**National Research University Higher School of Economics**

**Faculty of Computer Science**

**HSE and University of London Double Degree Programme in Data Science and Business Analytics**

**BACHELOR'S THESIS**

**(Research Project)**

**Cold start of recommendation system using methods of link prediction**

**Prepared by the student of Group 181 Year 4,**

**Andreev Mikhail Alexeevich**

**Thesis Supervisor:**

**Senior professor at SAS department, Titova Natalia Nikolaevna**

**Moscow**

**2022**

**Аннотация**

Рекомендательные системы в электронной коммерции призваны улучшить взаимодействие клиентов с интерфейсом, а также качество потребительского опыта. Однако ограниченность данных о новых пользователях в рекомендательных системах – холодный старт, не дает возможности эффективно использовать даже имеющуюся информацию. Для решения данной проблемы было предложено представить транзакционные данные о клиентах сначала в виде двухстороннего графа и в виде гетерогенного графа. В рамках работы приводится сравнение нескольких алгоритмов классификации и эвристик для предсказания существования ребер в графе. Кроме того были протестированы несколько алгоритмов для векторного представление графа и предложено улучшение для случая гетерогенных графов. Лучший результат в 0.74 и 0.77 по метрикам map@5 и map@10 соответственно был получен с использованием гетерогенного графа, векторным представлением с помощью алгоритма Node2Vec и ансамблем деревьев в качестве классификатора.

**Abstract**

Recommendation systems in e-commerce are used to improve interaction of clients with interface and quality of consumption experience. However, the sparsity of data about new clients in recommendation systems – cold start, limits the ability for effective usage even of present information. To tackle the presented problem was suggested the interpretation of transactional data about the clients in the form of the bipartite as well as heterogeneous graph. Within the work the comparison of the classification algorithms and heuristics for prediction of links existence is present. Apart from that, algorithms for vector representation of the graph like data were compared and suggested improvements for the case of heterogenous graphs. The best result of 0.74 and 0.77 of map@5 and map@10 metrics respectively were shown with the use of heterogenous graph, Node2Vec algorithm for vector representation and ensembles of decision trees for classification.

**Table of content**

[1. Introduction 4](#_Toc104556995)

[2. Goals of the work 6](#_Toc104556996)

[3. Literature review 7](#_Toc104556997)

[4. Problem standing 13](#_Toc104556998)

[5. Data collection 17](#_Toc104556999)

[6. Design and results 17](#_Toc104557000)

[7. Evaluation and Conclusion 25](#_Toc104557001)

[8. Bibliography: 25](#_Toc104557002)

1. Introduction

The number of internet users constantly increases, and in 2022 comprises almost 5 billion of people [1]. Moreover, the online purchases share of retail sales is forecasted to be a fifth in 2022 [2]. The latter encourages the electronic commerce development among all the market spheres. As in the case of classical retail stores the digital solutions’ clients demand and buying behavior are dependent on the shelf-spaces allocation and product display [3]. The showcase in ecommerce is a user’s screen the management of which benefits retailers as well as buyers.

Recommendation of products in online shops is one of the marketing departments staff tasks which can be used to optimize product liquidity, boosting of sails, decision power in negotiations with suppliers, popularity among them and brand building of the products. On the other hand, users of online retail stores benefit in their experience with shop content shortcutting to the required items. The clients’ experience with recommendation systems and other value metrics other than accuracy are considered as an important aspect as well [4]. The request for good recommendation systems is getting even more acute because in context of crisis the customers reduce their spendings and suppliers change their brand images [5].

The constant need to process and analyze large stacks of information made the instruments of machine learning the must in approaches to several marketing problems such as: predicting churn-rate, customer lifetime value, price and volume of purchases, calculating the effectiveness of a marketing campaign for a particular client etc.

The recommendation of products to clients is also one of the electronic commerce problems that is usually solved using machine learning techniques. The main objective of the task is to show clients only the products that they will purchase. Current state of the art approaches in recommendation systems are based on the neural network architectures and require considerable amount of information about clients as well as products [6, 7, 8]. However, the work of Maurizio Ferrari Dacrema [2] carefully criticise the versatility along with reproducibility of recommendation system algorithms considered on the RecSys conference [42] in 2018 and 2019. He made it clear that classical approaches with reasonable adjustments may outperform state of the art solutions. Moreover, not all of the retail stores can afford implementation and maintenance of state-of-the-art solutions due to financial limitations.

Another problem of middle-sized retail stores is in their poor data sources. While big competitors may afford tracking the time spent on the page, number of clicks, ratings and search history of the clients, those shops are framed to operate the information of transactions. Those datasets provide integrable information neither about client nor about the products, but rather the transactional specifications. Following, the smaller is the retail shop the less loyal is the client flow, meaning that clients may have only a few products in their baskets as well as few purchases throughout the relationship with the shop. The latter are some reasons for e-shops to halt the ideas of implementing recommendation systems. The problems related to the lack of data at the beginning of recommendation system work are called cold start. The possible way to overcome those problems may be the graph representation of the data with further link prediction algorithms implementation.

Thus, within the framework of the final qualification work, an extremely urgent task of cold start of recommendation system using methods of link prediction is considered.

1. Goals of the work

The **main goal** of the work is investigation and designing of recommendation system methods in the case of cold start using link prediction techniques.

The accomplishment of the works’ goals requires aiming following problems:

1. Formalization of the recommendation problem in the case of cold start using link prediction methods.
2. Design and implement the method for product recommendation in the case of cold start using link prediction methods.
3. Design the methodology of experimental investigation of the suggested method.
4. Collect and prepare the initial data based on transactional data of retail store as well as written description of product groups.
5. Conduct the experimental investigation of the suggested method and evaluate the results.

Therefore, the object of the study, conducted in the frames of bachelor’s degree work is the product recommendation method in the case of cold start using link prediction methods of machine learning. The subject of the study – approaches to implementation and application of new recommendation system method using link prediction techniques in case of cold start and justification of its need.

