Movie Recommendation System Using Big Data

Akshat Patel
Master's in Data Analytics
San Jose State University
San Jose, USA
akshat.n.patel@sjsu.edu

Pranavi Avula
Master's in Data Analytics
San Jose State University
San Jose, USA
pranavi.avula.sjsu.edu

Sneha Karri
Master's in Data Analytics
San Jose State University)
San Jose, USA
sneha.karri.sjsu.edu

Swaraj Kulkarni
Master's in Data Analytics
San Jose State University
San Jose, USA
swaraj.kulkarni.sjsu.edu

Suhail Chopra
Master's in Data Analytics
San Jose State University
San Jose, USA
suhail.chopra.sjsu.edu

***Abstract*—In order to solve the problem of content discovery in the entertainment sector, this paper develops a sophisticated recommendation system, which is essential in the age of abundant video streaming services and expanding content. Using AWS S3 and AWS Glue for big data processing and the alternating least squares (ALS) model, the study presents a collaborative filtering-based movie recommendation system. The system leverages recent search information to personalize movie recommendations based on user preferences, utilizing PySpark's data-driven methodology. Comprehensive analysis validates the approach's efficacy and shows enhanced platform engagement and content discovery. In the end, the paper highlights how crucial it is to adjust to changing user behavior and leverage big data analytics to drive personalized content searches, thereby addressing content discovery issues in the digital entertainment sector.**

***Index Terms*—**

I. Introduction

The sheer volume of content available in the quickly changing world of digital entertainment presents a major obstacle for users looking for recommendations that are both relevant and personalized. In order to address this issue, big data technologies have become increasingly important, especially when it comes to movie recommendations. The demand for effective and customized content discovery solutions grows as digital media consumption keeps rising. This paper explores how Big Data analytics, with its ability to process large and diverse datasets, is changing how users find and interact with movies. Specifically, it delves into the field of movie recommendation systems.Big Data makes it possible to create sophisticated recommendation models that improve user experience and advance our understanding of how the digital entertainment industry's consumption patterns are changing by utilizing user interactions, preferences, and past behaviors. This introduction lays the groundwork for a thorough examination of the integration of Big Data analytics and movie recommendations, emphasizing the role that this integration will play in influencing the direction of personalized content discovery in the future.

II. SIGNIFICANCE TO THE REAL WORLD

A. Maintaining the Integrity of the Specifications

Big Data-driven movie recommendation systems provide a revolutionary approach to the problem of content discovery. These systems enhance user satisfaction, deliver highly personalized user experiences, and maintain audience engagement in an era of abundant content choices by analyzing large datasets. Beyond the personal gains, the entertainment sector will be greatly impacted by the incorporation of Big Data in movie recommendations, which will optimize content delivery, increase user retention, and offer insightful data on changing consumer behavior. These systems are important because they have the potential to change the way that people engage with real-world cinematic content.

III. Project Motivation

The purpose of this project is to tackle the increasing difficulty of finding content in the quickly growing entertainment sector. Users frequently struggle to find and interact with movies that suit their tastes due to the abundance of content available on streaming platforms and their rapid proliferation. The purpose of this project is to develop a sophisticated recommendation system that uses cloud-based processing, PySpark, AWS EMR, and Glue, along with sophisticated algorithms to offer users personalized movie recommendations. By doing this, we hope to greatly improve user happiness, boost platform engagement, and position our platform as a pioneer in the field of tailored content discovery. Significant changes were brought about by the COVID-19 pandemic, especially in the field of Over-The-Top (OTT) platforms. Lockdowns and restrictions caused a significant shift in user behavior toward the consumption of digital entertainment. This increased the demand on platforms to improve their content discovery mechanisms in addition to providing a sizable library of content. People's interactions with OTT platforms have changed for the better as a result of the COVID-19 pandemic. These platforms were a vital resource for developing a feeling of community, promoting cross-cultural dialogue, and helping up-and-coming artists in addition to being a source of entertainment. The post-pandemic infatuation with over-the-top (OTT) platforms is indicative of a fundamental shift in our understanding of and usage of entertainment in the digital age. Our recommendation system puts us at the forefront of this sector-wide change in response. Furthermore, this project puts us at the forefront of technological innovation in the entertainment industry by aligning with trends towards more individualized and user-centric experiences.

IV. LITERATURE SURVEY

[1] The paper Intends to solve the probelms and limitations of traditional collaborative filtering algorithms when applied to large-scale datasets. The authors present a hybrid recommendation system that combines the benefits of the ALS model and a tag-based approach. The ALS model handles sparsity and huge datasets well. When dealing with several things, suggestion delay and accuracy can decrease. The suggested algorithm addresses these issues. The approach uses HDFS and MapReduce from Hadoop to parallelize and efficiently handle dataset files. MapReduce allows distributed computation across numerous nodes, speeding up and expanding the big data platform.

[2] This paper talks about a MapReduce based user-based collaborative filtering strategy Which solves the problem with high computational complexity of collaborative filtering models. The collaborative filter gives the missing entries in the rating matrix, the authors also review MapReduce-based collaborative filtering algorithms and their drawbacks. They also describe collaborative filtering and formulae for item similarity and rating. The paper implements collaborative filtering techniques utilizing Elastic MapReduce (EMR) clusters on Amazon Web Services (AWS). The AWS-managed Apache Spark framework runs PySpark scripts. The authors set up an AWS EMR cluster with Spark on all nodes.

