Nina Cunha

Layla Nassar

Ananya Soleti

Ananya Sripath

BERT LLM vs TF-IDF

Introduction

Using classic machine learning algorithms (e.g. logistic regression, Decision Trees, Gradient Boosting, etc.) we aim to see the effectiveness of these models compared to Natural Language Processing models, specifically BERT. We will use a spam detection data set to use an evaluation of both techniques, assessing the accuracy of spam detection. Spam messages and calls are only becoming more prominent, which is fueled by the rise of Artificial Intelligence (AI) and Large Language Models. These AI models are only able to develop more convincing human speech patterns. With the strides in development, it means that it may be used to target particularly vulnerable populations, for example, the elderly, who may not be privy to what are fake messages that are harmful vs not. Our primary goal is to utilize the same technology that actively causes harm to people but change the use of being able to stop the damage it is causing.

Traditional machine learning models tend to take in messages and represent them numerically through the usage of certain vectorization techniques such as Term Frequency-Inverse Document Frequency, or TF-IDF. This allows the machine learning model to achieve a binary output from these messages to classify them as spam or not spam. However, this approach is typically used for short messages and easy classification. By using the BERT Natural Language Processor, the model can gain a more comprehensive understanding of the text. Since BERT is a bi-directional encoder, BERT is able to understand the semantic meaning of the query. BERT has higher capabilities than those of traditional TF-IDF models in understanding text messages because it leverages the bi-directional nature with its large neural network to develop a substantial understanding of the human language. To research both methodologies, we conducted an exploratory data analysis of the SMS Spam Collection Dataset to assess the comparison in performances between traditional machine learning models and natural language processing and artificial intelligence techniques.

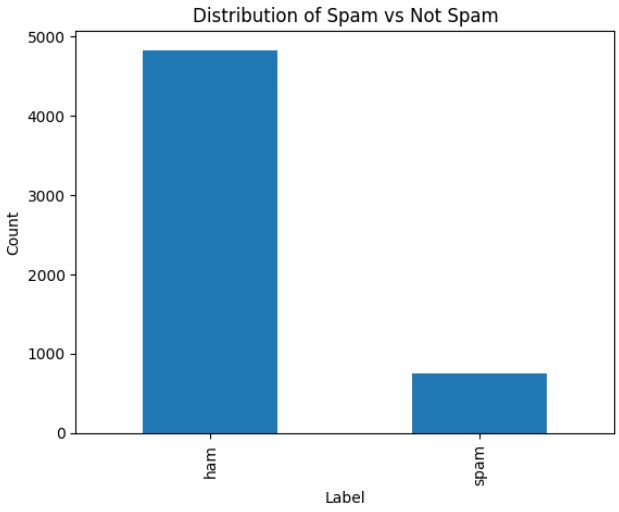
Problem Statement

How do traditional TF-IDF vector representations compare to modern BERT embeddings in improving machine learning models for spam detection?

Methodology

This dataset was taken from a Kaggle dataset which has 2 main columns, one containing the message and one containing “ham” for not spam and “spam” for a spam message. This dataset contains 5572 different messages, with each message having an average length of 80 characters. In the exploration of the dataset, it was determined to be extremely unbalanced, with most of the data being not spam messages. With the dataset being so unbalanced, there may be issues with misleading accuracy scores and a bias towards the majority class. To combat this,

stratification was applied to the train test split, which reduced bias in the training and testing sets. A word cloud was also generated to gather a general sense of the most common words and the distribution of words. Furthermore, correlation matrices were calculated to see if there were any major collinearity problems.



Modeling Techniques

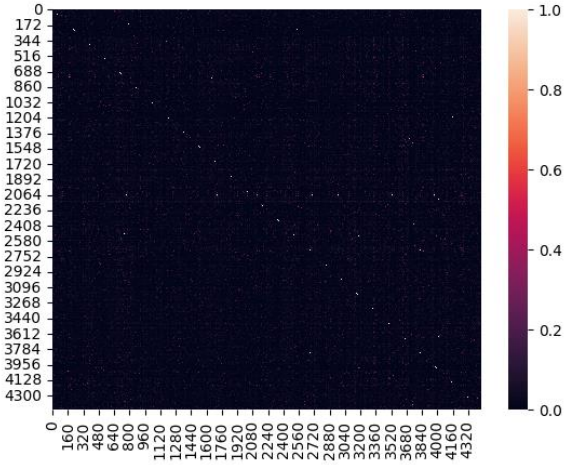
Four main models were chosen to evaluate the performance of TF-IDF with modern Natural Language Processing embeddings using the DistilBERT model, which is a lightweight faster variant of BERT. The chosen models, Logistic Regression, Decision Trees, Gradient Boosting and XGBoost, represent a progression in modeling complexity. To ensure each model operated under its best configuration, GridSearch was applied to each modeling technique to systematically find the most optimal parameters. It was optimized for precision rather than accuracy to deal with the data imbalance.

