INTERNATIONAL SCHOOL
VIETNAM NATIONAL UNIVERSITY, HANOI

FINAL REPORT

The IFOOD CASE Data Warehousing with BigQuery: Driving Business Analytics and Marketing Efficiency

DATA WAREHOUSING & BUSINESS ANALYTICS

ID: INS307301
Lecturer: A/Prof. Tran Thi Ngan

GROUP 05
Trần Thanh Hà - 21070067
Nguyễn Thị Hương Giang - 21070248
Vũ Tuyên Hoàng - 21070141
Nguyễn Việt Thành - 21070155

TABLE OF CONTENTS

I. Introduction

1. iFood Dataset Description

The iFood dataset originates from iFood, the leading food delivery service in Brazil, renowned for its extensive reach across over a thousand cities. With a substantial user base and high customer engagement, iFood continually seeks to enhance its market dominance through data-driven insights and innovative marketing strategies.

Publicly released in December 2020, the iFood dataset contains data spanning from 2012 to 2014. It includes 28 columns and 2,240 rows, capturing a wealth of information about customer behaviors and marketing performance. This dataset is part of a broader initiative to understand and improve customer interactions and responses to marketing campaigns. It encompasses various aspects of customer demographics, purchase behaviors, and campaign feedback, which are crucial for developing predictive models and optimizing marketing efforts.

Designed for use with SQL Server, the dataset is also compatible with other database management systems such as Oracle, MySQL, and PostgreSQL. It supports complex functionalities, including data processing with transactions, combining data from various tables, and using data constraints to maintain data integrity. Furthermore, the dataset is ideal for constructing a data warehouse, enabling the integration and centralization of data from multiple sources. Analysts can utilize this dataset for data visualization, training prediction and classification models, and performing advanced analytics.

2. Problem Overview

Selecting the iFood dataset for our project is strategically advantageous due to its rich and diverse information. This dataset encompasses detailed socio-demographic data, and customer interactions with marketing campaigns, which are essential for our analysis. The clarity and variety of this data enable us to develop sophisticated predictive models that forecast customer behaviors and campaign outcomes. This capability is crucial for designing targeted marketing initiatives that maximize customer engagement and profitability.

Leveraging the comprehensive attributes provided by the iFood dataset, we aim to build a data warehouse robust system for managing and analyzing data. Our goal is to

utilize the results from previous campaign acceptances and the characteristics of customers who responded positively to the first five campaigns. Specifically, we will analyze whether these customers, who accepted the first five campaigns and continued to accept the sixth, are likely to respond positively to future campaigns.

A data warehouse is essential in this context as it allows for the consolidation of large volumes of data from various sources into a centralized repository. This centralized data repository facilitates efficient querying, reporting, and analysis. By implementing a Data Warehouse, we can ensure that our data management system supports the complex analytical tasks required to identify high-potential customer segments and optimize marketing strategies.

By identifying the customers most likely to respond to marketing campaigns, we can prioritize sending targeted marketing materials to these high-potential customers. This targeted approach ensures that marketing efforts are focused on the most receptive audience, thereby improving the efficiency of every marketing dollar spent. Such precision not only enhances the return on investment for marketing campaigns but also strengthens customer relationships by delivering relevant and personalized content.

3. Project objectives

The iFood project addresses inherent challenges in managing and analyzing marketing data to enhance campaign effectiveness and customer engagement. By focusing on identifying high-potential customer segments and optimizing marketing strategies, this project aims to leverage data warehousing and advanced analytics to enhance marketing effectiveness and improve customer engagement.

The project unfolds through a meticulously structured sequence of phases, each contributing to the overarching objective of building a robust system for data management and analysis:

- Gain a Thorough Understanding of the iFood Dataset, focusing on its application within marketing analysis within the project.

- Acquaint Oneself with Essential Tools such as BigQuery, Nifi, Airflow, and Looker Studio, instrumental in constructing the data warehouse and implementing various machine learning algorithms for predictive objectives.

- Develop a list of questions for the project, outlining key queries that need to be addressed and identify metrics that will assist marketing managers in monitoring and fostering growth within marketing campaigns.

- Design a Data Staging Layer: Create a data staging layer in a star schema format based on the identified business questions, ensuring efficient data processing and analysis. This schema will organize data into fact and dimension tables to facilitate easy querying and reporting.

- Perform Data Mining Activities: Engage in data mining activities, including visualization and predictive modeling, utilizing the star schema staging layer. This phase involves exploring the data to uncover patterns and insights that can inform marketing strategies.

- Derive Insights from Analytical and Predictive Models: Generate actionable insights from analytical and predictive models, focusing on formulating practical strategies tailored for future marketing campaigns.

- Tailor Recommendations to Enhance Marketing Performance: Provide targeted recommendations to improve marketing performance, optimize campaign resources, and leverage emerging market trends.

4. Business Question

In line with our project design and focusing on the marketing aspect of the project, we propose the following key research questions:

Marketing Campaign Responses:

- What are the specific factors that contribute to the differences in response rates across various marketing campaigns?

- How can we optimize marketing strategies to enhance customer engagement and increase the acceptance rates of future campaigns?

- What are the underlying causes of the high acceptance rates for certain campaigns, and how can these successful tactics be applied to other campaigns?

Customer Segmentation:

- Which customer segments are most likely to respond positively to future marketing campaigns, based on historical data?

- How do demographic factors such as age, income, education level, and marital status influence customer segmentation and response rates to marketing efforts?

- What distinct groups of customers can be identified based on their purchasing behavior and campaign feedback, and how can marketing strategies be tailored to each segment?

Campaign Performance Analysis:

- How effective have past marketing campaigns been in driving sales and customer engagement?

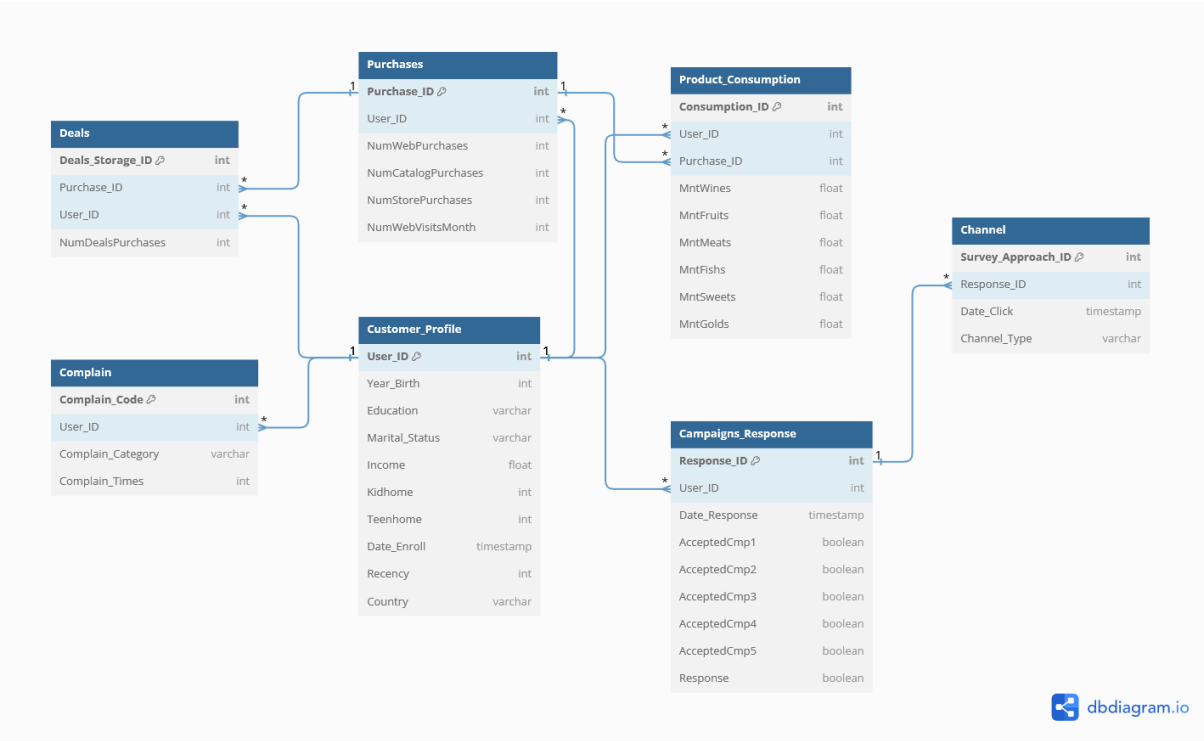
- What are the key performance indicators (KPIs) that measure the success of marketing campaigns, and how do these KPIs vary across different customer segments and product categories?

- How do factors like the frequency of web visits, number of purchases, and total spending correlate with the responsiveness to marketing campaigns?

II. Database Prepare

1. Data Design

The dataset represents a comprehensive schema designed for an online transaction processing (OLTP) system. The provided ERD (Entity Relationship Diagram) illustrates the intricate relationships between key business entities, explaining how customer management, sales, product consumption, and order processing interact dynamically.



The most common one-to-many relationships in the database dominate the ERD. The Customer_Profile table is at the core of the transaction system, linking customer information to their purchases, product consumption, campaign responses, deals, and complaints. Foreign keys and indexes enforce referential integrity and maintain query performance, which are prerequisites for OLTP systems built to support dynamic business operations. This robust design ensures comprehensive data integrity and facilitates complex queries essential for effective customer management, sales tracking, and order processing.

2. ETL Design and Development

2.1. Data Preparation and Transformation

To build our data warehouse, we began by extracting data from a flatten dataset. This involved adding ID columns for different departments (Campaign Response, Deals, Complains, Product Consumption, Purchases, and Survey Approach) to improve data organization and traceability.

```
Data columns (total 28 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   User_ID              2240 non-null   int64
 1   Year_Birth           2240 non-null   int64
 2   Education            2240 non-null   object
 3   Marital_Status       2240 non-null   object
 4   Income               2216 non-null   float64
 5   Kidhome              2240 non-null   int64
 6   Teenhome             2240 non-null   int64
 7   Date_Enroll          2240 non-null   object
 8   Recency              2240 non-null   int64
 9   MntWines             2240 non-null   float64
 10  MntFruits            2240 non-null   float64
 11  MntMeats             2240 non-null   float64
 12  MntFishs             2240 non-null   float64
 13  MntSweets            2240 non-null   float64
 14  MntGolds             2240 non-null   float64
 15  NumDealsPurchases    2240 non-null   int64
 16  NumWebPurchases      2240 non-null   int64
 17  NumCatalogPurchases  2240 non-null   int64
 18  NumStorePurchases    2240 non-null   int64
 19  NumWebVisitsMonth    2240 non-null   int64
 20  AcceptedCmp3         2240 non-null   int64
 21  AcceptedCmp4         2240 non-null   int64
 22  AcceptedCmp5         2240 non-null   int64
 23  AcceptedCmp1         2240 non-null   int64
 24  AcceptedCmp2         2240 non-null   int64
 25  Response             2240 non-null   int64
 26  Complain             2240 non-null   int64
 27  Country              2240 non-null   object
dtypes: float64(7), int64(17), object(4)
```

```
Data columns (total 38 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   User_ID              2240 non-null   int64
 1   Year_Birth           2240 non-null   int64
 2   Education            2240 non-null   object
 3   Marital_Status       2240 non-null   object
 4   Income               2216 non-null   float64
 5   Kidhome              2240 non-null   int64
 6   Teenhome             2240 non-null   int64
 7   Date_Enroll          2240 non-null   object
 8   Recency              2240 non-null   int64
 9   Country              2240 non-null   object
 10  Complain_Code        2240 non-null   object
 11  Complain_Category    2240 non-null   object
 12  Complain_Times       2240 non-null   int64
 13  Purchase_ID          2240 non-null   int64
 14  NumWebPurchases      2240 non-null   int64
 15  NumCatalogPurchases  2240 non-null   int64
 16  NumStorePurchases    2240 non-null   int64
 17  NumWebVisitsMonth    2240 non-null   int64
 18  Consumption_ID       2240 non-null   int64
 19  MntWines             2240 non-null   float64
 20  MntFruits            2240 non-null   float64
 21  MntMeats             2240 non-null   float64
 22  MntFishs             2240 non-null   float64
 23  MntSweets            2240 non-null   float64
 24  MntGolds             2240 non-null   float64
 25  Response_ID          2240 non-null   int64
 26  Date_Response        2240 non-null   object
 27  AcceptedCmp1         2240 non-null   bool
 28  AcceptedCmp2         2240 non-null   bool
 29  AcceptedCmp3         2240 non-null   bool
 30  AcceptedCmp4         2240 non-null   bool
 31  AcceptedCmp5         2240 non-null   bool
 32  Response             2240 non-null   bool
 33  Survey_Approach_ID   2240 non-null   int64
 34  Date_Click           2240 non-null   object
 35  Channel_Type         2240 non-null   object
 36  Deal_Storage_ID      2240 non-null   int64
 37  NumDealsPurchases    2240 non-null   int64
dtypes: bool(6), float64(7), int64(16), object(9)
```

Data Type Transformation:

We converted date values from objects to the Datetime format for accurate time-based analysis. Binary values representing Yes/No decisions (0 and 1) were changed from integers to booleans to enhance data readability and facilitate more intuitive data manipulation.

Renaming and Reordering Columns:

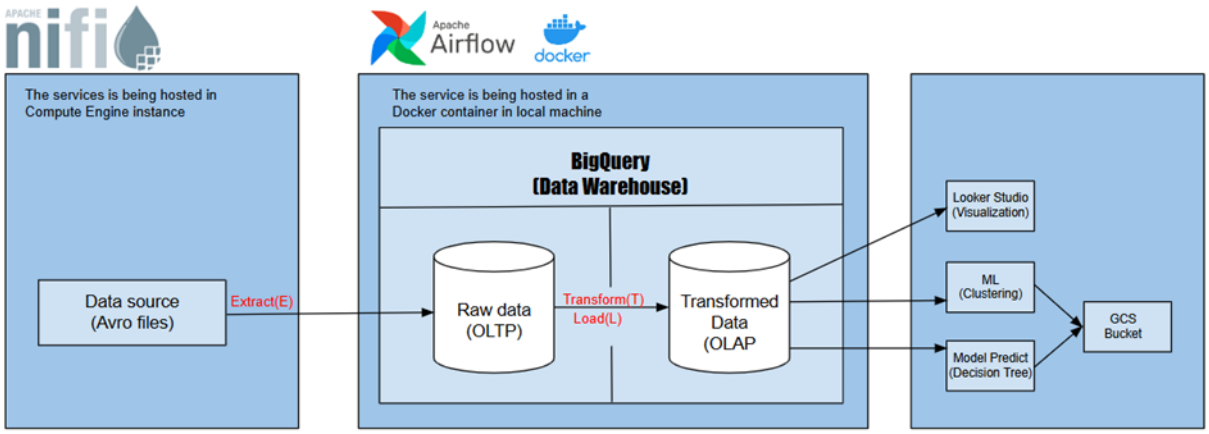
We renamed columns to be more descriptive, which improved the clarity and understanding of the dataset. Additionally, columns were reordered into logical clusters based on business processes, such as campaign responses and product consumption. This arrangement ensured data was logically structured and easily navigable.

Creating OLTP Tables:

The cleaned and transformed dataset was divided into seven tables: Customer, Campaign_Response, Deals, Product_Consumption, Purchases, and Survey_Approach.

Each table was designed to store specific operational data relevant to different business processes, optimizing data management and supporting effective analysis.

2.2. Designing and Implementing the ETL Process



Step 1: Extract Data: Use Apache NiFi to extract data from Avro files. NiFi will read the data from the source and move it to BigQuery in its raw form.

Step 2: Load Raw Data: The extracted data is loaded into BigQuery in a raw data table (OLTP).

Step 3: Transform Data: Use Apache Airflow to orchestrate the data transformation tasks. These tasks might include data cleaning, data aggregation, and format conversion.

Step 4: Load Transformed Data: The transformed data is loaded back into BigQuery in a processed data table (OLAP), ready for analysis and reporting.

Implementation:

- Configure Apache NiFi: Set up data flows to extract and move data from Avro files to BigQuery.

