UNIVERSITI
MALAYA

Faculty of Computer Science & Information Technology

Master of Data Science
Semester I Session 2023/2024

WQD7005 Data Mining

Alternative Assessment 1
Title: E-Commerce Customer Behaviour Analysis

GitHub: https://github.com/emilyxs1105/WQD7005-Data-Mining-AA1

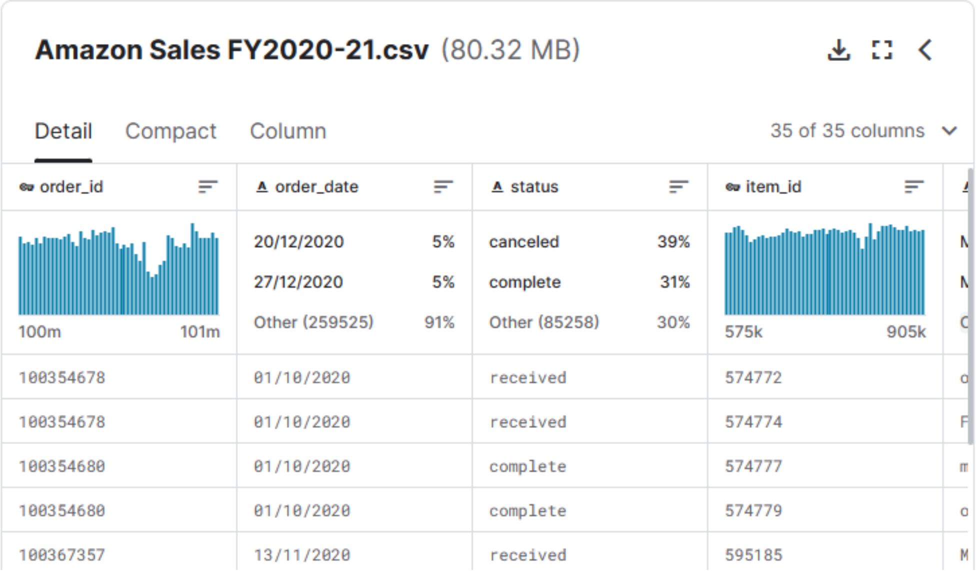
Prepared by
Emily Sia Zi Xuan (17205326)

Lecturer: Prof. Dr. Teh Ying Wah

Contents

Introduction

The dataset utilized in this case study comprises 1021 records, encompassing 12 attributes related to customer transactions sourced from the Amazon website. The original dataset, available on Kaggle (https://www.kaggle.com/datasets/earthfromtop/amazon-sales-fy202021/data), initially contains a total of 35 columns. For the purpose of this study, a subset of 1021 records were selected, focusing on specific columns including 'cust_id,' 'age,' 'Gender,' 'State,' 'total,' 'category,' 'order_date,' 'payment_method,' and 'full_name.' To align with the specified dataset structure, modifications and aggregations were applied to the obtained dataset.



Amazon Sales FY2020-21.csv (80.32 MB)

Detail Compact Column 35 of 35 columns

order_id	order_date	status	item_id
100m 101m	20/12/2020 5% 27/12/2020 5% Other (259525) 91%	canceled 39% complete 31% Other (85258) 30%	575k 905k
100354678	01/10/2020	received	574772
100354678	01/10/2020	received	574774
100354680	01/10/2020	complete	574777
100354680	01/10/2020	complete	574779
100367357	13/11/2020	received	595185

The column 'TotalSpent' was computed by aggregating the total transaction amount for each customer based on their transaction records. Similarly, 'TotalPurchases' represents the sum of the number of transactions made by a customer. 'PreferredPayment' indicates the most frequently used payment method by a customer, while 'LatestOrderDate' signifies the most recent transaction date. 'FavoriteCategory' is derived from the category most frequently ordered by a customer.

'MembershipLevel' is determined based on the 'TotalSpent' of the customer. If a customer's spending falls below the first quartile of the overall 'TotalSpent,' their membership level is categorized as standard. Bronze membership is assigned for spending beyond the first quartile, silver for surpassing the second quartile, gold for exceeding the third quartile, and platinum for expenditures surpassing 10000 US dollars.

Finally, the churn status is defined by evaluating whether a customer's 'LatestOrderDate' precedes May 2020. If the latest order date is before this threshold, the customer is considered churned. These data manipulations and aggregations were undertaken by using Python code to create a dataset structure aligning with the specified requirements for the case study.

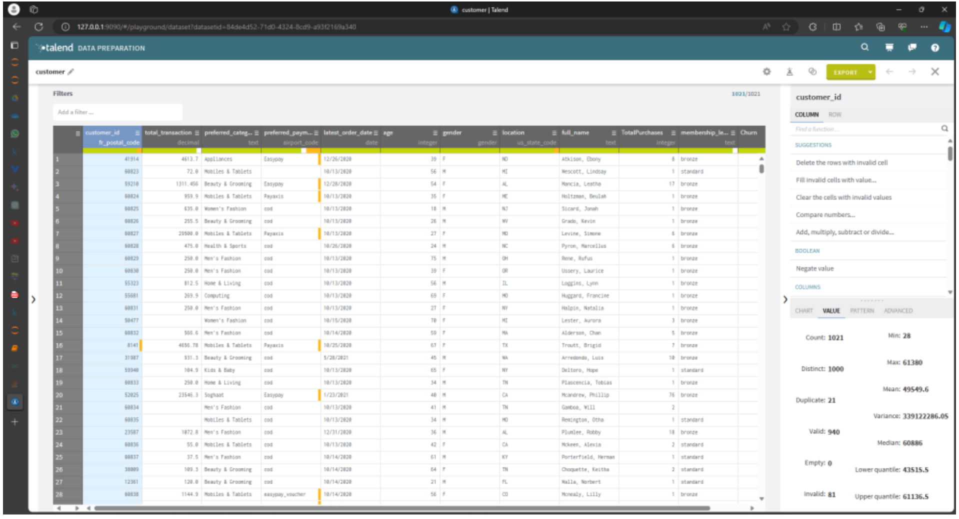
The objective of this case study is to (1) conduct comprehensive analysis of customer behavior and (2) utilize decision tree model and ensemble model for churn prediction. This objective aimed at gaining actionable insights into the diversity of customer behaviors, facilitating targeted marketing strategies, and improving personalized customer experiences. Also, the case study objective sought to enhance retention strategies and optimize loyalty programs for improved customer satisfaction and loyalty.

Data Import and Preprocessing

The dataset is imported into Talend Data Preparation software to inspect the structure of data and go through some necessary data preprocessing steps.

A. Invalid values under customer_id

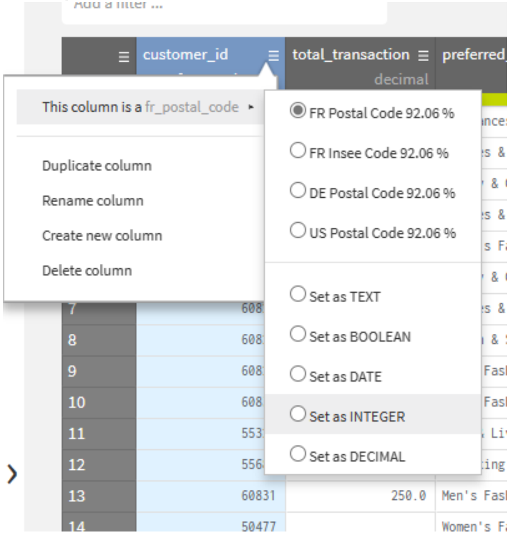
1. From the Talend Data Preparation software window, we can see that the attributes customer_id consists of 81 invalid values. This is because the datatype of the column is not correctly set.


customer | Talend
127.0.0.1:9090/#/playground/dataset?datasetid=84de4d52-71d0-4324-8cd9-a93f2169a340
talend DATA PREPARATION
customer
EXPORT
Filters
Add a filter ...
1021/1021
customer_id
COLUMN
ROW
Find a function ...
SUGGESTIONS
Delete the rows with invalid cell
Fill invalid cells with value...
Clear the cells with invalid values
Compare numbers...
Add, multiply, subtract or divide...
BOOLEAN
Negate value
COLUMNS
CHART
VALUE
PATTERN
ADVANCED

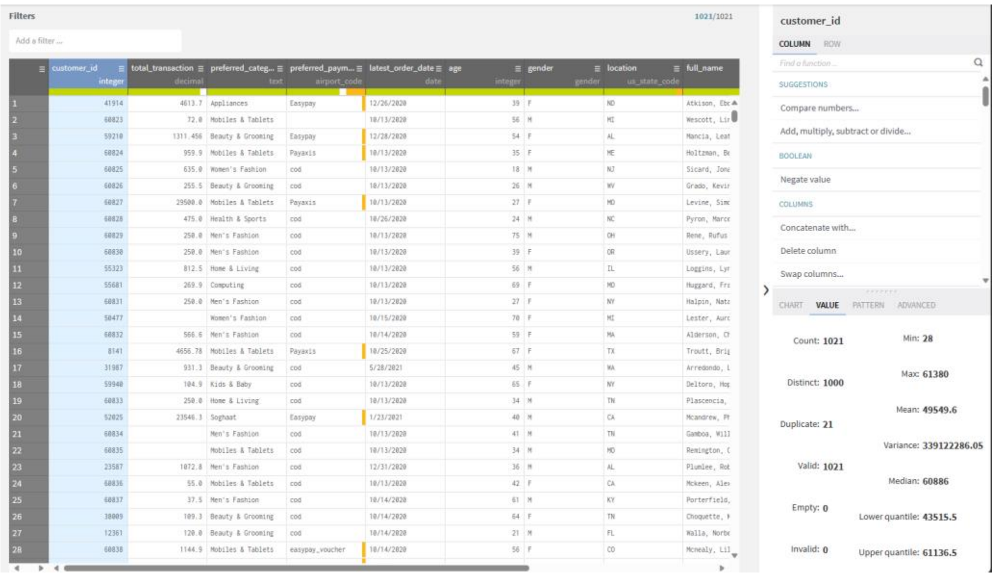
	customer_id	total_transaction	preferred_categ...	preferred_paym...	latest_order_date	age	gender	location	full_name	TotalPurchases	membership_le...	Churn
	fr_postal_code	decimal	text	airport_code	date	integer	gender	us_state_code	text	integer	text	
1	41914	4613.7	Appliances	Easypay	12/26/2020	39	F	MD	Atkison, Ebony	8	bronze	
2	60823	72.0	Mobiles & Tablets		10/13/2020	56	M	MI	Wescott, Lindsay	1	standard	
3	59210	1311.456	Beauty & Grooming	Easypay	12/28/2020	54	F	AL	Mancia, Leatha	17	bronze	
4	60824	959.9	Mobiles & Tablets	Payaxis	10/13/2020	35	F	ME	Holtzman, Beulah	1	bronze	
5	60825	635.0	Women's Fashion	cod	10/13/2020	18	M	NJ	Sicard, Jonah	1	bronze	
6	60826	255.5	Beauty & Grooming	cod	10/13/2020	26	M	WV	Grado, Kevin	1	bronze	
7	60827	29500.0	Mobiles & Tablets	Payaxis	10/13/2020	27	F	MD	Levine, Simone	6	bronze	
8	60828	475.0	Health & Sports	cod	10/26/2020	24	M	NC	Pyron, Marcellus	6	bronze	
9	60829	250.0	Men's Fashion	cod	10/13/2020	75	M	OH	Rene, Rufus	1	bronze	
10	60830	250.0	Men's Fashion	cod	10/13/2020	39	F	OR	Ussery, Laurice	1	bronze	
11	55323	812.5	Home & Living	cod	10/13/2020	56	M	IL	Loggins, Lynn	1	bronze	
12	55681	269.9	Computing	cod	10/13/2020	69	F	MO	Huggard, Francine	1	bronze	
13	60831	250.0	Men's Fashion	cod	10/13/2020	27	F	NY	Halpin, Natalia	1	bronze	
14	50477		Women's Fashion	cod	10/15/2020	70	F	MI	Lester, Aurora	3	bronze	
15	60832	566.6	Men's Fashion	cod	10/14/2020	59	F	MA	Alderson, Chan	5	bronze	
16	8141	4656.78	Mobiles & Tablets	Payaxis	10/25/2020	67	F	TX	Troutt, Brigid	7	bronze	
17	31987	931.3	Beauty & Grooming	cod	5/28/2021	45	M	WA	Arredondo, Luis	10	bronze	
18	59940	104.9	Kids & Baby	cod	10/13/2020	65	F	NY	Deltoro, Hope	1	standard	
19	60833	250.0	Home & Living	cod	10/13/2020	34	M	TN	Plascencia, Tobias	1	bronze	
20	52025	23546.3	Soghaat	Easypay	1/23/2021	40	M	CA	Mcandrew, Phillip	76	bronze	
21	60834		Men's Fashion	cod	10/13/2020	41	M	TN	Gamboa, Will	2		
22	60835		Mobiles & Tablets	cod	10/13/2020	34	M	MO	Remington, Otha	1	standard	
23	23587	1072.8	Men's Fashion	cod	12/31/2020	36	M	AL	Plumlee, Robby	18	bronze	
24	60836	55.0	Mobiles & Tablets	cod	10/13/2020	42	F	CA	Mckeen, Alexia	2	standard	
25	60837	37.5	Men's Fashion	cod	10/14/2020	61	M	KY	Porterfield, Herman	1	standard	
26	38009	109.3	Beauty & Grooming	cod	10/14/2020	64	F	TN	Choquette, Keitha	2	standard	
27	12361	120.0	Beauty & Grooming	cod	10/14/2020	21	M	FL	Walla, Norbert	1	standard	
28	60838	1144.9	Mobiles & Tablets	easypay_voucher	10/14/2020	56	F	CO	Mcnealy, Lilly	1	bronze	

Count: 1021	Min: 28
Distinct: 1000	Max: 61380
Duplicate: 21	Mean: 49549.6
	Variance: 339122286.05
Valid: 940	Median: 60886
Empty: 0	Lower quantile: 43515.5
Invalid: 81	Upper quantile: 61136.5

2. Click on the Hamburger icon beside the customer_id and change the datatype of this column to INTEGER will helps to remove the invalid value warning.



3. Now, the number of invalid values for this column become zero.



Filters

Add a filter ...

