eCommerce Analytics and Recommendations

Team Name - "FourYottabytes"

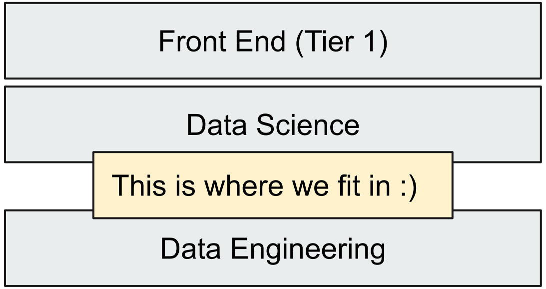
Members :-

Amogh Hassija (amoghhassija@iisc.ac.in)
Ashwin Korwarkar (kashwin@iisc.ac.in)
Henna Arora (hennaarora@iisc.ac.in)
Sai Harish Pathuri (saipathuri@iisc.ac.in)

Project Objectives

Value to the customers

- Users get recommendations based on their past behaviour
- Recommendations may help show products that users would be interested in
- User need not browse a lot to find a better suited product



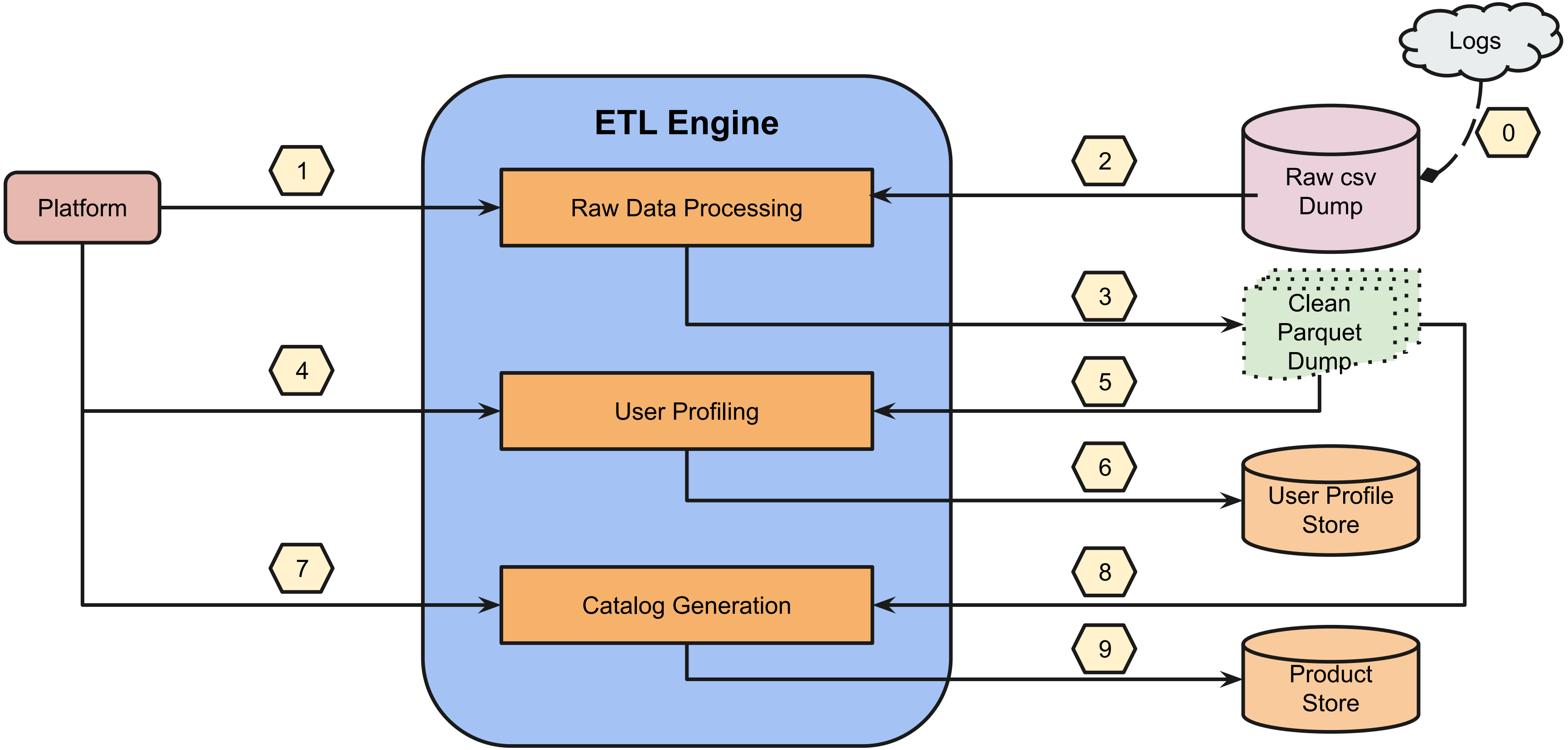
Value to the platform

- Platform suits the user's needs better
- Push more product off the shelf
- Entice the user to buy more, more frequently
- The business makes money!

