Автономная некоммерческая организация высшего образования
«Университет Иннополис»

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(БАКАЛАВРСКАЯ РАБОТА)
по направлению подготовки
09.03.01 - «Информатика и вычислительная техника»

GRADUATION THESIS
(BACHELOR'S GRADUATION THESIS)
Field of Study
09.03.01 – «Computer Science»

Направленность (профиль) образовательной программы
«Информатика и вычислительная техника»
Area of Specialization / Academic Program Title:
«Computer Science»

Тема / Topic	**Использование ИИ-алгоритмов в бизнес-процессах индустриальных компаний / Implementation AI algorithms in business processes of industrial companies**

Работу выполнил / Thesis is executed by	**Будник Георгий Васильевич / Budnik Georgii Vasilevich**	подпись / signature
Руководитель выпускной квалификационной работы / Supervisor of Graduation Thesis	**Лукманов Рустам Абубакирович / Lukmanov Rustam Abubakirovich**	подпись / signature

Иннополис, Innopolis, 2025

Contents

List of Tables

List of Figures

Abstract

This thesis examines the ways of digital transformation of companies in the industrial sector using Artificial Intelligence (AI) algorithms and without its participation. The paper considers the optimization of two key indicators: the time when a customer's order was placed by an employee of the company (Customer Checkout time) and the average number of product items in the order (SKU).

To speed up the customer's order processing time, it is proposed to implement an updated business process for barcode design, which will replace the printing of documents for issuing and paying for goods. To capture the average number of SKUs in a customer's order, we consider using an auxiliary recommendation system for sales consultants. The study includes a comparison of recommendation systems based on classical methods and deep learning, as well as analysis and representation of input data.

All experiments and data analysis were carried out within the framework of the largest construction company in the Far East. The final thesis consists of 71 pages, including 22 figures, 2 tables and 40 sources used.

Chapter 1

Introduction

1.1 Relevance of the Topic

The implementation of artificial intelligence (AI) in the business processes of industrial companies has become a vital and irreversible trend. Industry—including manufacturing, construction, and extraction—contributes approximately 25% to the global GDP, making it the second-largest economic sector after services [1].

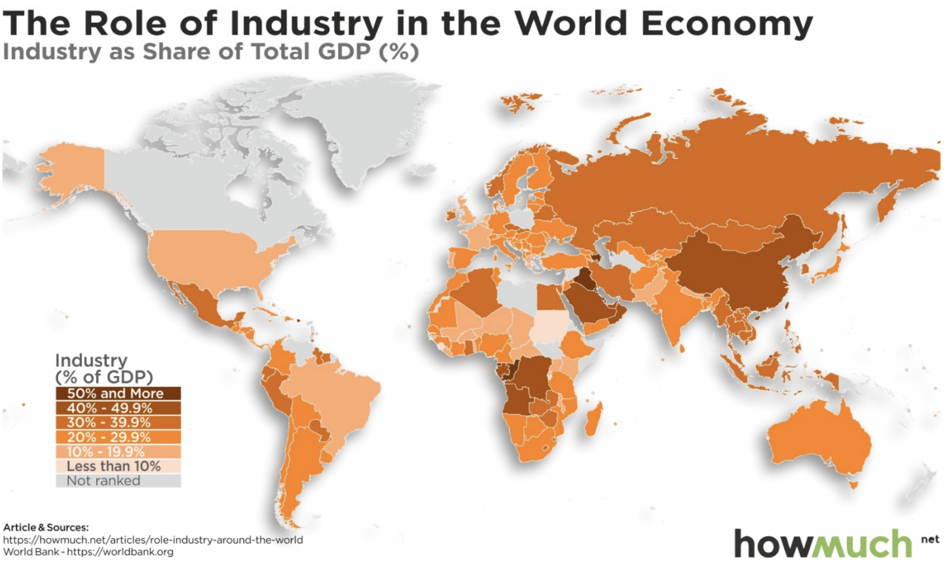


Fig. 1.1. Visualizing the Importance of the Industrial Economy in The World [2]

Despite a slight decrease in industrial employment (from 23.2% in 2014 to 22.9%), the demand for cost reduction, efficiency, and adaptability continues to grow. This makes AI not just an option but a necessity for optimizing production, logistics, and customer interactions in large-scale industrial operations.

Industrial operations encompass not only production and logistics but also customer service and material sales, which play a foundational role in shaping supply chains and end-market dynamics. The availability and quality of industrial products depend on how goods are distributed, marketed, and purchased by businesses and consumers.

Digital technologies and the integration of AI offer tremendous potential for optimizing key indicators that are embedded in the digital infrastructure of industrial retail and distribution systems. These indicators include:

- **Customer's heckout time** - time it took the store employee to place an order via a computer
- **The number of SKU (stock keeping unit) in the customer's order** - the number of items from the stores assortment

During the past year, I served as the Head of Digital Transformation at the largest chain of construction retail stores in the Russian Far East. This position provided me with a unique opportunity to study the transformation process from within a real, high-volume commercial organization operating in the complex landscape of construction logistics and customer service.

My primary responsibilities included diagnosing inefficient business processes, automating routine operations, designing digital solutions, and implementing IT initiatives across departments with virtually zero initial digital awareness among staff. The insights and prototypes developed in this context form the empirical foundation of this thesis. All solutions discussed in this study have been tested in a real environment and validated by operational performance.

Since the work is of commercial interest to an existing industrial company, part of the input data was encrypted. All materials used, developments, received materials and results are demonstrated in our public repository:

https://github.com/Goshmar/My-BS-Thesis

1.2 Research Objectives

The objectives of this study is to optimize key indicators that affect the revenue of a network of retail stores of building materials through the analysis, development and implementation of IT solutions.

Each of the indicators has its own relevant research question (RQ):

- **RQ1:** How much can the customer's checkout time be reduced by barcode card mechanism integration?
- **RQ2:** How much can the number of SKUs in a customer's order be increased by implementing an auxiliary recommendation system for sales consultants?

To test the hypotheses according to these research questions, the tasks of the work were formed:

1. Explore the market for the best research and IT solutions that have successfully introduced and optimized key applications
2. Collect and pre-process input data from a Personal computer (PC) of a retail chain of construction stores, preparing datasets for data analysis

3. Develop an IT solution within the resources, constraints and requirements of the company’s management

4. Test and implement the developed IT solutions

5. Evaluate the impact of new IT solutions

Chapter 2

Literature Review

This chapter provides an overview of existing solutions in the industrial sector. Section 2.1 reflects the study of a key stage in the digitalization of companies. Section 2.2 reviews the literature and presents more applied solutions considering the potential of using barcode mechanisms. Section 2.3 contains the trend of personalization of the environment of existing solutions and reviews recommendation systems in the field of this work.

2.1 Digital Transformation in the Industrial Retail

The sale of goods in the industrial sector has historically been dominated by manual work processes and store operations. In this article, this evolution is driven by economic pressures, customer expectations, and the availability of modern digital technologies [3]. The authors formulate this conclusion based on that over time, customers expect businesses to take an increasingly simplified approach to making purchases. This strategy increases customer loyalty, which directly affects our key indicators.

Enterprise resource planning (ERP) systems have become the foundation for digitalizing business processes in Russian industrial companies, with 1C:ERP being used by over 45% of all ERP-adopting firms [4], [5]. The company examined in this thesis also operates on a 1C-based infrastructure, which enabled the study of both RQ1 and RQ2 within a real ERP environment.

2.2 RQ1 - Customer's checkout time optimization

2.2.1 Existing modern methods

There are many research areas that allow us to consider customer optimization from a scientific point of view [6]–[8]. The general approach to solving optimization is to develop some kind of platform, service, or technology that is an intermediate link between the store and the customer, relieving the store employee of the burden on ERP systems.

Implementation of this solution is shown in the example of an Indonesian supermarket in the following article [9]. A system has been developed to reduce the cashier's work time by independently assembling the customer's smart shopping cart order:

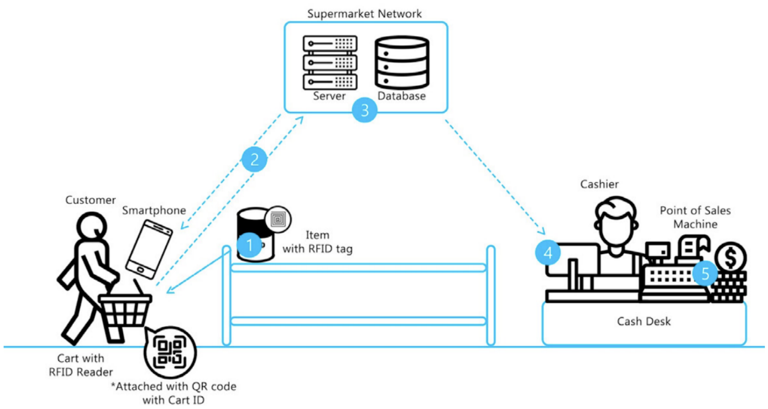


Fig. 2.1. Smart shopping schema

Each product has a radio frequency identification (RFID) tag (Step 1), which the buyer applied to the smart shopping cart receiver, which sent a request to the server, saving the total cost of the order (Step 2). When the customer arrives at the checkout, he dictates the order code to the cashier, who uses code to check all transactions for correctness (Step 4). After making the payment, the cashier confirms or rejects the completion of the order, prints the receipt (Step 5).

When building such a process, optimizing Customer's checkout time in a store based on experiments amounted to a reduction of 5 minutes for 10 people in a queue (each have 10 products) and 10 minutes for 20 products.

It is not only optimization on the user's device side that allows to achieve the desired result. The article [6] shows how oil specialists have reduced the time required to fill out a customer feedback form. By changing the patterns and interface, it was possible to achieve a fivefold reduction in the filling time (from 25 minutes to 5 minutes).

Another approach to optimizing the customer's path is self-checkout system. As mentioned in this article [10], self-service cash registers have moved from innovation to widespread use of stores in 5 years. This reduces the cashier's workload, but this approach has many disadvantages, such that such systems often fail in stores. According to research, the main causes of the failure are user sloppiness and failures of the central server Internet connection.

Successful experiments in optimizing the customer's path through self-checkout systems include a modification with YOLOv10 [11]. Unlike software using C2f-Dual convolutional design, which uses dual convolutional cores (Dual-Conv), computational costs are significantly reduced and the number of parameters, as well as the accuracy increases:

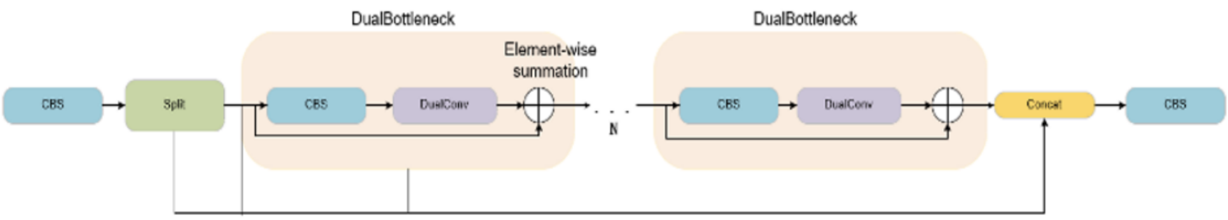


Fig. 2.2. Dual Convolutional C2f-Dual Design

Different cores in the group now process the same data set in parallel, optimizing the flow of information and feature extraction. In this way, the model has a better representation of the gradient flow, reducing noise during training.

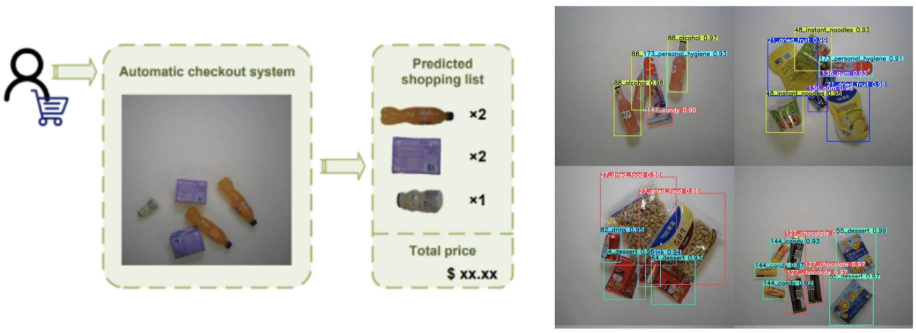


Fig. 2.3. YOLOv10 Results

Another example of technically complex automation in retail trade is showed in this source [12]. International clothing retailer "Uniqlo", due to the use of intelligent bins with an integrated RFID reader:

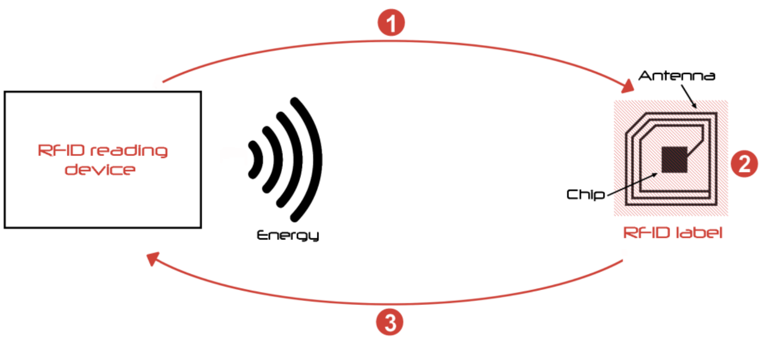


Fig. 2.4. How RFID label works?

