Automatic Detection of Textual Disinformation in WhatsApp Messages

A technical and concise approach to fake news classification using the FakeWhatsApp.BR_2018.csv corpus.

Corpus Building Strategy

The base dataset is FakeWhatsApp.BR_2018.csv, which focuses on messages from public WhatsApp groups. Strict selection and filtering were crucial to the quality of the training.

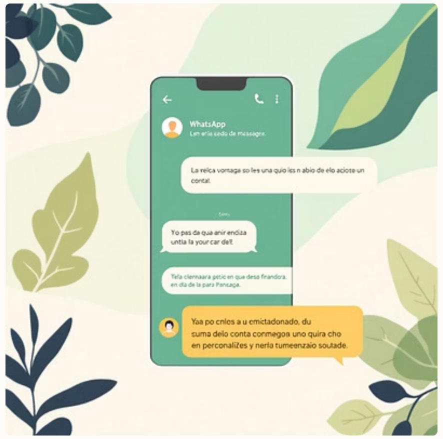
Data Selection

- Filter for viral texts (viral=1).
- Exclusion of non-media content (media=0).
- Elimination of duplicate and unlabeled texts (-1).

Classes Definition

- Class 1 (Fake): Misinformation texts.
- Class 0 (Real): Verified authentic texts.
- Total of 4588 texts after cleaning.

The final texts for analysis (**X**) are the non-media content from the text column, and the labels (**y**) are the binary values from the misinformation column.



