



# Recommendation Engine using pyspark

Our way of making our products  
reach our customer's hands

**TEAMMATES:-**

Isha Kumari (e23mcag0013)  
Kanupriya(e23mcag0109)

# Today's Agenda

## Key takeaways:

- Introduction
- Objective
- Goals
- Methodology
- Implementation
- Use cases



# INTRODUCTION

## Recommendation Engine

- A recommendation engine is a type of system or software application that analyzes patterns in data to suggest items or content to users.
- These items can include products, services, articles, movies, music, or any other type of content that the user may be interested in.
- Recommendation engines are commonly used in various online platforms, such as e-commerce websites, streaming services, social media platforms, and content websites.

# What is the purpose of PySpark?

- It allows you to seamlessly mix SQL queries with Spark programs.
- With PySpark DataFrames you can efficiently read, write, transform, and analyze data using Python and SQL.
- Whether you use Python or SQL, the same underlying execution engine is used so you will always leverage the full power of Spark.



# ■ Objective

- The objective is to anticipate and guide customers to better product choices for their needs, to make more purchases and to do repeat business .
- It provides a rich set of APIs for data manipulation, processing, and analysis, making it an ideal choice for data engineering tasks.
- **For Example :-** Companies such as Netflix and Amazon use recommendation engines to customize home screens with relevant product and content suggestions.



# GOALS

The Goals of the MovieLens recommendation engine include :--

- 1)Diversity - The system offers a diverse range of movie suggestion to cater to different user interests and preferences.
- 2)Serendipity: To provide recommendations that surprise and delight users, introducing them to movies they might not have considered otherwise.
- 3)Satisfaction: To enhance user satisfaction by consistently delivering valuable and enjoyable movie recommendations.
- 4)Accuracy: Align with the user's tastes and preferences to provide accurate movie recommendations.
- 5)Novelty: It helps users discover hidden gems to recommend new and lesser-known movies.
- 6)Business Growth: Contribute to the growth of the business by increasing user retention.

# Methodology

## ■ Data preprocessing:

- such as handling missing values
- removing duplicates
- feature engineering
- For Example:- extracting user preferences, movie genres).

## ■ Model selection:

- content-based filtering
- hybrid approaches
- discuss the rationale for selecting these models.

## ■ Model training and evaluation:

- splitting the dataset into training and testing sets
- training the selected models on the training data
- evaluating their performance using metrics such as MAE, RMSE, Precision, Recall, and F1-score.

# Implementation

- **Data loading and preprocessing:**

Provide more details on how the MovieLens dataset is loaded into memory and preprocessed to prepare it for model training.

- **Model development:**

Explain how the selected recommendation models are implemented using PySpark, including any libraries or frameworks used (e.g., Surprise, scikit-learn).

- **Training and evaluation:**

Elaborate on the process of training the recommendation models on the preprocessed dataset and evaluating their performance using the defined evaluation metrics.

- **Deployment considerations:**

Discuss potential deployment options for the recommendation engine, such as integrating it into a web application or exposing it as an API for real-time recommendations.



# Use Cases

## Real-world applications:

Provide examples of real-world applications of recommendation engines, such as:-

- **e-commerce platforms**
- **streaming services**
- **social media platforms and etc.**

to illustrate the broader impact of personalized recommendations on user engagement and satisfaction.



**THANKYOU**