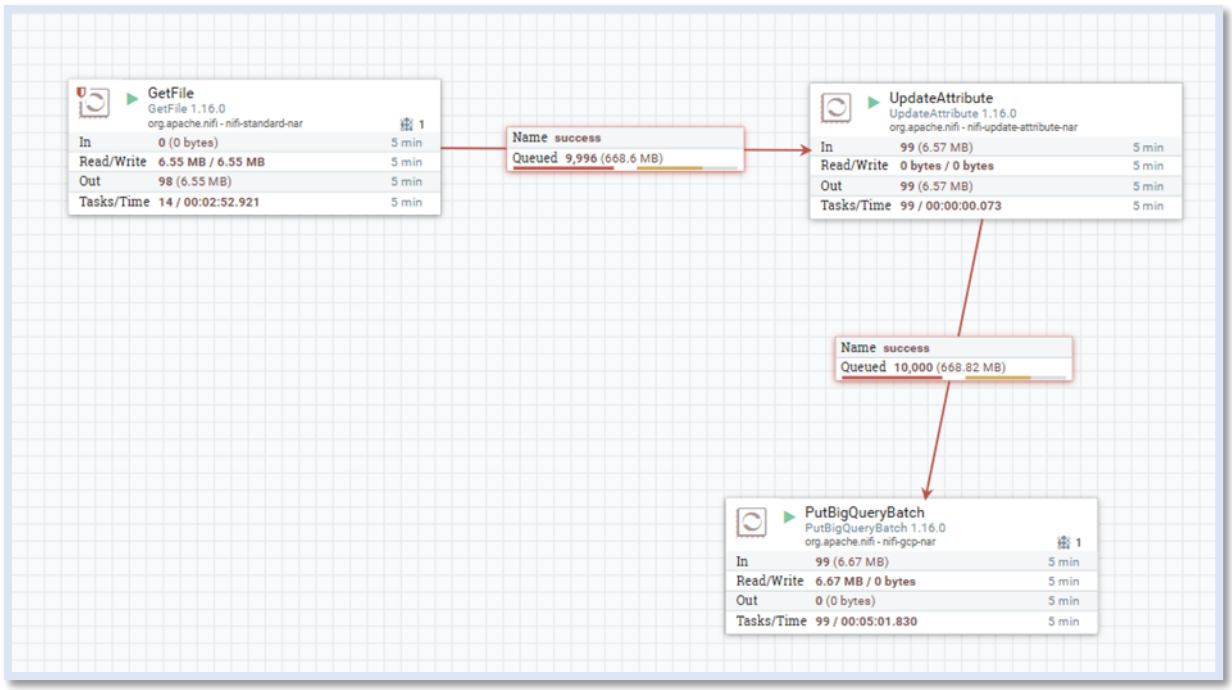
- Set Up BigQuery: Create the necessary tables to store raw and processed data.

- Configure Apache Airflow: Create DAGs (Directed Acyclic Graphs) in Airflow to schedule and manage the data transformation tasks.

- Deploy Docker Container: Ensure Airflow is deployed and running in a Docker container to provide a stable and reusable execution environment.

Detailed ETL Process:

Apache NiFi Flow



1. GetFile: This processor reads files from a specified directory.

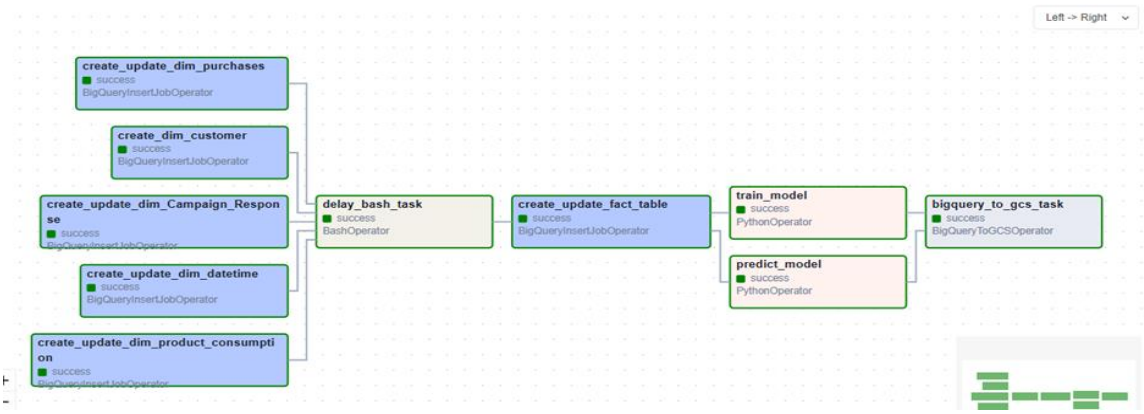
2. UpdateAttribute: This processor updates the attributes of flow files based on user-defined rules.

3. PutBigQueryBatch: This processor writes the flow files to a Google BigQuery table in batch mode.

This NiFi flow shows a the method to move data from a local file system to BigQuery, with a transformation step to update attributes before loading.

Apache Airflow Workflow

The Airflow DAG (Directed Acyclic Graph) provides a clear visualization of the data transformation and processing steps:



create_update_dim_purchases,

create_update_dim_customer,

create_update_dim_Campaign_Response,

create_update_dim_datetime,

create_update_dim_product_consumption:

- Function: These tasks update or create dimension tables in BigQuery.

- Operator: BigQueryInsertJobOperator, which runs BigQuery jobs for inserting or updating data.

delay_bash_task:

- Function: A Bash task to introduce delays or perform system-level commands.

- Operator: BashOperator, executing a bash command.

create_update_fact_table:

- Function: Updates or creates fact tables in BigQuery.

- Operator: BigQueryInsertJobOperator.

train_model and **predict_model**:

- Function: These tasks involve training a machine learning model and making predictions.

- Operator: PythonOperator, which runs custom Python functions for machine learning tasks.

bigquery_to_gcs_task:

- Function: Exports data from BigQuery to Google Cloud Storage (GCS).

- Operator: BigQueryToGCSOperator.

3. Configuration

Apache NiFi:

Create a Compute Engine instance

Basic information

Name	nifi-instance
Instance Id	7700608401821766804
Description	None
Type	Instance
Status	Running
Creation time	May 27, 2024, 1:38:55 PM UTC+07:00
Zone	asia-southeast1-c
Instance template	None
In use by	None
Reservations	Automatically choose
Labels	None
Tags	–
Deletion protection	Disabled
Confidential VM service	Disabled
Preserved state size	0 GB

Machine configuration

Machine type	e2-medium
CPU platform	Intel Broadwell
Minimum CPU platform	None
Architecture	x86/64
vCPUs to core ratio	–
Custom visible cores	–
Display device	Disabled Enable to use screen capturing and recording tools
GPUs	None
Resource policies	

Boot disk

Name	Image	Interface type	Size (GB)	Device name	Type	Architecture	Encryption	Mode	Wh
nifi-instance	ubuntu-2004-focal-v20240519	SCSI	10	nifi-instance	Balanced persistent disk	x86/64	Google-managed	Boot, read/write	Del

Working with Apache Nifi

Using Controller Service: *GCPCredentialsControllerService*

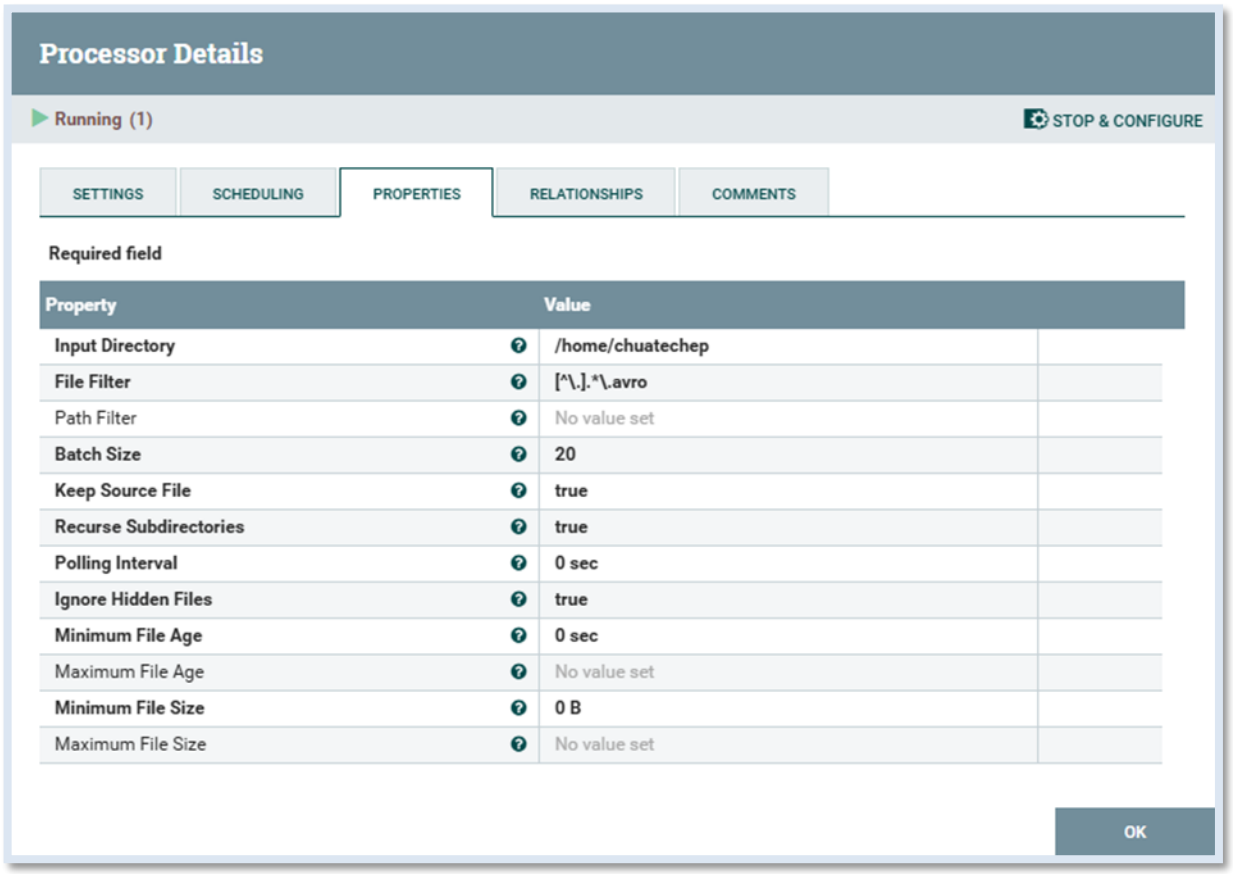
Controller Service Details | GCPCredentialsControllerService 1.16.0

SETTINGS | PROPERTIES | COMMENTS

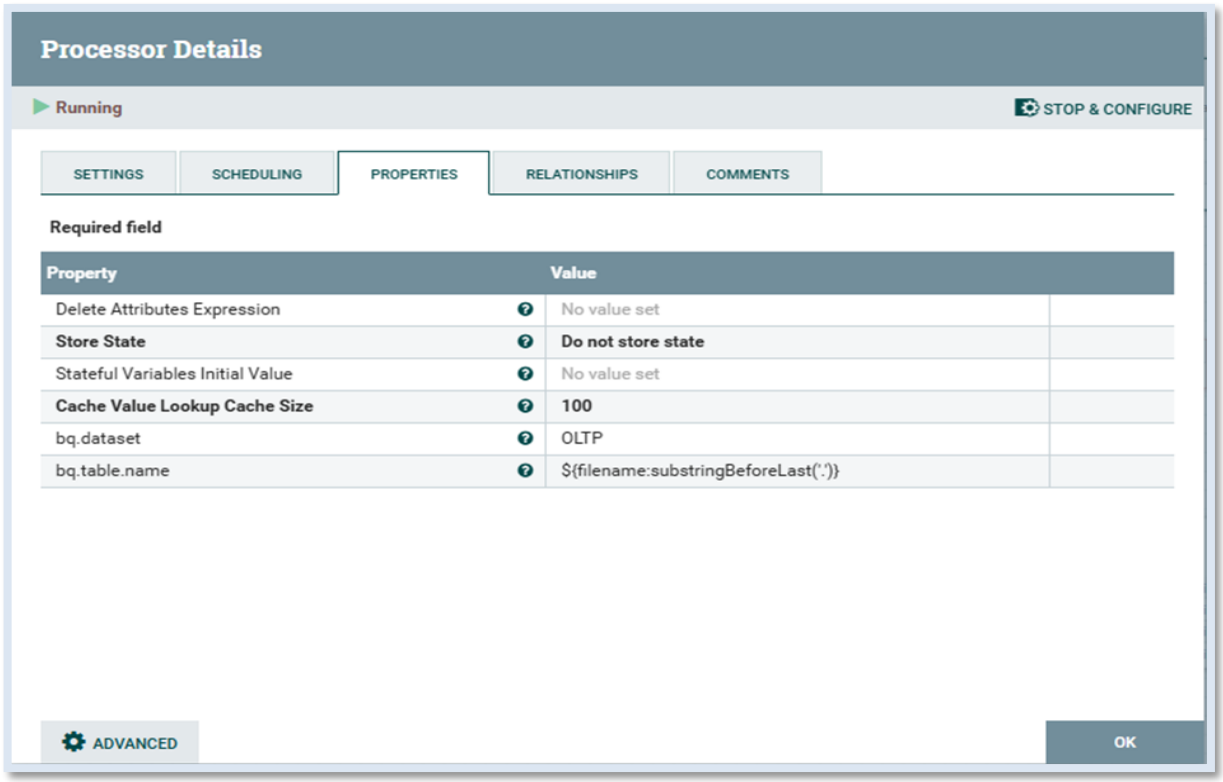
Required field

Property	Value
Use Application Default Credentials	false
Use Compute Engine Credentials	false
Service Account JSON File	/home/chuatechep/datawarehouse-422504-395...
Service Account JSON	No value set
Proxy Configuration Service	No value set

