CS432/532: Final Project Report

**Project Title: TMDB Movie Data Analysis**

**Team Member(s): Parnasi Chandrakant Meher[B01027641]; Mansi Srinivas[B01035912]**

# PROBLEM STATEMENT

The film industry is vast and diverse, with countless movies being produced and released every year. However, making informed decisions about which movies to produce, distribute, or watch can be challenging due to the sheer volume of available data. This project aims to address the issue by providing valuable insights and recommendations based on in-depth analysis of the TMDB Movies Dataset 2023.

The goal is to analyze a comprehensive movie dataset to extract valuable insights that will benefit movie producers, distributors, and audiences alike. The primary objective is to develop analytical tools and recommendation systems to enhance decision-making processes in the movie industry.

# TECHNOLOGY USED

1. **MongoDB**: MongoDB is a widely used NoSQL database management system known for its flexibility, scalability, and performance. Unlike traditional relational databases, MongoDB stores data in flexible, JSON-like documents, allowing for dynamic schema structures. MongoDB Compass is a graphical user interface (GUI) tool that provides a visual interface for working with MongoDB databases, making it easier to interact with and manage MongoDB data.
2. **TMDB Movie Dataset**: The TMDB (The Movie Database) dataset contains extensive information about movies, including attributes such as "overview," "release date," "production company," and more. With over 1,000,000 movies from the TMDB database, this dataset offers a rich source of movie-related data for analysis and exploration.

**Dataset\_Link**: https://www.kaggle.com/datasets/asaniczka/tmdb-movies-dataset-2023-930k-movies

1. **Uploading Dataset to MongoDB**: The TMDB movie dataset is uploaded into MongoDB, where each movie is represented as a document in a MongoDB collection. This allows for efficient storage and retrieval of movie data using MongoDB's document-oriented approach. NoSQL queries are used to interact with the MongoDB database, leveraging various libraries in Python such as PyMongo to perform CRUD (Create, Read, Update, Delete) operations and execute aggregation pipelines.
2. **Backend and Frontend Connection:** Flask APIs (Application Programming Interfaces) are used to establish the connection between the backend and frontend of the application. Flask is a lightweight web framework for Python that enables the development of web applications with ease. The backend, powered by Flask, handles incoming requests from the frontend, processes data, and sends back responses. Meanwhile, the frontend of the application is built using HTML for structure and JavaScript for dynamic behavior and visualization.
3. **Pipelines:** MongoDB aggregation pipelines are utilized to perform complex data processing and analysis tasks directly within the database. Aggregation pipelines consist of a sequence of stages, where each stage performs a specific operation on the input documents. In the context of the TMDB movie dataset, pipelines can be constructed to filter, transform, group, and aggregate movie data based on various criteria such as release year, genre, language, and more. These pipelines enable the extraction of valuable insights from the dataset, which can then be presented to users through the frontend of the application for visualization and exploration.

# N+1 nosql queries

**Query 1:**

The first task of the project involves analyzing movie data to rank production countries based on the popularity of movies released in a selected year and belonging to a specified genre. This task begins by allowing users to select a year and a genre. The application then retrieves movie data for the chosen year, predicts the genre for each movie based on its overview using predefined keyword lists, and filters out movies that match the selected genre. Next, the application ranks the production countries based on the cumulative popularity of movies they produced in the specified genre for the selected year. Finally, it presents the top five production countries along with their cumulative popularity scores and provides a detailed breakdown of movies produced by each country, including their titles, popularity, production companies, and ranks.

**Query 2:**

The second task of the project focuses on exploring the evolution of genre preferences among audiences speaking different languages over time using a movie dataset. Users select a specific year, and the application retrieves movies released in that year. For each movie, the application predicts its genre based on the movie overview and predefined keyword lists associated with various genres. The predicted genres are then grouped based on the languages spoken in the movies. Movies with multiple spoken languages are listed under each language. Finally, the movies within each language group are sorted based on their popularity, and the top-ranked movies are presented to users, providing insights into the most popular genres among different language-speaking audiences for the selected year.

**Query 3:**

The third task of the project involves retrieving movie data based on a user-selected genre keyword and analyzing the profitability of movies belonging to that genre. Users input a genre keyword, and the application queries the dataset to find movies whose overview contains keywords related to the selected genre, and which have both revenue and budget values greater than zero. The application then calculates the profitability of each movie by subtracting its budget from its revenue and sorts the movies based on profitability. Finally, it presents the top five most profitable movies in the selected genre, including their titles and profitability values, to the user.

# Project Outcome

Query 1:

The images below show the first task, here the movies and the production companies are ranked for each country based on a particular year selected by the user. It also includes a bar graph that shows the production countries vs. Popularity.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Query 2:

The images below show analysis task two, here the final output is displayed in the form of a table. Once the genre of the movies is predicted, the ranks are generated for them based on popularity and finally the tables with the languages are created by splitting the languages for each movie and listing the movies under them along with genre and rank.

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

Query 3:

Below is the image for query 3it is bar graph where the x-axis represents the different movies, and the y-axis represents the profit made by the movie.

Here we see that “Lion King” made the most profit followed “Lil Detective” and so on.

A screenshot of a graph

Description automatically generated

##### References

[1] <https://www.mongodb.com/docs/manual/core/aggregation-pipeline/>

[2] <https://flask.palletsprojects.com/en/latest/api/>

[3] <https://www.kaggle.com/datasets?search=No>

[4]<https://huggingface.co/datasets?sort=trending&search=ecommerce>

[5] <https://www.w3schools.com/html/>