WQD7005 SEM 1 2023/24

ALTERNATIVE ASSESSMENT (AA) 1: CASE STUDY

The GitHub link for all necessary documents and components is as follows: https://github.com/unplayable98/7005AA1-22079565

Overview:

This study is to apply various tools to perform various operations to analyse a dataset regarding the customer behaviour and aim to generate insights based on the result of modelling and provide business suggestions. if possible

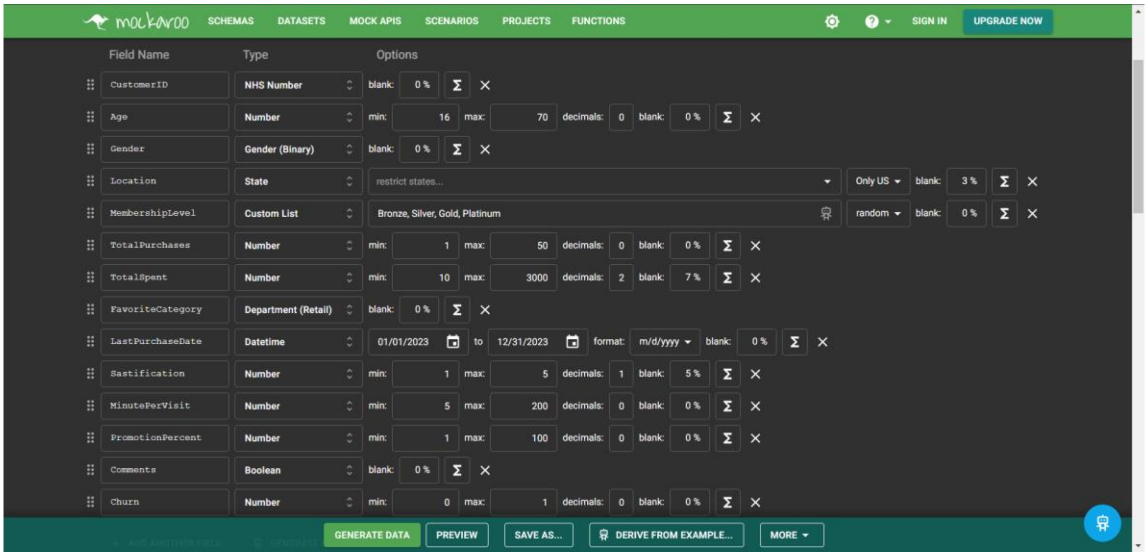
The applied tools and their roles are as follows:

Talend Data Integration (TDI): Perform data preprocessing such as joining of dataset.
Talend Data Preparation (TDP): Perform various data preprocessing to improve data quality such as data cleaning and data transformation.
SAS Enterprise Miner (SAS-EM): To perform decision tree-related modelling to analyse customer behaviours.

Dataset:

The dataset applied for this assessment is generated from an online data generator named Mockaroo (mockaroo.com). There are 1000 rows generated for this dataset, which is the maximum number of rows able to be generated on this website without a paid subscription.

While designing the data, the dataset "E-commerce Customer Behavior Dataset" by Laksika Tharmalingam on the Kaggle website is referred to. The designation of data is as follows:



Field Name	Type	Options
CustomerID	NHS Number	blank: 0 %
Age	Number	min: 16 max: 70 decimals: 0 blank: 0 %
Gender	Gender (Binary)	blank: 0 %
Location	State	restrict states... Only US blank: 3 %
MembershipLevel	Custom List	Bronze, Silver, Gold, Platinum random blank: 0 %
TotalPurchases	Number	min: 1 max: 50 decimals: 0 blank: 0 %
TotalSpent	Number	min: 10 max: 3000 decimals: 2 blank: 7 %
FavoriteCategory	Department (Retail)	blank: 0 %
LastPurchaseDate	Datetime	01/01/2023 to 12/31/2023 format: m/d/yyyy blank: 0 %
Sastification	Number	min: 1 max: 5 decimals: 1 blank: 5 %
MinutePerVisit	Number	min: 5 max: 200 decimals: 0 blank: 0 %
PromotionPercent	Number	min: 1 max: 100 decimals: 0 blank: 0 %
Comments	Boolean	blank: 0 %
Churn	Number	min: 0 max: 1 decimals: 0 blank: 0 %

Figure 1: Dataset designation.

Besides the listed columns in the question, there are four columns created in this dataset, making the total column numbers to 14. The additional columns and their description are listed as follows:

Satisfaction: The average satisfaction of the customer for every purchase. The range is from 1 (Very unsatisfied) to 5 (Satisfied)
MinutePerVisit: The time stay on the website in minutes per visit.
PromotionPercent: The proportion of items purchased during promotion.
Comments: Whether the customer has given comments after purchase before.

Noted that the columns Location, TotalSpent and Satisfaction (after renaming) are designated to have missing values for 3 to 7% of the rows.

Dataset reference:
https://www.kaggle.com/datasets/uom190346a/e-commerce-customer-behavior-dataset

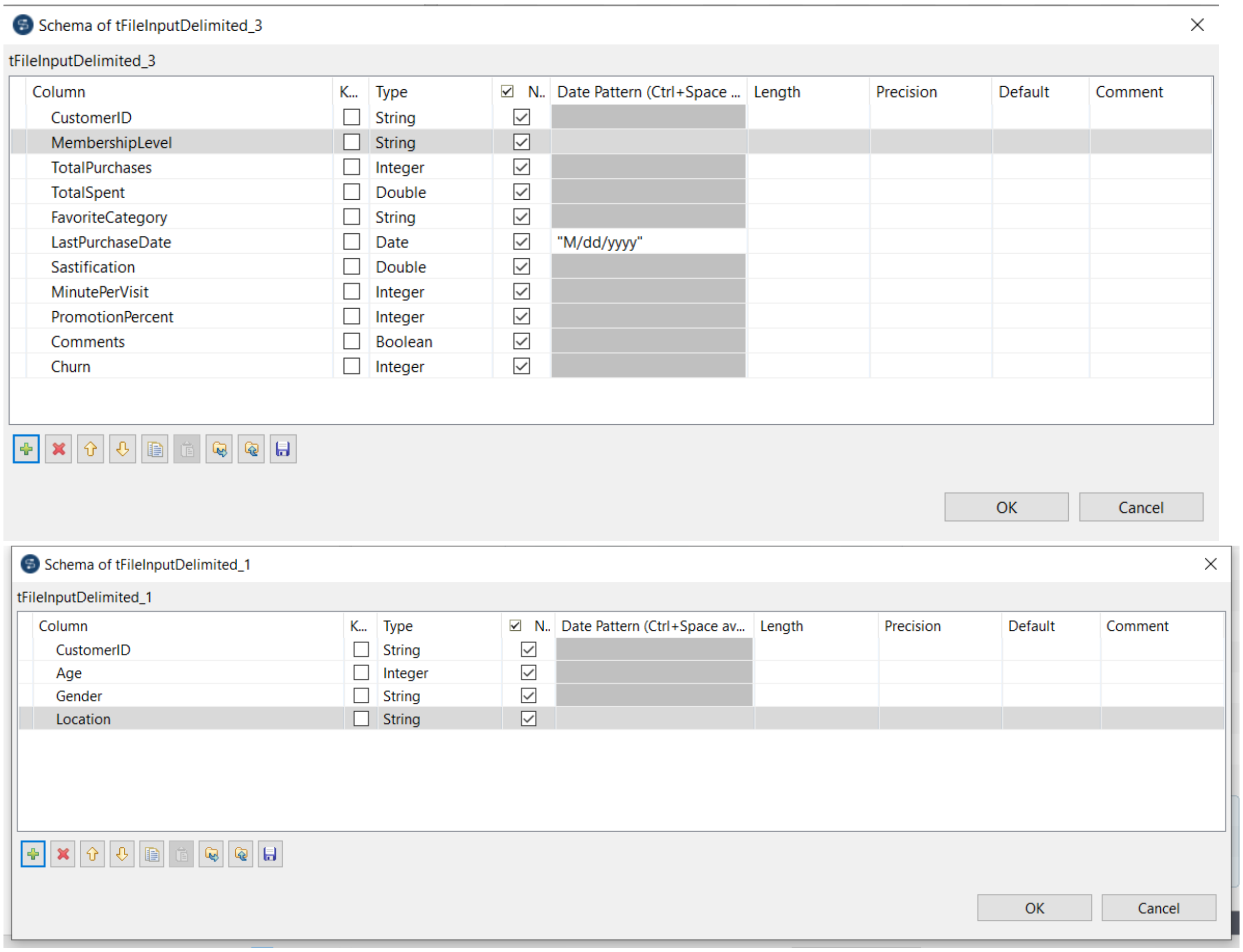
Dataset generated:
https://drive.google.com/file/d/1Q9DjruSQt2Ynp6NWz7ldTpDmym6oVMYJ/view?usp=sharing

Before entering the data, I separate a few columns from the dataset manually into a reduced dataset. The list of columns for the dataset that needed to load into the Talend Data Integration is as follows:

D7005AA1_DataA	D7005AA1_DataB
CustomerID Age Gender Location	CustomerID MembershipLevel TotalPurchases TotalSpent FavouriteCategory LastPurchaseData Satisfaction MinutePerVisit PromotionPercent Comments Churn

Data Preparation Using Talend Data Integration and Talend Data Preparation:

Before loading both datasets (D7005AA1_DataA and D7005AA1_DataB) into SAS Enterprise Miner (SAS EM) for analysis, the preprocessing of data is performed on Talend Data Integration (TDI) and Talend Data Preparation (TDP). First of all, the dataset is loaded into Talend Data Integration using the following schema:



Schema of tFileInputDelimited_3

tFileInputDelimited_3

Column	K...	Type	☑ N..	Date Pattern (Ctrl+Space ...	Length	Precision	Default	Comment
CustomerID	☐	String	☑					
MembershipLevel	☐	String	☑					
TotalPurchases	☐	Integer	☑					
TotalSpent	☐	Double	☑					
FavoriteCategory	☐	String	☑					
LastPurchaseDate	☐	Date	☑	"M/dd/yyyy"				
Sastification	☐	Double	☑					
MinutePerVisit	☐	Integer	☑					
PromotionPercent	☐	Integer	☑					
Comments	☐	Boolean	☑					
Churn	☐	Integer	☑					

