

- A. Understanding the concept of Customer Analytics
- B. How to use Customer Analytics
- C. Exploratory Analysis
 - 1. Funnels
 - a. Session of Actions Funnel
 - i. Daily Funnel
 - ii. Weekly Funnel
 - iii. Monthly Funnel
 - iv. Hourly Funnel
 - b. Customer of Actions Funnel
 - i. Daily Funnel
 - ii. Weekly Funnel
 - iii. Monthly Funnel
 - iv. Hourly Funnel
 - 2. Cohorts
 - a. Daily/Weekly Order Cohort
 - i. Daily Download to 1st Order Cohort
 - ii. Daily Cohort From 1st to 2nd Order
 - iii. Daily Cohort From 2nd to 3rd Order
 - iv. Daily Cohort From 3rd to 4th Order
 - v. Weekly Download to 1st Order Cohort
 - vi. Weekly Cohort From 1st to 2nd Order
 - vii. Weekly Cohort From 2nd to 3rd Order
 - viii. Weekly Cohort From 3rd to 4th Order
 - 3. Descriptive Stats
 - a. Stats Purchase
 - i. Daily Orders
 - ii. Hourly Orders
 - iii. Weekly Orders
 - iv. Monthly Orders
 - b. Descriptive Statistics
 - i. Weekly Average Session Count per Customer
 - ii. Weekly Average Purchase Count per Customer
 - iii. Payment Amount Distribution
 - iv. Weekly Average Payment Amount
 - 4. Product Analytics
 - a. Top Purchased Products
 - i. Most Combined Products
 - ii. Most Purchased Products
 - iii. Most Purchased Categories
- D. Machine Learning Works
 - 1. A/B Tests
 - a. A/B Test Promotion
 - i. Promotion Comparison
 - ii. Order and Payment Amount Difference for Before and after Promotion Usage
 - iii. Before - After time Periods customers' Average Purchase Payment Amount Test (Test Accepted!)
 - iv. Before - After time Periods customers' Average Purchase Payment Amount Test (Test Rejected!)
 - v. Before - After time Periods customers' Total Purchase Count Test (Test Accepted!)
 - vi. Before - After time Periods customers' Total Purchase Count Test (Test Rejected!)
 - b. A/B Test Product
 - i. Before - After time Periods customers' Average Purchase Payment Amount Test (Test Accepted!)
 - ii. Before - After time Periods customers' Average Purchase Payment Amount Test (Test Rejected!)
 - iii. Before - After time Periods customers' Total Purchase Count Test (Test Accepted!)
 - iv. Before - After time Periods customers' Total Purchase Count Test (Test Rejected!)

- c. A/B Test Customer Segment Change
 - i. Daily Customer Total Order Count per Customer Segment
 - ii. Weekly Customer Total Order Count per Customer Segment
 - iii. Monthly Customer Total Order Count per Customer Segment
 - iv. Daily Customers' Average Payment Amount per Customer Segment
 - v. Weekly Customers' Average Payment Amount per Customer Segment
 - vi. Monthly Customers' Average Payment Amount per Customer Segment
- 2. Customer Segmentation
 - a. RFM
 - i. RFM 3D Scatter
 - ii. Frequency - Recency
 - iii. Monetary - Frequency
 - iv. Recency - Monetary
 - b. Segmentation
 - i. Customer Segmentation TreeMap
 - ii. Frequency Segmentation
 - iii. Monetary Segmentation
 - iv. Recency Segmentation
- 3. CLV Prediction
 - i. Next Week CLV Prediction
 - ii. Customers of Segments of Total Predicted Purchase Amounts
- 4. Anomaly detection
 - i. Daily Funnel Anomaly
 - ii. Daily Cohort Anomaly
 - iii. Daily Cohort Anomaly With Scores (Download to First Order)
 - iv. Daily Order Anomaly
 - v. CLV RFM Vs Current RFM
- E. Configurations
 - 1. Profiles
 - 2. Settings
 - 3. Sign In/out
- F. CustomerAnalytics DashBoard
 - 1. Overall Dashboard
 - 2. Customers Dashboard
- G. Searching
 - 1. Products
 - 2. Clients
 - 3. Promotions
 - 4. Dimensions
- H. Data Pipeline for ML Works and Exploratory Analysis
- I. Use CustomerAnalytics via Python

A. Understanding the concept of CustomerAnalytics

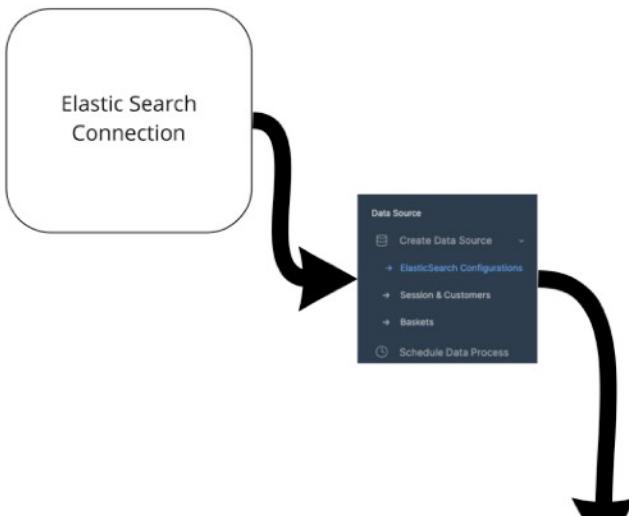
This platform helps the business related to e-commerce. It also helps the business which has users (B2C) and has their data related to engagement to the business.

At this platform, There are parts which are Exploratory analysis and Machine Learning Works. These parts have sub analysis and works. These parts also have individual built-in dashboards in or to visualize the results of the outputs at them.

CustomerAnalytics platform needs to be supported with data-source connections, data sources of columns matching to its data sources (Sessions, Customers, and Products), accessible ElasticSearch Service, and triggering data to the ElasticSearch by the users. The platform supports the data source connections which are '.csv', 'PostgreSQL', 'GCloud Bigquery', and 'AWS redShift'. **Sessions and Customers data sources** are required, but the **Products data source** is optional (without products data source, you can need to see the real charts for A/B Test Product, Product analytics). In addition to that additional actions are optional (you probably see the additional actions in Funnels). If you need to check the data source with a dimension (region, location, business type, etc), it is possible to assign dimensions columns at the session data source. You may filter each dashboard with the given dimensions after that. Promotions of each order can be added to the data sources (optional). After that, you may see A/B Test Promotions, promotion individual searching (check G. Searching) with real data sets.

At ElasticSearch, there are created 3 indexes. **Orders Index (orders)** is fetched data from Session and Products Data sources with columns. These Orders Index columns are id (session ID), client (client ID), session_start_date (timestamp), end_date (session end timestamp, optional), payment amount (float numeric amount of purchased orders), discount amount (discount of purchased orders, optional), promotion_id (optional), dimension (optional), has_purchased(if True, session is ended with purchase, if False, session is ended with no o-purchase). Order Index of Product data source columns is a basket which is an object with keys are also the products data source of columns which are order_id (Session ID with basket), price (the price of the product), category (category of the Product), product (product name of ID). **Downloads Index (downloads)** is also created with customer Data source in order to assign data for customers when they are first engaged to the business or when they downloaded or signup. 3rd Index is **Reports Index (reports)** which is stored with KPIs data sets, outputs, and results of Exploratory Analysis and Machine Learning Works. Each report is a document of the index with a separate date. report_name represents the main analysis, type represents sub-analysis and report_Date represents the created date of the report or analysis.

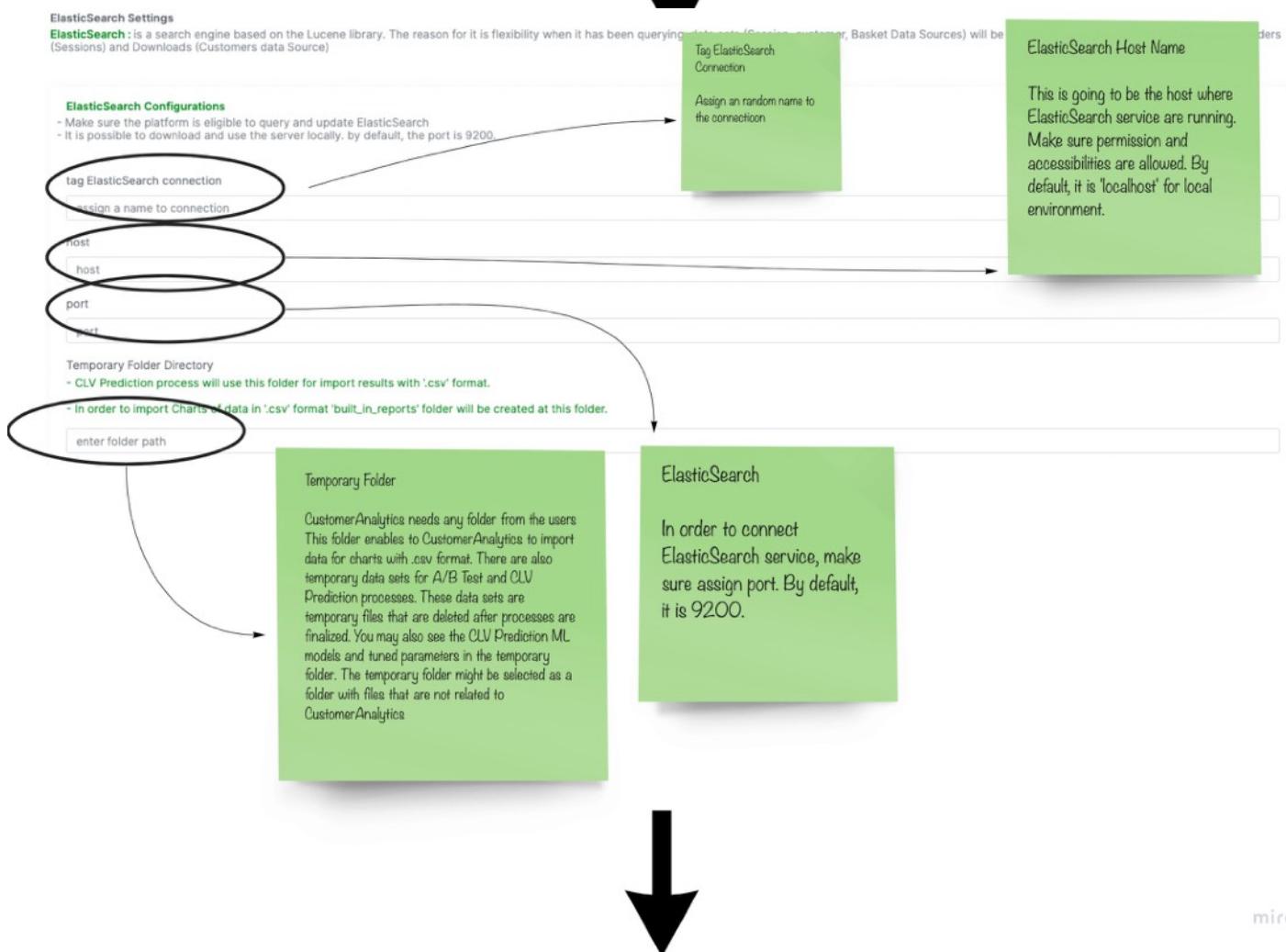
B. How to use CustomerAnalytics



There are 3 steps for initializing CustomerAnalytics;

1. Configuration of ElasticSearch Connection
2. Sessions & Customers & Products Data source Connections
3. Initialize the Schedule process

All data sets (Sessions, Customers, Products) are transferred to ElasticSearch Indexes. During the processes, there are temporary reports which are also stored at ElasticSearch. In order to run the platform, an accessible ElasticSearch Connection is required. In addition to that, there must be a folder with write/delete permission which the platform will use and store the reports with given folders



there.

Once you click on 'Connect', the platform will check for both connectivities of ElasticSearch host/url/port and temporary folder of accessibility.

It is also crucial that RAM and CPU consumption are checked by the user. While ElasticSerach is running, it might be useful to use with command;

```

"settings": {
  "number_of_shards": 2,
  "number_of_replicas": 0,
  "index.mapping.total_fields.limit": 20000,
  "index.query.default_field": 10000000,
  "index.max_result_window": 10000000,
  "index": {
    "analysis": {
      "filter": {
        "autocomplete_filter": {
          "type": "edge_ngram",
          "min_gram": 1,
          "max_gram": 10
        }
      },
      "analyzer": {
        "autocomplete": {
          "type": "custom",
          "tokenizer": "standard",
          "filter": [
            "lowercase",
            "autocomplete_filter"
          ]
        }
      }
    }
  }
}

"mappings": {
  "properties": {
    "date": {"type": "date"},
    "session_start_date": {"type": "date"},
    "discount_amount": {"type": "float"},
    "payment_amount": {"type": "float"},
    "actions": {
      "properties": {
        "has_sessions": {"type": "boolean"},
        "has_basket": {"type": "boolean"},
        "order_screen": {"type": "boolean"},
        "purchased": {"type": "boolean"}
      }
    }
  }
}

"mappings": {
  "properties": {
    "download_date": {"type": "date"},
    "signup_date": {"type": "date"}
  }
}

"mappings": {
  "properties": {
    "report_date": {"type": "date"},
    "report_types_to": {"type": "integer"},
    "report_types_from": {"type": "integer"},
    "frequency_segment": {"type": "float"},
    "monetary_segment": {"type": "float"},
    "recency_segment": {"type": "float"}
  }
}

```

`ES_JAVA_OPTS="-Xms4g -Xmx4g" ./elasticsearch`

When indexes are created there are some constant parameters that are stored at `configs.py`.

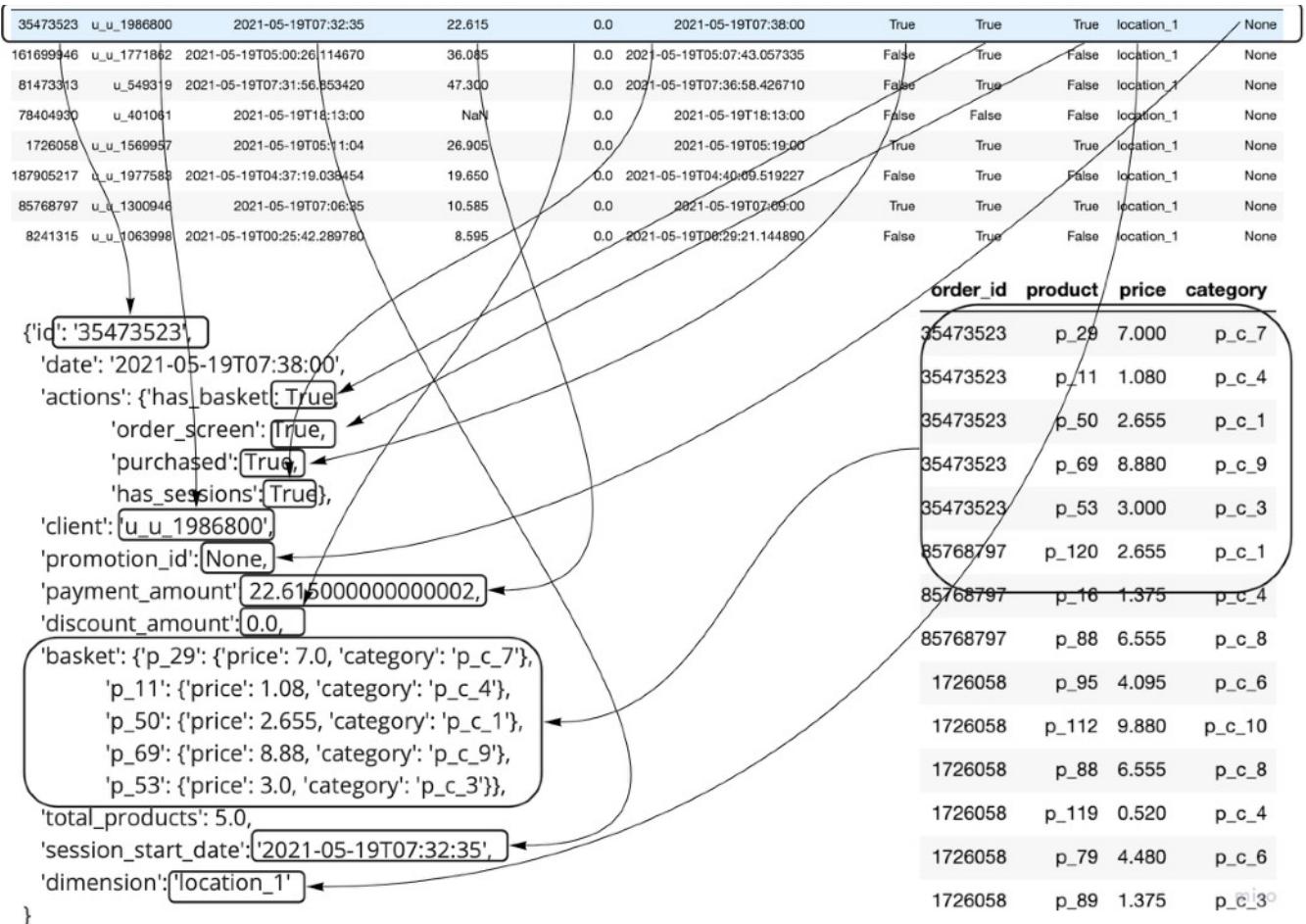
These constants are used at `settings` when indexes are created. These constants are;

1. **number_of_shard:** The number of primary shards that an index should have. Defaults to 2
2. **number_of_replicas:** The number of replicas each primary shard has. Defaults to 0.
3. **total field.limit:** maximum number of fields at document
4. **max_result_window:** The maximum value of `from + size` for searches to this index. Defaults to 10000000.

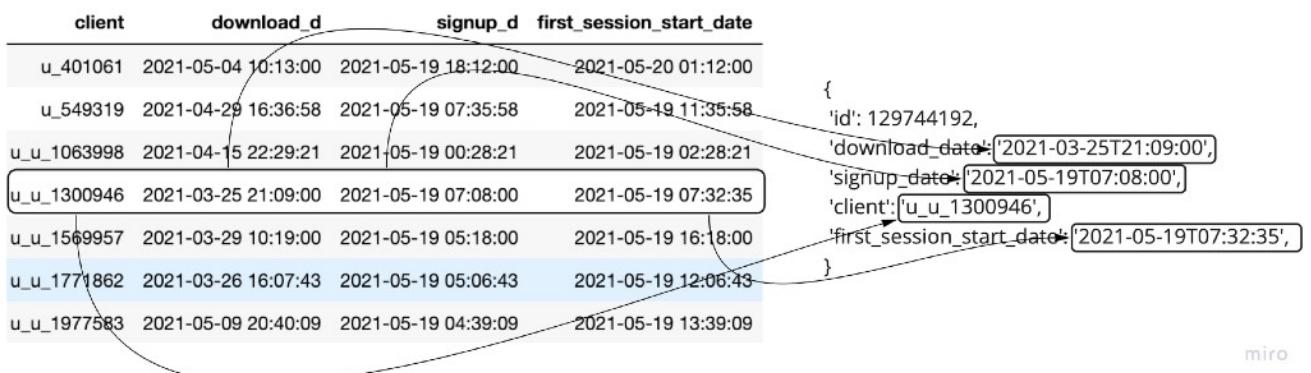
These are also fields that are mapping with their types;

1. **date:** orders index - type; date
2. **session_start_date:** orders index - type; date
3. **discount_amount:** orders index
4. **payment_amount:** orders index
5. **actions.has_sessions:** orders index
6. **actions.has_sessions:** orders index
7. **actions.order_screen:** orders index
8. **actions.purchased:** orders index
9. **download_date:** downloads index - type; date
10. **signup_date:** downloads index - type; date
11. **report_date:** reports index - type; date
12. **report_types.to:** reports index - type; integer
13. **report_types.from:** reports index - type; integer
14. **frequency_segments:** reports index - type; float
15. **recency_segments:** reports index - type; float
16. **monetary_segments:** reports index - type; float

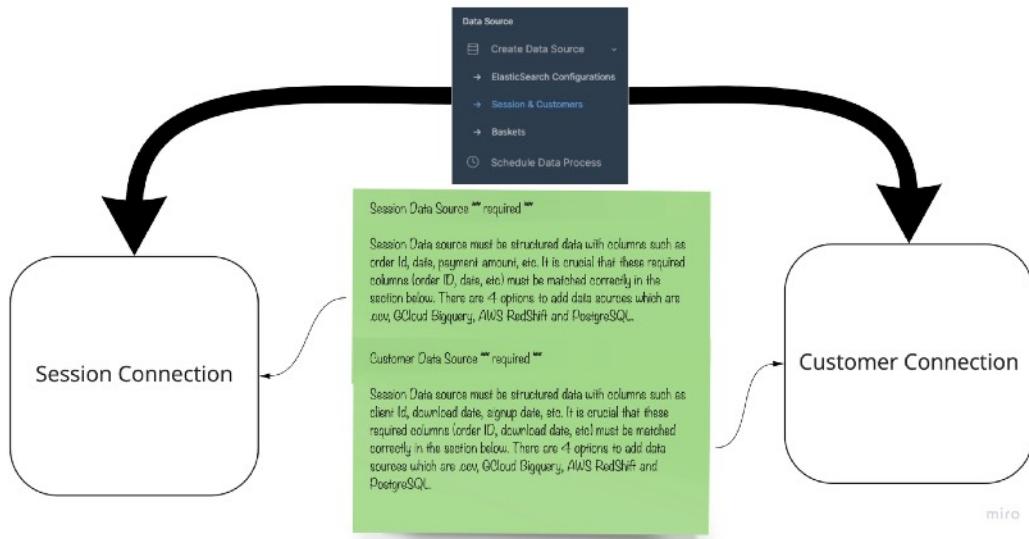
At ElasticSearch, Session and Products Data are stored in the same Index which is ***orders index***. In orders index, There is a ***basket object*** that refers to the products of the related order ID. This data set is transferred from the Products Data Source. There is an ***actions object*** that refers to actions columns of the given session. When Sessions Data Source is created, actions are able to be included.



