SDU University

UDC УДК 002.6-027.21; 002.6:001.8

ORYNBEKOVA KAMILA YERMUKHANBETOVNA

Advisory system for adapting a single machine problem to a distributed solution

6D070400 – Computing Systems and Software

Thesis for the degree of
doctor of philosophy (PhD)

Scientific supervisors
doctor PhD,
assoc. professor.
A.V. Bogdanchikov

Foreign scientific supervisor
doctor PhD,
assoc. professor
A. Adamov

Republic of Kazakhstan
Kaskelen, 2024

CONTENTS

NORMATIVE REFERENCES

This thesis uses references to the following standards:

These instructions are based on interstate standards GOST 7.1-2003. Bibliographic record. Bibliographic description. General requirements and rules of compilation.

Rules for awarding degrees, approved by Order of the Minister of Education and Science of the Republic of Kazakhstan dated September 28, 2018 No. 512.

ABBREVIATIONS

AI	– artificial intelligence
API	– application programming interface
AS	– Apache Spark
BD	– Big Data
CB	– Content-Based
CNN	– convolutional neural network
CPU	– a central processing unit
DL	– Deep Learning
DB	– Demographic-Based
DM	– Decision-Making
DS	– Distributed Systems
E	– Electronic
HDFS	– Hadoop Distributed File System
HR	– Human Resource
KB	– Knowledge-Based
LR	– Logistic Regression
LSTM	– Long-Short-Term memory
MBTI	– Myers–Briggs Type Indicator
ML	– Machine Learning
MR	– MapReduce
NB	– Naive Bayes
NLP	– Natural Language Processing
NN	– neural network
PS	– Problem-Solving
RAM	– Random-Access Memory
RDD	– Resilient Distributed Datasets
RGB	– Red, Blue, Green
RNN	– Recurrent neural networks
RS	– Recommendation system
RW	– Real-World
SA	– Sentiment analysis
SM	– Social Media
SVM	– Support vector machine
TB	– Text-Based
TF-IDF	– Term Frequency Inverse Document Frequency
XGBoost	– Gradient Boosting

INTRODUCTION

General characteristics of the work. The work encompasses developing an advisory system to recommend solutions for single-machine problems adaptable to distributed systems, mainly focusing on implementation within the MapReduce platform. Methodologically, an experiment evaluated learning effectiveness, while extensive data collection informed model development. Predictive models, including Naive Bayes and Logistic Regression, were optimized and integrated into a recommendation system validated through rigorous evaluation.

The **aim of the research** is to develop an advisory system that recommends single-machine problem solutions that adapt to distributed systems and are suitable for implementation on the MapReduce platform.

Relevance of the work. In the rapidly evolving big data analytics landscape, scalable and efficient data processing systems are paramount. The transition from single-machine to distributed solutions has become imperative to effectively handle the ever-increasing volumes of data generated across various domains. MapReduce, a distributed computing paradigm popularized by Google, and Apache Spark, a fast and general-purpose cluster computing system, stand out as leading frameworks for processing large datasets efficiently across clusters of machines.

Scalability and Performance Optimization: Using MapReduce and Apache Spark to transition single-machine problem solutions to distributed systems offers significant scalability and performance benefits. These frameworks enable parallel data processing across multiple nodes, facilitating faster computation and analysis. This scalability aspect is emphasized by Karau et al. [1], who underscore the importance of scalable architectures for efficient big data processing.

Cost-Effectiveness: Distributed systems, particularly those based on MapReduce and Apache Spark, provide cost-effective solutions for handling big data workloads. Organizations can achieve high-performance computing without substantial investments in specialized infrastructure by leveraging commodity hardware and parallel processing techniques. This aligns with the findings of Chen et al. [2], who highlight the cost-effectiveness of MapReduce and Apache Spark-based solutions for large-scale data analytics.

Enhanced Data Processing Capabilities: Adopting MapReduce and Apache Spark facilitates efficient processing of diverse data types, including structured and unstructured data. By adapting single-machine problem solutions to distributed systems using these frameworks, organizations can harness their versatility for analyzing complex datasets more effectively. This is supported by the research of Shaikh et al. [3], who demonstrate the adaptability of MapReduce and Apache Spark for processing various types of data in distributed environments.

Real-Time Decision-Making: Real-time data processing and analytics enable timely decision-making in dynamic environments. Implementing an advisory system recommending MapReduce and Apache Spark-based solutions for distributed data processing can expedite decision-making processes. This notion is corroborated by the study conducted by Ortiz et al. [4], which emphasizes the importance of real-time analytics in facilitating agile decision-making.

Fields of Application: MapReduce and Apache Spark are widely used in various fields, including finance [5], healthcare [6, 7], e-commerce [8, 9], social media analytics [10], and scientific research [11, 12]. These frameworks have been instrumental in analyzing large-scale datasets and extracting actionable insights across diverse domains. This multidisciplinary applicability underscores the versatility and relevance of transitioning single-machine problem solutions to distributed systems using MapReduce and Apache Spark.

The relevance of the research is further underscored by the call to adapt legislation to new technological phenomena, as highlighted in the Address to the People of Kazakhstan dated September 2, 2019, by the President of the Republic [13]. The need to align legislation with emerging technologies such as "smart cities," big data, blockchain, digital assets, and new digital financial instruments emphasizes the importance of advancing knowledge and practical applications in areas like distributed systems and big data processing.

By developing effective recommender systems for transitioning single-machine problem solutions to distributed systems within the MapReduce framework, the research contributes to the understanding and utilization of cutting-edge technologies in line with the evolving legislative landscape. This alignment with the national agenda for technological adaptation further emphasizes the significance of the research in addressing contemporary challenges and opportunities in the digital era.

The Objectives of the Research. In this dissertation, five primary goals are established for the research:

1. Exploring and creating an innovative method for teaching MapReduce and Apache Spark's fundamental concepts. This involves balancing theoretical explanations and practical exercises with real-world instances, enriching users` comprehension of these technologies.
2. Assessing the efficacy of a classification technique in enhancing user learning outcomes regarding MapReduce and Apache Spark within distributed systems. Additionally, the objective is to offer feedback for refining curriculum design based on this evaluation.
3. Collecting a diverse dataset representing problems encountered in real-world MapReduce applications from prominent books and scientific articles, covering fundamental concepts and advanced topics in distributed systems and big data processing.
4. Development of a Recommendation System based on trained classification models to accurately forecast the assignment of categorical labels to novel problem instances.
5. Evaluation of Recommender System that undergoes rigorous evaluation procedures to assess its predictive efficacy, including expert opinion analysis.

Methods of the Research. The research employed a multifaceted approach encompassing various methods to achieve its objectives. Initially, expert meetings were convened during the development phase to ensure the quality of course content and activities, focusing on investigating novel methodologies for teaching the MapReduce programming model and evaluating the impact on user performance. An instructional design methodology, coupled with expert opinion, was utilized to

categorize tasks into distinct problem categories, ensuring the effectiveness of the teaching approach. Additionally, data collection involved sourcing problems from prominent books in distributed systems, big data processing, and scientific articles to ensure a comprehensive representation of real-world MapReduce applications. Subsequent preprocessing and feature engineering techniques, including TF-IDF vectorization and the creation of additional feature combinations, enriched the dataset for classification model development. An artificial expansion strategy using paraphrasing techniques was implemented to address the challenge of a limited dataset. Feature selection and model development involved exploring various attribute combinations for classification models, with Naive Bayes and Logistic Regression models being developed and evaluated using cross-validation techniques. Summative evaluation methodologies, including statistical analysis such as an independent sample t-test and outlier analysis using box plots, were employed to assess the course's effectiveness and analyze the results, contributing to the successful execution of the research and derivation of meaningful conclusions. Moreover, a recommender system was developed based on the trained classification models to accurately forecast the assignment of categorical labels to novel problem instances, undergoing rigorous evaluation procedures, including expert opinion analysis, to assess its predictive efficacy.

The novelty of the research is a development of advisory system on the basis of a hybrid use of Naive Bayes and Logistic regression machine learning methods for training the Mapreduce programming model for transitioning single-machine problem solutions to distributed systems.

The research's significance is multifaceted and extends to both academic and practical realms. Academically, it contributes to advancing knowledge in distributed systems, big data processing, and educational methodologies. By investigating novel teaching methodologies for the MapReduce programming model and assessing their impact on student performance, the research fills a gap in understanding effective pedagogical approaches for complex computing paradigms. Additionally, developing and evaluating a recommender system for MapReduce problem classification offers insights into applying machine learning techniques in addressing real-world challenges in distributed computing.

Practically, the research has several implications. Firstly, the findings can inform the design of educational curricula and instructional materials for teaching MapReduce and related concepts, catering to the growing demand for skilled professionals in big data and distributed computing. The research also provides practical tools, such as the developed recommender system, which can aid practitioners in efficiently classifying and solving MapReduce problems in real-world applications. Furthermore, the methodologies and techniques employed in the research, such as data collection, preprocessing, feature selection, and model evaluation, offer a framework for addressing similar challenges in other domains of computer science and data analysis.

Overall, the significance of the research lies in its contributions to theoretical understanding and practical application, facilitating advancements in education,

research, and industry practices related to distributed systems and big data processing.

The results of research were published in following:

Research publications:

1. MapReduce Solutions Classification by Their Implementation // International Journal of Engineering Pedagogy (iJEP). – 2023. – Vol. 13, Issue 5. – P. 58-71.
2. Defining Semantically Close Words of Kazakh Language with Distributed System Apache Spark // Big Data and Cognitive Computing. – 2023. – Vol. 7.4. – P. 160.
3. Recommendation System for Human Resource Management by the Use of Apache Spark Cluster // 2023 17th International Conference on Electronics Computer and Computation (ICECCO), (IEEE, 2023).
4. MBTI personality classification using Apache Spark // 2021 16th International Conference on Electronics Computer and Computation (ICECCO), (IEEE, 2021).
5. Optimization of data segments and number of cores for defining popularity of Kazakh words using Apache Spark // Engineering Journal of Satbayev University. – 2021. – Vol. 143.3. – P. 39-42.
6. Computing feature vectors of students for face recognition using Apache Spark // 2019 15th International Conference on Electronics, Computer and Computation (ICECCO), (IEEE, 2019).

On April 16, 2024, the implementation certificate was received (Appendix A).

Structure and scope of the dissertation. The thesis consists of three chapters. It contains 19 figures and 8 tables, The list of used literature consists of 97 sources. The dissertation includes 68 pages.

1 BACKGROUND INFORMATION AND LITERATURE REVIEW

1.1 MapReduce paradigm

The MapReduce paradigm is a key concept in the field of distributed computing, particularly in the context of big data processing. It offers a powerful framework for processing and generating large datasets in parallel across a distributed cluster of computers. The essence of MapReduce lies in its ability to divide the processing into two main phases: the Map phase, where the input data is divided into smaller subproblems and processed in parallel, and the Reduce phase, where the results from the Map phase are combined to form the final output. One of the major advantages of the MapReduce paradigm is its scalability, as it can easily handle massive amounts of data by distributing the workload across multiple machines. This not only speeds up the processing but also provides fault tolerance, as the data is replicated across the cluster. Understanding the intricacies of the MapReduce paradigm is crucial for efficiently processing large-scale data in a distributed environment. This understanding forms the foundation for developing sophisticated algorithms and applications that harness the power of parallel processing for big data analytics [14].

The MapReduce paradigm was introduced by Google in 2004 as a means of efficiently processing large-scale data across a distributed network of computers. It was designed to address the challenges of processing massive datasets in a parallel and fault-tolerant manner. The origins of MapReduce can be traced back to functional programming concepts, particularly the map and reduce functions commonly found in functional programming languages. By harnessing the principles of functional programming and parallel processing, MapReduce revolutionized the way big data is processed and analyzed. Its simplicity and scalability have made it a fundamental framework for various big data technologies, including Apache Hadoop and Apache Spark. It's essential to explore the historical context and evolution of this paradigm, as well as its impact on the field of distributed computing and big data analytics. Understanding the origins of MapReduce will provide valuable insights into its fundamental principles and the underlying concepts that have shaped its widespread adoption in the industry [15].

The MapReduce paradigm consists of several key components that work together to process large-scale data across distributed clusters [16]. These components, shown in figure 1, include:

1. Mapper: The mapper is responsible for processing the input data and converting it into key-value pairs. It applies a specified map function to each input record and generates intermediate key-value pairs.
2. Reducer: The reducer takes the intermediate key-value pairs produced by the mapper and performs a reduction operation on the values associated with the same key. It combines these values to produce the final output.
3. Partitioner: The partitioner is responsible for distributing the intermediate key-value pairs across the reducers. It ensures that all intermediate key-value pairs with the same key are directed to the same reducer.

4. Shuffler: The shuffler is responsible for transferring the intermediate key-value pairs from the mappers to the reducers. It ensures that the key-value pairs are grouped and sorted before reaching the reducer.

The architecture of a MapReduce system typically consists of a master node, which coordinates the overall execution, and multiple worker nodes, which perform the actual processing. The master node is responsible for task scheduling, monitoring, and managing the overall workflow. The worker nodes execute the map and reduce tasks as directed by the master. Understanding these key components and the architecture of MapReduce is essential for effectively harnessing its power in processing large-scale data. With this knowledge, researchers and practitioners can design and implement efficient algorithms and applications for big data analytics [17].

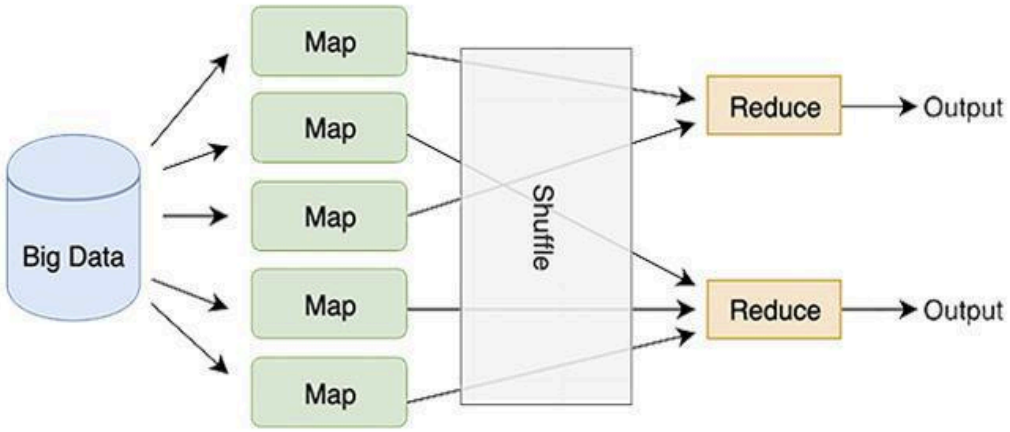

Figure 1 – MapReduce model

1.1.1 Optimizing MapReduce Performance

In order to fully leverage the capabilities of MapReduce in big data processing, it is imperative to focus on optimizing its performance. Several strategies can be employed to enhance the efficiency of MapReduce jobs [14, p. 843-852], such as:

1. Data Locality: By ensuring that computation is performed on the same node where the data resides, the overhead of transferring large volumes of data across the network can be minimized. This can significantly reduce processing time and enhance overall performance.

2. Combiner Functions: Introducing combiner functions can help in reducing the amount of data shuffled across the network during the MapReduce process. Combiners can aggregate the intermediate key-value pairs before they are transferred to the reducers, ultimately reducing the load on the network.

3. Partitioning Strategies: Implementing effective partitioning strategies can ensure a balanced distribution of data across reducers, preventing skewness and improving overall processing efficiency.

4. Optimized Input Formats: Choosing the appropriate input formats, such as SequenceFile or Columnar formats, can optimize the reading and processing of input data, thereby enhancing the overall performance of MapReduce jobs.

5. Task Parallelism: Dividing the processing tasks into smaller, parallelizable units can improve job execution time and maximize resource utilization across the cluster.

By understanding and applying these optimization strategies, researchers and practitioners can harness the full potential of MapReduce in big data processing, achieving improved performance and scalability for their data-intensive applications.

1.1.2 Algorithm Design and Optimization in MapReduce

Designing and optimizing algorithms for MapReduce involves considering various factors to ensure efficient processing of large-scale data [18]. One key aspect is the careful design of map and reduce functions to distribute and process the input data effectively. Additionally, optimizing the performance of these functions and the overall MapReduce jobs is essential to achieve the desired scalability and efficiency.

Efficient Data Partitioning

Effective data partitioning is crucial for the balanced distribution of data across the reducers. By employing appropriate partitioning techniques based on the characteristics of the input data, researchers can minimize data skewness and improve the overall performance of the MapReduce jobs. Common partitioning methods include range-based partitioning, hash partitioning, and customized partitioning based on specific data attributes.

Payload Optimization

Optimizing the size of intermediate data transferred between the mappers and reducers is vital for reducing network overhead and enhancing overall performance. This involves careful consideration of the key and value sizes, as well as utilizing serialization and compression techniques to minimize data transfer across the network.

Handling Skewed Data

Dealing with skewed data distribution poses a significant challenge in MapReduce processing. Efficient handling of skewed keys and the associated data can be addressed through techniques such as data skew detection, adaptive partitioning, and specialized handling within the reduce phase to prevent processing bottlenecks.

Iterative Algorithms

Supporting iterative algorithms in MapReduce involves implementing optimizations to reduce the overhead of repetitive job initialization and data transfer. Techniques like in-memory data caching, incremental processing, and specialized job chaining can significantly improve the performance of iterative algorithms within the MapReduce framework.

Resource Optimization

Optimizing resource allocation and utilization within MapReduce clusters is crucial for achieving efficient job execution. This includes considerations for data locality, task scheduling, memory management, and scaling the cluster resources

based on the processing demands. Efficient resource management contributes to improved performance and scalability of MapReduce jobs.

Integration with Advanced Frameworks

Exploring the integration of MapReduce with advanced frameworks and technologies, such as machine learning libraries, graph processing frameworks, and real-time data processing platforms, opens up opportunities for developing sophisticated algorithms and applications that leverage the strengths of both MapReduce and specialized processing paradigms. By focusing on algorithm design and optimization within the MapReduce paradigm, researchers and practitioners can develop robust, high-performance solutions for processing and analyzing large-scale data in distributed environments. The careful consideration of these factors contributes to the effective utilization of MapReduce for diverse application domains and complex data processing requirements.

1.1.3 Real-World Applications of MapReduce

MapReduce has been widely adopted in various real-world applications to process and analyze large-scale data efficiently. Let's explore some notable case studies that demonstrate the diverse use cases and benefits of MapReduce in different domains.

E-commerce Data Analysis

In the e-commerce industry, MapReduce has been instrumental in analyzing vast amounts of transactional data, customer behavior patterns, and product performance metrics. By employing MapReduce, e-commerce companies can derive valuable insights for personalized marketing, inventory optimization, and customer segmentation strategies. The parallel processing capabilities of MapReduce enable rapid analysis of massive datasets, contributing to data-driven decision-making and business growth [19].

Healthcare Informatics

MapReduce has revolutionized healthcare informatics by enabling the analysis of electronic health records, medical imaging data, and clinical research findings. Healthcare organizations leverage MapReduce to identify disease trends, optimize treatment protocols, and perform predictive analytics for patient outcomes. The scalability and fault tolerance of MapReduce facilitate the processing of diverse healthcare datasets, ultimately supporting improved healthcare delivery and research advancements [20].

Financial Risk Management

In the financial sector, MapReduce plays a vital role in managing and analyzing risk-related data, including market trends, credit risk assessments, and fraud detection. By leveraging MapReduce, financial institutions can conduct in-depth analysis of large-scale financial transactions, detect anomalies, and enhance regulatory compliance. The distributed computing capabilities of MapReduce enable efficient processing of complex risk models and the identification of emerging risk factors in dynamic financial markets [21].

