M.KUMARASAMY
COLLEGE OF ENGINEERING
NAAC Accredited Autonomous Institution
Approved by AICTE & Affiliated to Anna University
ISO 9001:2015 Certified Institution
Thalavapalayam, Karur - 639 113.

KR

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

RECORD NOTEBOOK

ACADEMIC YEAR 2023 – 2024

(EVEN Semester)

III YEAR / VI SEMESTER

COURSE CODE: ____

COURSE NAME: ____

NAME : ____

ROLL NO : ____

YEAR : ______________ SEM:___________________________

BRANCH : ____

M.KUMARASAMY COLLEGE OF ENGINEERING
(Autonomous)
KARUR-639113

REGISTER NUMBER: □

Certified that this is the Bonafide record of work done by

Selvan / Selvi____ of Sixth

Semester **B.Tech (ARTIFICIAL INTELLIGENCE AND DATA SCIENCE)**

branch during academic year 2023 – 2024 in

the____

FACULTY IN CHARGE **HEAD OF THE DEPARTMENT**

Submitted for the End Semester Practical Examination on______________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

CONTENTS

EXP :1 DATE:	COLLECT INITIAL DATA FOR THE TELECOM FIRM

AIM:

To execute a program which collect initial data for the telecom firm.

PROCEDURE:

IMPORT FROM MICROSOFT EXCEL:

STEP1: From the Sources palette, place an Excel node on the stream canvas.

STEP2: Edit the Excel node. Click the Data tab, if not already selected.

STEP 3: In the File type box, ensure that Excel 2007-2016 (*.xlsx) is selected.

STEP 4: In the Import file box, select telco x customer data.xlsx from the location where it is stored.

STEP 5: Ensure that the option First row has column names is enabled.

STEP 6: Click Preview.

STEP 7: Close the Preview output window.

STEP 8: Close the Excel dialog box.

IMPORT FROM A TAB-DELIMITED TEXT FILE:

STEP 1: From the Sources palette, add a Var. File node to the stream canvas.

STEP 2: Edit the Var. File node. Click the File tab, if not already selected.

STEP3: In the File box, select telco x products.tab from the location where it is stored.

STEP4: Ensure that the option Read field names from file is enabled.

STEP5: In the Field delimiters section, click the Comma check box to disable it.

STEP6: In the Field delimiters section, click the Tab check box to enable it.

STEP 7: Click Preview.

STEP 8: Close the Preview output window.

STEP 9: Close the Var. File dialog box.

IMPORT FROM IBM SPSS STATISTICS:

STEP 1: From the Sources palette, add a Statistics File node to the stream canvas.

STEP 2: Edit the Statistics File node. Click the Data tab, if not already selected.

STEP 3: In the File box, select telco x tariffs. sav from the location where it is stored.

STEP 4: Click the Use field format information to determine the storage checkbox to enable it.

STEP 5: Click Preview.

STEP 6: Close the Preview output window.

STEP 7: Close the Statistics File dialog box.

SET MEASUREMENT LEVELS:

STEP 1: From the Field Ops palette, add a Type node downstream from the Microsoft Excel node.

STEP 2: Edit the Type node.

STEP3: Click Read Values

STEP 4: Examine the results in the Values and Measurement column.

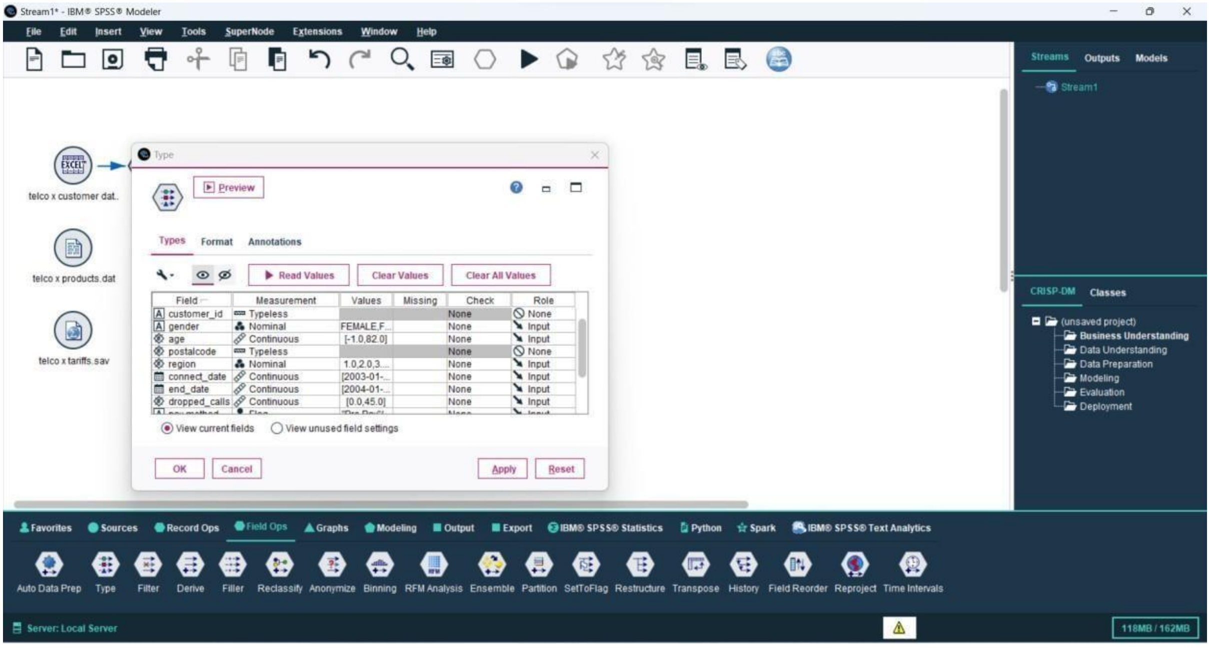
STEP 5: Click the cell in the POSTAL_CODE row, Measurement column, and then Click Categorical from the drop-down.

STEP6: Click the cell in the REGION row, Measurement column, and then Click Categorical from the drop-down.

STEP 7: Click Read Values.

STEP 8: Close the Type dialog box.

OUTPUT:


Stream1* - IBM® SPSS® Modeler
File Edit Insert View Tools SuperNode Extensions Window Help
Streams Outputs Models
Stream1
telco x customer dat..
telco x products.dat
telco x tariffs.sav
Type
Preview
Types Format Annotations
Read Values Clear Values Clear All Values
○ OK Cancel Apply Reset
CRISP-DM Classes
(unsaved project)
Business Understanding
Data Understanding
Data Preparation
Modeling
Evaluation
Deployment
Favorites Sources Record Ops Field Ops Graphs Modeling Output Export IBM® SPSS® Statistics Python Spark IBM® SPSS® Text Analytics
Auto Data Prep Type Filter Derive Filler Reclassify Anonymize Binning RFM Analysis Ensemble Partition SetToFlag Restructure Transpose History Field Reorder Reproject Time Intervals
Server: Local Server
118MB / 162MB

Field	Measurement	Values	Missing	Check	Role
customer_id	Typeless			None	None
gender	Nominal	FEMALE,F...		None	Input
age	Continuous	[-1.0,82.0]		None	Input
postalcode	Typeless			None	None
region	Nominal	1.0,2.0,3....		None	Input
connect_date	Continuous	[2003-01-...		None	Input
end_date	Continuous	[2004-01-...		None	Input
dropped_calls	Continuous	[0.0,45.0]		None	Input

● View current fields ○ View unused field settings

RESULT:

Thus, the Collect initial data for the telecom firm Program has been Executed Successfully.

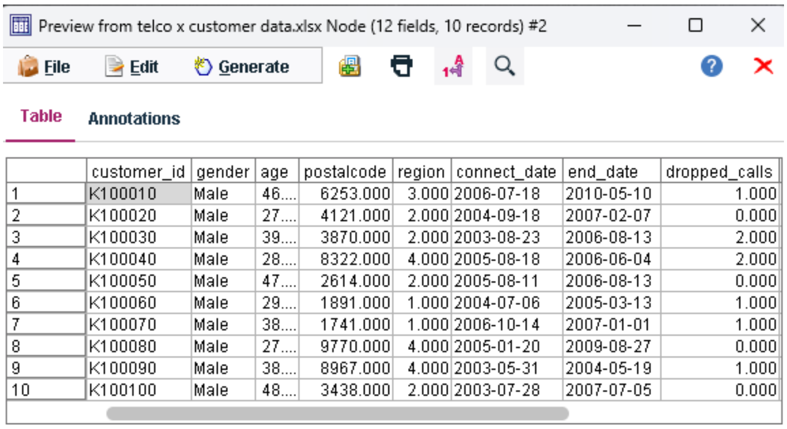
EXP :2 DATE:	UNDERSTAND TELECOMMUNICATIONS DATA

AIM:

To Create a stream for collecting initial telecom firm data and understand the data properties using the IBM SPSS modeler.

ALGORITHM:

STEP 1: From the **Sources** palette, place an **Excel** node then **import the input file**, as **telco x customer data.xlsx**.

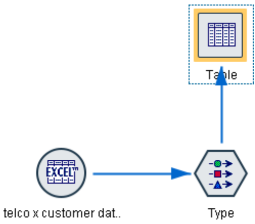


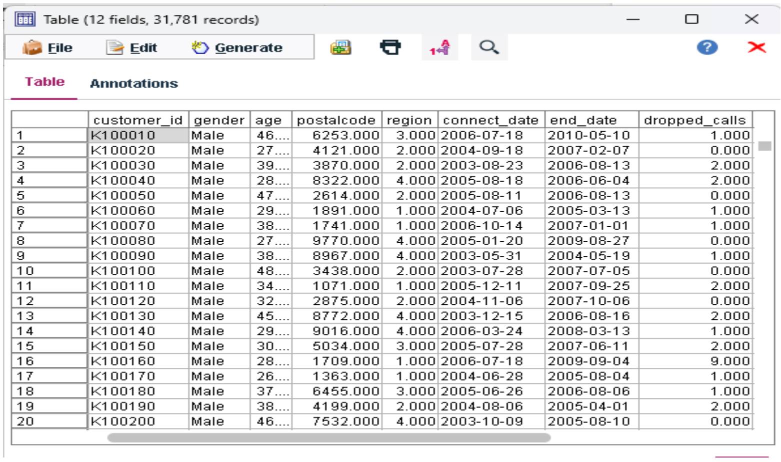
Preview from telco x customer data.xlsx Node (12 fields, 10 records) #2
File
Edit
Generate
Table
Annotations

	customer_id	gender	age	postalcode	region	connect_date	end_date	dropped_calls
1	K100010	Male	46....	6253.000	3.000	2006-07-18	2010-05-10	1.000
2	K100020	Male	27....	4121.000	2.000	2004-09-18	2007-02-07	0.000
3	K100030	Male	39....	3870.000	2.000	2003-08-23	2006-08-13	2.000
4	K100040	Male	28....	8322.000	4.000	2005-08-18	2006-06-04	2.000
5	K100050	Male	47....	2614.000	2.000	2005-08-11	2006-08-13	0.000
6	K100060	Male	29....	1891.000	1.000	2004-07-06	2005-03-13	1.000
7	K100070	Male	38....	1741.000	1.000	2006-10-14	2007-01-01	1.000
8	K100080	Male	27....	9770.000	4.000	2005-01-20	2009-08-27	0.000
9	K100090	Male	38....	8967.000	4.000	2003-05-31	2004-05-19	1.000
10	K100100	Male	48....	3438.000	2.000	2003-07-28	2007-07-05	0.000

Close the Preview output window and Excel dialog box.

STEP 2: From the **Field Ops** palette, add a Type node and read the values in the type node. Then from the **Output** palette, add a **Table node**. Then Run the Table node.

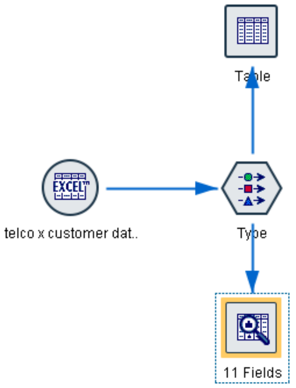




	customer_id	gender	age	postalcode	region	connect_date	end_date	dropped_calls
1	K100010	Male	46....	6253.000	3.000	2006-07-18	2010-05-10	1.000
2	K100020	Male	27....	4121.000	2.000	2004-09-18	2007-02-07	0.000
3	K100030	Male	39....	3870.000	2.000	2003-08-23	2006-08-13	2.000
4	K100040	Male	28....	8322.000	4.000	2005-08-18	2006-06-04	2.000
5	K100050	Male	47....	2614.000	2.000	2005-08-11	2006-08-13	0.000
6	K100060	Male	29....	1891.000	1.000	2004-07-06	2005-03-13	1.000
7	K100070	Male	38....	1741.000	1.000	2006-10-14	2007-01-01	1.000
8	K100080	Male	27....	9770.000	4.000	2005-01-20	2009-08-27	0.000
9	K100090	Male	38....	8967.000	4.000	2003-05-31	2004-05-19	1.000
10	K100100	Male	48....	3438.000	2.000	2003-07-28	2007-07-05	0.000
11	K100110	Male	34....	1071.000	1.000	2005-12-11	2007-09-25	2.000
12	K100120	Male	32....	2875.000	2.000	2004-11-06	2007-10-06	0.000
13	K100130	Male	45....	8772.000	4.000	2003-12-15	2006-08-16	2.000
14	K100140	Male	29....	9016.000	4.000	2006-03-24	2008-03-13	1.000
15	K100150	Male	30....	5034.000	3.000	2005-07-28	2007-06-11	2.000
16	K100160	Male	28....	1709.000	1.000	2006-07-18	2009-09-04	9.000
17	K100170	Male	26....	1363.000	1.000	2004-06-28	2005-08-04	1.000
18	K100180	Male	37....	6455.000	3.000	2005-06-26	2006-08-06	1.000
19	K100190	Male	38....	4199.000	2.000	2004-08-06	2005-04-01	2.000
20	K100200	Male	46....	7532.000	4.000	2003-10-09	2005-08-10	0.000

Close the Table output window

STEP 3: From the **Output** palette, add a **Data Audit** node downstream from the **Type** node. Then run the **Data Audit** node.