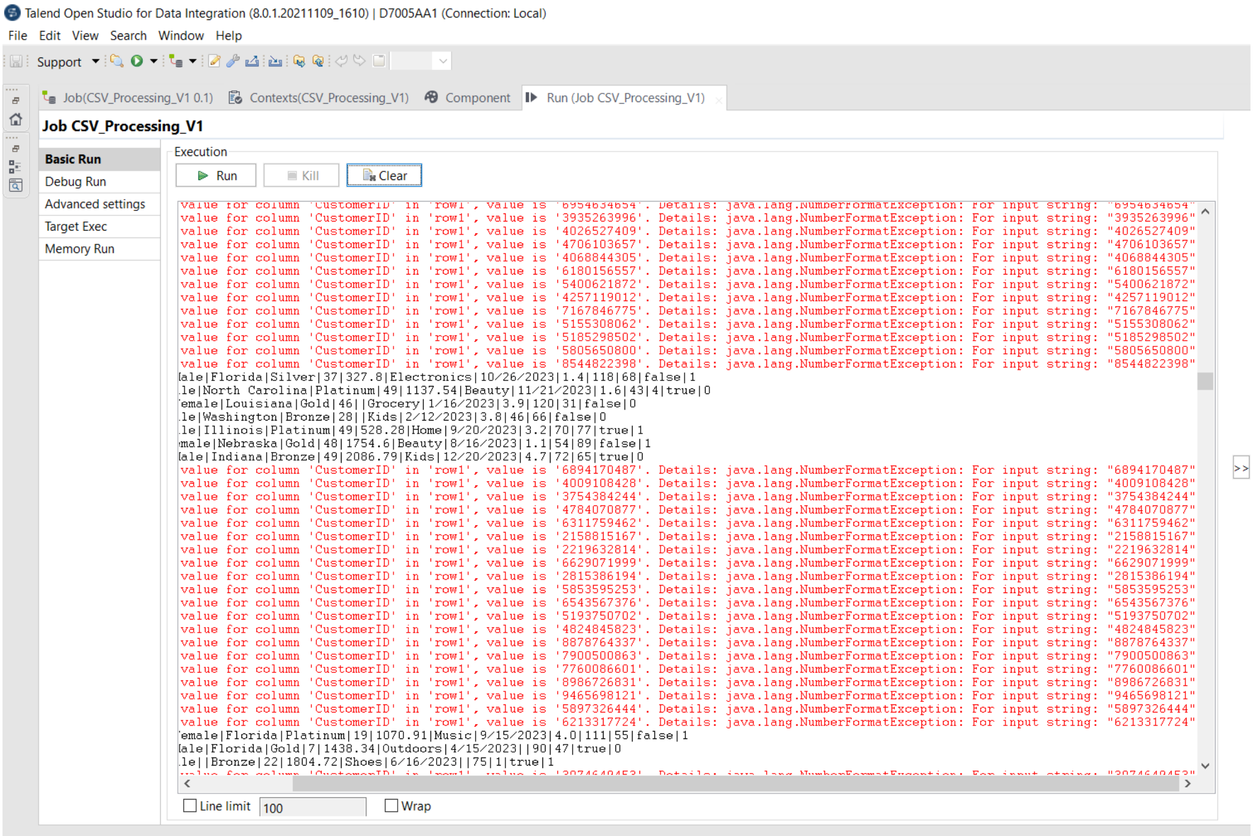
Schema of tFileInputDelimited_1

tFileInputDelimited_1

Column	K...	Type	☑ N..	Date Pattern (Ctrl+Space av...	Length	Precision	Default	Comment
CustomerID	☐	String	☑					
Age	☐	Integer	☑					
Gender	☐	String	☑					
Location	☐	String	☑					

Figure 2: Schema when loading the dataset into Talend Data Integration.

Although the CustomerID is made up of 10 to 11-digit integers, however when the data type of CustomerID is set as Integer, some errors exist as the system identifies some of the numbers as strings. As the CustomerID is only used to check for duplicates, which is not involved much in analysis, the string data type is acceptable for Customer ID. The error is shown in Figure 3.

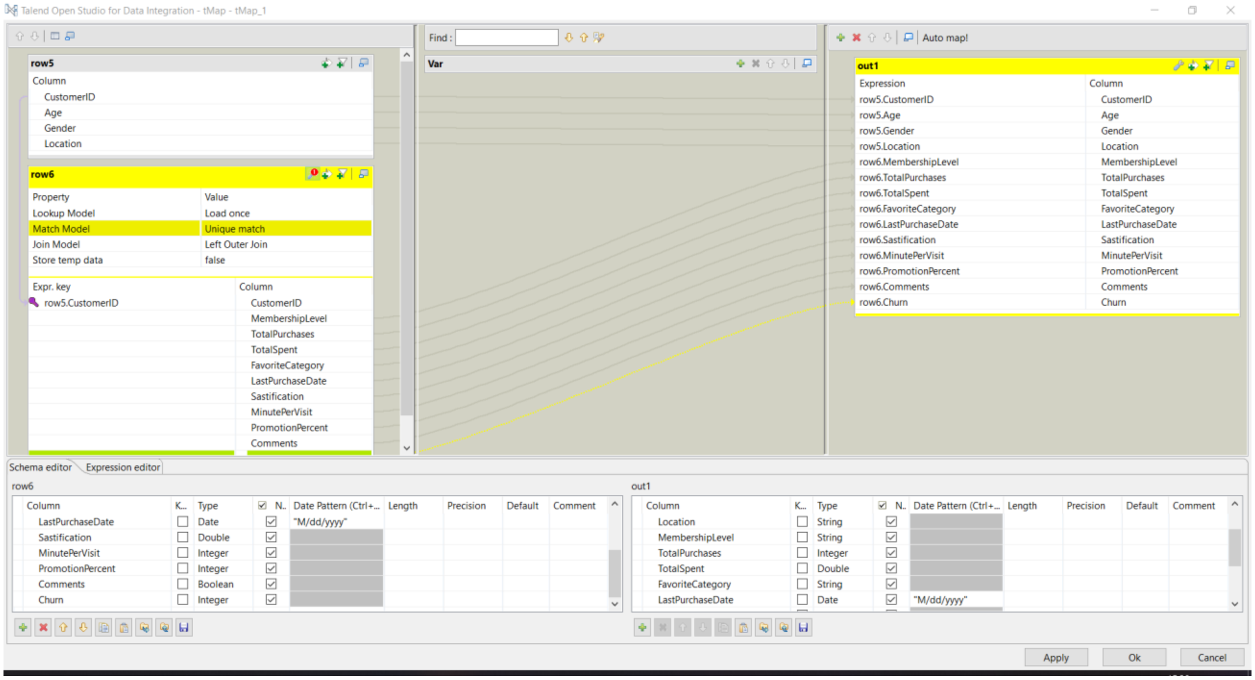

Talend Open Studio for Data Integration (8.0.1.20211109_1610) | D7005AA1 (Connection: Local)
File Edit View Search Window Help
Support
Job(CSV_Processing_V1 0.1)
Contexts(CSV_Processing_V1)
Component
Run (Job CSV_Processing_V1)
Job CSV_Processing_V1
Basic Run
Debug Run
Advanced settings
Target Exec
Memory Run
Execution
Run
Kill
Clear
>>

```
value for column 'CustomerID' in 'row1', value is '6954634654'. Details: java.lang.NumberFormatException: For input string: "6954634654"
value for column 'CustomerID' in 'row1', value is '3935263996'. Details: java.lang.NumberFormatException: For input string: "3935263996"
value for column 'CustomerID' in 'row1', value is '4026527409'. Details: java.lang.NumberFormatException: For input string: "4026527409"
value for column 'CustomerID' in 'row1', value is '4706103657'. Details: java.lang.NumberFormatException: For input string: "4706103657"
value for column 'CustomerID' in 'row1', value is '4068844305'. Details: java.lang.NumberFormatException: For input string: "4068844305"
value for column 'CustomerID' in 'row1', value is '6180156557'. Details: java.lang.NumberFormatException: For input string: "6180156557"
value for column 'CustomerID' in 'row1', value is '5400621872'. Details: java.lang.NumberFormatException: For input string: "5400621872"
value for column 'CustomerID' in 'row1', value is '4257119012'. Details: java.lang.NumberFormatException: For input string: "4257119012"
value for column 'CustomerID' in 'row1', value is '7167846775'. Details: java.lang.NumberFormatException: For input string: "7167846775"
value for column 'CustomerID' in 'row1', value is '5155308062'. Details: java.lang.NumberFormatException: For input string: "5155308062"
value for column 'CustomerID' in 'row1', value is '5185298502'. Details: java.lang.NumberFormatException: For input string: "5185298502"
value for column 'CustomerID' in 'row1', value is '5805650800'. Details: java.lang.NumberFormatException: For input string: "5805650800"
value for column 'CustomerID' in 'row1', value is '8544822398'. Details: java.lang.NumberFormatException: For input string: "8544822398"
ale|Florida|Silver|37|327.8|Electronics|10/26/2023|1.4|118|68|false|1
le|North Carolina|Platinum|49|1137.54|Beauty|11/21/2023|1.6|43|4|true|0
emale|Louisiana|Gold|46||Grocery|1/16/2023|3.9|120|31|false|0
le|Washington|Bronze|28||Kids|2/12/2023|3.8|46|66|false|0
le|Illinois|Platinum|49|528.28|Home|9/20/2023|3.2|70|77|true|1
male|Nebraska|Gold|48|1754.6|Beauty|8/16/2023|1.1|54|89|false|1
ale|Indiana|Bronze|49|2086.79|Kids|12/20/2023|4.7|72|65|true|0
value for column 'CustomerID' in 'row1', value is '6894170487'. Details: java.lang.NumberFormatException: For input string: "6894170487"
value for column 'CustomerID' in 'row1', value is '4009108428'. Details: java.lang.NumberFormatException: For input string: "4009108428"
value for column 'CustomerID' in 'row1', value is '3754384244'. Details: java.lang.NumberFormatException: For input string: "3754384244"
value for column 'CustomerID' in 'row1', value is '4784070877'. Details: java.lang.NumberFormatException: For input string: "4784070877"
value for column 'CustomerID' in 'row1', value is '6311759462'. Details: java.lang.NumberFormatException: For input string: "6311759462"
value for column 'CustomerID' in 'row1', value is '2158815167'. Details: java.lang.NumberFormatException: For input string: "2158815167"
value for column 'CustomerID' in 'row1', value is '2219632814'. Details: java.lang.NumberFormatException: For input string: "2219632814"
value for column 'CustomerID' in 'row1', value is '6629071999'. Details: java.lang.NumberFormatException: For input string: "6629071999"
value for column 'CustomerID' in 'row1', value is '2815386194'. Details: java.lang.NumberFormatException: For input string: "2815386194"
value for column 'CustomerID' in 'row1', value is '5853595253'. Details: java.lang.NumberFormatException: For input string: "5853595253"
value for column 'CustomerID' in 'row1', value is '6543567376'. Details: java.lang.NumberFormatException: For input string: "6543567376"
value for column 'CustomerID' in 'row1', value is '5193750702'. Details: java.lang.NumberFormatException: For input string: "5193750702"
value for column 'CustomerID' in 'row1', value is '4824845823'. Details: java.lang.NumberFormatException: For input string: "4824845823"
value for column 'CustomerID' in 'row1', value is '8878764337'. Details: java.lang.NumberFormatException: For input string: "8878764337"
value for column 'CustomerID' in 'row1', value is '7900500863'. Details: java.lang.NumberFormatException: For input string: "7900500863"
value for column 'CustomerID' in 'row1', value is '7760086601'. Details: java.lang.NumberFormatException: For input string: "7760086601"
value for column 'CustomerID' in 'row1', value is '8986726831'. Details: java.lang.NumberFormatException: For input string: "8986726831"
value for column 'CustomerID' in 'row1', value is '9465698121'. Details: java.lang.NumberFormatException: For input string: "9465698121"
value for column 'CustomerID' in 'row1', value is '5897326444'. Details: java.lang.NumberFormatException: For input string: "5897326444"
value for column 'CustomerID' in 'row1', value is '6213317724'. Details: java.lang.NumberFormatException: For input string: "6213317724"
emale|Florida|Platinum|19|1070.91|Music|9/15/2023|4.0|111|55|false|1
ale|Florida|Gold|7|1438.34|Outdoors|4/15/2023||90|47|true|0
le||Bronze|22|1804.72|Shoes|6/16/2023||75|1|true|1
```

☐ Line limit 100 ☐ Wrap

Figure 3: Error when loading the dataset to TDI when CustomerID type is set as integer.

