PRODUCT RECOMMENDATION

Black Friday Sales

Presented by
Solleti NagaRanjith Kumar
Shyam Sundar
Sowndarya S
Nivethithaa S

Contents

BLACK FRIDAY SALES

ABSTRACT

The main idea of this project is to recommend products using Black Friday sales transactions, which is regarded as the first day of the Christmas shopping season, on which retailers make many special offers. The dataset here is a sample of the transactions made in retail store. The store wants to get an idea about customer purchase behaviour against different products. The store has a loyalty program and wants to provide offers on products during non- sale period on certain categories of interest. Since the dataset contains both numerical and categorical values, it is difficult to impute the missing values or to remove outliers because outliers in terms of , say ,Purchase is significant . Based on this, we recommended products to customers by segmenting them based on the FM score using K-means clustering technique and finding how similar they are using Jaccard similarity coefficient to recommend product categories. A simple method using Linear algebraic methods of Matrix multiplication to create co-occurrence matrix was finally done to recommend top 100 products .By offering loyalty rewards using those products to categories of our interest we intend to reduce the customer retention and thereby build brand relationships.

INTRODUCTION

Every year the Friday after Thanksgiving is a day of shopping. Known as Black Friday, the day is characterized by high demand for retail goods. The analysis of the Black Friday dataset with respect to summary statistics are "The dataset is comprised of 537,577 observations about Black Friday shoppers in a retail store, it contains different kinds of variables numerical as well as categorical with missing values."

PROBLEM STATEMENT

According to research done by Frederick Reichheld of Bain & Company, acquiring a new customer can be **five times as expensive** as keeping an existing one and by increasing customer retention rates by 5% **increases profits by 25% to 95%.** A quality, rewarding loyalty program is quickly becoming a necessity to keep customers coming back so as to build brand relationships. A loyalty program isn't a quick fix for low sales volume. It's a long-term investment in better relationships with your customers. To do so you need to develop clear goals for your program and identify shopper needs and behaviors. Our objective is to develop accurate, comprehensive data analysis to scrap features customers don't respond to and enhance features that drive real results. It's crucial to analyze how customers interact during the high-volume holiday season, enabling you to make improvements for the next year.

DATASET DESCRIPTION

Following is 12 variables description

User_ID: Consumers took a part in the black Friday sale event. There are **5891** unique users participated in the sale.

Productid: Products sold in the event. There are **3623** unique products sold in the black Friday event.

Age:

Age 7 of groups participated in the sale event.

Age	Count	Description
0-17	218	Teenagers
18-25	1069	Youth
26-35	2053	Working class people
36-45	1167	
46-50	531	
51-55	481	Citizens
55+	372	Senior Citizens

Gender:

Sex	Count
Male	4225
Female	1666

Occupation:

There are 21 categories of occupation participated in the event. Occupation starts from range 0 to till 21. Problem in Occupation variable to define Occupation type.

Occupation	Count	Occupation	Count
0	688	**11**	128
1	517	**12**	376
2	256	**13**	140
3	170	**14**	294
4	740	**15**	140
5	111	**16**	235
6	228	**17**	491
7	669	**18**	67
8	17	**19**	71
9	88	**20**	273
10	192		

City Category:

Black Friday sale held in 3 type of city category.

City Category	Count	Description
A	1045	Rural
B	1707	Urban
C	3139	Metropolitan

Stay_In_Current_City_Years:

Totally 5 type of people under Staying current city years participated in the sale event.

StayInCurrentCityYears	Count	Description
0	772	New Residents
1	2086	
2	1145	
3	979	
4+	909	Residents/Natives

Marital Status:

Two type of categories.

Type	Count	Description
0	3417	Married
1	2474	Unmarried

Product_Category_1:

18 unique products are available in product category 1.

Product_Category_2:

17 unique products are available in product category 2. **166986 Null values** in the Product_Category2.

Product_Category_3:

15 unique products are available in product category 3. **373299 Null values** in the Product_Category3.

Purchase (Dollars):

Purchase done by the **5891 consumers is 5,017,668,378.00** Dollars in the black Friday sale.

If we look at the first few rows of our dataset, we can see that each row represents a different transaction, or item purchased by a specific customer. This will come into play later when we group all transactions by a specific *User_ID* to get a sum of all purchases made by a single customer

EXPLORATORY DATA ANALYSIS

The idea is to plot and visualise different attributes and finding the insights based on its comparison

SUMMARY STATISTICS

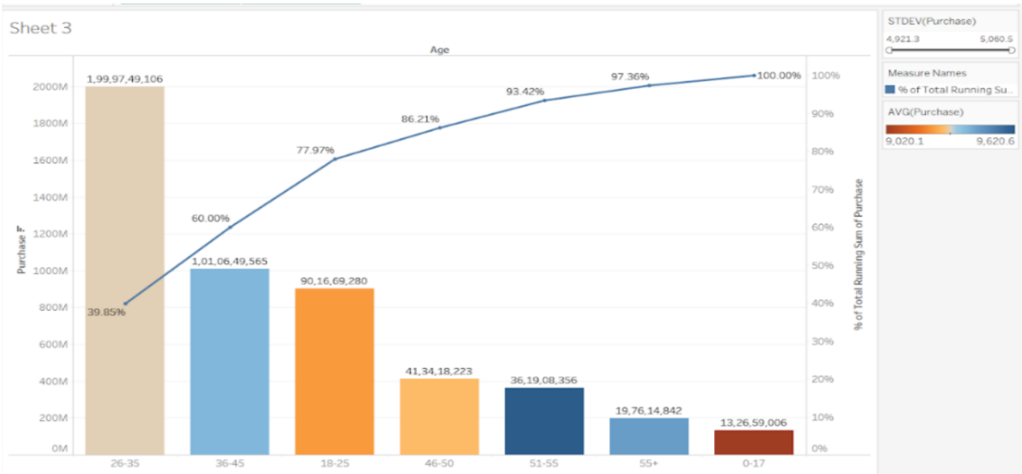
INFERENCE:

- Each customer purchased approximately **91** times
- The average time between his/her purchases is **15** mins
- The average customer value for his/her purchases is a whopping **$8,51,714**

  These metrics signify the customers perception about Black Friday sales season.