Another index is stored with customers of unique information that is called ***downloads index***. This index is shaped with unique customers of dates from 1st interaction with the business to further interactions. Each document at the downloads index represents the unique customer. The first interaction of each customer might be the downloaded date or signup date. If there isn't any downloaded date, it must be a date that indicates the first interaction date before the first sessions of each customer.



After the platform is connected to the ElasticSearch successfully, the **Data Sources Connection** process is eligible to connect to any data access.



Sessions Connection is the main responsibility of the data processed. See the example of the Session data set below in the [**Example of Session Data Source Columns Matching Process**](#). There are 2 parts to the connection process. First, Session Data Source of columns and ElasticSearch orders Index of fields must be matched. The main concept of matching is assigning the columns name to the orders index fields on Data Source configuration page. There are required columns which are Order ID, client, start date, amount, purchase. There are also optional columns which are date, discount, promotion, dimension, and actions. Second, Data Source connection type and other information such as user, pw, host, port DB. After connecting the data source, it is able to be changed when it is needed. See the Sessions Connection process below.

Order ID, represents unique ID for each session that is purchased/non-purchased with string type.

Client, represents the customer ID who creates the purchased/non-purchased session with the string type. Client values must be stored at Customers Data Source and be transferred to the downloads index.

Start Date, is the date when the session is started with date type and yyyy-mm-dd hh:mm format.

Amount, is the value of the purchased sessions with float type. If the session is non-purchased, it is directly assigned as 0.

Discount, is the value of discount at the purchased sessions with float type. If the session is non-purchased, it is directly assigned as 0.

Purchase, represents whether the session is ended with the purchased or non-purchased. It is represented with the boolean format.

Date, represents the end date of the purchased/non-purchased session with date type and yyyy-mm-dd hh:mm format.

Promotion, represents the promoted the purchased sessions with string type. It is optional. When it is assigned A/B Test for Promotion Analysis is enabled. Each order might have promotion ID which is an indicator of the organic/inorganic order. It would be more efficient if both promotion ID and discount amount are assigned to the same data set.

Dimension, helps for grouping data sets when it is needed. While you need to split the data set you need to assign a dimensional column to the session data set. This process will be directly added to the ElasticSearch orders (sessions) index as a field and you may choose dimensions as filters in each dashboard in the filter sections.

Actions, represents key points at each session that are good to be added to the data set. It could be **add-to-basket**, **purchase-screen**, **remove-product-from-basket**, etc. This might help us to see how customers engage with the business and which process might give them an idea to leave the session or end it with the purchase.

amount ** required **
value of purchase (float/int). If the order is not ended with purchase, please assign None, Null ' '.

Order ID ** required **

This indicates a unique ID for each session (purchased/non-purchased sessions). Please assign a column name that refers to Order ID at your data source. You need to write the correct column name in the box.

Sessions

- Each session ID must be unique. Sessions must cover both non-purchased and purchase transactions.
- Session Start Date (Whatever you will give as a column) must be a convenient date format.
- The payment Amount for each session transaction must be assigned as a Null value.
- There must be a purchased (True/False) column which is required.
- Discount Amount and session end date columns are optional.

also in the data set for Customers Data Source
It is good to be added. Funnels, Anomaly is
be assigned. No need to add as an action fo
be added. A/B test P
instance, you would like
and Analysis processes.
Source.

discount

discount of purchase (float/int). If the order is not ended with purchase, please assign None, Null ' '.

purchase ** required **

True/False. If it is True, session ends with purchased.
If it is False, session ends with non-purchased.

client ID ** required **

ID for a client this column can connect with client ID (client_2) are customer_fields.

Name

main_session connection

order ID

client

start date

end date

amount

discount

purchase

start date ** required **

eligible date format (yyyy-mm-dd hhmm). This indicates a related session of start date.

end date

eligible date format (yyyy-mm-dd hhmm). This indicates a related session of the end date.

Dimensional Column Name In Data Source

This indicates the dimension column name in data source. It allows us the categorizing data set.

Action Column Name In Data Source

This indicates actions such as add-to-basket, order purchase screen, etc during the sessions.

data source **required **

There are 4 types of data sources; .csv, AWS RedShift, GCloud Bigquery, PostgreSQL

Data Query or Path **required **

If data source is .csv it must be .csv file path, Otherwise, copy sql code here.

Dimensional Column Name in Data Source

Dimensional Column will be your grouping column in Exploratory Analysis, Machine Learning Processes, Dashboard, and Charts. This will also create a new ElasticSearch Index per unique Dimension on that column of values.

location, region, order_type, order_category, e.g.

Action Column Names in Data Source

Action Columns are typed below with comma separated for column name on the data set. Each Action column must have more understanding of Customers.

has_basket, has_order_screen, e.g.

data source

google big query
aws redshift
CSV

Data Query or Path

SELECT * FROM table / raw_data.csv

database

.csv; -
PostgreSQL; required
AWS RedShift; required
GCloud Bigquery; use
as GOOGLE_APPLICATION_CREDENTIALS json file path

Promotion Column

This indicates promotion ID for each session (purchased sessions). Please assign a column name which refers to Order ID at your data source. You need to write the correct column name in the box.

act
mean

user

.csv; -
PostgreSQL; required
AWS RedShift; required
GCloud Bigquery; -

password

.csv; -
PostgreSQL; required
AWS RedShift; required
GCloud Bigquery; -

port

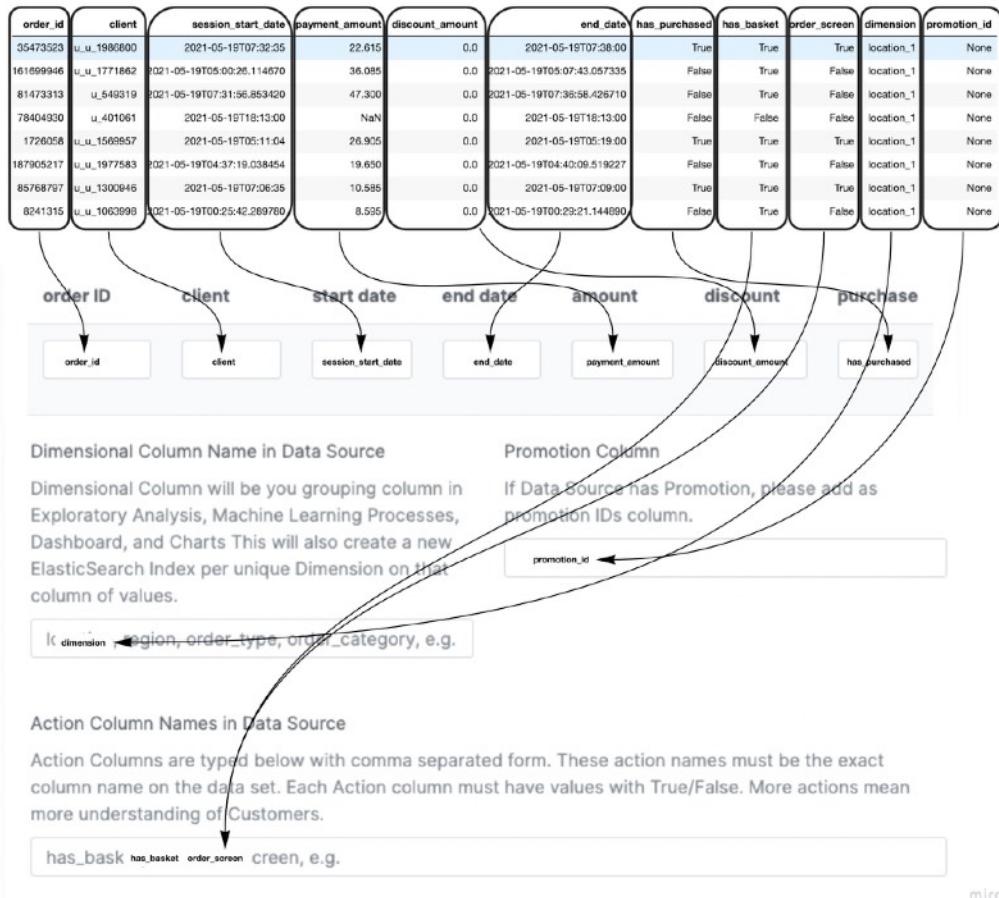
.csv; -
PostgreSQL; required
AWS RedShift; required
GCloud Bigquery; -

host

.csv; -
PostgreSQL; required
AWS RedShift; required
GCloud Bigquery; -

Connect

Example of Session Data Source Columns Matching Process



miro

Customer Connection is a data source related to customers' information such as download date, signup date, etc. Each customer (client field at ElasticSearch) at the Session source must be included at Customer Connections. See the example of Customer data set below in [Example of Customer Data Source Columns Matching Process](#). There are 2 parts of connection process. First, Customers Data Source of columns and ElasticSearch downloads Index of fields must be matched. Main concept of matching is the assigning the columns name to the downloads index fields in Data Source configuration page. There are required columns which are client, download date. Actions and signup date are optional columns. Second, Data Source connection type and other informations such as user, pw, host, port DB. After connecting the data source, it is able to be changed when it is needed. See the Customer Connection process below.

Client, represents unique identification for each customer with string type.

Download Date, represents the date which is the first interaction date to the business with date type and yyyy-mm-dd hh:mm format.

Actions, represents key points at each session that are good to be added to the data set. It could be **payment-amount-add**, **first-session**, etc. This might help us to see how customers engage with the business and which process might give them an idea to leave the session or end it with the purchase. Unlike sessions of Actions, Customers Actions must be with date type and yyyy-mm-dd hh:mm format.

Customers

- Each customer ID must be unique. Customer IDs are eligible to join with the Session of client IDs.
- Download Date is required.
- Signup Date is optional.
- Even there is no download date is available please make sure to assign any date valuable for the business.
- Actions must be date rather than True/False columns just like Sessions Data Source of Actions.
- Actions are optional. But it is good to be added. Funnels, Anomaly is related to Actions.

client ID ** required **

ID for client this column can connect with client ID (client_2) are customer_fields. It must be unique documents (row) per client.

Customer Connection Tag Name ** required **

This name will help you to find the connection on the schedule data process page.

signup date ** required **

It is an optional date column that refers to the signup date or any date indicator as action after downloading.

Customers Connection Tag Name

main_customer connection

Client ID

Download Date

Signup Date

Action Column Name in Data Source

This indicates actions such as first login action, first-order action, first payment method add action order purchase screen, etc during the sessions.

Action Names in Data Source

Action names are typed below with comma separated name on the data set. Each Action column must have understanding of Customers.

Download Date ** required **

It is a date when a client has first interaction with the business. If there isn't a downloaded date you may assign the first interaction date before the first session date. This will help us to create Customers Funnel and From Download to 1st Order Cohort. Make sure the assigned downloaded date is less than the 1st session date per customer.

first_session_start_date, first_order_date, e.g.

data source ** required **

There are 4 types of data sources; .csv, AWS RedShift, GCloud Bigquery, PostgreSQL

data source

google big query
aws redshift
csv

Data Query or Path

SELECT * FROM table / raw_data.csv

Data Query or Path ** required **

If data source is .csv, it must be .csv file path. Otherwise, copy sql code here.

Connect

database

.csv; -
postgreSQL; required
AWS RedShift; required
GCloud Bigquery; use
as GOOGLE_APPLICATION_CREDENTIALS json file path

user

csv; -
postgreSQL; required
AWS RedShift; required
GCloud Bigquery; -

password

csv; -
postgreSQL; required
AWS RedShift; required
GCloud Bigquery; -

port

csv; -
postgreSQL; required
AWS RedShift; required
GCloud Bigquery; -

host

csv; -
postgreSQL; required
AWS RedShift; required
GCloud Bigquery; -

database

Example of Customers Data Source Columns Matching Process



miro

Products Connection is a data source related to the basket of the sessions with products and it covers information such as product ID/name, price, product of the category. Customers and Sessions Data Sources are required, however, Product Connection is optional. A/B Test Products are related to products data sources. Products data source is transferring to orders index into the baskets object on each sessions document. See the example of Product data set below in

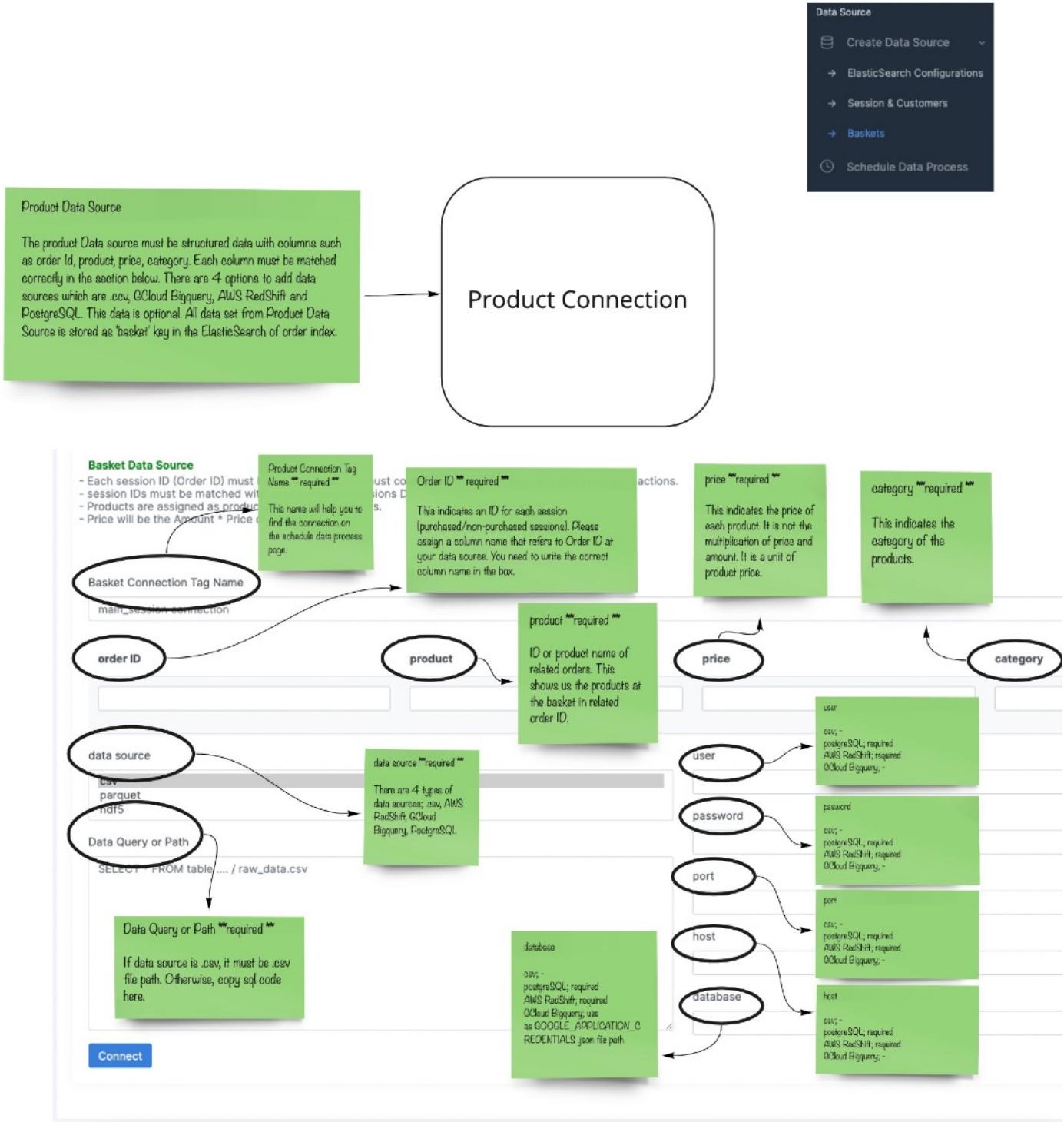
Example of Products Data Source Columns Matching Process. There are 2 parts to the connection process. First, Products Data Source of columns and ElasticSearch orders Index of fields must be matched. There are required columns which are order ID, product ID/name, category, price.

Order ID, indicates an ID for each session (purchased/non-purchased sessions). Please assign a column name that refers to Order ID at your data source. You need to write the correct column name in the box. It is crucial that all order IDs must be matched with order IDs at the session data source. Otherwise missing order IDs wouldn't be matched and can not be transferred to the orders index.

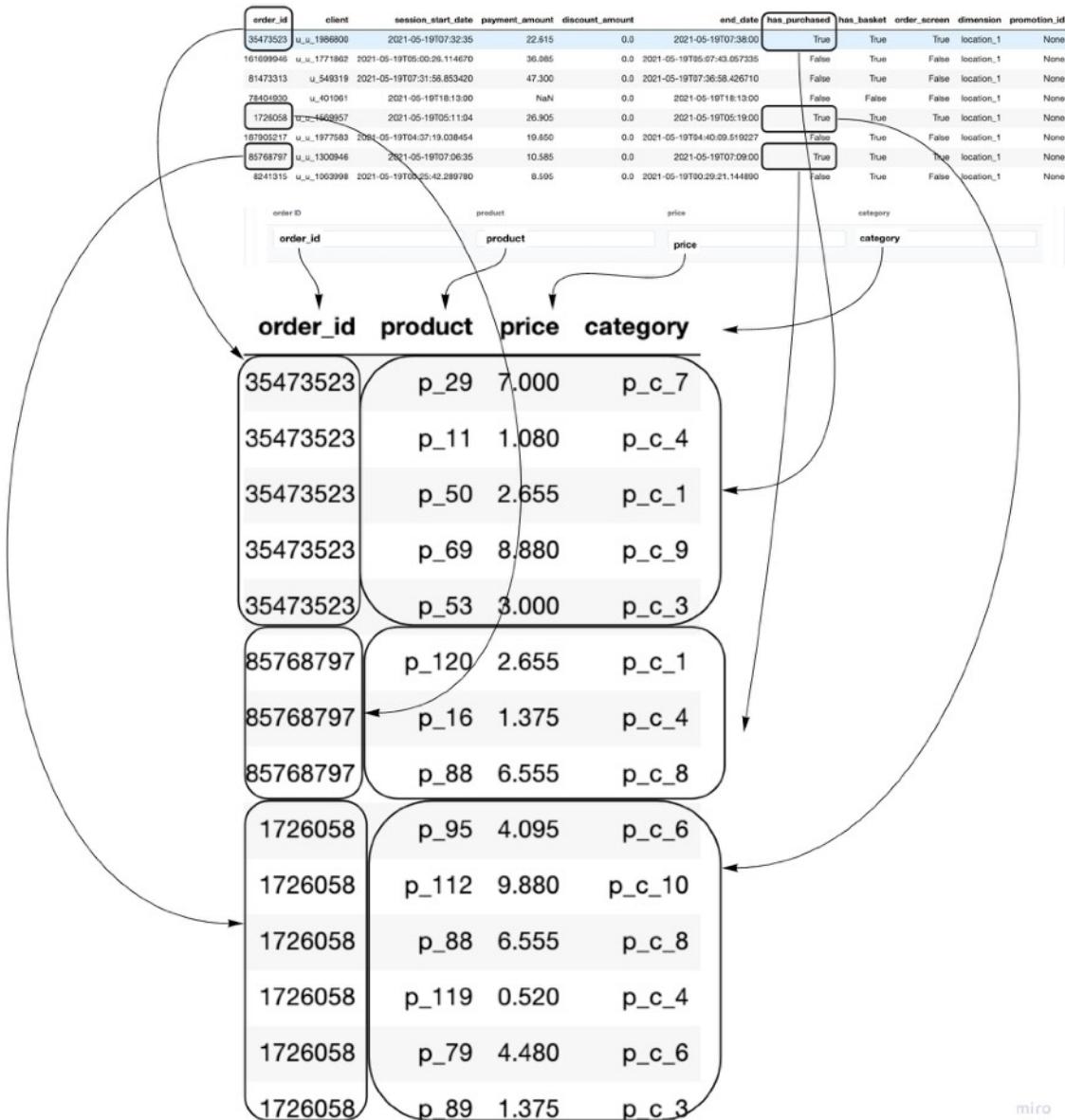
Product ID/Name, represents ID or product name of related orders. This shows us the products at the basket in related order ID.

