



InTrend Shoes Analysis
by
Freestylers (Team 3)



Google Cloud

Persona - Alice, CMO of InTrend Shoes



Attitude:

- Alice is the Chief Marketing Officer (CMO) of InTrend Shoes.
- She works full-time, loves her job, and is currently leading a major marketing project across six European cities – Amsterdam, Brussels, Frankfurt, London, Milan and Paris.

Goals:

- A desire for InTrend Shoes to be one of the best shoe companies across Europe, offering the best quality & value for money with tailored products for youth.
- For her company to achieve and maintain an excellent Net Promoter Score (NPS).
- To offer customers specific products at different times of the year, according to their location and seasonal weather.

Pain Points:

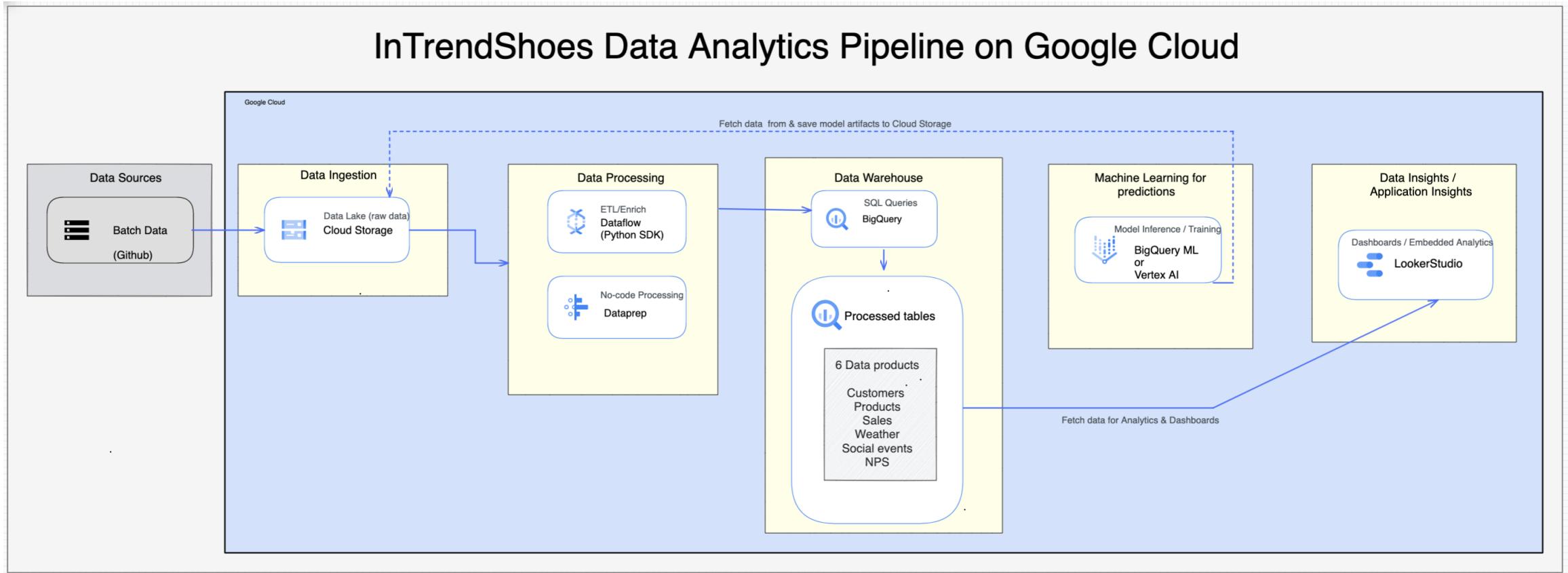
- Significant data inconsistencies (Data Quality Issues) and challenges across Europe.
- Key revenue targets recently missed, causing frustration and stress.
- Poor investment to the existing infrastructure, causing exploitation of specific data almost impossible.
- Very few ways of extracting the data consistently.
- No automated data ingestion services.
- No dashboards enabling reports on Data classification, Data lineages or metrics.
- No way of exploiting data relating to weather and location, in order to target customers with relevant products at specific times of the year.
- Very low NPS for InTrend Shoes.

Needs:

- A team of skilled data professionals in her project, who can extract the right data in the right way, across the European countries, in order to achieve the project goals.
- To identify, understand and process the necessary data relating to sales trends on specific products.
- To derive useful insights from the data relating to products sold at different times of the year.
- On-demand, end-to-end automated, data ingestion services to receive and collate data from various external sources, including weather and social event data, to derive richer insights.
- A dashboard enabling reports on Data classification, Data lineages and metrics, in a single place.
- To improve customer engagement, offering value for money and high product quality, to achieve excellence and an accompanying high NPS.

2. Solution Architecture

Based on Data Fabric



Scalable, Highly Available & Secure

Faster Data Ingestion & Processing

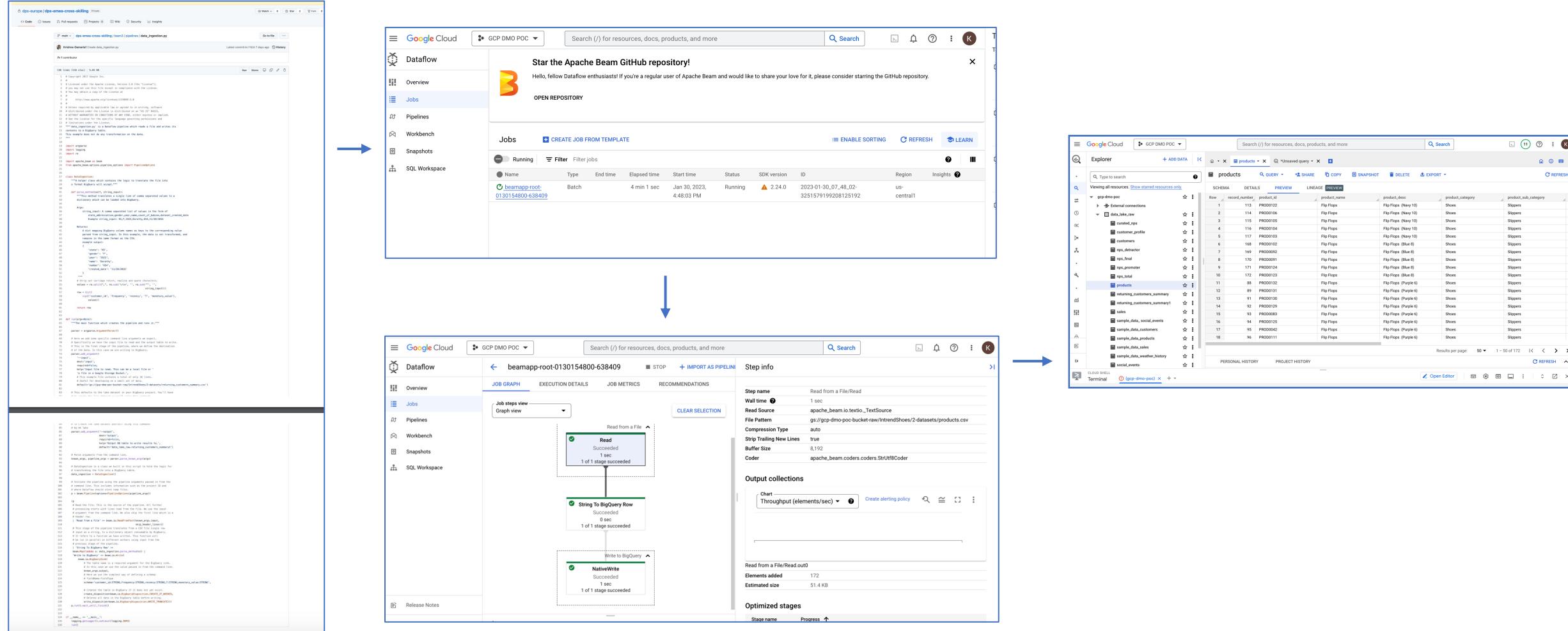
ML Powered Analytics Platform

Reduced Infrastructure & Management costs

Collaborative reporting

3. Data Ingestion

18 files were processed & ingested from Google cloud storage to BigQuery with Dataflow API (Beam Python ETL job)

