Movie Recommendation Engine

Introduction

Movies have been a fundamental part of mainstream culture since their inception. In the pre-internet era, word of mouth served as the primary means for individuals to decide what to watch. However, with the advent of streaming services, the landscape of movie selection has vastly expanded, offering an array of choices and platforms. With the abundance of options readily available to us at our fingertips, a pressing challenge emerges: efficiently presenting the right information to users.

The prospect of manually sorting through a vast sea of data would be overwhelming for the casual consumer. Fortunately, machine learning algorithms can be utilized to mitigate this burden, offering movie recommendations based on various factors such as past preferences, genre labels, and synopsis data. These algorithms analyze user behavior and feedback to continuously refine the accuracy of their suggestions.

Presenting subscribers with highly relevant recommendations is crucial for services like Netflix and Hulu whose business models rely on long-term user engagement and retention. The importance of an effective recommendation system cannot be overstated; it only serves to enhance the user experience and facilitate content discovery. If a recommendation system fails to provide precise and enticing suggestions, then it ultimately risks leading to unsatisfied customers who eventually churn and leave the service entirely.

Despite the apparent complexity of providing spot-on recommendations, organizations have been able to simplify the process through implementation of algorithm-based recommendation systems. Furthermore, platforms like IMDB provide an abundance of data encompassing movie titles, descriptions, ratings, and cast/crew information, which serve as valuable resources for building and training a modern recommendation system.

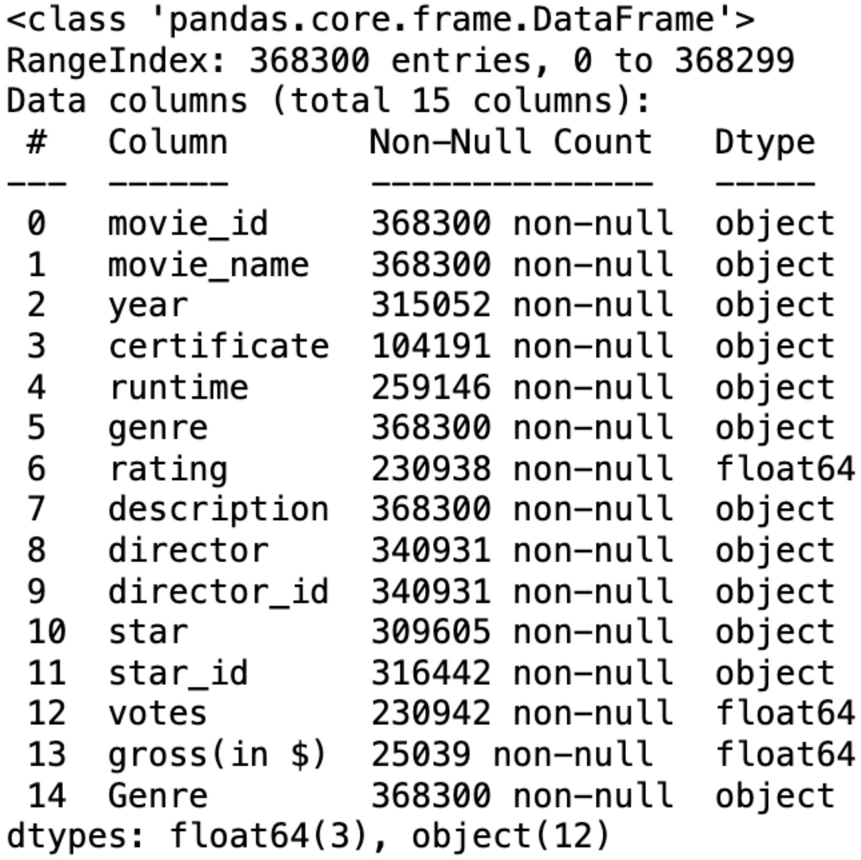
The hope is that in this rapidly evolving digital age, such interplay between available data and entertainment trends will pave the way for a more personalized and convenient movie browsing experience. This analysis carries the end goal of transforming the overwhelming task of navigating modern video streaming services into an effortless, enjoyable activity.

Analysis

Data Preparation

The data was retrieved from Kaggle but originated from IMDB.com. There were 16 separate CSV files, one for each genre. Each file contained information such as movie name, ID, year released, director, stars, votes, and revenue. To begin with, each file was iterated over, and the genre was added to the data frame based on the file name. This was done because the genre column in the original data had multiple categories per movie. The CSV files were then concatenated into one data frame. This resulted in a data frame with 368,300 rows and 15 columns.

