Frist Page

Content

# Introduction

In 2025, **watching movies is a widely accessible and popular activity**. Streaming platforms greatly benefit from this popularity and offer various services, such as providing online movie streaming services, managing subscription payments, and offering comprehensive customer support. One crucial service for retaining customers is **movie recommendation**. This process is driven by **data analysis** and powered by a **recommendation system** that utilizes high-quality data derived from these analytical processes.

The fundamental process that ensures high-quality data for data analytics and building models is **data engineering**. It focuses on efficient data processing and **data pipeline system maintenance**, ultimately leading to actionable, high-quality data.

This project utilizes the **TMDB (The Movie Database) dataset**, a free, community-driven platform providing detailed information about movies, TV shows, and cast members. It includes metadata such as genres, ratings, production details, and images. Developers can access this rich dataset via a public API, making it ideal for building recommendation systems and movie-related projects.

This project presents the architecture and data pipeline system utilizing tools within the **data engineering field**. For example, **Apache Airflow** for orchestrating and managing data pipelines, **Apache Spark** for processes and transforms the large TMDB dataset efficiently, and **BigQuery**, a data warehouse service from Google Cloud Platform (GCP), for storing high-quality data. This project demonstrates **the end-to-end data engineering process**. The outcome of this project is high-quality data prepared for data analytics and recommendation systems.

## Objective

1.**To design and implement a robust and scalable data architecture** that integrates all components (data ingestion, processing, storage, and serving) to support the end-to-end data pipeline.

2.**To build a data processing pipeline** that ensures high-quality data is extracted, transformed, and loaded into the data warehouse.

3.**To establish a scalable data warehouse** to house high-quality movie data, optimized to efficiently support a movie recommendation system.

## Output

The successful execution of this project, **TMDB-RecoFlow**, will yield the following key outputs:

* **A robust and scalable data pipeline**: This end-to-end pipeline, built with Apache Airflow for orchestration and Apache Spark for processing, ensures continuous data flow from ingestion to serving.
* **An optimized BigQuery data warehouse**: A well-structured central repository for high-quality, transformed movie metadata, designed for efficient data retrieval to support analytical queries and recommendation system training.
* **A high-quality dataset for recommendation**: The project will provide a clean, consistent, and readily accessible dataset specifically prepared to feed into a movie recommendation system for suggesting similar titles.
* **Demonstration of end-to-end data engineering**: This project will serve as a practical showcase of the complete data engineering lifecycle, from raw data acquisition to delivering actionable data for advanced analytics.

## Benefits

* **Enhanced Operational Efficiency**: The automated data pipeline, powered by Apache Airflow and Spark, streamlines data processing, reducing manual effort and ensuring timely data availability.
* **Actionable Business Insights**: The project delivers clean, structured data in BigQuery, enabling precise analytics and valuable insights into movie trends and user behavior.
* **Practical Skill Development**: For the developer, it provides invaluable hands-on experience in designing, implementing, and managing an end-to-end data engineering pipeline using industry-standard tools.

# Data Source

**Name:** Full TMDB Movies Dataset 2024 (1M Movies)

**Source:** <https://www.kaggle.com/datasets/asaniczka/tmdb-movies-dataset-2023-930k-movies/data>

**Data Characteristic:**

**The TMDB Movies Dataset (2023)** used in this project is a comprehensive and regularly updated collection of **film information**. It contains a vast number of movies, totaling **1,000,000 entries from the TMDB database**, with daily updates. Each entry includes essential details such as the movie's ID, Title, Average Vote, Vote Count, Status, Release Date, Revenue, and Runtime. Additionally**, the dataset features various other attributes that contribute to effective analysis and the development of robust movie recommendation systems.**

**Concern:**

* **Secondary Source**: The data is obtained from Kaggle, a secondary source, rather than directly from the TMDB API. This means the project relies on someone else's extraction and aggregation process, which might introduce unforeseen biases or limitations from their collection methodology.
* **Snapshot Nature**: The dataset is a static snapshot, containing data only up to (02-06-2025)

