2021

BigOrganics Business Case

SAS VIYA 3.5 PROJECT

SAFA EL AZRAK

DATA SCIENCETECH INSTITUTE |Paris

Summary

Introduction

This project is made for the Data & Machine Learning Visualisation course followed for the Applied MSc in Data Analytics at DSTI.

The statement is the following :

- Try to find the best model with Model Studio!
- Explain variables you try to add
- Explain the process
- Use and compare result from regressions, neural network, Forest, GB etc.
- Justify choice with screenshot.

Exploring the dataset

Dataset :

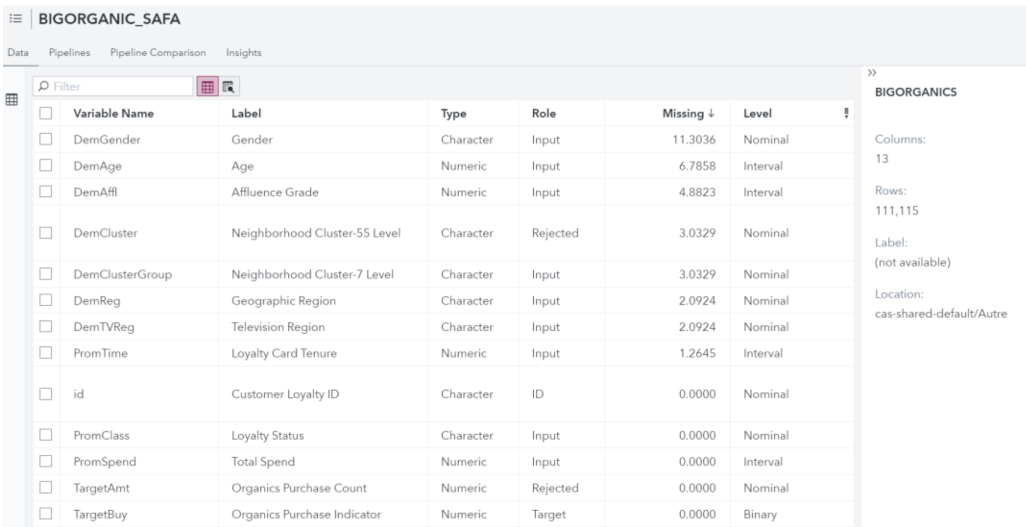
- Name : BIGORGANICS
- Columns : 13
- Rows : 111,115

Variable roles :

- 9 inputs
- 2 rejected
- 1 ID
- 1 target

Target variable:

- Label : “Targetbuy”
- Type : Binary


BIGORGANIC_SAFA

Data Pipelines Pipeline Comparison Insights

Variable Name	Label	Type	Role	Missing ↓	Level
DemGender	Gender	Character	Input	11.3036	Nominal
DemAge	Age	Numeric	Input	6.7858	Interval
DemAffl	Affluence Grade	Numeric	Input	4.8823	Interval
DemCluster	Neighborhood Cluster-55 Level	Character	Rejected	3.0329	Nominal
DemClusterGroup	Neighborhood Cluster-7 Level	Character	Input	3.0329	Nominal
DemReg	Geographic Region	Character	Input	2.0924	Nominal
DemTVReg	Television Region	Character	Input	2.0924	Nominal
PromTime	Loyalty Card Tenure	Numeric	Input	1.2645	Interval
id	Customer Loyalty ID	Character	ID	0.0000	Nominal
PromClass	Loyalty Status	Character	Input	0.0000	Nominal
PromSpend	Total Spend	Numeric	Input	0.0000	Interval
TargetAmt	Organics Purchase Count	Numeric	Rejected	0.0000	Nominal
TargetBuy	Organics Purchase Indicator	Numeric	Target	0.0000	Binary

BIGORGANICS

Columns: 13

Rows: 111,115

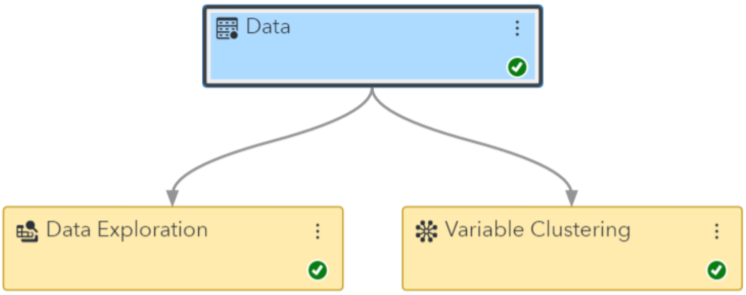
Label: (not available)

Location: cas-shared-default/Autre

Pipelines

1- Exploration

We begin our analysis with the first pipeline named Exploration. It has three nodes as shown below :



Those nodes allow us to explore the data to see if some preprocessing is needed (about outliers or missing values or negative values for example). It also allows us to know if we will need to do some transformations / imputations on the data or some feature selection.

The results of the data exploration node are shown below. Only some of the useful plots and tables will be shown.

Partition used for the exploration :

BIGORGANIC_SAFA > **"Data Exploration" Results**

Summary Output Data

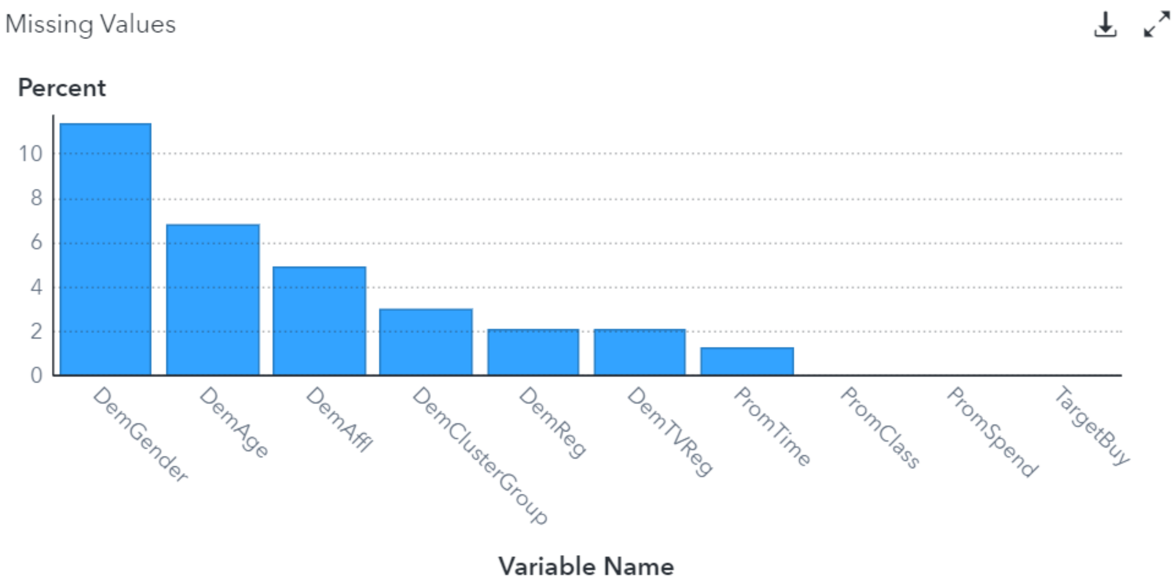
Data Partition Summary

Partition Used	Number of Observations
All data	111,115

Looking for missing values :

Displaying results of the Data Exploration node we notice that they're some important missing values.