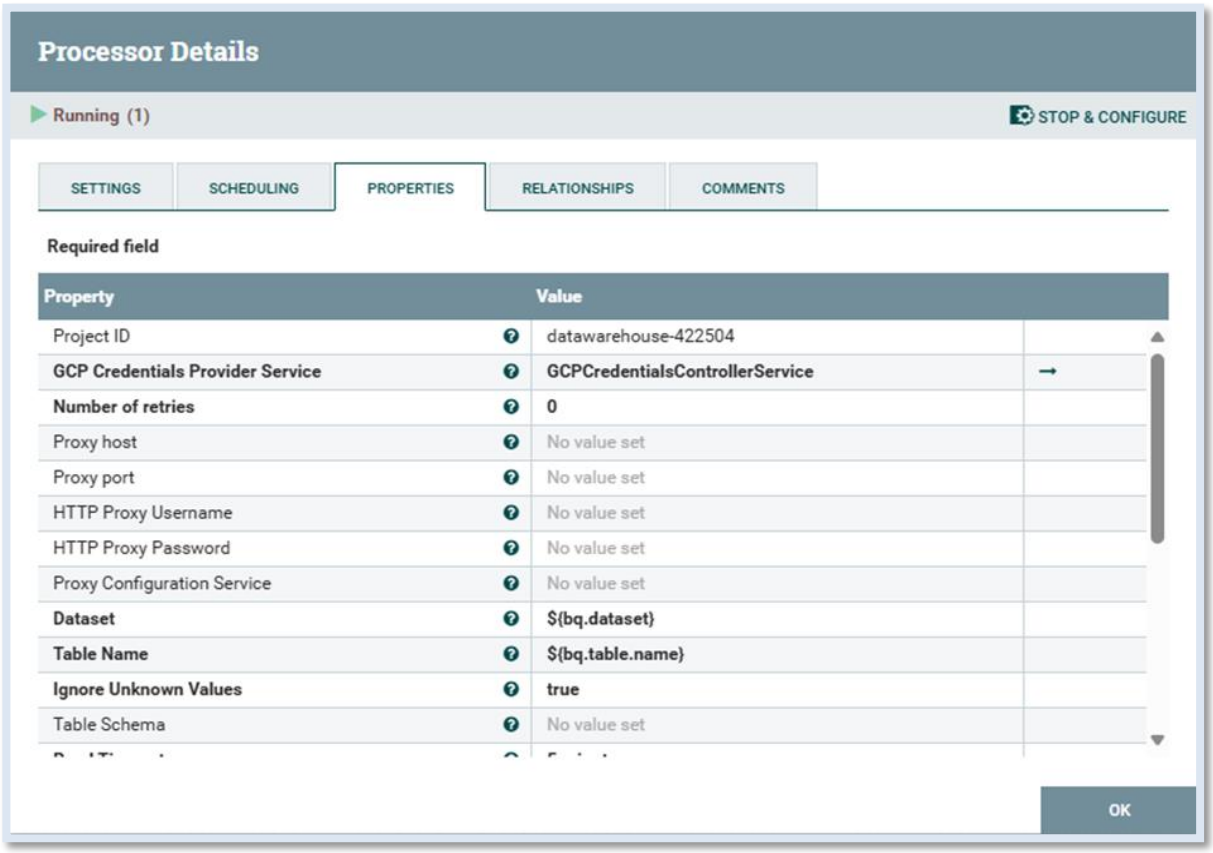
Configuration for *GetFile Processor:*

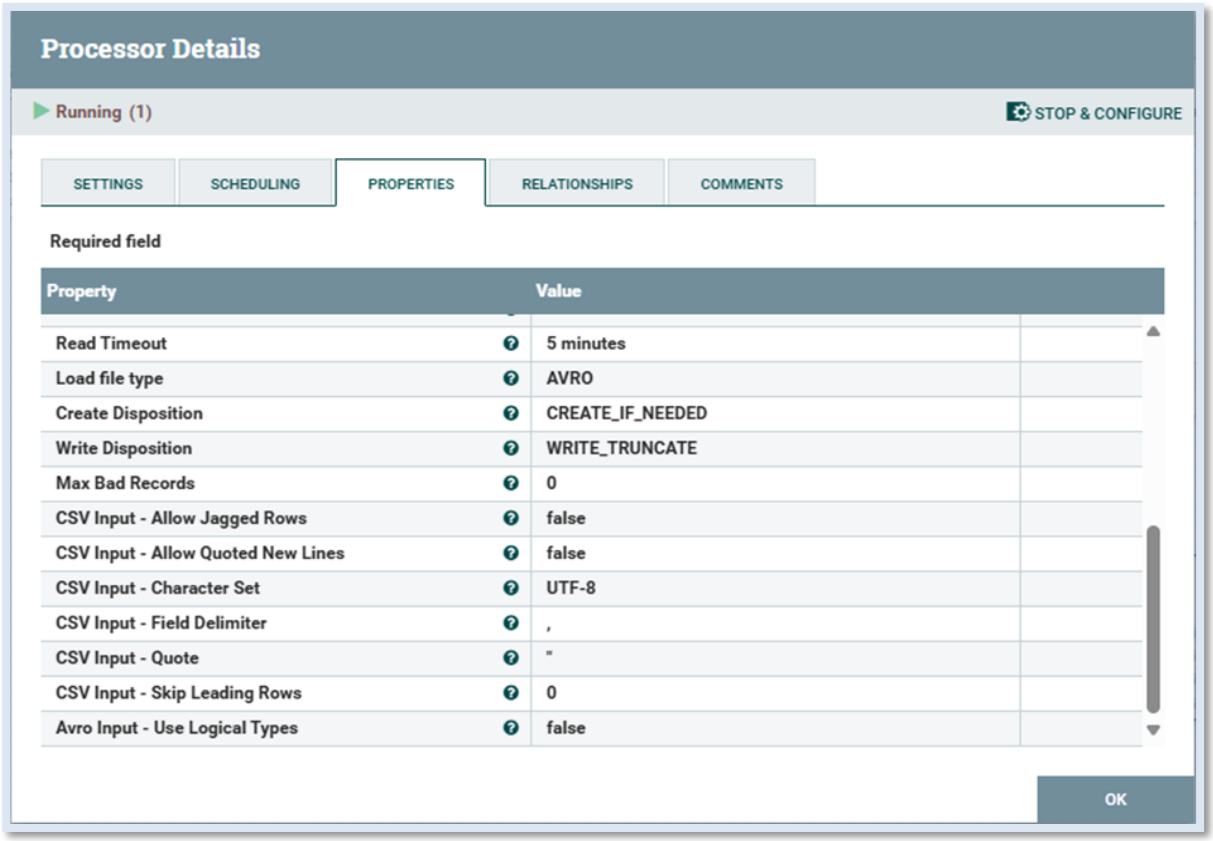


Configuration for *UpdateAttribute Processor:*



Configuration for *PutBigQueryBatch Processor:*





Apache Airflow:

```
# Import necessary modules and setting up DAG structure:
import os
from datetime import timedelta
from airflow.models.dag import DAG
from airflow.operators.python import PythonOperator
from airflow.providers.google.cloud.operators.bigquery import BigQueryInsertJobOperator
from airflow.providers.google.cloud.transfers.bigquery_to_gcs import BigQueryToGCSOperator
from airflow.utils.dates import days_ago
from airflow.operators.bash import BashOperator
from google.cloud import bigquery
import airflow
from Clustering_Kmeans_Model import train_model
from model_predicted_response import predict_model
from google.cloud import storage

os.environ['GOOGLE_APPLICATION_CREDENTIALS'] = '/opt/airflow/dags/datawarehouse-422504-39505bda63f7.json'

default_args = {
        'owner': 'airflow',
        'depends_on_past': False,
        'start_date': days_ago(1),
        'email_on_failure': False,
        'email_on_retry': False,
        'retries': None,
        'retry_delay': timedelta(minutes=2),
}

dag = DAG(
        'oltp_to_olap_transform',
        default_args=default_args,
        description='Transform OLTP to OLAP schema in BigQuery',
        template_searchpath='/opt/airflow/dags',
        schedule_interval=None,
)
# Function to read SQL Query needed for OLAP Transformation, which BigQuery will execute it:
def read_sql_file(file_path):
        with open(file_path, 'r') as file:
        return file.read()

# Read the SQL queries from files
DimCustomer_sql_query = read_sql_file('/opt/airflow/dags/dim_customer.sql')
DimDateTime_sql_query = read_sql_file('/opt/airflow/dags/dim_datetime.sql')
FactOrder_sql_query = read_sql_file('/opt/airflow/dags/fact.sql')
DimCampaignResponse_sql_query = read_sql_file('/opt/airflow/dags/dim_Campaigns_Response.sql')
DimPurchases_sql_query = read_sql_file('/opt/airflow/dags/dim_purchases.sql')
DimProductConsumption_sql_query = read_sql_file('/opt/airflow/dags/dim_product_consumption.sql')
# Define the loading task for each tables to BigQuery to execute:
# Define the tasks
t1 = BigQueryInsertJobOperator(
        task_id='create_dim_customer',
        configuration={
        "query": {
```

```
            "query": DimCustomer_sql_query,
            "useLegacySql": False
            }
            },
            dag=dag,
)

t2 = BigQueryInsertJobOperator(
            task_id='create_update_dim_Campaign_Response',
            configuration={
            "query": {
            "query": DimCampaignResponse_sql_query,
            "useLegacySql": False
            }
            },
            dag=dag,
)

t3 = BigQueryInsertJobOperator(
            task_id='create_update_dim_datetime',
            configuration={
            "query": {
            "query": DimDateTime_sql_query,
            "useLegacySql": False
            }
            },
            dag=dag,
)

t4 = BigQueryInsertJobOperator(
            task_id='create_update_dim_purchases',
            configuration={
            "query": {
            "query": DimPurchases_sql_query,
            "useLegacySql": False
            }
            },
            dag=dag,
)

t5 = BigQueryInsertJobOperator(
            task_id='create_update_dim_product_consumption',
            configuration={
            "query": {
            "query": DimProductConsumption_sql_query,
            "useLegacySql": False
            }
            },
            dag=dag,
)


t6 = BigQueryInsertJobOperator(
            task_id='create_update_fact_table',
            configuration={
            "query": {
            "query": FactOrder_sql_query,
```

```
            "useLegacySql": False
            }
            },
            dag=dag,
)



def model_training():
            train_model()

t7 = PythonOperator(
            task_id='train_model',
            python_callable=model_training,
            dag=dag,
)
# Create machine learning task and model predict:
def model_training():
            train_model()

t7 = PythonOperator(
            task_id='train_model',
            python_callable=model_training,
            dag=dag,
)
def model_predict():
            predict_model()

t8 = PythonOperator(
            task_id='predict_model',
            python_callable=model_predict,
            dag=dag,
)

TaskDelay = BashOperator(task_id="delay_bash_task",
                        dag=dag,
                        bash_command="sleep 5s")
# Store Fact table to GoogleCloudPlatform Buckets:
bq_to_gcs_task = BigQueryToGCSOperator(
            task_id='bigquery_to_gcs_task',
            source_project_dataset_table='datawarehouse-
422504.OLAP.fact_MarketingCampaignResponse',

            destination_cloud_storage_uris=['gs://datawarehouse12/fact_MarketingCampaignResponse.av
ro'],
            compression='NONE',
            export_format='AVRO',
            field_delimiter=',',
            print_header=True,
            dag=dag
)

# Define task dependencies
[t1, t2, t3, t4, t5] >> TaskDelay >> t6 >> [t7, t8] >> bq_to_gcs_task

if __name__ == "__main__":
            dag.cli()
```

#Define the train_model:

```
import os
from google.cloud import bigquery
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
import seaborn as sns
os.environ['GOOGLE_APPLICATION_CREDENTIALS'] = '/opt/airflow/dags/datawarehouse-422504-39505bda63f7.json'
# Create client for BQ
def train_model():
        client = bigquery.Client(project='datawarehouse-422504')

# Query from BigQuery
        query = """
        SELECT User_ID, Recency, Complain_Times, NumWebVisitsMonth, Total_Purchases, Total_Spent, Overall_Accept_Campaign, NumDealsPurchases, Response
        FROM `datawarehouse-422504.OLAP.fact_MarketingCampaignResponse`
        """
        df = client.query(query).to_dataframe()

        # Processing data
        X = df[
        ['Recency', 'Complain_Times', 'NumWebVisitsMonth', 'Total_Purchases', 'Total_Spent', 'Overall_Accept_Campaign',
        'NumDealsPurchases']]

        # data standard
        scaler = StandardScaler()
        X_scaled = scaler.fit_transform(X)

        # optimize cluster_num ussing elbow
        wcss = []
        for i in range(1, 11):
        kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=10, random_state=42)
        kmeans.fit(X_scaled)
        wcss.append(kmeans.inertia_)

        # elbow_graph
        plt.figure(figsize=(10, 5))
        plt.plot(range(1, 11), wcss, marker='o')
        plt.title('Elbow Method')
        plt.xlabel('Number of clusters')
        plt.ylabel('WCSS')
        plt.show()

        # train model KMeans with optimized cluster_num
        kmeans = KMeans(n_clusters=3, init='k-means++', max_iter=300, n_init=10, random_state=42)
        y_kmeans = kmeans.fit_predict(X_scaled)

        # add in label to DataFrame
        df['Cluster'] = y_kmeans
```

```
        # Cluster_summary
        cluster_summary = df.groupby('Cluster').mean()
        print("Cluster Summary:\n", cluster_summary)

        # visual
        sns.pairplot(df, hue='Cluster',
                     vars=['Recency', 'Complain_Times', 'NumWebVisitsMonth', 'Total_Purchases',
'Total_Spent',
                     'Overall_Accept_Campaign', 'NumDealsPurchases'])
        plt.show()
        # Push result back to BigQuery
        # Table existed or not
        table_id = "datawarehouse-422504.OLAP.Clustered_Customers"

        try:
        client.get_table(table_id)
        print("Existed. Replaced.")
        job_config =
bigquery.LoadJobConfig(write_disposition=bigquery.WriteDisposition.WRITE_TRUNCATE)
        except:
        print("Not exist. Create new.")
        job_config =
bigquery.LoadJobConfig(write_disposition=bigquery.WriteDisposition.WRITE_APPEND)

        job = client.load_table_from_dataframe(df, table_id, job_config=job_config)
        job.result()

        print("Successfull")
```

#Define the predict_model:

```
from google.cloud import bigquery
 import pandas as pd
 from sklearn.model_selection import train_test_split
 from sklearn.preprocessing import StandardScaler
 from sklearn.tree import DecisionTreeClassifier
 from sklearn.metrics import accuracy_score, classification_report
 from google.cloud.exceptions import NotFound
 import os

 os.environ['GOOGLE_APPLICATION_CREDENTIALS'] = '/opt/airflow/dags/datawarehouse-
422504-39505bda63f7.json'

 def predict_model():
 # Create client for BigQuery
        client = bigquery.Client(project='datawarehouse-422504')

 # Query data from BigQuery
        query = """
        SELECT User_ID, Recency, Complain_Times, NumWebVisitsMonth, Total_Purchases,
Total_Spent, Overall_Accept_Campaign, NumDealsPurchases, Response
        FROM `datawarehouse-422504.OLAP.fact_MarketingCampaignResponse`
        """
        df = client.query(query).to_dataframe()

 # Data processing
        X = df[['User_ID', 'Recency', 'Complain_Times', 'NumWebVisitsMonth', 'Total_Purchases',
```

```
'Total_Spent',
        'Overall_Accept_Campaign', 'NumDealsPurchases']]
        y = df['Response']

# split train/test
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# save User_ID
        X_train_user_id = X_train['User_ID']
        X_test_user_id = X_test['User_ID']

# user id not included in train/test
        X_train = X_train.drop(columns=['User_ID'])
        X_test = X_test.drop(columns=['User_ID'])

# data standard
        scaler = StandardScaler()
        X_train_scaled = scaler.fit_transform(X_train)
        X_test_scaled = scaler.transform(X_test)

# Decision Trees
        tree_model = DecisionTreeClassifier(random_state=42)
        tree_model.fit(X_train_scaled, y_train)

# predict on test
        y_pred = tree_model.predict(X_test_scaled)

# Evaluate
        accuracy = accuracy_score(y_test, y_pred)
        print("Accuracy:", accuracy)
        print("Classification Report:\n", classification_report(y_test, y_pred))

# Data frame for train
        df_results_train = pd.DataFrame({
        'User_ID': X_train_user_id.values,
        'Recency': X_train['Recency'].values,
        'Complain_Times': X_train['Complain_Times'].values,
        'NumWebVisitsMonth': X_train['NumWebVisitsMonth'].values,
        'Total_Purchases': X_train['Total_Purchases'].values,
        'Total_Spent': X_train['Total_Spent'].values,
        'Overall_Accept_Campaign': X_train['Overall_Accept_Campaign'].values,
        'NumDealsPurchases': X_train['NumDealsPurchases'].values,
        'Actual_Response': y_train.values,
        'Predicted_Response': tree_model.predict(X_train_scaled).astype(int)
        })

# DataFrame for test
        df_results_test = pd.DataFrame({
        'User_ID': X_test_user_id.values,
        'Recency': X_test['Recency'].values,
        'Complain_Times': X_test['Complain_Times'].values,
        'NumWebVisitsMonth': X_test['NumWebVisitsMonth'].values,
        'Total_Purchases': X_test['Total_Purchases'].values,
        'Total_Spent': X_test['Total_Spent'].values,
        'Overall_Accept_Campaign': X_test['Overall_Accept_Campaign'].values,
        'NumDealsPurchases': X_test['NumDealsPurchases'].values,
        'Actual_Response': y_test.values,
```

```
        'Predicted_Response': y_pred.astype(int)
        })

# Combine train test sets
        df_results = pd.concat([df_results_train, df_results_test])

        table_id = "datawarehouse-422504.OLAP.Predicted_Responses"
        try:
        client.get_table(table_id)
        print("Existed. Replaced.")
        job_config =
bigquery.LoadJobConfig(write_disposition=bigquery.WriteDisposition.WRITE_TRUNCATE)
        except NotFound:
        print("Create new table")
        job_config =
bigquery.LoadJobConfig(write_disposition=bigquery.WriteDisposition.WRITE_APPEND)

        job = client.load_table_from_dataframe(df_results, table_id, job_config=job_config)
        job.result()

        print("Successful")
```

III. Building Data Warehouse

1. Choose the Business Process

A business process represents a major operational activity within an organization, encapsulating a series of tasks or workflows that produce a specific service or product for customers. In the context of data warehousing, selecting an appropriate business process is crucial as it sets the foundation for structuring and organizing the data. The chosen process should be central to the organization's objectives and provide significant insights into operational efficiency and effectiveness.

In this project, the potential business processes are Marketing Campaign Responses, Customer Segmentation, and Campaign Performance Analysis. Marketing Campaign Responses involve analyzing how customers respond to marketing efforts. Customer Segmentation focuses on understanding different customer segments based on purchasing and response behaviors. Campaign Performance Analysis evaluates the overall effectiveness of marketing campaigns. Each of these business processes corresponds to a fact table included in the data warehouse schema. By focusing on these processes, we can address key business questions such as, "Which customer segments

are most likely to respond positively to future marketing campaigns?" and "How effective have past marketing campaigns been in driving sales and engagement?"

2. Declare the Grain

Transaction Level (Atomic Level):

For our project, the chosen grain is the transaction level, epitomized by the FACT_MarketingCampaignResponse fact table. At this finest granularity, each row within FACT_MarketingCampaignResponse meticulously captures the details of individual customer interactions with marketing campaigns. This approach aligns with the intrinsic nature of our primary OLTP data sources, which serve as transaction tables, providing comprehensive insights into each customer's campaign response.

Daily/Periodic Level:

To facilitate analysis over specific time frames, our design incorporates the DimDateTime dimension. This enables aggregation at daily or periodic levels, allowing us to summarize customer responses and purchase behaviors over distinct time intervals. The Response_Date_ID attribute in DimDateTime acts as a linchpin for time-based analyses, supporting the extraction of meaningful patterns and trends.