[3] This study describes a research on scalable recommender system using Apache Spark and Alternating Least Squares (ALS). They have used explicit and implicit feedback which are numerous so, they employed collaborative filtering techniques, particularly ALS,Apache Spark as the distribution platform which effectively processes huge data and iterative machine learning algorithms. They used AWS S3 to store the data. AWS EC2 servers managed by the EMR cluster run Apache Spark and process data using RDDs and DataFrames. Spark SQL queries and manipulates distributed and structured data. The ALS algorithm is trained on the data, and predictions are made to recommend products to users.

V. PROJECT OVERVIEW AND ARCHITECTURE

A. Methodology

We will start by defining the problem and choosing the recommendation type in order to develop a big data movie recommendation system. Next, we will begin gathering the varied data from user preferences and movie databases, clean and prepare it, and perform exploratory data analysis. The dataset is improved by feature engineering, and precise model evaluation is made possible by data splitting. We will select an appropriate algorithm, apply big data technologies, train the model, and perform performance optimization.

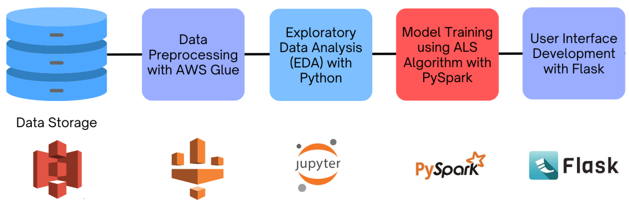

Data Storage
Data Preprocessing with AWS Glue
Exploratory Data Analysis (EDA) with Python
Model Training using ALS Algorithm with PySpark
User Interface Development with Flask
jupyter
PySpark
Flask

Fig. 1. Project Workflow.

After that, we'll deploy the model in a production setting, evaluate it using predetermined metrics, and keep an eye out for maintenance on a constant basis. We'll incorporate user input for ongoing enhancements, put security protocols in place, and record the complete procedure. The recommendation system is updated frequently in accordance with user preferences and new data to ensure it stays effective.

B. Architecture

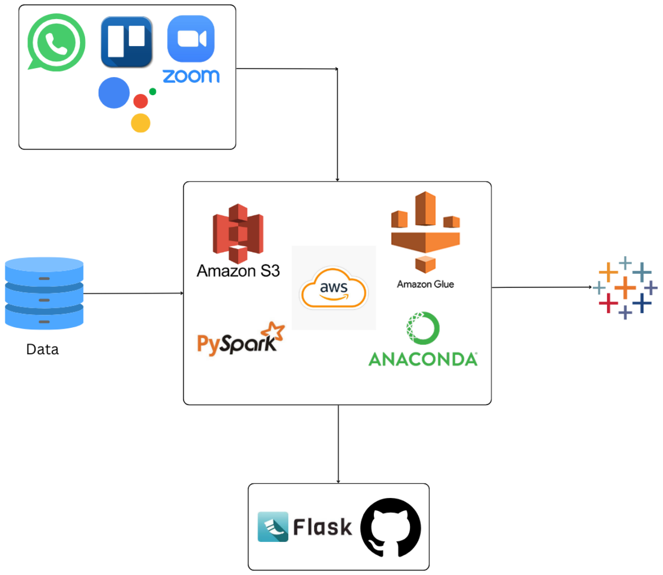


Fig. 2. Our technology stack for this project.

Data Storage: The architecture is based on the data storage layer. Usually, a distributed database system or reliable, scalable storage options like Amazon S3 are used to implement it. User ratings, metadata, raw movie data, and any other data needed by the system are all stored in this layer. In order to accommodate the frequent read and write operations by following components, the storage must offer high-throughput and low-latency activities.

Amazon Glue Data Preprocessing: An entirely managed ETL solution for preparing and transforming data for analytics is AWS Glue. It offers a code-free interface for creating, executing, and monitoring ETL processes in addition to automatically determining the data structure. Data cleaning, normalization, and transformation are steps in the AWS Glue preprocessing process that guarantee the data is in the proper format and structure for analysis. This stage may involve data type conversion, addressing missing values, deduplication, and other activities.

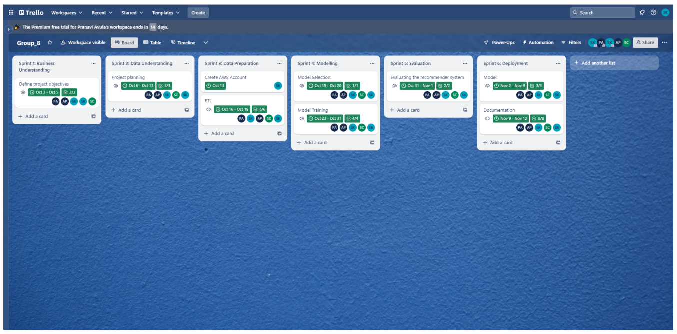
Using Python for Exploratory Data Analysis (EDA): In data science, exploratory data analysis (EDA) is a critical phase that enables analysts to recognize patterns, correlations, and anomalies in data. In a Jupyter Notebook, data scientists can visualize data distributions, find outliers, and get insights that guide feature selection and recommender system algorithm design by using Python tools like Pandas, Matplotlib, and Seaborn. In order to improve the data pipeline, this iterative procedure frequently feeds back into the preparation stage.

