Predviđanje prodaje na platformi e-trgovine, korišćenjem data mining modela

Stefana Janićević^{1*}, Đorđe Petrović², Miodrag Stefanović³

¹ITS Information Tehnology School, Beograd, stefana@turing.mi.sanu.ac.rs

²Western Serbia Academy of Applied Studies, djordje.petrovic@akademijazs.edu.rs

³Zea Stim Research & Development, miodrag@zeastim.com

Apstrakt: U ovom radu primenjen je algoritam za prodaju twinning proizvoda sa e-commerce web platforme. Kako bi se utvrdili relativno homogene grupe proizvoda u trgovinskom lancu na internetu tokom prethodne godine potrebno je bilo formirati prediktivni matematički model. Nakon izdvajanja objekata i određenih atributa iz MySQL baze podataka, određivanja skupa relevantnih varijabli koje će reprezentovati obeležja grupe, primenjeni su K-means algoritam u Python programskom okruženju, Market Basket model i Vector Distance model. Na osnovu analize izvornih i veštačkih promenljivih, predložen je broj klastera, koji je tokom izvršavanja algoritma fiksiran, a u cilju detekcije razdvojenosti i kompaktnosti klastera, korišćen je Silhouette indeks. Na osnovu podele po klasterima, urađeni su modeli koji predviđaju slične proizvode i analizirala se verovatnoća kupovine. Dobijeni rezultati mogu se koristiti u smislu planiranja prodajnih kampanja, optimizacije troškova marketinga, predlaganja novih programa lojalnosti, kao i boljeg razumevanja ponašanja potrošača sa ciljem zadržavanja postojećih i povećanja broja novih kupaca. Rezultati su izdvojeni u dve grupe - predloge koje treba ponuditi kupcima i predloge koje treba ponuditi prodavcima.

Ključne reči: Klaster analiza; PCA; Market Basket analiza; Vector Distance model; segmentacija tržišta

Sales prediction on e-commerce platform, by using data mining model

Abstract: In this paper we applied twinning algorithm for product that are sold via e-commerce platform. To establish relatively homogenous product groups that were on sale on this e-commerce platform during the last year, it was necessary to form predictive mathematical model. We determined set of relevant variables that will represent group attributes, and we applied K-means algorithm, Market Basket model and Vector Distance model. Based on analysis of basic and derived variables, fixed number of clusters was introduced. Silhouette index was used for the purposes of detecting whether these clusters are compact. Using these cluster separations, we created models that detect similar products, and try to analyze probability of sales for each product. Analysis results can be used for planning future sales campaigns, marketing expenses optimization, creation of new loyalty programs, and better understanding customer behavior in general.

Keywords: Cluster analysis; PCA; Market Basket analysis; Vector Distance model; marketplace segmentation

1. Introduction

Motive for research that is described in this paper is improving sale through rate of "Limundo/Kupindo" online e-commerce platform, which offers both buy-now marketplace and auction model for product buying/selling. Said improvements should be achieved by researching through the data on buyer/seller behavior collected over the course of several years, and by implementing advanced technologies and algorithms from this area.

"Limundo" and "Kupindo" are established in the year 2006, they currently have over million registered and verified customers, they have more than two million products on "Kupindo" marketplace and additional one million auctions that are initiated on Limundo auction platform every month. Around

two hundred thousand people visit these two platforms every day.

Based on analysis of selected product attributes from MySQL database of these e-commerce platforms, we proposed product grouping in nine clusters. Products in this clusters have common characteristics, and they evoke similar customer behavior and responses (Cheng and Chen 2009), (McDaniel and Gates, 2014).

1.1. Research background

Paper (Ismail et al., 2015) showcases procedure of data extraction from the e-commerce databases, including three usual algorithms: association, clustering and prediction. Authors highlight advantages of e-commerce related datamining in the sense of merchandise planning, sales prediction, shopping carts analysis, customer relations management and market segmentation, which can be achieved with three datamining algorithms. The paper also highlights challenges that can be faced during datamining, such as crawler identification and data transformation.

Paper (Rachid, 2015) points out that companies must have good insight into their data to be able to make the right decision and keep up with the competition and fulfill their customer needs. Author presents case study of LRFM model (length, recency, frequency, monetary) and clustering technique in the area of e-commerce, with the goal of evaluating customer value on the Moroccan ecommerce websites. Two-stage clustering method was implemented, which generated nine clusters based on L, R, F and M values. Bejjou (2016) points out evaluation and improvement suggestions for traditional price strategies by using datamining techniques from e-commerce websites. Suggested strategy can be implemented by optimizing decision trees in iterative process and by using historical data on buyers' transactions. Paper (Nur and Ayvaz, 2018) describes analysis of online shopping baskets data acquired from big hardware retail company. Apriori and FP-Growth algorithm were used. Data were split into two sets, correctness of rules extracted from the first set was compared to rules extracted from the second set. Kubiak and Weichbroth (2010) presented cross-selling and upselling marketing techniques in e-commerce.