1. Literature review
   1. **Recommendation systems**

Recommendation systems (RS) are not new tools, in the early 90’s researchers of Xerox Palo Alto Research Center (PARC) led by David Goldberg designed the system that allowed the user to search for documents in the electronic mail based on the topic as well as the ratings assigned to them by other users [11]. They used one of the low ranked methods called collaborative filtering. Collaborative filtering is based on the idea that the people with similar ranks assigned to the items will continue to have similar experience about the items in the future. The algorithm suggested by Goldberg and co. is not used in modern RS, yet its ideas stand behind many modern untrivial solutions, see [6, 12, 13], adopted by the big companies, see [14, 15, 16].

On the other hand, stand recommendation systems that are based on the similarity of the items. Those methods are summarized under the term content-based filtering. Content based filtering is based on the idea that similar products should be recommended together, and it is justified to suggest alike items as the output of RS. Those methods are useful since they don’t require data about users. Those ideas are shared in works [17, 18, 19] as well as used in the industry [20, 21].

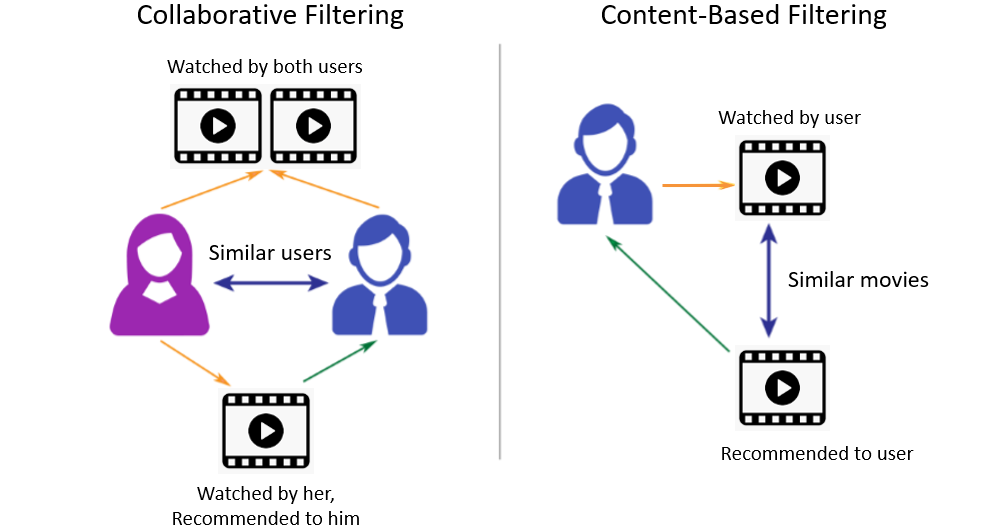


Figure 1 Difference between Collaborative Filtering and Content-Based Filtering recommendation systems (<https://jnyh.medium.com/content-based-recommender-using-natural-language-processing-nlp-159d0925a649> )

The most common problem of recommendation system is the absence of data about the objects. In this case, both the information from users and products are aimed to be mixed to form sufficient amount of information. The approaches that combine theory of content-based filtering and collaborative filtering are called hybrid methods [23].

**Top-N**

The most known and trivial approach to the ranking problem is a Top-N algorithm [27]. It is based on the idea of overall popularity of items for users. The algorithm finds the subset of items that were chosen by the users the most often. Then at the stage of the inference it suggests all users with the recommendation from that set. This approach is basic and can be considered only as a baseline for the problem.

**User-based kNN**

One of the common collaborative filtering approaches applicable to the target problem is user-based kNN [26]. This approach is based not on the parametric distribution assumptions, but on the similarity calculation. In this case the distance in the feature space of the user is calculated between every two elements. Then k elements with defined class are taken into consideration at the level of inference of the model. Being very convenient and straightforward approach collaborative filtering through kNN is usually taken as baseline of the recommendation problems, in this sense our case is not an exception.

* 1. **Problem of cold start**

The problem of the cold start may be found in all the cases of recommendation systems. It stands out as a lack of input information based on which the recommendation is planned to be conducted. Three types of cold start problems are recognized [22]. Those are:

* Item cold start. That case is related with the lack of information about the new product, which may be the features of the item themselves or any kinds of ratings assigned to them. In that case the content-based approaches are performing poorly.
* User cold start. That case is related to the lack of information about the new user, which may be the features of user themselves or any kinds of interactions with items. In that case the collaborative filtering approaches are performing poorly
* Systematic Bootstrapping. The case of the fresh start of the recommendation system with mostly the absence of the information on which the algorithms may rely.

**LightFM [23]**

One of the common approaches to that problem is hybrid method implemented in LightFM library which is based on the matrix factorization approach [24]. Both the elements of the user set and item set are vectorized in form of embeddings with bias terms. The probability of interactions between the item and the user is then estimated as the sigmoid of the dot product of two vectors with their bias terms. The method is also implemented with the use of the WARP and Bayesian Personalised Ranking losses, which tend to provide better results. The embeddings in LightFM are generated similarly to the Glove and word2vec [34] models, yet it does not operate the statistic of words appearance together, but rather is based on the user interaction data.

* 1. **Link prediction techniques**

In the theory of link prediction, the recommendation systems are formalized to the bipartite graphs [28]. Those are the graphs denoted as where and are two different sets of nodes, stands for the set of edges that connect vertices from with , having no within sets’ connections. Here is the set of users and is the set of items. The connection between them can be any investigated action, which in our case is a purchase.

The notation taken from [29] of the further measures in link prediction approaches is as follows:

– is a pair of user and product nodes without an edge between them.

– represents distance between two nodes.

– is the set of neighbors of node . refers to the set of all nodes connected by an edge of any type to x generally written as Γ (x) and used to represent the degree of a node.

– denotes the set of common neighbors between node and . In bipartite graph, this set is empty.

– contains all the common neighbors within -hop distance between nodes and . In bipartite graph, needs to be even to start and end path between same type of nodes and odd if path starts and ends between nodes of different types. As recommender system tries to recommend items to users, the odd length paths are meaningful.

