Comparative Analysis of Text Classification Algorithms

Beatriz Ribeiro Borges[1], **Elaine Ribeiro Faria**[1], **Paulo H. R. Gabriel**[1]

[1]Faculdade de Computação – Universidade Federal Uberlândia (UFU)
38408-100 – Uberlândia – MG – Brazil

{biarborges,elaine,phrg}@ufu.br

Abstract. *This study presents a comparative analysis of the performance of the text classification task with Transformer-based models (BERT/BERTimbau) in contrast to traditional machine learning algorithms (Decision Tree, XGBoost, Naive Bayes, SVM, MLP) using two textual representations: dense embeddings and TF-IDF. The evaluation was conducted on 5 datasets, 3 binary and 2 multiclass, with texts in Portuguese and English. While Transformers consistently delivered the best overall performance, TF-IDF proved highly competitive– outperforming embeddings and even matching or surpassing BERT in specific cases.*

1. Introduction

Text mining is a subfield of data mining focused on analyzing unstructured textual data, unlike classical data mining which deals with structured formats [Weiss et al. 2015]. A key task in this area is classification, where machine learning algorithms are used to assign textual items to predefined categories. This process plays a fundamental role in natural language processing (NLP), enabling the automatic organization of large volumes of text data into meaningful classes [Hassan et al. 2022].

Traditional text classification methods, that include algorithms such as Naive Bayes, k-Nearest Neighbors, Decision Tree, Support Vector Machine (SVM), typically require extensive preprocessing and the transformation of text into numerical vectors, such as those based on TF-IDF. In contrast, Transformer-based models like BERT (Bidirectional Encoder Representations from Transformers) have gained prominence for their ability to learn deep, bidirectional contextual representations directly from raw text. Also, BERT can be used both as a feature extractor and as a classifier, offering enhanced performance in various text classification tasks due to its capacity to capture semantic relationships in context.

Several studies have shown that Transformer-based models outperform traditional classifiers in text classification tasks, especially in English-language corpora and binary classification problems [Hassan et al. 2022]. While studies such as [Plath et al. 2022, Souto Moreira et al. 2023, Braz Junior and Fileto 2021] have evaluated the performance of multilingual BERT and BERTimbau in Portuguese, few works have conducted a systematic comparison between traditional and Transformer-based models across both binary and multiclass tasks, using different datasets and representation strategies.

Conducting studies in the area of text classification is crucial in order to develop strategies for intelligently and automatically classifying important information

[de Magalhães et al. 2019]. Thus, this study aims to compare the performance of classical and modern text classification algorithms, including BERT, the Portuguese-adapted BERTimbau, Decision Tree, XGBoost, Naive Bayes, SVM, and neural networks such as Multi-Layer Perceptron (MLP). The evaluation spans different datasets and considers two preprocessing and representation techniques, such as TF-IDF and contextual embeddings from BERT. The goal is to identify which methods offer the best performance across these scenarios.

2. Related Work

Recent studies have demonstrated the relevance of applying machine learning and Transformer-based models to Portuguese-language text classification. For instance, [Plath et al. 2022] explored the use of multilingual BERT alongside traditional algorithms like SVM and Naive Bayes to detect hate speech, highlighting the impact of preprocessing and showing that SVM with TF-IDF achieved the best performance, although it reached only an F1-score of 0.57. In contrast, [Souto Moreira et al. 2023] found that BERTimbau, a model pre-trained specifically for Portuguese, outperformed both traditional models and multilingual BERT in fake news detection.

Further contributions include [Braz Junior and Fileto 2021], who evaluated BERTimbau and multilingual BERT in textual coherence tasks, revealing that performance varies with text type–multilingual BERT excelled on forum data (99% accuracy), while BERTimbau performed better on journalistic texts (97%). Additionally, [Sun et al. 2019] examined aspect-based sentiment analysis using English datasets, showing that BERT surpassed models like LSTM and Memory Networks, reaching up to 85.5% accuracy.

Together, these studies underline the strong performance of Transformer models in NLP tasks, while also showing that traditional approaches may remain effective depending on the task and preprocessing strategies. This reinforces the need for comparative analyses across models and datasets to better understand their behaviors and limitations in varied scenarios.

3. Proposed Method for Comparing Text Classifiers

This section details the method used for the comparative analysis. Figure 1 illustrates the steps that were followed for the classification task using the Knowledge Discovery in Databases (KDD) process.