Their goal was to increase the value of single sale transaction and trust, and to lessen risk of losing customer to competition. Paper presents iterative model of customer service, which demonstrates separate roles of cross-selling and upselling techniques in processes of automatization integration of marketing in the context of customer relationship management.

1.2. Data description

MySQL (Percona fork) is being used as relational database management system. It is hosted on four physical server – two slaves and two masters in multimaster replication. Software coding was done in R programming language in R Studio, and in Python programming language in Jupyter environment.

Following tables were processed:

- Table *AuctionItems*, which contains detailed data on all auction items listed after 25.07.2018.
- Table *AuctionHistory*, which contains history of product flow (listing, relisting, selling, cancelling)
- Table *BidProxy*, which contains history of buyer bids for every product
- Table *FieldSchemaSelected*, which contains systematized data on each auction item characteristics (color, size, manufacturer, etc.)

Data from these starting tables were merged into new table *ItemBidFieldAuction* by using ETL process. *ItemBidFieldAuction* contains columns: ID, AuctionItemID, CategoryID, SubCategoryID, SubCategoryID2, AuctionCloseDate, BidHighPrice, NumberOfBids, SelingPrice, FieldSchemaValueID, Value, FieldSpeciesID, StartingPrice, Sold, TopCategorySelected, PurchasedImmediately. Some of the source tables contained historical data, so maximization and aggregation of data was necessary in those cases.

After working table *ItemBidFieldAuction* was formed, it was necessary to perform preprocessing which consists of scrubbing, auditing, wrangling, munging, outlier detection, extreme values detection, data limits detection and data validation. Applied techniques resulted in duplicate purging, invalid data elimination, and incomplete data detection. Also, database stability, data validity and data quality were

checked on regular basis. At one stage, database was reformatted by creating procedures for complex data structures. At some places damaged data was repaired, to ensure richer database content.

Some of the aforementioned variables were artificially produced through IF-THEN-ELSE recoding, and some were aggregated from source tables. Most of the variables were used directly from source tables. *CategoryRootID* is variable that was created empirically and it can have nominal value between 1 and 9. This value is generated by recoding ID values from source table. ID value represents code for basic category of product (clothing, footwear, furniture, machines, electronics etc.). Since there are more than 40 different codes, we aggregated them to total of 9 target codes by setting the same *CategoryRootID* for similar product types.

After preprocessing the data from the final database, modelling table had more than 500.000 records. Final table contains selection of following variables: *AuctionItemID*, *BidHighPrice*, *NumberOfBids*, *StartingPrice*, *CategoryRootID*, *ParentSequenceID*, *SetOfValues*, *Visits*.

Database is represented through a collection of objects (records) defined as $D = \{R_1, ..., R_n\}$ where each record is defined as $R_i = \{d_1, ..., d_n\}$, d_i are variables/features. Each record is a product that is represented through its characteristics, mathematically it is a tuple $d_i = (id(i), maxbid(i), numberbid(i), startprice(i), idroot(i), parent(i), values(i), visits(i))$ where vector coordinates are aforementioned variables.

First algorithm that was implemented in the model is cluster analysis, which finds groups of objects where objects in the same group are similar or connected, while objects in different groups are disparate or unconnected. Cluster analysis methods are divided into two basic groups: agglomerative methods and divisive methods (Hastie et al., 2009), (Anderson, 2003). Cluster analysis methods are also divided into: hierarchical methods; which are usually presented by special diagram – dendrogram, and nonhierarchical methods (Bishop, 2006), (Jain, 2010).

In this paper we applied nonhierarchical partitional clustering n observation in k clusters, with the goal of dividing data sets (products) into groups with the nearest midpoint. Cluster centers are used for data model-ling, but k-means clustering tends to find clusters of similar spatial size, while expectation maximization mechanism allows clusters to have different shapes. Nonhierarchical method generates classification by partitioning data set, producing set of non-overlapping groups that have no hierarchical relationships between them. In this paper non-overlapping subsets are constructed in such way that each data is contained in exactly one subset. Suggested clusters are based on center, which means that cluster represents set of objects where any object in the given cluster is closer (or more similar) to that clusters center than to any of the other clusters center. Cluster center is usually centroid (average of all points in the cluster) or medoid (the most representative point in the cluster), often starting centroid is chosen in a random way (Celebi et al., 2013), (Stefanović et al., 2018), while number of clusters are defined before running the algorithm execution (Nayak et al., 2016), (Wang et al., 2014).

During the partitional clustering implementation, we used model with the prototype, where starting centroids were chosen randomly, and each data from the set was assigned to cluster with the closest centroid. In each iteration we recalculate centroids for all clusters. Number of clusters was set to fixed value, which does not change during the algorithm execution, while data are iteratively sorted with the goal of achieving maximal cluster homogenization. As a measure for similarity we used Euclidean distance, and as a measure of separation (cluster compactness) we used Silhouette index (The MathWorks, Inc., 2020).