Device scans special clothing labels with unique codes, was able to speed up the checkout process (Step 1). Instead of going to the checkout, the customer throws his clothes into a bin can with a special antenna (Step 2). After that program, based on chip's logs, offers to pay for the goods (Step 3).

The manufacturer of these smart devices, Fast Retailing [13], such automation has reduced checkout time by about 50% and queues at physical cash registers.

2.2.2 Existing Barcode mechanisms integration

Despite the different approaches to optimizing the customer's route, barcoding is recognized as the best technology in the industrial sector [14]. The principle of barcoding elements is to identify the elements according to the encoded principle. The simplicity and interpretability of the results of such models causes demand for the use of barcoding.

The international article number is considered to be the European Article Number (EAN) type [15]. EAN-13 is a 13-digit barcode standard (12 data and 1 check), which is an addition to the original 12-digit Universal Product Code (UPC) system:

Fig. 2.5. Typical EAN-13 Barcode

As shown in the figure above, EAN is 13 and is divided into special groups of digits that classify information. A technically necessary element is a "Check Sum", which is created based on the following algorithm using the main specific fields:

Listing 2.1: Calculate ChecksumDigit pseudocode

```
Function CalculateChecksumDigit(sTemp):
    iSum = 0
    n = Length of sTemp

    For i from n down to 1:
        iDigit = Integer value of sTemp[i - 1]

        If i mod 2 == 0:
            iSum = iSum + (iDigit * 3)  // odd position
        Else:
            iSum = iSum + iDigit        // even position

    Return (10 - (iSum mod 10)) mod 10
```

Barcodes have a wide range in manufacturing-like areas field due to their flexibility and variability of application. For example, in this research [16] the authors have developed a full-fledged budget accounting application for a construction project based on barcode's tracking building materials. The barcode represent as ID in the general database of building materials, storing the values of the construction object, date, cost and other data that allows to analyze.

There are several more works in which IT solutions have been developed to optimize the customer's checkout time through the implementation of the barcode cards mechanism [17], [18]. The above studies arrive at a single pattern of barcode implementation by implementing the following Data Analysis Schema:

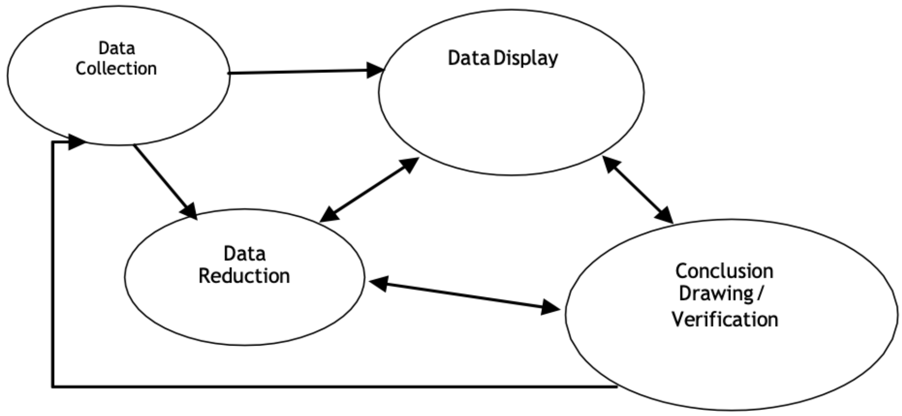


Fig. 2.6. Data Analysis Schema of Barcode cards implementation

Data collection reflects the algorithms for creating and linking the barcode ID to the item. After successful linking, it is required to notify user about it. Conclusion Drawing / Verification can be interpreted as the target event that was the reason for the introduction of the new ID reflects. If the ID is not needed, it is automatically cleaned.

2.3 RQ2 – Increasing number of SKUs

2.3.1 Existing modern methods

There are many studies on various ways to increase the SKU in a customer's order within retail stores. Scientific research [19], [20] highlights the general trend of digitalization, in which personalization is the main modification.

The personalization format is equally important. The article "Recommendation Systems: Algorithms, Challenges, Metrics, and Business Opportunities" [21] demonstrates the variety of forms of personalization. The authors emphasize the limitations of personalization for more effective improvement:

- **Business restrictions**: when recommendations do not match with the general ethical standards or principles of the company

- **Potential problems during training and testing**: when our recommendation does not carry value. For example, when we recommend the shopping package to everyone (the recommendation loses the property of personalization)

- **Field conditions**: In the case of building materials recommendation, we do not need a high accuracy of correct answers (compared to medical recommendations)

An authoritative article [19] highlights that the best way to apply personalization within the framework of the company's key indicators is through recommendation systems. It is emphasized that segmented recommendations more often achieve high results than classical approaches.

Scientists from the scientific community of NITYA publications believe that personalized actions are the best way for the including increasing the number of SKUs for sale [22]. Studies have been conducted comparing approaches without/with the use of AI. The results showed that the use of personalized product recommendations based on AI significantly increased customer loyalty, which led to increased conversions and customer loyalty rates [23].

2.3.2 Recommendation systems

Recommendation system is a Machine learning (ML) model designed to predict user preferences by analyzing past interactions between users and items. These interactions are commonly represented as a user-item interaction matrix, which records user explicit or implicit feedback on various products. Classical approaches can be categorized into user-based, item-based, and hybrid models:

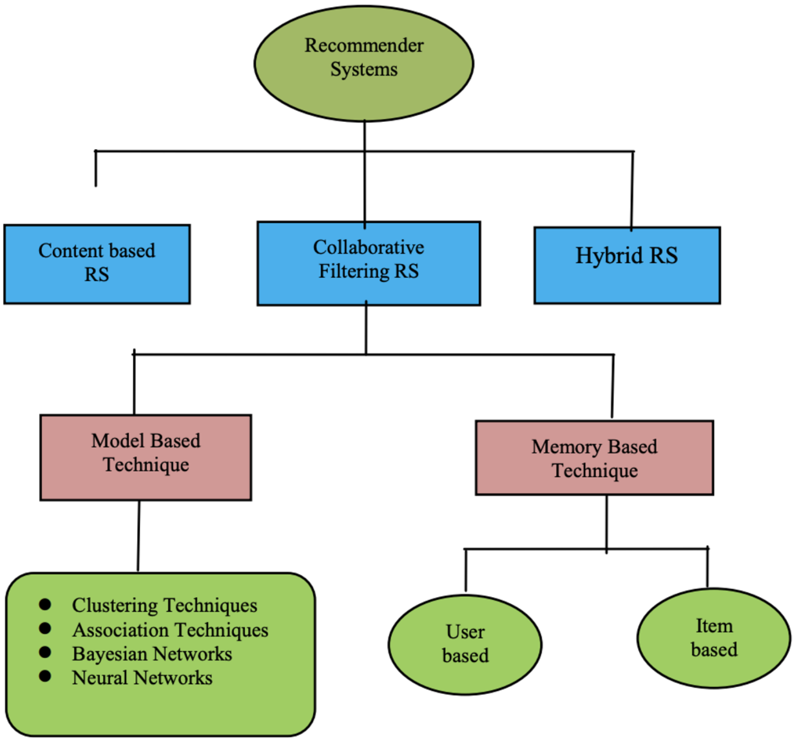


Fig. 2.7. Classification of Recommendation System [24]

Content-Based (CB) focuses on item attributes and builds user profiles from previously rated or interacted items. Similarity is measured using textual features, like TF-IDF (Term Frequency Inverse Document Frequency) or metadata, with ML models like decision trees or neural networks.

Collaborative Filtering (CF) leverages patterns in user-item interactions. Memory-based methods rely on similarity between users or items, while model-based methods use techniques like matrix factorization or neural architectures to learn latent factors. CF assumes that similar users prefer similar items, making it effective when user behavior data is abundant.

Hybrid Approaches combine CF and CB to downscale limitations, such as the cold-start and sparsity problems [25]. Methods include weighted combinations, switching mechanisms, data augmentation, and meta-level architectures.

Deep Learning-Based Models have recently transformed recommendation systems:

- Sequential recommendation models based on transformer encoders using bidirectional context to better capture item dependencies [26]. Unlike traditional unidirectional models, they condition each item on both past and future interactions, significantly enhancing the expressive power of learned representations [27].

- Autoencoders techniques [28] compress user-item interactions into dense representations, enabling dimensionality reduction and reconstruction-based recommendation.

- Generative Adversarial Networks [29] enhance recommendations by generating synthetic user-item pairs, though they face challenges like training instability.

In deep hybrid systems, features learned from content and collaborative components are fused into a single predictive model, improving robustness against data sparsity and evolving user preferences. Such hybrid models are more often used in the development of business solutions, as they can be considered using a balanced assessment of various personalization from different points of view. Due to the weights, the results of such models can be interpreted.

Chapter 3

Methodology

The chapter describes the problems of the state, the conditions for conducting experiments and the methods and tools used to solve RQ1 and RQ2, formulated in Chapter 1. Section 3.1 includes the formulation of the RQ1 problem, methods of its investigation for the general path of the client and the refinement of the ERP solution, a description of the date and methods of its analysis. Section 3.2 includes the formulation of the RQ2 problem, the description of the data and their representation, and the model architectures used. Section 3.3 describes the technical tools and libraries used to implement the methods.

3.1 RQ1 - Barcode card mechanism integration

3.1.1 Problem Statements

This research is conducted in the environment of the largest construction retail chain in the Russian Far East, known as ”Pomoshcnik”. Real data will allow you to verify and evaluate the overall financial effectiveness of integration. On the other hand, the optimization of existing business processes contains many nuances from related areas for successful implementation.

Retail in Vladivostok consists of more than 12 stores, which are united by a single database. sales and data access are carried out through the 1C:ERP program.

The overall **Customer Checkout Time** is composed of the following three stages:

$$t_{\text{checkout}} = t_{\text{order}} + t_{\text{services}} + t_{\text{completion}}, \tag{3.1}$$

where:

- t_{order} is the time spent by the sales consultant to register the products in the ERP system
- t_{services} is the time for configuring and printing optional services (e.g., delivery, loyalty programs, additional packaging)
- $t_{\text{completion}}$ is the time for the customer to reach the cashier, queue, and complete the payment

Each store employs two types of 1C users: sales consultants and cashiers. Each user has access to:

- Desktop PC with a dedicated user interface
- Barcode scanner for item input
- Document printer (sales consultant) and a receipt printer (cashier)
- Payment terminal (cashier only)

From a physical standpoint, store locations vary greatly. Each store has a customer service area of size (in square meters), which affects average customer traffic and movement time. Stores may consist second floor, where goods can be collected from one floor and paid for on another.

The type of customer especially affects the Customer Checkout time. Clients can represent a individual consumer, or they can represent a corporate consumer type. Corporate clients require extended registration procedures in the ERP system, increasing the t_{services} component.

Customer behavior also varies:

- Smaller stores with limited stock often require post-payment service adjustments
- Suburban stores located near construction zones primarily serve professionals who require minimal assistance
- Central city stores typically serve non-professionals who rely heavily on sales consultations (e.g., color matching, material selection)

Let $\boldsymbol{x} \in \mathbb{R}^n$ be the feature vector of a single order, where each component x_i represents a recorded stage duration. A complete customer journey can be represented as:

$$\boldsymbol{x} = \begin{bmatrix} t_{\text{order}} \\ t_{\text{services}} \\ t_{\text{completion}} \end{bmatrix} \in \mathbb{R}^{3\times 1}. \tag{3.2}$$

For all m recorded orders across stores, the resulting dataset can be described as matrix $\boldsymbol{X} \in \mathbb{R}^{m\times 3}$, with each row corresponding to an observed customer checkout timeline. This formalization enables us to analyze, benchmark, and optimize specific components of the process using digital tools, including business rules and AI-based techniques.