– denotes the set of paths connecting and by at most edges.

The measures aimed to investigate the nature of the graph that were chosen for the research are those from [28].

– **Common Neighbors (CN)**: The common neighbor measure in bipartite graph is given as follows:

This measure captures the idea of node being close to another node through high number of common nodes at hop distance 3. Jaccard Coefficient, Adamic Adar and Preferential Attachment measures are also neighborhood-based defined in a similar way as follows.

– **Jaccard Coefficient (JC)**: Jaccard Coefficient is the normalized CN measure

– **Adamic Adar (AA)**: This measure gives importance to the common neighbors with low degree. This measure attempts to linearize resource allocation measure, by taking the logarithm from the denominator.

– **Preferential Attachment (PA)** : This measure means that the more connected a node is, the more likely it is to receive new links.

The proposed bipartite graph measures are of low benefit in the case of cold start. The reason for that is sparsity of connections in those problems. Therefore, there is a need in increasing the number of entities in the information graph which are connected. Those may be the features of users or items. In that case the graph becomes **Heterogeneous Information Network** (HIN) [30].

[31] The HIN is denoted as , which consists of an object set 𝒱 and an edge set . A HIN is also associated with an object type mapping function ∅ ∶ 𝒱 → 𝒜 and a link type mapping function 𝜓 ∶ → . and denote the sets of predefined object and link types, where |𝒜| + || > 2.

[32] **Metapath** 𝜌 is defined on the network schema = (𝒜, ) and is denoted as a path in the form of a path. Among them, describes a composite relationship from to , where represents the combination operator on the relation.

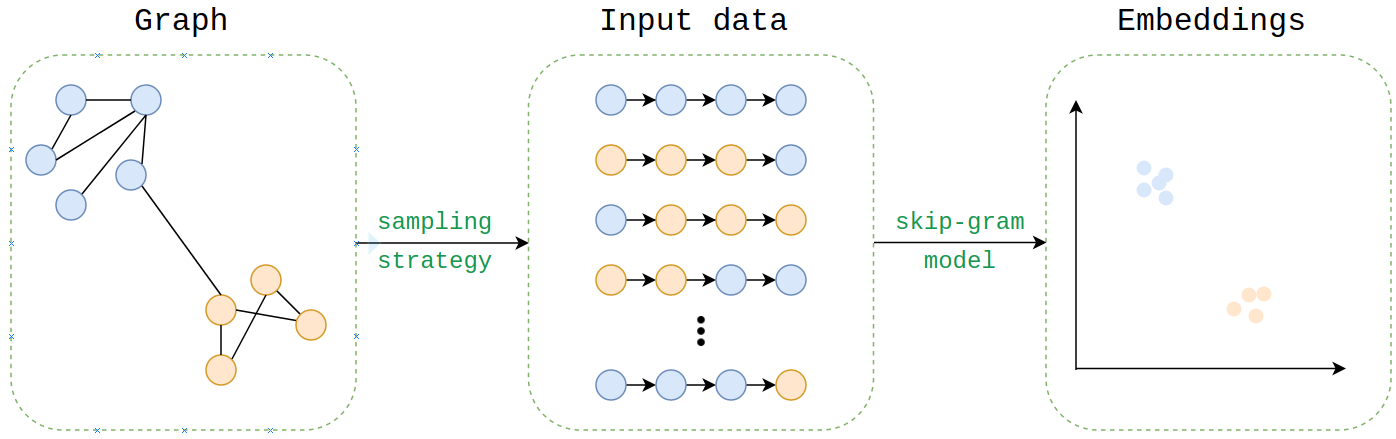
In this sense, considering some of the node’s features as different type of nodes in HIN, one may connect the objects of one type through the metapaths. For example, having an age or location of the user as a distinct type of nodes, it is possible to connect users through those new nodes’ types.

**Deep Walk**

Deep Walk [33] is one of the first algorithms that used word2vec [34] technique of vectorization in graph representation of data. It defines one or several sequence of nodes, which are constructed by random walks with some parameters, as a sentence in word2vec algorithm and then uses skip-gram approach to get vectorized representation of nodes. Basically, every node is found by the unique representation in vector space of the dimensionality much lower than the number of nodes in the graph. Moreover, the connectivity of the vertices between each other is also represented in numbers of vectors, having close vertices vectors like each other. That is usually the requirements for the input in supervised machine learning tasks based on the network data.

**Node2vec**

Node2vec [35] is an algorithm of mapping the nodes of the graph to the embedding space. It is very similar to the Deep Walk, yet has prime differences. The algorithm summarizes the random walks and skip gram theory from word2vec. It firstly generates the sequences of nodes through the random walks’ method. Those sentences are then used to make n-grams and processed by one hidden layer network which outputs the embeddings of the required size. This method shows improvement comparing to the classical heuristics listed above and Deep Walk vectorization because it combined the use of Breadth-first-Sampling and Depth-first-Sampling [35], which basically stimuli the random walk to go to the new parts of the graph or to stay in close distance from the starting point.



*Figure 2 Random walk based algorithms for node embeddings procedure (*[*https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.machinelearningmastery.ru%2Fnode2vec-embeddings-for-graph-data-32a866340fef%2F&psig=AOvVaw30ND3iLqtiMHiXe2zgcCLV&ust=1652705032624000&source=images&cd=vfe&ved=0CA0QjhxqF*](https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.machinelearningmastery.ru%2Fnode2vec-embeddings-for-graph-data-32a866340fef%2F&psig=AOvVaw30ND3iLqtiMHiXe2zgcCLV&ust=1652705032624000&source=images&cd=vfe&ved=0CA0QjhxqF) *)*

* 1. **Used technologies**

All the experiments were conducted using programming interpretation language python 3.6. The following open-source packages were installed and used to perform the experiments.

