2024

Utilizing Multinomial Naive Bayes for Enhanced Movie Genre Classification and Analysis

BY

SREECHARAN VANAM - 11544661
DURGA VINAY GULLAPALLI - 1175950
SAI ALEKHYA SAJJA - 1178303

Abstract

In the constantly developing multimedia entertainment industry, particularly classifying movies into genres is a challenging yet essential task for more effective user suggestions and content management. This work provides a unique approach to this issue through using the Multinomial Naive Bayes classifier on a mixed and diverse movie dataset. The project focuses on predicting movie genres utilizing a variety of factors, such as descriptions, ratings, and other pertinent metadata. The process starts with significant data preparation and feature extraction, followed by the implementation of the classifier, which is well-known for its effectiveness in text classification as well as handling categorical data employing multinomial naïve bayes classifier.

The study's essential significance is establishing the classifier's ability to consistently predict multiple movie genres, suggesting its potential in the digital content space. The classifier's performance when compared to typical genre classification methods is optimistic, indicating how it has the potential for enhancing content recommendation systems and media analytics. This study not only offers insight into the usefulness of machine learning in media classification, but it also provides way for further analyses into optimizing these methods for the entertainment business. The system will be developed as a Machine Learning Project using Python for data cleaning, model building and analysis.

Contents

Chapter 1

Introduction

As part of the digital revolution, the diversification of multimedia entertainment methods has been on the rise. The increase of streaming platforms has led to having a large number of content available from movies or series, and it becomes more challenging for the viewers to search for particular show or movie from such a large amount of content without any labels or classifications. One of the most important approach to classify movies into genre will be helpful to optimize user experience. Therefore, the problem of movie genres classification is important in movie recommendation service for reducing time searching for the contents. Classifying movies into genres is challenging because the movie contents are diverse and pervasive. This project is aimed to solve this problem by applying the model known as the Multinomial Naive Bayes classifier (MNB) for text and categorical features.

Statement of the Problem

The rise of the digital age and practically endless access to the same – theoretically at least – has ushered in a period of immense variety when it comes to watching movies, with thousands of films in a vast number of genres and subgenres to choose from. With so much available, and so many different ways to classify films, one would think we would have accuracy in recognising and categorising them well. But with detailed and nuanced plot and thematic elements, directorial styles and subjective and highly personal audience interpretations, the sheer complication of film beyond traditional genre methods means that categorisation really is more complex than ever. The development of genres over time – evolving and even merging with each other – is another challenge that an accurate classification and recommendation system would

have to account for. The old methods of manual tagging and crude algorithms used by the industry shouldn't extend into the commercial present.

Significance of the Study

Improving the category classification of genre systems have far-reaching implications for both users and content providers. For users, it addresses the current need of a more sophisticated and targeted content discovery mechanism for the multimedia entertainment industry. With better genre classification, we would also be able to better facilitate an intuitive 'browsing-and-discovery' experience to users for our digital content delivery system. For academia, it also contributes towards the ongoing discussion on 'media studies' and machine learning: in our current scenario, when technological tools have become more readily available to interpret complex content, it is important to understand both the capabilities and inherent limitations of these current tools. The potential for deeper and more effective recommendations and personalised deliveries in content from the data collected would then be an indirect yet tangible outcome of this research.

Purpose of the Study

Overall, the purpose of the research is to understand how the Multinomial Naive Bayes classifier can be used in movie genre classification from a large dataset containing movie descriptions, ratings, viewer demographics, and other metadata. The ultimate aim is to improve the accuracy of genre classification and also discover more about what genre actually means – how is it defined, understood and represented in film – as a preliminary step towards better and more customised, accurate and adaptive systems for recommending content to users of digital content platforms.

Research Hypothesis

This work builds on the hypothesis that the Multinomial Naive Bayes classifier, due to its relative ease in working with both text and categorical data, will massively enhance the movie genre classification task, surpassing previous approaches not only in terms of precision but also in terms of better mapping the delicate and usually quite subjective terrain of film genres, hence producing higher quality content recommendations to viewers.

Research Limitations

Despite this striving for completeness and accuracy, it is important to recognise that the work is constrained by many limitations. The representativeness and diversity of the data used in the training of the model will determine how well the trained system will generalise from one tradition of filmmaking to another, and to different demographics of viewers as well. The Multinomial Naive Bayes classifier also makes an assumption of feature independence; that is, it assumes that the different features of the movie that contribute to its being categorised within a particular genre have no meaningful relationship to one another. But genre itself is fundamentally elusive; the borders and boundaries between genre classifications are necessarily fuzzy and ambiguous.

Chapter 2

Background

Classical Solutions and Their Limitations

Conventional genre categorisation approaches have largely been based on the use of labour-intensive manual labelling, as well as automated systems relying on complex sets of verbal rules. The drawbacks to these methods are numerous, including the fact that this type of categorisation requires substantial amounts of human judgment and time for the creation of meaningful categories. This commonly leads to inaccurate genre assignments that don't capture the grainy specificity of content that we've come to expect in regard to films. The problem with conventional approaches [4] becomes even more pronounced as we embrace the idea of ever-expanding volumes of (wildly diverse) types of available content.

Decision Trees

In their tree-like, simple, rule-based structure, Decision Trees [1] provide perhaps the most intuitive way to classify. The problem with them, though, is that the way they overfit data (particularly when the data is as diverse and complicated as movies) means that they might work great on the training data, but when new, unseen data is presented to them, they can significantly lose accuracy while dealing with the movie genres nuances.

Support Vector Machines (SVM)

SVMs [2] are particularly effective when dealing with high-dimensional, complex data, such and require a fine-tuning of parameters that is too challenging for large, diverse datasets that characterise the movie industry. More importantly, SVMs are binary classifiers, and any multi-

class scenario – such as a task like genre classification – necessitates complex workarounds to handle more than two outcomes.

