NLP Mini-Project: Sentiment Analysis on Movie Reviews

Tony Lauze
ENSAE Paris - 2024-2025
tony.lauze@ensae.fr

Abstract

This project is based on the article *Learning Word Vectors for Sentiment Analysis* by Maas et al. (2011) [5], and aims to compare different approaches to **sentiment analysis**. Using the dataset introduced in that paper—which consists of positive and negative movie reviews from IMDb—we evaluate the performance of traditional machine learning algorithms (such as SVM and logistic regression) combined with text vectorization techniques, and compare them with more recent pre-trained deep learning models based on transformer architectures (e.g., BERT).

1 Problem description: sentiment analysis

Sentiment analysis, or opinion mining, is the process of determining if the sentiment expressed in a text is positive, negative, or neutral. It is a task in natural language processing (NLP) that helps identify and analyze the sentiment in content like movie reviews or product feedback.

Improving sentiment analysis techniques is important because it allows organizations to gain valuable insights from public opinion and customer feedback. This helps with decision-making, improving products, and creating effective marketing strategies. It can also help monitor customer sentiment, manage online reputation, and stay up-to-date with market trends.

Additionally, sentiment analysis plays a role in social and political discussions, helping researchers and policymakers understand public opinions on important issues, which may lead to better decision-making in a digital and connected world.

Movie review platforms, such as IMDB or Rotten Tomatoes, allow users to share their opinions about films. Since these reviews often take the form of short analytical texts accompanied by a score based on their appreciation of the movie, these platforms provide valuable datasets for training and evaluating models designed to classify the sentiment of a text.

The field of sentiment analysis has evolved through three main generations of methods: rule-based methods, traditional machine learning models, and deep learning models using neural networks, with the latest being transformer-based models such as BERT.

2 State-of-the-art methods and results

One traditional but now outdated rule-based approach for sentiment analysis of movie reviews involves computing the polarity score of a review by averaging the polarity scores of the words it contains.

For example, the `TextBlob` model provides each text with a polarity score (indicating sentiment) and a subjectivity score. It focuses particularly on adjectives, as they often carry strong emotional meaning. The overall polarity of a sentence is calculated as the weighted average of the sentiment scores of its individual words. `TextBlob` relies on the Pattern lexicon, which was created using online customer reviews. When combined with a Naive Bayes classifier, `TextBlob` can also be used for sentiment prediction tasks. However, as noted by Tetteh and Thushara (2023) [11], this approach tends to perform poorly, achieving only 73% accuracy on the IMDB dataset.

Today, however, machine learning methods are more commonly used, including traditional supervised ML techniques, deep learning methods, and more recently, large language models (LLMs). A recent review by Rahman Jim et al. (2024) [7] covers widely used datasets, preprocessing techniques, evaluation metrics, and discusses key models (ML, DL, LLMs) used in sentiment analysis.

Numerous studies have applied machine learning methods to movie sentiment analysis (see [1], [8], [10]). These methods often include SVM, Logistic Regression, and Naive Bayes. The performance of these models typically falls between 70% (as in Baid, Gupta, and Chaplot (2017) [3]) and just below 90% at most (see Amulya et al. (2022) [2], for instance), depending on the dataset and the algorithm used.

With the rise of deep learning, more complex architectures like RNNs, CNNs, and LSTMs emerged. However, the articles cited above do not necessary highlight a substantial increase in performance, especially when considering the significantly higher training times.

The advent of LLMs, especially with the introduction of BERT (Devlin et al. (2019) [4]), has transformed natural language processing. BERT is pre-trained on large corpora and uses a masked language modeling approach to capture bidirectional context. Fine-tuned models for classification tasks, generally outperform previous methods and have become the new standard. More recently, models like GPT-3/4 or BART allow sentiment classification tasks to be performed without specific training (zero-shot) or with few examples (few-shot).

These methods generally result in a significant improvement in sentiment classification performance due to their better understanding of syntax, which comes from training on large corpora. For instance, *Sentiment Analysis of Movie Reviews Using BERT* by Nkhata et al. (2025) [6] explores the use of BERT for sentiment analysis of movie reviews. In their approach, the authors fine-tune the BERT model with two additional input layers, and a BiLSTM (Bidirectional Long Short-Term Memory) layer at the end of the model. Their fine-tuned BERT model, when applied to the IMDB dataset, achieves an accuracy of approximately 97%.

