

OTT Movies: From Insights to Recommendations

Bobby Doshi, Anant Moudgalya & Apurva Khatri



Introduction



This project employs the Hadoop MapReduce framework to conduct a sophisticated analysis of the Netflix Prize dataset, which comprises over 100 million movie ratings by nearly 480,000 users across 17,000 titles. Utilizing statistical and machine learning techniques, specifically the Apriori algorithm for frequent itemset mining, we aim to both analyze trends in user behavior and develop a personalized movie recommendation system. The distributed processing capabilities of MapReduce allow us to efficiently handle large-scale data, facilitating the computation of complex metrics such as standard deviation, mode, and median of ratings, and the execution of multi-stage processing pipelines necessary for deriving high-confidence association rules for recommendations. This integration of big data analytics into the movie recommendation domain highlights the transformative potential of MapReduce in generating actionable insights from vast datasets.



Table of contents



01

Dataset

Size, Structure & Format Description

02

Frequent Itemset Mining

Statistical Analysis and Association based
recommender

03

Performance Analysis

Discussion on Scalability and Speedup



01

Dataset

Netflix Prize Data



Data Structure & Size




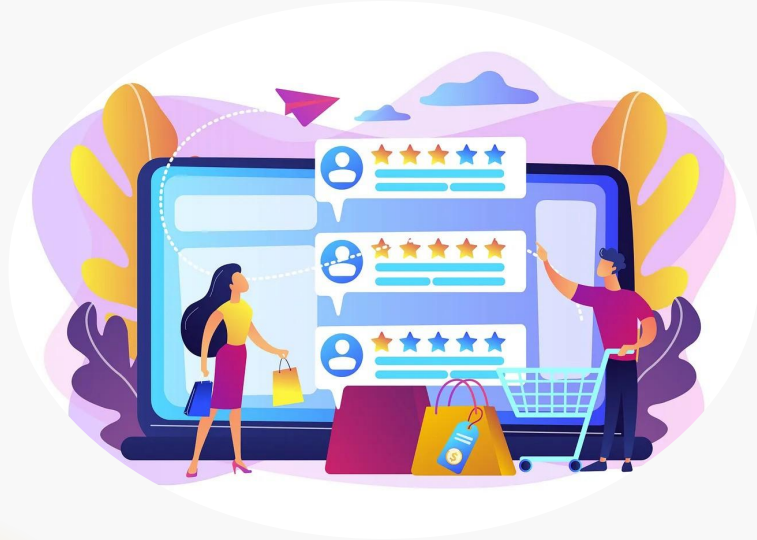
- **Data Points:** The dataset contains over 100 million movie ratings.
- **Users:** Ratings were provided by nearly 480,000 anonymized, randomly-chosen Netflix subscribers.
- **Movies:** The dataset covers over 17,770 movies.
- **Rating Scale:** Ratings are on a scale from 1 to 5 stars.
- **Time Frame:** The dataset includes movie ratings from October 1998 to December 2005.

Format:

movie_id:
user_id, rating, date
user_id2, rating, date

movie_id2:
user_id3, rating, date
user_id, rating, date





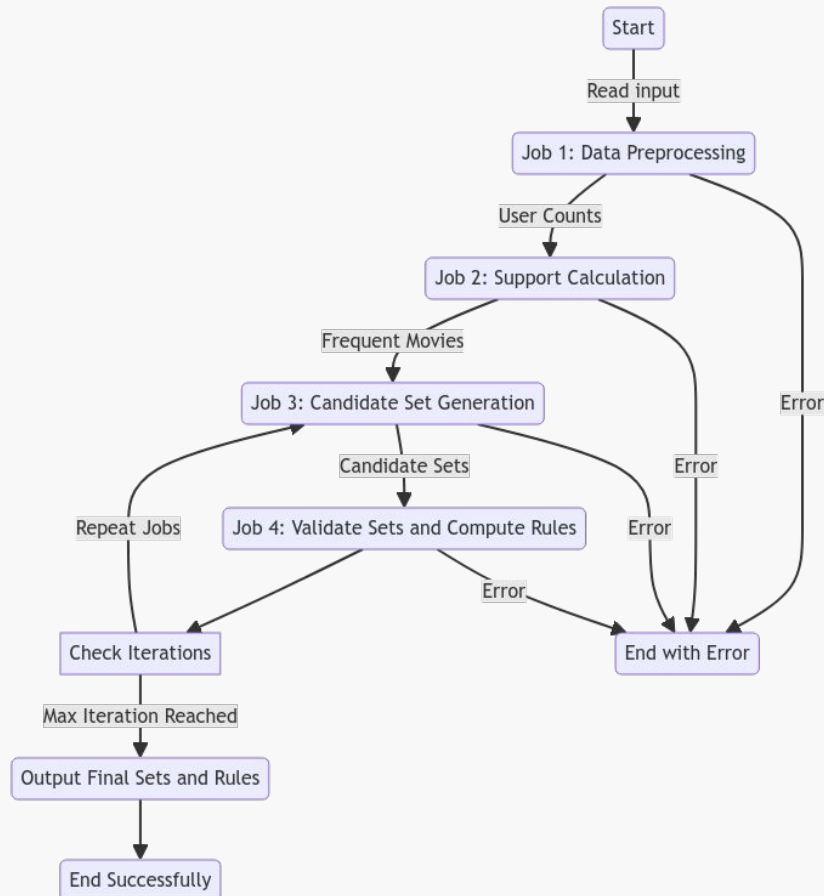
02

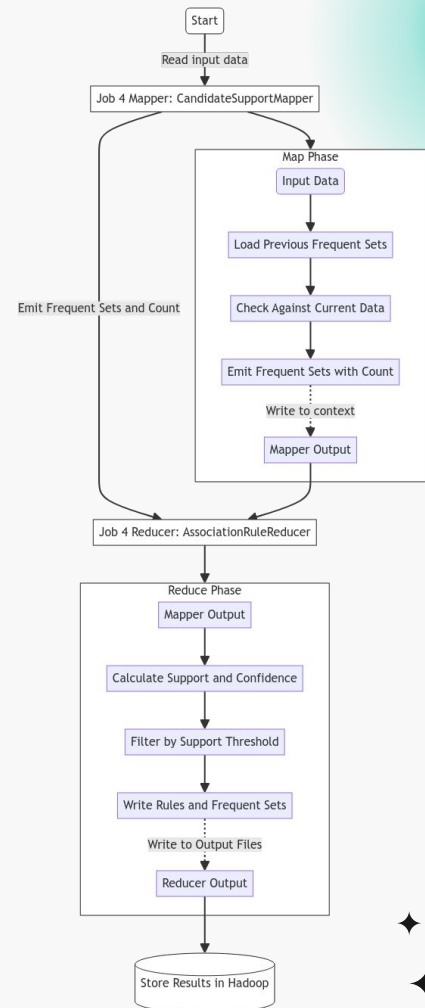
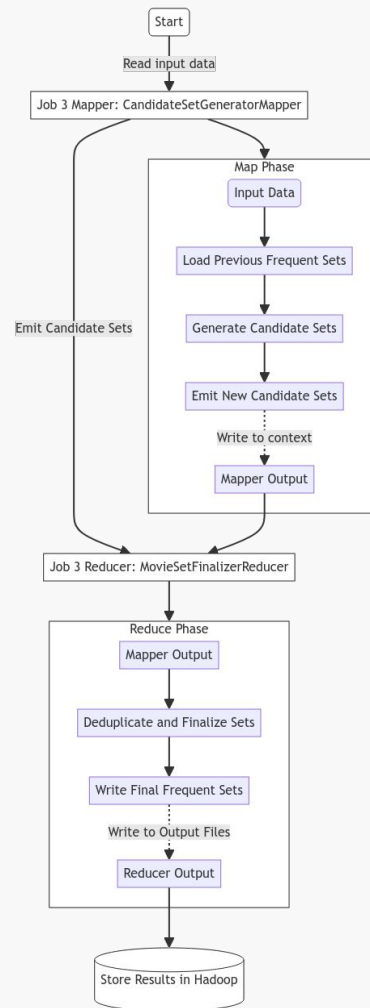
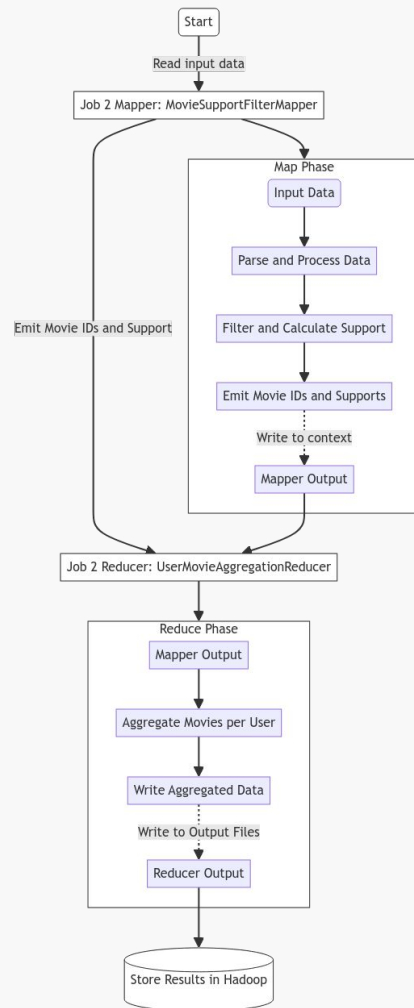
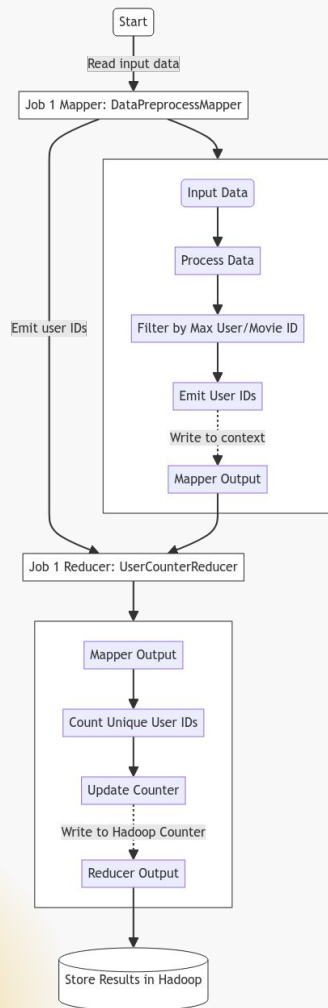
Frequent Itemset -Apriori