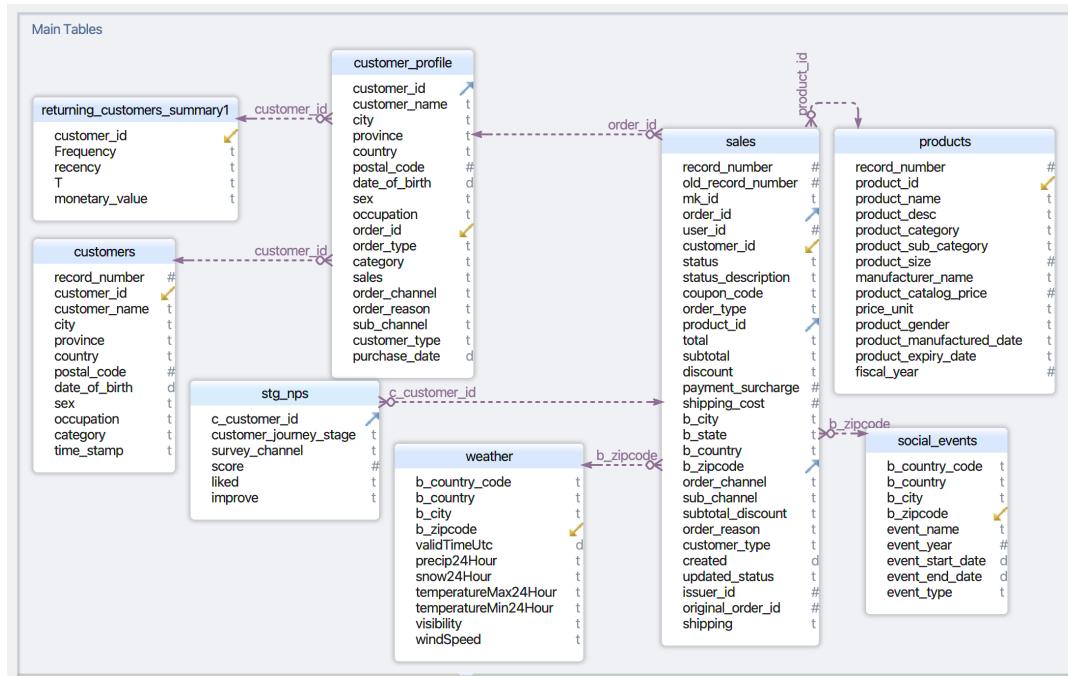


4. Data Model

18 BigQuery tables were clustered into 3 groups

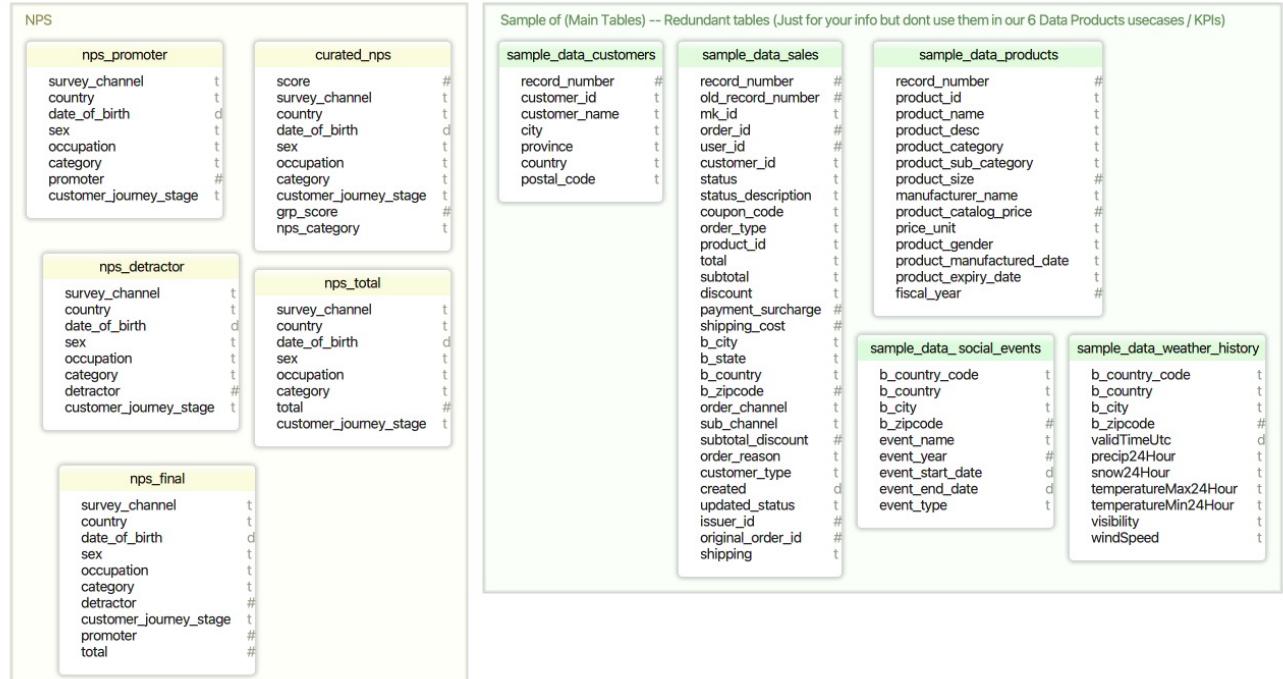
Main Tables Group

- Relational Data Model (Star Schema)
 - Sales
 - Customers
 - Products
 - Weather
 - Social events
 - NPS
 - Customer Profile
 - Returning customer summary



Symbols Legend:

- # - number
- t - string
- d - date



5. Data Mesh

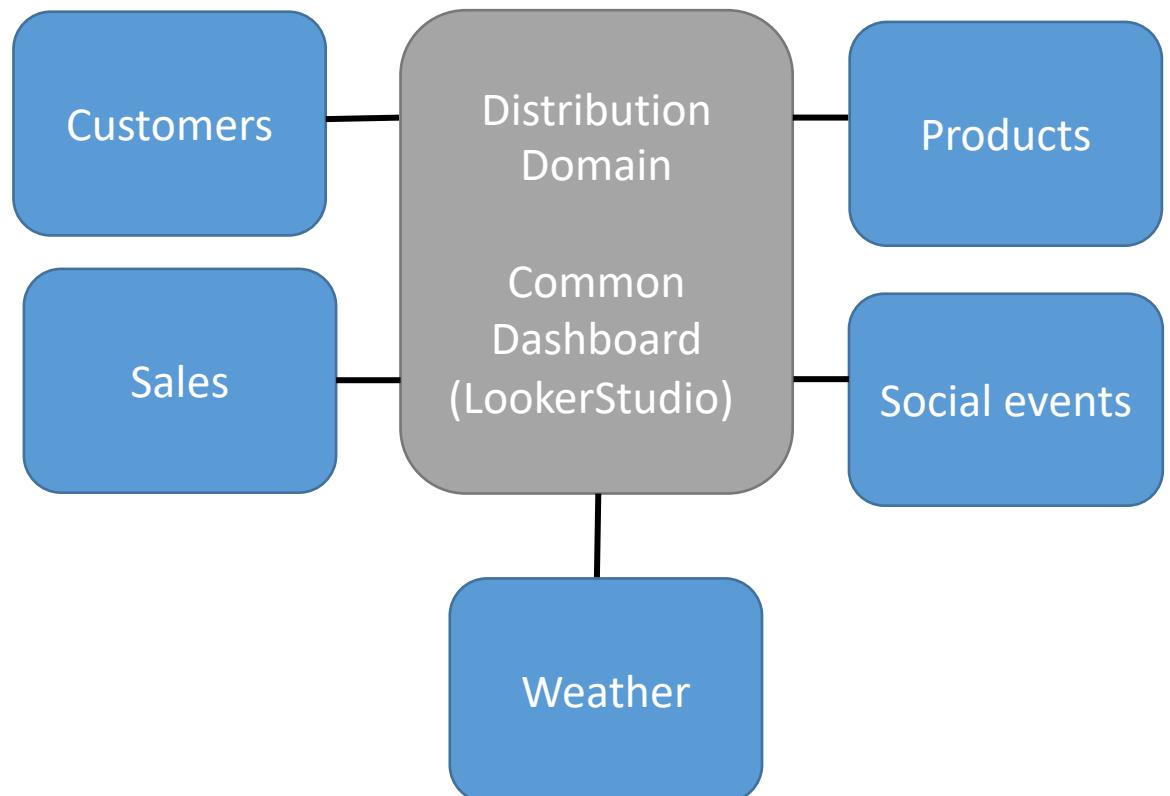
Cross grained and governed mesh model

Data Products:

- 5 KPIs were provided based on the given use cases, they are our data products

Benefits:

- promotes the adoption of cloud native and cloud platform technologies, moving away from batch processing into streaming pipelines (i.e. collect data in real-time)
- facilitates self-service applications from multiple data sources
- stronger security & compliance by enforcing standards for domain-agnostic data and access controls for sensitive data.



6. Data Products

Perform analysis in BigQuery on each one of the below Data Products / KPI's after making a copy of the tables relevant to a KPI from raw_zone to the processed zone.

Recommendations based on below 5 KPI's analysis

1. Sales trend on various products around the year

`customer_profile` --> Relevant for identifying how many products (table) a customer (table) has ordered to derive sales (table)

2. Revenue trend across various social events

3. Revenue trend across various weather conditions

4. Net promoter score

Join stg_nps with the customer profile, to see how customer reacted for each order (Preview nps tables of detractor, final, promoter tables)

5. Customer life time value

The returning customers can be build on the customer profile.

Viewing all resources. Show starred resources only.		
 Case1_RevenueTrend_WeatherConditions		
 Case2_RevenueTrend_SocialEvents		
 Case3_SalesTrend_Yearly		
 Case4_CustomerLifetimeValue		
 Case5_NetPromoterScore		
 Case6_GenZ_BuyingBehaviour		
  data_lake_raw		
 curated_nps		
 customer_profile		
 customers		
 nps_detractor		
 nps_final		
 nps_promoter		
 nps_total		
 products		
 returning_customers_summary1		
 sales		
 sample_data_social_events		
 sample_data_customers		
 sample_data_products		
 sample_data_sales		
 sample_data_weather_history		
 social_events		
 stg_nps		
 weather		

6.1. Sales trend on various products around the year

Data Visualisation (exploratory analytics) with LookerStudio reporting tool

The screenshot shows a Looker Studio dashboard titled "Sales trend on various products around the year". The dashboard includes a chart titled "Product Sales (FY)" and a table titled "Customer, Sales Products". A sidebar on the left lists numbered steps from 1 to 14. A callout arrow points from step 11 to the chart area.

Sales trend on various products around the year
Exploratory Analysis revealed Data Quality issues like mis-formatted data, depulication & anomalies

Product Sales (FY)

ADIDAS Record Count: 5,549

fiscal_year	manufacture_name	product_A_...	Record Count
1	2023	ADIDAS	20
2	2023	PUMA	20
3	2023	NIKE	20
4	2022	REEBOK	12
5	2022	NIKE	10
6	2022	PUMA	10
7	2019	ADIDAS	10
8	2023	REEBOK	10
9	2021	REEBOK	10
10	2014	ADIDAS	10
11	2023	PUMA	5

Customer, Sales Products

Data quality issues

Mis formatted categorical values. Ex: REEBOK has special characters in 7 rows

Deduplication

Product categorization has repeating values (shoes) in all rows.

Dates formatting issues in fiscal_year, Product expiry & manufactured

Anomalies

Prices of all product units are in INR instead of EUR

Ex: City, province, country, mk_id all 4 columns reveal that sales are happening only in 6 major cities

1
2
3
4
5
6
7
8
9
10
11
12
13
14

6.1. Sales trend on various products around the year

Data Engineering (& Data Processing): Dataprep vs Dataflow

Sales trend on various products around the year

Data processing with Dataprep is 3X faster than Dataflow / other ETL tools for Analytics / ML

- Addressed all DQ issues & cleaned the tables with **more than 16 data transformation rules**
- Normalized the data from **selected features** and passed them to **ML models** for predictions

The screenshot shows a table in the Google BigQuery Data Editor. The columns are labeled: status_description, sales_qty, order_channel, product_name, and price. A context menu is open over the first row, listing various data cleaning and transformation steps. Some of the items in the menu include:

- Delete code_id
- Delete customer_id
- Delete product_id
- Delete order_header
- Delete customer_table
- Delete created
- Set transaction_id as transaction_id -- REASON: REASON: None
- Delete customer_name
- Delete price_id
- Delete record_number
- Delete purchase_date
- Create dimension_product_instructional_data from Change data format of product_instructions data to MM/DD/YY-type
- Create dimension_product_usage_data from Change data format of product_usage data to MM/DD/YY-type
- Create dimension_local_view from Change data format of local_view to MM/DD/YY-type
- Create dimension_order_of_journey from Change data format of order_of_journey to MM/DD/YY-type
- Create dimension_order_of_journey from Change data format of order_of_journey to MM/DD/YY-type

Dataprep	Dataflow
Loaded processed data to BQ with Dataprep workflow in ~ 2 minutes	Loaded processed data to BQ with Dataflow Python ETL Job in ~ 5 minutes
Cleaned data visually by processing BQ tables from raw zone	Installed apache beam python SDK within docker env
Did pattern <u>transformation's</u> for mis-formatted data (Automatic schema detection)	Defined schema and transformations manually and ran the ETL job