Price, indicates the price of each product. It is not the multiplication of price and amount. It is a unit of product price.

Category, indicates the category of the products.



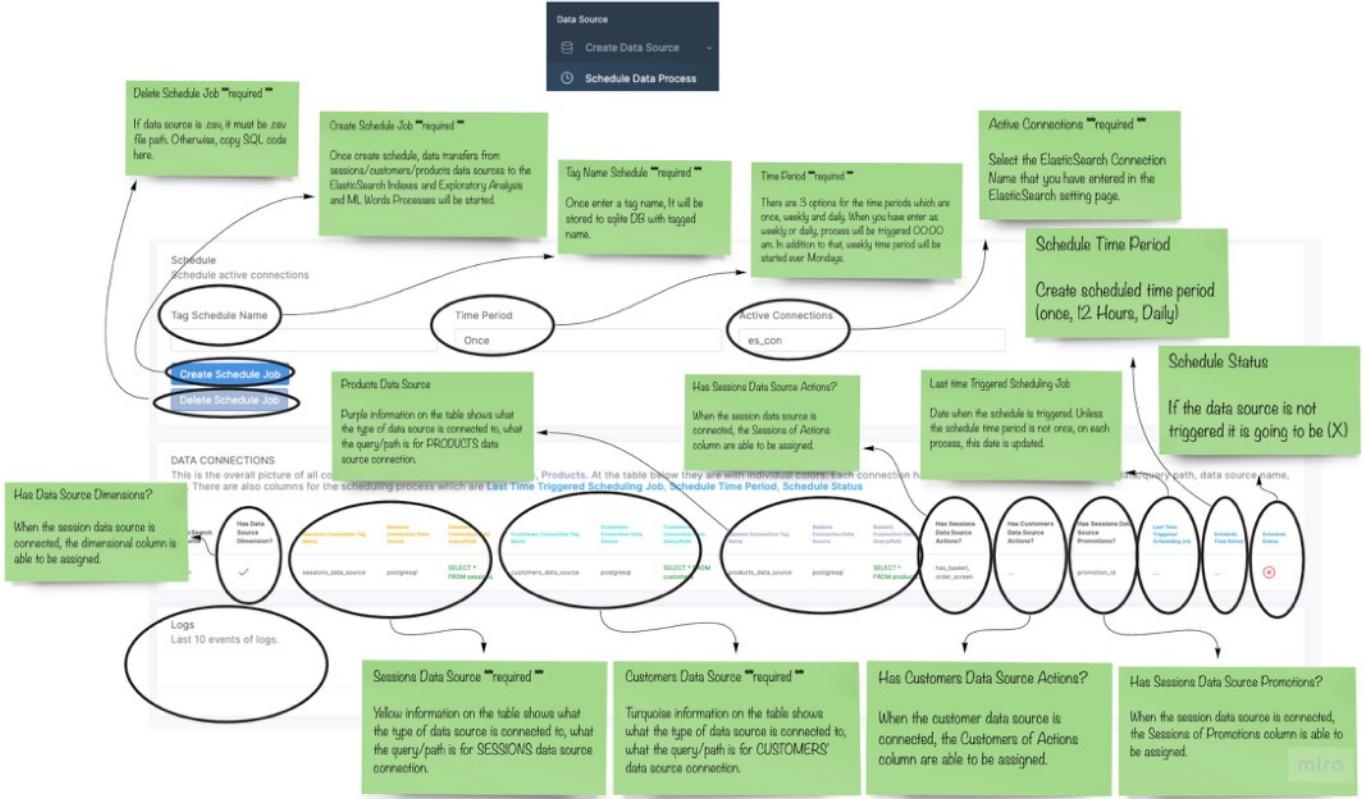
Example of Products Data Source Columns Matching Process



Schedule Data Process is the main console to track the data transferring processes, Exploratory Analysis, and ML Works of scheduling and triggering processes. Once the schedule is created, data transfers from sessions/customers/products data sources to the ElasticSearch Indexes, and Exploratory Analysis and ML Works Processes will be started. If it is aimed at triggering these processes once, choose 'once' for the time period Otherwise process is triggered according to the time period. When you delete the schedule, the process will stop there and never updates the data sets.

There are 3 parts to Schedule Data Process;

1. **Schedule** Assign Tag Schedule Name. Choose Time Period. Create Schedule Job.
2. **Data Connections:** This is the table where you can find Sessions/Customers/Products Connections information and Schedule Process of logs. This table of each cell is assigned while it is connected to the Sessions/Customers/Products.
3. **Logs:** In order to check which process is triggered, logs help us to find which process is running on.



C. Exploratory Analysis

CustomersAnalytics provides us some crucial and popular Analysis with individual dashboards and charts. These are Funnels, Cohorts, Descriptive Stats, Product Analytics, Churn Analysis, Promotion Analytics.

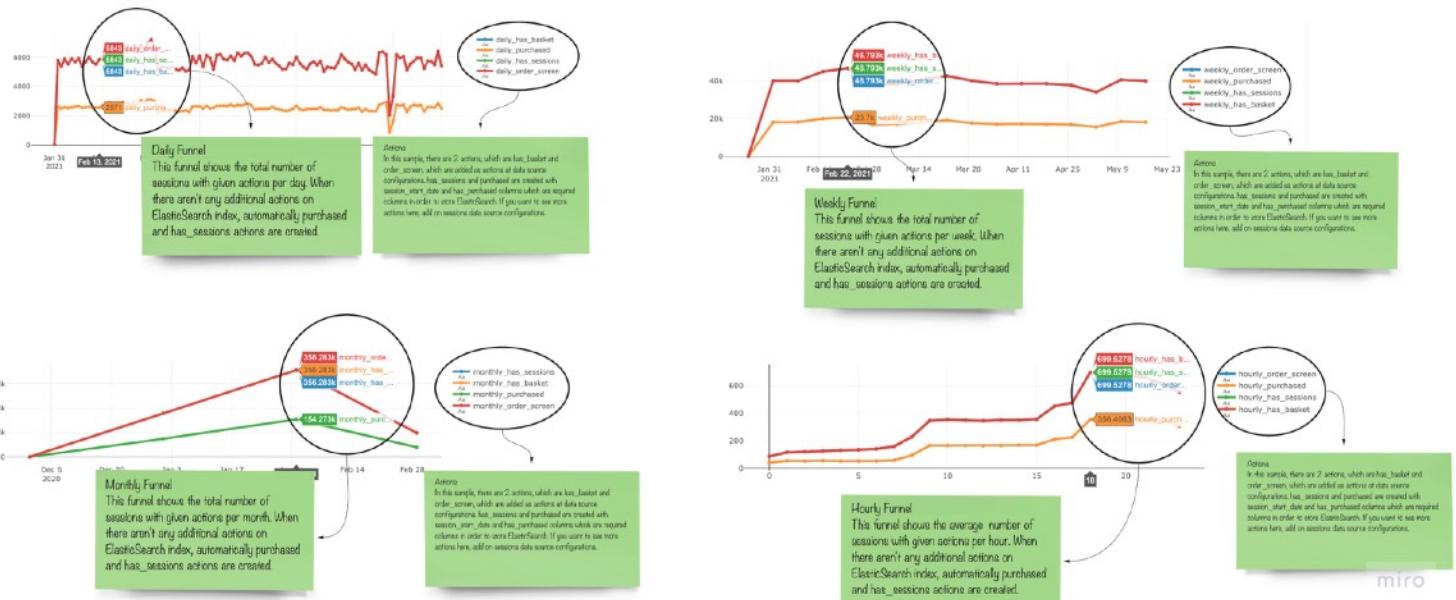
1. Funnels

They allow you to see where your potential customers fall within your desired goal flow and enable you to track and measure conversion rates for each step on that path. Funnel at CustomersAnalytics allows us to see conversion from 2 main parts. First One is the Session Funnels which covers from session start event to purchase event. If there is an action between session start and purchase event, these are also added to the Funnels as Action. These actions must be added to Session data source while it is created, Second one is the Customers Funnels which covers the customers of events and actions. This type of funnel covers all events from download date to session start date. If there are additional actions, they must be added as actions while Customers Data Source is created.

a. Sessions of Actions Funnel

Session start and purchase events are by default actions for this type of funnel. This funnel shows the values from session to purchase. At the ElasticSearch orders index, It is the 'actions' object that gives us the clue of the Sessions of Actions. Funnels count the True values of each metric at the 'actions' objects. If there are additional actions, they are also stored in 'actions' object. Each action has been calculated related to time periods. This process is an aggregation of related order id according to time periods. There are 4 time periods; monthly, weekly, daily, hourly.

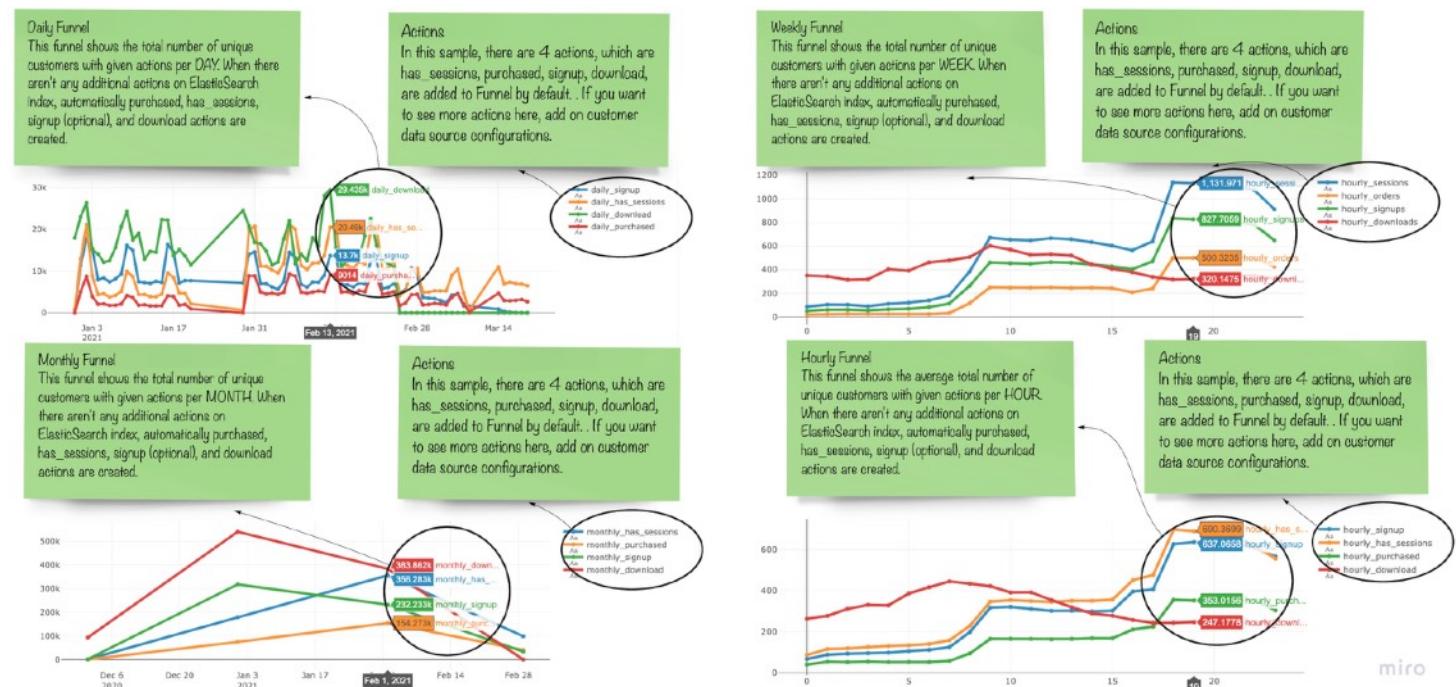
- i. **Daily Funnel:** Total order count per action (session and purchase actions included) per day
- ii. **Weekly Funnel:** Total order count per action (sess. and purch. actions included) per week
- iii. **Monthly Funnel:** Total order count per action (sess. and purch. actions included) per month
- iv. **Hourly Funnel:** Average total order count per action (sess. and purch. actions included) per hour



b. Customers of Actions Funnel

Customers Funnels are aggregated count of events per customer according to time periods. They are started to be counted from the user's first engagement events such as download or signup. Customers Funnels also covers first Sessions start and first purchase event, For instance, daily funnel for Feb. 12, 2021, There are 29.435K customers who have downloaded, 20.49K customers who have their first sessions, 13.7K customers who have their signups, and 9K customers who have their purchase. Just like Sessions of Actions funnels, there are 4 types of Funnels related to time periods which are daily, weekly, monthly, and hourly.

- Daily Funnel:** Total customer count per customer of action (download and purchase actions included) per day
- Weekly Funnel:** Total customer count per customer of action (download and purchase actions included) per week
- Monthly Funnel:** Total customer count per customer of action (sess. and purch. actions included) per month
- Hourly Funnel:** Average total customer count per customer of action (sess. and purch. actions included) per hour



2. Cohorts

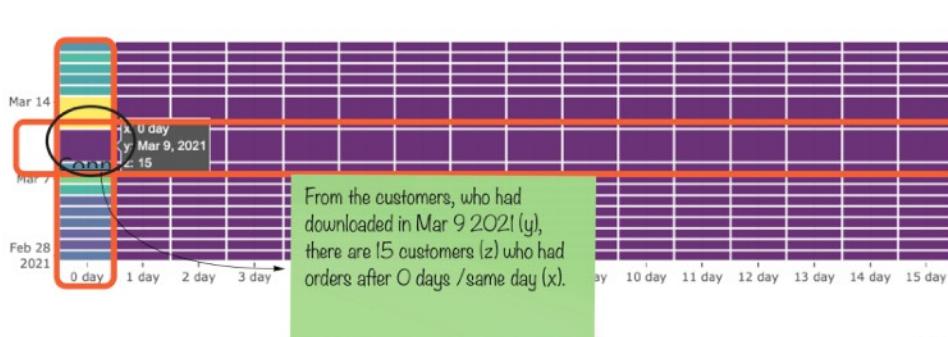
Cohorts are aggregated counts of customers who are acting in the same way at a given time period. In our study, cohorts cover the number of customers who have the same transaction in the given time period and upcoming days/weeks. At each cohort y-axes represent the date of the first action, x-axes represent the next action after x days/weeks. For instance, let's check the Daily Download to 1st Order Cohort. Y-axis is the y Downloaded Date and the x-axis is the x day after downloaded. if y; Feb 28, 2021, and x; 10 and z; 10 customers who had their 1st orders. From the customers, who had downloaded on Feb 28, 2021 (y), there are 10 customers (z) who had orders after 10 days (x). There are 2 types of time periods for cohorts which are daily and weekly. According to the time period, the y-axis changes. There is 4 kind of Cohorts which are Download to 1sr Order, From 1st to 2nd Order, From 2nd to 3rd Orders, From 3rd to 4th Orders. Here are the kind of Cohorts;

- **Download to First Order Cohort:** Download to First Order Cohort If there isn't any downloaded date, you may assign any date which is related to customers of the first event with your business. This cohort of date column represents download date. If it is Weekly Cohort it will represent the Mondays of each week. Otherwise, it will represent days. Each Numeric column from 0 to 15 is the day difference after the downloaded date. For instance, if the date column is 2021-05-06, and the numeric column is 10 and the value is 100, this refers that there are 100 customers who have downloads in 2021-05-06 and have first orders 10 days later.
- **First to Second Order Cohort:** This cohort of date column represents the first order date. If it is Weekly Cohort it will represent the mondays of each week. Otherwise, it will represent days. Each Numeric column from 0 to 15 is the day difference after the first order date. For instance, if the date column is 2021-05-06, and the numeric column is 10 and the value is 100, this refers that there are 100 customers who have first orders in 2021-05-06 and have second orders 10 days later.
- **Second to Third Order Cohort:** This cohort of date column represents second order date. If it is Weekly Cohort it will represent the mondays of each week. Otherwise, it will represent days. Each Numeric column from 0 to 15 is the day difference after the second order date. For instance, if the date column is 2021-05-06, and the numeric column is 10 and the value is 100, this refers that there are 100 customers who have second orders in 2021-05-06 and have third orders 10 days later.
- **Third to Fourth Order Cohort:** This cohort of date column represents the third order date. If it is Weekly Cohort it will represent the mondays of each week. Otherwise, it will represent days. Each Numeric column from 0 to 15 is the day difference after the third order date. For instance, if the date column is 2021-05-06, and the numeric column is 10 and the value is 100, this refers that there are 100 customers who have third orders in 2021-05-06 and have fourth orders 10 days later.

a. Daily/Weekly Order Cohort

There are 2 types of time periods for cohorts which are daily and weekly. According to the time period, the y-axis changes.

i. Daily Download to 1st Order Cohort



This cohort enables us to see the **daily** engagement of the customers and the time period between each customer of download and 1st order.

ii. Daily Cohort From 1st to 2nd Order



This cohort enables us to see the **daily** engagement of the customers and the time period between each customer of 1st order and 2nd order.

iii. Daily Cohort From 2nd to 3rd Order



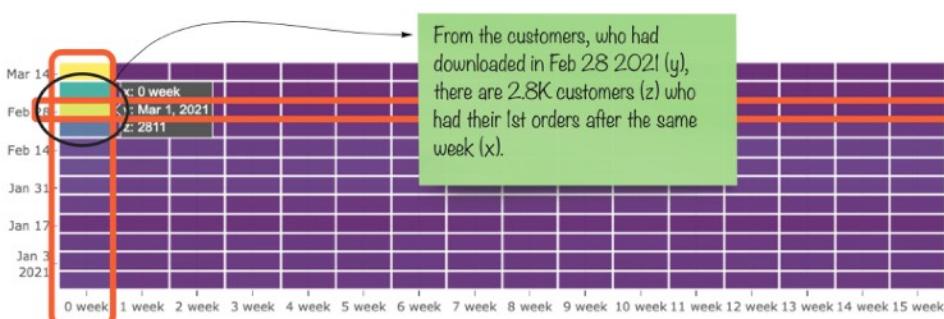
This cohort enables us to see the **daily** engagement of the customers and the time period between each customer of 2nd order and 3rd order.

iv. Daily Cohort From 3rd to 4th Order



This cohort enables us to see the **daily** engagement of the customers and the time period between each customer of 3rd order and 4th order.

v. Weekly Download to 1st Order Cohort



This cohort enables us to see the **weekly** engagement of the customers and the time period between each customer of download and 1st order.

vi. Weekly Cohort From 1st to 2nd Order

This cohort enables us to see the **weekly** engagement of the customers and the time period between each customer of 1st order and 2nd order.



vii. Weekly Cohort From 2nd to 3rd Order

This cohort enables us to see the **weekly** engagement of the customers and the time period between each customer of 2nd order and 3rd order.



viii. Weekly Cohort From 3rd to 4th Order

This cohort enables us to see the **weekly** engagement of the customers and the time period between each customer of 3rd order and 4th order.



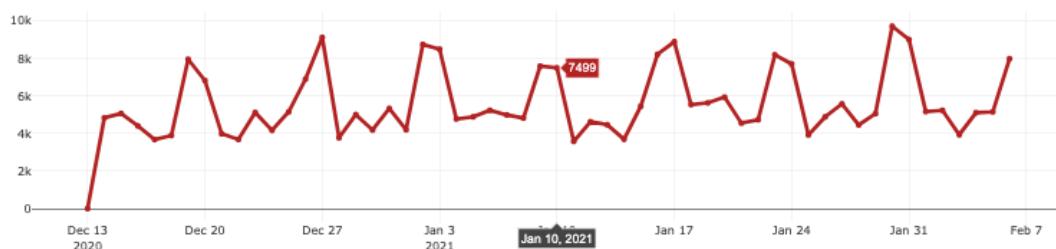
3. Descriptive Stats

Descriptive Stats are some overall values that need to check each day for businesses. There are two main dashboards for Descriptive Stats which are Stats Purchase and Descriptive Statistics. In addition to that, customerAnalytics Overall and Customers Dashboards, search dashboards related to promotion, clients have some charts related to Descriptive Stats of Analysis.

a. Stats Purchase

The dashboard covers the total orders per day/week/hour/month. It enables us to see the pattern of the order counts such as rush hours week day/weekend, summer/winter time.

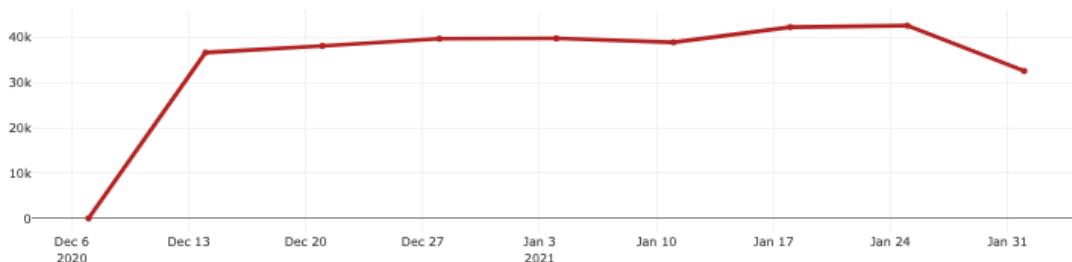
- i. **Daily Orders:** The total number of orders per day. Total number of unique session IDs which has has_purchase = True breakdown with day.