Summary Level:

Elevating our perspective for broader business insights, especially in monthly or yearly contexts, involves leveraging additional dimensions such as DimCustomer, DimPurchases, DimProduct_Consumption. Aggregating data based on these dimensions provides a higher-level view of campaign performance across key business aspects. This summary-level granularity proves invaluable for strategic decision-making and long-term planning.

The rationale behind implementing a transaction fact table stems from the inherent characteristics of our underlying data sources. By selecting this granularity, we opt for the lowest level of detail, capturing individual customer interactions with marketing campaigns. This strategic choice not only aligns with best practices but also positions us for flexibility in subsequent analyses and reporting. In essence, our data warehouse is designed to seamlessly transition from the intricacies of individual transactions to the broader perspectives essential for strategic business insights.

3. Choose the Dimensions

Dimension Table: Dim_Customer

- Attributes:

- User_ID: A unique identifier for each customer.

- Year_Birth: Birth Year of the customer

- Education: Education Level of the customer

- Marital_Status: Marital Status of the customer

- Income: Annual Income of the customer

- Kidhome: Number of Children in the Home

- Teenhome: Number of Teenagers in the Home

- Country: Country of the customer

```
CREATE OR REPLACE TABLE `datawarehouse-422504.OLAP.Dim_Customer` AS
SELECT DISTINCT
  User_ID,
  Year_Birth,
  Education,
  Marital_Status,
  Income,
  Kidhome,
  Teenhome,
  Country
FROM `datawarehouse-422504.OLTP.Customer`;
```

Dimension Table: Dim_DateTime

- Attributes:

- Response_Date_ID: A unique identifier for each date of response.

- Date: The actual date of entry

- Day: The day component of the date

- Month: The month component of the date

- Quarter: The quarter component of the date

- Year: The year component of the date

```
CREATE OR REPLACE TABLE `datawarehouse-422504.OLAP.Dim_DateTime` AS
WITH Date_CTE AS (
 SELECT
    ROW_NUMBER() OVER (ORDER BY CAST(Date_Response AS DATE)) AS
Response_Date_ID,
    CAST(Date_Response AS DATE) AS Date,
    EXTRACT(DAY FROM CAST(Date_Response AS DATE)) AS Day,
    EXTRACT(MONTH FROM CAST(Date_Response AS DATE)) AS Month,
    EXTRACT(QUARTER FROM CAST(Date_Response AS DATE)) AS Quarter,
    EXTRACT(YEAR FROM CAST(Date_Response AS DATE)) AS Year
 FROM `datawarehouse-422504.OLTP.Campaign_Response`
)
SELECT
 Response_Date_ID,
 Date,
 Day,
 Month,
 Quarter,
 Year
FROM Date_CTE;
```

Dimension Table: Dim_Campaign_Response

- Attributes:

- Response_ID: A unique identifier for each campaign response

- Transformation: For the AcceptedCmp 1 – 5: (Transform from String to Binary)

If the original value is 'True' (string), it is converted to 1.

If the original value is 'False' (string), it is converted to 0.

If the original value is neither 'True' nor 'False', it is cast to an INTEGER as-is.

- AcceptedCmp1: Response to Marketing Campaign 1

- AcceptedCmp2: Response to Marketing Campaign 2

- AcceptedCmp3: Response to Marketing Campaign 3

- AcceptedCmp4: Response to Marketing Campaign 4

- AcceptedCmp5: Response to Marketing Campaign 5

```
CREATE OR REPLACE TABLE `datawarehouse-422504.OLAP.Dim_Campaign_Response` AS
SELECT DISTINCT
 Response_ID,
 CASE
    WHEN CAST(AcceptedCmp1 AS STRING) = 'True' THEN 1
    WHEN CAST(AcceptedCmp1 AS STRING) = 'False' THEN 0
    ELSE CAST(AcceptedCmp1 AS INTEGER)
 END AS AcceptedCmp1,
 CASE
    WHEN CAST(AcceptedCmp2 AS STRING) = 'True' THEN 1
    WHEN CAST(AcceptedCmp2 AS STRING) = 'False' THEN 0
    ELSE CAST(AcceptedCmp2 AS INTEGER)
 END AS AcceptedCmp2,
 CASE
    WHEN CAST(AcceptedCmp3 AS STRING) = 'True' THEN 1
    WHEN CAST(AcceptedCmp3 AS STRING) = 'False' THEN 0
    ELSE CAST(AcceptedCmp3 AS INTEGER)
 END AS AcceptedCmp3,
 CASE
    WHEN CAST(AcceptedCmp4 AS STRING) = 'True' THEN 1
    WHEN CAST(AcceptedCmp4 AS STRING) = 'False' THEN 0
    ELSE CAST(AcceptedCmp4 AS INTEGER)
 END AS AcceptedCmp4,
 CASE
    WHEN CAST(AcceptedCmp5 AS STRING) = 'True' THEN 1
    WHEN CAST(AcceptedCmp5 AS STRING) = 'False' THEN 0
    ELSE CAST(AcceptedCmp5 AS INTEGER)
 END AS AcceptedCmp5
FROM `datawarehouse-422504.OLTP.Campaign_Response`;
```

Dimension Table: Dim_Product_Consumption

- Attributes:

- Consumption_ID: A unique identifier for each consumption of product

- MntWines: Amount Spent on Wine

- MntFruits: Amount Spent on Fruits

- MntMeats: Amount Spent on Meats

- MntFishs: Amount Spent on Fishes

- MntSweets: Amount Spent on Sweets

- MntGolds: Amount Spent on Gold Products

```
CREATE OR REPLACE TABLE `datawarehouse-422504.OLAP.Dim_Product_Consumptionr` AS
SELECT DISTINCT
 Consumption_ID,
 MntWines,
 MntFruits,
 MntMeats,
 MntFishs,
 MntSweets,
 MntGolds
FROM `datawarehouse-422504.OLTP.Product_Consumption`;
```

Dimension Table: Dim_Purchases

- Attributes:

- Purchase_ID: A unique identifier for each purchase

- NumWebPurchases: Number of Purchases through Website

- NumCatalogPurchases: Number of Purchases through Catalog

- NumStorePurchases: Number of In-Store Purchases

```
CREATE OR REPLACE TABLE `datawarehouse-422504.OLAP.Dim_Purchases` AS
SELECT DISTINCT
  Purchase_ID,
  NumWebPurchases,
  NumCatalogPurchases,
  NumStorePurchases
FROM
  `datawarehouse-422504.OLTP.Purchases`;
```

4. Choose the Facts

Fact_MarketingCampaignResponse Table Attributes

- User_ID: A unique identifier for each customer

- Purchase_ID: A unique identifier for each purchase

- Consumption_ID: A unique identifier for each consumption

- Response_ID: A unique identifier for each response

- Response_Date_ID: A unique identifier for each date of response

- Complain_Times: Number of customer complaint time

- NumWebVisitsMonth: Number of Website Visits

- Recency: Days since Last Purchase

- Total_Purchases: The total number of purchases made by the user across web, catalog, and store.

- Response: Response to the latest marketing campaign

If the response is 'True' (string), it is converted to 1.

If the response is 'False' (string), it is converted to 0.

If neither, it remains NULL.

- Total_Spent: The total amount spent by the user on all product categories

- Overall_Accept_Campaign: The total number of accepted campaigns by the user

- NumDealsPurchases: The number of purchases made by the user using deals

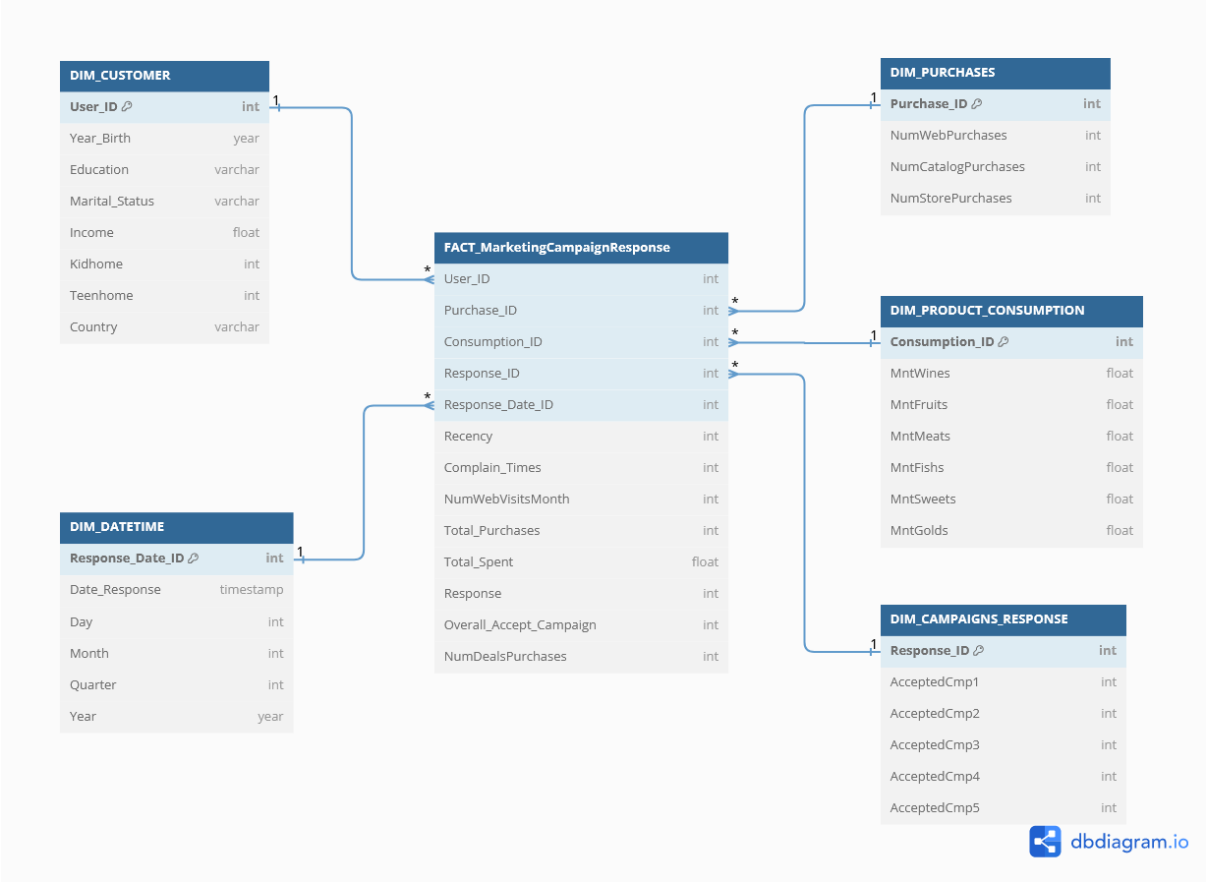
```
CREATE OR REPLACE TABLE `datawarehouse-422504.OLAP.fact_MarketingCampaignResponse`
AS
 WITH Campaign_Acceptance AS (
        SELECT
        cr.User_ID,
        MAX(dcr.AcceptedCmp1) AS AcceptedCmp1,
        MAX(dcr.AcceptedCmp2) AS AcceptedCmp2,
        MAX(dcr.AcceptedCmp3) AS AcceptedCmp3,
        MAX(dcr.AcceptedCmp4) AS AcceptedCmp4,
        MAX(dcr.AcceptedCmp5) AS AcceptedCmp5
```

```
        FROM
        `datawarehouse-422504.OLTP.Campaign_Response` cr
        LEFT JOIN `datawarehouse-422504.OLAP.Dim_Campaign_Response` dcr ON
cr.Response_ID = dcr.Response_ID
        GROUP BY
        cr.User_ID
),
Product_Spend AS (
        SELECT
        pc.User_ID,
        MAX(pc.MntWines) AS Total_Wines,
        MAX(pc.MntFruits) AS Total_Fruits,
        MAX(pc.MntMeats) AS Total_Meats,
        MAX(pc.MntFishs) AS Total_Fish,
        MAX(pc.MntSweets) AS Total_Sweets,
        MAX(pc.MntGolds) AS Total_Golds
        FROM
        `datawarehouse-422504.OLTP.Product_Consumption` pc
        GROUP BY
        pc.User_ID
),
Total_Purchase AS (
        SELECT
        pur.User_ID,
        MAX(pur.NumWebPurchases) AS Total_Web_Purchases,
        MAX(pur.NumCatalogPurchases) AS Total_Catalog_Purchases,
        MAX(pur.NumStorePurchases) AS Total_Store_Purchases
        FROM
        `datawarehouse-422504.OLTP.Purchases` pur
        GROUP BY
        pur.User_ID
)
SELECT DISTINCT
        cp.User_ID,
        MIN(pur.Purchase_ID) AS Purchase_ID,
        MIN(pc.Consumption_ID) AS Consumption_ID,
        MIN(cr.Response_ID) AS Response_ID,
        MIN(dd.Response_Date_ID) AS Response_Date_ID,
```

```
        cp.Recency,
        MIN(com.Complain_Times) AS Complain_Times,
        MIN(pur.NumWebVisitsMonth) AS NumWebVisitsMonth,
        COALESCE(SUM(tp.Total_Web_Purchases), 0) +
COALESCE(SUM(tp.Total_Catalog_Purchases), 0) + COALESCE(SUM(tp.Total_Store_Purchases), 0)
AS Total_Purchases,
        COALESCE(MAX(ps.Total_Wines), 0) + COALESCE(MAX(ps.Total_Fruits), 0) +
COALESCE(MAX(ps.Total_Meats), 0) + COALESCE(MAX(ps.Total_Fish), 0) +
COALESCE(MAX(ps.Total_Sweets), 0) + COALESCE(MAX(ps.Total_Golds), 0) AS Total_Spent,
        CASE
        WHEN MIN(cr.Response) = 'True' THEN 1
        WHEN MIN(cr.Response) = 'False' THEN 0
        ELSE NULL
        END AS Response,
        COALESCE(MAX(ca.AcceptedCmp1), 0) + COALESCE(MAX(ca.AcceptedCmp2), 0) +
COALESCE(MAX(ca.AcceptedCmp3), 0) + COALESCE(MAX(ca.AcceptedCmp4), 0) +
COALESCE(MAX(ca.AcceptedCmp5), 0) AS Overall_Accept_Campaign,
        MIN(de.NumDealsPurchases) AS NumDealsPurchases
 FROM
        `datawarehouse-422504.OLTP.Customer` cp
        LEFT JOIN `datawarehouse-422504.OLTP.Purchases` pur ON cp.User_ID = pur.User_ID
        LEFT JOIN `datawarehouse-422504.OLTP.Product_Consumption` pc ON cp.User_ID =
pc.User_ID
        LEFT JOIN Product_Spend ps ON cp.User_ID = ps.User_ID
        LEFT JOIN Total_Purchase tp ON cp.User_ID = tp.User_ID
        LEFT JOIN `datawarehouse-422504.OLTP.Campaign_Response` cr ON cp.User_ID =
cr.User_ID
        LEFT JOIN `datawarehouse-422504.OLAP.Dim_DateTime` dd ON
SAFE_CAST(cr.Date_Response AS DATE) = dd.Date
        LEFT JOIN `datawarehouse-422504.OLTP.Complain` com ON cp.User_ID = com.User_ID
        LEFT JOIN `datawarehouse-422504.OLTP.Deals` de ON cp.User_ID = de.User_ID
        LEFT JOIN Campaign_Acceptance ca ON cp.User_ID = ca.User_ID
 GROUP BY
        cp.User_ID, cp.Recency;
```

The OLAP (Online Analytical Processing) database is designed to facilitate complex queries and analytical processes. The provided ERD (Entity Relationship Diagram)

illustrates the relationships between key dimensional tables and the fact table, which are central to the data warehouse architecture.



The ERD demonstrates the star schema design, with the Fact_MarketingCampaignResponse table at the center, surrounded by dimension tables. This design supports efficient data retrieval for analytical purposes, enabling comprehensive insights into customer behavior, campaign responses, product consumption, and purchase activities. Foreign keys link the dimension tables to the fact table, ensuring referential integrity and optimized query performance, essential for OLAP systems.

This OLAP system is structured to support in-depth analysis and reporting, facilitating data-driven decision-making for marketing strategies, customer segmentation, and sales optimization.

