WQD7005 DATA MINING

1/2023/2024

Churn Prediction of E-Commerce Customer Behavior DataSet

Name	Matric Number
Tee Mei En	22079668

Objective:
This case study aims to gain comprehensive insights into customer behavior, develop and train three models, and subsequently conduct a rigorous comparative analysis to determine the most effective and efficient model for predicting and understanding customer interactions. The overarching goal is to leverage data-driven approaches to enhance our understanding of customer dynamics and optimize decision-making processes within the business.

Data Generation.
An online e-commerce dataset from Kaggle (https://www.kaggle.com/datasets/ankitverma2010/ecommerce-customer-churn-analysis-and-prediction) will be used as the referenced e-commerce dataset for this assignment. This dataset on Kaggle includes key attributes such as "CustomerID" for unique identification, "Churn" indicating customer retention status, and "Tenure" reflecting the duration of customer association. Preferences are captured through "PreferredLoginDevice" and "PreferredPaymentMode," while geographical aspects are represented by "CityTier" and "WarehouseToHome" distance. Metrics like "HourSpendOnApp," "NumberOfDeviceRegistered," and "NumberOfAddress" delve into customer engagement, and "PreferedOrderCat" reveals the favored order category. Financial elements encompass "OrderAmountHikeFromlastYear," "CouponUsed," "OrderCount," and "CashbackAmount." The dataset provides a comprehensive view of customer behavior, satisfaction ("SatisfactionScore"), and feedback ("Complain"), making it a valuable resource for E-commerce analysis and strategic insights.

Variable	Description
CustomerID	Unique customer ID
Churn	Churn Flag
Tenure	Tenure of customer in organization
PreferredLoginDevice	Preferred login device of customer
CityTier	City tier
WarehouseToHome	Distance in between warehouse to home of customer
PreferredPaymentMode	Preferred payment method of customer
Gender	Gender of customer
HourSpendOnApp	Number of hours spend on mobile application or website
NumberOfDeviceRegistered	Total number of devices registered on particular customer
PreferedOrderCat	Preferred order category of customer in last month
SatisfactionScore	Satisfactory score of customer on service
MaritalStatus	Marital status of customer
NumberOfAddress	Total number of address added on particular customer
Complain	Any complaint has been raised in last month
OrderAmountHikeFromlastYear	Percentage increases in order from last year
CouponUsed	Total number of coupon has been used in last month
OrderCount	Total number of orders has been places in last month
DaySinceLastOrder	Day Since last order by customer
CashbackAmount	Average cashback in last month

Figure 1: Definition of Variables in Original Dataset.

The structure of the Kaggle dataset aligns closely with the requirements outlined in the assignment guidelines. Essential variables such as CustomerID, Churn, and Gender are already present in the dataset. To enhance its suitability for the assignment, the "WarehousetoHome"

variable is proposed as a replacement for the "Location" variable. "WarehousetoHome" provides a tangible measure of the geographical distance between a customer's home and the warehouse, offering insights into delivery times and service coverage.

City Tier, while initially included, may not align with the project goals during the Exploration phase and might be dropped in the modify phase. The Membership Level requested by the Guideline will be created by incorporating City Tier, HourSpendOnApp, and Tenure in the Kaggle dataset. Higher city tiers may imply better financial backgrounds, longer tenure signifies loyalty, and increased hours spent on the app indicates higher engagement—factors influencing membership levels.

And so, the variable Membership Level is generated under the following conditions:
1. Default level is No membership: Labeled as “Not Member”
2. Membership Level should be Bronze if they are in city tiers 1 and 2, have a tenure greater than 2, and spend more than 2 hours on the app.
3. Membership Level should be Silver if they are in city tiers 2 and 3, have a tenure greater than 4, and spend more than 2 hours on the app.
4. Gold if they are in city tiers 2 and 3, have a tenure greater than 8, and spend more than 3 hours on the app.
5. Platinum if they are in city tier 3, have the highest tenure, and spend more than 3 hours on the app.

```
# Create a new 'MembershipLevel' column based on conditions
df['MembershipLevel'] = 'None'  # Default level

# Upgrade to Bronze for customers in higher city tiers with tenure > 2 and hours spent on app > 1
df.loc[(df['CityTier'].isin([1, 2])) & (df['Tenure'].gt(2)) & (df['HourSpendOnApp'].gt(2)), 'MembershipLevel'] = 'Bronze'

# Upgrade to Silver for customers in higher city tiers with tenure > 2 and hours spent on app > 2
df.loc[(df['CityTier'].isin([2, 3])) & (df['Tenure'].gt(4)) & (df['HourSpendOnApp'].gt(2)), 'MembershipLevel'] = 'Silver'

# Upgrade to Gold for customers with longer tenure and higher hours spent on the app
df.loc[(df['CityTier'].isin([2, 3])) & (df['Tenure'].gt(8)) & (df['HourSpendOnApp'].gt(3)), 'MembershipLevel'] = 'Gold'

# Upgrade to Platinum for customers with the highest tenure, CityTier equal to 3, and hours spent on app > 3
df.loc[(df['CityTier'].eq(3)) & (df['Tenure'].gt(16)) & (df['HourSpendOnApp'].gt(3)), 'MembershipLevel'] = 'Platinum'

# Print the first few rows of the DataFrame with the new 'MembershipLevel' column
df.head()
```

Figure 2: Jupyter Notebook Code to Generate Variable Membership Level.

The Total Purchases represent the total number of purchases made by the customer. This will be replaced by the OrderCount variable in the Kaggle dataset as this variable basically calculates the total number of orders placed in the last month, which is very similar in definition to Total Purchases. The PreferedOrderCat in the Kaggle dataset is equivalent to the FavoriteCategory in guideline while the DaySinceLastOrder in Kaggle dataset is equivalent to LastPurchaseDate. The Additional Attributes not implicitly or explicitly mentioned by the guidelines but are present in the Kaggle dataset are: PreferredLoginDevice, PreferredPaymentMode, HourSpendOnApp, NumberOfDevicesRegistered, SatisfactionScore, Marital Status, NumberOfAddress, Complain, OrderAmountHikeFromLastYear, CouponUsed,

CashbackAmount. The official dataset used for this assignment is then saved through Jupyternotebook into a csv file as shown in figure 3.

```
# Save the df into a .csv file

new_file_path = "./ecommerce_data.csv"
df.to_csv(new_file_path,index=False)
```

Figure 3: Creation of Official Dataset

With the official dataset now created, we can now proceed with applying SEMMA methodology as well as incorporating the two tools—Talend Data Prep, Knime, and SAS Enterprise Miner—-for the goal of this data mining assignment.

Sample.
A brief glance at the data shows that the dataset has around 5630 rows. The current dataset is not too big to the point that it causes constraint to computational resources, hence why the current decision is not to sample the dataset. An in-depth exploration should be done to unveil more issues and if sampling is deemed necessary by the exploration, we will return to the sampling step.

Beginning Exploration with Talend Data Prep.
For the beginning of the exploration, the data was first imported into Talend Data Prep.

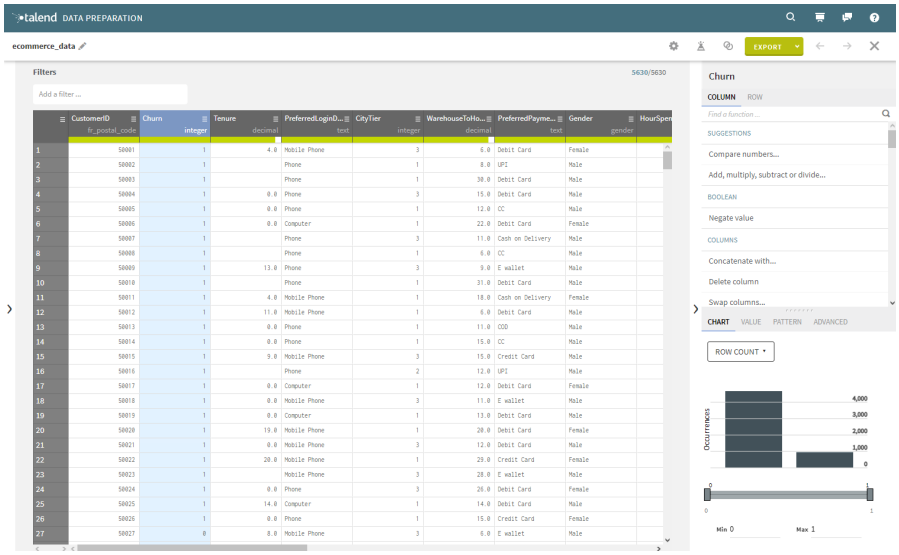

Figure 4: Data Profiling in Talend Data Prep (Part 1)

