Cukurova University Journal of Natural & Applied Sciences (2024) 3(2): 36-44

e-ISSN: 2822-2938

Cukurova University
Journal of
Natural & Applied Sciences

Original Paper

Design and Development of a Fashion Oriented Personalized Search Engine

T. Bahadır YALIN[1], A. Akyüz TUNÇ[1], Fatih ABUT[2], M. Fatih AKAY[2], Ceren ULUS[2]

[1]Trendyol, Department of Data Science, İstanbul, Turkey
[2]Cukurova University, Faculty of Engineering, Department of Computer Engineering, Adana, Turkey

ORCID IDs of the authors: T.B.Y. 0009-0003-9005-7405; A.A.T. 0000-0003-3456-2936; F.A. 0000-0001-5876-4116; M.F.A. 0000-0003-0780-0679; C.U. 0000-0003-2086-6381.

Cite this article as: Yalın, T.B., Tunç, A.A., Abut, F., Akay, M.F., Ulus, C. (2024). Design and Development of a Fashion Oriented Personalized Search Engine, Cukurova University Journal of Natural & Applied Sciences 3(2): 36-44. https://doi.org/10.70395/cunas.1515178.

Abstract

Finding the desired product in the e-commerce sector in the fastest and easiest way plays a vital role in customer satisfaction and revenue growth. Some interactive search engines have already been proposed in the literature and allow for text or visual queries. Nevertheless, these studies focus on finding items aesthetically similar to the query without considering the query personalization aspects. Query personalization allows that user preferences are provided as a user profile separately from the query and dynamically decide how this profile will affect the query results. In this study, a personalized search engine is proposed, which is fed with the product data of the e-commerce site trendyol.com operating in the fashion-oriented retailing sector. More specifically, a search engine has been developed to recognize and help online shoppers find what they are looking for and discover a broader and more relevant range of products in the trendyol.com catalog. The index, search, and data collection infrastructures and a brand-based user-segmented product listing algorithm have been designed and implemented to realize the search engine. As the outcome of the study, a fashion-oriented and personalized site search has been enabled thar successfully reveals products that have never been thought of before by directly associating the products the customers want. The results show that personalizing the search queries increase the odds of success. With the development of the personalized search engine, it is expected that Trendyol's revenues will grow in a short time through users visiting the site.

Keywords: Information Retrieval, Prediction, Query Personalization, Search Engine.

1. Introduction

An online product search engine, sometimes referred to as a query response module or program, allows buyers to filter and compare products based on price, features, ratings, and other criteria. Most online product search engines in the e-commerce sector aggregate product listings from many different retailers, but they don't sell products directly themselves, instead of making money from affili-ate marketing agreements. For detailed information about the structure of product search engines, the reader is referred to the respec-tive publications. An online product search engine, sometimes referred to as a query response module or program, allows buyers to filter and compare products based on price, features, ratings, and other criteria. Most online product search engines in the e-commerce sector aggregate product listings from many different retailers, but they don't sell products directly themselves, instead of making money from affiliate marketing agreements. For detailed information about the structure of product search engines, the read-er is referred to the respective publications [1-3].

Finding the desired product in the e-commerce sector in the fastest and easiest way plays a vital role in customer satisfaction and revenue growth. Designing an interactive search engine is required for several reasons, including (a) to avoid typos and narrow the

Address for Correspondence:
Ceren Ulus, e-mail: f.cerenulus@gmail.com

Received: Jul 12, 2024
Accepted: Aug 1, 2024

searches, (b) refine visitors' search queries and category listings, and (c) program and optimize the speed, ranking, filters, and face of the category pages.

Some interactive search engines have already been proposed in the literature and allow for text or visual queries, as outlined in Section 2. Nevertheless, these studies focus on finding items aesthetically similar to the query without considering the query personalization aspects. Query personalization allows that user preferences are provided as a user profile separately from the query and dynamically decide how this profile will affect the query result.

Personalized search engines analyze users' past search behaviors and interests to quickly provide the most accurate results based on this analysis. These engines save users time, reduce information pollution by offering more relevant results, and enhance the user experience. They also stand out as an important tool for advertisers by providing more effective marketing strategies through targeted ads.