The chart below shows variables ranked by the percentage of missing values they have. Here, for example: Demographic Gender has the most missing values …



We can also see that using the Data Tab :

BIGORGANIC_SAFA

Data Pipelines Pipeline Comparison Insights

Filter

Variable Name	Label	Type	Role	Missing ↓	Level
DemGender	Gender	Character	Input	11.3036	Nominal
DemAge	Age	Numeric	Input	6.7858	Interval
DemAffl	Affluence Grade	Numeric	Input	4.8823	Interval
DemCluster	Neighborhood Cluster-55 Level	Character	Rejected	3.0329	Nominal
DemClusterGroup	Neighborhood Cluster-7 Level	Character	Input	3.0329	Nominal
DemReg	Geographic Region	Character	Input	2.0924	Nominal
DemTVReg	Television Region	Character	Input	2.0924	Nominal
PromTime	Loyalty Card Tenure	Numeric	Input	1.2645	Interval
id	Customer Loyalty ID	Character	ID	0.0000	Nominal
PromClass	Loyalty Status	Character	Input	0.0000	Nominal
PromSpend	Total Spend	Numeric	Input	0.0000	Interval
TargetAmt	Organics Purchase Count	Numeric	Rejected	0.0000	Nominal
TargetBuy	Organics Purchase Indicator	Numeric	Target	0.0000	Binary

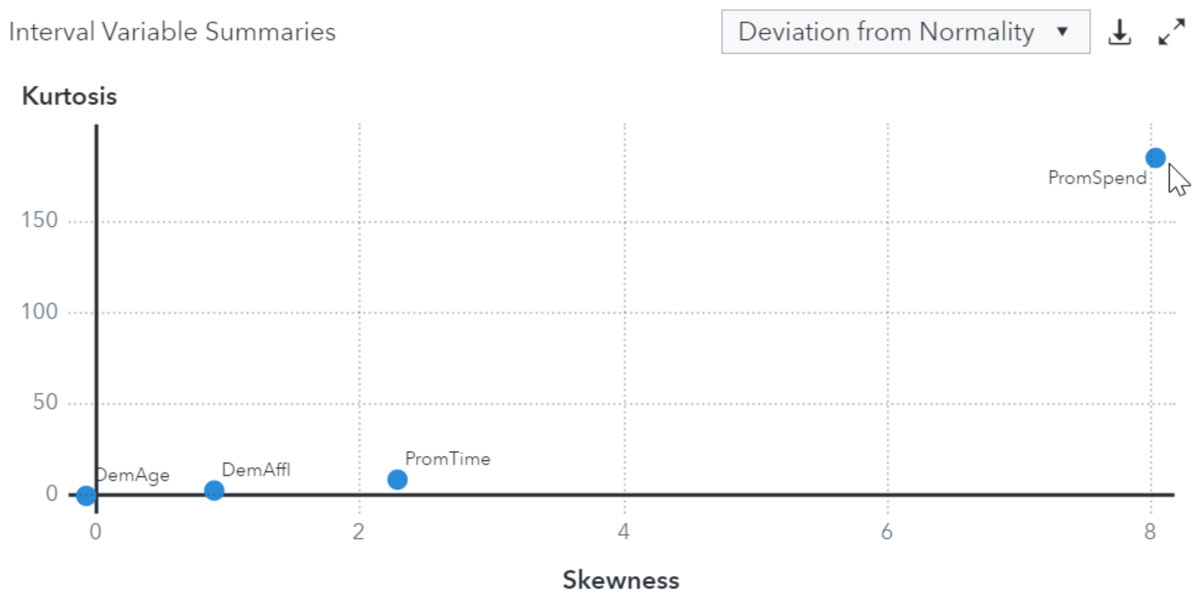
We will need to manage the missing values using the Imputation node in the next pipelines for the models that can't handle Missing values.

Looking for Skewness :

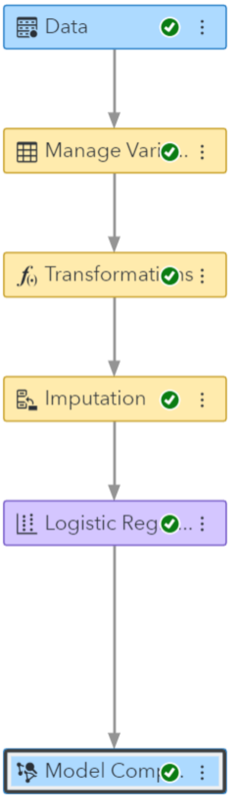
Interval Variable Moments					
Variable Name	**Minimum**	**Maximum**	**Mean**	**Standard De...**	**Skewness ↓**
PromSpend	0.0100	296,313.8500	4,420.5900	7,558.9115	8.0368
PromTime	0	39	6.5647	4.6570	2.2827
DemAffl	0	34	8.7119	3.4211	0.8916
DemAge	18	79	53.7972	13.2058	-0.0798

The variable PromSpend has a very high level of skewness so we will probably have to transform it (Log for example) to use it properly.

The chart below shows this as well :

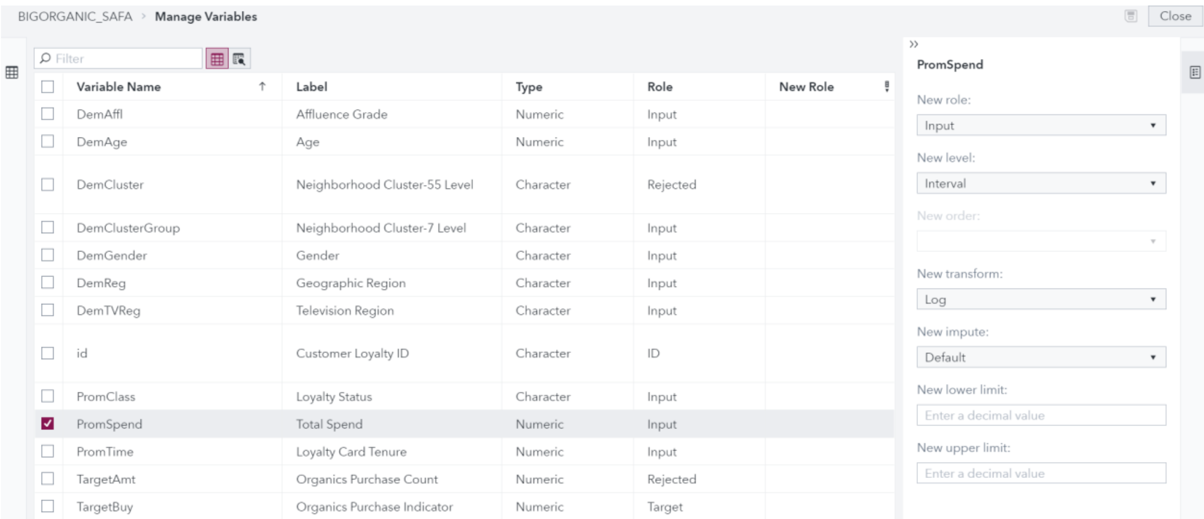


2- Starter Template



Nodes used are :

- **Manage Variables :** to select the variable to transform. Here, it's the PromSpend that has high level of skewness. We will transform it using "Log".