Logistic Regression was chosen to act as a base model for comparison within model types. This is because it is widely used in binary classification tasks. It allows for easy interpretability and to act as a baseline, and it requires much less computational resources compared to the other models that will be used. Due to the fact that Logistic Regression is mainly used for linear models, Decision Trees were chosen to address nonlinear patterns in the data. Especially any that may arise due to the transformation from text to vector or embedding. However, single Decision Trees have the capacity to have high variance and bias. Therefore, we also used Gradient Boosting which attempts to eliminate bias through using multiple weaker Decision Trees to build a strong model. Each tree is sequentially built upon fixing the errors of previous trees. Lastly, XGBoost, or eXtreme Gradient Boosting was chosen to improve the faults of Gradient Boosting. Gradient Boosting can fall victim to complex datasets, while XGBoost includes regularization techniques making it more robust, especially for complex datasets.

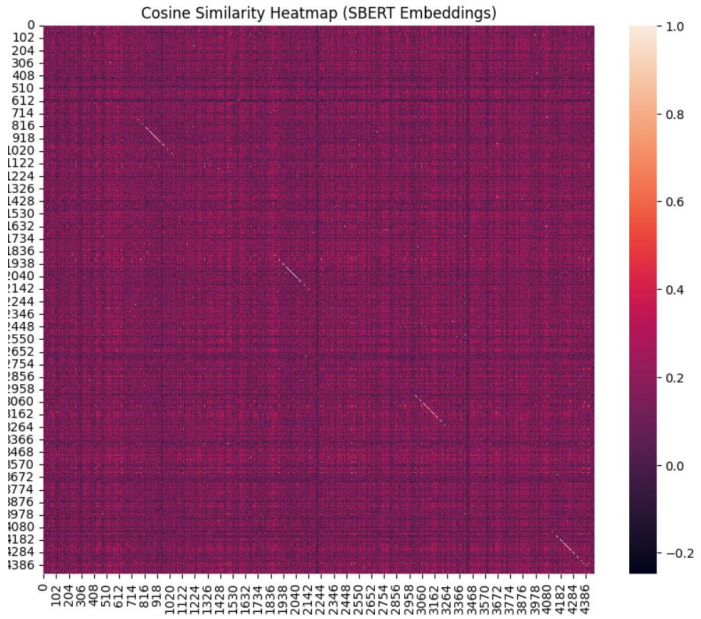
Preprocessing

When cleaning the dataset to use it in our models, the labels needed to be renamed and mapped to numerical values. Messages that were not spam were labeled as 0 and spam messages labeled as 1. The textual information was converted to all lowercases to ensure consistency along with removing any non-alphanumeric values. Then, in the train test split, a test size of 20% was used, which is a standard test size used in many cases. However, the main difference is that the labels were stratified again to reduce the bias in the train and test sets because of the extreme imbalance in our dataset.

We used a TF-IDF vectorizer with an n-gram range of (1, 2) to capture both single-word tokens and common two-word spam phrases such as ‘free entry’ or ‘claim now,’ which improved the model’s ability to detect short contextual patterns frequently found in spam messages. To visualize the resulting vectors, heatmap shows cosine similarity between the TF-IDF vectors of the emails. Dark areas represent low similarity, meaning most emails have quite different content. The diagonal is bright because each document is perfectly similar to itself. Off-diagonal yellow or red spots indicate pairs of emails that are unusually similar, likely duplicates or near-duplicate spam templates. As you can see, many of the resulting values from the TF-IDF vectors are not related, and this will not cause any collinearity issues within the models.



