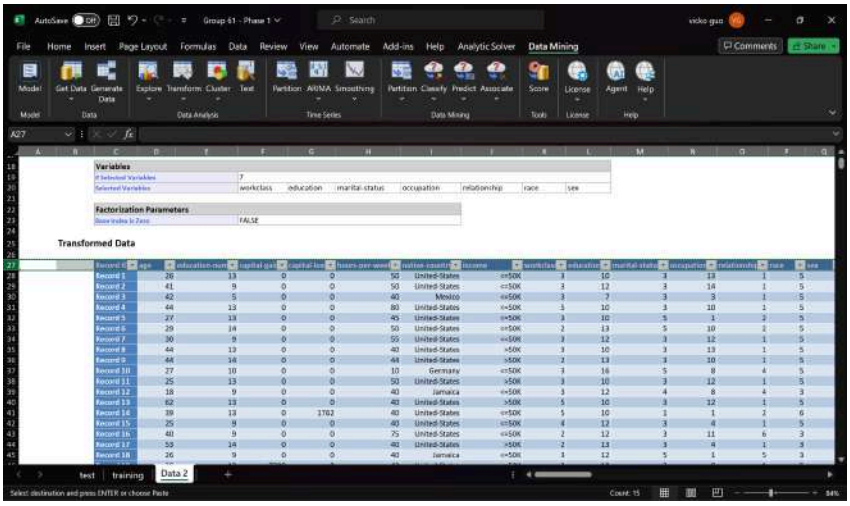


Fig. 2.1.2

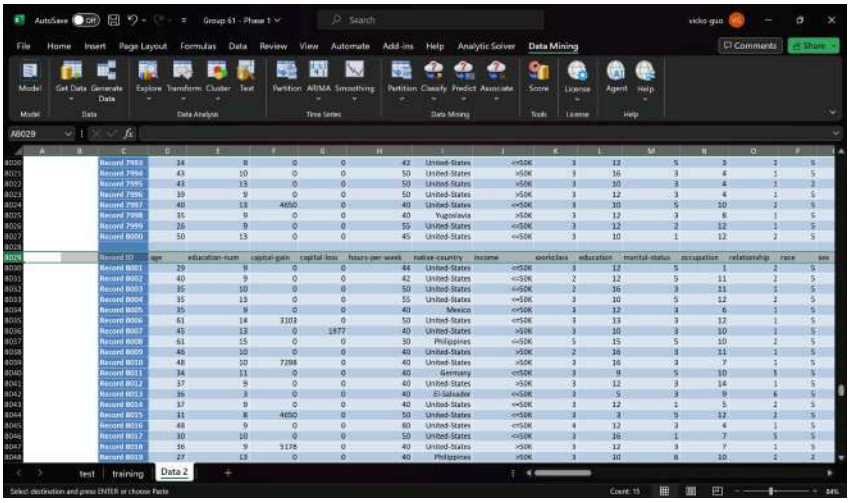


Fig 2.1.3

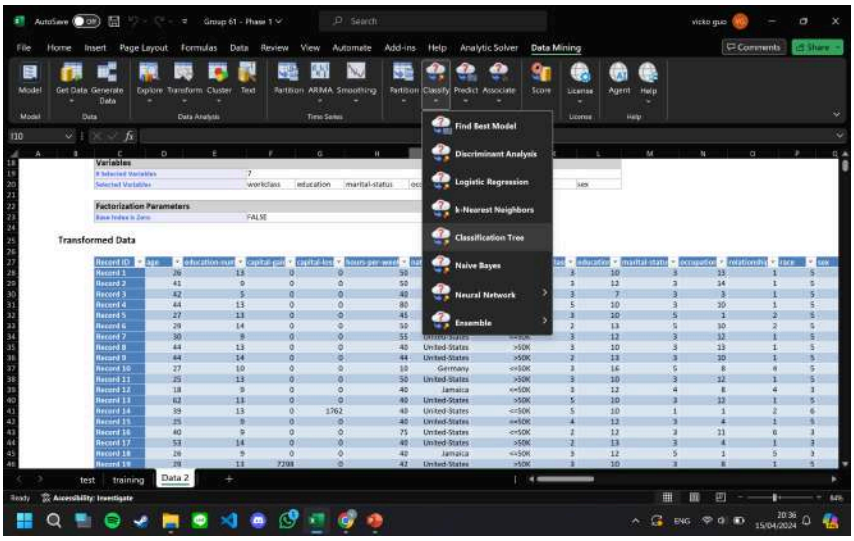


Fig. 3.0

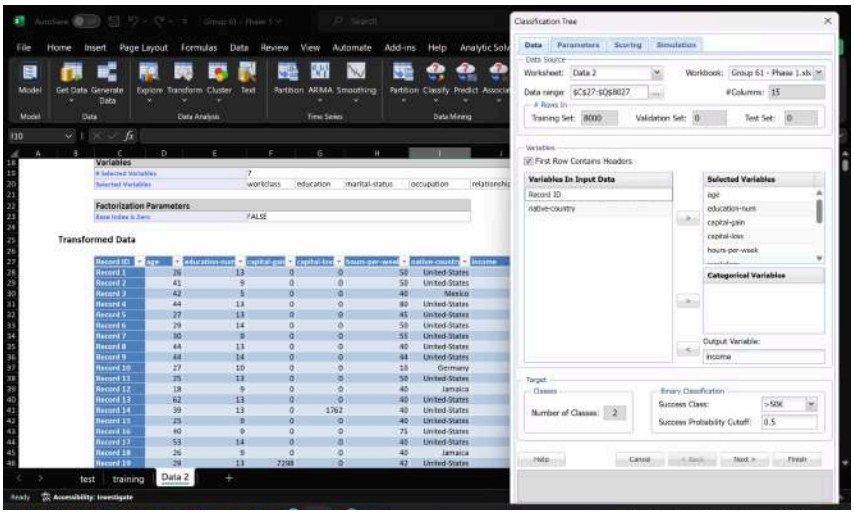


Fig. 3.1

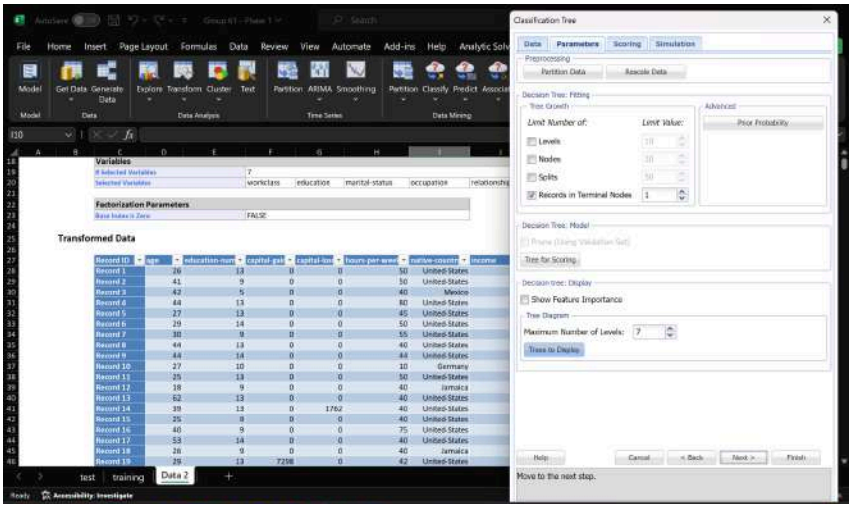


Fig. 3.2. Parameters Tab

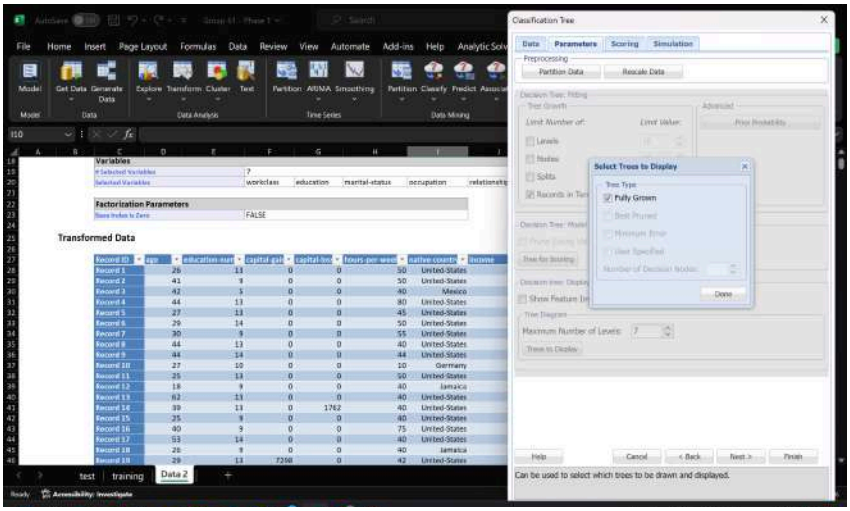


Fig. 3.3. Select Tree to Display type to Fully Grown

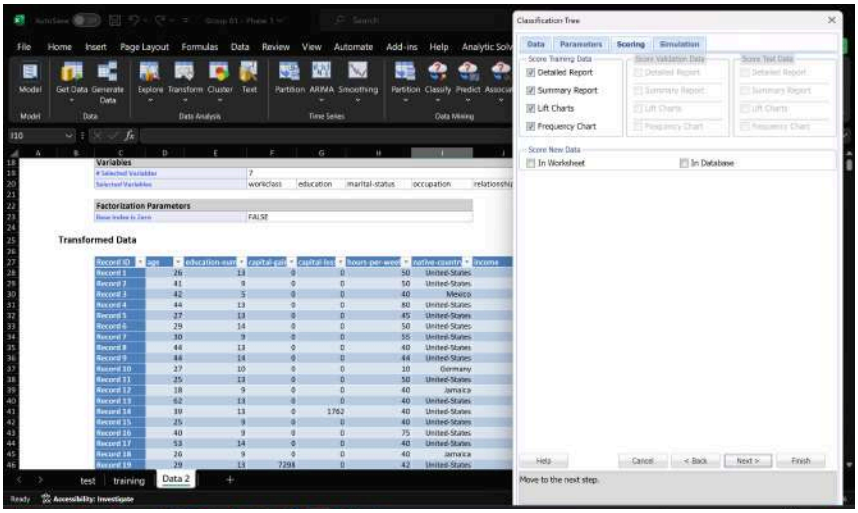


Fig. 3.4. Scoring tab

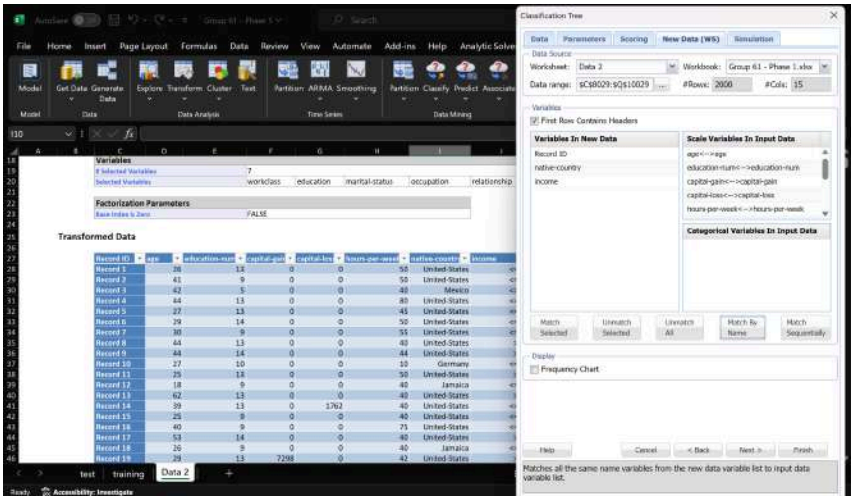


Fig. 3.5

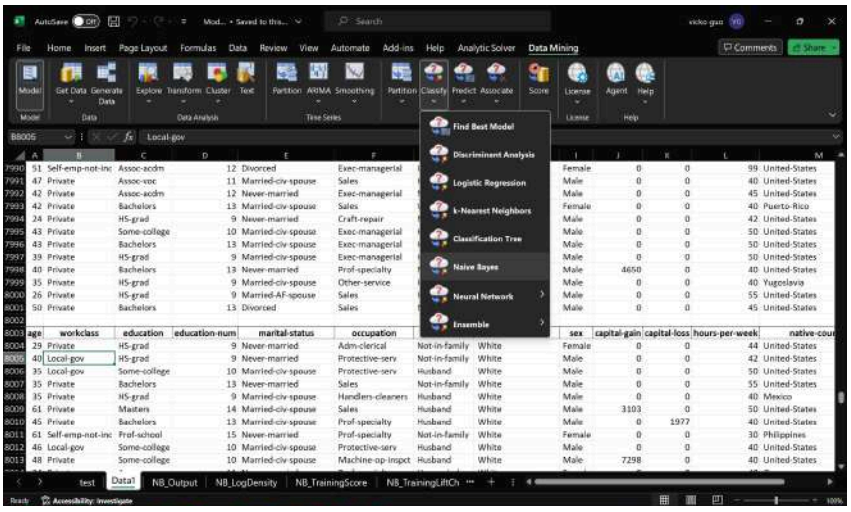


Fig. 4.0. Naive Bayesian Classifier Selection

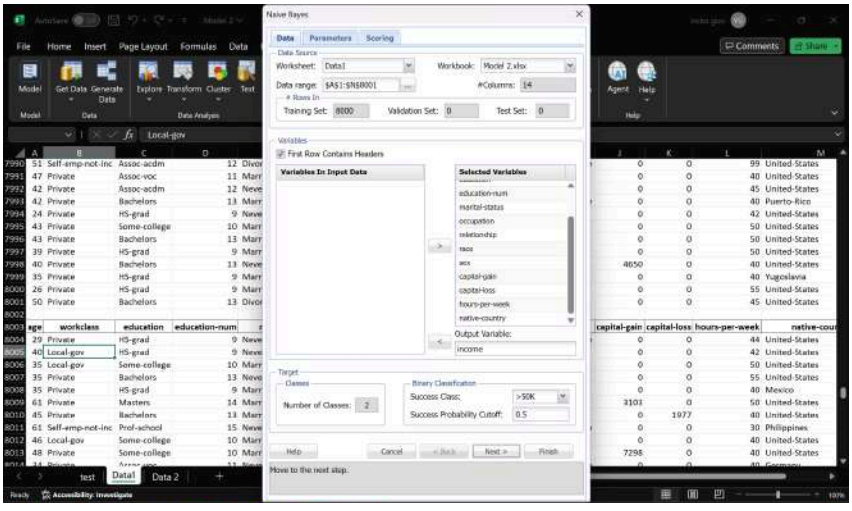


Fig. 4.1. Data TabSelected Variable and the Output Variable

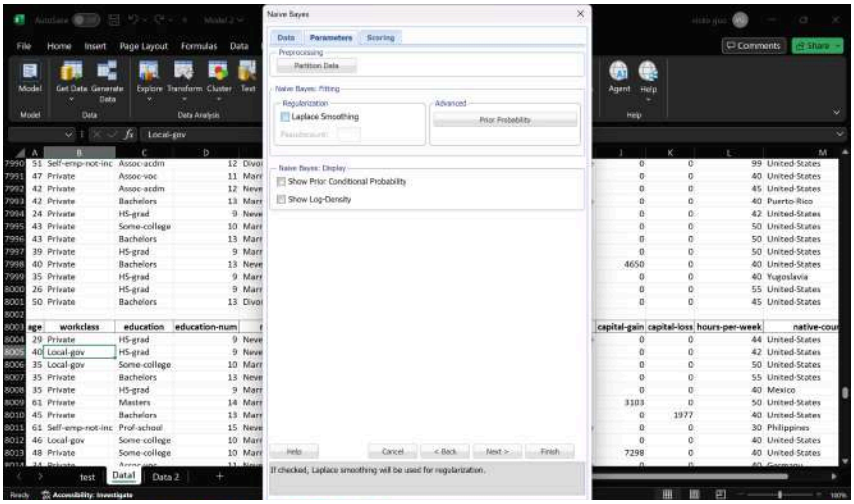


Fig. 4.2. Parameter Tabs

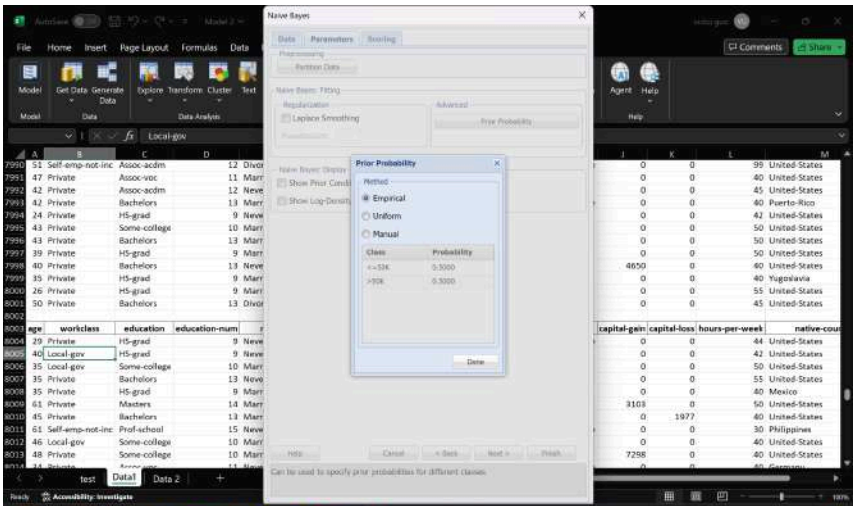


Fig. 4.2. Prior Probability Tab

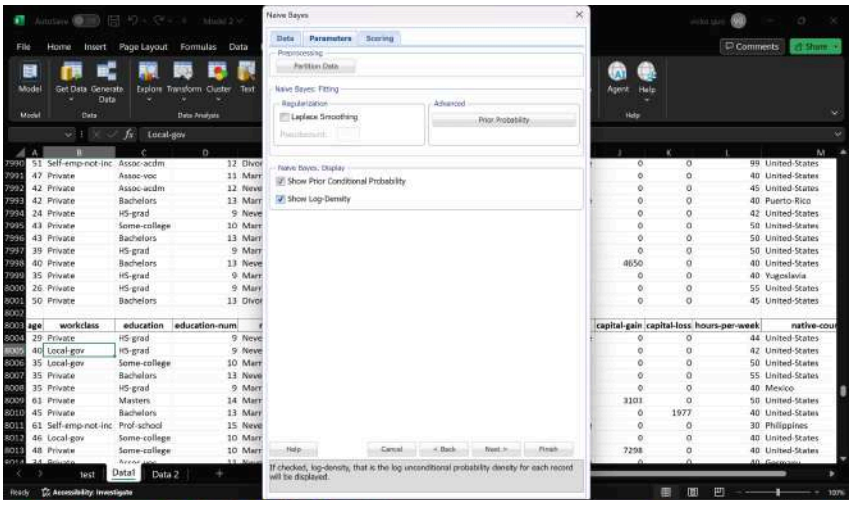


Fig. 4.4. Parameter Tabs

Fig. 4.5. Scoring Tab

Fig. 4.6. New Data(WS) Tab. Set the data range for the test set

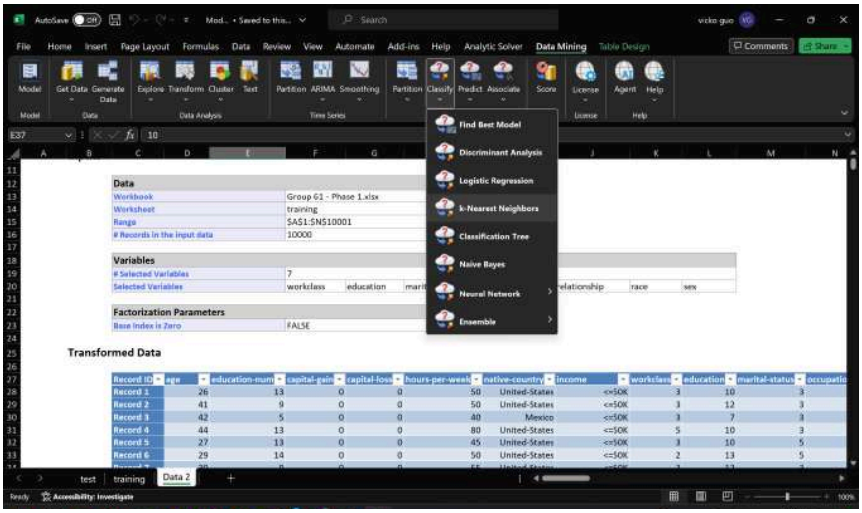


Fig. 5.0. K Nearest Neighbor Classifier Selection

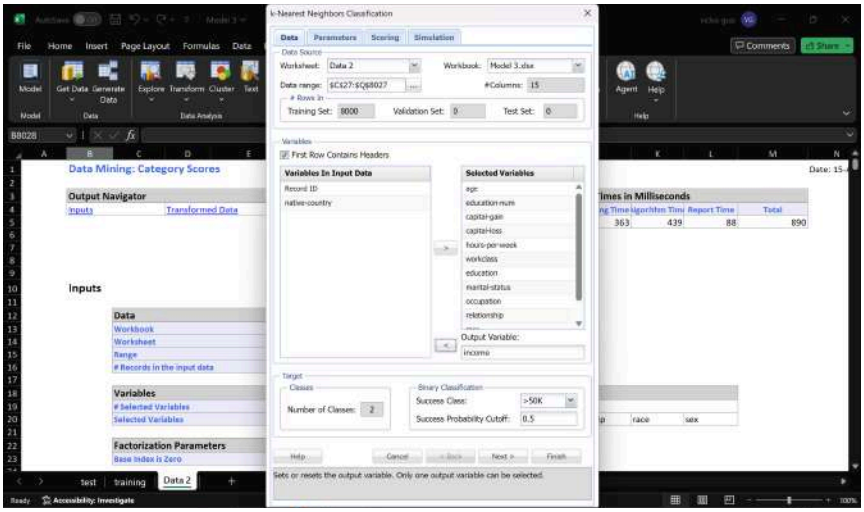


Fig. 5.1. Data Tab. Selected Variable and the Output Variable

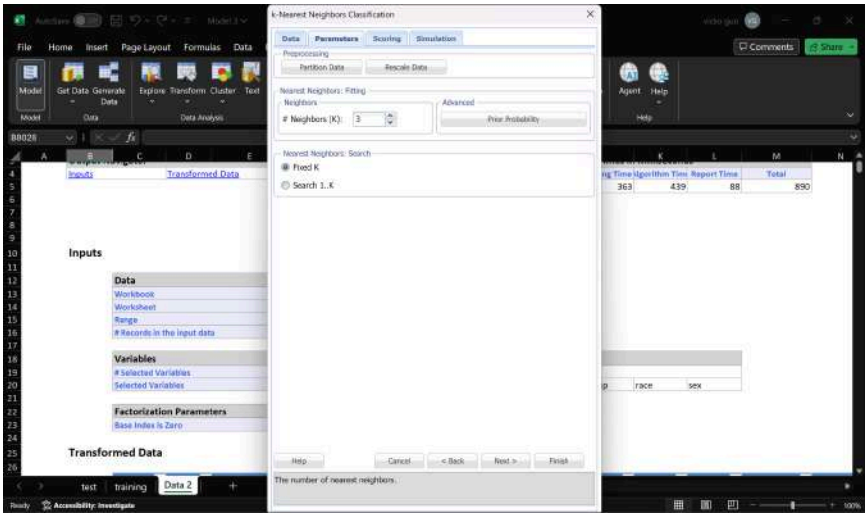


Fig. 5.2. Parameter Tabs

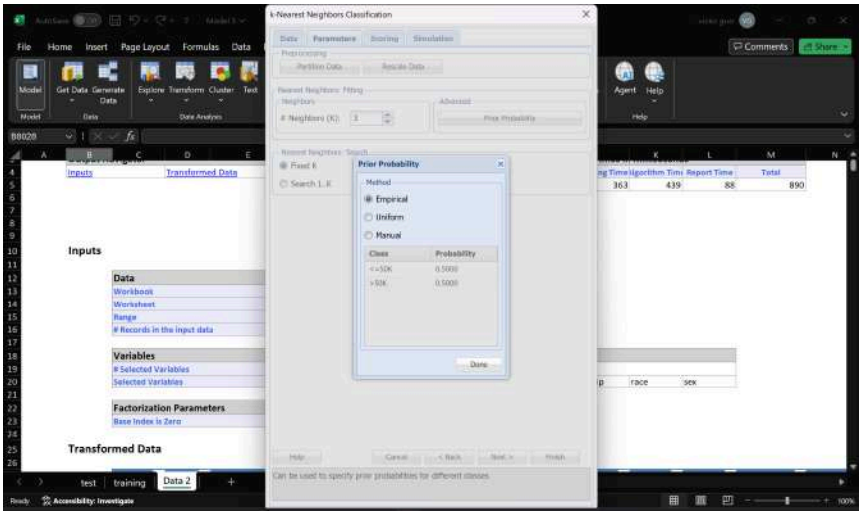


Fig. 5.3.Prior Probability Tab

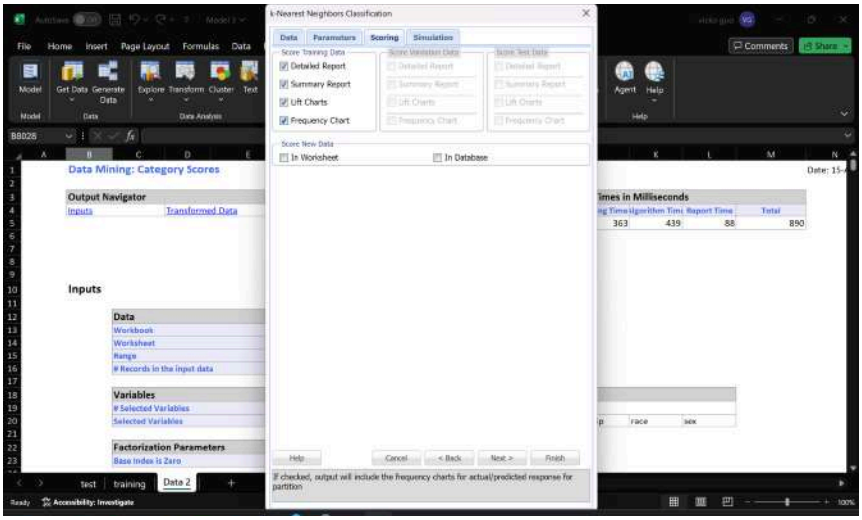


Fig. 5.4. Scoring Tabs

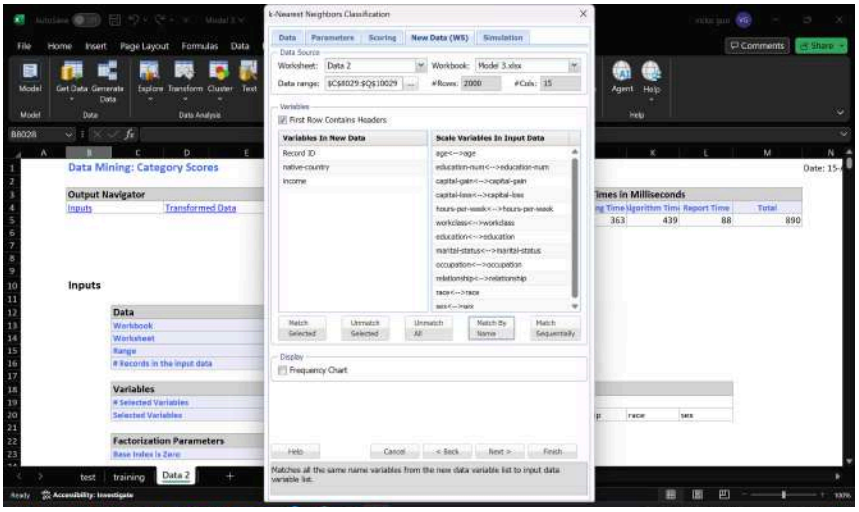


Fig. 5.5. New Data (WS) Tab.

