

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
DEPARTAMENTO ACADÊMICO DE INFORMÁTICA  
ESPECIALIZAÇÃO EM CIÊNCIA DE DADOS E SUAS APLICAÇÕES**

**RODRIGO CARDOSO DURGIEWICZ**

**DETECÇÃO DE FAKE NEWS UTILIZANDO O ALGORITMO NAIVE BAYES**

**CURITIBA**

**2021**

**RODRIGO CARDOSO DURGIEWICZ**

**DETECÇÃO DE FAKE NEWS UTILIZANDO O ALGORITMO NAIVE BAYES**

**Fake news detection applying Naïve Bayes Algorithm**

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Especialista em Ciência de Dados e suas Aplicações, do Departamento Acadêmico de Informática, da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Marcelo de Oliveira Rosa

**CURITIBA**

**2021**



Ministério da Educação  
**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**  
UTFPR - CAMPUS CURITIBA  
DIRETORIA-GERAL - CAMPUS CURITIBA  
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO - CAMPUS CURITIBA  
DEPARTAMENTO DE APOIO DAS ESPECIALIZAÇÕES LATO-SENSU DOS  
CURSOS DE INFORMÁTICA - CAMPUS CURITIBA  
CURSO DE ESPECIALIZAÇÃO EM CIÊNCIA DE DADOS E SUAS APLICAÇÕES



---

## TERMO DE APROVAÇÃO

### DETECÇÃO DE FAKE NEWS UTILIZANDO O ALGORITMO NAIVE BAYES.

por

**Rodrigo Cardoso Durgiewicz**

Este Trabalho de Conclusão de Curso foi apresentado às 20h30min do dia 26 de julho de 2021 por videoconferência como requisito parcial à obtenção do grau de Especialista em Ciência de Dados e suas Aplicações na Universidade Tecnológica Federal do Paraná - UTFPR - Campus Curitiba. O aluno foi arguido pela Banca de Avaliação abaixo assinados. Após deliberação, a Banca de Avaliação considerou o trabalho aprovado.

---

Prof. Dr. Marcelo de Oliveira Rosa (Presidente/Orientador – DAELT- CT/ UTFPR-CT)

---

Profa. Dra. Rita Cristina Galarraga Berardi (Avaliadora 1 – DAINF-CT/ UTFPR-CT)

---

Profa. Dra. Mariana Antonia Aguiar Furucho (Avaliadora 2– DAELT- CT/ UTFPR-CT)

**O Termo de Aprovação assinado encontra-se no sistema SEI- Processo nº 23064.029978/2021-47**

Aos familiares, amigos(as), professores e todos aqueles(as) que cruzaram minha vida, participando de alguma forma na construção e realização deste tão desejado sonho.

## **AGRADECIMENTOS**

À Deus, pelo dom da vida e por possibilitar que meus sonhos sejam realizados.

Aos meus pais pelo amor, incentivo e por confiar sempre que meus objetivos seriam alcançados.

À minha esposa pela confiança e afeto nos momentos de dificuldade.

Aos professores, Marcelo de Oliveira Rosa e Rita Cristina Galarraga Berardi, por sua tranquilidade, dedicação e atenção.

Muito Obrigado!

Se nossas vidas já estão escritas, seria preciso  
um homem corajoso para mudar o roteiro.  
(Alan Wake, 2010).

## RESUMO

*Fake news*, o termo que ganhou visibilidade durante as últimas eleições presidenciais de diversos países, propagam-se rapidamente, tornando-se uma preocupação para a sociedade. O objetivo deste trabalho é obter um modelo de aprendizado de máquina para detecção de notícias falsas utilizando o algoritmo *Naive Bayes*. Este estudo contempla o embasamento teórico, bem como as 4 etapas para o desenvolvimento do modelo: Coleta de Dados, Pré-Processamento, Mineração de Dados e Pós-Processamento. Os resultados foram obtidos através das métricas de acurácia, precisão, *recall* e F1-score. O algoritmo obteve uma acurácia de 84% para a detecção de *fake news*.

**Palavras-chave:** Notícias. *Fake News*. Aprendizado de Máquina. *Naive Bayes*.

## **ABSTRACT**

Fake News, the term that gained visibility during the latest presidential elections in several countries, spreads quickly, becoming a concern for society. The objective of this work is to obtain a machine learning model to detect fake news using the Naive Bayes algorithm. This study contemplates the theoretical basis, as well as the 4 steps for the model development: Data Collection, Pre-Processing, Data Mining and Post-Processing. The results were obtained through the accuracy, precision, recall and F1-Score metrics. The algorithm accuracy to detect fake news was 84%.

**Keywords:** News. Fake News. Machine Learning. Naïve Bayes.

## LISTA DE FIGURAS

Figura 1 – Etapas do Aprendizado de Máquina .....	15
Figura 2 – Amostra do conjunto de dados utilizados.....	21
Figura 3 – Distribuição de notícias falsas e verdadeiras .....	22
Figura 4 – Base de dados após o pré-processamento.....	23
Figura 5 – Matriz Esparsa .....	24
Figura 6 – Aplicação do algoritmo <i>Naive Bayes</i> .....	25
Figura 7 – Notícias previstas corretamente.....	26
Figura 8 – Notícias previstas: falsos negativos .....	27
Figura 9 – Notícias previstas: falsos positivos.....	27

## LISTA DE QUADROS

Quadro 1 – Matriz de Confusão .....	18
Quadro 2 – Matriz de Confusão do algoritmo <i>Naive Bayes</i> .....	26

## LISTA DE TABELAS

Tabela 1 - Desempenho do algoritmo <i>Naive Bayes</i> por classe .....	25
Tabela 2 – Desempenho geral do algoritmo <i>Naive Bayes</i> .....	25