Made with GAMMA

Distinctive Characteristics of Texts

Disinformation and real texts exhibit distinct lexical and structural patterns, which serve as clues for automatic detection.

Misinformation (Fake)

- Excessive use of **emojis** and visual repetition.
- **Imperative** language (call to action, "See explicitly").
- Chain structure, **seeking virality** and **urgency**.

Real Texts (Real)

- Similarity to **everyday conversations**.
- More **natural** and sporadic use of emojis.
- **Normalized** use of punctuation and capitalization (easy to identify intuitively).

Corpus Statistics After Filtering

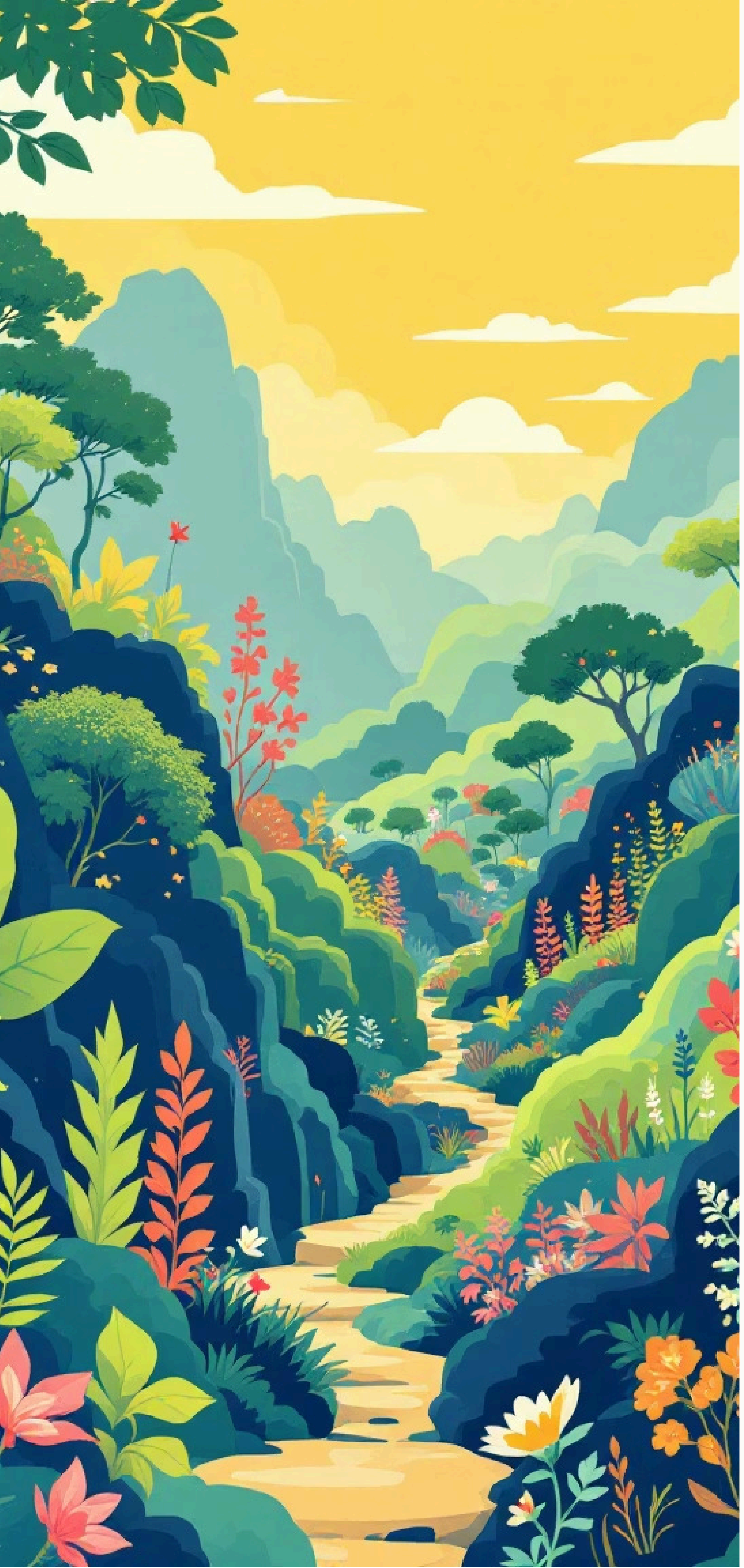
Table 1 summarizes the final corpus size and the averages of textual metrics, essential for understanding the information density in each class.

