Comparative Study of Machine Learning and Deep Learning Models for sentiment analysis

Topic Number: **02**

Sayanjit Ukil (2352146)

Under the Supervision and Guidance Of

Prof.(Dr.) Prativa Agarwalla

Department Of Electronics and Communications Engineering

Heritage Institute Of Technology

1 Introduction

1.1 Natural Language Processing

Natural Language Processing (NLP) is a field of computer science that focuses on enabling computers to understand, interpret, manipulate and generate human language(1). A major goal of NLP is to bridge the gap between human communication and computer understanding. It combines computational linguistics—rule-based modeling of human language—with statistical, machine learning, and deep learning models. The early roots of NLP date back to the 1950's, when John Searle introduced the Chinese Room Experiment(2), which involves feeding a collection of rules (for example, a Chinese playbook) to a computer, which then emulates natural language understanding (or other NLP tasks) by applying those rules to the data it confronts. The introduction of machine learning algorithms, in the late 1980's, revolutionized the field, partly due to the increase in computational power, in accordance with the Moore's Law(3), and partly due to the gradual lessening of the Chomskyan theories of linguistics(4).

In present times, the field has undergone another significant paradigm shift, this time driven by the advent of deep learning(5) methodologies. The development of the Transformer architecture(6), in particular, has been a watershed moment, enabling the creation of Large Language Models (LLMs) with heightened capabilities (7). Consequently, These models have achieved state-of-the-art performance across a diverse spectrum of NLP tasks, including machine translation, text summarization, and question-answering, thus altering the landscape of what is computationally achievable in the domain of language understanding and generation.

1.2 Sentiment Analysis

Sentiment analysis, also known as opinion mining, is a prominent application of Natural Language Processing (NLP) that computationally identifies and categorizes the opinions expressed in a piece of text. The primary goal is to determine the writer's attitude or emotional tone towards a particular topic, product, or service. This is typically classified into three main polarities: positive, negative, or neutral. In practice, sentiment analysis is widely used by businesses to gauge public opinion and customer feedback from vast amounts of unstructured data, such as social media posts, online reviews, and survey responses, allowing them to make more informed decisions (8).

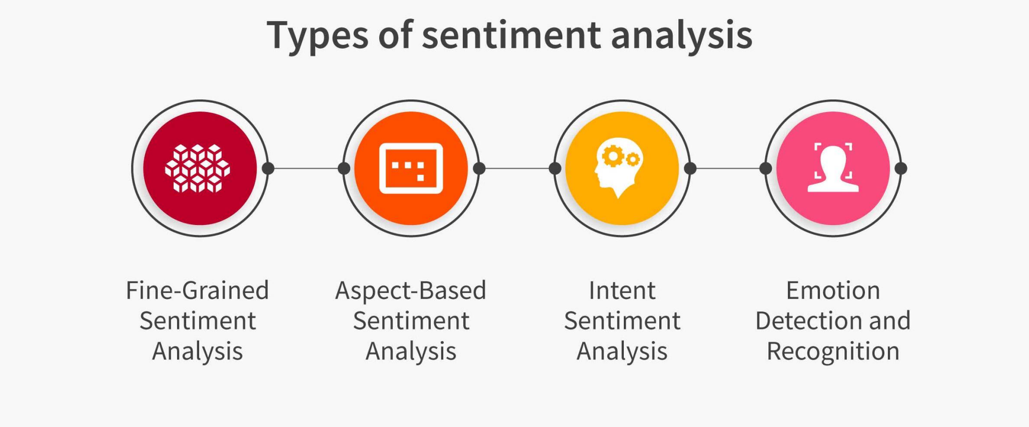

Types of sentiment analysis
Fine-Grained Sentiment Analysis
Aspect-Based Sentiment Analysis
Intent Sentiment Analysis
Emotion Detection and Recognition

Figure 1: Types of sentiment analysis

As shown in Figure 1, natural language processing can be classified into four broad types, namely :

- Fine-Grained Sentiment Analysis : This approach expands on basic sentiment by classifying it into more precise categories of intensity.
- Aspect-based Sentiment Analysis : This method breaks down text to identify sentiments associated with specific features or aspects of a product or service.
- Intent Sentiment Analysis : This type of analysis focuses on determining the underlying purpose or intention behind a user's text. It seeks to understand what the user wants to do.
- Emotion Detection and Recognition : Going beyond polarity, this technique aims to identify specific human emotions conveyed in the text. It can detect feelings such as happiness, sadness, anger, surprise, or fear.

