BigOrganics Report

SAS Tools: Predictive Analytics
By – Ayush Tankha (B00802762)

INDEX

Introduction – Overview of the Project

For this project we work on a dataset of the company BigOrganics. After importing the data on the SAS platform, we can conclude that our data has 13 columns respectively that include –

- DemAffl
- PromTime
- PromSpend
- DemAge
- DemClusterGroup
- DemGender
- DemReg
- DemTVReg
- PromClass
- TargetAmt
- DemCluster
- DemBuy

☐	Variable Name	Role ↑	Minimum ↑	Label
☐	id	ID		Customer Loyalty ID
☐	DemAffl	Input	0.0000	Affluence Grade
☐	PromTime	Input	0.0000	Loyalty Card Tenure
☐	PromSpend	Input	0.0100	Total Spend
☐	DemAge	Input	18.0000	Age
☐	DemClusterGroup	Input		Neighborhood Cluster-7 Level
☐	DemGender	Input		Gender
☐	DemReg	Input		Geographic Region
☐	DemTVReg	Input		Television Region
☐	PromClass	Input		Loyalty Status
☐	TargetAmt	Rejected	0.0000	Organics Purchase Count
☐	DemCluster	Rejected		Neighborhood Cluster-55 Level
☐	TargetBuy	Target	0.0000	Organics Purchase Indicator

Introduction – Objectives of the Project

The project tries to best cover the following important objectives -

- To explore the potential of data mining and machine learning techniques for the BigOrganics Business Case.

- To identify the best model for the given data using Model Studio in SAS Viya.

- To explain the process of data mining and machine learning using SAS Viya.

- To compare and evaluate the results obtained from different models such as regressions, neural network, Forest, GB, etc.

- To provide justification for the choice of the best model using appropriate metrics and visualizations.

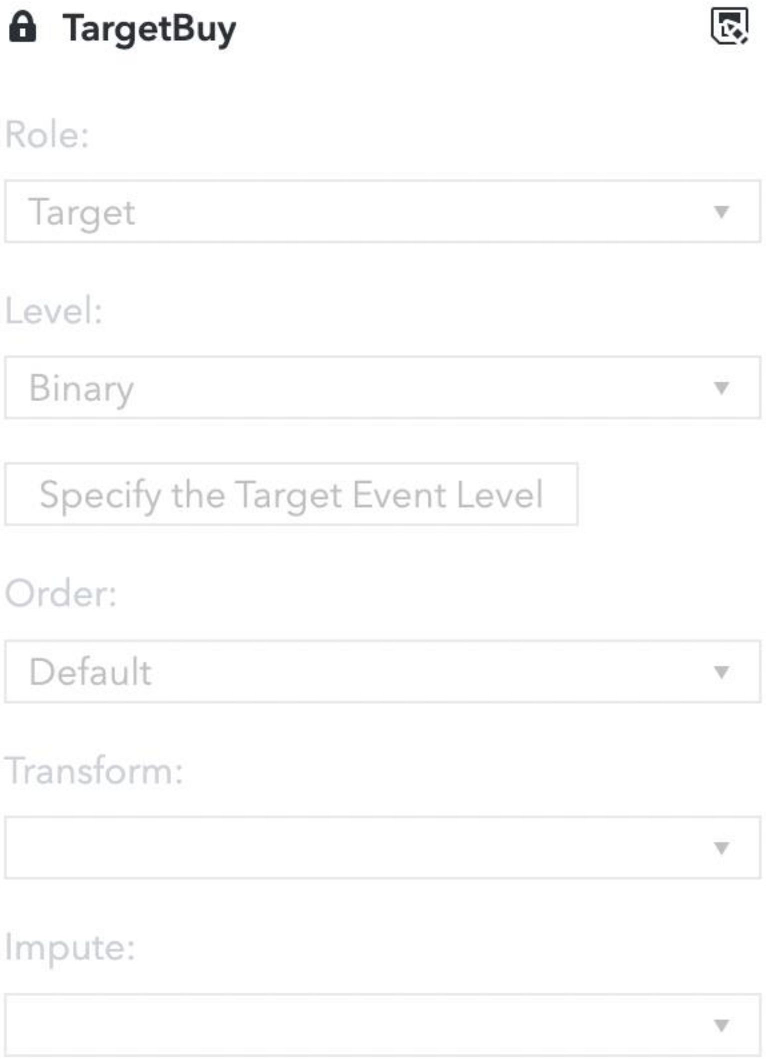
- To present an executive summary of the project, including a Return on Investment (RoI) analysis for BigOrganics Business Case.

