Milestone 1 Report

Project Title: Movie Recommendation Engine

Course: Introduction to Data Science

Student: Ashfaq Ahmed Mohammed (UFID: 86835927)

Section 1: Objective, Tools, and Datasets Used

1.1 Project Objective

This project aims to develop a Movie Recommendation Engine that suggests movies to users based on their preferences. The recommendation system will leverage user ratings, movie metadata, and other relevant attributes to generate personalized recommendations.

1.2 Tools and Libraries Used

Data Handling and Processing

• pandas – for handling datasets (loading, cleaning, and manipulating data).
• numpy – for numerical operations and array processing.

Data Visualization

• matplotlib – for plotting graphs and charts.
• seaborn – for statistical data visualization.

Machine Learning and Data Preprocessing

• scikit-learn – for building and evaluating recommendation models.
• MinMaxScaler (from sklearn.preprocessing) – for normalizing budget and revenue values to bring them within a standard range.
• nltk and Word2Vec

File Handling and Utilities

• os – for interacting with the file system and handling dataset directories.

1.3 Datasets Used

The project utilizes multiple datasets for building the recommendation engine:

Dataset Name	File Name	Description
The Movies Dataset (The Movies Dataset)	movies_metadata.csv ratings.csv links.csv credits.csv	Contains detailed information about movies such as title, genres, budget, revenue, runtime, and popularity. user ratings for movies. Links movies across different platforms such as IMDb and TMDb for cross-referencing Consists of Cast and Crew Information of the movies.

Streaming Platforms (Streaming Platforms)	MoviesOnStreamingPlatforms.csv	Contains information on whether a movie is available on Netflix, Hulu, Prime Video, or Disney+.
Top 1000 Highest grossing Movies (Highest Grossers per Year)	Top_1000_Highest_Grossing_Movies_Of_All_Time.csv	Contains information on the highest-grossing movies of all time.

1.4 Dataset Dimensions

CSV File	Rows (Approx.)	Columns	Description
movies_metadata.csv	45000	24	Contains metadata about movies including title, release date, genres, etc.
credits.csv	45000	3	Includes cast and crew details for each movie in JSON format.
ratings.csv	Millions	4	Contains user ratings for movies with user IDs, ratings, and timestamps.
links.csv	45000	3	Maps movie IDs to IMDb and TMDb identifiers.
highest_grossing_movies.csv	1000	6	Contains data on the top 1000 highest-grossing movies of all time, including title, year, and worldwide gross.
movies_on_streaming_platforms.csv	16000	12	Contains information about movies available on Netflix, Prime Video, Hulu, and Disney+, including title, year, and platform availability.

1.5 Project Overview, Enhancing Recommendations with Real-Time Ratings & Streaming Data

The user selects a few films of their liking from a list of films displayed, after this based on the user's taste the model will recommend similar movies. The movies along with their attributes (Genre, Budget, Revenue, Director, rating etc.) will be embedded in a vector space, we will perform clustering in the vector space and return the movies that are close to the movies selected by the user.

To make the model more practical I'm also taking into account additional datasets called the "Top 10,000 Highest-Grossing Movies" and "Streaming Platforms". The idea is to use the data from these datasets to make the recommendation engine more natural, Highest grossing movies are generally well-received by the public, so this data will impart a bias to the engine.

Incorporating Real-Time User Ratings

The ratings.csv dataset, which contains real-time user ratings for movies, was merged with the movies_metadata.csv dataset using the movieId and IMDB ID as linking attributes. This integration allows the recommendation engine to dynamically update its suggestions based on audience preferences.

Incorporating Streaming Service Availability

The whole idea of recommending movies is to watch them, So, the “Streaming platforms” data set will be used to recommend the user movies that can be watched readily.

The availability of movies on popular streaming platforms such as Netflix, Prime Video, Hulu, and Disney+ will be incorporated into the recommendation system. To improve relevance, movies available on these platforms will be given a positive bias in recommendations.

- Improves Accessibility: People are more inclined to watch films that are easily accessible on the streaming services to which they have a subscription.

- Promotes Platform-Specific Suggestions: helps customise recommendations according to the user's access to various streaming services.

- Enhances the Suggestion Accuracy: The system guarantees useful and actionable recommendations by giving priority to films that are easily accessible.

1.6 Project Plan

S.no	**Task**	**Deadline**
1.	Data collection and Preprocessing	23rd February 2025
2.	Exploratory Data Analysis	23rd February 2025
3.	Feature Engineering and Feature selection	11th March 2025
4.	Data Modeling	21st March 2025
5.	Evaluation and Testing	15th April 2025
6.	Interpreting and visualizing results	23rd April 2025

Section 2: Data Dictionary

Movies Metadata

Attribute	Data Type	Description
id	Integer	Unique movie identifier.
title	String	Movie title.
original_language	String	Original language of the movie.
genres	String (JSON)	List of genres associated with the movie.
release_date	Date	The release date of the movie.

budget	Integer	Budget of the movie (in USD).
revenue	Integer	Revenue earned by the movie (in USD).
runtime	Float	Duration of the movie in minutes.
popularity	Float	Popularity score based on user interactions.
vote_average	Float	Average user rating (out of 10).
vote_count	Integer	Number of users who rated the movie.

Ratings Data

Attribute	Data Type	Description
userId	Integer	Unique identifier for the user.
movieId	Integer	Unique identifier for the movie.
rating	Float	User rating given to the movie (scale of 0 to 5).
timestamp	Integer	Timestamp of when the rating was given.