Metric	Total Count	Characteres Mean	Words Mean	Shares Mean
Total Corpus	4588	574.50	87.25	11.89

Comparative Statistics by Class

Class	N.º Texts	Words Mean	Shares Mean	Characteres Mean
Misinformation (1)	2041	113.87	4.974	719.40
Real (0)	2547	60.044	3.454	408.92

It is observed that misinformation tends to be longer (more words) and to be shared more, reflecting its design for viralization.



Pre-Processing and Feature Engineering

Preprocessing is essential to transform raw text into data that ML models can interpret effectively, focusing on the most informative words.

Text Cleaning

Inserting spaces for **emojis** and **punctuation**, followed by their removal.

Stopwords Removal

Elimination of high-frequency words (pre-defined and manual) that do not add meaning.

Normalization

Stemming words and replacing URLs with their domains (e.g., www.google.com).

Truncation and Filtering

Ignore patterns like kkk+ and keep only the **first 100 words** of each sentence.

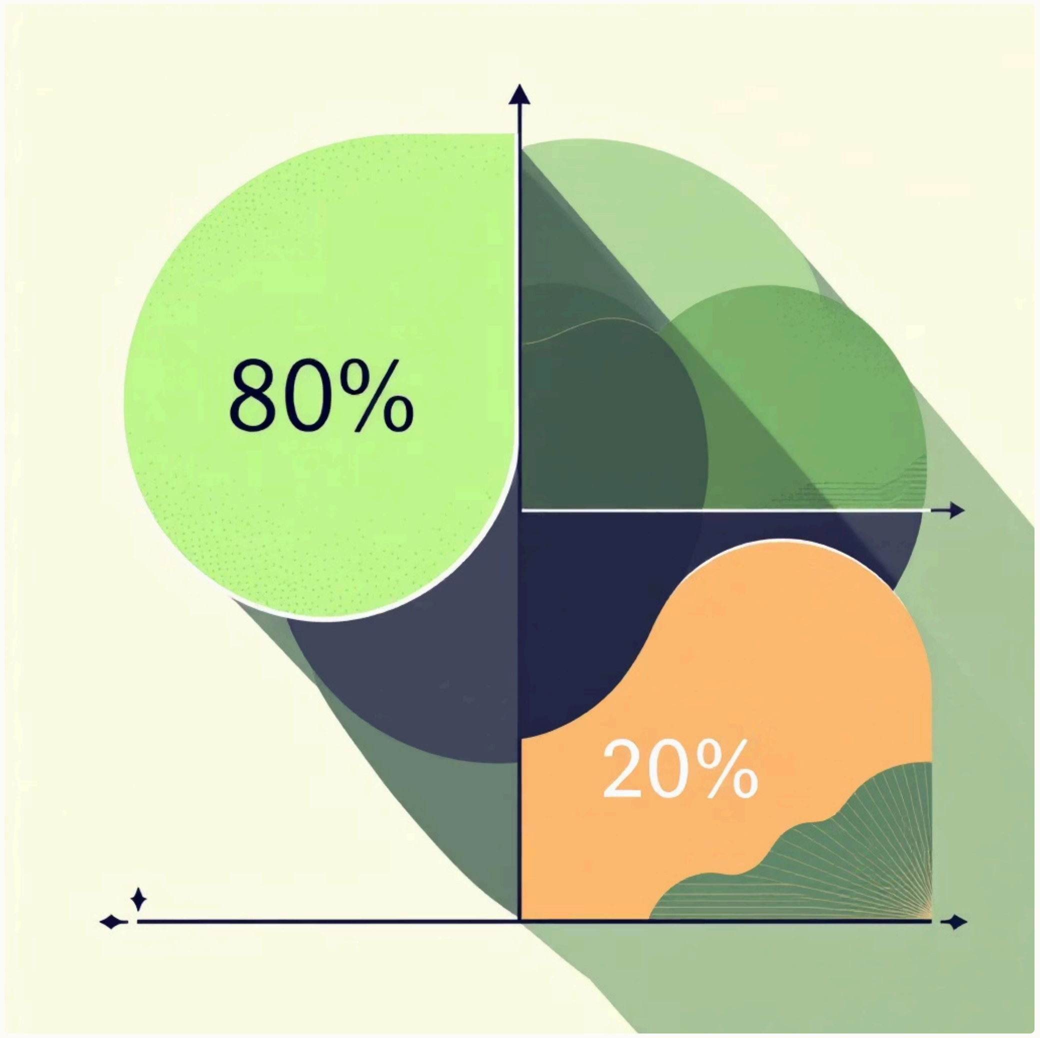
Division and Vectorization Strategy

The dataset division and vectorization technique were defined to ensure robustness and performance of the models.

Dataset Division

80% (Training) / 20% (Test) split with random_state=42 for replicability.

Stratification (stratify=y) was applied to maintain the proportion of classes (Fake/Real) in both sets.