3.1.2 Cutomer's pipeline optimization

The barcode mechanism is developed inside the ERP system by refining the current configuration in the 1C programming language. Let's divide t_{checkout} into subsections methodology [30] and determine the appropriate point for barcode integration:

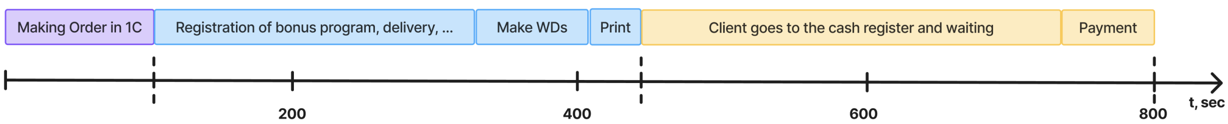


Fig. 3.1. Customer's Checkout Time example

1. **Making Order Time** (t_{order}):

   - Making order in 1C: After consulting the client, the sales consultant types the necessary items (SKUs) through the product range search and enters their number

2. **Service Registration Time** (t_{services}):

   - Registration of the bonus program, delivery and other services: After forming the list of goods, the client is asked about his bonus program, if not, create a loyalty card. Also, client is asked about delivery services (about 50% of orders have delivery). The cost is announced to the client, which he can refuse. The last step in providing services of specific products (is sawing, mixing colors, and so on)

- Make warehouse documents (WD): Customer's order be able to pick up the goods from the warehouse. Due to the registration of the "Expense order" document, a formal operation is performed.
- Printing of documents: The sales consultant prints the customer's order and "Expense order". The client is followed with the documents to the cashier

3. **Completion Time** ($t_{\text{completion}}$):

   - Client goes to the cash register and waiting: The customer goes to wait in line. Depending on the time of day, a customer can wait in line for up to 30 minutes
   - Payment, Receipt printing: The customer writes off his bonuses, takes snacks or drinks (optional actions) and pays for the purchase

Highlighting the subsections related to information technology, it can be noted that t_{services} takes the most time on average, which involves registration services and processing (printing documents).

We can conclude that the most vulnerable point that needs to be optimized is document printing. The cost of printing, the turn-on time, printing the printer (2-3 A4 papers) and signing them takes valuable time. The barcode mechanism allows you to digitalize this step by optimizing t_{services} and $t_{\text{completion}}$.

Based on articles from Chapter 2, we can technically describe our replacing printed documents with a reusable linked barcode card mechanism:

1. The barcode will be pushed by the sales consultant after the Customer's order is placed

2. When the registration of services is completed, the sales consultant gives the customer a barcode card and follows his to the cashier

3. So, we do not hand over printed documents to the client. We can follow the client to the cashier, parallelize process the warehouse documents

4. When it's the customer's turn, the cashier takes a barcode card instead of documents and, based on the ID, ERP program opens the appropriate order

5. After completing the work on the Order, the cashier cleans the link of the barcode card and returns it to the counter of the sales consultant

6. So each sales consultant has about 15 barcode cards in personal stock

Let's display the proposed modification on the Customer's Checkout Time example below, demonstrating the effectiveness of the barcode card implementation:

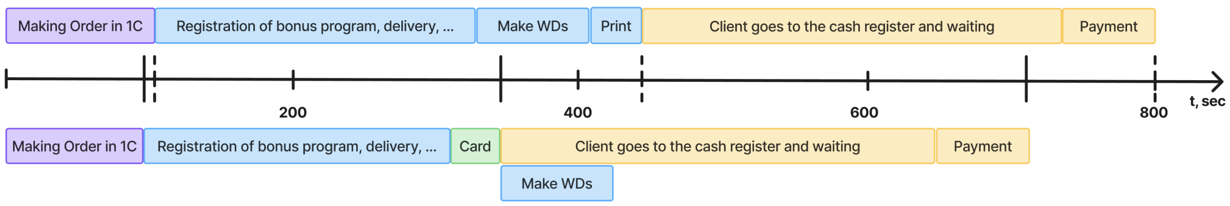


Fig. 3.2. Customer's Checkout Time example before/after optimization

3.1.3 ERP configuration improvements

To implement the barcode mechanism, it is necessary to modify the current ERP program configuration: programs from the client and server sides. This mechanism will be enable seamless linking between a physical barcode card and a digital Customer Order.

The process begins when a recognized barcode (EAN-13 format with a specific mask prefix) is scanned by a client-facing device. The client-side handler checks whether the scanned value corresponds to a registered card. If it does, and the Customer order has already been saved to the database, the barcode is validated and then linked to the order using the **BindCardToOrder procedure**. This server-side routine ensures that each barcode is uniquely assigned, avoiding conflicts such as multiple links between the same card and multiple documents or vice versa.

The system includes logic for displaying the linked barcode on the client form when reopening the order. This functionality is implemented in the **OnOrderFormOpen procedure**, which fetches and displays the associated barcode. The linkage is stored in a dedicated specific register.

For back-side and cashier-side processing, the system provides barcode recognition handlers to automatically retrieve the corresponding Customer's order based on a scanned card. If a barcode card is scanned that does not belong to any linked order, the system informs the operator accordingly. Additionally, when a barcode needs to be unlinked (e.g., after payment or reuse), a separate **HandleBarcodeUnlinking procedure** safely retrieves and displays the current linkage status to assist in card return and reallocation.

Overall, this algorithm is suite form for paperless customer order workflow, accelerating Customer Checkout Time operations. The system enhances user experience while maintaining data integrity and process traceability through tightly controlled business logic on both client and server sides.

3.1.4 Data analysis

In order to evaluate the effect of the introduction of barcode cards, we extract the duration of each main stage in the Customer's Checkout Time based on flagship stores of the company:

Fig. 3.3. Flagship stores of the company

Dataset Description. The data was collected from three stores — *Umasheva*, *Borodinskaya*, and *Tikhaya* — covering the three months before implementation (Q3: July–September) and the three months after (Q4: October–December). The dataset contains 59,998 observations. Each entry includes:

- Order Date, Service Registration Date, Completion Time – timestamps of each process stage
- Manager – the employee responsible for order placement
- Order Amount – value of the transaction
- Customer Type – individual or legal entity
- Has Extra Services – indicator of additional services
- Making Order Time, Service Registration Time, Completion Time – time deltas in minutes for each phase

- Period – the phase Q3 or Q4
- Store – location of service

Data Preprocessing. To ensure data quality, duplicate rows were removed. Outliers were filtered using the Interquartile Range (IQR) method [31]. This technique keeps only the central range of data and removes extreme values that can distort statistical calculations.

The interquartile range is calculated as follows:

$$IQR = Q_3 - Q_1 \tag{3.3}$$

where Q_1 is the lower quartile (25th percentile) and Q_3 is the upper quartile (75th percentile). An observation x is considered an outlier if it satisfies the condition:

$$x < Q_1 - 1.5 \cdot IQR \quad \text{or} \quad x > Q_3 + 1.5 \cdot IQR \tag{3.4}$$

This approach was selected to retain only values within the central distribution range, ensuring that the analysis focuses on representative (high-confident) cases rather than exceptional ones.

After preprocessing, the resulting dataset comprised 59,491 filtered customer orders, denoted as $D_{RQ1} = \{x_1, x_2, \ldots, x_N\}$, where each $N-$observation corresponds to a unique customer path of t_{checkout}.

3.2 RQ2 - Recommendation system integration

3.2.1 Problem Statements

Let's consider the task of recommending additional products in real time in a retail environment. Formally, the goal is to create an AI algorithm capable of offering relevant products to a customer's current shopping cart. The customer's current shopping cart implies the state when the sales consultant entered the list of goods that the client has announced to buy.

The key limitation of the task is the lack of prior user identification during product selection. This is due to the fact that the company's loyalty program is introduced as soon as the seller completes the customer's order. Therefore, there is no data for past periods for a particular client, and the model must operate in cold start mode [25].

Let $B = \{i_1, i_2, \ldots, i_n\}$ denote the set of items of customer's current shopping cart, where each n-item represents a stock keeping unit (SKU) from the company's product catalog. The recommendation system aims to compute a ranked list of additional items $\hat{R}(B) = \{r_1, r_2, \ldots, r_k\}$, where each k-item is selected based on relevance to the current basket B, using global purchase history of other customers as the learning signal.

To train and evaluate the recommendation algorithms, we use a real-world dataset extracted from the company's 1C ERP system, denoted as $G = \{O_1, O_2, \ldots, O_m\}$, where each m-orders represents a single customer order and contains the following structured fields:

- Order — integer order identifier
- $\mathrm{Timestamp}$ — date and time of order placement in the format $\mathrm{dd.mm.yyyy}$

hh:mm:ss

- Store — store location where the order was completed
- Total_amount — total monetary value of the order in rubles
- Items — list of purchased products, each described as:
  - Nomenclature — full hierarchical path of the item in the product catalog
  - Group — higher-level nomenclature group
  - Quantity — number of units purchased
  - Unit_price — price per unit
  - Line_amount — total price for the item (Quantity × Unit_price)

Dataset includes 665,733 customer orders from 2023 to 2025 from Vladivostok stores and 12,881 unique products. All product names and store identifiers have been anonymized according to with internal privacy policies. The data is randomly split into training and testing sets using an 80/20 ratio. Orders containing more than one SKU in the test set are used to evaluate the recommendation quality.

After preprocessing, the dataset containing $492,999$ customer orders and $12,881$ unique products. All product names and store identifiers have been anonymized according to internal privacy policies. The data is randomly split into training and testing sets using an 80/20 ratio. Orders containing more than one SKU in the test set are used to evaluate the recommendation quality.

To formalize evaluation, for each test order O_i, the set of items is split into two equal halves:

$$O_i = B_i \cup T_i, \quad B_i \cap T_i = \emptyset,$$

where B_i is the observed basket and T_i is the set of held-out items used to validate recommendations. The recommender outputs a top-k list $\hat{R}(B_i)$, and the predicted relevance is assessed using ranking-based metrics [32] such as Recall@20 and nDCG@20. These metrics are computed as follows:

- **Recall@k**:
$$\mathrm{Recall} \bullet \mathrm{k} = \frac{\mid \hat{R}(B_i) \cap T_i \mid}{\mid T_i \mid}, \tag{3.5}$$
  which measures the fraction of relevant items retrieved within the top-k predictions

- **nDCG@k** (Normalized Discounted Cumulative Gain):
$$\mathrm{nDCG} \bullet \mathrm{k} = \frac{1}{\mathrm{IDCG} \bullet \mathrm{k}} \sum_{j=1}^{k} \frac{\nVdash[r_j \in T_i]}{\log_2(j+1)}, \tag{3.6}$$
  where $\nVdash[r_j \in T_i]$ is an indicator function equal to 1 if the predicted item r_j is relevant, and $\mathrm{IDCG} \bullet \mathrm{k}$ is the ideal DCG, computed over the perfect ranking of items in T_i

The recommendation system is trained to maximize these metrics across all test orders, thereby providing personalized suggestions to assist sellers in real-time. The ultimate goal is to increase the average number of SKUs per customer order while maintaining relevance and accuracy, despite the lack of individual user history at prediction time.

Applied model evaluation. The experimental period for evaluating the effectiveness of the recommendation system within the framework of the thesis will be 3 working days. The sales consultant will be asked to recommend the products of the proposed models instead of their own. The average values of the number of SKUs based on the available dataset will be compared with the values during the experimental period.

3.2.2 Data Preparation

Data Cleaning. Initial steps included removing duplicate orders, entries associated with warehouse-only transactions, and orders generated using non-standard documentation workflows that are not representative of retail store behavior. Outliers were further filtered using the Interquartile Range (IQR) method [31], which is defined as:

$$\mathrm{IQR} = Q_3 - Q_1, \tag{3.7}$$

where Q_1 and Q_3 denote the first and third quartiles of a given metric, respectively. Any order duration z falling outside the interval $[Q_1 - 1.5 \cdot \mathrm{IQR}, Q_3 + 1.5 \cdot \mathrm{IQR}]$ was considered an outlier and removed. Additionally, 5% of extreme minimum and maximum values were excluded to reduce noise and improve generalization.

After cleaning, the resulting dataset comprised $492{,}999$ filtered customer orders, denoted as $D_{RQ2} = \{O_1, O_2, \ldots, O_N\}$, where each N-order corresponds to a unique client transaction using for recommendation models.

User-Item Matrix Construction. In the context of our recommendation problem, each customer order O_i is treated as an individual user, since user identity is not available prior to checkout. Let U be the set of orders (users) and I be the set of unique items. A sparse matrix $\boldsymbol{X} \in \mathbb{R}^{|U|\times|I|}$ was constructed in two versions:

- **Binary matrix**: $\boldsymbol{X}_{ui} = 1$ if item i was included in order u, and 0 otherwise
- **Quantity matrix**: $\boldsymbol{X}_{ui} = q$ if item i appeared in order u with quantity q

These matrices were used as input for collaborative filtering models.

Item Embedding Extraction. Recent advances [33] in natural language processing (NLP) suggest that semantic representation of item names can improve similarity-based recommendation models. To exploit this, we embedded all unique product names using two transformer-based models:

- $\mathrm{paraphrase\text{-}MiniLM\text{-}L6\text{-}v2}$ [34], a compact English-language model optimized for semantic similarity
- $\mathrm{sbert\_large\_nlu\_ru}$ [35], a high-capacity Russian SBERT variant suitable for domain-specific and Cyrillic text embeddings

Each item $i \in I$ was encoded as a dense vector $\boldsymbol{e}_i \in \mathbb{R}^{384}$, followed by L2-normalization:

$$\boldsymbol{e}_i \leftarrow \frac{\boldsymbol{e}_i}{\|\boldsymbol{e}_i\|_2}, \tag{3.8}$$

ensuring that all embeddings reside on the unit hypersphere, thereby enabling cosine similarity computation.