1.3 Relevance and Importance

Sentiment Analysis has widespread applications in several domains, like business and marketing, where it is used to monitor brand reputation and conduct market research by analyzing customer reviews. It is pivotal in customer service for automatically prioritizing support tickets, in finance for informing algorithmic trading strategies by interpreting market sentiment, and in politics for tracking voter opinions and shaping campaign strategies. Furthermore, its role in social media monitoring is crucial for gauging public reaction to events and managing crises in real-time.

1.4 Objective

The primary objective is to provide a detailed comparative analysis between different classical Machine Learning models and Deep Learning architectures by benchmarking their performance and computational cost on openly available sentiment analysis datasets.

The rest of this investigative report is organized as follows: Section-2 details the experimental methods and the datasets used for benchmarking. Section-3 presents the detailed analysis of experimental results. Finally, conclusions are provided in Section-4.

2 Methodology

2.1 Dataset Details

The dataset which is used for benchmarking is the Sentiment Analysis Dataset from Kaggle, available at https://www.kaggle.com/datasets/abhi8923shriv/sentiment-analysis-dataset/data. It consists of several tweets and comments gathered from many different social media sites. The distribution of classes is shown in figure 2.

Figure 2 shows that the dataset has more than 10,000 neutral tweets, nearly 8,000 negative tweets and nearly 8,500 positive tweets. Some dataset samples are provided in table 1.

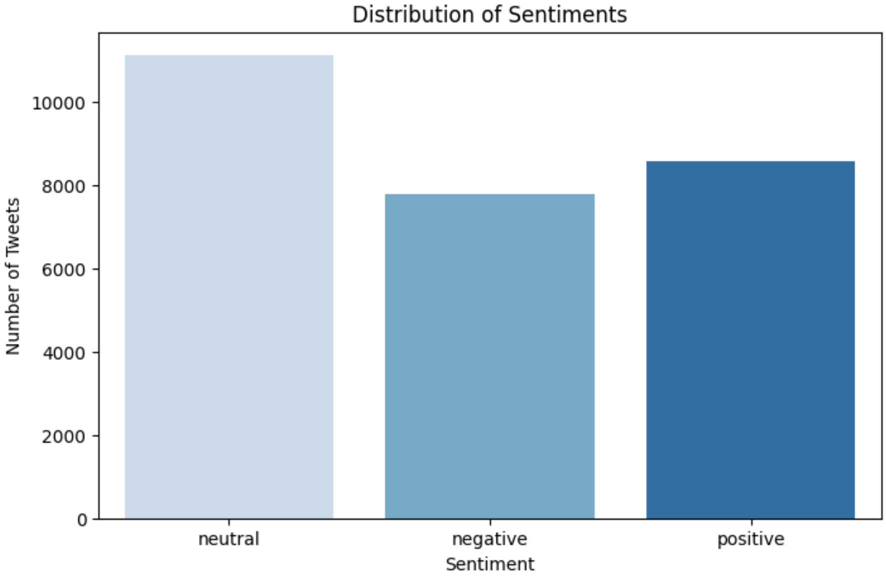


Figure 2: Distribution of classes

Text	**Sentiment**
Sooo SAD I will miss you here in San Diego!!!	negative
I‘d have responded if I were going	neutral
That‘s very funny. Cute kids.	positive
Born and raised in NYC and living in Texas for the past 10 years! I still miss NY	negative
On the way to Malaysia...no internet access to Twit	negative

Table 1: Training data samples

2.2 Data Pre-processing

Various data pre-processing(16) techniques were used to make the dataset, as a whole, more compatible with the algorithms and the models. The data pre-processing pipeline is illustrated in figure 3.