TF-IDF Vectorization

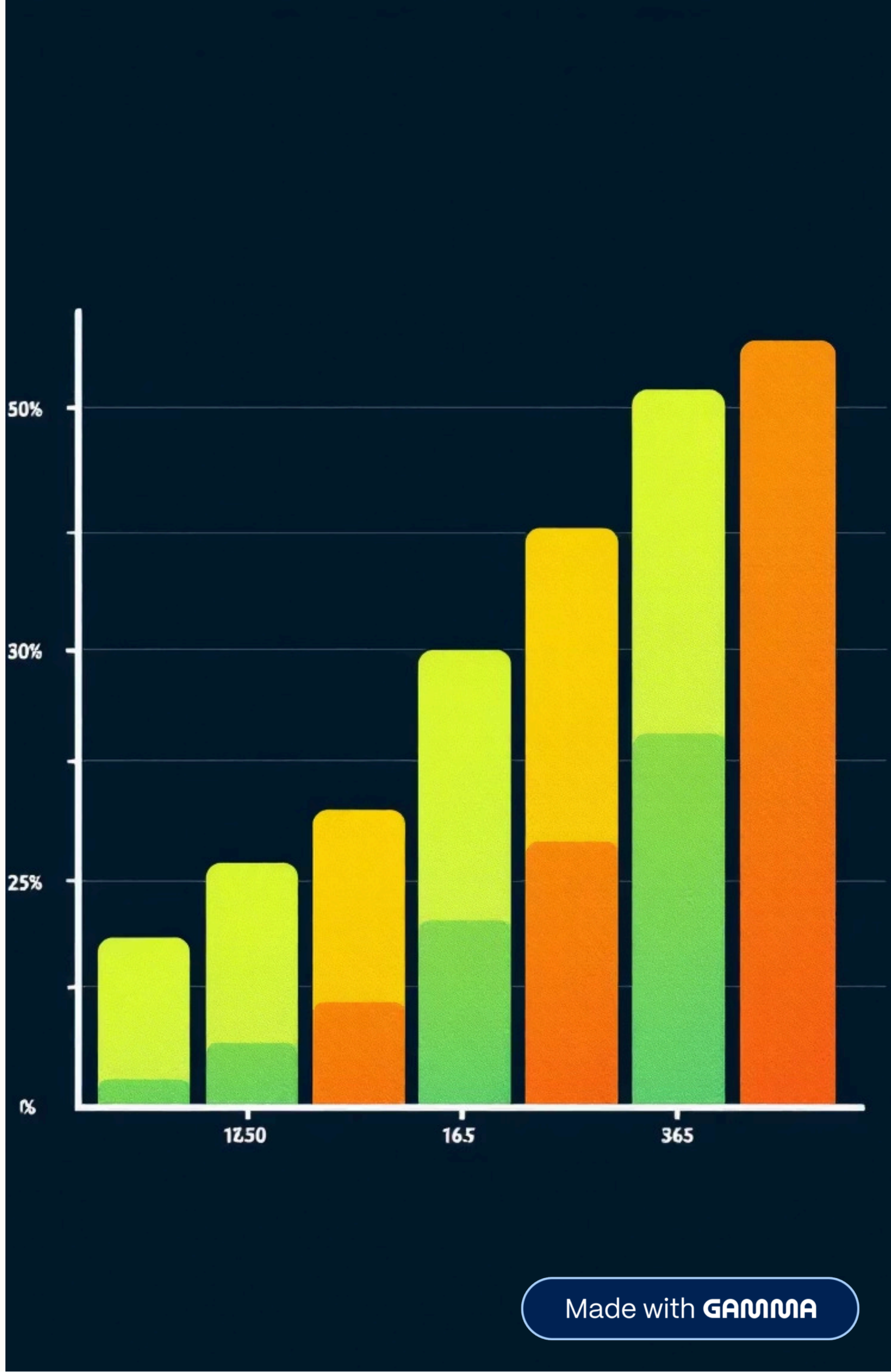
- Using Term Frequency-Inverse Document Frequency (TF-IDF).
- Creating vectors for unigrams, bigrams, and trigrams.
- The vectorizer learns the mapping from the training set only; the test set is transformed using this mapping.

Model Training and Metric Selection

Seven classification models were evaluated, focusing on metrics that prioritize the correct identification of misinformation, avoiding harmful false positives.

