Table of Contents

MIS 576 Data Mining for Business Analytics

Assignment 3 – Predictive Analysis
(Using Supervised Data Mining Techniques for Classification and Prediction)

Assignment Submitted By	**Sudharsan Elangovan**

1. After reading and following the instructions of Assignment 3 Lesson Plan (*Assignment 3.PredictiveAnalysis.Lesson Plan.pdf*) available on canvas with *Assignment 3 – Predictive Analysis*, start following the step by step instruction of this document to perform classification and prediction using Decision Trees and Regression. And provide response to all highlighted Questions (1-14) as mentioned in the steps. To recall SAS Enterprise Miner Project automatically saves your work and you can work on the project, close the project, and come back where you left to start working again.

To recall, our aim in this assignment is to answer following two research questions:

- **RQ 1: To identify the significant/important factors that can be used to classify the car as an expensive or inexpensive car.** (Note: *To answer this research question, the target variable in ToyotaCar data set will be Expensive. Since the target variable is a categorical variable, we will use Classification Trees and Logistic Regression to perform this analysis*).

- **RQ 2: To identify the significant/important factors that can be used to predict the price of used Toyota cars.** (Note: *To answer this research question, the target variable in ToyotaCar data set will be Invoice. Since the outcome is a numerical variable we will use Regression Trees and Multiple Linear Regression to perform this analysis*).

2. **Now we will work on the continuation of Step 5.12 of the lesson plan (*Assignment 3.Predictive Analysis Lesson Plan.pdf*) available on canvas with *Assignment 3 – Predictive Analysis*.**

Exploring the Data

3. Ensure your *NS.PredictiveAnalysis* Project is open in UIS Citrix. Double Click on *NS.Classification* diagram to develop classification tree to answer research question 1 (RQ1).
4. The variable description is available in Appendix.
5. Right click on ToyotaCar node of *NS.Classification* diagram. Click on "Edit Variables" as shown in Figure 1.

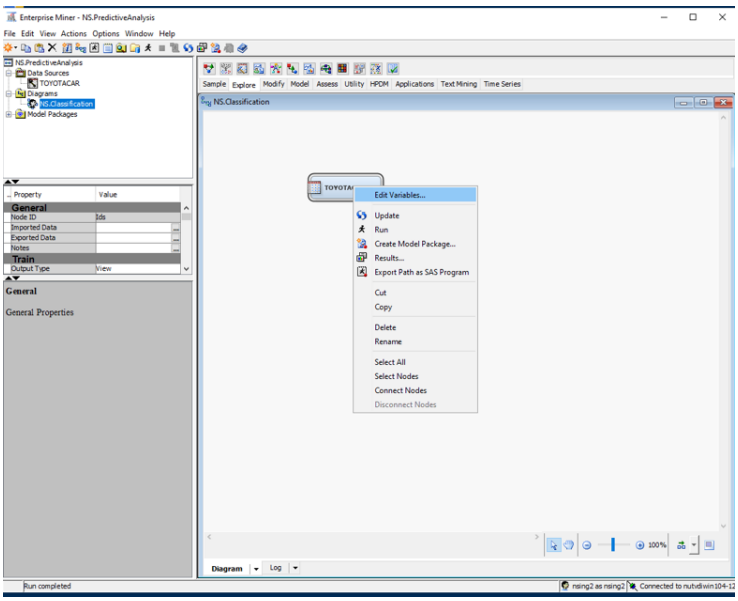


Figure 1

6. In the Edit Variables Window, click on the first variable row and either use CTRL+A keys or while holding the CTRL key click on the last row. All the variables will be selected in the Variables-Ids window and now you can click on Explore Button (Figure 2). This will open the Explore window to show the sample/summary statistics of variables, data distribution (histograms for each variable), sample properties of data and data in tabular format (Figure 3). You can Maximize any window to see the details. This is a quick way to explore the data to identify any anomalies. If we investigate the histograms, almost all variables are normally distributed.

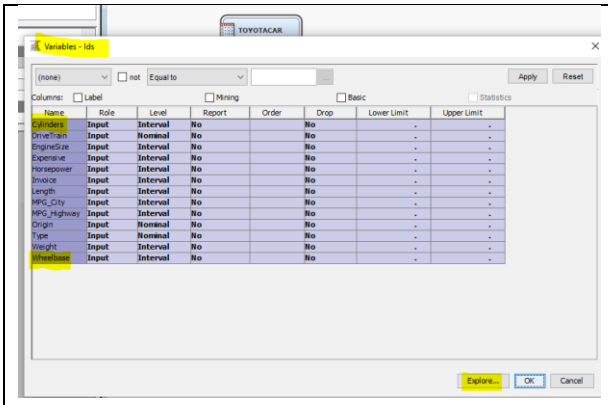


Figure 2

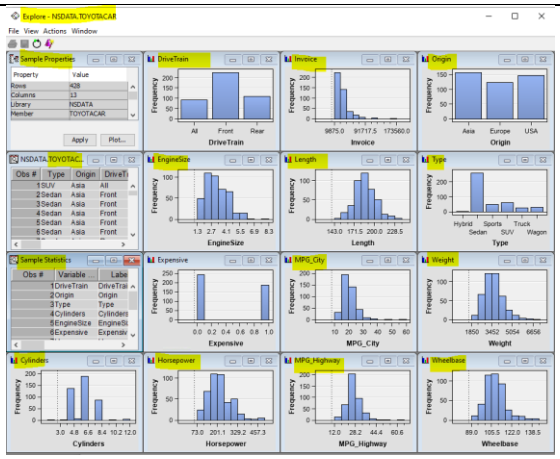


Figure 3

7. Close the Explore Window and Variable – Ids Window.

RQ1: Classification - Setting the Roles and Levels of Variables

8. Now we will start working on developing the predictive model of RQ 1 in which we need to identify the factors to classify the car as expensive or inexpensive.
9. To set the role and level of each variable of your dataset, Right click on ToyotaCar node of *NS.Classification* diagram. Click on "Edit Variables" again to launch the Variables-Ids Window. Click on the Role cell and Level cell to enable the dropdown arrow and to choose the roles and levels of the variables. Now set the roles and levels for each variable as mentioned in Table 1.

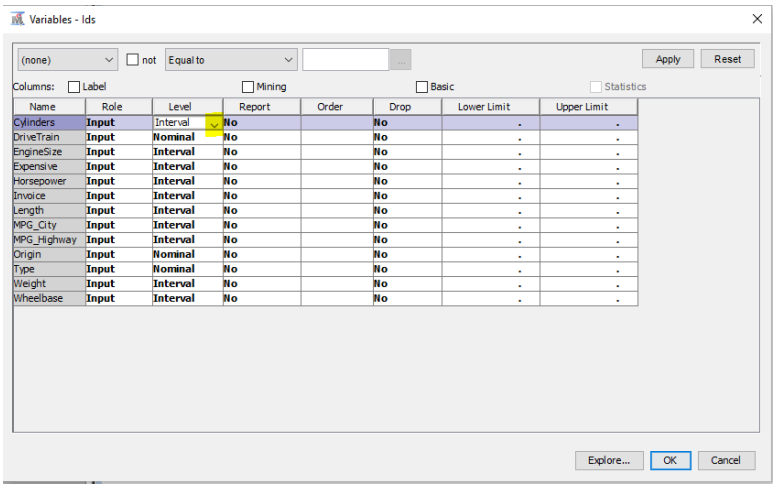


Figure 4

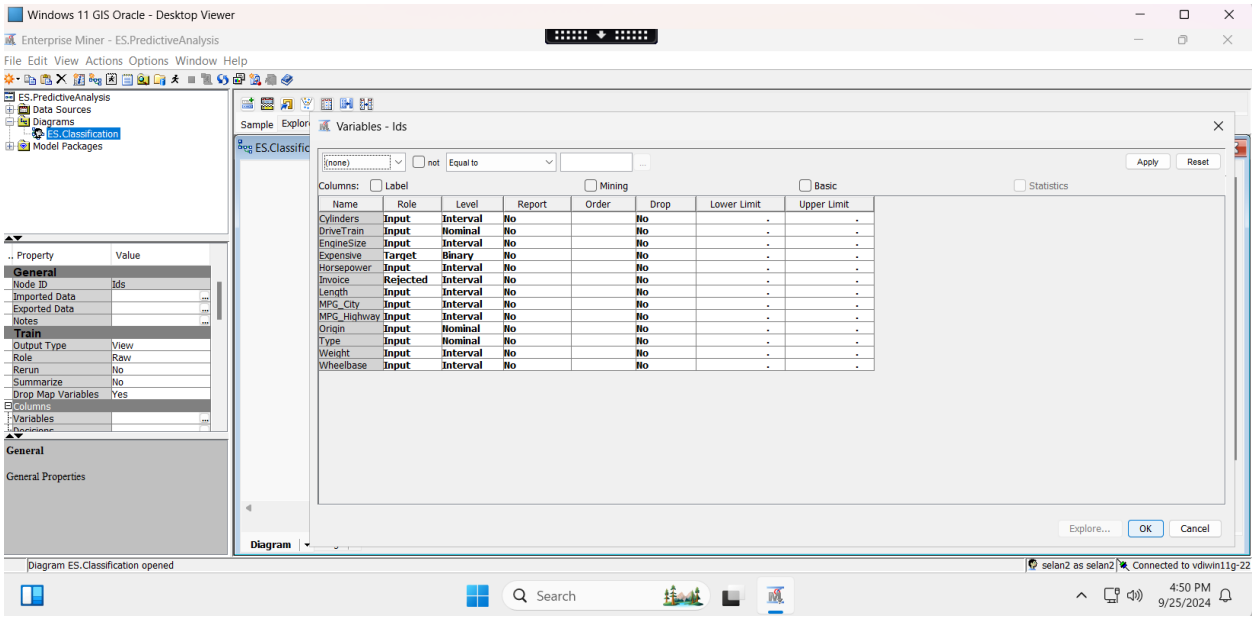
Variable Name	Roles	Variable Type (Levels)
Cylinders	Input	Interval
DriveTrain	Input	Nominal
EngineSize	Input	Interval
Expensive	**Target**	**Binary**
Horsepower	Input	Interval
Invoice	**Rejected**	**Interval**
Length	Input	Interval
MPG_City	Input	Interval
MPG_Highway	Input	Interval
Origin	Input	Nominal
Type	Input	Nominal
Weight	Input	Interval
Wheelbase	Input	Interval

Table 1. Roles and Levels of ToyotaCar Variables to develop Classification Tree

Note: This means at this point we are using the Toyotacar data set to answer our RQ1. To recall we want to classify the car as Expensive or Inexpensive, Expensive is our Target variable.

Question 1: Variables for RQ1

Provide the screenshot of Edit Variables screen showing the roles and levels of all the variables you have set in the previous step for developing your Classification Tree. Ensure to capture NetID and machine number at the lower right corner. Also, explain why you are using Expensive as a target variable here and why you are rejecting Invoice for Classification Tree (50-100 words).



- **Expensive is set as the Target variable (Binary), where the value 1 indicates an expensive car, and 0 indicates an inexpensive car.**
- **Other variables like Cylinders, EngineSize, Horsepower, and MPG_City are set as Input variables with an Interval data type.**
- **DriveTrain, Origin, and Type are Input variables with a Nominal level.**
- **Invoice is Rejected for classification purposes, as it represents the price of the car and does not directly influence the binary classification of "Expensive."**

10. Click OK to close the Edit Variables window.

Create a Classification Tree to Classify a new Car

11. In this section we will develop a classification tree predictive model to classify a new car as expensive or inexpensive (RQ1). So, Double Click on *NS.Classification* diagrams in the project pane. *ToyotaCar* node is on this diagram with the roles and levels edited as shown in Table 1.
12. Click on the "Sample" tab on the toolbar and click and drag "Data Partition" (the second from left) onto the diagram as shown below. (*Data Partition node randomly splits the data into training, validation, and test sets. The default in SAS EM is Training-40%; Validation:30%; Test:30%*)

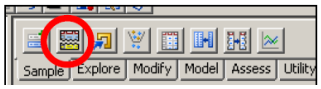


Figure 5

13. Click on the "Model" tab on the toolbar and click and drag the "Decision Tree" node to the diagram (Second node from the left on Model Tab).

Figure 6

14. Connect *ToyotaCar*, *Data Partition* and *Decision Tree* Nodes as shown below.

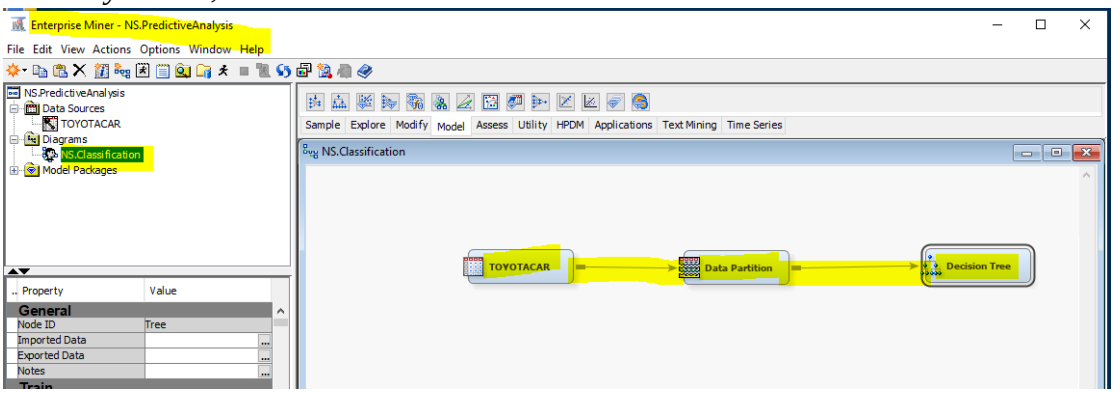


Figure 7

Note: Since our target variable (Expensive) is a categorical variable, decision tree will be referred to as classification tree

15. Click on the "Decision Tree" node to open its properties on the left pane. Observe the properties of decision tree node. The splitting rule property shows the maximum depth and branch for the tree. Default is Maximum Branch 2 and Maximum Depth 6.

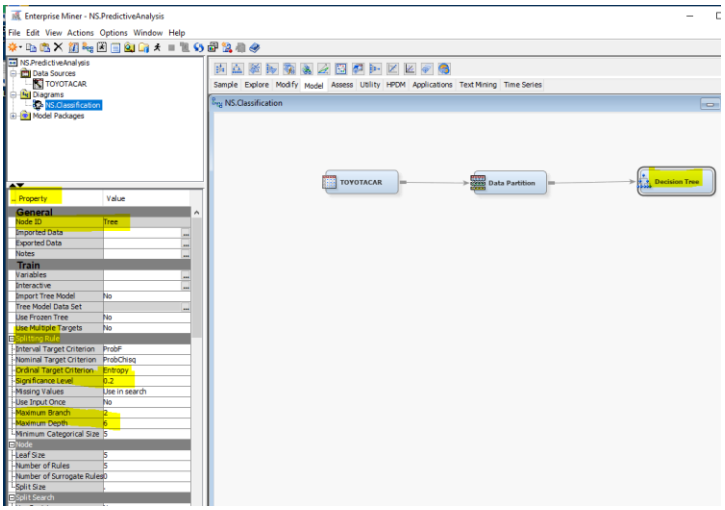


Figure 8

16. Right Click on Decision Tree Node in Figure 8 and rename it as **ClassDecTree B2D6.** Renaming helps us to better analyze the model by recalling their basic properties. But we must ensure that renaming the node is just a label, the properties need to be set in the properties windows to implement the criteria accordingly.
17. Now our classification tree model is created as a process flow. ***Since the target variable is Expensive which is a categorical variable with two categories the recursive partition algorithm of decision tree will develop a classification tree which will help us to classify the new car as expensive or inexpensive***. Right click on **ClassDecTree B2D6** node and click on Run to execute your classification predictive model. After the Run is completed click on Results as shown below.

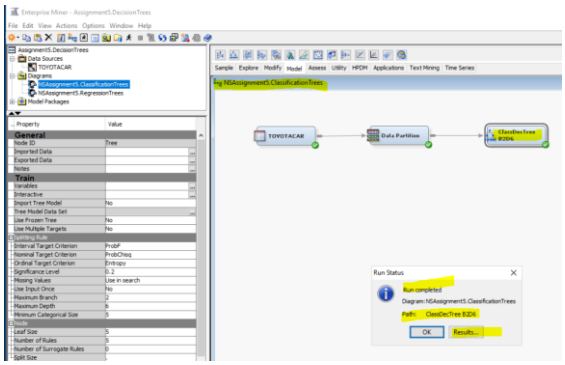


Figure 9

Note: If by mistake you clicked on OK. Don't worry. Just right click on <u>ClassDecTree B2D6</u> Node and you will find the Results option. Just click on it to see the results.

18. As shown in Figure 10, you will see that Results window comprises of Score Rankings Overlay: Expensive Window, Tree Window, Leaf Statistics Window, Fit Statistics Window, Treemap Window, and Output Window. Maximize each window to see the details.

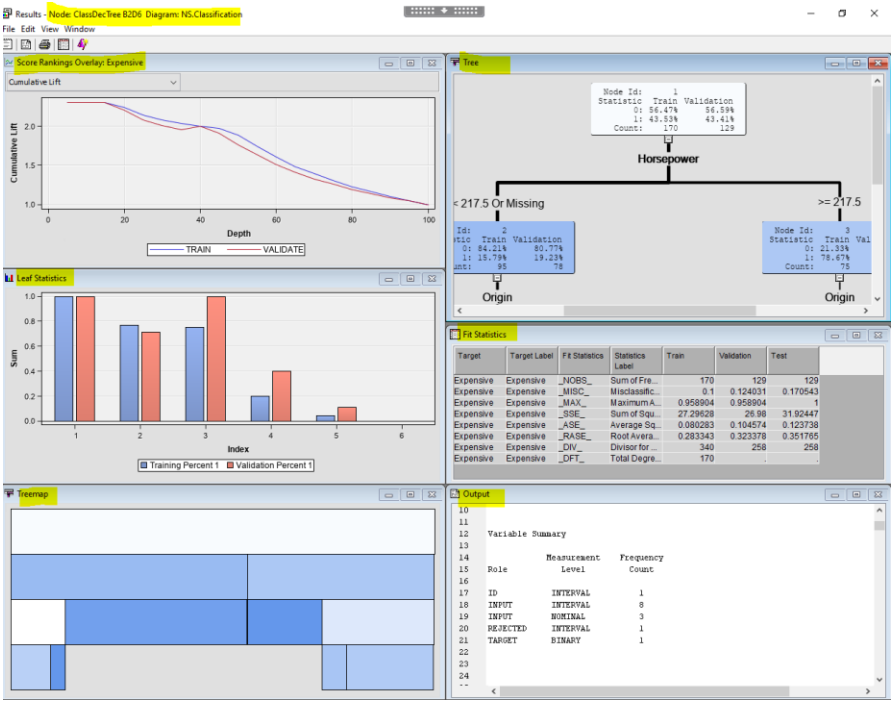


Figure 10

19. Close the Results Window of ***ClassDecTree B2D6*** node.

Analyzing the Results of Classification Tree

20. To analyze the results of classification predictive model, first we need to identify the baseline accuracy and baseline misclassification rate which is referred to as Naïve Rule.
    - **To recall the target variable in a classification model is a categorical variable. The naïve rule helps to identify the majority class of the target variable. The percentage of majority class in the data set is the baseline accuracy and the percentage of minority class is the baseline misclassification. This means the sum of baseline accuracy and baseline misclassification is 100%.**
21. Let's identify the **baseline misclassification rate**:
    1. Click on the ToyotaCar node of the *NS.Classification* diagram to open its properties in the left pane.
    2. Click on the ellipsis for "Variables".
    3. Click on the variable name "Expensive" and
    4. Click "Explore".
    5. The bar chart (called "Expensive") shows the number of expensive cars vs. inexpensive cars. Hovering the mouse over each bar shows the number of cars in each category.