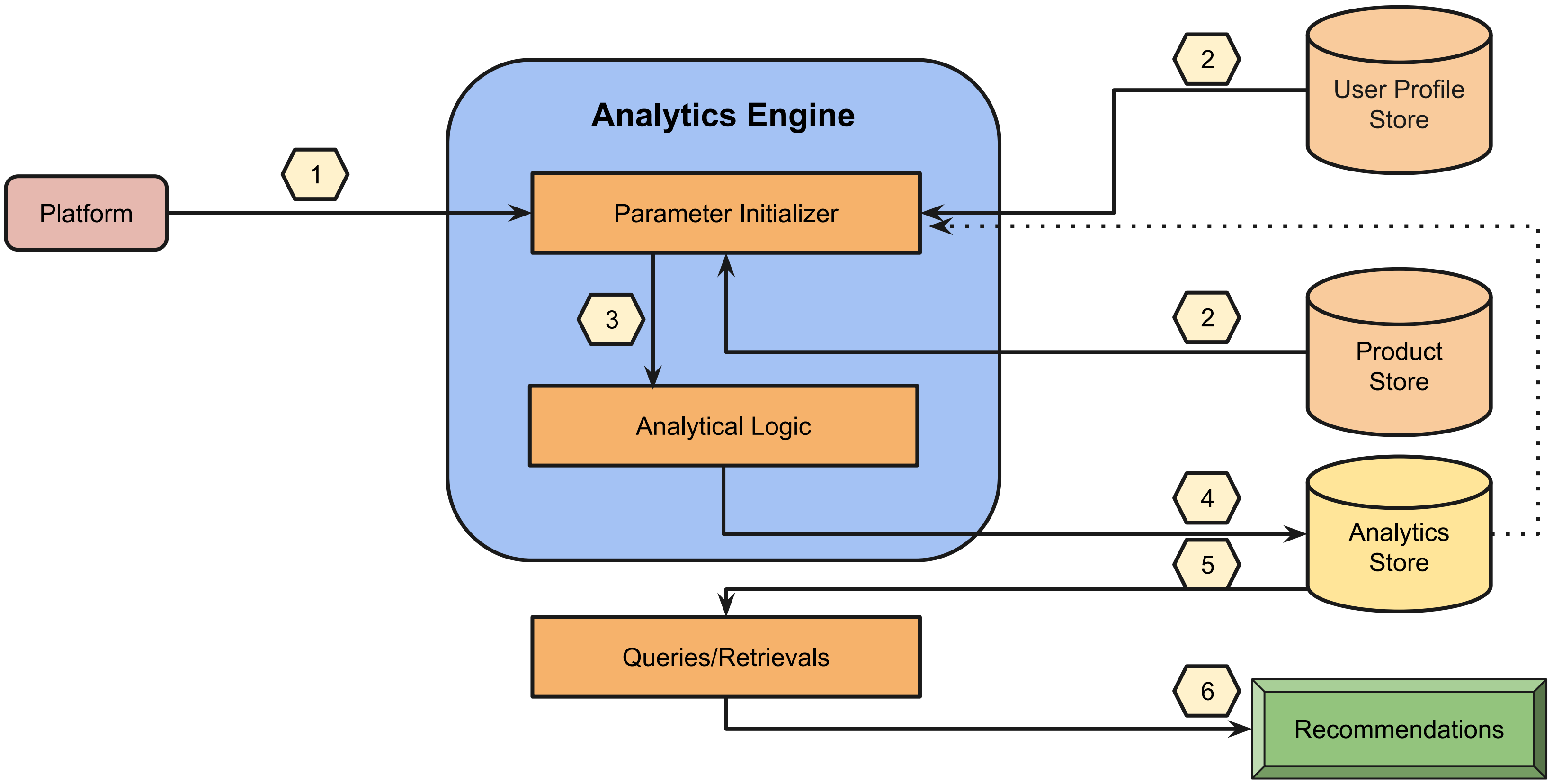
Salient Features

- Promoting underdog products
- User-brand preference ranking
- Customer segmentation based on price-point
- Recommendations based on one or multiple of the above

