Data engineering

**Target data pipeline**

Business Case

Scaler

Topic: Data engineering

**Mindset**

1. Evaluation will be kept lenient, so make sure you attempt this case study.
2. It is understandable that you might struggle with getting started on this. Just brainstorm, discuss with peers, or get help from TAs.
3. There is no right or wrong answer. We have to become comfortable with dealing with uncertainty in business. This is exactly the skill we want to develop.

**Problem Statement**

Target Corporation is an American retail corporation headquartered in Minneapolis, Minnesota. It is the eighth largest retailer in the United States, and a component of the S&P 500 Index. As of 2023, Target operates 1,948 stores throughout the United States, and is ranked No. 32 on the 2022 Fortune 500 list of the largest U.S. corporations by total revenue.

You are working as a Data Engineer for Target,which has numerous stores across the United States. Your role is to build a robust data pipeline to process and analyze sales transaction data from these stores.Your task is to provide valuable insights to Target, such as understanding their sales across cities and states on a daily and weekly basis. Additionally, you need to provide insights on product reviews and trends to help inform business decisions. Your ultimate goal is to build a reliable and efficient data pipeline that can handle the large volume of sales transaction data generated by Target's stores.

**Dataset:**

Link: <https://drive.google.com/file/d/1WjfvOvsPT_iwxEV_odUj2U5ZG0Ha7MF0/view?usp=sharing>

**Column Profiling:**

* Id: A unique identifier
* order\_status: The status of the order, such as "delivered" or "shipped"
* order\_products\_value: The total value of the products in the order.
* order\_freight\_value: The cost of shipping the order.
* order\_items\_qty: The number of items in the order.
* order\_purchase\_timestamp: The date and time the order was placed.
* order\_aproved\_at: The date and time the order was approved by the retailer.
* order\_delivered\_customer\_date: The date and time the order was delivered to the customer.
* customer\_city: The city where the customer lives.
* customer\_state: The state where the customer lives.
* customer\_zip\_code\_prefix: The zip code prefix of the customer's address.
* review\_score: The score assigned by the customer for the purchased product(s), typically ranging from 1 (worst) to 5 (best).

**Concept Used:**

* Kafka
* Pyspark
* Hive

**How to begin:**

* Read the data from the CSV file using the pandas library and create a Kafka topic using the kafka-python library.
* Create a Kafka producer to produce data row by row to the Kafka topic and a Kafka consumer to read data from the Kafka topic using the kafka-python library. (both run parallel, consumer should start before producer ready to consume)
* Create a Spark DataFrame by appending all the data read from Kafka.
* Check the schema of the Spark DataFrame and correct the data types of columns if they are incorrect using the cast function of Spark DataFrame.
  + Remove null values if any.
* Perform exploratory data analysis (EDA) on the Spark DataFrame using Spark SQL or Spark DataFrame operations.
* EDA Questions:

1. Calculate the Mean of “order\_products\_value” , ”order\_freight\_value”
2. What is the distribution of “order\_status”
3. How many unique states are there?
4. Are there any missing values in the dataset?
5. Top 5 cities with most order
6. Percent of orders delivered/ canceled etc

* Perform data processing on the Spark DataFrame to transform, and filter the data using Spark SQL or Spark DataFrame operations.
* Questions:

1. Calculate the Total sales in each customer city
2. Correlation between order value order freight and item quantity
3. Calculate the Average Order delivery time / order approval time
4. Calculate the Average review score per Order
5. Find the top 3 cities with fastest and slowest delivery times
6. Relation between delivery time and review score (average delivery time for each review score) / Correlation between delivery time and review score

* Save the processed data to a Hive table using the saveAsTable function of the Spark DataFrame.

**Evaluation criteria**

1. Creating kafka producer (20 points)
   1. Creating kafka producer, topic
   2. Sending data row wise from the target\_data.csv
2. Creating kafka consumer (20 points)
   1. Creating the kafka consumer to get the data from the producer.
   2. Reading row wise data from the producer
   3. Add it to an empty pandas df
3. Converting from pandas to pyspark df (20 points)
   1. Creating spark context
   2. Converting from pandas to pyspark df
   3. Solving the data types and correcting the data structure
   4. Remove null values if any
4. EDA using pyspark df or using pyspark sql(10 points)
   1. Calculate the Mean of “order\_products\_value” , ”order\_freight\_value”
   2. What is the distribution of “order\_status”
   3. How many unique states are there?
   4. Are there any missing values in the dataset?
   5. Top 5 cities with most order
   6. Percent of orders delivered/ canceled etc
5. Some analysis on the df using pyspark df or pyspark sql (20 points)
   1. Calculate the Total sales in each customer city
   2. Correlation between order value order freight and item quantity
   3. Calculate the Average Order delivery time / order approval time
   4. Calculate the Average review score per Order
   5. Find the top 3 cities with fastest and slowest delivery times
   6. Relation between delivery time and review score (average delivery time for each review score) / Correlation between delivery time and review score
6. Save the processed data to a Hive table (10 points)

**Submission Process:**

* Type your insights and recommendations in the text editor.
* Convert your jupyter notebook into PDF (Save as PDF using Chrome browser’s Print command), upload it in your Google Drive (set the permission to allow public access), and paste that link in the text editor.
* Optionally, you may add images/graphs in the text editor by taking screenshots or saving matplotlib graphs using plt.savefig(...).
* After submitting, you will not be allowed to edit your submission.