- **Transformation** : apply the Log transformation made at the manage variable node

Transformed Variables Summary

Transforme...	Method	Input Variable	Formula	Variable Level
LOG_PromSpend	LOG	PromSpend	log('PromSpend'n + 1)	INTERVAL

- **Imputation** : to replace missing values

Input Variable Statistics

Input Variable	Variable Level	Number of ...	Percent Mis...	Imputable
DemAffl	INTERVAL	3,220	4.8298	1
DemAge	INTERVAL	4,563	6.8443	1
DemClusterGroup	NOMINAL	2,050	3.0749	1
DemGender	NOMINAL	7,597	11.3951	1
DemReg	NOMINAL	1,427	2.1404	1
DemTVReg	NOMINAL	1,427	2.1404	1
LOG_PromSpend	INTERVAL	0	0	0

Imputed Variables Summary

Imputed Va...	Method	Input Variable	Value	Percent Mis...
IMP_DemAge	MEAN	DemAge	53.7874	6.8443
IMP_DemClusterGroup	COUNT	DemClusterGroup	C	3.0749
IMP_DemGender	COUNT	DemGender	F	11.3951
IMP_DemReg	COUNT	DemReg	South East	2.1404
IMP_DemTVReg	COUNT	DemTVReg	London	2.1404
IMP_PromTime	MEAN	PromTime	6.5748	1.2630

Results of the pipeline :

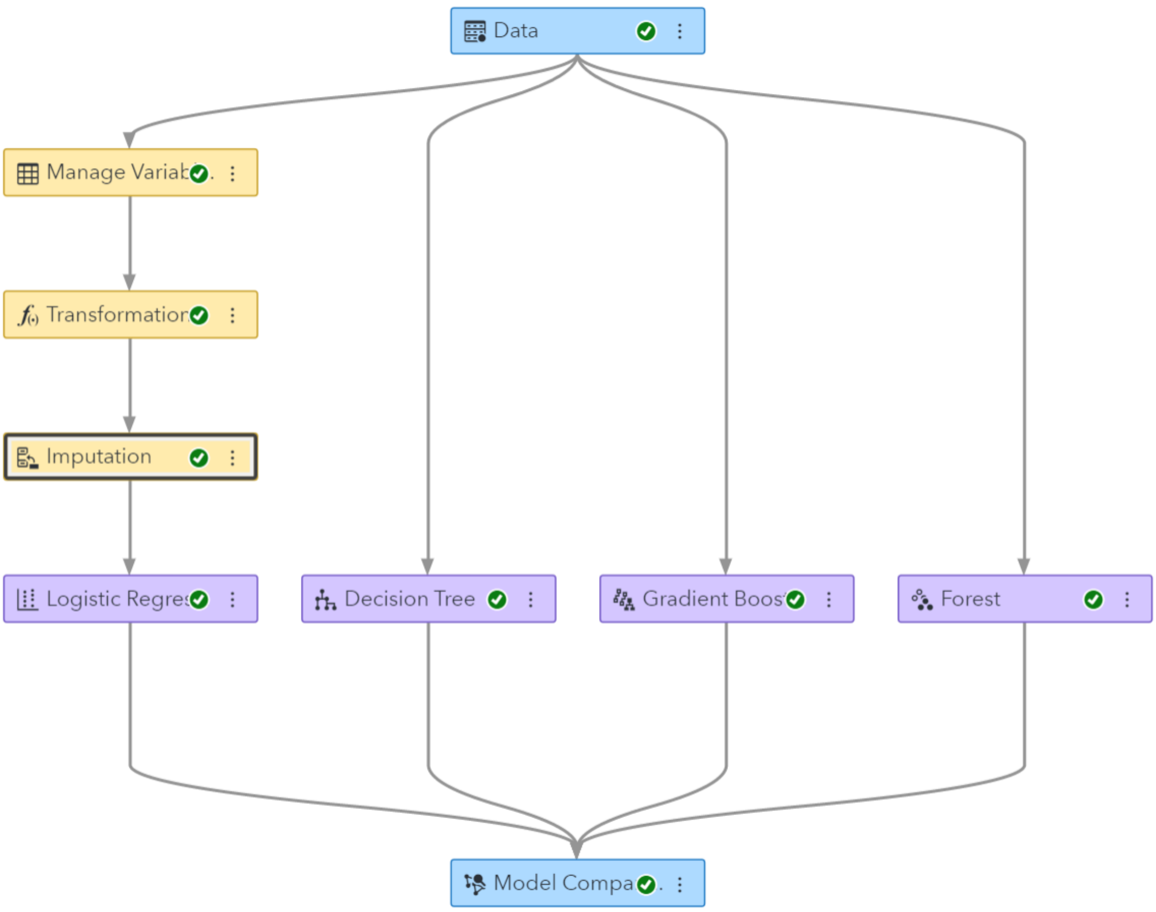
Model Comparison

Champion	Name	Algorithm Name	KS (Youden)	Misclassification Rate
★	Logistic Regression	Logistic Regression	0.4296	0.1976

The logistic regression model has a KS of 0.4296 in this pipeline.

3- Tree based Default models

This pipeline contains 4 models to compare : Tree based models such as Decision tree, Gradient Boosting and Forest that don't need any data preparation and the logistic regression for which we added the pre-processing defined earlier. Here, we run the models with their default settings.



Results of the pipeline :

Model Comparison

Champion	Name	Algorithm Name	KS (Youden)	Misclassification Rate
☒	Forest	Forest	0.6357	0.1377
	Gradient Boosting	Gradient Boosting	0.5149	0.1783
	Decision Tree	Decision Tree	0.4683	0.1804
	Logistic Regression	Logistic Regression	0.4296	0.1976

As we know, the Model Comparison node chooses the Champion model based on the default KS (Youden).

Consequently, in this pipeline, the champion model is Forest.

We can also check other parameters :

- If we are interested in Decision prediction, we can used Misclassification Rate to choose the Champion model. Here also, the smaller the better ➔ Forest also the champion.
- If we want to predict a probability or an estimate, we look at the average squared error (expanding the window). Smaller is Better and Forest is again the champion model
- Finally, if we are interested in ranks, we can look at the area under the Curve (ROC chart) where bigger is better and here also Forest is the best model.

Model Comparison

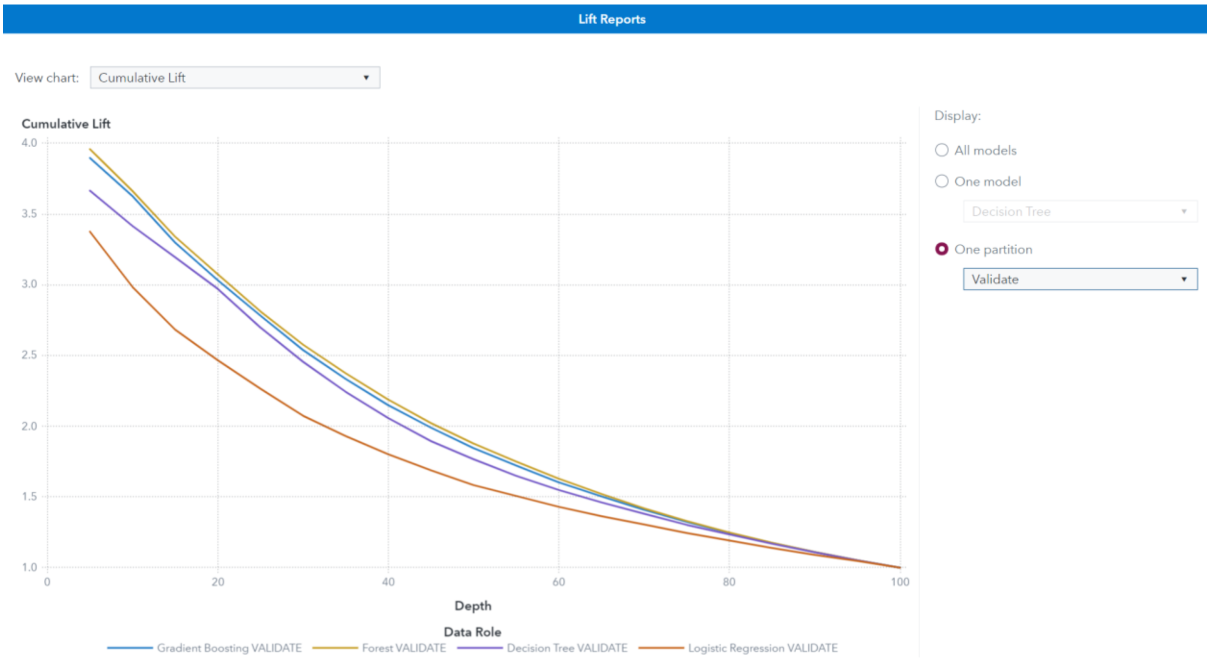
Champi...	Name	Algorith...	KS (You...	Misclas...	Misclas...	Root Av...	Averag...	Sum of ...	Multi-Cl...	Gini Co...	Area Un...	Gain
	Forest	Forest	0.6357	0.1377	0.1377	0.3178	0.1010	33,334	0.3295	0.8168	0.9084	2.7592
	Gradient Boosting	Gradient Boosting	0.5149	0.1783	0.1783	0.3571	0.1275	33,334	0.4013	0.6794	0.8397	2.2094
	Decision Tree	Decision Tree	0.4683	0.1804	0.1804	0.3664	0.1343	33,334	0.4262	0.6036	0.8018	2.1632
	Logistic Regressio n	Logistic Regressio n	0.4296	0.1976	0.1976	0.3777	0.1427	33,334	0.4472	0.5724	0.7862	1.9853

Thus, Forest model is the best according to this pipeline and according to the numerical analysis of the models.

