# Shopping Basket Analisys for Mining Equipment: Comparison and Evaluation of Modern Methods

Egor Nikitin
ITMO University
Saint-Petersburg, Russia
307646@niuitmo.ru

Alexey Kashevnik
SPC RAS, Saint-Petersburg, Russia
Petrozavodsk State University,
Petrozavodsk, Russia
{alexey.kashevnik}@iias.spb.su

Nikolay Shilov
SPC RAS
Saint-Petersburg, Russia
{nick}@iias.spb.su

***Abstract*—The problem of forecasting volatile sparse demand is rather different from forecasting mass sales. The paper is devoted to finding a solution for analyzing the dependencies between demands for various goods in a shopping basket for mining industry. Mining equipment is usually characterized by high price and stock costs, rare sales per item, and availability of little statistical data on the sales. As a result, high quality sales forecast could be very useful for companies selling this equipment. Association rules is one of the most promising approaches in this domain in terms of accuracy and running time. We consider three different algorithms for association rules search (Apriori, ECLAT, and FP-Growth) and evaluate those on the data of mining equipment sales dataset. We compare the methods and chose the best one for shopping basket analysis in the considered domain.**

## I. Introduction

The volumes of modern databases, which are quite impressive, have caused a steady demand for new scalable data analysis algorithms. Sales forecast is one of the application areas for these.

Most of the sales forecast approaches are aiming at demand forecast for mass sales, however, it is also extremely important to forecast very volatile sales (when a product is sold once in a few months) of expensive products. Equipment and spare parts for mining and metallurgy industries fall into this case. Some specifics of such sales are as follows:

- High stock costs per product. It is very expensive to store such products in a warehouse for a long time.
- High losses due to lost deals. Since the price of a product is high, losing even one deal due to the absence of the required product in a warehouse can be a significant loss for the company.
- Availability of little statistical data on the sales of such products.

Hence, the mining industry is characterized by a small number of sales of expensive goods.

One of the popular methods of knowledge discovery in data is the search for association rules. Association rules allow to find patterns between related events. An example of such a rule is the statement that a buyer who purchases "Bread" will also purchase "Milk" with a probability of 75%. In our case, we are investigating the relationship between products of different types for the mining industry [2].

Initially, the idea of searching for association rules appeared when answering the question: "Are there any typical patterns of shopping basket?". With the development of technology and an increase in consumer demand, it was necessary to find relationships between various products in large amounts of data.

The problem of forecasting volatile demand has arisen as a result of the rapid growth of engineering enterprises. Therefore, to simplify further forecasting, various methods for finding dependencies between sales of various goods were investigated.

The purpose of this work is to analyze various methods for searching for association rules of the mining and metallurgical industry equipment and choose the most suitable method. The article is a development of the previously published work of the authors [1]. The object of this study is anonymized data on sales of the above-mentioned products provided by a mining company. The relevance of the study lies in the novelty of comparing shopping basket analysis methods for goods with volatile demand.

The structure of the paper is as follows. In Section II we present the related work on the topic of using association rules. We presented data analysis and processing in Section III. The conclusion summarizes the paper.

## II. Related work

The search for association rules is widely used in various areas of human life, where it is necessary to search for dependencies. We analysed several papers on the related topic and summarized them in Table I. Since our goal of using association rules is to improve sales forecasting we have also included several articles related to different methods and approaches in sales forecasting with sparse demand.

As mentioned above, the search for association rules is used in various aspects. For example, in paper [3] association rules generated by using the Apriori algorithm provide power ramp direction maps for Spatio-Temporal analysis. In combination with K-means clustering it is used as a fast and effective decision-making tool with qualitative results for the system operator with minimal expert knowledge.

