INSTITUTO
UNIVERSITÁRIO
DE LISBOA

Predicting Success in Latin-Language Cinema: A Machine Learning Approach to Identify Key Factors in Film Performance

Margarida Paias Moura

Master in Computer Science and Business Management

Supervisors:
PhD *Ricardo Daniel Santos Faro Marques Ribeiro*, Associate Professor,
Iscte – Instituto Universitário de Lisboa

PhD *Fernando Manuel Marques Batista*, Associate Professor,
Iscte – Instituto Universitário de Lisboa

October, 2024

TECHNOLOGY
AND ARCHITECTURE

Department of Information Science and Technology

Predicting Success in Latin-Language Cinema: A Machine Learning Approach to Identify Key Factors in Film Performance

Margarida Paias Moura

Master in Computer Science and Business Management

Supervisors:
PhD *Ricardo Daniel Santos Faro Marques Ribeiro*, Associate Professor,
Iscte – Instituto Universitário de Lisboa

PhD *Fernando Manuel Marques Batista*, Associate Professor,
Iscte – Instituto Universitário de Lisboa

October, 2024

Dedication to my Parents, Teachers, and Friends.

Acknowledgment

I would like to extend my sincere thanks to my parents, whose unwavering support and love have been the foundation of my journey. Without their encouragement and belief in me, I would not be the person I am today.

I am also deeply grateful to my boyfriend and my friends, who have supported me at all times, giving me comfort and motivation whenever I needed it.

A special thanks to Professors Ricardo Ribeiro and Fernando Batista, my thesis supervisors, for their patience, understanding, and encouraging guidance throughout this process. Their support has been invaluable in helping me achieve this goal.

Resumo

Esta tese tem como objetivo de identificar os principais fatores que influenciam o resultados na previção do sucesso de filmes em línguas latinas. Focando em filmes em português, francês, espanhol, italiano, romeno, galego, catalão e provençal, o estudo aborda uma lacuna na investigação, que frequentemente se centra nos cinemas de língua inglesa e indiana. Foi criado um conjunto de dados abrangente, incluindo variáveis como orçamento, popularidade do elenco, influência do realizador e data de lançamento, com o objetivo de identificar as características que mais afetam o sucesso de um filme. Esta pesquisa aplica técnicas de aprendizagem automática para prever o sucesso de filmes, como um problema de classificação binário. Após comparar algoritmos como Random Forest, Support Vector Machines (SVM), K-Nearest Neighbors (KNN), XGBoost e Redes Neuronais, o modelo XGBoost demonstrou um desempenho superior. Os resultados revelam que fatores como a popularidade do filme, o elenco e o orçamento têm um impacto significativo no sucesso da produção. Este estudo oferece observações valiosas para cineastas e profissionais da indústria, permitindo decisões baseadas em dados para aumentar o sucesso das produções em línguas latinas. Futuramente poderão ser incorporadas métricas de sentimentos nas redes sociais e de envolvimento do público para melhorar ainda mais a precisão preditiva. Este trabalho contribui para uma melhor compreensão dos determinantes de sucesso na indústria cinematográfica em línguas latinas, destacando o potencial da aprendizagem automática para fornecer observações baseadas em dados a um segmento sub-representado do cinema global.

Palavras Chave: *Aprendizagem Automática*, *Predição do successo de filmes*, *Indústria Cinematográfica*, *Filmes em Linguagem Latina*

Abstract

This dissertation aims to identify the key factors that influence the prediction of success in Latin-language films. Focusing on Portuguese, French, Spanish, Italian, Romanian, Galician, Catalan, and Occidental films, the study addresses a gap in research often centered on English-speaking and Indian cinemas. A comprehensive dataset consolidating information, including variables like budget, revenue, cast popularity, director influence, and release date, was developed to identify features that most affect a movie's success. This research applies machine learning to predict the movie's success, as a binary classification problem. After comparing algorithms such as Random Forest, SVM, KNN, XGBoost, and Neural Networks, the XGBoost model demonstrated superior performance. Findings reveal that factors like movie popularity, cast, and budget significantly impact the movie's success. This study offers valuable insights for filmmakers and industry professionals, enabling data-driven decisions to enhance the success of Latin-language productions. Future research may incorporate social media sentiment and audience engagement metrics to improve predictive accuracy further. This work contributes to a better understanding of success determinants in the Latin-language film industry, highlighting the potential of machine learning to provide data-driven insights into an underrepresented segment of global cinema.

Keywords: Machine Learning, Movie Success Prediction, Film Industry, Latin-Language movies

Contents

List of Figures

List of Tables

List of Acronyms

DSA: Dual Sentiment Analysis

GAN: Generative Adversarial Networks

KNN: K-Nearest Neighbors

ML: Machine Learning

MLP: Multilayer Perceptron

SLR: Systematic Literature Review

SVC: Support Vector Classification

SVM: Support Vector Machines

ReLU: Rectified Linear Unit

CHAPTER 1

Introduction

Nowadays, the film industry has a significant impact in society. This industry that started to entertain has grown over the years and become a billion-dollar business which achieved total revenue of 5.8 billion euro in European Cinema operators, according to UNIC (International Union of Cinemas). [1] Despite its financial success, the film industry is characterized by short product life cycles and high-risk factors, making it imperative for everyone involved in movie production, including directors, producers, and actors, to strategically minimize the risk of a film becoming a commercial failure [1].

1.1. Motivation and Context

Machine learning has become an indispensable tool in the film industry, offering transforming potential by enabling the prediction of a movie's success. By analyzing attributes such as genre, language, director, actor, and budget, stakeholders can make informed decisions that reduce risk and increase revenue [2].

Selecting the right features to build a model is crucial in identifying the factors that significantly impact a movie's profitability. By accurately predicting a movie's success based on relevant parameters, movie studios have the potential to save hundreds of millions of dollars annually [3].

Many studies explore success factors in American, British, or Indian films, but there is a gap in understanding Latin-language movies. This study aims to fill that gap by identifying key features for predicting success in Latin-language films. Doing so contributes to a better understanding of what makes hit movies globally.

1.2. Objective and Research Questions

This work aims to contribute to risk reduction in movie production, facilitating better management practices and ultimately leading to higher stakeholder revenue. Additionally, it benefits consumers by providing predictions about a film's potential success before its release, demonstrating the remarkable flexibility and effectiveness of machine learning in addressing complex problems by adapting to various data sources [4].

To achieve that goal, this dissertation aims to answer the following research questions:

- Q1: Which techniques can be used to predict the movie's success?
- Q2: Which available features can be used for predicting movie success?
- Q3: Can machine learning models build with the right features provide meaningful insights to stakeholders in the film industry?

[1] `https://www.unic-cinemas.org/fileadmin/user_upload/Publications/2023/UNIC_Annual_Report_2023.pdf`

1.3. Methodology

The methodology chosen for this study is CRISP-DM, which stands for Cross-Industry Standard Process for Data Mining [5] and is a systematic approach used in data mining projects. It consists of six phases as shown in Figure 1.1

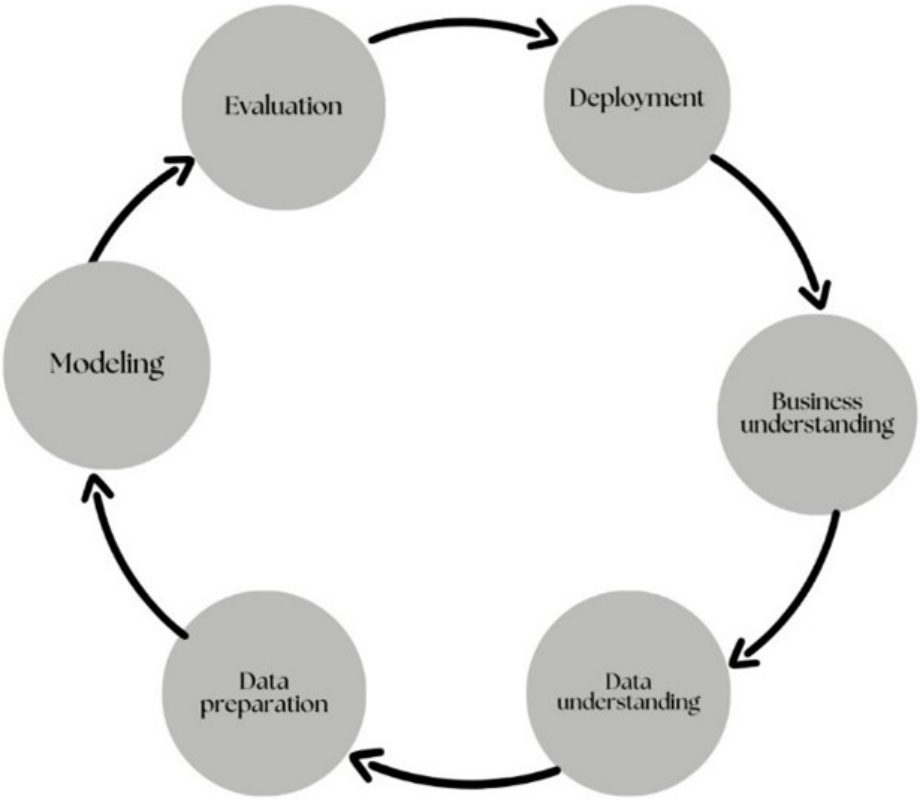

FIGURE 1.1. CRISP-DM methodology

The CRISP-DM methodology in this project has been streamlined into two main stages: *Data* and *Experiments and Results*

The first stage, *Data*, details the data collection process, and the techniques and features selected. This section also emphasizes Data Preparation and Feature Selection, organizing the data and identifying key features, through tables and figures to clarity.

The second stage, *Experiments and Results*, explains the models used, and why they were chosen, and presents both detailed and overall results to demonstrate their effectiveness.

1.4. Document Organization

This dissertation is structured into five chapters. This chapter 1 highlights the background, motivation, and objectives of the research. This section also frames the problem and explains the study's relevance within its field. Chapter 2 is relative to the Related Work which presents a review of the literature and a resume of the studies related to the topic. The Related Work is divided into three subsections: Planning the Review, Conducting the Review and Reporting the Review. Chapter 3 focuses on the Data used

for the research. It includes Data Collection, which explains how the data was obtained, and Data Preparation and Feature Selection, detailing how the data was cleaned and which variables were selected for analysis. Chapter 4 goes on to explain the data through Experiments and Results. The examination of the importance of features begins with subsections dedicated to various feature-importance techniques such as mutual information, random forest permutation-based feature importance, and XGBoost classifier. Followed by the description of the machine learning models used in the experiments, including Neural Networks, Random Forest, XGBoost, SVM, and KNN. Finally, the Results section summarizes the outcomes of these experiments. Finally, Chapter 5 contains the Conclusion and Future Work, collects the findings of the study, discussing the implications of the results and potential areas for future work.

CHAPTER 2

Related Work

The current development in machine learning makes it easier to build models that predict the success of a movie. The latest research done in this area can be divided into two sub-areas: regression research and classification research.

Regression research predicts success through numbers, like revenue, box office or a rating (like IMDB ratings).

In 2012, researchers Deniz Demir, Olga Kapralova, and Hongze Lai [6] conducted a study using Google Trends to predict the IMDb rating of movies. They employed a dimensional reduction technique to choose the most effective features for their model. To predict movie popularity, they utilized supervised learning algorithms, specifically logistic regression, SVM (Support Vector Machines), and multi-layer perception. These algorithms were applied to analyze Google search frequencies to forecast IMDb ratings.

Jeffrey Sparrow developed a movie profit prediction system using machine learning, social network mining, data from social media platforms (Facebook, Twitter, blogs, YouTube), and IMDB and Box-office Mojo. Their primary prediction method relied on measuring audience sentiment, with more optimism correlating with higher profitability [7].

In 2015 Michael T. Lash and Kang Zhao [8] conducted a study to predict the profitability of movies during their early production stages. They used historical data gathered from sources such as Box-Office Mojo and IMDB. They considered what people were saying on social media platforms like Twitter and YouTube, as well as blogs and articles. The focus of their research was on movies exclusively in the USA.

On the other hand, classification research is often focused on building models that predict whether a movie will be a hit or a flop.

In 2010, Asur and Huberman built a model to predict a movie's success using Twitter (a social media platform) [9].

Last year, Sandipan Sahu, Raghvendra Kumar, Hoang Viet Long and Pathan Mohd Shafi [10] published their search that had the goal of predicting the success in binary approach by predicting if a movie was a hit or a flop. For that, they built a K-fold Hybrid Deep Ensemble Learning Model (KHDEM) that delivered a 96% accuracy with more than 4000 movie data collected by IMDB. This research differs from other studies by applying movies with Indian language diversity.

There is also research involving the classification and regression problem. In 2017, an article in the International Journal of Machine Learning and Computing by Prashant Rajput, Priyanka Sapkal, and Shefali Sinha that included Dual Sentiment Analysis (DSA) [11]

that considers the positive and negative comments in a social media platform, Twitter nowadays called X and other techniques like Bag-Of-Words (BOW) and Multivariate Linear Regression. This research aimed to predict the success of a movie and its approximate revenue, and the conclusion was that the accuracy increased as they added features like sequel, genre, star-cast and holiday effect.

2.1. Planning the Review

The methodology chosen in this study was the Systematic Literature Review (SLR) based on the guidelines of the author Kitchenham (2004) [12].

As depicted in Figure 2.1, the structure implemented combined three phases: planning the review, conducting the review, and reporting the review.