Movie Links

Attribute	Data Type	Description
movieId	Integer	Unique movie identifier (same as in ratings.csv).
imdbId	String	Corresponding IMDb ID.
tmdbId	String	Corresponding TMDb ID.

Streaming Platforms

Attribute	Data Type	Description
ID	Integer	Unique identifier for the movie.
Title	String	Movie title.
Netflix	Boolean (0/1)	Indicates if the movie is available on Netflix.

Hulu	Boolean (0/1)	Indicates if the movie is available on Hulu.
Prime Video	Boolean (0/1)	Indicates if the movie is available on Prime Video.
Disney+	Boolean (0/1)	Indicates if the movie is available on Disney+.

Top 10,000 Highest-Grossing Movies

Attribute	Data Type	Description
Rank	Integer	Rank of the movie based on revenue.
Title	String	Movie title.
Revenue	Integer	Total revenue earned (in USD).
Budget	Integer	Production budget (in USD).
Profit	Integer	Profit calculated as Revenue - Budget.
Release Year	Integer	Year the movie was released.
Genres	String	List of genres associated with the movie.
Director	String	Name of the director.
IMDB Rating	Float	Average IMDb user rating (out of 10).
Votes	Integer	Total number of votes on IMDb.

Credits

Attribute	Data Type	Description
Cast	String (JSON)	It has a list of cast members in that particular movie; cast members include directors and actors.
Crew	String (JSON)	It has a list of the crew that worked on the movie
id	Integer	This is the movie ID that maps to the id column of the movies Metadata Dataset.

Section 3: Handling Missing Values

3.1 Identifying Missing Values

During data preprocessing, the dataset was analyzed for missing values using the `.isnull().sum()` function. I counted how many missing values were in each column. Then, I came up with strategies to deal with these gaps in the data.

3.2.1 Filling Categorical Columns with Mode

For categorical variables where missing values were rare, the mode (most frequent value) was used to fill in missing entries. This ensures consistency across the dataset.

Column	Filling Strategy	Reason
original_language	Filled with mode (most frequent language)	Most movies are in a few dominant languages, making mode a logical choice.
production_companies	Filled with mode	Large production companies dominate the dataset.
production_countries	Filled with mode	Most movies are produced in a handful of countries.

3.2.2 Filling Numerical Columns with Mean or Median

For numerical columns, the mean or median was chosen based on the distribution of the data:

Column	Filling Strategy	Reason
revenue	Filled with mean	Revenue data is skewed but follows a normal trend.
runtime	Filled with median	Some movies have extreme runtimes, making median more robust.
popularity	Filled with median	Popularity scores contain extreme outliers, making median the better choice.
vote_count	Filled with median	Vote counts have a long-tail distribution, making median more appropriate.
vote_average	Filled with mean	Movie ratings follow a normal distribution, so mean is a good estimate.