6.1. Sales trend on various products around the year

Descriptive analytics on historical sales data

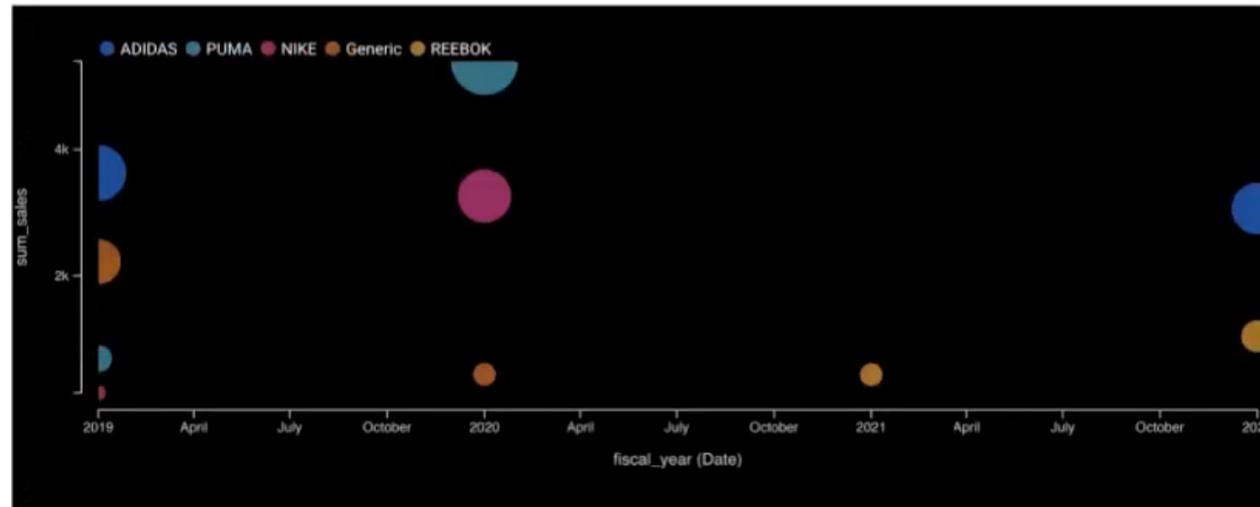
Product Sales (FY 2019 - 2022)

In 2022, 4K Sports shoes from Adidas & Reebok (mostly Running) were sold to Millineals (289 males, 19 females) under 55% discount sale (order_status: complete). Same customers (satisfactorily returning) from Amsterdam & Brussels bought shoes multiple times across FY 2022 from IntrendShoes store.

In 2019, Millineals (All students) bought slippers under 50% discount sale within unisex product category.

fiscal_year

Through out all 4 years, 100% correlation is found between Product sub category "Slippers" & Product gender "unisex"



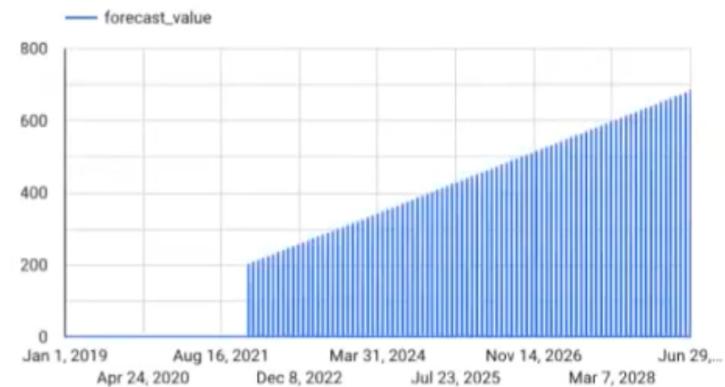
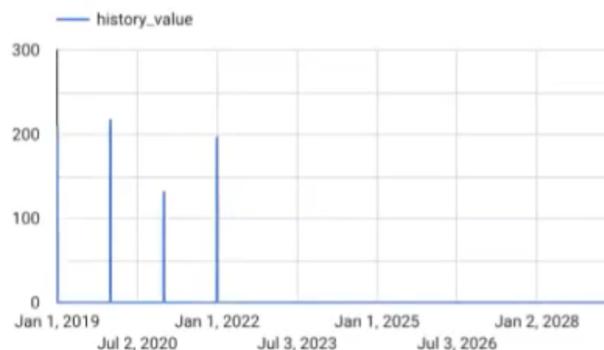
fiscal_year	manufacturer_na...	product_sub_cat...	product_gender	b_city
1. Jan 1, 2022, 12:00:0...	REEBOK	Sports shoes	Male	Paris
2. Jan 1, 2022, 12:00:0...	ADIDAS	Sports shoes	Male	Milan
3. Jan 1, 2022, 12:00:0...	ADIDAS	Sports shoes	Female	Paris
4. Jan 1, 2022, 12:00:0...	ADIDAS	Sports shoes	Female	Brussels
5. Jan 1, 2022, 12:00:0...	ADIDAS	Sports shoes	Male	Brussels
6. Jan 1, 2022, 12:00:0...	REEBOK	Sports shoes	Female	Brussels

6.1. Sales trend on various products around the year

Predictive & Prescriptive analytics on historical sales data

Forecast of Product Sales (FY 2023 - 2030)

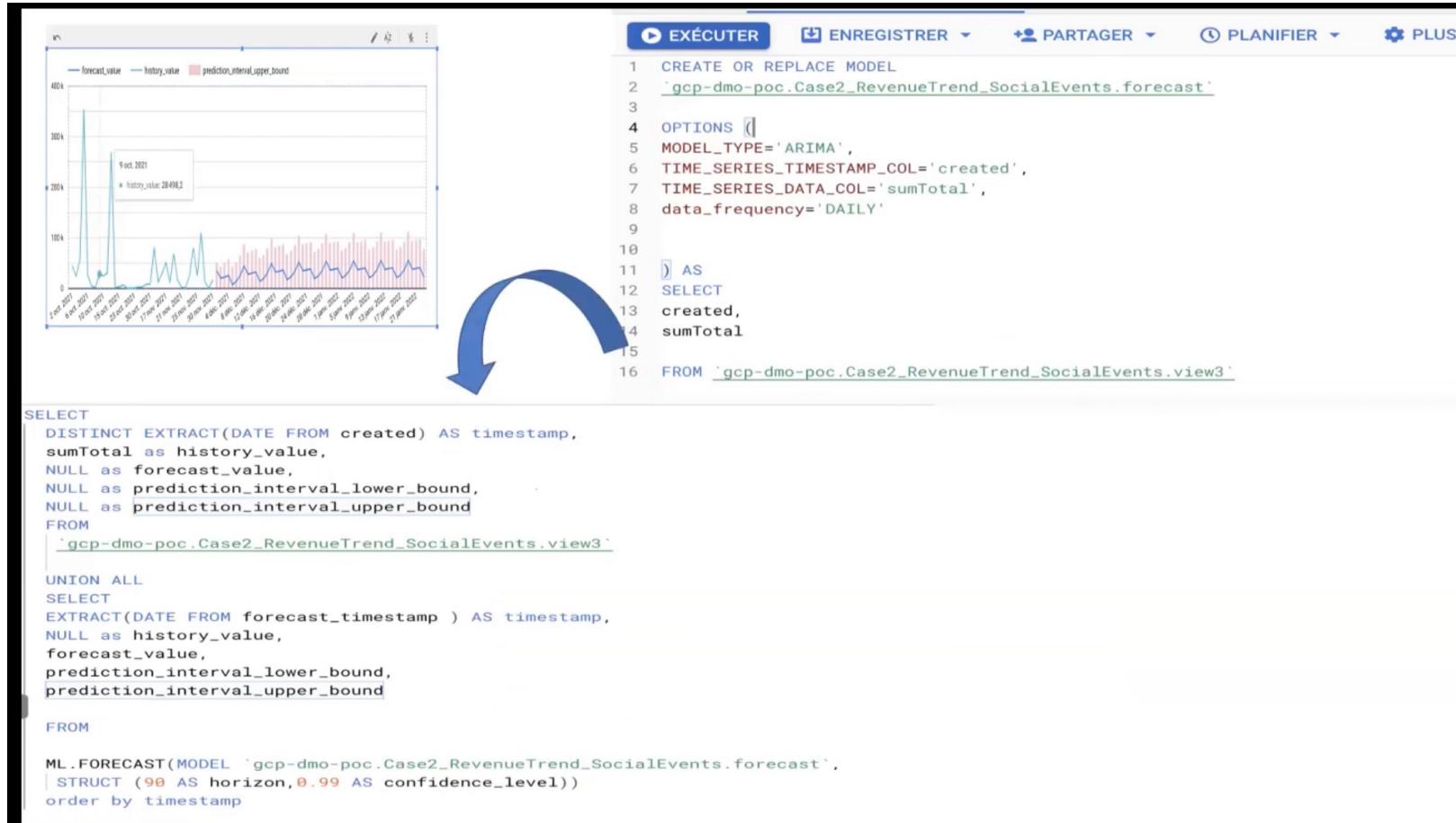
Recommendation: Stock up sports shoes from Adidas & Nike (top-seller across millennials & other generation customers) as predictions show vertical growth in sales
Based on the historical product sales data, forecast of the sales for the coming years expects a positive growth



fy ▾	history_value	forecast_value
1.	Jul 1, 2029	683.5485386628354
2.	Jun 1, 2029	678.1463458879242
3.	May 1, 2029	672.744153113013
4.	Apr 1, 2029	667.3419603381018
5.	Mar 1, 2029	661.9397675631906
6.	Feb 1, 2029	656.5375747882794
7.	Jan 1, 2029	651.1353820133681