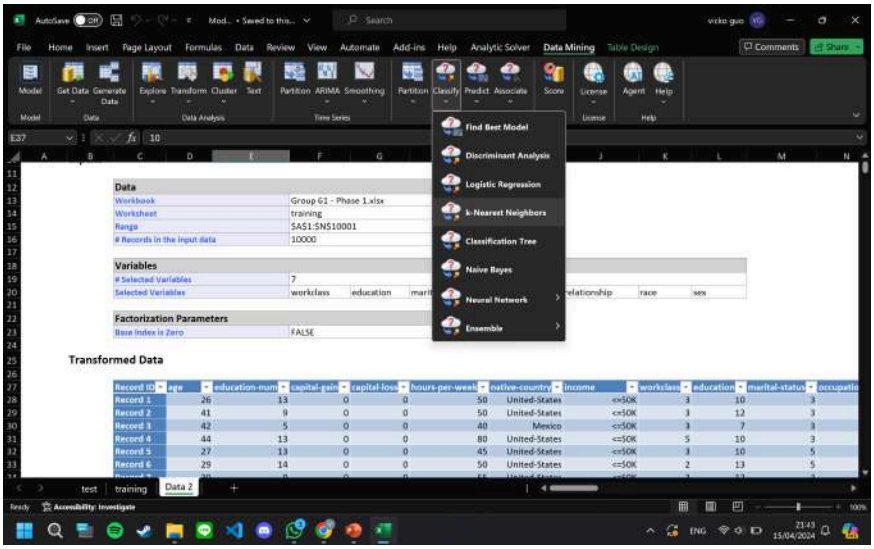


Fig. 6.0. K Nearest Neighbor Classifier Selection

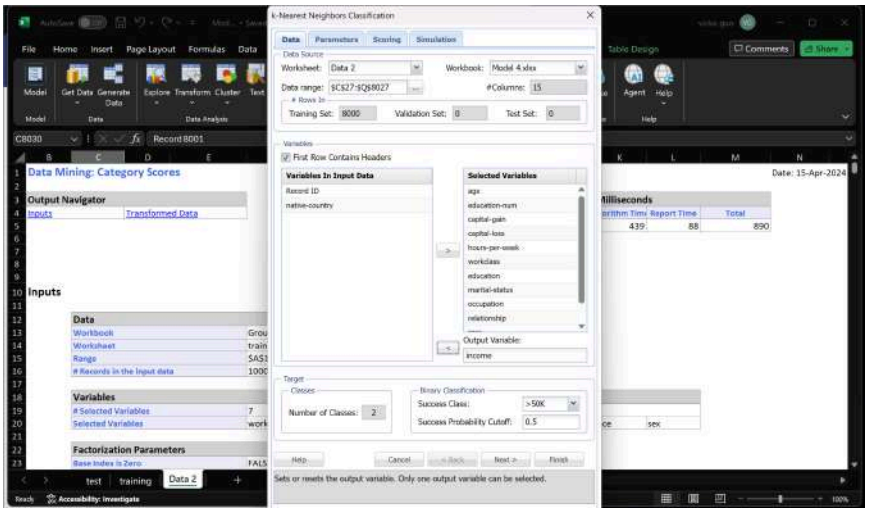


Fig. 6.1. Data Tab

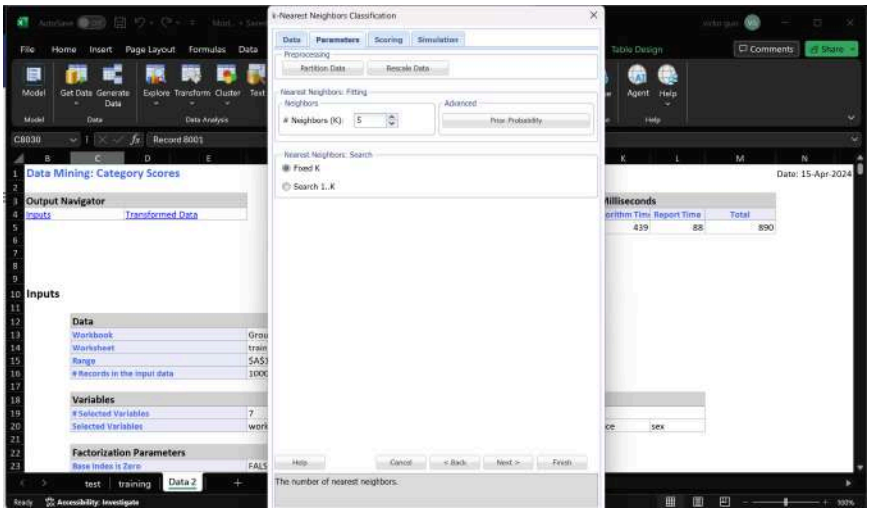


Fig. 6.2. Parameter Tabs

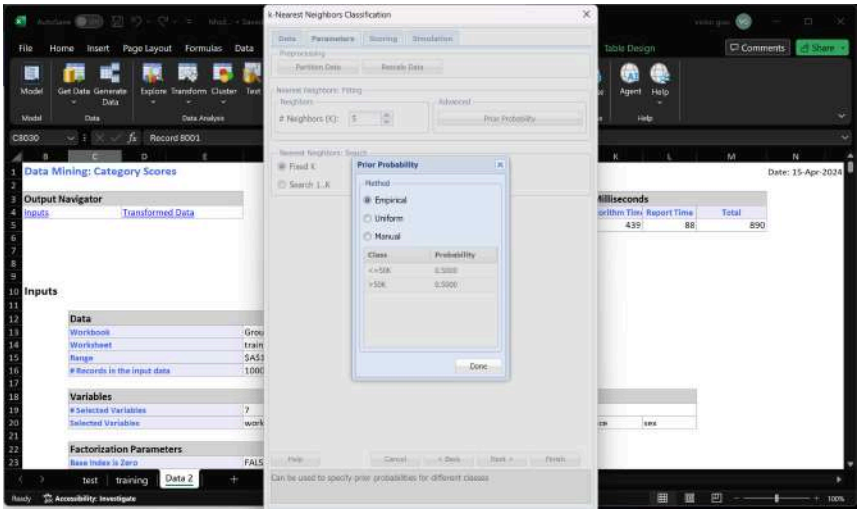


Fig. 6.3. Prior Probability Tab

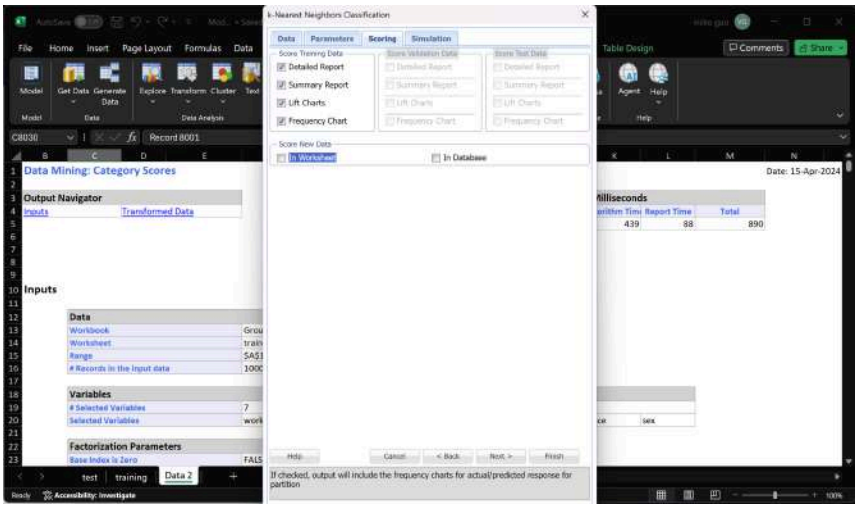


Fig. 6.4. Scoring Tabs

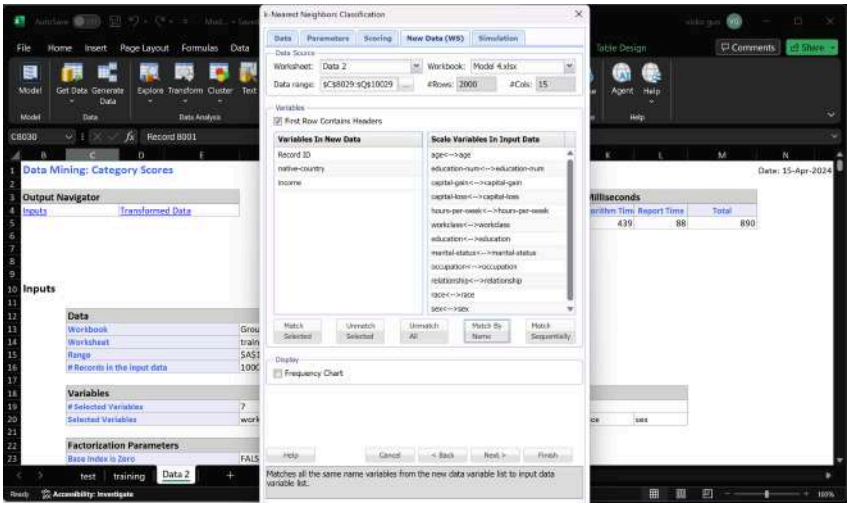


Fig. 6.5. New Data (WS) Tab

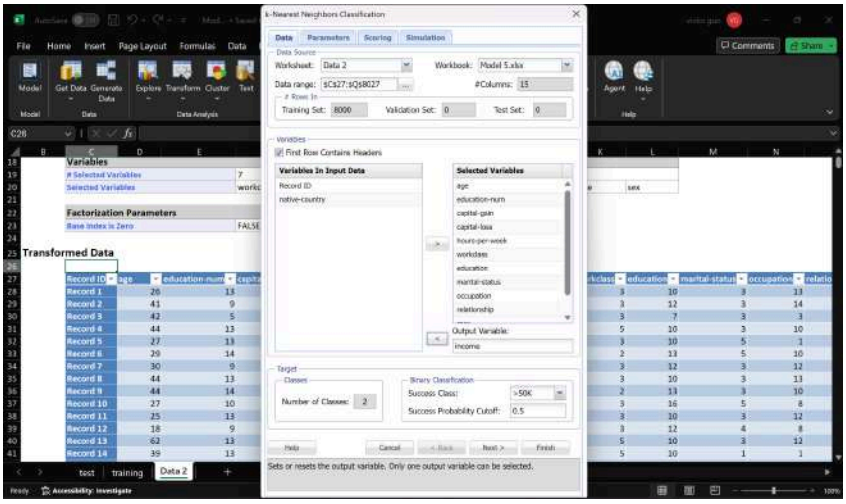


Fig. 7.1

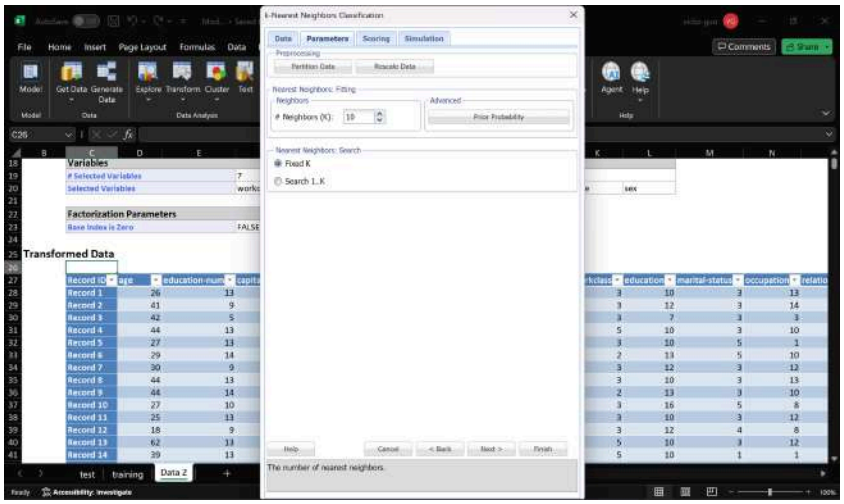


Fig. 7.2

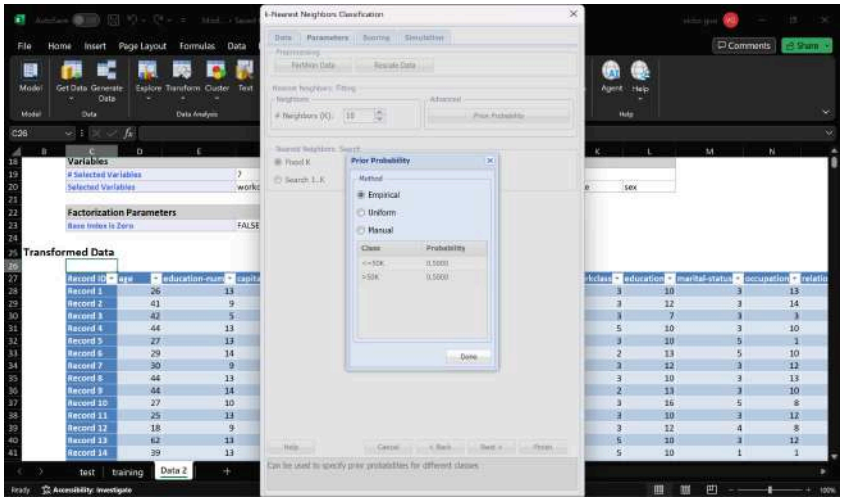


Fig. 7.3

Fig. 7.4

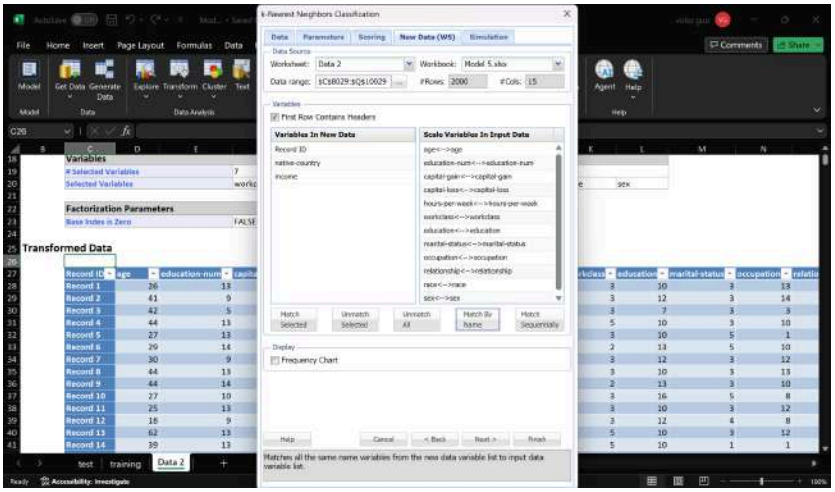

k-Nearest Neighbors Classification
Data
Parameters
Scoring
New Data (WS)
Simulation
Data Source
Worksheet: Data 2
Workbook: Model 5.xlsx
Data range: $C$8029:$Q$10029
#Rows: 2000
#Cols: 15
First Row Contains Headers
Variables In New Data
Record ID
native-country
income
Scale Variables In Input Data
age<->age
education-num<->education-num
capital-gain<->capital-gain
capital-loss<->capital-loss
hours-per-week<->hours-per-week
workclass<->workclass
education<->education
marital-status<->marital-status
occupation<->occupation
relationship<->relationship
race<->race
sex<->sex
Match Selected
Unmatch Selected
Unmatch All
Match By Name
Match Sequentially
Display
Frequency Chart
Help
Cancel
< Back
Next >
Finish
Matches all the same name variables from the new data variable list to input data variable list.
Variables
Selected Variables
Selected Variables
Factorization Parameters
Base Index in Zero
FALSE
Transformed Data
Record ID
age
education-num
Record 1
Record 2
Record 3
Record 4
Record 5
Record 6
Record 7
Record 8
Record 9
Record 10
Record 11
Record 12
Record 13
Record 14
test
training
Data 2

Fig. 7.5

II. Result and Observation

1. Introduction

In this part of the report, we are going to calculate the percentage of the error on each our model, and then generate the prediction model based on the test data set from the “test” sheet data from Phase 1 of this project.

2. Model 1

Recall that we have the following parameters for Model 1:

- Data: Data 2
- Type of Model: Decision Tree
- Parameter used:
  - Success Class: >50K
  - Number of Classes: 2
  - Success probability cut-off: 0.5
  - Tree to display: Fully Grown
  - Records in Terminal Nodes: 1
  - Maximum Number of Leaves: 7
  - Set the Test set and Training set variables to Match by Name

With this information, and the steps provided in the Design Report section, the following output will be provided by XLMiner.

A. Outputs and Observation

A.1. Error Model

Confusion Matrix		
Actual\Predicted	<=50K	>50K
<=50K	4326	556
>50K	928	2190

Error Report			
Class	# Cases	# Errors	% Error
<=50K	4882	556	11,38877509
>50K	3118	928	29,76266838
Overall	8000	1484	18,55

The figure table is the result of the accuracy of Model 1, according to the result. For the accuracy model, we use Confusion Matrix and also Error Report. This method is used to determine how accurate the result is based on the given input.

A.1.1. Confusion Matrix

A Confusion Matrix is a table that helps us understand the performance of our prediction model. It's called a "confusion" matrix because it shows how often our model is getting "confused" and making incorrect predictions.

- Here's how to construct it:
  - "Actual <=50K" and "Predicted <=50K": This is the number of times our model correctly predicted that the data would be "<=50K". We call these True Positives (TP).
  - "Actual >50K" and "Predicted >50K": This is the number of times our model correctly predicted that the data would be ">50K". We call these True Negatives (TN).
  - "Actual <=50K" and "Predicted >50K": This is the number of times our model incorrectly predicted that the data would be ">50K". We call these False Positives (FP).

- ➢ “Actual >50K” and “Predicted <=50K”: This is the number of times our model incorrectly predicted that the data would be "<=50K". We call these False Negatives (FN).

The sum of “Actual” data values (TP+TN) tells us how often our model is making correct predictions. The sum of “Predicted” data values (FP+FN) tells us how often our model is making incorrect predictions.

In this case, the sum of “Actual” data values (TP+TN) is 4326 + 2190 which is 6516, and the sum of “Predicted” data values (FP+FN) is 928 + 556 which is 1484. The large value of the sum of “Actual” data values (TP+TN) indicates that the model is quite accurate.

A.1.2. Error Report

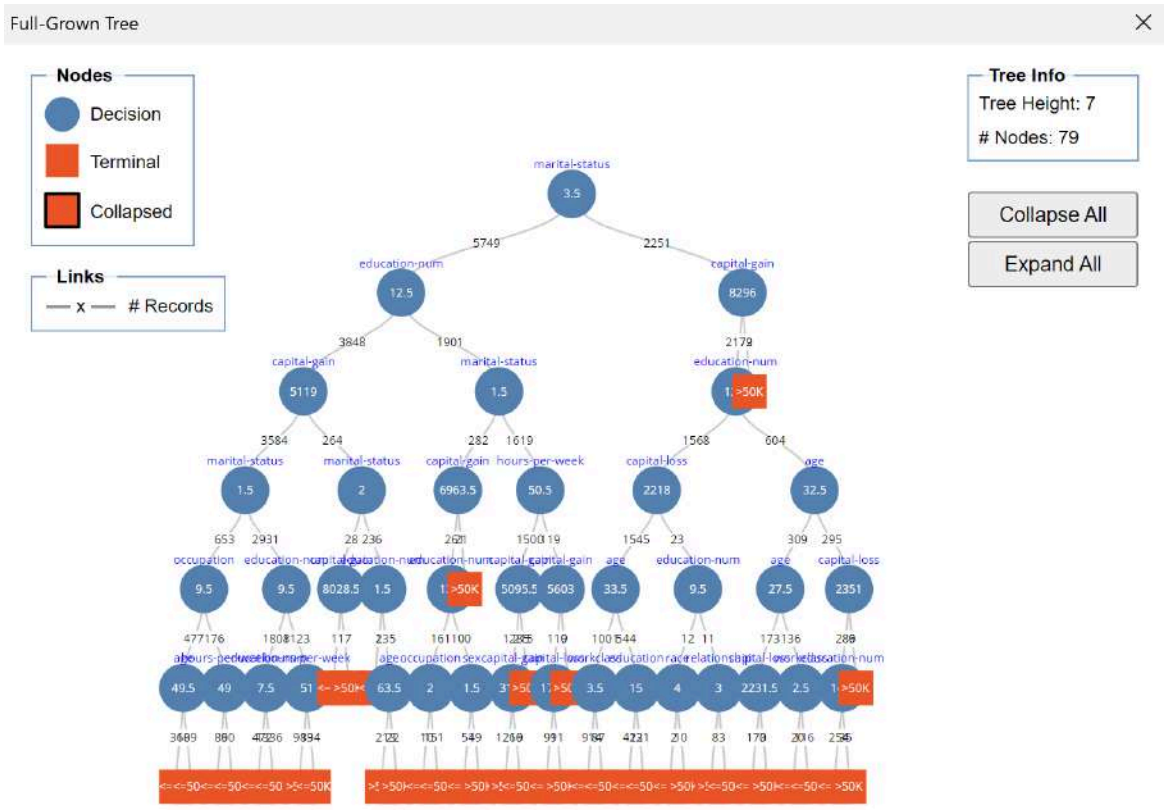
Error Report in the context of a prediction model is a summary of the mistakes made by the model during the prediction process. It’s a way to understand the performance of our model, whether our model accuracy is good or not.

- ❖ The following are the necessary terms and steps to construct Error Report:
  - ➢ First we need to create a table based on the image above
  - ➢ The term “#Cases” represents the total number of cases we used in the training data, whether the result of the income of each individual is “>50K” or “<=50K”
  - ➢ The term “#Error” represents the total number of cases we used in the training data that the prediction is actually false. For example, in case A, the output should be “<=50K” however, the prediction model detect it as “>50K”, therefore we put this data to the column of # error and the row of “>50K” indicating that it is an error and the output should be “<=50K”
  - ➢ The term “%Error” indicates the error divided by the number of the case that is given the desired output times 100%. For example, “# Error” for “>50K” divided by “# Cases” “>50K” times 100% will be the result for “%Error” (928/3118 * 100% = 29,76266838%).

