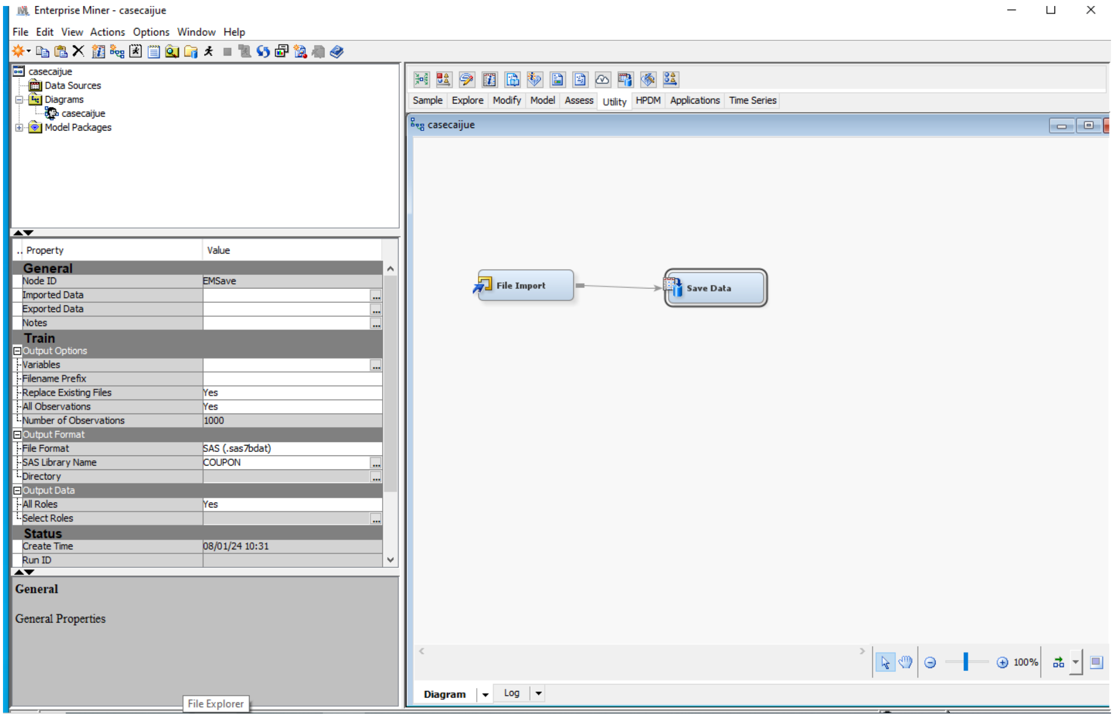
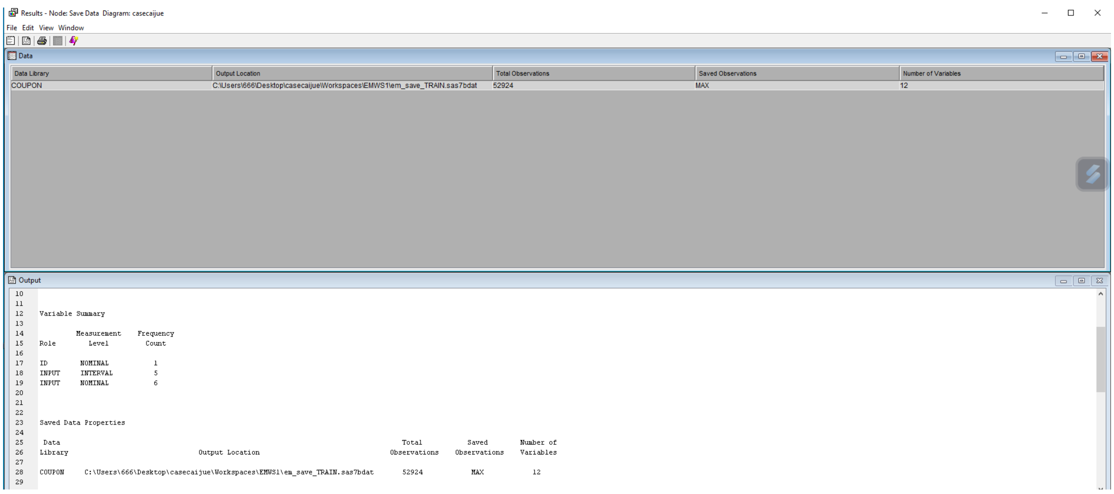
Exploration in SAS

22078878 CAI JUE

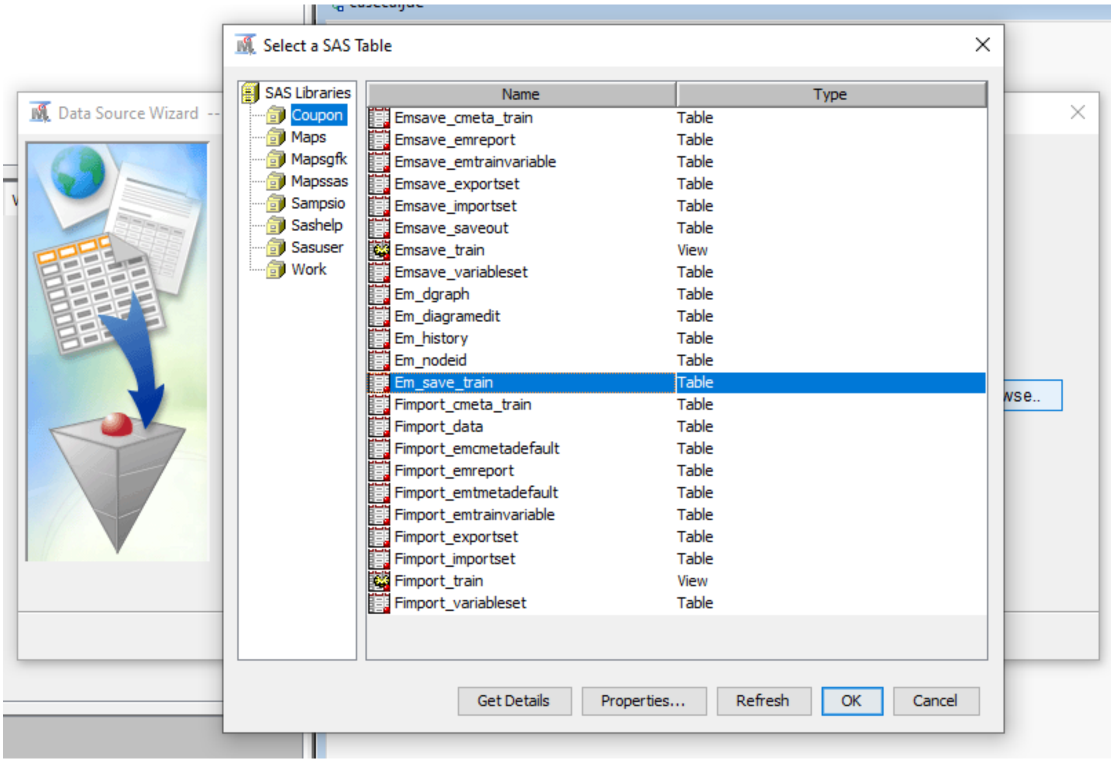
1.Data imported and saved under COUPON’s library