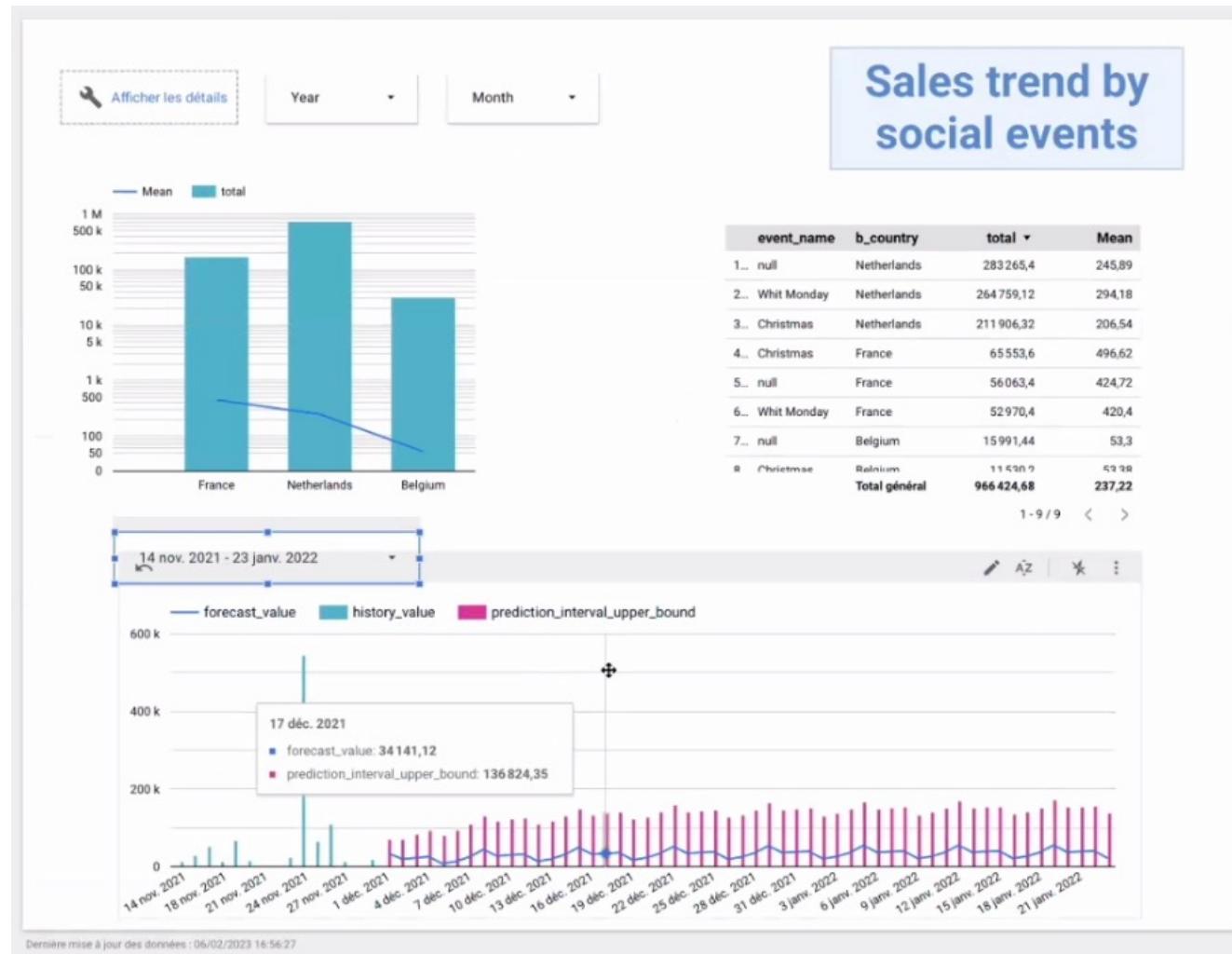
6.2. Revenue trend across various Social Events

Bigquery ML Arima time-series prediction model



6.2. Revenue trend across various Social Events

Visually explore social events by year & month drop-down



6.3. Revenue trend across various Weather conditions

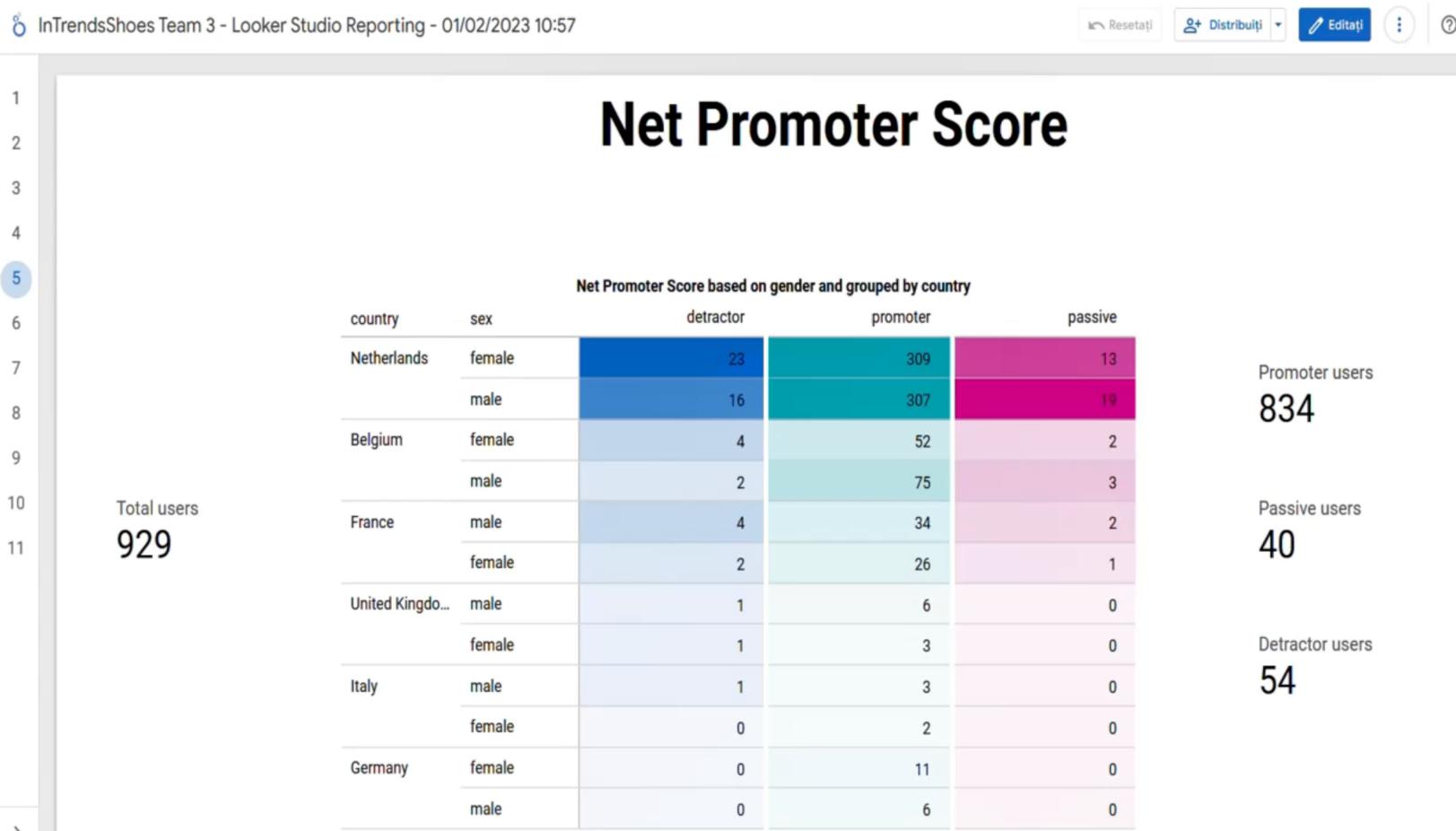
Bigquery ML Arima time-series prediction model

