Foundation University Journal of Engineering & Applied Sciences
Vol. 1, Issue. 1. June 2020

Research article

Data Mining Assisted Purchase Prediction

Muhammad Faseeh*[1], Zahoor-ur-Rehman[1]

[1] Department of computer science, COMSATS University Islamabad Attock campus, Pakistan

*Corresponding Author: Muhammad Faseeh Email: ifaseeh1@gmail.com

Abstract:

With the revolution from physical businesses to shopping online, predicting client behavior in e-commerce is becoming increasingly important. It can increase customer satisfaction and sales, resulting in higher conversion, by enabling a more individualized shopping process. Today, most users want to save their time using and they prefer to shop using the platform provided for e-commerce. Millions of transaction records are available in the databases of such websites using which, a customer shops something. Using the transaction to find something can be helpful for the organization or merchants. Using the available databases or datasets, to find some useful pattern can increase the business, to check out the customer satisfaction level, to check the customer behavior about the product, etc. Some of the useful information can be to find out which item will be purchased by the customer in the next visit, or which new items can be purchased by the customer in the next visit. Using this information, an organization or Shops can control the quantity and increase the maximum purchased items, improving the quality of products for the customers. We use supervised learning techniques for prediction Because most of the data is labeled. Many researchers used supervised methods but some of the researchers also used unsupervised methods too. We created a supervised model for predicting the basket items. Due to the large dataset, it was very difficult to extract the features and it takes a lot of time. We have performed feature engineering, to choose the best ones. After the training model, our model shows better performance than the previous results.

Keywords: Basket Prediction, MBA, e-commerce

1. Introduction

Today, many tech companies are putting high investment to purchase customer transaction information from the different available databases. To extract meaningful information and patterns from these databases is one of the major challenges. Several studies are being conducted on market basket analysis such as popular product finding, customer interest products, patterns involve in a multi-branch store system[1] to maximize profit[2]. The market basket analysis technique is widely used by many companies as a means of discovering product associations and retailers on their promotional strategies. It is easy to make the normal decision such as where to place the product, what will be the price, how to promote the product, to define the profit, and which product is famous among the customers. The major advantage of the association technique is to put the relevant product closed with another product to increase the sale. For this, retailers must be familiar with the needs of the customer and take the necessary steps.

DM become particularly popular for the last many years after computers get computational power to calculate the huge amount of data. Mining techniques create ease to discover information from large amounts of data. Knowledge extraction is utilized to detect and investigate customer groups and to forecast upcoming conduct. DM is an operative technique to ensure enhanced service to the customer as well as to determine proposals proposed for their needs and incentives.

Market basket analysis provides decent reseller info around the relevant sales based on groups of goods. Customers who bought bread can be interested to buy relevant products such as bread, milk, butter, or jam. In this way, a sense is developed that these products must be placed near to others. Using this way customers

DOI: *10.32604/x.2021*

can easily access that product and put that product in the basket. There are some other products which are depending on these items. After identifying relevant products, place them near to these products which reminds the customer to take them conveniently. Market basket analysis is a technique of DM[3] that focuses to identify buying patterns of a customer by extracting association or co-occurrence of transactional data store. Market basket analysis helps to find out the products, which can be bought together and to rearrange the layout of the supermarket, to reshape an advertising campaign so that the purchase of the product can be improved. Therefore, it can be beneficial to analyze customer behavior to fulfill the needs, which is possible through different DM techniques.

DM facilitates the discovery of novel patterns in datasets via association rules, correlation, sequence analysis, classifiers, clustering, and a variety of other techniques. Extraction of association rules is one of the most challenging and popular tasks in DM. Correlation (alternatively referred to as association rule mining) identifies noteworthy relationships between a large number of data components. Association rules are established for often occurring things using support and trust as a criterion. A set of items with limited support is referred to as a frequent itemset. The percentage of transactions in a data collection that contain the itemset that support the itemset. Confidence is described as a level of certainty or assurance linked with the discovery of a pattern. Confidence is required to derive association rules. Data mining algorithms such as Apriori[4], FP-Growth algorithm[5], Eclat[6], and K-Apriori[7] are frequently used to generate itemsets. Knowledge Discovery in Databases is considered to be one of the most critical aspects of data mining (KDD). KDD is advantageous for identifying innovative, legitimate, and valuable patterns in data[8].

Instacart is one of the top growing businesses in groceries. Instacart is an online platform that makes it easy for the customer to buy favorite grocery items at any time. Using the online platform, the Customer adds the products to the digital basket and personal shoppers review the order. After reviewing the order, personal shoppers make in-store shopping according to the order. The dataset used for the research contains about 3 million transaction records which are recorded for six months. Dataset is provided by the Kaggle contains retailers, departments, aisles, products, and items record information. Currently, Instacart is using transactional data for developing the systems. In this research we will try to find out:

a) How to find out which items/products will be bought by the consumer?

b) How to find which items will be purchased for the first time?

To find out the buying item/product based on the demographic information is crucial. It is also very difficult to predict which item will be bought on the next visit by a user with acceptable accuracy. It is desired to develop a model that can generate a list of items/products that are expected to be bought by an individual user.

The rest of the paper will elaborate on the implementation of the proposed model for cart prediction. In section 2, we discussed the related work to predict the cart items. Section 3 explains the methodology of our proposed framework and a detailed description of each module. Section 4 includes the results and evaluation and section 5 includes the conclusion and future directions.

2. Related work

Market basket analysis has been a very catchy topic for researchers and companies ever since the emergence of the first e-commerce store, where customers can buy products online and gave stores a chance to store data regarding personal and demographic sales over time. For the problem under consideration, a lot of algorithms are being used and a lot of other new and optimized versions of previous algorithms are being presented and created by every passing year. As we know that e-commerce is growing rapidly. Most of the users want to save their time using and they prefer to shop using the platform provided for e-commerce. Millions of transaction records are available in the databases of such websites using which, a customer shops something. Using the transaction to find something can be helpful for the organization or merchants. Using the available databases or datasets, to find some useful pattern can increase the business, to check out the customer satisfaction level, to check the customer behavior about the product, etc. Some of the useful information can be; to find out which item will be purchased by the customer in the next visit,

or which new items can be purchased by the customer in the next visit. Using this information, an organization or Shops can control the quantity and increase the maximum purchased items, improving the quality of products for the customers.