Cosine similarity is used as a key metric for ranking candidate items based on their proximity to the basket's average embedding vector. The similarity between two items i and j, represented by their normalized embeddings $\boldsymbol{e}_i$ and $\boldsymbol{e}_j$, is calculated as:

$$\text{sim}(i,j) = \frac{\boldsymbol{e}_i \cdot \boldsymbol{e}_j}{\mid \boldsymbol{e}_i \mid \cdot \mid \boldsymbol{e}_j \mid}, \tag{3.9}$$

where $\mid \boldsymbol{e}_i \mid$ is the L2 norm (Euclidean length) of vector $\boldsymbol{e}_i$, computed as:

$$\mid \boldsymbol{e}_i \mid = \sqrt{e_{i,1}^2 + e_{i,2}^2 + \cdots + e_{i,d}^2}, \tag{3.10}$$

where d is the dimension of the embedding. All embeddings are L2-normalized in advance, making the cosine similarity equivalent to a simple dot product. This allows effective ranking of recommendations based on semantic proximity.

Item Embedding will be used both separately and in the analysis presented below. Separate use implies that the recommendation function will rank the received products embedded from the second half of the basket. The recommendation will be arranged in descending order of similarity to the average vector of the first half of the basket.

3.2.3 Learning Representations from Purchase Behavior

To uncover latent semantic structure in product purchase patterns, we applied a five-step representation learning strategy. High-dimensional sparse co-purchase data was reduced via matrix factorization, followed by nonlinear manifold projection. Then, product embeddings were clustered using density-based algorithms to support content-aware recommendation logic.

Step 1: Semantic Labeling of Clusters. Each product was annotated using expert-driven hierarchy parsing rules to infer its *power group* — a category-level descriptor extracted from its nomenclature path. This allowed post-clustering inspection of dominant product groups inside each cluster.

Step 2: Dimensionality Reduction via SVD and UMAP. Given the binary user-item interaction matrix $\boldsymbol{X} \in \mathbb{R}^{|U| \times |I|}$, we applied Truncated Singular Value Decomposition (SVD) to extract latent item representations [36]:

$$\boldsymbol{X}^\top \approx \boldsymbol{U}\Sigma\boldsymbol{V}^\top, \tag{3.11}$$

where:

- $\boldsymbol{U} \in \mathbb{R}^{|I| \times k}$ — matrix of item embeddings
- $\Sigma \in \mathbb{R}^{k \times k}$ — diagonal matrix of singular values
- $k = 100$ — dimensionality of the reduced space

To make the embedding space suitable for visual analysis and clustering, we employed Uniform Manifold Approximation and Projection (UMAP) [37], a nonlinear dimensionality reduction technique that preserves local and global structure. UMAP optimizes a cross-entropy loss between high-dimensional and low-dimensional neighbor graphs:

$$\mathcal{L}_{\text{UMAP}} = \sum_{(i,j)} \left[p_{ij} \log \frac{p_{ij}}{q_{ij}} + (1 - p_{ij}) \log \frac{1 - p_{ij}}{1 - q_{ij}} \right], \tag{3.12}$$

where:

- p_{ij} — probability of edge (i, j) in high-dimensional space
- q_{ij} — corresponding probability in the 2D embedding space

The resulting 2D coordinates $z_i = \mathrm{UMAP}(\boldsymbol{e}_i) \in \mathbb{R}^2$ for each item i were saved for visualization and clustering:

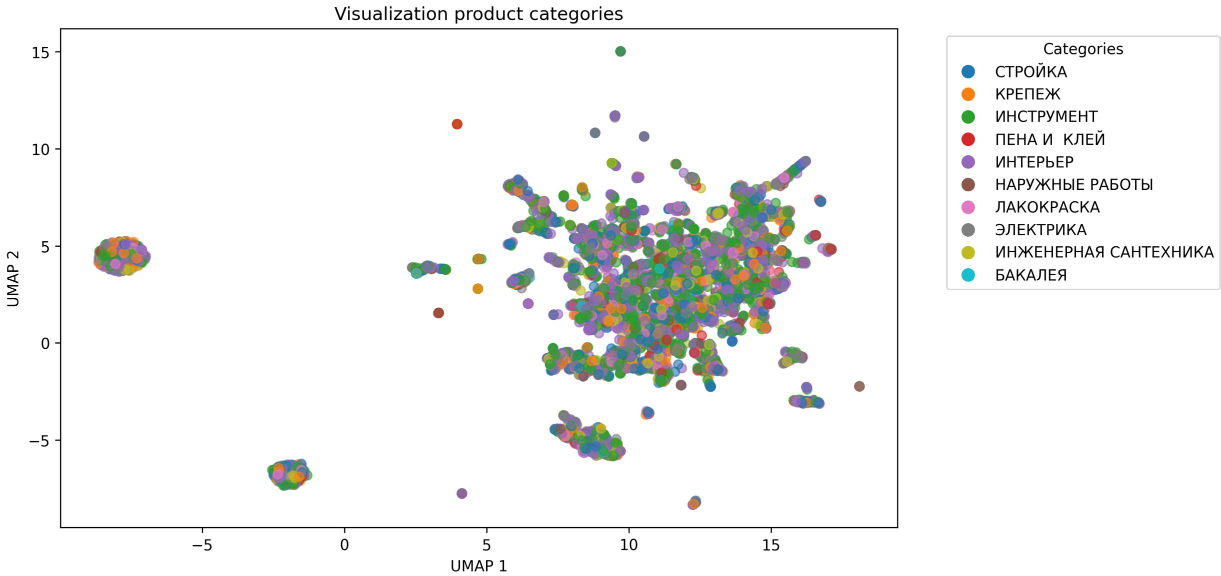


Fig. 3.4. UMAP projection of item embeddings colored by powered product categories.

Step 3: Density-Based Clustering. DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [38] defines clusters as regions of high point density. After reducing item embeddings into a two-dimensional space using UMAP, clustering is performed by examining the local neighborhood of each point a_i. A point is assigned to a cluster if the number of neighboring points within a radius ϵ exceeds a predefined threshold t (minimum samples):

$$\mathrm{cluster}(a_i) = \begin{cases} c, & \text{if } \mid \mathcal{N}_\epsilon(a_i) \mid \geq t, \\ -1, & \text{(classified as noise otherwise).} \end{cases} \tag{3.13}$$

Here, $\mathcal{N}_\epsilon(a_i)$ denotes the set of neighbors of a_i within the radius ϵ, and c is the cluster label. The parameter t (commonly referred to as "minPts") determines the minimum number of points required to form a dense region. If this condition is not met, the point is considered noise.

Step 4: Density Surface Analysis. To better understand the distribution of product embeddings in the reduced space, we performed density analysis using Kernel Density Estimation (KDE).

KDE [39] is a non-parametric way to estimate the probability density function of a random variable. In this context, it helps identify regions with high concentrations of products, which may reflect frequent co-purchase behavior.

Mathematically, the KDE for a two-dimensional space is defined as:

$$\hat{f}(x,y) = \frac{1}{nh^2} \sum_{i=1}^{n} K\left(\frac{x-x_i}{h}, \frac{y-y_i}{h}\right), \tag{3.14}$$

where $\hat{f}(x,y)$ is the estimated density at point (x,y), n is the number of observations, h is the bandwidth (smoothing parameter), and $K(\cdot)$ is the kernel function, typically a Gaussian.

This visualization technique allows us to highlight density peaks, which often correspond to tightly-knit clusters of related products. These peaks were further analyzed to understand product grouping and labeling for business insights. The discovered clusters were labeled and saved as item metadata for further use in hybrid re-ranking models and interpretability analysis.

Step 5: Hierarchical Density-Based Clustering. HDBSCAN (Hierarchical DBSCAN) [40] generalizes DBSCAN by eliminating the need to set a fixed density threshold ϵ. Instead, it builds a hierarchy of clusters based on mutual reachability distances and selects the most stable clusters. A cluster is considered *stable* if it persists over a wide range of density levels.

Each cluster C_j is evaluated for its stability score S_j, defined as:

$$S_j = \sum_{a_i \in C_j} \left(\lambda_{\text{max}}(a_i) - \lambda_{\text{birth}}(a_i) \right), \tag{3.15}$$

where $\lambda_{\text{birth}}(a_i)$ is the inverse density (i.e., reachability) at which point a_i first enters the cluster, and $\lambda_{\text{max}}(a_i)$ is the density level at which it leaves or the cluster splits. The clusters with higher S_j values are considered more robust and meaningful.

The resulting HDBSCAN clusters are integrated into the final ranking logic of the recommendation function. For each input basket, the top-3 most common clusters are computed from the encoded products. If a recommended item belongs to one of these top clusters, its similarity score is boosted as follows:

- Cluster rank 1: $\times 1.2$
- Cluster rank 2: $\times 1.1$
- Cluster rank 3: $\times 1.05$

This adjustment is designed to prioritize items aligned with typical construction purchase patterns, and empirical evaluation showed consistent improvements in nDCG@20 when cluster-aware ranking was applied.

3.2.4 Model Architectures

To identify the most effective recommendation strategy for industrial retail, we compare multiple algorithmic families, each representing a distinct approach to the problem of item suggestion in real-time environments.

Baseline: Global Top-N Recommendation A simple baseline model serves as the starting point of our evaluation. This model recommends the top 30 most frequently purchased products across the entire dataset. It is useful as a benchmark due to its interpretability and deterministic nature. However, it lacks personalization and typically recommends high-frequency auxiliary items such as packaging or consumables, which are not personalization way.

Collaborative Filtering Approaches: ALS and Item2Item. Collaborative filtering techniques are among the most widely used in recommender systems. We evaluate two prominent variants [21]:

- **Alternating Least Squares (ALS)** decomposes the user-item matrix $\boldsymbol{R}$ into two low-rank matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ by minimizing the following objective:

$$\min_{\boldsymbol{U},\boldsymbol{V}} \sum_{(i,j)\in\mathcal{K}} (R_{ij} - \boldsymbol{u}_i^T \boldsymbol{v}_j)^2 + \mu(\|\boldsymbol{u}_i\|^2 + \|\boldsymbol{v}_j\|^2) \tag{3.16}$$

  where $\mathcal{K}$ denotes the set of observed user-item interactions, and μ is a regularization parameter.

- **Item-to-Item (Item2Item)** similarity computes cosine distances between item vectors derived from user interactions (based on the cosine similarity formula from subsection 3.2.2). This method benefits from simplicity and efficiency, especially with sparse matrices.

Both models are tested with two types (binary and quantity) of user-item matrices $\boldsymbol{X}_{ui}$. Further, we enrich item representations using MiniLM and SBERT embeddings, and re-rank the recommendations according to clustering information obtained from DBSCAN and HDBSCAN. This hybridization improves relevance by incorporating semantic and categorical proximity.

Transformer-Based Model: BERT4Rec. **BERT4Rec** [27] is a sequential recommendation model that adapts the bidirectional encoder structure of BERT to predict masked items in user interaction sequences. Let a sequence of previously interacted items be represented as $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_M$, where each $\boldsymbol{v}_t \in \mathbb{R}^d$ is an embedded vector of items in a d-dimensional vestor space, and M is the maximum sequence length. During training, one or more items are randomly replaced with a special token [MASK], and the model is trained to reconstruct the original item:

$$\hat{\boldsymbol{v}}_t = \text{Transformer}(\boldsymbol{v}_1 + \boldsymbol{p}_1, \ldots, \boldsymbol{v}_{t-1} + \boldsymbol{p}_{t-1}, [\text{MASK}] + \boldsymbol{p}_t, \ldots, \boldsymbol{v}_M + \boldsymbol{p}_M) \quad (3.17)$$

Here, $\boldsymbol{p}_t$ denotes the positional embedding for index t, and $\hat{\boldsymbol{v}}_t$ is the predicted embedding of the masked item. The model is trained using a masked language modeling objective optimized for item recommendation.

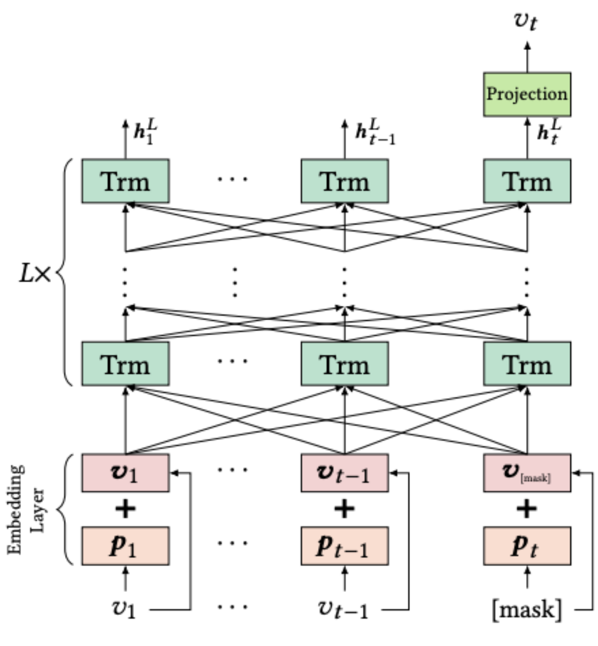


Fig. 3.5. BERT4Rec model architecture [27]

The Transformer encoder learns contextualized representations $\boldsymbol{h}_t^L$ for each position, which are then passed through a linear projection layer to yield final predictions.