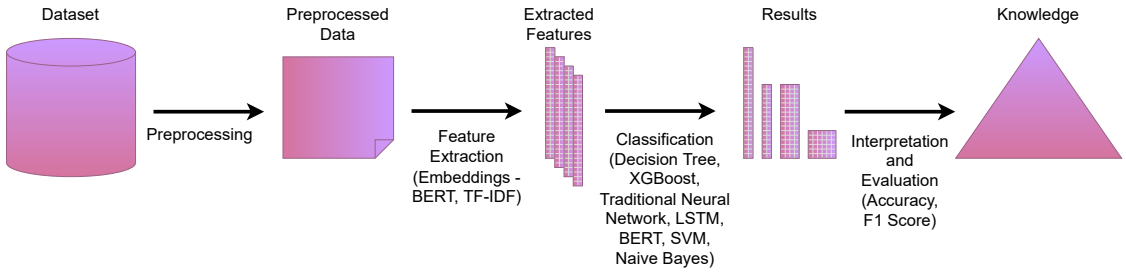

Figura 1. Overview of the steps that make up the KDD process – Source: Adapted from [Fayyad et al. 1996].

3.1. Databases

To ensure comparability, different datasets were selected. Each dataset is described in the following paragraphs, and Table 1 summarizes the class distributions across all datasets.

OlistKaggle: The OlistKaggle[1] dataset, available publicly on the Kaggle platform, contains user reviews in Portuguese extracted from various e-commerce websites and was specifically designed for sentiment analysis tasks in Brazilian Portuguese. Among the available review files in the compressed set, the file related to the Olist website was selected, as it presents a binary classification task and is the least imbalanced option among those available. This dataset has undergone initial preprocessing, in which the strings were converted to lowercase and the alphabet was standardized to English by removing diacritical marks. The target variable, named "polarity", was derived from star ratings: reviews with 4 and 5 stars were considered positive, while reviews with 1 and 2 stars were considered negative; reviews with 3 stars were discarded. The complete dataset contains 41,744 records; however, after removing entries with missing values in the "polarity" variable, the total used in this study amounts to 38,079 records, 11,408 negatives and 26,671 positives.

Fake.br and Fake.br Multi: The Fake.br[2] corpus was initially developed by the *Núcleo Interinstitucional de Linguística Computacional* (NILC-USP) for the detection of fake news, as cited in [Souto Moreira et al. 2023]. The dataset consists of news articles previously classified as either true or fake. In total, the dataset contains 7,200 news items, with 3,600 labeled as true and 3,600 as fake. The articles are categorized into six different topics: Politics (4180 texts), TV/Celebrities (1544 texts), Society and Daily Life (1276 texts), Science and Technology (112 texts), Economy (44 texts) and Religion (44 texts). This study uses all 7,200 articles for the binary classification task between true and fake news. Additionally, it also explores the labels available to perform a multiclass classification task.

SentiHood: SentiHood[3] is a dataset created to help identify the polarity of opinions on aspects such as safety, price, and location in urban neighborhoods. This dataset was used in the study by [Sun et al. 2019], which aims to perform sentiment analysis using a pretrained BERT model. The classification is binary, divided into positive and negative, with a total of 5,215 sentences originally. After removing the missing values, a total of 3,632 instances were used in this study, with 2,344 labeled as positive and 1,288 as negative.

NewsKaggle: NewsKaggle[4], available on the Kaggle platform, is a dataset that provides a comprehensive collection of news articles covering various domains, including business, technology, sports, education, and entertainment. The data is sourced from the news magazine "The Indian Express´´, written in English. This dataset was created for applications in Natural Language Processing and Machine Learning. It contains a total of 10,000 news articles, evenly distributed with 2,000 articles per category.

[1] https://bit.ly/4jNb4gN
[2] https://github.com/roneysco/Fake.br-Corpus/tree/master
[3] https://github.com/uclnlp/jack/tree/master/data/sentihood
[4] https://bit.ly/4iZzcLP

Tabela 1. Summary of class distributions across all datasets

Dataset	Language	Task Type	Classes	Instances
OlistKaggle	Portuguese	Sentiment Analysis (Binary)	Positive / Negative	26,671 / 11,408
Fake.br	Portuguese	Fake News Detection (Binary)	True / Fake	3,600 / 3,600
Fake.br Multi	Portuguese	News Classification (Multiclass)	Politics / TV&Celebrities / Society / Sci&Tech / Economy / Religion	4,180 / 1,544 / 1,276 / 112 / 44 / 44
SentiHood	English	Sentiment Analysis (Binary)	Positive / Negative	2,344 / 1,288
NewsKaggle	English	News Classification (Multiclass)	Business / Technology / Sports / Education / Entertainment	2,000 each