Figure: Combined data frame



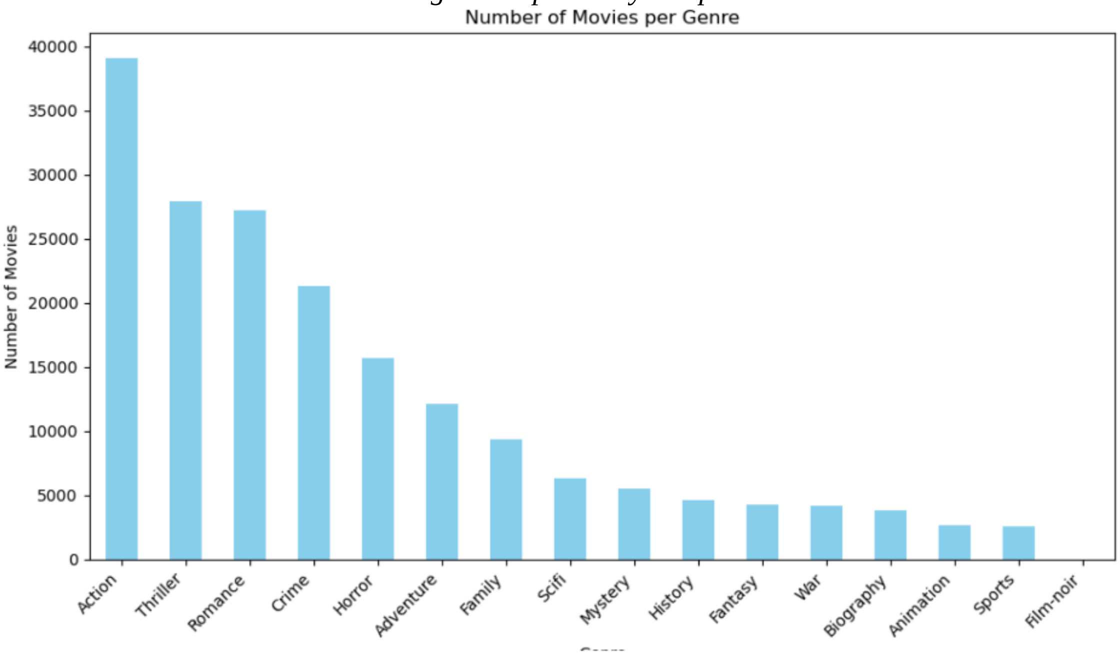
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 368300 entries, 0 to 368299
Data columns (total 15 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   movie_id     368300 non-null  object
 1   movie_name   368300 non-null  object
 2   year         315052 non-null  object
 3   certificate  104191 non-null  object
 4   runtime      259146 non-null  object
 5   genre        368300 non-null  object
 6   rating       230938 non-null  float64
 7   description  368300 non-null  object
 8   director     340931 non-null  object
 9   director_id  340931 non-null  object
 10  star         309605 non-null  object
 11  star_id      316442 non-null  object
 12  votes        230942 non-null  float64
 13  gross(in $)  25039 non-null   float64
 14  Genre        368300 non-null  object
dtypes: float64(3), object(12)
```

The combined data frame was then checked for duplicate values, where half of the values were removed. The columns used for modeling were the description and genre field where there are no null values found in those columns. The description field was then vectorized using Count Vectorizer and assigned to a new data frame. Additional cleaning was done on the data frame to remove numbers, stem the words and remove any word with a length of less than 3. Once the data frame was cleaned, it was split into testing and training sets.

Exploratory Data Analysis

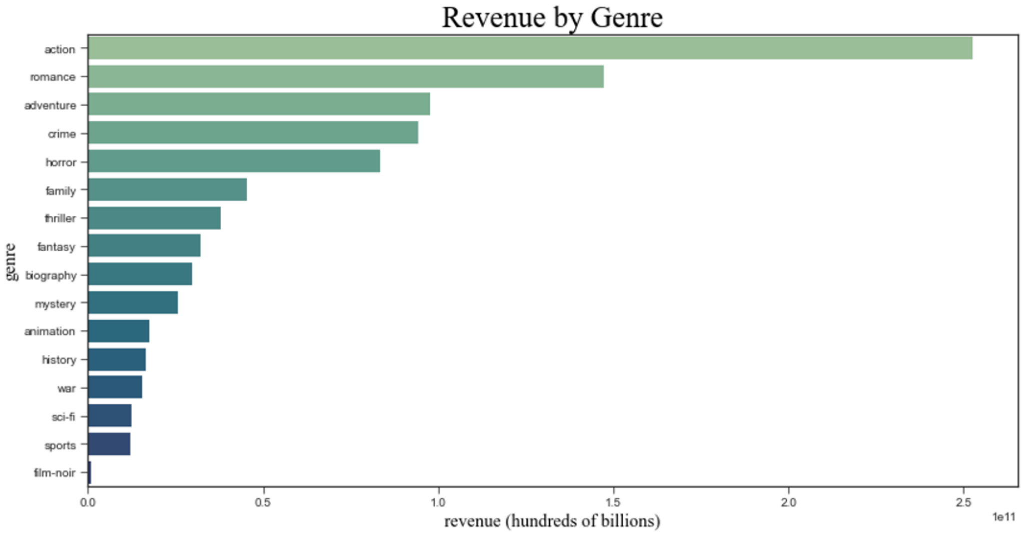
To begin with, exploratory data analysis was done on the number of movies per genre. The top five highest count of movies was action, thriller, romance, crime, and horror.

Figure: Exploratory bar plot



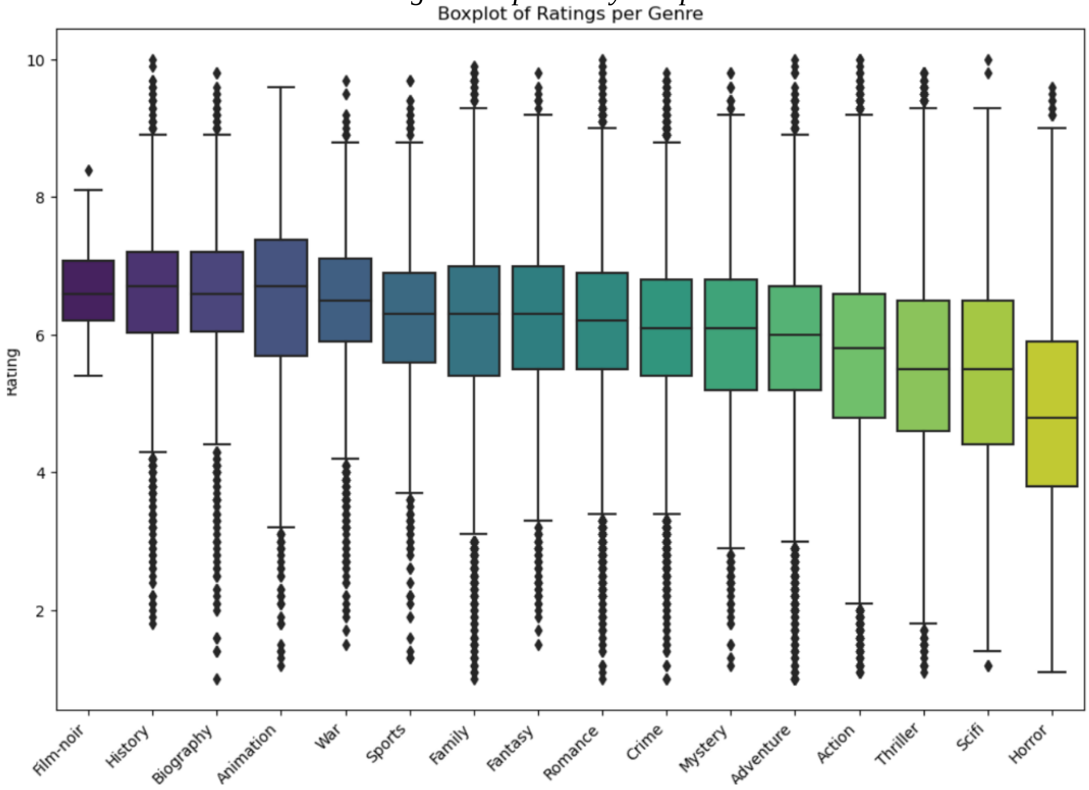
Action movies also have the highest revenue of all the genres, followed by romance, adventure, crime and horror.

Figure: Exploratory bar plot



Looking at the highest reviews per genre, the film-noir genre has the highest average rating while all other movies have a relatively larger spread of ratings. Horror movies have the lowest average ratings of all the genres.

Figure: Exploratory box plot



Models and Methods