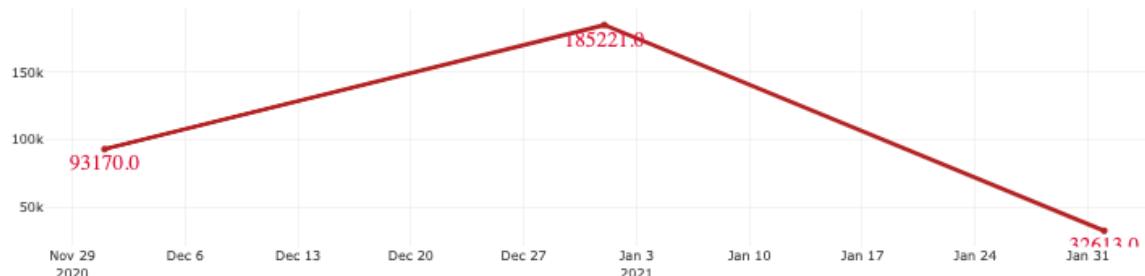
- ii. **Hourly Orders:** Average total order count per day. The total number of order counts per day and per hour is aggregated. The next step is calculating the average order count per hour by using aggregated data from the previous step.



- iii. **Weekly Orders:** The total number of orders per week. Each week is represented by Mondays.

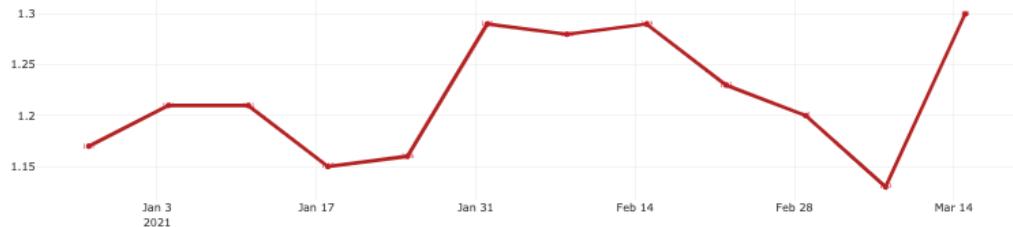


- iv. **Monthly Orders:** Total number of orders per month; the total number of purchase unique session IDs are counted per month.

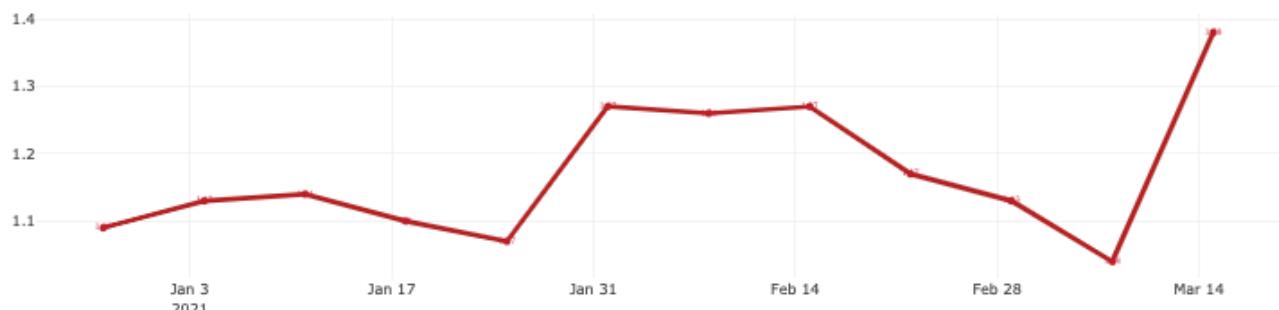


b. Descriptive Statistics

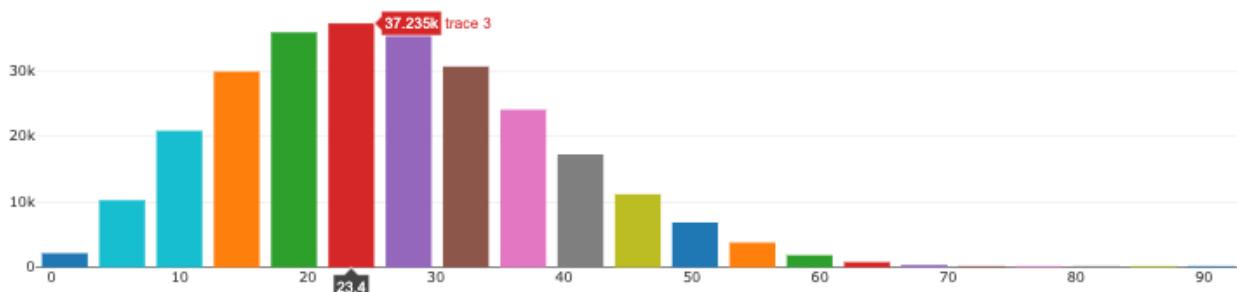
- i. **Weekly Average Session Count per Customer:** Each customer's total unique session count per week is calculated. By using this aggregated date average session count per customer per week is calculated. This might give us the clue of which weeks the customers of engagement to the business are increased/decreased. In addition to that, It gives us information about the customers' session frequency per week.



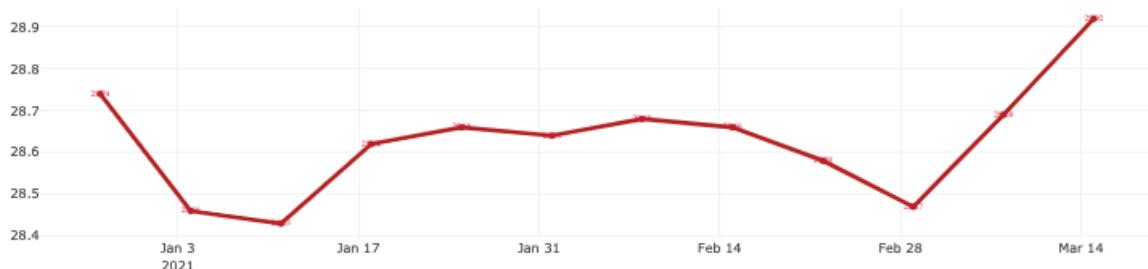
- ii. **Weekly Average Purchase Count per Customer:** Each customer's total order count per week is calculated. By using this aggregated date average order count per customer per week is calculated. This might give us the clue of which weeks the customers of order possibilities are increased/decreased. In addition to that, It gives us information about the customers' purchase frequency per week.



- iii. **Payment Amount Distribution:** This chart is a histogram which shows the bins of total customers. Each bin represents a range of payment amounts related to purchased orders.



- iv. **Weekly Average Payment Amount:** Average payment amount of orders user who has the purchased orders.



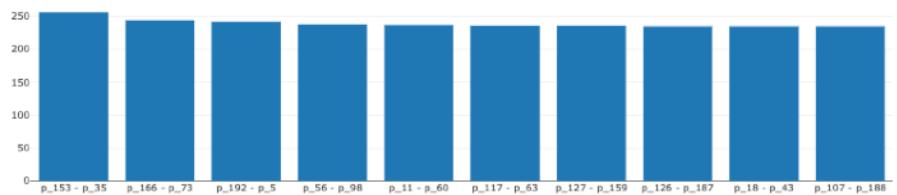
4. Product Analytics

When Product Data Sources is added and the data is transferred to the ElasticSearch, Product Analytics Charts are able to be updated. These give the basic idea of the products and their categories of purchase processes. These charts help us figure out the relation between products and the customers of order probability.

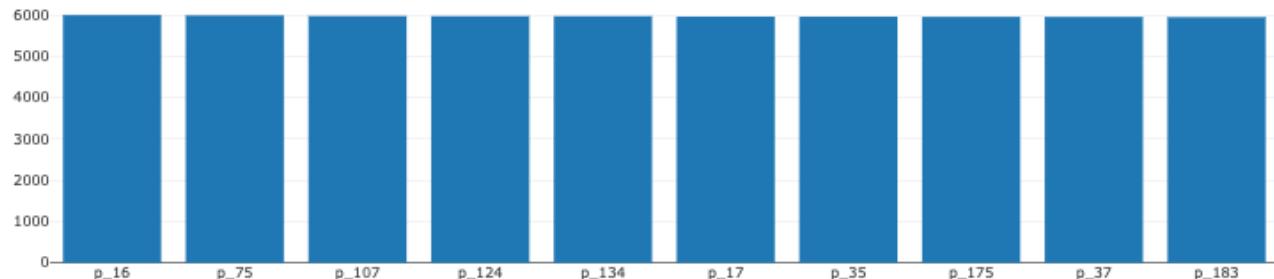
a. Top Purchased Products:

i. Most Combined Products:

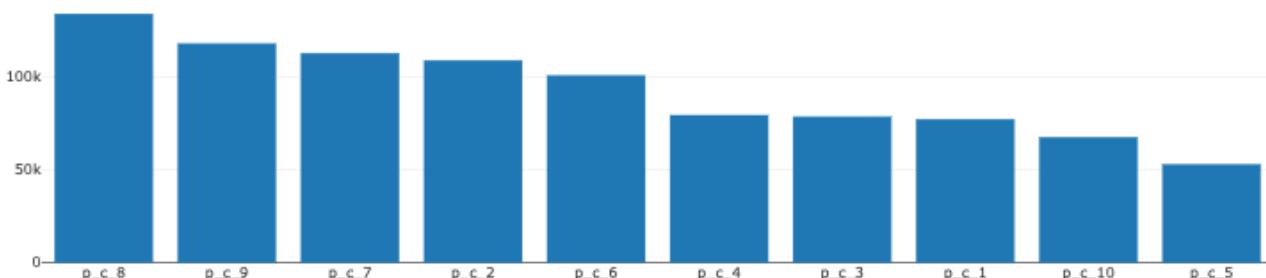
Products: The most preferred products for the customers. Each bar represents the total number of orders per product. According to the example below, There are 250 orders which have both p_153 and p_166 in their baskets.



ii. Most Purchased Products: The most preferred product categories for the customers. Each bar represents the total number of orders per product category.



iii. Most Purchased Categories: Each product pairs have counted an order and the Total number of order count per product pairs are calculated.



D. Machine Learning Works

Additional to Exploratory Analysis, ML Works enables us the more data investigation with deeper methods in the CustomerAnalytics. These methods are popular on Customer Analytics Data Scientist teams which are A/B Test, Customer Lifetime Value Prediction, Anomaly Detection, and Customer Segmentation. These complex methods will help the business in decision-making.

At each ML Works part, There is an individual dashboard with charts. Customer Segmentation is presented with RFM values which are Recency, Monetary, Frequency. CLV Prediction predicts the next 6 months of daily total payment amounts and Segmented Customers of total payment amounts. A/B Test compares before and after the time periods with metrics which are order

counts or average payment amounts for products and promotions. It gives results about how well the product improves the order count or average payment amount.

There are Machine Learning Algorithms that help us to calculate these ML processes. Customer Segmentation classifies the customers with K-Means, CLV Predictions creates results with LSTM & Conv NN, A/B Tests creates tests by finding the statistical distribution of the data sets (Poisson, Binomial, Gaussian, Gamma, Beta) and applies Hypothesis in order to calculate results. Anomaly Detections finds the abnormal data points with Deep Learning AutoEncoder.

1. A/B Tests

At Customers Analytics, A/B Test compares the before - after the metrics. There are 3 types of A/B Tests which are Products Comparisons, Promotions Comparisons, And Customer Segments Comparisons. These comparisons are measured by 2 metrics individually which are Purchase Order Count and Average Payment Amount.

Products and Promotion Comparisons are applied similar way. They are applied for each product/promotion of usage before and after time periods. Each customer of order with product/promotion is assigned as 0 points. Before 0 points of orders are compared with after 0 points of Orders. Assigning as 0 point process is applied for the customers of each product/promotion usage order.

Unlike Products and Promotion Comparisons at Segment Comparison, Before and After time periods are constants time periods which are day, week, and month. Each segment of the current day, current week, and current month are compared with the last 4 weeks of the same week day of the current week day, the last 4 weeks, and the last 4 months.

a. A/B Test Promotion

A/B Test Promotions are a bunch of tests that compare the effect of Promotions on the business. There are two metrics that help us to measure comparison values. These are the Number of orders counts per promotion and the Average purchase amount per promotion. Promotion of Usage related to these metrics are tested with Before-After Analysis in order to answer the question is "Is there any significant increase/decrease on Order Count / Average Payment Amount per User after the Promotion is used?" While we need to see to Promotion which increases/decreases the number of orders and the average purchase amount, This Section can help us to find the right promotion.

- Hypothesis are;
H₀ : There is no significant Difference between A - B of order count/purchase amount.
H₁ : There is a significant Difference between A - B of order count/purchase amount.
- Bootstrapping Method;

Each test process is applied with Bootstrapping Method. So, Iteratively randomly selected A and B sample customers of order count / average purchase amount will be tested acceptable enough time. Each iteration A - B samples are randomly selected. The accepted ratio will show us the confidence of our test results. For instance, there are 100 tests and 75 of them are ended with H₀ Accepted. So, tests are % 75 our tests are H₀ Accepted.

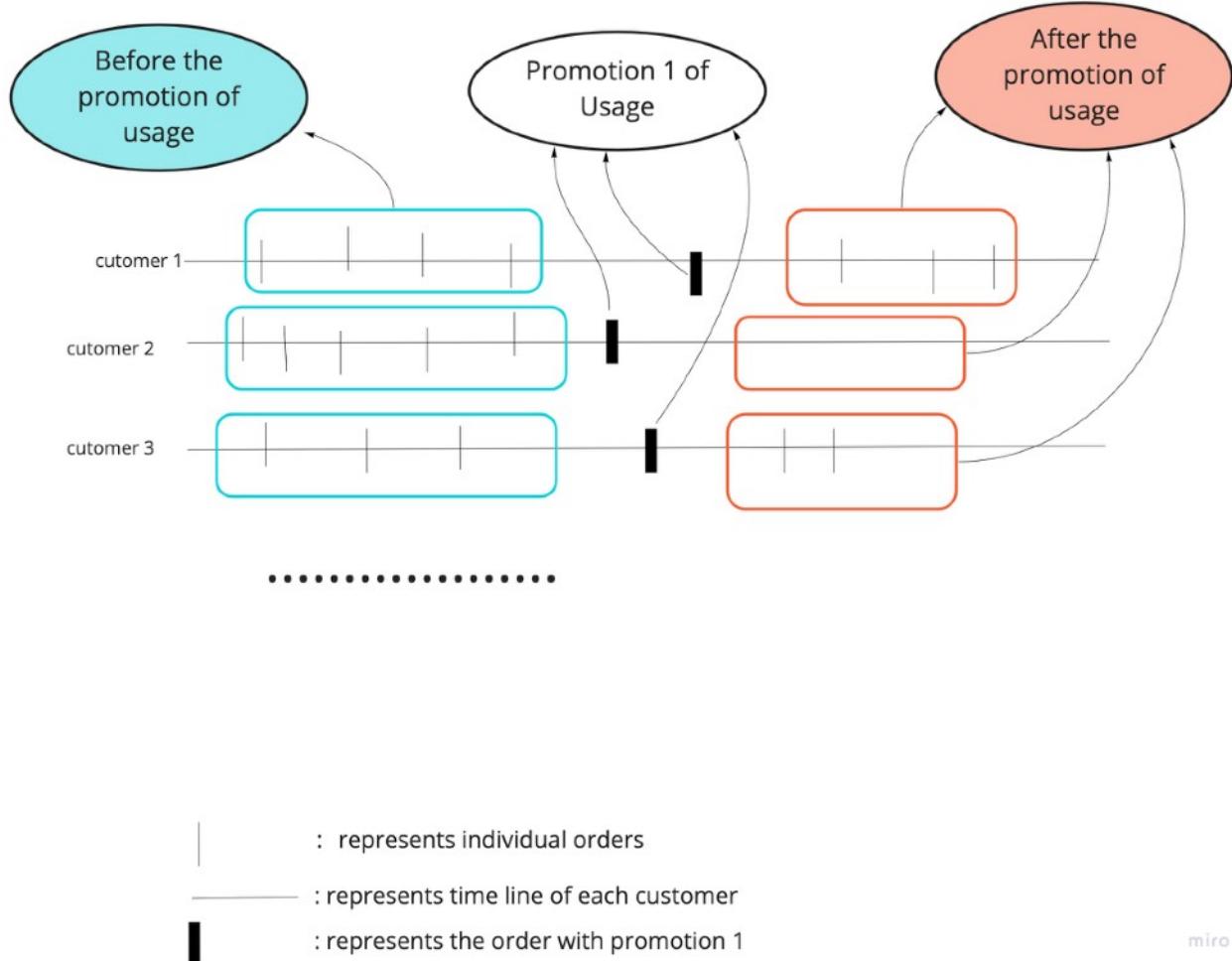
- How are Promotions Compared?

Each combination of promotions is tested individually. The average of Purchase Payment Amount per User Sample for Promotion (1) is compared to the Average of Purchase Payment Amount per User Sample for Promotion (2).

- How are Before - After Promo Usage Tests designed?

It might be very confusing and it might seem a very sophisticated A/B Test process, however, it basically compares promotions related to customers of usage. Each promotion of usage is tested separately. Each promotion of used users and their used timestamp are collected for each promotion. Each customer of one week before promotion usage and one week after the promotion use time periods are compared with to metrics which are order count and average

purchase amount. We aim to detect a significant increase after the time period while comparing to before the time period about purchase amount or order count.



i. **Promotion Comparison:**

Y-axis represents the total number of Test Accepted Count. The X-axis represents Test Reject Ratio. Each dot on the scatter plot represents each promotion. This is a comparison of each promotion pair. Each promotion pair is tested and Each dot represents 1st promotion of each promotion pair. More valuable Promotions are the dots that have less Reject Ratio and more Accept Count.

At Promotion Comparison, each promotion of usage per customer is collected. These customers of orders before and after the used promotion of orders are compared. The average payment Amount per customer is compared. For instance, there are 1K customers who used promotion p_1. From the 1K population, the average payment amount per customer is randomly sampled 100 times and tested. In the end, the number of Null Hypotheses of Reject and Accept test count is calculated. If there is a significant increase in the average payment amount after the p_1 usage, the promotion would be selected as Perfect Promotion.

At this instance below, there is 40 test which says this promotion did not affect the payment amount after the usage of promotion. In addition to that 36 % of the tests are rejected which means that they have resulted in a significant increase in the payment amount.

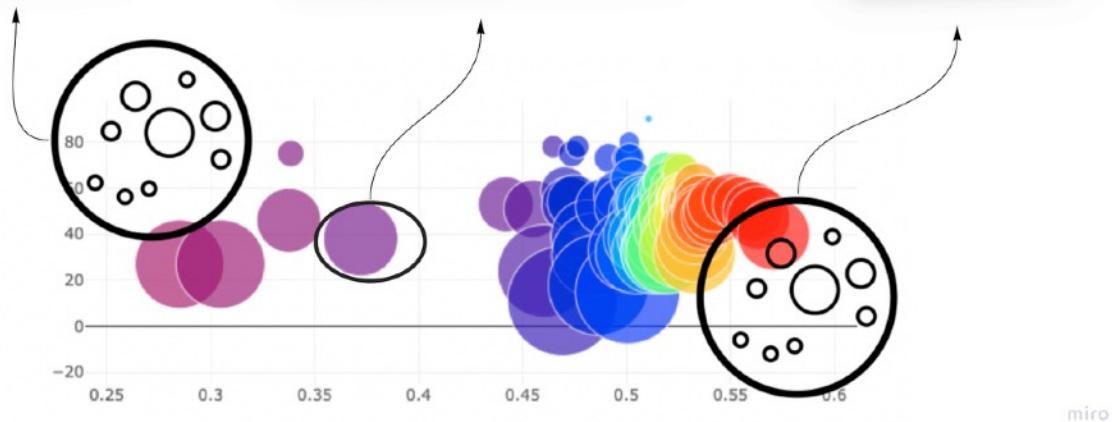
Worst Promotions

These promotions have low Null Hypothesis Reject Ratio and a high number of Null Hypothesis Accepted Tests

This promotion of Usage is tested. Each promotion of usage per customer is collected. These customers of orders before and after the used promotion of orders are tested. At this instance below, there is 40 test which says this promotion didn't increase the payment amount after the usage of promotion. In addition to that 36 % of the tests are rejected which means that they have resulted in a significant increase in the payment amount.

