IE7275 Fall 2023 Term Project Report
Reenae Victor and Saravanan Arumugam

Movie Clustering and Rating Prediction for Content Organization

Team Responsibility

Our team consists of Saravanan Arumugam and Reenae Victor and both of us shared the project responsibilities equally.

Introduction

Movies are becoming a more and more important element in people's lives. There are a number of choices of movies, and what a person may like is an interesting topic. In this project, we have built a Rating Prediction System as well as performed Movie Clustering. This project focuses on developing a smart movie clustering system for improved organization and user experience. Additionally, it aims to create a precise movie rating prediction model, aligning with industry demands for tailored content delivery.

We used an IMDB dataset collected from Kaggle with movies all the way back from the 1890's till current date. The data included over 10 million instances of Movies, TV- Shows, TV-Series, Short Films, etc. 80% of the data was considered as the training dataset to build this movie rating prediction model with and without using the dimensionality reduction. The other 20% dataset is used as a test dataset to evaluate the model using RMSE, MSE, MAE, R-squared, SSE (clustering) and Computing Time. The Clustering process could help for better content organization and search functionality, which would help the recommendation systems and the users in finding what they are looking for more efficiently.

Problem Definition

In the contemporary landscape of digital entertainment, the vast and diverse array of movies poses a significant challenge for users seeking personalized content experiences. The current methods of content recommendation often require very high computing power to capture the nuanced preferences of individual users. As a result, there is a critical need for innovative approaches that not only categorize movies into meaningful groups but also predict user ratings with precision. This project addresses this challenge through a two-fold initiative: Movie Clustering and Rating Prediction.

Conventional genre-based categorization systems are often insufficient, failing to capture the intricate connections that exist between movies with similar themes, cast, crew, and user appeal. The absence of a robust clustering mechanism leaves viewers navigating a sea of content without personalized guidance. Additionally, marketers face the challenge of delivering targeted advertising that resonates with specific audience segments. The Movie Clustering aspect of the project seeks to overcome these challenges by implementing advanced clustering algorithms, unveiling the latent relationships between movies, and thereby enhancing the content recommendation landscape.

Understanding the complex factors that influence a user's movie rating is pivotal for delivering personalized content recommendations. Current rating prediction models often overlook the subtleties of individual preferences, leading to inaccuracies in predicting user reactions to new content. The lack of a precise rating prediction system hinders the development of tailored content experiences that resonate with a user's unique taste. This project aims to address this challenge by creating a sophisticated predictive model that goes beyond conventional rating predictions.

The successful implementation of Movie Clustering and Rating Prediction not only addresses the immediate challenges outlined above but also holds broader implications for the digital entertainment industry. By crafting a more personalized content experience, the project is poised to redefine user engagement, retention, and satisfaction. Furthermore, the insights gained from the intricate factors influencing user ratings provide content providers and marketers with a strategic edge, enabling them to optimize content libraries and advertising strategies for maximum impact.

In summary, the dual-fold nature of this project aims to tackle the challenges in content recommendation and user rating prediction, ultimately contributing to a betterment in the way viewers interact with digital entertainment. The potential for content enhancement and a more tailored content experience underscores the critical importance of this initiative in the evolving landscape of personalized digital content consumption.