This study proposes a personalized search engine, which is fed with the product data of the e-commerce site trendyol.com operating in the fashion-oriented retailing sector. The contributions of the study can be summarized as follows:

- A personalized search engine has been developed to recognize and help online shoppers find what they are looking for and discover a broader and more relevant range of products in the trendyol.com catalog.
- The index, search, and data collection infrastructures and a brand-based user-segmented product listing algorithm have been designed and implemented to realize the search engine.
- A fashion-oriented and personalized site search has been enabled that successfully reveals products that have never been thought of before by directly associating the products the customers want.

The rest of the paper is organized as follows. Section 2 outlines the related works. Section 3 introduces the details of indexing, search, and data collection infrastructures. Section 4 describes the brand-based user-segmented product listing algorithm. Section 5 presents the results and discussion. Finally, Section 6 concludes the paper along with possible future works.

2. Related Works

This section first outlines the current textual and visual search solutions offered in the literature. Afterward, the differences between the current study and studies from related literature are summarized.

The first methods that proposed to address textual information retrieval were based on token counts, e.g. Bag-of-Words [4] or TF-IDF [5]. Later, [6] designed two model architectures for computing continuous vector representations of words from extensive data sets. The quality of these representations is measured in a word similarity task, and the results are compared to the previously best-performing techniques based on different types of neural networks. Significant improvements in accuracy at much lower computa-tional cost were reported. [7] proposed GloVe, a new global log-bilinear regression model for the unsupervised learning of word representations that outperforms other models on word analogy, word similarity, and named entity recognition tasks. GloVe com-bines the advantages of the two prominent model families in the literature: global matrix factorization and local context window methods. [8] proposed an intelligent retrieval method based on the semantic characteristics of intelligent retrieval. This methodolo-gy performs research from both the document retrieval model and user perspective. [9] proposed a multimodal search engine that combines visual and textual cues to retrieve items from a multimedia database aesthetically similar to the query. The goal of the engine is to enable the intuitive retrieval of fashion merchandise such as clothes or furniture. [10] proposed search engine using ML tech-nique that will give more relevant web pages at the top for user queries. [11] proposed a personalized ranking mechanism based on a user's search and click history. [12] proposed a deep learning-based search ranking framework that can expeditiously update the ranking model by capturing real-time user clickstream data. [13] proposed a framework that combines contextual information with advanced information representation techniques to help existing web tools understand domain-specific concepts and provide more accurate and personalized results. This framework, which integrates information representation and long-term memorization, has been found to enhance the search engine experience, offering the potential for more personalized and contextually accurate results. [14] aimed to develop a collaborative search engine that helps reach specific decisions by improving search quality through the inte-gration of user experiences with hierarchical user profiles. They uploaded the collected data to recommendation servers of search engines like Google/Yahoo. [15] presented a new personalized search engine based on user keyword profiles and long-tail keyword optimization techniques, aiming to provide better results in terms of relevance, recall rate, and accuracy compared to existing search engines. The model was developed using the Hypertext Preprocessor platform for simulation. [16] proposed a Cognitive Personal-ized Search (CoPS) model that integrates Large Language Models (LLMs) with a cognitive memory mechanism inspired by human cognition. CoPS, which uses LLMs to enhance user modeling and search experience, used a three-step approach supported by sen-sory memory for quick responses, working memory for sophisticated cognitive responses, and long-term memory for storing histor-ical interactions. According to the experimental results, CoPS was indicated to outperform baseline models in zero-shot scenarios.

Nevertheless, these previous studies focus on finding items aesthetically similar to the textual or visual query without considering the query personalization aspects. Unlike the studies in the literature, a personalized search engine is proposed, which is fed with the product data of the e-commerce site trendyol.com operating in the fashion-oriented retailing sector. In comparison to the studies from the related literature, the study has many innovative aspects;

- Search engine can index and merge different formats such as Comma-Separated Values, eXtensible Markup Language, and JavaScript Object Notation from various sources.
- Search engine can dramatically edit the search phrase when the search engine cannot find a match in the subword, e.g. by changing letters.
- Search engine supports multiple languages.
- Option to show items from the same store/brand/campaign side by side in search results.
- All network traffic is recorded and analyzed in MSSQL database for volume and customer behavior. Different statistics such as response time and memory usage are monitored.