IV. Business Analytics

1. Query in OLAP

Following the creation of the OLAP database, we proceed to leverage BigQuery for

formulating analytical queries aimed at extracting valuable insights.

```
--Which country has the highest Marketing Campaign Acceptance Rate?
SELECT
  dc.Country,
  SUM(fmc.Overall_Accept_Campaign) AS Total_Accepted,
  COUNT(fmc.Response) AS Total_Campaigns,
  (SUM(fmc.Overall_Accept_Campaign) / COUNT(fmc.Response)) * 100 AS Acceptance_Rate
FROM
  `OLAP.fact_MarketingCampaignResponse` fmc
JOIN
  `OLAP.Dim_Customer` dc ON fmc.User_ID = dc.User_ID
GROUP BY
  dc.Country
ORDER BY
  Acceptance_Rate DESC;
```

Query results

JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS

Row	Country	Total_Accepted	Total_Campaigns	Acceptance_Rate
1	Montenegro	1	3	33.33333333333...
2	Canada	87	268	32.46268656716...
3	Spain	355	1095	32.42009132420...
4	Germany	38	120	31.66666666666...
5	India	39	148	26.35135135135...
6	South Africa	86	337	25.51928783382...
7	United States	26	109	23.85321100917...
8	Australia	35	160	21.875

Based on the results, Montenegro, Canada, and Spain show the highest acceptance rates for marketing campaigns, indicating these countries should be key focus areas for future efforts. Germany, India, and South Africa also demonstrate strong potential and should be targeted with tailored marketing strategies. For the United States and Australia, optimizing campaign approaches could improve their moderate acceptance rates.

--Which customer segments have historically spent the most across different categories (e.g., MntWines, MntFruits)

```
SELECT
   dc.User_ID,
   dc.Country,
   AVG(pc.MntWines) AS Avg_Wines,
   AVG(pc.MntFruits) AS Avg_Fruits,
   AVG(pc.MntMeats) AS Avg_Meats,
   AVG(pc.MntFishs) AS Avg_Fish,
   AVG(pc.MntSweets) AS Avg_Sweets,
  FROM
   `OLAP.fact_MarketingCampaignResponse` fmc
JOIN
   `OLAP.Dim_Customer` dc ON fmc.User_ID = dc.User_ID
JOIN
   `OLAP.Dim_Product_Consumption` pc ON fmc.Consumption_ID = pc.Consumption_ID
GROUP BY
   dc.User_ID, dc.Country
ORDER BY
   (Avg_Wines + Avg_Fruits + Avg_Meats + Avg_Fish + Avg_Sweets) DESC
LIMIT 10;
```

Query results

JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	User_ID	Country	Avg_Wines	Avg_Fruits	Avg_Meats	Avg_Fish	Avg_Sweets
1	1763	Canada	1259.0	172.0	815.0	97.0	148.0
2	5350	South Africa	1156.0	120.0	915.0	94.0	144.0
3	5735	Spain	1156.0	120.0	915.0	94.0	144.0
4	10133	Canada	1302.0	68.0	731.0	89.0	114.0
5	4580	South Africa	1394.0	22.0	708.0	89.0	91.0
6	4475	Spain	1315.0	22.0	780.0	145.0	0.0
7	5453	South Africa	1083.0	108.0	649.0	253.0	151.0
8	737	Spain	1493.0	86.0	454.0	112.0	43.0
9	6248	Spain	1276.0	24.0	746.0	94.0	29.0
10	7503	Germany	1032.0	105.0	779.0	137.0	105.0

The results indicate that customers from Spain and South Africa have historically spent the most across various product categories, including wine, fruits, meats, fish, sweets, and gold. Notably, these countries also exhibit high marketing campaign acceptance rates, with Spain at 32.42% and South Africa at 25.52%. This correlation suggests that Spain and South Africa are highly valuable markets, where increased marketing investment could yield significant returns. Therefore, focusing on these markets could enhance campaign effectiveness and customer engagement.

```
--What are the acceptance rates of marketing campaigns across different purchase methods (web, catalog, store) in each country?
SELECT
  dc.Country,
  SUM(dp.NumWebPurchases) AS Total_Web_Purchases,
  SUM(dp.NumCatalogPurchases) AS Total_Catalog_Purchases,
  SUM(dp.NumStorePurchases) AS Total_Store_Purchases,
  SUM(fmc.Overall_Accept_Campaign) AS Total_Accepted,
  ROUND((SUM(fmc.Overall_Accept_Campaign) / SUM(dp.NumWebPurchases)) * 100, 2) AS Web_Acceptance_Rate,
  ROUND((SUM(fmc.Overall_Accept_Campaign) / SUM(dp.NumCatalogPurchases)) * 100, 2) AS Catalog_Acceptance_Rate,
  ROUND((SUM(fmc.Overall_Accept_Campaign) / SUM(dp.NumStorePurchases)) * 100, 2) AS Store_Acceptance_Rate
FROM
  `OLAP.fact_MarketingCampaignResponse` fmc
JOIN
  `OLAP.Dim_Customer` dc ON fmc.User_ID = dc.User_ID
JOIN
  `OLAP.Dim_Purchases` dp ON fmc.Purchase_ID = dp.Purchase_ID
GROUP BY
  dc.Country
ORDER BY
  Total_Accepted DESC
LIMIT 10;
```

Query results

JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	Country	Total_Web_Purchase	Total_Catalog_Purchases	Total_Store_Purchases	Total_Accepted	Web_Acceptance_Rate	Catalog_Acceptance_Rate	Store_Acceptance_Rate
1	Spain	4382	2849	6352	355	8.1	12.46	5.59
2	Canada	1154	735	1568	87	7.54	11.84	5.55
3	South Africa	1397	917	1988	86	6.16	9.38	4.33
4	India	584	365	785	39	6.68	10.68	4.97
5	Germany	477	332	721	38	7.97	11.45	5.27
6	Australia	654	419	879	35	5.35	8.35	3.98
7	United States	484	331	658	26	5.37	7.85	3.95
8	Montenegro	18	15	19	1	5.56	6.67	5.26

The results show that Spain has the highest total campaign acceptances and notable acceptance rates across all purchase methods, particularly through catalog purchases at 12.46%. Canada also demonstrates strong performance in catalog acceptance at 11.84%,

while South Africa shows balanced acceptance rates across web, catalog, and store purchases. Interestingly, India has the highest web acceptance rate at 6.68%, suggesting a strong engagement through online channels. These insights indicate that focusing marketing efforts on catalog and web channels could be particularly effective in these countries.

```
--How do factors like marital status, number of children at home (Kidhome, Teenhome), and education level affect customer purchasing habits and responsiveness to campaigns?
SELECT
  dc.Marital_Status,
  dc.Education,
  dc.Kidhome,
  dc.Teenhome,
  SUM(fmc.Total_Spent) AS Total_Spent,
  SUM(fmc.Overall_Accept_Campaign) as Total_accepted,
  COUNT(fmc.Response) AS Total_Responses
FROM
  `OLAP.fact_MarketingCampaignResponse` fmc
JOIN
  `OLAP.Dim_Customer` dc ON fmc.User_ID = dc.User_ID
GROUP BY
  dc.Marital_Status, dc.Education, dc.Kidhome, dc.Teenhome
ORDER BY
  Total_accepted DESC
LIMIT 10;
```

Query results

JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS | EXECUTION GRAPH

Row	Marital_Status	Education	Kidhome	Teenhome	Total_Spent	Total_accepted	Total_Responses
1	Married	Graduation	0	0	119322.0	69	116
2	Together	Graduation	0	0	99742.0	57	75
3	Married	PhD	0	0	64595.0	38	49
4	Single	Graduation	0	0	91311.0	38	86
5	Together	Graduation	0	1	63117.0	29	84
6	Married	Graduation	0	1	90145.0	26	122
7	Divorced	Graduation	0	0	30056.0	25	31
8	Single	PhD	0	0	41074.0	24	38
9	Married	PhD	0	1	52146.0	23	65
10	Single	Master	0	0	42489.0	22	29

Based on the query results, customers with a marital status of "Married" and "Together" exhibit the highest total spending and campaign acceptance rates, indicating a higher level of engagement and responsiveness to marketing efforts. Customers with

a "Graduation" level of education consistently appear among the top spenders and responders, suggesting that educational attainment may positively influence purchasing behavior and campaign engagement. Interestingly, the presence of children at home (Kidhome, Teenhome) does not significantly appear among the top segments, implying that family size might have a lesser impact on spending and responsiveness compared to marital status and education level. These insights suggest focusing marketing strategies on married or partnered individuals with higher education levels to maximize campaign effectiveness.

```
--How does the frequency of web visits correlate with purchasing behavior and campaign responsiveness?
SELECT
   dc.User_ID,
   fmc.NumWebVisitsMonth,
   SUM(fmc.Total_Spent) AS Total_Spent,
   SUM(fmc.Overall_Accept_Campaign) AS Total_Accepted
FROM
   `OLAP.fact_MarketingCampaignResponse` fmc
JOIN
   `OLAP.Dim_Customer` dc ON fmc.User_ID = dc.User_ID
GROUP BY
   dc.User_ID, fmc.NumWebVisitsMonth
ORDER BY
    fmc.NumWebVisitsMonth DESC
LIMIT 10;
```

Query results

JOB INFORMATION | RESULTS | CHART | JSON | EXECUTION DETAILS

Row	User_ID	NumWebVisitsMonth	Total_Spent	Total_Accepted
1	4303	20	137.0	0
2	6862	20	8.0	0
3	5899	20	49.0	1
4	9931	19	9.0	0
5	10749	19	178.0	0
6	4246	17	373.0	0
7	3955	14	6.0	0
8	11110	14	5.0	0
9	9303	13	32.0	0
10	7286	10	55.0	0

The data shows that high web visit frequency does not necessarily lead to high spending or campaign acceptance, as many users with frequent visits have low total spending and campaign acceptance rates. For instance, several users with 20 web visits have minimal spending and zero campaign acceptance. This suggests a need for personalized campaigns and enhanced engagement strategies to convert frequent web visitors into active spenders and campaign responders.

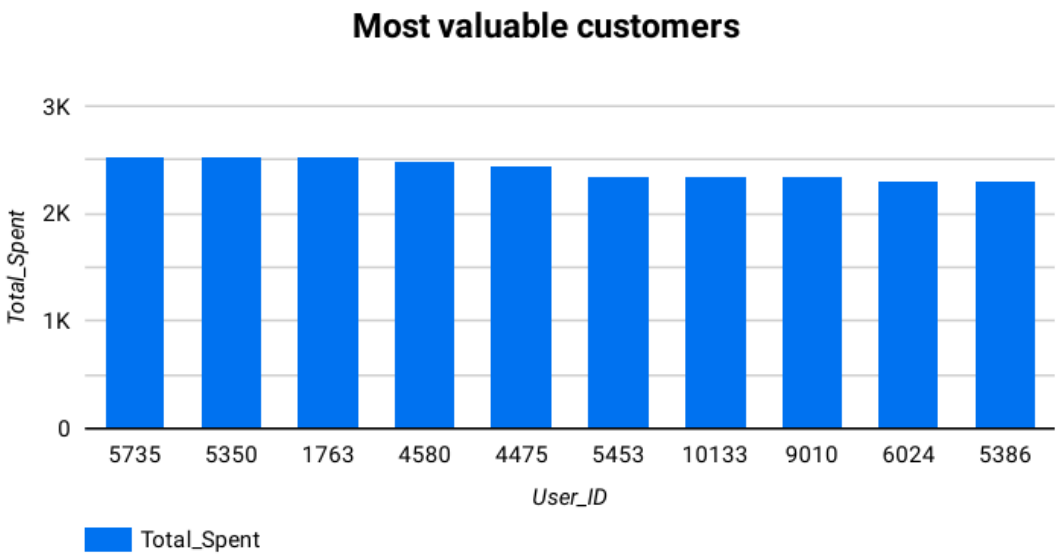
2. Data Visualization

2.1. Introduction

This report provides a comprehensive analysis of customer spending patterns, campaign effectiveness, and segment-specific behaviors based on various visualizations from the marketing campaign dashboard. The goal is to derive actionable insights for optimizing marketing strategies and improving customer engagement and retention.

2.2. Analysis and Insights

2.2.1. Most valuable customers



Explanation:

This chart displays the total amount spent by each customer. The horizontal bars represent individual customers (identified by User_ID), and the length of each bar indicates the total spending by that customer across all product categories. The chart is sorted to show the top spenders at the top.

Insights:

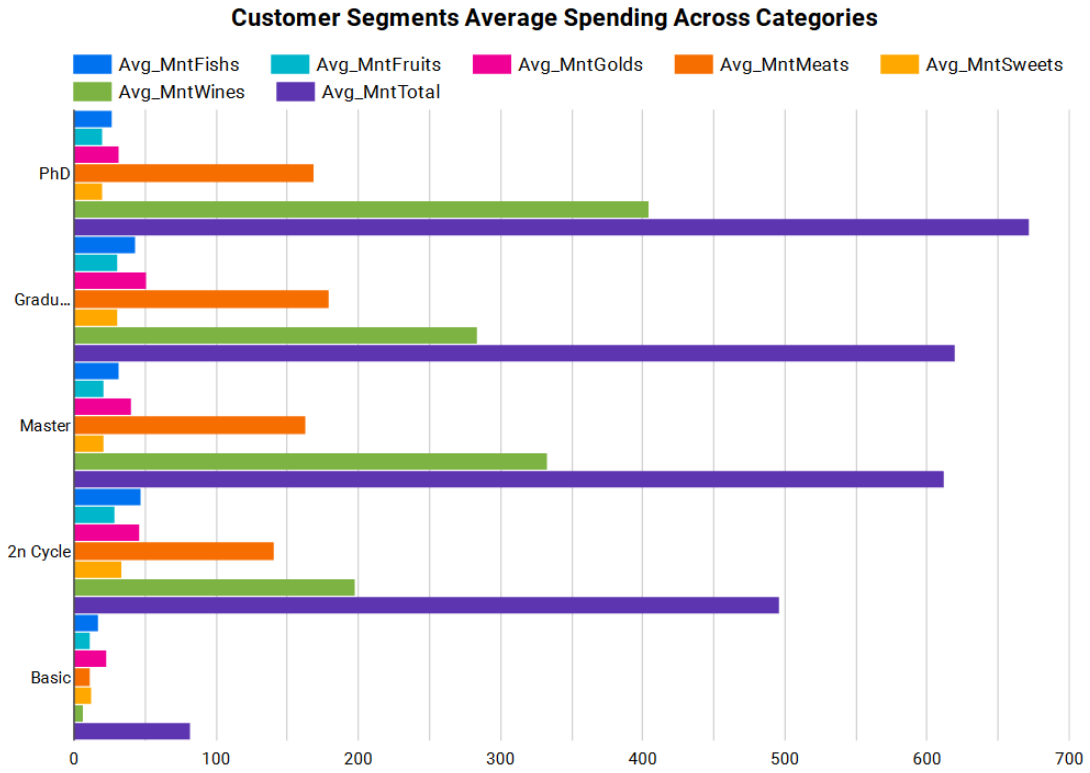
The chart highlights the top 10 spenders. These customers are crucial for revenue generation and should be prioritized in marketing strategies.

Focusing retention efforts on these high-spending customers can prevent potential revenue loss. Personalized loyalty programs and exclusive offers can help maintain their engagement.

Interpretation:

Understanding who the most valuable customers are allows the business to allocate more resources to maintain and grow these relationships. Exclusive offers, loyalty programs, and personalized communications can be tailored to these customers to increase engagement and spending.

2.2.2. Customer segments average spending across categories



Explanation:

This chart illustrates the average spending across various product categories segmented by education level and marital status.

Insights:

PhD Holders: Highest average spending on Wines and Meats, indicating a strong

preference for premium products. Promotions emphasizing quality, exclusivity, and premium attributes of wines and meats may be particularly effective for this segment.

Graduates: Balanced average spending across multiple categories, with noticeable spending on Fruits and Meats. Graduates may respond well to health-focused campaigns highlighting fresh fruits and lean meats.

Masters: Consistent average spending across categories, with significant spending on Meats and Wines, similar to PhD holders.

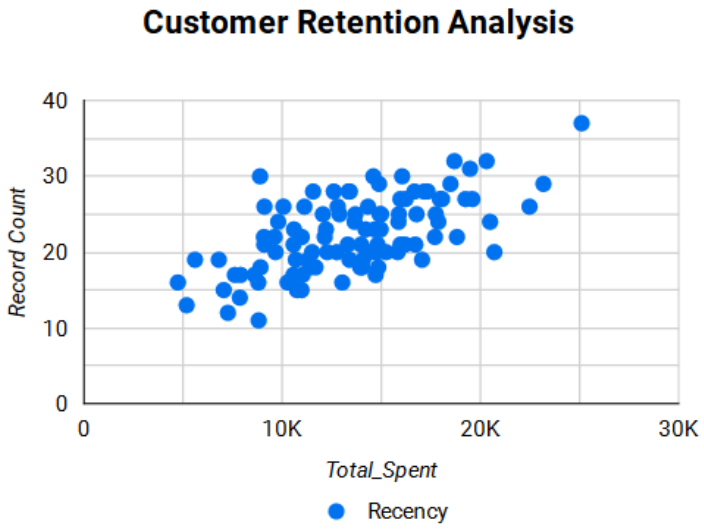
2nd Cycle: Moderate average spending, with preferences for Wines and Meats, but at lower levels compared to higher education segments.