2.1 Market basket analysis

Trnka [16] describes the detailed implementation of the MBA of Six Sigma methodology. Data mining techniques provide useful opportunities in the business sector. One of the most famous techniques is known as Market Basket Analysis (MBA). Using the MBA technique with Six Sigma methodology, results are better than the previous ones and the performance becomes more batter. Yanthy et al. [17] try to explore the hidden pattern/knowledge which is present inside the data and study the proposed algorithms which are utilized to extract the meaning information. Some of the rules which are proposed are not beneficial. Proposed rules are evaluated using support, confidence, lift, gain, and other information. He also studies the relationship among algorithms and interesting measures.

Rastogi et al. [18], present the optimized association method on association rules that hold instantiated attributes. The basic aim was to determine the relationship among two or more items through support and confidence and analyze the behavior of two items is maximized or not. He also describes the useful techniques for searching the available space during the computation of categorical as well as numerical data. Neesha et al. [19] analyze the technological advancement in the field of data mining. In 2009, she performed an experiment using the three famous association mining algorithms namely Apriori, Tertius, and Predictive Apriori. In 2012, three algorithms were applied to different nature of the dataset. Results were better than the previous task. . This proves that combined techniques perform better than the single technique which is applied on market basket analysis.

2.2 Market basket analysis using association rule

Several types of research are being conducted to deal with market basket analysis and association rules. Many papers are published to elaborate the discovery of association rules and many of the papers are published on its application. According to an article published in [20], a 'extracted probability' can be used to adjust the confidence of MBA rules in order to develop more efficient rules. The concept of 'anticipated utility' was first introduced to MBA by [21] in a manner similar to this. This research defines "association rules of high utility" as rules that help the company achieve a specific business goal. With the use of the high-utility rule mining (HURM) approach, developed by Lee et al. [22], it is now possible to quantitatively characterise a firm's preferences using utility values. A high-utility rule, on the other hand, is a rule that can be met within a certain utility limit.

A similar technique begins by identifying an optimal solution in advance and then identifying a collection of relevant rules that satisfy it mathematically. This is known as high-utility itemset mining (HUIM) [10][23]. Yao et al. [24] used the same technique to study MBA, although other studies demonstrated that the HUIM algorithm can be slightly adjusted (e.g., Hu and Mojsilovic, 2007 [25]; Yao and Hamilton, [26]). Other authors devised a methodology called weighted association rule mining (WARM), stating that conventional association rule mining methods are inadequate at uncovering meaningful rules with low support but a high weight. The WARM approach in MBA addresses this issue by taking into consideration the weight of each item in the basket (Tao et al., 2003 [27]). Recently, some authors attempted to improve MBA by using product networks, which connect product nodes to edges that indicate the relationship between two things. For example, this method was used to establish a bipartite customer product network that links consumers to products.

2.3 Predicting customer purchase behavior

Three different research streams are committed to predicting purchase behaviour. Marketing and retailing researchers have made significant efforts to estimate client purchase behaviour, which can assist firms in finding potential customers of particular products or services and launching cross- and up-selling campaigns. The association rule technique, which was originally developed for market basket analysis, has gained popularity as a method for predicting customers' purchase behavior by the extraction of associations, or co-occurrences, from store transaction records. Following that, these purchasing patterns can be utilised to

forecast future behavior of customers. To represent the cross-selling effect, it develops a loss criteria similar to the association rule. A rational customer, according to Ge. At et al.[28], will choose the product which delivers the greatest total net utility. Traditionally, recommendation algorithms reduce the influence of product qualities on client purchasing decisions. The objective of this study is to include customers' preferences for product qualities into predictions of purchasing behaviour.

Liu, Guimei, et al. [29] develop a technique that makes it possible to grab the customer for the next 6 months using the Intelligent system. In 2015, Alibaba hosted an International competition for the repeat buyer prediction. Many profiles were created like users' profiles, merchant's brands, different categories of items, and their relation using extensive feature engineering. Using AUC score, different algorithms were applied on data like Factorization machine, Logistic Regression, Random Forest, Bagging of Random, GBM, XGBoost, and Blending Model. The winning solution was based on comprehensive feature engineering and model training.

Kazmi, Auon Haidar, Gautam Shroff, and Puneet Agarwal. [30] shows that Predicting Shopping behavior especially the repeat behavior of a customer is really meaningful in e-commerce. These analyses are very helpful especially in the advertisement because it directly affects budgeting, product placement as well as directly target the customer. So, this problem will be resolved using standard predictive models, which use ad-hoc features. Our proposed model can also abstract the different dimensions of data which is present in the transactional dataset. This model and engineering techniques are tested on Kaggle-AVS and IJCAI-RBP datasets too.

You can't rely on only listing contacts, you need full information about the past purchases of the customer to predict future sales. Gupta, Aditya Kumar, and Chakit Gupta [31] measure customer stratification especially when you want to grow your business. The main aim is to focus on customer business strategies using data mining techniques. Techniques include Classification, regression, Link analysis, and segmentation. Nikulin [33] try to predict the new products which will be purchased by the existing customer during the discounts offers. For the merchants, the most valuable customer is one, who comes back from the shopping without any offer. For this purpose, the purchase history is utilized which contains a huge amount of data. later, huge data becomes compressed and is transferred to the data stream (Standard Rectangle format). After this practice, ranks become assigned based on their loyalty or intentions. Later, this practice merged with the practical application. For the testing, this model is tested with the dataset of the Kaggle-based Acquire Valued Shoppers Challenge in 2014[32].

2.4 Binary Classification

Data mining has a variety of applications, the most popular being classification. Classification, as a predictive analytics task, seeks to forecast a categorical target variable from a group of input factors. This target variable can be stated in a variety of ways, including categorical or binary. Because the goal variable has two categories: buying and not buying, the task is a binary classification task. A generalised link between the input and target variables is learned using a labelled dataset in order to predict the target variable. It is then used to classify new data. Algorithms for machine learning should be able to fit the training data and adapt effectively to new data. Numerous machine learning algorithms exist that employ a variety of different techniques for deriving this link..

2.5 Learning algorithms for binary classification

Vector-based approaches are the most often used form of machine learning algorithm for binary classification tasks. This category includes DTs, RFs, SVMs, LRs, and FNNs. This study's baseline model also corresponds with this category being a component of the DT algorithms. These techniques are all using supervised learning to learn given a set of feature vectors and matching outputs[34]. They are eager learning models, in which a classification model is formed using a training dataset and then used to classify additional data. On the other hand, lazy learners, such as the K-nearest Neighbour (KNN) algorithm, merely store training data and wait for a test data point to be classified. All of the approaches described above are stateless machine learning algorithms; they lack memory and always provide the same result when given the same input. This is suitable for the majority of classification jobs, but it is difficult to represent patterns across time, as the many states must be modelled using extensive feature engineering, introducing inaccuracies and rising complexity as the number of input characteristics increases.