Relationship between Revenue trend and Weather



6.4. Net Promoter Score

To know positive & negative responses from customers based on country and gender



6.4. Net Promoter Score

Grouped by customer journey - To know where customer become unsatisfied

Net Promoter Score based on category and grouped by customer journey stage					
customer_jou...	category	promoter	passive	detractor	total
One week after...	Millennials	153	6	14	173
	GenZ	117	2	0	119
	GenX	85	11	7	103
	BabyBoomers	53	5	4	62
After placing t...	Millennials	149	7	2	158
	GenZ	96	4	0	100
	GenX	84	1	6	91
	BabyBoomers	50	4	1	55
After abandon...	Millennials	14	0	6	20

occupation	Net Promoter Score based on occupation			total
	promoter	passive	detractor	
Service	426	20	28	474
Business	228	9	12	249
Student	163	9	13	185
Retired	17	2	1	20
Net Promoter Score based on category				
category	promoter	passive	detractor	total
Millennials	316	13	22	351
GenZ	229	6	6	241
GenX	177	12	15	204
BabyBoomers	112	9	11	132

6.4. Net Promoter Score

The screenshot shows a data exploration interface with two main panes. The left pane, titled 'Explorer', displays a tree view of resources under a folder named 'gcp-dmo-poc'. The right pane shows a query editor with a SQL script for calculating Net Promoter Score (NPS) categories.

```
1 WITH nps_data AS (
2     SELECT
3         country, sex, occupation, category, score,
4         survey_channel,
5         CASE
6             WHEN score <= 6 THEN 'detractor'
7             WHEN score >= 9 THEN 'promoter'
8             ELSE 'passive'
9         END AS customer_group
10    FROM
11        `NetPromoterScore_Case5.nps_promoters`
12    UNION ALL
13
14    SELECT
15        country, sex, occupation, category, score,
16        survey_channel,
17        CASE
18            WHEN score <= 6 THEN 'detractor'
19            WHEN score >= 9 THEN 'promoter'
20            ELSE 'passive'
21        END AS customer_group
22    FROM
23        `NetPromoterScore_Case5.nps_detractor`
24    UNION ALL
25
26    SELECT
27        country, sex, occupation, category, score,
28        survey_channel,
29        CASE
30            WHEN score <= 6 THEN 'detractor'
31            WHEN score >= 9 THEN 'promoter'
32            ELSE 'passive'
33        END AS customer_group
34    FROM
35        `NetPromoterScore_Case5.nps_passive`
36
37    ) SELECT
38        survey_channel, country, sex, occupation, category,
39        SUM(CASE WHEN customer_group = 'detractor' THEN 1 ELSE 0 END) AS detractor,
40        SUM(CASE WHEN customer_group = 'promoter' THEN 1 ELSE 0 END) AS promoter,
41        SUM(CASE WHEN customer_group = 'passive' THEN 1 ELSE 0 END) AS passive,
42        COUNT(*) AS total
43    FROM nps_data
44    GROUP BY 1, 2, 3, 4, 5
```

6.5 CLV

Customer Lifetime Value



Customer lifetime value or LTV is the company's total revenue from one customer during its lifetime. It helps predict customers' responses and spending on their products and services.

Customer Lifetime Value Formulas

$$\text{LTV} = \text{Average Value of Sale} \times \text{Number of Transactions} \times \text{Retention time} \times \text{Profit Margin}$$

Historical

Predictive

$$\text{LTV} = \frac{\text{Total Revenue For Chosen Period}}{\text{Total Number Of Customers.}}$$

$$\text{LTV} = \frac{T \times \text{AOV} \times \text{AGM} \times \text{ALT}}{\text{Number Of Customers}}$$

6.5 CLV

Historical Approach

Source table: sales

BigQuery script

BigQuery table



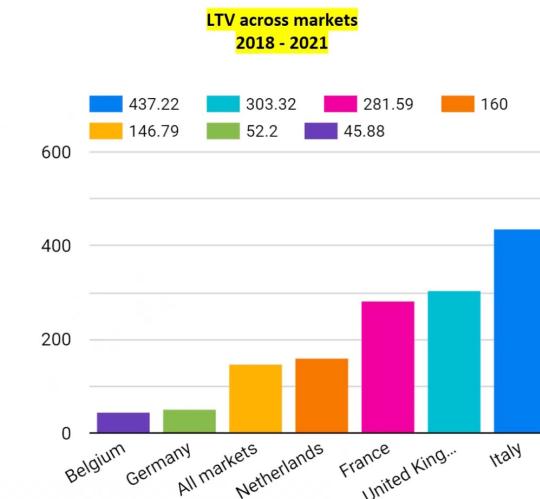
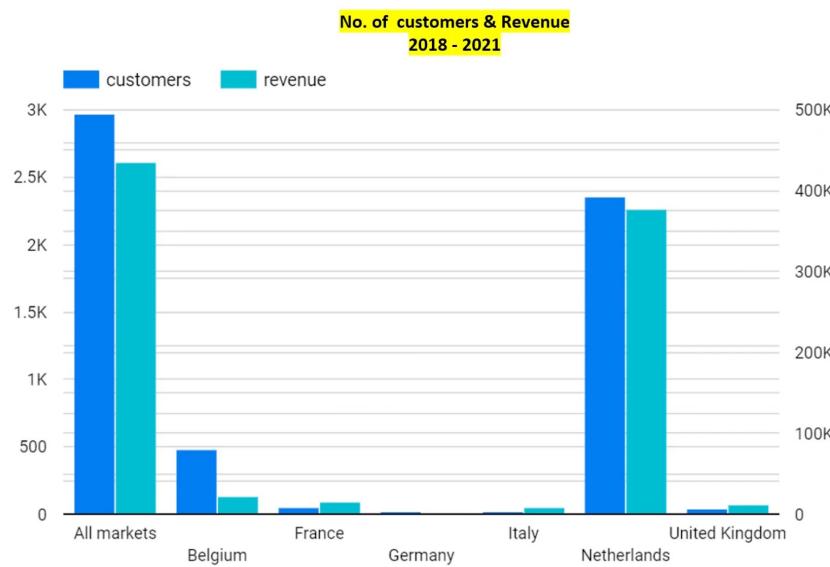
Row	country	revenue	customers	LTV
1	All markets	435685.19	2968.0	146.79
2	Belgium	21976.8	479.0	45.88
3	France	15487.5	55.0	281.59
4	Germany	939.6	18.0	52.2
5	Italy	8307.18	19.0	437.22
6	Netherlands	377447.76	2359.0	160.0
7	United Kingdom	11526.35	38.0	303.32

```
create or replace table Case4_CustomerLifetimeValue.historical_LTV as ()  
with base_table as (  
| select *;  
| RANK() over (PARTITION BY b_country, customer_id ORDER BY updated_status DESC) as order_rank  
from `Case4_CustomerLifetimeValue.sales`  
where lower(status_description) in ('complete','delivered','shipped')  
)  
  
base_country_LTV as (  
select  
| b_country as country,  
| sum(total) as total_revenue,  
| count (user_id) as total_customers  
from base_table  
where order_rank = 1  
group by b_country  
)  
  
calculate_country_LTV as (  
select  
| country,  
| round (total_revenue,2) as revenue,  
| round(total_customers,2) as customers,  
| round(total_revenue/total_customers,2) as LTV  
from base_country_LTV  
)  
  
calculate_total_LTV as (  
select  
| 'All markets' as country,  
| round (total_revenue,2) as revenue,  
| round(total_customers,2) as customers,  
| round(total_revenue/total_customers,2) as LTV  
from (  
select  
| sum(total) as total_revenue,  
| count (user_id) as total_customers  
from `Case4_CustomerLifetimeValue.sales`  
where lower(status_description) in ('complete','delivered','shipped')  
)  
  
select *  
from calculate_country_LTV  
union all  
select *  
from calculate_total_LTV  
order by country
```

