SIADS 696 Milestone II Project Report

Content Recommendation Using IMDB Dataset

Younghoon Oh (hooni@umich.edu), Eric Kim (erikkim@umich.edu), Shin Choo (shinchoo@umich.edu)

Introduction

In today's world, where a myriads of films and TV shows fight for attention, finding what truly matches our taste can be overwhelming. Traditional recommendation systems largely rely on explicit user interactions or collaborative filtering, often tend to fail with new users or fresh contents. Our project's vision is to forge a recommendation system driven by the semantic richness of movie plot summaries, metadata and user ratings from the IMDB/OMDB/MovieLens dataset, aiming to offer personalized suggestions even in data-scarce environments

In this project, we employed both supervised and unsupervised learning methodologies to build more accurate movie recommendation models.

- **Supervised Learning:** Our goal is to predict a movie's rating tier based on its metadata and plot summaries. By implementing a multi-class classification framework, we strive to identify any genre or era-related biases, thereby improving prediction accuracy through the integration of structured and unstructured data

- **Unsupervised Learning:** The point is to cluster films by plot similarity and selected metadata to support cold-start recommendations and thematic discovery. This methodology aims to discover new contents based on storyline and reduce reliance on user-driven biases

Through this project, we aim to enhance the accuracy of traditional recommendation models and also ensure the recommendations are insightful, driven by plot semantics rather than just casual dashboard metrics.

Data Source, Scope, Pre-processing

Our project's foundational backbone is built on a multipronged approach to data collection, utilizing several Movie information data sources to drive comprehensive analysis and model development. For details of each dataset, refer to Appendix B – Data Schema

IMDB Dataset

The IMDB open dataset serves as a cornerstone, providing extensive metadata about movies. This includes details like genre, director, runtime, and ratings, all critical for understanding the structural elements of films. This IMDB dataset offers:

- **Diverse Metadata:** Key attributes from IMDB csv includes tconst (a unique identifier for each title), startYear, primaryTitle, genres, runtimeMinutes, and director_names. These elements provide a multi-dimensional view of films, essential for both historical and contemporary analyses

- **Comprehensive Coverage:** Known as one of the most popular movie information sources, IMDB provides historical insights alongside contemporary movie data, making it indispensable for trend analysis and temporal studies

OMDB Dataset

The OMDB dataset augments our analysis with additional metadata, aligning closely with IMDB data but often providing extra contextual details. The OMDB dataset offers:

- **Narrative and Content Details:** OMDB csv enriches our dataset with attributes such as plot, language, awards, actors_omdb, and rated (PG-13, R etc.,). This allows for a deeper narrative analysis and content classification

- **Purchased Access:** Due to the high quality and structured nature of OMDB data, we made a purchase to access the dataset to ensure it provides more accurate and reliable information for our models

MovieLens Dataset

The MovieLens csv dataset introduces a layer of user interaction to our framework, critical for personalizing recommendations and understanding viewer preferences. The MovieLens dataset offers:

- **User Ratings:** Detailed entries that show how individual users (e.g., user_id1, user_id2) rate specific movies on a scale, offering insights into subjective viewer preferences and satisfaction

- **Community Insights:** As a dataset generated from a user community developed by GroupLens Research at the University of Minnesota, it provides real-world data on viewing habits and preferences, which are all crucial information for our models

Preprocessing

This project integrates three data sources—IMDb, OMDB, and MovieLens—for comprehensive analysis and modeling. The pre-processing pipeline consists of the following major steps:

- **IMDb Data Collection and Filtering :** We downloaded title.basics.tsv.gz and filtered for entries where titleType is "movie". From this subset, we selected only the columns tconst, primaryTitle, startYear, **runtimeMinutes, and genres. This dataset provides foundational metadata such as movie titles, genres, and runtime.**

- **OMDB Data Retrieval and Merging :** Using the tconst identifier, we queried the OMDB API to retrieve additional movie details such as plot, actors, language, country, and awards. The collected data was saved locally as omdb_merged_data.csv to avoid redundant API calls. We merged this data with the IMDb dataset using the shared imdbID key.

- **MovieLens Integration :** We extracted user rating information from ratings.csv and links.csv, calculating each movie’s average rating (avg_rating) and number of ratings (num_rating). This enriched rating data was then merged with the existing IMDb+OMDB dataset using imdbID.

- **Missing Value Handling and Filtering :** We removed rows with missing values in critical fields such as director, writer, actors, plot, language, country, avg_rating, and num_rating. After filtering, we selected only the necessary columns to construct the final dataset.

- **Dataset Freezing for Reproducibility :** Because the upstream data (especially OMDB) may change over time, we fixed the dataset to a snapshot containing 62,188 rows and saved it as df_final_frozen_62188.csv. This static version was used throughout the modeling process to ensure reproducibility.

Part A. Supervised Learning

1. Methods description

Step	Detail
Data Integration	Combining IMDB/OMDB metadata + MovieLens ratings
Feature Eng.	• Plot → BERT embedding (all-mpnet-base-v2, 768D)

	• Categorical(genres, actors, director, writer, country, language) → **multi-hot**
	• Numerical(runtimeMinutes, startYear, num_rating)
Imbalance Correction	Apply SMOTE (optional) to the training set
Learning Model	① **Random Forest** (Tree) ② **XGBoost** (Boosting) ③ **Logistic Regression** (Linear/Prob.) ④ **KNN** (Instance)
Hyperparameter	Using GridSearchCV + 5-fold CV (grid for each model)
Ensemble	Soft voting + probability averaging method (3 or 4 models)

In the supervised learning component of this project, we constructed a binary classification model to predict movie sentiment ratings (0-2: Negative, 3-5: Positive) using integrated datasets from IMDB, OMDB, and MovieLens. Our feature engineering incorporated:

- **Text Features:** Movie plot summaries were converted into 768-dimensional embedding vectors via the pre-trained BERT model all-mpnet-base-v2
- **Categorical Features:** genres, actors, director, writer, country, and language were encoded using a multi-hot encoding approach
- **Numeric Features:** Features such as runtimeMinutes, startYear, and num_rating were used either in their original form or standardized for better integration into the model.

