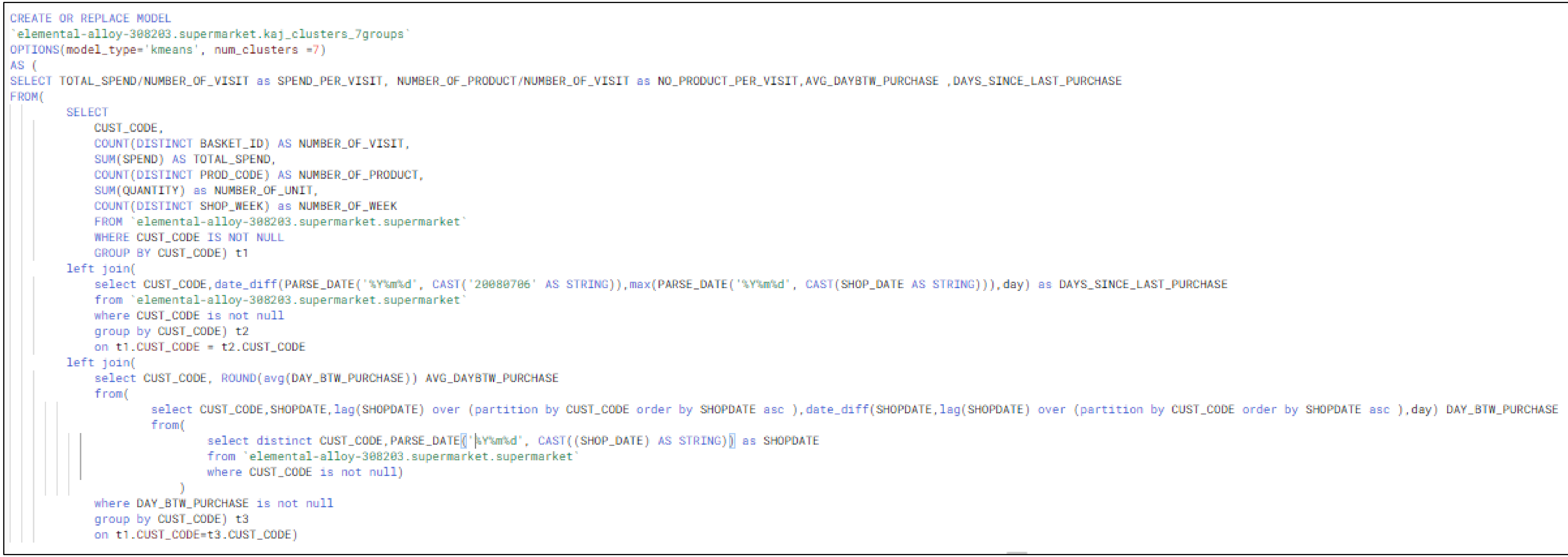
Customer Segmentation (BigQuery)

Step

1. Query data from supermarket data
2. Create customer single view for those with CUST_CODE
   - 5 features
   - Average day between purchase, Day since last purchase, Number of product per visit, Number of visit per week, Spend per visit
3. Create K-Mean using BigQuery ML
   - Try different value of K (Number of clusters)
   - Choose K that minimize Davies–Bouldin index (WSS/BSS) in order to maximize between sum of square and minimize within sum of square distance
4. See result of centroid value for each features
5. Interpretation and possible action for each cluster

