

Identifying Product Bundle from Market Basket Analysis

Ashish Neupane^{1*}, Bibek Dhakal², Bijay Aryal³, Nabin Kandel⁴

¹Himalaya College of Engineering, address, Nepalgunj, Nepal, ashishneupane1997@gmail.com

²Himalaya College of Engineering, Pokhara-4, Pokhara, Nepal, dhakalbibek84@gmail.com

³Himalaya College of Engineering, Kaligandaki-1, Gulmi, Nepal, bairjyal@gmail.com

⁴Himalaya College of Engineering, Kathekoal-6, Baglung, nabin03963@gmail.com

Abstract

Bundling has emerged as an increasingly popular promotional strategy, offering numerous benefits to buyers and sellers and aligning perfectly with the goals of a transaction process. From the consumer's perspective, bundling enables them to enjoy substantial savings, with an average percentage off, when purchasing a bundle package at a discount price. This significant cost reduction serves as a critical motivator for embracing bundling. Furthermore, bundling allows customers to streamline their purchasing experience by minimizing search costs. They can conveniently find all the desired products and services in a comprehensive package offered by the seller. Additionally, bundles are favorable to some individuals due to their ability to mitigate compatibility risks between various components. From the seller's view, adopting bundling can lead to increased sales and a broader customer base. Bundling facilitates the attraction of buyers, while simultaneously raising awareness and acceptance of newly released products... The scope of this project, titled "Identifying Product Bundles from Sales Data using Market Basket," a highly performant model has been developed to aid in the identification of product bundles and market basket determination. Recognizing the multitude of prediction challenges during product sales, this project utilizes historical sales data to predict optimal product bundles. Leveraging a dataset obtained from Instacart, the project incorporates clustering and analysis processes. By thoroughly analyzing the data, the model can predict the most effective product bundles, enabling the selling company to boost its sales potential. This project specifically caters to e-commerce websites seeking to address product bundling and market basket analysis challenges. It provides a valuable platform for applying various techniques to solve problems associated with product bundling, generating comprehensive theoretical and practical resources for research-based studies. Ultimately, this project empowers businesses to make more reliable predictions about the future, enhancing their decision-making processes.

Keywords: Bundles, Clustering

1. Introduction

Have you ever wondered how you can simplify your daily routine? Maybe you find yourself rushing to the gym or making a grocery run after a busy day at work. That's where Instacart comes in – a same-day grocery delivery service designed to save you that trip to the market. With Instacart, you're just a few clicks away from connecting with personal shoppers in your area, who will handpick and deliver your groceries from your favorite stores in as little as an hour.

In this project, we delve into the immense potential of the Instacart Public Datasets to address three critical challenges:

Recommend New Products to Customers: We aim to enhance your shopping experience by suggesting new products that align with your preferences and needs.

Recommend Products to Be Bought Together: Through data-driven insights, we identify product combinations that are commonly purchased together, making your shopping experience more convenient and efficient.

Predict Which Products Will Be in Your Next Order: Anticipating your future needs, we employ predictive analytics to determine the items that are likely to be on your shopping list for your next word.

The significance of this project extends beyond convenience—it holds substantial business value for Instacart. In a highly competitive market with rivals like Amazon Fresh and Ship, Instacart aims to acquire and retain more customers. By simplifying the process of restocking your refrigerator and pantry with your favorites, we empower Instacart to provide delightful shopping experiences. Through precise recommendations and predictions, Instacart can boost sales and profits while attracting more customers and saving valuable shopping time.

While traditional methods involve purchasing products or services separately at their original prices, we propose a different approach—product bundling. In this strategy, we recommend bundles of products based on mining historical transaction data. This bundling strategy offers flexibility, allowing customers to choose between purchasing the entire bundle or individual items. Moreover, it assists the system in predicting which products you're likely to order next.

The core objective of our project is twofold: to recommend new products that cater to your preferences and to predict which products will make it on your next shopping list. With these innovations, we aim to redefine your shopping experience and empower Instacart to thrive in a competitive market.