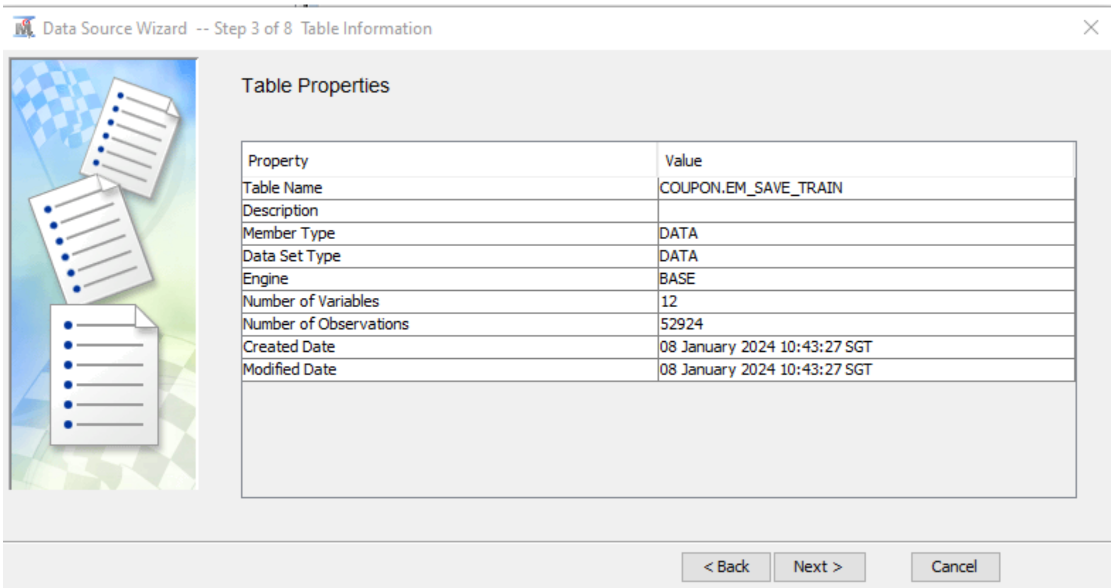


2.Check the name of the save and open this, em_save_TRAIN.sas7bat, for the next step of exploration.

