

SPSS PRACTICAL

Name: Ganesh Agrahari

Class: BCA DS 33

Roll No.: 1230258176

Submitted To: Mr. Robin Tyagi

INDEX

SPSS Practical 8

This practical includes:

- Objective and Introduction
- Theoretical Background
- Methodology and Procedure
- Data Analysis
- Results and Interpretation
- Screenshots and Output
- Conclusion
- References

Practical: 8

Definition:

You work for a telecommunications firm where you have to cleanse and enrich a dataset that stores demographic and churn data on the company's customers. Using the transformed dataset

Outcomes/Learning:

- Learned to derive new calculated and conditional fields using **CLEM expressions**.
- Understood how to perform **data cleansing and enrichment** for improved data quality.
- Gained experience in **string, date, and numeric field manipulation** using Derive Nodes.
- Prepared a complete and reliable dataset for **churn analysis and predictive modelling**.

Required Tool:

IBM SPSS MODELER

Working:

In this practical, the goal is to **cleanse, transform, and enrich** a customer dataset to prepare it for analysis and modelling.

The dataset (*telco x subset.dat*) is imported using a **Var. File Node**, and several **Derive Nodes** are applied to create new calculated and conditional fields using **CLEM (Control Language for Expression Manipulation)**.

New fields such as *MONTHS_CUSTOMER*, *DOMAIN NAME*, *MEAN_REVENUES*, and *CHURN* are generated to improve data consistency, fill missing information, and enhance analytical readiness.

The main nodes used are:

- Var. File Node** – to import the dataset.
- Derive Node** – to create calculated, conditional, and flag fields using CLEM expressions.
- Table Node** – to verify and display transformed data.

Step 1: Importing the Dataset

- Open IBM SPSS Modeler → create a new stream.
- Add a **Var. File Node** from the Sources tab → browse and select *telco x subset.dat*.
- Click **Apply** → **OK** to load the dataset.
- Attach a **Table Node** → click **Run** to view the imported records.

Step 2: Creating the MONTHS_CUSTOMER Field

Attach a **Derive Node** to the Var. File Node.

Configure as follows:

- Rename the field to **MONTHS_CUSTOMER**.
- Set **Derive As = Formula** and **Field Type = Default**.
- Enter the expression:
- `date_months_difference(CONNECT_DATE, END_DATE)`

Click Apply → OK, then attach a Table Node to verify results.

Step 3: Creating the _MONTHS Field

Connect another Derive Node to the MONTHS_CUSTOMER Node.

Configure as follows:

- Rename the field to **_MONTHS**.
- Set **Mode = Multiple**, **Derive As = Formula**, **Field Type = Default**.
- Enter the expression:
- `datetime_month_name(datetime_month(@FIELD))`

Click **Apply → OK**, then view results using a **Table Node**.

Step 4: Creating the E-MAIL ADDRESS OK Field

Attach a new Derive Node to the **_MONTHS** Node.

Configure as follows:

- Rename it to **E-MAIL ADDRESS OK**.
- Set **Mode = Single**, **Derive As = Flag**, **Field Type = Flag**.
- Enter the expression:
- `count_substring('E-MAIL ADDRESS', '@') = 1`

Click **Apply → OK**, and check results with a **Table Node**.

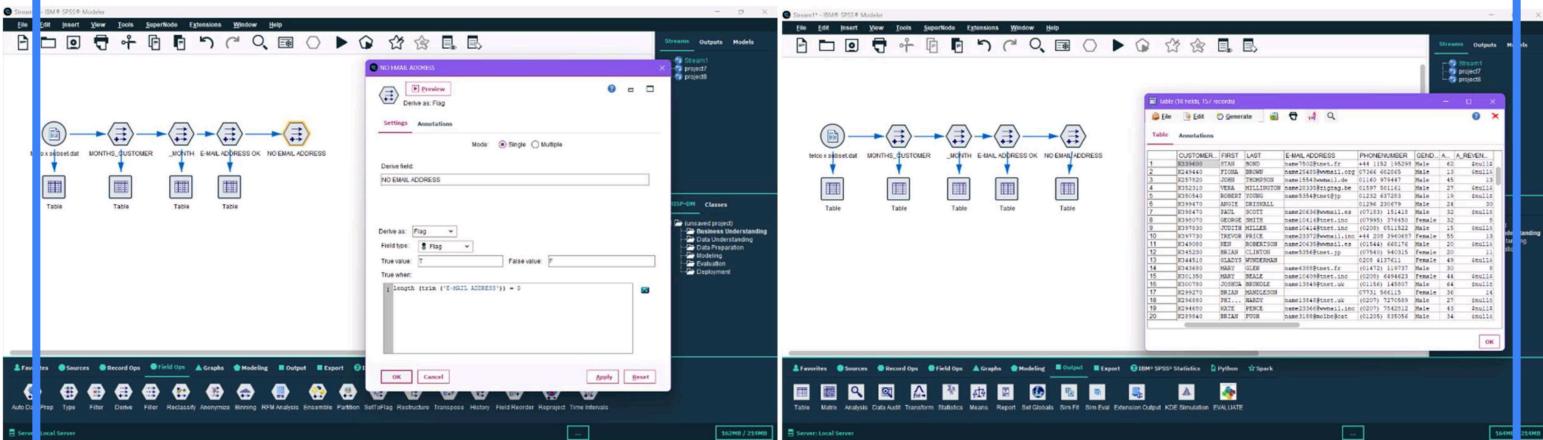
Step 5: Creating the NO EMAIL ADDRESS Field