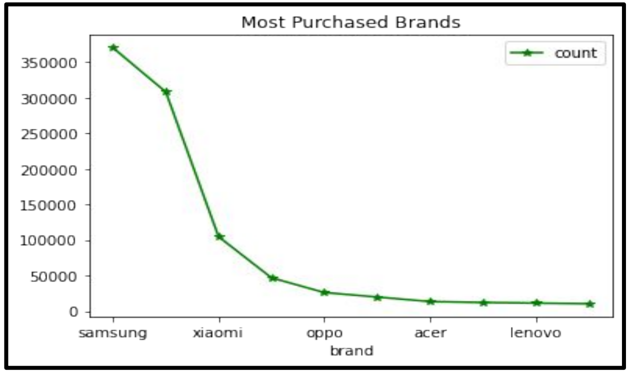
Data Preparation - ETL Pipeline

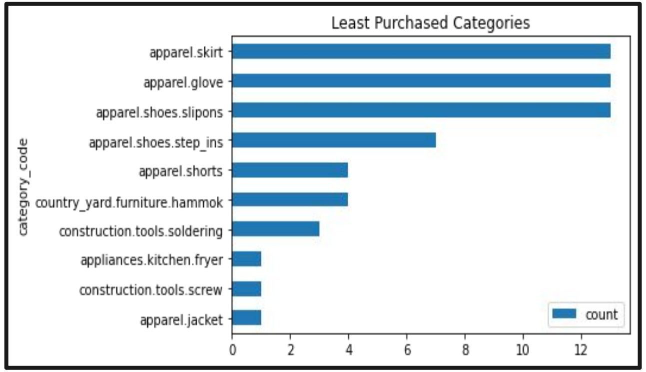


Recommendation Pipeline

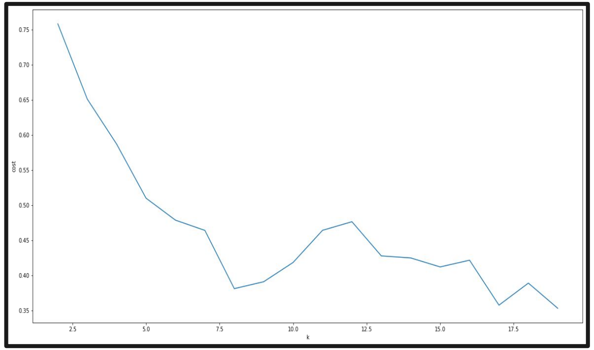


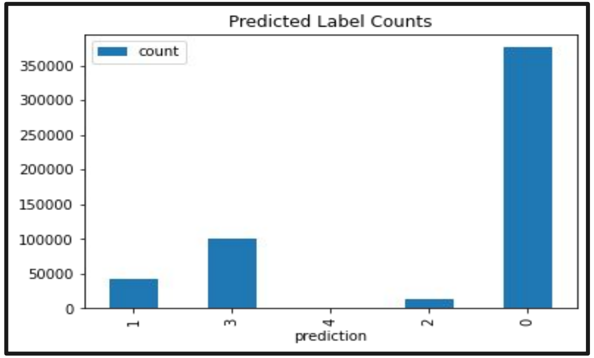
Exploratory Data Analysis

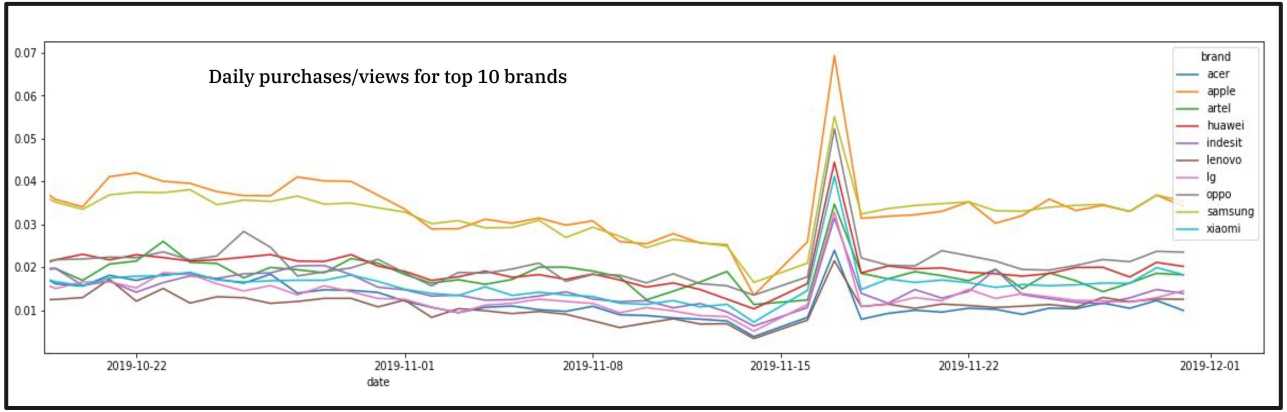




Cost vs K(#Clusters) for Customer segmentation







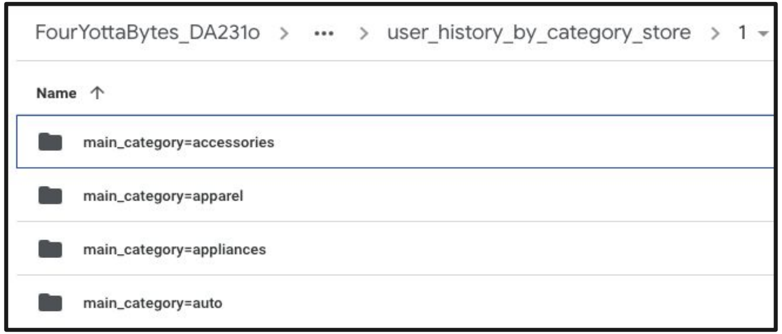
Personalized Product Recommendation

Use Case Definition: To provide product **recommendations** to the users **based on** their **average spendings or views** on the portal

Motivation: Having a **pipeline** to make **personalized recommendations**, with a minimum **threshold** on user's historical activity data.

