**SRH Hochschule Heidelberg**

**Faculty of Media and Design**

**Mexican Toy Store**

**Data Management 2**

Paolo Vega Prof. Dr. Ajinka Prabhune

Nov 9, 2021 Ashish Chouhan

**Context**

However, in our last **customer satisfaction** score, we saw that it **dropped from 25% to 22%** in the last three months. Therefore, an immediate action must be taken to ensure that our customers hare happy with our products.

Moreover, in our last **customer feedback** results, an anonymous user last week stated that he prefers the competition because whenever he visits our website, he is **not able to find** any other product **recommendation** in his profile and need to manually search for alternative products. He claimed that it is a time-consuming task and for that reason he prefers the competition.

**Our SMART Customer Questions are:**

1. As we want to increase our customer satisfaction among our current customers, **how can we segment our clients for our reward program?**
2. As we of products is growing, **how can we make sure that our customers get relevant recommendations online?**

**The following objectives below are the points we consider**

1. **Increase** our Customer Satisfaction score **by 15%** in the next Q1 for the next year compared to our current score (22%).
2. **Increase** our number of online sales **by 20%** in the next 6 months compared to our current sales margin.

**How can we achieve these objectives?**

The picture shows the steps that are needed to answer our business questions at hand. A brief description of each stage is described below.

Graphical user interface, application, Word

Description automatically generated

Steps:

1. **Data Sources**. We already have 6 datasets that are normally used to generate the weekly reports. Below is a brief description for each one as they contain valuable information for our business questions.

|  |  |  |
| --- | --- | --- |
| **Dataset** | **No Rows** | **Description** |
| Customers | 1,683 | Information about the customers including demographic information such as name, age, gender, marital status, type of customer and their corresponding spending score. |
| Ratings | 53,099 | Product ratings given by users at a certain point in time |
| Inventory | 1,593 | Information about the available products in physical stores |
| Sales | 53,099 | Transactional information about the purchases of products that have occurred during the first two months of 2020. |
| Stores | 50 | Information about the stores in Mexico |
| Products | 35 | Information about the product such as category, cost and price for the current products available in the company |

1. **STG**. This is the first stage in our data pipeline as we prefer not working with CSV files; instead, we need to upload our information into our DB server where we can perform cleaning and transformation processes to our data. This steps just adds everything into the DB for profiling analysis.

Text

Description automatically generated

1. **Data Profiling**. This is when we analyze the information we have in each table and detect those data quality dimensions that are required for each column to have a deeper understanding of the rules we need to apply in our ETL process. All reports are generated during the ETL execution and can be found under **logs/profiling\_reports/** folder. A brief summary of the results is shown below:

|  |  |
| --- | --- |
| **Dataset** | **Data Quality Dimensions** |
| Product | Completeness Accuracy Uniqueness Consistency Validity Conformity |
| Customer | Completeness Accuracy Uniqueness Consistency Validity Conformity |
| Stores | Completeness Accuracy Uniqueness Consistency Validity Conformity |
| Inventory | Completeness Accuracy Uniqueness Consistency Validity Conformity |
| Sales | Completeness Accuracy Uniqueness Consistency Validity Conformity |
| Ratings | Completeness Accuracy Uniqueness Consistency Validity Conformity |

1. **ETL** (Data Cleaning, Transformation and Loading). After analyzing which data quality dimensions are meet by each table, it’s time to apply such rules to all existing rows.

In the table below, we can see a detailed description about the data quality that are considered in each column as well as the respective data cleaning rule to ensure such criteria. The last column indicates the document in which the Data Profiling can be found under logs/profiling\_reports/