Model Training with PySpark and the ALS Algorithm: Based on prior behavior, the ALS algorithm is a collaborative filtering technique that forecasts user preferences. To forecast ratings for unrated films, a movie recommender system would look at user ratings from the past. PySpark makes it possible to divide this processing among several cluster nodes, making it possible to handle big datasets effectively. Model training entails choosing the right hyperparameters, using historical data to train the model, and assessing how well it performs.

Flask User Interface Development: Flask was selected for the user interface due to its ease of use, flexibility, and small weight. The front end of the system where users review movies and get recommendations is called the user interface. With Flask, you can create webpages that show movie listings, accept user input, and show user-recommended movies. In the background, Flask directs user requests to the Python backend, which retrieves and shows the recommendations by interacting with the trained recommender model.

C. Tools

- Trello (Agile project life-cycle management) - With Agile in mind, we use Trello to manage the project life-cycle. We typically meet for one hour on Fridays of each week to work on homework and class projects.
- Grammarly - To check the grammar in documentation
- Python - For data processing, we will use python programming language
- GitHub and GitHub Desktop - Project version control
- Tableau - Data visualization tool for data analytics
- Amazon S3 - The preprocessed dataset will be stored on Amazon S3, which enables scalability and durability for data storage.
- AWS Glue - For this dataset, AWS Glue is a fully managed extract, transform, and load (ETL) service that simplifies data transfers between data storage.
- Jupyter Notebook - We will perform data exploration, data cleaning, data processing and loading.
- PySpark - PySpark, the Python Spark API, allows us to write Spark applications in Python. We will be building a PySpark application to read data from AWS EMR, AWS S3, perform transformations, and prepare the data for modeling.
- Exploratory Data Analysis - The distribution of ratings, the number of ratings per user, and the ratings for each movie can all be found using EDA.
- Alternating Least Squares(ALS) - We will be using this algorithm for model training with PySpark.
- Flask - The movie recommendation system's user interface is implemented using Flask. Users can view suggested movies, interact with the recommendation system, and enter preferences using this web interface.
- Canva - A graphic design tool used for creating professional visuals for project report and documentation, offering a wide range of design elements.
- LaTeX (Over leaf)- Used for high-quality typesetting and formatting our documents according to IEEE standards.
- DALL-E - AI generated visuals for presentation.

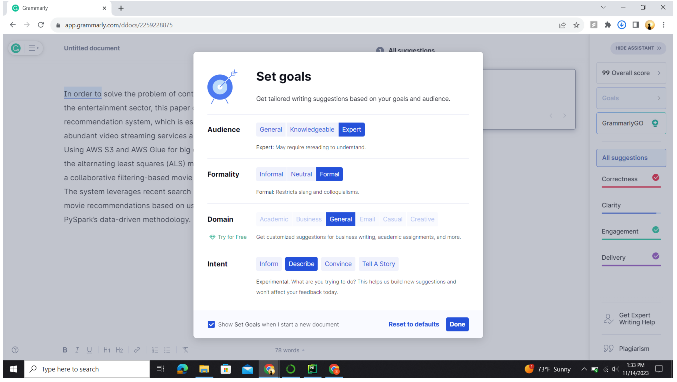


The Premium free trial for Pranavi Avula's workspace ends in 14 days.

Group_8

Sprint 1: Business Understanding	Sprint 2: Data Understanding	Sprint 3: Data Preparation	Sprint 4: Modelling	Sprint 5: Evaluation	Sprint 6: Deployment
Define project objectives Oct 3 - Oct 5 · 3/3	Project planning Oct 6 - Oct 13 · 3/3	Create AWS Account Oct 13	Model Selection: Oct 19 - Oct 20 · 1/1	Evaluating the recommender system Oct 31 - Nov 1 · 2/2	Model: Nov 2 - Nov 9 · 3/3
		ETL Oct 16 - Oct 19 · 6/6	Model Training Oct 23 - Oct 31 · 4/4		Documentation Nov 9 - Nov 12 · 8/8

Fig. 3. Trello for project management.



app.grammarly.com/ddocs/2259228875

Untitled document

In order to solve the problem of cont
the entertainment sector, this paper
recommendation system, which is es
abundant video streaming services a
Using AWS S3 and AWS Glue for big
the alternating least squares (ALS) m
a collaborative filtering-based movie
The system leverages recent search
movie recommendations based on us
PySpark's data-driven methodology.

Set goals

Get tailored writing suggestions based on your goals and audience.

Audience: Expert
Expert: May require rereading to understand.

Formality: Formal
Formal: Restricts slang and colloquialisms.

Domain: General
Get customized suggestions for business writing, academic assignments, and more.

Intent: Describe
Experimental. What are you trying to do? This helps us build new suggestions and won't affect your feedback today.

☑ Show Set Goals when I start a new document

99 Overall score

78 words

Fig. 4. Grammarly for grammar correction.