Social Media Analytics

MapReduce is extensively utilized in social media analytics to process and analyze user-generated content, sentiment analysis, and network interactions. Social media platforms leverage MapReduce to gain insights into user behavior, trending topics, and content engagement metrics. The scalability and parallel processing capabilities of MapReduce empower social media companies to derive actionable insights for content curation, targeted advertising, and user experience enhancement [22].

Environmental Monitoring and Analysis

MapReduce technology has been applied in environmental monitoring and analysis to process geospatial data, climate models, and ecological observations. Organizations involved in environmental research and conservation leverage MapReduce to analyze large-scale environmental datasets, predict natural disasters, and monitor biodiversity trends. The distributed computing framework of MapReduce supports the integration of diverse environmental data sources and the development of proactive strategies for environmental sustainability. These case studies highlight the versatility and impact of MapReduce in addressing complex data processing and analysis requirements across different industries. The scalability, fault tolerance, and parallel processing capabilities of MapReduce continue to empower organizations to extract actionable insights and drive innovation in their respective domains [23].

1.1.4 Performance Evaluation of MapReduce Frameworks

Performance evaluation of MapReduce frameworks is essential for understanding their capabilities and limitations in real-world scenarios. Various factors contribute to the effectiveness of a MapReduce framework, including scalability, fault tolerance, resource utilization, and overall processing efficiency. Conducting comprehensive performance evaluations helps organizations and researchers make informed decisions when selecting and optimizing MapReduce frameworks for their specific use cases [24].

Scalability Analysis

Scalability is a fundamental aspect of MapReduce frameworks, particularly in handling growing datasets and increasing computational demands. Performance evaluations assess the ability of MapReduce frameworks to scale seamlessly across clusters, accommodate higher workloads, and efficiently distribute processing tasks. By benchmarking scalability, organizations can determine the framework's suitability for accommodating future data growth and evolving processing requirements.

Fault Tolerance and Reliability

The fault tolerance mechanism of MapReduce frameworks is critical for ensuring uninterrupted processing in the presence of node failures and data inconsistencies. Performance evaluations include stress testing and fault injection to assess the reliability of frameworks in diverse failure scenarios. Evaluating fault tolerance capabilities provides insights into the framework's ability to maintain data integrity, recover from failures, and sustain high availability, which are essential for mission-critical applications.

Resource Utilization and Efficiency

Efficient resource management is crucial for optimizing the utilization of computing resources within MapReduce clusters. Performance evaluations examine resource allocation, task scheduling algorithms, and memory utilization to identify opportunities for improving overall efficiency. By quantifying resource utilization metrics and analyzing performance overhead, organizations can fine-tune their MapReduce frameworks to minimize wastage and maximize computational throughput.

Processing Throughput and Latency

Analyzing the processing throughput and latency of MapReduce frameworks involves benchmarking the execution speed and response times for various job configurations and input data sizes. Performance evaluations quantify the efficiency of data processing, intermediate data transfer, and task completion, providing insights into the framework's ability to deliver timely results and meet latency requirements for interactive applications.

Comparative Analysis of Frameworks

Conducting comparative performance evaluations of different MapReduce frameworks, such as Apache Hadoop, Apache Spark, and others, enables organizations to understand the strengths and limitations of each framework in specific use cases. Comparative analyses consider factors like data processing speed, resource utilization patterns, fault tolerance mechanisms, and integration capabilities with other technologies, aiding in informed decision-making for framework selection and optimization.

By systematically evaluating the performance of MapReduce frameworks, organizations can assess their suitability for diverse application domains, identify optimization opportunities, and make informed decisions to meet their specific processing and analysis requirements. Performance evaluations contribute to the continuous improvement of MapReduce frameworks and their effective utilization in real-world scenarios.

1.1.5 Challenges and Limitations of MapReduce

While MapReduce offers significant benefits in processing and analyzing large-scale data, it also presents several challenges and limitations that organizations need to consider [25].

Complex Programming Model

One of the challenges associated with MapReduce is the complexity of its programming model. Writing MapReduce programs requires a deep understanding of distributed computing concepts, data partitioning, and parallel processing, which can pose a learning curve for developers and data analysts. As a result, organizations may face challenges in recruiting and training personnel with the necessary expertise to effectively utilize MapReduce for their data processing needs.

Latency in Iterative Processing

MapReduce was originally designed for batch processing of data, which introduces latency in iterative algorithms and real-time data analytics. Iterative tasks, such as machine learning algorithms and graph processing, may experience performance overhead due to the disk-based intermediate data storage and the lack of

built-in support for iterative computations. This limitation can affect the responsiveness of real-time applications and time-sensitive data analytics processes.

Overhead of Disk I/O Operations

MapReduce frameworks rely on disk-based Input/Output (I/O) operations for storing intermediate data during the map and reduce phases. The overhead associated with disk I/O can impact the overall processing efficiency, especially for tasks that involve frequent data shuffling and intermediate results exchange. Minimizing disk I/O overhead and optimizing data transfer between processing stages are ongoing challenges in enhancing the performance of MapReduce frameworks.

Limited Support for Complex Data Flows

While MapReduce excels in processing structured and semi-structured data through map and reduce functions, it may encounter limitations in handling complex data flows and nested data structures. Tasks requiring multi-stage data transformations, hierarchical data processing, and inter-job data dependencies may necessitate workarounds or custom implementations within MapReduce, leading to increased development complexity and maintenance overhead.

Alternative Frameworks and Paradigms

The evolving landscape of big data processing has seen the emergence of alternative frameworks and paradigms, such as Apache Spark, which offer advantages in memory-based processing, iterative computations, and stream processing. Organizations evaluating MapReduce must consider the growing diversity of data processing technologies and choose the most suitable framework for their specific use cases, taking into account factors like performance, programming ease, and integration with existing infrastructure.

As organizations continue to harness the power of MapReduce for diverse data processing and analysis tasks, they must navigate the challenges and limitations inherent in the framework. Addressing these challenges requires ongoing research and development efforts to enhance the performance, flexibility, and usability of MapReduce in the context of evolving data processing requirements and technological advancements. Despite its limitations, MapReduce remains a foundational and valuable tool for organizations seeking to derive insights from large-scale datasets and drive innovation across various industry domains.

1.2 MapReduce problems

MapReduce, a programming model introduced by Google, has revolutionized how large-scale data processing tasks are handled. As the volume of data generated escalates exponentially, the need for efficient and scalable data processing techniques becomes increasingly paramount. Two seminal papers, namely "The survey on MapReduce" by Vijayalakshmi, Akila, and Nagadivya [26], and "Parallel data processing with MapReduce: a survey" by Lee et al. [27], provide comprehensive overviews of the MapReduce paradigm, its applications, challenges, and future directions.

In their paper, Vijayalakshmi et al. delve into the fundamental concepts of MapReduce, elucidating its architecture, components, and workflow. They highlight the key stages of a MapReduce job, including mapping, shuffling, and reducing, and

explicate how these stages facilitate the parallel processing of data across distributed clusters of computing nodes. Additionally, the authors examine various implementations of MapReduce frameworks, such as Apache Hadoop and Apache Spark, shedding light on their features, capabilities, and comparative advantages.

Complementing the work of Vijayalakshmi et al., Lee et al. provide a comprehensive survey focusing on the parallel data processing aspects of MapReduce. They elucidate how MapReduce enables the seamless parallelization of computations, thereby accelerating the processing of massive datasets. Moreover, the authors investigate the evolution of MapReduce frameworks and discuss their adaptability to diverse application domains, ranging from web indexing to machine learning and scientific computing. Through an in-depth analysis, Lee et al. underscore the significance of MapReduce in facilitating scalable and fault-tolerant data processing, particularly in the context of cloud computing environments.

Both papers underscore the transformative impact of MapReduce on the field of big data analytics. They recognize its pivotal role in enabling the efficient processing of large-scale datasets, thereby empowering organizations to extract valuable insights and make data-driven decisions. Moreover, the surveys elucidate the challenges inherent in MapReduce, including issues related to fault tolerance, scalability, and resource management. Despite these challenges, both papers remain optimistic about the future prospects of MapReduce, envisaging its continued evolution and adoption in emerging application domains.

In conclusion, the surveys conducted by Vijayalakshmi et al. and Lee et al. offer comprehensive insights into the MapReduce paradigm, its applications, and its challenges. By elucidating the fundamental concepts and practical implementations of MapReduce, these papers serve as invaluable resources for researchers, practitioners, and organizations seeking to harness the power of big data analytics for competitive advantage.

1.2.1 Sentiment analysis

In the realm of sentiment analysis, the utilization of big data frameworks like Hadoop MapReduce has garnered significant attention owing to its ability to process vast amounts of data efficiently. Several studies have explored the application of MapReduce in sentiment analysis, aiming to extract valuable insights from social media and textual data. This literature review examines four key papers in this domain, providing insights into their methodologies, contributions, and findings.

Madani, Erritali, and Bengourram [28] proposed a novel approach for sentiment analysis leveraging semantic similarity and Hadoop MapReduce. Their work focuses on enhancing sentiment classification accuracy by integrating semantic similarity measures into the analysis process. By utilizing MapReduce, the authors demonstrate the scalability and efficiency of their approach in handling large-scale textual data, thereby facilitating more accurate sentiment analysis.

Ha, Back, and Ahn [29] explored the application of MapReduce functions for sentiment analysis of social big data. Their study emphasizes the importance of analyzing sentiment information from social media platforms to effectively understand user opinions and sentiments. Leveraging MapReduce functions, the

authors developed a framework capable of processing vast amounts of social data efficiently, enabling organizations to extract valuable insights for decision-making.

Nasir, Zafar, and Alamgir [30] conducted a study on social media sentiment analysis using MapReduce. Their research delves into the challenges of sentiment analysis in the context of social media data, highlighting the need for scalable and efficient processing techniques. By utilizing MapReduce, the authors proposed a framework capable of handling the complexities of social media data, thereby facilitating sentiment analysis at scale.

Nodarakis et al. [31] introduced MR-SAT, a MapReduce algorithm explicitly designed for big data sentiment analysis on Twitter. Their work addresses the challenges associated with sentiment analysis on microblogging platforms, such as data sparsity and noise. By leveraging MapReduce, the authors developed an efficient algorithm capable of processing Twitter data in parallel, enabling real-time sentiment analysis at scale.

In summary, these studies underscore the significance of MapReduce in sentiment analysis, particularly in analyzing social media data. By harnessing the scalability and efficiency of MapReduce frameworks, researchers have developed novel approaches and algorithms for sentiment analysis, thereby advancing the understanding of user opinions and sentiments in large-scale textual data.

1.2.2 K-means clustering

K-means clustering is a widely used unsupervised machine learning algorithm for partitioning data into clusters based on similarity. With the advent of big data, there has been a growing interest in developing scalable and efficient implementations of K-means clustering using distributed computing frameworks such as MapReduce. This literature review provides an overview of five key papers that explore various aspects of MapReduce-based K-means clustering algorithms.

Cui et al. [32] introduced an optimized approach for K-means clustering using MapReduce, aiming to improve the efficiency and scalability of the algorithm for big data applications. Their work focuses on optimizing the data partitioning and centroid update steps within the MapReduce framework, resulting in reduced computation time and improved clustering performance.

Anchalia, Koundinya, and Srinath [33] proposed a MapReduce design for the K-means clustering algorithm, aiming to leverage the parallel processing capabilities of distributed computing environments. Their approach focuses on distributing the computation of distance metrics and centroid updates across multiple computing nodes, thereby enabling the efficient clustering of large-scale datasets.

Gopalani and Arora [34] conducted a comparative study between Apache Spark and traditional MapReduce frameworks for implementing K-means clustering algorithms. Their research emphasizes the importance of performance analysis in selecting the appropriate distributed computing platform for big data analytics tasks. Through empirical evaluation, the authors demonstrate the superior performance of Apache Spark over MapReduce for K-means clustering.

Sardar and Ansari [35] analyzed distributed document clustering using a MapReduce-based K-means algorithm. Their study explores the application of

K-means clustering in text-mining tasks, focusing on the challenges and opportunities of distributed document clustering. Through experimentation, the authors evaluate the scalability and effectiveness of the MapReduce-based approach for document clustering.

Mao et al. [36] proposed a novel MapReduce-based K-means clustering algorithm designed to handle large-scale datasets efficiently. Their approach leverages the MapReduce framework to parallelize the computation of distance metrics and centroid updates, thereby enabling scalable and distributed K-means clustering. Through empirical evaluation, the authors demonstrate the effectiveness of their algorithm in clustering large datasets.

In summary, these studies highlight the significance of MapReduce in enabling scalable and efficient implementations of the K-means clustering algorithm for big data analytics. By leveraging distributed computing frameworks, researchers have been able to overcome the challenges associated with processing large-scale datasets, thereby facilitating the application of K-means clustering in various domains such as text mining, document clustering, and data analysis.

1.2.3 Decision tree

Decision tree algorithms are widely used in machine learning and data mining for classification and regression tasks due to their simplicity, interpretability, and effectiveness. With the advent of big data, there has been a growing interest in developing parallel and scalable implementations of decision tree algorithms using distributed computing frameworks such as MapReduce. This literature review provides an overview of five key papers that explore various aspects of MapReduce-based decision tree algorithms.

Dai and Ji [37] proposed a MapReduce implementation of the C4.5 decision tree algorithm, aiming to leverage the parallel processing capabilities of MapReduce for scalable and efficient decision tree induction. Their work focuses on partitioning the dataset and distributing the computation of decision tree nodes across multiple computing nodes, thereby reducing the overall computation time and enabling the processing of large-scale datasets.

Koli and Shinde [38] presented a parallel decision tree algorithm with a MapReduce model tailored for big data analytics. Their research emphasizes the importance of parallelization in handling large-scale datasets effectively. By leveraging MapReduce, the authors developed a scalable and efficient decision tree algorithm capable of processing massive amounts of data in parallel, thereby facilitating big data analytics tasks.

Es-sabery and Hair [39] proposed a MapReduce-based C4.5 decision tree algorithm augmented with a fuzzy rule-based system. Their approach integrates fuzzy logic into the decision tree induction process, enabling the algorithm to handle uncertainty and imprecision in the data. Through empirical evaluation, the authors demonstrate the effectiveness of their approach in improving the accuracy and interpretability of decision tree models for classification tasks.

Es-Sabery et al. [40] introduced a MapReduce opinion mining framework for classifying COVID-19-related tweets using an enhanced ID3 decision tree classifier.

Their research addresses the challenge of sentiment analysis and opinion mining in social media data, particularly during the COVID-19 pandemic. By leveraging MapReduce, the authors developed a scalable and efficient classification system capable of processing large volumes of tweets in real-time, thereby facilitating the analysis of public sentiment towards the pandemic.

Mu et al. [41] proposed a parallel fuzzy rule-based decision tree algorithm in the framework of MapReduce. Their research extends traditional decision tree algorithms by incorporating fuzzy logic and parallel processing techniques. Through experimental evaluation, the authors demonstrate the effectiveness of their approach in handling complex and imprecise data, thereby enhancing the accuracy and robustness of decision tree models for classification tasks.

In summary, these studies highlight the significance of MapReduce in enabling parallel and scalable implementations of decision tree algorithms for big data analytics. By leveraging distributed computing frameworks, researchers have been able to develop novel approaches for decision tree induction, thereby facilitating the analysis of large-scale datasets in various application domains.

1.2.4 Apriori algorithm

Apriori algorithm, a classic method in association rule mining, has been widely applied in various domains for discovering frequent patterns and associations within large datasets. With the advent of big data, the need for efficient and scalable implementations of the Apriori algorithm has become increasingly important. This literature review examines six key papers that explore different approaches for implementing the Apriori algorithm using the MapReduce framework.

Sornalakshmi et al. [42] presented an efficient Apriori algorithm tailored for frequent pattern mining in healthcare data using MapReduce. Their work focuses on optimizing the Apriori algorithm's performance within the MapReduce paradigm to handle large-scale healthcare datasets efficiently. Through empirical evaluation, the authors demonstrate the effectiveness of their approach in improving the efficiency of frequent pattern mining tasks in healthcare data.

Yange et al. [43] proposed a multi-nodal implementation of the Apriori algorithm for big data analytics using the MapReduce framework. Their research aims to leverage the parallel processing capabilities of MapReduce to enhance the scalability and efficiency of the Apriori algorithm for large-scale datasets. Through experimentation, the authors demonstrate the effectiveness of their approach in accelerating frequent pattern mining tasks in big data analytics.

Verma, Malhotra, and Singh [44] explored the application of MapReduce-Apriori framework for big data analytics in the retail industry. Their study focuses on leveraging the Apriori algorithm within the MapReduce framework to analyze large-scale retail datasets and extract valuable insights for decision-making purposes. Through empirical analysis, the authors highlight the benefits of using MapReduce-Apriori framework for retail analytics tasks.

Sharma and Tripathi [45] proposed a hybrid version of the Apriori algorithm using MapReduce, aiming to enhance its performance and scalability for big data analytics. Their research integrates traditional Apriori algorithm with optimization

techniques and parallel processing using MapReduce, thereby improving the efficiency of frequent pattern mining tasks. Through experimental evaluation, the authors demonstrate the superiority of their hybrid approach over traditional Apriori algorithm.

Wang and Gao [46] conducted research on the parallelization of the Apriori algorithm in association rule mining. Their study focuses on optimizing the Apriori algorithm's performance by parallelizing the computation of frequent itemsets using MapReduce. Through empirical evaluation, the authors demonstrate the effectiveness of their parallelization approach in accelerating association rule mining tasks on large datasets.

Sundarakumar et al. [47] proposed an enhanced Apriori algorithm for improving data processing speed on large datasets in a Hadoop multinode cluster. Their research focuses on optimizing the Apriori algorithm's performance within a distributed computing environment using Hadoop and MapReduce. Through experimentation, the authors demonstrate the effectiveness of their enhanced Apriori algorithm in achieving faster data processing speeds on large datasets.

In summary, these studies highlight the significance of MapReduce in enabling scalable and efficient implementations of the Apriori algorithm for big data analytics. Researchers have developed novel approaches and optimizations for accelerating frequent pattern-mining tasks on large-scale datasets across various domains by leveraging distributed computing frameworks.

1.2.5 Additional works

In recent years, the utilization of Apache Spark for various computational tasks has garnered significant attention within the academic community. This review synthesizes findings from several studies employing Apache Spark for diverse applications, ranging from face recognition to semantic analysis of language.

Kariboz et al. [48] presented a novel approach for computing feature vectors of students to enhance face recognition systems. Leveraging Apache Spark, their method demonstrated promising results in accurately identifying individuals in large datasets, showcasing the scalability and efficiency of Spark in handling complex computational tasks.

Meraliyev et al. [49] explored the optimization of data segmentation and core allocation in Apache Spark for determining the popularity of Kazakh words. By efficiently distributing computation across cores, their study achieved improved performance in analyzing large volumes of linguistic data, highlighting the potential of Spark for natural language processing tasks.

Orynbekova et al. [50] contributed to the field of personality classification by proposing a method for MBTI personality classification utilizing Apache Spark. Their study underscored the efficacy of Spark in processing and analyzing large-scale datasets for psychological profiling, opening avenues for personalized recommendation systems and targeted interventions.

Serek et al. [51] addressed the challenges in human resource management through the development of a recommendation system leveraging an Apache Spark

cluster. By harnessing Spark's distributed computing capabilities, their system offered scalable and efficient solutions for talent acquisition and workforce optimization.

Ayazbayev et al. [52] extended the applicability of Apache Spark to semantic analysis tasks, particularly focusing on the identification of semantically similar words in the Kazakh language. Their study demonstrated the utility of Spark for distributed semantic analysis, enabling efficient extraction of meaningful insights from large text corpora.

Collectively, these studies underscore the versatility and effectiveness of Apache Spark across diverse domains, ranging from image analysis and natural language processing to recommendation systems and talent management. By leveraging Spark's distributed computing framework, researchers continue to push the boundaries of computational capabilities, offering innovative solutions to complex real-world challenges.