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>12</b>
1.1 OBJETIVO GERAL .....	12
1.2 OBJETIVOS ESPECÍFICOS .....	12
1.3 ORGANIZAÇÃO DO TRABALHO .....	13
<b>2 FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>14</b>
2.1 FAKE NEWS .....	14
2.2 APRENDIZADO DE MÁQUINA .....	14
2.3 ALGORITMOS DE APRENDIZADO DE MÁQUINA .....	16
2.3.1 Naive Bayes .....	17
2.4 MÉTRICAS DE DESEMPENHO .....	18
<b>3 SOLUÇÃO PROPOSTA</b> .....	<b>21</b>
3.1 COLETA DE DADOS E ANÁLISE EXPLORATÓRIA .....	21
3.2 PRÉ-PROCESSAMENTO .....	22
3.3 MINERAÇÃO DE DADOS .....	23
3.4 PÓS-PROCESSAMENTO .....	25
<b>4 CONSIDERAÇÕES FINAIS</b> .....	<b>28</b>
<b>REFERÊNCIAS</b> .....	<b>29</b>
<b>APÊNDICE A - Código do Trabalho em Python</b> .....	<b>32</b>

## 1 INTRODUÇÃO

A desinformação na mídia não é algo novo. Tem estado conosco desde o desenvolvimento dos primeiros sistemas de escrita (MARCUS, 1993). Segundo Allcott e Gentzkow (2017), diante da facilidade de criação e compartilhamento de informações, as mídias sociais tornaram-se um meio prático para a disseminação de *fake news*, tais notícias vêm causando grandes transtornos para a sociedade.

A utilização de *fake news* surge principalmente em tempos de eleições. *Fake news* foram recebendo mais atenção nos últimos anos, especialmente desde as eleições nos Estados Unidos em 2016 (AHMED; TRAORE; SAAD, 2017).

É difícil para os humanos detectarem notícias falsas. Pode-se argumentar que a única maneira de uma pessoa identificar manualmente notícias falsas é ter um vasto conhecimento dos tópicos cobertos nas notícias. Mesmo com o conhecimento, é consideravelmente difícil identificar com sucesso se as informações são reais ou falsas. (AHMED; TRAORE; SAAD, 2017).

A natureza aberta da internet e das mídias sociais, além do recente avanço na ciência da computação, simplificam o processo de criação e disseminação de notícias falsas. (AHMED; TRAORE; SAAD, 2017). Entretanto, certos padrões nas *fake news* as diferenciam de notícias verdadeiras. Essas características podem auxiliar na identificação das mesmas.

Diante disso, este trabalho objetiva aplicar técnicas de inteligência artificial para a detecção automática de *fake news*, apresentar os resultados e avaliar a eficácia do modelo proposto.

### 1.1 OBJETIVO GERAL

Este trabalho tem por objetivo a detecção automática de *fake news* dentre notícias reais utilizando técnicas de inteligência artificial, especificamente a técnica *Naive Bayes*.

### 1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos são:

- Pesquisar sobre o Teorema de Bayes;
- Utilizar a técnica *Naive Bayes* para detecção de *fake news*;
- Apresentar as métricas de desempenho e os resultados do modelo proposto.

### 1.3 ORGANIZAÇÃO DO TRABALHO

Este trabalho está dividido em quatro capítulos, sendo o primeiro a introdução, que contém o objetivo geral e objetivos específicos desta monografia.

No segundo capítulo, Fundamentação Teórica, são apresentados os conceitos abordados neste trabalho: definição de *fake news*, aprendizado de máquina, algoritmos de aprendizado de máquina para detecção de *fake news* e as métricas de desempenho.

O terceiro capítulo, Solução Proposta, apresenta as etapas realizadas para o desenvolvimento do modelo proposto para a detecção de *fake news*, bem como os resultados obtidos pelo modelo.

No quarto capítulo, Considerações Finais, são abordadas as conclusões sobre o modelo proposto e sugestões para trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 FAKE NEWS

Segundo Allcot e Gentzkow (2017), *fake news* são notícias intencionalmente e certificadamente falsas que poderiam enganar os leitores. De acordo com Tandoc Jr., Lim e Ling (2018) as *fake news* são notícias com conteúdo falso e sem embasamento, com o intuito de enganar e manipular o leitor.

O que é comum nas definições de *fake news* é como as notícias falsas se apropriam da aparência e sentimento das notícias reais; como os *websites* se parecem; como os artigos são escritos; como as imagens são fotografadas. As *fake news* se escondem sob um verniz de legitimidade à medida que assumem alguma forma de credibilidade, tentando parecer uma notícia real. (Tandoc Jr., Lim e Ling, 2018, p. 11, tradução nossa).

O ser humano possui grandes dificuldades, não somente para checagem de *fake news*, mas também para conteúdos enganosos em geral (SILVA et al., 2020). Os indivíduos possuem entre 50 e 63% de acurácia para detectar fraudes (RUBIN; CONROY, 2011).

Segundo Silva et al. (2020), a constatação de *fake news*, baseada em algoritmos computacionais, é extremamente atrativa, pois estes sistemas são mais objetivos que os julgamentos humanos, os quais são propensos a vieses. Por estas razões, é fundamental que existam ferramentas disponíveis com o objetivo de auxiliarem na detecção de notícias falsas.

### 2.2 APRENDIZADO DE MÁQUINA

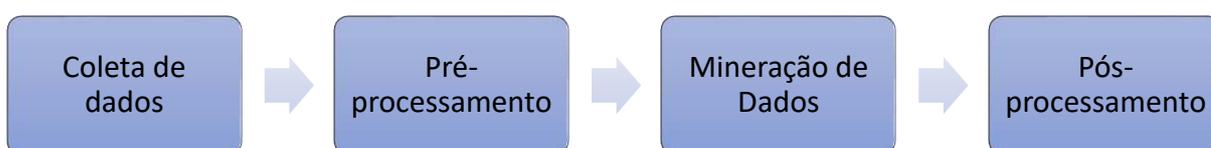
Segundo Géron (2019), aprendizado de máquina (*machine learning*) é a ciência (e arte) de programar computadores para que eles possam aprender com os dados.