ISSN 2305-7254

TABLE I. SCIENTIFIC PAPERS

| Name | Algorithm/Method | Applying area | Results |
|---|---|---|---|
| Apriori and K-Means algorithms of machine learning for spatio-temporal solar generation balancing [3] | Apiori | Solar powered devices | New power ramp direction maps for Spatio-Temporal analysis |
| A heuristic approach for load balancing the FP-growth algorithm on MapReduce [4] | FP-Growth | Big Data | Improving the algorithm by solving its problems with performance and scalability |
| Factors influencing the patterns of wrong-way driving crashes on freeway exit ramps and median crossovers: Exploration using 'Eclat' association rules to promote safety [5] | ECLAT | Road traffic | The findings can provide an opportunity for departments of transportation to develop safety strategies and engineering solutions to tackle the issues associated with crashes |
| FR-Tree: A novel rare association rule for big data problem [6] | Based of FP-Growth | Big Data | Creation of a new algorithm called FR-Tree |
| A Hierarchical Energy Conservation Framework (HECF) of Wireless Sensor Networks by Temporal Association Rule Mining for Smart Buildings [7] | Apriori | Electricity | Creation of a new energy-saving system (HECF) which achieves 16% better energy conservation |
| A compound-Poisson Bayesian approach for spare parts inventory forecasting [8] | Compound Poisson Bayesian (CPB) method, Smoothed Mean Absolute Deviation (MAD) | Big Data | A new Bayesian method based on composite Poisson distributions, negative binomial distribution (NBD) |
| The development of a hierarchical forecasting method for predicting spare parts demand in the South Korean Navy [9] | SVM, Logistic regression, Poisson model, Croston methods | Military | Adaptation of the support vector machine (SVM) to predict the occurrence of non-zero demand for spare parts and the development of a hybrid mechanism for integrating the results of the SVM forecast and the relationship of the occurrence of non-zero demand with independent variables. |
| An improved method for forecasting spare parts demand using extreme value theory [10] | Croston method, WSS | Big Data | Extreme value theory (the behavior of many uncertain quantities encountered in practice can be modeled using the Generalized Pareto Distribution (GPD)) |
| A spare parts inventory control model based on Prognostics and Health monitoring data under a fill rate constraint [11] | Gamma distributions for simulating magnification degradation rate during each period | Medicine | PHM (Prognostics and Health Monitoring) system monitors the level of component degradation and provides at the beginning of each period a forecast for the RUL of each component. |
| A product-centric data mining algorithm for targeted promotions [16] | Algorithm which includes the market target (mt) model and the FCM clustering approach | Mass Marketing | An algorithm and model that makes it easy to make marketing decisions that do not only limit marketing spending, but also allow for expansion of the customer base and avoid unnecessary spending. |

As mentioned above, the search for association rules is used in various aspects. For example, in paper [3] association rules generated by using the Apriori algorithm provide power ramp direction maps for Spatio-Temporal analysis. In combination with K-means clustering it is used as a fast and effective decision-making tool with qualitative results for the system operator with minimal expert knowledge.

The authors of paper [4] offer to improve the FP-Growth algorithm using heuristic load balancing strategy, which will be particularly useful for sparse datasets, that distributes the load on all cluster nodes in such way that the load on the nodes is more or less the same for the FP-growth's parallel implementation.

In paper [5] authors use association rule search algorithms and machine learning algorithms to determine various factors that lead to road accidents.

Paper [6] includes a study devoted to the creation of a new algorithm based on FP-Growth algorithm. This algorithm aims to extract rare association rules with a high degree of certainty.

Authors of paper [7] explore the application of association rules in the field of electricity. As a result, the authors proposed a novel Hierarchical Energy Conservation Framework (HECF) which aims to conserver energy at each layer of a network by using the hierarchical temporal association rule mining in multistory buildings. The study is based on the Apriori algorithm.

In paper [8] a new Bayesian method based on composite Poisson distributions is proposed. The proposed method is compared to the Poisson-based Bayesian method with a Gamma prior distribution as well as to a parametric frequentist method and to a non-parametric one. Prediction accuracy of the constructed model 80 – 88%.

The aim of paper [9] is to establish an appropriate forecasting strategy for predicting the demand for consumable spare parts in the South Korean Navy. The authors develop several forecasting models with an accuracy of 70 to 90%, depending on the chosen method.

In paper [10] authors tried to improve the empirical method by applying extreme value theory to model the tail of the demand distribution at the lead time, comparing it with the WSS, Croston, and SBA methods for a range of demand distributions and got an accuracy of 82%.