1. NetworkX [36, 37] is the library with the open-source code provides many solutions to the operations with graph like data structures as well as access to the basic algorithm for simple analysis of those structures.
2. GENSIM [38, 39] is the library with the open-source code provides many solutions to the Natural Language Processing tasks. It also contains pretrained on large corpuses models. The library was used to access the Word2Vec [34] algorithm for word vectorization.
3. Node2vec [35, 40] is the library with the open-source code provides the implementation of the Node2Vec algorithm.
4. LightFM [23, 41] is the library with the open-source code provides the implementation of the LightFM algorithm.
5. Problem standing

The given problem of product recommendation does not consider recommendation of new products not presented on the stage of learning, the data summarizes miserable history of clients’ purchases, and it does not contain the information of clients’ feedback or preferences related to the products in explicit format. Therefore, among other recommendation system problems, the problem can be formalized as ranking problem.

Let’s start with describing the notational framework. denote the set of all users and the set of all item groups. The ratings of on are collected in the matrix , where represents the rating of user assigned to item . Here products with interactions such as purchases, or ordering take higher ratings according to the total spendings of the client on that category. The classical result of ranking algorithm for object is a map that allocates each with the weight .

The goal of the work is the solution of the product groups ranking problem. In other words, it is required to build an algorithm that would provide ratings for each pair user item reflecting the interest of the former in the latter relative to other items. The relative dependency interest aspect will be considered in the description of metrics.

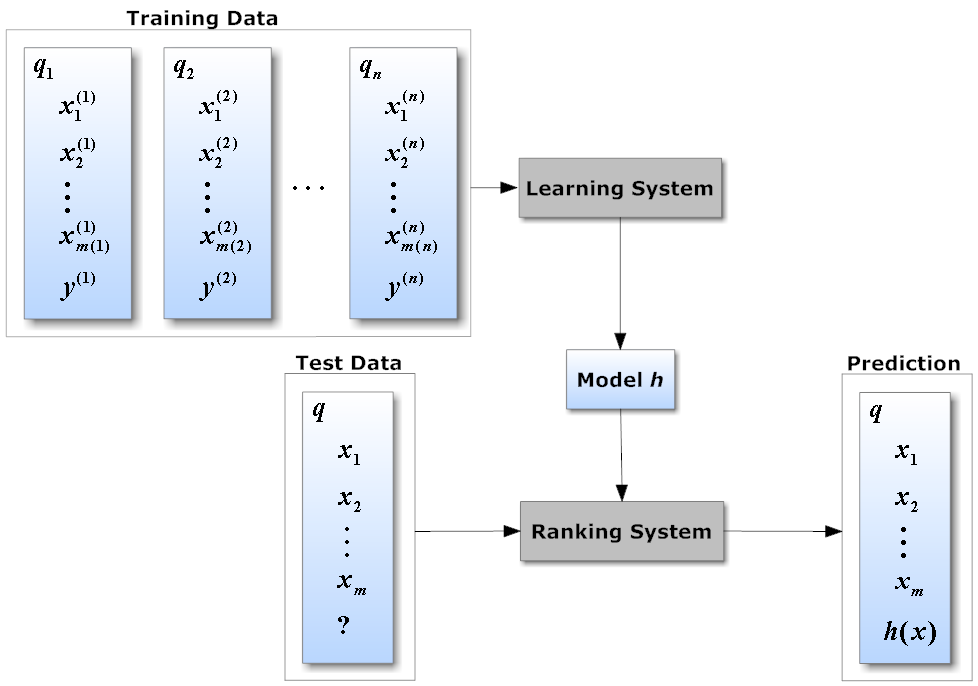


Figure 3 Ranking Problem (<https://medium.com/@nikhilbd/intuitive-explanation-of-learning-to-rank-and-ranknet-lambdarank-and-lambdamart-fe1e17fac418#:~:text=Learning%20to%20Rank%20(LTR)%20is,single%20instance%20at%20a%20time>. )

**Data description**

Изображение выглядит как текст, черный, монитор, снимок экрана

Автоматически созданное описаниеDesigned method is tested on the real transactional dataset of online retail shop amounted in more than raw 530 000 lines (transactions), 130 000 of applicable users and 32 000 of products or 14 product groups that should be ranked. Applicable users are the users that have conducted at least 1 purchase for the train data set and at least 2 for the test data set because it is unfair to use transactional data of the user product purchase that is not yet conducted. Moreover, in the case when there is no previous purchases there are no information about client at all and the problem is converged to top-N prediction. All the purchases were conducted during the 3 months of spring 2016.

Figure 4 Example of initial data

Under the term transaction in the scope of the work is understanded the purchase or one line in the dataset that defines the client which did the purchase and the product which was purchased as well as the features of the former.

The sample of 171 967 transactions with the information about users and product group in online retail store is divided in traditionally applied proportion of (80:20):

* 137 573 transactions – training set
* 34 394 transactions – test set

The distribution in 2 sets undergoes the process of selection users with 2 and more transactions in test set. The following method is used because there are practically no data for the input in those cases. The only algorithm that can work in that case is top-N algorithm, which will be taken as baseline prediction. The use of it in more than 70% cases (proportion of clients with one order only) will complicate the comparison of the result. The predictions are made in a consistent timeline manner meaning that the transaction made in May was not used to predict rank of products in March. Therefore, transactions in the test set are also the latest among the sample.

Analyzed data can be allocated to two types:

* Transactional table like data
* Description of product groups in text format

The latter is collected additionally through manual search of stacked product descriptions that fall into 14 broader groups, which are car accessories, kids’ products, games soft and entertainment, climate, large home appliances, furniture, small home appliances, handmade products, sport and activities, TV and audio, products for home, services, digital technology, accessories, others.

The quality control of the algorithm is conducted on the separated test set, which is not used in the process of the model training.

**Metrics**

Criterions of quality prediction are: Mean average precision at K (map@K), Normalized Discounted Cumulative Gain.

Metric of quality Mean Average Precision at K (map@K) [9] is the metrics that is used to measures the average precision@K averaged over all queries, where one query is a sequence of items’ indexes. The metric does value not only the relevance of the particular item for the user, but also the position in the query of recommended items. The closer an item is to its position the higher is the value of the metric.