	HourSpendOnApp	NumberOfDevic...	PreferedOrderCat	SatisfactionScore	MaritalStatus	NumberOfAddress	Complain	OrderAmountHi...
	decimal	integer	text	integer	text	integer	integer	decimal
6	3.0	5	Mobile Phone	5	Single	2	1	22.0
7	2.0	3	Laptop & Accessory	2	Divorced	4	0	14.0
8	3.0	3	Mobile	2	Divorced	3	1	16.0
9		4	Mobile	3	Divorced	2	1	14.0
10	2.0	5	Mobile	3	Single	2	0	12.0
11	2.0	3	Others	3	Divorced	2	0	
12	3.0	4	Fashion	3	Single	10	1	13.0
13	2.0	3	Mobile	3	Single	2	1	13.0
14	3.0	4	Mobile	3	Divorced	1	1	17.0
15	3.0	4	Fashion	2	Single	2	0	16.0
16	3.0	3	Mobile	5	Married	5	1	22.0

Figure 5: Data Profiling in Talend Data Prep (Part 2)

CouponUsed	OrderCount	DaySinceLastOr...	CashbackAmount	MembershipLevel
decimal	decimal	decimal	decimal	text
4.0	6.0	7.0	139.19	Not Member
0.0	1.0	0.0	120.86000000000001	Not Member
2.0	2.0	0.0	122.93	Not Member
0.0	1.0	2.0	126.83000000000001	Not Member
1.0	1.0	1.0	122.93	Not Member
9.0	15.0	8.0	295.45	Not Member
0.0	1.0	0.0	153.81	Bronze
2.0	2.0	2.0	134.41	Not Member

Figure 6: Data Profiling in Talend Data Prep (Part 3)

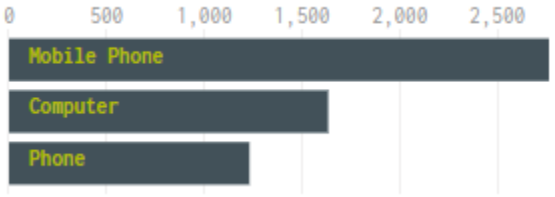

Figure 7: Chart of PreferredLoginDevice

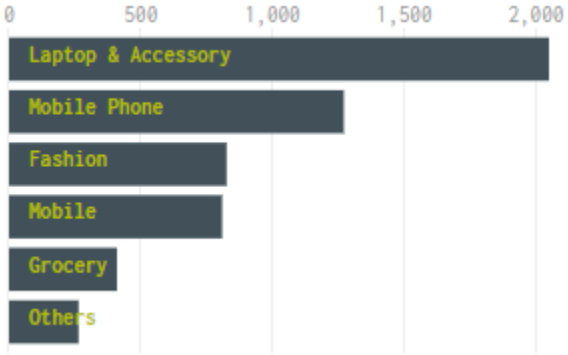

Figure 8: Chart of PreferedOrderCat

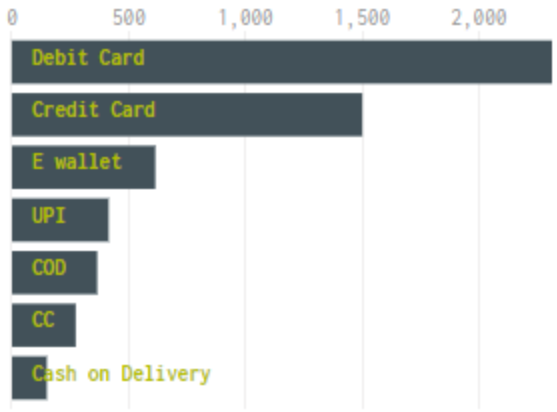

Figure 9: Chart of PreferredPaymentMode

CashbackAmount ☰ decimal
159.93
120.9
120.28
134.07
129.6
139.19
120.86000000000001
122.93
126.83000000000001
122.93
295.45
153.81
134.41
133.88
196.19
120.72999999999999

Figure 10: Rounding Inconsistencies in CashbackAmount.

Figure 4-6 gives an overview of the dataset. For the variable “Churn,” Talend Data Prep shows that there are 4682 occurrences of 0 while only 948 occurrences of 1. Through the bar chart on the bottom right of figure 4, the churn data can be visualized as an imbalance data. This indicates that sampling should be done before the model building phase as imbalance data

can result in a bias toward the majority class, which might indirectly contribute to challenges that resemble overfitting or underfitting.

Through the Quality Bars of the Talend Data Prep, it is clear that the variables “Tenure”, “WarehouseToHome”, “HourSpendOnApp”, “OrderAmountHikeFromlastYear”, “CouponUsed”,”OrderCount”, “DaySinceLastOrder” have missing values. A first glance of figure 4-5 shows that there are inconsistent naming in the variables “PreferredLoginDevice”, “PreferedOrderCat”, “PreferredPaymentMode.” When the charts of these two variables are clicked as shown in figure 7-8, it shows that the category of phones appeared more than once. (Figure 7 showing “Mobile Phone” and “Phone” while Figure 8 showing “Mobile Phone” and “Mobile.” Figure 9 shows the Inconsistent Naming in “PreferredPaymentMode” being CC and Credit Card and COD and Cash on Delivery.Figure 10 reveals the inconsistencies in rounding for the variable CashbackAmount.

Modify with Talend Data Prep

With the beginning exploration done, some of the issues identified will then be addressed in Talend Data Prep.

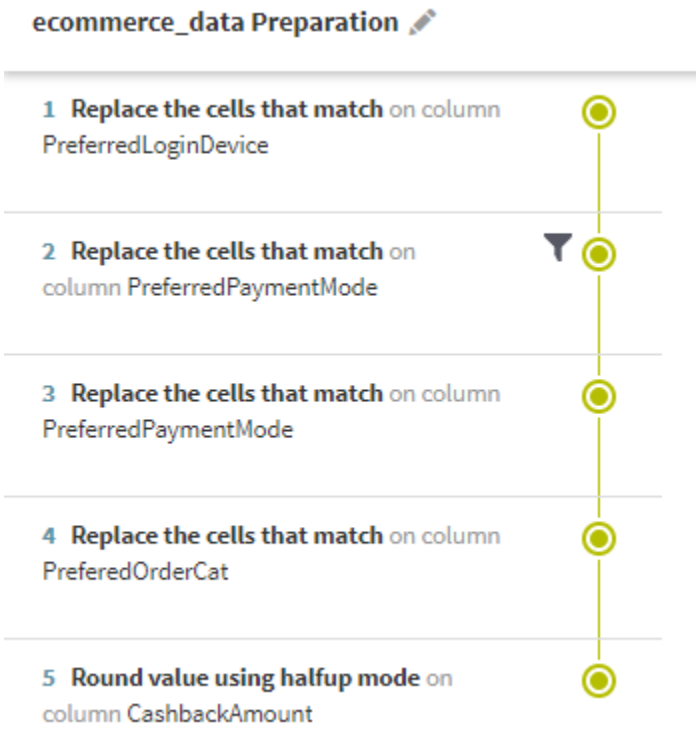

Figure 11: Correction of Issues in Talend Data Prep

As shown in figure 11, the naming inconsistent issues were first corrected using the replacement function. The text “Mobile Phone” was replaced with “Phone” in the PreferredLoginDevice variable.“CC” was replaced with “Credit Card” in the PreferredPaymentMode variable. “COD” was replaced with “Cash on Delivery” in the PreferredPaymentMode variable.“Mobile Phone”

was replaced with “Mobile” in PreferedOrderCat variable. Lastly, the rounding format inconsistency as shown in figure 10 was corrected using Round value using half up mode with precision being 2.

Missing values occasionally communicate insights. Therefore, the missing values identified in Talend Data Prep will not be cleaned for now. The missing values will be dealt with after a more advanced exploration using SAS Enterprise Miner.

Dataset Workflow in Knime

With some preliminary exploration and formatting done with Talend Data Prep, we further use Knime to establish a workflow for eventually preparing the data for modeling.