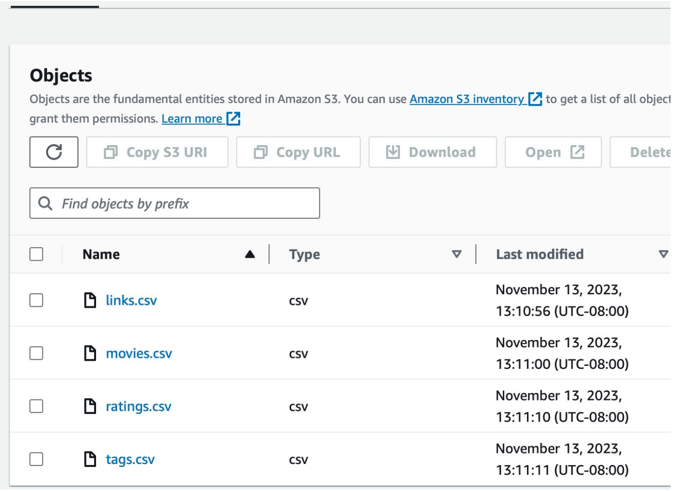
VI. Data, Tools and Technologies

A. Dataset

Name - Movie Recommendation using Big Data
https://grouplens.org/datasets/movielens/

About the dataset - A foundation for building and evaluating movie recommendation systems is provided by the MovieLens dataset, which is available in various sizes and contains user ratings, movie metadata, and timestamps. It allows for the creation of user-item matrices, which are crucial for recommendation algorithms, with unique movie and user IDs. A broad range of user preferences can be accommodated by the genre diversity of the dataset. This dataset can be used for research on personalized recommendation, benchmarking, algorithm development, and temporal analysis, which will greatly advance the field of big data-driven movie recommendation systems.

B. Data Storage: AWS S3



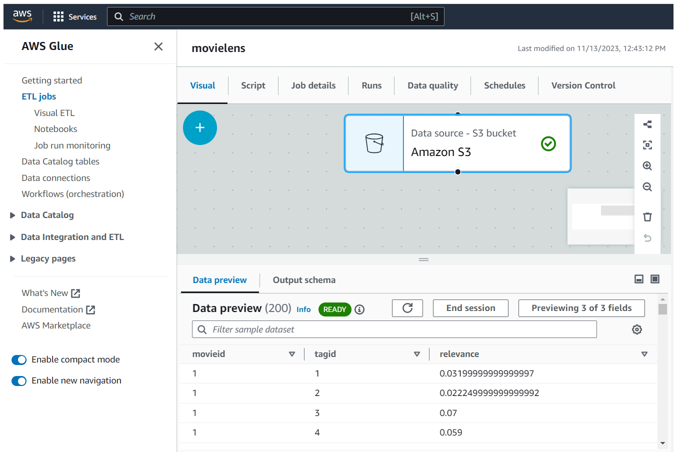
Objects

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all object grant them permissions. Learn more

□	Name ▲	Type ▽	Last modified ▽
□	links.csv	csv	November 13, 2023, 13:10:56 (UTC-08:00)
□	movies.csv	csv	November 13, 2023, 13:11:00 (UTC-08:00)
□	ratings.csv	csv	November 13, 2023, 13:11:10 (UTC-08:00)
□	tags.csv	csv	November 13, 2023, 13:11:11 (UTC-08:00)

Fig. 5. Data Stored in S3 bucket.

Simple Storage Service (Amazon S3) buckets are essential to cloud storage architecture because they provide a reliable and safe way to store enormous volumes of data. These buckets serve as containers where data objects, such as files and databases, are stored. Every S3 bucket is located within a particular AWS Region that has been selected to maximize considerations such as latency, cost, and data sovereignty. Because bucket names are globally unique on AWS, every data object may be uniquely addressed through URLs. The security model for S3 is comprehensive, allowing fine-grained access control through policies and permissions. While root user credentials can create and manage buckets, best practices dictate using AWS Identity and Access Management (IAM) for enhanced security. Additionally, S3 offers features to prevent public access, ensuring that data is only accessible to authorized users.



movieid	tagid	relevance
1	1	0.031999999999999997
1	2	0.022249999999999992
1	3	0.07
1	4	0.059

Fig. 6. AWS Glue ETL Job Configuration.

C. ETL: AWS Glue

Our project's ETL procedure uses Amazon Glue to expedite the loading, transformation, and extraction of our data—which is made up of CSV files stored on Amazon S3—into our system. AWS Glue is a managed service that orchestrates these ETL workflows, enabling us to focus on data analysis. The AWS Glue and Spark execution contexts, which are essential for the distributed processing of massive datasets, are initialized by the ETL operation as it is specified.

More specifically, the operation reads through the CSV files stored in S3, parses them using the specified schema—which includes headers and comma separators—and extracts data from them. Complete data extraction, including any nested directories inside the S3 route, is guaranteed by the'recurse' option in the job parameters. The integrity of our project depends on this exhaustive extraction since it ensures that all relevant data is included.

We are able to streamline the process of extracting data by using AWS Glue's ETL service. This opens the door to further transformation phases, such as data enrichment, consolidation, and cleaning. The extraction phase is sealed and made available for the subsequent stages of our data pipeline by the task commit action at the end of the procedure. This method not only makes our data operations more efficient, but it also guarantees that our datasets are ready for top-notch analysis and insight production.