To further enhance performance, we perform re-ranking using semantic similarity between the predicted $\hat{\boldsymbol{v}}_t$ and the SBERT- and MiniLM-based embeddings of all candidate items. Additionally, a cluster-based boosting strategy using DBSCAN and HDBSCAN cluster labels is applied to prioritize items aligned with the user's historical preferences.

3.3 Selected tools and libraries

The Python programming language was used throughout the study for data preprocessing, analysis, modeling, and visualization. All experiments were performed in the Google Colab environment, which provides an integrated ecosystem for data manipulation, reproducible experiments, and result visualization in a single platform.

System operations such as connection to the GitHub repository and unzipping archives were handled via the os and zipfile libraries. Data loading, transformation, and cleansing — including duplicate removal, document type filtering, and outlier detection using the IQR method — were conducted using pandas and numpy. The resulting user-item interaction matrices were built in both binary and quantity form using scipy.sparse.

For learning representations of purchase behavior, matrix factorization was performed using TruncatedSVD, followed by projection into two-dimensional space with UMAP. Clustering algorithms DBSCAN and HDBSCAN were applied to identify groups of semantically or behaviorally similar items. Visualization of categories, clusters, and density distributions was carried out with matplotlib and seaborn. Semantic and lexical structure of product names was analyzed using CountVectorizer from scikit-learn.

Model architectures were implemented and evaluated using a combination of classical and neural techniques. The implicit library was used for collaborative filtering via Alternating Least Squares (ALS) and Item2Item models. Semantic product representations were integrated using pretrained transformers from the sentence-transformers library (MiniLM and SBERT). For training a session-based neural recommender, the BERT4Rec model was implemented with PyTorch and transformers. Final evaluations were performed using top-k metrics such as nDCG@20 and Recall@20, implemented with numpy.

Chapter 4

Results

The chapter describes the results of the implementation of IT solutions proposed in the framework of research issues. Section 4.1 presents the barcode mechanism in three flagship stores and data analytics for three months before/after implementation. Section 4.2 demonstrates what the study of data analysis and clustering has led to, the results of the architecture models discussed in Chapter 3, and the impact after implementing the recommendation system in stores.

4.1 RQ1 - Barcode mechanism results

In order to optimize the customer's order processing time, the ERP program configuration components and the production of barcode cards were developed. The adaptation of the new business process of the customer's path was carried out. The evaluation of the result was carried out in three flagship stores of the company, which have different building characteristics and limitations. For the analysis, data was taken for three months before the implementation and three months after.

Data analysis of the $59{,}491$ observations showed the following changes for the three months before implementation (Q3: July–September) consisting of $24{,}841$ observations and the three months after (Q4: October–December), consisting of $29,650$ observations, that make up Customer checkout time:

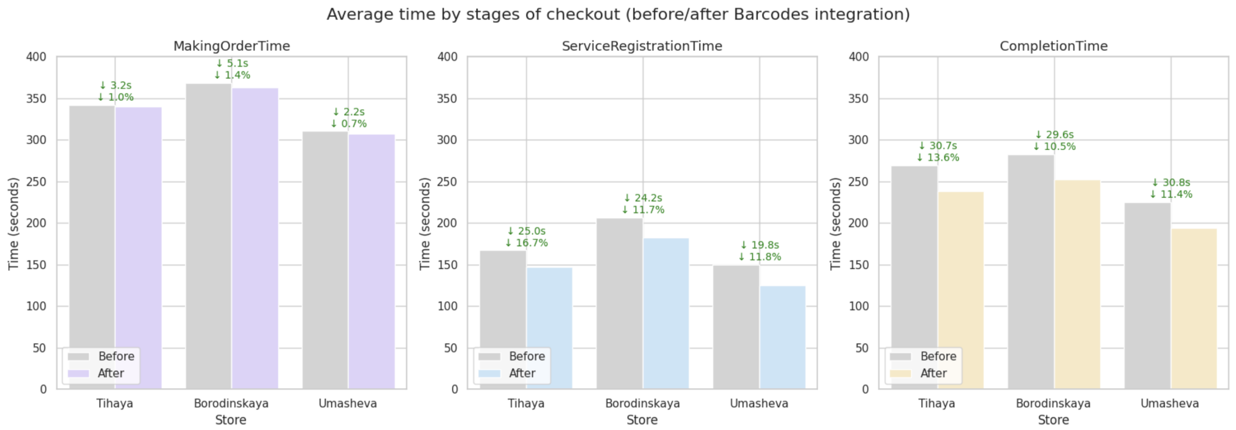


Fig. 4.1. Time periods before/after barcode cards integration

As the results of the implementation demonstrate, the barcode mechanism has significantly reduced Service Registration Time from 11.7% to 16.7% (from 19.8 s to 25.0 s) and Completion Time from 10.5% to 13.6% (from 29.6 s to 30.8 s). The Making Order Time reflects the time of the client's first contact, which demonstrates minor changes.

The overall decrease in **Customer Checkout Time was from 6.8% to 8.6%**, which means that each customer spends an average of 52.8 seconds to 59 seconds less, depending on the store:

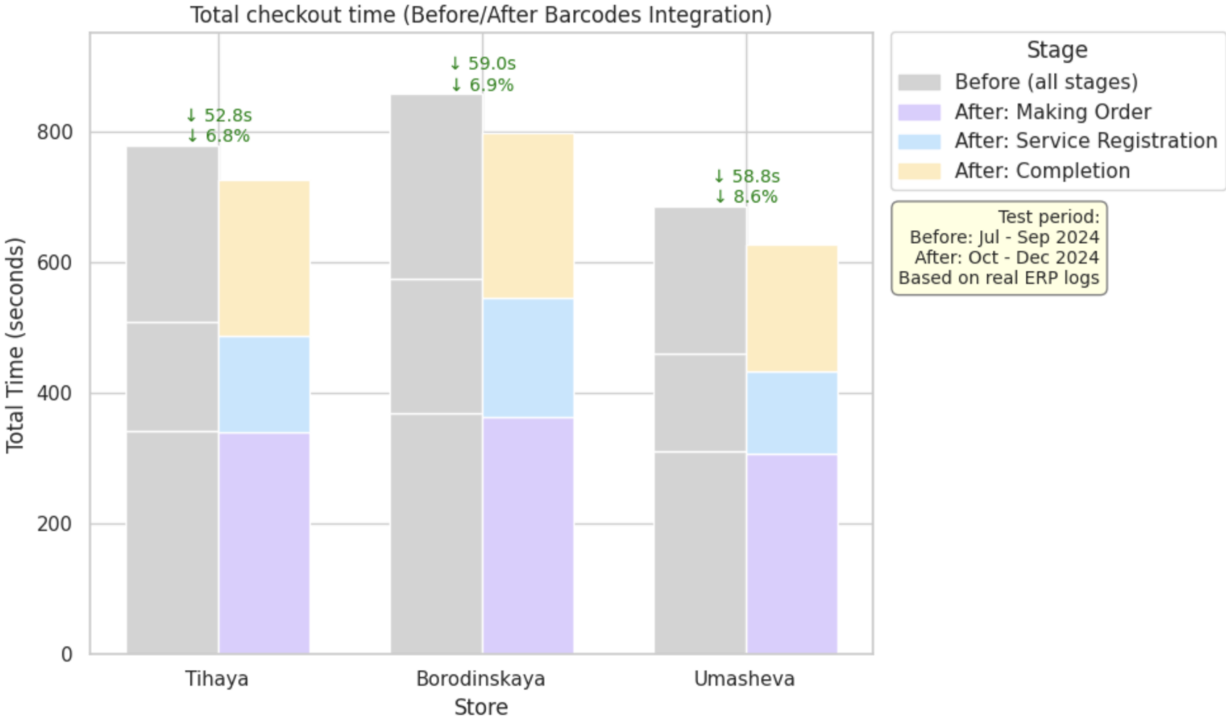


Fig. 4.2. Customer's Checkout Time Results

Also, the introduction of barcode cards made it possible to remove printers from the workplaces of sales consultants, which reduced the cost of printing, buying cartridges and paper.

Expert assessment. After the introduction and analytical evaluation of the barcode cards, the company's Executive Director conducted an audit of the integration, which resulted in an **overall operating efficiency of 78,050,420 rubles per year** across the entire construction retail network. Based on the time elapsed after integration, the efficiency plan is being implemented.

4.2 RQ2 - Recommendation system results

4.2.1 Learning Representations

Before directly training the model, UMAP-based data analysis was able to identify different customer patterns. DBSCAN clustering at $\epsilon = 0.87$ divided the products into 10 classes (with a minimum number of points in a class of 60). At the same time, 243 points were attributed to noise (approximately 1.9% of the total). The visualization of the classes is presented below:

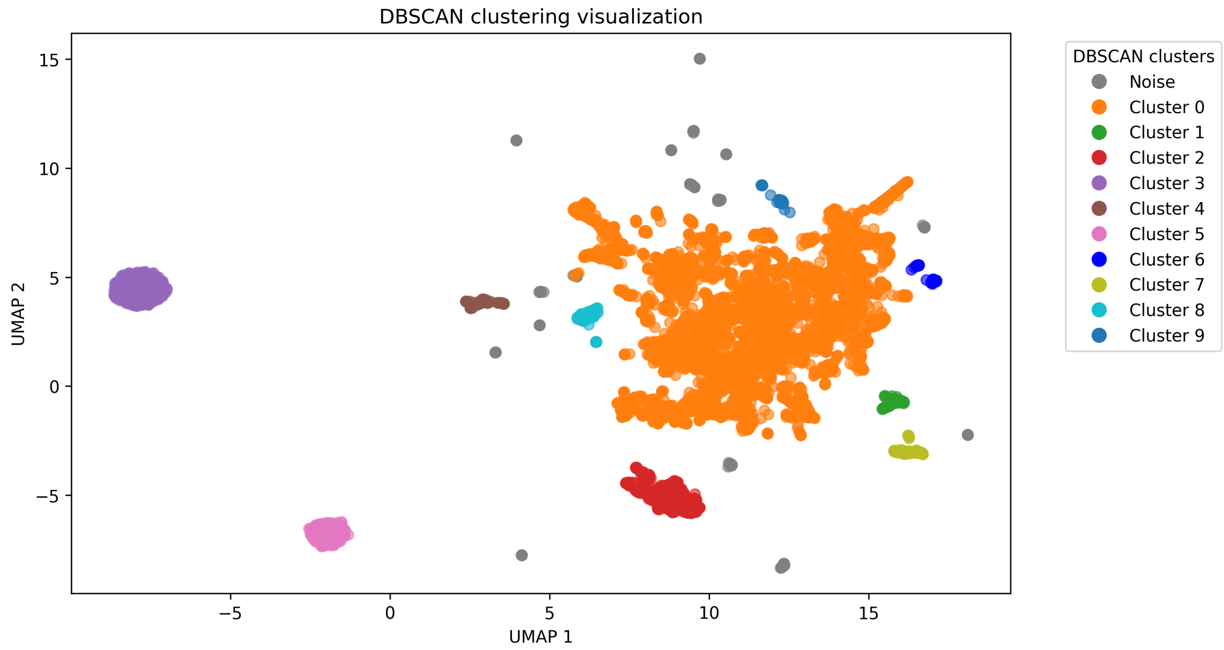


Fig. 4.3. Clustering result using DBSCAN on UMAP-reduced product embeddings.