2.5.1 Decision Trees

A DT is a collection of split conditions that divides a heterogeneous population into smaller, more homogeneous subgroups based on a single characteristic. The objective is to form the most homogeneous subgroups possible. There are other techniques for finding the optimal splits, including the Hunts algorithm, which employs a greedy strategy based on local optimum decisions [35]. Simple DTs have the advantage of being easily converted to straightforward classification rules. This level of comprehension diminishes as models become larger and more imbalanced [36]. In general, DTs provide a pretty rapid rate of learning and prediction. While the various varieties vary in terms of comprehension, they are all more understandable than black-box models such as FNNs or SVMs. The disadvantages include the requirement for feature engineering, the inability to implicitly describe temporal sequences, and the additional complexity associated with trees containing multiple category variables. Non-ensemble DTs, or methods that use solely singular DTs, have a tendency to overfit and be unstable in the presence of noisy data.

Boosted Decision Trees*:* Boosted DTs are ensemble approaches that employ several DTs. A sequence of trees is constructed in this section, with each tree derived from the prediction residuals of the preceding three. The baseline model employed in this study is an illustration of this type of strategy. While boosted DTs have demonstrated their power in predictive analytics by winning numerous Kaggle machine learning competitions, they are less intelligible than simple DTs due to their multiple trees.

DTs have demonstrated excellent results when applied to very similar problems as this study. [37], for example, classified a session as a buying or non-buying session using several DT algorithms. Along with clickstream data, an additional setup containing item sales data was used to improve prediction performance. When only clickstream data was used, which is the most representative of our use case, a Bagging RepTree produced the best results. It was implemented in Weka, a tool for data analysis using several machine learning techniques[38], and demonstrated 0.824 precision, 0.808 recall, 0.806 F1-score, and 0.889 ROC Area under the curve (AUC).

DTs have demonstrated excellent results when applied to very similar problems as this study. [37], for example, classified a session as a buying or non-buying session using several DT algorithms. Along with clickstream data, an additional setup containing item sales data was used to improve prediction performance. When only clickstream data was used, which is the most representative of our use case, a Bagging RepTree produced the best results. It was implemented in Weka, a tool for data analysis using several machine learning techniques[38], and demonstrated 0.824 precision, 0.808 recall, 0.806 F1-score, and 0.889 ROC Area under the curve (AUC).

3. Methodology

3.1 Framework

Instacart is a grocery-on-demand start-up that, in 2017, released a dataset containing 3 million orders from 200,000 (anonymized) users. A now-completed Kaggle competition asked entrants to predict which previously purchased products would be in a consumer's next order. In addition, Instacart would like to develop recommender systems that could predict which products a use would buy for the first time and which products would be added to a user's cart in future orders. Section 6 makes predictions according to the Kaggle competition, that is, from a given user's set of previously purchased products, predict which of those products the user will ultimately order. The model includes different product application variants, for instance

- More precise predictions too, for example, auto-populate a user's cart
- Top-N most-likely products a particular user will purchase to, for example, display on a web page of fixed results

This section contains a detailed explanation of the implementation methodology framework which we utilized to structure this study. Figure 1 contains a detailed picture of our model:

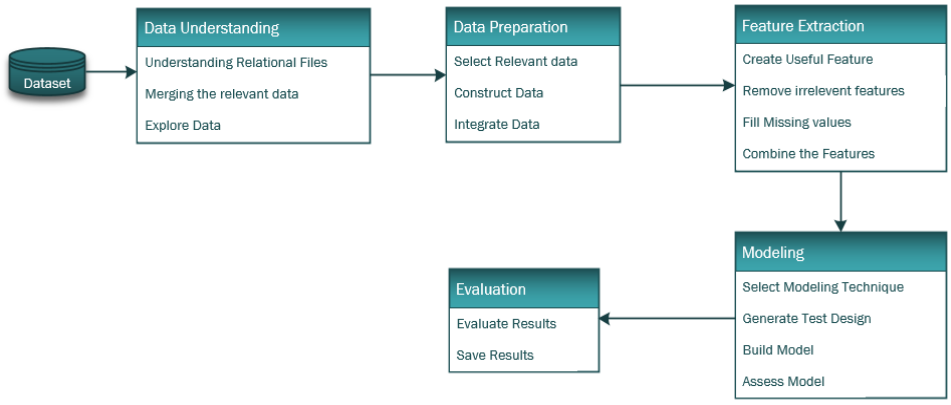

Figure 1: Methodology Framework Phases

- **Dataset:** Instacart is a grocery-on-demand start-up that, in 2017, released a dataset containing 3 million orders from 200,000 (anonymized) users.
- **Data Understanding:** Before starting the implantation, it is most important to understand the data briefly. It is important to know each feature that is available in the dataset, as well as the significance of that feature.
- **Data Preparation:** there are 6 different CSV files are available in the dataset. All the data which is available is relational. So, it is important to combine the data using different relational techniques likes joins.
- **Feature extraction:** after the data combining, the next step is to extract/create the new feature which will be helpful during the prediction process.
- **Modeling:** in this phase, model techniques will be applied. In this phase model selection and tuning of the model, parameters are involved.

3.2 Data: The Instacart Online Grocery Shopping Dataset

The Instacart public dataset release is a sample of 3 million orders from 200,000 anonymized users released in 2017. For each user, Instacart has provided between 4 and 100 of their orders, including the intra order sequence in which products were purchased, the week and hour of the day the order was placed, the relative time between orders, and the grocery store department to which each product belongs. A blog post by Instacart provides some information about the dataset and a Kaggle competition provides additional details. All data was obtained from the Kaggle competition website. The dataset is a relational set of .csv files which describe customer orders over relative times. Each entity (customer, order, department, etc.) has a unique id.

3.3 Exploration

It is necessary to explore each table for understanding. First, we display the *.csv files forming the Raw Data. Here are the names of all available files: aisles.csv, departments.csv, order_products__prior.csv, order_products__train.csv, orders.csv, products.csv, sample_submission.csv one important point is all the data that is present in the CSV file is relational. So, we have to merge all the tables into one table using the different SQL techniques. After that, we will apply further operations for getting the desired results. Figure 2 contains all product transactions record. Figure 3 contains all orders details done by a user. Below in Figure 4, the product table is depending on aisles and department table.

order_id	product_id	add_to_cart_order	reordered
1	49302	1	1
1	11109	2	1
1	10246	3	0
1	49683	4	0
1	43633	5	1

While only order_products__train.csv is displayed, order_products__prior.csv has the same structure.