Section 4: Data Normalization and Outlier Handling

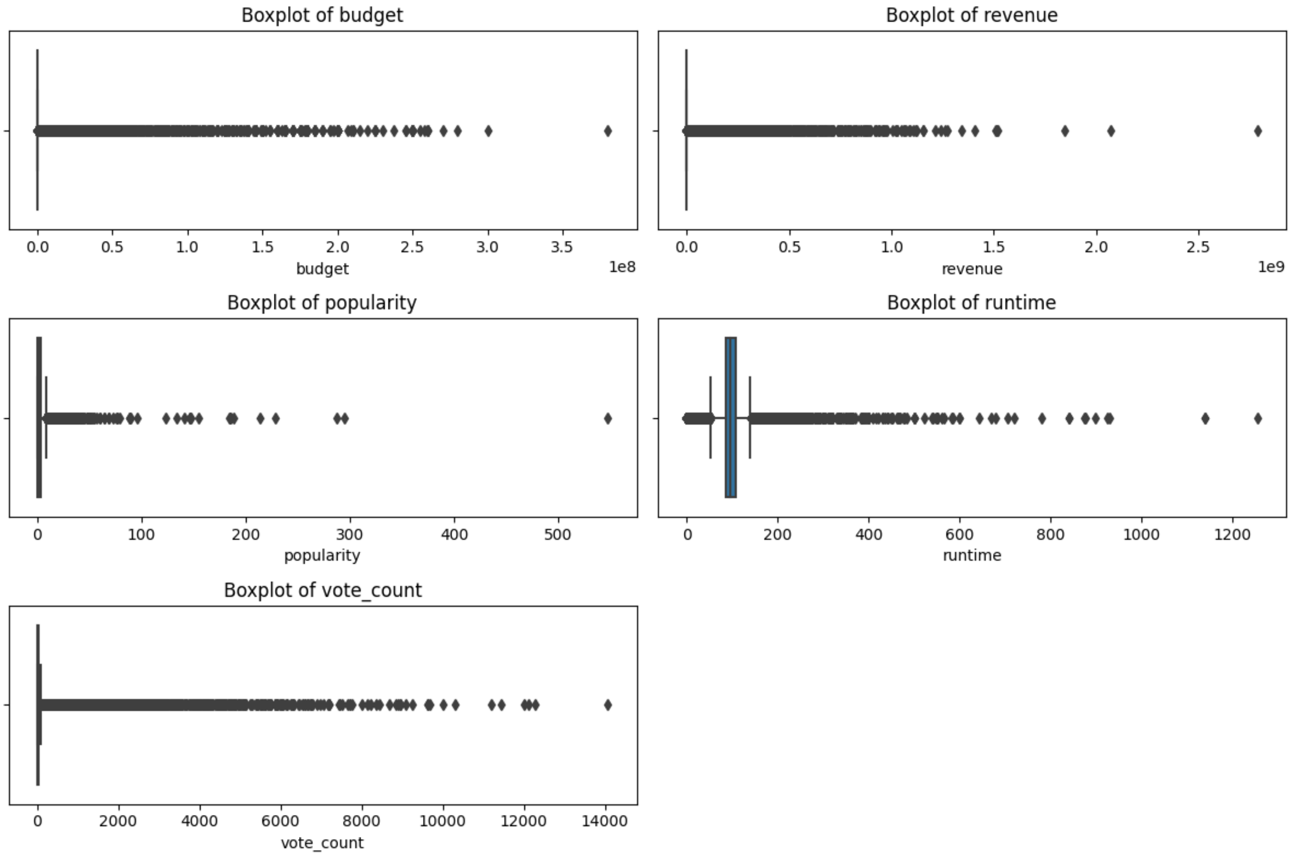
Section 4.1 Data Normalization

To ensure numerical features are on a consistent scale, MinMaxScaler was applied to normalize the following attributes:

- Budget
- Revenue
- Popularity
- Runtime
- Vote Average

Normalisation limits all numeric attributes to the range of 0 to 1, so attributes with larger magnitudes (for example, budget, revenue) do not dominate the analysis. This improves the model's performance, increases interpretability in correlation analysis, and maintains equal weighting for various numeric attributes.

Section 4.2 Outlier Handling



Budget & Revenue:

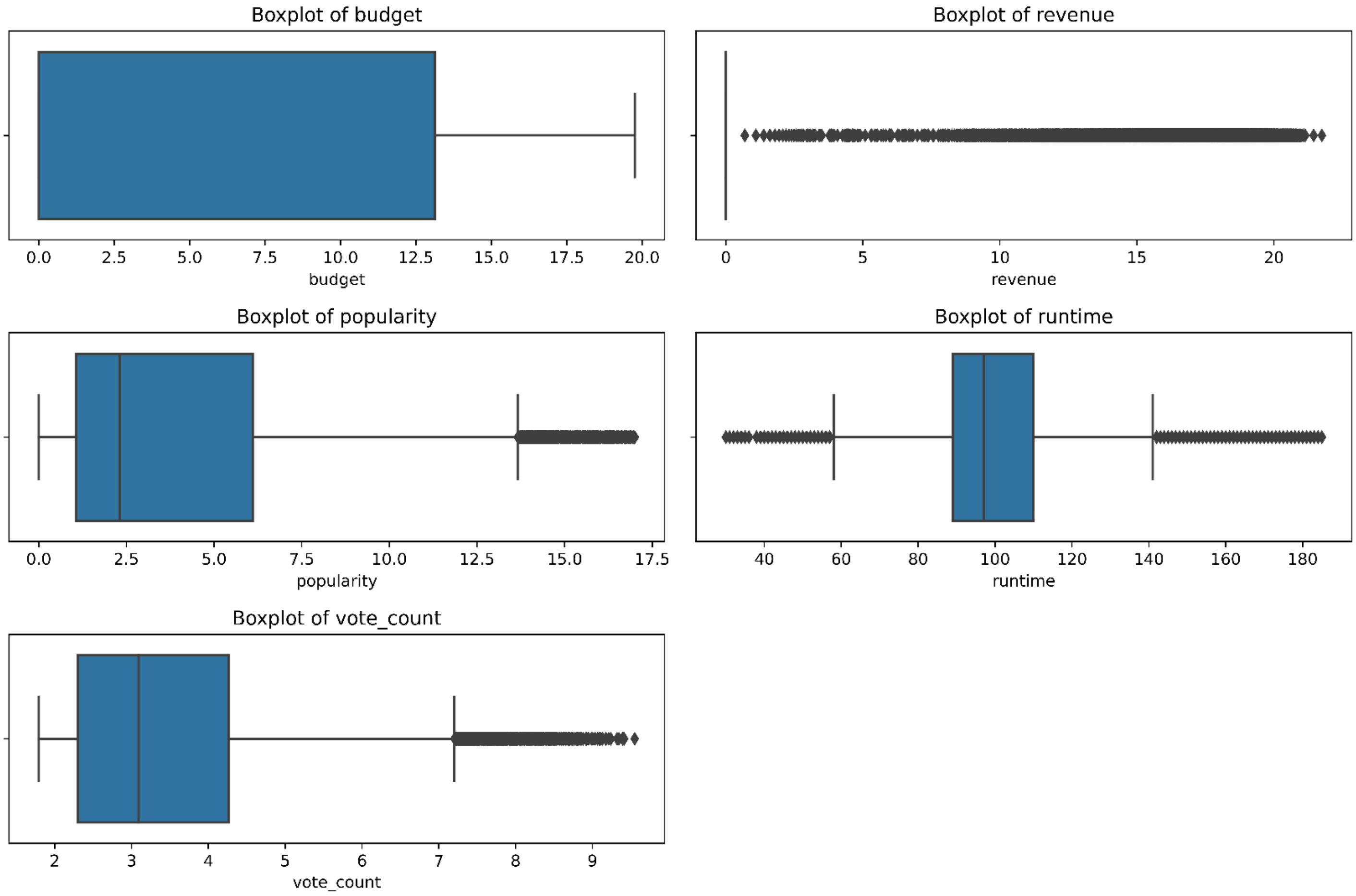
Both have a long tail on the upper end, suggesting that a small number of movies have extremely high budgets and revenues. Most values are concentrated towards the lower end. So, I performed a log transformation on the Budget and Revenue values to prevent the large outlier from dominating the model.