Second analysis we used is Market Basket analysis, this algorithm is used in sales to detect connections between products. Algorithm searches through combination of products which are frequently appear together in transactions Basis for this algorithm contains rules of grouping to explore connections and rules that appear in transactions. Frequencies are statistical parameter from which this kind of analysis starts. After establishing frequencies, we calculate parameters such as support, confidence and lift. For a model to have statistically relevant results it is necessary to analyze thousands of transactions. Purchase order represents unique event that customer generated. Purchase order array is a data structure that was necessary to create for Market Basket analysis. Tracking purchase order through time allowed separation of statistically important products.

Market Basket analysis answers the questions on average number of purchase orders per buyer, trending of average purchase orders per customer during time, average number of unique products per purchase, average size of purchase, most popular products sales, least popular products, average purchase size by specific dimensions (geography, payment method, shipment method, time of the year etc.). Since we didn't observe this problem through the focus on buyers and sellers but exclusively through products, we modified Market Basket algorithm. This decision was made because in this particular case most of the buyers are buying only one product, and in the most cases sellers sell one product.

In this algorithm we applied association rule mining method, which finds patterns, associations, correlations and causal structures amongst sets of items in transactions. This analysis is frequently used in classical data analysis, cross-marketing, catalogue planning, loss analysis, web logs analysis, fraud detection, etc.

Vector Distance model is original model that was used in result optimization. Auction Items are defined by their characteristics, which are aggregated into string variables *ParentSequenceID* and *SetOfValues*. Characteristics and catalogue subgroups for each product are represented by their respective IDs, and they are formatted as string variables.

This model is based on the two-vector distance. If we have Euclidean n-dimensional space, and two vectors $\vec{u}, \vec{v} \in \mathbb{R}^n$, then the distance between vectors is given as:

$$d(\vec{u}, \vec{v}) = \|\vec{u} - \vec{v}\| = \sqrt{(u_1 - v_1)^2 + (u_2 - v_2)^2 \dots (u_n - v_n)^2}$$

Algorithm descriptions, choosing relevant variable set and visualization of generated clusters are shown in chapter two. In chapter three we show segmentation of products on diagrams that illustrate connections between number of sales, total value of sales and product type. In concluding chapter, we present possibilities of proposed algorithm usage, and further optimizations in terms of future research.

2. Product segmentation description

Method we used in this research consists of the following algorithms:

- Cluster analysis
- Cluster separation and compactness detection
- Market basket analysis
- Vector distance model

2.1.Cluster analysis

First algorithm we applied is K-means algorithm.

Primary goal of cluster analysis grouping is separation of objects into two or more groups based on the similarity of certain features, represented by cluster variables (Jain, 2010). Homogeneity inside the cluster means that all the data in the given cluster are as similar as possible, while heterogeneity between different clusters means that the data belonging to different cluster must differ as much as possible. The concept of similarity is determined depending on the actual data. In this paper we used Euclidean distance between data as the measure of similarity, and buyer data were used for grading to ensure reasonable distance value ranges. Centroids that represents centers of the clusters does not need to be the data from source data set, because they represent points that are calculated by averaging of parameters belonging to points in given clusters (Kanungo et al., 2002). As an output data we also get to which of the K clusters each data belongs.

Based on analysis of the attributes from MySQL database of the given e-commerce platform, we generated the proposal of grouping the products that have similar characteristics and excite similar behavior in customers in nine *CategoryRootID* categories. Since the given e-commerce platform sells wide variety of products, we had to create initial product groups having in mind similarities between products, and each product was given his own *CategoryRootID* identifier. After that, clustering model was applied to every group separately.

CategoryRootID:

1. clothing, footwear
2. accessories
3. babies, pets
4. body care, sport
5. art, hobbies
6. for house
7. machines, technics, cars
8. computers, mobile phones
9. services, real estates

Variables for clustering were defined based on scatter plot and Principal Component analysis.

Scatter plot diagram represents graph with scattering, and it uses cartesian coordinates to show values for typically two variables. Data are dot sets, and each dot has a value of one variable. Diagram is used for clustering model preparation, having in mind simple visualization of variable couples' correlation.

Table 1. Scatter plot diagrams

Although K-Means result is not too disturbed by correlated variables, they need to be eliminated. However, if we eliminate wrong variables or if we do it in a wrong way, we can disrupt information, or to artificially bring closer points that are not close in general population. That is why we use PCA analysis which compress data dimension and thus eliminate correlated data while preserving data variance. Variance should be normalized before analysis to avoid mistakenly removing variables only because they have different range in their values. PCA assumes orthogonal base, so if we encounter

data with variance in non-orthogonal direction – these data will not be correctly clustered with K-Means algorithm.