We can also have a graphical analysis to confirm this result :

- **Lift Reports and choosing the "Cumulative Lift"**

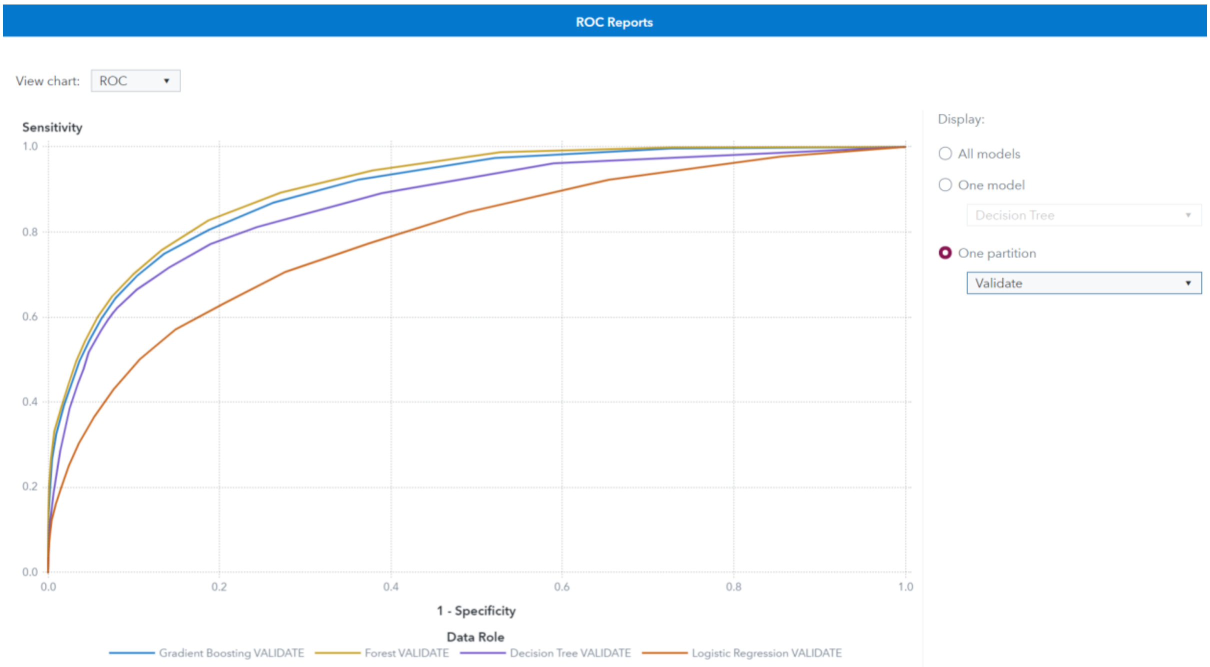
I focused on the Validation partition :



No matter how depth we are in the horizontal axis, the line representing Forest is always on the top, giving the highest cumulative lift so Forest is the best model.

- **ROC chart : choosing ROC**

Also here, focusing on the validation data :

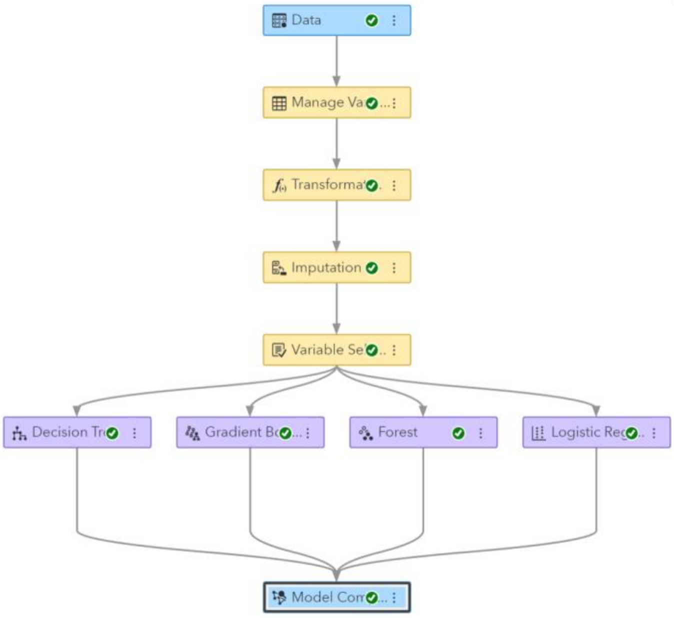


It's similar to the chart before: based on different cutoffs, no matter what cutoff we're at, the yellow line (Forest model) is always the one that's closest to the upper left hand corner of the graph.

Therefore, for this pipeline, numerically & graphically, Forest model is the winner.

4- Tree based Improved models

This pipeline has the same models to compare but I optimized them, improving their efficiency by changing some parameters and by pre-processing the data before applying the tree based models also.



Changes made to the Decision Tree :

- Changes in the structure parameters of the Decision tree :
  - Increase the maximum depth (to grow a larger tree)
  - Increase minimum leaf size (helps prevent overfitting)
  - Increase number of interval bins
- Changes in Recursive Partitioning parameters :
  - Change the grow criterion
    - Class target criterion to Gini
- Changes in the Pruning parameters :
  - Subtree method to Reduced Error

Changes made to improve the Gradient boosting :

- Increase the number of trees
- Increase the maximum depth in the tree splitting options
- Increase the minimum leaf size
- Increase the number of interval bins

Changes made to the Forest model :

- Decrease the number of trees
- Change the class target criterion to Entropy
- Decrease the maximum depth
- Increase the minimum leaf count
- Increase the number of interval bins

Results of the pipeline:

Numerically :

Model Comparison

Champion	Name	Algorithm Name	KS (Youden)	Misclassification Rate
	Forest	Forest	0.5875	0.1602
	Gradient Boosting	Gradient Boosting	0.5625	0.1645
	Decision Tree	Decision Tree	0.5379	0.1603
	Logistic Regression	Logistic Regression	0.4293	0.1984

The model with the best KS is the Forest model so it's the champion model again in this pipeline.