Analytics:

- The engine tries to **analyse user behaviour** (given enough historical data)
- Top products are picked based on certain **metrics of the user**, taking threshold into account.
- This might make the **user buy more** out of what he/she sees
- The recommendations can be made from products in a particular category (like a Big Basket category page) or across all products (like your Amazon Homepage)



Input Data: Catalog and User Profile Information

user_id	event_type	event_count	avg_event_price	stddev	event_history	product_history
512370084	purchase	1	94.9800033569336	0.0	[94.98]	[17300136]
512370084	cart	2	94.9800033569336	0.0	[94.98, 94.98]	[17300136]
512370084	view	4	465.32500076293945	488.18483152851655	[1285.49, 385.85,...	[17300136]
512399877	purchase	6	1434.518330891927	139.26739369014393	[1376.87, 1541.87...	[1005105, 1005124...
512399877	view	20	1448.9830078125	173.77757957282486	[1376.87, 1376.87...	[1005105, 1005124...

product_id	event_type	event_count	avg_price	category_code	brand
100000181	view	1	25.350000381469727	electronics.telep...	milavitsa
100000743	view	1	24.450000762939453	kids.toys	vega
100001988	view	1773	98.54140423583985	electronics.audio...	adagio
100001988	purchase	23	98.54140423583985	electronics.audio...	adagio
100001988	cart	79	98.54140423583985	electronics.audio...	adagio

Output Data: Recommendations

user_id	event_type	avg_event_price	stddev	lower_bound	upper_bound	product_id	event_count	avg_price	category_code	brand	rank
628167977	purchase	393.0672720128839б	180.67117508017432	302.7316844727968	483.4028595529711	5100337	2235	328.2105505987747	electronics.clocks	apple	1
628167977	purchase	393.0672720128839б	180.67117508017432	302.7316844727968	483.4028595529711	1307555	1882	343.44943201958273	electronics.audio.headphone	asus	2
628167977	purchase	393.0672720128839б	180.67117508017432	302.7316844727968	483.4028595529711	5100689	1195	331.94396912712415	electronics.clocks	apple	3
628167977	purchase	393.0672720128839б	180.67117508017432	302.7316844727968	483.4028595529711	1307566	845	418.63087480666496	electronics.audio.headphone	acer	4

product_id	category_code	brand	avg_price	users
1307545	electronics.audio.headphone	lenovo	279.2772795800501	['512389317', '513696407', '514688413', '494701812', '512571292', 'coldster1', 'coldster2']
100011103	electronics.audio.headphone	acer	377.43323713953	['512389317', '513696407', '514688413', '494701812', '512571292', 'coldster1', 'coldster2']
100170834	electronics.audio.headphone	hp	271.05239046130697	['512389317', '513696407', '514688413', '494701812', '512571292', 'coldster1', 'coldster2']
5100855	electronics.clocks	apple	552.3748293541594	['512389317', '513696407', '514688413', '494701812', '512571292', 'coldster1', 'coldster2']

Price Point Analysis

Use Case Definition :- **Quantitative estimation** of the user's ***purchasing power***.

Motivation :- Different users purchase products belonging to ***different classes*** (cheapest/budget/mid-range/top-line) based on their ***capacity and preference***.

We regard ***purchase history*** as a strong ***metric*** of the user's ***behavioural aspects*** and try to quantify the same (price_point).

$$price\ point = \frac{(product\ price\ - category\ mean\ price)}{category\_StdDev}$$

Clean Parquet Dump

Category Statistics		
category_code	category_mean_price	num_products

Product Price Point Analytics Store		
product_id	category_code	product_price_point

User's Price Point Analytics Store		
user_id	user_price_point	purchase_count

i. Calculate ***mean*** and ***standard deviation*** of prices for ***category***.

ii. Calculate the ***price point of a product*** w.r.t. its category.

iii. ***Quantify*** the ***behavioural aspect of a user*** based on price points of products purchased.

Price Point Analysis

Category	UserIdX	UserIdY
electronics-mobile	iPhone13 - Rs. 95,000	Redmi Note 11 - Rs. 18,000
electronics-tv	Sony Bravia TV - Rs. 1,30,000	LG LCD TV - Rs. 34,000
apparel-shoes	Nike Air Jordan 1 - Rs. 16,000	Bata Casuals - Rs. 1700

Big spends

Budget buys

Recommendations using Price Point Analysis

```
query_PPA2(["568782581", "512480149"], "electronics.smartphone")
```