Popularity & Vote Count:

The distribution is skewed with a few movies having extremely high popularity scores and vote counts. I performed the same log transformation on the popularity and vote count values while also capping the values at their 99th percentile to avoid the large values dominating the model.

Runtime:

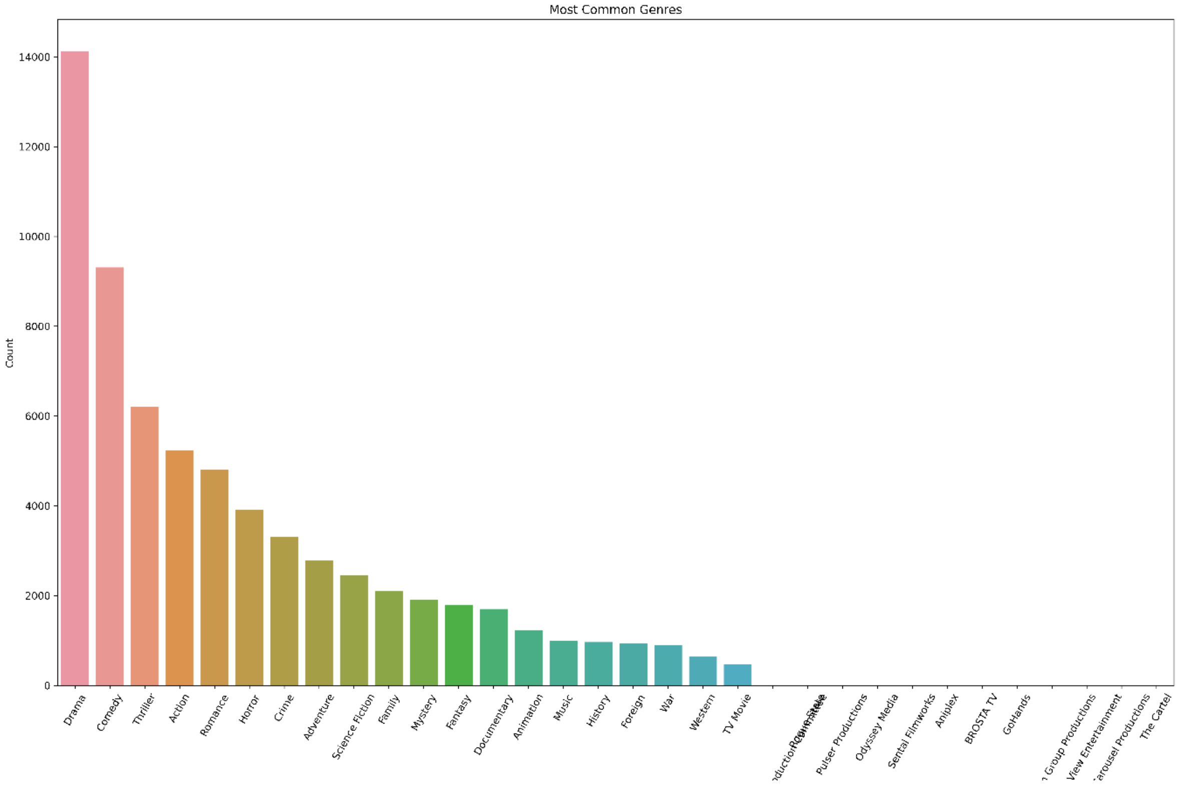
Most movies have a runtime between a specific range (likely around 80-120 minutes). However, some extreme outliers extend beyond 600-1200 minutes, which might indicate errors or special cases (e.g., series mistakenly classified as movies). I also removed movies with a runtime of less than 30 minutes (likely short films or incorrect entries).



The figure above shows the boxplots of the attributes after applying the log transformation, as we can see the data appears considerably less skewed. However, the "Revenue" column is still skewed, which means that the data in this column is highly skewed.