3 Data

In this section, we provide an overview of the dataset used in this project and present a descriptive analysis through the application of the unsupervised Latent Dirichlet Allocation (LDA) technique.

3.1 Presentation of the dataset

The dataset introduced by Maas et al. (2011) [5] consists of 50,000 movie reviews, evenly split into 25,000 for training and 25,000 for testing. The overall distribution of labels is balanced, with 25,000 positive and 25,000 negative reviews in total. Additionally, 50,000 unlabelled reviews are included for unsupervised learning, but will not be used in this project.

To ensure variety, no more than 30 reviews are allowed per movie, as reviews for the same movie may have correlated ratings. Moreover, the training and testing sets contain disjoint sets of movies, preventing performance gains from memorizing movie-specific terms associated with observed labels.

In the labelled training and testing sets, a negative review has a score of 4 or lower out of 10, while a positive review has a score of 7 or higher. Neutral reviews are thus excluded from these sets.

3.2 Descriptive analysis using Latent Dirichlet Allocation

First, Figure 1 confirms that the reviews are evenly distributed between positive and negative sentiments. Figure 2 shows that the length distribution is similar for both types of reviews: most of them are under 250 words, with a concentration around 100 words.

In Maas et al. (2011) [5], LDA is used not only for topic modeling but also as part of a predictive framework that improves the learning of sentiment-associated word meanings. In this project, however, we use LDA solely to provide an overview of the underlying themes in the dataset.

Latent Dirichlet Allocation is a probabilistic generative model for documents that assumes each document is composed of a mixture of hidden topics. Each topic is characterized by a probability distribution over words, denoted $p(w|T)$, which represents the likelihood of observing word w within topic T. By training an LDA model with k topics, one can build a k-dimensional representation of words, where each word is associated with its probabilities across topics. This results in a word–topic matrix whose rows reflect the semantic profile of each word across the discovered topics.

When combined with WordClouds package in Python, LDA allows us to visualize the most frequent words in each topic and to interpret what the topics are about.

Figure 3 shows the word clouds of the 10 discovered topics. We observe that Topic 6 refers to television series, Topic 5 seems to involve family-related stories, Topic 2 focuses on the musical genre, and Topic 3 appears to correspond to western films.

Looking more closely at the topic distribution for two example reviews (Figures 4 and 5)—one positive and the other negative, both about a family film—we see that Topics 7 and 8 seem to reflect sentiment. Topic 8 appears more in the negative review, while Topic 7 is dominant in the positive one.

Figure 6 partially confirms this analysis. It shows the dominant topic of each review and counts the number of reviews for each topic. Topic 8 indeed dominates the negative reviews, while the positive reviews are more evenly spread across Topics 1, 7, and 8. In this sense, we echo one of the conclusions from Maas et al. (2011) [5]: topic modeling alone does not capture sentiment very effectively.

