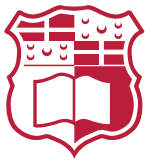
Analysing Script Features for Performance Forecasting in Film Production: A Data-Driven Approach Leveraging NLP

Karl Cini

Supervisor: Prof. John Abela

May 2025

Submitted in partial fulfilment of the requirements for the degree of M.Sc. Data Science.



L-Università ta' Malta
Faculty of Information & Communication Technology

Abstract

The film industry has always been an important entertainment avenue for audiences of all ages. Following the impact of the COVID-19 pandemic, the industry has undergone significant changes due to the rise of streaming platforms, yet thousands of films are produced every year and billions of dollars are generated annually worldwide. With the number of viewers expected to reach 1.9billion by 2029 [1] the film industry remains a strong inspiration for writers and producers alike.

Despite the shift to streaming, demand for good quality scripts remains a core element of this industry, rendering the screenplay a pivotal tool in deciding whether to green light a movie or not. This dissertation explores the application of Natural Language Processing (NLP) and Machine Learning (ML) techniques to analyse movie scripts, with the aim of extracting valuable insights and patterns that are able to predict the audience rating as collated by the Internet Movie Database (IMDb) [2].

This research investigates methods for concatenating features extracted from scripts to known information at evaluation stage and compile a vector that will be the basis for training an ML model. It attempts to address the problem faced by producers in deciding in which movies to invest their funds. By providing a sound method to sift through and rank the various script projects presented to them, they can focus on scripts that are likely to perform better.

Methods adopted in this research include the use of lexicons for the extraction of linguistic features, the analysis of emotional arcs in movies, embedding strategies for the script and statistical features generated from sentiment analysis. These features are concatenated to writer, producer, director and actor specific factors to train various regression models. Using a forward rolling window training strategy, the models are tested on previously unseen data achieving, using the best performing model, an R^2 of 0.5255, a MAPE of 0.1183, an RMSE of 0.7604, and a MAE of 0.5859 for regression metrics, and a one-away score of 0.8476, accuracy of 0.8375, and an F1-score of 0.7913 for the classification section.

Acknowledgements

This dissertation is the culmination of significant effort, and I am profoundly grateful for the support that made its completion possible.

My foremost thanks go to Professor John Abela for his insightful guidance, constructive criticism, and unwavering encouragement, which were instrumental throughout this research. I am also deeply indebted to the lecturers at the Faculty, who collectively fostered an environment of intellectual growth and rigor, essential to my studies and this work.

Beyond the academic sphere, I want to express my sincere gratitude to my family for their constant motivation and unwavering belief in my academic pursuits. My heartfelt thanks also to my partner for her immense patience, constant support, and understanding during this demanding period. Finally, I am truly appreciative of my close friends for their steadfast encouragement and support, even through the most challenging times.

To all who supported me, thank you for your invaluable contributions to this journey.

The research work disclosed in this dissertation is funded by the Tertiary Education Scholarships Scheme.

Contents

List of Figures

List of Tables

List of Abbreviations

AG Auto-Gaussian.

AI Artificial Intelligence.

ANN Artificial Neural Network.

API Application Programming Interface.

AUC Area Under Curve.

BOW Bag of Words.

CNN Convolutional Neural Network.

DCT Discrete Cosine Transformation.

DSPs Digital Video Subscription Platforms.

DTW Dynamic Time Warping.

GPU Graphics Processing Unit.

IDCT Inverse Discrete Cosine Transformation.

IMDb Internet Movie Database.

IMSDB Internet Movie Script Database.

IQR Inter Quartile Range.

LDA Latent Dirichlet Allocation.

LIWC Linguistic Inquiry and Word Count.

LLMs Large Language Models.

LOWESS Locally Weighted Scatterplot Smoothing.

LSA Latent Semantic Analysis.

LSTM Long Short-Term Memory.

MAE Mean Absolute Error.

MAPE Mean Absolute Percentage Error.

MPAA Motion Picture Association of America.

MTEB Massive Text Embedding Benchmark.

NER Named Entity Recognition.

NLP Natural Language Processing.

NLTK Natural Language Toolkit.

NRC National Research Council.

NRC-EIL NRC Emotion Intensity Lexicon.

OLS Ordinary Least Squares.

PDF Portable Document Format.

POS Part-of-Speech.

RFE Recursive Feature Elimination.

RMSE Root Mean Squared Error.

RNN Recurrent Neural Network.

ROI Return on Investment.

SOTA State of the Art.

SVR Support Vector Regression.

TF-IDF Term Frequency-Inverse Document Frequency.

VADER Valence Aware Dictionary and sEntiment Reasoner.

VIF Variance Inflation Factor.

1 Introduction

With thousands of established or aspiring screenwriters, the volume of movie scripts written each year compared with the amount of producers, production companies and available budgets, requires a proper and careful selection of scripts to assess which can be given a green light for production. While membership in associations like the Screenwriters Association [3] or the International Screenwriters Association [4] may provide scriptwriters with a direct link to industry professionals, producers still face the challenge of evaluating hundreds of scripts from multiple, and possibly unknown, sources. To avoid overlooking potential gems, they must carefully scrutinize these submissions, often relying on intuition or a 'comps'-based approach. Based on experience, movie producers typically predict a script's potential success by finding past movies with similar storylines or content and using how well those films performed financially as a guide to estimate the new movie's likely earnings [5]. Additionally, unknown writers, as with many other professions, have the added difficulty of trying to break through in a highly competitive and established industry. Despite inspiring stories like Sylvester Stallone's, who pitched *Rocky* after a failed screen test, unknown scriptwriters typically struggle to get their work reviewed by producers.

A frequently cited quote from a memoir entitled *Adventures in the Screen Trade*, originally published in 1983 sees William Goldman, a two time *Academy Award* winner for best screenplay, stating that in Hollywood 'nobody knows anything' [6]. This was later interpreted as, despite producers gaining knowledge from past results, predicting a new movie's commercial success remains almost impossible. [7].

1.1 Motivation

The aim of this dissertation is to build a machine learning model that can extract patterns and features from movie scripts, which, based on a supervised algorithm, can generate a prediction on the user rating for the produced movie.

The difficulty with assessing the success potential of a movie script at such an early stage lies in the limitation of information that can be used to make such a prediction. Traditionally, important elements such as the cast and crew, music composer, budget, release date, visuals and sound are available after the release of a movie, when most, if not all the budget allocated to it would have been spent.

With experienced producers, the process of reading through a script makes use of their experience in the sector, knowing what works and what may not work with audiences. Getting a script to the right producer is hence an important step in having a script professionally evaluated for its potential. Features associated with the success of

the producer are created and used in the building of this machine learning model.

Similar aspects for the writer and main actors are also included in the analysis. Experience in terms of written scripts, produced scripts, the relative performance of those produced scripts for writers and recent historical success of actors are elements that can be vectorized into features.

The genre or genres of the screenplay is a tool that can be used to assess the appeal to the right audience. Different genres carry inherent expectations from the audience. In a horror movie, the audience is expected to be scared, while in a romantic movie it is expected to be emotionally moved. Having a screenplay that achieves these expectations would generally achieve a good rating. While different genres would appeal to different audiences, some genres consistently attract higher audience scores. This is the case with action and adventure movies which normally offer exciting visuals and entertainment. Comedy movies are generally enjoyed for their lighthearted escape feeling while science fiction, fantasy or superhero related movies would typically incorporate special effects and stimulate imagination. It is relatively easy for a writer to categorise the script into one or more genres.

Some established directors are usually associated with one or more genre type of movies. This allows for a choice of directing style and while the availability or acceptance by a particular director may not be available at the time of an early review of a script, one can easily associate a particular director with one or more typical genres. As with producers, directors come with a history of experience, with the more established and successful directors, possibly attracting higher audience rating due to what is referred to in this dissertation as *director power*. The experience of that director in terms of movies directed as well as the ability to work well with a diverse set of crew and producers are hence also used in this dissertation as a feature in the building of the model.

Finally we have the screenplay itself. Original, well paced screenplays with an engaging plot are deemed to fare better, as those with relatable characters, a strong dialogue, emotional resonance and a satisfying resolution, which leaves a lasting impact on the audience. Table 1.1 shows how in 2024, movies with an original screenplay topped the charts in terms of gross revenue.

Rank	Source Material	2024 Gross ($ Millions)	Movies	Av. Gross ($ Millions)	Share (%)
1	Original Screenplay	4842.19	379	12.78	54.82
2	Based on Fiction Book/Short Story	949.78	88	10.79	10.75
3	Based on Comic/Graphic Novel	912.84	35	26.08	10.33
4	Based on Musical or Opera	579.42	7	82.77	6.56
5	Spin-Off	399.24	6	66.54	4.52
6	Based on Real Life Events	321.42	125	2.57	3.64
7	Based on Game	203.15	2	101.57	2.30
8	Based on Factual Book/Article	191.24	19	10.07	2.16
9	Based on TV	152.16	4	38.04	1.72
10	Based on Short Film	102.25	3	34.08	1.16

Table 1.1 2024 Box Office Gross by Source Material. Adapted from [8]

1.2 Aims and Objectives

1.2.1 Aims

This dissertation aims to examine features, extract patterns and other information pertaining to movies, and collate a dataset that is used to train a model to predict the audience rating. It is being taken into consideration that a number of human related factors influence the final product released to the audience. These factors cannot be measured at green-lighting stage and include the talent and compatibility of the cast (cast dynamics), leadership and team dynamics (crew compatibility), cultural and social relevance, any adaptations carried out to the script, market and distribution strategies, among others. In addition, audience related factors which include expectations, the emotional connection with the characters, any social influences such as word of mouth and social media, comparisons made with other movies and movie content such as special effects and the music score, would not be yet known or available at such an early stage.

1.2.2 Objectives

The achievement of this aim can be further categorised into several sub-objectives:

- Deliver an analysis of scholarly literature relating to the main elements being explored in this dissertation.

- Identify or build a suitable dataset of movie scripts or subtitles that can be used for the training of the model.
- Build a model that can predict the user rating improving on results that can be obtained using a basic average rating prediction.
- Test various models and optimise the results to obtain a generalised enough prediction.
- Evaluate such a system by comparing it to similar models and visualise results.

1.3 Proposed Solution

No universal or curated dataset could be located for this purpose and hence one needed to be collated from scratch. As the availability of full and updated screenplays was very limited, we augmented the script database with subtitles, which, while being stripped of important scene and mood setting descriptives as well as character analysis opportunities, provide adequate dialogue content that can be analysed across various dimensions.

The script based features were extracted using various techniques and tools. Linguistic features were extracted using LIWC-22 [9]. LIWC-22 is fundamentally a software program that employs a dictionary. This dictionary acts as a bridge, linking key psychosocial concepts and theories to specific words, phrases, and linguistic structures. It comes with over 100 built-in dictionaries created to capture people's social and psychological states [9] and can classify, generally in terms of percentage of words, a corpus in several dimensions which include:

- Linguistic Processes
- Psychological Processes
- Social Processes
- Emotional Tone
- Cognitive Processes
- Part-of-Speech

The emotional arc of the movie itself was analysed using a fully open-sourced lexicon, Valence Aware Dictionary and sEntiment Reasoner (VADER) which is a rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media [10]. A compound score is computed by summing the valence scores of

each word in the lexicon, adjusted according to the rules, and then normalised to be between -1 (most extreme negative) and +1 (most extreme positive). The results for each script are standardised, re-dimensioned and smoothed to become comparable. The resultant vectors are clustered to identify interesting patterns that match pre-defined story patterns. Statistical features that capture the form and shape of the arcs, were also extracted and used as features.

Another popular and well known source, the National Research Council (NRC) Sentiment and Emotion Lexicons, which is a collection of seven lexicons, was used to extract basic emotion indicators as well as intensity measures from the scripts. Two of these lexicons were used:

- The Word-Emotion Association (a.k.a. NRC Emotion Lexicon) is a list of English words and their associations with eight basic emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) and two sentiments (negative and positive) [11].
- The NRC Emotion Intensity Lexicon (NRC-EIL) is a list of English words with real-valued scores of intensity for eight basic emotions (anger, anticipation, disgust, fear, joy, sadness, surprise, and trust), where, for a given word w and emotion e, the scores range from 0 to 1 [12].

Finally, the whole script, following a pre-processing routine, has been embedded into 384, 768 or 1024 dimensional vectors using one or more state of the art embedding models. Various transformer models have been used for this purpose, specifically to assess the predicting factor between less and more advanced embedding models. For this purpose The Massive Text Embedding Benchmark (MTEB) [13] was used to choose the embedding models, depending on the infrastructure available to train the models. The MTEB can be viewed on a regularly maintained leaderboard [14]. The embedded vectors were concatenated to the other features to train the model.

Various regression models were tested for the experiments. Some of the tested models include:

- XGBoost [15]
- LightGBM [16]
- Lasso [17]
- Support Vector Regression [18]
- CatBoost [19]
- ElasticNet [17]

- MLPRegressor [17]

While XGBoost, LightGBM and CatBoost are gradient boosting machines models and rely on decision trees, Lasso and ElasticNet focus on linear relationships. MLPRegressor which is a neural network, introduces non-linearity and complex feature interactions. Support Vector Regression (SVR) is a powerful kernel-based method that can effectively model complex relationship.

The Mean Absolute Percentage Error (MAPE) together with R^2, Root Mean Squared Error (RMSE) and the Mean Absolute Error (MAE) were chosen as regression metrics to evaluate results. The combination of these metrics provides a more robust analysis of the results. With RMSE being more sensitive to outliers, large errors are amplified, while the MAE, being less sensitive to outliers, provides a more direct measure of the average prediction error. MAPE provides an analysis of the results in a percentage format, making it easier to understand. To enhance the interpretability of results a number of classification metrics, including one-away score, accuracy and F1-score were used.

1.4 Chapter Overview

The rest of this dissertation is structured as follows: in Chapter 2 we start by providing a background to the various notions and topics dealt with in this project, followed by a review of previous research conducted in the area of movie performance prediction. In this section works on measures of success in movie production, box office related analysis, the use of natural language processing and word embeddings with reference to the movie performance prediction are also reviewed.

Chapter 3 will focus on the methodology adopted for the experiments conducted and Chapter 4 elaborates on the implementation of the experiments. Chapter 5 will provide an outline of the tests conducted and the evaluation of the results. Finally we conclude in Chapter 6 with a summary of key findings, issues and limitations encountered and an outline for possible future work that could be performed in this area.

2 Background

2.1 The Movie Industry

With global box office revenues reaching $23 billion in 2024 and the average production budget peaking to $83 million in 2023, movies remain a significant sector within the world economy. But with movie studios releasing more movies than viewers can watch [20], 2024 saw the release of more than five thousand movies [8], an increase of almost 300% over the movies released in 2010. Arthur De Vany, an economist, estimated that 6% of films accounted for 80% of the industry's profits between 1993 and 2003 with 78% of movies losing money over the same period [21].

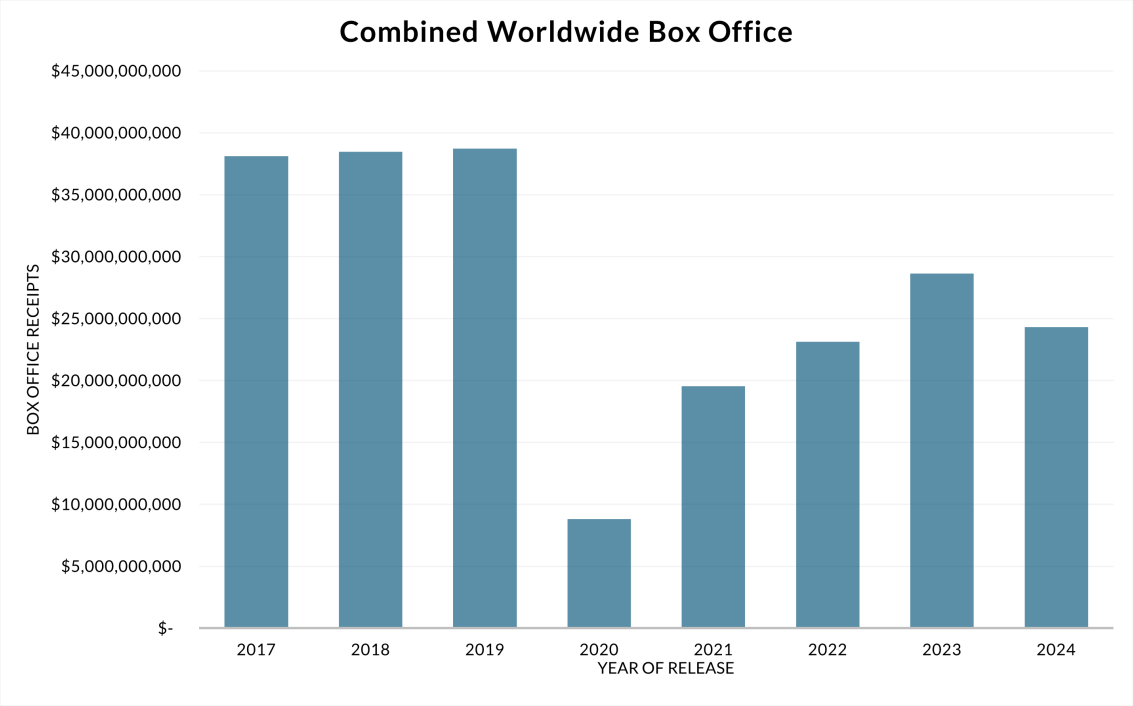


Figure 2.1 Global Box Office Receipts. *Adapted from [8]*

While success stories such as *Avatar* which grossed more than $1.8billion and costing $339million, and *Avengers: Endgame* grossing $1.4billion and costing $475million, do make headlines, flops like Disney's *Jungle Cruise* which lost more than $150million and *Moonfall* which lost more than $138million raise eyebrows, and with production budgets on an upward trend, the risk of bankrupcty, especially for smaller production companies, is not that remote. Both *Jungle Cruise* and *Moonfall* had a good recipe and basis, with A rated actors and crew, yet they failed to achieve a return on investment. Investments in promising movies that fail to deliver can be detrimental, yet turning down a prospect that is eventually taken up by another production company turning it into a success, is a massive opportunity lost.

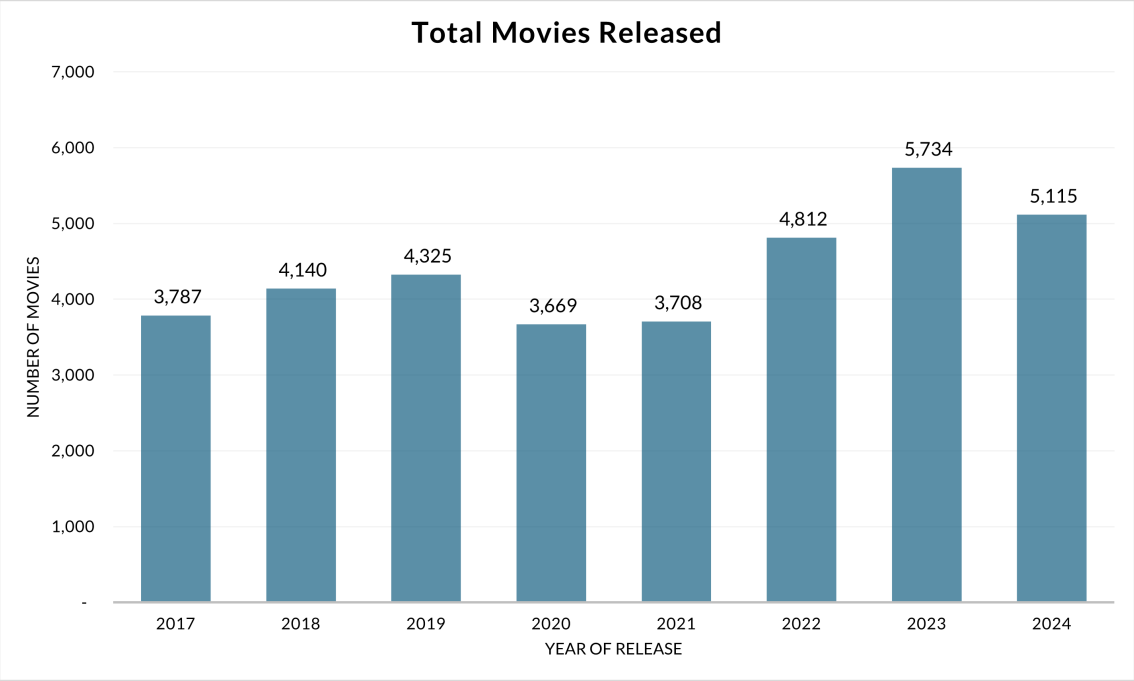


Figure 2.2 Total movies released. *Adapted from [8]*

Movies develop through a series of stages. A general idea, which forms the story concept, can take the form of an already published novel, an amusement park ride, as with the concept behind *Jungle Cruise*, a video game such as *Uncharted*, a real life event such as *Schindler's List* or even the life story of a famous person such as *Bohemian Rhapsody* [22]. This idea is then structured into a summary of a plot line which would indicate the type of movie, a draft sequence of events and an eventual outcome. This general sense of the movie would give an indication to the producer whether this could be a project that could go well with audiences [23].

Following a successful evaluation of these indications, a writer would normally be commissioned to draft a screenplay. It is also possible that scripts from writers are provided to producers for evaluation by agents or writers' associations. This is a crucial stage in the movie development process, given that the serious investment would start after green lighting the script. Many of the factors known to influence the outcome of a movie are unknown at this stage. Issues such as the number of movies that would be released at the same time, reactions of critics and audiences, award nominations and even remote situations such as global pandemics are unknown. Yet the producer needs to be able to take a decision. It would be hence helpful for studios, producers and investors to have some form of intelligence that could assist their decision making process, prior to getting to the production stage, where most of the financial investment is spent [22] ; [24] ; [25].

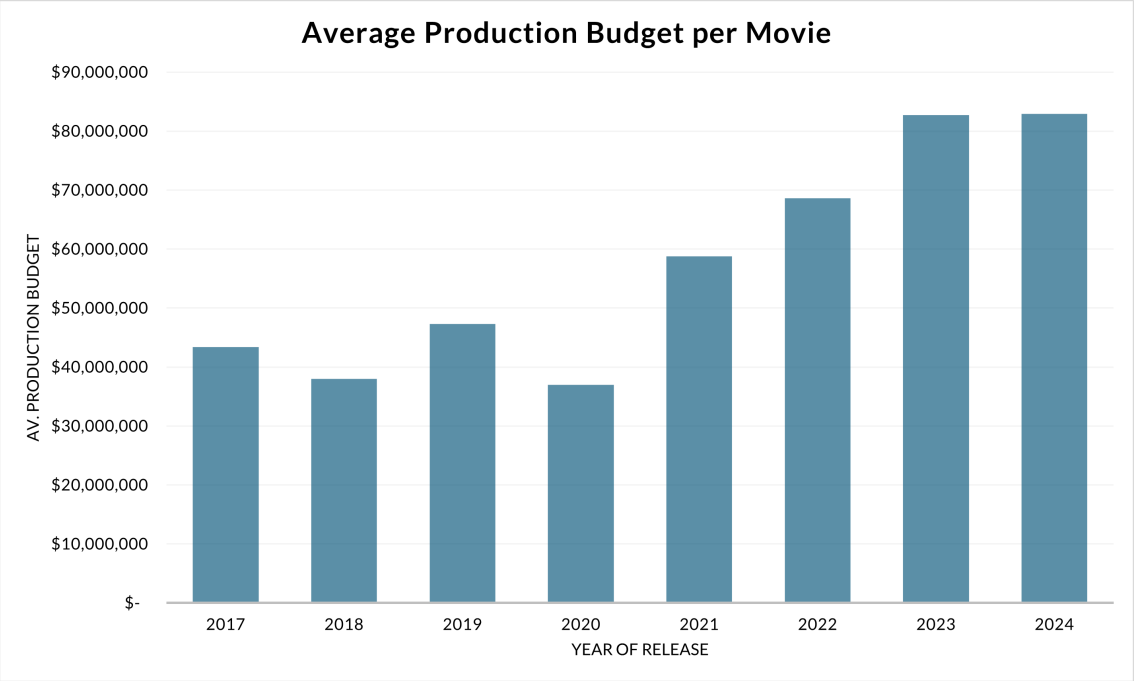


Figure 2.3 Average Production Budget. *Adapted from [8]*

2.2 Natural Language Processing

The study and processing of natural language with reference to computer technology has been evolving since the early 1950s [26]. A succinct definition of Natural Language Processing (NLP) is provided by IBM [27],

> 'Natural language processing (NLP) is a subfield of computer science and artificial intelligence (AI) that uses machine learning to enable computers to understand and communicate with human language.
>
> NLP enables computers and digital devices to recognise, understand and generate text and speech by combining computational linguistics—the rule-based modeling of human language—together with statistical modeling, machine learning and deep learning.'

2.2.1 The importance of NLP

It is estimated that the global big data analytics market was valued at USD 348 billion in 2024 and is projected to grow to USD 924 billion by 2032 [28]. Data is generally classified as *structured* or *unstructured*, depending on its format and the presence or absence of schema rules. Structured data follows a well-defined format, making it easily interpretable by data analytics tools, machine learning models, and human users. It is commonly stored in tabular formats like spreadsheets and relational databases. In

contrast, unstructured data lacks a predefined format and accounts for approximately 90% of all enterprise-generated data. It can include both textual and non-textual elements, incorporating both qualitative and quantitative information. [29].

Figure 2.4 NLP within the AI eco-system. *Adapted from [30]*

NLP forms part of the Artificial Intelligence (AI) eco-system, which includes machine learning and deep learning, and fills a gap that exists between human communication and computer understanding. Techniques developed for such a purpose analyse large bodies of typically *unstructured* text data that may include documents, log files, transcripts, reviews, social media posts, financial reports, books and research papers, among others [30]. NLP techniques combined with machine learning have a very important role within the analysis, interpretation and use of text based unstructured data.

The output of such techniques and applications, as shown in Figure 2.5, are equally varied. In this dissertation *sentiment analysis*, *topic extraction*, *knowledge graphs* and *text similarity* will be the main outputs that will be generated by the adopted NLP techniques. These are combined with traditional machine learning and other statistical techniques to build the predictive model.

2.2.2 Sentiment Analysis

Two central areas of NLP are sentiment analysis and emotion recognition. While they might seem to be used interchangeably, they differ in a few respects [31]. Sentiment analysis, sometimes referred to as opinion mining, uses natural language processing to extract and analyse emotions, attitudes, and opinions expressed in text [32]. It is a classifying exercise to assess if data is positive, negative or neutral.

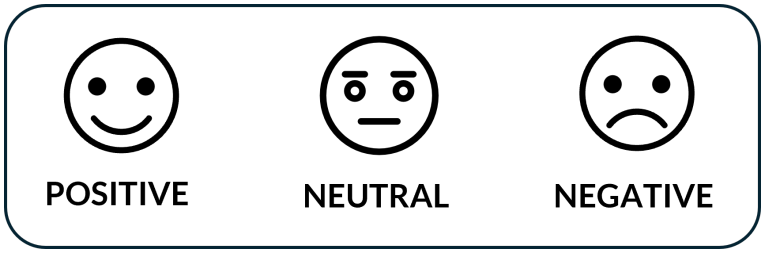


Figure 2.6 Classification output of Sentiment Analysis. *Image created by author*

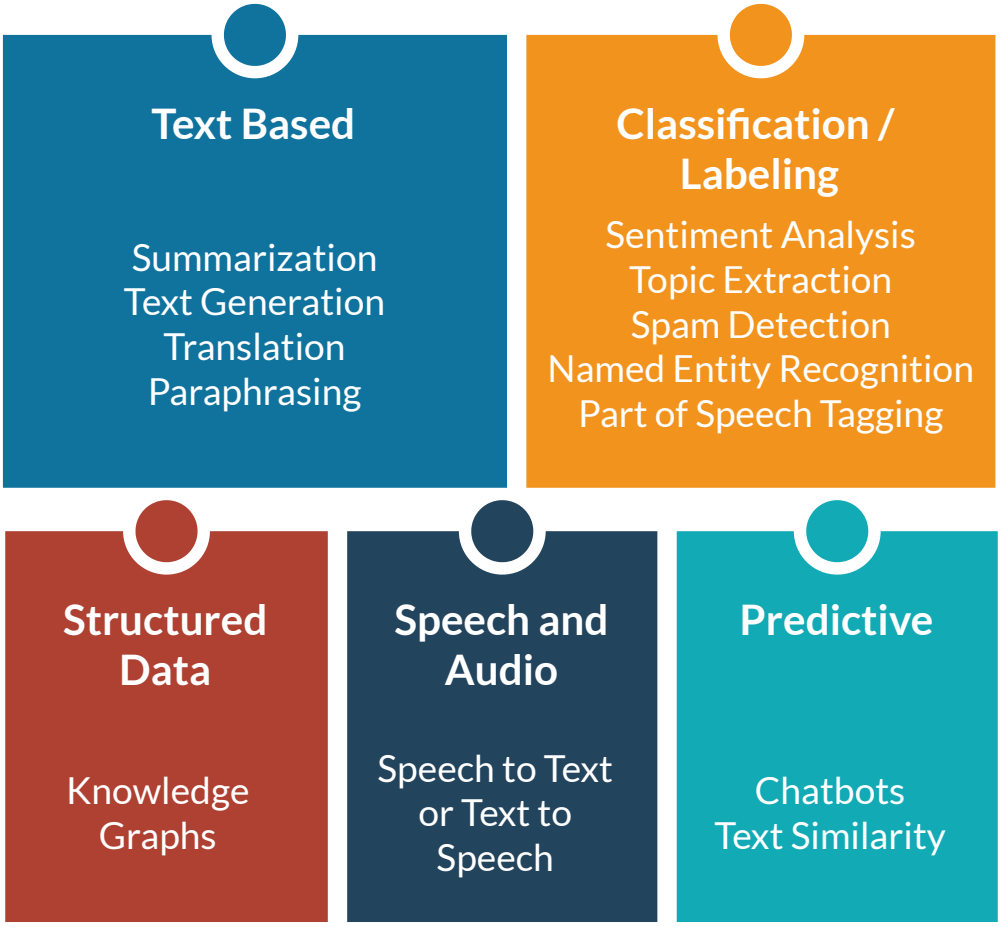


Figure 2.5 Types of NLP techniques generated output. *Image created by author*

Emotion recognition, or detection, is a means to identify distinct human emotion types such as sadness, joy, anger and fear. With an estimated global number of social media users calculated at 5.24 billion in 2024 [33] and the proliferation of platforms such as Facebook, Instagram, TikTok and X, people have at their fingertips an easy vehicle through which they can express opinions, write reviews and in general express their feelings and emotions towards anything and everyone. A 2024 study [34] reveals a massive amount of unstructured text generated per minute: 229 million meeting minutes on Microsoft Teams, 251 million emails, and 1.04 million Slack messages. This substantial volume necessitates analysis for sentiment and emotion detection.