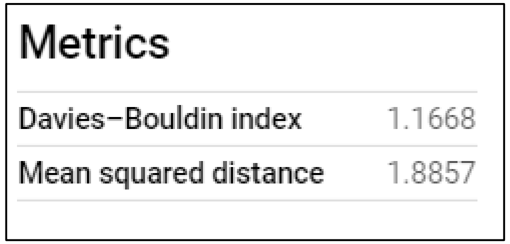
Query



```
CREATE OR REPLACE MODEL
`elemental-alloy-308203.supermarket.kaj_clusters_7groups`
OPTIONS(model_type='kmeans', num_clusters =7)
AS (
SELECT TOTAL_SPEND/NUMBER_OF_VISIT as SPEND_PER_VISIT, NUMBER_OF_PRODUCT/NUMBER_OF_VISIT as NO_PRODUCT_PER_VISIT,AVG_DAYBTW_PURCHASE ,DAYS_SINCE_LAST_PURCHASE
FROM(
        SELECT
            CUST_CODE,
            COUNT(DISTINCT BASKET_ID) AS NUMBER_OF_VISIT,
            SUM(SPEND) AS TOTAL_SPEND,
            COUNT(DISTINCT PROD_CODE) AS NUMBER_OF_PRODUCT,
            SUM(QUANTITY) as NUMBER_OF_UNIT,
            COUNT(DISTINCT SHOP_WEEK) as NUMBER_OF_WEEK
            FROM `elemental-alloy-308203.supermarket.supermarket`
            WHERE CUST_CODE IS NOT NULL
            GROUP BY CUST_CODE) t1
        left join(
            select CUST_CODE,date_diff(PARSE_DATE('%Y%m%d', CAST('20080706' AS STRING)),max(PARSE_DATE('%Y%m%d', CAST(SHOP_DATE AS STRING))),day) as DAYS_SINCE_LAST_PURCHASE
            from `elemental-alloy-308203.supermarket.supermarket`
            where CUST_CODE is not null
            group by CUST_CODE) t2
            on t1.CUST_CODE = t2.CUST_CODE
        left join(
            select CUST_CODE, ROUND(avg(DAY_BTW_PURCHASE)) AVG_DAYBTW_PURCHASE
            from(
                    select CUST_CODE,SHOPDATE,lag(SHOPDATE) over (partition by CUST_CODE order by SHOPDATE asc ),date_diff(SHOPDATE,lag(SHOPDATE) over (partition by CUST_CODE order by SHOPDATE asc ),day) DAY_BTW_PURCHASE
                    from(
                            select distinct CUST_CODE,PARSE_DATE('%Y%m%d', CAST((SHOP_DATE) AS STRING)) as SHOPDATE
                            from `elemental-alloy-308203.supermarket.supermarket`
                            where CUST_CODE is not null)
                        )
            where DAY_BTW_PURCHASE is not null
            group by CUST_CODE) t3
            on t1.CUST_CODE=t3.CUST_CODE)
```