2. Literature Review

From the literature review, we found how to drastically reduce the number of passes needed to obtain, in parallel, a good initialization. The proposed initialization algorithm k-means obtaining a nearly optimal solution after a logarithmic number of passes and shows that in practice a constant number of passes suffices (Bahmani, 2012). To find the optimal bundling of exclusive features by using a greedy algorithm can achieve quite a good approximation ratio which can reduce the number of features without hurting the accuracy of split point determination by much (Chen, 2017). Apply association mining to generate meaningful candidate bundles reduce computation costs and analyze consumer and product data, taking demand and inflation factors into consideration (Herve, 2012). Capable of predicting a customer's ultimate pleasure based on the customer's characteristics, a product will be recommended to the potential buyer if the model predicts his/her high level of satisfaction (Jiang, 2010). Formally, explained that low consumption level consumers prefer bundles composed of more commodities with lower prices. The product with the higher cost level should be bundled with a smaller bundle size and higher prices (L.YanFang, 2017). Studied to determine the value of newly available information by comparing the performance of decision-making on product bundling based on types of data on online shopping behaviors (Lai, 2006). The classification model is used to determine which product should be recommended to the customer. This model demonstrates the silhouette coefficient, support, confidence, and accuracy value are higher when both customer loyalty level and market segmentation variables are used in product bundling (M.Gholamin, 2018). Draw a set of key guidelines for bundling and pricing in stylized economic models and attempt to showcase the extant methodologies for bundle designing and pricing (Venkatesh, 2009). Simply observe the patterns of purchase, and this technology depends on the extent to which products are related to each other and the quantities sold (Mais, 2017).

3. Methodology

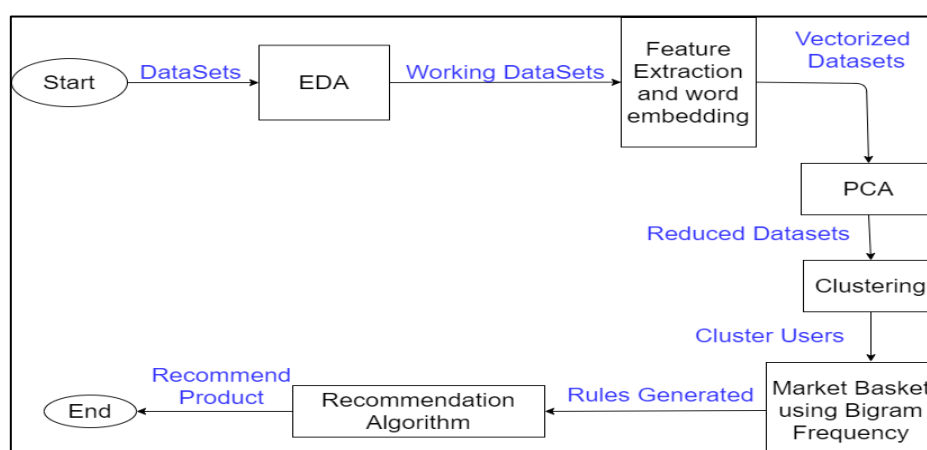


Figure 1. Workflow Diagram

The first step was to collect the dataset of around 3 million grocery orders from the two hundred thousand Instacart users. The dataset must include information about the order, order product, department of product, user’s details, and so on. Furthermore, exploring the dataset was necessary to look at the ordering habits of the customers each day. Combining orders made each day and each hour, Saturday evening and Sunday morning are the prime time for the orders.