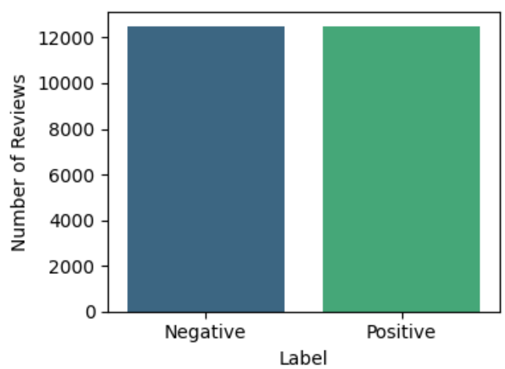

Figure 1: Number of positive and negative reviews in the training dataset

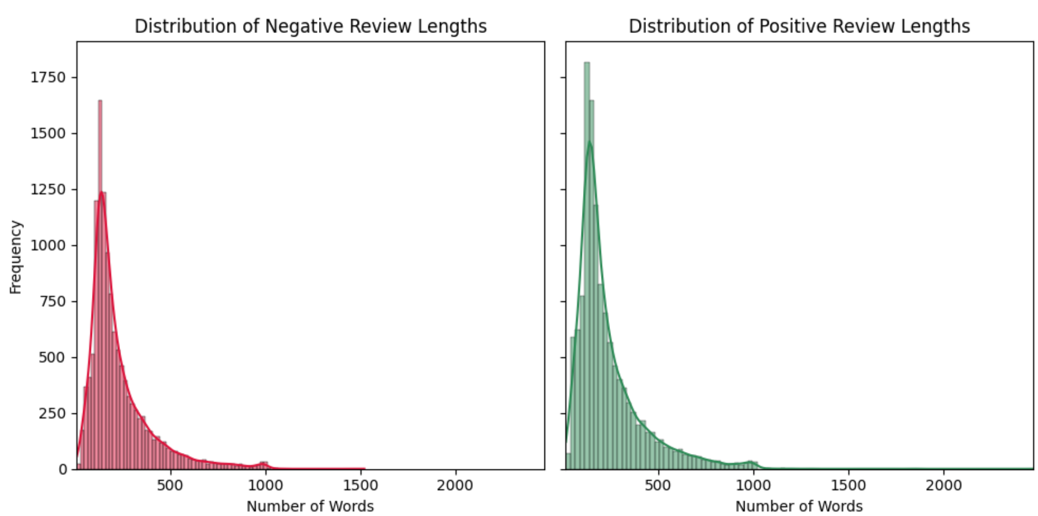

Figure 2: Length distribution of reviews depending on the sentiment

Figure 3: Most frequent words in each of the 10 underlying topics uncovered by LDA analysis

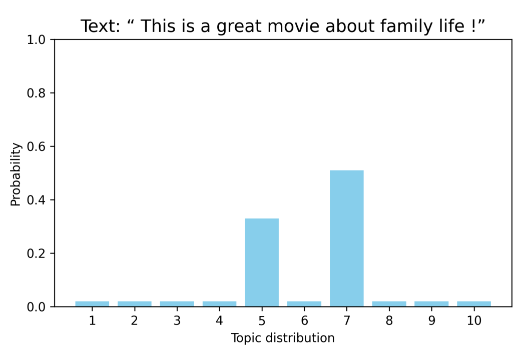

Figure 4: Topics composition for a text conveying positive sentiment about a family movie

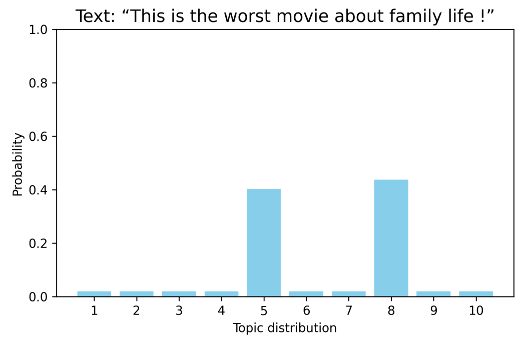

Figure 5: Topics composition for a text conveying negative sentiment about a family movie

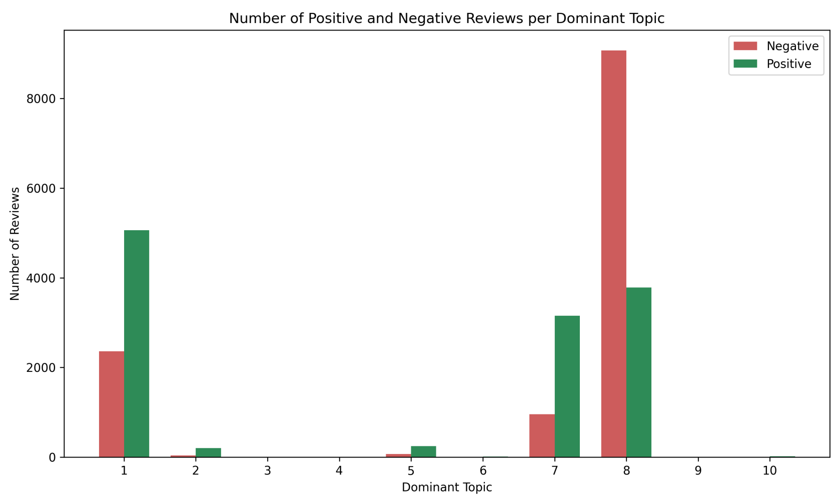

Figure 6: Number of reviews per theme, based on the first-occurring topic in the decomposition

4 Analysis

For this project, our goal is first to present some benchmark results by applying some traditional ML algorithms to vectorized version of texts, in a way that is close to what is done in the article by Maas et al. (2011) [5]. Then, we check whether applying some pre-trained light version of BERT algorithm can improve the performance of the prediction. This section presents briefly both approaches and details how models are structured and trained.