Data Description

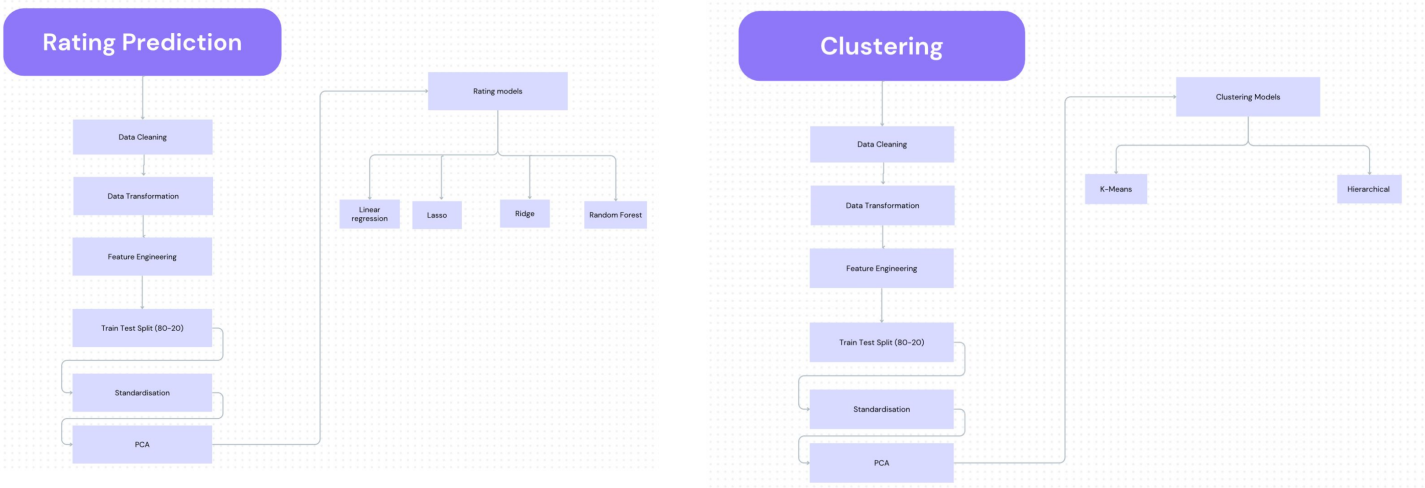
The IMDB dataset, obtained from Kaggle, served as a comprehensive resource. The dataset is composed of key files, each offering unique insights. The "title.akas.tsv.gz" file provides details on localized titles, regions, and languages, offering a nuanced understanding of title variations. "title.basics.tsv.gz" encompasses information on titles, including type, primary/original titles, genres, and release details. "title.principals.tsv.gz" sheds light on the principal cast and crew for various titles, detailing their roles and contributions. The "title.ratings.tsv.gz" file is crucial for assessing popularity, providing IMDB ratings and vote statistics. Lastly, "name.basics.tsv.gz" offers comprehensive information about industry individuals, including birth/death years, professions, and notable titles.

This dataset's significance lies in its ability to support in-depth analyses of movie titles, genres, cast and crew dynamics, user ratings, and the profiles of industry contributors. Leveraging this dataset enables researchers and analysts to unearth patterns and trends, contributing to a richer understanding of the dynamics within the realm of movies. Furthermore, it serves as a valuable resource for developing models and algorithms, aligning with the project's goals of movie clustering and rating prediction for personalized content enhancement.