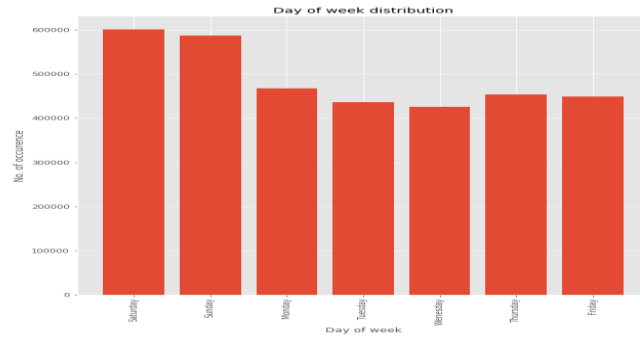


Figure 2. Weekly distribution

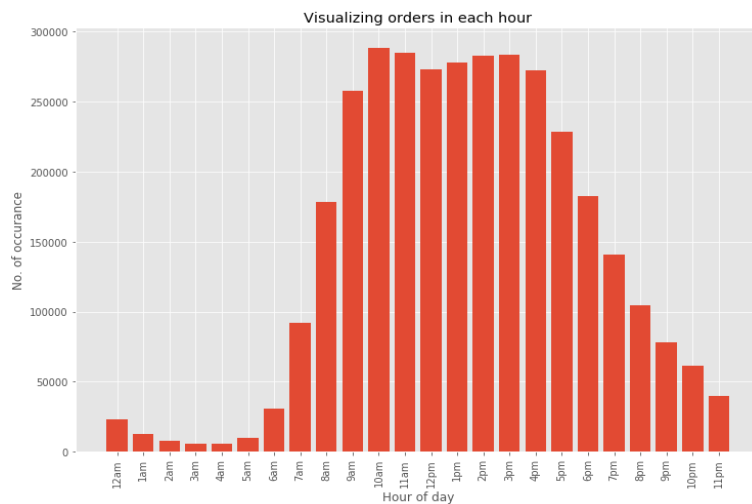


Figure 3. Order visualization

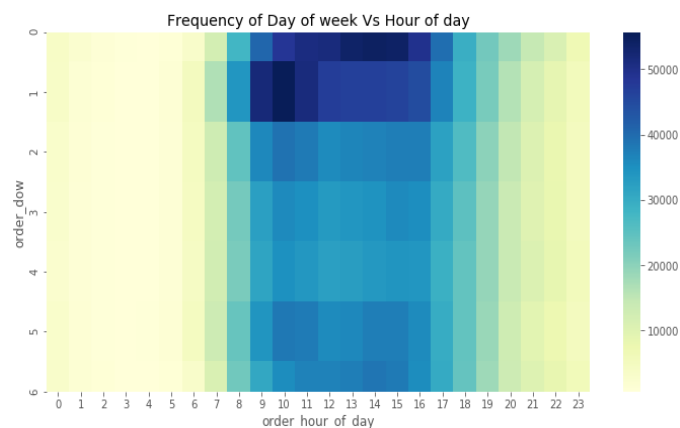


Figure 4. Frequency comparison

Once the final working dataset is ready, some of the features are directly extracted while others to be done by using word2vec analysis, by which the working dataset is reduced to more manageable groups for processing. The Word2vec technique is used for learning word embedding, which is the vector representation. So, the system does

not need to process plain text or string during clustering. For word2vec analysis, we combine all the product names into one row per user and the customer buying habit, which includes the day of week, hour, days since the last order, and the number of total orders. So, input data for word2vec in each row per user considering each row as a bag of words:

user_id	product_name
1	Soda Original Beef Jerky Pistachios Organic St...
2	Artichoke Spinach Dip Chipotle Beef & Pork Rea...
5	Uncured Genoa Salami Plain Whole Milk Yogurt W...
7	85% Lean Ground Beef Organic Apple Slices Appl...
8	Organic Baby Spinach Michigan Organic Kale Bag...