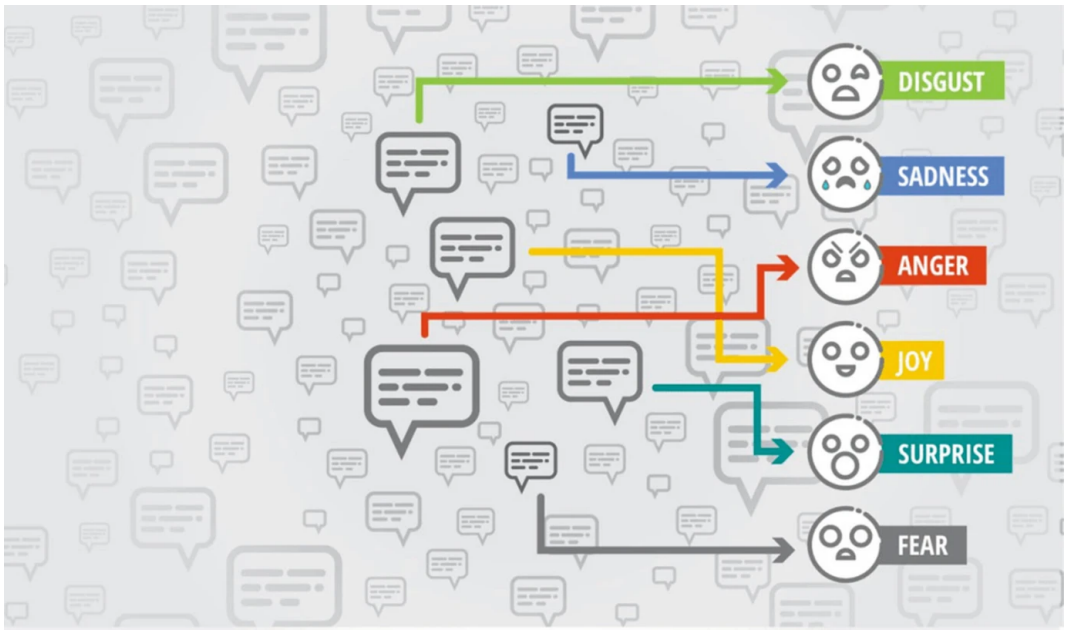


Figure 2.7 Classification output of Emotion recognition. *Source: [35]*

Sentiment and emotion analysis, applicable across diverse domains, employs varied methodologies, primarily categorised as lexicon-based, machine learning-based, and deep learning-based approaches, each with inherent advantages and limitations. Researchers encounter significant challenges in sentiment and emotion recognition, including contextual dependency, handling sarcasm, resolving polysemy, interpreting evolving web slang, and addressing lexical and syntactic ambiguity [31].

In a lexicon-based methodology a word dictionary is maintained, in which each positive and negative word is assigned a value. The sentiment value can then be calculated at a sentence or document level. Popular lexicons include SentiWordNet [36] and Valence Aware Dictionary and Sentiment Reasoner (VADER) [10] for sentiment analysis and NRC word-emotion lexicon [11] for emotion extraction. For machine learning based approaches, supervised classification models would require curated training and testing sets and use traditional algorithms such as Naïve Bayes and Support Vector Machine (SVM). In deep-learning based approaches, no feature engineering is required as these algorithms are able to detect sentiments or opinions through their architecture.

2.3 Word Embedding

In the field of NLP, word embedding is a central term which refers to the technique of representing words for text analysis in the form of real-valued vectors. Using this approach, algorithms are used to allow similar words to have similar vector representations, as shown in Figure 2.8, thus retaining semantic as well as syntactic

information [37]. Each word becomes represented by a real-valued vector which may have multiple dimensions.

Once words can be converted into machine readable data, traditional machine learning tools can be applied across a wide range of applications, from simple classification of spam email to more complex regression analysis such as document readbility scoring. Sentiment analysis, machine translation and named entity recognition are key tasks making use of word embeddings.

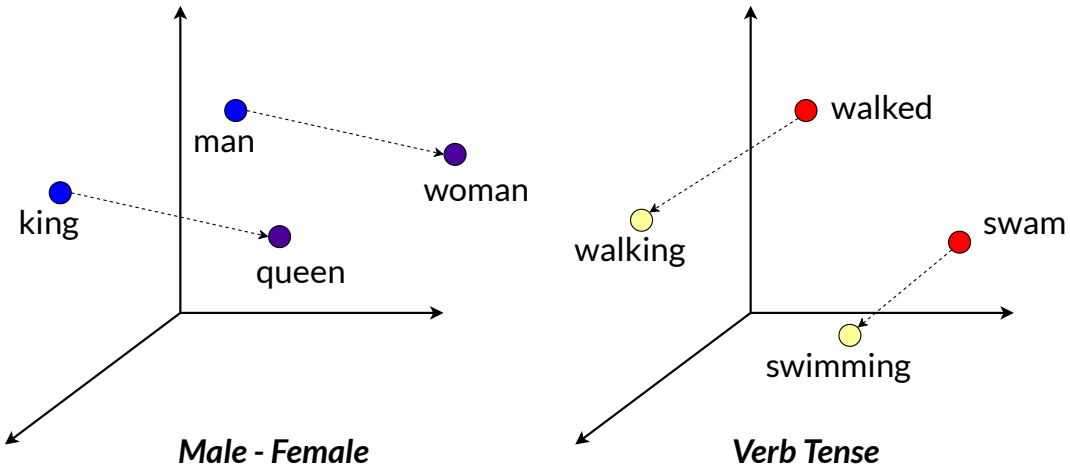


Figure 2.8 Representation of similarity in vector embeddings. Embeddings generated by Word2Vec. Image source: [38]

Early algorithms, such as Term Frequency-Inverse Document Frequency (TF-IDF) worked on a statistical measure of finding word relevance across a single or multiple documents. While Term Frequency (TF) measures the frequency of words in a particular document, the Inverse Document Frequency (IDF) measures the rarity of the words in the text. The formula for TF-IDF is given by:

$$\textbf{tf-idf}_{i,j} = \text{Term Frequency}_{i,j} \times \text{Inverse Document Frequency}_i$$

Where,

$$\textbf{Term Frequency}_{i,j} = \frac{\text{Term } i \text{ frequency in document } j}{\text{Total no. of terms in document } j} \tag{2.1}$$

$$\textbf{Inverse Document Frequency}_i = \log\left(\frac{\text{Total documents}}{\text{No. of documents containing term } i}\right)$$

The algorithm can be applied to a sequence of string items using *Scikit-Learn's TfidfVectorizer* module [39] which returns a matrix of TF-IDF features. One drawback of the TF-IDF algorithm is that it is not able to adequately capture the semantic meaning of words in a sequence [37].

The Bag of Words (BOW) technique is an equivalently simple algorithm to

understand, wherein each value in the representative vector would represent the count of words in a particular document or sentence [37]. This technique can also practically applied using *Scikit-Learn's CountVectorizer* [40].

Developed by Google in 2013, Word2Vec was an early technique which applied shallow neural networks to learn word relationships from text. It worked by assuming that words appearing close together tend to have similar meanings. Using cosine similarity, it projects to the final layer embedding vectors which are geometrically close, where, the underlying mapped words are also semantically similar [37].

While these earlier and simpler models have provided important advances in the processing of natural language, they are subject to certain limitations. While the older techniques of BOW and TF-IDF are still useful for some tasks where computational efficiency might be an issue, they lack the ability to capture semantic relationships and word order as they are mostly concerned with word frequencies. They also suffer from out of vocabulary issues, meaning that if a new word appears in a new document that was not present in the original vocabulary, it would just be ignored. While being more advanced, Word2Vec still suffers from a limitation on the context window given that it can only capture relationships between words that are within that window. It also produces fixed embeddings for words, meaning that in whichever context it appears, a word will always be represented by the same vector embedding.

With the introduction of GloVe [41] the concept of a co-occurrence matrix enabled the algorithm to capture long-range word relationships, extending the range from a pre-defined window. While it captures more global patterns and is more efficient in training due to its matrix factorization approach, it still produces the same representation for a word irrespective of the context.

2.3.1 Static vs Contextualized Embeddings

The transition from static word embeddings to contextualized embeddings is a pivotal advancement in NLP as it redefined language representation, allowing models to dynamically adjust word meanings based on contextual dependencies. By capturing semantic and syntactic variations, these contextualized embeddings have significantly enhanced the comprehension of human language by machines and form the foundations of today's modern architectures for Large Language Models (LLMs) [42].

The difference between static and contextualized embeddings can be visualized in Figure 2.9 where the word *bank* is being used in different contexts. While using static embedding models such as Word2Vec and GloVe the vector representation of the word would remain the same, using contextualized embedding models results in a different vector representation depending on the context in which that word is used.

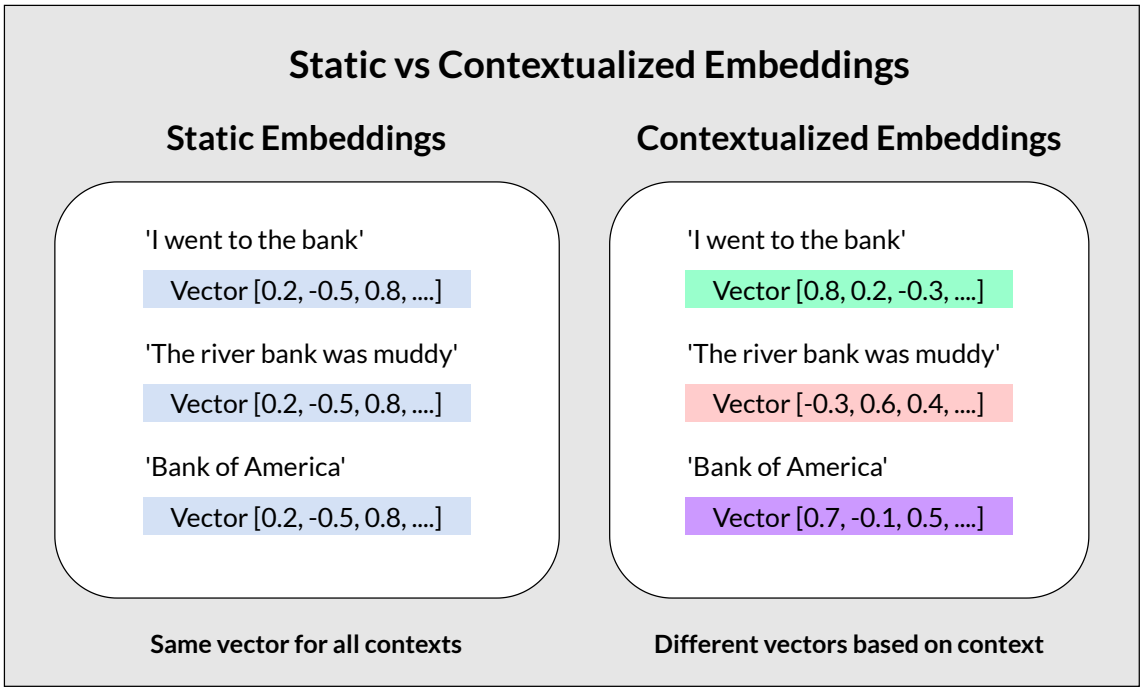


Figure 2.9 Examining the vector representation for the word 'bank'. Image source: [42]

The major breakthrough came with the introduction of BERT in 2019 [43] with a fully bidirectional contextualization and transformer based architecture. BERT was the first model to make use of the self-attention mechanism introduced by Vaswani et. al [44] with input embeddings being the sum of token embeddings, segmentation embeddings and position embeddings as shown in Figure 2.10.

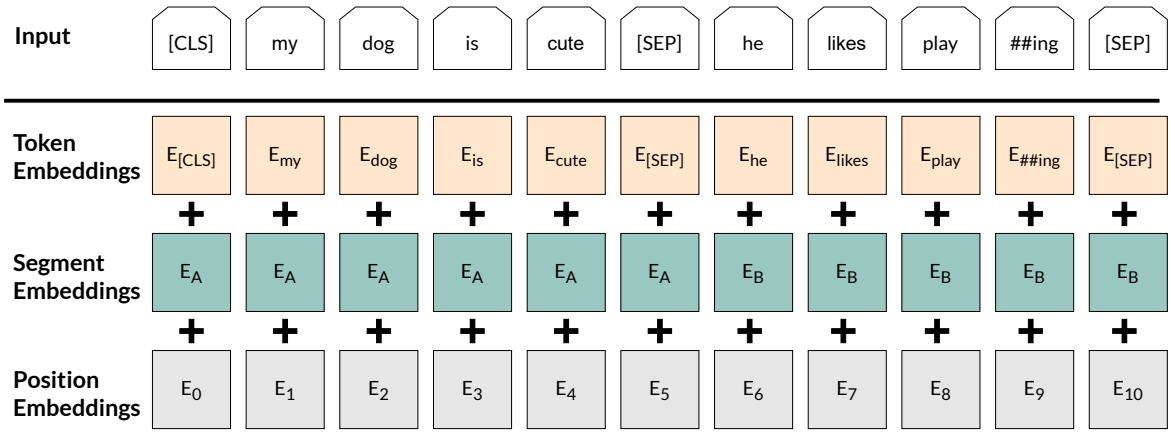


Figure 2.10 BERT input representation. Image source: [43]

Contextualized embedding enabled words to receive different vectors depending on their context. This resulted in the better handling of polysemy and homonyms. The multiple layers of the architecture allowed for the capture of syntax and basic patterns at the lower layers up to processing semantic relationships and task-specific features at the higher layers. Such fundamental improvements helped

LLMs to have an enhanced understanding, allowing for transfer learning and scalability by having better generalization to unseen contexts [42].

Various embedding models, with different levels of sophistication and licences, have been developed since the launch of BERT. In 2022 a paper launched a set of 8 embedding tasks covering a total of 58 dataset and 112 languages for the purposes of setting a benchmark against which to evaluate embedding models [13]. A dedicated webpage is regularly updated with the most recent released models, together with their respective ranking [14].

With the launch of larger models over the years, embedding models have grown from a few million parameters to 7 billion parameters, increasing embedding dimensions from 384 to 8960 and allowing for maximum input tokens to increase from the standard 512 tokens to 131,072 for the most recent models. Assuming most models use 32 bit floating-point numbers (4 bytes) per parameter, a rough calculation would indicate that while a model having 33 million parameters fits into around 125Mb of disk space, one with 1.5billion parameters needs 5.7GB while the top end of the models having 7billion parameters need 26GB of space. To speed up processing a Graphics Processing Unit (GPU) is recommended, and consumer sized GPUs have a memory ranging between 4GB to 24GB. Embedding models that excel on the MTEB Leaderboard [14] usually contain a large number of parameters and high vector dimensions. For example, *NV-Embed-v2* [45]; [46] and *bge-en-icl* [47];[48] have 7 billion parameters and 4096 dimensional vector representations, leading to slow inference speeds and high storage costs, posing a significant challenge to their direct practical application by the average consumer [49].

2.3.2 Long-context embedding

Current embedding models allow for maximum input tokens to be between 512 and 131,072. In an analysis of 12,000 movie scripts [50], the median length of all scripts was 106 pages. Taking an average of 200 words per page, this translates to circa 21,200 words per script. Using an online OpenAI Token Calculator [51] this converts to circa 28,267 tokens. Loading the weights of a 1.5billion parameter model requires circa 5.7GB in memory for a 32 floating-point set up or 3GB for a 16 floating-point setup. As 30,000 tokens require circa 15GB of GPU memory, given the restricted infrastructure available for this dissertation, we limited our analysis of embedding models to those having up to 1.5billion parameters.

Self-attention mechanisims, which form an integral part of the architecture of embedding models, require an attention matrix of size $N \times N$ as each token attends to every other token, The computational complexity thus becomes $O(N^2 \cdot d)$ where d is the embedding dimension. This causes runtime and memory requirement to grow quadratically, making the embedding of longer scripts at one go, unpractical for

consumer based infrastructures. Many applications require retrieving smaller text segments, and dense vector-based retrieval systems tend to perform better with shorter inputs, as they preserve semantic details more effectively. To prevent information loss from overly compressed embeddings, practitioners often break longer documents into smaller chunks and encode them separately [52]. This process, termed *'chunking'*, will be used for the purposes of embedding the available screenplays.

2.4 Keyword Extraction

Keywords play an important role in text processing. Keyword extraction, or as is sometimes referred to as *keyphrase* or *key term extraction*, is a process which aims to find the most representative words or phrases within a text [53] [54]. It provides a base for various text mining tasks, including text summarization [55], document classification [56], document retrieval [57] and topic modeling [58].

Over time, keyword extraction has advanced, adopting diverse approaches including *statistical analysis*, *rule-based systems*, *machine learning* and *deep learning techniques*. *Statistical methods* used in keyword extraction identify keywords based on their statistical properties within the text with TF-IDF being a well known approach where a higher TF-IDF score would indicate more significance. In *rule-based methods*, linguistic rules, patterns and heuristics are used to identify keywords. Popular techniques include Part-of-Speech (POS) Tagging which assigns grammatical tags to words in sentence and considering specific POS patterns to extract keywords. With *machine learning* based methods, algorithms and models are trained on labeled data to identify keywords. While supervised learning techniques classifiers such as SVM, Random Forests or Recurrent Neural Network (RNN)s work on labeled data to classify words or phrases as keywords, unsupervised techniques involve clustering, topic modeling or word embeddings to extract keywords based on their co-occurrence, semantic similarity or topic associations [59].

With the introduction of the transformer based architecture, models have been able to learn complex contextual representations of words and phrases, enabling more accurate keyword identification. Embeddings generated by BERT [43] have been used to create a keyword extraction technique that leverages these embeddings to create keywords and keyphrases that are most similar to a document. KeyBERT [60], being a minimum and easy to use technique, was developed using such embeddings and is used in this dissertation to extract keywords from movie plots.

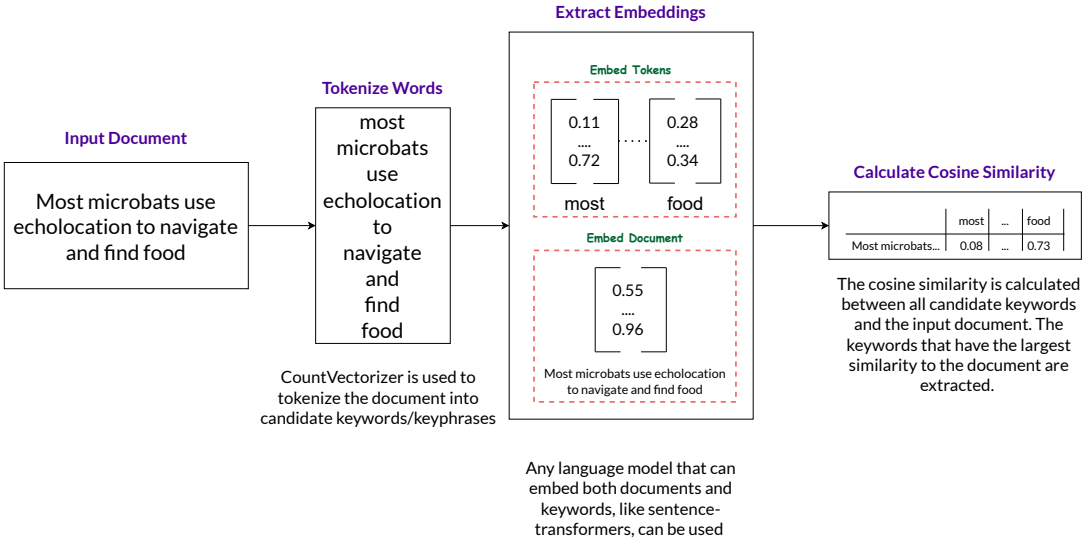


Figure 2.11 keyBERT keyword and keyphrase extraction process. *Image source [60]*

2.5 Knowledge Graphs

Modern applications rely on uncovering and analysing relationships. From personalized recommendations and fraud detection to constructing complex knowledge graphs, understanding how data points connect is crucial. Traditional databases can struggle with the intricate nature of these connections. Graph databases, on the other hand, excel by providing a more intuitive and efficient way to model and explore these relationships [61].

Knowledge graphs are defined as

> 'a graph of data intended to accumulate and convey knowledge of the real world, whose nodes represent entities of interest and whose edges represent potentially different relations between these entities.' [62].

Such relationships can be analysed using graph theory. The term graphs and networks are sometimes used interchangeably as they convey the same type of information. A graph $\mathcal{G}$ is defined as an ordered pair $\langle \mathcal{V}, \mathcal{E} \rangle$, where $\mathcal{V}$ is a finite nonempty set of vertices or nodes and $\mathcal{E}$ is the set of edges or links between the vertices $\mathcal{E} \subseteq \{(u, v) \mid u, v \in \mathcal{V}\}$ [63]. Visuals such as in Figure 2.12 can be used to link cast and crew on the basis of movies on which they collaborated.

An important concept within the study of knowledge graphs is *centrality*, which deals with distinguishing important nodes in a graph. It involves the analysis of every node as they are connected to the other nodes in the graph. As different perspectives of importance can define what makes a node more important than another, various metrics can be employed to study each node from each such perspective. These

different perspectives are analysed using various indices, which are collectively known as *centrality measures* [64].

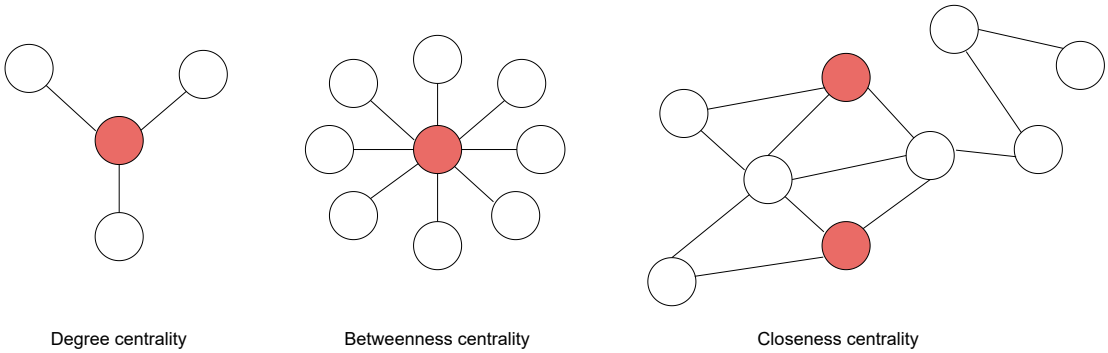


Figure 2.12 A sample of centrality measures. *Image source [65]*

In Figure 2.12 the red nodes are the most central ones in the network, indicating they are highly important based on different measures. *Degree centrality* is a simple count of a node's direct neighbors (the red node has 3). *Betweenness centrality* shows how often a node acts as a bridge on the shortest paths between other nodes. *Closeness centrality* indicates how easily or quickly a node can access all other nodes in the network [65].

2.6 Machine Learning

IBM define machine learning as

> 'a branch of AI focused on enabling computers and machines to imitate the way that humans learn, to perform tasks autonomously, and to improve their performance and accuracy through experience and exposure to more data.' [66].

Literature presents a number of different typologies of machine learning. As described in Figure 2.13 the most common categories are *Supervised* and *Unsupervised* learning. There are other two less coined but equally important categories, being *Semi-Supervised* and *Reinforcement* learning.

In this dissertation, supervised and unsupervised techniques will be used. In a supervised learning environment, the outcome of the model can be either categorical or continuous. For the former output, a set of classification algorithms are available, while for the latter, regression models are used. As the primary objective of the model being trained in this dissertation is to predict a numeric output, a focus will be placed on regression models.

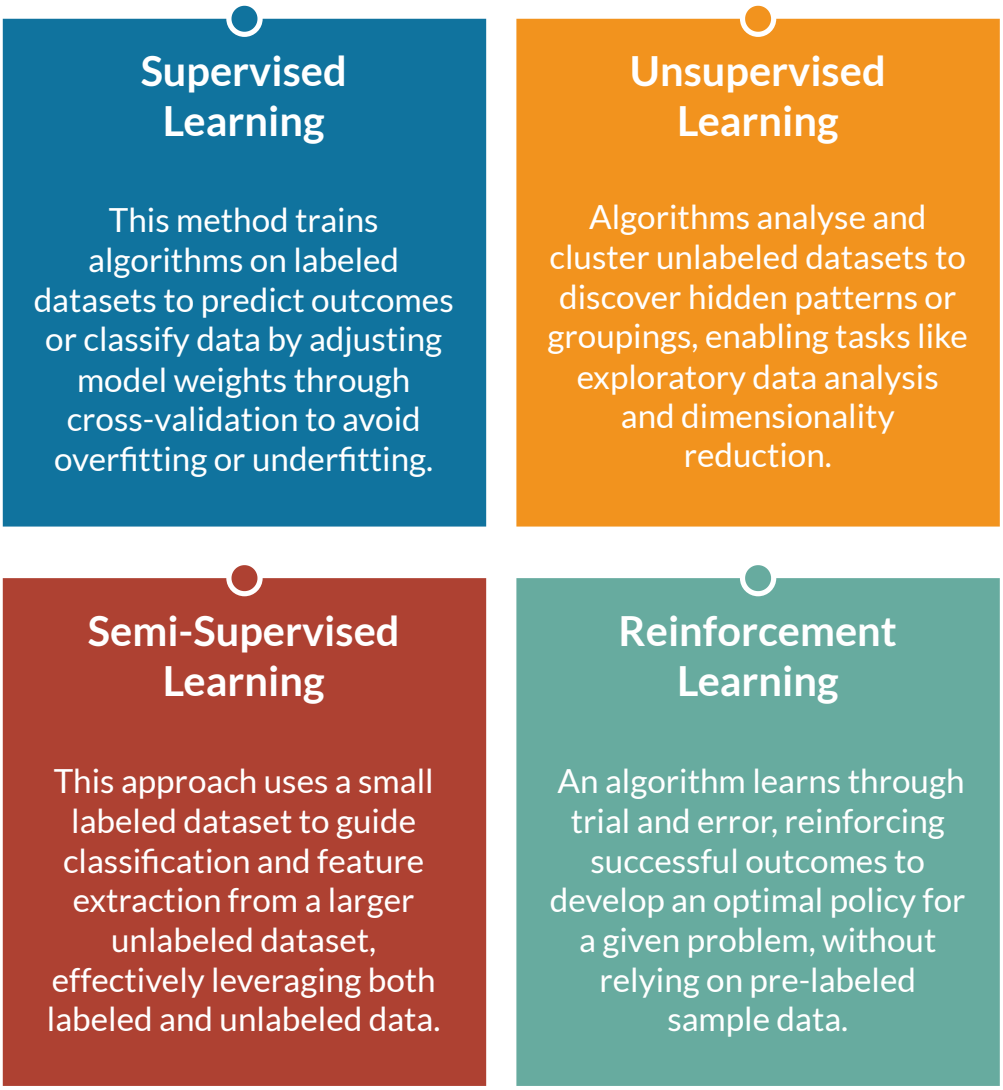


Figure 2.13 Categories of Machine Learning. *Adapted from [66]*

Originally used in statistics, regression is a method for predicting a numerical outcome based on other pieces of information. It learns a general rule from data by examining the relationship between the influencing factors (input or independent variables) and the result that is to be predicted (the output or dependent variable). While classification is used to predict discrete labels or categories, regression is used specifically for predicting continuous, numeric values [67]. In its mathematical form, regression can be denoted as:

$$Y = f(X) + \alpha,$$
where $X = \{x_1, x_2, \ldots, x_m\}$ represents the m input variables,
and Y refers to the output variable(s),
while α is the error term that captures all kinds of errors. (2.2)

2.7 Literature Review

2.7.1 Measure of Success

Several predictive models have been researched for predicting movies' financial results, total number of tickets sold or the general abstract concept of success [68]. In 2009, three main criteria of cinematic success were identified. Termed as *'The Success Triad'* this research concluded that at the time, with few exceptions, almost every empirical study made in this area used one or more of three main success criteria: *critical evaluations*, *financial performance* and *movie awards* [69]. Since global box office revenue remains lower than pre-2020 figures, ticket sales alone may not be the most relevant measure of a movie's success today. The growth of Digital Video Subscription Platforms (DSPs), such as Netflix and Disney+, has introduced new ways to evaluate films. Newer models prioritise the value of the content itself, now including factors like subscription revenue, customer growth and retention, and income from cross-sales and advertising [70]. Moving from simply totaling box office money to using a complex calculation that combines all these new data points to determine a film's post-release performance requires significant data work, let alone predicting success during the initial planning stage.

Another challenge in figuring out how well movies perform is the lack of official data on how many people actually watch them. While *Netflix* began releasing some selected viewing numbers starting June 28, 2021 [71], very few, if any, of the many other streaming services worldwide provide similar statistics. Although websites like *FlixPatrol* collect data from over 885 platforms worldwide [72] and offer daily streaming and on-demand popularity charts, these rankings are not based on official viewership. Instead, they use their own methods that track online activity and social media trends, such as fan counts and mentions on platforms like Facebook, Twitter (now *X*), and Instagram, as well as Wikipedia views and file-sharing data. For the scope of this dissertation, a movie's performance will be quantified using user ratings sourced from the Internet Movie Database (IMDb). It is crucial to acknowledge, however, that while IMDb ratings offer valuable insight into audience reception, they should not be

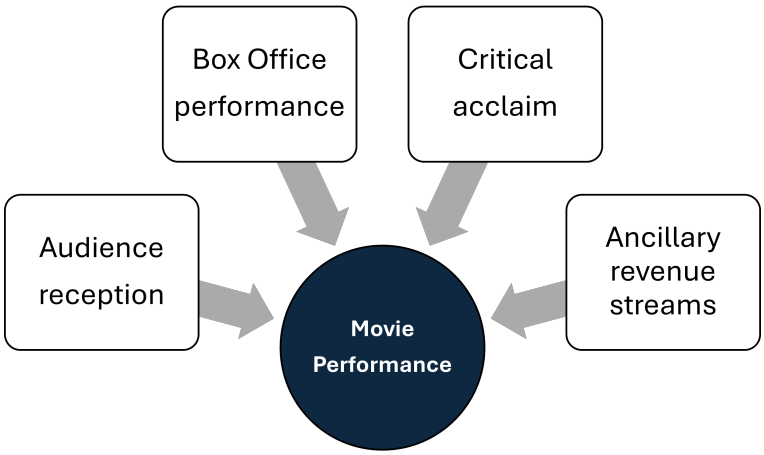


Figure 2.14 Movie performance indicators. *Adapted from [22]*

considered a definitive and exhaustive measure of a film's success. These ratings do not directly reflect critical financial metrics such as box office earnings or return on investment, nor do they fully encapsulate the broader cultural or societal influence a movie may achieve.

Early researchers in 1994 focused on the explanatory influence on movie performance and find that quality and marketing expenditures are important determinants of a motion picture's financial success and that production cost and the presence of star performers are only important when marketing is not included [73]. Additional research was also conducted in the effect of star crew (actors and the director) indicating that replacing an average star with a top star would increase revenues by an average of $16.6million [74].