The paper [11] includes methods and models such as gamma distributions for simulating magnification degradation rate during each period and Prognostics and Health Monitoring (PHM) system. This paper aims at presenting a

novel spare parts inventory control model for non-repairable items with periodic review. Authors declare the accuracy of the model to be 84%.

The Research in the paper [16] aims to address some of the shortcomings such as “false positives” and contributes to a growing body of research as it presents a new algorithm and mathematical model to help retailers improve both product mix and customer focus for marketing promotions for additional purchases and promotions.

Based on the presented articles we decided to use three algorithms for searching for association rules: Apriori, ECLAT and FP-Growth. These algorithms can be considered as “classical”. They are the most common and are the basis for other algorithms.

## III. Data analysis and processing

**Association rules.** Association rules are built based on the use of transaction. Each transaction is a set of goods purchased by the buyer in one visit. Such a transaction is also called a market basket. The purpose of using association rules is to establish the following dependencies: if a certain set of X elements is encountered in a transaction, then based on this we can conclude that another set of Y elements must also appear in this transaction. Establishing such dependencies enables us to find very simple and intuitive rules [12].

We also introduce a few key concepts that will be used during the experiments. The first one is *Definition Support*. This is an indicator of the "frequency" of this item set in all analyzed transactions.

$$supp(x_1 \cup x_2) = \frac{\sigma(x_1 \cup x_2)}{|T|},$$

where $\sigma$ is the number of transactions containing $x_1, x_2$ and T is the number of transactions. The next key concept is *Confidence*. This is an indicator of how often our rule is triggered for the entire dataset.

$$conf(x_1 \cup x_2) = \frac{supp(x_1 \cup x_2)}{supp(x_1)}.$$

The third concept is *Lift* indicating the ratio of "dependency" of items to their "independence". Lift shows how items depend on each other.

$$lift(x_1 \cup x_2) = \frac{supp(x_1 \cup x_2)}{supp(x_1) \times supp(x_2)}$$

The last key concept is *Conviction*. In general, conviction is the “error rate” of our rule. The higher the result of the formula above 1, the better.

$$\text{conv}(x_1 \cup x_2) = \frac{1 - supp(x_2)}{1 - conf(x_1 \cup x_2)}$$

**Data Preparation.** To prepare the data for further analysis we analyzed spare parts sales data from the mining company in a period (see Tab. 2). The data includes deal number (Deal ID), product type (Type), deal status (Stage), customer number (Company ID), source (Source), opening and closing dates (estimated closing) of the deal (Created and Assumed close) date respectively), product (Element ID), the quantity of goods sold (Q-ty) and unit of measure (UOM).

Further work was carried out with the provided data: cleaning, sorting, filtering, etc. For the convenience of the primary analysis, a pivot table was built, in which only those goods were taken into account, for which deals have already taken place. We did not sort the goods by their units of measurement in any way so that further analysis would be more truthful (see Fig. 2).

Then, only those products were selected for which there is a minimum sales statistic (more than ten units were sold). However, we have also removed anomalous outliers so that they do not affect further calculations.

We proposed the following workflow to analyze the data: (1) reading data from a file, (2) data filtering, (3) use of various algorithms for association rule search, (4) analyzing results. The workflow have been implemented a program code for "Jupyter notebook" and "Google Colab".

We started working with data by filtering them, since we only need completed transactions or transactions close to completion (paid, but not sent, paid, but not signed a contract, etc.), for more details see Fig. 3.

After preparing the data, the three most popular algorithms for searching for association rules have been evaluated and compared: Apriori Algorithm, ECLAT Algorithm, and FP-Growth Algorithm.

**Apriori Algorithm.** The algorithm is based on the concept of *Support*. Support is the number of transactions in which a particular product (or combination of products) occurs. The first step of the algorithm is to calculate the Support for each individual element. It basically boils down to counting for each product in how many transactions it occurs [13].

Once the support for each of the individual products has been calculated, it must be used to filter out some of the products that are not common. To do this, one needs to independently choose the appropriate support threshold by analyzing your data set and determining the "scatter" of support values for each product.

The next step is to run the same analysis, but now using product pairs instead of individual products. The distinctive feature of the Apriori algorithm is that all pairs containing any infrequent elements will be ignored. Because of this, we have far fewer pairs of elements to scan.