Try different value of K

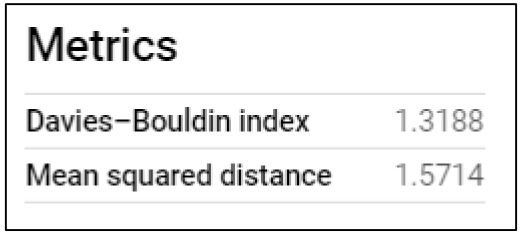
K=3



Metrics

Davies–Bouldin index	1.1668
Mean squared distance	1.8857

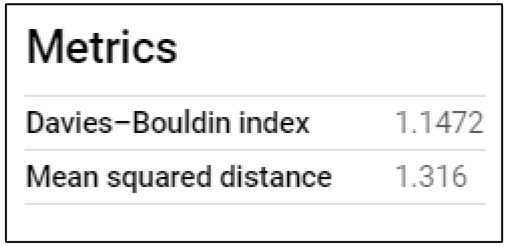
K=4



Metrics

Davies–Bouldin index	1.3188
Mean squared distance	1.5714

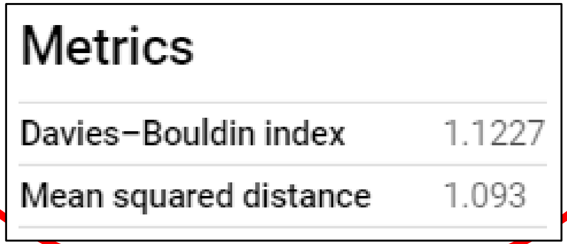
K=5



Metrics

Davies–Bouldin index	1.1472
Mean squared distance	1.316

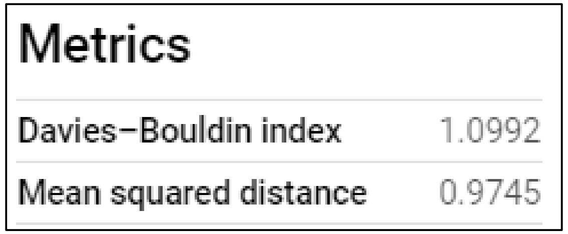
K=6



Metrics

Davies–Bouldin index	1.1227
Mean squared distance	1.093

K=7

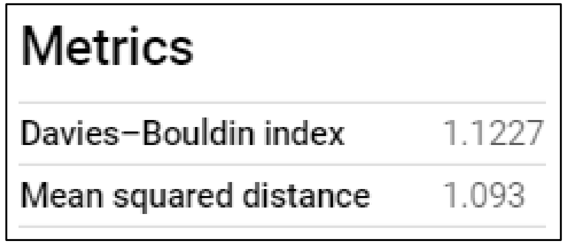


Metrics

Davies–Bouldin index	1.0992
Mean squared distance	0.9745

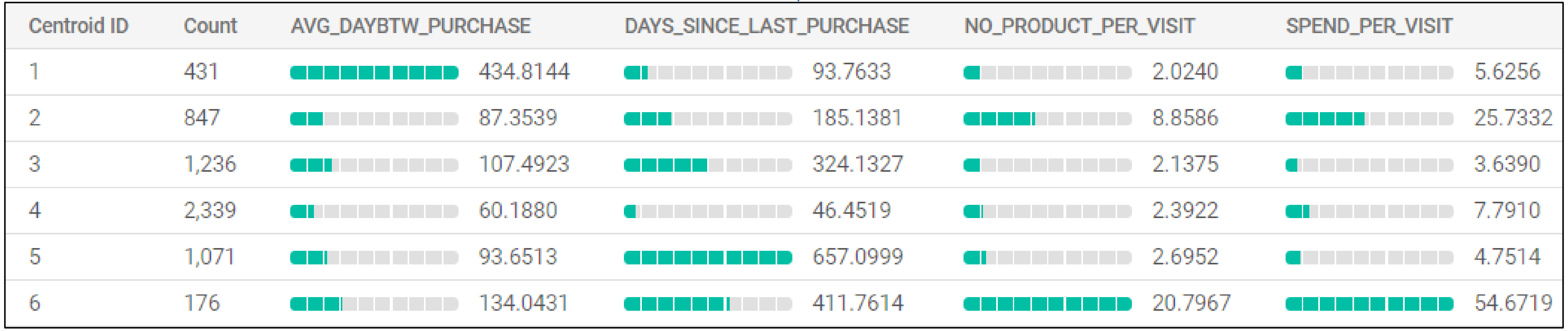
Choose smallest Davies–Bouldin index (Minimize WSS and Maximize BSS) but since 7 might be hard to interpret I will choose 6

Interpretation of each cluster



Metrics	
Davies–Bouldin index	1.1227
Mean squared distance	1.093

6100 customers



Centroid ID	Count	AVG_DAYBTW_PURCHASE	DAYS_SINCE_LAST_PURCHASE	NO_PRODUCT_PER_VISIT	SPEND_PER_VISIT
1	431	434.8144	93.7633	2.0240	5.6256
2	847	87.3539	185.1381	8.8586	25.7332
3	1,236	107.4923	324.1327	2.1375	3.6390
4	2,339	60.1880	46.4519	2.3922	7.7910
5	1,071	93.6513	657.0999	2.6952	4.7514
6	176	134.0431	411.7614	20.7967	54.6719

Group1 – Purchase once a year, low spending

Group2 – High ticket size with several product per visit

Group3 – Low ticket size, purchase 3 times a year and have not purchased for almost a year

Group4 – Frequent purchaser with low ticket size and just purchased no more than 2 months

Group5 – Idle for almost 2 year but also have very low ticket size

Group6 – High ticket size, purchase many product but haven’t purchased for a year

Action: Focus on Group2&6

✓ good

✗ bad

	Ticket size	Recency	Time to purchase	Variety	ACTION
Group1	✗	✓	✗	✗	Get them to purchase more frequent Since time to purchase is almost a year
Group2	▬	✓	✓	▬	Get them to purchase again and try To upsell to increase ticket size since This group has 2nd highest ticket size
Group3	✗	▬	✓	✗	Try to get them to purchase again since Their last purchase has been a year
Group4	✗	✓	✓	✗	Ignore since very low ticket size and they Purchase regularly
Group5	✗	✗	✓	✗	Try to get them to purchase again since Their last purchase has been 2 year
Group6	✓	▬	✓	✓	Give very special promotion especially Get them to purchase again since this group has highest ticket size but Idle for year