In a study of movies released between 1977 and 2006, film performance was indicated to be shaped by a wide array of explanatory factors. Among *production factors*, leading actors' star power and the production budget emerged as particularly influential, with star power serving as a strong indicator of audience draw and budget signaling the scope and quality of a film's execution. In terms of *distribution factors*, the prominence of previews, screenings, and trailers significantly impacted audience anticipation and engagement pre-release. *Exhibition factors* such as screen coverage in the first week were also critical, as wide availability often correlated with stronger box office returns. On the organisational level, the track record of the head of production proved essential, reflecting the strategic and creative competence driving the film. Additionally, individual spectator traits, particularly psychological and socio-cultural elements, shaped how different audiences connected with the film's content. Lastly, *third-party information sources*—notably word of mouth from non-expert viewers and professional critics' reviews—played a central role in shaping public perception and influencing audience turnout [75].

2.7.2 Box-office Receipts

Regression analysis has been used since 1983 to predict movie success [76]. Box-office receipts are frequently used as a measure of performance as Return on Investment (ROI) is a main key performance indicator for investors. Tables 2.1 and 2.2 show the best five and worst five movies respectively, in terms of profitability. Expected revenue from a movie was analysed in a research paper which classified box-office receipts into one of nine categories, ranging from 'flop' to a 'blockbuster'. Neural networks were used on features normally available upon release to achieve an average correct classification rate of 36.9% across the nine categories [77]. Movie revenues were also researched by using features about *'who'* is in the cast, *'what'* is the movie about, *'when'* a movie would be released as well as *'hybrid'* features such as annual profitability percentage by genres, average genre expertise of cast, keywords and competition in terms of other movies that would be released at the same time. Using a Lasso Regressor the researchers achieved a RMSE of 0.878 [78].

Year	**Movie**	**Income** (in millions)	**Expense** (in millions)	**Profit** (in millions)
2009	Avatar	1895.19	339.78	1555.40
2019	Avengers: Endgame	1482.01	475.56	1006.45
2025	Ne Zha 2	989.55	84.12	905.43
2013	Frozen	1052.69	251.33	801.36
2018	Avengers: Infinity War	1128.30	368.81	759.48

Table 2.1 Most Profitable Movies, Based on Absolute Profit on Worldwide Gross. Adapted from [8]

In a study of 170 movies from 2010 and 2011, researchers aimed to predict how much money a movie would earn on its first weekend. They analysed the movie scripts using a special technique that looks at the historical origins of words and how they are combined. This created a network representation of the language in each script, grouping words based on their shared roots and connections. A computer program called *Automap*, developed by the CASOS Institute [79], was used for this analysis. They then measured the size of the most connected part of these language networks in the screenplays.

Using the Ordinary Least Squares (OLS) statistical approach, they built a model to predict opening weekend earnings. They also accounted for other factors that could influence box office, such as if the movie was a drama, its rating, if it was an original story, the success of the screenwriter's previous film, and the year the movie was released. The main factor they tested as a predictor was a measure of the variety of word origins in the central part of the screenplay's language network, referred to as Logsize. Their most successful model achieved an R^2 score of 0.46 [80].

The decision on whether a movie is successful or not was also based on *box office receipts* in a study that compares various machine learning approaches for movie

Year	Movie	Income (in millions)	Expense (in millions)	Loss (in millions)
2022	Strange World	39.60	217.65	-178.04
2022	Turning Red	8.26	181.24	-172.98
2021	Jungle Cruise	114.08	264.22	-150.14
2011	Mars Needs Moms	26.79	170.17	-143.38
2020	Mulan	59.37	200.00	-140.63

Table 2.2 Biggest Money Losers, Based on Absolute Loss on Worldwide Earnings. Adapted from [8]

success prediction. A Kaggle based dataset [81] with 4000 movies represented by 11 attributes was used for a classification exercise in which actor name, director name, genre, IMDb rating, budget, estimated profit and profit percentage were used as input features to determine whether a movie will be a hit or a flop. Accuracy of 85.4% was achieved using Gaussian Naive Bayes [82]. Besides being based on features that are only known at a very advanced stage of the production, using features such as profit percentage and IMDb ratings introduced an element of data leakage which unfortunately does not add to the predictability power of the proposed model.

2.7.3 User Rating

Other studies focused on the *user rating* of movies. Table 2.3 highlights the top 10 highest-rated movies on IMDb, demonstrating the significance of user ratings in evaluating movie performance and underscoring the role of machine learning techniques in predicting audience preferences. A study of *Twitter* and *YouTube* textual features (comments) on a dataset of 70 movies achieved an RMSE of 0.4459 [83]. In another study published in 2014, the authors used linear combination, multiple linear regression and neural networks on features extracted from IMDb, including genres, directors, actors, writers, country, film locations and runtime, to achieve what they refer to as an average Prediction Absolute Error, which essentially is the Mean Absolute Error, of 0.6973, with 76.8% of the 1462 movies in the test set being predicted within 1 unit of rating [84]. An interesting aspect of this study was the weight that the actors had in their multiple linear regression model. One might argue that at a very early stage of analysis, the actors would not yet have been locked in and hence this feature might not be a realistic feature to use in such a model.

Later in 2020, a study on *user rating* prediction used convolutional neural networks with three dense layers based on historical values, metadata features, categorical features, topical and social features such as director, actor, genres and production companies' historical average ratings, *Facebook* likes or gross receipts (as may be applicable) together with budgets, duration of the movie, release dates, plot keywords, tagline, content rating and storyline summary, to achieve a Mean Absolute Error of 0.83 by training on a dataset containing 3819 movies released from 2000 until

2013 and testing on a dataset containing 498 movies released after 2013 [85]. This approach yet again made use of features which are likely not to be available at green lighting stage.

In 2022 a transformer encoder, RoBERTa [43], was used to encode inputs such as budget, date of release, storyline and description from IMDb, into an embedded vector which was fed into a neural network for predicting user ratings. The authors used a basic accuracy metric to determine whether a movie is a hit or a flop, with movies scoring 4 or less being considered a flop and movies scoring 7 or more being considered a hit, hence converting the output of the model to a binary classification exercise. The model achieved 85% accuracy [86].

Rank	Movie	Year	Rating	Votes
1	The Shawshank Redemption	1994	9.3	3M
2	The Godfather	1972	9.2	2.1M
3	The Dark Knight	2008	9.0	3M
4	The Godfather Part II	1974	9.0	1.4M
5	12 Angry Men	1957	9.0	917K
6	The Lord of the Rings: The Return of the King	2003	9.0	2.1M
7	Schindler's List	1993	9.0	1.5M
8	Pulp Fiction	1994	8.9	2.3M
9	The Lord of the Rings: The Fellowship of the Ring	2001	8.9	2.1M
10	The Good, the Bad and the Ugly	1966	8.8	847K

Table 2.3 Top 10 Movies by Rating with Year of Release and Number of Votes. Source : [87]

The IMDb *user rating* was used as the primary demarcation factor for determining the success of a movie, where a rating of 7.5 and higher or a rating higher or equal to 7 but less than 7.5 but with total number of votes equal to or exceeding a median number of votes, classified the movie as a 'hit' while a rating of less than 7 or a movie with a rating of 7 but less than 7.5 and with a total number of votes less than a pre-set median number of votes, is classified as a 'flop'. The ratings and the number of votes where hence taken as the measure of success in this study. *Viewer-based features*, and most of *release-based features* were considered as being features available for late prediction, being features that would not be available at an early stage of production. *Movie intrinsic features*, such as genre, month of release, language, duration, budget and year of release were combined with derived features such as star performers' facebook likes, popularity rank, popularity factor, experience as well as a weighting to the year coefficient for popularity based features, to train a K-fold hybrid ensemble model (KHDEM) which is an ensemble of deep learning models like Convolutional Neural Network (CNN) and Artificial Neural Network (ANN) and conventional classifiers which include Random Forest, ExtraTreesClassifier and Gradient Tree Boosting. The dataset used contained 4043 Indian movies with 1688 positive and 2355 negative samples, split into 80:20 ratio. The proposed model achieved an Area Under Curve (AUC) score of 0.79 when

using just the movie intrinsic features and 0.989 when adding the derived features [88]. While the final model represents a notable improvement over other similar models, one can note that besides the data leakage occuring in the split of the dataset, certain intrinsic features such as duration of the movie, month of release and budget may not be available at the very early stages of script evaluation. Other derived factors such as the star performers' related features may also not be available at that early stage.

2.7.4 Using Natural Language Processing

Research into predicting movie success using NLP began with a 2006 study that used the simple "bag-of-words" approach, which only analysed movie spoilers, not full scripts [89]. An improvement came in 2010 when researchers extracted features like genre, word counts, and word meanings directly from actual screenplays [90].

By 2016, more advanced methods such as Latent Semantic Analysis (LSA) and TF-IDF were used on a large corpus to build a model predicting production performance [91].

Additional work was conducted in 2017, focusing on the analysis of movie scripts as unstructured text [92]. Using 978 movie scripts collected from the Internet Movie Script Database (IMSDB) [93], the authors applied the SentiWordNet [36] vocabulary dictionary to analyse each script into a series of positive and negative scores. They then smoothed the plots using spline interpolation, Locally Weighted Scatterplot Smoothing (LOWESS) [94], Savitsky Golay Filter and Gaussian Smoothing Filtering. From the resulting plots they extracted the highest and lowest positive score of a window for each script and used SimpleKMeans to cluster the scripts using the WEKA toolkit and setting the number of clusters to six, with the results suggesting that there are two popular patterns of story progressions.

Sentiment analysis on movie scripts and reviews to predict positive or negative reviews was researched in 2020 by using CountVectorizer to convert movie scripts into vectors and combining them with emotional features extracted using the National Research Council (NRC) [12] lexicon. Using the workflow depicted in Figure 2.15, Multinomial Naive Bayes was used to train the model consisting of 747 movie scripts collated from IMSDB and 78,000 reviews from Rotten Tomatoes [95] to achieve an accuracy of 76.8% [96].

Similarly to the work carried out by Eliashberg et al. in 2010, researchers in 2020 used the Latent Dirichlet Allocation (LDA) to extract hidden textual structures from 597 movie spoilers. Following the extraction of such topics, the authors calculate the probability of a script being assigned to each latent topic. Such topic assignment probabilities are used as the main predictors for movies' financial performance by adopting various classification algorithms. The target feature, being the ROI, is split

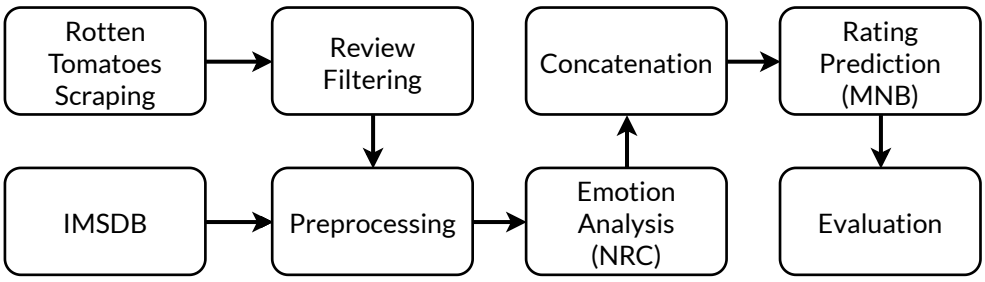


Figure 2.15 Applied workflow. *Source: [96]*

using the median, with movies with an ROI larger than the median being assigned a value of 1 and those movies with an ROI below the median being assigned a value of 0, thereby creating a classification exercise. In a model that includes the above mentioned topic assignment probabilities and when resampling data and performing model averaging they obtain an F1 score of 68.01% [97].

In an interesting approach to predict the success of a movie, the ROI was taken as the dependent target, with movies of ROI of less than 1 being considered as a failure and those with a ROI of 1 and over being considered as a hit. The novel approach taken in this study was the use of time series forecasting [98]. Using NLTK Sentiment Analyzer VADER(Valence Aware Dictionary for sentiment Reasoning) [10] mean positive, negative and compound scores were extracted from movie plots of 5019 movies taken from a Kaggle dataset [99]. LDA was then used to extract 15 topics from the same movie plots and used as features. The authors also split their dataset, which included movies released from 1990 to 2015, in a time split fashion using forward chaining technique which maintained the time order and reduced time-variability. Basing the selection on time, movies released before 2010 were used for training and the rest were used for testing purposes. Comparing their results with those obtained by Kim et. al [100] the model achieves an improvement in F1-Score of 11.9%, scoring 75.1% on their best performing dataset and model which was a voting based hybrid approach with Random Forest, Support Vector Machines, Gradient Boosting, Multi Layer Perceptron and Kim's Deep Learning Model [100]. Features used included the movie plot sentiment analysis features (including topic extraction) as well as features related to the genre of the movie. An embedding of the movie plot was also concatenated to these features to train Kim's Deep Learning Model, using ELMO [101]. This study highlighted the importance of the story of a movie, indicating that stories related to popular topics and the viewers' sentiment state at the time of release, have a higher opportunity of achieving success.

The use of the Linguistic Inquiry and Word Count (LIWC) analysis tool on movie scripts collated from IMSDB and *SimplyScripts.com* was combined with consumption experience carryover theory to predict movie revenues in a 2021 published study [102]. The authors concatenated 93 LIWC extracted topic features with movie properties

such as director power, star power, Motion Picture Association of America (MPAA) rating, whether a movie is a sequel, the genre, year of release and the movie studio involved and used an Auto-Gaussian (AG) spatial model [103] which augmented the LIWC categories by estimating each category in two measures: the traditional measure for the observed effect and the unobserved correlations between prior movies and the new movie. The idea behind the study was to use consumption experiences (which related to many different topics) to identify which topics have a positive or negative carryover effect from prior movies to new movies. Using a cross-movie similarity measure, the authors identify similar movies to a focal movie, thereby estimating the revenue of that focal movie based on the revenues of similar movies. Having included consumer reviews in their analysis, their best model, which achieved an adjusted R^2 score of 0.6695 utilized information which is only available after initial screening. Models based on script similarity alone also fared well with an an adjusted R^2 of 0.58.

The use of emotional arcs and the clustering of movie titles based on the shape of such arcs was demonstrated in a research which mapped the emotional trajectory of 6,174 movie script subtitles. These clusters were then used to predict the overall success parameters of the movies including box office revenues and viewer satisfaction levels (through IMDb ratings), awards as well as a number of reviews. In this study the authors find that movie stories are dominated by six basic shapes (see Figure 2.16) which reflect the work and observations of Kurt Vonnegut [104], with the highest revenues associated with the *Man in a Hole* shape, which resulted in financially successful movies independent of genre type and production budget [105]. While the prospect of using these well defined clusters for predicting user rating was promising, it was found that despite slight differences, mean ratings across the six clusters varied from 6.45 to 6.64 with a standard deviation between 0.94 and 0.99. Such variances may not be significant for creating important features out of these clusters. The authors' results showed that only the *Man in a Hole* cluster had significantly higher earnings in its home market than the others. A separate analysis also revealed a strong positive link between movies belonging to the *Rags to Riches* group and their IMDb ratings.

2.7.5 Word Embeddings

The growing popularity of word embeddings introduced a valuable tool for enhancing traditional frequency based metrics in text analysis. While the latter tools such as *Bag-of-Words* struggle with the semantics encoded in rich text and may not always consider the order of words, recent NLP researches introduced new techniques, mostly based on vector representation of text, as these proved to recover and retain semantic and syntactic information about natural langauge [107]. These models improved classification tasks, sentiment analysis, sentence classification, summarization, Named

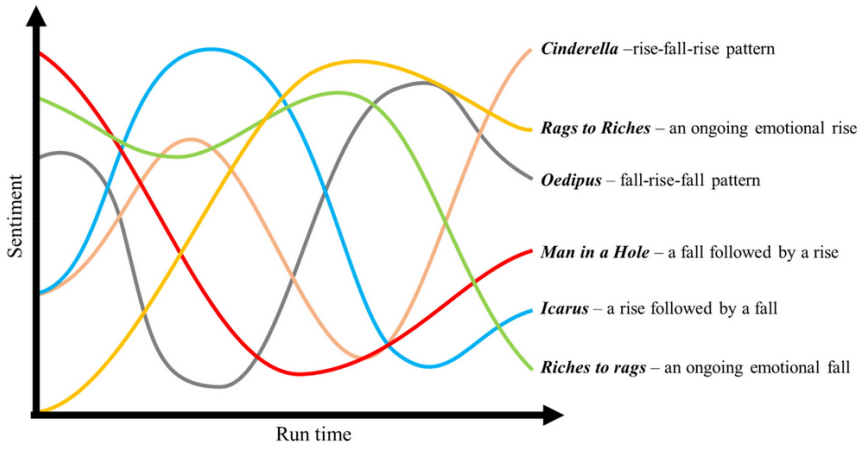


Figure 2.16 Six emotional trajectories of movies. *Adapted from [106]*

Entity Recognition (NER) and other NLP tasks.

While neural embeddings were first proposed in 2003 in the form of a feed-forward neural network model [108], early popular models included the Google developed model Word2Vec [109], [110] and the later modified Doc2Vec [111], Global Vectors represenation: Glove [41] and FastText [112]. The latter models work on different levels, with Word2Vec and Glove working on a word level, FastText on a character level and Doc2Vec on a paragraph level. These models were an inspiration to various research involving movie and tv series scripts. One project used Word2Vec with both a pre-trained dataset as well as a domain specific dataset to represent the text of 3000 TV series pilot episodes, combining the result with statistic and network features, distributed representation variables and NLP based features to predict genre classification [113] while a later thesis introduced deep learning networks to automate the extraction of features from a movie script that is converted into a vector using Word2Vec for the purpose of predicting box office gross revenue and total quantity of tickets sold [68].

Sentiment analysis was one of the tasks that has been improved by the use of word embeddings. Work carried out in 2021 [114], which employed a novel architecture combining Long Short-Term Memory (LSTM) with word embedding, extracted the semantic relationship between the neighboring words. Using a weighted self-attention applied to extract the key terms from movie reviews researchers trained a model on 25,000 perfectly balanced reviews and tested on an equally balanced test set of 25,000 reviews, managing to achieve an F1 Score of 88.67%. They used GloVe as the word embedding model, the results of which were passed through an LSTM model and a self-attention layer, a batch normalisation layer and a fully connected layer to obtain the predictions. A limitation identified by the authors in this work was the inability of GloVe to handle the embedding of words that would not form part of the model's vocabulary. FastText [112] tried to improve upon the Word2Vec model by

representing words as character n-grams, however the most notable improvements were registered by the use of embedding models based on the more modern transformer architecture [115]. This is due to a number of improvements offered by transformers, namely the contextualizing of embeddings where the embedding of a word changes depending on the surrounding words within a sentence [116], the tokenization of words into smaller units which improves the handling of rare words and the possibility of fine-tuning transformer models on domain specific datasets.

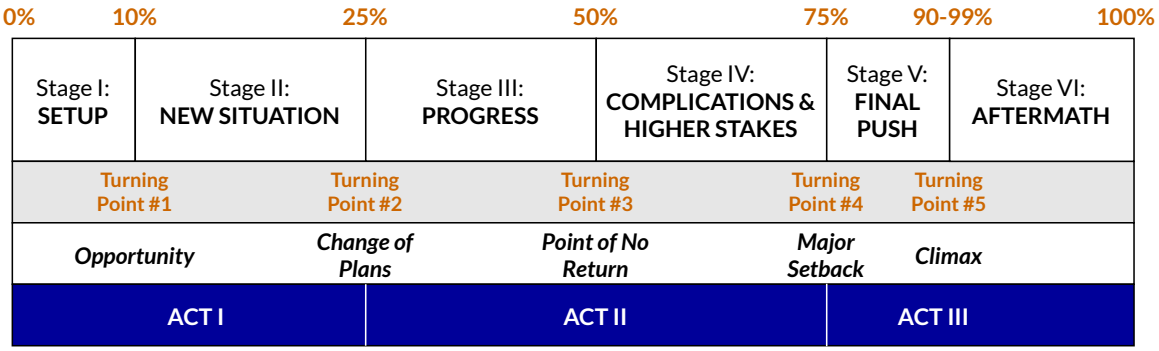


Figure 2.17 Michael Hauge's Six Stage Plot Structure. *Adapted from [117].* Image source: *https://storymastery.com/*

Transformers have also been used as screenplay summarization tools in an architecture which replaces LSTM to identify turning point events that change the story direction. Dialogue information, which is an attribute that can be extracted from screenplays, was shown to have a relation with such turning points in the story. Training a model to identify these turning points required the adding of a dialogue feature to the input embedding. Figure 2.17 shows how well structured stories consist of six stages which includes *Setup*, *New Situation*, *Progress*, *Complications*, *The Final Push* and finally *Aftermath*. Turning points divide such stories into multiple sections, thus defining the screenplay's structure [118]. Using the TRIPOD dataset [119] which contains screenplays and their turning points, the authors obtained improved performance in identifying such turning points by using transformers [120].

3 Methodology

In this chapter we outline the methodology used to achieve the objectives identified in the Introduction to this dissertation. In the previous chapter, after giving a background to the main researched topics, we carried out an analysis of scholarly literature relating to each element being explored in this dissertation. We will now delve further into work carried out to:

- Source data for the final model;
- Extract linguistic features from screenplays;
- Perform sentiment analysis on scripts, extract emotional arcs and cluster movies with similar emotional trajectories;
- Enhance script analysis with emotional intensity analysis;
- Embed scripts using various State of the Art (SOTA) embedding models;
- Explore inter-relationships between cast and crew;
- Choose and implement regression models;
- Choose and implement appropriate evaluation metrics.

3.1 Sourcing data

While a number of topics related to this dissertation have been extensively researched we could not locate or source a ready-made, curated dataset containing all the various elements and features required for the model. The first and most important element of the dataset comprises the scripts or screenplays of movies. Most research conducted on screenplay analysis made use of screenplays sourced from the IMSDB. At time of compilation 1,231 screenplays were downloaded from the website. Other scripts were sourced from [121], [122], [123] and GitHub repositories [124], [125], [126], [127], [128]. After converting Portable Document Format (PDF) into text, removing duplicates and unreadable text formats, a combined dataset of 2,785 full length screenplays was collated.

Given the restricted amount of available screenplays it was decided to augment the dataset with subtitles of movies. Subtitles have been used in other research, namely that of [106], [129], [130] and [105] with an adequate level of success in clustering similar titles. In the work of [131] which focused on using information derived from movie subtitles to enhance recommender systems, a user study of 247 individuals

indicated that information derived from subtitles can have an influence in determining whether a user will like the movie or not. Subtitles for this dissertation were sourced from [132], [133] and [131]. Duplicates, titles already obtained in screenplay format, non-English language titles, documentaries, series episodes, short movies and videos were removed and the subtitles were matched to the respective IMDb reference number using a process of matching titles to metadata obtained from other datasets. An additional 32,077 titles were added to the full length screenplay dataset, resulting in a total of 34,862 titles before filtering.

In conjunction with the sourcing of scripts and subtitles, the *metadata* which would be incorporated in the training data was being collated from multiple sources. Given the vast amount of research conducted in this area, numerous datasets could be identified for such data, mainly those from [134] and [135]. As not all scripts in the collated dataset had a matching IMDb reference number, a process of title similarity using Levenshtein distance [136] was applied to match titles to IMDb references. In cases of similar titles, the content of the script was analysed to match character names to the names contained in the respective IMDb page of that movie. Titles from [133] were scraped and matched using code developed by [137]. Once all scripts had a respective matching IMDb reference, the respective metadata could be linked to the proper title. In the process, any missing data was obtained by linking directly to IMDb using the Application Programming Interface (API) provided by the *Python* package *Cinemagoer* [138].

Another important source of direct information was obtained from IMDb. As from 18 March 2024, IMDb started to release and maintain, on a daily basis, datasets backed by a new data source. These datasets [139] contain selected fields of data pertaining to titles contained on the IMDb website. These datasets were used to update features that would be used in the dataset, fill in any missing data as well as for the crew relationship analysis that will be discussed later.

The following metadata features were collated for each movie title: *movie name*, *movie id*, *filename*, *genres*, *rating*, *votes*, *writer*, *producer*, *director*, *year of release*, *actors*. While the writer and producer might be known at the time of the first review of the script, it might be contended that the name of the director and main actors would only be known at a later stage.

In his dissertation [91] categorises prediction methods into three main categories:

- Early predictions - which are made before the film has been shot. [140], [141], [142], [5], [143], [80]
- Hype-based predictions - which are made after the film has been finalised but not yet released.

- Late predictions - which are made after the initial release of the movie.

In deciding what features to include in the training dataset, a balance between early prediction methods, historical statistics and other available factors to include in the training dataset was struck. The underlying rationale was to choose those features which can reasonably be assumed to be available before any major production budget is expended. We have decided to include director, writer, producer and actor related factors on the basis that these are the basic cast and crew who will be the first to be secured. For each key crew and cast member identified above we have looked at their past productions and calculated an average rating based on the IMDb rating of the movies they participated in the previous 5 years. We have also calculated an *experience* factor, by counting the number of productions they were part of. Using a rolling window training and evaluation methodology, to avoid any data leakage we ensured that ratings pertaining to the indicated cast and crew were only available up to the cut off year of the training dataset. For those titles for which the director, writer, producer or main actors would not be known, we have calculated an average of ratings for that category.

3.1.1 Limitations inherent to Subtitles

As detailed in Section 1.3, subtitles were employed to expand the dataset's size. However, because subtitles primarily consist of dialogue, they lack essential narrative components like scene directions, descriptions of visual atmosphere, pacing, and emotional nuances conveyed through non-verbal actions. These limitations detract from the intended immersive experience and omit emotional cues often expressed through facial expressions. Consequently, extracting linguistic features and charting emotional arcs for movies represented solely by subtitles becomes challenging due to these omissions, potentially affecting the predictive accuracy of both linguistic features and emotional arcs.

3.2 Linguistic Features

The linguistic features of the sourced scripts were analysed using the LIWC tool. The internal dictionary of this tool is composed of over 12,000 words, word stems, phrases and select emoticons. Each entry is allocated to one or more categories, designed to assess various psychosocial constructs [9]. Over 100 dimensions of feature data were extracted for each movie title.

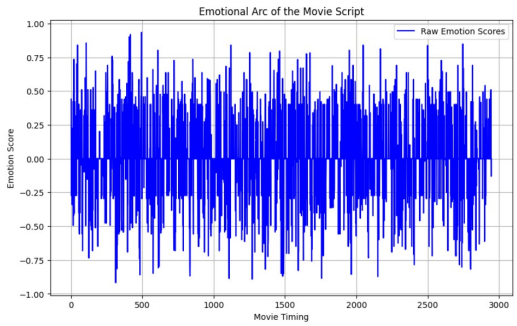
The use of LIWC is particularly relevant to this dissertation given that movie dialogue is an integral part of the word corpus used to build the software. A corpus of 19,970 movie subtitles was provided by *OpenSubtitles.org* for the construction of this

lexicon. For such purposes it is deemed that the extracted psychosocial cues are indicative of the expected audience perception of the movie. After pre-processing the text to lowercase the content, removing stop words and retaining only alpha characters, the scripts have been processed using the downloadable software.

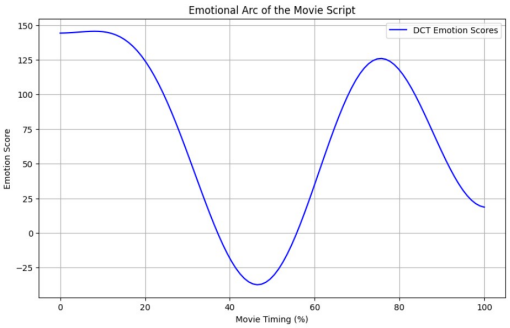
LIWC natively extracts percentages for most of the specific dimension contents within the whole corpus provided. Some dimensions, such as *word count* are reported in absolute values. After experimentation, it was deemed that converting these percentages to absolute values provides a better result in predicting user rating. Besides the primary *LIWC-Analysis* module, we also utilized the *Narrative Arc* companion module, which automatically evaluates texts based on how their narrative structure develops over time. This module generates corresponding graphs and metrics, indicating the extent to which each text aligns with a typical narrative pattern [9]. As many of the features exhibited multicollinearity when related to the rating value and between themselves, Recursive Feature Elimination (RFE) was employed to retain 115 out of the 151 dimensions to be concatenated to the main metadata. The full list of retained features is in Appendix A.

3.3 Emotional Arcs

A story's ability to resonate with an audience is often linked to the emotional journey it creates for them [144]. While script content is only part of the complete journey which is amplified by audio-visual features that make up the whole movie experience, analysing script content can provide a good insight into this emotional trajectory. To extract meaningful values from the script, a sentiment analysis tool was employed. We have used the VADER lexicon to extract the *compound* element of the polarity score of the text/script, which is split into sentences. VADER is a sentiment analysis tool that analyses feelings in text using a word list and rules. It is specifically tuned to understand sentiments expressed on social media [10]. The compound score is the overall sentiment measure for a sentence, scaled from -1 (most negative) to +1 (most positive). It is computed by adding up the sentiment scores of the words, adjusting them according to VADER's rules, and then normalizing the total. This single, normalised value is considered the most useful indicator of sentiment for a given piece of text.



(a) VADER raw compound values for the movie *A Few Good Men*. Script Source: [93]



(b) DCT values for the movie *A Few Good Men*. Script Source: [93]

Figure 3.1 Constructing the Emotional Arc for movies.

Given the different lengths of movie scripts, the raw score results are not homogeneous in length. In addition, plots generated based on the raw values do not provide a clear visual of the arcs. To smooth the results and represent them in a 100 dimensional vector (representing 100% of the length of a movie) we have used Discrete Cosine Transformation (DCT) from the *Scipy* package [145] which is a mathematical transformation that breaks down the input data into a sum of cosine waves of varying frequencies (3.1).

$$X(k) = \sqrt{\frac{2}{N}} \cdot c(k) \cdot \sum_{n=0}^{N-1} x(n) \cdot \cos\left(\frac{(2n+1)k\pi}{2N}\right) \tag{3.1}$$

where:

- $X(k)$ is the k-th DCT coefficient,
- $x(n)$ is the n-th input signal sample,
- N is the length of the sequence,
- $c(k)$ is the normalisation factor (3.2):

$$c(k) = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } k = 0, \\ 1 & \text{otherwise.} \end{cases} \tag{3.2}$$

After transformation, a low-pass size filter of 5 is applied to remove unwanted noise or fluctuations. Then Inverse Discrete Cosine Transformation (IDCT) is applied to convert the modified frequency components back to the time domain, thereby resulting in the smooth emotional arc, like that obtained in Figure 3.1b.

3.3.1 Clustering Emotional Arcs

Having extracted the emotional arcs and storing them into homogeneous 100 dimensional vectors, the next step required the selection of an adequate distance metric to use for clustering the vectors. Using the package *dtadistance* [146] the vectors are first pre-processed by applying *differencing* and *z-normalisation* using Scipy's *zscore* function [145].