Supervised machine learning, specifically multinomial Naïve Bayes classifier, Bernoulli Naive Bayes classifier, Support Vector Machine (SVM) classifier, and K-means clustering will be used to determine the outcomes of the genre classification models. The Naive Bayes models will utilize sklearn's CountVectorizer feature while the SVM models will utilize the same module's TFIDVectorizer feature along with three different kernels. These SVM kernels include linear, radial basis function (RBF), and polynomial. Additionally, the K-means clustering models will use two separate kernels (three and four).

An unsupervised machine learning technique known as Latent Dirichlet Allocation (LDA) will group features, specifically unique words, into topics based on similarities. Five different topics will be produced.

Prior to modeling, vectorization was applied to the data frame. The CountVectorizer's parameters included inputs being set to content, English stop words being removed, having all letters changed to lowercase, and max features being set to 300 solely for the Naive Bayes models. The Bernoulli model included all these mentioned parameters while also having the binary feature set to true.

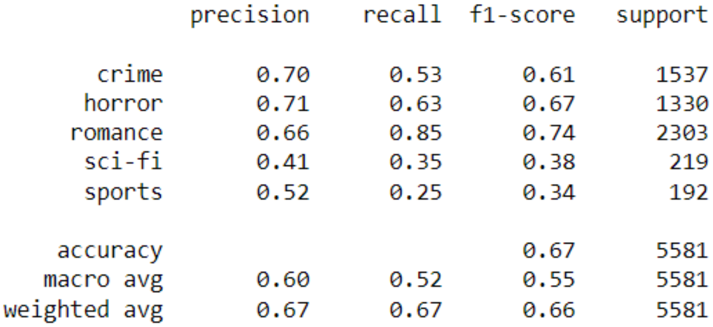
Regarding the Navie Bayes models, the entire data frame was sampled to only contain 30% of movie descriptions to ensure smooth analyses in a timely manner. Then, the data frame was split into training and testing data using sklearn's model selection feature. It was then split so that training and testing data would have 30% and 70% of the original data frame with a random state of 42. The data frame was split further so that one data frame with their respective labels and another with their respective unique words, for a total of four data frames.

Results

Model 1: Multinomial Naive Bayes

After instantiating and fitting the multinomial Naïve Bayes classifier with CountVectorizer to the vectorized data frame, a prediction was formed from the unlabeled test data based on log probabilities. This yielded an accuracy score of approximately 66.91%, which fell just below the industry standard (70-90%). A classification report was compiled to analyze the model's precision, recall, F1-scores, and support. Precision represents the accuracy of correct predictions, recall represents the completeness of correct predictions, F1-scores measures predictive performance, and support refers to the number of instances relative to each class.

Figure: Classification Report for Multinomial Naïve Bayes

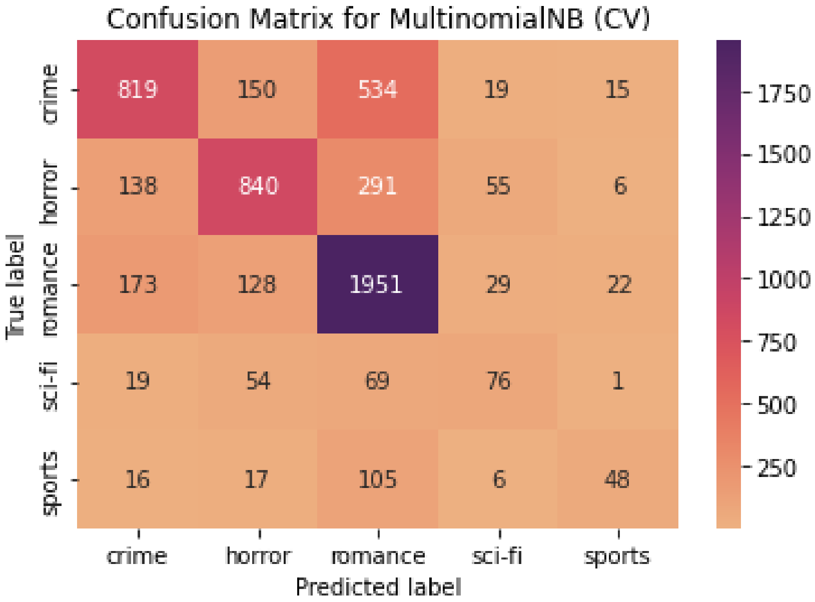