6.5 CLV

Visualization – Looker Studio

Good Practice: Customer Lifetime Value should be **at least three times greater than your Customer Acquisition Cost (CAC)**



Lesson Learned

➤ Technical

- Maintain backup snapshots of VMs and Databases/Datasets
- Check for data consistency and integrity
- Use automated pipelines for data ingestion and pre-processing
- Define a common architecture to deliver the data products
- Make the product available via an https site and follow data governance rules

➤ Business

- Define team member responsibilities early in the game
- Be flexible and resilient when resources become unavailable
- Test the product and get user feedback on individual features.

спасибо
obrigado
dziekuje
hvala
danke
sagolun
sukriya
terima kasih
감사합니다
merci
dank je
ευχαριστώ
thank you
dank je
gracias
mochchakkeram
go raibh maith agat
arigatō
takk
dakujem
мерси
ngiyabonga
teşekkür ederim
gracias
mochchakkeram
go raibh maith agat
arigatō
takk
dakujem
мерси
narrative text

7. DevOps Workflows with Cloud Build

The screenshot shows the Google Cloud Platform interface with two main windows open:

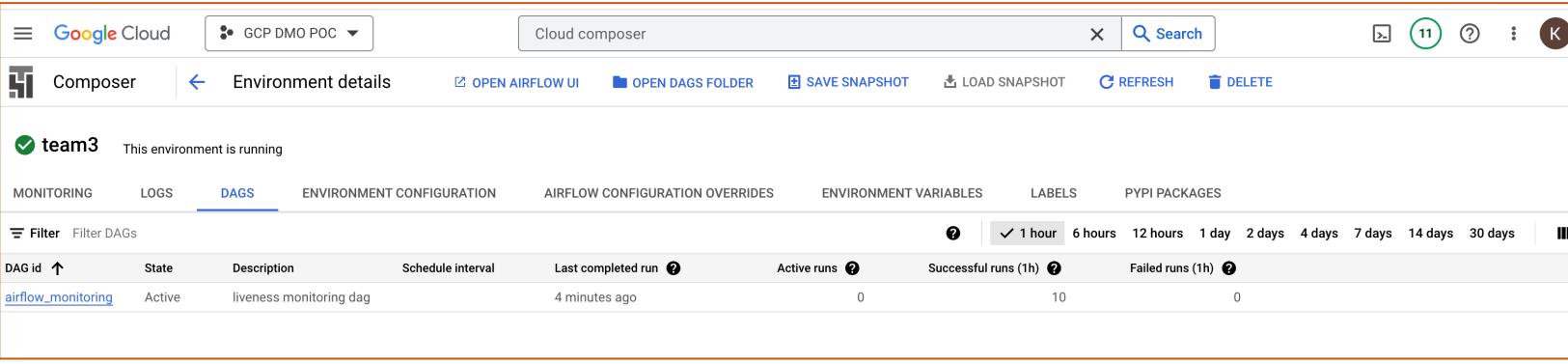
Cloud Source Repositories (Top Window):

- Header: Cloud Source Repositories, Data search bar, Cloud Console link.
- Team dropdown: Team3.
- Message: "Your repository is currently empty. Add some code using a selected method and then refresh your browser. Contents added to this repository can take some time to show up in search results. [Learn more](#)".
- Text: "Select an option to push code to your repository:"
- Push code from a local Git repository
- Clone your repository to a local Git repository
- Text: "Select your preferred authentication method"
- Buttons: SSH authentication (selected), Google Cloud SDK, Manually generated credentials.
- List of steps:
 1. Setup SSH key. [Learn how](#). If you already have a SSH key on your machine, skip to step 2. [Find SSH Keys on your machine](#).
 2. Register the SSH key with Google Cloud.
 3. Clone this repository to a local Git repository.

Cloud Build (Bottom Window):

- Header: Google Cloud, GCP DMO POC dropdown, search bar (cloud build), DISMISS, ACTIVATE buttons.
- Sidebar: Cloud Build, Dashboard, History, Triggers (selected), Settings.
- Region dropdown: europe-west3.
- Filter input: Enter property name or value.
- Table: Triggers
 - Name: test3-test
 - Description: test3
 - Repository: Team3
 - Event: Push to branch
 - Build configuration: cloudbuild.yaml
 - Status: Enabled
 - Action: RUN
- Terminal prompt: \$ git push -u origin main