Frequent Itemset – Apriori Principle

- Itemset Mining
 - A data mining technique focused on discovering frequent patterns, associations, or strong rules in datasets.
 - Useful in market basket analysis to identify items often purchased together.
- Program Purpose:
 - Computes frequent itemsets and their probabilities ensuring all non-empty subsets are also frequent (Apriori principle).
 - Calculates conditional probabilities for movie recommendations.
- Data Processing Techniques:
 - Utilizes map-only joins and theta joins for data merging.
 - Theta join is optimal given the lower number of unique users (approx. 480k) compared to movie ratings (17 million).
- Program Optimization:
 - Reduces data transfer and computation duplication.
 - Tasks are optimized based on output dependencies from previous stages.

Apriori Mapreduce Flowchart





Statistical Analysis



Average ratings
per user

Standard Deviation
of rating per movie

Mode of rating
per movie

Execution Time with 3 Cores	54 seconds	50 seconds	46 seconds
Execution Time with 6 Cores	44 seconds	44 seconds	48 seconds
Speedup	1.23	1.14	0.9853

Mean of rating
per movie

Median of rating
per movie

Histogram of
rating per
movie

Execution Time with 3 Cores	37.3 seconds	35.6 seconds	44 seconds
Execution Time with 6 Cores	35.4 seconds	34.8 seconds	38 seconds
Speedup	1.06	1.02	1.15

Dataset size - 813,419



02

Performance Analysis

What we did to optimise?



