Preprints.org

Article

Not peer-reviewed version

Efficient Big Data Processing and Recommendation System Development with Apache Spark

Shanqi Zhan * and Yujuan Qiu

Posted Date: 13 June 2025

doi: 10.20944/preprints202506.1150.v1

Keywords: Big Data; Apache Spark; Recommendation Systems; Data Preprocessing; MovieLens Dataset

Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 June 2025 doi:10.20944/preprints202506.1150.v1

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Efficient Big Data Processing and Recommendation System Development with Apache Spark

Shanqi Zhan * and Yujuan Qiu

[1] Municipal Parking Services, Inc. (MPS) Minnetonka, Minnesota, USA

[2] The George Washington University, Washington, D.C., USA

* Correspondence: Zhans.scholar@gmail.com

Abstract: The rapid development of big data analytics has revolutionized data analysis and decision-making processes across industries. This paper explores how to use Apache Spark to analyze the MovieLens 20M dataset and identify the top movies in Minnesota. By integrating robust data preprocessing and collaborative filtering techniques, a novel recommendation system is developed. The results reveal the popular movies in Minnesota, major genres such as drama and comedy, and related tags such as "original" and "finale." Additionally, a detailed tag correlation analysis is conducted to optimize recommendation accuracy. The study further illustrates Spark's application in large-scale data processing, demonstrating its effectiveness in recommendation systems. These findings bridge the gap between theoretical frameworks and practical applications, providing a replicable approach to address challenges in preprocessing, analysis, and personalized recommendations.

Keywords: big data; Apache Spark; recommendation systems; data preprocessing; MovieLens dataset

I. Introduction

Big data is playing an increasingly transformative role in modern analytics, revolutionizing industries by enabling deeper insights and more informed decision-making [1]. One of the most impactful areas ofbig data analytics is in entertainment and recommendation systems, particularly in generating localized recommendations, such as identifying popular movies in specific regions (e.g., Minnesota). To effectively process and analyze large-scale datasets, frameworks like Apache Spark have become essential due to their scalability, performance, and flexibility. Apache Spark is an open-source distributed computing framework designed for big data processing and analytics. It provides superior speed via in-memory processing and accommodates various workloads, such as batch processing, interactive queries, and machine learning.

However, working with large datasets presents several significant challenges, primarily related to data quality, scalability, and the development of efficient algorithms. Data quality issues often stem from semantic heterogeneity and the prevalence of unstructured data, which complicate integration and analysis efforts [2,3]. For instance, inconsistent formatting or missing information can reduce the reliability of results. Scalability also remains a critical concern, as traditional data processing tools struggle to manage the vast volumes and velocities of big data, necessitating advanced frameworks like Hadoop and Spark. These frameworks enable distributed computing but require substantial infrastructure and expertise to implement effectively [2,4]. Furthermore, extracting meaningful insights from complex datasets necessitates the development of efficient algorithms, which often rely on high-quality training data to achieve optimal results [2,5]. While big data analytics holds tremendous potential, addressing these challenges is essential to fully unlock its benefits [4,6].

Apache Spark offers significant advantages in analyzing and processing large datasets. Its architecture, based on the resilient distributed dataset (RDD) programming model, supports efficient

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 June 2025 doi:10.20944/preprints202506.1150.v1

data partitioning and locality-aware task placement, optimizing resource utilization and minimizing execution times [7]. Spark's ability to process structured, semi-structured, and unstructured data through DataFrames enhances its performance compared to traditional RDDs, making it a preferred framework for diverse applications, including recommendation systems [8]. Additionally, Spark's integration with artificial intelligence techniques enables robust batch processing, supporting the analysis of massive datasets in fields such as medical and entertainment analytics [9,10]. Despite its advantages, performance tuning and optimization remain critical for fully leveraging Spark's capabilities in the rapidly evolving landscape of big data [11].

This paper addresses a pressing challenge in modern data science: analyzing and extracting actionable insights from large- scale datasets. It contributes to the fields of big data analytics, recommendation systems, and regional data analysis by demonstrating how scalable frameworks like Apache Spark can effectively handle real-world challenges. Specifically, the paper bridges the gap between theoretical big data concepts and practical applications. Using the MovieLens 20M dataset, the research demonstrates an efficient, novel methodology for identifying the top-ranked movies. The results highlight the effectiveness of the proposed approach in addressing key challenges such as recommendation systems, data preprocessing, large-scale analysis, and user preference modeling. This work provides a replicable blueprint for building recommendation systems, making it valuable for both researchers in big data and practitioners seeking scalable solutions for regional data analysis and personalized recommendations.