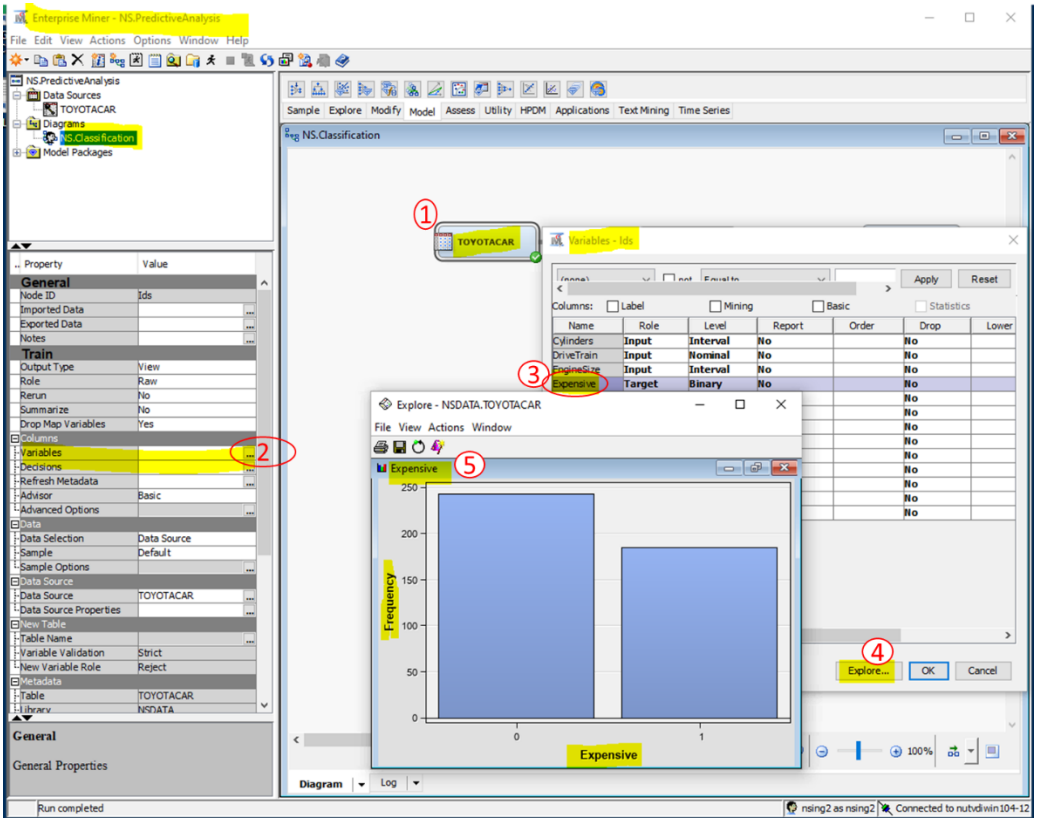


Figure 11

Note: Based on this bar chart for Expensive variable, inexpensive car (0) is the majority class. So, the baseline accuracy is 56.8% (243/(243+185)) because if we assume that all cars were inexpensive, we would be correct by this much for this data set. The baseline misclassification rate is 43.2% (100-56.8).

Question 2: Naïve Rule - Baseline Misclassification and Accuracy

As mentioned in Step 21, **explore the bar chart for your Expensive Variable** and answer questions in below table (Table 2) **to identify the baseline misclassification rate for your ToyotaCar data set**.

How many expensive cars ("1") are there in your bar chart?	185 expensive cars
How many inexpensive cars ("0") are there in your bar chart?	243 inexpensive cars
Which class is the majority class for your Expensive variable in the ToyotaCar data set?	Inexpensive car(0) is the majority class for my expensive variable.
Calculate the percentage of majority class. This percentage is your baseline accuracy.	(243/(243+185)) * 100 = 56.8%
Subtract the baseline accuracy from 100%, this will give you the baseline misclassification rate. So, what is your baseline misclassification rate?	100 – 56.8 = 43.2%

Table 2. Baseline Misclassification Rate

22. Close the "Explore" window and "Variables" window of previous step. So, we have explored the baseline misclassification and accuracy (naïve rule). (*Hint: If the misclassification rate of validayion set of the model is greater than the baseline classification then model is not performing good. However, misclassification rate of model less than baseline misclassification shows model is performing better to accurately classify the observation. Further, the misclassification rate should decrease from training to validation to test or difference should be less than 5% of Training for Training and Validation and less than 5% of Validation for Validation and Test)*
23. Now, let's identify the misclassification rate of classification tree ***ClassDecTree_B2D6*** executed above in step 17.
24. Right click on ***ClassDecTree B2D6*** and open the Results window.
25. Under "Fit Statistics" column, find "**_MISC_**" (this is the misclassification rate).

Question 3: Classification Tree - Misclassification Rate

Question 3a

What is the misclassification rate of Validation partition? Based on the baseline misclassification and model misclassification, discuss if your model is performing better (50-100 words).

The validation partition has a misclassification rate of 0.124 (12.4%).

Interpreting the Baseline Error, As we answered in the previous question the baseline misclassification rate is 43.2%, when entirely predicting the majority class, as naive rule. Since the misclassification rate of the validation partition is 12.4%, which is significantly lower than the baseline, this indicates that the model is performing better than the naive classifier. This suggests that the decision tree model is effective in classifying the cars as expensive or inexpensive based on the provided features.

Question 3b

What are the misclassification rates of Training, Validation and Test partitions?

Data Partition	Misclassification rate_(_MISC_)
Training Set	0.1 (10%)
Validation Set	0.124031 (12.4%)
Test Set	0170543 (17.1%)

Table 3. Misclassification Rate for *ClassDecTree B2D6* model

26. In the Results window of ***ClassDecTree_B2D6***, maximize the "**ScoreRankings Overlay: Expensive**" window. This is the cumulative lift chart of the model. This chart can also be used to determine whether there is overfitting or not. If the Train and Validate data curves are almost same, then there are less chances of overfitting. The farther the curves, the higher the overfitting. (*Hint: In any model there is always some overfitting, and our aim is to minimize the overfitting to get the best model.*)

Question 3c

Based on the "ScoreRankings Overlay: Expensive" and analysis in questions 3a & 3b is there any evidence of overfitting in the classification model (*ClassDecTree_B2D6*)? (*Note:*** *Overfitting exists when the model performs well on training, but not so well in validation/test sets).* Explain in 50-100 words.**

Based on the Score Rankings Overlay, The training (blue line) and validation (red line) curves are relatively close to each other, especially in the early stages (lower depth). As the depth increases, there is a slight divergence, but it is not significant. This suggests that overfitting is minimal in this model. Overfitting typically occurs when the training curve performs much better than the validation curve. Since the curves are close together, it indicates that the model generalizes well to unseen data, and the risk of overfitting is low. The misclassification rates from the previous question further support this, as the rates are similar across the training, validation, and test partitions.

27. Minimize the "ScoreRankings Overlay: Expensive" window to its original size.
28. To examine the classification matrix, maximize the "Output" window in the Results window of ClassDecTree B2D6.
29. Scroll down in the "Output" window and find the section "Event Classification Table". Look at the section for "Data Role=VALIDATE" (*this section is for the validation data set*). This classification or confusion matrix is used to calculate the misclassification rate for validation set as shown in by _MISC_ for validation in the "Fit Statistics" Window of the Results.
30. This is the classification matrix. You can observe the model's effectiveness by looking at the classifications.
31. Minimize the "Output" window to its original size.
32. Now, let's observe the Tree by maximizing the window called "Tree" in the Results window of ***ClassDecTree B2D6*** as shown in below screenshot. The first node is called the root node. The first splitting variable is "Horsepower". So, there are two branches: if Horsepower is <217.5 or Horsepower is >=217.5. From each branch, there are other branches based on other variables. All the highlighted variables in Figure 12 are the significant/important variables identified by this classification tree to classify the car as expensive or inexpensive.

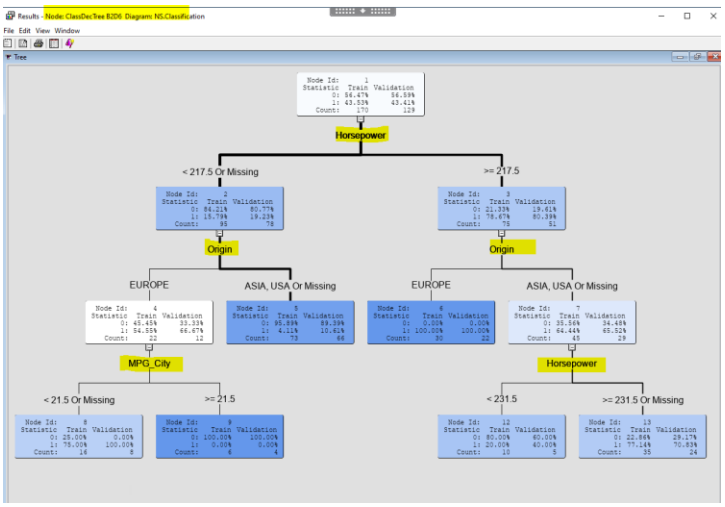


Figure 12

Note: If the whole tree is not visible on the screen or the font is too small to read the tree, right click anywhere in the Tree window, select "View" then change the view percentage accordingly.

33. In any tree, the node with splitting rules is referred to as "Decision Node" and the nodes with no further splitting rules is referred to as leaf node. Leaf node in tree is used for classifying the new observation. For example, in the above tree, nodes 1, 2, 3, 4, and 7 are Decision Nodes and nodes 5, 6, 8, 9, 12, and 13 are leaf nodes. To classify a new car based on the provided characteristics we need to navigate to the leaf node. The statistics in leaf node will help us to classify the car as expensive or inexpensive using the majority vote for a class in the leaf node.
34. Now, let's classify a new car (manually) using the below tree. For example, if a new car has Horsepower=150 and Origin=USA; the car would be classified using Node ID 5. Since Node 5 is **a "leaf" node**, it **can be used for making classifications**. W**e use the "majority rule" in the leaf node to make classifications**. The Train column in this node has "0" as the majority class (95.89%). **Therefore, the car with Horsepower of 150 and Origin as USA will be classified as inexpensive (0) car as shown in below tree.**

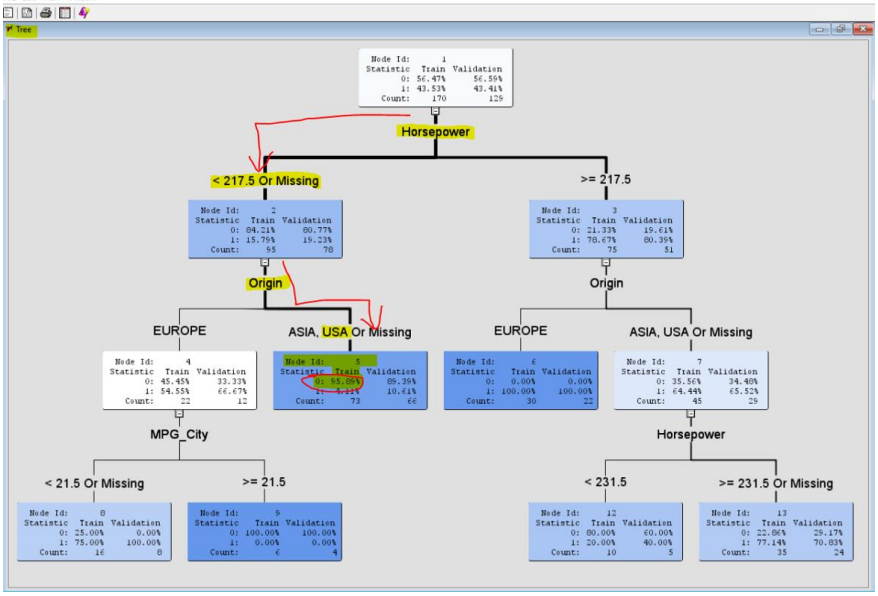


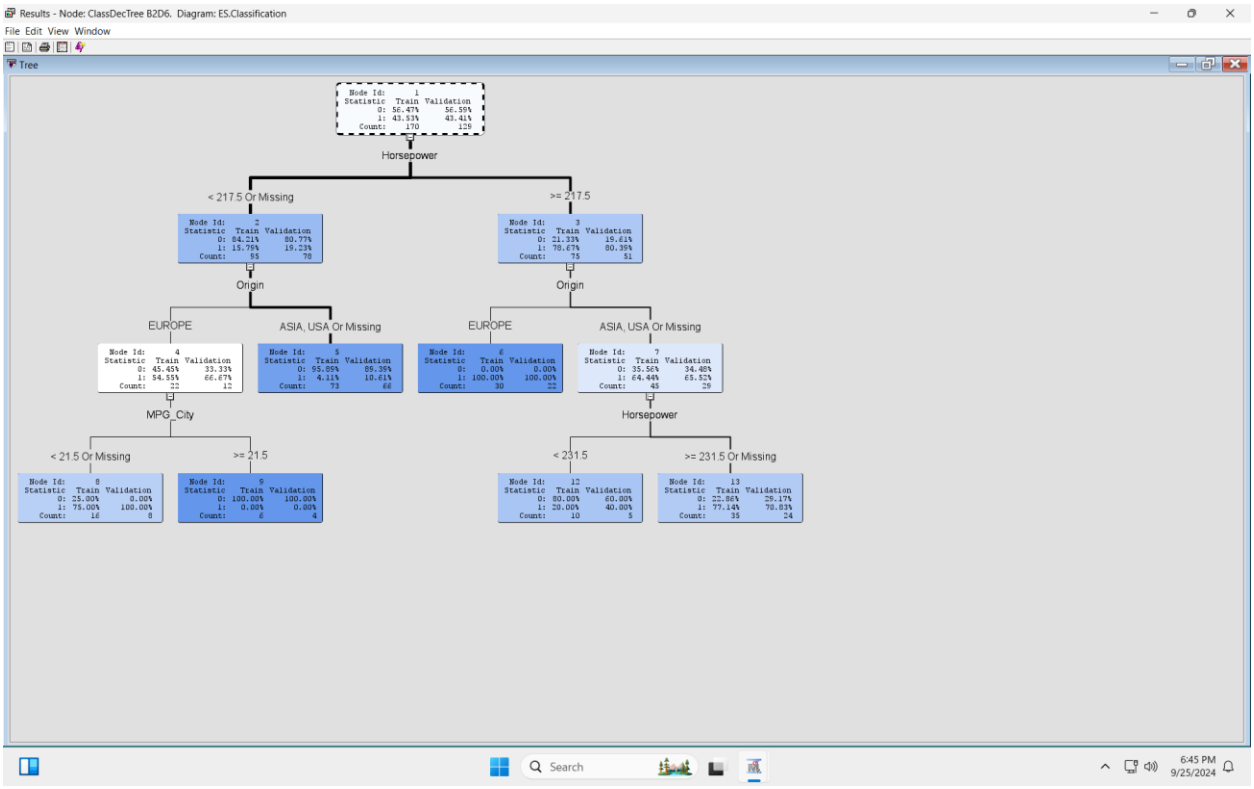
Figure 13

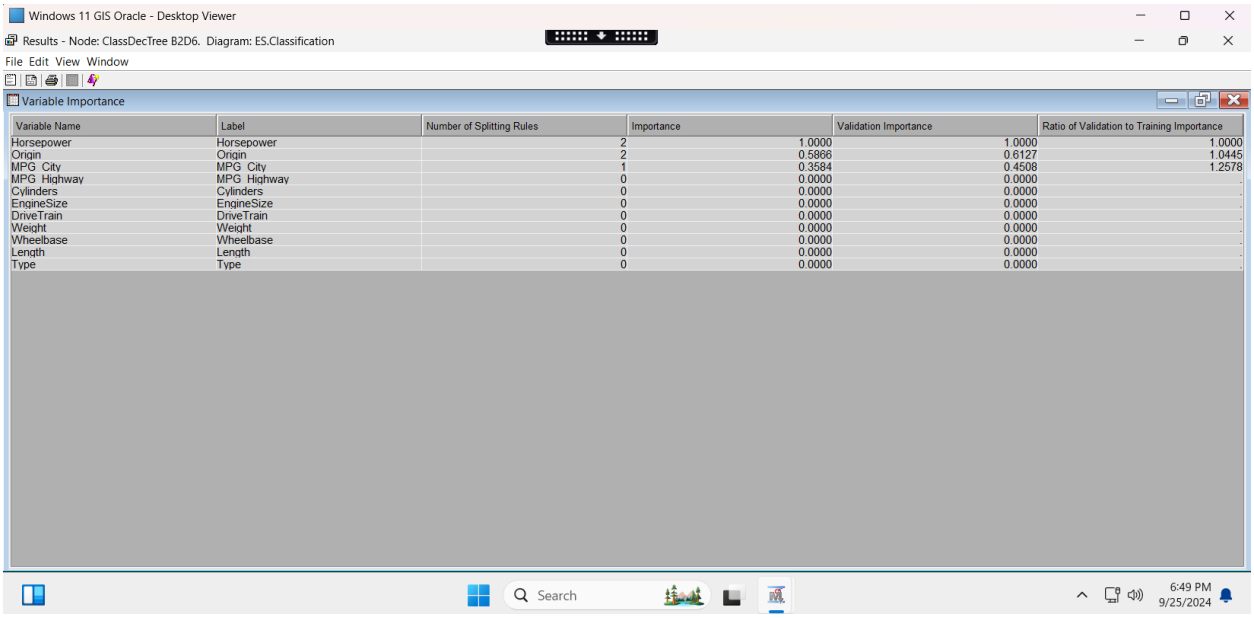
Note: The variables Horsepower, Origin, and MPG_City are the important variables used to develop this decision tree (classification tree). The variable importance can also be explored by going to "View" Menu>Model>Variable Importance. The variables with "Importance" of greater than Zero are important variables which are used to develop the classification tree using recursive partitioning algorithm. The variable with "Importance" of 1 is the most important variable used for classifying the outcome (car in this case). The variable with "Importance" of 0 is not at all an important variable for the classification of outcome variable.

Question 4: Classifying a new car

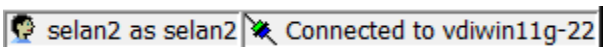
Question 4a

Explain the important variables identified by classification tree to develop the model (50-100 words). Also, take the screen shot of your "Tree" Window and ensure to capture your NetID and Machine number. Insert the screenshot here.





Variable Name	Label	Number of Splitting Rules	Importance	Validation Importance	Ratio of Validation to Training Importance
Horsepower	Horsepower	2	1.0000	1.0000	1.0000
Origin	Origin	2	0.5866	0.6127	1.0445
MPG_City	MPG_City	1	0.3584	0.4508	1.2578
MPG_Highway	MPG_Highway	0	0.0000	0.0000	.
Cylinders	Cylinders	0	0.0000	0.0000	.
EngineSize	EngineSize	0	0.0000	0.0000	.
DriveTrain	DriveTrain	0	0.0000	0.0000	.
Weight	Weight	0	0.0000	0.0000	.
Wheelbase	Wheelbase	0	0.0000	0.0000	.
Length	Length	0	0.0000	0.0000	.
Type	Type	0	0.0000	0.0000	.



The important variables identified by the classification tree to develop the model are Horsepower, Origin, and MPG_City. These variables had a significant impact on classifying cars as expensive or inexpensive based on their values. Among them, Horsepower is the most critical factor, followed by Origin and MPG_City. The decision tree's first split occurs based

on Horsepower, showing its primary importance in determining whether a car is classified as expensive or inexpensive.

Question 4b

a. **If a new car has the following values: Horsepower=210 and Origin=Europe and MPG_City=22; how would you classify this car based on the classification tree analysis. Also, mention the Node Id that helped you to make the classification. (***Note: Follow the nodes in the tree view of Question 4a; use the train column of leaf node for classification***)?**

Based on the classification tree analysis, this car would be classified as inexpensive. According to the node rules, the car with Horsepower = 210 and Origin = Europe will follow the condition for Horsepower < 217.5. The next condition based on MPG_City = 22 places it in Node ID 9 since MPG_City >= 21.5. The relevant node for this classification is Node ID 9. In Node ID 9, the predicted outcome is Expensive = 0 (inexpensive) with 100% confidence. Therefore, this car would be classified as inexpensive.

b. **If a new car has the following values: Horsepower=235 and Origin=USA; how would you classify this car based on the classification tree analysis. Also, mention the Node Id that helped you to make the classification. (***Note: Follow the nodes in the tree view of Question 4a; use the train column of leaf node for classification***)?**

Based on the classification tree, this car would be classified as expensive. The car with Horsepower = 235 follows the condition Horsepower >= 231.5. The Origin = USA places it in Node ID 13. The relevant node for this classification is Node ID 13. In Node ID 13, the predicted outcome is Expensive = 1 (expensive) with 77% confidence. Therefore, this car would be classified as expensive.

35. Minimize the "Tree" window to its original size.
36. In the "Results" window of ***ClassDecTree B2D6***, we can see the rules for the leaf nodes as well. Click on the "View" menu of Results window. Go to Model>Node Rule. These **node rules** are also referred to as **English Rules** or **If then else rules**. The **node rules are only available for leaf nodes** as those are the terminal nodes in the decision tree which are used for classification of new observation.