3.2. Preprocessing and Feature Extraction

Two separate preprocessing pipelines were implemented: one for feature extraction using term frequency-inverse document frequency (TF-IDF), and another for generating embeddings using BERT. To prepare the text data for TF-IDF vectorization, a preprocessing pipeline was implemented using Python and NLTK[5]: (1) lowercasing all text; (2) removing punctuation and special characters, while preserving letters with accents when appropriate; (3) normalizing whitespace; (4) tokenization using language-specific rules from NLTK's `word_tokenize`; (5) removal of stopwords based on the respective language (Portuguese or English); and (6) stemming using the Snowball stemmer for each language. This preprocessing ensures consistent text representation across both binary and multiclass classification datasets in Portuguese and English. The `TfidfVectorizer`[6] from the scikit-learn library was then applied to generate a TF-IDF matrix.

For the feature extraction using BERT embeddings, in contrast to the TF-IDF pipeline, this preprocessing retained punctuation and focused only on cleaning unwanted special characters and normalizing whitespace, as BERT models are pretrained on natural language with punctuation. After preprocessing, the texts were passed through a pretrained BERT model to generate dense vector representations. Specifically, the model used was BERTimbau's neuralmind/bert-base-portuguese-cased[7] for texts in Brazilian Portuguese, and BERT's bert-base-multilingual-cased[8] for English datasets. Each text was tokenized and encoded using the corresponding tokenizer, and the resulting embeddings were extracted from the [CLS] token of the final hidden layer. The [CLS] token is a special token added at the beginning of each textual input passed to the BERT model. Its final hidden state is commonly used as a contextual summary of the entire sequence [Souza 2020]. These embeddings were stored for use as input features in the classification models.

To address the high dimensionality of the TF-IDF feature space, a dimensionality reduction step was applied prior to model training. Specifically, we employed the `SelectKBest` method with the chi-squared (χ^2) statistical test to retain only the most informative features with respect to the target categories. Three values of

[5]`https://www.nltk.org/`

[6]`https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html`

[7]`https://huggingface.co/neuralmind/bert-base-portuguese-cased`

[8]`https://huggingface.co/google-bert/bert-base-cased`

$k \in \{1000, 2000, 3000\}$ were evaluated through 3-fold cross-validation, and the value that yielded the highest weighted F1-score was selected. In this subject, for the BERT-based algorithms, TF-IDF was not used as a representation method. BERT models operate on raw text inputs and leverage contextual embeddings generated by transformer architectures, which inherently capture semantic and syntactic information from the input sequences.

3.3. Classification Algorithms

For the classification tasks, the following algorithms were used: Decision Tree, XGBoost, Naive Bayes, SVM, BERT (for English datasets), BERTimbau (for Portuguese datasets), and MLP using both BERT embeddings and TF-IDF representations. These algorithms were selected not only due to their use and effectiveness in previous text classification studies as discussed in Section 2, but also because they represent diverse paradigms: probabilistic, tree-based, margin-based, neural networks, and Transformer-based. Traditional models were implemented with the `scikit-learn` library, while BERT-based approaches relied on the `Transformers` library. The complete implementation is available on GitHub[9].

Datasets were split into 85% for training and validation and 15% for testing, with `random_state=42` to ensure reproducibility. Hyperparameter tuning was performed via grid search with 5-fold cross-validation on the training/validation split, using the weighted F1-score as the evaluation metric to account for class imbalance.

Decision Tree: The Decision Tree algorithm constructs a hierarchical structure of decision rules based on feature values, partitioning the input space into regions with homogeneous class labels [Quinlan 1986]. The hyperparameter optimization explored different configurations for maximum tree depth (`max_depth` = 10, 15, 20), the minimum number of samples required to split an internal node (`min_samples_split` = 2, 5, 10), and the minimum number of samples required to be at a leaf node (`min_samples_leaf` = 1, 2, 4). The model was trained with class balancing enabled (`class_weight='balanced'`) and a fixed random seed (`random_state=42`) to ensure consistent results, as with all the other algorithms.