Dynamic Time Warping (DTW) is a widely used technique for aligning time-dependent sequences by optimally matching their shapes, even when they vary in speed or timing. This nonlinear warping method allows sequences to be compared despite distortions in their temporal structure. Originally developed for automatic speech recognition to compare different speech patterns [147], DTW has since been applied in fields such as data mining and information retrieval to handle variations in time-dependent data [148]. Given its ability to measure similarities between sequences with differing temporal progressions, DTW is particularly well-suited for analysing the shapes of emotional arcs extracted from movie dialogue, where emotions may unfold at different rates across movies.

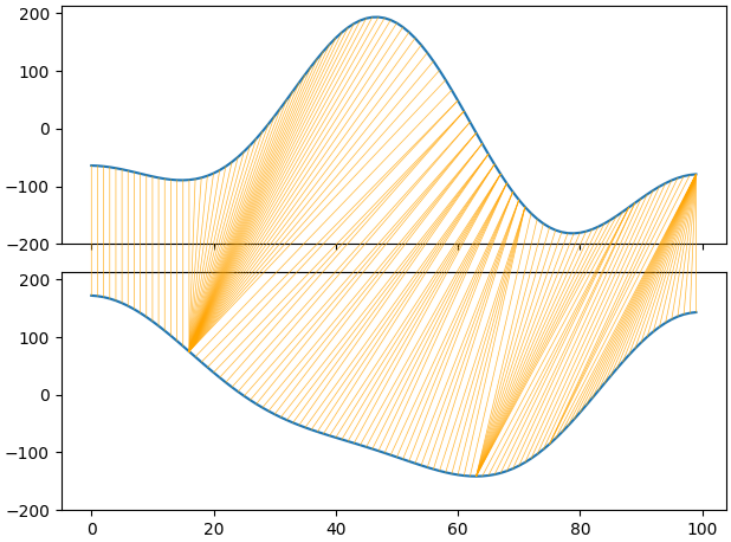


Figure 3.2 Using Dynamic Time Warping for measuring distance between two emotional arcs (Top arc: *Lethal Weapon*, Bottom arc: *Harry Potter and the Chamber of Secrets*). Image Generated by Author

Having created a distance matrix based on distance time warping we proceeded to select a clustering algorithm. Clustering the vectors into specific labels creates a label for that particular movie. Similarly clustered movies would have the same emotional trajectory and such label can be used as a feature in the regression model. As the distance between each vector was pre-computed, we needed to select an algorithm that allows for such parameter to be featured in the structure of the algorithm. As more popular algorithims such as *KMeans* do not allow for a pre-computed distance matrix to

be used, *KMedoids* [149] was selected, as implemented in the *Python* package *scikit-learn-extra* [150].

To select the optimal number of clusters the *silhouette coefficient* was used as a metric. This score measures the quality of clustering by looking at cluster cohesion, which is the space between clusters. The range of values for the silhouette coefficient is between -1 and 1, where the highest value indicates a better cohesion [151]. Iterating the clustering algorithm over 2 to 10 clusters, we calculated the silhouette coefficient using the *Scikit-Learn* implementation [152] for each output based on the pre-computed distance matrix. The algorithm achieved the highest silhouette coefficient at 6 clusters. The vector dataset representing the emotional arcs of the movies in the training dataset was updated with the labels representing the respective cluster each movie would be assigned to. The respective groups were subsequently plotted, with a mean arc, smoothed using *UnivariateSpline* from the package *SciPy* [153], plotted for each cluster, to represent the predominant emotional trajectory within that cluster of movies.

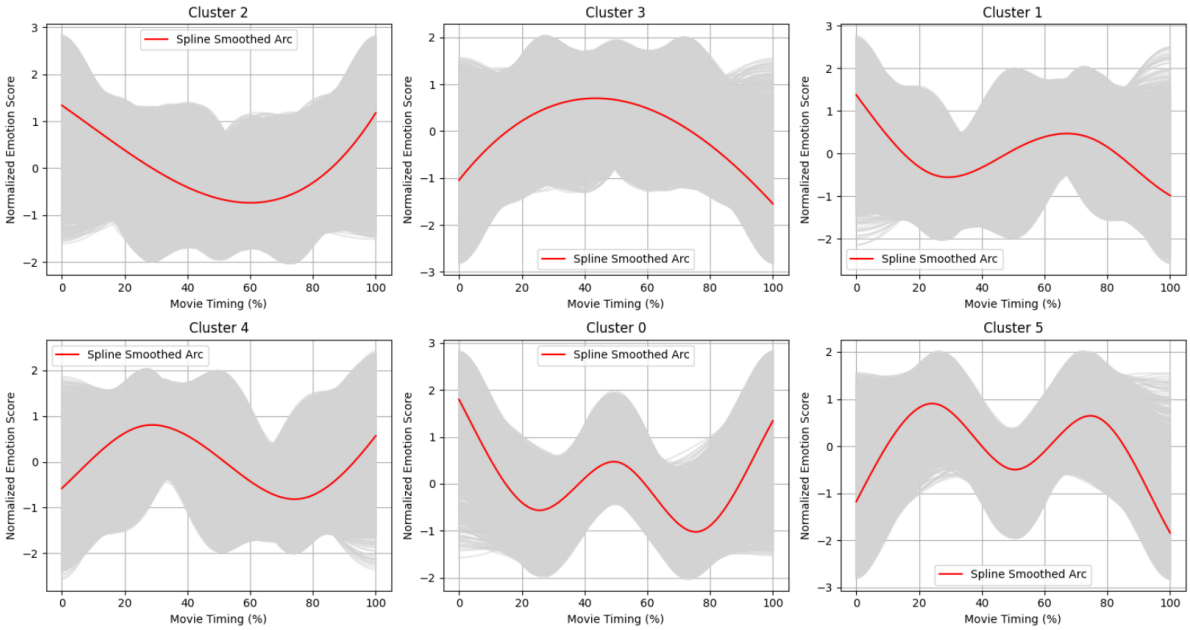


Figure 3.3 Univariate Spline Plots of the Six different clusters. Image Generated by Author

A research at the University of Vermont's Computational Story Lab analysed data from more than 1,700 English novels to reveal six basic story types, that form the building blocks for more complex stories [154]. These are:

1. *Rags to riches* – a steady rise from bad to good fortune
2. *Riches to rags* – a fall from good to bad, a tragedy
3. *Icarus* – a rise then a fall in fortune
4. *Oedipus* – a fall, a rise then a fall again

5. *Cinderella* – rise, fall, rise
6. *Man in a hole* – fall, rise

The outcome of the clustering exercise in this dissertation indicates that while the *Icarus*, *Oedipus*, *Cinderella* and *Man in a hole* type of stories have been also identified during the analysis, the more basic *Rags to riches* and *Riches to rags* type of stories, having a steady rise or steady fall type of story were not adequately embraced by the emotional trajectories of the dataset.

Instead, two different and opposing patterns have emerged in Cluster 0 and Cluster 5 respectively. Cluster 0 indicates a *[fall, rise, fall, rise]* type of narrative whereas Cluster 5 shows a *[rise, fall, rise, fall]* type of story.

Movie plots can be quite complex and while the clustering algorithm may have limitations in grouping data based on similarity in the vector space, the *Rags to riches* and *Riches to rags* patterns may not be distinctly represented, but rather may be blended into other clusters. While it is not in the scope of this dissertation to delve much further into the analysis of these patterns, it is significant to observe these two new patters which have been, for the purposes of this dissertation classified as:

1. *Phantom* – [fall, rise, fall, rise] - reflecting stories where characters experience alternating setbacks and recoveries, culminating in a bittersweet or redemptive ending
2. *Scarface* – [rise, fall, rise, fall] - reflecting stories where characters achieve success, face a downfall, recover, but ultimately experience another fall.

For the purposes of this dissertation the predictive power of such clusters has been explored further using a variety of statistical tests.

Table 3.1 Cluster Statistics.

Cluster	Count	Mean	Std	Min	25%	50%	75%	Max
0	2616	6.021 904	1.108 227	1.5	5.4	6.2	6.8	8.9
1	3176	6.099 937	1.076 559	1.7	5.5	6.3	6.9	9.0
2	3762	6.005 768	1.101 789	1.7	5.4	6.2	6.8	9.2
3	3317	6.130 329	1.074 615	1.5	5.5	6.3	6.9	8.9
4	3332	6.051 591	1.080 601	1.7	5.4	6.2	6.8	9.3
5	1946	6.070 709	1.091 908	1.8	5.5	6.2	6.8	9.2

As the ratings of the dataset do not follow a normal distribution, the Kruskal-Wallis H-test was implemented on the clustered ratings of the dataset. With a

p-value of 1.03e-06 the test indicated that the IMDb ratings exhibit statistically significant difference between clusters. This was analysed further using Tukey's HSD (honestly significant difference) test. This test interprets the statistical significance of the difference between selected means [155]. The results from this test showed that:

- Cluster 3 consistently has higher mean ratings than Clusters 0, 2, and 4.
- Cluster 2 has significantly lower mean ratings than Cluster 1 and 3.
- Clusters 4 and 5 do not significantly differ from any other group.

Finally we also computed Cramer's V measure which at the value of 0.027 indicated that being very close to 0, there is almost no relationship, in terms of strength, between the clusters and the ratings. Nonetheless, given the statistically significant difference in the means of the clusters, it was decided to retain the cluster as a feature for the regression model.

3.4 Emotion Analysis

3.4.1 Emotional presence and intensity

Emotions characterise human life. Affective processes are responsible for our behaviour, enacting and sustaining motivation. Research has found connections between affect and human performance, such as creativity and decision making [156]. There are various data analysis tools which can assist a researcher in extracting various emotional features from text. We have chosen two, lexicon based tools, to analyse the scripts in the dataset. These lexicons have been chosen to calculate intensity and presence scores.

- *The NRC Emotion Intensity Lexicon (NRC-EIL)* : The lexicon has close to 10,000 entries for eight emotions that Robert Plutchik argued to be basic or universal [12]. This lexicon is used to calculate the intensity score of the text.
- *NRC Word-Emotion Association Lexicon (aka EmoLex)* : The NRC Emotion Lexicon is a list of English words and their associations with eight basic emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) and two sentiments (negative and positive). The annotations were manually done by crowdsourcing [11]. This lexicon is used to calculate the presence score of the text.

The strength of different emotions in the text was calculated using a software component that identifies emotions linked to words and provides total scores for each

emotion. This tool, built upon the NRC Affect Intensity Lexicon, sometimes can link a word to multiple emotions with varying strengths. The process hence involves finding words from the text within this lexicon and retrieving the associated emotions and their respective intensity ratings (which range from 0 to 1, including decimals). It then adds up these intensity ratings for each type of emotion present in the text. Finally, it adjusts these total emotion scores by considering how many times those emotion-related words appeared, providing a weighted final score for the intensity of each emotion in the text [157].

The presence score was computed using the *Python* package *NRCLex* [158]. The affect dictionary contains approximately 27,000 words, based on the NRC and the Natural Language Toolkit (NLTK) Library's *WordNet* synonym sets [159]. The scores are exhibited as a percentage of the total number of words in the script.

Emotional affects measured include the following: *fear*, *anger*, *anticipation*, *trust*, *surprise*, *positive*, *negative*, *sadness*, *disgust*, *joy*.

Using Plutchik's Wheel of Emotions [160] we also computed features for composite emotions: *love*, *submissions*, *awe*, *disappointment*, *remorse*, *contempt*, *aggressiveness*, *optimism*.

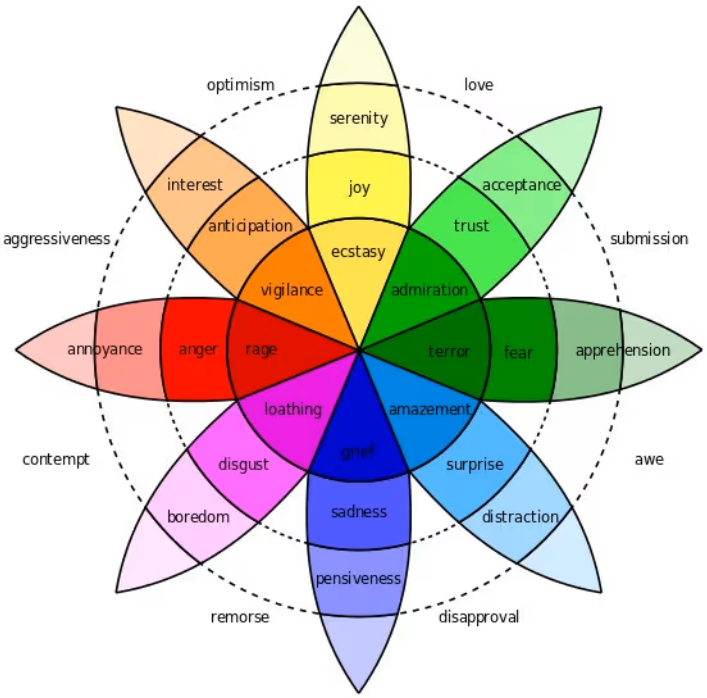


Figure 3.4 Plutchik's Wheel of Emotions. Image by Machine Elf 1735 [161]

Given the overlapping nature of some emotions, it was anticipated to have a number of emotions exhibiting multicollinearity. The respective correlation matrices for the scores indicated a high spearman correlation betweeen some of these variables, for

example *joy* and *optimism*. A Variance Inflation Factor (VIF) is a measure of the amount of multicollinearity in regression analysis [162]. In general terms, variables having a VIF greater than 5 are highly correlated. High correlation between independent variables can affect the regression analysis. Using the *variance_inflation_factor* class within *statsmodels* [163], only the features that had a VIF of 5 or less were retained. These included *surprise_intensity*, *love_intensity*, *contempt_intensity*, *positive*, *surprise*, *trust*, *fear*, *negative*, *disappointment* and *optimism*.

3.4.2 Statistical features from emotion analysis

Having extracted raw and DCT values from the script in Section 3.3, we combined the results in one comprehensive dataset to extract statistical features, with the aim of capturing patterns in the development of the emotional arc of each movie.

The features extracted from the raw and DCT values include *average*, *variance*, *standard deviation*, *minimum value*, *maximum value*, *range*, *median*, *kurtosis*, *skewness*, *average slope*, *maximum slope*, *minimum slope*, *total curvature*, *mean trough to peak slope*, *permutation entropy*, *shannon entropy*, *autocorrelation*, *mean rate of change*, *standard deviation of rate of change*, *zero crossings*, *positive proportion*, *negative proportion*, *neutral proportion*, *distance peak to peak*, *volatility*, *mean peak to trough slope*, *number of peaks*, *number of troughs*

Using a similar approach adopted in the selection of features from the presence and intensity emotion scores, we removed independent features which were highly correlated to each other and using the *mutual_info_regression* class from Scikit-learn [164] we retained only those features which contributed a positive mutual information score. Mutual information is the application of information gain to feature selection and is calculated between two variables, measuring the reduction in uncertainty for one variable given a known value of the other variable. Higher values indicate a closer connection to, in this case, the rating dependent variable [165].

The retained features include *zero_crossings*, *skewness*, *negative_proportion*, *max_slope*, *range*, *shannon*, *rate_of_change_std*, *volatility*, *num_peaks*, *kurtosis*, *rate_of_change_mean*, *average*, *max*, *peak_to_peak*, *std*, *variance*, *general_positive_proportion*, *neutral_proportion*, *mean_peak_to_trough_slope*, *min*, *min_slope*, *average_slope*.

3.5 Script Embedding

Applying the concepts and restrictions discussed in Section 2.3 required the consideration of:

- choice of embedding model;

- choice of chunking strategy;
- what chunk size to use;
- what pooling mechanism to use for the final embedding vector.

3.5.1 Embedding Models

The performance of the regression model was tested using these embedding models: *all-MiniLM-L6-v2* [166]; [167], *gte-modernbert-base* [168]; [169]; [170], *ModernBERT-embed-large* [171]; [172], *UAE-Large-V1* [173]; [174], *stella_en_400m_v5* [49]; [175], *jasper_en_vision_language_v1* [49], *stella_en_1.5B_v5* [49]; [176].

Name of Model	No. of Parameters	Max. Sequence Length	Embedding Dim.
all-MiniLM-L6-v2	22M	256	384
gte-modernbert-base	149M	8192	768
UAE-Large-V1	335M	512	1024
ModernBERT-embed-large	395M	8192	1024
stella_en_400m_v5	435M	8192	4096
jasper_en_vision_language_v1	1B	131072	8960
stella_en_1.5B_v5	1B	131072	8960

Table 3.2 Comparison of embedding models used in this dissertation.

The above models where chosen based on the available infrastructure for embedding the selected dataset. At the time of writing most of these models were also among the top performing models in the MTEB Leaderboard. While some of the models are capable of a maximum output embedding dimension of 4096 or more, due to computation time restrictions we decided to retain a maximum of 1024 dimensions.

3.5.2 Chunking Strategy and chunk size

As all of the tokenized screenplays exceed the maximum input size of the selected models, chunking was required. There are different methods for chunking, applicable to different situations, the most common being *fixed-size chunking*, where the number of tokens in each chunk is pre-determined. While other methods such as *sentence splitting*, *recursive chunking* and *semantic chunking* [177] exist, given the straightforward understandability of *fixed-size chunking*, it was decided to apply this method for the embedding models.

Determining the size of each chunk presents a large number of options. With the objective of finding a balance between preserving context and maintaining accuracy, one has also to keep in mind the maximum number of input tokens each model is able to

handle. As the focus of this dissertation is not the determination of optimal strategies for the embedding model, it was decided to use a fixed-size chunk of 512 tokens for models capable of handling such a number of input tokens and 256 tokens for the smaller *all-MiniLM-L6-v2*.

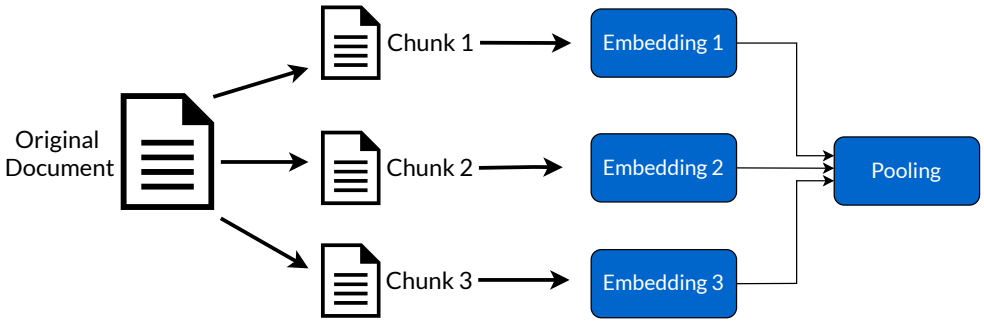


Figure 3.5 The chunking process. Image generated by author

3.5.3 Pooling mechanism

Splitting screenplays of circa 30,000 tokens into fixed-size chunks of 512 tokens each, results in around 60 vector embeddings. While an option to represent the whole movie would be that of concatenating all the embeddings, this would result in extremely long vectors, leading to slow processing and possible noise in the dataset. Additionally, given the different lengths of screenplays, the resulting number of vector embeddings would be different for each movie. To this end, a common practice to aggregate token embeddings into unified representations is to employ a pooling strategy. Popular pooling strategies include *Mean Pooling*, which averages embeddings to create a balanced representation, *Max Pooling*, which selects the most prominent features, and *Weighted Sum Pooling*, which assigns learned weights to highlight contextually important tokens [178].

Although researchers have theorized about story structure for a long time, few studies have empirically tested these theories. While classifying different plot types has value, this approach does not explain whether certain story elements are more captivating or why [179]. The *Peak End Rule* principle suggests that individuals' overall perception of an experience is heavily influenced by its most intense moments and its conclusion, rather than the experience in its entirety. Originally identified in cognitive psychology, this principle applies to various domains, including cinema. In the context of movies, the ending plays a crucial role in shaping viewers' lasting impressions. A compelling or unsatisfactory conclusion can significantly impact the audience's overall assessment of the film [180];[181].

To this end a custom weighted approach was taken to unifying embeddings. Using a linearly weighted scale, an increasing weight was allocated to embeddings as

the movie progresses, with embeddings for the ending part of the movie being allocated the highest weighting. An example of this weighting mechanism for a movie's screenplay, split into 60 vector embeddings, is shown in Figure 3.6.

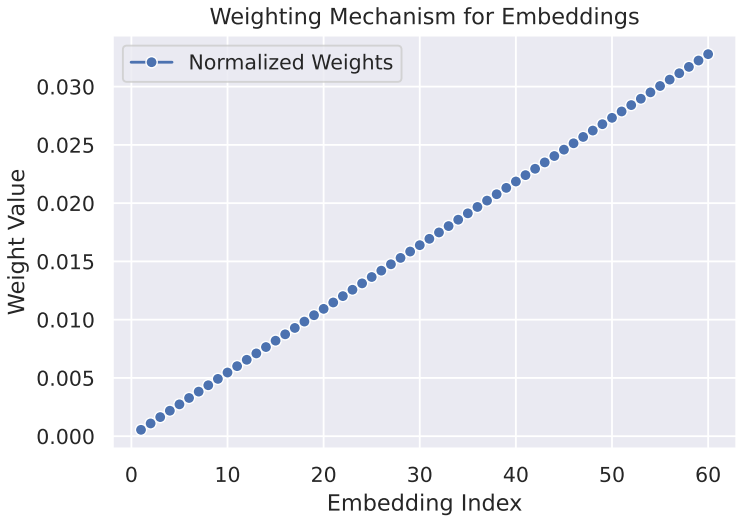


Figure 3.6 A weighting mechanism structure for a movie split into 60 vector embeddings. Image generated by author

3.5.4 Plot keywords

Given the final embedding vector of the average script results in the compression of the number of tokens required for a whole script by circa 95% (average script having circa 30,000 tokens versus the final embedding vector of 1024 dimensions for the models having such a dimension output size), it was also decided to include another feature which represents keywords from the plot of each movie. Keywords were obtained from [99], [182], [183] as well as manually extracted using *KeyBert* [60] from plot summaries.

Saved as a list, the keywords were cleaned by using *WordNetLemmatizer* from NLTK [159] and removing stop words, The processed keywords were subsequently used to train a *Word2Vec* model with vector size 50, a window of 15, min_count of 1 and sg of 1. Each 50 dimensional vector was added to the main dataset as a feature.

3.6 Cast and Crew Inter-relationships

A significant amount of data is made available by IMDb with its regularly updated non-commercial datasets [139]. Besides extracting *star power*, *director power*, *writer power*, *producer power* and related experience features, given the historical data provided, it was also possible to analyse the interactions between cast and crew, through graph analysis. The quality of a film is shaped by the various artists and experts who collaborate during its production. While studies have largely concentrated on

whether movie stars ensure success in terms of making money, these stars are only one among many contributors to the final film [69]. The ideal composition of a film crew varies based on the specific measure of cinematic success. Research in 2005 on Italian movies found that strong professional relationships between directors, producers, and distributors contributed to better financial performance. In contrast, critical acclaim and industry awards were more common when directors maintained weaker connections with writers, actors, and cinematographers [184].

Having created a dataset made up of individuals (being producers, writers, actors or directors) working on each movie as shown in Table3.3, we used *rating* as weights to create a list of collaboration edges between individuals.

movie_id	category	Name	Rating	Year
1	director	William K.L. Dickson	5.7	1894
1	producer	William K.L. Dickson	5.7	1894
2	director	Émile Reynaud	5.6	1892
3	director	Émile Reynaud	6.5	1892
3	producer	Julien Pappé	6.5	1892

Table 3.3 First 5 rows from the Principals Dataset

Using *cugraph*, which is a library of graph algorithms which integrates into the RAPIDS data science ecosystem [185], we created the graph to incorporate the collaboration edges previously collated. *NetworkX*, which is a Python package for the creation, manipulation and study of the structure, dynamics and functions of complex networks [186] was subsequently used to better visualize the connections.

Feature extraction was mainly conducted by analysing centrality measures. These quantify how central or how important vertices or edges are inside a network [63]. The first centrality measure calculated was the *degree centrality*. Vertices with large degrees are central to the network. Within the movie industry network, an individual with a high degree centrality would have worked with many different people in the industry, hence indicating experience and reputation, influence and centrality.

A node is often seen as more important if many shortest paths between other nodes pass through it. This is quantified by *betweenness centrality*, which measures how often a node lies on the shortest route between every pair of other nodes in the network. Nodes with high betweenness can have considerable influence because they act as crucial connectors, potentially controlling the flow between others [63]. Putting this into the context of the movie industry could identify individuals who bridge different groups of collaborators.

PageRank, which was originally developed by Google to rank web pages in search results, measures the importance of a node based on the number and quality of incoming links. The better the incoming links, the more important is that node. Within the current dataset, *PageRank* can be used to assess the influence of cast or crew.

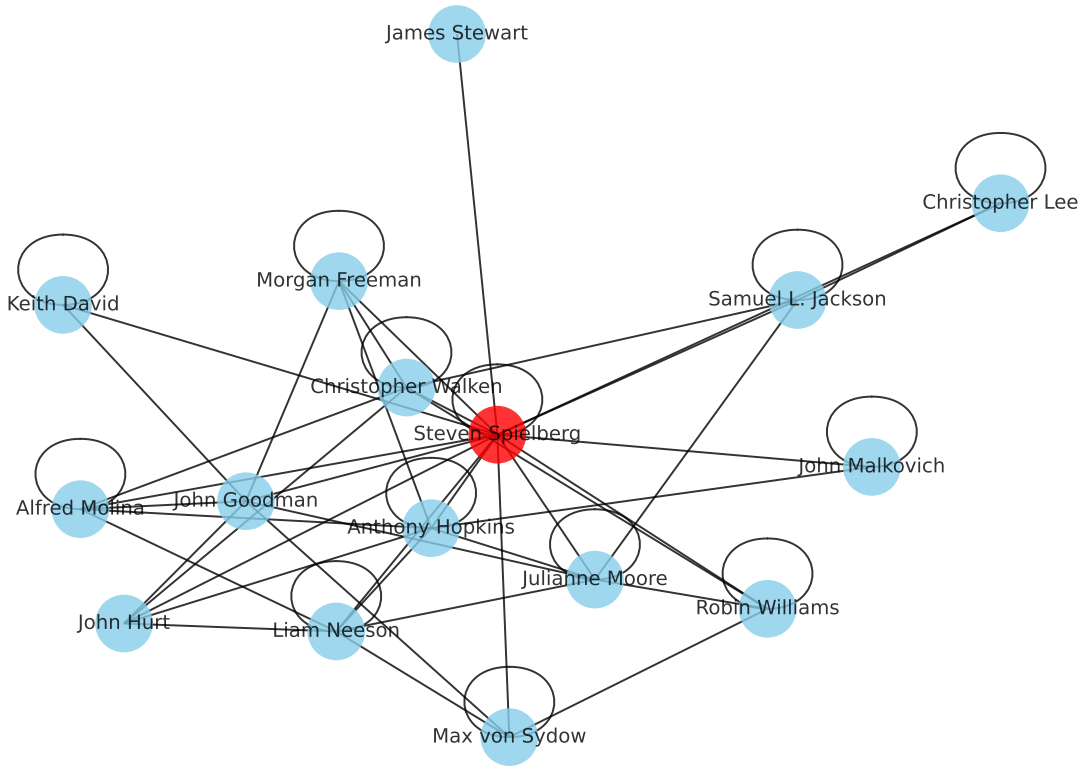


Figure 3.7 An example of the top 15 connections with the director Steven Spielberg. Image generated by author

Principals who have worked with many other influential individuals would receive a higher *PageRank* score.

Eigenvector centrality is similar to *PageRank* in that a node is important if it is connected to other important nodes, but *Eigenvector centrality* places more emphasis on the quality of those connections. It is calculated by taking the eigenvectors of the adjacency matrix of the network. The eigenvector corresponding to the largest eigenvalue represents the centrality scores of the nodes. Again, this feature helps to identify the most influential individuals within the network of cast and crew.

As I have used *rating* as weights for the edges, an *average collaboration rating* was calculated for each vertex, as an additional feature.

3.7 Selection of Regression Models

Given the nature of the dependent variable, the IMDb rating, in this study being a continuous value, a regression model was deemed to be most relevant to the task. Various regression models were tested for the experiments:

- XGBoost [15]

- LightGBM [16]
- Support Vector Machine [187], [188]
- Lasso [189], [188]
- CatBoost [19]
- ElasticNet [190], [188]
- MLPRegressor [188]

The study employed a diverse set of machine learning models for predicting movie audience ratings. Gradient boosting machines like *XGBoost*, *LightGBM*, and *CatBoost*, which are based on decision trees, were used alongside linear models such as *Lasso* and *ElasticNet*. A neural network model, *MLPRegressor*, was included to capture non-linear relationships and complex feature interactions. Additionally, SVR, a kernel-based method effective for modeling intricate relationships, was integrated, with regularization to improve results despite its potential computational cost with large datasets.

To prevent overfitting and enhance model performance, regularization techniques were applied. Lasso (L1 regularization) [189] encourages feature selection by shrinking coefficients of less important features to zero. Ridge Regression (L2 regularization) [191] reduces the magnitude of coefficients, addressing multicollinearity without eliminating features entirely. ElasticNet combines both L1 and L2 regularization, offering a balance between feature selection and coefficient shrinkage.

3.8 Selection of Evaluation Metrics

Having trained various regression algorithms, with the predicted target being a continuous value, the comparison to actual ratings naturally necessitated regression evaluation metrics. Nonetheless, as was discussed in Section 2.7, a number of researchers have used a classification evaluation to the regression analysis. Classification mainly required movie ratings to be allocated to pre-defined bins and categorising movies into one of pre-defined performance measures. In this dissertation a similar approach has also been taken to complement the regression evaluation metrics.

3.8.1 Regression Evaluation Metrics

As with most data science projects, no universally optimal metric exists for evaluating the performance of a regression model [192]. We have used some metrics for optimising

model training while reporting a mixture of others to assess model performance.

R-squared

R-squared (R^2), represents the proportion of variance explained by a model. It corresponds to the degree to which the variance in the dependent variable (in our case the audience rating) can be explained by the independent variables (the collated features) [192]. The formula for R^2 is given by:

$$R^2 = 1 - \frac{RSS}{TSS} = 1 - \frac{\Sigma_{i=1}^{N} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{N} (y_i - \bar{y}_i)^2} \tag{3.3}$$

where RSS is the sum of squared residuals (capturing the prediction error of the model) and TSS is the total sum of squares.

Usually having a range of 0 to 1, a higher value indicates a better fit. However in the case of non-linear models, it is possible to obtain a negative R^2. In such cases the model's predictions would have fit the data worse than a simple mean model. This metric is to be viewed in conjunction with the field it is being interpreted in. While hard sciences like physics and chemistry, having well-defined laws, may lead to high predictability and hence expectations, social sciences which would also include the analysis of movie audience ratings, are afflicted with human behaviour and other social phenomena that are difficult to measure. As such, lower R^2 are more common and acceptable.