Basic Education: Lower average spending overall, with higher relative spending on Sweets and Fish. Value-based promotions may be more effective for this segment.

Interpretation:

Marketing strategies can be tailored based on educational segments to emphasize products that align with their spending habits.

2.2.3. Customer retention analysis



Explanation:

This scatter plot visualizes the relationship between recency (how recently a customer made a purchase) and the total amount spent by each customer.

Insights:

Customers with high recency scores (far right on the x-axis) and low total spending (low

on the y-axis) are at a higher risk of churn. These customers have not made a purchase recently and have a low total spending history.

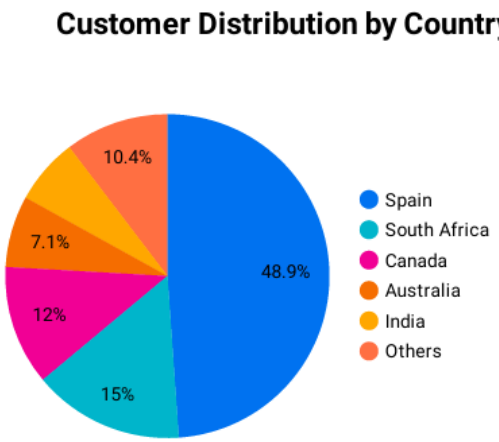
Customers with high total spending (high on the y-axis) but also high recency scores represent valuable opportunities for re-engagement campaigns.

Customers with low recency scores (far left on the x-axis) and high total spending (high on the y-axis) are loyal customers who are actively making purchases.

Interpretation:

Proactive retention strategies can be developed by focusing on customers who have not purchased recently but have a history of high spending. Personalized offers or reactivation campaigns can be designed to bring these customers back. Additionally, maintaining engagement with loyal customers through consistent communication and rewards can ensure their continued patronage.

2.2.4. Customer distribution by country



Explanation:

This pie chart shows the distribution of customers across different countries. Each slice represents a country, with the size of the slice indicating the proportion of customers from that country.

Insights:

Spain represents the largest market, accounting for nearly half of the customer base: 48.9%. This indicates a high market share and potentially strong brand presence in Spain. The significant customer base in Spain suggests that localized marketing strategies tailored to Spanish preferences and cultural norms could be highly effective.

South Africa, Canada, and Australia are also notable markets, each contributing over 10% to the customer base. These regions should also be considered for targeted marketing efforts.

India represents a notable portion of the customer base, indicating potential for growth. Marketing strategies that cater to Indian consumer behavior and preferences could help expand market share in this region.

Interpretation:

Marketing strategies should be tailored to the cultural and regional preferences of major markets. Localization of content and targeted regional promotions can enhance campaign effectiveness.

Given Spain's significant customer base, efforts to deepen market penetration there could be particularly fruitful. Meanwhile, secondary markets like South Africa, Canada, and Australia should also receive focused marketing efforts to maintain and grow their customer bases.

2.2.5. Customer Lifetime Value (CLV)

Customer Lifetime Value Heatmap

	Income	Marital_Status	CLV ▾
1.	90638	Single	5,050
2.	83844	Together	4,722
3.	94384	Together	4,604
4.	67546	Married	4,252
5.	86857	Single	4,228
6.	98777	Single	4,016
7.		Together	3,996
8.	87771	Together	3,914

1 - 100 / 2008 < >

Explanation:

This heatmap visualizes the Customer Lifetime Value (CLV) across different income levels and marital statuses. Each cell represents a combination of income and marital status, with the color intensity indicating the CLV.

Insights:

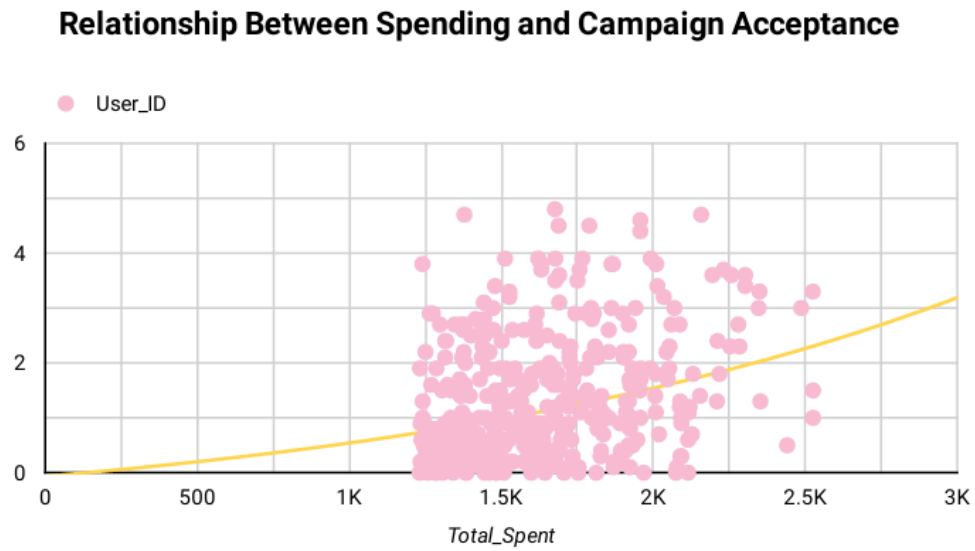
Married customers and those with higher incomes generally show higher CLV, indicating that these segments are more valuable over the long term.

High CLV segments should be targeted with loyalty programs and personalized marketing to enhance customer retention and lifetime value.

Interpretation:

By identifying high CLV segments, resources can be more effectively allocated to retain these valuable customers. Tailored strategies for high-income, married customers can increase engagement and spending.

2.2.6. Spending and Campaign Acceptance



Explanation:

This plot shows the relationship between total spending and the number of campaigns accepted by customers. Each point represents a customer, with the position on the x-axis indicating total spending and the position on the y-axis indicating the total number of accepted campaigns.

Insights:

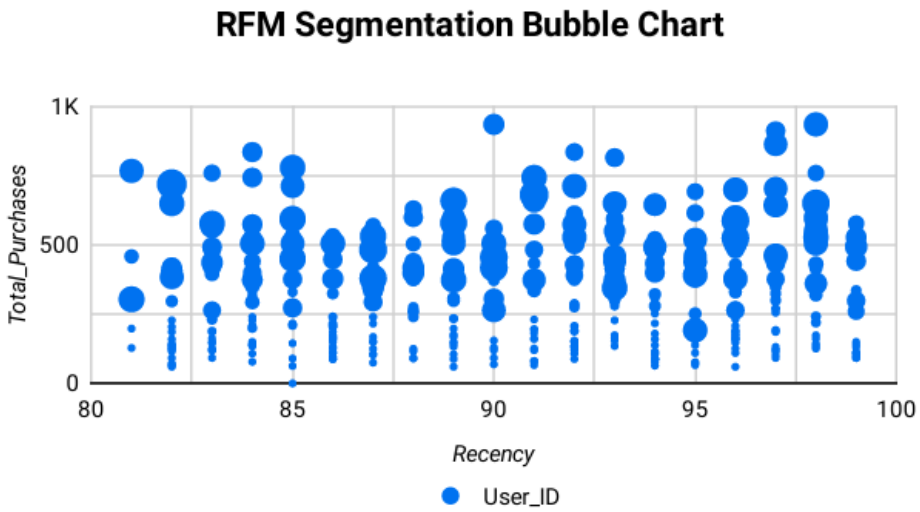
There is a positive correlation between total spending and campaign acceptance, suggesting that customers who spend more are also more likely to accept marketing campaigns.

High spenders who have accepted multiple campaigns are prime targets for future marketing efforts, as they are more engaged and responsive.

Interpretation:

Focusing marketing efforts on high spenders who have accepted multiple campaigns can lead to increased effectiveness and ROI. Personalized campaigns for these segments can further boost engagement.

2.2.7. RFM Segmentation



Explanation:

Visualizes the RFM (Recency, Frequency, and Monetary Value) segmentation of customers. Each bubble represents a customer segment, with the position on the x-axis indicating recency, the position on the y-axis indicating frequency, and the size of the bubble representing monetary value.

Insights:

Champions: Customers in the top-left quadrant (high frequency, recent purchases, high monetary value) are ideal for loyalty programs and exclusive offers to maintain their engagement.

Loyal customers: Customers with high frequency and high monetary value but varying recency scores can be nurtured with regular communications and personalized recommendations.

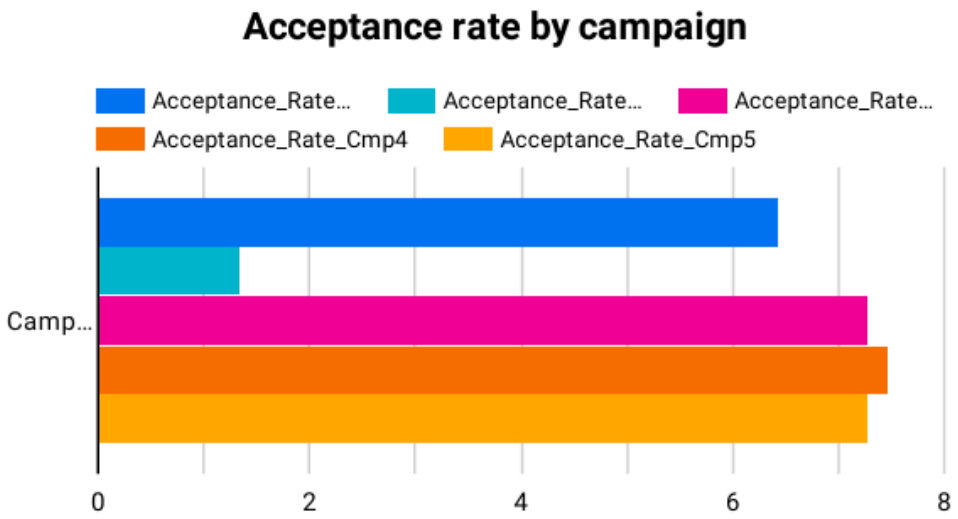
At-risk customers: Those in the bottom-right quadrant (low frequency, low recency, low monetary value) need reactivation campaigns, possibly through special discounts or re-engagement emails.

Promising customers: Customers in the top-right quadrant (high monetary value, recent purchases, low frequency) show potential for increased engagement through targeted promotions and incentives to increase purchase frequency.

Interpretation:

By understanding the different RFM segments, businesses can create tailored marketing strategies to address the specific needs and behaviors of each customer segment. This can improve customer retention, increase customer lifetime value, and enhance overall marketing effectiveness.

2.2.8. Acceptance rates comparisons



Explanation:

This bar chart displays the acceptance rates of different marketing campaigns (Acceptance_Rate_Cmp1 to Acceptance_Rate_Cmp5). Each bar represents a campaign, and the height of the bar indicates the acceptance rate for that campaign.

Insights:

Campaign 4 has the highest acceptance rate, indicating it was the most effective among the customers.

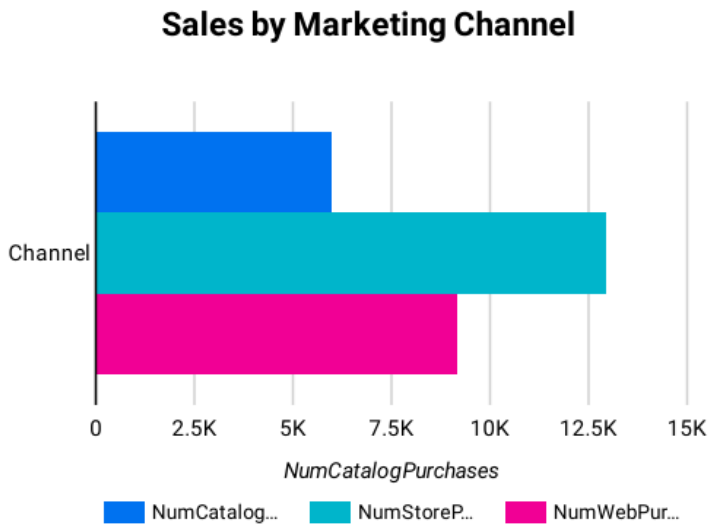
Campaign 2 has the lowest acceptance rate and is much lower than the other campaigns. It indicates campaign 2 was the least effective.

There is a notable difference in adoption rates between campaign 2 and the other campaigns, highlighting the need to analyze the specific factors that caused campaign 2 to fail so that we can avoid repeating those mistakes in future campaigns.

Interpretation:

Understanding which campaigns resonate most with customers allows businesses to refine their marketing strategies.

2.2.9. Sales in different marketing channels



Explanation:

This bar chart shows the number of purchases made through different sales channels: Catalog, Store, and Web.

Insights:

The store channel has the highest number of purchases, indicating it is the most effective sales channel. Next, the web is the second most effective channel. Finally, catalog is the least effective channel with only about half the sales volume of the store channel.

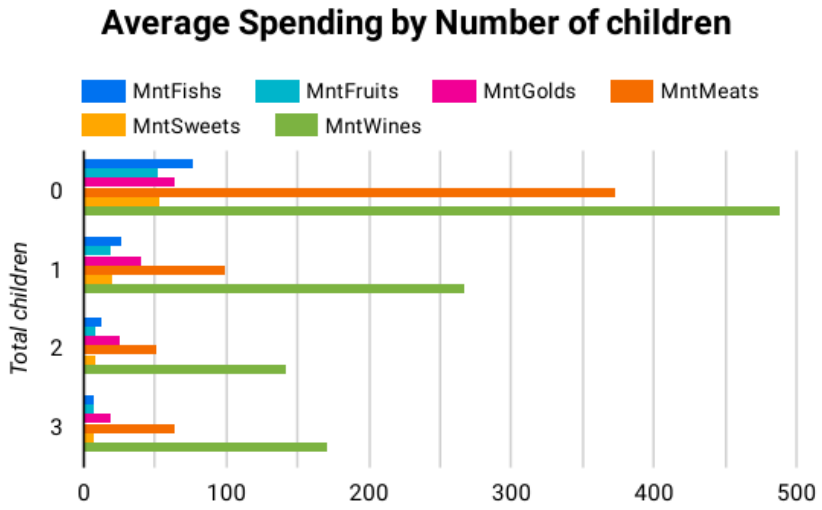
The preference for in-store purchases suggests that customers value the physical shopping experience or may prefer to see and feel products before buying.

The substantial number of web purchases highlights the importance of having a strong online presence and user-friendly e-commerce platform.

Interpretation:

By understanding the effectiveness of different sales channels, businesses can allocate resources more efficiently and develop strategies to optimize each channel. Enhancing the strengths of the most effective channels while addressing the weaknesses of less effective ones can lead to increased sales and customer satisfaction.

2.2.10. Average spending by number of children



Explanation:

This chart shows the average spending across various product categories segmented by the number of children in the household.

Insights:

Households with 2 Children: Show the highest average spending across most categories. This suggests that, on a per-family basis, households with 2 children tend to spend more on average.

Households with 1 Child: Show substantial average spending, indicating they are also significant contributors on a per-family basis.

Households with No Children: Have moderate average spending, lower than households with children.

Households with 3 Children: Show the least average spending, which could be influenced by budget constraints or different purchasing habits.

Interpretation:

By understanding how the number of children in a household affects average spending, businesses can tailor their marketing strategies to better meet the needs of different family structures.

2.3. Recommendations for strategy development

Customer retention and engagement

Prioritize high-value customers by developing personalized loyalty programs and exclusive offers to maintain their engagement and prevent potential revenue loss. Focus on re-engaging customers with high total spending but high recency scores through targeted campaigns and personalized offers. Maintain engagement with loyal customers through consistent communication and rewards.

Tailored marketing strategies

Customize marketing campaigns based on customer segmentation, particularly by education level and family structure. For example, promote premium products to higher-educated customers and create family-sized product bundles for households with children. Tailor offers to meet the specific needs and preferences of different segments to maximize impact.

Campaign optimization

Analyze and replicate the success factors of high acceptance rate campaigns to improve future campaign effectiveness. Investigate and address the shortcomings of lower-performing campaigns to refine strategies and avoid past mistakes. Focus marketing efforts on high spenders who have shown a propensity to accept multiple campaigns for better ROI.

Channel optimization

Enhance the in-store shopping experience through personalized services and exclusive promotions. Strengthen the online shopping experience by ensuring a seamless, secure checkout process and offering online-exclusive deals. Explore innovative strategies to boost the effectiveness of the catalog channel.