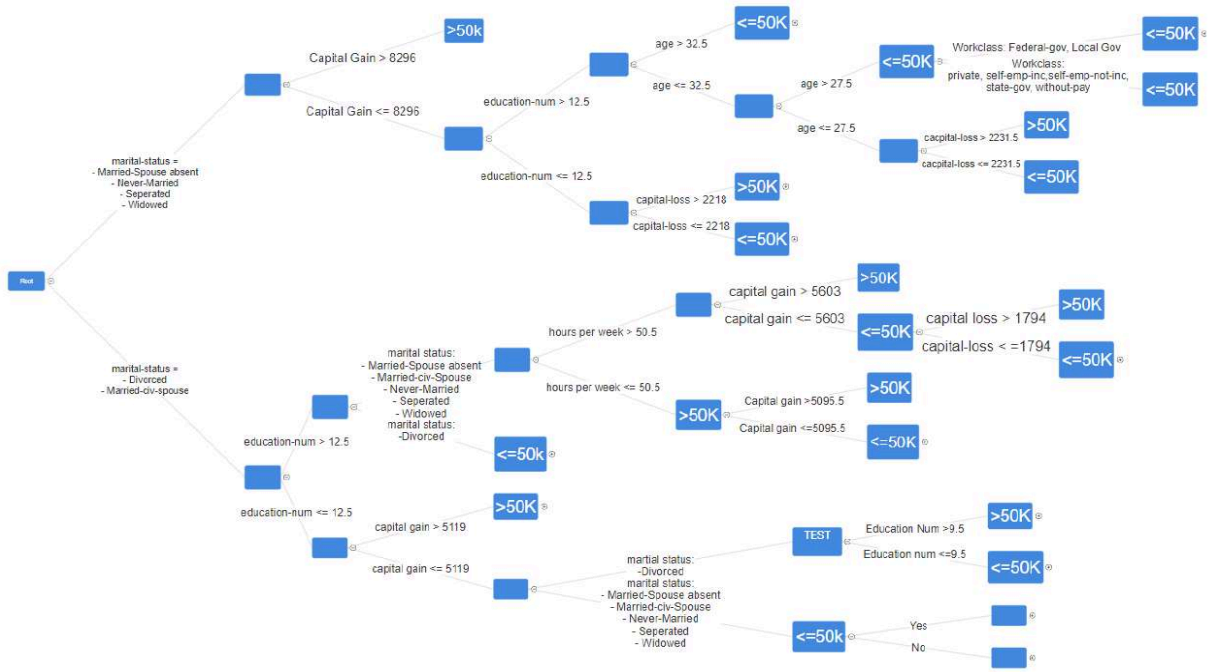
In our situation, the error is about 18.55%. That means that about 18.55% of the time, our Model 1 doesn't get things right. This is actually a pretty good track record. It means that if we use this model to make predictions on new data, we can expect it to be off the mark about 18.55% of the time.

A.2. Model 1: Decision Tree

The following is the decision Tree generated by XLMiner:



The following is the modified version of the decision Tree that is generated by XLMiner based on the Lecture Notes provided in HKUST Course COMP 1942:



And the following are the matrices generated by XLminer using decision tree model:

Metrics	
Metric	Value
Accuracy (#correct)	6516
Accuracy (%correct)	81,45
Specificity	0,886112
Sensitivity (Recall)	0,702373
Precision	0,797524
F1 score	0,74693
Success Class	>50K
Success Probability	0,5

B. Test if there is 2 raw output

Based on the given decision tree we can predict whether a person has an income “>50K” or “<=50K”. This prediction accuracy is around 82.45 % based on the error report, now given 2 new data of a person:

age	workclass	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country
51	Private	Prof-school	15	Married-civ-spouse	Prof-specialty	Husband	White	Male	0	0	50	United-States
30	Private	Masters	14	Never-married	Prof-specialty	Not-in-family	White	Male	0	0	40	Ireland

We can now predict what is the income for both person, wether it is “>50K” or “<=50K”. The following are the steps for predicting the income based on the given Model, Model 1:

1. **First Data**

- We will generate the result based on the decision tree generated by XLMiner with a modification from the lecture note of the COMP 1942 Course.
- First we see the “marital-status” of the person which is Married-civ-spouse . So we move to the “marital-status = -Divorced -Married-civ-spouse” arrow arriving in the next node
- Then the next node of the tree request us to find the value of the “education-num”, which in this case is 15, so we move to the “education-num > 12.5” arrow arriving to the next node.
- In the next node, it also request “marital-status” of the person with a different condition, so next we go to “marital status: - Married-Spouse absent - Married-civ-Spouse - Never-Married - Separated - Widowed” arrow.
- Next node, it requests the number of “Hours per work”. Again, since the number of “Hours per work” is 50, we move to the “Hours per work <=50.5” arrow.
- We also do the same one for “capital-gain”. Since “capital-gain” = 0, we go to “capital gain <= 5095.5”, and finally we reach the answer that the income value is “>50K”

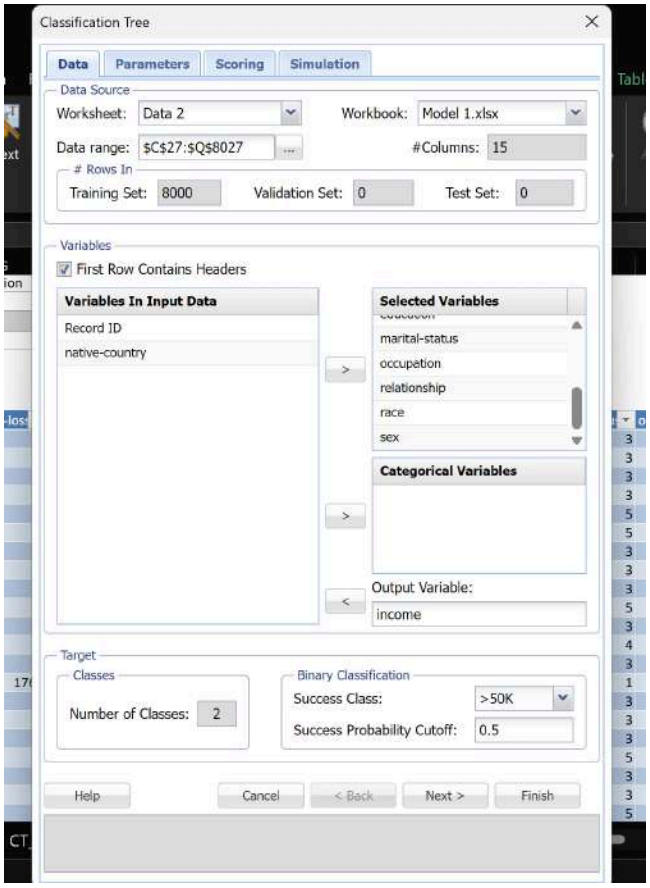
2. **Second Data**

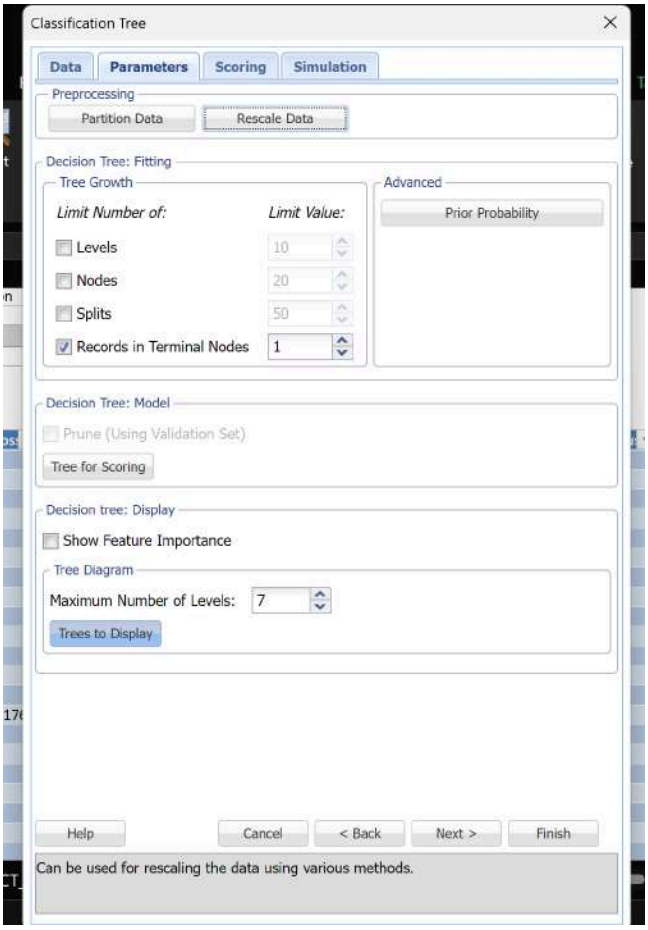
- Similar to the first data, we will generate the result for this one using a decision tree generated by XLMiner with a modification from the lecture note of the COMP 1942 Course. We first see the “marital-status” of the person who is Never Married. So we move to the “marital-status = -Married-Spouse absent - Never-Married - Separated - Widowed” arrow arriving in the next node
- Then the next node of the tree requests us to find the value of the “capital-gain”, which in this case is 0, so we move to the “capital-gain > 8296” arrow arriving at the next node.
- In the next node, it also requests the “education-num” of the person with a different condition, we know that the “education-num” of this person is 14. So next we go to “education-num >12.5” arrow.

- In the next node, it request the number of “age”. Since the number of “Age” is 30, we move to the “Age <= 32.5”arrow.
- Similar to the previous node, it request the number of “age” with a different condition. Since the number of “Age” is 30, we now go to “Age > 27” arrow.
- In this node, it request the information of “work-class”. Since the person on this data “work-class” is “Private” we go to Workclass: private, self-emp-inc,self-emp-not-inc, state-gov, without-pay. Which will give a result “income” of “<=50K”

C. Prediction of Data 4

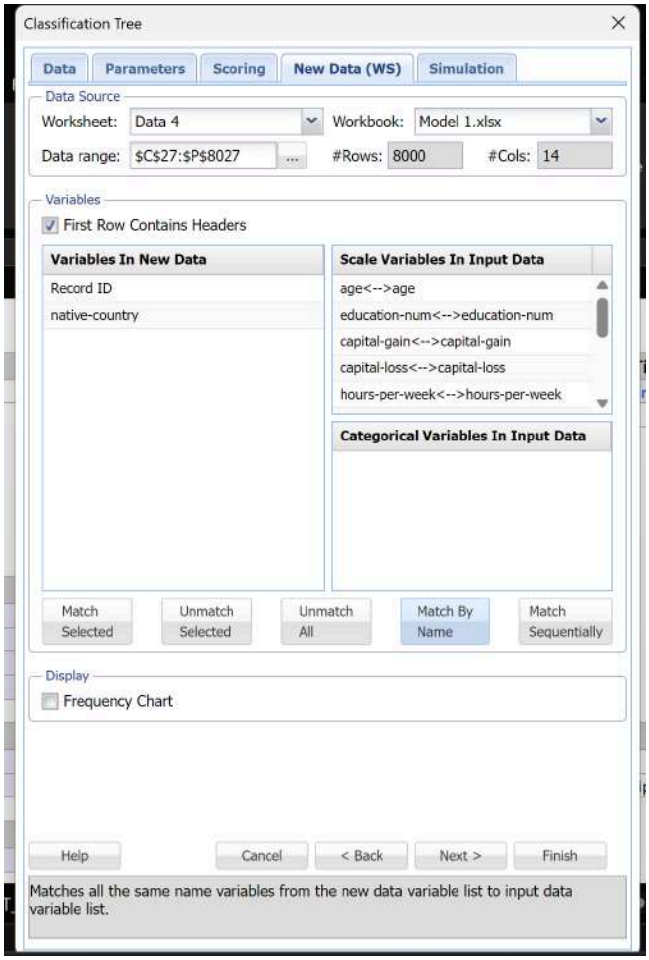
Now we are going to predict Data 4, which is the numerical “Test” Datasheet from phase 1. We will be using XLMiner to predict the data. Notice the steps are similar with the steps that is generated in phase 2. We just do a little modification from Phase 2. The following are the images of the modification steps to generate the prediction of Data 4 using Decision Tree Classifier with XLMiner:

- To generate, first select classify then classification tree in “Data 2” sheet. Now we see the data tab, set the tab to “Data” tab, and set the data range to C27 until C8027, as we want to use the first 8000 entries as our training data. We set all the variables in “Variables in Input Data” to selected Variables except “Record ID”, “native-country”, and “Income”, set the “Ouput Variable” to “Income”. Finally, keep the success Probability Cutoff to 0.5, the Number of Classes to 2,	

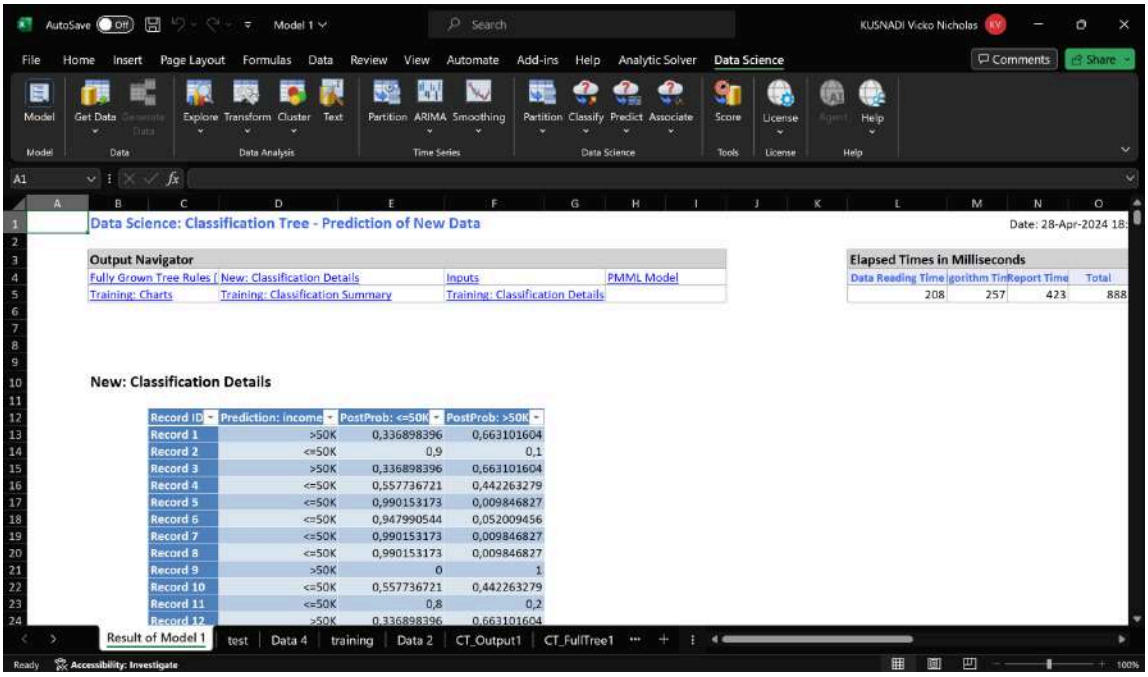
and also the Success class to >50K.	
- In the Parameters Tab, we make changes to certain settings. We enable the inclusion of records in terminal nodes and set it to 1. We also limit the maximum number of levels to 7, which is the maximum allowed. Clicking on Trees to Display allows us to choose the Fully Grown option for displaying the complete classification tree.	

- Moving to the Scoring Tab, we evaluate the performance of the model on the training data by selecting all the relevant checkboxes which are “Detailed Report”, “Summary Report”, “Lift Charts”, and “Frequency Chart”. In the Score New Data section, we choose the option to score data within the same worksheet as the training set. This leads to the creation of a new tab called the New Data (WS) Tab, where we set up the test data.

- Change the worksheet to “Data 4”. Then select the data range from C27 to P8027. After that, press Match By Name button, then press Finish button.



The result of Model 1 is as follows



The result file of this Model will be sent on TXT file

D. Observation

Based on the result of this model,as we mentioned before, we manage to create a decision tree, and the following are the "unique" rules that must be applied on this Model:

- Rule 1:
  - If marital-status = Married-Spouse absent or Never-Married or Seperated or Widowed capital-gain is more than 8296
    - Then the person must have income ">50K" with confidence of 100%
- Rule 2:
  - If martial-status = Married-Spouse absent or Never-Married or Seperated or Widowed capital-gain is less than or equal 8296 and age is less than or equal to 32.5
    - Check: If capital loss more than 2231.5
      - Then the person must have income ">50K" with confidence of 100%
    - Else:
      - Then the person must have income "<=50K" with confidence of 100%

- Rule 3: If marital-status = Divorced or Married-civ-spouse and capital-gain less than or equal to 5119 and
  - Check: If Marital-status = Divorce
    - Then the person must have income “<=50K” with confidence of 100%
  - Else:
    - If hours per week more than 51
      - Then the person must have income “<=50K” with confidence of 100%
    - If hours per week less than or equal to 51
      - Then the person must have income “>50K” with confidence of 100%

E. Conclusion

Data mining using a decision tree model has give out promising results in predicting income levels with an accuracy of approximately 82.45%. By analyzing various attributes such as marital status, education level, hours worked per week, and capital gain, our model, referred to as Model 1, successfully predicts whether an individual's income falls within the ">50K" or "<=50K" range.

We tested the model on two new data points to demonstrate its effectiveness. For the first data point, the decision tree analysis identified a marital status of "Married-civ-spouse" and an education level of 15, leading to a prediction of ">50K" income. Similarly, for the second data point, the model considered attributes like marital status, capital gain, education level, age, and work class, ultimately predicting an income of "<=50K".

In summary, using decision tree models to predict income levels is quite efficient. With an accuracy of 82.45%, our model provides a reliable tool for predicting income and can contribute to informed decision-making and resource allocation in various sectors.

3. Model 2

Recall that we have the following parameters for Model 2:

- Data: Data 1
- Type of Model: Naive Bayesian
- Parameter used:
  - Success Class: >50K
  - Number of Classes: 2
  - Success probability cut-off: 0.5
  - Laplace Smoothing: False
  - Prior Probability Method: Empirical
  - Show Prior Conditional Probability: True
  - Show Log Density: True
  - Set the Test set and Training set variables to Match by Name

With this information, and the steps provided in the Design Report section, the following output will be provided by XLMiner.

A. Outputs and Observation

A.1. Error Model

Confusion Matrix

Actual\Predicted	<=50K	>50K
<=50K	4155	727
>50K	436	2682

Error Report

Class	# Cases	# Errors	% Error
<=50K	4882	727	14,89143794
>50K	3118	436	13,98332264
Overall	8000	1163	14,5375

The figure table is the result of the accuracy of Model , according to the result. For the accuracy model, we use Confusion Matrix and also Error Report. This method is used to determine how accurate the result is based on the given input.

A.1.1. Confusion Matrix

A Confusion Matrix is a table that helps us understand the performance of our prediction model. It's called a "confusion" matrix because it shows how often our model is getting "confused" and making incorrect predictions.

- Here's how to construct it:
  - “Actual <=50K” and “Predicted <=50K”: This is the number of times our model correctly predicted that the data would be "<=50K". We call these True Positives (TP).
  - “Actual >50K” and “Predicted >50K”: This is the number of times our model correctly predicted that the data would be ">50K". We call these True Negatives (TN).
  - “Actual <=50K“ and “Predicted >50K”: This is the number of times our model incorrectly predicted that the data would be ">50K". We call these False Positives (FP).
  - “Actual >50K” and “Predicted <=50K”: This is the number of times our model incorrectly predicted that the data would be "<=50K". We call these False Negatives (FN).

The sum of “Actual” data values (TP+TN) tells us how often our model is making correct predictions. The sum of “Predicted” data values (FP+FN) tells us how often our model is making incorrect predictions.

In this case, the sum of “Actual” data values (TP+TN) is 4155 + 2682 which is 6837 and the sum of “Predicted” data values (FP+FN) is 727 + 436 which is 1163. The large value of the sum of “Actual” data values (TP+TN) indicates that the model is quite accurate.

A.1.2. Error Report

Error Report in the context of a prediction model is a summary of the mistakes made by the model during the prediction process. It’s a way to understand the performance of our model, whether our model accuracy is good or not.

- The following are the necessary terms and steps to construct Error Report:
  - First we need to create a table based on the image above

- The term “#Cases” represents the total number of cases we used in the training data, whether the result of the income of each individual is “>50K” or “<=50K”
- The term “#Error” represents the total number of cases we used in the training data that the prediction is actually false. For example, in case A, the output should be “<=50K” however, the prediction model detect it as “>50K”, therefore we put this data to the column of # error and the row of “>50K” indicating that it is an error and the output should be “<=50K”
- The term “%Error” indicates the error divided by the number of the case that is given the desired output times 100%. For example, “# Error” for “>50K” divided by “# Cases” “>50K” times 100% will be the result for “%Error” (436/3118 * 100% = 13.98332264%).

In our situation, the error is about 14.54%. That means that about 14.54% of the time, our Model 2 doesn't get things right. This is a pretty good track record. It means that if we use this model to make predictions on new data, we can expect it to be off the mark about 14.54% of the time.

A.2. Model 1: Naive Bayesian

The following are the prior-probability generated by Naive Bayesian Classifier using XLMiner:

Prior Probability

Class	Probability
<=50K	0,61025
>50K	0,38975

Prior Conditional Probability: Training

Prior Conditional Probability: Training-age		
Value/Class	<=50K	>50K
26	0,03011061	0,005772931
27	0,037279803	0,0121873
41	0,02458009	0,035920462
42	0,022326915	0,036882617
19	0,01351905	0
44	0,021917247	0,034958307
30	0,030929947	0,019884541
74	0,000614502	0,001603592
29	0,033183122	0,019243105
25	0,033183122	0,006414368
90	0,00102417	0,001282874
45	0,019459238	0,036882617
18	0,00553052	0
62	0,006349857	0,00609365
64	0,006964359	0,005131495
39	0,024375256	0,038486209
40	0,026833265	0,038806928
53	0,015977059	0,027261065
68	0,003072511	0,002565747
35	0,03461696	0,029826812
37	0,02806227	0,038806928
79	0,000204834	0,000641437
24	0,027242933	0,003848621
69	0,001433839	0,003527903
34	0,030929947	0,03624118
51	0,014748054	0,029506094
32	0,031954117	0,024695318
20	0,014952888	0
23	0,02458009	0,00192431
65	0,003277345	0,004490058
38	0,027447767	0,033354715
36	0,029700942	0,036882617
43	0,021712413	0,037844772
21	0,01454322	0,000641437
60	0,008398197	0,013790892
46	0,020483408	0,035920462
57	0,009627202	0,014432328
54	0,011880377	0,021167415
33	0,03461696	0,023733162
17	0,003891848	0
75	0,000614502	0,000320718
28	0,036050799	0,016677357
67	0,002662843	0,004490058
50	0,013109381	0,032392559
22	0,019459238	0,000962155
59	0,01003687	0,013470173
52	0,014133552	0,021808852
55	0,012904547	0,015073765
61	0,006759525	0,011225144
47	0,02007374	0,031751123
49	0,017820565	0,025016036
63	0,004916018	0,004490058
58	0,010241704	0,01603592
56	0,011265875	0,019563823
31	0,03461696	0,027261065
48	0,018844736	0,024695318
70	0,002662843	0,000641437
85	0,000204834	0
72	0,001843507	0,000641437
66	0,003891848	0,006414368
81	0,00102417	0,000320718
73	0,001433839	0,001603592
71	0,001638673	0,002565747
77	0,000614502	0,000962155
76	0,000409668	0
78	0,000409668	0,000962155
82	0,000614502	0
84	0,000204834	0

Prior Conditional Probability: Training-workclass		
Value/Class	<=50K	>50K
Private	0,751536256	0,668056446
Self-emp-not-in	0,113273249	0,077934573
Self-emp-inc	0,032363785	0,068313021
Local-gov	0,054281032	0,085311097
Federal-gov	0,019868906	0,050994227
State-gov	0,02806227	0,049069917
Without-pay	0,000614502	0,000320718

Prior Conditional Probability: Training-education		
Value/Class	<=50K	>50K
Masters	0,043015158	0,114175754
Bachelors	0,145022532	0,313021167
HS-grad	0,34350676	0,199486851
9th	0,025809095	0,002565747
Assoc-voc	0,036665301	0,048107761
Some-college	0,193158542	0,183771648
Prof-school	0,008193363	0,037844772
11th	0,034412126	0,008338679
Assoc-acdm	0,032363785	0,037524054
1st-4th	0,019254404	0,000962155
7th-8th	0,031544449	0,004169339
12th	0,013314215	0,005131495
10th	0,027242933	0,007697242
Doctorate	0,006964359	0,033675433
5th-6th	0,033387956	0,003527903
Preschool	0,006145023	0

Prior Conditional Probability: Training-education-num		
Value/Class	<=50K	>50K
13	0,145022532	0,313021167
10	0,193158542	0,183771648
9	0,34350676	0,199486851
14	0,043015158	0,114175754
5	0,025809095	0,002565747
7	0,034412126	0,008338679
12	0,032363785	0,037524054
11	0,036665301	0,048107761
2	0,019254404	0,000962155
4	0,031544449	0,004169339
8	0,013314215	0,005131495
15	0,008193363	0,037844772
6	0,027242933	0,007697242
16	0,006964359	0,033675433
3	0,033387956	0,003527903
1	0,006145023	0