```
              precision    recall  f1-score   support

       crime       0.70      0.53      0.61      1537
      horror       0.71      0.63      0.67      1330
     romance       0.66      0.85      0.74      2303
      sci-fi       0.41      0.35      0.38       219
      sports       0.52      0.25      0.34       192

    accuracy                           0.67      5581
   macro avg       0.60      0.52      0.55      5581
weighted avg       0.67      0.67      0.66      5581
```

The resulting classification report showed that all three scores were about the same (60-67%) in terms of weighted average. The precision, recall, and F1-score values for each genre yielded higher scores for those with larger support compared to those with smaller support. In terms of precision, crime and horror scored the highest, followed closely by romance. Regarding recall, romance had a score of 85%, well above the industry standard (60-80%). Romance also secured the top score (74%) in terms of F1-score, with anything about 70% being deemed as good. The two genres that essentially weighed down the averages were sci-fi and sports with scores ranging from 25-52% across all percentage-based metrics. This could be attributed to their significantly smaller support compared to the other genres.

Utilizing the labeled testing data and the prediction model, a confusion matrix was created to display the correct predictions made by the model.

Figure: Confusion Matrix for Multinomial Naïve Bayes

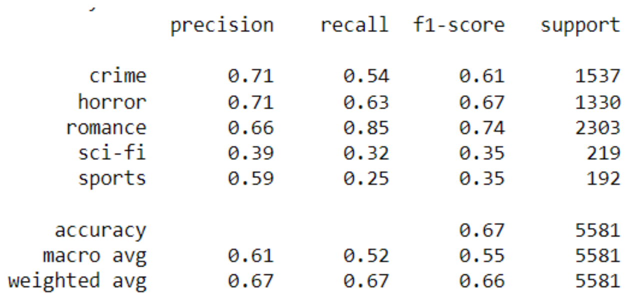


The matrix shows that of the 5,581 descriptions in the testing data, 3734 were predicted correctly. 819 crime descriptions were predicted correctly while 718 were incorrectly predicted. 840 horror descriptions were predicted correctly while 490 were incorrectly predicted. 1951 romance descriptions were predicted correctly while only 352 were incorrect. 76 sci-fi descriptions were predicted correctly compared to 143 incorrect predictions. Lastly, 48 sports descriptions were predicted correctly compared to 144 being incorrect.

Model 2: Bernoulli Naive Bayes

After instantiating and fitting the Bernoulli Naïve Bayes classifier with CountVectorizer to the vectorized data frame, a prediction was formed from the unlabeled test data based on log probabilities. This yielded an accuracy score of approximately 66.87%, which fell just below the industry standard (70-90%) and was almost identical to the multinomial Naive Bayes model. A classification report was compiled to analyze the model's precision, recall, F1-scores, and support.

Figure: Classification Report for Bernoulli Naïve Bayes



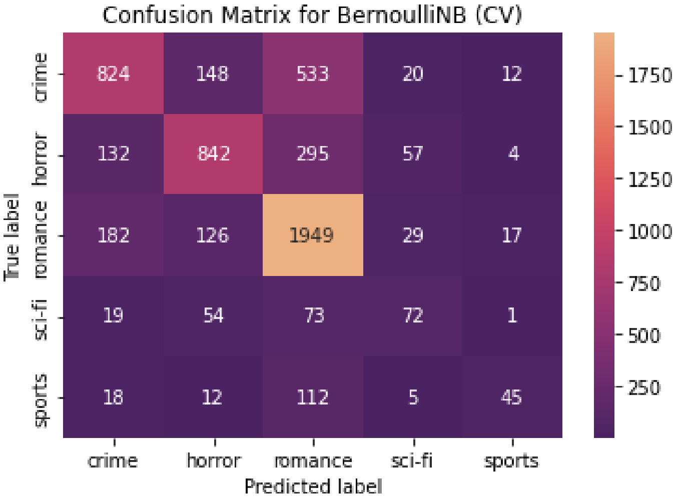
```
              precision    recall  f1-score   support