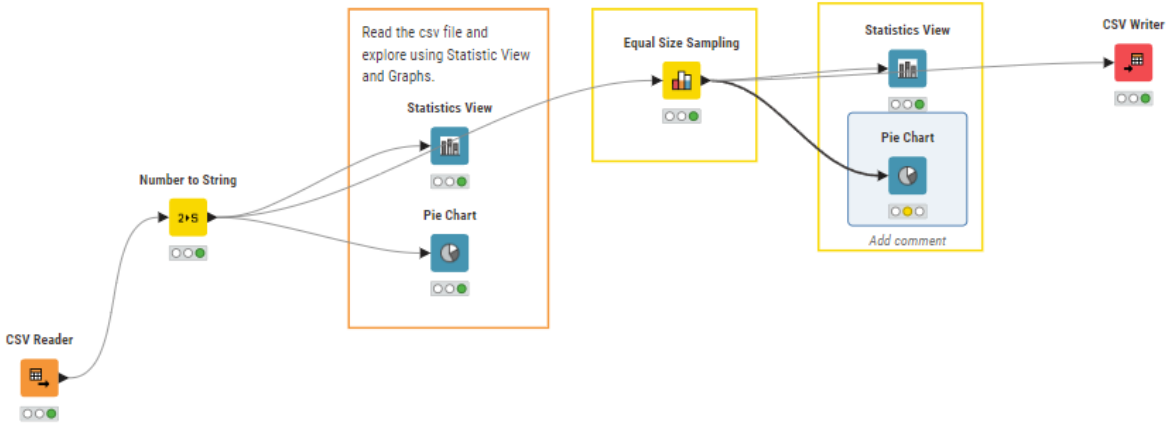

Figure 12: Knime Diagram Workflow

Exploring with Knime

Within Knime, a CSV Reader node was used to read the file. Due to Knime not having a binary category, the number to string node was used to convert the Churn integer variable to a string. The dataset was first explored using Statistics View and a Pie Chart was generated for a visualization of the distribution of the target Variable Churn.

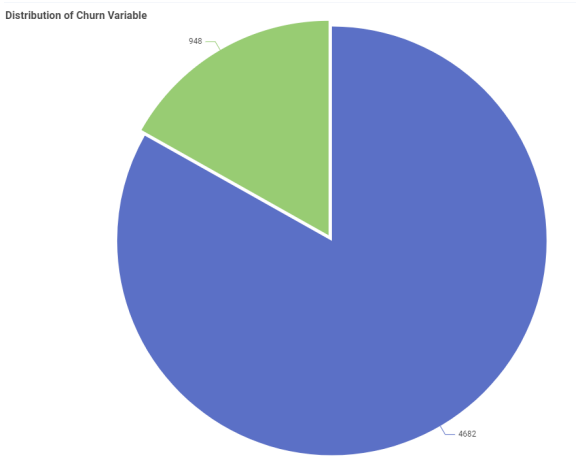

Figure 13: Pie Chart of Churn Variable

The pie chart of the churn variable generated in Knime confirms the imbalance state of the variable as shown through Talend Data Prep. With this confirmation, it shows the need for the balancing of data.

Returning to Sampling Step with Knime.

The Equal Size Sampling Node in Knime was utilized for approximate sampling with the objective of creating a more balanced distribution of the 'Churn' variable. This aims to mitigate the risk of the model being overly influenced by the majority class, leading to improved model sensitivity.

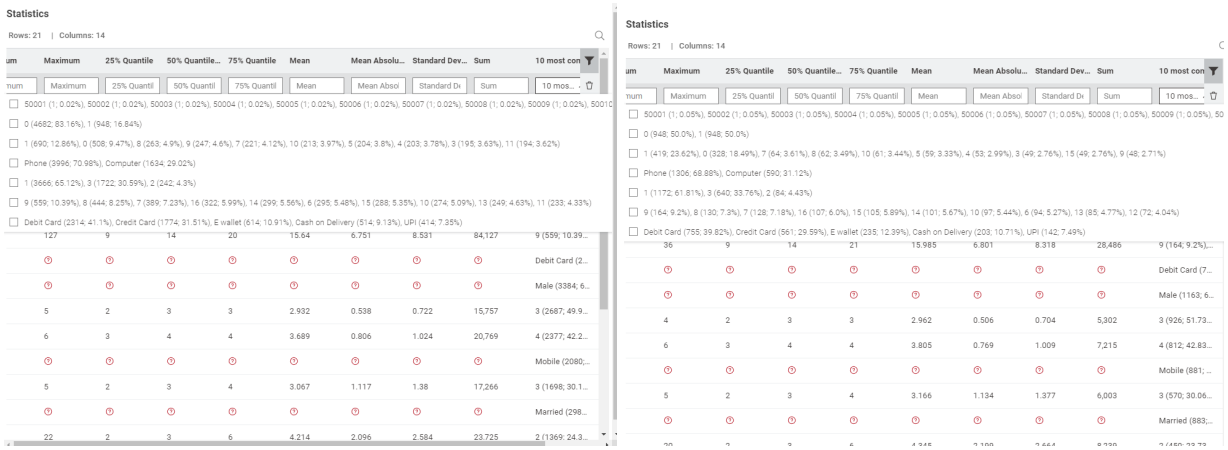

Figure 14: Comparison of Distribution for Data Before and After Sampling

The Statistics View (LEFT) was done on data prior to sampling and vice versa. The comparison of the statistical views showed that the size of the data has reduced from 5630 to 1919. Considering that the model choices are decision tree and random forest and both of these models show robustness to dataset sizes, the reduction in size should not affect the training of model performance. The Equal Size Sampling node was set to retain the proportion of each stratum as close as possible.The comparison of the statistical views showed that the proportion before and after sampling are very similar, which means this is a good sample that reflects the population.

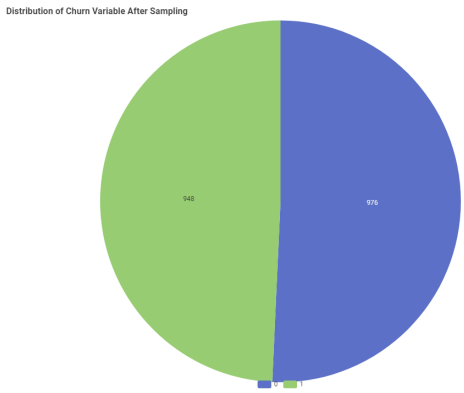

Figure 15: Distribution of Churn Variable After Sampling

SAS Enterprise Miner for Exploratory Data Analysis of Sample Data

Using the file import function, the dataset prepped with Talend Data Prep and Knime was first imported into SAS Enterprise Miner using the File Import Node. The variables were then edited to ensure that the role and levels are correct.

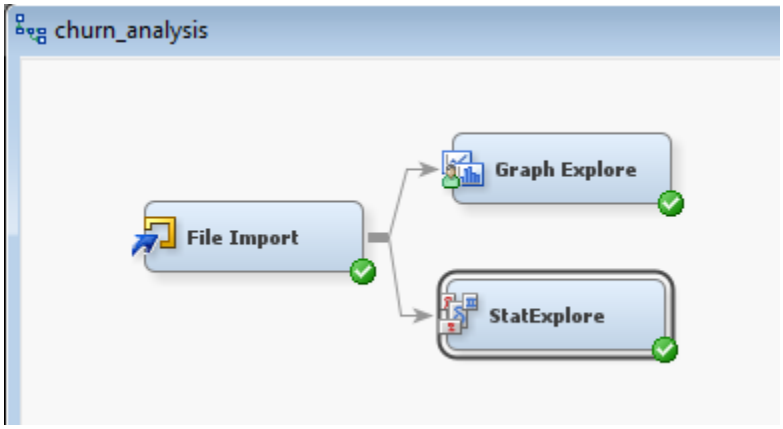

Figure 15: Exploration Step in SAS Enterprise Miner

Name	Role	Level	Report	Order	Drop	Lower Limit	Upper Limit
CashbackAmoun	Input	Interval	No		No	.	.
Churn	Target	Binary	No		No	.	.
CityTier	Input	Ordinal	No		No	.	.
Complain	Input	Interval	No		No	.	.
CouponUsed	Input	Interval	No		No	.	.
CustomerID	ID	Unary	No		No	.	.
DaySinceLastOr	Input	Interval	No		No	.	.
Gender	Input	Nominal	No		No	.	.
HourSpendOnAp	Input	Interval	No		No	.	.
MaritalStatus	Input	Nominal	No		No	.	.
MembershipLeve	Input	Nominal	No		No	.	.
NumberOfAddre	Input	Interval	No		No	.	.
NumberOfDevice	Input	Interval	No		No	.	.
OrderAmountHik	Input	Interval	No		No	.	.
OrderCount	Input	Interval	No		No	.	.
PreferedOrderC	Input	Nominal	No		No	.	.
PreferredLoginD	Input	Nominal	No		No	.	.
PreferredPayme	Input	Nominal	No		No	.	.
SatisfactionScor	Input	Ordinal	No		No	.	.
Tenure	Input	Interval	No		No	.	.
WarehouseToHo	Input	Interval	No		No	.	.

Figure 16: Variable Editing In File Import

It is important to gain a more in-depth understanding of the dataset to better understand whether some features should be dropped as it might generate noise during the modeling step. As shown in figure 12, the data was explored using Graph Explore and StatExplore nodes.