Prior Conditional Probability: Training-marital-status		
Value/Class	<=50K	>50K
Married-civ-spo	0,436911102	0,848620911
Never-married	0,321999181	0,063181527
Widowed	0,023965588	0,008980115
Married-spouse	0,024989758	0,005452213
Divorced	0,157927079	0,061577935
Married-AF-spo	0,000204834	0,00192431
Separated	0,034002458	0,010262989

Prior Conditional Probability: Training-occupation		
Value/Class	<=50K	>50K
Sales	0,109791069	0,144964721
Tech-support	0,023351086	0,051314945
Adm-clerical	0,087054486	0,072161642
Prof-specialty	0,108152397	0,250481078
Craft-repair	0,139287177	0,099743425
Transport-movi	0,063088898	0,050994227
Other-service	0,100368701	0,016356639
Exec-managerial	0,128840639	0,221616421
Protective-serv	0,015772224	0,037203335
Machine-op-ins	0,075583777	0,035279025
Farming-fishing	0,092585006	0,008659397
Handlers-clean	0,046087669	0,010583708
Priv-house-serv	0,01003687	0
Armed-Forces	0	0,000641437

Prior Conditional Probability: Training-relationship		
Value/Class	<=50K	>50K
Husband	0,39819746	0,745028865
Unmarried	0,11634576	0,027902502
Not-in-family	0,307046293	0,112251443
Own-child	0,100368701	0,010904426
Wife	0,026833265	0,100705581
Other-relative	0,051208521	0,003207184

Prior Conditional Probability: Training-race		
Value/Class	<=50K	>50K
White	0,844940598	0,864336113
Black	0,06267923	0,062219371
Asian-Pac-Islan	0,06820975	0,058691469
Other	0,017410897	0,007697242
Amer-Indian-Es	0,006759525	0,007055805

Prior Conditional Probability: Training-sex		
Value/Class	<=50K	>50K
Male	0,728799672	0,839961514
Female	0,271200328	0,160038486

Prior Conditional Probability: Training-capital-gain		
Value/Class	<=50K	>50K
2174	0,001433839	0
0	0,931380582	0,790891597
25236	0	0,000320718
18481	0	0,000641437
2907	0,000819336	0
10566	0,000614502	0
6497	0,001229005	0
3942	0,001433839	0
7298	0	0,038165491
7688	0	0,034637588
2829	0,00102417	0
6418	0	0,000641437
5556	0	0,000320718
2977	0,000614502	0
4934	0	0,000641437
15024	0	0,036561899
15831	0	0,001282874
3674	0,000819336	0
4064	0,003072511	0
3137	0,003891848	0
3471	0,000819336	0
99999	0	0,015394484
2290	0,000409668	0
6849	0,004301516	0
3818	0,000819336	0
2580	0,000409668	0
4650	0,004916018	0
3325	0,004711184	0
3908	0,003482179	0
4508	0,001638673	0
2964	0,000204834	0
4386	0,000614502	0,008980115
5721	0,000409668	0
2993	0,000409668	0
3103	0,000614502	0,012828736
3456	0,000204834	0
5178	0	0,013149455
4931	0,000409668	0
3411	0,002253175	0
2885	0,000614502	0
27828	0	0,004810776
5013	0,007783695	0
6514	0	0,00192431
14344	0	0,003527903
2463	0,000614502	0
594	0,000614502	0
4865	0,001638673	0
3464	0,001433839	0
4787	0	0,003207184
9386	0	0,00192431
2936	0,000204834	0
20051	0	0,005131495
2407	0,000409668	0
5455	0,001229005	0
6767	0,000204834	0
7430	0	0,001603592
3273	0,00102417	0
8614	0	0,007055805
1471	0,000204834	0
10605	0	0,001282874
13550	0	0,003848621
14084	0	0,004169339
7978	0,000204834	0
10520	0	0,005131495
41310	0,000409668	0
4101	0,001229005	0
2202	0,001433839	0
3781	0,001433839	0
15020	0	0,000641437
4416	0,001229005	0
5060	0,000204834	0
2635	0,000204834	0
3887	0,000409668	0
2176	0,000409668	0
9562	0	0,000641437
1506	0,000614502	0
1848	0,000819336	0
11678	0	0,000320718
3432	0,000204834	0
3418	0,000204834	0
2009	0,000204834	0
6723	0,000409668	0
2062	0,000204834	0
7443	0,000409668	0
2105	0,000409668	0
6097	0	0,000320718
1831	0,000204834	0
2653	0,000204834	0

Prior Conditional Probability: Training-capital-loss		
Value/Class	<=50K	>50K
1980	0,002662843	0
1977	0	0,019563823
880	0,000204834	0
0	0,929537075	0,901860167
1762	0,000819336	0
1408	0,003072511	0
1628	0,00102417	0
2179	0,001229005	0
1590	0,004301516	0
1848	0	0,007376523
1411	0,000204834	0
2205	0,002048341	0
2258	0,00102417	0,001603592
1258	0,000614502	0
1564	0	0,002245029
1887	0	0,024695318
1672	0,003277345	0
1902	0,001638673	0,022450289
1944	0,000204834	0
1735	0,000204834	0
1669	0,002458009	0
1573	0,00102417	0
1721	0,000614502	0
1602	0,001433839	0
1485	0,001638673	0,003527903
2149	0,000204834	0
1876	0,003891848	0
2057	0,001229005	0
2129	0,000614502	0
1651	0,001229005	0
2559	0	0,001603592
2444	0	0,002245029
1719	0,00102417	0
2051	0,001229005	0
1741	0,003277345	0
625	0,001433839	0
2231	0	0,001603592
974	0,000204834	0
1579	0,002867677	0
1504	0,001433839	0
2339	0,002458009	0
1617	0,00102417	0
2201	0	0,000320718
2001	0,001638673	0
1726	0,000819336	0
2415	0	0,005772931
1974	0,002048341	0
1340	0,000204834	0
2754	0,000204834	0
2002	0,001843507	0
3900	0,000409668	0
2246	0	0,000962155
1380	0,000819336	0
2377	0,000819336	0,000320718
1092	0,000819336	0
1740	0,004711184	0
2547	0	0,000962155
1668	0,000614502	0
2267	0,000409668	0
3683	0,000204834	0
1825	0	0,000320718
2467	0,000204834	0
3004	0	0,000320718
2392	0	0,000320718
1138	0,000204834	0
2174	0	0,000641437
2824	0	0,000962155
2457	0,000204834	0
1429	0,000204834	0
3770	0,000204834	0
1755	0	0,000320718
2042	0,000204834	0
213	0,000409668	0
2603	0,000409668	0
653	0,000204834	0
1594	0,000409668	0
2238	0,000204834	0
155	0,000204834	0

Prior Conditional Probability: Training-hours-per-week		
Value/Class	<=50K	>50K
50	0,212208111	0,195317511
40	0,323228185	0,489416292
91	0,000409668	0
19	0,000614502	0
44	0,018639902	0,01603592
55	0,049979517	0,003207184
77	0,001433839	0
80	0,010651372	0,001282874
98	0,000819336	0,000320718
10	0,002253175	0,002565747
90	0,002458009	0,000641437
45	0,134985662	0,161962797
28	0,001229005	0
2	0,001433839	0,000641437
75	0,005325686	0
97	0,000204834	0,000320718
42	0,023351086	0,016356639
25	0,007169193	0,000320718
66	0,001843507	0
7	0	0,000320718
70	0,02908644	0,003207184
15	0,002867677	0
48	0,003277345	0,039769083
60	0,008603032	0,0121873
68	0,000819336	0
35	0,017206063	0,003848621
12	0,001638673	0,001282874
11	0,000204834	0
99	0,010241704	0,007055805
73	0,000409668	0
4	0,000204834	0,000641437
84	0,005325686	0,000320718
52	0,011675543	0,000320718
43	0,016386727	0,010904426
36	0,003687014	0,000641437
32	0,003482179	0,000320718
20	0,012904547	0,001603592
30	0,011675543	0,001603592
38	0,005325686	0,000962155
65	0,01454322	0,001282874
5	0,000409668	0,001603592
85	0,001229005	0,000320718
72	0,006554691	0,000641437
8	0,002253175	0,002245029
51	0,001843507	0
56	0,000409668	0,001282874
46	0,000204834	0,008659397
31	0,000204834	0,000320718
24	0,003891848	0,000320718
79	0,000204834	0
54	0,004096682	0
1	0,001433839	0,000320718
16	0,001229005	0,000320718
37	0,001638673	0,000320718
86	0,000409668	0
6	0,000409668	0,001282874
47	0,000614502	0,001603592
41	0,003891848	0,000962155
63	0,000409668	0,000320718
33	0,000409668	0
64	0	0,000320718
9	0,000204834	0,000641437
14	0,000614502	0
53	0,002458009	0
96	0,001229005	0,000320718
78	0,000614502	0
27	0,000204834	0,000320718
17	0,000204834	0
39	0,000409668	0
95	0,000204834	0
88	0,000409668	0
22	0,000204834	0
21	0,000409668	0,000320718
67	0,000204834	0,000641437
58	0,000204834	0
49	0	0,001282874
87	0,000204834	0
89	0,000204834	0
74	0,000204834	0
3	0,000204834	0,000320718
62	0,000409668	0
18	0,000204834	0
59	0,000204834	0
81	0,000409668	0
23	0,000204834	0
34	0,000204834	0
13	0	0,000641437
26	0,000204834	0
92	0,000204834	0

Prior Conditional Probability: Training-native-country		
Value/Class	<=50K	>50K
United-States	0,664686604	0,866581142
Cambodia	0,001638673	0,000641437
Philippines	0,020483408	0,01411161
Cuba	0,008807866	0,00609365
Haiti	0,006145023	0,001603592
Mexico	0,093609177	0,009300834
Puerto-Rico	0,016181893	0,004169339
Germany	0,013723884	0,009942271
England	0,007374027	0,009621552
Jamaica	0,01003687	0,002565747
Outlying-US(Gu	0,002662843	0
Canada	0,012085211	0,009300834
Italy	0,007578861	0,005131495
India	0,009627202	0,014432328
Guatemala	0,009422368	0
South	0,007374027	0,003848621
Columbia	0,009832036	0,000962155
Poland	0,006145023	0,002245029
Japan	0,006145023	0,004810776
Ecuador	0,004301516	0,000641437
Greece	0,002458009	0,002245029
Vietnam	0,008398197	0,000641437
El-Salvador	0,01556739	0,001603592
Hong	0,002048341	0,002245029
Dominican-Rep	0,007169193	0,000641437
China	0,006964359	0,005772931
Ireland	0,002662843	0,001282874
Holand-Netherl	0,000204834	0
Iran	0,003891848	0,002565747
Taiwan	0,003482179	0,005131495
Laos	0,002048341	0,000320718
Yugoslavia	0,001229005	0,002245029
France	0,002048341	0,003207184
Hungary	0,001843507	0,00192431
Portugal	0,00450635	0,001603592
Nicaragua	0,00450635	0,000641437
Thailand	0,002867677	0,000962155
Honduras	0,00102417	0,000320718
Peru	0,00450635	0,000641437
Trinadad&Toba	0,002253175	0
Scotland	0,002458009	0

The following is the matrice generated by Naive Bayesian Clasifier using XLMiner:

B. Test if there is 2 raw output

Based on the given output by the Naive Bayesian Classifier we can predict whether a person has an income ">50K" or "<=50K". This prediction accuracy is around 85.46 % based on the error report, now given 2 new data of a person:

age	workclass	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country
51	Private	Prof-school	15	Married-civ-spouse	Prof-specialty	Husband	White	Male	0	0	50	United-States
30	Private	Masters	14	Never-married	Prof-specialty	Not-in-family	White	Male	0	0	40	Ireland

We can now predict what the income is for both persons, whether it is ">50K" or "<=50K". Before we predict the data, we will be filtering all the Prior Conditional Probability data based on Data 1 and Data 2 attributes

Prior Probability

Class	Probabilit
<=50K	0,61025
>50K	0,38975

Prior Conditional Probability: Training

Prior Conditional Probability: Training-age		
Value/Class	<=50K	>50K
30	0,03092995	0,019884541
51	0,01474805	0,029506094

Prior Conditional Probability: Training-workclass		
Value/Class	<=50K	>50K
Private	0,75153626	0,668056446

Prior Conditional Probability: Training-education		
Value/Class	<=50K	>50K
Masters	0,04301516	0,114175754
Preschool	0,00614502	0

Prior Conditional Probability: Training-education-num		
Value/Class	<=50K	>50K
14	0,04301516	0,114175754
15	0,00819336	0,037844772

Prior Conditional Probability: Training-marital-status		
Value/Class	<=50K	>50K
Married-civ-spouse	0,4369111	0,848620911
Never-married	0,32199918	0,063181527

Prior Conditional Probability: Training-occupation		
Value/Class	<=50K	>50K
Prof-specialty	0,1081524	0,250481078

Prior Conditional Probability: Training-relationship		
Value/Class	<=50K	>50K
Husband	0,39819746	0,745028865
Not-in-family	0,30704629	0,112251443

Prior Conditional Probability: Training-race		
Value/Class	<=50K	>50K
White	0,8449406	0,864336113

Prior Conditional Probability: Training-sex		
Value/Class	<=50K	>50K
Male	0,72879967	0,839961514

Prior Conditional Probability: Training-capital-gain		
Value/Class	<=50K	>50K
0	0,93138058	0,790891597

Prior Conditional Probability: Training-capital-loss		
Value/Class	<=50K	>50K
0	0,92953707	0,901860167

Prior Conditional Probability: Training-hours-per-week		
Value/Class	<=50K	>50K
50	0,21220811	0,195317511
40	0,32322819	0,489416292

Prior Conditional Probability: Training-native-country		
Value/Class	<=50K	>50K
United-States	0,6646866	0,866581142
Ireland	0,00266284	0,001282874

The following are the steps for predicting the income based on the given Model, Model 2:

1. **Data 1**

In order to know the output "Income" of a data, we need to find both the probability of Income is ">50K" and "<=50K" given all the attributes(i.e, age, education-num, and else). The probability is denoted as P(Income = ">50K" or "<=50K" | age = 51, education-num = 15, capital-gain = 0, capital-loss = 0, hours-per-week = 50, workclass = Private, education = Prof-School, marital-status = Married-civ-spouse, occupation = Prof-Speciality, relationship = Husband, race = White, sex = Male, native-country = United-States). Notice that by the Prior Conditionally Probability data, we can get the information as follows

P(Income = ">50K" | Other attributes) = P(Income = ">50K" | age = 51, education-num = 15, capital-gain = 0, capital-loss = 0, hours-per-week = 50, workclass = Private, education = Prof-School, marital-status = Married-civ-spouse, occupation = Prof-Speciality, relationship = Husband, race = White, sex = Male, native-country = United-States)
P(age = 51, education-num = 15, capital-gain = 0, capital-loss = 0, hours-per-week = 50, workclass = Private, education = Prof-School, marital-status = Married-civ-spouse, occupation = Prof-Speciality, relationship = Husband, race = White, sex = Male, native-country = United-States | Income = ">50K") **Times (*)** P(Income = ">50K") **Divided by (÷)** M = P(age = 51, education-num = 15, capital-gain = 0, capital-loss = 0, hours-per-week = 50, workclass = Private, education = Prof-School, marital-status = Married-civ-spouse, occupation = Prof-Speciality, relationship = Husband, race = White, sex = Male, native-country = United-States)

The result is as follows

$$P(Income = >50K|OtherAttribues) = \frac{1.52733 * 10^{-7}}{M}$$

<u>P(Income = "<=50K" | Other attributes) =</u> P(Income = ">50K" | age = 51, education-num = 15, capital-gain = 0, capital-loss = 0, hours-per-week = 50, workclass = Private, education = Prof-School, marital-status = Married-civ-spouse, occupation = Prof-Speciality, relationship = Husband, race = White, sex = Male, native-country = United-States)

P(age = 51, education-num = 15, capital-gain = 0, capital-loss = 0, hours-per-week = 50, workclass = Private, education = Prof-School, marital-status = Married-civ-spouse, occupation = Prof-Speciality, relationship = Husband, race = White, sex = Male, native-country = United-States | Income = "<=50K")

Times (*)

P(Income = "<=50K")

Divided by (÷)

M = P(age = 51, education-num = 15, capital-gain = 0, capital-loss = 0, hours-per-week = 50, workclass = Private, education = Prof-School, marital-status = Married-civ-spouse, occupation = Prof-Speciality, relationship = Husband, race = White, sex = Male, native-country = United-States)

The result is as follows

$$P(Income = <= 50K|OtherAttribues) = \frac{6.41983 * 10^{-10}}{M}$$

Since the result of P(Income = ">50K" | Other attributes) is larger than P(Income = "<=50K" | Other attributes), we chose P(Income = ">50K" | Other attributes) as the result, so the income output of Data 1 is Income = ">50K"

2. **Data 2**

Similar to Data 1, In order to know the output "Income" of a data, we need to find both the probability of Income is ">50K" and "<=50K" given all the attributes(i.e, age, education-num, and else). The probability is denoted as P(Income = ">50K" or "<=50K" | age = 30, education-num = 14, capital-gain = 0, capital-loss = 0, hours-per-week = 40, workclass = Private, education = Master , marital-status = Never-married, occupation = Prof-Speciality, relationship = Not-in-family, race = White, sex = Male, native-country =Ireland). Notice that by the Prior Conditional Probability data, we can get the information as follows

P(Income = ">50K" | Other attributes) = P(Income = ">50K" | age = 30, education-num = 14, capital-gain = 0, capital-loss = 0, hours-per-week = 40, workclass = Private, education = Master , marital-status = Never-married, occupation = Prof-Speciality, relationship = Not-in-family, race = White, sex = Male, native-country =Ireland).
P(age = 30, education-num = 14, capital-gain = 0, capital-loss = 0, hours-per-week = 40, workclass = Private, education = Master , marital-status = Never-married, occupation = Prof-Speciality, relationship = Not-in-family, race = White, sex = Male, native-country =Ireland | Income = ">50K") **Times (*)** P(Income = ">50K") **Divided by (÷)** M = P(age = 30, education-num = 14, capital-gain = 0, capital-loss = 0, hours-per-week = 40, workclass = Private, education = Master , marital-status = Never-married, occupation = Prof-Speciality, relationship = Not-in-family, race = White, sex = Male, native-country =Ireland)

The result is as follows

$$P(Income = > 50K|OtherAttribues) = \frac{5.8354 * 10^{-11}}{M}$$

<u>P(Income = "<=50K" | Other attributes) =</u> P(Income = ">50K" | age = 30, education-num = 14, capital-gain = 0, capital-loss = 0, hours-per-week = 40, workclass = Private, education = Master , marital-status = Never-married, occupation = Prof-Speciality, relationship = Not-in-family, race = White, sex = Male, native-country =Ireland)

P(age = 30, education-num = 14, capital-gain = 0, capital-loss = 0, hours-per-week = 40, workclass = Private, education = Master , marital-status = Never-married, occupation = Prof-Speciality, relationship = Not-in-family, race = White, sex = Male, native-country =Ireland | Income = "<=50K")

Times (*)

P(Income = "<=50K")

Divided by (÷)

M = P(age = 30, education-num = 14, capital-gain = 0, capital-loss = 0, hours-per-week = 40, workclass = Private, education = Master , marital-status = Never-married, occupation = Prof-Speciality, relationship = Not-in-family, race = White, sex = Male, native-country =Ireland

The result is as follows

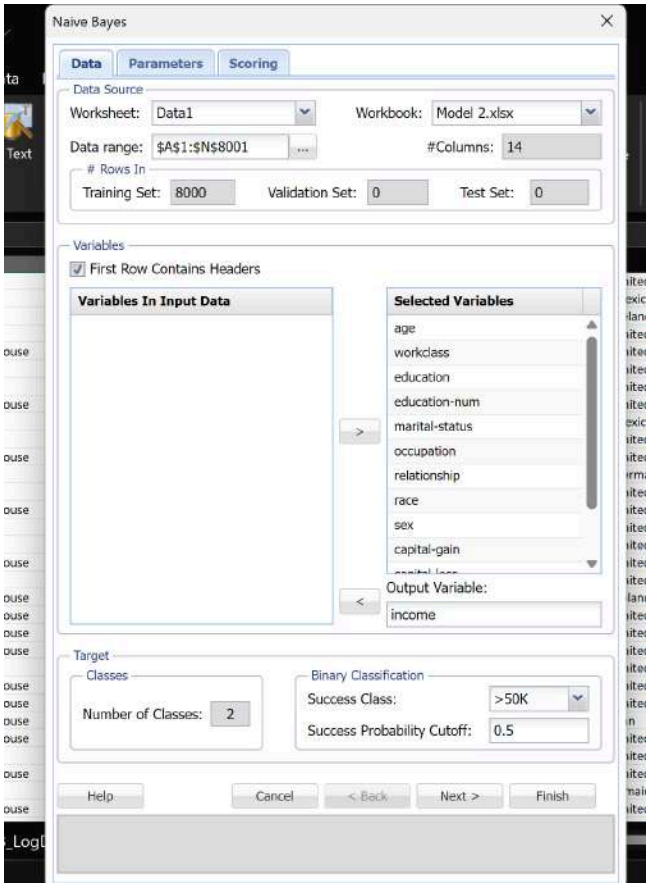
$$P(Income = <= 50K|OtherAttribues) = \frac{1.2878 * 10^{-10}}{M}$$

Since the result of P(Income = ">50K" | Other attributes) is less than P(Income = "<=50K" | Other attributes), we chose P(Income = "<=50K" | Other attributes) as the result, so the income output of Data 1 is Income = "<=50K"

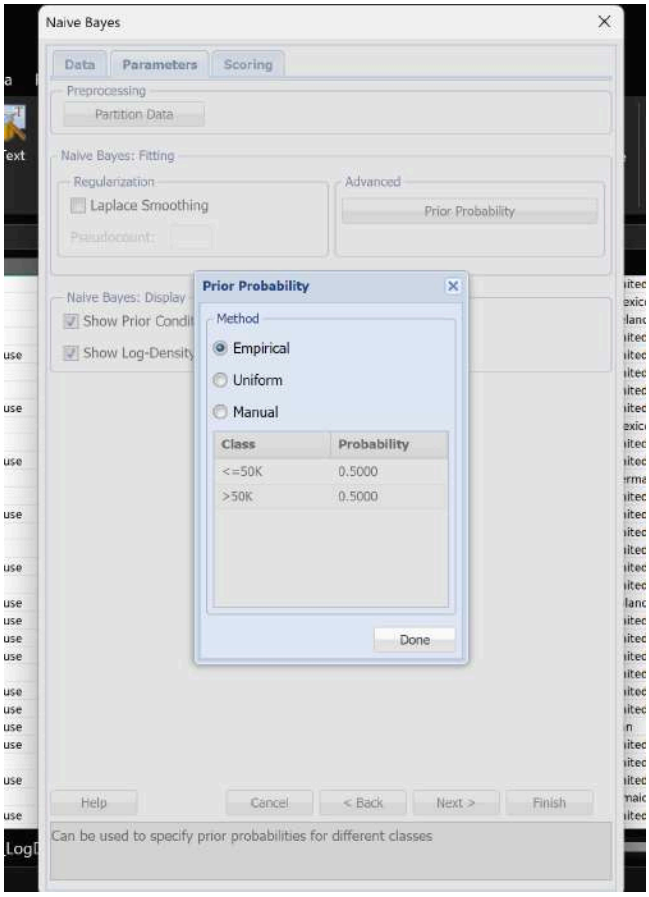
C. Prediction of Data 3

Now we are going to predict Data 3, which is the "Test" Datasheet from phase 1. We will be using XLMiner to predict the data. Notice the steps are similar with the steps that is generated in phase 2. We just do a little modification from Phase 2. The following are the images of the modification steps to generate the prediction of Data 3 using Naive Bayesian Classifier with XLMiner:

- The first step to generate the data is to select "Classify" and then "Naive Bayes" in the "Data 1" sheet. In the data tab, set the data range to A1 until A8001, as we want to use the first 8000 entries as our training data. Then, set the success probability cutoff to 0.5, the number of classes to 2, and the success class to ">50K". Next, press the "Next" button.

