INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA DE INTELIGÊNCIA ARTIFICIAL




NLP Based Recommendation Systems for Movies Platforms

JOÃO DOS SANTOS TAVARES DA SILVA
Setembro de 2024

P.PORTO

NLP Based Recommendation Systems for Movies Platforms

João Santos Tavares da Silva

Aluno nº: 1220192

Dissertação para obtenção do Grau de
Mestre em Engenharia de Inteligência Artificial

Orientador: Doutor Luiz Felipe Rocha Faria, Professor Coordenador do Instituto Superior de Engenharia do Porto do Instituto Politécnico do Porto

Júri:

Presidente:

Joaquim Filipe Peixoto dos Santos, Professor Adjunto do Instituto Superior de Engenharia do Porto do Instituto Politécnico do Porto

Vogais:

Luiz Felipe Rocha de Faria, Professor Coordenador do Instituto Superior de Engenharia do Porto do Instituto Politécnico do Porto

Luís Manuel Silva Conceição, Professor Adjunto do Instituto Superior de Engenharia do Porto do Instituto Politécnico do Porto

Porto, October 2024

Resumo

Nos dias de hoje, o uso de plataformas web aumenta cada vez mais. Deste modo, torna-se crucial melhorar a eficácia dos sistemas de recomendação na filtragem e personalização de conteúdo.

Esta tese explora a integração de técnicas de Processamento de Língua Natural (NLP) em sistemas de recomendação com o objetivo de aprimorar a precisão e relevância das sugestões que são fornecidas aos utilizadores. Tendo como foco a recomendação de filmes, esta pesquisa investiga como técnicas de NLP podem ser utilizadas para analisar conteúdo gerado pelos utilizadores tal como avaliações e comentários, para que exista uma maior compatibilidade entre as recomendações geradas e as preferências do utilizador.

O objetivo desta tese é propor um modelo que combina análise de sentimentos e modelação de tópicos para que possa compreender e prever as preferências dos utilizadores de uma forma mais eficaz. O modelo utilizará técnicas de NLP e aprendizagem automática para processar e analisar grandes conjuntos de dados de interações de utilizadores e meta dados de filmes.

Irá ser apresentada uma análise comparativa da utilização de modelos de NLP em sistemas de recomendação em contraste com sistemas de recomendação tradicionais, salientando as melhorias na precisão e satisfação dos utilizadores.

Os resultados obtidos somente com a utilização de técnicas NLP no modelo proposto para recomendação de filmes atingem o valor de 0.409 para RMSE e 0.339 para MAE. Este resultado é a combinação de um modelo de regressão logística com uma acurácia de 90% e do modelo BERTopic com uma coerência de tópicos de 0,53 que, para um sistema de recomendação que é apenas baseado em tarefas de análise de sentimento e modelação de tópicos a reviews feitas por utilizadores, são resultados satisfatórios e que permitem efetuar recomendações adequadas.

Palavras-chave: Análise de Sentimentos, Processamento de Linguagem Natural, Modelação de Tópicos, Processamento Semântico, Sistemas de Recomendação, Aprendizagem Automática

Abstract

Nowadays, the usage of web platforms is increasingly growing. In this light, it becomes crucial, to improve the efficiency of recommendation systems in filtering and personalizing content.

This thesis explores the integration of Natural Language Processing (NLP) techniques into recommendation systems with the aim of enhancing the accuracy and relevance of the suggestions provided to users. Focusing on movie recommendations, this research investigates how NLP techniques can be used to analyse user-generated content such as reviews and comments, in order to provide better compatibility between the generated recommendations and the user preferences.

The aim of this thesis is to propose a model that combines sentiment analysis and topic modeling to more effectively understand and predict user preferences. The model will utilise NLP and machine learning techniques to process and analyse large datasets of user interactions and movie metadata.

A comparative analysis will be presented on the use of NLP models in recommendation systems as opposed to traditional recommendation systems, highlighting improvements in accuracy and user satisfaction.

The results obtained using only NLP techniques in the proposed movie recommendation model achieved an RMSE value of 0.409 and a MAE of 0.339. This result is a combination of a Logisitic Regression model with 90% accuracy and a BERTopic model with a topic coherence of 0.53. For a recommendation system based only on sentiment analysis and topic modeling of users reviews, these are satisfactory results that allow for making appropriate recommendations.

Keywords: Sentiment Analysis, Natural Language Processing, Topic Modeling, Semantic Processing, Recommendation Systems, Machine Learning

Acknowledgements

I would like to express my sincere gratitude to my family because I wouldn’t be here if it wasn’t for they’re efforts. I always had they’re unconditional support through this journey and they always provided me with strenght and motivation during the most challenging moments.

I would also like to thank my girlfriend, whose support, care and encouragement were fundamental on helping me stay focused and determined to complete this work.

To my friends, who were always there to offer support and moments of relaxation, helping me find the necessary balance.

Finally, I would like to thank my supervisor whose knowledge and guidance were essential for the development of this work.

Index

Figures

Tables

Acronyms and Symbols

Lista de Acrónimos

ML	Machine Learning
NLP	Natural Language Processing
NLU	Natural Language Understanding
NLG	Natural Language Generation
CNN	Convolutional Neural Network
NN	Neural Network
RNN	Recurrent Neural Network
LSTM	Long-Short Term Memory
RecNN	Recursive Neural Network
MN	Memory Network
BERT	Bidirectional Encoder Representations Transformers
SVM	Support Vector Machine
DT	Decision Tree
LR	Logistic Regression
NB	Naïve Bayes
LDA	Latent Dirichlet Allocation
LSA	Latent Semantic Analysis
PLSA	Probabilistic Latent Semantic Analysis
DMM	Dirichlet Multinomial Mixture
GSDMM	Gibbs Sampling Dirichlet Multinomial Mixture
LF-DMM	Latent Feature Dirichlet Multinomial Mixture
GPU-DMM	Generalised Pólya Urn Dirichlet Multinomial Mixture
GPU-PDMM	Generalised Pólya Urn Poisson Dirichlet Multinomial Mixture

BTM	Biterm Topic Model
WNTM	Word Network Topic Model
SATM	Self-Aggregation Based Topic Modeling
PTM	Pseudo-Document-Based Topic Modeling
TF-IDF	Term Frequency-Inverse Document Frequency
SVD	Singular Value Decomposition
NMF	Non-negative Matrix Factorization
RSME	Root Square Mean Error
MAE	Mean Absolute Error

1 Introduction

On this document, it will be presented the work done on exploring the integration of NLP techniques on recommendation systems.

With this, it will be explored traditional techniques of recommendation systems and how the NLP can help those traditional techniques improving the delivery of more personalised content to the users considering the big amounts of data that are available online.

On this chapter, it will be presented the context of this thesis and the problem that is trying to be solved. Along with that, it will be mentioned the expected results and the structure of this document.

1.1 Context

In the digital age, the proliferation of online content has led to an unprecedent need for effective recommendation systems (Pavitha et al., 2022). These systems are crucial in helping users navigate throughout all the existing content such as movies, books, articles, products and so on. The traditional recommendation systems primarily relied on collaborative filtering, content-based filtering methods and hybrid filtering methods. Although these methods are effective, they often fall short in comprehensively understanding user preferences, especially in the nuanced and subjective domains like movie recommendations.

Natural Language Processing has opened new doors in recommendation systems. NLP is primarily focused on the interactions between computers and human language. It enables machines to read, understand and make sense of human languages in a valuable and meaningful way. By joining these capabilities on recommendation systems, those will be allowed to perform deeper analysis of user-generated content, such as reviews and comments.

The integration of NLP into recommendation systems marks a significant shift from traditional methods. By analysing textual data, NLP-based systems can interpret user sentiments, preferences and even cultural aspects which are overlooked by the conventional algorithms.

NLP in recommendation systems allows the processing of user's sentiments through movies reviews offering a more personalised experience for the user (Pavitha et al., 2022).

The application of NLP in recommendation systems has its own challenges such as ambiguities, idioms, and varied expressions. Also, the ethical implications of using user-generated data, such as privacy concerns and the potential for bias in AI algorithms, must be addressed.

Despite these challenges, the potential of NLP in revolutionizing recommendation systems is huge. As technology advances, so does NLP techniques, offering even more accuracy on understanding user preferences. This thesis aims to explore the integration of NLP into movie recommendation systems to improve the delivery of personalized content.

1.2 Problem Description

The core problem addressed in this thesis revolves around the limitations of traditional recommendation systems and the challenges in integrating Natural Processing Language (NLP) to enhance their performance, particularly in the context of movie recommendations.

The traditional recommendation systems, while certain scenarios, often rely heavily on user's ratings. These systems typically use collaborative filtering, which suggests items based on preferences of similar users, or content-based filtering, which recommends items similar to those a user has liked in the past. Using only this information, they struggle to capture the depth and complexity of user preferences, especially when it comes to subjective domains such as movies.

The traditional content-based approaches that use topic modeling have some difficulties on scalability, which means they have trouble when trying to scale when the corpus size increases since it causes ineffective over time. It also has an issue with comprehensiveness which causes ineffectiveness on recommendations. Detecting relationships between topics in a corpus can also be an issue. (Bagul & Barve, 2021)

Another identified issue on content-based filtering is the called "cold start". Content-based filtering algorithms don't need to know past activities from the user since they build user profile through the interactions that are made. Since the system has no context of new user's preferences, it doesn't know what to recommend. (Shah et al., 2023)

These recommendation systems are also very likely to have the "cold start" problem, where new items or users with limited interaction history are challenging to recommend.

By adding NLP on those systems, it will allow the analysis of user-generated content such as reviews and comments. NLP has the potential to extract deeper insights on user preferences, going beyond mere ratings or viewing history. This approach can include sentiment analysis, topic modeling and semantic analysis to better understand the user's interests.

Integrating NLP on recommendation systems is not a straightforward work and has its own challenges such as the complexity of the natural language itself. Remember that some nuances such as sarcasm, irony may be included on user's reviews and comments. The systems needs to have this in consideration in order to not return false positive results.

1.3 Goals

The goal of this study is to explore ways to improve the capabilities of the recommendation systems through NLP techniques. In the first place, the goal is to investigate theoretical aspects of NLP on recommendation systems, providing a comprehensive understanding of how sentiment analysis and topic modeling can be effectively integrated to improve performance.

The model for this study is designed to use textual data efficiently, ensuring that it is really able to capture user preferences and content features accurately. The effectiveness of the model will also be tested through multiple NLP models for sentiment analysis and topic modeling. The comparison of the results of the different used models will also be part of this study.

The evaluation of different sentiment analysis techniques is very important in understanding how the models are interpreting user reviews and feedbacks. The comparison between different models will help refining the recommendation algorithms making them more aligned with the user's preferences. On the same way, the investigation of topic modeling approaches is essential for understanding content, ensuring that the system can really align the user's interests with relevant categories.

On this study, another goal is to discuss future directions, providing insights into potential advancements in NLP integration with recommendation systems and discussing the model's adaptability to multiple domains and larger datasets.

By achieving the mentioned goals, the study aims to contribute to the field of recommendation systems demonstrating the impact of NLP on improving the user experience and the system efficiency.

1.4 Document Structure

This document will be divided in six chapters. This section aims to describe in what consists each one of the chapters of this document.

The first chapter *Introduction* has the goal of describe the context of this document, the description of the problem that it aims to solve and the goals that is trying to achieve.

During the chapter *State of the Art*, this document presents a literature review that goes through multiple topics such as related work on recommendation systems that uses NLP, Natural Language Processing, Recommendation Systems, Sentiment Analysis, Topic Modeling and Matrix Factorization.

On third chapter called *Dataset and Methodology*, is going to be described the models that were selected for this study along with the selected metrics to evaluate the model's performance. It will also include a description of the used datasets, how they were pre-processed, privacy and ethical considerations.

On the *Recommendation System Model* chapter, it will be described a high-level architecture of the model and how the models were trained and tested.

During the fifth chapter, *Discussion of Results*, it will be presented the results of the models during the testing phase.

Finally, there is a *Conclusion* chapter where will be presented the conclusions of this document and future work.

2 State of the Art

On this chapter aims to present content related to the state of the art and content that is important for the development of this thesis.

The contents that are considered relevant to talk about on this chapter are Recommendation Systems, Sentiment Analysis, Topic Modeling and correlation between sentiment and topic with matrix factorization.

2.1 Related Work

While researching topics for this thesis, several articles were identified that provided relevant information on various aspects of recommendation systems, sentiment analysis and topic modeling. This section synthesizes the insights and findings from these articles to present a comprehensive overview of the current state of research in this field.

One of the articles focused on a movie recommendation system that employs cosine similarity to suggest movies based on user preferences. The system enhances user experience by incorporating sentiment analysis on movie reviews, using algorithms like Naïve Bayes and Support Vector Machine (SVM). The study found SVM to be more accurate, with a notable accuracy of 98.63%. This approach underscores the effectiveness of sentiment analysis in refining recommendation systems (Pavitha et al., 2022), as demonstrated by its ability to discern nuanced feelings about movies. The study also compared other models like Random Forest, LSTM, and CNN, with SVM showing superior performance. The use of cosine similarity in this context yielded promising results, effectively recommending movies with similar themes.

Another significant contribution came from a project named Sentirec, which diverges from movie recommendations to focus on content recommendation in social media. This system analyzes Twitter data to understand user sentiments and suggest content accordingly. The process involves collecting tweets, pre-processing them for sentiment analysis, and then

classifying them using various algorithms like SVM, Decision Tree, and Logistic Regression. The final model combines the strengths of multiple algorithms, each weighted according to its accuracy, to enhance the system's predictive capability (Oswalt Manoj et al., 2023).

A different approach was explored in an article about a plot-topic based movie recommendation system using WordNet. This system moves beyond genre-based recommendations to focus on movie storylines. It employs content-based filtering, extracting movie topics through topic modeling of plot summaries from IMDb. The system also utilizes NLP clustering approaches and semantic similarity measures to assign topics to movies. This method acknowledges the complexity of movie plots, which often encompass multiple topics, and seeks to assign topics more accurately to enhance recommendation relevance (Institute of Electrical and Electronics Engineers & Institute of Electrical and Electronics Engineers. Delhi Section, n.d.).

Another paper goes through the challenges of data sparsity in collaborative filtering. It proposed a recommendation system based on deep sentiment analysis and matrix factorization. This system uses Latent Dirichlet Allocation (LDA) for topic modeling, BERT for sentiment analysis, and matrix factorization to create user-feature and item-feature matrices. The integration of these matrices with sentiment analysis data aims to improve the accuracy of recommendations and forecast ratings more effectively (Rahman & Hossen, 2019).

Lastly, an article titled "Recommendation Systems: Principles, methods and evaluation" provided a broad overview of recommendation systems. It discussed the increasing need for efficient information filtering due to the growing volume of online content. The paper explored various prediction techniques, including content-based, collaborative, and hybrid filtering, and their potential in personalizing content recommendations. It also addressed the evaluation of recommendation algorithms, focusing on accuracy and coverage as key metrics (Isinkaye et al., 2015).

In summary, these articles collectively offer a rich tapestry of research in the field of recommendation systems and sentiment analysis. They highlight the diversity of approaches and methodologies employed to enhance the accuracy and user experience of recommendation systems, from movie suggestions to social media content curation.

2.2 Natural Language Processing

Now a days, machines are able to listen natural language and, if needed respond with natural language. This is something that can be made on intelligent machines when it is applied natural language processing (NLP) (SCAD College of Engineering and Technology & Institute of Electrical and Electronics Engineers, n.d.).

Natural Language Processing is a research field that aims to develop ways for the computers to understand native language in a intelligent way. In other words, the main goal is to make a computer understand and analyse human language by translating it to computer language. The biggest usages of Natural Language Processing are speech analysis, machine translation, automatic summarization, speech recognition and so on (Shivahare et al., 2022).

2.2.1 Natural Language Components

Natural Language Processing has two main components. Those components are defined as:

- Natural Language Understanding (NLU)
- Natural Language Generation (NLG)

Natural Language Understanding is the component that helps the machine extracting information from the text (Shivahare et al., 2022).

The Natural Language Generation works in the opposite way, its goal is to translate computer information to native text and it can involve text editing, phrase editing, and content recognition (Shivahare et al., 2022).

NLG can go even further since it can translate the text into audible speech through a text-to-speech conversion. To do this, NLG starts by identifying the content of the information that needs to be translated into text. After this, the words and sentences are going to be structured using lexicon and grammar rules. As the final step, it will use a speech database along with its model to form a coherent string sentence by assembling all the recorded phonemes (SCAD College of Engineering and Technology & Institute of Electrical and Electronics Engineers, n.d.).

2.2.2 Natural Language Processing General Applications

NLP is a widely used tool on voice assistant where this technique is able to understand user's voice commands and perform actions considering the extracted information.

Another use of NLP techniques would be on chatbots. Chatbots can assist companies dealing with tedious tasks and solving problems. Chatbots need NLP so they become able to understand and generate answers for the user.

A very useful usage of Natural Language Processing is the automatic translation. Automatic translation is the process of translating a sentence that was written on a language to another language. The first translators that were created were mainly based on dictionary and legal principles and didn't achieve great results. With the improvements of the data, the effectiveness of the new machines, the machine translation has become even more accurate when translating from one language to another.

2.3 Recommendation Systems

There has been a huge growth of the amount of digital information and the number of visitors has created a potential challenge of information overload (Isinkaye et al., 2015).

The creation of search engines such as Google, DevilFinder and so on have partially solved this problem but, on those engines, there is no content personalization. Considering this, the need of recommendation systems has been increasing more than ever before.

Recommendation systems are systems that can handle the problem of information overload by filtering content according to user's preferences, interests or even observed behaviours. They also have the ability to predict whether a particular user would prefer an item or not based on the information found on that user's profile (Isinkaye et al., 2015).

These systems can benefit both service providers and users since they are able to reduce transaction costs of finding selecting items in an online shopping environment. They also have proved to improve decision making process and quality (Isinkaye et al., 2015).

2.3.1 Recommendation Process Phases

The recommendation process, can be divided in three different processes:

- Information Collection Phase
- Learning Phase
- Prediction/Recommendation Phase

The Information Collection Phase, is to collect all the essential information on user's preferences so it can generate a user profile or model for the prediction tasks including that user's attributes, behaviours or content of the resources that the user accesses.

Once the recommendation system has this information well gathered and organized, it'll be able to function accurately. It needs to know as much as possible from the user so that the suggested content can match, in a more accurate way, with the user's preferences.

These recommendation systems often rely in different types on input such as:

- Explicit feedback: Usually the system asks the user, through the user interface to provide ratings for items in order to improve the model. The accuracy of this system depends on the quantity of ratings provided by the user. The downside of this method is that it requires effort form the user's side and not always the user is ready to provide that information accurately. Besides these downsides, it seems that this is the method that provides more reliable information since it doesn't need processes like preferences extraction.
- Implicit feedback: By monitoring user's actions such as history of navigation history and time spent on some articles, books, movies descriptions, the system can automatically infer the user's preferences. This reduces the need of the users to explicitly tell the system his preferences but is less accurate. In the other hand, this method is more objective since there is no bias arising from users responding in a socially desirable way.
- Hybrid feedback: It gathers the strengths of implicit and explicit feedback in order to minimize their weaknesses and get a best performing system. This method can work when the implicit feedback is working most of the time but sometimes it checks explicit ratings to confirm its recommendations.

After collecting the needed information of the user's preferences, the learning phase is mainly to apply the learning algorithm so it can filter and exploit those preferences from the gathered feedback.

The final phase, prediction/recommendation, is to recommend or predict the kind of content the user may prefer. This can be made directly based on the collected dataset or throughout the user's activity that was observed by the system itself

2.3.2 Recommendation Filtering Techniques

To a recommendation system to provide useful and good recommendations, it needs accurate techniques. There are different recommendation techniques and it is important to understand how they work when developing a recommendation system.

The existing recommendation filtering techniques are:

- Content-based filtering
- Collaborative filtering
- Knowledge-based filtering
- Hybrid filtering

2.3.2.1 Content-based Filtering

This technique works by analysing the attributes of items in order to generate predictions. By publishing new contents, like news, products, or in this thesis specific case, movies, the

system will analyse the features that were extracted from the content of previous evaluations made by the user in the past. (Isinkaye et al., 2015)(Vedaswi et al., 2023)

This technique can use different models to find similarities between documents (Vedaswi et al., 2023). Those models can be Term Frequency Inverse Document Frequency (TF/IDF) or probabilistic models like Naïve Bayes Classifier, Decision Trees or NN. (Isinkaye et al., 2015)

Unlike collaborative filtering, this technique does not need to know about other user's profile, only the target user since it only compares the features of new content with the profile of that user to make recommendations.

2.3.2.2 Collaborative Filtering

This filtering technique does not rely on data from just one user. Instead, it gathers data from another users that may influence the given recommendations by the system. Collaborative filtering can be seen as a regression model where the output is the rating as a numerical value (Vedaswi et al., 2023).