Geographic focus

Develop localized marketing strategies tailored to the preferences and cultural norms of major markets such as Spain. Target secondary markets like South Africa, Canada, and Australia with focused marketing efforts to maintain and grow their customer bases. Expand market share in emerging markets like India by aligning marketing strategies with local consumer behavior.

Product and pricing strategy

Promote high-spend product categories such as wines and meats, especially among segments with high average spending. Consider dynamic pricing models to balance volume and profit margins effectively.

Online and social media presence

Strengthen the company's online and social media presence to build credibility and engage with a broader audience. Regular updates, interactive content, and customer engagement on social media platforms can enhance brand visibility and attract new customers.

3. Machine Learning Model

3.1. Predict Customer's Response - Decision Tree Model

3.1.1. Objective

The goal was to build a model to predict which customers are likely to respond positively to future marketing campaigns. This helps in targeting high-potential customers, thereby reducing marketing costs.

3.1.2. Methodology

Data Collection: Data was collected from the fact_MarketingCampaignResponse table in BigQuery.

Data Preprocessing: The dataset was processed to handle missing values, convert data types, and standardize features.

Model Selection: A Decision Tree Classifier was chosen for its interpretability and effectiveness in handling categorical data.

Model Training and Evaluation: The model was trained on a split dataset (80% training, 20% testing) and evaluated for accuracy and classification metrics.

```
# lib
!pip install --upgrade google-cloud-bigquery
!pip install --upgrade google-auth
!pip install --upgrade google-auth-oauthlib
!pip install scikit-learn pandas

# use_auth
from google.colab import auth
auth.authenticate_user()

# lib
from google.cloud import bigquery
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report
from google.cloud.exceptions import NotFound

# client BigQuery
client = bigquery.Client(project='datawarehouse-422504')

# Query from BigQuery
query = """
SELECT User_ID, Recency, Complain_Times, NumWebVisitsMonth, Total_Purchases,
Total_Spent, Overall_Accept_Campaign, NumDealsPurchases, Response
FROM `datawarehouse-422504.OLAP.fact_MarketingCampaignResponse`
"""
df = client.query(query).to_dataframe()

# Data processing
X = df[['User_ID', 'Recency', 'Complain_Times', 'NumWebVisitsMonth', 'Total_Purchases',
```

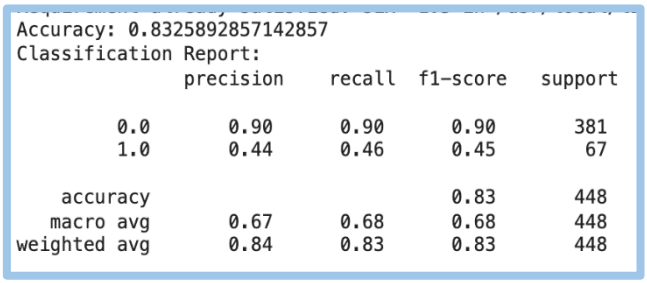
```
'Total_Spent', 'Overall_Accept_Campaign', 'NumDealsPurchases']]
y = df['Response']

# split train/test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# save User_ID
X_train_user_id = X_train['User_ID']
X_test_user_id = X_test['User_ID']

# user id not included in train/test
X_train = X_train.drop(columns=['User_ID'])
X_test = X_test.drop(columns=['User_ID'])

# data standard
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Decision Trees
tree_model = DecisionTreeClassifier(random_state=42)
tree_model.fit(X_train_scaled, y_train)

# predict on test
y_pred = tree_model.predict(X_test_scaled)

# Evaluate
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
print("Classification Report:\n", classification_report(y_test, y_pred))

# Data frame for train
df_results_train = pd.DataFrame({
  'User_ID': X_train_user_id.values,
  'Recency': X_train['Recency'].values,
  'Complain_Times': X_train['Complain_Times'].values,
```

```
  'NumWebVisitsMonth': X_train['NumWebVisitsMonth'].values,
  'Total_Purchases': X_train['Total_Purchases'].values,
  'Total_Spent': X_train['Total_Spent'].values,
  'Overall_Accept_Campaign': X_train['Overall_Accept_Campaign'].values,
  'NumDealsPurchases': X_train['NumDealsPurchases'].values,
  'Actual_Response': y_train.values,
  'Predicted_Response': tree_model.predict(X_train_scaled).astype(int)
})

# DataFrame for test
df_results_test = pd.DataFrame({
  'User_ID': X_test_user_id.values,
  'Recency': X_test['Recency'].values,
  'Complain_Times': X_test['Complain_Times'].values,
  'NumWebVisitsMonth': X_test['NumWebVisitsMonth'].values,
  'Total_Purchases': X_test['Total_Purchases'].values,
  'Total_Spent': X_test['Total_Spent'].values,
  'Overall_Accept_Campaign': X_test['Overall_Accept_Campaign'].values,
  'NumDealsPurchases': X_test['NumDealsPurchases'].values,
  'Actual_Response': y_test.values,
  'Predicted_Response': y_pred.astype(int)
})

# Combine train test sets
df_results = pd.concat([df_results_train, df_results_test])



table_id = "datawarehouse-422504.OLAP.Predicted_Responses"
try:
  client.get_table(table_id)
  print("Existed. Replaced.")
  job_config =
bigquery.LoadJobConfig(write_disposition=bigquery.WriteDisposition.WRITE_TRUNCATE)
except NotFound:
  print("Create new table")
  job_config =
```

```
bigquery.LoadJobConfig(write_disposition=bigquery.WriteDisposition.WRITE_APPEND)

job = client.load_table_from_dataframe(df_results, table_id, job_config=job_config)
job.result()

print("Successful")
```

3.1.3. Results

3.1.3.1. Classification Report:



```
Accuracy: 0.8325892857142857
Classification Report:
              precision    recall  f1-score   support

         0.0       0.90      0.90      0.90       381
         1.0       0.44      0.46      0.45        67

    accuracy                           0.83       448
   macro avg       0.67      0.68      0.68       448
weighted avg       0.84      0.83      0.83       448
```

Accuracy: The overall accuracy of the model is 83%, indicating that the model correctly predicted the customer response for 83% of the cases in the test set.

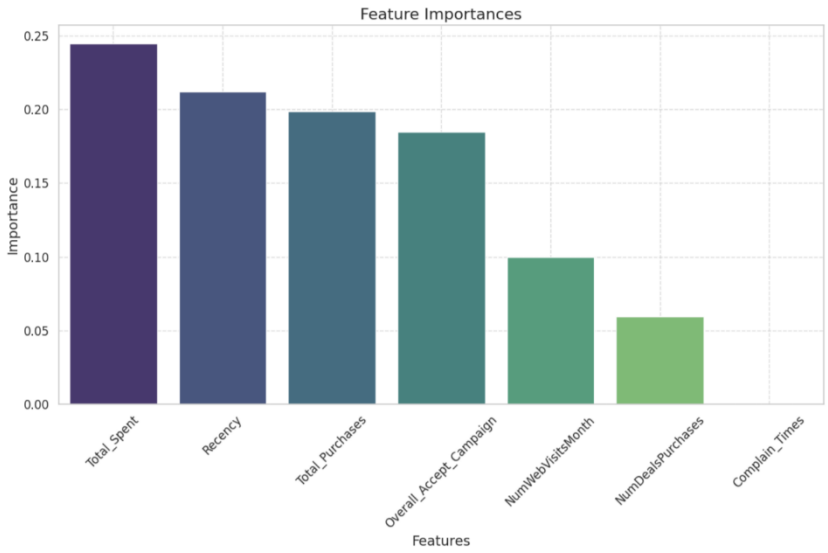
Macro Average: Precision: 0.67, Recall: 0.68, F1-Score: 0.68. The unweighted mean of the precision, recall, and F1-score across both classes. This gives equal importance to each class.

Weighted Average: Precision: 0.84, Recall: 0.83, F1-Score: 0.83. The mean of the precision, recall, and F1-score across both classes, weighted by the number of true instances in each class. This gives more importance to the class with more instances.

3.1.3.2. Feature Importance

The graph shows the importance of each feature in predicting customer responses to marketing campaigns using a Decision Tree model.

Total Spent: The amount of money a customer has spent is the most critical feature in predicting their response to marketing campaigns. Customers who have spent more



in the past are more likely to respond positively to future marketing efforts.

Recency, or how recently a customer interacted with the company, is the second most important feature. Customers who interacted more recently are more likely to respond positively.

Total Purchases: The total number of purchases made by a customer is also a significant predictor. Frequent buyers are more likely to respond positively to marketing campaigns.

Overall Accept Campaign: The overall acceptance of past campaigns by the customer is another crucial feature. If a customer has accepted previous campaigns, they are more likely to respond positively to new ones.

NumWebVisitsMonth: The number of web visits in the past month is a moderately important feature. More frequent website visits indicate higher engagement and a higher likelihood of responding positively.

NumDealsPurchases: The number of deals purchased by the customer has some influence, but it is less critical than the above features. Customers who have bought more deals might have a slightly higher tendency to respond positively.

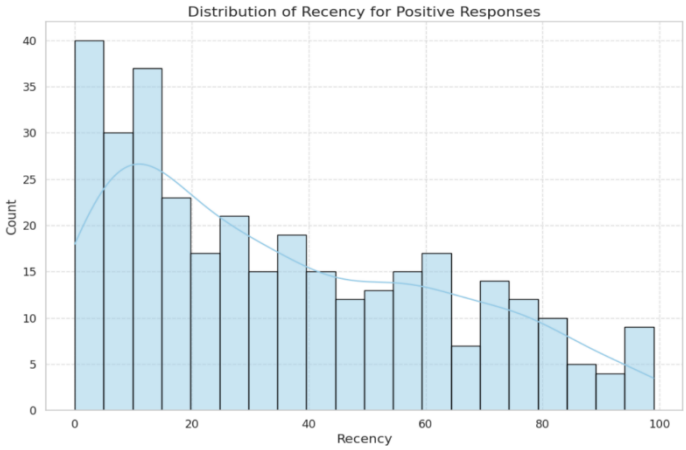
ComplainTimes: The number of complaints filed by the customer is the least important feature in this model. It suggests that the frequency of complaints has a minimal impact on predicting positive responses to marketing campaigns.

3.1.3.3. Distribution of Recency for Positive Responses Graph

The "Recency" metric measures how recently customers have interacted with the company's marketing campaigns.

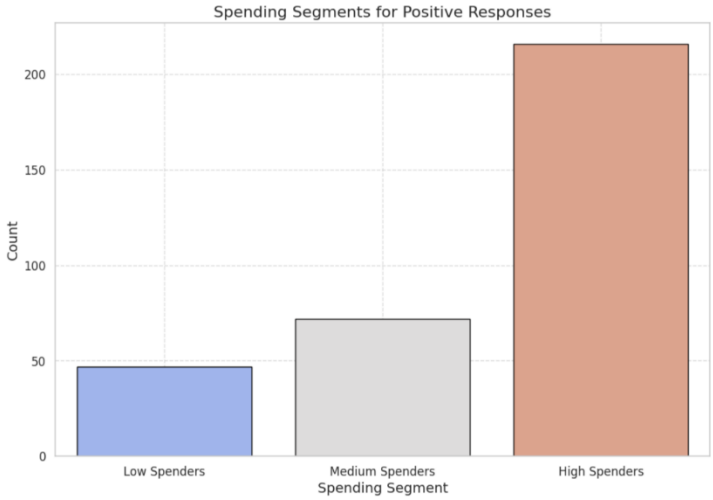
From the graph, we can observe that customers who responded positively to the marketing campaigns generally have lower recency values. This indicates that more recent interactions (lower recency values) are associated with higher positive response rates. As the recency increases, the number of positive responses decreases. The peak of the distribution is at the lower end of the recency scale, showing that customers who have interacted with the company more recently are more likely to respond positively to marketing efforts.

The business implication is that marketing efforts should focus on customers who have interacted with the company recently, as they are more likely to respond positively.



3.1.3.4. Spending Segments for Positive Responses Graph

This graph segments customers based on their total spending and shows the count of positive responses within each segment.



The spending segments are defined as:

Low Spenders: Total spending less than $100

Medium Spenders: Total spending between $100 and $500

High Spenders: Total spending greater than $500

The graph clearly shows that "High Spenders" have the highest number of positive responses, followed by "Medium Spenders," and then "Low Spenders." This suggests that customers who spend more money are more likely to respond positively to marketing campaigns.

The business implication is that high spenders should be a primary target for marketing campaigns, as they are more likely to respond positively and have a higher lifetime value.

3.2 Customers Segmentation - Clustering Model - KMeans Algorithms

3.2.1. Objective:

The aim of this section is to identify customer segments based on their likelihood to respond positively to marketing campaigns. By clustering customers into distinct groups, the business can tailor marketing strategies more effectively and reduce costs by targeting the most promising segments.

3.2.2. Methodology

Data Collection: Data was collected from the fact_MarketingCampaignResponse table in BigQuery. This table includes relevant customer data such as recency of purchases, frequency of complaints, number of web visits, total purchases, total spending, acceptance of campaigns, and number of deal purchases.

Data Preprocessing: The collected data was preprocessed to handle any inconsistencies and standardize features. This ensures that all variables contribute equally to the clustering process.

Model Selection: KMeans clustering was chosen for its simplicity and effectiveness in segmenting customers based on multiple features. The optimal number of clusters was determined using the Elbow method.

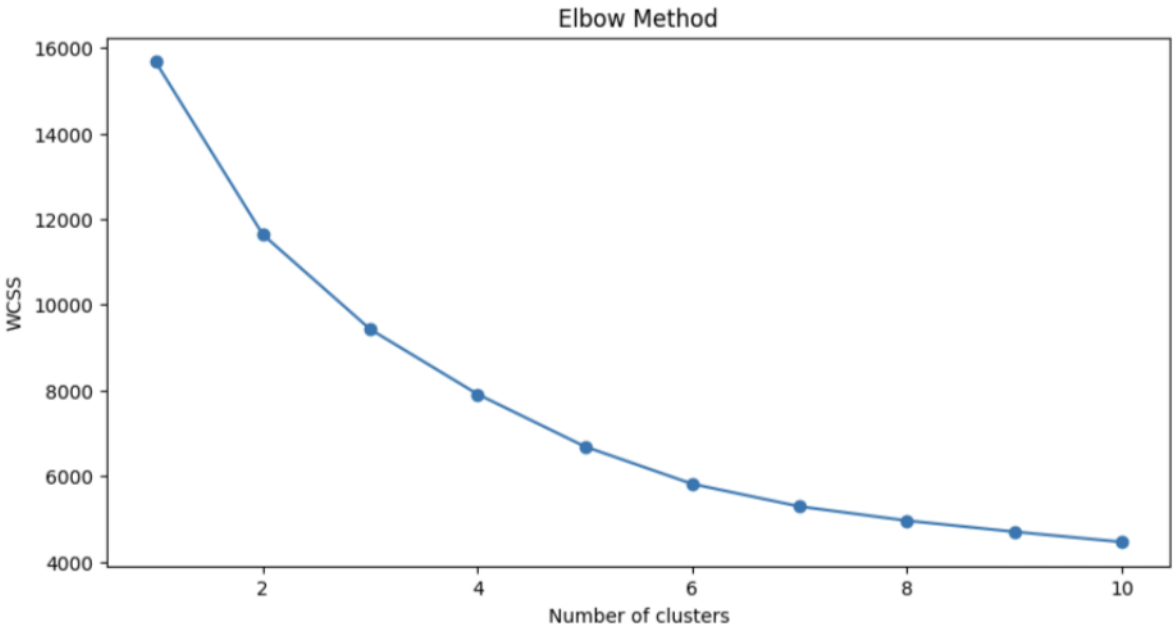
Model Training and Evaluation: The KMeans model was trained on the preprocessed data, and the resulting clusters were analyzed to gain insights into customer segments. The summary statistics of each cluster provide a clear understanding of the characteristics of each group.

```
# Processing data
X = df[['Recency', 'Complain_Times', 'NumWebVisitsMonth', 'Total_Purchases', 'Total_Spent',
'Overall_Accept_Campaign', 'NumDealsPurchases']]

# data standard
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# optimize cluster_num ussing elbow
wcss = []
for i in range(1, 11):
  kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=10, random_state=42)
  kmeans.fit(X_scaled)
  wcss.append(kmeans.inertia_)

# elbow_graph
plt.figure(figsize=(10, 5))
```

```
plt.plot(range(1, 11), wcss, marker='o')
plt.title('Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()

# train model KMeans with optimized cluster_num
kmeans = KMeans(n_clusters=3, init='k-means++', max_iter=300, n_init=10, random_state=42)
y_kmeans = kmeans.fit_predict(X_scaled)

# add in label to DataFrame
df['Cluster'] = y_kmeans

# Cluster_summary
cluster_summary = df.groupby('Cluster').mean()
print("Cluster Summary:\n", cluster_summary)

# Push result back to BigQuery
# Table existed or not
table_id = "datawarehouse-422504.OLAP.Clustered_Customers"

try:
  client.get_table(table_id)
  print("Existed. Replaced.")
  job_config =
bigquery.LoadJobConfig(write_disposition=bigquery.WriteDisposition.WRITE_TRUNCATE)
except:
  print("Not exist. Create new.")
  job_config =
bigquery.LoadJobConfig(write_disposition=bigquery.WriteDisposition.WRITE_APPEND)


job = client.load_table_from_dataframe(df, table_id, job_config=job_config)
job.result()

print("Successfull")
```