PCA (*Principal Component Analysis*) is statistical procedure with which we map the set with possibly correlated variables into a set of uncorrelated variables. Each of the attributes from the new set is called principal component, and they are all linearly uncorrelated. When own values and vectors are calculated using preprocessed data table, we choose first k sorted own vectors whose own values are different than 0. If we work on such formed own vectors, we will not lose information.

$k \leq \min(n-1, m)$, where $M(m, n)$ is a matrix of own vectors with m rows and n columns. To perform PCA analysis efficiently (and later on cluster analysis as well), we need to standardize data by the principle of z-score formula over all variables:

$$z = (x - \mu) / \sigma$$

PCA analysis is applied on final table with 4 variables of numerical type: *Highest Bid*, *Number of bids*, *Starting price* and *Number of visits*, and on the combination and pairs of these variables. Largest variance was found in variable pairs of *Starting price* and *Number of bids*.

Axis shows vectors that starts from the center, Variables *Starting price* and *Number of bids* are used in PC1 with their highest values giving most information on the data from the sample. PC3 and PC4 explain small percentage of total variance, so they are not too informative for detecting of the apparent patterns.

Table 2. Importance of Components

	PC1	PC2	PC3	PC4
Standard deviation	1.5237	1.0799	0.59997	0.39030
Proportion of Variance	0.5804	0.2915	0.08999	0.03808
Cumulative Proportion	0.5804	0.8719	0.96192	1.00000

As a conclusion, PCA and Scatter plot showed that interesting attributes are *Starting price* and *Number of bids*.

At the start we created two new tables, from the final table. They represent train and test set. Sets are di-vided by using random sample method. Cluster analysis was done for all 9 tables, and we ran them through distribution check for aforementioned variables, to adjust variable metrics as a part of basic preprocessing. Next procedure is using Elbow method to determine optimal number of clusters. Based on the results of this method, we chose to have 3 clusters for each group, with most of 15 clusters per iteration. Elbow method uses *wss* procedure to explain homogeneity inside the cluster. *Wss* measure is being drawn in relation to the recommended number of clusters, so we call it in iterations.

Preparation of train and test set was the next phase in this part of algorithm. Both sets were formed from the final preprocessed set of data, by the principle: 75% train set – 25% test set, based on random choice.

Cluster analysis is performed on mean value metrics, which means that centroids are calculated with the function *mean()*, based on all elements in cluster group. This model was applied on train set of data, and when we got satisfactory data, we put test set through verification, and it also yielded satisfactory results. One of the first result was calculating the distance from centroid for each of the elements in the cluster, is the distance of all products from the cluster centroid. Centroid is not one exact product, but the mean value of *Starting price* and *Number of bids* columns of all elements within the cluster.

Since we formed 9 groups and we defined 3 clusters, the result is 27 different non-overlapping cluster groups.

When the seller wants to list his product on website for sale, he has a product with certain array of properties which he chooses from predefined set of properties, and his only dilemma is what starting price for his auction he should choose. So, the role of this research should be to make it easier for seller to choose the right starting price. Our goal is to recommend the starting price to seller, based on the

existing data and with displayed data of historical bids on similar products in relation with starting price.

Besides starting price, second important parameter in this research is the *Number of bids*. Auction with large *Number of bids* value will have bigger final price (seller satisfaction will be increased) and website as a whole will have more visitors.

Based on these assumptions, *Starting price /Number of bids* relationship was chosen for clustering. For cluster separation in three clusters on test set with one hundred products we got the following data (show on Figure 1):

Fig. 1 Sample Clustering

We can see three different clusters:

- yellow – products with large starting price and small number of bids
- purple – products with small starting price and small number of bids
- green – products with small starting price and large number of bids

This kind of clustering is repeated during the algorithm execution on real data, as will be shown below. Clustering method is being applied on 9 tables, each separated in 3 clusters, to keep cluster within their *CategoryRootID* properties.

Product group, Clothing and footwear

Fig. 2. Clothing and footwear Clustering

Centroids for Clothing and footwear:

- Starting price=405.86, Number of bids=2.55
- Starting price=2052.83, Number of bids=2.65
- Starting price=439.94, Number of bids=24.43

Product group accessories

Centroids for Accessories:

- Starting price=2309.50, Number of bids=3.3
- Starting price=173.81, Number of bids=36.2
- Starting price=362.39, Number of bids=3.34

Product group Babies and pets

Centroids for Babies and pets:

- Starting price=212.13, Number of bids=13.63
- Starting price=356.6, Number of bids=1.78
- Starting price=167.5, Number of bids=40.7

Product group Care and sport

Centroids for Care and sport:

- Starting price=318.24, Number of bids=3.12
- Starting price=151.35, Number of bids=24.33
- Starting price=3551.1, Number of bids=2.69

Product group Art and hobby

Fig. 6 Art and hobby

Centroids for Art and hobby:

- Starting price=129.24, Number of bids=11.14
- Starting price=159.45, Number of bids=1.77
- Starting price=148.32, Number of bids=31.99

Product group House items

There is one product group which is segmented as *CategoryRootID=6*. This group represents home products (for house). Since there was no valid data in the timeframe, this group was not included in further modeling. This is the reason, why this group did not position itself in clustering.