XGBoost: XGBoost is an ensemble method based on gradient boosting that combines multiple weak learners–typically decision trees–into a strong classifier [Chen and Guestrin 2016]. The search space included values for maximum tree depth (`max_depth` = 3, 5, 7), learning rate (`learning_rate` = 0.01, 0.1, 0.2), number of estimators (`n_estimators` = 50, 100, 150), and subsampling ratio for training instances (`subsample` = 0.8, 1.0). For binary classification tasks, the model was configured with `objective='binary:logistic'` and `eval_metric='logloss'`, while for multiclass scenarios, the configuration used `objective='multi softprob'`.

Naive Bayes: Naive Bayes is a probabilistic classifier based on Bayes' theorem, assuming independence among features [McCallum and Nigam 1998]. Unlike the other models, no hyperparameter tuning via grid search was applied to Naive Bayes, as it generally performs well with default settings and offers limited tunable parameters. Two variants were employed depending on the input representation: the `GaussianNB` was

[9] `https://anonymous.4open.science/r/BertTFIDFAlgorithmsClassification-0879`

used for BERT embeddings, as it is designed to handle continuous input features; and the `MultinomialNB` was used for TF-IDF vectors, which are non-negative and typically integer-like, aligning with the assumptions of the multinomial distribution.

SVM: SVM constructs a hyperplane that maximally separates data points of different classes in the feature space [Cortes and Vapnik 1995]. For the experiments, the LinearSVC implementation was used. The hyperparameter search space included variations in the regularization parameter (`C` = 0.01, 0.1, 1, 10, 100), the loss function (`loss` = 'hinge' or 'squared_hinge'), and the tolerance for the stopping criteria (`tol` = 1e-3, 1e-4, 1e-5). The model was configured with class balancing (`class_weight='balanced'`) and a maximum of 5000 iterations (`max_iter=5000`) to ensure convergence.

MLP: MLP is a class of feedforward neural networks composed of fully connected layers with nonlinear activation functions, enabling the learning of complex decision boundaries [Bishop 1995]. The search space included variations in the number and size of hidden layers (`hidden_layer_sizes` = [(100,), (100, 50)]), activation functions (`activation` = ['identity', 'logistic', 'tanh', 'relu']), optimization solvers (`solver` = ['lbfgs', 'sgd', 'adam']), L2 regularization term (`alpha` = [0.0001, 0.001]), and learning rate schedules (`learning_rate` = ['constant', 'invscaling', 'adaptive'], where this parameter is only relevant when `solver='sgd'`). The model was configured with early stopping to prevent overfitting (`early_stopping=True`), a maximum of 500 iterations (`max_iter=500`).

BERT: BERT is a pretrained language model based on the Transformer architecture that captures bidirectional contextual information and can be fine-tuned for NLP tasks [Devlin et al. 2019]. For fine-tuning, the pretrained model `bert-base-cased` and tokenizer were used. Hyperparameter optimization was performed using Optuna over a search space including learning rate (1e-5 to 5e-5), batch size (4 or 8), gradient accumulation steps (4, 8, or 16), number of training epochs (2 to 4), and weight decay (0 to 0.1). The evaluation strategy was set to perform validation at each epoch, with no checkpoint saving during the search phase. After 10 trials, the best hyperparameters found were used to retrain the model, saving checkpoints every epoch and loading the best model at the end for final evaluation.

BERTimbau: BERTimbau is a Portuguese-language adaptation of BERT, pretrained on large Brazilian Portuguese corpora [Souza et al. 2020]. The fine-tuning configuration was the same as for BERT, except that the tokenizer and model were replaced with `neuralmind/bert-base-portuguese-cased` to better suit the Portuguese language.

3.4. Methods and Evaluation Metrics

After training with the best hyperparameters selected via Optuna, the models were evaluated on a separate test set comprising 15% of the data. The evaluation metrics used were accuracy, F1-score, and the confusion matrix, with implementation carried out using the `scikit-learn` library.

The weighted F1-score (`average='weighted'`) was adopted, as it is appropriate for imbalanced datasets. It computes the F1-score for each class individually and produces a weighted average based on the class frequencies. In balanced datasets, this

weighted average closely approximates the macro average, since all classes contribute equally to the final score.

4. Results and Discussions

This section presents the accuracy (Table 2) and F1-score (Table 3) results for the six algorithms evaluated on each dataset using both TF-IDF and embedding representations. For each dataset, the highest score across all models and feature representations is shown in bold in the tables.

Tabela 2. Accuracy results for all datasets using different classification algorithms with TF-IDF and BERT-based embeddings.