With the existence of this neighbourhood, the user will receive recommendations of content it never rated before but the neighbourhood rated it positively. Considering this, the recommendations made by this technique can either be a prediction or a recommendation. The prediction is a numerical value that expresses the predicted score of an item to a user.

There are two different categories inside this technique:

- Memory based technique: This technique will use the items that the user already rated before to find matches with a neighbour that shares interests or preferences. This can be made in two ways through user-based and item-based techniques. On user-based technique, it will be calculated the similarity between users by comparing their ratings to the exact same item and with this information it will compute the predicted rating for an item by the active user as a weighted average of the ratings of the item by users similar to the active user where weights are similar. The item-based filtering will compute the prediction using similarities with items instead users. This technique will build a model of item similarities by retrieving all items rated by an active user-item matrix. This matrix will be used to calculate how similar the retrieved items are to the target item and selects the k nearest items. This prediction is made by taking a weighted average of the active users rating on the similar items.
- Model-based techniques: This technique makes use of previous ratings so the model can learn. Since this technique uses a pre-computed model and have proved to produce recommendation results similar to neighbourhood recommendation techniques, this can quickly recommend a set of items. It can be used techniques like Single Value Decomposition (SVD), Regressions, Clusterings, Latent Semantics and so on

(Isinkaye et al., 2015)

2.3.2.3 Knowledge-based Filtering

This technique works in a different way than the previous technique. To provide recommendations to the user, knowledge-based filtering technique uses particular queries instead of a user's rating history (Ramesh & Vijayalakshmi, 2022).

For this technique to start providing recommendations, it will ask the user to provide some criteria for the expected results and also an example of an item. Giving the provided information by the user, the system will query a database to extract content based on that (Ramesh & Vijayalakshmi, 2022) .

2.3.2.4 Hybrid Filtering

This hybrid filtering technique can combine different recommendation techniques so it can get an optimization to avoid limitations of the pure techniques.

This is a combination of the two previous approaches, utilizing content-based filtering along with collaborative approach or utilizing collaborative techniques along with a content-based approach (Ramesh & Vijayalakshmi, 2022) (Isinkaye et al., 2015).

2.4 Sentiment Analysis

Sentiment analysis is a very popular task now a days. Sentiment analysis is also known as opinion mining and is very useful technique related with natural language processing and its goal is to classify whether a set of data is positive, neutral or negative. It gives us sentiment comprehension of gathered data that can be found on news, reviews, websites and so on (Rahman & Hossen, 2019).

On the case of recommendation systems, this technique is very important because it allows the system to gather deeper information on user's preferences and interests.

For this task, there are some techniques that can be used. Between the most common ones, we can find techniques like Logistic Regression (LR), Support Vector Machine (SVM), Naïve Bayes (NB) and Neural Networks (NN).

2.4.1 Logistic Regression

Logistic Regression is a machine learning algorithm that is very useful for binary classification tasks, such as sentiment analysis where the outcome is a positive sentiment or a negative sentiment. LR calculates the log odds for each feature in the dataset and applies a sigmoid function to those log odds. This process will return the probability of a given instance to belong to a particular class, making LR a great tool for determining sentiment in various types of text data. It can be easily applied on domains such as movie reviews, twitter, products feedback and so on.

Despite its strengths, LR also has some limitations. LR is very sensitive to noise, it can affect the accuracy of the model. Also, although LR is very good to perform binary classifications, it's not so good when it comes to perform multi class classifications. In order to perform multi class classifications, it needs some more extensions to deal with more complex scenarios. Nevertheless, when compared to other classification methods, such as decision trees, SVMs, LR is way easier to apply and can scale very well on large datasets.

2.4.2 Support Vector Machines

SVM models are supervised machine learning algorithms that are most commonly used for two purposes which are classification (SVC) and regression (SVR). On this model, each data item is placed on a n-dimensional space and the classification happens by finding the hyper-plane that differentiates the classes.

There can be found several hyper-plans but we choose the one that maximizes the distance between classes (Rahman & Hossen, 2019) (Medhat et al., 2014).

Support Vector Machines can help a lot on sentiment classification thanks to the sparsity of the text data (Medhat et al., 2014) since few features are irrelevant but they tend to be, somehow, correlated with one another.

SVMs have the ability to build a nonlinear decision surface in feature space. That means, the existing data instances can be separated in a nonlinear way (Medhat et al., 2014).

One of common usage of SVM is to classify reviews. There are usages of this techniques to evaluate the information quality in product reviews.

2.4.3 Naive Bayes

Naive Bayes is a classification algorithm that is based on statistics and has the principle that every feature is independent from one another (Rahman & Hossen, 2019).

The mathematical expression that represents Naïve Bayes is:

$$P(x|Y) = \frac{P(Y|x)P(x)}{P(Y)}$$

To calculate $P(Y|x)$, the equation is:

$$P(Y|x) = P(y1|x)P(y2|x) \dots P(yn|x)$$

On the equations, x is the class variable and Y is the dependent feature vector. On this algorithm it is determined the posterior probability since $P(x|Y)$ is the probability of x given Y.

The $P(x)$ represents the prior probability of a label or the likelihood that a random feature set the label. $P(Y|x)$ represents the prior probability that a given feature is being classified as a label and finally $P(Y)$ is the prior probability that a given feature has occurred (Medhat et al., 2014).

There are three types of Naïve Bayes which are:

- Gaussian: Features follow a normal distribution
- Multinomial: Most used for discrete counts such as text classification
- Bernoulli: Most useful when dealing with binary features

Naïve Bayes algorithm is mostly used for tasks such as sentiment analysis, spam detection and much more.

2.4.4 Neural Networks

Neural networks have a structure that is highly inspired by the human brain and they have a great ability such as to recognize patterns.

A Neural Network is usually composed by neurons that is the basic unit of a neural network. Each neuron receives an input, processes it and sends it through its output (Medhat et al., 2014).

They may be composed with an input layer that receives the initial data, a hidden layer where the inputs are processed and forwarded to the next layer of neurons, and the output layer where the final result is returned.

To its training, each connection between neurons has weights and bias. These values are adjusted during the training.

During the training process, the input data is inserted on the input layers and will be processed in the hidden layers where, in the end, an output will be generated. Not all neurons are activated during this process because neural networks make use of activation functions that determines whether a neuron should be activated or not. It also makes use of algorithms like Gradient Descent for backpropagation. Backpropagation is a very important part of the training since it's here that the values of the weights and bias are updated based on the error. It also makes use of loss functions that measures the error in the output. The NN will keep training until it's able to minimize the error.

The neural network models have been improving during time and now there are deep learning models that have vast applications. These deep learning models have lots of potential on solving NLP problems and on handling the complexity of a text for sentiment polarity detection (Hoda et al., n.d.).

Some of these models can be:

- CNN
- RNN
- RecNN
- MN

CNN is a deep neural network that is designed to image identification and processing by handling pixel inputs. This neural network works by applying filters and layers to photos before analysing results and they can achieve high levels of accuracy on this task (Khare, 2022). CNN's have also become a popular model for image sentiment classification. On NLP matter, the input is usually a sentence where each row indicates a word and each word indicates a vector (Hoda et al., n.d.).

RNN's are neural networks that can easily deal with sequences since the output of the preceding layer is inserted into the succeeding layer as an input. This RNN's ability has made them popular on solving NLP related problems. RNN's are also able to hold memory that stores sequences and help to predict the next word (Hoda et al., n.d.).

There are some RNN variants that can also be very useful on NLP related tasks like LSTM's and GRU's (Hoda et al., n.d.). LSTM's was mainly created to avoid the long-term dependency problem that exists on the traditional RNN since LSTM has the ability to remember that long-term information (Universitas Sumatera Utara & Institute of Electrical and Electronics Engineers, n.d.). GRU's are quite similar to LSTM's since it has gates to manage the information flow but with a little advancement. GRU does not hold separate memory cells, instead it keeps only the hidden state to make the process simpler and faster (Hoda et al., n.d.).

The RecNN model is an adaptative model based on deep tree complex inherent chains. This model will represent the phrases in dimensional vectors and follows the method of binary trees for composition (Hoda et al., n.d.).

The MN contains a memory *m* with four components which are the *Input Feature Map, Generalisation, Output Feature Map* and *Response*. This model works basically like a CPU with memory and is widely used to process large amounts of data such as for knowledge base questions and answering tasks (Hoda et al., n.d.).

There are pre-trained models that can also be used for the task of sentiment analysis like BERT. BERT is a pre-trained language representation model that is able to collect information of the text through its self-attention mechanism. It has no forgetting gate so all the information is retained. BERT is very good expressing the semantic information of a sentence and on finding correlation features between words. (Liu & Zhao, 2023)

2.5 Topic Modeling

The topic modeling task is an existing way to analyse untagged data. Each topic is usually defined as a group of words that normally occur together in a big frequency. Topic modeling is able to connect words that have similar meanings. These algorithms can unveil the hidden thematic structure in a document.

There are several techniques such as Latent Semantic Analysis (LSA), Probabilistic Latent Semantic Analysis (PLSA), Latent Dirichlet Allocation (LDA) and so on.

The problem of the previous mentioned topic modeling techniques, is that they are very suitable for large corpus and when applying these models on reviews that generally are small documents, the accuracy may not be the expected. There are models that can help on this situation like Dirichlet Multinomial Mixture (DMM) and Biterm Topic Modeling (BTM) (Yamunathangam et al., 2021).

2.5.1 Latent Semantic Analysis

When we have the need to analyse how a document and the terms in that document are correlated, Latent Semantic Analysis comes into play. LSA is a technique that helps on the analysis of the relation between a group of documents and the terms that are present on those. It manages to do that through the generation of a set of concepts that are related to the documents and terms (IEEE Electron Devices Society et al., n.d.).

LSA will create a vector-based representation for the text so it can generate semantic content. To lower the dimension of this vector, LSA makes use of the Singular Value Decomposition Technique (SVD) to factorize matrices. The results that are returned by LSA are able to be used as input for further algorithm processing (IEEE Electron Devices Society et al., n.d.) (Hoblos, 2020).

2.5.2 Probabilistic Latent Semantic Analysis

This technique has many applications and most of them is related with retrieval of information through natural language processing (Hofmann, 2001).

PLSA is able to discover the latent meaning between documents and words to process the sparse word-document matrix that can be used later to, for example, find similar documents by applying distance metrics such as cosine similarity.

This technique uses a statistical latent class model to estimate the transformation matrix. This matrix is what will allow the discovery of underlying latent meaning from the sparse matrix. This is done through the assignment of probability distributions over classes to words and documents (Hofmann, 2001).

2.5.3 Latent Dirichlet Allocation

Latent Dirichlet Allocation is one of the most popular probabilistic topic model and is largely used and preferred to analyse very large documents (IEEE Electron Devices Society et al., n.d.) (Negara et al., n.d.).

This technique can also be used for summarization purposes and clustering purposes since LDA is capable of generating a list of topics for each analysed document. The distribution of topics per document is called Dirichlet distribution and the results of this distribution will allow to allocate words in the document to different topics (Negara et al., n.d.)

On LDA, each topic is represented as a set of words with a certain probability that defines how possible is a term to be related to a topic (IEEE Electron Devices Society et al., n.d.).

The approach that LDA uses is to use the input of individual documents and several parameters and will produce results in the form of a model that consists of weights that can be normalized according to probability. The probabilities are the probability of a document produce a specific topic and the probability of a topic produce certain words (Negara et al., n.d.) .

There are three variants of LDA which are the Correlated Topic Modeling that is a statistical model which shows correlation between topics. Another variant is the Dynamic Topic Modeling which is a generative model that can be used to analyse and study the evolution of the topics of a collection of documents over time and, finally, LDA has a variant called Hierarchical LDA. This last variant will locate a tree-like hierarchy of topics on a corpus. On that three, each level is more precise than the previous (IEEE Electron Devices Society et al., n.d.).

2.5.4 Short Text Topic Modeling

In a short text corpus *D* consisting of *N* documents and a vocabulary *W* of size *V*, each document *d* contains *nd* words. Topics that are present in *D* are defined as multinomial distributions over *W* and the topic representation of a document *d* is a multinomial distribution over *K* predefined latent topics. The goal of topic modeling is to identify *K* salient topics from *D* and determine the topic representation for each document. Classical probabilistic topic models such as LDA uses the Dirichlet prior for both topics and document topics representation. This approach helps smooth the topic mixture in documents and the word distribution in each topic, addressing overfitting issues common in PLSA, especially with increasing topic numbers and vocabulary size. The short text topic modeling techniques adopt the Dirichlet distribution as the prior distribution. The main goals of short text topic modeling are to learn the word representation of topics and learn the sparse topic representation of documents (Qiang et al., 2022).

For short text topic modeling, there are several methods that can be applied such as:

- Dirichlet Multinomial Mixture Based Methods
- Global Word Co-Occurrences Based Methods
- Self-Aggregation Based Methods

Dirichlet Multinomial Mixture model is based on the assumption that each document is related to only one topic. This assumption fits more for short texts than the assumption of each text is generated by multiple topics. From this model, more models were proposed taking this assumption into consideration like:

- GSDMM
- LF-DMM
- GPU-DMM
- GPU-PDMM

(Qiang et al., 2022)

The Global Word Co-Occurrence based methods takes into consideration that the closer two words, the more relevance the two words have. These methods earn the latent topics from the global word co-occurrences obtained from the original corpus. They work as a sliding window for extracting word co-occurrences. If the average of length of the documents is ten, the sliding window will also be defined with a size of ten, else they can directly take each document as a sliding window. Some Global Word Co-Occurrence methods can be:

- BTM
- WNTM

(Qiang et al., 2022)

The Self-Aggregation based methods will merge short texts into long pseudo documents before inferring the latent topics. Previously, the self-aggregation techniques started by merging the texts and apply topic models right next. The most recent techniques integrate the clustering and topic modeling in a single iteration.

The most recent Self-Aggregation techniques are:

- SATM
- PTM

(Qiang et al., 2022)

2.5.5 GSDMM

GSDMM stands for Gibbs Sampling Dirichlet Multinomial and it is a modified version of LDA that was specifically designed to deal with short texts such as tweets, movie reviews and it assumes that each document is related to one topic. GSDMM has the ability to automatically determine the number of clusters while achieving a good balance between the topic robustness and its uniformity.

It can effectively address the challenge of sparsity and high-dimensionality in short texts by generating coherent word clusters. GSDMM can converge more rapidly than LDA which makes it well-suited for hidden topics extraction on brief texts (Udupa et al., 2022).

In order for GSDMM to form word clusters, it offers a model called Movie Group Process. This model is an analogy of grouping items based on shared interests. The formed clusters by this model will be differentiated through each cluster differences. This model forms clusters considering two main goals, completeness and homogeneity.

The completeness goal, aims to ensure that each cluster include all the relevant members, with larger and more diverse clusters indicating popularity. On the other hand, homogeneity aims to ensure that each cluster is internally consistent, containing only members with closely related interests (Islami et al., 2023).

Movie Group Process model works with mainly three parameter which are *k*, *alpha* and *beta*. The *k* parameter represents the number of clusters that we wish the model to converge to. *Alpha* is a parameter that influences the likelihood of a document being inserted into a cluster, in other words, influences the probability distribution. The parameter *beta* will influence the distribution of words within each cluster, it will help determining the variety of words that are likely to be found in documents assigned to the same cluster (Udupa et al., 2022).

2.5.6 BERTopic

With BERTopic, the documents are embedded into a high-dimensional vector space with the usage of neural networks, making their representations semantically compared. On this process, it is assumed that documents with similar subjects will be positioned closely in the vector space. On the embedding stage, BERTopic uses Sentence-BERT which is a pre-trained language model to convert sentences into dense vector representations. These embeddings does not directly produce topics but they can be used to group semantically similar documents. Since the model can focus on syntactic correlations, the quality of generated topics is improved. When getting to the clustering phase, the dimensionality of the generated embeddings is reduced to improve efficiency.

When it comes to clustering, BERTopic makes use of HDBSCAN. HDBSCAN is an extension of DBSCAN that facilitates hierarchical clustering and allows for soft clustering, where noise can be modeled as outliers. This method is effective on the prevention of misallocation of

unrelated documents to clusters which can improve the overall quality of the topic representations. BERTopic also makes use of UMAP (Uniform Manifold Approximation and Projection) to reduce dimensionality. The usage of this method has been shown to improve clustering performance, preserving more of the local and global properties which are characteristic of high-dimensional data projected in lower dimensions. The combination of these techniques ensures that BERTopic can efficiently generate high quality topics even from complex and high-dimensional data.

Regarding topic representations, those are modeled based on the documents within each cluster. Each cluster is assigned a specific topic, and the relevance of words to those topics can be adjusted using TF-IDF. Usually, TF-IDF measures the importance of a word to its document, but on BERTopic, this metric can be adapted to reflect the significance of a term to a topic, improving the clarity of the topic. This approach ensures that the identified topics are not only semantically coherent but also meaningfully distinct from one another. (Udupa et al., 2022)

2.6 Matrix Factorization

Matrix Factorization technique is one of the most used techniques when applying collaborative filtering on recommendation systems. Its popularity is very based on its effectiveness when handling large datasets and the ease of implementation (Liu & Zhao, 2023) (Xu, 2019).

The main concept of this matrix is the assumption that the user's preferences and the item's preferences are, somehow, influenced by some latent factors that are the key to determine how users interact with items (Xu, 2019) (Liu & Zhao, 2023).

During the matrix factorization process, the user-item matrix is mapped into two or more lower dimensional matrices so the complexity of the data can be reduced (Liu & Zhao, 2023).

For a recommendation system with *n* users and *m* items, the rating matrix will be defined as R_{n*m}. This matrix will be decomposed now in two matrices U_{n*k} and V_{k*m} where *k* represents the dimension of the latent factors which are much smaller than *n* and *m* since it represents the reduced feature space in which users and items are characterized (Xu, 2019) (Liu & Zhao, 2023).

This matrices *U* and *V* are learned from an optimization process that minimizes the regularized mean squared error (RMSE) between the actual ratings and the predicted ratings that are calculated as the product of the latent factor vectors for users and items (Liu & Zhao, 2023).

Matrix Factorization is a powerful technique that simplifies the complexity of user-item interactions into manageable latent factors. This will help the recommendation system give more accurate predictions giving user preferences and item characteristics (Xu, 2019) (Liu & Zhao, 2023).

2.6.1 SVD

Singular Value Decomposition is a matrix factorization technique that is very used on collaborative filtering. It usually acts over a user-item matrix where each row is a user and each column is an item and each entry is the rating. When SVD decomposes this matrix, it gets the latent relationships between users and items, uncovering underlying patterns and preferences that are not immediately apparent in the data. This enables the recommendation system to predict how a user might rate unrated items.

The principle of SVD is to break down the original high-dimensional matrix into three smaller matrices that capture the essential features of the data while reducing noise and redundancy. The resulting lower-dimensional representation makes computations more efficient and improves the system's ability to generalize from known ratings to unknown ones. Despite this reduction, the factorization maintains the essential structure that is necessary for effective recommendations by focusing on the most important latent factors.

From a high-level perspective, SVD's matrix factorization aims to find two or more matrices whose product gets close to the original user-item rating matrix. These factorized matrices represent abstract concepts such as user taste profiles and item attribute profiles, making it easier to understand the interactions within the dataset (*2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2020).

2.6.2 NMF

Non-negative Matrix Factorization, was introduced by Lee and Seung and is a technique for uncovering hidden features within data by decomposing matrices into non-negative components. This approach has a good advantage where data is inherently non-negative, such as text analysis, where it is crucial to maintain the interpretability of the results.

By decomposing high-dimensional data into a set of non-negative matrices, NMF reveals latent structures that might otherwise remain hidden. This aspect of NMF makes it very valuable on topic modeling tasks.

NMF has also great advancements with the integration of sparseness, this is a good improvement to the quality of the decompositions it produces. Sparseness constraints help on getting more interpretable components by enforcing that only a few elements in the factorized matrices are non-zero, thereby ensuring that the components are more distinct and easier to interpret (Samprith Jagtap, 2024).

3 Dataset And Methodology

On this chapter it will be found a description of the dataset for sentiment analysis as well for the application of topic modeling. Along with this, this chapter will describe the models that were selected for both sentiment analysis and topic modeling and will also describe the metrics used for the model's performance evaluation. It will also be mentioned ethical considerations and data protection aspects.

3.1 Considered Datasets for Sentiment Analysis

For the sentiment analysis task, there were considered two different datasets:

- IMDb Dataset for Sentiment Analysis (Lakshmipathi N, n.d.)
- Massive Rotten Tomatoes Movies & Reviews (Andrea Villa, n.d.)

3.1.1 IMDb Dataset for Sentiment Analysis

The *IMDb Dataset for Sentiment Analysis* that was made available on Kaggle and contains a big collection of movie's reviews and it is strictly design for tasks related with natural language processing such and text analytics, more specifically, sentiment analysis.

The dataset is composed by 50000 movie reviews extracted from IMDb and those reviews can be categorized as positive or negative. The division of reviews was evenly made since there are 25000 reviews for each sentiment. This even division ensures that the models that will consume this data have a balanced representation of both positive and negative sentiments.