Example

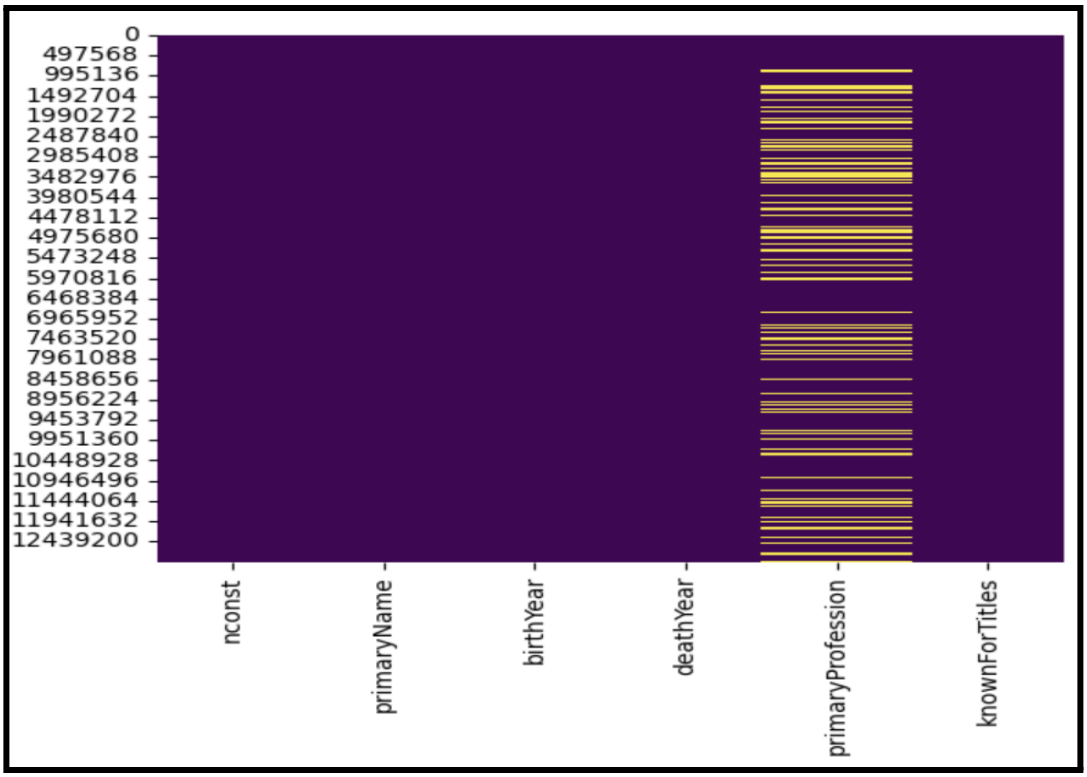
Proposed Methods

The proposed methods for the Rating Prediction and the Movie clustering are depicted in the flow charts attached below,



Data Cleaning:

The tab separated files were read using pandas into a jupyter notebook for the analysis and modeling. Duplicates, Missing values were handled using pandas. The summary statistics and the data types of the features (columns) were also checked using pandas functions.



This plot shows the null values in the "primary profession" column of the "name" dataframe. We used the seaborn package, heatmap, to create this plot. These null values were removed since it accounted for less than 0.4% of the dataset.

Data Transformation:

New data frames were created using the "pd.merge" function of pandas and the data types were changed into int and float to support the requirements of the machine learning models that were used for the prediction and clustering.

The data frame used for prediction contains 485,000 records and 39 features after feature engineering and dimensionality reduction using PCA with 95% variance.

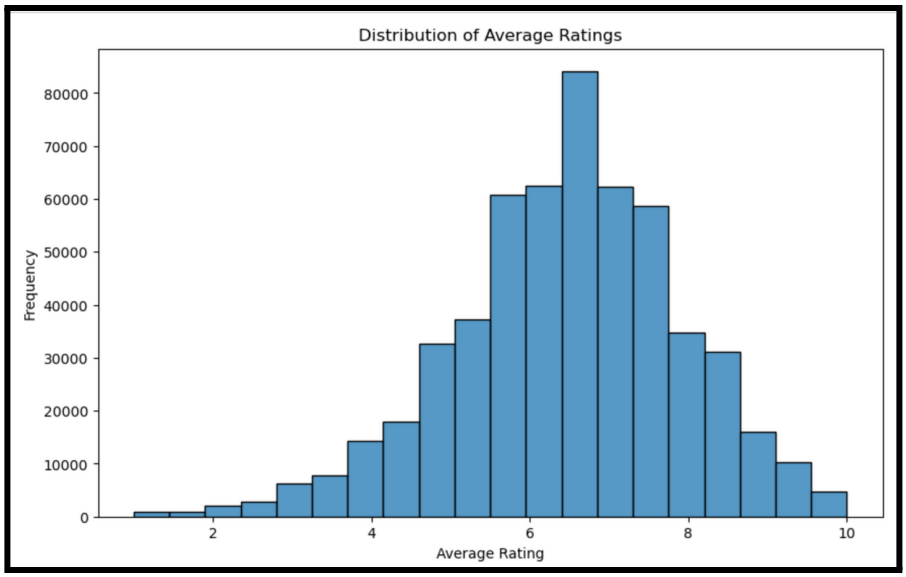
The data frame used for clustering contains 5,935,000 records and 47 features after feature engineering and dimensionality reduction using PCA with 95% variance.