K-Nearest Neighbors (KNN)

Because KNN [3] is quite simple and non-parametric (thus relatively robust across a wide range of applications), it is often the favoured technique for classification tasks. However, the curse of dimensionality and sensitivity to noise generally hurt the results of KNN in data sets for movies. Because KNN compares object based on distance, its performance can suffer in spaces where high-dimensionality makes it harder to define what ‘close’ means.

Advantages of the Multinomial Naive Bayes Approach

Finally, the next classifier, Multinomial Naive Bayes, is a more detailed solution than probabilistic and, because it is inherently probabilistic, much better at handling the built-in linguistic ambiguities of movie descriptions and information, and can theoretically enable bottom-up genre classification – understanding the constituents that generically define a movie genre – as well as top-down techniques. It also has several computing advantages over the previous methods and also can be applied on a larger scale. It’s the basis of information processing and numerical analysis in a range of digital domains that deal with petabytes of data, including movies More precisely, the Multinomial Naive Bayes classifier is a step forward, overcoming some of the shortcomings of the old techniques regarding genre classification of movies. It’s a step towards a more precise and dynamic genre taxonomy of movies.

Chapter 3

Model Architecture

Overview

A Multinomial Naive Bayes (MNB) classifier is a probabilistic learning procedure, like the one we looked at previously in text classification. MNB is well-adapted because it uses features that are categorical or discrete (such as word counts or frequencies in text), and the 'naive' assumption getting baked into the model is that the predictive features are independent of each other given a class label - a simplification that allows efficient computation [5] of the conditional probabilities required to make decisions.

$$\mathbf{Pr}(\boldsymbol{W} \mid C_k) = \frac{(\sum_{i=1}^{n} \boldsymbol{w}_i)!}{\prod_{i=1}^{n} \boldsymbol{w}_i!} \times \prod_{i=1}^{n} p_{ki}^{w_i}$$

Figure 3.1: General Multinomail Naïve Bayes Classifier

Preprocessing and Vectorization: The raw text data is converted into matrix format as per the Term Frequency-Inverse Document Frequency (TF-IDF) vectorisation method. Which means, the raw descriptions of the films are ultimately saved in such a format so that we can now employ it in the next relevant layer.

Probability Estimation: For each genre (class), a model can estimate the probability of observing a given term (in descriptions of that genre). For each genre, (the model can also estimate the probability that an actual topic described by that genre was indeed the actual topic). All these probabilities are based on checks involving training data (for each of the genres).

Prediction: Given a description of a new movie and its TF-IDF vector, multiply the vector by the (log) probability for each term in each class and sum the (log) prior probability of the classes. Choose the class (genre) with the highest (log) probability to be the predicted genre of the movie.

Comparison with different models

Decision Trees and K-Nearest Neighbors (KNN): Unlike decision trees which split decision areas along features which provide the highest information gain, MNB makes its predictions based on the feature-wise statistics, and because it assumes independence among the features, in some cases it is computationally less expensive to train, and can be faster compared with the decision trees especially when we work with text data. Unlike the KNN which computes the distance between instances and finds its neighbours in the input space, thus potentially suffering from the curse of dimensionality in highdimensional spaces (eg, sparse high-dimensional vectors like TF-IDF weights), the MNB is capable of working with high-dimensional inputs since it supports probabilistic statistics.

Support Vector Machines (SVM): SVMs find a hyperplane that best separates the two classes in feature space. In the real world, this process can be computationally expensive, especially for larger datasets, and requires careful parameter tuning. MNB, on the other hand, uses a much more straightforward probabilistic approach that is usually faster to compute and easier to set up, and can still perform robustly for text classification use-cases.

Performance advantages

For large datasets that contains textual description, the advantage of its simplicity and efficiency can be critical to allowing its use. Its model assumptions - words occur independently, so that the probability of occurrence of any particular word or feature is unaffected by the presence or

absence of other words - are just a simplifying assumption; in practice, this assumption has pretty good correspondence for text data where words do not in general appear simultaneously and so have some independence from each other, particularly if large vocabularies are considered. The feature independence, the third big advantage of MNB, allows the model to handle the complex high-dimensional nature that can occur in many text classification problems, and represents a powerful trade-off between classification accuracy and computational efficiency.

Environment Setup

A tested set of tools that will allow us to execute the Multinomial Naive Bayes (MNB) classifier project will be installed in the environment. The goals of this section are to specify and install the software and libraries necessary to load the dataset, train the MNB model, and evaluate the MNB model on the test data. In particular, this section will give instructions on how to setup the environment.

Python: Python is the one and only programming language used for this project. You should use Python 3.8 or newer to use all the latest libraries and features, but it will still work with older versions. Go to the Python website [7] to download and install Python.

Essential Libraries

Pandas: Pandas [8] will make your life easier when it comes to manipulating structured data (eg, CSV files): using fast data structures, it has optimised functions to load and manipulate data.

NumPy: Arrays and matrices of various dimensions up to millions with powerful mathematical functions operating on these arrays. One of the most important libraries for scientific Python computing [9].

Scikit-learn: A machine learning library [10] in Python, open-source. Includes algorithms for classification, regression and clustering, including the Multinomial Naïve Bayes classifier. Includes tools to support model fitting, data preprocessing, model selection and evaluation.

Matplotlib and Seaborn: Libraries for plotting and visualising the data [11], [12] – these packages are absolutely necessary for inspecting the dataset and the outcome of our work.

Development Environment

Jupyter Notebook: Provides an interactive computing environment that combines text and code and has the ability to publish documents that contain executable code, equations, visualizations and narrative text which is mandatory for data exploration and visualization [6].

Dataset

The dataset being used here has been collected from Kaggle [13]. This comprehensive and extensive data of movie and TV shows data-base that are accessed on the Netflix platform. It provides detailed information about the platforms running shows and movies that essentially have different genres, these will be analysed and used for training purposes.

Total Rows: 8,807

Columns: 12

The data includes fields that could be a title, director, actors, country, year and a description. The fact that we have this variety of fields allows us to recognise many different features that could be useful for our classification.