Property	Value
Train	
Variables	...
Data	
Number of Observations	ALL
Validation	No
Test	No
Standard Reports	
Interval Distributions	Yes
Class Distributions	Yes
Level Summary	Yes
Use Segment Variables	No
Cross-Tabulation	...
Variable Selection	
Hide Rejected Variables	Yes
Number of Selected Variables	1000
Chi-Square Statistics	
Chi-Square	Yes
Interval Variables	Yes
Number of Bins	5
Correlation Statistics	
Correlations	Yes
Pearson Correlations	Yes
Spearman Correlations	No

Figure 17: StatExplore Setting

Class Variable Summary Statistics
(maximum 500 observations printed)

Data Role=TRAIN

Data Role	Variable Name	Role	Number of Levels	Missing	Mode	Mode Percentage	Mode2	Mode2 Percentage
TRAIN	CityTier	INPUT	3	0	1	60.50	3	34.97
TRAIN	Gender	INPUT	2	0	Male	61.80	Female	38.20
TRAIN	MaritalStatus	INPUT	3	0	Married	47.00	Single	39.13
TRAIN	MembershipLevel	INPUT	5	0	Not Member	63.94	Bronze	22.67
TRAIN	PreferedOrderCat	INPUT	5	0	Mobile	44.76	Laptop & Accessory	31.47
TRAIN	PreferredLoginDevice	INPUT	2	0	Phone	70.09	Computer	29.91
TRAIN	PreferredPaymentMode	INPUT	5	0	Debit Card	39.50	Credit Card	29.70
TRAIN	SatisfactionScore	INPUT	5	0	3	29.23	5	22.62
TRAIN	Churn	TARGET	2	0	0	50.60	1	49.40

Figure 18: SAS StatExplore Result: Class Variable Summary Statistics

The StatExplore Node's Class Variable Summary Statistics, as shown in Figure 18, presents a summary of key statistics for various categorical variables in a dataset. All of the categorical variables do not have any missing values.

For CityTier, the mode is "1" with a percentage of 60.50%, suggesting that category "1" is the most prevalent. Similarly, in the Gender variable, "Male" is the mode with a percentage of 61.80%, indicating a higher prevalence of males in the dataset. Both the MembershipLevel variable and PreferedOrderCat, the percentages are distributed across multiple categories without a single dominant preference. For the satisfaction score, the mode has the value 3 but a percentage of 29.23%, which means that customer satisfaction within the dataset is diverse, with a range of experiences rather than a concentrated preference for a particular satisfaction level. The PreferredPaymentMode variable shows that the two most prevalent payment modes are "Debit Card" and "Credit Card," with percentages of 39.50% and 29.70%, respectively, meaning that there is a strong preference for debit card usage. In addition, the PreferreLoginDevice variable analysis show that "Phone" is the preferred login device for majority of the customer. The target variable, Churn, now displays a 50.60% majority for class 0 (non-churn) and a 49.49% for class 1 (churn), which shows that this dataset is now balanced.

Interval Variable Summary Statistics
(maximum 500 observations printed)

Data Role=TRAIN

Variable	Role	Mean	Standard Deviation	Non Missing	Missing	Minimum	Median	Maximum	Skewness	Kurtosis
CashbackAmount	INPUT	171.2684	45.7321	1919	0	0	157.11	324.99	1.342415	1.653398
Complain	INPUT	0.384575	0.486622	1919	0	0	0	1	0.474887	-1.77633
CouponUsed	INPUT	1.781233	1.94847	1865	54	0	1	16	2.514005	8.975384
DaySinceLastOrder	INPUT	4.049614	3.64091	1814	105	0	3	46	1.660002	9.563346
HourSpendOnApp	INPUT	2.946733	0.714771	1821	98	0	3	5	0.023897	-0.75243
NumberOfAddress	INPUT	4.343408	2.666821	1919	0	1	3	20	1.010593	0.609275
NumberOfDeviceRegistered	INPUT	3.760813	1.04318	1919	0	1	4	6	-0.34897	0.537319
OrderAmountHikeFromlastYear	INPUT	15.62075	3.666338	1851	68	11	15	26	0.862262	-0.14274
OrderCount	INPUT	2.965854	2.911646	1845	74	1	2	16	2.281132	5.301953
Tenure	INPUT	7.625763	8.167407	1801	118	0	5	61	1.080426	0.842986
WarehouseToHome	INPUT	16.12396	8.780523	1807	112	5	14	126	1.745004	12.6385

Figure 19: SAS StatExplore Result: Interval Variable Summary Statistics

The StatExplore's Interval Variable Summary Statistics, as shown in Figure 15, show consistent results as the Talend Data Prep's finding. Of the 11 interval variables, 7 of them show missing values. The dataset's interval variables also provide detailed insights into customer behavior. The CashbackAmount variable indicates a moderate variability, with an average of $177.22 and a positive skewness of 1.15, suggesting a right-tailed distribution and the presence of customers receiving higher cashback amounts. The Complain variable shows a low mean of 0.28, indicating infrequent complaints, with a positive skewness of 0.95 suggesting a subset of customers complaining more frequently. CouponUsed has a mean of 1.75 and a wide range (0 to 16), with positive skewness (2.55), indicating some customers use coupons more frequently. DaySinceLastOrder has a mean of 4.54, and positive skewness (1.19), suggesting a group of customers making orders more frequently than the average. HourSpendOnApp shows an average of 2.93 hours, with a slightly left-skewed distribution (skewness -0.03) indicating some customers spend less time on the app. NumberOfAddress has a mean of 4.21, positive skewness (1.09), suggesting a group with a higher number of addresses. NumberOfDeviceRegistered averages 3.69 with slightly negative skewness (-0.40), indicating some customers have fewer devices registered. OrderAmountHikeFromlastYear has a mean of 15.71%, positive skewness (0.79), indicating a group experiencing higher order amount hikes. OrderCount has a mean of 3.01 with positive skewness (2.20), suggesting a group with a higher order count. Tenure averages 10.19 with positive skewness (0.74), reflecting varied customer loyalty. WarehouseToHome has a mean of 15.64 with positive skewness (1.62), indicating varied distances. These specific values offer a comprehensive understanding of customer behaviors and characteristics within the dataset.

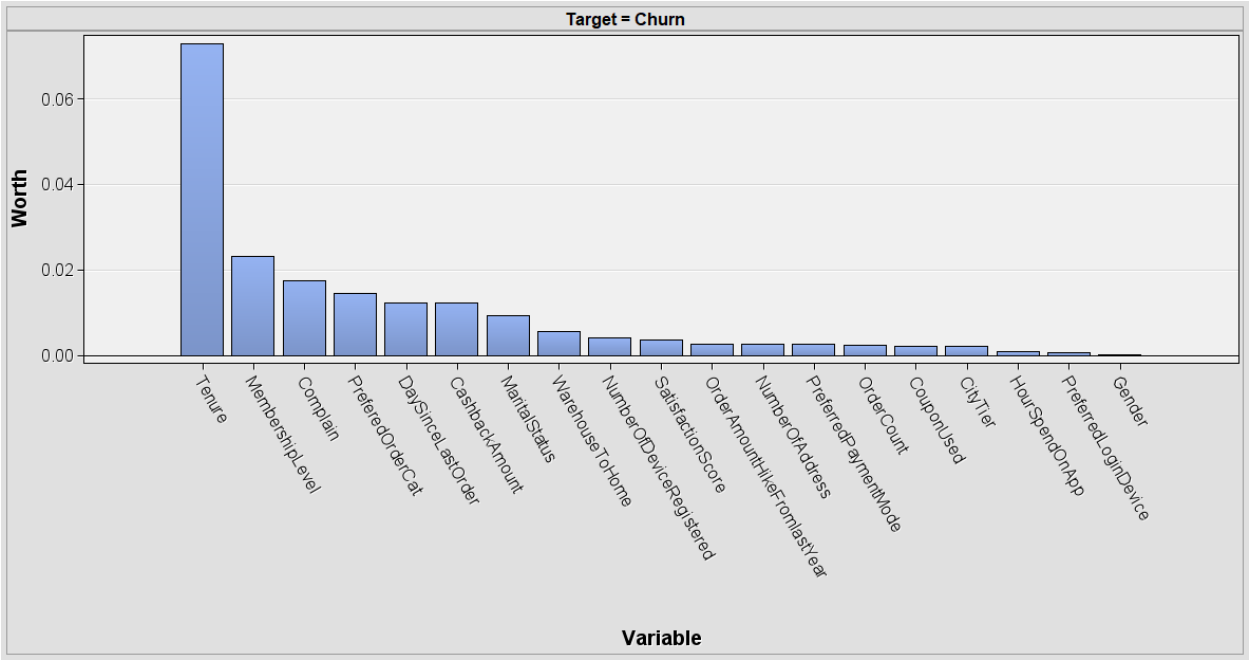

Figure 20: Variable Worth Chart of Data

The variable worth chart suggests that these ten variables—Tenure, MembershipLevel, Complain, PreferredOrderCat, DaysSinceLastOrder, CashbackAmount, MaritalStatus,

WarehouseToHome, NumberOfDeviceRegistered, and SatisfactionScore—are considered the highest contributing variables in predicting the target variable.