By achieving these objectives, the project aims to demonstrate the effectiveness of data mining and machine learning in improving business outcomes for BigOrganics Business Case and provide insights into the potential benefits of using SAS Viya in data analytics.

I will cover different processes that improve our understanding of the data and increment our accuracy. This includes creating data pipeline , data pre-processing , implementing machine learning models, selecting target variables, imputation and transforming input variables as per requirements.

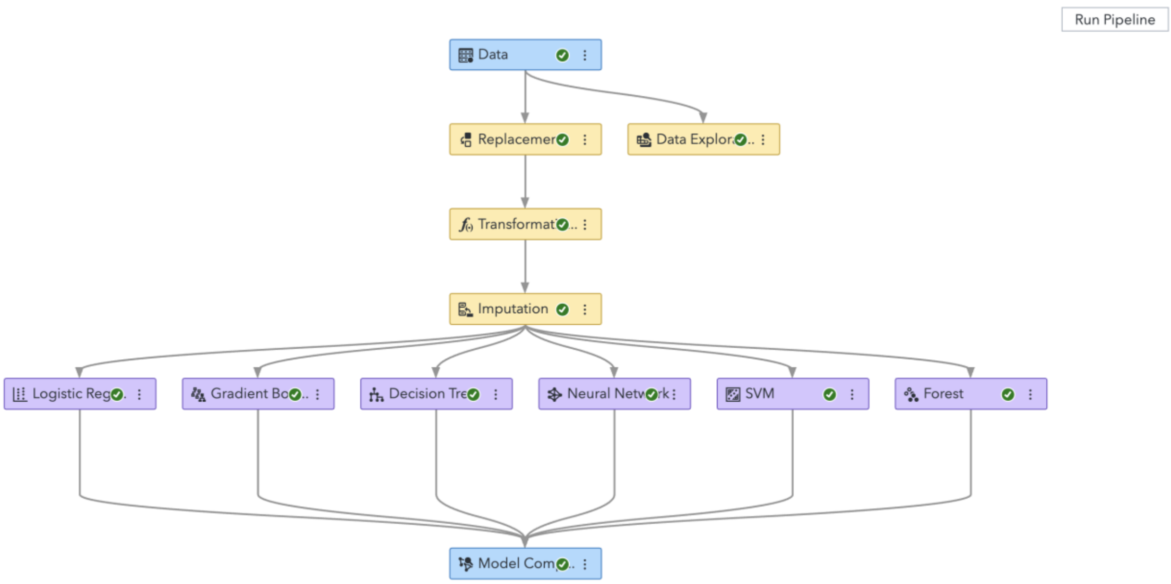
Building Data Pipeline

We initiate the process of building a data learning pipeline using the BigOrganics Dataset. We make sure that we have only 1 target variable (that is TargetBuy in this case).



We select target event level at (1)- 24% and save our data pipeline.