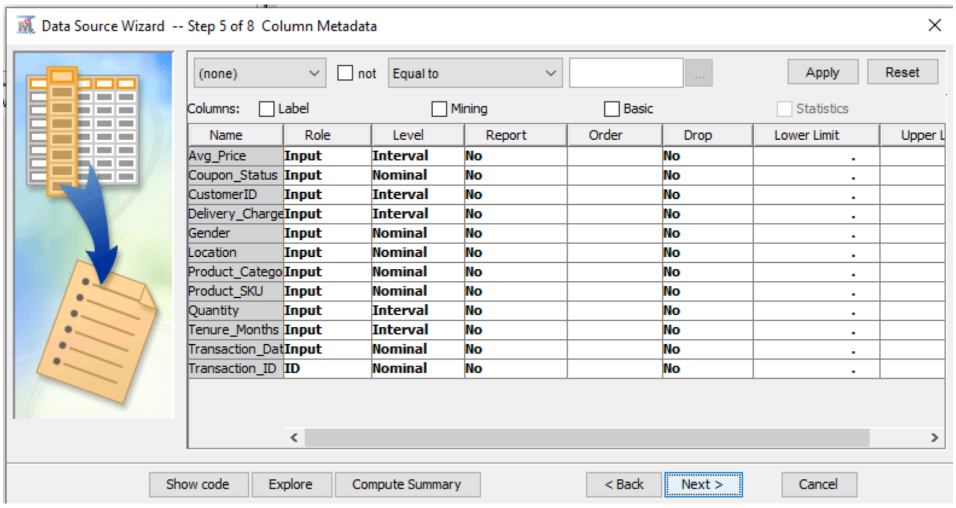


3. Select this file to explore

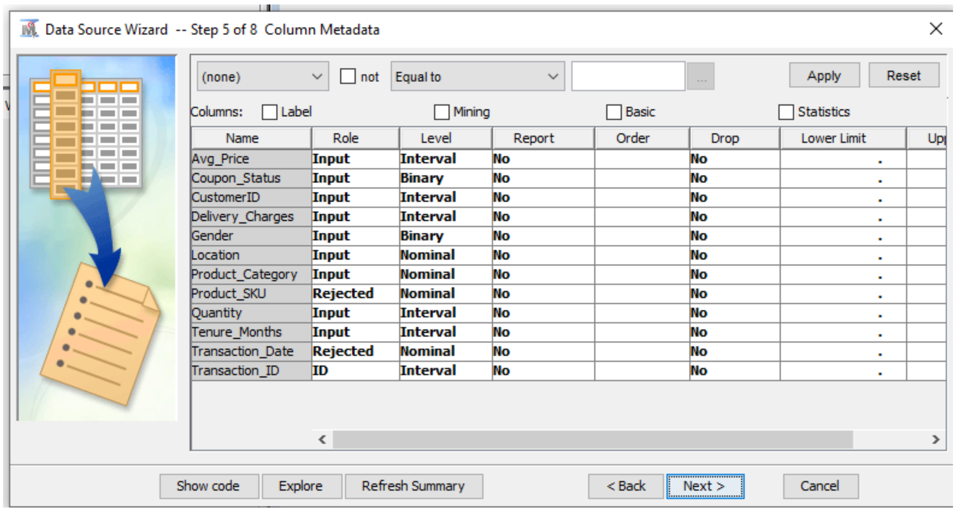




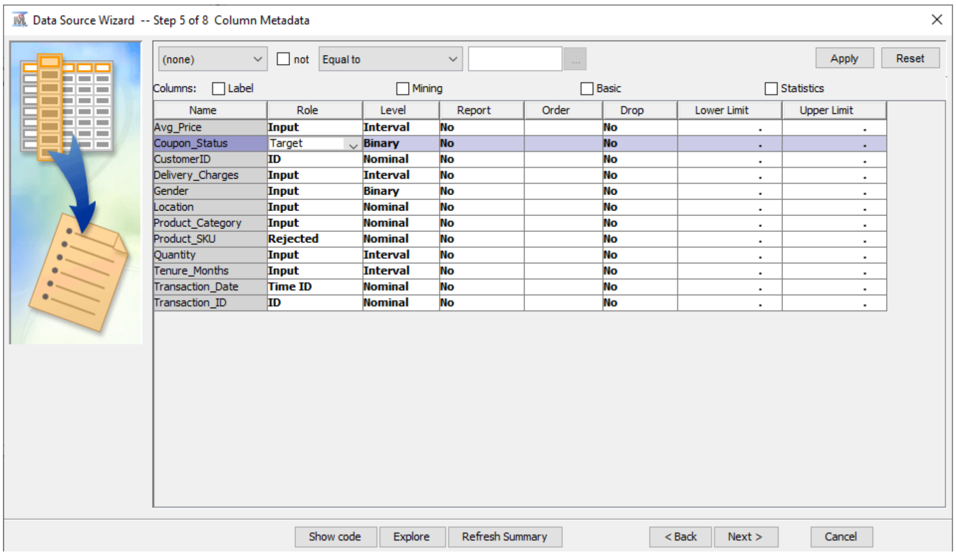
Basic Version:



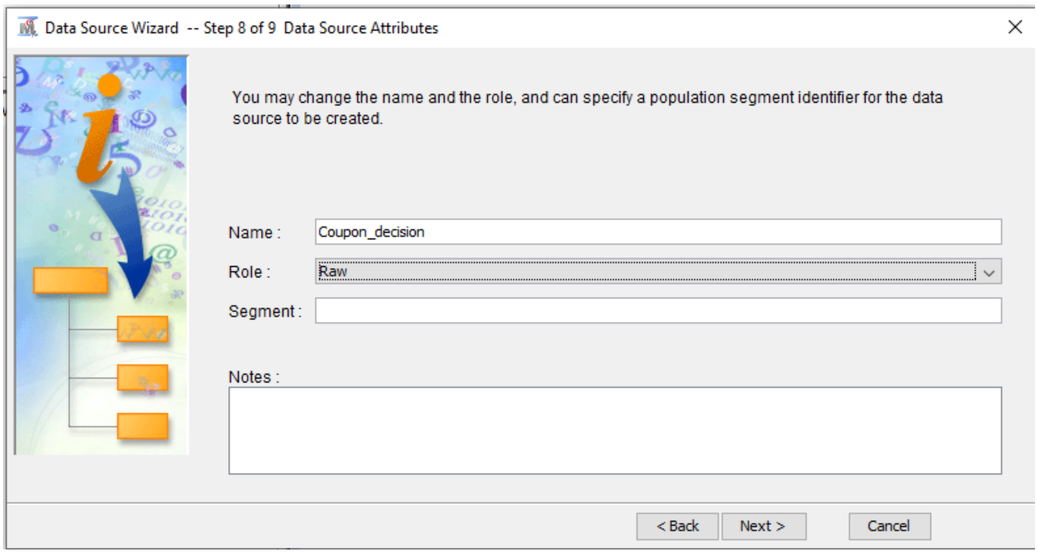
Advanced Version:



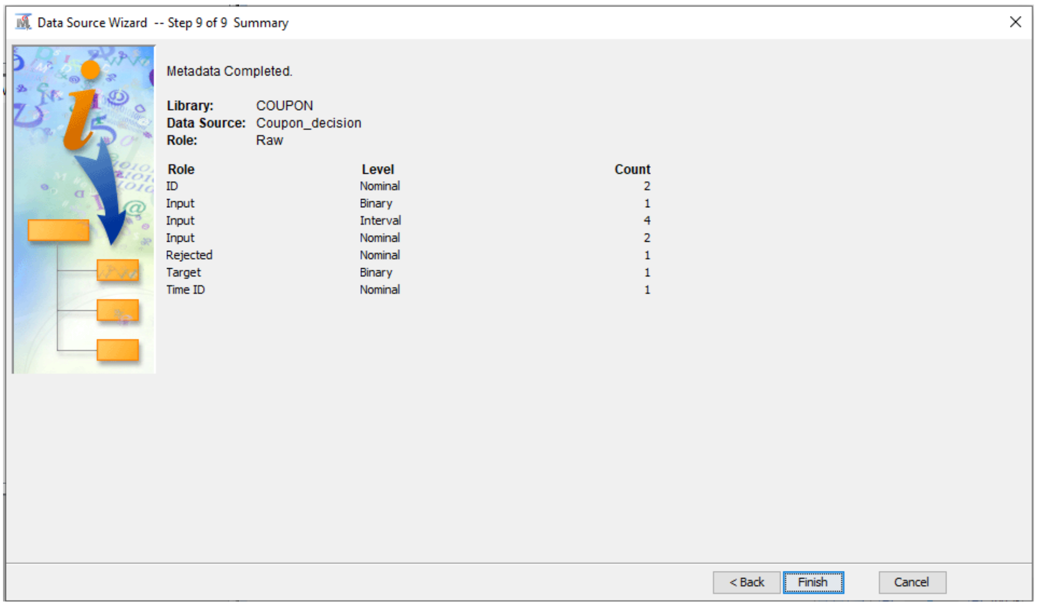
After manual adjustment:



Renamed Coupon_decision for better readability.