Feature Engineering:

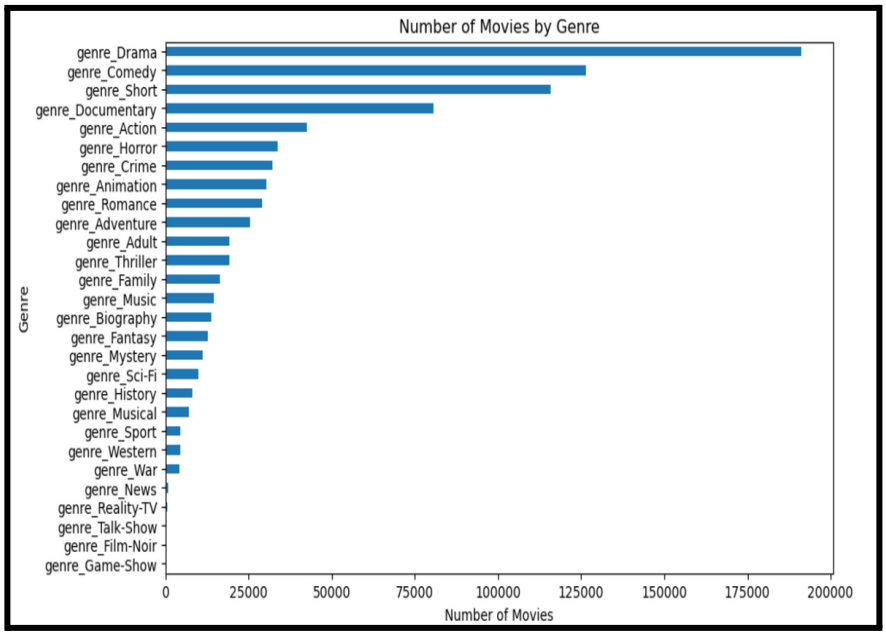
Feature engineering techniques like one hot encoding, manual binning were applied to convert the nominal categorical features into numerical features so that both clustering and prediction algorithms such as K-mean, Linear Regression, and Random Forest can utilize the data for the intended purposes.

Exploratory Data Analysis:

EDA was conducted to learn about the dataset per and post the pre-processing steps. We explored different visualizations to understand the data distribution, correlation, and impact of genre and rating over time.

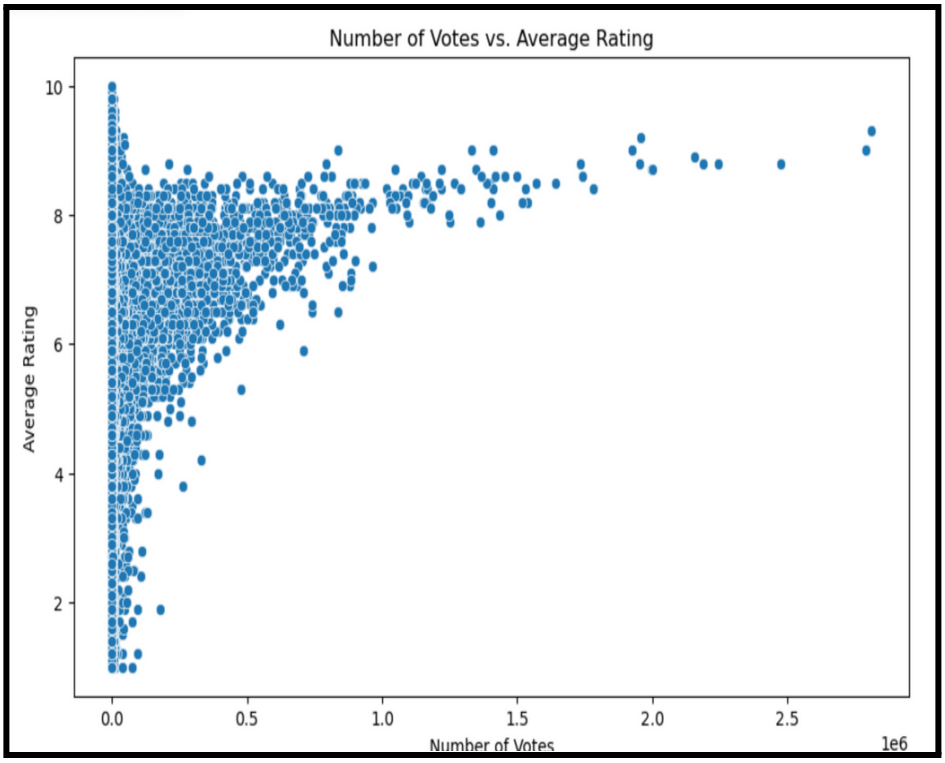


This plot shows the distribution of rating over time using a histogram.