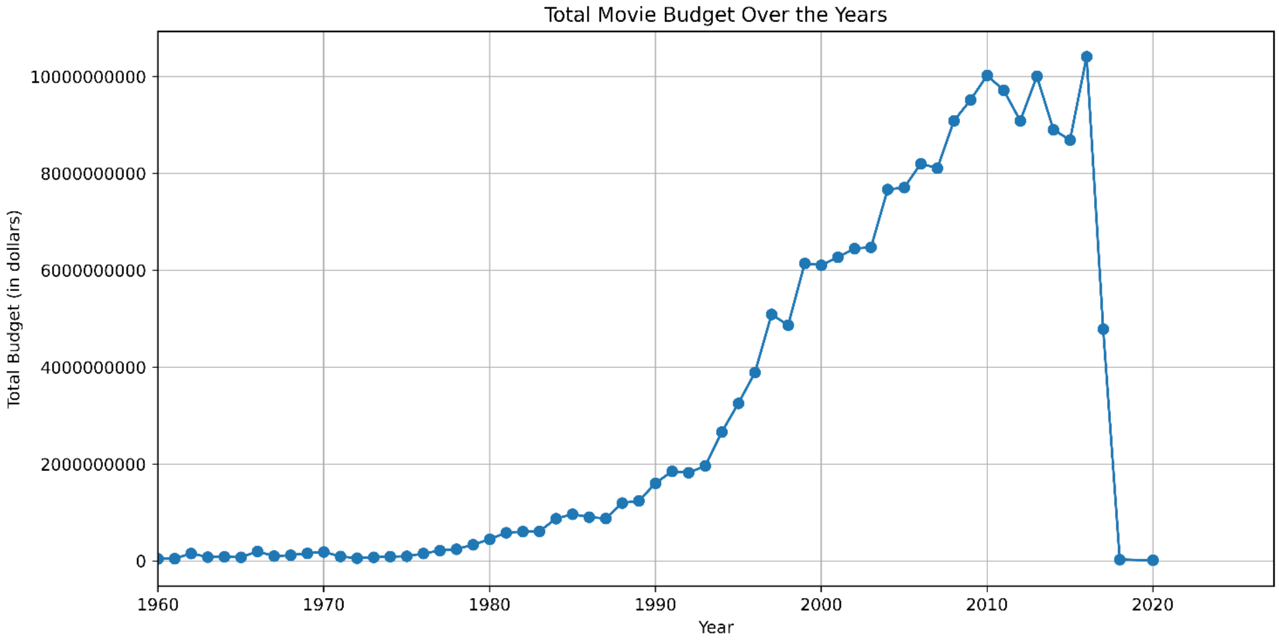
Section 5: Exploratory Data Analysis

Section 5.1: Most Common Genres (Bar Graph)



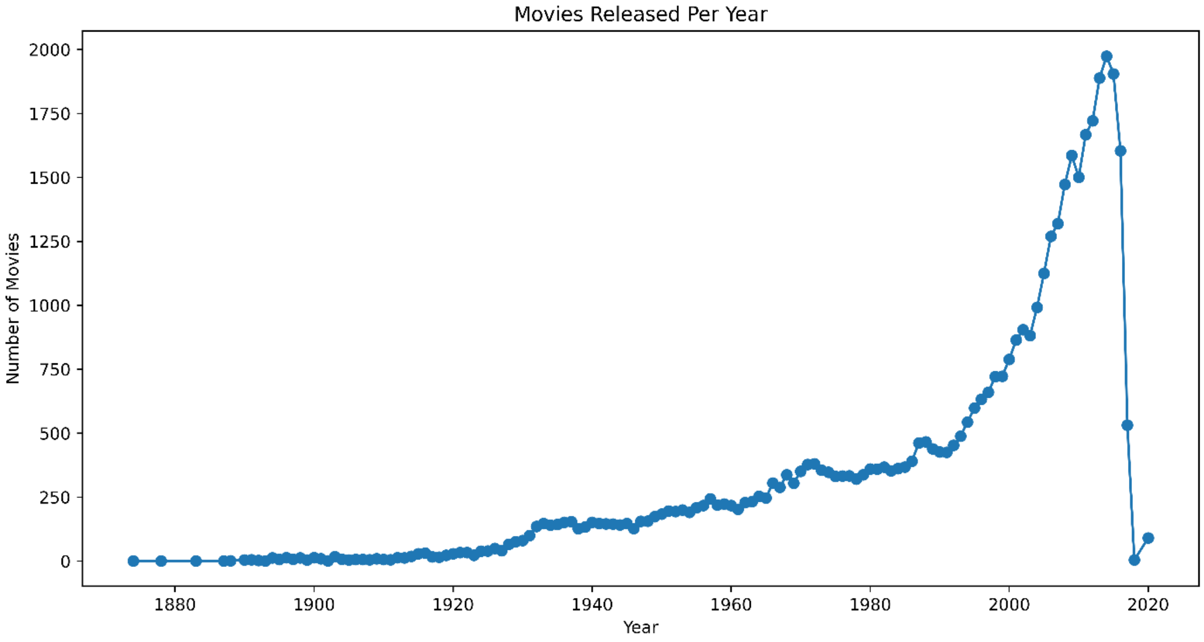
This bar graph represents the most common genres in the dataset. Drama is the most dominant genre, followed by Comedy and Thriller. This skewed distribution implies that recommended movies might be biased toward these popular genres. Data balancing techniques like oversampling underrepresented genres or applying weighted loss functions might be required.

Section 5.2: Movie Budget (line Graph)



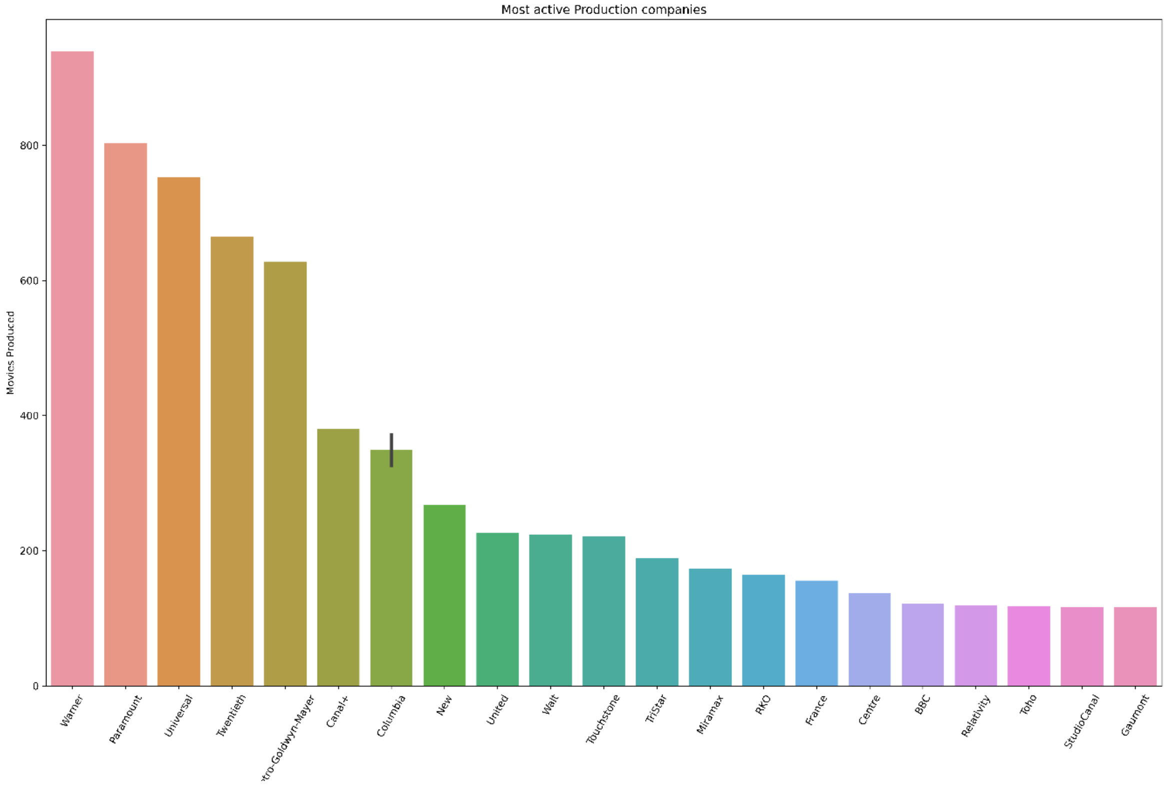
Up until the late 2010s, the trend in film budgets showed exponential growth after that, there was a sharp decline. The recommendation engine may use this information to suggest films that were produced in the last few decades because, presumably, the quality of films has improved as the overall budget for film production has increased over the past few decades.