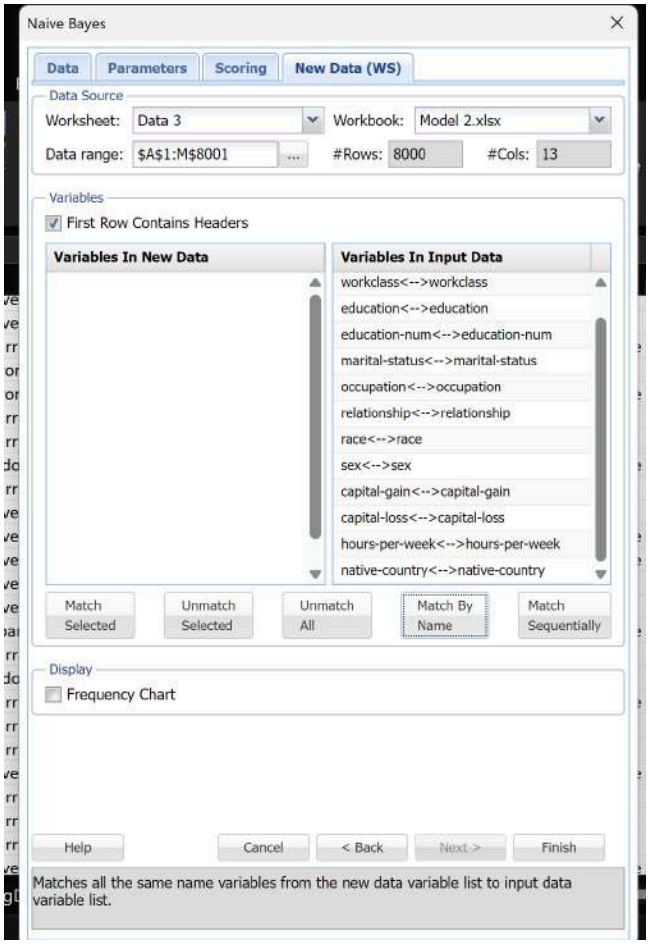


- In the parameters tab, disable Laplace Smoothing and select "Prior Probability". In the pop-up, select the "Empirical" option for the Prior Probability Method. In the "Display Options", check all the boxes in the "Naive Bayes: Display" section, and then press the "Next" button again to move to the "Scoring Tab".

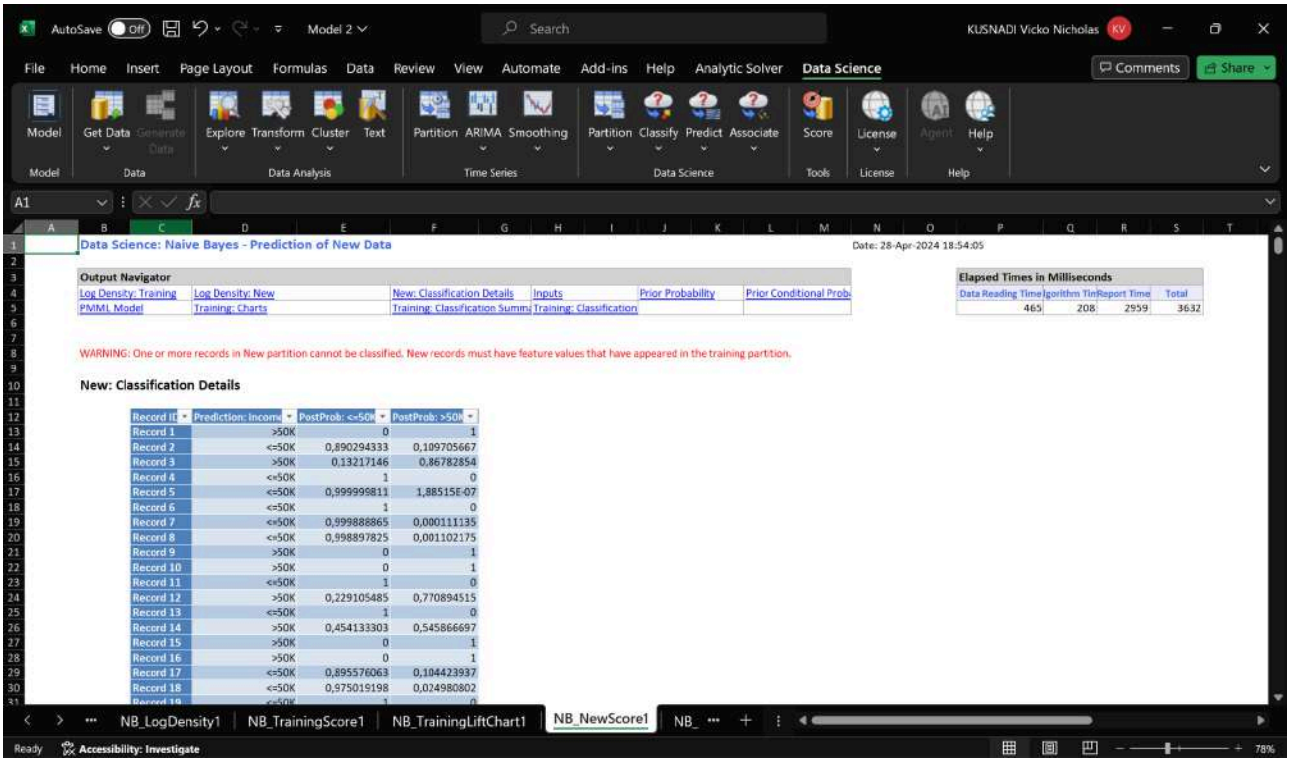


- In the "Scoring Tab", check all the boxes in the "Score training data" section, namely "Detailed Report", "Summary Report", "Lift Charts", and "Frequency Chart". In addition, in the "Score New Data" area, select the "In Worksheet" option. After selecting the "In Worksheet" option, the "New Data (WS)" tab will appear.

- In the New Data (WS) Tab, Change the worksheet to “Data 3”. Then select the data range from A1 to M8001. After that, press the “Match By Name” button, then press the “Finish” button.



The result of Model 2 is as follows



The result file of this Model will be sent on TXT file

D. Observation

Based on the result of this model,as we mentioned before, we manage to create a decision tree, and the following are the "unique" rules that must be applied on this Model:

- Let X be the new data that have attributes "A"
- Let P1 be the probability of X that have Income ">50k" given attributes "A"
- Let P2 be the probability of X that have Income "<=50k" given attributes "A"
- Rule 1:
  - If the value of P1 is more than the value of P2
    - Set X Income value to ">50K"
- Rule 2:
  - If the value of P2 is more than the value of P1
    - Set X Income value to "<=50K"

Based on the result of the model also, given new data X, based on the prior probability of this model, because the prior probability of Income "<=50k" is higher than the prior probability of income

“>50k” the probability of X have Income “<=50k” is higher than having income “>50k”. This also means, based on the prior probability, if X has certain attributes that gives an impact to the probability of an income (e.g income “>50K”), the probability of X having that income is higher(e.g income “>50K”).

E. Conclusion

In summary, we applied Model 2, which is a Naive Bayesian classifier, to predict income levels using Data 1. We utilized various parameters such as the success class (>50K), number of classes (2), success probability cut-off (0.5), Laplace smoothing (False), prior probability method (Empirical), and displayed prior conditional probability and log density.

The evaluation of Model 2 involved analyzing the accuracy of the predictions using Confusion Matrix and Error Report. The Confusion Matrix provided insights into the model's performance, showing the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). The high sum of actual data values (TP+TN) indicated that the model exhibited good accuracy.

The Error Report summarized the mistakes made by the model during the prediction process. It presented the number of cases and errors for each income class, along with the percentage of errors. In the case of Model 2, the error rate was approximately 14.54%, indicating that the model had a solid track record with a relatively low margin of error.

In conclusion, the Naive Bayesian classifier (Model 2) applied to Data 1 showed promising results in predicting income levels. With an error rate of around 14.54%, the model demonstrated good accuracy. These findings highlight the effectiveness of the Naive Bayesian approach in predicting income and provide valuable insights for decision-making in various domains related to income analysis and allocation of resources.

4. Model 3:

Recall that we have the following parameters for Model 3:

- Data: Data 2
- Type of Model: K Nearest Neighbor
- Parameter used:
  - Success Class: >50K
  - Number of Classes: 2
  - Success probability cut-off: 0.5
  - Fixed K with K = 3
  - Prior Probability to Empirical
  - Set the Test set and Training set variables to Match by Name

With this information, and the steps provided in the Design Report section, the following output will be provided by XLMiner.

A. Outputs and Observation

A.1. Error Model

Confusion Matrix		
Actual\Predicted	<=50K	>50K
<=50K	4369	513
>50K	352	2766

Error Report			
Class	# Cases	# Errors	% Error
<=50K	4882	513	10,50798853
>50K	3118	352	11,28928801
Overall	8000	865	10,8125

The figure table is the result of the accuracy of Model 3, according to the result. For the accuracy model, we use Confusion Matrix and also Error Report. This method is used to determine how accurate the result is based on the given input.

A.1.2. Confusion Matrix

A Confusion Matrix is a table that helps us understand the performance of our prediction model. It's called a "confusion" matrix because it shows how often our model is getting "confused" and making incorrect predictions.

- Here's how to construct it:
  - “Actual <=50K” and “Predicted <=50K”: This is the number of times our model correctly predicted that the data would be "<=50K". We call these True Positives (TP).
  - “Actual >50K” and “Predicted >50K”: This is the number of times our model correctly predicted that the data would be ">50K". We call these True Negatives (TN).
  - “Actual <=50K“ and “Predicted >50K”: This is the number of times our model incorrectly predicted that the data would be ">50K". We call these False Positives (FP).
  - “Actual >50K” and “Predicted <=50K”: This is the number of times our model incorrectly predicted that the data would be "<=50K". We call these False Negatives (FN).

The sum of “Actual” data values (TP+TN) tells us how often our model is making correct predictions. The sum of “Predicted” data values (FP+FN) tells us how often our model is making incorrect predictions.

In this case, the sum of “Actual” data values (TP+TN) is 4369 + 2766 which is 7135 and the sum of “Predicted” data values (FP+FN) is 513 + 352 which is 865. The large value of the sum of “Actual” data values (TP+TN) indicates that the model is quite accurate.

A.1.3. Error Report

Error Report in the context of a prediction model is a summary of the mistakes made by the model during the prediction process. It’s a way to understand the performance of our model, whether our model accuracy is good or not.

- The following are the necessary terms and steps to construct Error Report:
  - First we need to create a table based on the image above

- ➢ The term "#Cases" represents the total number of cases we used in the training data, whether the result of the income of each individual is ">50K" or "<=50K"
- ➢ The term "#Error" represents the total number of cases we used in the training data that the prediction is actually false. For example, in case A, the output should be "<=50K" however, the prediction model detect it as ">50K", therefore we put this data to the column of # error and the row of ">50K" indicating that it is an error and the output should be "<=50K"
- ➢ The term "%Error" indicates the error divided by the number of the case that is given the desired output times 100%. For example, "# Error" for ">50K" divided by "# Cases" ">50K" times 100% will be the result for "%Error" (928/3118 * 100% = 29,76266838%).

In our situation, the error is about *insert data*. That means that about 10.81% of the time, our Model 3 doesn't get things right. This is actually a pretty good track record. It means that if we use this model to make predictions on new data, we can expect it to be off the mark about 10.81% of the time.

A.2.1. Model 3: K Nearest Neighbor Classifier

The following are the metrics based on K Nearest Neighbor Classifier:

Metrics	
Metric	Value
Accuracy (#correct)	7135
Accuracy (%correct)	89,1875
Specificity	0,89492
Sensitivity (Recall)	0,887107
Precision	0,84355
F1 score	0,86478
Success Class	>50K
Success Probability	0,5

B. Test if there is 2 raw output

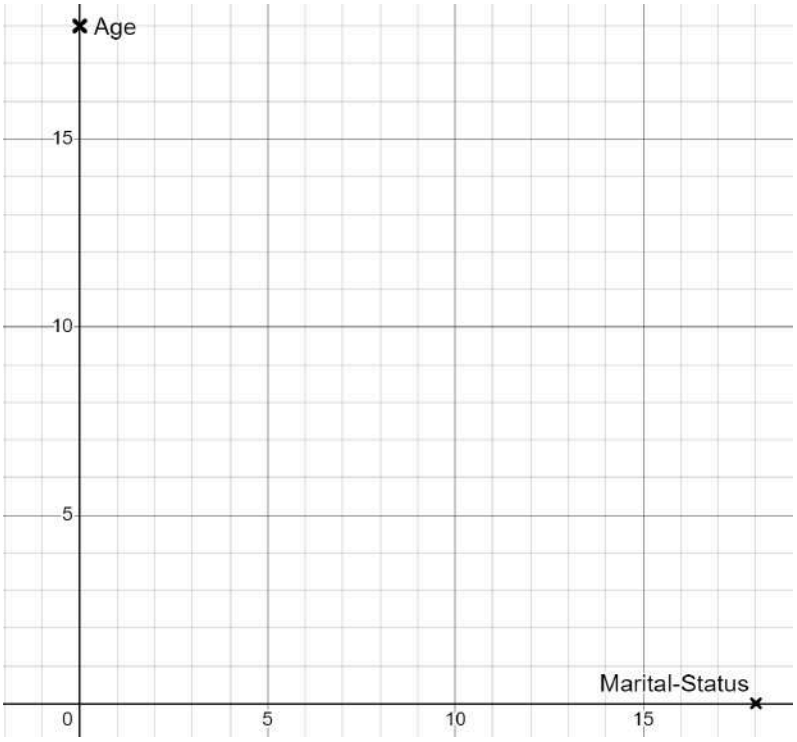
Based on the given output by K Nearest Neighbor Classifier we can predict wether a person have an income ">50K" or "<=50K". This prediction accuracy is around 89.19 % based on the error report, now given 2 new data of a person:

age	workclass	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country
51	Private	Prof-school	15	Married-civ-spouse	Prof-specialty	Husband	White	Male	0	0	50	United-States
30	Private	Masters	14	Never-married	Prof-specialty	Not-in-family	White	Male	0	0	40	Ireland

We can now predict what is the income for both person, wether it is ">50K" or "<=50K". However, in order to give an easy explanation, we will be using the transformation data (numerical data) as shown below:

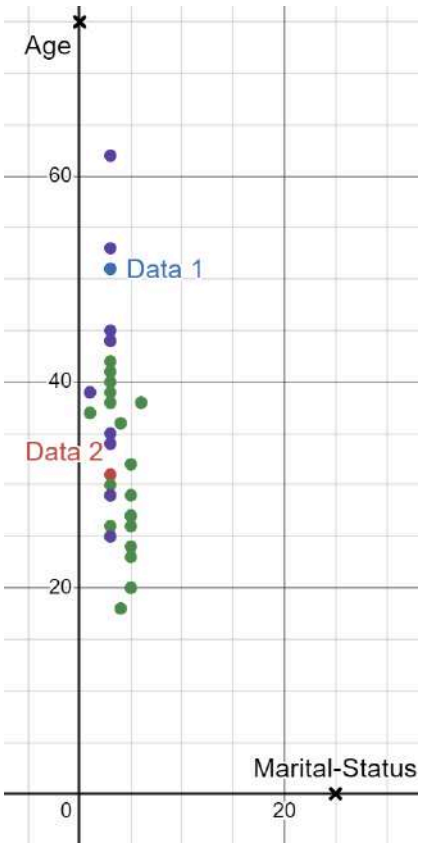
age	education-num	capital-gain	capital-loss	hours-per-week	native-country	workclass	education	marital-	occupation	relationship	race	sex
51	15	0	0	50	United-States	3	15	3	10	1	5	2
30	14	0	0	40	Ireland	3	13	5	10	2	5	2

We will also use a 2 dimensional graph assumption for the steps required for generating the result of K Nearest Neighbor Classifier. Since we are using 2 dimensional graph, we could only take 2 attributes as an input. In this case we are going to use "Marital status" attribute as the X component of a cartesian graph and "age" attribute as the Y component of a cartesian graph, just like the image below:



Lastly, we are also going to use the first 40 given data from phase 2, the following are the first 40 given data based on Phase 2 design report:

Record ID	marital-status	age	income
Record 1	3	26	<=50K
Record 2	3	41	<=50K
Record 3	3	42	<=50K
Record 4	3	44	<=50K
Record 5	5	27	<=50K
Record 6	5	29	<=50K
Record 7	3	30	<=50K
Record 8	3	44	>50K
Record 9	3	44	>50K
Record 10	5	27	<=50K
Record 11	3	25	>50K
Record 12	4	18	<=50K
Record 13	3	62	>50K
Record 14	1	39	<=50K
Record 15	3	25	<=50K
Record 16	3	40	<=50K
Record 17	3	53	>50K
Record 18	5	26	<=50K
Record 19	3	29	>50K
Record 20	3	35	>50K
Record 21	5	26	<=50K
Record 22	1	37	<=50K
Record 23	5	24	<=50K
Record 24	3	34	>50K
Record 25	3	51	>50K
Record 26	5	32	<=50K
Record 27	3	51	>50K
Record 28	5	20	<=50K
Record 29	5	23	<=50K
Record 30	1	37	<=50K
Record 31	3	39	<=50K
Record 32	6	38	<=50K
Record 33	5	20	<=50K
Record 34	5	27	<=50K
Record 35	3	45	>50K
Record 36	3	38	<=50K
Record 37	3	29	<=50K
Record 38	3	51	<=50K
Record 39	4	36	<=50K
Record 40	1	39	>50



The Image on the left are the 2 dimensional graphical vision. The purple dot represents the data that have an output "Income" of ">50K" and the green dot represent the data that have an output "Income" of "<=50K" The following are the steps for predicting the income based on the given Model, Model 3:

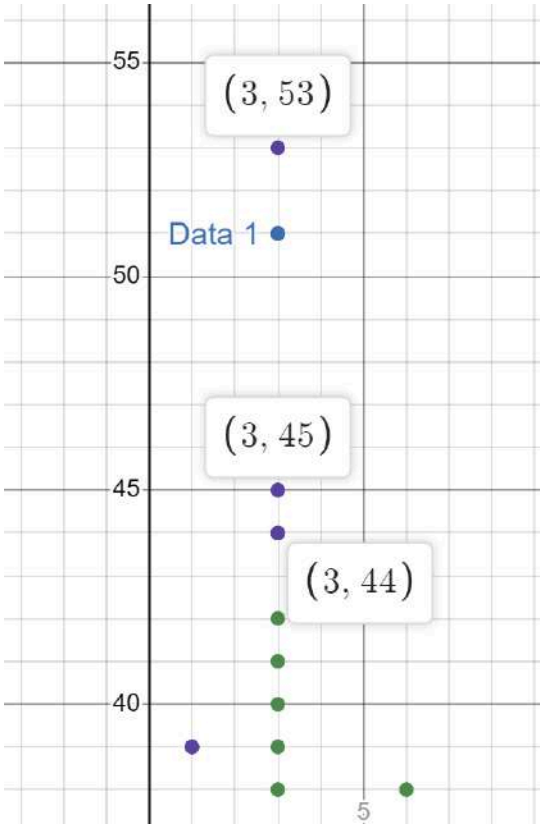
1. **First Data**

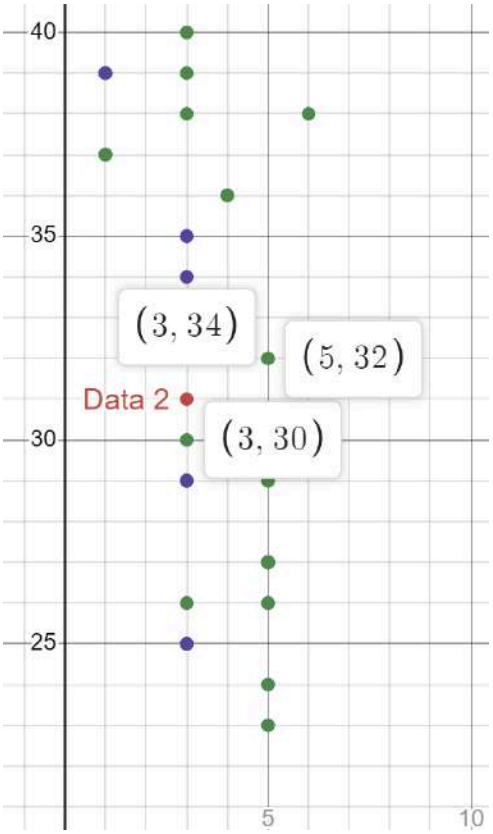
- Since we only used 2 data in a 2 dimensional graph, our first input Data will be "Data 1" with the coordinate of X and Y as follows : (3,51)
- We also set the parameters of K = 3, which means that we only use 3 nearest neighbor to determine the output of "Data 1"
- Notice that by the Image on the left, 3 data which have the closest distance from Data 1 are all purple color, so we predict that "Data 1" have an output "Income" value of ">50K".

2. **Second Data**

- Similar to "Data 1", Since we only used 2 data in a 2 dimensional graph, our first input Data will be "Data 2" with the coordinate of X and Y as follows : (3,30)
- We also set the parameters of K = 3, which means that we only use 3 nearest neighbor to determine the output of "Data 2"
- Notice that by the Image Above, the 3 data which have the closest distance from "Data 2" are 2 green color and 1 purple color. Since the quantity of the green color dot that is closest to "Data 2" is more than the quantity of the purple color dot that is closest to "Data 2". We predict that "Data 2" have an output "Income" value of "<=50K".

  Below are the clearer Image to determine the "Income" value of "Data 1" and "Data 2"





3. Additional method to determine the distance

Sometimes, just by looking, it's hard to tell which two points/dots are closest. So, we often use some mathematical methods to help us out. Two of the most popular methods are:

1. Euclidean Distance: This is a widely used method to figure out the distance between two points. It's based on the Pythagorean theorem, a fundamental principle in geometry that helps us calculate the direct distance between two points. In a 2 dimensional graph, the Euclidian Distance Formula used to find out the distance between point A (with coordinates x1, y1) and point B (with coordinates x2, y2) is as follows:

$$distance = \sqrt{(x2-x1)^2 + (y2-y1)^2}$$

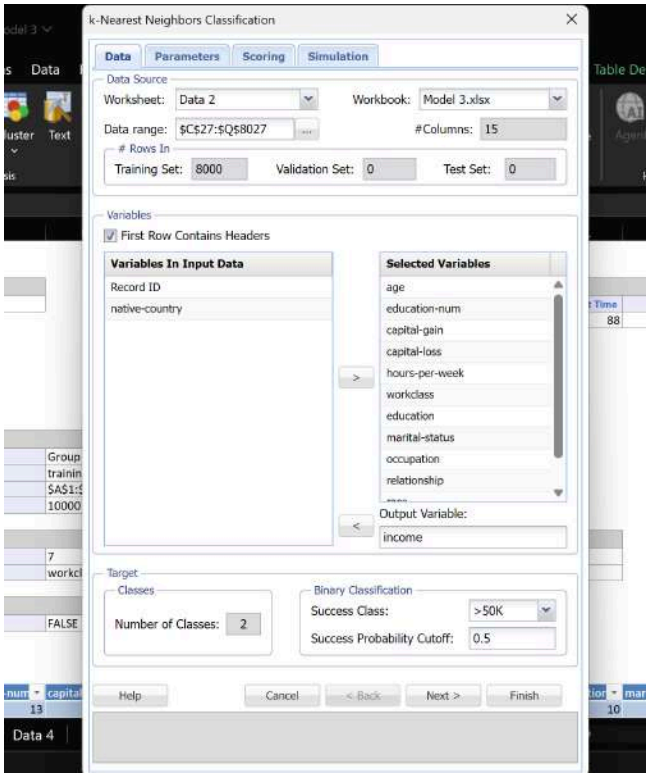
2. Manhattan Distance: Also known as the "city block" distance, the Manhattan distance is the sum of the absolute differences between the coordinates of the two points. In a 2-dimensional space, the Manhattan distance formula to calculate the distance between point A(x1, y1) and point B(x2, y2) is

$$Distance = |x2-x1| + |y2-y1|$$

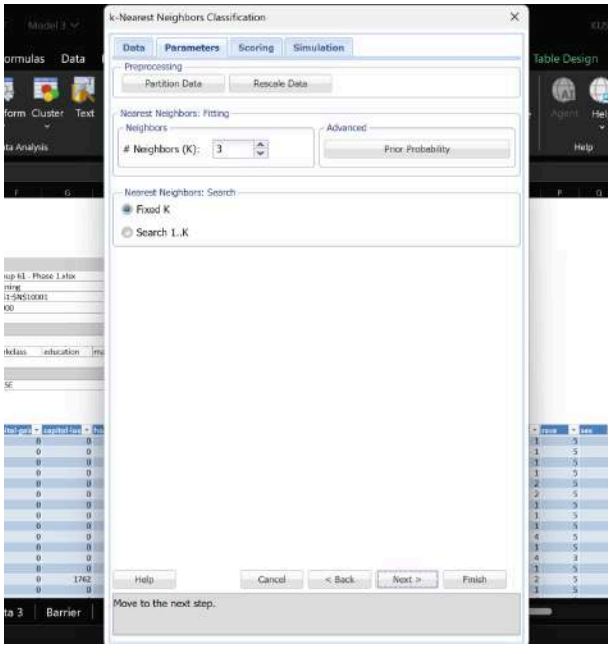
These methods are super handy, especially when we're dealing with more than just a flat surface. Not to mention in K nearest Neighbor Classifier, we're often dealing with more than just two attributes, so we're essentially working with a multi-dimensional graph.