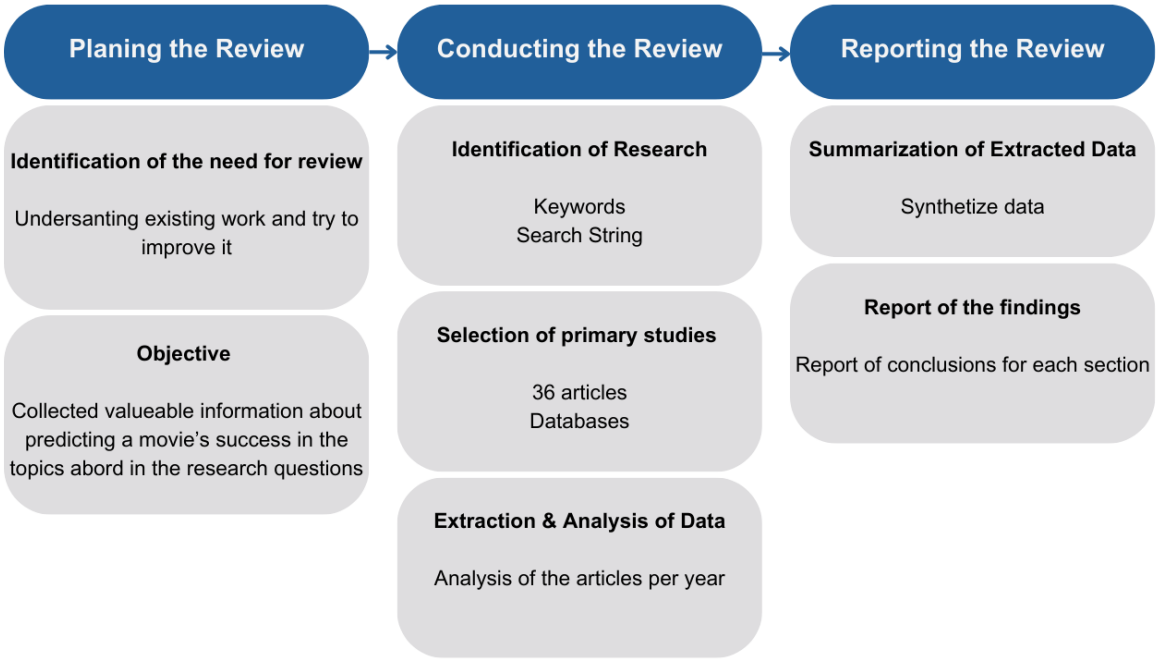

FIGURE 2.1. Phases of SLR

A literature review is essential in research. It is crucial for understanding existing work and improving on it. On the other hand, it also helps in deciding the techniques and features built from a machine-learning model.

Furthermore, a literature review can offer valuable guidance on identifying the gaps in the research done before and these gaps represent areas where more exploration is needed and where to begin.

This is the first phase, planning the review, as we show how we are performing the review. The next sections address the following steps.

2.2. Conducting the Review

This section represents the second phase of the SLR methodology and aims to identify the research, the selection of the studies and the analysis of the extracted data.

To answer the research questions mentioned above, several keywords were chosen and combined into a search string, as shown in Figure 2.2.

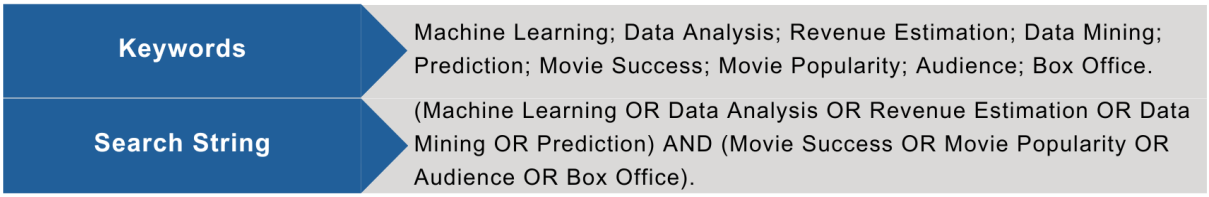

FIGURE 2.2. Description of keywords and search string.

The primary objective of this SLR is to identify the most effective features and the main techniques used to predict a movie's success. Additionally, clarify how the result of the prediction can be helpful for the ones involved in the movie. The studies used were published in 2013. The databases chosen to search for the research are:

- Scopus (https://www.scopus.com);
- IEE Xplore Digital Library (https://ieeexplore.ieee.org/Xplore/home.jsp);
- ACM Digital Library (https://dl.acm.org/);
- Web of Science (https://www.webofscience.com/wos/woscc/advanced-search).

The search process involved several filters to refine the selection of relevant articles. First, we checked if the search string appeared in the full text or all metadata. The second filter focused on narrowing down articles by applying combined keywords specifically to the abstracts. The third filter focused on articles in which the search string matched the title. Next, articles not in English were excluded from the fourth filter. The fifth filter addressed duplicates, aiming to keep unique articles. Lastly, manual filtering was implemented with a critical perspective to retain only the most relevant articles.

TABLE 2.1. Stages of the studies selection process.

Database	Full Text/ All Fields	Abstract	Title	Only English	Delete Duplicates	Manual Filter
Scopus	38772	3309	151	150	129	10
IEEE Xplore Digital Library	21861	471	45	45	40	15
ACM Digital Library	21522	523	33	33	27	6
Web of Science	2353	1592	90	87	77	5
Total	**84508**	**5895**	**319**	**315**	**273**	**36**

After the application of six filters, the analysis of the studies was conducted. The sample extract from the databases is made of 36 articles published between 2013 and 2023 and most of the studies are in 2020.

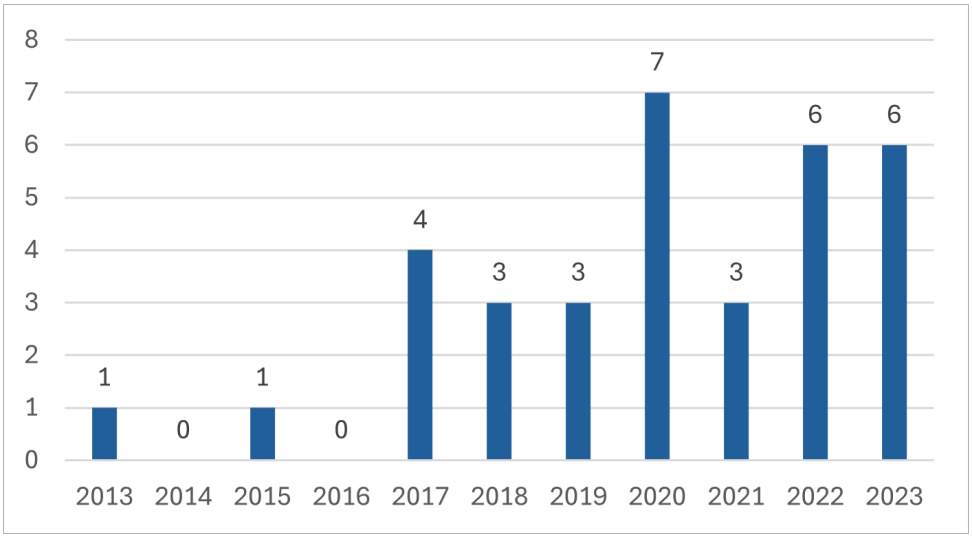

FIGURE 2.3. Articles per year.

2.3. Reporting the Review

In this final phase of SLR methodology, the conclusions of the extracted data are demonstrated and the report of the findings.

Through the analysis of the articles, depicted in Table 2.2, it is possible to conclude that most studies focus on using movie identifiers since they play a crucial role in assessing a movie's success. These identifiers include key details such as the director's name, genre, date of release, and budget, among others. These factors are the initial information known to the audience and significantly influence their movie choices.

The second most used features are extracted from social media before or after (depending on the research) the movie is released. They permit study on the real audience opinion and make more accurate predictions.

Dynamic Features, such as the number of tickets sold on a specific day and location, are also essential variables providing timely information for predicting a movie's success. that gives information in time and provides the possibility to predict the success of the movie.

Furthermore, there is an interesting feature highlighted by Lee *et al.* [9], that focuses on the impact of storytelling on the audience and its relation to a movie's success.

Analyzing the articles with a focus on the databases used as shown in Table 2.3. The conclusions extracted are that the main articles use databases like IMDB, TMDB and Rotten Tomatoes. This observation is logical since these databases provide vital details about movies, including genre, release dates in specific countries, main actors, directors, time of duration, number of views, a specific rating from each platform and some other information.

TABLE 2.2. Summary of the features used in the articles

Features	Articles	Total
Movie identificators (Director name, Main Actors Names, Year of Release, Genre)	(Chen n.d.); (Apala et al. 2013) ;(Cheang et al. 2021); (Zheng et al. 2016); (Nemzer et al. 2020); (Verma and Verma 2020); (Menaga et al. 2023); (Madongo et al. 2023); (Rajput et al. 2017); (Velingkar et al. 2022); (Gupta et al. 2019); (Gupta et al. 2023); (Lakshmi et al. 2020); (Kanitkar n.d.); (Gegres et al. 2022); (Ruwantha et al. 2020); (Sharma et al. 2020); (Ahmad et al. n.d.); (Athira et al. 2020); (Darapaneni et al. 2020); (Nihalaani et al. 2021); (Verma et al. 2019); (Sindhu et al. 2023); (Yi et al. 2022).	24
Social Media Features (likes, comments, followers on actors and directors)	(S A Wijekoon et al. 2017); (Choudhery et al. 2017); (Apala et al. 2013); (Ni et al. 2022); (Rajput et al. 2017); (Velingkar et al. 2022); (Tripathi et al. 2023); (Khan et al. 2022); (Athira et al. 2020); (Sindhu et al. 2023); (Lu n.d.).	11
Dynamic Features	(Jiang et al. 2020).	1
Others	(Lee et al. 2018).	1

Social Media Platform was the second database chosen in the articles because the information collected can be helpful for researchers to understand patterns, and classify comments (good, neutral, bad). These valuable insights also include the possibility of knowing before a movie is released if it is going to be a success based on the audience's opinions.

The third database most chosen is Box Office Mojo which also has a version dedicated to Bollywood movies. This choice is straightforward as Box Office Mojo is a premium API of IMDB. It is worth noting that articles specifying IMDB as their database might implicitly include data from Box Office Mojo, given their close integration.

The database with fewer articles is others, and it refers to databases specific to another country, such as Douban.com for China movies.

2.3.1. Predicting movie success

Machine Learning (ML) has been widely employed to make predictions using various algorithms [13].

In current literature, supervised learning models like SVM, Naïve Bayes, KNN, Random Forest, Logistic Regression, Linear Regression, and Gradient Boosting Models are

TABLE 2.3. Summary of the databases used in the articles

Database	Articles	Total
IMDB or/and TMDB or/and Rotten Tomatoes	(Ruwantha et al. 2020); (Jyang et al. 2020); (Nemzer et al. 2020); (Verma et al. 2020); (Madongo et al. 2023); (Velingkar et al. 2022); (Gupta et al. 2019); (Gupta et al. 2023); (Lee et al. 2018); (Tripathi et al. 2023); (Sahu et al. 2023); (Ankit et al. 2020); (Quader et al. 2017); (Kanitkar n.d.); (Gegres et al. 2022); (Sharma et al. 2020); (Ahmad et al. n.d.);(Athira et al. 2020); (Dhir et al. 2018); (Darapaneni et al. n.d.); (Nihalaani et al. 2021); (Verma et al. 2019); (Sindhu et al. 2023); (Apala et al. 2013); (Cheang et al. 2021); (Menaga et al. 2023);	26
Box-Office Mojo or Box Office India	(Wu et al. n.d.); (Verma et al. 2019); (Gupta et al. 2023);(Apala et al. 2013);	4
Social Platforms (X and/or Youtube and/or facebook)	(Chen n.d.); (Choudhery and Leung 2017) ;(Rajput et al. 2017); (Ruwantha and Kumara 2020); (Jyang et al. 2020); (S A Wijekoon et al. 2017); (Cheang et al. 2021); (Khan et al. 2022); (Tripathi et al. 2023); (Athira et al. 2020)	9
Others	(Yu et al. 2022); (Zheng et al. 2016); (Lu n.d.); (Ni et al. 2022);	4

frequently applied. Some articles applied ensemble methods, which combine the less accurate models to achieve a more accurate result [14]. These models are used to predict either numerical values or classes as target variables [15].

On the other hand, unsupervised learning models such as K-Means Clustering, which helps classify predictions (HIT or Flop, for example), and Generative Adversarial Networks (GAN) [16]. GAN combine two convolution neural networks: generative and discriminate, this technique produces a second output like the target in the training set that will try to beat the first output, while continuously optimizing its parameters, improving the in that way performance.

Additionally, Sentiment Analysis, also known as opinion mining, is also an Machine Learning (ML) technique that is present in most of the studies. It aims to classify social media comments into three categories positive, negative and neutral comments [17]. This technique is helpful for researchers to understand public opinions expressed online.

2.3.2. Features for predicting a movie's success

A study by Linxi Chen in 2021 focused on predicting movie box office performance by utilizing both dynamic and static features, as explored by [18]. Dynamic features involved daily audience data for 120 movies, while static features included factors such as release day, number of release days, schedule ratio, total viewership, audience ratio, average audience per show, average ticket price, total seats provided, seat percentage, attendance rate, user refund rate, and holiday status.

The experiment compared predictions using only static features, multi-scale dynamic features, and single-scale dynamic features over different periods (one day, two days, five days, and ten days after movie release). The results indicated that combining both

dynamic and static features produced the most accurate predictions. The key finding was that dynamic features played a crucial role in short-term box office predictions, and the integration of both feature types yielded the best overall outcomes.

In 2020, Luyao Jiang and Hao Yu conducted a study to predict daily audience data, box office revenue and the number of audiences using 200 films released in three years (2017, 2018 and 2019), where 15 different daily attributes of 30 days are provided for each film. They concluded that the three days after a movie is released is when the film has a bigger audience, and the results are better before the 20 days [19].

According to Table 2.3 the most used databases in the studies selected are: IMDB or/and TMDB or/and Rotten Tomatoes. However, researchers explain also the importance of the other databases.

Studies say that nowadays, social media wields considerable influence on a movie's success or failure. Platforms like YouTube and Twitter are responsible for sharing opinions and comments on all types of subjects worldwide [17]. YouTube, with its constant influx of content, allows users to upload, share and discuss videos all day so it can provide real opinions, making it a valuable source for real-time opinions on movies. It is estimated that 72 hours of videos are uploaded every minute on YouTube, providing a vast pool of data for analyzing audience sentiments.

Twitter, now called X, is a well-known platform that gauges the popularity of key figures like actors, directors and producers involved in a movie. Monitoring discussions on Twitter can offer insights into the buzz surrounding a film and the public's perception of its creators.

Despite the significance of social media data in predicting a movie's success, a challenge arises – this data needs to be collected before the film is released. This timing constraint adds a layer of complexity to utilizing social media insights for predictive purposes.

There are several renowned platforms such as IMDB (Internet Movie Database), TMDB (The Movie Database), Rotten Tomatoes and Box Office Mojo, that serve as key databases for all the movies, offering valuable rankings and information.

IMDB and TMDB are the most used ones since they provide key details about movies, including information about actors, directors, the popularity of both and a ranking given by the people who have seen the film.

Rotten Tomatoes, IMDB and TMDB, offer a combination of critics 'ratings and audience scores [20]. It provides, in that way, a holistic view of a movie's reception.

Box Office Mojo, on the other hand, is notable for its focus on box office value, a critical source of income in the film industry [21]. It tracks and presents information on a movie's financial performance, offering insights into its commercial issues.

The features most utilized in studies are those gathered from historical data collected before the study was carried out.

An example of this approach is evident in a 2020 study made by Ruwantha and Kumara [22], where they selected only tweets of 10 successful and 10 unsuccessful movies. So, in

that way, their analysis was based on the tweets and the result of the success of the movie. This information was then used to train their model and make predictions.