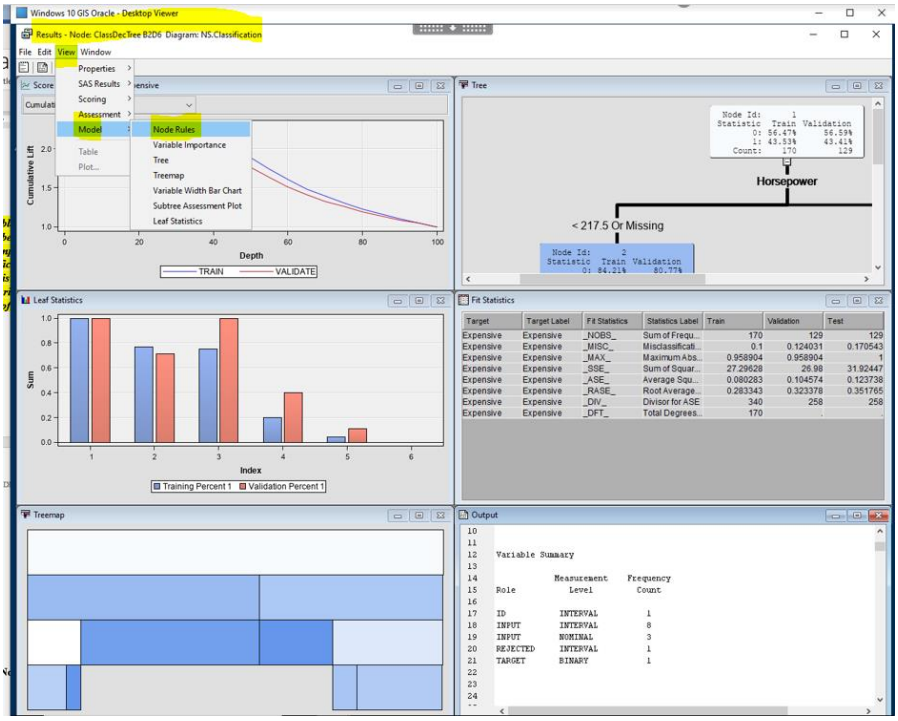


Figure 14

37. The statistics of all the leaf nodes of your tree will be represented in the form of **Node Rules** or **English Rules** or **If then else rules** as follows:

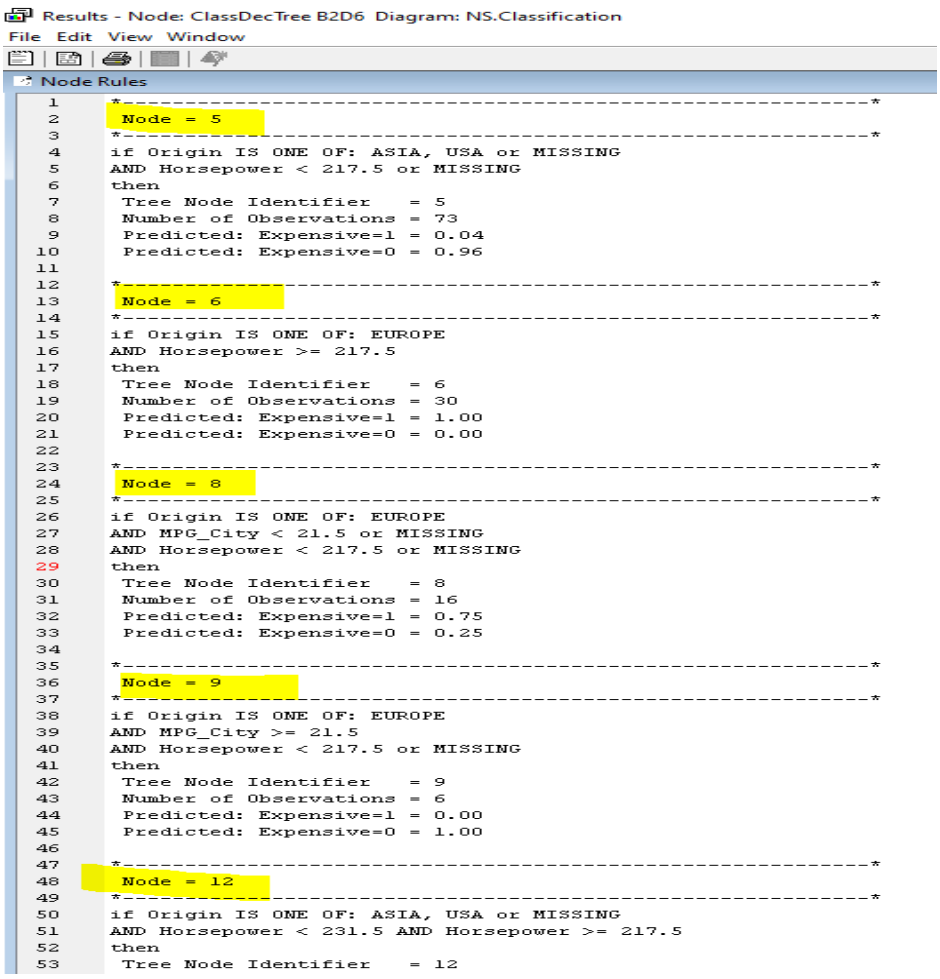


Figure 15

38. Close the Results Window of ***ClassDecTree B2D6.***

Pruning the Tree

39. Now, to optimize the decision tree and identifying a tree with minimum overfitting and highest accuracy, we will try pruning the tree by changing its maximum depth and branches and some other characteristics.
40. In the *NS.Classification* diagrams, add a new decision tree node and rename it to **ClassDecTree B2D4.**
41. Click on **ClassDecTree B2D4** to open the property window of this decision tree node on the left pane. In the properties pane on the left: **change the "Maximum Depth" to 4**. This determines how far the tree can go down. (So, you are pruning the tree after 4 levels.)
42. Connect the **Data Partition** Node to **ClassDecTree B2D4** node. Right click on **ClassDecTree B2D4** node and click Run to execute the decision tree pruned to a maximum depth of 4.

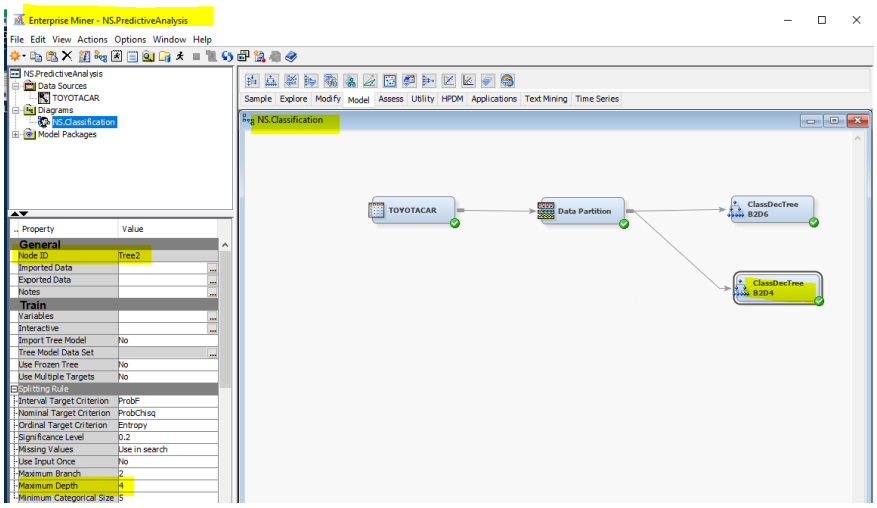


Figure 16

43. Again, in the *NS.Classification* diagrams, add a new decision tree node and rename it to **ClassDecTree B2D2.**
44. Click on **ClassDecTree B2D2** to open the property window of this decision tree node on the left pane. In the properties pane on the left: **change the "Maximum Depth" to 2**. This determines how far the tree can go down. (So, you are pruning the tree after 2 levels.)
45. Connect the **Data Partition** Node to **ClassDecTree B2D2** node. Right click on **ClassDecTree B2D2** node and click Run to execute the decision tree pruned to a maximum depth of 2.

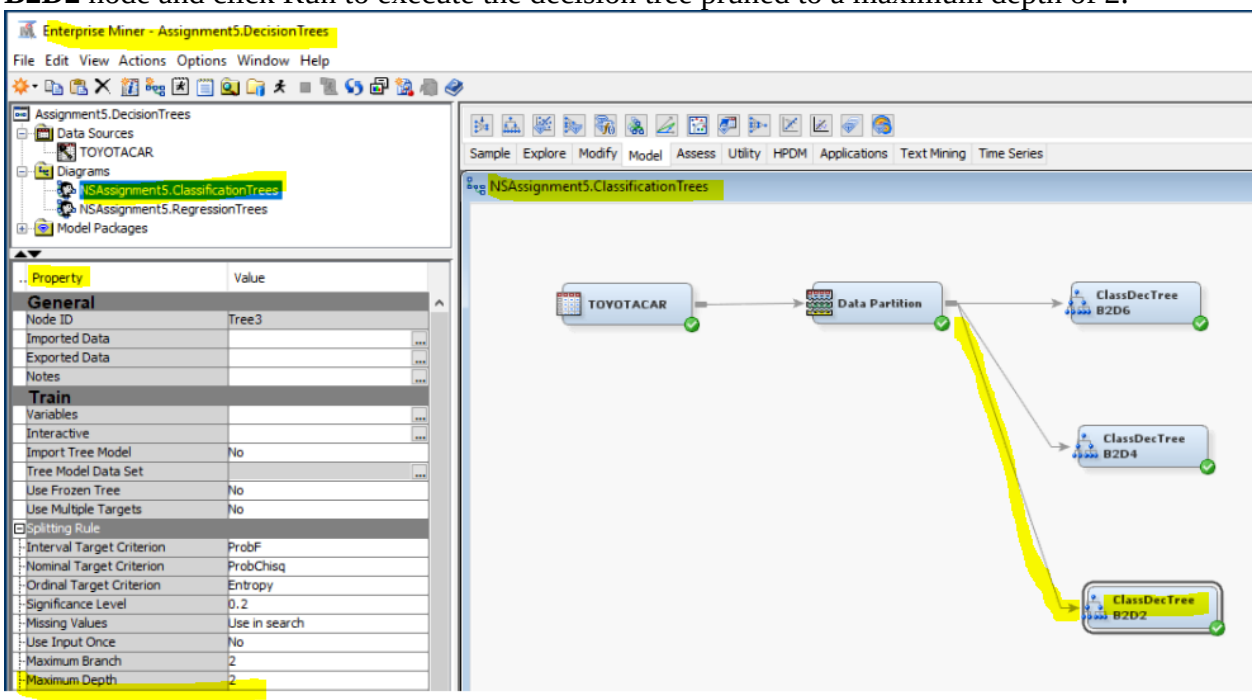


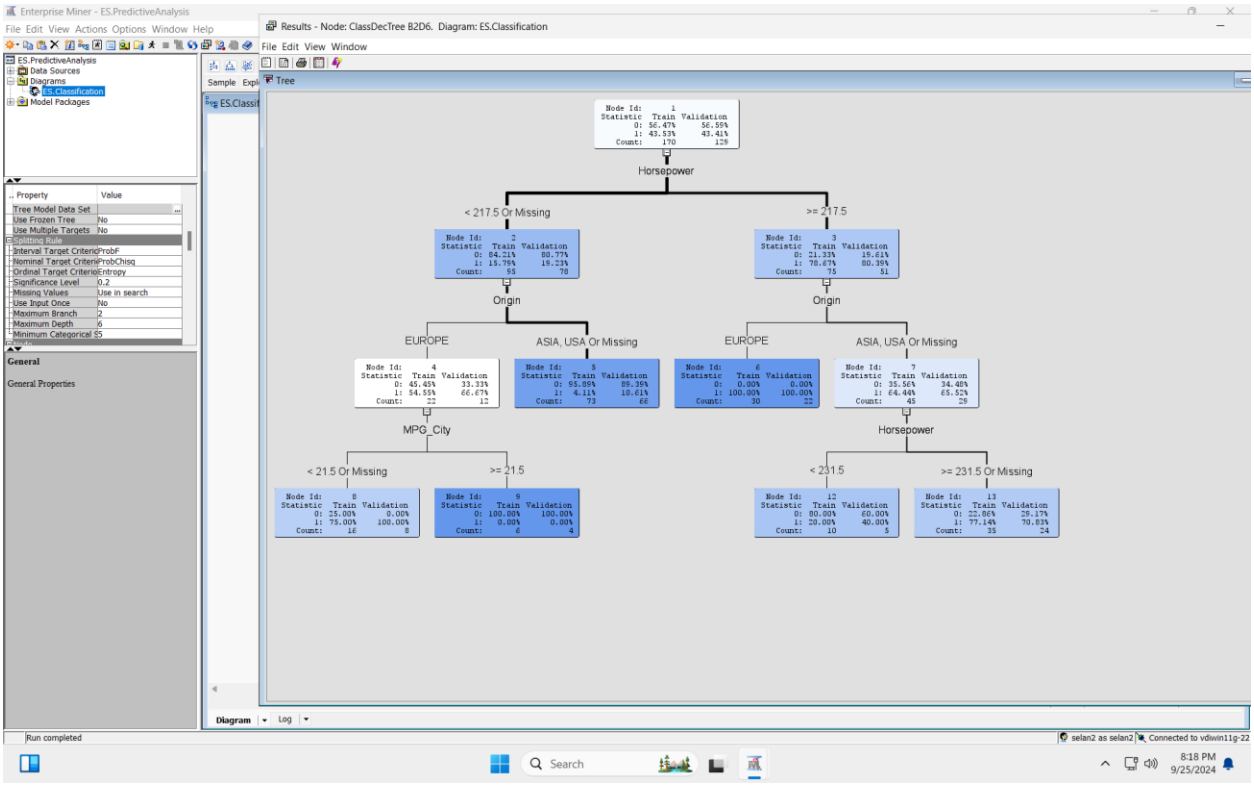
Figure 17

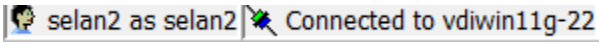
Question 5: Best Classification Tree

Question 5a

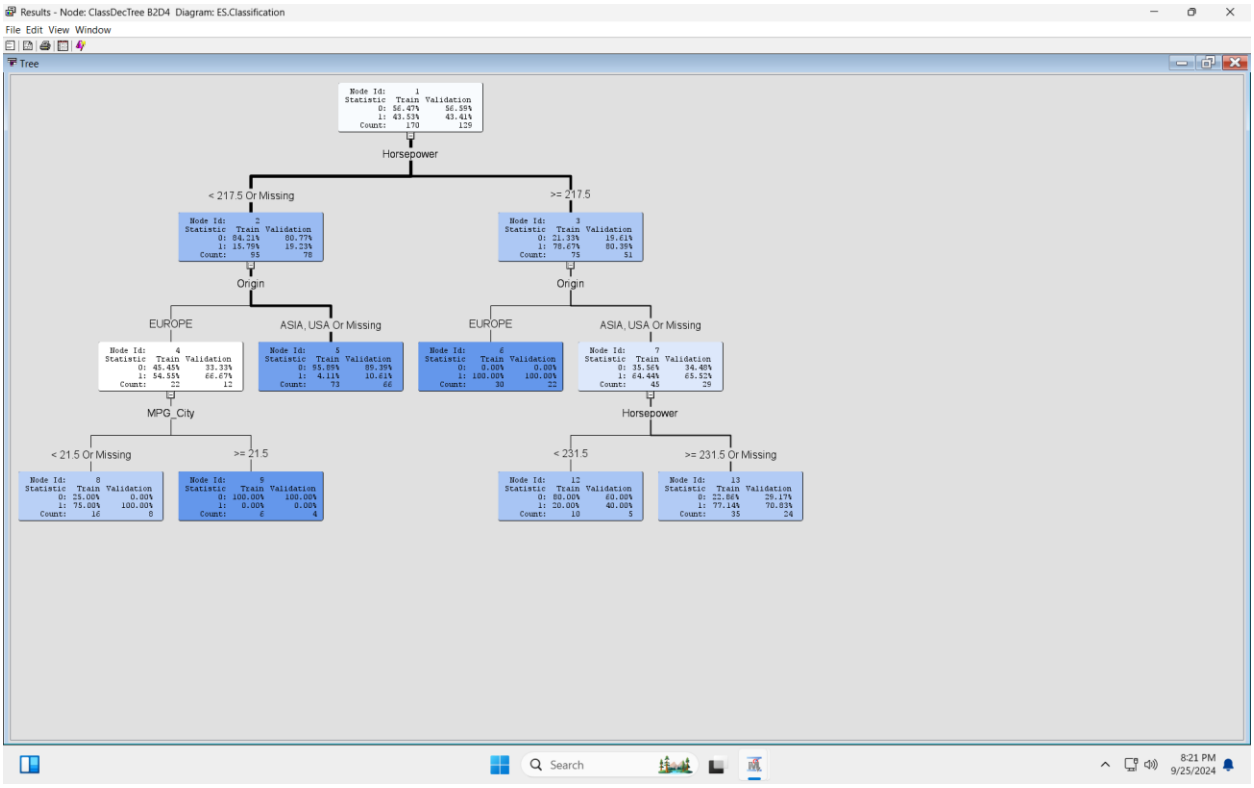
Provide the screenshots of "Tree" window of "ClassDecTree B2D6", "ClassDecTree B2D4", and "ClassDecTree B2D2". Ensure to capture the NetID and machine number while taking screenshots. Insert all the three screenshots here. (*Note: Tree window includes the tree developed by the models as shown above in Figure 12 and 13.*)

ClassDecTree B2D6:



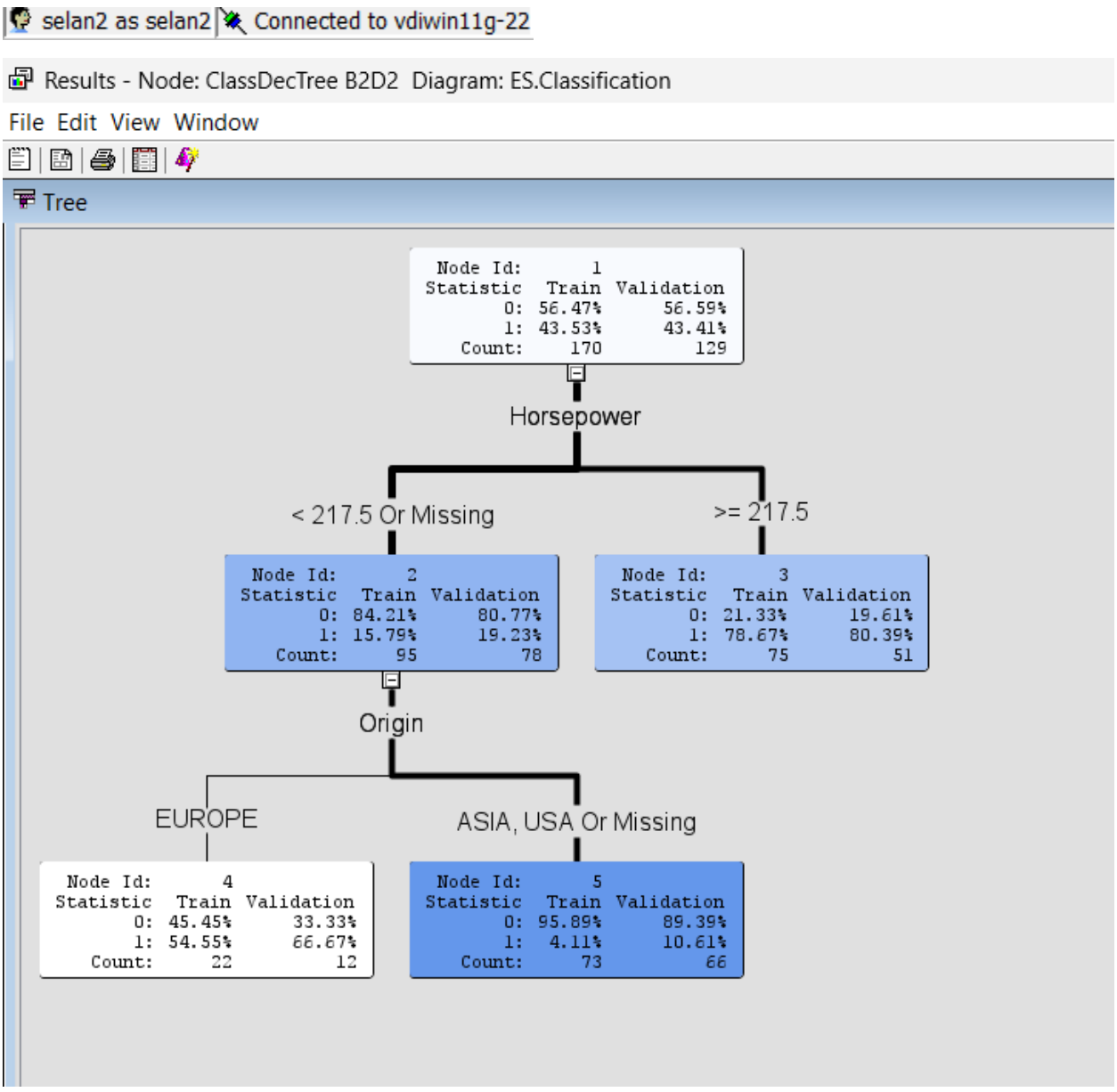


ClassDecTree B2D4:



selan2 as selan2 | Connected to vdiwin11g-22

ClassDecTree B2D2:



Question 5b

What are the misclassification rates of Training, Validation and Test partitions for "ClassDecTree B2D6", "ClassDecTree B2D4", and "ClassDecTree B2D2" classification decision trees? (*Note: Misclassification rate (_MISC_) is usually available in the Fit Statistics and/or Output window of the Results of each model*)

Data Partition	ClassDecTree B2D6 Misclassification rate_(_MISC_)	ClassDecTree B2D4 Misclassification rate_(_MISC_)	ClassDecTree B2D2 Misclassification rate_(_MISC_)
Training Set	0.1(10%)	0.1(10%)	0.170588(17.1%)
Validation Set	0.124031(12.4%)	0.124031(12.4%)	0.162791(16.2%)
Test Set	0.170543(17.1%)	0.170543(17.1%)	0.186047(18.65)

Question 5c

Based on the baseline misclassification and information in table of Question 5b, which of the three models is performing best, and which model is performing worst? Explain (50-100 words).