Normalized Discounted Cumulative gain [10] is another metric that measures the relevance of the recommended items based on their position in the suggested query for each user. However, it does not consider extra items or missed ones in the query on the last positions. The latter is the reasons why this metric used only as a supportive about the order of those n items. In this paper the main role is playing the map@K.

1. Data collection

The data collection process was conducted in two steps.

* Transactional data collection

The main source of data was the online shop with multiple goods supply. That data was gained directly from the qualification work supervisor in the raw format. It may be accessed through the [link](https://drive.google.com/file/d/1e-gR25qb3fSXBRYffbLIKsgvCTVPpyi5/view?usp=drivesdk), it is also similar to the one presented on the figure 4, but is in the CSV format. The raw data consist of the 530 000 lines of transactions where every line summarizes the information about the purchase action in spring 2016. The sparsity of the data involves the assumption that every retail store is able to collect and provide information of the similar type.

* Group description texts collection

The data that represents 14 product groups through the text description format was collected manually. The need in that data can be justified by the lack of the item’s features, which are required for the content based recommendations. The information was collected through the internet search on the webpages of retail shops characteristics and Wikipedia definitions as an exception. The assumption taken here is that the sellers describe the target product groups in the best possible way. The shortest and the longest description consists of 269 and 2411 words respectively. The raw file may be accessed by the [link](https://github.com/Lina-moon/Diplom_CS_2022_DSBA_Andreev_Mikhail).

1. Design and results
2. Data preprocessing

The processing of the data preprocess included dealing with missing values, restructuring, visualization, deleting outliers, extracting features, defining uncorrelated features, building clusters of users.

Some key steps will be outlines in this section defining the process of data preprocessing. All the data with missing key information was eliminated. Under the term key here stands the information about the purchase date, price, order number, location, category of the item, payment type and shipment. The values of product type were mostly recovered based on the higher level groups description. The identification of the user was worked around through coded phone number to save the anonymous information of customers. The data was then restructured according to the user item relation, having several lines with the same user number corresponding to several purchases of the user. The outliers were deleted on the rule of 0.99 and 0.01 percentile along all the features. Visualization of the categorical as well as numerical features helped to highlight meaningless features straightforwardly by visual analysis. After constructing most of the important features their number was 42. By the following correlation analysis only 10 of them were left on the basis of exceeding 0.4 Pearson correlation threshold. The list of the initial features can be accessed through the [link](https://github.com/Lina-moon/Diplom_CS_2022_DSBA_Andreev_Mikhail) to the project’s github. The 10 left features are listed in the next section of this paper.

1. Plan of the work

The following methods were used in the process of building the recommendation system:

1. Straightforward approaches

In this section the justified and reasonable baselines were produced, having all the methods easily repeated and fast in realization. Those would be considered as the first attempts to the problem. Those are: Top-N, kNN on user data, Logistic regression, SGD, Random Forest.

All those algorithms provide the probabilities of existing connection between a particular user and one of the 14 item groups. Practically speaking, at this stage the problem of multiclass classification is solved. The probabilities are summed for every user along its purchases. The features that are used for the predictions are as follows: Number of bills for the user, Mean number of bills for the user, Mean number of items in bill for the user, Number of shipments to user, Percentage of bills paid by cash, destination region (a categorical feature coded by one-hot encoded central, privolzie, siberia, southern, ural regions of Russia).

Table 1 Results of straightforward approaches on table like data.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method \ Metric** | **Map@5** | **Map@10** | **NDCG@5** | **NDCG@10** |
| Top-N | 0.39 | 0.44 | 0.50 | 0.42 |
| kNN | 0.29 | 0.29 | 0.46 | 0.41 |
| Log-reg | 0.36 | 0.38 | 0.45 | 0.42 |
| SGD | 0.39 | 0.44 | **0.54** | **0.47** |
| Random Forest | **0.41** | **0.45** | 0.52 | 0.44 |

1. Link prediction methods based on graph features

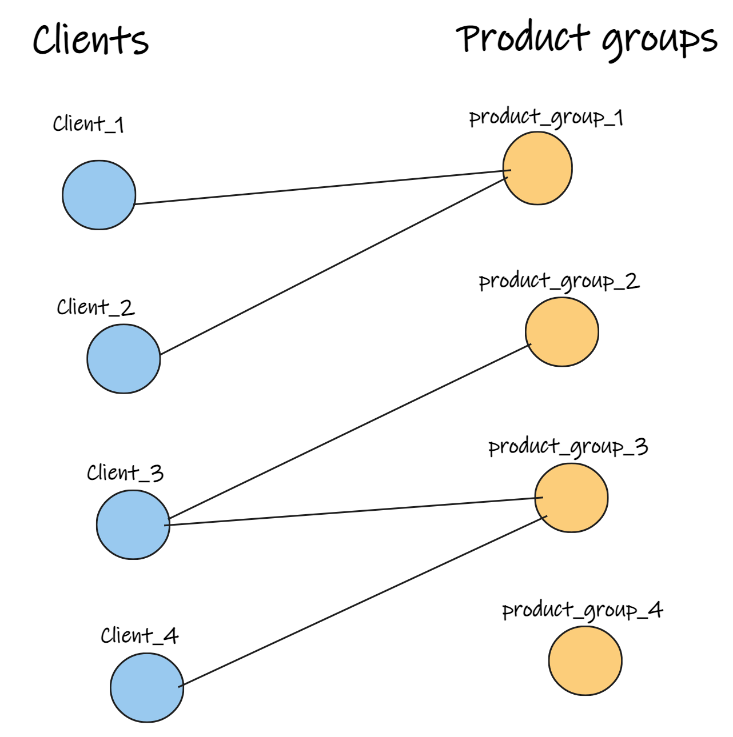
In this section one may find algorithms that require graph structure of the data. Those are: Common Neighbors, Jaccard Coefficient, Adamic Adar, Preferential Attachment. Based on the calculated heuristics between each two nodes of interest the thresholds are chosen and used on the inference stage. All methods are based on the graph heuristics that are very sensitive to the graph structure. Therefore, it is required to define the conversion of tablelike data to the graph.