Perfect Promotions

These promotions have a high Null Hypothesis Reject Ratio and a low number of Null Hypothesis Accepted Tests



miro

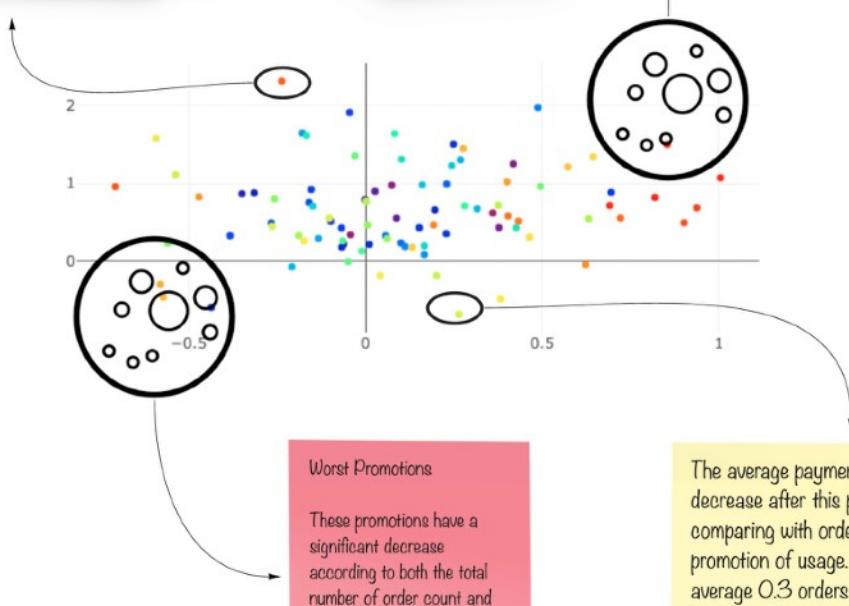
ii. Order and Payment Amount Difference for Before and after Promotion Usage

Y-axis represents the difference in Average Payment Amount per customer. The X-axis represents the difference in the Average Purchase Count per customer. Differences are calculated per user and per promotion. Differences are calculated between customers of purchase before and after using the promotion. Each dots represents the promotion with the difference in Average Payment.

The average payment amount has 2.4 increase on after this promotion of usage while comparing with orders that are before the promotion of usage. In addition to that average of 0.3 orders decrease are also detected for this promotion.

Perfect Promotions

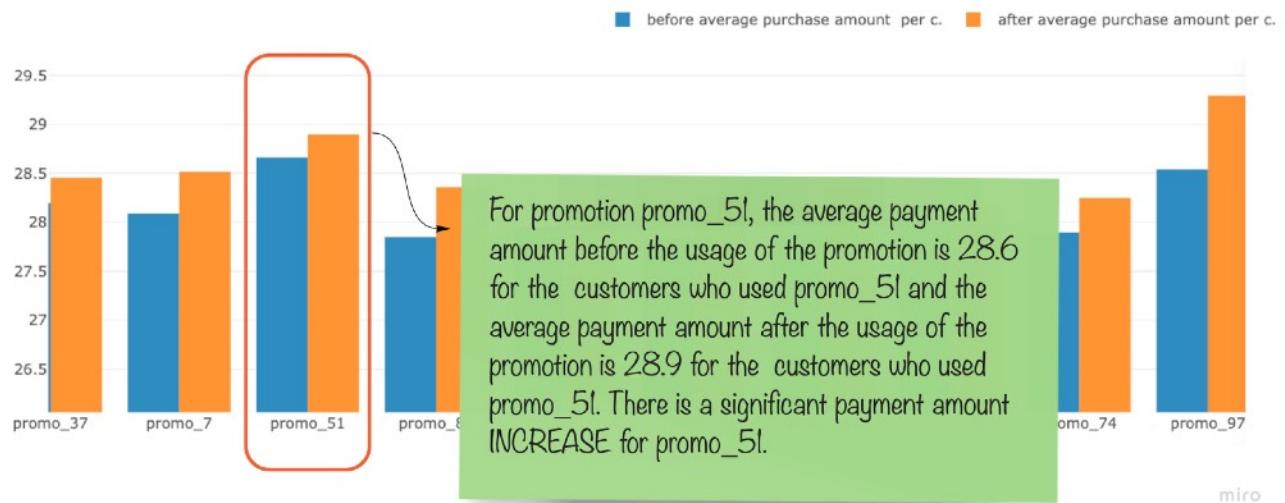
These promotions have a significant increase according to both the total number of order count and average payment amount.



The average payment amount has a 0.4 decrease after this promotion of usage while comparing with orders that are before the promotion of usage. In addition to that average 0.3 orders increase are also detected for this promotion.

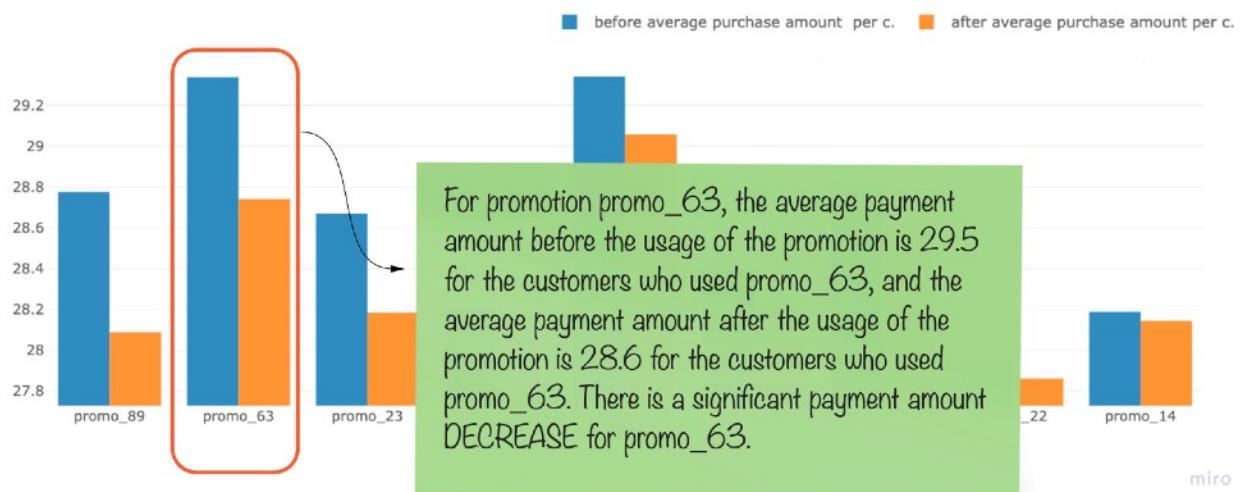
iii. Before - After time Periods customers' Average Purchase Payment Amount Test (Test Accepted!)

Average Payment Amount per customer is calculated according to before and after the usage of the promotions. The assumption is “There is a significant payment amount INCREASE after the promotion of usage”. So, the accepted Null Hypothesis according to this assumption is shown in the chart below.



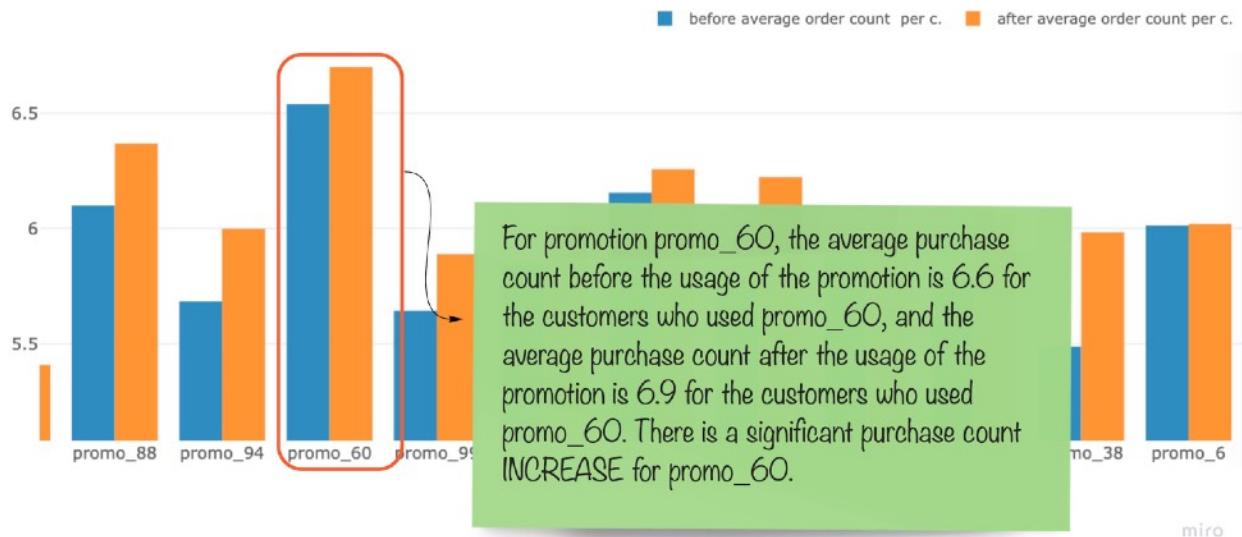
iv. Before - After time Periods customers' Average Purchase Payment Amount Test (Test Rejected!)

Average Payment Amount per customer is calculated according to before and after the usage of the promotions. The assumption is “There is a significant payment amount DECREASE after the promotion of usage”. So, the accepted Null Hypothesis according to this assumption is shown in the chart below.



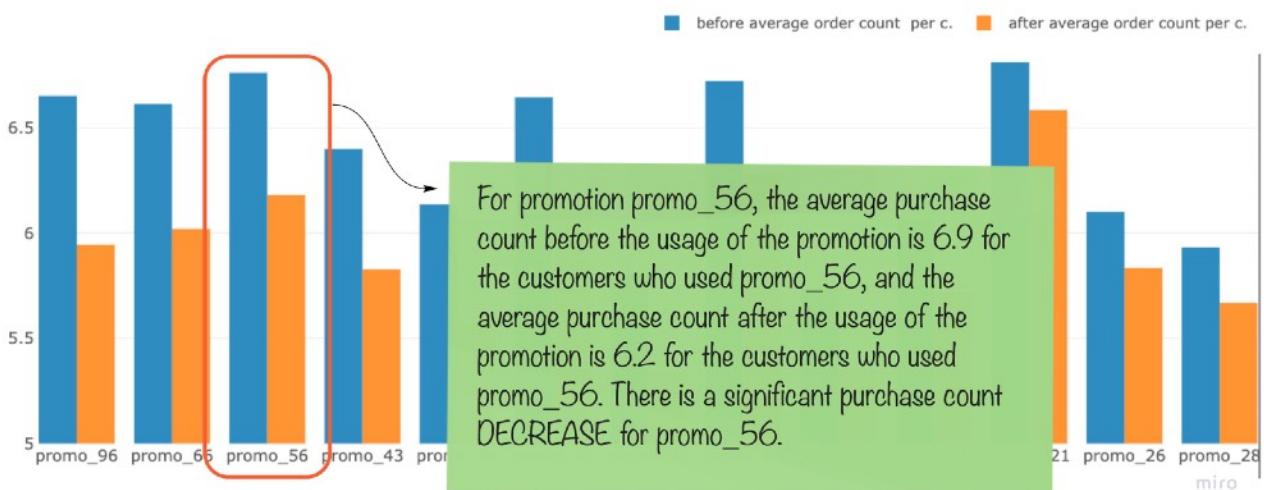
v. Before - After time Periods customers' Total Purchase Count Test (Test Accepted!)

The total purchase count per customer is calculated according to before and after the usage of the promotions. The assumption is “There is a significant purchase count INCREASE after the promotion of usage”. So, the accepted Null Hypothesis according to this assumption is shown in the chart below.



vi. Before - After time Periods customers' Total Purchase Count Test (Test Rejected!)

The total purchase count per customer is calculated according to before and after the usage of the promotions. The assumption is “There is a significant purchase count DECREASE after the promotion of usage”. So, the accepted Null Hypothesis according to this assumption is shown in the chart below.



b. A/B Test Product

A/B Test Products are a bunch of tests that compare the effect of Products on the business. There are two metrics that help us to measure comparison values. These are the Number of orders counts per product and the Average purchase amount per product. Products of Purchases related to these metrics are tested with Before-After Analysis in order to answer the question that is "Is there any significant increase/decrease on Order Count / Average Payment Amount per User after the Products is purchased?" While we need to see to the product which increases/decreases the number of orders and the average purchase amount, This Section can help us to find the right product for the business.

- Hypothesis are;

H0 : There is no significant difference between A - B of order count/purchase amount.

H1 : There is a significant difference between A - B of order count/purchase amount.

- Bootstrapping Method;

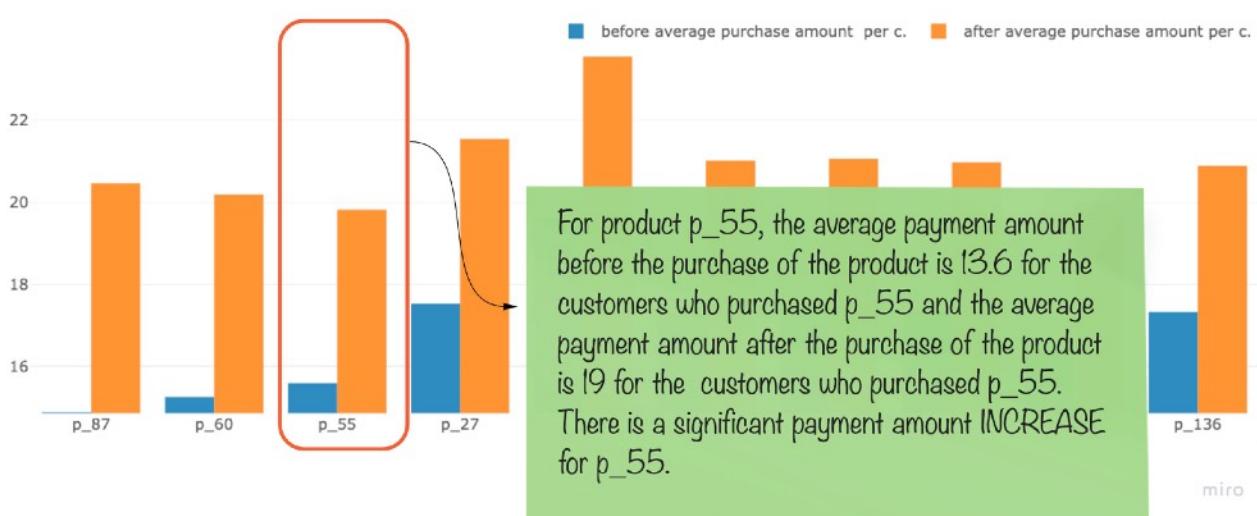
Each test process is applied with Bootstrapping Method. So, Iteratively randomly selected A and B sample customers of order count / average purchase amount will be tested acceptable enough time. Each iteration A - B samples are randomly selected. The accepted ratio will show us the confidence of our test results. For instance, there are 100 tests and 75 of them are ended with H0 Accepted. So, tests are % 75 our tests are H0 Accepted.

- How are Before - After Product Usage Tests designed?

It might be very confusing and it might seem a very sophisticated A/B Test process, however, it basically compares products related to customers of product selection at the basket. Each product selection at the basket is tested separately. Each timestamp of the selected product at the basket is collected for each product per customer individually. Each customer of one week before product selection and one week after the product selection time periods are compared with to metrics which are order count and average purchase amount. We aim to detect a significant increase after the time period while comparing to before the time period about purchase amount or order count.

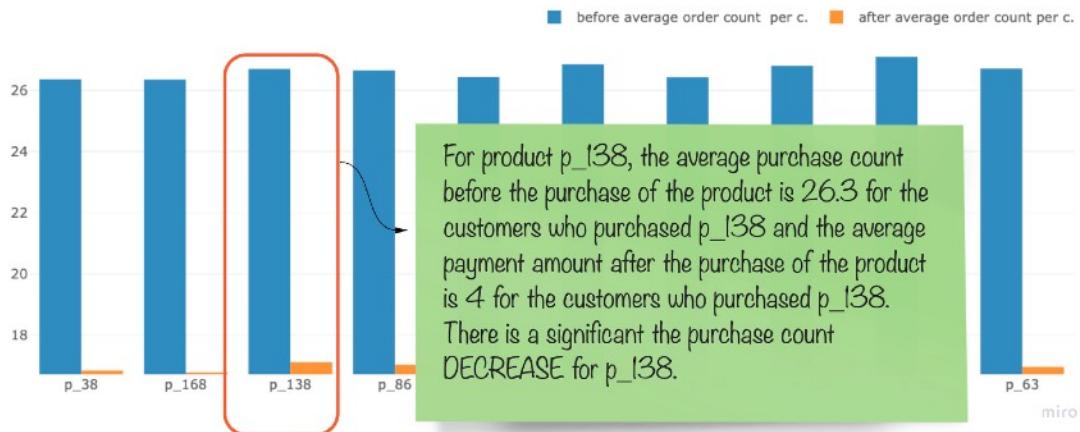
i. **Before - After time Periods customers' Average Purchase Payment Amount Test (Test Accepted!)**

The average Payment Amount per customer is calculated according to before and after the purchase of products. The assumption is "There is a significant payment amount INCREASE after the purchase of products". So, the accepted Null Hypothesis according to this assumption is shown in the chart below.



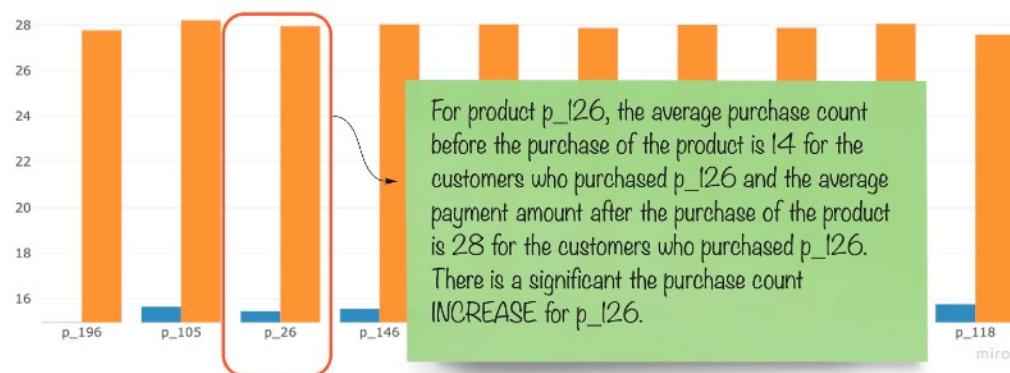
ii. Before - After time Periods customers' Average Purchase Payment Amount Test (Test Rejected!)

The average Payment Amount per customer is calculated according to before and after the purchase of products. The assumption is “There is a significant payment amount DECREASE after the purchase of products”. So, the accepted Null Hypothesis according to this assumption is shown in the chart below.

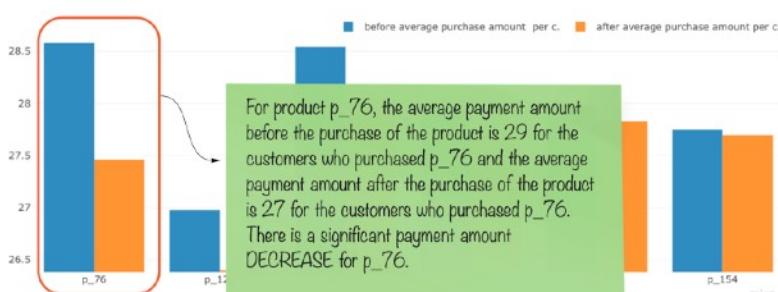


iii. Before - After time Periods customers' Total Purchase Count Test (Test Accepted!)

The total purchase count per customer is calculated according to before and after the purchase of products. The assumption is “There is a significant purchase count INCREASE after the purchase of products”. So, the accepted Null Hypothesis according to this assumption is shown in the chart below.



iv. Before - After time Periods customers' Total Purchase Count Test (Test Rejected!)



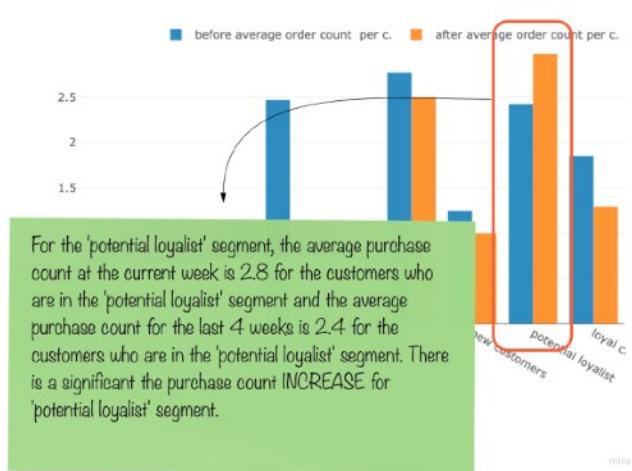
The total purchase count per customer is calculated according to before and after the purchase of products. The assumption is “There is a significant purchase count DECREASE after the purchase of products”. So, the accepted Null Hypothesis according to this assumption is shown in the chart below.

c. A/B Test Customer Segment Change

A/B Test Customer Segment Change is a bunch of tests that compare the effect of Customers Segments of Change within Time Periods on the business. Before and After time periods are constants time periods which are day, week and month. Each segment of the current day, current week and current month are compared with the last 4 weeks of the same weekday of current weekday, last 4 weeks and last 4 months. There are two metrics that help us to measure comparison values. These are the number of orders counts per segment and the Average purchase amount per segment. Difference of Segments between before and after time periods related to these metrics are tested with Before-After Analysis. in order to answer the question is "Is there any significant increase/decrease on Order Count / Average Payment Amount per Customer Segment" While we need to see to Segment which increases/decreases the number of orders and the average purchase amount, This section can help us to find the right Segment of Improvements.