The remainder of this paper is organized as follows: Section 2 details the proposed methodology, including the data preprocessing steps and model design. Section 3 presents the experimental results. Finally, Section 4 offers a summary of conclusions and potential avenues for future research.

II. Methodology

A. Dataset

The dataset used in this project was sourced from the GroupLens website, a research initiative from the Department of Computer Science and Engineering at the University of Minnesota. GroupLens provides a variety of datasets, including MovieLens, HetRec2011, WikiLens, Book-Crossing, Jester, and EachMovie. For this research, the MovieLens 20M dataset was selected because it is recommended for new research applications.

The MovieLens 20M dataset contains six CSV files and one text file, summarizing 27,278 movies and 138,493 randomly selected users. In terms of the dataset structure, the first file, genome-scores.csv, consists of three columns: movieId, tagId, and relevance. The relevance column represents how strongly a specific tag matches a particular movie, with values ranging from 0 to 1. The second file, genome-tags.csv, contains two columns: tagId and tag. This file provides user-generated labels for movies, which are often descriptive words or phrases. The third file, links.csv, contains movieId, imdbId, and tmdbId, which allow cross-referencing of movies with external databases such as IMDb and TMDb.

The fourth file, movies.csv, contains three columns: movieId, title, and genres. The title includes the movie name along with its release year in parentheses, and the genres column lists the categories associated with each movie in a pipe-separated format. The fifth file, ratings.csv, has four columns: userId, movieId, rating, and timestamp. The rating ranges from 0.5 to 5.0, and each user has rated at least 20 movies. Finally, the sixth file, tags.csv, records user-generated tags along with timestamps. A text file named README provides a detailed description of the dataset and its structure.

B. Data Preprocessing

Data preprocessing was a crucial step in this research to ensure the integrity and usability of the dataset [12]. One significant issue encountered was with movie titles containing commas, which disrupted the parsing of the movies.csv file. For example, the title "Godfather, The (1972)" caused

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 June 2025 doi:10.20944/preprints202506.1150.v1

the genre column to be misread as part of the title. To address this, the preprocessing included validating data formats and correcting misaligned fields. Specifically, the input file was filtered to only include rows where all columns satisfied specific data type requirements. For instance, the project ensured that the movie ID consisted of numeric characters and that ratings adhered to a valid floating-point format.

A test case was designed to identify such issues by filtering records for specific movie IDs, such as movieId 858, and checking whether the genre column displayed the correct value. In the case of movieId 858, the expected genre, "Crime|Drama," was initially replaced by part of the title. By addressing this misalignment, the preprocessing step ensured that subsequent analyses could be conducted without errors. Other than this issue, the dataset was found to be clean and ready for further processing.

C. Spark Algorithm

The analysis relied on Apache Spark for its distributed computing capabilities [13], which facilitated efficient processing of the large dataset. First, the ratings.csv, movies.csv, and genome-scores.csv files were read into Spark as RDDs. Each file was parsed by splitting rows into arrays based on commas. A case class was defined for each dataset to provide structure, allowing elements to be assigned appropriate names and data types such as strings or doubles. The RDDs were then converted into DataFrames for ease of manipulation, and their schemas were printed to verify the data structure.

Second, headers were removed from each dataset by applying filters to exclude rows that did not conform to expected patterns. This step ensured that all subsequent computations were based on valid data. For instance, the ratings dataset was filtered to retain only rows where the movieId and rating columns matched numeric patterns.

Third, key transformations were applied to derive meaningful insights. The ratings dataset was used to calculate statistics such as the count, minimum, maximum, mean, and standard deviation of ratings. Additionally, genres and tags were joined with ratings data to provide a comprehensive view of each movie's characteristics. For example, the project performed an intersection of movies with five-star ratings and movies with high tag relevance to identify the top-ranked films.

Finally, the top 10 movies were extracted by ranking movies based on their ratings and relevance. These results were saved to an HDFS output directory for further verification. To optimize performance and scalability, Spark's configuration settings were adjusted, including increasing the number of executors and memory allocation. This ensured that the system could handle larger datasets with minimal degradation in performance.

