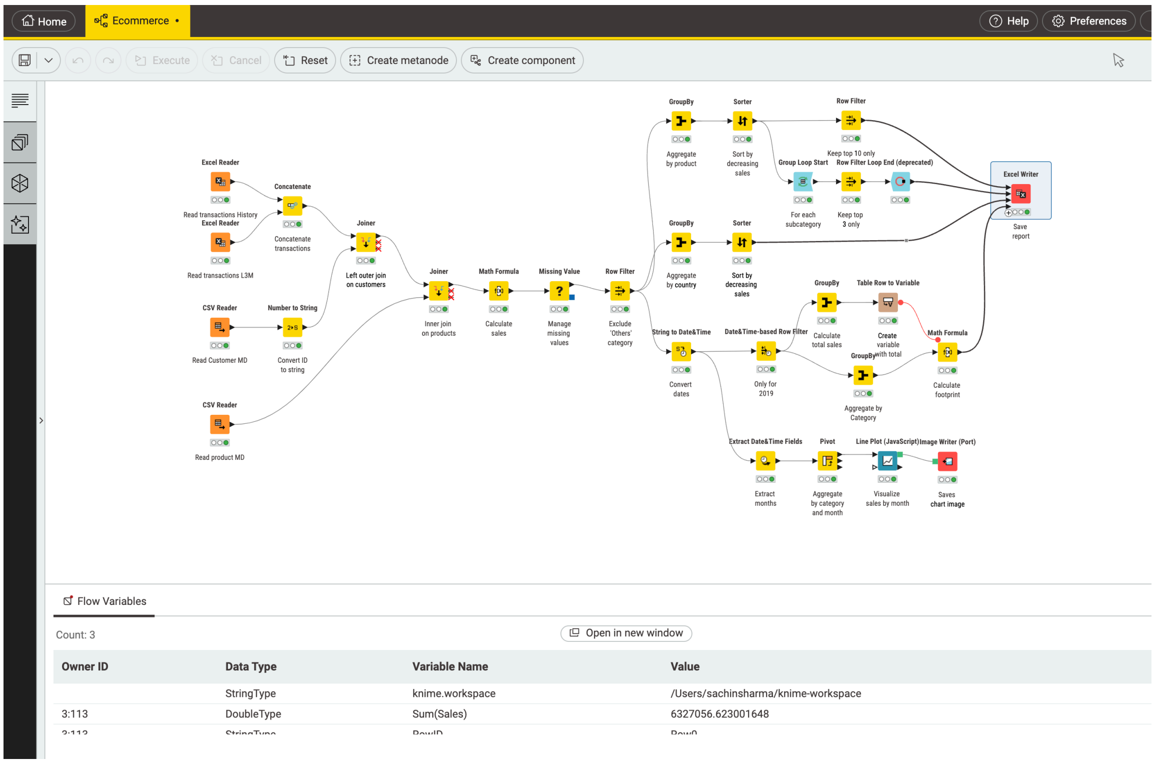
Assignment 4: KNIME Ecommerce workflow

1. Start up the KNIME tool on your personal computer.

   Downloaded the Ecommerce workflow from KNIME Hub using the below link and imported the workflow in KNIME:
   https://hub.knime.com/adm/spaces/Public/Workflows/Data%20Analytics%20Made%20Easy/Chapter%203/Ecommerce~EubUUKYQrZ4z-hTk/current-state

2. Execute and inspect the Ecommerce workflow on your personal computer.

   We have executed each node of Ecommerce Data workflow and inspected the input and output. Attaching the below screenshot of workflow.



Count: 3

Owner ID	Data Type	Variable Name	Value
	StringType	knime.workspace	/Users/sachinsharma/knime-workspace
3:113	DoubleType	Sum(Sales)	6327056.623001648
3:113	StringType	RowID	Row0

The workflow begins by loading two Excel files and two CSV files. The Excel files contain transaction data with fields such as Invoice No, Stock Code, Quantity, Price, Customer ID, and Invoice time. One CSV file holds Customer ID and Country data, while the other includes Stock Code, Description, Product Category, and subcategory details.

Following the data loading phase, the workflow merges the datasets based on common column values and applies various pre-processing techniques. These techniques involve Joining, Filling Missing Values, filtering out irrelevant rows and columns, conducting group by operations, employing sorting methods, and adjusting data types.