The next step is to convert the elements into association rules. Association rules go one step further than just listing products that are often found together. The algorithm uses the following statement: $if\ X \subseteq Y\ then\ supp(X) \geq supp(Y)$. From this follows the following 2 properties:

- If Y occurs frequently, then any subset of $X : X \subseteq Y$ also occurs frequently.
- If X is rare, then any superset $Y : Y \supseteq X$ is also rare.

TABLE II. SALES DATE FROM THE MINING COMPANY

| Deal ID | Type | Stage | Company ID | Source | Assumed close date | date | Element ID | Q-ty | UOM |
|---|---|---|---|---|---|---|---|---|---|
| 4901 | Goods | Price indication | 4 | E-parts | 30.06.2020 | 26.05.2020 | 3010008941 | 1 | PCE |
| 4901 | Goods | Price indication | 4 | E-parts | 30.06.2020 | 26.05.2020 | 3020000078 | 1 | PCE |
| 4901 | Goods | Price indication | 4 | E-parts | 30.06.2020 | 26.05.2020 | 3010001040 | 1 | PCE |
| 4901 | Goods | Price indication | 4 | E-parts | 30.06.2020 | 26.05.2020 | 7060000193 | 1 | PCE |
| 4901 | Goods | Price indication | 4 | E-parts | 30.06.2020 | 26.05.2020 | 3010000181 | 1 | PCE |
| 4901 | Goods | Price indication | 4 | E-parts | 30.06.2020 | 26.05.2020 | 3020000077 | 1 | PCE |
| 4901 | Goods | Price indication | 4 | E-parts | 30.06.2020 | 26.05.2020 | 3010000180 | 1 | PCE |
| 4901 | Goods | Price indication | 4 | E-parts | 30.06.2020 | 26.05.2020 | 3020000079 | 1 | PCE |
| 4901 | Goods | Price indication | 4 | E-parts | 30.06.2020 | 26.05.2020 | 7060000192 | 1 | PCE |
| 4901 | Goods | Price indication | 4 | E-parts | 30.06.2020 | 26.05.2020 | 7060000722 | 1 | PCE |
| 4901 | Goods | Price indication | 4 | E-parts | 30.06.2020 | 26.05.2020 | 3010005315 | 1 | PCE |
| 4901 | Goods | Price indication | 4 | E-parts | 30.06.2020 | 26.05.2020 | 3010003101 | 1 | PCE |
| 4901 | Goods | Price indication | 4 | E-parts | 30.06.2020 | 26.05.2020 | 7010000574 | 1 | PCE |
| 7024 | Goods | Negotiations | 494 | E-parts | 31.12.2020 | 30.10.2020 | 4020002938 | 785 | LINM |
| 4901 | Goods | Price indication | 4 | E-parts | 30.06.2020 | 26.05.2020 | 7010000921 | 1 | PCE |
| 4901 | Goods | Price indication | 4 | E-parts | 30.06.2020 | 26.05.2020 | 7010000379 | 1 | PCE |
| 4901 | Goods | Price indication | 4 | E-parts | 30.06.2020 | 26.05.2020 | 7010002731 | 1 | PCE |