Figure 2: order_products.csv

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

The abbreviated column headers are order_hour_of_day and days_since_prior_order.

Figure 3: orders.csv

(a) products.csv

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

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

(b) aisles.csv

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

(c) departments.csv

Figure 4: id's and product relation

Now, using all these files necessary patterns are being explored. Firstly, we are exploring order**_products_train** and **order_products_prior** files. These files contain the information on which products is purchased in which order. More precisely, **order_products__prior** contains previous order contents for all customers and **order_products_train** contains the last orders for some customers only. Considering all the available CSV files, we have found the following relevant information:

Table 1: Exploring data

Total orders	**33819106**
Total available products	49685
Total Prior orders	3214874
Total Train orders	131209
Total Test orders	75000

Here is a brief picture of relevant information extracted from the available data.

a) Total Number of products that people ordered

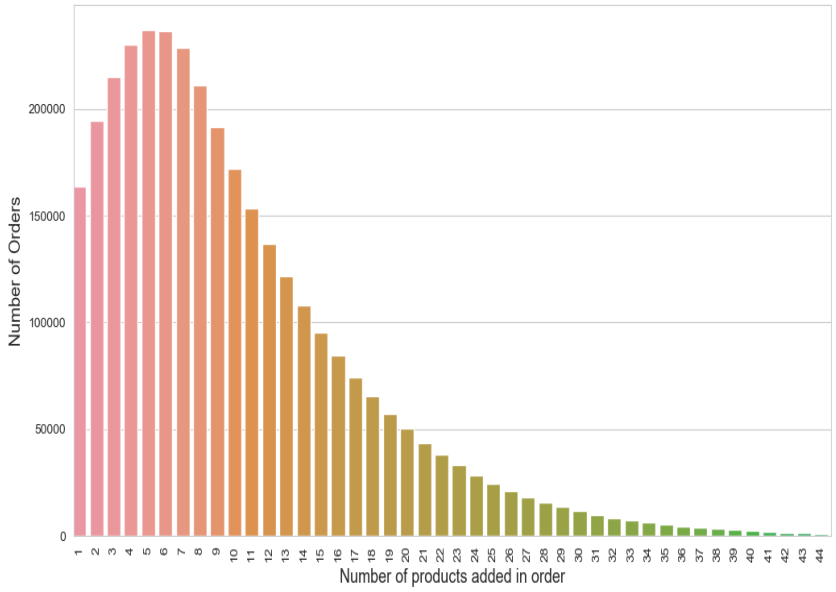

Figure 5: Total number of products ordered by people

b) Most ordered Products

Table 2: Most ordered Products

row_no	product_id	Total_reorders	product_name
24849	24852	491291	Bag of Organic Bananas
13173	13176	394930	Organic Strawberries
21134	21137	275577	Organic Baby Spinach
21900	21903	251705	Organic Hass Avocado
47205	47209	220877	Organic Avocado
47762	47766	184224	Large Lemon
47622	47626	160792	Strawberries
16794	16797	149445	Limes
26206	26209	146660	Organic Whole Milk

It is clearly illustrated that products which are mostly ordered are Fruits like bananas, and strawberries.

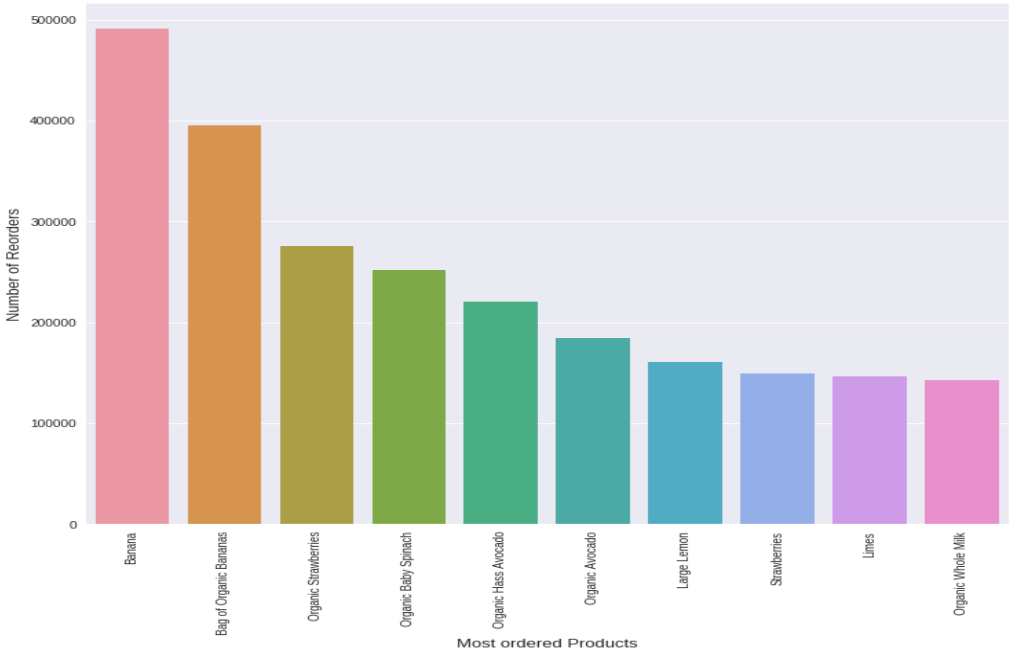

Figure 6: Most ordered products

c) Product Reorder Frequency

A product which is ordered/bought already, reordered or not. 0 represent 'No' and 1 represent 'Yes'.

