PROBLEM STATEMENT

Definition

Analyzing e-Commerce customer behavior to provide recommendations for a better experience using Apache Spark Dataframes. The data being used is eCommerce user behaviour data from kaggle which consists of user's event history on the platform.

Motivation

User behavior data refers to the data collected by e-commerce platforms about the actions and activities of their users on their website or app. This data can include information such as the pages a user visits, the products they view, the searches they perform, and the actions they take, such as adding items to their cart or making a purchase.

Having access to this type of data can be incredibly useful for e-commerce platforms, as it can help them better understand their users and tailor their services to meet their needs. For example, by analyzing user behavior data, an e-commerce platform can identify the most popular products among its users, as well as the factors that influence a user's decision to make a purchase. This information can then be used to optimize the platform's product offering and improve the user experience.

User behavior data can also be used to personalize the shopping experience for individual users. By analyzing a user's behavior data, an e-commerce platform can make recommendations for products that are likely to be of interest to that user based on their previous actions and interactions with the platform. This can help to increase engagement and drive sales, as users are more likely to make purchases when they are presented with personalized recommendations that are relevant to their interests and preferences.

In addition to helping e-commerce platforms understand and serve their users, user behavior data can also be used to identify potential issues and improve the overall performance of the platform. For example, by analyzing user behavior data, an e-commerce platform can identify areas of the website or app that are causing users to become frustrated or confused and make changes to improve the user experience. This can help to reduce user churn and increase customer satisfaction, which are crucial factors for the success of any e-commerce platform.

Overall, user behavior data is a valuable resource for e-commerce platforms, as it can help them understand their users and provide a better shopping experience. By using this data to optimize their product offering, personalize the user experience, and identify potential issues, e-commerce platforms can improve their performance and drive business growth.

Design Goals

E-commerce data is a prime example of big data, as it includes many transactions, customer interactions, and other types of data. This data can be challenging to analyze using traditional methods, so big data processing tools like Apache Spark are needed to process and analyze it effectively. Also, in an ideal setup, there can be many teams working in the Analytics department, each formulating a way of recommending products to the users which have the highest chance of purchase. Each of these teams would need to have access to some kind of data. Having a centralized data pipeline is very much required in this situation. The ETL pipelines would have to strategically process the data in a way that they meet every team's requirements. All these factors have been taken into consideration while developing the modules.

Features Required

The following features have been built to support the analysis we have performed for the last four weeks.

1. Centralized ETL Pipeline

2. Compound Analysis Recommendations
3. Price Point Analysis Recommendations
4. Market Basket Analysis Recommendations
5. Personalized Placement Recommendations

Scalability and Performance Goals

There are several ways a big data platform can scale, depending on the specific needs and goals of the organization using it. One approach is to use a distributed architecture, which involves dividing the data and processing it across multiple machines or nodes. This can help improve the performance and scalability of the platform, as it allows for parallel processing of data and enables the system to handle large amounts of data more efficiently.

This strategy perfectly fits for the current use-case because the number of customers and events on the e-Commerce platform can dramatically increase. In case of a rise in the number of events/transactions, the platform must ensure that it is able to meet the demands of all the customers. From the platform point of view, this includes that all the features that were discussed in the above section are able to loosely scale even in the event of sudden rise in traffic. Since we have used Apache Spark as the base processing tool, it provides access to certain parameters like number of cores, executors, and memories, among many other tunings which can help scale the platform in such cases.

SOLUTION APPROACH

High Level Design

We have followed a modular design for developing a solution to the problem statement. Modular design refers to the practice of breaking down a complex problem statements into smaller, independent modules or components. Each module performs a specific task and can be easily reused in different parts of the program or in other programs. This approach makes the code more organized, maintainable, and scalable. It also allows for easier testing and debugging, as individual modules can be tested and modified independently.

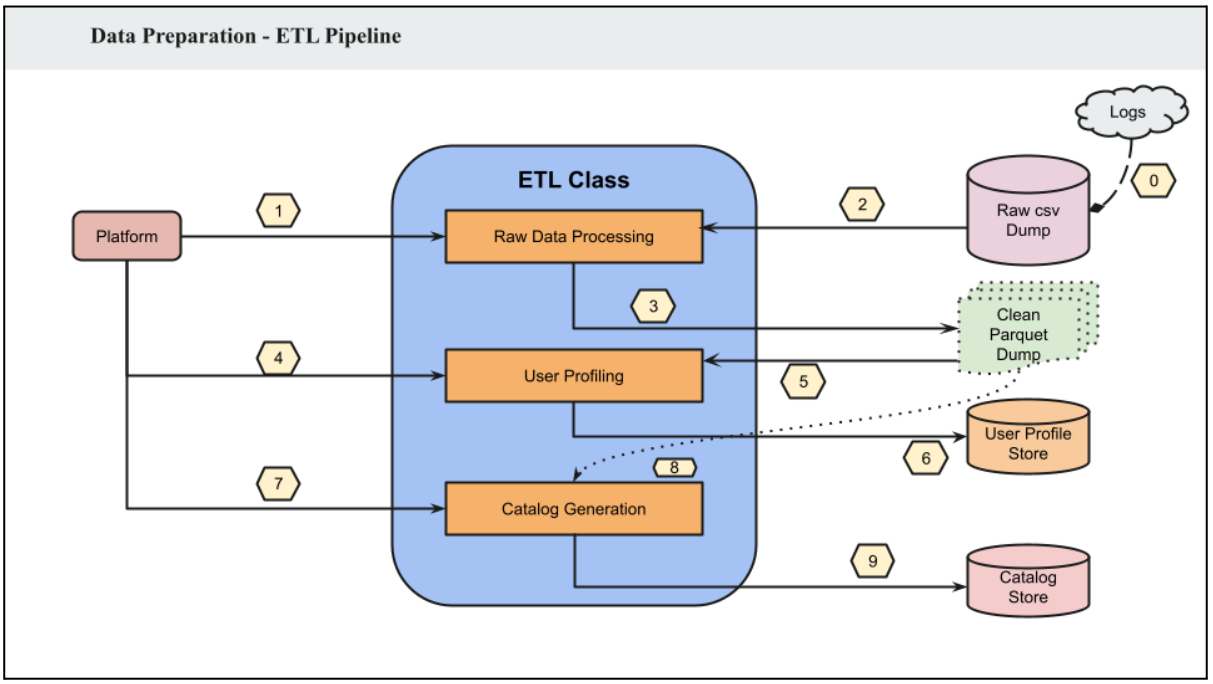
The solution comprises of two key modules:

1. ETL Engine
2. Recommendation Engine

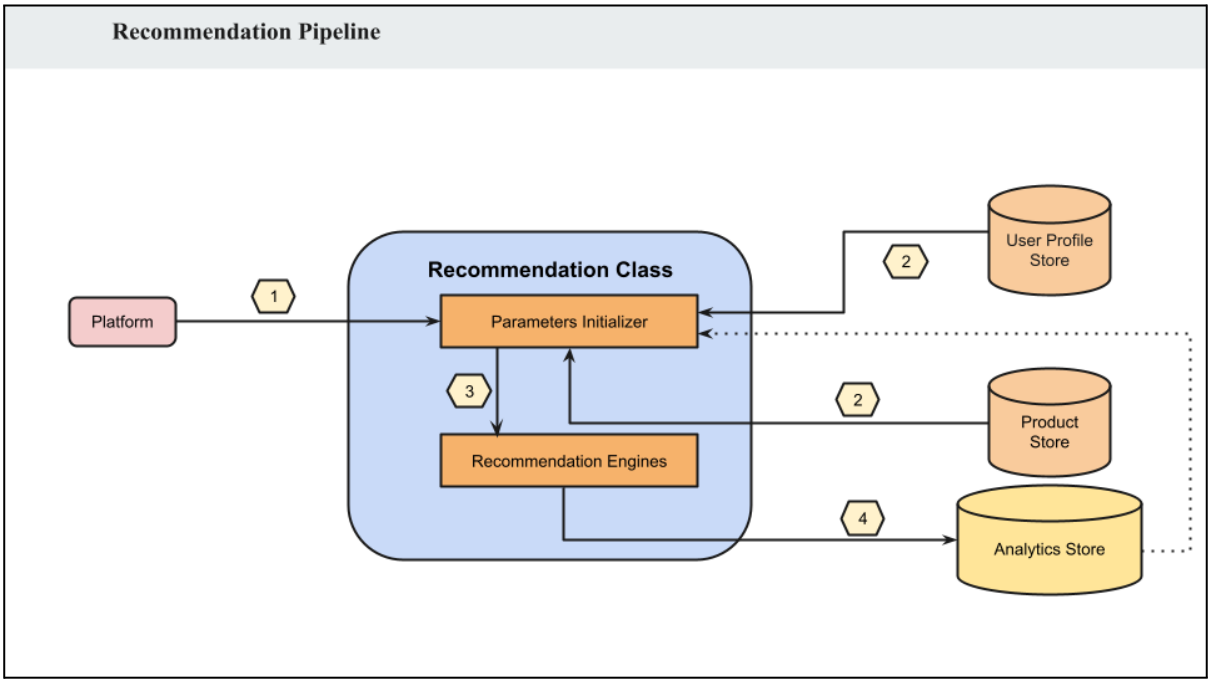
The ETL Engine: The extract step involves extracting data from the raw dumps. The transform step involves cleaning, filtering, and transforming the data to a consistent format and structure. The load step involves loading the transformed data into a data warehouse or database (in suitable formats such as transaction logs, customer profiles, and product catalogs) for further analysis and reporting. ETL is typically performed using the specialized ETL tool: Apache Spark, to ensure efficient and reliable data processing.

Recommendation Engine: The data generated from the ETL is analyzed to identify patterns and trends in customer behavior, such as preferences and interests. Based on these insights, the recommendation engine generates personalized recommendations, such as product or content suggestions, to individual customers. These recommendations are typically displayed on the website or app, and can improve customer engagement and conversion rates. We have used various methods of recommendations for various types of customers.

Architecture Diagram



The ETL class consists of the necessary sub modules required for processing the data for specific requirements. The three key responsibilities of the ETL Class are: Raw Data Processing, User Profiling, Catalog Generation. In the figure above, the numbers signify the sequence of order of events. The logs are continuously dumped as csv files. If the platform requests for processing the newly received data, the Raw Processing Module fetches the raw csv files and does the necessary processing like correcting the schema, deleting the null values from the data. It then saves the data in parquet format partitioned by date. Upon receiving a request for generating user profile, the User Profiling module fetches the necessary parquet files and does some aggregation operations on the data based on user identification column and stores the user profile in the User Profile store. Similarly, the Catalog Generation module saves the product related data.



The Recommendation Class consists of Parameter Initializer and Recommendation Engine Modules. Parameter Initializer module is responsible for reading the data from the necessary data sources and the Recommendation Engine hosts the algorithms required for providing the recommendations. The recommendations are then sent to an Analytics Store.

Data Model

Raw Data

Parquet Dump

User Profile => for compound analytics

X Factor Dump:

1. For User
2. For Product

Product Catalog

Asociation Rules

Preference Rank

Compound Analysis Recommendations

Price Point Analysis Recommendations

Big Data Platforms

This project has used Apache Spark Dataframes for processing and analyzing the data. We have also used some Spark ML algorithms like K-Means Clustering to understand the user groups and segment the users as part of EDA. As part of Market Basket Analysis, we have used FP Growth Algorithm from Spark ML-lib.

EVALUATION

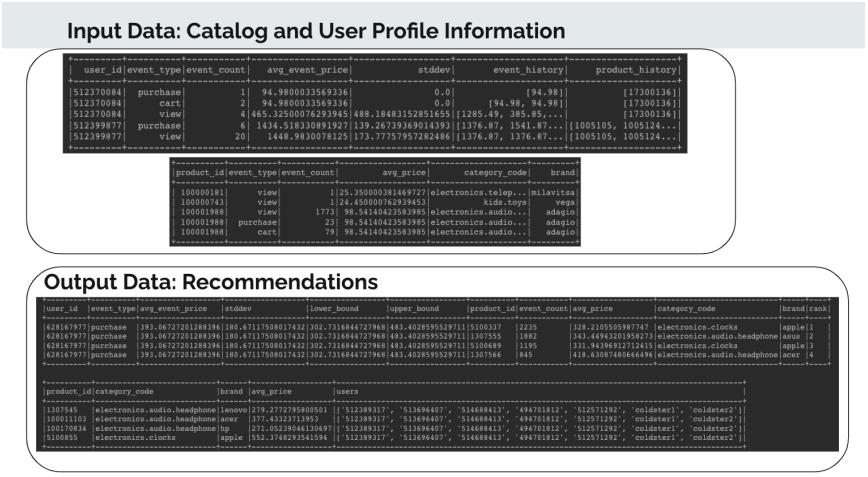
Experiment Design

1. Personalized Product Recommendation: To provide product recommendations to the users based on their average spendings or views on the portal

Motivation: We need a more robust engine that could provide personalized recommendations rather than simply recommending products that sold the most.

Analytics:

- Using the given the threshold values, the engine categorizes the users into **purchase, view or cold-start**
- Top products are picked based on the average price of the user.
- This ensures that the probability of conversion of the event from “view” to “purchase” is high.
- The recommendations are done for Homepage (All categories) or Category page(Example: Electronics)



2. Underdog Product Analysis: To find out products with low views but high conversion rate.

Motivation:

- Boost products which are unexplored but when visible mostly leads to a sale
- Product visibility is an important analytics tool to drive business sales.

Analytics

- Generate product conversion rate (PCR) & product view count (PVC)
- Generate product's category average conversion rate (CCR) and average category view count (CVC)
- If (PCR > x% of CCR) && (PVC < y% of CVC) are underdog products

3. Personalized Product Rank:

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

4. Market Basket Analysis

Definition: A set of measures that show what combination of products most frequently occur together in orders.

Motivation

- A standard technique to uncover association between items.
- It fit well to our need and was MUST to build a good recommendation platform.
- We had the information available to build association between products such as
- Product Purchases - product purchases per user, product purchases per user session
- Product Views - product views per user, product views per user session

Analytics:

We built below analytics on top of the ETL data generated in user and product stores.

1. Our own associative products confidence scores for two products

2. Spark ML based FPGrowth data mining model to build rules and predict the product that can be added to a new/unseen product combination(s).

5. Price Point Analysis:
The Aim: 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 behavioral aspects and try to quantify the same (price_point).

Analysis:

- Calculate mean and standard deviation of prices for categories.
- Calculate the price point of a product w.r.t. its category.
- Quantify the behavioral aspect of a user based on price points of products purchased.

Scalability Metrics

Since the data we have is for around 9 months, we ran the ETL jobs for various counts of months to generate the analysis for the last 'k' months. The ETL module scaled as follows:

User Profiling

Months	Ram in GB	Cores	Time in Minutes
1	8	2	13
2	8	2	28
3	8	2	42
4	8	4	66
4	40	12	8

Recommendation Profiling

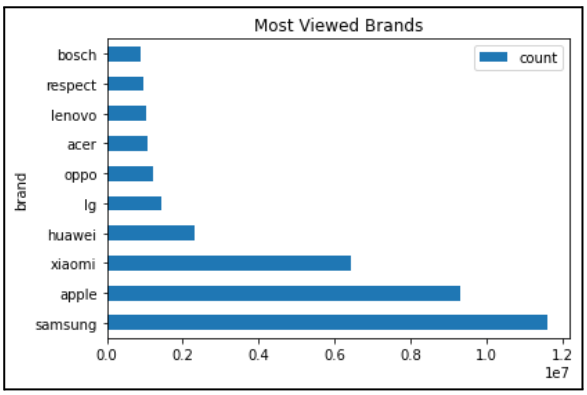
Months	Ram in GB	Cores	Time in Min	Purchase Threshold	View Threshold	#users
1	2	2	3	4	30	20
2	12	2	3	10	50	120
2	40	12	1	30	100	120
4	40	12	0.8	10	200	500
4	40	12	2.5	10	200	2050
6	40	12	6.5	10	200	4000

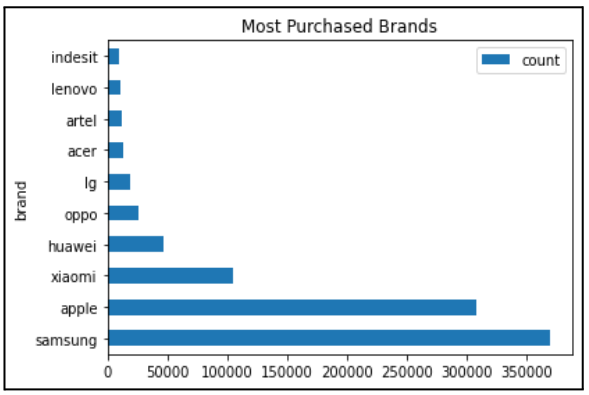
We also ran the user recommendation script for various counts of users and it scaled as follows:

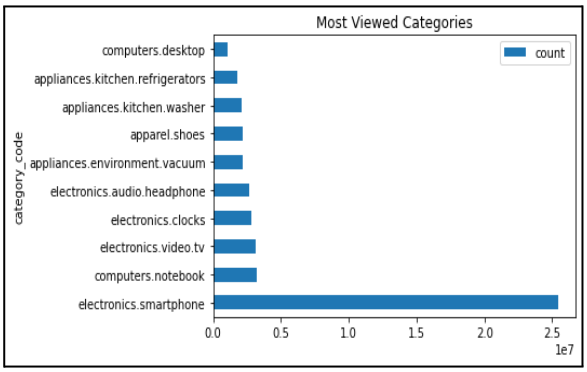
#TO-DO

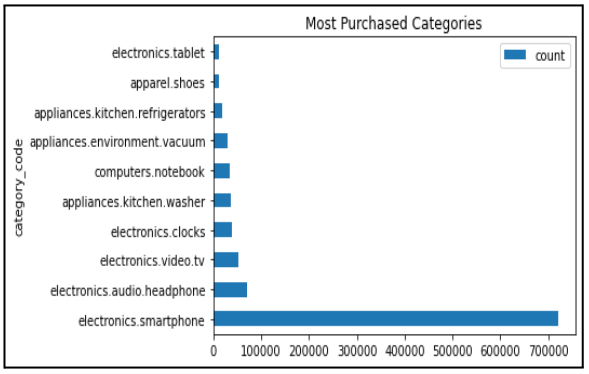
Plots and Analysis - EDA

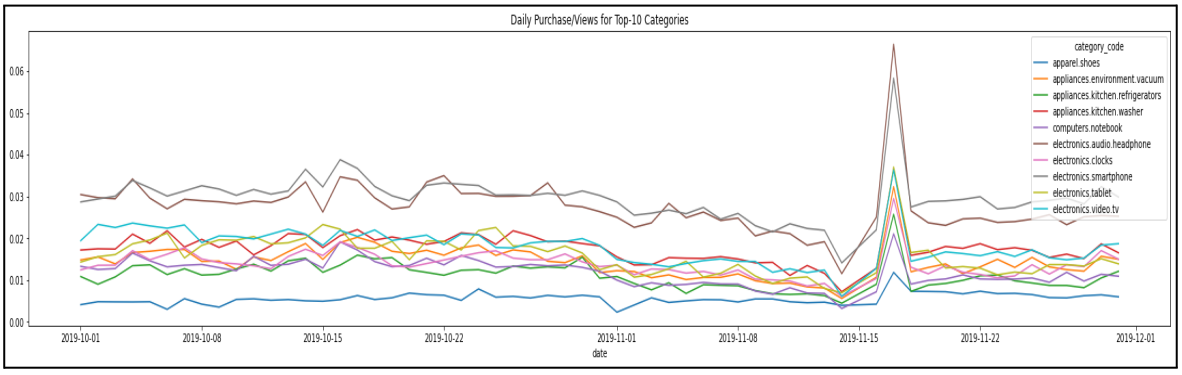
As part of EDA, we have thoroughly studied the data to understand the columns and their related information. Below are some of the charts that showcase the preliminary analysis of the dataset.



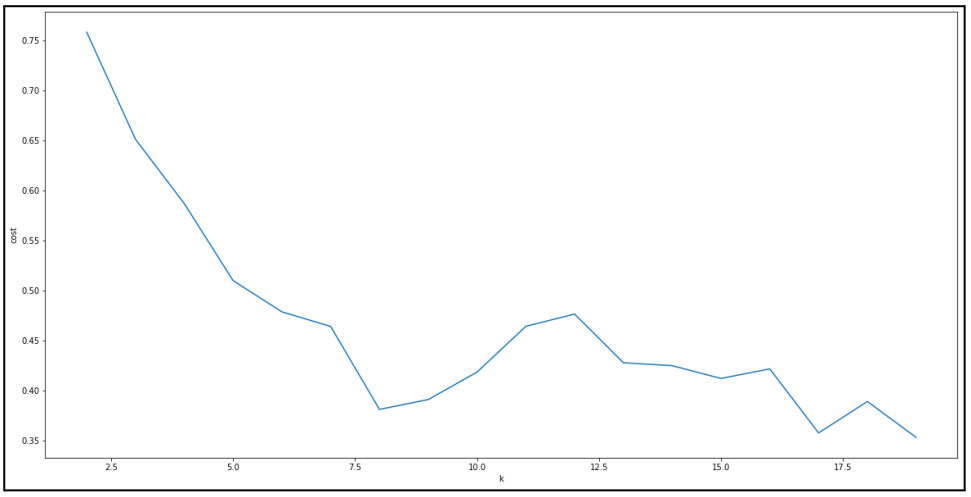






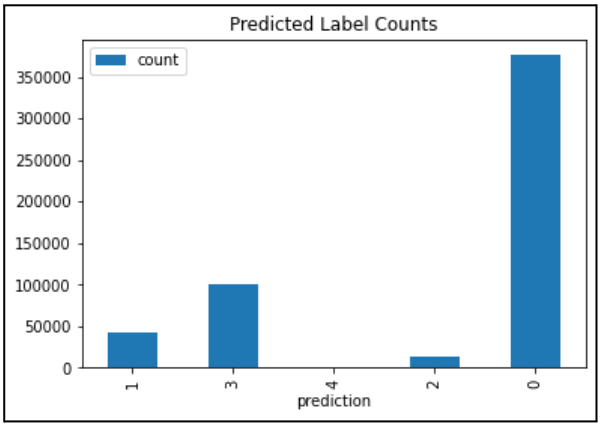


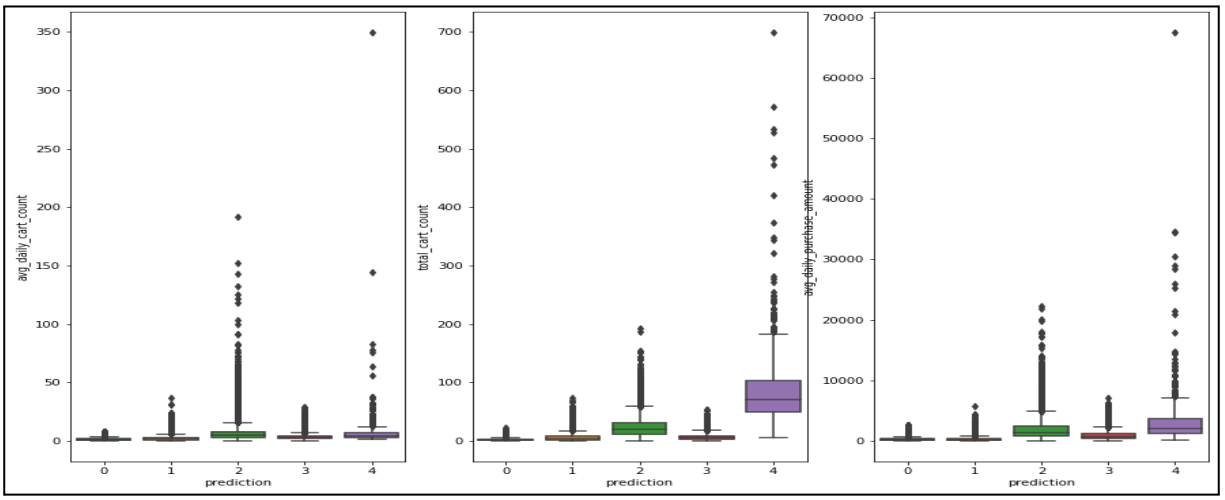
Customer Segmentation:



Above is the plot of k (Number of clusters) vs cost (silhouette score) for customer segmentation.

Below is the count of users in each cluster for k=5. (#4 cluster has very less number of users)





SUMMARY

Through this project, the team has understood how recommendation systems are an important feature of ecommerce platforms, and how they are designed to help users discover new products that they might be interested in purchasing. Our recommenders analyzed a user's past purchasing history, browsing behavior, to make personalized recommendations for products that the user might like.

The major goal in the project proposal was to do some compound analysis on the behaviour data to bring out some interesting insights. Through these recommender systems and the analysis, we believe we have fared well. Our solutions showcased some scaling as well. We have also used Spark ML packages to use some algorithms like K-Means for customer segmentation, FPGrowth for mining the associations between products for each user. These algorithms have helped us do the Market Basket Analysis, which give an insight into how the association rules can help recommend products which have highest probability of being bought together.

As a future study, we can build some ML based recommenders using techniques like collaborative filtering, and content based similarity recommenders. A comparison between current recommenders and the ML recommenders can also be taken up to gauge the performance of the models.