1.3 Recommender systems

Recommendation systems have become an integral part of daily lives, being influenced by the way new products, movies, music, and even news articles are discovered. These systems are designed to analyze user preferences and provide personalized recommendations, ultimately aiming to enhance user experience and increase user engagement. Understanding how recommendation systems work and the various approaches used in their implementation is deemed crucial for both businesses and consumers [53].

1.3.1 Collaborative Recommender system

One of the most popular approaches to recommendation systems is the collaborative filtering method. This system analyzes user behavior and preferences to predict a user's interests automatically. Collaborative filtering can be implemented in two ways: user-based or item-based. User-based collaborative filtering recommends items based on similarities between users, while item-based collaborative filtering recommends items based on similarities between items themselves.

Collaborative recommendation systems are particularly effective in scenarios where there is a large amount of user interaction data available, such as in e-commerce platforms or social media websites. By leveraging the collective wisdom of users, collaborative filtering can provide accurate and personalized recommendations, leading to improved user satisfaction and increased engagement.

In addition to collaborative filtering, there are other techniques such as content-based filtering and hybrid methods that combine multiple approaches to provide even more accurate recommendations. Understanding the strengths and limitations of each method is essential for businesses looking to implement recommendation systems that effectively meet the needs and expectations of their users [54].

1.3.2 Content-Based Recommender System

In addition to collaborative filtering, another popular approach to recommendation systems is the content-based filtering method. Content-based

recommender systems analyze the attributes and features of items to make recommendations based on the similarity between these attributes and a user's preferences.

For example, in a music streaming platform, a content-based recommender system could recommend new songs to a user based on the genre, artist, or similar attributes of the songs they have previously listened to and liked. Similarly, in e-commerce platforms, content-based recommendation systems can recommend products based on their attributes such as category, brand, or characteristics.

Content-based systems are particularly effective in scenarios where there is a significant amount of item-related information available, allowing for accurate predictions based on the content itself. By understanding the content of items and how it relates to user preferences, content-based recommender systems can provide personalized recommendations, ultimately enhancing user satisfaction and engagement.

Implementing a content-based recommender system involves understanding and processing the attributes and features of items to create a profile for each user that captures their preferences. This approach is especially useful when user interaction data may be limited or when items have well-defined and structured attributes that can be used for recommendation purposes.

In practice, many recommendation systems leverage a combination of collaborative filtering, content-based filtering, and hybrid methods to provide diverse and accurate recommendations that meet the diverse preferences of users. This multi-faceted approach allows businesses to cater to the varying needs and interests of their user base while enhancing the overall user experience [54, p. 361-366].

1.3.3 Demographic-Based Recommender System

Another approach to recommendation systems is the demographic-based method, which considers demographic information such as age, gender, location, and other relevant attributes of users to make personalized recommendations. By understanding the demographics of users, this approach can provide recommendations that are more tailored to individual preferences based on demographic factors.

Demographic-based recommender systems are particularly effective in scenarios where demographic information is available and can be used to enhance the personalization of recommendations. For example, in a movie streaming platform, this approach could recommend movies based on the user's age group, gender, or location. In e-commerce platforms, demographic-based systems can recommend products based on the user's age, gender, or location-specific preferences.

By integrating demographic information into the recommendation process, businesses can create more targeted and relevant recommendations for their users, ultimately improving user satisfaction and engagement. Understanding the demographic characteristics of users and how they influence their preferences is essential for the successful implementation of a demographic-based recommender system.

In practice, combining demographic-based filtering with other recommendation approaches, such as collaborative filtering or content-based filtering, can further enhance the accuracy and relevance of recommendations, providing a comprehensive approach to catering to the diverse needs of users across different demographic segments [55].

1.3.4 Utility-Based Recommender System

Another crucial approach to recommendation systems is the utility-based method. Utility-based recommender systems consider explicit ratings or implicit feedback from users to understand their preferences and make personalized recommendations. This approach focuses on maximizing a user's utility or satisfaction by recommending items that are most likely to be highly rated or preferred by the user.

In utility-based recommendation systems, the system evaluates the utility of each item for a user based on the user's historical interactions, ratings, or feedback. This evaluation helps in predicting the likelihood of a user's satisfaction with a particular item. For instance, in a movie streaming platform, a utility-based recommender system can analyze user ratings and preferences to recommend movies that align with the user's tastes and preferences.

By leveraging utility-based approaches, businesses can enhance user satisfaction by providing recommendations that are more likely to be well-received and enjoyed by the users. Understanding the explicit and implicit feedback from users and translating it into personalized recommendations is pivotal for the effective implementation of utility-based recommender systems.

In practice, integrating utility-based filtering with other recommendation approaches such as collaborative filtering, content-based filtering, or demographic-based filtering can further refine the recommendation process, leading to more accurate and customized recommendations that meet the diverse preferences of users while maximizing user satisfaction and engagement [56].

1.3.5 Knowledge-Based Recommender System

Another important approach to recommendation systems is the knowledge-based method. Knowledge-based recommender systems use explicit knowledge about items and user preferences to make recommendations. This approach is particularly useful when there is a limited amount of user interaction data available, as it can rely on structured information about items and user preferences.

Knowledge-based recommender systems often use rule-based or knowledge representation techniques to infer recommendations based on explicit knowledge about items and user preferences. For example, in a travel booking platform, a knowledge-based recommender system could recommend destinations based on specific user preferences such as travel dates, budget, preferred activities, and accommodation preferences.

By leveraging structured knowledge about items and user preferences, knowledge-based recommender systems can provide personalized and contextually

relevant recommendations, ultimately leading to improved user satisfaction and engagement.

Implementing a knowledge-based recommender system involves understanding the explicit knowledge about items and users, and representing this knowledge in a way that can be used to infer recommendations. This approach is particularly useful in domains where items have well-defined attributes and where user preferences can be explicitly captured.

In practice, combining knowledge-based filtering with other recommendation approaches such as collaborative filtering, content-based filtering, demographic-based filtering, or utility-based filtering can further enhance the accuracy and relevance of recommendations, providing a holistic approach to catering to the diverse needs of users while maximizing user satisfaction and engagement [57].

1.3.6 Hybrid Recommender System

One of the most advanced and effective approaches to recommendation systems is the hybrid method, which combines multiple recommendation techniques to provide diverse and accurate recommendations that cater to the varying needs and interests of users. A hybrid recommender system leverages the strengths of different recommendation approaches such as collaborative filtering, content-based filtering, demographic-based filtering, utility-based filtering, and knowledge-based filtering to create a comprehensive and well-rounded recommendation system.

By integrating multiple recommendation techniques, a hybrid recommender system can overcome the limitations of individual approaches and improve the overall quality of recommendations. For example, by combining collaborative filtering with content-based filtering, the system can take advantage of both user interactions and item attributes to provide more personalized recommendations. Similarly, by integrating demographic-based filtering and utility-based filtering, the system can factor in both demographic information and user preferences to enhance the personalization of recommendations.

In practice, businesses can benefit greatly from implementing a hybrid recommender system as it enables them to provide more accurate, diverse, and personalized recommendations to their users. This multi-faceted approach ensures that recommendations are tailored to individual preferences while considering demographic factors, user interactions, explicit ratings, and structured knowledge about items.

Implementing a hybrid recommender system involves combining and integrating different recommendation techniques in a way that maximizes the strengths of each approach and mitigates their individual limitations. By doing so, businesses can deliver a superior recommendation experience to their users, leading to increased user satisfaction and engagement.

In conclusion, the hybrid recommender system represents a cutting-edge approach to recommendation systems that harnesses the power of multiple techniques to deliver more accurate, diverse, and personalized recommendations. Businesses that

adopt this approach can effectively cater to the diverse needs of their user base while maximizing user satisfaction and engagement [58].

1.3.7 The Role of Knowledge in Recommendation Systems

Knowledge plays a crucial role in the effectiveness of recommendation systems. Integrating domain-specific knowledge allows for a more comprehensive understanding of user preferences and interests, leading to more accurate and personalized recommendations.

One approach to incorporating knowledge into recommendation systems is through the use of ontologies and semantic modeling. These techniques enable recommendation systems to understand the relationships between different items and users, facilitating better recommendation generation [59].

Furthermore, the integration of text-based approaches, such as natural language processing and sentiment analysis, provides valuable insights into user behavior and preferences. By analyzing textual data, recommendation systems can extract semantic information and sentiment from user reviews, social media posts, and other sources, enriching the understanding of user preferences [60].

1.3.8 Text-Based Approaches in Recommendation Technologies

Natural Language Processing (NLP) plays a significant role in text-based approaches within recommendation systems. By leveraging NLP techniques, recommendation systems can extract valuable insights from textual data, such as user reviews, product descriptions, and social media posts. These insights, in turn, enable the system to understand and interpret user preferences and sentiments more effectively.

NLP facilitates the processing and analysis of unstructured textual data, allowing recommendation systems to identify key phrases, topics, and sentiments expressed by users. This understanding of the textual content contributes to the generation of more contextually relevant recommendations, ultimately enhancing user satisfaction and engagement [61].

In the realm of recommendation systems, natural language processing has been extensively utilized to enhance user experience and provide more accurate recommendations. One notable example is the deployment of sentiment analysis in social media platforms to understand user preferences and opinions. By analyzing the sentiment behind user posts and comments, recommendation systems can gauge the user's receptiveness to certain products or services, leading to tailored recommendations that resonate with the user's sentiment.

Moreover, NLP has been instrumental in the development of chat-based recommendation systems. These systems leverage NLP techniques to understand and interpret user queries and conversations, delivering personalized product recommendations in a conversational manner. This approach not only improves user experience but also fosters increased user engagement and satisfaction.

In addition, e-commerce platforms have harnessed NLP to parse through product descriptions and customer reviews, extracting valuable information about features, benefits, and user sentiments. This enables the recommendation systems to

offer more relevant and personalized suggestions to users based on their expressed preferences and interests [62].

1.3.9 Future Directions in NLP-Driven Recommendation Systems

As NLP continues to advance, the future of recommendation systems holds exciting possibilities. With the growing sophistication of NLP models, recommendation systems will be capable of a more nuanced understanding of user preferences, including subtle contextual cues and language nuances. This will result in more precise and tailored recommendations, further enhancing user satisfaction and retention.

Furthermore, the integration of multimodal NLP, which combines textual analysis with visual and auditory inputs, will open up new dimensions for recommendation systems. By analyzing not just text but also images, videos, and audio content, recommendation systems can gain a more holistic view of user preferences, leading to even more personalized and engaging recommendations.

In conclusion, the integration of NLP in recommendation systems has revolutionized the way users interact with products and services, paving the way for a more personalized and enriching user experience. As businesses continue to adopt and refine NLP-driven recommendation systems, the potential for innovation and improvement in user engagement remains limitless [63].

1.3.10 Synergy of Knowledge and Text-Based Methods

The synergy of knowledge-based and text-based methods in recommendation systems presents a powerful opportunity to elevate user experience and algorithm accuracy. By integrating domain-specific knowledge with text-based insights, recommendation systems can achieve a holistic understanding of user preferences and provide truly tailored recommendations.

One way to achieve this synergy is through the incorporation of semantic knowledge graphs, which encapsulate domain-specific knowledge and relationships between items. These knowledge graphs can be enriched with textual data extracted through NLP, creating a comprehensive knowledge base that fuels recommendation algorithms with both structured knowledge and unstructured insights [64].

1.3.11 Problem-Solving with Advanced Recommendation Systems

The evolution of recommendation systems has paved the way for more sophisticated and personalized user experiences. Advanced recommendation systems leverage a combination of advanced technologies, including machine learning, deep learning, and collaborative filtering, to generate highly tailored recommendations that cater to individual preferences and behaviors.

Machine learning algorithms form the backbone of modern recommendation systems, enabling the automatic learning of user preferences and behavior patterns. These algorithms utilize historical user data, such as past purchases, browsing history, and interaction patterns, to predict and recommend items that are most likely to resonate with each user [65].

By continuously learning and adapting to user feedback and interactions, machine learning-powered recommendation systems enhance their recommendation

accuracy and relevance over time, ultimately leading to improved user satisfaction and engagement.

Deep learning techniques, including neural networks and embeddings, have significantly elevated the capabilities of recommendation systems. Through deep learning, recommendation systems can capture complex patterns and dependencies within user data, enabling more nuanced and precise recommendations.

The use of deep learning models allows recommendation systems to uncover intricate correlations between user preferences, contextual factors, and item characteristics, leading to the generation of highly personalized recommendations that align with each user's unique tastes and interests [66].

1.4 Recommender systems works

Recommendation systems play a crucial role in various domains, including e-learning, e-commerce, and entertainment, by assisting users in discovering relevant items or resources based on their preferences and behaviors. This literature review provides insights into five key papers that survey different aspects of recommendation systems, including machine learning techniques, causal inference, data sparsity resolution, and deep learning methods.

Khanal et al. [67] conducted a systematic review of machine learning-based recommendation systems for e-learning. Their research focuses on analyzing various machine-learning algorithms and techniques employed in e-learning recommendation systems. Through a comprehensive review of existing literature, the authors identify key challenges and opportunities in the design and implementation of recommendation systems for e-learning platforms.

Ko et al. [68] presented a survey of recommendation systems, covering recommendation models, techniques, and application fields. Their survey provides an overview of different recommendation approaches, including collaborative filtering, content-based filtering, and hybrid methods. By analyzing recent advancements and emerging trends in recommendation systems, the authors offer insights into state-of-the-art techniques and their applications across diverse domains.

Gao et al. [69] conducted a survey on causal inference in recommender systems, exploring the challenges and future directions in this research area. Their survey investigates various causal inference techniques employed in recommender systems to infer causal relationships between users, items, and interactions. Through a comprehensive analysis, the authors identify key research gaps and propose future directions for advancing causal inference in recommender systems.

Natarajan et al. [70] addressed the data sparsity and cold start problems in collaborative filtering recommendation systems using linked open data. Their research focuses on leveraging linked open data sources to enrich user-item interaction data and alleviate data sparsity issues. Through empirical evaluation, the authors demonstrate the effectiveness of their approach in improving recommendation quality and addressing the cold start problem in collaborative filtering.

Da'u and Salim [71] conducted a systematic review of recommendation systems based on deep learning methods. Their research explores the application of

deep learning techniques, such as neural networks and deep autoencoders, in recommendation systems. Through a comprehensive analysis of existing literature, the authors identify key challenges and opportunities in leveraging deep learning for personalized recommendations, offering insights into future research directions.

In summary, these studies provide comprehensive insights into different aspects of recommendation systems, including machine learning techniques, causal inference, data sparsity resolution, and deep learning methods. By surveying recent advancements and emerging trends in recommendation systems research, these papers contribute to the understanding of state-of-the-art techniques and offer valuable insights for future research directions.

1.4.1 Knowledge-based recommender systems

Knowledge-based recommender systems leverage domain knowledge to provide personalized recommendations to users. This literature review explores five key papers, shown in Table 1, that delve into different aspects of knowledge-based recommendation systems, including ontology-based systems, sentiment analysis integration, deep learning, and efficiency improvements.

Table 1 – Summary of Knowledge-Based Recommendation Systems in Various Applications

Authors	Summary	Main Findings
R. Burke	Knowledge-based recommender systems	Provides an overview of knowledge-based recommender systems, focusing on their architecture and functionality.
J.K. Tarus, Z. Niu, G. Mustafa	Review of ontology-based recommender systems for e-learning	Evaluates ontology-based recommender systems in the context of e-learning, highlighting their application and effectiveness.
R.L. Rosa, G.M. Schwartz, W.V. Ruggiero, D.Z. Rodríguez	Knowledge-based recommendation system with sentiment analysis and deep learning	Presents a recommendation system incorporating sentiment analysis and deep learning techniques, emphasizing its integration of multiple approaches for enhanced performance.
B. Prasad	Knowledge-based product recommendation system for e-commerce	Introduces a knowledge-based recommendation system tailored for e-commerce platforms, focusing on product recommendations based on user preferences and behavior.
H. El Bouhissi, M. Adel, A. Ketam, A.-B.M. Salem	Development of an efficient knowledge-based recommendation system	Discusses efforts toward creating an efficient knowledge-based recommendation system, potentially addressing optimization techniques or advancements in the field.
Note – Compiled from source [72-76]		

Burke [72, p. 175-185] provides an early overview of knowledge-based recommender systems in the Encyclopedia of Library and Information Systems. The paper discusses how these systems utilize domain knowledge, such as taxonomies

and ontologies, to enhance recommendation accuracy and relevance. Burke outlines various approaches and techniques used in knowledge-based recommendation systems, laying the groundwork for subsequent research in the field.

Tarus, Niu, and Mustafa [73, p. 21-47] focus on ontology-based recommender systems for e-learning in their review paper. They explore how ontologies facilitate the representation and utilization of domain knowledge in recommendation processes. Through a comprehensive analysis of existing literature, the authors highlight the advantages of ontology-based approaches in enhancing recommendation quality and user satisfaction in e-learning environments.

Rosa et al. [74, p. 2124-2134] present a knowledge-based recommendation system that integrates sentiment analysis and deep learning techniques. Their research aims to improve the effectiveness of recommendation systems by considering user preferences and sentiments expressed in textual data. By leveraging deep learning models and sentiment analysis algorithms, the authors demonstrate enhanced recommendation accuracy and user satisfaction in industrial informatics applications.

Prasad [75, p. 18-35] introduces a knowledge-based product recommendation system for e-commerce. The paper focuses on leveraging domain knowledge about products and user preferences to generate personalized recommendations. Through a case study, Prasad demonstrates the effectiveness of the knowledge-based approach in improving recommendation accuracy and promoting customer engagement in e-commerce platforms.

El Bouhissi et al. [76, p. 38-48] propose enhancements to knowledge-based recommendation systems to improve efficiency. Their research addresses the computational challenges associated with large-scale knowledge bases and recommendation processes. By introducing optimization techniques and efficient algorithms, the authors aim to streamline recommendation generation and enhance scalability in knowledge-based systems.

In summary, these studies highlight the importance of domain knowledge in recommendation systems and explore various techniques for leveraging knowledge to enhance recommendation accuracy, relevance, and efficiency. By integrating ontologies, sentiment analysis, deep learning, and optimization techniques, researchers continue to advance knowledge-based recommendation systems, paving the way for more personalized and effective recommendations across different application domains.

1.4.2 Text-based recommender systems

Text-based recommendation systems have gained significant attention due to their ability to leverage textual information to provide personalized recommendations across various domains. This literature review discusses five key papers that explore different aspects of text-based recommendation systems shown in table 2, including text mining, collaborative filtering, and multi-criteria analysis.

Kanwal et al. [77] conducted a comprehensive review of text-based recommendation systems, covering various techniques and methodologies employed in text-based recommendation algorithms. Their research provides insights into the state-of-the-art approaches for processing textual data and integrating it into

recommendation systems. Through a systematic analysis, the authors identify key challenges and opportunities in text-based recommendation research.

Miao and Lang [78] proposed a recommendation system based on text mining, aiming to exploit textual information to enhance recommendation accuracy and relevance. Their research focuses on extracting valuable insights from textual data using text-mining techniques and incorporating them into the recommendation process. Through empirical evaluation, the authors demonstrate the effectiveness of their approach in improving recommendation quality.