Section 5.3: Movies Released per year



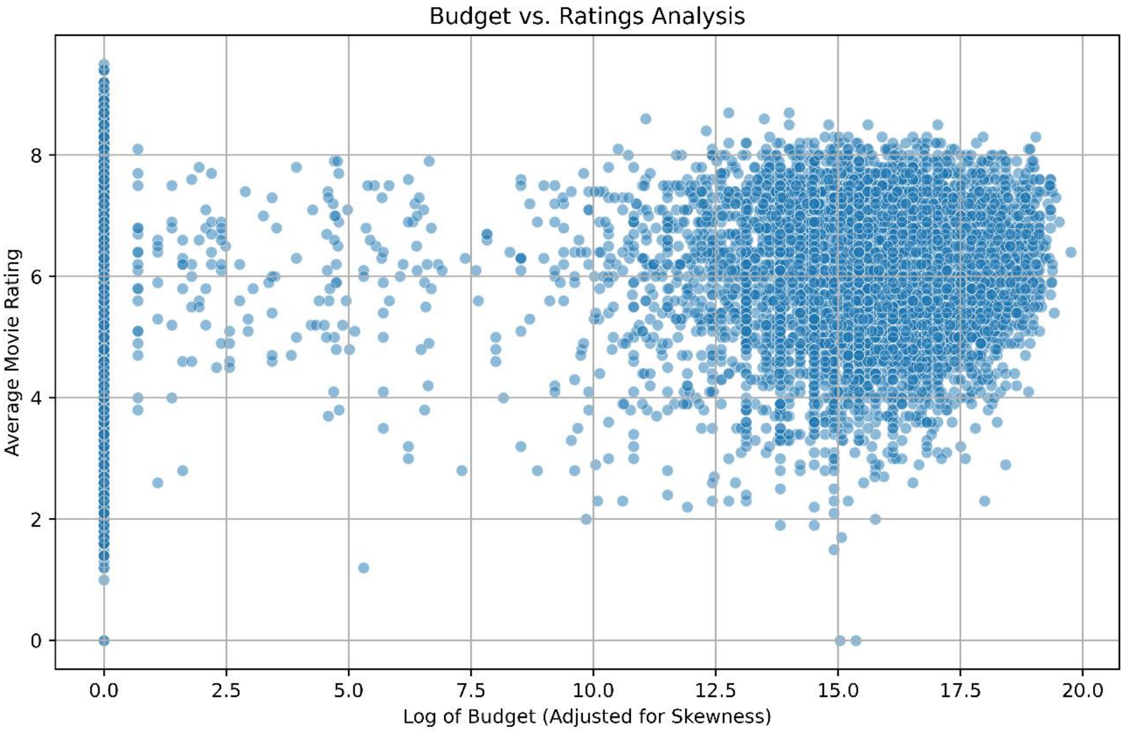
The trend of movies released per year follows a similar trajectory to movie budgets, with a steep rise in the 2000s and a drastic fall in recent years. This suggests that factors affecting movie releases are industry-wide. So, the recommendation engine will be biased towards movies released in recent years, to compensate for this a positive reinforcement bias has to be imparted for old movies that are genre-defining.

Section 5.4: Production Activity



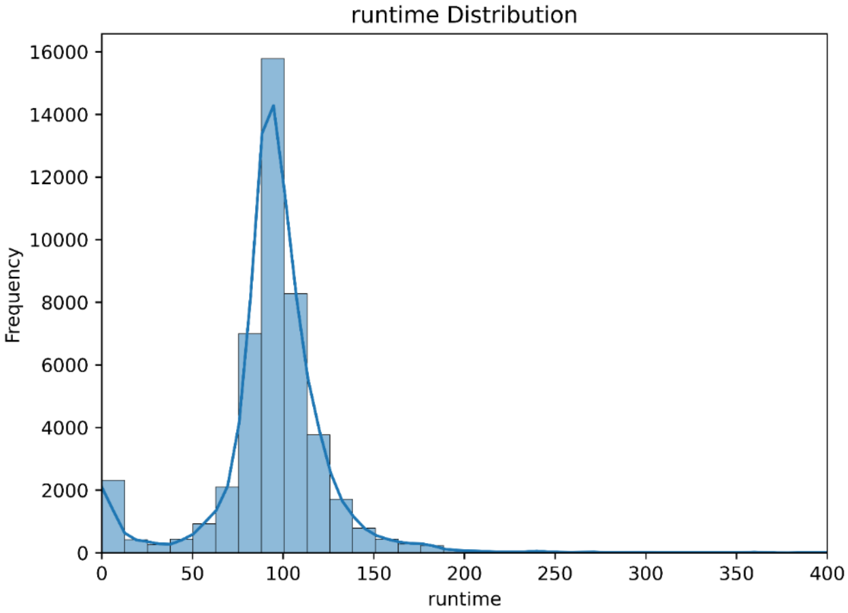
The production activity chart highlights the dominance of major studios like Warner Bros, Paramount, and Universal. Smaller studios may focus on niche genres or lower-budget films, and this insight can be leveraged for more targeted recommendations.