The minimum value for ***AGE is -1***, which is clearly an ***invalid value***.

STEP 4: Edit the **Type** node Click the cell in the **Values** column, **AGE** row (where it reads [-1.0, 82.0]), and then click **Specify** the AGE Values sub-dialog box opens.

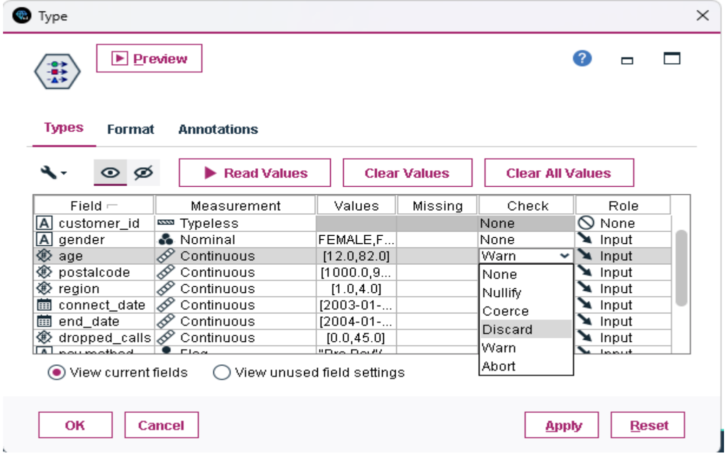
Click the **Specify** values and labels option, set then set **Lower** to **12** and **Upper** to **90**.

Close the **AGE Values** sub-dialog box.

STEP 5: Click the cell in the **Check** column, **AGE** row, and then click **Warn** from the drop-down

STEP 6: Close the **Type** dialog box. You will rerun the Data Audit node to examine the effect of specifying a valid range. Run the **Data Audit** node. The minimum AGE value is still -1, so it seems as if nothing has changed. Close the **Data Audit** output window.

STEP 7: Edit the **Type** node, click the cell in the **Check** column in the **END_DATE** row, and then set the action to **Discard**.

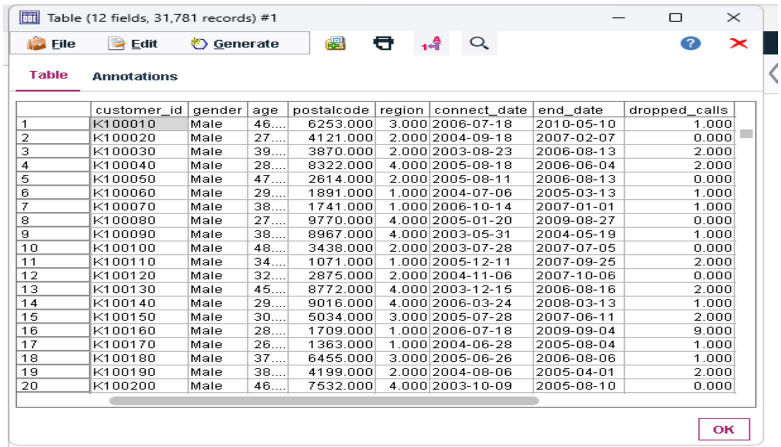


Field	Measurement	Values	Missing	Check	Role
customer_id	Typeless			None	None
gender	Nominal	FEMALE,F...		None	Input
age	Continuous	[12.0,82.0]		Warn	Input
postalcode	Continuous	[1000.0,9...			Input
region	Continuous	[1.0,4.0]			Input
connect_date	Continuous	[2003-01-...			Input
end_date	Continuous	[2004-01-...			Input
dropped_calls	Continuous	[0.0,45.0]			Input

Close the **Type** dialog box.

STEP 8: Run the **Table** node that is downstream from the **Type** node.

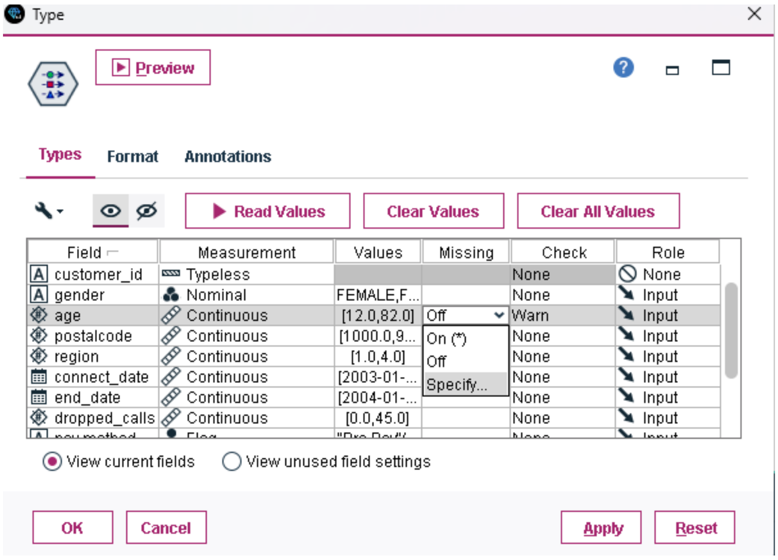
Scroll to the right so that you can view **END_DATE** and then scroll down to verify that **END_DATE** is never $null$.



	customer_id	gender	age	postalcode	region	connect_date	end_date	dropped_calls
1	K100010	Male	46....	6253.000	3.000	2006-07-18	2010-05-10	1.000
2	K100020	Male	27....	4121.000	2.000	2004-09-18	2007-02-07	0.000
3	K100030	Male	39....	3870.000	2.000	2003-08-23	2006-08-13	2.000
4	K100040	Male	28....	8322.000	4.000	2005-08-18	2006-06-04	2.000
5	K100050	Male	47....	2614.000	2.000	2005-08-11	2006-08-13	0.000
6	K100060	Male	29....	1891.000	1.000	2004-07-06	2005-03-13	1.000
7	K100070	Male	38....	1741.000	1.000	2006-10-14	2007-01-01	1.000
8	K100080	Male	27....	9770.000	4.000	2005-01-20	2009-08-27	0.000
9	K100090	Male	38....	8967.000	4.000	2003-05-31	2004-05-19	1.000
10	K100100	Male	48....	3438.000	2.000	2003-07-28	2007-07-05	0.000
11	K100110	Male	34....	1071.000	1.000	2005-12-11	2007-09-25	2.000
12	K100120	Male	32....	2875.000	2.000	2004-11-06	2007-10-06	0.000
13	K100130	Male	45....	8772.000	4.000	2003-12-15	2006-08-16	2.000
14	K100140	Male	29....	9016.000	4.000	2006-03-24	2008-03-13	1.000
15	K100150	Male	30....	5034.000	3.000	2005-07-28	2007-06-11	2.000
16	K100160	Male	28....	1709.000	1.000	2006-07-18	2009-09-04	9.000
17	K100170	Male	26....	1363.000	1.000	2004-06-28	2005-08-04	1.000
18	K100180	Male	37....	6455.000	3.000	2005-06-26	2006-08-06	1.000
19	K100190	Male	38....	4199.000	2.000	2004-08-06	2005-04-01	2.000
20	K100200	Male	46....	7532.000	4.000	2003-10-09	2005-08-10	0.000

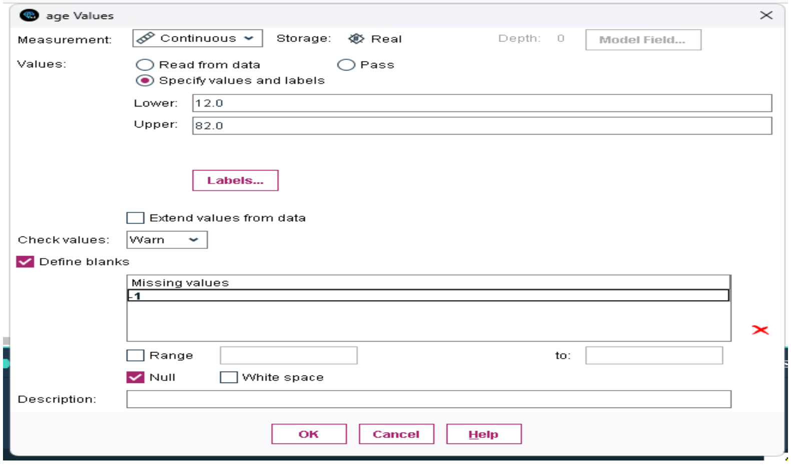
Only 14,698 records are retained. Close the **Table** output window.

STEP 9: Edit the **Type** node. Click the cell in the **Missing** column, **AGE** row, and then click **Specify.**



Field	Measurement	Values	Missing	Check	Role
customer_id	Typeless			None	None
gender	Nominal	FEMALE,F...		None	Input
age	Continuous	[12.0,82.0]	Off	Warn	Input
postalcode	Continuous	[1000.0,9...		None	Input
region	Continuous	[1.0,4.0]		None	Input
connect_date	Continuous	[2003-01-...		None	Input
end_date	Continuous	[2004-01-...		None	Input
dropped_calls	Continuous	[0.0,45.0]		None	Input

STEP 10: Click the **Define blanks** check box to enable it. And Below **Missing values**, type **-1**.



age Values

Measurement: Continuous

Storage: Real

Depth: 0

Values:

○ Read from data

○ Pass

● Specify values and labels

Lower: 12.0

Upper: 82.0

□ Extend values from data

Check values: Warn

☑ Define blanks

Missing values

-1

□ Range

to:

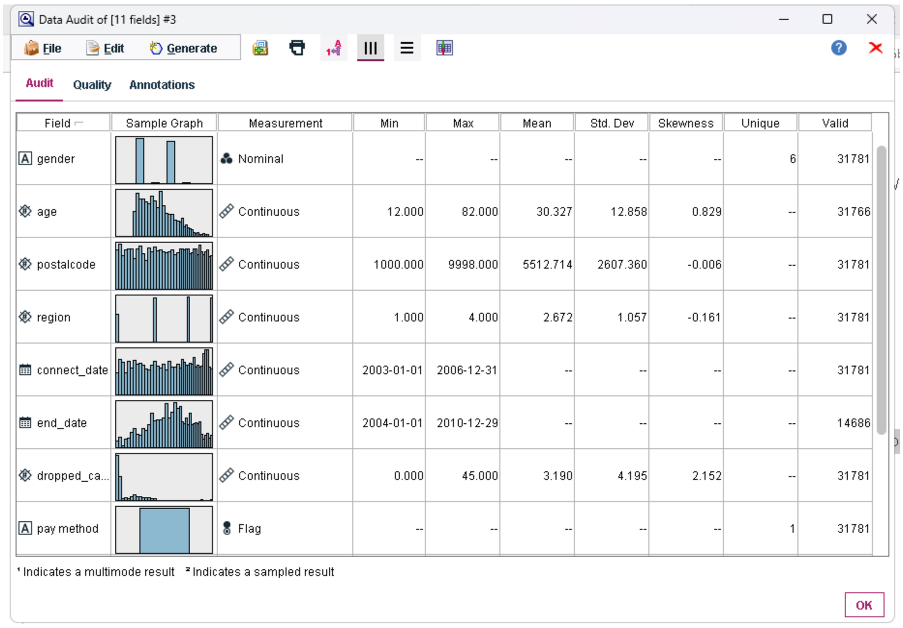
☑ Null

□ White space

Description:

Close the **AGE Values** sub dialog box. **And** Close the **Type** dialog box.

Run the **Data Audit** node.