The model having the lowest misclassification rate is treated as the best-performing model and the one with a high misclassification rate is termed as worst worst-performing model. As per the data we have, the top performing models are B2D6 and B2D4 because it compares to misclassify the low model B2D2. I found that ClassDecTree B2D6 is the most suitable model. ClassDecTree B2D2 performs the worst, with consistently higher misclassification rates across all partitions.

46. Now, in the *NS.Classification* diagrams, add a new decision tree node and rename it to **ClassDecTree B3D6.**
47. Click on **ClassDecTree B3D6** to open the property window of this decision tree node on the left pane. In the properties pane on the left: **change the "Maximum Branch" to 3 and keep the "Maximum Depth" 6**. This determines how wider the tree can go. (So, you are pruning the tree to maximum 3 branches.)
48. Connect the **Data Partition** Node to **ClassDecTree B3D6** node. Right click on **ClassDecTree B3D6** node and click Run to execute the decision tree pruned to a maximum branch of 3.

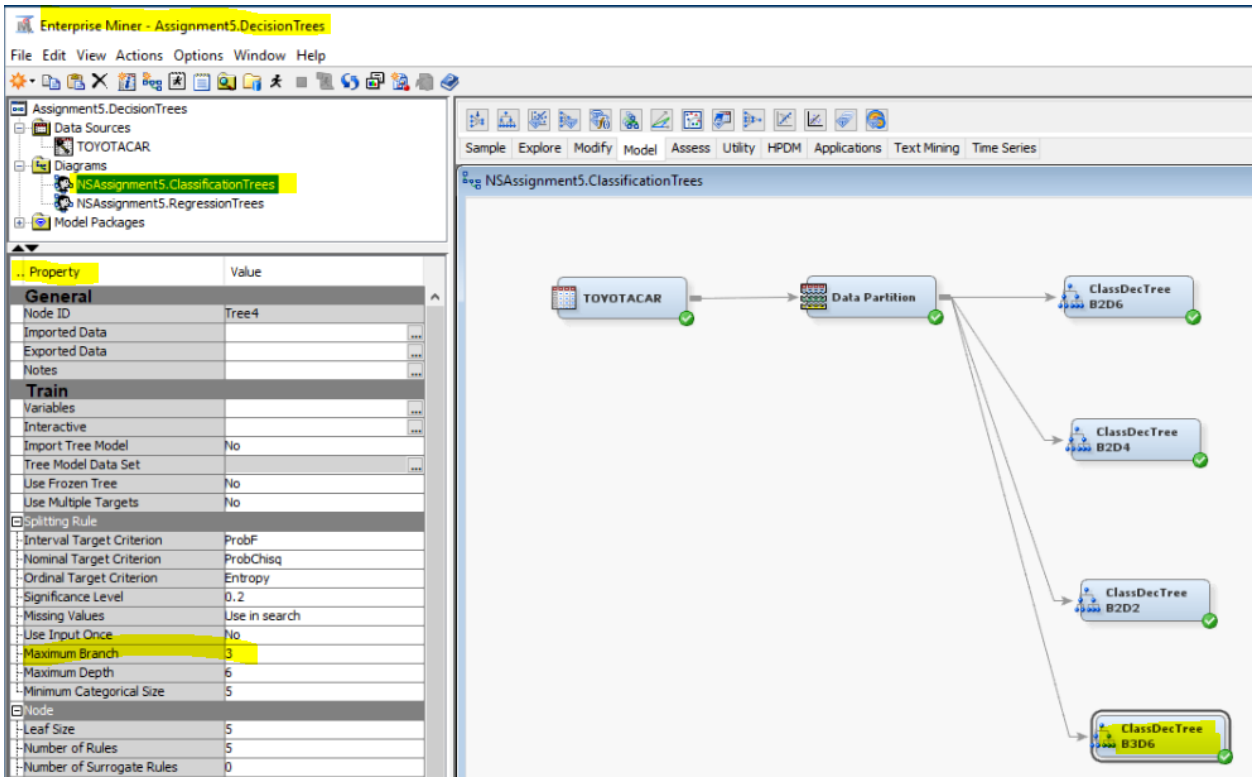


Figure 18

49. Examine the Results window for misclassification rate and Tree of **ClassDecTree B3D6** node.

> **Note**: To increase the accuracy of your decision tree, you can change the following settings:
> - Maximum depth: the number of levels the tree can go down (be careful: increasing this number can cause overfitting)
> - Leaf size: the minimum number of training observations that are allowed in a leaf node. Permissible values are integers greater than or equal to 1. The default setting is 5.
> - Split size: the smallest number of training observations that a node must have before it can be split further. Permissible values are integers greater than or equal to 2.

Model Comparison (Comparing various classification trees)

50. Let's compare all the classification decision tree models to find the best performing model.
51. In the *NS.Classification* diagrams, go to the "Assess" tab on the toolbar and click and drag the "Model Comparison" Node (Third node from the left). Rename the Model Comparison Node as **ClassDecTree Model Comparison**
52. Connect **ClassDecTree B2D6** node to **ClassDecTree Model Comparison** Node.
53. Connect **ClassDecTree B2D4** node to **ClassDecTree Model Comparison** Node.
54. Connect **ClassDecTree B2D2** node to **ClassDecTree Model Comparison** Node.
55. Connect **ClassDecTree B3D6** node to **ClassDecTree Model Comparison** Node.
56. Right Click on **ClassDecTree Model Comparison**, click on Run to execute the model comparison node. (*Model comparison node will compare the misclassification rate of all the*

four classification decision tree models and also compare the overfitting and the model with minimum misclassification rate and minimum overfitting will be selected as the best model for classifying the car as expensive or inexpensive).

57. Right Click on **ClassDecTree Model Comparison** and click on Results.
58. In the results, the Fit Statistics window shows the best Selected Model as Y in the first column. The criteria used for identifying the best model is Misclassification rate of validation set and minimum overfitting. Also, the ROC curve shows the comparison of Sensitivity (True Positive) and 1-Specificity (False Positive) for all the compared models. As we can see in below figure the Classification Tree with 2 Branches and 6 Depth is the best Classification identified by model comparison with a misclassification rate of 12.4% and an accuracy of 87.6% (1-Misclassification rate) as shown below.

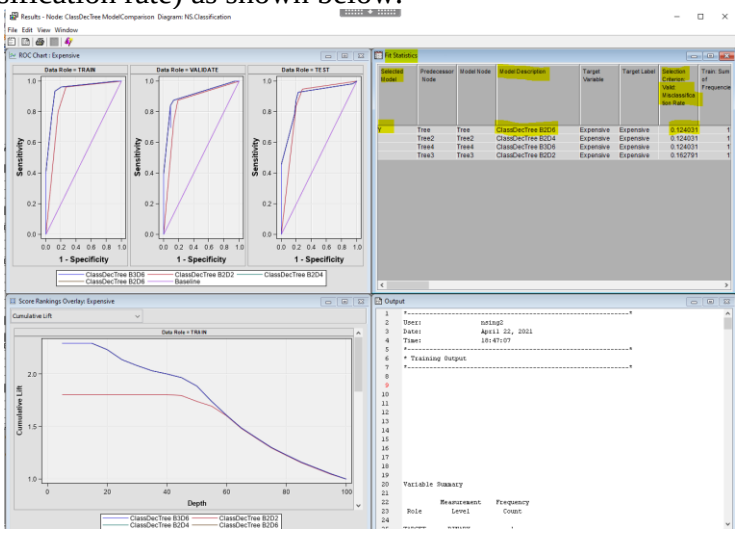


Figure 19

59. To answer our research question 1 (RQ1), we will go to the results (Figure 20) of **ClassDecTree B2D6** and further explore the Tree Window and Variable importance (View->Model->Variable Importance) window to identify the significant/important factors/characteristics to classify the car as expensive or inexpensive just like we did in Question 4.

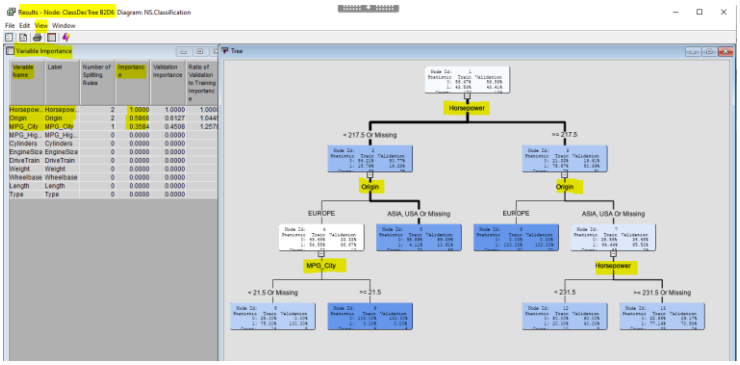
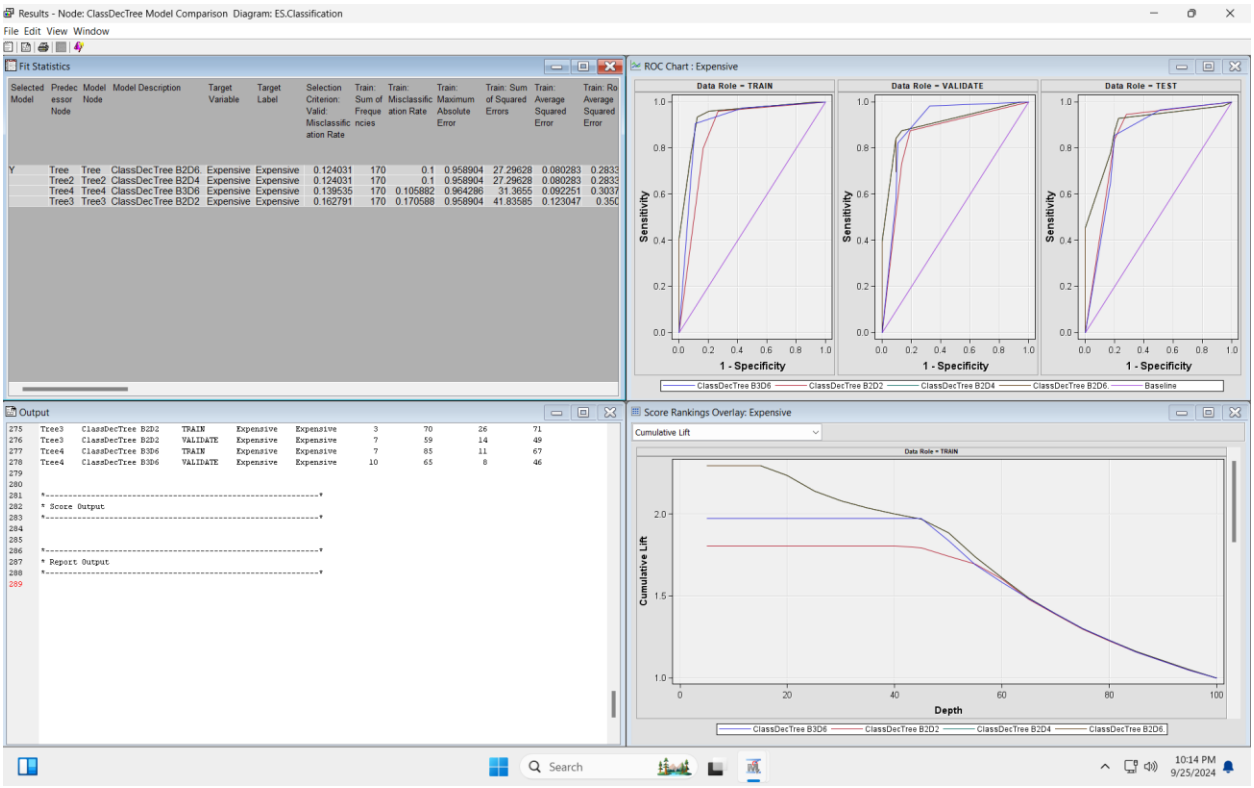


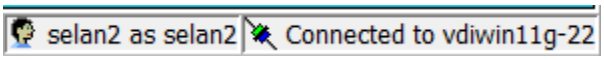
Figure 20

Question 6: Best Model Identified Manually vs Model Comparison

Is the best model identified by ClassDecTree Model Comparison the same as the best model identified by you in Question 5c? Explain in 50-100 words. Provide the screenshot of the results window of the ClassDecTree Model Comparison. Ensure to capture the NetID and machine number while taking screenshots.

The best model identified by ClassDecTree model comparison matches the model I identified in question 5c. In Question 5c, I found that ClassDecTree B2D6 was the most suitable models. The Class Model Comparison determined that ClassDecTree B2D6 was the best model. ClassDecTree B2D6 and ClassDecTree B2D4 have comparable misclassification rates, however ClassDecTree B2D6 is considered the superior model because it was run first in the ClassDecTree Model Comparison.





Comparing classification trees with Logistic Regression supervised techniques

60. Now let's compare the Classification Decision Trees developed so far with Logistic Regression (Exhaustive, Forward, Backward, and Stepwise) to see which of the classification supervised learning technique performs the best classification to classify the new car as expensive or inexpensive.
    a. In the *NS.Classification* diagram, click on the "Model" tab on the toolbar and click, drag and drop four Regression Nodes to the diagram workspace. Regression Node is the third node from right on the Model Tab. Rename the each Regression Node and set the properties as mentioned below:

- Rename one Regression Node as **Exhaustive Regression** and keep the default property. You will see the **Selection Model in Property window for this node is None** and that means it uses the Exhaustive Regression algorithm.

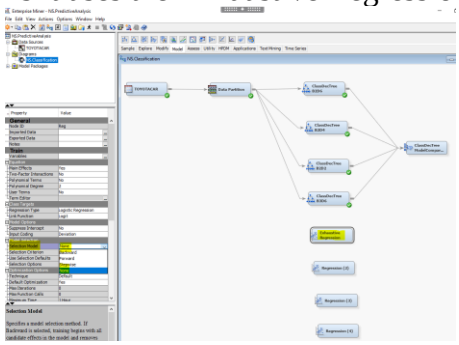


Figure 21

- Rename second Regression Node as **Forward Regression** and set the **Selection Model to Forward in the Property window to use the Forward regression** algorithm.

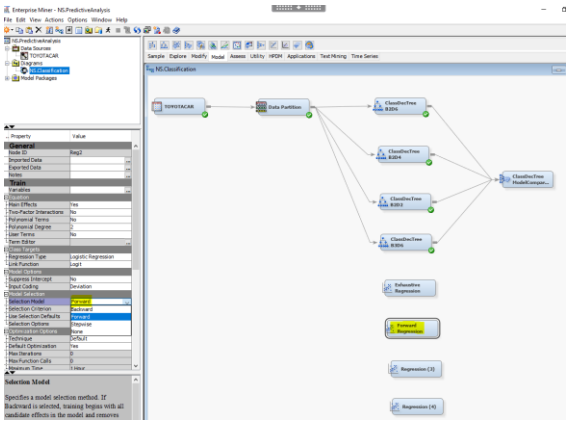


Figure 22

- Rename third Regression Node as **Backward Regression** and set the **Selection Model to Backward in the Property window to use the Backward regression** algorithm.

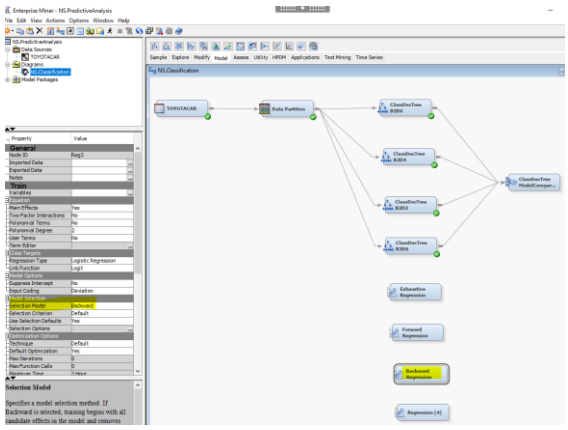


Figure 23

- Rename fourth Regression Node as **Stepwise Regression** and set the **Selection Model to Stepwise in the Property window to use the Stepwise regression** algorithm.

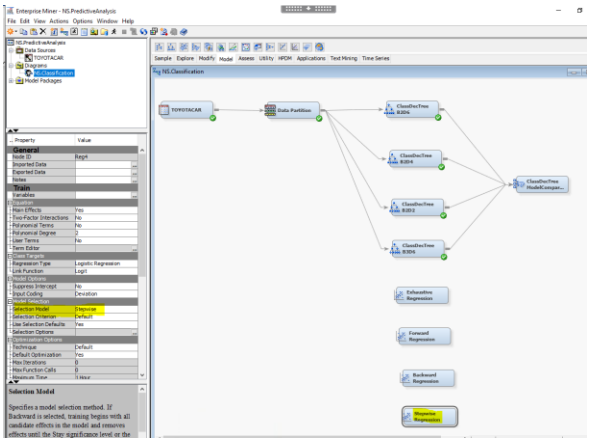


Figure 24

b. Now go to the Assess tab and add a new Model Comparison Node to the diagram. Rename the Model Comparison node as **LogisticReg ModelComparison**
c. Now connect the all the Regression nodes **from the Data Partition Node** and **to the LogisticReg ModelComparison** node as shown in Figure 23.

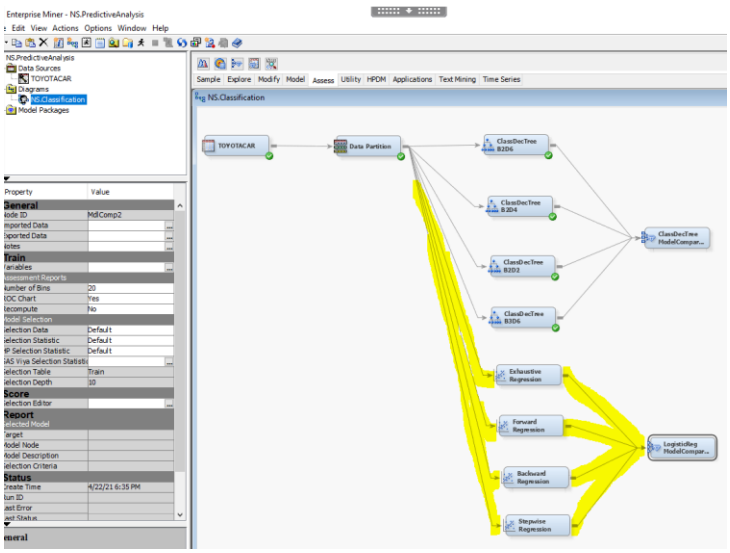


Figure 25

61. Now add one more Model comparison node from Assess Model to your diagram and rename it as Class Model Comparison. Connect it from ClassDecTree Model Comparison and LogisticReg ModelComparison nodes.

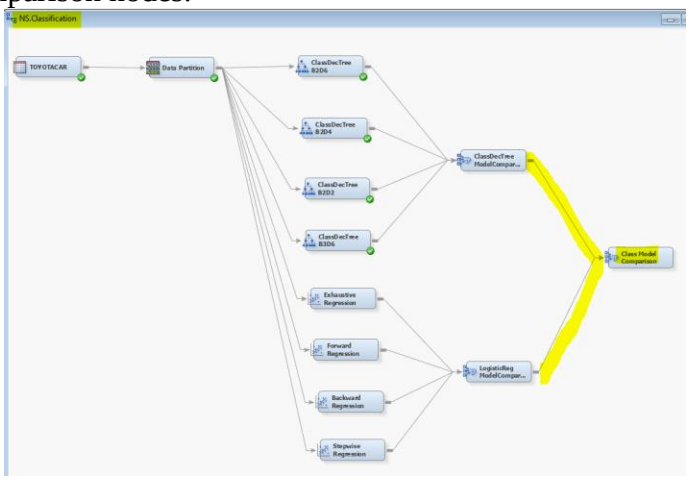


Figure 26

Note: This will help you to perform rigorous analysis of your data to develop the best and most accurate classification prediction model using various types of Classification Tree and Logistic Regression algorithms.

62. Right click the Class Model Comparison node to execute and see the results to identify the best model to classify a car as expensive or inexpensive.