Data Pipeline Architecture

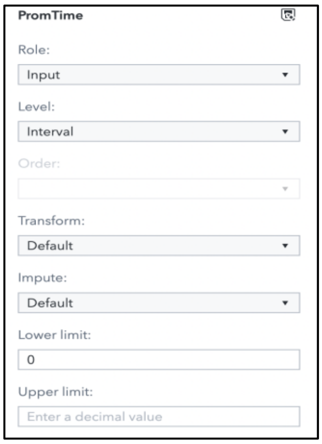


Components of our Data Pipeline –

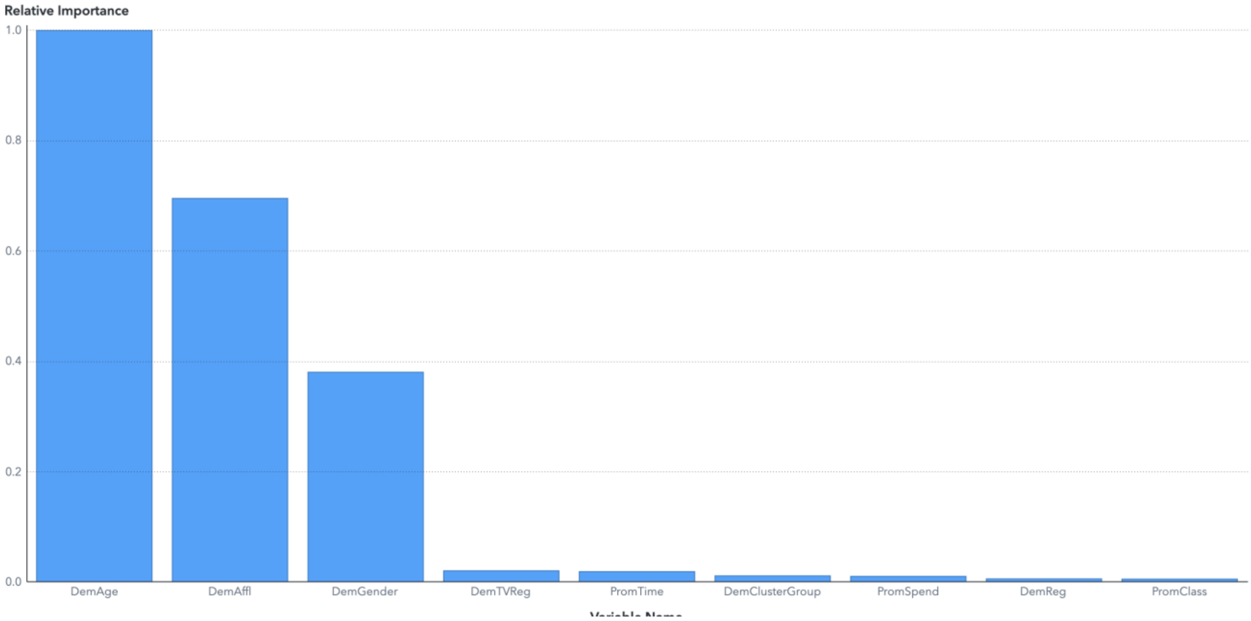
Data – BigOrganics Dataset

Replacement – The replacement node allows us to replace the outliers with unknown class levels with specified values. In our case we first sort our Data Inputs and Roles into ascending order so that it groups the negative values together.

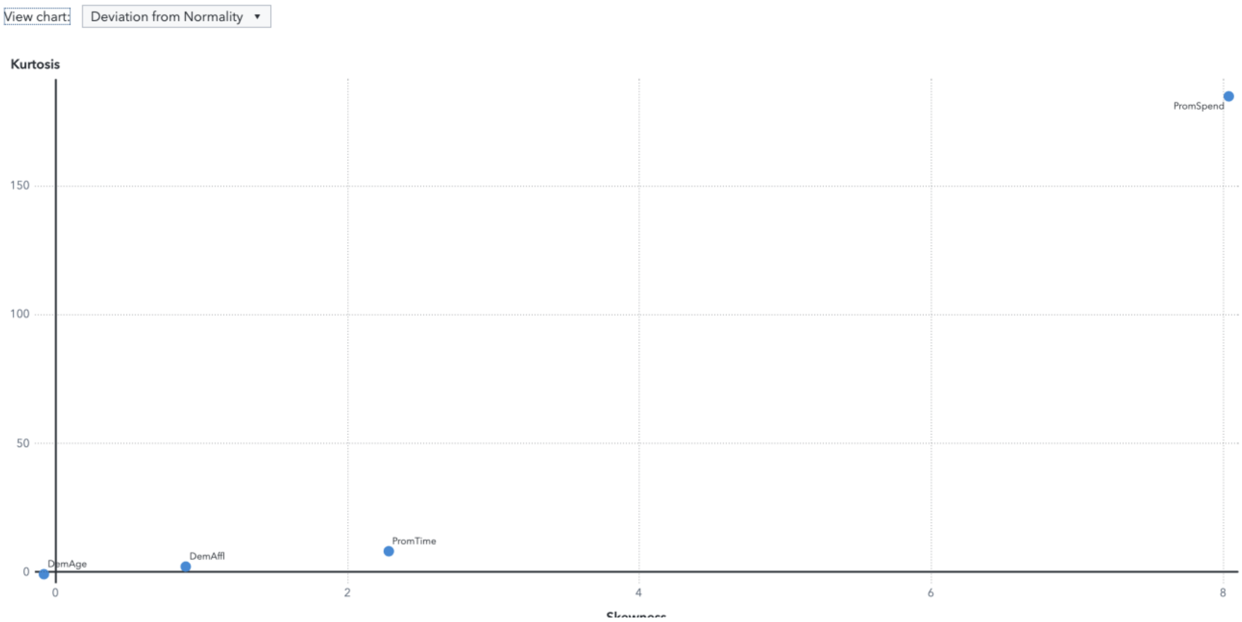
We then select the "numeric" type "input variables" and set their lower limit as 0.