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Table** | **Column** | **Data type** | **Data Dimensions** | **Business Rules/ Data Cleaning Rules** | **Remarks** | **Data Cleaning Rules** | **Data Profiling Report** |
| Product | Product\_Id | Int | - Uniqueness - Validity | Unique Product Identifier | Functional Dependency with other columns | data type conversion | stg\_products\_2021-11-08 |
| Product\_Name | Char | -Consistency - Validity | Descriptions should capitalize the first letters |  | title case |
| Product\_Category | Char | -Consistency - Validity | Descriptions should capitalize the first letters |  | title case |
| Product\_Cost | Float | - Validity - Conformity - Precision | Should always be positive. Should follow the ##.## pattern |  | numbers replace $ -> '' float conversion |
| Product\_Price | Float | - Validity - Conformity - Precision | Should always be positive. Should follow the ##.## pattern |  | numbers and $ replace $ -> '' float conversion |
| Delivered\_Data | Date | NA | Remove empty column |  |  |
| Customer | Customer\_ID | Int | - Uniqueness - Validity | Should only contain numbers | Functional Dependency with other columns |  | stg\_customer\_2021-11-08.html |
| Customer\_Name | Char | - Consistency - Validity | Descriptions should capitalize the first letters |  | title case |
| Type\_Customer | Char | - Consistency - Validity | Descriptions should capitalize the first letters |  | title case |
| Sex | Char | - Consistency - Validity | Descriptions should capitalize the first letters |  | title case |
| Marital\_Status | Char | - Consistency - Validity | Descriptions should capitalize the first letters |  | title case |
| Age | Int | - Validity - Conformity - Precision | Should always be positive. Should follow the ##.## pattern |  | numbers int conversion |
| Spending\_Score | Int | - Validity - Conformity - Precision | Should always be positive. Should follow in the range between 1 and 100 Should follow the ##.## pattern |  | numbers int conversion |
| Stores | Store\_ID | Int | - Uniqueness - Validity | Should only contain numbers | Functional Dependency with other columns | int conversion | stg\_stores\_2021-11-08 |
| Store\_Name | Char | - Consistency - Validity | Descriptions should capitalize the first letters |  | title case |
| Store\_City | Char | - Consistency - Validity | Descriptions should capitalize the first letters |  | title case |
| Store\_Location | Char | - Consistency - Validity | Descriptions should capitalize the first letters |  | title case |
| Store\_Open\_Date | Date | - Conformity - Validity | Dates should be in the mm/dd/yyyy format |  | date format dd/mm/yyyy |
| Store\_Country | Char | - Consistency - Validity | Descriptions should capitalize the first letters |  | title case |
| Inventory | Store\_ID | Char | - Consistency | Descriptions should capitalize the first letters | Functional Dependency with other columns | title case | stg\_inventory\_2021-11-08 |
| Product\_ID | Char | - Consistency | Descriptions should capitalize the first letters | title case |
| Stock\_On\_Hand | Int | - Validity - Precision | Cost should always be a positive number |  | int conversion |
| Sales | Sales\_ID | Int | - Uniqueness - Validity |  | Functional Dependency with other columns |  | stg\_sales\_2021-11-08 |
| Date | Date | - Validity - Consistency - Conformity | Dates should be in the mm/dd/yyyy format |  | date format dd/mm/yyyy |
| Store | Char | - Consistency - Validity | Descriptions should capitalize the first letters |  | title case |
| Product | Char | - Consistency - Validity | Descriptions should capitalize the first letters |  | title case |
| Status | Char | NA | Remove empty column |  | title case |
| Delivered\_Date | Date | - Validity - Consistency - Conformity | Dates should be in the mm/dd/yyyy format |  | date format dd/mm/yyyy |
| Purchase | Char |  | Descriptions should capitalize the first letters |  | title case |
| Customer | Char | - Consistency - Validity | Descriptions should capitalize the first letters |  | title case |
| Units | Int | - Validity - Precision | Should always be positive. |  | int conversion |
| Discount | Int | - Validity - Precision | Should always be positive. |  | float conversion Remove "%" |
| Ratings | Date | Date | - Validity - Consistency - Conformity | Dates should be in the mm/dd/yyyy format |  | date format dd/mm/yyyy | stg\_ratings\_2021-11-08 |
| Product\_ID | Char | - Uniqueness - Consistency - Validity | Descriptions should capitalize the first letters |  | title case |
| Customer\_ID | Char | - Uniqueness - Consistency - Validity | Descriptions should capitalize the first letters |  | title case |
| Rating | int | - Consistency - Validity | Should always be positive and the range is between 1 and 5 |  | int conversion |

**Schema**. After profiling the data, we then need to define a proper schema to store our information. A Snowflake Schema seems to be the more suitable model for our needs as avoiding redundancy is one of our primary goals. The ER Diagram is below, where the tables in the middle (inventory, ratings, and sales) are our main facts and the rest the catalogues for each dimension.