Accuracy						
Model	**Representation**	**Olist Kaggle**	**Fake.br**	**Fake.br Multi**	**Senti Hood**	**News Kaggle**
Decision Tree	**TF-IDF**	0.8859	0.9120	0.5278	0.7486	0.9240
	Embedding	0.8778	0.8407	0.5426	0.6422	0.7660
XGBoost	**TF-IDF**	0.9237	0.9667	0.6778	0.8000	0.9787
	Embedding	0.9338	0.9713	0.6991	0.7835	0.9500
Naive Bayes	**TF-IDF**	0.9065	0.6389	0.6046	0.7633	0.9800
	Embedding	0.8601	0.8935	0.6241	0.7156	0.6393
SVM	**TF-IDF**	0.9261	0.9620	0.6806	0.8550	0.9793
	Embedding	0.9398	0.9731	0.6565	0.7633	0.9693
MLP	**TF-IDF**	0.9249	0.9584	**0.7130**	0.8367	0.9793
	Embedding	0.9426	0.9759	0.6843	0.7541	0.9680
BERT-based	**Embedding**	**0.9501**	**0.9926**	0.7028	**0.9009**	**0.9887**

Tabela 3. F1-Score results for all datasets using different classification algorithms with TF-IDF and BERT-based embeddings.

F1-Score						
Model	**Representation**	**Olist Kaggle**	**Fake.br**	**Fake.br Multi**	**Senti Hood**	**News Kaggle**
Decision Tree	**TF-IDF**	0.8856	0.9118	0.5416	0.7108	0.9242
	Embedding	0.8797	0.8406	0.5480	0.6462	0.7664
XGBoost	**TF-IDF**	0.9234	0.9666	0.6542	0.7892	0.9786
	Embedding	0.9341	0.9713	0.6770	0.7736	0.9502
Naive Bayes	**TF-IDF**	0.9061	0.5860	0.4805	0.7362	0.9800
	Embedding	0.8636	0.8934	0.6448	0.7200	0.6433
SVM	**TF-IDF**	0.9272	0.9619	0.6795	0.8525	0.9792
	Embedding	0.9403	0.9730	0.6659	0.7665	0.9694
MLP	**TF-IDF**	0.9247	0.9583	**0.7130**	0.8294	0.9793
	Embedding	0.9430	0.9758	0.6558	0.7512	0.9681
BERT-based	**Embedding**	**0.9500**	**0.9925**	0.6856	**0.9004**	**0.9886**

Across nearly all datasets, the BERT-based model using contextual embeddings consistently achieved the highest accuracy and F1-scores. For instance, on the Fake.br dataset, it reached 99.26% accuracy and a 99.25% F1-score, outperforming all traditional classifiers. These results highlight the effectiveness of the BERT architecture for classification tasks that require nuanced language understanding, such as fake news detection and sentiment analysis.

Despite being a simpler and more static representation, TF-IDF combined with models such as SVM and MLP often achieved competitive results, even surpassing those obtained using embeddings. For example, on the Fake.br Multi dataset, MLP with TF-IDF achieved an accuracy of 71.3%, outperforming the same model using embeddings (68.43%). Notably, Decision Tree with TF-IDF outperformed its embedding-based counterpart in most datasets. For XGBoost, embeddings achieved slightly better results overall, but TF-IDF remained competitive with very close values. A similar trend was observed with both SVM and MLP, indicating that TF-IDF still offers a robust baseline.

Naive Bayes and Decision Tree consistently underperformed compared to the other models. On the Fake.br Multi dataset, for example, Naive Bayes with TF-IDF achieved an F1-score of only 48.05%. Decision Tree showed only marginal improvements, with F1-scores of 54.16% for TF-IDF. These results suggest that both models struggle particularly in multiclass settings, especially when dealing with sparse or high-dimensional feature representations.

Among traditional algorithms, MLP consistently achieved the best performance across most datasets and representations. For example, in the OlistKaggle dataset, MLP with embeddings reached an F1-score of 94.3%, nearly matching the performance of the BERT-based model (95.0%). These findings suggest that well-tuned traditional classifiers, especially MLP and SVM, remain strong alternatives to Transformer-based models, particularly in scenarios with limited computational resources.

The News Kaggle dataset, which features balanced binary labels, led to high scores across all models. In contrast, the Fake.br Multi dataset, likely more challenging due to its multiclass structure and class imbalance, had the lowest scores overall, particularly for simpler models like Naive Bayes and Decision Tree.