2.3.3. Meaningful insights for predicting a movie's success

ML models, when used to predict the success of a movie, can offer valuable information for the ones who participate in them.

Data mining techniques are useful in various areas such as profit prediction, weather forecasts, simulations, visualization tools and medicinal purposes [23]. These techniques can be used to identify hidden patterns and relationships among various variables.

Predicting a movie's box office goes beyond forecasting a simple number. It reflects a film's performance, indicating the economic value useful for investment companies can generate and drive growth in related sectors such as internet clicks, advertising, and product promotions [24].

When stakeholders invest in a movie, they consider all the variables that will make the film a success. They pay attention to every aspect of the film, adjusting elements like cinematography and effects to match the preferences and characteristics of the audience.

The prediction of movie success involves considerable time and investment. For this reason, the stakeholders need to have fewer uncertainties involved.

CHAPTER 3

Data

In this chapter, we combine the Business Understanding and Data Understanding phases. In this way, the section Data Collection highlights the importance of choosing the right features for predicting the movie's success and describes how they were selected and chosen and gathered. Section 3.2. presents an overview of the database, along with various analyses, to help clarify the data.

3.1. Data collection

The extraction of the data for the dataset started in September 2023, which is why the decision was made for the database to contain films until 2022. It wouldn't make sense to only extract films up to September. Even so, it was necessary to add more variables from different sources throughout the process. As a result, the extraction period took around six months.

The database includes all the data variables useful for the prediction, with movies released between 2012-2022, specifically focusing on Latin-languages (Portuguese, Spanish, Italian, French, Catalan, Galilean, Occidental and Romanian) movies.

Since the study focused on movies that were not used in any other project, it was necessary to apply techniques to collect all the necessary data.

The data was collected from different websites with data extraction capabilities, including:

- TMDB Database (`https://www.themoviedb.org/`);
- Box Office Mojo (`https://www.boxofficemojo.com/`).

Two primary techniques were employed to extract the data from the websites mentioned above. The websites provide APIs to fetch structured data efficiently. Second, web scraping was used, specifically the tool BeautifilSoup [25] on a program basis to extract data from HTML content. Through these extraction techniques, a reliable dataset was ensured for the study.

To proceed with the study, a database was created. The sample of the database variables is presented in Table 3.1.

The main purpose of this study is to predict the movie's success. For that it was necessary:

- Search for reliable data sources;
- Determine what variables were necessary;
- Distinguish which variables are most important for the study;
- Create an efficient and reliable database

TABLE 3.1. Database sample

	movie_name	language	release_date	movie_popularity	actor1	...	worldwide	tmdb_id	vote_average	fair_score
4	Dogman	it	17/05/2018	25.710	Marcello Fonte	...	5080147	944401	7.849	7.801867
45	No Manches Frida 2	es	15/03/2019	46.678	Omar Chaparro	...	26493648	554596	7.968	7.918239

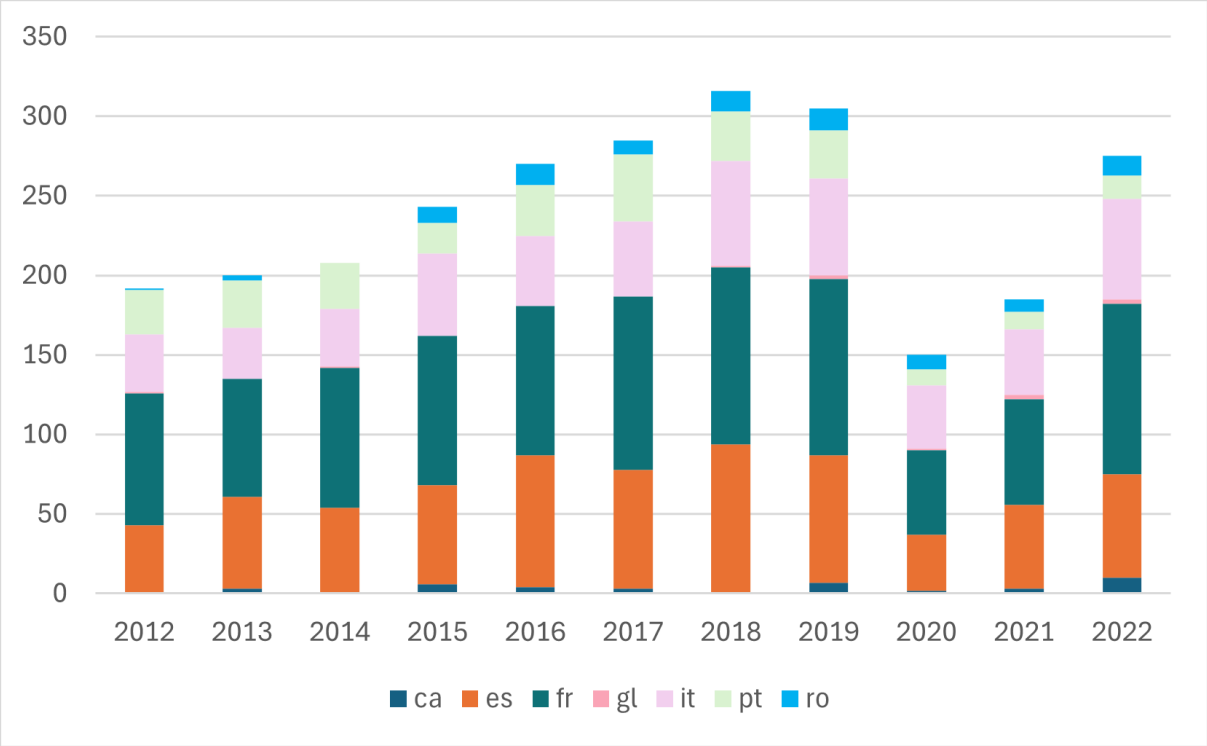

FIGURE 3.1. Demonstration of the movie's languages per year of release

The Figure 3.1 describes the number of movie releases in a given year per the original language of it. As observed over the years movie releases lower are 2020 and 2021. The COVID-19 pandemic created unprecedented challenges for the film industry, requiring new disease mitigation protocols and expert communication to enable a safe return to production [26]. In that way, studies were conducted to allow film production to resume with all the necessary measures.

However, it is important to note that not all the movies were considered in this graph because a small proportion of films do not have the data in a specific order.

3.2. Data preparation and features selection

In this chapter, the aim is to get a deeper understanding of the variables relevant to predicting the success of movies.

The database is composed of four different types of variables.

- String for the variables that include words and give context to the others.
- Data for the date that a movie was released.

- Float for the numbers with decimals.
- Integer for the variables that are integers.

It includes a variety of variables that describe different aspects of movies, such as basic identifiers, popularity metrics, cast and crew information, financial information and voting statistics as exhibited in Figure 3.2.

TABLE 3.2. List of the variables in the dataset

Variables	Variable Type	Description	Source
movie_name	String	Name of the movie.	TMDB
original_title	String	Original title of the movie	TMDB
language	String	Language spoken in the movie	TMDB
release_date	Date	Release date of the movie, in the country where was producted	TMDB
movie_popularity	Float	Popularity of the movie, based on the IMDB rating.	TMDB
actor1	String	Name of the second caracther of the movie.	TMDB
actor2	String	Name of the principal caracther of the movie.	TMDB
actor3	String	Name of the principal caracther of the movie.	TMDB
actor4	String	Name of the principal caracther of the movie.	TMDB
actor5	String	Name of the principal caracther of the movie.	TMDB
actor1_popularity	Float	Popularity of the principal caracter.	TMDB
actor2_popularity	Float	Popularity of the second caracter.	TMDB
actor3_popularity	Float	Popularity of the second caracter.	TMDB
actor4_popularity	Float	Popularity of the second caracter.	TMDB
actor5_popularity	Float	Popularity of the second caracter.	TMDB
director	String	Name of the director.	TMDB
director_popularity	Float	Popularity of the director.	TMDB
producer1	String	Name of the first producer.	TMDB
producer1_popularity	Float	Popularity of the first producer.	TMDB
producer2	String	Name of the second producer.	TMDB
producer2_popularity	Float	Popularity of the second producer.	TMDB
budget	Float	Budget for the production.	TMDB
revenue	Float	Revenue of the movie.	TMDB
domestic	Float	Value of revenue in the domestic country.	Box-Office Movie
international	Float	Value in international level revenue.	Box-Office Movie
worldwide	Float	Value in worldwide level revenue.	Box-Office Movie
tmdb_id	Int	IMDB id for movies.	TMDB
vote_average	Float	Vote average.	TMDB
vote_count	Int	Vote count.	TMDB
fair_score	Float	Fair score calculated based on other variables.	Manual

In order to establish the database variables it was necessary first to substitute the non-numeric values for "NaN", i.e. Not a Number and then assign a type to all variables

The movie's database has 9084 lines (i.e. movies) and 30 columns (i.e. features) divided into different languages.

3.2 shows how the languages are divided, and according to this, all movies in the database have a language attributed to them.

In an attempt to reduce the dimension of the database, the fair score variable was created. This uses the calculations provided by IMDB.[1]

$$W = \frac{R \cdot v + C \cdot m}{v + m}$$

where:

- R is the mean rating for the movie,
- v is the number of votes for the movie,

[1] https://en.wikipedia.org/wiki/IMDb

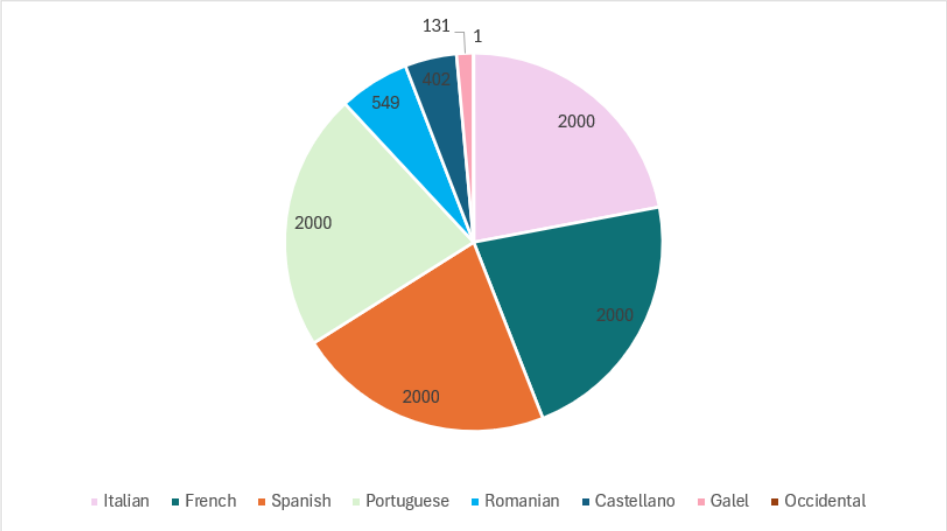

FIGURE 3.2. Demonstration of the distribution of the languages

- m is the minimum votes required (i.e the value of the first quarter of the vote count variable),
- C is the mean vote across the whole report (i.e. the mean of vote average variable).

Creating and utilizing a fair score can mitigate bias and also give a better understanding of the actual vote.

The heat map as shown in Figure 3.3 was constructed using the numerical variables, i.e. float 64 and int 32, except TMDB ID, since it is a variable that was only used to help extract the data, and year release because it is the year in which the movie was released and is of no interest for the correlation calculation.

Analyzing the heat map, it is possible to verify:

- The biggest correlation (0.99) is between the features international and worldwide because both variables represent similar values.
- There is a high correlation (0.70) between the fair score and vote average variables, which is explained by the fact that the fair score variable was calculated using the number of average votes.
- It is also important to note the correlation between the feature revenue and international (0.59) and worldwide (0.61).
- It is worth noting the low correlation (0.23) between movie popularity and fair score, in that way both variables represent the same thing in theory. However, the fair score is a more reliable variable, since it was created based on other variables in the database and not just extracted from the TMDB API.

In this chapter, the data analysis performed was crucial to understanding the key patterns and variables that influence the success of Latin-language films. Additionally,

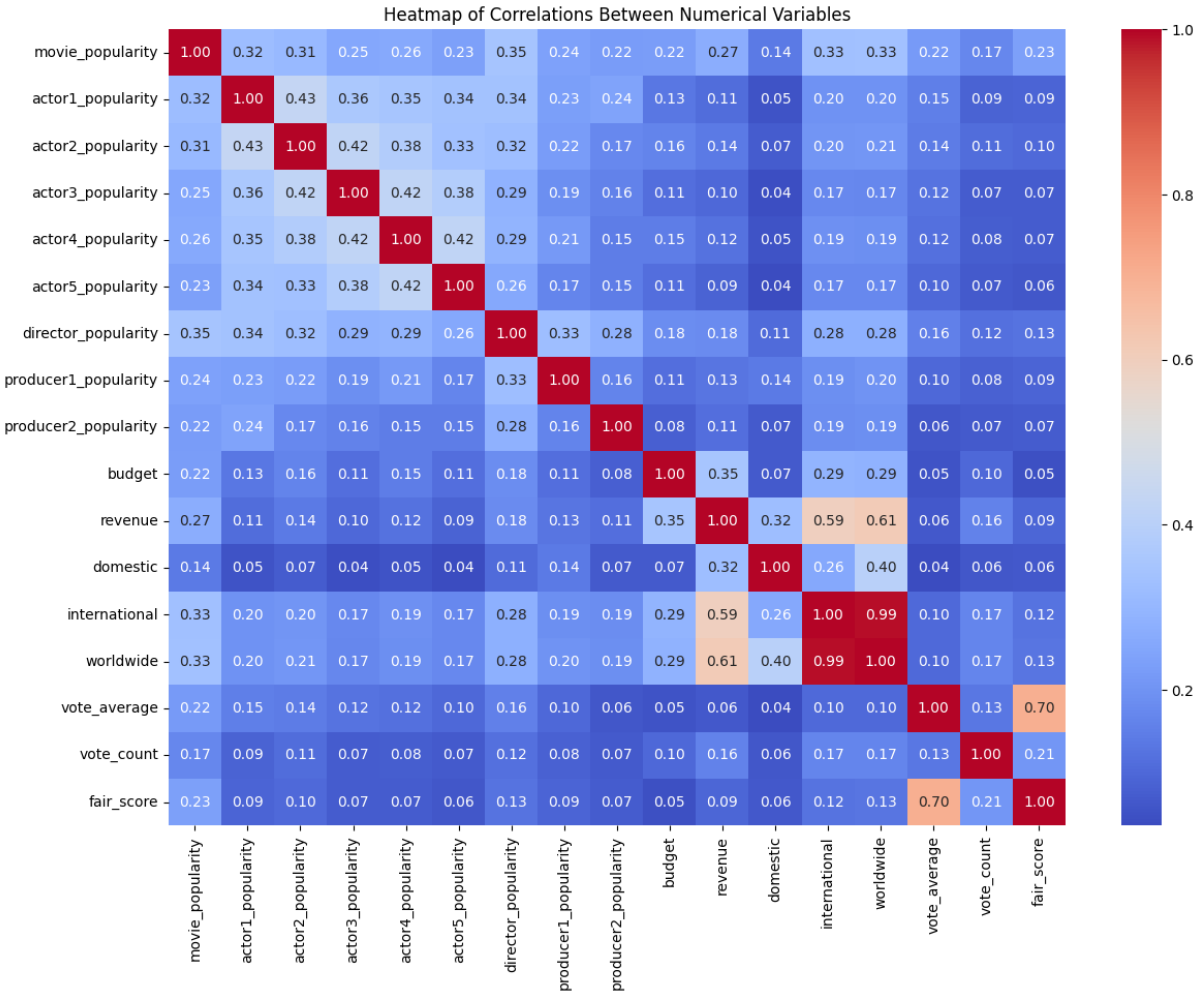

FIGURE 3.3. Demonstration of the Heat-map of Correlations between numeric Variables

the analysis offered a clear view of the relationships between variables, enabling the selection of the most appropriate modelling approaches to maximize the accuracy of success predictions in the film industry.