Table 3: Product reorder frequency

reordered	Total_products	Ratios
0	13863746	0.410
1	19955360	0.590

Almost 59% of products were reordered.

d) Orders Time

Which products are ordered at which hour of the day? Most of the people ordered between 9 AM and 4 PM.

e) Order Days within a week

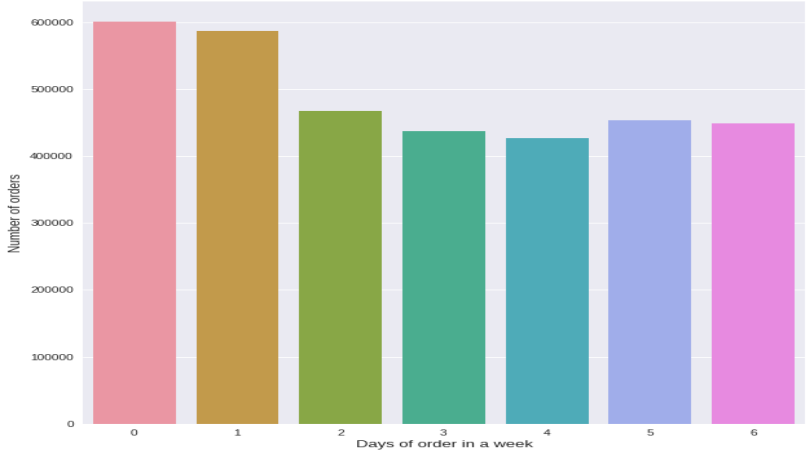

Figure 7: Order Days within a week

Most of the orders placed on Sunday (a week is starting from Sunday, denoted with 0)

f) Reorder days within a month

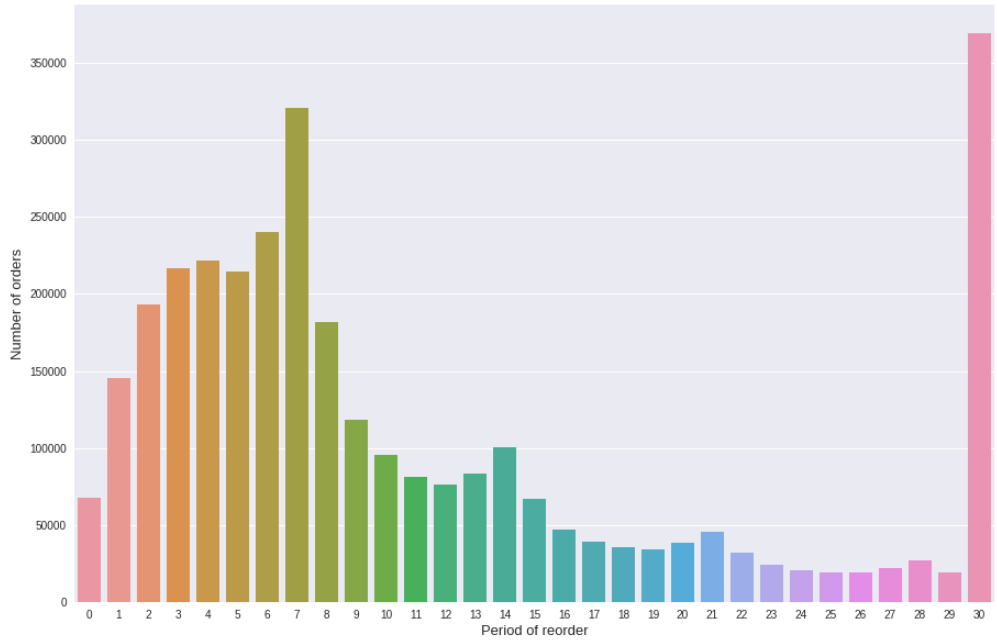

Figure 8: Reorder days within a month

Most orders are placed on the first weekend and last day of the month.

g) Most important Departments (by number of products)

Table 4: Important Department (By number of products)

department	Total_products	Ratio
personal care	6563	0.132
snacks	6264	0.126
pantry	5371	0.108
beverages	4365	0.088
frozen	4007	0.081
dairy eggs	3449	0.069
household	3085	0.062

h) Best Selling Aisles over all Departments.

Table 5: Important Department (By number of products)

Aisle	Total_orders	Ratio
fresh vegetables	150609	0.109
fresh fruits	150473	0.109
packaged vegetables fruits	78493	0.057
yogurt	55240	0.040
packaged cheese	41699	0.030

3.4 Feature Engineering

The brief description of the data dataset is illustrated in section 3.2 and the complete analysis is present in section 3.3.

3.4.1 Import the required python packages

For the implementation, pandas, NumPy, GC, scipy, xgboost, and sklearn are utilized.

3.4.2 Loading CSV files

We used 5 files provided by Instacart. Here is a short description of each file.

Aisles: Aisles file contains overall 134 records and it has two features. It shows that there are a total of 134 aisles in our store and each aisle has been assigned an identification number along with a unique name.

Departments: Store has 21 departments and they are represented by a name and an identification number in the given data. These departments hold different products separately.

Products: Each product Is presented by its unique identification number, its name, the aisle where it is placed, and the department to which it belongs. There is a total of 49,700 products currently listed for sale.

Orders: Each order has the following features: Order identification number, user's identification number who placed this order, evaluation set to which it belongs, number of orders i.e. serial number customers order, the day of the week when the order was placed, the hour when the order was placed, and days past since this user placed his last order.

Order Products: These are two files containing more than 3.4 million records. They are split into training and testing data by default. It contains information about individual products. The features in this file are the identification number of orders, the identification number of products, the order in which they were added to the cart, and a binary feature representing whether this product was ever reordered previously or not.

3.4.3 Merging all orders

In the first step, all orders from **orders CSV** and **order_products_prior** will be combined using inner join based on **order_id**. The new table keeps the **user_id**, all the orders placed by a user (**order_id**), and products Id's purchased by a relevant customer. Now it is easy to calculate the different features relevant to the product, user, and the combination of both.

3.4.4 User features

In this step, we created two features:

- Number of orders per customer
- How frequently a customer has reordered products?

3.4.4.1 Number of orders per customer

In this step, total orders which are placed by a customer are calculated.

3.4.4.2 How frequent a customer has reordered products

This feature is ratio-based calculation. Using the record of the customer, it is analyzed whether the user will reorder a product or not. Here is the simple formula for calculating the ratio.

$$prob.\ reordered(user\_id) = \frac{total\ times\ of\ reorders}{total\ purchased\ products\ from\ all\ baskets} \quad (1)$$

The nominator shows the products purchased by the user and reordered (reordered=1). The denominator contains all the product's statuses that are reordered or not (reordered=0 & reordered=1).

E.g., a user ordered a total of 6 products, 3 times products were reordered, the ratio will be:

$$mean = \frac{0+1+0+0+1+1}{6} = 0.5 \quad (2)$$

3.4.5 Product features:

In product features, we are interested in the following features:

- total purchases of each product
    - to calculate the probability of each product to be reordered.

3.4.5.1 Number of purchases of each product:

In this step, we calculated the total purchases of a product from all the transactions which are one by the user.