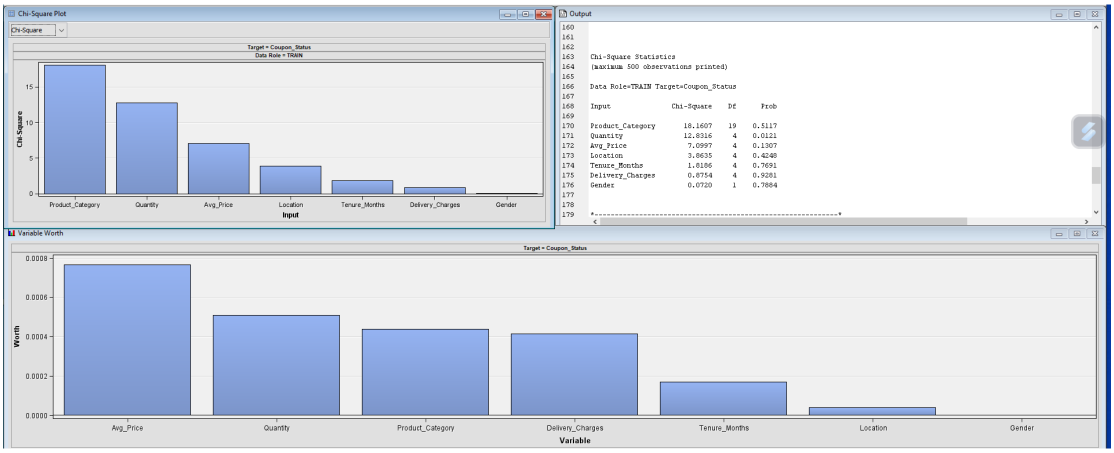


Final presentation

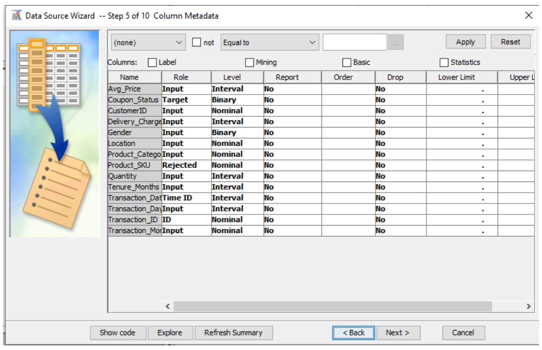


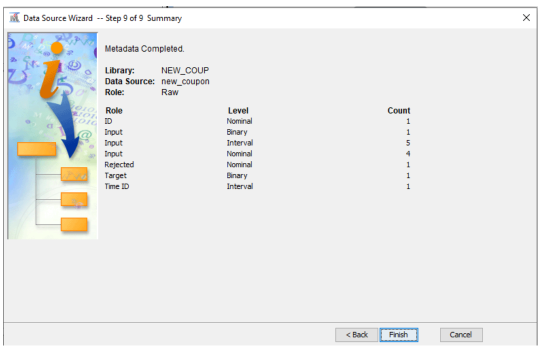
4.使用 StatExplore

Looking at the relationship between INPUT and TARGET, since the cardinality shows no obvious relationship and only QUANTITY presents a certain relationship, I considered the possibility that the month of purchase and the day of the month might also have an effect, so I felt that from creating a new SOURCE for the comparison, splitting the date of the DATA in it into month and day, made it possible to explore more relationships. Because the current simple exploration of the relationships presented is not promising.

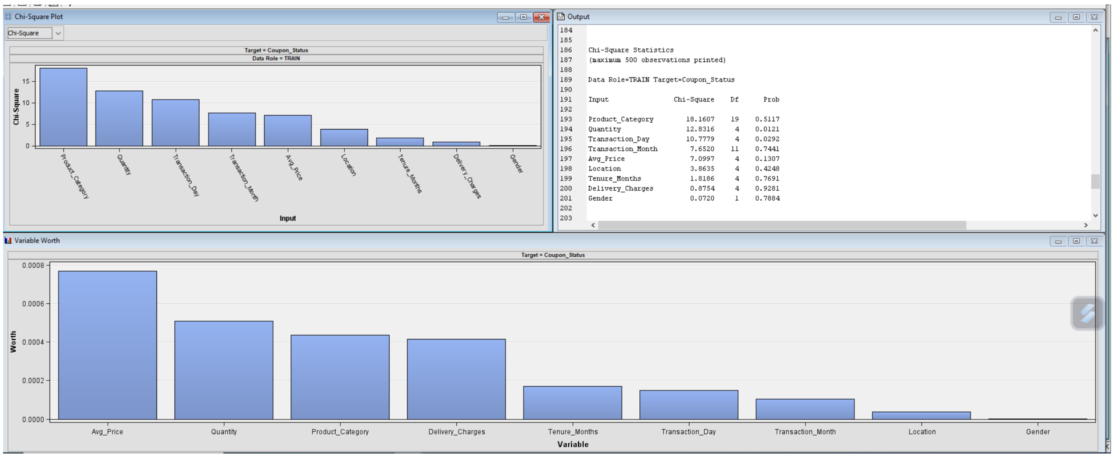


Split in python and form new dataset and new source “new_coupon” in SAS.