3. The Purposed Fashion Oriented Personalized Search Engine Figures

Three teams worked together for the development of the Trendyol product sorting mechanism. The Indexing team developed the system where the stock, price, campaign, name, and scores of the products are kept. The Search team created a product pool based on the search term when a search occurs. Data Science creates the scores that will be used to rank the products.

The Data Science team, which calculates the product scores, transfers the data to the Indexing team with the Kafka manager tool. These scores are processed in the database where the information of the products is stored. The Search team, on the other hand, creates the product pool using the database created by the indexing team when a search request comes in and ranks the products according to the scores created by the Data Science team.

For segmented product ranking, the system is managed through different score IDs created on the products. There is a database where customization documents are stored. In this database, the products in the search result of the users who have the document are sorted by the score corresponding to the ID assigned to them. In this way, products are sorted differently for users. Figure 1 illustrates how the entire system works in an end-to-end manner, i.e., what the input is, how the data flows, what each component does, and finally, the output.

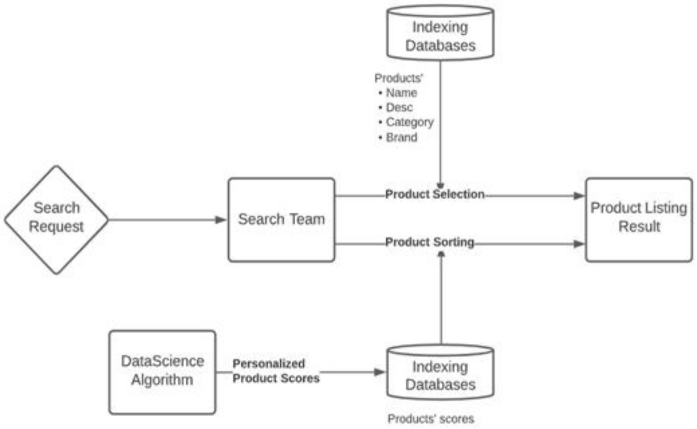

Figure 1. Overview of the Entire System Working in an End-To-End Manner

3.1. Indexing, Search and Data Collection Infastructures

The indexing infrastructure is one of the essential parts of the personalized search engine architecture. The primary goal of this infra-structure is to index millions of documents very quickly and without compromising data consistency. The sources from which the data is collected and the structure that will index the data must work quickly and consistently. To this end, a study has been carried out on which data are needed primarily. According to the results, it was necessary to collect and index the data from approximately ten different places. These places of data include (a) product master data, (b) campaigns to which the product is affiliated, (c) promo-tions that can be applied to the product, (d) suppliers of the product, (e) supplier-based prices of the product, (f) supplier-based stocks of the product, (g) the scores of the product to be used for ranking, (h) information on which supplier will stand out when showing the product, and (i) product brands and categories.

As we prefer the microservice architecture, nine different services are required to collect all these data. On the other hand, R&D stud-ies have been carried out to rapidly index millions of data from so many various sources. The actor model was preferred for indexing a large amount of data. According to the actor model approach, after collecting common data, thousands of actors at the same time will process thousands of products in parallel and allow them to be indexed. After deciding on the indexing model, it was time to choose which programming language to encode it with. Options included .NET Core, JAVA, and Scala. As the language in which the actor model works best and effectively, Scala came to the fore in all our research.