Figure 5. Bipartite graph schema

* Bipartite graph structure

In this case, the nodes set consists of two entities: item groups and users. The edges may be found only between the edges of the different node types. One may find the schema with 4 users and 4 product groups in figure 5. Here can be seen the problem calculating the graph statistics based on the connection density. There are few paths of length 3 which starts from client nodes (on the figure 4 there is just one such path), yet most of those heuristics require the number of neighbors at length 3.

Table 2 Results of graph heuristics on bipartite data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method \ Metric** | **Map@5** | **Map@10** | **NDCG@5** | **NDCG@10** |
| Common Neighbors | 0.06 | 0.07 | 0.15 | 0.09 |
| Jaccard Coefficient | 0.03 | 0.06 | 0.07 | 0.04 |
| Adamic Adar | 0.06 | 0.07 | 0.22 | 0.16 |
| Preferential Attachment | **0.14** | **0.17** | **0.30** | **0.23** |
| Preferential Attachment\_ui | 0.07 | 0.09 | 0.17 | 0.10 |

* Heterogenous graph structure with multiple types of nodes

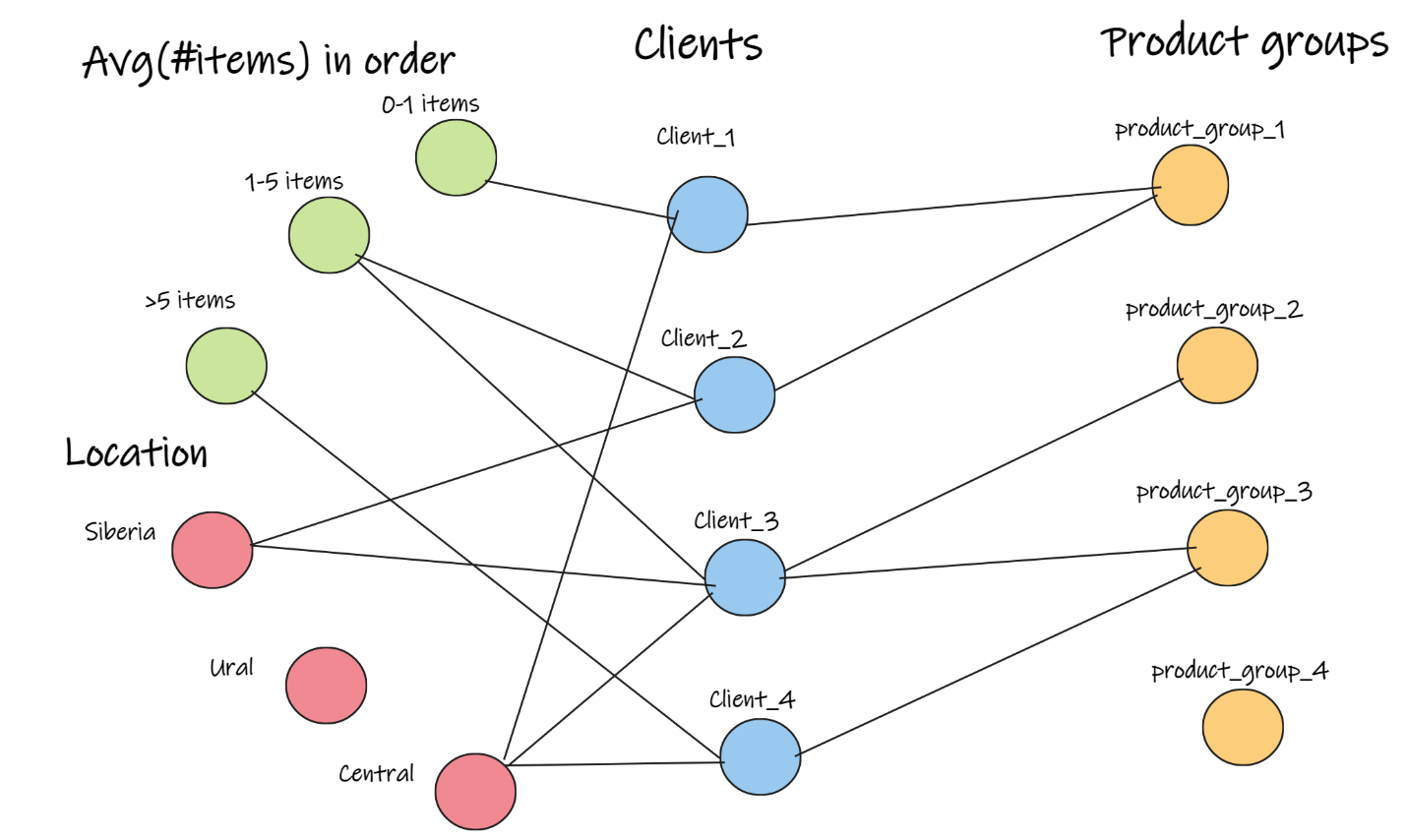
This representation of the data is driven by the will to include user features into the graph by establishing new node types. Those are provided through the information about the users. The troubles may occur with numerical features, yet they may be represented through several bins. The number of bins corresponds to the number of user clusters found on the data preprocessing step and is 5. One may find the representation of that structure on the Figure 6. In this case, by adding few new nodes, the connectivity of the graph increases. At first in the case of bipartite graph only 0.002% of the possible edges existed, while after extracting the attributes that value is 0.017%.