To further investigate the results on the Fake.br Multi dataset, which consistently posed the greatest classification challenge, Figure 2 presents the confusion matrix for BERTimbau, the best-performing model among those using embeddings. Figure 3 shows the confusion matrix for MLP with TF-IDF, which not only outperformed all other TF-IDF-based models but also achieved the highest overall performance. In contrast, Figure 4 presents the confusion matrix for Naive Bayes with TF-IDF, which recorded the lowest F1-score among all models and feature types.

The confusion matrix for the BERTimbau model shows relatively strong performance for the “Politics” class, but significant challenges in distinguishing “Economy”, “Science/Technology” and “Religion”, as well as considerable confusion between “Society” and “TV/Celebrities”. The “Politics” category had 550 documents correctly classified, indicating solid performance. However, 24 “Politics” documents were misclassified as “Society”, and 53 as “TV/Celebrities”. The “Economy” and “Religion” classes, although very small, had all documents incorrectly classified. The “Science/Technology”

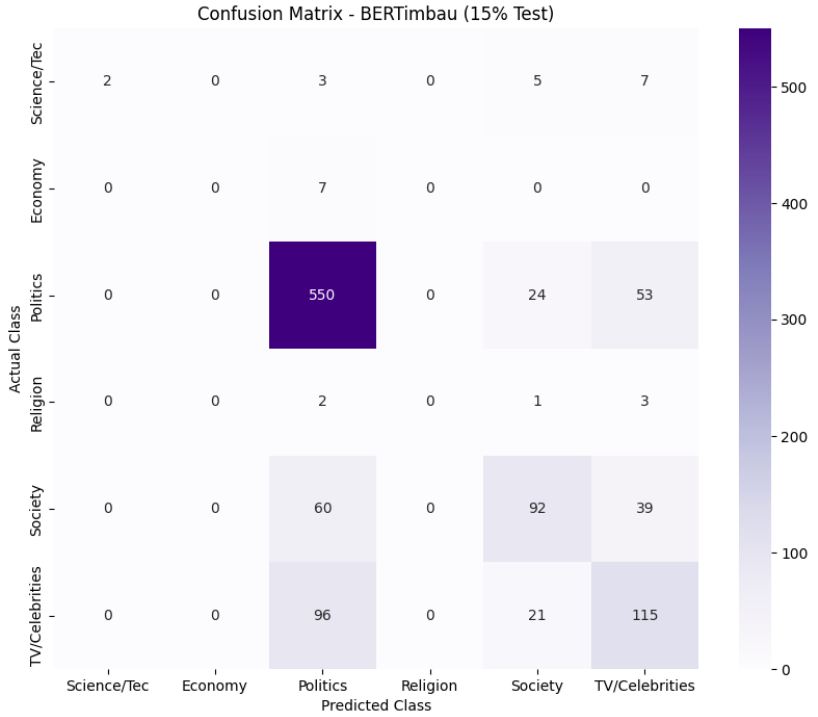

Figura 2. Confusion matrix of BERTimbau on the Fake.br Multi dataset – Source: Created by the authors (2025).

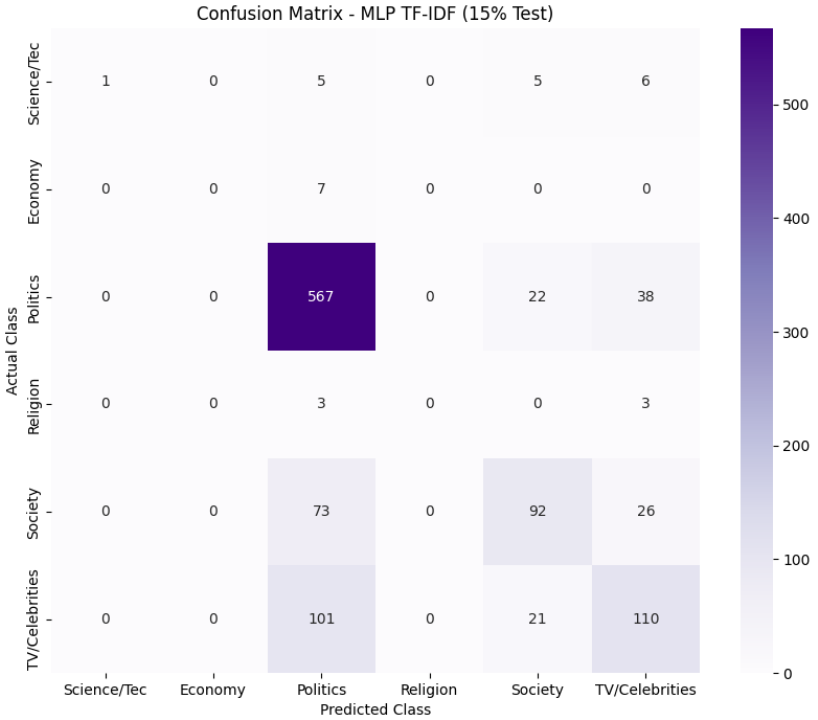

Figura 3. Confusion matrix of MLP with TF-IDF on the Fake.br Multi dataset – Source: Created by the authors (2025).