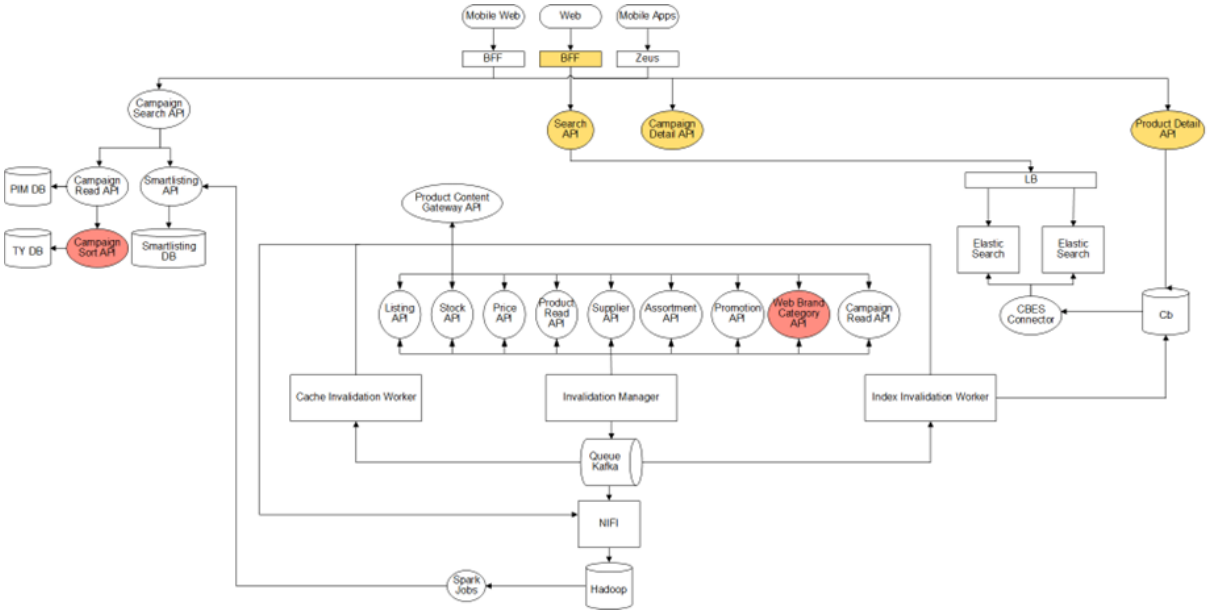

Figure 2. Indexing Infrastructure

Figure 2 shows the indexing infrastructure. Product Content Gateway API collects all the required data like stock, price, promotion etc. from different APIs to create content data. Index Invalidation Worker gets the content data from Product Content Gateway API and sends them to our ElasticSearch and Couchbase data sources. Product Detail Services use Couchbase data source and Search Services use ElasticSearch data sources. Content data is sent to Hadoop via Nifi jobs. Our Spark jobs use these data to calculate listing and campaign scores. Campaigns and contents are sorted according to these scores.

The design of the search infrastructure was the work we started in parallel with the design of the indexing infrastructure. When we reveals our needs in this process, features such as searching, filtering, sorting, error correction, and automatic suggestions ap-peared. The ElasticSearch structure, which we can use as an open source for both the suggestion system and search pages, has been pre-ferred. Figure 3 illustrates the designed search infrastructure.