2.2.1 Data Cleaning

Data cleaning involves cleaning the dataset in order to feed the model a clean training data so that it can learn and generalize more robustly. Data cleaning involves removing unnecessary features which are irrelevant to the current task at hand, such as user Id’s and timestamps. This decreases both the noise and the computational complexity.

Another integral part of data cleaning is the handling of missing or null values which is essential as machine learning models cannot process null or empty entries. The two primary strategies are removal, when the number of missing values are small, and imputation, which involves filling the missing values with substitute entries.

The Sentiment Analysis dataset which has been used for benchmarking also consisted of several feature columns like *textID*, *selected-text*, *Time of tweet*, *Age of user*, among others. These were dropped and the cleaned dataset is represented in table 1.

2.2.2 Text Cleaning

Text cleaning is the process of standardizing raw text by removing noise that provides little value for analysis, such as punctuation, special characters, numbers, URLs, and HTML tags. This is typically achieved using **regular expressions (re)**, which define patterns to find and replace these unwanted elements. A key part of this process is also converting all text to lowercase, which ensures that words like "Happy" and "happy" are treated as the same token, reducing the vocabulary size and helping the model recognize word patterns more effectively.

2.2.3 Label Encoding

Label encoding is the process of converting categorical text labels, such as "positive," "negative," and "neutral," into a numerical format that machine learning models can understand. Since algorithms perform mathematical calculations, they cannot operate on string values for the target variable. This technique assigns a unique integer to each distinct category.

Scikit Learn's Label Encoder was used to encode the sentiment values to three distinct labels-0 for 'negative', 1 for 'neutral' and 2 for 'positive'.

2.2.4 Vectorization

TF-IDF (Term Frequency-Inverse Document Frequency)(17) is a feature extraction technique that converts a corpus of text into a meaningful matrix of numerical values. Instead of just counting words, it calculates an "importance" score for each word in a document. This score is a product of two metrics: Term Frequency (TF), which measures how often a word appears in a single document, and Inverse Document Frequency (IDF), which penalizes words that are common across all documents. The resulting TF-IDF score gives higher weight to words that are frequent in a specific document but rare overall, making it an effective way to represent the most significant words in the text.

The Bag-of-Words (BoW)(19) model is a foundational feature extraction technique used to convert text into a numerical format suitable for machine learning algorithms. This method represents a document as a vector based on the frequency of each word from a predefined vocabulary. In this approach, grammatical structure and word order are disregarded, and the text is treated as an unordered collection. The resulting vectors are typically high-dimensional and sparse, with each dimension corresponding to the occurrence count or frequency of a unique word in the corpus.

Word2Vec(20) is a predictive model utilized to generate dense vector representations of words, known as word embeddings. Unlike the sparse vectors from BoW, Word2Vec captures the contextual and semantic relationships between words by learning from their co-occurrence patterns in a large text corpus. Using a shallow neural network, it maps words to a continuous, low-dimensional vector space where semantically similar words are positioned in close proximity. These learned embeddings serve as powerful, context-rich features that effectively preserve linguistic nuances for downstream natural language processing tasks.

2.2.5 Tokenization

Tokenization for Transformer models is a process that converts raw text into a numerical format suitable for a transformer model's architecture. Unlike traditional methods that split text by words, Transformers employ subword tokenization algorithms like Byte-Pair Encoding (BPE) or WordPiece. This approach breaks down rare or unknown words into smaller, meaningful subunits (e.g., ”unseen” might become ”un” and ”##seen”) while keeping common words as single tokens. This method offers a crucial advantage: it effectively handles out-of-vocabulary words and maintains a fixed, manageable vocabulary size, all while allowing the model to understand word morphology. Finally, special tokens such as [CLS] for classification, [SEP] for separating sentences, and [PAD] for ensuring uniform sequence length are added to the numerical sequence to provide the model with essential structural context.