Graphical user interface, application

Description automatically generated

**Transformation**. Now that we have defined the tables in which we need to store the information, we need to run the ETL process to start analyzing our data. The following diagram shows the flow in which the ETL runs. First, it extracts the data from the CSV files and store it in the STG destination. Then, it runs the profiling report, so we know which data quality dimensions we need to apply in each stage. Once we have the definition for such rules, we start populating catalogues (our dimensions). For example, as we can see below, we fill information about Customers, Stores and Products and we populate our facts until every other table has been populated to ensure integrity in our data.

Diagram

Description automatically generated

**Data Analysis Results**

With the data ready from our previous step, we are now ready to start analyzing our data and try to understand first the current information we have after all transformations in our warehouse. After that, we would be able to go ahead and answer the main questions that our different departments are interested in knowing for the current objectives in mind.

Chart

Description automatically generated

Now, let’s see how we can help them to continue with their strategy. Both models are located under the models/ folder.

1. **How can we segment our clients for our reward program?**

This question requires us to understand more about our customers. The “customers” table seems to be the one we can use for this problem as we have acquired information for all our current customers such as age, marital status, gender, which customer type they are and a spending score that has been assigned based on previous behavior.

As we have not defined the number of groups we need, we then could use an unsupervised algorithm where we can cluster our customer based on certain criteria and let the algorithm decide for us how the groups need to be determined by looking at the “Elbow Method”. Our suggestion is based on a KNN algorithm, where **Age and Spending Score** where our two prime features to consider the groups, given that we have a diverse group of customers purchasing in both physical stores and online via our website. The final number of clusters is 3 as it can be observed in our image below.

Chart, line chart

Description automatically generated

The final customer segmentation is shown below, where can clearly see **3 different groups** of customers given their corresponding age and spending score. The marketing team can now use part of their reward program to target people less than 50 years old (on average) with an active purchasing score. In addition, they have the possibility to offer an alternative discount or redeem coupons to the other two groups if necessary.

Chart, scatter chart

Description automatically generated

The last part of this model is to enable an API to consume our model where we can predict the result for new customer. An example of this request is show below, where we just simulate **how the classification would be for a customer 34 years old and with a spending score of 80**.

Graphical user interface, text, application, email

Description automatically generated

The code of this model can be found under models/customer\_segmentation.ipnb

1. **How can we make sure that our customers get recommendations of our products?**

This question requires us to recommend products to our users, especially those using our website as it may lead us to increase our revenue. For this point, the “ratings” table contains what we need because here we know which product has been rated by the user. Therefore, we can use a collaborative filtering algorithm to have possible recommendations for those products that have not been seen by the user.

In this case, we need to use an item-based collaborative filter that allows to show at least 5 recommendations to our users whenever then navigate to our main website as it is an important feature to be competitive in the market. Our suggestion is based on a Singular Value Decomposition algorithm (SVD), as it is a more scalable algorithm as our products and customers will increase eventually and we need to use an optimal algorithm to support this scenario. Therefore, we just need to predict a possible score (from range 1 to 5) for those products that have not been purchased by the user. An example of this result is show in the image below.

A picture containing text, monitor

Description automatically generated

The last part of this model is to enable an API for our developers to consume our model from our website and shows recommendations for every user logged in. For example, in the image below we can see those products that have been rated by the user and below that, we show 5 more products that have not been rated by our customer with id **1684**, which in fact are our recommendations.

Graphical user interface, table

Description automatically generated

Graphical user interface, text, email

Description automatically generated

The code of this model can be found under models/item\_recommendation.ipnb