First of all, both datasets are related so they needed to be joined together. The tLogRow component provides an output that enables us to determine whether the data under processing is correct. The tMap is used to join both datasets.



row5

Column
CustomerID
Age
Gender
Location

row6

Property	Value
Lookup Model	Load once
Match Model	Unique match
Join Model	Left Outer Join
Store temp data	false

Expr. key	Column
row5.CustomerID	CustomerID
	MembershipLevel
	TotalPurchases
	TotalSpent
	FavoriteCategory
	LastPurchaseDate
	Sastification
	MinutePerVisit
	PromotionPercent
	Comments

out1

Expression	Column
row5.CustomerID	CustomerID
row5.Age	Age
row5.Gender	Gender
row5.Location	Location
row6.MembershipLevel	MembershipLevel
row6.TotalPurchases	TotalPurchases
row6.TotalSpent	TotalSpent
row6.FavoriteCategory	FavoriteCategory
row6.LastPurchaseDate	LastPurchaseDate
row6.Sastification	Sastification
row6.MinutePerVisit	MinutePerVisit
row6.PromotionPercent	PromotionPercent
row6.Comments	Comments
row6.Churn	Churn

row6

Column	K...	Type	☑ N..	Date Pattern (Ctrl+...	Length	Precision	Default	Comment
LastPurchaseDate	☐	Date	☑	"M/dd/yyyy"				
Sastification	☐	Double	☑					
MinutePerVisit	☐	Integer	☑					
PromotionPercent	☐	Integer	☑					
Comments	☐	Boolean	☑					
Churn	☐	Integer	☑					

out1

Column	K...	Type	☑ N..	Date Pattern (Ctrl+...	Length	Precision	Default	Comment
Location	☐	String	☑					
MembershipLevel	☐	String	☑					
TotalPurchases	☐	Integer	☑					
TotalSpent	☐	Double	☑					
FavoriteCategory	☐	String	☑					
LastPurchaseDate	☐	Date	☑	"M/dd/yyyy"				

Figure 4: Setting of tMap.

Figure 4 shows both datasets are joined together based on the unique CustomerID match. After the output schema is determined, the input column is linked to the desired output column.

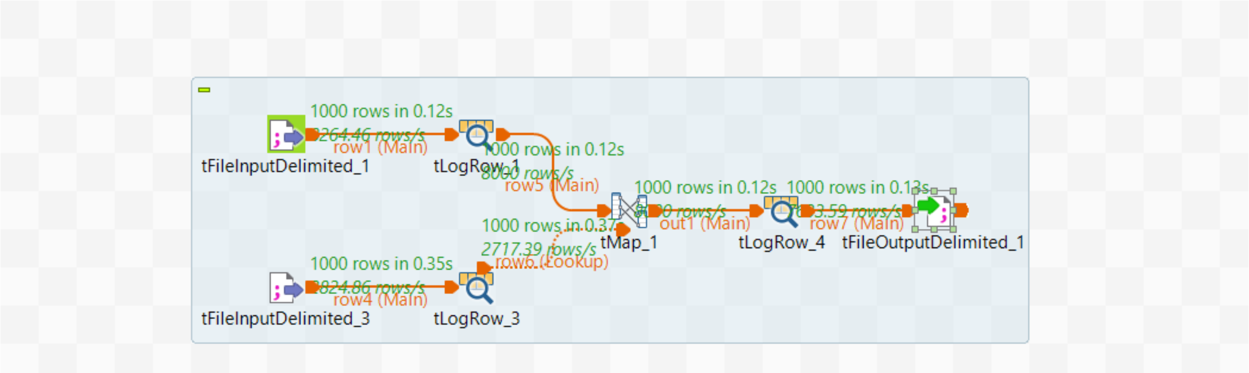
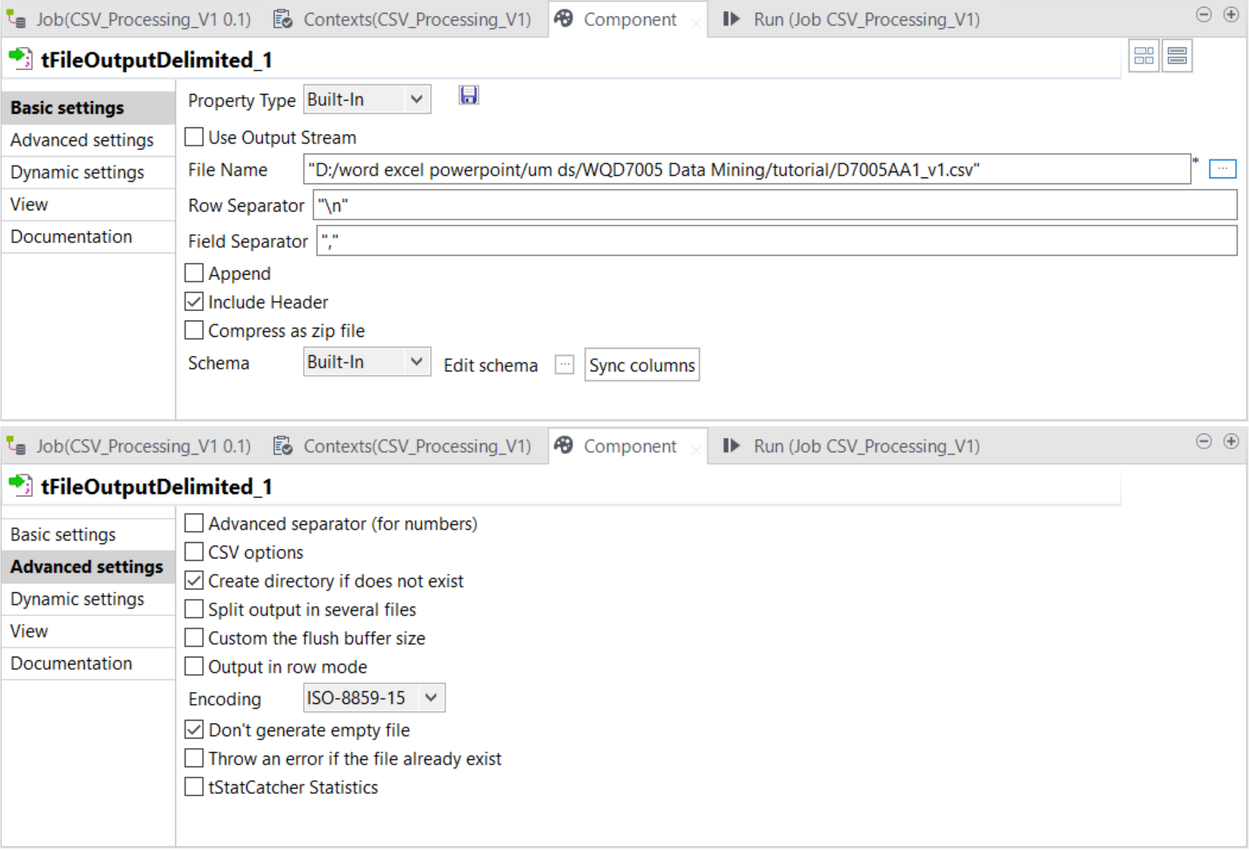


Figure 5: Workflow of producing combined CSV dataset.

As shown in Figure 5, after the datasets are joined, the combined dataset is now ready to be exported for further usage. To ensure smoother loading of data in the further step, the settings as shown in Figure 6 are applied.


Job(CSV_Processing_V1 0.1)
Contexts(CSV_Processing_V1)
Component
Run (Job CSV_Processing_V1)

tFileOutputDelimited_1

Basic settings
Advanced settings
Dynamic settings
View
Documentation

Property Type: Built-In
□ Use Output Stream
File Name: "D:/word excel powerpoint/um ds/WQD7005 Data Mining/tutorial/D7005AA1_v1.csv"
Row Separator: "\n"
Field Separator: ","
□ Append
☑ Include Header
□ Compress as zip file
Schema: Built-In
Edit schema
Sync columns

Job(CSV_Processing_V1 0.1)
Contexts(CSV_Processing_V1)
Component
Run (Job CSV_Processing_V1)

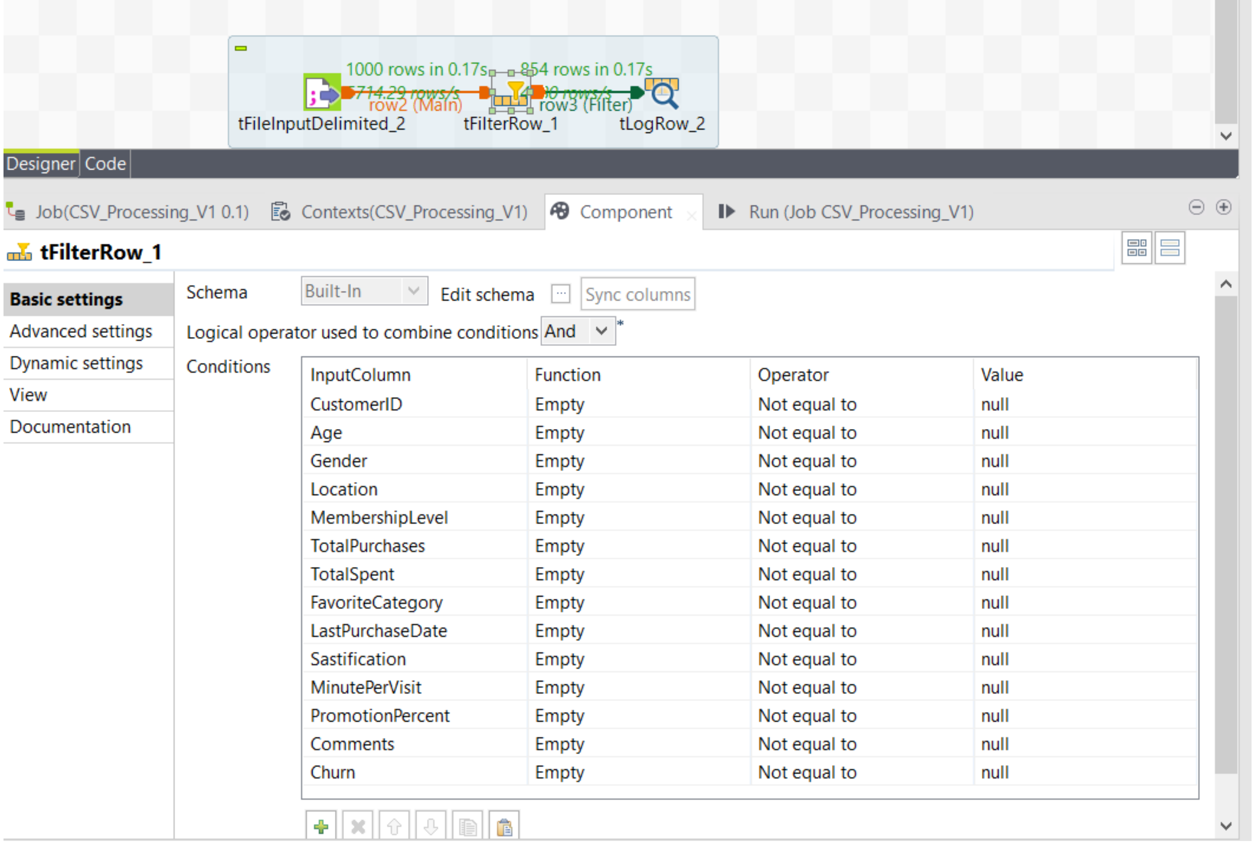
tFileOutputDelimited_1

Basic settings
Advanced settings
Dynamic settings
View
Documentation

□ Advanced separator (for numbers)
□ CSV options
☑ Create directory if does not exist
□ Split output in several files
□ Custom the flush buffer size
□ Output in row mode
Encoding: ISO-8859-15
☑ Don't generate empty file
□ Throw an error if the file already exist
□ tStatCatcher Statistics

Figure 6: Setting of export of combined CSV dataset.