Connect another **Derive Node** to the E-MAIL ADDRESS OK Node.

Configure as follows:

- Rename to **NO EMAIL ADDRESS**.
- Set **Mode = Single, Derive As = Flag, Field Type = Flag**.
- Enter the expression:
- $\text{length(trim('E-MAIL ADDRESS'))} = 0$

Click **Apply → OK**, and view results with a **Table Node**.



The screenshot shows the SPSS Modeler interface with a data flow. The flow starts with a 'Table' node, followed by a 'MONTH' node, then a 'CUSTOMER' node. The 'CUSTOMER' node is connected to an 'E-MAIL ADDRESS OK' node. This is followed by a 'NO EMAIL ADDRESS' node, which is then connected to a final 'Table' node. The 'NO EMAIL ADDRESS' node has its 'Derive as' set to 'Flag', 'Field type' to 'Flag', and the 'True value' field contains '0'. The 'Formula' field contains the expression `length(trim('E-MAIL ADDRESS')) = 0`. To the right of the flow, a 'Table' node is shown with 20 records. The columns are CUSTOMER, FIRST, LAST, E-MAIL ADDRESS, PHONE NUMBER, and GEND. A. A. REVEN. The 'E-MAIL ADDRESS' column shows various email addresses, and the 'NO EMAIL ADDRESS' column shows values 0 or 1 indicating if the email is empty or not.

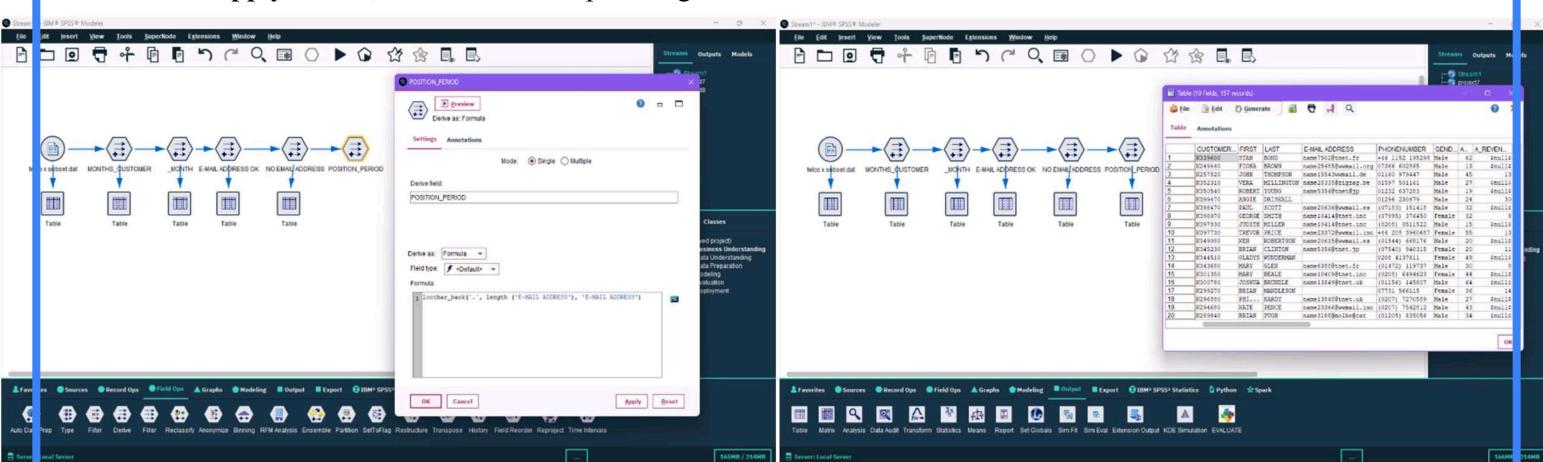
Step 6: Creating the POSITION_PERIOD Field

Attach a **Derive Node** to the NO EMAIL ADDRESS Node.