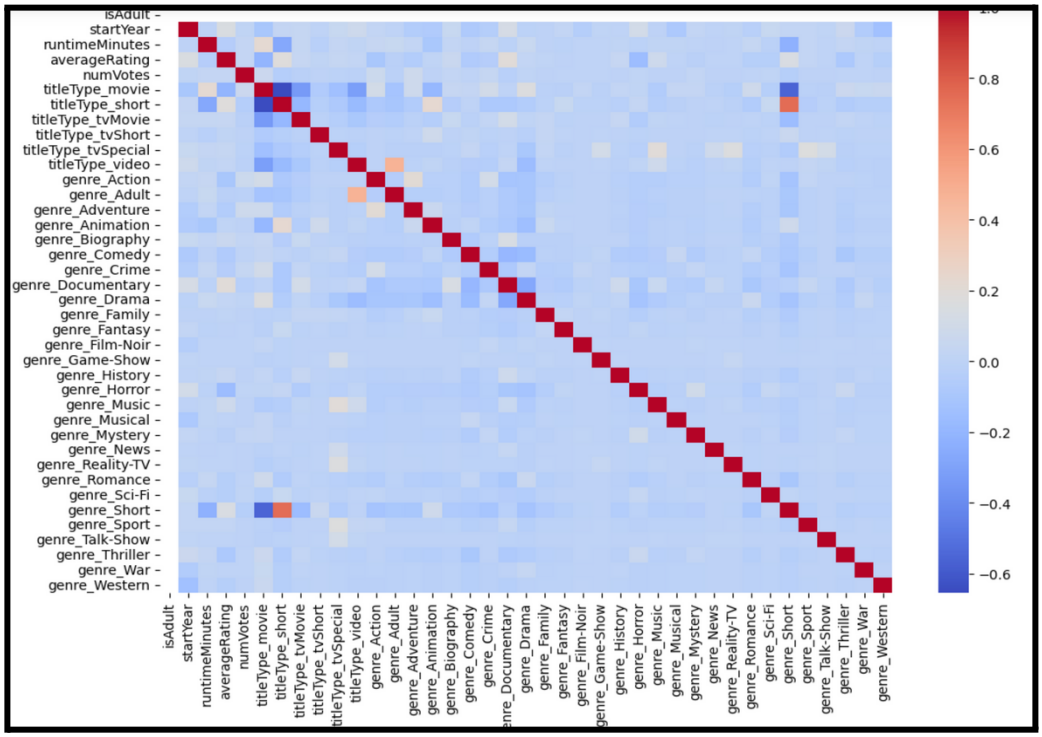


This plot shows the distribution of movies in different genres using a horizontal bar plot.

The plot below shows the relationship between average rating and number of votes using a scatter plot. We can see that this image negates the hypothesis that if a movie has more votes, its rating will decrease. We clearly see no such inversely proportional relationship between these two variables.



The heatmap below shows the correlation between all the variables in the data frame used for the rating prediction task. We can clearly see that only one pair of variables are highly positively correlated and one pair is highly negatively correlated. Upon further examination, they were identified as “genre_shortfilm: and “type_shotfilm”. They were removed before modeling.



Modeling:

The data frames were split into training (80%) and testing (20%) datasets using the scikit-learn package's train_test_split method. Then they were normalized separately using the standard scaler method to preserve any data leak.
Principal Component Analysis (PCA) a Dimensionality Reduction technique was employed to reduce the high feature space (dimension), to reduce the complexity of the model while maintaining the efficiency. We trained all the models for prediction with and without PCA to compare its impact on the results.