Model Comparison

Champi...	Name	Algorith...	KS (You...	Misclas...	Misclas...	Root Av...	Averag...	Sum of ...	Multi-Cl...	Gini Co...	Area Un...	Gain
	Forest	Forest	0.5875	0.1602	0.1602	0.3363	0.1131	33,334	0.3598	0.7553	0.8777	2.5533
	Gradient Boosting	Gradient Boosting	0.5625	0.1645	0.1645	0.3422	0.1171	33,334	0.3721	0.7340	0.8670	2.4322
	Decision Tree	Decision Tree	0.5379	0.1603	0.1603	0.3467	0.1202	33,334	0.3948	0.6925	0.8463	2.4232
	Logistic Regressio n	Logistic Regressio n	0.4293	0.1984	0.1984	0.3777	0.1427	33,334	0.4473	0.5722	0.7861	2.0012

We can notice that different models have improved :

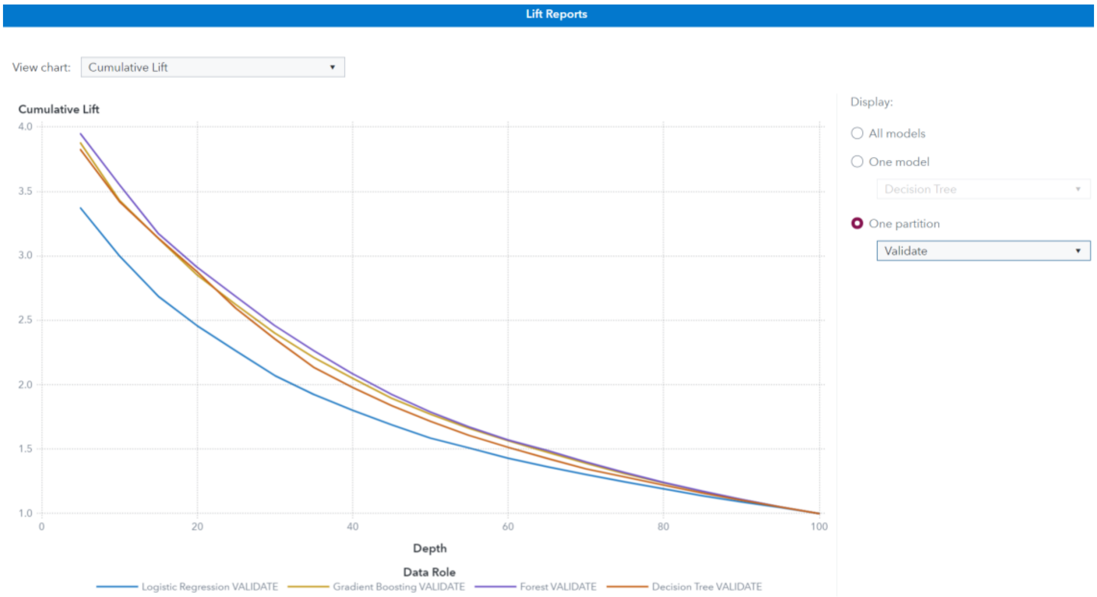
- ***Decision tree model :***
  - KS went from 0.4683 to 0.5379
  - Average squared error went from 0.1343 to 0.1202.
- ***Gradient Boosting model***:
  - KS went from 0.5149 to 0.5625
  - Average squared error went from 0.1275 to 0.1171

On the other hand, the ***Forest model***, even if it's still the champion model didn't improve :

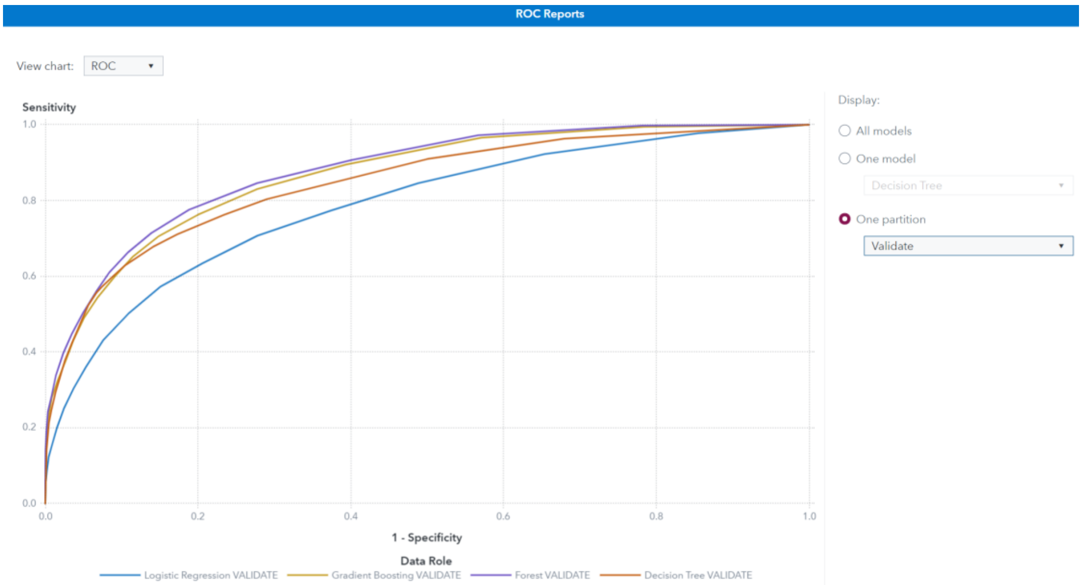
  - KS went from 0.6357 to 0.5875
  - Average squared error went from 0.1010 to 0.1131

Graphically :

Cumulative Lift report : On the validation partition, the line representing the Forest model is on the top no matter how deep we go.



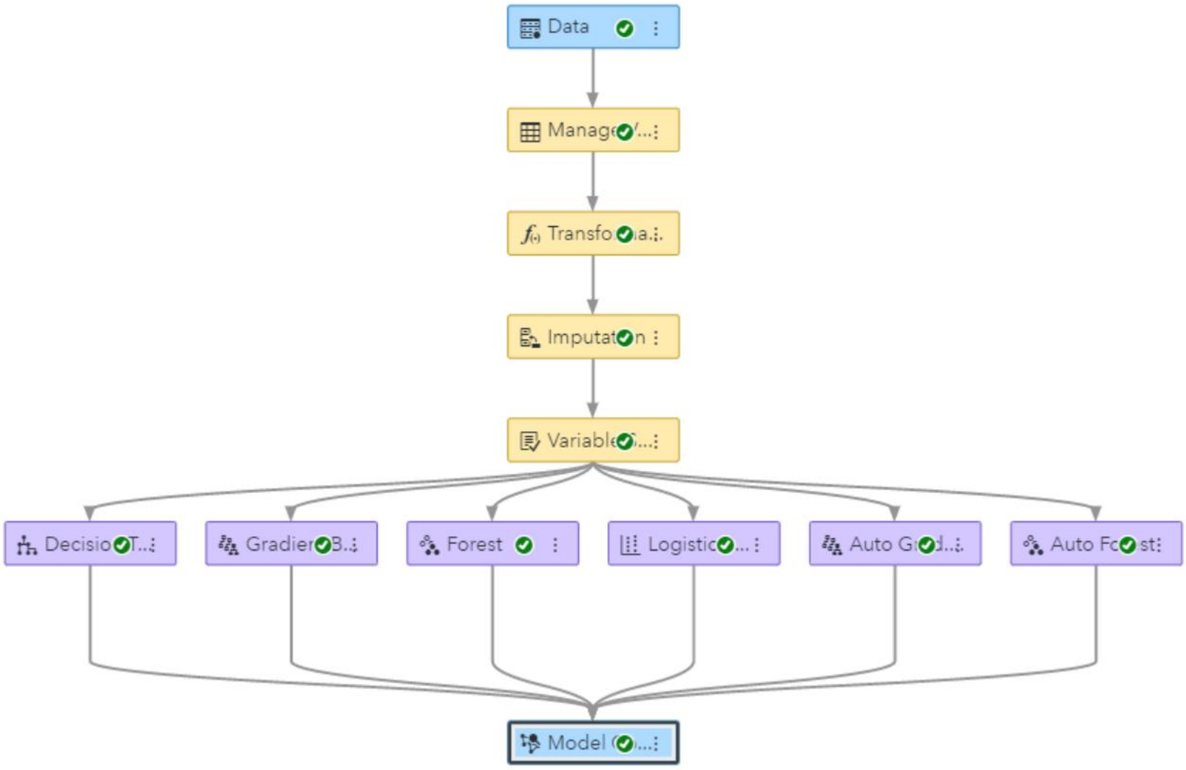
ROC chart : Similarly, Forest model's line is always on the top left hand of the graph.



We can clearly see that Forest model is the best graphically also.

Performing Autotuning on this pipeline

I added several nodes to this pipeline to perform autotuning and see if the champion model would change.