Figure 5. Input data for word2vec analysis

Such input is trained by the word2vec model. The model maps each word to a unique fixed-size vector. In our implementation of word2vec, we use the skip-gram model. The training objective of this model is to learn word vector representations that are good at predicting their context in the same sentence. Mathematically, given a sequence of training words $w_1, w_2, w_3, \dots, w_r$ the objective of the skip-gram model is to maximize average log-likelihood.

$$\frac{1}{T} \sum_{t=1}^T \sum_{j=-k}^{j=k} \log p(w_t + \frac{j}{w_t})$$

Equation 1

Where k = Size of training window

In the skip-gram model, every word w is associated with two vectors u_w and v_w which are vector representations of w as word and context respectively. The probability of correctly predicting word w_i given word w_j is determined by the SoftMax model, which is

$$\frac{\exp(u_{w_i}^T * v_{w_j})}{\sum_{l=0}^V \exp(u_l^T * v_{w_j})}$$

Equation 2

Where V is the vocabulary size

After learning a mapping from each row to vectors and saving the result as:

```

+-----+-----+
|           product_name |           result |
+-----+-----+
|[Hass, Avocados, ... | [0.35486072038398... |
|[Dulce, de, Leche... | [-0.4102908031394... |
+-----+-----+
only showing top 2 rows
    
```

Figure 6. Output from Word2vec analysis

To use it or apply it as input for k-mean clustering, we will save all the vector values by reshaping all the values into five vectorized features as:

	vectorized_feature_1	vectorized_feature_2	vectorized_feature_3	vectorized_feature_4	vectorized_feature_5	user_id
0	0.354861	0.275951	-0.142631	0.043631	-0.104238	160334
1	-0.410291	0.491902	-0.108292	-0.270677	-0.421957	88565
2	0.023476	0.175154	-0.221996	-0.103921	-0.421835	79012
3	0.196822	0.120681	-0.193381	-0.111424	-0.134239	169304
4	0.345206	0.037274	-0.129840	-0.088376	-0.154438	55886

Figure 7. Vectorized Dataset

The obtained result after word2Vec analysis is further reduced to 2-Dimension using PCA (principal component analysis) while preserving as much information as possible.

Step 1: Standardization

In the first step, we standardize or normalize the range of continuous initial variables so that each one of them contributes equally to the analysis. Mathematically, this can be done by subtracting each value from the mean and dividing by the standard deviation for each variable's value.

$$Z = \frac{\text{value} - \text{mean}}{\text{standard deviation}} \quad \text{Equation 3}$$

Once the standardization is done, all the variables will be transformed to the same scale, which will solve the problem of dominance that means, if there are significant differences between the ranges of initial variables, those variables with extensive ranges will dominate over the small range, which is not acceptable.

Step 2: Calculation of covariance matrix

The step aims to understand how the inputs vary from the means concerning variables each other or, in other words, to see if there is any relationship between them. Since the dataset we took is 2-dimensional, this will result in a 2*2 covariance matrix.

$$\text{Matrix}(\text{covariance}) = \begin{bmatrix} \text{cov}(x, x) & \text{cov}(x, y) \\ \text{cov}(y, x) & \text{cov}(y, y) \end{bmatrix} \quad \text{Equation 4}$$

It is the sign of the covariance matrix that matters:

- If positive, then: two variables increase or decrease together (correlated)
- If negative, then: one increases when the other decreases (inversely correlated)

Step 3: Computing the Eigenvectors and corresponding Eigenvalues of the covariance matrix to identify the principal components.

For a 2-dimensional dataset, there are 2 variables, therefore there are 2 Eigenvectors with 2 corresponding Eigenvalues. Eigenvectors of the covariance matrix are the directions of the axes where there is the most variance (most information) which we call principal components and Eigenvalues are simply the coefficients attached to Eigenvectors.

γ is an Eigenvalue for a matrix A if it is a solution of the characteristic equation:

$$|\gamma * I - A| = 0 \quad \text{Equation 5}$$

where I is an identity matrix of the same dimension as A. For each Eigen value γ , a corresponding Eigen vector v can be found by solving:

$$(\gamma * I - A) * v = 0 \quad \text{Equation 6}$$

Step 4: Choosing components and forming a feature vector.

We order Eigenvalue from largest to smallest so that it gives us components in order of significance. To reduce the dimensions, we choose p as the first eigenvalue and ignore the rest. We lost some information in the process but if values are small, we do not lose much information.