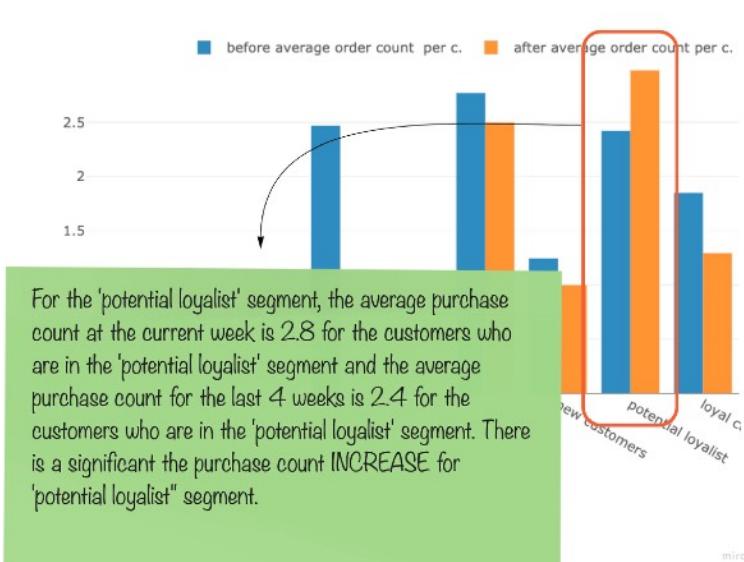
The main concept of the Customer Segment Change Analysis is for detecting the significant increase or decrease at segments. This might help us to see how growth works for each segment.

i. Daily Customer Total Order Count per Customer Segment



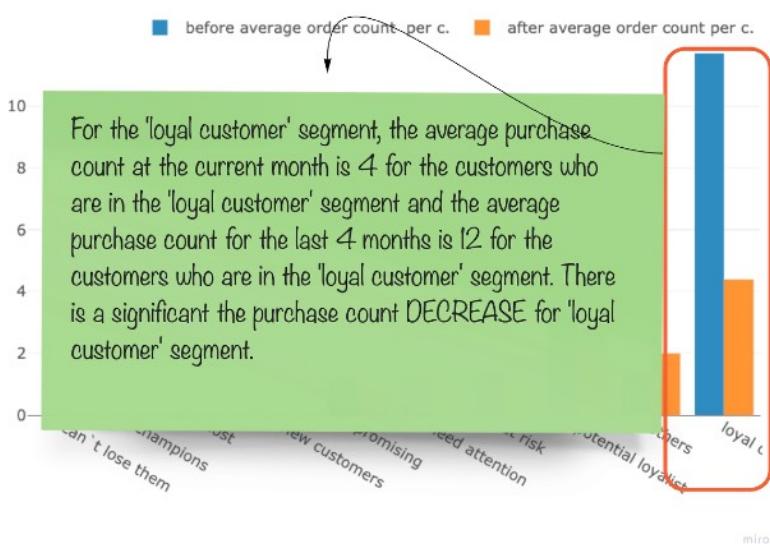
The current day of total order count is compared with the last 4 weeks of the weekday which is the same weekday as the current-day weekday. For instance, If today is Friday, the last 4 Fridays are compared with today. Each segment of customers of orders is collected separately. If there are customers whose segments changed during the time period (last 4 weeks) are not included.

ii. Weekly Customer Total Order Count per Customer Segment



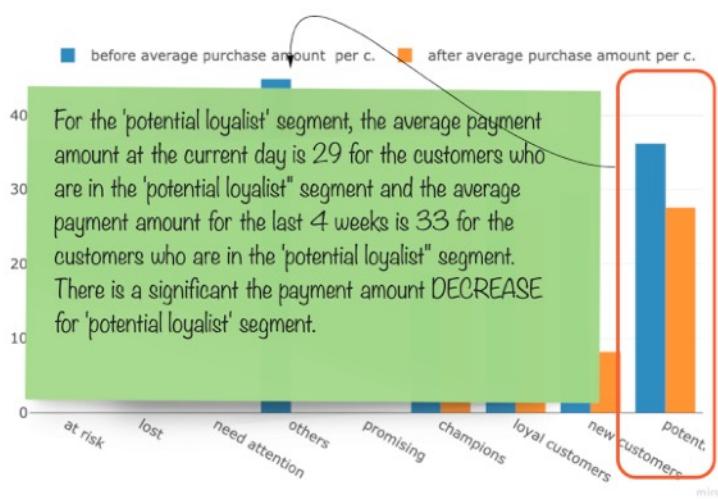
The current week of total order count is compared with the last 4 weeks. For instance, If today is at week 40, the last 4 weeks are weeks 36, 37, 38, 39 is compared with week 40. Each segment of customers of orders is collected separately. If there are customers whose segments changed during the time period (last 4 weeks) are not included.

iii. Monthly Customer Total Order Count per Customer Segment



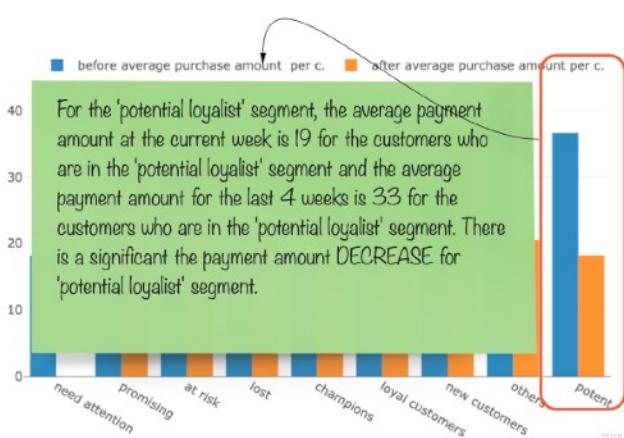
The current month's total order count is compared with the last 4 months. For instance, If today is in July, the last 4 months are March, April, May, June is compared with July. Each segment of customers of orders is collected separately. If there are customers whose segments changed during the time period (last 4 months) are not included.

iv. Daily Customers' Average Payment Amount per Customer Segment



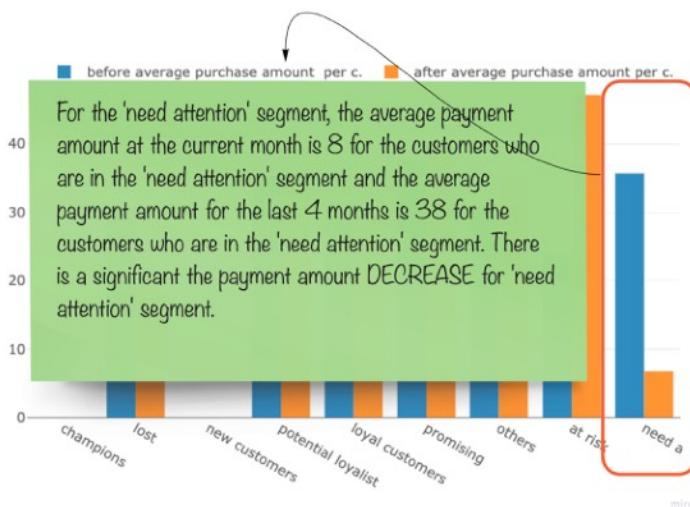
The current day of total order count is compared with the last 4 weeks. For instance, If today is on Friday, the last 4 weeks of Fridays are compared with the current day. Each segment of customers of orders is collected separately. If there are customers whose segments changed during the time period (last 4 weeks) are not included.

v. Weekly Customers' Average Payment Amount per Customer Segment



The current week of total order count is compared with the last 4 weeks. For instance, If today is in week 40, the last 4 weeks (week 39, 38, 37, 36) are compared with the current week. Each segment of customers of orders is collected separately. If there are customers whose segments changed during the time period (last 4 weeks) are not included.

vi. Monthly Customers' Average Payment Amount per Customer Segment



The current month of total order count is compared with the last 4 months. For instance, If today is in July, the last 4 months (March, April, May, June) are compared with the current month. Each segment of customers of orders is collected separately. If there are customers whose segments changed during the time period (last 4 months) are not included.

2. Customer Segmentation

Customer Segmentation is one of the popular arguments while analyzing the CustomerAnalytics. It uses one of the popular Segmentation perspective which is RFM (recency - monetary - frequency).

RFM is the most common Marketing measurement for each business. It is easy to apply for any kind of business. It is the combination of 3 metrics which are Recency - Monetary - Frequency.

The customers, who are engaged with the business, are counted as millions. However, Each individual customers have their unique behavior with the business, mostly they can be clustered in the same way according to their engagement and attention to your business. In that case, Customer Segmentation allows us to see this similarity of customers by separating the customers' population to the individual homogeny sub-groups.

- Recency
It is a time difference measurement related to how recently the customer engages with the business.
- Monetary
It is the value of the purchases per customer.
- Frequency
It is a time difference measurement related to how average hourly difference between 2 orders per customer in the business.

a. RFM

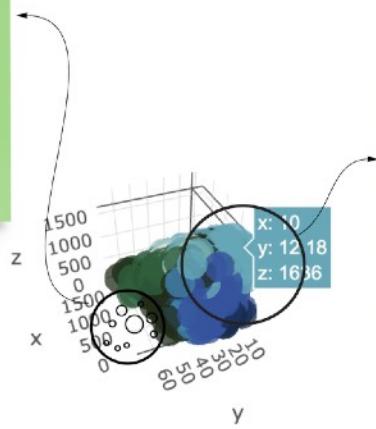
RFM is the most common Marketing measurement for each business. It is easy to apply for any kind of business. It is the combination of 3 metrics which are Recency - Monetary - Frequency.

i. RFM 3D Scatter

Each point represents individual customers' RFM values The X-axis represents recency; Y-axis represents monetary, Z-Axis represents frequency values.

Champions

customers, who has with the lowest recency and frequency and the highest monetary values, are segmented as 'champions'.



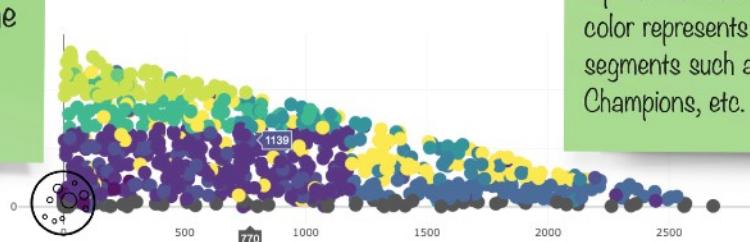
x-axis represents recency, y-axis represents monetary, z-axis represents frequency. Each color represents the segments such as Loyal Customer, Champions, etc.

miro

ii. Frequency - Recency

The X-axis represents frequency; Y-axis represents recency values. Colors represent Customer Segments.

customers, whose recency and frequency values are the lowest, are able to be detected from the chart.



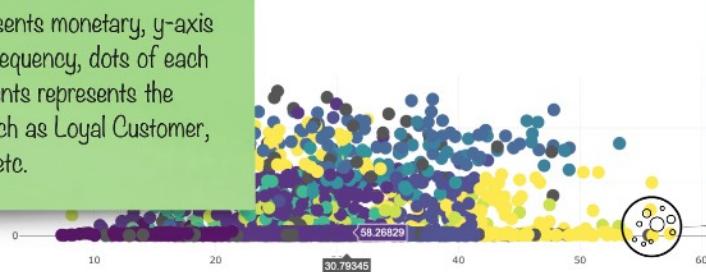
x-axis represents frequency, y-axis represents recency, dots of each color represent the segments such as Loyal Customer, Champions, etc.

miro

iii. Monetary - Frequency

The X-axis represents monetary; Y-axis represents frequency values. Colors represent Customer Segments.

x-axis represents monetary, y-axis represents frequency, dots of each color represent the segments such as Loyal Customer, Champions, etc.

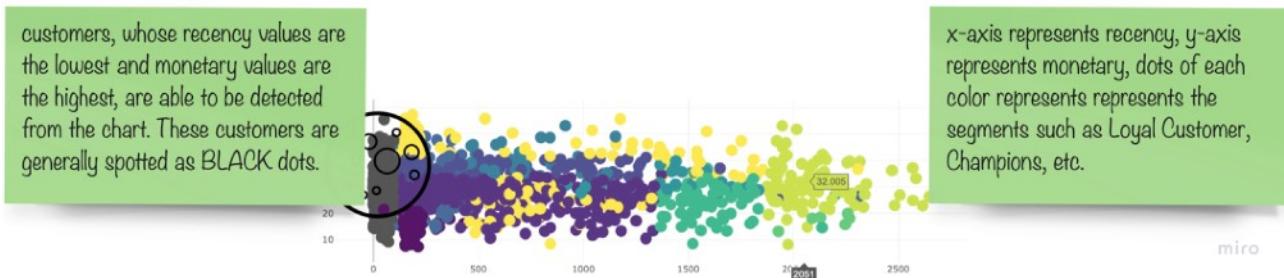


customers, whose frequency values are the lowest and monetary values are the highest, are able to be detected from the chart. These customers are generally spotted as YELLOW dots.

miro

iv. Recency - Monetary

The X-axis represents recency; Y-axis represents monetary values. Colors represent Customer Segments.



b. Segmentation

The customers, who are engaged with the business, are counted as millions. However, Each individual customers have their unique behavior with the business, mostly they can be clustered in the same way according to their engagement and attention to your business. In that case, Customer Segmentation allows us to see this similarity of customers by separating the customers' population to the individual homogeny sub-groups.

- **Segments:** By using RFM values it is possible to segment each customer. We aim to find how they engage with the business which part of them can be encouraged to engage the business, who are gone? Who are new to the business? Here are the segments;
 - **Champions:** They love your business and are pleased with your services or products. Probably, They won't let you down very soon. You are the rock start for them.
 - **Loyal Customer:** Just like champions, they do love your business and not thinking to leave you soon. They are pleased about your service and products of quality. Just They do love your rock band and never miss your concert, however, they would like to watch it at the back.
 - **Potential Loyalist:** This group of customers would like to engage with your business recently, frequently, but they are still not sure that they would like to totally engage. They probably need a push.
 - **Can't lose them:** Just like the 'potential loyalist' segment of customers, these are engaged customers, but their average basket value is a bit higher when we compare with loyal customers.
 - **At risk:** They probably leave and become churn customers. Make sure to keep them engaged in the business.
 - **Need Attention:** Just like 'at risk' segment of customers, these are going to become churn customers. Like the 'at risk' segment of customers, you have more chance to make them engaged again.
 - **Lost:** They are churn customers or very close to become the churn customers.
 - **Newcomers:** They have just arrived at your business. And we might not know about their satisfaction because of the lack of information about them. Track their behaviors and see how they become churn or champions in the future.

- **How does Customer Segmentation work?**

The calculated RFM values are segmented individually with 5 segments ($K=5$). New process will be the assigning of segment name according the top segments for RFM values. For instance, A customer who has recency; 0.2 hr (average recency; 3 days), monetary; 70 (average monetary is 30), frequency; 1 day (average frequency is 4 days). This customer's frequency segment will be 1, recency segment will be 1, monetary segment will be 1. This customer will be segmented as 'Champions'.

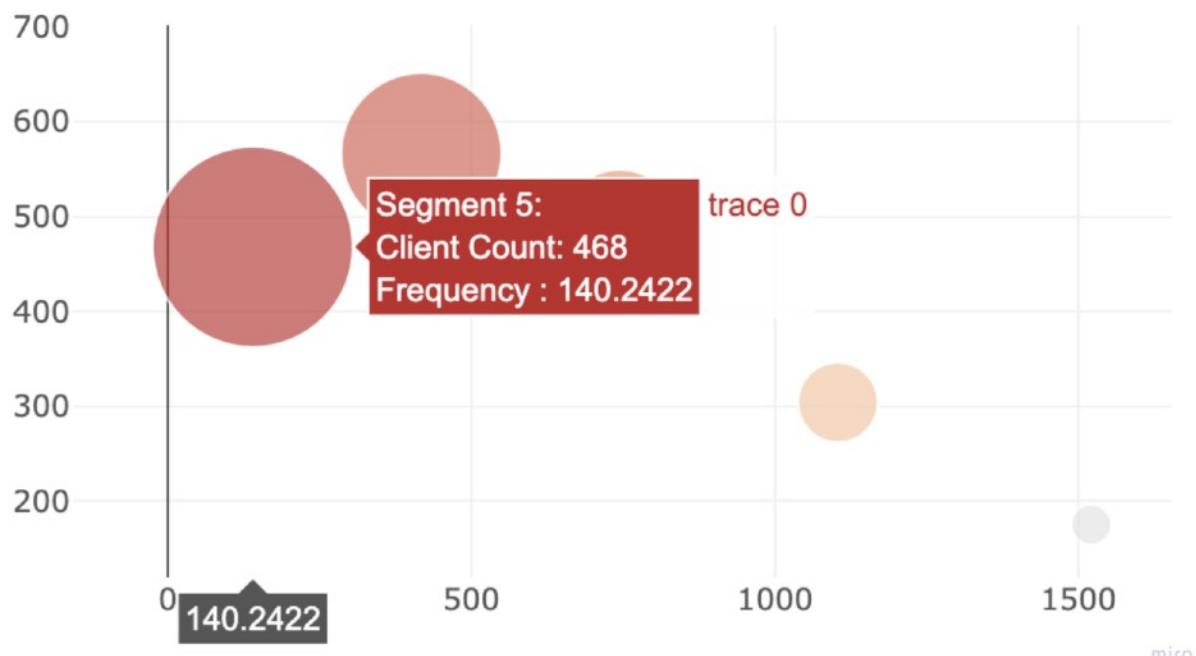
i. Customer Segmentation TreeMap

This is a treemap with % of total unique customer count per segment.



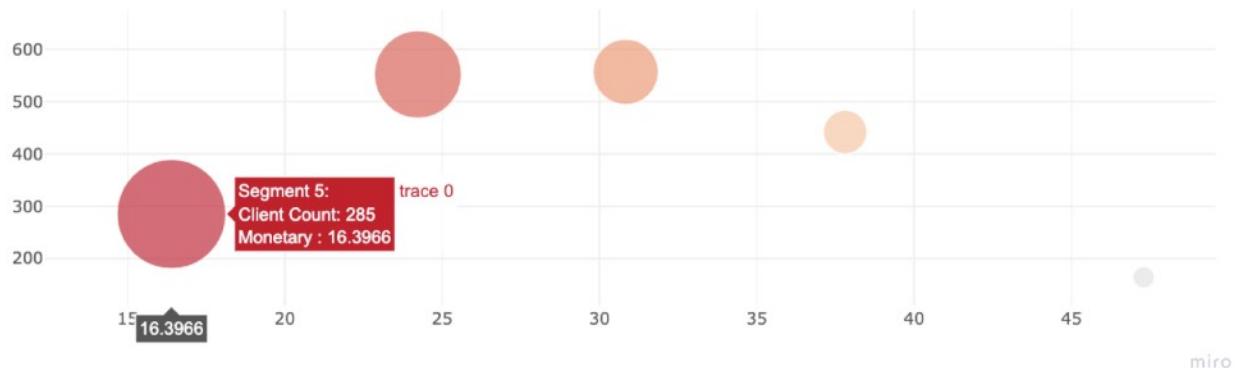
ii. Frequency Segmentation

Each customer's average time difference is calculated (Frequency Calculation). Frequency values are clustered with K-Means clustering into the 5 clusters ($K=5$). According to each frequency value number of customers is counted per frequency segment. The X-axis represents average frequency values. Y-Axis represents the total number of client count per average frequency per frequency segment.