For DistilBERT and RoBERTa, the DistilBertTokenizerFast and the RobertaTokenizerFast from the transformers library were used, respectively.

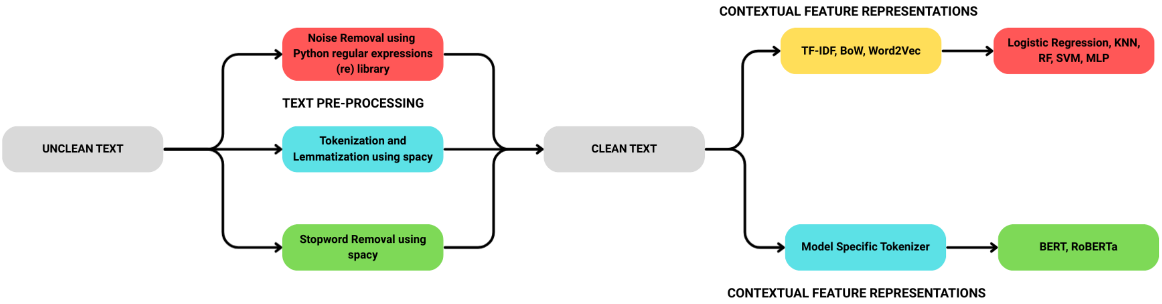

CONTEXTUAL FEATURE REPRESENTATIONS
Noise Removal using Python regular expressions (re) library
TF-IDF, BoW, Word2Vec
Logistic Regression, KNN, RF, SVM, MLP
TEXT PRE-PROCESSING
UNCLEAN TEXT
Tokenization and Lemmatization using spacy
CLEAN TEXT
Stopword Removal using spacy
Model Specific Tokenizer
BERT, RoBERTa
CONTEXTUAL FEATURE REPRESENTATIONS

Figure 3: Data Pre-processing Pipeline

2.3 Experimental Setup

2.3.1 Models and Methods

To compare the effectiveness of different classifiers for sentiment analysis, various machine learning and deep learning models were evaluated. The baseline models included Logistic Regression(9), K-Nearest Neighbors (KNN)(10), Random Forest(11), Support Vector Machines (SVM)(12) and XgBoost(13). The deep learning and transformer models included a classic Multi-Layered Perceptron (MLP), DistilBERT(14) (Bidirectional Encoder Representations from Transformers) and its robustly optimized variant RoBERTa(15).

2.3.2 Evaluation Metrics

To evaluate the performance of each model, four standard classification metrics were employed:

- Accuracy: The ratio of correctly predicted instances to the total instances.

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

- Precision: The ratio of true positives to the total predicted positives, indicating the model's accuracy in predicting the positive class.

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

- Recall: The ratio of true positives to the total actual positives, measuring the model's ability to identify all positive instances.

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

- F1-Score: The harmonic mean of Precision and Recall, providing a single score that balances both metrics. It is particularly useful for datasets with imbalanced classes.

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{4}$$

3 Experimental Results

Table 2: Model Accuracy Comparison Across Feature Representations.

Method	**TF-IDF (%)**	**BoW (%)**	**Word2Vec (%)**
Logistic Regression	**73.40**	**71.73**	**59.11**
XGBoost	68.90	68.22	57.70
Random Forest	71.76	69.78	55.74
SVM	72.47	71.56	55.86
KNN	54.39	57.47	51.44

Based on the experimental results of the traditional models presented in table 2, Logistic Regression paired with the TF-IDF(17) feature representation is the optimal combination for this task, achieving the highest accuracy of 73.40%. Notably, linear models like Logistic Regression and SVM consistently outperformed the ensemble tree-based methods, while KNN proved to be the least effective classifier. In terms of feature engineering, the frequency-based methods of TF-IDF(17) and BoW(19) yielded significantly better results than the averaged Word2Vec(20) embeddings, which caused a substantial drop in performance across all models. This suggests that sentiment analysis as a whole is well-suited to linear decision boundaries in a high-dimensional sparse feature space, where the keyword-specific vectors captured by TF-IDF are more impactful than the generalized semantic meaning from averaged embeddings.