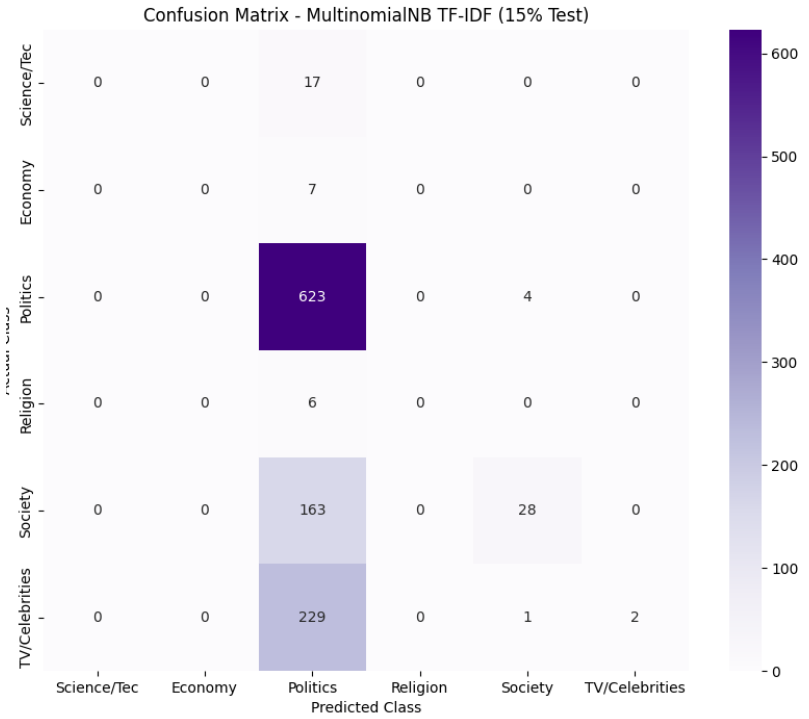

Figura 4. Confusion matrix of Naive Bayes with TF-IDF on the Fake.br Multi dataset – Source: Created by the authors (2025).

category had only 2 correct predictions, with the rest misclassified as "Politics" (3), "Society" (5), and "TV/Celebrities" (7), highlighting the model's difficulty in distinguishing this class. The "Society" class had 92 correct predictions, but 60 were confused with "Politics" and 39 with "TV/Celebrities". Finally, "TV/Celebrities" achieved 115 correct classifications, though 96 documents from this class were wrongly labeled as "Politics" and 21 as "Society".

In the MLP model with TF-IDF, the results are similar, with "Politics" again being the best-performing class. There were 567 correct predictions for "Politics", slightly outperforming BERTimbau. However, 22 real "Politics" documents were misclassified as "Society" and 38 as "TV/Celebrities", showing a slight improvement over BERTimbau regarding misclassification. The "Economy" and "Religion" classes once again had all incorrect classifications. "Science/Technology" had only 1 correct prediction, with errors distributed to "Politics" (5), "Society" (5), and "TV/Celebrities" (6), reaffirming the pattern of BERTimbau. The "Society" class maintained 92 correct predictions but saw an increase in confusion, with 73 real documents classified as "Politics" and 26 as "TV/Celebrities", indicating slightly weaker performance than BERTimbau. "TV/Celebrities" had 110 correct predictions, with 101 documents misclassified as "Politics" and 21 as "Society".

The Naive Bayes model with TF-IDF displayed a strong bias toward the "Politics" class. This model achieved 623 correct classifications for "Politics", the best among the three. Only 4 documents from this class were misclassified as "Society". However, this strong performance came at the cost of errors in classification for all other classes. All documents from "Economy", "Science/Technology", and "Religion" were misclassified.

The “Society” class had only 28 correct predictions, with 163 documents wrongly labeled as “Politics”. Similarly, “TV/Celebrities” had just 2 correct classifications, with 229 documents misclassified as “Politics”.