De acordo com Faceli et. al. (2011) aprendizado de máquina é uma área de pesquisa da Inteligência Artificial que visa ao desenvolvimento de programas de computador com a capacidade de aprender a executar uma dada tarefa com sua própria experiência.

Algoritmos de aprendizado de máquina são pedaços de código que ajudam os cientistas de dados a explorar e analisar conjuntos de dados complexos e entendê-los. Em modelos de aprendizado de máquina, o objetivo é criar ou descobrir padrões que possam ser utilizados para realizar previsões ou classificar informações

Em uma visão geral, pode-se resumir o Aprendizado de Máquina em 4 etapas: Coleta de dados, Pré-processamento (formatação de dados), Mineração de dados (aplicação de algoritmos), Pós-processamento (avaliação dos resultados).

**Figura 1 – Etapas do Aprendizado de Máquina**



**Fonte: Próprio Autor (2021)**

Na etapa de coleta de dados, o conjunto de dados utilizado terá um papel fundamental para medir a eficiência dos algoritmos, sabendo disso é de suma importância verificar a qualidade e a quantidade dessas informações que serão usadas para ter uma melhor performance e acerto de previsão (MOHRI; ROSTAMIZADEH; TALWALKAR, 2018).

Na etapa de Pré-processamento, é necessário normalizar, limpar e reduzir a base de dados. Este processo nos auxiliará a reduzir o tamanho do *dataset* removendo informações irrelevantes que possam existir no banco de dados (AHMED; TRAORE; SAAD, 2017). Especificamente para o problema proposto, utilizou-se técnicas de processamento de linguagem natural tais como: padronização dos termos em caixa baixa e eliminação de pontuação e acentos, remoção de *StopWords* onde são removidas palavras irrelevantes para o texto, por exemplo, artigos e preposições e, por último, *Lemmatization*, processo de desflexionar uma palavra para determinar o seu lema, por exemplo, as palavras “tiver”, “tenho”, “tinha”, “tem” são do mesmo lema “ter”.

Segundo Tsytarau et. al. (2012), na etapa de Mineração de dados, é aplicado o algoritmo escolhido para a resolução do problema proposto. Deste modo, treina-se o modelo para identificar e distinguir as características mais relevantes do *dataset*.

Finalmente, o algoritmo é utilizado para realizar novas previsões em novos dados. A seção 2.3 traz os detalhes do algoritmo escolhido para a detecção de *fake news*.

Por fim, na etapa Pós-processamento, são avaliados os resultados obtidos de acordo com as métricas explanadas na seção 2.4.

## 2.3 ALGORITMOS DE APRENDIZADO DE MÁQUINA

De acordo com Gillispie (2013), algoritmos são estruturas codificadas para a transformação de entrada de dados numa desejável saída, baseadas em cálculos específicos. Ou ainda, um algoritmo nada mais é do que um conjunto de regras finitas organizadas a fim de resolver um problema específico ou executar uma tarefa específica (SILVEIRA, 2018).

Ao fim do processo de Aprendizado de Máquina, é gerado um algoritmo que pode ser classificado principalmente em dois tipos: supervisionado e não supervisionado. (MARUMO, 2018).

Os algoritmos supervisionados correspondem a tarefas em que os algoritmos tentam prever um resultado generalizado usando como base um *dataset* com saídas já rotuladas. Já os algoritmos não supervisionados, tem como objetivo aprender as características pertinentes em um *dataset* não classificado, reunidos por comportamento (MOHRI; ROSTAMIZADEH; TALWALKAR, 2018).

O aprendizado supervisionado procura resolver dois tipos de problemas: classificação e regressão. O primeiro, a classificação, analisa os dados onde a saída esperada é um valor discreto, como exemplo o tema deste trabalho, a classificação de notícias como verdadeira ou *fake*. O segundo, a regressão, analisa os dados onde os valores de saída são valores contínuos. Um exemplo seria prever o preço de uma casa, utilizando como entrada os valores da localização, tamanho do terreno e demanda de imóveis na região. (BISHOP, 2006)

Neste trabalho, será utilizado o algoritmo *Naive Bayes*, baseado no Teorema de Bayes para a detecção de *fake news* e ao final serão avaliados os resultados obtidos.

### 2.3.1 Naive Bayes

O método bayesiano é baseado na teoria da decisão estatística e utiliza o Teorema de *Bayes* para determinar a distribuição da probabilidade de padrões pertencentes a cada classe (DUDA; HART; SOTRK, 2001). O *Naive Bayes* é um modelo simples de classificação, porém proporciona bons resultados mesmo quando comparados aos obtidos com classificadores mais complexos (DUDA; HART; SOTRK, 2001).

As notações que seguem foram apresentadas no livro de Mitchell (1997), estas são utilizadas para definição do Teorema de Bayes:

#### **Teorema de Bayes:**

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)} \quad (1)$$

Onde:

$h$  é a hipótese;

$D$  é um conjunto de treino;

$P(h)$  é a probabilidade inicial da hipótese  $h$  ;

$P(D)$  é a probabilidade do conjunto de treino  $D$  ser observado;

$P(D|h)$  é a probabilidade de ocorrer  $D$  dado que  $h$  ocorre;

$P(h|D)$  é a probabilidade do evento  $h$  ocorrer dado o conjunto  $D$ .

Em muitos cenários de aprendizagem, o algoritmo considera algum conjunto de hipóteses candidatas  $H$  e está interessado em encontrar a hipótese mais provável  $h \in H$  dado o conjunto de treino observado  $D$ . Qualquer hipótese maximamente provável é chamada de hipótese *maximum a posteriori* (MAP). Pode-se determinar a hipótese MAP utilizando o teorema de Bayes para calcular a probabilidade posterior de cada hipótese candidata. Mais precisamente, diz-se que  $h_{MAP}$  é a hipótese MAP dada por:

$$\begin{aligned}
 h_{MAP} &= \operatorname{argmax}_{h \in H} P(h|D) \\
 &= \operatorname{argmax}_{h \in H} \frac{P(D|h)P(h)}{P(D)} \\
 &= \operatorname{argmax}_{h \in H} P(D|h)P(h)
 \end{aligned}
 \tag{2}$$

No passo final acima, o termo  $P(D)$  é desconsiderado uma vez que ele é constante independente de  $h$ .