Table 2 – Summary of Text-based Recommendation Systems in Different Domains

Authors	Summary	Main Findings
S. Kanwal, S. Nawaz, M. K. Malik, Z. Nawaz	Review of text-based recommendation systems	Examines text-based recommendation systems, exploring their methodologies, applications, and advancements in the field.
D. Miao, F. Lang	Text mining-based recommendation system	Introduces a recommendation system utilizing text mining techniques, potentially emphasizing the extraction of valuable insights from textual data for personalized recommendations.
H. Khatter, S. Arif, U. Singh, S. Mathur, S. Jain	Product recommendation system combining collaborative filtering and textual clustering	Presents a product recommendation system for e-commerce platforms, leveraging collaborative filtering and textual clustering methods to enhance recommendation accuracy.
R. K. Roul, K. Arora	Text summarization-based recommendation system for electronic products	Discusses a recommendation system that employs text summarization techniques for generating concise product descriptions, potentially aiding in personalized recommendations.
Y. Sharma, J. Bhatt, R. Magon	Multi-criteria review-based hotel recommendation system	Describes a hotel recommendation system that considers multiple criteria from reviews, potentially offering more comprehensive and tailored suggestions to users.
Note – Compiled from source [77, p. 31638-31660; 78-81]		

Khatter et al. [79, p. 612-617] presented a product recommendation system for e-commerce using collaborative filtering and textual clustering. Their research integrates collaborative filtering with textual clustering techniques to provide personalized recommendations based on both user-item interactions and textual product descriptions. Through experimentation, the authors demonstrate the effectiveness of their hybrid approach in enhancing recommendation accuracy in e-commerce platforms.

Roul and Arora [80, p. 13183-13203] conducted a review of text summarization-based recommendation systems for electronic products. Their research explores the use of text summarization techniques to generate concise representations of textual product descriptions, which are then used for

recommendation purposes. Through empirical analysis, the authors highlight the potential of text summarization in improving recommendation relevance and efficiency.

Sharma et al. [81, p. 687-690] proposed a multi-criteria review-based hotel recommendation system, aiming to provide personalized recommendations based on user reviews and preferences. Their research integrates multi-criteria analysis techniques with textual reviews to generate personalized recommendations tailored to individual user preferences. Through case studies and evaluation, the authors demonstrate the effectiveness of their approach in enhancing recommendation quality in the hospitality industry.

In summary, these studies provide valuable insights into different approaches and methodologies for developing text-based recommendation systems. By leveraging textual information and advanced data mining techniques, researchers have been able to enhance recommendation accuracy, relevance, and personalization across various application domains.

1.4.3 Hybrid recommender systems

Hybrid recommendation systems combine multiple recommendation approaches to improve recommendation accuracy, coverage, and user satisfaction. This literature review discusses seven key papers that explore different aspects of hybrid recommendation systems across various domains shown in table 3.

Table 3 – Summary of Hybrid Recommendation Systems in Various Domains

Authors	Summary	Main Findings
1	2	3
V. Kavinkumar, R. R. Reddy, R. Balasubramanian, M. Sridhar, K. Sridharan, D. Venkataraman	Hybrid recommendation system with added feedback component	Presents a recommendation system integrating multiple approaches, with an added feedback component, potentially enhancing recommendation accuracy and adaptability.
J. P. Lucas, N. Luz, M. N. Moreno, R. Anacleto, A.A. Figueiredo, C. Martins	Hybrid recommendation approach for a tourism system	Describes a hybrid recommendation approach tailored for a tourism system, potentially combining collaborative filtering, content-based filtering
Y. Tian, B. Zheng, Y. Wang, Y. Zhang, Q. Wu	College library persona lized recommendation system based on hybrid recommendation algorithm	Discusses a personalized recommendation system for college libraries, utilizing hybrid recommendation algorithms to cater to diverse user preferences and enhance user experience.
R. Passi, S. Jain, P.K. Singh	Hybrid recommendation system	Introduces a hybrid recommendation system, potentially integrating collaborative filtering, content-based filtering, or other techniques.
P.B. Thorat, R. M. Goudar, S. Barve	Survey on collaborative filtering, content-based filtering, and hybrid recommendation system	Presents a survey highlighting collaborative filtering, content-based filtering, and hybrid recommendation systems, potentially offering insights into the strengths and limitations of each approach.

Continuation of table 3

1	2	3
B. Walek, V. Fojtik	Hybrid recommender system for recommending relevant movies using an expert system	Proposes a hybrid recommender system for movie recommendations, potentially leveraging an expert system
K. Al Fararni, et. al.	Hybrid recommender system for tourism based on big data and AI	Presents a conceptual framework for a hybrid recommender system in tourism, integrating big data and AI techniques
Note – Compiled from source [82-88]		

Kavinkumar et al. [82, p. 745-751] proposed a hybrid approach for recommendation systems with an added feedback component. Their research integrates collaborative filtering and content-based filtering techniques, along with user feedback, to enhance recommendation accuracy and relevance. Through experimentation, the authors demonstrate the effectiveness of their hybrid approach in providing personalized recommendations.

Lucas et al. [83, p. 3532-3549] presented a hybrid recommendation approach for a tourism system. Their research combines collaborative filtering, content-based filtering, and knowledge-based techniques to provide personalized recommendations for tourists. By leveraging multiple recommendation approaches, the authors aim to address the diverse preferences and requirements of tourists in different contexts.

Tian et al. [84, p. 490-493] developed a college library personalized recommendation system based on a hybrid recommendation algorithm. Their research integrates collaborative filtering and content-based filtering techniques to recommend library resources tailored to individual user preferences and interests. Through case studies, the authors demonstrate the applicability and effectiveness of their hybrid recommendation system in enhancing user satisfaction.

Passi et al. [85, p. 117-127] proposed a hybrid approach for recommendation systems, combining collaborative filtering, content-based filtering, and knowledge-based techniques. Their research aims to leverage the strengths of each recommendation approach to provide more accurate and diverse recommendations to users. Through empirical evaluation, the authors demonstrate the effectiveness of their hybrid approach in addressing the limitations of individual recommendation techniques.

Thorat et al. [86, p. 31-35] conducted a survey on collaborative filtering, content-based filtering, and hybrid recommendation systems. Their research provides insights into the strengths, weaknesses, and application domains of different recommendation approaches. By analyzing existing literature, the authors highlight the importance of hybrid recommendation systems in improving recommendation quality and user satisfaction.

Walek and Fojtik [87, p. 113452] developed a hybrid recommender system for recommending relevant movies using an expert system. Their research integrates collaborative filtering, content-based filtering, and rule-based expert systems to provide personalized movie recommendations to users. Through experimentation, the

authors demonstrate the effectiveness of their hybrid approach in improving recommendation accuracy and coverage.

Al Fararni et al. [88, p. 47-54] proposed a hybrid recommender system for tourism based on big data and artificial intelligence. Their research integrates collaborative filtering, content-based filtering, and knowledge-based techniques, leveraging big data analytics and AI algorithms to provide personalized recommendations to tourists. Through a conceptual framework, the authors demonstrate the potential of their hybrid system in enhancing tourist experiences and satisfaction.

In summary, these studies highlight the significance of hybrid recommendation systems in providing more accurate, diverse, and personalized recommendations across various domains. By integrating multiple recommendation approaches, hybrid systems aim to overcome the limitations of individual techniques and enhance the overall recommendation performance.

Tawfik et al. [89] employed statistical analysis to develop a case-based recommendation system for educational environments. This method involved initial evaluations by five experts who rated problem-related topics and narratives within the case library, ensuring the system's alignment with educational goals and enhancing learning through targeted case retrieval based on their assessments. Based on the comparison between Tawfik et al. and the current research outlined in the table, several notable differences emerge. Tawfik et al. focused on developing a recommendation system to support problem-solving in educational settings, whereas the current research also centers on a recommendation system but spans a broader scope with 107 problems compared to Tawfik et al.'s 6 problems. In terms of categorization, Tawfik et al. categorized cases into 20 categories, whereas the current research employs 5 categories. Additionally, while Tawfik et al. relied on expert opinions for rating, the current research integrates machine learning algorithms for its assessment framework, diverging from Tawfik et al.'s use of statistical rating methods. These differences highlight advancements and shifts in methodology and scope within the field of recommendation systems and problem-solving in educational contexts.

2 MAPREDUCE TASK-BASED CLASSIFICATION AND CURRICULUM EVALUATION

2.1 Instructional design methodology

The utilization of the ADDIE model, a renowned instructional design framework, is pivotal in addressing primary research question. This systematic and methodical approach serves as the cornerstone for the meticulous crafting of instructional materials and activities that not only foster effectiveness but also efficiency. The ADDIE process encompasses five distinct yet interrelated stages: Analysis, Design, Development, Implementation, and Evaluation. During the analysis phase, a comprehensive collection of pertinent information concerning the target audience, their specific requirements, and the overarching instructional objectives of the course is meticulously undertaken. This invaluable data is sourced from a myriad of reliable channels, including expert opinions and a diverse array of research methodologies such as needs analysis and formative evaluations. Subsequently, the insights gleaned from this analysis are meticulously integrated into the design phase, where instructional materials and activities are finely tailored to impeccably suit the unique needs of the audience while impeccably aligning with the educational objectives at hand. This phase is characterized by the meticulous identification of specific content and skills, alongside the establishment of a meticulously structured course outline to guide the learning journey. As the development phase ensues, meticulous attention is dedicated to the creation of instructional materials and activities, ensuring seamless alignment with established best practices and evidence-based approaches within the field. Implementation, the subsequent stage, calls for the actual delivery of instruction by teachers, underpinned by expert input to uphold fidelity to the intended instructional approach. Finally, the evaluation stage serves as the definitive litmus test, meticulously assessing the effectiveness of the instructional materials, activities, and the overall course. This comprehensive assessment endeavor scrutinizes whether the instructional goals were effectively met while concurrently gauging the reception of students towards the learning experiences provided [90].

2.2 Summative evaluation and statistical analysis

The summative evaluation method is implemented as a comprehensive strategy to thoroughly assess the efficacy of a course upon its completion. This evaluative approach entails a meticulous examination aimed at quantifying the extent to which the course objectives have been achieved, thereby enabling requisite refinements for subsequent iterations. Within the realm of summative evaluation, one methodological approach widely employed involves the utilization of an experimental design framework, wherein both a control group and an experimental group are utilized for comparative analysis. In this study, the experimental cohort, comprising 21 students, was exposed to an innovative instructional methodology, while an equivalent number of students in the control group received conventional teaching practices. Notably, the demographic profile of students in both groups is detailed, highlighting factors such as age range (19-20 years old) and academic year (predominantly in their third or

fourth year of study), as well as gender distribution, with ratios specified for males and females in both groups. To ascertain the presence of statistically significant differences in mean performance between the experimental and control groups, a rigorous analysis was conducted employing a statistically independent sample t-test. Before conducting this test, an outlier analysis utilizing box plots was carried out to identify and address any outliers that may unduly influence the data analysis process, thereby ensuring the integrity and accuracy of current findings. Furthermore, to uphold the principles of robust sampling methodology, the initial selection process was subjected to rigorous hypothesis testing, particularly focusing on student GPAs, to ensure the sample's representativeness and the randomness of the sampling procedure, thereby enhancing the validity and reliability of the study outcomes.

2.3 Implementation-based classification of single machine tasks

In this section, the feasibility of developing a curriculum that effectively imparts the core principles of MapReduce on Apache Spark, specifically targeting the classification of single-machine tasks within distributed systems, is explored. To address the deficiencies detected in the traditional Distributed Big Data Systems course, the ADDIE methodology, a structured instructional design approach, was utilized over an approximate duration of eight weeks. Initially, the course's learning outcomes were meticulously scrutinized by a panel of three experts, each possessing pertinent expertise, revealing their failure to meet expectations. Consequently, a novel curriculum was proposed by these astute experts, grounded in their invaluable insights and experience. This endeavor entailed the formulation of fresh course objectives imbued with active learning methodologies aimed at enhancing the overall learning experience. Following this, the responsibility of designing the course content, materials, and potential activities was assumed by the primary author of the study, tasked with translating the expert-derived vision into tangible educational assets.

Throughout the subsequent development phase, regular intervals were convened by the expert panel, spanning an approximate duration of five weeks, to meticulously refine the course content and activities, ensuring alignment with their exacting standards. Through a rigorous process encompassing analysis, design, and development, the course materials underwent meticulous refinement, ensuring their coherence and effectiveness in achieving the desired learning outcomes. The resulting curriculum delineated five distinct classes, each intricately juxtaposing a single-machine solution with a distributed MapReduce solution. For instance, tasks such as pirate speech and log analysis (SQL injection) were identified as relatively straightforward to implement on MapReduce, owing to the absence of intricate data interactions. In such scenarios, the functionality of only the map steps is deemed feasible, devoid of a reducer. The mapping process closely mirrors that of a single-machine solution but operates on a line-by-line basis, iteratively processing each line until the culmination of the dataset. Additionally, while sentiment analysis emerged as another task amenable to execution without a reducer (see figure 2), alternative solutions were also thoughtfully deliberated within the confines of the research, fostering a comprehensive exploration of the subject matter.

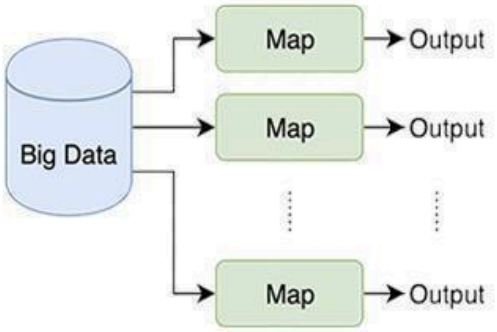

Figure 2 – Job process without reducer

Another notable issue arises in the realm of counting, encompassing challenges such as word count and Google Play frequency analysis. Fortunately, resolving these issues is relatively straightforward: the map function effectively partitions the data and dispatches it to reducer keys necessitating enumeration, assigning a uniform value of 1 to each instance. Consequently, these individual values are amalgamated and transmitted to the reducer as a collective set of 1s. Upon receipt, the reducer meticulously processes each key alongside its corresponding set of values. In these instances, the reducer diligently aggregates the values, culminating in the emission of both the key and its respective count. This pivotal process is aptly illustrated in figure 3 for further clarity and understanding.

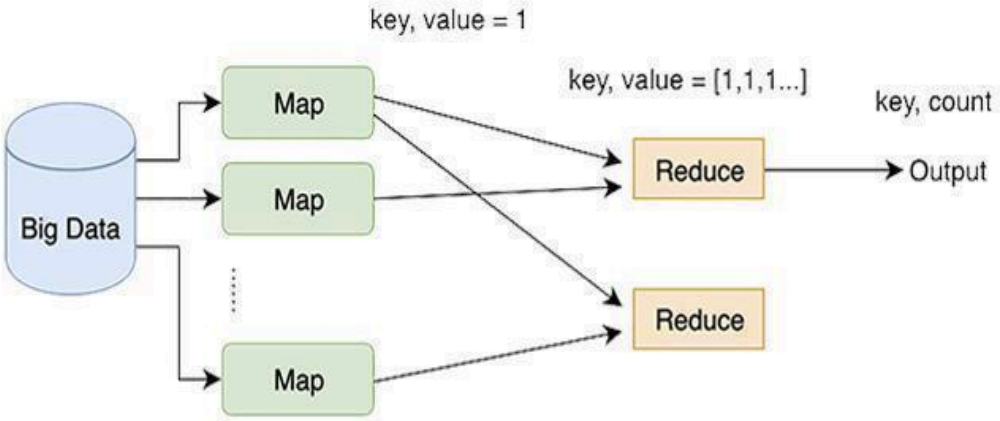

Figure 3 – MapReduce process, WordCount example

The issue of trending word count encapsulates the intricacies of data integration, where the results from one MapReduce job seamlessly transition into serving as input for subsequent jobs. Additionally, this scenario extends to situations

where the outputs generated by multiple MapReduce tasks converge to form the input for a singular subsequent MapReduce operation. Illustrated in figure 4, this dynamic elucidates the complex interplay within the environment, particularly shedding light on the shuffle phase intricacies encompassing sorting and combining processes. Through this depiction, users are afforded a deeper comprehension of the underlying mechanisms governing data organization and processing during this pivotal stage of computation.

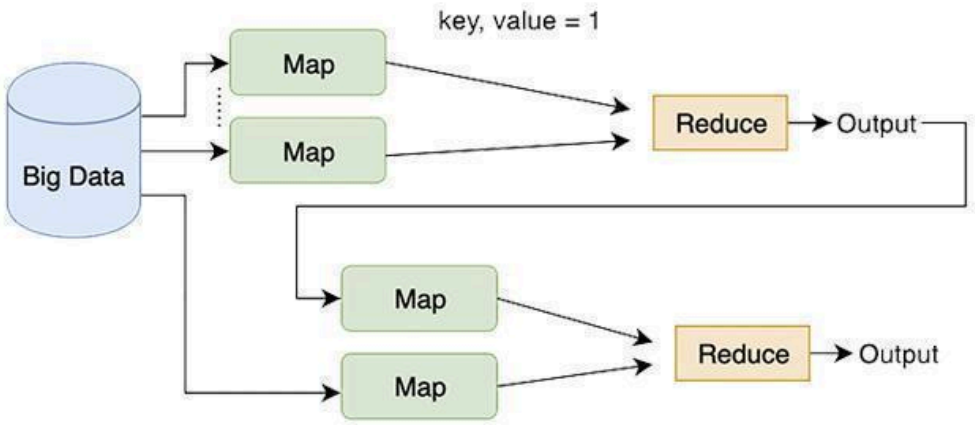

Figure 4 – MapReduce process, joining data

Difficulties arise in scenarios characterized by a limited number of keys yet an abundance of associated values, a common occurrence when tackling tasks such as determining the minimum, maximum, and conducting k-means analysis. In such situations, the keys are typically predefined, setting the stage for the mapping phase to discern and align the relevant values with their corresponding keys. Following this initial step, the reducer assumes the mantle of processing each key alongside its associated set of values. Through a series of carefully orchestrated actions, the reducer meticulously manipulates the data set, culminating in the emission of the key paired with the resultant outcome. This intricate process, crucial for tasks involving data aggregation and analysis, is succinctly depicted in figure 5, providing a visual aid to elucidate the underlying procedures for enhanced comprehension.

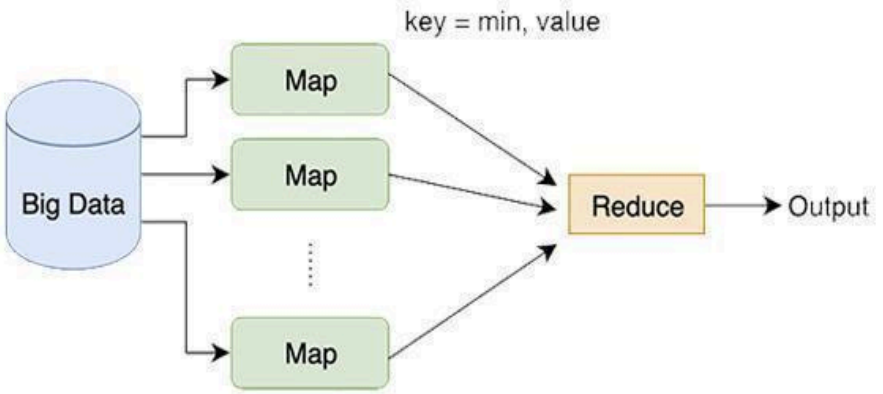

Figure 5 – MapReduce process, finding min/max example

Implementing decision tree and apriori algorithm tasks poses significant challenges, comparable to those encountered in single-machine solutions. Within this context, data lines interact, resulting in dynamic changes to both keys and values throughout the execution process. Moreover, these tasks necessitate the repeated iteration of the MapReduce job until the specified condition is satisfied. The intricacies of this iterative process are visually represented in figure 6, providing a comprehensive illustration of the iterative nature inherent in these tasks within the MapReduce framework. Through this visual aid, readers gain a deeper understanding of the complexities involved in executing decision tree and apriori algorithm tasks in a distributed computing environment.