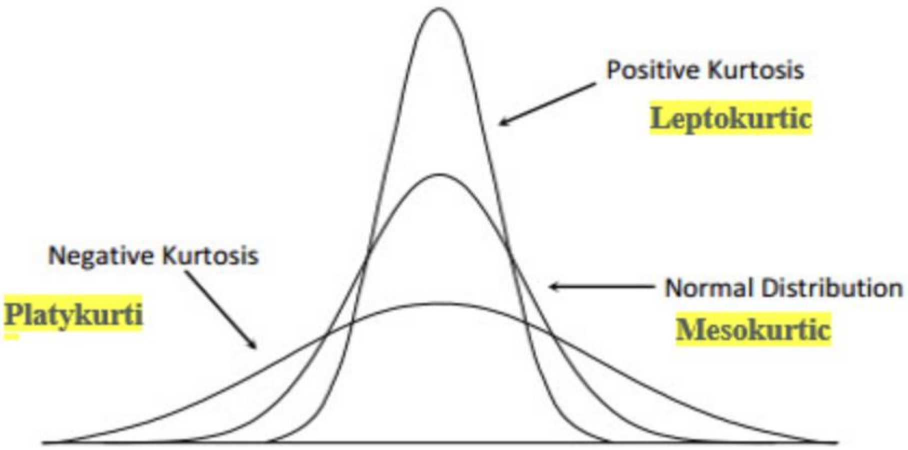
Data Visualization



We observe the relative importance of three main input variables – DemAge, DemAffl and DemGender.



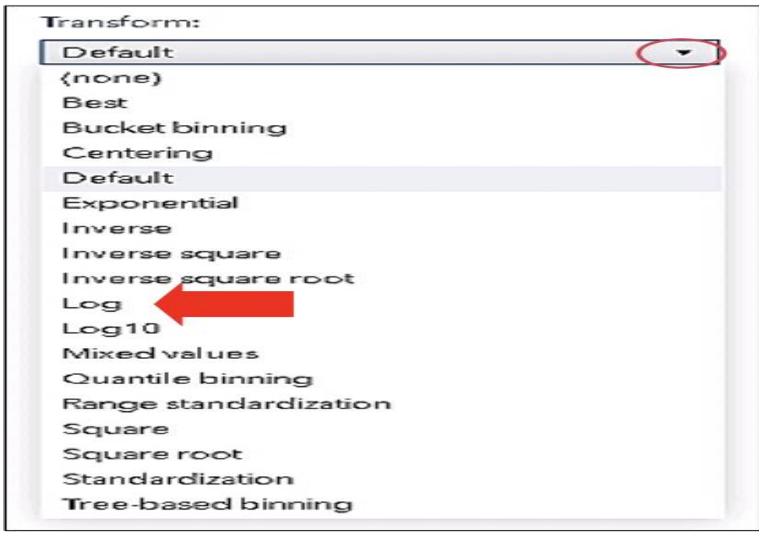
From the kurtosis chart for deviation from normality we can observe that PromSpend has the highest kurtosis and needs to be transformed into a normal distribution as shown below



Transformation

Transforming variables can serve several purposes, such as altering the shape of their distribution by stretching or compressing them, mitigating the impact of outliers or heavy tails, or standardizing inputs to be on the same range and scale. Additionally, transformations can be applied to inputs to minimize bias in model predictions, thereby improving their accuracy and reliability.