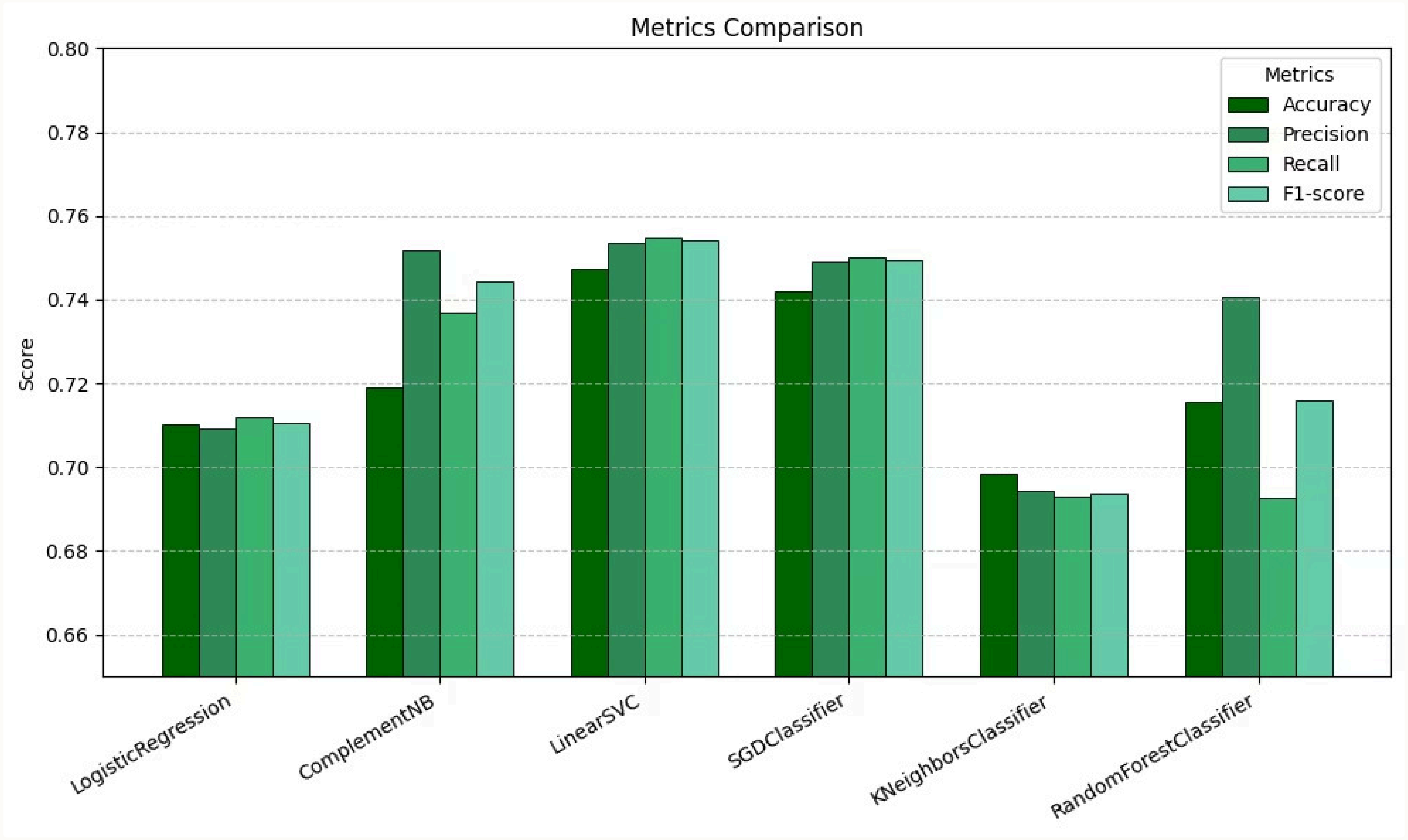
7	1	3
Tested Models	**Max. Recall**	**Best Models**
LR, NB, LSVM, SGD, SVM, KNN, RF.	Priority metric: proportion of correctly identified fake messages.	LSVM, SGD, and NB, with F1-Score as the balance metric.

The **F1-Score** was adopted to reconcile **Recall** (detecting fake news) and **Precision** (minimizing the marking of real messages as false, which is highly harmful).



Made with GAMMA

Metrics Comparison



Experiment 1: Weight Assignment Analysis (LIME)

I used LIME (Local Interpretable Model-agnostic Explanations) to compare the keywords that LSVM, SGD, and NB models use to classify a 200-index sentence.

Analyzed Text(Index 200)