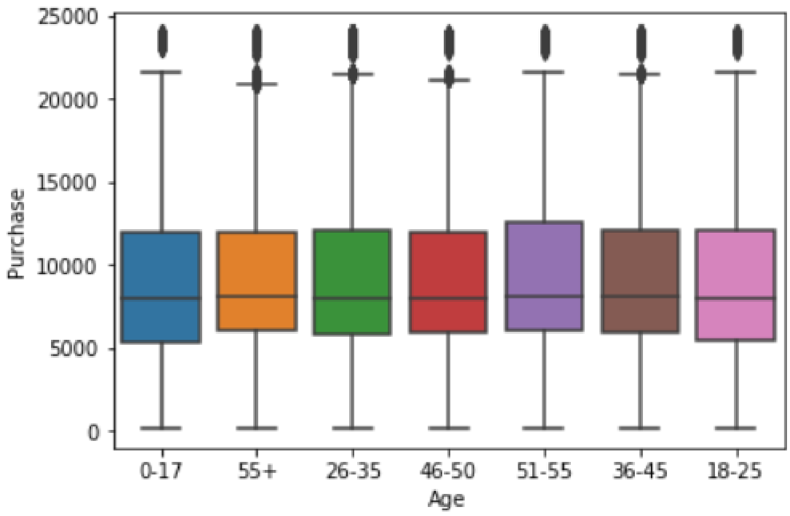
Bivariant Analysis:

Age Vs Purchase



Inference:

- **Ages 18-45** contribute to **77.97%** of the sales being the vibrant working age population, with the Age group 26-35 contributing 40% to the sales
- Average Purchase value is **$9333.0** with standard deviation of **$188.0**.



- It is interesting to note that there are outliers in every age segment and this information will need to be looked at while creating personalised recommendation to age groups

Gender Vs Purchase:

Gender Count

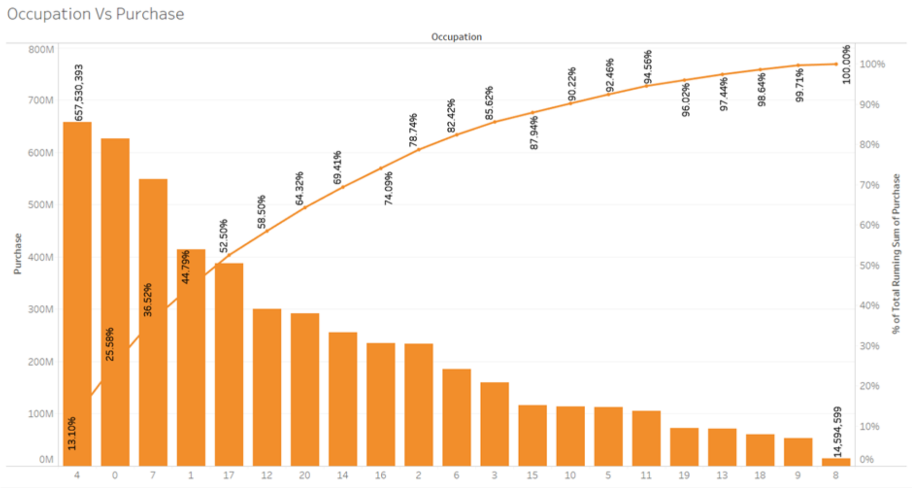
Gender	Avg. Purchase	Distinct count of User ID
F	8,810	1,666
M	9,505	4,225

- Average purchase done by male (9504.8) and female (8809.8) regardless of the count of gender.

Occupation Category Vs Purchase:

Inference:

- Minimum Average purchase is by customers of Occupation 9 and maximum average purchase by Occupation 17.
- More consumers are from occupation 4,0,1,7,17. Consumers from these occupation categories occupied 52.50% of total sale.
- Average Purchase from all occupation categories have less variance i.e range from 8714 to 9906 dollars.



Stay in current city years Vs Purchase:

StayinCityYearsCount

Stay In C..	Avg. Purchase	Distinct count of User ID
0	9,247	772
1	9,320	2,086
2	9,398	1,145
3	9,351	979
4+	9,346	909

Inference:

- Average Purchase of all type of people staying in the current city in years is almost equal range from 9247.2 to 9397.6 dollars regardless count of the people.
- Being a weekend sale, all type of residents are showing interest in the sale event.

Marital Status Vs Purchase

Married,Unmarried Count

Marital S..	Avg. Purchase	Distinct count of User ID
0	9,333	3,417
1	9,335	2,474

Inference:

- Average purchase of married and unmarried are almost the same with deviation of 0.924 dollars from the median (9333.9) dollars.

Insights on Products:

Product_Category_1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Product_ID																		
P00000142	0	0	1130	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P00000242	0	371	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P00000342	0	0	0	0	238	0	0	0	0	0	0	0	0	0	0	0	0	0
P00000442	0	0	0	0	92	0	0	0	0	0	0	0	0	0	0	0	0	0
P00000542	0	0	0	0	146	0	0	0	0	0	0	0	0	0	0	0	0	0
P00000642	512	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P00000742	0	0	0	0	238	0	0	0	0	0	0	0	0	0	0	0	0	0
P00000842	0	36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P00000942	54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P00001042	494	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

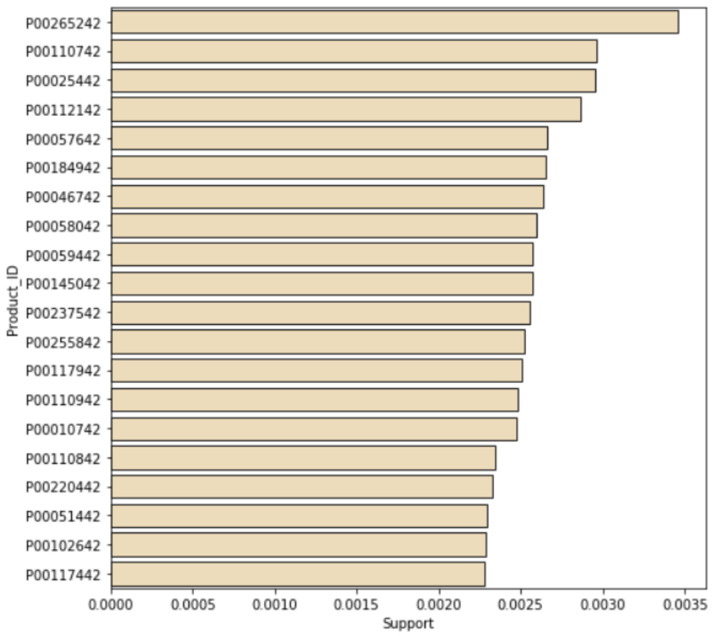
Each product belongs to only one category as denoted above having one value per row

User_ID	1000001	1000002	1000003	1000004	1000005	1000006	1000007	1000008	1000009	1000010
0	3589	3547	3594	3610	3517	3577	3607	3547	3566	3401
1	34	76	29	13	106	46	16	76	57	222

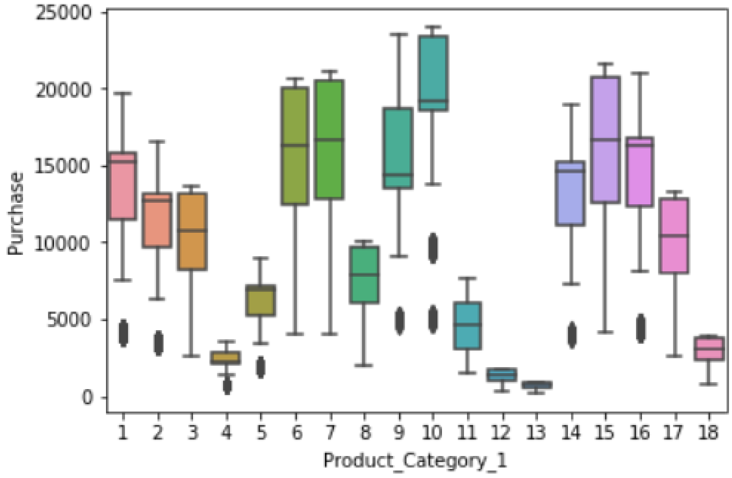
2 rows × 5891 columns

Each product has been bought only once as denoted above with Product count for each customer being either 0 or 1

Product Support :



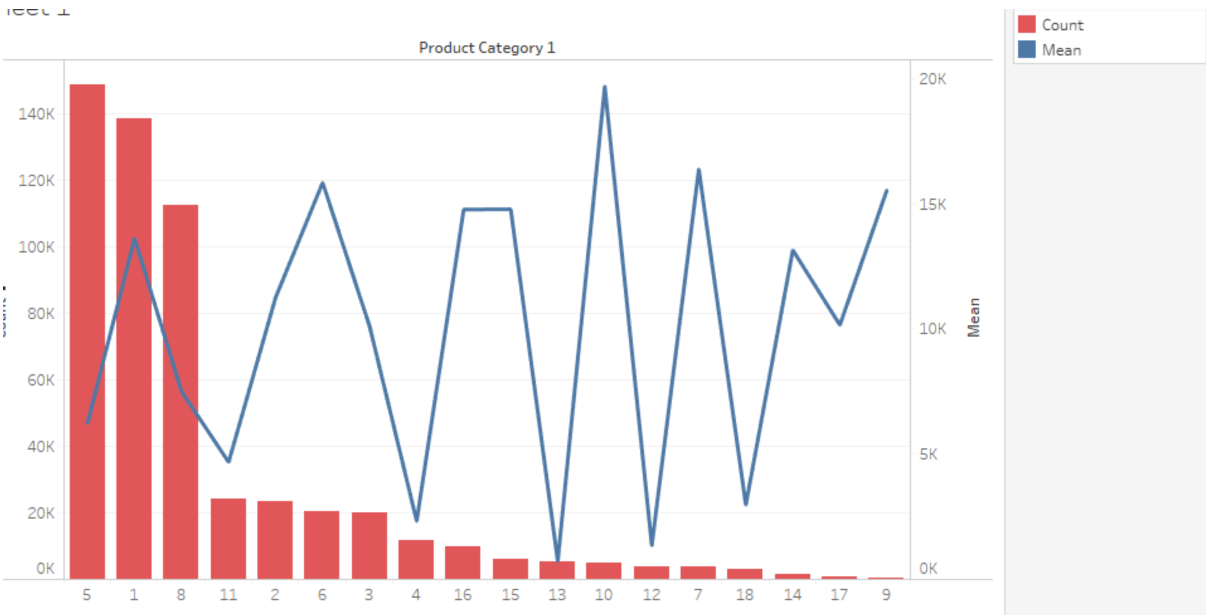
These are the top Products with the highest support which gives an idea about the popularity of the products with Product ID **P00265242** being the most popular one among customers



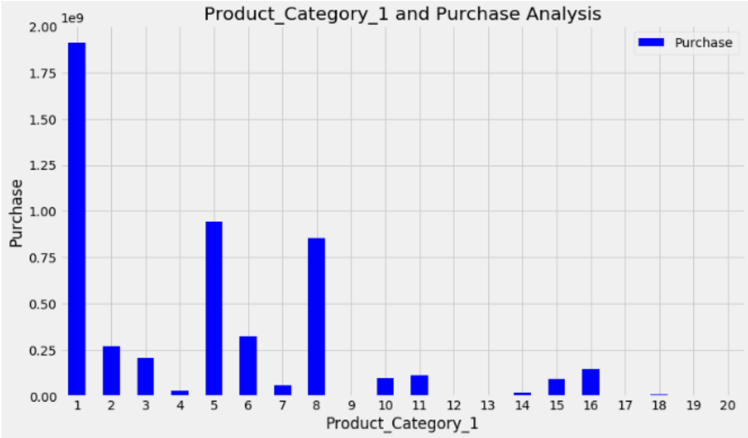
It is quite interesting to note that there seems to be cyclical peaks in the average Purchase amount of the product categories which might give us some insight about the ways in which these products were split into 18 categories

Product_category_1 VS Purchase analysis:

If you see the value spent on average for Product_Category_1 you see that although there were more products bought for categories 5,8 the average amount spent for those three is not the highest. It is interesting to see other categories appearing with high purchase values despite having low impact on sales number.



For examples, if instead of the average spent, we look at the amount spent on purchase, as illustrated in the chart below, that distribution that we saw for this predictor previously appears here. For example, those three product categories have the highest sum of sales since there were three most sold products. Revenue generating categories include Products belonging to **1,6,16,15,10,7** in that order. **Category 5** seems to be the most affordable segment with high sales as well as high revenue



We can see the same behaviour for the other two categories.

Frequency and Monetary Score

RFM_analysis is a customer segmentation technique that uses past **purchase behaviour** to divide customers into groups.

For Recency we need transaction date to track the user behaviour. Since that variable is not available in our data. we consider frequency of visiting and purchase value.

RECENCY (R): Time since last purchase
FREQUENCY (F): Total number of purchases
MONETARY VALUE (M): Total monetary value

RFM analysis was first used by the direct mail industry more than four decades ago, yet it is still an effective way to **optimize your marketing**.

RFM Customer Segmentation

To perform **RFM analysis**, we divide customers into Five equal groups according to the distribution of values for **recency**, **frequency**, and **monetary** value.
Four equal groups across three variables create 25 (5x5) different **customer segments**, which is a manageable number.

Frequency(F)	Monetary Value(M)
Quartile (F=1)	Quartile (F=1)
Quartile (F=2)	Quartile (F=2)
Quartile (F=3)	Quartile (F=3)
Quartile (F=4)	Quartile (F=4)
Quartile (F=5)	Quartile (F=5)

For example, let's look at a customer who:

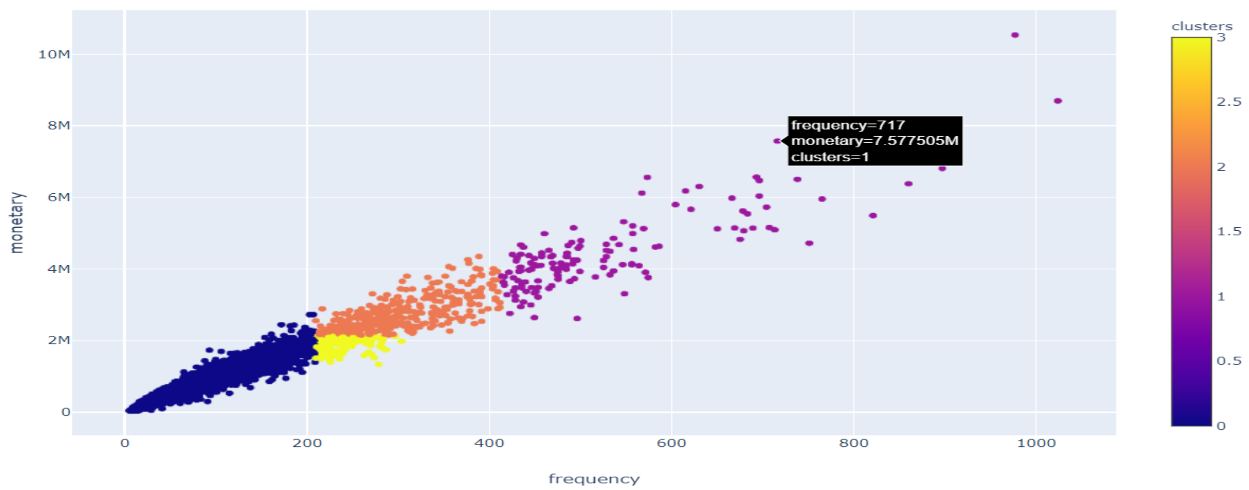
- is within the group who purchased **most quantity** (F=5)

- is within the group who **spent the most** (M=5)

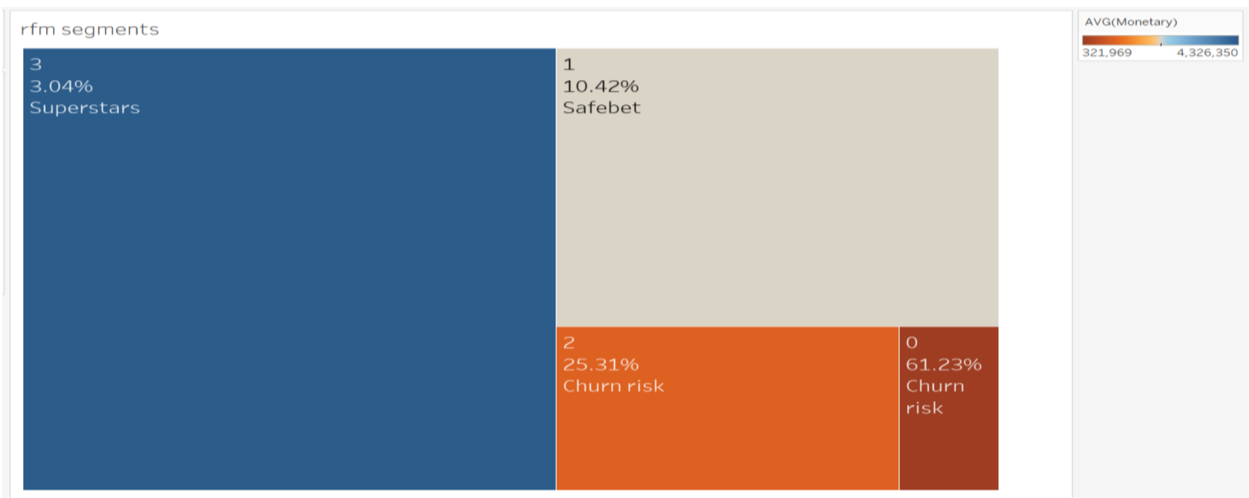
Clustering using F,M scores

We can use k-means clustering method to segment the users based on their corresponding frequency and monetary scores. This way we can get the categories of customers who are of interest to the retail manager and who is more valuable to the manager so that we can recommend top 100 products to these set of customers.

Plotting the customers based on FM scores is shown below:



As it can be seen , the most valuable customers are in the top right corner with highest frequency as well as monetary scores , these customers are of interest to the retail store



To get a better sense of our customers the above Tile chart shows that our '**Superstar**' customers account for 3.04 % of our customer base and '**Safebet**' are those customers who have a F = 3 and M = 3 who account for 10.42 % of the customer base . The retail manager should focus on these customers aws part of the loyalty program to achieve better conversion rate as well as become more profitable

Exploratory Data Analysis using FM Score:

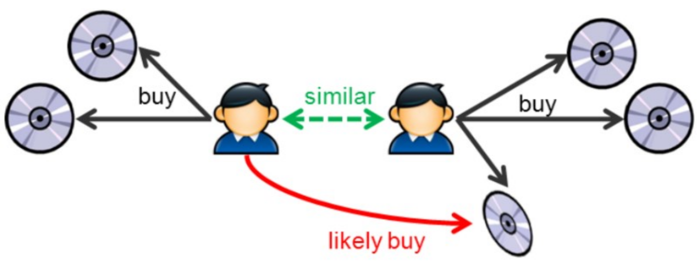
Purchase_Rank along User ID	1	2	3	4	5
Frequency_Ranking along User ID					
1	1470.0	728.0	143.0	1.0	NaN
2	3.0	145.0	595.0	133.0	NaN
3	NaN	11.0	142.0	608.0	129.0
4	NaN	NaN	3.0	142.0	748.0
5	NaN	NaN	NaN	NaN	890.0

- Most customers are From FM Score—11 followed by FM Score—55.
- Customers from FM Score (14,24) visited less frequently but made good purchase on the sales day.

Collaborative filtering

Now, that we have segmented the customers based on the FM score we can recommend product categories as a prototype to customers to check the effectiveness of theour recommendation

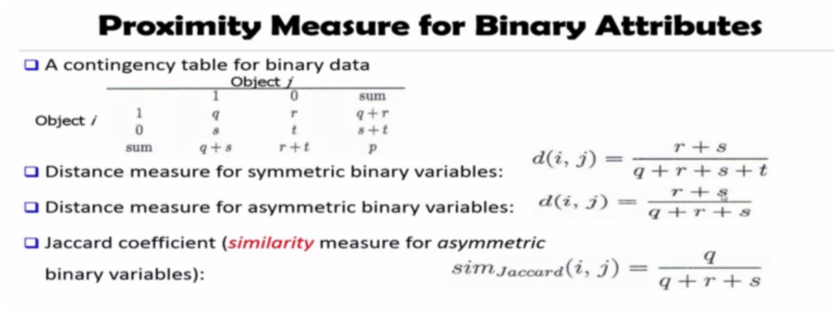
Collaborative filtering works on the premise that the customers with similar purchase behaviour tend to buy items which are not common when recommended to either of them



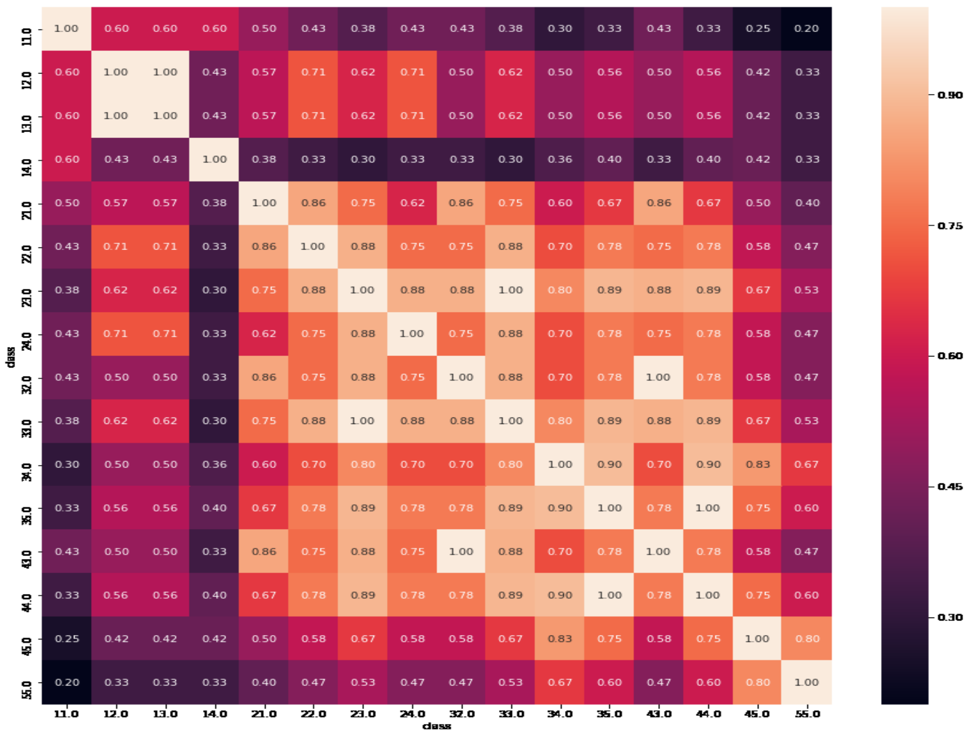
Since we have less information about the product characteristics , scoring the customers based on a similarity metric would be more meaningful . Since we have no data to judge the liking of the customer towards a product category ,such as a rating, we have no choice but to create a binary matrix of Customers against Product category - a '0' indicates that the customer didn't from the Product category a '1' indicates that the Customer bought from the particular category

class Product_Category_1	11.0	12.0	13.0	14.0	21.0	22.0	23.0	24.0	32.0	33.0	34.0	35.0	43.0	44.0	45.0	55.0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
3	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1
4	0	0	0	0	1	1	1	0	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	0	1	1	0	0	1	1	1	0	1	1	1	0	1	1	1
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1

The metric chosen was **Jaccard similarity** and since we are dealing with binary data, which is asymmetric meaning a 1 is more important than a 0 we calculate the Jaccard similarity score after excluding the data where both customers haven't bought from either of the categories (t) as shown below



Taking this into account, the data was converted to Boolean type and the Jaccard score for the 16 class of customers was calculated as shown below:



Based on this similarity score we can recommend a product category to potential class of customers who are likely to buy. For example, Customers with FM score of 23 purchase 89%

similar to customers having FM score of 35 and 44, so the probability of the former buying from the category which the latter Customers bought is high.

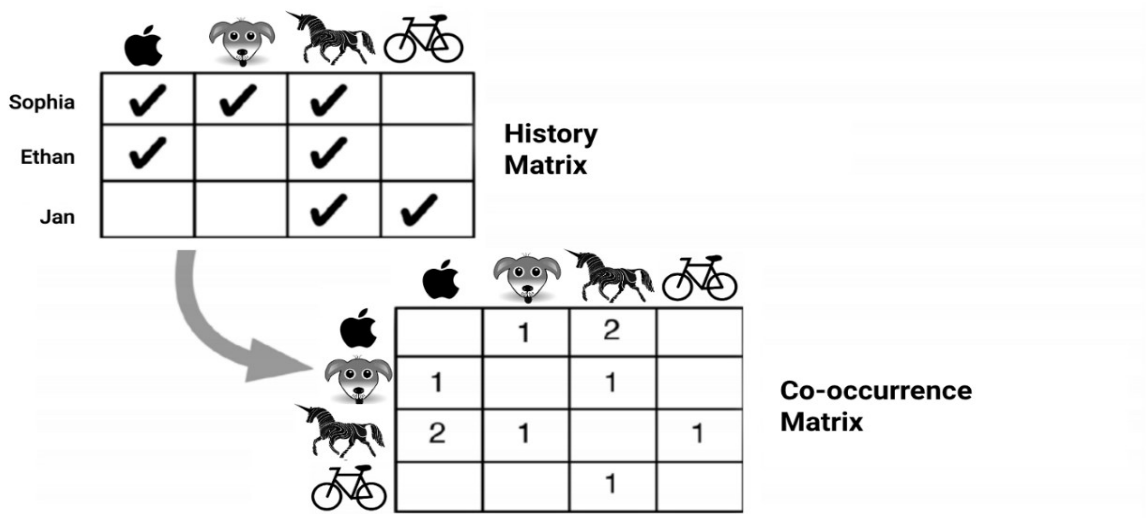
Recommendations:

If we set the threshold of similarity as **70%,** then we can recommend the remaining product categories to the corresponding Customer class as shown below :

Customer_class1	Customer_class2	Jacc_Similarity	Recommended
55.0	45.0	0.800000	[7 12 18]
45.0	44.0	0.750000	[10 13 15]
45.0	35.0	0.750000	[10 13 15]
45.0	34.0	0.833333	[13 15]
44.0	43.0	0.777778	[6 16]
44.0	34.0	0.900000	[10]
44.0	33.0	0.888889	[16]
44.0	32.0	0.777778	[6 16]
44.0	24.0	0.777778	[4 16]
44.0	23.0	0.888889	[16]
44.0	22.0	0.777778	[11 16]
43.0	24.0	0.750000	[4 6]

PRODUCT RECOMMENDATION

Now that we have the intuition behind recommending product category, we can go ahead and get the top 100 products using simple matrix multiplication and recommend them to the category of customers of our interest which are '**Superstars'** in this case



To be able to do that, we need to construct a couple of matrices

- **History matrix**

  This matrix tells us the customers purchase behaviour towards certain products. The idea is to use this history to recommend new products based another matrix called 'Co-occurrence matrix '

The history matrix for the '**Superstar'** customer of this dataset is shown below :

Product_ID User_ID	P00000142	P00000242	P00000342	P00000442	P00000542	P00000642	P00000742	P00000842	P00000942	P00001042	...	P0098942	P0099042	P0
1000195	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	...	0.0	0.0	
1000216	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	...	0.0	0.0	
1000245	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	
1000302	1.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	
1000319	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	...	0.0	0.0	

5 rows × 3623 columns

As it can be seen, the matrix is sparse with a lot of 0's which is rather obvious because each customer will only buy certain products

- **Co- occurrence matrix**

  The cooccurrence matrix quantifies the number of occurrences between each Product pairs, the higher the number the more popular the product pair.

So how do we construct such a matrix. This is where Linear Algebra comes into picture and a simple matrix multiplication between the transpose of the history matrix with size 3623 x 139 with the original matrix of size 139 x 3623, the resulting matrix will have a size of 3623 x 3623 and the values give the number of occurrences of each pair

Product_ID Product_ID	P00000142	P00000242	P00000342	P00000442	P00000542	P00000642	P00000742	P00000842	P00000942	P00001042	...	P0098942	P0099042	P0
P00000142	0.0	152.0	87.0	43.0	57.0	174.0	87.0	15.0	15.0	170.0	...	9.0	48.0	
P00000242	152.0	0.0	44.0	14.0	28.0	70.0	38.0	13.0	14.0	110.0	...	2.0	22.0	
P00000342	87.0	44.0	0.0	19.0	31.0	33.0	49.0	5.0	8.0	58.0	...	3.0	17.0	
P00000442	43.0	14.0	19.0	0.0	20.0	24.0	25.0	1.0	4.0	20.0	...	7.0	8.0	
P00000542	57.0	28.0	31.0	20.0	0.0	22.0	28.0	3.0	13.0	35.0	...	5.0	11.0	

5 rows × 3623 columns

The above matrix shows that the product **P00000242** and **P00000142** occur 152 times in the transaction

Now that we have the co-occurrence matrix we can recommend the products by multiplying the history matrix with the co-occurrence matrix to get the weights for each product

But before that we need to take into account the fact that the products already bought by the customer, we don't want to recommend the same product once again so we just subtract the history matrix from 1 to take into account only the products which are not bought by the customer

We can now multiply this new history matrix with the co-occurrence matrix to get the weights for the products that have not been bought by the customer

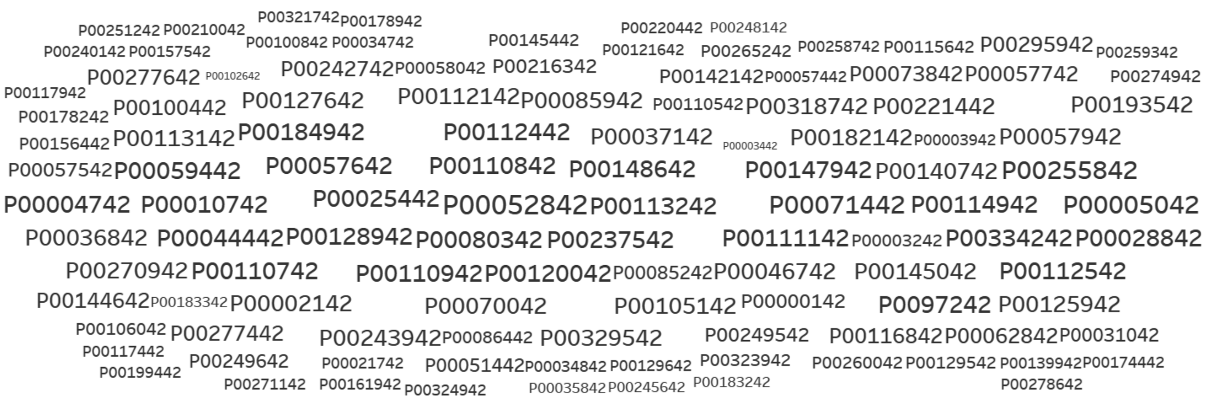
Product_ID	P00000142	P00000242	P00000342	P00000442	P00000542	P00000642	P00000742	P00000842	P00000942	P00001042	...	P0098942	P0099042	P0
User_ID														
1000195	48402.0	0.0	13417.0	6303.0	9647.0	27950.0	13675.0	0.0	4218.0	28483.0	...	2400.0	8589.0	
1000216	44441.0	19382.0	0.0	0.0	9138.0	24332.0	13720.0	2102.0	3529.0	0.0	...	2328.0	8637.0	
1000245	48616.0	23028.0	13947.0	6080.0	9759.0	29118.0	13694.0	2373.0	4446.0	30508.0	...	2330.0	8625.0	
1000302	0.0	0.0	15802.0	7205.0	11690.0	31824.0	16091.0	2647.0	5185.0	32998.0	...	2596.0	9433.0	
1000319	45397.0	18910.0	12597.0	6319.0	9049.0	0.0	14195.0	2080.0	3639.0	24580.0	...	2566.0	8820.0	

5 rows × 3623 columns

The resulting matrix will look something like this where a 0 value indicates the products already bought by the customer

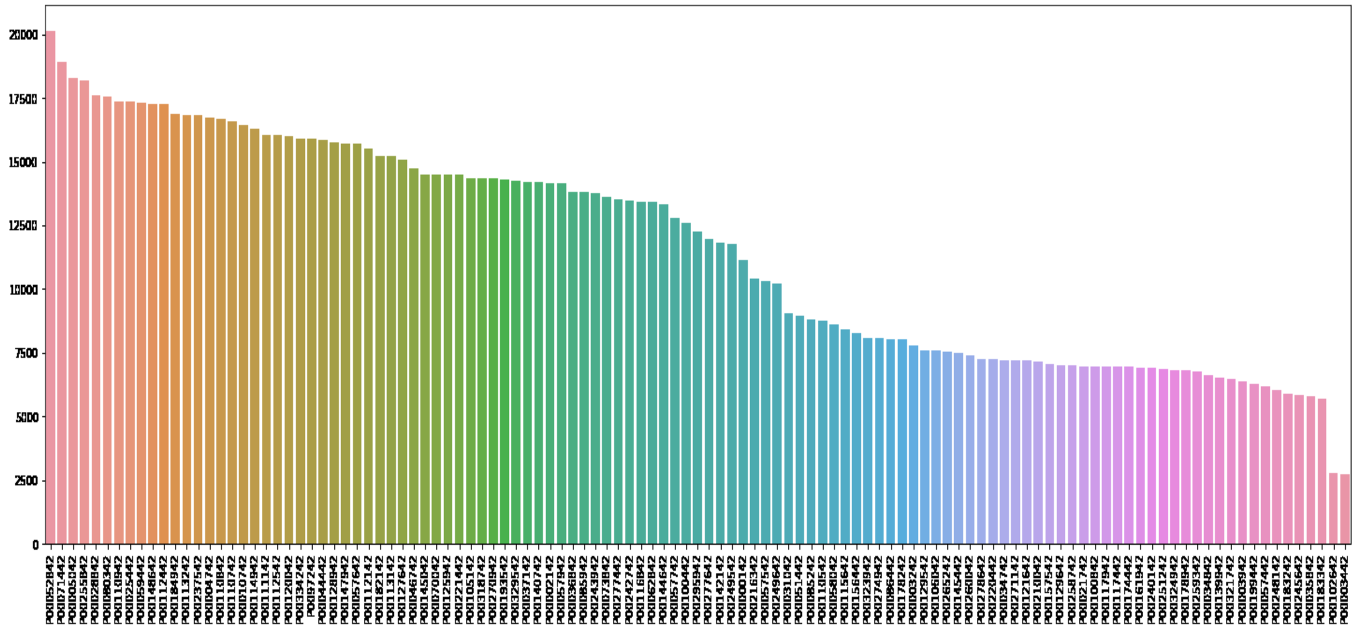
Using this matrix we can recommend products with top k values to each customer.To get the top 100 products we just take the value k = 40 i.e, the top 40 products of each customer . This value gave us 100 unique products as shown below:

Word chart of products with average price as size

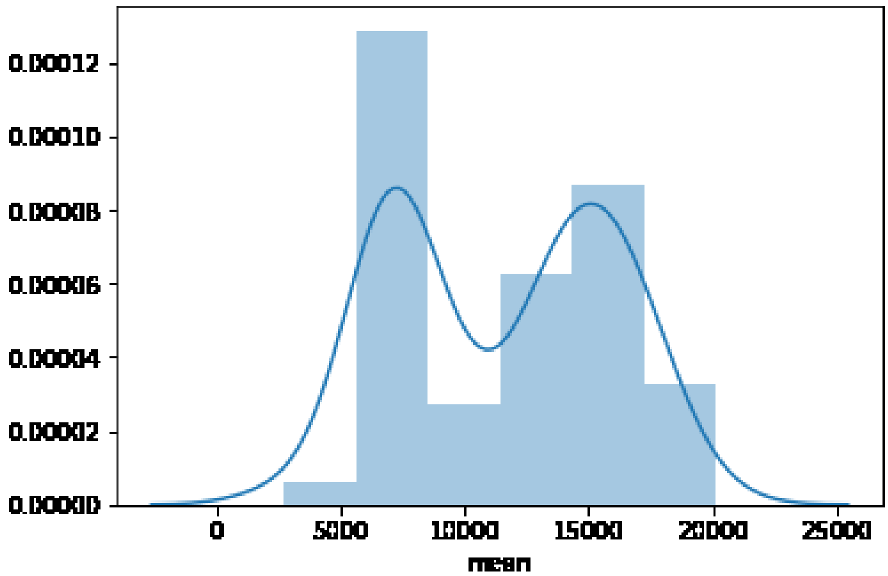


The size of the products indicating price doesn't vary much

Average price of top 100 products



Distribution of the average price



The average price of these 100 products is **11,523.86**
If we assume an 80 % conversion rate for each customer i.e, 80 % of 40 = 32, we get a customer lifetime value (CLV) of (32 x 11523.86)
$ **368,763.52**

Therefore the average revenue from our ‘Superstars’ is(139 x 368763.52)
$ **51,258,129.28**

BUSINESS RECOMMENDATIONS:

Timely Promotions
For customers, loyalty takes effort. They receive lots of promotional ads everyday to try products from other brands. Appreciating your consumer's urge to resist the hype is important. They expect timely promotions that not only fit their buying habits but also their lifestyles

Personalized Rewards
Everyone likes to be rewarded. It signifies that you've done something commendable. And incentives compel you to continue the rewarded behaviour. Recognize the value of your customer's actions. Because that's what you're rewarding.

Notice churning signs in advance
The most obvious way to ensure customer retention is to prevent a customer from leaving. If you really pay attention, you can always detect the signals of your customer's impending departure. we need to identify the key variables of customer behaviour, such as purchase patterns, product usage and history of customer service enquiries

Customer Service:
American Express found 33% of customers will consider switching companies after just one instance of poor customer service. So effective customer service should be a top priority

By simply automating the process of content marketing, email marketing, and social media marketing, you can see success in multiple ways:

- Staying in touch with your current customer base.

- Reminding their customers of upcoming events and seasonal tips.
- Maintaining a professional appearance all the way to the inbox and beyond.
- Taking advantage of opportunities to reach out to clients you have not talked to in years.
- Getting more referrals sent your way.

CONCLUSION

Customer loyalty programs are crucial. The goal of loyalty initiatives is to engage, not pander more products to frequent buyers. This report suggests the Black Friday sales data from the retail store should employ multiple marketing strategies. By instituting a loyalty program, you not only improve customer appreciation of your business, but you also increase the chances that existing clients will share this joy with those close to them. Consumer data must be analysed to create highly targeted product recommendation offers. Analyse consumer data such as demographics, lifestyle, products purchased by category and type, frequency of purchase, and purchase value

APPENDIX

JACCARD SIMILARITY between customer classes :

```
jac_sim = 1 - pairwise_distances(userclasses.T.astype(bool), metric = "jaccard")
jac_sim = pd.DataFrame(jac_sim, index=userclasses.columns, columns=userclasses.columns)
jac_sim
```

Setting the threshold :

```
cc = []
for i in jac_sim.columns.values:
  for j in jac_sim.columns.values:
    cc.append(jac_sim.loc[i,j])
comb['Jacc_Similarity'] = cc
comb7 = comb[comb['Jacc_Similarity']>0.7]
comb7 = comb7[comb7['Jacc_Similarity']<1]
comb7 = comb7.sort_values(0,ascending=False)
comb7 = comb7[comb7[0]>comb7[1]]
```

Recommendations:

```
a = []
for i,j in zip(comb7[0],comb7[1]):
  a.append(str(np.where(userclasses[i]!= userclasses[j])[0]+1))
comb7['Recommended'] = a
```

Calculations of FM Score:

For Frequency Percentile

- RANK_PERCENTILE (SUM([Number of Records]),'asc')

For Frequency Ranking

```
if [Frequency_Percentile]>=0.85 THEN 5
ELSEIF [Frequency_Percentile]<0.85 AND  [Frequency_Percentile]>=0.70 THEN 4
ELSEIF [Frequency_Percentile]<0.70 AND  [Frequency_Percentile]>=0.55 THEN 3
ELSEIF [Frequency_Percentile]<0.55 AND  [Frequency_Percentile]>=0.40 THEN 2
ELSE 1
```

END

For Purchase Percentile

RANK_PERCENTILE (sum([Purchase]),'asc')

For Purchase Ranking

if [Purchase_Percentile]<=0.25 THEN 1

ELSEIF [Purchase_Percentile]>0.25 and [Purchase_Percentile]<=0.4 THEN 2

ELSEIF [Purchase_Percentile]>0.40 and [Purchase_Percentile]<=0.55 THEN 3

ELSEIF [Purchase_Percentile]>0.55 and [Purchase_Percentile]<=0.7 THEN 4

ELSE 5

END

Recommendations :

recommvector = user_history_matrix.dot(cooccurrence_matrix)

recommvector = recommvector*(1- user_history_matrix)

Top 40 products of each user :

dic1 = {}

product_weights = {}

for i in recommvector.columns:

product_weights.update({i:recommvector[i].sort_values(ascending=False)[:40]})

top100 = pd.DataFrame(product_weights)

top100.index   # TOP 100 products