We use the log function for transforming the 'PromSpeed' variable.



Imputation

Imputation involves filling in missing values with information derived from non-missing values in the training data. While simple imputation methods, such as replacing a missing value with the mean or mode of the variable's non-missing values, are commonly used, they may not always be suitable for variables with non-normal distributions or a high proportion of missing values. In such cases, simple imputation can significantly alter a variable's distribution and adversely affect predictive accuracy. To address these issues, it is recommended to create missing markers and use them alongside the newly imputed variables in the model. Another approach involves using decision trees to derive imputed values. A decision tree can be trained using a variable with missing values as its target and all other variables as inputs, enabling it to learn plausible replacement values for missing values in the target variable. However, this approach can be computationally expensive for large, dirty training sets, as it requires a decision tree for each input variable with missing values. For small data sets, more sophisticated imputation methods, such as multiple imputation (MI), should be considered.

Input Va...	Variable...	Number...	Percent ...	Imputable	Minimum	Maximum	Mean	Midrange	Standar...	Skewness	Kurtosis	Variable...
DemClusterGroup	NOMINAL	2,359	3.0329	1	.	.	.	.	.	.	.	Neighborhood Cluster-7 Level
DemGender	NOMINAL	8,828	11.3498	1	.	.	.	.	.	.	.	Gender
DemReg	NOMINAL	1,639	2.1072	1	.	.	.	.	.	.	.	Geographic Region
DemTVReg	NOMINAL	1,639	2.1072	1	.	.	.	.	.	.	.	Television Region
PromClass	NOMINAL	0	0	0	.	.	.	.	.	.	.	Loyalty Status
REP_DEMAFFL	INTERVAL	3,815	4.9048	1	0	34	8.7140	17	3.4188	0.8798	1.9927	Replacement: Affluence Grade
REP_DEMAGE	INTERVAL	5,297	6.8101	1	18	79	53.8042	48.5000	13.2273	-0.0802	-0.8450	Replacement: Age
REP_PROMSPEND	INTERVAL	0	0	0	0.0100	296,313.8500	4,413.9100	148,156.9300	7,501.1621	7.7493	168.1661	Replacement: Total Spend
REP_PROMTIME	INTERVAL	962	1.2368	1	0	39	6.5558	19.5000	4.6725	2.2941	8.0882	Replacement: Loyalty Card Tenure

Model Comparison

Model Comparison

Champion	Name	Algorithm Name	Lift	Misclassification Rate
★	Forest	Forest	2.0104	0.1401
	Gradient Boosting	Gradient Boosting	1.9644	0.1831
	Logistic Regression	Logistic Regression	1.7711	0.1969
	SVM	SVM	1.7609	0.2038
	Decision Tree	Decision Tree	1.6525	0.1878
	Neural Network	Neural Network	1.0609	0.2477

From our analysis we can see the best performing model is Forest with a Lift = 2.0104 and Misclassification Rate = 0.1401.

Best Model Analysis – Forest

The SAS System

The FOREST Procedure

Model Information	
Number of Trees	100
Number of Variables Per Split	3
Seed	12345
Bootstrap Percentage	60
Number of Bins	50
Number of Input Variables	9
Maximum Number of Tree Nodes	3127
Minimum Number of Tree Nodes	1487
Maximum Number of Branches	2
Minimum Number of Branches	2
Maximum Depth	20
Minimum Depth	20
Maximum Number of Leaves	1564
Minimum Number of Leaves	744
Maximum Leaf Size	7845
Minimum Leaf Size	5
OOB Misclassification Rate	0.14191126
Average Number of Leaves	1106.8