Query product recommendations for one or more users,
from a particular category or in general.

```
User-568782581
Cold Start User
User-512480149
+----------+--------------------+
|product_id|category_code       |
+----------+--------------------+
|1003532   |electronics.smartphone|
|1003705   |electronics.smartphone|
|1004777   |electronics.smartphone|
|1004732   |electronics.smartphone|
|1003989   |electronics.smartphone|
|1004740   |electronics.smartphone|
|1004819   |electronics.smartphone|
|1003548   |electronics.smartphone|
|1003707   |electronics.smartphone|
+----------+--------------------+
```

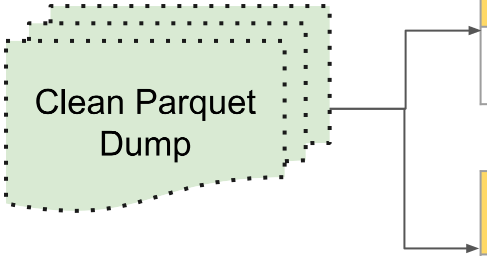
Market Basket Analysis

Use Case Definition :- A technique to uncover ***associations between products***.

Motivation :- Improving on top of the basic analytics stores. Analysing to provide recommendations of the form "**Frequently bought together**" or "**Customers also bought**."
Market Basket Analysis fits our requirements well.

Associations made between products ***based on*** their -

- **Purchases -** product purchases per user, product purchases per user session
- **Views -** product views per user, product views per user session



User Statistics		
user_id	user_product_purchases	user_product_views

User Session Statistics		
user_session	session_product_purchases	session_product_views

Market Basket Analysis

A set of measures that show what ***combinations of products*** occur together in orders most frequently.

$$Support = \frac{frq(X,Y)}{N}$$

Measure of how frequent the itemSet appears in the dataset

$$Confidence = \frac{frq(X,Y)}{frq(X)}$$

Measure of how often Y is present, given X is already in the dataset

Analytics mentioned below prepared from the clean parquet dump

- Our own **associative products confidence scores** for two products
- Spark ML based **FP-Growth data mining** model to build rules and **predict the product** that can be added to a new/unseen product combination(s).

INPUT

user_id	uniq_prod_history
441522689	[1004838]
461023190	[14100275]
470193237	[1004428]
512385518	[1004775]
512386977	[8700025, 1004657...

OUTPUT

Frequent Itemsets

items	freq
[1004856]	19228
[1004767]	14410
[1005115]	8352
[1004833]	8340
[4804056]	7563

Association Rules

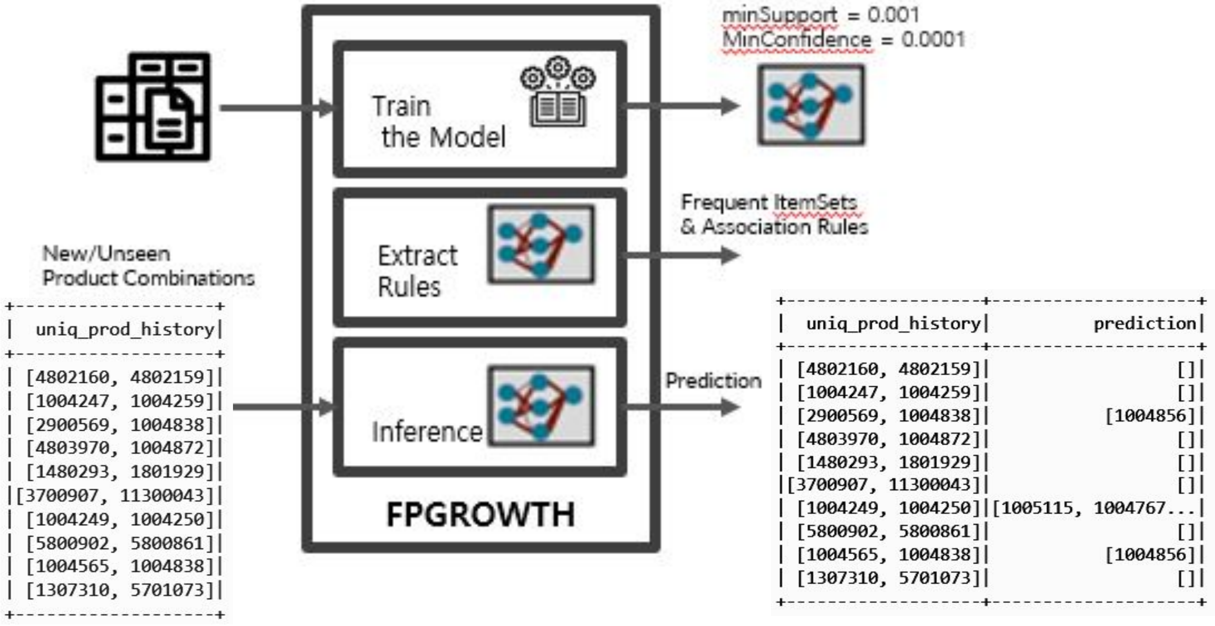
antecedent	consequent	confidence	lift
[1004870, 1004856]	[1004767]	0.471	8.618
[1004833, 1004767]	[1004856]	0.408	5.584
[1004767, 1004856]	[1004833]	0.262	8.260
[1004833, 1004856]	[1004767]	0.262	4.787
[1004870, 1004767]	[1004856]	0.242	3.322

ML Model to predict product mix

Motivation :- Moving past the analytical features such as price-point analysis, personalized recommendations and association rules to recommend products,
We wanted a ***prediction model*** *on* ***unseen/new product combinations*** to complete our product recommendations.

We used the ***frequent pattern mining FP-Growth*** algorithm supported in ***PySpark*** to build our model.

The hyperparameters of "***minSupport***" and "***minConfidence***" were manually tuned to find the best value for our dataset.



Personalized Placement Analysis

Use Case Definition :-

- **Product Rank** - Ranking position of any product in the organic eCommerce search results.
- **Personalized Product Rank** - Incorporate **user preference** to get revised rank

Motivation :-

- **Enhance** user experience by giving a **personalized** touch
- **Higher conversion rate** to drive enhanced sales

Analytics

- Generate **organic product rank** based on conversion rates
- Generate user preference based on **adjustment factor** (brand purchase count/total purchase count)
- Adjustment factor multiplier on organic product rank to get **revised product rank**

Adjustment Factor (AF)
Brand Purchase Count (BPC)
Total Purchase Count (TPC)
AF = BPC/TPC

User Statistics				
User ID	Brand	BPC	TPC	AF

Product Statistics		
Product ID	Brand	Organic Rank

Product	Organic Rank	Personalised Product Rank
Samsung S21	1	3
iPhone 14	2	1
Google Pixel 7	3	2

“Underdog” Product Recommendations

Use Case Definition :-

- Push “**Underdog**” products – Products with lesser views but **high conversion rate**

Motivation :-

- **Boost** products which are **relatively unexplored**,
  but when viewed are **more likely converted** into a sale
- Product visibility drives business sales

$$\mathit{Underdog\ Product\ Ids}\ (\mathit{UPiD})$$
$$\mathit{Product\ Ids}\ (\mathit{PiD})$$
$$\mathit{if}\ (\mathit{PCRi} > x\%\ \mathit{of}\ \mathit{CCRi}\ \&\&\ \mathit{PVCi} < y\%\ \mathit{of}\ \mathit{CVCi})$$
$$\mathit{UPiDs.append}(\mathit{PiD})$$