3.4.5.2 Product reordered probability

It is most important to find out which product has more chances to be reordered. To calculate the probability, the following formula is used.

$$prob.reordered\ (product_{id}) = \frac{number\ of\ reorders}{total\ number\ of\ orders} \quad (3)$$

For example, the product having product_id=4 is purchased 80 times in different purchases. After the first purchase, the product is reordered 14 times. So,

$$p\_reorder(product\_id == 4) = \frac{14}{80} = 0.175$$

3.4.6 User-product features

This step is concerned with the mixed features extracted from the user and product section. So, listed features are extracted using the user-product feature.

- Total time user bought the product
- How frequently a product is being purchased by the user after the first time
- Total times a shopper purchase a product in its last 5 orders

3.4.6.1 Total time user bought the product

We combine all orders of a customer and count each product purchased by a single customer.

3.4.6.2 How frequently a product is being purchased by the user after the first time

This feature is ration based feature in which overall transactions are observed.

This ratio describes, how many times a shopper purchases a product out of how many times are chances to buy it again (considering all the purchases from the start).

We use this formula to extract this feature:

$$prob.reordered\ (user\_id, order\_id) = \frac{Times\_bought\_N}{Order\_Range\_D} \quad (4)$$

Times_Bought_N = Times user purchased a product

Order_Range_D = Total orders placed since the first user's order of a product

The **Order_Range_D** variable is a combination of two values:

Total_orders = Total number of orders of each user

First_order_number = The order number where the customer bought a product for the first time.

3.4.7 Merge all features

As a final step, all features which are created above, and existing ones will be combined with a different label (test and train). Later, the file is divided into two different files namely test and train (based on test and train label). So, here is the complete list of all features in the training file.

user_id, product_id, uxp_total_bought, uxp_reorder_ratio, times_last5, u_total_orders, u_reordered_ratio, p_reorder_ratio, eval_set, order_id, reordered

3.5 Algorithm

3.5.1 LightGBM

We tried a lot of machine learning algorithms after the success of experiment 2 but none of them yielded any results other than Microsoft's Light GBM Algorithm. So, we used it in our final implementation.

Microsoft created Light GBM, a high-performance gradient boosting system based on decision tree methods. It is utilized in machine learning for ranking, classification, and a variety of other tasks [40]. It grows trees vertically instead of other tree algorithms which grow trees horizontally, meaning it grows itself leaf-wise instead of level-wise. We chose to use this dataset as it was the most common machine learning algorithm used for this dataset on Kaggle. As Pushkar at Medium also explains why it is gaining more

popularity among all machine learning algorithms. Basic reasons are its speed, accuracy, GPU support, handling large datasets, and taking up low memory[41].

3.5.2 XGBoost

XGBoost is becoming more popular in machine learning and data mining fields for supervised learning. It is designed for performing flexible and highly efficient tasks[42]. It is being implemented under the Gradient Boosting framework. XGBoost works along the parallel tree boosting which is also known as GBDT or GMB. XGBoost is utilized especially when we need accurate solutions in a short time. That is why XGBoost is becoming more popular in the data science field. XGBoost support is available with many other languages like Java, Python, etc. XGBoost performance on some machine learning tasks makes it more trustworthy for the data sciences field[43].

4. Experimental Results

We discussed the performance of our suggested XGBoost-based model for product cart prediction below. We conducted and assessed numerous trials to get the highest possible accuracy and to establish the best fit technique for product predictions. We conducted our studies using the Instacart dataset, which was initially available in relational format. First, we analyze the dataset deeply and then apply processing on our dataset and convert the relational dataset into a single file. Then using this model, we will test our model on testing data to calculate evaluation scores.

4.1 Dataset

The Instacart dataset for this competition is a relational set of files describing customer orders over time. The goal of the competition is to predict which products will be in a user's next order. The dataset is anonymized and contains a sample of over ~3 million grocery orders from more than 200,000 Instacart users.

The next section presents the experimental setup for our model. To achieve the best results, we have tried different models, and come to the end that the proposed model gave the best accuracy.

4.2 Experimental Setup

This section will validate our tests on the Instacart dataset. After the converting relational dataset into a single file, we applied different boosting models to come up with this model. Different boosting techniques were applied to the dataset. LightGBM and XGBoost are the models which are applied. Initially LightGBM perform better than all other previous results. Different parameters were changed to improve the accuracy and F1 score. When we tried XGBoost, the result becomes better than the LightGBM. Accuracy and F1 Score become better than LightGBM.

For this implementation, we use Core i7 (6th gen.), 16GB RAM, 256 GB SSD system.

Parameter setting: Parameters settings are the most important step in boosting algorithms. Wrong chosen parameters/settings may affect the overall performance of the model. In this experiment, we have selected different parameters and later we change their values for the best performance. The configuration parameter is different for each dataset. On this dataset, we experimented with a variety of factors. To our knowledge, these chosen parameters showed the best performance and produced the best outcomes. Table 6 details the parameterization of our model's values.

Table 6: Parameter setting of the proposed model

Parameter	Value
max_depth	3
learning_rate	0.1
n_estimators	100
objective	'binary:logistic'
booster	gbtree
n_jobs	1
min_child_weight	1

random_state	7
test_size	0.1

While designing our model, different parameters were affecting the result. Some parameters are the same in boosting models, but their parameters were affecting the overall results. We have applied LightGBM and XGBoost for training our model and evaluated the performances. As illustrated in Figure 9, the XGBoost algorithm achieved the highest accuracy on our dataset. Similarly, when we examined the size of the training data, we discovered that as the quantity of the training data decreases, the accuracy and F1 score decrease slightly.

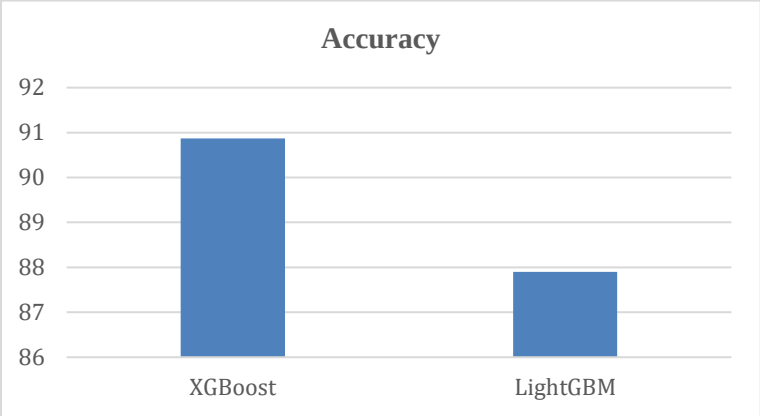

Figure 9: Comparison of XGBoost applied on the dataset.