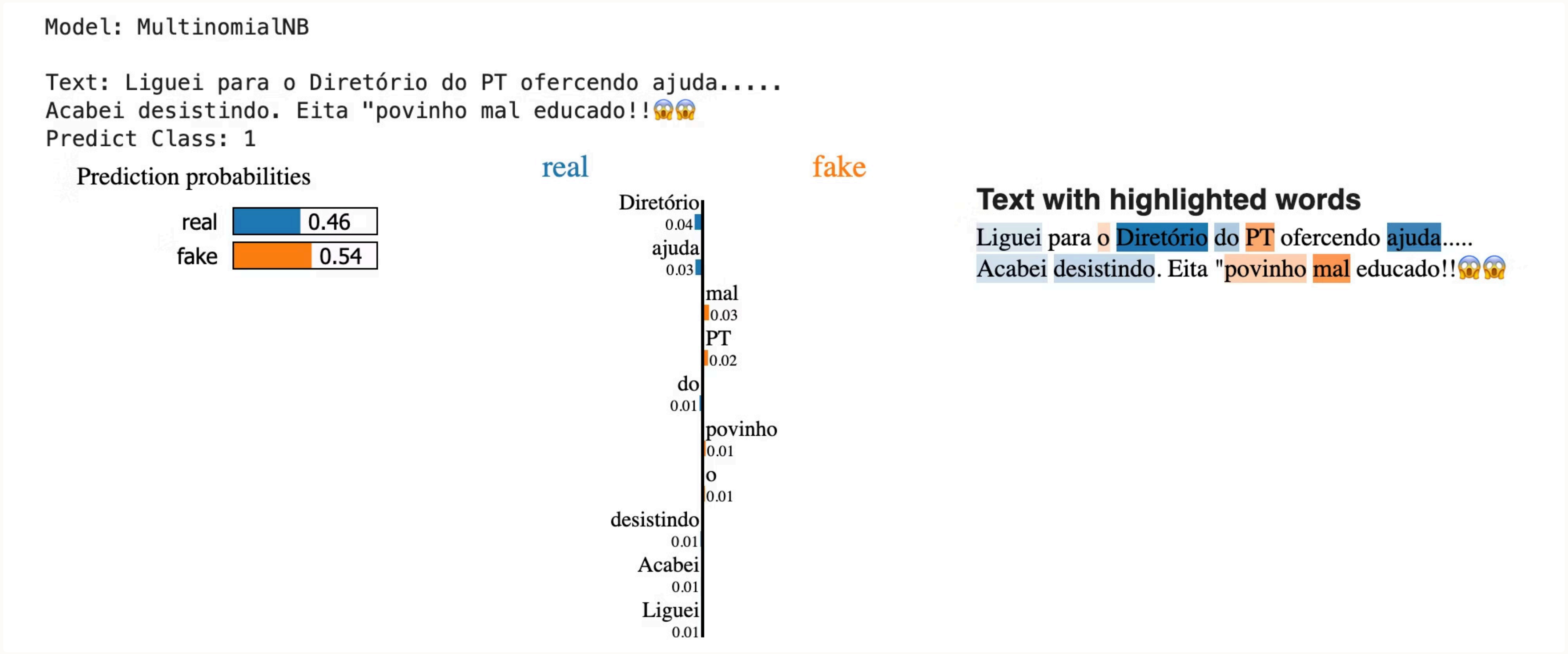
> "Liguei para o **Diretório** do **PT** oferecendo ajuda.... Acabei **desistindo**. Eita "povinho mal educado!!"."

This sentence presents mixed predictions, making it ideal for model consistency analysis. I selected the four words with the highest combined weight for comparison:

Word/Model	MNB	LSVM	SGD
PT	0.02 (Fake - 4ª)	0.10 (Fake - 1ª)	0.14 (Fake - 1ª)
Diretório	0.04 (Real - 1ª)	0.09 (Real - 2ª)	0.11 (Real - 2ª)
do	0.01 (Real - 1ª)	0.03 (Fake - 9ª)	0.04 (Fake - 8ª)
desistindo	0.01 (Real - 8ª)	0.04 (Real - 4ª)	0.04 (Real - 9ª)

The models agree that **PT** increases the probability of being **Fake** and **Diretório** increases the probability of being **Real**, demonstrating consistency. The word "do" has irrelevant weight, as expected for a pseudo-stopword.