Results after autotuning :

Model Comparison

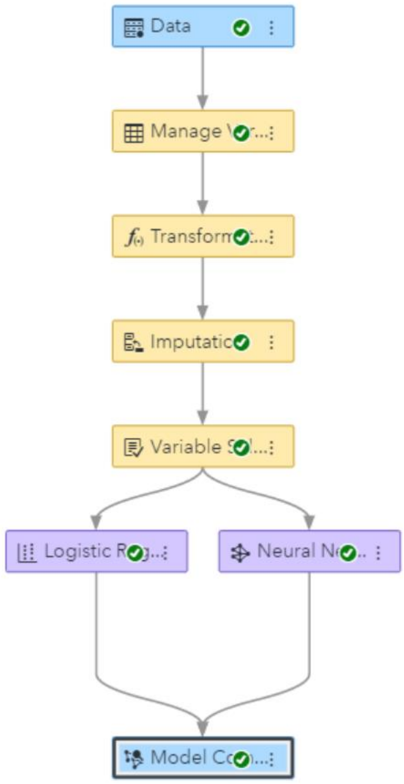
Champi...	Name	Algorith...	KS (You...	Misclas...	Misclas...	Root Av...	Averag...	Sum of ...	Multi-Cl...	Gini Co...	Area Un...	Gain
★	Auto Grad. B	Gradient Boosting	0.7285	0.1109	0.1109	0.2794	0.0781	33,334	0.2523	0.8820	0.9410	2.9373
	Auto Forest	Forest	0.7264	0.1126	0.1126	0.2864	0.0820	33,334	0.2733	0.8845	0.9423	2.9500
	Forest	Forest	0.5875	0.1602	0.1602	0.3363	0.1131	33,334	0.3598	0.7553	0.8777	2.5533
	Gradient Boosting	Gradient Boosting	0.5625	0.1645	0.1645	0.3422	0.1171	33,334	0.3721	0.7340	0.8670	2.4322
	Decision Tree	Decision Tree	0.5379	0.1603	0.1603	0.3467	0.1202	33,334	0.3948	0.6925	0.8463	2.4232
	Logistic Regression	Logistic Regression	0.4293	0.1984	0.1984	0.3777	0.1427	33,334	0.4473	0.5722	0.7861	2.0012

After performing autotuning on Gradient Boosting model and Forest we can see that the champion model has changed. The new one is the ***Autotuned Gradient Boosting model***.

5- Neural Networks

In this pipeline, I added two models : Neural Network and Logistic Regression.

I left the Neural Network to its default settings.



Results of the pipeline:

Model Comparison

Champi...	Name	Algorith...	KS (You...	Misclas...	Misclas...	Root Av...	Averag...	Sum of ...	Multi-Cl...	Gini Co...	Area Un...	Gain
▣	Logistic Regression	Logistic Regression	0.4303	0.1984	0.1984	0.3778	0.1427	33,334	0.4474	0.5741	0.7871	2.0023
	Neural Network	Neural Network	0	0.2477	0.2477	0.4315	0.1862	33,334	0.5594	0	0.5000	1.4440

In this pipeline, according to the Misclassification rate or the Average squared error, the champion model is the Logistic Regression.

Improving the model:

Changes made to the Neural Network model :

- Change the input standardization to “Z score”
- Clear the checkbox for “Use same number of neurons on hidden layers”
- Decrease Custom Hidden Layer 1

Change the Learning and Optimization Parameters :

- Increase the L1 weight decay
- Decrease the L2 weight decay

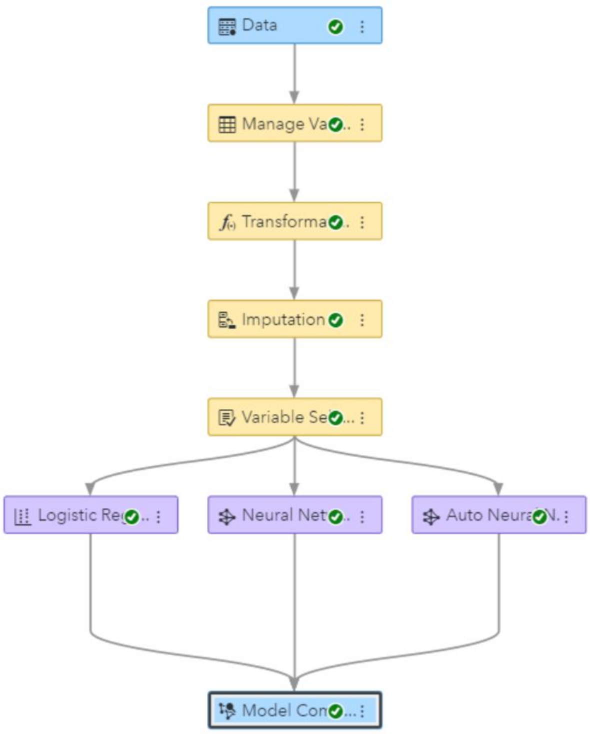
Results of the improved pipeline:

Model Comparison													
Champi…	Name	Algorith…	Misclas…	Misclas…	Root Av…	Averag…	Sum of …	Multi-Cl…	KS (You…	Gini Co…	Area Un…	Gain
	Neural Network	Neural Network	0.1981	0.1981	0.3785	0.1433	33,334	0.4494	0.4331	0.5715	0.7857	1.9793
	Logistic Regression	Logistic Regression	0.1984	0.1984	0.3778	0.1427	33,334	0.4474	0.4303	0.5741	0.7871	2.0023

The champion has changed ! The new one is the improved Neural Network model (according to the Misclassification rate : went from 0.2477 to 0.1981 and its Average squared error really improved also!)

Performing Autotuning on this pipeline

I added a new node to this pipeline to perform autotuning and see if the champion model would change.



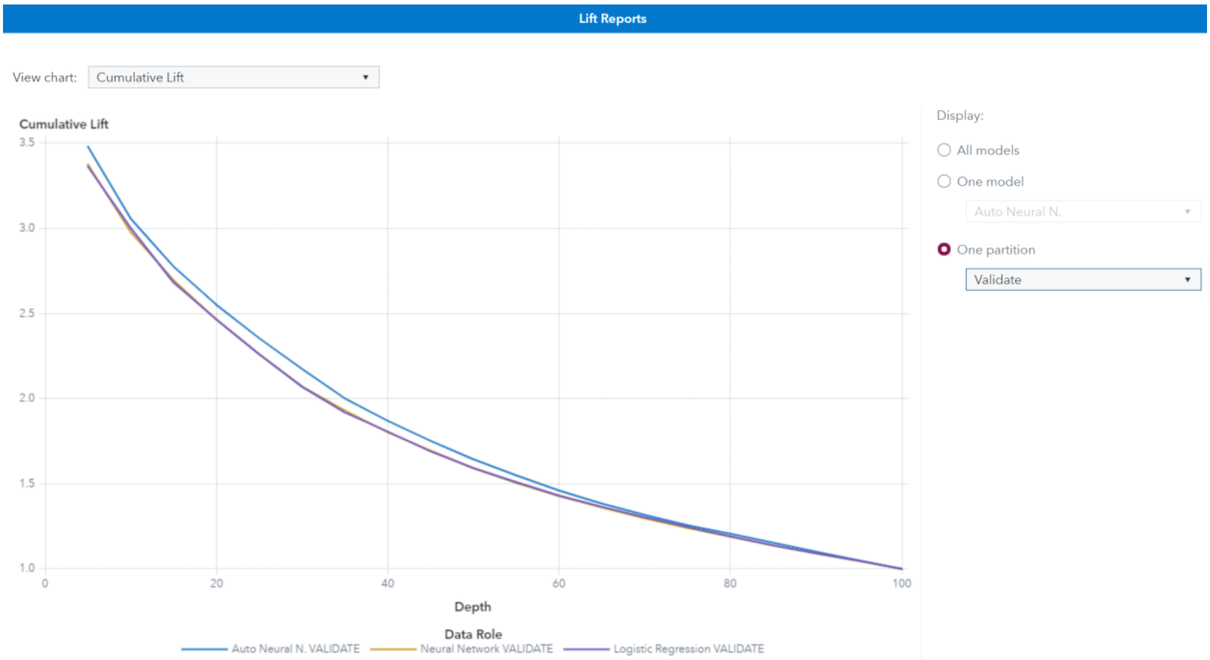
Results after autotuning :