Table 3 reveals a significant performance advantage for the transformer-based models, with RoBERTa achieving the highest accuracy at 79.77%. DistilBERT follows very closely with

Table 3: Performance Comparison of the deep learning methods

Methods	**Accuracy**	**Precision**	**Recall**	**F1-score**
ANN (MLP)	70.15%	0.72	0.70	0.71
DistilBERT	79.63%	0.80	0.80	0.80
RoBERTa	**79.77%**	**0.80**	**0.80**	**0.80**

an accuracy of 79.63% and identical precision, recall, and F1-scores of 0.80, indicating nearly equivalent performance. The MLP matches the performance of the traditional algorithms, reaching 70.15% accuracy and showing lower scores across all other evaluation metrics, when compared to the transformer models. However, it may be noted that both RoBERTa and DistilBERT are computationally very expensive, with nearly 125M and 67M parameters, respectively.

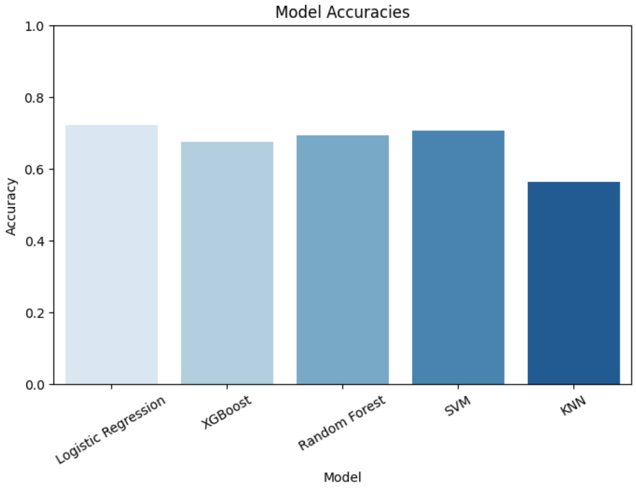


Figure 4: Accuracy of the ML algorithms

Figure 4 illustrates the accuracies of the different models. The Classification Metrics, per model, are detailed in Figure 5. Once again, it is evident that Logistic Regression and SVM are the top performing models, achieving the highest scores across all the four classification metrics. The KNN model is a notable outlier, this can be attributed to a fundamental weakness: the model's core mechanism, which relies on measuring distances between data points, becomes ineffective when dealing with the high-dimensional and sparse vectors used to represent text. This is sometimes referred to as the 'Curse of Dimensionality'(18).