After completing these pre-processing steps, the workflow generates an Excel file comprising multiple reports. These reports encompass details like Top Products, Top 3 Products for each subcategory, leading countries by sales, and total 2019 sales for each category. This final Excel document serves as a comprehensive overview of the analyzed data.

3. What are the data files used in this workflow? Describe the content of the datasets provided in these files.

These are the below data files used in the workflow. Providing the file name, file web url and File contents.

1. **eCommerce-TransactionsHistory.xlsx**
   - **URL**: knime://My-KNIME-Hub/Users/adm/Public/Workflows/Data%20Analytics%20Made%20Easy/Chapter%203/eCommerce-TransactionsHistory.xlsx
   - **Contents**:
     - **Invoice**:
       - Description: Represents the Invoice number associated for every transaction.
       - Data Type: String
       - Observation: This column meticulously captures unique Invoice numbers, each intricately linked with specific product and customer identifiers. Importantly, there are no instances of missing values, ensuring the integrity of the dataset.
     - **StockCode:**
       - Description: Represents the Unique number assigned to each product.
       - Data Type: String
       - Observation: The StockCode column serves as a unique identifier for each product. It plays a crucial role in linking to product details such as description, category, and sub-category. Notably, this column exhibits completeness with no missing values.

- **Quantity**:
  - Description: Represents the total quantity of each purchased product.
  - Data Type: Number
  - Observation: The Quantity column accurately records the quantity of products purchased. Although free from missing values, a few negative values are present. It's worth noting that the handling of these negative values is a task yet to be addressed in the workflow.
- **Price**:
  - Description: Represents the price assigned to each product.
  - Data Type: Number (Double)
  - Observation: The Price column reflects the monetary value of products, showing no signs of missing values. However, a minimum value of 0 raises concerns, prompting the suggestion to exclude rows with zero prices. This exclusion ensures precision in subsequent calculations involving the Sales column.
- **Customer_ID**:
  - Description: Represents the Unique Identifier for each individual.
  - Data Type: String
  - Observation: The Customer_ID column contains unique integer values (stored as strings) representing individual users. This particular column plays a pivotal role in amalgamating results from both Excel transaction files and CSV files, with no instances of missing values.
- **Invoice_time**:
  - Description: Represents the Timestamp of each Invoice.
  - Data Type: String

        - Observation: The Invoice_time column meticulously logs timestamp values, encapsulating both date and time information. This temporal data opens avenues for detailed analysis based on year, month, and date. Notably, there are no missing values in this column.

2. **eCommerce-TransactionsL3M.xlsx**
    - **URL**: [eCommerce-TransactionsL3M.xlsx](knime://My-KNIME-Hub/Users/adm/Public/Workflows/Data%20Analytics%20Made%20Easy/Chapter%203/eCommerce-TransactionsL3M.xlsx)
    - **Contents**: Similar to eCommerce-TransactionsHistory.xlsx.
3. **eCommerce-CustomerMD.csv**
    - **URL**: [eCommerce-CustomerMD.csv](knime://My-KNIME-Hub/Users/adm/Public/Workflows/Data%20Analytics%20Made%20Easy/Chapter%203/eCommerce-CustomerMD.csv)
    - **Contents**:
        - **Customer_ID**:
            - Description: Represents the Unique Identifier for each individual.
            - Data Type: String (converted from integer for consistency).
            - Observation: The Customer_ID column holds unique integer values, crucial for identifying and representing individual users. This column assumes significance as it forms the basis for combining results from various files. The decision to convert the Customer_ID data type to a string aligns with maintaining consistency across datasets, ensuring smooth integration.
        - **Country**:

        - Description: Represents the Country name of each individual customer.
        - Data Type: String
        - Observation: The Country column unveils the geographic information associated with each individual customer. This information, devoid of any missing values, will play a pivotal role in identifying and categorizing sales based on the Country column.