1021/1021

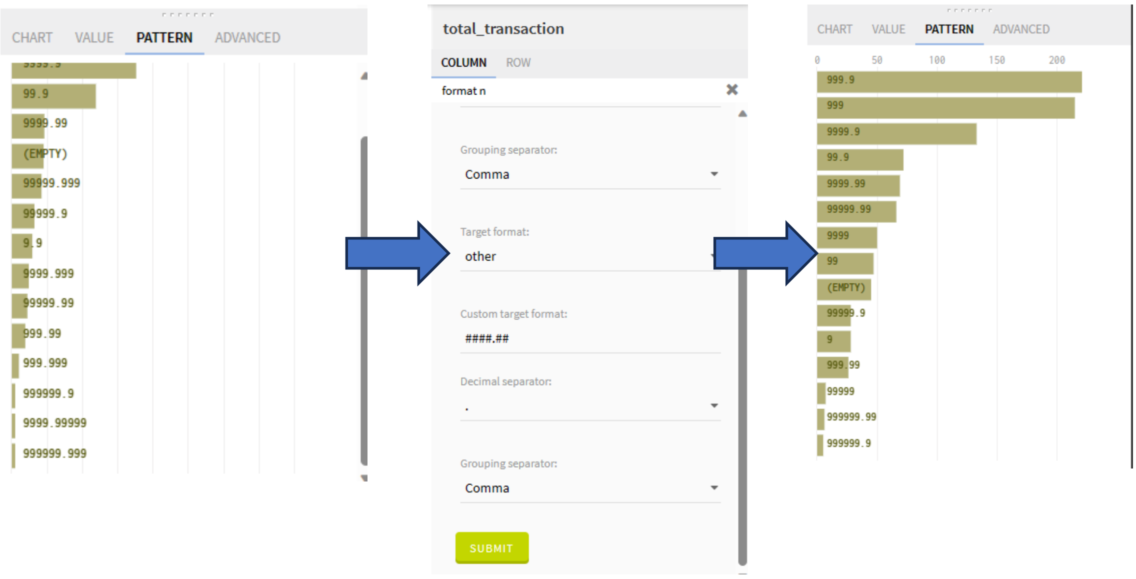
	customer_id	total_transaction	preferred_categ...	preferred_paym...	latest_order_date	age	gender	location	full_name
	integer	decimal	text	airport_code	date	integer	gender	us_state_code	
1	41914	4613.7	Appliances	Easypay	12/26/2020	39	F	ND	Atkison, Ebc
2	60823	72.0	Mobiles & Tablets		10/13/2020	56	M	MI	Wescott, Lir
3	59210	1311.456	Beauty & Grooming	Easypay	12/28/2020	54	F	AL	Mancia, Leat
4	60824	959.9	Mobiles & Tablets	Payaxis	10/13/2020	35	F	ME	Holtzman, Be
5	60825	635.0	Women's Fashion	cod	10/13/2020	18	M	NJ	Sicard, Jone
6	60826	255.5	Beauty & Grooming	cod	10/13/2020	26	M	WV	Grado, Kevir
7	60827	29500.0	Mobiles & Tablets	Payaxis	10/13/2020	27	F	MD	Levine, Simc
8	60828	475.0	Health & Sports	cod	10/26/2020	24	M	NC	Pyron, Marcc
9	60829	250.0	Men's Fashion	cod	10/13/2020	75	M	OH	Rene, Rufus
10	60830	250.0	Men's Fashion	cod	10/13/2020	39	F	OR	Ussery, Laur
11	55323	812.5	Home & Living	cod	10/13/2020	56	M	IL	Loggins, Lyr
12	55681	269.9	Computing	cod	10/13/2020	69	F	MD	Huggard, Fra
13	60831	250.0	Men's Fashion	cod	10/13/2020	27	F	NY	Halpin, Natz
14	50477		Women's Fashion	cod	10/15/2020	70	F	MI	Lester, Aurc
15	60832	566.6	Men's Fashion	cod	10/14/2020	59	F	MA	Alderson, Ch
16	8141	4656.78	Mobiles & Tablets	Payaxis	10/25/2020	67	F	TX	Troutt, Brig
17	31987	931.3	Beauty & Grooming	cod	5/28/2021	45	M	WA	Arredondo, L
18	59940	104.9	Kids & Baby	cod	10/13/2020	65	F	NY	Deltoro, Hop
19	60833	250.0	Home & Living	cod	10/13/2020	34	M	TN	Plascencia,
20	52025	23546.3	Soghaat	Easypay	1/23/2021	40	M	CA	Mcandrew, Ph
21	60834		Men's Fashion	cod	10/13/2020	41	M	TN	Gamboa, Will
22	60835		Mobiles & Tablets	cod	10/13/2020	34	M	MO	Remington, (
23	23587	1072.8	Men's Fashion	cod	12/31/2020	36	M	AL	Plumlee, Rot
24	60836	55.0	Mobiles & Tablets	cod	10/13/2020	42	F	CA	Mckeen, Alex
25	60837	37.5	Men's Fashion	cod	10/14/2020	61	M	KY	Porterfield,
26	38009	109.3	Beauty & Grooming	cod	10/14/2020	64	F	TN	Choquette, I
27	12361	120.0	Beauty & Grooming	cod	10/14/2020	21	M	FL	Walla, Norbe
28	60838	1144.9	Mobiles & Tablets	easypay_voucher	10/14/2020	56	F	CO	Mcnealy, Lil

customer_id

Count: 1021
Distinct: 1000
Duplicate: 21
Valid: 1021
Empty: 0
Invalid: 0
Min: 28
Max: 61380
Mean: 49549.6
Variance: 339122286.05
Median: 60886
Lower quantile: 43515.5
Upper quantile: 61136.5

B. Data transformation for total_transaction

The total_transaction values are inconsistently formatted with varying decimal places, deviating from the standard convention for displaying prices. In Talend Data Preparation window, we search for the function "Format number..." and specify the desired number format as ####.## to ensure the number is displayed with two decimal places. Then, run the process. Now the total_transaction values are formatted with 2 decimal places.



total_transaction

format n

Grouping separator: Comma

Target format: other

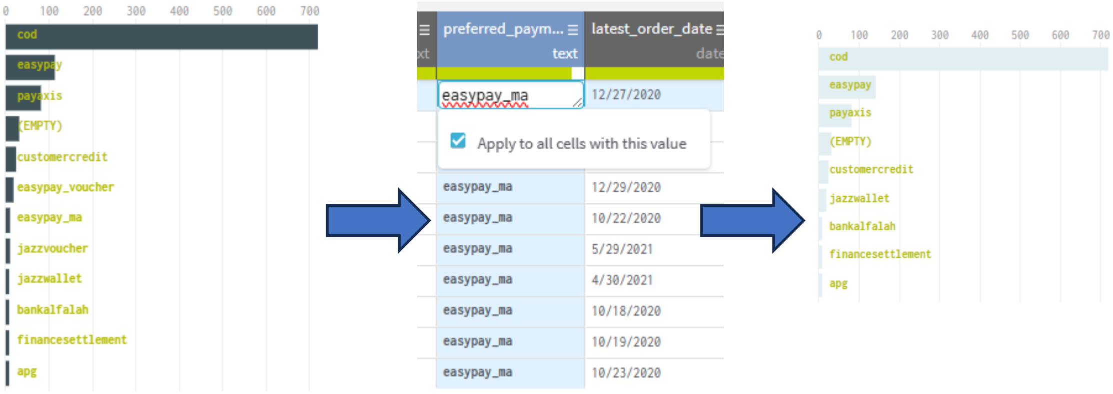
Custom target format: ####.##

Decimal separator: .

Grouping separator: Comma

C. Group similar preferred_payment_method

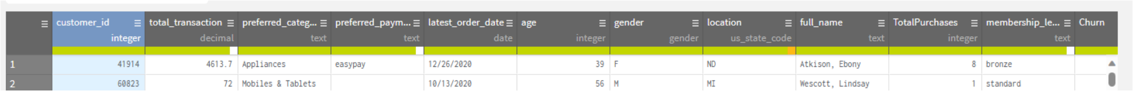
From the column "preferred_payment_method," we observe that certain payment methods can be considered equivalent and grouped into a single category. For instance, "easypay," "easypay_voucher," and "easypay_ma" can be consolidated under the category "easypay." Therefore, we replace "easypay_voucher" and "easypay_ma" with "easypay" and select the option "Apply to all cells with this value" to ensure the change is applied universally. The same process is applied to "jazzwallet" and "jazzvoucher," both of which are replaced with "jazzwallet". Now, the similar preferred_payment_method is grouped, total 8 unique values are under this column with missing value.



preferred_paym... text	latest_order_date date
easypay_ma	12/27/2020
☑ Apply to all cells with this value	
easypay_ma	12/29/2020
easypay_ma	10/22/2020
easypay_ma	5/29/2021
easypay_ma	4/30/2021
easypay_ma	10/18/2020
easypay_ma	10/19/2020
easypay_ma	10/23/2020

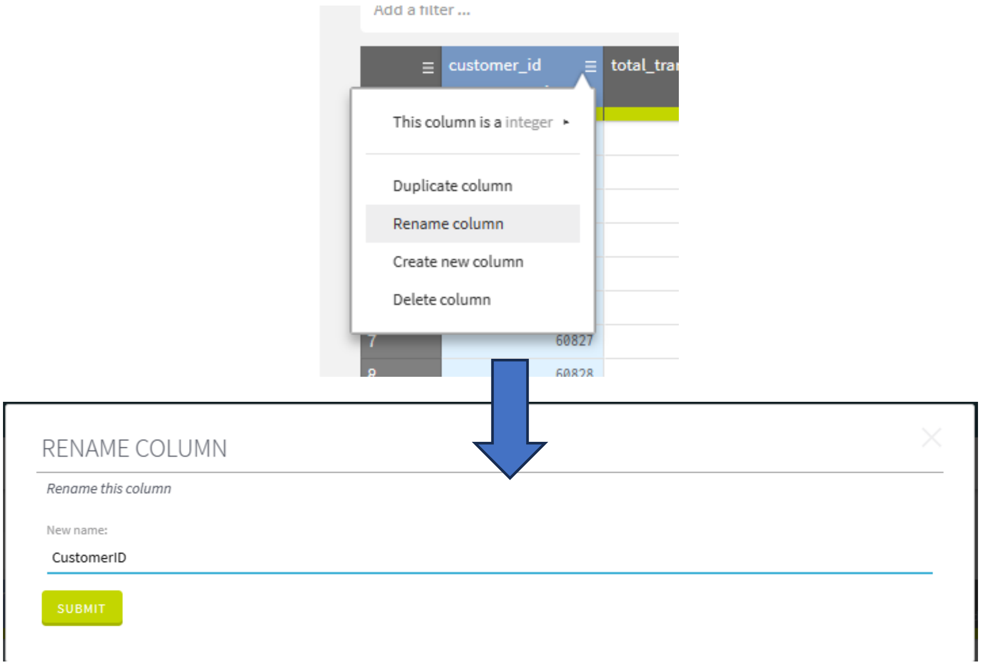
D. Rename the column to follow naming convention

1. We observed that the column names exhibit different naming styles, with some in snake case (e.g., customer_id) and others in pascal case (e.g., TotalPurchase). We can standardize the column names using Talend Data Preparation.



	customer_id integer	total_transaction decimal	preferred_categ... text	preferred_paym... text	latest_order_date date	age integer	gender gender	location us_state_code	full_name text	TotalPurchases integer	membership_le... text	Churn
1	41914	4613.7	Appliances	easypay	12/26/2020	39	F	ND	Atkison, Ebony	8	bronze	
2	60823	72	Mobiles & Tablets		10/13/2020	56	M	MI	Wescott, Lindsay	1	standard	

2. Right click on the column we want to rename, select “Rename Column”. Then, insert the new name into the pop-up text field and click submit. The column is now renamed.



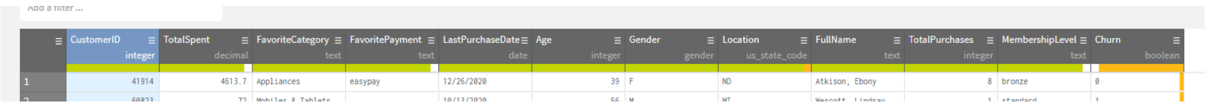
RENAME COLUMN

Rename this column

New name:

CustomerID

3. After that, we can view that the column names are renamed. In this dataset, we rename all the column name to follow pascal case. Now, the new column names are: "CustomerID","TotalSpent","FavoriteCategory","FavoritePayment","LastPurchaseDate","Age","Gender","Location","FullName","TotalPurchases","MembershipLevel" and "Churn".



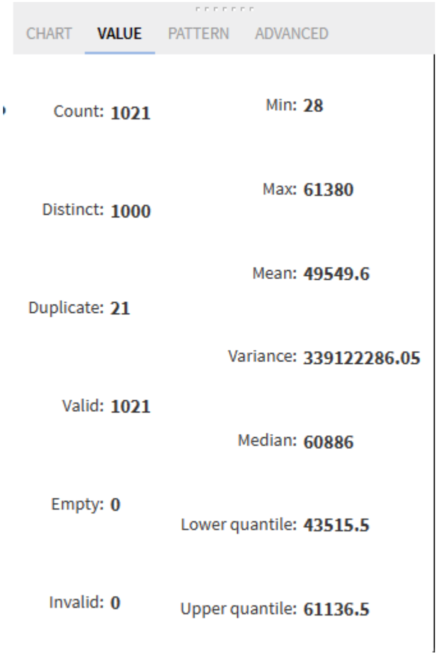
Add a filter ...

	CustomerID integer	TotalSpent decimal	FavoriteCategory text	FavoritePayment text	LastPurchaseDate date	Age integer	Gender gender	Location us_state_code	FullName text	TotalPurchases integer	MembershipLevel text	Churn boolean
1	41914	4613.7	Appliances	easypay	12/26/2020	39	F	ND	Atkison, Ebony	8	bronze	0
2	60823	72	Mobiles & Tablets		10/13/2020	56	M	MT	Wescott, Lindsay	1	standard	1

E. Handle duplicates values

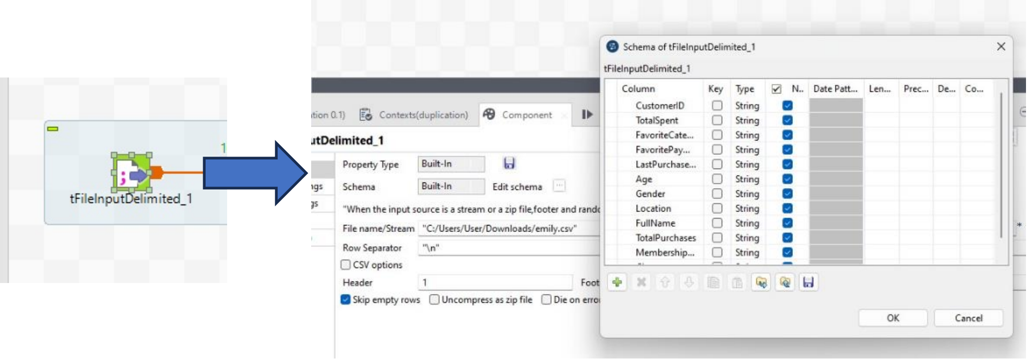
The Talend Data Preparation Desktop Free version has limited function. Hence, we continue the data preprocessing in Talend Open Studio for Data Integration.

1. The CustomerID act the unique index in this dataset, however, as shown in the Talend Data Preparation Window, there are 21 duplicates in the CustomerID columns. This means that there are 21 duplicates data in this dataset.



Count: 1021
Min: 28
Distinct: 1000
Max: 61380
Mean: 49549.6
Duplicate: 21
Variance: 339122286.05
Valid: 1021
Median: 60886
Empty: 0
Lower quantile: 43515.5
Invalid: 0
Upper quantile: 61136.5

2. First add the tFileInputDelimited node into the diagram. In the properties' component tab of the node, we type the dataset csv file path in the File name/Streams text field. Then, we click on the "…" button of the Edit schema and define the schema of the dataset.

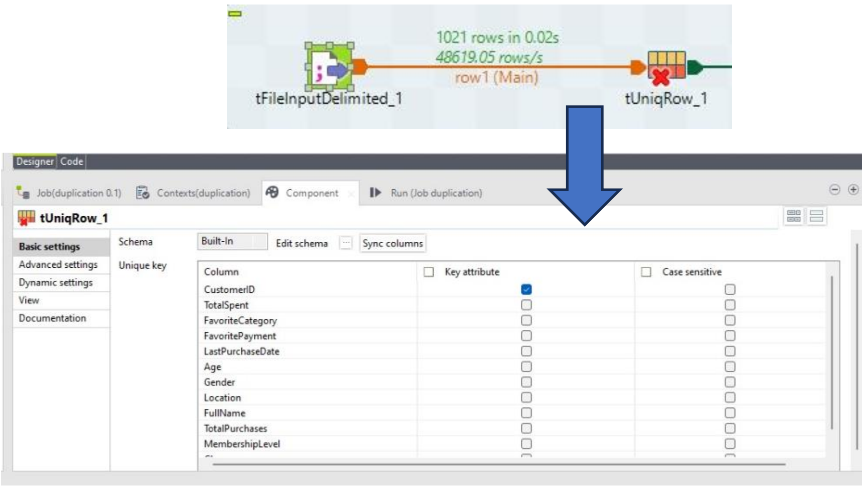


Property Type: Built-In
Schema: Built-In Edit schema ...
"When the input source is a stream or a zip file,footer and rand
File name/Stream: "C:/Users/User/Downloads/emily.csv"
Row Separator: "\n"
□ CSV options
Header: 1 Foot
☑ Skip empty rows □ Uncompress as zip file □ Die on erro

tFileInputDelimited_1

Column	Key	Type	☑ N..	Date Patt...	Len...	Prec...	De...	Co...
CustomerID	□	String	☑					
TotalSpent	□	String	☑					
FavoriteCate...	□	String	☑					
FavoritePay...	□	String	☑					
LastPurchase...	□	String	☑					
Age	□	String	☑					
Gender	□	String	☑					
Location	□	String	☑					
FullName	□	String	☑					
TotalPurchases	□	String	☑					
Membership...	□	String	☑					

3. Next, we put in the tUniqRow node and connect it with the previous node. In the properties' component tab of the node, we select the CustomerID as the key attribute as it is the unique index in this dataset. This node helps to remove the duplicates rows.