Data Cleaning and Preprocessing

Using basic data science operations, the dataset went through various preprocessing steps to make it usable for input into the Multinomial Naive Bayes model:

Filtering Type: Returned only results where the field type was 'Movie'. I made sure not to return anything for the field type 'TV show' so as to stay strictly within the topic of classifying movie genres.

Handling Missing Values: We drop entries lacking identification card information like 'listed_in' and 'description', as these fields are fundamental to the analysis and classification operations.

Excluding Specific Genres: Movies tagged with 'Stand-Up Comedy' were excluded from the analysis to curate a data set of genres leaning more in the direction of narrative structure, considering that the textual characteristics of stand-up comedy's tags might distort the model's learning process.

Final Dataset: The final dataset after cleaning has 5788 entries for a movie. This is one data point for training the machine learning process for recommendation purposes, with the cleaned data containing no null entries in the relevant features or columns for understanding whether a movie is similar in genre as another one. Hence, a robust dataset is now at our disposal for machine learning purposes.

Implementation

The step-by-step instructions required for implementing the model for the Multinomial Naive Bayes (MNB) classifier shows the entire follow-through from raw data to predictions and obtaining conclusions.

Workflow:

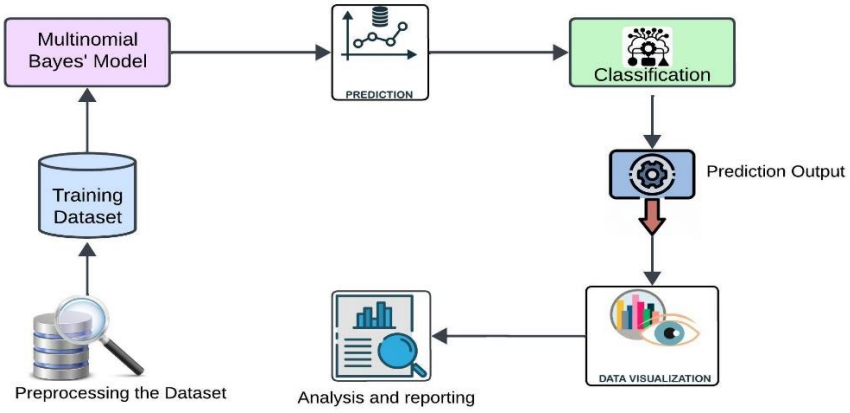

Figure 3.2: Workflow of Our Proposed Project

This process encapsulates an entire strategy for a more widely applicable model implementation - to continuously correct and improve the classification procedure for better accuracy and performance.

Preprocessing the dataset

1. Data Loading: First, the sample data-set is loaded into a Python script, using the Pandas library which makes working with tabular data easy.
2. Data Cleaning: The dataset will undergo preprocessing to remove irrelevant or empty entries, which is needed to ensure the quality of input training data.
3. Text Processing: Descriptions are first preprocessed to convert to lowercase, remove punctuation and stopwords, and potentially lemmatise (reduce to root) or stem the words.
4. Vectorization: Lastly, the descriptions of the movies are converted to a numerical vector based on TF-IDF vectorization that indicates how important is a word to a document in a corpus.

Model Training

1. Training the Model: At this stage, we train the MNB model on the processed set of data so that it learns to associate the frequencies of terms appearing in the descriptions with the corresponding genres.
2. Hyperparameter Tuning: After building the model, you tune its performance by adjusting some parameters called hyperparameters.

Classification and Prediction

1. Prediction: model will be trained, then used to predict the genre of test movies from their descriptions. Input is the vectorised test descriptions; output is the predicted genres.
2. Confidence Scores: Along with the prediction, for each prediction the model can provide probability scores that denote how confident the model is in each prediction, which could be harnessed for decision-making or analysis based on some threshold.

Data Visualization

1. Visualizing Results: we are able to see the outputs of our predictions as data visualisation techniques are applied to them. Matplotlib and Seaborn are just two of the many libraries you can utilise to represent your model's performance in confusion matrices, bar plots or any other kind of plot.
2. Interpreting Visualizations: The visualisations are interpreted as reporting performance and bringing out aspects of the model's functioning that inform the user about the structure or coherence of the dataset and the ability of the model to classify it.

Chapter 4

Model Evaluation

To assess the performance of Multinomial Naive Bayes (MNB) classifier to classify data into movie genres, various metrics were examined to determine its precision, recall and accuracy altogether. This evaluation was performed using the classification report generated from the test data, and also with the visualizations of the two comparisons of the model performance.

As the classification report shown in the Figure 4.1, for each category of genre we can find different metrics and support (number of true positive for each label).MNB had a high precision in some genres meaning that the model provided high effectiveness when it came to labelling these again with the same category. Recall, on the other hand, differs from genre to genre and reflected the capability of the model to recognise all the same instances of the genre.

accuracy			0.83	3620
macro avg	0.81	0.83	0.82	3620
weighted avg	0.82	0.83	0.82	3620

Figure 4.1: Accuracy of MNB in Classifying Movie Genres.