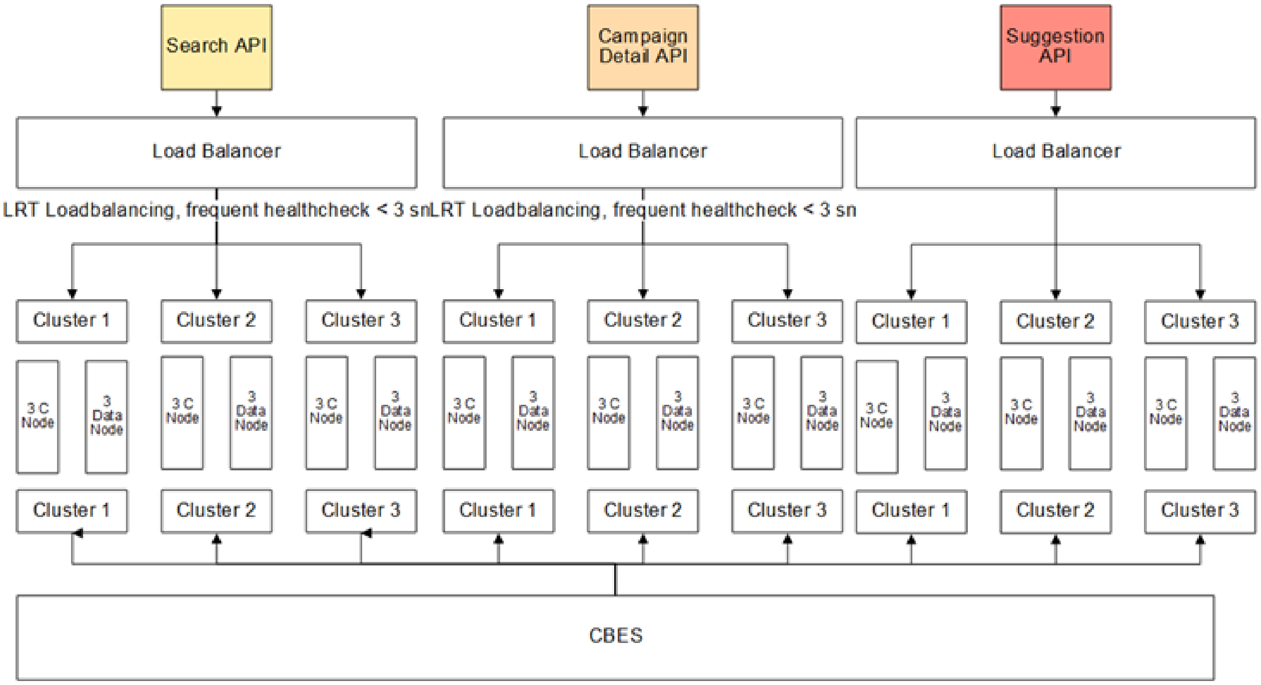

Figure 3. Search Infrastructure

The clusters shown in Figure 3 are ElasticSearch clusters that contain multiple nodes. We have both master nodes responsible for the cluster's health, and data nodes responsible for writing and reading operations. We have tried many configurations for the performance and found out that master nodes and data nodes are essential. The indexing system is responsible for write operations. Search and Campaign Detail APIs provide search services. They convert different types of requests to queries which will be run on ElasticSearch clusters. Suggestion API is an autocomplete service that suggests related keywords while the user starts typing the keyword for search. Since there are multiple ElasticSearch clusters behind each service, load balancer systems have been enabled to cause equal load on each cluster and provide high availability. The load balancer is checking the health of each cluster and forwards queries in the order of Least Response Time (LRT) pattern. So, the faster the clusters the more queries they execute.

Another critical decision is the design of the data collection infrastructure that we will use for personalization. It must serve the personalization infrastructure by collecting all the customers' activities, even under high traffic. Since the size of the data to be collected would be very large, the cloud seemed to be the best solution. We need to collect and process terabytes of data. From an architectural point of view, we thought of a structure as illustrated in Figure 4. Our clickstream application collects user events from our several clients (i.e., web, mobile web, and mobile apps). Once data has been sent to the application, it immediately stores it on several topics in our Apache Kafka cluster. An Apache Flume application continuously ingests topics and writes the user data to object storage to the cloud provider we are using. Our batch Apache Spark applications run in a scheduled way, process new raw data in object storage, and store it in object storage again. We have many tables in Apache Hive Metastore, pointing to the processed data paths in object storage, so processed data can be queried via Apache Hive or Presto query engine. Lastly, reporting applications like Tableau use Presto to create analytics reports.