To counteract class imbalance, we applied SMOTE (Synthetic Minority Oversampling Technique) selectively during training.

We evaluated the following four algorithms, tailored to utilize their strengths:

- **Random Forest:** An intuitive tree-based ensemble model adept at processing mixed categorical and numerical inputs
- **XGBoost:** A gradient model effective in minimizing overfitting and recognizing feature interactions.
- **Logistic Regression:** A straightforward linear classifier that is both fast and interpretable
- **K-Nearest Neighbors (KNN):** A distance-based model included for comparison with non-parametric methods

Hyperparameter tuning was conducted using GridSearchCV with five-fold cross-validation, and we explored ensemble models through soft voting and probability averaging to refine performance.

2. Supervised Evaluation

2-1. Full Results Report

Learning Model	CV Macro F1 (mean ± std)	Test Accuracy	Test Macro F1
Random Forest	0.7049 ± 0.0346	0.6682	0.6426
XGBoost	0.7107 ± 0.0429	**0.694**	**0.6721**
Logistic Regression	0.6709 ± 0.0040	0.6788	0.6536
KNN	0.6589 ± 0.0360	0.5978	0.5978
4-Model Ensemble	–	0.6788	0.6695
3-Model Ensemble (Remove KNN)	–	**0.6922**	0.6677

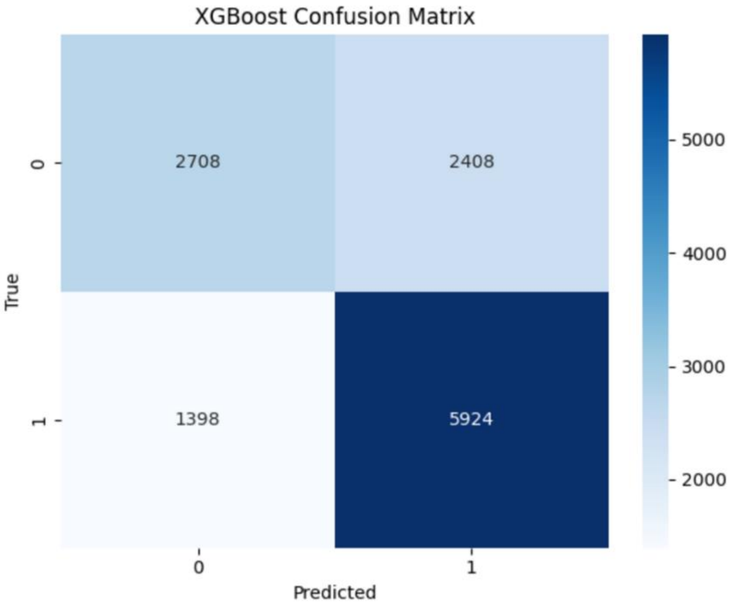
The performance of each model was evaluated based on accuracy and Macro F1-score on the test set. XGBoost demonstrated superior performance with an accuracy of 0.694 and an F1-score of 0.6721, whereas KNN underperformed relative to other models. The 3-model ensemble, excluding KNN, achieved an accuracy of 0.6922, showcasing that excluding less effective models can enhance overall performance

2-2. Feature Importance and Ablation Analysis

Strategy	Learning Model	SMOTE	Accuracy	F1-score
1st	RandomForest	Apply	0.6682	0.6426
	XGBoost	Apply	0.6865	0.6748
	LogisticRegression	Apply	0.6788	0.6461
	KNN	apply	0.5978	0.5978
2nd	4-model Ensemble	Apply	0.6788	0.6695
3rd	4-model Ensemble	Not apply	0.6897	0.6619
	XGBoost	Not apply	**0.694**	**0.6721**
4th	3-model Ensemble (Remove KNN)	Apply	**0.6922**	0.6677

In all experiments, the all-mpnet-base-v2 embeddings from SentenceTransformer were utilized as they performed the best compared to alternatives like TF-IDF. Experiments 2,3, and 4 incorporated hyperparameter tuning and ensemble strategies, which were crucial for optimizing performance

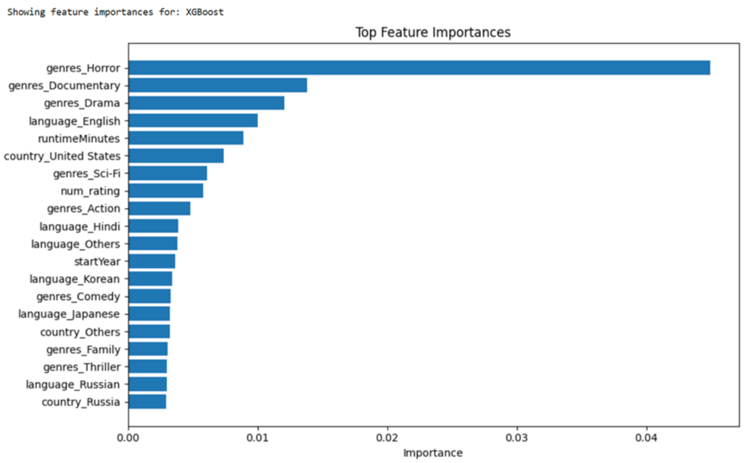
[Confusion Matrix of Xgboost Best Model (3rd Experiment)]



The strategy-specific performance changes served as effective ablation experiments, helping us analyze model performance and explore improvement strategies:

- **Remove Smote:** Resulted in slight performance degradation in simpler models like Random Forest and KNN, but had no effect on XGBoost
- **Remove KNN:** Improved the ensemble's performance, indicating the negative contribution of KNN
- **User of Ensemble:** Improved accuracy while balancing individual model strengths

[Feature Importance of Xgboost Best Model]



```
Showing feature importances for: XGBoost
```

We evaluated feature importance using XGBoost's built-in importance scores. The visualization above captures the top 20 features influencing predictions the most.