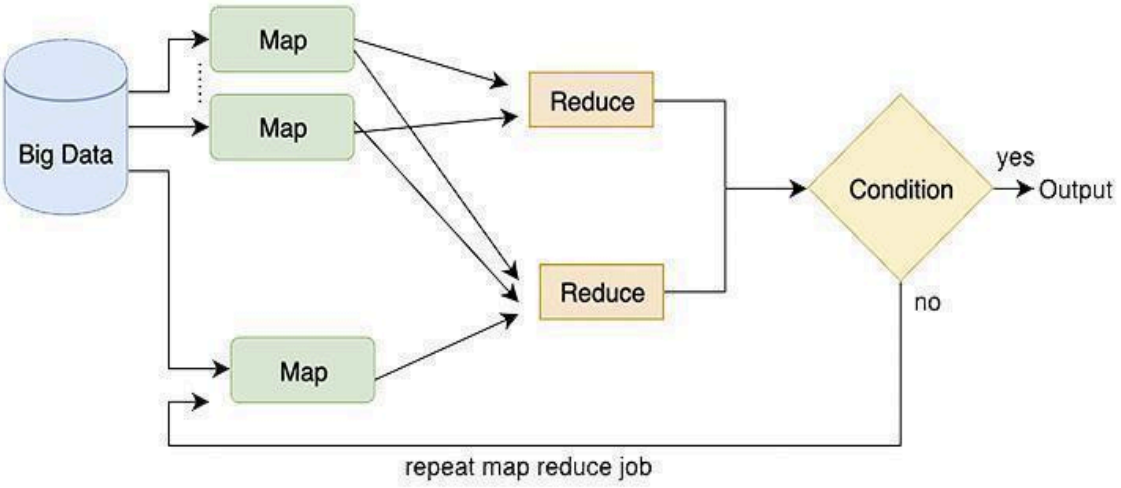

Figure 6 – MapReduce process, with condition

The implementation of the new teaching methodology spanned a period of approximately two weeks, marking a notable departure from the traditional six-week format previously utilized. In the inaugural class session, the experimental group was introduced to the innovative approach along with the accompanying instructional materials. To aid in comprehending the intricacies of the MapReduce concept, students were provided with an expanded version of the previously covered material for self-study purposes (table 4). This supplementary resource aimed to bolster their understanding.

Following this introductory phase, a dynamic role-playing exercise unfolded in subsequent sessions. Here, students assumed diverse roles within the Hadoop environment, ranging from mappers and reducers to data batch processors. Through this engaging activity, participants were encouraged to immerse themselves in the practical application of course concepts, fostering a deeper understanding.

Subsequent class sessions adopted a problem-based active learning strategy, offering students a range of take-home assignments designed to stimulate critical thinking and practical application of knowledge. Regular assessments were administered to monitor student progress, with tailored feedback provided to support ongoing improvement efforts.

Throughout the duration of the course, experts made periodic visits to the classroom setting, organizing meetings to assess the effectiveness of the newly introduced teaching approach. Despite the condensed timeframe, students faced challenges in fully mastering the material. However, the integration of problem-based learning and active learning strategies was strategically implemented to address these potential limitations.

Table 4 – Ten MapReduce problems and their classified types

Title	Type
WordCount*: the aim is to calculate the frequency of appearance of words in text	#2 (related to counting)
Pirate speech** aims to change the text's style to a pirate's style. Ex.: change "ing" to "in'", "the" to "da" etc.	#1 (lines of data do not interact with each other)
Google Play frequency: the aim is to find how many applications were created on Google Play each month.	#2 (related to counting)
Log analysis***: the aim is to determine the SQL injection and DDOS attacks9	#1 (lines of data do not interact with each other), #2 (related to counting)
Finding min and max****: the aim is to find min and max	#3 (contains few known keys and many unknown values)
Trending WordCount #: the aim is to identify the frequency of word appearance by date and total in Twitter, expected output: word, date, sum_date, total_sum	#4 (needs to join data and use the output of one MapReduce job in another MapReduce job)
Sentiment analysis##: the aim is to identify if the sentence is positive, negative, or neutral.	#1 (lines of data do not interact with each other), #2 (related to counting)
K-means clustering###: the aim is to find centroids of clusters	#3 (contains few known keys and many unknown values), #5 (problem is related to the condition)
Decision Tree####: the aim is to build decision tree	#5 (the problem is related to the condition)
Apriori algorithm#####: the aim is to calculate the frequency of a set with symptoms of diseases.	#5 (problem is related to the condition)
* – [91]; ** – [92]; *** – [93]; **** – [91, p. 17-22]; # – [92, p. 103-107]; ## – [94]; ### – [95]; #### – [96]; ##### – [97]	

In summation, the new teaching methodology demonstrated its efficacy in fostering a more profound understanding of the MapReduce programming model

among students. Future research endeavors could explore the longitudinal impact of this approach across extended timeframes and within diverse educational contexts, further enriching the understanding of its effectiveness.

2.4 Evaluation of the novel approach

In this section, the assessment findings of the proposed methodology are presented. A hypothesis test was conducted to compare the average GPA between the control group (with a mean of 2.73 and standard deviation of 0.56) and the experimental group (with a mean of 2.80 and standard deviation of 0.64). The results of the test revealed a t-value of -0.37 and a p-value of 0.71, indicating insufficient evidence to reject the null hypothesis, which asserts equality in mean GPAs between the two groups. This suggests that the distribution of GPAs in both groups can be considered random.

An independent sample t-test was employed to examine whether students' performance improves with the proposed novel approach as opposed to the traditional teaching method. Before conducting the t-test, outlier detection was carried out using box plots, as depicted in figure 7.

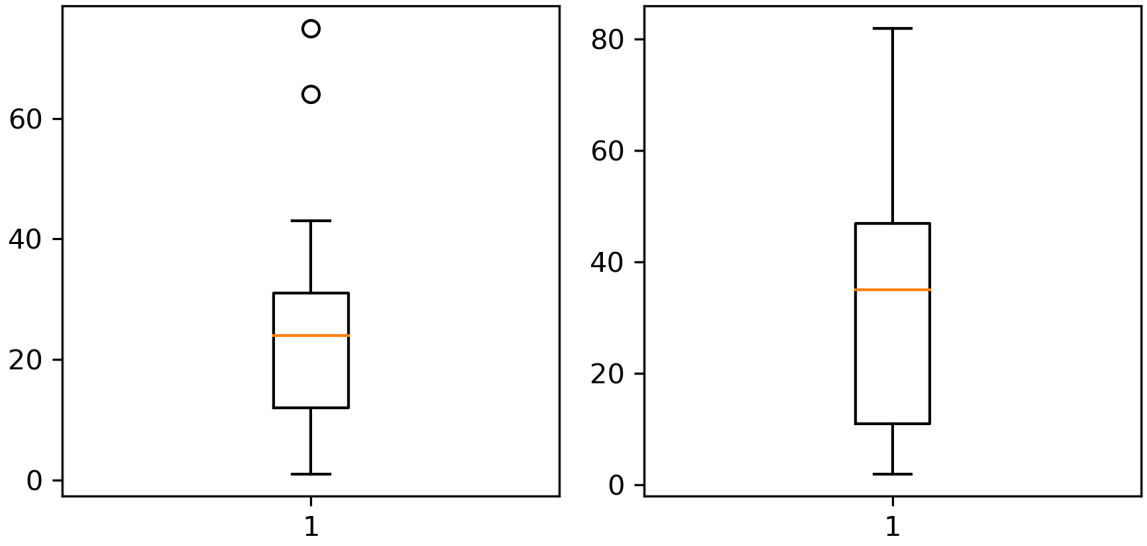

Figure 7 – The box plot of MapReduce exam grades for control and experimental groups, respectively

Upon scrutinizing the box plots, it becomes apparent that two outliers are evident within the control group. Upon their removal, the control group's composition narrows down to 19 students, displaying an average score of 20.47 (with a standard deviation of 12.25). In contrast, the experimental group comprises 21 students, showcasing an average score of 32.67 (with a standard deviation of 25.12). Employing an independent sample, one-tailed t-test enabled an exploration into the potential statistical disparity between the mean scores of the experimental and control groups. The resulting analysis yielded a t-value of -1.92 alongside a corresponding p-value of 0.0313. Given that the calculated p-value falls below the conventional

threshold of 0.05, it substantiates the assertion that the experimental group markedly surpasses the control group in performance.

In conclusion, the findings derived from this study offer promising insights, suggesting that the novel teaching approach under examination holds the potential to significantly elevate student performance in MapReduce examinations compared to conventional teaching methodologies. However, it's imperative to acknowledge the imperative for further extensive investigations with larger sample sizes to validate and consolidate these preliminary findings. Such comprehensive studies would lend greater credence and robustness to the observed trends and conclusions drawn herein.

The primary objective of this study was to delve into novel methodologies aimed at teaching the intricacies of the MapReduce programming model, with a specific focus on task-based classification, and subsequently, to gauge the effectiveness of these methodologies on student performance. In pursuit of this objective, the study formulated two primary research questions.

The first inquiry aimed to meticulously identify and categorize various problem types that lend themselves to solutions using the MapReduce programming model. To accomplish this task, an instructional design methodology was meticulously applied, complemented by invaluable input from experts in the field. Through this collaborative effort, tasks were systematically classified into five distinct problem categories, encompassing scenarios ranging from those necessitating no reducers to issues involving intricate data joining and key and value shuffling. These categorizations, thoughtfully presented in Table 4, serve as a comprehensive resource for readers, aiding in the nuanced understanding of each problem category and facilitating efficient problem classification and solution retrieval.

The second research question sought to ascertain the efficacy of the proposed teaching model. To this end, a summative evaluation methodology was employed, with the resulting data subjected to rigorous analysis utilizing a statistically independent sample t-test. The outcomes of this analysis revealed a noteworthy statistical difference ($p < 0.05$) in student performance on the summative MapReduce examination, clearly favoring the proposed novel approach over traditional teaching methods.

In summation, this study aimed to explore innovative teaching methodologies for the MapReduce programming model and evaluate their impact on student performance comprehensively. By meticulously identifying and categorizing various problem types, the study not only underscored the effectiveness of task-based classification in facilitating MapReduce learning but also showcased the superiority of the proposed approach in enhancing student performance compared to conventional methods. Additionally, readers are empowered with a deeper comprehension of the MapReduce concept and problem-solving strategies, even in the absence of extensive experience in MapReduce job writing.

Moreover, the study emphasized the dynamic nature of MapReduce problem domains, hinting at the need for ongoing exploration into new problem categories to expand the utility of the MapReduce programming model further. Furthermore, while the study primarily focused on evaluating student performance through summative examinations, future research endeavors could delve into exploring the long-term

retention of knowledge and skills acquired through this approach. Additionally, efforts to balance theoretical explanations with practical exercises and real-world examples were made; however, subsequent research could explore strategies to provide additional support to students grappling with technical aspects.

As a policy implementation, the proposed classification methodology holds immense promise for adoption in both educational and professional settings, offering invaluable support in acquiring the requisite knowledge and skills for effective utilization of distributed systems in real-world scenarios.

3 RECOMMENDER SYSTEM FOR ADAPTING SINGLE MACHINE PROBLEMS TO DISTRIBUTED SYSTEMS WITHIN MAPREDUCE

3.1 Data, models training, and evaluation

In the process of data collection, an exhaustive dataset comprising a total of 107 unique problem instances was meticulously compiled. This initial dataset originated primarily from two esteemed literary works within the distributed systems and big data processing domain: "Hadoop in Action" [92, p. 3-330] and "MapReduce Design Patterns: Building Effective Algorithms and Analytics for Hadoop and Other Systems" [91, p. 3-248]. These foundational problems covered a wide array of essential concepts and standard algorithms commonly utilized in MapReduce applications, spanning from fundamental tasks like word count and pirate talk to more intricate challenges involving the determination of minimum and maximum values.

Furthermore, to enrich the dataset with a broader spectrum of complexities and real-world scenarios, additional problem instances were extracted from scholarly articles. These supplementary problems delved into advanced topics such as sentiment analysis [94, p. 4-7], k-means classification [95, p. 247-260], decision trees [96, p. 49-59], and the Apriori algorithm [97, p. 571-580]. By integrating problems sourced from both authoritative books and scholarly literature, the aim was to ensure the dataset comprehensively represented the diverse range of problem instances encountered in practical MapReduce applications, thereby providing a robust foundation for the subsequent phases of the study.

The assembled dataset encompassed a multitude of attributes, including to Title, Category, Keywords, Input, Output, and Goal, ensuring a comprehensive representation of the problem instances. The selection process for problems adhered to a dual-pronged approach: firstly, a concerted effort was made to incorporate problems showcasing a diverse array of methodologies and approaches towards resolving distributed systems challenges. This deliberate strategy aimed to imbue the dataset with a rich variety of techniques, thus facilitating a holistic understanding of problem-solving paradigms within the MapReduce framework. Secondly, emphasis was placed on including problems exhibiting similarities in their solution strategies yet diverging in their specific problem contexts. By prioritizing such problems, the dataset was engineered to enable a nuanced exploration of the efficacy of various algorithms and techniques across a spectrum of real-world scenarios.

The overarching objective of the data collection endeavor was to curate a dataset that not only captured the breadth and depth of MapReduce problem instances but also served as a robust foundation for comprehensive problem classification endeavors.

The meticulous selection of problems for dataset inclusion was spearheaded by domain experts possessing a profound understanding of distributed systems operating under the MapReduce paradigm. Following a rigorous analysis of the dataset, three independent experts meticulously identified five primary characteristics encapsulating solutions to challenges prevalent within distributed MapReduce systems:

1. MR category - lines of data do not interact with each other.
2. MR category - related to counting.
3. MR category - contains few known keys and many unknown values.
4. MR category - needs to join data and use the output of one MapReduce job in another MapReduce job.
5. MR category - the problem is related to the condition.

This rigorous vetting process culminated in the formulation of a meticulously curated problems dataset, as exemplified in table 5. Each problem was meticulously assigned a binary label "0" or "1," denoting its adherence or deviation from the established characteristics, based on prior research conducted by the authors [98].

Table 5 – Labeled data: examples of classified problems by five categories

Problem attributes (title, category, keywords, input, output, goal)	#1 MR category - lines of data do not interact with each other	#2 MR category - related to counting	#3 MR category - contains few known keys and many unknown values	#4 MR category - needs to join data and use the output of one MapReduce job in another MapReduce job	#5 MR category - the problem is related to the condition
problem 1	1	0	0	0	0
problem 2	1	1	1	0	0
problem 3	0	1	1	1	1
…	…	…	…	…	…

The preprocessing stage involves several meticulous steps to ensure the effective transformation of textual data into numerical representations suitable for classification tasks. Initially, textual features extracted from the 'Goal' column undergo TF-IDF vectorization, a process that converts them into numerical representations shown in figure 8. This transformation is crucial for facilitating further analysis and modeling. Additionally, to maintain consistency and enhance the quality of the data, the vectorizer is configured to exclude common English stop words and to standardize all text to lowercase.

```
text_features =  problems_data['Goal']
tfidf_vectorizer = TfidfVectorizer(stop_words='english', lowercase = True)
X = tfidf_vectorizer.fit_transform(text_features)
Y_1 = problems_data['DataNotInteract'].values
Y_2 = problems_data['CountingRelated'].values
Y_3 = problems_data['FeatureX'].values
Y_4 = problems_data['JoinData'].values
Y_5 = problems_data['ConditionProblem'].values
```

Figure 8 – Example of preprocessing

Subsequently, target variables essential for classification, including 'DataNotInteract', 'CountingRelated', 'FeatureX', 'JoinData', and 'ConditionProblem',

are identified and extracted from relevant columns within the dataset. Moreover, to enrich the feature set and provide a more comprehensive understanding of the data, various combinations of text features are generated by concatenating different columns. These combinations, such as 'Goal' and 'Title', 'Goal', 'Title', and 'Category', and others, aim to capture diverse aspects of the problem instances (figure 9).

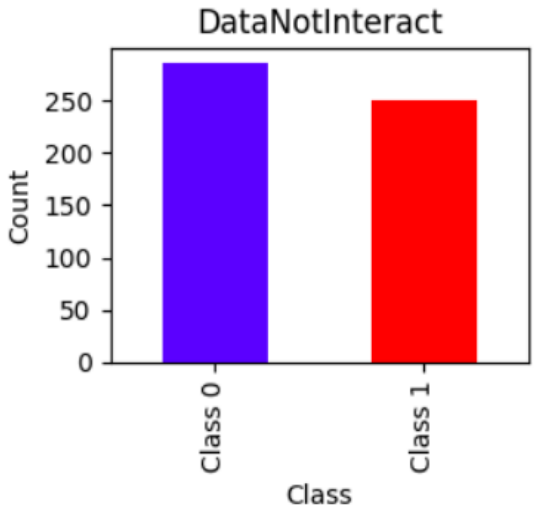

Figure 9 – Balance between labels in #1 MR category

For each of these feature combinations, TF-IDF vectorization is again applied using the previously configured settings. This meticulous process ensures that all textual data is effectively transformed into numerical representations while maintaining the integrity of the dataset. Furthermore, the target variables remain consistent across all feature combinations, ensuring coherence and facilitating subsequent analysis (figure 10).

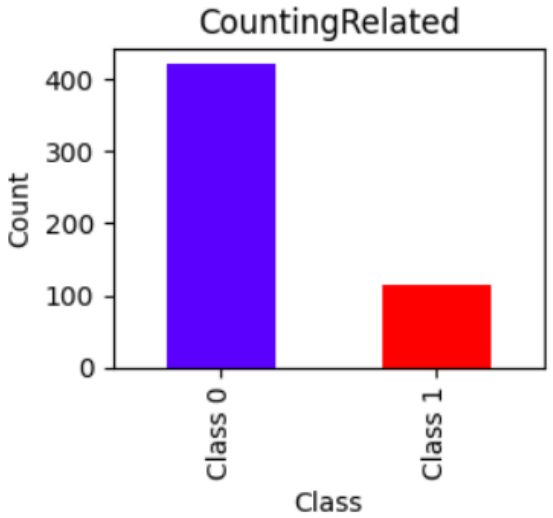

Figure 10 – Imbalance between labels in #2 MR category

To address the challenge posed by the limited size of the dataset, an artificial expansion strategy is implemented. This strategy leverages a paraphrasing technique facilitated by the nlpaug library, utilizing advanced contextual word embedding models like BERT. By generating paraphrased versions of existing textual data, the dataset is expanded, thereby increasing its diversity and mitigating limitations associated with its initial size.

In parallel, feature selection is conducted to identify the most informative attributes for classification models. Various combinations of attributes, including 'Goal', 'Title', 'Category', 'Input', 'Output', and 'Keywords', are explored to comprehensively evaluate their impact on model performance. This iterative process ensures that the selected features effectively capture the essential characteristics of the problem instances.

After the meticulous process of data collection, it became evident that the collected dataset exhibited an imbalance between the binary labels "0" and "1", shown in figures 9, 10, 11, 12, 13, with a notably higher frequency of instances labeled as "0" compared to those labeled as "1". This imbalance posed a challenge for subsequent classification tasks, potentially leading to biased model outcomes. A resampling method known as RandomUnderSampler was employed to address this issue. This technique systematically reduced the number of instances belonging to the majority class (in this case, instances labeled as "0") to achieve a more balanced distribution between the two classes. By randomly undersampling the abundant class, the dataset was rebalanced, ensuring that both classes were represented more equally, thus enhancing the effectiveness and fairness of subsequent classification models.

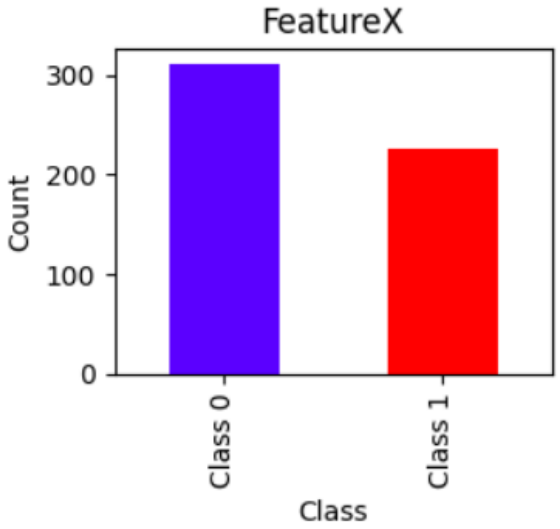

Figure 11 – Imbalance between labels in #3 MR category

As shown in figure 10, figure 12, and figure 13, it became evident that a notable imbalance exists within the categories CountingRelated, JointData, and ConditionProblem. This imbalance is primarily attributed to the scarcity of tasks meeting these criteria within openly available resources, which is reflected not only in the collected dataset but also across various open repositories and sources.