LIME output (e.g. MultinomialNB)



```
Model: MultinomialNB

Text: Liguei para o Diretório do PT ofercendo ajuda.....
Acabei desistindo. Eita "povinho mal educado!!😱😱
Predict Class: 1
```

Prediction probabilities

real	0.46
fake	0.54

real		fake
Diretório	0.04	
ajuda	0.03	
	0.03	mal
	0.02	PT
do	0.01	
	0.01	povinho
	0.01	o
desistindo	0.01	
Acabei	0.01	
Liguei	0.01	

Text with highlighted words

Liguei para o Diretório do PT ofercendo ajuda.....
Acabei desistindo. Eita "povinho mal educado!!😱😱

Made with GAMMA

Experiment 2: Global Clue Analysis (LIME)

In this experiment, I use LIME to discover which words contribute most to classifying 300 WhatsApp texts as fake or real, calculating the average and frequency of LIME explanatory words across all models to determine general clues.

The most common words from the top three models were:

- Fake: sentiu/Senhor/suspende/eleitorado/tô
- Real: Maria/padre/Maior/acertou/______

REAL

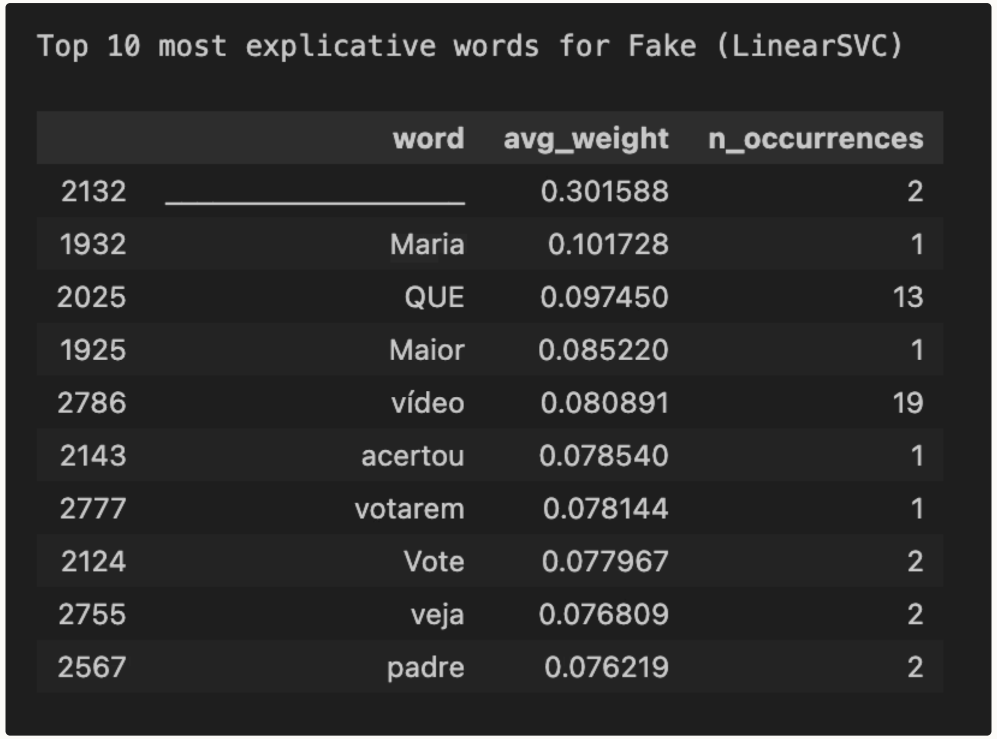
Word	MNB	LSVM	SGD
sentiu	-0.070213 (5ª)	-0.137195 (2ª)	-0.17831 (3ª)
Senhor	-	-0.102820 (6ª)	-0.134738 (8ª)
suspende	-0.070602 (4ª)	-0.122129 (3ª)	-0.183526 (2ª)
eleitorado	-0.074198 (3ª)	-0.147824 (1ª)	-0.197468 (1ª)
tô	-0.066830 (6ª)	-0.099314 (10ª)	-0.125804 (6ª)

FAKE

Word	MNB	LSVM	SGD
Maria	0.065470 (3ª)	0.102127 (2ª)	0.136981 (2ª)
padre	0.064031 (5ª)	0.076601 (10ª)	0.110970 (7ª)
Maior	0.052582 (8ª)	0.086258 (4ª)	0.113512 (5ª)
acertou	0.067527 (2ª)	0.081617 (5ª)	0.112295 (6ª)
______	0.148821 (1ª)	0.300264 (1ª)	0.332011 (1ª)