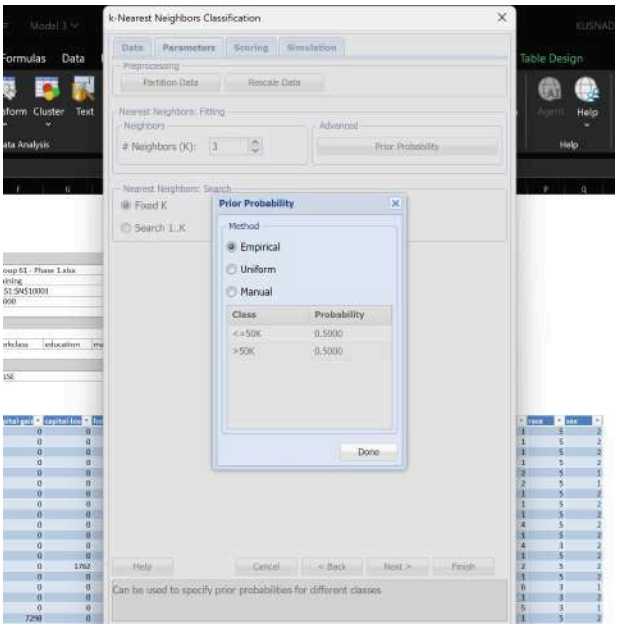
C. Prediction of Data 4

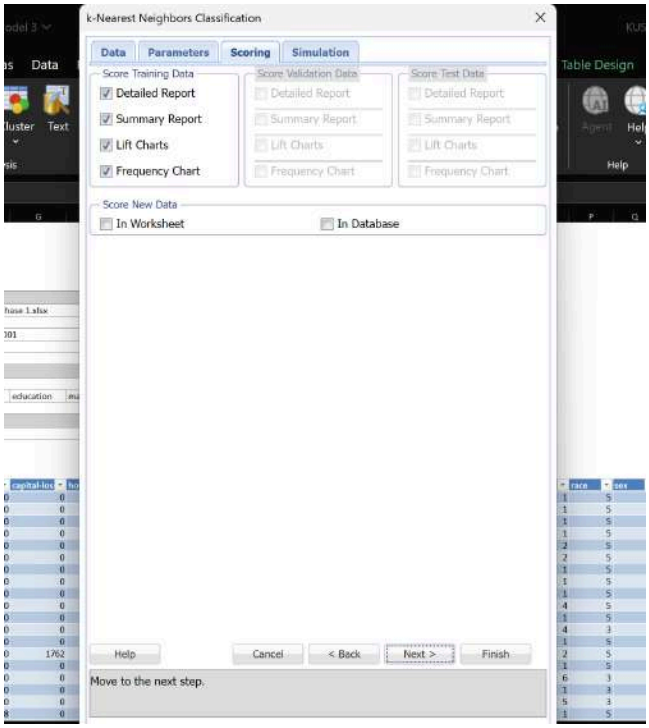
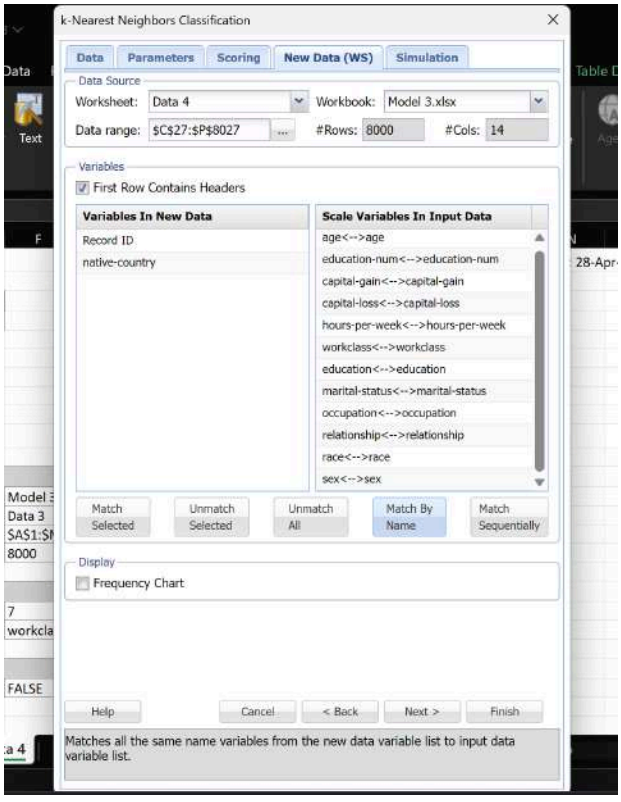
Now we are going to predict Data 4, which is the numerical "Test" Data sheet from phase 1. We will be using XLMiner to predict the data. Notice the steps are similar with the steps that is generated in phase 2. We just do a little modification from Phase 2. The following are the image of the modification steps to generate the prediction of Data 3 using K Nearest Neighbor Classifier with XLMiner:

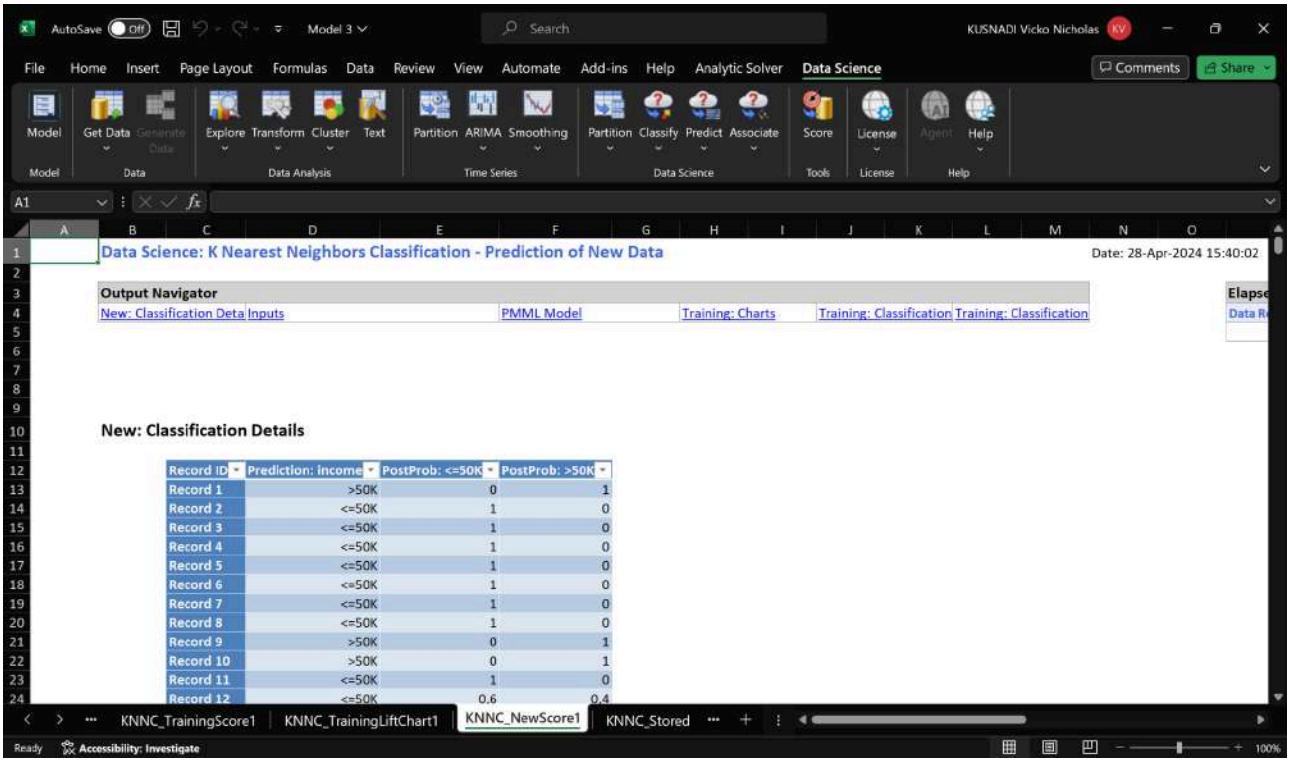
- In the Data 2 Sheet, select Data mining tab on Excel, and then select Classify, and choose K Nearest Neighbor as the Algorithm type for this model . In the Data Tab, set the data range from C27 to C8029. All variables except "native-country" and "Income" were selected as the Selected Variables. The "Income" variable was designated as the Output Variable. After that, press the Next button to go to the Parameter Tab	

- Moving to the Parameters Tab, the K Nearest Neighbor algorithm was configured with a value of K=3, indicating that the three nearest neighbors were considered for classification. The nearest neighbor search method was set to "Fixed K.". After that, press the prior probability button, and the Prior Probability Tab will appear, set the Prior Probability Empirical. Then, press the next button again to move to the Scoring tab





- In the Scoring Tab, all the checkboxes in the Score Training data section were selected to evaluate the model's performance on the training set. Next, in the scoring new data section, enable the In Worksheet option, to indicate the test data was present in the same worksheet as the training set.	
- After enabling the In Worksheet option, a new tab w, the New Data (WS) Tab, will appear to set up the test data. Change the worksheet to “Data 4”. Then select the data range from C27 to P8027. After that, press Match By Name button, then press Finish button.	
The result of Model 3 is as follows	



The result file of this Model will be sent on TXT file

D. Observation

Based on the result of this model, as we mentioned before, we manage to create a Model using K Nearest Neighbor (K= 3), and the following are the "unique" rules that must be applied on this Model:

- Rule 1:
- If the distance of a data to it 3 nearest data points (Neighbor) is define:
  - Check: If all 3 nearest data points have an "income" attribute value of ">50k"
    - Assign the data "Income" attribute value to ">50k"
  - Else: If not all 3 nearest data points have an "income" attribute value of ">50k"
    - Check: If the majority of nearest data points have an "income" attribute value of ">50k"
      - Assign the data "Income" attribute value to ">50k"
    - Otherwise, assign the data "Income" attribute value to "<=50k"
  - Else: go to rule number 2
- Rule 2:
- If the distance of a data to it 3 nearest data points (Neighbor) is define:

- ➢ Check: If all 3 nearest data points have an “income” attribute value of “<=50k”
  - ■ Assign the data “Income” attribute value to “<=50k”
- ➢ Else: If not all 3 nearest data points have an “income” attribute value of “<=50k”
  - ■ Check: If the majority of nearest data points have an “income” attribute value of “>50k”
    - ● Assign the data “Income” attribute value to “<=50k”
  - ■ Otherwise, assign the data “Income” attribute value to “>50k”
- ➢ Else: go to rule number 1

E. Conclusion

In summary, Model 3 utilized the K Nearest Neighbor algorithm applied to Data 2 for predicting income levels. The parameters used included the success class (>50K), number of classes (2), success probability cut-off (0.5), fixed K value (K=3), and prior probability set to empirical. The test set and training set variables were matched by name.

The evaluation of Model 3 involved analyzing its accuracy using the Confusion Matrix and Error Report. The Confusion Matrix provided a breakdown of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) to assess the model's performance. The sum of actual data values (TP+TN) indicated that the model achieved good accuracy.

The Error Report summarized the model's prediction mistakes, presenting the number of cases and errors for each income class, along with the percentage of errors. In the case of Model 3, the error rate was approximately 10.81%, indicating a relatively low margin of error. This suggests that the K Nearest Neighbor approach, with a K value of 3, performed well in predicting income levels using Data 2.

In conclusion, Model 3 demonstrated effective performance in predicting income levels using the K Nearest Neighbor algorithm. With an error rate of around 10.81%, the model exhibited good accuracy, making it a reliable tool for predicting income levels and aiding decision-making in relevant domains.

5. Model 4:

Recall that we have the following parameters for Model 4:

- Data: Data 2
- Type of Model: K Nearest Neighbor
- Parameter used:
  - Success Class: >50K
  - Number of Classes: 2
  - Success probability cut-off: 0.5
  - Fixed K with K = 5
  - Prior Probability to Empirical
  - Set the Test set and Training set variables to Match by Name

With this information, and the steps provided in the Design Report section, the following output will be provided by XLMiner.

A. Outputs and Observation

A.1. Error Model

Confusion Matrix		
Actual\Predicted	<=50K	>50K
<=50K	4273	609
>50K	455	2663

Error Report			
Class	# Cases	# Errors	% Error
<=50K	4882	609	12,47439574
>50K	3118	455	14,59268762
Overall	8000	1064	13,3

The figure table is the result of the accuracy of Model 4, according to the result. For the accuracy model, we use Confusion Matrix and also Error Report. This method is used to determine how accurate the result is based on the given input.

A.2. Confusion Matrix

A Confusion Matrix is a table that helps us understand the performance of our prediction model. It's called a "confusion" matrix because it shows how often our model is getting "confused" and making incorrect predictions.

- Here's how to construct it:
  - “Actual <=50K” and “Predicted <=50K”: This is the number of times our model correctly predicted that the data would be "<=50K". We call these True Positives (TP).
  - “Actual >50K” and “Predicted >50K”: This is the number of times our model correctly predicted that the data would be ">50K". We call these True Negatives (TN).
  - “Actual <=50K“ and “Predicted >50K”: This is the number of times our model incorrectly predicted that the data would be ">50K". We call these False Positives (FP).
  - “Actual >50K” and “Predicted <=50K”: This is the number of times our model incorrectly predicted that the data would be "<=50K". We call these False Negatives (FN).

The sum of “Actual” data values (TP+TN) tells us how often our model is making correct predictions. The sum of “Predicted” data values (FP+FN) tells us how often our model is making incorrect predictions.

In this case, the sum of “Actual” data values (TP+TN) is 4273+ 2663 which is 6936 and the sum of “Predicted” data values (FP+FN) is 455 + 609 which is 1064. The large value of the sum of “Actual” data values (TP+TN) indicates that the model is quite accurate.

A.3. Error Report

Error Report in the context of a prediction model is a summary of the mistakes made by the model during the prediction process. It’s a way to understand the performance of our model, whether our model accuracy is good or not.

- The following are the necessary terms and steps to construct Error Report:
  - First we need to create a table based on the image above
  - The term “#Cases” represents the total number of cases we used in the training data, whether the result of the income of each individual is “>50K” or “<=50K”
  - The term “#Error” represents the total number of cases we used in the training data that the prediction is actually false. For example, in case A, the output should be “<=50K” however, the prediction model

detect it as ">50K", therefore we put this data to the column of # error and the row of ">50K" indicating that it is an error and the output should be "<=50K"

- The term "%Error" indicates the error divided by the number of the case that is given the desired output times 100%. For example, "# Error" for ">50K" divided by "# Cases" ">50K" times 100% will be the result for "%Error" (455/3118 * 100% = 14,59268762%).

In our situation, the error is about 13.3 %. That means that about 13.3% of the time, our Model 4 doesn't get things right. This is actually a pretty good track record. It means that if we use this model to make predictions on new data, we can expect it to be off the mark about 13.3% of the time.

A.2.1. Model 3: K Nearest Neighbor Classifier

The following are the metrics based on K Nearest Neighbor Classifier:

Metrics	
Metric	Value
Accuracy (#correct)	7135
Accuracy (%correct)	89,1875
Specificity	0,89492
Sensitivity (Recall)	0,887107
Precision	0,84355
F1 score	0,86478
Success Class	>50K
Success Probability	0,5

B. Test if there is 2 raw output

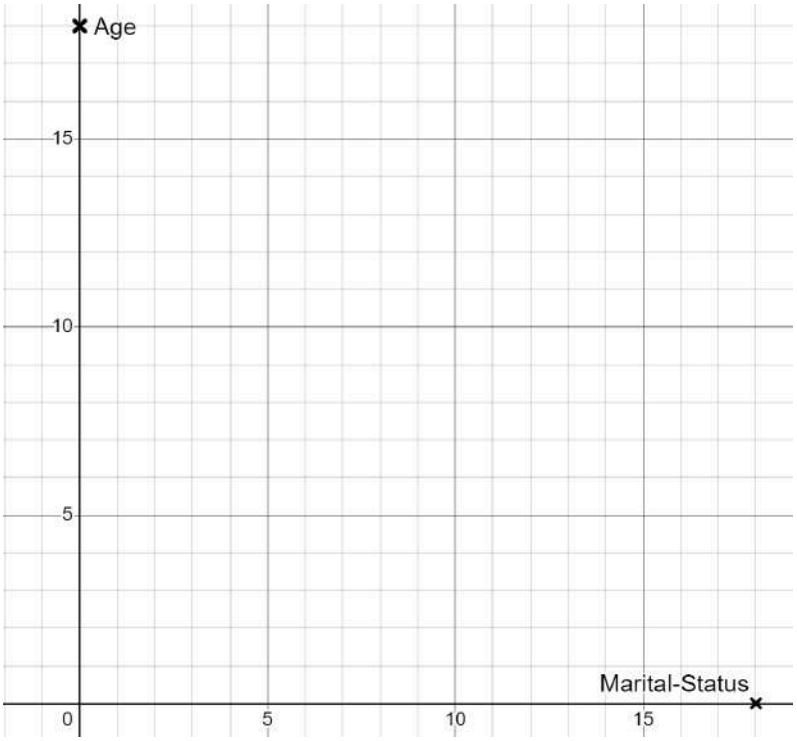
Based on the given output by K Nearest Neighbor Classifier we can predict wether a person have an income ">50K" or "<=50K". This prediction accuracy is around 89.19 % based on the error report, now given 2 new data of a person:

age	workclass	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country
51	Private	Prof-school	15	Married-civ-spouse	Prof-specialty	Husband	White	Male	0	0	50	United-States
30	Private	Masters	14	Never-married	Prof-specialty	Not-in-family	White	Male	0	0	40	Ireland

We can now predict what the income is for both people, whether it is ">50K" or "<=50K". However similar to model 3, in order to give an easy explanation, we will be using the transformation data (numerical data) as shown below:

age	education-num	capital-gain	capital-loss	hours-per-week	native-country	workclass	education	marital-	occupation	relationship	race	sex
51	15	0	0	50	United-States	3	15	3	10	1	5	2
30	14	0	0	40	Ireland	3	13	5	10	2	5	2

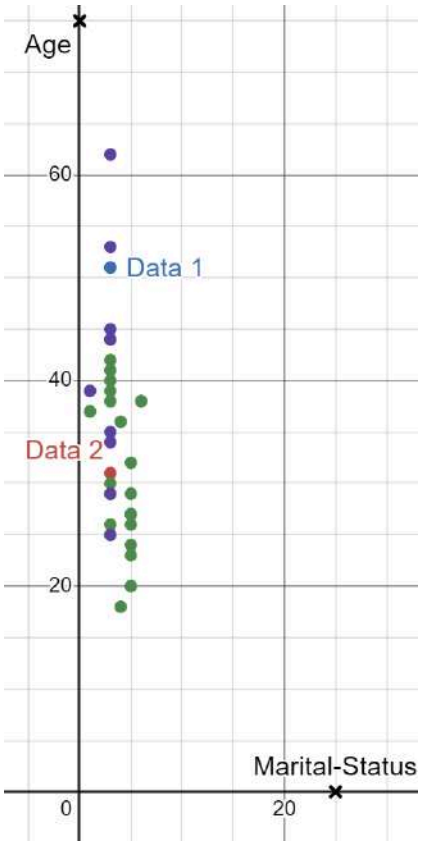
We will also use a 2 dimensional graph assumption for the steps required for generating the result of K Nearest Neighbor Classifier. Since we are using 2 dimensional graph, we could only take 2 attributes as an input. In this case we are going to use "Marital status" attribute as the X component of a cartesian graph and "age" attribute as the Y component of a cartesian graph, just like the image below:



Lastly, we are also going to use the first 40 given data from phase 2, the following are the first 40 given data based on Phase 2 design report:

Record ID	marital-status	age	income	Record 5	5	27	<=50K
Record 1	3	26	<=50K	Record 6	5	29	<=50K
Record 2	3	41	<=50K	Record 7	3	30	<=50K
Record 3	3	42	<=50K	Record 8	3	44	>50K
Record 4	3	44	<=50K	Record 9	3	44	>50K

Record 10	5	27	<=50K	Record 26	5	32	<=50K
Record 11	3	25	>50K	Record 27	3	51	>50K
Record 12	4	18	<=50K	Record 28	5	20	<=50K
Record 13	3	62	>50K	Record 29	5	23	<=50K
Record 14	1	39	<=50K	Record 30	1	37	<=50K
Record 15	3	25	<=50K	Record 31	3	39	<=50K
Record 16	3	40	<=50K	Record 32	6	38	<=50K
Record 17	3	53	>50K	Record 33	5	20	<=50K
Record 18	5	26	<=50K	Record 34	5	27	<=50K
Record 19	3	29	>50K	Record 35	3	45	>50K
Record 20	3	35	>50K	Record 36	3	38	<=50K
Record 21	5	26	<=50K	Record 37	3	29	<=50K
Record 22	1	37	<=50K	Record 38	3	51	<=50K
Record 23	5	24	<=50K	Record 39	4	36	<=50K
Record 24	3	34	>50K	Record 40	1	39	>50
Record 25	3	51	>50K				



The Image on the left are the 2 dimensional graphical vision. The purple dot represents the data that have an output "Income" of ">50K" and the green dot represent the data that have an output "Income" of "<=50K" The following are the steps for predicting the income based on the given Model, Model 3:

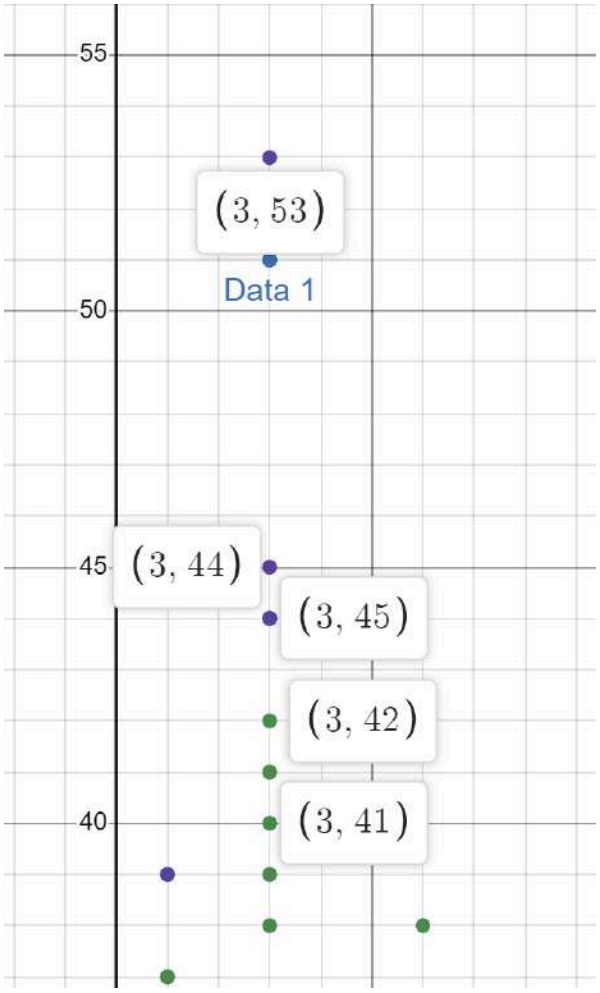
4. **First Data**

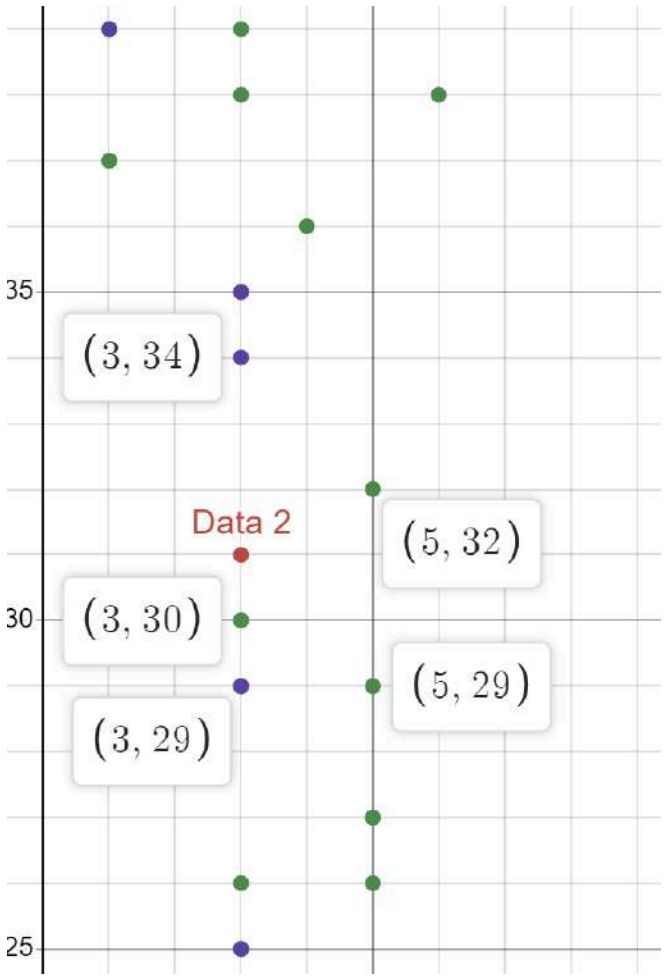
- Since we only used 2 data in a 2 dimensional graph, our first input Data will be "Data 1" with the coordinate of X and Y as follows : (3,51)
- We also set the parameters of K = 5, which means that we only use 5 nearest neighbor to determine the output of "Data 1"
- Notice that by the Image on the left, 5 data which have the closest distance from Data 1 are 3 purple and 2 green. Since the quantity of the green color dot that is closest to "Data 1" is less than the quantity of the purple color dot that is closest to "Data 1",so we predict that Data 1 have an output "Income" value of ">50K".