Mean Absolute Error

MAE is expressed in the same scale as the target variable, making it easier to interpret for the specific dataset on which the model is being evaluated, but it does not allow for comparison between different datasets. While it does not indicate the direction of the errors (the model may be constantly over or underforecasting the ratings), it treats all errors equally, rendering it robust to outliers. Its mathematical formula is given by:

$$MAE = \frac{1}{N} \Sigma_{i=1}^{N} |y_i - \hat{y}_i| \tag{3.4}$$

This metric is particularly useful for the evaluation of this dissertation's model, in that it provides an understandable value of the average absolute error in the ratings prediction. A result of 0.8, for example, would indicate that on average, errors in rating prediction are of $\pm$0.8.

Root Mean Squared Error

By squaring the errors, RMSE puts a higher penalty on large errors. Consequently, in datasets containing outliers, RMSE may suffer slightly. It is however useful for optimisation algorithms, but while it is also on the same scale as the target, given the squaring of the errors, it cannot be interpreted in the same easy way as the MAE. Its formula is given by:

$$RMSE = \sqrt{\frac{1}{N}\Sigma_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{3.5}$$

This metric was used for evaluation and optimisation of certain models to assess and help the models focus on minimising large errors. Due to the presence of some very low or very high ratings, with a predominance of a cluster of ratings within the dataset, it was also deemed appropriate to analyse the results of the predictions from this perspective.

Mean Absolute Percentage Error

MAPE is another easily understandable and interpretable metric. Expressed as a percentage, it makes it a scale-independent metric and can hence be used to compare predictions across different datasets. Favouring the side of caution, this metric is suitable for the analysis of the results for this dissertation in that it puts a heavier penalty on negative errors (where rating predictions are higher than actuals). While being easily understandable, a downside of this metric is its relative nature. For a rating prediction of 3 and an actual of 3.6, the absolute error is 0.6 and the MAPE is 0.2 (or 20%) while for a prediction of 9 and an actual of 8, the absolute error is 1.0 but the MAPE is 0.125 (or 12.5%) which might give a false impression that the second prediction was better. It is hence important to keep in mind that MAPE provides the result relative to the actuals and should be interpreted accordingly.

$$MAPE = \frac{1}{N}\Sigma_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{3.6}$$

3.8.2 Classification Evaluation Metrics

The results obtained from this dissertation allowed for a wider use of metrics to explain different aspects of the results. To provide a more comprehensive evaluation we have decided to use the following additional classification metrics.

One Away Score

This score is a specific adaptation of the simple accuracy score, to give a percentage of records from the test set where the predicted rating was within ± 1 of the actual rating.

$$One\ Away\ Score = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\left(|\hat{y}_i - y_i| \leq 1\right) \tag{3.7}$$

Categorising movies

For the following set of classification metrics movies were categorised into three categories, based on the actual or predicted rating as per Figure (3.4). Such categorisation was also implemented in the works of [193] and [194].

Rating Range	Classification
0.0 – 4.0	Low
4.1 – 7.0	Medium
7.1 and higher	High

Table 3.4 Classification of Movies Based on Ratings

Various evaluation metrics can be considered, including accuracy, precision, recall, and the F1-score. Given that the F1-score accounts for both precision and recall, this metric, along with accuracy and a confusion matrix, will be utilized for performance evaluation.

Prior to discussing accuracy and the F1-score, it is essential to introduce the concept of the confusion matrix. In the context of movie rating predictions, where movies are categorised into three possible classes as described above, a multi-class confusion matrix is employed. This matrix provides a structured representation of the actual versus predicted classifications, capturing the model's performance across all categories. Each entry in the matrix corresponds to the number of instances where a movie belonging to one category (e.g. Low) is predicted as another (e.g. Medium or High). The diagonal entries of the matrix represent the instances where the model correctly classified the data point into its true class. The off-diagonal entries, conversely, show the instances that were incorrectly classified into a different class.

Actual \ Predicted	Low	Medium	High
Low	TP(Low)	FN(Low→Med)	FN(Low→High)
Medium	FP(Med→Low)	TP(Medium)	FN(Med→High)
High	FP(High→Low)	FP(High→Med)	TP(High)

Table 3.5 Confusion Matrix for Movie Rating Predictions

In Table 3.5 True Positives (TP) refer to correctly classified movies, False Negatives (FN) are actual movies of a class that were misclassified into another class,

while False Positives (FP) are movies predicted to be a certain class but actually belong to another class. The difference in the latter two nomenclatures emanates from the perspective with which one would be analysing the misclassification, whether from the actual class (FN) or the predicted class (FP) [195].

Accuracy

Accuracy measures the proportion of correctly classified instances out of all instances. In this part of the dissertation's context, it tells how often the model predicts the correct movie rating category. It is calculated as:

$$Accuracy = \frac{\sum \text{True Positives}}{\sum \text{All Samples}} \tag{3.8}$$

F1-Score

F1-Score is the harmonic mean of Precision (how many of the movies predicted as a given class were actually of that class) and Recall (how many of the actual instances of a class were correctly predicted), calculated for each class and then averaged. As we have a multi-class classification, the F1-score for each class needs to be individually computed and then take a weighted average of these F1-scores, where the weights are determined by the number of true instances for each class.

Precision (P) is calculated as:

$$P = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{3.9}$$

while Recall (R) is calculated as:

$$R = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{3.10}$$

The F1-Score is calculated for each class as:

$$F1 = 2 \cdot \frac{P \cdot R}{P + R} \tag{3.11}$$

and then weighted accordingly:

$$\text{Weighted } F1 = \sum \left(\frac{\text{Total Instances of Class}}{\text{Total Samples}} \times F1(\text{Class}) \right) \tag{3.12}$$

F1-score is provided as a value between 0 and 1, with overall high scores indicating that the model has both good precision and high recall, meaning it rarely misclassifies categories of movies and mostly correctly identifies most movies in each respective categories, leading to reliable results.

4 Implementation

In the previous chapter we laid out the methodology adopted for this research. We now elaborate on the implementation of this methodology.

4.1 The training dataset

Feature data has been extracted from screenplays or subtitles, as the core source of information for this dissertation. These have been concatenated to movie metadata sourced from IMDb.

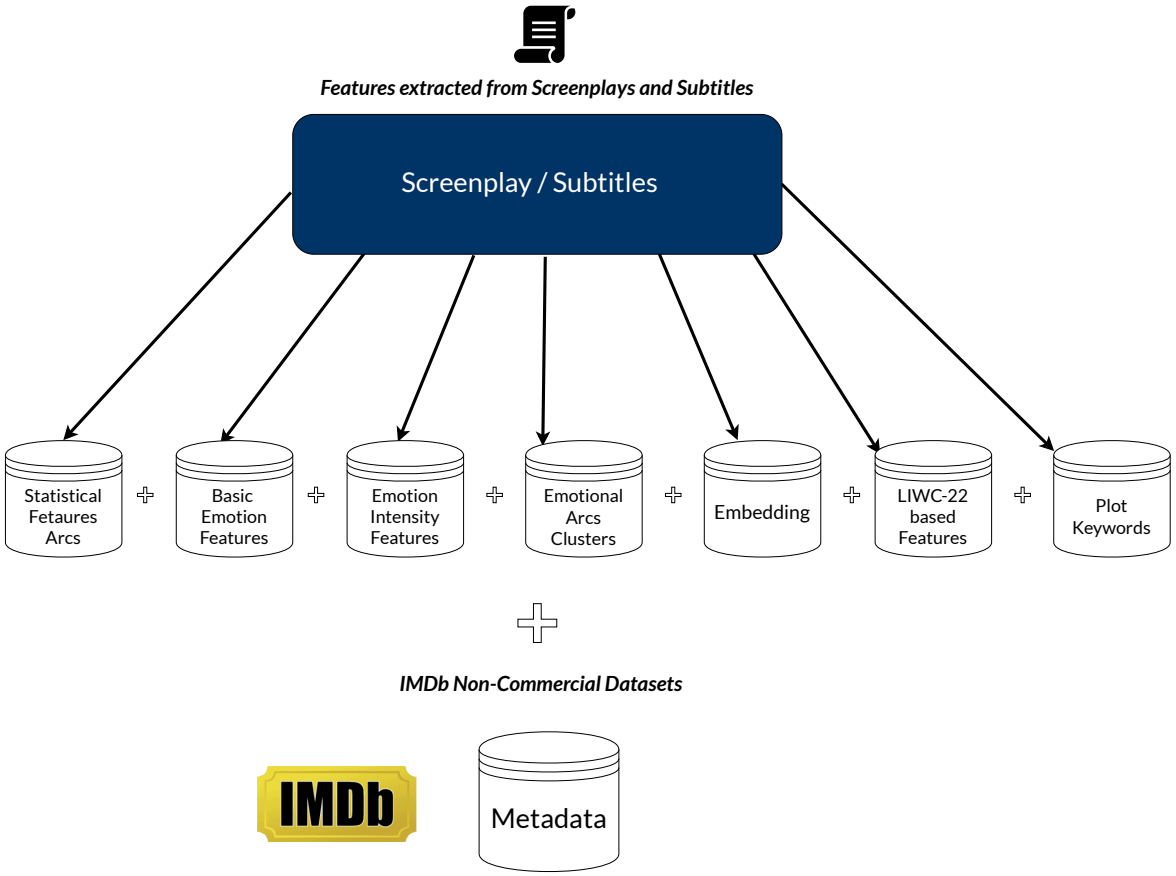


Figure 4.1 Composition of training dataset. Image created by author

The ratings of the movies have been updated as at 19 February 2025. IMDb's rating system aggregates individual ratings submitted by registered users to compute a single rating. However, instead of using a simple arithmetic mean (i.e., the total sum of votes divided by the number of votes), IMDb employs a weighted average. While the mean and average votes are displayed on the vote breakdown page, the rating shown on a title's page reflects a calculation that assigns different weights to different votes. This

means that although all user votes are counted, not all have the same influence on the final rating.

To maintain the system's integrity, IMDb may apply an alternative weighting method when unusual voting patterns are detected. The exact formula used to generate ratings remains undisclosed to prevent manipulation and ensure fairness. This methodology makes IMDb's rating system a reliable reflection of user opinions. However, to highlight more widely recognised films, any movie with fewer than 1,000 registered votes is excluded from the ratings display [196].

Movies not having English as their original language, documentaries, music or biographies were removed with the final dataset containing 18,143 titles released between 1915 and 2024.

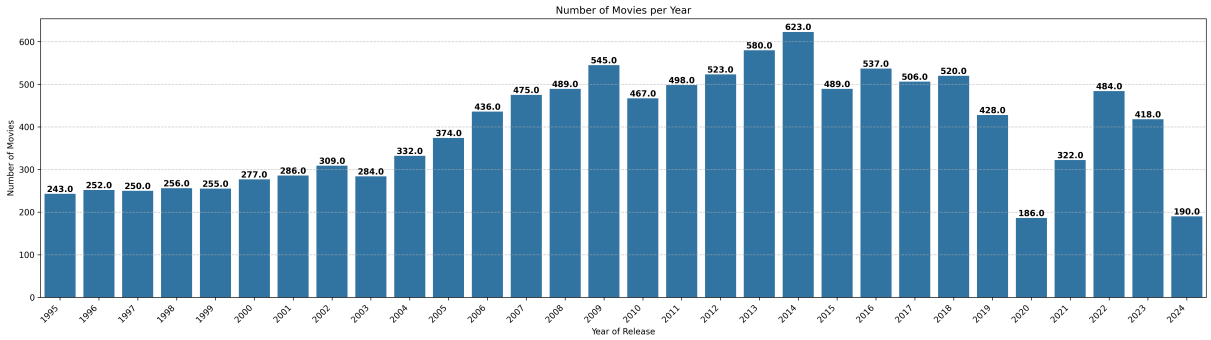


Figure 4.2 Number of movies from 1995 onwards. Image created by author

The analysis of the dataset forms an important part of the understanding of the data, especially with respect to the distribution of ratings across the years and within each year, and to determine whether any outliers are present. Figure 4.2 shows the number of movies present in the dataset released from 1995 onwards. A general decrease in audience rating is highlighted in Figure 4.3.

In Figure 4.4 a box plot and a histogram of ratings for the dataset show the first and third quartile, with the bulk of the ratings falling in between these two boundaries. The upper and lower bounds are calculated as $\pm 1.5 \times InterQuartileRange(IQR)$. In the context of movie evaluations, very low or very high ratings normally reflect strong audience opinions about the movies and hence should not be treated as anomalies.

4.2 Rolling Window Validation Strategy

Building a training strategy for a movie rating prediction algorithm necessitated the consideration of the time factor. As it was demonstrated above, audience affinity may change over time. Certain genres, like the super hero themed movies boom, may result in a successful release of a number of similarly themed movies over a span of years. The experience of crew and cast would indicate a general trend towards better ratings.

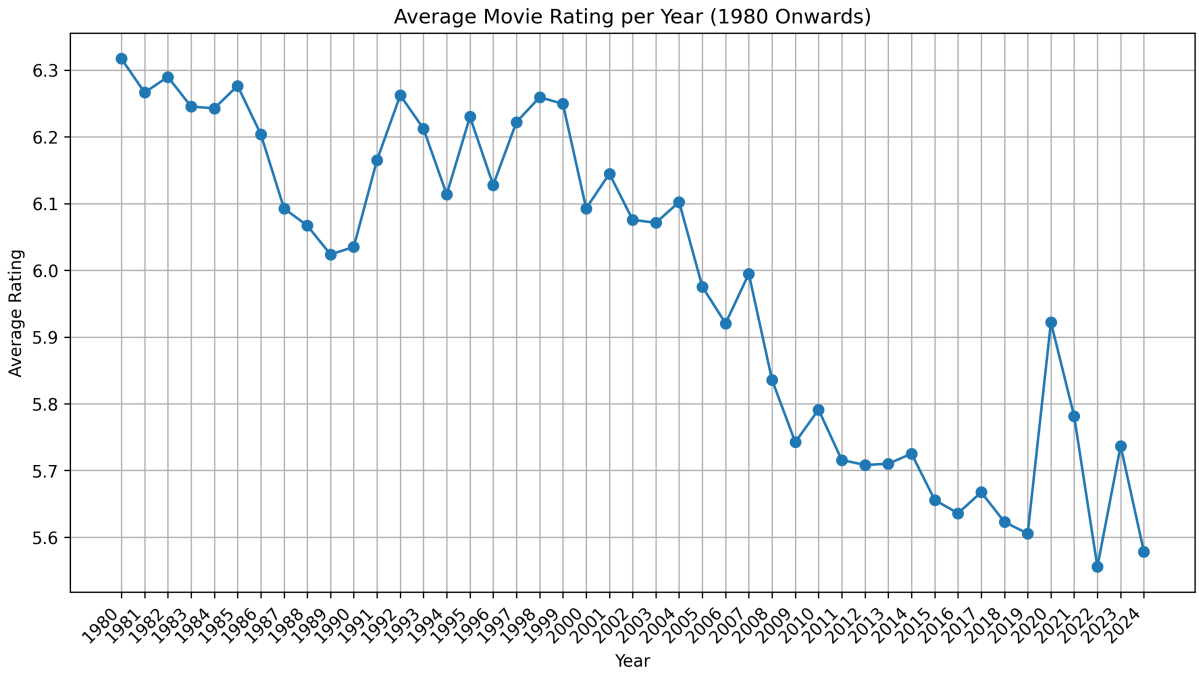


Figure 4.3 Average rating of movies from 1980 onwards. Image created by author. Source: [139]

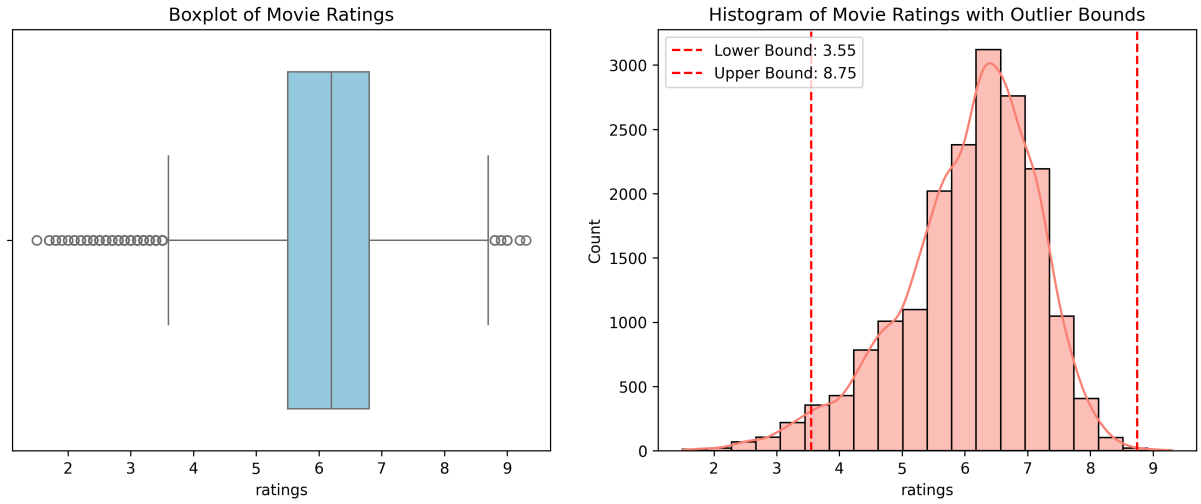


Figure 4.4 Analysis of potential outliers. Image created by author. Source: [139]

Actors establishing themselves would tend to be more selective on the movies they star in, in a strive to keep up the good reputation. All these factors, together with trends in creativity, improved visuals and the medium of screening, require a model to be up to date in terms of the latest trends, for it to retain its generic predictability power. To account for these factors, a rolling window validation strategy was adopted.

An initial cut-off year of 2015 was selected as the total amount of movies released before this year represent circa 80% of the whole dataset, with movies released 2015 up to 2024 representing circa 20% of the dataset. Training folds comprised movies released up to 20 years before the cut-off year and validation was carried out on movies released in the cut-off year and the subsequent 2 years. This

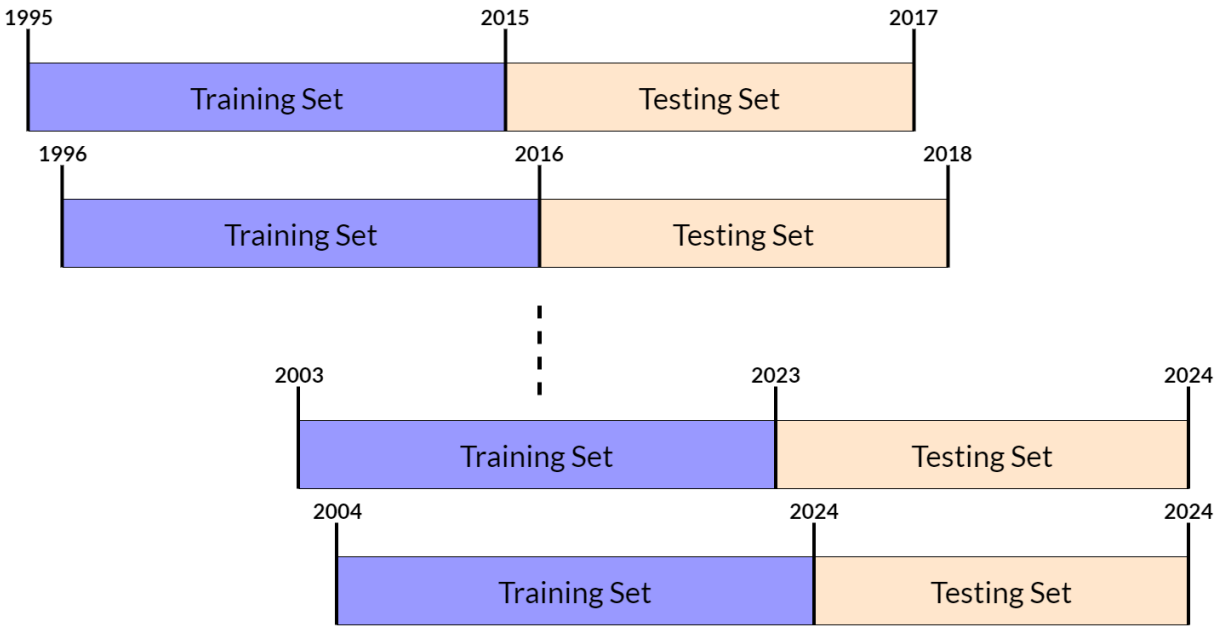


Figure 4.5 Rolling Window Validation Strategy. Image created by author

training and validation sequence is repeated by moving the cut-off year forward by one year until the last validation set being the movies released in 2024. Using this fixed-length training window (20 years) that shifts forward over time, model evaluation respects the chronological order of data to prevent data leakage.

4.2.1 Updating of Training Set

A key component of the rolling window validation strategy was the need to update the training set with each forward shift. This update focused on the historical rating average for directors, writers, producers and main actors. Sourcing from a dictionary which stored the rolling average ratings of the previous five years of productions by each crew/cast member for each year from 2014 onwards, the training set features pertaining to these cast and crew members are automatically updated for the subsequent training fold. Similarly, the *experience* feature is also updated with the count of any new movies the respective cast or crew members would have been involved in the previous year.

To account for new cast or crew members, the median of the rating of the respective category is taken, less an arbitrary 1.0 rating as a *newbie* discounting factor. This value is taken in situations where the model would encounter a new cast or crew member, not present in the dictionary. In the case of the *experience* feature, where new cast or crew members are encountered the value of 0 is imputed.

It is important to emphasise the considerable effort undertaken to rigorously prevent any form of data leakage, particularly concerning retrospective popularity or success indicators associated with crew and cast members. Data leakage in this context

would occur if our training data inadvertently included information about a crew or cast member's performance or popularity in movies released after the film we are attempting to predict the rating for. This could happen if, for example, traditional k-fold cross validation strategies would be used. Such leakage would provide the model with an unfair advantage, as this future information would not have been available at the time the movie was being made, leading to an artificially inflated assessment of the model's predictive capabilities.

To mitigate this risk, our rolling window validation strategy incorporates time-bound historical information. This ensures that when training on movies released in, say, 2018 to 2020, the cast and crew features are based solely on their performance in films released up to 2017. Similarly, the experience feature is updated by counting only the new movies a crew or cast member was involved in during the preceding years, reflecting their recent activity without peeking into their future projects.

This temporal alignment is crucial. By using these rolling averages and experience counts, we ensure that the cast and crew features in our training data represent the information landscape as it existed at the time the script of the movie being evaluated would have been considered. This guardrail against data leakage is consistently applied across all time-based feature engineering, as evidenced by the 2015 cut-off date implemented for our knowledge graph-based features, which prevents the incorporation of relationship or popularity information that emerged after that specific point in time. This commitment to temporal integrity provides a more realistic and reliable evaluation of our model's ability to predict movie ratings based on information that would have been genuinely available at the relevant time.

4.3 Evaluation implementation

A *Python* function was developed to automate the process of training and evaluating models across multiple folds. The key steps involved in this function are outlined in Algorithm 1 below:

Algorithm 1 Movie Performance Prediction Algorithm

Initialize Variables
Define $cutoff\_years$, $movies\_dataset$
Initialize $training\_mask$, $validation\_mask$, $training\_set$, $validation\_set$
for each $cutoff\_year$ **do**
    $training\_mask \leftarrow$ Select movies released in the last 20 years before $cutoff\_year$
    $validation\_mask \leftarrow$ Select movies released in $cutoff\_year$ and next 2 years
    Apply masks to filter $movies\_dataset$
    Store filtered data in $training\_set$, $validation\_set$
end for
Feature Engineering
Update crew features using rolling dictionaries
Compute $combined\_experience$, $combined\_crew\_power$
Apply value clipping (1%-99%)
Apply Min-Max scaling
Optionally apply PCA
Model Training
if multiple models are provided **then**
    Train each model
    Use Voting Regressor for ensemble learning
else
    Train single model
end if
Prediction and Evaluation
Clip predictions to [0,10]
Evaluate using regression and classification metrics
Analyse results:

- Compute combined results
- Analyse metrics for full screenplays and subtitles
- Identify major absolute errors
- Generate scatter plot

Aggregate weighted results across test years

5 Evaluation

After outlining the experiments implementation plan we will now present the results.

The various components of the training algorithm allow for a wide range of possible combinations. Besides the optimisation of the hyper-parameters of the individual models used in the algorithm, one can also look at different feature selection techniques, embedding models, chunking strategies, arc smoothing techniques and sentiment analysis tools, to name a few. For the evaluation part of this dissertation results are presented for two sizes of fixed size windows for the training set and for each of the embeddings models described in Table 3.2. The calculated metrics for each combination include R-squared (R^2), RMSE , MAE, MAPE and one-away classification for the regression based predictions, and accuracy and F1-score for the categorised movies evaluation section.

While most of the processing and experimentation was carried out on an infrastructure comprising an 11th Gen Intel® Core™ i7-11700F × 16 processor, running on Ubuntu 24.04.2 LTS, having 64.0 GB Ram and an NVIDIA GeForce RTX™ 3060 Ti GPU with 8 GB GDDR6 memory, some of the larger embedding models were run on a notebook system provided by Kaggle *(www.kaggle.com)* having an NVIDIA Tesla P100 GPU with 16GB of memory.

5.1 Analysis of Training and Testing Sets

The analysis has been spread over 10 training and evaluation sets as follows:

Fixed Training Window Size : 20 years		**Fixed Training Window Size : 10 years**	
Training Set	**Testing Set**	**Training Set**	**Testing Set**
1994 - 2014	2015 / 2016 / 2017	2004 - 2014	2015 / 2016 / 2017
1995 - 2015	2016 / 2017 / 2018	2005 - 2015	2016 / 2017 / 2018
1996 - 2016	2017 / 2018 / 2019	2006 - 2016	2017 / 2018 / 2019
1997 - 2017	2018 / 2019 / 2020	2007 - 2017	2018 / 2019 / 2020
1998 - 2018	2019 / 2020 / 2021	2008 - 2018	2019 / 2020 / 2021
1999 - 2019	2020 / 2021 / 2022	2009 - 2019	2020 / 2021 / 2022
2000 - 2020	2021 / 2022 / 2023	2010 - 2020	2021 / 2022 / 2023
2001 - 2021	2022 / 2023 / 2024	2011 - 2021	2022 / 2023 / 2024
2002 - 2022	2023 / 2024	2012 - 2022	2023 / 2024
2003 - 2023	2024	2013 - 2023	2024

Table 5.1 Comparison of Training and Testing Sets for Different Fixed Training Window Sizes

5.2 Metrics

The results are being presented in a tabular format for better visualization. A base model is presented for comparative purposes, wherein the various test datasets are

predicted using the mean rating of the training dataset. Two sets of comparative tables are presented, one which shows the results for the dataset containing crew related features (such as *combined_crew_power* and *combined_crew_experience*), and one without, thereby focusing principally on script related features. The results for the best performing model, SVR are being presented. Other results can be found in Appendix C.

	Regression					Categorisation	
Model	R^2	**RMSE**	**MAE**	**MAPE**	**One-Away**	**Accuracy**	**F1**
Base	-0.0334	1.1244	0.8845	0.1848	0.6693	0.8236	0.7440
Fixed Window: 10							
all-MiniLM-L6-v2	0.4915	0.7875	0.6116	0.1225	0.8289	0.8332	0.7805
gte-modernbert-base	0.4936	0.7858	0.6087	0.1223	0.8264	0.8337	0.7772
UAE-Large-V1	0.4944	0.7847	0.6095	0.1215	0.8296	0.8321	0.7789
ModernBERT-embed-large	0.4979	0.7822	0.6093	0.1219	0.8316	0.8337	0.7812
jasper_en_vision_language_v1	0.5121	0.7709	0.5963	0.1200	0.8369	0.8373	0.7887
stella_en_400m_v5	0.5082	0.7742	0.5986	0.1202	0.8349	0.8371	0.7872
stella_en_1.5B_v5	0.5129	0.7703	0.5956	0.1200	0.8405	0.8362	0.7861
Fixed Window: 20							
all-MiniLM-L6-v2	0.4894	0.7892	0.6133	0.1230	0.8284	0.8339	0.7881
gte-modernbert-base	0.5031	0.7786	0.6034	0.1213	0.8349	0.8350	0.7855
UAE-Large-V1	0.5100	0.7727	0.6003	0.1200	0.8412	0.8326	0.7811
ModernBERT-embed-large	0.5127	0.7708	0.5984	0.1199	0.8366	0.8354	0.7881
jasper_en_vision_language_v1	0.5140	0.7696	0.5944	0.1197	0.8416	0.8362	**0.7929**
stella_en_400m_v5	0.5212	0.7640	0.5909	0.1188	0.8407	0.8350	0.7890
stella_en_1.5B_v5	**0.5255**	**0.7604**	**0.5859**	**0.1183**	**0.8476**	**0.8375**	0.7913

Table 5.2 Performance Metrics using model with *crew_features*

The results in Table 5.2 and Table 5.3 give a strong indication that *ceteris paribus*, the choice of embedding model for the script influences the metrics, with larger models (having more parameters) yielding better scores. Reviewing the results in Table 5.2 using the best performing embedding model, one can note an R^2 of 0.5255. This indicates that approximately 52.55% of the variability in the IMDb ratings is explained by the model. As discussed in Section 3.8.1, movie ratings include a human element and hence, a larger R^2 is not expected. Comparing the MAE and the RMSE one can note a slight disparity in the values, with RMSE slightly higher at 0.7604 compared to the MAE of 0.5859. While the latter provides a measure of the average magnitude of errors (irrespective of direction) and hence can be directly interpreted as the predictions being, on average, 0.5859 units away from the actual IMDb rating, the higher RMSE value suggests that there might be some larger errors that are affecting the RMSE more significantly. To analyse prediction errors, a scatter plot was generated, displaying predicted ratings against actual ratings. The size of each point reflects the magnitude of the absolute error. Figure 5.1 illustrates the model's performance on the validation set, which consists of movies released in 2016-2018, while the model was trained on movies from 1995-2015.