CHAPTER 4

Experiments and Results

This section includes the remaining phases of the CRISP-DM methodology. The chapter begins with Data preparation, where missing values and outliers are managed. Section 4.2. focuses on Modeling, in this way tests multiple machine learning models to find the best fit for the data and problem. Evaluation of the models is also conducted in this section, using metrics like accuracy, precision, and recall. Finally, Section 4.3 identifies the best-performing model and interprets its results concerning the project's goals.

The variable fair score was transformed into a binary target variable by dividing its range into two intervals and assigning labels [0, 1] to each interval.

This study was defined as a classification problem where the target variable is a fair score and aims to predict if the feature is a 0 "Flop" or 1 "Hit".

TABLE 4.1. Missing values per feature

Features	Missing Values
actor1	602
actor2	956
actor3	1301
actor4	1669
actor5	2135
actor1_popularity	602
actor2_popularity	956
actor3_popularity	1301
actor4_popularity	1669
actor5_popularity	2135
director	116
director_popularity	116
producer1	3756
producer1_popularity	3756
producer2	5924
producer2_popularity	5924
budget	8230
revenue	8376
domestic	8620
international	6482
worldwide	6454
tmdb_id	15
vote_average	1527
vote_count	1527

After analyzing the database, it was necessary to implement an action plan to resolve the missing values found. It is important to break them down by feature as shown in Table 4.1.

Once the large number of missing variables was discovered the features budget, revenue and domestic were deleted. To ensure that as much data as possible was being used and given that the worldwide and international variables had a 99% correlation, an average revenue variable was created which aims to use one of the variables when they exist in the row or the sum of both followed by the average when we have both for the same film.

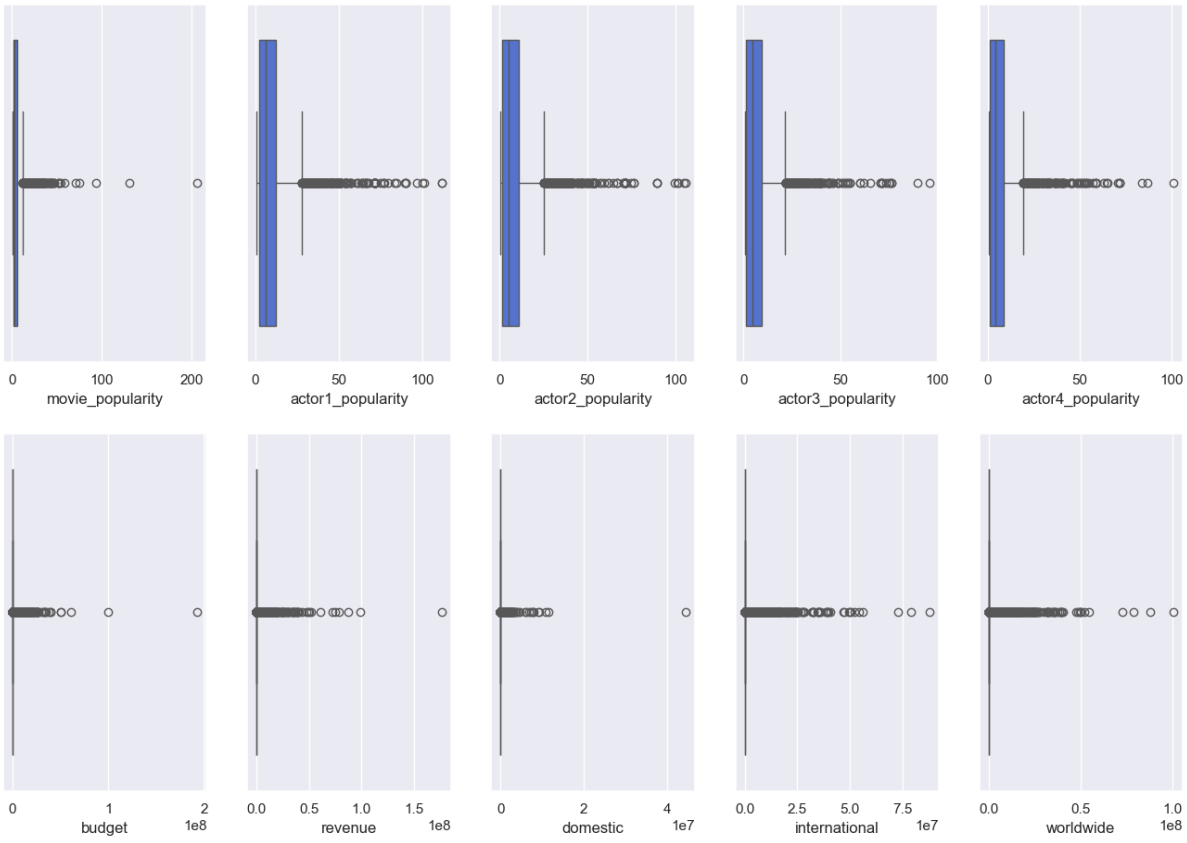

FIGURE 4.1. Numeric Variables'Box Plots Part I

Figures 4.1 and 4.2 contain box plots representing the distribution of various numeric variables in the dataset. The features: Movie Popularity, Actor Popularity (1 to 5), Director Popularity and Producer Popularity (1 and 2) show a similar pattern, where most values are concentrated at the lower end, with a few outliers representing significantly more popular movies, actors, director or producers depending on the graph of the variable to be analyzed. The graph that represents the budget reveals a significant range, with many small budgets and a few very high outliers. Distribution of revenues is similar to budget, where most values are relatively low, and a few projects have very high revenue outliers. Domestic Revenue'Box Plot indicates low values for domestic revenue, with some large outliers. International and Worldwide variables are also heavily skewed with a few significant outliers. The Vote Average is the feature where the majority of vote averages are concentrated in the mid-range, between 5 and 7, with a more symmetric distribution.

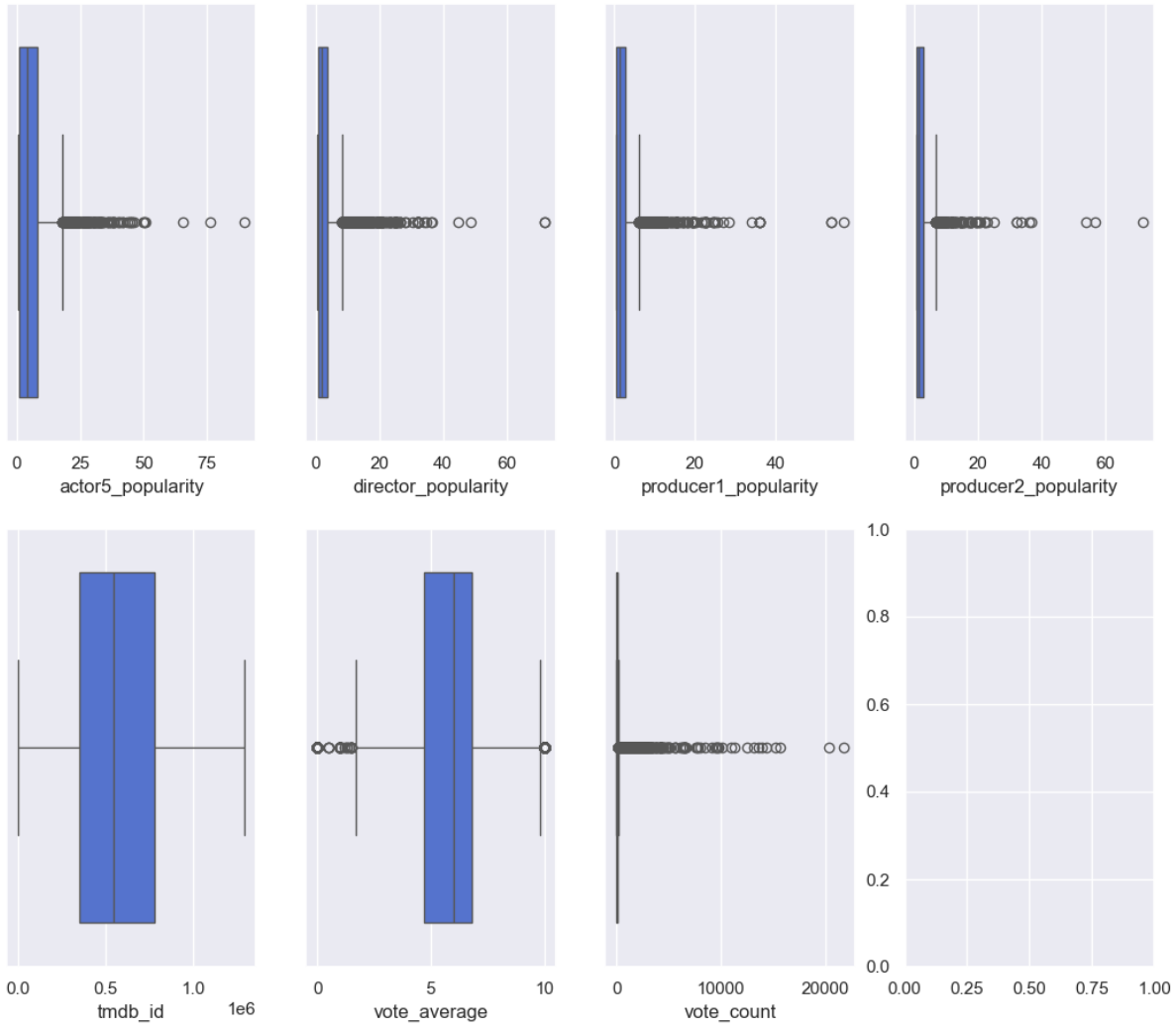

FIGURE 4.2. Numeric Variables'Box Plots Part II

Finally, the Vote Count is highly skewed, with most values on the lower side but with some films receiving tens of thousands of votes.

The target variable chosen is **fair score** since it is the most reliable because it depends on two other variables: vote count and vote average, so these features were also left out of the training and testing of the model.

The correlation matrix was created with the variables that will be used to build the model and the graph in Figure 4.3 shows:

- Higher correlations after data processing;
- Popularity variables related to actors and directors tend to have moderate positive correlations with each other, ranging from around 0.5 to 0.7.

After that, all the variables chosen were normalized using a MinMaxScaler from Sklearn, which scales the data to a range between 0 and 1 (by default).

In order to proceed with the study, we needed to split the data. We used the train test split function to divide the dataset into training 80% and testing 20% subsets. By setting the stratify parameter to y, we ensured that the class distribution in the target variable

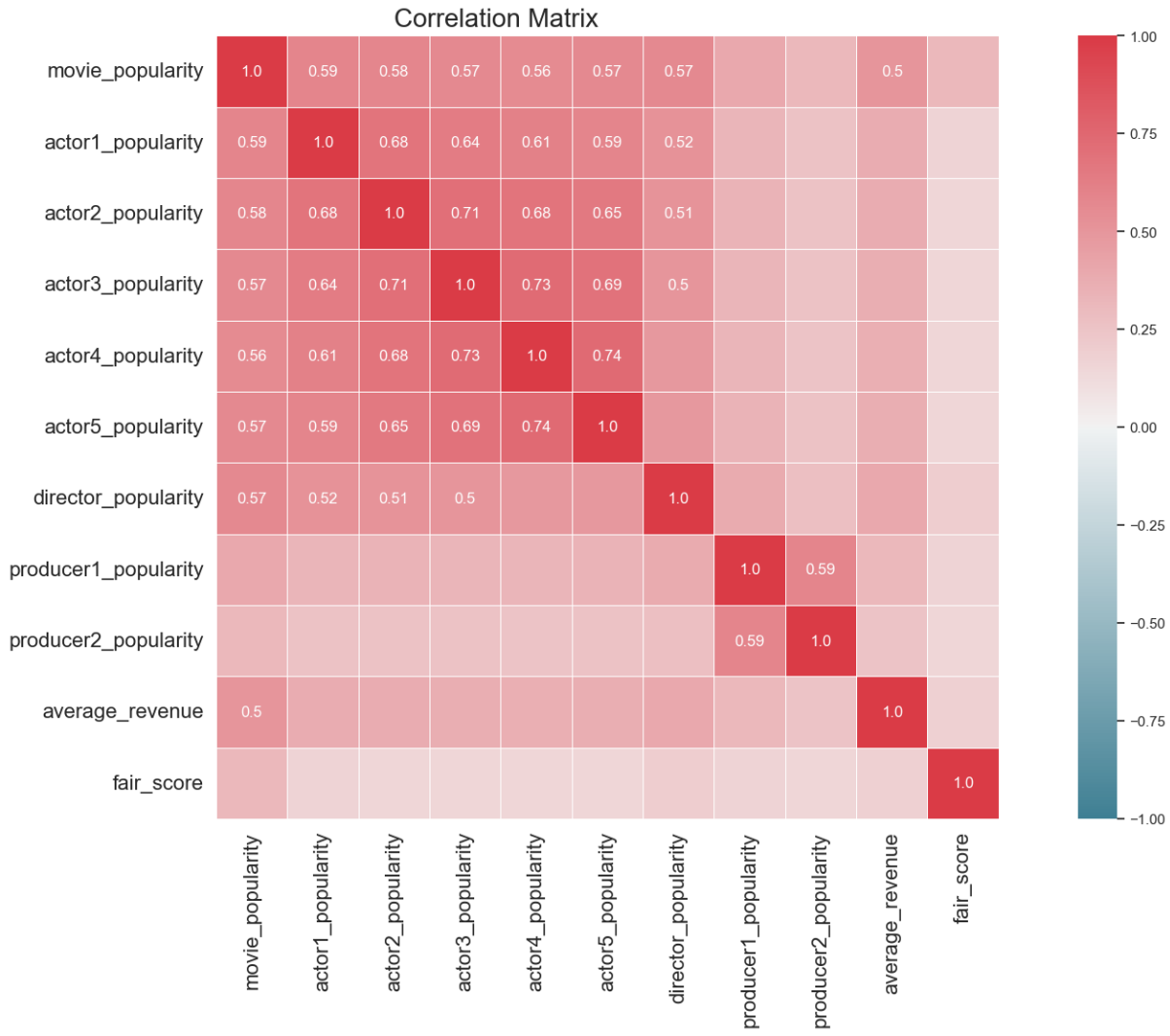

FIGURE 4.3. Correlation Matrix

was maintained in both the training and validation sets. This is especially important for datasets with imbalanced classes. Since the random state was not specified, the split will differ each time the code is run, allowing for slight variability in results across different executions.