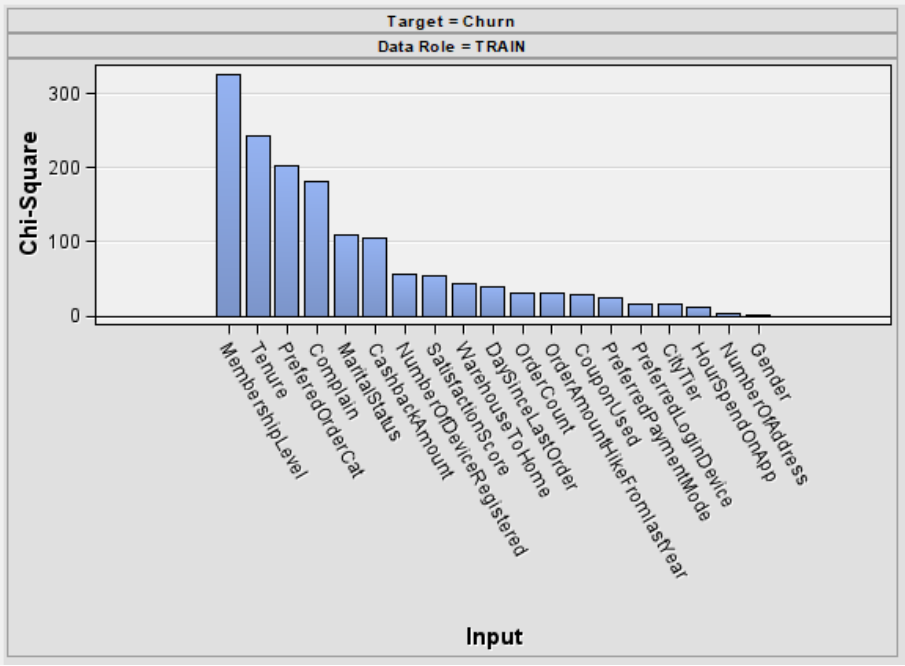

Figure 21: The Chi-Square Statistic Bar Chart

The Chi-Square Statistics of the variables reveal similar findings as the variable worth chart in which MembershipLevel, Tenure, PreferredOrderCat, and the other variables share highly significant association with the target variable “Churn.” . However, NumberOfAddress and Gender exhibit non-significant associations. These results inform feature selection for predictive modeling.

Modify: SAS Enterprise Miner to Perform More Advanced Data Imputation.

Imputation.

The identification of outliers are very important to determine which central tendency values would be more suitable for the imputation method.

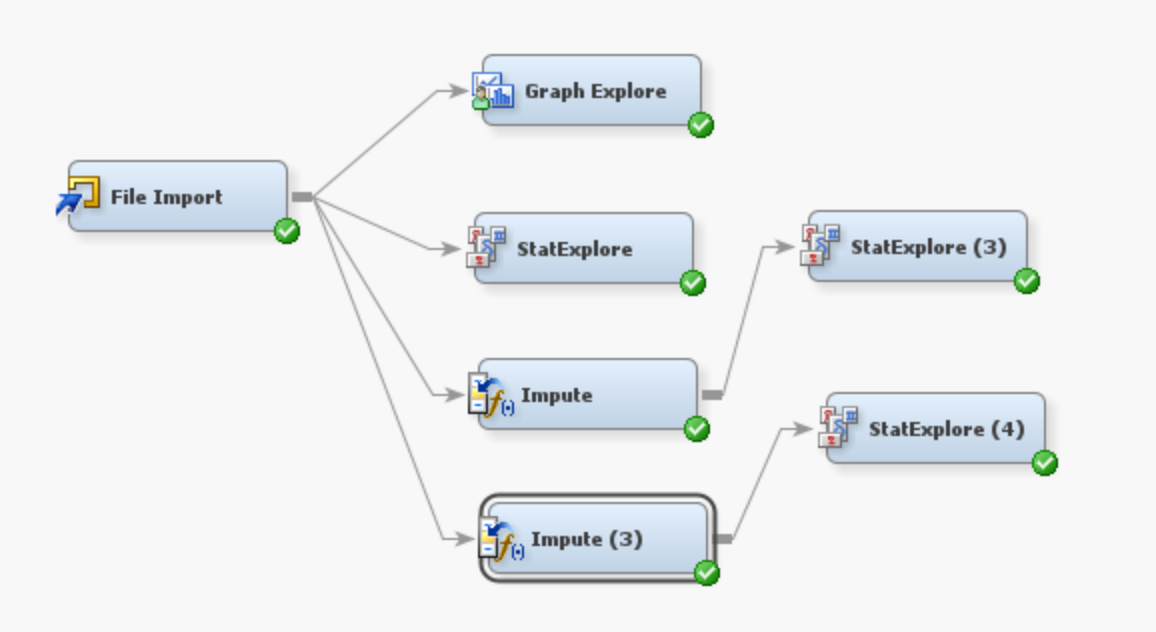

Figure 22: Addition of Imputation Nodes to Diagram for Comparison

As shown in the exploration phase, the variables CashbackAmount, CouponUsed, DaySinceLastOrder, NumberofAddress, OrderAmountHikeFromLastYear, OrderCount, Tenure, and WarehouseToHome show positive skewness and relatively high maximum value, which means that using the central tendency median might be more suitable while the the mean value can be used to impute the missing value in the other variables. Through the explore step, it was identified that among the 7 variables with missing values, only HourSpendOnApp has a rather normal distribution, which supposedly could be imputed using mean. However, the HourSpendOnApp was decided to also be imputed using the default setting median due to the mean of this variable being 2.946733 which is approximately the same as its median 3. And so, all seven interval variables with missing values were imputed with median. Two types of imputation method—Imputing with Median and Imputing with Tree Surrogate—were done and a comparison table was made to decide on the more suitable imputation method for each variable.

Table 1: Comparison of Imputation Methods

	Original	**Impute with Median**	**Impute with Tree Surrogate**
CouponUsed	Mean: 1.781233 SD: 1.94847	Mean: 1.75925 SD: 1.925187	Mean: 1.81452 SD: 1.966894
DaySinceLastOrder	Mean: 4.049614 SD: 3.64091	Mean: 3.992183 SD: 3.547891	Mean: 4.176847 SD: 3.659903

HourSpendOnApp	Mean: 2.946733 SD: 0.714771	Mean: 2.949453 SD: 0.69637	Mean: 2.917789 SD: 0.711339
OrderAmountHikeFromlastYear	Mean: 15.62075 SD: 3.666338	Mean: 15.59875 SD: 3.602588	Mean: 15.62256 SD: 3.603694
OrderCount	Mean: 2.965854 SD: 2.911646	Mean: 2.928609 SD: 2.860979	Mean: 3.048725 SD: 2.946886
Tenure	Mean: 7.625763 SD: 8.167407	Mean: 7.464304 SD: 7.937297	Mean: 7.300242 SD: 8.045827
WarehouseToHome	Mean: 16.12396 SD: 8.780523	Mean: 16 SD: 8.534845	Mean: 16.08424 SD: 8.523746

Based on the comparison table, the imputation for variables OrderAmountHikeFromlastYear and WarehouseToHome result in the mean and standard deviation closest to original dataset and should be imputed using the decision tree surrogate.The imputation setting was then changed with the default being median and the two variables manually set as tree surrogate imputation. .