| | ⊞2018 | ⊟2019 | | | | | | | | | | | ⊟2020 | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Detail ID | | Jan | Feb | Mar | Apr | May | June | July | Aug | Sept | Oct | Nov | Dec | Jan | Feb | Mar | Apr | May | June | July | Aug | Sept | Oct | Nov |
| 1010000044 | | | | | | 1 | | | | | | | | | | | | | | | | | | |
| 1010000047 | | | | | | | | | | | | | | 1 | | | | | | | | | | |
| 1010000057 | | | | | | | | | | | | | | 1 | | | | | | | | | | |
| 1010000058 | | | | | | | | | | | | | | 1 | | | | | | | | | | |
| 1010000059 | | | | | | | | 1 | | 1 | 1 | | | 2 | 1 | 1 | | 1 | | | 2 | 1 | 1 | |
| 1010000060 | | | | | | | | | | | 1 | | | 2 | 1 | 1 | | 1 | | | 1 | 1 | 2 | |
| 1010000061 | | | | 1 | | | | 2 | 1 | 1 | 1 | | 1 | 2 | 1 | | | 1 | | | | 1 | 2 | |
| 1010000062 | | | | | | | | 2 | 1 | 2 | 1 | | | 2 | 2 | | 1 | 2 | | | | 1 | 3 | |
| 1010000063 | | | | | | | | 1 | | | 1 | | 1 | 3 | 1 | | 1 | 1 | | | | 1 | 2 | |
| 1010000064 | | | 1 | 1 | | 1 | | 3 | | 2 | | | | 3 | | 1 | 2 | | | 2 | | | 3 | |
| 1010000065 | | | | | | 1 | | | | 2 | 1 | 1 | | 4 | | 1 | 1 | | | | | | 2 | |
| 1010000066 | | 1 | | | 1 | | 2 | 1 | 2 | 1 | | | 1 | 2 | | | 1 | | 1 | | 1 | | 2 | |
| 1010000067 | | 2 | | | 1 | | 2 | 2 | 2 | 3 | 1 | 1 | 2 | | | | | | 4 | | 1 | 8 | 9 | |
| 1010000068 | | | | | | 1 | 3 | | 1 | 1 | | | | 4 | | | | | | | 1 | | 1 | |
| 1010000069 | | 3 | 1 | 2 | 2 | 1 | 5 | 5 | 5 | 3 | 2 | | 2 | 4 | | | 2 | | | 1 | 1 | 1 | 3 | |
| 1010000070 | | | | | | 1 | | 3 | 1 | 1 | | 1 | | 4 | | | 1 | | | | 1 | 1 | 4 | |
| 1010000071 | | | | | 1 | | 2 | 1 | | 1 | 1 | | | 1 | | | | | | | | | | |
| 1010000072 | | | 1 | | | 1 | 1 | 2 | | | | | | | 1 | | | | | | | 6 | 8 | |
| 1010000073 | | | | | | 1 | | 2 | | | | 1 | | 1 | | | | | | | | | 2 | |
| 1010000074 | | | | | | | | 1 | | | | | | 1 | | 1 | 1 | 1 | | | | | | |
| 1010000075 | | | | | | 1 | | 1 | | 2 | | | 1 | | 1 | | | 1 | | | | 1 | 1 | |
| 1010000076 | | 1 | | | | 1 | | 1 | | | | 1 | | 3 | | 1 | 2 | | | | | | 1 | |
| 1010000077 | | 1 | | | | | 2 | 1 | 1 | | 1 | | 2 | 1 | | | | | | | | | | |
| 1010000078 | | | | | | 1 | 1 | 1 | 1 | | | | | 1 | | 1 | 1 | | | 1 | | | 2 | |

Fig.2. Pivot table example

```
prep_table = data.loc[((data['Stage'] == stages[4]) | (data['Stage'] == stages[7]) | (data['Stage'] == stages[8]))]
[['Type', 'Stage', 'date', 'Element ID', 'Q-ty', 'UOM']]

prep_table
```

| | Deal ID | Type | Stage | Company ID | Source | Assumed close date | date | Element ID | Q-ty | UOM |
|---|---|---|---|---|---|---|---|---|---|---|
| **60** | 3258 | Goods | Won | 12 | - | 2019-12-18 | 2019-12-17 11:04:30 | 7020000009 | 1 | шт |
| **64** | 2712 | Goods | Won | 12 | - | 2019-09-30 | 2019-09-27 17:17:41 | 1010000107 | 1 | шт |
| **65** | 2712 | Goods | Won | 12 | - | 2019-09-30 | 2019-09-27 17:17:41 | 1010000106 | 1 | шт |
| **79** | 1535 | Goods | Won | 12 | - | 2019-03-20 | 2019-03-12 17:37:06 | 1010000262 | 6 | pcs |
| **80** | 1535 | Goods | Won | 12 | - | 2019-03-20 | 2019-03-12 17:37:06 | 7040000015 | 1 | pcs |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **50898** | 6963 | Goods | Quote approval | 6478 | Own initiative | 2020-12-28 | 2020-10-28 14:54:55 | 2010001339 | 1 | PCE |
| **50899** | 6963 | Goods | Quote approval | 6478 | Own initiative | 2020-12-28 | 2020-10-28 14:54:55 | 1010000366 | 1 | PCE |
| **50927** | 1170 | Goods | Won | 20 | - | 2018-11-07 | 2018-11-06 12:33:27 | 2010000399 | 5 | Set |
| **50998** | 1216 | Goods | Won | - | - | 2018-11-27 | 2018-11-27 11:17:53 | 1010000480 | 1 | pcs. |
| **50999** | 1216 | Goods | Won | - | - | 2018-11-27 | 2018-11-27 11:17:53 | 1010003443 | 60 | pcs. |