Next, feature vectors are formed which are Eigenvectors. For 2-Dimension case:

$$\text{Feature vector} = (\text{eigenvector1}, \text{eigenvector2}) \quad \text{Equation 7}$$

Where eigenvector1 has more significance or carries more information than eigenvector2.

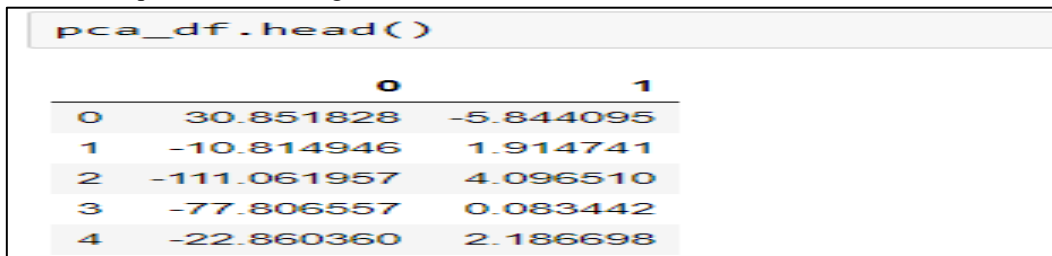
Step 5: Forming principal components.

To form principal components or new variables, we take the transpose of the feature vector and multiply it with the transpose of the original dataset.

$$\text{NewData} = \text{Feature vector}^T * \text{StandardizedOriginalData}^T \quad \text{Equation 8}$$

Where NewData is a matrix consisting of principal components and StandardizedOriginalData is version of original dataset.

From the above steps, we reduced large dataset into 2-Dimension as:



	0	1
0	30.851828	-5.844095
1	-10.814946	1.914741
2	-111.061957	4.096510
3	-77.806557	0.083442
4	-22.860360	2.186698

Figure 8. Reduced datasets after PCA,

After dimensionality reduction, the next step is to cluster users which is done based on their buying habits and preferences. The customer buying habit includes the day of the week, hour, days since the last order, and number

of total orders whereas preference includes number of total products and preferred products from word2vec analysis.

K-means is one of the simplest unsupervised learning algorithms that solve the well-known clustering problem.

Algorithm:

```

BEGIN
  Find the optimal number of cluster K using the elbow method and get the cluster centers  $m = \{m_1, m_2, \dots, m_k\}$ .
  Repeat
  For each object  $p \in D$ 
  set  $p$  to the cluster  $G_i \leftarrow \arg \min_i |p - m_i|^2$ 
  For each cluster  $G_i \in G$ 
   $m_i \leftarrow$  the mean value of all objects belongs to  $G_i$ 
  until no changes;

```

END

Before applying the k-mean clustering algorithm, we need to first find the optimal number of clusters. Either we go with a random number of clusters initially, or the best solution to make a balance between maximum accuracy and maximum compression is to find an optimal number of clusters. Here, the optimal number of clusters is found by calculating the the within-set sum of squared error (WSSSE). WSSSE is a metric that measures how good our clusters are. It works as:

- Step1: Error is calculated, which is the distance from each point of the dataset to it is centroid (final centroid in each cluster)
- Step 2: A square of that error is taken and finally summed up for the entire dataset dataset.
- Step 3: Plot the curve of WSSSE according to the number of clusters.

Following the above step, WSSSE is just the measure of how far apart each point is from its centroid. Obviously, if there is a lot of error in our model they will tend to be far apart from our centroid. So, the best choice to find the optimal number of clusters is to look at the elbow of WSSSE graph, which is also known as the elbow method. The location of a bend (knee) in the plot is generally considered as an indicator of the appropriate number of clusters that is 40 as shown in the figure in our case.



Figure 9. WSSSE Of K-means.