D. Tableau

Tableau desktop is a powerful data visualization tool, where we can create dynamic visual representation used for analysis. It provides the facility to connect live data from AWS S3 bucket, and perform visual analysis on it. In the figure below we can see the bar chart titled "Movies with top ten ratings," showing top ten films with their corresponding ratings count. "The Shawshank Redemption (1994)" has the highest rating count, followed by other popular films from the 1990s, indicating these movies are rated most favorably in the dataset and a pie chart where Drama is the largest segment with 270,425,

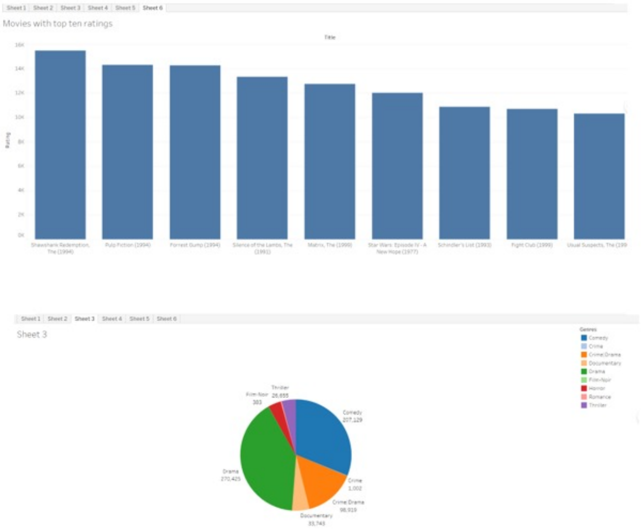
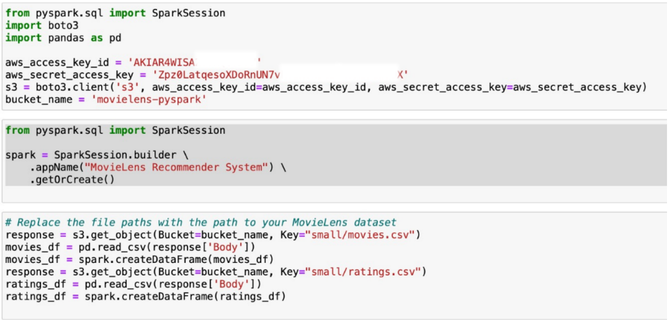


Fig. 7. Visualizations with tableau

followed by Comedy and Crime/Drama, while Film-Noir is the smallest with 383, indicating Drama is the most prevalent genre in the dataset.

E. Pyspark Connection



```
from pyspark.sql import SparkSession
import boto3
import pandas as pd

aws_access_key_id = 'AKIAR4WISA[redacted]'
aws_secret_access_key = 'Zpz0LatqesoXDoRnUN7v[redacted]X'
s3 = boto3.client('s3', aws_access_key_id=aws_access_key_id, aws_secret_access_key=aws_secret_access_key)
bucket_name = 'movielens-pyspark'
```

```
from pyspark.sql import SparkSession

spark = SparkSession.builder \
    .appName("MovieLens Recommender System") \
    .getOrCreate()
```

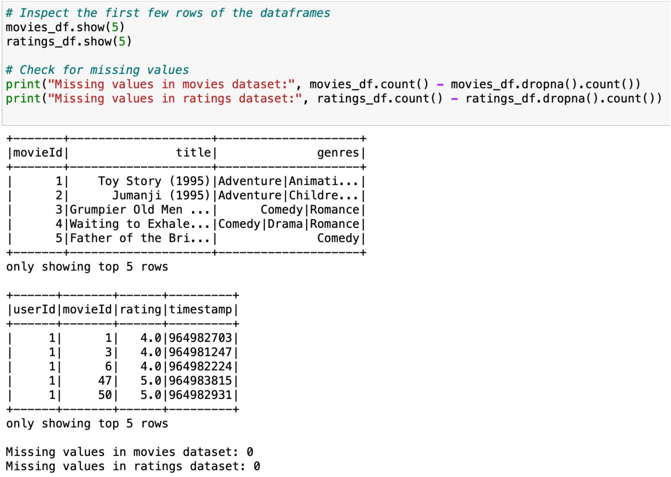
```
# Replace the file paths with the path to your MovieLens dataset
response = s3.get_object(Bucket=bucket_name, Key="small/movies.csv")
movies_df = pd.read_csv(response['Body'])
movies_df = spark.createDataFrame(movies_df)
response = s3.get_object(Bucket=bucket_name, Key="small/ratings.csv")
ratings_df = pd.read_csv(response['Body'])
ratings_df = spark.createDataFrame(ratings_df)
```

Fig. 8. PySpark AWS Connection

The key component of our data pipeline is the section of Python code that is shown in the image. It represents the step where we connect to AWS services and load the data into our processing environment. The code begins by importing the libraries required for AWS access and data manipulation using the PySpark library. After that, programmatic access to the S3 data storage is enabled by securely setting the AWS credentials. Then, we build a Spark session and give it an application name. This is how our data operation within the Spark ecosystem begins.

The code then uses the boto3 library to communicate with the S3 service and retrieve CSV files containing our datasets. The reading of these files into pandas DataFrames serves as a preliminary step to data transformation. Finally, by utilizing Spark's distributed data processing capabilities, these DataFrames are transformed into Spark DataFrames. Large-scale data handling and the ensuing analytics activities in our project, such creating a movie recommendation engine, depend on this conversion. This pipeline component is crucial for connecting our stored data to Spark's analytical capabilities, completing the picture from data storage to data insight.