3.2.3 Results:

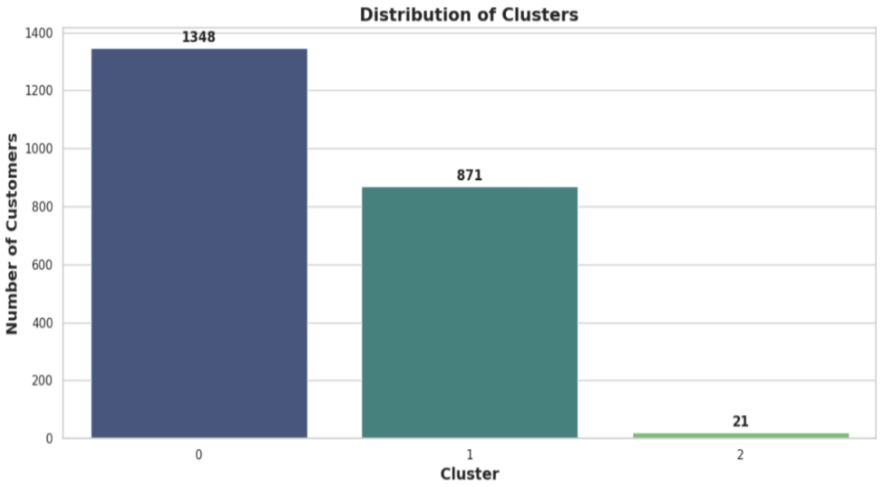
3.2.3.1. Elbow Method Interpretation:



The Elbow Method plot shows the within-cluster sum of squares (WCSS) for different numbers of clusters (from 1 to 10). The WCSS decreases as the number of clusters increases. The "elbow" point is typically where the WCSS starts to decrease more slowly, indicating an optimal number of clusters. In this case, the elbow point appears to be at 3 clusters, as the WCSS reduction rate starts to slow down after this point.

3.2.3.2. Cluster Summary:

```
Cluster Summary:
              User_ID    Recency  Complain_Times  NumWebVisitsMonth  \
Cluster
0         5575.038576  48.777448             0.0           6.432493
1         5591.399541  49.528129             0.0           3.577497
2         6722.714286  53.047619             1.0           5.809524

          Total_Purchases  Total_Spent  Overall_Accept_Campaign  \
Cluster
0              200.098665   200.301187                  0.10089
1              509.570608  1238.894374                   0.6062
2              278.904762   376.428571                 0.142857

          NumDealsPurchases  Response
Cluster
0                  2.606825  0.097181
1                  1.888634  0.229621
2                  2.333333  0.142857
```



*** Cluster 0:**

Proportion: This is the largest cluster (1348 customers), containing the majority of customers (approximately 60% of the total customer base).

Characteristics: Based on previous cluster summary insights, customers in Cluster 0 have moderate recency, high web visits, lower total purchases and spending, the highest number of deal purchases, and the lowest campaign acceptance rate and response rate. These customers might be engaging more frequently with smaller transactions or promotional deals.

*** Cluster 1:**

Proportion: This cluster represents the second largest group of customers (871 customers, approximately 39% of the total customer base).

Characteristics: Customers in Cluster 1 exhibit similar recency as Cluster 0 but have fewer web visits, the highest total purchases and spending, and the highest acceptance rate of campaigns. Their response rate is also the highest. This cluster consists of high-value customers who are more responsive to marketing campaigns and engage with the brand more significantly.

*** Cluster 2:**

Proportion: This is the smallest cluster, containing a very small portion of the customer base (21 customers, approximately 1%).

Characteristics: Customers in Cluster 2 have the longest recency, the highest number of complaints, moderate web visits, total purchases, and spending. Their campaign acceptance rate is slightly higher than Cluster 0 but much lower than Cluster 1, with a moderate response rate. This cluster may represent a specific group of customers with unique behaviors, such as frequent complaints, who might need targeted interventions to improve satisfaction and engagement.

Business Implications

*** Marketing Strategies:**

Cluster 0 (Low Spenders, High Engagement): These customers are frequent but low-value spenders. Marketing strategies could focus on increasing the average transaction value through upselling and cross-selling. Since they respond poorly to campaigns, experimenting with different types of offers or personalized communication may help improve response rates.

Cluster 1 (High Spenders, High Response): This group should be the primary target for marketing campaigns. They are high-value customers with a high response rate. Personalized and premium offers, loyalty programs, and retention strategies should be prioritized for this cluster to maintain and increase their engagement.

Cluster 2 (High Complaints, Moderate Response): This small but unique cluster requires special attention. Focus on addressing their complaints and improving their satisfaction through customer service improvements. Once their issues are resolved, these customers could potentially become more responsive to campaigns.

*** Resource Allocation:**

Allocate more resources and budget to Cluster 1 due to their high spending and responsiveness. Ensure that marketing efforts for this cluster are highly personalized and data-driven.

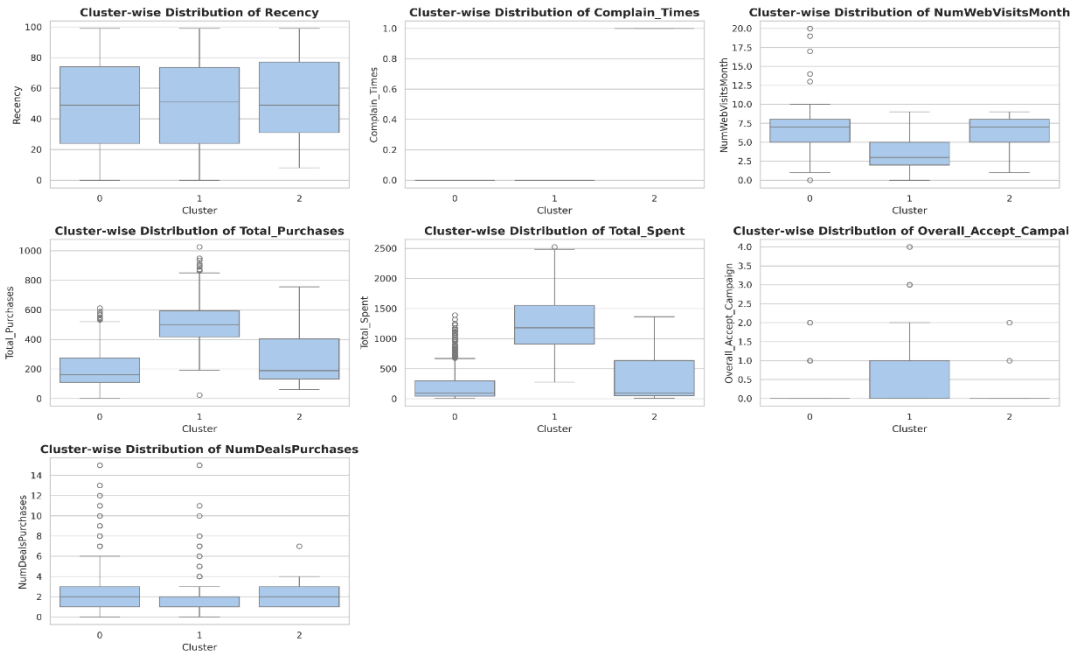
Dedicate efforts to understanding and resolving the issues faced by Cluster 2 customers. Improving their satisfaction could convert them into more valuable customers.

For Cluster 0, consider low-cost, broad-reach marketing strategies and continuously experiment to identify what types of offers or campaigns could resonate more with them.

*** Customer Engagement:**

Enhance engagement strategies for each cluster based on their characteristics. For example, Cluster 1 may appreciate exclusive offers and early access to sales, while Cluster 0 may respond better to regular promotional communications and loyalty rewards.

3.2.3.3. Box Plot of each Feature



Features Explaination:

*** Cluster-wise Distribution of Recency**

Cluster 0: Median recency is around 48, with a wide interquartile range (IQR) indicating diverse recency values among customers.

Cluster 1: Similar to Cluster 0, with a median recency around 49.

Cluster 2: Median recency is slightly higher at around 53, suggesting these customers have been less recent in their purchases compared to Clusters 0 and 1.

*** Cluster-wise Distribution of Complain_Times**

Cluster 0 and Cluster 1: No complaints (values at 0).

Cluster 2: Higher number of complaints, indicating these customers have had issues with the service or product.

*** Cluster-wise Distribution of NumWebVisitsMonth**

Cluster 0: Higher median number of web visits per month, indicating frequent online interactions.

Cluster 1: Lower median number of web visits per month, suggesting less frequent online engagement.

Cluster 2: Similar to Cluster 0 but slightly lower, still indicating frequent online interactions.

*** Cluster-wise Distribution of Total_Purchases**

Cluster 0: Lower total purchases compared to other clusters, indicating lower engagement in buying activities.

Cluster 1: Highest total purchases, indicating this cluster consists of the most active buyers.

Cluster 2: Moderate total purchases, higher than Cluster 0 but lower than Cluster 1.

*** Cluster-wise Distribution of Total_Spent**

Cluster 0: Lowest total spending among the clusters.

Cluster 1: Highest total spending, correlating with their high total purchases.

Cluster 2: Moderate spending, more than Cluster 0 but less than Cluster 1.

*** Cluster-wise Distribution of Overall_Accept_Campaign**

Cluster 0: Low acceptance of campaigns.

Cluster 1: Highest acceptance rate of campaigns, indicating responsiveness to marketing efforts.

Cluster 2: Moderate acceptance rate, higher than Cluster 0 but lower than Cluster 1.

*** Cluster-wise Distribution of NumDealsPurchases**

Cluster 0: Highest number of deal purchases, suggesting these customers are deal-sensitive.

Cluster 1: Moderate number of deal purchases.

Cluster 2: Similar to Cluster 1 but slightly higher.

Business Implications:

*** Cluster 0:**

Characteristics: Lower total purchases and spending, frequent web visits, high deal purchases, and low campaign acceptance.

Strategy: Focus on upselling and cross-selling to increase their total purchases and spending. Experiment with different types of offers to increase campaign acceptance.

*** Cluster 1:**

Characteristics: Highest total purchases and spending, lower web visits, and highest campaign acceptance.

Strategy: Prioritize this cluster for marketing campaigns due to their high engagement and responsiveness. Offer personalized and premium offers to maintain and increase their loyalty and spending.

*** Cluster 2:**

Characteristics: Higher recency, more complaints, frequent web visits, moderate purchases, and spending.

Strategy: Address customer complaints and improve service quality to enhance their satisfaction and engagement. Tailor marketing efforts to convert their frequent web visits into purchases.

V. Results

1. Key Findings

Based on the analysis of the iFood dataset and the implementation of our data warehouse, several key findings have emerged:

1.1. Customer Segmentation and Behavior

High-Value Customers: Customers with higher total spending and frequent purchases are more responsive to marketing campaigns. These customers should be prioritized for targeted marketing efforts.

Geographical Insights: Spain and South Africa are key markets with high campaign acceptance rates and significant spending across various product categories. Spain, in particular, has shown the highest market share and acceptance rates.

Segment-Specific Preferences: Customers with higher education levels, such as PhD holders, tend to spend more on premium products like wines and meats. Married customers with higher incomes exhibit higher customer lifetime value (CLV).

1.2. Campaign Effectiveness

Most Effective Campaigns: Campaign 4 had the highest acceptance rate, indicating it resonated well with customers. In contrast, Campaign 2 had the lowest acceptance rate, suggesting it needs a strategic review and potential revamp.

Sales Channels: In-store purchases remain the most effective sales channel, followed by online purchases. Catalog purchases are the least effective but still significant in specific markets like Canada.

1.3. Predictive Analytics

Feature Importance: Total spending, recency of purchases, and overall campaign acceptance are the most critical features in predicting positive responses to marketing campaigns.

Customer Clusters: Using KMeans clustering, three distinct customer segments were identified: frequent low spenders (Cluster 0), high-value customers (Cluster 1), and customers with high complaints but moderate spending (Cluster 2).

1.4. Behavioral Insights

Web Visit Frequency: High web visit frequency does not necessarily translate to high spending or campaign acceptance. Personalized campaigns could help convert frequent visitors into active spenders.

Family Influence: The number of children in a household does not significantly impact spending or campaign responsiveness compared to other factors like marital status and education level.

2. Recommendations

The following recommendations are proposed to optimize marketing strategies and improve customer engagement:

2.1. Focus on High-Value Segments

Targeted Campaigns: Prioritize marketing efforts on high-value customers (Cluster 1) with personalized offers and loyalty programs to enhance their engagement and spending.

Geographical Targeting: Develop localized marketing strategies for high-potential markets like Spain and South Africa, leveraging regional preferences and cultural norms.

2.2. Enhance Campaign Strategies

Campaign Optimization: Analyze the success factors of Campaign 4 and replicate these strategies in future campaigns. Revamp Campaign 2 by addressing its shortcomings and testing new approaches.

Channel Strategy: Strengthen the online shopping experience and in-store promotions. Explore innovative strategies to boost catalog sales, particularly in markets where it remains effective.

2.3. Leverage Predictive Analytics

Personalized Marketing: Utilize predictive models to identify customers likely to respond positively to campaigns and tailor marketing efforts accordingly.

Customer Reactivation: Focus on re-engaging customers with high recency scores and high spending history through targeted reactivation campaigns.

2.4. Improve Customer Satisfaction

Address Complaints: Pay special attention to customers with frequent complaints (Cluster 2) to resolve issues and improve satisfaction, potentially converting them into loyal customers.

Engagement Strategies: Develop strategies to increase engagement with frequent web visitors, such as personalized content and targeted offers.

Data-Driven Decision Making:

2.5. Continuous Monitoring

Regularly monitor and analyze customer behavior and campaign performance to make data-driven decisions and adjustments.

Feedback Loop: Implement a feedback loop to continuously gather insights from campaign performance and customer interactions to refine strategies.

VI. Conclusion

The iFood dataset project aimed to enhance marketing strategies and customer engagement through the development of a comprehensive data warehouse and advanced analytics. The project was carried out in a structured manner, from understanding the dataset and essential tools to implementing a robust ETL process and performing business analytics.

The project's results underscore the importance of a data-driven approach in modern marketing strategies. By understanding customer behavior and preferences, businesses can tailor their marketing efforts to maximize engagement and profitability. The implementation of a robust data warehouse and advanced analytics tools enabled the project to achieve its objectives and provide a clear roadmap for enhancing marketing effectiveness.

In conclusion, the iFood dataset project successfully demonstrated the power of data warehousing and business analytics in transforming raw data into actionable insights. The project's comprehensive approach, from data preparation to predictive modeling, has laid a strong foundation for future data-driven marketing initiatives. By continuously refining strategies based on data insights, the organization can achieve sustained growth and improved customer satisfaction.

VII. References

[1] Nailson (2020) *GitHub - nailson/ifood-data-business-analyst-test: iFood Brain team data challenge for Data Analysts role*. https://github.com/nailson/ifood-data-business-analyst-test/tree/master.

[2] Studocu *I Food Data Analyst Case - iFood CRM Data Analyst Case iFood is the lead food delivery app in - Studocu*. https://www.studocu.com/vn/document/truong-dai-hoc-ngoai-thuong/data-science/i-food-data-analyst-case/74070510.

[3] *ifood dataset* (2022). https://www.kaggle.com/datasets/prabhrajsingh/ifood-dataset.

[4] Google Colab - Colab.google. Colab + BigQuery—Perfect Together. https://colab.google/articles/bq.

[5] Park, S.J. (2024) 'Google CoLab 101: Connecting to & Querying BigQuery Data,' *Medium*, 11 January. https://songjoyce.medium.com/google-colab-101-connecting-to-bigquery-3a2481706907.

[6] *Google Cloud BigQuery Operators — apache-airflow-providers-google Documentation* (no date). https://airflow.apache.org/docs/apache-airflow-providers-google/stable/operators/cloud/bigquery.html.

[7] Nassirova, E. (2024) *Looker Studio (Google Data Studio) tutorial: Dashboard for Beginners*. https://blog.coupler.io/google-data-studio-tutorial-for-beginners/.