       crime       0.71      0.54      0.61      1537
      horror       0.71      0.63      0.67      1330
     romance       0.66      0.85      0.74      2303
      sci-fi       0.39      0.32      0.35       219
      sports       0.59      0.25      0.35       192

    accuracy                           0.67      5581
   macro avg       0.61      0.52      0.55      5581
weighted avg       0.67      0.67      0.66      5581
```

The report shows that all three scores were about the same (60-67%) in terms of weighted average. Like the previous model, precision, recall, and F1-score values for each genre yielded higher scores for those with larger support than those with smaller support. While the results were nearly the same as the previous model across the board, the differences were this model's precision scores for the sci-fi and sports genre. Sci-fi had a precision score of 39%, which was a 2% decrease from the previous model, and sports had a precision score of 59%, which was a 7% increase compared to the previous model.

In terms of precision, crime and horror scored the highest, followed closely by romance. Regarding recall, romance had a score of 85%, well above the industry standard (60-80%). Romance also secured the top score (74%) in terms of F1-score, with anything about 70% being deemed as good. The two genres that weighed down the averages were sci-fi and sports with scores ranging from 25-59% across all percentage-based metrics. This could be attributed to their significantly smaller support compared to the other genres.

Utilizing the labeled testing data and the prediction model, a confusion matrix was created to display the correct predictions made by the model.

Figure. Confusion Matrix for Bernoulli Naïve Bayes



	crime	horror	romance	sci-fi	sports
crime	824	148	533	20	12
horror	132	842	295	57	4
romance	182	126	1949	29	17
sci-fi	19	54	73	72	1
sports	18	12	112	5	45

The matrix shows that of the 5,581 descriptions in the testing data, 3732 were predicted correctly. 824 crime descriptions were predicted correctly while 713 were incorrectly predicted. 842 horror descriptions were predicted correctly while 488 were incorrectly predicted. 1949 romance descriptions were predicted correctly while only 354 were incorrect. 72 sci-fi descriptions were predicted correctly compared to 147 incorrect predictions. Lastly, 45 sports descriptions were predicted correctly compared to 147 being incorrect.

When comparing the results of both Naive Bayes models, it appears that they are both nearly identical. Their similar accuracy score implies that the features are not highly correlated and that the genres were well separated by movie description features. The low classification report scores for sci-fi and sports were most likely due to low support.

Model 3: Support Vector Machine (SVM)

Support Vector Machine (SVM) models offer a potent approach for classifying movie genres based on textual descriptions. Initially, the textual descriptions were transformed into numerical representations through techniques like TF-IDF vectorization, accompanied by thorough data cleansing to remove extraneous elements such as stop words.

As part of the initial feature extraction, each movie description was converted into a feature vector in which each dimension corresponded to a unique term or word. This step ensured that the text data was formatted appropriately for SVM model training.