4.1. Importance of the features

This chapter highlights the importance of choosing the right features, and how they can be important to predict a better outcome.

4.1.1. Mutual Information

Mutual information is a powerful tool for feature selection in high-dimensional datasets, addressing the curse of dimensionality and improving classifier performance [27].

In Figure 4.4, both Occidental and Galician languages show no Mutual Information Score.

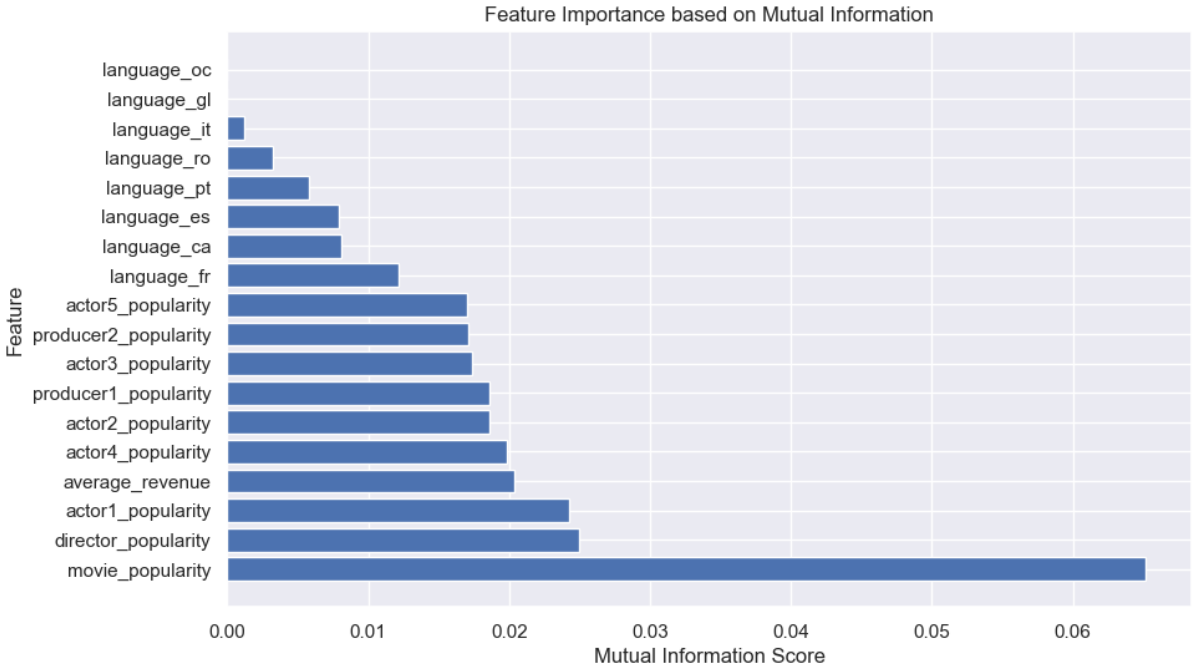

FIGURE 4.4. Feature Importance based on Mutual Information

4.1.2. Random Forest Permutation Based Feature Importance

The Random Forest Permutation Feature Importance approach generates 'feature importance scores' that can be used to create a list of features ranked according to their importance. This allows the modeler to focus on the most important features and it is the key quality of the machine learning model [28].

Permutation importance is a method for evaluating feature relevance in Random Forest models by measuring the decrease in model performance when a feature is randomly shuffled [29].

Figure 4.5 shows that the most important feature is movie popularity when permuted, it causes the largest decline in model performance, as indicated by the highest bar on the chart. This suggests that the popularity of a movie is the strongest predictor in the dataset.

Average revenue and director popularity are also important, but their impact is considerably smaller than movie popularity. They play a notable role in the model's predictive power but are less dominant.

Features related to actors, such as Actor1 Popularity, Actor5 Popularity, and Actor4 Popularity, have moderate importance, indicating that the popularity of certain actors contributes to the model's predictions, but not as strongly as the overall movie popularity or revenue.

The other features such as Producer Popularity and various language variables (e.g., language ro, language it, language fr) have much smaller bars. These features have a minimal impact on the model's performance when permuted, suggesting they are less important in predicting the outcome.

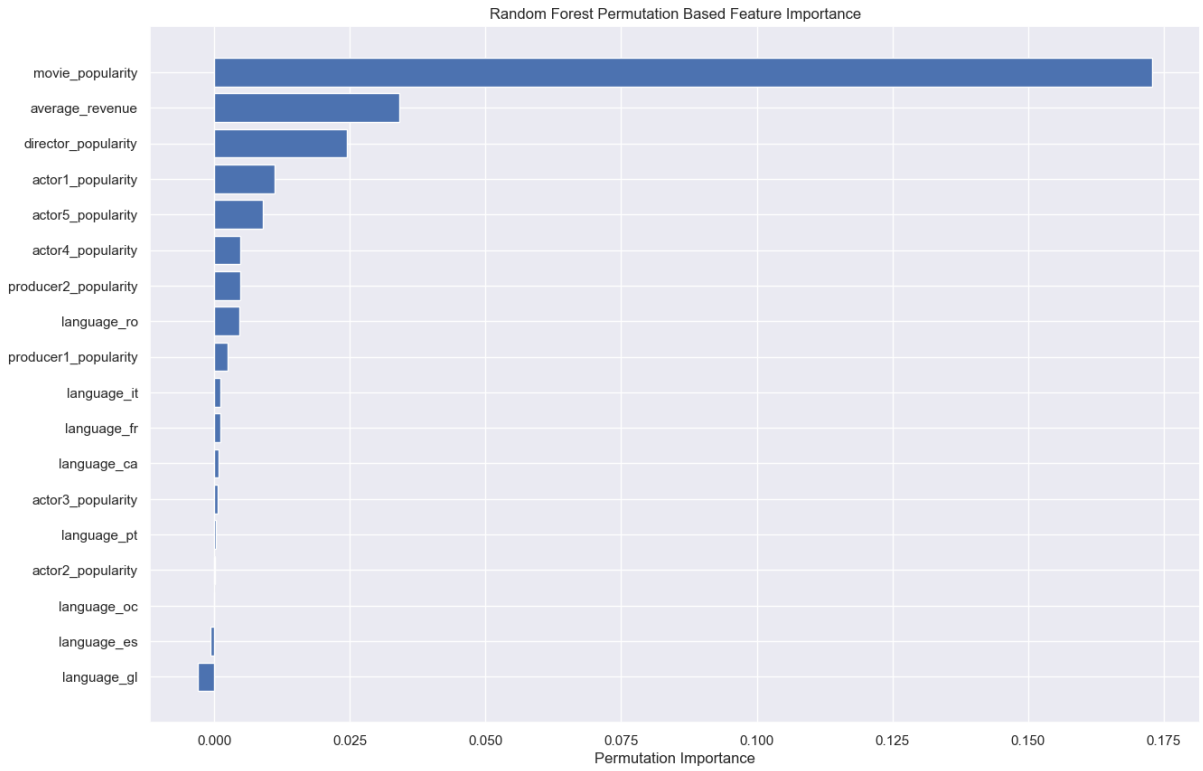

FIGURE 4.5. Random Forest Permutation Based Feature Importance

A few language variables, such as language gl and language es, have almost zero importance, indicating that these features do not contribute significantly to the model's predictions.

4.1.3. XGBoost Classifier

Feature selection algorithms and hyper-parameter optimizations need to be considered during model training [30].

The Grid Search CV was used to find the best hyper-parameters to apply to the XGBoost algorithm. In that way, Figure 4.5 the model performs best with a few features (around 2-3), suggesting that the most important variables are highly predictive and others may introduce noise or multicollinearity.

This graph highlights the importance of feature selection. Adding too many features, especially irrelevant or redundant ones, can degrade the model's performance.

Despite the two selected variables giving a good performance, other features might still have valuable information that contributes to the overall prediction.

4.2. Results

This chapter expands on the previous by applying predictive models to the analyzed data. Each model is briefly outlined. Based on the literature review in Chapter 2, the most effective models were selected. These five models can be classified into four categories.

The models to be analyzed are:

- Neural Networks;

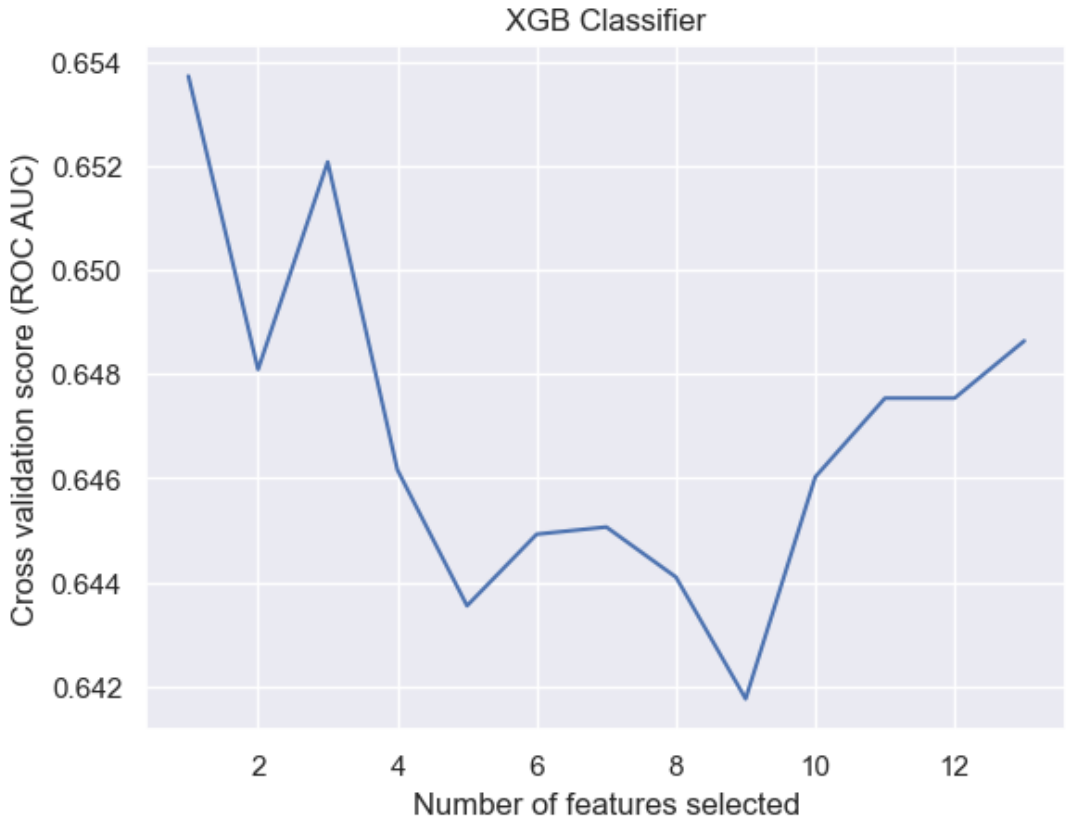

FIGURE 4.6. Number of features selected and the cross-validation score (ROC AUC)

- Random Forest;
- XGBoost;
- Support Vector Machines (SVM);
- K-Nearest Neighbors (KNN).

Performance metrics play a crucial role in the development, selection, and evaluation of machine learning models [31]. The following metrics provide valuable insights that help in making informed decisions regarding the appropriate model by presenting the results for each one.

The following metrics will be employed in assessing the models, which are calculated based on the parameters:

- TP : True Positives,
- TN : True Negatives,
- FP : False Positives,
- FN : False Negatives.

Accuracy is the most simple and common measure derived from the confusion matrix. The Accuracy formula calculates the proportion of correct predictions by dividing the sum of True Positives and True Negatives by the total number of all cases in the confusion matrix [32]. In simpler terms, Accuracy tells us the chance that the model will correctly predict the class of a randomly chosen unit. The formula for accuracy is:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

The Recall metric represents the fraction of True Positive elements divided by the total number of positively classified units. In particular False Negative are the elements that have been labeled as negative by the model, but they are positive [32]. The formula for Recall or True Positive Rate is:

$$Recall = \frac{TP}{\text{Actual Positive instances}} = \frac{TP}{TP + FN}$$

Precision measures how accurate the positive predictions are. It looks at how many of the reviews predicted as positive are positive, based on the total number of reviews that the model labelled as positive [33]. The formula for Precision is:

$$Precision = \frac{TP}{\text{Predicted Positive instances}} = \frac{TP}{TP + FP}$$

The F-measure, a balanced performance metric, considers both recall and precision. The commonly used F1-measure is the harmonic mean of precision and recall [32]. The formula for the F1-score is:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2TP}{2TP + FP + FN}$$

In every section, that represents a model experimented will be depicted in figures, the classification report, for a better understanding of the performance of the models. The metric measures will focus on the evaluation of potential overfitting between the results from the training and validation. A confusion matrix will also be provided to highlight the most and least accurately predicted classes.

4.2.1. Neural Networks

Artificial Neural Networks (ANN) are mathematical computational models inspired by the workings of biological neural networks, used in computing and machine learning.

The main reason for choosing this model was the capability to handle complex data relationships and process large amounts of information quickly [34].