4. **eCommerce-ProductMD.csv**
    - **URL**: [eCommerce-ProductMD.csv](knime://My-KNIME-Hub/Users/adm/Public/Workflows/Data%20Analytics%20Made%20Easy/Chapter%203/eCommerce-ProductMD.csv)
    - **Contents**:
        - **StockCode**:
            - Description: Represents the Unique number assigned to each product.
            - Data Type: String
            - Observation: The StockCode column assign a unique id for each product which can be used to link with other datasets. Notably, it remains free from missing values.
        - **Description**:
            - Description: Represents the description of each product.
            - Data Type: String
            - Observation: The Description column holds textual information describing each product. Notably, it remains free from missing values and consists solely of non-empty strings.
        - **Category**:

    - Description: Represents the category of each product.
    - Data Type: String
    - Observation: The Category column categorizes each product, providing essential information for analysing overall sales with a categorical filter. With approximately eight unique product categories, this column exhibits diversity.
  - **Subcategory**:
    - Description: Represents the sub-category of each product.
    - Data Type: String
    - Observation: The Subcategory column further refines the categorization of each product, offering granularity for sales analysis with a sub-category filter. Boasting around 27 unique sub-product categories and devoid of missing values, this column is instrumental in providing detailed insights into product classifications.

4. Document what every node in the workflow does.

These are below nodes which are used in Ecommerce Data workflow. Let us discuss in detail what each node is doing.

Excel Reader: This node specializes in parsing Excel files of various formats, including xlsx, xlsm, xlsb, and xls. It seamlessly handles both single and multiple file inputs, processing one sheet per file. In this scenario, it's utilized to extract data from "eCommerce-TransactionsHistory.xlsx" and “eCommerce-TransactionsL3M.xlsx” files hosted on KNIME URL. The output of this node serves as vital input data for subsequent workflow operations.

CSV Reader: Tailored for CSV files, this node simplifies data extraction. With an option to auto-guess file structure, it effortlessly reads 2 different CSV files from KNIME URL. Noteworthy features include data preview and automatic detection of column data types based on row values. The resultant data becomes readily available for further processing steps.

Concatenate: This node merges tables by combining two input tables. Specifically, it's employed to merge data from "eCommerce-TransactionsHistory.xlsx" and “eCommerce-TransactionsL3M.xlsx” files. Columns sharing identical names are concatenated, utilizing a Union operation while managing duplicate values with a "_dup" suffix. The output is a consolidated table containing rows from both sources.

Number to String: This node converts numerical values within specified columns into string representations. While it offers basic functionality, for more complex transformations like rounding or scientific notation, users may opt for the "Round Double" node. In our workflow, we utilize this node to convert Customer_ID column values to strings for seamless integration with Excel sheet data.

Joiner: Functioning akin to a database join, this node merges two tables based on specified column values, akin to SQL joins. Rows from the top input port are matched with corresponding rows from the bottom port. Unmatched rows can also be included in the output. We employ this node to merge datasets from both Excel and CSV files based on the Customer_ID column, using a left outer join to combine results and discard unmatched rows. Subsequently, we use the same Joiner node to merge the overall dataset of three files with an additional CSV file based on the StockCode column, extracting product information such as description, category, and sub-category through an Inner Join operation.

Math Formula: This node evaluates mathematical expressions using row values from a table, generating calculated results that can be appended as new columns or used to replace existing ones. In our workflow, we utilize this node to calculate sales by multiplying Quantity by Price, creating a new column with the results for each record. Additionally, we employ this node in later stages of the workflow to compute a "Footprint" column using values from the Sales column.

Missing Value: This node handles missing data within the input table by providing options for default handling applicable to all columns, as well as customized settings for individual columns. We utilize this node to eliminate rows containing "unspecified" values in the country column.

Row Filter: This node enables precise row filtering based on defined criteria such as row numbers, RowIDs, or values within specific columns. We employ this node to filter out rows where the category column contains "Others" values, ensuring that only relevant data is retained for analysis. We are using this Row Filter node in later part of workflow.

GroupBy: This node arranges table rows based on unique values within specified group columns. Each distinct set of values in the chosen group column generates a new row, with remaining columns undergoing aggregation according to defined settings. Consequently, the resulting output table comprises a single row for each unique combination of values from the selected group columns. In our workflow, we utilize the GroupBy node for the following operations:

- Grouping rows based on columns (StockCode, Description, Category, Subcategory), with aggregation performed on Sales and Quantity.
- Grouping rows based on the Country column, with aggregation applied to Sales and Quantity.

- Conducting GroupBy operations in a later stage of the workflow. Initially, we convert the Invoice_time column from string to DateTime format and filter rows where the year is greater than or equal to 2019. Subsequently, we perform GroupBy operations to calculate total sales and group rows based on the Category column.