Once the optimal number of clusters for each user is calculated then the center for each user is also calculated and visualizing the centers, it looks like this:

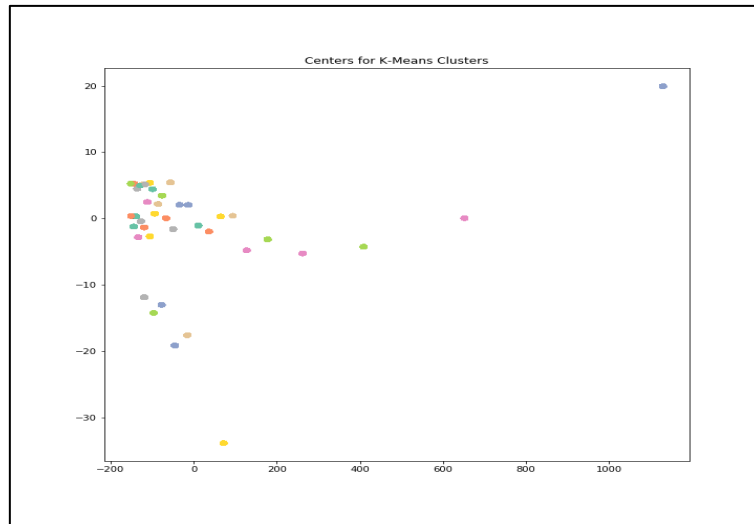


Figure 10. Centre Of K-means Clusters

After applying K-mean clustering, the result below shows the top common products in all clusters. Observing individually in all clusters, it was also found that bananas, bags of organic bananas, organic strawberries, organic Hass avocado, Limes are top products in each cluster. In this way the most selling products are found.

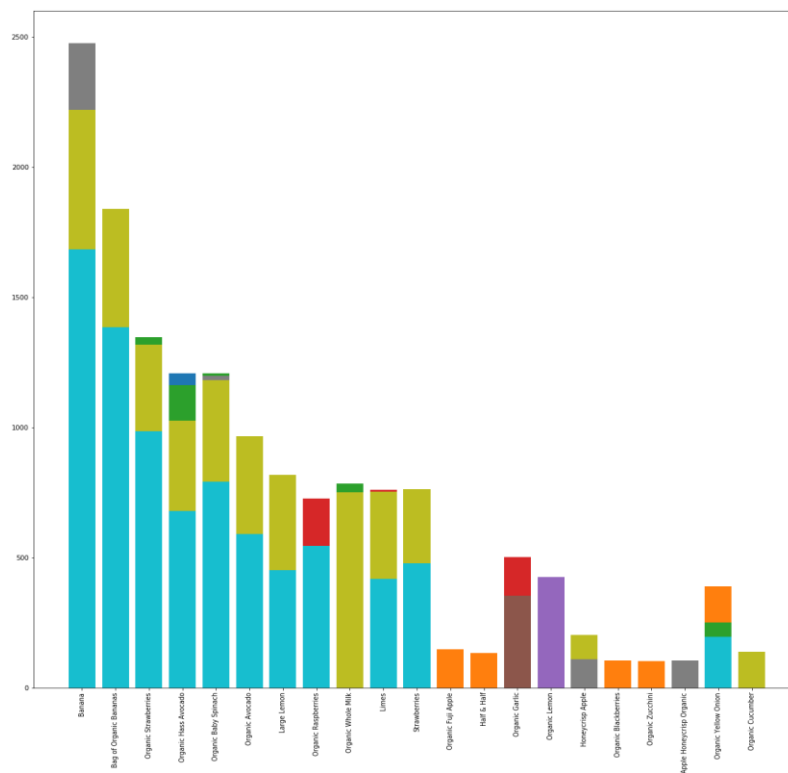


Figure 11. MSPL using k-means clustering.

The final task is to find out which products are frequently bought together to generate rules. For this purpose, we will use bigram and count bigram frequency. A bigram is a sequence of two adjacent elements from a string of tokens, which are typically letters, syllabi, or words. The frequency distribution of every bigram in a string is commonly used for simple statistical analysis of text in many applications, including computational linguistics, cryptography, speech recognition, and so on (Collins, 1996). The probability of a token $W_{(n)}$ given the preceding

token W is equal to the probability of their $n-1$ bigram, or the co-occurrence of the two tokens $P(W, W)$ divided by the probability of the preceding $n-1$ n token.