	Regression					Categorisation	
Model	R^2	**RMSE**	**MAE**	**MAPE**	**One-Away**	**Accuracy**	**F1**
Base	-0.0334	1.1244	0.8845	0.1848	0.6693	0.8236	0.7440
Fixed Window: 10							
all-MiniLM-L6-v2	0.4146	0.8446	0.6581	0.1301	0.8000	0.8295	0.7715
gte-modernbert-base	0.4229	0.8385	0.6525	0.1302	0.8015	0.8308	0.7753
UAE-Large-V1	0.4241	0.8374	0.6527	0.1289	0.8051	0.8291	0.7764
ModernBERT-embed-large	0.4321	0.8316	0.6481	0.1287	0.8046	0.8309	0.7783
jasper_en_vision_language_v1	0.4476	0.8200	0.6352	0.1266	0.8170	0.8353	0.7859
stella_en_400m_v5	0.4510	0.8175	0.6334	0.1265	0.8164	0.8324	0.7819
stella_en_1.5B_v5	**0.4588**	**0.8115**	**0.6278**	**0.1259**	**0.8212**	**0.8391**	**0.7906**
Fixed Window: 20							
all-MiniLM-L6-v2	0.4242	0.8379	0.6538	0.1291	0.8002	0.8289	0.7750
gte-modernbert-base	0.4283	0.8345	0.6487	0.1293	0.8046	0.8310	0.7780
UAE-Large-V1	0.4375	0.8274	0.6444	0.1273	0.8098	0.8304	0.7780
ModernBERT-embed-large	0.4426	0.8240	0.6407	0.1271	0.8100	0.8310	0.7810
jasper_en_vision_language_v1	0.4566	0.8135	0.6292	0.1254	0.8184	0.8337	0.7858
stella_en_400m_v5	0.4600	0.8108	0.6287	0.1253	0.8225	0.8318	0.7842
stella_en_1.5B_v5	**0.4681**	**0.8046**	**0.6216**	**0.1248**	**0.8235**	**0.8389**	**0.7932**

Table 5.3 Performance Metrics using model without *crew_features*

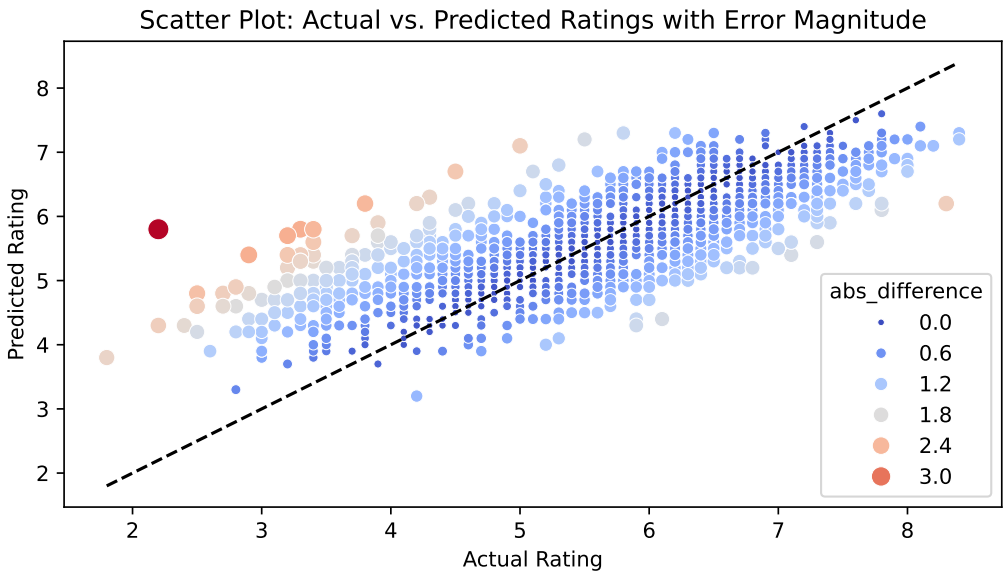


Figure 5.1 Scatter Plot - Actual vs Predicted Ratings - Cut-off year 2015. Image created by author

The model struggled to accurately predict ratings at the extreme ends (3 and below, 8 and above) due to the data's inherent imbalance, where these ratings were significantly less common, as evidenced by the larger error markers in the scatter plot. This resulted in higher errors for these ratings, with the consequential higher value for RMSE.

The MAPE of 0.1183 or 11.83% means that on average, the model's predictions deviate from the actual IMDb ratings by 11.83%. The one-away accuracy metric indicates that the model was, on average (across all training folds), able to predict the

IMDb rating within one point of the actual rating for 84.76% of the movies in the validation sets.

For the classification exercise, ratings were binned into 3 categories, [0-4], [4.1-7], [7.1-10]. The model managed to correctly place into the correct bin 83.75% of the movies in the validation sets, while achieving an F1-score of 0.7913. Having an imbalanced dataset, the F1-score was of particular importance as it provides a balanced measure of the model's performance. The result indicates that the model is both good at correctly identifying movies within each rating bin, defined by *precision*, as well as capturing a high proportion of the actual movies within each bin, defined by *recall*. Of particular interest is the fact that for the classification metrics, predictions from both models (with or without crew related features) were more or less equal.

Within each training and validation fold, a plot of residual errors is also output. Figure 5.2 shows the residual error distribution plot for the fold relating to the model trained on movies from 1995-2015 and validated on movies released in 2016-2018. Residuals represent the difference between actual and predicted values from the regression model.

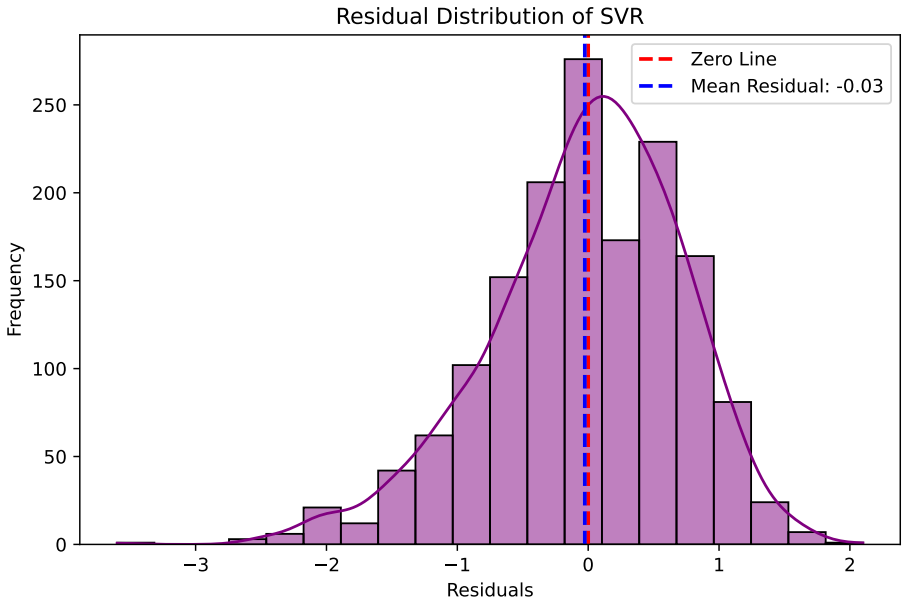


Figure 5.2 Residual Errors Distribution Plot - Cut-off year 2015. Image created by author

With a *p* value less than 0.05 obtained using the *Shapiro* Test, the distribution of the residuals is not deemed to be normally distributed. With skewness of -0.571 and kurtosis of 0.515 the distribution is quite symmetrical with a longer left tail (a number under predicted titles) and reasonably normal. The mean of the residuals is close to zero, indicating that the model is not systematically over or under-predicting the target variable.

5.2.1 Dissecting the Metrics

An in-depth analysis of the validation dataset composition was conducted to inspect whether the same metrics were achieved for full screenplays *versus* subtitles. Each validation dataset was analysed into metric results obtained for each category of movie scripts. Table 5.4 illustrates the results for the overall metrics as well as those for each category, for the best performing model with a training time window of 20 years. The number of individual movies in the training sets and validation sets is also provided.

	Number of Movies		Evaluation Metrics				
Time-frame	**Training Set**	**Testing Set**	R^2	**RMSE**	**MAE**	**MAPE**	**One-Away**
1994 - 2014	7965	1530	0.5446	0.7558	0.5907	0.1196	0.8497
Full Screenplays		163	0.2523	**0.6685**	**0.5350**	**0.0802**	**0.8834**
Subtitles		1367	**0.4914**	0.7656	0.5974	0.1243	0.8456
1995 - 2015	8242	1562	0.5567	0.7414	0.5782	0.1170	0.8528
Full Screenplays		162	0.2465	**0.7267**	**0.5617**	**0.0889**	**0.8642**
Subtitles		1400	**0.5188**	0.7431	0.5801	0.1203	0.8514
1996 - 2016	8535	1454	0.5598	0.7643	0.5876	0.1222	0.8514
Full Screenplays		158	0.2834	**0.7227**	**0.5544**	**0.0879**	**0.8734**
Subtitles		1296	**0.5150**	0.7692	0.5916	0.1264	0.8488
1997 - 2017	8789	1134	0.5422	0.7654	0.5804	0.1194	0.8483
Full Screenplays		131	0.2696	**0.7111**	**0.5275**	**0.0854**	**0.8931**
Subtitles		1003	**0.5007**	0.7722	0.5873	0.1239	0.8425
1998 - 2018	9059	936	0.5303	0.7764	0.5974	0.1206	0.8462
Full Screenplays		109	0.2880	**0.5882**	**0.4541**	**0.0682**	**0.9450**
Subtitles		827	**0.4714**	0.7979	0.6163	0.1275	0.8331
1999 - 2019	9231	992	0.5038	0.7639	0.5956	0.1172	0.8458
Full Screenplays		110	0.2167	**0.5785**	**0.4536**	**0.0691**	**0.9545**
Subtitles		882	**0.4482**	0.7839	0.6133	0.1232	0.8322
2000 - 2020	9162	1223	0.5174	0.7459	0.5738	0.1135	0.8545
Full Screenplays		111	0.3789	**0.6018**	**0.4685**	**0.0749**	**0.9550**
Subtitles		1112	**0.4695**	0.7588	0.5844	0.1173	0.8444
2001 - 2021	9207	1091	0.4913	0.7659	0.5822	0.1171	0.8387
Full Screenplays		83	0.4063	**0.6411**	**0.5072**	**0.0831**	**0.9277**
Subtitles		1008	**0.4498**	0.7753	0.5884	0.1199	0.8313
2002 - 2022	9405	607	0.4487	0.7665	0.5779	0.1131	0.8402
Full Screenplays		39	**0.4888**	**0.7040**	**0.5462**	**0.0968**	**0.8974**
Subtitles		568	0.4134	0.7706	0.5801	0.1143	0.8363
2003 - 2023	9513	190	0.3392	0.8399	0.6458	0.1287	0.8000
Full Screenplays		6	-0.0918	**0.7472**	**0.6167**	**0.1071**	**0.8333**
Subtitles		184	**0.3383**	0.8427	0.6467	0.1294	0.7989

Table 5.4 Metric Results - Categorised per Type of Script

The results are summarized in Table 5.5. Averages and Standard Deviation were calculated on a weighted basis. Of particular interest was the fact that with the exception of the R^2 metric, all other metrics were significantly better for full screenplays when compared with those of subtitles. Values indicate that predictions for full screenplays tend to have lower absolute errors. At 90.48%, predictions for full screenplays are also more probable to be within one unit of the actual rating.

Observations from the standard deviations indicate that while having lower absolute errors, predictions on full screenplays have the most inconsistent performance. This could be the result of the smaller proportion in both training and validation sets.

	Mean					Standard Deviation				
Type	R^2	**RMSE**	**MAE**	**MAPE**	**One-Away**	R^2	**RMSE**	**MAE**	**MAPE**	**One-Away**
Overall	0.5255	0.7604	0.5859	0.1183	0.8476	0.0388	0.0150	0.0110	0.0031	0.0080
Full Screenplays	0.2898	0.6658	0.5162	0.0814	0.9048	0.0723	0.0570	0.0415	0.0082	0.0351
Subtitles	0.4795	0.7699	0.5936	0.1224	0.8412	0.0360	0.0176	0.0138	0.0038	0.0093

Table 5.5 Mean and Standard Deviation of Metrics per Type of Script

A visualisation of the metrics over the respective validation time frames is shown in Figure 5.3. The volatility of the full screenplay predictions can be clearly noticed, where predictions on subtitle based scripts tend to be more stable over the validation time-frames.

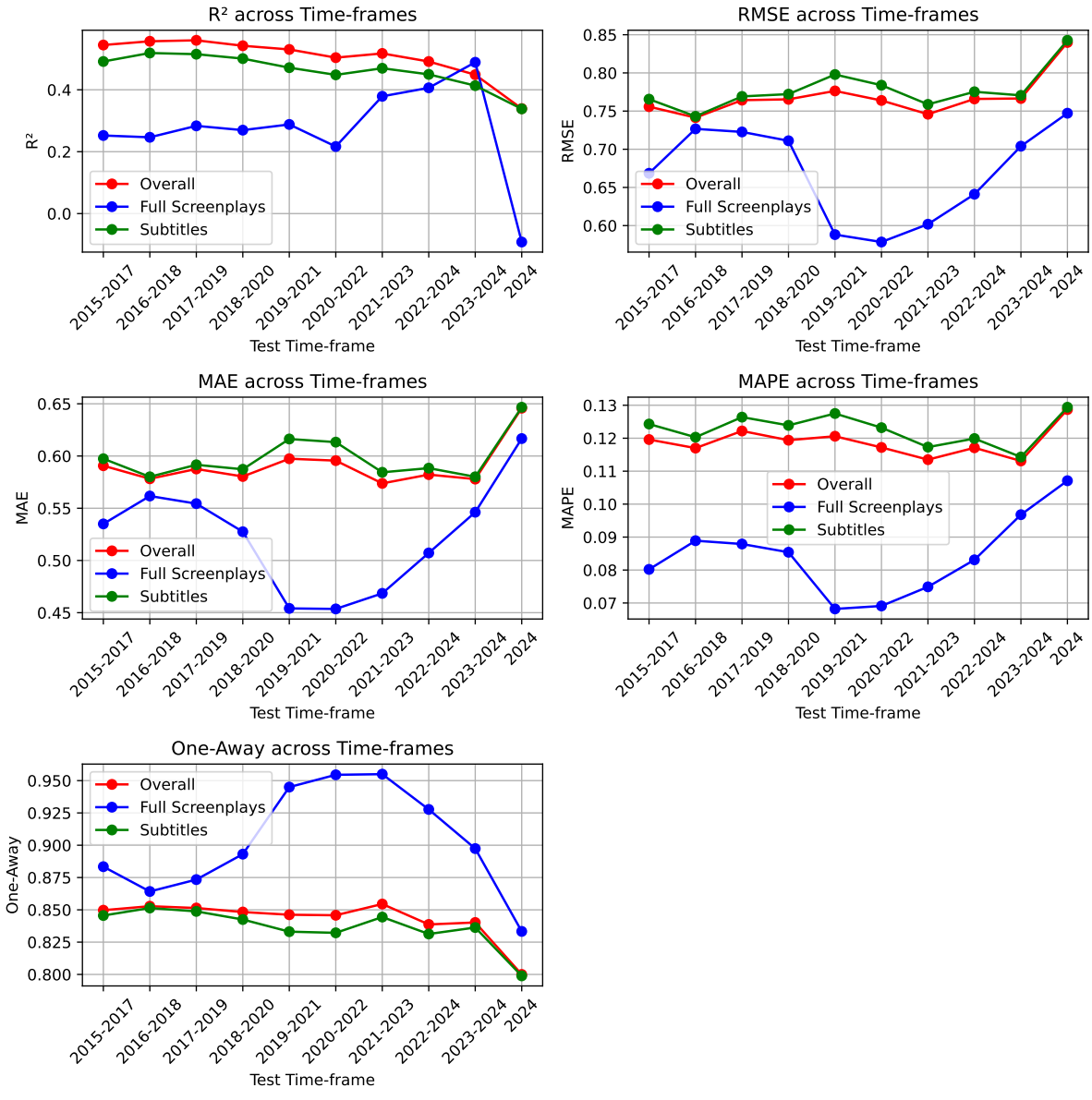


Figure 5.3 Metrics across time frames. Image created by author

5.2.2 Analysing Genres

The validation datasets contained 25 unique genres. For the purposes of this analysis, the results have been filtered to retain those genres represented by more than 500 count instances in the combined validation datasets. This reduced the genres to 13 unique categories. The average MAE across all movies in the validation datasets was 0.5859.

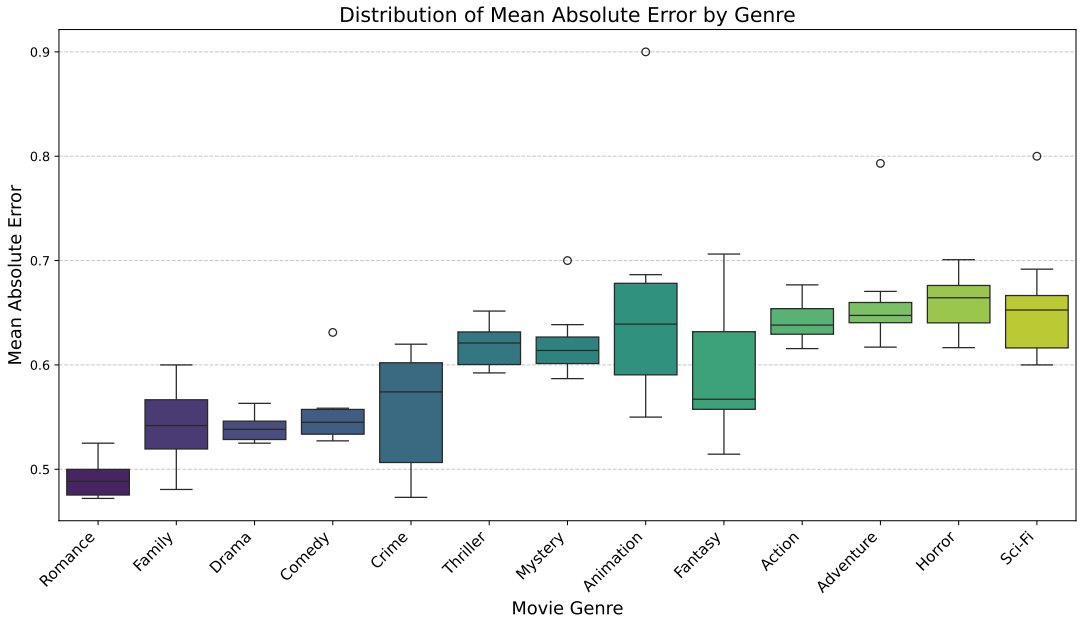


Figure 5.4 Boxplot of Average MAE per Genre. Image created by author

As illustrated in Figure 5.4, the model's performance varies significantly across genres. *Romance* films show the lowest average MAE at 0.4909, suggesting they are the best predicted. This is followed closely by the *Drama* genre with an average MAE of 0.5389. *Comedy*, *Family* and *Crime* suggest decent prediction accuracy with MAE values below the overall average. In contrast, *Adventure*, *Horror*, *Sci-Fi* and *Animation* films exhibit the highest average MAE ranging from 0.65 to 0.66, indicating they are harder to predict. With a standard deviation of 0.0121, *Drama* emerges as the genre whose MAE was most stably predicted, followed by *Romance* with a standard deviation of 0.0178 and *Action* and *Thriller*, indicating a fairly stable performance. On the other hand, *Animation* has the highest variability with a standard deviation of 0.0986.

This analysis indicates that some genres are inherently more predictable than others, with *Romance* and *Drama* being the most predictable and *Action*, *Adventure*, *Sci-Fi*, *Horror* and *Animation* being the more challenging. For genres like *Animation*, where the MAE varies significantly, it would be beneficial to investigate the specific characteristics of the datasets that lead to very high or very low MAEs. This could reveal factors that influence model performance within those genres.

5.2.3 Analysing Emotional Arcs

The clustering exercise carried out in Section 3.3 produced 6 distinct clusters, which were mapped to the respective movie.

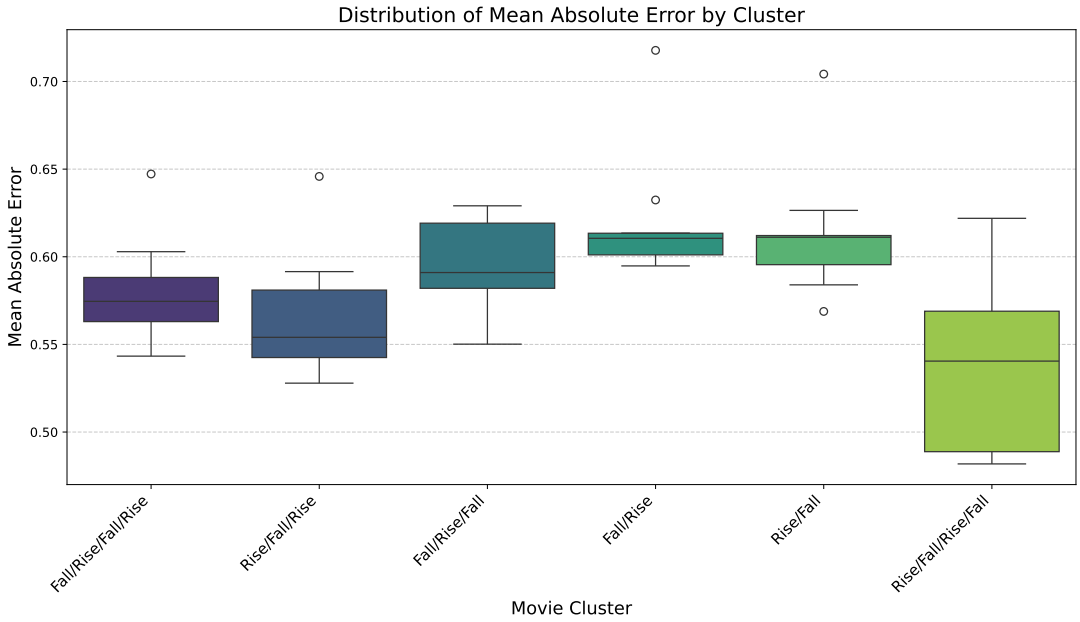


Figure 5.5 Boxplot of Average MAE per Cluster. Image created by author

Figure 5.5 reveals that the model's performance also shows considerable variation depending on the emotional trajectory cluster of the film. Movies with a *Rise/Fall/Rise/Fall* trajectory exhibit the lowest average MAE at 0.5380, indicating they are the most accurately predicted. This is closely followed by the *Rise/Fall/Rise* cluster with an average MAE of 0.5661, and the *Fall/Rise/Fall/Rise* cluster with 0.5797, all suggesting good prediction accuracy. Conversely, simpler emotional arcs like *Fall/Rise* and *Rise/Fall* presented more challenges for the model, displaying higher average MAE values of 0.6194 and 0.6125 respectively, suggesting they are harder to predict.

When examining consistency in predictions, the *Fall/Rise/Fall* cluster demonstrates the most stable predictions with the lowest standard deviation of 0.0271, followed closely by *Fall/Rise/Fall/Rise* with a standard deviation of 0.0292. In contrast, despite having the lowest average MAE, the *Rise/Fall/Rise/Fall* cluster shows the highest variability in its predictions, with a standard deviation of 0.0514. This suggests that while it performs best on average, its MAE can fluctuate more significantly across different datasets.

This analysis indicates that the complexity and specific patterns of emotional trajectories within films play a significant role in their predictability. For clusters like *Rise/Fall/Rise/Fall*, where the MAE shows higher variability, further investigation into the characteristics of the datasets leading to particularly high or low MAEs could uncover

factors influencing model performance within these emotional arcs.

5.3 Comparative Analysis

A proper comparative analysis of the results could not be adequately performed due to the heterogeneity in the datasets. In addition, very few papers focused on the prediction of IMDb ratings using similar features with complementary regression related metrics. Nonetheless a review of some similar metric results was conducted.

The evaluation of the model presented in this dissertation reveals its competitive edge over existing methodologies in movie rating prediction. In comparison to Ning et al.'s work [197], which proposed a generative convolutional neural network leveraging pre-production movie features like genres, budget, cast, director, and plot information, this model demonstrates improved performance. Their model, trained on 1219 movies released between 2003 and 2013 and tested on 252 movies released after 2013, achieved a Mean Squared Error (MSE) of 0.628 and a one-away accuracy of 85.3%. In contrast, this SVR model with a fixed 20-year window achieved a lower MSE of 0.5782 (derived from an RMSE of 0.7604) and a slightly lower one-away accuracy of 84.76%.

Similarly, Abarja et al. [85] employed a CNN on a larger dataset of 3,819 training movies (2000-2013) and 498 testing movies (after 2013), incorporating metadata, historical, topical, and categorical features, but reported a higher MSE of 1.20 and a Mean Absolute Error (MAE) of 0.83. Furthermore, our model's MAE of 0.5859 significantly outperforms the MAE (referred to as Prediction Absolute Error, PAE) of 0.6973 obtained by Hsu [84] in a study using IMDb data without a strict chronological split between training and testing sets, highlighting the importance of the temporal validation approach used in this thesis.

A dissertation based on Kaggle's IMDB Movies Dataset achieved an R^2 of 0.335 and an RMSE of 0.859 using a random forest algorithm [198]. Finally, Zabaleta's dissertation [199], which utilized features such as release year, runtime, 22 genres, 70 regions, and over 3000 artists, reported an MAE of 0.684, again higher than the MAE achieved by our model. These comparisons highlight the effectiveness of this thesis' regression model in predicting movie ratings.

This dissertation advanced movie audience rating prediction by developing new data and script-driven features and a robust, time-aware validation strategy. These innovations, which included analysing crew/cast relationships, leveraging diverse script embedding models, and characterising emotional arcs, led to improved prediction performance compared to existing methods.

6 Conclusions and Future Work

In this final chapter the aims and objectives of this dissertation are revisited, with the evaluation of results achieved leading to an analysis of the limitations encountered. Suggestions are made for possible future work in this area and a conclusion on the potential impact the work carried out might have in the industry is presented.

6.1 Revisiting the Aims and Objectives

This dissertation sought to develop a model capable of predicting movie audience ratings by analysing script features, acknowledging the inherent complexity introduced by potentially unmeasurable human and audience factors influencing a film's reception. To achieve this, a comprehensive literature review was conducted, a suitable dataset was compiled, and various models were trained, optimised, and evaluated, ultimately aiming to surpass basic average rating prediction methods and provide a more generalized and accurate predictive system.

6.2 Limitations and Future Work

6.2.1 Data and Feature Limitations

Building a model that aims to capture audience emotions and evaluation of a movie presented a number of challenges, predominantly related to the human element involved in the analysis. The work tried to capture and evaluate features existing at a specific point in time during the lifetime of a movie, specifically that which precedes major capital outflow by the investors. Numerous features influence the quality of the final product and other factors being part of the shooting session or post production, such as the visual effects, musical scores and marketing, have not been included in this analysis.

While the results of our regression analysis has revealed statistically significant associations between some of the input features and the IMDb user ratings, it is important to emphasise that these observed relationships should be interpreted as correlations and not as evidence of direct causal effects. The absence of key variables such as marketing budget and distribution strategy is a significant limitation when considering the potential causal effects of such variables. These factors are widely acknowledged to have a substantial impact on the movie's visibility and reach, thereby influencing its IMDb rating.

For a producer, understanding why a script might succeed, determined by causality, is arguably more valuable than just knowing how the extracted features correlate with successful productions. Future work in this area could employ different modeling techniques to explore potential causal links by incorporating data on marketing expenditure, distribution scope and critics reception. Such an approach, combined with the insights presented in this dissertation, could evolve the model from a correlational ranking tool into a more robust decision-support system, thereby fulfilling the initial aim of this dissertation, to aid such producers in their investment decisions.

6.2.2 Script Analysis and NLP Challenges

Evidence suggests the script, along with descriptive features like genre, significantly contributes to how these features correlate with audience perception of a movie. Nonetheless, while advancements in NLP techniques and tools have facilitated the extraction of meaningful features from scripts, certain limitations have to be reported. Current embedding models have a maximum token input restriction, which requires text to be split into chunks, losing contextuality. Even with larger input token limits, compressing the whole script into a few thousand dimension vectors causes information loss, as well as potentially introducing noise where embedded vector dimensions are larger than 1024. Newer, larger and more advanced models are being released on a more frequent basis and with the right infrastructure, a better vector representation can be achieved.

Embedding models are also trained on specific general datasets. As with large language models, embedding models can be fine tuned for a specific task. The *all_miniLM-L6_v2* [167], which is one of the smaller models used in this disseration, was trained on a large and diverse dataset of over 1 billion training pairs [167]. Fine-tuning models would require substantial computer power and infrastructure, but such a process could potentially result in a specific model for embedding movies, the clustering of which would be more adept at finding similarly rated movies.

While becoming more sophisticated, embedding models can still miss subtleties such as the pacing of the script, humour or audience sentiment. Furthermore, embeddings are capturing the textual element of the story. At this stage there is no visual or auditory material that can be analysed. However, with improvements in multi-modal models such analysis can be introduced during shooting to analyse scenes as they are shot. At an earlier stage these models can be applied to mood boards or even soundtrack samples. With models such as Sora [200], story boards can already be generated at lightning speed and with minimal cost simply from a textual input of the scene.

The availability of full screenplays for academic research was restricted. Results

have indicated that while being more volatile, predictions on full screenplays outperform those on subtitle based screenplays. The added contextual scene information found in full screenplays enables a richer interpretation and understanding of specific movie features by models and having a larger database of full screenplays would provide researchers with the possibility to reduce this volatility and train a better generalised model.

6.2.3 Crew-Related Features and Network Analysis

In this dissertation, crew related features have been used with a level of marked improvement in results when compared to the model that does not include such features. This may be used in future work for the development of a model that allows for *what-if* analysis. Given a set of actors, musical composers and directors, the rating of a movie could be predicted by feeding in a different combination of crew members and assess the model's capability to infer the importance of certain crew members given a pre-determined set of features.

Emphasizing on the discussion presented in Section 6.2.1, given that the inclusion of crew-related features in the dataset resulted in better evaluation metrics, it is again important to point out that this model does not definitely prove their *causal* impact on ratings.

While a brief introduction has been provided on the use of graph networks within the aims of this work, such networks and related measures can be further utilised to extract more meaningful relationships not only between crew members themselves but also in relation to other categorical features, such as genre. This combination would also assist producers identify crew members that would be best suited for a particular movie.

6.2.4 Model Optimisation and Evaluation Challenges

As a large number of features were used to build the model, the resulting dataset, with over 1200 dimensions and 10 different training folds, provided difficulty in training multiple versions. Hyper-parameter optimisation of models is a key factor in achieving better results from a modified model. While a limited use of Optuna [201] has been employed to obtain the final hyperparameters of the models, using this tool together with others such as GridSearch [202] may reveal more optimal parameters for the tested models, hence achieving better results. This was not possible considering the time and resources allocated.

In this analysis, the target feature was based on one indicator for audience reception. While being a robust and respected indicator, it is not the only source of audience perception for movies. Further work could incorporate other sources and

forms of audience ratings and averaged to produce a more robust indicator.

The structuring of the training and evaluation sets, with the updating of training sets with each forward transition, provides a realistic method for generalising and keeping the model relevant to the changing times. While different training windows were tested (10 years and 20 years) this choice of window was arbitrary and further testing could be performed to identify optimal training windows. Audience preferences may drift over time and the model has to be kept updated with these drifts.

This study, like many others, has not considered the effect of the change in release platforms. While traditional releases were restricted to cinemas and portable media, streaming has proliferated a number of titles to a much wider audience, which may affect visibility and rating distributions. Some movies are also being released exclusively on streaming platforms. This factor may in itself be a subject for further study.

Finally, as with hyperparameter optimisation, feature selection is also an important element of building an optimal and noise free model. With the number of features and the available infrastructure, only limited feature selection was conducted.