The newly exported dataset is now able to be used. After loading the combined dataset into TDI, the first attempt of handling missing values is operated, which is dropping all the rows with missing values. By using the tFilterRow component with the setting as shown in Figure 7, the remaining number of rows is shown.



tFilterRow_1

Schema: Built-In Edit schema Sync columns

Logical operator used to combine conditions: And *

Conditions

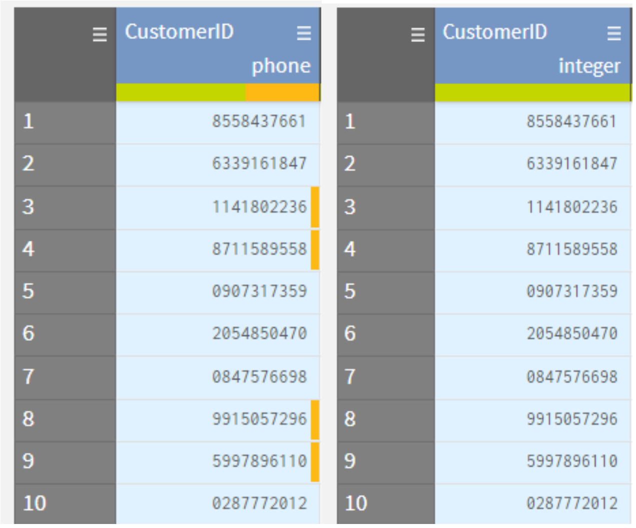
InputColumn	Function	Operator	Value
CustomerID	Empty	Not equal to	null
Age	Empty	Not equal to	null
Gender	Empty	Not equal to	null
Location	Empty	Not equal to	null
MembershipLevel	Empty	Not equal to	null
TotalPurchases	Empty	Not equal to	null
TotalSpent	Empty	Not equal to	null
FavoriteCategory	Empty	Not equal to	null
LastPurchaseDate	Empty	Not equal to	null
Sastification	Empty	Not equal to	null
MinutePerVisit	Empty	Not equal to	null
PromotionPercent	Empty	Not equal to	null
Comments	Empty	Not equal to	null
Churn	Empty	Not equal to	null

Figure 7: Filtering out all rows with missing values.

As shown in the figure above, only 854 rows are remaining in the dataset, which means 146 records (14.6%) are dropped. As the proportion of rows with missing values is not small enough to be removed, data imputation seems to be the better option to handle missing values.

The data preprocessing is now moved to Talend Data Preparation (TDP) as TDP provide an easier yet robust way to process the data.

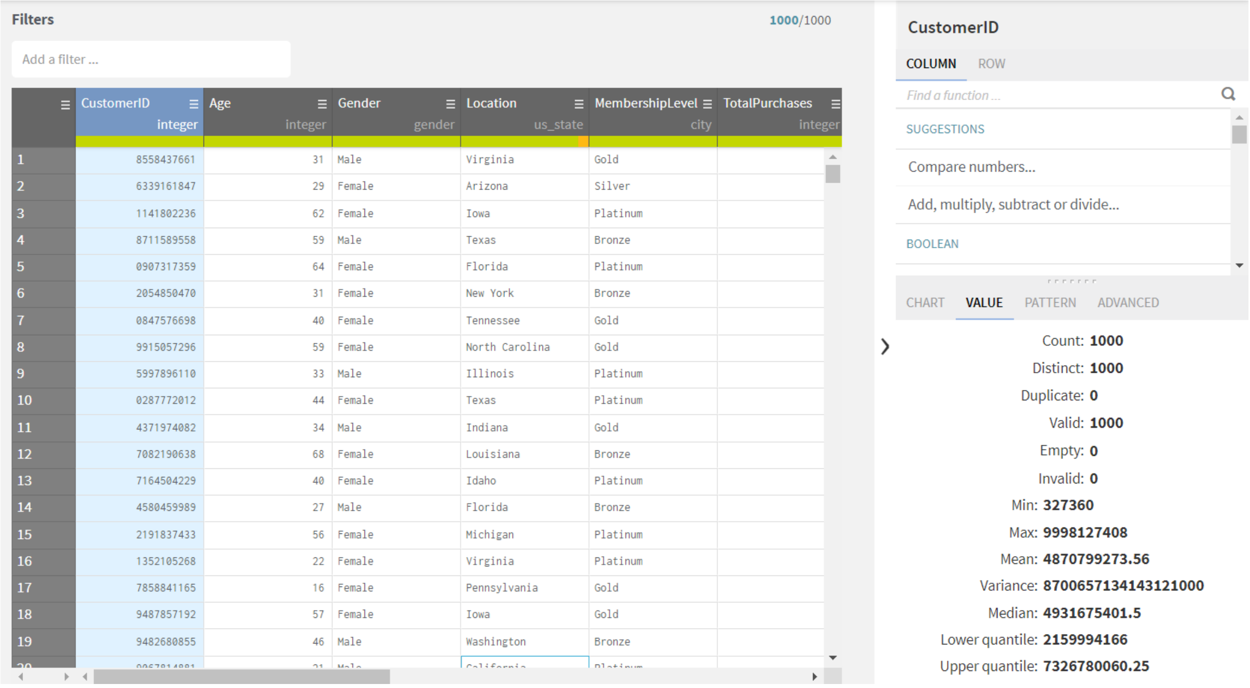
After importing the dataset to TDP, the data health bar for every column is inspected. The first step is to change the data type of CustomerID to Integer from Phone Number to eliminate the invalid data issues questioned by the tool. Note that TDP accepts all CustomerID as integers, unlike TDI, indicating the CustomerID values from the original dataset might not have any issues with the data type.



	CustomerID phone
1	8558437661
2	6339161847
3	1141802236
4	8711589558
5	0907317359
6	2054850470
7	0847576698
8	9915057296
9	5997896110
10	0287772012

	CustomerID integer
1	8558437661
2	6339161847
3	1141802236
4	8711589558
5	0907317359
6	2054850470
7	0847576698
8	9915057296
9	5997896110
10	0287772012

Figure 8: Data health increased to perfect green after changing the data type of CustomerID.



Filters

Add a filter ...

1000/1000

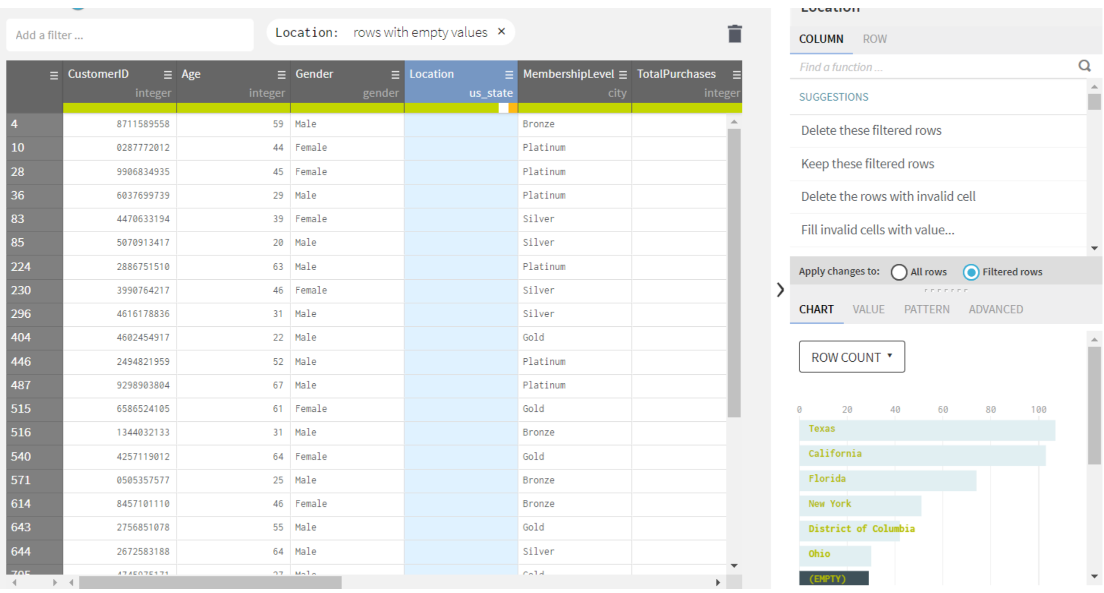
	CustomerID integer	Age integer	Gender gender	Location us_state	MembershipLevel city	TotalPurchases integer
1	8558437661	31	Male	Virginia	Gold	
2	6339161847	29	Female	Arizona	Silver	
3	1141802236	62	Female	Iowa	Platinum	
4	8711589558	59	Male	Texas	Bronze	
5	0907317359	64	Female	Florida	Platinum	
6	2054850470	31	Female	New York	Bronze	
7	0847576698	40	Female	Tennessee	Gold	
8	9915057296	59	Female	North Carolina	Gold	
9	5997896110	33	Male	Illinois	Platinum	
10	0287772012	44	Female	Texas	Platinum	
11	4371974082	34	Male	Indiana	Gold	
12	7082190638	68	Female	Louisiana	Bronze	
13	7164504229	40	Female	Idaho	Platinum	
14	4580459989	27	Male	Florida	Bronze	
15	2191837433	56	Female	Michigan	Platinum	
16	1352105268	22	Female	Virginia	Platinum	
17	7858841165	16	Female	Pennsylvania	Gold	
18	9487857192	57	Female	Iowa	Gold	
19	9482680855	46	Male	Washington	Bronze	

CustomerID

Count: 1000
Distinct: 1000
Duplicate: 0
Valid: 1000
Empty: 0
Invalid: 0
Min: 327360
Max: 9998127408
Mean: 4870799273.56
Variance: 8700657134143121000
Median: 4931675401.5
Lower quantile: 2159994166
Upper quantile: 7326780060.25

Figure 9: The right plane shows there is no duplicate CustomerID.

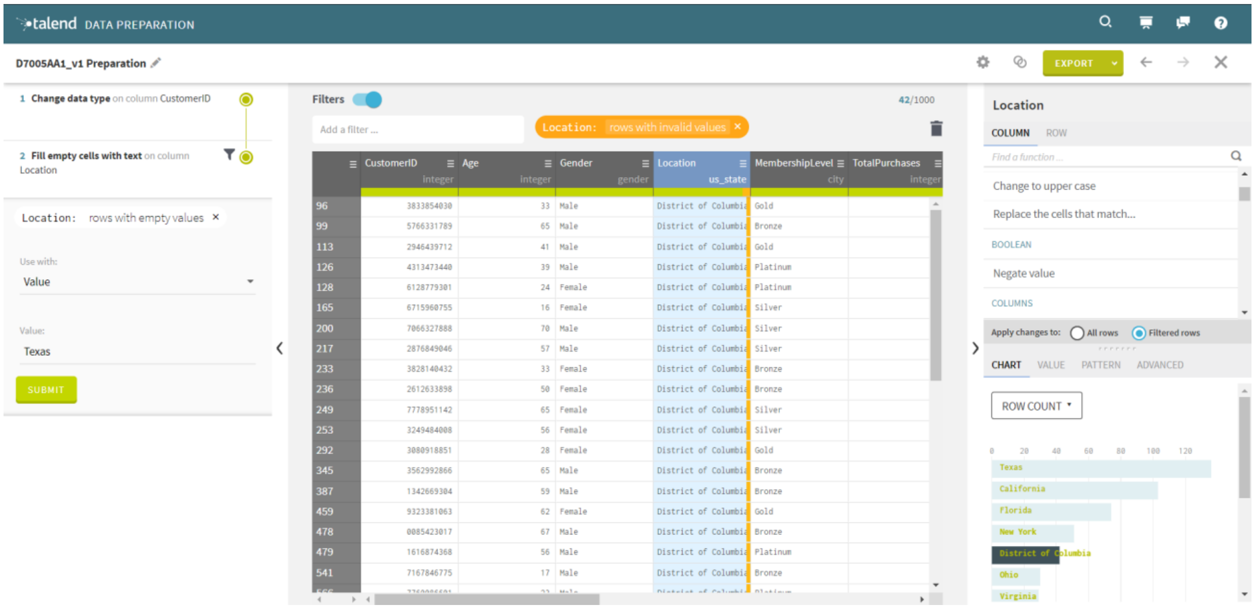
As shown in the output of Figure 9, as the duplicate check is performed, no duplicates are occurring at the CustomerID column which every row should have unique values. Hence, there is no action taken for the removal of duplicate data.