	precision	recall	f1-score	support
Action & Adventure	0.83	0.85	0.84	85
Action & Adventure, Anime Features, International Movies	0.99	1.00	0.99	66
Action & Adventure, Comedies	0.83	0.91	0.87	69
Action & Adventure, Comedies, Dramas	0.91	1.00	0.95	72
Action & Adventure, Comedies, International Movies	0.91	0.85	0.88	73
Action & Adventure, Dramas	0.77	1.00	0.87	72
Action & Adventure, Dramas, International Movies	0.78	0.64	0.70	74
Action & Adventure, International Movies	0.74	0.89	0.81	70
Action & Adventure, Sci-Fi & Fantasy	0.79	0.91	0.85	66
Children & Family Movies	0.75	0.71	0.73	70
Children & Family Movies, Comedies	0.88	0.65	0.74	79
Children & Family Movies, Dramas	0.90	1.00	0.95	72
Children & Family Movies, Music & Musicals	0.90	1.00	0.95	74
Comedies	0.91	0.76	0.83	68
Comedies, Dramas	0.96	1.00	0.98	79
Comedies, Dramas, Independent Movies	0.70	0.71	0.71	70
Comedies, Dramas, International Movies	0.47	0.29	0.35	70
Comedies, Independent Movies	0.96	1.00	0.98	77
Comedies, Independent Movies, International Movies	0.98	1.00	0.99	90
Comedies, International Movies	0.64	0.69	0.67	65
Comedies, International Movies, Music & Musicals	0.86	0.91	0.89	81
Comedies, International Movies, Romantic Movies	0.63	0.62	0.63	69
Comedies, Romantic Movies	0.70	0.92	0.79	60
Documentaries	0.69	0.32	0.44	75
Documentaries, International Movies	0.74	0.68	0.71	73
Documentaries, International Movies, Music & Musicals	0.92	1.00	0.96	82
Documentaries, International Movies, Sports Movies	0.82	1.00	0.90	70
Documentaries, LGBTQ Movies	0.90	1.00	0.95	70
Documentaries, Music & Musicals	0.83	0.88	0.86	77
Documentaries, Sports Movies	0.97	0.94	0.96	80
Dramas	0.82	0.66	0.73	76
Dramas, Independent Movies	0.80	0.80	0.80	74
Dramas, Independent Movies, International Movies	0.51	0.29	0.37	72
Dramas, Independent Movies, Romantic Movies	0.85	1.00	0.92	71
Dramas, Independent Movies, Thrillers	0.90	1.00	0.95	61
Dramas, International Movies	0.11	0.01	0.03	70
Dramas, International Movies, Music & Musicals	0.80	0.90	0.85	59
Dramas, International Movies, Romantic Movies	0.68	0.68	0.68	77
Dramas, International Movies, Sports Movies	0.94	1.00	0.97	72
Dramas, International Movies, Thrillers	0.80	0.65	0.71	79
Dramas, Romantic Movies	0.84	0.95	0.89	73
Dramas, Sports Movies	0.98	1.00	0.99	65
Dramas, Thrillers	0.84	0.93	0.88	72
Horror Movies	0.97	0.84	0.90	70
Horror Movies, International Movies	0.90	1.00	0.95	72
Horror Movies, International Movies, Thrillers	0.90	1.00	0.95	64
Horror Movies, Thrillers	0.94	0.96	0.95	81
International Movies, Thrillers	0.77	0.99	0.87	72
Movies	0.91	0.97	0.94	75
Thrillers	0.76	0.96	0.85	67

Figure 4.1: Accuracy of MNB in Classifying each Genre.

1. **Accuracy:** The accuracy of the MNB model we obtained by training and predicting is 0.83 showing that it was successful in correctly predicting the genres 83% of the time across so many diverse genres in the Netflix dataset.
2. **Precision and Recall:** For some genres, the precision gets as high as 1.00, using that genre's label: all the movies we predicted to be in it, were indeed that genre. Similarly, some genres yield a perfect recall score: the model got all the movies of that genre present the test data.
3. **F1-Score:** Measures the harmonic mean of precision and recall, and is useful when the class distribution is skewed. The MNB classifier's f1-scores reflected a similar pattern of performance between precision and recall, with an average macro score of 0.81 and weighted score of 0.82, across these different genres.

Confusion Matrices: The visualization of the Confusion Matrices has helped us with identifying True Positives, False Positives, False Negatives and True Negatives. We can observe the prediction score of each genre just by looking at the colour using a heatmap representation(the denser the color the higher the accuracies).