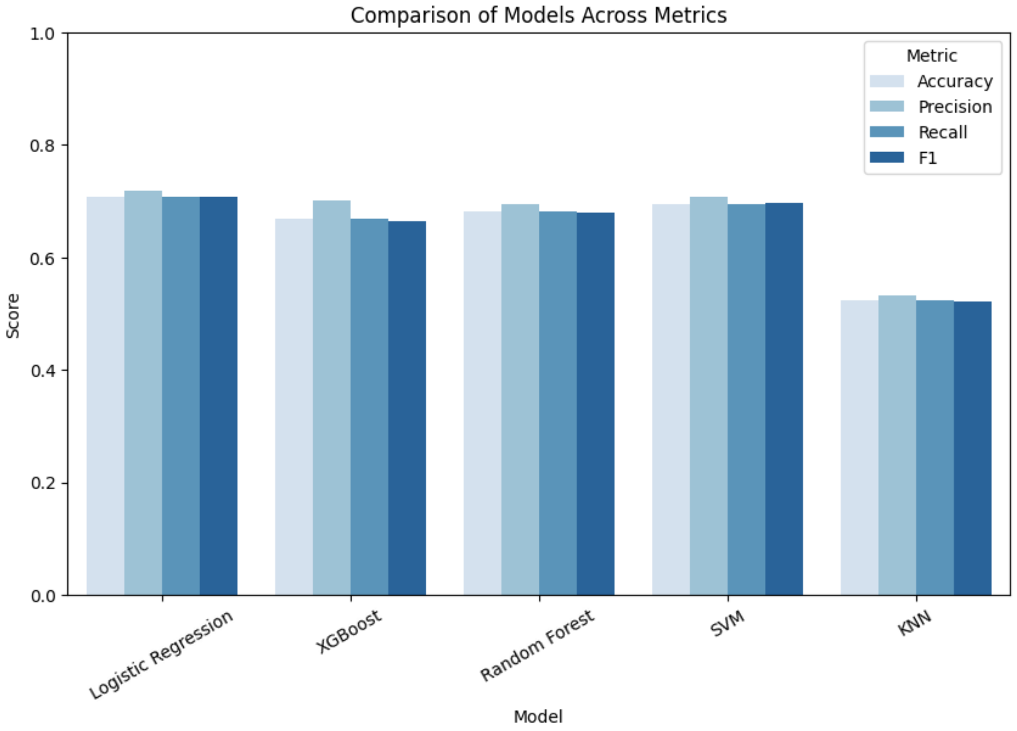
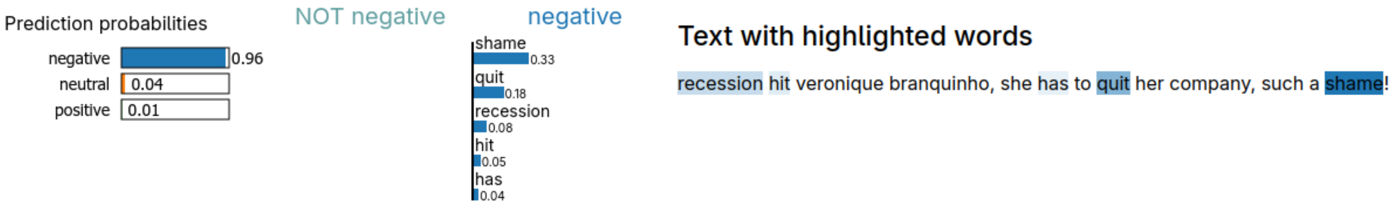


Figure 5: Classification metrics of the ML algorithms

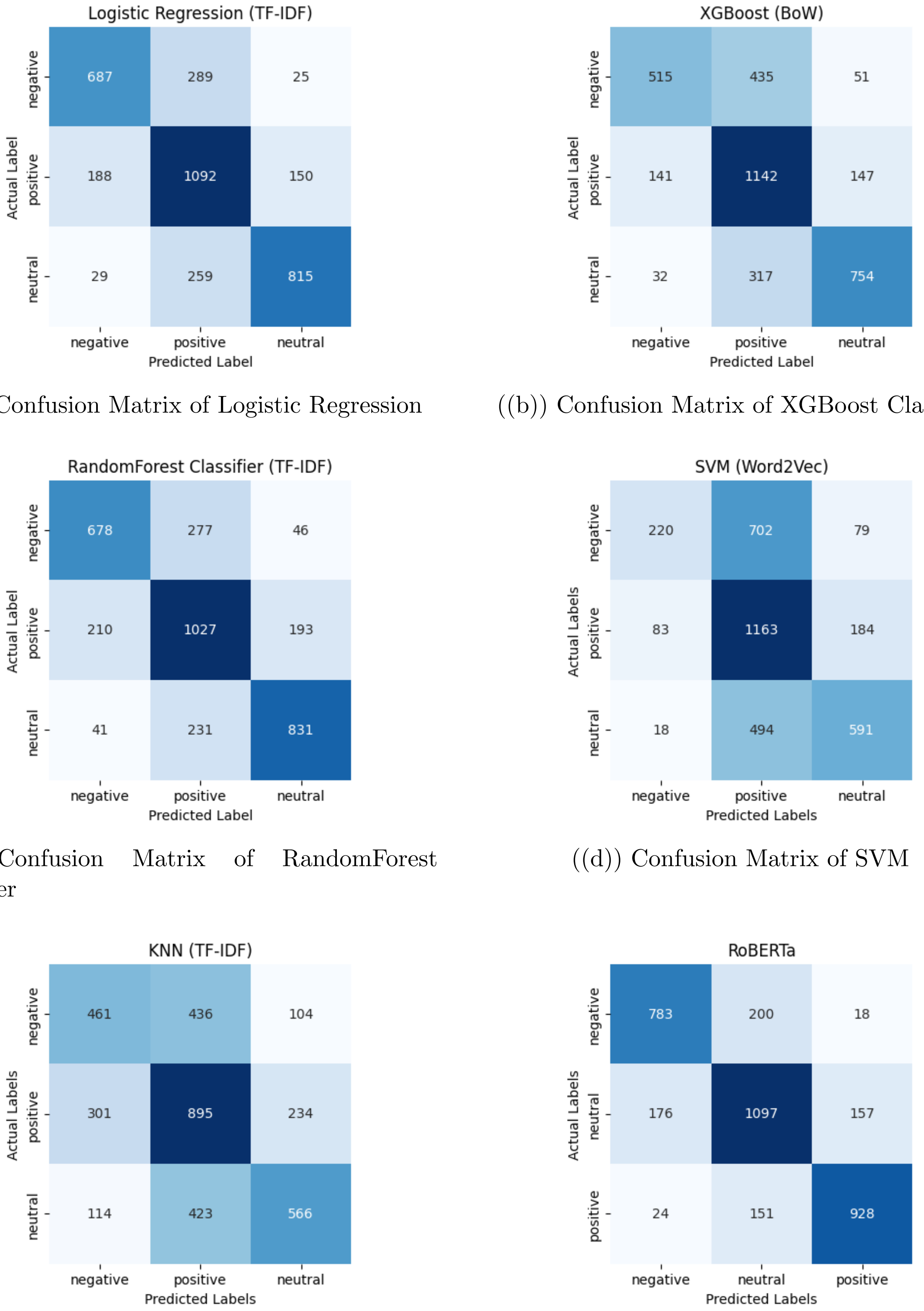
Figure 6 shows a local explanation for an individual prediction by DistilBERT using the LIME (Local Interpretable Model-agnostic Explanations) technique. For the input text, the prediction is negative. As illustrated by both the bar chart and the highlighted text, the word *shame* was the most influential feature driving the negative prediction, followed by other semantically relevant words like *quit* and *recession*.

Figure 7 presents the confusion matrices for the six evaluated models, offering a detailed visualization of their classification performance across the negative, positive, and neutral classes. A qualitative analysis shows that the RoBERTA model 7(f) achieves the most effective class separation. This is evidenced by the high concentration of predictions along the main diagonal—with 783 true negatives, 1097 true positives, and 928 true neutrals—and comparatively lower values in the off-diagonal cells, indicating fewer misclassifications.



recession hit veronique branquinho, she has to quit her company, such a shame!

Figure 6: LIME Explanation



((a)) Confusion Matrix of Logistic Regression

((b)) Confusion Matrix of XGBoost Classifier

((c)) Confusion Matrix of RandomForest Classifier

((d)) Confusion Matrix of SVM

((e)) Confusion Matrix of KNN

((f)) Confusion Matrix of RoBERTa

Figure 7: Comparison of six confusion matrices across different models.

4 Conclusion

This project successfully conducted a comparative study of traditional machine learning algorithms and modern Transformer-based models for the task of sentiment analysis. The experiments, performed on a dataset of social media tweets, provided clear insights into the trade-offs between model complexity, computational cost, and predictive performance.

- **Transformer Superiority**: The Transformer-based models, **RoBERTa** and **DistilBERT**, significantly outperformed all traditional machine learning methods.RoBERTa emerged as the top-performing model with an accuracy of **79.77%**.
- **The Efficiency Trade-off**: This higher accuracy comes at a considerable computational cost, as the Transformer models are significantly more complex, with DistilBERT having 67M and RoBERTa having 125M parameters.
- **Best Traditional Model**: Among the classical algorithms, **Logistic Regression** proved to be the most effective, achieving a respectable accuracy of **72.30%** and an F1-score of 0.72.
- **Effectiveness of TF-IDF**: The TF-IDF feature engineering technique was the most effective meth, as compared to Bag of words and Word2Vec. This is attributed to the fact that Tf-Idf weighs terms by their contextual importance, providing a richer and more informative feature set than simpler frequency-based or embedding-based methods.
- **Limitations of KNN**: The **K-Nearest Neighbors (KNN)** algorithm was the least suitable for this task, yielding the lowest accuracy of just **56.37%**. This is attributed to its reliance on distance metrics, which become ineffective in the high-dimensional vector spaces used to represent text, a phenomenon known as the 'Curse of Dimensionality'.

In conclusion, this study validates that while Transformer architectures provide state-of-the-art performance for sentiment analysis, they require substantial computational resources. For applications where efficiency is paramount, well-tuned classical models like Logistic Regression can still serve as a strong and effective baseline. The choice of model is therefore dependent on the specific requirements of the application, balancing the need for maximum accuracy against the constraints of computational cost and inference speed.

References

[1] Collobert, Ronan, et al. "Natural language processing (almost) from scratch." Journal of machine learning research 12.7 (2011).

[2] Cole, David. "The Chinese room argument." (2004).

[3] Schaller, Robert R. "Moore's law: past, present and future." IEEE spectrum 34.6 (2002): 52-59.

[4] Chomsky, Noam. Aspects of the Theory of Syntax. No. 11. MIT press, 2014.

[5] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." nature 521.7553 (2015): 436-444.

[6] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

[7] Minaee, Shervin, et al. "Large language models: A survey." arXiv preprint arXiv:2402.06196 (2024).

[8] Wankhade, Mayur, Annavarapu Chandra Sekhara Rao, and Chaitanya Kulkarni. "A survey on sentiment analysis methods, applications, and challenges." Artificial Intelligence Review 55.7 (2022): 5731-5780.

[9] Bertsimas, Dimitris, and Angela King. "Logistic regression: From art to science." Statistical Science (2017): 367-384.

[10] Guo, Gongde, et al. "KNN model-based approach in classification." OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". Berlin, Heidelberg: Springer Berlin Heidelberg, 2003.

[11] Biau, Gérard, and Erwan Scornet. "A random forest guided tour." Test 25.2 (2016): 197-227.

[12] Fletcher, Tristan. "Support vector machines explained." Tutorial paper 1118 (2009): 1-19.

[13] Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016.

[14] Sanh, Victor, et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." arXiv preprint arXiv:1910.01108 (2019).

[15] Liu, Yinhan, et al. "Roberta: A robustly optimized bert pretraining approach." arXiv preprint arXiv:1907.11692 (2019).

[16] Maharana, Kiran, Surajit Mondal, and Bhushankumar Nemade. "A review: Data preprocessing and data augmentation techniques." Global Transitions Proceedings 3.1 (2022): 91-99.

[17] Ramos, Juan. "Using tf-idf to determine word relevance in document queries." Proceedings of the first instructional conference on machine learning. Vol. 242. No. 1. 2003.

[18] Köppen, Mario. "The curse of dimensionality." 5th online world conference on soft computing in industrial applications (WSC5). Vol. 1. 2000.

[19] Zhang, Yin, Rong Jin, and Zhi-Hua Zhou. "Understanding bag-of-words model: a statistical framework." International journal of machine learning and cybernetics 1.1 (2010): 43-52.

[20] Church, Kenneth Ward. "Word2Vec." Natural Language Engineering 23.1 (2017): 155-162.