F. Data Cleaning, and Preprocessing:



```
# Inspect the first few rows of the dataframes
movies_df.show(5)
ratings_df.show(5)

# Check for missing values
print("Missing values in movies dataset:", movies_df.count() - movies_df.dropna().count())
print("Missing values in ratings dataset:", ratings_df.count() - ratings_df.dropna().count())
```

```
+-------+--------------------+--------------------+
|movieId|               title|              genres|
+-------+--------------------+--------------------+
|      1|    Toy Story (1995)|Adventure|Animati...|
|      2|      Jumanji (1995)|Adventure|Childre...|
|      3|Grumpier Old Men ...|      Comedy|Romance|
|      4|Waiting to Exhale...|Comedy|Drama|Romance|
|      5|Father of the Bri...|              Comedy|
+-------+--------------------+--------------------+
only showing top 5 rows

+------+-------+------+---------+
|userId|movieId|rating|timestamp|
+------+-------+------+---------+
|     1|      1|   4.0|964982703|
|     1|      3|   4.0|964981247|
|     1|      6|   4.0|964982224|
|     1|     47|   5.0|964983815|
|     1|     50|   5.0|964982931|
+------+-------+------+---------+
only showing top 5 rows

Missing values in movies dataset: 0
Missing values in ratings dataset: 0
```

Fig. 9. Cleaning and Processing the Data.

Meticulous steps have been taken to prepare the dataset for the movie recommendation system. To prevent errors, duplicate values were eliminated, and missing values were dealt with by deletion. To ensure dependability and clarity, uniform data formats and formatting guidelines were applied to all tables.

G. Algorithm Implementation

Alternating Least Squares Method

The Alternating Least Squares (ALS) algorithm factorizes a given matrix into two factors in such a way that the product of the transpose of the first matrix and the second matrix gives us the parent matrix. These factor matrices can be called the user and item matrix, respectively. Latent factors are given to the algorithm.

To find the user and item matrix we use the following optimization problem:

$$\sum_{(i,j)\in\text{ratings}} (r_{ij} - p_i \cdot q_j)^2 + \lambda \left(\sum_i \|p_i\|^2 + \sum_j \|q_j\|^2 \right) \quad (1)$$

Where:

- r_{ij} is the rating given by user i to item j.
- p_i is the latent vector for user i.
- q_j is the latent vector for item j.
- λ is the regularization parameter.

The optimization problem is solved iteratively by alternately fixing P and optimizing Q, and then fixing Q and optimizing P. The update rules for P and Q are given by:

$$p_i = (Q^T Q + \lambda I)^{-1} Q^T r_i, \tag{2}$$

$$q_j = (P^T P + \lambda I)^{-1} P^T r_j, \tag{3}$$

where:

- r_i is the vector of ratings given by user i.
- r_j is the vector of ratings for item j.

Our movie recommendation system uses the Alternating Least Squares (ALS) method to handle vast amounts of data effectively and generate precise recommendations. When predicting user preferences based on limited information, ALS excels at identifying latent elements from user-movie interactions. Its capacity to factorize the user-item matrix even in the presence of sparse data guarantees that we can provide consumers with tailored movie recommendations, hence enhancing their experience. The suggestions are further refined with the aid of ALS's iterative optimization technique, which switches between correcting user and item aspects. This methodology helps us to improve the overall efficacy of our recommendation system by accommodating a wide range of user preferences and continuously adapting to fresh data.

By forecasting the interests of incoming users based on the combined preferences of the current user base, ALS also helps recommendation systems overcome common problems like the cold start problem. The regularization parameter of the algorithm is crucial in avoiding overfitting, maintaining the robustness of our model and enabling its good generalization to novel, unseen user-movie interactions. We are able to improve the precision of our suggestions and create a framework that can expand with our dataset by incorporating ALS into our system. This guarantees that our recommendation system stays cutting-edge, offering a fun and exciting movie discovery experience.

VII. Output

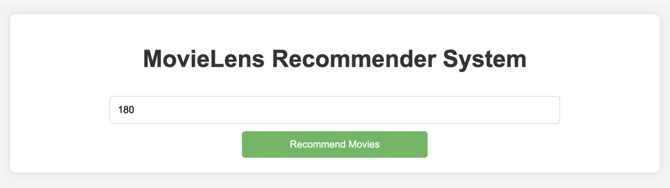
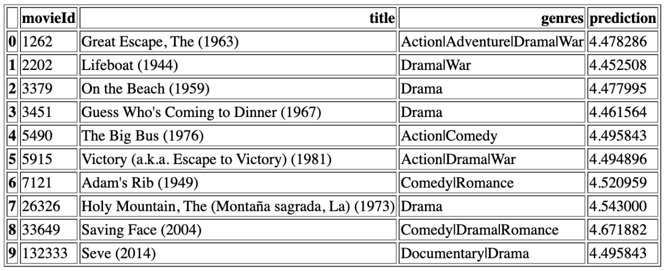


Fig. 10. UI.