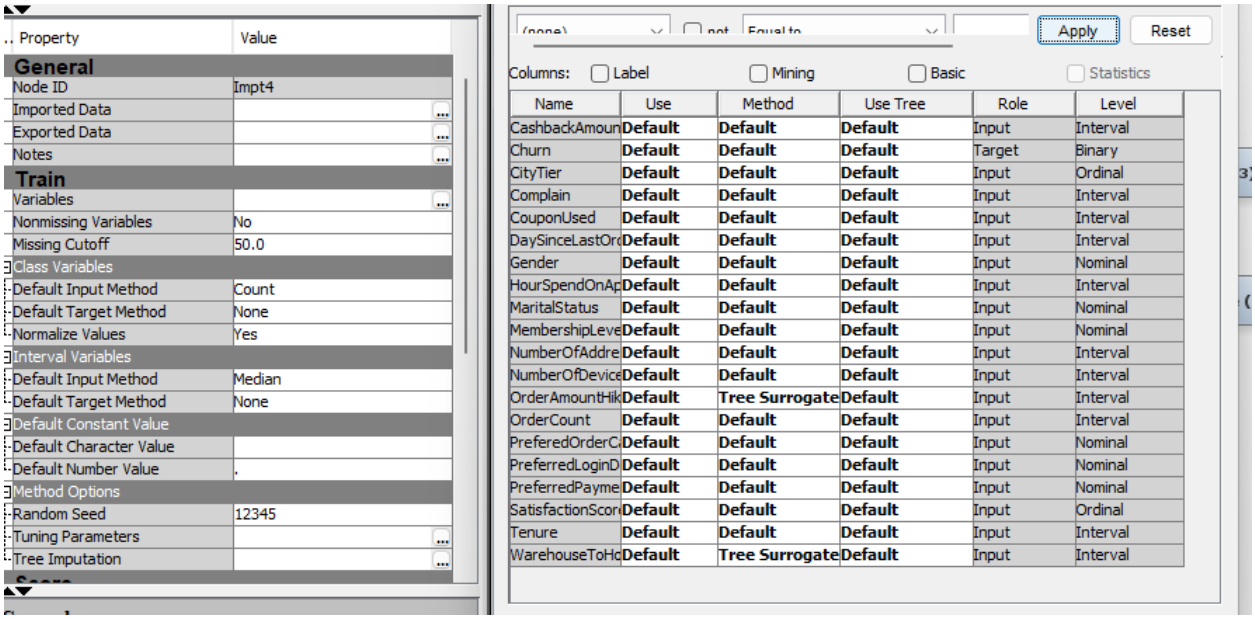

Figure 22: Final Imputation Settings

Interval Variable Summary Statistics
(maximum 500 observations printed)

Data Role=TRAIN

Variable	Role	Mean	Standard Deviation	Non Missing	Missing	Minimum	Median	Maximum	Skewness	Kurtosis
CashbackAmount	INPUT	171.2684	45.7321	1919	0	0	157.11	324.99	1.342415	1.653398
Complain	INPUT	0.384575	0.486622	1919	0	0	0	1	0.474887	-1.77633
IMP_CouponUsed	INPUT	1.75925	1.925187	1919	0	0	1	16	2.565279	9.32804
IMP_DaySinceLastOrder	INPUT	3.992183	3.547891	1919	0	0	3	46	1.742904	10.28089
IMP_HourSpendOnApp	INPUT	2.949453	0.69637	1919	0	0	3	5	0.012811	-0.63293
IMP_OrderAmountHikeFromlastYear	INPUT	15.62256	3.603694	1919	0	11	15	26	0.874548	-0.04883
IMP_OrderCount	INPUT	2.928609	2.860979	1919	0	1	2	16	2.349231	5.68359
IMP_Tenure	INPUT	7.464304	7.937297	1919	0	0	5	61	1.163516	1.136186
IMP_WarehouseToHome	INPUT	16.08424	8.523746	1919	0	5	14.86595	126	1.810043	13.61262
NumberOfAddress	INPUT	4.343408	2.666821	1919	0	1	3	20	1.010593	0.609275
NumberOfDeviceRegistered	INPUT	3.760813	1.04318	1919	0	1	4	6	-0.34897	0.537319

Figure 23: Imputation Result

In summary, after imputation, the interval variables generally exhibit modest changes in their summary statistics. The central tendency values, such as mean and median, show slight adjustments for variables like DaySinceLastOrder, OrderCount, and Tenure. However, these changes are relatively small, suggesting that the imputation process did not significantly impact the overall distribution or central tendency of the variables. Overall, the imputation process seems to have maintained the integrity of the data, with the imputed variables remaining consistent with their original distributions.

Feature Selection

Considering the Chi-Square Analysis and Variable Worth Chart and for the sake of reducing noise, the variable Gender was dropped from the dataset using the Drop Node. The setting for the Drop Node is shown in

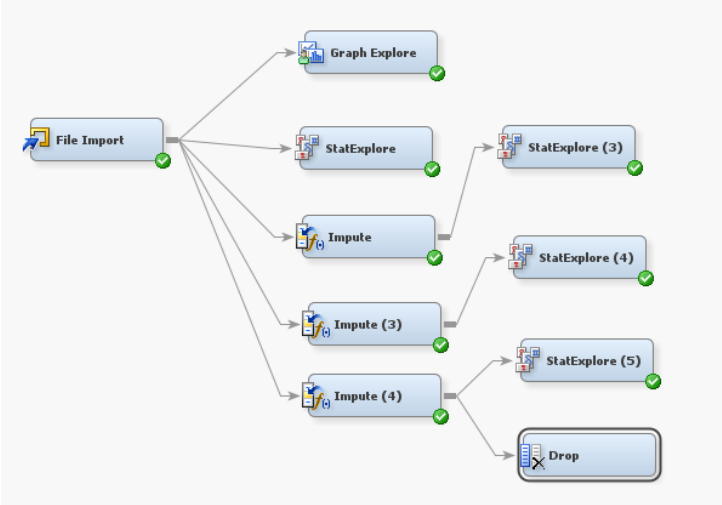

Figure 24: Addition of Drop Node

Modeling

With the amount of preparation done to the data, such as fixing the imbalance issue and dropping the irrelevant variable, this dataset should be ready for the modeling process.

Data Partition.

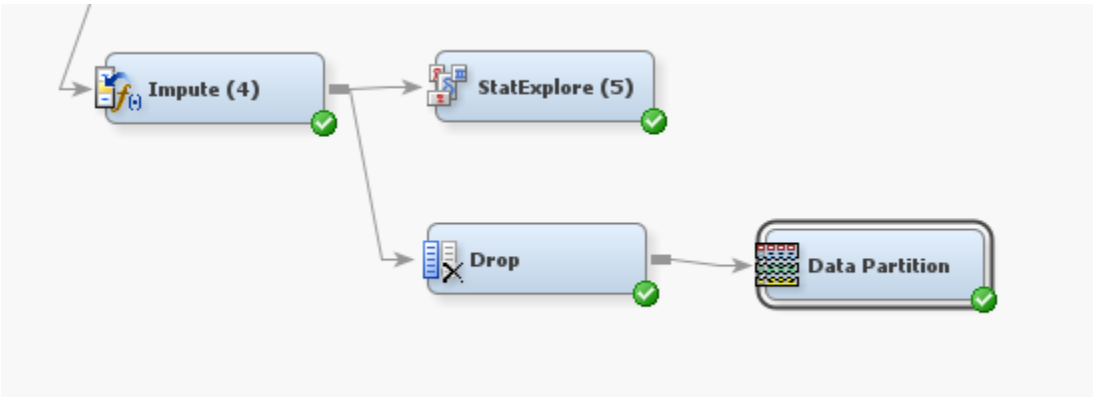

Figure 25: Addition of Data Partition Node

Property	Value
General	
Node ID	Part
Imported Data	
Exported Data	
Notes	
Train	
Variables	
Output Type	Data
Partitioning Method	Default
Random Seed	12345
Data Set Allocations	
Training	70.0
Validation	30.0
Test	0.0
Report	
Interval Targets	Yes
Class Targets	Yes
Status	
Create Time	1/7/24 5:28 PM
Run ID	c378bd89-a199-47d8-ab06-005
Last Error	
Last Status	Complete
Last Run Time	1/7/24 7:06 PM
Run Duration	0 Hr. 0 Min. 3.65 Sec.

Figure 26: Train-Validation Split Setting

Summary Statistics for Class Targets

Data=DATA

Variable	Numeric Value	Formatted Value	Frequency Count	Percent	Label
Churn	.	0	971	50.5993	
Churn	.	1	948	49.4007	

Data=TRAIN

Variable	Numeric Value	Formatted Value	Frequency Count	Percent	Label
Churn	.	0	679	50.5961	
Churn	.	1	663	49.4039	

Data=VALIDATE

Variable	Numeric Value	Formatted Value	Frequency Count	Percent	Label
Churn	.	0	292	50.6066	
Churn	.	1	285	49.3934	

Figure 27: Summary Statistics of Target Variable in Both Sets.

The dataset was first split into a training and validation set with the setting being a 70-30 ratio, which is shown in figure 26. The result of the data partition showed that training has 1342 observations and testing has a total of 577 observations. Figure 27 shows that the number of target variables in both the training and testing set are similar and the dataset is balanced.