Data Audit of [11 fields] #3

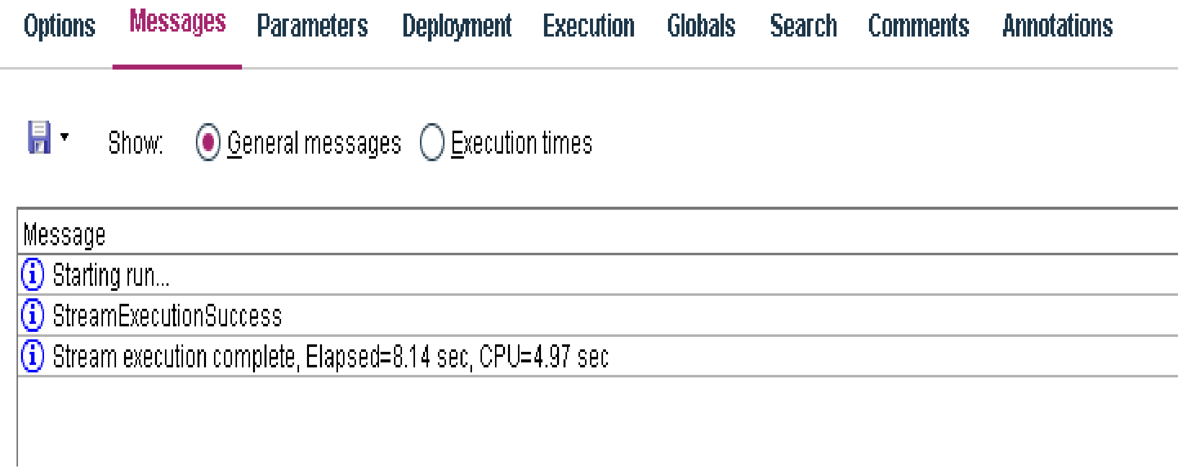
Field	Sample Graph	Measurement	Min	Max	Mean	Std. Dev	Skewness	Unique	Valid
gender		Nominal	--	--	--	--	--	6	31781
age		Continuous	12.000	82.000	30.327	12.858	0.829	--	31766
postalcode		Continuous	1000.000	9998.000	5512.714	2607.360	-0.006	--	31781
region		Continuous	1.000	4.000	2.672	1.057	-0.161	--	31781
connect_date		Continuous	2003-01-01	2006-12-31	--	--	--	--	31781
end_date		Continuous	2004-01-01	2010-12-29	--	--	--	--	14686
dropped_ca...		Continuous	0.000	45.000	3.190	4.195	2.152	--	31781
pay method		Flag	--	--	--	--	--	1	31781

¹ Indicates a multimode result ² Indicates a sampled result

The minimum value for AGE is 12 instead of -1.

There are no stream messages so no out-of-range values were found.

OUTPUT:



Show: ● General messages ○ Execution times

```
Message
ⓘ Starting run...
ⓘ StreamExecutionSuccess
ⓘ Stream execution complete, Elapsed=8.14 sec, CPU=4.97 sec
```

RESULT:

Thus, the Colleting and Understanding the telecommunication data. Program has been Executed Successfully.

EXP :3 DATE:	**SET THE UNIT OF ANALYSIS FOR THE TELECOMMUNICATIONS DATA**

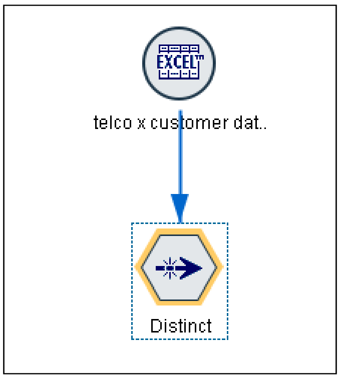
AIM:

To remove duplicate records in the customer dataset and transform a transactional dataset into a dataset that has one record per customer using the IBM SPSS modeler.

PROCEDURE TO IMPLEMENTATION:

1. TO REMOVE DUPLICATE RECORDS

STEP 1: Import the data file **telco x customer data.xlsx** using the Excel source node. Then add a **Distinct** node from the **Record Ops** palette.



STEP 2: Edit the **Distinct** node. In the **Settings** tab. Click the **Mode** drop-down, to view the options. From the **Mode,** drop-down click ***Include only the first record in each group***.

[*The Include only the first record in each group* option retains only the first

record of the group. You will need this option to remove duplicate record

STEP 3: Click the **Pick from the set of available fields** button, click **All** and then click **OK**.

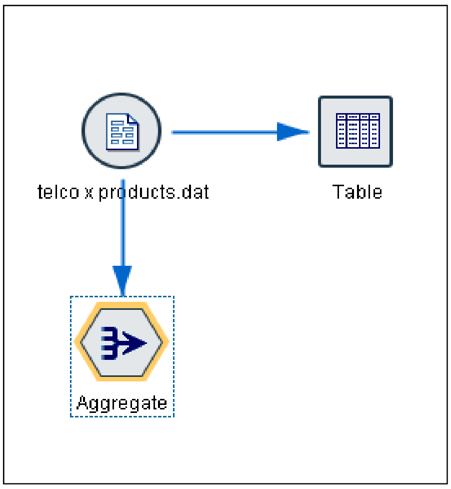
Connect a Table node and execute it. Check the results.

Starting records – 31,781 and Current result records: 31,769

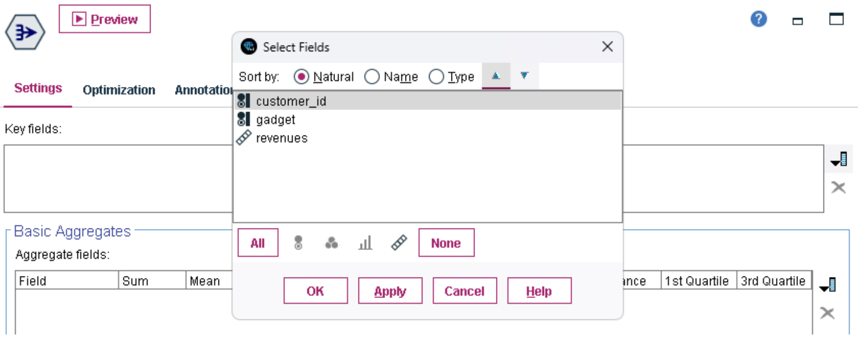
2. AGGREGATE TRANSACTIONAL DATA

STEP 1: Import the data file telco x products.dat file chose the field delimiter as ***Tab*** and ***newline***, and add a Table node to it. Then Run this Table node. Close the **Table** output window.

STEP 2: From the **Record Ops** palette, add an **Aggregate** node downstream to the source node **telco x products.dat**.



STEP 3: Edit the **Aggregate** node. Click the **Settings** tab. In the **Key Fields** box, select **CUSTOMER_ID**.



Select Fields

Sort by: ● Natural ○ Name ○ Type

customer_id
gadget
revenues

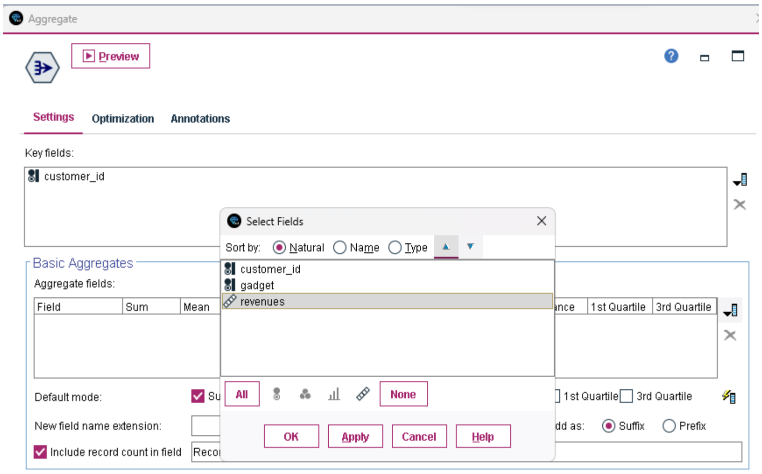
Key fields:

Basic Aggregates

Aggregate fields:

Field	Sum	Mean	ance	1st Quartile	3rd Quartile

STEP 4: In the **Basic Aggregates** section, in the **Aggregate fields** sub-section, select **REVENUES**.



Key fields: customer_id

Sort by: ● Natural ○ Name ○ Type

customer_id
gadget
revenues

Aggregate fields:

Default mode: ☑ Su ... □ 1st Quartile □ 3rd Quartile

New field name extension:

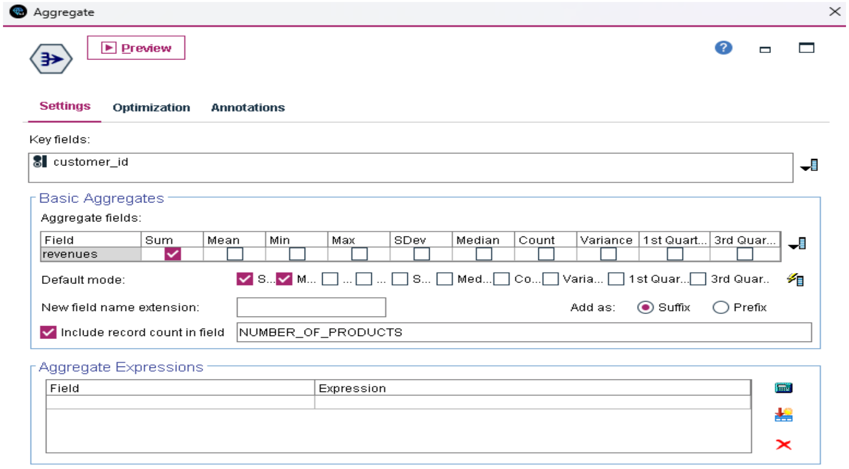
...dd as: ● Suffix ○ Prefix

☑ Include record count in field: Recor...

Two statistics will be computed by default, mean and sum. Only the **sum is required in this exercise.**

So, Click the check box in the **Mean** column so that it is disabled.

STEP 5: Ensure that the **Include record count in the field** check box is enabled and then type **NUMBER_OF_PRODUCTS** in the text box.



Key fields: customer_id

Aggregate fields:

Field	Sum	Mean	Min	Max	SDev	Median	Count	Variance	1st Quart...	3rd Quar...
revenues	☑	□	□	□	□	□	□	□	□	□

Default mode: ☑ S... ☑ M... □ ... □ ... □ S... □ Med... □ Co... □ Varia... □ 1st Quar... □ 3rd Quar..

New field name extension:

Add as: ● Suffix ○ Prefix

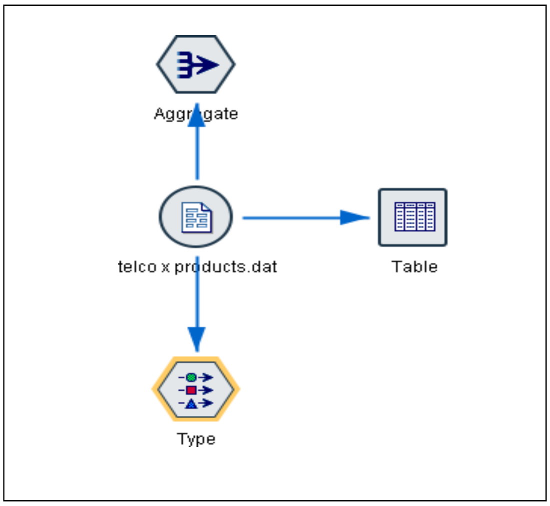
☑ Include record count in field: NUMBER_OF_PRODUCTS

Field	Expression

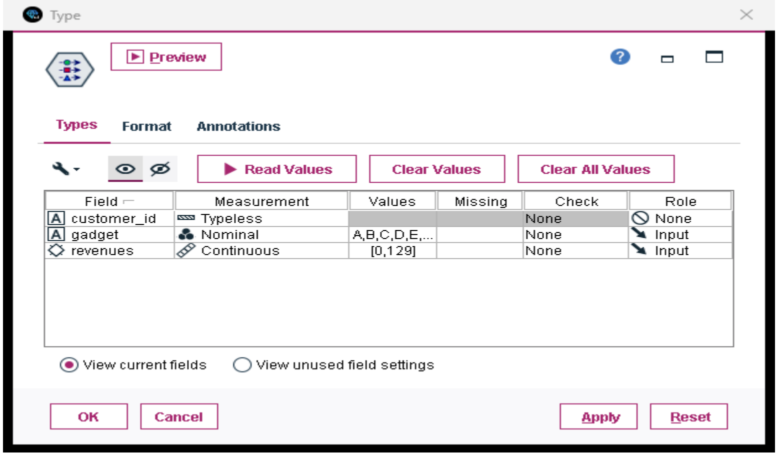
STEP 6: Click the **Optimization** tab. Click the **Keys are contiguous** check box to enable it.

3. CREATE FLAG FIELDS AND AGGREGATE THE DATA

STEP 1: From the **Field Ops** palette, add a **Type** node downstream from the **Var. File** node named **telco x products**.



STEP 2: Edit the **Type** node. Click the **Types** tab and then Click **Read Values**

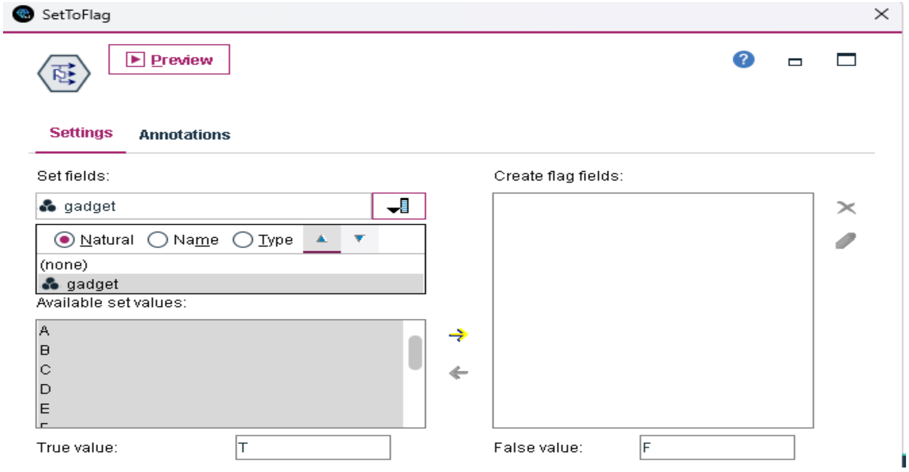


Field	Measurement	Values	Missing	Check	Role
customer_id	Typeless			None	None
gadget	Nominal	A,B,C,D,E,...		None	Input
revenues	Continuous	[0,129]		None	Input

● View current fields ○ View unused field settings

STEP 3: From the **Field Ops** palette, add a **SetToFlag** node downstream from the **Type** node.