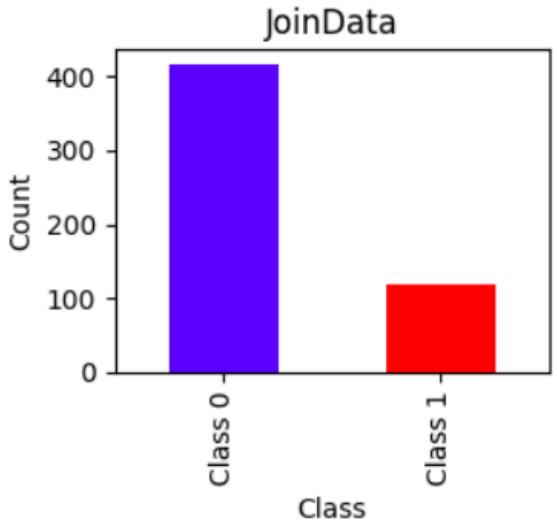

Figure 12 – Imbalance between labels in #4 MR category

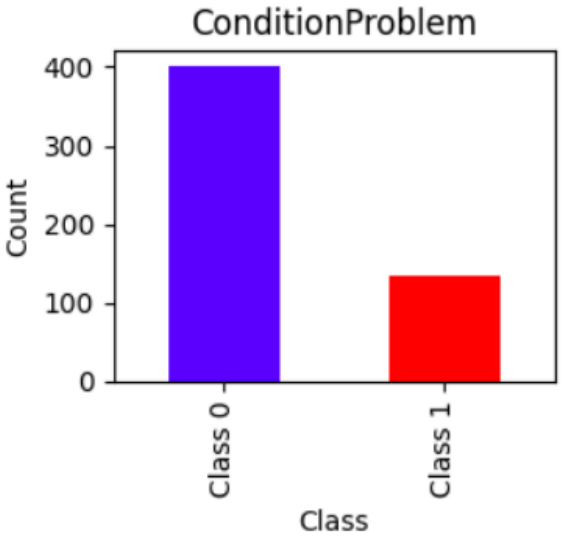

Figure 13 – Imbalance between labels in #5 MR category

The experiment, shown in figure 14, involves several key steps aimed at hyperparameter tuning and model selection using Logistic Regression. Initially, a parameter grid is defined to facilitate hyperparameter tuning, focusing specifically on the regularization parameter 'C', with values ranging from 0.001 to 10.

Subsequently, a list of target variables, denoted as Y_values, is created to represent different binary classification tasks. These target variables are crucial for splitting the dataset into multiple subsets for training and testing.

The experiment then proceeds through a loop iterating over each target variable in Y_values. For each iteration, the data is divided into training and testing sets using train_test_split(). A RandomUnderSampler instance is applied to the training data to address dataset imbalance effectively.

Hyperparameter tuning is conducted using Logistic Regression within a nested loop. Each value of 'C' from the parameter grid is iterated over. For each 'C', a Logistic Regression model is instantiated, trained on the resampled training data, and evaluated on the testing data using the F1 score metric. The 'C' value leading to the highest F1 score is identified as the optimal hyperparameter for the model.

The best model obtained for each target variable and its corresponding F1 score and 'C' value are printed.

```
param_grid = {'C': [0.001, 0.01, 0.1, 1, 10]}

Y_values = [Y_1, Y_2, Y_3, Y_4, Y_5]
models = []

for i, Y in enumerate(Y_values):
    X_train, X_test, Y_train, Y_test = train_test_split(
        X, Y, test_size=0.2, random_state=42)

    undersampler = RandomUnderSampler(random_state=42)

    X_resampled, Y_resampled = undersampler.fit_resample(X_train, Y_train)

    skf = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
    grid_search = GridSearchCV(LogisticRegression(random_state=42),
                               param_grid, cv=skf, scoring='f1')
    grid_search.fit(X_resampled, Y_resampled)
    best_model = grid_search.best_estimator_

    Y_pred = best_model.predict(X_test)
    best_f1 = f1_score(Y_test, Y_pred)

    print(f"Best F1 Score for Y_{i + 1}: {best_f1} (C={best_model.C})")
```

Figure 14 – Optimizing Logistic Regression Models with Cross-Validation

Moreover, it's noteworthy to mention that cross-validation using Stratified K-Fold technique has been integrated into the hyperparameter tuning process, ensuring robustness in model evaluation. Finally, the best model for each target variable is saved.

The experiment aims to assess the effectiveness of Gaussian Naive Bayes when combined with cross-validation and random undersampling techniques. The primary objective is to determine whether this combination enhances the model's predictive capabilities in binary classification tasks. The experiment commences with the preparation of binary target variables, denoted as Y_1 to Y_5, each representing a distinct classification task. Subsequently, the dataset undergoes preprocessing, including the splitting into training and testing sets with an 80-20 ratio and the application of random undersampling to address class imbalance. For each classification task, Gaussian Naive Bayes models are trained and evaluated using 5-fold cross-validation on the resampled training data, with the F1 score serving as the evaluation metric (figure 15).

```
Y_values = [Y_1, Y_2, Y_3, Y_4, Y_5]
models = []

for i, Y in enumerate(Y_values):
    X_train, X_test, Y_train, Y_test = train_test_split(
        X, Y, test_size=0.2, random_state=42)
    undersampler = RandomUnderSampler(random_state=42)

    X_resampled, Y_resampled = undersampler.fit_resample(X_train, Y_train)

    best_f1 = 0
    best_model = None
    alpha_values = [1.0, 0.1, 0.01, 0.001, 0.0001]
    best_alpha = None

    for alpha in alpha_values:
        model = GaussianNB(alpha=alpha)
        cv_scores = cross_val_score(model, X_resampled, Y_resampled, cv=5, scoring='f1')
        mean_cv_score = cv_scores.mean()
        if mean_cv_score > best_f1:
            best_f1 = mean_cv_score
            best_model = model.fit(X_resampled, Y_resampled)
            best_alpha = alpha

    print(f"Best F1 Score for Y_{i + 1}: {best_f1} (Alpha={best_alpha})")

    if best_model:
        model_filename = f"{save_path}naive_bayes_model_Y_{i + 1}.joblib"

print("All resampled models saved successfully.")
```

Figure 15 - Optimizing Naive Bayes Models with Cross-Validation

The mean F1 score across cross-validation folds is computed, and if it surpasses the previous best score, the model is retrained on the entire resampled training data and considered the best model. The experiment outputs the best F1 score achieved for each task, and the optimized models are saved for future use.

3.2 Results and Discussion

The Logistic Regression model underwent extensive testing across incremental sets of textual features, affirming its efficacy in addressing the binary classification problem and its ability to leverage diverse textual features to enhance predictive accuracy significantly. A thorough comparative analysis between the Naive Bayes and Logistic Regression models was conducted to glean deeper insights into their predictive capabilities and identify potential areas of strength and improvement.

Employing a hybrid approach, the study capitalized on the strengths of both Naive Bayes and Logistic Regression models. Through a meticulous optimization process, the research aimed to pinpoint the most effective model and feature combination for each column, highlighting the adaptability of employing different models and feature sets to address distinct target variables. This nuanced strategy aimed to harness the unique advantages offered by each model, resulting in a tailored and optimized predictive framework customized to the specific requirements of each column within the dataset.

By exploring various combinations of models and features, the study sought to enhance the predictive performance across all target variables. This iterative process allowed for a comprehensive examination of the predictive capabilities of different models and feature sets, ultimately leading to the development of a robust and reliable predictive framework for each column (table 6).

Table 6 – Naive Bayes Results

MR category	Features	Best Cross-Validated F1 Score	Best F1 Score on Test Set	Alpha
#1 MR category	Goal+Title+Category+Input+Output	0.99	1	0.01
#2 MR category	Goal+Title+Category	0.94	1	0.01
#3 MR category	Goal+Title+Category+Input+Output+Keywords	0.99	1	0.001
#4 MR category	Goal+Title+Category+Input+Output	0.96	0.9742	0.001
#5 MR category	Goal+Title+Category	0.98	0.9740	0.001

Table 6 provides a comprehensive overview of the tangible results obtained from our Naive Bayes models applied across various MapReduce (MR) categories, shedding light on the impact of employing different feature combinations. Particularly noteworthy is the exceptional performance observed in the first MR category, where our model achieved an outstanding Best Cross-Validated F1 Score of 0.99 and a flawless Best F1 Score of 1 on the test set, highlighting its remarkable predictive accuracy. Moreover, the second and third categories also displayed robust capabilities, further emphasizing the effectiveness of the feature sets selected for our analysis. These findings underscore the reliability and versatility of our approach in effectively addressing diverse problem categories within the MapReduce framework.

Table 7 – Logistic Regression Results

MR category	Features	Best Cross-Validated F1 Score	Best F1 Score on Test Set	C
#1 MR category	Goal+Title+Category+Input+Output	0.98	1	1
#2 MR category	Goal+Title	0.98	1	1
#3 MR category	Goal+Title+Category+Input	0.97	1	1
#4 MR category	Goal+Title+Category+Input+Output+Keywords	0.97	1	1
#5 MR category	Goal+Title+Category+Input+Output+Keywords	0.99	1	1

Table 7 showcases the discernible outcomes gleaned from our Logistic Regression models applied across distinct MapReduce (MR) categories, shedding light on noteworthy features and performance metrics. The results underscore the significance of our approach in providing detailed insights into the predictive capabilities of each model configuration.

Significantly, the first MR category stands out for its exceptional performance, boasting a remarkable Best Cross-Validated F1 Score of 0.98 and a perfect Best F1 Score of 1 on the test set. This highlights the robustness of our model in accurately predicting outcomes within this category. Similarly, the second and third categories, leveraging different feature sets, exhibited high predictive accuracy, with F1 scores of 0.98 and 0.97, respectively.

Of particular interest is the consistent selection of the optimal regularization parameter (C) set to 1 across all categories, indicative of the stability and reliability of our Logistic Regression models amidst varying feature combinations. These findings underscore the effectiveness and versatility of our approach in delivering precise and reliable recommendations tailored to single-machine problems transitioning to distributed systems in the context of MapReduce.

The outcomes gleaned from the meticulous examination of Naive Bayes and Logistic Regression models across an array of MapReduce (MR) categories offer valuable insights into the effectiveness of employing diverse feature combinations. Notably, the hybrid approach, showcased in Table 8, which amalgamates the strengths of multiple models, proves particularly potent, especially when tackling tasks involving MapReduce categories with varying data types and patterns to consider. The process of discerning the best-performing models for each category and leveraging their respective strengths serves to enhance prediction accuracy and efficiency. This approach can be likened to assembling a toolbox filled with different techniques and selecting the most appropriate tool for each specific task at hand. Such adaptability enables the tailoring of solutions to the unique characteristics of each category, thereby optimizing overall performance. The consistently high F1 scores observed across different MR categories underscore the robustness of the models, a fact that is especially pronounced in the first category where Logistic Regression achieved a near-perfect score. The meticulous selection of optimal alpha values for Naive Bayes and regularization parameters (C) for Logistic Regression speaks to the careful consideration given to hyperparameters, ultimately contributing to the stability and reliability of the models.

Table 8 – Hybrid Approach Results

MR category	Algorithm	Features	Best Cross-Validated F1 Score	Best F1 Score on Test Set	Alpha / C
#1 MR category	Naive Bayes	Goal+Title+Category+Input+Output	0.99	1	0.01
#2 MR category	Logistic Regression	Goal+Title	0.98	1	1
#3 MR category	Naive Bayes	Goal+Title+Category+Input+Output+Keywords	0.99	1	0.001
#4 MR category	Logistic Regression	Goal+Title+Category+Input+Output+Keywords	0.97	1	1
#5 MR category	Logistic Regression	Goal+Title+Category+Input+Output+Keywords	0.99	1	1

Consequently, a meticulously crafted recommender system was developed, integrating five distinct models intended to accurately predict the assignment of five categorical labels to novel problem instances. Following the system's development, rigorous evaluation procedures were implemented to assess its predictive efficacy. Expert opinion was sought and thoroughly scrutinized to evaluate the system's performance in predicting the labels of newly encountered problems. The outcome of this evaluation revealed promising results, affirming the effectiveness of the recommender system in its predictive capabilities.

However, it is imperative to acknowledge certain limitations within the study. Despite efforts to expand the dataset, its relatively small size raises questions regarding its generalizability to a broader range of problem instances, warranting further investigation. Additionally, the study primarily focuses on binary classification, and extending the approach to address multi-class scenarios could be a potential avenue for future research. Nevertheless, the comprehensive evaluation and hybrid approach presented in this study make a valuable contribution to the field of recommender systems for single-machine problems transitioning to distributed systems in the context of MapReduce.

3.3 Recommender system as a web application

The starting page, shown in Figure 15, of the web application, titled "Algoanalyses," features a clear and concise directive aimed at guiding users through the problem-solving process. The page is designed to encourage users to analyze problems effectively, practice on similar problems, and ultimately take action by clicking the prominent "Let's Get Started" button.

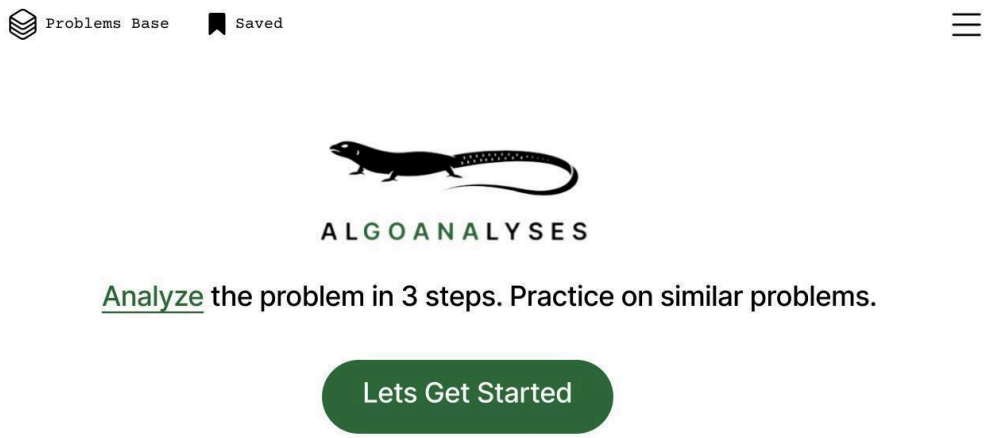

Figure 15 - Starting page of web application

At the top of the page, prominently displayed, is the title "Algoanalyses." This title succinctly communicates the purpose of the web application, suggesting that it is a tool for analyzing algorithms.

Immediately below the title, there is instructional text that provides a brief overview of the problem-solving approach encouraged by the application. The text is divided into three clear steps:

Analyze the Problem in Three Steps: This indicates that users should approach problem analysis systematically, breaking it down into manageable steps.

Practice on Similar Problems: Emphasizes the importance of practicing problem-solving skills by working on similar problems. This suggests that users can benefit from applying learned techniques to a variety of scenarios.

The focal point of the starting page is a prominent button labeled "Let's Get Started." This button serves as the primary call to action, inviting users to begin their problem-solving journey within the application.

The design of the starting page is likely clean and uncluttered, with attention drawn to the instructional text and the call to action button. The use of contrasting colors makes the button stand out and encourages user interaction.

Additionally, the starting page may include a simple navigation menu or links to other sections of the web application, providing users with easy access to additional features or resources.

Overall, the starting page of the web application sets the tone for an intuitive and user-friendly experience, guiding users through the problem-solving process and encouraging them to take action with the click of a button.

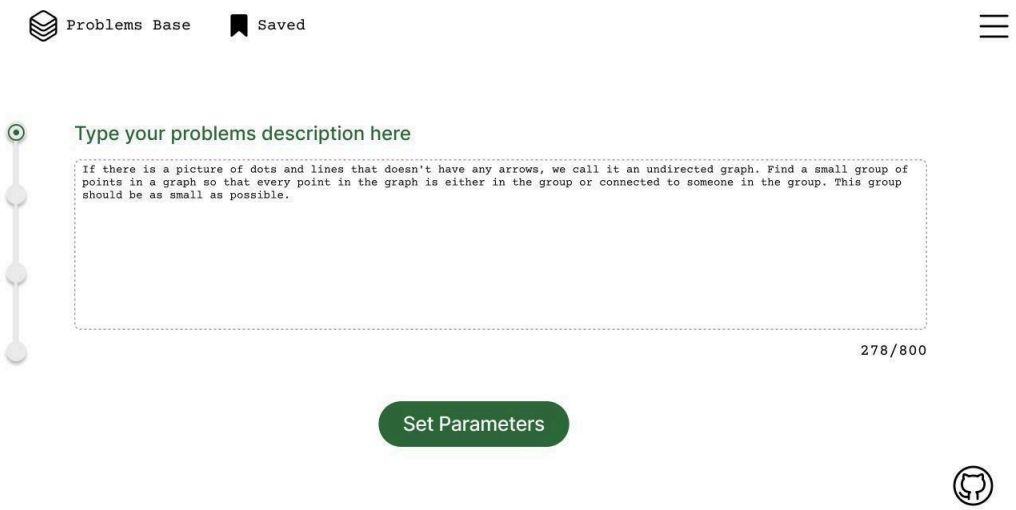

Figure 16 - Query Input Page: Users Enter Their Problem Description

The next page of the web application, showm in figure 16, is designed to facilitate inputting the user's problem description. At the top of the page, the text "Type your problem description here" is prominently displayed. This heading serves as a clear instruction to the user, indicating the purpose of the text field below.

Immediately below the heading, there is a large text field where users can input their problem description. The text field allows for a maximum length of 800 characters, ensuring that users can provide detailed descriptions while maintaining a manageable input size.

Beneath the text field, there is a button labeled "Set Parameters." This button enables users to proceed to the next step of the problem-solving process once they have entered their problem description. Clicking this button indicates that the user has completed inputting their problem description and is ready to move forward.

Located on the left side of the page, there is a progress bar that visually indicates the user's progression through the problem-solving process. This progress bar gives users a sense of context and helps them understand where they are in the overall process.

The page is designed with a clean and minimalist layout, with the text field and button positioned prominently for easy access. The progress bar features a visual indicator as color changes to denote progress.

Overall, this page serves as a user-friendly interface for users to input their problem descriptions and progress through the problem-solving process within the web application.