As you can see my conjecture is valid and a new significance variable appears which is the trading day, but the model construction is complex so I will look at the results of the two performed decision trees.



5.Running the first DECISION TREE reveals that 1 is not OUTCOME（Explore solutions）

```
Classification Table

Data Role=TRAIN Target Variable=Coupon_Status Target Label=' '

                          Target      Outcome     Frequency      Total
Target     Outcome     Percentage   Percentage      Count      Percentage

  0           0          66.1722       100          14008        66.1722
  1           0          33.8278       100           7161        33.8278


Data Role=VALIDATE Target Variable=Coupon_Status Target Label=' '

                          Target      Outcome     Frequency      Total
Target     Outcome     Percentage   Percentage      Count      Percentage

  0           0          66.1691       100          10505        66.1691
  1           0          33.8309       100           5371        33.8309
```

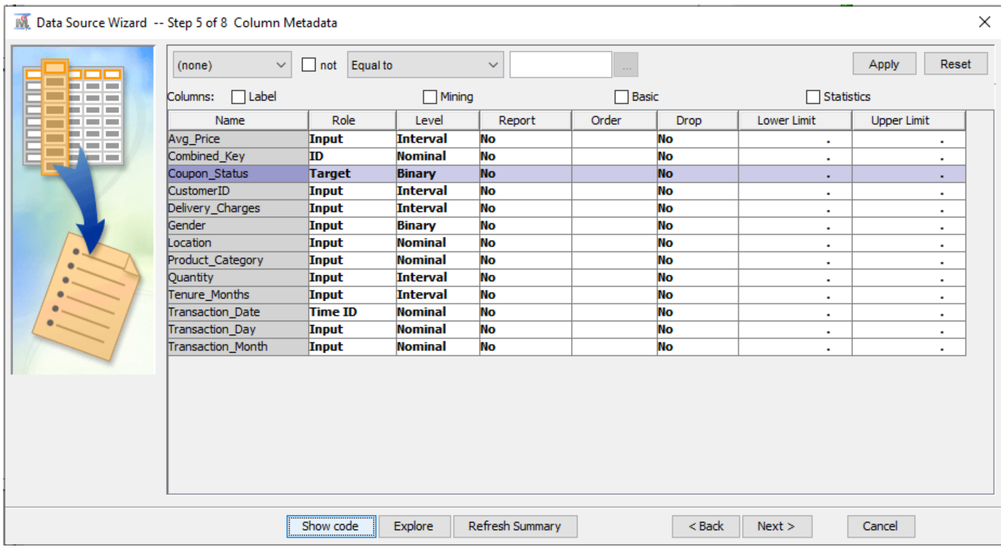
Problem1-solution based on dataset: Adjusting the data structure to continue the exploration, I found a complication in my dataset is that “CustomerID” and “TransactionID” are duplicates, so I need to create a key combination TransactionDd+Produc_SKU to uniquely identify each of the Combined_IDs. for each piece of data, which can then be used for better modeling. Then delete transactionid+Produc_SKU. Done in python.

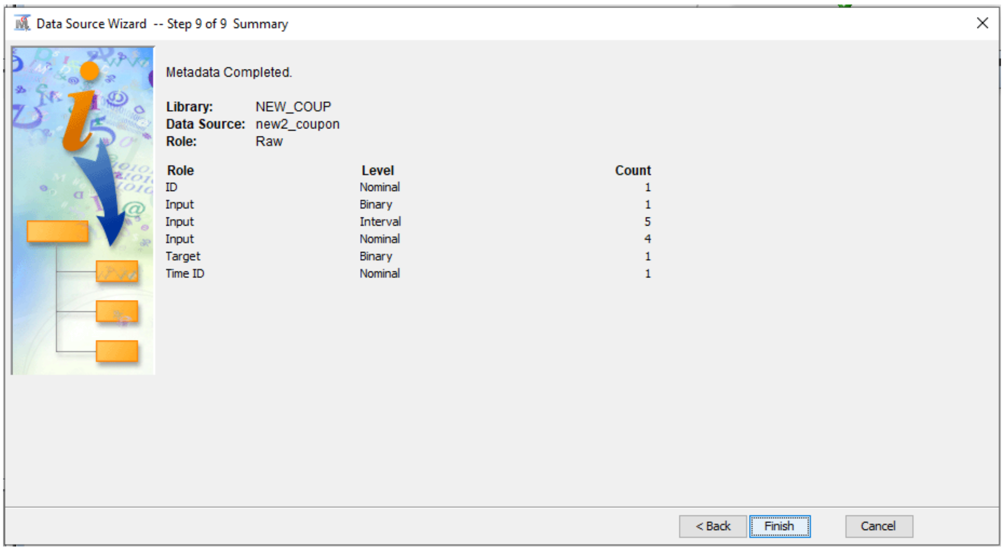
```
data['Combined_Key'] = data['Transaction_ID'].astype(str) + "-" + data['Product_SKU']
```

```
data.to_csv('datasetfinal_new2.csv', index=False)
```