Density Surface Analysis showed other differences, highlighting 4 peaks, one of which is many times higher than the others. The result was illustrated using contour maps and 3D surfaces, which revealed product hotspots and provided input for the HDBSCAN clustering:

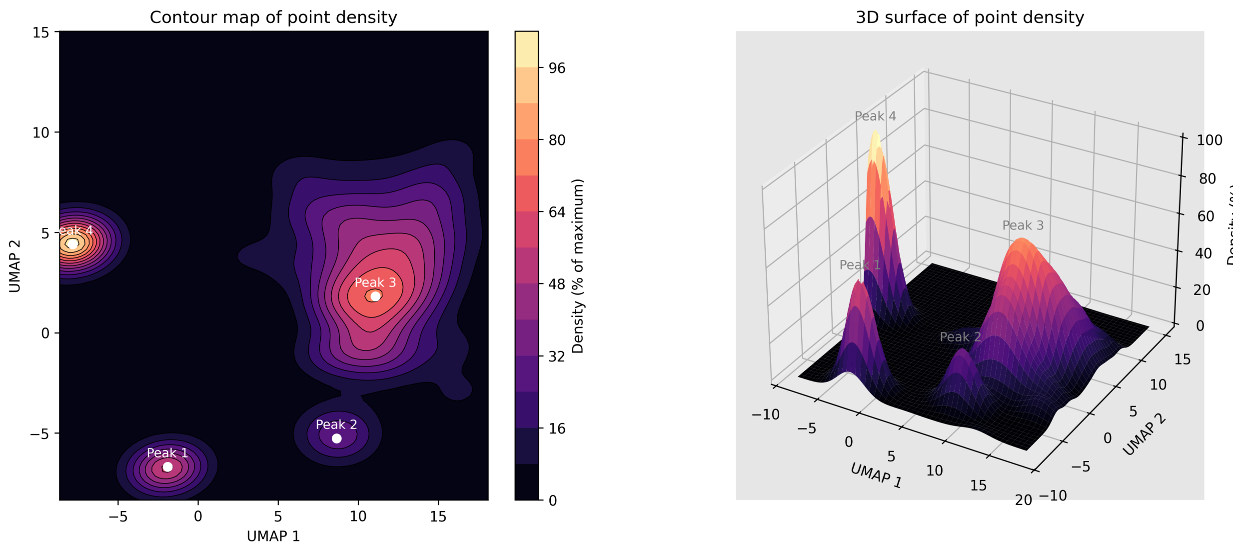


Fig. 4.4. Left: 2D density contour plot. Right: 3D density surface showing concentration of product types.

As needed to be seen, HDBSCAN identified 4 classes with 266 noise points (approximately 2% of the total). This separation was achieved empirically with the minimum number of points in each cluster being 250. In this case, the cluster density was determined by the min sample parameter equal to 30. The HDBSCAN clustering result were visualized as follows:

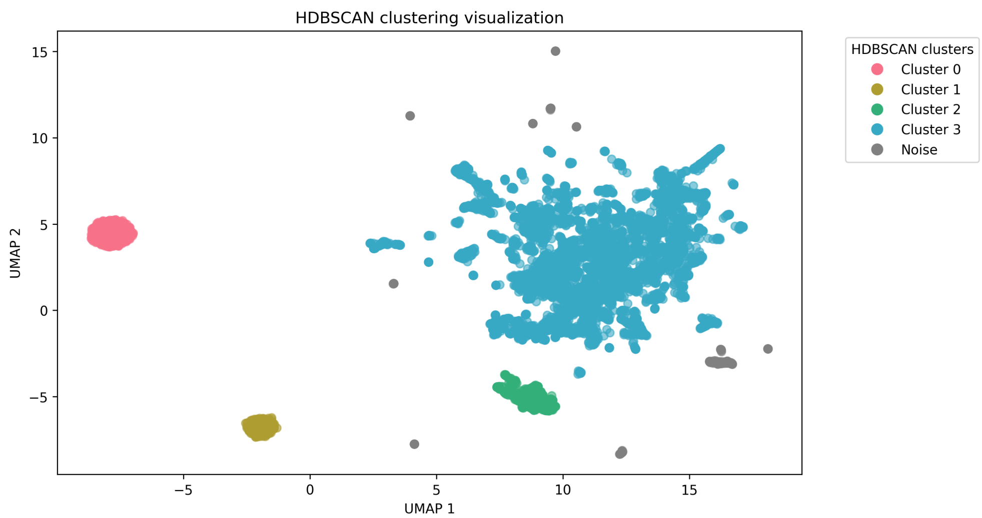


Fig. 4.5. Clustering result using HDBSCAN, which dynamically adjusts density thresholds.

4.2.2 Models evaluation

The general model training pipeline consists of downloading all the necessary datasets consisting $492,999$ customer orders, $12,881$ unique products and metadata for conducting the current experiment, selecting the model and configuring the parameters of the recommendation function, training and evaluation by metrics.

Baseline: Global Top-N Recommendation. Training the baseline model (output of the top 30 most frequently purchased products) on all train data resulted in $\mathrm{nDCG@20} = 0.1146$ and $\mathrm{Recall} = 0.1908$. The model does not imply adding or modifying data, as it is the initial values from which we will proceed to evaluate the results.

Collaborative Filtering Approaches: ALS and Item2Item. Models were initially tested on Binary and Quantity matrices. When configuring ALS, the number of latent factors for lower dimensions latent space projected from user-item interaction matrix was empirically $16 \text{ over } 8 \text{ iterations}$ of the model. The item2item setting implies the number of elements most similar in cosine distance: $k = 10$ (also selected empirically based on the conclusions of the model).

The results showed that both models work better on Binary user-item matrices instead of Quantity. It was this combination that was able to obtain indicators superior to the baseline model. In addition, the use of individual item embeddings and the use of clustering for ranking did not improve.

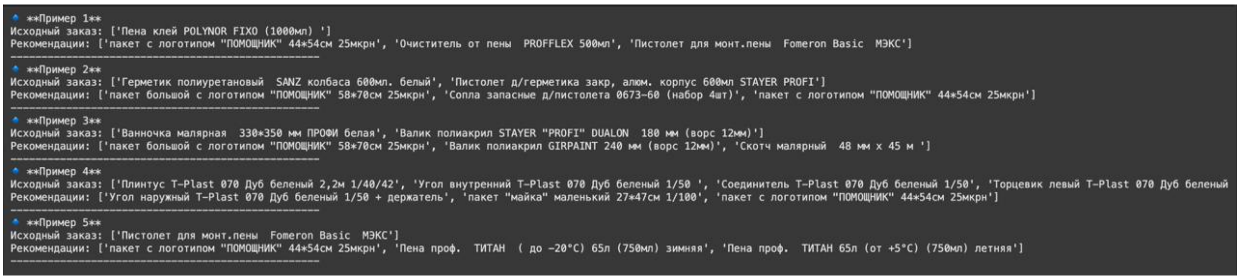
Transformer-Based Model: BERT4Rec. In our implementation, BERT4Rec uses a Transformer encoder with $L = 2$ layers and 4 self-attention heads per layer. Each item is embedded into a 128-dimensional vector space ($d = 128$). The input sequence is padded or truncated to a fixed length of $M = 10$. After 15 epochs of training, the best metric attempt, taking into account the use of metadata, was able to produce: $\mathrm{nDCG@20} = 0.2473$ and $\mathrm{Recall@20} = 0.3790$. The most effective method was the use of BERT4Rec with product ranking using HDBSCAN clusters.

The methods were combined by choosing the best option for a particular modification to test the next change in the model training plan. The results of all experiments are shown below:

TABLE 4.1
Performance comparison of recommendation models (nDCG@20 and Recall@20 metrics)

Model	**nDCG@20**	**Recall@20**
BERT4Rec + HDBSCAN [27]	**0.2473**	**0.3790**
BERT4Rec + DBSCAN [27]	0.2432	0.3795
Item2Item + Binary Matrix [21]	0.1754	0.2481
ALS + Binary Matrix [21]	0.1225	0.2021
Global Top (Baseline)	0.1146	0.1908
Item2Item + Binary Matrix + MiniLM [21]	0.1086	0.2410
Item2Item + Binary Matrix + SBERT [21]	0.1081	0.2378
Item2Item + Binary Matrix + MiniLM + HDBSCAN [21]	0.1049	0.2307
Item2Item + Binary Matrix + MiniLM + DBSCAN [21]	0.0936	0.1991
ALS + Quantity Matrix [21]	0.0877	0.1442
ALS + Binary Matrix + MiniLM [21]	0.0852	0.1831

The recommendation system was deployed for sales consultants as a tool suggesting products during the order process. Below is an example of a visual image seen by a sales consultant during work:



```
**Пример 1**
Исходный заказ: ['Пена клей POLYNOR FIXO (1000мл) ']
Рекомендации: ['пакет с логотипом "ПОМОЩНИК" 44*54см 25мкрн', 'Очиститель от пены  PROFFLEX 500мл', 'Пистолет для монт.пены  Fomeron Basic  МЭКС']
--------------------------------------------------
**Пример 2**
Исходный заказ: ['Герметик полиуретановый  SANZ колбаса 600мл. белый', 'Пистолет д/герметика закр, алюм. корпус 600мл STAYER PROFI']
Рекомендации: ['пакет большой с логотипом "ПОМОЩНИК" 58*70см 25мкрн', 'Сопла запасные д/пистолета 0673-60 (набор 4шт)', 'пакет с логотипом "ПОМОЩНИК" 44*54см 25мкрн']
--------------------------------------------------
**Пример 3**
Исходный заказ: ['Ванночка малярная  330*350 мм ПРОФИ белая', 'Валик полиакрил STAYER "PROFI" DUALON  180 мм (ворс 12мм)']
Рекомендации: ['пакет большой с логотипом "ПОМОЩНИК" 58*70см 25мкрн', 'Валик полиакрил GIRPAINT 240 мм (ворс 12мм)', 'Скотч малярный  48 мм х 45 м ']
--------------------------------------------------
**Пример 4**
Исходный заказ: ['Плинтус T-Plast 070 Дуб беленый 2,2м 1/40/42', 'Угол внутренний T-Plast 070 Дуб беленый 1/50 ', 'Соединитель T-Plast 070 Дуб беленый 1/50', 'Торцевик левый T-Plast 070 Дуб беленый
Рекомендации: ['Угол наружный T-Plast 070 Дуб беленый 1/50 + держатель', 'пакет "майка" маленький 27*47см 1/100', 'пакет с логотипом "ПОМОЩНИК" 44*54см 25мкрн']
--------------------------------------------------
**Пример 5**
Исходный заказ: ['Пистолет для монт.пены  Fomeron Basic  МЭКС']
Рекомендации: ['пакет с логотипом "ПОМОЩНИК" 44*54см 25мкрн', 'Пена проф.  ТИТАН  ( до -20°С) 65л (750мл) зимняя', 'Пена проф.  ТИТАН 65л (от +5°С) (750мл) летняя']
--------------------------------------------------
```

Fig. 4.6. Recommendations that the sales consultant saw in the next window

Comparative Evaluation of differ Models. Let's compare the different results of the models of the most outstanding models. Let's take models that differ architecturally and look at examples of recommendations they received for the same orders:

TABLE 4.2
Model comparison based on sample customer baskets

Customer Basket (Input)	**Model**	**Recommended Items**
White Interior Paint 10L + Roller for Painting	Global Top (Baseline)	Trash Bags, Plastic Bags, Adhesive Tape
	Item2Item + Binary Matrix	Masking Tape, Stirring Stick, Brush
	BERT4Rec + HDBSCAN	**Masking Tape**, **Stirring Stick**, **Floor Protection Film**
Metal Door Frame + Fastening Screws	Global Top (Baseline)	Trash Bags, Plastic Bags, Adhesive Tape
	Item2Item + Binary Matrix	Door Hinges, Fastening Bracket, Insulation Tape
	BERT4Rec + HDBSCAN	**Door Hinges**, **Mounting Foam**, **Level**
PPR Pipes + Connectors	Global Top (Baseline)	Trash Bags, Plastic Bags, Adhesive Tape
	Item2Item + Binary Matrix	Valve Kit, Clamps, Sealing Tape
	BERT4Rec + HDBSCAN	**Insulation Sleeves**, **Valve Kit**, **Clamps**

For the basket **White Interior Paint 10L + Roller for Painting**, the *Global Top* model suggests generic packaging-related products, demonstrating no contextual awareness. The *Item2Item + Binary* model starts to capture utility items such as a masking tape and brush, but lacks structure. In contrast, **BERT4Rec + HDBSCAN** completes the painting scenario by suggesting a floor protection film, showing understanding of user intention and sequence.

In the case of **Metal Door Frame + Fastening Screws**, the baseline again fails to connect with the task at hand. *Item2Item* provides more relevant fittings but lacks semantic structure. **BERT4Rec + HDBSCAN** provides installation-aware suggestions like mounting foam and a level tool, making it the most semantically coherent choice.

For **PPR Pipes + Connectors**, only the transformer-based model complements the basket with insulation sleeves, a valve kit, and clamps — covering both thermal and mechanical needs. *Item2Item* proposes compatible components but not in any prioritized order. The baseline again offers irrelevant mass-market goods.

These examples demonstrate that the model learned contextual associations across frequently bought-together items, increasing the quality of recommendations and customer satisfaction.

Applied model evaluation. For applied evaluation, we integrate the models with the best metric values into the flagship store "Umasheva" and compare them with the changes, taking 3 days (14-16 Feb 2025) for testing before implementing the recommendation system and 3 days (17-19 Feb 2025) after. In the period from February 14 to February 16, 306 orders were collected. During February 17-19, 296 orders were collected.

An unfinished order was submitted to the referral system after the sales consultant had finished typing the goods that the client had said or selected by the sales consultant. So, the recommendation system displayed the result on a laptop nearby, and the sales consultant selected the goods for the client.