The feature with the greatest significance was genres_Horror, highlighting a strong impact of genre on evaluations, particularly when ratings diverge between positive and negative. Other top features included genres_Documentary, genres_Drama, language_English, and runtimeMinutes. Numerical features like `num_rating` and `startYear` also played meaningful roles. Notably, genres_Horror stood out with the highest importance score, suggesting that genre exerts a major influence on the model's predictions

[Insight]

① Use of SMOTE

- When SMOTE was removed (3rd trial), XGBoost performance remained strong, but RandomForest and KNN saw slight declines
- SMOTE enhances performance in simpler models, whereas XGBoost remains robust even with imbalance data

② Exclusion of KNN

- In the fourth trial, omitting KNN from the ensemble led to slight improvements in accuracy and F1-score
- This indicates that KNN may have negatively affected the ensemble's overall performance

③ Ensemble Structure

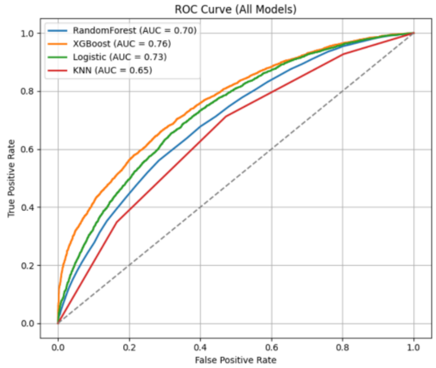
- Individual models (like XGBoost, Logistic Regression) excelled in specific metrics
- The ensemble model achieved the highest accuracy and balanced F1-score, stabilizing overall performance

④ XGBoost Hyperparameter Tuning

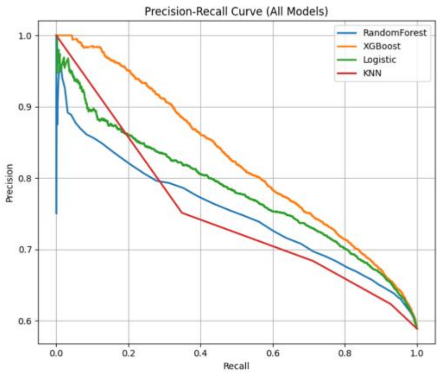
- A consistent XGBoost tuning strategy was applied across experiments, with performance differences arising from configuration factors like SMOTE use, ensemble composition, and model setup

[Qualitative analysis of model performance]

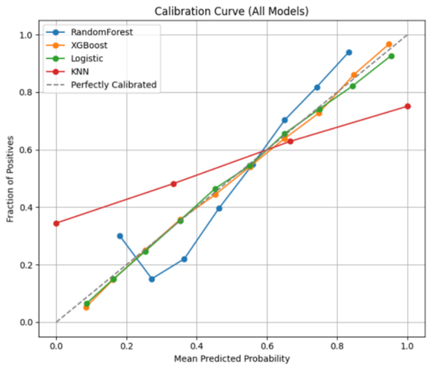
To assess model performance qualitatively, we examined the ROC Curve, Precision-Recall Curve, and Calibration Curve during the third experiment, which yielded the best results



ROC Curve: All models surpassed baseline performance (random prediction), with XGBoost and the 3-Model ensemble achieving the most stable AUC values



Precision-Recall Curve: Useful for evaluating sensitivity to the positive class. Models employing SMOTE retained high precision alongside good recall rates



Calibration Curve: Visualizes the reliability of predicted probabilities. Logistic Regression offered the best calibration, whereas tree-based models exhibited slight overconfidence

2-3. Sensitivity Analysis (XGBoost)

Parameter	Range	Observation Results
n_estimators	100 → 200 → 300	100 is best performance, 200 is similar, 300 and above is minimal improvement
max_depth	3 → 5 → 7	Performance improves as depth increases, optimal at 7 / anything above that may result in overfitting
learning_rate	0.01 → 0.1 → 0.2	Most stable performance at 0.1, 0.01 tends to underfit

Through various experiments, we evaluated how changes in key hyperparameters affect XGBoost performance:

- **n_estimators:** Testing with 100, 200, and 300 estimators revealed that the highest Macro F1 score occurred at 100, with minimal gains observed as we increased to 200 or 300. Thus, simply increasing the number of trees doesn't always equate to better performance, indicating a performance plateau
- **max_depth:** Performance improved with depth, peaking at a depth of 7. Depths beyond this raised overfitting concerns, as overly complex decision boundaries can reduce performance
- **learning_rate:** The rate of 0.1 yielded the most stable results. At 0.01, models tended to underfit, while 0.2 led to faster convergence but a higher risk of overfitting. Choosing the right learning rate is crucial as it affects both convergence speed and model accuracy

2-4. Trade-off Analysis

Several trade-offs were observed during the analysis:

① **Speed vs. Accuracy:** Logistic Regression is very fast, but its performance is lower than that of XGBoost

② **Complexity vs. Interpretability:** Although XGBoost and Random Forest offer high performance, they're complex and require longer training times. Logistic Regression, in contrast, provides ease of interpretation at the expense of lower overall performance

③ **Precision vs. Recall:** XGBoost and ensemble models leaned more towards recall, whereas Logistic Regression revealed relatively higher precision

3. Analysis of misclassification (failure) cases $^{\text{(Exp. 3 – XGBoost, SMOTE not applied)}}$

Example	Actual	Prediction	Cause Estimation
Low budget emotional drama	(+)	(-)	Lack of metadata, emotional lines not reflected
Low-rated action blockbuster	(-)	(+)	Over-reliance on genre and running time
Foreign language art films	(+)	(-)	Country/Language Samples Scarce

① False Negative in Emotion-Focused Dramas

- **Characteristics:** These movies, despite receiving positive reviews, were classified negatively
- **Analysis:** The model seemed insufficient in capturing nuanced emotional content, particularly when relying solely on BERT embeddings
- **Cases:** *Husband Material* (India, Hindi) and *The Bonfire of the Vanities* were misclassified in this manner
- **Improvements:** Enhance rating distribution representation and adjust genre-based precision

② False Positive in Commercial Films

- **Characteristics:** Commercial films rated negatively were predicted positively
- **Analysis:** Similarities in metadata like genre and runtime with positively-reviewed films led to misclassification
- **Cases:** *Jennifer's Body* (USA, Horror/Comedy) and *The Bonfire of the Vanities* were misclassified in this manner
- **Improvements:** Enhance rating distribution representation and adjust genre-based precision