Configure as follows:

- Rename to **POSITION_PERIOD**.
- Set **Mode = Single, Derive As = Formula, Field Type = Default**.
- Enter the formula:
- $\text{locchar_back('!', length('E-MAIL ADDRESS'), 'E-MAIL ADDRESS')}$

Click **Apply → OK**, and check the output using a **Table Node**.



The screenshot shows the SPSS Modeler interface with a data flow. The flow starts with a 'Table' node, followed by a 'MONTH' node, then a 'CUSTOMER' node. The 'CUSTOMER' node is connected to an 'E-MAIL ADDRESS OK' node. This is followed by a 'NO EMAIL ADDRESS' node, then a 'POSITION_PERIOD' node, and finally a 'Table' node. The 'POSITION_PERIOD' node has its 'Derive as' set to 'Formula', 'Field type' to 'Default', and the 'Formula' field contains the expression `locchar_back('!', length('E-MAIL ADDRESS'), 'E-MAIL ADDRESS')`. To the right of the flow, a 'Table' node is shown with 20 records. The columns are CUSTOMER, FIRST, LAST, E-MAIL ADDRESS, POSITION_PERIOD, and PHONE NUMBER. The 'E-MAIL ADDRESS' column shows various email addresses, and the 'POSITION_PERIOD' column shows the position of the '@' symbol in the email address.

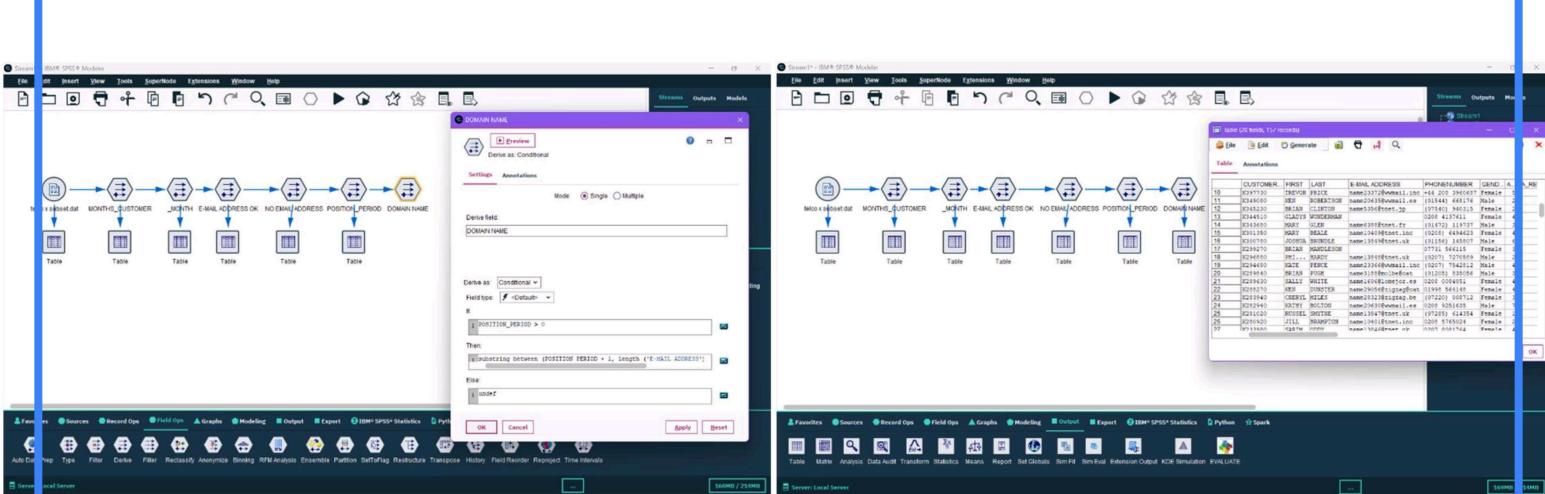
Step 7: Creating the DOMAIN NAME Field

Add another **Derive Node** connected to the POSITION_PERIOD Node.

Configure as follows:

- Rename to **DOMAIN NAME**.
- Set **Mode = Single, Derive As = Conditional, Field Type = Default**.
- Enter the expression:
- If $\text{POSITION_PERIOD} > 0$ Then $\text{substring_between}(\text{POSITION_PERIOD} + 1, \text{length('E-MAIL ADDRESS')})$ Else undef

Click **Apply → OK**, and validate results using a **Table Node**.