Step 1: Extract bigrams and calculate bigram frequency.

Bigram is extracted based on the order id of each cluster which means in one order ID of each cluster group which products have been bought together. So, in our case, bigram will be in the format of

```
(
    Order ID: [products bought in that order ID]
)
```

Once the bigram is extracted, the bigram frequency is calculated which is obtained by adding the same products bought in that order.

Step 2: Save bigrams and bigram frequency to JSON file as rules.

Bigrams are stored in a nested dictionary wherever.

- The first layer key is the first word in a bigram.
- The second layer key is the second word in a bigram.
- The second layer value is the frequency.

Then, the dictionary is converted to a JSON file as rules that will be used by the recommendation algorithm to generate recommendations for each product.

The format for the rule is:

```
{
    Product Name: {
        [{Recommended product list: frequency}]
    }
}
```

We developed this algorithm for predicting product bundles; and recommend a list of products that are likely to be bought together if a customer buys a certain product.

- Suppose we would like to recommend k products to be bought together after product S and the number of bigrams starting with S is denoted as n .
- First, we sort the frequencies for each bigram starting with the product S in decreasing order, so we have n bigrams arranged in order from the highest frequency to the lowest.
- Second, we compare k with n and fill the recommendation list with products in bigrams one by one based on the ordered bigram frequency. We consider the following two cases:
 - When $n \geq k$, if after sorting, there are some bigrams with the same frequency, we will pick each one with equal probability until the recommendation list reaches the total number of k
 - When $n \leq k$, we will not have enough product to fill the recommendation list of product S . In this case, we will first recommend all these n products. Then goes back to the product that is of highest frequency in bigrams of S , denoted as T , and fill the rest places in recommendation list with the products followed by T , adopting the same rule as before

```
Try an example: 15 Products recommended after "Organic_Mint_Bunch".

> print(getRecommend("Organic_Mint_Bunch", 15))

['Coco_Crunch_Sprouted_Granola', 'Organic_Oat_Non-Dairy_Original_Beverage', 'Vine_Ripe_Tomatoes', 'Trilogy_Kombucha_Drink', 'Sherry_Vinegar', 'Banana', 'Organic_Navel_Orange', 'Organic_Sliced_Peaches', 'Banana', 'Peanut_Butter_Chocolate_Chip_Fruit_&_Nut_Food_Bar', 'Petite_Brussels_Sprouts', 'Flaky_Biscuits', 'Organic_Avocado', 'Original_Hummus', 'Cucumber_Kirby']
```

Figure 12. Recommendation Algorithm Example

4. Discussion and Result

After exploring the datasets, merging the required columns, and removing the incomplete records the final working dataset is ready using Pandas and NumPy for feature extraction. Some of the features are extracted manually based on the total number of products brought, total number of orders made, and mean of orders placed during the day of the week and the hour of the day while some are extracted using word2vec analysis and for this PySpark is used for working with such large datasets where we combine all product name into one row per user and based on the customer buying habits then each bag of words are transformed into the vector using word2vec models which result into dimensionality reduction and more reliable model estimation. Then the result obtained is reduced further to 2-Dimension using PCA and further clustered the user based on their buying habit and preferences. K-means is used for clustering and the optimal number of clusters is found by calculating the WSSSE metric. After clustering the most selling products are found in each cluster as a result. The final task involves generating rules for products that are frequently bought together using market basket analysis and saving the rules in JSON format for working with the recommendation algorithm. The algorithm predicts the product bundles and recommends a list of products that are likely to be purchased together if a customer buys a certain product. Then Django is used as the backend and JavaScript is used on the frontend to allow customers to interact with our system where customers can place their orders and based on stored rules on JSON, display the recommended products to them. Thus, the system

- Recommends customers with favorite products.
- Recommends bundles of products to customers.
- Predicts which product the customer will buy.
- Helps E-commerce website on increasing sales.