D. Performance Evaluation

The methodology also included an evaluation of the algorithm's scalability. The dataset, which was approximately 150 MB in size, required four minutes to process under the initial configuration. To accommodate larger datasets, the project proposed increasing the number of executors to 30 and allocating 1 GB of memory per executor. This adjustment significantly improved processing speed, demonstrating the robustness of the Spark framework for handling big data tasks.

III. Experimental Results

A. Statistics of the Dataset

After applying data preprocessing techniques, the dataset was analyzed to extract key characteristics, uncover trends, and gain insights into its structure. This section highlights the distribution of genres, user rating patterns, and tag relevance, offering a comprehensive view of the dataset.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 June 2025 doi:10.20944/preprints202506.1150.v1

Tag correlation analysis provides crucial insights into user preferences and the latent attributes of movies. The most relevant tags, including "original," "mentor," and "great ending," indicate strong user engagement and specific expectations for movies. By leveraging these insights, recommendation algorithms can refine movie suggestions by prioritizing films with strong correlations to high-rated tags. Incorporating tag correlation into collaborative filtering models enhances recommendation accuracy by considering the qualitative aspects of user feedback rather than just numerical ratings. As illustrated in Figure 1, the distribution of genres across movies is highly diverse, but certain genres dominate the dataset. Drama is the most prevalent genre, reflecting its universal appeal and storytelling depth. This is followed by Comedy, which caters to a wide audience seeking lighthearted entertainment, and Thriller, known for its suspenseful and engaging narratives. The prominence of these genres suggests they are central to the dataset and may align with popular user preferences. Other genres such as Action, Romance, and Horror also hold notable shares, underscoring the variety in movie offerings and user interests.

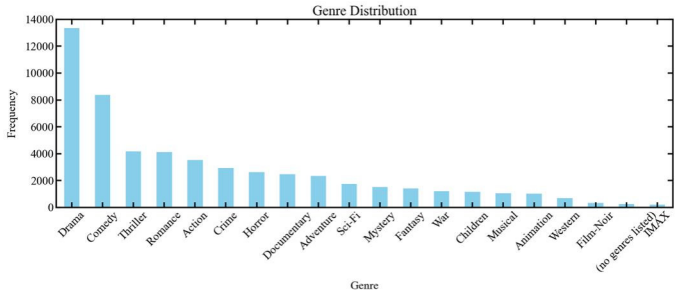

Figure 1. The distribution of genres.

The user rating patterns, visualized in Figure 2, show a clear positive bias. Most ratings are concentrated in the range of 3.5 to 4.5, with a peak around 4.0, indicating that users tend to rate movies favorably. This skew toward higher ratings could be attributed to the dataset's nature, where movies that receive poor ratings may be less frequently watched or rated. The relatively small proportion of ratings below 2.0 suggests that either the movies in the dataset are generally well-regarded or that users are reluctant to provide harsh ratings. The histogram in Figure 2 provides a detailed representation of this trend, helping to understand the overall sentiment in user feedback.

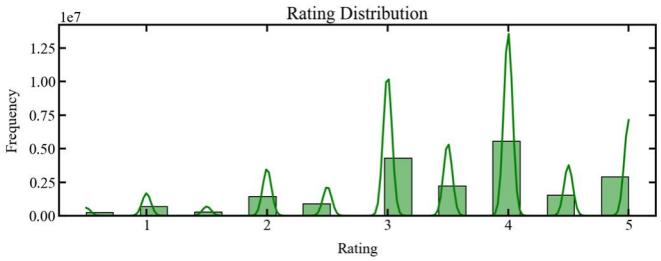

Figure 2. Histogram showing the distribution of user ratings.

An analysis of tag relevance, presented in Figure 3, reveals the most dominant tags associated with movies. These tags provide an additional layer of context to the dataset, highlighting specific qualities and attributes that resonate with users. The tags “original,” “mentor,” and “great ending” emerged as the most relevant, indicating that users often value originality, the presence of a guiding or inspiring character, and a satisfying resolution. Other notable tags include “emotional,” “unexpected twist,” and “visually stunning,” which reflect diverse user preferences and experiences. This analysis underscores the depth of user engagement and the role of specific attributes in shaping movie perceptions.