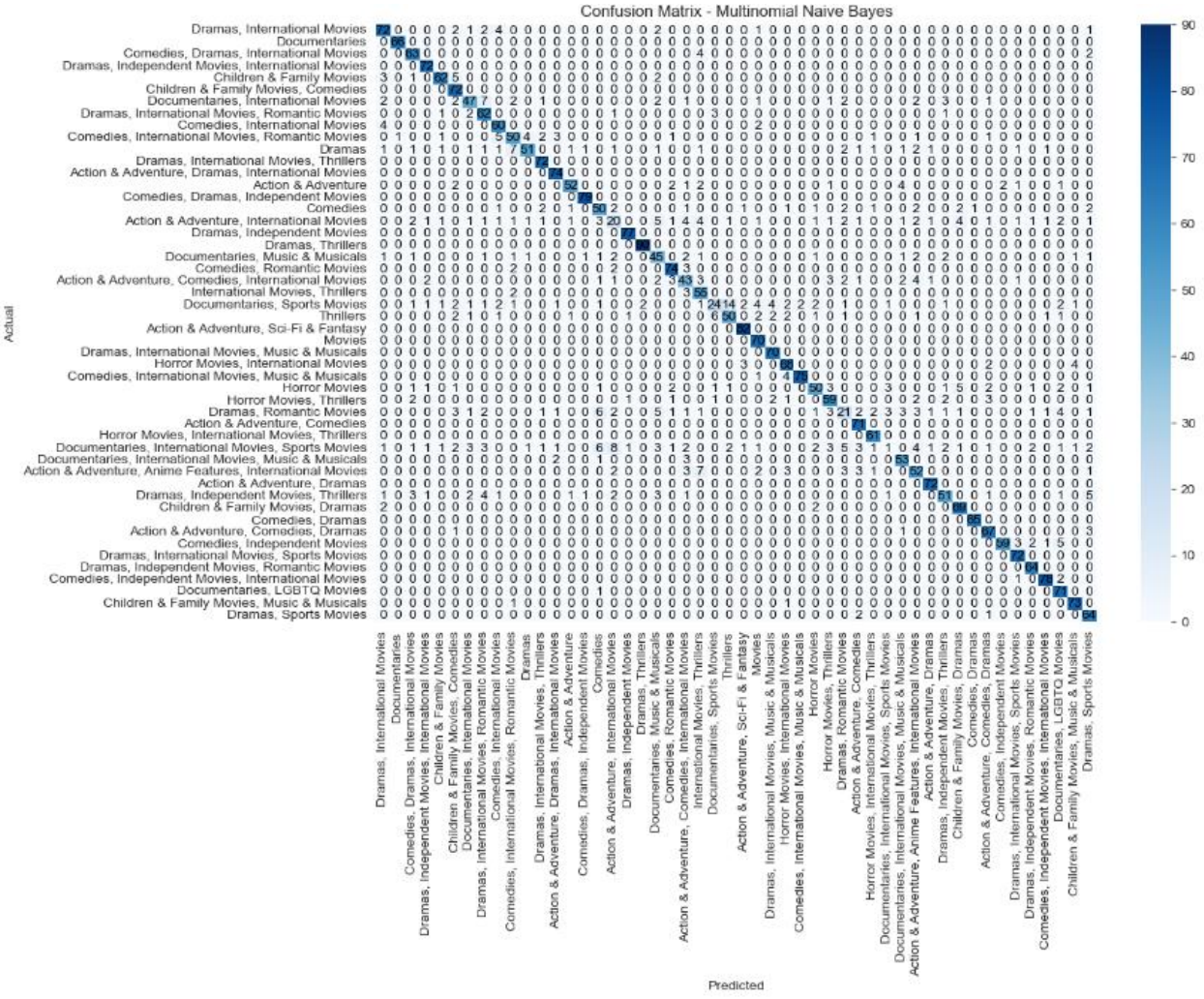

Figure 4.2: Confusion Matrix of MNB in Classifying Movie Genres.

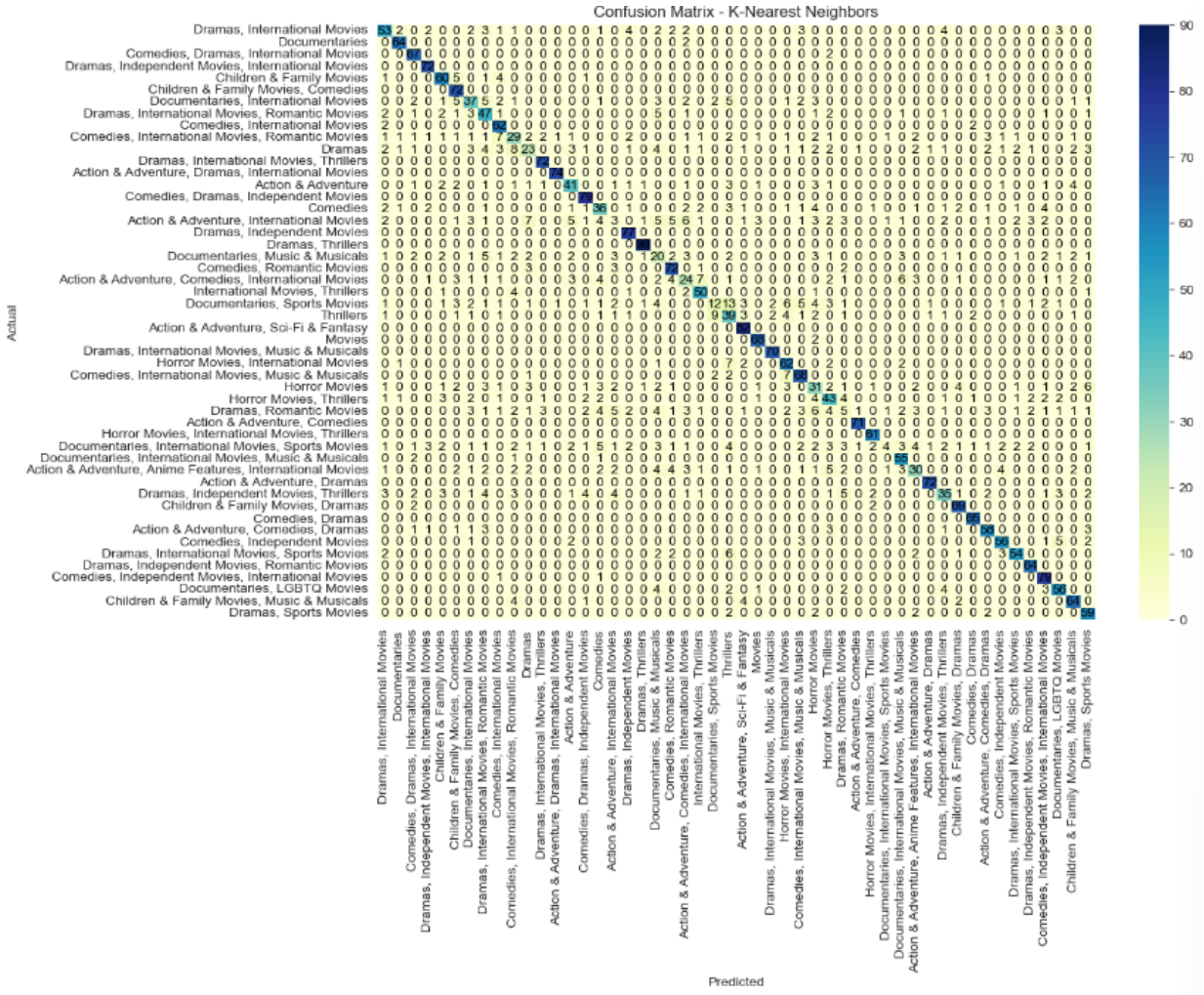

Figure 4.3: Confusion Matrix of KNN in Classifying Movie Genres.

Accuracy by Category: As we can observe from the Accuracy by Category graph below, there was a clear variation in the accuracy of the MNB model across categories. It is pretty obvious from seeing the results that our model performs good in common genres or most frequent genres in the dataset, but still there are some improvements needed to be done to handle less frequent genre combinations.