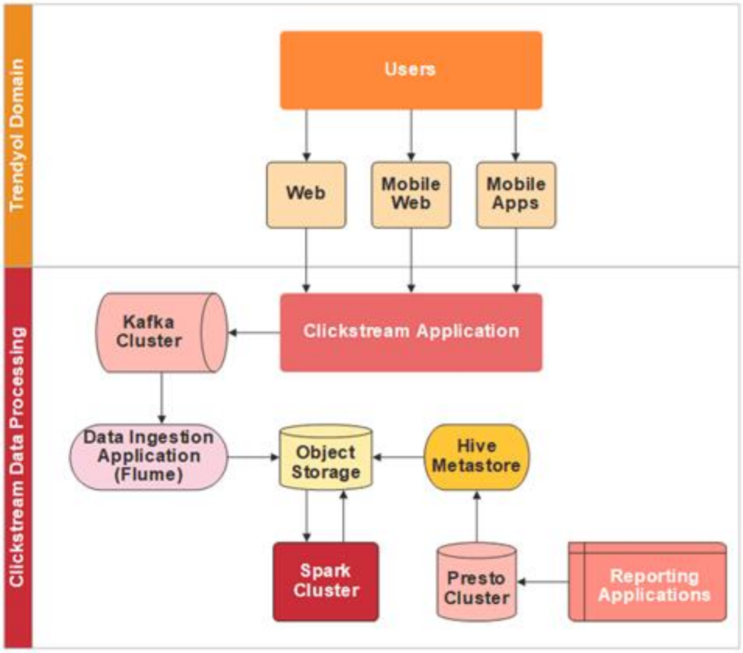

Figure 4. Data Collection Architecture

3.2. Brand-Based User-Segmented Product Listing Algorithms

The purpose of the brand-based user-segmented product listing algorithm is to prioritize the products the user may be interested in by making segment-based product sorting. To make segment-based product sorting, user segments must be created first. To create user segments, brands are first divided into segments. Then, users were assigned to these brand segments so that user segments were created. Then, product sales prediction models specific to user segments were designed and product scores were studied for segment-based product ranking.

Products belonging to approximately 550 different brands in the fashion category are used for brand segmentation. Three different methods were studied for segmentation. The first method segments the brands into different numbers of quantiles only according to their prices (such as cheap, medium, and expensive brands). The second approach intends to create features of brands (i.e., price, relative price to category, ratio, filter count, impression 14Day etc.) and make a segmentation on this feature space. Another method is to use BuytoBuy, SearchtoSearch and VisittoVisit similarity metrics. The rationale is to calculate BuytoBuy similarity between brands A and B, while calculating a similarity between sessions where brand A sells and sessions where brand B sells, using the intersection over union (IoU) method. This value should always be between 0 and 1 regardless of its volume (i.e., brands A and B never sold in the same session = 0; brands A and B always sold in the same session = 1; and being symmetrical (sim (A, B) = sim (B, A)). Subsequently, a brand similarity matrix of 550x550 is formed, and brand clusters are formed by clustering methods. DBSCAN, K-Means, and K-Medoids were used as clustering methods. With these methods, brands are clustered so that the number of clusters varies between 1 and 20.

Assigning users to brand clusters answers the following question: For every user, what is the most likely brand-cluster that he/she will place an order? To this end, shopping possibilities for all brand clusters are calculated for each user with logistic regression. The model calculates the probability of a user to shop on a brand in the next seven days based on the values of product visit count, favorite product count, product order count, product basket count, product order count target in the past 28 days. Users are assigned the brand segment with the highest probability to shop.

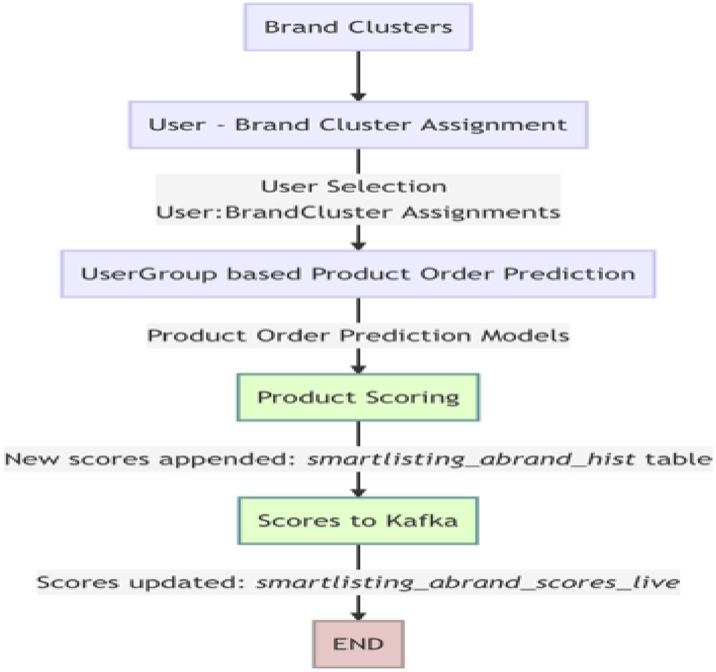
Brand Clusters
User - Brand Cluster Assignment
User Selection
User:BrandCluster Assignments
UserGroup based Product Order Prediction
Product Order Prediction Models
Product Scoring
New scores appended: *smartlisting_abrand_hist* table
Scores to Kafka
Scores updated: *smartlisting_abrand_scores_live*
END