In summary, the dataset demonstrates a rich diversity in genres, a tendency toward positive ratings, and meaningful engagement with specific movie attributes through tagging. These insights

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 June 2025 doi:10.20944/preprints202506.1150.v1

lay the foundation for further analyses, such as recommendation systems or predictive modeling, by emphasizing the factors that most significantly influence user preferences.

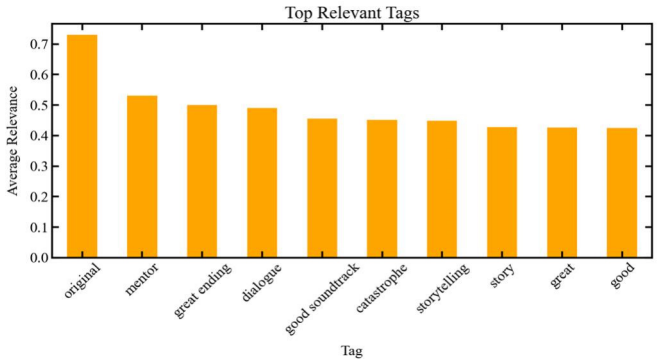

Figure 3. Analyzing the relevance of tags to movies.

B. Results and Performance from Spark Algorithm

The primary goal of this project was to identify the top ten movies in the state of Minnesota based on their movie IDs, titles, and genres. However, due to certain data inconsistencies, some movie titles and genres in the output required clarification. For instance, the movie with MovieId 94833 was incorrectly listed with the title "The (2012)" and genre "The (2012)." The corrected title is "Pirates! Band of Misfits, The (2012)," and the corrected genre is "Adventure|Animation|Children|Comedy." Similarly, for MovieId 858, the title should be "Godfather, The (1972)" instead of "Godfather," and its genre should be "Crime|Drama." Another notable example is MovieId 78772, which was listed as "Twilight Saga: Eclipse," with an incomplete genre. The corrected title and genre are "Twilight Saga: Eclipse, The (2010)" and "Fantasy|Romance|Thriller|IMAX," respectively. Each movie in the list typically has at least two genres, with some having as many as six. Table 1 provides the corrected list of the top eight movies in Minnesota, including their Movie IDs, titles, and genres.

Table 1. The top eight movies in Minnesota with their Movie IDs, titles, and genres.

Number	MovieId	Title	Genres
1	94833	Pirates! Band of Misfits	Adventure
2	89028	Don't Be Afraid of the Dark (2010)	Horror
3	88129	Drive (2011)	Crime
4	858	Godfather	Crime
5	84942	Drive Angry (2011)	Action
6	8360	Shrek 2 (2004)	Adventure
7	80831	Let Me In (2010)	Drama
8	800	Lone Star (1996)	Drama

To ensure the correctness of the output, test cases were implemented and evaluated. For example, the first movie in the top 10 list, "Pirates! Band of Misfits, The (2012)" with MovieId 94833, was verified using the following test case: "val check2 = step3.filter(x => x._ 1.toInt == 94833).take(10)". The test returned: "check2: Array[(String, String)] = Array(94833, "Pirates! Band of

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 June 2025 doi:10.20944/preprints202506.1150.v1

Misfits, genre: The(2012)")". This output confirmed the presence of the movie in the top 10 list, validating the algorithm's accuracy.

Apache Spark played a pivotal role in efficiently handling the large-scale MovieLens dataset. By leveraging distributed computing, Spark significantly reduced data processing time compared to traditional approaches. The use of DataFrames enabled optimized memory utilization, while Spark's SQL- based transformations improved data querying efficiency. Additionally, Spark's built-in machine learning libraries facilitated the implementation of collaborative filtering techniques, ensuring an optimized recommendation process. The dataset used in this project had a size of 150 MB, and the entire process executed within 4 minutes. To handle larger datasets efficiently, performance scaling was evaluated. When doubling the dataset size, execution time can be optimized by increasing the number of executors to 30 and allocating 1 GB of memory per executor. This scaling approach ensures that the Spark algorithm remains efficient even as the dataset size grows, highlighting its suitability for large-scale data processing tasks.