4.1 Traditional ML Algorithms: SVC and Logistic Regression with TF-IDF and LSA Preprocessing

Preparation of the training data To prepare the movie reviews for traditional machine learning algorithms, we first convert the raw text into a numerical representation. We use `TfidfVectorizer` from `scikit-learn`, which builds a bag-of-words representation weighted by *Term Frequency-Inverse Document Frequency* (TF-IDF). This method down-weights words that appear frequently across the entire corpus, as these tend to be less informative.

Once the TF-IDF matrix is constructed, we apply *Latent Semantic Analysis* (LSA), corresponding to a *Truncated Singular Value Decomposition* (SVD), using `TruncatedSVD` from `scikit-learn`. This dimensionality reduction technique projects the high-dimensional term-document matrix into a lower-dimensional latent space, allowing us to capture the main semantic structure of the corpus and reduce noise.

We then rely on two ML algorithms that are frequently used in the litterature for sentiment classification : Logistic regression and SVC.

Logistic Regression Logistic regression is a linear classifier that estimates the probability of a binary label $y \in \{0, 1\}$ given an input vector $x \in \mathbb{R}^d$. It models the conditional probability using the sigmoid function:

$$P(y = 1 \mid x) = \sigma(w^\top x + b) = \frac{1}{1 + e^{-(w^\top x + b)}}$$

The parameters w and b are learned by minimizing the regularized logistic loss over the training set:

$$\mathcal{L}(w, b) = -\sum_{i=1}^{n} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] + \lambda \|w\|^2$$

where $\hat{y}_i = \sigma(w^\top x_i + b)$, and λ controls the strength of $L2$ regularization.

We use the `LogisticRegression` class from `scikit-learn`, with the regularization strength $C = 1/\lambda$ selected through cross-validation.

Support Vector Classifier (SVC) Support Vector Machines (SVM) seek to find a hyperplane that separates the data with the largest margin. Given a dataset $\{(x_i, y_i)\}_{i=1}^n$ with $y_i \in \{-1, 1\}$, the optimization problem is:

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{n} \max\left(0, 1 - y_i \left(K(x_i, x) + b\right)\right)$$

Where $K(x_i, x_j)$ is the kernel function, which computes the inner product in a higher-dimensional feature space, and where C controls the trade-off between maximizing the margin and minimizing the classification error. We use the `SVC` class from `scikit-learn`, experimenting with different values of C and kernel functions.

Model Training and Cross-Validation To train the models and select optimal hyperparameters, we perform k-fold cross-validation with $k = 3$. The training data is split into 3 subsets; for each round, one fold is held out as validation data while the others are used for training. The best combination of hyperparameters is selected based on validation accuracy, and the final model is trained on the entire training set using those values.

4.2 BERT and DistilBERT Models for Sentiment Analysis

BERT Architecture BERT was introduced by researchers from Google [4] and represents a significant advancement in NLP. Unlike previous models that processed text in a single direction (either left-to-right or right-to-left), BERT focuses on pre-training deep bidirectional representations from unlabeled text by jointly conditioning on both left and right contexts in all layers. This bidirectional approach allows BERT to develop a more nuanced understanding of language context and semantics.

There are two primary versions of BERT:

- **BERT$_{\textbf{BASE}}$**: Contains 12 layers (transformer blocks), 768 hidden states, 12 attention heads, and 110M parameters
- **BERT$_{\textbf{LARGE}}$**: Features approximately twice the specifications with 24 layers, 1024 hidden states, 16 attention heads, and 340M parameters

DistilBERT Architecture DistilBERT was introduced by [9] as a distilled version of BERT. It represents an application of Knowledge Distillation, a compression technique proposed by **?**, where a smaller student model is trained to reproduce the behaviour of a larger teacher model.