6.3 Concluding Remarks

In this research a custom dataset was built from movie scripts and subtitles. Features were extracted using linguistic analysis tools like LIWC-22, VADER for the emotional trajectory of movies, and NRC Emotion Lexicons for basic emotions and intensity. Script embeddings were generated using various state of the art models of different sizes, selected using the MTEB leaderboard, and combined with extracted features. Regression models, including gradient boosting machines, linear models, neural networks, and SVR, were tested. Model performance was evaluated using R^2, MAPE, RMSE, and MAE for regression, and one-away score, accuracy, and F1-score for classification metrics, providing a comprehensive assessment of prediction accuracy and robustness. The best model achieved on average an R^2 of 0.5255, a MAPE of 0.1183, an RMSE of 0.7604, and a MAE of 0.5859 for regression metrics, and a one-away score of 0.8476, accuracy of 0.8375, and an F1-score of 0.7913 for the classification section.

References

[1] Statista. "Cinema worldwide," Accessed: Dec. 17, 2024. [Online]. Available: https://www.statista.com/outlook/amo/media/cinema/worldwide.

[2] IMdB. "Imdb," Accessed: Dec. 17, 2024. [Online]. Available: https://www.imdb.com/.

[3] "Screenwriters association," Accessed: Jan. 7, 2025. [Online]. Available: https://www.swaindia.org/.

[4] "International screenwriters association," Accessed: Jan. 7, 2025. [Online]. Available: https://www.networkisa.org/.

[5] J. Eliashberg and S. K. a. Z. Hui, "Assessing box office performance using movie scripts: A kernel-based approach," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 11, pp. 2639–2648, 2014.

[6] W. Goldman, *Adventures in the Screen Trade: A Personal View of Hollywood and Screenwriting*. Abacus, 1996, ISBN: 9780349107059. [Online]. Available: https://books.google.com.mt/books?id=uofAAQAACAAJ.

[7] R. Caves, *Creative Industries: Contracts Between Art and Commerce* (Nova et vetera iuris gentium: Modern international law). Harvard University Press, 2000, ISBN: 9780674001640. [Online]. Available: https://books.google.com.mt/books?id=imfTUHj8uVcC.

[8] T. Numbers. "Movie index," Accessed: Jan. 9, 2025. [Online]. Available: https://www.the-numbers.com/movies/#tab=year.

[9] R. L. Boyd, A. Ashokkumar, S. Seraj, and J. W. Pennebaker, "The development and psychometric properties of liwc-22.," 2022. Accessed: Jan. 7, 2025. [Online]. Available: https://www.liwc.app.

[10] C. Hutto and E. Gilbert, "Vader: A parsimonious rule-based model for sentiment analysis of social media text," presented at the Eighth International Conference on Weblogs and Social Media (ICWSM-14) (Ann Arbor, MI, Jun. 2014), Jun. 2014. Accessed: Jan. 7, 2025. [Online]. Available: https://vadersentiment.readthedocs.io/en/latest/pages/introduction.html.

[11] S. Mohammad and P. Turney, "Crowdsourcing a word-emotion association lexicon," *Computational Intelligence*, vol. 29, pp. 436–465, 3 2013. Accessed: Jan. 7, 2025. [Online]. Available: https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm.

[12] S. M. Mohammad, "Word affect intensities," presented at the Proceedings of the 11th Edition of the Language Resources and Evaluation Conference (LREC-2018) (May 2018), Miyazaki, Japan. Accessed: Jan. 7, 2025. [Online]. Available: https://saifmohammad.com/WebPages/AffectIntensity.htm.

[13] N. Muennighoff, N. Tazi, L. Magne, and N. Reimers, "Mteb: Massive text embedding benchmark," *arXiv preprint arXiv:2210.07316*, 2022. DOI: 10.48550/ARXIV.2210.07316. Accessed: Jan. 7, 2025. [Online]. Available: https://arxiv.org/abs/2210.07316.

[14] M. T. E. Benchmark. "Embedding leaderboard," Accessed: Jan. 7, 2025. [Online]. Available: https://huggingface.co/spaces/mteb/leaderboard.

[15] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16, San Francisco, California, USA: ACM, 2016, pp. 785–794, ISBN: 978-1-4503-4232-2. DOI: 10.1145/2939672.2939785. [Online]. Available: http://doi.acm.org/10.1145/2939672.2939785.

[16] G. Ke et al., "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems*, vol. 30, pp. 3146–3154, 2017.

[17] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[18] V. Vapnik, S. Golowich, and A. Smola, "Support vector method for function approximation, regression estimation and signal processing," in *Advances in Neural Information Processing Systems*, M. Mozer, M. Jordan, and T. Petsche, Eds., vol. 9, MIT Press, 1996. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/1996/file/4f284803bd0966cc24fa8683a34afc6e-Paper.pdf.

[19] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, *Catboost: Unbiased boosting with categorical features*, 2019. arXiv: 1706.09516 [cs.LG]. [Online]. Available: https://arxiv.org/abs/1706.09516.

[20] T. H. Davenport and J. G. Harris, "What people want (and how to predict it)," *MIT Sloan Management Review*, vol. 50, no. 2, 2009.

[21] W. Walls and A. De Vany, "Motion picture profit, the stable paretian hypothesis, and the curse of the superstar," *Journal of Economic Dynamics and Control*, vol. 28, pp. 1035–1057, Mar. 2004. DOI: 10.1016/S0165-1889(03)00065-4.

[22] B. Gunter, *Predicting Movie Success at the Box Office*. Cham: Springer International Publishing AG, 2018. DOI: 10.1007/978-3-319-71803-3.

[23] D. Trottier, “The screenwriter’s bible; a complete guide to writing, formatting, and selling your script, 4th ed,” *Reference and Research Book News*, vol. 20, no. 4, 2005.

[24] J. Eliashberg, J.-J. Jonker, M. S. Sawhney, and B. Wierenga, “Moviemod: An implementable decision-support system for prerelease market evaluation of motion pictures,” *Marketing science (Providence, R.I.)*, vol. 19, no. 3, pp. 226–243, 2000. DOI: 10.1287/mksc.19.3.226.11796.

[25] R. Neelamegham and P. Chintagunta, “A bayesian model to forecast new product performance in domestic and international markets,” *Marketing science (Providence, R.I.)*, vol. 18, no. 2, pp. 115–136, 1999. DOI: 10.1287/mksc.18.2.115.

[26] Z. Feng, *Formal Analysis for Natural Language Processing: A Handbook*. Singapore: Springer Nature Singapore, 2023. DOI: 10.1007/978-981-16-5172-4.

[27] C. Styker and J. Holdsworth. “What is nlp?” Accessed: Mar. 17, 2025. [Online]. Available: https://www.ibm.com/think/topics/natural-language-processing.

[28] F. B. Insights. “Big data analytics market size, share & industry analysis, by component (software, hardware, and services), by enterprise type (large enterprises and small & medium enterprises (smes)), by application (data discovery and visualization, advanced analytics, and others), by vertical (bfsi, automotive, telecom/media, healthcare, life sciences, retail, energy & utility, government, and others), and regional forecast, 2024-2032,” Accessed: Mar. 17, 2025. [Online]. Available: https://www.fortunebusinessinsights.com/big-data-analytics-market-106179.

[29] A. Jonker and A. Gomstyn. “Structured vs. unstructured data: What’s the difference?” Accessed: Mar. 17, 2025. [Online]. Available: https://www.ibm.com/think/topics/structured-vs-unstructured-data.

[30] 7wData. “Natural language processing: Taking your business to the next level,” Accessed: Mar. 17, 2025. [Online]. Available: https://7wdata.be/article-general/natural-language-processing-taking-your-business-to-the-next-level/.

[31] P. Nandwani and R. Verma, “A review on sentiment analysis and emotion detection from text,” *Social network analysis and mining*, vol. 11, no. 1, p. 81, 2021. DOI: 10.1007/s13278-021-00776-6.

[32] B. Liu, *Sentiment analysis and opinion mining*. Springer Nature, 2022.

[33] S. J. Dixon. “Social media - statistics and facts.” Statista, Ed., Accessed: Mar. 18, 2025. [Online]. Available: https://www.statista.com/topics/1164/social-networks/#topicOverview.

[34] Domo. “Data never sleeps 12.0,” Accessed: Mar. 18, 2025. [Online]. Available: https://www.domo.com/learn/infographic/data-never-sleeps-12.

[35] R. S. T. Lee, *Natural Language Processing: A Textbook with Python Implementation*, 1st ed. Singapore: Springer, 2023, ISBN: 9789819919987. DOI: 10.1007/978-981-99-1999-4.

[36] S. Baccianella and F. Esuli Andrea and Sebastiani, “Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining,” in *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC‘10)*, Valletta, Malta: European Language Resources Association (ELRA), May 2010. [Online]. Available: https://aclanthology.org/L10-1531/.

[37] Turing. “A guide on word embeddings in nlp,” Accessed: Feb. 7, 2025. [Online]. Available: https://www.turing.com/kb/guide-on-word-embeddings-in-nlp.

[38] unknown. [Online]. Available: https://miro.medium.com/v2/resize:fit:1838/1*gcC7b_v7OKWutYN1NAHyMQ.png.

[39] Scikit-learn. “Tfidfvectorizer,” Accessed: Feb. 6, 2025. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html.

[40] Scikit-learn. “Countvectorizer,” Accessed: Feb. 6, 2025. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html.

[41] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162.

[42] Manikanth. “Understanding contextualized word embeddings: The evolution of language understanding in ai,” Accessed: Feb. 7, 2025. [Online]. Available: https://manikanthgoud123.medium.com/understanding-contextualized-word-embeddings-the-evolution-of-language-understanding-in-ai-8bf79a98eb51.

[43] J. Devlin, C. Ming-Wei, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *CoRR*, vol. abs/1810.04805, 2018. [Online]. Available: http://arxiv.org/abs/1810.04805.

[44] A. Vaswani et al., "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017. arXiv: 1706.03762. [Online]. Available: http://arxiv.org/abs/1706.03762.

[45] C. Lee et al., *Nv-embed: Improved techniques for training llms as generalist embedding models*, 2025. arXiv: 2405.17428 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2405.17428.

[46] G. de Souza P. Moreira, R. Osmulski, M. Xu, R. Ak, B. Schifferer, and E. Oldridge, *Nv-retriever: Improving text embedding models with effective hard-negative mining*, 2024. arXiv: 2407.15831 [cs.IR]. [Online]. Available: https://arxiv.org/abs/2407.15831.

[47] S. Xiao, Z. Liu, P. Zhang, N. Muennighoff, D. Lian, and J.-Y. Nie, *C-pack: Packed resources for general chinese embeddings*, 2024. arXiv: 2309.07597 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2309.07597.

[48] C. Li et al., *Making text embedders few-shot learners*, 2024. arXiv: 2409.15700 [cs.IR]. [Online]. Available: https://arxiv.org/abs/2409.15700.

[49] Z. Dun, L. Jiacheng, Z. Ziyang, and W. Fulong, *Jasper and stella: Distillation of sota embedding models*, 2025. arXiv: 2412.19048 [cs.IR]. [Online]. Available: https://arxiv.org/abs/2412.19048.

[50] S. Follows. "Defining the average screenplay, via data on 12,000+ scripts," Accessed: Feb. 9, 2025. [Online]. Available: https://stephenfollows.com/p/what-the-average-screenplay-contains.

[51] "Openai token calculator," Accessed: Feb. 9, 2025. [Online]. Available: https://www.gptcostcalculator.com/open-ai-token-calculator.

[52] M. Günther, I. Mohr, D. J. Williams, B. Wang, and H. Xiao, *Late chunking: Contextual chunk embeddings using long-context embedding models*, 2024. arXiv: 2409.04701 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2409.04701.

[53] R. Khatun and A. Sarkar, "Deep-keywordnet: Automated english keyword extraction in documents using deep keyword network based ranking," *Multimedia Tools and Applications*, vol. 83, no. 27, pp. 68 959–68 991, 2024. DOI: 10.1007/s11042-024-18110-5.

[54] M. Martinc, B. Škrlj, and S. Pollak, "Tnt-kid: Transformer-based neural tagger for keyword identification," *Natural language engineering*, vol. 28, no. 4, pp. 409–448, 2022. DOI: 10.1017/S1351324921000127.

[55] S. K. Bharti and S. B. Korra, "Automatic keyword extraction for text summarization: A survey," *arXiv.org*, 2017.

[56] M. Litvak, M. Last, and A. Kandel, “Degext: A language-independent keyphrase extractor,” *Journal of ambient intelligence and humanized computing*, vol. 4, no. 3, pp. 377–387, 2013. DOI: 10.1007/s12652-012-0109-z.

[57] S. Beliga, A. Meštrović, and S. Martin□ić-Ipšić, “An overview of graph-based keyword extraction methods and approaches,” *Journal of information and organizational sciences*, vol. 39, no. 1, pp. 1–20, 2015.

[58] B. V. Barde and A. M. Bainwad, “An overview of topic modeling methods and tools,” in *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, IEEE, 2017, pp. 745–750.

[59] Sinhasagar. “The keyword quest: Techniques and challenges,” Accessed: Mar. 21, 2025. [Online]. Available: https://medium.com/accredian/the-keyword-quest-techniques-and-challenges-a06f6fcb7667.

[60] M. Grootendorst, *Keybert: Minimal keyword extraction with bert*. Version v0.3.0, 2020. DOI: 10.5281/zenodo.4461265. [Online]. Available: https://doi.org/10.5281/zenodo.4461265.

[61] A. Sukumaran. “Unraveling cinematic connections: Analyzing movie people relationships with spanner graph!” Accessed: Mar. 21, 2025. [Online]. Available: https://medium.com/google-cloud/unraveling-cinematic-connections-analyzing-movie-people-relationships-with-spanner-graph-47ccc50b07ec.

[62] A. Hogan et al., “Knowledge graphs,” *ACM computing surveys*, vol. 54, no. 4, pp. 1–37, 2021. DOI: 10.1145/3447772.

[63] T. Christiano Silva and L. Zhao, *Machine Learning in Complex Networks*, 1st ed. Cham: Springer International Publishing, 2016, ISBN: 3-319-17290-5. DOI: 10.1007/978-3-319-17290-3.

[64] Turing. “Graph centrality measures,” Accessed: Mar. 21, 2025. [Online]. Available: https://www.turing.com/kb/graph-centrality-measures.

[65] O. Tanglay, N. Dadario, E. Chong, S. Tang, I. Young, and M. Sughrue, “Graph theory measures and their application to neurosurgical eloquence,” *Cancers*, vol. 15, p. 556, Jan. 2023. DOI: 10.3390/cancers15020556.

[66] IBM. “What is machine learning?” Accessed: Mar. 22, 2025. [Online]. Available: https://www.ibm.com/think/topics/machine-learning.

[67] W. Banzhaf, P. Machado, M. Zhang, W. Banzhaf, M. Zhang, and P. Machado, *Handbook of Evolutionary Machine Learning*, 1st ed. Singapore: Springer, 2023. DOI: 10.1007/978-981-99-3814-8.

[68] R. C. Corrêa de Sá, "Movie's box office performance prediction: An approach based on movie's script, text mining, and deep learning," Mar. 2020.

[69] D. K. Simonton, "Cinematic success criteria and their predictors: The art and business of the film industry," *Psychology |& marketing*, vol. 26, no. 5, pp. 400–420, 2009. DOI: 10.1002/mar.20280.

[70] R. Kübler, R. Seifert, and M. Kandziora, "Content valuation strategies for digital subscription platforms," *Journal of cultural economics*, vol. 45, no. 2, pp. 295–326, 2021. DOI: 10.1007/s10824-020-09391-3.

[71] "Tudum by netflix," Netflix. [Online]. Available: https://www.netflix.com/tudum/top10.

[72] "Flixpatrol," Accessed: Jan. 13, 2025. [Online]. Available: https://flixpatrol.com/.

[73] J. Prag and J. Casavant, "An empirical study of the determinants of revenues and marketing expenditures in the motion picture industry," *Journal of cultural economics*, vol. 18, no. 3, pp. 217–235, 1994. DOI: 10.1007/BF01080227.

[74] R. A. Nelson and R. Glotfelty, "Movie stars and box office revenues: An empirical analysis," *Journal of cultural economics*, vol. 36, no. 2, pp. 141–166, 2012. DOI: 10.1007/s10824-012-9159-5.

[75] A. L. Hadida, "Motion picture performance: A review and research agenda," *International journal of management reviews : IJMR*, vol. 11, no. 3, pp. 297–335, 2009. DOI: 10.1111/j.1468-2370.2008.00240.x.

[76] B. R. Litman, "Predicting success of theatrical movies: An empirical study," *Journal of popular culture*, vol. 16, no. 4, pp. 159–175, 1983. DOI: 10.1111/j.0022-3840.1983.1604_159.x.

[77] S. Ramesh and D. Dursun, "Predicting box-office success of motion pictures with neural networks," *Expert Systems with Applications*, vol. 30, no. 2, pp. 243–254, 2006, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2005.07.018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417405001399.

[78] M. T. Lash and K. Zhao, "Early predictions of movie success: The who, what, and when of profitability," *Journal of Management Information Systems*, vol. 33, no. 3, pp. 874–903, 2016. DOI: 10.1080/07421222.2016.1243969.

[79] K. Carley and J. Diesner, *Automap: Software for network text analysis*, Carnegie Mellon University, Pennsylvania, 2005.

[80] S. D. Hunter III, S. Smith, and S. Singh, "Predicting box office from the screenplay: A text analytical approach," *Journal of Screenwriting*, vol. 7, no. 2, pp. 135–154, 2016. DOI: 10.1386/josc.7.2.135_1. [Online]. Available: https://research.ebsco.com/linkprocessor/plink?id=01e4bad2-5478-39ee-bb75-79f53d79cfc2.

[81] orgesleka. [Online]. Available: https://www.kaggle.com/orgesleka/IMDbmovies.

[82] Ankit, M. Lakshmi, K. A. Shastry, A. Sandilya, and R. Shekhar, "A comparative analysis of machine learning approaches for movie success prediction," in *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 2020, pp. 684–689. DOI: 10.1109/I-SMAC49090.2020.9243589.

[83] A. Oghina et al., "Predicting imdb movie ratings using social media," vol. 7224, Berlin, Heidelberg: Springer-Verlag, 2012, pp. 503–507. DOI: 10.1007/978-3-642-28997-2_51.

[84] P.-Y. Hsu et al., "Predicting movies user ratings with imdb attributes," vol. 8818, Cham: Springer International Publishing, 2014, pp. 444–453. DOI: 10.1007/978-3-319-11740-9_41.

[85] R. Abarja, "Movie rating prediction using convolutional neural network based on historical values," *International Journal of Emerging Trends in Engineering Research*, vol. 8, pp. 2156–2164, May 2020. DOI: 10.30534/ijeter/2020/109852020.

[86] A. Jose et al., "Predicting imdb movie ratings using roberta embeddings and neural networks," in Singapore: Springer, 2022, vol. 940, pp. 181–189. DOI: 10.1007/978-981-19-4453-6_13.

[87] IMDb. "Imdb top 250 movies," Accessed: Mar. 21, 2025. [Online]. Available: https://www.imdb.com/chart/top/.

[88] S. Sahu, R. Kumar, H. V. Long, and P. M. Shafi, "Early-production stage prediction of movies success using k-fold hybrid deep ensemble learning model," English, *Multimedia Tools and Applications*, vol. 82, no. 3, pp. 4031–4061, Jan. 2023.

[89] J. Eliashberg, A. Elberse, and M. A. A. M. Leenders, "The motion picture industry: Critical issues in practice, current research, and new research directions," *Marketing science (Providence, R.I.)*, vol. 25, no. 6, pp. 638–661, 2006. DOI: 10.1287/mksc.1050.0177.

[90] J. Eliashberg, S. K. Hui, and Z. J. Zhang, "Green-lighting movie scripts revenue forecasting and risk management," 2010. [Online]. Available: https://api.semanticscholar.org/CorpusID:31818948.

[91] S. O'Driscoll, "Early prediction of a film's box office success using natural language processing techniques and machine learning," 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:55660493.

[92] S.-H. Lee, h. y. yu hye yeon, and Y.-G. Cheong, "Analyzing movie scripts as unstructured text," *BigDataService*, no. 2017.43, pp. 249–254. 10.1109, 2017.

[93] "Internet movie script database," Accessed: Jan. 17, 2025. [Online]. Available: https://www.imsdb.com.

[94] W. S. Cleveland, "Lowess: A program for smoothing scatterplots by robust locally weighted regression," *The American statistician*, vol. 35, no. 1, p. 54, 1981. DOI: 10.2307/2683591.

[95] "Rotten tomatoes," Accessed: Jan. 17, 2025. [Online]. Available: https://www.rottentomatoes.com.

[96] P. Frangidis, K. Georgiou, and S. Papadopoulos, "Sentiment analysis on movie scripts and reviews: Utilizing sentiment scores in rating prediction," in ((eds) Artificial Intelligence Applications and Innovations. AIAI 2020. IFIP Advances in Information and Communication Technology, vol 583.), I. Maglogiannis, L. Iliadis, and E. Pimenidis, Eds., (eds) Artificial Intelligence Applications and Innovations. AIAI 2020. IFIP Advances in Information and Communication Technology, vol 583. Cham: Springer International Publishing, 2020, vol. 583, pp. 430–438. DOI: 10.1007/978-3-030-49161-1_36.

[97] J. Kim, Y. Lee, and I. Song, "From intuition to intelligence: A text mining–based approach for movies' green-lighting process," *Internet research*, vol. 32, no. 3, pp. 1003–1022, 2022. DOI: 10.1108/INTR-11-2020-0651.

[98] M. H. Shahid, M. A. Islam, and M. Beg, "Exploiting time series based story plot popularity for movie success prediction," *Multimedia Tools and Applications*, vol. 82, no. 3, pp. 3509–3534, 2023. DOI: 10.1007/s11042-022-13219-x.

[99] R. Banik. "The movies dataset," Accessed: Jan. 5, 2025. [Online]. Available: https://www.kaggle.com/datasets/rounakbanik/the-movies-dataset/data.

[100] Y. Kim, Y. Cheong, and J. Lee, "Prediction of a movie's success from plot summaries using deep learning models," Jan. 2019, pp. 127–135. DOI: 10.18653/v1/W19-3414.

[101] M. E. Peters, M. Neumann, M. Iyyer, C. Gardner Matt and Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," *CoRR*, vol. abs/1802.05365, 2018. arXiv: 1802.05365. [Online]. Available: http://arxiv.org/abs/1802.05365.

[102] S. Moon, N. Jalali, and R. Song, "Green-lighting scripts in the movie pre-production stage: An application of consumption experience carryover theory," *Journal of business research*, vol. 140, pp. 332–345, 2022. DOI: 10.1016/j.jbusres.2021.11.004.

[103] N. Cressie, *Statistics for spatial data*. John Wiley |& Sons, 2015.

[104] K. Vonnegut. "The shapes of stories," Accessed: Jan. 5, 2025. [Online]. Available: https://www.youtube.com/watch?v=oP3c1h8v2ZQ.

[105] M. Del V., A. Kharlamov, G. Parry, and G. Pogrebna, "The data science of hollywood: Using emotional arcs of movies to drive business model innovation in entertainment industries," *arXiv.org*, 2018. DOI: 10.1080/01605682.2019.1705194.

[106] M. Del Vecchio, A. Kharlamov, G. Parry, and G. Pogrebna, "Improving productivity in hollywood with data science: Using emotional arcs of movies to drive product and service innovation in entertainment industries," *Journal of the Operational Research Society*, vol. 72, pp. 1–28, Mar. 2020. DOI: 10.1080/01605682.2019.1705194.

[107] V. Major, A. Surkis, and Y. Aphinyanaphongs, "Utility of general and specific word embeddings for classifying translational stages of research," *AMIA Annual Symposium proceedings*, vol. 2018, pp. 1405–1414, 2018.

[108] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin, "A neural probabilistic language model," *J. Mach. Learn. Res.*, vol. 3, no. null, pp. 1137–1155, Mar. 2003.

[109] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[110] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, 2013, pp. 3111–3119.

[111] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," *CoRR*, vol. abs/1405.4053, 2014. arXiv: 1405.4053. [Online]. Available: http://arxiv.org/abs/1405.4053.

[112] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," *CoRR*, vol. abs/1607.01759, 2016. arXiv: 1607.01759. [Online]. Available: http://arxiv.org/abs/1607.01759.

[113] J. Wang, "Information extraction from tv series scripts for uptake prediction," 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:67153700.

[114] S. Sivakumar and R. Rajalakshmi, "Analysis of sentiment on movie reviews using word embedding self-attentive lstm," *International journal of ambient computing and intelligence*, vol. 12, no. 2, pp. 33–52, 2021. DOI: 10.4018/IJACI.2021040103.

[115] N. Vaswani Ashish and Shazeer, N. Parmar, L. Uszkoreit Jakob and Jones, A. N. Gomez, and I. Kaiser Lukasz and Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017. arXiv: 1706.03762. [Online]. Available: http://arxiv.org/abs/1706.03762.

[116] E. I. T. C. Academy. "How does the concept of contextual word embeddings, as used in models like bert, enhance the understanding of word meanings compared to traditional word embeddings?" Accessed: Jun. 11, 2024. [Online]. Available: https://eitca.org/artificial-intelligence/eitc-ai-adl-advanced-deep-learning/natural-language-processing/advanced-deep-learning-for-natural-language-processing/examination-review-advanced-deep-learning-for-natural-language-processing/how-does-the-concept-of-contextual-word-embeddings-as-used-in-models-like-bert-enhance-the-understanding-of-word-meanings-compared-to-traditional-word-embeddings/.

[117] C. Caudill. "Writing craft wednesday : Michael hauge's 6 stage story structure," Accessed: Jul. 4, 2025. [Online]. Available: https://crystalcaudill.com/tag/complications-and-higher-stakes/.

[118] J. E. Cutting, "Narrative theory and the dynamics of popular movies," *Psychonomic bulletin |& review*, vol. 23, no. 6, pp. 1713–1743, 2016. DOI: 10.3758/s13423-016-1051-4.

[119] P. Papalampidi, F. Keller, and M. Lapata, "Movie plot analysis via turning point identification," *arXiv.org*, 2019.

[120] M. Lee, H. Kwon, J. Shin, W. Lee, B. Jung, and J.-H. Lee, "Transformer-based screenplay summarization using augmented learning representation with dialogue information," in *Proceedings of the Third Workshop on Narrative Understanding*, Association for Computational Linguistics, Jun. 2021, pp. 56–61. DOI: 10.18653/v1/2021.nuse-1.6. [Online]. Available: https://aclanthology.org/2021.nuse-1.6/.

[121] KseniaGur. "Movie scripts corpus," Accessed: Jun. 20, 2024. [Online]. Available: https://www.kaggle.com/datasets/gufukuro/movie-scripts-corpus/data.

[122] P. Chokhra. "Movie scripts," Accessed: Jun. 20, 2024. [Online]. Available: https://www.kaggle.com/datasets/parthplc/movie-scripts.

[123] T. S. Savant. “The script savant,” Accessed: Jun. 20, 2024. [Online]. Available: https://thescriptsavant.com/movies.html.

[124] PedroUria. “Nlp-movie-scripts,” Accessed: Jun. 20, 2024. [Online]. Available: https://github.com/PedroUria/NLP-Movie_Scripts/tree/master/scripts.

[125] AdeboyeML. “Film script analysis,” Accessed: Jun. 21, 2024. [Online]. Available: https://github.com/AdeboyeML/Film_Script_Analysis/tree/master/Films.

[126] compSciKai. “Cmpt419-special-topics-ai-movie-script-sentiment,” Accessed: Jun. 21, 2024. [Online]. Available: https://github.com/compSciKai/cmpt419-special-topics-ai-movie-script-sentiment/tree/master/dataset/movie_scripts.

[127] JoeKarlsson. “Bechdel-test,” Accessed: Jun. 21, 2024. [Online]. Available: https://github.com/JoeKarlsson/bechdel-test/tree/develop/scripts.

[128] hellomasaya. “Sexism-in-movies,” Accessed: Jun. 21, 2024. [Online]. Available: https://github.com/hellomasaya/sexism-in-movies/tree/master/data/scripts.

[129] M. Kordabadi, A. Nazari, and M. Mansoorizadeh, “A movie recommender system based on topic modeling using machine learning methods,” *International Journal of Web Research*, vol. 5, no. 2, pp. 19–28, 2022. DOI: 10.22133/ijwr.2022.370251.1139. eprint: https://ijwr.usc.ac.ir/article_164092_06a193e69311f81adb3b070763733cc5.pdf. [Online]. Available: https://ijwr.usc.ac.ir/article_164092.html.

[130] B. Chu, D. Xu, and D. Wu, “Comparison and analysis of the machine learning in the movie subtitle document classification,” *Applied and Computational Engineering*, vol. 5, pp. 642–648, 2023.

[131] S. Eden, A. Livne, O. Sar Shalom, B. Shapira, and D. Jannach, “Investigating the value of subtitles for improved movie recommendations,” in *Proceedings of the 30th ACM Conference on User Modeling, Adaptation and Personalization*, 2022, pp. 99–109. [Online]. Available: https://github.com/sagiede/SubtitleCF.

[132] adiamaan. “Movie subtitle dataset,” Accessed: Jun. 22, 2024. [Online]. Available: https://www.kaggle.com/datasets/adiamaan/movie-subtitle-dataset.

[133] Springfield. “Springfield! springfield!” Accessed: Aug. 12, 2024. [Online]. Available: https://www.springfieldspringfield.co.uk/.

[134] D. Grijalva. “Movie industry,” Accessed: Jul. 15, 2024. [Online]. Available: https://www.kaggle.com/datasets/danielgrijalvas/movies.

[135] A. Lo Bello. “The ultimat film statistics dataset for ml,” Accessed: Jul. 16, 2024. [Online]. Available: https://www.kaggle.com/datasets/alessandrolobello/the-ultimate-film-statistics-dataset-for-ml.

[136] V. I. Levenshtein, “Binary codes capable of correcting deletions, insertions, and reversals,” *Soviet Physics Doklady*, vol. 10, no. 8, pp. 707–710, 1966.

[137] N. Fasano. “Enhanced movie recommendations with collaborative topic modeling,” Accessed: Sep. 22, 2024. [Online]. Available: https://github.com/nfasano/movie_recsys.

[138] D. Alberani, H. Uyur, and G. Corbelli, *Cinemagoer*, version 2023.05.01, 2023. [Online]. Available: https://cinemagoer.github.io/.

[139] IMDb. “Imdb non-commercial datasets,” Accessed: Sep. 11, 2024. [Online]. Available: https://developer.imdb.com/non-commercial-datasets/.

[140] J. Eliashberg, S. K. Hui, and Z. J. Zhang, “From story line to box office: A new approach for green-lighting movie scripts,” *Management science*, vol. 53, no. 6, pp. 881–893, 2007. DOI: 10.1287/mnsc.1060.0668.

[141] M. Burgos, M. L. Campanario, J. Lara, and D. Lizcano, “Using decision trees to characterize and predict movie profitability on the us market,” *Lecture Notes in Engineering and Computer Science*, vol. 1, pp. 274–279, Mar. 2015.

[142] W. N. Goetzmann, S. A. Ravid, and R. Sverdlove, “The pricing of soft and hard information: Economic lessons from screenplay sales,” *Journal of cultural economics*, vol. 37, no. 2, pp. 271–307, 2013. DOI: 10.1007/s10824-012-9183-5.