The classification of the sentiment on this dataset is binary, is either positive or negative. Each review is labelled with the respective classification. By making binary classifications, it makes straightforward for the training models to distinguish between positive and negative sentiments.

This dataset has a high usability score on Kaggle and that means that it is a very complete dataset with good credibility and compatibility. It includes essential metadata like the file format and descriptions that help users on their researches. It can be used for multiple ends, including recommendation systems.

It was made available in a CSV format and its size is around 27MB. Contains two columns which are the reviews and the sentiment label (positive or negative).

This dataset has an Apache 2.0 license and is a valuable resource for NLP researchers and developers because it offers content for training and testing sentiment analysis models. Its

balanced sentiment classifications, size and reviews that were actually made on IMDb platform, make this dataset a great choice for developing algorithms that are meant to classify human sentiments in the text.

Out[3]:

	review	sentiment
0	One of the other reviewers has mentioned that ...	positive
1	A wonderful little production. The...	positive
2	I thought this was a wonderful way to spend ti...	positive
3	Basically there's a family where a little boy ...	negative
4	Petter Mattei's "Love in the Time of Money" is...	positive

Table 1 - Top 5 IMDb Reviews

Out[4]:

	review	sentiment
count	50000	50000
unique	49582	2
top	Loved today's show!!! It was a variety and not...	positive
freq	5	25000

Table 2 - IMDb Dataset Description

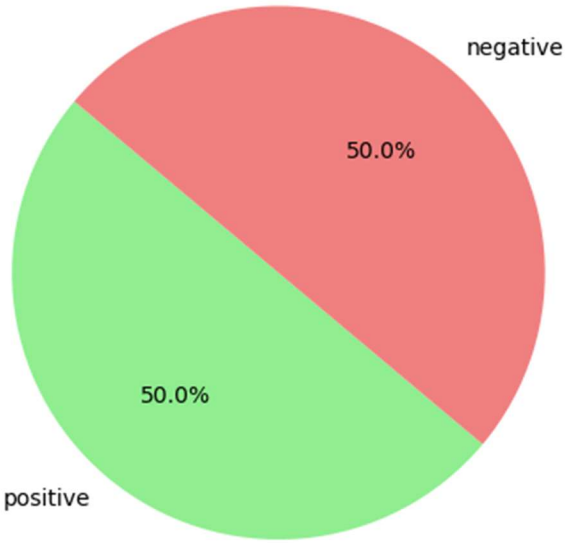

Figure 1 - Sentiment Distribution IMDb Dataset

3.1.2 Massive Rotten Tomatoes Movies & Reviews

This dataset, along with the *IMDb Dataset for Sentiment Analysis* was made available on Kaggle and also contains a great collection of movie data, reviews and ratings. It contains information that can fit really well on the sentiment analysis purpose.

Rotten Tomatoes is platform that is mainly used for critic reviews. On this platform, users can compare scores given by other reviewers and scores given by professional critics who are actually accredited members of various writing guilds or film critic organisations.

This dataset contains around 1.4 million reviews that were extracted from Rotten Tomatoes platform and those reviews are also binary (positive or negative). The division is not quite balanced on this dataset. We can verify that 67% of the critics are positive and 33% are negative.

When downloading this dataset, it comes with two CSV files on which only one of them contains the actual reviews of the movies. The size of the dataset is around 400MB. The file with the reviews contains 11 columns such as id (identifier for each movie), the critic review id, the creation date of the review, the name of the critic who wrote the review, a boolean that indicates if the critic is a top critic or not, the provided score by the critic, the state of the review, the name of the review, the review itself and the score sentiment that determines if a review is positive or negative.

Out[3]:

	id	reviewId	creationDate	criticName	isTopCritic	originalScore	reviewState	publicatioName	reviewText	scoreSentiment
0	beavers	1145982	2003-05-23	Ivan M. Lincoln	False	3.5/4	fresh	Deseret News (Salt Lake City)	Timed to be just long enough for most youngste...	POSITIVE
1	blood_mask	1636744	2007-06-02	The Foywonder	False	1/5	rotten	Dread Central	It doesn't matter if a movie costs 300 million...	NEGATIVE
2	city_hunter_shinjuku_private_eyes	2590987	2019-05-28	Reuben Baron	False	NaN	fresh	CBR	The choreography is so precise and lifelike at...	POSITIVE
3	city_hunter_shinjuku_private_eyes	2558908	2019-02-14	Matt Schley	False	2.5/5	rotten	Japan Times	The film's out-of-touch attempts at humor may ...	NEGATIVE
4	dangerous_men_2015	2504681	2018-08-29	Pat Padua	False	NaN	fresh	DCist	Its clumsy determination is endearing and some...	POSITIVE

Table 3 - Top 5 Rotten Reviews

	id	reviewId	creationDate	criticName	isTopCritic	originalScore	reviewState	publicatioName	reviewText	scoreSentiment	
count	1444963	1.444963e+06	1444963	1444963	1444963	1009745	1444963	1444963	1375738	1444963	
unique	69263	NaN	8510	15510	2	1729	2	2707	1359771	2	
top	parasite_2019	NaN	2000-01-01	Emanuel Levy	False	3/5	fresh	New York Times	Parental Content Review	POSITIVE	http://www.j
freq	954	NaN	48083	13274	1008156	116711	963799	19853	236	963799	
mean	NaN	9.035203e+06	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
std	NaN	2.575716e+07	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
min	NaN	1.000000e+00	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
25%	NaN	1.610366e+06	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
50%	NaN	2.200337e+06	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
75%	NaN	2.587024e+06	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
max	NaN	1.027962e+08	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

Table 4 - Rotten Dataset Description

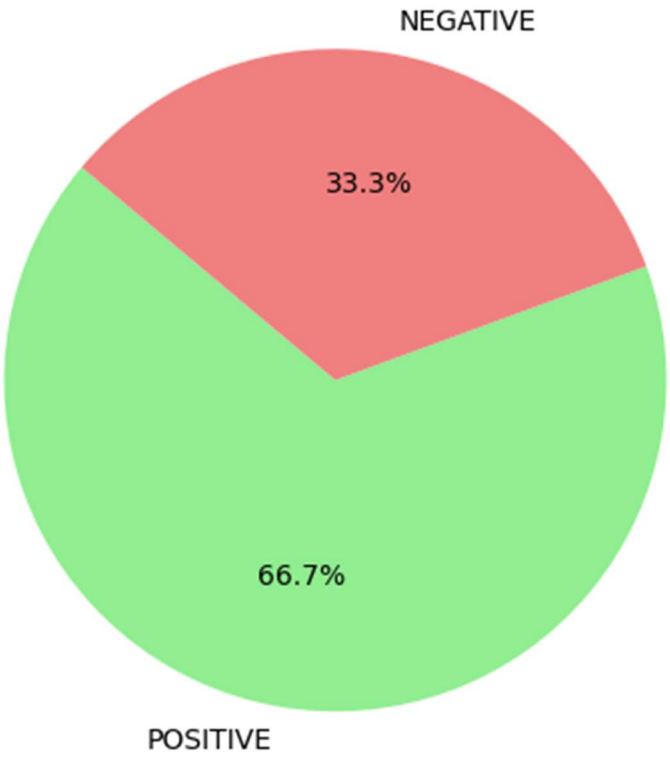

Figure 2 - Sentiment Distribution Rotten Dataset

3.2 Data Pre-processing

The selected datasets can help on both sentiment analysis and topic modeling for a context of movies because they are inserted in the context of movies so, they will help the model to properly classify sentiment from a review and extract topics that are related with movies. Thus, it is very important to guarantee the quality of the data that is injected to our models. In order to the models to get the best performance, there is a need to treat the data to ensure that no noise is being inserted so the model can give us the best classifications.

The steps that were followed on the datasets are:

- Cleaning of HTML tags, punctuation and numbers
- Tokenization
- Normalization to convert all words to lowercase
- Removal of stop words
- Lemmatization

3.2.1 Cleaning

The step of cleaning involves the removal of noisy and unnecessary content of the reviews such as HTML tags, punctuation and numbers. The removal of this elements can improve significantly the accuracy since it allows the model to focus on the words and elements that really matters.

Considering the top 5 reviews of the *IMDb Dataset for Sentiment Analysis,* it could be checked that there were com HTML tags like *
* and some elements like “ and ‘ that can simply be removed from the reviews.

After applying those removals, a new column on the dataset was created with the cleaned data.

On the table 5, is possible to verify on the left side column that contains the elements that were mentioned to be removed and on the right-side column, those elements are no longer present on the reviews.

Table 5 - Cleaned Review on IMDb Dataset

	review	sentiment	clean_review
0	One of the other reviewers has mentioned that ...	positive	One of the other reviewers has mentioned that ...
1	A wonderful little production. The...	positive	A wonderful little production The filming tech...
2	I thought this was a wonderful way to spend ti...	positive	I thought this was a wonderful way to spend ti...
3	Basically there's a family where a little boy ...	negative	Basically theres a family where a little boy J...
4	Petter Mattei's "Love in the Time of Money" is...	positive	Petter Matteis Love in the Time of Money is a ...

3.2.2 Tokenization

The tokenization process is important to steps that come ahead. By tokenizing a sentence, it is easier to apply stemming or lemmatizing operations to the words and, before that, applying normalization to those tokens like converting all the text to lowercase.

It was created a new column on the dataset that contains a list of tokens that are present on the respective review.

On the table 6, it is possible to see the last column that contains a list with all the tokens present on the cleaned review.

Table 6 - Tokenization on IMDb Dataset

	review	sentiment	clean_review	tokenized_review
0	One of the other reviewers has mentioned that ...	positive	One of the other reviewers has mentioned that ...	[One, of, the, other, reviewers, has, mentione...
1	A wonderful little production. The...	positive	A wonderful little production The filming tech...	[A, wonderful, little, production, The, filmin...
2	I thought this was a wonderful way to spend ti...	positive	I thought this was a wonderful way to spend ti...	[I, thought, this, was, a, wonderful, way, to,...
3	Basically there's a family where a little boy ...	negative	Basically theres a family where a little boy J...	[Basically, theres, a, family, where, a, littl...
4	Petter Mattei's "Love in the Time of Money" is...	positive	Petter Matteis Love in the Time of Money is a ...	[Petter, Matteis, Love, in, the, Time, of, Mon...

3.2.3 Normalization

The normalization step consists on grabbing all the tokens that were extracted from the reviews on the last step and convert all of those on lowercase.

By applying this lowercase conversion on the tokens, we are ensuring that the model, when consuming the data, is going to treat equal words in an identical way. If we have the same word written in different ways like “love”, “Love” or “LOVE”, when applying this normalization, the model will look at these words equally.

A new column, similar to the tokenized review column was created but, in this new column all the words will be in lowercase.

On the table 7, it is possible to compare the original tokens and the lowercase tokens.

tokenized_review	lowercase_review
[One, of, the, other, reviewers, has, mentione...	[one, of, the, other, reviewers, has, mentione...
[A, wonderful, little, production, The, filmin...	[a, wonderful, little, production, the, filmin...
[I, thought, this, was, a, wonderful, way, to,...	[i, thought, this, was, a, wonderful, way, to,...
[Basically, theres, a, family, where, a, littl...	[basically, theres, a, family, where, a, littl...
[Petter, Matteis, Love, in, the, Time, of, Mon...	[petter, matteis, love, in, the, time, of, mon...

Table 7 - Review's Tokens with lowercase

3.2.4 Removal of Stop Words

Along with the first step, *Cleaning*, this one is also very important in order to allow the model to focus on the words that matters. By removing stop words like “the”, “is” and so on, we are helping the model improving its accuracy since it has no focus on words that do not bring any additional value to the review.

Since all the reviews on both datasets were written in English, the removal of stop words was made by having in consideration the English language.

On this step, another column was created where tokens that match stop words from the list of each review’s tokens.

lowercase_review	stopwords_removed_review
[one, of, the, other, reviewers, has, mentione...	[one, reviewers, mentioned, watching, oz, epis...
[a, wonderful, little, production, the, filmin...	[wonderful, little, production, filming, techn...
[i, thought, this, was, a, wonderful, way, to,...	[thought, wonderful, way, spend, time, hot, su...
[basically, theres, a, family, where, a, littl...	[basically, theres, family, little, boy, jake,...
[petter, matteis, love, in, the, time, of, mon...	[petter, matteis, love, time, money, visually,...

Table 8 - Stop Words removal from IMDb Dataset

3.2.5 Lemmatization

The step of lemmatization is the step that will reduce all the tokens present on the review to their base or dictionary form.

Lemmatization is an important step on pre-processing because, when reducing words into their original forms, we are allowing the model to look at different forms of the word in an equal way, by doing that we are simplifying the analysis. Since all words are reduced to their original form, the model will treat the words with similar meanings equally independently of the form that they are injected for analysis on the input. When reducing the words form, we are also reducing the feature space which removes complexity to the model and improves efficiency.

Another column was created for the lemmatized tokens on the dataset. On the table 9, it is possible to validate that some of the tokens present on the column of stop words removal suffered lemmatization.

stopwords_removed_review	lemmatized_review
[one, reviewers, mentioned, watching, oz, epis...	[one, reviewer, mentioned, watching, oz, episo...
[wonderful, little, production, filming, techn...	[wonderful, little, production, filming, techn...
[thought, wonderful, way, spend, time, hot, su...	[thought, wonderful, way, spend, time, hot, su...
[basically, theres, family, little, boy, jake,...	[basically, there, family, little, boy, jake, ...
[petter, matteis, love, time, money, visually,...	[petter, matteis, love, time, money, visually,...

Table 9 - IMDb Reviews Lemmatization

After all the pre-processing we can now illustrate a word cloud with the most frequent words on IMDb and Rotten datasets.

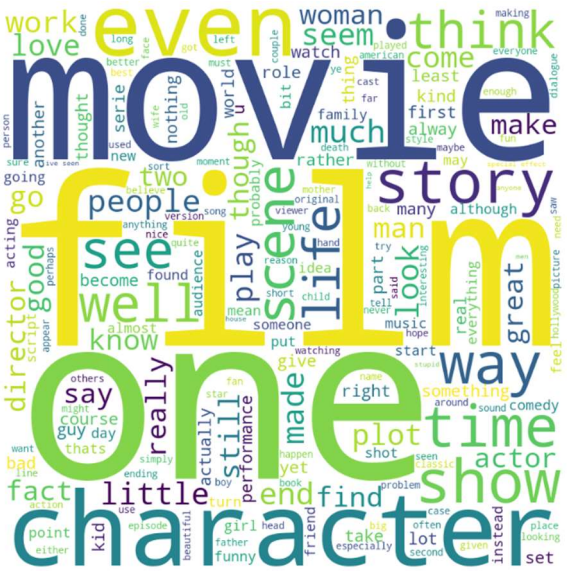

Figure 4 - IMDb Dataset Word Cloud

Figure 3 – Rotten Dataset Word Cloud

All the mentioned steps above were made on both *IMDb Dataset for Sentiment Analysis* and for *Massive Rotten Tomatoes Movies & Reviews*. For the Massive Rotten Tomatoes Movies & Reviews, it was needed some more steps:

- Feature Reducing
- Data Pruning

3.2.6 Feature Reducing

Feature reducing is about selecting only the features that are actually going to be useful for the model. This dataset is a very large dataset and contains more features than necessary for sentiment analysis and topic modeling.

On the table 10, is possible to see all the present features on this dataset. For the relevant tasks of sentiment analysis and topic modeling for the context of the recommendation system, the only features that can be useful are the reviews and the respective sentiment.

Table 10 - Rotten Dataset Total Features

id	reviewId	creationDate	criticName	isTopCritic	originalScore	reviewState	publicatioName	reviewText	scoreSentiment	reviewUrl
1444963	1.444963e+06	1444963	1444963	1444963	1009745	1444963	1444963	1375738	1444963	1234038
69263	NaN	8510	15510	2	1729	2	2707	1359771	2	1138350

After proceeding with this operation, a new dataset containing only reviews and their respective sentiment was created.

	reviewText	scoreSentiment
count	1375738	1444963

Table 11 - Reduced Rotten Dataset

3.2.7 Data Pruning

On the figures above, it can be verified that the number of reviews does not match the number of sentiments. This clearly means that, on this dataset, there are rows with empty reviews but still with an associated sentiment.

Since the review itself is the primary source of information for a sentiment analysis task and, in the context of this recommendation system, topic modeling task, an empty review is not actually contributing with meaningful data for the sentiment classification.

After analysing the number of null reviews, it was checked that the dataset contained 69225 records with a null review.

Considering that the total number of rows on this dataset was about 1.4 million records, the removal of these rows would not cause a major impact.

The table 12 contains the description of the dataset after the removal of the records with null reviews. It is possible to check now that the number of reviews does now match with the number of sentiments.

Table 12 - Rotten Dataset After Null Rows Removal

	reviewText	scoreSentiment
count	1375738	1375738
unique	1359771	2
top	Parental Content Review	POSITIVE
freq	236	922510

It was also possible to verify that, with the removal of this rows, the distribution of sentiments had a very small change of 0.4% where the percentage of positive reviews increased and the percentage of negative reviews decreased.

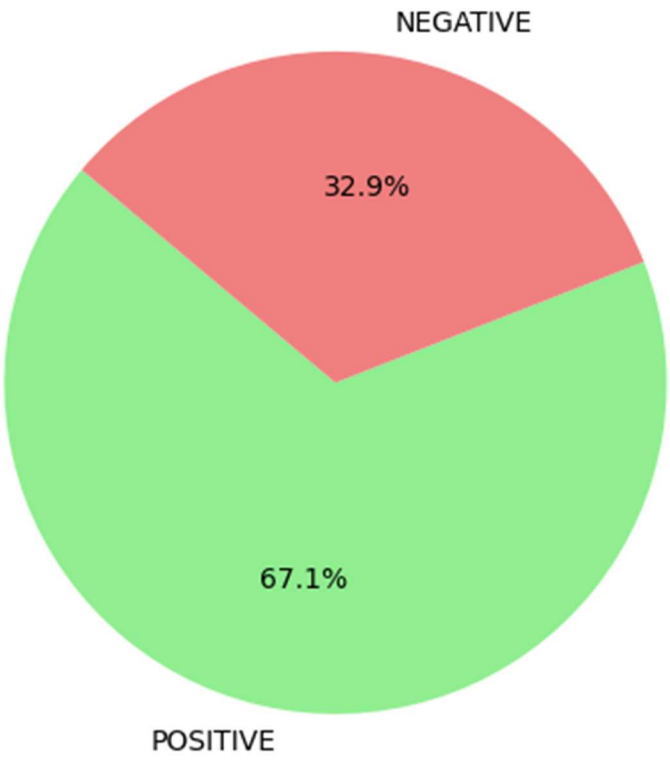

Figure 5 - Sentiment Distribution Rotten Dataset After Removing Rows

3.3 Methodology

The model implemented on the recommendation system puts together three different tasks. Those tasks are sentiment analysis, topic modeling and the correlation between that data using matrix factorization.

The training data and the testing data will be differently distributed. The training data will have between 70% and 80% of the data and the testing data will have an amount of data between 20% and 30% of the total data. The two datasets will be tested separately to evaluate which one can give the models the best performance.

For the testing phase, the pre-processed data will be injected on the models so they can learn from the data. After training the model it will be a validation phase where the values of the model's hyperparameters will be changed in order to understand if the model's performance can be improved. Finally, the model's performance is going to be tested with an unused set of data and some performance metrics will be applied to measure the model's effectiveness.

3.3.1 Sentiment Analysis

To the sentiment analysis task there will be tested traditional machine learning models such as Naïve Bayes and Logistic Regression. A deep learning model, BERT, will also be tested to perform sentiment analysis since it is a model that brought great advancements to NLP tasks.

For the BERT model, some evaluation metrics will be applied such precision, recall and F1-Score. Precision can be important to ensure that the positive sentiments that were identified are truly positive and the recall ensures that most of the positive sentiments were able of being captured. The F1-Score is the balance between precision and recall.

The considered metrics to evaluate Naïve Bayes model were accuracy, precision, recall and F1-Score. Accuracy is a very straightforward metric to measure the correctly classified instances out of all classifications and it can measure the overall performance of the model. The precision, recall and F1-Score were also selected since it is important to guarantee that the model was able to classify the sentiments correctly.

To evaluate Logistic Regression model were accuracy and ROC-AUC. Accuracy to give an overall performance of the model and the ROC-AUC because Logistic Regression provides probabilities of a class membership. Since that in sentiment analysis the distinction between classes may not always be straightforward, ROC-AUC helps identify how well the model was able to distinguish between classes.

3.3.2 Topic Modeling

For the topic modeling task, two models will be tested and those models will be BERTopic and GSDMM. Since movies overviews and user's reviews commonly have a small amount of text, it would be interesting to make a comparison between the mentioned models to understand which can fit best on the context of movies reviews.

The selected metric to evaluate these models is topic coherence. The topic coherence metric will measure the degree of semantic similarity between high scoring words in the topic. If the coherence is high, it means the topic is more interpretable.