Configuration of Models

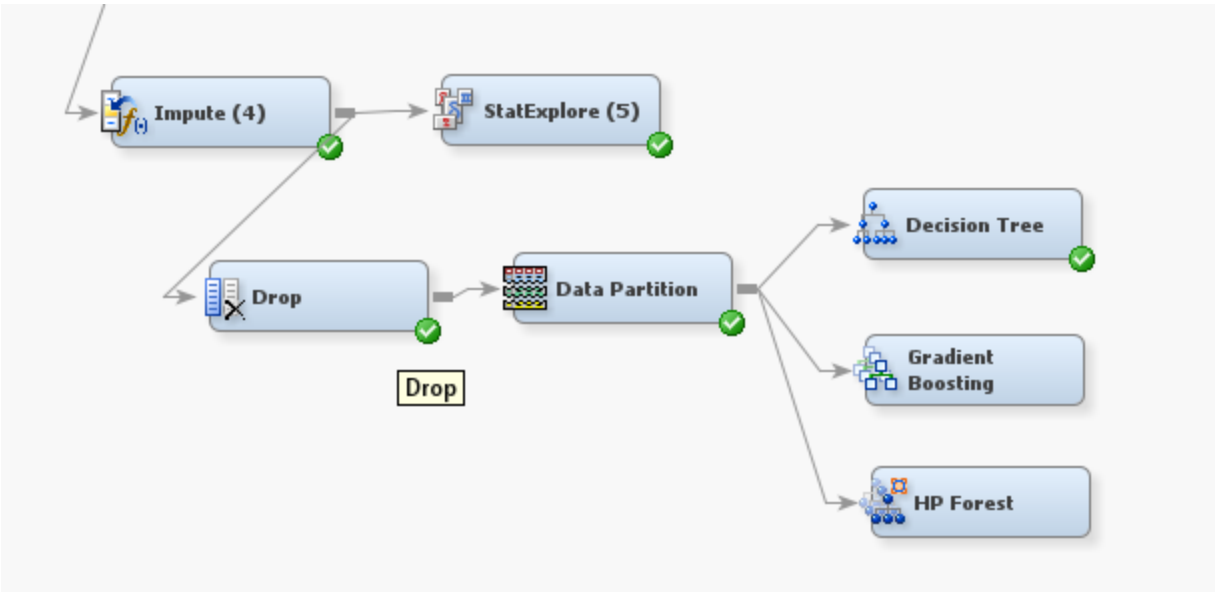

Figure 28: Addition of Model Nodes to Diagram

The three nodes used for modeling are the Decision Tree, Gradient Boosting, and HP Forest. The configuration of each model will be explained in the following sections.

Decision Tree Configuration.

Property	Value
Minimum Categorical Size	5
Node	
Leaf Size	8
Number of Rules	5
Number of Surrogate Rules	0
Split Size	.
Split Search	
Use Decisions	No
Use Priors	No
Exhaustive	5000
Node Sample	20000
Subtree	
Method	Assessment
Number of Leaves	1
Assessment Measure	Decision
Assessment Fraction	0.25
Cross Validation	
Perform Cross Validation	Yes
Number of Subsets	10
Number of Repeats	1
Seed	12345
Observation Based Importance	
Observation Based Importance	No
Number Single Var Importance	5

Figure 28: Setting of First Decision Tree

Property	Value
General	
Node ID	Boost
Imported Data	
Exported Data	
Notes	
Train	
Variables	
Series Options	
N Iterations	50
Seed	12345
Shrinkage	0.1
Train Proportion	60
Splitting Rule	
Huber M-Regression	No
Maximum Branch	2
Maximum Depth	10
Minimum Categorical Size	5
Reuse Variable	1
Categorical Bins	30
Interval Bins	100
Missing Values	Use in search
Performance	Disk
Node	
Leaf Fraction	0.001

Figure 29: Setting of Gradient Boosting

Property	Value
General	
Node ID	HPDMForest
Imported Data	
Exported Data	
Notes	
Train	
Variables	
Tree Options	
Maximum Number of Trees	100
Seed	12345
Type of Sample	Proportion
Proportion of Obs in Each Sample	0.6
Number of Obs in Each Sample	.
Splitting Rule Options	
Maximum Depth	50
Missing Values	Use In Search
Minimum Use In Search	1
Number of Variables to Consider	.
Significance Level	0.05
Max Categories in Split Search	30
Minimum Category Size	5
Exhaustive	5000
Node Options	
Method for Leaf Size	Default

Figure 30: Setting of HP Random Forest.

The input data will be modeled using different decision trees. Firstly, a full decision tree will be trained and configured to have a maximum depth property of 10 and a leaf size of 8. A maximum depth of 10 allows the decision tree to have a reasonably complex structure, capturing intricate patterns in the data. There is little risk of overfitting as the dataset is large enough. The leaf size of 8 would add regularization and prevent the model from being too sensitive to noise in the training data as the training data has 20 features. Gradient Boosting was set to have a maximum depth of 10. With Gradient Boosting often combining weak learners, this should also capture a more complex pattern than the default depth of 2 and should

be less prone to overfitting than the decision tree. The HP Random Forest setting was kept the same since it is more robust to overfitting as compared to decision trees and this should be a good starting point for a prediction of customer churn.

Assess.

Decision Tree Results.

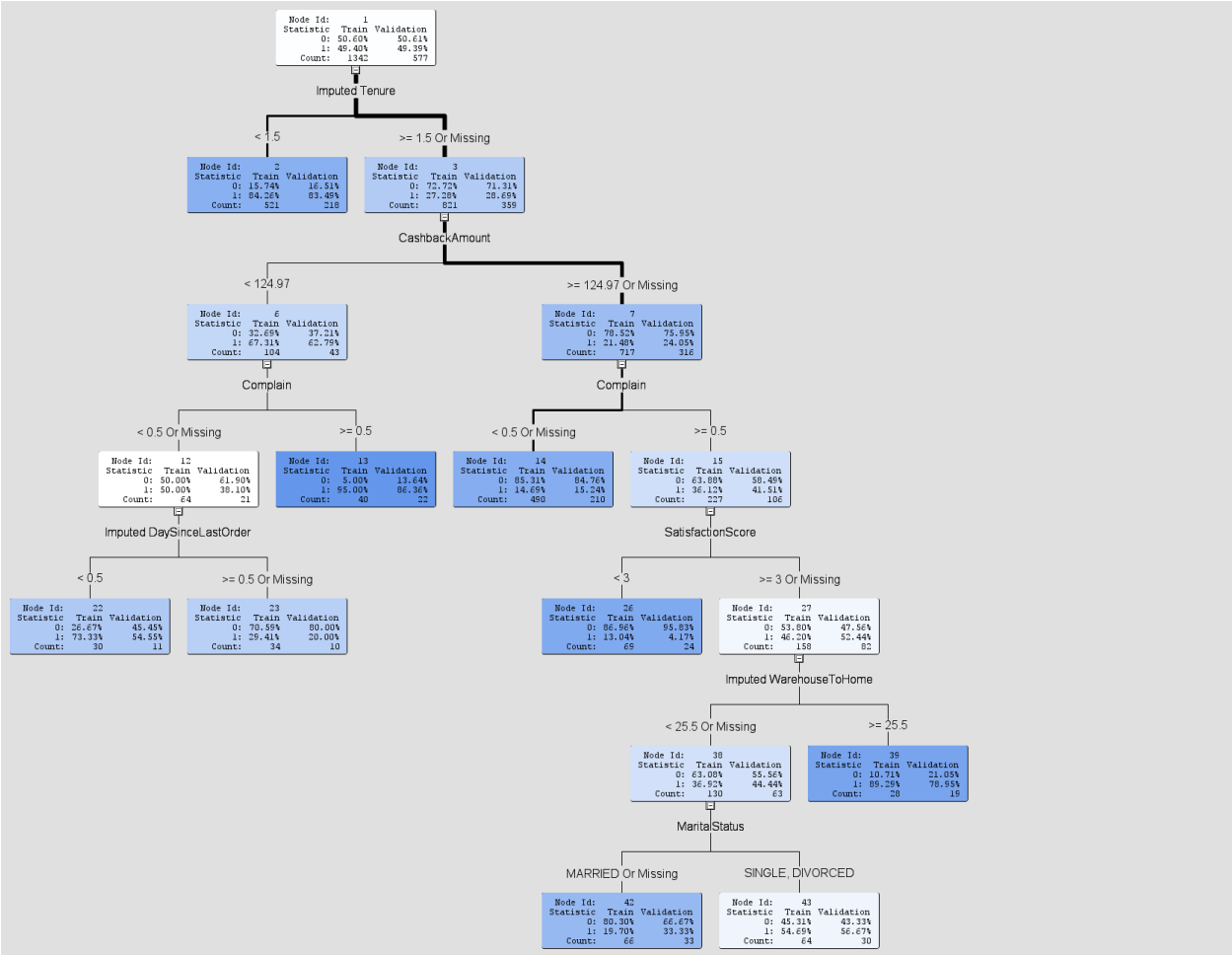

Figure 31: Decision Tree Map

Classification Table

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

Target	Outcome	Target Percentage	Outcome Percentage	Frequency Count	Total Percentage
0	0	84.2185	81.7378	555	41.3562
1	0	15.7815	15.6863	104	7.7496
0	1	18.1552	18.2622	124	9.2399
1	1	81.8448	84.3137	559	41.6542

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

Target	Outcome	Target Percentage	Outcome Percentage	Frequency Count	Total Percentage
0	0	83.3935	79.1096	231	40.0347
1	0	16.6065	16.1404	46	7.9723
0	1	20.3333	20.8904	61	10.5719
1	1	79.6667	83.8596	239	41.4211

Figure 32: Classification Table of Decision Tree Result.