③ Underestimation of Non-English Language Films (False Negatives):

- **Characteristics:** Non-English or multinational films with positive feedback were classified negatively
- **Analysis:** Insufficient training data for languages and countries, and cultural nuances were not captured
- **Cases:** *Ju-Rei: The Uncanny* (Japan, Japanese) and *Back from Eternity* had misclassification
- **Improvements:** Consider oversampling language/country categories and strengthening relevant metadata embeddings

Part B. Unsupervised Learning

1. Methods description

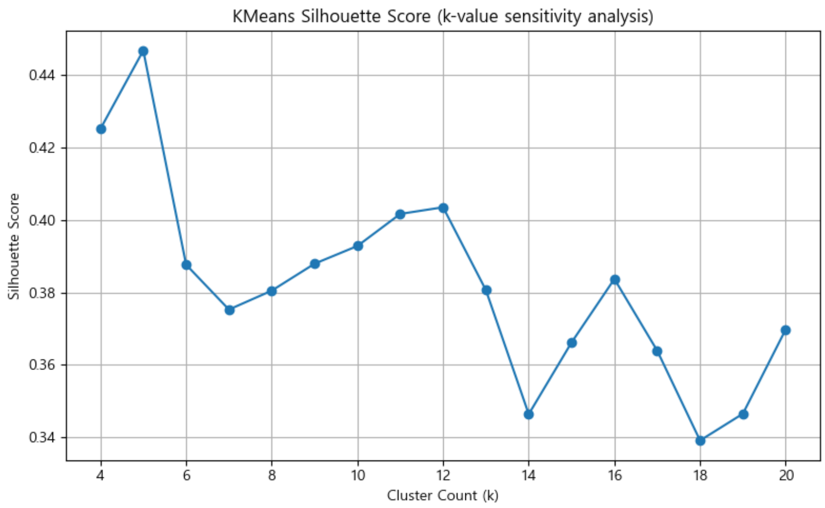
Step	Detail
Data Integration	Combining IMDB/OMDB metadata + MovieLens ratings
Text Preprocessing	Convert plot texts to root form using SpaCy for lemmatization
Feature Eng.	• Plot → TF-IDF embedding (max features: 5000)
	• Categorical(genres, actors, director) → multi-hot encoding
	• Numerical(runtimeMinutes, startYear, num_rating) -> Standard scaling
Dimensionality Reduction	Apply UMAP to reduce dimensions to 3 for visualization
Clustering	KMeans clustering across various k values (range: 4-20)
Evaluation	Silhouette score used to determine optimal k

In the unsupervised learning section of this project, we clustered movies based on plot and additional features following these steps:

- **Text Preprocessing:** We used TfidfVectorizer from scikit-learn to convert plot texts into numerical features.
    - **People-Related Terms:** Words like “man”,”woman”,”person”, etc.., deemed not relevant for content differentiation
    - **Time-related Terms:** Words like “day”,”year”,”time” were removed given their ubiquity and limited discriminative power
    - **Common Verbs:** General verbs such as “come”,”go”,”get”,”want”, and “decide” were filtered out as they are too generic to offer meaningful insights
- **Text Features:** Movie plots were vectorized using a TF-IDF, focusing on a maximum of 5,000 features to capture textual similarities effectively
- **Categorical Features:** Features like actors and director names were transformed using multi-hot encoding, focusing on the most frequent categories
- **Numerical Features:** Numerical attributes such as runtimeMinutes and startYear were standardized using StandardScaler to ensure consistent scaling across features
- **Dimensionality Reduction:** We applied UMAP to reduce the feature space to 3 dimensions, enhancing visualization and clustering potential
- **Clustering Evaluation:** Using silhouette scores, we performed KMeans clustering for various k values to determine optimal clusters, ensuring meaningful segmentation of movies
- **Clustering Details:** Identified the optimal clustering, storing cluster labels and silhouette scores of the best configurations

2. Unsupervised Evaluation

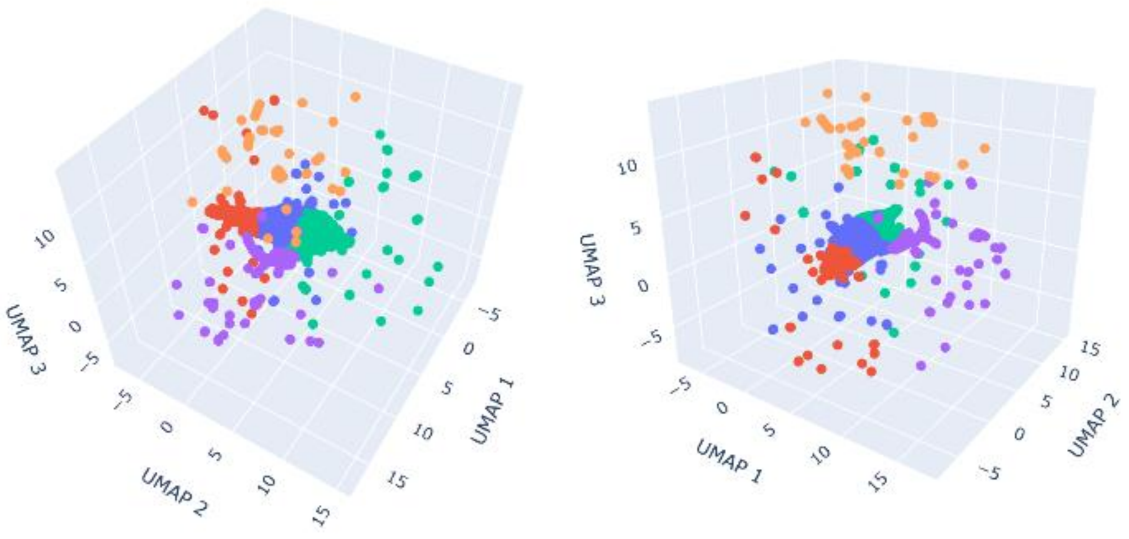
2-1. Optimal Clustering Using Silhouette Scores