Product group Machines, technics and cars

Fig. 7 Machines, technics and cars Clustering

Centroids for Machines, technics and cars Clustering:

- Starting price=798.9, Number of bids=3.06
- Starting price=281.21, Number of bids=21.58
- Starting price=191.06, Number of bids=49.69

Product group Computers and mobile phones

Fig. 8 Computers and mobile phones Clustering

Centroids for Computers and mobile phones:

- Starting price=1032.19, Number of bids=3.46
- Starting price=341.54, Number of bids=23.7
- Starting price=217.85, Number of bids=57.41

Product group Services and real estates

Fig. 3 Services and real estates Clustering

Centroids for Services and real estates:

- Starting price=453.44, Number of bids=2.55
- Starting price=223.54, Number of bids=22.02
- Starting price=187.16, Number of bids=60.01

We confirmed that cluster distribution is the same for sample data and for data separated by CategoryRootID.

2.2. Cluster separation and compactness detection

To detect separation and compactness of our clusters, we used Silhouette index, S . If $X\tau=\{X_1, \dots, X_N\}$ is data set, then $C=(C_1, \dots, C_K)$ represents clustering of that set on K clusters. $d(X_k, X_l)$ is distance between X_k and X_l . If $C_j = \{X_1^j, \dots, X_{m_j}^j\}$ is j cluster, $j=1, \dots, K$, then $m_j=|C_j|$. Mean distance a_i^j between i -th vector in cluster C_j and other vectors in the same cluster is defined by the following formula:

$$a_i^j = \frac{1}{m_j - 1} \sum_{\substack{k=1 \\ k \neq j}}^{m_j} d(X_i^j, X_k^j), i = 1, \dots, m_j \quad (1)$$

Minimal mean distance between i -th vector in cluster C_j and all vectors from clusters $C_k, k=1, \dots, K, k \neq j$, is defined with:

$$b_i^j = \min_{\substack{n=1, \dots, K \\ n \neq j}} \left\{ \frac{1}{m_n} \sum_{\substack{k=1 \\ k \neq j}}^{m_n} d(X_i^j, X_k^n) \right\}, i = 1, \dots, m_j \quad (2)$$

where d represents the distance, which can be calculated as Euclidean distance, Minkovski distance, Manhattan distance etc. (Anderson, 2003). Silhouette width of i -th vector from cluster C_j is defined as:

$$s_i^j = \frac{b_i^j - a_i^j}{\max(a_i^j, b_i^j)} \quad (3)$$

based on that we get that $-1 \leq s_i^j \leq 1$. Silhouette width for cluster C_j is defined as:

$$S_j = \frac{1}{m_j} \sum_{i=1}^{m_j} s_i^j \quad (4)$$

while global Silhouette clustering index is defined as:

$$S = \frac{1}{K} \sum_{j=1}^K S_j \quad (5)$$

Consequence of that is that cluster width, as well as global Silhouette clustering index, have values between -1 i 1. Silhouette index value calculated by using silhouette function in MATLAB programming environment (The MathWorks, Inc., 2020) for suggested clusters, is shown on Fig .10.

Fig. 4 Silhouette index value for chosen clusters

Visual representation of clusters, with mean value of relevant parameters, is shown on Fig.2, Fig.3, Fig.4, Fig.5, Fig.6, Fig.7, Fig.8 and Fig.9.

2.3. Market Basket analysis

Second algorithm we applied was the Market Basket algorithm. We have collection of objects as in the be-ginning: $D = \{R_1, \dots, R_n\}$ where each record is noted as $R_i = \{d_1, \dots, d_n\}$ and where d_i are variables (features). Each record is a product which is represented through its characteristics, mathematically it is a tuple $d_i = (id(i), maxbid(i), numberbid(i), startprice(i), idroot(i), parent(i), values(i), visits(i))$. Within

this collection of objects, we added another feature we called “Cluster”, which is derived from clustering from previous step. Clustering is calculated on nine tables, and for each product group we have three clusters.

The point of this model is to count and compare characteristics from *ParentSequenceID* and *SetOfValues* for each *AuctionItemID*. Ratio of products count from some *ParentSequenceID* to total product count varies from 0.002‰ (when there are just one product for the given *ParentSequenceID*, like 26!!!1129, 11!!!43, 6!!!144, 26!!!1129!!!1140), to 2,94%, (as is the case for *ParentSequenceID*=35!!!1214!!!31).

Model result is to identify the products which are similar through *ParentSequenceID*, with the goal of finding the product similar to one buyer already bought or put in his wish list.

For example, by making query for *ParentSequenceID*=5!!!1282!!!1633, over the set of data in table *Final*, we would get 312 products, and the product recommended to buyer would be the one with the closest finishing time.