Both BERTopic and GSDMM will make use of topic coherence because GSDMM can produce a variable number of topics and it is important to guarantee coherence between them. It also can involve a qualitative analysis that involves a manual evaluation of the top words of each topic and validate coherences.

3.3.3 Matrix Factorization

The considered models to apply matrix factorization was SVD and NMF. The most common metrics and the considered metrics to validate the matrix factorization are RMSE (Root Mean Square Error) and MAE (Mean Absolute Error).

RMSE will measure the error of a model when predicting quantitative data. On the recommendation systems case, it will measure the difference between predicted and actual ratings. The lower the RMSE, the better the performance

MAE will measure the average magnitude of errors in a set of predictions. This metric is less sensitive to outliers compared to RMSE and provides a simpler interpretation.

RMSE and MAE can both be used as loss functions in the optimization process of the SVD algorithm.

3.4 Ethical Considerations and Data Protection

As we know, recommendation systems are algorithms that aims to provide suggestions of content that the user may like considering that user's preferences. With the evolution of technology and the increase of its demand, these recommendation systems have been showing great impact. By having big impact, some ethical considerations must be considered.

One of the challenges for recommendation systems could be suggesting inappropriate content where some studies were made in order to propose systems that can actually filter content based on cultural and ethical matters. Privacy, is another ethical challenge since some private data is usually used on these kinds of systems. The system must be designed in way that guarantees user's data protection and privacy. Another ethical consideration would be the user's autonomy and personal identity. The recommendation systems can impact user's autonomy by leading them towards a certain content or limiting exposure to another options. Opacity is another ethical topic to have in consideration by explaining how personalized recommendations are generated to increase transparency. Social effects can also be an ethical consideration since recommendation systems can create filter bubbles that may lead to self-

reinforcing biases and impacting public debate and democratic institutions (Milano et al., 2020).

Recommendation systems are design to provide content to the users with their preferences into consideration. The more data the system has about the user's preferences, that can be collected through reviews, ratings or even through search history, the more accurate will be the provided recommendations.

Since private data, such as user's preferences, needs to be supplied to the recommendation system for it to work properly, there must be cautions with that data so it doesn't get leaked or exposed avoiding risks to the privacy of the users (Wu & Gong, 2021).

To protect the user's data, all the personal data that is present on the training datasets must be deleted and only the reviews and the sentiment will be made available for training and test effects.

For the recommendation system to work properly, it needs to establish a connection between the user and its preferences, for this a profile must be created so that the system can perform accurate recommendations based on the user's profile. Having this in consideration, the profile can correlate the user with its preferences through a simple identifier. This way, the model will have the necessary data to work properly and will know the preferences of the user without really knowing who it is since private data such as name, birth date, gender and so on is not available to the model.

It also must be considered that the users should always have control on their data, including the right to access it, rectify and delete information.

3.5 Conclusion

Both selected datasets were extracted from Kaggle and both datasets can be of great use for a recommendation system since they both contain reviews and the sentiment that is related to that review, making it easy to train models to classify sentiment behind reviews. Both datasets are also useful to extract other kinds of information such as the topic that those reviews are related to.

By having datasets that have the ability to train different models of different tasks, a final model can be built to make the correlation between the sentiment and the related topic so the recommendation system can make rating predictions based on a user profile that consists on those correlations.

The *IMDb Dataset for Sentiment Analysis* contains 50000 records of reviews and the linked sentiment classification with a very even distribution (50% positive and 50% negative) and the *Massive Rotten Tomatoes Movies & Reviews* have more than 1.4 million records with a

distribution of 67% positive records and 33% negative records. The fact that these two datasets have a different distribution of sentiment classes, allows to experiment different approaches on the models to handle that imbalance along with the application of different evaluation metrics.

The datasets went through a pre-processing phase in order to arrange the data for it to be more understandable for the model and to help it reach the best results. The pre-processing consisted on several phases like Cleaning where all HTML tags, symbols and numbers were removed, the tokenization phase to split all sentences into tokens, normalization to convert all characters for lowercase so the model can look to the same word in the same way even if the user writes it in uppercase, removal of stop words to remove words that are not useful for the model to perform classification and lemmatization to reduce to words to their root form. *The Massive Rotten Tomatoes Movies & Reviews* needed two extra steps on the pre-processing which were the Feature Reducing to remove features from the dataset that were not going to be useful for the models and Data Pruning to delete rows with null reviews.

Different models will be tested for both sentiment analysis and topic modeling tasks. For sentiment analysis, traditional machine learning models like Naïve Bayes and Logistic Regression were selected and will be evaluated using metrics such as accuracy, precision, recall, F1-Score and ROC-AUC. A deep learning model, BERT, will also be tested on this context and is going to be evaluated using precision, recall and F1-Score metrics. Regarding topic modeling tasks, there were selected the algorithms BERTopic and GSDMM that are both techniques that are widely used for topic modeling with GSDMM more designed for short text processing. These topic modeling techniques will be evaluated with the metric topic coherence and a more manual approach like qualitative analysis. The matrix factorization will make use of the SVD and NMF algorithm and will be evaluated with metrics like RMSE and MAE.

4 Recommendation System Model

The goal of this chapter is to provide a detailed description of the practical steps that are involved on the development of the proposed recommendation system. It will outline the stages of implementation including sentiment analysis, topic modeling, the creation of *user-topic* and *movie-topic* matrices and the factorization of those matrices. The integration of these components will also be approached on this chapter to form the recommendation system itself.

To start this chapter, it will be introduced an overview of the system architecture, showing how each component interacts to form the final solution. Since all the steps that were performed on text processing were described on the previous chapter, this one will not get into a very detailed description of those steps. Instead, the pre-processing steps will only be mentioned as part of the recommendation system process.

During this chapter, it will be explained how the sentiment analysis and topic modeling techniques were applied as well as the different approaches that were experimented for both. After detailing those implementations, it will be explained how the information returned for those models will be used to form the *user-topic* and *movie-topic* matrices. The matrix factorization process will also be covered including the choice of factorization techniques like NMF and SVD as well as the motivations that led to the choice of the number of latent factors.

Lastly, it will be discussed the implementation of the recommendation system by focusing on how the factorized matrices are used to generate the user's recommendations.

By the end of this chapter, we are going to have a better understanding on how the system was developed and on the next chapter it will be presented the results of the proposed solution by this chapter.

4.1 Model High Level Architecture

The figure 18 represents the flow of the model. It starts by collecting the review and the necessary information for the construction of the matrices such as, userId and movieId. After gathering the review, it will be pre-processed where it will be removed symbols, numbers, punctuation, stop words, all tokens will be transformed into lower case and it will be applied lemmatization.

Once the review is ready to be used, it will be sent to the topic modeling task and for the sentiment analysis task which will also retrieve important information for the matrices such as the topic of the review and the associated sentiment.

After the review goes through all the needed tasks, the matrices user-topic and movie-topic will be created based on the userId, movieId, the sentiment and the topic. The user-topic matrix will represent the relevance of a topic to a user where the rows will be users and columns will be the topics. The movie-topic matrix will represent how much a topic is associated with a movie where the rows are the movies and the columns are the topics.

The next step will be the extraction of the latent factors of the previous created matrices. Those latent factors will give hidden patterns that gives information of the preferences of the users.

Once the latent factors are extracted, they will be used to calculate the preference score for each user-movie pair.

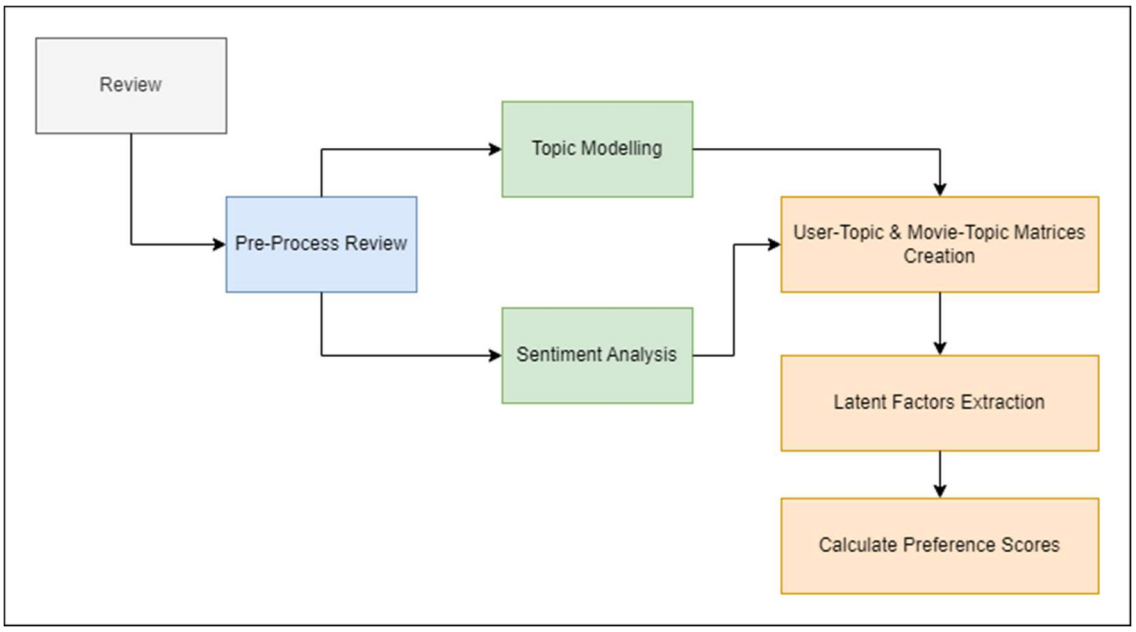

Figure 6 - Model High Level Diagram

4.2 Sentiment Analysis

The sentiment analysis process is a critical component of the recommendation system. By performing the sentiment analysis on a user review we can understand how the user feels about the topic mentioned on the review that will be related to the movie. This sentiment information will be incorporated into the *user-topic* and *movie-topic* matrices. The sentiment score for each review was used to adjust the influence of the topics in the matrices and, finally, influencing the final recommendations.

For the sentiment analysis task, multiple models were tested such as Logistic Regression, Naïve Bayes and BERT.

For each tested model on the sentiment analysis task, the dataset was splitted into a training subset and into a testing subset using the *train_test_split* module from Scikit-learn (*Scikit-Learn*, n.d.).

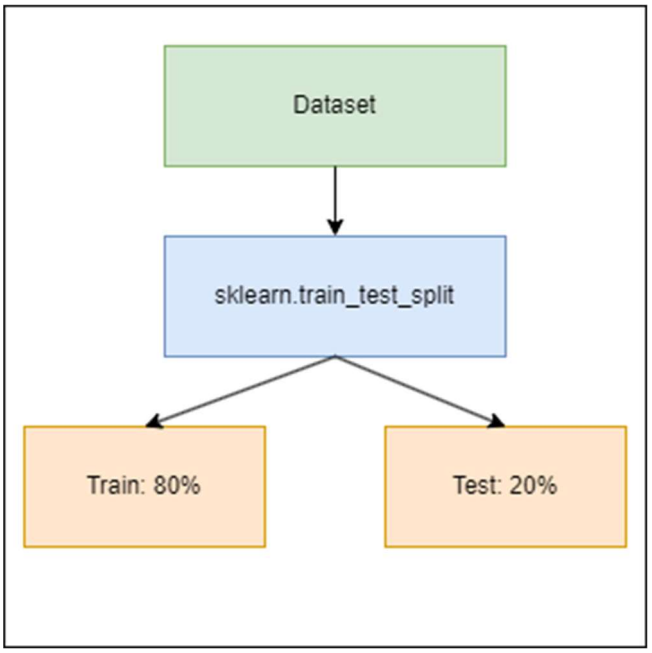

Figure 7 - Splitting Data into Training and Testing Subsets

4.2.1 Logistic Regression

For the Logistic Regression implementation, the most used python library was Scikit-learn. Besides the Logistic Regression itself, from this library it was also made usage of TF-IDF vectorizer and Count Vectorizer.

The model can have different results and perform differently when using different vectorizers. So, logistic regression model was trained in multiple ways using different vectorizers such as TF-IDF and Count Vectorizer. It was also tested the integration of the logistic regression model with the usage of embedding models such as Word2Vec.

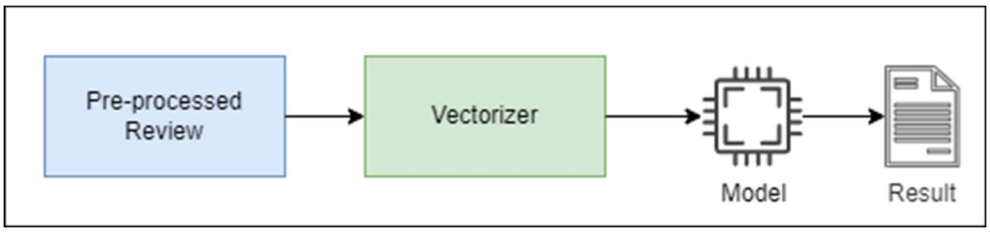

Figure 8 - Basic Logistic Regression Pipeline

The basic pipeline to implement the logistic regression model, as presented on the image, starts by gathering the user's pre-processed review. This pre-processed review was already made available by the previous described pre-processing steps on the previous section, so this review is already in lower case, without symbols, numbers and punctuations, without stop words and with each token already lemmatized.

This pre-processed review will be inserted into our vectorizer. There were tested two different vectorizers, TF-IDF vectorizer and Count Vectorizer. Besides both vectorizers will create a numeric representation of the documents they will generate those representations in a different way.

TF-IDF vectorizer will focus on the frequency of a term in a document and on the rarity of that term in another documents. Count vectorizer will only focus on the gross frequency of a term in the documents. Since the approach of both vectorizers is different when creating numeric representations of the document, the results between them are expected to be different.

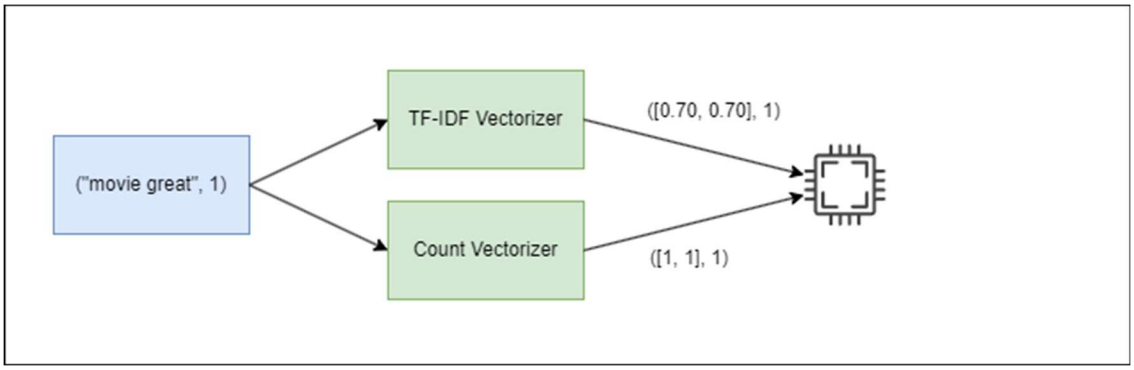

Figure 9 - Vectorization Representation

The vectorizers themselves were tested in two different ways. When creating representations using individual tokens, it may result on loss on contextual information. By using *n-grams*, we are allowing the vectorizers to capture more contextual information which can improve the interpretability of the reviews. This way, both vectorizers were tested with and *without n-grams*. The range of the *n*-grams is defined on the exact moment when the vectorizer is initialized.

After gathering the numerical representations of the documents, it begins the training phase of the Logistic Regression model. To help on this phase, the Scikit-learn's library GridSearchCV was used to help discover the best trained model.

To help on the discovery of the best model, it was initially defined a list of values for each model's hyperparameter. For Logistic Regression model, it was defined a list for the hyperparameter *C* and for the hyperparameter *penalty*. The hyperparameter *C* is a parameter that will control the model's regularization by penalizing coefficients to avoid overfitting. The hyperparameter *penalty* defines the type of regularization will be used (L1 or L2), L1 will add a penalty that is proportional to the absolute sum of the coefficients and L2 will add a penalty proportional to the squares of the coefficients.

The GridSearchCV will create all the possible parameter combinations and will select the one model with the best results. To each combination, it will be applied a cross validation and the best model will be saved.

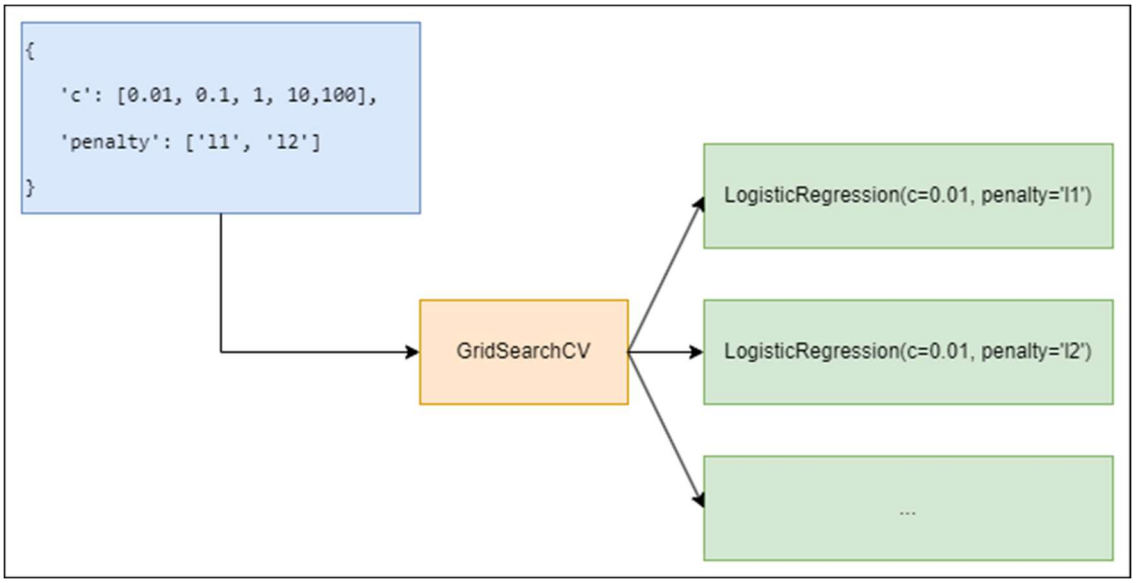

Figure 10 - GridSearchCV with Logistic Regression

After the GridSearchCV gives us the best estimator, the model went to the testing phase where it was exposed to unseen data and where the predicted sentiment by the model was compared with the actual sentiment of the review.

The extraction of these metrics were also possible with the usage of the module *Metrics* of the Scikit-learn library.

4.2.2 Logistic Regression with Embeddings (Word2Vec)

Similarly to the previous Logistic Regression sub-section, to start the experiment the process starts by gathering all the already pre-processed reviews.

Once the data is gathered, the Word2Vec model is initialized with vector size of three hundred, this means that each vector generated for each token, will have a size of three hundred. The greater the number of dimensions, the better the semantic relationships are captured, but, in the other hand, it will need more computational cost.

Once the model is initialized and trained, it is used to transform the reviews into vectors. Those vectors are going to be divided into training data and testing data and then they are going to be used by the Logistic Regression model for both phases.

During the training phase of the Logistic Regression model, the library GridSearchCV was also used to help perform hyperparameter combinations in order to find the best model. The parameters that were changed on this process were the same that were on the previous Logistic Regression process and with the exact same values.

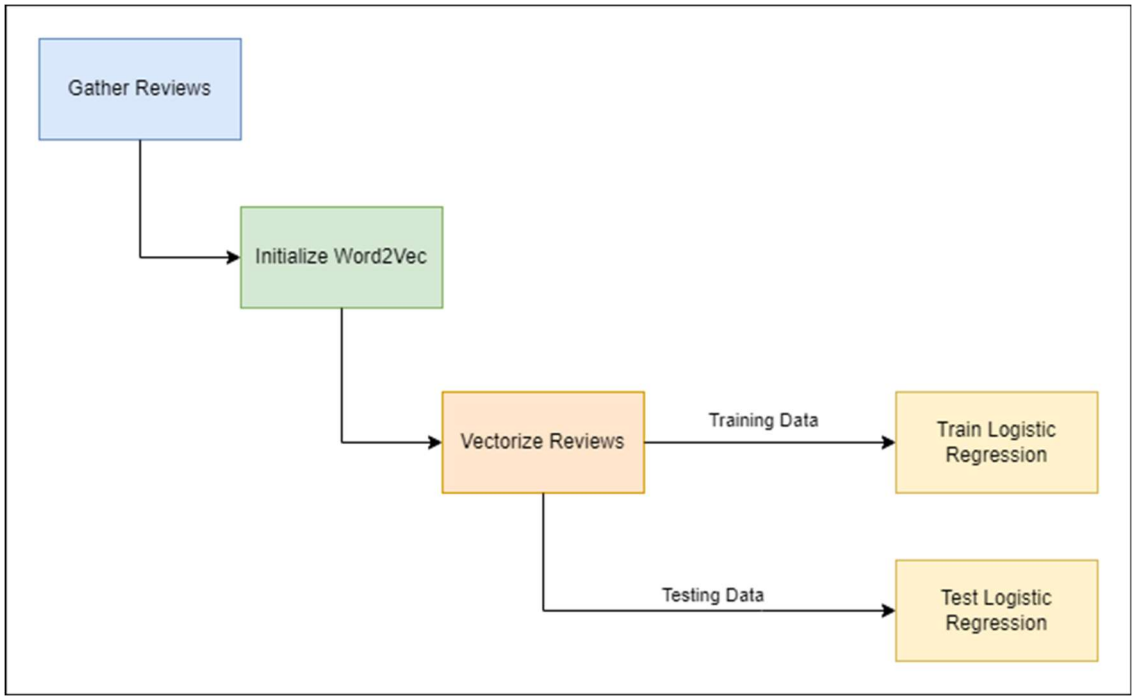

Figure 11 - Flow Word2Vec and Logistic Regression

After the conclusion of the training phase, it was performed the test and the validation of the best returned model by GridSearchCV.