4.3 Implementation

In our approach, we have predicted the cart items/products with better results. The proposed model was implemented in Python using the different libraries, for example, pandas, NumPy, xgboost, scikitplot libraries, etc. The model was trained for 8474661 records. We split the dataset into 80/20 (training/validation) during the training phase for the model accuracy. After the model training, we have tested the model on 4833292 test records. Our model shows good results during the training as well as on testing.

4.4 Evaluation Metrics

This section describes the evaluation metrics used to compare the algorithms. Resulting from the understanding, the evaluation will be based on performance metrics, interpretability as well as prediction latency. For the evaluation of our proposed model, we have used standard evaluation metrics that are as follows.

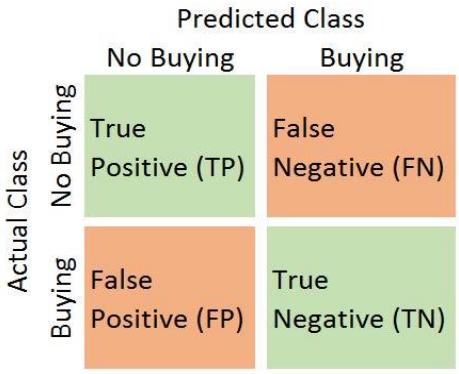

Figure 10: Confusion matrix, showing true positives, true negatives, false positives

True Positive (TP) are duplicate pairs that are correctly classified as duplicates.
True Negative (TN) are non-duplicate pairs that are correctly classified as non-duplicate samples.
False Positive (FP) pairs are non-duplicates that are misclassified duplicate pairs.
False Negative (FN) sentence pairs are duplicates that are misclassified non-duplicate pairs.

Numerous metrics exist for evaluating and comparing machine learning models' performance on a binary classification problem. These metrics are derived from the so-called confusion matrix, Figure 10, from which the correctly predicted situations, denoted in green as true positives and negatives, can be deduced. These are the instances in which a visitor did not make a purchase and a no-purchase session was anticipated, as well as the instances in which a purchase occurred and was also predicted. Additionally, as noted in orange, the incorrectly predicted cases can be recognized, including the false negatives and positives, in which a purchase occurred but none was expected, or where no purchase occurred but one was projected.

Accuracy is defined as the proportion of true negative and true positive samples concerning the total number of samples. Precision is a measure of the proportion of projected positive cases that occur. While recall refers to the fraction of true positive cases that were predicted correctly. The F-measure is the harmonic mean of precision and recall.

4.5 Results

This section presents our results and comparison with different existing approaches. Our proposed model is performing more efficiently than existing approaches. Abhishek [45,46] predicts the repeat purchase with 65%and 67% accuracy. While Pereira, E. [44] showed 87% accuracy. Our proposed model shows a 90.86% accuracy rate. Which is higher than the previous results. We have tested our proposed model on real-world datasets. Our proposed model shows effectiveness against the provided datasets. The results validate the suggested approach and indicate its broad potential applicability in e-commerce purchase prediction.

Table 7 Accuracy comparison of proposed Model with existing approaches

Model	Accuracy
Abhishek [44]	65%
Abhishek [45]	67%
Pereira, E. [44]	87%
Proposed Model (2019)	**90.86%**

4.6 Summary

In this section, we explained the detailed implementation and result details by applying Boosting model (XGBoost) to the Instacart dataset. Our model outperformed most of the existing approaches. Next, we will

conclude our proposed Boosting model.

5. Conclusion & Future Research Directions

We can see how seeing a dataset from a different dimension, engineering its features for the problem at hand, and engineering target class labels as per our own choice can result in such impressive results. Given more time, processing power and memory one can implement other algorithms on the features we have already extracted and saved in separate files to produce a comparative analysis of the predictions. The only hurdle in way of testing more and more machine learning algorithms on present datasets is memory overflow.

We conclude our work by a statement that performing exploratory data analysis, feature engineering, and class label engineering can result in great predictions even from such a basic machine learning model. We can also feed a model generated from one algorithm to another algorithm for predicting results to see how some algorithms can handle models from other algorithms and perform even better.

References

[1] S.-S. Weng and M.-J. J. E. S. w. A. Liu, "Feature-based recommendations for one-to-one marketing," vol. 26, no. 4, pp. 493-508, 2004.

[2] Y.-L. Chen, K. Tang, R.-J. Shen, and Y.-H. J. D. s. s. Hu, "Market basket analysis in a multiple store environment," vol. 40, no. 2, pp. 339-354, 2005.

[3] K. Kasemsap, "Multifaceted applications of data mining, business intelligence, and knowledge management," in *Intelligent Systems: Concepts, Methodologies, Tools, and Applications*: IGI Global, 2018, pp. 810-825.

[4] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proc. 20th int. conf. very large data bases, VLDB*, 1994, vol. 1215, pp. 487-499.

[5] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," in *ACM sigmod record*, 2000, vol. 29, no. 2, pp. 1-12: ACM.

[6] K. J. C. Babu, Techniques and Benefits, "Business intelligence: Concepts, components, techniques and benefits," 2012.

[7] L. C. Annie and A. J. I. J. C. I. I. Kumar, "Frequent Item set mining for Market Basket Data using K-Apriori algorithm," vol. 1, no. 1, pp. 14-18, 2011.

[8] G. Klepac and K. L. Berg, "Proposal of analytical model for business problems solving in big data environment," in *Web Services: Concepts, Methodologies, Tools, and Applications*: IGI Global, 2019, pp. 618-638.

[9] J. W. Kim, B. H. Lee, M. J. Shaw, H.-L. Chang, and M. J. I. J. o. E. C. Nelson, "Application of decision-tree induction techniques to personalized advertisements on internet storefronts," vol. 5, no. 3, pp. 45-62, 2001.

[10] A. F. Pathan, S. Palande, T. Shende, R. Patil, and V. S. Gutte, "A STUDY ON MARKET BASKET ANALYSIS AND ASSOCIATION MINING," in *Proceedings of National Conference on Machine Learning*, 2019.

[11] M. Prediger, R. Huertas-Garcia, J. C. J. J. o. R. Gázquez-Abad, and C. Services, "Store flyer design and the intentions to visit the store and buy: The moderating role of perceived variety and perceived store image," vol. 51, pp. 202-211, 2019.