5. Conclusion and Enhancement

5.1 Conclusion

Word2vector, k-means algorithm, and different algorithmic processes have helped to generate the market basket of the product very effectively. We implemented three major functions - recommending new products to customers, recommending product bundles, and building predictive modeling on a user's next order - on the Instacart public datasets. In terms of recommendation, we are predicting an accuracy of around 20%, which means, that 1 in 5 products we recommend using our algorithm is purchased by the customer. We do think it can create business value for Instacart and some techniques may transfer to other industries.

Future Improvements could be made in these two aspects:

- Create more correlated features.

5.2 Future Enhancement

- None of payment method is integrated, Integration of different payment method will enhance the system.

Acknowledgments

The success and outcome of this project require a lot of guidance and assistance from many people, and we are extremely privileged to have got this through the start of the project. This project is possible only due to such supervision and assistance of the Teaching staffs of the Electronics and Computer Engineering department and we would not forget to thank them. We respect and thank our department for providing us with such an opportunity and helping us directly or indirectly to do this project. We are thankful to our project coordinator **Er. Narayan Adhikari Chettri** and supervisor **Er. Ramesh Tamang** for providing us with constant encouragement, support, and guidance.

We would like to express our deepest appreciation to all those who saw the possibility in us for doing this project. A special thanks to those who contributed stimulating suggestions and encouragement and helped us to coordinate our project, especially in writing this report.

We are thankful to our college administration for equipping us with all the resources and providing us with a pleasant environment to work in. Also, we would like to thank our seniors and colleagues for their valuable comments and suggestions for this project.

References

- Bahmani, B., 2012. Scalable k-means++. *Proceedings of the VLDB Endowment*, March, Volume 5, pp. 622-633.
- Chen, T., 2017. NIPS'17:Proceeding of the 31st International Conference on Neural Information Processing Systems. *LightGBM: a highly efficient gradient boosting decision tree*, December, Volume 12, pp. 3149-3157.
- Chen, T., 2017. NIPS'17:Proceeding of the 31st International Conference on Neural Information Processing Systems. *LightGBM: a highly efficient gradient boosting decision tree*, December, Volume 12, pp. 3149-3157.
- Herve, P., 2012. The Role of Bundling in Firm's Marketing Strategies: A Synthesis. *Recherche et Application en Marketing*, Volume 27, pp. 91-105.
- Jiang, J., 2010. Maximizing Customer Satisfaction through a online recommendation system:A novel associative classification model. *Decision Support Systems*, February, Volume 48, pp. 470-479.
- Jiang, J., 2010. Maximizing Customer Satisfaction through an online recommendation system: A novel associative classification model. *Decision Support Systems*, February, Volume 48, pp. 470-479.
- L.YanFang, 2017. Bundle-Pricing Decision Model of Multiple Products. *Procedia Computer Science*, Volume 112, pp. 2147-2154.
- Lai, T.-C., 2006. Comparison of Product bundling strategies on different online. *Science Direct*, 12 June.pp. 295-304.
- M.Gholamin, 2018. A novel model for product bundling and direct marketing in e-commerce based on market segmentation. *Decision Science Letters*, January, Volume 7, pp. 39-54.
- M.Venkatesh, 2009. The design and pricing of bundles.A review of normative guidelines and practical approaches. *Handbook of Pricing Research in Marketing*, January.pp. 232-257.
- Mais, Q. O., 2017. Market Basket Analysis. *Management Information Systems*, Volume 6, pp. 50-62.
- Mehta, M., 2012. *Instacart*. [Online] Available at: <http://www.instacart.com/datasets/grocery-shopping-2017> [Accessed 25 February 2020].
- Venkatesh, M., 2009. The design and pricing of bundles.A review of normative guidelines and practical approaches. *Handbook of Pricing Research in Marketing*, January. pp. 232-257.