IV. Conclusion

This study presents a scalable and efficient approach to analyzing the MovieLens 20M dataset using Apache Spark, with a focus on identifying the top-ranked movies in Minnesota. The analysis of genres, user ratings, and tag correlations provides key insights into regional user preferences. Furthermore, integrating tag correlation analysis enhances the accuracy of recommendation systems by considering qualitative user preferences beyond numerical ratings. Apache Spark's role in this study demonstrates its effectiveness in large-scale data processing, significantly improving computational efficiency and scalability. The findings emphasize the importance of distributed computing frameworks in modern recommendation systems and highlight potential areas for further enhancement. Despite its strengths, this methodology has certain limitations. The reliance on collaborative filtering approaches may struggle to capture evolving user preferences or address the cold-start problem, where limited data for new users or items hinders recommendations. Additionally, the analysis does not incorporate external contextual factors, such as social media sentiment or real-time user interactions, which could further enhance predictive accuracy. Future research will explore advanced machine learning techniques, such as neural collaborative filtering, to improve personalization and adapt to non-linear user-item relationships. Moreover, integrating external data sources, including social media sentiment analysis and real-time user interactions, will provide a more comprehensive understanding of user preferences and trends. By addressing these aspects, future studies can further optimize recommendation systems and refine big data analytics techniques, ensuring more accurate and personalized content recommendations for users.

References

1. Janssen, M., Van Der Voort, H., & Wahyudi, A. (2017). Factors influencing big data decision-making quality. Journal of business research, 70, 338-345.
2. Farhana, Zaman, Rozony., Mst, Nahida, Aktar, Aktar., Md, Ashrafuzzaman., Ashraful, Islam. (2024). 1. A systematic review of big data integration challenges and solutions for heterogeneous data sources. doi: 10.69593/ajbais.v4i04.111
3. Omaiyma, Abbas., Romaytha, Salih., Samah, Abdalla., Aisha, Elhassan., Al-Alas, Mohammed., Shima, Suliman. (2023). 5. Big data issues and challenges. International Research Journal of Modernization in Engineering Technology and Science, doi: 10.56726/irjmets33205
4. Nand, Kumar, Et, al.. (2023). 2. Harnessing the Power of Big Data: Challenges and Opportunities in Analytics.. doi:10.52783/tjjpt.v44.i2.193
5. Salil, Bharany., Nasser, Taleb., Muhammad, Tariq, Sadiq., Nayab, Kanwal., Taher, M., Ghazal., Manas, Pradhan., Ateeq, Ur, Rehman. (2023). 4. A Comprehensive Review on Big Data Challenges. doi: 10.1109/ICBATS57792.2023.10111375

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 13 June 2025 doi:10.20944/preprints202506.1150.v1

6. Saeed, I., & KUMAR, R. (2023). Challenges and Emerging Patterns in Big Data Analytics. Authorea Preprints.
7. Vishnu, Prasad, Verma., T., P., Sinha., Santosh, Kumar., Nenavath, Srinivas, Naik. (2024). 2. Performance Analysis of Apache Spark Job Schedulers for Big Data Processing. 2017 IEEE Region 10 Symposium (TENSYMP), doi: 10.1109/tensymp61132.2024.10752267
8. Ashima, Sahni. (2024). 3. A Comparative Analysis of Apache Spark Dataframes over Resilient Distributed Datasets (RDDs). Indian Scientific Journal Of Research In Engineering And Management, doi: 10.55041/ijsrem36566
9. Himanshu, Gupta. (2024). 1. Big Data Analytics using Artificial Intelligence: Apache Spark for Scalable Batch Processing. International journal of innovative science and research technology, doi: 10.38124/ijisrt/ijisrt24aug1656
10. Dragan, Stojanović., Dušan, Jovanović., Natalija, Stojanović. (2024). 4. Big Medical Data Analytics Using Apache Spark Framework. doi: 10.1109/icetran62308.2024.10645083
11. Chaganti, Sri, Karthikeya, Sahith., Satish, Muppidi., Suneetha, Merugula. (2023). 5. Apache Spark Big data Analysis, Performance Tuning, and Spark Application Optimization. doi: 10.1109/easct59475.2023.10393086
12. García, S., Luengo, J., & Herrera, F. (2016). Tutorial on practical tips of the most influential data preprocessing algorithms in data mining. Knowledge-Based Systems, 98, 1-29.
13. Salloum, S., Dautov, R., Chen, X., Peng, P. X., & Huang, J. Z. (2016). Big data analytics on Apache Spark. International Journal of Data Science and Analytics, 1, 145-164.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.