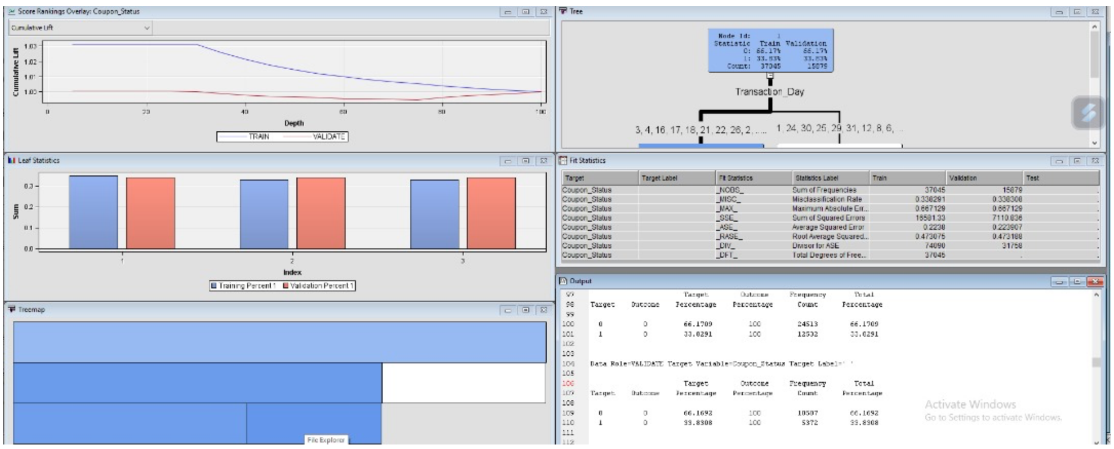
```
data = data.drop(['Transaction_ID', 'Product_SKU'], axis=1)
```

The new data variables are as follows:

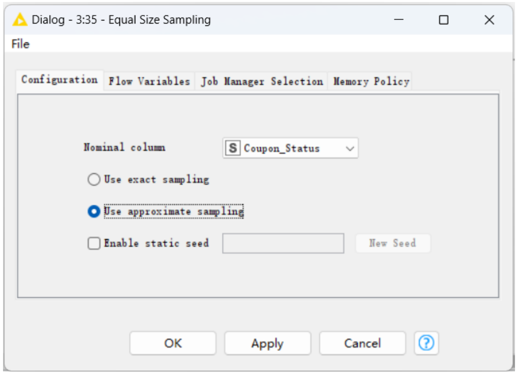


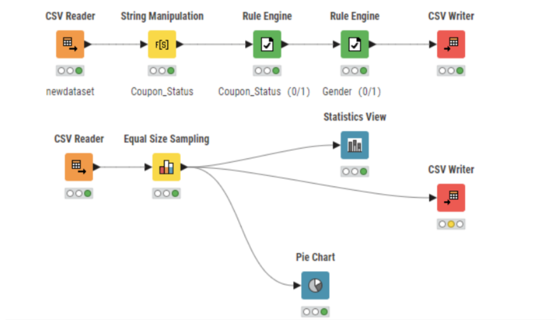


Running the decision tree, the problem is still not solved.

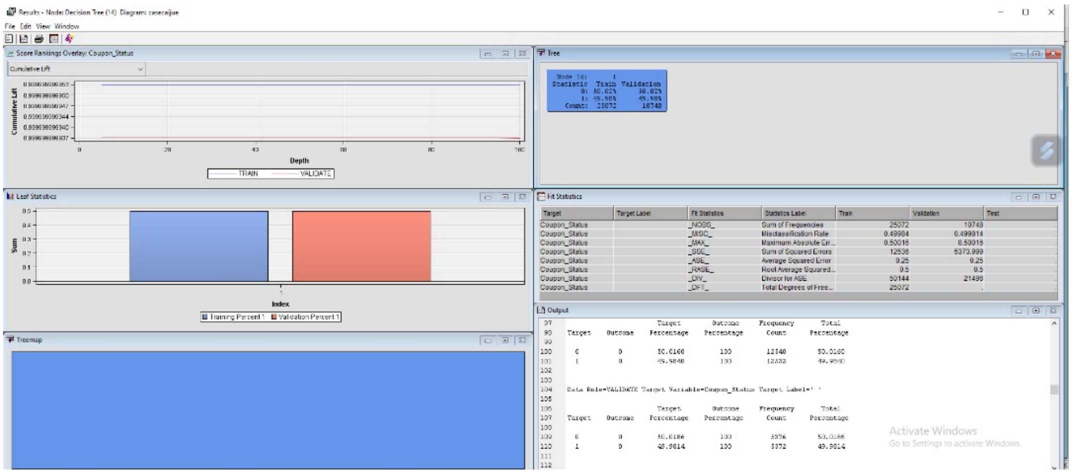


Problem2-solution based on dataset: Checked the 0 and 1 of the coupon status and found that as a TARGET it's 0 and 1 are not balanced, maybe that's why the 0 can't be read and explored, so I solved the data imbalance by exploring it. Since I am not skilled in using sas software, I completed the imbalance processing of the data in KINME.



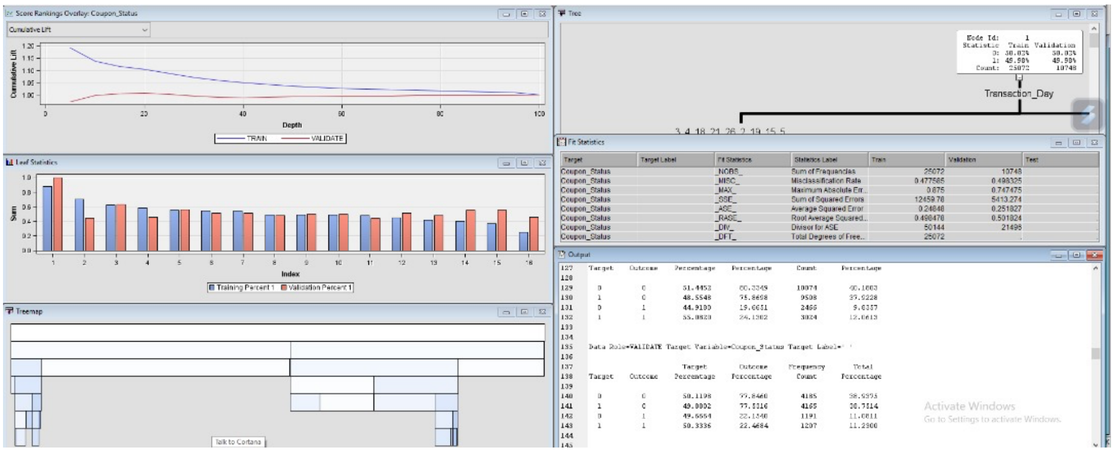


After exporting the new dataset BALANCE and repeating the above steps, the rendering is still poor.