Model Comparison

Champi...	Name	Algorith...	KS (You...	Misclas...	Misclas...	Root Av...	Averag...	Sum of ...	Multi-Cl...	Gini Co...	Area Un...	Gain
★	Auto Neural N.	Neural Network	0.4684	0.1916	0.1916	0.3700	0.1369	33,334	0.4296	0.6133	0.8066	2.0579
	Neural Network	Neural Network	0.4331	0.1981	0.1981	0.3785	0.1433	33,334	0.4494	0.5715	0.7857	1.9793
	Logistic Regression	Logistic Regression	0.4303	0.1984	0.1984	0.3778	0.1427	33,334	0.4474	0.5741	0.7871	2.0023

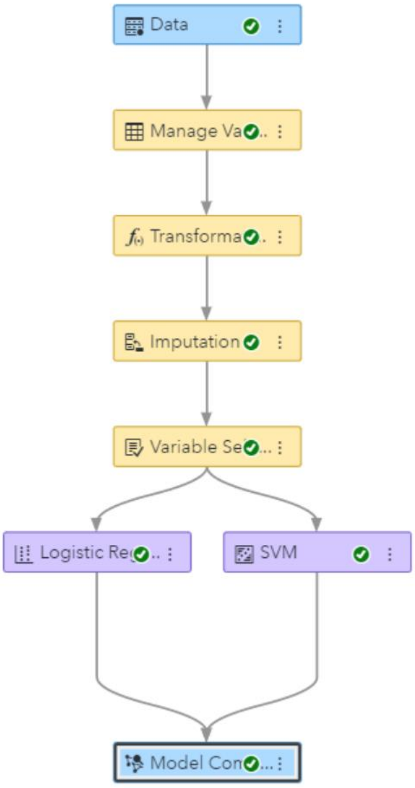
The champion model in this pipeline after performing autotuning is the Auto neural network.

We can check it graphically by looking at the Cumulative lift Report on the validation partition :



The line representing the Autotuned Neural Network model is, at any depth, on the top.

6- Support Vector Machine



Results of the pipeline:

Model Comparison

Champi...	Name	Algorith...	KS (You...	Misclas...	Misclas...	Root Av...	Averag...	Sum of ...	Multi-Cl...	Gini Co...	Area Un...	Gain
	Logistic Regression	Logistic Regression	0.4303	0.1984	0.1984	0.3778	0.1427	33,334	0.4474	0.5741	0.7871	2.0023
	SVM	SVM	0.4229	0.2039	0.2039	0.4057	0.1646	33,334	0.5098	0.5638	0.7819	1.9805

In this pipeline, the champion model is the Logistic Regression according to the KS or the Misclassification rate.

Improving the model:

Changes made to the SVM model :

- Decrease the Penalty / to change the methods of solution parameters
- Change the Kernel setting to Polynomial
- Increase the tolerance to avoid overfitting
- Decrease the maximum iterations

Results of the improved pipeline:

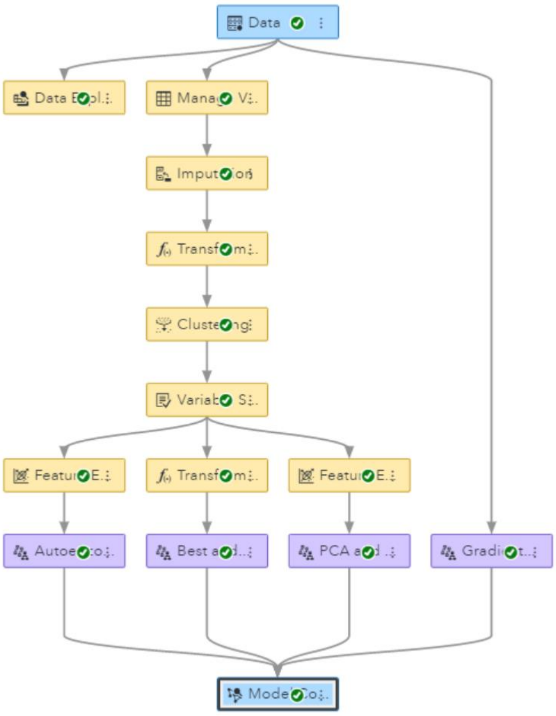
Model Comparison

Champi...	Name	Algorith...	KS (You...	Misclas...	Misclas...	Root Av...	Averag...	Sum of ...	Multi-Cl...	Gini Co...	Area Un...	Gain
☑	Logistic Regression	Logistic Regression	0.4303	0.1984	0.1984	0.3778	0.1427	33,334	0.4474	0.5741	0.7871	2.0023
	SVM	SVM	0.4269	0.1981	0.1981	0.4215	0.1777	33,334	0.5400	0.5623	0.7812	1.9696

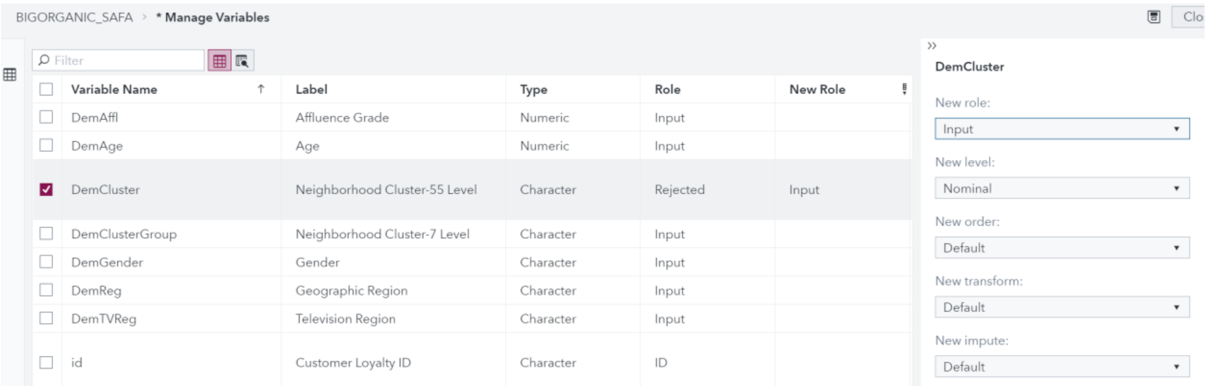
After modifying the parameters of the SVM model, the KS and the Misclassification rate improved but not the Average squared error which is higher than before. But still, even with this general improvement, the champion model is the Logistic Regression.

7- Feature Engineering

This pipeline is based on the feature engineering template available in Model Studio.