The specific architecture used, was the Multilayer Perceptron (MLP), a classifier that consists of an input layer, one or more hidden layers, and an output layer as seen in Figure 4.7. Each layer has interconnected nodes (neurons), and data moves forward

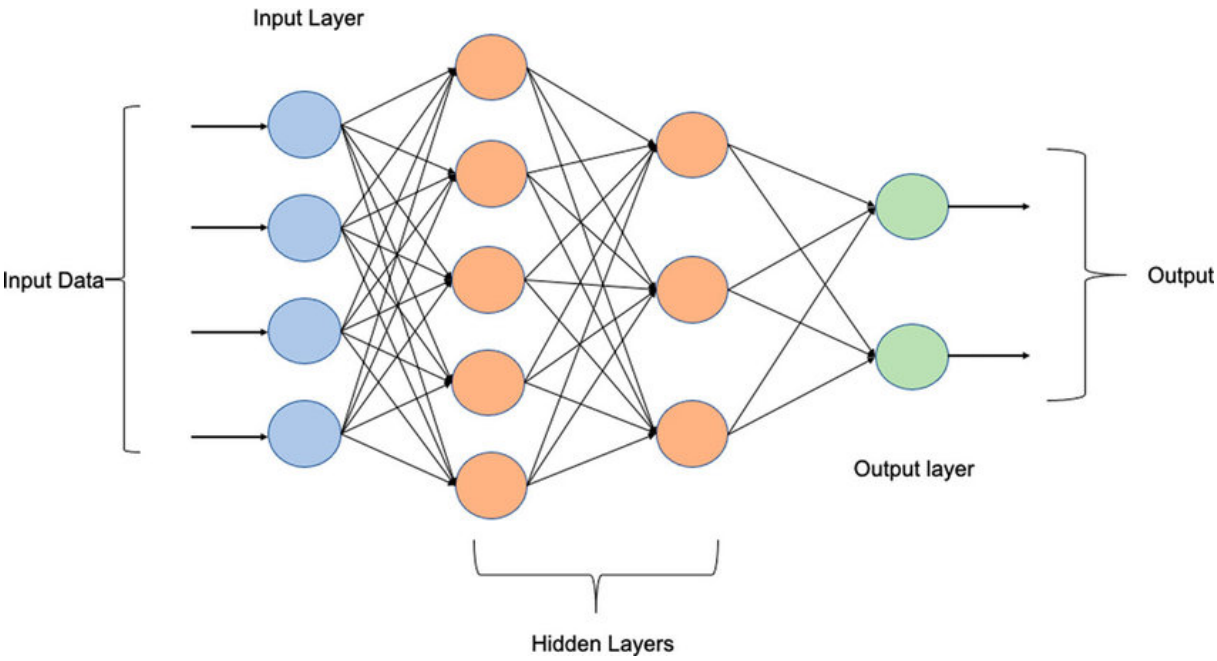

FIGURE 4.7. MLP Architecture

through the network during both training and prediction. In this scenario, a 0,001 learning rate was used for stable learning, and the activation function selected was the Rectified Linear Unit (ReLU) because of this effectiveness.

This architecture was chosen since it is easy to implement and works well with structured data [35]. However, it may have difficulty handling complex, high-dimensional data which is not an issue in this case.

TABLE 4.2. Classification Report of Neural Networks

	Precision	Recall	F1-score	Support
0	0,636	0,638	0,637	859
1	0,674	0,672	0,673	958
accuracy	-	-	0,656	1817
macro avg	0,655	0,655	0,655	1817
weighted avg	0,656	0,656	0,656	1817

In the classification report shown in Table 4.2 it is possible to see a difference between the class 0 and 1, i.e. "Flop" or "Hit". The metrics reveal that the model predicts better the number of class 1 corresponding to the movie's Hits.

Other metrics such as accuracy mean that the model correctly predicted the class of 66% of the instances. Macro and weighted averages of precision, recall, and F1-score are similar approximately (0,66), which indicates that the distribution between classes might be relatively balanced.

Table 4.3 provides key performance metrics for the Neural Network model, comparing the training and validation dataset. In this way, the table shows that:

TABLE 4.3. Metric Measures of Neural Networks

Accuracy - Training	0,66
Accuracy - Validation	**0,656**
Recall - Training	0,658
Recall - Validation	**0,655**
Precision - Training	0,659
Precision - Validation	**0,655**
F1 score - Training	0,659
F1 score - Validation	**0,655**

- **Accuracy** - The consistent of the accuracy indicates that the model performs similarly on both datasets.
- **Recall** - The slight decrease in recall during validation (0,658 vs. 0,655) shows the model's ability to identify positive cases generalizes well to unseen data.
- **Precision** - The training precision is higher than the validation, meaning the model maintains reasonable precision in generally.
- **F1-score** - The F1-scores are all identical in both cases which can indicate the balance between precision and recall remains stable across datasets, confirming that the model is well-calibrated.

The model performs similarly on both the training and validation datasets, showing balanced precision and recall with no major signs of overfitting.

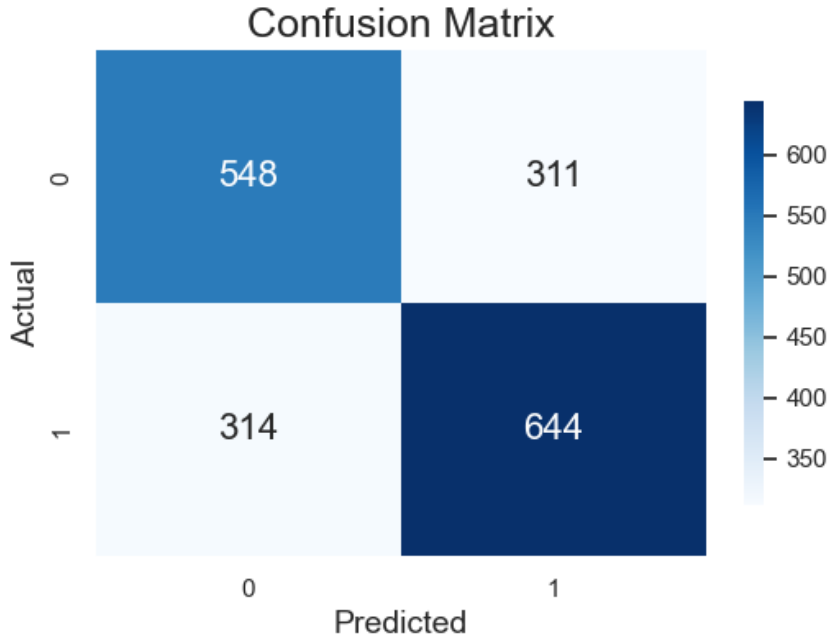

FIGURE 4.8. Confusion Matrix Neural Networks

The Confusion Matrix in Figure 4.8 shows that the highest values 644 and 548 correspond to the instances where the model correctly predicts Positive and Negative classes.

This suggests that the model's performance is in line with expectations, particularly showing better results in identifying the "Hit" class.

4.2.2. Random Forest

The Random Forest algorithm is a Decision Tree-based classifier. It selects the best classification tree as the classification algorithm of the final classifier by voting [36].

A Random Forest algorithm consists of multiple decision trees working together. The randomness comes from two main aspects: (1) each decision tree is trained on a randomly chosen subset of the entire dataset, and (2) at each split in the tree, only a random subset of features is considered. This randomness helps reduce similarities between the trees, ensuring that each one follows different decision paths and has a unique structure, which improves the overall accuracy and robustness of the model [37].

This classifier generates multiple decision trees using randomly chosen subsets of the training data. It then combines the predictions from these different trees by taking a majority vote to determine the final classification for the test object [38]. In this scenario, was also used a maximum depth of 16 to control the overfitting.

TABLE 4.4. Classification Report of Random Forest

	Precision	Recall	F1-score	Support
0	0,654	0,567	0,607	859
1	0,653	0,731	0,69	958
accuracy			0,653	1817
macro avg	0,653	0,649	0,648	1817
weighted avg	0,653	0,653	0,651	1817

In the Classification Report shown in Table 4.4 it is possible to show a difference between the class 0 and 1, i.e. "Flop" or "Hit". The metrics reveal that the model predicts better the number of class 1 corresponding to the movie's Hits with the exception of the precision metric that it is equal in both situations.

Other metrics such as accuracy mean that the model correctly predicted the class of 65% of the instances. Macro and weighted averages of precision, recall, and F1-score are the same (0,65), which indicates that same as Neural Network model the distribution between classes might be relatively balanced.

Table 4.5 provides a objective comparison between the training and the validation dataset for the Random Forest model. In this way, the table shows that:

- **Accuracy** - The training accuracy is higher (0,796) than the validation accuracy (0,653), which indicates the model performs better on the training data compared to the unseen validation data. Which could be a sign of subtle overfitting.
- **Recall and Precision** - Recall and Precision are lower on the validation set (0,649 and 0,653) than on the training set (0,794 and 0,796), indicating that

TABLE 4.5. Metric Measures of Random Forest

Accuracy - Training	0,796
Accuracy - Validation	**0,653**
Recall - Training	0,794
Recall - Validation	**0,649**
Precision - Training	0,796
Precision - Validation	**0,653**
F1 score - Training	0,795
F1 score - Validation	**0,648**

the model is likely overfitting and has more false positives and/or false negatives when applied to unseen data.

- **F1-score** - The F1-score on the validation set (0,648) is lower than on the training set (0,795), confirming that the model does not generalize well to new data and is likely overfitted to the training set.

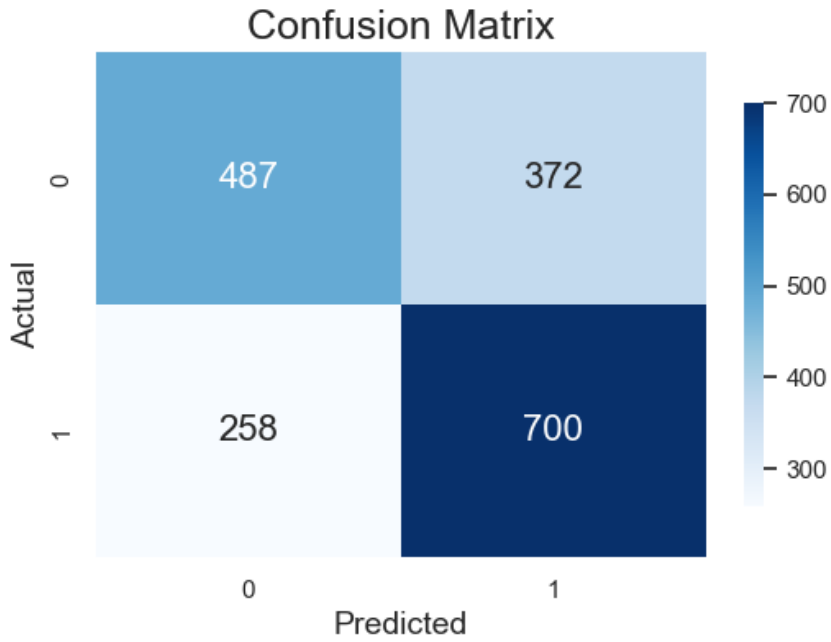

FIGURE 4.9. Confusion Matrix Random Forest

The Confusion Matrix in Figure 4.9 shows that the highest values 700 and 487 correspond to the instances where the model correctly predicts positive and negative classes. This suggests that the model's performance is in line with expectations, particularly showing better results in identifying the "Hit" class. It is also important to note that the lowest value is 258 and corresponds to False Negatives. These are instances where the actual class was positive, but the model incorrectly predicted it as negative. This is also called a Type II error or a missed detection.

4.2.3. XGBoost

XGBoost is a highly efficient gradient-boosting algorithm known for its speed and performance. It creates an ensemble of weak models sequentially, intending to minimize prediction errors at each step [39]. The algorithm uses regularization methods to avoid overfitting and can manage missing data.

XGBoost enhances split-finding efficiency, which is the most time-consuming part of decision tree construction. It uses pre-sorted, compressed column-based data structures, sorting attributes only once. Also, a parallel computation of split candidates for each attribute. The algorithm applies a method that tests only a subset of candidate splits (based on data percentiles), reducing computational complexity [40].

This method can adapt to any type of data and gives good results even in a situation where the number of variables exceeds the number of observations [41].

In this case, a 0.1 learning rate was used to allow the model to learn more gradually and achieve better accuracy by avoiding overfitting. Was also set, the number 3 as the maximum depth of each tree. Additionally, the col-sample by-tree was set to 0,8 meaning that 80% of the features are sampled for building each tree. All, these parameters were chosen in order to reduce the overfitting and improving the performance of the model.

TABLE 4.6. Classification Report of XGBoost

	Precision	Recall	F1-score	Support
0	0,661	0,584	0,62	859
1	0,662	0,731	0,695	958
accuracy	-	-	0,662	1817
macro avg	0,661	0,658	0,657	1817
weighted avg	0,661	0,662	0,659	1817

In Table 4.6, the Classification Report highlights a clear difference in the model's performance between the two classes, "Flop" (class 0) and "Hit" (class 1). The metrics show that the model is better at predicting Hits (class 1) compared to Flops (class 0).

The model's overall accuracy is 66%, meaning that it correctly classified 66% of all the instances, whether Flop or Hit.

The macro average and weighted average of precision, recall, and F1-score are all approximately 0,66, suggesting that the distribution between the two classes is relatively balanced. This pattern is similar to what was observed in the performance of a Neural Network model, where both classes were treated fairly equally in terms of the number of predictions.

Table 4.7 presents important performance metrics for the XGBoost model, comparing its results on both the training and validation datasets. It highlights how the model performs across these two sets, allowing for a clear comparison of its behavior on the

TABLE 4.7. Metric Measures of XGBoost

Accuracy - Training	0,686
Accuracy - Validation	**0,662**
Recall - Training	0,686
Recall - Validation	**0,662**
Precision - Training	0,686
Precision - Validation	**0,661**
F1 score - Training	0,685
F1 score - Validation	**0,659**

data it learned from versus how well it generalizes to unseen data. The table effectively demonstrates the model's performance across both datasets. Therefore the table shows:

- **Accuracy** - The model's accuracy is 0,662 on the validation set and 0,686 on the training set, since this values are quite close, this could indicate that the model is learning well from the training data and it is maintaining the same level of performance on unseen data.
- **Recall** - Recall values are similar, with a small decrease on the validation set (0,662 compared to 0,686) in training set. This consistency suggest that the model is relatively stable on correctly identifying positive cases.
- **Precision** - Precision also remains consistent between the validation set (0,661) and the training set (0,686), which could be interpreted as a good sign, since the model is not overly influenced by the training data.
- **F1-score** - The F1-score also shows a minimal difference in the both scenarios. Which means that the model is well-balanced and able to generalize, without overfitng or underfiting.

The confusion matrix in Figure 4.10 shows that the model has 502 True Negatives and 700 True Positives, meaning it correctly classified many instances for both classes. However, there are 357 False Positives (incorrectly predicting "Hit") and 258 False Negatives (incorrectly predicting "Flop"), which could indicate some errors in both types of predictions.