O classificador obtido usando esta função discriminante é chamado de classificador *Naive Bayes*. O termo *naive* (ingênuo) vem da hipótese de que os atributos de um exemplo são independentes entre si, o que normalmente não ocorre em problemas de classificação (FACELI et al., 2011).

## 2.4 MÉTRICAS DE DESEMPENHO

Em um contexto de classificação binária, essas são as principais métricas que são usadas para medir o desempenho dos algoritmos de classificação (KOHAVI; PROVOST, 1998).

A matriz de confusão (*confusion matrix*) é usada para termos um cenário mais completo quando estamos avaliando o desempenho de um modelo (KOHAVI; PROVOST, 1998). O Quadro 1 mostra uma matriz de confusão para um problema de duas classes.

Quadro 1 – Matriz de Confusão

		Valor Previsto	
		Positivo	Negativo
Valor Verdadeiro	Positivo	TP Verdadeiro Positivo	FN Falso Negativo
	Negativo	FP Falso Positivo	TN Verdadeiro Negativo

Fonte: Próprio Autor (2021)

As classes verdadeiras são representadas nas linhas da matriz, as classes previstas são representadas nas colunas. Onde:

**Verdadeiro Positivo (TP):** O valor previsto é “notícia *fake*” e o valor real é “notícia *fake*”;

**Falso Negativo (FN):** O valor previsto é “notícia verdadeira” e o valor real é “notícia *fake*”;

**Falso Positivo (FP):** O valor previsto é “notícia *fake*” e o valor real é “notícia verdadeira”;

**Verdadeiro Negativo (TN):** O valor previsto é “notícia verdadeira” e o valor real é “notícia verdadeira”.

A partir dessa matriz, uma série de outras medidas podem ser usadas para medir o desempenho do classificador (FACELI et al., 2011). As notações que seguem foram apresentadas no artigo de Rodrigues (2019), estas são utilizadas para definição das métricas de avaliação do classificador:

**Acurácia:** Indica a performance geral do modelo. Dentre todas as classificações, quantas o modelo classificou corretamente.

$$Acurácia = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

**Precisão:** Dentre todas as classificações de classe positivo que o modelo fez, quantas estão corretas. Pode ser calculada para ambas as classes, *fake* e real.

$$Precisão = \frac{TP}{TP + FP} \quad (4)$$

**Recall (ou sensibilidade):** Dentre todas as situações de classe positivo como valor esperado, quantas estão corretas. Pode ser calculada para ambas as classes, *fake* e real.

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

**F1-Score:** Média harmônica entre precisão e recall. Pode ser calculada para ambas as classes, *fake* e *real*.

$$F1 - Score = \frac{2 * TP}{2 * TP + FP + FN} \quad (6)$$

### 3 SOLUÇÃO PROPOSTA

Para a execução deste trabalho, utilizou-se a plataforma Google Colaboratory<sup>1</sup>. Google Colab, ou Google Colaboratory, é um serviço de armazenamento em nuvem de notebooks voltados à criação e execução de códigos em Python, diretamente em um navegador, sem a necessidade de nenhum tipo de instalação de software em uma máquina (ROVEDA, 2019).

Neste capítulo, são descritas as 4 etapas: Coleta de dados, Pré-processamento (formatação de dados), Mineração de dados (aplicação de algoritmos) e Pós-processamento (avaliação dos resultados) para avaliar a solução proposta.

#### 3.1 COLETA DE DADOS E ANÁLISE EXPLORATÓRIA

O conjunto de dados utilizado neste trabalho contém milhares de artigos de notícias. Este conjunto de dados foi extraído da plataforma *Kaggle*<sup>2</sup>. *Kaggle* é uma comunidade online para cientistas de dados, pertencente ao grupo Google, que oferece competições de *machine learning*, *datasets*, *notebooks* e educação na área de ciência de dados.

Antes de iniciar a limpeza dos dados, é de extrema importância explorar as características do *dataset*. A base de dados escolhida<sup>3</sup> possui 4 atributos: ID#, Título, Texto e Label:

**Figura 2 – Amostra do conjunto de dados utilizados**

	ID#	Title	Text	Label
0	8476	You Can Smell Hillary's Fear	Daniel Greenfield, a Shillman Journalism Fello...	FAKE
1	10294	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg LinkedIn Reddit Stumbleu...	FAKE
2	3608	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	REAL
3	10142	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	FAKE
4	875	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	REAL

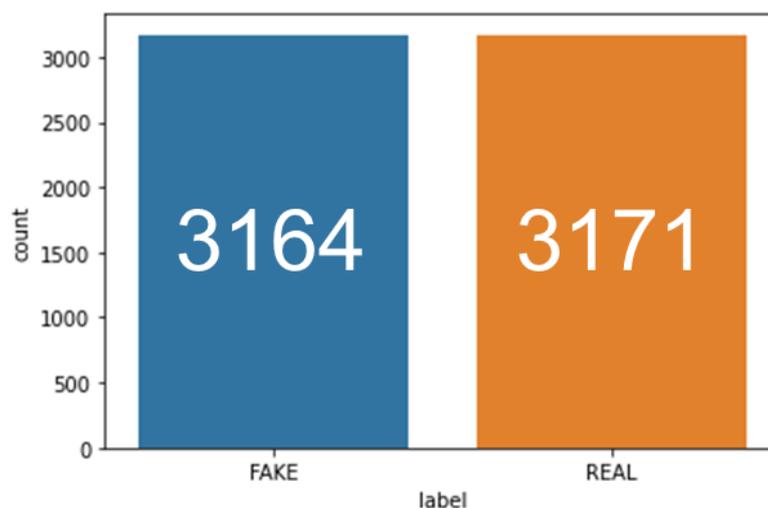
**Fonte: Próprio Autor (2021)**

O atributo label pode assumir dois valores, “FAKE” para notícias falsas e “REAL” para notícias verdadeiras. Neste repositório, têm-se um total de 6335 notícias, sendo 3164 notícias falsas e 3171 notícias verdadeiras.