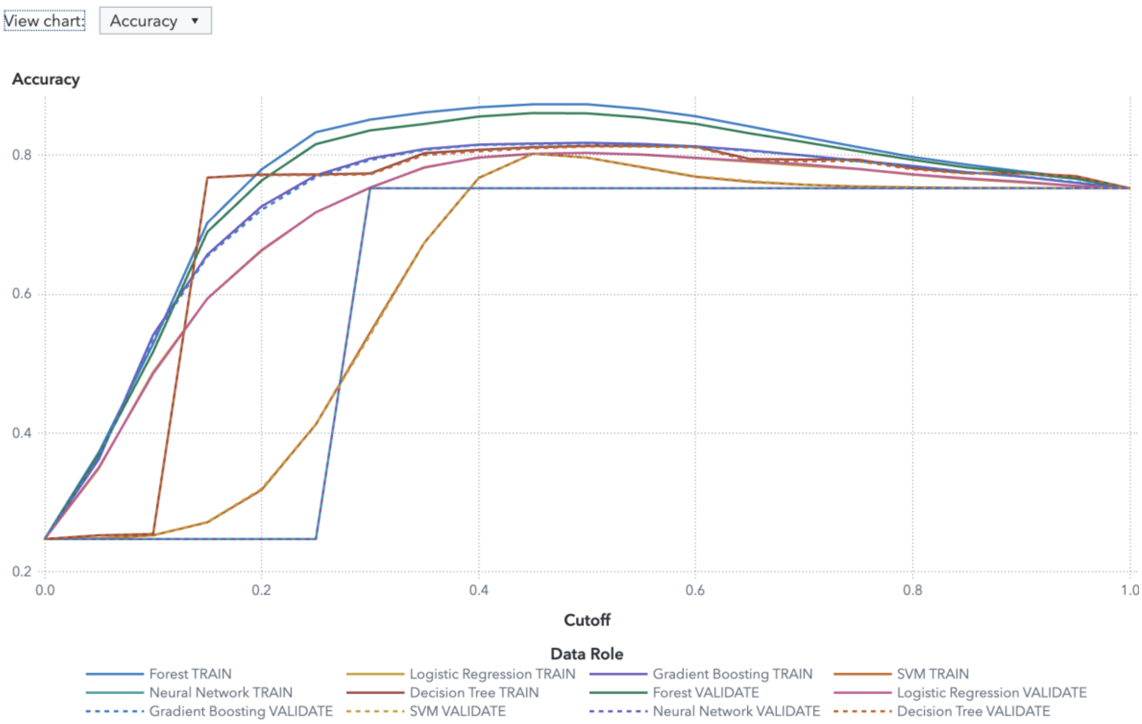
	Training	Validation	Total
Number of Observations Read	77781	33334	111115
Number of Observations Used	77781	33334	111115

ROC (Receiver Operating Characteristic) curve analysis is used to evaluate the performance of binary classification models. It plots the true positive rate (sensitivity) against the false positive rate (1-specificity) over a range of decision thresholds, illustrating the trade-off between the two rates and allowing the selection of a threshold that balances the classification accuracy of positive and negative cases.

ROC curve analysis is a useful tool for evaluating the performance of classification models, as it provides a comprehensive summary of their sensitivity and specificity across different decision thresholds. It allows the comparison of different models and provides insights into the strengths and weaknesses of each. In addition, the area under the ROC curve (AUC) provides a single summary measure of model performance that is commonly used to compare different models. From our graph below that the area under the curve is maximum for Forest.

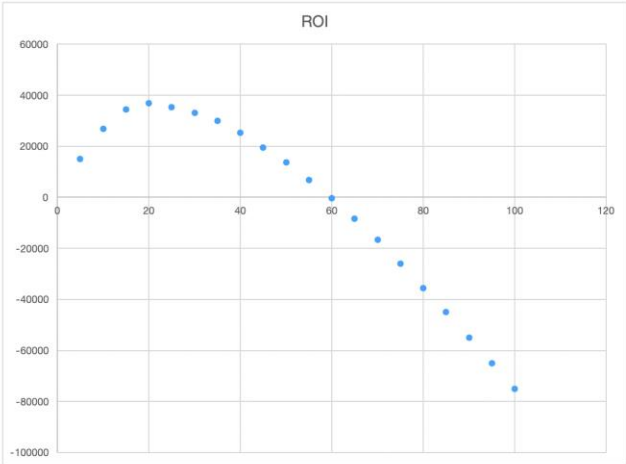
Accuracy = (TP + TN) / (TP + TN + FP + FN)