- **Silhouette Score Analysis:** The silhouette score is a crucial metric in cluster analysis, providing insights into the cohesion within clusters and the separation between them. It evaluates how similar an object is to its own cluster compared to other clusters, offering a simple interpretation.
- **Silhouette Score Range:** The score ranges from -1 to 1. A high silhouette score close to 1 indicates that the data points are well-matched within their clusters and distinctly set apart from other clusters. A score around 0 suggests overlapping clusters, while negative values indicate potential misclassification.
- **Role in KMeans Clustering:** In our KMeans clustering analysis, the silhouette score guided the selection of the optimal number of clusters (k). By computing this score for different k values, we aimed to find the configuration that best captures the underlying structure of the data
- **Current Finding:** During our experiments with k ranging from 4 to 20, the silhouette score identified k = 5 as the optimal number of clusters, yielding the highest silhouette score in this range. This result indicates that with five clusters, the movies are grouped in such a way that intra-cluster similarity is maximized, and inter-cluster differences are distinct, providing a meaningful division of the movie dataset

2-2. 3D Graph Analysis

The 3D visualization of movie clusters, generated using UMAP and rendered with Plotly, provides a rich graphical representation of the dataset’s structure. This visualization involves three dimensions - UMAP1, UMAP2, and UMAP3 - with each axis spanning from -10 top 15. Here’s an interpretation of the graph:



- **Axis Range and Distribution:** The axes, ranging from -10 to 15 across UMAP1, UMAP2, and UMAP3, cover a broad spectrum of the feature space. This spread effectively captures the separation and distribution of clusters within the embedded space, allowing for clear differentiation between distinct groups.
- **Cluster Arrangement:**
  - **V-Shaped Core:** In the central part of the plot, a prominent V-shaped arrangement is visible, composed of four main colors - green, purple, blue, and red. Each color signifies a distinct cluster, suggesting a natural partitioning of the dataset into these groups based on similarities captured in the underlying features
  - **Orange Cluster:** In addition to the central clusters, an orange-colored cluster appears slightly spread out above the V-shaped core. This suggests that the movies in the orange cluster possess a unique set of characteristics or themes that distinguish them from the central clusters. Their spread indicates a potential diversity or broader scope within or beyond the typical thematic boundaries defined by the inner clusters
- **Interpretation**
  - The color differentiation helps highlight the distinctiveness of each cluster. The central V-shape could indicate a core set of thematic similarities shared among the main movie clusters, while the more diffuse orange cluster could represent niche or outlier categories, potentially holding movies with specialized appeal or mixed genres
  - Understanding these spatial arrangements aids in identifying both the concentrated and diffuse nature of thematic explorations within the movie dataset, offering insights into how genres and narrative styles cluster together in a multidimensional space.

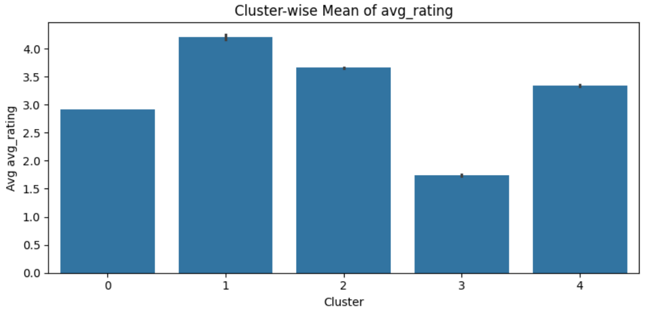
2-3. Cluster Analysis Overview

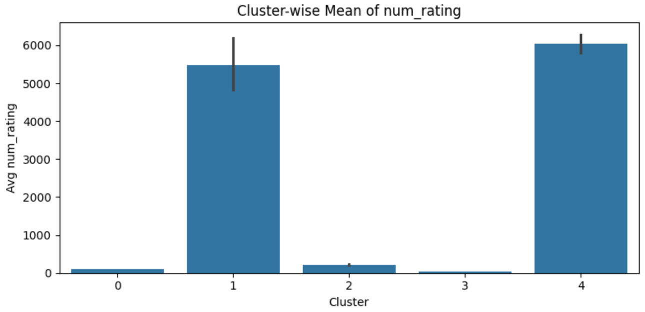
In our analysis, we examined movie clusters based on key metrics, representative films, keyword frequencies, and visualized the findings using WordClouds to capture thematic distinctions.

2-3-1. Cluster Statistics

The table below summarizes key statistics for each cluster, focusing on average ratings and total number of reviews:

Cluster	Avg Rating	Num of Reviews
0	2.91	3,130,631
1	4.21	9,168,236
2	3.66	3,997,948
3	1.74	383,390
4	3.34	14,240,554





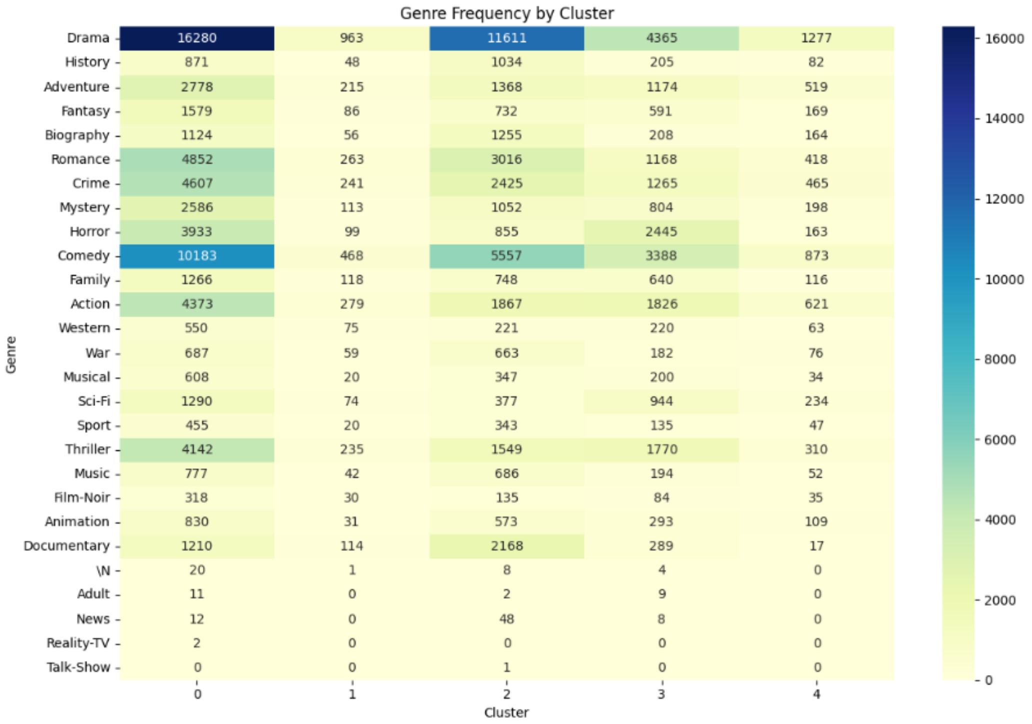
Key Observations:

- Cluster 1 shows the highest average rating, highlighting its strong appeal with impactful movies
- Cluster 3, with the lowest average rating, primarily features family and comedy, reflecting lighter entertainment choices

2-3-2. Top Movies by Cluster (Top 5 by Review Count)

- **Cluster 0:** *The Devil's Own* and *Paranormal Activity*.
- **Cluster 1:** *The Shawshank Redemption* and *Pulp Fiction*.
- **Cluster 2:** *Dial M for Murder* and *Notorious*.
- **Cluster 3:** *The Muppet Christmas Carol* and *Police Academy 3*.
- **Cluster 4:** *Deadpool* and *The Good, the Bad and the Ugly*.

2-3-3. Genre Frequency by Cluster



2-3-4. Plot Keyword Analysis

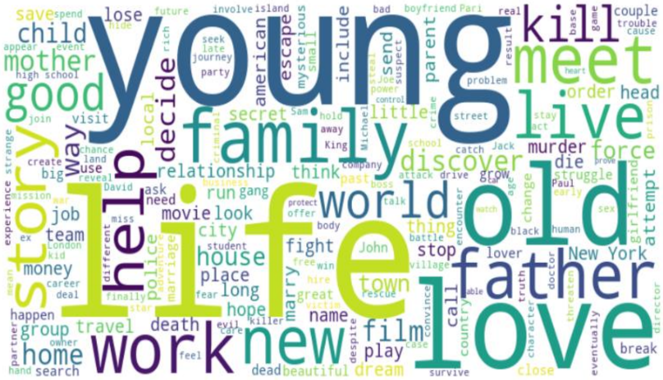
For each cluster, we extracted and ranked frequently occurring keywords from movie plots, illuminating distinct thematic focuses across clusters. For example:

- **Cluster 0:** Keywords like "murder" and "escape" highlight intense plot elements.
- Each keyword was quantitatively assessed (e.g., "look": 0.29, "run": 0.28), offering a measurable insight into thematic prevalence.

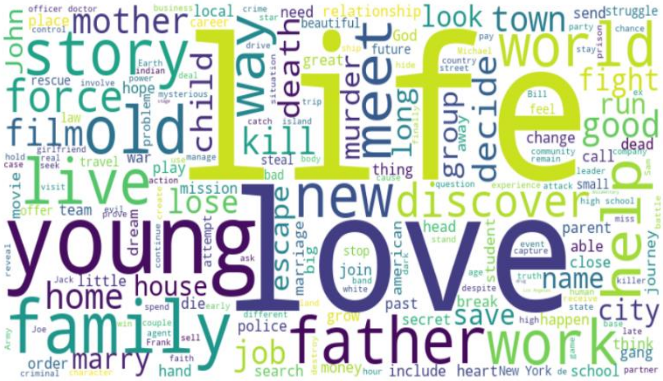
2-3-5. WordCloud Visualizations

We utilized WordClouds to visually represent the most prominent keywords across clusters, offering a quick glimpse into their thematic core. This visualization technique accentuates textual characteristics and helps identify dominant themes without exhaustive manual analysis