Products are being found and separated based on matching *ParentSequenceID* data and matching the data values from *SetOfValues*.

For purposes of testing the model and quick separation of products which are similar with the given product (for example, with the product seller is entering), we first created table with the columns: *AuctionItemID*, *ClusterID*, and the product distance from the center of the cluster. This table remains connected to the original data table. We then constructed appropriate SQL query which for any value from *SetOfValues* and for any *CategoryRootID* returns the list of products that are similar to the given product, sorted descending by the distance from centroid. First on the list is the product with the similar properties to the given product and with the smallest distance to centroid, which can be then shown to the seller when he is entering starting price for his product.

Following are examples of finding the products with the same or similar parameters as a given new product, which are closest to the cluster centers, accompanied with our conclusions:

Example 1: New product which has *ParentSequenceID*=5!!!359!!!428 and whose data set contains values 10547 and 8603, has the following similar existing products (products closest to the cluster center):

- For cluster 0, *AuctionItemID* = {70626333, 72267701}
- For cluster 1, *AuctionItemID* = {68290377}
- For cluster 2, *AuctionItemID* = {69694565, 68475485}

Product description for each cluster:

- Products from the cluster 0 had medium starting price (450 din) and 3 bids each
- Product from the cluster 1 had big starting price (4699 din) and 1 bid.
- Products from the cluster 2 had small starting price (99 din) and 36 bids each.

Conclusion: For new product, recommended starting price is 99 din.

Example 2: New product which has *ParentSequenceID*=5!!!26, has the following products from previous auctions (which are closest to the cluster center):

- For cluster 0, *AuctionItemID* = {69999289, 68625913, 69309753, 71842385}
- For cluster 1, *AuctionItemID* = {65331265}
- For cluster 2, *AuctionItemID* = {69168109}

Product description for each cluster:

- Products from the cluster 0 had medium starting price (450 din) and 3 bids each.
- Products from the cluster 1 had big starting price (4607 din) and 1 bid.
- Product from the cluster 2 had small starting price (99 din) and 37 bids.

Conclusion: For new product, recommended starting price is 99 din.

Example 3: New product which has *ParentSequenceID*=36!!!1283!!!1289, has the following products from previous auctions (which are closest to the cluster center):

- For cluster 0, AuctionItemID = {71007505, 70295133, 71755825}
- For cluster 1, AuctionItemID = {71975677}
- For cluster 2, AuctionItemID = {70518725, 70221645}

Product description for each cluster:

- Products from the cluster 1 had smaller starting price (199 din) and 24 bids.
- Products from the cluster 0 had medium starting price (2000 din) and 3 bids each.
- Products from the cluster 2 had bigger starting price (400 din) and 3 bids.

Conclusion: For new product, recommended starting price is 199 din.

Example 4: New product which has *ParentSequenceID*=29!!!1263!!!584, has the following products from previous auctions (which are closest to the cluster center):

- For cluster 0, AuctionItemID = {70870433 }
- For cluster 1, AuctionItemID = {70733609, 70962033}
- For cluster 2, AuctionItemID = {69548173}

Product description for each cluster:

- Product from the cluster 0 had small starting price 99 din and 41 bid.
- Products from the cluster 1 had bigger starting price 350 din and 2 bids each.
- Product from the cluster 2 had smaller starting price 199 din and 14 bids.

Conclusion: For new product, recommended starting price is 99 din.

Example 5: New product which has *ParentSequenceID*=10!!!73, has the following products from previous auctions (which are closest to the cluster center):

- For cluster 0, AuctionItemID = {71649689}
- For cluster 1, AuctionItemID = {68371893}
- For cluster 2, AuctionItemID = {71219521}

Product description for each cluster:

- Product from the cluster 0 had medium starting price 333 din and 3 bids.
- Product from the cluster 1 had small starting price 199 din and 24 bids.
- Product from the cluster 2 had big starting price 3500 din and 3 bids.

Conclusion: For new product, recommended starting price is 199 din.

Example 6: New product which has *ParentSequenceID*=35!!!1214!!!31, has the following products from previous auctions (which are closest to the cluster center):

- For cluster 0, AuctionItemID = {66797725}
- For cluster 1, AuctionItemID = {68399293}
- For cluster 2, AuctionItemID = {72988029}

Product description for each cluster:

- Product from the cluster 0 had medium starting price 150 din and 2 bids.
- Product from the cluster 1 had small starting price 99 din and 33 bids.
- Product from the cluster 2 had medium starting price 149 din and 11 bids.

Conclusion: For new product, recommended starting price is 99 din.

Example 7: New product which has *ParentSequenceID*=14!!!56!!!289, has the following products from previous auctions (which are closest to the cluster center):

- For cluster 0, AuctionItemID = {69356353}
- For cluster 1, AuctionItemID = {72756249}
- For cluster 2, AuctionItemID = {69717989, 70441877}

Product description for each cluster:

- Product from the cluster 0 had small starting price 10 din and 48 bids.
- Products from the cluster 1 had big starting price 799 din and 3 bids.
- Products from the cluster 2 had medium starting price 199 din and 21 bids.