Figure 5. Flowchart of Brand-Based User-Segmented Product Listing Algorithm

Figure 5 illustrates the steps followed by the brand-based user-segmented product listing algorithm. First, the best brand sets are extracted according to the metrics we use (Brand Clusters). Then, the probability of purchasing products from these sets is calculated for each user, and these users are included in the brand sets (User - Brand Cluster Assignment). For each cluster created by the users defined in the brand sets, the purchasing forecast of all products on the site is modeled (UserGroup based Product Order Prediction). Hourly script runs to change product scores on the site, and order predictions are prepared by normalizing in accordance with the score scale on the site. Product and score information is added to the smartlisting_abrand_hist table at the end of each job (Product Scoring), where historical data is kept. Product scores are left to Queue Kafka shown in Figure 1, and current product scores are held in the smartlisting_abrand_scores_live table (Scores to Kafka).

For each user segment, a product-based sales forecast model was trained using only people's purchasing behavior in that user segment. Here, the sales volume of the products in the next 12 hours from the user segment data of the last one month is estimated.

4. Results and Discussion

As the result of this research, a personalized search engine, fed with the product data of the e-commerce site trendyol.com, has been developed to recognize and help online shoppers find what they are looking for and discover a broader and more relevant range of products in the trendyol.com catalog. In more detail, the following activities were carried out and successfully implemented within the scope of the project;

- Real-time synchronization and catalog indexing,
- Combining and synchronizing different data sources and formats,
- Creating a list of related results in addition to direct results,
- Showing query suggestions that will lead to similar results such as the original query,
- Making suggested spelling corrections,
- Automatically solving minor typos and generating results without any human intervention,
- Searching for other things than products (e.g., advertisements, blogs, articles, etc.),
- Determining popular queries and auto-complete suggestions and suggestions according to the frequency of popularity,
- Development of static synchronization filters determined by order such as out of stock, available in the store, not yet published.

- Development of filters created based on live dynamic user information,
- Creating a ranking according to popularity,
- Sorting products by any attribute (e.g., popularity, price, newest / oldest, etc.),
- Listing results and categories based on a user's unique personal taste using the browser cookie or user ID,
- Making reflections on all devices used by a person with cross-device personalization.

The proposed personalized search engine went live on January 2022, and its efficiency has been compared to our previous platform based on quarter 1 (Q1) of the years 2021 and 2022. Table 1 shows the comparison results.

Table 1. Comparing the personalized search engine and our previous platform

	Previous Search Platform	**Proposed Search Platform**
Listing rate of products that personally fits users	2021 Q1: 33%	2022 Q1: 72%
Loading time of search results	2021 Q1: 0.83 s	2022 Q1: 0.65 s
Users shopping rate	2021 Q1: 69%	2022 Q1: 81%
Query personalization	n/a	Personalized site search
Fashion-oriented search capabilities	n/a	Available

5. Conclusion and Future Work

In this study, a fashion-oriented and personalized site search was enabled that successfully reveals products that have never been thought of before by directly associating the products the customers wanted. The index, search, and data collection infrastructures and a brand-based user-segmented product listing algorithm have been designed to realize the search engine. The results show that personalizing the search queries increase the odds of success. With the development of the personalized search engine, it is expected that Trendyol's revenues will grow in a short time through users visiting the site.