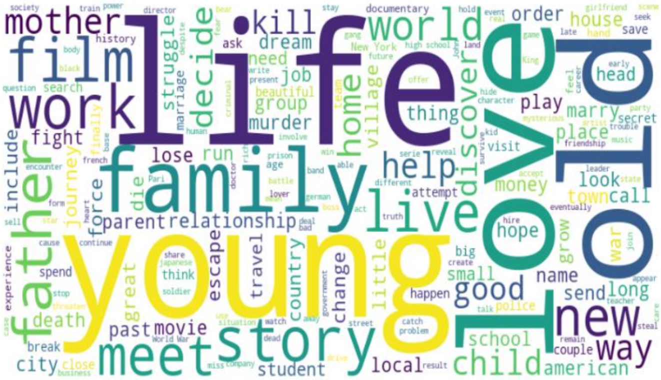
Cluster 0 Plot WordCloud



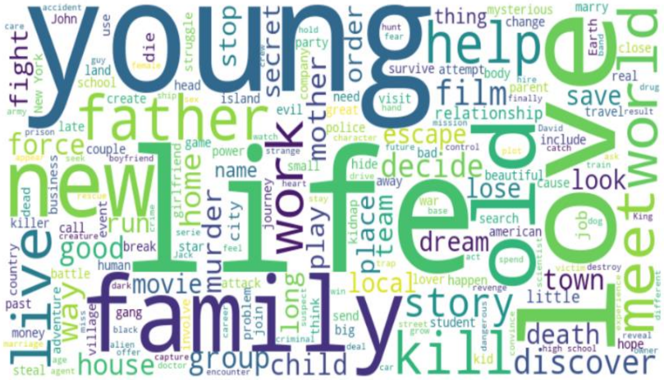
Cluster 1 Plot WordCloud



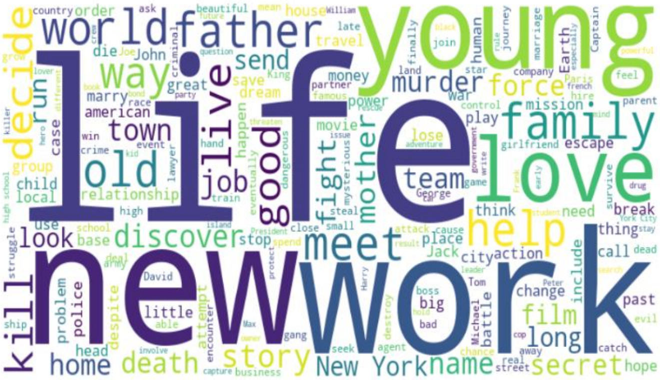
Cluster 2 Plot WordCloud



Cluster 3 Plot WordCloud



Cluster 4 Plot WordCloud



2-4. Clustering Insights

By employing the KMeans algorithm, we analyzed movie data to uncover unique characteristics for each cluster, based on elements such as genre, directors, actors, and key plot keywords.

- **Cluster 0: Diverse and Intense Genre Mix**
    - **Average Rating:** 2.91
    - **Number of Reviews:** 3,130,631
    - **Top Genres:** Action, Comedy, Drama
    - **Representative Movies:** *The Devil's Own, National Lampoon's European Vacation, Paranormal Activity*
    - **Most Frequent Keywords:** murder, run, death, escape
    - **Characteristics:** This cluster encapsulates films with a blend of various genres, often characterized by intense plot elements. These movies leverage diverse genre components to captivate audiences
- **Cluster 1: Impactful Dramas and Stories**
    - **Average Rating:** 4.21
    - **Number of Reviews:** 9,168,236
    - **Top Genres:** Drama, Crime, Romance
    - **Representative Movies:** *The Shawshank Redemption*, *Forrest Gump*, *Pulp Fiction*
    - **Most Frequent Keywords:** life, love, young, family
    - **Characteristics:** This cluster focuses on deeply moving dramas and storytelling, emphasizing emotional connections and life narratives, which often result in higher ratings.
- **Cluster 2: Classic Style Creations**
    - **Average Rating:** 3.66
    - **Number of Reviews:** 3,997,948
    - **Top Genres:** Drama, Mystery, Romance
    - **Representative Movies:** *Dial M for Murder, Notorious, Rebecca*
    - **Most Frequent Keywords:** life, young, love, old
    - **Characteristics:** Dominated by classic and traditional-style films, this cluster primarily involves mystery and romance, appealing to fans of traditional cinema.
- **Cluster 3: Family and Drama Comedies**
    - **Average Rating:** 1.74
    - **Number of Reviews:** 383,390
    - **Top Genres:** Comedy, Drama, Family
    - **Representative Movies:** *The Muppet Christmas Carol*, *Police Academy 3*, *Secondhand Lions*
    - **Most Frequent Keywords:** life, young, family, love
    - **Characteristics:** This cluster centers on family and comedy dramas, emphasizing warm atmospheres and family relationships, with a substantial focus on casualness and humor.
- **Cluster 4: Modern Action and Suspense**
    - **Average Rating:** 3.34
    - **Number of Reviews:** 14,240,554
    - **Top Genres:** Action, Comedy, Thriller
    - **Representative Movies:** *The Game*, *Deadpool*, *The Good, the Bad and the Ugly*
    - **Most Frequent Keywords:** life, new, work, young
    - **Characteristics:** Featuring modern and dynamic action and suspense films, this cluster includes popular movies that offer fresh experiences and continuous tension to their audiences.
- **Summary of Differences Between Clusters:** Cluster 1 vs. Cluster 3: Cluster 1 focuses on storytelling-centric, impactful dramas achieving high ratings frequently, while Cluster 3 encompasses lighter content, including comedy and family dramas.

2-5. Sensitivity Analysis

2-5-1. Parameter Sensitivity: KMeans Clustering

- **Key Hyperparameters:** The number of clusters (k) is a pivotal parameter in KMeans. While silhouette scores suggested a range of potential k values, the practical alignment of cluster themes led us to finalize k=5.
- **Impact Assessment:** We tested k values from 4 to 20. The silhouette score initially favored k = 2; however, meaningful thematic grouping was only achieved at k=5, illustrating the sensitivity of thematic coherence to the chosen number of clusters.
- **Conclusion:** Minor adjustments in k resulted in notable shifts in cluster assignments and thematic interpretations, highlighting the critical role of this hyperparameter in extracting meaningful insights from the data.

2-5-2. Feature Sensitivity: Text Representation with TF-IDF

- **Feature Engineering Approach:** We utilized TF-IDF to convert plot texts into feature vectors with a max feature size of 5000. This setting provided a robust balance between computational efficiency and thematic richness
- **Testing Variations:** Altering the max_features parameter showed that increasing the feature set captured more nuanced possibilities, leading to more scatter clusters, whereas reducing features diminished thematic clarity and led to overlap
- **Conclusion:** TF-IDF feature selection is crucial for aligning the dimensional reduction outcomes with thematic expectations. Proper tuning is vital for maintaining cluster integrity

Discussion

Learnings from Part.A (Supervised Learning)

Through the process of supervised learning in Part A, we gained meaningful insights into how both textual and structured features influence movie rating prediction. We found that using pre-trained BERT-based sentence embeddings captured nuanced semantic information in plot summaries, while structured metadata—such as genre, country, and runtime—provided complementary signals. Among the models tested, XGBoost consistently yielded the best performance, leveraging its strength in handling non-linear relationships and mixed feature types. Interestingly, simpler models like Logistic Regression sometimes performed comparably, especially when combined with SMOTE, suggesting that model complexity is not always the key determinant of success.

We encountered several challenges. The first was class imbalance, which we addressed using SMOTE, though its effectiveness varied across experiments. The second was feature sparsity resulting from high-dimensional multi-hot encoding of categorical variables; we resolved this by limiting the number of categories to a top-K list and grouping the rest under "others." We also struggled with the computational burden of generating sentence embeddings, especially in limited cloud environments. To mitigate this, we cached the embedding results as .npy files and offloaded training to local machines when necessary.