<sup>1</sup> Google Colaboratory: <https://colab.research.google.com/>

<sup>2</sup> Kaggle: <https://www.kaggle.com/>

<sup>3</sup> fake\_or\_real\_news.csv: <https://www.kaggle.com/hassanamin/textdb3>

**Figura 3 – Distribuição de notícias falsas e verdadeiras**

Fonte: Próprio Autor (2021)

### 3.2 PRÉ-PROCESSAMENTO

Na etapa de Pré-processamento, é necessário normalizar, limpar e reduzir a base de dados. Segundo Ahmed et. al. (2017), este processo auxilia reduzir o tamanho do *dataset*, removendo informações irrelevantes que possam existir no banco de dados. Desta forma, o pré-processamento se deu nos seguintes procedimentos:

1. **Padronização dos termos:** Todas as palavras nas notícias são padronizadas em caixa baixa, eliminação de pontuação, acentos, *links* de *websites*, palavras que possuam números.
2. **Remoção de StopWords:** São removidas palavras irrelevantes para o texto como, por exemplo, artigos e preposições.
3. **Lematização:** Processo de desflexionar uma palavra para determinar o seu lema. Por exemplo, as palavras “tiver”, “tenho”, “tinha”, “tem” são do mesmo lema “ter”.

Após os procedimentos de pré-processamento, realizou-se limpeza de colunas que não são necessárias (ID# e Título) para a aplicação do modelo e alterou-

se os labels “FAKE” para “0” e “REAL” para “1”. Assim, obteve-se o *dataset* a ser utilizado para a aplicação do algoritmo *Naive Bayes*:

**Figura 4 – Base de dados após o pré-processamento**

	Text	Label
0	smell hillary's fear daniel greenfield shillma...	0
1	watch exact moment paul ryan commit political ...	0
2	kerry go paris gesture sympathy u secretary st...	1
3	bernie supporter twitter erupt anger dnc try w...	0
4	battle new york primary matter primary day new...	1

**Fonte: Próprio Autor (2021)**

### 3.3 MINERAÇÃO DE DADOS

Após a realização do pré-processamento, separou-se a totalidade dos dados em dois grupos, sendo o primeiro responsável pelo aprendizado do modelo, e o segundo por realizar os testes e medir a eficácia do algoritmo. O *dataset* foi separado em 75% para aprendizado e 25% para teste.

A solução proposta para a detecção de *fake news*, neste trabalho, foi o algoritmo *Naive Bayes*. Para a aplicação de tal algoritmo, é necessário realizar a extração de características numéricas dos textos. Para isso, utilizou-se a técnica de ponderação de termos *Term Frequency-Inverse Document Frequency* (TF-IDF).

A primeira porção deste método, “TF”, refere-se a *term frequency*, ou frequência do termo, em português, e indica quantas vezes os termos podem ser encontrados nos documentos. A frequência do termo, sozinha, não é suficiente para obter as características; no entanto, como há muitas palavras comuns, como as preposições e artigos, que não carregam muitas informações sobre a notícia, não é adequado dar peso a estas palavras mais do que termos que raramente aparecem e são mais informativos. Por este motivo, aplicou-se a remoção de *StopWords* na etapa de pré-processamento.

A segunda porção deste método, “IDF”, refere-se a *inverse document frequency* ou frequência inversa do documento, em português. IDF é utilizado para

penalizar tais termos que ocorrem comumente em diferentes contextos sem adicionar informações relevantes. A equação de frequência inversa do documento é dada por:

$$idf(t) = \log \frac{1 + n}{1 + df(t)} + 1 \quad (7)$$

$n$  representa o total de documentos, ou notícias;

$t$  é o termo em questão;

$df(t)$  representa a frequência do documento desse termo;

Quanto menos frequente um termo ocorrer em diferentes documentos, menor será o denominador, tornando a fração maior e, por conseguinte,  $idf(t)$  maior.

A ponderação de termos TF-IDF permite atribuir importância aos *tokens* que realmente carregam algumas informações, equilibrando a frequência geral do *token* com a sua frequência nas notícias. TF-IDF é o produto entre a frequência do termo e a frequência inversa do documento, matematicamente dada por:

$$tf - idf(t, d) = tf(t, d) * idf(t) \quad (8)$$

Onde  $d$  representa um documento. Quanto mais comumente a palavra aparecer, maior será o valor de  $tf$ . Mas, se for o caso em documentos diferentes, será penalizado com um pequeno  $idf$ . Por outro lado, uma palavra que raramente ocorre pode ter um valor menor de  $tf$ , mas ser destacada por valores de  $idf$  maiores por não ocorrer com frequência em documentos diferentes.

Aplicando o método TF-IDF para extração de características numéricas dos textos, obtém-se uma matriz esparsa com o tamanho de número de documentos por número de palavras únicas.

**Figura 5 – Matriz Esparsa**

```
<4751x92489 sparse matrix of type '<class 'numpy.float64''>'
  with 1180857 stored elements in Compressed Sparse Row format>
```

**Fonte: Próprio Autor (2021)**

Após a obtenção da matriz esparsa, é possível aplicar o algoritmo *Naive Bayes* para calcular suas probabilidades e classificar as notícias do *dataset*.