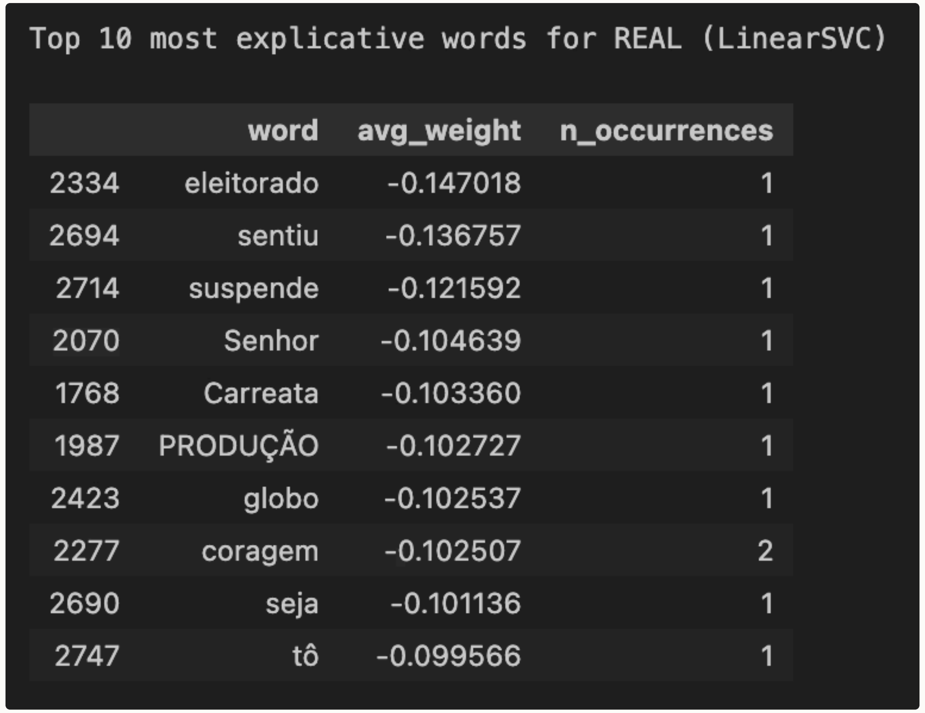
The overall analysis reveals that the models use similar semantic and contextual cues, with the Fake class featuring terms more closely linked to proper nouns or less political themes.

Example (LSVM):



Top 10 most explicative words for Fake (LinearSVC)

	word	avg_weight	n_occurrences
2132	____________________	0.301588	2
1932	Maria	0.101728	1
2025	QUE	0.097450	13
1925	Maior	0.085220	1
2786	vídeo	0.080891	19
2143	acertou	0.078540	1
2777	votarem	0.078144	1
2124	Vote	0.077967	2
2755	veja	0.076809	2
2567	padre	0.076219	2



Top 10 most explicative words for REAL (LinearSVC)

	word	avg_weight	n_occurrences
2334	eleitorado	-0.147018	1
2694	sentiu	-0.136757	1
2714	suspende	-0.121592	1
2070	Senhor	-0.104639	1
1768	Carreata	-0.103360	1
1987	PRODUÇÃO	-0.102727	1
2423	globo	-0.102537	1
2277	coragem	-0.102507	2
2690	seja	-0.101136	1
2747	tô	-0.099566	1

Conclusion and Next Steps

Detecting misinformation requires a balance between the model's ability to capture viral signals (Recall) and accuracy on neutral texts (Precision).

Models Coherence

LSVM, SGD, and NB are the best predictors, showing remarkable similarities in the learned words (LIME cues).

Crucial Clues

Highly politicized or imperative words are strong indicators of misinformation. The average number of words/shares also reinforces this distinction.

Future Improvements

- Apply LIME on examples in which the model fail to classify correctly (fake positives/negative).
- Execute LIME multiple times on the same example and see if the the wieghts are the same.