The SVM classifier was trained on the feature vectors derived from the movie descriptions, with genre labels serving as the target variable. Throughout this stage, experimentation with three kernels – linear, radial basis function (RBF), and polynomial – alongside various costs were conducted to determine the optimal approach for the dataset. The linear kernel, while simplistic, was expected to offer the most computationally efficient model. In contrast, the RBF and polynomial kernels were hypothesized to yield higher accuracy scores after precise parameter tuning.

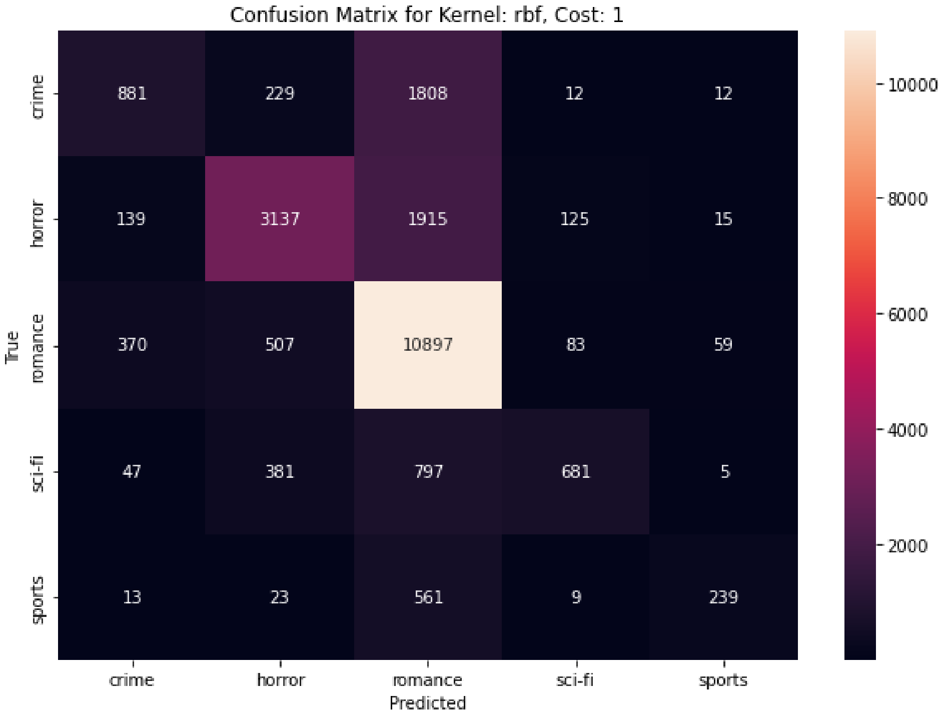
Kernel	Cost	Accuracy	Accuracy (approx.)
Linear	0.1	0.8226193070385705	82.26%
Linear	1	0.8264545652647636	82.65%
Linear	10	0.8277620396600566	82.78%
RBF	0.1	0.8140335585094792	81.4%
RBF	1	0.8348659838744825	83.49%
RBF	10	0.7933318805840052	79.33%
Polynomial	0.1	0.6707125735454347	67.07%
Polynomial	1	0.7256264981477446	72.56%
Polynomial	10	0.6805186315101329	68.05%

Based on the results, the effectiveness of various SVM model configurations can be assessed at a high level. Generally, increasing the cost parameter for the linear kernel from 0.1 to 10 led to a slight improvement in accuracy. This is because higher values of the cost parameter within the linear kernel allow the model to penalize misclassified points more heavily, leading to a tighter decision boundary (and potentially better generalization). However, the recorded improvements in accuracy were relatively small across different cost values. This suggests that the linear kernel may have reached its performance limit on this dataset.

Regarding the RBF kernel, the accuracy score improved notably when transitioning from a low-cost value of 0.1 to a higher cost value of 1. This indicates that a moderate cost value was potentially more suitable for this kernel on this dataset. In other words, a stricter margin (achieved with a higher cost parameter) was more effective in capturing relationships between features and labels. Despite this observation, increasing the cost parameter further to 10 resulted in a decrease in accuracy. This could indicate that the decision boundary became too rigid, leading to overfitting on the training data and reduced generalization.

Finally, the polynomial kernel displayed a generally lower accuracy compared to that of the linear and RBF kernels across all cost values. This could be attributed to the nature of the dataset as well as the specific parameters used. Increasing the cost value from 0.1 to 1 led to a significant improvement in accuracy, suggesting that a moderate cost value allowed was more appropriate for the polynomial kernel. Ultimately, increasing the cost parameter to 10 resulted in a decrease in accuracy, similar to the behavior observed with the RBF kernel. This indicates that the decision boundary may have been too complex which led to overfitting.

Figure: Confusion matrix for SVM (highest performing configuration)

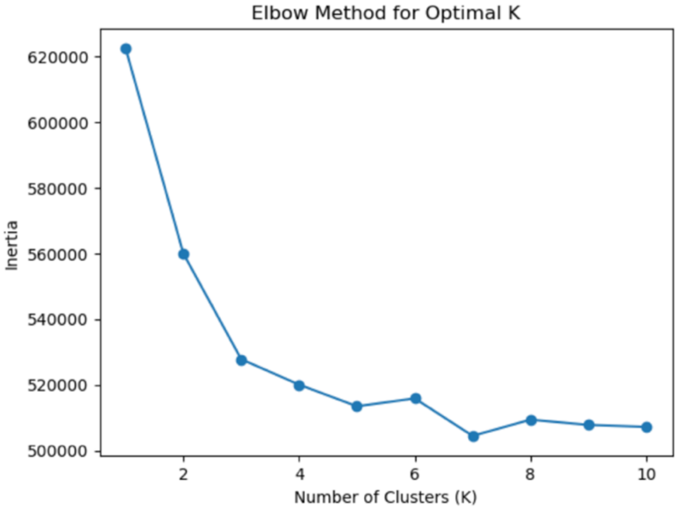


Model 4: K-Means Clustering

K-means clustering is a model that groups together similar data points into clusters to find patterns in the data. The clusters are decided upon beforehand, specifically using the elbow method. For each cluster a centroid is placed among the data points, and the goal of the model is to have the least amount of distance between centroid and data point. The model will then examine the points compared to the centroid to determine if it’s the best fit, if not it will reassign the centroids. The model will continue to do this step until it finds the least amount distance between the centroids and data points based on the number of clusters.

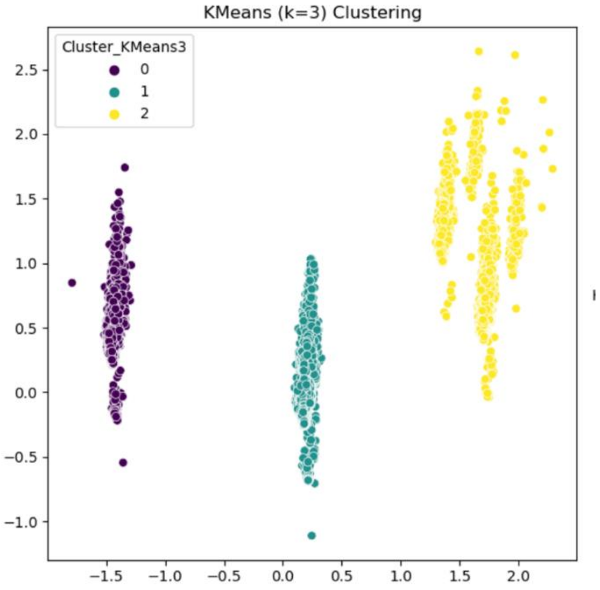