Analyzing data from the internal 1C:ERP program, the average number of products per order before deploying the recommendation system was 9.784 SKU items. During the experimental phase of the **BERT4Rec + HDBSCAN** model, the average number of items per order increased to 10.534 SKU. This represents a relative **increase of approximately +0.75 SKU positions** (or **+7.660%** growth in the number of goods per order).

The distribution of SKU quantities across customer orders before and after the deployment is presented below in the form of boxplot and histogram for comparative analysis:

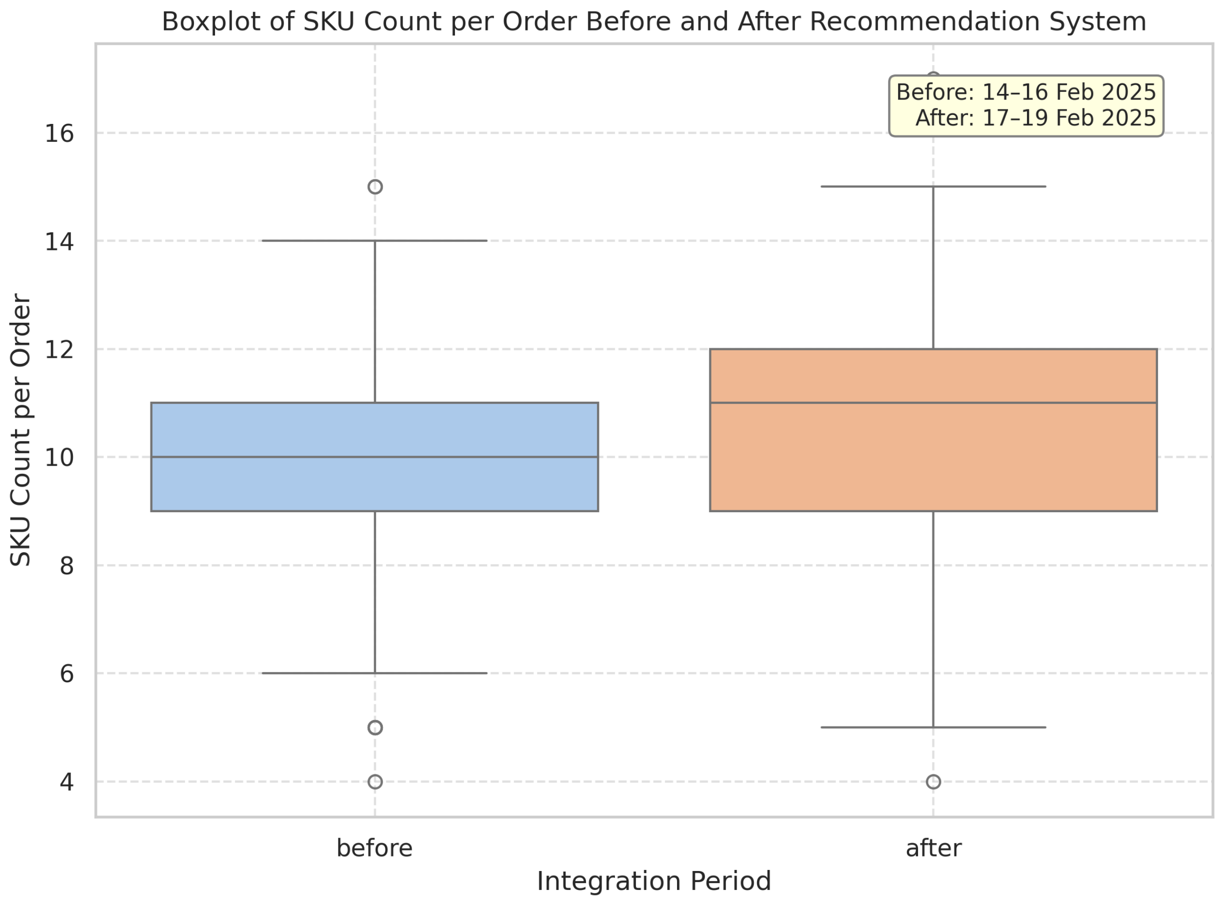


Fig. 4.7. Comparison of SKU item counts per order before and after model deployment

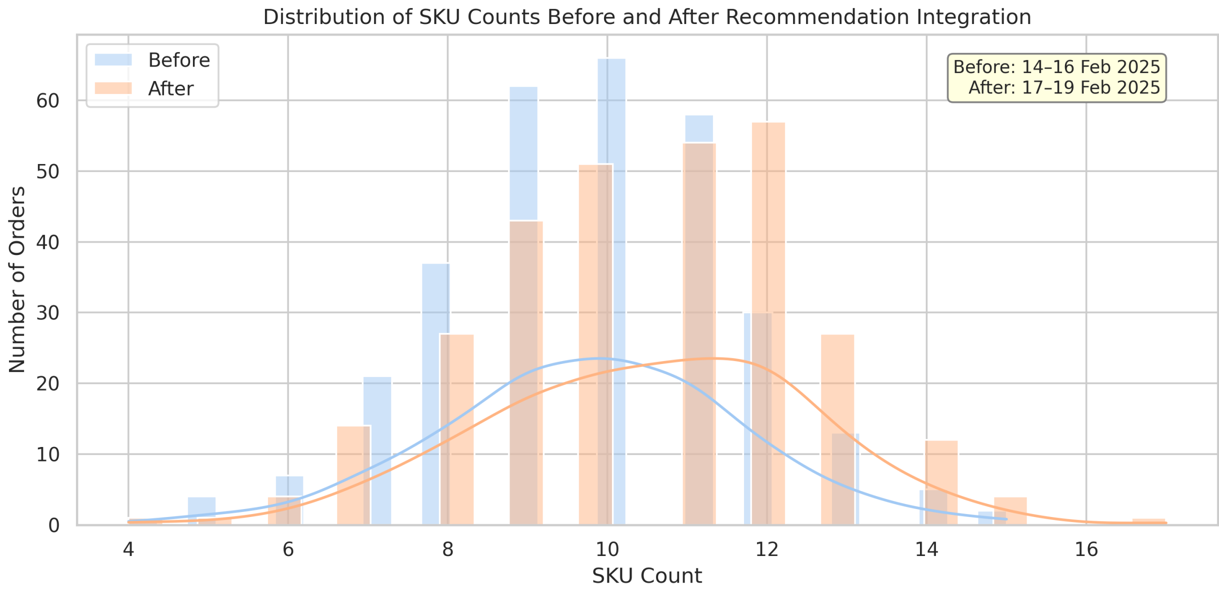


Fig. 4.8. Distribution of SKU item counts

Expert assessment. After the introduction and analytical evaluation of the Recommender system results the company's Executive Director conducted an audit of the integration, which resulted in an **overall operating efficiency of 42,673,540 rubles per year** across the entire construction retail network. Based on the results of a three-day experiment and profit reports for a store employee's workplace.

Chapter 5

Discussion

The chapter reviews the results obtained on RQ1 and RQ2, set out in Chapter 1, and further work prospects. Section 5.1 explains the optimizations of the client's path for the RQ1. Section 5.2 provides an overview of the results obtained and the best solutions from Chapter 4 for the RQ2. Section 5.3 is dedicated to external feedback from experts from the "Pomoschnik" company. Section 5.4 considers possible continuation of this work to achieve larger results that go beyond the boundaries of the work.

5.1 RQ1 – Optimizing the work of employees

The numerical results shown in Table 4.1 showed that the introduction of universal EAN-13 barcodes [15] resulted **in an increase in efficiency, averaging 7.43% (56.87 s)**. Such results can be explained by the fact that instead of printing documents, the sales consultant now does not waste time on this and gives a barcode card that is behind his workstation. It is also worth emphasizing that now some of the documents are processed in parallel while the customer goes to the checkout.

In addition to the optimization, the introduction of the barcode mechanism has reduced several key points in the work of the sales consultant:

- **Optimization of cash processes**: In specific stores, barcode cards have made it possible to reduce queues at the checkout by bringing the cashier's actions to a single template

- **Structurality**: Barcodes build a consistent approach to making multiple orders, allowing you to process the incoming flow of adhesives faster

Each of the above points affects the Service Registration Time and Completion Time parameters. The influence of moments increases when we consider the specifics of the store described in Chapter 3. The presence of a second floor (the Tikhaya store) or a large store area (the Umasheva store) partially transforms the queue parameters and the order processing process.

In general, the implementation of the barcode mechanism based on feedback from users (store employees) and numerical results can be considered successful. We noted the advantages of barcode cards, which are insignificant for technical research, but significant for business:

1. **Logging the actions of store employees**: Barcodes allow you to extract valuable features, based on which you can analyze not only the work of the barcode cards, but also the speed of operations of specific employees and their productivity

2. **The peripheral takes up less space**: By giving up the printer and paper, the area of the sales consultant's workplace for work has increased

3. **Promotions**: Barcode cards allows you to place advertisements and information about the company's promotions

5.2 RQ2 – Analysis of best practices and results

The model that showed the best performance on our metrics and was tested in practice is **BERT4Rec + HDBSCAN**. The results in Table 4.1 highlight her record $\mathrm{nDCG@20}$ score. As the research [33] highlights, on the basis of which the experiments were built, the BERT4Rec architecture allows learning not only based on comparisons of orders (like other models), but also to extract the context within the order, finding patterns of customer behavior. HDBSCAN allows you to highlight and validate patterns that BERT4Rec has found, thus improving the result.

Let's compare the best results with other models and their modifications. The results obtained in Table 4.1 need to be validated based on the recommendation approach described in Chapter 3. For example, Baseline issued high recommendations based on Collaborative filtering. The baseline recommendation will not be applicable, as it does not carry any value for the consumer (useless recommend shopping package) and is not personalized, which is indispensable for increasing the number of SKUs [22].

If we compare the methods of Collaborative Filtering in isolation, then in our case item2item (best $\mathrm{nDCG@20} = 0.1754$) is superior to ALS (best $\mathrm{nDCG@20} = 0.1225$). This is due to the fact that in our interpretation of the task, the number of users (492,999) is many times higher than the number of unique products (12,881). The research [21] notes that in cases where user-item matrix has several times more users than items, item2item is the most appropriate model by evaluating the similarity between products rather than users (as ALS does). The superiority of item2item can be highlighted by a significant increase in the $\mathrm{Recall@20}$ metric, which in our case reflects the proportion of correct answers.

Using Binary matrix showed higher results than Quantity matrix. The specifics of the industrial segment described in Chapter 2, may lead us to the answer that the quantity of goods has little effect on purchases and the number of SKUs. For example, a retail customer can buy 1 can of paint (to finish painting a corner), 5 cans (to paint a wall), and 20 cans of paint (to paint an apartment). Then, we can interpret the results of the model as the fact that the quantity of goods purchased worsens the performance of the model, incorrectly prioritizing goods for recommendation systems.

Using the item embedding approach (SBERT and MiniLM) in direct use did not show high results (Item2Item + Binary Matrix + MiniLM: $\mathrm{nDCG@20} = 0.1086$). It can be assumed that despite the promising nature of the method, the vectors of product names do not have enough context to identify patterns due to business constraints, which is a prerequisite for the operation of the method [33]: The purchasing department of the ”Pomoshnik” company does not have general rules for filling in product names between all categories. Each employee of the purchasing department fills in their own group/subgroup of products according to their individual rules. Thus, a single format is preserved in a specific group/subgroup, but not among common names, so item embeddings do not work effectively.

On the other hand, the approach based on empirically powerful hierarchy, reducing dimensionality through SVD and UMAP continued the idea of item embeddings and were able to cluster data so that DBSCAN and HDBSCAN gave the best performance among all models (BERT4Rec + HDBSCAN, BERT4Rec +

DBSCAN)

In addition, if we compare two better models (BERT4Rec models), based on the study, we can also note that the first model ranks products better (the best indicator is $\mathrm{nDCG@20}= 0.2473$), and the second model predicts recommendations better (the best indicator is $\mathrm{Recall@20} = 0.3795$).

5.3 Expert validation and assessment

The results of the innovations were presented to the company's management to assess the effectiveness and correctness of the implementation.

The integration of the barcode mechanism made it possible to see the consistency of actions, as stated by the Head of the retail trade, to whom the solution was presented. He was noted that the optimization of the customer's path was performed correctly, without disrupting other business processes of the store, the mechanism significantly saves on reducing printing supplies, which account for a significant proportion of the regular costs of retail outlets.

As for the recommendation systems, the director of one of the store where the testing was conducted identified high signs of product completeness based on temporary and seasonal characteristics. For example, to recommend a specialized glue for winter work to the heat-recovery plates, instead of a universal one, which has worse properties. Such cases are often difficult for sales consultants, even with existing work experience. Generally, the recommendation system copes with such tasks.