4.2.4. Support Vector Machines (SVM)

SVM was chosen because it is particularly effective for binary classification problems. The key idea behind SVM is to find a decision boundary (margin) that maximizes the separation between different classes in a high-dimensional space, called the feature space. This approach minimizes classification errors and improves generalization, especially when dealing with small datasets. During training, SVM identifies a small subset of data points, known as support vectors, representing the classification task [42].

SVM use little computational power to produce considerable accuracy. This algorithm aims to find a hyperplane that distinctly classifies the data points in an N-dimensional

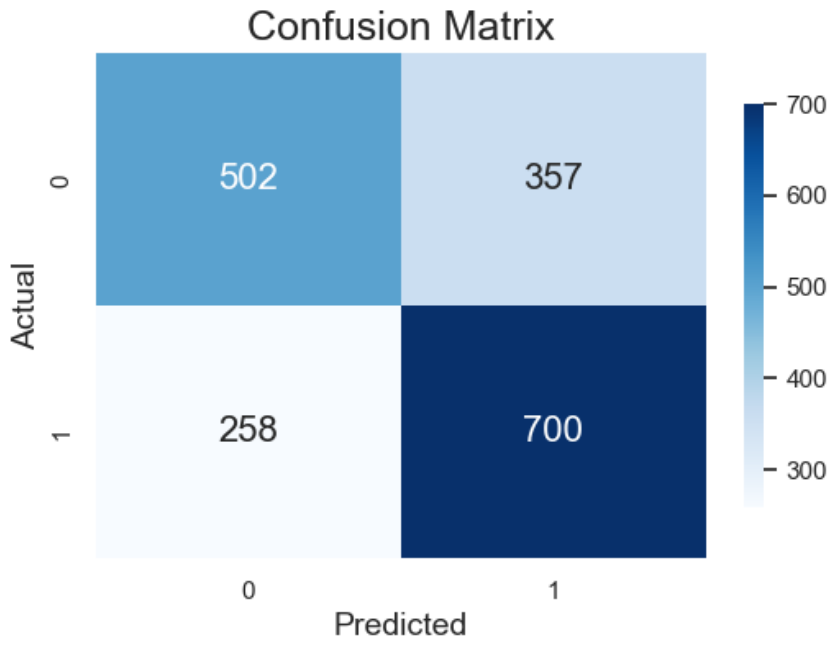

FIGURE 4.10. Confusion Matrix XGBoost

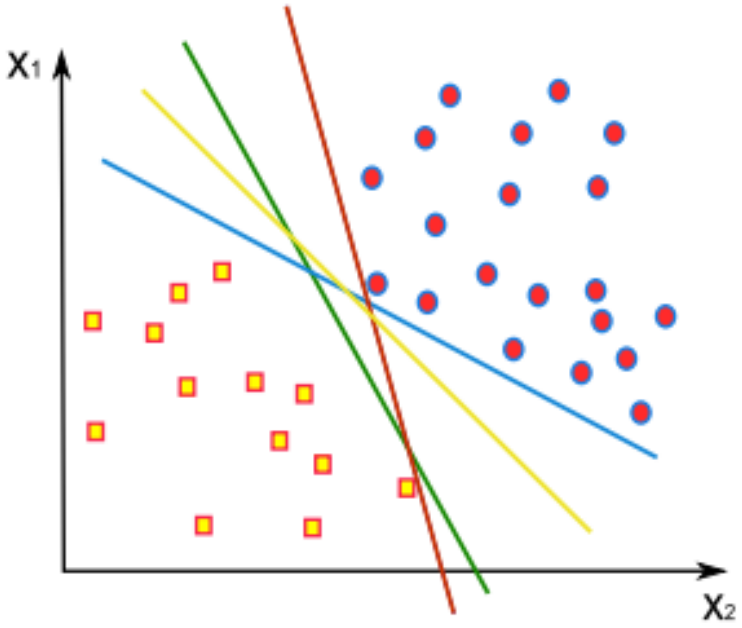

FIGURE 4.11. SVM - Separation hyperplanes

space, where N is the number of features. A hyperplane is a decision boundary that classifies data points such that those on one side of the hyperplane belong to one class, while those on the other belong to another. Some hyperplanes make it possible to separate any two given classes, as shown in Figure 4.11. SVM finds such a plane that the distance between data points of both classes is maximum. This is called maximum margin and can be determined using the data points closest to the hyperplane. Such data points are called support vectors, and they influence the orientation and position of the hyperplane.

The idea behind maximizing margin distance is that it adds to the expectation that test data points can be classified more accurately and confidently [4].

Support Vector Classification (SVC) is a kernel-based machine learning algorithm, derived from the SVM framework. It is designed to effectively group data points into clusters using a two-step process: training and labelling [43]. This type of algorithm can overcome the limitations of traditional clustering methods. SVC can handle clusters of complex shapes and does not require the number of clusters to be specified in advance. This makes it a powerful tool for clustering tasks, as it adapts to the natural structure of data without needing prior initialization.

TABLE 4.8. Classification Report of SVM

	Precision	Recall	F1-score	Support
0	0,622	0,593	0,607	859
1	0,65	0,677	0,663	958
accuracy	-	-	0.637	1817
macro avg	0,636	0,635	0,635	1817
weighted avg	0.637	0,637	0,637	1817

Table 4.8 represents the SVM's Classification Report that shows almost identical performance for both classes, although it performs slightly better at predicting class 1 (Hits) than class 0 (Flops).

The model achieves an overall accuracy of 64%, with precision, recall, and F1-scores of 64% also.

Since the dataset is fairly balanced between Flops (859) and Hits (958), the model shows slightly better performance in identifying Hits than Flops, as evidenced by higher Recall and F1-score for class 1.

Additionally, the macro and weighted averages are close suggesting the model is consistent in how it handles predictions across both categories.

TABLE 4.9. Metric Measures of SVM

Accuracy - Training	0,619
Accuracy - Validation	**0,662**
Recall - Training	0,616
Recall - Validation	**0,635**
Precision - Training	0,617
Precision - Validation	**0,636**
F1 score - Training	0,616
F1 score - Validation	**0,635**

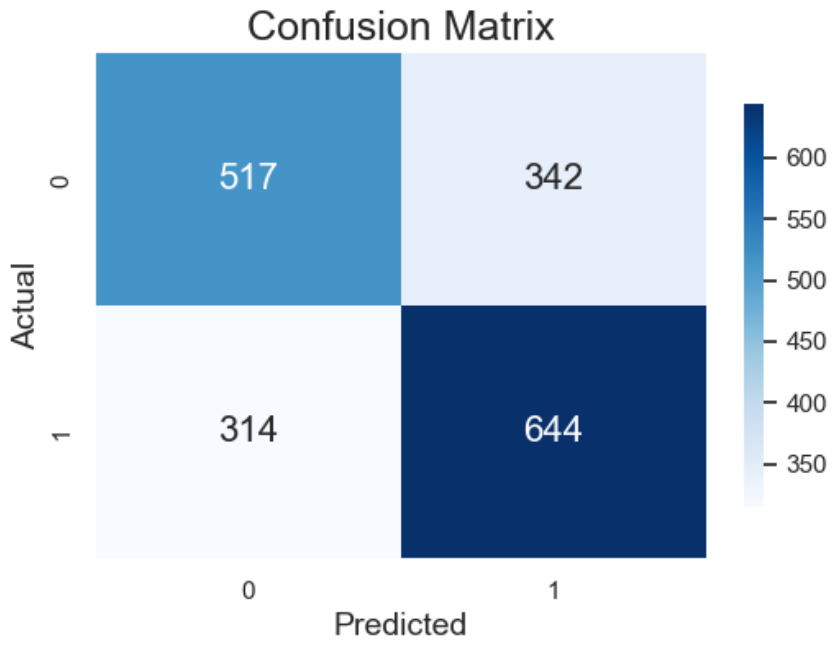

FIGURE 4.12. Confusion Matrix SVM

Table 4.9 indicates that the model's performance improves on the validation data compared to the training data across all metrics. This means that the model generalizes well and has not over-fitted the training data. Therefore the table demonstrate:

- **Accuracy** - The model's accuracy is higher (0,662) on the validation set comparing to the training set (0,619), what may suggest that the SVM model is underfitting.
- **Recall** - Recall is higher on the validation set (0,635) than on the training set (0,616), indicating that the model has generalized reasonably well, though the lower recall on training data suggests underfitting.
- **Precision** - Precision is higher on the validation set (0,637), indicating more reliability in avoiding false positives slightly better on validation data.
- **F1-score** - The F1-score on the validation set (0,635) is higher when compared to the training set (0,616), indicating that the model has a better balance between recall and precision on unseen data.

The Confusion Matrix in Figure 4.12 shows that the model has 517 True negatives and 644 True positives, meaning it correctly classified many instances for both classes. However, there are 342 False Positives (incorrectly predicting "Hit") and 314 false negatives (incorrectly predicting "Flop"), which could indicate some errors in both types of predictions.

4.2.5. K-Nearest Neighbors (KNN)

The KNN algorithm is a simple supervised machine learning method used for classification. It predicts the class of unlabeled data by comparing it to the labelled training data. KNN

works by selecting the nearest neighbours and the data points closest to the new data, and using majority voting to determine the final classification.

KNN is popular for classification tasks because it is easy to understand, adaptable, and can handle different types of datasets, including those with varying sizes, label numbers, noise levels, and contexts and that is why this algorithm was chosen [44].

For this model, a function was created to use cross-validation to determine the optimal number of neighbours for the given scenario. This function evaluates different values for the number of neighbours and identifies the one that yields the best performance, ensuring a well-suited configuration for the model.

TABLE 4.10. Classification Report of KNN

	Precision	Recall	F1-score	Support
0	0,622	0,602	0,612	859
1	0,653	0,672	0,663	958
accuracy	-	-	0,639	1817
macro avg	0,638	0,637	0,637	1817
weighted avg	0,638	0,639	0,639	1817

The Classification Report in Table 4.10 indicates that the model performs fairly similarly for both classes, with precision, recall, and F1-scores in the low to mid 63% range. On the other hand, it slightly performs better for Hits (class 1) than for Flops (class 0), as reflected in the higher metrics for class 1.

The model demonstrates moderate performance in predicting both flops and hits, with similar metrics across the two classes.

TABLE 4.11. Metric Measures of KNN

Accuracy - Training	0,648
Accuracy - Validation	**0,662**
Recall - Training	0,646
Recall - Validation	**0,637**
Precision - Training	0,646
Precision - Validation	**0,638**
F1 score - Training	0,646
F1 score - Validation	**0,637**

The Metric Measures in the Training Set and Validation Set in Table 4.11 demonstrate that the accuracy has a slightly improvement on the validation data compared to the

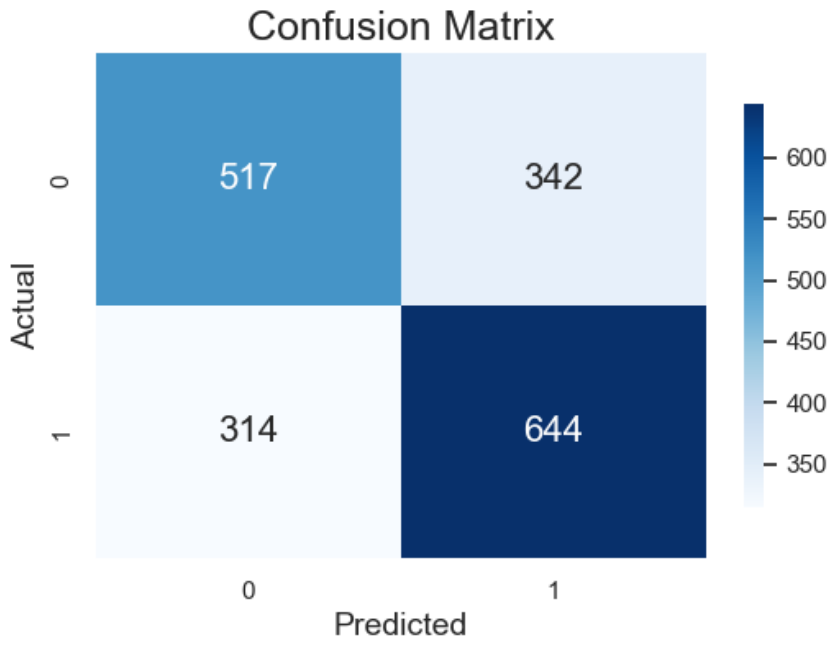

FIGURE 4.13. Confusion Matrix KNN

training data. However, the remain metrics are slightly lower on the validation set. This means the model is consistently across both sets. Therefore the table indicates:

- **Accuracy** - The model's accuracy is 0,662 on the validation set and 0,648 on the training set. The small difference suggests that the KNN model generalizes reasonably well.
- **Recall** - Recall is consistent across both training (0,646) and validation (0,637) on the validation set (0,66), meaning the model has a balanced capacity to capture positive instances
- **Precision** - Precision is also similar in training and validation sets, indicating effectiveness in avoiding false positives on validation data as it is on training data.
- **F1-score** - The F1-score is also identical on the training set (0,646) and in the validation set (0,637). This consistency reinforces that the model is well-balanced in terms of both capturing true positives and minimizing false positives, with no significant deviation between datasets.

The Confusion Matrix in Figure 4.13 indicate that the model accurately identified 517 True negatives and 644 True positives, in that way, it correctly classified many instances for both classes. On the contrary, there are also 342 False positives (incorrectly predicting "Hit") and 314 False negatives (incorrectly predicting "Flop"), which could indicate some errors in both types of predictions.

4.3. Final Results

The Results section of this study presents a detailed analysis of the outcomes from the experiments conducted using various ML models. The main focus is evaluating the performance of different algorithms based on the features selected and the effectiveness of each model in predicting the target outcomes.

In Section 4.1, the importance of features is clarified, providing insights into the most influential variables for the predictions. Techniques such as Mutual Information, Random Forest Permutation-Based Feature Importance, and XGBoost Classifier highlight different aspects of the data, offering a comprehensive view of the feature's importance.

The results indicate that, according to the Mutual Information technique, the least important features are the Occidental language and Galician. However, these variables were not excluded from the dataset to avoid the risk of the model overfitting.

Finally, the XGBoost shows that the best results, apart from those when the model had fewer than four variables, occur when the model includes more than 12 features.

Therefore, despite the analysis of feature importance, the decision was made to keep all variables in the dataset.

TABLE 4.12. Results across all the models

Models	Accuracy	Recall	Precision	F1-score
Neural Networks	0,656	0,655	0,655	0,655
Random Forest	0,653	0,649	0,653	0,648
XGBoost	**0,662**	**0,658**	**0,661**	**0,657**
Suppot Vector Machine	**0.637**	**0,635**	**0,636**	**0,635**
K-Nearest Neighbors	0,639	0,637	0,638	0,637

Subsequently, a study was conducted to analyze the performance of the machine learning models, which were thoroughly evaluated in the experiments. The study tests a range of algorithms, including Neural Networks, Random Forests, XGBoost, Support Vector Machines (SVM), and KNN. For each model, key performance metrics, such as accuracy, precision, recall, and F1-score, were used.