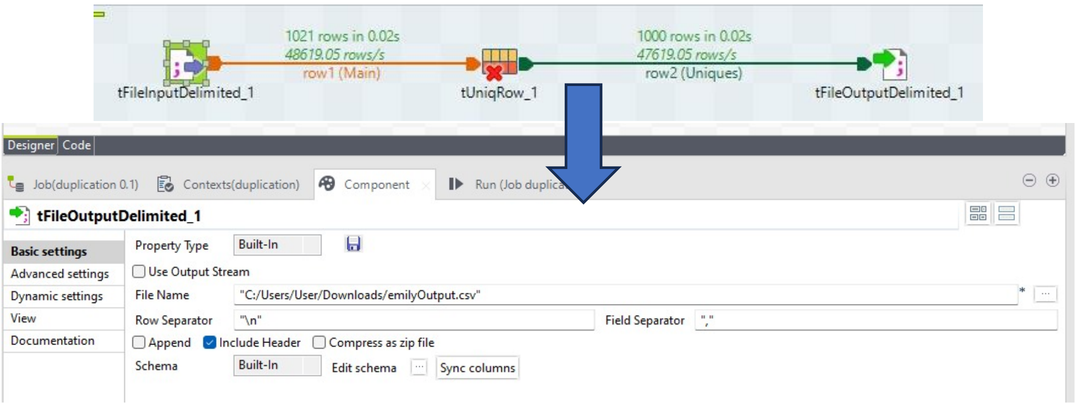
tUniqRow_1

Schema: Built-In

Unique key:

Column	Key attribute	Case sensitive
CustomerID	☑	□
TotalSpent	□	□
FavoriteCategory	□	□
FavoritePayment	□	□
LastPurchaseDate	□	□
Age	□	□
Gender	□	□
Location	□	□
FullName	□	□
TotalPurchases	□	□
MembershipLevel	□	□

4. Then, we add the tFileOutputDelimited node into the diagram. In the properties' component tab of the node, we specify the output path and edit the output file schema. Lastly, we run the job flow. An output csv file is generated in the specified path.



tFileOutputDelimited_1

Property Type: Built-In

□ Use Output Stream

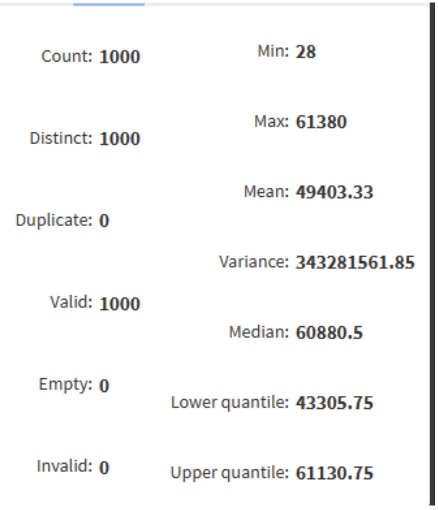
File Name: "C:/Users/User/Downloads/emilyOutput.csv"

Row Separator: "\n" Field Separator: ","

□ Append ☑ Include Header □ Compress as zip file

Schema: Built-In

5. We open the dataset in Talend Data Preparation Software and inspect the summary of the CustomerID column. Now, there are no duplicates data in the dataset.



Count: 1000 Min: 28

Distinct: 1000 Max: 61380

Mean: 49403.33

Duplicate: 0

Variance: 343281561.85

Valid: 1000

Median: 60880.5

Empty: 0

Lower quantile: 43305.75

Invalid: 0

Upper quantile: 61130.75

F. Import dataset into SAS Enterprise Miner

Next, we continue the data preprocessing in SAS Enterprise Miner.

1. In the opened project of SAS Enterprise Miner, create a new diagram called ‘CustomerBehaviorAnalysis’.

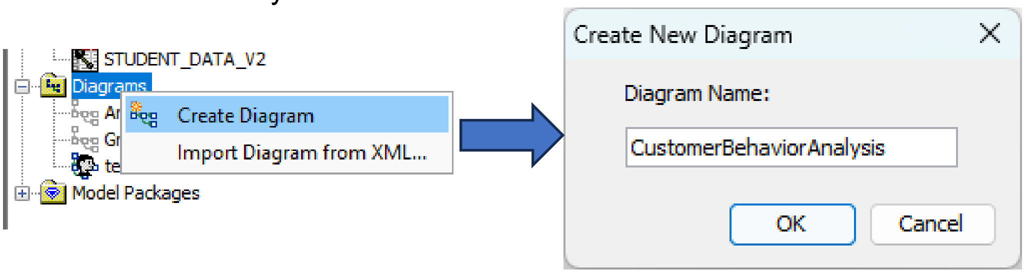
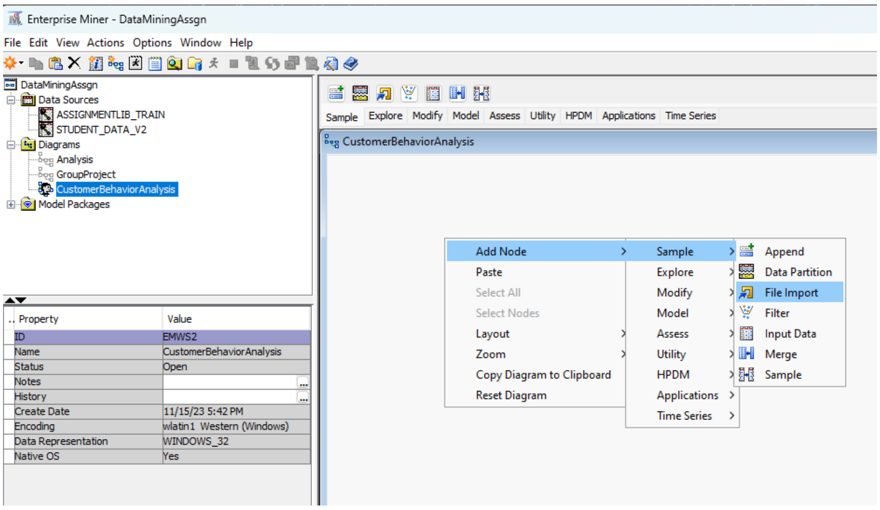


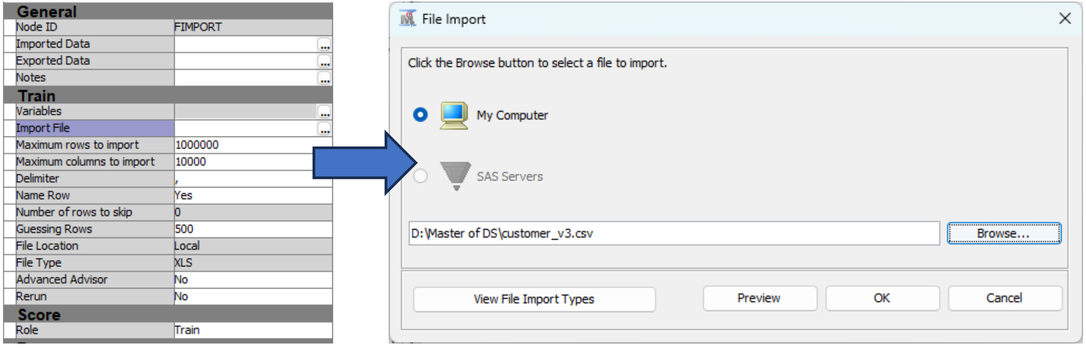
Diagram Name: CustomerBehaviorAnalysis

2. In the panel of the newly created diagram, right and select Add Node > Sample > File Import to add the File Import Node. We use the File Import Node because the dataset is in CSV file format and it considered as external file.



Property	Value
ID	EMWS2
Name	CustomerBehaviorAnalysis
Status	Open
Notes	...
History	...
Create Date	11/15/23 5:42 PM
Encoding	wlatin1 Western (Windows)
Data Representation	WINDOWS_32
Native OS	Yes

3. Click on the File Import node, in the properties of the node, click on the “…” button in Import File. An import file wizard pop-up. Select My Computer and click on “Browse…” to specify the file path. Lastly press the OK button to import the file.



General	
Node ID	FIMPORT
Imported Data	...
Exported Data	...
Notes	...
Train	
Variables	...
Import File	...
Maximum rows to import	1000000
Maximum columns to import	10000
Delimiter	,
Name Row	Yes
Number of rows to skip	0
Guessing Rows	500
File Location	Local
File Type	XLS
Advanced Advisor	No
Rerun	No
Score	
Role	Train

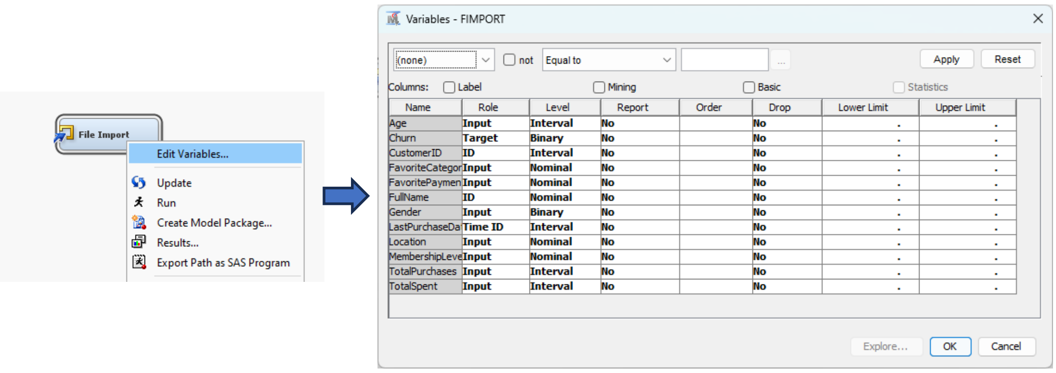
Click the Browse button to select a file to import.

● My Computer

○ SAS Servers

D:\Master of DS\customer_v3.csv

4. Right-click on the File Import node, and select "Edit Variable..." to define the variables' roles and levels. Set the CustomerID and FullName role as ID and specify LastPurchaseDate as the TimeID. In this dataset, our primary interest is in churn classification; therefore, we designate "Churn" as the target variable.



Variables - FIMPORT

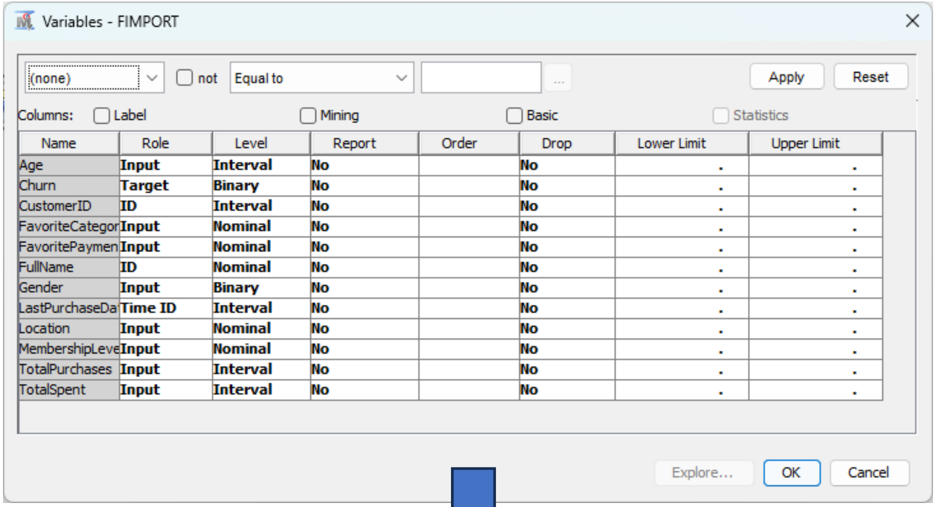
(none) □ not Equal to … Apply Reset

Columns: □ Label □ Mining □ Basic □ Statistics

Name	Role	Level	Report	Order	Drop	Lower Limit	Upper Limit
Age	Input	Interval	No		No	.	.
Churn	Target	Binary	No		No	.	.
CustomerID	ID	Interval	No		No	.	.
FavoriteCategor	Input	Nominal	No		No	.	.
FavoritePaymen	Input	Nominal	No		No	.	.
FullName	ID	Nominal	No		No	.	.
Gender	Input	Binary	No		No	.	.
LastPurchaseDa	Time ID	Interval	No		No	.	.
Location	Input	Nominal	No		No	.	.
MembershipLeve	Input	Nominal	No		No	.	.
TotalPurchases	Input	Interval	No		No	.	.
TotalSpent	Input	Interval	No		No	.	.

Explore... OK Cancel

5. In the Edit Variable window, we can also inspect the statistic of the variable. For example, we select the TotalSpent variable and click Explore…, this will pop out a explore window showing the statistic of the Total Spent variable. For interval variables, the missing values are shown as a missing bin in histogram We can see that there are missing values under this variable as shown, we should impute the missing value.



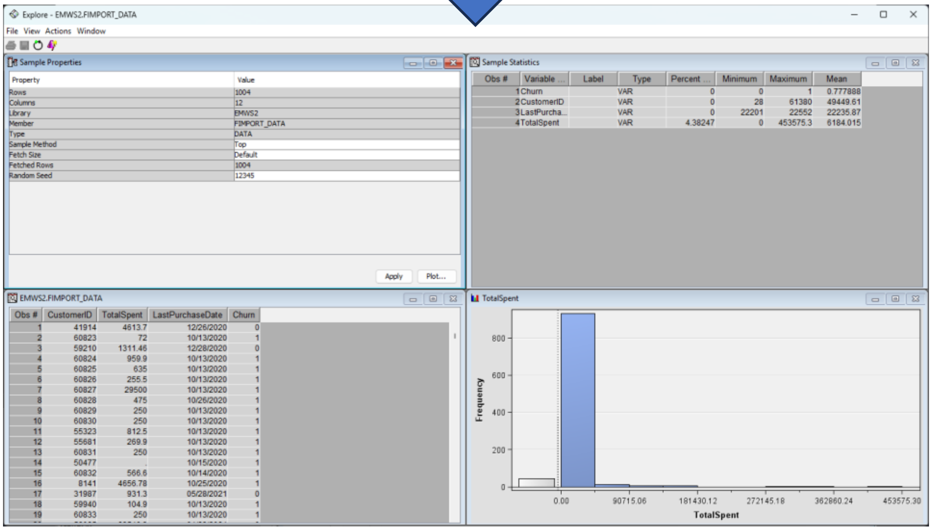
Variables - FIMPORT

(none) □ not Equal to … Apply Reset

Columns: □ Label □ Mining □ Basic □ Statistics

Name	Role	Level	Report	Order	Drop	Lower Limit	Upper Limit
Age	Input	Interval	No		No	.	.
Churn	Target	Binary	No		No	.	.
CustomerID	ID	Interval	No		No	.	.
FavoriteCategor	Input	Nominal	No		No	.	.
FavoritePaymen	Input	Nominal	No		No	.	.
FullName	ID	Nominal	No		No	.	.
Gender	Input	Binary	No		No	.	.
LastPurchaseDa	Time ID	Interval	No		No	.	.
Location	Input	Nominal	No		No	.	.
MembershipLeve	Input	Nominal	No		No	.	.
TotalPurchases	Input	Interval	No		No	.	.
TotalSpent	Input	Interval	No		No	.	.