Section 5.5: Budget Vs Ratings



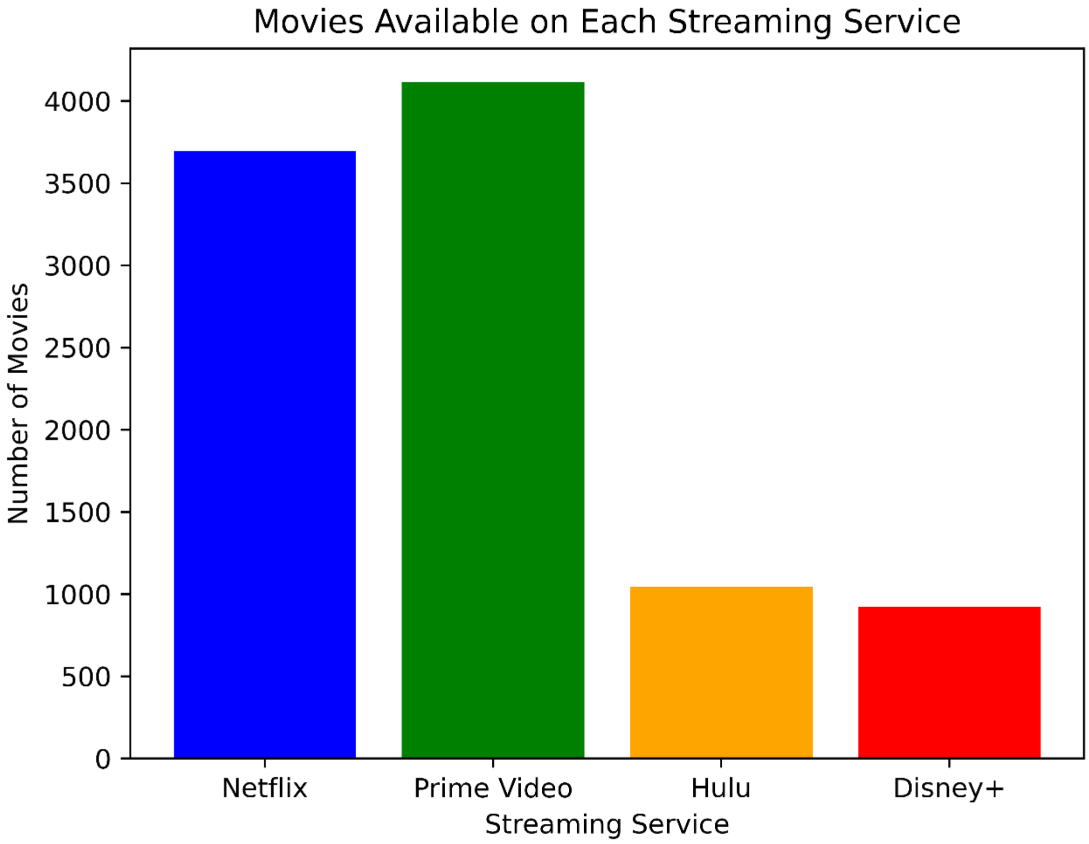
The scatter plot illustrates the correlation between average movie rating and log-transformed budget. There is no apparent connection between the data points, suggesting that better ratings are not always correlated with larger budgets. There are many films in the mid-to-high budget range, most of which have ratings in the 5–8 range. The plot emphasises that low-budget films can still earn high ratings, proving that audience reaction is not solely influenced by budget.