	CustomerID integer	Age integer	Gender gender	Location us_state	MembershipLevel city	TotalPurchases integer
4	8711589558	59	Male		Bronze	
10	0287772012	44	Female		Platinum	
28	9906834935	45	Female		Platinum	
36	6037699739	29	Male		Platinum	
83	4470633194	39	Female		Silver	
85	5070913417	20	Male		Silver	
224	2886751510	63	Male		Platinum	
230	3990764217	46	Female		Silver	
296	4616178836	31	Male		Silver	
404	4602454917	22	Male		Gold	
446	2494821959	52	Male		Platinum	
487	9298903804	67	Male		Platinum	
515	6586524105	61	Female		Gold	
516	1344032133	31	Male		Bronze	
540	4257119012	64	Female		Gold	
571	0505357577	25	Male		Bronze	
614	8457101110	46	Female		Bronze	
643	2756851078	55	Male		Gold	
644	2672583188	64	Male		Silver	

Figure 10: Handling missing values for Location.

Referring to Figure 10, there are missing values in location as designated during data generation. For this column, we select the state with the highest occurrence to perform data imputation for missing locations. The way to impute the data is shown in the left pane of Figure 8.



D7005AA1_v1 Preparation

1 **Change data type** on column CustomerID

2 **Fill empty cells with text** on column Location

Location: rows with empty values ×

Use with: Value

Value: Texas

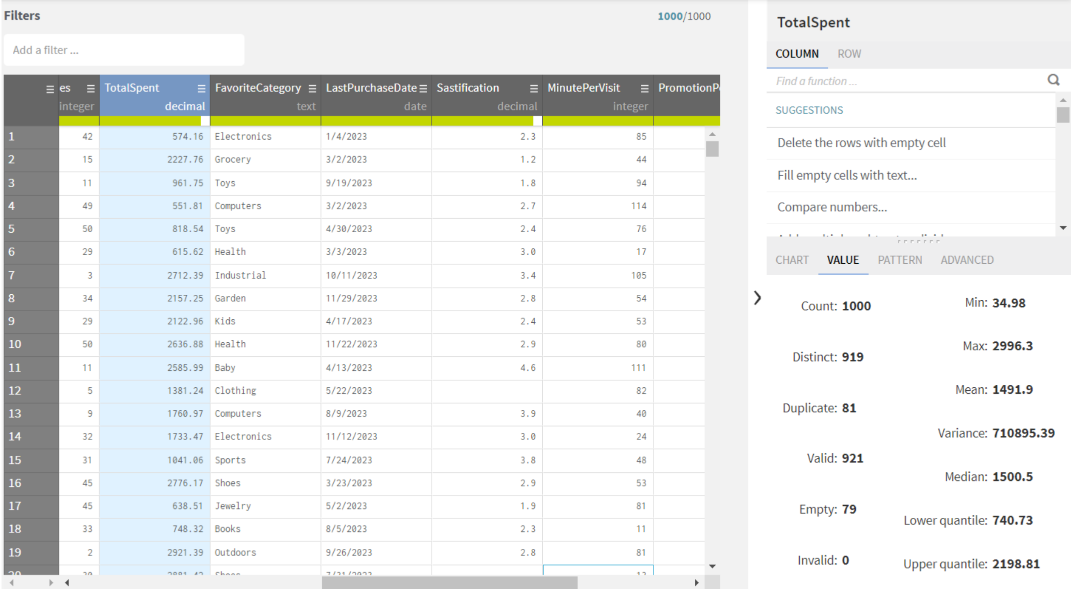
SUBMIT

42/1000

	CustomerID integer	Age integer	Gender gender	Location us_state	MembershipLevel city	TotalPurchases integer
96	3833854030	33	Male	District of Columbia	Gold	
99	5766331789	65	Male	District of Columbia	Bronze	
113	2946439712	41	Male	District of Columbia	Gold	
126	4313473440	39	Male	District of Columbia	Platinum	
128	6128779301	24	Female	District of Columbia	Platinum	
165	6715960755	16	Female	District of Columbia	Silver	
200	7066327888	70	Male	District of Columbia	Silver	
217	2876849046	57	Male	District of Columbia	Silver	
233	3828140432	33	Female	District of Columbia	Bronze	
236	2612633898	50	Female	District of Columbia	Bronze	
249	7778951142	65	Female	District of Columbia	Silver	
253	3249484008	56	Female	District of Columbia	Silver	
292	3080918851	28	Female	District of Columbia	Gold	
345	3562992866	65	Male	District of Columbia	Bronze	
387	1342669304	59	Male	District of Columbia	Bronze	
459	9323381063	62	Female	District of Columbia	Gold	
478	0085423017	67	Male	District of Columbia	Bronze	
479	1616874368	56	Male	District of Columbia	Platinum	
541	7167846775	17	Male	District of Columbia	Bronze	

Figure 11: Location imputation and potential invalid location.

Besides missing values, there exist invalid values too. However, these values are the same, which is the District of Columbia, which is not a US state, but a federal district of the United States which not belong to any state. Hence, it is safe to ignore this flag as it is still an area of the United States.



Filters 1000/1000

Add a filter ...

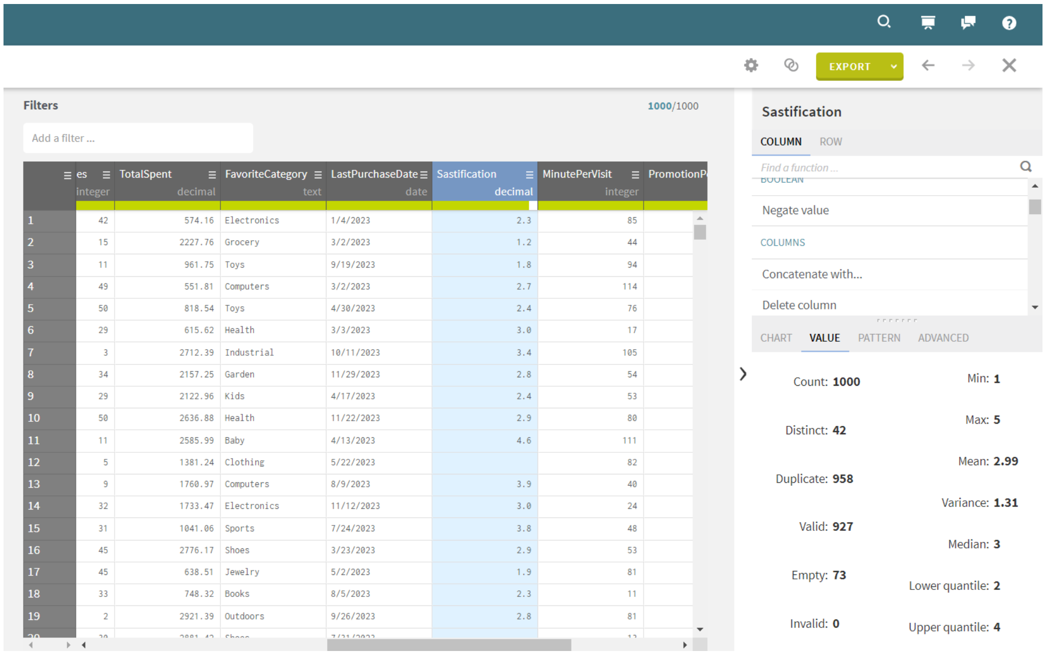
	es integer	TotalSpent decimal	FavoriteCategory text	LastPurchaseDate date	Sastification decimal	MinutePerVisit integer	PromotionP
1	42	574.16	Electronics	1/4/2023	2.3	85	
2	15	2227.76	Grocery	3/2/2023	1.2	44	
3	11	961.75	Toys	9/19/2023	1.8	94	
4	49	551.81	Computers	3/2/2023	2.7	114	
5	50	818.54	Toys	4/30/2023	2.4	76	
6	29	615.62	Health	3/3/2023	3.0	17	
7	3	2712.39	Industrial	10/11/2023	3.4	105	
8	34	2157.25	Garden	11/29/2023	2.8	54	
9	29	2122.96	Kids	4/17/2023	2.4	53	
10	50	2636.88	Health	11/22/2023	2.9	80	
11	11	2585.99	Baby	4/13/2023	4.6	111	
12	5	1381.24	Clothing	5/22/2023		82	
13	9	1760.97	Computers	8/9/2023	3.9	40	
14	32	1733.47	Electronics	11/12/2023	3.0	24	
15	31	1041.06	Sports	7/24/2023	3.8	48	
16	45	2776.17	Shoes	3/23/2023	2.9	53	
17	45	638.51	Jewelry	5/2/2023	1.9	81	
18	33	748.32	Books	8/5/2023	2.3	11	
19	2	2921.39	Outdoors	9/26/2023	2.8	81	

TotalSpent

Count: 1000
Distinct: 919
Duplicate: 81
Valid: 921
Empty: 79
Invalid: 0
Min: 34.98
Max: 2996.3
Mean: 1491.9
Variance: 710895.39
Median: 1500.5
Lower quantile: 740.73
Upper quantile: 2198.81

Figure 12: Deciding the imputation method of missing TotalSpent values.

For the TotalSpent, since the range of the value is large (34.98 to 2996.3), it is safer to select the median as the imputation of missing values to minimise the effect and bias caused by the potential outliers although the boxplot does not significantly skew to either side.



Filters 1000/1000

Add a filter ...

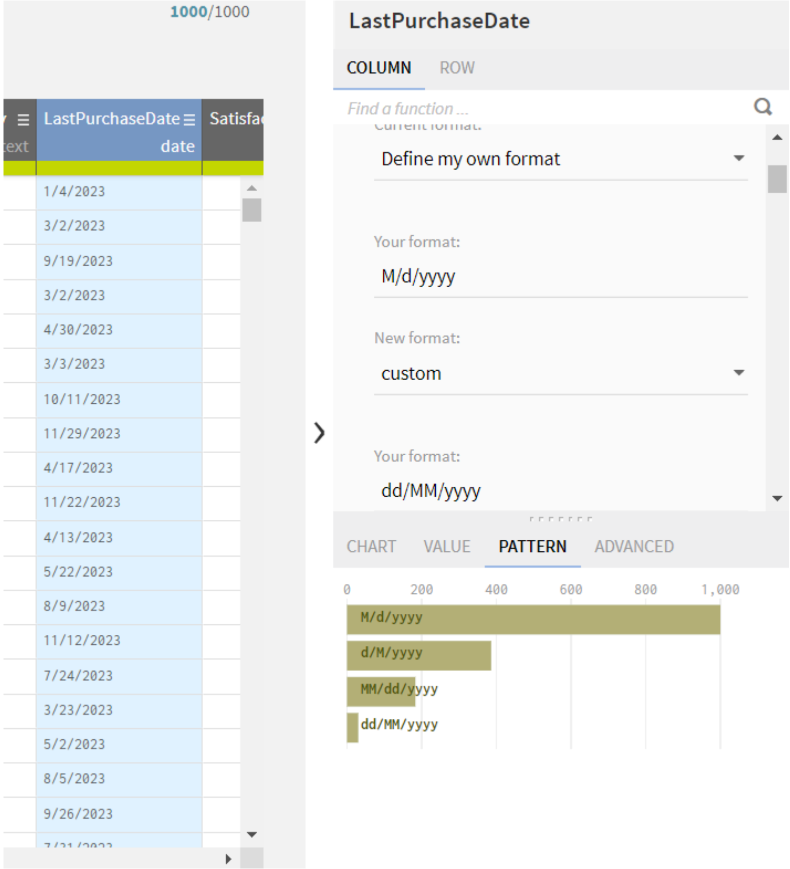
	es integer	TotalSpent decimal	FavoriteCategory text	LastPurchaseDate date	Sastification decimal	MinutePerVisit integer	PromotionP
1	42	574.16	Electronics	1/4/2023	2.3	85	
2	15	2227.76	Grocery	3/2/2023	1.2	44	
3	11	961.75	Toys	9/19/2023	1.8	94	
4	49	551.81	Computers	3/2/2023	2.7	114	
5	50	818.54	Toys	4/30/2023	2.4	76	
6	29	615.62	Health	3/3/2023	3.0	17	
7	3	2712.39	Industrial	10/11/2023	3.4	105	
8	34	2157.25	Garden	11/29/2023	2.8	54	
9	29	2122.96	Kids	4/17/2023	2.4	53	
10	50	2636.88	Health	11/22/2023	2.9	80	
11	11	2585.99	Baby	4/13/2023	4.6	111	
12	5	1381.24	Clothing	5/22/2023		82	
13	9	1760.97	Computers	8/9/2023	3.9	40	
14	32	1733.47	Electronics	11/12/2023	3.0	24	
15	31	1041.06	Sports	7/24/2023	3.8	48	
16	45	2776.17	Shoes	3/23/2023	2.9	53	
17	45	638.51	Jewelry	5/2/2023	1.9	81	
18	33	748.32	Books	8/5/2023	2.3	11	
19	2	2921.39	Outdoors	9/26/2023	2.8	81	

Sastification

Count: 1000
Distinct: 42
Duplicate: 958
Valid: 927
Empty: 73
Invalid: 0
Min: 1
Max: 5
Mean: 2.99
Variance: 1.31
Median: 3
Lower quantile: 2
Upper quantile: 4

Figure 13: Deciding the imputation method of missing Satisfaction values.