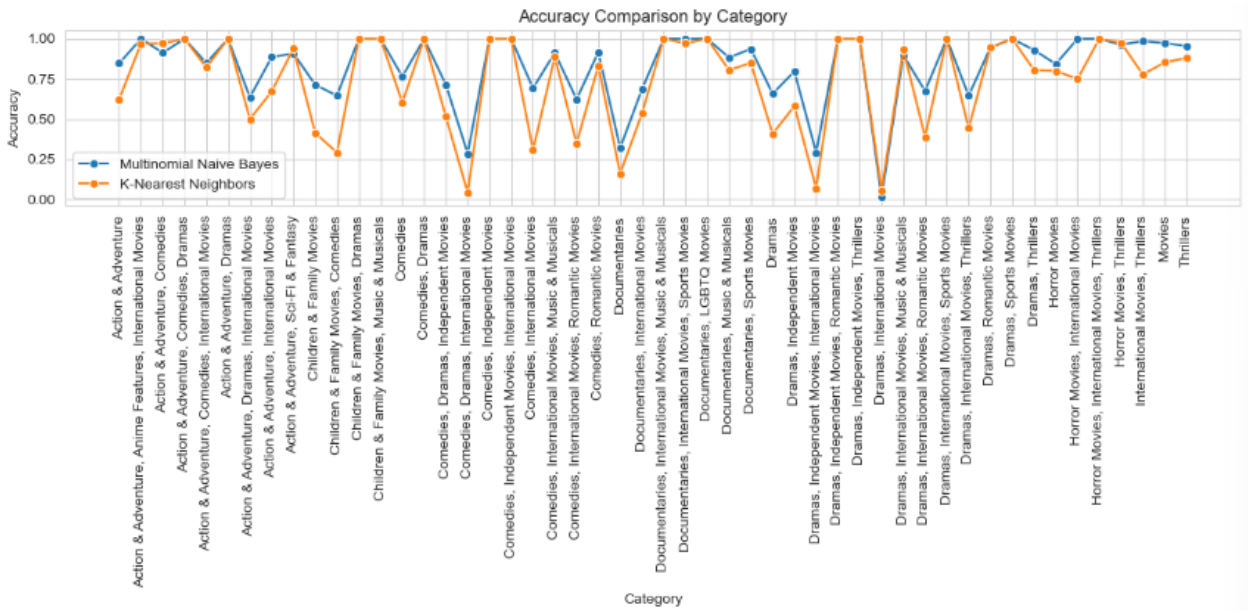

Figure 4.4: Line-Graph Comparison of each Category Classification accuracy of both models.

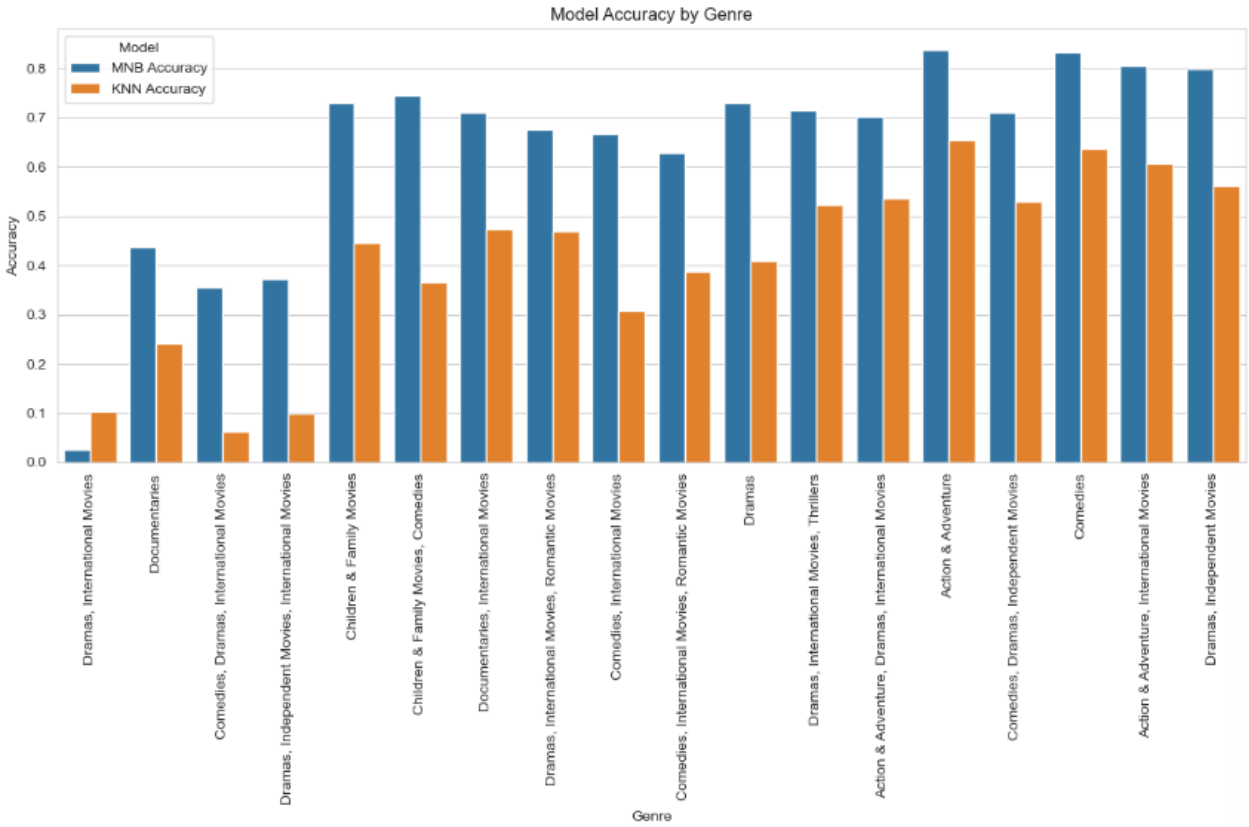

Figure 4.5: Bar-Graph Comparison of each Category Classification accuracy of both models.