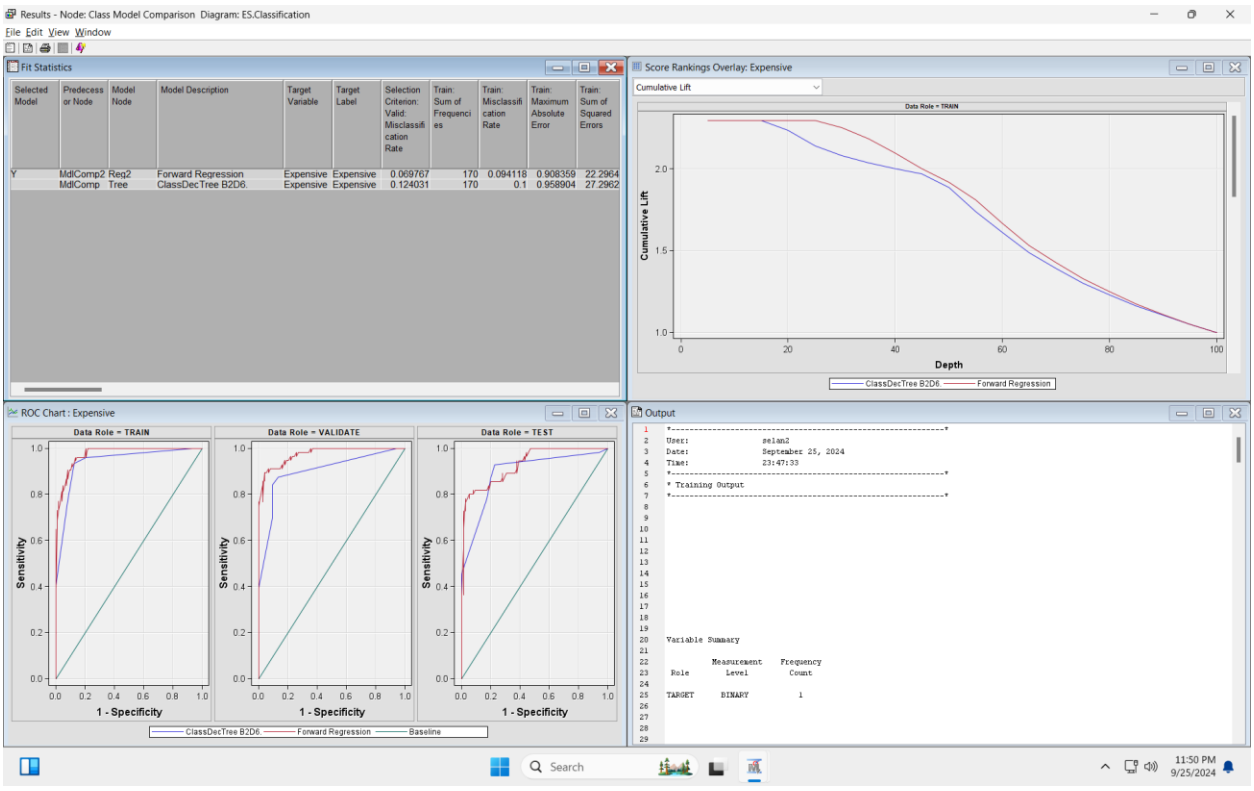
Question 7: Accuracy of Best Classification Model

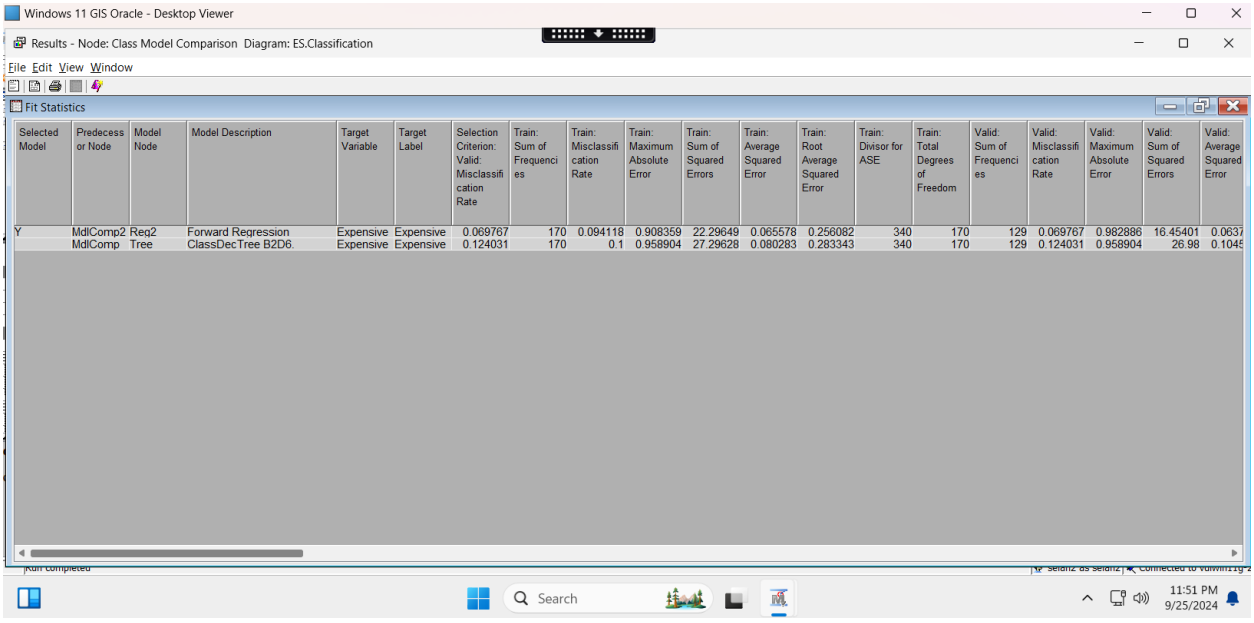
Explain in 50-100 words which model is selected by SAS for you as the best model and what is the accuracy of the best model? Also, provide the screenshot of the Results Window of the Class Model Comparison node ensuring you capture:

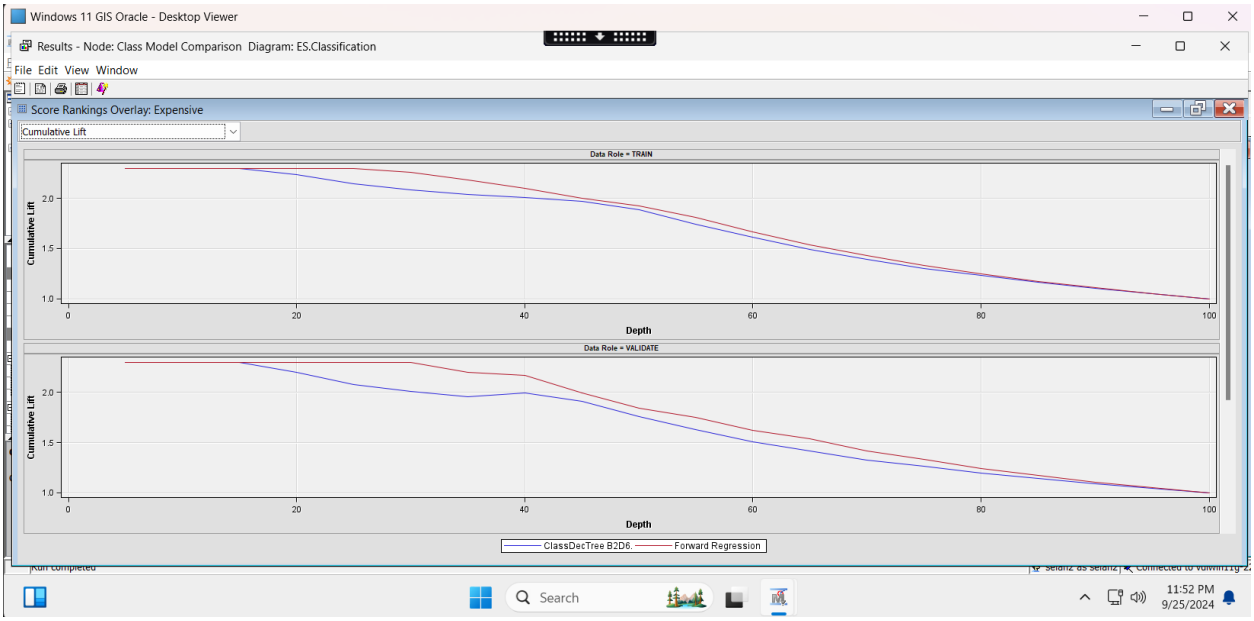
- **ROC Chart: Expensive**

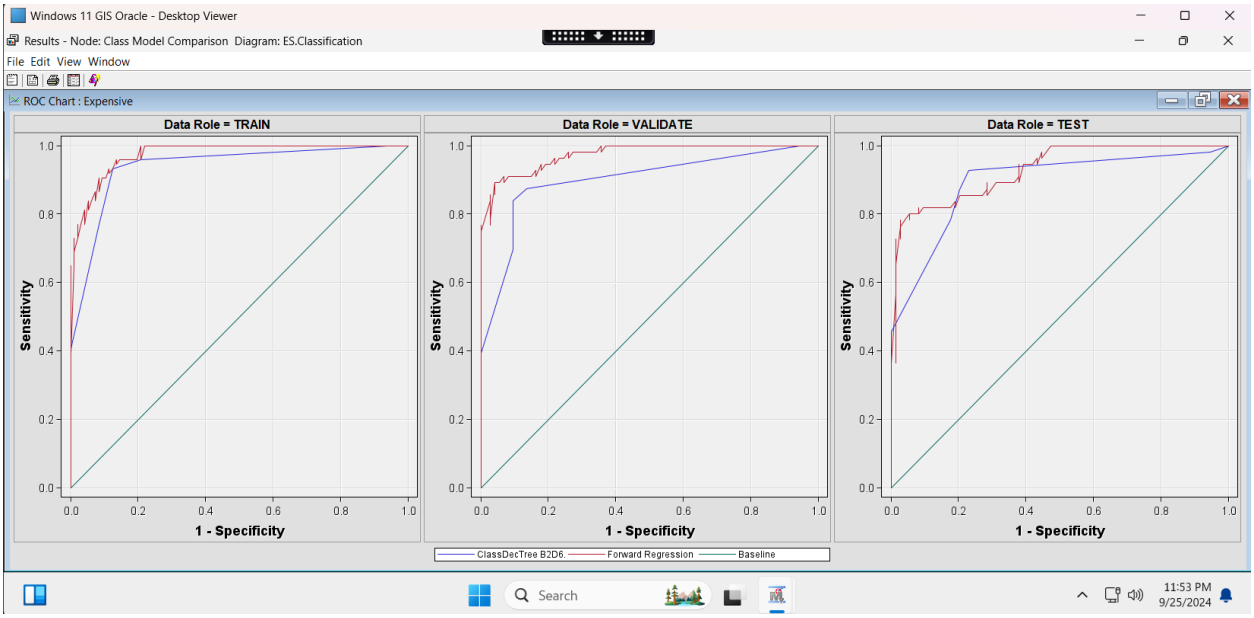
- **Output Window**
- **Fit Statistics Window**
- **Score Rankings Overlay: Expensive**

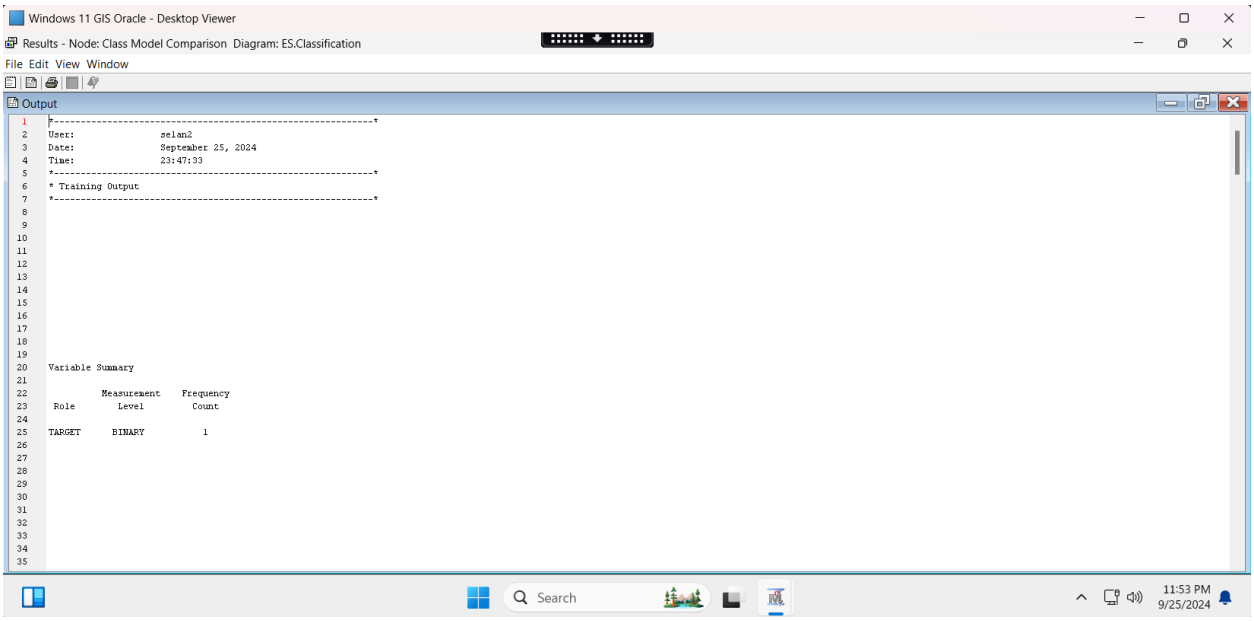
along with your NetID and Machine number. Insert the screenshot here highlighting the best model in the Fit Statistics Window.

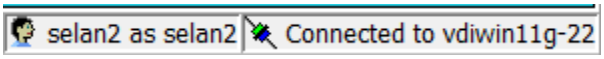





Windows 11 GIS Oracle - Desktop Viewer
Results - Node: Class Model Comparison Diagram: ES.Classification
File Edit View Window
Score Rankings Overlay: Expensive
Cumulative Lift
Data Role = TRAIN
Cumulative Lift
Depth
Data Role = VALIDATE
Cumulative Lift
Depth
ClassDecTree B2D6.
Forward Regression
Search
11:52 PM
9/25/2024


Windows 11 GIS Oracle - Desktop Viewer
Results - Node: Class Model Comparison Diagram: ES.Classification
File Edit View Window
ROC Chart : Expensive
Data Role = TRAIN
Data Role = VALIDATE
Data Role = TEST
Sensitivity
1 - Specificity
ClassDecTree B2D6.
Forward Regression
Baseline
Search
11:53 PM
9/25/2024





Accuracy:

F	G	H	I	J	K	L	M	N	O	P
Model						FALSE	TRUE	FALSE	TRUE	
Node	Model Description		Data Role	Target	Label	Negative	Negative	Positive	Positive	Accuracy
Tree	ClassDecTree	B2D6.	TRAIN	Expensive	Expensive	5	84	12	69	0.9
Tree	ClassDecTree	B2D6.	VALIDATE	Expensive	Expensive	9	66	7	47	0.875969
Reg2	Forward	Regressior	TRAIN	Expensive	Expensive	8	88	8	66	0.905882
Reg2	Forward	Regressior	VALIDATE	Expensive	Expensive	6	70	3	50	0.930233

Accuracy = 1−Misclassification Rate.

For the Reg2 (Forward Regression) model, the misclassification rate on the validation set is 0.070, so the accuracy is:

Accuracy=1−0.070=0.930 or 93%

For the Tree (ClassDecTree) model, the misclassification rate on the validation set is 0.070, so the accuracy is:

Accuracy=1−0.124=0.876 or 87.6%

SAS selected the Forward Regression model (Reg2) as the best model based on the validation misclassification rate, which was 0.070, compared to 0.124 for the Class Decision Tree model. This indicates that the Forward Regression model had the lower error rate in predicting the correct outcomes on the validation dataset, making it the best choice. The model's ROC index is 0.974, indicating a high accuracy in predicting the target variable. In comparison, the decision tree (ClassDecTree B2D6) had a higher misclassification rate of 0.124 and a lower ROC index of 0.900, making the Forward Regression model the best option. Thus, the accuracy of the best model (Forward Regression) is reflected in the ROC index, which is 0.974 on the validation set, indicating a high level of accuracy

63. Close the Results Window of **Class Model Comparison** node.
64. Close ***NS.Classification*** Diagram.

Scoring the Best Model

65. After developing the models, we want to finally score the model on the new data set to further validate the accuracy of the best model on new data. This is also referred to as deploying the developed predictive model on new data. We will perform scoring on the classification supervised models of Figure 26.
66. **Download the ScoreToyotaCar.sas7bdat** available with Assignment 3 on canvas to SAS EM Assignment folder on UISCitrix as shown in Figure 27.

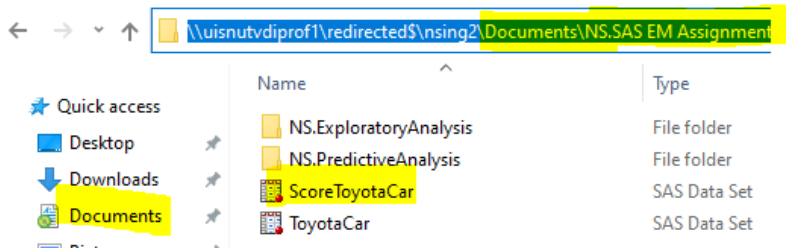


Figure 27

67. Go to the NS.PredictiveAnalysis Project opened in SAS Enterprise Miner. Right Click on **Data Sources** on the left project pane to add the *ScoreToyotaCar* data to the project. Follow the **Create Data Source** Wizard to add the ScoreToyotaCar data. (You can refer to Step 5.7 of lesson plan in case of any doubts related to adding the data set to the project).

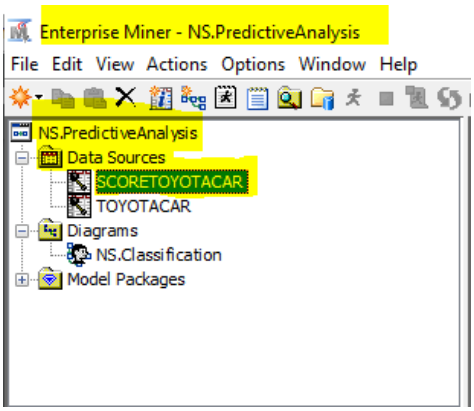


Figure 28

68. Now double click the ***NS.Classification*** diagram to open the workspace as shown in Figure 29.

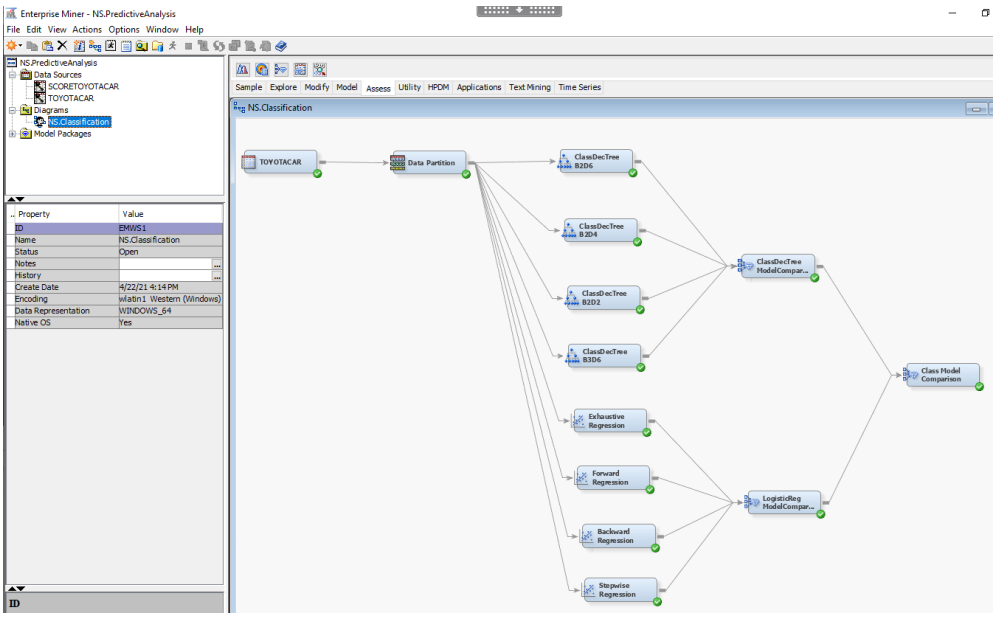


Figure 29

69. Go to the "Assess" tab on toolbar of NS.Classification. Drag the "Score" node from Assess tab to the diagram (Score Node is the second from right on the Assess tab).
70. Connect the Class Model Comparison Node to the Score Node.