5244 rows × 10 columns

Fig. 3. Filtering data by specified criteria

TABLE III. RESULTS OF THE APRIORI ALGORITHM

| items | ordered_statistics | | | |
|---|---|---|---|---|
| 2010000390<br>2010000397<br>2010000403 | items_base | items_add | confidence | lift |
| | 2010000397 | 2010000390, 2010000403 | 1.0 | 112.199 |
| | 2010000390, 2010000397 | 2010000403 | 1.0 | 112.199 |
| | 2010000397, 2010000403 | 2010000390 | 1.0 | 53.428 |
| Support: 0.0062 | | | | |

The Apriori algorithm traverses the prefix tree on each tree and calculates the frequency of occurrence of subsets X in dataset. Thus, according to the algorithm:

- Rare subsets and all their supersets are excluded.
- Calculate supp(X) for each eligible candidate X of size k at level k.

This principle formed the basis for writing an algorithm that gave the following results. Thresholds for key parameters have been set:

- Support – 0.005.
- Confidence – 0.8.
- Lift – 40.
- Length – 2 (minimum possible value, which means that we are also interested in pairs of goods).

We chose these threshold values experimentally to balance between the algorithm running time and accuracy. Based on the set values, more than 14 thousand rules were obtained. Table II shows what the rule for three items looks like.

In the presented results (Table I) we see that the item column contains all the items that participate in the rule. The “items_base” and “items_add” columns show which item(s) are compared in different sequences. Columns “confidence” and “lift” represent confidence and lift respectively. The value of the parameter “support” is the same for all rules by definition. It is interesting that the lift parameter has values from 53 to 112, which show how dependent the products are on each other. The higher the lift value the better (the higher the dependence). The confidence parameter shows how often the given rule is triggered in the entire dataset. In this case, we see the strongest possible dependence of one product on another.

**ECLAT Algorithm.** There are two faster alternatives to the Apriori algorithm: one is FP Growth (this algorithm will be discussed below), and the other is ECLAT (Equivalence CLAss Transformation). There is no clear winner between FP Growth and ECLAT in terms of execution time: it will depend on different data and different algorithm settings [14].

ECLAT, unlike other algorithms, does not provide the confidence and lift indicators that are needed for interpretation in alternative models. This allows the model to be faster because the user has a choice between speed and having more metrics.

The first step is to create a list containing a list of transaction IDs for each product, in which the product occurs. The next step is to choose a value called *Minimum support*. The minimum support will serve to filter out products that don't occur often enough to be considered.

Next, we will repeat the same operation as in step 1, but now for product pairs. The interesting notice about the ECLAT algorithm is that this step is performed using the intersection of the two input sets. This distinguishes it from the Apriori algorithm. The ECLAT algorithm is faster because it is much easier to determine the intersection of a set of transaction IDs than to scan each individual transaction for product pairs (as the Apriori algorithm does).

Then, just like before, we filter out the results that don't reach the minimum support. Further, all steps are repeated until it becomes possible to create new pairs above the threshold support level.

So, as mentioned before the idea behind the ECLAT algorithm is to speed up the calculation of supp(X). To do this, we need to index our database D so that it allows us to quickly calculate supp(X).

It is easy to see that if t(X) denotes the set of all transactions where a subset of X occurs, then $\mathrm{t(XY)} = t(X) \cup t(Y)$ and $supp(XY) = |t(XY)|$ that is, supp(XY) is equal to the size of the set t(XY). Unlike the Apriori algorithm, ECLAT performs a depth-first search. In this regard it sometimes is called "vertical".