Model Comparison: These measures were performed on a pair-wise comparison against K-Nearest Neighbors (KNN) to show that KNN is outperformed by MNB model on most category of the dataset. This is demonstrated by comparing line graphs and bar plots for the accuracy by category for both models.

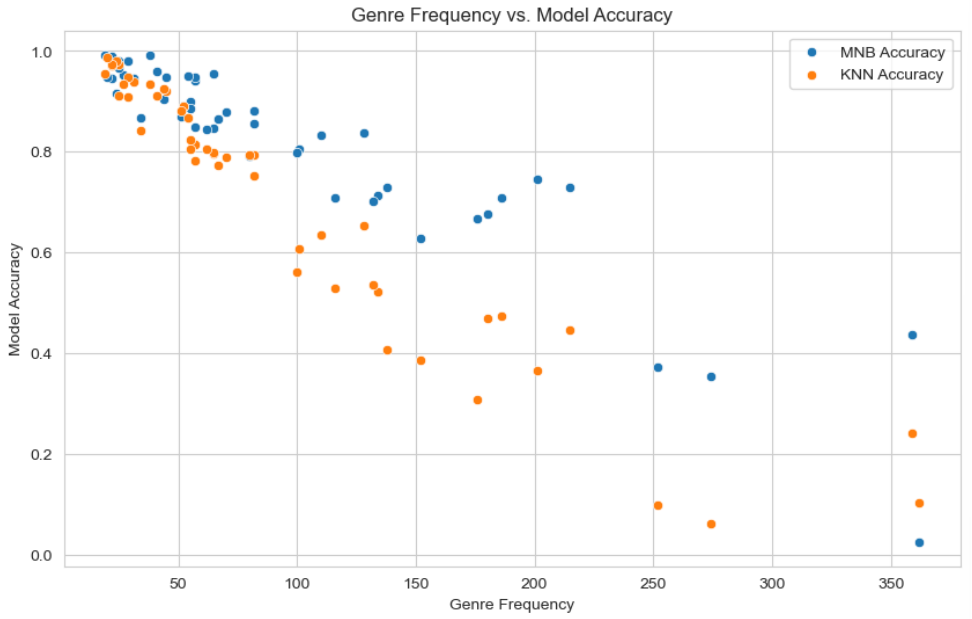

Figure 4.6: Scatter plot of Genre Frequency vs. Model Accuracy.

The scatter plot depicts how the genre frequency is related to model accuracy for both the Multinomial Naive Bayes (MNB) and K-Nearest Neighbors (KNN) model.Overall, it's clear that both the MNB and KNN model complement each other - both tend to perform better on genres with higher frequency. This suggests that the bigger the frequency of an example within a genre in the training set, the more accurate a model could be when predicting on unseen data. The cloud of points towards higher accuracy for the MNB model at all genre frequencies shows consistency in its performance across both common and rarer genres. This robustness in performance shows that the MNB model is well-suited for text classification given its sensitivity to sparseness of the dataset.

On the other hand, the points for KNN come across as scattered. The variability in its prediction tells us that KNN is quite fragile compared to the MNB model. Furthermore, comparing the highest accuracy point for KNN with that of MNB across all genres, the difference is striking. As such, text classification seems to be a perfect fit for MNB given that natural language data has a high dimensional space.

The best comparative measure of how well these different approaches perform was the results of the K-Nearest Neighbors model (i.e., the KNN model). The KNN model uses distance metric to make predictions. One of the primary problems that distance-metric based models face when working in a space with a large number of dimensions (i.e., the curse of dimensionality) is that the search space plays havoc with their performance. That was the case with the results on our genre task. The KNN model's overall accuracy might not be as bad as the KMeans model results but everything else is: it does well on those genres that contain more instances leading to a denser region of the feature space that one would use to make a prediction and poorly on the much sparser genres. This difference in performance highlights that you need to choose the right model for the right task. If you are working with the curse of dimensionality, then you are tasked with picking a distance-based model that will work well. Your options would be limited. If, on the other hand, your higher-dimensional problem admits of a probabilistic framing, then a good probabilistic model might be the right choice for the task. Had we been working with text data on this task, repeating the process for the MNB and KNN models, we would discover that the probabilistic model better serves the task than the distance-based model at hand.

To conclude, the MNB classifier is efficient for predicting movie genres from text descriptions. Existing features such as high-dimension text representation and multi-class problem are solved in just one model. That is why the MNB classifier is competent at handling content

recommendation problems. In the future, augmentation of the MNB classifier can be attempted on overfitting to under-represented genre problem in the current data, which can be addressed by advanced text processing or integration of more contextual features during the training.

Model Experiments

This was proceeded by a series of experiments, that aimed in exploring the degree to which the Multinomial Naive Bayes (MNB) classifier performed its job well, in general - digging deeper into the model's effectiveness across different genre-labels, and comparing against other classification approaches, for different datasets. Ultimately, these experiments offered both qualitative data and metrics to determine the range and mode of operation of this model - something that was highly desirable to gain a good understanding of this 'black box'. Each of these experiments was in service of this one lofty, but certainly desirable, goal: understanding how the MNB classifier works when tasked with the challenging job of assigning movie genres to textual descriptions.

Then, we examined the performance across different genres. The figure shows that our MNB classifier performed well for some genres with very high precision and recall value, suggesting a good performance of the model in not only correctly classifying instances in these genre within the dataset, but also in retrieving most of the instances of these genres within the dataset. Other genres, however, performed poorly - that is, these genres that are less frequently captured in the dataset perform with low performance metrics, suggesting the model's difficulty in classifying infrequent or nuanced genres.

Another important factor was how to deal with unbalanced data, as not all genres of theatre were equally frequent in the dataset, so some will necessarily have more training examples than

others. Using Random Over Sampling, the experiments showed how the model could improve its skills in predicting minority classes, at the risk of overfitting and performing better on training data than on unseen data.