STEP 4: Edit the **SetToFlag** node. In **Settings** tab click the **Set fields** drop down and then click **gadget**.



Set fields:

gadget

○ Natural ○ Name ○ Type

(none)

gadget

Available set values:

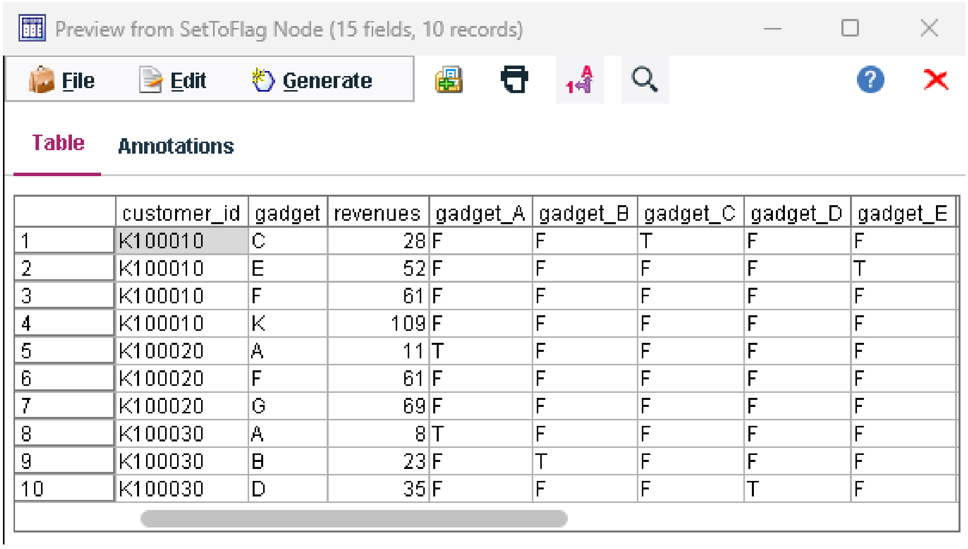
A
B
C
D
E

True value: T

Create flag fields:

False value: F

STEP 5: Select all values in the **Available set values** box and then move them into the **Create flag fields** box.

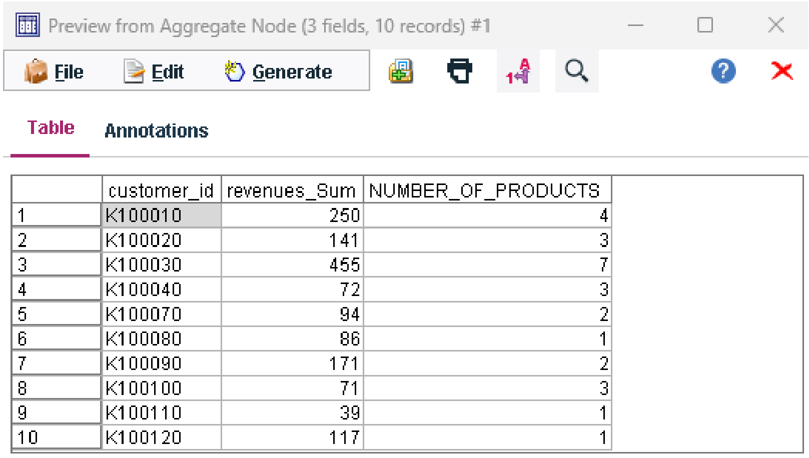


Preview from SetToFlag Node (15 fields, 10 records)

	customer_id	gadget	revenues	gadget_A	gadget_B	gadget_C	gadget_D	gadget_E
1	K100010	C	28	F	F	T	F	F
2	K100010	E	52	F	F	F	F	T
3	K100010	F	61	F	F	F	F	F
4	K100010	K	109	F	F	F	F	F
5	K100020	A	11	T	F	F	F	F
6	K100020	F	61	F	F	F	F	F
7	K100020	G	69	F	F	F	F	F
8	K100030	A	8	T	F	F	F	F
9	K100030	B	23	F	T	F	F	F
10	K100030	D	35	F	F	F	T	F

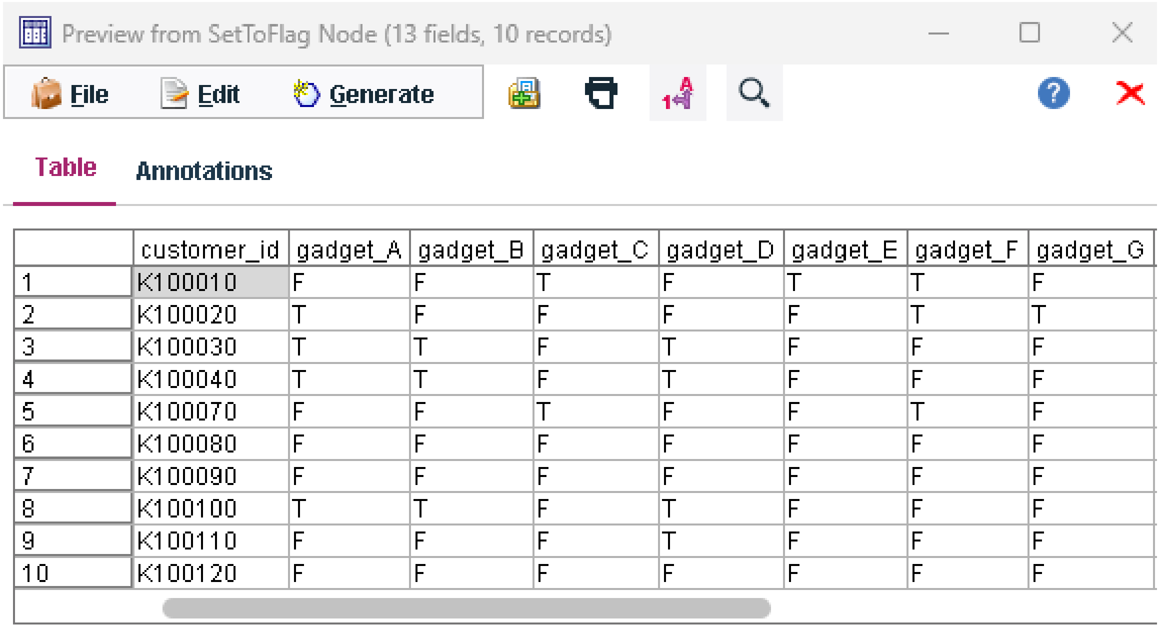
STEP 6: Click the **Aggregate keys** check box to enable it. In the **Aggregate keys** box, select **CUSTOMER_ID**. Click preview

OUTPUT[AGGREGATE]:



	customer_id	revenues_Sum	NUMBER_OF_PRODUCTS
1	K100010	250	4
2	K100020	141	3
3	K100030	455	7
4	K100040	72	3
5	K100070	94	2
6	K100080	86	1
7	K100090	171	2
8	K100100	71	3
9	K100110	39	1
10	K100120	117	1

OUTPUT [SET TO FLAG]:



	customer_id	gadget_A	gadget_B	gadget_C	gadget_D	gadget_E	gadget_F	gadget_G
1	K100010	F	F	T	F	T	T	F
2	K100020	T	F	F	F	F	T	T
3	K100030	T	T	F	T	F	F	F
4	K100040	T	T	F	T	F	F	F
5	K100070	F	F	T	F	F	T	F
6	K100080	F	F	F	F	F	F	F
7	K100090	F	F	F	F	F	F	F
8	K100100	T	T	F	T	F	F	F
9	K100110	F	F	F	T	F	F	F
10	K100120	F	F	F	F	F	F	F

RESULT:

Thus, the Set unit of analysis for the data Remove, Aggregate, Create Program has been Executed Successfully.

EXP :4 DATE:	IDENTIFY RELATIONSHIPS IN THE TELECOMMUNICATIONS DATA

AIM:

To identify relationships between the following:

a) Examine the relationship between categorical.

b) Examine the relationship between a categorical and continuous field.

PROCEDURE TO IMPLEMENTATION:

a) Examine the Relationship between Categorical fields

STEP1: Import the file telco x data.txt. From the **Output** palette, add a **Matrix** node downstream from the **Type** node.

STEP 2: Edit the **Matrix** node.

- In the **Rows** box, select **HANDSET**.
- In the **Columns** box, select **CHURN**.
- Click the **Include missing values** check box to disable it.

STEP 3: In the **Appearance** tab. Click the **Percentage of row** check box to enable it. And also Click the **Include row and column totals** check box to enable it. Click **Run**.

The churn rate for customers with handset ASAD170 is 4.627%, whereas it is 94.856% for those with handset ASAD90.

Close the **Matrix** output window.

STEP 4: From the **Graphs** palette, add a **Distribution** node downstream from the **Type** node.

STEP 5: Edit the **Distribution** node.

In the **Field** box, select **HANDSET**. In the **Color** box, select **CHURN**.

Click the **Normalize by color** check box to enable it. Click **Run**.

b) Examine the Relationship between Categorical and Continuous field

STEP 1: From the **Output** palette, add a **Means** node downstream from the **Type** node.

STEP 2: Edit the **Means** node. In the **Grouping field** box, select **CHURN**.

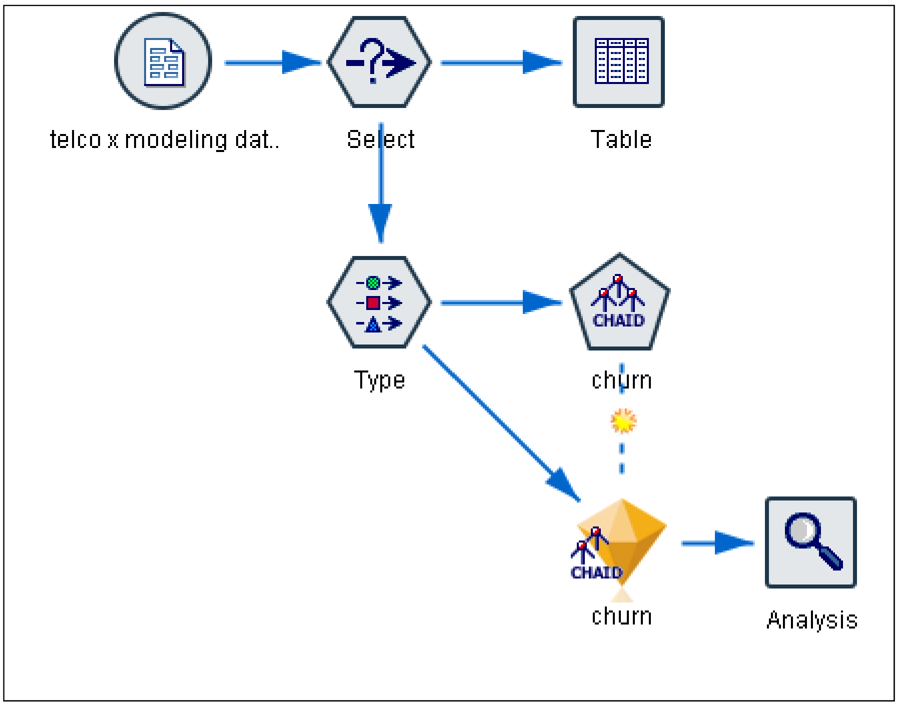
Similarly In the **Test field(s)** box, select **DROPPED_CALLS**. Click **Run**.

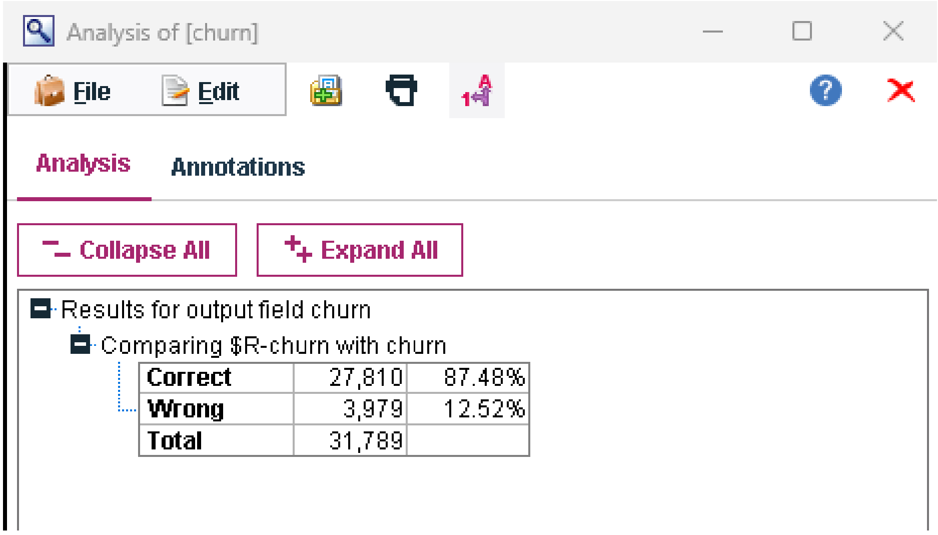
Close the **Means** output window.