Table 2: Confusion Matrix of Training Data		
	1	0
1	559	104
0	124	555

Table 3: Confusion Matrix of Validate Data		
	1	0
1	239	46
0	61	231

For the Training Dataset, the overall performance of the decision tree shows that the majority class (Churn = 0) has a higher frequency and the model performs well in predicting this class.For the non-churn class (Churn = 0), the model correctly predicted 555 instances. The

accuracy percentage is 81.74%, indicating that the model correctly identified 81.74% of non-churn cases. The model also correctly identified 84.31% of the Churn = 1 instances. Table 2 shows the confusion matrix of the training data for this decision tree.

For the validation dataset, the model correctly predicted 231 instances, indicating that the model correctly identified 79.11% of non-churn cases. For the churn class (Churn = 1), the model correctly predicted 239 instances, which is 83.85% accuracy.

Comparing the confusion matrices, we can observe how the model's predictions differ between the training and validation datasets. Both matrices show similar patterns, with higher accuracy in predicting non-churn instances compared to churn instances.There isn't any significant differences in performance metrics between the training and validation data, which means that there is probably no sign of overfitting.

Gradient Boosting Results.

Classification Table

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

Target	Outcome	Target Percentage	Outcome Percentage	Frequency Count	Total Percentage
0	0	100	100	679	50.5961
1	1	100	100	663	49.4039

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

Target	Outcome	Target Percentage	Outcome Percentage	Frequency Count	Total Percentage
0	0	91.1565	91.7808	268	46.4471
1	0	8.8435	9.1228	26	4.5061
0	1	8.4806	8.2192	24	4.1594
1	1	91.5194	90.8772	259	44.8873

Figure 33: Classification Tables of Gradient Boosting

Table 4: Confusion Matrix For Training Data		
	1	0
1	663	0
0	0	679

Table 5: Confusion Matrix For Validation Dataset		
	1	0
1	259	26
0	24	268

For the training dataset, Gradient Boosting model correctly predicted all instances for both Churn = 0 and Churn = 1. There is perfect accuracy on the training data and the instances are classified correctly for both non-churn and churn cases. Table 3 shows the confusion matrix of this model, which is perfect accuracy. This is reflected in the 100% Outcome percentage as shown in figure 33.

However, in the validation dataset, the model achieved 91.16% accuracy and 8.84% false positives for prediction of Churn = 0. For the prediction of Churn = 1, it achieved 91.52% accuracy and 8.22% false negatives. These differences between the training and validation performance suggest that the model may be overfitting the training data, as it does not generalize well to new, unseen data. In this case, the perfect accuracy on the training set is not reflected in the validation set, indicating a potential overfitting issue. Regularization techniques, adjusting model complexity, or obtaining more diverse data could help address overfitting.

HP Forest Result

Classification Table

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

Target	Outcome	Target Percentage	Outcome Percentage	Frequency Count	Total Percentage
0	0	81.1295	86.7452	589	43.8897
1	0	18.8705	20.6637	137	10.2086
0	1	14.6104	13.2548	90	6.7064
1	1	85.3896	79.3363	526	39.1952

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

Target	Outcome	Target Percentage	Outcome Percentage	Frequency Count	Total Percentage
0	0	79.4212	84.5890	247	42.8076
1	0	20.5788	22.4561	64	11.0919
0	1	16.9173	15.4110	45	7.7990
1	1	83.0827	77.5439	221	38.3016

Figure 34: Classification Tables of HP Random Forest

Table 6: Confusion Matrix For Training Dataset		
	1	0
1	526	137
0	90	589

Table 7: Confusion Matrix for Validation Dataset		
	1	0
1	221	64
0	45	247

In the training set, HP Random Forest achieved 81.3% accuracy for prediction of Churn = 0 and 85.39% accuracy for prediction of Churn = 1. For the validation set, it achieved 79.42% for prediction of Churn = 0 and 83.08 for prediction of Churn = 1. Both the training and validation sets are showing false positives and false negatives. In the validation set, the false positive rate is higher (20.58%), which indicates a potential issue with overestimating non-churn cases. On the other hand, for the occurrences of false negatives, the false negative rate is higher in the training set, which means that the model is more likely to misclassify instances of Churn when it equals 1. In the context of predicting churn, false negatives represent instances where the model fails to identify customers who are likely to churn, leading to missed opportunities for intervention or targeted retention strategies.

Comparison of Model Performance

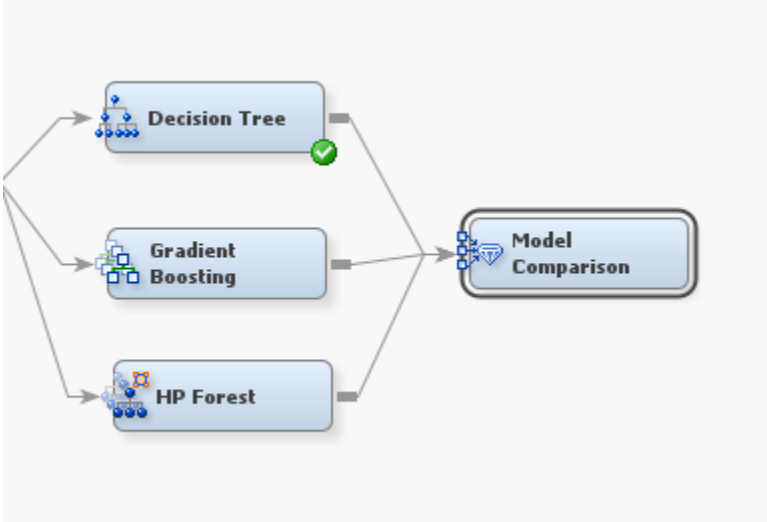

Figure 35: Addition of Model Comparison Node

To assess and compare the models, a model comparison node was added to the diagram.

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	Boost	Gradient Boosting	0.08666	0.00336	0.00000	0.07459
	Tree	Decision Tree	0.18544	0.13573	0.16990	0.14661
	HPDMForest	HP Forest	0.18891	0.12096	0.16915	0.13986

Figure 36: Fit Statistics of Three Models

The fit statistics for model comparison reveal differences in the performance of three models on the training data. For the Gradient Boosting model, it exhibits very low misclassification rates on the training set and slightly higher but still acceptable rates on the validation set, suggesting good generalization. The Decision Tree and HP Forest models show higher misclassification rates on both training and validation sets, indicating that they may not perform as well as the Gradient Boosting model in terms of classification accuracy.

The Gradient Boosting model stands out as the best model among the three models, as shown by its lower misclassification rates and squared errors on both the training and validation sets in figure 36. In the validation phase, Gradient Boosting achieved a remarkably low misclassification rate of 0.08666, coupled with a minimal squared error of 0.00336, showcasing its ability to still have a higher accuracy in predicting outcomes. In contrast, the Decision Tree and HP Forest models exhibited higher misclassification rates and squared errors, indicating a lesser degree of precision and potentially overfitting to the training data. The Gradient Boosting model should be the preferred model for predictive accuracy.

Challenges and Recommendations.

One challenge encountered in the training process was the necessity for meticulous feature selection. During the feature selection process, only one variable could be dropped due to uncertainty about which variables are truly important for this prediction. With a dataset containing 20 features, the task of identifying which variables are directly related to the target variable becomes intricate, especially without comprehensive domain knowledge. Even if a variable worth chart was generated, there remains uncertainty regarding whether the chart accurately reflects the most pertinent variables associated with the target variable.

For future work, a more comprehensive exploration of customer behavior analytics can significantly enhance feature selection processes. Exploring clustering techniques helps uncover unique groups of customers with specific preferences. By tailoring the model to these distinct segments, we enhance its ability to capture varied behaviors. This personalized approach ensures the model remains sensitive to different customer dynamics, leading to better predictions. The improvement of models can be achieved by using clustering to make the feature selection process more granular and by going through a trial-and-error process of tuning hyperparameters, such as the learning rate, tree depth, and the number of boosting rounds.

Additionally, employing cross-validation can guide the selection of optimal hyperparameters, ensuring the model generalizes well to unseen data. This iterative and exploratory approach to model refinement holds the potential to uncover hidden patterns and enhance the overall predictive performance of the system.