5. **Second Data**

- Similar to "Data 1", Since we only used 2 data in a 2 dimensional graph, our first input Data will be "Data 2" with the coordinate of X and Y as follows : (3,30)
- We also set the parameters of K = 5, which means that we use 5 nearest neighbor to determine the output of "Data 2"
- Notice that by the Image Above, the 5 data which have the closest distance from "Data 2" are 3 green color and 2 purple color. Since the quantity of the green color dot that is closest to "Data 2" is more than the quantity of the purple color dot that is closest to "Data 2". We predict that "Data 2" have an output "Income" value of "<=50K".

  Below are the clearer Image to determine the "Income" value of "Data 1" and "Data 2"





6. Additional method to determine the distance

Sometimes, just by looking, it's hard to tell which two points/dots are closest. So, we often use some mathematical methods to help us out. Two of the most popular methods are:

3. Euclidean Distance: This is a widely used method to figure out the distance between two points. It's based on the Pythagorean theorem, a fundamental principle in geometry that helps us calculate the direct distance between two points. In a 2 dimensional graph, the Euclidian Distance Formula used to find out the distance between point A (with coordinates x1, y1) and point B (with coordinates x2, y2) is as follows:

$$distance = \sqrt{(x2-x1)^2 + (y2-y1)^2}$$

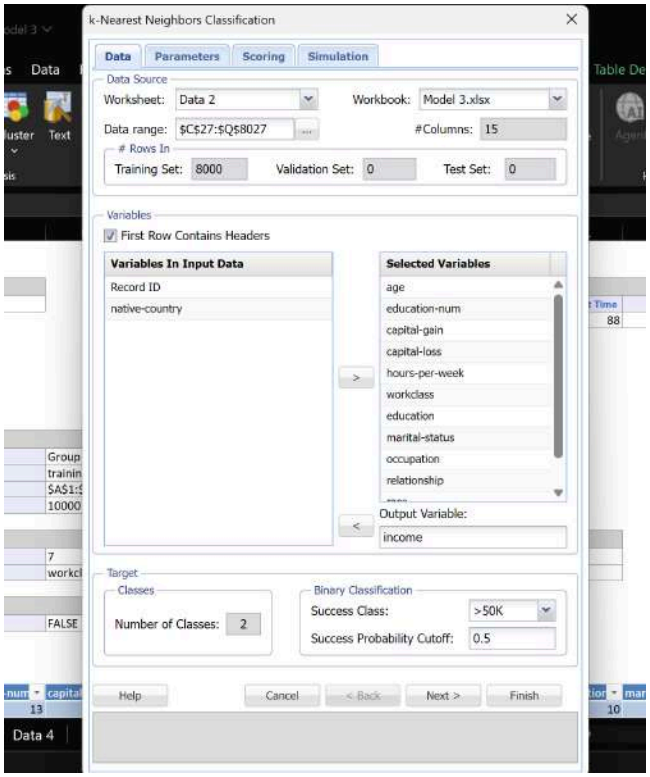
4. Manhattan Distance: Also known as the "city block" distance, the Manhattan distance is the sum of the absolute differences between the coordinates of the two points. In a 2-dimensional space, the Manhattan distance formula to calculate the distance between point A(x1, y1) and point B(x2, y2) is

$$Distance = |x2 - x1| + |y2 - y1|$$

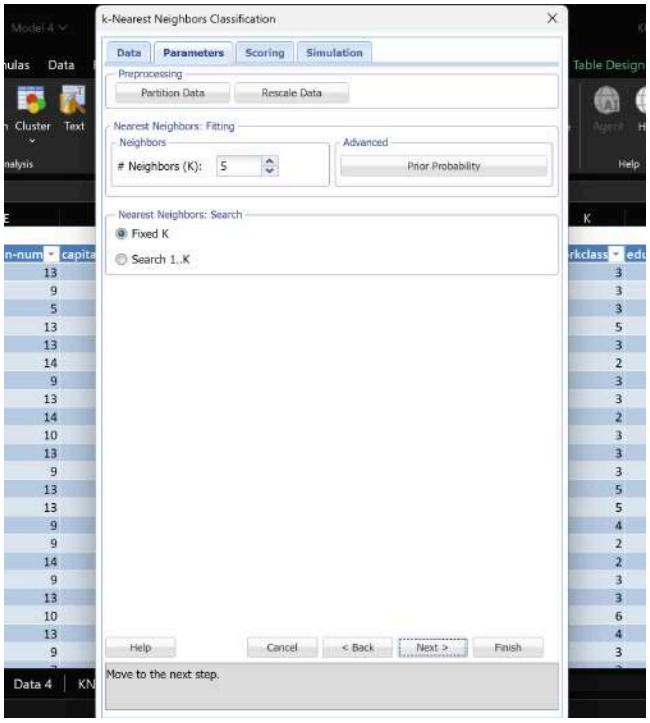
These methods are super handy, especially when we're dealing with more than just a flat surface. Not to mention in K nearest Neighbor Classifier, we're often dealing with more than just two attributes, so we're essentially working with a multi-dimensional graph.

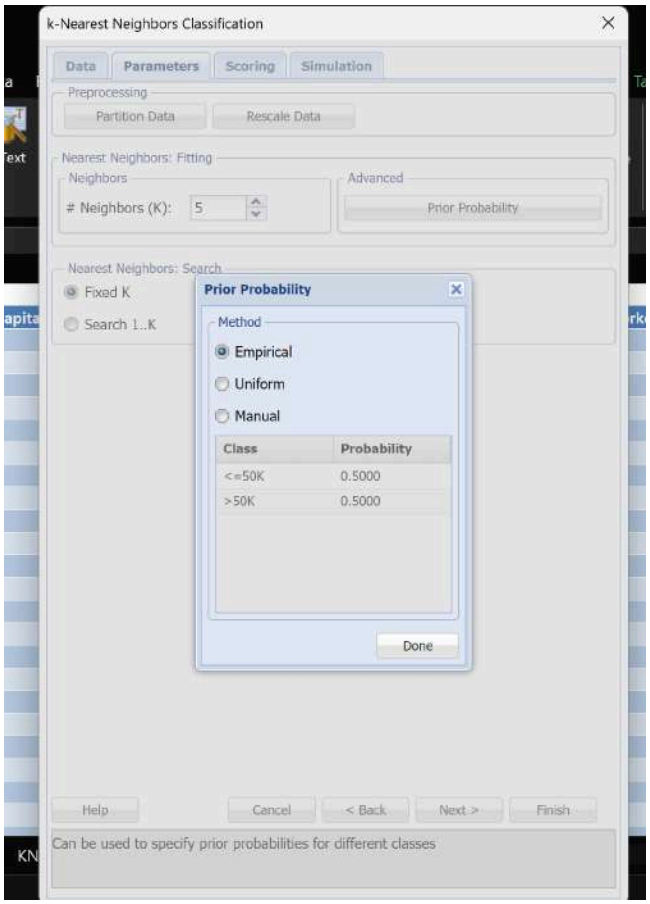
C. Prediction of Data 4

Now we are going to predict Data 4, which is the numerical "Test" Data sheet from phase 1. We will be using XLMiner to predict the data. Notice the steps are similar with the steps that is generated in phase 2. We just do a little modification from Phase 2. The following are the image of the modification steps to generate the prediction of Data 3 using K Nearest Neighbor Classifier with XLMiner:

- In the Data 2 Sheet, select Data mining tab on Excel, and then select Classify, and choose K Nearest Neighbor as the Algorithm type for this model . In the Data Tab, set the data range from C27 to C8029. All variables except "native-country" and "Income" were selected as the Selected Variables. The "Income" variable was designated as the Output Variable. After that, press the Next button to go to the Parameter Tab	

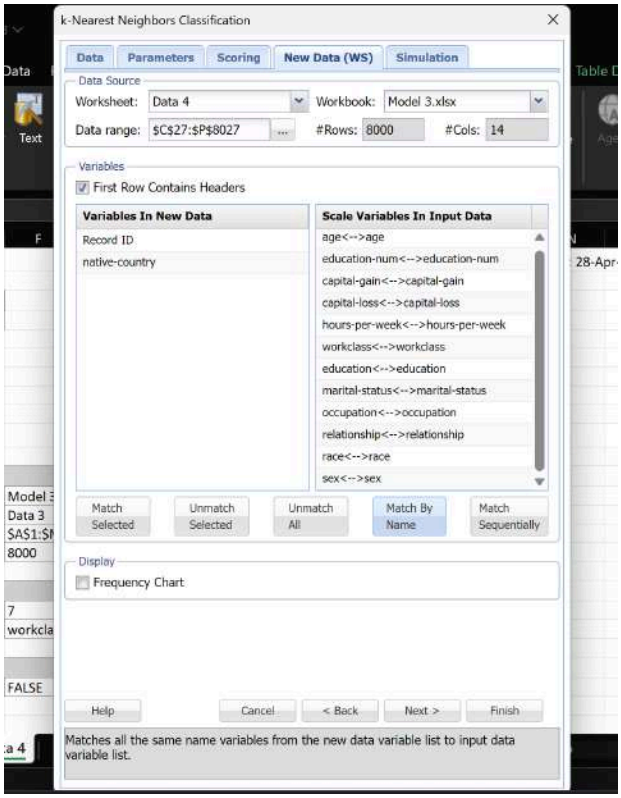
- Moving to the Parameters Tab, the K Nearest Neighbor algorithm was configured with a value of K=5, indicating that the three nearest neighbors were considered for classification. The nearest neighbor search method was set to "Fixed K.". After that, press the prior probability button, and the Prior Probability Tab will appear, set the Prior Probability Empirical. Then, press the next button again to move to the Scoring tab



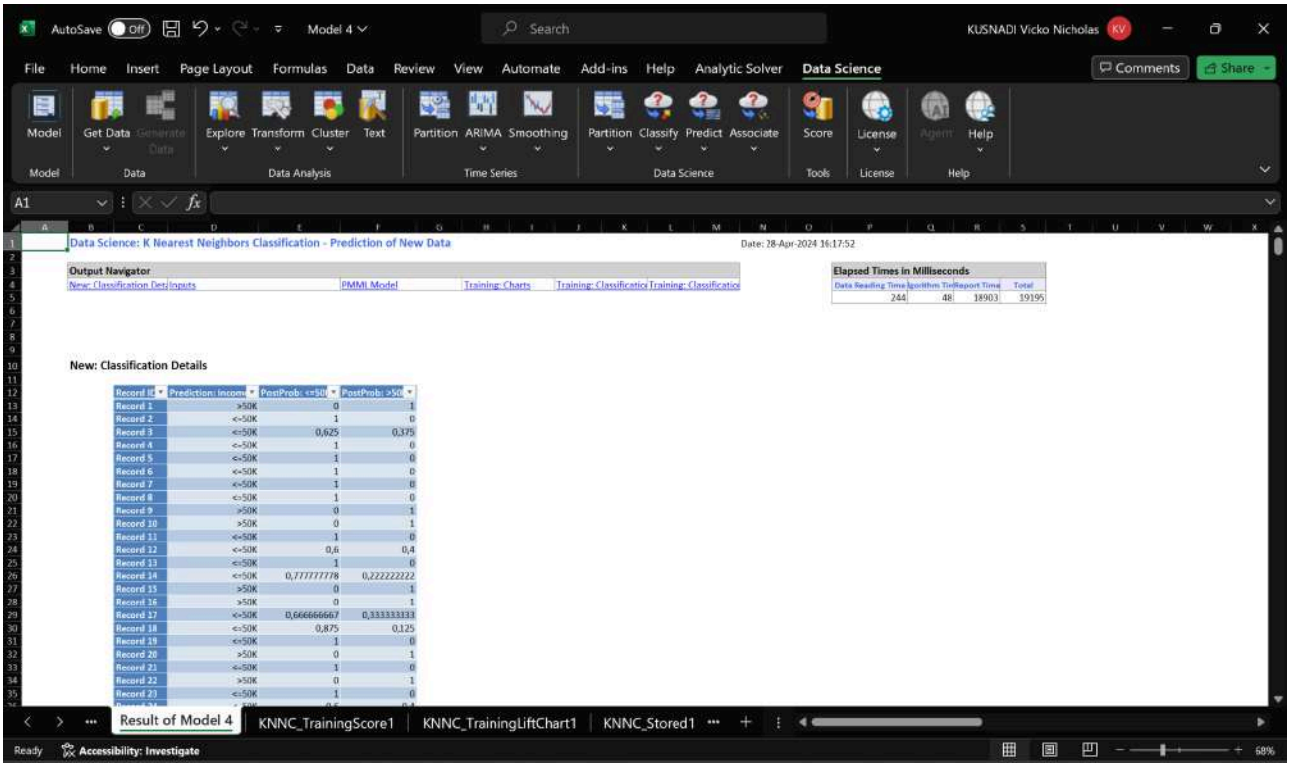


- In the Scoring Tab, all the checkboxes in the Score Training data section were selected to evaluate the model's performance on the training set. Next, in the scoring new data section, enable the In Worksheet option, to indicate the test data was present in the same worksheet as the training set.

- After enabling the In Worksheet option, a new tab w, the New Data (WS) Tab, will appear to set up the test data. Change the worksheet to “Data 4”. Then select the data range from C27 to P8027. After that, press Match By Name button, then press Finish button.



The result of Model 4 is as follows



The result file of this Model will be sent on TXT file

D. Observation

Based on the result of this model, as we mentioned before, we manage to create a Model using K Nearest Neighbor (K= 5), and the following are the "unique" rules that must be applied on this Model:

- Rule 1:
- If the distance of a data to it 5 nearest data points (Neighbor) is define:
  - Check: If all 5 nearest data points have an "income" attribute value of ">50k"
    - Assign the data "Income" attribute value to ">50k"
  - Else: If not all 5 nearest data points have an "income" attribute value of ">50k"
    - Check: If the majority of nearest data points have an "income" attribute value of ">50k"
      - Assign the data "Income" attribute value to ">50k"
    - Otherwise, assign the data "Income" attribute value to "<=50k"
  - Else: go to rule number 2
- Rule 2:
- If the distance of a data to it 5 nearest data points (Neighbor) is define:
  - Check: If all 5 nearest data points have an "income" attribute value of "<=50k"
    - Assign the data "Income" attribute value to "<=50k"
  - Else: If not all 5 nearest data points have an "income" attribute value of "<=50k"
    - Check: If the majority of nearest data points have an "income" attribute value of ">50k"
      - Assign the data "Income" attribute value to "<=50k"
    - Otherwise, assign the data "Income" attribute value to ">50k"
  - Else: go to rule number 1

E. Conclusion

In summary, Model 4 utilized the K Nearest Neighbor algorithm applied to Data 2 for predicting income levels. The parameters used included the success class (>50K), number of classes (2), success probability cut-off (0.5), fixed K value (K=5), and prior probability set to empirical. The test set and training set variables were matched by name.

The evaluation of Model 4 involved analyzing its accuracy using the Confusion Matrix and Error Report. The Confusion Matrix provided a breakdown of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) to assess the model's performance. The sum of actual data values (TP+TN) indicated that the model achieved good accuracy.

The Error Report summarized the model's prediction mistakes, presenting the number of cases and errors for each income class, along with the percentage of errors. In the case of Model 4, the error rate was approximately 13.3%, suggesting a reasonably low margin of error. This implies that the K Nearest Neighbor algorithm, with a K value of 5, performed well in predicting income levels using Data 2.

In conclusion, Model 4 demonstrated effective performance in predicting income levels using the K Nearest Neighbor algorithm. With an error rate of around 13.3%, the model exhibited satisfactory accuracy, making it a reliable tool for predicting income levels and aiding decision-making in relevant domains.

6. Model 5:

Recall that we have the following parameters for Model 5:

- Data: Data 2
- Type of Model: K Nearest Neighbor
- Parameter used:
  - Success Class: >50K
  - Number of Classes: 2
  - Success probability cut-off: 0.5
  - Fixed K with K = 10
  - Prior Probability to Empirical
  - Set the Test set and Training set variables to Match by Name

With this information, and the steps provided in the Design Report section, the following output will be provided by XLMiner.

A. Outputs and Observation

A.1. Error Model

Confusion Matrix		
Actual\Predicted	<=50K	>50K
<=50K	4149	733
>50K	493	2625

Error Report			
Class	# Cases	# Errors	% Error
<=50K	4882	733	15,01433839
>50K	3118	493	15,81141758
Overall	8000	1226	15,325

The figure table is the result of the accuracy of Model 5, according to the result. For the accuracy model, we use Confusion Matrix and also Error Report. This method is used to determine how accurate the result is based on the given input.

A.2. Confusion Matrix

A Confusion Matrix is a table that helps us understand the performance of our prediction model. It's called a "confusion" matrix because it shows how often our model is getting "confused" and making incorrect predictions.

- Here's how to construct it:
  - “Actual <=50K” and “Predicted <=50K”: This is the number of times our model correctly predicted that the data would be "<=50K". We call these True Positives (TP).
  - “Actual >50K” and “Predicted >50K”: This is the number of times our model correctly predicted that the data would be ">50K". We call these True Negatives (TN).
  - “Actual <=50K“ and “Predicted >50K”: This is the number of times our model incorrectly predicted that the data would be ">50K". We call these False Positives (FP).
  - “Actual >50K” and “Predicted <=50K”: This is the number of times our model incorrectly predicted that the data would be "<=50K". We call these False Negatives (FN).

The sum of “Actual” data values (TP+TN) tells us how often our model is making correct predictions. The sum of “Predicted” data values (FP+FN) tells us how often our model is making incorrect predictions.

In this case, the sum of “Actual” data values (TP+TN) is 4149 + 2625 which is 6774 and the sum of “Predicted” data values (FP+FN) is 733 + 493 which is 1226 The large value of the sum of “Actual” data values (TP+TN) indicates that the model is quite accurate.

A.3. Error Report

Error Report in the context of a prediction model is a summary of the mistakes made by the model during the prediction process. It’s a way to understand the performance of our model, whether our model accuracy is good or not.

- The following are the necessary terms and steps to construct Error Report:
  - First we need to create a table based on the image above

- The term "#Cases" represents the total number of cases we used in the training data, whether the result of the income of each individual is ">50K" or "<=50K"
- The term "#Error" represents the total number of cases we used in the training data that the prediction is actually false. For example, in case A, the output should be "<=50K" however, the prediction model detect it as ">50K", therefore we put this data to the column of # error and the row of ">50K" indicating that it is an error and the output should be "<=50K"
- The term "%Error" indicates the error divided by the number of the case that is given the desired output times 100%. For example, "# Error" for ">50K" divided by "# Cases" ">50K" times 100% will be the result for "%Error" (928/3118 * 100% = 29,76266838%).

In our situation, the error is about *insert data*. That means that about 15.32% of the time, our Model 5 doesn't get things right. This is actually a pretty good track record. It means that if we use this model to make predictions on new data, we can expect it to be off the mark about 15.32% of the time.

A.2.1. Model 3: K Nearest Neighbor Classifier

The following are the metrics based on K Nearest Neighbor Classifier:

Metrics	
Metric	Value
Accuracy (#correct)	7135
Accuracy (%correct)	89,1875
Specificity	0,89492
Sensitivity (Recall)	0,887107
Precision	0,84355
F1 score	0,86478
Success Class	>50K
Success Probability	0,5

B. Test if there is 2 raw output

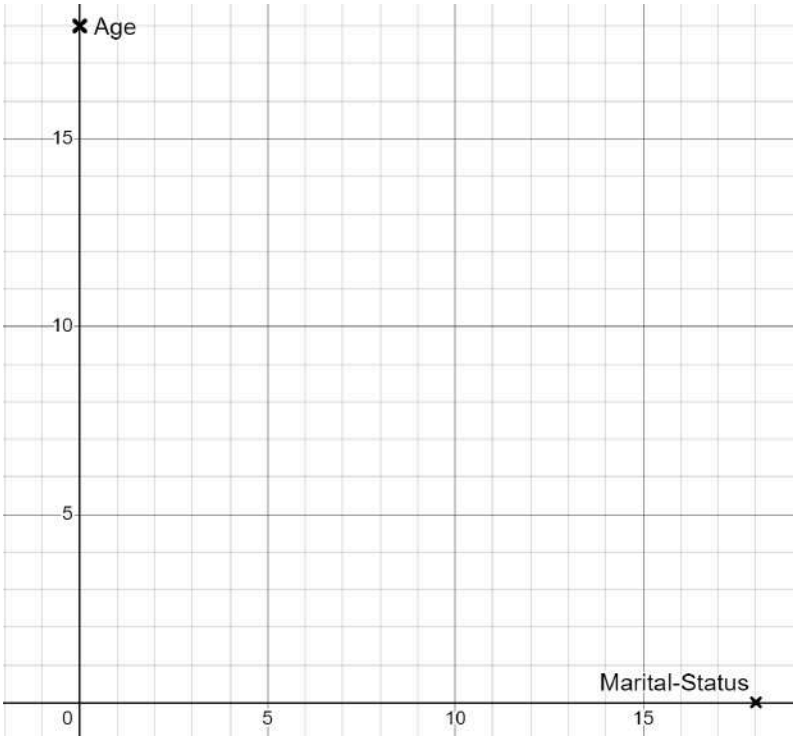
Based on the given output by K Nearest Neighbor Classifier we can predict wether a person have an income ">50K" or "<=50K". This prediction accuracy is around 89.19 % based on the error report, now given 2 new data of a person:

age	workclass	education	education-num	marital-status	occupation	relationship	race	sex	capital-gain	capital-loss	hours-per-week	native-country
51	Private	Prof-school	15	Married-civ-spouse	Prof-specialty	Husband	White	Male	0	0	50	United-States
30	Private	Masters	14	Never-married	Prof-specialty	Not-in-family	White	Male	0	0	40	Ireland

We can now predict what the income is for both people, whether it is ">50K" or "<=50K". However similar to model 3, in order to give an easy explanation, we will be using the transformation data (numerical data) as shown below:

age	education-num	capital-gain	capital-loss	hours-per-week	native-country	workclass	education	marital-	occupation	relationship	race	sex
51	15	0	0	50	United-States	3	15	3	10	1	5	2
30	14	0	0	40	Ireland	3	13	5	10	2	5	2

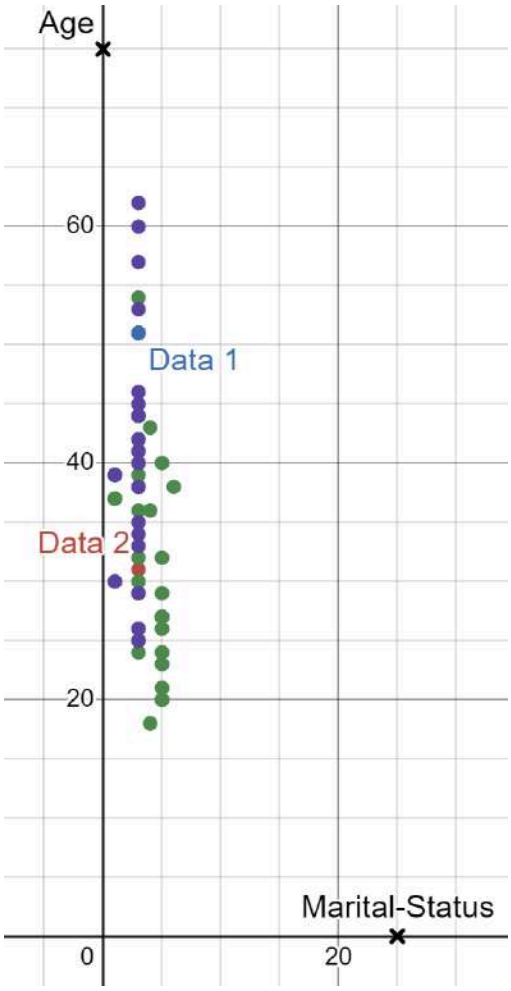
We will also use a 2 dimensional graph assumption for the steps required for generating the result of K Nearest Neighbor Classifier. Since we are using a 2 dimensional graph, we could only take 2 attributes as an input. In this case we are going to use "Marital status" attribute as the X component of a cartesian graph and "age" attribute as the Y component of a cartesian graph, just like the image below:



Lastly, different from Model 3 and Model 4, we are going to use the first 60 given data from phase 2, this because the increase number of K. the following are the first 60 given data based on Phase 2 design report:

Record ID	marital-status	age	income
Record 1	3	26	<=50K

Record 2	3	41	<=50K	Record 33	5	20	<=50K
Record 3	3	42	<=50K	Record 34	5	27	<=50K
Record 4	3	44	<=50K	Record 35	3	45	>50K
Record 5	5	27	<=50K	Record 36	3	38	<=50K
Record 6	5	29	<=50K	Record 37	3	29	<=50K
Record 7	3	30	<=50K	Record 38	3	51	<=50K
Record 8	3	44	>50K	Record 39	4	36	<=50K
Record 9	3	44	>50K	Record 40	1	39	>50K
Record 10	5	27	<=50K	Record 41	3	41	>50K
Record 11	3	25	>50K	Record 42	4	43	<=50K
Record 12	4	18	<=50K	Record 43	3	44	>50K
Record 13	3	62	>50K	Record 44	3	32	<=50K
Record 14	1	39	<=50K	Record 45	5	21	<=50K
Record 15	3	25	<=50K	Record 46	3	24	<=50K
Record 16	3	40	<=50K	Record 47	3	38	>50K
Record 17	3	53	>50K	Record 48	3	60	>50K
Record 18	5	26	<=50K	Record 49	3	42	>50K
Record 19	3	29	>50K	Record 50	3	36	<=50K
Record 20	3	35	>50K	Record 51	3	51	<=50K
Record 21	5	26	<=50K	Record 52	3	25	<=50K
Record 22	1	37	<=50K	Record 53	3	26	>50K
Record 23	5	24	<=50K	Record 54	3	46	>50K
Record 24	3	34	>50K	Record 55	3	57	>50K
Record 25	3	51	>50K	Record 56	3	54	<=50K
Record 26	5	32	<=50K	Record 57	5	40	<=50K
Record 27	3	51	>50K	Record 58	1	30	>50K
Record 28	5	20	<=50K	Record 59	3	40	>50K
Record 29	5	23	<=50K	Record 60	3	33	>50K
Record 30	1	37	<=50K				
Record 31	3	39	<=50K				
Record 32	6	38	<=50K				



The Image on the left are the 2 dimensional graphical vision. The purple dot represents the data that have an output "Income" of ">50K" and the green dot represent the data that have an output "Income" of "<=50K" The following are the steps for predicting the income based on the given Model, Model 3:

1. **First Data**

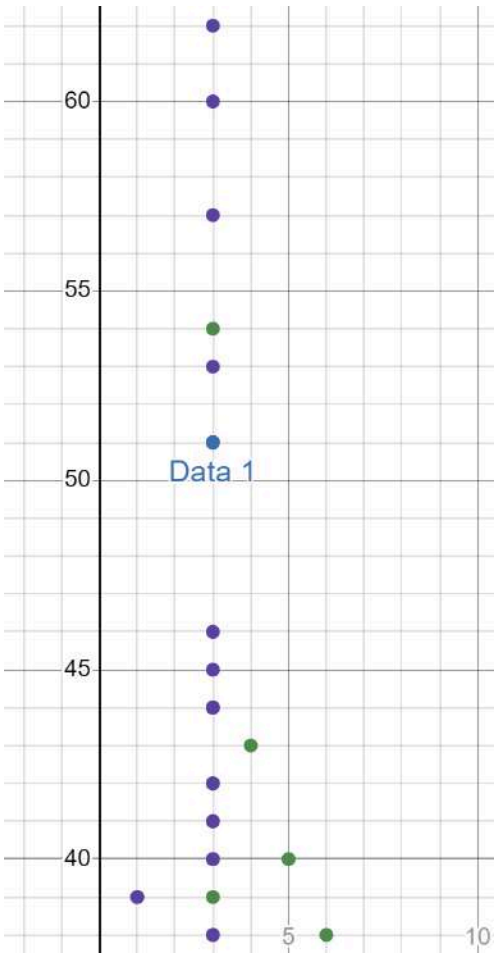
- Since we only used 2 data in a 2 dimensional graph, our first input Data will be "Data 1" with the coordinate of X and Y as follows : (3,51)
- We also set the parameters of K = 10, which means that we will use 10 nearest neighbor to determine the output of "Data 1"
- Notice that by the Image on the left, 10 data which have the closest distance from Data 1 are 8 purple and 2 green. Since the quantity of the green color dot that is closest to "Data 1" is less than the quantity of the purple color dot that is closest to "Data 1",so we predict that Data 1 have an output "Income" value of ">50K".