For the BERT-based embedding approach, each message was tokenized using the DistilBERT tokenizer, padded to the maximum sequence length, then the padded tokens were masked to ignore the padded tokens. The DistilBERT encoder then generated the embeddings for each message. The embeddings allow the models to take advantage of BERT’s bi-directional semantic understanding of the messages. Also to measure the similarity of the embeddings, their cosine score was calculated and represented in the following heatmap. Most messages showed low to moderate similarity, which is expected since BERT captures meaning rather than exact wording; however, groups of spam messages exhibited higher similarity due to shared intent or phrasing patterns.

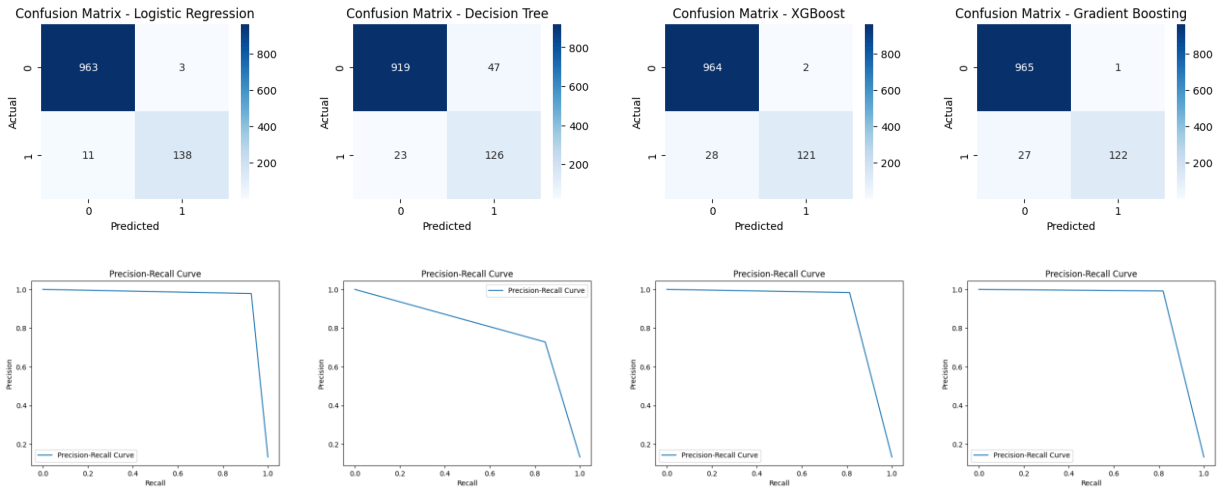


Results

In this study, we evaluated four different supervised machine learning models (Logistic Regression, Decision Tree, Gradient Boosting, and XGBoost) across two feature representation methods: TF-IDF and BERT embeddings. The objective of this evaluation was to study how classical sparse vectorization would compare to more contextualized embeddings regarding classification precision, recall, F1-score, accuracy, and interpretability. Results are reported using confusion matrices, heatmaps, model summaries, and Precision-Recall curves.