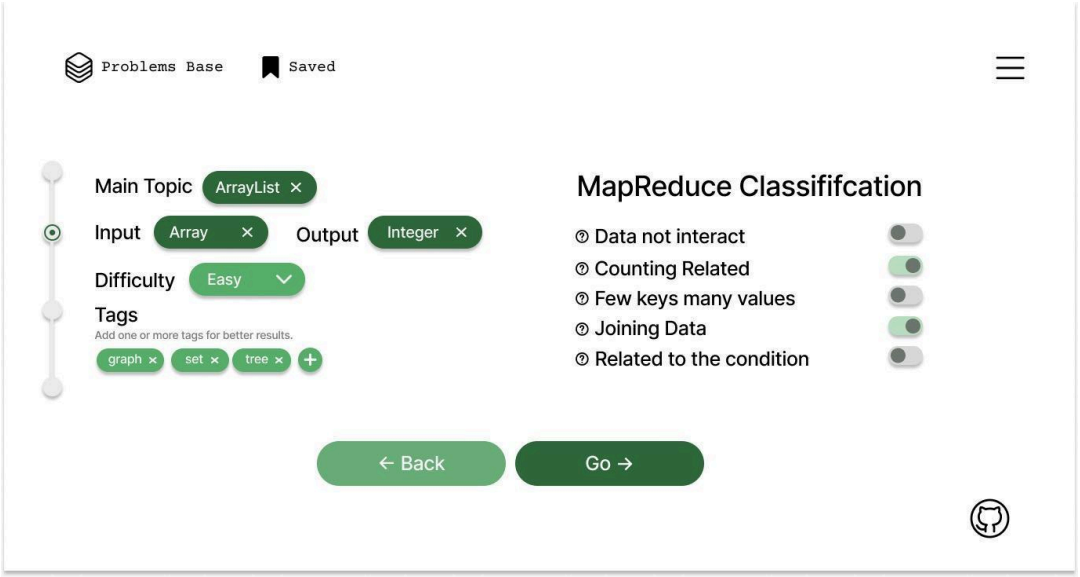

Figure 17 - Problem Characteristics and Categories Prediction Page

The next page of the web application, shown in figure 17, serves as an intermediary step where the predicted problem parameters and MapReduce classifications are displayed to the user, along with the option to adjust them if necessary. At the top of the page, the predicted problem parameters are displayed. These parameters include:

1. Main Topic: The main topic or category to which the problem belongs.
2. Input and Output Types: The types of input and output expected for the problem, such as integer or array.
3. Difficulty Level: An estimation of the problem's difficulty, categorized as easy, medium, or difficult.
4. Tags: Additional tags or keywords associated with the problem, such as graph, tree, or set.

Below the predicted problem parameters, the five MapReduce classifications are displayed. Each classification is accompanied by a switch button that allows the user to toggle its status. The classifications include:

1. Data Does Not Interact: Indicates whether the problem involves data that does not interact with each other.
2. Counting Related: Indicates if the problem is related to counting.
3. Few Keys Many Values: Specifies if the problem involves few known keys and many unknown values.
4. Joining Data: Indicates if the problem requires joining data and using the output of one MapReduce job in another.
5. Related to the Condition: Specifies if the problem is related to a condition.

Next to each predicted parameter and classification, there are options for the user to manually adjust them if needed. This ensures that the user has the flexibility to refine the predictions based on their own understanding of the problem.

At the bottom of the page, there are two buttons:

1. Back: Allows the user to go back to the previous page to make any changes or corrections.
2. Go: Enables the user to proceed to the next step of the problem-solving process once they are satisfied with the predicted parameters and classifications.

The page is likely designed with a clear and intuitive layout, with the predicted parameters and classifications presented in a structured format. Switch buttons may be designed to be easily identifiable and clickable, while navigation buttons are typically styled for clarity and accessibility.

Overall, this page serves as a crucial checkpoint for users to review and adjust the predicted parameters and classifications before moving forward with the problem-solving process in the web application.

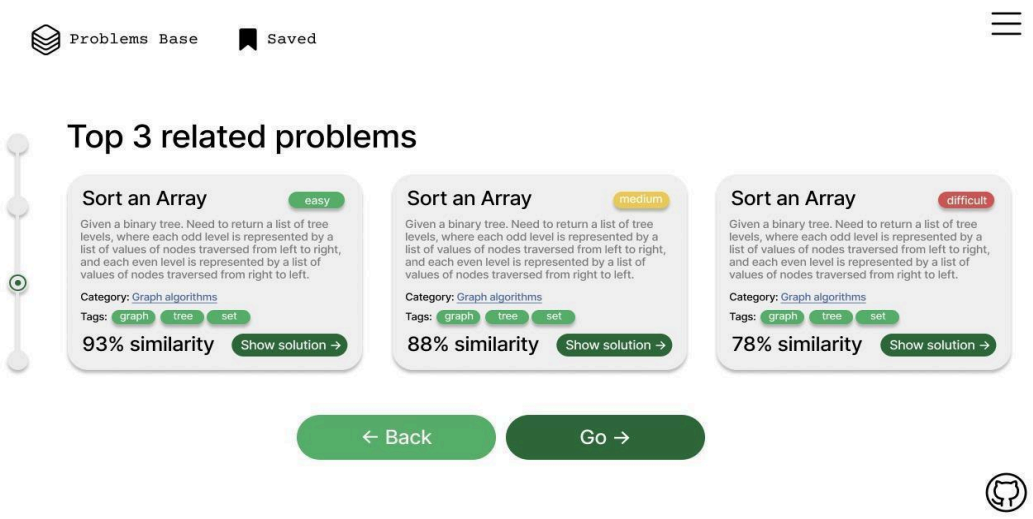

Figure 18 - Top Three Recommended Problems Page

The next page of the web application, shown in figure 18, presents the user with the top three related problems based on the input parameters provided earlier. At the top of the page, a text heading reads "Top 3 Related Problems." This heading indicates to the user that the following content will display three problems that are most closely related to the input parameters.

Below the heading, information about each related problem is presented separately. This information typically includes:

1. Title: The title of the problem.
2. Difficulty: The estimated difficulty level of the problem (e.g., easy, medium, difficult).
3. Description: A brief description or summary of the problem.
4. Category: The category or main topic to which the problem belongs.
5. Tags: Additional tags or keywords associated with the problem.
6. Similarity Percentage: The percentage indicating the similarity between the user's input problem and the related problem. This percentage gives the user an idea of how closely the related problem matches their input.

For each related problem displayed, there is an option for the user to choose it by clicking on it. Upon selection, the chosen related problem becomes the focus for further analysis or action.

A progress bar is included on the page to provide visual feedback on the user's progression through the problem-solving process. This progress bar may be similar to the one seen on previous pages, helping users understand where they are in the overall process.

At the bottom of the page, there are two buttons:

1. Back: Allows the user to go back to the previous page to make any changes or corrections to the input parameters.
2. Go: Enables the user to proceed to the next step of the problem-solving process after selecting a related problem. Clicking this button indicates the user's decision to move forward with the chosen problem.

Overall, this page serves as a pivotal point where users can review the top related problems and select one to focus on for further analysis or action within the web application.

The next page of the web application, shown in figure 19, provides detailed information about the chosen problem along with a code editor field displaying the solution to the problem.

Information of the Chosen Problem placed on the left side of the page:

1. Title: The title of the chosen problem.
2. Difficulty: The estimated difficulty level of the problem (e.g., easy, medium, difficult).
3. Description: A detailed description or summary of the problem.
4. Category: The category or main topic to which the problem belongs.
5. Tags: Additional tags or keywords associated with the problem.
6. Similarity Percentage: The percentage indicating the similarity between the user's input problem and the chosen problem.

The right side of the page contains a code editor field where the solution to the chosen problem is displayed. This allows users to view the solution code and potentially make modifications or annotations as needed.

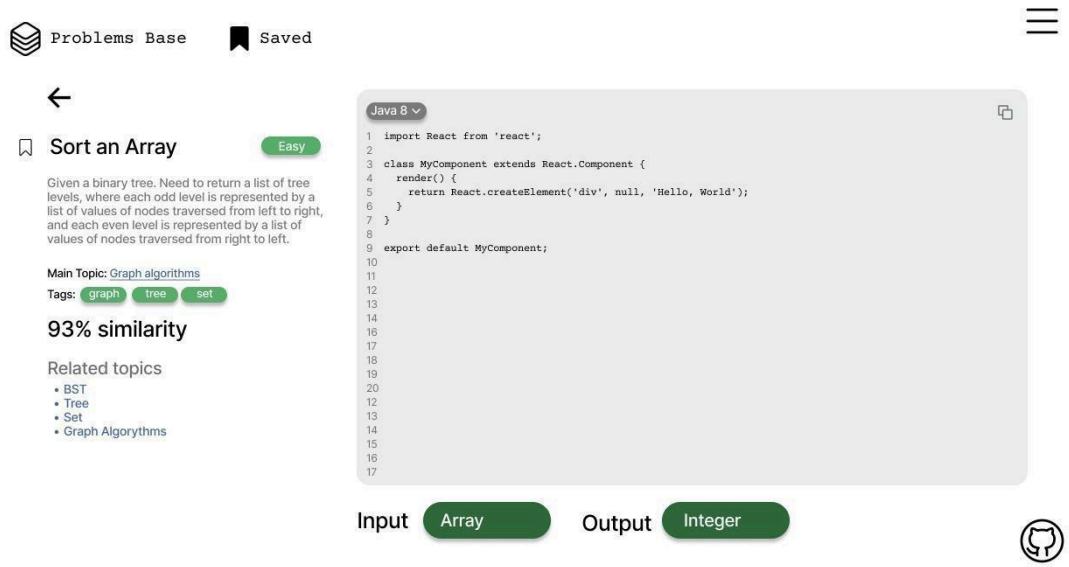

Figure 19 - Recommended Problem`s Solution Page

The code editor field may support syntax highlighting, line numbering, and other features commonly found in code editors to enhance readability and usability.

Positioned in the top left corner or above the information of the chosen problem, there is a back button represented as an arrow icon. This button allows the user to navigate back to the previous page if they wish to choose another problem or make changes to the parameters.

This page represents the final result of the problem-solving process within the web application. Users have access to both the detailed information about the chosen problem and its solution in the code editor field.

From this page, users can go back to the previous page to choose another problem or modify the parameters if they want to explore different options.

The page is designed with a clean and balanced layout, with the information of the chosen problem displayed prominently on the left side and the code editor field on the right side.

Visual cues such as borders or shading may be used to separate the two sections and provide visual clarity.

The back button is styled as an arrow icon to ensure ease of navigation.

Overall, this page provides users with a comprehensive view of the chosen problem and its solution, offering a user-friendly interface for exploring and understanding the problem-solving process within the web application.

CONCLUSION

In conclusion, this dissertation significantly contributes to the understanding of recommender systems in the context of transitioning single-machine problems to distributed systems within the MapReduce framework, aligning with the objective of developing an advisory system for recommending solutions.

The study concludes by highlighting substantial progress in teaching methodologies for MapReduce and Apache Spark, alongside the effectiveness of classification techniques within distributed systems. Through innovative approaches, the research aimed to deepen users' comprehension of these technologies while enhancing learning outcomes.

The exploration and creation of an innovative teaching methodology for MapReduce and Apache Spark's fundamental concepts, as a first objective of the research, have resulted in significant advancements in enhancing users' comprehension of these technologies.

Through the utilization of the ADDIE methodology over approximately eight weeks, a novel curriculum addressing the classification of single-machine tasks within distributed systems was developed. This curriculum underwent meticulous refinement through expert input, culminating in five distinct classes intricately juxtaposing single-machine solutions with distributed MapReduce solutions.

Course content, materials, and activities were designed and tasked with translating expert-derived insights into tangible educational assets. Regular refinement sessions overseen by an expert panel ensured alignment with exacting standards and coherence in achieving desired learning outcomes.

The second objective of assessing the efficacy of a classification technique in enhancing user learning outcomes regarding MapReduce and Apache Spark within distributed systems has been achieved through rigorous evaluation and analysis.

In this study, a hypothesis test comparing the average GPA between a control group and an experimental group was conducted, revealing no significant difference in mean GPAs. However, further analysis employing an independent sample t-test demonstrated that students exposed to the novel teaching approach significantly outperformed those taught using traditional methods in MapReduce exam scores.

These findings suggest that the proposed methodology has the potential to significantly improve student performance in MapReduce examinations compared to conventional teaching methodologies. Nevertheless, further research with larger sample sizes is recommended to validate and strengthen these initial conclusions.

The study successfully evaluated the effectiveness of the classification technique in enhancing user learning outcomes, showcasing the superiority of the novel approach in facilitating comprehension and performance in MapReduce and Apache Spark within distributed systems.

The third objective of collecting a diverse dataset representing problems encountered in real-world MapReduce applications from prominent books and scientific articles, covering fundamental concepts and advanced topics in distributed systems and big data processing, has been successfully achieved.

An exhaustive dataset comprising 107 unique problem instances was meticulously compiled, primarily originating from esteemed literary works such as "Hadoop in Action" and "MapReduce Design Patterns: Building Effective Algorithms and Analytics for Hadoop and Other Systems." These foundational problems covered a wide array of essential concepts and standard algorithms commonly utilized in MapReduce applications, ranging from fundamental tasks like word count and pirate talk to more intricate challenges involving the determination of minimum and maximum values.

To enrich the dataset with a broader spectrum of complexities and real-world scenarios, additional problem instances were extracted from scholarly articles. These supplementary problems delved into advanced topics such as sentiment analysis, k-means classification, decision trees, and the Apriori algorithm. By integrating problems sourced from authoritative books and scholarly literature, the dataset comprehensively represented the diverse range of problem instances encountered in practical MapReduce applications, providing a robust foundation for subsequent phases of the study.

The selection process for problems adhered to a dual-pronged approach: incorporating problems showcasing a diverse array of methodologies and approaches towards resolving distributed systems challenges, and including problems exhibiting similarities in solution strategies yet diverging in specific problem contexts. This deliberate strategy facilitated a holistic understanding of problem-solving paradigms within the MapReduce framework.

The meticulous selection of problems for dataset inclusion was led by domain experts possessing profound understanding of distributed systems operating under the MapReduce paradigm.

The development of a Recommendation System based on trained classification models to accurately forecast the assignment of categorical labels to novel problem instances has been successfully accomplished as a forth objective of research. Through extensive testing and optimization, both Naive Bayes and Logistic Regression models were employed, leveraging diverse textual features to enhance predictive accuracy significantly. The study meticulously explored various combinations of models and features, aiming to develop a tailored and optimized predictive framework for each column within the dataset.

The hybrid approach, which amalgamated the strengths of Naive Bayes and Logistic Regression models, proved particularly potent in enhancing prediction accuracy and efficiency across MapReduce categories with varying data types and patterns. The resulting recommender system demonstrated robust predictive capabilities, as evidenced by the F1 scores obtained from the evaluation process. For instance, the Naive Bayes models achieved high F1 scores ranging from 0.94 to 0.99 across different MapReduce categories, while the Logistic Regression models displayed similarly impressive F1 scores ranging from 0.97 to 0.99. Additionally, the hybrid approach exhibited consistent performance with F1 scores of 0.97 to 0.99 across various categories.

The Evaluation of Recommender System, as a fifth objective of the research, underwent rigorous evaluation procedures to assess its predictive efficacy, including

expert opinion analysis. The outcomes of this evaluation, supplemented by the high F1 scores obtained, affirmed the effectiveness of the recommender system in accurately predicting the labels of newly encountered problems. Despite certain limitations regarding dataset size and focus on binary classification, the comprehensive evaluation and hybrid approach presented in this study contribute significantly to the field of recommender systems for single-machine problems transitioning to distributed systems in the context of MapReduce. Further research endeavors may explore enlarging the dataset and extending the approach to handle multi-class scenarios, enhancing the system's versatility and applicability.

Future research endeavors may explore enlarging the dataset and extending the approach to handle multi-class scenarios, enhancing the system's versatility and applicability. Additionally, further investigations could focus on refining the recommender system's algorithms and methodologies to address evolving challenges in distributed systems and improve predictive accuracy.

In essence, the conclusion underscores the importance of meticulous methodology in driving advancements in MapReduce problem classification and recommender system development. It paves the way for scalable and refined methodologies, ultimately contributing to a broader understanding and practical application of these technologies in real-world scenarios.

REFERENCES

1 Karau H., Warren R. High performance Spark: best practices for scaling and optimizing Apache Spark. – Sebastopol: O'Reilly Media, Inc., 2017. – 280 p.

2 Chen Y., Lu J., Chen C. et al. Cost-effective resource provisioning for spark workloads // Proceed. of the 28th ACM internat. conf. on Information and Knowledge Management. – Beijing, 2019. – P. 2477-2480.

3 Shaikh E., Mohiuddin I., Alufaisan Y. et al. Apache spark: A big data processing engine // Proceed. of the 2019 2nd IEEE Middle East and North Africa COMMunications conf. (MENACOMM). – Manama, 2019. – P. 1-6.

4 Ortiz G., Caravaca J.A., García-de-Prado A. et al. Real-time context-aware microservice architecture for predictive analytics and smart decision-making // IEEE Access. – 2019. – Vol. 7. – P. 183177-183194.

5 Ali M., Iqbal K. The Role of Apache Hadoop and Spark in Revolutionizing Financial Data Management and Analysis: A Comparative Study // Journal of Artificial Intelligence and Machine Learning Management. – 2022. – Vol. 6, Issue 2. – P. 14-28.

6 Alotaibi S. et al. Sehaa: A big data analytics tool for healthcare symptoms and diseases detection using Twitter, Apache Spark, and Machine Learning // Applied Sciences. – 2020. – Vol. 10, Issue 4. – P. 1398-1-1398-29.

7 Suneetha V., Suresh S., Jhananie V. A novel framework using apache spark for privacy preservation of healthcare big data // Procced. 2nd internat. conf. on Innovative Mechanisms for Industry Applications (ICIMIA). – Bangalore, 2020. – P. 743-749.

8 Gosh S., Nahar N., Wahab M.A. et al. Recommendation system for e-commerce using alternating least squares (ALS) on apache spark // International Conference on Intelligent Computing & Optimization. – Cham: Springer, 2020. – P. 880-893.

9 Jha B.K., Sivasankari G., Venugopal K. Product recommendation system using scalable alternating least square algorithm and collaborative filtering using apache spark in e-commerce // Annals of the Romanian Society for Cell Biology. – 2021. – Vol. 25, Issue 4. – P. 2611-2622.

10 Podhoranyi M. A comprehensive social media data processing and analytics architecture by using big data platforms: a case study of twitter flood-risk messages // Earth Science Informatics. – 2021. – Vol. 14, Issue 2. – P. 913-929.

11 Ahmed N., Barczak A.L., Susnjak T. et al. A comprehensive performance analysis of Apache Hadoop and Apache Spark for large scale data sets using HiBench // Journal of Big Data. – 2020. – Vol. 7, Issue 1. – P. 110-1-110-18.

12 Ibtisum S., Bazgir E., Rahman S.A. et al. A comparative analysis of big data processing paradigms: Mapreduce vs. apache spark // World Journal of Advanced Research and Reviews. – 2023. – Vol. 20, Issue 1. – P. 1089-1098.

13 Идрышева С.К. О Цифровом кодексе Казахстана // https://online.zakon.kz/Document. 10.11.2023.

14 Li R., Hu H., Li H. et al. MapReduce parallel programming model: a state-of-the-art survey // International Journal of Parallel Programming. – 2016. –

Vol. 44. – P. 832-866.

15 Dean J., Ghemawat S. MapReduce: a flexible data processing tool // Communications of the ACM. – 2010. – Vol. 53, Issue 1. – P. 72-77.

16 Maitrey S., Jha C. MapReduce: simplified data analysis of big data // Procedia Computer Science. – 2015. – Vol. 57. – P. 563-571.

17 Hashem I.A.T., Anuar N.B., Gani A. et al. MapReduce: Review and open challenges // Scientometrics. – 2016. – Vol. 109. – P. 389-422.

18 Paul A.K., Zhuang W., Xu L. et al. Chopper: Optimizing data partitioning for in-memory data analytics frameworks // Procced. IEEE internat. conf. on Cluster Computing (CLUSTER). – Taipei, 2016. – P. 110-119.

19 Painuly S., Sharma S., Matta P. Big Data Driven E-Commerce Application Management System // Procced. 6th internat. conf. on Communication and Electronics Systems (ICCES). – Coimbatre, 2021. – P. 1-5.

20 Kumar Y., Sood K., Kaul S., Vasuja R. Big data analytics and its benefits in healthcare // In book: Big Data Analytics in Healthcare. – Cham, 2020. – P. 3-21.

21 Liang Y., Quan D., Wang F. et al. Financial big data analysis and early warning platform: a case study // IEEE Access. – 2020. – Vol. 8. – P. 36515-36526.