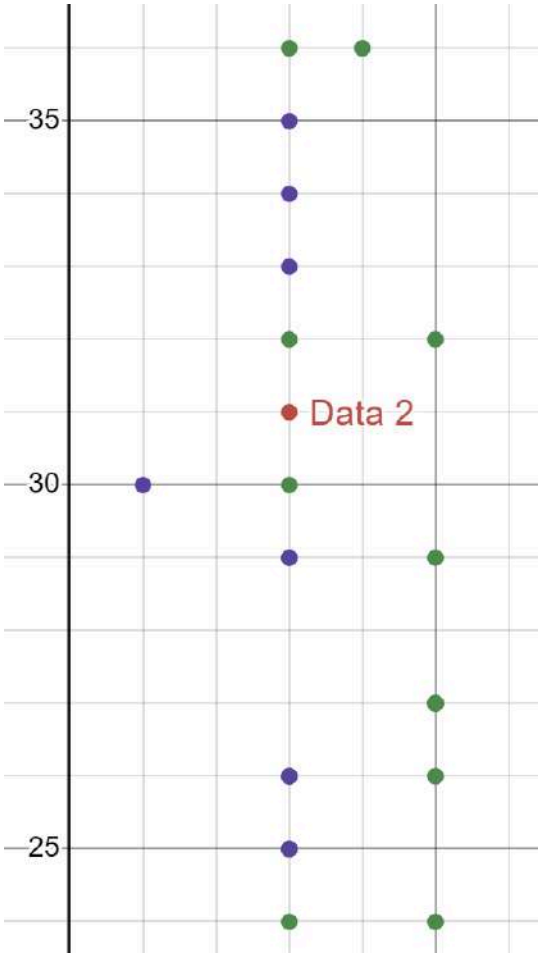
2. **Second Data**

- Similar to "Data 1", Since we only used 2 data in a 2 dimensional graph, our first input Data will be "Data 2" with the coordinate of X and Y as follows : (3,30)
- We also set the parameters of K = 10, which means that we use 10 nearest neighbor to determine the output of "Data 2"

- Notice that by the Image Above, the 10 data which have the closest distance from “Data 2” are 5 green color and 5 purple color. Since the quantity of the green color dot that is closest to “Data 2” is equal to the quantity of the purple color dot that is closest to “Data 2”. We cannot determine the value of the Income. Therefore, one of the solution is we decrease the value of K to 9. With the value of K = 9, we predict that “Data 2” have an output “Income” value of “>50K”. This because the 9 data which have the closest distance from “Data 2” are 4 green color and 5 purple color.

  Below are the clearer Image to determine the “Income” value of “Data 1” and “Data 2”





3. Additional method to determine the distance

Sometimes, just by looking, it’s hard to tell which two points/dots are closest. So, we often use some mathematical methods to help us out. Two of the most popular methods are:

1. Euclidean Distance: This is a widely used method to figure out the distance between two points. It's based on the Pythagorean theorem, a fundamental principle in geometry that helps us calculate the direct distance between two points. In a 2 dimensional graph, the Euclidian Distance Formula used to find out the distance between point A (with coordinates x1, y1) and point B (with coordinates x2, y2) is as follows:

$$distance = \sqrt{(x2-x1)^2 + (y2-y1)^2}$$

2. Manhattan Distance: Also known as the "city block" distance, the Manhattan distance is the sum of the absolute differences between the coordinates of the two points. In a 2-dimensional space, the Manhattan distance formula to calculate the distance between point A(x1, y1) and point B(x2, y2) is

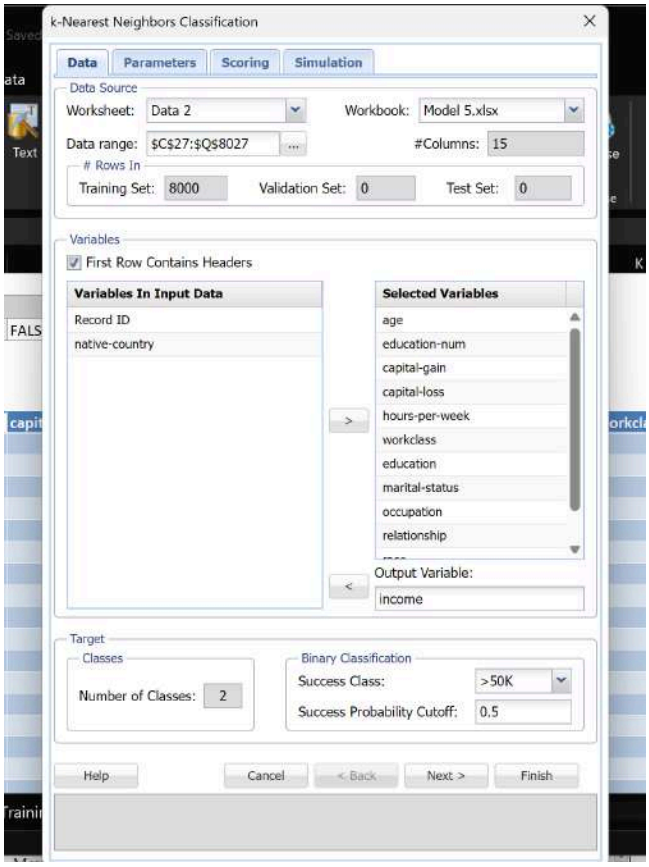
$$Distance = |x2-x1| + |y2-y1|$$

These methods are super handy, especially when we're dealing with more than just a flat surface. Not to mention in K nearest Neighbor Classifier, we're often dealing with more than just two attributes, so we're essentially working with a multi-dimensional graph.

C. Prediction of Data 4

Now we are going to predict Data 4, which is the numerical "Test" Data sheet from phase 1. We will be using XLMiner to predict the data. Notice the steps are similar with the steps that is generated in phase 2. We just do a little modification from Phase 2. The following are the image of the modification steps to generate the prediction of Data 3 using K Nearest Neighbor Classifier with XLMiner:

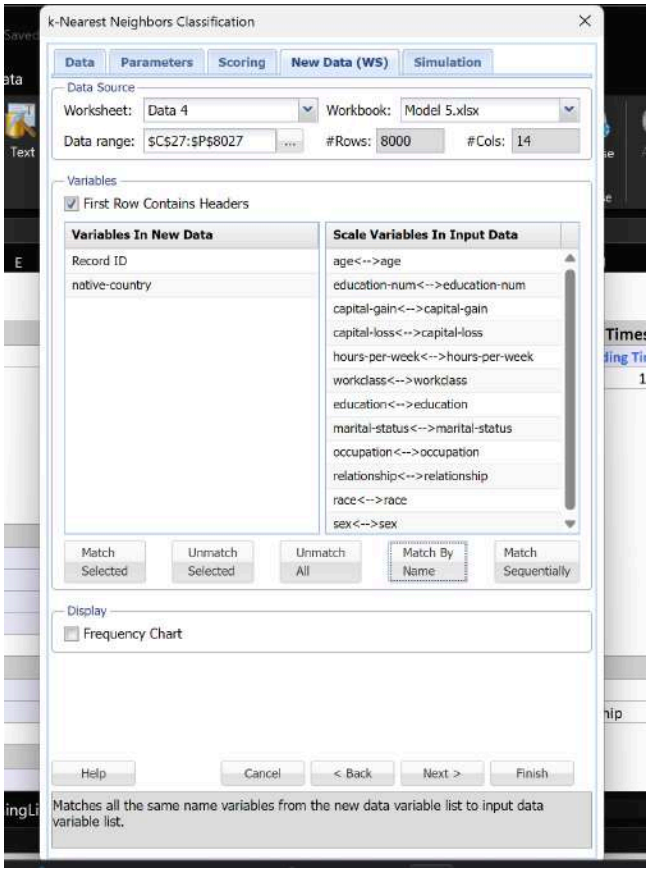
- In the Data 2 Sheet, select Data mining tab on Excel, and then select Classify, and choose K Nearest Neighbor as the Algorithm type for this model . In the Data Tab, set the data range from C27 to C8029. All variables except "native-country" and "Income" were selected as the Selected Variables. The "Income" variable was designated as the Output Variable. After that, press the Next button to go to the Parameter Tab



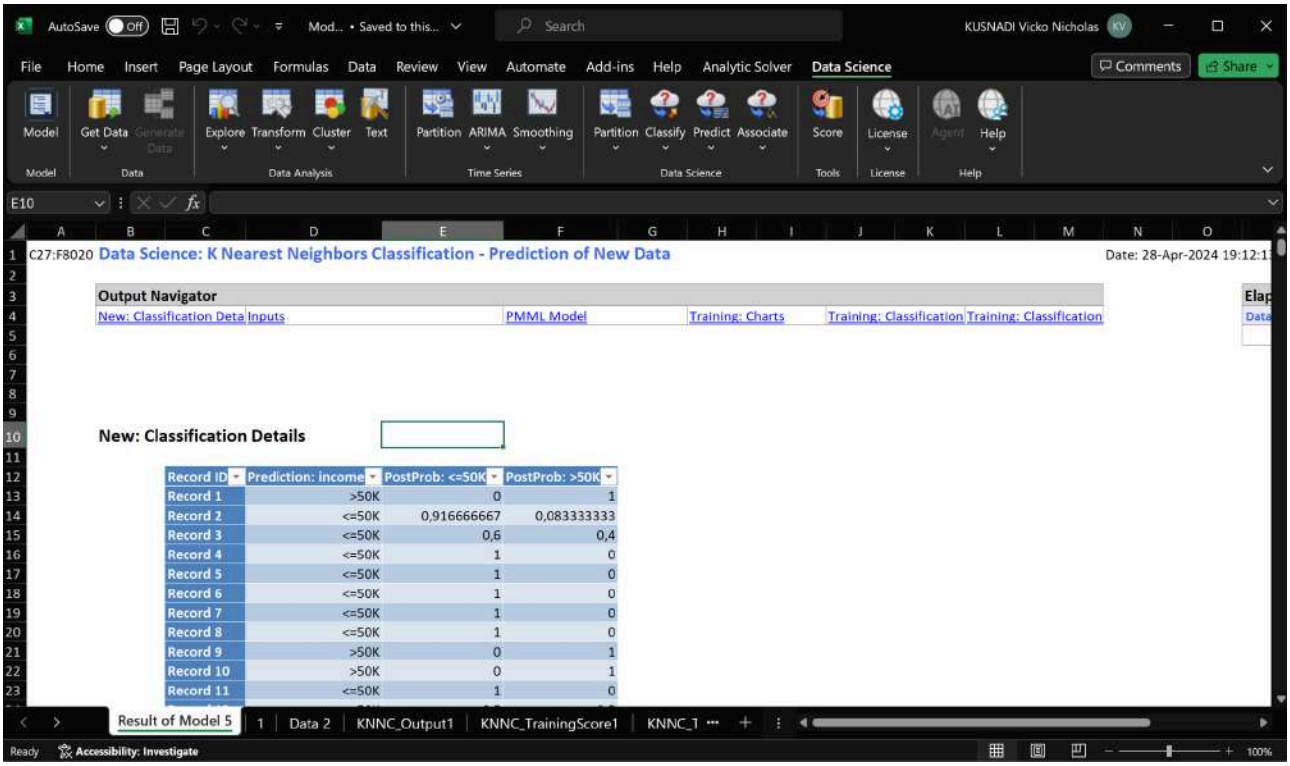
- Moving to the Parameters Tab, the K Nearest Neighbor algorithm was configured with a value of K=10, indicating that the three nearest neighbors were considered for classification. The nearest neighbor search method was set to "Fixed K.". After that, press the prior probability button, and the Prior Probability Tab will appear, set the Prior Probability Empirical. Then, press the next button again to move to the Scoring tab	

- In the Scoring Tab, all the checkboxes in the Score Training data section were selected to evaluate the model's performance on the training set. Next, in the scoring new data section, enable the In Worksheet option, to indicate the test data was present in the same worksheet as the training set.

- After enabling the In Worksheet option, a new tab w, the New Data (WS) Tab, will appear to set up the test data. Change the worksheet to “Data 4”. Then select the data range from C27 to P8027. After that, press Match By Name button, then press Finish button.



The result of Model 5 is as follows



The result file of this Model will be sent on TXT file

D. Observation

Based on the result of this model, as we mentioned before, we manage to create a Model using K Nearest Neighbor (K= 10), and the following are the "unique" rules that must be applied on this Model:

- Rule 1:
- If the distance of a data to it 10 nearest data points (Neighbor) is define:
  - Check: If all 10 nearest data points have an "income" attribute value of ">50k"
    - Assign the data "Income" attribute value to ">50k"
  - Else: If not all 10 nearest data points have an "income" attribute value of ">50k"
    - Check: If the majority of nearest data points have an "income" attribute value of ">50k"
      - Assign the data "Income" attribute value to ">50k"
    - Otherwise, If the majority of nearest data points have an "income" attribute value of "<=50k"
      - Assign the data "Income" attribute value to "<=50k"
    - Otherwise, go to either rule 2 or rule 1 with K = K - 1

- ➢ Else: go to rule number 2

- ❖ Rule 2:
- ❖ If the distance of a data to it 10 nearest data points (Neighbor) is define:
  - ➢ Check: If all 10 nearest data points have an “income” attribute value of “<=50k”
    - ■ Assign the data “Income” attribute value to “<=50k”
  - ➢ Else: If not all 10 nearest data points have an “income” attribute value of “<=50k”
    - ■ Check: If the majority of nearest data points have an “income” attribute value of “<=50k”
      - ● Assign the data “Income” attribute value to “<=50k”
    - ■ Otherwise, If the majority of nearest data points have an “income” attribute value of “>50k”
      - ● Assign the data “Income” attribute value to “>50k”
    - ■ Otherwise, go to either rule 2 or rule 1 with K = K - 1
  - ➢ Else: go to rule number 1

E. Conclusion

In conclusion, Model 5 employed the K Nearest Neighbor algorithm on Data 2 to predict income levels. The model's parameters included a success class of ">50K," two classes, a success probability cut-off of 0.5, a fixed K value of 10, and prior probability set to empirical. The test set and training set variables were also matched by name.

The evaluation of Model 5 involved assessing its accuracy using the Confusion Matrix and Error Report. The Confusion Matrix provided insights into the model's performance, distinguishing true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). The sum of actual data values (TP+TN) indicated a high level of accuracy, suggesting that the model made correct predictions for a significant number of cases.

The Error Report summarized the mistakes made by the model during the prediction process, presenting the number of cases and errors for each income class, along with the percentage of errors. The specific error rate for Model 5 was not provided in the provided information. However, it was

mentioned that the error rate for Model 1 was approximately 13.3%, which was considered a good track record.

In summary, Model 5 demonstrated effective performance in predicting income levels using the K Nearest Neighbor algorithm. With a high accuracy indicated by the Confusion Matrix and a relatively low error rate (based on Model 1), the model proved to be a reliable tool for predicting income levels. However, without the specific error rate for Model 5, it is difficult to make a precise assessment of its performance in comparison to Model 1.

7. Overall Conclusion

After analyzing the five data mining models based on their error percentages, we can rank them as follows:

1. Model 3: K Nearest Neighbor (K = 3) with an error percentage of 10.81%.
2. Model 2: Naive Bayesian with an error percentage of 14.54%.
3. Model 4: K Nearest Neighbor (K = 5) with an error percentage of 13.3%.
4. Model 5: K Nearest Neighbor (K = 10) with an error percentage of 15.32%.
5. Model 1: Decision Tree with an error percentage of 18.55%.

Model 3, utilizing the K Nearest Neighbor algorithm with K = 3, achieved the lowest error percentage of 10.81%. This indicates a relatively high accuracy in predicting the target variable.

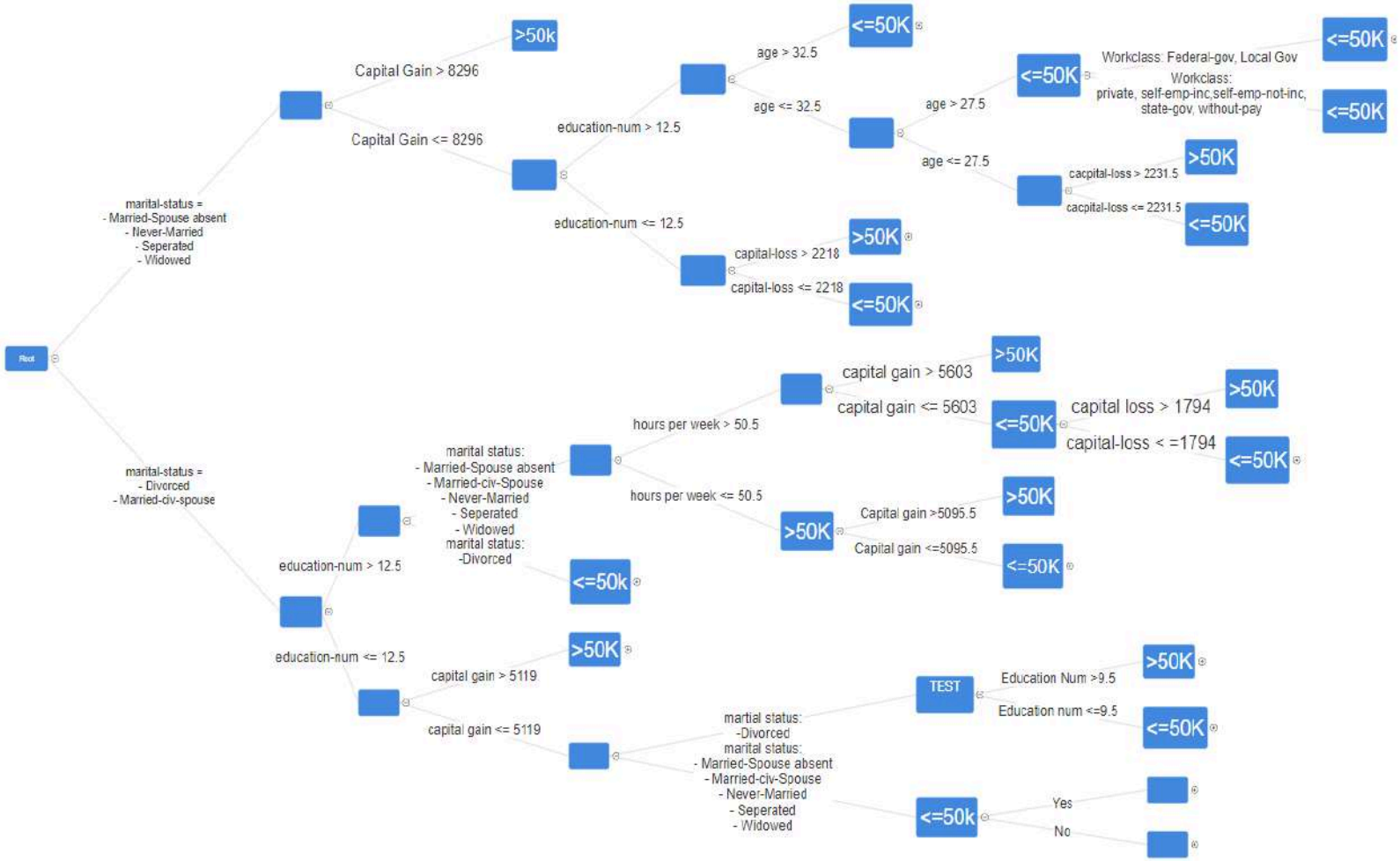
Model 2, employing the Naive Bayesian algorithm, had an error percentage of 14.54%. While a bit higher than Model 3, it still demonstrates reasonable accuracy in its predictions.

Model 4, using K Nearest Neighbor with K = 5, had an error percentage of 13.3%. It performed better than Model 5 but fell slightly short compared to Models 2 and 3.

Model 5, also utilizing K Nearest Neighbor but with K = 10, had an error percentage of 15.32%. Although it performed worse than Models 2, 3, and 4, it still showed a reasonable level of accuracy.

Model 1, based on the Decision Tree algorithm, had the highest error percentage of 18.55%. It performed less accurately compared to the other models.

In conclusion, Model 3 with K Nearest Neighbor (K = 3) achieved the lowest error percentage, indicating the highest level of accuracy among the models evaluated. However, Models 2, 4, and 5 also demonstrated reasonably accurate predictions, while Model 1 had the highest error rate. Researchers and practitioners may consider prioritizing Model 3 or exploring alternative algorithms to further improve prediction accuracy.


Root
marital-status =
- Married-Spouse absent
- Never-Married
- Seperated
- Widowed
Capital Gain > 8296
>50k
Capital Gain <= 8296
education-num > 12.5
age > 32.5
<=50K
age <= 32.5
age > 27.5
<=50K
Workclass: Federal-gov, Local Gov
<=50K
Workclass:
private, self-emp-inc,self-emp-not-inc,
state-gov, without-pay
<=50K
age <= 27.5
cacpital-loss > 2231.5
>50K
cacpital-loss <= 2231.5
<=50K
education-num <= 12.5
capital-loss > 2218
>50K
capital-loss <= 2218
<=50K
marital-status =
- Divorced
- Married-civ-spouse
education-num > 12.5
marital status:
- Married-Spouse absent
- Married-civ-Spouse
- Never-Married
- Seperated
- Widowed
marital status:
-Divorced
<=50k
hours per week > 50.5
capital gain > 5603
>50K
capital gain <= 5603
<=50K
capital loss > 1794
>50K
capital-loss < =1794
<=50K
hours per week <= 50.5
>50K
Capital gain >5095.5
>50K
Capital gain <=5095.5
<=50K
education-num <= 12.5
capital gain > 5119
>50K
capital gain <= 5119
martial status:
-Divorced
marital status:
- Married-Spouse absent
- Married-civ-Spouse
- Never-Married
- Seperated
- Widowed
TEST
Education Num >9.5
>50K
Education num <=9.5
<=50K
<=50k
Yes
No