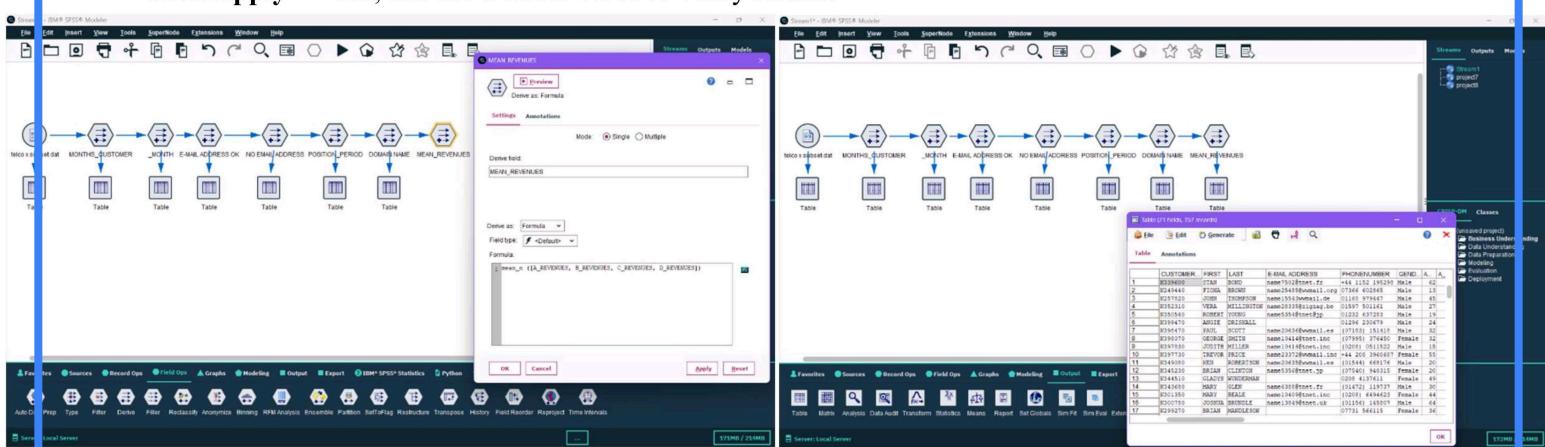
Step 8: Creating the MEAN_REVENUES Field

Attach a **Derive Node** to the **DOMAIN NAME** node.

Configure as follows:

- Rename to **MEAN_REVENUES**.
- Set **Mode = Single**, **Derive As = Formula**, **Field Type = Default**.
- Enter the expression:
- $\text{mean_n}([\text{A_REVENUES}, \text{B_REVENUES}, \text{C_REVENUES}, \text{D_REVENUES}])$

Click **Apply → OK**, and use a **Table Node** to verify results.