The data set was split into testing and training in a 70/30 split. The testing set also had the labels dropped as K-means is an unsupervised method. Once the data was selected, deciding how many clusters to use was next. The elbow method was used to determine the optimal number of clusters for the mode. It first assigns several clusters and the sum of squared distances of the datapoints to their nearest cluster center. Then the values are plotted, and the optimal number of clusters is found at the "elbow point" of this curve of the graph. The bend in the curve represents the point where increasing the number of clusters no longer gives significant improvement in fitting the mode. For the movie dataset, the optional number of clusters is either three or four.

Figure: Elbow method plot (K=3)



K means was then instantiated using three clusters to start, and the data frame was normalized. The results of the model were then plotted to show the three clusters.

Figure: Clustering plot (K=3)



Based on the plot, we can see that cluster 0 and 1 have a lot of similarities with a few outliers. However, cluster 2 is very spread out and doesn't share as many similarities among the movie genres. If a viewer watched a movie from 0, and wanted another movie from this cluster, the recommendation would be a good match. This also applies to cluster 1, where the next recommendation would be a good match. However, given the disparity among the last cluster, it may be difficult to provide an accurate recommendation for the view. Reviewing the silhouette score of 0.16 further suggests that while the clusters are distinct enough to be separate, the separation is not particularly strong. This means that that the descriptions contain ambiguous terms or that the genres are not distinctly different in the features used for clustering.

Figure: Silhouette visualizer plot (K=3)

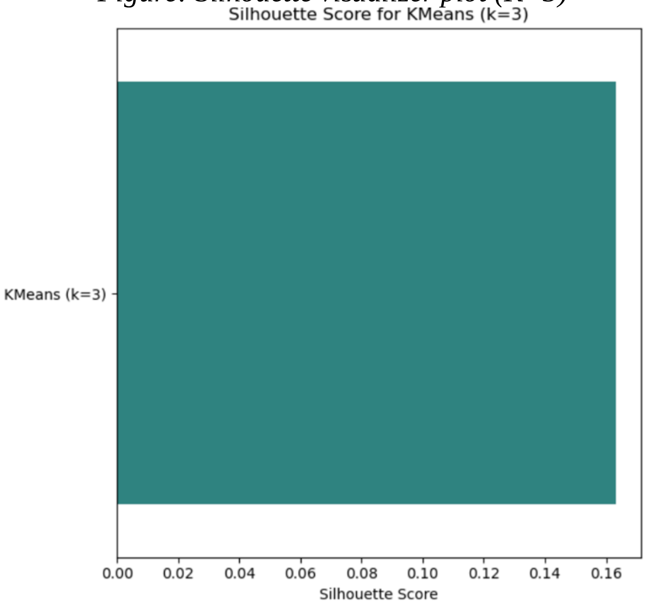
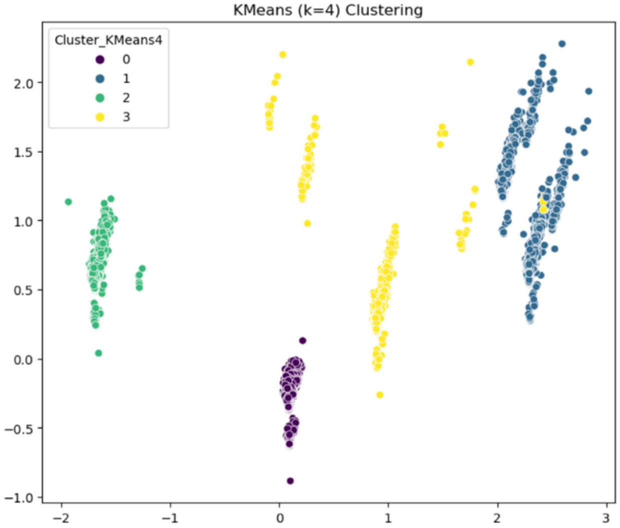
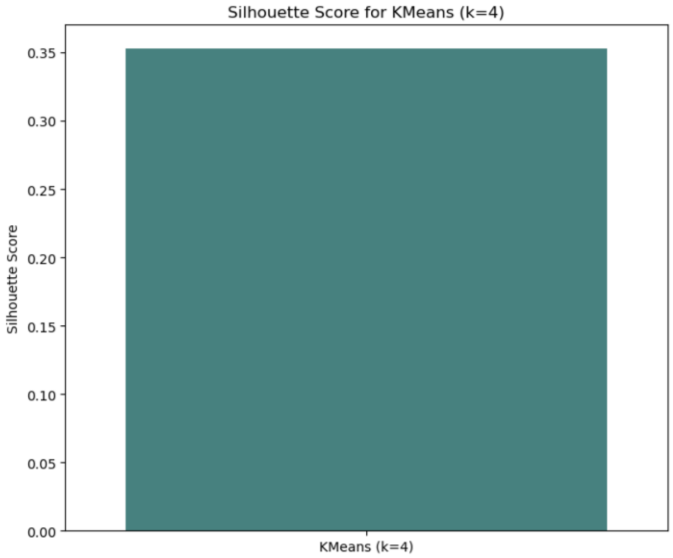


Figure: Clustering plot (K=4)



The model was also run with 4 clusters as the elbow seems to plateau between 3 and 4 clusters. There are improvements in the clusters, with the addition of a new clustering absorbing some of the data points that were found in other clusters. Cluster 1 does have some overlap with cluster 3 but is otherwise more defined compared to the other plot. Cluster 3 has a relatively wider spread, indicating there may not be as many similarities among this cluster. Cluster 0 and 2 contain some outliers, but otherwise have a more defined shape which does indicate some similarities in the words found in the description.

Figure: Silhouette visualizer plot (K=4)



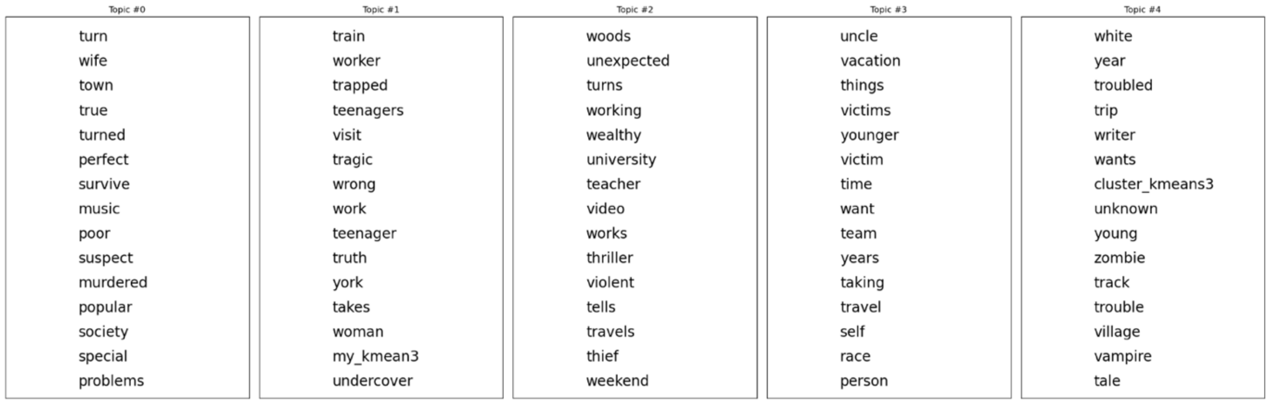
Reviewing the silhouette for 4 clusters the score has improved to 0.35. This score suggests that, on average, the clusters are reasonably well separated and that the points within each cluster are relatively close to each other.

Model 5: Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) is an unsupervised machine learning model, that informs us of the topic(s) of our documents. LDA has “assumptions” that topics will have similar words and it uses the grouping of those words to predict the topic. It works by taking the probability of a word belonging to a topic. The output shows how many words appear for each topic in the text.

Based on the LDA model that was run, it is difficult to identify a clear genre for either topic. Based on the image, Topic 0 could be crime films. Words like *suspect*, *murdered* and *survive* would align best with movies containing criminal elements. Topic 2 could be related to horror movies. Words like *thriller*, *violent*, *woods,* and *unexpected* indicate something fearful which is the basis of the horror genre. Topic 3 could potentially be about sports movies, with words like *team*, *race*, and *time*. Topic 4 seems best aligned to sci-fi with *zombie*, *vampire* and *tale*. Topic 1 does not have a clear genre; however, based on words like *woman*, *tragic*, and *teenager* the topic could be pertaining to romance movies.