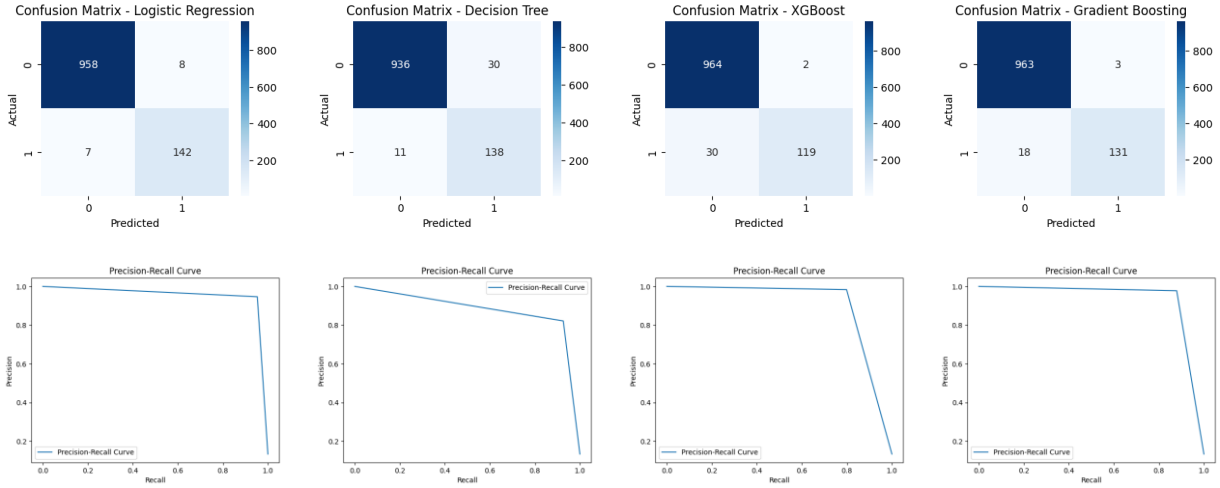
TF-IDF

Logistic Regression | Decision Tree | XGBoost | Gradient Boosting

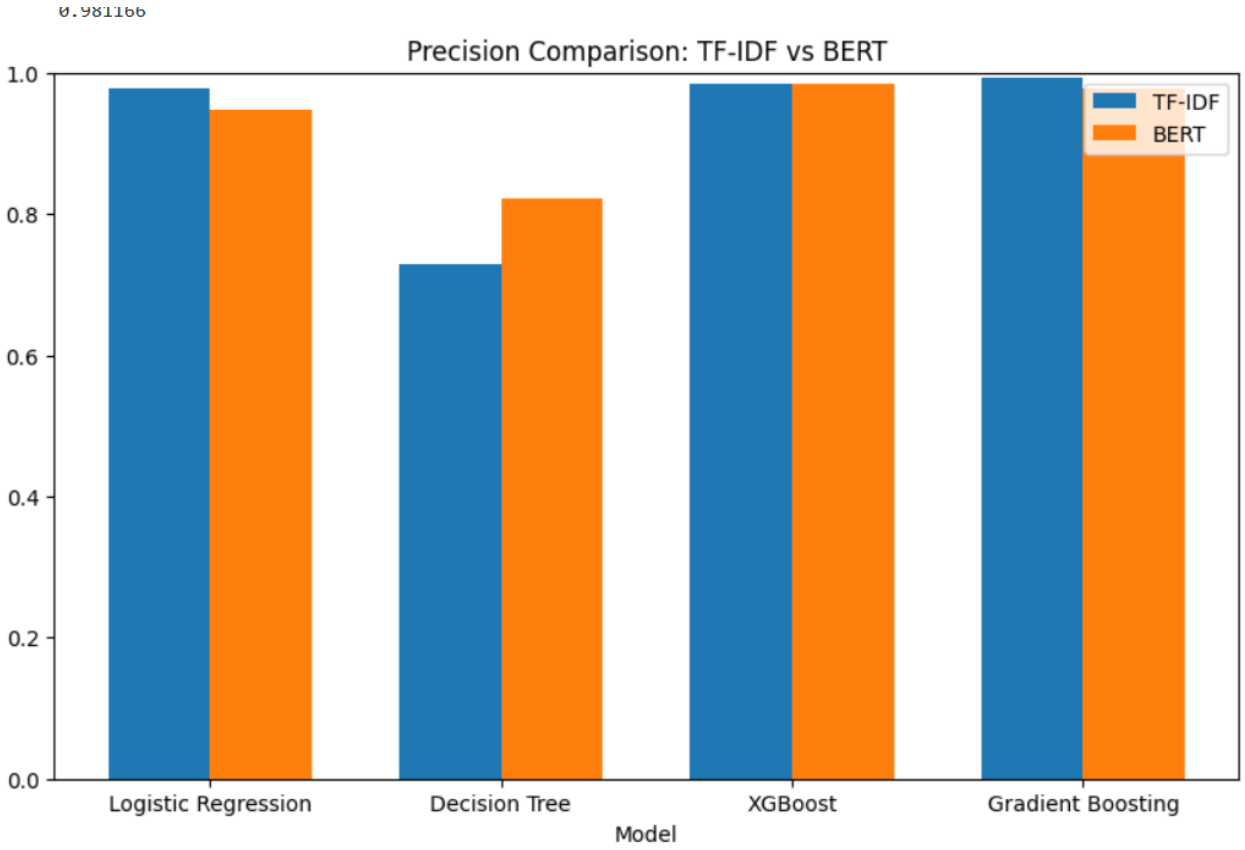


BERT

Logistic Regression	Decision Tree	XGBoost	Gradient Boosting



The confusion matrices revealed that most of the models correctly identified most of the messages, only with a small number of false positives and false negatives. The precision recall curves were also evaluated to ensure a well-rounded understanding of the results, due to the imbalanced dataset. When optimizing the hyperparameters, it was optimized using the metric of precision, illustrating high precision values across all models.