22 Yan P. Mapreduce and semantics enabled event detection using social media // Journal of Artificial Intelligence and Soft Computing Research. – 2017. – Vol. 7, Issue 3. – P. 201-213.

23 Akanbi A. et al. A distributed stream processing middleware framework for real-time analysis of heterogeneous data on big data platform: Case of environmental monitoring // Sensors. – 2020. – Vol. 20, Issue 11. – P. 3166-1-3166-25.

24 Veiga J., Expósito R.R., Pardo X.C. et al. Performance evaluation of big data frameworks for large-scale data analytics // Procced. IEEE internat. conf. on Big Data (Big Data). – Washington, 2016. – P. 424-431.

25 Weets J.-F., Kakhani M.K., Kumar A. Limitations and challenges of HDFS and MapReduce // Procced. internat. conf. on Green Computing and Internet of Things (ICGCIoT). – Greater Noida, 2015. – P. 545-549.

26 Vijayalakshmi V., Akila A., Nagadivya S. The survey on MapReduce // International Journal of Engineering Science and Technology. – 2012. – Vol. 4, Issue 07. – P. 143-151.

27 Lee K.-H. et al. Parallel data processing with MapReduce: a survey // ACM SIGMOD Record. – 2012. – Vol. 40, Issue 4. – P. 11-20.

28 Madani Y., Erritali M., Bengourram J. Sentiment analysis using semantic similarity and Hadoop MapReduce // Knowledge and Information Systems. – 2019. – Vol. 59. – P. 413-436.

29 Ha I., Back B., Ahn B. MapReduce functions to analyze sentiment information from social big data // International Journal of Distributed Sensor Networks. – 2015. – Vol. 11, Issue 6. – P. 417502.

30 Nasir N., Zafar K., Alamgir Z. Sentiment Analysis of Social Media Using MapReduce // https://scholar.google.ru/citations?view_op=view. 10.11.2023.

31 Nodarakis N., Sioutas S., Tsakalidis A.K. et al. MR-SAT: a MapReduce algorithm for big data sentiment analysis on Twitter // Procced. internat. conf. on Web Information Systems and Technologies (SCITEPRESS). – Rome, 2016. – P. 140-147.

32 Cui X., Zhu P., Yang X. et al. Optimized big data K-means clustering using MapReduce // Journal of Supercomputing. – 2014. – Vol. 70. – P. 1249-1259.

33 Anchalia P.P., Koundinya A.K., Srinath N. MapReduce design of K-means clustering algorithm // Procced. internat. conf. on Information Science and Applications (ICISA). – Pattaya, 2013. – P. 1-5.

34 Gopalani S., Arora R. Comparing apache spark and map reduce with performance analysis using k-means // International Journal of Computer Applications. – 2015. – Vol. 113, Issue 1. – P. 8-11.

35 Sardar T.H., Ansari Z. An analysis of distributed document clustering using MapReduce based K-means algorithm // Journal of the Institution of Engineers (India): Series B. – 2020. – Vol. 101, Issue 6. – P. 641-650.

36 Mao Y., Gan D. et al. A MapReduce-based K-means clustering algorithm // Journal of Supercomputing. – 2022. – Vol. 78, Issue 4. – P. 1-22.

37 Dai W., Ji W. A mapreduce implementation of C4. 5 decision tree algorithm // International Journal of Database Theory and Application. – 2014. – Vol. 7, Issue 1. – P. 49-60.

38 Koli A., Shinde S. Parallel decision tree with map reduce model for big data analytics // Procced. internat. conf. on Trends in Electronics and Informatics (ICEI). – Tirunelveli, 2017. – P. 735-739.

39 Es-sabery F., Hair A. A MapReduce C4. 5 decision tree algorithm based on fuzzy rule-based system // Fuzzy Information and Engineering. – 2019. – Vol. 11, Issue 4. – P. 446-473.

40 Es-Sabery F. et al. A MapReduce opinion mining for COVID-19-related tweets classification using enhanced ID3 decision tree classifier // IEEE Access. – 2021. – Vol. 9. – P. 58706-58724.

41 Mu Y., Liu X., Wang L. et al. A parallel fuzzy rule-base based decision tree in the framework of map-reduce // Pattern Recognit. – 2020. – Vol. 103. – P. 107326.

42 Sornalakshmi M. et al. An efficient apriori algorithm for frequent pattern mining using mapreduce in healthcare data // Bull. Electr. Eng. Inform. – 2021. – Vol. 10, Issue 1. – P. 390-403.

43 Yange T.S., Gambo I.P., Ikono R. et al. A multi-nodal implementation of apriori algorithm for big data analytics using MapReduce framework // Int. J. Appl. Inf. Syst. – 2020. – Vol. 12, Issue 31. – P. 8-28.

44 Verma N. et al. Big data analytics for retail industry using MapReduce-Apriori framework // J. Manag. Anal. – 2020. – Vol. 7, Issue 3. – P. 424-442.

45 Sharma A., Tripathi K. Hybrid version of apriori using mapreduce // Mobile Radio Communications and 5G Networks: proceed. of (MRCN 2020). – Cham: Springer, 2021. – P. 585-592.

46 Wang H.-B., Gao Y.-J. Research on parallelization of Apriori algorithm in association rule mining // Procedia Comput. Sci. – 2021. – Vol. 183. – P. 641-647.

47 Sundarakumar M. et al. Improving data processing speed on large datasets in a hadoop multinode cluster using enhanced apriori algorithm // J. Intell. Fuzzy Syst. – 2023. – Vol. 45, Issue 1. – P. 1-17.

48 Kariboz D., Bogdanchikov A., Orynbekova K. Computing feature vectors

of students for face recognition using Apache Spark // Procced. 15th internat. conf. on Electronics, Computer and Computation (ICECCO). – Abuja, 2019. – P. 1-3.

49 Meraliyev M., Orynbekova K., Talasbek A. et al. Optimization of data segments and number of cores for defining popularity of kazakh words using apache spark // Eng. J. Satbayev Univ. – 2021. – Vol. 143, Issue 3. – P. 39-42.

50 Orynbekova K., Talasbek A., Omar A. et al. MBTI personality classification using Apache Spark // Procced. 16th internat. conf. on Electronics Computer and Computation (ICECCO). – Kaskelen, 2021. – P. 1-4.

51 Serek A., Orynbekova K., Talasbek A. et al. Recommendation System for Human Resource Management by the Use of Apache Spark Cluster // Procced. 17th internat. conf. on Electronics Computer and Computation (ICECCO). – Kaskelen, 2023. – P. 1-4.

52 Ayazbayev D., Bogdanchikov A., Orynbekova K. et al. Defining Semantically Close Words of Kazakh Language with Distributed System Apache Spark // Big Data Cogn. Comput. – 2023. – Vol. 7, Issue 4. – P. 160-1-160-13.

53 Liling L. Summary of recommendation system development // Journal of Physics: Conference Series. – 2019. – Vol. 1187, Issue 5. – P. 052044.

54 Benkessirat S., Boustia N., Rezoug N. Overview of recommendation systems // In book: Smart Education and e-Learning. – Cham: Springer, 2019. – P. 357-372.

55 Avila J., Riofrlo X., Palacio-Baus K. et al. Decategorizing demographically stereotyped users in a semantic recommender system // Procced. 42 Latin American Computing conf. (CLEI). – Valparaiso, 2016. – P. 1-7.

56 Deldjoo Y. et al. A review of modern fashion recommender systems // ACM Comput. Surv. – 2023. – Vol. 56, Issue 4. – P. 1-37.

57 Guo Q. et al. A survey on knowledge graph-based recommender systems // IEEE Trans. Knowl. Data Eng. – 2020. – Vol. 34, Issue 8. – P. 3549-3568.

58 Pande C., Witschel H.F., Martin A. New hybrid techniques for business recommender systems // Appl. Sci. – 2022. – Vol. 12, Issue 10. – P. 4804-1-4804-17.

59 Teja Santosh D., Kakulapati V., Basavaraju K. Ontology-based sentimental knowledge in predicting the product recommendations: A data science approach // J. Discrete Math. Sci. Cryptogr. – 2020. – Vol. 23, Issue 1. – P. 1-18.

60 Tarus J.K., Niu Z., Mustafa G. Knowledge-based recommendation: a review of ontology-based recommender systems for e-learning // Artif. Intell. Rev. – 2018. – Vol. 50. – P. 21-48.

61 Khurana D., Koli A., Khatter K. et al. Natural language processing: State of the art, current trends and challenges // Multimed. Tools Appl. – 2023. – Vol. 82, Issue 3. – P. 3713-3744.

62 Bose S., Choudhary C., Behra A. et al.. An Innovative Recommender System for E-Commerce Websites using Natural Language Processing // International Journal of Recent Technology and Engineering (IJRTE). – 2020. – Vol. 8, Issue 6. – P. 4085-4089.

63 Tarnowska K.A., Ras Z. NLP-based customer loyalty improvement recommender system (CLIRS2) // Big Data Cogn. Comput. – 2021. – Vol. 5, Issue 1. – P. 4-1-4-16.

64 Zeng Z. et al. Knowledge transfer via pre-training for recommendation: A review and prospect // Front. Big Data. – 2021. – Vol. 4. – P. 602071.

65 Wu L. et al. A survey on large language models for recommendation // https://arxiv.org/abs/2305.19860. 11.11.2023.

66 Guo W. et al. Deep natural language processing for search and recommender systems // Proceed. of the 25th ACM SIGKDD internat. conf. on Knowledge Discovery & Data Mining. – NY., 2019. – P. 3199-3200.

67 Khanal S.S., Prasad P., Alsadoon A. et al. A systematic review: machine learning based recommendation systems for e-learning // Educ. Inf. Technol. – 2020. Vol. 25, Issue 4. – P. 2635-2664.

68 Ko H., Lee S., Park Y. et al. A survey of recommendation systems: recommendation models, techniques, and application fields // Electronics. – 2022. – Vol. 11, Issue 1. – P. 141-1-141-48.

69 Gao C., Zheng Y., Wang W. et al. Causal inference in recommender systems: A survey and future directions // ACM Trans. Inf. Syst. – 2024. – Vol. 42, Issue 4. – P. 1-32.

70 Natarajan S., Vairavasundaram S., Natarajan S. et al. Resolving data sparsity and cold start problem in collaborative filtering recommender system using linked open data // Expert Syst. Appl. – 2020. – Vol. 149. – P. 113248.

71 Da'u A., Salim N. Recommendation system based on deep learning methods: a systematic review and new directions // Artif. Intell. Rev. – 2020. – Vol. 53, Issue 4. – P. 2709-2748.

72 Burke R. Knowledge-based recommender systems // Encycl. Libr. Inf. Syst. – 2000. – Vol. 69, Suppl. 32. – P. 175-186.

73 Tarus J.K., Niu Z., Mustafa G. Knowledge-based recommendation: a review of ontology-based recommender systems for e-learning // Artif. Intell. Rev. - 2018. – Vol. 50. – P. 21-48.

74 Rosa R.L., Schwartz G.M., Ruggiero W.V. et al. A knowledge-based recommendation system that includes sentiment analysis and deep learning // IEEE Trans. Ind. Inform. – 2018. – Vol. 15, Issue 4. – P. 2124-2135.

75 Prasad B. A knowledge-based product recommendation system for e-commerce // Int. J. Intell. Inf. Database Syst. – 2007. – Vol. 1, Issue 1. – P. 18-36.

76 El Bouhissi H., Adel M., Ketam A. et al. Towards an Efficient Knowledge-based Recommendation System. // IntelITSIS'2021: procced. 2nd internat. Workshop on Intelligent Information Technologies and Systems of Information Security. – 2021, Khmelnytskyipp, 2021. – P. 38-49.

77 Kanwal S., Nawaz S., Malik M.K. et al. A review of text-based recommendation systems // IEEE Access. – 2021. – Vol. 9. – P. 31638-31661.

78 Miao D., Lang F. A recommendation system based on text mining // Procced. internat. conf. on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC). – Nanjing, 2017. – P. 318-321.

79 Khatter H., Arif S., Singh U., Mathur S., Jain S. Product recommendation system for E-commerce using collaborative filtering and textual clustering // Procced. 3rd internat. conf. on Inventive Research in Computing Applications (ICIRCA). – Coimbatore, 2021. – P. 612-618.

80 Roul R.K., Arora K. A nifty review to text summarization-based recommendation system for electronic products // Soft Comput. – 2019. – Vol. 23. – P. 13183-13204.

81 Sharma Y., Bhatt J., Magon R. A multi-criteria review-based hotel recommendation system // Procced. IEEE internat. conf. on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing. – Liverpool, 2015. – P. 687-691.

82 Kavinkumar V., Reddy R.R., Balasubramanian R. et al. A hybrid approach for recommendation system with added feedback component // 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI). – Kochi, 2015. – P. 745-752.

83 Lucas J.P., Luz N., Moreno M.N. et al. A hybrid recommendation approach for a tourism system // Expert Syst. Appl. – 2013. – Vol. 40, Issue 9. – P. 3532-3550.

84 Tian Y., Zheng B., Wang Y. et al. College library personalized recommendation system based on hybrid recommendation algorithm // Procedia Cirp. – 2019. – Vol. 83. – P. 490-494.

85 Passi R., Jain S., Singh P.K. Hybrid approach for recommendation system // Proceed. of the 2nd internat. conf. on Data Engineering and Communication Technology (ICDECT 2017). – Cham: Springer, 2019. – P. 117-128.

86 Thorat P.B., Goudar R.M., Barve S. Survey on collaborative filtering, content-based filtering and hybrid recommendation system // Int. J. Comput. Appl. – 2015. – Vol. 110, Issue 4. – P. 31-36.

87 Walek B., Fojtik V. A hybrid recommender system for recommending relevant movies using an expert system // Expert Syst. Appl. – 2020. – Vol. 158. – P. 113452.

88 Al Fararni K., Nafis F., Aghoutane B. et al. Hybrid recommender system for tourism based on big data and AI: A conceptual framework // Big Data Min. Anal. – 2021. – Vol. 4, Issue 1. – P. 47-55.

89 Tawfik, A.A., Alhoori, H., Keene, C.W. et al. Using a Recommendation System to Support Problem Solving and Case-Based Reasoning Retrieval // Tech Know Learn – 2018. – Vol. 23. – P. 177-187.

90 Reiser R.A., Dempsey J.V. Trends and issues in instructional design and technology. – Boston: Pearson, 2012. – 397 p.

91 Miner D., Shook A. MapReduce design patterns: Building effective algorithms and analytics for Hadoop and other systems. – Sebastopol: O'Reilly Media, Inc., 2012. – 250 p.

92 Lam C. Hadoop in action. – NY.: Simon and Schuster, 2010. – 336 p.

93 Azizi Y., Azizi M., Elboukari M. Log files analysis using MapReduce to improve security // Procedia Comput. Sci. – 2019. – Vol. 148. – P. 37-44.

94 Gupta P., Kumar P., Gopal G. Sentiment analysis on hadoop with hadoop streaming // Int. J. Comput. Appl. – 2015. – Vol. 121, Issue 11. – P. 4-8.

95 Sardar T.H., Ansari Z. Partition based clustering of large datasets using MapReduce framework: An analysis of recent themes and directions // Future Comput. Inform. J. – 2018. – Vol. 3, Issue 2. – P. 247-261.

96 Dai W., Ji W. A mapreduce implementation of C4. 5 decision tree algorithm // Int. J. Database Theory Appl. – 2014. – Vol. 7, Issue 1. – P. 49-60.

97 Choi S.-Y., Chung K. Knowledge process of health big data using MapReduce-based associative mining // Pers. Ubiquitous Comput. – 2020. – Vol. 24. – P. 571-581.

98 Orynbekova K., Bogdanchikov A., Cankurt S. et al. MapReduce Solutions Classification by Their Implementation // Int. J. Eng. Pedagogy. – 2023. – Vol. 13, Issue 5. – P. 58-71.

APPENDIX A

Act of implementation

Қосымша 1

«SDU University»
мекемесі

Учреждение
«SDU University»

SDU
1996
UNIVERSITY

040900, Алматы облысы, Қарасай ауданы,
Қаскелең қаласы, Абылай хан көшесі 1/1
тел.:+7727 307-95-60, факс: 307-95-58

040900, Алматинская область, Карасайский
район, г. Каскелен қаласы, ул. Абылай хана
1/1 тел.:+7727 307-95-60, факс: 307-95-58

БЕКІТЕМІН
Оқу істер жөніндегі проректор
Богданчиков А.В.
«16» сәуір 2024 ж.

Оқу-әдістемелік және ғылыми жұмыс нәтижелерін оқу процессіне енгізу туралы АКТ

1. **Жұмыс енгізілетін мекеме атауы:** «SDU University» мекемесі
2. **Ұсыныс атауы:** Curriculum Development on Task-Based Classification
3. **Оқу-әдістемелік құрал және ғылыми жұмыс нәтижесін оқу процессіне енгізу формасы:** Developing educational tutorial based on the content of the curriculum and problem-solving recommender system
4. **Енгізу аясы:** Distributed Big Data filed in Education and Industry
5. **Апробация мерзімі:** 1 academic semester
6. **Апробация нәтижелері:** Task-based classification and instructional design led to better student performance compared to the traditional approach.
7. **Енгізуге жауапты:** Orynbekova Kamila
8. **Енгізу тиімділігі:** Student Performance, Resource Utilization
9. **Енгізудің қажеттілігі:** high

Факультет деканы: **Ахмедов Р.**

Кафедра меңгерушісі: **Мукаш Ж.**

Оқу-әдістемелік Кеңесінің № 9 **, «** 16 **»** 04 **2024 ж. хаттамасымен**

APPENDIX B

Collected data: 107 problems

ID	Title	Keywords	Category	Input	Output	Goal	MR#1	MR#2	MR#3	MR#4	MR#5
1	Sort an Array	sorting, selectior	Sorting algorithm	array of integers	sorted array of ir	Sort the given ar	1	0	0	0	0
2	Longest Commo	dynamic prograr	Dynamic prograi	two strings	length of the lon	Find the length o	1	1	1	0	0
3	Binary Tree Leve	tree, BFS	Graph algorithm	list of vertices, a	breadth first trav	You need to trav	1	0	1	0	0
4	Substring Searc	string, substring	Search algorithm	string	index of the first	Given two string	0	1	1	1	1
5	Maximum Subar	dynamic prograr	Array algorithms	array of element	maximum sum o	Given an array o	0	0	1	1	0
6	Dijkstra's Algorit	graph	Graph algorithm	graph with weigh	shortest paths	The problem is t	0	0	1	0	1
7	Merge Sort	sorting, recursio	Sorting algorithm	array of element	sorted array of e	The Merge Sort	0	0	1	1	1
8	Binary Search	search, array	Array algorithms	array of integers	index of the elen	There is an array	0	0	1	0	1
9	Set Cover	set	Optimization alg	subsets	minimum numbe	Given set [X] and	0	0	1	1	1
10	Dominating Set	graph, set	Graph algorithm	undirected graph	subset of vertice	Given an undirec	0	0	1	1	1
11	Bin Packing	greedy algorithm	Dynamic Progra	list of item sizes	number of conta	Given [n] objects	0	0	1	0	1
12	Longest Increas	array, dynamic p	Dynamic Progra	array of integers	longest increasir	There is an array	1	0	0	0	1
...	...	...	...	...	...	...	...	...	...	...	...
104	Word count	string, hash tabl	Array algorithms	string	list of words and	Count the numbe	0	1	0	0	0
105	Predict the Winn	dynamic prograr	Dynamic prograi	array of integers	boolean(if the ga	Given an array o	1	0	0	0	1
106	K-means cluster	ML, clustering	Machine Learnin	set of points in th	k clusters, each	Partition the set	0	0	0	1	1
107	Decision Tree	ML, classificatio	Machine Learnin	test dataset	prediction for tes	Create a decisio	0	0	1	1	1