Regarding missing values of Satisfaction, the mean and median values are nearly identical which indicates that the distribution of values is possibly evenly distributed and both values are located at the middle of the range. Hence, the mean can be selected to be imputed. However, to ensure data formalisation which allows only one decimal place for input, the mean value of 2.99 is rounded off to 3.0 for imputation of missing values.



LastPurchaseDate
1/4/2023
3/2/2023
9/19/2023
3/2/2023
4/30/2023
3/3/2023
10/11/2023
11/29/2023
4/17/2023
11/22/2023
4/13/2023
5/22/2023
8/9/2023
11/12/2023
7/24/2023
3/23/2023
5/2/2023
8/5/2023
9/26/2023

Current format: Define my own format

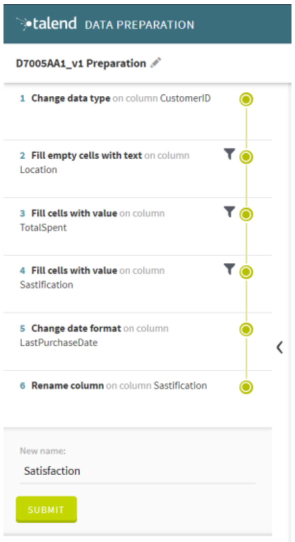
Your format: M/d/yyyy

New format: custom

Your format: dd/MM/yyyy

Figure 14: Transforming data format.

As this is a US-related dataset, the date format is recorded in American style. To change the date format to a format that is used more frequently in Malaysia, choose the “Change date format” at the right pane of the interface and apply the setting in Figure 14.



D7005AA1_v1 Preparation

1 Change data type on column CustomerID

2 Fill empty cells with text on column Location

3 Fill cells with value on column TotalSpent

4 Fill cells with value on column Sastification

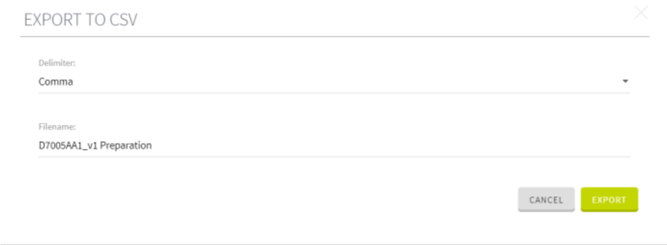
5 Change date format on column LastPurchaseDate

6 Rename column on column Sastification

New name: Satisfaction

Figure 15: Data processing workflow on Talend Data Preparation.

Finally, rename the unsatisfied column name (for example, wrong spelling) during the final check. As the data quality of all columns is satisfied, the processed dataset is exported into CSV format to the local environment as shown in Figure 16.


EXPORT TO CSV

Delimiter: Comma

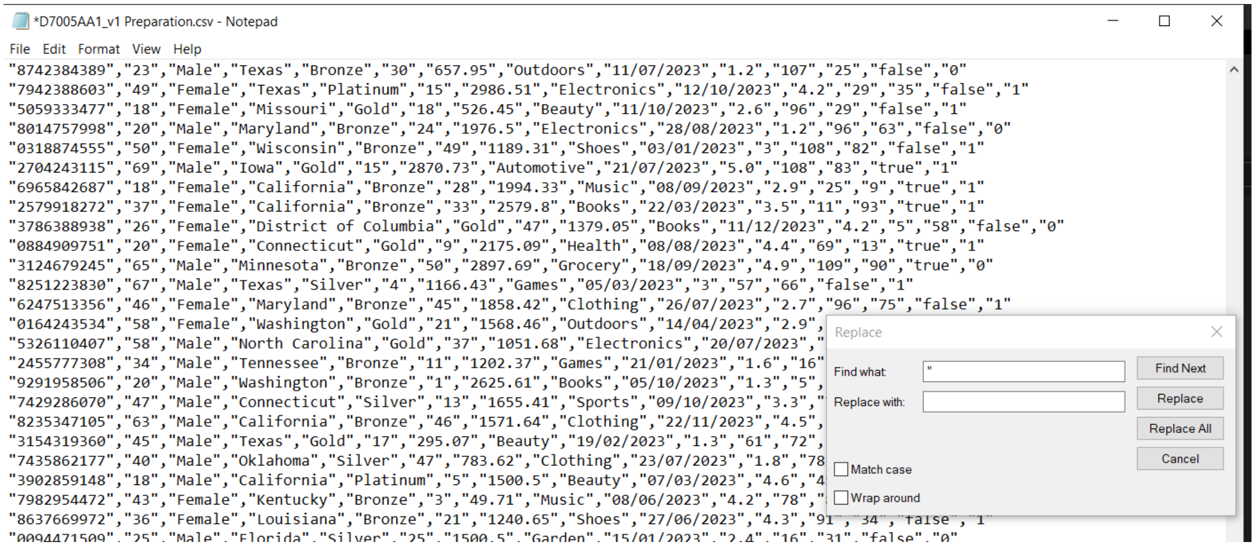
Filename: D7005AA1_v1 Preparation

Figure 16: Exporting processed dataset.

Modelling using SAS Enterprise Miner

After a new diagram in a new project is created, the File Import node is placed into the diagram. This node is to import the processed dataset exported from the Talend Data Preparation.

Before that, open the CSV file with Notepad, notice that all values are covered with quotation marks. Hence, we can remove the quotation marks by using the Replace function as shown in Figure 17 below. This step is to ensure the data can be identified as Interval level when setting the schema of the dataset later in Figure 18.



*D7005AA1_v1 Preparation.csv - Notepad

```
"8742384389","23","Male","Texas","Bronze","30","657.95","Outdoors","11/07/2023","1.2","107","25","false","0"
"7942388603","49","Female","Texas","Platinum","15","2986.51","Electronics","12/10/2023","4.2","29","35","false","1"
"5059333477","18","Female","Missouri","Gold","18","526.45","Beauty","11/10/2023","2.6","96","29","false","1"
"8014757998","20","Male","Maryland","Bronze","24","1976.5","Electronics","28/08/2023","1.2","96","63","false","0"
"0318874555","50","Female","Wisconsin","Bronze","49","1189.31","Shoes","03/01/2023","3","108","82","false","1"
"2704243115","69","Male","Iowa","Gold","15","2870.73","Automotive","21/07/2023","5.0","108","83","true","1"
"6965842687","18","Female","California","Bronze","28","1994.33","Music","08/09/2023","2.9","25","9","true","1"
"2579918272","37","Female","California","Bronze","33","2579.8","Books","22/03/2023","3.5","11","93","true","1"
"3786388938","26","Female","District of Columbia","Gold","47","1379.05","Books","11/12/2023","4.2","5","58","false","0"
"0884909751","20","Female","Connecticut","Gold","9","2175.09","Health","08/08/2023","4.4","69","13","true","1"
"3124679245","65","Male","Minnesota","Bronze","50","2897.69","Grocery","18/09/2023","4.9","109","90","true","0"
"8251223830","67","Male","Texas","Silver","4","1166.43","Games","05/03/2023","3","57","66","false","1"
"6247513356","46","Female","Maryland","Bronze","45","1858.42","Clothing","26/07/2023","2.7","96","75","false","1"
"0164243534","58","Female","Washington","Gold","21","1568.46","Outdoors","14/04/2023","2.9",
"5326110407","58","Male","North Carolina","Gold","37","1051.68","Electronics","20/07/2023","
"2455777308","34","Male","Tennessee","Bronze","11","1202.37","Games","21/01/2023","1.6","16"
"9291958506","20","Male","Washington","Bronze","1","2625.61","Books","05/10/2023","1.3","5",
"7429286070","47","Male","Connecticut","Silver","13","1655.41","Sports","09/10/2023","3.3","
"8235347105","63","Male","California","Bronze","46","1571.64","Clothing","22/11/2023","4.5",
"3154319360","45","Male","Texas","Gold","17","295.07","Beauty","19/02/2023","1.3","61","72",
"7435862177","40","Male","Oklahoma","Silver","47","783.62","Clothing","23/07/2023","1.8","78
"3902859148","18","Male","California","Platinum","5","1500.5","Beauty","07/03/2023","4.6","4
"7982954472","43","Female","Kentucky","Bronze","3","49.71","Music","08/06/2023","4.2","78","
"8637669972","36","Female","Louisiana","Bronze","21","1240.65","Shoes","27/06/2023","4.3","91","34","false","1"
"0094471509","25","Male","Florida","Silver","25","1500.5","Garden","15/01/2023","2.4","16","31","false","0"
```

Replace

Find what: "

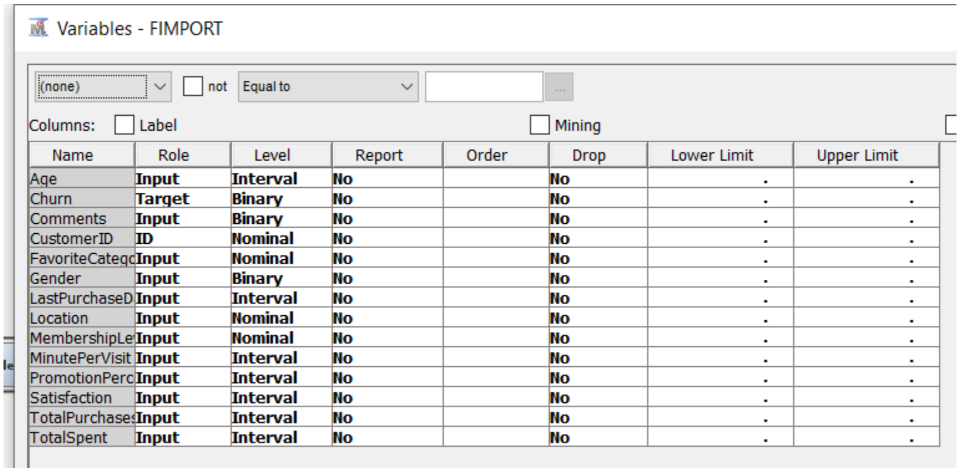
Replace with:

□ Match case

□ Wrap around

Figure 17: Removing quotation marks in NotePad.

The roles of each variable can be assigned during or after data import by selecting “Edit Variables” on the File Import Node. The setting for each variable is shown in Figure 18.