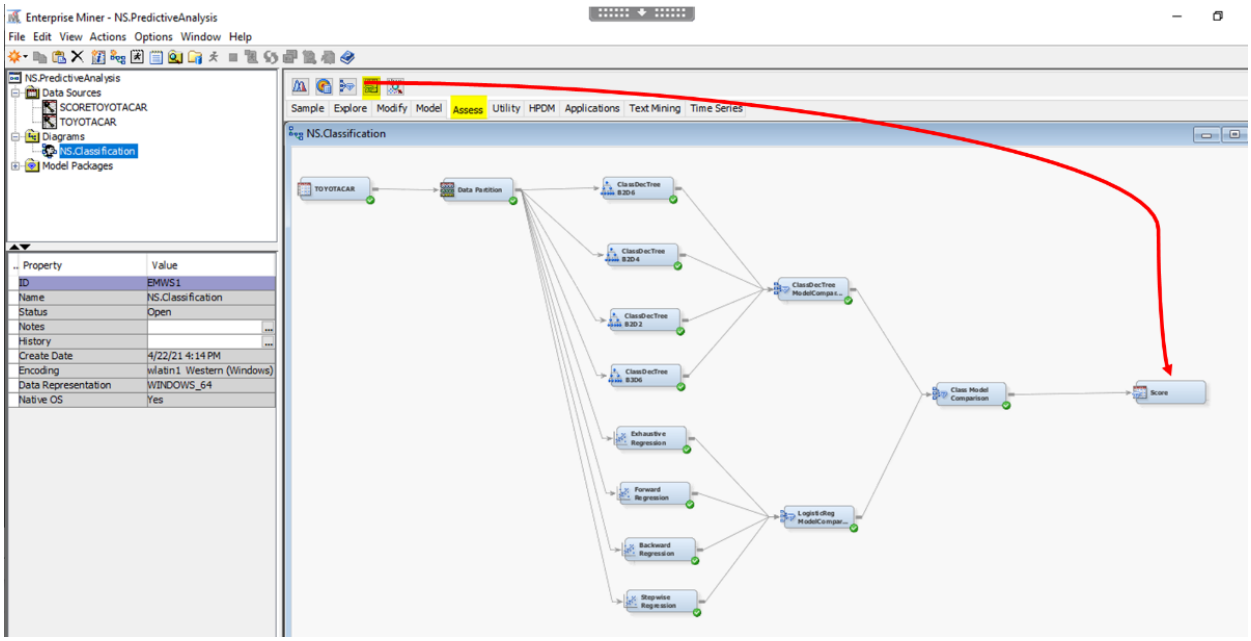


Figure 30

71. Drag and Drop ScoreToyotaCar data from Project Pane to the diagram. Connect ScoreToyotaCar node to the Score Node as shown in Figure 31.
72. Click on ScoreToyotaCar node to open the properties window. **In the properties window of ScoreToyotaCar node on the left pane locate the Role property under Train properties. Set the Role to Score as shown below (Figure 31):**

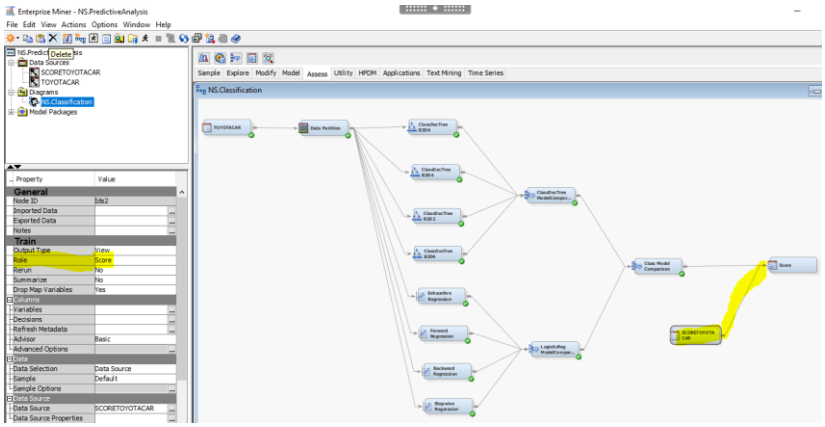


Figure 31

73. Right Click on *ScoreToyotaCar* Node (Figure 31) and Click on Edit Variables. Observe that the roles of all variables are set as Input variables. Also, the levels (type of variables) are kept as default. This means we do not specify the roles and levels of variables, but the final best model identified in Step 62 will assign the roles and levels of the significant/important predictors to predict the outcome variable. This is referred to as scoring the model on new data.
74. Right click on Score Node and click Run. It will execute the score node which will use the best model from Class Model Comparison Node (step 62) and deploy it on the ScoreToyotaCar

data. This scoring will identify the input and target variables based on the best model identified by the Class ModelComparison Node and will classify the new observations in the score data using the classification model developed by us.

75. After the Score node is executed successfully click on Results of the Score Node.

Question 8: Scoring Results

Question 8a

Take the screenshot of the Results Window of Score Node highlighting the sections highlighted in below screenshot. Replace the below screenshot with your screenshot.

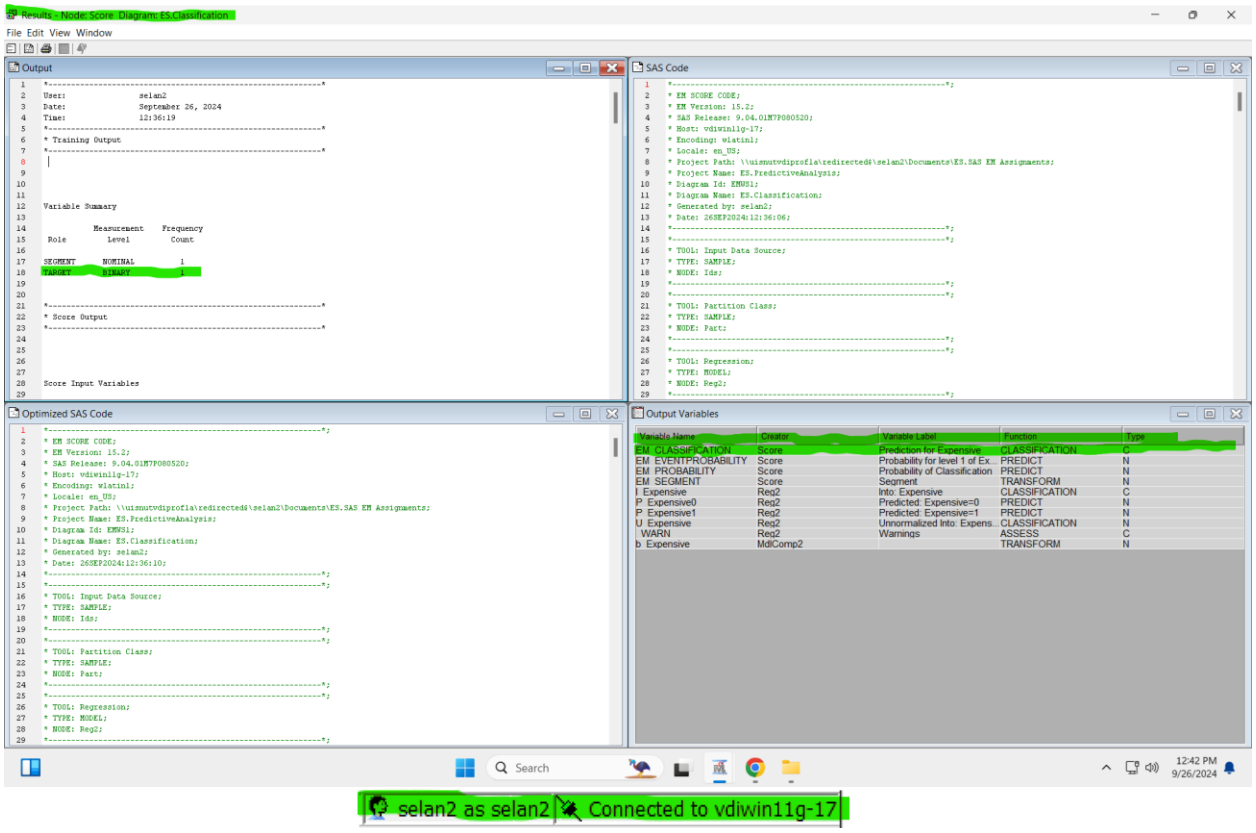


Figure 32

Note: In the Output Variables Window you will see a variable with name EM_Classification and label Prediction for Expensive. This means in the final scored data the Prediction for Expensive variable will tell us based on the characteristics of car if the car is classified as expensive (1) or inexpensive (0).

76. As shown in Figure 33, to explore and analyze the score data, click on Score Node. On the left pane in the properties window of Score Node click on the ellipses of Exported Data to open Exported Data – Score Wizard as shown in Figure 33. In the Exported Data – Score Wizard, click on SCORE row. Finally, click on Explore button. This will open the score data window as shown in Figure 34. In the scoretoyotacar there was an expensive variable. So, now we want

to compare the values in Expensive variable with the values in Prediction for Expensive variable for all the observations. This will help in identifying the accuracy of the model on score data and to ensure if the model is accurate enough to roll to the real-world scenario or not. Just like in excel, we can drag and drop the highlighted columns in Figure 34 to bring them next to each other as shown in Figure 35.

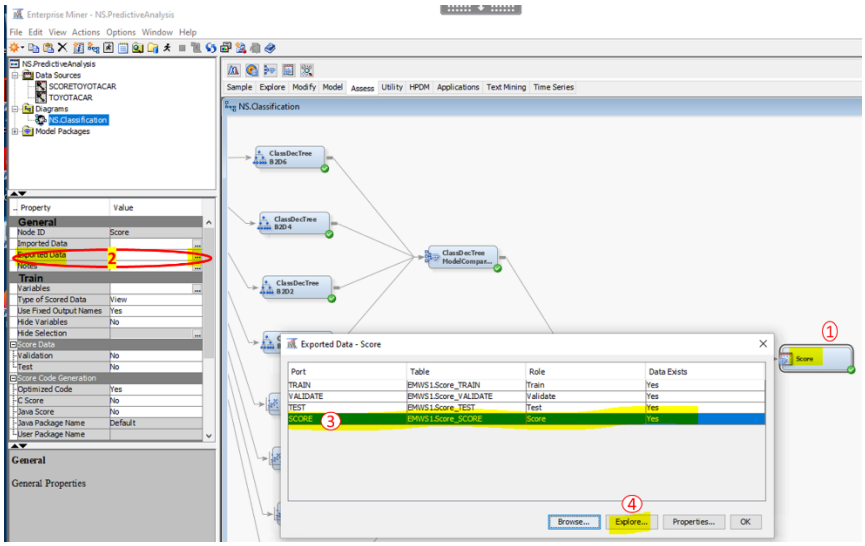


Figure 33

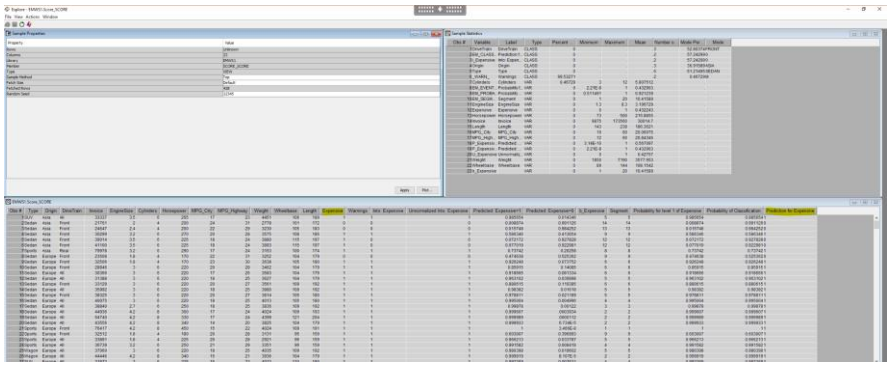


Figure 34

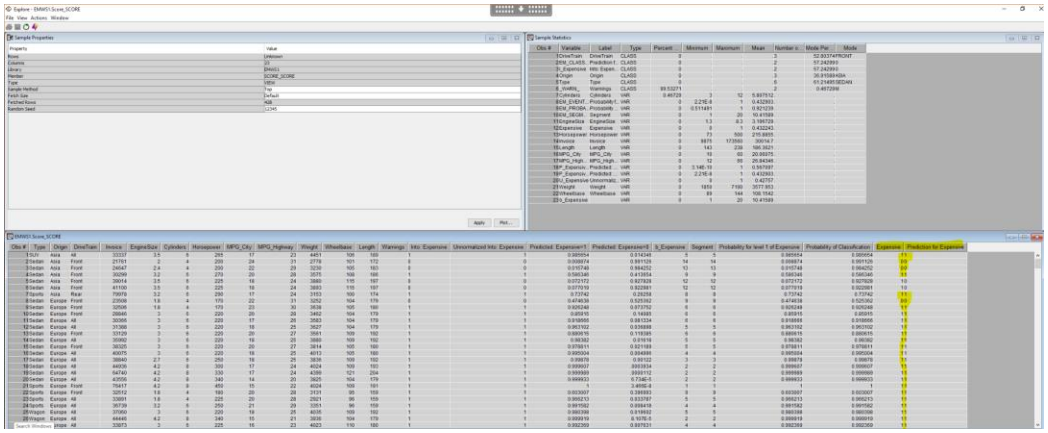


Figure 35

77. As we see exploring the data in Figure 35 is not very user friendly. There is a way around for it. We will save the scored data in excel format and analyze the true positives and true and negatives to identify the accuracy of the model on the scored data. Close the Score data window and Explore Score data window.
78. Click on the Utility tab of NS.Classification diagram. Drag and drop Save Node (Third Node from Right) from Utility tab to the diagram as shown in Figure 36. Also, ensure to set the following properties for Save Node:
    - File Format as Excel Spreadsheet (.xlsx)
    - Directory as SAS EM Assignment under Documents Folder on UIS Citrix
    - FilenamePrefix as PredictiveAnalysis.Scored

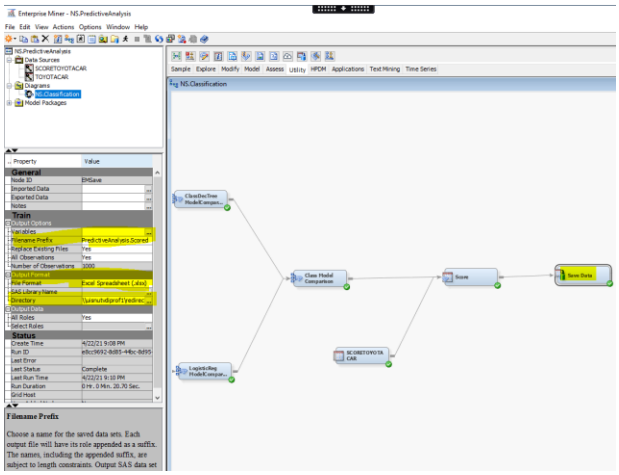


Figure 36

79. Now Run the Save Node and navigate to the Directory NS.SAS EM Assignment under Documents Folder on UIS Citrix. You will see an excel file named as PredtiveAnalysis.Scored_SCORE. Open this file and explore the data. More specifically, to recall as mentioned in the Note after Figure 32, compare the values in Expensive column (Actual Value) with values in EM_Classification column (Predicted Values) to answer Question 8. *(Hint: To do the comparison you can save the score data in an excel format using the "Save" Node which is available under Utility tab of SAS Enterprise Miner. Kindly ensure to set the File Format to .xlsx and select the Directory where you want to save the excel file as shown in Figure 36.)*

Question 8b

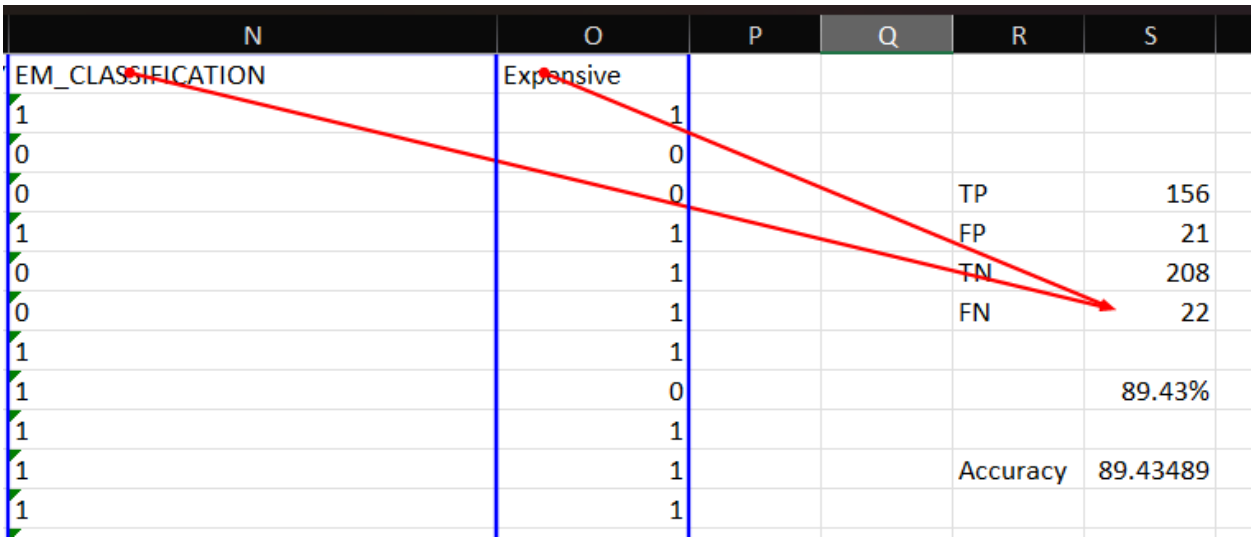
Based on the analysis and comparison of "Expensive" and "EM_Classification" columns in Step 79 answer the questions in below table (Table 4). This will help in identifying the accuracy of your developed model. Also, discuss the percentage accuracy of your developed model in 50-100 words using the analysis from Table 4. *(Hint: To do the comparison you can save the score data in an excel format using the "Save" Node which is available under Utility tab of SAS Enterprise Miner. Kindly ensure to set the File Format to .xlsx and select the Directory where you want to save the excel file as shown in Figure 36)*

Questions	Analysis
How many Expensive cars (1) were accurately classified as Expensive (1) while scoring the best model? (True Positives)	**156**
How many Expensive cars (1) were inaccurately classified as Inexpensive (0) while scoring the best model? (False Negatives)	**22**
How many Inexpensive cars (0) were accurately classified as Inexpensive (0) while scoring the best model? (True Negatives)	**208**
How many Inexpensive cars (0) were inaccurately classified as Expensive (1) while scoring the best model? (False Positives)	**21**

Table 4. Analyzing the score data for model accuracy

Accuracy:

According to my result, The model's accuracy is approximately 89.43%, meaning that about 89% of the cars were correctly classified as either expensive or inexpensive. This suggests that the model performs well, but there is still room for improvement in reducing false positives and false negatives, to improve True positives and True negatives.



80. This completes our analysis of Research Question 1.
81. Close the NS.Classification Diagram.

RQ2: Prediction – Predicting the price of the car

82. Now, we already have one diagram in our project. As we have spent tremendous amount of time in developing the model using decision tree and regression. We can reuse the model structure and tweak in some node properties to align with RQ2 in which we want to predict the price of the using the same ToyotaCar data set.

Extracting and Importing a SAS Enterprise Miner Project Diagram

83. In your NS.PredictiveAnalysis Project, double click NS.Classification Diagram to open it.
84. As shown in Figure 37, in project pane right click on NS.Classification and click on Save As. In the Save dialog box Navigate to the NS.PredictiveAnalysis Project Folder and save the File as NS.Prediction and keep the File Type as XML Files (*.XML). Click on Save

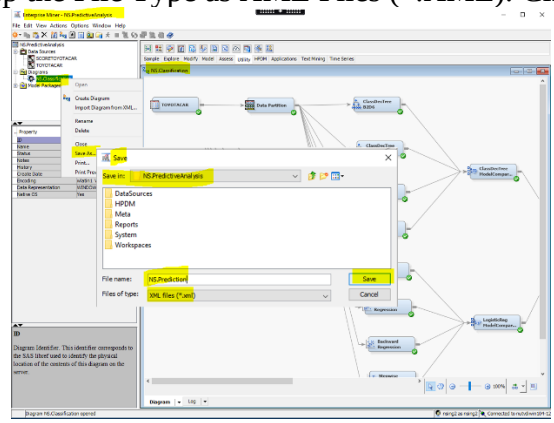


Figure 37

85. As shown in Figure 37, in project pane right click on NS.Classification and click on Save As. In the Save dialog box Navigate to the NS.PredictiveAnalysis Project Folder and save the File as NS.Prediction and keep the File Type as XML Files (*.XML). Click on Save to save the NS.Prediction.XML file in NS.PredictiveAnalysis Project Folder.
86. Now, right click on the Diagrams in project pane of NS.PredictiveAnalysis Project. Click on Import Diagram from XML. In the Open Dialog, navigate to the NS.Prediction XML file as saved in Step 85. Click on NS.Prediction and Click Open as shown in Figure 38.

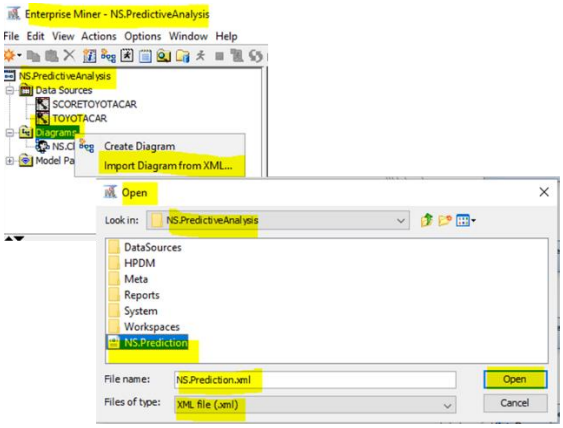


Figure 38

87. It will take some time to Import the Diagram and you will see the NS.Prediction Diagram Imported and Launched in your NS.PredictionAnalysis Project. Don't worry if the NS.Prediction diagram in not opened on the workspace simply double click on it the left project pane to open it as shown in Figure 39. You will observe that the diagram is the duplicate of NS.Classification diagram except the nodes are not executed as there is no green check mark on the nodes.