VIII. Lesson Learnt

This project made us learn about different software's in big data and how different they are from traditional storage and modelling techniques. We could comprehend and use some related concepts like RDD's and MapReduce. We learnt how to navigate the Amazon Web services in depth, how to connect various tools with one another, using crawlers and we have delved deeply into the algorithms, learnt how to create html files and search engines.



	movieId	title	genres	prediction
0	1262	Great Escape, The (1963)	Action|Adventure|Drama|War	4.478286
1	2202	Lifeboat (1944)	Drama|War	4.452508
2	3379	On the Beach (1959)	Drama	4.477995
3	3451	Guess Who's Coming to Dinner (1967)	Drama	4.461564
4	5490	The Big Bus (1976)	Action|Comedy	4.495843
5	5915	Victory (a.k.a. Escape to Victory) (1981)	Action|Drama|War	4.494896
6	7121	Adam's Rib (1949)	Comedy|Romance	4.520959
7	26326	Holy Mountain, The (Montaña sagrada, La) (1973)	Drama	4.543000
8	33649	Saving Face (2004)	Comedy|Drama|Romance	4.671882
9	132333	Seve (2014)	Documentary|Drama	4.495843

Fig. 11. Output

IX. Innovation

This-project combines multiple technologies and tools to solve a real world problem, making it innovative. We can discuss about these-tools here. Using PySpark shows how flexible-Python-is for Spark-applications. The project's cloud-based-strategy leverages S3, EMR and other AWS-services-to-improve-scalability and data storage. With AWS-Glue, ETL simplifies data transfer and streamlines workflow. The project uses exploratory data analysis-(EDA) to explore patterns and improvements-in-data structures for data-driven-decision making. The use of collective filtering techniques such as Alternating Least-Squares (ALS) for recommendation algorithms-implies advanced machine learning. Interactive data visualization in Tableau and Tableau Public Server improves project performance-and-presentation. The project's strategy, which encompasses technology from data pipeline storage and processing-to analysis and visualization, distinguishes it, and enables for insights from the movie-quantitative-database.

X. Team Work

TABLE I
Team Member Roles

Team Members	Role
Akshat Patel	Data Processing, Cloud Configuration, Technical Documenter
Pranavi Avula	Architecture Development, Data Analysis, Technical Documenter
Swaraj Kulkarni	Data Analysis, Recommender System , Quality Assurance
Sneha Karri	Data Analysis, Quality Assurance, Technical Documenter
Suhail Chopra	Recommender System, Cloud Configuration, Architecture Development

XI. Pair Programming

We have achieved progress through collaborating in pairs and working on the code, as it results in members from different backgrounds to contribute whatever knowledge they have. Individuals with experience in Cloud computing provided support to the other programmer who lacked proficiency in this area. At the same time, Member who possessed expertise in Python programming have contributed to the development of the recommender system and have offered valuable insights and approaches for effectively handling data. Individuals that possess strong skills in documentation and visualization gave assistance to their peers through the provision of guidance.

XII. Relavance to the course

Our project involves AWS S3 and Glue which are integral parts of the Big Data ecosystem. We needed to have an understanding of how to process and store large datasets in distributed environments. We used PySpark, which is is a Python library for Apache Spark, which is a fast and general-purpose cluster computing system. The project involves using PySpark for collaborative filtering (ALS algorithm), demonstrating practical application of Spark in a big data context. The ALS algorithm of collaborative filtering is a recommender algorithm that is an extension of MapReduce paradigm which is covered in our course. The Resilient Distributed Datasets (RDDs), DataFrames, and Datasets are a core part which are crucial for effective big data processing with Spark. principles of collaborative filtering involve understanding relationships and connections between users and items, akin to mining graphs. AWS S3 is a key component for polyglot persistence in the project.

XIII. Technical Difficulties

We have had issue regarding managing the project of this scope, the team had not worked with Big Data tools before, so we had to learn everything from beginning. We had multiple errors while installing Apache Spark in the system and figure which platform to use it on. The Datasets which we wanted to use to solve the business problems were not up to the mark, so we used Movie Lens dataset. There was a problem with giving access of the IAM role to all the team members, we faced some errors there. The ETL process was challenging, and time taking, and we had to try multiple times before we got it right. The exporting of the data from this transformation was a time and resource consumer. We had to train the data in multiple ways, for improved performance.

XIV. Novelty

Fig. 12. Example of a figure caption.

XV. Impact

With entertainment industry being a huge business, the recommendation system which enhances user engagement, would make a huge impact. By using Collaborative Filtering, we provide opersonal recommendations, increasing user retention and more content to be discovered by them leads to more revenue. The integration of AWS S3, AWS Glue, Flask, ensures budget friendly deployment, contributing to a seamless experience of the user.

XVI. Discussions and conclusion:

Our movie recommendation system, utilizing filtering algorithms and AWS services has an improvement on user engagement, in the competitive streaming industry. After implementing ETL processes, which improve quality of our data, we can provide recommendations using our algorithm, that enhance content discovery and encourage users to stay engaged ultimately leading to increased viewer involvement and potential revenue growth. To improve accuracy, we can consider integrating Other Pyspark models by doing some research in the field and exploring real time recommendation capabilities while incorporating user feedback loops. It's crucial for the system to broaden its scope by including content types and adapting to emerging technologies to remain relevant in a changing landscape.

Acknowledgment

We would like to extend our appreciation to the Department of Applied Data Science at San Jose State University. We extend our sincere gratitude to Professor Pendyala, for his support and insights during the course of this project. We would like to extend our gratitude towards the Instructional Student Assistants (ISAs) for their valuable support regarding any doubts.

