

Hands On Exercise

RFM model Classic but efficient

CRM extract example

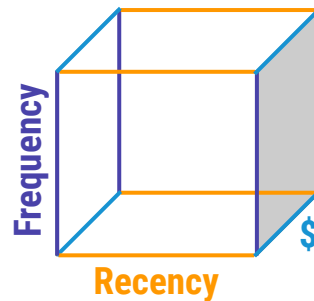
Order date	Customer ID	Order Value
01-02-15	177237	\$140
02-02-15	177323	\$30
01-03-15	177237	\$35

Recency and Frequency of
purchases at user level

Monetary Value



Customers scored with a
LTV (RFM) model



Instructions



1. Upload CSV File into BigQuery about customer orders



2. Calculate in SQL Frequency, Recency, Monetization, average basket, average items, average time between purchases (Excluding returns)



3. Create a clustering using BigQuery Kmeans and decide strategy per cluster

Hands on exercise - Upload CSV to BQ

Create table

Source

Create table from:

Google Cloud Storage

Select file from GCS bucket:

powerweek/dataset.csv

Browse

File format:

CSV

Destination

Project name

Adwords Scripts

Dataset name

PowerWeek

Table type

Native table

Table name

test

Schema

Auto detect

Schema and input parameters

Schema will be automatically generated.

Partition and cluster settings

Partitioning:

No partitioning

Clustering order (optional):

Clustering order determines the sort order of the data. Clustering can only be used on a partitioned table, and works with tables partitioned either by column or ingestion time.

Comma-separated list of fields to define clustering order (up to 4)

Advanced options

Create table

Cancel

Show debug pa

Hands on exercise - Transform Data to obtain RFM

SELECT

```
cast(customer_id as string ) as id,  
sum(order_value) as monetary, count(distinct if(order_value >0,order_date,null)) as frequency,  
sum( order_value)/count(distinct if(order_value >0,order_date,null)) as avg_basket,  
sum(if( order_qty_articles>0,order_qty_articles,null) )/count(distinct if(order_value >0,order_date,null)) as avg_items_dnn,  
date_diff(DATE(2011,12,09),max(if(order_value>0,order_date,null)), DAY) as recency,  
date_diff(max(if(order_value>0,order_date,null)),min(if(order_value>0,order_date,null)),DAY) / count(distinct if(order_value  
>0,order_date,null)) as time_between
```

from `KSCHOOL.orders_dataset`

group by 1

Row	id	monetary	frequency	avg_basket	avg_items_dnn	recency	time_between
1	12592	437.6	2	218.8	145.0	92	60.5
2	12399	1108.65	4	277.1625	294.5	119	35.5

Hands on exercise - Kmeans Clustering

```
CREATE MODEL PowerWeek.kmeans_model_5clusters OPTIONS(  
  model_type='kmeans', num_clusters=5, distance_type='euclidean') AS
```































```
SELECT
```

```
  cast(customer_id as string ) as id,  
  sum(order_value) as monetary, count(distinct if(order_value >0,order_date,null)) as frequency,  
  sum( order_value)/count(distinct if(order_value >0,order_date,null)) as avg_basket,  
  sum(if( order_qty_articles>0,order_qty_articles,null) )/count(distinct if(order_value >0,order_date,null)) as avg_items_dnn,  
  date_diff(DATE(2011,12,09),max(if(order_value>0,order_date,null)), DAY) as recency,  
  date_diff(max(if(order_value>0,order_date,null)),min(if(order_value>0,order_date,null)),DAY) / count(distinct if(order_value  
>0,order_date,null)) as time_between
```































```
from `KSCHOOL.orders_dataset`
```

```
group by 1
```

Hands on exercise - Define Strategy Per Cluster

Centroid Id	Count	monetary	frequency	avg_basket	avg_items_dnn	recency	time_between
1	410	 881.6812	 2.7390	 321.9702	 182.4925	 193.7000	 36.7018
3	1487	 1,907.5553	 4.8554	 399.0776	 246.4978	 31.2771	 38.1896
5	219	 9,410.0584	 20.0000	 489.6593	 283.8831	 9.7580	 20.1926
2	30	 62,189.7347	 19.4333	 3,396.0492	 2,177.0158	 26.8333	 28.4400
4	631	 935.1316	 2.7464	 341.0379	 207.1088	 44.0951	 98.2244

Hands on exercise - Define Strategy Per Cluster

Centroid Id	Count	monetary	frequency	avg_basket	avg_items_dnn	recency	time_between
1	410	 881.6812	 2.7390	 321.9702	 182.4925	 193.7000	 36.7018
3	1487	 1,907.5553	 4.8554	 399.0776	 246.4978	 31.2771	 38.1896
5	219	 9,410.0584	 20.0000	 489.6593	 283.8831	 9.7580	 20.1926
2	30	 62,189.7347	 19.4333	 3,396.0492	 2,177.0158	 26.8333	 28.4400
4	631	 935.1316	 2.7464	 341.0379	 207.1088	 44.0951	 98.2244

2

Top Customers (B2B) - Ad hoc offers and proactive approach

5

Top B2C Customers - Retain and Similar Audiences

3

New Potential Customers - Boost through coupons

4

Occasional Customers - Increase frequency

1

Old Customers - Get them back

Hands on exercise - Create Buckets to Activate GA

```
SELECT CENTROID_ID,id
```

```
FROM ML.PREDICT(MODEL `PowerWeek.kmeans_model_5clusters`, (select  
  cast(customer_id as string ) as id,  
  sum(order_value) as monetary, count(distinct if(order_value >0,order_date,null)) as frequency,  
  sum( order_value)/count(distinct if(order_value >0,order_date,null)) as avg_basket,  
  sum(if( order_qty_articles>0,order_qty_articles,null) )/count(distinct if(order_value >0,order_date,null)) as avg_items_dnn,  
  date_diff(DATE(2011,12,09),max(if(order_value>0,order_date,null)), DAY) as recency,  
  date_diff(max(if(order_value>0,order_date,null)),min(if(order_value>0,order_date,null)),DAY) / count(distinct if(order_value  
>0,order_date,null)) as time_between
```

```
FROM `KSCHOOL.orders_dataset`  
group by 1  
)
```