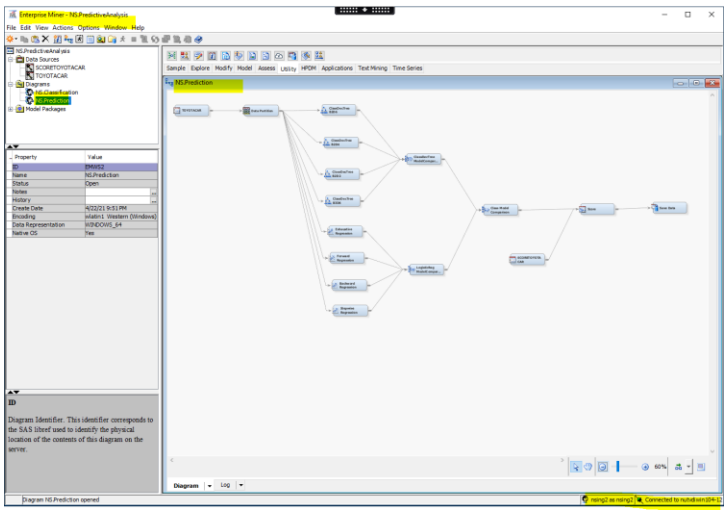


Figure 39

88. To recall, our RQ 2 is to predict the price of the car. So, before executing any node in the NS.Prediction diagram we need to set the variable roles and level according to the requirement of RQ2.

RQ 2 Prediction - Setting the Roles and Levels of Variables

89. In the *NS.Prediction* diagram set the roles and levels of each variable of your ToyotaCar data node as shown in Table 5. To do that right click on ToyotaCar node of *NS.Prediction* diagram and click on "Edit Variables". Clicking on "Edit variables…" will open the Edit Variables screen.

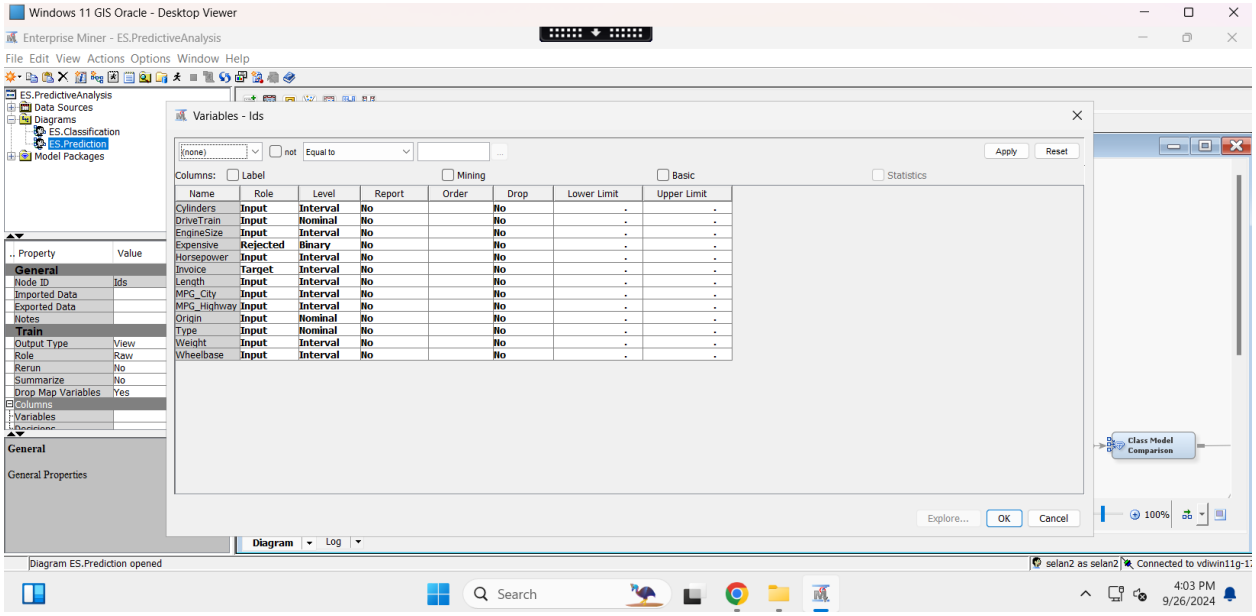
Variable Name	Roles	Variable Type (Levels)
Cylinders	Input	Interval
DriveTrain	Input	Nominal
EngineSize	Input	Interval
Expensive	**Rejected**	Binary
Horsepower	Input	Interval
Invoice	**Target**	**Interval**
Length	Input	Interval
MPG_City	Input	Interval
MPG_Highway	Input	Interval
Origin	Input	Nominal
Type	Input	Nominal
Weight	Input	Interval
Wheelbase	Input	Interval

Table 5. Roles and Levels of ToyotaCar Variables to develop Regression Tree(s) and Multiple Linear Regression Models

Note: This means at this point we are using the Toyotacar data set to answer our RQ2.

Question 9: Variables for RQ2

Provide the screenshot of Edit Variables screen showing the roles and levels of all the variables you have set in previous step for answering RQ2. Ensure to capture NetID and machine number at the lower right corner. Also, explain why we you are using Invoice as a target variable here and why you are rejecting Expensive for Regression Tree (50-100 words).



Invoice is selected as the target variable because it represents the numerical price value that needs to be predicted. The variable Expensive is rejected for the regression tree as it is binary and does not directly aid in predicting a continuous variable like the car's price.

- **Cylinders, EngineSize, Horsepower, Length, MPG_City, MPG_Highway, Weight, Wheelbase are set as Input with Interval levels.**
- **DriveTrain, Origin, Type are set as Input with Nominal levels.**
- **Invoice is the Target (Interval), and Expensive is Rejected for regression.**

90. Click OK to close the Edit Variables window.

Rename the Nodes of Model to Predict the Price of a new Car

91. To recall the names on the decision tree, regression, and model comparison nodes are just labels. However, the property of each node signifies the algorithm to be used. The interesting part about SAS Enterprise Miner model nodes is depending on the data type of Target variable, the model node chooses the specific algorithm. For example, if the target is categorical decision

tree node will use Classification Tree algorithm and if the target is numerical it will execute Regression Tree. Similarly, for Regression analysis if target is categorical Regression node will use Logistic Regression algorithm and if target variable is numerical/interval regression node will implement Multiple Linear Regression (MLR). Another, difference is in the performance evaluation metric for Regression Tree and MLR that uses Squared Errors instead of confusion matrix or misclassification rate to identify the best model. Model with lowest Squared Error is considered as the best model.

92. Based on the discussion in Step 92, we will keep the node properties for all nodes of *NS.Prediction* diagram as is. But for simplification and better understanding, we will just change the names/labels of following nodes:
    - Rename ClassDecTree B2D6 to RegTree B2D6
    - Rename ClassDecTree B2D4 to RegTree B2D4
    - Rename ClassDecTree B2D2 to RegTree B2D2
    - Rename ClassDecTree B3D6 to RegTree B3D6
    - Rename ClassDecTree ModelComparison to RegTree ModelComparison
    - Rename LogisticReg ModelComparison to MLR ModelComparison
    - Rename Class Model Comparison to Predict Model Comparison

93. Right click on Predict Model Comparison to run all the nodes of the process flow. It will take some time to complete the execution as it will execute all the four Regression tree algorithms and all the four Multiple Linear Regression algorithms to identify the best model for predicting the price of the car.

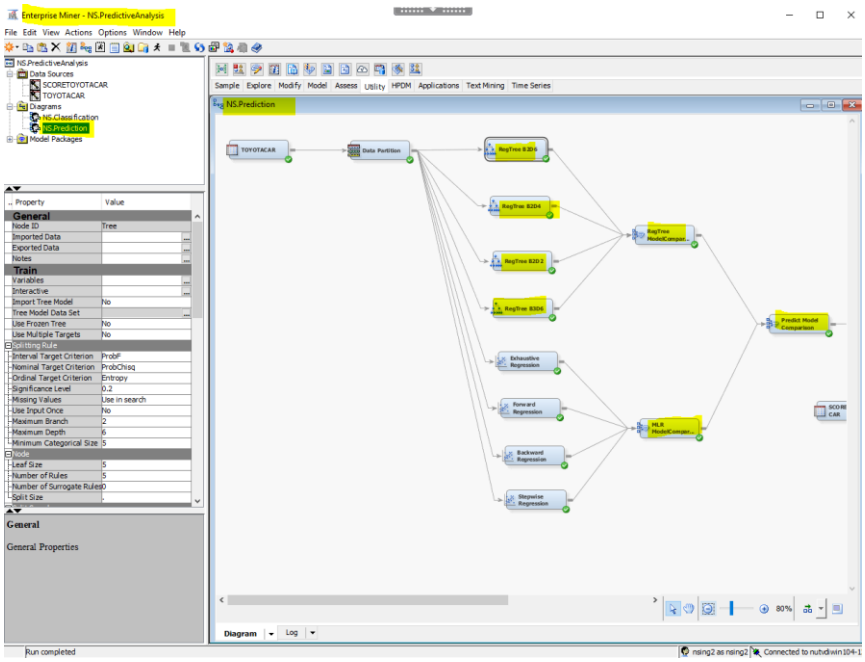


Figure 40

Analyzing the Results of Best Regression Tree

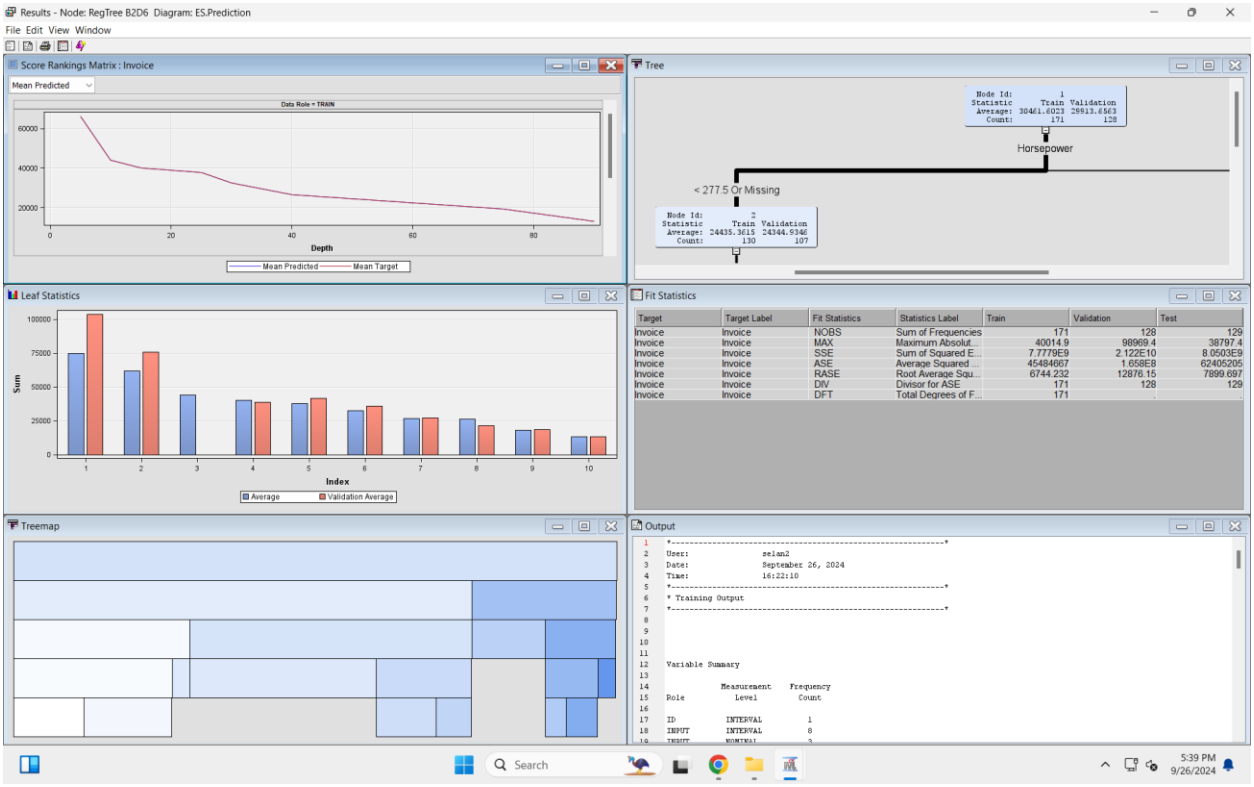
94. *Right click on the RegTree ModelComparison Node and identify the best Regression Tree.*

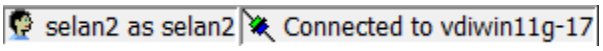
95. *Navigate to the Results window of your Best Regression Tree and Look at the Fit Statistics window. Under "Fit Statistics" column, find "_ASE_" (this is the Average Squared Error value).*

Question 10: Best Regression Tree for Predicting the Price of Car

Question 10a

Provide a screenshot of the Results window of the Best Regression Tree. Ensure to capture the NetID and machine number while taking screenshots.





Fit Statistics

Target=Invoice Target Label=Invoice

Fit Statistics	Statistics Label	Train	Validation	Test
NOBS	Sum of Frequencies	171.00	128.00	129.00
MAX	Maximum Absolute Error	40014.90	98969.40	38797.40
SSE	Sum of Squared Errors	7777878024.19	21221788908.54	8050271495.83
ASE	Average Squared Error	45484666.81	165795225.85	62405205.39
RASE	Root Average Squared Error	6744.23	12876.15	7899.70
DIV	Divisor for ASE	171.00	128.00	129.00
DFT	Total Degrees of Freedom	171.00	.	.

What is the average squared error for train set?	**45484666.81**
What is the average squared error for validation set?	**165795225.85**
What is the average squared error for Test set?	**62405205.39**

Question 10b

Based on the analysis in question 10a and Score Rankings Overlay in Results Window, is there any evidence of overfitting in the best Regression Tree prediction model? (*Note: Overfitting exists when the model performs well on training, but not so well in validation/test sets).*

Based on the analysis in question 10a and the Score Rankings Overlay, there is evidence of potential overfitting in the best regression tree model(B2D6). Overfitting occurs when a model performs very well on the training set but does not generalize well to new, unseen data (in this case, the validation set).

In this situation, the training set has a much lower average squared error (ASE) compared to the validation set. The large difference between these errors suggests that the model might be overfitting, as it has learned patterns specific to the training data but does not perform as well on the validation data, which represents unseen data. This discrepancy indicates the model is too tailored to the training data and might not generalize well to other data.

Predict the Price of a New Car using the Best Regression Tree model (Manually)

96. Now, let's predict the price of a new car (manually) using the Tree of your Best Regression tree Model. For example, if Figure 45 is the Tree of your best Regression Tree Model and a new car has Horsepower=280 and Origin=Europe, Type=Sedan, and Wheelbase=150; the price of the car would be predicted using Node ID 21 (We use the "average" value shown in the leaf node to predict the value of target variable) of Figure 45. Since Node 21 is **a "leaf" node**, it **can be used to predict the price of new car**. W**e use the "average" value of Train**

data in the leaf node to make predictions. The Train column in this node has value of 61733 for "Average". **Therefore, the car with Horsepower of 280, Origin as Europe, Sedan car with a wheelbase of 150 will have an average price of $61733.00 as highlighted in below tree.**

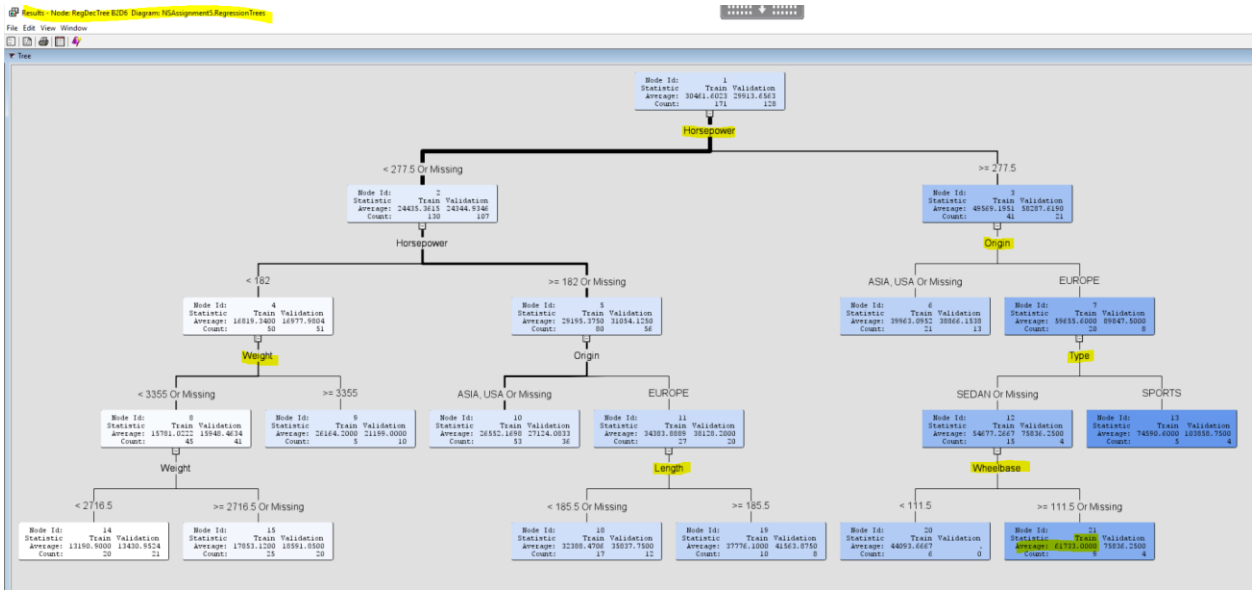


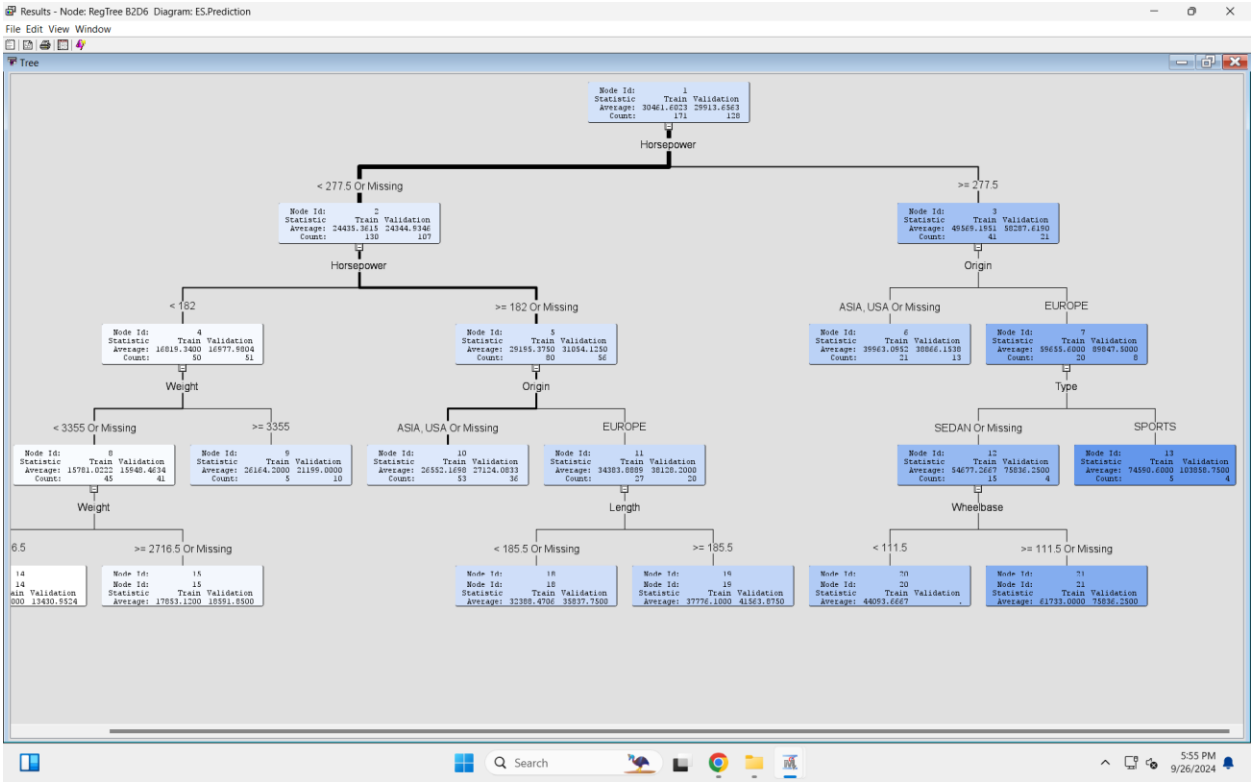
Figure 41

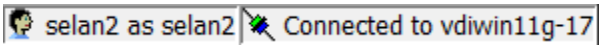
Note: The variables Horsepower, Origin, Weight, Type, Length and Wheelbase are the important variables used to develop this decision tree. The variable importance can also be explored by going to "View" Menu>Model>Variable Importance. The variables with "Importance" of greater than Zero are important variables which are used to develop the regression tree using recursive partitioning algorithm. The variable with "Importance" of 1 is the most important variable and the variable with "Importance" of 0 is not all an important variable to predict the outcome.

Question 11: Predicting the price of a new car

Question 11a

Take the screen shot of your "Tree" Window from the Results Window of your Best Regression tree identified in step 94 and 95. Also, ensure, to capture your NetID and Machine number. Insert the screenshot here.





Question 11b

a. What is the first splitting variable from the root node?