A comparison with baselines (eg, Decision Trees, K-Nearest Neighbours, or KNN) suggests that, for this task and using the relevant metric (eg, the F1-score, which combines precision and recall into one number), the MNB classifier does better than most alternatives. Such a comparative path is obtainable, useful, and important for validating the use of the MNB model for this classification task. The model fits well the type of high-dimension sparse data one gets by vectorising text using TF-IDF.

The stability and sophistication of the model were further emphasised by the consistency in performance metrics across the genres and stability of the metrics across several runs with different test-train splits. This consistency was important to ensure that the model was learning 'real' patterns in the data, and not learning some sort of noise. Several methods of visualising the results helped determine what genre distributions overlap with one another the most. The confusion matrix, shown in Figure 4.3 and Figure 4.2 above in Chapter 4, is a useful tool that can illuminate these overlaps and show us what the boundaries of the model's predictive space are in action.

In general summary, the entire chain of experiments confirmed the suitability of the MNB classifier for the task of genre prediction. However, while it was possible to demonstrate how adequately the model copes with the task in hand - that is, its ability to predict a movie's genre from its description - the results were also rather telling regarding the difficulties yet to be coped with by such predictive models as low “true positive rate” of genres that are not well-represented in the training dataset, and the problem of overfitting.

Chapter 5

Conclusion

The project involved improving genre classification of movies using a Multinomial Naive Bayes (MNB) classifier to make content recommendation systems faster and more precise. As a result of the study, the MNB model was successfully implemented and evaluated for the task of textual data classification by showing its great ability to classify genres from movie text actions. A simple but potent classifier was engineered by employing rigorous preprocessing, training and testing methods of the movie descriptions.

These experiments revealed how well the model was performing – showing, for example, that it was well suited for handling high-dimensional data and that it was good at grouping together textual descriptions with similar genre features. High precisions and recalls for some genres demonstrated the relevance and utility of the genre classifier, while, at the same time, acknowledging the dangers of mislabelling were flagged by a high variability in performance across genres, due in part to class imbalance, but also because genre boundaries might not be as clear-cut as the model has to assume in order function.

In conclusion, MNB classifier is able to do automated movie genre classification. This also paves the way for further improvements in the way text is analysed by the multimedia entertainment industry. Language is complex, but despite that we can treat its complexity as the complicated noise that is hiding in the data and also make machine learning sense of more things. We can now say that using probabilistic models to classify movie genres is a good approach.

Future Work

For future research in studying movie genres, the landscape is rich in possibilities for expanding and refining the current model. Improving the Multinomial Naïve Bayes classifier's algorithm could consider adding ensemble methods, or hybrid models that combine the basic probabilistic approach with more complex neural network architectures - such models might be better equipped to handle subtleties in language that a more straightforward one could miss.

Further growing the datatset therefore will be a prerequisite - further, skew-balancing the set by providing a greater density of examples for underrepresented, lesser genres. Thanks to more data, difficulty of imbalance would thus mitigate. Refinement and generalisation across the space of genres should therefore improve when data is skew-balanced. Furthermore, we can leverage more recent advances in Natural Language Processing to help the classifier 'read between the lines'. For instance, models relying on semantic word embeddings [14] or more recently on transformer-based models [14] can help the model learn the semantic context of a movie description.

And we can also pursue the intriguing possibility of learning how adding further metadata could influence the model's treatment of genre: there are directorial styles, types of cast ensemble, and historical trends in movie production that could influence genre classification but that the model is just learning how to exploit.

Using user feedback to train and evolve the current system, makes it more dynamic and reflective of the viewer's own personal preferences, yet feeding it back again into the same feedback loop. Finally, you could test the classifier on cross-domain data sets, or even mix multiple types of features into a multimodal configuration.

Finally, as automation of decisionmaking become more famous, we'll necessarily begin to enquire into the process of how machine learning models are causing their conclusions. As such, we hope future work will also explore improving the explainability (and interpretability) of the classifier used. Explainable AI won't just serve to make users trust in automated decisionmaking, it will also be interesting. This might reveal the true multivalency of the set of features that define our favourite genres in movies.

References

[1] G. Ramadhan and E. B. Setiawan, "Collaborative Filtering Recommender System Based on Memory Based in Twitter Using Decision Tree Learning Classification (Case Study: Movie on Netflix)," 2022 International Conference on Advanced Creative Networks and Intelligent Systems (ICACNIS), Bandung, Indonesia, 2022, pp. 1-6

[2] Jair Cervantes, Farid García-Lamont, Lisbeth Rodríguez-Mazahua, Asdrúbal López Chau: A comprehensive survey on support vector machine classification: Applications, challenges and trends. Neurocomputing 408: 189-215 (2020)

[3] Mangolin, R.B., Pereira, R.M., Britto, A.S. et al. A multimodal approach for multi-label movie genre classification. Multimed Tools Appl 81, 19071–19096 (2022).

[4] Zakaria Suliman Zubi, Ali A. Elrowayati, Ibrahim Saad Abu Fanas. A Movie Recommendation System Design Using Association Rules Mining and Classification Techniques. WSEAS Transactions on Computers. 2022;21:189-199. 10.37394/23205.2022.21.24

[5] Multinomial Naïve Bayes' For Documents Classification and Natural Language Processing (NLP)

[6] Installing Jupyter: Get up and running on your computer

[7] Anaconda Software Distribution. (2020). Anaconda Documentation. Anaconda Inc. Retrieved from https://docs.anaconda.com/

[8] Pandas Library

[9] NumPy The fundamental package for scientific computing with Python

[10] scikit-learn: Machine Learning in Python

[11] Matplotlib: Visualization with Python

[12] seaborn: statistical data visualization

[13] https://www.kaggle.com/datasets/ginnyshai/netflix-dataset

[14] Unal FZ, Guzel MS, Bostanci E, Acici K, Asuroglu T. Multilabel Genre Prediction Using Deep-Learning Frameworks. Applied Sciences. 2023; 13(15):8665. https://doi.org/10.3390/app13158665