STEP 4: From the **Graphs** palette, add a **Histogram** node downstream from the **Type** node

STEP 5: Edit the **Histogram** node. In the **Field** box, select **DROPPED_CALLS**. In the **Color** box, select **CHURN**. Click **Run**.

OUTPUT:





Results for output field churn

Comparing $R-churn with churn

Correct	27,810	87.48%
Wrong	3,979	12.52%
Total	31,789	

RESULT:

Thus, the Predict Customer churn in the telecom data Program has been Executed Successfully.

EXP :5 DATE:	PREDICT CUSTOMER CHURN IN THE TELECOM DATASET

AIM:

To Write a Program to Predict Customer churn in the telecom dataset.

a) Build Model using CHAID

b) Examine the CHAID Model

c) Apply the model to new data

PROCEDURE TO IMPLEMENTATION:

IMPORT DATASET:

STEP 1: Import the dataset telco x modeling data. Excel

STEP 2: Insert a Select node which will only keep the valid records You can insert a Table node and check the output.

STEP 3: From the Field Ops palette, add a Type node downstream from the Selectnode.

STEP 4: Edit the Type node.

STEP 5: Click the Types tab, if not already selected. Click the Read Values button.

STEP 6: Click the cell in the CHURN row, Role column and then click Target from the drop down. STEP 7: Click the cell in the RETENTION row, Role column and then click None from the drop down.

STEP 8: Click the cell in the DATA_KNOWN row, Role column and then click None from the drop down.

BUILD MODEL:

STEP 1: Click the Modeling tab, Add the CHAID node, located at the far right in the palette, downstreamfrom the Type node.

STEP 2: Run the CHAID node (right-click it and then click Run).

EXAMINE THE MODEL:

STEP 1: Edit the CHAID model nugget (the yellow diamond)

STEP 2: Click the Model tab, if not already selected.

STEP 3: Click the Viewer tab. Navigate to the root of the tree.

STEP 4: Click Preview.

STEP 5: Scroll all the way to the right in the Table output window.

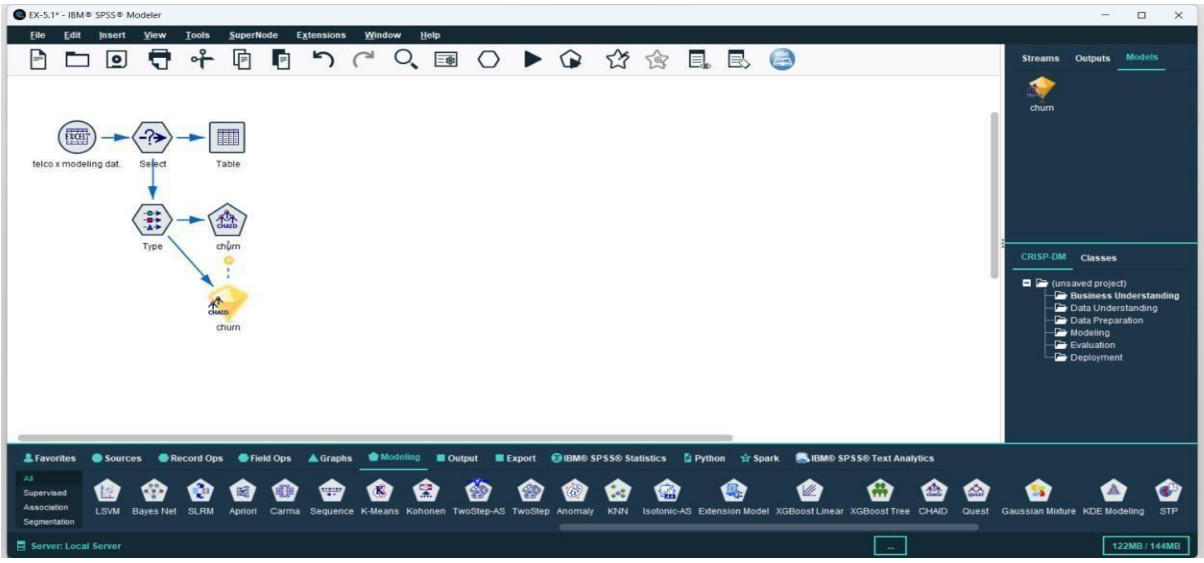
STEP 6: Close the CHAID model nugget; you will return to the stream.

STEP 7: You can also add an Analysis node from the Output palette in order to check accuracy.

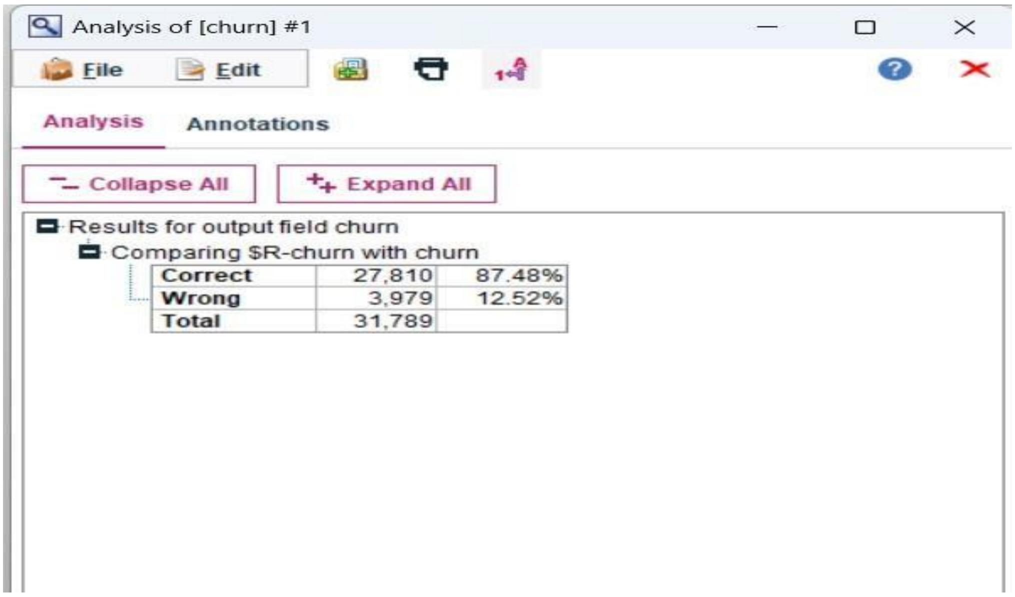
STEP 8: Run the Analysis Node.

OUTPUT:

Build Model using CHAID:


EX-5.1* - IBM® SPSS® Modeler
File Edit Insert View Tools SuperNode Extensions Window Help
telco x modeling dat..
Select
Table
Type
churn
churn
Streams Outputs Models
churn
CRISP-DM Classes
(unsaved project)
Business Understanding
Data Understanding
Data Preparation
Modeling
Evaluation
Deployment
Favorites Sources Record Ops Field Ops Graphs Modeling Output Export IBM® SPSS® Statistics Python Spark IBM® SPSS® Text Analytics
All
Supervised
Association
Segmentation
LSVM Bayes Net SLRM Apriori Carma Sequence K-Means Kohonen TwoStep-AS TwoStep Anomaly KNN Isotonic-AS Extension Model XGBoost Linear XGBoost Tree CHAID Quest Gaussian Mixture KDE Modeling STP
Server: Local Server
122MB / 144MB

Examine the CHAID model:

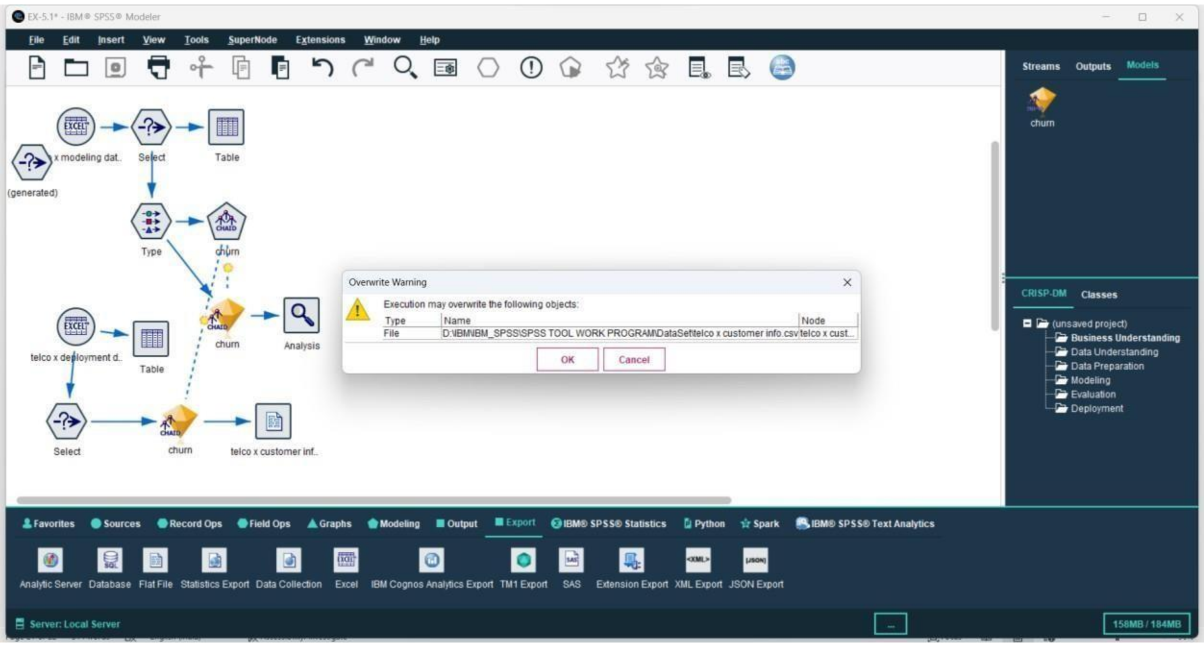


Results for output field churn

Comparing $R-churn with churn

Correct	27,810	87.48%
Wrong	3,979	12.52%
Total	31,789	

Apply the model to new data:



Overwrite Warning

Execution may overwrite the following objects:

Type	Name	Node
File	D:\IBM\IBM_SPSS\SPSS TOOL WORK PROGRAM\DataSet\telco x customer info.csv	telco x cust...

OK Cancel

RESULT:

Thus, the Predict Customer churn in the telecom data Program has been Executed Successfully.

EXP :6 DATE:	**CREATE HOMOGENEOUS GROUPS (CLUSTERS) OF CUSTOMERS BASED ON USAGE PATTERNS**

AIM:

To Create homogeneous groups of customers using Segmentation model in IBM SPSS Modeler.

PROCEDURE TO IMPLEMENTATION:

STEP 1: Insert **Type** node after importing **telco x modeling data.csv**

STEP 2: View the **Type** node. BILL_PEAK and BILL_OFFPEAK have role Input,so the clusters will be based on these two fields.

Records with similar values for BILL_PEAK and BILL_OFFPEAK will be put intothe same cluster.

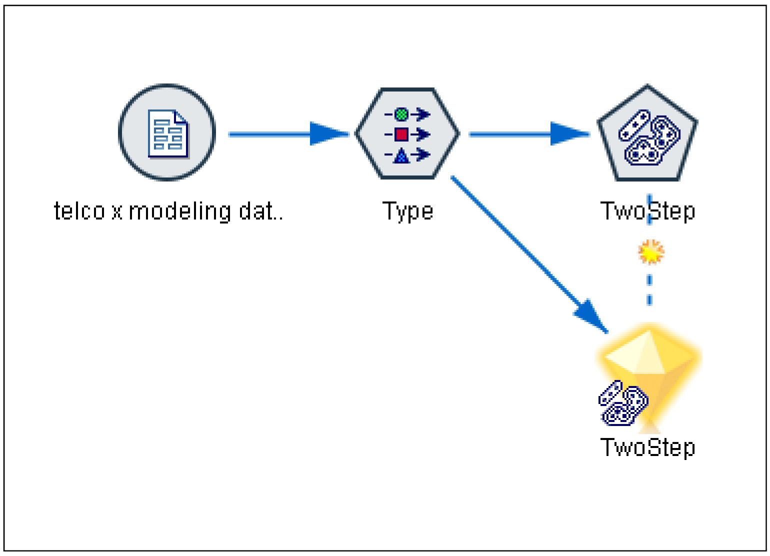
STEP 3: Click the **Modeling** palette, if not already selected. Click **Segmentation** sub palette at the left side.