When comparing the three models, “Politics” is consistently the best recognized class. In contrast, “Science/Technology”, “Economy”, and “Religion” are weak points for all three models, with very low or no correct classifications, due to the limited amount of data available for these categories. “Society” and “TV/Celebrities” are frequently confused with each other and with “Politics”, resulting in many classification errors. This confusion could be especially true in cases where local celebrities appear in political news or publicly express political opinions, blurring the line among these categories. Notably, although the Naive Bayes model performs poorly across most categories, it exhibits a remarkably low error rate for the “Politics” class. This behavior suggests a pronounced bias toward the dominant class. Such bias can be explained by the model’s probabilistic framework and its assumption of feature independence which, when combined with data imbalance, tends to favor the majority class disproportionately [Forman 2003].

5. Conclusion

This study compared traditional algorithms (Decision Tree, XGBoost, Naive Bayes, SVM, MLP) and Transformer-based models (BERT, BERTimbau) across five text classification datasets in Portuguese and English, using TF-IDF and embeddings as input representations. In the experiments, Transformer models consistently outperformed others, demonstrating their ability to capture complex semantic nuances. However, SVM and MLP proved to be robust alternatives when the computational demands of Transformers are a limiting factor. Embeddings had great results, but TF-IDF was competitive, especially in SentiHood and NewsKaggle, sometimes outperforming embeddings and approaching BERT’s performance. Nonetheless, Transformer-based models steadily achieve the best results.

Multiclass classification tasks were more challenging, as seen by the performance drop in Fake.br Multi (BERTimbau F1-Score: 68.56%) compared to binary Fake.br (BERTimbau F1-Score: 99.25%), likely due to class imbalance and semantic overlap. In contrast, NewsKaggle (multiclass, English) achieved high performance, suggesting that well-defined balanced classes ease classification. The results indicate that language (Portuguese or English) is not a key factor; task nature, class structure, balance, and vocabulary matter more. Choosing between TF-IDF and embeddings for traditional models should be based on empirical evaluation of each dataset.

Future research could investigate more recent or specialized Transformer architectures to benchmark against BERT and BERTimbau, also incorporating advanced vectorization approaches such as those based on LLaMA or GPT architectures. Additionally, exploring data augmentation or synthetic data generation techniques could help address class imbalance, particularly in challenging multiclass classification tasks such as Fake.br.

Referências

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford university press.

Braz Junior, O. and Fileto, R. (2021). Investigando coerência em postagens de um fórum de dúvidas em ambiente virtual de aprendizagem com o BERT. In *Anais do XXXII*

Simpósio Brasileiro de Informática na Educação, pages 749–759, Porto Alegre, RS, Brasil. SBC.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 785–794.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

de Magalhães, L. H., Matos, F. F., and Souza, R. R. (2019). Comparação entre algoritmos de classificação aplicados na predição de notícias de jornais on-line. In *XX Encontro Nacional de Pesquisa em Ciência da Informação*, Florianópolis. ENANCIB.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. *NAACL-HLT*.

Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI Magazine*, 17(3):37.

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3(Mar):1289–1305.

Hassan, S. U., Ahamed, J., and Ahmad, K. (2022). Analytics of machine learning-based algorithms for text classification. *Sustainable Operations and Computers*, 3:238–248.

McCallum, A. and Nigam, K. (1998). A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization*, volume 752, pages 41–48.

Plath, H. O., Paiva, M. E. O., Pinto, D. L., and Costa, P. D. P. (2022). Detecção de discurso de Ódio contra mulheres em textos em português brasileiro: Construção da base MINA-BR e modelo de classificação. *Revista Eletrônica de Iniciação Científica em Computação*, 20(3).

Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1(1):81–106.

Souto Moreira, L., Machado Lunardi, G., de Oliveira Ribeiro, M., Silva, W., and Paulo Basso, F. (2023). A study of algorithm-based detection of fake news in brazilian election: Is BERT the best. *IEEE Latin America Transactions*, 21(8):897–903.

Souza, F., Nogueira, R., and Lotufo, R. (2020). Bertimbau: Pretrained bert models for brazilian portuguese. In *Brazilian Conference on Intelligent Systems (BRACIS)*.

Souza, F. C. d. (2020). BERTimbau: Pretrained BERT models for brazilian portuguese. Master's thesis, Universidade Estadual de Campinas.

Sun, C., Huang, L., and Qiu, X. (2019). Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–385, Minneapolis, Minnesota. Association for Computational Linguistics.

Weiss, S. M., Indurkhya, N., and Zhang, T. (2015). *Fundamentals of Predictive Text Mining*. Springer International Publishing, London, second edition edition.