4.2.3 Naïve Bayes

To apply the Naïve Bayes model, it was used the module *MultinomialNB* that is made available by the python's Scikit-learn library.

Although there are more Naïve Bayes variants besides, the Multinomial variant fits better for the sentiment analysis tasks or text classification tasks since it has the capacity to deal with word frequency. As the tokens that are present on the reviews will be transformed into a Bag-of-Words or TF-IDF, each feature will represent the frequency of a particular word in the text.

As for Naïve Bayes, the development process was quite similar. The process starts by gathering the pre-processed reviews and then split the data into training subsets (80%) and testing subsets (20%).

With the divided data, vectorization is applied to both subsets. The vectorizers applied were the same as for the Logistic Regression process, TF-IDF and Count Vectorizer. Both were also tested with n-grams and without n-grams.

To train the model, GridSearchCV was also used to try different values for the hyperparameters. In this case, only one hyperparameter was changed and that parameter was *alpha*. On Naïve Bayes the alpha parameter is a smoothing parameter. It will help handle the

case off zero probabilities in the training set. The lower the alpha, there are more chances for the model to overfit.

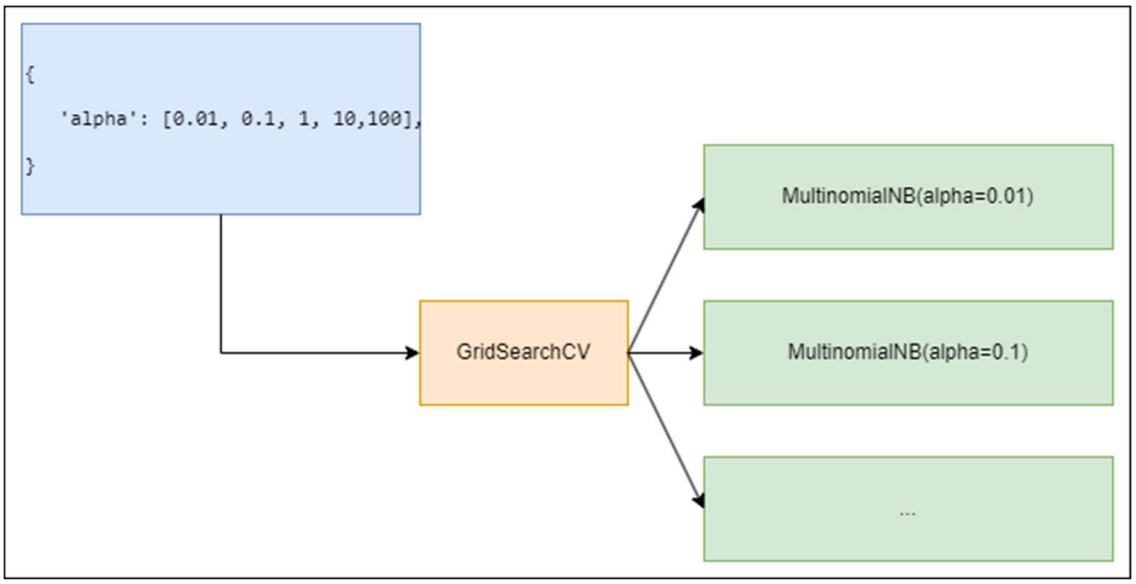

Figure 12 - GridSearchCV with Naive Bayes

The GridSearchCV was also able to return the model with the best results after applying cross validation to all the possibilities.

Finally, the best estimator returned by the GridSearchCV was used to perform the model testing and validations by exposing it to new data which was present on the subset of testing data.

Similarly to Logistic Regression, the metrics were extracted through the usage of the metrics module provided by Scikit-learn library.

4.2.4 BERT

Although the process applied on BERT is also very similar, the tools that were applied were a bit different than the ones applied for Logistic Regression and Naïve Bayes.

The library *transformers* that was applied on this process provides more than one useful tool. Those tools are:

- BertForSequenceClassification
- AdamW
- BertTokenizer

The BertForSequenceClassification is a model that is based on BERT and was specifically fine-tuned for classification tasks like sentiment analysis, spam detection. BERT itself, has the ability to capture both left and right context simultaneously in a bidirectional manner, which makes it very efficient to understand natural data such as text. BertForSequenceClassification tries to add to BERT the ability to perform classifications. To do this, the output of the original BERT is injected into a feedforward neural network, which is the classification head, and the classification is made.

The AdamW is an algorithm that will help BERT on its optimization. It applies a weight decay as a regularization technique that will penalize large weights in the model by adding a penalty to avoid overfitting.

The BertTokenizer is the model that is responsible for pre-processing text specifically for BERT. It converts raw text into tokens that will be fed into the BERT model since it requires the inputs to be tokenized into specific formats.

To the process of sentiment analysis task, the beginning is very similar. It starts by gathering the reviews that were already pre-processed and used by the previous models and by dividing the data into training subset and testing subset.

The next steps are the ones that are quite different. The difference here is that the vectorizers Count Vectorizer and TF-IDF will not be used. Instead, BertTokenizer was used to convert the inputs into the specific format that BERT requires.

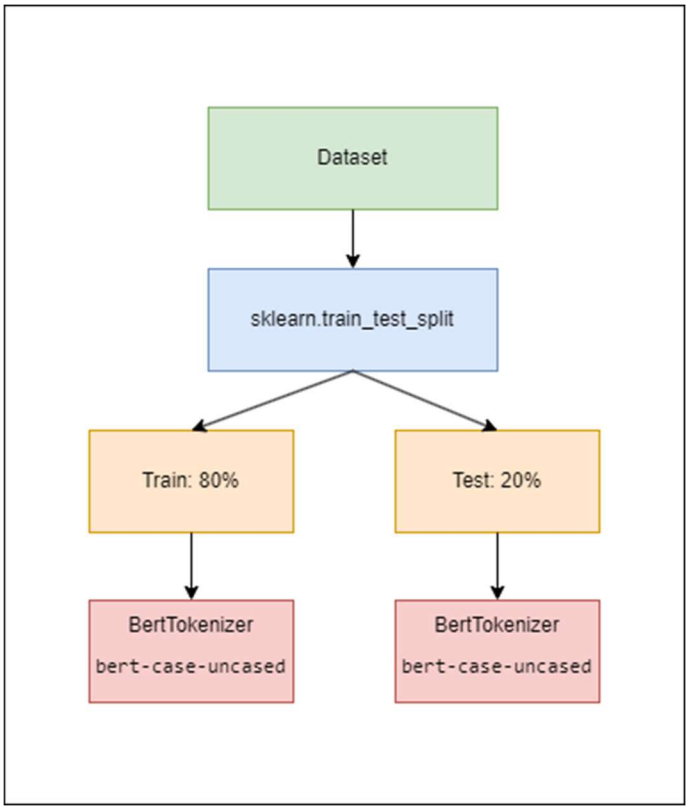

Figure 13 - Usage of BertTokenizer

After gathering the encodings provided by the BertTokenizer, it starts to enter on training phase of the model. To train the model, the library *Torch* was used. It was created a class called *ReviewsDataset* and this class will make use of the *Torch's* module, *Dataset*. By having this custom class using *Dataset* module it allows that data to be converted into tensors and be inserted into the model during the training process and also allows PyTorch to create batches of our data during the training (*PyTorch Website*, n.d.).

To make the training phase faster, torch also gives the possibility to transfer the training task to the GPU using CUDA. The model was trained with the CPU and the GPU. With the CPU, the model took around thirty hours to be trained. In the other hand, with the GPU, the model reduced the training time significantly to seventy minutes using the IMDB dataset.

After setting up the training data and setting up the AdamW optimizer, the model started to be trained. The model was trained in ten epochs where each epoch took around seven minutes to be completed, making a total of around seventy minutes on the end of all the ten epochs. As for the hardware used to train BERT:

- CPU: AMD Ryzen 5 3600 6-Core Processor, 3.60 GHz
- GPU: NVIDIA GeForce RTX 3060 Ti
- RAM: 16GB

As made on the previous models, this one was also tested and validated. To do so, the testing data was also ingested into the model so it can start making predictions.

4.3 Topic Modeling

In this recommendation system, topics were used to extract from the user's reviews some insights of what it could be some of its interests. Whenever a user makes a review to a movie, the topic of the review gets related with the user that reviewed and movie that was reviewed, contributing for the influence of each topic on each user and movie. The influence of the extracted topic will be also influenced by the analysis if the sentiment that was approached on the previous sub-chapter.

To perform topic modeling experimentations, there were selected two models which were:

- GSDMM, which is a model that is particularly effective for short documents such as reviews and groups documents into clusters. Those clusters will be represented as topics
- BERTopic, which is a model that can be used for both short and long texts and has great capacity to interpret the review's context since it combines embeddings with clustering to discover topics.

By experimenting with both models, the goal was to identify which one could suit better on this context and the one that could offer the best balance between a good topic coherence, diversity and always ensuring that the extracted topics aligned well with the overall goals of this recommendation system.

4.3.1 GSDMM

GSDMM is a probabilistic clustering model that is very suited for short text documents, such as reviews. It will group documents into clusters based on word frequency distributions with each cluster representing a topic. This way, GSDMM assumes that the documents inside the same cluster are very likely to discuss the same topic.

For the process to apply GSDMM algorithm, it starts by gathering all the needed data. This needed data are the pre-processed reviews.

The pre-processing of these reviews, as previously said, were the tokenization, removal of symbols, numbers and stop words, lower case all the tokens and lemmatization. The provided dataset to the model was already pre-processed with all the steps now mentioned. To train GSDMM there the TF-IDF vectorizer was used to form n-grams with a range from one to two.

To train the model, it was considered multiple hyperparameter combinations. The changed hyperparameters were:

- α: This parameter controls the distribution of topics inside a document. A high value might mean that a single review could touch several topics. A low value might indicate that the document is more likely to focus in only one topic.
- β: This parameter controls the distribution of words inside a topic. A high value might indicate that a topic can have a larger variety of words, making the topic more generic. A lower value might indicate that a topic will have more coherent words.
- *K*: This parameter indicates the target number of cluster that we want GSDMM to converge to.

The number of iterations was also a parameter that could be changed but, it was kept as one hundred iterations. For the parameter α it was tested three different values, for β two different values were tested and for *K* four values were tested.

The set of data that was used to train the model, was way more short than usual because each combination was taking very long time to train. In the end, each GSDMM model was trained with five thousand reviews and, even for this small number of reviews, each combination took around four hours to be tested. Since that twenty-four combinations were tested, the needed time to train all the combinations was ninety-six hours.

It was important to test different combinations so it could be possible to understand what options would fit best to the recommendation system.

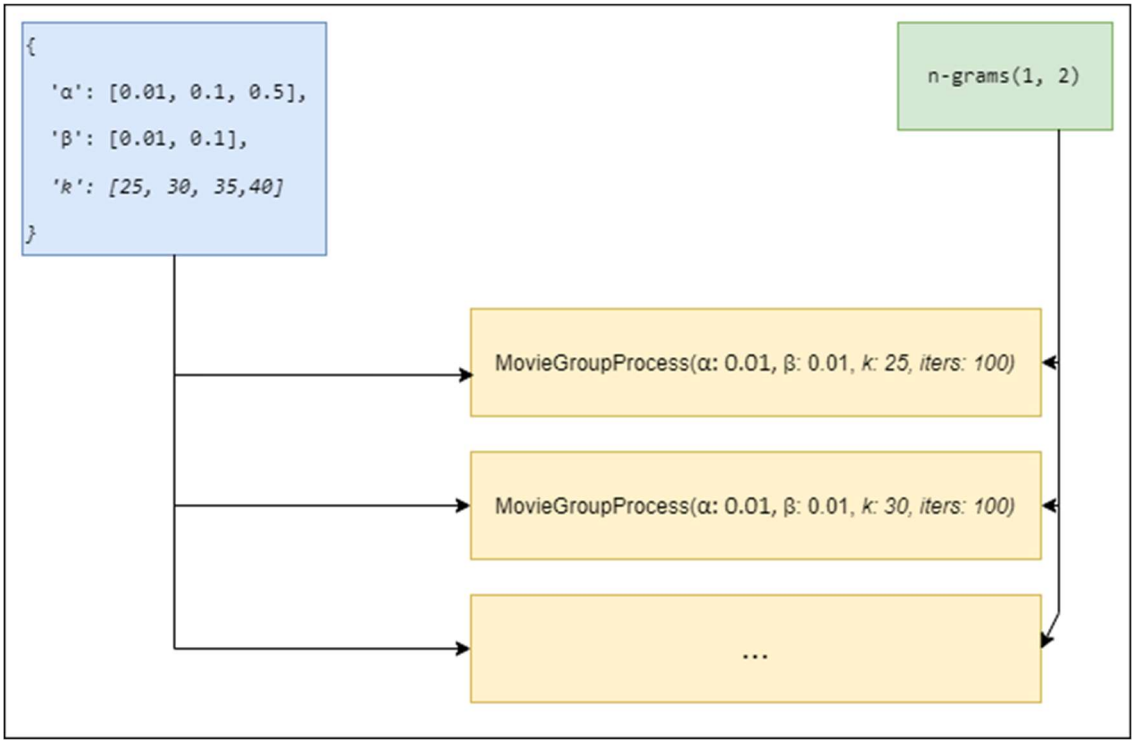

Figure 14 - GSDMM Model Tests

The values of each parameter on the image above, were the exact same values that were used to create different combinations. For each model, a different combination of hyperparameters was provided along with the needed n-grams for the model to apply clustering. As seen on the image, the value of the numbers of iterations, kept the same for each combination.

MovieGroupProcess is the model that GSDMM offers to generate topics based on the data we provide with the help of the different values of the hyperparameters.

After each combination finished training, the model was evaluated by calculating topic coherence. To calculate the topic coherence for each topic, it was gathered all the generated topics and it was extracted the top ten words of each. Those top ten words of each topic were ingested into the *CoherenceModel,* that is provided by the python library Gensim, and it calculated the topic coherence (*Gensim Website*, n.d.).

Although the combinations were evaluated by calculating topic coherence, the generated topics of each combination was manually analysed in order to understand how topics were being organized.

The figure 27 represents the flow that occurred at end of the training of every combination. The top words of the model's topics were extracted and ingested on *CoherenceScore* class that calculated the score.

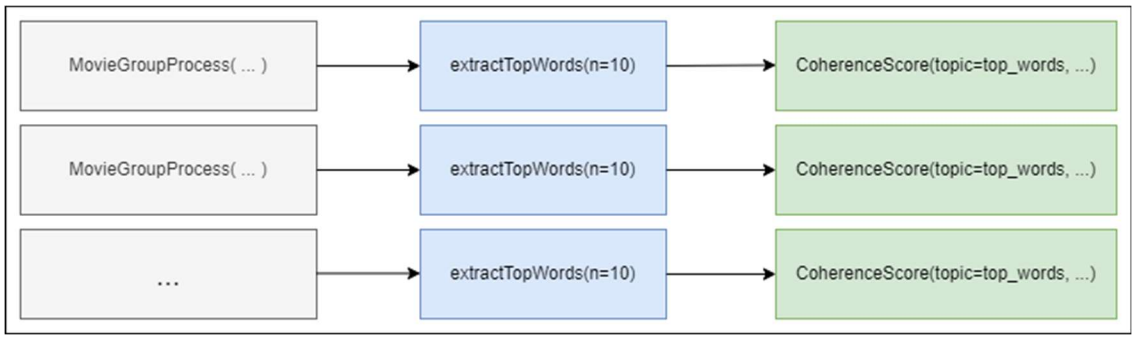

Figure 15 - Top Words Extraction for Topic Coherence Calculation

4.3.2 BERTopic

The BERTopic is a pre-trained model that is provided by a python library called *bertopic*. BERTopic needs specific embedding models. For this process, it was used a SentenceTransformer with the pre-trained model *paraphrase-mpnet-base-v2*. This pre-trained model is a model that will convert the provided reviews into high-dimensional representations that can capture semantic meaning of the text. Since it can capture semantic meaning from sentences, it fits very well for the purpose of forming clusters by calculating similarities between the numerical representations of the review.

To start applying the BERTopic, it was gathered all the reviews there were previously pre-processed. After getting all the reviews, the SentenceTransformer was initialized and used to transform those into numerical representations.

After getting the embeddings, the next step to train the model was to set up the UMAP and HDBSCAN models so they can be used by the BERTopic model. The UMAP is going to decrease our embeddings dimensionality. By reducing dimensionality, the model will be able to get a better performance because it will be working with lower dimensions and, although UMAP reduces the dimensionality, it preserves the original semantic structure. With those reduced embeddings, the cluster algorithm HDBSCAN will identify dense regions, which are regions where many points are close to each other, and will identify topics. HDBSCAN is also capable of identifying if a certain document is relevant for a topic, if not, it can identify the document as noise.

After setting up the UMAP and HDBSCAN models, the BERTopic model itself was initialized by passing the previous created models as parameters so they can be used by it. Similarly to the previous processes, different combinations were made in order to get the best result out of those.

To help on the creation of combinations, it was used the Optuna python library. Optuna is a hyperparameter optimization framework designed to automate the process of tuning ML models.

Some of these changed hyperparameters belonged to UMAP model, others to HDBSCAN model and other to the BERTopic model itself.

As seen in the figure 28, the values for each parameter are defined inside a range. The reason why those values are defined within a range is because Optuna will create a certain number of trials to train the model and, on each trial, a value inside of that range will be selected and used by the model during its training.

At the end of every trial, it will be performed a calculation of the topic coherence of that model and, in the end, all the trials, along with the respective selected parameters and results, will be stored in a Dataframe created with the help of the *pandas* library. This information will be used later to analyse the results a find a reasonable balance between topic coherence score and the number of topics.

The model that represents the best balance between coherence score and the number of topics, will be the selected model to be used on the final recommendation system.

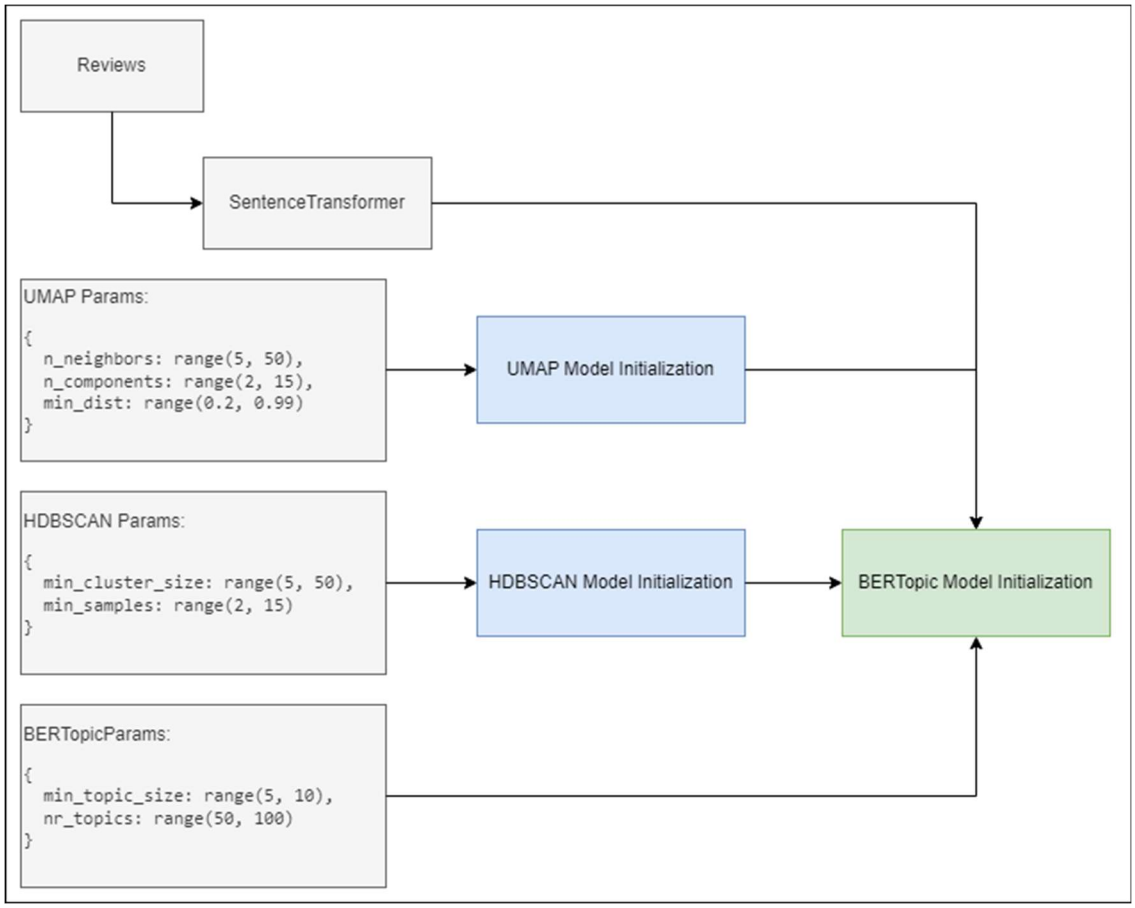

Figure 16 - Dependencies for BERTopic Initialization

4.4 User-Topic and Movie-Topic Matrices

The creation of the matrices is one of the most important parts of the recommendation system. On this part of the recommendation process, it will be gathered all the information provided by the sentiment analysis and the topic modeling.