String to Data&Time: This node facilitates the conversion of string values within specified columns into Date & Time format. Users have the flexibility to choose from a range of commonly used formats or define a custom format tailored to their data's structure. In our workflow, we employ this node to convert the timestamp values stored as strings in the Invoice_time column back to their original format, utilizing the specified format ('D'dd/M/yy'T'HH:mm:ss).

Sorter: This node orchestrates the arrangement of rows based on user-defined criteria. Within the configuration dialog, users select the columns by which the data should be sorted and specify whether the sorting should be in ascending or descending order. We utilize this node in two instances within our workflow to sort the Sales column values in descending order, ensuring optimal organization of the data.

Date&Time-based Row Filter: This node extracts all rows where the time value of the designated column falls within a specified time window from the input data. Users define the time window by setting a start date (and time) along with either an end date (and time), a duration, or a numerical value paired with a granularity. We are using this node to fitter Invoice_time column and only selecting rows where start year is 2019 or greater than that.

Extract Date&Time Fields: This node facilitates the extraction of specific fields from a column containing Local Date, Local Time, Local Date Time, or Zoned Date Time data. It adds the extracted values as corresponding integer or string columns. In our workflow, we utilize this node to extract the Month from the Invoice_time datetime column, creating a new column to store the extracted month values.

Group Loop Start: In a Group Loop Start, each iteration processes a distinct group of rows based on specified columns. Before the looping commences, the input data table is sorted based on the specified columns. In our workflow, we employ the Subcategory column to define the groups for this loop.

Loop End (deprecated): This node marks the conclusion of a workflow loop, collecting intermediate results by concatenating incoming tables row-wise. The loop's initiation is defined by the loop start node, where users specify the loop's execution frequency (either fixed or derived from data, such as with the "Group Loop Start" node). In our workflow, we execute the Group Loop Start node for each Subcategory, filter only the top 3 values, and repeat the process for each Subcategory. This Loop End node concludes this iterative process, providing the top 3 results for each Subcategory.

Table Row to Variable: This node utilizes the values in the first row of a data table to create new flow variables. The names of these variables correspond to the column names, while their assignments are determined by the values found in the respective row. In our workflow, we use this node to extract the Sum(Sales) obtained from the GroupBy node and handle Missing Sales values, incorporating them as flow variables for further processing.

Pivot: This node performs a pivot operation on the input table, utilizing specified columns for grouping and pivoting. Group columns generate unique rows, while pivot values are transformed into columns for each combination of columns, accompanied by designated aggregation methods. In our workflow, we pivot the results based on the Month column, creating rows for each month and reevaluating all values according to the pivot column.

Line Plot (JavaScript): This node generates a line plot using a JavaScript-based charting library. Users can access the visualization either through the Interactive View action on the executed node or via the KNIME Web Portal. We configure the chart with the Month column along with selected columns such as Christmas, Easter, and Garden values for visualization.

Image Writer (Port): This node writes an image port object to a specified file or remote location, typically specified by a URL. It takes the output of the Line Plot node, which generates a chart graph, and saves it using this node. The generated chart graph can be saved to a local directory or a URL for further use or sharing.

Excel Writer: This node exports the input data table into a spreadsheet within an Excel file. In our workflow, we utilize this node to consolidate all results and store them in an Excel file with separate sheets. Specifically, we create four Excel sheets for Top Products, Top Products by Subcategory, Top Countries, and 2019 Sales. Each sheet is computed based on product data and sales information, providing a comprehensive summary in the Excel file.

5. Create the seasonality line chart and save it in a file.

Seasonality graph has been created by using Line Plot (JavaScript) node and saved as SVG by using Image Writer (Port) node.

Attaching below screenshot:

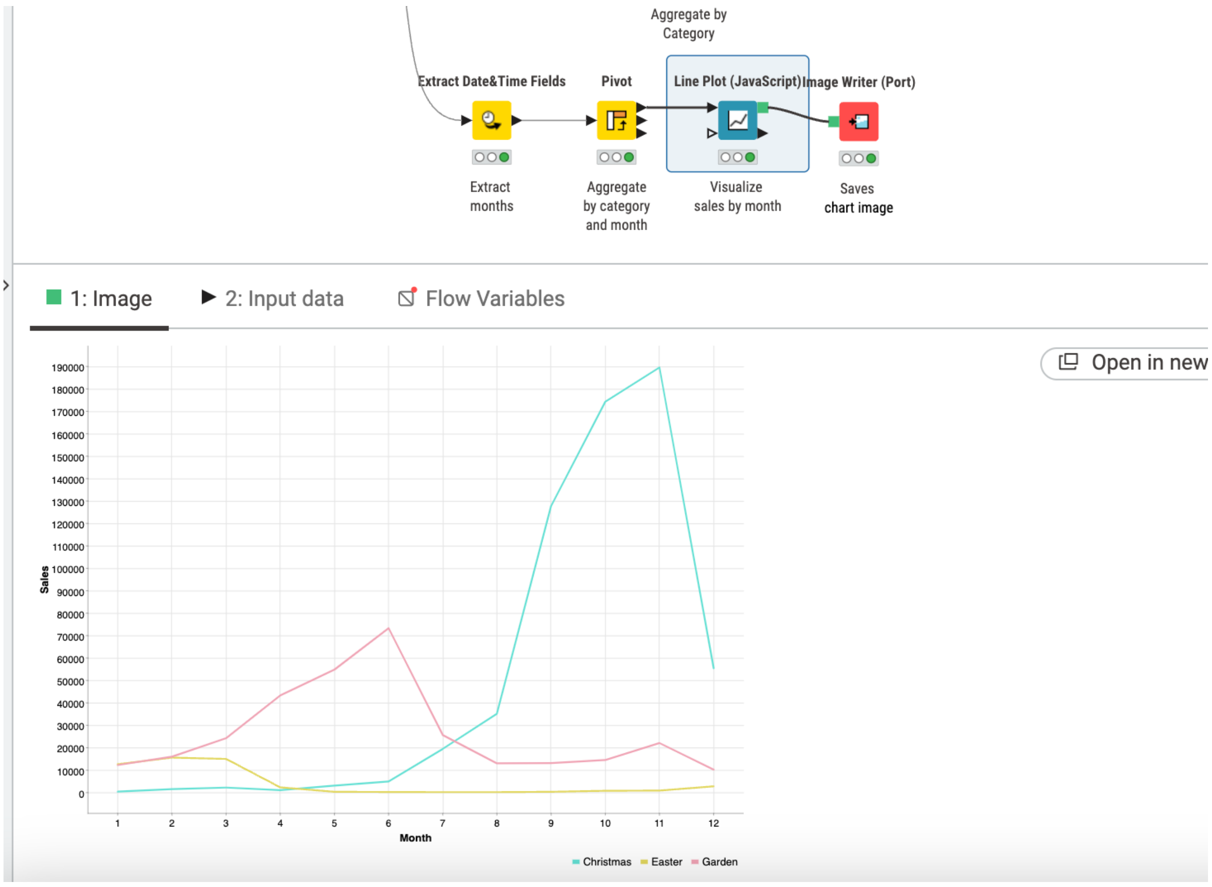
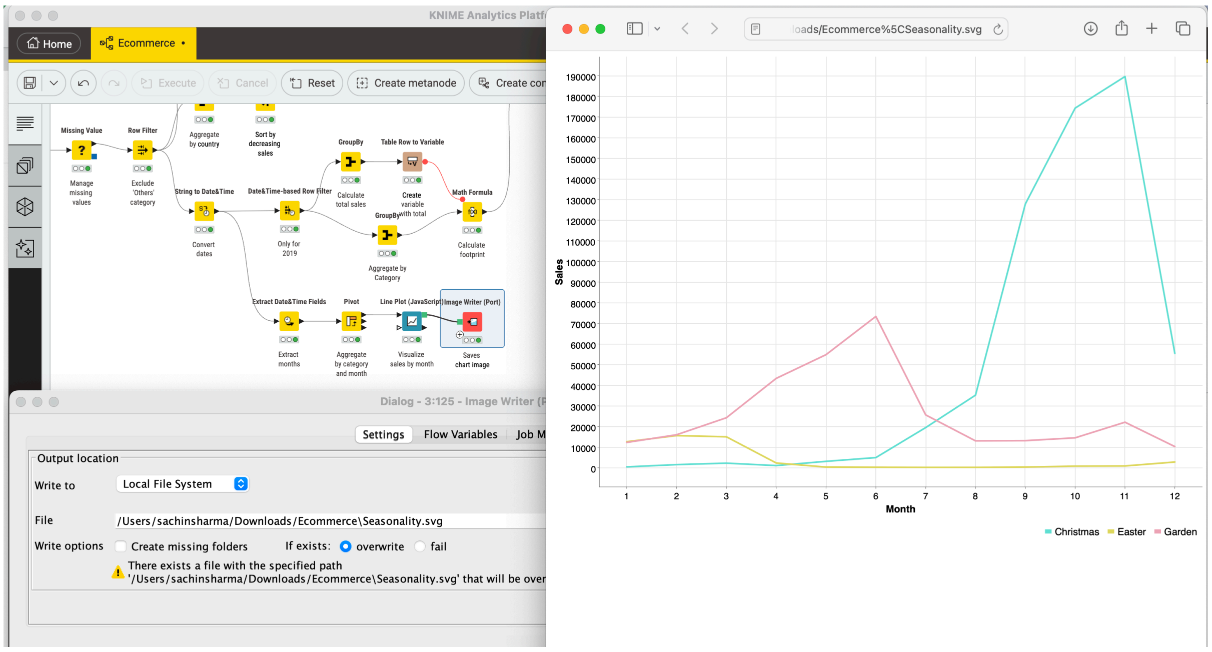