TP = true positives ,TN = true negatives, FP = false positives & FN = false negatives



Conclusion – Return On Investment

Sum of Frequencies	Depth	Data Role	Cumulative Lift	ROI
1667	5	TEST	3,991764563	14948,52852
1667	10	TEST	3,758023495	26975,29369
1667	15	TEST	3,431431916	34339,34843
1667	20	TEST	3,076177789	36904,44471
1667	25	TEST	2,734164951	35442,65472
1667	30	TEST	2,485164103	33193,65387
1666	35	TEST	2,287236803	30066,61015
1667	40	TEST	2,107908441	25395,42207
1668	45	TEST	1,947707669	19558,55638
1665	50	TEST	1,818820395	13676,27468
1667	55	TEST	1,698613848	6779,702071
1667	60	TEST	1,596019539	-298,5345765
1669	65	TEST	1,498029644	-8285,091437
1665	70	TEST	1,409712971	-16650,11505
1667	75	TEST	1,322029793	-26059,70692
1666	80	TEST	1,245307012	-35469,29878
1667	85	TEST	1,176185625	-45030,27734
1666	90	TEST	1,111111111	-55000
1667	95	TEST	1,052631579	-65000
1665	100	TEST	1	-75000



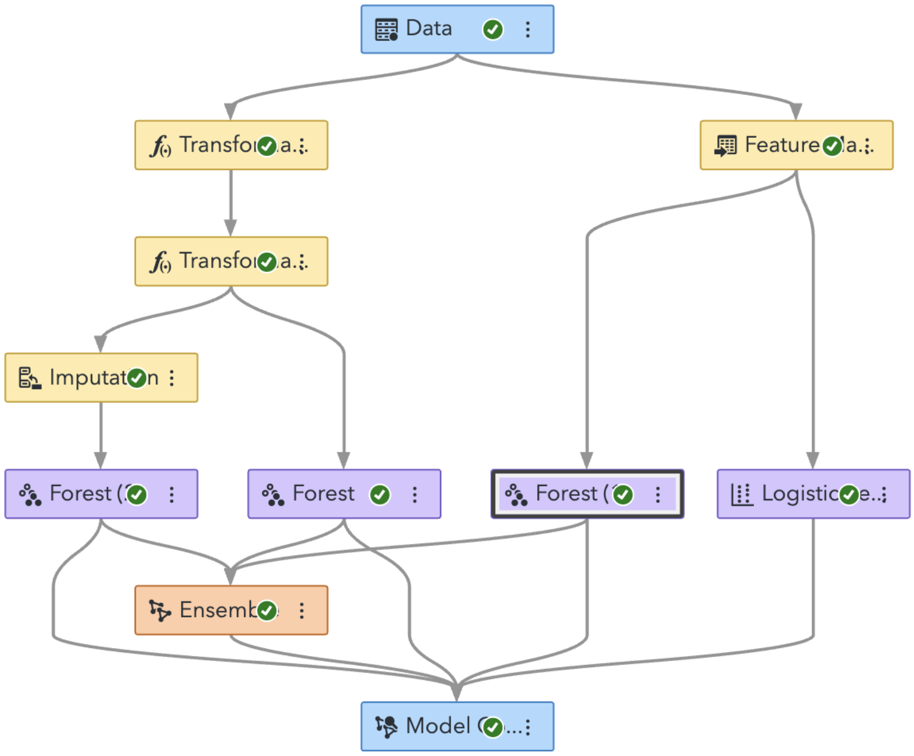
The Organics Table comprises a vast customer base of over 100,000 individuals. Of this customer base, approximately 25% purchase organic products, which serves as the default base rate for analysis. With a cost of £2 per letter, each letter sent to customers incurs this expense. However, the benefits of each letter, such as increased sales revenue, amount to £5.

After exporting the data in CSV and including the above constraints we can find the ROI by using the below formula :-
=100000*A2/100*(-2+5*F2*25/100)

We achieve the highest ROI at 20th Percentile which is equal to $36904.44471

Alternate Method – Generating Automatic Pipeline

We use the power of SAS analytics tool to generate an automatic pipeline after leaving the program to run for 20 minutes. After the given time we got an auto generated pipeline that followed the below data pipeline architecture.