Explore... OK Cancel

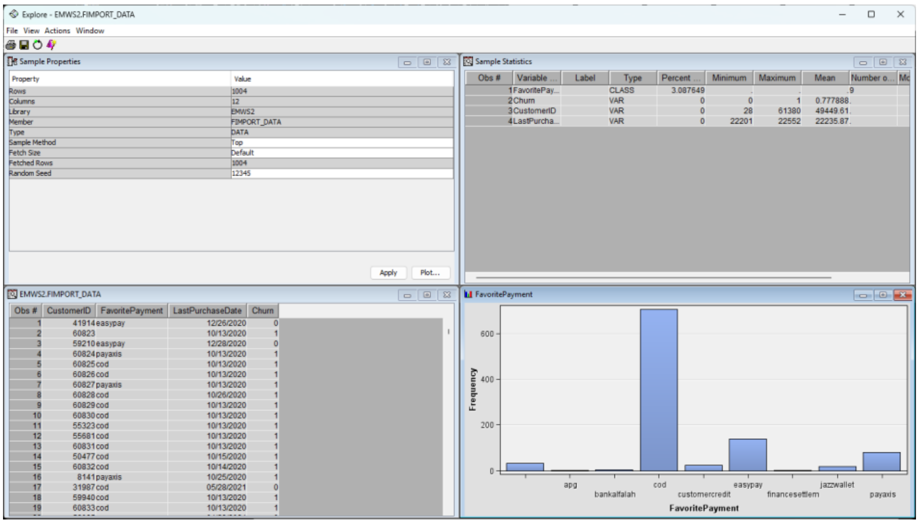


Property	Value
Rows	1004
Columns	12
Library	EMWS2
Member	FIMPORT_DATA
Type	DATA
Sample Method	Top
Fetch Size	Default
Fetched Rows	1004
Random Seed	12345

Obs #	Variable ...	Label	Type	Percent ...	Minimum	Maximum	Mean
1	Churn		VAR	0	0	1	0.777888
2	CustomerID		VAR	0	28	61380	49449.61
3	LastPurcha...		VAR	0	22201	22552	22235.87
4	TotalSpent		VAR	4.38247	0	453575.3	6184.015

Obs #	CustomerID	TotalSpent	LastPurchaseDate	Churn
1	41914	4613.7	12/26/2020	0
2	60823	72	10/13/2020	1
3	59210	1311.46	12/28/2020	0
4	60824	959.9	10/13/2020	1
5	60825	635	10/13/2020	1
6	60826	255.5	10/13/2020	1
7	60827	29500	10/13/2020	1
8	60828	475	10/26/2020	1
9	60829	250	10/13/2020	1
10	60830	250	10/13/2020	1
11	55323	812.5	10/13/2020	1
12	55681	269.9	10/13/2020	1
13	60831	250	10/13/2020	1
14	50477	.	10/15/2020	1
15	60832	566.6	10/14/2020	1
16	8141	4656.78	10/25/2020	1
17	31987	931.3	05/28/2021	0
18	59940	104.9	10/13/2020	1
19	60833	250	10/13/2020	1

6. Similarly for other class variables such as FavouritePayment and MembershipLevel, the missing values is shown as a class in the bar chart with empty class name.
   a. Favourite Payment

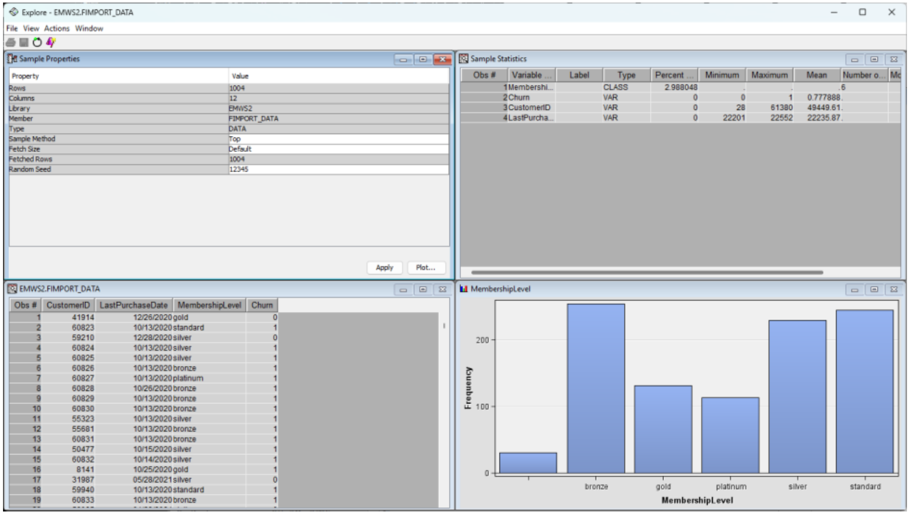


Property	Value
Rows	1004
Columns	12
Library	EMWS2
Member	FIMPORT_DATA
Type	DATA
Sample Method	Top
Fetch Size	Default
Fetched Rows	1004
Random Seed	12345

Obs #	Variable ...	Label	Type	Percent ...	Minimum	Maximum	Mean	Number o...	Mc
1	FavoritePay...		CLASS	3.087649	.	.	.	9	
2	Churn		VAR	0	0	1	0.777888	.	
3	CustomerID		VAR	0	28	61380	49449.61	.	
4	LastPurcha...		VAR	0	22201	22552	22235.87	.	

Obs #	CustomerID	FavoritePayment	LastPurchaseDate	Churn
1	41914	easypay	12/26/2020	0
2	60823		10/13/2020	1
3	59210	easypay	12/28/2020	0
4	60824	payaxis	10/13/2020	1
5	60825	cod	10/13/2020	1
6	60826	cod	10/13/2020	1
7	60827	payaxis	10/13/2020	1
8	60828	cod	10/26/2020	1
9	60829	cod	10/13/2020	1
10	60830	cod	10/13/2020	1
11	55323	cod	10/13/2020	1
12	55681	cod	10/13/2020	1
13	60831	cod	10/13/2020	1
14	50477	cod	10/15/2020	1
15	60832	cod	10/14/2020	1
16	8141	payaxis	10/25/2020	1
17	31987	cod	05/28/2021	0
18	59940	cod	10/13/2020	1
19	60833	cod	10/13/2020	1

b. MembershipLevel

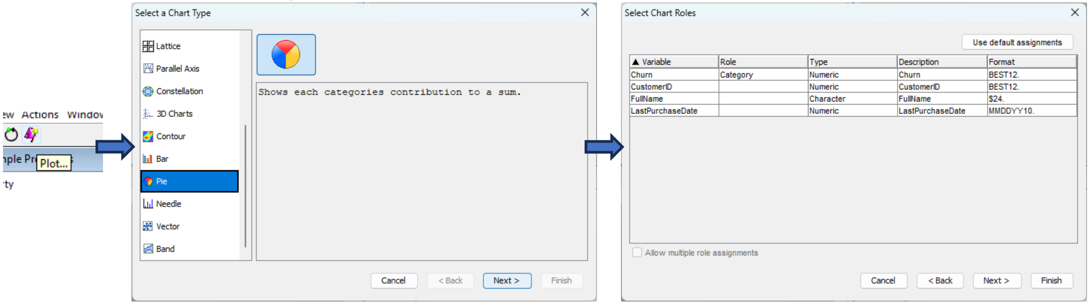


Property	Value
Rows	1004
Columns	12
Library	EMWS2
Member	FIMPORT_DATA
Type	DATA
Sample Method	Top
Fetch Size	Default
Fetched Rows	1004
Random Seed	12345

Obs #	Variable ...	Label	Type	Percent ...	Minimum	Maximum	Mean	Number o...	Mc
1	Membershi...		CLASS	2.988048	.	.	.	6	
2	Churn		VAR	0	0	1	0.777888	.	
3	CustomerID		VAR	0	28	61380	49449.61	.	
4	LastPurcha...		VAR	0	22201	22552	22235.87	.	

Obs #	CustomerID	LastPurchaseDate	MembershipLevel	Churn
1	41914	12/26/2020	gold	0
2	60823	10/13/2020	standard	1
3	59210	12/28/2020	silver	0
4	60824	10/13/2020	silver	1
5	60825	10/13/2020	silver	1
6	60826	10/13/2020	bronze	1
7	60827	10/13/2020	platinum	1
8	60828	10/26/2020	bronze	1
9	60829	10/13/2020	bronze	1
10	60830	10/13/2020	bronze	1
11	55323	10/13/2020	silver	1
12	55681	10/13/2020	bronze	1
13	60831	10/13/2020	bronze	1
14	50477	10/15/2020	silver	1
15	60832	10/14/2020	silver	1
16	8141	10/25/2020	gold	1
17	31987	05/28/2021	silver	0
18	59940	10/13/2020	standard	1
19	60833	10/13/2020	bronze	1

7. We can also use the same method to view the distribution of the target variable – Churn. Select the Plot… icon in the Churn Explore window. Then from the pop-up window, create a pie chart and click Next >. Define the role of Churn variable as Category, and click Finish to create a pie chart for Churn.

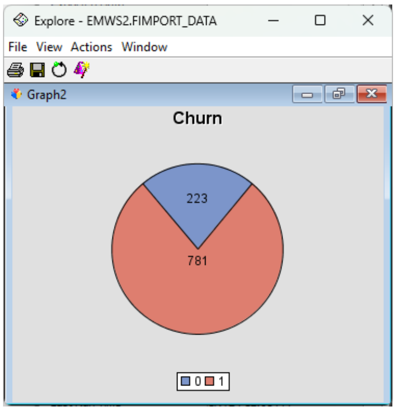


Shows each categories contribution to a sum.

▲ Variable	Role	Type	Description	Format
Churn	Category	Numeric	Churn	BEST12.
CustomerID		Numeric	CustomerID	BEST12.
FullName		Character	FullName	$24.
LastPurchaseDate		Numeric	LastPurchaseDate	MMDDYY10.

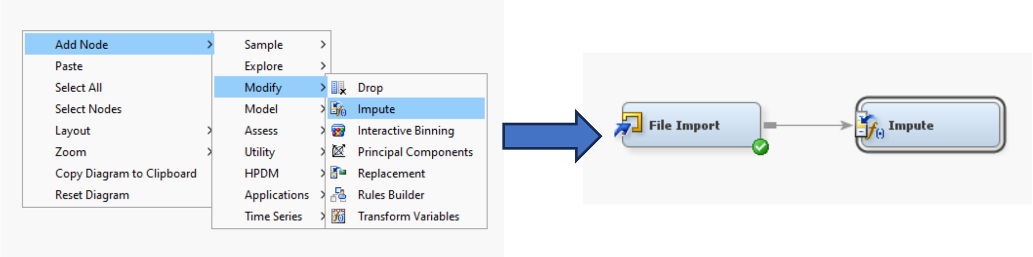
□ Allow multiple role assignments

From the pie chart shown, we can see that the variable has imbalance data.

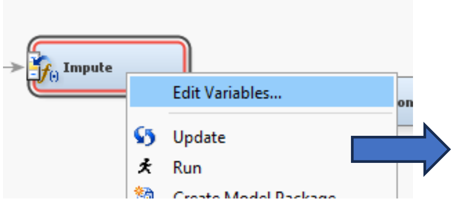


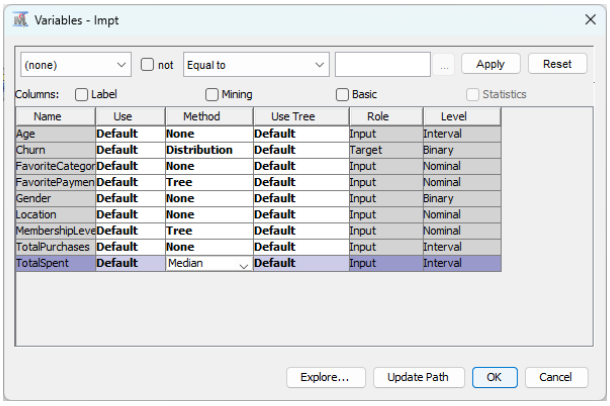
E. Impute missing values

1. To impute the missing value, right click on the diagram panel, select Add Node > Modify > Impute to create an Impute node. Connect the File Import node to the Impute node.



2. Right click on the Impute node, select Edit Variables… to edit the variables needed to be imputed and method to impute the missing values.
For the interval variable TotalSpent, we select the impute method as Median as the variable data is skewed and has outliers as shown in the histogram, the median become a better choice for imputation. Median is less sensitive to extreme values than the mean.
For the class variables, FavouritePayment and MembershipLevel, we choose the imputation method as Tree. The Tree method uses the Tree setting to replace missing class variable values with replacement values that are estimated by analyzing each input as a target. The remaining input and rejected variables are used as predictors. Because the imputed value for each input variable is based on the other input variables, this imputation technique might be more accurate than simply using the variable mean or median to replace the missing tree values.
Finally, we select Distribution as the method to impute the Churn class variable. This method uses the Distribution setting to replace missing target variable values with replacement values that are calculated based on the random percentiles of the variable's distribution. In this case, the assignment of values is based on the probability distribution of the nonmissing observations. The distribution imputation method typically does not change the distribution of the data very much.





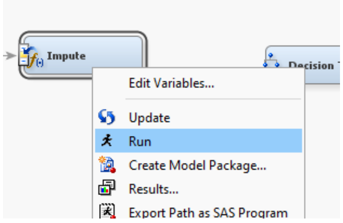
Variables - Impt

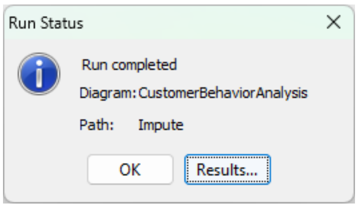
(none) □ not Equal to ... Apply Reset

Columns: □ Label □ Mining □ Basic □ Statistics

Name	Use	Method	Use Tree	Role	Level
Age	Default	None	Default	Input	Interval
Churn	Default	Distribution	Default	Target	Binary
FavoriteCategor	Default	None	Default	Input	Nominal
FavoritePaymen	Default	Tree	Default	Input	Nominal
Gender	Default	None	Default	Input	Binary
Location	Default	None	Default	Input	Nominal
MembershipLeve	Default	Tree	Default	Input	Nominal
TotalPurchases	Default	None	Default	Input	Interval
TotalSpent	Default	Median	Default	Input	Interval

3. Next, right click on the Impute to select Run to run the node. After the node is successfully ran, click om Results to view the imputation summary and output. The impute node creates new variables with naming starts with “IMP_” to store the imputed variables.



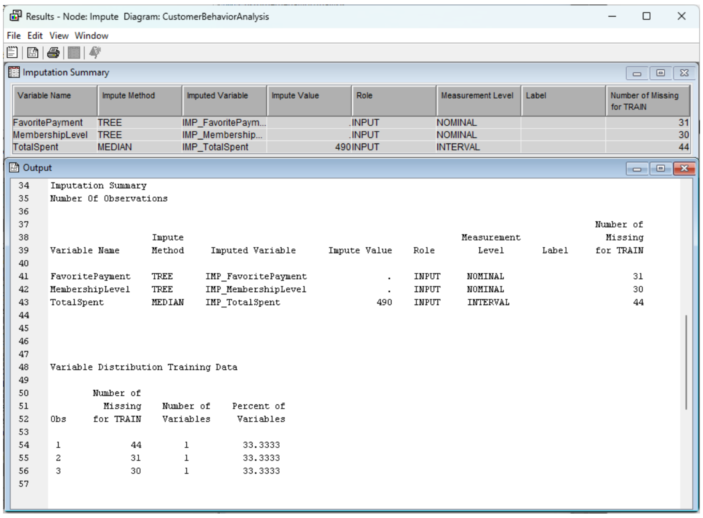


Run Status

Run completed

Diagram: CustomerBehaviorAnalysis

Path: Impute



Results - Node: Impute Diagram: CustomerBehaviorAnalysis

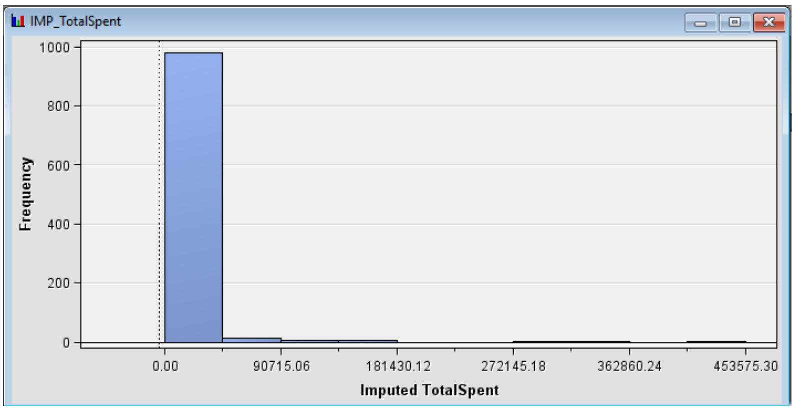
Imputation Summary

Variable Name	Impute Method	Imputed Variable	Impute Value	Role	Measurement Level	Label	Number of Missing for TRAIN
FavoritePayment	TREE	IMP_FavoritePaym...	.	INPUT	NOMINAL		31
MembershipLevel	TREE	IMP_Membership...	.	INPUT	NOMINAL		30
TotalSpent	MEDIAN	IMP_TotalSpent	490	INPUT	INTERVAL		44

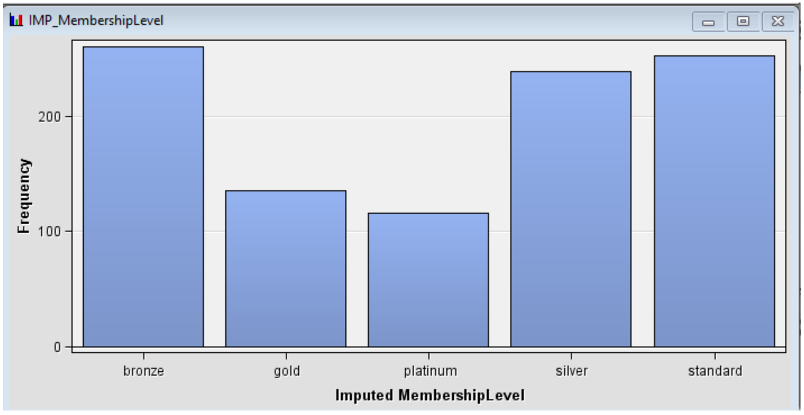
Output

```
Imputation Summary
Number Of Observations

                                                                              Number of
                 Impute                                          Measurement               Missing
Variable Name    Method     Imputed Variable     Impute Value   Role      Level       Label   for TRAIN

FavoritePayment  TREE       IMP_FavoritePayment             .   INPUT     NOMINAL                   31
MembershipLevel  TREE       IMP_MembershipLevel             .   INPUT     NOMINAL                   30
TotalSpent       MEDIAN     IMP_TotalSpent                490   INPUT     INTERVAL                  44




Variable Distribution Training Data

         Number of
           Missing    Number of    Percent of
Obs      for TRAIN    Variables     Variables

1               44        1          33.3333
2               31        1          33.3333
3               30        1          33.3333