Figure 6 Heterogenous graph structure with multiple types of nodes schema

Figure 7 Heterogenous graph structure with multiple types of nodes schema

Table 3 Results of graph heuristics on heterogeneous data

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method \ Metric** | **Map@5** | **Map@10** | **NDCG@5** | **NDCG@10** |
| Common Neighbors | 0.25 | 0.27 | **0.45** | **0.36** |
| Jaccard Coefficient | 0.11 | 0.14 | 0.29 | 0.23 |
| Adamic Adar | 0.12 | 0.17 | 0.22 | 0.16 |
| Preferential Attachment | **0.31** | **0.32** | 0.40 | 0.34 |
| Preferential Attachment\_ui | 0.16 | 0.21 | 0.33 | 0.30 |

Along with the known metrics were tested several custom heuristics. However only the Preferential Attachment on the users showed comparable result. That metric calculates the intersection of only users at length 3 instead of intersection of all nodes.

1. Link prediction methods with embeddings

In this section the algorithms for node embeddings on the graph are used to vectorize the entities before classification. Those are DeepWalks, Node2vec and their modification. Moreover, in this section they are compared with the LightFM solution from the box that utilizes the linear combinations of the user and item features.

All the methods presented were tested on the heterogeneous graphs with extracted attributes. The DeepWalk algorithm was used with following hyperparameters: window\_size: 5, embedding\_size: 128, walk\_length: 20, walks\_per\_node: 10.

The Node2vec model was trained with following hyperparameters: embedding\_dim=128, walk\_length=20, context\_size=5, walks\_per\_node=10, num\_negative\_samples=1, p=4, q=1.25.

At the same time LightFM was used with default hyperparameters: no\_components=10, k=5, n=10, learning\_schedule='adagrad', loss='logistic', learning\_rate=0.05, rho=0.95, epsilon=1e-06, item\_alpha=0.0, user\_alpha=0.0, max\_sampled=10, random\_state=None.

The modified version of Node2vec utilizes the concept of and hyperparameters that affect the direction of the random paths development. It also implies the knowledge about target metapaths which are in our case User AttributeUser Item Group (UAUI), having started from the User, and Item Group User Attribute User (IUAU) starting from the Item Group. That requires observing several case scenarios

* Start in User1
  + Jumping to Item Group is not plausible, therefore the probability of such action is set to 0.
  + Jumping to one of the connected Attribute is exactly what is required, therefore, probability is 1.
* Start in Attributei
  + Jumping back to the User1 node is meaningless, therefore its probability is 0.
  + Jumping to another connected User2 node will stimuli the deep walk through the graph, therefore the probability of doing that is .
  + Jumping to another visited User is ambiguous, therefore the probability is set to .
* Start in User2
  + Jumping to the Attributei is not plausible, hence, the probability is 0.
  + Jumping to Item group is in target, therefore the probability is set to 1.
* Start in Item Group
  + Jumping to visited User node will stimuli the broader search, therefore the probability is set to .
  + Jumping to unvisited new User node will stimuli the deeper search, therefore the probability is set to .

Table 5 Results of embedding algorithms

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method \ Metric** | **Map@5** | **Map@10** | **NDCG@5** | **NDCG@10** |
| Deep Walk + RF | 0.64 | 0.69 | 0.75 | 0.68 |
| Deep Walk + SGD | 0.63 | 0.66 | 0.81 | 0.75 |
| Node2Vec + RF | **0.74** | **0.77** | 0.89 | 0.80 |
| Node2Vec + SGD | 0.69 | 0.75 | **0.95** | **0.90** |
| LightFm Logreg | 0.53 | 0.55 | 0.67 | 0.61 |
| Node2Vec + Metapath rule + RF | 0.65 | 0.70 | 0.68 | 0.60 |
| Node2Vec + Metapath rule + SGD | 0.62 | 0.66 | 0.72 | 0.63 |

1. Evaluation and Conclusion

In this work the comparison of several methods for recommendation system was performed. The best map@5 and map@10 metric values are reached using Random Forest classifier as link prediction algorithm on the data converted to node2vec embeddings build on heterogeneous graph data modified by the text vectorization through cosine similarity and are 0.74 and 0.77 respectively.

The conducted research on the cold start in recommendation system using link prediction methods has shown that the problem of the cold start may be successfully approached through the graph representation of the data. Moreover, extracting the node attributes for increasing the connectivity of the graph, the methods of natural language processing, and embedding of the graphs can increase the map@5 metric value by more than 0.3. The results of improved Node2vec using metapaths rule are less than original, yet they are on the same level as Deep Walk embeddings. Those observations make the proposed embedding method compatible with complex algorithms and at the same time better than simple solution from the box like LightFM algorithm which doesn’t require the thin adjustments.

The results of the research support the idea of solving recommendation problem tasks trough link prediction techniques. It also promotes the representation of sparse data in the form of graphs for extracting new information. The reached model may be improved and used in further researches and medium size retail stores in order to improve the customers’ experience from buying online.