Linear Regression, Lasso Regression, Ridge Regression, and Random Forest algorithms were used for the rating prediction. K-means clustering and Hierarchical clustering algorithms were used for the movie clustering. The results of these models are explored in the next section.

Experimental Results:

The Prediction Results - From the experiments, Linear Regression, Lasso Regression, and Ridge Regression yielded similar predictive performance, while Random Forest exhibited superior accuracy with lower error metrics.

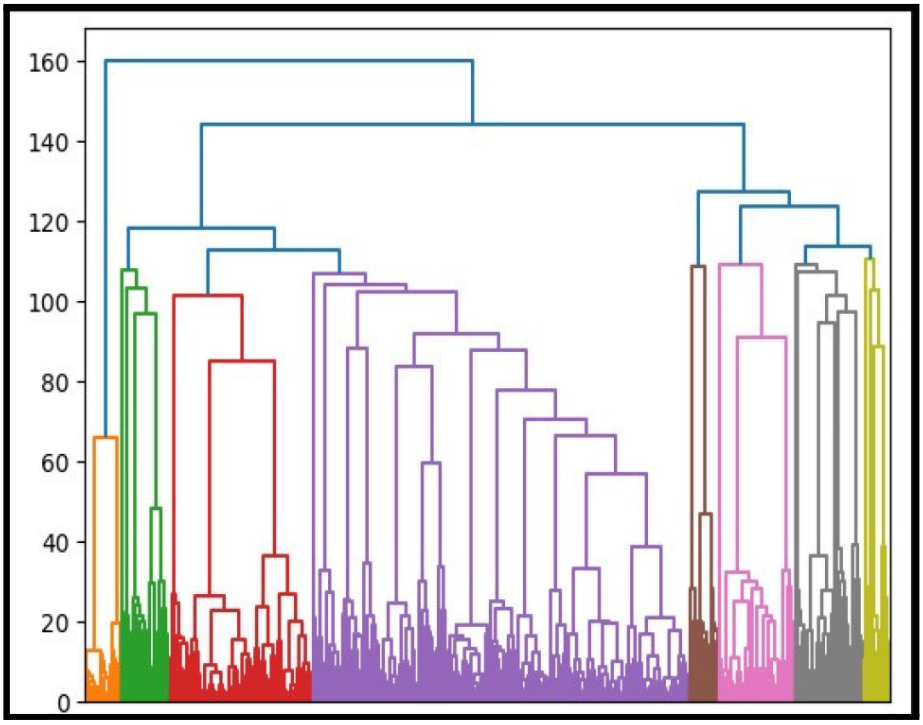
However, the computational efficiency of the linear models contrasts with the higher computational time required for Random Forest.

Model	MSE (Without PCA)	RMSE (Without PCA)	MAE (Without PCA)	R-Squared (Without PCA)	Computing Time (Without PCA) in seconds
Linear Regression	1.652	1.285	0.996	0.162	0.223
Lasso Regression	1.972	1.404	1.096	- 1.444e-05	0.178
Ridge Regression	1.652	1.285	0.996	0.162	0.141
Random Forest	1.524	1.234	0.933	0.227	271.833