Horsepower

b. If a new car has the following characteristics: Horsepower=265 and Origin=Europe and Length=180; what would be the price for this car based on your regression tree analysis and which Node ID helped you to predict the price of the car (*Note: Follow the nodes in the tree view of Question 10a; use the train column of leaf node for prediction***)?**

Horsepower = 265, it falls into the condition where Horsepower < 277.5 but >= 182. Since the Origin is Europe and the Length is 180, the next relevant node would be Node 18, which is governed by the rule Horsepower between 182 and 277.5, Length < 185.5, and Origin = Europe. **It will predict in Node 18, the predicted invoice price is $32,388.47-$35,837.75 based on the regression analysis**

c. **If a new car has the following values: Horsepower=295, Origin=Europe, and Type=Sports; what would be the price for this car and which Node ID helped you to predict the price of the car (***Note: Follow the nodes in the tree view of Question 10a; use the train column of leaf node for prediction***)?**

Horsepower = 295, it falls into the condition where Horsepower >=277.5. **Since the Origin is Europe and the Type is Sports, the next relevant node would be Node 13.** **It will predict in Node 13, the predicted invoice price is $74,590.70 - $103,858.75 based on the regression analysis**

97. Close the Tree Window and Results Window of your best Regression Tree Model.

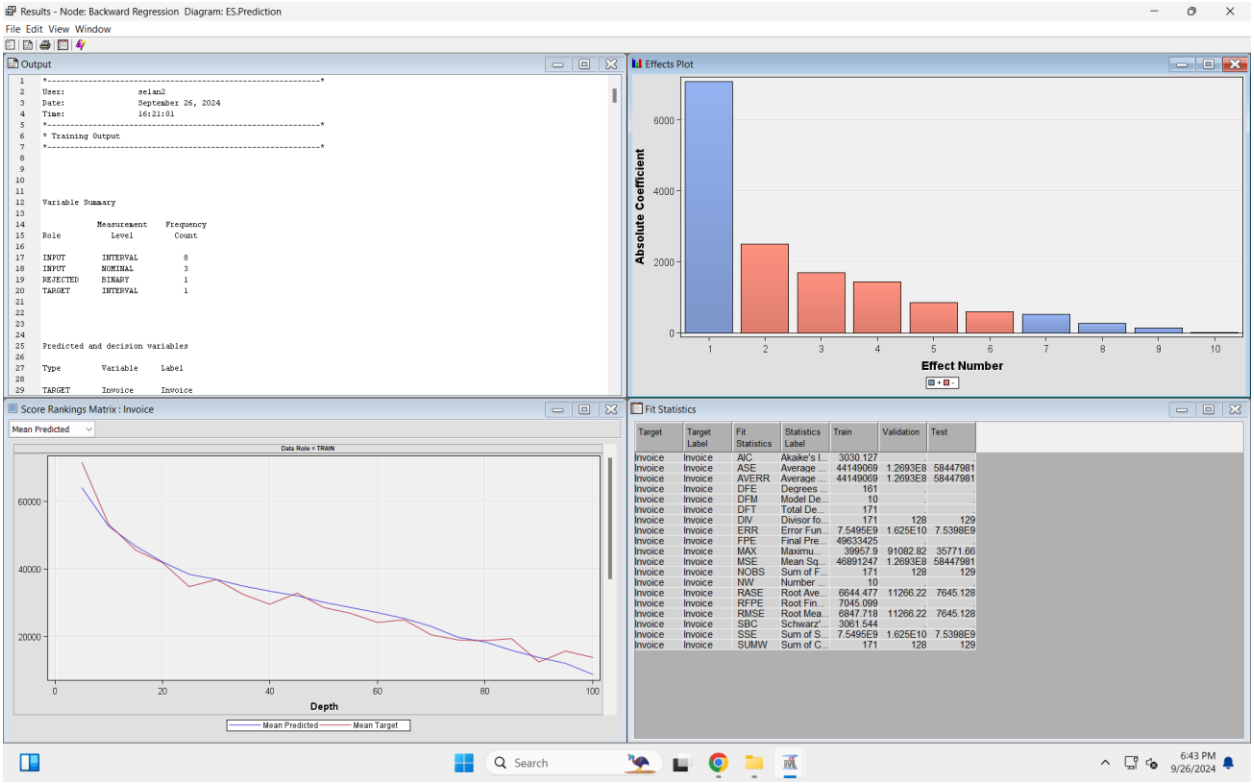
Comparing Regression Trees with Multiple Linear Regression supervised techniques

98. *Right click on the MLR ModelComparison Node and identify the best MLR Model.*
99. *Navigate to the Results window of your Best MLR and Look at the Fit Statistics window. Under "Fit Statistics" column, find "_ASE_" (this is the Average Squared Error value).*

Question 12: Best MLR for Predicting the Price of Car

Question 12a

Provide a screenshot of the results window of the Best MLR. Ensure to capture the NetID and machine number while taking screenshots.



selan2 as selan2 | Connected to vdiwin11g-17

Fit Statistics

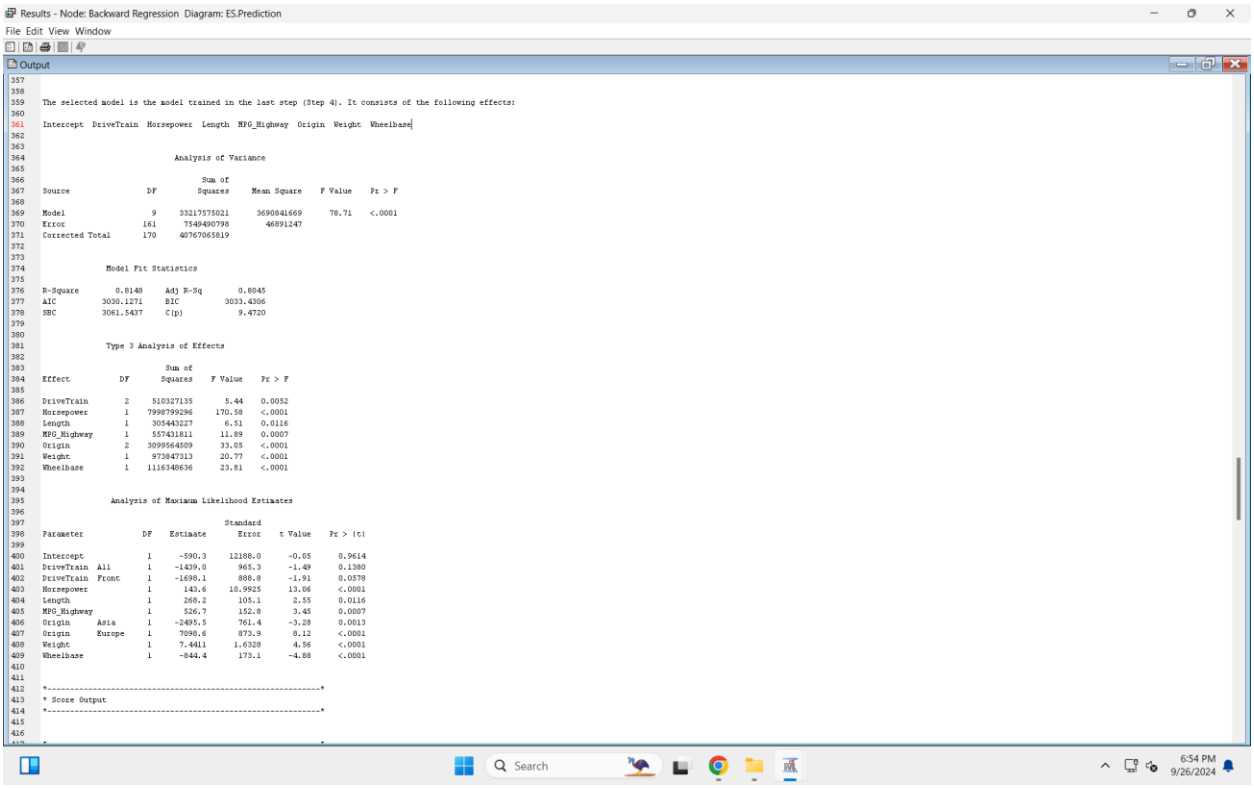
Target=Invoice Target Label=Invoice

Fit Statistics	Statistics Label	Train	Validation	Test
AIC	Akaike's Information Criterion	3030.13	.	.
ASE	Average Squared Error	44149068.99	126927625.01	58447981.42
AVERR	Average Error Function	44149068.99	126927625.01	58447981.42
DFE	Degrees of Freedom for Error	161.00	.	.
DFM	Model Degrees of Freedom	10.00	.	.
DFT	Total Degrees of Freedom	171.00	.	.
DIV	Divisor for ASE	171.00	128.00	129.00
ERR	Error Function	7549490797.59	16246736001.01	7539789602.96
FPE	Final Prediction Error	49633425.39	.	.
MAX	Maximum Absolute Error	39957.90	91082.82	35771.66
MSE	Mean Square Error	46891247.19	126927625.01	58447981.42
NOBS	Sum of Frequencies	171.00	128.00	129.00
_NW_	Number of Estimate Weights	10.00	.	.
RASE	Root Average Sum of Squares	6644.48	11266.22	7645.13
RFPE	Root Final Prediction Error	7045.10	.	.
RMSE	Root Mean Squared Error	6847.72	11266.22	7645.13
SBC	Schwarz's Bayesian Criterion	3061.54	.	.
SSE	Sum of Squared Errors	7549490797.59	16246736001.01	7539789602.96
SUMW	Sum of Case Weights Times Freq	171.00	128.00	129.00

What is the average squared error for train set?	**44149068.99**
What is the average squared error for validation set?	**126927625.01**
What is the average squared error for Test set?	**58447981.42**

Question 12b

Look into the Output Window of your Best MLR Model and enter the following statistical values for your best MLR Model (Enter the values in below table). Also, provide the screenshot of the Output Window ensure to capture the NetID and machine number while taking screenshots.



Analysis of Variance	F-value	78.71
Analysis of Variance	p-value (Pr>F)	<.0001
Model Fit Statistics	Adj R-Sq	0.8045
Model Fit Statistics	C(p)	9.4720

100. To answer question 12b, go to the results of your best MLR Model and maximize the Output Window. For example, if my MLR ModelComparison identifies Backward Regression as the best model, I will go to the Results of Backward Regression Node on NS.Prediction Diagram. In the Results window, I will Maximize the Output Window to further analyze the MLR Backward Regression model results.

- Now scroll down in Output window and **locate the Analysis of Variance section which is immediately after the "The selected Model is the model trained in the last step…."** (Figure 42).

  Note: In Backward, forward, and stepwise regression significant predictors are identified iteratively to maximize the variance and predictive power of the model. So, there will be several Analysis of variance, Model Fit Statistics, Type 3 Analysis of Effects, and Analysis of Maximum Likelihood Estimates sections in the output of these three regression models.

  - The Analysis of Variance, Model Fit Statistics, Type 3 Analysis of Effects, and Analysis of Maximum Likelihood Estimates statistics after the **"The selected Model is the model trained in the last step…."** Section (Figure 42) is always used to identify the significance of predictors and overall model. This will also be used to answer the Table in Question 12b.

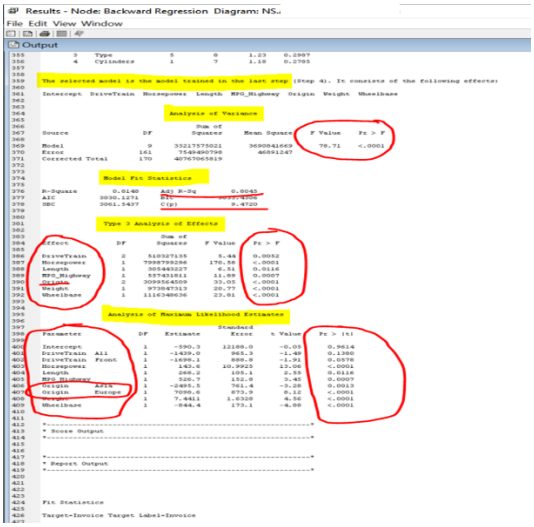


Figure 42

101. Now to explore and find the significant input/predictor variables ($p < 0.05$) which are important to predict the price. To recall, here backward regression is the best model (it can be different for you). The non-significant variables ($p > 0.05$) identified by the regression can be eliminated (rejected) from your data set if you are using Regression for dimension reduction. Nevertheless, regression model will only use important/significant to predict the value of target variable (in our case Price/Invoice).
102. To find the significant variables from the Output window locate the **Analysis of Maximum Likelihood Estimates** after the **"The selected Model is the model trained in the last step…."** Section **(Figure 42) (***Hint: Depending on the type of Regression there could be multiple "Analysis of Maximum Likelihood Estimates" section in the result window. Ensure to use the last/final "Analysis of Maximum Likelihood Estimates" section to find the non-significant variables identified by the specific regression***)**.

Question 13: Significant/Important Variables identified by Best MLR for Predicting Price

Question 13a

Which variables are significant (p<.05)? Provide the list of significant variables. Also, provide a screenshot of the section in the Output window from where significant variables are identified. Ensure to capture the NetID and machine number while taking screenshots.

Important Variables:

Horsepower , Length, MPG_Highway, Origin, Weight, Wheelbase.

Analysis of Maximum Likelihood Estimates

Parameter		DF	Estimate	Standard Error	t Value	Pr > |t|
Intercept		1	-590.3	12188.0	-0.05	0.9614
DriveTrain	All	1	-1439.0	965.3	-1.49	0.1380
DriveTrain	Front	1	-1698.1	888.8	-1.91	0.0578
Horsepower		1	143.6	10.9925	13.06	<.0001
Length		1	268.2	105.1	2.55	0.0116
MPG_Highway		1	526.7	152.8	3.45	0.0007
Origin	Asia	1	-2495.5	761.4	-3.28	0.0013
Origin	Europe	1	7098.6	873.9	8.12	<.0001
Weight		1	7.4411	1.6328	4.56	<.0001
Wheelbase		1	-844.4	173.1	-4.88	<.0001

These are the variables that significantly contribute to predicting the price based on the regression analysis.

Question 13b

Write the regression equation to predict the price of the car using the Intercept and significant variables from your analysis in Question 13a (Figure 42) to predict the price of a Toyotacar. *(Hint: You can refer to the notes provided in this document and chapter slides for MLR to answer this question.* ***E.g. Regression equation:*** $Y_{Price}=\propto+\beta_{DriveTrain}*DriveTrain+\beta_{Horsepower}*Horsepower+\beta_{Length}*Length+\ldots..+\beta_{Wheelbase}*Wheelbase$ when DriveTrain, Horsepower, Length, and Wheelbase are significant variables in the MLR results.)

YPrice=α+β1(Horsepower)+β2(Length)+β3(MPG Highway)+β4(Origin: Asia)+β5 (Origin: Europe)+β6(Weight)+β7(Wheelbase)

YPrice=−590.3+143.6(Horsepower)+268.2(Length)+526.7(MPG Highway)−2495.5(Origin: Asia)+7098.6(Origin: Europe)+7.44(Weight)−844.4(Wheelbase)

In this equation:

- **YPrice represents the predicted price (Invoice) of the Toyota car.**
- **α\alpha is the intercept (constant term in the regression equation).**
- **β1,β2,β3,β4,β5, β6 are the coefficients for Horsepower, Weight, Length, Origin, Wheelbase, and MPG_City respectively**

103. Now right click on the PredictModelComparison Node to identify the best model of predicting the price of the car.

Scoring the Best Model

104. Just like you have done scoring for RQ1. You can perform scoring for RQ 2. Right click on Score node of NS.Prediction to score the best predictive model on ScoreToyotaCar data to predict price of new cars. Use the Save Node to save the scored data in excel file for analysis. Ensure to change the Filename Prefix (e.g., Price.Scored) for Save Node otherwise it will overwrite your Scored excel file for Classification. Now you can compare the values in Price and EM_Prediction column of Price.Scored_SCORE excel file to see the accuracy.

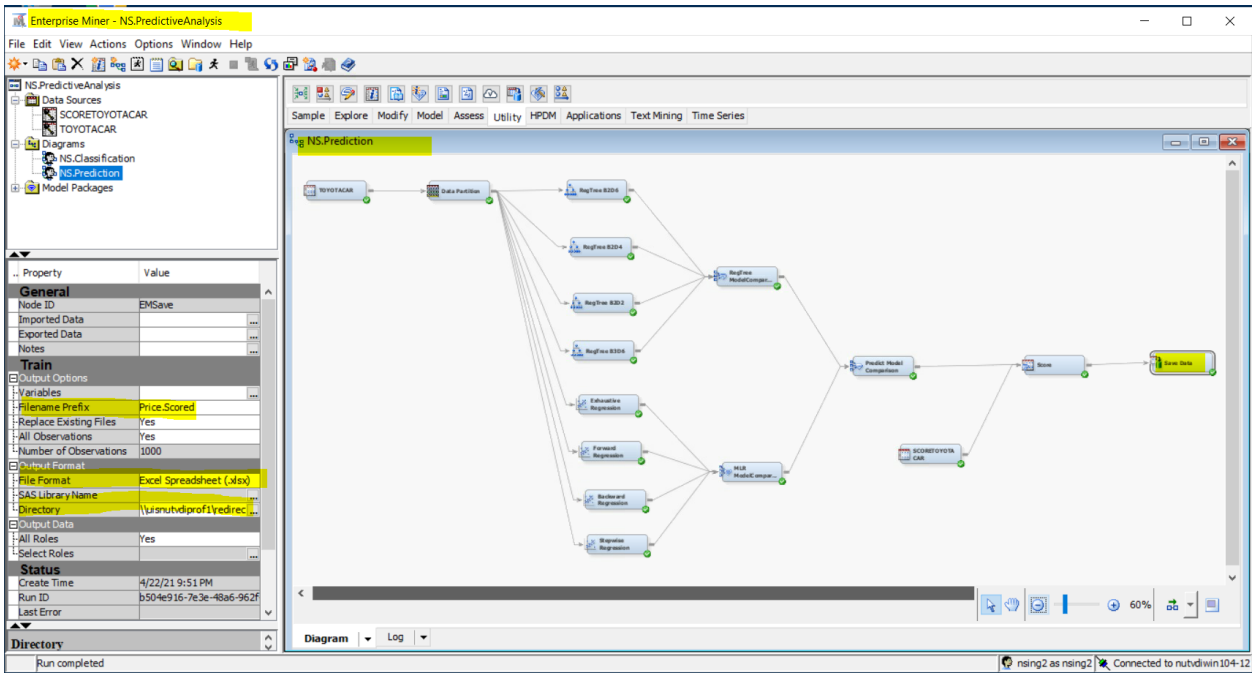


Figure 43

105. Close SAS Enterprise Miner and sign out from UIS Citrix.

Question 14: Learning Outcome to implement in project and Strategies to develop Predictive Models with highest accuracy.

Discuss the key take-away points from Assignment 3 that will be helpful to work on your project. What is the purpose of scoring the best model on score data (new data)? Also, discuss the strategies for developing a better decision tree (150-200 words). (*Note: Assignment 3 includes the whole process of working on predictive analysis part of your project. This assignment includes the basic steps to implement various supervised algorithms to develop the best predictive model and deploy it on the score data. This assignment can be used to perform predictive analysis in your project.***)**

Key Takeaways:

- **Assignment 3 provides insights on the entire predictive modeling process starting from how to apply a supervised learning algorithm, such as classification and regression trees, then we learn about multiple linear regression.**
- **The assignment emphasizes the importance of fine-tuning models through techniques like pruning decision trees to reduce overfitting and selecting appropriate hyperparameters such as the depth and number of branches in trees. Comparing different models using validation metrics helps identify the best performing model.**

- **Scoring Data:**

  The purpose of scoring the best model on new data (score data) is to validate the model's performance and generalizability. This step ensures that the model is not overfitting and performs well on unseen data, which is crucial for making accurate real-world predictions. It helps validate the model's accuracy and ensures that the model generalizes well to new and unseen data, which is crucial for real-world applications.

Strategies:

- **In terms of strategies to develop better predictive models, one essential strategy is minimizing overfitting by adjusting model complexity. This can be done by pruning decision trees to reduce depth, limiting the number of branches, and ensuring that the model does not become too tailored to the training data.**

- **Another strategy is performing cross-validation to test the model on different subsets of data to ensure robustness. Feature selection is also critical, as it helps in identifying the most important variables that influence the prediction outcome, thereby simplifying the model and improving interpretability without sacrificing accuracy.**

- **Lastly, comparing models using validation metrics allows you to pick the most accurate and reliable model for deployment.**

Appendix – Data Set Variable Description

Variable Name	Variable Description
Cylinders	The number of cylinders of the engine
DriveTrain	Whether the automobile is "All" wheel, "Rear" wheel, or "Front" wheel drive
EngineSize	The size of the engine
Expensive	Whether the Invoice price of the automobile is greater than or equal to $28,000 or not (binary variable: 1 if invoice >=28,000; 0 otherwise)
Horsepower	The horsepower of the engine
Invoice	The invoice price of the automobile (i.e., the purchase price)
Length	The length of the automobile
MPG_City	Miles per gallon (in city)
MPG_Highway	Miles per gallon (on highway)
Origin	The origin of the automobile
Type	The type of the automobile (whether hybrid, sedan, sports, SUV, truck, or wagon)
Weight	The weight of the automobile
Wheelbase	The distance between the front and rear axles of the automobile