Fig 2 Seasonality Graph creation



Output location

Write to: Local File System

File: /Users/sachinsharma/Downloads/Ecommerce\Seasonality.svg

Write options: Create missing folders   If exists: [✓ overwrite] fail

There exists a file with the specified path '/Users/sachinsharma/Downloads/Ecommerce\Seasonality.svg' that will be over

Fig 3 File saved in local machine

6. Describe how business process analytics, as illustrated in this assignment, can be useful to organizations.

Business Process Analytics, as demonstrated in the KNIME Commerce workflow, can offer various benefits to organizations by providing valuable insights into their e-commerce operations. Here are several ways in which Business Process Analytics can be useful:

1. **Performance Monitoring:**

   Analyzing transactional data and key performance indicators (KPIs) allows organizations to monitor the performance of their e-commerce processes. This includes tracking sales trends, identifying popular products, and assessing overall business performance.

2. **Customer Segmentation:**

   By utilizing customer data, organizations can employ analytics to segment their customer base. This segmentation can help in understanding different customer behaviors, preferences, and needs. It enables targeted marketing strategies and personalized customer experiences.

3. **Sales Forecasting:**

   Time-series analysis and forecasting techniques applied to historical sales data can assist organizations in predicting future sales trends. Accurate sales forecasts are crucial for inventory management, ensuring that products are available to meet demand.

4. **Product Recommendations:**

Recommender systems, powered by analytics, can provide personalized product recommendations to customers based on their historical preferences and behaviors. This enhances the user experience and encourages additional purchases.

5. **Fraud Detection:**

   Business Process Analytics can help identify unusual patterns or anomalies in transaction data, aiding in the detection of fraudulent activities. Early detection of fraud is essential for maintaining the integrity of e-commerce operations.

6. **Operational Efficiency:**

   Analyzing the workflow and processing steps in the e-commerce system allows organizations to identify bottlenecks, inefficiencies, or areas that can be streamlined. This leads to improved operational efficiency and resource optimization.

7. **Customer Retention Strategies:**

   By analyzing customer behavior and purchase history, organizations can develop effective customer retention strategies. Understanding why customers may churn and implementing targeted retention efforts can be instrumental in maintaining a loyal customer base.

8. **Marketing Effectiveness:**

   Business Process Analytics helps assess the effectiveness of marketing campaigns. Organizations can analyze customer responses to different channels, promotions, or advertisements, enabling them to optimize marketing strategies for better returns on investment.

9. **Supply Chain Optimization:**

   For e-commerce businesses managing physical products, analytics can optimize the supply chain by providing insights into inventory levels, order fulfillment times, and supplier performance. This ensures a streamlined and cost-effective supply chain.

10. Strategic Decision-Making:

Business leaders can make informed, data-driven decisions by leveraging insights gained from analytics. This includes decisions related to product offerings, pricing strategies, market expansion, and overall business strategy.

In summary, Business Process Analytics in the context of the KNIME Commerce workflow empowers organizations to understand, evaluate, and optimize various aspects of their e-commerce operations. This data-driven approach enhances decision-making, improves customer experiences, and ultimately contributes to the overall success and growth of the organization.