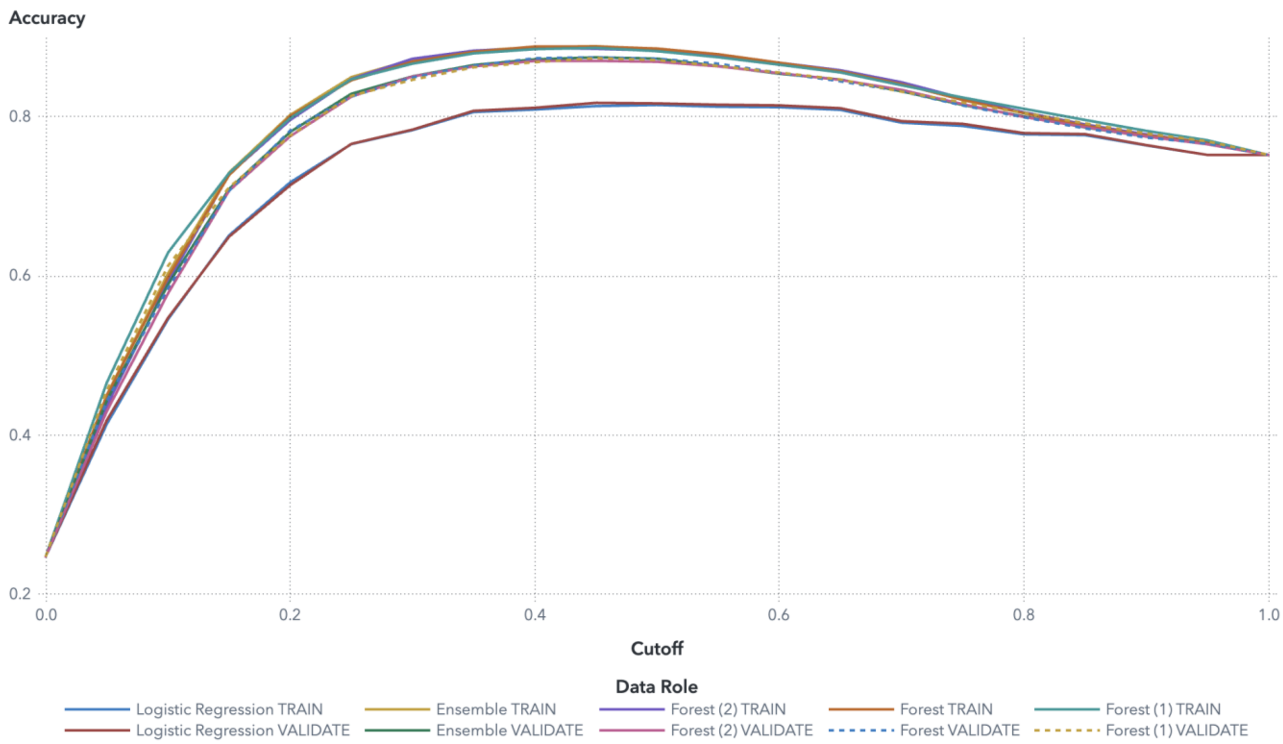


We observe better results as compared to our manual pipeline.

Model Comparison

Champion	Name	Algorithm Name	Lift	Misclassification Rate
	Ensemble	Ensemble	2.3834	0.1268
☆	Forest (1)	Forest	2.3834	0.1287
	Forest	Forest	2.3083	0.1270
	Forest (2)	Forest	2.2236	0.1308
	Logistic Regression	Logistic Regression	1.9834	0.1831

Here again Forest is the best machine learning algorithm for our required dataset



We can further compare our manual and automated generated pipelines to compare our results.

☐	Champion	Name	Algorithm Name	Pipeline Name	Lift	Sum of Frequencies
☑		Forest (1)	Forest	Pipeline 2	2.383	33,334
☐		Forest	Forest	Starter Template	2.010	33,334

In conclusion we can achieve an even better and accurate data pipeline if we use SAS automation tool and let it run for even longer duration but we prefer to create a manual data pipeline to make our understanding of the software simpler and concise for this project.