1. Bibliography:
2. DIGITAL 2022: ANOTHER YEAR OF BUMPER GROWTH - We Are Social UK [Electronic resource]. URL: https://wearesocial.com/uk/blog/2022/01/digital-2022-another-year-of-bumper-growth-2/ (accessed: 21.04.2022).
3. Ecommerce Market Share (2019–2025) [Feb 2022 Upd] | Oberlo [Electronic resource]. URL: <https://www.oberlo.com/statistics/ecommerce-share-of-retail-sales> (accessed: 21.04.2022).
4. Chen Y.-L., Chen J.-M., Tung C.-W. A data mining approach for retail knowledge discovery with consideration of the effect of shelf-space adjacency on sales // Decision Support Systems. 2006. Vol. 42, № 3. P. 1503–1520.
5. Knijnenburg B. et al. Explaining the user experience of recommender systems // User Modeling and User-Adapted Interaction. 2012. Vol. 22. P. 441–504.
6. Geyser W. Coronavirus (COVID-19) Marketing & Ad Spend Report (July 2021) [Electronic resource] // Influencer Marketing Hub. 2020. URL: https://influencermarketinghub.com/coronavirus-marketing-ad-spend-report/ (accessed: 21.04.2022).
7. He X. et al. Neural Collaborative Filtering // arXiv:1708.05031 [cs]. 2017.
8. Hidasi B. et al. Session-based Recommendations with Recurrent Neural Networks // arXiv:1511.06939 [cs]. 2016.
9. Cheng H.-T. et al. Wide & Deep Learning for Recommender Systems // arXiv:1606.07792 [cs, stat]. 2016.
10. Yohanandan S. mAP (mean Average Precision) might confuse you! [Electronic resource] // Medium. 2020. URL: https://towardsdatascience.com/map-mean-average-precision-might-confuse-you-5956f1bfa9e2 (accessed: 26.04.2022).
11. Järvelin K., Kekäläinen J. Cumulated gain-based evaluation of IR techniques // ACM Trans. Inf. Syst. 2002. Vol. 20, № 4. P. 422–446.
12. Goldberg D. et al. Using collaborative filtering to weave an information tapestry. P. 10.
13. Dervishaj E., Cremonesi P. GAN-based Matrix Factorization for Recommender Systems // arXiv:2201.08042 [cs]. 2022.
14. Maurera F.B.P., Dacrema M.F., Cremonesi P. An Evaluation Study of Generative Adversarial Networks for Collaborative Filtering // arXiv:2201.01815 [cs]. 2022.
15. Quora. How Did Spotify Get So Good At Machine Learning? [Electronic resource] // Forbes. URL: https://www.forbes.com/sites/quora/2017/02/20/how-did-spotify-get-so-good-at-machine-learning/ (accessed: 26.04.2022).
16. Linden G., Smith B., York J. Amazon.com recommendations: item-to-item collaborative filtering // IEEE Internet Comput. 2003. Vol. 7, № 1. P. 76–80.
17. Engineering P. Applying deep learning to Related Pins [Electronic resource] // The Graph. 2017. URL: https://medium.com/the-graph/applying-deep-learning-to-related-pins-a6fee3c92f5e (accessed: 26.04.2022).
18. Bhagavatula C. et al. Content-Based Citation Recommendation // arXiv:1802.08301 [cs]. 2018.
19. Gopalan P.K., Charlin L., Blei D. Content-based recommendations with Poisson factorization // Advances in Neural Information Processing Systems. Curran Associates, Inc., 2014. Vol. 27.
20. Philip S., Shola P., Abari O. Application of Content-Based Approach in Research Paper Recommendation System for a Digital Library // International Journal of Advanced Computer Science and Applications. 2014. Vol. 5.
21. Abdulghani A., Bayoumi S., Al-Yahya H. A Recommender System for Mobile Applications of Google Play Store // International Journal of Advanced Computer Science and Applications. 2020. Vol. 11.
22. In-Depth Guide: How Recommender Systems Work | Built In [Electronic resource]. URL: https://builtin.com/data-science/recommender-systems (accessed: 26.04.2022).
23. Ekstrand M.D. Collaborative Filtering Recommender Systems // FNT in Human–Computer Interaction. 2011. Vol. 4, № 2. P. 81–173.
24. FROLOV E. LOW RANK MODELS FOR RECOMMENDER LOW RANK MODELS FOR RECOMMENDER SYSTEMS WITH LIMITED PREFERENCE INFORMATION DOCTORAL PROGRAM IN COMPUTATIONAL AND DATA. Moscow: Skoltech, 2018. 197 p.
25. Kula M. Metadata Embeddings for User and Item Cold-start Recommendations // arXiv:1507.08439 [cs]. 2015.
26. Johnson C.C. Logistic Matrix Factorization for Implicit Feedback Data. P. 9.
27. Tibshirani S., Friedman H. Valerie and Patrick Hastie. P. 463–475.
28. Cremonesi P., Koren Y., Turrin R. Performance of recommender algorithms on top-N recommendation tasks. 2010. P. 39–46.
29. Lakshmi T.J., Bhavani S.D. Link Prediction Approach to Recommender Systems // arXiv:2102.09185 [cs]. 2021.
30. Lakshmi T., Surampudi D. Link Prediction Measures in Various Types of Information Networks: A Review. 2018. P. 1167.
31. Shao Y., Liu C. H2Rec: Homogeneous and Heterogeneous Network Embedding Fusion for Social Recommendation // International Journal of Computational Intelligence Systems. 2021. Vol. 14.
32. Shi C. et al. Semantic Path based Personalized Recommendation on Weighted Heterogeneous Information Networks // Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. New York, NY, USA: Association for Computing Machinery, 2015. P. 453–462.
33. Sun Y. et al. PathSim: meta path-based top-K similarity search in heterogeneous information networks // Proc. VLDB Endow. 2011. Vol. 4, № 11. P. 992–1003.
34. Perozzi B., Al-Rfou R., Skiena S. DeepWalk: Online Learning of Social Representations // Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. 2014. P. 701–710.
35. Mikolov T. et al. Efficient Estimation of Word Representations in Vector Space: arXiv:1301.3781. arXiv, 2013.
36. Grover A., Leskovec J. node2vec: Scalable Feature Learning for Networks: arXiv:1607.00653. arXiv, 2016.
37. NetworkX [Electronic resource]. 2022. URL: https://github.com/networkx/networkx (accessed: 19.05.2022).
38. NetworkX — NetworkX documentation [Electronic resource]. URL: https://networkx.org/ (accessed: 19.05.2022).
39. Gensim: topic modelling for humans [Electronic resource]. URL: https://radimrehurek.com/gensim/models/word2vec.html (accessed: 19.05.2022).
40. R Rehr Uv Rek, P Sojka. Software Framework for Topic Modelling with Large Corpora. Unpublished, 2010.
41. eliorc/node2vec: Implementation of the node2vec algorithm. [Electronic resource]. URL: https://github.com/eliorc/node2vec (accessed: 19.05.2022).
42. lyst/lightfm: A Python implementation of LightFM, a hybrid recommendation algorithm. [Electronic resource]. URL: https://github.com/lyst/lightfm (accessed: 19.05.2022).
43. RecSys – ACM Recommender Systems [Electronic resource]. URL: https://recsys.acm.org/ (accessed: 20.05.2022).