**Figura 6 – Aplicação do algoritmo *Naive Bayes***

```
[58] nb_classifier = MultinomialNB()
      nb_classifier.fit(tfidf_train, y_train)

      pred1 = nb_classifier.predict(tfidf_test)
```

**Fonte: Próprio Autor (2021)**

### 3.4 PÓS-PROCESSAMENTO

A avaliação da execução do algoritmo é realizada conforme descrito na seção 2.4. O resultado obtido encontra-se nas Tabela 1 e Tabela 2.

**Tabela 1 - Desempenho do algoritmo *Naive Bayes* por classe**

Notícia / Casse	Precisão	Recall	F1-Score
Fake / 0	96%	72%	82%
Real / 1	77%	98%	86%

**Fonte: Próprio Autor (2021)**

**Tabela 2 – Desempenho geral do algoritmo *Naive Bayes***

Algoritmo	Acurácia
<i>Naive Bayes</i>	84%

**Fonte: Próprio Autor (2021)**

O classificador *Naive Bayes* apresentou uma acurácia de 84% de predições corretas. A métrica precisão das classes *fake* e *real* obtiveram os seguintes resultados, respectivamente, 96% e 77%, representando as predições positivas realizadas corretamente.

A métrica *Recall* das classes *fake* e *real* apresentaram os seguintes valores, respectivamente, 72% e 98%, representando se o algoritmo conseguiu identificar corretamente a classe da notícia.

Na Tabela 3, pode-se observar os resultados da matriz de confusão do algoritmo *Naive Bayes*, na matrix 2 x 2, na horizontal o Valor Verdadeiro e na vertical o Valor Previsto.

**Quadro 2 – Matriz de Confusão do algoritmo *Naive Bayes***

		Valor Previsto	
		Positivo	Negativo
Valor Verdadeiro	Positivo	<b>565</b>	<b>225</b>
	Negativo	<b>22</b>	<b>772</b>

Fonte: Próprio Autor (2021)

O classificador obteve um total de 565 acertos para notícias *fakes* que realmente eram notícias falsas. Por outro lado, o classificador obteve um total de 772 acertos para notícias verdadeiras que realmente eram verdadeiras. Na figura 7, pode-se observar exemplos de notícias previstas corretamente.

**Figura 7 – Notícias previstas corretamente**

	Text	Valor_Real	Valor_Previsto
150	shutdown clash return force december notable n...	1	1
795	trey gowdy "what word loretta lynch talking co...	0	0
1306	paul christie embrace obamacare day j fred buz...	1	1
270	exclusive siege islamabad brig asif h raja oct...	0	0
6059	obama throw hillary clinton bus support direct...	0	0

Fonte: Próprio Autor (2021)

Em contrapartida, nas figuras 8 e 9, observa-se notícias previstas incorretamente, respectivamente, falsos negativos e falsos positivos.

**Figura 8 – Notícias previstas: falsos negativos**

	Text	Valor_Real	Valor_Previsto
4621	george laura bush celebrate birthday daily wir...	0	1
5012	assassination attempt reno trump rally assassi...	0	1
4605	trump see today final killer blunder channel l...	0	1
4506	sc gov nikki haley announces she's voting dona...	0	1
3320	trump voter listens carefully call national un...	0	1

Fonte: Próprio Autor (2021)

**Figura 9 – Notícias previstas: falsos positivos**

	Text	Valor_Real	Valor_Previsto
2056	putin say abandon russian ukraine nationalist ...	1	0
3733	thanksgiving lesson syrian refugee debate pari...	1	0
7	'britain's schindler' dy czech stockbroker sav...	1	0
67	funny republican upset facebook column gop opp...	1	0
2070	dr manny water crisis flint tip iceberg water ...	1	0

Fonte: Próprio Autor (2021)

## 4 CONSIDERAÇÕES FINAIS

*Fake news*, o termo que ganhou visibilidade durante as últimas eleições presidenciais de diversos países, tornou-se uma preocupação para a sociedade. As *fake news* contribuem de uma forma negativa no meio social onde caminha em direção oposta a importância do objetivo de informar de maneira clara e segura.

Este trabalho teve como objetivo o desenvolvimento de um modelo de detecção de *fake news* combinando as áreas de classificação de texto, processamento de linguagem natural, aprendizado de máquina e estatística.

O modelo levou em consideração a abordagem de representação estatística TF-IDF. Após aplicado o método TF-IDF para extração de características numéricas dos textos, obteve-se uma matriz esparsa, possibilitando a aplicação do algoritmo *Naive Bayes*. Esse modelo, quando avaliado nos dados de teste, obteve uma acurácia de 84% de predições corretas. O classificador obteve um total de 565 acertos para notícias *fakes* que realmente eram notícias falsas e obteve um total de 772 acertos para notícias verdadeiras que realmente eram verdadeiras.

Tais resultados não podem ser avaliados isoladamente, deve-se entender o contexto das notícias avaliadas. Tendo em vista que a base de dados utilizada foi extraída do *Kaggle* e continha em torno de 6300 notícias, acredita-se que o resultado pode ser aprimorado com bases de dados maiores e diferentes contextos.

Como trabalhos futuros, objetiva-se fazer testes com outros algoritmos classificadores para a detecção de *fake news*, tais como, Regressão Logística, Árvore de Decisão, *Support Vector Machine* e Redes Neurais, afim de comparar o resultado entre os mesmos.

## REFERÊNCIAS

ALLCOTT, H.; GENTZKOW, M. **Social media and fake news in the 2016 election.** Journal of Economic Perspectives, v. 31, n. 2, p. 211–36, 2017.

AHMED, H.; TRAORE, I.; SAAD, S. **Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques.** In: Traore I., Woungang I., Awad A. (eds) Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments. ISDDC 2017. Lecture Notes in Computer Science, vol 10618. Springer, Cham (pp. 127-138), 2017.

ALBRIGHT, J. 2016. **“The #Election 2016 Micro-Propaganda Machine.”** Medium. Disponível em: <https://medium.com/@d1gi/the-election2016-micro-propaganda-machine-383449cc1fba>. Acesso em 06 jun 2021.