Conclusion: For new product, recommended starting price is 10 din.

Example 8: New product which has *ParentSequenceID*=31!!!845, has the following products from previous auc-tions (which are closest to the cluster center):

- For cluster 0, AuctionItemID = {69718421}
- For cluster 1, AuctionItemID = {71972929}
- For cluster 2, AuctionItemID = {68733521}

Product description for each cluster:

- Product from the cluster 0 had small starting price 199 din and 25 bids.
- Product from the cluster 1 had big starting price 1000 din and 4 bids.
- Product from the cluster 2 had small starting price 199 din and 61 bids.

Conclusion: For new product, recommended starting price is 199 din.

Example 9: New product which has *ParentSequenceID*=30!!!1785!!!1813, has the following products from previ-ous auctions (which are closest to the cluster center):

- For cluster 0, AuctionItemID = {69465069}
- For cluster 1, AuctionItemID = {71107713}
- For cluster 2, AuctionItemID = {69452041}

Product description for each cluster:

- Product from the cluster 0 had big starting price 500 din and 2 bids.
- Product from the cluster 1 had small starting price 99 din and 58 bids. (In cluster 1, there were no products with *ParentSequenceID*=30!!!1785!!!1813, so we took product with *ParentSequenceID*=30 as an ex-ample for comparison)
- Product from the cluster 2 had starting price of 100 din and 18 bids.

Conclusion: For new product, recommended starting price is 99 din.

Table 3 Example from the resulting table

ParentSequenceID	Number of products	Ratio in tenth of percent (%)
35!!!1214!!!31	14955	29.39973814
3!!!81!!!622	14651	28.80211057
6!!!118!!!1416	10863	21.35535643
33!!!978!!!982	9606	18.88424504
36!!!1283!!!1289	8750	17.20145161
33!!!978!!!1419	8220	16.15953511

2.4.Vector Distance model

Third algorithm described in twining sales model is the Vector Distance model. We have object collection as in the beginning $D = \{R_1, ..., R_n\}$ where each record is noted as $R_i = \{d_1, ..., d_n\}$ and d_i are variables (features). Each record is the product which is represented through characteristics, in mathematical notation it is a tuple $d_i = (id(i), maxbid(i), numberbid(i), startprice(i), idroot(i), parent(i), values(i), visits(i))$. This algorithm calculates products which have the same or similar parameters and which are closest to the center of the cluster or to the sold product. The distance is calculated with Euclidean formula:

$$d(p, q) = d(q, p) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \dots + (q_n - p_n)^2} = \sqrt{\sum_{i=1}^n (q_i - p_i)^2}$$

$p = (p_1, p_2, \dots, p_n)$ i $q = (q_1, q_2, \dots, q_n)$ are two points in Euclidean space R_n , in cartesian coordinates. Distance vectors were created by comparing coordinates of *ParentSequenceID* vectors for each *CategoryRootID* and for each pair of tuples (records), tuples were sorted by the distance from centroids. We formed the resulting array $r = (r_1, r_2, \dots, r_n)$ which contained only values 1 or 0, these values were assigned based on matching of *ParentSequenceID* values (1=successful match, 0=unsuccessful match).

For all cluster members which represents one tuple (product, record) and for all coordinates of *ParentSequenceID* vector and *SetOfValues*, we check whether the product is similar to other products and in what measure, by coordinates. If two products are matched, we set 1 in the resulting vector on appropriate position, and vice versa – we set 0 when there is no match. We must save all possible r vectors, their count is size cluster – 1, which is very large number. For each product we choose the vector with the most 1 on coordinate places.

3. Results of segmented products

With the goal of finding correlation between variables *StartingPrice* and *NumberOfBids*, we used cluster analysis and segmentation on three clusters, which showed the following:

1. First cluster contains products which had big starting price and small number of bids
2. Second cluster contains products which had small starting price and medium number of bids.
3. Third cluster contains products which had small starting price and big number of bids.

First conclusion is that big starting price bears the risk of small number of bids. Second conclusion is that products with small starting price can have small but also big number of bids.

We suggest that price recommendation system for sellers should be implemented. System should help seller to make a decision on starting price. After the seller enters product data, system will show other products data – for products that are similar to the one seller entered. These products are representatives from three clusters with the smallest distance from the cluster centroid. In that way we can suggest to seller when to expect smaller number of bids for bigger starting price, and when it is good to set smaller starting price to achieve bigger number of bids.

4. Future work

In future work, we can add the following analysis:

- Create cluster analysis and determine the influence of auction duration (time between start and end of an auction) to final price difference (difference between starting and ending price), and then to suggest to seller which is the optimal auction duration in days.
- Collecting and analysis of the data on interest of site customers for auctions.
- Collecting and analysis of the data on buyers who bought more than one product.