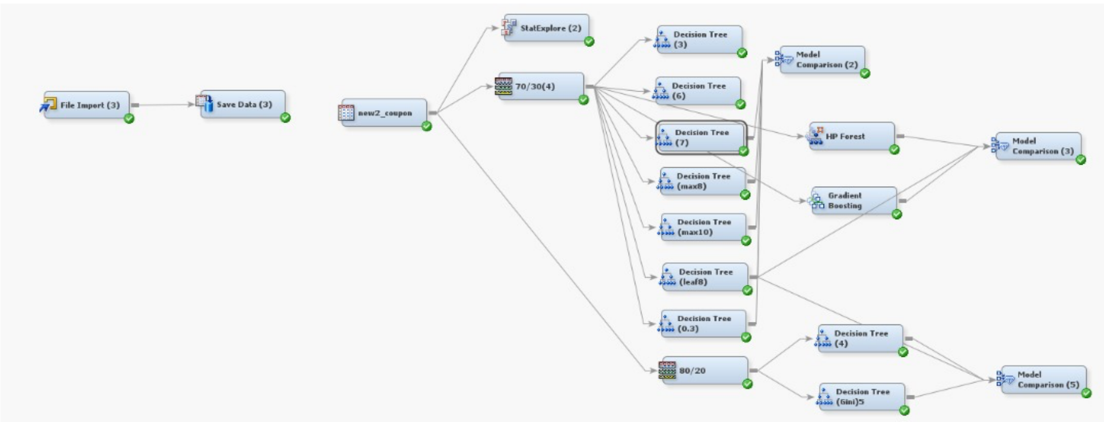
Solution3--solution based on <u>decision tree setting</u>. - **Resolved**

Experimenting with the third possibility,the data was too noisy for another segmentation method, using gini, it was found that a certain amount of decision making occurred although the values were still poor, but it was decided to adjust the decision tree parameters to continue to improve.



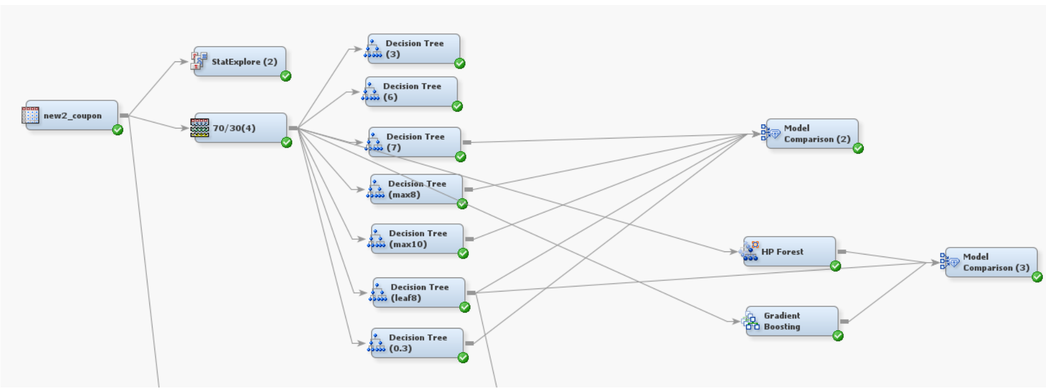
6. Explore decision tree results on both datasets.

6.1 new2_coupon



- **Data set split 70/30**

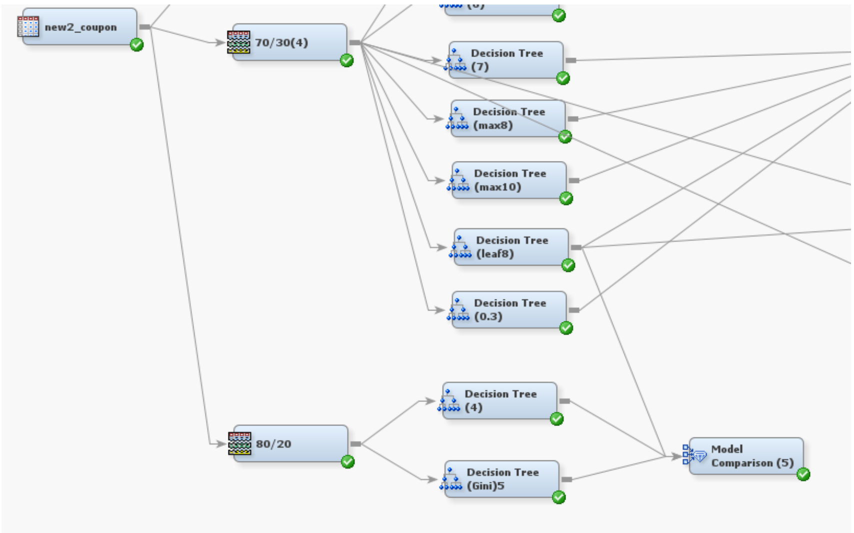
  With a 70/30 score dataset, adjusted for multiple branchleaf values and dength values, Gini leaf8 dength10 performed the best.



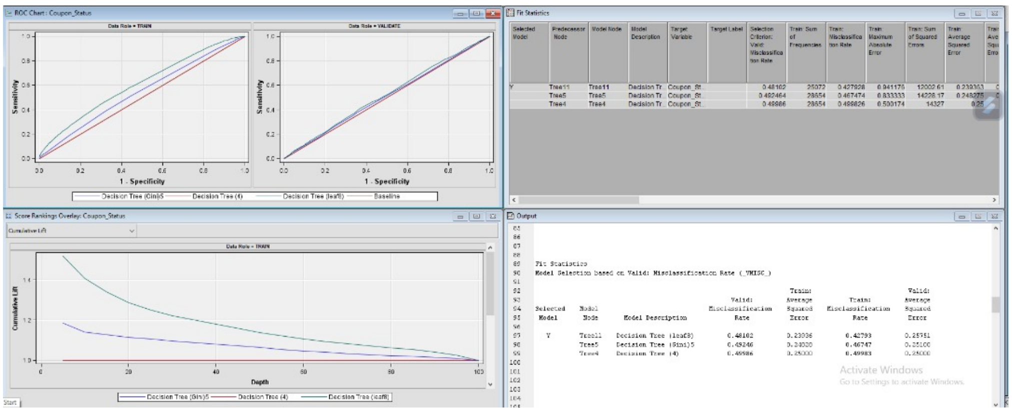
Model Comparison（2）Outcome.