BISHOP, C. M. **Pattern recognition and machine learning.** [S.l.]: springer, 2006.

DUDA, R. O.; HART, P. E.; STORK, D. G. **Pattern Classification.** 2º Edição. Wiley, 2001.

FACELI, K.; LORENA, A. C.; GAMA, J. ; CARVALHO, A. C. P. L. F. de. **Inteligência artificial: uma abordagem de aprendizado de máquina.** Rio de Janeiro: LTC, 2011.

GÉRON, A. **Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems.** 2nd. [S.l.]: O'Reilly Media, Inc., 2019.

GILLESPIE, T. **The Relevance of Algorithms. in Media Technologies: Essays on Communication, Materiality, and Society,** ed. Tarleton Gillespie, Pablo Boczkowski, and Kirsten Foot. Cambridge, MA: MIT Press. 2013. Disponível em: [https://www.microsoft.com/en-us/research/wp-content/uploads/2014/01/Gillespie\\_2014\\_The-Relevance-of-Algorithms.pdf](https://www.microsoft.com/en-us/research/wp-content/uploads/2014/01/Gillespie_2014_The-Relevance-of-Algorithms.pdf). Acesso em: 06 jun 2021

KOHAVI, R.; PROVOST, F. **Glossary of terms.** Machine Learning, v. 2, p. 271–274, 01 1998.

MARCUS, J. **Mesoamerican Writing Systems: Propaganda, Myth, and History in Four Ancient Civilizations**. Princeton, NJ: Princeton University Press, 1993.

MARUMO, F. S. **Deep Learning para classificação de Fake News por sumarização de texto**. Universidade Estadual de Londrina, 2018.

MITCHELL, T. M. **Machine Learning**. 1º Edição. McGraw-Hill, 1997.

MOHRI, M.; ROSTAMIZADEH, A.; TALWALKAR, A. **Foundations of machine learning**. [S.l.]: MIT press, 2018.

RODRIGUES, V. **Métricas de Avaliação: acurácia, precisão, recall... quais as diferenças?** Abril 2019. Disponível em: <https://vitorborbarodrigues.medium.com/m%C3%A9tricas-de-avalia%C3%A7%C3%A3o-acur%C3%A1cia-precis%C3%A3o-recall-quais-as-diferen%C3%A7as-c8f05e0a513c>. Acesso em: 01 de aug 2021.

ROVEDA, U. **Google Colab: o que é, como usar e quais são as vantagens?** Novembro 2019. Disponível em: <https://kenzie.com.br/blog/google-colab/>. Acesso em: 09 de jun 2021.

RUBIN, Victoria L.; CONROY, Niall J. **Challenges in automated deception detection in computer-mediated communication**. Proceedings of the ASIST Annual Meeting, v. 48, 2011. Disponível em: <https://asistdl.onlinelibrary.wiley.com/doi/epdf/10.1002/meet.2011.14504801098>. Acesso em 21 de jun 2021.

SILVA, Renato M.; SANTOS, Roney L.S.; ALMEIDA, Tiago A.; PARDO, Thiago A.S. **Towards automatically filtering fake news in Portuguese**. Expert Systems with Applications, v. 146, p. 113199, 2020. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0957417420300257>. Acesso em 21 de jun 2021.

SILVEIRA, S. **Máquinas não são preconceituosas**. Tecnologia UOL. 14/05/2018. Disponível em: <https://tecnologia.uol.com.br/blogs-e-colunas/coluna/stefanie-silveira/2018/05/14/maquinas-nao-sao-preconceituosas.htm>. Acesso em: 06 jun 2021

TANDOC JR., E. C.; LIM, Z. W.; LING, R. **Defining “fake news” a typology of scholarly definitions**. *Digital journalism*, Taylor & Francis, 2018.

TSYTSARAU, M.; PALPANAS, T. **Survey on mining subjective data on the web.** **Data Mining and Knowledge Discovery**, Kluwer Academic Publishers, v. 24, n. 3, p. 478-514, 2012.

## APÊNDICE A - Código do Trabalho em Python

### ▼ Detecção de Fake News utilizando Naive Bayes

#### ▼ Bibliotecas

```
[42] import pandas as pd
import numpy as np
import seaborn as sns
from google.colab import files
import io

import nltk
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('punkt')
from nltk.corpus import stopwords
from nltk.util import ngrams
from nltk.tokenize import word_tokenize, sent_tokenize
import re
import string

from sklearn.model_selection import train_test_split

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.naive_bayes import MultinomialNB

from sklearn import metrics
from sklearn.metrics import classification_report
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix, plot_confusion_matrix
import matplotlib.pyplot as plt
```

```
▶ [nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data] Package stopwords is already up-to-date!
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data] Package wordnet is already up-to-date!
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data] Package punkt is already up-to-date!
```

#### ▼ Coleta de Dados e Análise Exploratória

```
[2] uploaded = files.upload()
```

Escolher Arquivos fake\_or\_real\_news.csv

- **fake\_or\_real\_news.csv**(application/vnd.ms-excel) - 30696129 bytes, last modified: 19/10/2019 - 100% done

Saving fake\_or\_real\_news.csv to fake\_or\_real\_news.csv

```
[3] # Dataset do Kaggle https://www.kaggle.com/hassanamin/textdb3
```

```
df = pd.read_csv(io.BytesIO(uploaded['fake_or_real_news.csv']))
```

```
[4] print(df.shape)
print('\n')
print(df.info())
print('\n')
df.head()
```

```
(6335, 4)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6335 entries, 0 to 6334
Data columns (total 4 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Unnamed: 0  6335 non-null  int64
1   title       6335 non-null  object
2   text        6335 non-null  object
3   label       6335 non-null  object
dtypes: int64(1), object(3)
memory usage: 198.1+ KB
None
```

```
[4] Unnamed: 0 title text label
0 8476 You Can Smell Hillary's Fear Daniel Greenfield, a Shillman Journalism Fello... FAKE
1 10294 Watch The Exact Moment Paul Ryan Committed Pol... Google Pinterest Digg LinkedIn Reddit Stumbleu... FAKE
2 3608 Kerry to go to Paris in gesture of sympathy U.S. Secretary of State John F. Kerry said Mon... REAL
3 10142 Bernie supporters on Twitter erupt in anger ag... — Kaydee King (@KaydeeKing) November 9, 2016 T... FAKE
4 875 The Battle of New York: Why This Primary Matters It's primary day in New York and front-runners... REAL
```

```
[6] df.describe()
```