With the provided information from our sentiment analysis and topic modeling models, it will be created two matrices:

- User-Topic matrix
- Movie-Topic matrix

These matrices represent the relationship between users, movies and topics that are reflected on the reviews. The user-topic matrix will capture the strength of a relation between a user and the identified topics, based on the reviews they made. The movie-topic matrix will reflect how strongly a movie is related to the topics. All this information is provided from the reviews that the users made.

The construction of these matrices is very important for the recommendation system, as they serve as the foundation for determining user preferences and movie characteristics. These relationships were further influenced by the sentiment analysis performed on each review, allowing for a more realistic representation of the user's interest and the movie's attributes.

4.4.1 Creation Process

As mentioned before, to form both user-topic and movie-topic matrices, it started by gathering information about the topic of the review and the sentiment that was associated with it. When gathering this information, the matrices were created with a very similar process.

After gathering the topic that the review was related with, it will be created the user-topic matrix where the rows represent the users and the columns represent the topics. For that user and for that specific topic, the value of that cell will be incremented by one.

As for the movie-topic matrix, the rows will represent the target movie and the columns represent the topic. Similarly to the user's matrix process, for that specific movie and topic, the value of that cell will be incremented by one.

By now, the user-topic and movie topic matrices are created and for each occurrence of a topic from a user or for a specific movie, the value will always be incremented by one. Although the matrices, by now, are created, they are not yet completed because it needs the influence of the sentiment to adjust the relevance of the topic for a user and for a movie. Since the sentiment analysis model provides a binary classification (positive or negative), it is not possible to provide a very dynamic weight of the sentiment to the relevance of the topic. This way, the followed strategy was to apply a factor of twenty-five percent considering the sentiment. In other words, if the sentiment of a review was positive, the relevance of the detected topic will increase twenty-five percent and if the review is negative, the relevance of that topic will decrease twenty-five percent.

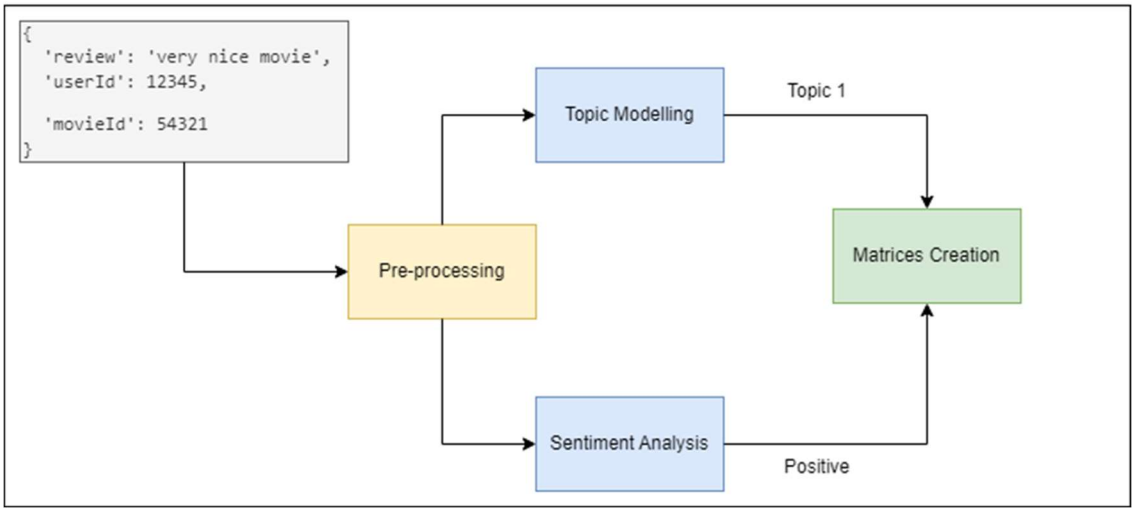

Figure 17 - Sentiment Analysis and Topic Modeling Integration

Using the example of the image above, once the matrices creation step receives the all the needed information:

- UserId
- MovieId
- Topic
- Sentiment

The matrices, without the sentiment influence, will look like this:

User-Topic Matrix	
User/Topic	Topic 1
12345	1

Table 13 - Created User-Topic Matrix

Movie-Topic Matrix	
Movie/Topic	Topic 1
54321	1

Table 14 - Created Movie-Topic Matrix

Considering that after the analysis of the sentiment of the review, it was classified as positive. Therefore, for both matrices, the value will increase twenty-five percent. The updated matrices, would look like this:

User-Topic Matrix	
User/Topic	Topic 1
12345	1.25

Table 15 - User-Topic Matrix with Positive Sentiment

Movie-Topic Matrix	
Movie/Topic	Topic 1
54321	1.25

Table 16 - Movie-Topic Matrix with Positive Sentiment

After the matrices being created, the final step was to normalize the values between zero and one. This normalization was necessary to standardize the influence of each topic across users and movies, ensuring that no user or movie influenced the model in a disproportionately way by having more reviews. The normalization helped on balancing the values so that every user and movie had comparable weights of their respective topic associations.

4.4.2 Update Matrices Process

The update process of user-topic and movie-topic matrices has very similar steps to the creation process of both matrices. The difference on this process is that a validation occurs to ensure if the user is present on the user-topic matrix, along with the topic and add those to the matrix. Similarly, a validation is performed in order to ensure that the movie and the topic are both present on the movie-topic matrix, if not present, both are added.

By using the previous matrices above, in a scenario where both user, topic and movie are new, both matrices will be updated with those values. So, the process of updating matrices, starts by gathering all the needed information such as:

- UserId
- MovieId
- Review

From the review, the model will extract sentiment and topic. Assuming that we already have the needed data to update the matrices with new data:

```
{
   'userId': 6789,
   'movieId': 9876,
   'topic': 'Topic 2',
   'sentiment': 0
}
```

Figure 18 - New Data to Update Matrices

Has the userId, movieId and topic are not present on any of the matrices, both will be added to user-topic and movie-topic matrices. For that value of relevance for the new topic for and for the new user on the user-topic matrix will initially be one but, since the review had a negative sentiment, the relevance will be downgraded in twenty-five percent.

User-Topic Matrix		
User/Topic	Topic 1	Topic 2
12345	1.25	0
6789	0	0.75

Table 17 - Updated User-Topic matrix

The same thing will happen for the movie-topic matrix. The value of association of the new movie and the new topic will be initially settled as one but, with the negative sentiment that was interpreted by the sentiment analysis model, the value goes down by twenty-five percent. So, the matrix would look like the following.

Movie-Topic Matrix		
Movie/Topic	Topic 1	Topic 2
54321	1.25	0
9876	0	0.75

Table 18 - Updated Movie-Topic matrix

After updating the original matrices, they will also be normalized once again in order to keep the values the most regularized as possible.

4.5 Latent Factors Extraction

Once the user-topic and movie-topic matrices are created and normalized, the next crucial step for the recommendation system is to extract the latent factors of both matrices. The extracted latent factors will represent hidden patterns or underlying relationships between users, movies and topics. The extraction of these latent factors allows the system to go beyond direct associations and uncover more complex relationships that are not immediately visible in the raw matrix.

By decomposing the user-topic and movie-topic matrices into latent factors, it is possible to map users and movies int a shared latent space where similarities and preferences can be calculated in an easier way. The latent factors will serve as foundation for predicting user

preferences for movies that they have not yet reviewed, enabling the recommendation engine to generate personalized suggestions.

For the latent factors extraction, two methods were applied in order to evaluate which one would fit best and have the best results. The two methods that were selected to this experiment were:

- NMF (Non-negative Factorization Matrix)
- SVD (Single Value Decomposition)

For this part of the process, the IMDB Dataset was not used. Instead, it was used the Rotten dataset since it contained data regarding the reviewers and the target movie along with the review itself. The user information and the movie information were crucial to the creation and usage of the matrices. No information about the user is explicit since the user identification is hashed before being inserted to the model.

To test the model in the end of the calculation of the preferences, it was needed some information that was not present of the dataset. To calculate RMSE and MAE, it was needed some ratings already defined on the training data, but it was not available. As a work around, for each review of the training and testing data, the sentiment was classified and if it was positive, the rating was settled as 0.75, if negative the rating was settled as 0.25.

4.5.1 NMF

NMF is a matrix factorization technique that can decompose a matrix into lower dimensional matrices while ensuring that all elements remain non-negative, which is ideal for capturing positive associations between users, movies and topics.

The latent factors extraction process starts by decomposing the user-topic and movie-topic matrices. Using user-topic matrix as an example, the matrix with dimensions *m x n* (where *m* is the number of users and *n* is the number of topics) will be decomposed in two matrices. Those matrices will be:

- User latent matrix (*m* x *k*): represents the latent factors for the users where *m* are the users and *k* are the latent factors
- Topic latent matrix (*k* x *n*): represents the latent factors of the topics where *k* are the latent factors and *n* are the topics

For the movie-topic matrix with dimensions *p* x *n* (where *p* is the number of movies and *n* is the number of topics), the same matrices will be created:

- Movie latent matrix (*p x k)*: represents the latent factors for movies where *p* are the movies and *k* are the latent factors
- Topic latent matrix (*k x n)*: represents the latent factors for the topics similarly to the one that was generated by decomposing user-topic matrix.

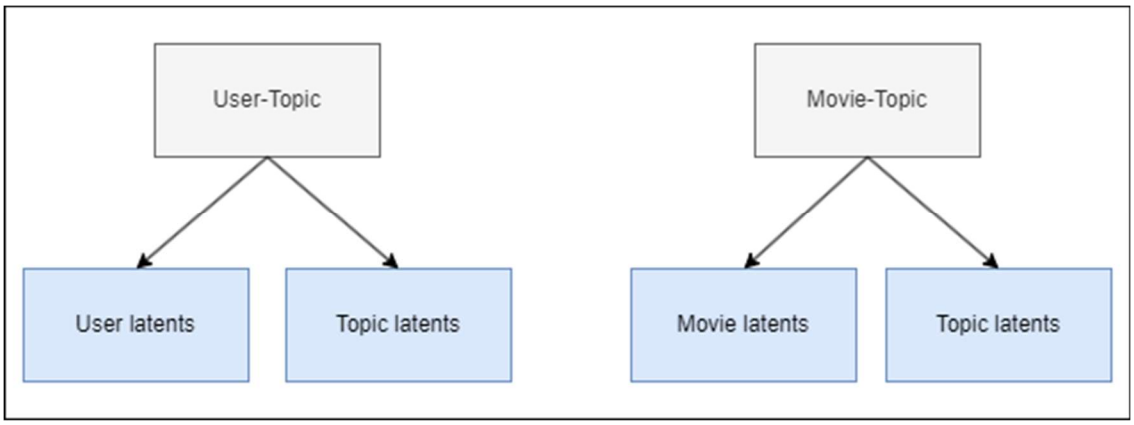

Figure 19 - Latent Factors Extraction NMF

Now that the latent factors are extracted from both user-topic and movie-topic matrices, the users and movies now need to be mapped into a common latent space. This will allow the model to calculate user preferences for movies.

To calculate the preferences, it will be used the user-latent matrix that was generated through the decomposition of user-topic matrix and will also be used the movie-latent matrix that was generated through the decomposition of the movie-topic matrix.

After gathering those two matrices, it will be calculated how much a user would like a movie, even if the user has not reviewed it yet, by calculating the similarity between the user-latent and movie-latent matrices. The predicted score will be calculated as:

$$Preferences(Ul, Ml) = Ul * Ml^T$$

On this equation:

- *Ul* represents the matrix user-latent
- *Ml* represents the matrix movie-latent

The result of this dot product will represent the predicted strength of the user's preference for a movie based on the hidden factors shared between the user and the movie.

After calculating the preference scores for each user-movie pair, the scores were normalized to ensure they were placed between zero and one, similar to the normalization process that was performed to the original user-topic and movie-topic matrices.

The original matrices were generated with only eighty percent of the data and the other twenty percent were saved for testing purposes. After the normalization of the preferences, the model was exposed to the unseen twenty percent of the data to perform test and validations.

4.5.2 SVD

As mentioned before, SVD was one of the two techniques that were used on this experiment. SVD is a technique that, similarly to NMF, can decompose matrices. But, differently from NMF, SVD decomposes a matrix in three different matrices which are the left singular matrix, diagonal matrix of singular values, right singular matrix.

Staring by the user-topic matrix (*m* x *n*), when applying SVD, the generated matrices will be the following:

- Left Singular Matrix: will contain the latent factors for the users
- Diagonal Matrix of Singular Values: will contain the importance of each latent factor
- Right Singular Matrix: will contain the latent factors of the topics

Similarly, for the movie-topic matrix:

- Left Singular Matrix: will contain the latent factors for the movies
- Diagonal Matrix of Singular Values: will contain the importance of each latent factor
- Right Singular Matrix: will contain the latent factors of the topics

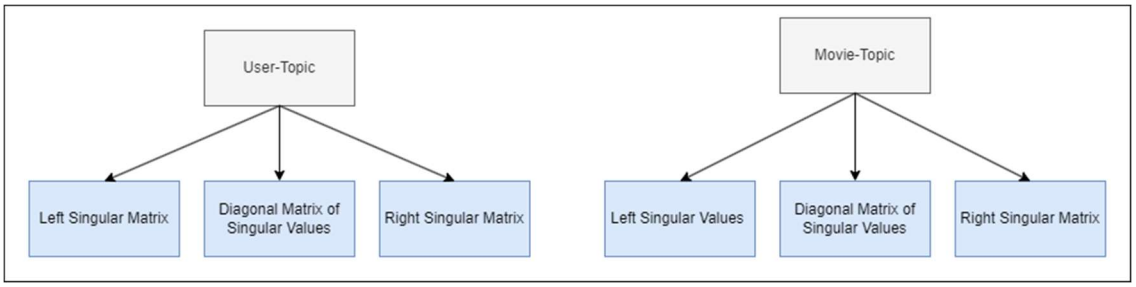

Figure 20 - Latent Factors Extraction SVD

Similarly to the process followed for the NMF, it was used the users and movies that shared the same latent space provided by SVD.

On SVD, each user is represented as a vector in the latent space by multiplying their latent factors from the left singular matrix from the values of the diagonal matrix. The same applies for the movies, the movies latent space is represented by applying the left singular values by the diagonal values.

$$Ul(Ulm, Udm) = Ulm * Udm \qquad Ml(Mlm, Mdm) = Mlm * Mdm$$

Where:

- *Ulm* represents the user's left matrix
- *Udm* represents the user's diagonal matrix
- *Mlm* represents the movie's left matrix
- *Mdm* represents the movie's diagonal matrix

Once the latent vectors are calculated for both users and movies, it is possible to calculate preferences using the following formula:

$$Preferences(Ul, Ml) = Ul * Ml^T$$

Where:

- *Ul* is the user-latent matrix
- *Ml* is the movie-latent matrix

After calculating the preferences, the generated matrix was also normalized between zero and one to ensure consistency and interpretability.

5 Discussion of Results

In this chapter, it will be presented used metrics and the results that were obtained from the implementation and evaluation of the components of the suggested recommendation system:

- Sentiment Analysis
- Topic Modeling
- User Preference Predictions

Each of these components played a very important role on the construction of the system that aims to capture the interest of the users and to generate personalized movie recommendations.

The chapter begins by discussing the performance of the multiple sentiment analysis models that were applied to the reviews. This includes a comparison between Logistic Regression, Naïve Bayes and BERT analysing their effectiveness in capturing the emotional tone of the reviews.

The next focus will be on the topic modeling techniques that were used to extract meaningful topics from the reviews. Here, it was compared two different models such as GSDMM and BERTopic where the topic coherence was evaluated and manual validations to the generated topics were also performed.

Finally, it will be discussed the results of the latent factors extraction using matrix factorization techniques. These latent factors are very important to predict user preferences and, in this section, it will be demonstrated how the system is able to calculate the use preferences based on observed hidden patterns on the data.

This chapter aims to provide an analysis of the components of the suggested recommendation system, offering insights of their performance, effectiveness and contribution to the goal of generating movie recommendations.

5.1 Metrics

On this section, it will be described the metrics that were used on the different components of this recommendation system. It will start by describing metrics used on sentiment analysis such as confusion matrix, accuracy, precision, recall, f1-score and ROC-AUC. For the topic modeling task, it was used the validation metric topic coherence along with manual validations. And, finally, to evaluate the preference scores after the latent factor's extraction, the used metrics were RMSE and MAE.

5.1.1 Accuracy

Accuracy represents the percentage of correct predictions that the model performs. The formula to calculate accuracy is the following:

$$acc = \frac{tp + tn}{tp + tn + fp + fn}$$

Where:

- *Tp* represents the true positives
- *Tn* represents the true negatives
- *Fp* represents the false positives
- *Fn* represents the false negatives

(Ray & Susan, 2022)

5.1.2 Precision & Recall

Precision represents the ratio of true positives considering the total number of positives detected by the model. It is defined by the following formula:

$$prec = \frac{tp}{tp + fp}$$

The recall represents the true positive rate and is defined by:

$$recall = \frac{tp}{tp + fn}$$

Where:

- *Tp* represents the true positives
- *Fp* represents the false positives
- *Fn* represents the false negatives

(Ray & Susan, 2022)

5.1.3 F1-Score

F1-Score is the balance score between the metrics precision and recall. A high value for the F1-Score indicates good accuracy. This metric is defined by:

$$F1 - Score = \frac{2 * precision * recall}{precision + recall}$$

(Ray & Susan, 2022)

5.1.4 ROC-AUC

The area under curve is calculated through the ROC which is placed between the true positive rate and the false positive rate. An area under curve that is close to a value 0.5 is considered as a bad classifier. The higher the value the more accurate are the classifications (Ray & Susan, 2022).

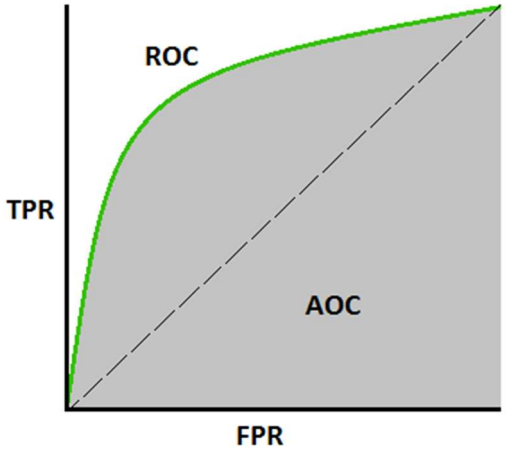

Figure 21 - Example of ROC-AUC

5.1.5 Confusion Matrix

A confusion matrix is a table that is used to evaluate the performance of a classification model. The rows represent the actual truth labels of the samples and the columns represent the predicted labels generated by the model.

- True Positives (TP): Number of instances that were correctly predicted as positive
- True Negatives (TN): Number of instances that were correctly predicted as negative
- False Positives (FP): Instances predicted as positive but are negative
- False Negatives (FN): Instances predicted as negative but are positive

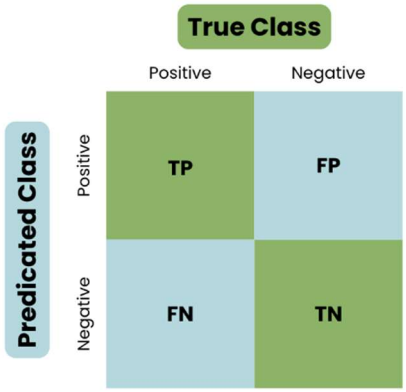

Figure 22 - Confusion Matrix

5.1.6 Topic Coherence

Topic coherence is a metric used to evaluate the quality of topic models, particularly in how well the words within a topic cluster are semantically close. It measures how words inside a topic can make sense. If the top words for a topic a frequently seen together in actual contexts, the topic might be considered coherent.