Variables - FIMPORT

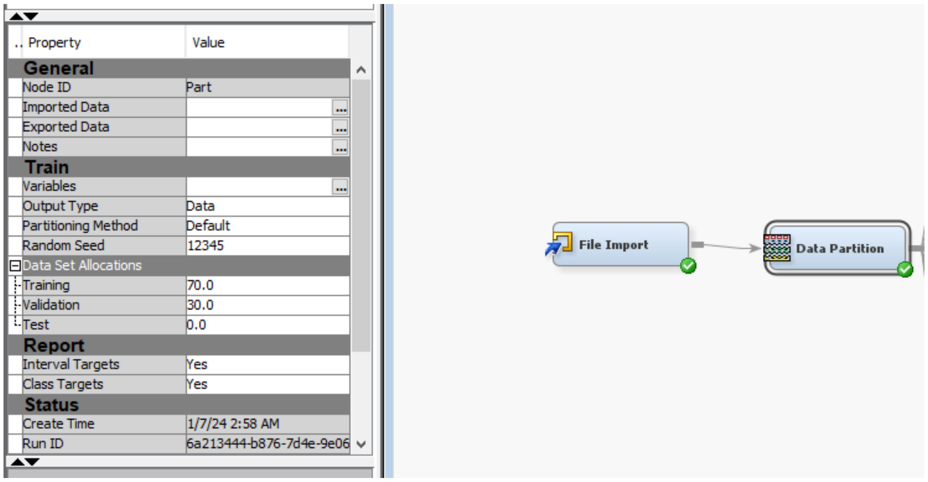
(none) □ not Equal to

Columns: □ Label □ Mining

Name	Role	Level	Report	Order	Drop	Lower Limit	Upper Limit
Age	Input	Interval	No		No	.	.
Churn	Target	Binary	No		No	.	.
Comments	Input	Binary	No		No	.	.
CustomerID	ID	Nominal	No		No	.	.
FavoriteCatego	Input	Nominal	No		No	.	.
Gender	Input	Binary	No		No	.	.
LastPurchaseD	Input	Interval	No		No	.	.
Location	Input	Nominal	No		No	.	.
MembershipLe	Input	Nominal	No		No	.	.
MinutePerVisit	Input	Interval	No		No	.	.
PromotionPerc	Input	Interval	No		No	.	.
Satisfaction	Input	Interval	No		No	.	.
TotalPurchases	Input	Interval	No		No	.	.
TotalSpent	Input	Interval	No		No	.	.

Figure 18: Data Schema settings.

The most important setting in Figure 18 is to set “Churn” as the target variable and make “CustomerID” as ID type to avoid the inclusion of this column during modelling. Next is data partitioning. A “Data Partition” node can added to the diagram and linked with “File Import”. In this study, the partition of data is set as 70% testing and 30% visualization.



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 2:58 AM
Run ID	6a213444-b876-7d4e-9e06

Figure 19: Setting up data partition.

Now, the data is ready to undergo modelling. In this study, decision tree-related models are used to model customer behaviour. Various decision tree models are constructed based on step-by-step tuning. The first two models are constructed as below:

Model Name	Nominal / Ordinal Criterion	Max Branch	Max Depth	Min Leaf Size	Validation Misclass Rate	Training Misclass Rate
DT1	Entropy	2	6	5	0.475	0.352
DT2	Gini	2	6	5	0.449	0.336

The misclassification results can be view via construction and running of Model Comparison node with the node linked to the model node. DT1 and DT2 are using fully default parameters except the nominal and ordinal target criterion, which is a part of parameter tuning. Observing the misclassification rate for validation, both models seem to have a large room for improvement available. As there are 51 locations and multiple options in FavouriteCategory available, the maximum branch is now increased to 3 as did for models DT3 and DT4.

Model Name	Nominal / Ordinal Criterion	Max Branch	Max Depth	Min Leaf Size	Validation Misclass Rate	Training Misclass Rate
DT3	Entropy	3	6	5	0.385	0.250
DT4	Gini	3	6	5	0.422	0.332

Compared to model DT1, DT3 with extra maximum branch brings a huge improvement in reducing misclassification rate for both validation and training set. For the Gini tune, DT4 does improve in classification rate although it is not as large as the Entropy tune. Now, the maximum allowed branches is increasing to 4 to observe what will happen.

Model Name	Nominal / Ordinal Criterion	Max Branch	Max Depth	Min Leaf Size	Validation Misclass Rate	Training Misclass Rate
DT5	Entropy	4	6	5	0.445	0.360
DT6	Gini	4	6	5	0.425	0.299

DT5 suffering a huge drop in performance of validation set classification tasks when compared to DT3 which has the same Entropy tune. The Gini-tuned DT6 is better than DT5 but not DT4. This situation indicates that is overfitting. Next, the maximum depth is adjusted based on models DT3 and DT4. The maximum depth is increasing by 1.

Model Name	Nominal / Ordinal Criterion	Max Branch	Max Depth	Min Leaf Size	Validation Misclass Rate	Training Misclass Rate
DT3	Entropy	3	6	5	0.385	0.250
DT4	Gini	3	6	5	0.422	0.332
DT7	Entropy	3	7	5	0.385	0.250
DT8	Gini	3	7	5	0.418	0.332

Increasing maximum depth do not improve the classification rate for both tuned criterion. That means increasing the maximum depth to 7 is causing overfitting. The DT3 and DT4 are still selected as the best models for each tune. Now, the maximum depth is decreased to 5 to observe whether this action can reduce the misclassification rate for validation set.

Model Name	Nominal / Ordinal Criterion	Max Branch	Max Depth	Min Leaf Size	Validation Misclass Rate	Training Misclass Rate
DT3	Entropy	3	6	5	0.385	0.250
DT4	Gini	3	6	5	0.422	0.332
DT9	Entropy	3	5	5	0.408	0.316

DT10	Gini	3	5	5	0.425	0.346

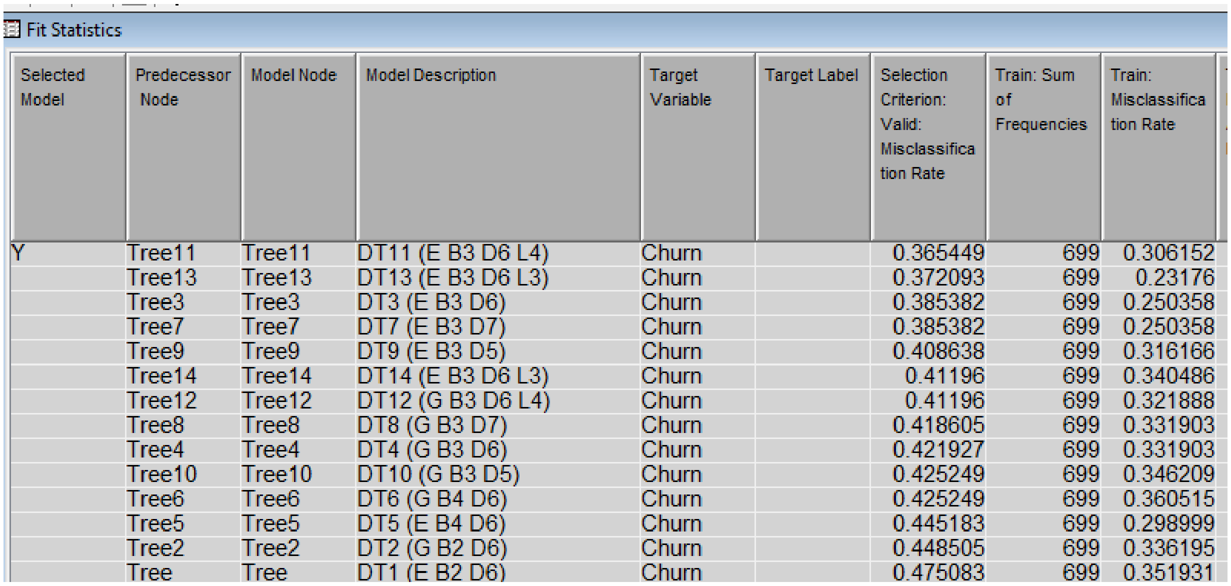
Both reduced maximum depth models (DT9 and DT10) are performing worse than before with increasing of misclassification rate for both validation and training set, indicating occurrence of underfitting. Hence the maximum branch of six is the best depth for this model. The next tuning is based on minimum leaf size.

Model Name	Nominal / Ordinal Criterion	Max Branch	Max Depth	Min Leaf Size	Validation Misclass Rate	Training Misclass Rate
DT3	Entropy	3	6	5	0.385	0.250
DT4	Gini	3	6	5	0.422	0.332
DT11	Entropy	3	6	4	0.365	0.306
DT12	Gini	3	6	4	0.412	0.322

Both reduced minimum leaf size models (DT11 and DT12) are improving the performance in reducing misclassification rate for validation rate. For now, DT11 is the best Entropy-tuned model while DT12 is the best Gini-tuned model. Next, the minimum leaf size is further reduced to 3.

Model Name	Nominal / Ordinal Criterion	Max Branch	Max Depth	Min Leaf Size	Validation Misclass Rate	Training Misclass Rate
DT11	Entropy	3	6	4	0.365	0.306
DT12	Gini	3	6	4	0.412	0.322
DT13	Entropy	3	6	3	0.372	0.231
DT14	Gini	3	6	3	0.412	0.340

Both DT13 and DT14 does not further improve the classification result of validation set. For DT13, the increasing difference between the validation and training set misclassification rate indicates the model is overfitting at this stage. There is no further decision tree tuning so the model DT11 with Entropy-tuned, maximum 3 branches, maximum 6 layer depth and minimum leaf size of 4 is the best decision tree model for this study. The overall performance ranking is tabulated in the output of Figure 20.



Fit Statistics

Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Target Label	Selection Criterion: Valid: Misclassification Rate	Train: Sum of Frequencies	Train: Misclassification Rate
Y	Tree11	Tree11	DT11 (E B3 D6 L4)	Churn		0.365449	699	0.306152
	Tree13	Tree13	DT13 (E B3 D6 L3)	Churn		0.372093	699	0.23176
	Tree3	Tree3	DT3 (E B3 D6)	Churn		0.385382	699	0.250358
	Tree7	Tree7	DT7 (E B3 D7)	Churn		0.385382	699	0.250358
	Tree9	Tree9	DT9 (E B3 D5)	Churn		0.408638	699	0.316166
	Tree14	Tree14	DT14 (E B3 D6 L3)	Churn		0.41196	699	0.340486
	Tree12	Tree12	DT12 (G B3 D6 L4)	Churn		0.41196	699	0.321888
	Tree8	Tree8	DT8 (G B3 D7)	Churn		0.418605	699	0.331903
	Tree4	Tree4	DT4 (G B3 D6)	Churn		0.421927	699	0.331903
	Tree10	Tree10	DT10 (G B3 D5)	Churn		0.425249	699	0.346209
	Tree6	Tree6	DT6 (G B4 D6)	Churn		0.425249	699	0.360515
	Tree5	Tree5	DT5 (E B4 D6)	Churn		0.445183	699	0.298999
	Tree2	Tree2	DT2 (G B2 D6)	Churn		0.448505	699	0.336195
	Tree	Tree	DT1 (E B2 D6)	Churn		0.475083	699	0.351931

Figure 20: Overall decision tree model ranking.

Notice that the best Gini-tuned model (DT12) is only can be ranked six out of fourteen models, indicating that Gini-tuning is not quite suitable in this study.

Several models are included in the modelling such as Bagging Ensemble Method (Random Forest), Gradient Boosting HP Tree and HP Forest. The Ensemble model 1 is the ensemble of the Top 2 best decision tree models (DT11 and DT13) while the Ensemble model 2 is the ensemble of the best Entropy-tuned model (DT11) and the best Gini-tuned model (DT12). Both HP environment tree and forest and the Gradient Boosting use the default parameters. The overall workflow is shown in Figure 21 and the overall results are in the output of Figure 22.