[12] N. Schröder, A. Falke, H. Hruschka, and T. J. J. o. I. M. Reutterer, "Analyzing the Browsing Basket: A Latent Interests-Based Segmentation Tool," vol. 47, pp. 181-197, 2019.

[13] S. M'zungu, B. Merrilees, and D. J. J. o. S. B. M. Miller, "Strategic and operational perspectives of SME brand management: A typology," vol. 57, no. 3, pp. 943-965, 2019.

[14] R. Moodley, F. Chiclana, F. Caraffini, J. J. J. o. R. Carter, and C. Services, "A product-centric data mining algorithm for targeted promotions," p. 101940, 2019.

[15] Y.-S. Huang, Y.-H. Gu, C.-C. J. I. J. o. S. S. O. Fang, and Logistics, "Pricing of perishable products with a speculator and strategic customers," vol. 6, no. 4, pp. 301-319, 2019.

[16] A. Trnka, "Market basket analysis with data mining methods," in *2010 International Conference on Networking and Information Technology*, 2010, pp. 446-450: IEEE.

[17] W. Yanthy, T. Sekiya, and K. Yamaguchi, "Mining interesting rules by association and classification algorithms," in *2009 Fourth International Conference on Frontier of Computer Science and Technology*, 2009, pp. 177-182: IEEE.

[18] R. Rastogi, K. J. I. T. o. K. Shim, and D. Engineering, "Mining optimized association rules with categorical and numeric attributes," vol. 14, no. 1, pp. 29-50, 2002.

[19] N. Maheshwari, N. K. Pandey, and P. J. I. J. o. C. A. Agarwal, "Market Basket Analysis using Association Rule Learning," vol. 975, p. 8887, 2016.

[20] C.-W. Liao, Y.-H. Perng, and T.-L. Chiang, "Discovery of unapparent association rules based on extracted probability %J Decis. Support Syst," vol. 47, no. 4, pp. 354-363, 2009.

[21] S. Park, D. Lee, and W. Oh, "From free to fee: Monetizing digital content through utility-based business rule analytics," Working Paper, KAIST, College of Business, Seoul, Korea2012.

[22] D. Lee, S.-H. Park, and S. Moon, "Utility-based association rule mining: A marketing solution for cross-selling %J Expert Syst. Appl," vol. 40, no. 7, pp. 2715-2725, 2013.

[23] R. Chan, Q. Yang, and Y.-D. Shen, "Mining High Utility Itemsets," presented at the Proceedings of the Third IEEE International Conference on Data Mining, 2003.

[24] H. Yao and H. J. Hamilton, "Mining itemset utilities from transaction databases %J Data Knowl. Eng," vol. 59, no. 3, pp. 603-626, 2006.

[25] J. Hu and A. Mojsilovic, "High-utility pattern mining: A method for discovery of high-utility item sets %J Pattern Recogn," vol. 40, no. 11, pp. 3317-3324, 2007.

[26] G.-C. Lan, T.-P. Hong, and V. S. J. E. S. w. A. Tseng, "Discovery of high utility itemsets from on-shelf time periods of products," vol. 38, no. 5, pp. 5851-5857, 2011.

[27] F. Tao, F. Murtagh, and M. Farid, "Weighted Association Rule Mining using weighted support and significance framework," presented at the Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, Washington, D.C., 2003.

[28] Y. Ge, S. Xu, S. Liu, S. Geng, Z. Fu, and Y. Zhang, "Maximizing Marginal Utility per Dollar for Economic Recommendation," in *The World Wide Web Conference*, 2019, pp. 2757-2763: ACM.

[29] G. Liu *et al.*, "Repeat buyer prediction for e-commerce," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 155-164: ACM.

[30] A. H. Kazmi, G. Shroff, and P. Agarwal, "Generic Framework to Predict Repeat Behavior of Customers Using Their Transaction History," in *Web Intelligence (WI), 2016 IEEE/WIC/ACM International Conference on*, 2016, pp. 449-452: IEEE.

[31] A. K. Gupta and C. Gupta, "Analyzing Customer Behavior using Data Mining Techniques: Optimizing Relationships with Customer," *Management Insight*, vol. 6, no. 1, 2012.

[32] J. Qiu, Z. Lin, and Y. Li, "Predicting customer purchase behavior in the e-commerce context," *Electronic commerce research*, vol. 15, no. 4, pp. 427-452, 2015.

[33] V. Nikulin, "Prediction of the shoppers loyalty with aggregated data streams," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 6, no. 2, pp. 69-79, 2016.

[34] J. Heaton, "An empirical analysis of feature engineering for predictive modeling," in *SoutheastCon 2016*, 2016, pp. 1-6: IEEE.

[35] W. A. W. A. Bakar, M. Y. M. Saman, Z. Abdullah, and T. J. J. T. Herawan, "Mining dense data: Association rule discovery on benchmark case study," vol. 78, no. 2-2, 2016.

[36] H. M. Sani, C. Lei, and D. Neagu, "Computational Complexity Analysis of Decision Tree Algorithms," in *International Conference on Innovative Techniques and Applications of Artificial Intelligence*, 2018, pp. 191-197: Springer.

[37] V. Bogina, T. Kuflik, and O. Mokryn, "Learning item temporal dynamics for predicting buying sessions," in *Proceedings of the 21st International Conference on Intelligent User Interfaces*, 2016, pp. 251-255: ACM.

[38] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[39] X. Niu, C. Li, and X. Yu, "Predictive analytics of E-commerce search behavior for conversion," 2017.

[40] G. Ke *et al.*, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems*, 2017, pp. 3146-3154.

[41] H. Zhang, S. Si, and C.-J. J. a. p. a. Hsieh, "GPU-acceleration for Large-scale Tree Boosting," 2017.

[42] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785-794: ACM.

[43] T. Chen, T. He, M. Benesty, V. Khotilovich, and Y. J. R. p. v.-. Tang, "Xgboost: extreme gradient boosting," pp. 1-4, 2015.

[44] Pereira, E., Light Gradient Boosting, 2007. [Online]. Available: https://www.kaggle.com/errolpereira/light-gradient-boosting

[45] Abhishek, Order prediction, 2007. [Online], Available: https://www.kaggle.com/abhisheks02/order-prediction

[46] Abhishek, Order prediction using XGBOOST, 2007. [Online], Available: https://www.kaggle.com/abhisheks02/order-prediction-using-xgboost