The formula of topic coherence is based on the comparison of words inside a topic and looking on how frequently the co-occur across documents. A higher value for topic coherence indicates more meaningful topics (Qiang et al., 2022).

5.1.7 RMSE

RMSE, which stands for Root Mean Square Error, represents the square root of MSE. It is capable to measure the average magnitude of the errors between the predicted values and the actual ones. The smaller the RMSE the higher the recommendation quality will be. It is defined by:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(yi - \hat{y}i)^2}$$

Where:

- N is the number of observations
- Yi is the actual value of the observation
- $\hat{y}$I is the predicted value for the observation

(Xu, 2019)

5.1.8 MAE

The metric MAE represents the average of the absolute difference between the actual value and the predicted value. The formula of this metric is the following:

$$MAE = \frac{1}{n}\sum_{i=1,}^{n} |yj - \widehat{y\imath}|$$

Where:

- N is the number of observations
- Yi is the actual value of the observation
- $\hat{y}$I is the predicted value for the observation

(Liu & Zhao, 2023)

5.2 Results

In this sub-section, it will be presented the results of the models that were tested for the recommendation system components (sentiment analysis, topic modeling and latent factors extraction techniques).

5.2.1 Sentiment Analysis

For the sentiment analysis, there were tested three different models such as Logistic Regression, Naïve Bayes and BERT. Both Logistic Regression and Naïve Bayes, were tested with different vectorizers like TF-IDF vectorizer and Count Vectorizer and also with and without n-grams.

For the sentiment analysis task, the metrics that were used to perform the evaluation of the models were:

- Accuracy
- Precision
- Recall
- F1-Score
- ROC-AUC
- Confusion Matrix

Model	Accuracy	Precision	Recall	F1-Score
LR with Count Vectorizer	0.8812	0.89	0.89	0.89
LR with TF-IDF	0.8919	0.89	0.89	0.89
LR with Count Vectorizer (1 – 3)	0.8934	0.90	0.90	0.90
LR with TF-IDF (1 – 3)	0.8982	0.90	0.90	0.90
LR with Word2Vec	0.62	0.62	0.62	0.62
NB with Count Vectorizer	0.8565	0.86	0.86	0.86
NB with TF-IDF	0.8621	0.86	0.86	0.86
NB with Count Vectorizer (1 – 3)	0.8852	0.89	0.89	0.89
NB with TF-IDF (1 – 3)	0.8936	0.89	0.89	0.89
BERT	0.8586	0.8664	0.8591	0.8580

Table 19 - Sentiment Analysis Models Comparison

As can be seen on the table, Logistic Regression has the best performance. Although being a model that is very simple to apply, it showed the best results also due to the nature of dataset which offers a binary classification. The result of Logistic Regression without n-grams was already a very good result but, with the integration of n-grams, the model was able to better understand context and capture word dependencies.

It is also possible to validate that both Naïve Bayes had better results with the usage of n-grams during the training phase. The usage of n-grams allows the model to better understand the context.

BERT also showed a great performance but not better than Logistic Regression. The reason ot didn't got better performance was due to limited resources to perform model tuning to improve performance.

The figures 35 and 36 represent the ROC-AUC for both Logistic Regression and Naïve Bayes with TF-IDF with an n-gram range between one and three. As can be seen the ROC curves for both models are very distant from the diagonal and very close to the upper-left corner and with an AUC of 0.97 and 0.96 which means that Logistic Regression and Naïve Bayes are 97% and 96%, respectively, likely to correctly distinguish between positive and negative.

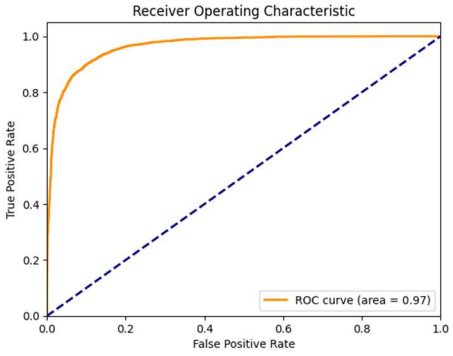

Figure 24 - ROC-AUC LR with TF-IDF

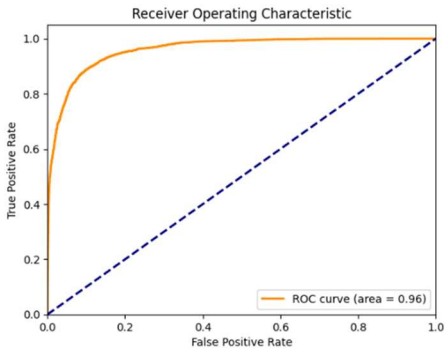

Figure 23 - ROC-AUC NB TF-IDF

The confusion matrices also show good performance on both models, the number of correct classifications is very balanced between both models.

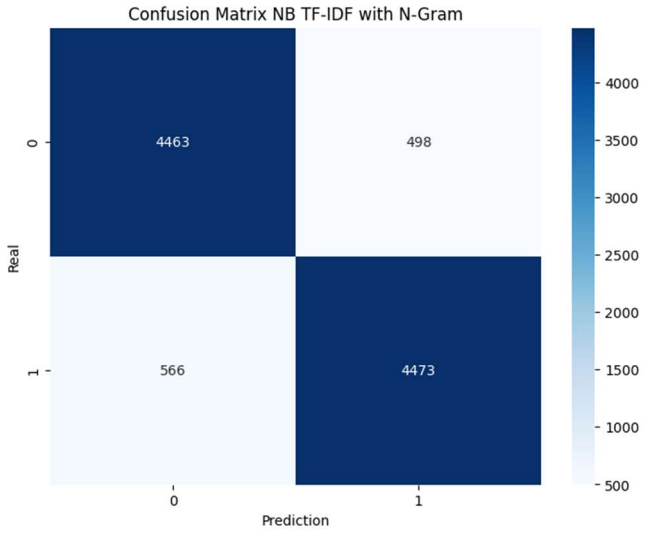

Figure 25 - Confusion Matrix NB with TF-IDF

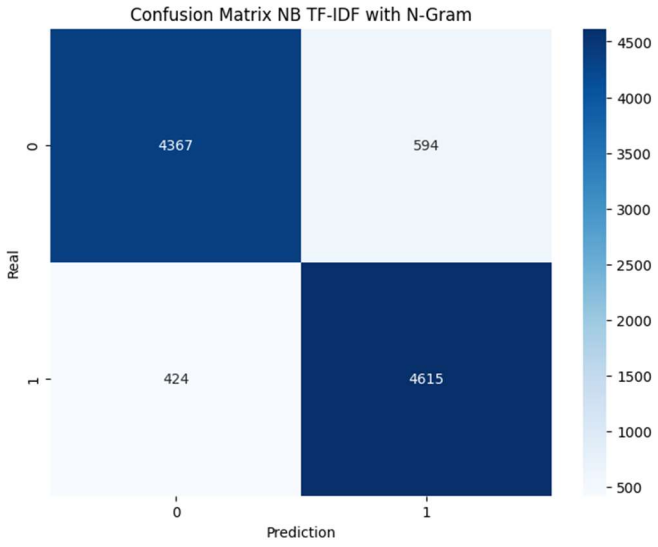

Figure 26 - Confusion Matrix LR with TF-IDF

The Logistic Regression and Naïve Bayes models with TF-IDF and n-grams were the models with best performances and, between those tow models, the model that showed best performance was Logistic regression with an accuracy of 90%.

5.2.2 Topic Modeling

On the topic modeling task, the tested models were GSDMM and BERTopic due to they're ability to work with short text. On GSDMM, TF-IDF was used form n-grams on a range of one to three so the model has the capacity to obtain more context and generate more significant topics. When testing BERTopic, n-grams were also used to help BERTopic form context along with the sentence transformer that creates embeddings for the model.

The metrics that were used to evaluate the models of the topic modeling task were:

- Topic Coherence
- Topic Manual Validations

Starting with GSDMM, multiple combinations of hyper-parameters were performed in order to get the model with the best performance. As mentioned in the previous chapter, the parameters that were changed on the different combinations were α, β and *K*.

As seen on the figure 39, the topic coherence for the different combinations (α, β, *K*) was very concentrated in values between 0.36 and 0.39 with the last combination having a better performance and reaching a topic coherence above of 0.44. All the combinations were trained with the same number of iterations which was 100.

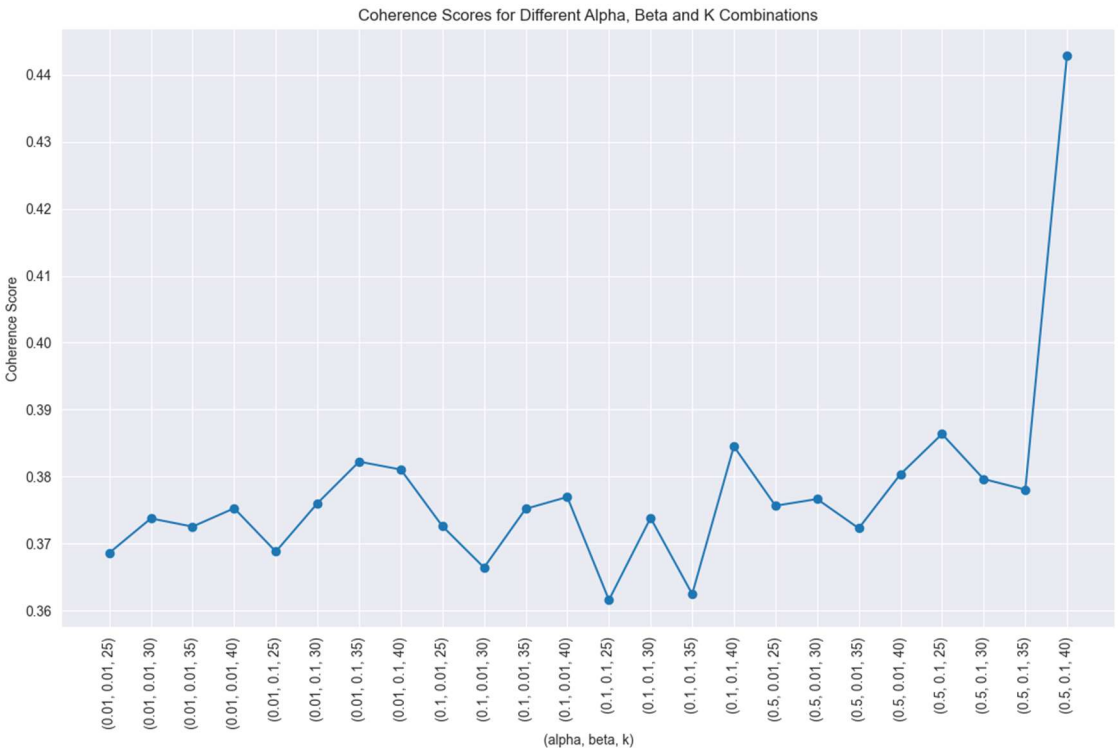

Figure 27 - GSDMM Results

In some of the models, although the target number of topics is being defined, the model converged into a smaller number of topics. The reason behind this could be that a smaller sample of reviews were used to train the model due to the time of training. An amount of 5000 reviews were used to train the model because it was taking a long time to be trained. For the number of 100 iterations, each combination took between four to five hours to be trained with a sample of 5000 reviews. The approximate time of training for all the combinations was 120 hours.

After calculating the topic coherence, the three best combinations were:

- α: 0.01, β: 0.1, K: 35
- α: 0.1, β: 0.1, K: 40
- α: 0.5, β: 0.1, K: 40

When analysing the generated topics for these best models, the ten top words of each was extracted. Although some topics have clear context, the majority of them didn't have a meaningful context and there were lots of topics that were very similar to each other.

The tables 20, 21 and 22 contains samples of ten topics of each combination that had the best topic coherence. Although some of the topics are clear on their context, it is possible to see that the majority of them are not very meaningful and very clear on their context. Also, some of the topics seems to be focusing on the same things which might mean that the model is generating topics that are a bit generic.

Table 20 - Sample of Topics for Combination α: 0.01, β: 0.1, K: 35

	α: 0.01, β: 0.1, K: 35
Topic 1	read, book, write, review, comment, say, film, movie, novel, imdb
Topic 2	like, look, black, white, movie, man, guy, monster, girl, house
Topic 3	minute, year, two, first, film, time, hour, one, last, scene
Topic 4	movie, film, plot, act, bad, scene, story, good, character, make
Topic 5	actor, great, film, cast, performance, best, role, one, play, good
Topic 6	movie, see, film, watch, time, one, ever, first, make, like
Topic 7	film, year, show, series, tv, movie, night, release, one, come
Topic 8	war, state, world, go, police, use, agent, secret, government, american
Topic 9	film, year, show, series, tv, movie, night, release, one, come
Topic 10	billy, scott, george, play, bob, danny, character, jeff, custer, gloria

α: 0.1, β: 0.1, K: 40	
Topic 1	film, movie, one, make, good, great, best, like, bad, well
Topic 2	like, around, get, one, car, scene, look, go, man, head
Topic 3	character, film, story, movie, plot, end, make, one, scene, way
Topic 4	war, khan, texas, brazil, massacre, leader, chainsaw, vote, gang, lincoln
Topic 5	time, film, year, movie, first, watch, minute, one, see, last
Topic 6	police, state, agent, officer, secret, award, get, force, take, go
Topic 7	de, la, le, niro, vega, film, brian, like, murphy, du
Topic 8	john, play, director, film, michael, james, like, jack, robert, tom
Topic 9	film, make, director, movie, work, use, art, lack, attempt, sense
Topic 10	mr, play, dr, david, like, robert, bill, frank, wood, movie

Table 21 - Sample of Topics for Combination α: 0.1, β: 0.1, K: 40

α: 0.5, β: 0.1, K: 40	
Topic 1	war, world, life, film, people, american, take, come, live, make
Topic 2	von, max, sydow, film, become, lola, villain, maria, self, trier
Topic 3	like, get, around, one, look, scene, go, guy, head, man
Topic 4	film, movie, one, horror, great, story, comedy, make, like, best
Topic 5	music, film, work, use, camera, score, great, shot, song, cinematography
Topic 6	film, movie, make, like, much, good, would, give, time, people
Topic 7	get, go, know, movie, dont, one, like, say, really, think
Topic 8	movie, film, bad, good, act, plot, story, one, scene, make
Topic 9	new, get, take, go, away, york, place, find, try, film
Topic 10	actor, role, play, performance, cast, character, great, good, film, lead

Table 22 - Sample of Topics for Combination α: 0.5, β: 0.1, K: 40

For BERTopic, it was possible to feed all the reviews and the training performance was way better than GSDMM. As mentioned on the previous chapter, it was also created different combinations to train BERTopic, the number of combinations was 200 and each combination took a little bit longer than 1 minute so, to train 200 combinations of BERTopic, it took around 3 hours.

To get the best model, it was evaluated the models with the best balance between the number of topics and the topic coherence. A good topic coherence with a very few numbers of topics might indicate that the topics are being generic, but if the number of topics is too high, it might mean that the topics are very specific. The target number of topics was higher than 20 with a topic coherence higher than 0.40, so the selected trials were:

Trial	Topic Coherence	# Topics
18	0.42	28
26	0.44	23
50	0.45	33
75	0.48	24
89	0.53	41
96	0.56	26
136	0.56	36

Table 23 - BERTopic Best Trials

After evaluating the selected trials, the most balanced trial was the trial 89 with a topic coherence of 0.53 and 41 topics. By analysing the generated topics, topics it was possible to validate that they were very meaningful and had a clear context. Is also important to mention that the ability of BERTopic to generate such coherent topics, comes from the ability that it has to identify noise.

Table 24 - Sample Topics from Selected BERTopic Trial

BERTopic Trial 89 Results	
1_tarzan_jungle_ape_firefighter	tarzan, jungle, ape, firefighter, film, elephant
2_batman_superman_burton_joker	batman, superman, burton, joker, catwoman, gotham
7_christian_jesus_movie_bible	christian, jesus, movie, bible, mormon, religion
11_alien_scifi_movie_planet	alien, scifi, movie, planet, film, robot
13_fight_martial_movie_kung	fight, martial, movie, kung, japanese, chinese
14_western_wagon_indian_film	western, wagon, indian, film, mann, buffalo
18_war_soldier_hitler_movie	war, soldier, hitler, movie, battle, story
20_dinosaur_snake_crocodile_carnosaur	dinosaur, snake, crocodile, carnosaur, movie, train
28_murder_crime_detective_play	murder, crime, detective, play, movie, kill
33_zombie_movie_dead_bad	zombie, movie, dead, bad, zombi, undead

Considering the results of both models, BERTopic showed best topic coherence scores than GSDMM, where coherence scores didn't go beyond 0.44 on GSDMM on BERTopic coherence scores reached a maximum of 0.65, also BERTopic has produced more meaningful topics. Besides the results, the training time of each model was very different since for each BERTopic's trial, the training time was around 1 minute and for each GSDMM's trial, the training time was between 4 to 5 hours. Giving this, by comparing both models, BERTopic outperformed GSDMM by giving a great balance between training time and results.

5.2.3 User Preferences Calculation

Here, it is intended to evaluate how close the predicted ratings are from the original users ratings considering the latent factors extracted by NMF and SVD. To calculate user preferences, it was extracted the users and movies latent factors and was performed a scalar product between those. The techniques that were used to extract the latent factors of the user-topic and movie-topic matrices were NMF and SVD and both were evaluated with the metrics:

- RMSE
- MAE

Starting by NMF, to extract the latent factors it was considered multiple values for *K*. *K* is the number of latent factors to be extracted from the matrices. The considered values went from a range of 1 to 100 with an interval of 5 and the RMSE results of NMF for each value of *K* were the following:

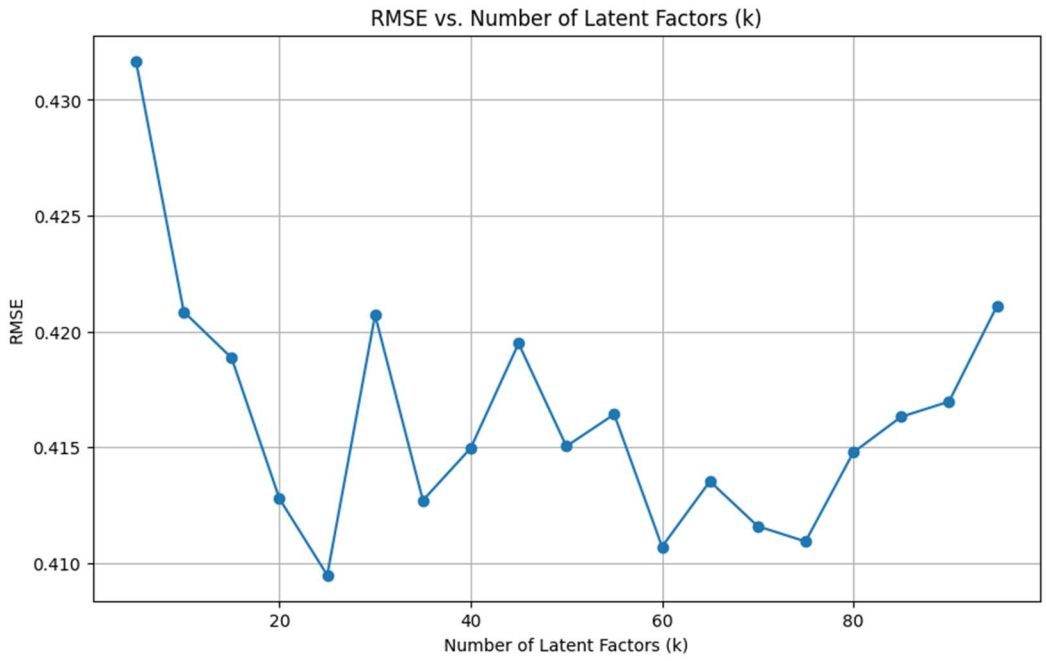

Figure 28 - RMSE Values for NMF

And the MAE results for each value of *K* for NMF were:

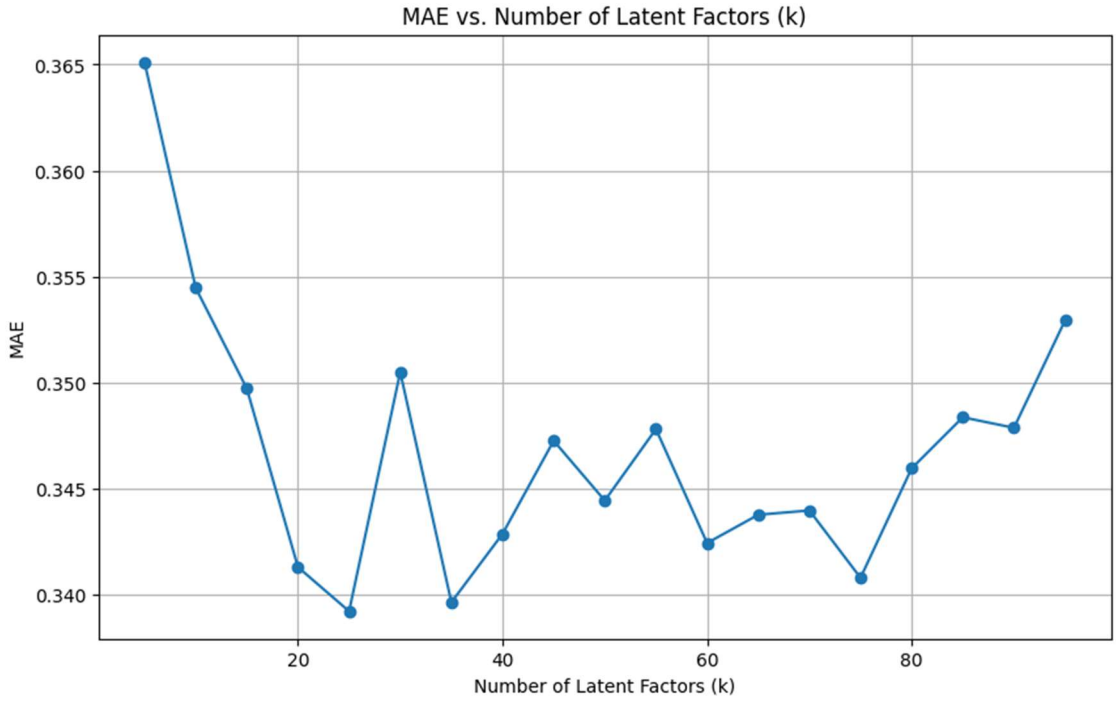

Figure 29 - MAE Values for NMF

The results above show that the best number of latent factors was 25 giving a RMSE of 0.409 and a MAE of 0.339.