Selected Model	Model Node	Model Description	Valid: Misclassification Rate	Train: Average Squared Error	Train: Misclassification Rate	Valid: Average Squared Error
Y	Tree11	Decision Tree (max10)	0.48102	0.23936	0.42793	0.25751
	Tree9	Decision Tree (max10)	0.48214	0.23923	0.43327	0.25687
	Tree10	Decision Tree (max8)	0.49088	0.24463	0.45333	0.25386
	Tree7	Decision Tree (7)	0.49833	0.24848	0.47758	0.25183

- **Compare the difference between dataset splits 70/30, 80/20**

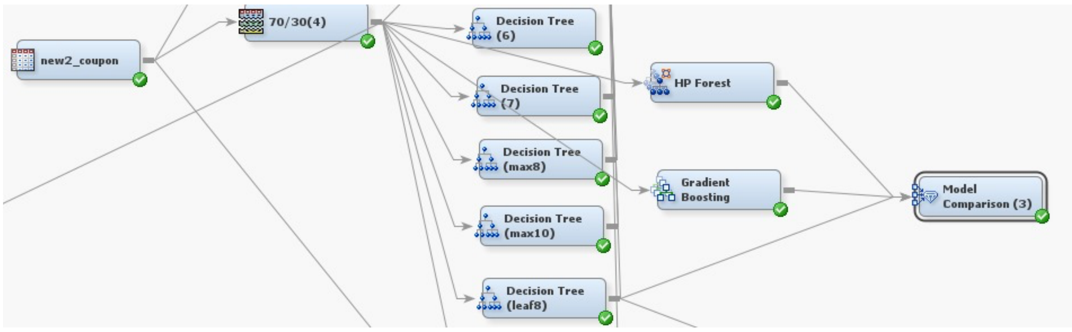


Model Comparison（5）Outcome.



It was found that splitting the size of the other dataset does not drastically affect the accuracy of the model, so it is possible to use 70/30 to continue exploring, since splitting does not make sense.

- **Apply Bagging and Boosting, using the Random Forest algorithm as a Bagging example.**



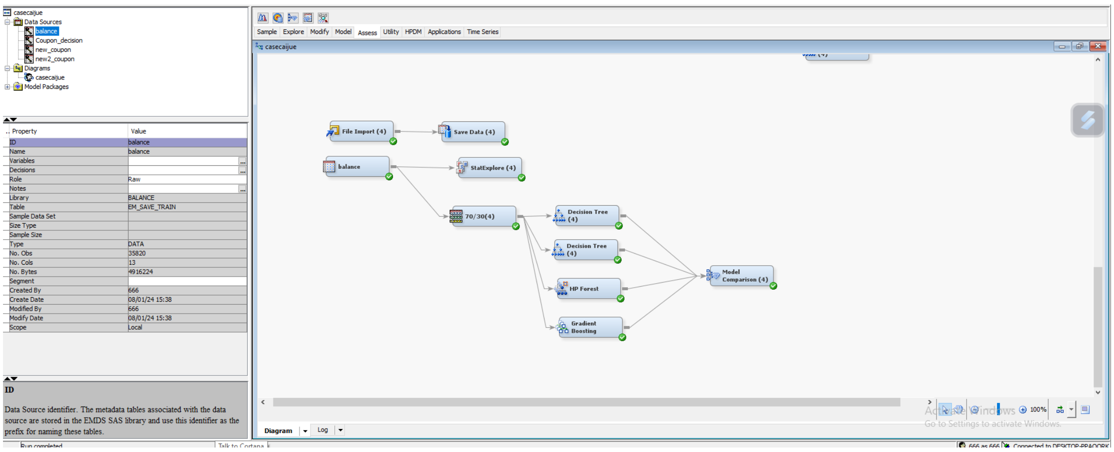
Model Comparison（3）Outcome.

Fit Statistics
Model Selection based on Valid: Misclassification Rate (_VMISC_)

Selected Model	Model Node	Model Description	Valid: Misclassification Rate	Train: Average Squared Error	Train: Misclassification Rate	Valid: Average Squared Error
Y	Tree11	Decision Tree (leaf8)	0.48102	0.23936	0.42793	0.25751
	Boost	Gradient Boosting	0.50502	0.24981	0.48413	0.25012
	HPDMForest	HP Forest	0.50735	0.24970	0.48062	0.25025

Adding bagging and boost tends to make, accuracy improve, but sadly it didn’t in my data project. Suggesting that I need to start this whole project with a review of the meaning of the dataset I’m exploring and whether the categorization really makes sense.

6.2 balance



Fit Statistics
Model Selection based on Valid: Misclassification Rate (_VMISC_)

Selected Model	Model Node	Model Description	Valid: Misclassification Rate	Train: Average Squared Error	Train: Misclassification Rate	Valid: Average Squared Error
Y	Tree13	Decision Tree (4)	0.48809	0.23641	0.41644	0.26199
	Boost2	Gradient Boosting	0.49070	0.24759	0.44241	0.25009
	Tree12	Decision Tree (4)	0.49116	0.23888	0.42266	0.25973
	HPDMForest2	HP Forest	0.51107	0.24967	0.47818	0.25025

The data after balancing with KNIME is also still not good, there are many reasons for this, it could be that the specific technique used for balancing doesn't work and exploring this I think I need to go back to the beginning and reexamine all my work.

General overview of the process: