ANALYSIS OF GROCERY ORDERS DATA

MS BANA Capstone Project

Submitted by:

S.V.G. SRIHARSHA,

M10574454

First Reader:

Dr. Yichen Qin

Second Reader:

Dr. Jeffery Mills

Table of Contents

1) **Abstract……2**

2) **Introduction**

2.1) Structure of Data………………………………………………………………………………………………2

2.2) Executive Summary……………………………………………………………………………………….… 5

3) **Exploratory Analysis…………………………………………………………………………………………….……6**

4) **Methodology**

3.1) Association Rule Mining………………………………………………………………………………….… 8

3.2) Classification of Customers using K-Means clustering..………………………………………….12

3.3) Product Embedding……………………………………………………………………………………….....16

5) **Results**

4.1) Association Rule Mining…………………………………………………………………………………...18

4.2) Customer Clustering…………………………………………………………………………………………18

4.3) Product Embedding…………………………………………………………………………………………22

6) **References………………………………………………………………………………………………………..…...24**

1) Abstract:

Objective of this analysis is to study order pattern of users of Instacart, a grocery delivering company and provide key insights about the customer behavior. There are 206209 users in the database and 49687 different products available to order through instacart which can be characterized to 21 different departments. Current database consists of the details about 3421083 orders placed by the users over a certain amount of time.

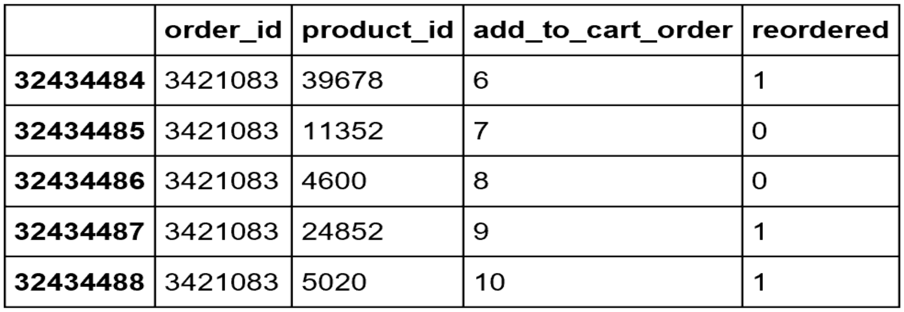
This analysis starts with exploration of variables then moves on to i) Association rule mining using apriori algorithm, ii) Unsupervised classification of customers based on their buying behavior using K-means clustering algorithm, iii) Product embedding using Word2Vec analysis and concludes with summary of results.

2) Introduction:

2.1 Structure of Data:

Data is composed of four tables, 1) Orders, 2) Order-Details, 3) Products and 4) Departments

1) Orders:

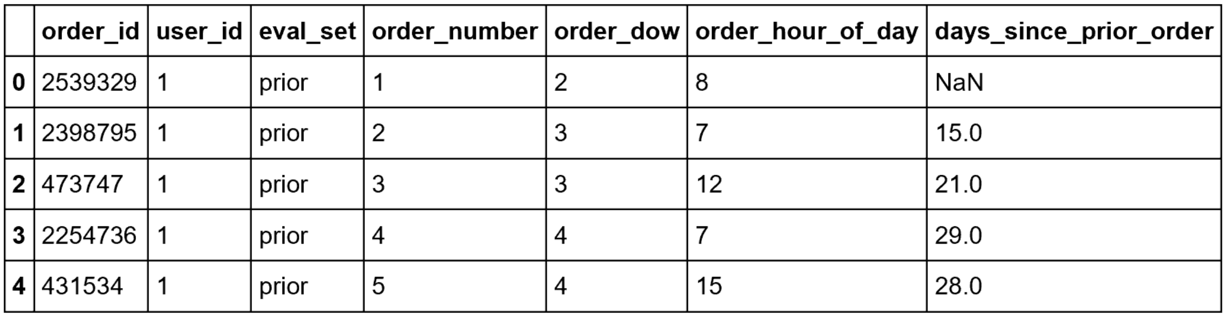


	order_id	product_id	add_to_cart_order	reordered
32434484	3421083	39678	6	1
32434485	3421083	11352	7	0
32434486	3421083	4600	8	0
32434487	3421083	24852	9	1
32434488	3421083	5020	10	1

Fig 1.1: Orders table

"Orders" table consists of all the products in a particular order with one product per row. The column "add_to_cart_order", represents the quantity(units) of the product purchased under that order. The column "reordered", represents whether the product has been reordered by the user or not.

2) Order-Details:

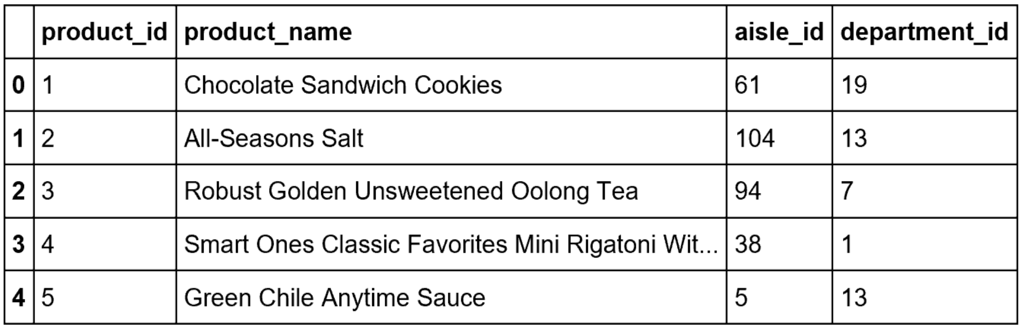


	order_id	user_id	eval_set	order_number	order_dow	order_hour_of_day	days_since_prior_order
0	2539329	1	prior	1	2	8	NaN
1	2398795	1	prior	2	3	7	15.0
2	473747	1	prior	3	3	12	21.0
3	2254736	1	prior	4	4	7	29.0
4	431534	1	prior	5	4	15	28.0

Fig 1.2: OrderDetails table

"OrderDetails" table gives other details of the order. Each order is noted as a unique observation. The column "user_id" represents a unique id per customer. "eval_set" is an indicator whether the order a recent one or not. "order_number" column represents the sequence of the order by user. "order_dow" and "order_hour_of_day" represent day of the week and time of the day of the order respectively. "days_since_prior_order", represents no.of days passed after the last of that particular user.

3) Products:

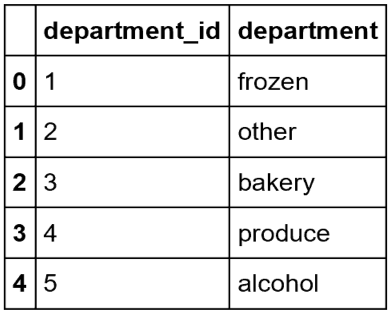


	product_id	product_name	aisle_id	department_id
0	1	Chocolate Sandwich Cookies	61	19
1	2	All-Seasons Salt	104	13
2	3	Robust Golden Unsweetened Oolong Tea	94	7
3	4	Smart Ones Classic Favorites Mini Rigatoni Wit...	38	1
4	5	Green Chile Anytime Sauce	5	13

Fig 1.3: Products table

Products" table consists of product_id, product_name, aisle_id and department_id, which are self-explanatory.

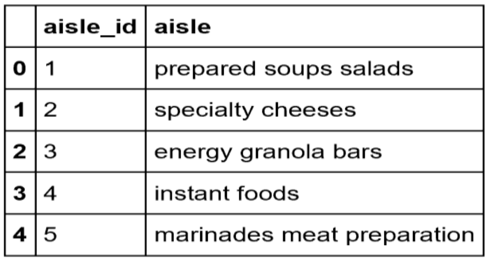
4) Departments:



	department_id	department
0	1	frozen
1	2	other
2	3	bakery
3	4	produce
4	5	alcohol

Fig 1.4: Departments table

5) Aisles:



	aisle_id	aisle
0	1	prepared soups salads
1	2	specialty cheeses
2	3	energy granola bars
3	4	instant foods
4	5	marinades meat preparation

Fig 1.5: Aisles table

Departments and Aisles tables map department_id and aisle_id to their respective names.

Comments:

- The variable “number of days” since prior is capped at 30
- There are 21 departments which are further categorized into 134 aisles comprising of 49687 products in total
- Each user has ordered 4 to 100 orders. No. of orders by a user is capped at 100

2.2) Executive Summary:

1) Association rule mining:

Transaction database of instacart orders is mined for relevant association rules using apriori algorithm. Based on higher lift ratio and confidence limit of 0.75 (minimum), some of the major association rules obtained are as follows: 1) "*Moisturizing Non-Drying Facial Wash" is bought 100 % of the times when " Moisturizing Facial Wash" is bought*. 2) "*Premium Classic Chicken Recipe Cat Food" is bought 75% of the times when "Ocean Whitefish" is bought, 3) "Thousand Island Salad Snax" is bought 80% of the times when "Raspberry Vinaigrette Salad Snax" is bought and 4) "2nd Foods Turkey Meat" is bought 78% of the times when "2nd Foods Chicken & Gravy is bought"*

2) Classification of Customers:

Customers are classified into clusters based on their buying behavior (i.e., number of orders placed during specific times of the day, specific days of the week, number of days between subsequent orders, number of orders placed) using K-means clustering algorithm. 5 clusters of customers were obtained and the variation of customer behavior among different clusters is discussed in the results section in detail.

3) Product embedding:

Word2Vec analysis using skip-grams network is used to embed products onto a 280-dimensional vector space based on their relative occurrences in grocery orders. This vector codes specific unknown characteristics of products that can be learned from the available data which further can be used in building recommender systems or grouping similar products.

3) Exploratory Analysis of the data:

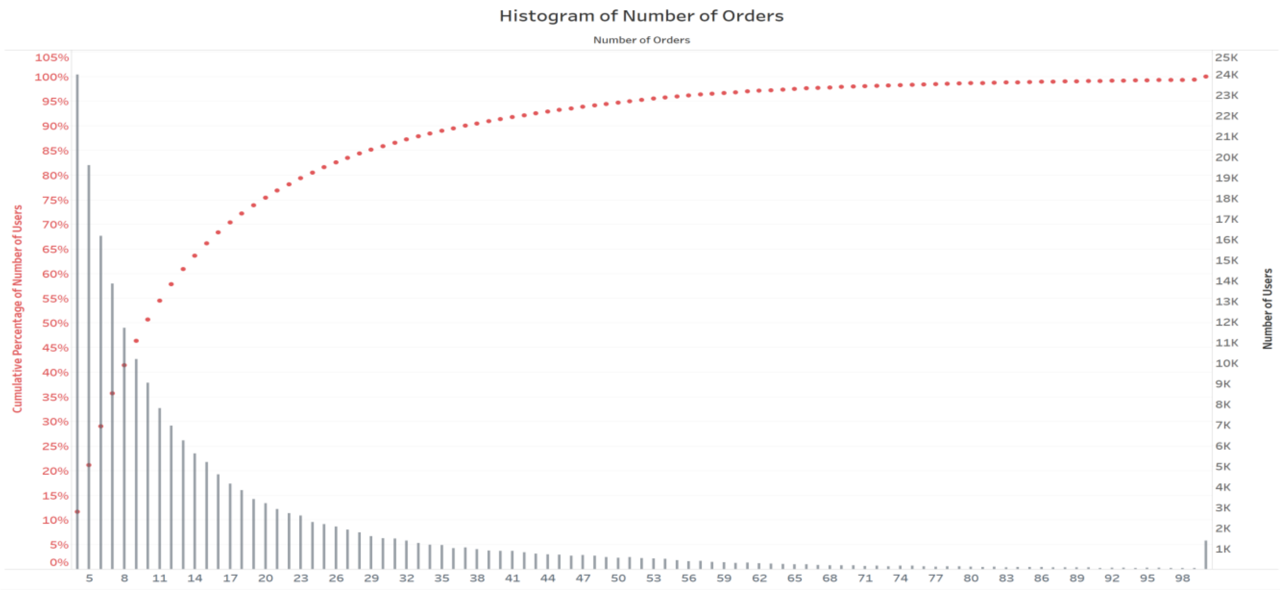


Fig 2.1: Histogram of number of orders by a user

Majority of the users (> 90%) have ordered less than 40 orders each. 50% of the customers ordered less than 12 orders. There is a sudden spike at 100 in the above histogram as the number of orders per customer is capped at 100.

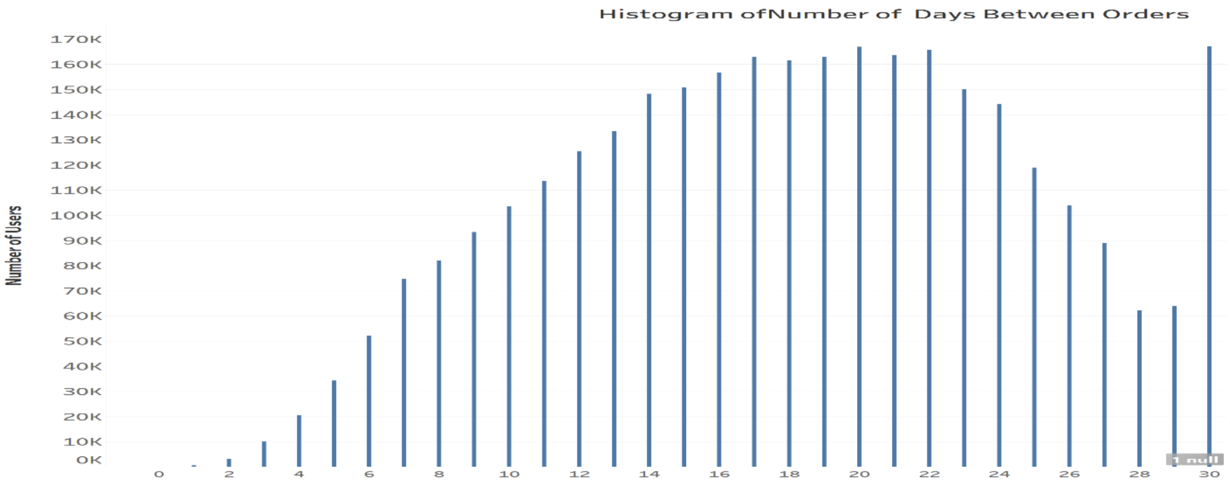


Fig 2.2: Number of Days Between Orders

Average number of days between subsequent orders follows a near normal distribution with average number of days between orders at around 18 days. There is an abrupt spike at 30 in the above histogram because the variable "Number of days since prior order" is capped at 30. There are around 400 users with average number of days between subsequent orders less than 2 days.

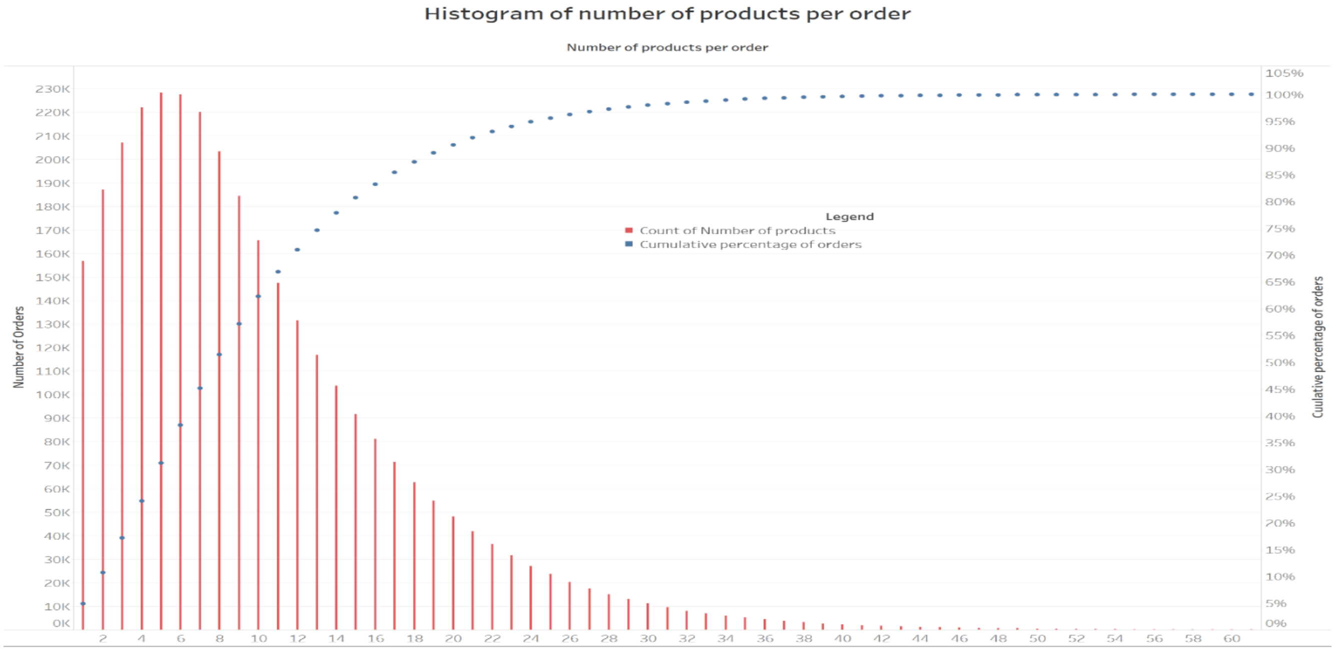


Fig 2.3: Number of products per order

Average number of products (distinct) bought per order follows a skewed near normal distribution as above. 50% of the orders contain less than 10 products each. Though there are outliers (some orders with >50 products), 95% of the orders contain less than 24 products.

4) Methodology:

4.1) Association Rule Mining:

Association rule mining is a data mining technique used to mine relevant rules about co-occurrence of items in a transaction database using apriori algorithm. Apriori algorithm proceeds by building possible item-sets from the transaction database and association rules are obtained by quantifying frequent item-sets using different characteristics such as support, confidence and lift-ratio.

Support of an item-set: Relative frequency of the item-set in the transaction database. Confidence of an association rule X => Y: support(X U Y)/ Support(X), probability of Y being in the transaction given X. Lift – ratio of X => Y: Support(X U Y)/ (Support(X) * Support(Y)), indicates the level of dependency between the items X and Y. If lift- ratio > 1, X and Y are independent. Higher the confidence and lift-ratio of the rule better the efficiency. Generally in large databases involving large number of transactions and products, support of the rules will be lower.

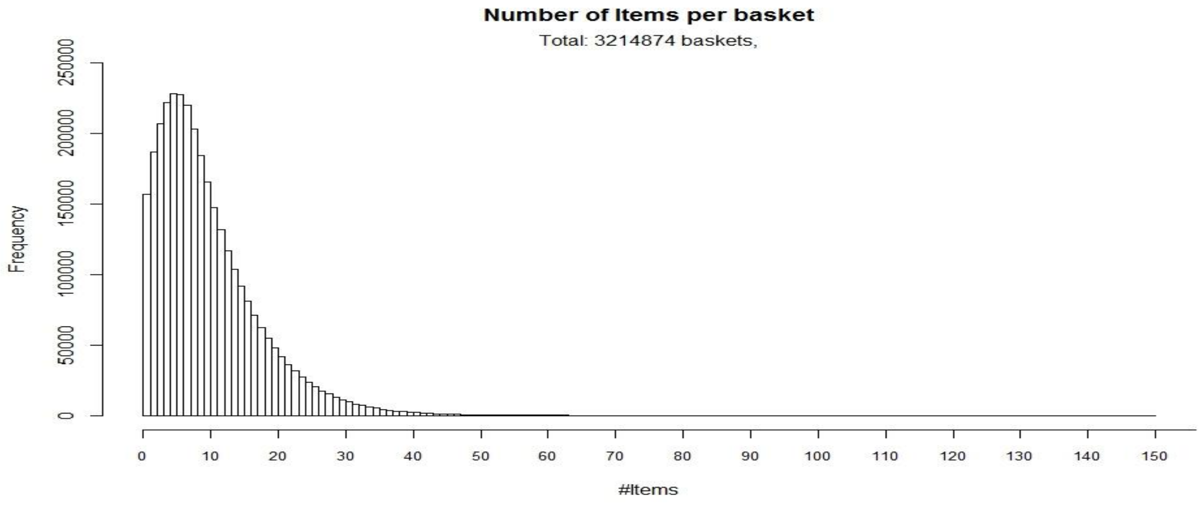


Fig 4.1.1: Histogram of item-sets

There are 3214874 possible combination of item-sets. As it can be noticed from the above plot, majority of the item-sets are of length around 10 to 20 items. There are around 230000 possible combinations of 10 item-sets.

Frequent items:

Here is a view of the frequent items with minimum support of 0.025, i.e., occurring in atleast 80000 transactions.

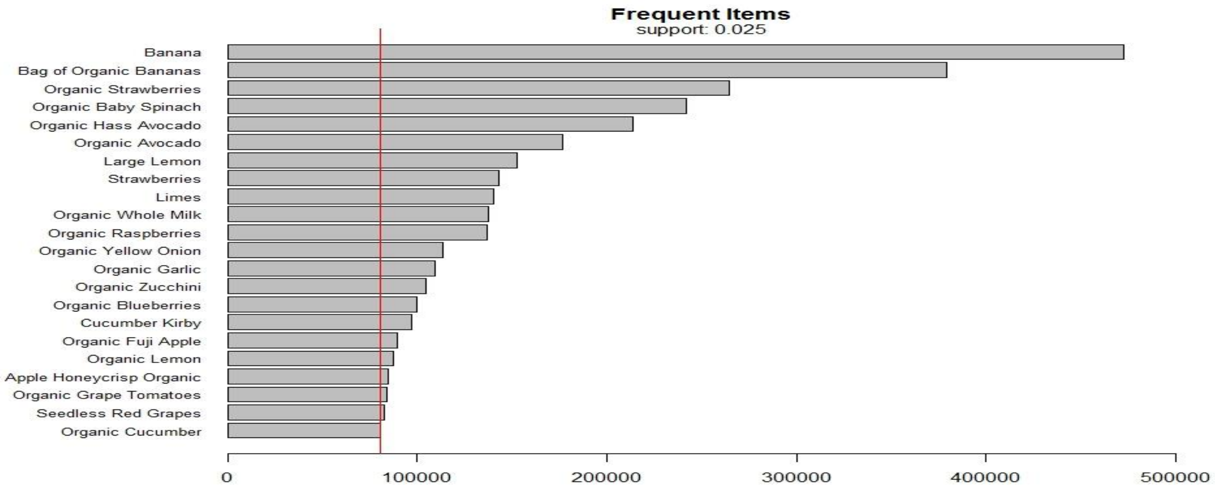


Fig 4.1.2: Frequent Items

Majority of the frequent items are fruits, vegetables and milk. Interestingly most of them are organic.

Frequent Item-sets:

Frequent item-sets with at least two items and a minimum support of 0.008(i.e., occurring in at least 27800 out of 3421083 orders) are as follows:

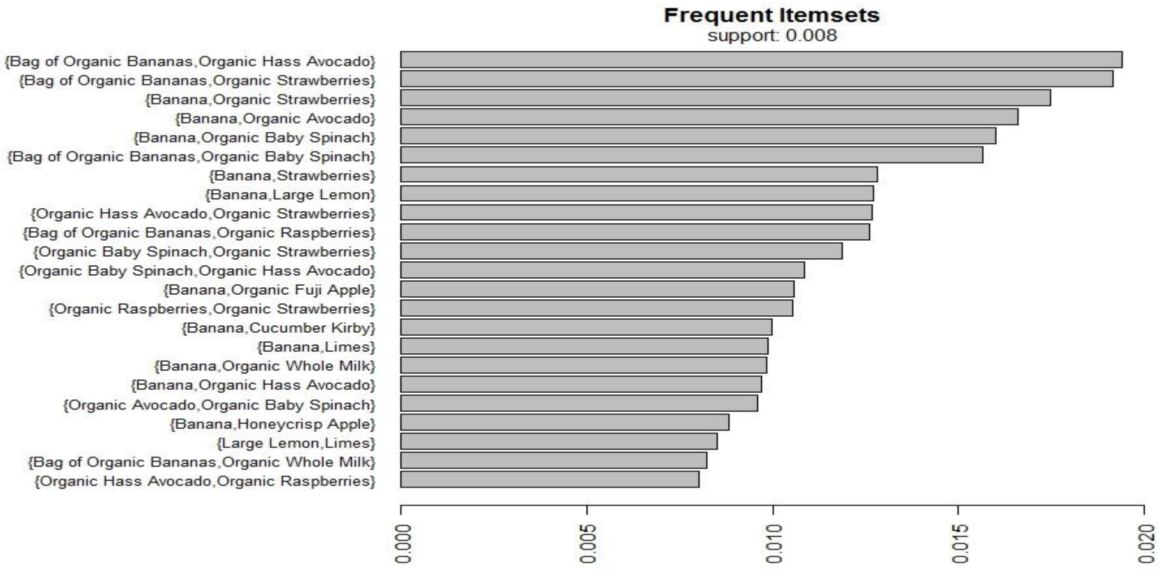


Fig 4.1.3: Frequent Item-sets

All the item-sets are two item-sets. Majority of them involve frequent items from above plot. Each of the eight pairs with highest support contains bananas. Nearly most of the items are either fruits or vegetables. There is just one frequent pair that contains milk.

Association rules:

Summary of association rules with minimum support of 0.00001 and minimum confidence of 0.6 using apriori algorithm:

```
    support             confidence          lift     
 Min.   :0.00001026   Min.   :0.6000   Min.   :    4.08  
 1st Qu.:0.00001089   1st Qu.:0.6552   1st Qu.:   71.40  
 Median :0.00001244   Median :0.7375   Median :  306.47  
 Mean   :0.00001533   Mean   :0.7624   Mean   : 1209.16  
 3rd Qu.:0.00001555   3rd Qu.:0.8605   3rd Qu.: 1196.74  
 Max.   :0.00085042   Max.   :1.0000   Max.   :76544.62  
```

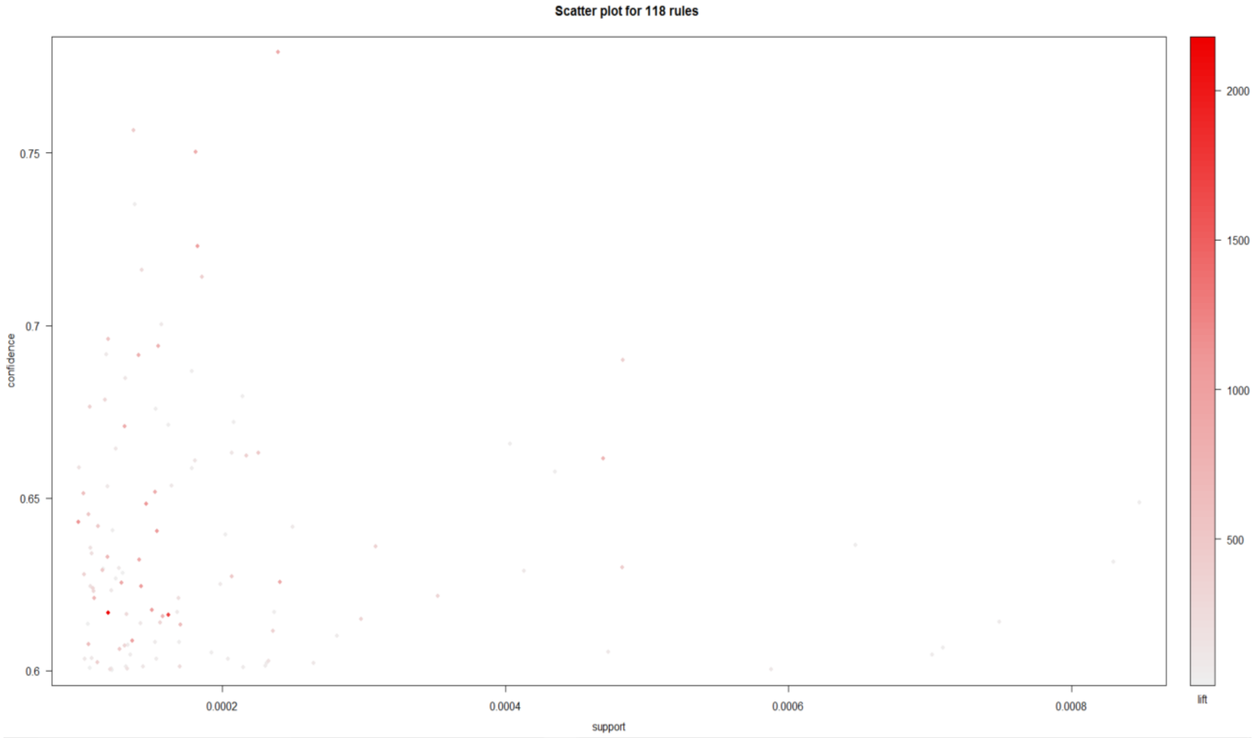


Fig 4.1.4: Scatter Plot of Association rules

Lift ratios of the rules are high. There are few rules with confidence level > 0.7 and very high lift indicating strength of association between the items involved.

Rules with maximum lift ratio:

```
[1]   {Moisturizing Facial Wash}=>
      {Moisturizing Non-Drying Facial Wash}
      support: 0.000013  confidence: 1  lift:76544.62

[2]   {Prepared Meals Simmered Beef Entree Dog Food}=>
      {Prepared Meals Beef & Chicken Medley Dog Food}
      Support: 0.000012  confidence: 0.6 lift: 32211.59

[3]   {Ocean Whitefish}=>
      {Premium Classic Chicken Recipe Cat Food}
      support: 0.000010  confidence: 0.75 lift: 32148.74

[4]   {Ancient Grains Apricot Blended Low-Fat Greek Yogurt}=>
```

```
     {Oats Ancient Grain Blend with Mixed Berry Low-Fat Greek Yogurt}
     Support: 0.000014 confidence: 0.7419355  lift:29088.16

[5]  {Thousand Island Salad Snax}=>
     {Raspberry Vinaigrette Salad Snax}
     support: 0.000021  confidence: 0.6160714 lift: 23030.14

[6]  {Organic Baby Food Fruit Mashup Strawberry Patch 9+ Mo}=>
     {Organic Baby Food Fruit Mashup Mama Bear Blueberry 7+ Mo}
     support:0.0000118 confidence: 0.6229508 lift:21081.14
```

List of rules with maximum confidence:

```
[1]  {Moisturizing Facial Wash}=>
      {Moisturizing Non-Drying Facial Wash}
      Support: 0.000013  confidence: 1 lift:76544.619

[2]  {Raspberry Vinaigrette Salad Snax}=>
     {Thousand Island Salad Snax}
     Support: 0.00002  confidence: 0.8023256 lift:23030.140

[4]  {Extra Virgin Olive Oil Spray}=>
     {All-Purpose Unbleached Flour}
     Support: 0.00001244217  confidence: 0.7843137  lift: 8055.814

[5]  {2nd Foods Turkey Meat}=>
     {2nd Foods Chicken & Gravy}
     Support: 0.00006283294  confidence: 0.7769231  lift: 4887.886

[6]  {Apple Strawberry Banana Squeezable Fruit}=>
     {Graduates Grabbers Fruit & Yogurt Strawberry Banana}
     Support: 0.00001026479  confidence: 0.7674419  lift: 9908.550
```

3.2) Classification of customers using K-means clustering:

As we don’t have the demographic information about the customers, which can be quite useful in classifying the customer-base into various categories, customers are clustered based on their buying behavior i.e., average number of transactions per aisle per order, average number of orders at different time of the day and day of the week.

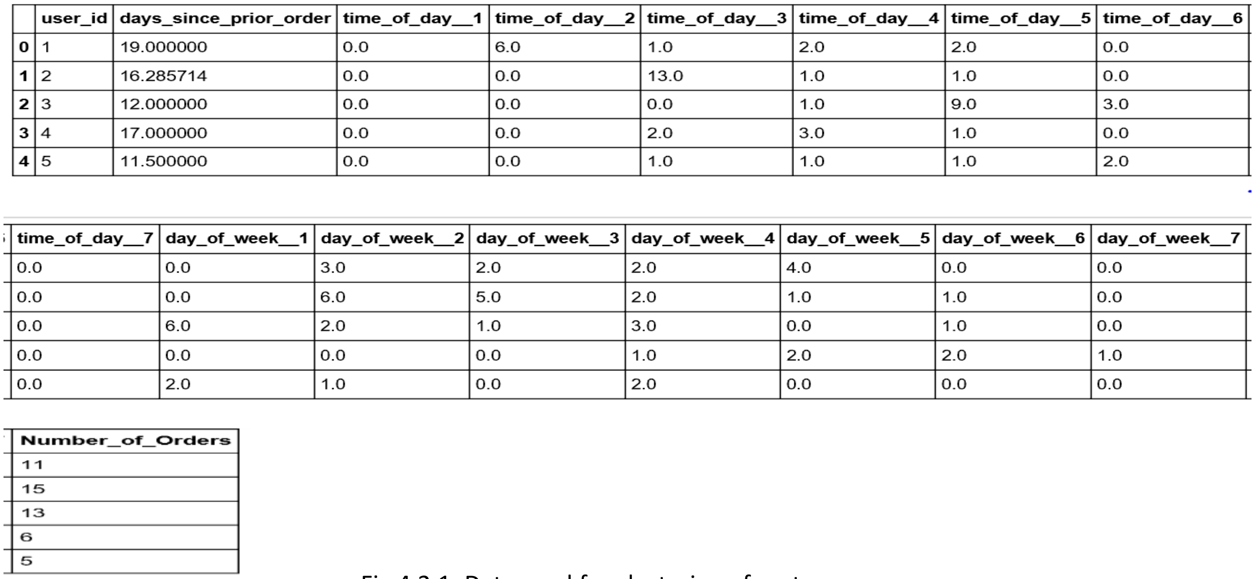
Data Preparation:

- There are no missing values except for the variable ‘number_of_days_since_prior_order’, which is not available for 1st order of each customer (missing values are replaced replaced with zero)
- Variable ‘time_of_the_day’ is bucketed to represent categories:
  1) Midnight to 5AM, 2) 5 to 9 AM, 3) 9 AM to Noon, 4) Noon to 3 PM, 5) 3 to 6 PM, 6) 6 to 9 PM and 7) 9 PM to Midnight

Input data:

- (i) average number of orders during different times of the day, (ii) average number of orders during each week of the day and (iii) average number of days between consecutive orders

Sample data:



	user_id	days_since_prior_order	time_of_day__1	time_of_day__2	time_of_day__3	time_of_day__4	time_of_day__5	time_of_day__6	time_of_day__7	day_of_week__1	day_of_week__2	day_of_week__3	day_of_week__4	day_of_week__5	day_of_week__6	day_of_week__7	Number_of_Orders
0	1	19.000000	0.0	6.0	1.0	2.0	2.0	0.0	0.0	0.0	3.0	2.0	2.0	4.0	0.0	0.0	11
1	2	16.285714	0.0	0.0	13.0	1.0	1.0	0.0	0.0	0.0	6.0	5.0	2.0	1.0	1.0	0.0	15
2	3	12.000000	0.0	0.0	0.0	1.0	9.0	3.0	0.0	6.0	2.0	1.0	3.0	0.0	1.0	0.0	13
3	4	17.000000	0.0	0.0	2.0	3.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	2.0	2.0	1.0	6
4	5	11.500000	0.0	0.0	1.0	1.0	1.0	2.0	0.0	2.0	1.0	0.0	2.0	0.0	0.0	0.0	5

Fig 4.2.1: Data used for clustering of customers

- There are 206209 observations of 15 variables each in the data used for clustering

Exploratory analysis of the variables:

- Time of the day:

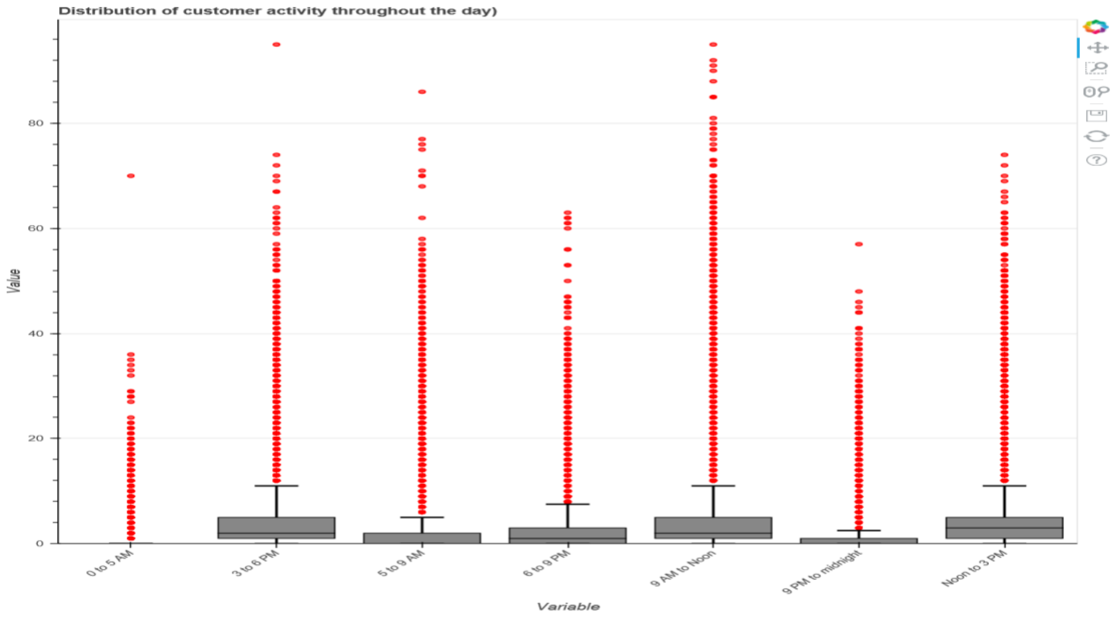


Fig 4.2.2: Distribution of the variables representing time of the day

The above plot shows the distribution of the variables representing the number of orders placed by a customer during various times of the day. As discussed earlier, customer activity is high during 9 AM to 6 PM. All of them have been normalized before clustering.

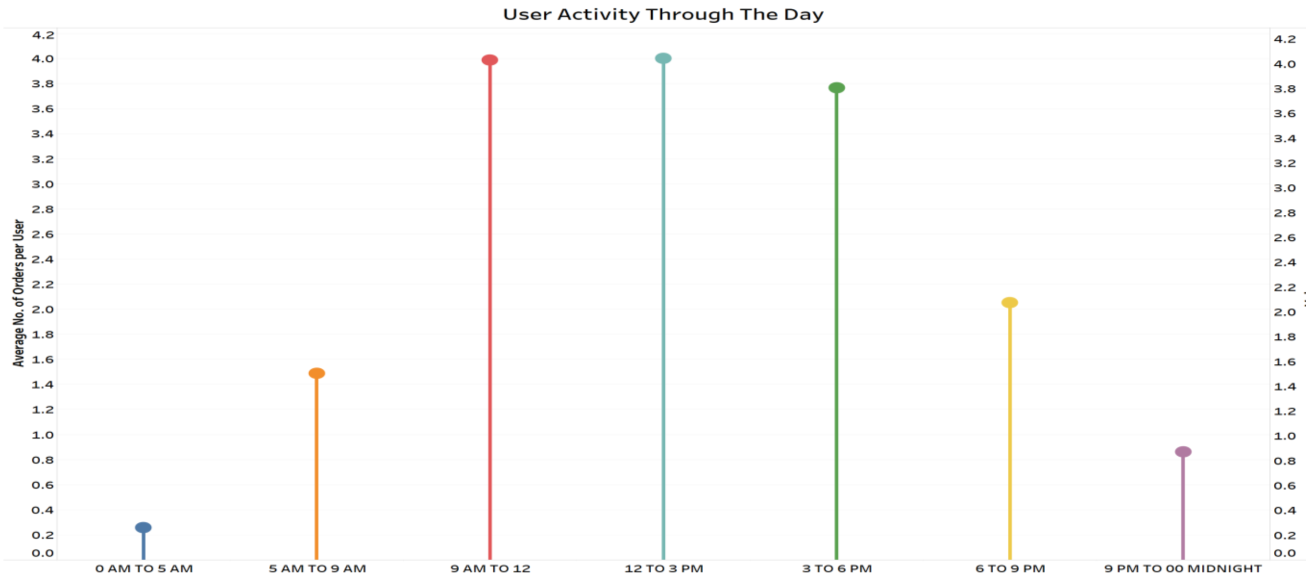


Fig 4.2.3: Average Customer activity throughout the day

As expected, customer activity is very low during odd timings of 9 PM to 5 AM. User activity is very high i.e., around 3.8 orders per customer on average during 9 AM to 6 PM.

Day of the week:

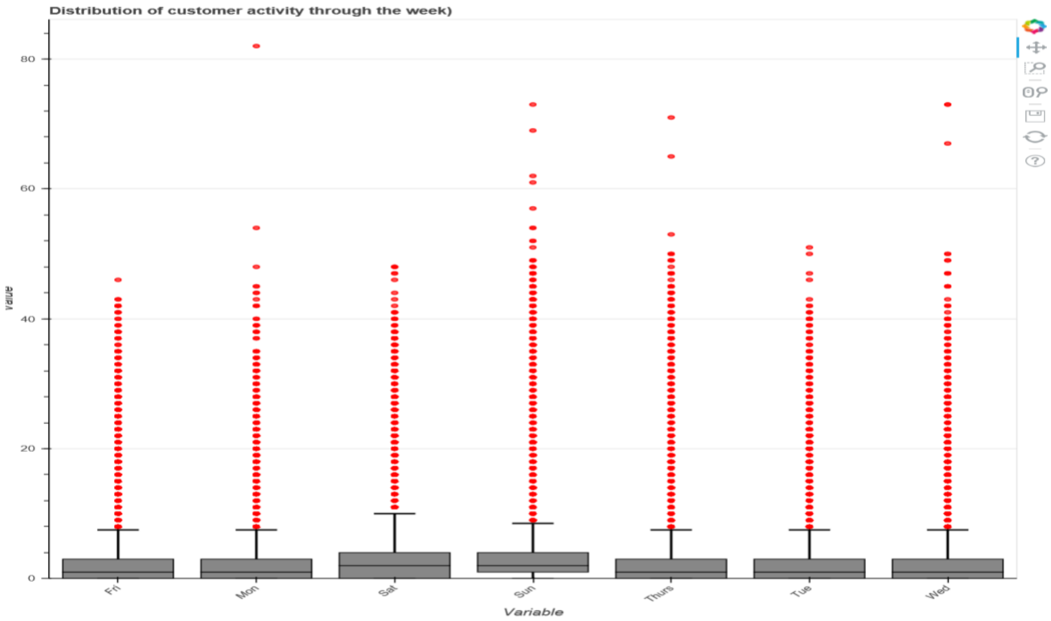


Fig 4.2.4: Distribution of variables representing activity during different days of the week

The above plot shows the distribution of number of orders placed by a customer during particular day of the week. There are a few outliers in all the variables. All of the variables are normalized before clustering.

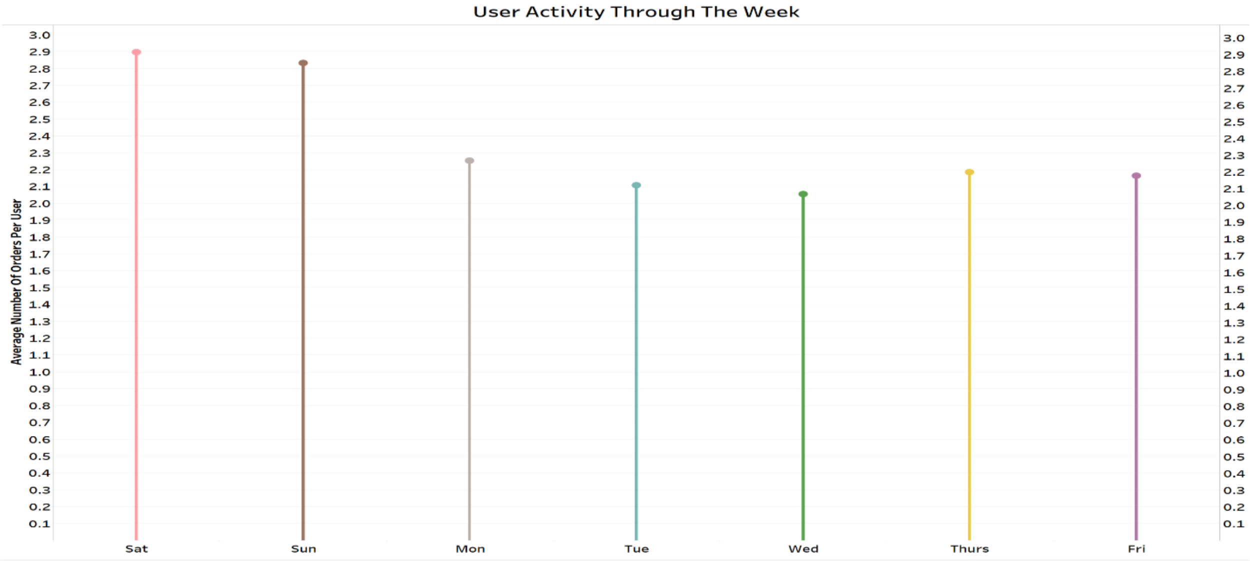


Fig 3.2.5: Average Customer Activity During Different Days of the Week

User activity is high during weekends and slowly drops during the weekdays. On average, user activity is around 2.5 to 3 orders per customer per day on any given day of the week.

Choosing Ideal Number of Clusters:

Ideal number of clusters is chosen based on effectiveness score. **Effectiveness Score** is the negative of sum of within-cluster sum of squared distances from centroid.

Effectiveness Score = - $\sum_{i=1}^{k}\sum_{\mathbf{x}\in S_i}\|\mathbf{x}-\boldsymbol{\mu}_i\|^2$, where k represents number of clusters, μ_i represents the centroid of cluster i and S_i represents the set of observations belonging to cluster i. Larger the score (maximum limit is zero), higher the effectiveness of the clustering. Below is the plot of effectiveness score vs number of clusters.

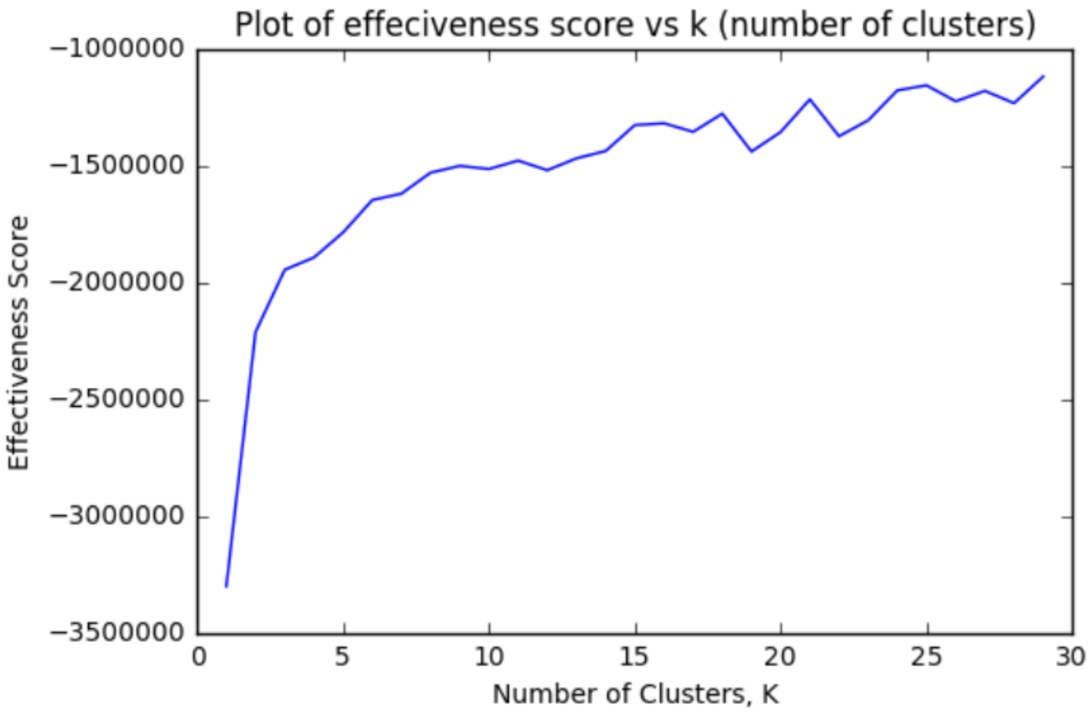


Fig 3.2.6: Elbow Plot

From the above plot using elbow method, ideal number of clusters in this case turned out to be 5.

4.3) Word2Vec analysis of products:

Word2Vec analysis is in general used in Natural Language Processing to categorize words based on their relative occurrence with each other by embedding each word into a n-dimensional vector, so that similar words are coded similar. Similarly, Word2Vec analysis (using skip-grams) is used to embed products based on their relative occurrences to embed similar products together.

Skip – Grams:

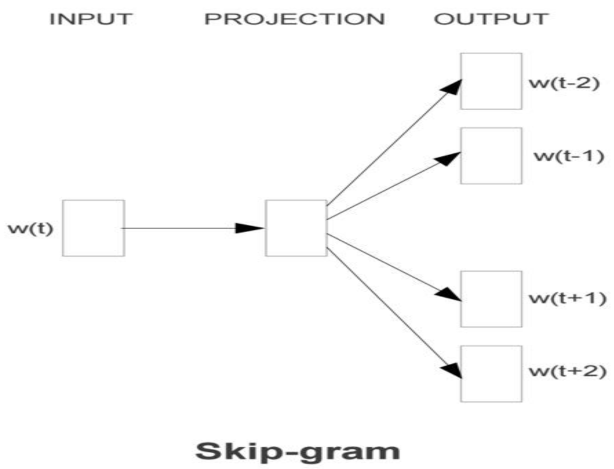


Fig 4.3.1: Skip-grams model

Skip-gram network architecture:

Suppose there are p number of unique products in our transaction database and if we would like to learn n-dimensional vector representing various characteristics of each product.

Input: One-hot encoded p-dimensional vector representing each product

Hidden_layer: p*n matrix of random weights, which when multiplied by the input vector produces p-dimensional vector of 1s and zeros, representing products co-occurring with input product in an order.

Each row of this hidden layer weight matrix consists of n-dimensional vector specific to each product, which is fitted based on the products that are bought along with this product. These vector representations of words can be used in other predictive models. Skip-grams network takes as input

Data Preparation:

Sample Text:

23734 15422 4932 46361 20114 13129 5031 2447 28931 <period> 13129 28931 20114 15422 5031 2447 23734 4932 46361 <period> 5031 23734 4932 13129 46361 15422 2447 20114 28931 <period> 18811 41665 32553 43195 21936 <period> 21936 41665 43195 18811 32553 <period> 41665 32553 18811 21936 43195 <period> 23099 18027 16987 13176 27086 11594 33055 38689 <period> 27086 33055 38689 18027 11594 23099 13176 16987 <period> 13176 38689 23099 16987 27086 11594 33055 18027 <period> 4421 13176 19048 47626 8555 4472 25753 33214 39877 48287 21903 35417 <period>

Each order is written down as a sentence containing product_id as its words separated by a ‘<period>’. For example, first text string ‘23734’ represents a specific product in our database. All such products till the text string ‘<period>’ comprise a single order. ‘<period>’ is coded to represent transition from one order to the other.

Train Data:

Input – One-hot encoded vector representing each product_id

Target – One-hot encoded vector representing (less than 10) random products from that order

Weight-Matrix: Skip-grams is used to learn 280 -dimensional vector to represent each product

5) Results:

5.1) Association Rules:

Major association rules (with minimum confidence of 0.7 and highest lift ratio) are as follows:

```
[1]  {Moisturizing Facial Wash}=>
     {Moisturizing Non-Drying Facial Wash}
     support: 0.000013  confidence: 1  lift:76544.62

[3]  {Ocean Whitefish}=>
     {Premium Classic Chicken Recipe Cat Food}
     support: 0.000010  confidence: 0.75 lift: 32148.74


[4]  {Raspberry Vinaigrette Salad Snax}=>
     {Thousand Island Salad Snax}
     Support: 0.00002  confidence: 0.8023256 lift:23030.140

[5]  {Extra Virgin Olive Oil Spray}=>
     {All-Purpose Unbleached Flour}
     Support: 0.00001244217  confidence: 0.7843137  lift: 8055.814

[6]  {2nd Foods Turkey Meat}=>
     {2nd Foods Chicken & Gravy}
     Support: 0.00006283294  confidence: 0.7769231  lift: 4887.886
```

Remarks:

As the transaction database is very huge, support (though very less) of the above rules is acceptable. Moisturizing Facial Wash => Moisturizing Non-Drying Facial Wash has a confidence of 1 which implies these two are bought together 100% of the time. With minimum confidence of 0.75 and lift-ratio way greater than 1, all of the above rules are very ueful.

4.2) Customer Clusters:

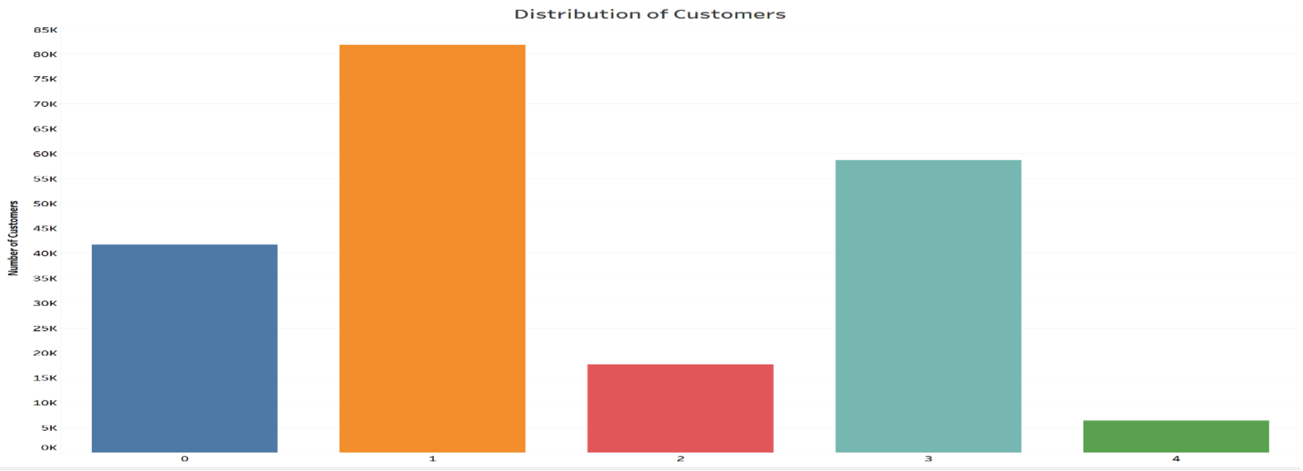


Fig 4.2.1: Number of Users per Cluster

The above plot represents number of customers in each cluster. Cluster1 consists of around 80k customers which makes up around 39% of total number of customers. Cluster4 is smaller in terms of customer population with around 6000 customers.

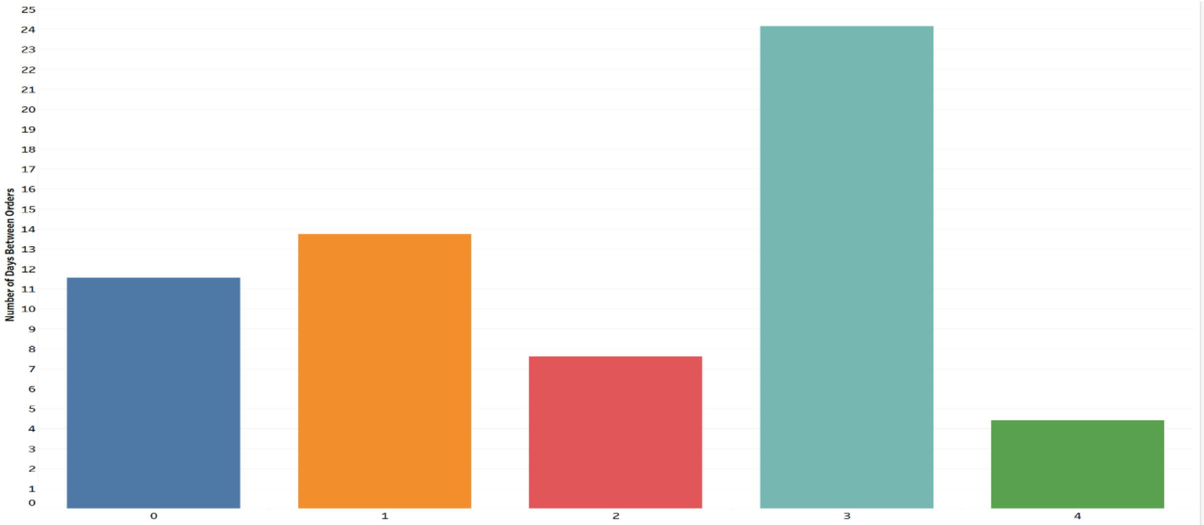


Fig 4.2.2: Distribution of Number of Days Between Orders

Customers from cluster4 are more frequent buyers relative to other clusters. Cluster3 represents customers with longer periods of dormancy between orders with about 24 days between subsequent orders.

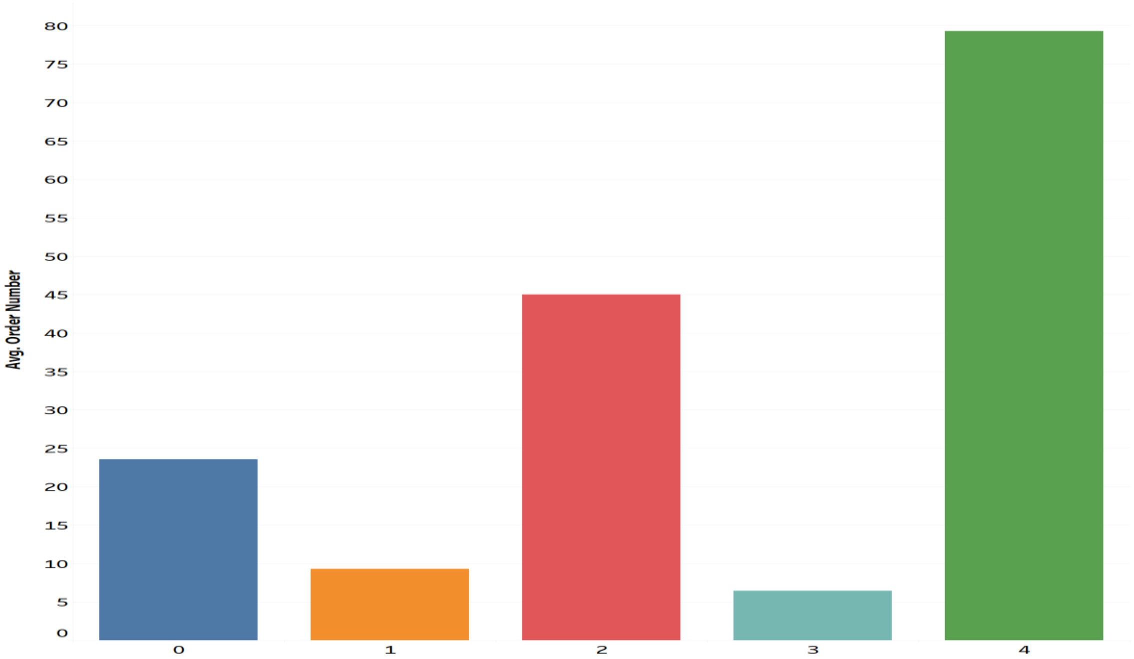


Fig 4.2.3: Average Number of Orders by a Customer

As noticed earlier, customers from cluster4 and cluster 2 are more frequent buyers and have more number of orders, but make up 12% of the customer-base

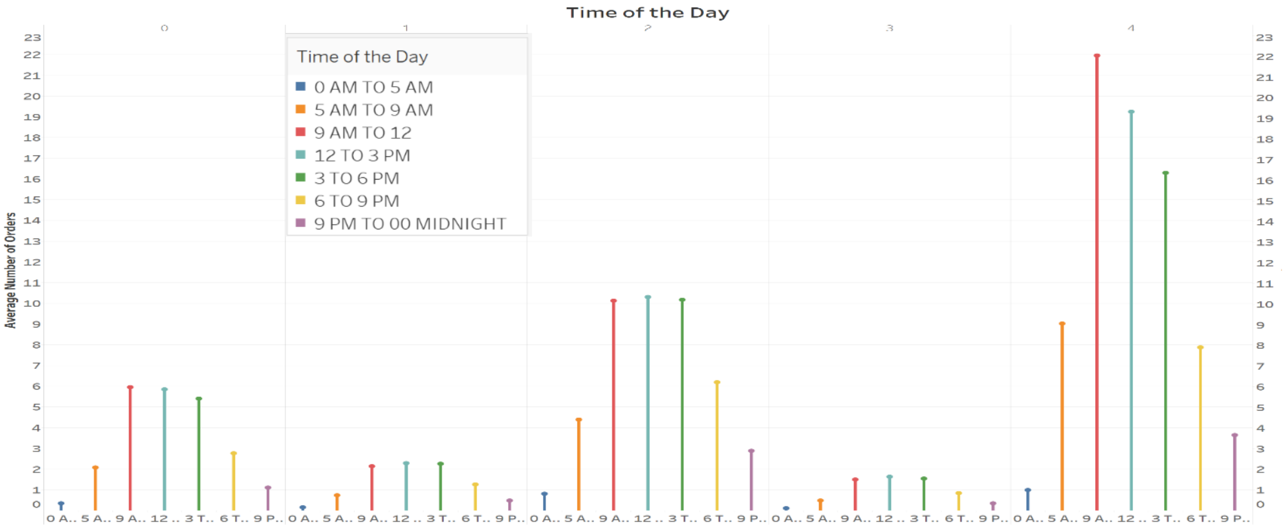


Fig 4.2.4: Customer Activity through the day

Distribution of customer activity throughout the day is similar across the clusters, i.e., increases during 9 AM to 6PM. For cluster 4 and 2, customer activity is in general very high relative to other clusters throughout the day (as noticed earlier). For cluster 4, customer activity is significantly high during 9 AM to 12 PM.

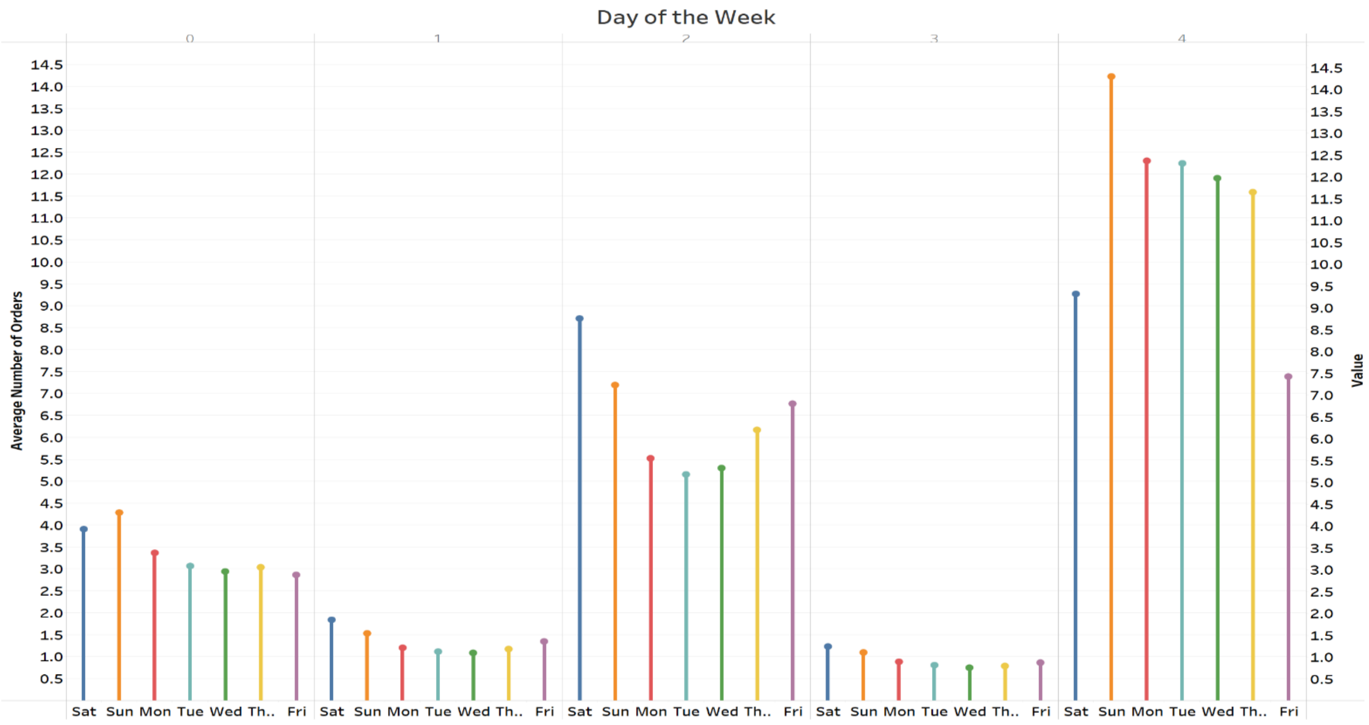


Fig 4.2.5: Customer Activity through the week

Average number of orders per user is around 3.2 for cluster 4 on Sunday, which is significantly higher relative to other clusters. Customer activity of cluster 3 throughout the week is relatively lower than other clusters. Cluster 4 follows a different trend of relatively higher customer activity during weekdays.

Major departments among the clusters of customers:

Major departments are those most preferred by the customers from each cluster. Major departments are obtained by sorting the departments based on the average value of average number of transactions per order by a customer from a department. Top 7 such major departments among different clusters are as follows:

```
Cluster0: 'produce', 'dairy eggs', 'beverages', 'snacks', 'frozen', 'pantry', 'bakery'

Cluster1: 'produce', 'dairy eggs', 'beverages', 'snacks', 'frozen', 'pantry', 'bakery'

Cluster2: 'produce’, 'dairy eggs', 'snacks', 'beverages', 'frozen', 'pantry', 'bakery'

Cluster3: 'produce’, 'dairy eggs', 'beverages', 'snacks', 'frozen', 'pantry',
          ‘canned goods’

Cluster4: 'produce', 'dairy eggs', 'beverages', 'snacks', 'pantry', 'frozen', 'bakery'
```

There is no notable variation in department preference across the clusters.

Remarks:

Customerbase of 206209 users is classified into 5 clusters with cluster0 comprising of around 41k, cluster1 around 81k, cluster2 around 17k, cluster3 around 58k and cluster4 around 6k users. Clusters 2 and 4 represent highly active customers and clusters 0, 1 slightly inactive customers. Cluster 3 on the other hand consists of highly dormant customers.

5.3) Product Embedding using Word2Vec Analysis:

Projection of vectors obtained from word2vec of various products on to 3- dimensional space using T-SNE:

- Aisle: red-wine:

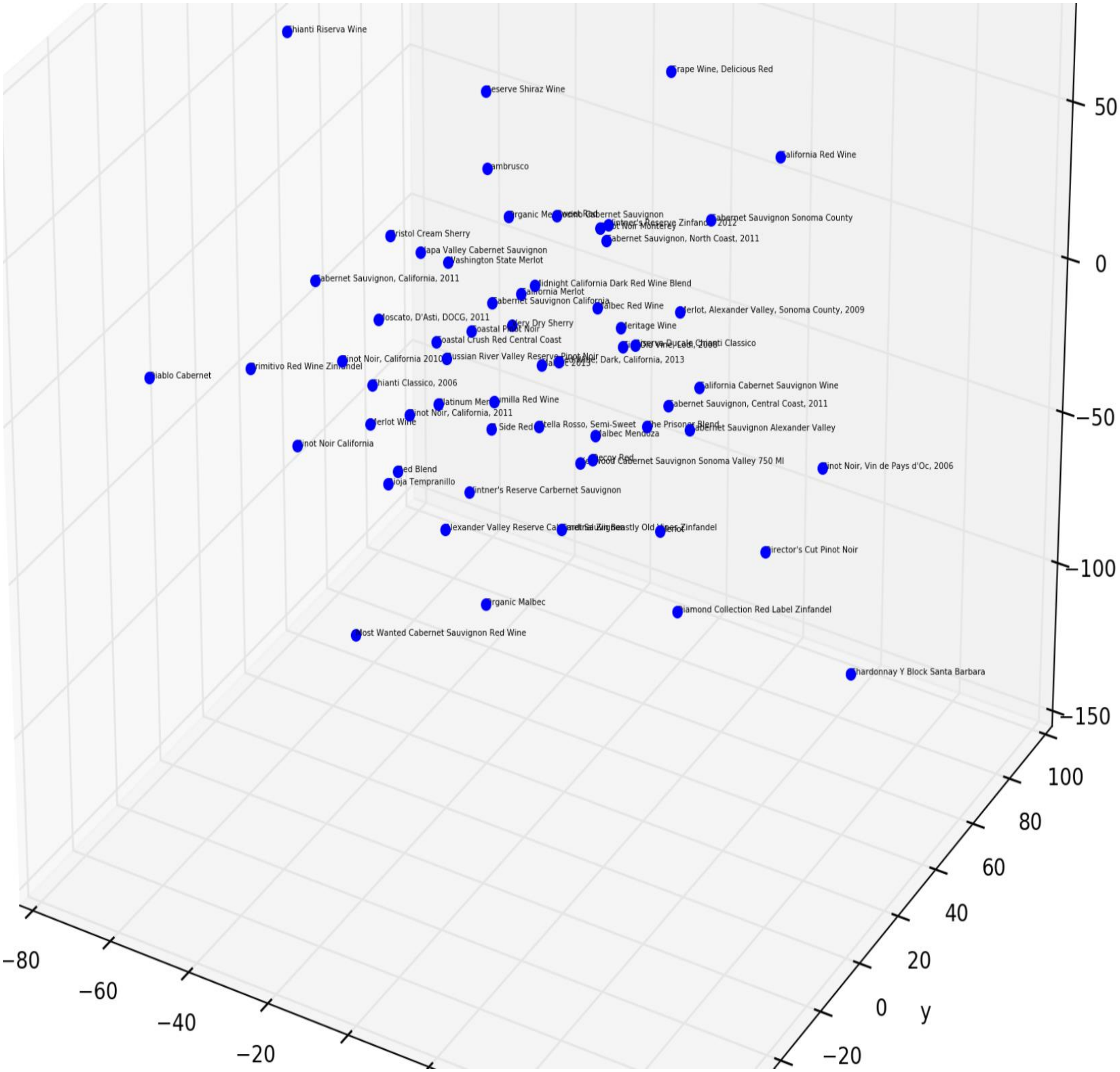


Fig 5.3.1: Red-wine Aisle

- Aisle – bread:

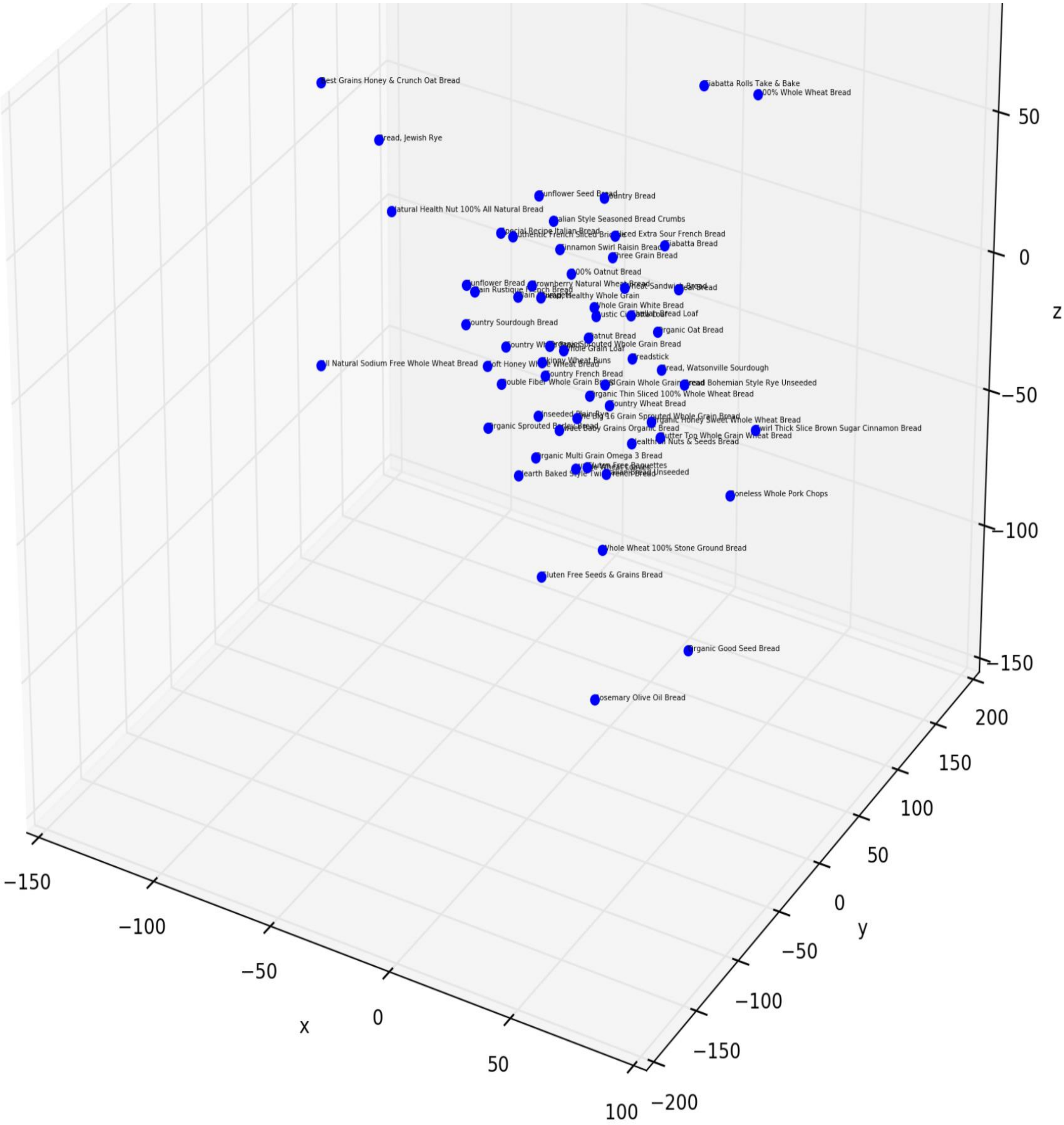


Fig 5.3.2: Bread Aisle

Above plots represent products from an aisle with similar characteristics represented by the 280-dimensional vector populated using skip-grams network grouped together. As it is impossible to visualize products in a 280-dimensional space, these vectors have been represented in three dimensional space using TSNE. Though there is nothing much to infer from these product embedding as it is, it is very useful in building recommender systems.

6) References:

- https://tech.instacart.com/3-million-instacart-orders-open-sourced-d40d29ead6f2
- http://scikit-learn.org/stable/auto_examples/cluster/plot_mini_batch_kmeans.html
- https://lvdmaaten.github.io/tsne/
- Udacity.com – Foundations of Deep Learning (Word2Vec analysis)
- Efficient Estimation of Word Representations in VectorSpace, arXiv:1301.3781v3 [cs.CL] 7 Sep 2013 (https://arxiv.org/pdf/1301.3781.pdf)
- http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/
- https://en.wikipedia.org/wiki/Association_rule_learning
- https://en.wikipedia.org/wiki/Apriori_algorithm
- https://www.tensorflow.org/tutorials/word2vec
- Visualizing Data using t-SNE, Journal of Machine Learning Research 9 (2008) 2579-260
- http://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html