[143] M. Ghiassi, D. Lio, and B. Moon, “Pre-production forecasting of movie revenues with a dynamic artificial neural network,” *Expert Systems with Applications*, vol. 42, no. 6, pp. 3176–3193, 2015, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2014.11.022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417414007088.

[144] E. Chu and D. Roy, “Audio-visual sentiment analysis for learning emotional arcs in movies,” IEEE, 2017, pp. 829–834. DOI: 10.1109/ICDM.2017.100.

[145] P. Virtanen et al., “Scipy 1.0: Fundamental algorithms for scientific computing in python,” *Nature Methods*, vol. 17, pp. 261–272, 2020. DOI: 10.1038/s41592-019-0686-2. [Online]. Available: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.zscore.html.

[146] W. Meert, K. Hendrickx, T. Van Craenendonck, P. Robberechts, H. Blockeel, and J. Davis, *Dtaidistance (version v2). zenodo*. DOI: http://doi.org/10.5281/zenodo.5901139. [Online]. Available: https://github.com/wannesm/dtaidistance.

[147] L. R. Rabiner and B. Juang, *Fundamentals of speech recognition* (Prentice Hall signal processing series), eng. Prentice Hall, 1993, ISBN: 0130151572.

[148] "Dynamic time warping," in *Information Retrieval for Music and Motion*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 69–84, ISBN: 978-3-540-74048-3. DOI: 10.1007/978-3-540-74048-3_4. [Online]. Available: https://doi.org/10.1007/978-3-540-74048-3_4.

[149] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. New York: John Wiley |& Sons, 1990, ISBN: 978-0-471-87876-6.

[150] T. scikit-learn-extra development team. "Scikit-learn-extra: A python module for machine learning that extends scikit-learn. "[Online]. Available: https://github.com/scikit-learn-contrib/scikit-learn-extra.

[151] W.-M. Lee, *Python machine learning*, 1st ed. Indianapolis, Indiana: Wiley, 2019, ISBN: 1-5231-2845-3.

[152] Scikit-learn, *Silhouette score*. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html.

[153] SciPy, *Univariate spline*. [Online]. Available: https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.UnivariateSpline.html.

[154] A. J. Reagan, L. Mitchell, D. Kiley, C. M. Danforth, and P. S. Dodds, "The emotional arcs of stories are dominated by six basic shapes," *EPJ data science*, vol. 5, no. 1, pp. 1–12, 2016. DOI: 10.1140/epjds/s13688-016-0093-1.

[155] Wikipedia. "Tukey's range test," Accessed: Feb. 4, 2025. [Online]. Available: https://en.wikipedia.org/wiki/Tukey%27s_range_test.

[156] M. Marvel, *Encyclopedia of New Venture Management*, M. Marvel, Ed. SAGE Publications, Incorporated, 2012, pp. 120–121. eprint: ProQuestEbookCentral.

[157] tazwar2700. "Emotion nrc affect lexicon. "[Online]. Available: https://github.com/tazwar2700/Emotion_NRC_affect_lexicon.

[158] metalcorebear. "Nrclex. "[Online]. Available: https://github.com/metalcorebear/NRCLex.

[159] S. Bird, E. Loper, and E. Klein, *Natural language processing with python*, 2009. [Online]. Available: https://www.nltk.org/.

[160] "The 8 basic emotions of the plutchik's wheel," Accessed: Feb. 6, 2025. [Online]. Available: https://wellwo.es/en/the-8-basic-emotions-of-the-plutchiks-wheel/.

[161] "Putting some emotion into your design – plutchik's wheel of emotions," Interaction Design Foundation - IxDF, Accessed: Feb. 6, 2025. [Online]. Available: https://www.interaction-design.org/literature/article/putting-some-emotion-into-your-design-plutchik-s-wheel-of-emotions.

[162] T. I. Team. "Variance inflation factor(vif)." R. C. Kelly, Ed., Accessed: Feb. 6, 2025. [Online]. Available: https://www.investopedia.com/terms/v/variance-inflation-factor.asp.

[163] Statsmodels. "Statsmodels.stats.outliers$_influence.variance_inflation_factor$," Accessed: Feb. 6, 2025. [Online]. Available: https://www.statsmodels.org/stable/generated/statsmodels.stats.outliers_influence.variance_inflation_factor.html.

[164] Scikit-learn. "Sklearn.feature_selection.mutual_info_regression," Accessed: Feb. 6, 2025. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.mutual_info_regression.html.

[165] R. R. .-. II. "Mutual information score - feature selection." Medium, Ed., Accessed: Feb. 7, 2025. [Online]. Available: https://bobrupakroy.medium.com/mutual-information-score-feature-selection-8eb19071664b.

[166] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Nov. 2019. [Online]. Available: https://arxiv.org/abs/1908.10084.

[167] "Sentence transformers all-minilm-l6-v2," Accessed: Dec. 10, 2024. [Online]. Available: https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2.

[168] X. Zhang et al., "Mgte: Generalized long-context text representation and reranking models for multilingual text retrieval," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, 2024, pp. 1393–1412.

[169] Z. Li, X. Zhang, Y. Zhang, D. Long, P. Xie, and M. Zhang, "Towards general text embeddings with multi-stage contrastive learning," *arXiv preprint arXiv:2308.03281*, 2023.

[170] Alibaba-NLP. "Gte-modernbert-base. "[Online]. Available: https://huggingface.co/Alibaba-NLP/gte-modernbert-base.

[171] B. Warner et al., *Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference*, 2024. arXiv: 2412.13663 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2412.13663.

[172] lightonai. "Modernbert-embed-large," Accessed: Mar. 20, 2025. [Online]. Available: https://huggingface.co/lightonai/modernbert-embed-large.

[173] X. Li and J. Li, "Angle-optimized text embeddings," *arXiv preprint arXiv:2309.12871*, 2023.

[174] WhereIsAI. "Uae-large_v1," Accessed: Feb. 2, 2025. [Online]. Available: https://huggingface.co/WhereIsAI/UAE-Large-V1.

[175] NovaSearch. "Stella_en_400m_v5," Accessed: Feb. 2, 2025. [Online]. Available: https://huggingface.co/NovaSearch/stella_en_400M_v5.

[176] NovaSearch. "Uaestella_en_1.5b_v5," Accessed: Feb. 2, 2025. [Online]. Available: https://huggingface.co/NovaSearch/stella_en_1.5B_v5.

[177] R. Schwaber-Cohen. "Chunking strategies for llm applications," Accessed: Feb. 9, 2025. [Online]. Available: https://www.pinecone.io/learn/chunking-strategies/.

[178] J. Xing, D. Luo, C. Xue, and R. Xing, *Comparative analysis of pooling mechanisms in llms: A sentiment analysis perspective*, 2025. arXiv: 2411.14654 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2411.14654.

[179] J. Berger, Y. D. Kim, and R. Meyer, "What makes content engaging? how emotional dynamics shape success," *The Journal of consumer research*, vol. 48, no. 2, pp. 235–250, 2021. DOI: 10.1093/jcr/ucab010.

[180] R. J. Melnick, "The importance of morally satisfying endings: Cognitive influences on storytelling in gillian flynn's 'gone girl.'," presented at the Proceedings of the 41st Annual Meeting of the Cognitive Science Society, 2019, pp. 145–151.

[181] T. Evans. "How bad endings ruin great experiences, The peak-end rule and why it skews perceptions." M. Quirk, Ed., Accessed: Feb. 10, 2025. [Online]. Available: https://www.psychologytoday.com/au/blog/trust-games/202111/how-bad-endings-ruin-great-experiences.

[182] Praneeth0045, Accessed: Dec. 10, 2024. [Online]. Available: https://www.kaggle.com/datasets/praneeth0045/imdb-top-10000-movie-plots-keywords.

[183] K. Sudipta, M. Suraj, L.-M. A. Pastor, and S. Thamar, "Mpst: A corpus of movie plot synopses with tags," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, N. C. (chair) et al., Eds., Miyazaki, Japan: European Language Resources Association (ELRA), May 2018, ISBN: 979-10-95546-00-9.

[184] G. Delmestri, F. Montanari, and A. Usai, "Reputation and strength of ties in predicting commercial success and artistic merit of independents in the italian feature film industry," *Journal of management studies*, vol. 42, no. 5, pp. 975–1002, 2005. DOI: 10.1111/j.1467-6486.2005.00529.x.

[185] Nvidia, *Rapids graph documentation*, version 24.12.00, 2025. [Online]. Available: https://docs.rapids.ai/api/cugraph/stable/.

[186] NetworkX, *Networkx - network analysis in python*, version 3.4.2, 2025. [Online]. Available: https://networkx.org/.

[187] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995. DOI: 10.1007/BF00994018.

[188] F. Pedregosa et al., "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, pp. 2825–2830, 2011. DOI: 10.5555/1953048.2078195.

[189] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society.Series B, Methodological*, vol. 58, no. 1, pp. 267–288, 1996. DOI: 10.1111/j.2517-6161.1996.tb02080.x.

[190] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society.Series B, Statistical methodology*, vol. 67, p. 768, 2005. DOI: 10.1111/j.1467-9868.2005.00527.x.

[191] A. Nagpal. "L1 and l2 regularization methods, explained." B. Whitfield, Ed., Accessed: Feb. 25, 2025. [Online]. Available: https://builtin.com/data-science/l2-regularization.

[192] E. Lewinson. "A comprehensive overview of regression evaluation metrics," Nvidia, Accessed: Feb. 22, 2025. [Online]. Available: https://developer.nvidia.com/blog/a-comprehensive-overview-of-regression-evaluation-metrics/.

[193] H. Afzal, "Prediction of movies popularity using machine learning techniques," *International Journal of Computer Science and Network Security*, vol. 16, pp. 127–131, Aug. 2016.

[194] W. Bristi, Z. Zaman, and N. Sultana, "Predicting imdb rating of movies by machine learning techniques," Jul. 2019, pp. 1–5. DOI: 10.1109/ICCCNT45670.2019.8944604.

[195] S. Rajesh. "Confusion matrix for multiclass classification," Accessed: Feb. 25, 2025. [Online]. Available: https://medium.com/@sreenilarajesh/confusion-matrix-for-multiclass-classification-fe7b6901a541.

[196] IMDb. "Ratings faq," Accessed: Dec. 17, 2024. [Online]. Available: https://help.imdb.com/article/imdb/track-movies-tv/ratings-faq/G67Y87TFYYP6TWAV?showReportContentLink=true#.

[197] X. Ning, L. Yac, X. Wang, B. Benatallah, M. Dong, and S. Zhang, "Rating prediction via generative convolutional neural networks based regression," *Pattern Recognition Letters*, vol. 132, pp. 12–20, 2020. DOI: 10.1016/j.patrec.2018.07.028.

[198] G. San, "Movie success prediction using machine learning algorithms," Universitat Polotecnica De Catalunya Barcelonatech, Jul. 31, 2020. Accessed: Apr. 11, 2025. [Online]. Available: https://upcommons.upc.edu/bitstream/handle/2117/334930/TFM_Guille.pdf.

[199] I. Z. de Larranaga, "Using objective data from movies to predict other movies' approval rating through machine learning," Kristianstad University Sweden, Apr. 30, 2021. Accessed: Apr. 11, 2025. [Online]. Available: https://www.diva-portal.org/smash/get/diva2:1574293/FULLTEXT01.pdf.

[200] OpenAI. "Sora. "[Online]. Available: https://openai.com/sora/.

[201] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[202] S.-k. learn. "Gridsearchcv. "[Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html.

Appendix A LIWC extracted features

In this Appendix we list the features that have been retained from the LIWC extraction process after using RFE. In total, 115 out of the 151 dimensions were retained for concatenation to the main metadata vector.

WC	Analytic	Clout	Authentic	Tone
WPS	BigWords	Dic	Linguistic	function
pronoun	ppron	i	we	you
shehe	they	ipron	det	article
number	auxverb	adverb	conj	negate
verb	adj	quantity	Drives	affiliation
achieve	power	Cognition	allnone	cogproc
insight	cause	discrep	tentat	differ
memory	Affect	tone_pos	tone_neg	emotion
emo_pos	emo_neg	emo_anx	emo_sad	swear
Social	socbehav	prosocial	polite	conflict
moral	comm	socrefs	family	female
male	Culture	politic	ethnicity	tech
Lifestyle	leisure	home	work	money
relig	Physical	illness	wellness	mental
substances	sexual	food	death	need
want	acquire	lack	fulfill	fatigue
reward	risk	curiosity	allure	Perception
attention	space	visual	auditory	feeling
time	focuspast	focuspresent	Conversation	netspeak
assent	nonflu	filler	AllPunc	Period
Comma	QMark	Exclam	Apostro	Narrativity_PlotProg
Narrativity_CogTension	Staging_10	PlotProg_10	CogTension_1	CogTension_8

Table A.1 LIWC Extracted Features from Movie Scripts

Appendix B Statistical features - emotion analysis

In this Appendix we give a more detailed explanation of the statistical features extracted from the emotion analysis in Section 3.4.2.

Feature	Description
Average	The mean value of the emotional arc over time.
Variance	A measure of the spread or dispersion of the emotional arc values.
Standard Deviation	The square root of variance, indicating the amount of variation in the emotional arc.
Minimum Value	The smallest value in the emotional arc.
Maximum Value	The largest value in the emotional arc.
Range	The difference between the maximum and minimum values in the emotional arc.
Median	The middle value of the emotional arc when sorted in ascending order.
Kurtosis	A measure of the ”tailedness” of the emotional arc distribution.
Skewness	A measure of the asymmetry of the emotional arc distribution.
Average Slope	The mean rate of change between consecutive points in the emotional arc.
Maximum Slope	The steepest positive rate of change in the emotional arc.
Minimum Slope	The steepest negative rate of change in the emotional arc.
Total Curvature	The cumulative curvature of the emotional arc, indicating how much the arc bends.
Mean Trough to Peak Slope	The average slope between troughs and subsequent peaks in the emotional arc.
Permutation Entropy	A measure of the complexity of the emotional arc based on ordinal patterns.
Shannon Entropy	A measure of the unpredictability or randomness in the emotional arc.

Table B.1 List of features extracted from emotional arcs with descriptions (Part 1).

Feature	Description
Autocorrelation	A measure of the similarity between the emotional arc and a time-lagged version of itself.
Mean Rate of Change	The average rate at which the emotional arc changes over time.
Standard Deviation of Rate of Change	The variability in the rate of change of the emotional arc.
Zero Crossings	The number of times the emotional arc crosses the zero line.
Positive Proportion	The proportion of the emotional arc that lies above the zero line.
Negative Proportion	The proportion of the emotional arc that lies below the zero line.
Neutral Proportion	The proportion of the emotional arc that lies close to the zero line.
Distance Peak to Peak	The average distance between consecutive peaks in the emotional arc.
Volatility	A measure of the variability or fluctuations in the emotional arc.
Mean Peak to Trough Slope	The average slope between peaks and subsequent troughs in the emotional arc.
Number of Peaks	The total number of local maxima in the emotional arc.
Number of Troughs	The total number of local minima in the emotional arc.

Table B.2 List of features extracted from emotional arcs with descriptions (Part 2).

Appendix C Results from using other regression models

In this Appendix we present the results of the other regression models discussed in the dissertation. Together with each model we also present the hyperparameters used for the training and evaluation.

Performance using the *stella_en_1.5B_v5* embedding was evaluated on two feature sets: one retaining, the other excluding, crew-specific attributes. This exclusion addressed features with potential availability constraints and significant multicollinearity, which had contributed to overfitting in non-linear models. The reduced feature space resulted in more homogeneous model performance metrics.

C.1 Dataset with crew power/experience features

Model	R^2	RMSE	MAE	MAPE	One-Away	Accuracy	F1
Base	-0.0334	1.1244	0.8845	0.1848	0.6693	0.8236	0.7440
Fixed Window: 20							
SVR Regressor	0.5255	0.7604	0.5859	0.1183	0.8476	0.8375	0.7913
XGBoost Regressor	0.4319	0.8325	0.6439	0.1282	0.7920	0.8316	0.8032
CatBoost Regressor	0.4042	0.8529	0.6594	0.1315	0.8004	0.8311	0.8043
LGBM Regressor	0.4483	0.8205	0.6348	0.1265	0.8161	0.8350	0.8043
Lasso Regressor	0.4078	0.8506	0.6584	0.1312	0.8074	0.8257	0.7999
ElasticNet Regressor	0.5030	0.7787	0.6072	0.1213	0.8331	0.8380	0.7928
MLP Regressor	0.4481	0.8204	0.6365	0.1266	0.8173	0.8267	0.7995

Table C.1 Regression and categorisation metrics for other models - Dataset with Crew Power/ExperienceFeatures.

C.1.1 Models Hyperparameters

SVR Regressor:

- C: 0.5,
- epsilon: 0.05
- gamma: 'scale'

XGB Regressor:

- n_estimators: 914, learning_rate: 0.1253, max_depth: 5
- min_child_weight: 6.316, subsample: 0.6619, colsample_bytree: 0.5291

- gamma: 0.0702, reg_alpha: 0.5037, reg_lambda: 19.0384, seed: 44

LightGBM Regressor:

- n_estimators: 2647, learning_rate: 0.0079, max_depth: 8
- num_leaves: 49, min_child_samples: 17, min_child_weight: 4.984
- subsample: 0.7850, colsample_bytree: 0.6226, reg_alpha: 0.0648
- reg_lambda: 5.0322, boosting_type: gbdt, random_seed: 50, verbosity: -1

CatBoost Regressor:

- iterations: 2825, learning_rate: 0.0712, depth: 5, objective: RMSE
- l2_leaf_reg: 0.7802, border_count: 139, colsample_bylevel: 0.7580
- grow_policy: SymmetricTree, min_data_in_leaf: 31, bootstrap_type: Bernoulli
- subsample: 0.8162, random_seed: 44

Lasso Regressor:

- max_iter: 3000, alpha: 0.0001, tol: 1e-3, random_state: 55

ElasticNet Regressor:

- max_iter: 10000, alpha: 0.02, l1_ratio: 0.01, random_state: 554

MLP Regressor:

- Architecture: hidden_layer_sizes=(64,32), activation=ReLU
- Solver: Adam, alpha=1e-6, batch_size='auto'
- Learning rate: adaptive, init=0.0001, power_t=0.5
- max_iter: 3000, tol: 1e-5, random_state: 554, shuffle=True
- early_stopping=True, validation_fraction=0.1
- momentum: 0.9, nesterovs_momentum=False
- beta_1: 0.9, beta_2: 0.999, epsilon: 1e-08
- n_iter_no_change: 200, max_fun: 20000

C.2 Dataset without crew power/experience features

Model	R^2	RMSE	MAE	MAPE	One-Away	Accuracy	F1
Base	-0.0334	1.1244	0.8845	0.1848	0.6693	0.8236	0.7440
Fixed Window: 20							
SVR Regressor	0.4681	0.8046	0.6216	0.1248	0.8235	0.8389	0.7932
XGBoost Regressor	0.4666	0.8056	0.6365	0.1233	0.8018	0.8362	0.7917
CatBoost Regressor	0.4643	0.8074	0.6367	0.1232	0.8139	0.8361	0.7932
LGBM Regressor	0.4678	0.8045	0.6345	0.1232	0.8164	0.8325	0.7910
Lasso Regressor	0.4146	0.8444	0.6628	0.1290	0.7946	0.8297	0.7970
ElasticNet Regressor	0.4267	0.8357	0.6578	0.1296	0.8041	0.8317	0.7801
MLP Regressor	0.4455	0.8220	0.6425	0.1258	0.8095	0.8283	0.8027

Table C.2 Regression and categorisation metrics for models using dataset without Crew Power/Experience Features.

C.2.1 Models Hyperparameters

SVR Regressor:

- C: 0.5,
- epsilon: 0.05
- gamma: 'scale'

XGB Regressor:

- n_estimators: 1320, learning_rate: 0.01319, max_depth: 6
- min_child_weight: 9.837, subsample: 0.6777, colsample_bytree: 0.6048
- gamma: 0.9303, reg_alpha: 0.19596, reg_lambda: 11.54133, seed: 44

LightGBM Regressor:

- n_estimators: 1362, learning_rate: 0.038979, max_depth: 8
- num_leaves: 28, min_child_samples: 20, min_child_weight: 5.1938
- subsample: 0.71269, colsample_bytree: 0.720899, reg_alpha: 0.0176
- reg_lambda: 96.441, boosting_type: gbdt, random_seed: 50, verbosity: -1

CatBoost Regressor:

- iterations: 2251, learning_rate: 0.019145, depth: 5, objective: RMSE
- l2_leaf_reg: 2.5211, border_count: 177, colsample_bylevel: 0.74528
- grow_policy: Depthwise, min_data_in_leaf: 46, bootstrap_type: Bernoulli
- subsample: 0.68368, random_seed: 44

Lasso Regressor:

- max_iter: 3000, alpha: 0.0001, tol: 1e-3, random_state: 55

ElasticNet Regressor:

- max_iter: 10000, alpha: 0.02, l1_ratio: 0.01, random_state: 554

MLP Regressor:

- Architecture: hidden_layer_sizes=(64,32), activation=ReLU
- Solver: Adam, alpha=1e-6, batch_size='auto'
- Learning rate: adaptive, init=0.0001, power_t=0.5
- max_iter: 3000, tol: 1e-5, random_state: 554, shuffle=True
- early_stopping=True, validation_fraction=0.1
- momentum: 0.9, nesterovs_momentum=False
- beta_1: 0.9, beta_2: 0.999, epsilon: 1e-08
- n_iter_no_change: 200, max_fun: 20000

Appendix D Description of various code parts

In this Appendix we list and explain the Jupyter notebooks used in this analysis. Each notebook contains code which was used in this dissertation. Some additional files are also being explained. For ease of reference, the notebooks and the files are being uploaded onto an icloud repository for easier retrieval.

The link to the icloud repository:
Click here to access the repository

D.1 Jupyter Notebooks

- *training_testing_rolling.ipynb* - This is the main notebook which loads the metadata, embeddings and other features and concatenates them to form the dataset, that is eventually split into the training and testing sets for each validation fold. The notebook also presents the evaluation metrics and visualizations for each fold.
- *crew_scores_rolling.ipynb* - This notebook contains the process of extracting the relevant information from the IMDb non-commercial datasets to construct the network graphs. Selected information from the respective datasets was merged into one dataset, filtered from movies which did not have at least 1000 votes. This dataset 'merged_filtered_for_network_graph.csv' was saved for use in another notebook for constructing the network graphs and extracting the centrality measures. The notebook also generates the *rolling_ratings.json* and *rolling_experience.json* files used in the *training_testing_rolling.ipynb* notebook to update the training dataset with each roll forward of the cut-off year.
- *collab-network-crew.ipynb* - The 'merged_filtered_for_network_graph.csv' dataset is loaded into this notebook. Movies released on or after 2015 are filtered out to avoid any data leakage. This notebook makes use of CUDA and the RAPIDS ecosystem from NVIDIA to utilise the GPU. The dataset is converted to *cudf* format for speedier access and processing. Collaboration edges are created based on the average rating of the related connecting movie. A *cugraph* is created from these connections and a function to visualise the connections of a particular crew member is provided. A sample of the top 15 collaborators for Steven Spielberg is provided as an example. The various centrality measures are calculated, collated in a dataframe and saved as 'graph_people.csv'.
- *graph_features_update_metadata.ipynb* - The main screenplays/subtitles datasets are updated in this notebook. The 'graph_people.csv', which includes the centrality measures calculated in the previous notebook are loaded and allocated to each movie

based on the list of crew who worked on that movie. For each crew member (director, writer, producer, actor) the maximum value is taken as the centrality measure for that crew type for that movie. A scatter plot for each measure is provided against the rating to identify any correlation. A correlation matrix is also presented. Finally the respective main screenplays/subtitles are saved with the changes.

- *liwc_data_extraction-new.ipynb* - The various linguistic features calculated by the LIWC-22 software are are extracted the main screenplays/subtitles datasets via this notebook. Scripts and subtitles files are pre-processed and saved in a folder that is subsequently read by the LIWC-22 software for processing. The extracted features are saved in a csv file and then re-imported into this notebook. As most of the results produced by the LIWC-22 software are in a percentage format, these are converted to absolute values based on the relative word count for each movie script. Csv files are created for each type of dataset (screenplays and subtitles) and saved accordingly.

- *emotions_extraction.ipynb* - In this notebook, after loading, reading and pre-processing the scripts, the VADER sentiment values are extracted for each movie, normalised using discrete cosine transformation. After plotting some examples (in raw and normalised versions) the normalised values are clustered using dynamic time warping as a distance measure, yielding 6 discrete clusters. These are grouped and using spline interpolation, a mean arc shape is presented for each cluster. Cluster labels are saved and analysed using various kde plots. The labels are subsequently used as a feature in the main dataset.

- *emotion_intensity_wip.ipynb* - This is an extension of the previous notebook as it processed the same scripts to extract, using various lexicons, the intensity and presence scores of various emotions. The intensity scores are saved as 'nrc_intensity.csv' and the presence scores are saved as 'nrc_lexicon_emotions.csv'. Scatter plots are also presented for each score against the rating and a filtering for multicollinearity was performed to retain 7 main features to be amalgamated to the main dataset.

- *emotions_features_training.ipynb* - In this notebook, statistical features are extracted from the emotional arcs calculated in the *emotions_extraction.ipynb* notebook. A scatter plot analysis is also presented for each feature against the rating. Using mutual information a filtering of the features is conducted to retain the ones that will be concatenated to the main dataset.

- *keywords_database.ipynb* - The keywords pertaining to each movie are collated in this notebook. While some keywords were obtained from external datasets, others had to be extracted manually, by obtaining the 'overview' or 'synopsis' of the movie from

IMDb and extracting the keywords using the KeyBERT transformer model, hosted on Huggingface. The keywords are then encoded using Word2Vec and tested for similarity using cosine similarity. The embedded keywords (having a dimension of 50 each) were saved as 'movies_merged_keywords.csv' to be then imported into the main *training_testing_rolling.ipynb* notebook to be concatenated to the main dataset.

- *modernbert_embeddings.ipynb* - A number of notebooks were created for each of the embedding models used to embed the scripts. The format for each embedding followed a similar process. Models which were available via SentenceTransformers utilised the typical calling structure provided by the SentenceTransformers script, while others, utilized the standard torch based coding for loading and using the model. In this notebook an example for the SentenceTransformers models is provided, for the *modernbert-embed-large* model. After pre-processing the text, a chunking function is created, followed by a function that gets the processed text and converts it into an embedded vector. After concatenating the various embeddings, a similarity test using FAISS library is conducted to check the semantic similarity of the embedded vectors, in terms of movie story similarity. The embeddings are saved for subsequent loading into the *training_testing_rolling.ipynb*.

D.2 Files

- *movies_full4.csv* - This file contains the metadata for full screenplays.
- *subtitles_df5.csv* - This file contains the metadata for subtitles scraped from various sites.
- *movielens_subtitles_v6.csv* - This file contains the metadata for subtitles extracted from the Movielens-20M Dataset.
- *spring_subtitles.csv* - This file contains the metadata for the subtitles extracted from the website https://www.springfieldspringfield.co.uk/.
- *liwc_data_full.csv* - Contains feature data extracted using LIWC-22 for full screenplays.
- *liwc_data_subtitles_full.csv* - Contains feature data extracted using LIWC-22 for subtitles.
- *liwc_data_movielens_full.csv* - Contains feature data extracted using LIWC-22 for Movielenes-20M dataset subtitles.
- *liwc_data_springfield_full.csv* - Contains feature data extracted using LIWC-22 for subtitles extracted from the website https://www.springfieldspringfield.co.uk/.

- *mini_embeddings_merged.csv* - Contains the embeddings generated from scripts using *all-MiniLM-L6-v2*
- *modernbert_base_embeddings_merged.csv* - Contains the embeddings generated from scripts using *gte-modernbert-base*
- *angle_embeddings_merged.csv* - Contains the embeddings generated from scripts using *UAE-Large-V1*
- *modernbert_large_embeddings_merged.csv* - Contains the embeddings generated from scripts using *ModernBERT-embed-large*
- *stella400_embeddings_new_merged.csv* - Contains the embeddings generated from scripts using *stella_en_400m_v5*
- *jasper_embeddings_merged.csv* - Contains the embeddings generated from scripts using *jasper_en_vision_language_v1*
- *stella_embeddings_merged_140225.csv* - Contains the embeddings generated from scripts using *stella_en_1.5B_v5*
- *sentiment_df.csv* - Contains the clustering labels for each movie, as extracted from *emotions_extraction.ipynb*
- *filtered_general_features.csv* - Contains the statistical feature for each movie, as extracted from *emotions_features_training.ipynb*
- *nrc_lexicon_emotions.csv* - Contains the emotion presence scores as extracted by the NRC Lexicon in the notebook *emotion_intensity_wip.ipynb*. The columns selected for upload are 'movie_id','positive', 'surprise', 'trust', 'fear', 'negative', 'disappointment', 'optimism'.
- *nrc_intensity.csv* - Contains the emotion intensity scores as extracted by the NRC Lexicon in the notebook *emotion_intensity_wip.ipynb*. The columns selected for upload are 'movie_id','surprise_intensity', 'love_intensity', 'contempt_intensity'.
- *movies_merged_keywords.csv'* - Contains the embedded keywords using the Word2Vec model in the notebook *keywords_database.ipynb*.
- *emotional_arcs_mae.csv* - Contains the MAE results for each emotional arc cluster label across the 10 validation datasets. Used to analyse results.
- *genres_mae.csv* - Contains the MAE results for each genre across the 10 validation datasets. Used to analyse results.

Appendix E Description of Data Usage

The data for this research includes a collection of movie screenplays. These screenplays served as the source material for computational analysis aimed at exploring the relationship between script characteristics and movie audience reception, specifically as reflected in IMDb user ratings. This is a non-commercial, academic endeavour.

The methodology involved processing the full text of each screenplay to extract quantifiable features. This process included:

- **Linguistic Feature Extraction**: Analysing the text to identify and quantify various linguistic elements.
- **Narrative Structure Analysis**: Examining the script structure to derive metrics related to narrative flow and organisation.
- **Emotional Trajectory Analysis**: Assessing the emotional content within the dialogue and descriptions throughout the script.
- **Script Embedding**: Transforming the entire script text into vector representations that capture semantic and stylistic information.

The screenplays were utilized strictly as data inputs for these automated analytical techniques. The output of this processing is a dataset of numerical features and vector representations derived from the scripts.

Crucially, this research does not involve the reproduction, distribution, or display of the original movie screenplays, subtitles or any significant portions thereof in a human-readable format within the thesis. No dialogue, scene descriptions, or other textual content from the scripts will be quoted or presented for the reader to consume as they would the original work. Furthermore, no audio or visual components of the movies are used or included. The analysis operates on the computational representations of the scripts, and the thesis presents the findings derived from this data and the resulting predictive model. The use of the screenplays is confined to their role as data within this analytical research process.