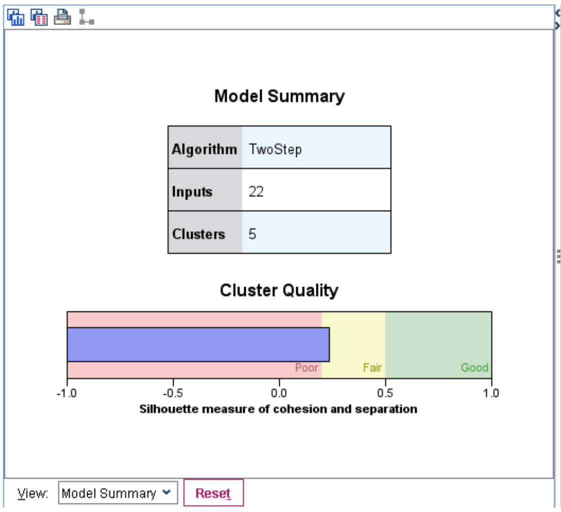
STEP 4: Add a **Two Step** node downstream from the **Type** node in the lower stream.

STEP 5: Run the **Two Step** node.

STEP 6: Edit the **Two Step** model nugget that wasgenerated.

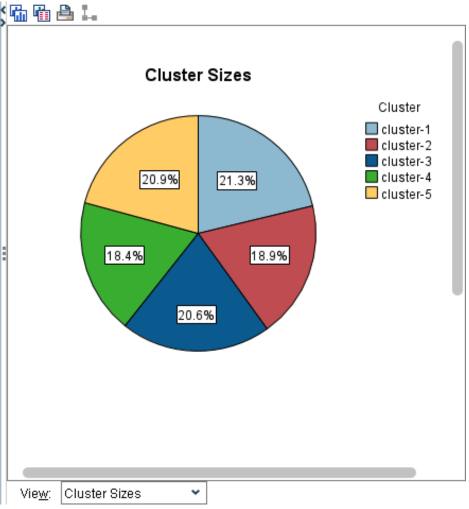
OUTPUT:





Model Summary

Algorithm	TwoStep
Inputs	22
Clusters	5



RESULT:

Thus, the Creating homogeneous groups of customers using Segmentation model Program has been Executed Successfully.

EXP :7 **DATE:**	**USING FUNCTIONS IN IBM SPSS MODELER**

AIM:

To derive new fields using ***Date Functions*** and ***String Function*** to cleanse and enrich a dataset that stores demographic and churn data on the company's customers in IBM SPSS Modeler.

ALGORITHM:

Use date functions to derive fields:

STEP1: From the **Sources** palette, double click the **Var. File** node to add it to the stream canvas.

STEP 2: Double-click the **Var. File** node to edit the **Var. File** node.

STEP 3: To the right of the **File** field, click the **Browse for file** button, navigate to the relevant folder, click the **telco x subset.csv** file, and then click **Open** to import the data. Do not close the **Var. File** dialog box.

STEP 4: In the **Var. File** dialog box, click **Preview**, and then scroll to the last fields in the **Preview** output window.

STEP 5: Click **OK** to close the **Preview** output window and Click **OK** to close the **Var. File** dialog box.

STEP 6: From the **Field Ops** palette, double-click the **Derive** node to add it downstream from the **Var. File** node.

STEP 7: Note: Placing node B downstream from node A means that the data flows from A to B.

STEP 8: Edit the **Derive** node. Under **Derive field**, type **MONTHS_CUSTOMER**.

STEP 9: Under **Formula**,enter **date_months_difference(CONNECT_DATE, END_DATE)**.

STEP 10: Note: Type the expression or use the Expression Builder to construct the expression. In this course "enter" refers to typing or using the Expression Builder, according to your preference. Here, when you use the **Expression Builder**, look for the **date_months_difference** function in the **Date and Time** function group.

STEP 11: Click **Preview**, and then scroll to the last fields in the **Preview** output window. The new field stores the number of months that elapsed between the two dates, as a real number. If you want the result as an integer, use a function such as round, intof or to_integer.

STEP 12: Close the **Preview** output window. Close the **Derive** dialog box.

STEP 13: From the **Field Ops** palette, add a **Derive** node downstream from the **Derive** node named **MONTHS_CUSTOMER**.

STEP 14: Edit the **Derive** node, and then set the **Mode** to **Multiple**.

STEP 15: The Derive dialog box reflects the change. The Derive field box is replaced by a Derive from box where the source fields are selected.

STEP 16: Under **Derive from**, click the **Pick from the set of available fields** button, Ctrl+click **CONNECT_DATE** and **END_DATE**, and then click **OK**.

STEP 17: Beside **Field name extension**, replace the current extension by **_MONTH**.

STEP 18: Under **Formula**, enter **datetime_month_name (datetime_month (@FIELD))**. If you use the **Expression Builder**, locate the **datetime_month** and **datetime_month_name** function in the **Date and Time** function group. Locate the **@FIELD** function in the **@ Functions** function group.

STEP 19: Click **Preview**, and then scroll to the last fields in the **Preview** output window.

STEP 20: Close the **Preview** output window. Close the **Derive** dialog box.

Use string functions to derive fields:

STEP 1: From the **Output** palette, add a **Table** node downstream from the **Derive** node named **_MONTH**.

STEP 2: Right-click the **Table** node and then click **Run** then Close the **Table** output window.

STEP 3: From the **Field Ops** palette, add a **Derive** node downstream from the **Derive** node named **_MONTH** (make room by repositioning the Table node, if preferred).

STEP 4: Edit the **Derive** node. Under **Derive field**, type **E-MAIL ADDRESS OK**.

STEP 5: Beside **Derive as**, click **Flag** from the list.

STEP 6: Under **True when**, enter **count_substring('E-MAIL ADDRESS', "@") = 1.** If you use the **Expression Builder**, locate the **count_substring** function in the **String** function group.

STEP 7: Click **Preview**, and then move **E-MAIL ADDRESS OK** next to **E-MAIL_ADDRESS** in the **Preview** output window. (Note: move E-MAIL ADDRESS OK by dragging it to the left, until it is just right from E-MAIL ADDRESS.)

STEP 8: Close the **Preview** output window. Close the **Derive** dialog box.

STEP 9: From the **Field Ops** palette, add a **Derive** node downstream from the **Derive** node named **E-MAIL ADDRESS OK**.

STEP 10: Edit the **Derive** node. Under **Derive field**, type **NO E-MAIL ADDRESS**.

STEP 11: Beside **Derive as**, click **Flag** from the list.

STEP 12: Under **True when**, enter **length ('E-MAIL ADDRESS') = 0**. If you use the **Expression Builder**, locate the **length** function in the **String** function group.

STEP13: Click **Preview**, and then move **NO E-MAIL ADDRESS** next to **E-MAIL ADDRESS**.

STEP 14: Close the **Preview** output window.

STEP15: Under **True when**, enter **length (trim ('E-MAIL ADDRESS')) = 0**. If you use the **Expression Builder**, locate the **length** and **trim** function in the **String** function group.

Click **Preview**.

STEP 16: Close the **Preview** output window. Then Close the **Derive** dialog box.

STEP 17: From the **Field Ops** palette, add a **Derive** node downstream from the **Derive** node named **NO E-MAIL ADDRESS**.

STEP 18: Edit the **Derive** node. Under **Derive field**, type **POSITION PERIOD**.

STEP 19: Under **Formula**, enter **locchar_back (., length ('E-MAIL ADDRESS'), 'E-MAIL ADDRESS')**. If you use the **Expression Builder**, locate the **locchar_back** and **length** function in the **String** function group.

STEP 20: Click **Preview**, and then move **POSITION PERIOD** next to **E-MAIL ADDRESS**.

STEP 21: Close the **Preview** output window. Close the **Derive** dialog box.

STEP 22: From the **Field Ops** palette, add a **Derive** node downstream from the **Derive** node named **POSITION PERIOD**.

STEP 23: Edit the **Derive** node**.** Under **Derive field** type **DOMAIN NAME**.

STEP 24: Beside **Derive as**, click **Conditional** from the list.

STEP 25: Under **If**, enter '**POSITION PERIOD' > 0**.

STEP 26: Under **Then**, enter **substring_between ('POSITION PERIOD' + 1, length ('E-MAIL ADDRESS'), 'E-MAIL ADDRESS')**. If you use the **Expression Builder**, locate the **substring_between** and **length** function in the **String** function group.

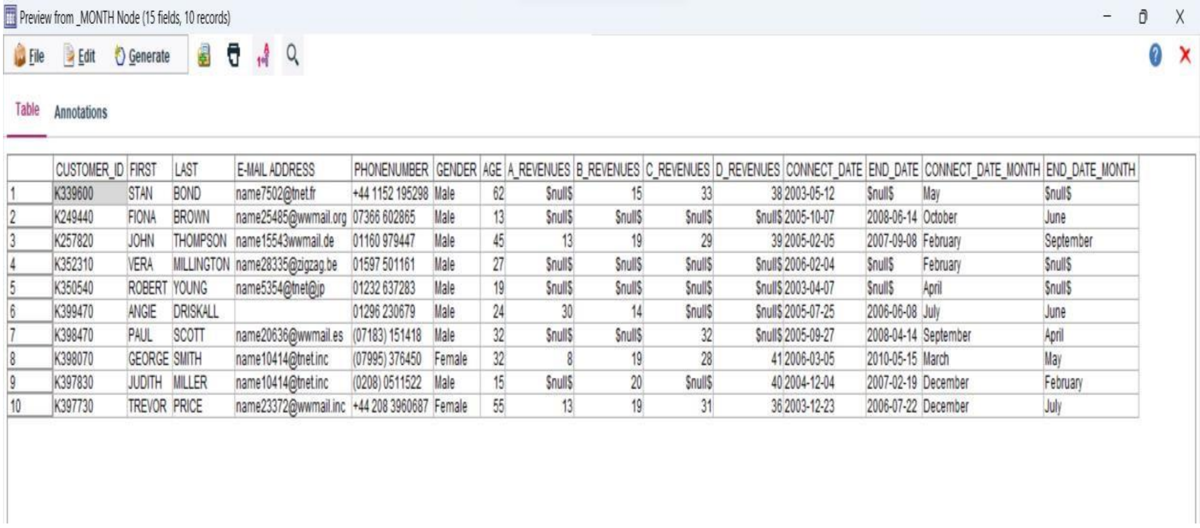
STEP 27: Under **Else**, type **undef**.

Click **Preview**, and then scroll to the last fields in the **Preview** output window.

STEP 28: Close the **Preview** output window. Then Close the **Derive** dialog box.

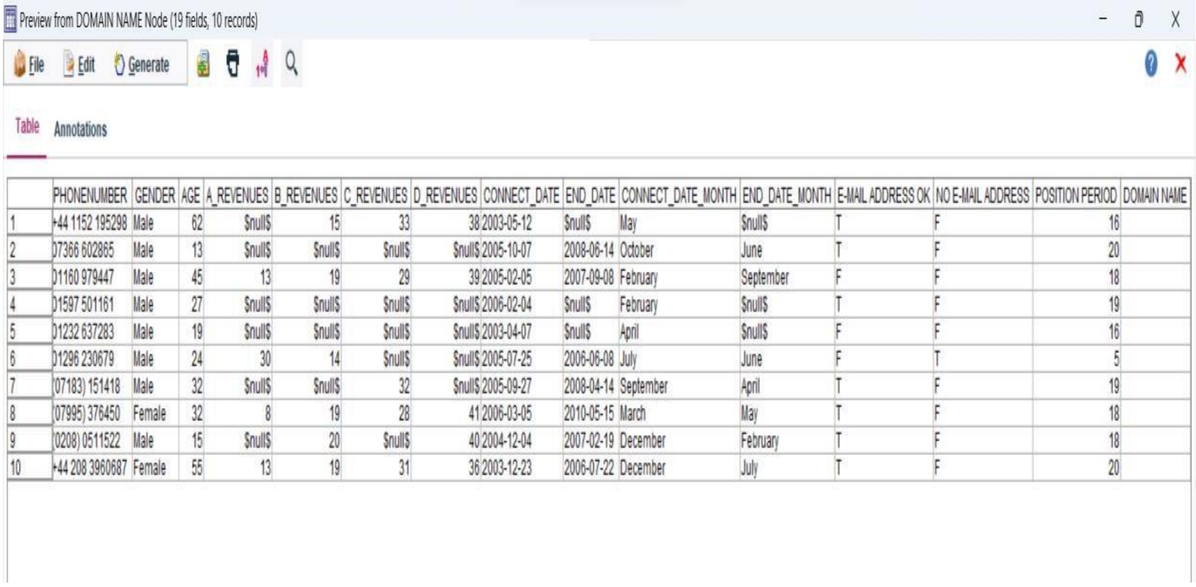
OUTPUT:

Use date functions to derive fields:



	CUSTOMER_ID	FIRST	LAST	E-MAIL ADDRESS	PHONENUMBER	GENDER	AGE	A_REVENUES	B_REVENUES	C_REVENUES	D_REVENUES	CONNECT_DATE	END_DATE	CONNECT_DATE_MONTH	END_DATE_MONTH
1	K339600	STAN	BOND	name7502@tnet.fr	+44 1152 195298	Male	62	$null$	15	33	38	2003-05-12	$null$	May	$null$
2	K249440	FIONA	BROWN	name25485@wwmail.org	07366 602865	Male	13	$null$	$null$	$null$	$null$	2005-10-07	2008-06-14	October	June
3	K257820	JOHN	THOMPSON	name15543wwmail.de	01160 979447	Male	45	13	19	29	39	2005-02-05	2007-09-08	February	September
4	K352310	VERA	MILLINGTON	name28335@zigzag.be	01597 501161	Male	27	$null$	$null$	$null$	$null$	2006-02-04	$null$	February	$null$
5	K350540	ROBERT	YOUNG	name5354@tnet@jp	01232 637283	Male	19	$null$	$null$	$null$	$null$	2003-04-07	$null$	April	$null$
6	K399470	ANGIE	DRISKALL		01296 230679	Male	24	30	14	$null$	$null$	2005-07-25	2006-06-08	July	June
7	K398470	PAUL	SCOTT	name20636@wwmail.es	(07183) 151418	Male	32	$null$	$null$	32	$null$	2005-09-27	2008-04-14	September	April
8	K398070	GEORGE	SMITH	name10414@tnet.inc	(07995) 376450	Female	32	8	19	28	41	2006-03-05	2010-05-15	March	May
9	K397830	JUDITH	MILLER	name10414@tnet.inc	(0208) 0511522	Male	15	$null$	20	$null$	40	2004-12-04	2007-02-19	December	February
10	K397730	TREVOR	PRICE	name23372@wwmail.inc	+44 208 3960687	Female	55	13	19	31	36	2003-12-23	2006-07-22	December	July

Use string functions to derive fields:



	PHONENUMBER	GENDER	AGE	A_REVENUES	B_REVENUES	C_REVENUES	D_REVENUES	CONNECT_DATE	END_DATE	CONNECT_DATE_MONTH	END_DATE_MONTH	E-MAIL ADDRESS OK	NO E-MAIL ADDRESS	POSITION PERIOD	DOMAIN NAME
1	+44 1152 195298	Male	62	$null$	15	33	38	2003-05-12	$null$	May	$null$	T	F	16	
2	07366 602865	Male	13	$null$	$null$	$null$	$null$	2005-10-07	2008-06-14	October	June	T	F	20	
3	01160 979447	Male	45	13	19	29	39	2005-02-05	2007-09-08	February	September	F	F	18	
4	01597 501161	Male	27	$null$	$null$	$null$	$null$	2006-02-04	$null$	February	$null$	T	F	19	
5	01232 637283	Male	19	$null$	$null$	$null$	$null$	2003-04-07	$null$	April	$null$	F	F	16	
6	01296 230679	Male	24	30	14	$null$	$null$	2005-07-25	2006-06-08	July	June	F	T	5	
7	(07183) 151418	Male	32	$null$	$null$	32	$null$	2005-09-27	2008-04-14	September	April	T	F	19	
8	(07995) 376450	Female	32	8	19	28	41	2006-03-05	2010-05-15	March	May	T	F	18	
9	(0208) 0511522	Male	15	$null$	20	$null$	40	2004-12-04	2007-02-19	December	February	T	F	18	
10	+44 208 3960687	Female	55	13	19	31	36	2003-12-23	2006-07-22	December	July	T	F	20	

RESULT:

Thus, the Create Segmentation Model Program has been Executed Successfully

EXP :8 **DATE:**	**ADD FIELDS TO THE TELECOMMUNICATIONS DATA**

AIM:

To write a Program for Add Fields to the Telecommunication data.

a) Drive fields as formula
b) Derive fields as flag or nominal.

ALGORITHM:

Derive fields as formula:

STEP 1: Import the dataset **telco x data.txt**

STEP 2: From the **Field Ops** palette, add a **Derive** node downstream from the **Type** node.

STEP 3: Edit the **Derive** node. Click the **Settings** tab, if not already selected.

STEP 4: In the **Derive field** box, type **BILL_PEAK**.

STEP 5: Click the **Derive as** drop down. From the **Derive as** drop down, click **Formula**, if not already selected.

STEP 6: Click the **Field type** drop down. Click the **Launch expression builder** button.

STEP 7: In the **Formula** box, enter **PEAK_MINS * PEAK_RATE/100**, by typing it or by pasting the field names from the list of fields, whatever you feel comfortable with.

STEP 8: Click the **Check** button. Click **OK** to close the **Expression Builder**.

STEP 9: From the **Field Ops** palette, add a **Derive** node downstream from the **Derive** node named **BILL_PEAK**.

STEP 10: Edit the **Derive** node. Click the **Settings** tab, if not already selected.

STEP 11: Derive field box: **BILL_OFFPEAK**

STEP 12: Derive as: **Formula** (the default)

STEP 13: Field type: **<Default>**; the field will then be auto-typed as Continuous

STEP 14: Expression: **OFFPEAK_MINS * OFFPEAK_RATE/100** (if preferred, use the Expression Builder)

Close the **Derive** dialog box.

STEP 15: From the **Field Ops** palette, add a **Derive** node downstream from the **Derive** node named **BILL_OFFPEAK.**

STEP 16: Edit the **Derive** node. Click the **Settings** tab, if not already selected.

STEP 17: Derive field box: **BILL_TOTAL**

STEP 18: Derive as: **Formula** (the default)

STEP 19: Field type:<Default>; the field will then be auto-typed as Continuous

STEP 20: Expression: **BILL_PEAK + BILL_OFFPEAK** (if preferred, use the Expression Builder)

STEP 21: Click **Preview**. Then Close the **Preview** output window.

Close the **Derive** dialog box.

Derive fields as flag or nominal:

STEP 1: From the **Field Ops** palette, add a **Derive** node downstream from the **Derive** node named **BILL_TOTAL**.

STEP 2: Edit the **Derive** node. Click the **Settings** tab, if not already selected.

STEP 3: Derive field box: **BILL_GT_0**

STEP 4: Derive as: **Flag**

STEP 5: Field type: **Flag** (should be set automatically to Flag when you choose Derive as: Flag)

STEP 6: True value: **T** (the default) and **False value**: **F** (the default)

STEP 7: True when box: **BILL_TOTAL > 0.** Close the **Derive** dialog box.

STEP 8: From the **Field Ops** palette, add a **Derive** node downstream from the **Derive** node named **BILL_GT_0**.

STEP 9: Edit the **Derive** node Click the **Settings** tab, if not already selected.

STEP 10: Derive field box: **SEGMENT**

STEP 11: Derive as: **Nominal** and **Field type**: **Ordinal**

STEP 12: Click the cell in the **Set field to** column and then type **1**.

STEP 13: Click the cell in the **If this condition is true** column and then type **BILL_TOTAL <= 100**.

STEP 14: Repeat the previous two steps for the following values and expressions:

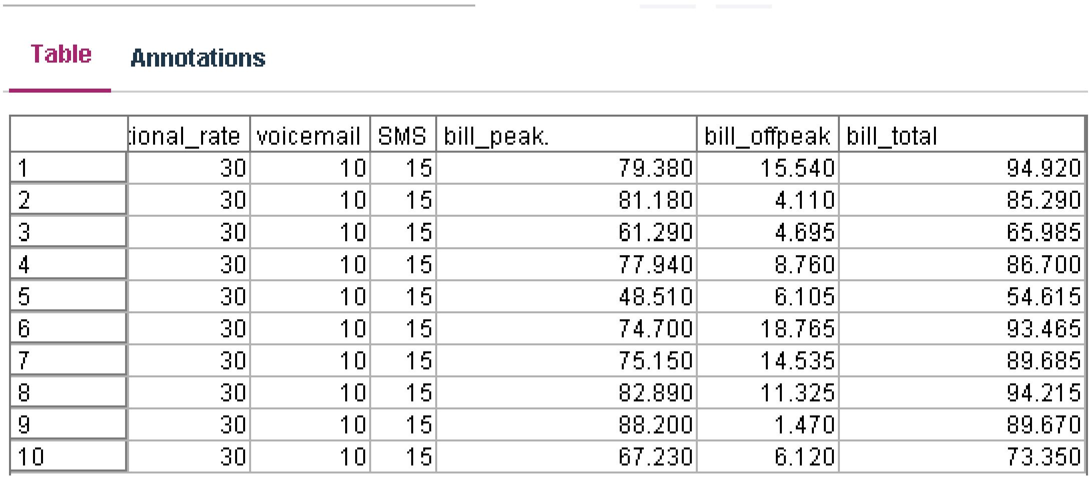
STEP 15: Set field to: 2 If this condition is true: BILL_TOTAL <= 200

STEP 16: Set field to: 3 If this condition is true: BILL_TOTAL > 200

STEP 17: In the **Default value** box, type **undef**. Then Close the **Derive** dialog box.

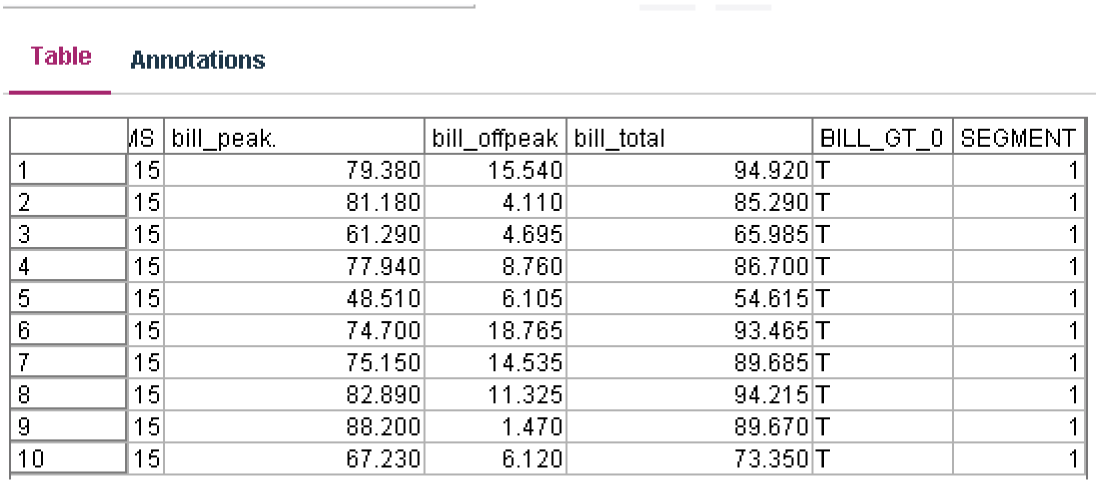
OUTPUT:

Drive fields as formula:

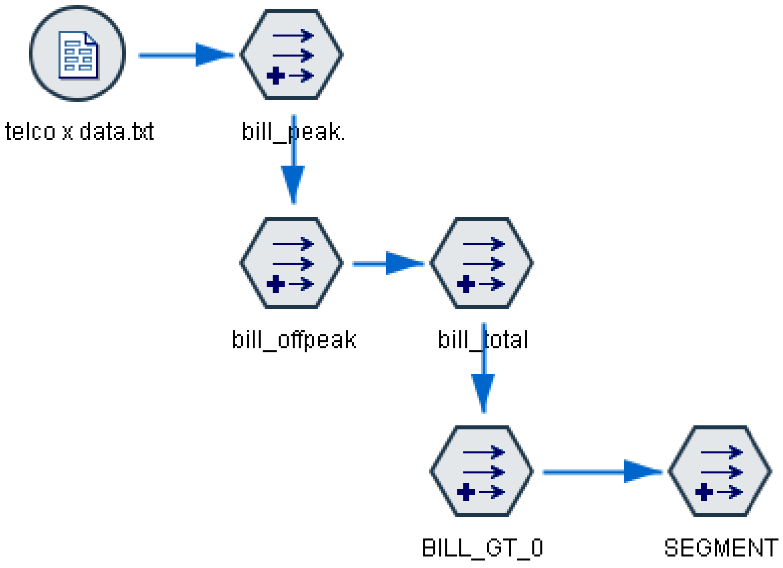


	ional_rate	voicemail	SMS	bill_peak.	bill_offpeak	bill_total
1	30	10	15	79.380	15.540	94.920
2	30	10	15	81.180	4.110	85.290
3	30	10	15	61.290	4.695	65.985
4	30	10	15	77.940	8.760	86.700
5	30	10	15	48.510	6.105	54.615
6	30	10	15	74.700	18.765	93.465
7	30	10	15	75.150	14.535	89.685
8	30	10	15	82.890	11.325	94.215
9	30	10	15	88.200	1.470	89.670
10	30	10	15	67.230	6.120	73.350

Derive fields as flag or nominal:



	IS	bill_peak.	bill_offpeak	bill_total	BILL_GT_0	SEGMENT
1	15	79.380	15.540	94.920	T	1
2	15	81.180	4.110	85.290	T	1
3	15	61.290	4.695	65.985	T	1
4	15	77.940	8.760	86.700	T	1
5	15	48.510	6.105	54.615	T	1
6	15	74.700	18.765	93.465	T	1
7	15	75.150	14.535	89.685	T	1
8	15	82.890	11.325	94.215	T	1
9	15	88.200	1.470	89.670	T	1
10	15	67.230	6.120	73.350	T	1



RESULT:

Thus, the Add fields to the Telecommunication data Program have been Executed Successfully.

EXP :9 **DATE:**	**CREATE A LINEAR REGRESSION MODEL TO PREDICT EMPLOYEE SALARIES**

AIM:

To Create a Linear Regression Model to predict Employee Salaries.

ALGORITHM:

Import and examine the data

STEP1: From the **Sources** palette, add a **Var. File** node to a blank stream canvas, edit the node, point **to employee_data.txt**, and then close the **Var. File** dialog box.