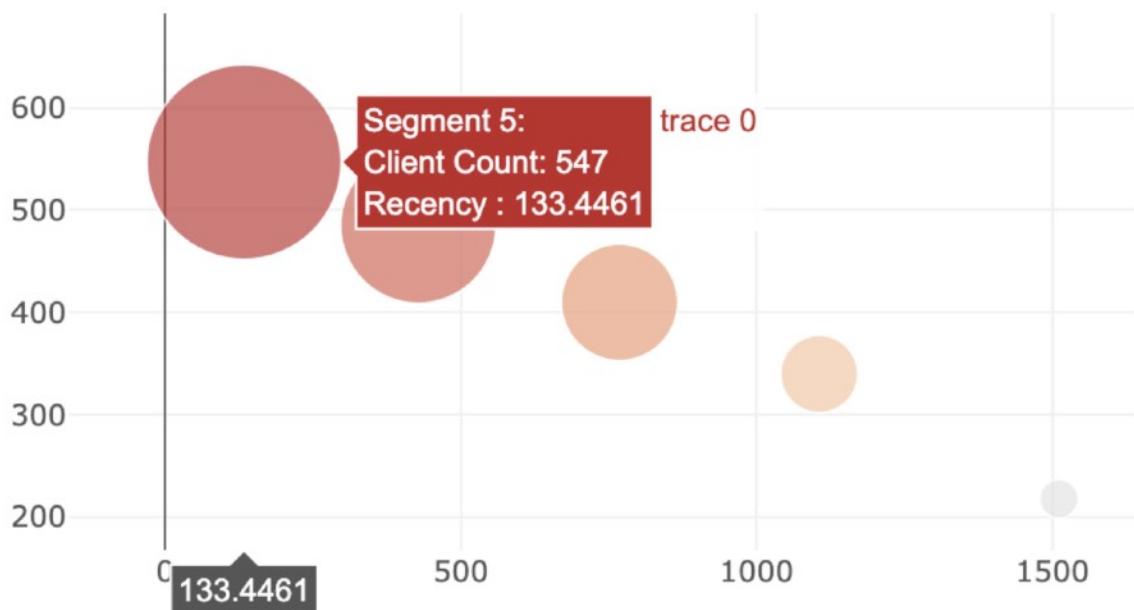
iii. Monetary Segmentation

Each customer's average purchased amount is calculated (Monetary calculation). Monetary values are clustered with K-Means clustering into the 5 clusters (K=5). According to each monetary value number of customers is counted per monetary segment. The X-axis represents average monetary values. Y-Axis represents the total number of client count per average monetary per monetary segment.



iv. Recency Segmentation

Each customer's average time difference between the current date to last purchased date is calculated (recency calculation). Recency values are clustered with K-Means clustering into the 5 clusters (K=5). According to each recency value number of customers is counted per recency segment. The X-axis represents average recency values. Y-Axis represents the total number of client count per average recency per recency segment.



3. CLV Prediction

CLV Prediction enables us to see how the customers engage with the business, what this engagement brings to the businesses. In order to predict the future of customer engagement, each of them must be well understood. At CustomerAnalytics, it is applied for next 6 months. While it is scheduling, the predictions for the future 6 months is applied once in two weeks.

• How does CLV Prediction work?

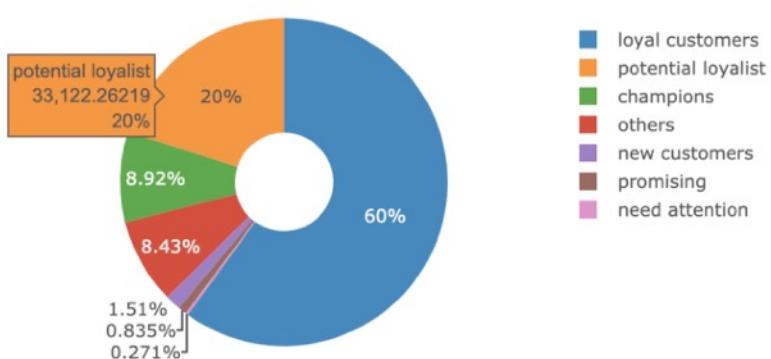
Each customer's next purchase date is predicted for the next 6 months. According to their purchase amount per their order predicted purchase dates of amounts are also predicted. However, we are only involved by using this mythology, newcomers of the total purchase amount are also predicted individually. CLV Prediction processes are not executed per day. It is executed per week. Each week the clv predictions are calculated for the next 6 months. You probably see overlapping days on the line chart. This is because of the weekly prediction process. In addition to that, This overlapping also helps us to track the CLV Performance about comparing Actual and Predicted Total Payment Amount per day.

i. Next Week CLV Prediction

In the timeline, you probably see the daily total amount and predicted purchase amount per day. In that case, it is possible to see how well your business works. There are two main categorical data which are 'actual' and 'predicted'. Actual data is the total purchase amount per day. Predicted Data is the total predicted amount per day.



ii. Customers of Segments of Total Predicted Purchase Amounts



This chart shows us the Predicted Customers of Segments. Their segments of % give us the idea of the business of growth.

4. Anomaly Detection

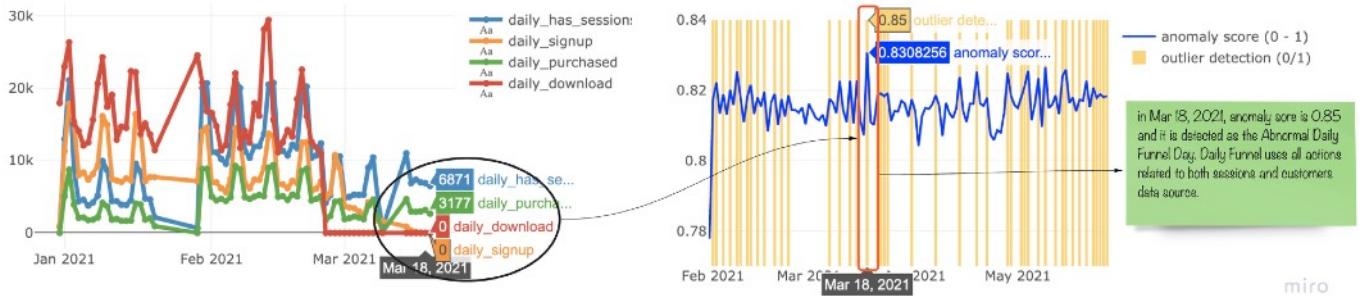
In order to catch the significant increase/decrease, even we think that the business doing ok/not well, anomaly detection allows us the alarming days or weeks or time periods. This process also allows us to see where the business did well and where can be improved.

- **How does Anomaly Detection work?**

Anomaly Detection mainly concerns with the significant increase/decrease of a data point in the given metric. These metrics are Daily Funnel, Daily Cohort Anomaly, Daily Orders Anomaly, CLV Prediction of RFM Vs Current RFM Anomaly. Each metric of values of abnormal measurements is detected by AutoEncoder. AutoEncoder generates scores for each metric of the data point. Residuals (the difference between the actual value and predicted value) are calculated. The outliers of the residuals are the Abnormal values.

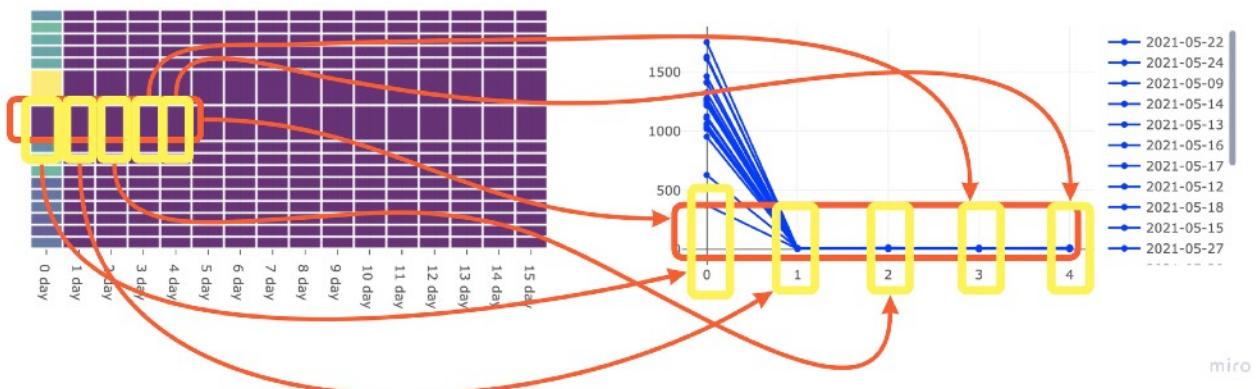
i. Daily Funnel Anomaly

Daily Funnel is the actions of totals per day. With the combination of all actions (from session count to purchase count per day), it is possible to detect abnormal days by using ML techniques.



ii. Daily Cohort Anomaly

Date columns represent downloaded day and each purchase count column represents the day that the customers had the first order after they have download. If there is an abnormal date which has a significant low/high first purchase count related to the downloaded date, this chart allows us to see the exact downloaded date as the abnormal date.



iii. Daily Cohort Anomaly With Scores (Download to First Order)

Date columns represent downloaded day and each purchase count column represents the day that the customers had the first order after they have download. If there is an abnormal date that has a significant low/high first purchase count related to the downloaded date, this chart allows us to see the exact downloaded date as the abnormal date.



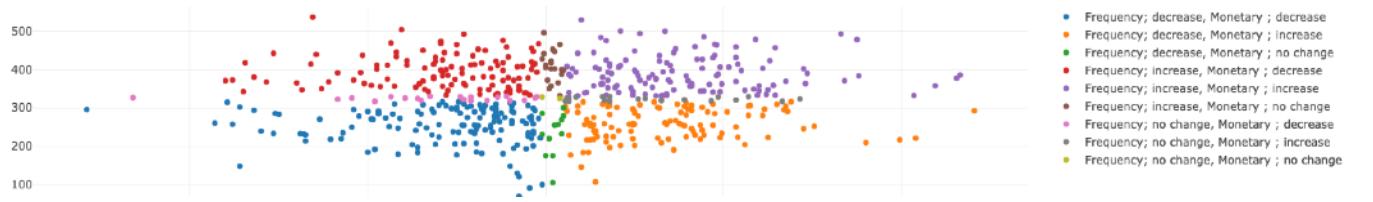
iv. Daily Order Anomaly

This chart allows us to see the % of increase/decrease compared with previous days of purchase counts.



v. CLV RFM Vs Current RFM

There are engaged customers whose purchases are predicted with CLV Prediction. We also know their Recency - Monetary - Frequency values that are calculated with their historic purchases. If we calculate their future RFM values via the CLV prediction and subtract them in order to detect a significant increase/decrease for each metric, we might clearly see how our customers' behaviors might change in the future.



F. Configurations

CustomerAnalytics allows to change profile pictures and chat according to individual charts in the system. For instance, you would like to create a comment about anomaly increase on Feb 20, 2021, on the Daily Orders Chart. It is possible with creating the message.

1. Profiles

At profiles, you can create messages or you can change your profile picture.

The screenshot illustrates the CustomerAnalytics platform's configuration features across three main sections:

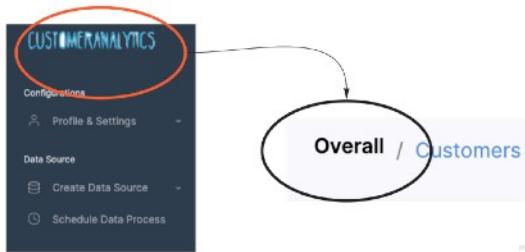
- Profile:** Shows a placeholder for a profile picture, skill tags (HTML, JavaScript, Python, Java), and a message input field.
- Activities:** Displays a line chart titled "caglan Weekly Funnel of Anomalies" showing trends from December 13, 2020, to January 17, 2021, for various metrics like daily_order_screen, daily_baskets, daily_sessions, etc.
- Chat:** Shows a conversation between users "caglan" and "caglan" with messages such as "I guess there is a problem after the date after July 2021".

Annotations with red arrows highlight specific interactions, such as pointing to the "Weekly Funnel of Anomalies" chart and the "Daily Funnel" selection in the dimension selection dropdown.

G. CustomerAnalytics DashBoard

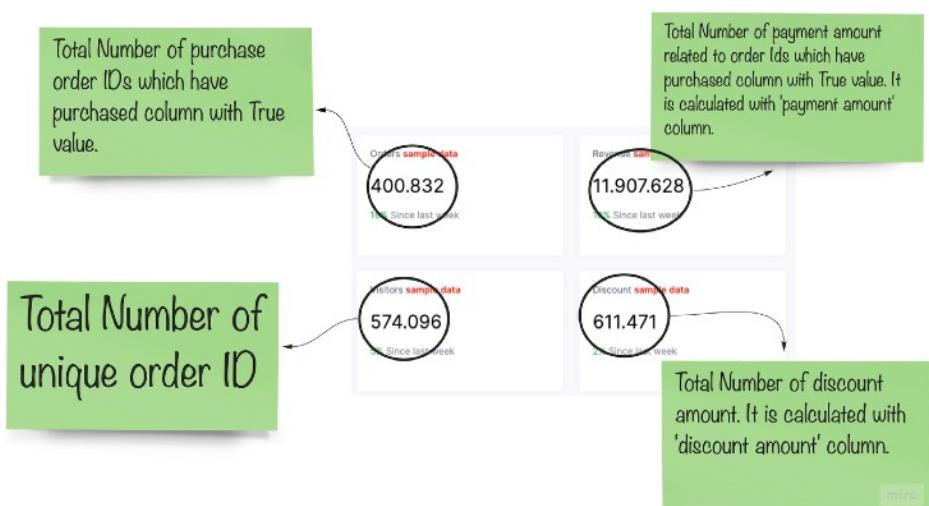
It is a dashboard with a combination of dashboards related to Exploratory analysis and Machine Learning Works. When it is logged to the CustomersAnalytics, first you are directed to the Overall Dashboard includes Orders - Revenue - Visitors - Discount KPIs, Daily Orders, Customer Journey, Churn Rate, Churn Rate Weekly, Top 10 Purchased Products, Top 10 Purchased Categories. At the other tab, you can see the Customer Dashboard includes Payment Amount Distribution, Total Number Customer Breakdown with Purchased Order Count, RFM, Download to First Order Cohort, Daily Funnel.

1. Overall Dashboard



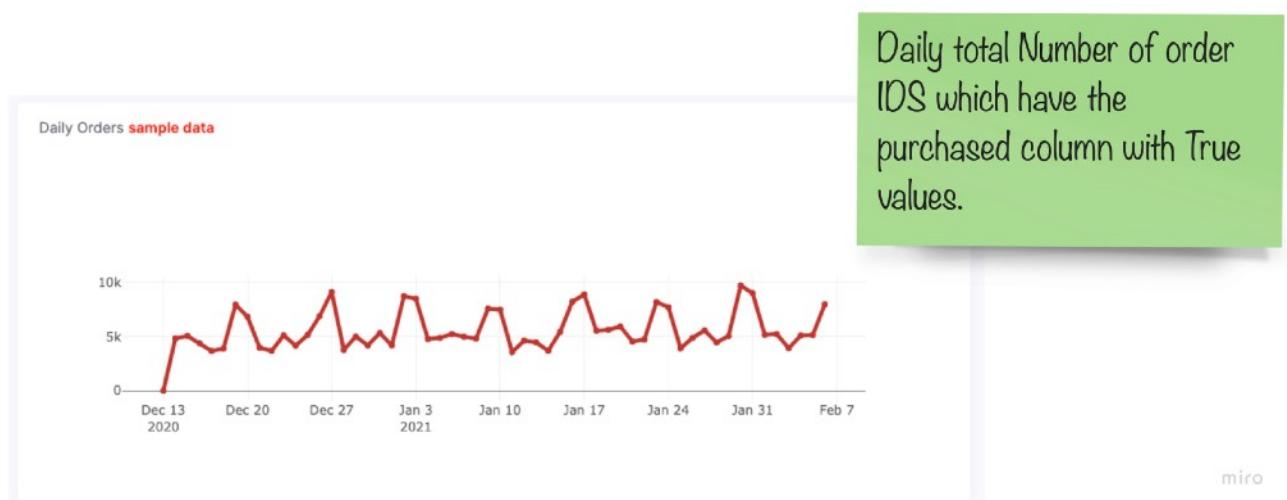
• Orders - Revenue - Visitors - Discount

Orders; Number of purchase count Revenue; Total Purchase Amount Visitors; Total Unique Visitors Count Discount; Total Discount Amount (Optional)



• Daily Orders

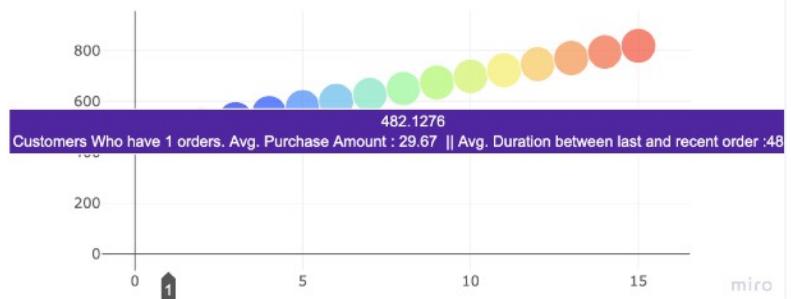
Total Number of Success Purchase Transaction per day.



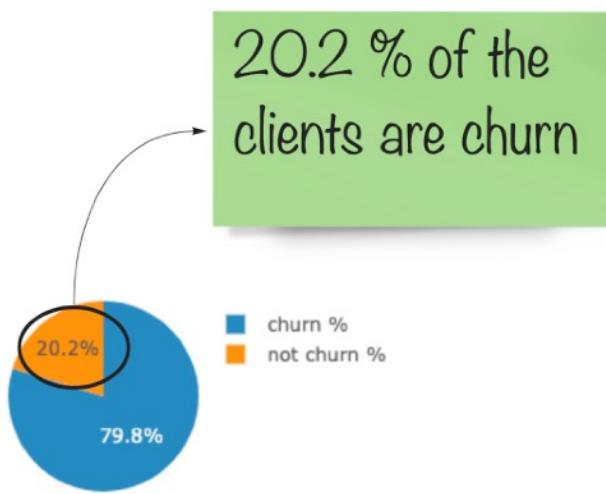
- **Customer Journey**

Customers Journey Calculation; 1. Calculate average Hour difference from Download to 1st orders. 2 . Calculate average order count 3. For each calculated average order count, calculate the average purchase amount, Example; average 2 orders, 1st orders avg 30.3£, 2nd orders avg 33.3£ Calculate average recent hours customers last order to a recent date.

X-axis represents average order count per customer. Y-axis represents average payment amount per customers of orders. At 1st order, customers paid average 482.12 and average duration from download to 1st order is 48 hr.



- **Churn Rate**

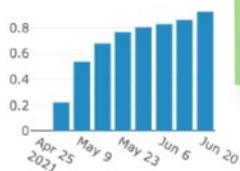


First, the Frequency value is calculated for each user. Frequency is the time difference between each sequential order per customer. Each customer's last order date is not expected to be purchased before the average frequency day. In other words, Each customer is expected to order at most average frequency days before. A number of unique customers who have ordered between average frequency date before the current date and current date are engaged users Churn is engaged users divided by total unique ordered users.

- **Churn Rate Weekly**

It is calculated the same as the churn rate per week (each Monday of the week).

monthly churn rates, range between 0 and 1.



- **Top 10 Purchased Products**

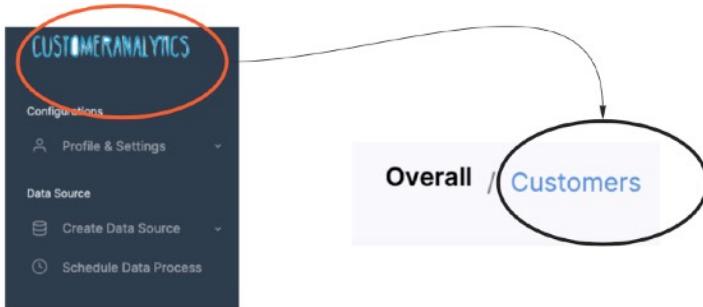
The most preferred products for the customers. Each bar represents the total number of order per product (for more details check Product Analytics).

- **Top 10 Purchased Categories**

The most preferred product categories for the customers. Each bar represents the total number of order per product category (for more details check Product Analytics).

2. Customers Dashboard

- **Payment Amount Distribution**



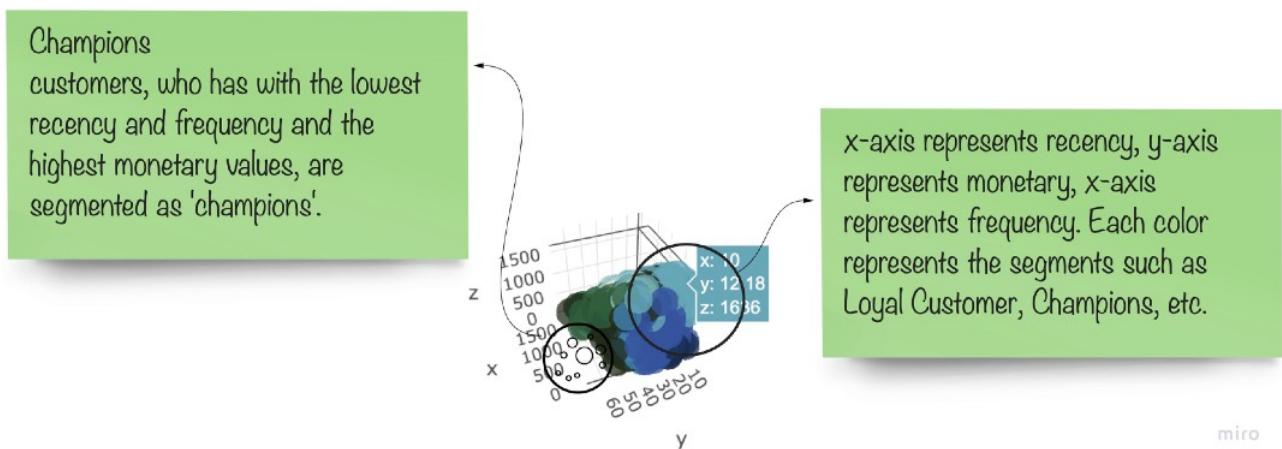
The X-Axis is the purchase amount values with bins. Y-Axis is the number of Unique Customer count related to their average purchase amounts. This Distribution allows us the see how much customers are willing to pay for a purchase.