Appendix

A. Code Walkthrough

We have explained each part of thew code used in different stages of architecture, and a detailed demo is given during the presentation.

B. Discussion / Q&A

We have allocated time for queries and prospect for discussion at the end of each presentation.

C. Demo

Recommender system is being executed during the presentation to demonstrate the results.

D. Report

We have meticulously documented each stage of the project and demonstrated the result of each stage in the form of snippets. We used formal language for the entire project in an appropriate way. We have adhered to the IEEE guidelines, used LaTex for formatting the report.

E. Version control

We have created a GitHub repository for tracking the source code. We have used Git commands like git init and git push to synchronize our local project with the GitHub repository, ensuring that rubrics and project files are included in the commit.

F. Lessons Learned

In the course of creating the project, we had the opportunity to learn about many new technologies, where we encountered numerous challenges like errors while coding, errors with amazon products, but it was a big learning experience. We have described these learnings in our Lessons Learnt chapter in the report.

G. Prospects of winning competition / Publication

We have used collaborative filtering algorithms and AWS services tackles a streaming industry obstacle, making this project a potential competitor and eligible for publication. We have emphasized scalability, efficiency, and user-centricity, which affects viewer engagement and profitability. Advanced machine learning and content adaptation improve popularity. Transdisciplinary potential and technical relevance make the project competitive in changing conditions.

H. Innovation

The innovation lies in the combination of Amazon services and pyspark for this project, we have mentioned about it in our report.

I. Teamwork

All the team members have been involved in every stage of the process of the project's course, with everyone giving input based on their strong suite. We have conducted several meetings and shared the project's ideas and helped each other with errors encountered. We have mentioned the roles in the teamwork section of our report.

J. Technical difficulty

We have included a section for the errors faced during the course of the project.

K. Pair Programming

We have documented the process of pair programming, in our report under the section, pair programming.

L. Practiced agile / scrum

We have used Trello board by Atlassian for Project Management and to keep track of the progress of the project. We have used the Kanban board as it was user friendly. We have created stages in the form of CRISP DM method, dividing the work into each stage under cards and lists, then assigned the tasks to members. We have created Sprints for each stage depending on the difficulty of the task and adhered to it, to provide results on time by conducting regular meetings over Zoom. The Gannt table, is a visual representation of each sprint, and it impact. The ISA's have been invited to join the board and verify the board.
https://trello.com/b/ee5GUAUC/group8

M. Used Grammarly/ other tools

We have used Grammarly for checking the grammatical errors for the research paper.

N. Presentation Techniques

We used Prezi to create our presentations for both Elevator pitch video and the main presentation. We found it to be non-linear and had a dynamic approach towards visual engagement compared to Microsoft power point. It is more flexible than tools like Canva.

References

[1] L. Peng and A. Maalla, "Hybrid Collaborative Filtering Recommendation Algorithm for ALS Model Based on a Big Data Platform," 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, 2021, pp. 1808-1813, doi: 10.1109/IAEAC50856.2021.9391008

[2] S. Manakkadu, S. P. Joshi, T. Halverson and S. Dutta, "Top-k User-Based Collaborative Recommendation System Using MapReduce," 2021 IEEE International Conference on Big Data (Big Data), Orlando, FL, USA, 2021, pp. 4021-4025, doi: 10.1109/BigData52589.2021.9671395.

[3] L. Chen, R. Li, Y. Liu, R. Zhang and D. M. -k. Woodbridge, "Machine learning-based product recommendation using Apache Spark," 2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), San Francisco, CA, USA, 2017, pp. 1-6, doi: 10.1109/UIC-ATC.2017.8397470.

[4] Sreeram Nudurupati, Essential PySpark for Scalable Data Analytics: A beginner's guide to harnessing the power and ease of PySpark 3 , Packt Publishing, 2021.

[5] K. More, P. Jawale, F. Francis and A. Narote, "Review of S3 Data Summarization and Visualization," 2023 International Conference on Intelligent Systems for Communication, IoT and Security (ICISCoIS), Coimbatore, India, 2023, pp. 355-360, doi: 10.1109/ICISCoIS56541.2023.10100403.

[6] Serkan Sakinmaz, Python Essentials for AWS Cloud Developers: Run and deploy cloud-based Python applications using AWS , Packt Publishing, 2023.

[7] Y. K. Guntupalli, V. S. Saketh, S. Amudheswaran and D. S. Vaishnav, "High-Scale Food Recommendation Built on Apache Spark using Alternating Least Squares," 2020 5th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE), Jaipur, India, 2020, pp. 1-5, doi: 10.1109/ICRAIE51050.2020.9358277.

[8] F. Liu, S. P. R. Asaithambi and R. Venkatraman, "Hybrid Personalized Book Recommender System Based on Big Data Framework," 2023 25th International Conference on Advanced Communication Technology (ICACT), Pyeongchang, Korea, Republic of, 2023, pp. 333-340, doi: 10.23919/ICACT56868.2023.10079457.

[9] Prashant Lakhera, AWS for System Administrators: Build, automate, and manage your infrastructure on the most popular cloud platform – AWS , Packt Publishing, 2021.

[10] Manos Samatas, Actionable Insights with Amazon QuickSight: Develop stunning data visualizations and machine learning-driven insights with Amazon QuickSight , Packt Publishing, 2022.