There are 2 main differences from BERT$_{\text{BASE}}$:

- **Reduced layer count**: 6 layers instead of the original 12
- **Omission of token-type embeddings**: No Next Sentence Prediction objective

These modifications result in a model with approximately **66 million parameters** - about 40% fewer than $\text{BERT}_{\text{BASE}}$ - while maintaining 97% of its performance. Notably, DistilBERT offers 60% faster inference speed on CPU compared to its teacher model, making it more suitable for production environments and resource-constrained settings.

DistilBERT for Sentiment Analysis For our sentiment analysis task, we specifically employ **distilbert-base-uncased-finetuned-sst-2-english**, a version of the base DistilBERT that has been fine-tuned on the Stanford Sentiment Treebank v2 (SST-2) dataset. The SST-2 corpus comprises 11,855 individual sentences extracted from movie reviews, which were further expanded into 215,154 unique phrases. Each phrase was annotated for sentiment (positive or negative) by three human judges, creating a robust dataset for binary sentiment classification tasks.

This model is particularly well-suited for our purposes as it has been specifically optimized for binary sentiment classification in English, exactly matching our task of analyzing movie review sentiments.

Implementation Details We implement our sentiment analysis pipeline using the `Transformers` library from Hugging Face, which provides a convenient interface for loading and utilizing pre-trained models. Our sentiment prediction function processes the text inputs through the tokenizer, passes them to the model, and interprets the output logits to determine the sentiment classification. To ensure computational efficiency, we conduct our experiments on a subsample of 2,000 reviews from our dataset.

5 Results

5.1 Classification metrics

To measure the efficiency of the models, we consider four classification metrics from the Python Scikit library (Accuracy, Precision, Recall, F-score). As a reminder, these metrics are defined as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

$$F\text{-score} = \frac{2(\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}} \tag{4}$$

where TP represents True Positive predictions, TN represents True Negatives, FP and FN denote False Positives and False Negatives respectively.

5.2 Results and model comparison

First, we observe that the two traditional ML algorithms, combined with a TF-IDF + LSA preprocessing pipeline, achieve an accuracy of around 0.86. This is consistent with the performance reported in Maas et al. (2011) [5], and lies in the upper range of what is typically found in the literature for this type of model.

In contrast, using the DistilBERT model does lead to higher sentiment prediction accuracy, reaching an average accuracy of 0.90 on the 2,000 test samples used for evaluation. While prediction with DistilBERT is slower, it does not require a training phase, unlike traditional ML models.

None of the three models perform noticeably better or worse on one sentiment class over the other (see Figures 7, 8, 9), which confirms that the dataset is well balanced between positive and negative reviews.

Table 1: Performance Comparison of our models

Model	Accuracy	Precision		F1-Score	
		Class 0	Class 1	Class 0	Class 1
DistilBERT (fine-tuned on SST-2)	0.90	0.88	0.91	0.90	0.90
Logistic Regression	0.85	0.86	0.85	0.85	0.85
SVC (RBF kernel)	0.86	0.86	0.86	0.86	0.86