Section 5.6: Runtime Distribibuition



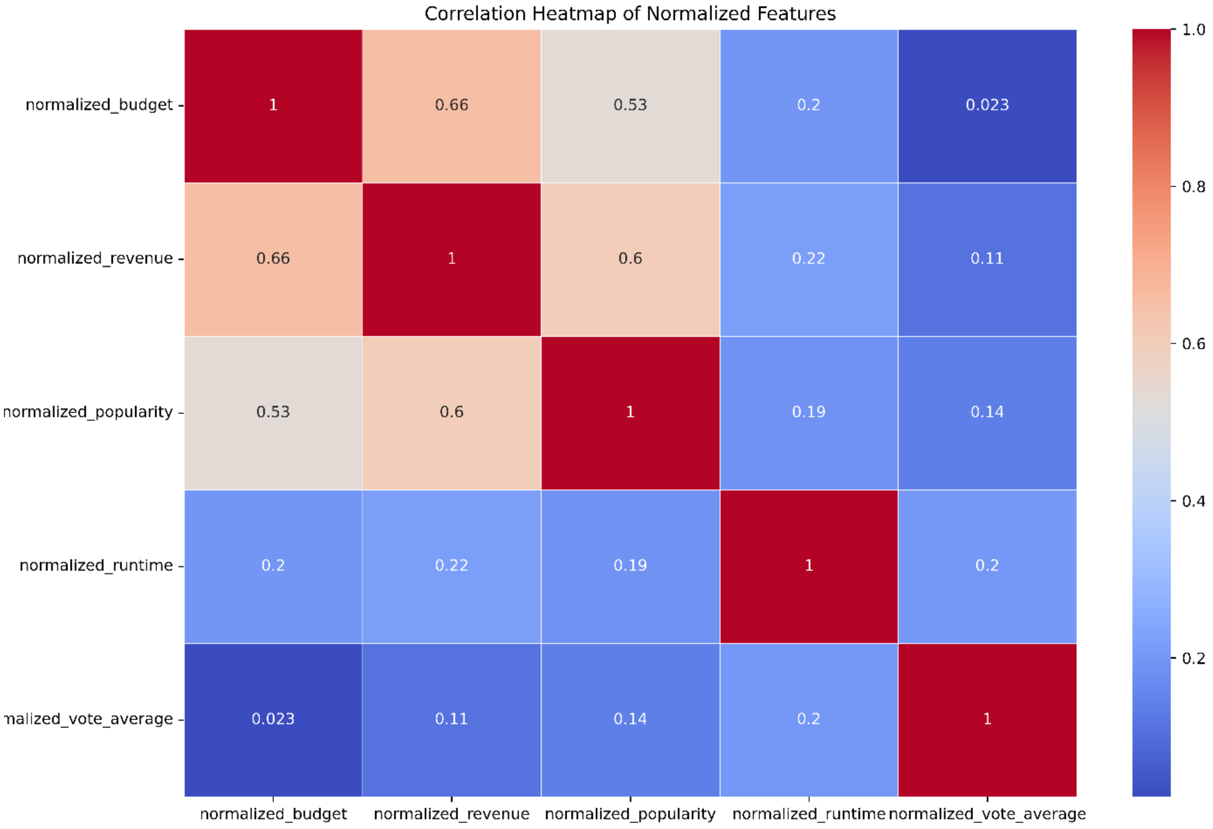
The runtime distribution is right-skewed, indicating that most movies fall within the 80-120-minute range, while exceptionally long movies are rare. Excessively long runtimes do not necessarily mean the movie is good in the same way excessively short runtimes also mean the same. So, to tackle this issue movie runtime columns can be binned into categories (short, standard, long) that could improve predictive accuracy.

Section 5.7: Movies Streaming



This bar chart provides insights into the distribution of movies across streaming platforms. Prime Video and Netflix dominate the industry, offering the largest catalogs, while Hulu and Disney+ have fewer movies. For a recommendation system, this data can help optimize content suggestions by factoring in platform availability.

Section 5.8: Correlation Matrix



The correlation heatmap provides valuable insights into relationships between features. A strong positive correlation between budget and revenue indicates that higher-budget movies tend to generate more revenue. Similarly, the correlation between popularity and revenue suggests that widely talked about movies perform better at the box office.

However, the vote average shows almost no correlation with budget or revenue, implying that expensive movies do not necessarily receive higher ratings. These insights can help in feature selection, allowing models to prioritize high-impact variables such as budget and popularity over weak predictors like vote average.

Section 5.9: Meta-Data

I wanted to combine two important datasets one with general movie details and another with cast and crew information to create a more complete dataset for building a recommendation system. To do this, I first cleaned the movie ID column by converting it to a consistent format and removing any missing values. This was important to ensure that the data matched correctly when merging the two datasets.

I then merged them using a left join, so that all movies from the cast and crew dataset were kept, along with any available details from the movie details dataset. After merging, I selected only the most relevant columns, such as budget, revenue, popularity, runtime, ratings, production companies, genres, lead actors, and directors. This step was crucial because these features will help the recommendation system understand different aspects of movies, allowing it to make better suggestions based on user preferences.

- Director
- Lead Actor
- Original Language
- Genre
- Production House
- Title

The above-mentioned attributes are Text based categorical attributes that need to be converted into numerical form in order to be used. From the application point of view these attributes are very important to establish similarities between movies, for instance a director is known for making certain types of films (Martin Scorsese and Gangster movies for example) So a user who liked one of his movies has a very high probability of liking another movie from the same director. So, as part of Milestone 2 Feature Engineering, I plan on vectorizing these attributes and using them in building the recommendation engine.

Section 6 – Summary of the EDA

The Exploratory Data Analysis (EDA) provided key insights into the movie dataset, helping to understand the trends and distributions of various attributes. The following analyses were conducted:

- **Genre Distribution:** Drama and Comedy dominate the dataset, which may bias recommendations.
- **Popularity & Vote Count Analysis:** A few movies are highly popular and receive high votes, while most remain obscure.
- **Runtime Distribution:** Most movies are between 80-120 minutes, with some outliers exceeding 200 minutes.
- **Movies Released Per Year:** The number of movies has increased significantly since the 1990s, peaking around 2018-2019.
- **Budget and Revenue Trends:** Higher-budget movies generally generate higher revenue, but exceptions exist.
- **Budget vs Ratings:** There is no apparent correlation between budget and ratings, high budget movies do not necessarily translate to better ratings.
- **Streaming Service Availability:** Prime Video and Netflix have the most extensive movie libraries.
- **Correlation Analysis:** Budget and revenue are strongly correlated, while other factors like runtime have weak correlations.
- **Meta Data and Dataset Merging and Feature Selection:** I plan on encoding the text based categorical data (Through vector embedding) and utilizing them to train the model.