Figure: Topic groupings



Topic #0	Topic #1	Topic #2	Topic #3	Topic #4
turn	train	woods	uncle	white
wife	worker	unexpected	vacation	year
town	trapped	turns	things	troubled
true	teenagers	working	victims	trip
turned	visit	wealthy	younger	writer
perfect	tragic	university	victim	wants
survive	wrong	teacher	time	cluster_kmeans3
music	work	video	want	unknown
poor	teenager	works	team	young
suspect	truth	thriller	years	zombie
murdered	york	violent	taking	track
popular	takes	tells	travel	trouble
society	woman	travels	self	village
special	my_kmean3	thief	race	vampire
problems	undercover	weekend	person	tale

K-means clustering highlighted three distinct clusters, with one cluster showing much more variability than the other two. This model has a low silhouette score, making it less than ideal to be used for predicting and recommending movies.

While LDA is effective at providing a general idea of the top features per topic, it was unfortunately not able to provide definitive genres classifications. This is likely due to the overlap between the text descriptions and the fact that movies can belong to multiple distinct genres. Therefore, LDA may not be a very accurate model to use for a recommendation system.

Conclusion

The model outputs have shown that genre identifiers can indeed overlap, and that the description data may not be enough to correctly predict and suggest a movie genre matching a viewer's preferences. Creating a consistently reliable recommendation system is also made more difficult by the fact that one movie can fit into multiple genre categories, posing a challenge when aiming to associate the movie with one singular genre. Thus, it is crucial to further refine the recommendation system and ensure that it generates precise suggestions by integrating additional details like movie ratings, cast/crew information, and user activity.

The data exploration and subsequent analysis using different machine learning models such as Multinomial Naive Bayes, Bernoulli Naive Bayes, SVM, K-Means Clustering, and LDA provides a high-level overview of the strengths and limitations to each approach. While each model offers unique insight into the dataset, they collectively highlight the complexity behind recommending movie genres based solely on movie descriptions.

Results from each model highlight the performance of techniques used in the analysis, with SVM demonstrating a superior ability to classify movie genres accurately. However, the challenge of genre overlap and the multifaceted nature of movie descriptions underscore the limitations of relying exclusively on text mining for recommendation purposes. The experimentation with K-Means Clustering and LDA further emphasize the difficulty of capturing the nuances of viewer preferences through unsupervised learning models.

In essence, the endeavor to build an effective movie recommendation system highlights the necessity of utilizing techniques beyond traditional text classification of genres. Future text mining efforts conducted in this space should advocate a more holistic approach, leveraging more data points. The journey towards optimizing these algorithms is ongoing; however, it is evident that the future of content discovery lies in harnessing data in order to cater to the varied tastes of audiences worldwide.