Table 4.12 shows that in terms of Accuracy, the model with the better performance is XGBoost with 0,662. In the same table, the metric Recall also exhibits the best outcome in the XGBoost models with a rate of 0,658.

The metric Precision reflects the reliability of a model's positive predictions, i.e., when the model predicts a movie will succeed and how often is that precision correct. Notably, XGBoost outperforms all other models in terms of precision with 0,661. This also transmits that compared to others the XGBoost has the lowest rate of false positives.

The last metric is F1-score which, as explained above combines precision and recall, offering a balanced view of the model's performance. Also in the F1-score the XGBoost model achieves the highest value 0,657, suggesting that the model is accurately identifying

successful movies and minimizing false positives. This makes XGBoost the ideal choice for predicting the success of a movie.

In that way, based on the analysis carried out on the results demonstrated by the various models and due to the high outcome in most metrics, the XGBoost model was chosen as the best model for predicting the success of a Latin-language movie.

CHAPTER 5

Conclusions and Future Work

The film industry plays an important role in the life of everybody. The movies we watch make us laugh, feel emotions and sometimes give us a reason to not feel lonely. Because of this, it is important to help the filmmakers and the industry among them to feel and be certain that every movie made and tested will be a success.

In this study, the focus is on a specific segment of the film industry: Latin-language movies, a group of smaller, but equally important, productions. The research builds upon previous studies in the same field, using their findings as a foundation for the work presented here.

The goal of this study is to find the key factors to develop a ML model that can accurately predict whether a Latin-language movie will be successful. This builds on prior studies within the field of movie success prediction while tailoring the analysis to this specific segment of the film industry. The process began with gathering all the necessary data, which was then carefully analyzed. Based on this, a machine learning model was developed to accurately predict whether a movie will be successful or not. Multiple machine learning models were evaluated, including Neural Networks, SVM, KNN, Random Forest, and XGBoost Classifier. The models were compared using performance metrics such as precision, recall, and F1-score. Among the models evaluated, XGBoost consistently outperforms the others, especially in terms of precision and F1-score, making it the most reliable option when accurate predictions of successful movies are critical.

This analysis provides a foundation for selecting the most appropriate machine learning model based on the specific needs of the binary classification problem in the context of movie success prediction.

In this way, and addressing the questions made in Chapter 1, it is possible to conclude the following.

Q1: *Which techniques can be used to predict the movie's success?*
In this study, several ML techniques were explored and chosen according to the binary classification problem. In that way, the models were selected (Random Forest, SVM, KNN, Neural Network and XGBoost Classifier) for their effectiveness and for demonstrating a good performance for this task

Q2: *Which available features can be used for predicting movie success?*
The features chosen for predicting movie success were based on a review of available online data and previous studies. These focus not only on the specific characteristics of the films but also on the actors, producers and directors who make them up, as well as economic variables such as revenue.

Q3: *Can machine learning models built with the right features provide meaningful insights to stakeholders in the film industry?*

The results of this study suggest that machine learning models, built using carefully selected features, can indeed offer valuable insights for stakeholders in the film industry, helping inform decisions around production, marketing, and strategic planning.

In conclusion, this study represents a significant advancement in predicting the success of Latin-language films by leveraging machine learning techniques. By addressing the challenges inherent in the film industry, such as diverse audience preferences and financial risks, the research highlights the importance of using robust models like XGBoost for accurate predictions.

One possible direction for future research would be to explore the integration of more diverse datasets, such as social media, including trends, audience engagement metrics, and other external factors that influence movie success. By adding features from different sources, it would be possible to obtain a model that more closely resembles the real world of Latin-language movies and viewer comments and ratings. Another avenue for future research could be to design models that reflect regional and cultural variations in movie preferences. This could be especially useful for predicting Latin-language film performance in various markets, highlighting on how regional tastes and viewing patterns influence box office results and audience ratings.

Bibliography

[1] I. Sindhu and F. Shamsi, "Prediction of imdb movie score and movie success by using the facebook," in *2023 International Multi-Disciplinary Conference in Emerging Research Trends, IMCERT 2023*, Institute of Electrical and Electronics Engineers Inc., 2023. DOI: 10.1109/IMCERT57083.2023.10075189.

[2] D. Menaga and A. Lakshminarayanan, "A method for predicting movie box-office using machine learning," Institute of Electrical and Electronics Engineers Inc., 2023, pp. 1228–1232. DOI: 10.1109/ICESC57686.2023.10192928.

[3] C. Schröer, F. Kruse, and J. M. Gómez, "A systematic literature review on applying crisp-dm process model," vol. 181, Elsevier B.V., 2021, pp. 526–534. DOI: 10.1016/j.procs.2021.01.199.

[4] R. Nihalaani, A. Shete, and D. Khan, "Movie success prediction using naïve bayes, logistic regression and support vector machine," in *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions), ICRITO 2021*, Institute of Electrical and Electronics Engineers Inc., 2021, ISBN: 9781665417037. DOI: 10.1109/ICRITO51393.2021.9596138.

[5] J. S. Saltz, "Crisp-dm for data science: Strengths, weaknesses and potential next steps," in *2021 IEEE International Conference on Big Data (Big Data)*, 2021, pp. 2337–2344. DOI: 10.1109/BigData52589.2021.9671634.

[6] D. Demir, O. Kapralova, and H. Lai, *Predicting imdb movie ratings using google trends*, 2012. [Online]. Available: http://recsys.acm.org/2011.

[7] J. S. Simonoff and I. R. Sparrow, *Predicting movie grosses: Winners and losers, blockbusters and sleepers*. [Online]. Available: www.imdb.com.

[8] M. T. Lash and K. Zhao, "Early predictions of movie success: The who, what, and when of profitability," Jun. 2015. DOI: 10.1080/07421222.2016.1243969. [Online]. Available: http://arxiv.org/abs/1506.05382%20http://dx.doi.org/10.1080/07421222.2016.1243969.

[9] K. Lee, J. Park, I. Kim, and Y. Choi, "Predicting movie success with machine learning techniques: Ways to improve accuracy," *Information Systems Frontiers*, vol. 20, pp. 577–588, 3 Jun. 2018, ISSN: 15729419. DOI: 10.1007/s10796-016-9689-z.

[10] 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," *Multimedia Tools and Applications*, vol. 82, pp. 4031–4061, 3 Jan. 2023, ISSN: 15737721. DOI: 10.1007/s11042-022-13448-0.

[11] P. Rajput, P. Sapkal, and S. Sinha, "Box office revenue prediction using dual sentiment analysis," *International Journal of Machine Learning and Computing*, vol. 7, pp. 72–75, 4 Aug. 2017, ISSN: 20103700. DOI: 10.18178/ijmlc.2017.7.4.623.

[12] B. A. Kitchenham, "Procedures for performing systematic reviews," 2004. [Online]. Available: https://api.semanticscholar.org/CorpusID:54019416.

[13] S. S. I. of Engineering, Technology, I. of Electrical, E. E. M. Section, A.-I. C. for Technical Education, I. of Electrical, and E. Engineers, *Movie success prediction using ensemble classifier*, ISBN: 9781728145143.

[14] V. Gupta, N. Jain, H. Garg, S. Jhunthra, S. Mohan, A. H. Omar, and A. Ahmadian, "Predicting attributes based movie success through ensemble machine learning," *Mfultimedia Tools and Applications*, vol. 82, pp. 9597–9626, 7 Mar. 2023, ISSN: 15737721. DOI: 10.1007/s11042-021-11553-0.

[15] J. Tripathi, S. Tiwari, A. Saini, and S. Kumari, "Prediction of movie success based on machine learning and twitter sentiment analysis using internet movie database data," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 29, pp. 1750–1757, 3 Mar. 2023, ISSN: 25024760. DOI: 10.11591/ijeecs.v29.i3.pp1750-1757.

[16] S. P. C. Federation, I. of Electrical, and E. Engineers, *Applying Machine Learning to Precit Film Daily Audience Data: System and Datase*, ISBN: 9781728197418.

[17] I. of Electrical, E. Engineers, A. for Computing Machinery, O. I. W. on Business Applications of Social Network Analysis (4th : 2013 : Niagara Falls, O. I. W. on Mining, A. S. N. for Decision Support (5th : 2013 : Niagara Falls, O. I. W. on Social Network Analysis in Applications (3rd : 2013 : Niagara Falls, O. I. W. on Semantic, D. A. of Information Networks (2013 : Niagara Falls, O. I. I. W. on Multiagent Systems, S. N. (: 2. : N. Falls, and O. W. on Web Behavior Analytics (2013 : Niagara Falls, *Prediction of Movies Box Office Performance Using Social Media*, p. 1498, ISBN: 9781450322409.

[18] Y. M. Cheang and T. C. Cheah, "Predicting movie box-office success and the main determinants of movie box office sales in malaysia using machine learning approach," Association for Computing Machinery, Feb. 2021, pp. 57–62, ISBN: 9781450388825. DOI: 10.1145/3457784.3457793.

[19] J.-Y. Nie, I. of Electrical, E. Engineers, and I. C. Society, *2017 IEEE International Conference on Big Data : proceedings : Dec 11- 14, 2017, Boston, MA, USA*, ISBN: 9781538627150.

[20] N. Quader, M. O. Gani, D. Chaki, and M. H. Ali, "A machine learning approach to predict movie box-office success," in *2017 20th International Conference of Computer and Information Technology (ICCIT)*, 2017, pp. 1–7. DOI: 10.1109/ICCITECHN.2017.8281839.

[21] S. Wu, Y. Zheng, Z. Lai, F. Wu, and C. Zhan, *Movie box office prediction based on ensemble learning.*

[22] W. M. Ruwantha and B. T. Kumara, "Lstm based approach for classifying twitter posts for movie success prediction," Institute of Electrical and Electronics Engineers Inc., Nov. 2020, pp. 1160–1165, ISBN: 9781728196770. DOI: 10.1109/DASA51403.2020.9317163.

[23] J. Ahmad, P. Duraisamy, A. Yousef, and B. Buckles, *Movie success prediction using data mining*.

[24] W. Lu, *Research on prediction of box office based on internet comments*.

[25] R. Acerado, "Cmata: Cyber trafficking monitoring and tracking prototype," *International Journal of Future Computer and Communication*, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:257382155.

[26] L. E. Phillips, P. Dhillon, A. Kotas, R. Kusler, J. Shih, and J. Kause, "Film production during the covid-19 pandemic," *Occupational Medicine*, vol. 74, pp. 24–28, 1 Jan. 2024, ISSN: 14718405. DOI: 10.1093/occmed/kqac102.

[27] Y. Zheng and C. K. Kwoh, "A feature subset selection method based on high-dimensional mutual information," *Entropy*, vol. 13, pp. 860–901, 2011. [Online]. Available: https://api.semanticscholar.org/CorpusID:12877142.

[28] A. Orlenko and J. H. Moore, "A comparison of methods for interpreting random forest models of genetic association in the presence of non-additive interactions," *BioData Mining*, vol. 14, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:231727935.

[29] A. Altmann, L. Tolosi, O. Sander, and T. Lengauer, "Permutation importance: A corrected feature importance measure," *Bioinformatics*, vol. 26 10, pp. 1340–7, 2010. [Online]. Available: https://api.semanticscholar.org/CorpusID:8680614.

[30] Y. Wang and X. S. Ni, "A xgboost risk model via feature selection and bayesian hyper-parameter optimization," *ArXiv*, vol. abs/1901.08433, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:59222733.

[31] M. Gong, "A novel performance measure for machine learning classification," *Decision-Making in Operations Research eJournal*, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:233756840.

[32] M. Grandini, E. Bagli, and G. Visani, "Metrics for multi-class classification: An overview," *ArXiv*, vol. abs/2008.05756, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:221112671.

[33] B. Abdualgalil and S. Abraham, "Applications of machine learning algorithms and performance comparison: A review," *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, pp. 1–6, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:216587771.

[34] S. Joshi, V. Kumar, V. Venkataramanan, and K. C S, "A review on neural networks and its applications," *Journal of Computer Technology & Applications*, vol. 14, p. 2023, Sep. 2023. DOI: 10.37591/jocta.v14i2.1062.

[35] F. Xoliyarov, S. Gulomov, and S. Bozorov, "The impact of artificial neural network architecture on network attack detection," *Proceedings of the 7th International Conference on Future Networks and Distributed Systems*, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:269762504.

[36] N. M. Abdulkareem and A. M. Abdulazeez, "Machine learning classification based on radom forest algorithm: A review," 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:265733945.

[37] I. Reis, D. Baron, and S. Shahaf, "Probabilistic random forest: A machine learning algorithm for noisy data sets," *The Astronomical Journal*, vol. 157, p. 16, 1 Jan. 2019, ISSN: 0004-6256. DOI: 10.3847/1538-3881/aaf101.

[38] R. Dhir and A. A. Raj, "Movie success prediction using machine learning algorithms and their comparison," *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*, pp. 385–390, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:145051725.

[39] Z. Balfagih, "Decoding cinematic fortunes: A machine learning approach to predicting film success," *2024 21st Learning and Technology Conference (L&T)*, pp. 144–148, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:268611551.

[40] C. Bentéjac, A. Csörgo, and G. Martínez-Muñoz, "A comparative analysis of gradient boosting algorithms," *Artificial Intelligence Review*, vol. 54, pp. 1937–1967, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:221283893.

[41] P. C. Murschetz, C. Bruneel, J.-L. Guy, D. Haughton, N. Lemercier, M.-D. McLaughlin, K. Mentzer, Q. Vialle, C. Zhang, and B. Bakhtawar, "Movie industry economics: How data analytics can help predict movies' financial success," 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:229532848.

[42] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," *Neurocomputing*, vol. 408, pp. 189–215, 2020, ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2019.10.118. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231220307153.

[43] A. B. S. Drid, D. Abdelhamid, and A. Taleb-Ahmed, "Support vector machine based clustering: A review," in *2022 International Symposium on iNnovative Informatics of Biskra (ISNIB)*, 2022, pp. 1–6. DOI: 10.1109/ISNIB57382.2022.10076027.

[44] S. Uddin, I. Haque, H. Lu, M. A. Moni, and E. Gide, "Comparative performance analysis of k-nearest neighbour (knn) algorithm and its different variants for disease prediction," *Scientific Reports*, vol. 12, Apr. 2022. DOI: 10.1038/s41598-022-10358-x.