When developing the algorithm, we take into account only the minimum support (0.005), the maximum (19) and minimum (2) number of elements in any of the transactions. There were several records in our data set that have a "strong outlier" in the number of items in a transaction (151, 92, 80, etc.). We chose the parameters empirically to balance between running time and accuracy. The results of the algorithm look like this (Table IV). Since this algorithm provides the least number of statistics, we can see that of all the parameters, only support is presented here. However, in this case, it is calculated for each combination of goods in one rule.

TABLE IV. RESULTS OF THE ECLAT ALGORITHM

| Items | Support |
|---|---|
| 2010000390, 2010000403, 2010000397 | 1.0340% |
| 2010000403, 2010000398, 2010000397 | 0.8863% |
| 2010000399, 2010000397, 2010000401 | 0.7386% |

**FP-Growth Algorithm.** FP-Growth proposes to abandon the generation of candidates, which is the basis of Apriori and ECLAT algorithms. Theoretically, this approach will further increase the speed of the algorithm and use even less memory [15].

This is achieved by storing a prefix tree in the memory not from combinations of candidates, but from the transactions themselves.

In this case, FP-Growth generates a header table for each item whose support is higher than the user-specified one. This header table keeps a linked list of all nodes of the same type in the prefix tree. Thus, the algorithm combines the advantages of BFS due to the header table and DFS due to building a prefix tree.

So, the idea behind the FP Growth Algorithm is to find frequently occurring sets of elements in a dataset, being faster than the Apriori algorithm. The Apriori algorithm basically accesses the dataset to check if the products in the dataset match.

To be faster, the FP algorithm changes the organization of the data to a tree instead of sets. This tree-like data structure allows for faster scanning, and this is where the algorithm wins time. The result of our FP-Growth Algorithm is shown below (Table V).

As one can see in the table, the results and parameters themselves are very similar to the results and parameters of another algorithm. Confidence is also a 1, which once again confirms the strong dependence between the goods in the rule. The lift parameter takes values of 37.6 - 75.22, which is lower than in the results of the Apriori algorithm, but still high enough to argue for a high dependence between products

Having applied all three algorithms for our dataset we can conclude that the fastest was the ECLAT Algorithm (2-3 minutes versus 5-6 FP-Growth and 30 minutes for Apriori Algorithm), which confirms the main idea of this algorithm. However, if the thresholds are reduced (support), the algorithm may take more than six hours to run, which means that a thorough analysis of key parameters is necessary.

The Apriori algorithm from the point of view of implementation is the simplest, but the average time of its work is about half an hour.

As already described above, the idea of the FP-Growth algorithm is to obtain association rules, being faster than the Apriori algorithm, which it succeeds in doing. However, due to its specific approach, this algorithm presents the greatest difficulty in implementing and interpreting the results.

TABLE V. RESULTS OF THE FP-GROWTH ALGORITHM

| antecedents | consequents | support | conf | lift |
|---|---|---|---|---|
| 2010000390, 2010000397 | 2010000403 | 0,010 | 1 | 75,22 |
| 2010000397 | 2010000390, 2010000403 | 0,010 | 1 | 75,22 |
| 2010000403, 2010000397 | 2010000390 | 0,010 | 1 | 37,6 |

However, we observe for the same associative pairs similar values of support level for the two algorithms: ECLAT and FP-Growth. There is also a clear convergence in the values of the confidence parameter for the algorithms Apriori and FP-Growth. In terms of speed, the Apriori algorithm works on the same data for half an hour, while the other two spend time from 3 (FP-Growth) to 10 (ECLAT) minutes. As a result, the FP-Growth algorithm looks the most attractive for further work.

Even though we have chosen the FP-Growth algorithm as the most promising and attractive, it is still necessary to take into account the results of the other two algorithms in order to get a more complete picture of the shopping basket.

## IV. CONCLUSION

In the case of mining equipment sales (when the sales are rare) most of the available forecast tools are not suitable for analysis, since there is very little information available. Besides, this data has to be filtered, for a more descriptive, representative sample, without any errors, such as an empty customer field, an incorrect date, etc., what even worsen the situation.