The bar chart with the precision comparison between the machine learning models across TF-IDF (blue) and BERT (orange) shows that TF-IDF came out with higher precision scores for three out of the four models. For example, the model Gradient Boosting had the highest TF-IDF precision score at 0.992, closely followed by the model XGBoost at a score of 0.984, which showcases that TF-IDF is a strong embedding strategy to reduce false positives. Logistic Regression also maintained strong precision with TF-IDF. The precision analysis bar chart shows that TF-IDF has higher scores for spam detection focused on precision, while BERT is stronger for the Decision Tree model.

However, it is much clearer in the bar chart that a single decision tree did not perform as well as the other models in this exploration. This could be due to single decision trees including too much noise from the training set and overfitting the model, or since we did have a heavily imbalanced dataset, it was hard for the model to generalize the data well.

Conclusion

In the evaluation of different machine learning models across TF-IDF and BERT, the results show that the choice of feature representation does not have a large factor in influencing the behavior of the models in the context of our dataset. This contrasts what might be assumed,

where understanding the words and contexts of words would allow for easier detection of spam messages. The assumptions as to why BERT did not outperform TF-IDF could be that much of the spam messages in the data set used shorter sparse and choppy wording, which may be hard for BERT to gather the full semantic meaning if there is little to no meaning at all. This is where TF-IDF would thrive as the messages were short and keyword heavy. The short messages do not allow BERT to leverage its contextual strengths. Another reason that the results did not come out as expected is BERT was not finetuned; rather, the base pre-trained BERT was used to calculate semantically meaningful embeddings.

While our models performed very strongly across both embedding strategies, there are some ways we can improve this work in the future. For example, we can evaluate performance on larger, more diverse datasets. Our dataset was strong and effective for testing, but it was also heavily imbalanced. This may have impacted performance metrics despite our best use of techniques to mitigate the imbalancing. Doing the train test split and model evaluation on a more diverse dataset of ham and spam messages can lead to more robust training of the models, especially those such as Logistic Regression and Decision Tree. Moreover, finetuning BERT paired with using the full BERT model, rather than DistilBERT may result in a more meaningful difference between it and TF-IDF results.

This exploratory research and analysis significantly broadened our current knowledge in how text is represented and evaluated using various machine learning models. Our models brought up some interesting results that would be fascinating to further research using the improvements preciously noted. While the improvement between embedding strategies may not yet justify full application, our findings highlight how classical and modern approaches can complement each other to address the growing challenge of AI-driven spam.

Bibliography

[1]

GeeksforGeeks, “Toxic Comment Classification using BERT,” GeeksforGeeks, Jul. 31, 2023. https://www.geeksforgeeks.org/machine-learning/toxic-comment-classification-using-bert/

[2]

“Google Colab,” Google.com, 2019. https://colab.research.google.com/github/jalammar/jalammar.github.io/blob/master/notebooks/bert/A_Visual_Notebook_to_Using_BERT_for_the_First_Time.ipynb#scrollTo=39UVjAV56PJz (accessed Dec. 09, 2025).

[3]

GeeksforGeeks, “Sentiment Classification Using BERT,” GeeksforGeeks, Aug. 31, 2020. https://www.geeksforgeeks.org/nlp/sentiment-classification-using-bert/

[4]

A. Simha, “Understanding TF-IDF for Machine Learning,” Capital One, Oct. 07, 2021. https://www.capitalone.com/tech/machine-learning/understanding-tf-idf/

[5]

GeeksforGeeks, “Top 6 Machine Learning Classification Algorithms,” GeeksforGeeks, Feb. 26, 2024. https://www.geeksforgeeks.org/machine-learning/top-6-machine-learning-algorithms-for-classification/

[6]

GeeksforGeeks, “Text Classification using Logistic Regression,” GeeksforGeeks, Mar. 04, 2024. https://www.geeksforgeeks.org/machine-learning/text-classification-using-logistic-regression/