Considering that all the ratings are placed between 0 and 1, the meaning of a RMSE value of 0.409, is that on average the predictions can deviate 0.409, this shows that the model still has points of improvement but it is still quite accurate on its recommendations. The meaning of a MAE value of 0.339 is that predictions are off by 0.339 from the actual ratings. Since the RMSE is slightly higher than MAE, it indicates that the model can perform larger errors but those errors are not critical enough to affect the precision of the recommendations.

As for SVD, the values of *K* that were tested were the same as NMF. The results of RMSE for SVD were the following:

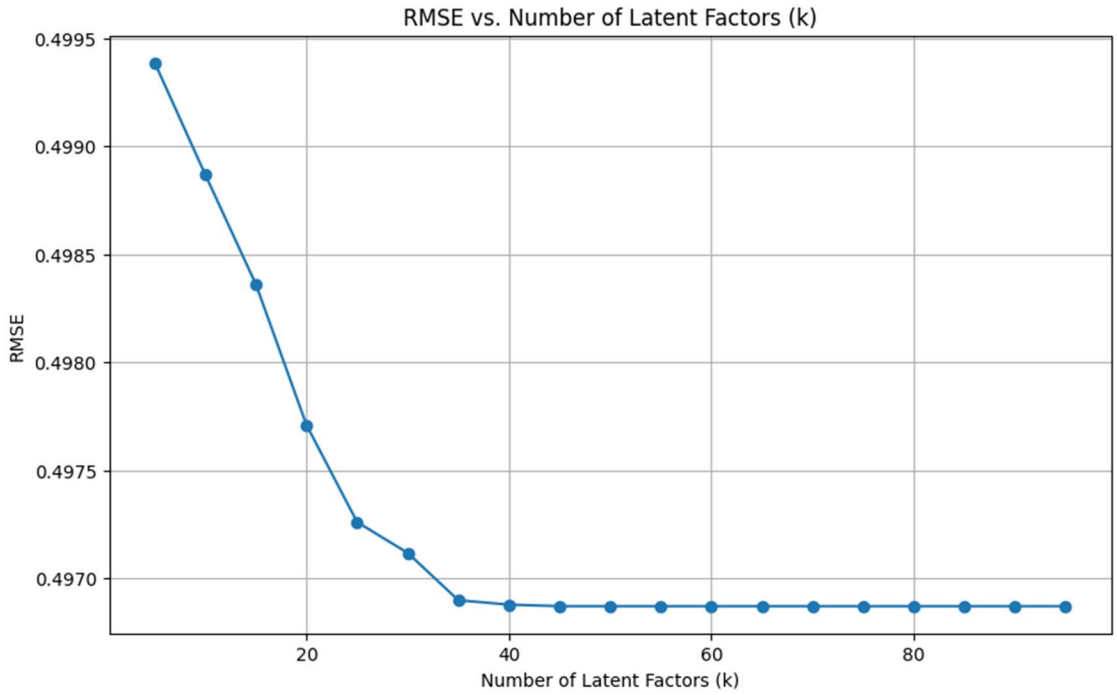

Figure 30 - RMSE Values for SVD

And the MAE results for each K values of SVD were:

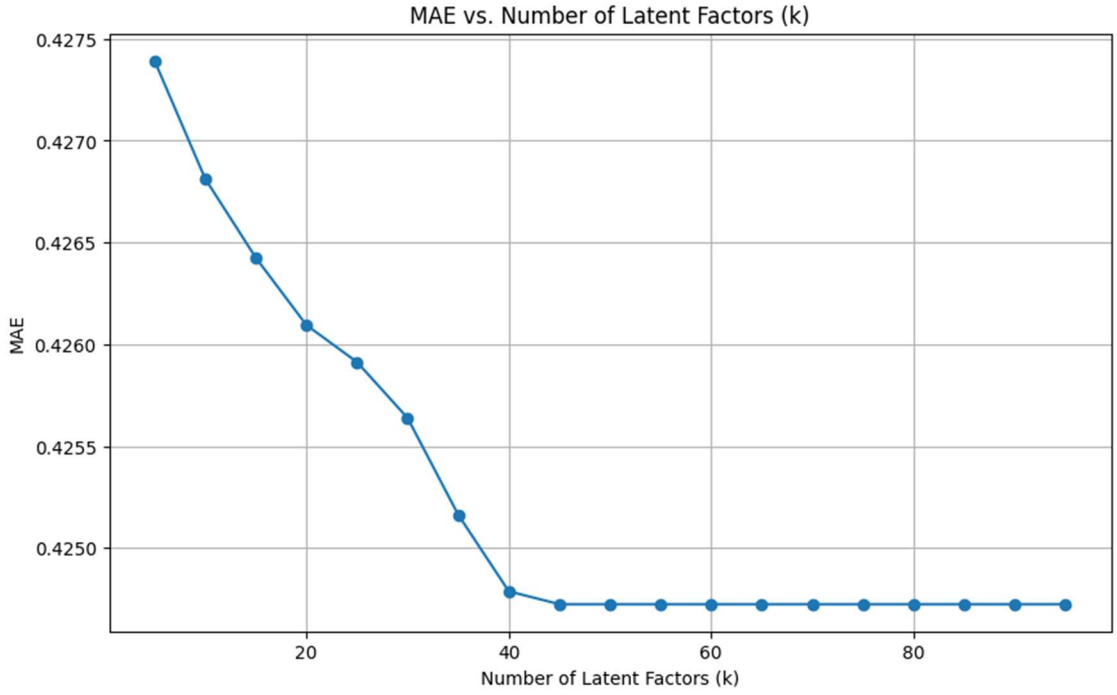

Figure 31 - MAE Values for SVD

By analysing the results of RMSE and MAE, it is possible to see that from 40 latent factors forward the results doesn't seem to change much, therefore, for the best result of both RMSE and MAE, the number of latent factors should be 40 with an RMSE of 0.496 and a MAE of 0.424.

By comparing the results of both NMF and SVD best models:

Technique	K Latent Factors	RMSE	MAE
NMF	25	0.409	0.339
SVD	40	0.496	0.424

Table 25 - Comparison Between NMF and SVD Results

It is possible to see that NMF has the best results with a minor error of prediction comparing to SVD. NMF, comparing to SVD, can perform preference predictions in 10% more closer to the actual preference scores.

6 Conclusions

This chapter aims to deliver a summary of the developed model and to present the conclusions that it was able to provide such as overall performance, findings and limitations. Along with it, it will be presented points of improvement as future work.

6.1 Summary and Conclusions

The primary goal of this project was to develop a recommendation system that could effectively capture user preferences and movie characteristics using sentiment analysis, topic modeling and latent factors extraction and provide personalized movie recommendations by predicting user preferences based on the content of their reviews.

For this recommendation system, different models were tested on the different identified tasks. For the sentiment analysis, the tested models were Logistic Regression, Naïve Bayes and BERT where Logistic Regression showed the best result accuracy result of 90% with the usage of TF-IDF vectorizer and the usage on n-grams. For the topic modeling task, the tested models were GSDMM and BERTopic. For both models, multiple combinations were tested and the model that performed the best was BERTopic where the selected model had a coherence score of 0.53 and 41 available topics while GSDMM coherence scores didn't go beyond 0.44. For the user preferences calculation, it was needed the extraction of latent factors from the user-topic and movie-topic matrices. The tested techniques for the latent factors extraction were NMF and SVD where NMF had the best result giving a RMSE of 0.409 and a MAE of 0.339 meaning that the predicted ratings can be, on average, 0.339 units away from the true ratings.

One of the key findings of this research was that combining sentiment analysis with topic modeling can indeed improve the creation of user and movie profiles. Comparing to different projects such as SAMF (Liu & Zhao, 2023), where it was used BERT for sentiment analysis and LDA for topic modeling, the MAE result was quite similar since the best MAE for SAMF was 0.29. By incorporating sentiment analysis, the model is able to better capture user preferences. Another project such STMF that used a lexicon method to calculate user sentiment and tested multiple matrix factorization like TopicMF, SVD++, showed very promising results where, in a scale of 1 to 5 for the ratings, the lowest value for MSE was 0.157 which represents a great potential of the usage of sentiment analysis, topic modeling and matrix factorization in a recommendation system (Nagoya Kōgyō Daigaku & Institute of Electrical and Electronics Engineers, n.d.). The recommendation systems can actually be enhanced by the using not only user ratings but also sentiment extracted from user reviews. This project aims to contribute to the field of content-based recommendation systems by offering a model that could also be applied to other domains such as books, music or products.

Some identified limitations were physical limitations. Some models needed more resources to be trained and took way longer than expected to be trained. Due to this limitation, it was difficult to perform some hyper tuning the model's parameters, more specifically to BERT. Also, it was needed to apply a limit of reviews for GSDMM to train due to the big amount of time that it was needing for training.

Another limitation was the dataset used to train the sentiment analysis model. Since it was a binary classification dataset (positive or negative), it was not possible to distinguish intensities of sentiment inside of each class. In other words, it would be better for the model to classify multiple sentiment intensities inside each class. The ability to distinguish intensities of sentiment, would allow a more dynamic calculation of topics relevance inside the user-topic and movie-topic matrices leading to more accurate prediction of user preferences.

6.2 Future Work

As future work, some points of improvement that could be applied on the developed model could be the development of a hybrid filtering algorithm joining the best of content-based filtering and collaborative filtering. The goal of this improvement would be the addition of a collaborative filtering that could calculate preferences similarities between users and even movies. With this improvement, the recommendation system would be able to recommend to a user, movies that were recommended to other users with similar preferences and also recommend movies that present similar characteristics to movies that are part of the user preferences.

As previously mentioned, there were some limitations regarding physical resources for BERT training on sentiment analysis task. As a future work, it would be interesting to explore different combinations of parameters for BERT and also try another pre-trained models for the task.

The exploration of more deep learning techniques would also be a point to improve in order to increase the model's ability to classify sentiment and better understand user's preferences by providing a sentiment classification in a continuous scale instead of a binary classification such as positive or negative.

Another point of improvement is to extend the model filters to calculate the preferences, this means including interaction history, not only reviews but also searching history, time-based filtering to understand if the user prefers old movies or recent movies, also perform an analysis to the movie's synopsis to understand in more detail what is the movie about.

References

2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS). (2020). IEEE.

Andrea Villa. (n.d.). *Rotten Tomatoes Dataset*. Https://Www.Kaggle.Com/Datasets/Andrezaza/Clapper-Massive-Rotten-Tomatoes-Movies-and-Reviews. Retrieved September 29, 2024, from https://www.kaggle.com/datasets/andrezaza/clapper-massive-rotten-tomatoes-movies-and-reviews

Bagul, D. V., & Barve, S. (2021). A novel content-based recommendation approach based on LDA topic modeling for literature recommendation. *Proceedings of the 6th International Conference on Inventive Computation Technologies, ICICT 2021*, 954–961. https://doi.org/10.1109/ICICT50816.2021.9358561

Gensim Website. (n.d.). Https://Radimrehurek.Com/Gensim/. Retrieved September 26, 2024, from https://radimrehurek.com/gensim/

Hoblos, J. (2020, December 14). Experimenting with latent semantic analysis and latent dirichlet allocation on automated essay grading. *2020 7th International Conference on Social Network Analysis, Management and Security, SNAMS 2020*. https://doi.org/10.1109/SNAMS52053.2020.9336533

Hoda, M. N., Bharati Vidyapeeth's Institute of Computers Applications and Management Delhi, & Institute of Electrical and Electronics Engineers Delhi Section. (n.d.). *Proceedings of the 17th INDIACom; 2023 10th International Conference on Computing for Sustainable Global Development (15th-17th March, 2023) INDIACom-2023*.

Hofmann, T. (2001). *Unsupervised Learning by Probabilistic Latent Semantic Analysis* (Vol. 42).

IEEE Electron Devices Society, Institute of Electrical and Electronics Engineers, & Vaigai College of Engineering. (n.d.). *Proceeding of the 2018 International Conference on Intelligent Computing and Control Systems (ICICCS) : June 14-15, 2018*.

Institute of Electrical and Electronics Engineers, & Institute of Electrical and Electronics Engineers. Delhi Section. (n.d.). *Proceedings of the 2022 9th International Conference on Computing for Sustainable Global Development (INDIACom) : 23rd- 25th March, 2022, New Delhi, India.*

Isinkaye, F. O., Folajimi, Y. O., & Ojokoh, B. A. (2015). Recommendation systems: Principles, methods and evaluation. In *Egyptian Informatics Journal* (Vol. 16, Issue 3, pp. 261–273). Elsevier B.V. https://doi.org/10.1016/j.eij.2015.06.005

Islami, A. M., Maulani, I., Zumadila, R., Yoga Putra, A. M., & Santoso, B. J. (2023). Enhancing Anomaly Classification Over Log Files through Topic Modeling and Ensemble Methods.

Proceeding - International Conference on Information Technology and Computing 2023, ICITCOM 2023, 57–61. https://doi.org/10.1109/ICITCOM60176.2023.10442730

Khare, S. (2022). Accuracy Enhancement During Sentiment Analysis in Twitter Using CNN. *2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions), ICRITO 2022*. https://doi.org/10.1109/ICRITO56286.2022.9964733

Lakshmipathi N. (n.d.). *IMDB Dataset*. Https://Www.Kaggle.Com/Datasets/Lakshmi25npathi/Imdb-Dataset-of-50k-Movie-Reviews. Retrieved September 29, 2024, from https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews

Liu, N., & Zhao, J. (2023). Recommendation System Based on Deep Sentiment Analysis and Matrix Factorization. *IEEE Access*, *11*, 16994–17001. https://doi.org/10.1109/ACCESS.2023.3246060

Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, *5*(4), 1093–1113. https://doi.org/10.1016/j.asej.2014.04.011

Milano, S., Taddeo, M., & Floridi, L. (2020). Recommender systems and their ethical challenges. *AI and Society*, *35*(4), 957–967. https://doi.org/10.1007/s00146-020-00950-y

Nagoya Kōgyō Daigaku, & Institute of Electrical and Electronics Engineers. (n.d.). *ICCCS 2018 : 2018 3rd International Conference on Computer and Communication Systems : April 27-30, 2018, Nagoya, Japan.*

Negara, E. S., Triadi, D., & Andryani, R. (n.d.). *Topic Modelling Twitter Data with Latent Dirichlet Allocation Method*.

Oswalt Manoj, S., Catherine Chandralekha, T., Dharshini, R. M., & Hemadharsini, P. (2023). Sentirec - A Sentiment Analysis with Recommendation System. *Proceedings of the 8th International Conference on Communication and Electronics Systems, ICCES 2023*, 694–701. https://doi.org/10.1109/ICCES57224.2023.10192840

Pavitha, N., Pungliya, V., Raut, A., Bhonsle, R., Purohit, A., Patel, A., & Shashidhar, R. (2022). Movie recommendation and sentiment analysis using machine learning. *Global Transitions Proceedings*, *3*(1), 279–284. https://doi.org/10.1016/j.gltp.2022.03.012

PyTorch Website. (n.d.). Https://Pytorch.Org/Docs/Stable/Index.Html. Retrieved September 26, 2024, from https://pytorch.org/docs/stable/index.html

Qiang, J., Qian, Z., Li, Y., Yuan, Y., & Wu, X. (2022). Short Text Topic Modeling Techniques, Applications, and Performance: A Survey. In *IEEE Transactions on Knowledge and Data Engineering* (Vol. 34, Issue 3, pp. 1427–1445). IEEE Computer Society. https://doi.org/10.1109/TKDE.2020.2992485

Rahman, A., & Hossen, M. S. (2019, September 1). Sentiment Analysis on Movie Review Data Using Machine Learning Approach. *2019 International Conference on Bangla Speech and Language Processing, ICBSLP 2019*. https://doi.org/10.1109/ICBSLP47725.2019.201470

Ramesh, R., & Vijayalakshmi, S. (2022). Improvement to Recommendation system using Hybrid techniques. *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering, ICACITE 2022*, 778–782. https://doi.org/10.1109/ICACITE53722.2022.9823879

Ray, S. K., & Susan, S. (2022). Performance Evaluation using Online Machine Learning Packages for Streaming Data. *2022 International Conference on Computer Communication and Informatics, ICCCI 2022*. https://doi.org/10.1109/ICCCI54379.2022.9741068

Samprith Jagtap, D. (2024). Cross-Media Topic Modeling: A Comparative Analysis of NMF, LDA and CoRex combined with BERT on Quora and Twitter. *ICCDS 2024 - International Conference on Computing and Data Science*. https://doi.org/10.1109/ICCDS60734.2024.10560395

SCAD College of Engineering and Technology, & Institute of Electrical and Electronics Engineers. (n.d.). *Proceedings of the 4th International Conference on Trends in Electronics and Informatics (ICOEI 2020) : 15-17, June 2020*.

Scikit-Learn. (n.d.). Https://Scikit-Learn.Org/Stable/. Retrieved September 26, 2024, from https://scikit-learn.org/stable/

Shah, D., Shokeen, C., Khanzode, S., Kale, P., & Kn, D. (2023). Recommendation System using NLP and Collaborative Filtering. *2023 IEEE 8th International Conference for Convergence in Technology, I2CT 2023*. https://doi.org/10.1109/I2CT57861.2023.10126261

Shivahare, B. D., Singh, A. K., Uppal, N., Rizwan, A., Vaathsav, V. S., & Suman, S. (2022). Survey Paper: Study of Natural Language Processing and its Recent Applications. *Proceedings - 2022 2nd International Conference on Innovative Sustainable Computational Technologies, CISCT 2022*. https://doi.org/10.1109/CISCT55310.2022.10046440

Udupa, A., Adarsh, K. N., Aravinda, A., Godihal, N. H., & Kayarvizhy, N. (2022). An Exploratory Analysis of GSDMM and BERTopic on Short Text Topic Modelling. *2022 4th International Conference on Cognitive Computing and Information Processing, CCIP 2022*. https://doi.org/10.1109/CCIP57447.2022.10058687

Universitas Sumatera Utara, & Institute of Electrical and Electronics Engineers. (n.d.). *2020 4th International Conference on Electrical, Telecommunication and Computer Engineering (ELTICOM) : proceedings : Medan, Indonesia, September 3, 2020.*

Vedaswi, K., Krishna, N. V., Poojitha, T. V., Lokesh, P., Ashesh, K., & Kumar, P. M. A. (2023). Movie Recommendation using Collaborative filtering and Content-based Filtering

Approach. *6th International Conference on Inventive Computation Technologies, ICICT 2023 - Proceedings*, 821–826. https://doi.org/10.1109/ICICT57646.2023.10134213

Wu, Q., & Gong, X. (2021). A Study on the User Privacy-Preserving Personalized Recommendation Based on Random Perturbation for online Services in the Background of Big Data. *Proceedings - 2021 International Conference on Big Data and Intelligent Decision Making, BDIDM 2021*, 80–85. https://doi.org/10.1109/BDIDM53834.2021.00023

Xu, X. (2019). Matrix factorization recommendation algorithm based on deep neural network. *2019 2nd International Conference on Information Systems and Computer Aided Education, ICISCAE 2019*, 320–323. https://doi.org/10.1109/ICISCAE48440.2019.221643

Yamunathangam, D., Priya, C. B., Shobana, G., & Latha, L. (2021). An Overview of Topic Representation and Topic Modelling Methods for Short Texts and Long Corpus. *2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation, ICAECA 2021*. https://doi.org/10.1109/ICAECA52838.2021.9675579