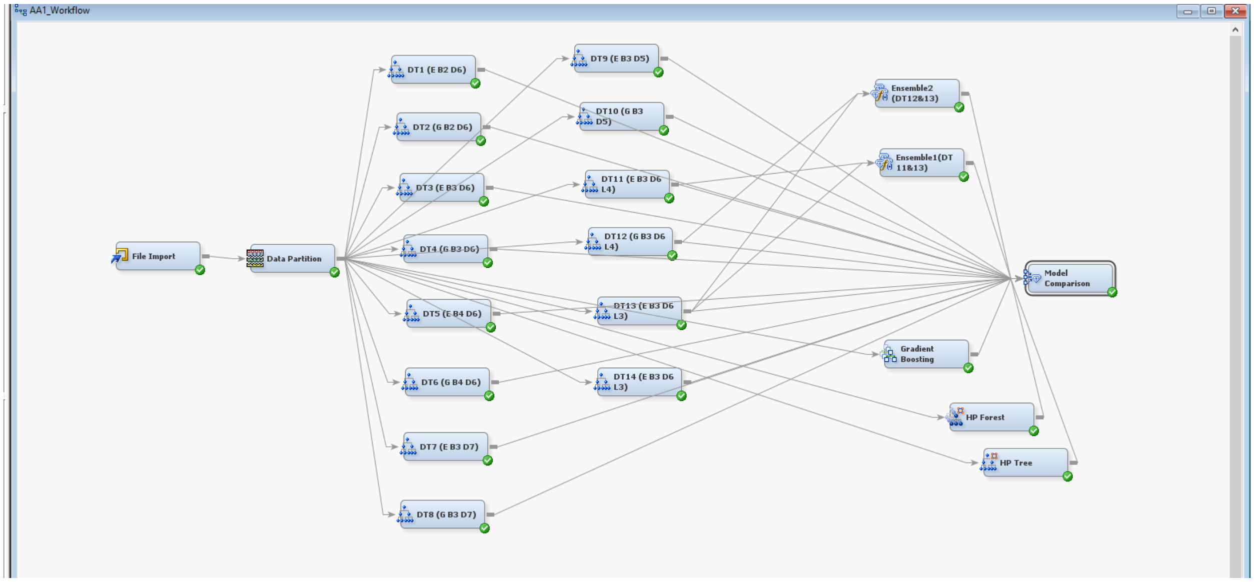
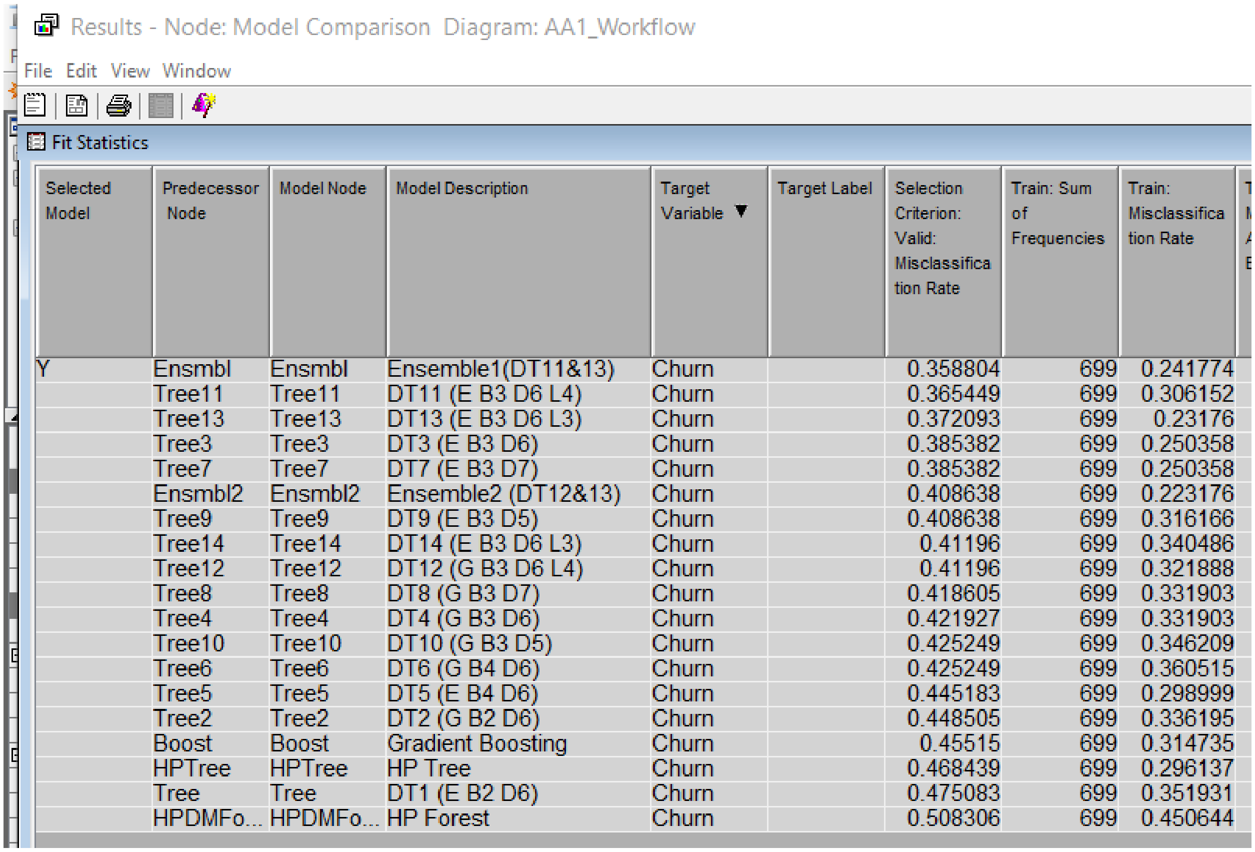


Figure 21: Overall Modelling Diagram in SAS EM.

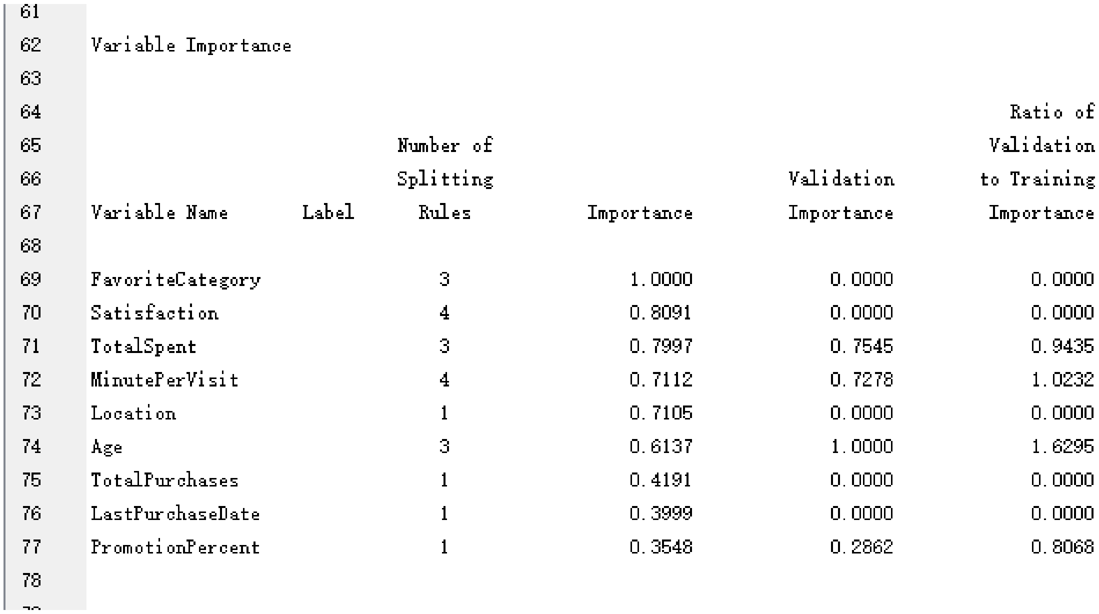


Selected Model	Predecessor Node	Model Node	Model Description	Target Variable ▼	Target Label	Selection Criterion: Valid: Misclassification Rate	Train: Sum of Frequencies	Train: Misclassification Rate
Y	Ensmbl	Ensmbl	Ensemble1(DT11&13)	Churn		0.358804	699	0.241774
	Tree11	Tree11	DT11 (E B3 D6 L4)	Churn		0.365449	699	0.306152
	Tree13	Tree13	DT13 (E B3 D6 L3)	Churn		0.372093	699	0.23176
	Tree3	Tree3	DT3 (E B3 D6)	Churn		0.385382	699	0.250358
	Tree7	Tree7	DT7 (E B3 D7)	Churn		0.385382	699	0.250358
	Ensmbl2	Ensmbl2	Ensemble2 (DT12&13)	Churn		0.408638	699	0.223176
	Tree9	Tree9	DT9 (E B3 D5)	Churn		0.408638	699	0.316166
	Tree14	Tree14	DT14 (E B3 D6 L3)	Churn		0.41196	699	0.340486
	Tree12	Tree12	DT12 (G B3 D6 L4)	Churn		0.41196	699	0.321888
	Tree8	Tree8	DT8 (G B3 D7)	Churn		0.418605	699	0.331903
	Tree4	Tree4	DT4 (G B3 D6)	Churn		0.421927	699	0.331903
	Tree10	Tree10	DT10 (G B3 D5)	Churn		0.425249	699	0.346209
	Tree6	Tree6	DT6 (G B4 D6)	Churn		0.425249	699	0.360515
	Tree5	Tree5	DT5 (E B4 D6)	Churn		0.445183	699	0.298999
	Tree2	Tree2	DT2 (G B2 D6)	Churn		0.448505	699	0.336195
	Boost	Boost	Gradient Boosting	Churn		0.45515	699	0.314735
	HPTree	HPTree	HP Tree	Churn		0.468439	699	0.296137
	Tree	Tree	DT1 (E B2 D6)	Churn		0.475083	699	0.351931
	HPDMFo...	HPDMFo...	HP Forest	Churn		0.508306	699	0.450644

Figure 22: Overall modelling result.

From the output in Figure 22, all three models with the default setting (Gradient Boosting, HP Tree and HP Forest) are among the weakest models in classification. Future tuning with better understanding is recommended so that the fined-tuned models have the opportunity to surpass the best decision tree model. The ensemble model of DT11 and DT13 is the best-performed model in validation set churn classification. Although the bagged ensemble model is the best model, for easier interpretation, the second best model which is DT11 is used to generate insight regarding customer behaviour.

Although the misclassification rate is 36% which is considered high, by observing the decision tree diagram, the Location and the favoured Content are the top two layers for every branch. However, by observing the variable importance in Figure 23, the most important variable for validation is age. Hence, I would like to suggest the e-commerce owner put more focus on location, age and favoured category as these factors are likely to influence the churn of the customer.



```
Variable Importance


                                                                              Ratio of
                               Number of                                    Validation
                               Splitting                      Validation   to Training
Variable Name       Label        Rules       Importance       Importance    Importance

FavoriteCategory                   3           1.0000           0.0000        0.0000
Satisfaction                       4           0.8091           0.0000        0.0000
TotalSpent                         3           0.7997           0.7545        0.9435
MinutePerVisit                     4           0.7112           0.7278        1.0232
Location                           1           0.7105           0.0000        0.0000
Age                                3           0.6137           1.0000        1.6295
TotalPurchases                     1           0.4191           0.0000        0.0000
LastPurchaseDate                   1           0.3999           0.0000        0.0000
PromotionPercent                   1           0.3548           0.2862        0.8068
```

Figure 23: Result output of model DT11.

Summary and Difficulties

The random forest which is the bagged ensemble method of the decision trees is the best type of model for classifying the churn of customers. More detailed insights are subject to be discovered as at the moment there is no much analysis that can be explained due to a quite high misclassification rate for the best model available. Other models other than decision trees are advised to be applied in this study. The high rate of misclassification may be due to the data is generated highly randomly with limited limitations applied. Besides, data from 50 United States and 1 federal district (District of Columbia) is insufficient for 1000 rows, which increases the difficulty of insight generations. With the existence of this limitation, the extension of this study can include the tuning of HP environment model and Gradient Boosting in attempt of generate more valuable insights from this dataset.