In the manage variable node I assigned a new role to "DemCluster" from Rejected to Input :


BIGORGANIC_SAFA > * Manage Variables

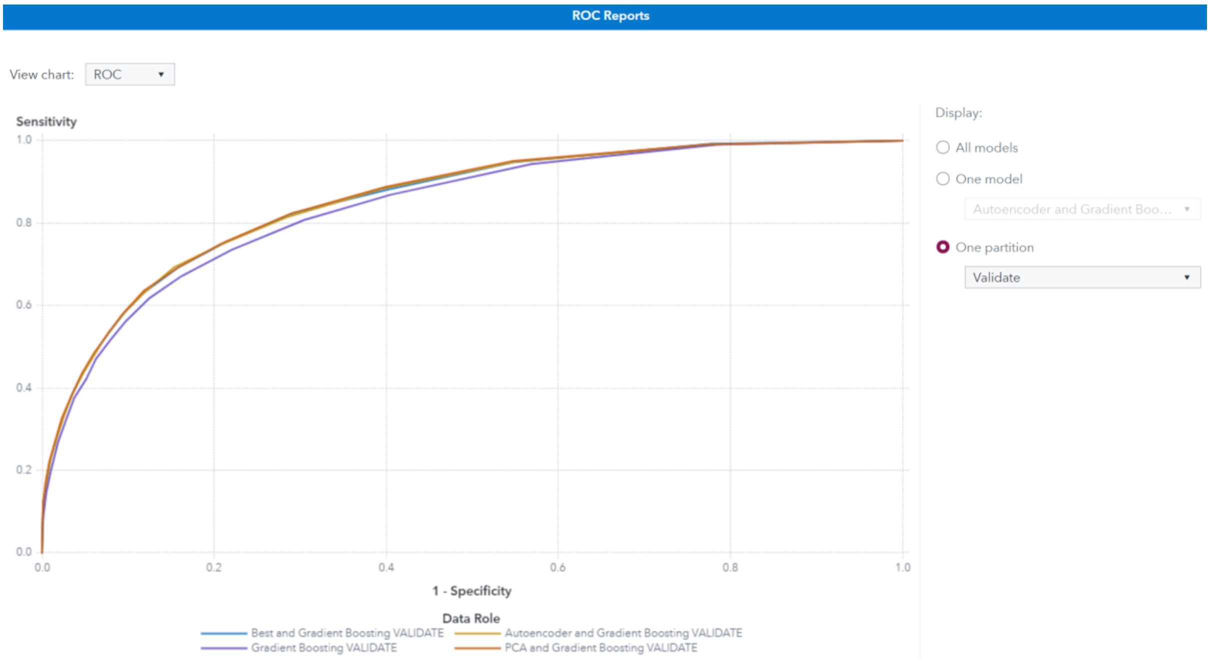
	Variable Name	Label	Type	Role	New Role
☐	DemAffl	Affluence Grade	Numeric	Input	
☐	DemAge	Age	Numeric	Input	
☑	DemCluster	Neighborhood Cluster-55 Level	Character	Rejected	Input
☐	DemClusterGroup	Neighborhood Cluster-7 Level	Character	Input	
☐	DemGender	Gender	Character	Input	
☐	DemReg	Geographic Region	Character	Input	
☐	DemTVReg	Television Region	Character	Input	
☐	id	Customer Loyalty ID	Character	ID	

DemCluster
New role: Input
New level: Nominal
New order: Default
New transform: Default
New impute: Default

Results of the pipeline:

Model Comparison

Champi...	Name	Algorith...	Misclas...	Misclas...	Root Av...	Averag...	Sum of ...	Multi-Cl...	KS (You...	Gini Co...	Area Un...	Gain
	PCA and Gradient Boosting	Gradient Boosting	0.1730	0.1730	0.3499	0.1224	33,334	0.3862	0.5420	0.7100	0.8550	2.3087
	Best and Gradient Boosting	Gradient Boosting	0.1736	0.1736	0.3503	0.1227	33,334	0.3876	0.5405	0.7063	0.8531	2.2990
	Autoencoder and Gradient Boosting	Gradient Boosting	0.1744	0.1744	0.3505	0.1229	33,334	0.3879	0.5400	0.7059	0.8530	2.2930
	Gradient Boosting	Gradient Boosting	0.1783	0.1783	0.3571	0.1275	33,334	0.4013	0.5149	0.6794	0.8397	2.2094



In this pipeline, the champion model is the PCA and Gradient Boosting model.

Pipeline Comparison

Now that we have the champion model for each pipeline, we can run a comparison of all these champions using the Pipeline Comparison Tab.

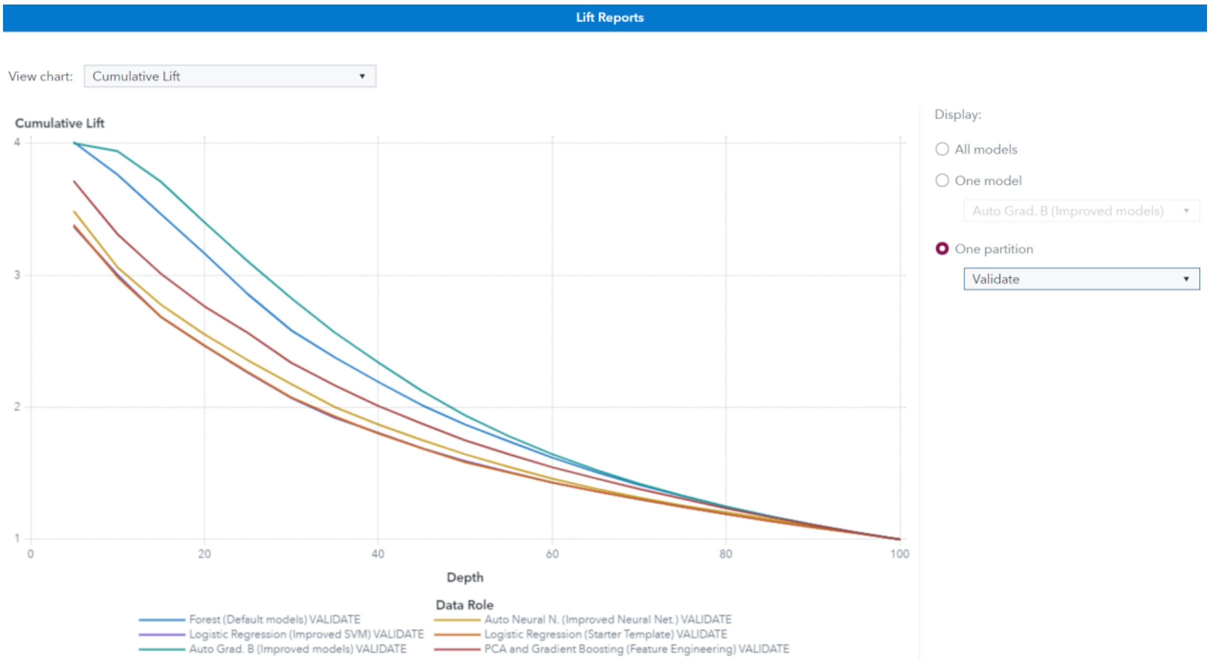
Here are the results :

Pipeline Comparison

Champion	Name	Algorithm Name	Pipeline Name	KS (Youden)	Sum of Frequencie
★	Auto Grad. B	Gradient Boosting	Improved models	0.728	33,334
	Forest	Forest	Default models	0.636	33,334
	PCA and Gradient Boosting	Gradient Boosting	Feature Engineering	0.542	33,334
	Auto Neural N.	Neural Network	Improved Neural Net.	0.468	33,334
	Logistic Regression	Logistic Regression	Improved SVM	0.430	33,334
	Logistic Regression	Logistic Regression	Starter Template	0.430	33,334

The champion model, all pipelines combined, is the ***Gradient Boosting performed with autotuning***. It has a KS of 0.728.

We can confirm that by looking at the Cumulative Lift report, on the Validation partition :



CONCLUSION

Our target variable is of type binary so we could process several models to analyse it and model it.

When choosing a model for a specific dataset, it is important to consider some points like the size and the nature of the data. Some models are more efficient with a specific size of the dataset, but overall they apply from small to large datasets (even if tree based models and neural network are better with mid sized to large datasets).

We need to know what we are trying to achieve with the model, how accurate it has to be, how much time do we have to train it and most of all how interpretable it needs to be.

In this analysis, after running several pipelines with their own models and modifying parameters to improve them, the best model of the whole project is the Gradient Boosting performed with autotuning. It is important to notice that the Gradient boosting model has a Moderate level of interpretability and is instable with small training sets. Therefore, even if the pipeline comparison chose this champion model for us, it is up to business needs to define which model to go on with.