Finally, the overall financial impact of all innovations was assessed by the company's executive director, taking into account all cost optimizations (mainly related to RQ1) and shown in Chapter 4:

1. Barcode mechanism integration (RQ1) is **overall cost optimization of 78,050,420 rubles per year** across the entire construction retail network

2. Recommendation system integration (RQ2) is **overall potential income of 42,673,540 rubles per year** across the entire construction retail network

All the innovations proposed in this work have been tested by all internal tests of the ”Pomoschnik” company for integration into the life of the company. The technology stack was approved by the head of the IT department, and the algorithms for the ERP solution modification were acceptable.

5.4 Limitations and Future work

The work has several limitations that can improve the implementation result and the overall performance of the methods used.

In future barcode implementation mechanisms (RQ1), it is worth considering the study of generated barcodes that the customer photographs/scans, which will implement another technological step in the customer path. Such an approach could optimize the approach discussed in this paper, reducing the cashier's workload.

The current version of the auxiliary recommendation system for sales consultants (RQ2) does not consider a combination of different models that would form a diverse view of the customer's current basket, which could potentially improve the results of metrics.

Another limitation of the recommendation system is to adapt the recommendation depending on the store. In future work, it would be possible to expand the results of our research by studying algorithms for dynamically ranking goods based on the specifics of the store, time of year, sales season, fashion for specific group of materials.

During the research and evaluation of metrics, marketing research can also be conducted as part of a separate work, which makes it possible to assess the impact of changes on product metrics based on statistical tests (A/B testing [41]):

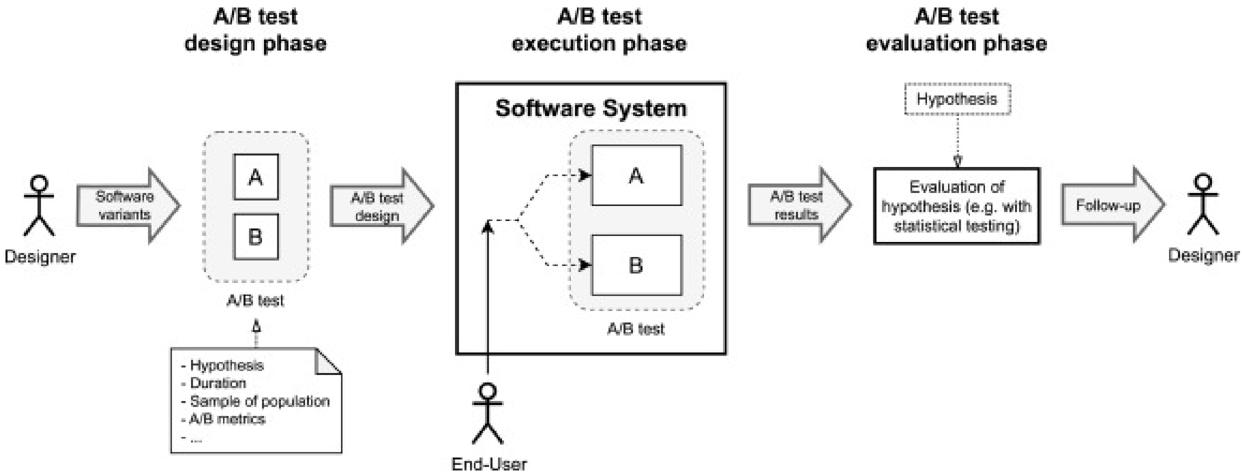


Fig. 5.1. A/B testing scheme

Studying related metrics and metrics such as the total time to download data from the server (for RQ1) or the average cost per SKU (for RQ2) will show the business result of the innovations in more detail.

Chapter 6

Conclusion

Digitalization of the industrial sector is an established process of economic development. The emergence of Artificial intelligence and accessible digital solutions accelerates technological progress, which is already a prerequisite for the existence of companies. In this work, two innovations were presented that successfully improve performance based on two key indicators: Customer checkout time and the number of SKUs in a customer order, using the example of ”Pomoschnik”, the largest chain of construction stores in the Far East.

Reducing Customer's checkout time was achieved by introducing the barcode mechanism into the work of store employees. Physical barcode cards were developed and the configuration of the company's ERP program was finalized for reading, processing and operating barcodes in the 1C programming language. The barcode cards provided by the sales consultant instead of printing the customers was able to shorten the customer's path **by an average of 7.43%** (**56.87 s**). This solution made it possible to reduce paper consumption and structure the work of employees.

An increase in the average number of SKUs in a customer order was achieved by finding the **BERT4Rec recommendation model** that is optimal based on research. The model was **modified based on HDBSCAN clustering** of client patterns based on empirically powerful hierarchy and density data analysis. The solution was able to score the best metrics: $\mathrm{nDCG@20} = 0.2473$ and $\mathrm{Recall@20} = 0.3790$. This approach was able to increase the average value of the number of SKUs in the customer order **by approximately 7.660%** (**+ 0.75 SKU items**) based on the experiments conducted to evaluate the model.

All the experiments reviewed were recognized by the construction company and accepted for implementation in the company's business processes due to their **overall financial efficiency**, which the company estimated **at 120,723,960 rubles per year**.

The research results emphasize the relevance and relevance of the digital transformation of the industrial sector. This work allows us to expand the potential for the introduction of new technologies, which will allow us to discover more advanced methods and optimize key indicators.

Bibliography cited

[1] World Bank, *World development indicators*, https://wdi.worldbank.org/table/4.2, 2023.

[2] *Visualizing the importance of the industrial economy in the world*, https://howmuch.net/articles/role-industry-around-the-world, 2019.

[3] "Information and communication technologies adoption in smes: Literature review," *J. of Small Business and Enterprise Development*, 2010, Accessed: May 22, 2025. [Online]. Available: https://www.emerald.com/insight/content/doi/10.1108/17561391011019041/full/html.

[4] CyberLeninka, *Implementation of erp systems by small and medium-sized companies in russia: Barriers and prospects*, https://cyberleninka.ru/article/n/vnedrenie-erp-sistem-malymi-i-srednimi-kompaniyami-v-rossii-bariery-i-perspektivy, 2023.

[5] ResearchGate, *Erp solutions as a factor of supporting business structures*, https://www.researchgate.net/publication/357703008_ERP_SOLUTIONS_IS_A_FACTOR_OF_SUPPORTING_BUSINESS_STRUCTURES, 2021.

[6] IEOM Society, *Optimization of order processing procedures to reduce order execution time*, https://index.ieomsociety.org/index.cfm/article/view/ID/11421, 2022.

[7] CyberLeninka, *Improving the efficiency of production processes by optimizing business processes of technological processing of orders*, https://cyberleninka.ru/article/n/povyshenie-effektivnosti-proizvodstvennyh-protsessov-za-schet-optimizatsii-biznes-protsessov-tehnologicheskoy-prorabotki-zakazov, 2020.

[8] Academia.edu, *Online inventory and order management system for pharmacy*, https://www.academia.edu/76667631/Online_Inventory_and_Order_Management_System_for_Pharmacy?auto=download, 2018.

[9] "Data transmission using rfid system on smart shopping carts for checkout process efficiency in supermarket at indonesia," *Procedia Computer Science*, vol. 179, pp. 674–681, 2021, Accessed: May 22, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050921001101.

[10] Self-Checkout Study Group, *An evaluation of self-checkout systems*, https://www.wsj.com/articles/uniqlos-parent-company-bets-big-on-tiny-rfid-chips-600b124f, 2007.

[11] R. Zhang, W. Li, and M. Chen, "Enhanced self-checkout system for retail based on improved yolov10," *Journal of Imaging*, vol. 10, no. 10, p. 248, 2024. DOI: 10.3390/jimaging10100248.

[12] The Wall Street Journal, *Uniqlo's parent company bets big on tiny rfid chips*, https://www.wsj.com/articles/uniqlos-parent-company-bets-big-on-tiny-rfid-chips-600b124f, 2023.

[13] Japan Insides, *Uniqlo self checkout is 50% faster in japan with rfid technology*, https://japaninsides.com/uniqlo-self-checkout-is-50-faster-in-japan-with-rfid-technology-39569, 2023.

[14] F. Yamazaki, "Bar code applications in construction," *Journal of Construction Engineering and Management*, vol. 114, no. 2, pp. 263–278, 1988. DOI: 10.1061/(ASCE)0733-9364(1988)114:2(263).

[15] International Journal of Science and Research (IJSR), *Generating ean-13 standard barcodes*, https://www.ijsr.net/archive/v3i6/MDIwMTQ0OTg=.pdf, 2012.

[16] International Journal of Engineering and Technology, *A development of practical cost control system for small and medium-sized contractors*, https://www.ijetch.org/papers/358-P10048.pdf, 2012.

[17] M.-C. Tsai and C.-Y. Yang, *Mobile construction supply chain management using pda and bar codes*, https://ir.lib.nycu.edu.tw/bitstream/11536/13559/1/000229288500002.pdf, 2005.

[18] ResearchGate, *An application of auto-identification system in steel structural construction process*, https://www.researchgate.net/publication/320560887_An_Application_of_Auto-Identification_System_in_Steel_Structural_Construction_Process, 2004.

[19] ResearchGate, *Personalized product recommendation and user satisfaction: Reference to industry 5.0*, https://www.researchgate.net/publication/

382110524_The_Role_of_Artificial_Intelligence_in_Personalized_Marketing_Strategies, 2023.

[20] S. Nahavandi, "State of industry 5.0—analysis and identification of current research trends," *Journal of Applied Sciences*, vol. 5, no. 1, p. 27, 2022. DOI: 10.3390/app12010427.

[21] F. Ricci and L. Rokach, "Recommendation systems: Algorithms, challenges, metrics, and business opportunities," *Applied Sciences*, vol. 10, no. 21, p. 7748, 2020. DOI: 10.3390/app10217748.

[22] ResearchGate, *The role of artificial intelligence in personalized marketing strategies*, https://shorturl.at/AN9gS, 2024.

[23] McKinsey & Company, *The value of getting personalization right—or wrong—is multiplying*, https://www.mckinsey.com/capabilities/growth-marketing-and-sales/our-insights/the-value-of-getting-personalization-right-or-wrong-is-multiplying, 2021.

[24] ResearchGate, *Research on recommendation systems using deep learning models*, https://www.researchgate.net/publication/357620803_Research_on_Recommendation_Systems_using_Deep_Learning_Models, 2022.

[25] Elsevier, *Beyond movie recommendations: Solving the continuous cold start problem in e-commerce recommendations*, https://www.sciencedirect.com/science/article/pii/S0164121224000542, 2016.

[26] ResearchGate, *Sequence recommendation based on deep learning*, https://www.researchgate.net/publication/369048688_Sequence_Recommendation_based_on_Deep_Learning, 2023.

[27] F. Sun, J. Liu, J. Wu, *et al.*, "Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer," in *Proc. 28th ACM Int. Conf. on Information and Knowledge Management (CIKM)*, 2019, pp. 1441–1450.

[28] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, "Autorec: Autoencoders meet collaborative filtering," *Proc. 24th Int. Conf. on World Wide Web (WWW)*, pp. 111–112, 2015. [Online]. Available: https://users.cecs.anu.edu.au/~akmenon/papers/autorec/autorec-paper.pdf.

[29] A. Da'u and N. Salim, "Ganrec: Gan-based recommendation system using collaborative filtering," *Proc. 12th ACM Conf. on Recommender Systems*, pp. 514–518, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S095741742202173X.

[30] IBM, *Order management systems (oms) explained*, https://www.ibm.com/cloud/learn/order-management-system, 2021.

[31] "Using the interquartile range in infection prevention and control research," *Infection Prevention in Practice*, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2590088924000015.

[32] Y. Tamm, R. Damdinov, and A. Vasilev, *Quality metrics in recommender systems: Do we calculate metrics consistently?* https://arxiv.org/pdf/2206.12858, 2022.

[33] R. Gonzalez-Marquez and D. Kobak, "Learning representations of learning representations," in *Data-centric Machine Learning Research (DMLR) Workshop at ICLR 2024*, 2024. [Online]. Available: https://arxiv.org/pdf/2404.08403.

[34] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proc. 2019 Conf. on Empirical Methods in Natural Language Processing (EMNLP)*, 2019, pp. 3982–3992.

[35] SberDevices, *Natural language model training with bert and tensorflow*, https://habr.com/ru/companies/sberdevices/articles/527576/, 2020.

[36] A. Hasanvand, *An introduction to singular value decomposition*, https://www.researchgate.net/publication/370402152_An_Introduction_to_Singular_Value_Decomposition, 2023.

[37] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[38] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD)*, 1996, pp. 226–231.

[39] S. Weglarczyk, *Kernel density estimation and its application*, https://www.researchgate.net/publication/328785939_Kernel_density_estimation_and_its_application, 2018.

[40] L. McInnes, J. Healy, and S. Astels, "Hdbscan: Hierarchical density based clustering," *Journal of Open Source Software*, vol. 2, no. 11, p. 205, 2017. DOI: 10.21105/joss.00205.

[41] Elsevier, *A/b testing: A systematic literature review*, https://www.sciencedirect.com/science/article/pii/S0164121224000542, 2024.