5. Conclusion

We introduce e-commerce Recommender System consists of Market Basket analysis and Vector Distance model with Clustering model in this paper. The model is given as a conceptual combination of several algorithms since it tries to pull customers into increasing average order value by offering products that go along with the current purchase.

A Recommender System works like a great salesman who has intelligent decisions for cross selling and upselling. It uses information such as revenues and different counts (purchases, clicks, etc.). These data are various ratings that customers leaves for products that are trained for models. These information provided solution for great customer experience.

The main aim in this model was to properly segment the groups of products that were later used for Clustering, Market Basket and Vector Distance. Products were segmented according to feature *CategoryRootID* variable and the Clustering was applied on every segment with 3 clustering groups.

Vector Distance model was applied further on every cluster group, since the features of product were recoded in numerical values and it was possible to create distances between them.

Based on the data, buyers and website visitors, we can recommend product which are currently on auction in the following ways:

- Based on the data of previous sales, it is possible to find similar active products and recommend them to buyers
- Based on data correlation through *ParentSequenceID*
- Based on correlation of the data from *SetOfValues*
- To use the same algorithm as for finding products similar to the one's customer bought, but to implement it on the data on product buyer is interested in (wishlist for example).

To use the data on buyers who bought more than one product for "Market basket" analysis, and based on that analysis to recommend products from other categories.

References

1. Agresti A. (2002.), *Categorical Data Analysis*, 2nd ed., Wiley, New York
2. Anderson T. (2003.), *An Introduction to Multivariate Statistical Analysis*, 3rd ed., Wiley, New York
3. Bejju A. (2016.), *Sales Analysis of E-Commerce Websites using Data Mining Techniques*, International Journal of Computer Applications, 133, pp. 36-40
4. Bishop C. (2006.), *Pattern Recognition and Machine Learning*, Springer, New York
5. Celebi M. E., Kingravi H. A., Vela P. A. (2013.), *A comparative study of efficient initialization methods for the k-means clustering algorithm*, Expert Syst. Appl., pp. 200-210.
6. Cheng C.-H., Chen Y.-S. (2009.), *Classifying the segmentation of customer value via RFM model and RS theory*, Expert Systems with Applications 36, 2009, pp. 4176-4184.
7. Hastie T., Tibshirani R., Friedman J. (2009.), *The elements of statistical learning: Data Mining, Inference, and Prediction*, 2nd ed., Springer
8. Ismail M., Ibrahim M., Sanusi Z., Cemal Nat N. (2015.), *Data Mining in Electronic Commerce: Benefits and Challenges*, International Journal of Communications, Network and System Sciences, 08, pp. 501-509
9. Jain A. K. (2010.), *Data clustering: 50 years beyond K-means*, Pattern Recogn. Lett. 31, pp. 651-666.
10. Kanungo T., Mount D. M., Netanyahu N. S., Piatko C. D., Silverman R., Wu A. Y. (2002.), *An efficient kmeans clustering algorithm: Analysis and implementation*, IEEE Trans. Pattern Anal. Mach. Intell. 24(7), pp. 881-892.
11. Kubiak B., Weichbroth P. (2010.), *Cross- And Up-selling Techniques In E-Commerce Activities*, Journal of Internet Banking and Commerce, 15
12. The MathWorks, Inc., *Silhouette plot - MATLAB silhouette* [Online] Available: <https://www.mathworks.com/help/stats/silhouette.html>
13. McDaniel C., Gates R. (2014.), *Marketing Research*, 10th ed, John Wiley & Sons
14. Nayak J., Kanungo D.P., Naik B., Behera H.S. (2016.), *Evolutionary improved swarm-based hybrid K-means algorithm for cluster analysis*, Proceedings of Int. Conference on Computer and Communication Technologies, Springer, New Delhi, pp. 343-352.
15. Nur Sagin A., Ayvaz B. (2018.), *Determination of Association Rules with Market Basket Analysis: Application in the Retail Sector*, Southeast Europe Journal of Soft Computing
16. Rachid A. (2015.), *Customer Segmentation Model in E-commerce Using Clustering Techniques and LRFM Model: The Case of Online Stores in Morocco*, International Journal of Computer, Electrical, Automation, Control and Information Engineering, 9, pp. 1795 – 1805
17. Stefanović H., Veselinović R., Bjelobaba G., Savić A. (2018.), *An adaptive car number plate image segmentation using K-means clustering*, Proceedings of Int. Scientific Conference on Information Technology and Data Related Research-SINTEZA 2018, pp. 74-78.

18. Tibshirani R., Walther G., Hastie T. (2001.), *Estimating the number of clusters in a dataset via the gap statistic*, Journal of the Royal Statistical Society, Series B. 32(2) pp. 411–423.
19. Wang Z., Xu Z., Liu S., Yao Z. (2014.), *Direct clustering analysis based on intuitionistic fuzzy implication*, Applied Soft Computing 23, pp. 1-8.