```

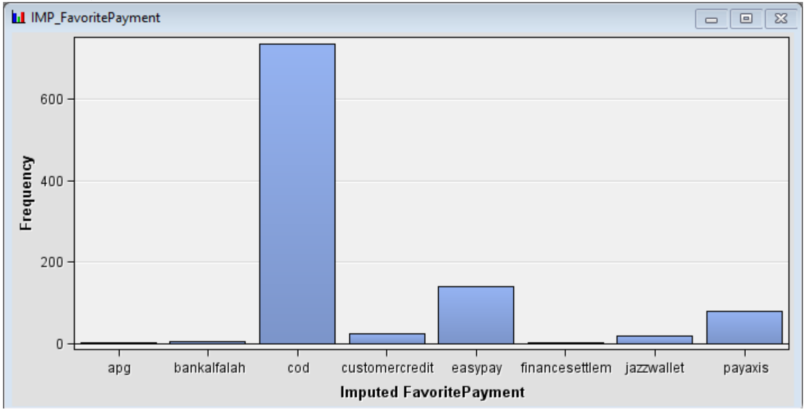
4. Then, we inspect the imputed variables, noting that there are no missing values remaining.
   a. TotalSpent



b. MembershipLevel



c. FavoritePayment



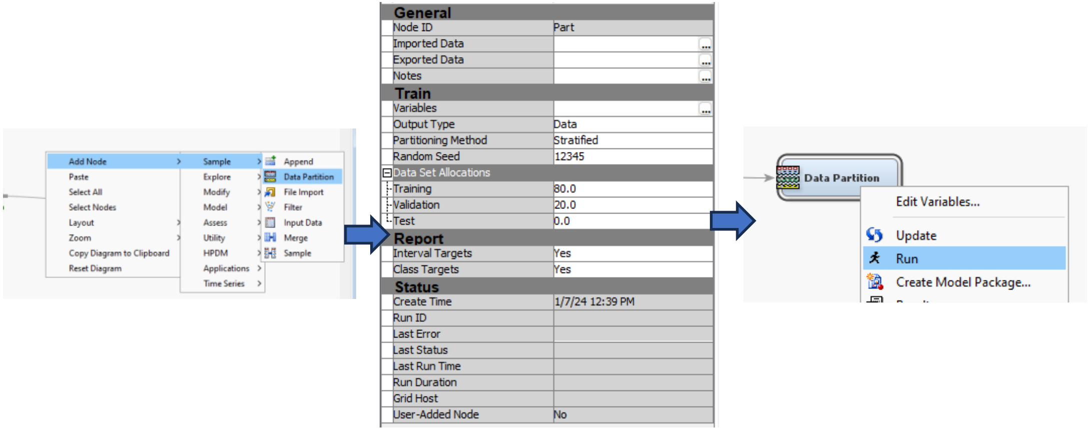
The dataset is now imported and preprocessed successfully. We can proceed with the analysis of machine learning models.

Decision Tree Analysis

A. Create training and validation set

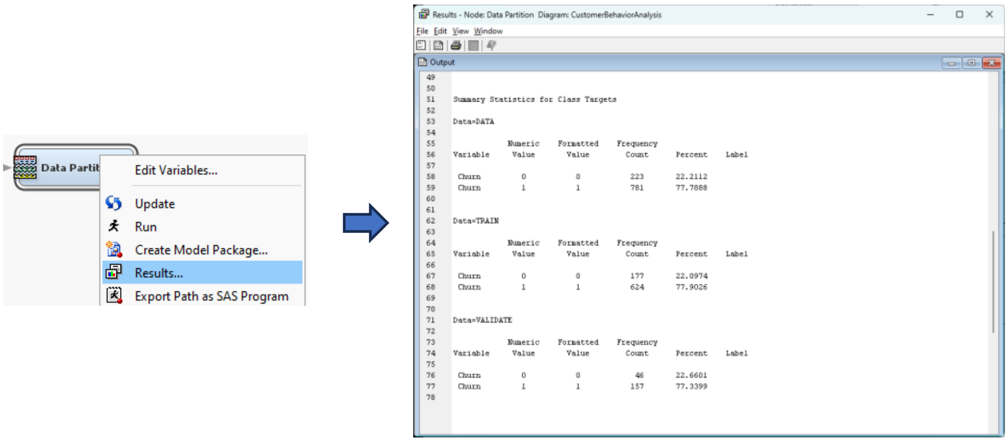
Before applying decision tree analysis, we have to create training and validation sets from the dataset. This step is crucial for assessing the performance of the model on data it has not been trained on. The training set is used to build the decision tree model, while the validation set allows us to evaluate how well the model generalizes to new, unseen data. This process helps ensure that the decision tree accurately captures patterns in the data and can make reliable predictions on new instances.

1. Right-click on the diagram panel, go to Add Node > Sample > Data Partition to create a Data Partition node. In the properties of the node, specify the partitioning method as Stratified. By using stratified partitioning, we specify variables to form strata (or subgroups) of the total population. Within each stratum, all observations have an equal probability of being written to one of the partitioned data sets. We perform stratified partitioning to preserve the strata proportions of the population within each partition data set. This might improve the classification precision of fitted models. Also, allocate the data set with training = 80.0 and validation = 20.0 to achieve an 80%-20% split. Lastly, right click on the node to Run the node.



General	
Node ID	Part
Imported Data	...
Exported Data	...
Notes	...
Train	
Variables	...
Output Type	Data
Partitioning Method	Stratified
Random Seed	12345
Data Set Allocations	
Training	80.0
Validation	20.0
Test	0.0
Report	
Interval Targets	Yes
Class Targets	Yes
Status	
Create Time	1/7/24 12:39 PM
Run ID	
Last Error	
Last Status	
Last Run Time	
Run Duration	
Grid Host	
User-Added Node	No

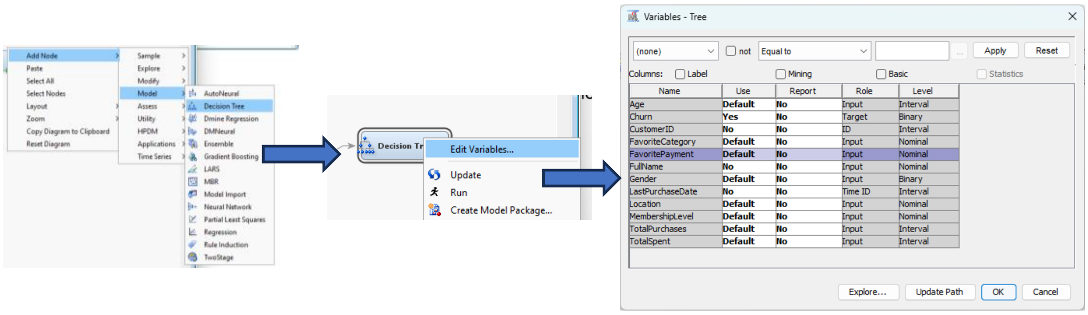
2. Right click on the node, select Results… to view the result of the node. In the output of the Results window, we can view how the data is spitted in to training and validation sets.



```


Summary Statistics for Class Targets

Data=DATA

                Numeric    Formatted    Frequency
Variable         Value       Value        Count       Percent    Label

 Churn             0           0            223       22.2112
 Churn             1           1            781       77.7888


Data=TRAIN

                Numeric    Formatted    Frequency
Variable         Value       Value        Count       Percent    Label

 Churn             0           0            177       22.0974
 Churn             1           1            624       77.9026


Data=VALIDATE

                Numeric    Formatted    Frequency
Variable         Value       Value        Count       Percent    Label

 Churn             0           0             46       22.6601
 Churn             1           1            157       77.3399

```

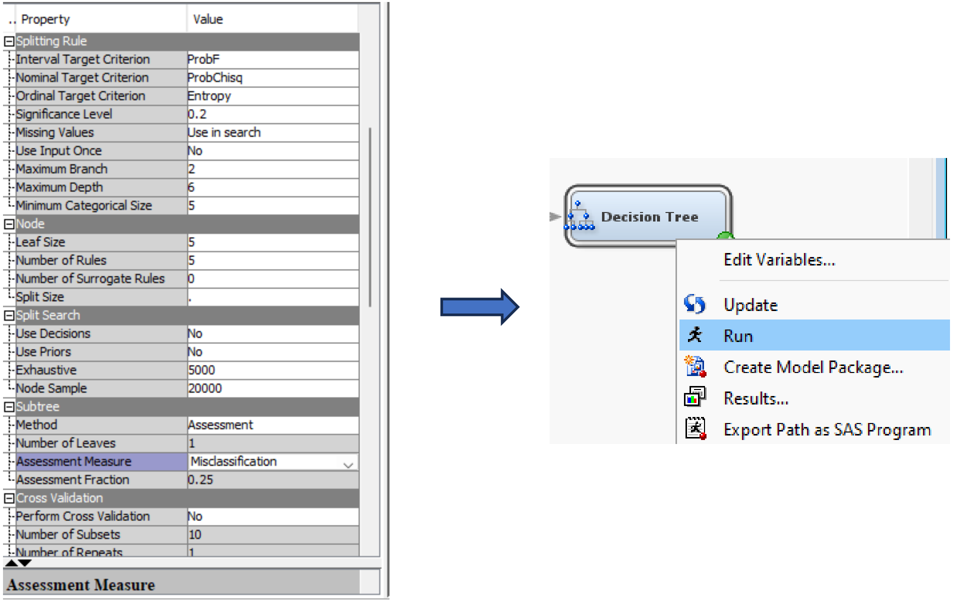
B. Decision Tree Analysis

1. Right-click on the diagram panel, select Add Node > Model > Decision Tree to create a Decision Tree node. Then, right-click on the node, select Edit Variables... to define the variable roles. In this case, we will not use CustomerID, FullName, and LastPurchaseDate in this decision tree analysis as they are ID variables. Excluding these variables is justified to avoid introducing noise and overfitting, as CustomerID is a unique identifier, FullName is likely unrelated to behavior, and LastPurchaseDate may not contribute meaningfully to the predictive power of the model.



Name	Use	Report	Role	Level
Age	Default	No	Input	Interval
Churn	Yes	No	Target	Binary
CustomerID	No	No	ID	Interval
FavoriteCategory	Default	No	Input	Nominal
FavoritePayment	Default	No	Input	Nominal
FullName	No	No	Input	Nominal
Gender	Default	No	Input	Binary
LastPurchaseDate	No	No	Time ID	Interval
Location	Default	No	Input	Nominal
MembershipLevel	Default	No	Input	Nominal
TotalPurchases	Default	No	Input	Interval
TotalSpent	Default	No	Input	Interval

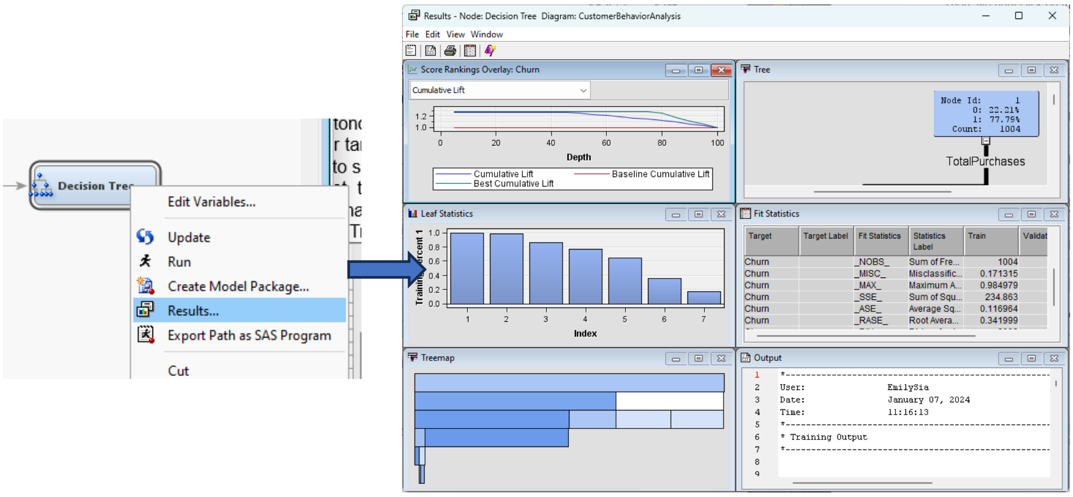
2. Next, to create a decision tree autonomously, we use misclassification as the model assessment measure because our target variable – Churn is categorical. This model assessment measure will help us to select the best tree, based on the validation data when the Method property is set to Assessment. By using misclassification as assessment measure, the tree that has the smallest misclassification rate is selected. Then, we right click on the Decision Tree to run the node.



Property	Value
Splitting Rule	
Interval Target Criterion	ProbF
Nominal Target Criterion	ProbChisq
Ordinal Target Criterion	Entropy
Significance Level	0.2
Missing Values	Use in search
Use Input Once	No
Maximum Branch	2
Maximum Depth	6
Minimum Categorical Size	5
Node	
Leaf Size	5
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	Misclassification
Assessment Fraction	0.25
Cross Validation	
Perform Cross Validation	No
Number of Subsets	10
Number of Repeats	1

Assessment Measure

3. Right click on the Decision Tree node and select Results… to view the decision tree result.

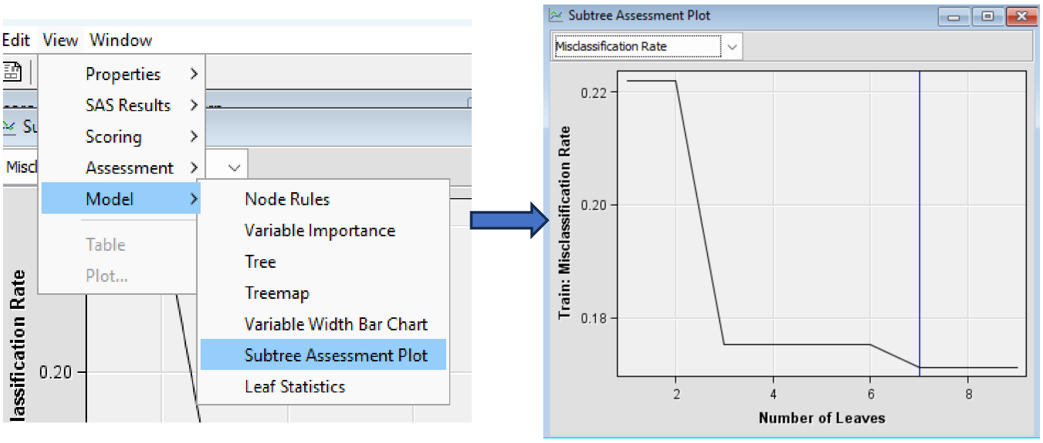


```
Node Id:     1
     0:  22.21%
     1:  77.79%
 Count:   1004
```

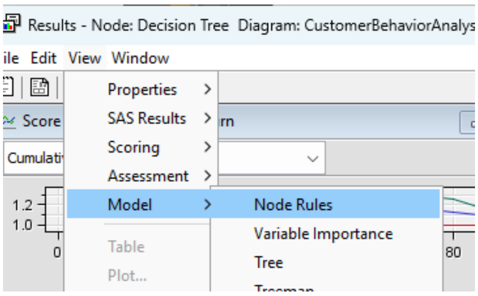
Target	Target Label	Fit Statistics	Statistics Label	Train	Validat
Churn		_NOBS_	Sum of Fre...	1004	
Churn		_MISC_	Misclassific...	0.171315	
Churn		_MAX_	Maximum A...	0.984979	
Churn		_SSE_	Sum of Squ...	234.863	
Churn		_ASE_	Average Sq...	0.116964	
Churn		_RASE_	Root Avera...	0.341999	

```
*------------------------------------------------------------
User:                    EmilySia
Date:                    January 07, 2024
Time:                    11:16:13
*------------------------------------------------------------
* Training Output
*------------------------------------------------------------


```

4. Select View > Model > Subtree Assessment Plot from the Result window menu. The Subtree Assessment Plot window opens. This is the easiest way to determine the number of leaves in the tree. Using misclassification rate as the assessment measure results in a tree with 7 leaves.