Association rules is one of the efficient mechanisms to deal with this. The paper considers three association rules constructing algorithms on the example of sales data for mining equipment, namely: Apriori, ECLAT, and FP-Growth. It was shown that the results of the FP-Growth and Apriori algorithms mostly correlate, however the speed of their operation is different. The ECLAT and FP-Growth algorithms overperform the Apriori algorithm for the price of producing less metrics for the result evaluation. However, the speed of the ECLAT algorithm is significantly affected by the choice of thresholds values. Based on the experiments results the FP-Growth algorithm was selected as the most efficient one for the considered task.

In the future, it is planned to continue the study for a better and more complete assessment of the shopping basket.

## ACKNOWLEDGMENT

Research is due to Russian State Research, project number FFZF-2022-0005.

## REFERENCES

[1] E.D. Nikitin, A.M. Kashevnik, N.G. Shilov, “Spare Parts Sales Forecasting for Mining Equipment: Methods Analysis and Evaluation”, *ISDA 2021*

[2] Loginom, Introduction to Association Rule Analysis, Web: https://loginom.ru/blog/associative-rules.

[3] N.Y. Yürüşen, B. Uzunoğlu A.P. Talayero, A. L. Estopiñán “Apriori and K-Means algorithms of machine learning for spatio-temporal solar generation balancing”, *Renewable Energy,* vol. 175, 2021, pp. 702 – 717.

[4] S. Bagui, K. Devulapalli, J. Coffey, “A heuristic approach for load balancing the FP-growth algorithm on MapReduce”, *Array*, vol. 7, 2020, 10035.

[5] S. Das, A. Dutta, M. Jalayer, A. Bibeka, L. Wu, “Factors influencing the patterns of wrong-way driving crashes on freeway exit ramps and median crossovers: Exploration using ‘Eclat’ association rules to promote safety”, *International Journal of Transportation Science and Technology*, vol. 7, 2018, pp. 114 – 123.

[6] M.A. Mahdi, K. M.Hosny, I. Elhenawy, “FR-Tree: A novel rare association rule for big data problem”, *Expert Systems with Applications*, vol. 187, 2022, 115898
[7] F. S. Ujager, A. Mahmood, M. Usman, M. S. Rathore, “A Hierarchical Energy Conservation Framework (HECF) of Wireless Sensor Networks by Temporal Association Rule Mining for Smart Buildings”, *Egyptian Informatics Journal*, 2021
[8] M.Z. Babai, H. Chen, A.A. Syntetos, D. Lengu, “A compound-Poisson Bayesian approach for spare parts inventory forecasting”, *International Journal of Production Economics*, 2021, 108954.
[9] S. Moon, C. Hicks, A. Simpson, “The development of a hierarchical forecasting method for predicting spare parts demand in the South Korean Navy—A case study”, *International Journal of Production Economics*, vol.140, 2012, pp. 794-802.
[10] S. Zhu, R. Dekker, W.V. Jaarsveld, R.W. Renjie, A.J. Koning, “An improved method for forecasting spare parts demand using extreme value theory”, *European Journal of Operational Research*, vol.261, 2017, pp. 169-181.
[11] L.R. Rodrigues, T. Yoneyama, “A spare parts inventory control model based on Prognostics and Health monitoring data under a fill rate constraint”, *Computers & Industrial Engineering*, vol.148, 2020, 106724.
[12] Harb. Associative rules, or beer with diapers, Web: https://habr.com/ru/company/ods/blog/353502
[13] Towards Data Science. The Apriori algorithm Web: https://towardsdatascience.com/the-apriori-algorithm-5da3db9aea95
[14] Towards Data Science. The Eclat algorithm Web: https://towardsdatascience.com/the-eclat-algorithm-8ae3276d2d17
[15] Towards Data Science. The FP Growth algorithm, Web: https://towardsdatascience.com/the-fp-growth-algorithm-1ffa20e839b8
*[16]* R. Moodley, F. Chiclana, F. Caraffini, J. Carter, “A product-centric data mining algorithm for targeted promotions”, *Journal of Retailing and Consumer Services,* vol.54, 2020, 101940