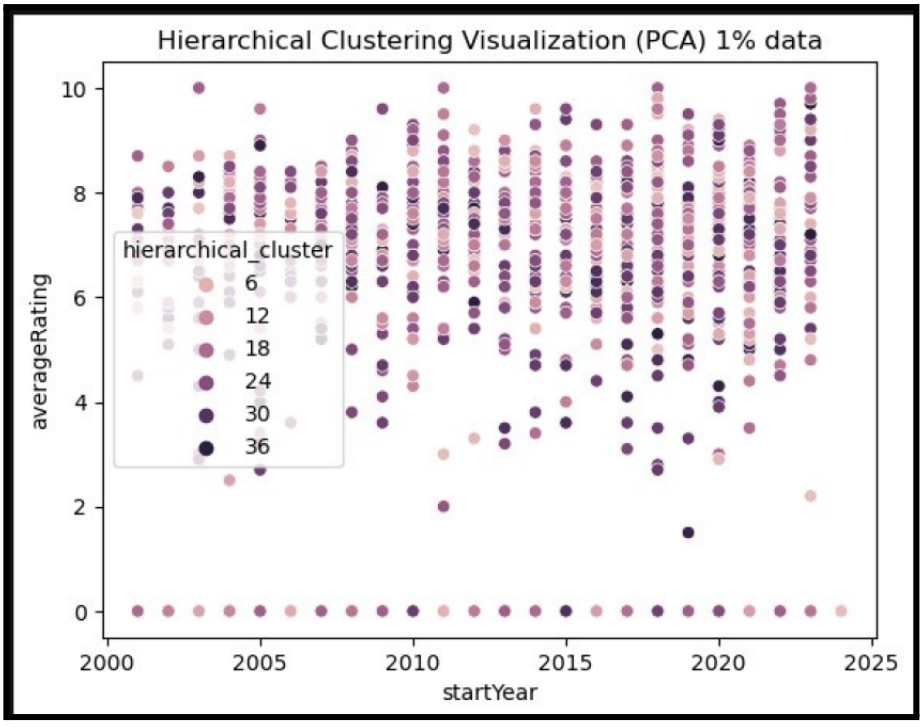
Model	MSE (Without PCA)	RMSE (Without PCA)	MAE (Without PCA)	R-Squared (Without PCA)	Computing Time (Without PCA) in seconds
Linear Regression	1.652	1.285	0.996	0.162	0.223
Lasso Regression	1.972	1.404	1.096	- 1.444e-05	0.178
Ridge	1.652	1.285	0.996	0.162	0.141

Regression					
Random Forest	1.524	1.234	0.933	0.227	271.833

Clustering Results - The hierarchical clustering algorithm's output dendrogram and the visualization of clustering based on a sample threshold distance can be seen below.

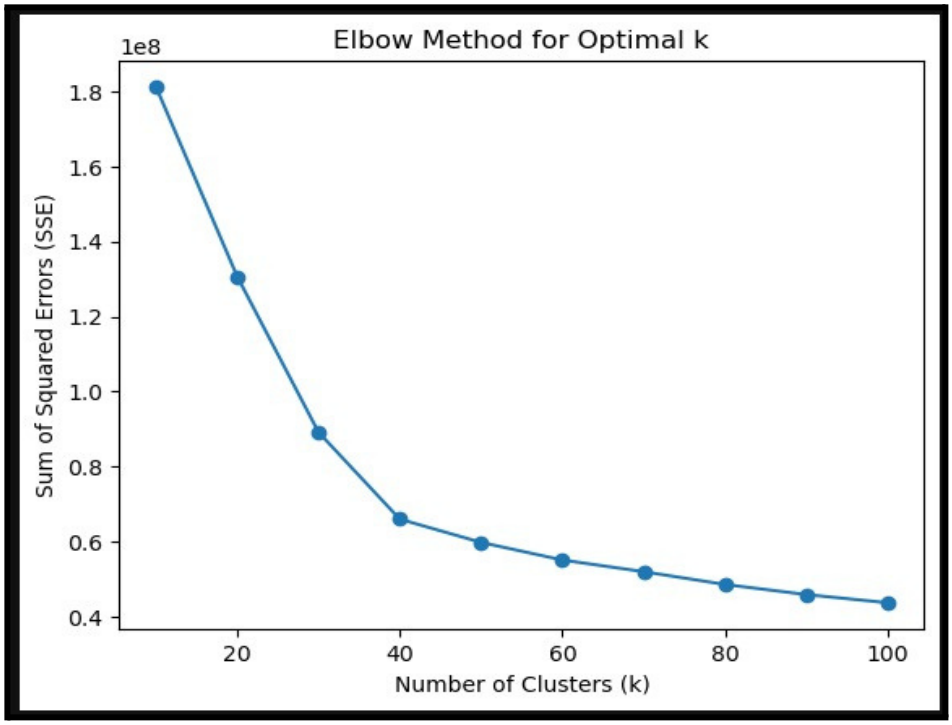


The dendrogram shows all the possible clusters that can be formed using the "complete linkage" method.