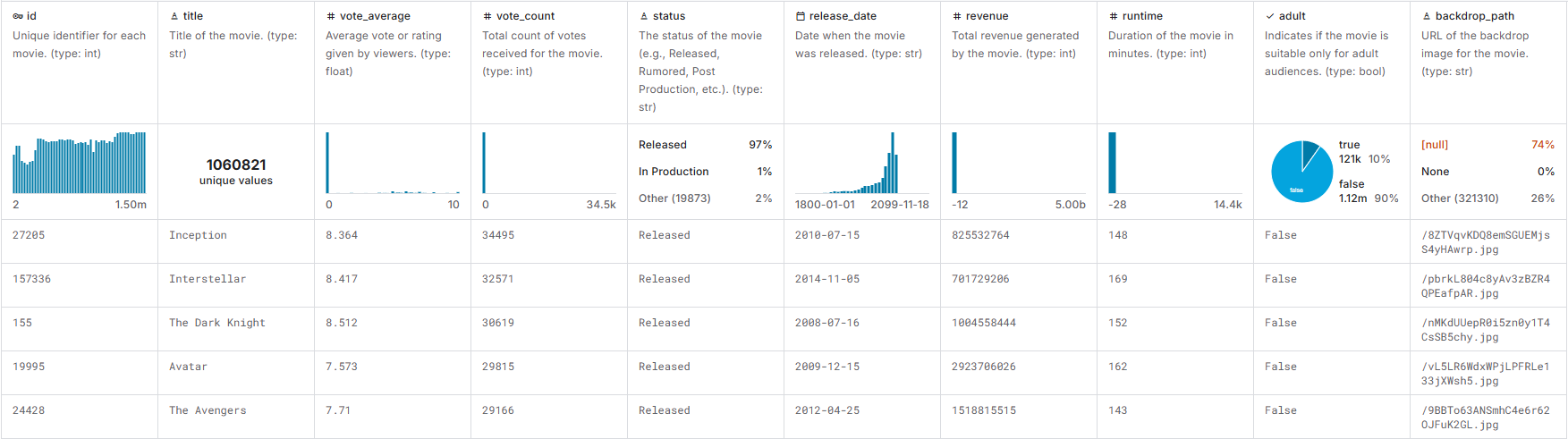
**Data Dictionary:**

The dataset exported from Kaggle contains 1,234,214 rows and 24 columns. This snapshot was created on 02‑06‑2025 (DD-MM-YYY)

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Description** |
| id | int | Unique identifier for each movie |
| title | str | Title of the movie |
| vote\_average | float | Average rating given by viewers |
| vote\_count | int | Total number of votes received |
| status | str | Production status (e.g., Released, Rumored) |
| release\_date | str | Date when the movie was/will be released |
| revenue | int | Total box office revenue |
| runtime | int | Duration of the movie in minutes |
| adult | bool | Whether the movie is categorized as adult-only |
| backdrop\_path | str | URL path to the movie's backdrop image |
| budget | int | Production budget of the movie |
| homepage | str | Official homepage URL for the movie |
| imdb\_id | str | IMDb identifier |
| original\_language | str | Original language code (e.g., en, fr) |
| original\_title | str | Original title of the movie |
| overview | str | Synopsis or description |
| popularity | float | Popularity score determined by TMDB |
| poster\_path | str | URL path to the movie's poster image |
| tagline | str | Official tagline or slogan |
| genres | str/list | List of genres (as strings) |
| production\_companies | str/list | List of production companies |
| production\_countries | str/list | List of production countries |
| spoken\_languages | str/list | List of languages spoken in the movie |
| keywords | str/list | List of associated keywords |

**Column Classification in Dataset**

|  |  |  |
| --- | --- | --- |
| **Numeric** | **Categorical (incl. bool/list)** | **Other/String/ID** |
| vote\_average | status | id |
| vote\_count | adult | title |
| revenue | original\_language | release\_date |
| runtime | genres | backdrop\_path |
| budget | production\_companies | homepage |
| popularity | production\_countries | imdb\_id |
|  | spoken\_languages | original\_title |
|  | keywords | overview |
|  |  | poster\_path |
|  |  | tagline |

****

**Figure 1: Sample data from Guest Gale**

# Architecture overview

This project implements a containerized **ETL (Extract, Transform, Load)** pipeline to prepare data for analytics and recommendation services. The system is built using **Docker Compose** to orchestrate multiple services including Airflow, Spark, and Big -Query integration. Below is a breakdown of the architecture components and their roles in the pipeline.

A screenshot of a phone

AI-generated content may be incorrect.

Figure

|  |  |
| --- | --- |
| **Component** | **Version/Tag** |
| Apache Airflow | apache/airflow:2.10.5-python3.12 |
| Apache Spark | bitnami/spark:3.5.2 |
| JupyterLab | jupyter/pyspark-notebook:spark-3.5.0 |
| Docker Engine | Platform base |

**Pipeline Process**

**Data Source**

* **Format**: CSV files (e.g., TMDB Movie Dataset)
* **Role**: Acts as the raw dataset to be ingested.

**Data Ingestion**

* **Tool**: Apache Airflow
* **Containerized**: Yes (Docker)
* **Description**: Airflow DAGs orchestrate the loading of raw CSV datasets from external or local sources into the data pipeline. It automates scheduled ingestion and tracks lineage.

**Data Processing**

* **Tool**: Apache Spark (Bitnami image)
* **Containerized**: Yes (Docker)
* **Description**: Spark processes the ingested raw data by performing transformation, cleaning, and enrichment. The output is a cleaned dataset (CSV format) ready for warehousing.

**Data Warehouse**

* **Platform**: Google BigQuery
* **Zones**:
  + **Staging Area / Landing Zone**: Temporarily holds transformed datasets before final loading.
  + **Cleaned Dataset**: Final structured dataset is loaded into BigQuery for analysis and serving.

**Data Service**

* **Query Layer**: BigQuery provides SQL-based analytics queries over the cleaned dataset.
* **Recommendation System Interface**: Cleaned data is used to develop ML-based recommendation systems that query processed features like genres, companies, and languages.

**Gantt Chart Plan (June–July 2025)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Task** | **Week 1** | **Week 2** | **Week 3** | **Week 4** | **Week 5** | **Week 6** | **Week 7** | **Week 8** |
| **Phase 1: Design & Plan (June 2025)** |  |  |  |  |  |  |  |  |
| Learn Spark & BigQuery |  |  |  |  |  |  |  |  |
| Lab Implementation (Component Setup) |  |  |  |  |  |  |  |  |
| System Architecture Design |  |  |  |  |  |  |  |  |
| Phase 1 Report Writing |  |  |  |  |  |  |  |  |
| **Phase 2: Implementation (July 2025)** |  |  |  |  |  |  |  |  |
| Component Implementation |  |  |  |  |  |  |  |  |
| Integration Testing |  |  |  |  |  |  |  |  |
| System Demo Preparation |  |  |  |  |  |  |  |  |
| Final Presentation & Report |  |  |  |  |  |  |  |  |