STEP2: From the **Output** palette, add a **Table** node downstream from the **Var. File** node, run it, and then examine the output. The dataset is comprised of 474 employees.

Close the **Table** output window.

STEP 3: From the **Output** palette, add a **Data Audit** node downstream from the **Var. File** node, run it, and then examine the output.

Set measurement levels and roles:

STEP 1: From **Field Ops**, add a **Type** node downstream from the **Var. File** node.

STEP 2: Edit the **Type** node. Click **Read Values**

STEP 3: set the **Measurement** for **educational_level** to **Ordinal**

STEP 4: The **Role** from **gender** to **months_previous_experience** is set to **Input**

STEP 5: set the **Role** for **current_salary** to **Target**

Create Linear Regression Model:

STEP 1: From the **Modeling** palette, add a **Linear** node downstream from the **Type** node.

STEP 2: Edit the **Linear** node. Click the **Build Options** tab

STEP 3: click the **Basics** item and clear the **Automatically prepare data** check box

STEP 4: click the **Model Selection** item and set the Model Selection method to **Include all predictors**

STEP 5: click **Run**

STEP 6: Edit the generated model nugget, and then click the Model Summary item in the pane on the left.

STEP 7: Click the **Predictor Importance** item in the pane on the left.

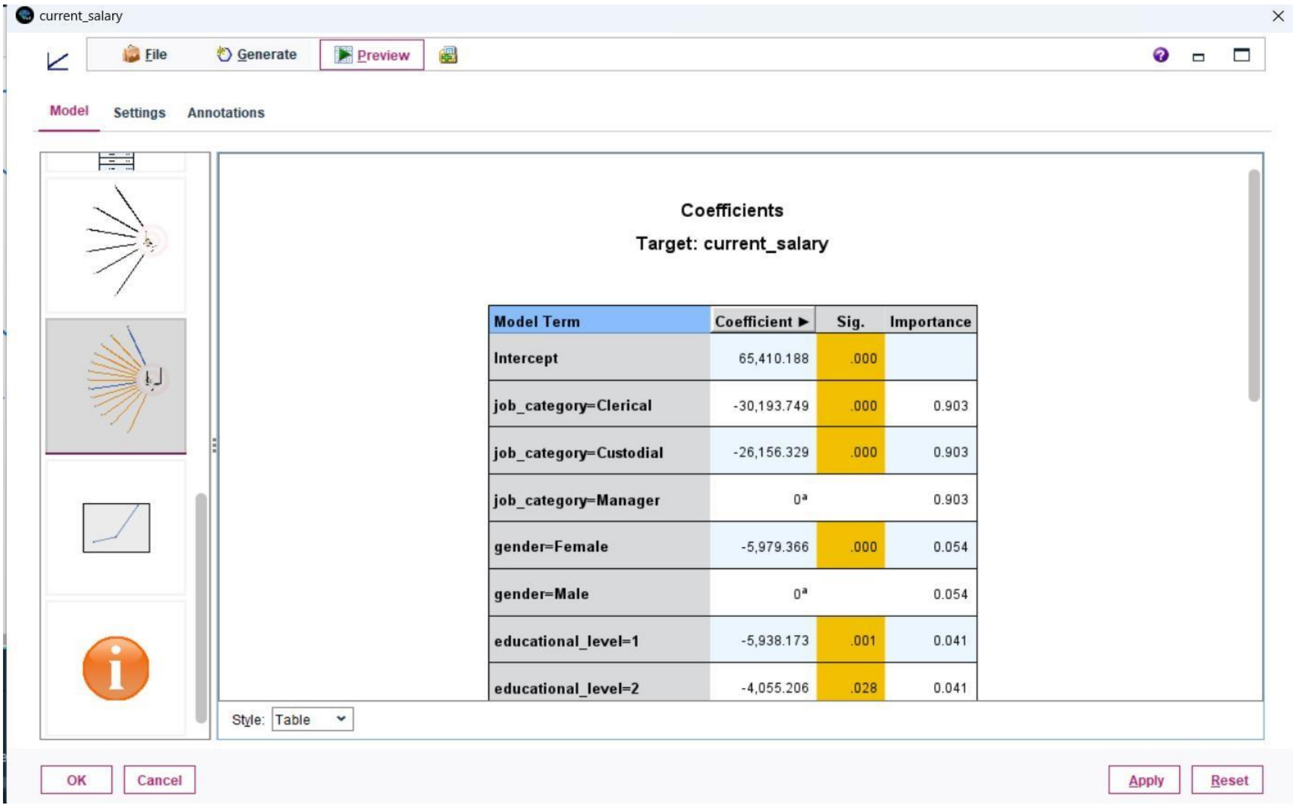
STEP 8: The job_category field is by far the most important predictor. Gender is the second most important field. Region and age are least important.

STEP 9: Click the **Predicted by Observed** item in the pane on the left.

STEP 10: The points are not scattered around the diagonal and the predicted values seem to break up in two categories.

STEP 11: Click the **Coefficients by Observed** item in the pane on the left, and then, from the Style list, select Table.

OUTPUT:



Coefficients

Target: current_salary

Model Term	Coefficient ▶	Sig.	Importance
Intercept	65,410.188	.000	
job_category=Clerical	-30,193.749	.000	0.903
job_category=Custodial	-26,156.329	.000	0.903
job_category=Manager	0ª		0.903
gender=Female	-5,979.366	.000	0.054
gender=Male	0ª		0.054
educational_level=1	-5,938.173	.001	0.041
educational_level=2	-4,055.206	.028	0.041

RESULT:

Thus, the Create Linear regression model to predict Employee Salaries Program has been Executed Successfully.

EXP :10 **DATE:**	**USE LOGISTIC REGRESSION TO PREDICT RESPONSE TO A CHARITY PROMOTION CAMPAIGN**

AIM:

To write a Use Logistic Regression to Predict Response to a Charity Promotion Campaign

ALGORITHM:

Import and examine the data

STEP 1: From the Sources palette, double-click the Var. File node to add it to the stream. Import the dataset **charity.csv**

STEP 2: From the **Output** palette, add a **Data Audit** node downstream from the **Var. File** node, run the **Data Audit** node

STEP 3: double-click the Sample Graph for the **response to campaign** field.

Partition the data and set the roles:

STEP 1: From the **Field Ops** palette, add a **Partition** node downstream from the **Var. File** node,

STEP 2: Set the **Training partition size to 70%** and the **Testing partition size to 30%**. Ensure that the **Repeatable partition assignment** option is enabled, with seed value **1234567**.

STEP 3: From the **Field Ops** palette, add a Type node downstream from the Partition node.

STEP 4: Edit the **Type** node, and then click the **Read Values** button. The Values column is populated with values from the data.

STEP 5: Set the **Role** for **gender, age, mosaic bands, pre-campaign expenditure, and pre-campaign visits** to **Input**

STEP 6: Set the **Role** for **response to campaign** to **Target**

STEP 7: Ensure that the **Role** for the **Partition** field is set to **Partition**

STEP 8: Set the **Role** for all other fields to **None**

Create the Logistic Regression Models:

STEP 1: From the **Modeling** palette, add a **Logistic** node downstream from the **Type** node.

STEP 2: Edit the **Logistic** node and **click** the **Model** tab

STEP 3: For **Procedure**, select the **Binomial** option

STEP 4: close the **Logistic** dialog box

STEP 5: Add a second **Logistic** node downstream from the **Type** node.

STEP 6: Edit the second **Logistic** node, and then: click the **Model** tab

STEP 7: For **Procedure**, select the **Binomial** option

STEP 8: below **Categorical inputs**, select **mosaic bands**, and for **Base Category**, select **First**

STEP 9: click the **Annotations** tab, select the **Custom** option, and type **custom** close the **Logistic** dialog box

STEP 10: Select the two **Logistic** nodes, right-click one of them, and **click Run Selection**.

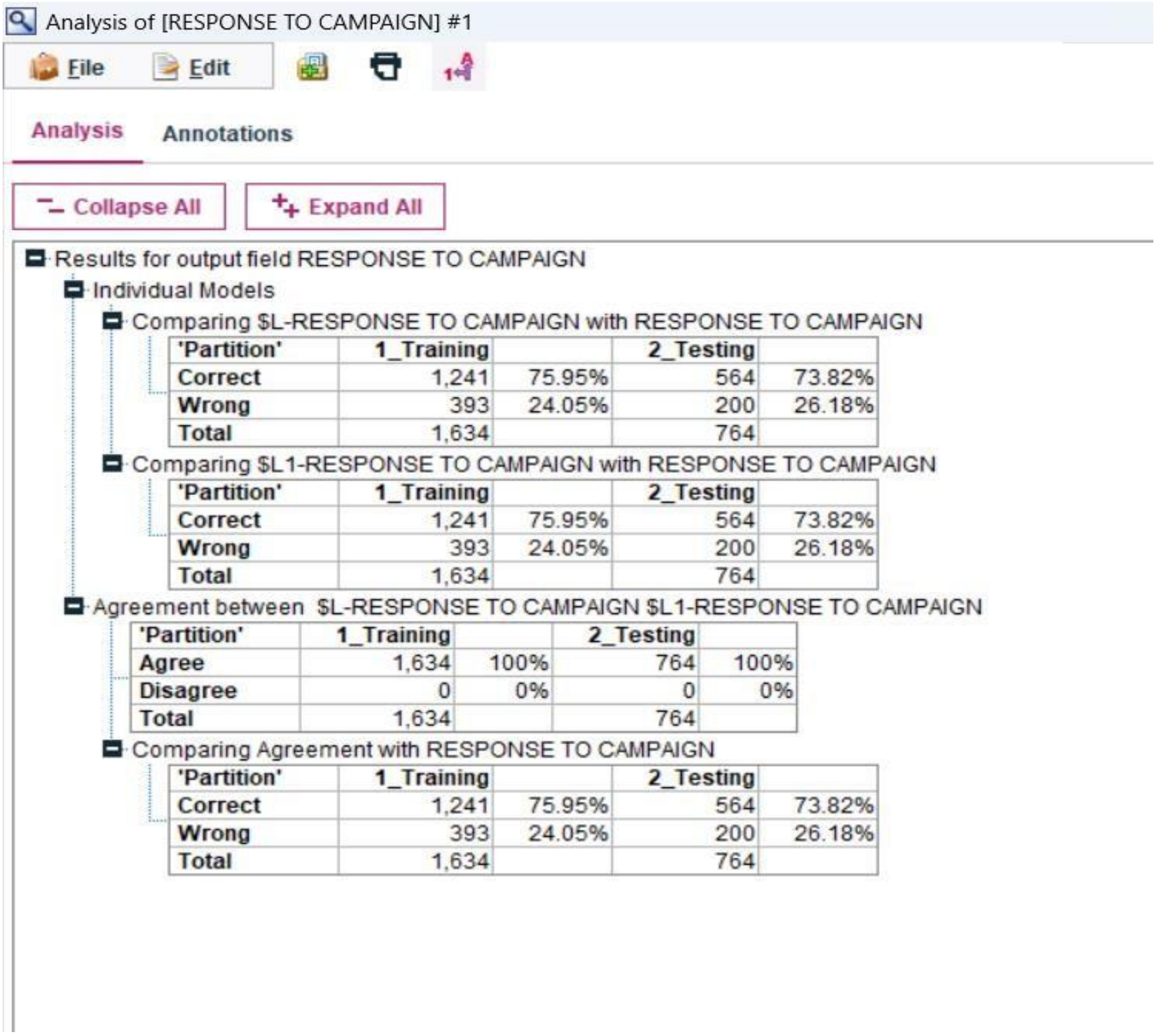
STEP 11: Edit the Logistic model nugget named response to campaign, click the Advanced tab, and scroll down to the Variables in the Equation table (the last table in the output).

STEP 12: Close the Logistic output window.

STEP 13: Edit the Logistic model nugget named custom, click the Advanced tab, and scroll to the Categorical Variables Coding's table.

STEP 14: You can add an **Analysis** node at the end and check accuracy levels

OUTPUT:



Results for output field RESPONSE TO CAMPAIGN

Individual Models

Comparing $L-RESPONSE TO CAMPAIGN with RESPONSE TO CAMPAIGN

'Partition'	1_Training		2_Testing	
Correct	1,241	75.95%	564	73.82%
Wrong	393	24.05%	200	26.18%
Total	1,634		764	

Comparing $L1-RESPONSE TO CAMPAIGN with RESPONSE TO CAMPAIGN

'Partition'	1_Training		2_Testing	
Correct	1,241	75.95%	564	73.82%
Wrong	393	24.05%	200	26.18%
Total	1,634		764	

Agreement between $L-RESPONSE TO CAMPAIGN $L1-RESPONSE TO CAMPAIGN

'Partition'	1_Training		2_Testing	
Agree	1,634	100%	764	100%
Disagree	0	0%	0	0%
Total	1,634		764	

Comparing Agreement with RESPONSE TO CAMPAIGN

'Partition'	1_Training		2_Testing	
Correct	1,241	75.95%	564	73.82%
Wrong	393	24.05%	200	26.18%
Total	1,634		764	

RESULT:

Thus, the Use of Logistic Regression to Predict Response to a Charity Promotion Campaign Program has been Executed Successfully.