5. Select View > Model > Node Rule to view the interpretable node definition for the leaf node in a tree model.

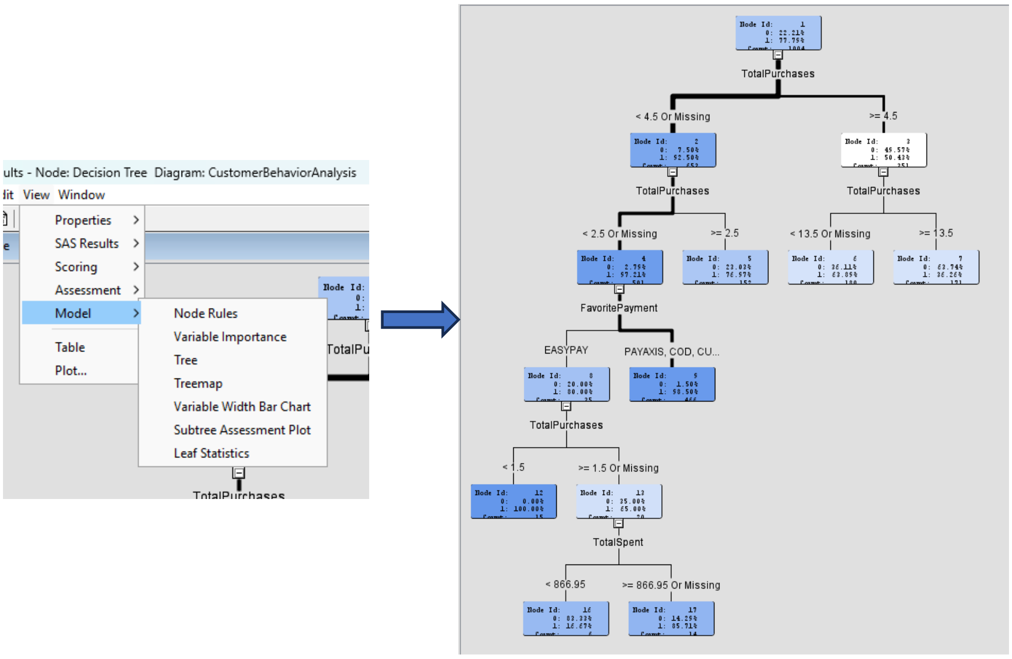


Below shows the node rule for the decision tree with 7 leaves. The decision tree analysis reveals distinct segments within the dataset, each characterized by specific conditions and associated predictions for customer churn. In Node 5, customers with TotalPurchases between 2.5 and 4.5 exhibit a high predicted churn probability of 0.77, while Node 6 encompasses those with TotalPurchases between 4.5 and 13.5 or missing, with a lower predicted churn probability of 0.64. Conversely, Node 7 captures customers with TotalPurchases exceeding 13.5, predicting a reduced churn probability of 0.36. Node 9 identifies a substantial segment with TotalPurchases less than 2.5 or missing, wherein the predicted churn probability is notably high at 0.98. Node 12 focuses on a smaller group with TotalPurchases less than 1.5 and a specific FavoritePayment method, resulting in a predicted churn probability of 1.00. Nodes 16 and 17 further stratify customers based on TotalSpent, TotalPurchases, and FavoritePayment, providing nuanced predictions with varying churn probabilities.

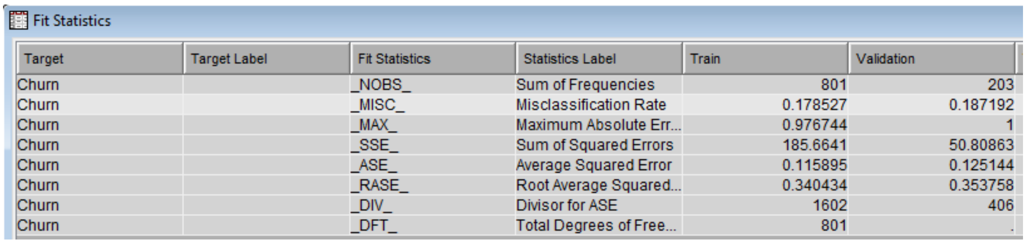
```
*------------------------------------------------------------*
 Node = 5
*------------------------------------------------------------*
if TotalPurchases < 4.5 AND TotalPurchases >= 2.5
then
 Tree Node Identifier   = 5
 Number of Observations = 152
 Predicted: Churn=1 = 0.77
 Predicted: Churn=0 = 0.23

*------------------------------------------------------------*
```

```
 Node = 6
*------------------------------------------------------------*
if TotalPurchases < 13.5 AND TotalPurchases >= 4.5 or MISSING
then
 Tree Node Identifier   = 6
 Number of Observations = 180
 Predicted: Churn=1 = 0.64
 Predicted: Churn=0 = 0.36

*------------------------------------------------------------*
 Node = 7
*------------------------------------------------------------*
if TotalPurchases >= 13.5
then
 Tree Node Identifier   = 7
 Number of Observations = 171
 Predicted: Churn=1 = 0.36
 Predicted: Churn=0 = 0.64

*------------------------------------------------------------*
 Node = 9
*------------------------------------------------------------*
if TotalPurchases < 2.5 or MISSING
AND FavoritePayment IS ONE OF: PAYAXIS, COD, CUSTOMERCREDIT, JAZZWALLET or MISSING
then
 Tree Node Identifier   = 9
 Number of Observations = 466
 Predicted: Churn=1 = 0.98
 Predicted: Churn=0 = 0.02

*------------------------------------------------------------*
 Node = 12
*------------------------------------------------------------*
if TotalPurchases < 1.5
AND FavoritePayment IS ONE OF: EASYPAY
then
 Tree Node Identifier   = 12
 Number of Observations = 15
 Predicted: Churn=1 = 1.00
 Predicted: Churn=0 = 0.00

*------------------------------------------------------------*
 Node = 16
*------------------------------------------------------------*
if TotalSpent < 866.95
AND TotalPurchases < 2.5 AND TotalPurchases >= 1.5 or MISSING
AND FavoritePayment IS ONE OF: EASYPAY
then
 Tree Node Identifier   = 16
 Number of Observations = 6
 Predicted: Churn=1 = 0.17
 Predicted: Churn=0 = 0.83

*------------------------------------------------------------*
 Node = 17
*------------------------------------------------------------*
if TotalSpent >= 866.95 or MISSING
AND TotalPurchases < 2.5 AND TotalPurchases >= 1.5 or MISSING
AND FavoritePayment IS ONE OF: EASYPAY
then
 Tree Node Identifier   = 17
 Number of Observations = 14
 Predicted: Churn=1 = 0.86
 Predicted: Churn=0 = 0.14
```

6. Select View > Model > Tree to open the Tree Plot window.



7. According to the fit statistics of the decision tree, the misclassification rate for the training set is recorded as 0.178527, while the validation set shows a slightly higher misclassification rate of 0.187192.
   These misclassification rates serve as indicators of the model's performance on the respective datasets. A lower misclassification rate suggests better accuracy in predicting the target variable (Churn in this case). The slightly higher misclassification rate in the validation set compared to the training set may indicate that the model is generalizing well to new, unseen data, but there might still be room for improvement.

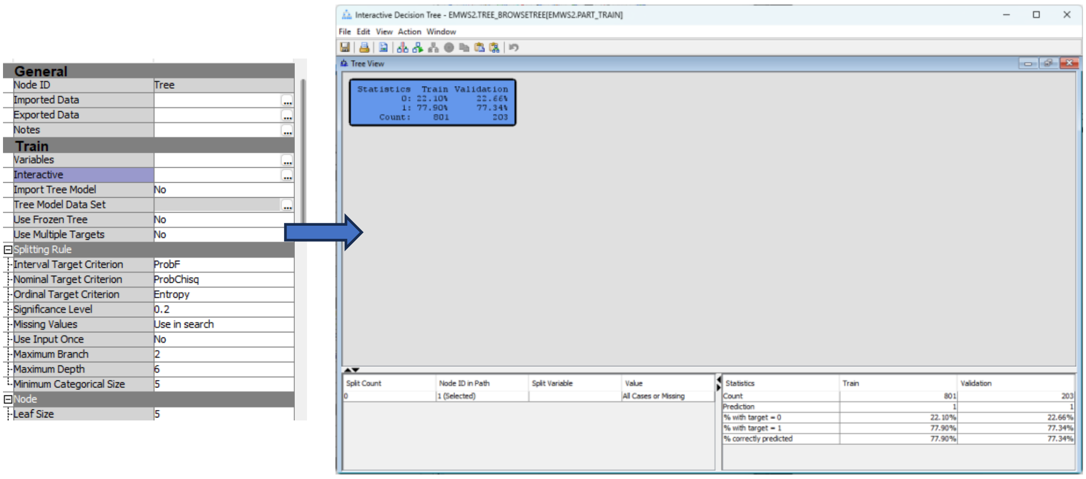


Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation
Churn		_NOBS_	Sum of Frequencies	801	203
Churn		_MISC_	Misclassification Rate	0.178527	0.187192
Churn		_MAX_	Maximum Absolute Err...	0.976744	1
Churn		_SSE_	Sum of Squared Errors	185.6641	50.80863
Churn		_ASE_	Average Squared Error	0.115895	0.125144
Churn		_RASE_	Root Average Squared...	0.340434	0.353758
Churn		_DIV_	Divisor for ASE	1602	406
Churn		_DFT_	Total Degrees of Free...	801	.

C. Underfitting of decision tree

1. Click on the Interactive "…" button from the decision tree node Properties panel. The SAS Enterprise Miner Tree window opens.



General	
Node ID	Tree
Imported Data	...
Exported Data	...
Notes	...
Train	
Variables	...
Interactive	...
Import Tree Model	No
Tree Model Data Set	...
Use Frozen Tree	No
Use Multiple Targets	No
⊟Splitting Rule	
Interval Target Criterion	ProbF
Nominal Target Criterion	ProbChisq
Ordinal Target Criterion	Entropy
Significance Level	0.2
Missing Values	Use in search
Use Input Once	No
Maximum Branch	2
Maximum Depth	6
Minimum Categorical Size	5
⊟Node	
Leaf Size	5

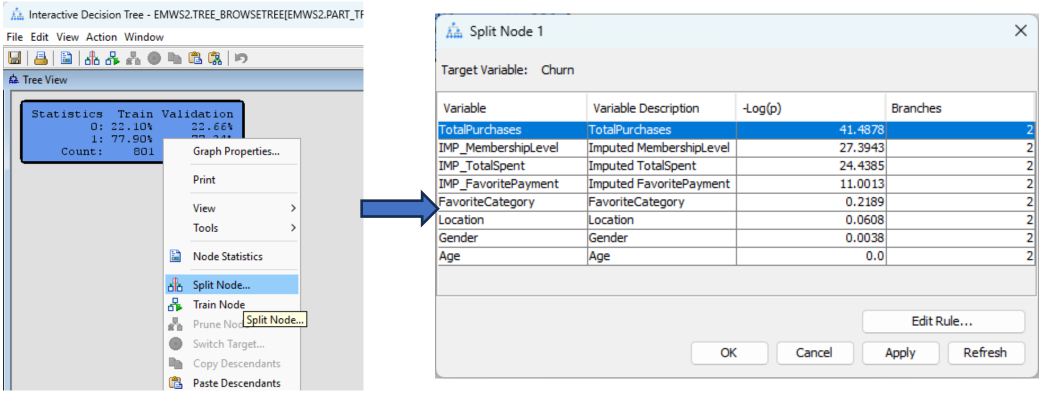
Interactive Decision Tree - EMWS2.TREE_BROWSETREE[EMWS2.PART_TRAIN]
File Edit View Action Window
Tree View

```
Statistics   Train  Validation
        0:  22.10%      22.66%
        1:  77.90%      77.34%
   Count:     801         203
```

Split Count	Node ID in Path	Split Variable	Value
0	1 (Selected)		All Cases or Missing

Statistics	Train	Validation
Count	801	203
Prediction	1	1
% with target = 0	22.10%	22.66%
% with target = 1	77.90%	77.34%
% correctly predicted	77.90%	77.34%

2. Right click the root node and select Split Node… from the menu. The Split node 1 window opens. The Split node 1 window shows the TotalPurchases is used for the first split. Competing splits are IMP_MembershipLevel, IMP_TotalSpent, IMP_FavoritePayment, FavoriteCategory, Location, Gender and Age.



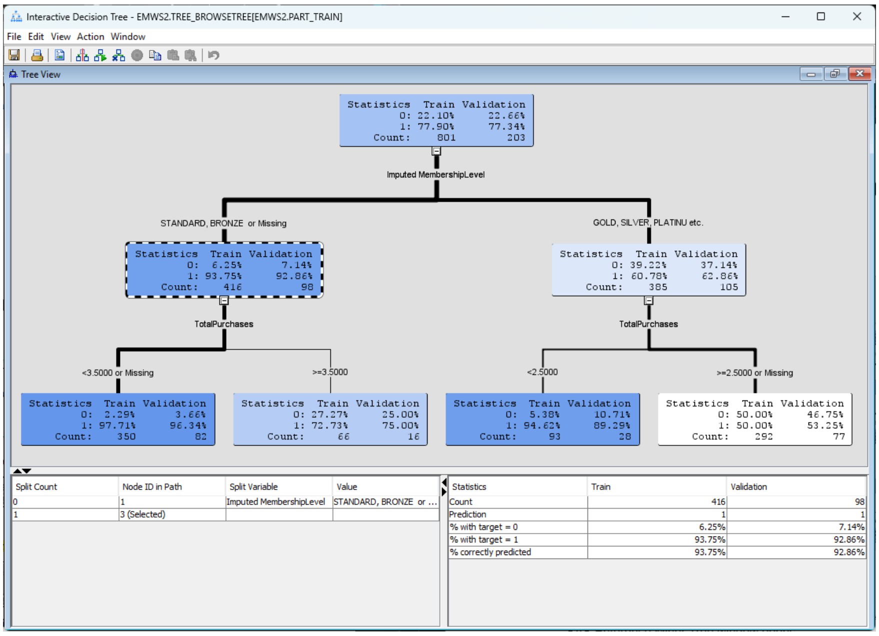
```
Statistics   Train  Validation
        0:  22.10%      22.66%
        1:  77.90%
   Count:     801
```

Split Node 1

Target Variable: Churn

Variable	Variable Description	-Log(p)	Branches
TotalPurchases	TotalPurchases	41.4878	2
IMP_MembershipLevel	Imputed MembershipLevel	27.3943	2
IMP_TotalSpent	Imputed TotalSpent	24.4385	2
IMP_FavoritePayment	Imputed FavoritePayment	11.0013	2
FavoriteCategory	FavoriteCategory	0.2189	2
Location	Location	0.0608	2
Gender	Gender	0.0038	2
Age	Age	0.0	2

3. Next, we apply the first 2 variables for the split. The resulting Tree plot is shown as below.



Interactive Decision Tree - EMWS2.TREE_BROWSETREE[EMWS2.PART_TRAIN]
File Edit View Action Window
Tree View
Imputed MembershipLevel
STANDARD, BRONZE or Missing
GOLD, SILVER, PLATINU etc.
TotalPurchases
TotalPurchases
<3.5000 or Missing
>=3.5000
<2.5000
>=2.5000 or Missing

```
Statistics   Train  Validation
        0:  22.10%      22.66%
        1:  77.90%      77.34%
   Count:     801         203
```

```
Statistics   Train  Validation
        0:   6.25%       7.14%
        1:  93.75%      92.86%
   Count:     416          98
```

```
Statistics   Train  Validation
        0:  39.22%      37.14%
        1:  60.78%      62.86%
   Count:     385         105
```

```
Statistics   Train  Validation
        0:   2.29%       3.66%
        1:  97.71%      96.34%
   Count:     350          82
```

```
Statistics   Train  Validation
        0:  27.27%      25.00%
        1:  72.73%      75.00%
   Count:      66          16
```

```
Statistics   Train  Validation
        0:   5.38%      10.71%
        1:  94.62%      89.29%
   Count:      93          28
```

```
Statistics   Train  Validation
        0:  50.00%      46.75%
        1:  50.00%      53.25%
   Count:     292          77
```

Split Count	Node ID in Path	Split Variable	Value
0	1	Imputed MembershipLevel	STANDARD, BRONZE or …
1	3 (Selected)		

Statistics	Train	Validation
Count	416	98
Prediction	1	1
% with target = 0	6.25%	7.14%
% with target = 1	93.75%	92.86%
% correctly predicted	93.75%	92.86%

The generated decision tree nodes exhibit signs of underfitting, specifically due to the simplicity of the conditions and the limited depth of the tree. In an underfit model, the decision tree may struggle to capture the complexity and nuances within the data, resulting in overly generalized rules that do not adequately distinguish between different scenarios.