Job 2

- **Support Threshold Pruning:** Movie groups (itemsets) that do not meet the support threshold are discarded early, which reduces unnecessary computations in subsequent iterations.
- **Aggregation Optimization:** Aggregates data at the reducer to minimize data movement across the network and focus computational efforts on potentially frequent itemsets.

Job 3

- **Use of Distributed Cache:** Frequent itemsets from previous iterations are loaded into the distributed cache, making them quickly accessible to all nodes and reducing data redundancy.
- **Efficient Self-Join:** Only attempts to join sets that have a high likelihood of being frequent, using the property that all subsets of a frequent itemset must also be frequent (Apriori property)(Candidate Pruning).

Job 4

- **Incremental Validation:** Each mapper checks current data against previously validated frequent sets, ensuring that only potentially viable candidates are processed.
- **Selective Rule Calculation:** Confidence calculations are performed only for those itemsets that meet the support criteria, significantly reducing the number of calculations required.
- **Efficient Rule Derivation:** Utilizes the confidence measure to derive rules only from the subsets that meet the support threshold, ensuring that the generated rules are both relevant and strong.



SpeedUp

# of Machines	(Max Movie ID, Max User ID)	Running Time (min)
5	(No Limit, No Limit)	113
7	(No Limit, No Limit)	88

The speedup achieved by increasing the number of machines from 5 to 7 is approximately 1.28 which corresponds to a 22.12% decrease in running time.

* Experiments were performed using m5.xlarge machines using 0.15 as support threshold



Scale Up

Movie ID Limit	User ID Limit	Running Time(min)
1000	10000	12
5000	10000	17
10000	10000	25
10000	100000	30

* Experiments were performed using m5.xlarge machines using 0.1 as support threshold

Result Analysis

- First Format: Commonly Reviewed Movie Sets
 - Example: 14240,15124,2452,1905,11521
 - Percentage (e.g., 0.151094673) indicates the proportion of Netflix users who reviewed any movie and also reviewed all movies in the itemset.
 - Notable Set: 14240,2452,1905,11521,15107 corresponds to:
 - Lord of the Rings: The Return of the King (2003)
 - Lord of the Rings: The Fellowship of the Ring (2001)
 - Lord of the Rings: The Two Towers (2002)
 - Pirates of the Caribbean: The Curse of the Black Pearl (2003)
 - Ocean's Eleven (2001)
 - Shows logical grouping based on movie series or similar themes.

Result Analysis

- Second Format: Predictive Analysis of User Viewing Patterns
- Format: [Already Viewed Movies] -> Investigated Movie
 - Example:
 - [3962, 4306] -> 1905 0.8782051282051282
 - [3962, 4306] -> 1905 shows 87.8% of users who reviewed movies 3962 and 4306 also reviewed movie 1905.
- Useful for predicting likelihood of a user reviewing a particular movie based on their past reviews.



References



1. S. Singh, R. Garg, and P. K. Mishra, “Review of apriori based algorithms on MapReduce framework,” CoRR, vol. abs/1702.06284, 2017, Available: <http://arxiv.org/abs/1702.06284>
2. J. Leskovec, A. Rajaraman, and J. Ullman, Mining of Massive Datasets. Cambridge University Press, 2019, pp. 213–251. Available: <http://infolab.stanford.edu/~ullman/mmds/ch6.pdf>
3. “3 9 Frequent Itemsets 29 50,” www.youtube.com, Jul. 23, 2016. <https://www.youtube.com/watch?v=6Bl4wYJl3rY> (accessed March 15, 2024).
4. “3 10 A Priori Algorithm 13 07,” www.youtube.com, Jul. 23, 2016. https://www.youtube.com/watch?v=n2E4Tzt_Teo (accessed March 15, 2024).
5. “3 11 Improvements to A Priori 17 26 Advanced,” www.youtube.com, Jul. 23, 2016. <https://www.youtube.com/watch?v=AGAkNiQnbjY> (accessed March 15, 2024).

Thank You!

Contributors

Bobby Doshi
Anant Moudgalya
Apurva Khatri

CREDITS: This presentation template was created by
Slidesgo, and includes icons by **Flaticon**, and
infographics & images by **Freepik**