Looking forward, with more time and resources, we would expand the feature set to include additional metadata (e.g., director reputation or actor popularity), explore fine-tuning of transformer models, and apply interpretability techniques like SHAP to better understand feature contributions. We also see value in examining potential biases—such as those tied to genre or language—and integrating fairness metrics to ensure responsible modeling. Overall, this part of the project taught us the importance of balancing model complexity, data engineering, and evaluation rigor in real-world machine learning workflows.

Learning from Part.B (Unsupervised Learning)

Through Part B of the project, we gained valuable insights into the role of text preprocessing—such as lemmatization and stopword removal—in improving the quality of clustering results. By leveraging TF-IDF embeddings for plot summaries and using UMAP for dimensionality reduction, we were able to visualize and cluster the dataset more effectively. KMeans helped uncover genre and theme-based groupings that aligned with audience preferences.

Interestingly, although k=2 yielded the highest silhouette score, it failed to reflect the rich thematic diversity of the movie dataset. By applying domain knowledge, we identified k=5 as a more interpretable and useful configuration, which led to more meaningful segmentation. WordClouds of each cluster provided vivid representations of their thematic distinctions.

We faced challenges with balancing the scaling of numerical and categorical features and determining the appropriate number of clusters. These were addressed through careful preprocessing and qualitative evaluation of cluster coherence. If given more time, we would explore sentiment analysis of user reviews and experiment with advanced unsupervised methods such as hierarchical clustering or LDA topic modeling to further deepen our understanding of movie characteristics and audience behavior.

Ethical Consideration

Supervised Learning Ethical Consideration

The supervised learning model developed in this project predicts ratings based on movie metadata and plot summaries. When applied to real-world user recommendations or automated evaluations, several ethical concerns may emerge:

- **Transfer and Reinforcement of Data Bias**: The training data may inherently possess biases favoring certain genres, countries, or languages. For example, a model primarily trained on English-language commercial films might produce biased predictions against non-English independent films. This imbalance could undermine cultural diversity and compromise fairness.
  - **Mitigation:** During data collection, it's vital to ensure a diverse representation of films across various countries and languages. Following model training, the predictions should be evaluated across different subgroups to detect possible biases. Implementing reweighting strategies or fairness constraints can help address detected biases.
- **Possibility of Avoiding Responsibility Due to Lack of Explanatory Power:** If the model fails to articulate the rationale behind its predictions, it can lead users or developers to deflect responsibility for incorrect decisions or misinterpret causality.
  - **Mitigation:** It's important to enhance the transparency of model predictions using interpretability tools like feature importance and SHAP. Providing comprehensible explanations will aid users in understanding the decision-making process behind predictions.

Unsupervised Learning Ethical Consideration

Implementing clustering algorithms in the context of movie data analysis brings forward several ethical considerations, which we addressed in our project:

- **Bias and Representation**: Models can reflect and even enhance biases present in source datasets. For example, algorithms may inadvertently favor genres or directors widely known in more comprehensive datasets.
  - **Mitigation:** 1) To counteract content visibility bias, we incorporate a diversity of languages and countries, enhancing dataset representativeness 2) Implementing a feedback system within our recommendation engine will enable users to identify and report biases or inaccuracies, pushing for continuous improvement and reflective adjustments in our modeling process
- **Impact on Content Availability:** Cluster analysis impacts how content recommendations are made, potentially affecting which genres or types of films receive visibility
  - **Mitigation:** We regularly reviewed clustering results to ensure fair visibility across diverse cinematic themes and adjusted parameters to minimize any unintended bias influencing recommendations

Statement of Work

- Eric Kim: Overall project management, data collection, and schedule management

- Younghoon Oh: Supervised Learning Model Development, Visualization

- Shin Choo: Unsupervised Learning Model Development, Visualization

- Common : Writing Proposal/Final Report

Appendix A – Related Work

[1] Karandikar, Y. (2015). *CSE 255 Assignment 1: Movie Rating Prediction using the MovieLens dataset*. University of California, San Diego.
https://mcauleylab.ucsd.edu/public_datasets/cse255/projects/wi15/Yashodhan_Karandikar.pdf

Accessed May 8, 2025

Description: The article focuses on predicting movie ratings using linear regression, collaborative filtering and latent factor models with the MovieLens 1M dataset.

Differences: Our method leverages the semantic richness of textual data by utilizing BERT embeddings for plot summaries

[2] Hedden, S. (2022, February 22). *How to download and explore movie data: Using the TMDB API and Python to explore the 2022 Oscar nominees using network analysis*. Medium.

https://medium.com/data-science/how-to-download-and-explore-movie-data-1948a887c530

Accessed May 6, 2025

Description: This article focuses on employing Python to execute API calls, gather full cast and crew lists, construct networks, and perform network analysis utilizing tools like NetworkX for graph-based insights.

Differences: While Hedden's tutorial is focused on retrieving and analyzing network connections among movies using the TMDB API, our project extends these ideas into the realm of content-based thematic analysis through machine learning. We employ advanced unsupervised learning techniques to cluster movies based on narrative content, offering a deeper exploration into the thematic and semantic elements of film data.

[3] Kunzler, J. (2021, November 3). *Choosing the best regression model: IMDB movie rating prediction*. Medium. URL

Choosing the Best Regression Model -IMDB Movie Rating Prediction | by Jingkunzler | Medium

Accessed May 6, 2025

Description: This article explores the process of predicting IMDB movie ratings by comparing various regression techniques, including KNN, Stochastic Gradient Descent, Random Forest, and Gradient Boosting Trees.

Differences: Our project integrates both supervised and unsupervised learning techniques, incorporating advanced natural language processing (NLP) tools like BERT for embedding plot summaries.

Appendix B – Data Schema

Data Sources:

- https://datasets.imdbws.com/title.basics.tsv.gz
  - Clicking to the link downloads the data on local computer
- https://www.omdbapi.com
  - You can download data using your API Key
- https://files.grouplens.org/datasets/movielens/ml-32m.zip
  - Clicking to the link downloads the data on local computer