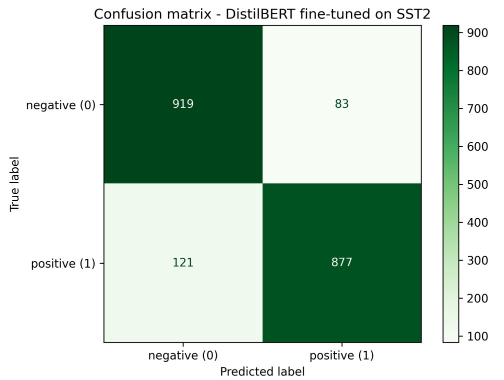

Figure 7: Confusion matrix for DistilBERT fine-tuned model

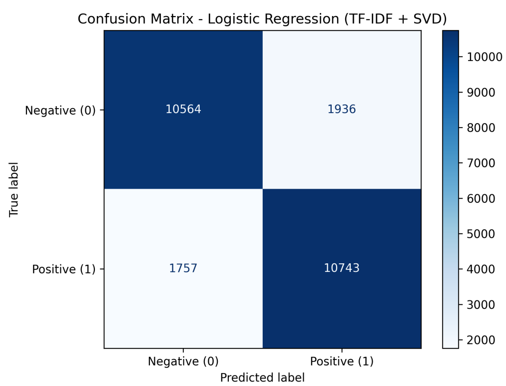

Figure 8: Confusion matrix for Logistic Regression model

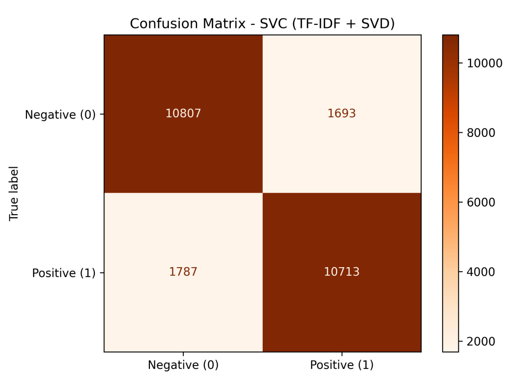

Figure 9: Confusion matrix for SVC (RBF kernel) model

6 Conclusion

The goal of this project was to compare traditional machine learning approaches and transformer-based models for sentiment analysis on movie reviews, making use of the dataset and methods used in Maas et al. (2011) [5].

Results show that the fine-tuned DistilBERT model achieves superior performance (90% accuracy) compared to traditional machine learning methods such as Logistic Regression (85%) and SVC with RBF kernel (86%). This tends to confirm the effectiveness of pre-trained language models for sentiment classification tasks, as they better capture contextual information and semantic nuances.

The traditional ML approaches combined with TF-IDF and LSA preprocessing still provide reasonable performance, consistent with results reported in previous literature and in our reference paper.

These findings seem to align with the current trend in NLP research, which shows that transformer-based architectures consistently outperform traditional methods across various text classification tasks.

References

[1] Noor Latiffah Adam, Nor Hanani Rosli, and Shaharuddin Cik Soh. "Sentiment analysis on movie review using Naıve Bayes". In: *2021 2nd international conference on artificial intelligence and data sciences (AiDAS)*. IEEE. 2021, pp. 1–6.

[2] K. Amulya et al. "Sentiment Analysis on IMDB Movie Reviews using Machine Learning and Deep Learning Algorithms". In: *2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)*. 2022, pp. 814–819. DOI: 10.1109/ICSSIT53264.2022.9716550.

[3] Palak Baid, Apoorva Gupta, and Neelam Chaplot. "Sentiment Analysis of Movie Reviews using Machine Learning Techniques". In: *International Journal of Computer Applications* 179 (Dec. 2017), pp. 45–49. DOI: 10.5120/ijca2017916005.

[4] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL]. URL: https://arxiv.org/abs/1810.04805.

[5] Andrew L. Maas et al. "Learning Word Vectors for Sentiment Analysis". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Ed. by Dekang Lin, Yuji Matsumoto, and Rada Mihalcea. Portland, Oregon, USA: Association for Computational Linguistics, June 2011, pp. 142–150. URL: https://aclanthology.org/P11-1015/.

[6] Gibson Nkhata, Usman Anjum, and Justin Zhan. *Sentiment Analysis of Movie Reviews Using BERT*. 2025. arXiv: 2502.18841 [cs.CL]. URL: https://arxiv.org/abs/2502.18841.

[7] Jamin Rahman Jim et al. "Recent advancements and challenges of NLP-based sentiment analysis: A state-of-the-art review". In: *Natural Language Processing Journal* 6 (2024), p. 100059. ISSN: 2949-7191. DOI: 10.1016/j.nlp.2024.100059. URL: https://www.sciencedirect.com/science/article/pii/S2949719124000074.

[8] Tirath Prasad Sahu and Sanjeev Ahuja. "Sentiment analysis of movie reviews: A study on feature selection & classification algorithms". In: *2016 International Conference on Microelectronics, Computing and Communications (MicroCom)*. Ieee. 2016, pp. 1–6.

[9] Victor Sanh et al. *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. 2020. arXiv: 1910.01108 [cs.CL]. URL: https://arxiv.org/abs/1910.01108.

[10] Isaiah Steinke et al. "Sentiment analysis of online movie reviews using machine learning". In: *Int. J. Adv. Comput. Sci. Appl* 13.9 (2022), pp. 618–624.

[11] Maxwell Tetteh and Mg Thushara. "Sentiment Analysis Tools for Movie Review Evaluation - A Survey". In: *2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS)*. 2023, pp. 816–823. DOI: 10.1109/ICICCS56967.2023.10142834.