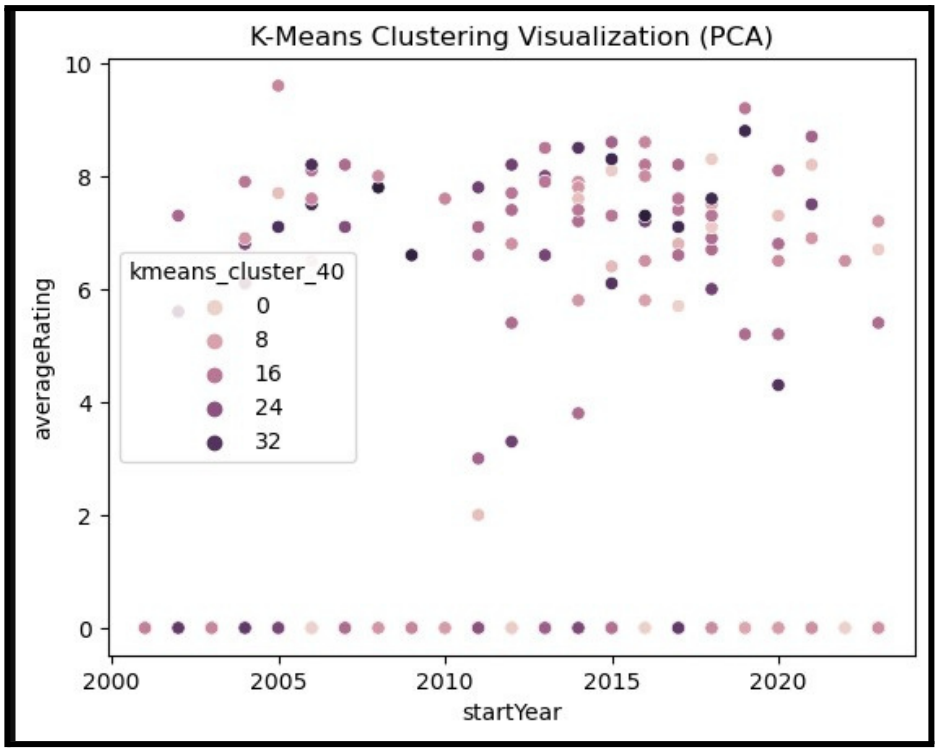


This plot visualizes the clusters formed using the hierarchical clustering algorithm on 1% of the data after applying PCA as a scatter plot of two arbitrary features.

K-Means clustering was executed for multiple values of K and an elbow chart is generated to aid in the identification of the optimal K.



This plot shows the values of SSE (Sum of Squared Error) - an evaluation metric for the clustering algorithm against the number of clusters initialized for the K-Means clustering algorithm. We see a clear trend of reduction in error with an increase in K value.



This plot visualizes the clusters formed using the K-Means clustering algorithm on 0.01% of the data after applying PCA as a scatter plot of two arbitrary features.

7. Conclusion

Linear Models and PCA Impact:
Linear Regression, Lasso, and Ridge models showed similar performance, with a slight difference in metrics.
PCA had limited impact on linear models, suggesting careful consideration of the trade-off between computational efficiency and model performance.

Random Forest Trade-off:
Random Forest outperformed Linear Regression but incurred higher computational costs.
Consider the trade-off between predictive accuracy and efficiency when deciding on model selection.

Clustering Efficiency:
K-Means proved more efficient for larger datasets compared to hierarchical clustering.
Considering the minimal time difference (200 seconds for hierarchical vs. 300 seconds for K-Means) between the models and the fact that hierarchical was executed using only 0.1% of the dataset compared to 100% for the K-means, we recommend prioritizing K-Means.

Future Scope:
Prediction - Include textual features using NLP to improve prediction models.
Clustering - More complex models capture relationships better.

References

1. Ricci, F., Rokach, L., & Shapira, B. (2011). Recommender systems: Introduction and advances. Springer.
2. Berry, M. W. (2007). Machine learning for content organization. Springer.