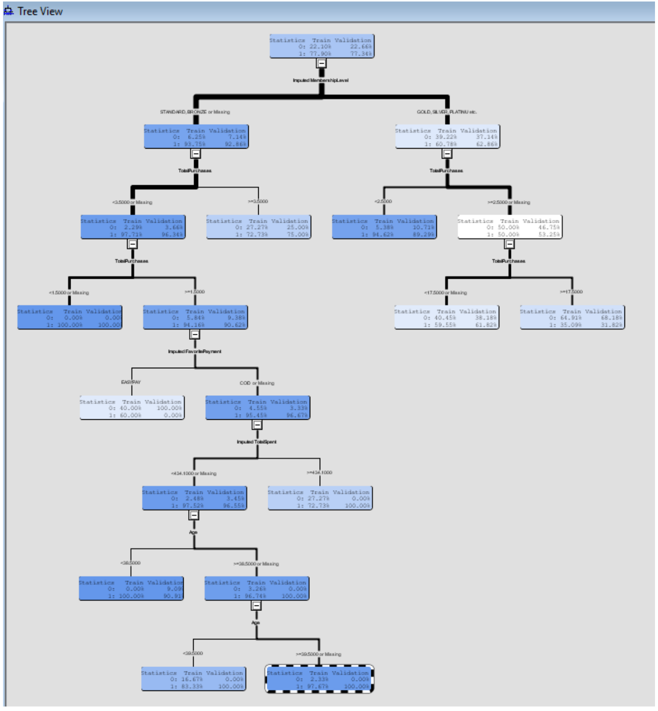
Examining the node rules, Node 3 and Node 4 act as the primary nodes for MembershipLevel, with Node 3 encompassing STANDARD, BRONZE, or missing values and Node 4 covering GOLD, SILVER, PLATINUM levels. These broad conditions at the root level and subsequent nodes suggest that the decision tree is making simplistic splits based on MembershipLevel and TotalPurchases, leading to a lack of depth and detail in the decision-making process.

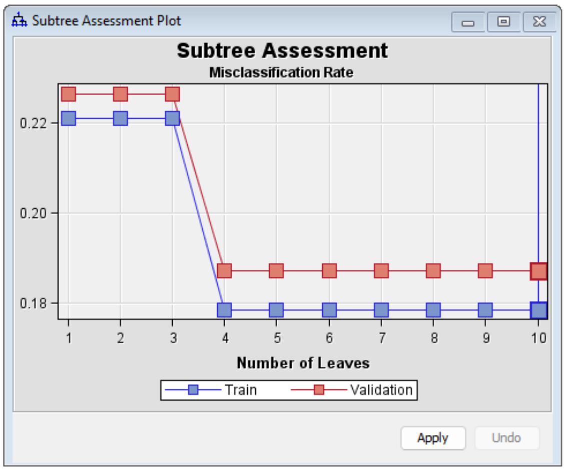
Furthermore, Node 7 and Node 8 introduce TotalPurchases as a criterion only for STANDARD, BRONZE, or missing MembershipLevels, and Nodes 5 and 6 do the same exclusively for GOLD, SILVER, PLATINUM levels. This limited incorporation of features and conditions, along with the absence of more intricate decision paths, contributes to an underfit model. An underfit decision tree may struggle to capture the variability in the data, leading to less accurate predictions and a failure to discern subtle patterns or interactions among different variables.

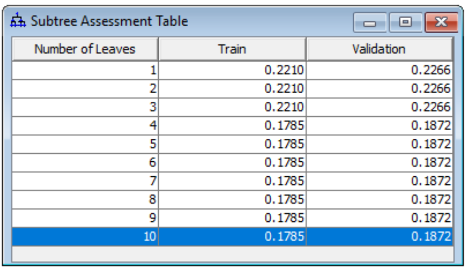
D. Overfitting of decision tree

1. We continue to split the tree until it reaches a point with 10 leaves and a depth of 8, as illustrated in the figure below. Subsequently, we access the Subtree Assessment plot and table to observe the trend of the misclassification rate. Analyzing the trend, it becomes apparent that the misclassification rates for both the training set and the validation set stabilize when the tree attains 4 leaves.

   This observation suggests that further splitting beyond 4 leaves may not significantly improve the model's predictive performance. The stabilization of misclassification rates indicates that additional complexity in the tree structure might lead to overfitting, where the model starts to capture noise in the training data rather than general patterns.

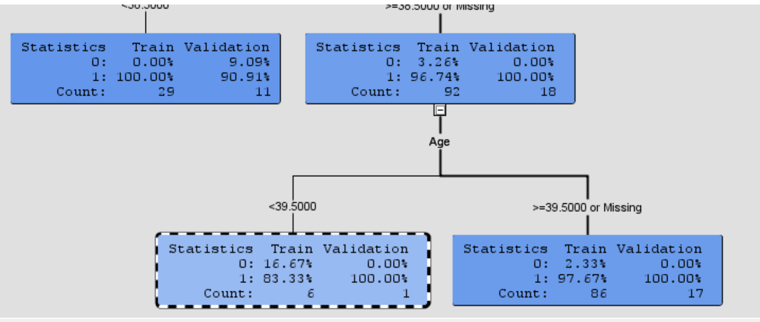

Tree View
Statistics Train Validation 0: 22.10% 22.66% 1: 77.90% 77.34%
Imputed MembershipLevel
STANDARD, BRONZE or Missing
GOLD, SILVER, PLATINU etc.
Statistics Train Validation 0: 6.25% 7.14% 1: 93.75% 92.86%
Statistics Train Validation 0: 39.22% 37.14% 1: 60.78% 62.86%
TotalPurchases
TotalPurchases
<3.5000 or Missing
>=3.5000
<2.5000
>=2.5000 or Missing
Statistics Train Validation 0: 2.29% 3.66% 1: 97.71% 96.34%
Statistics Train Validation 0: 27.27% 25.00% 1: 72.73% 75.00%
Statistics Train Validation 0: 5.38% 10.71% 1: 94.62% 89.29%
Statistics Train Validation 0: 50.00% 46.75% 1: 50.00% 53.25%
TotalPurchases
TotalPurchases
<1.5000 or Missing
>=1.5000
<17.5000 or Missing
>=17.5000
Statistics Train Validation 0: 0.00% 0.00% 1: 100.00% 100.00%
Statistics Train Validation 0: 5.84% 9.38% 1: 94.16% 90.62%
Statistics Train Validation 0: 40.45% 38.18% 1: 59.55% 61.82%
Statistics Train Validation 0: 64.91% 68.18% 1: 35.09% 31.82%
Imputed FavoritePayment
EASYPAY
COD or Missing
Statistics Train Validation 0: 40.00% 100.00% 1: 60.00% 0.00%
Statistics Train Validation 0: 4.55% 3.33% 1: 95.45% 96.67%
Imputed TotalSpent
<434.1000 or Missing
>=434.1000
Statistics Train Validation 0: 2.48% 3.45% 1: 97.52% 96.55%
Statistics Train Validation 0: 27.27% 0.00% 1: 72.73% 100.00%
Age
<38.5000
>=38.5000 or Missing
Statistics Train Validation 0: 0.00% 9.09% 1: 100.00% 90.91%
Statistics Train Validation 0: 3.26% 0.00% 1: 96.74% 100.00%
Age
<39.5000
>=39.5000 or Missing
Statistics Train Validation 0: 16.67% 0.00% 1: 83.33% 100.00%
Statistics Train Validation 0: 2.33% 0.00% 1: 97.67% 100.00%





Number of Leaves	Train	Validation
1	0.2210	0.2266
2	0.2210	0.2266
3	0.2210	0.2266
4	0.1785	0.1872
5	0.1785	0.1872
6	0.1785	0.1872
7	0.1785	0.1872
8	0.1785	0.1872
9	0.1785	0.1872
10	0.1785	0.1872

2. In Node 16, the proportion of “0” is 3.26% in the training and 0% in the validation data sets. This node is then split into Node 17 and Node 18. In Leaf 17, the proportion of “0” is 16.67% for the training data set but 0% for the validation data set. Additionally, there are only 6 observations in the training data set. This makes the leaf unreliable. The model is too complex and specific to the training data, and reduces its ability to adapt to new data.



```
Statistics   Train  Validation
        0:   0.00%       9.09%
        1: 100.00%      90.91%
    Count:      29          11
```

```
Statistics   Train  Validation
        0:   3.26%       0.00%
        1:  96.74%     100.00%
    Count:      92          18
```

```
Statistics   Train  Validation
        0:  16.67%       0.00%
        1:  83.33%     100.00%
    Count:       6           1
```

```
Statistics   Train  Validation
        0:   2.33%       0.00%
        1:  97.67%     100.00%
    Count:      86          17
```

Ensemble Methods

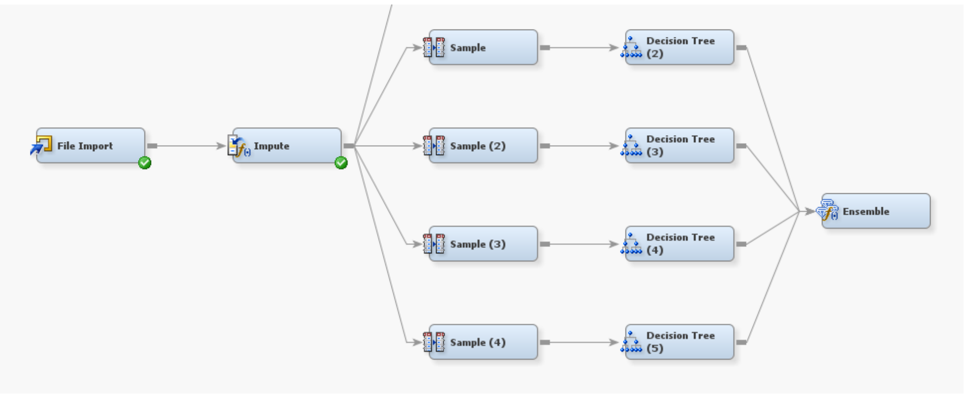
Ensemble models are the product of training several similar models and combining their results in order to improve accuracy, reduce bias, reduce variance, and provide robust models in the presence of new data.

A. Bagging – Random Forest

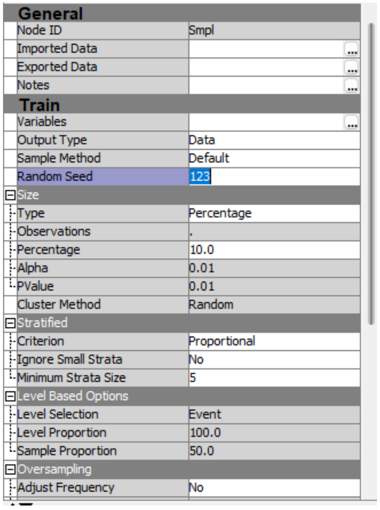
Bagging stands for bootstrap aggregating. It consists of creating several samples to train models in parallel and combining the predicted probabilities.

One way to do bagging is to build a diagram to train multiple decision trees by using different samples of the training data. We build this diagram by including all nodes as the default, except for the random seeds of the sample nodes. We specify a different random seed for each of the sample nodes, connect a decision tree, and then use a default Ensemble node to average the predicted probabilities of all connected models. An advantage of this averaging ensemble method is that it smooths the linear cut points of a decision tree, making the model more robust in handling new data.

1. Build a diagram as shown in the figure below. Multiple decision trees node are create and connected to a Ensemble node to create a Random Forest algorithm.

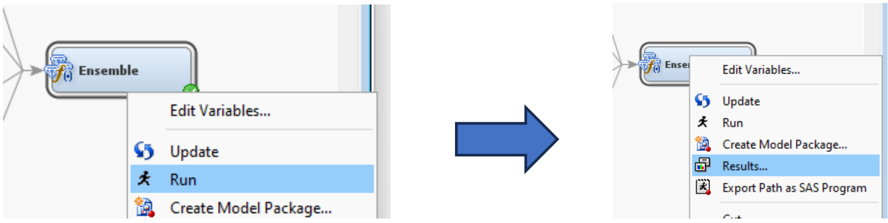


2. Specify different Random Seed for each sample nodes to create different samples.

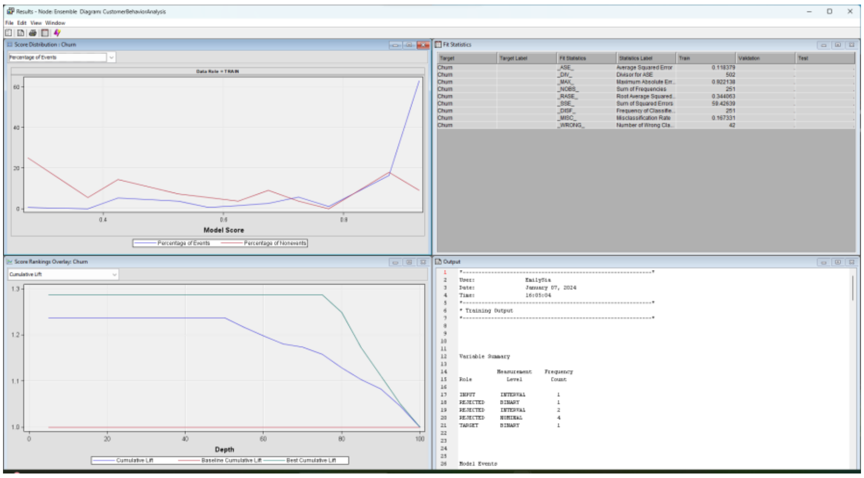


General	
Node ID	Smpl
Imported Data	...
Exported Data	...
Notes	...
Train	
Variables	...
Output Type	Data
Sample Method	Default
Random Seed	123
⊟Size	
Type	Percentage
Observations	.
Percentage	10.0
Alpha	0.01
PValue	0.01
Cluster Method	Random
⊟Stratified	
Criterion	Proportional
Ignore Small Strata	No
Minimum Strata Size	5
⊟Level Based Options	
Level Selection	Event
Level Proportion	100.0
Sample Proportion	50.0
⊟Oversampling	
Adjust Frequency	No

3. Right click on the Ensemble node, select Run to run the node. Then click the Result to view the ensemble node result.



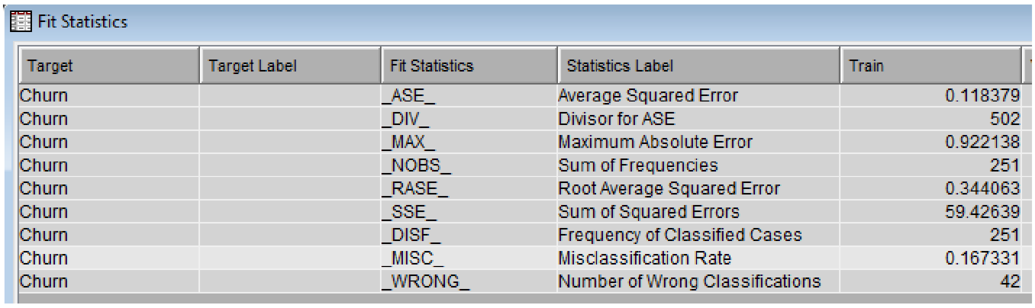
4. The result window is shown.



Target	Target Label	Fit Statistics	Statistics Label	Train	Validation	Test
Churn		_ASE_	Average Squared Error	0.118379	.	.
Churn		_DIV_	Divisor for ASE	502	.	.
Churn		_MAX_	Maximum Absolute Err...	0.922138	.	.
Churn		_NOBS_	Sum of Frequencies	251	.	.
Churn		_RASE_	Root Average Squared...	0.344063	.	.
Churn		_SSE_	Sum of Squared Errors	59.42639	.	.
Churn		_DISF_	Frequency of Classifie...	251	.	.
Churn		_MISC_	Misclassification Rate	0.167331	.	.
Churn		_WRONG_	Number of Wrong Cla...	42	.	.

```
*------------------------------------------------------------*
User:                EmilySia
Date:                January 07, 2024
Time:                16:05:04
*------------------------------------------------------------*
* Training Output
*------------------------------------------------------------*




Variable Summary

             Measurement    Frequency
Role            Level         Count

INPUT         INTERVAL          1
REJECTED      BINARY            1
REJECTED      INTERVAL          2
REJECTED      NOMINAL           4
TARGET        BINARY            1




Model Events
```

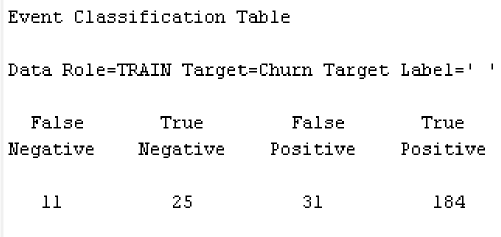
5. In the fit statistics section, specifically looking at the misclassification rate, the value is recorded as 0.167331. This rate indicates the proportion of incorrect predictions made by the ensemble model. In this case, the number of wrong classifications is specified as 42 instances.



Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train
Churn		_ASE_	Average Squared Error	0.118379
Churn		_DIV_	Divisor for ASE	502
Churn		_MAX_	Maximum Absolute Error	0.922138
Churn		_NOBS_	Sum of Frequencies	251
Churn		_RASE_	Root Average Squared Error	0.344063
Churn		_SSE_	Sum of Squared Errors	59.42639
Churn		_DISF_	Frequency of Classified Cases	251
Churn		_MISC_	Misclassification Rate	0.167331
Churn		_WRONG_	Number of Wrong Classifications	42

6. The classification table of the bagging Random Forest model is shown below. In the training dataset, the model correctly identified 25 instances as negative (True Negatives) and accurately predicted 184 instances as positive (True Positives). However, it made 31 false-positive predictions and 11 false-negative predictions.

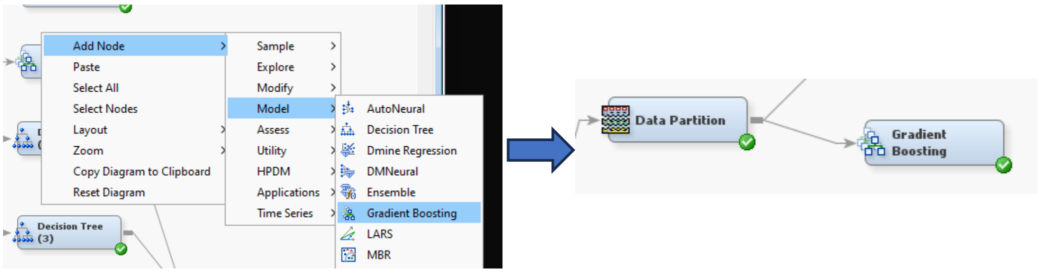


```
Event Classification Table

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

  False       True       False       True
Negative    Negative    Positive    Positive

   11          25          31          184
```

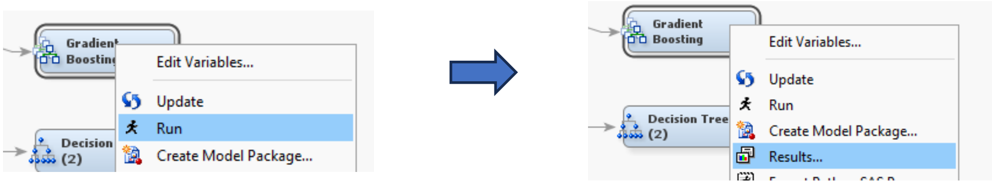
B. Boosting – Gradient Boosting

Gradient boosting is a boosting approach that resamples the analysis data set several times to generate results that form a weighted average of the re-sampled data set. Tree boosting creates a series of decision trees which together form a single predictive model. A tree in the series is fit to the residual of the prediction from the earlier trees in the series. The residual is defined in terms of the derivative of a loss function. Boosting is less prone to overfit the data than a single decision tree, and if a decision tree fits the data fairly well, then boosting often improves the fit.

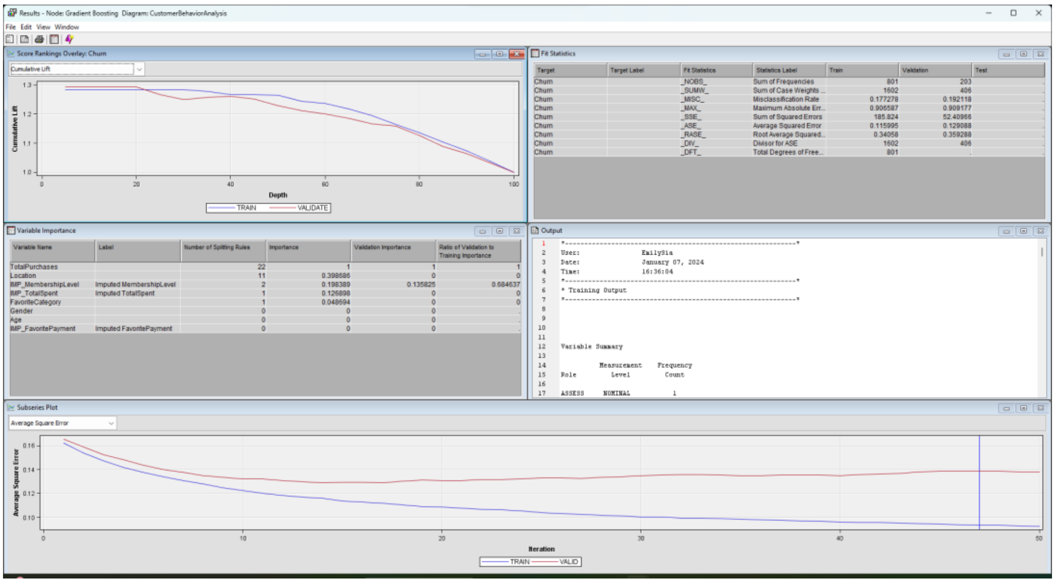
1. Right click on the diagram panel, select Add Node > Model > Gradient Boosting to create a Gradient Boosting node. Then connect the Gradient Boosting node to Data Partition node.



2. Right click on the Gradient Boosting node, select Run to run the node. Then, select Results... to view the result.

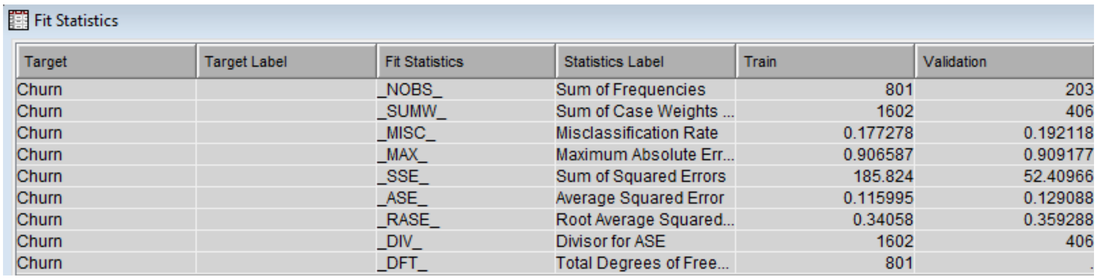


3. The Result window shows.



```
*------------------------------------------------------------*
User:                EmilySia
Date:                January 07, 2024
Time:                16:36:04
*------------------------------------------------------------*
* Training Output
*------------------------------------------------------------*




Variable Summary

            Measurement    Frequency
Role           Level         Count

ASSESS       NOMINAL           1
```

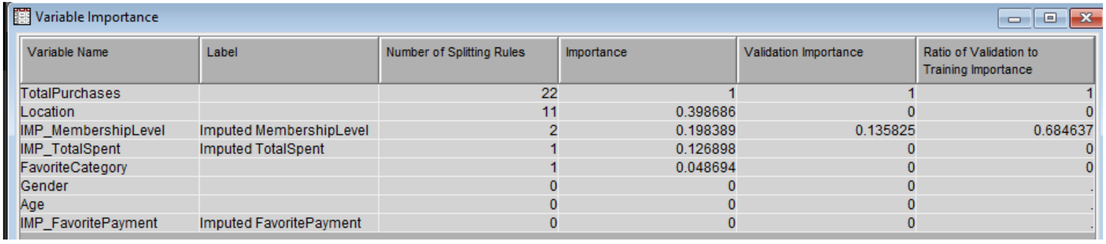
4. From the fit statistics, the misclassification rate for training set is 0.177278 and for validation set is 0.192118.



Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation
Churn		_NOBS_	Sum of Frequencies	801	203
Churn		_SUMW_	Sum of Case Weights ...	1602	406
Churn		_MISC_	Misclassification Rate	0.177278	0.192118
Churn		_MAX_	Maximum Absolute Err...	0.906587	0.909177
Churn		_SSE_	Sum of Squared Errors	185.824	52.40966
Churn		_ASE_	Average Squared Error	0.115995	0.129088
Churn		_RASE_	Root Average Squared...	0.34058	0.359288
Churn		_DIV_	Divisor for ASE	1602	406
Churn		_DFT_	Total Degrees of Free...	801	.

5. The most important variable for the Gradient Boosting model is Total Purchases, followed by Location, IMP_MembershipLevel, IMP_TotalSpent and FavoriteCategory.

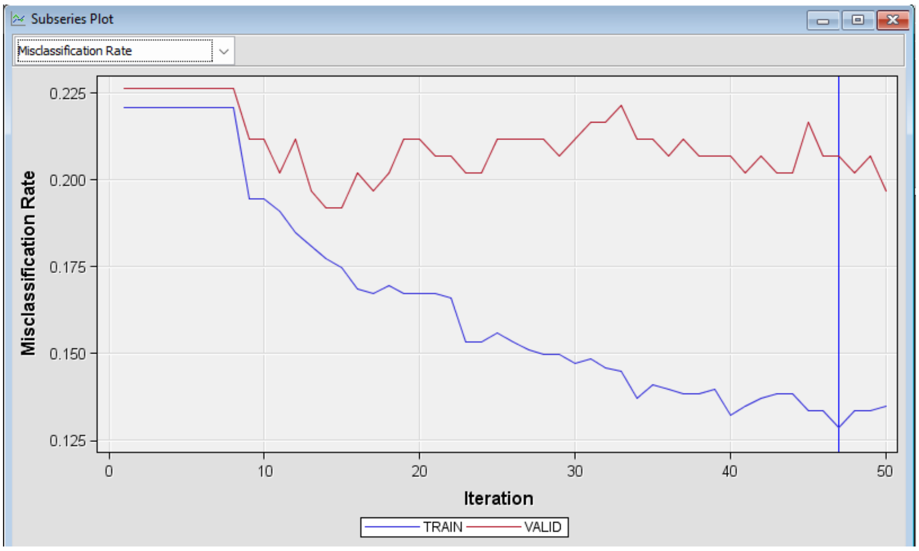


Variable Importance

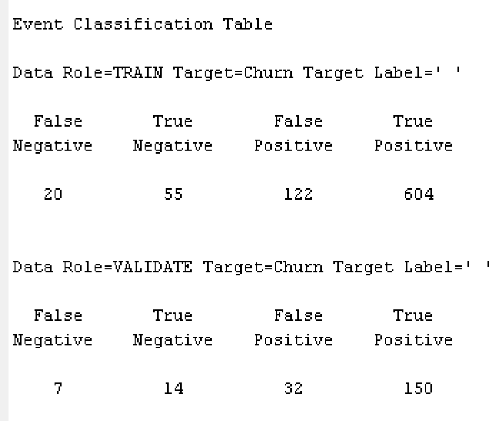
Variable Name	Label	Number of Splitting Rules	Importance	Validation Importance	Ratio of Validation to Training Importance
TotalPurchases		22	1	1	1
Location		11	0.398686	0	0
IMP_MembershipLevel	Imputed MembershipLevel	2	0.198389	0.135825	0.684637
IMP_TotalSpent	Imputed TotalSpent	1	0.126898	0	0
FavoriteCategory		1	0.048694	0	0
Gender		0	0	0	.
Age		0	0	0	.
IMP_FavoritePayment	Imputed FavoritePayment	0	0	0	.

6. From the Subseries Plot for misclassification rate, the vertical line serves as a noteworthy indicator. In this specific case, the presence of a vertical line at the 47th

iteration suggests a crucial point in the training process where the model achieves optimal performance in terms of minimising misclassification.



7. The classification table of the gradient boosting model is shown below. In the training dataset, the model correctly identified 55 instances as negative (True Negatives) and accurately predicted 604 instances as positive (True Positives). However, it made 20 false-positive predictions and 122 false-negative predictions. Similarly, in the validation dataset, the model correctly classified 14 instances as negative and 150 instances as positive, but it also produced 7 false positives and 32 false negatives.



```
Event Classification Table

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

  False       True       False       True
Negative    Negative    Positive    Positive

   20          55         122         604


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

  False       True       False       True
Negative    Negative    Positive    Positive

    7          14          32         150
```

Analysis of decision tree and ensemble methods

The decision tree analysis with 7 leaves reveals valuable insights into customer behavior, delineating distinct segments based on conditions such as TotalPurchases, TotalSpent, and FavoritePayment. Particularly, Node 9 identifies a substantial segment with low TotalPurchases, indicating a high predicted churn probability of 0.98. Additionally, Node 12 focuses on a smaller group with specific characteristics, resulting in a predicted churn probability of 1.00. These findings suggest varying levels of churn risk among customer segments. However, the fit statistics reveal a misclassification rate of 0.178527 for the training set and 0.187192 for the validation set, indicating a need for refinement. Despite generalization to new data, there is room for improvement, suggesting potential adjustments to model parameters or the consideration of additional features.

The Bagging - Random Forest ensemble method, utilizing bootstrap aggregating to combine multiple decision trees, demonstrates an improved misclassification rate of 0.167331 compared to the standalone decision tree. The ensemble method's advantage lies in its robustness, smoothing linear cut points, and effectively handling new data. Insights gained from the Random Forest model can guide business strategies, identifying critical factors influencing churn and informing targeted retention efforts.

On the other hand, the Boosting - Gradient Boosting method, despite achieving a misclassification rate of 0.177278 for the training set and 0.192118 for the validation set, offers additional insights into variable importance. Total Purchases emerges as the most crucial variable, followed by Location, IMP_MembershipLevel, IMP_TotalSpent, and FavoriteCategory. The presence of a significant point in the training process, indicated by a vertical line in the Subseries Plot, suggests an optimal iteration for minimizing misclassification. The Gradient Boosting model underscores the importance of variables such as Location and IMP_MembershipLevel in predicting churn. Location-specific trends and the impact of membership levels on customer behavior can inform geographically targeted campaigns and personalized loyalty programs.

Across the decision tree, Random Forest, and Gradient Boosting models, Total Purchases consistently stands out as a pivotal variable. Customers with lower total purchases are often associated with higher predicted churn probabilities. This suggests that monitoring and encouraging increased transaction frequency or higher purchase amounts could positively impact customer retention.

The decision tree identifies distinct segments based on TotalPurchases, revealing different levels of churn risk. Notably, Node 9 highlights a substantial segment with low TotalPurchases and a high predicted churn probability. Businesses can use this segmentation to tailor retention strategies, offering targeted incentives or promotions to customers in this segment.

Both Random Forest (Bagging) and Gradient Boosting (Boosting) models outperform the standalone decision tree in terms of misclassification rates. Employing ensemble methods enhances the models' predictive accuracy and robustness, providing more reliable insights into customer behavior.

Business strategies can leverage these insights by tailoring retention efforts to segments identified by the decision tree and incorporating ensemble methods for improved predictive performance. Focusing on customer segments with high predicted churn probabilities, implementing targeted promotions or personalized communication may prove effective. Additionally, monitoring Total Purchases, Location, and other influential variables highlighted by ensemble methods can aid in proactive customer engagement and satisfaction, ultimately contributing to improved customer retention.

Reflection and Learning Outcome

This case study has been a valuable learning experience, providing insights into the complexities of predictive modeling and customer behavior analysis. Through this endeavor, several key reflections and learning outcomes have emerged.

One of the significant lessons learnt has been the importance of understanding and interpreting the decision tree and ensemble method outputs. Learning to interpret the node rules, misclassification rates, and variable importance has enhanced my ability to derive actionable insights from the models. This case study highlighted the critical role of feature engineering in improving model performance. Identifying and selecting relevant variables, handling missing values, and transforming data were crucial steps in creating more robust models. The experience underscored the need for a thoughtful approach to data preparation.

The application of ensemble methods, particularly Bagging (Random Forest) and Boosting (Gradient Boosting), demonstrated their efficacy in enhancing predictive accuracy. Understanding the principles behind these methods and their ability to mitigate the limitations of individual decision trees was a valuable learning outcome. This case study provided a real-world application of predictive analytics in understanding customer behavior. The ability to translate model outputs into actionable business strategies, such as targeted retention efforts, demonstrated the practical utility of predictive modelling in decision-making.

Dealing with data challenges, including handling missing values and selecting relevant features, was a key aspect of the learning journey. Implementing effective data preprocessing techniques and understanding their impact on model outcomes was a hands-on learning experience.

In conclusion, this case study has equipped me with practical skills in applying predictive modeling techniques, interpreting model outputs, and making informed decisions based on data-driven insights. The challenges faced during the process served as valuable learning opportunities, fostering a problem-solving mindset. Moving forward, the knowledge gained from this case study will undoubtedly contribute to my proficiency in leveraging predictive analytics for business applications.

==END==