- **Total Number Customer Breakdown with Purchased Order Count**

X Axis is Order Counts staring with 1. Y Axis is number of Unique Customer count related to their order counts. This Distribution allows us the see overall potential of your business.

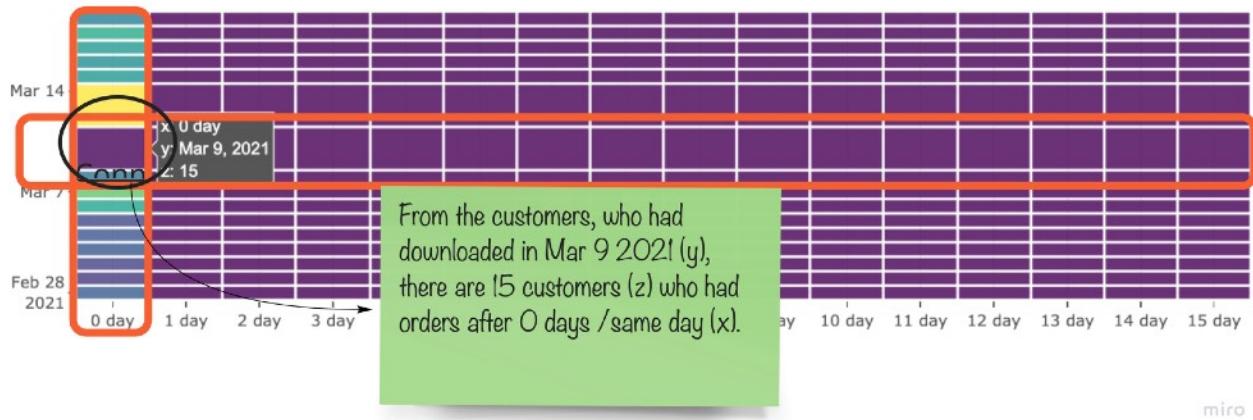
- **RFM**

X Axis represents 'recency'. Y Axis represents 'monetary'. Z Axis represents 'frequency'. Recency metric is a time difference between customers of last purchase transaction date to recent date. Monetary metric is average purchase amount per customer. Frequency metric is average time difference per between purchase date of each order per customer. There is a colored dots. These are represents the segmented customers according to their RFM values.



- **Download to First Order Cohort**

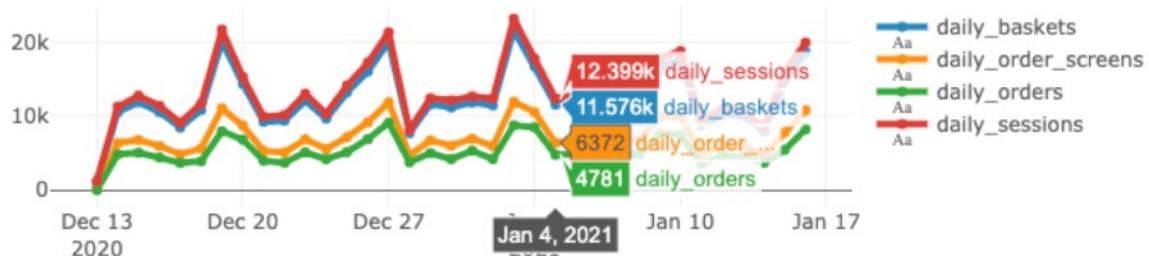
If there isn't any downloaded date, you may assign any date which is related to customers of first event with your business. This cohort of date column represent download date. If it is Weekly Cohort it will represent the mondays of each week. Otherwise it will represent days. Each Numeric column from 0 to 15 are the day differences after the downloaded date. For instance, if date columns is 2021-05-06, and numeric column is 10 and value is 100, this refers that there are 100 customers who have downloads in 2021-05-06 and have first orders 10 days later.



miro

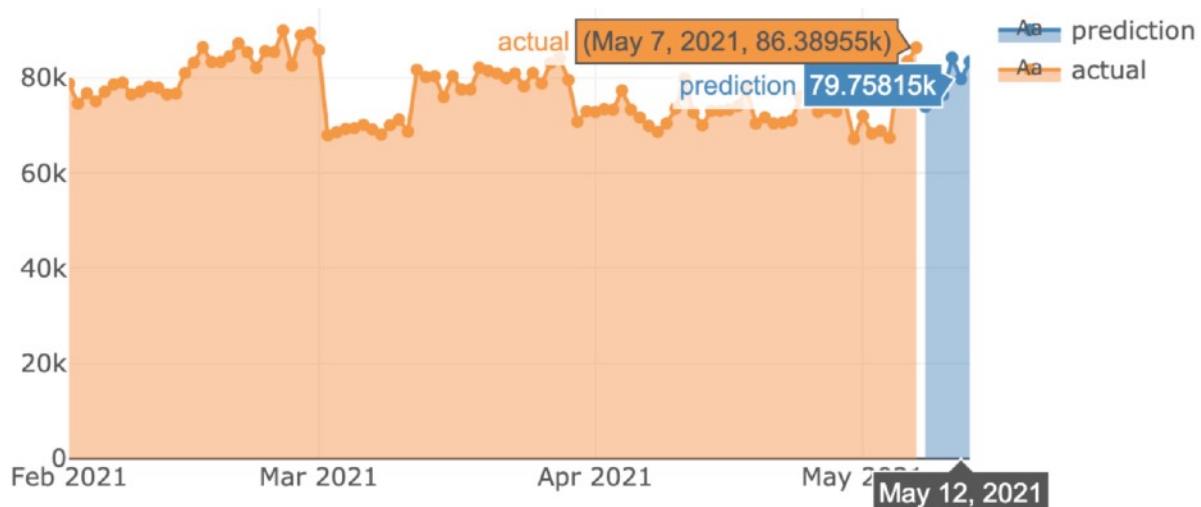
• Daily Funnel

X Axis represents days that are stored in ElasticSearch. Y Axis represents number of transactions such as order count, session count, add product the basket transaction count per day. In order to show actions such as add product the basket transaction count except order count, session count, these actions must be added to actions label while creating Sessions data source (For more information pls. check Create Data Source - Sessions & Customers).



• CLV Prediction - Daily

In a timeline, you probably see the daily total amount and predicted purchase amount per day. In that case, it is possible to see how well your business works. There are two main categorical data which are 'actual' and 'predicted'. Actual data is the total purchase amount per day. Predicted Data is the total predicted amount per day.



H. Searching

There are 4 types of search;

- Products
- Clients
- Promotions
- Dimensions

Each type of search represents an individual dashboard with results. When the expected search value is typed on the search bar;

- **Results are checked individually for each type of search.**

- Create ngrams of search value.
- Create ngrams for each list of search types (product IDs list, client IDs list, etc).
- Calculate the similarity between ngrams of search value and ngrams for each list of each types.
- Remove score = 0.
- Find the top score and assign it as a detected search results.

- **Create a temporary .csv file at temporary_folder_path that is assigned by the user.**

There are 3 charts of .csv files;

- chart_2_search.csv is positioned at right top.
- chart_3_search.csv is positioned at left bottom.
- chart_4_search.csv is positioned at right bottom.

There are 4 KPIs with 1 .csv file;

chart_1_search.csv is positioned at left top. These KPIs will be changed.

- **Each chart of data is created at temporary file.**

(chart_1_search.csv, ..., chart_4_search.csv) will be removed after the dashboard is shown at the user interface.

- **Each search type has individual charts and KPIs.**

So, each creation of charts and KPIs .csv files must be applied individually.

1. Product

Each product ID or name can be searched from the search bar. Product search is able to check unique product IDs or names at product columns that are created at Products Data Source. Once the scheduling process has been done, it is possible to search for unique products. In order to search for the new product name or ID, it must be rescheduled.



2. Client

Each client or client ID can be searched from the search bar. Client search is able to check unique clients at the client column that is created at Sessions Data Source and Customers Data Source. Once the scheduling process has been done, It is possible to search for unique clients. In order to search for the new client name or ID, it must be rescheduled.

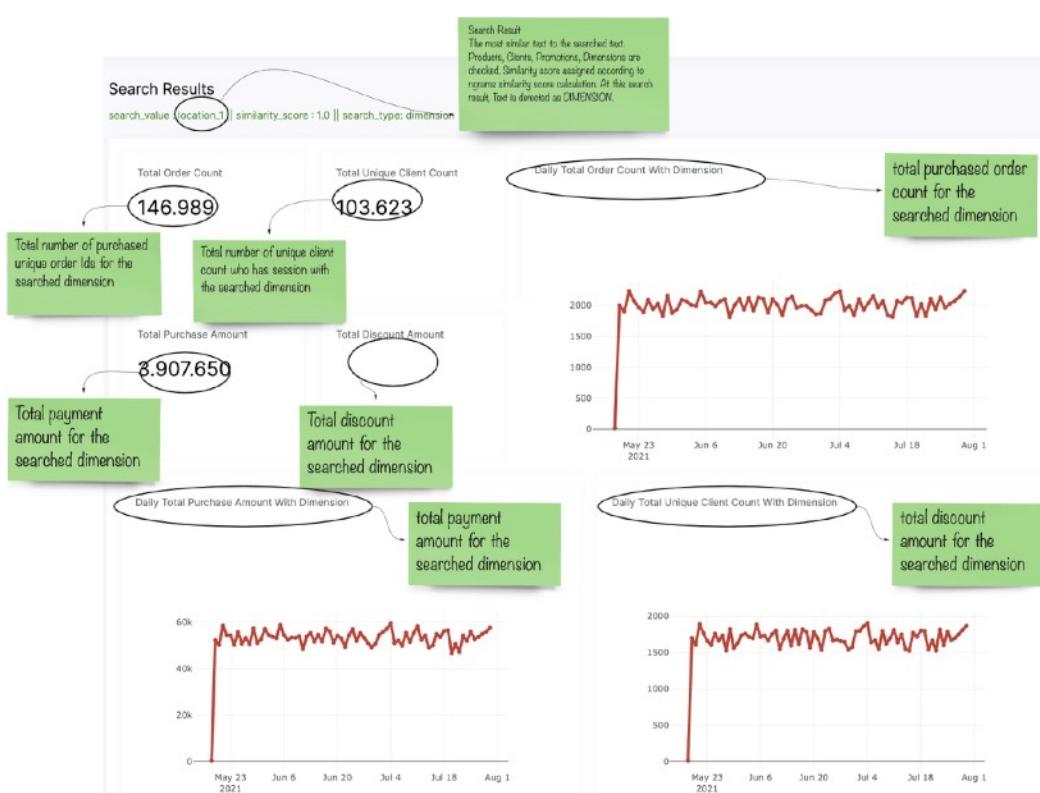


3. Promotion

Each promotion or promotion ID can be searched from the search bar. Promotion search is able to check unique promotions at the promotion column that is created at Sessions Data Source and it is the optional column. Once the scheduling process has been done, It is possible to search for unique promotions. In order to search for the new promotion name or ID, it must be rescheduled.



4. Dimension



Each dimension value can be searched from the search bar. Dimension search is able to check unique dimensions at the dimension column that is created at Sessions Data Source and it is the optional column. Once the scheduling process has been done, It is possible to search for unique dimensions. In order to search for the new dimension, it must be rescheduled.

I. Use CustomerAnalytics via Python

Creating ElasticSearch connection, temporary data folder, Sessions - Customers - Products Data Sources, Exploratory Analysis, Machine Learning Works and running/shutting down CustomerAnalytics user interface are able to be operated via Python code.

```
import customeranalytics
```

1. Creating ElasticSearch connection and Temporary Data Folder

It enables us to configure ElasticSearch with host and port. Another requirement which is a temporary path is for importing files such as CLV Prediction of model files and .csv format files with the build_in_reports folder.

- **port** : ElasticSearch port
- **host**: ElasticSearch host
- **temporary_path**: the folder path for importing data into the given directory in .csv format

```
customeranalytics.create_ElasticSearch_connection(  
    port="9200,  
    host="localhost",  
    temporary_path="/*****")
```

2. Sessions - Customers - Products Data Sources

This process is for the connecting data sources which are **SESSIONS, CUSTOMERS PRODUCTS**. Each data source has its own unique form to connect. This process checks the connection failure, before store the connection information to the SQLite DB. There are 3 main updating processes for the store process;

1. **db connection**; These process included data_source_name, data_source_type, DB_name, DB_user, DB_name, data_source_path/query, etc.
2. **actions**; This is for actions both for Sessions and Customer Data Source. Example of actions; "has_basket, order_screen". Actions are split with ',' and string format.
3. **column names**; data source columns must be matched with ElasticSearch fields for each data source (Session, Customers, products) individually. column names must be string format. Example of columns;
 1. **sessions_fields**:
 - **order_id**: unique ID for each session (purchased/non-purchased sessions).
 - **client**: ID for client this column can connect with client ID (client_2) or customer_fields.
 - **session_start_date**: eligible date format (yyyy-mm-dd hh:mm)
 - **date**: eligible date format (yyyy-mm-dd hh:mm)
 - **payment_amount**: value of purchase (float/int). If it not purchased, please assign None, Null '-'
 - **discount_amount**: discount of purchase (float/int). If it not purchased, please assign None, Null '-'
 - **has_purchased**: True/False. If it is True, session ends with purchased. If it is False, session ends with non-purchased.
 - **optional_columns**; date, discount_amount
 2. **Customer fields**:
 - **client_2**: unique ID for client this column can connect with client ID (client) or session_fields.

- **download_date**: eligible date format (yyyy-mm-dd hh:mm). This date can be any date which customers first appear at the business.
 - **signup_date**: eligible date format (yyyy-mm-dd hh:mm). First event of timestamp after the download_Date per customer.
 - **optional columns**; signup_date
- 3. Product fields;**
- **order_id**: Order ID for each session which has the created basket. This column is eligible to merge with Order Id column at session fields.
 - **product**: product Id or name. Make sure it is easy to read from charts.
 - **price**: price per each product.
 - **category**: product of category.

Products and sessions data sets are stored into the orders Index at ElasticSearch. Customer data sets are stored in the customers Index at ElasticSearch.

- Parameters
- **customers_connection**: dictionary with data_source, data_query_path, host, port, password, user, db. Ex;

```
customers_connection = {'data_source_type': "postgresql",
                       'data_query_path': """
                           SELECT *
                           FROM customers
                           """
                       ,
                       'user': "c*****",
                       'password': "1*****",
                       'port': "5432",
                       'host': "127.0.0.1",
                       'db': "c*****"
}
```

- **sessions_connection**: dictionary with data_source, data_query_path, host, port, password, user, db. Ex;

```

sessions_connection = {'data_source_type': "postgresql",
                      'data_query_path': """
                                      SELECT *
                                      FROM sessions
                                      """,
                      'user': "c*****",
                      'password': "1*****",
                      'port': "5432",
                      'host': "127.0.0.1",
                      'db': "c*****"
                    }

```

- ***products_connection:*** dictionary with data_source, data_query_path, host, port, password, user, db. Ex;

```

products_connection = {'data_source_type': "postgresql",
                      'data_query_path': """
                                      SELECT *
                                      FROM products
                                      """,
                      'user': "c*****",
                      'password': "1*****",
                      'port': "5432",
                      'host': "127.0.0.1",
                      'db': "c*****"
                    }

```

- **sessions_fields:** dictionary with order_id, client, session_start_date, date, payment_amount, discount_amount, has_purchased. Ex;

```
sessions_fields = {'order_id': 'order_id',
                   'client': 'client',
                   'session_start_date': 'session_start_date',
                   'date': 'end_date',
                   'payment_amount': 'payment_amount',
                   'discount_amount': 'discount_amount',
                   'has_purchased': 'has_purchased'}
```

- **customer_fields:** client_2, download_date, signup_date. Ex;

```
customer_fields = {'client_2': 'client',
                   'download_date': 'download_d',
                   'signup_date': 'signup_d'}
```

- **product_fields:** order_id, product, price, category. Ex;

```
product_fields = {'order_id': 'order_id',
                  'product': 'product',
                  'price': 'price',
                  'category': 'category'}
```

- **actions_sessions:** string with comma separated for sessions. Ex;

```
actions_sessions = "has_basket, order_screen"
```

- **actions_customers:** string with comma separated for customers. Ex;

```
actions_customers = "first_login_date"
```

- ***promotion_id***: string column name for promotions. Ex;

```
promotion_id = "promotion_id"
```

- ***dimension_sessions***: string column name for dimensions. Ex;

```
dimension_sessions = "dimension"
```

Here is the example of how to create Sessions - Customers - Products Data Sources;

```
customeranalytics.create_connections(
    dimension_sessions=dimension_sessions,    # optional
    customers_connection=customers_connection, # required
    sessions_connection=sessions_connection,   # required
    sessions_fields=sessions_fields,          # required
    customer_fields=customer_fields,          # required
    product_fields=product_fields,            # required if there is a.
product data source

    products_connection=products_connection,  # optional
    actions_sessions=actions_sessions,        # optional
    actions_customers=actions_customers,      # optional
    promotion_id=promotion_id    # optional
)
```

3. Run CustomerAnalytics user interface

```
import customeranalytics
customeranalytics.create_user_interface()
```

<http://127.0.0.1:5000>

4. Shut Down CustomerAnalytics user interface

```
customeranalytics.kill_user_interface()
```

5. Run Schedule Process

```
customeranalytics.create_schedule(time_period='once')
```

6. Delete Schedule Process

```
customeranalytics.delete_schedule()
```

7. Collecting Reports

If any report needs as a pandas data-frame, this can help us collect the report with the given name. Name list of the reports are available at **report_names**.

```
customeranalytics.collect_report(  
'daily_clv', date=None, dimension='main')
```

8. Collecting all report names

```
customeranalytics.report_names()
```

This enable us to see the whole report list. Here is the list of reports;

```
['weekly_funnel',  
'daily_clv',  
'daily_funnel',  
'daily_funnel_downloads',  
'weekly_average_payment_amount',  
'product_usage_before_after_orders_reject',  
'chart_4_search',  
'recency_clusters',  
'weekly_cohort_from_2_to_3',  
'daily_dimension_values',  
'daily_cohort_from_3_to_4',  
'promotion_comparison',  
'promotion_usage_before_after_orders_reject',
```

```
'most_ordered_products',
'daily_organic_orders',
'monetary_clusters',
'frequency_recency',
'segments_change_monthly_before_after_amount',
'daily_promotion_revenue',
'chart_1_search',
'daily_cohort_from_2_to_3',
'weekly_cohort_from_3_to_4',
'product_usage_before_after_orders_accept',
'client_feature_predicted',
'frequency_clusters',
'monthly_orders',
'daily_products_of_sales',
'hourly_funnel',
'weekly_average_order_per_user',
'daily_inorganic_ratio',
'daily_cohort_downloads',
'hourly_inorganic_ratio',
'promotion_number_of_customer',
'inorganic_orders_per_promotion_per_day',
'promotion_usage_before_after_orders_accept',
'chart_3_search',
'recency_monetary',
'segments_change_monthly_before_after_orders',
'customer_journey',
'purchase_amount_distribution',
'dfunnel_anomaly',
'dcohort_anomaly_2',
'dimension_kpis',
'segments_change_weekly_before_after_orders',
'user_counts_per_order_seq',
'daily_cohort_from_1_to_2',
'hourly_organic_orders',
'weekly_cohort_downloads',
'hourly_funnel_downloads',
'product_usage_before_after_amount_accept',
'client_kpis',
'dorders_anomaly',
'churn',
'monthly_funnel_downloads',
'most_combined_products',
'avg_order_count_per_promo_per_cust',
'dcohort_anomaly',
'weekly_cohort_from_1_to_2',
'segments_change_daily_before_after_orders',
'rfm',
'promotion_usage_before_after_amount_accept',
'monthly_funnel',
'kpis',
'churn_weekly',
'clvsegments_amount',
'order_and_payment_amount_differences',
'hourly_orders',
'clvrdfm_anomaly',
'weekly_funnel_downloads',
```

```
'product_usage_before_after_amount_reject',
'weekly_average_session_per_user',
'chart_2_search',
'daily_promotion_discount',
'product_kpis',
'segments_change_weekly_before_after_amount',
'daily_orders',
'weekly_orders',
'segmentation',
'segments_change_daily_before_after_amount',
'most_ordered_categories',
'promotion_usage_before_after_amount_reject',
'monetary_frequency',
'promotion_kpis']
```