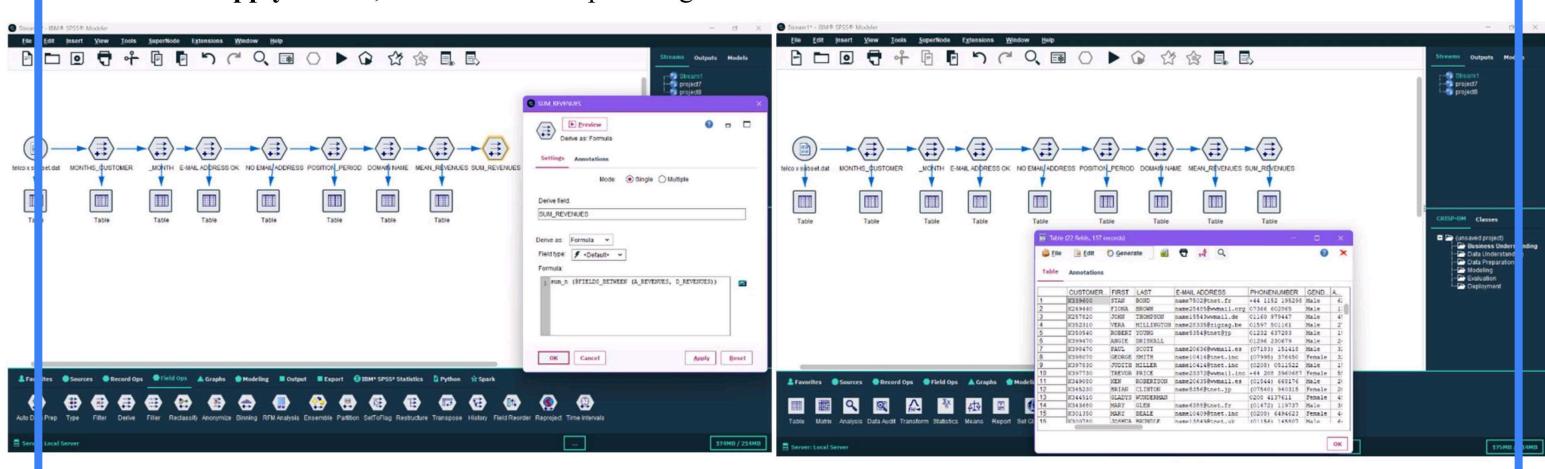
Step 9: Creating the SUM_REVENUES Field

Connect another **Derive Node** to the **MEAN_REVENUES** Node.

Configure as follows:

- Rename to **SUM_REVENUES**.
- Set **Mode = Single**, **Derive As = Formula**, **Field Type = Default**.
- Enter the expression:
- $\text{sum_n}(@\text{FIELDS_BETWEEN}(\text{A_REVENUES}, \text{D_REVENUES}))$

Click **Apply → OK**, and check the output using a **Table Node**.



Step 10: Creating the SUM_REVENUES_OK Field

Attach a **Derive Node** to the SUM_REVENUES Node.

Configure as follows:

- Rename to **SUM_REVENUES_OK**.
- Set **Mode = Single**, **Derive As = Conditional**, **Field Type = Default**.
- Enter the expression:
- If $\text{count_non_nulls}(@\text{FIELDS_BETWEEN}(\text{A_REVENUES}, \text{D_REVENUES})) > 0$ Then $\text{sum_n}(@\text{FIELDS_BETWEEN}(\text{A_REVENUES}, \text{D_REVENUES}))$ Else **undef**

Click **Apply → OK**, and view results using a **Table Node**.

The screenshot shows the IBM SPSS Modeler interface with a data flow. The flow starts with an 'Infix xlsx' input node, followed by a 'MONTHS_CUSTOMER' node, 'MONTH' node, 'E-MAIL ADDRESS OK' node, 'NO E-MAIL ADDRESS' node, 'POSITION' node, 'PERIOD' node, 'DOMAIN' node, 'MEAN REVENUES' node, 'SUM REVENUES' node, and finally the 'SUM REVENUES_OK' derive node. The 'SUM REVENUES_OK' node is highlighted in purple, indicating it is selected. A preview window for this node is open, showing the 'Derive as: Conditional' settings. The 'Mode' is set to 'Single', and the 'Derive field' is 'SUM_REVENUES_OK'. The 'Field type' is 'Default'. The 'Then:' condition is 'count_non_nulls(@FIELDS_BETWEEN(A_REVENUES, D_REVENUES)) > 0'. The 'Else:' condition is 'undef'. Below the preview window, there are 'OK', 'Cancel', 'Apply', and 'Reset' buttons. To the right of the preview window, a table node is connected to the 'SUM REVENUES_OK' node, and a 'Table (24 fields, 157 records)' output window is open, displaying the results of the data transformation.

Step 11: Creating the CHURN Field

Connect a **Derive Node** to the SUM_REVENUES_OK Node.

Configure as follows:

- Rename to **CHURN**.
- Set **Mode = Single**, **Derive As = Flag**, **Field Type = Flag**.
- Enter the formula:
- $\text{not}(@\text{NULL}(\text{END_DATE}))$

Click **Apply → OK**, then attach a **Table Node** to view the final results.

The screenshot shows the IBM SPSS Modeler interface with a data flow. The flow starts with an 'Infix xlsx' input node, followed by a 'MONTHS_CUSTOMER' node, 'MONTH' node, 'E-MAIL ADDRESS OK' node, 'NO E-MAIL ADDRESS' node, 'POSITION' node, 'PERIOD' node, 'DOMAIN' node, 'MEAN REVENUES' node, 'SUM REVENUES' node, 'SUM REVENUES_OK' node, and finally the 'CHURN' derive node. The 'CHURN' node is highlighted in purple. A preview window for this node is open, showing the 'Derive as: Flag' settings. The 'Mode' is set to 'Single', and the 'Derive field' is 'CHURN'. The 'Field type' is 'Flag'. The 'True value' is '1' and the 'False value' is '0'. The 'True when' condition is 'not(@NULL(END_DATE))'. Below the preview window, there are 'OK', 'Cancel', 'Apply', and 'Reset' buttons. To the right of the preview window, a table node is connected to the 'CHURN' node, and a 'Table (24 fields, 157 records)' output window is open, displaying the final results of the data transformation.