8. Data Orchestration with Composer



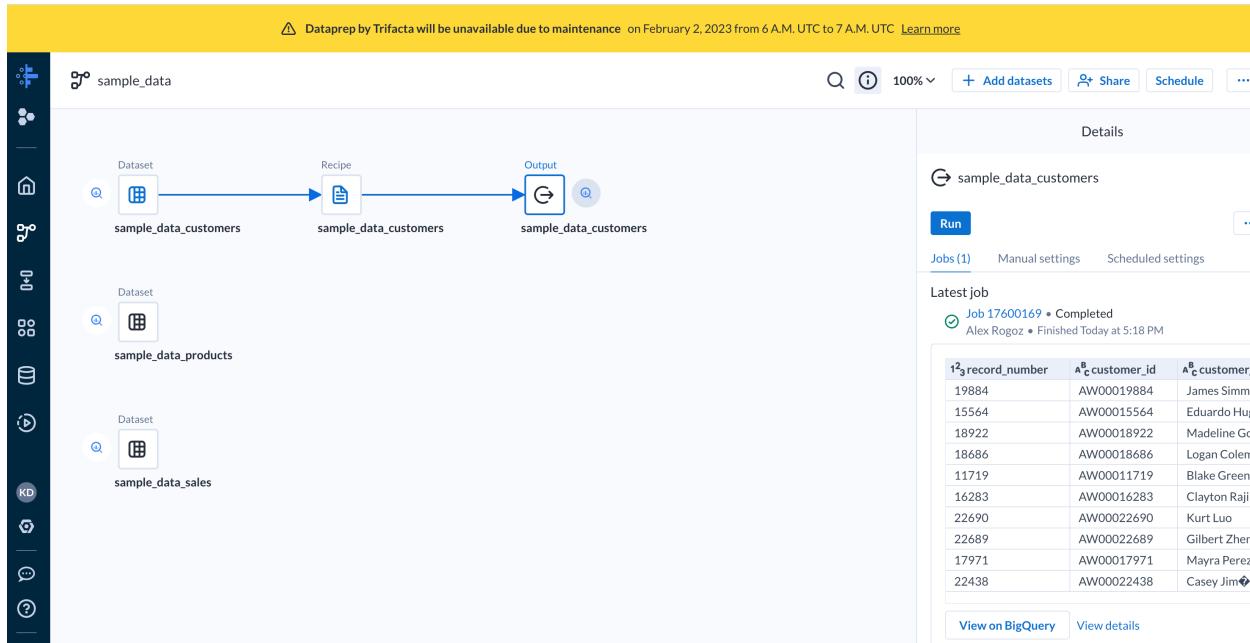
The screenshot shows the Google Cloud Composer interface for an environment named "team3". The "DAGS" tab is selected. A table lists one DAG: "airflow_monitoring", which is active and has run 10 times successfully in the last hour. The interface includes tabs for MONITORING, LOGS, ENVIRONMENT CONFIGURATION, AIRFLOW CONFIGURATION OVERRIDES, ENVIRONMENT VARIABLES, LABELS, and PYPI PACKAGES. It also features a search bar, a sidebar with icons for monitoring, logs, and airflow, and a top navigation bar with options like OPEN AIRFLOW UI, OPEN DAGS FOLDER, and various snapshot and refresh buttons.

DAG id	State	Description	Schedule interval	Last completed run	Active runs	Successful runs (1h)	Failed runs (1h)
airflow_monitoring	Active	liveness monitoring dag		4 minutes ago	0	10	0

6.5. Sales trend on various products around the year

Tables cleaning options

- Identifies which tables need data processing (solving data quality issues like common time standards, null, table join keys etc) with below tools
 - Visually: Dataprep / Dataplex on tables in BigQuery
 - Programmatically: BigQuery SQL Scripts / Dataflow ETL script (Python / Java etc)



The screenshot shows the Trifecta Dataprep interface with a preview of the 'products' dataset. The preview table has columns: price_unit, product_gender, product_manufactured_date, product_expiry_date, and fiscal_. The data shows various product entries with their respective gender, manufacturing dates, and expiry dates. To the right of the preview, there is a sidebar with several data cleaning actions:

- Delete rows: where ISMISMATCHED(product_manufactured_date,'mm'-'dd'-'yyyy'))
- Keep rows: where ISMISMATCHED(product_manufactured_date,'mm'-'dd'-'yyyy'))
- Create a new column: ISMISMATCHED(product_manufactured_date,'mm'-'dd'-'yyyy'))
- Set: Set product_manufactured_date to IFNULL(Datetime,'mm'-'dd'-'yyyy'))

6.5. Sales trend on various products around the year

Tables cleaning with Dataflow Python ETL script

- Invalid date: '09-23-2021' Field: product_manufactured_date; Value: 09-23-2021
- Need access to the gs folder for log analysis -> gs://gcp-dmo-poc-bucket-raw/test/beamapp-root-0206144421-803039.1675694661.803300/227596339129495248/dax-tmp-2023-02-06_06_44_23-5333317336131316020-S01-0-688ba7961500b8c9/-shard--try-0b2f9830dba94de8-endshard.json

The screenshot shows the Google Cloud Storage interface. At the top, there is a log file named 'data-ingestion.py' with the following content:

```
INFO: Up. Rows: 1; errors: 1; max bad: 0; error percent: 0; error: Invalid date: '09-23-2021' Field: product_manufactured_date; value: 09-23-2021
INFO:apache_beam.runners.dataflow.runner:2023-02-06T14:48:30.374Z: JOB_MESSAGE_DETAILED: Cleaning up.
INFO:apache_beam.runners.dataflow.runner:2023-02-06T14:48:30.416Z: JOB_MESSAGE_DEBUG: Starting worker pool teardown.
INFO:apache_beam.runners.dataflow.runner:2023-02-06T14:48:30.443Z: JOB_MESSAGE_BASIC: Stopping worker pool...
INFO:apache_beam.runners.dataflow.runner:2023-02-06T14:49:12.765Z: JOB_MESSAGE_DETAILED: Autoscaling: Resized worker pool from 1 to 0.
INFO:apache_beam.runners.dataflow.runner:2023-02-06T14:49:12.807Z: JOB_MESSAGE_BASIC: Worker pool stopped.
INFO:apache_beam.runners.dataflow.runner:2023-02-06T14:49:12.830Z: JOB_MESSAGE_DEBUG: Tearing down pending resources...
INFO:apache_beam.runners.dataflow.runner:Job 2023-02-06_06_44_23-5333317336131316020 is in state JOB_STATE_FAILED
Traceback (most recent call last):
  File "data_ingestion.py", line 150, in <module>
    run()
  File "data_ingestion.py", line 145, in run
    p.run().wait_until_finish()
  File "/usr/local/lib/python3.7/site-packages/apache_beam/runners/dataflow/dataflow_runner.py", line 1633, in wait_until_finish
    self)
apache_beam.runners.dataflow.dataflow_runner.DataflowRuntimeException: Dataflow pipeline failed. State: FAILED, Error: Workflow failed. Causes: S01:ReadFromString To BigQuery Row+Write to BigQuery/NativeWrite failed., BigQuery import job "dataflow_job_227596339129494027-B" failed., BigQuery import job "dataflow_job_227596339129494027-B" in project "gcp-dmo-poc" finished with error(s): errorResult: Error while reading data, error message: JSON table encountered too many errors, giving up. Rows: 1; errors: 1. Please look into the errors[] collection for more details. File: gs://gcp-dmo-poc-bucket-raw/test/beamapp-root-0206144421-803039.1675694661.803300/227596339129495248/dax-tmp-2023-02-06_06_44_23-5333317336131316020-S01-0-688ba7961500b8c9/-shard--try-0b2f9830dba94de8-endshard.json, error: Error while reading data, error message: JSON table encountered too many errors, giving up. Rows: 1; errors: 1. Please look into the errors[] collection for more details. File: gs://gcp-dmo-poc-bucket-raw/test/beamapp-root-0206144421-803039.1675694661.803300/227596339129495248/dax-tmp-2023-02-06_06_44_23-5333317336131316020-S01-0-688ba7961500b8c9/-shard--try-0b2f9830dba94de8-endshard.json, error: Error while reading data, error message: JSON processing encountered too many errors, giving up. Rows: 1; errors: 1; max bad: 0; error percent: 0, error: Invalid date: '09-23-2021' Field: product_manufactured_date; value: 09-23-2021
```

Below the log, there is a list of files in the 'beamapp-root' folder:

Name	Type	Size	Last Modified	Action
root@483110b0cbfc:/dataflow/pipelines#				
apache_beam-2.24.0-cp37-cp37m	application/octet-stream	8.1 MB	Feb 6, 2023, 3:44:22 PM	Download More
dataflow_python_sdk.tar	application/octet-stream	2.1 MB	Feb 6, 2023, 3:44:22 PM	Download More
pickled_main_session	application/octet-stream	4.3 KB	Feb 6, 2023, 3:44:22 PM	Download More
pipeline.pb	application/octet-stream	15.7 KB	Feb 6, 2023, 3:44:22 PM	Download More