Analytics

- Generate Product Conversion Rate (**PCR**) & Product View Count (**PVC**)
- Generate product’s Category-average Conversion Rate (**CCR**) and Category-average View Count(**CVC**)
- Products with (**PCR > x% of CCR**) && (**PVC < y% of CVC**) are underdog products

```
[26] # for 10/2019
     product_master_df.count()

     16737

[27] underdog_df_v1.count()

     861
```

category_code	product_id	PVC	PCR	CCR	CVC
computers.periphe...	9200019	26	115.385	11.896	113.138
computers.periphe...	9200679	21	95.238	11.896	113.138
computers.periphe...	9200024	41	73.171	11.896	113.138
computers.periphe...	9200516	17	58.824	11.896	113.138
computers.periphe...	9200220	11	90.909	11.896	113.138
computers.periphe...	9200578	32	125.000	11.896	113.138
computers.periphe...	9200717	46	43.478	11.896	113.138
computers.periphe...	9200359	48	41.667	11.896	113.138
appliances.kitche...	14500009	88	45.455	11.544	238.229
appliances.kitche...	14500070	72	55.556	11.544	238.229

Benchmarking

# Months	Ram (GB)	# Cores	Purchase Threshold	View Threshold	# Users	Time (min)
1	2	2	4	30	20	3
2	12	2	10	200	120	3
2	40	12	30	100	120	1
4	40	12	10	200	500	0.8
4	40	12	10	200	2050	2.5
6	40	12	10	200	4000	6.5

Recommendation Benchmark

Running the recommendation pipeline for :-

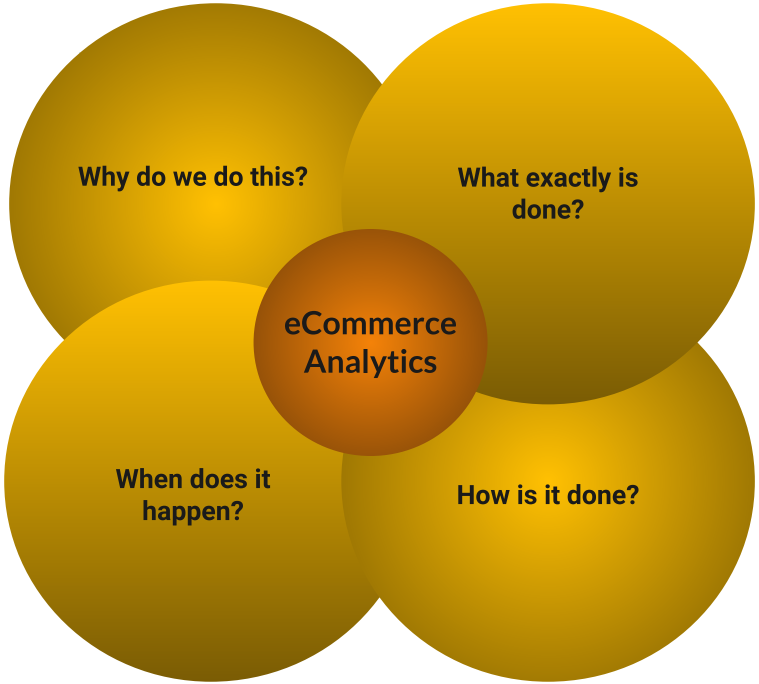
- Different numbers of users
- Different threshold values
- Various system configurations.

# Months	Ram (GB)	# Cores	Time (min)
1	8	2	13
2	8	2	28
3	8	2	42
4	8	4	66
4	40	12	8

ETL Benchmark

Running the ETL pipeline under various system configurations.

Summary



Exploratory Data Analysis

Identifying our scope

- Cleaning and Normalization
- Sales Trend Insights
- User behaviour patterns

Value Addition

Aimed at business and customer growth

- Personalized Product Recommendations
- Price-point analysis
- Market Basket Analysis
- Promoting “Underdogs”
- Rank personalizations

Keeping it scalable

The platform grows, and with it, our system

- Successful analysis on 55 GB of eCommerce data
- Promising performance results

More ideas enroute

Ever-growing analytics and possible options

- Explore advanced ML techniques
  - Matrix factorization
  - TensorFlow Garden NeuMF
- Advanced analytics
  - Traffic analysis
  - Price effect on sales

Thank You

And best of luck!