With the experience and results we gained in this study, we need to include more users in the personalization structure with the growing product pool and user base. We aim to increase the number of clusters we had created from 5-6 levels to 100-500 clusters in the next steps. However, we are planning to obtain more accurate product sets for all product groups on the site by adding non-fashion products to the product sets we use in clustering. We aim to provide the user with a dynamic and personal experience in all search results on Trendyol by meeting the computational power needed in the new structure with more effective and scalable data flows.

References

[1] Aravindhan, R., Shanmugalakshmi, R. (2014). Comparative analysis of Web 3.0 search engines: A survey report. In 2013 International Conference on Advanced Computing and Communication Systems; IEEE; pp. 1-6. https://doi.org/10.1109/ICACCS.2013.6938715

[2] López-Ortiz, A. (2005). Search engines and web information retrieval. Lecture Notes in Computer Science; 3405: pp. 183-191. https://doi.org/10.1007/11527954_18

[3] Sánchez, D., Martínez-Sanahuja, L., Batet, M. (2018). Survey and evaluation of web search engine hit counts as research tools in computational linguistics. Information Systems; 73: pp. 50-60. https://doi.org/10.1016/j.is.2017.12.007

[4] Harris, Z. S. (1954). Distributional Structure. WORD; 10(2–3): pp. 146-162. https://doi.org/10.1080/00437956.1954.11659520

[5] Salton, G., Wong, A., Yang, C. S. (1975). A Vector Space Model for Automatic Indexing. Communications of the ACM; 18(11): pp. 613-620. https://doi.org/10.1145/361219.361220

[6] Mikolov, T., Chen, K., Corrado, G., Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space; arXiv preprint arXiv:1301.3781.

[7] Pennington, J., Socher, R., Manning, C. D. (2014). GloVe: Global vectors for word representation. EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing; Proceedings of the Conference; pp. 1532-1543. https://doi.org/10.3115/v1/d14-1162

[8] Fan, J., Gao, X., Wang, T., Liu, R., Yang, Y. (2021). Research and Application of Automated Search Engine Based on Machine Learning. International Conference on High Performance Big Data and Intelligent Systems (HPBD&IS); pp. 69-73. https://doi.org/10.1109/HPBDIS53214.2021.9658474

[9] Tautkute, I., Trzcinski, T., Skorupa, A. P., Lukasz, K., Marasek, K. (2019). DeepStyle: Multimodal Search Engine for Fashion and Interior Design. IEEE Access; 7: pp. 84613-84628. https://doi.org/10.1109/ACCESS.2019.2923552

[10] Karwa, R., Honmane, V. (2019). Building search engine using machine learning technique. International Conference on Intelligent Computing and Control Systems; pp. 1061-1064. https://doi.org/10.1109/ICCS45141.2019.9065846

[11] Yoganarasimhan, H. (2019). Search Personalization Using Machine Learning. Management Science; 66(3): pp. 1045-1070. https://doi.org/10.1287/MNSC.2018.3255

[12] Li, Y., Jiang, Y., Yang, C., Yu, M., Kamal, L., Armstrong, E. M., Huang, T., Moroni, D., McGibbney, L. J. (2020). Improving search ranking of geospatial data based on deep learning using user behavior data. Computers & Geosciences; 142; 104520. https://doi.org/10.1016/J.CAGEO.2020.104520

[13] Gupta, S., Tiwari, K. K. (2024). Search Query Refinement Using Context, Knowledge and Long-term Memorization. Asian Journal of Research in Computer Science; 17(2): pp. 27-36.

[14] Murshleen, M., Prabhu, A., Jain, G. (2024). Personalized Collaborative Search Engine for Decision Making. In 2024 International Conference on Communication, Computer Sciences and Engineering (IC3SE); IEEE; pp. 202-206.

[15] Raghuvanshi, M. K., Bele, M. S. (2024). Novel Design and Implementation of the Personalized Search Engine in Context with User Keywords Profile and Keywords Optimization; Recent Advancements in Science and Technology; pp. 306-310.

[16] Zhou, Y., Zhu, Q., Jin, J., Dou, Z. (2024). Cognitive personalized search integrating large language models with an efficient memory mechanism. In Proceedings of the ACM on Web Conference 2024; pp. 1464-1473.