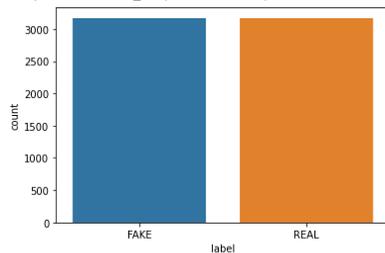
```
Unnamed: 0
count 6335.000000
mean 5280.415627
std 3038.503953
min 2.000000
25% 2674.500000
50% 5271.000000
75% 7901.000000
max 10557.000000
```

```
[7] df.isnull().sum()
```

```
Unnamed: 0 0
title 0
text 0
label 0
dtype: int64
```

```
[8] sns.countplot(df.label)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/_decorators.py:43: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the
FutureWarning
<matplotlib.axes._subplots.AxesSubplot at 0x7f5d6b427910>
```



## ▼ Pré-processamento

```
[9] df.columns = ['ID#', 'Title', 'Text', 'Label']
```

```
df = df.replace('FAKE',0)
df = df.replace('REAL',1)
```

```
df['Text'] = df['Title'] + ' ' + df['Text']
df.head()
```

	ID#	Title	Text	Label
0	8476	You Can Smell Hillary's Fear	You Can Smell Hillary's Fear Daniel Greenfield...	0
1	10294	Watch The Exact Moment Paul Ryan Committed Pol...	Watch The Exact Moment Paul Ryan Committed Pol...	0
2	3608	Kerry to go to Paris in gesture of sympathy	Kerry to go to Paris in gesture of sympathy U....	1
3	10142	Bernie supporters on Twitter erupt in anger ag...	Bernie supporters on Twitter erupt in anger ag...	0
4	875	The Battle of New York: Why This Primary Matters	The Battle of New York: Why This Primary Matte...	1

```
[10] news = df.drop(['ID#', 'Title'], axis=1)
news.head()
```

#Dataset a ser trabalhado

	Text	Label
0	You Can Smell Hillary's Fear Daniel Greenfield...	0
1	Watch The Exact Moment Paul Ryan Committed Pol...	0
2	Kerry to go to Paris in gesture of sympathy U...	1
3	Bernie supporters on Twitter erupt in anger ag...	0
4	The Battle of New York: Why This Primary Matte...	1

```
[11] # Letras minusculas, remover [], links, palavras que tenham numeros e pontuações
```

```
def clean_text(text):
    text = str(text).lower()
    text = re.sub('[.?!\']', '', text)
    text = re.sub('https?://\S+|www\.\S+', '', text)
    text = re.sub('<.*?>+', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), '', text)
    text = re.sub('\n', '', text)
    text = re.sub('\w*\d\w*', '', text)
    return text

news['Text'] = news['Text'].apply(lambda x:clean_text(x))

news.head()
```

	Text	Label
0	you can smell hillary's fear daniel greenfield...	0
1	watch the exact moment paul ryan committed pol...	0
2	kerry to go to paris in gesture of sympathy us...	1
3	bernie supporters on twitter erupt in anger ag...	0
4	the battle of new york why this primary matter...	1

```
[12] # Remover stop words
```

```
stop = stopwords.words('english')
news['Text'] = news['Text'].apply(lambda x: ' '.join([word for word in x.split() if word not in (stop)]))

news.head()
```

	Text	Label
0	smell hillary's fear daniel greenfield shillma...	0
1	watch exact moment paul ryan committed politic...	0
2	kerry go paris gesture sympathy us secretary s...	1
3	bernie supporters twitter erupt anger dnc trie...	0
4	battle new york primary matters primary day ne...	1

```
🔍 #Lemmatize (Deixar apenas a raiz das palavras)
```

```
def lemmatize_words(text):
    wn1 = nltk.stem.WordNetLemmatizer()
    lem = ' '.join([wn1.lemmatize(word) for word in text.split()])
    return lem

news['Text'] = news['Text'].apply(lemmatize_words)

news.head()
```

```
🔍
```

	Text	Label
0	smell hillary's fear daniel greenfield shillma...	0
1	watch exact moment paul ryan committed politic...	0
2	kerry go paris gesture sympathy u secretary st...	1
3	bernie supporter twitter erupt anger dnc tried...	0
4	battle new york primary matter primary day new...	1

## ▼ Mineração de Dados

```
#Dividir o Dataset em Treino e Teste
X_train, X_test, y_train, y_test = train_test_split(news['Text'],
                                                    news['Label'], test_size=0.25)
display(X_train.head())
print('\n')
display(y_train.head())

print("\nThere are {} documents in the training data.".format(len(X_train)))
```

```
2409  blasphemy law fanatic went rampage men killed ...
439   benghazi bust trey gowdy gop colleague embarra...
5035  assange point hillary's email confirm oligarch...
2302  assad say boy ambulance fake prof assad say bo...
4766  clinton deplorables slip campaign hint gamecha...
Name: Text, dtype: object
```

```
2409  1
439   1
5035  0
2302  0
4766  1
Name: Label, dtype: int64
```

There are 4751 documents in the training data.

```
[16] #Extração de Features
my_tfidf = TfidfVectorizer(stop_words='english', max_df=0.7)

tfidf_train = my_tfidf.fit_transform(X_train)
tfidf_test = my_tfidf.transform(X_test)

tfidf_train
```

```
<4751x92489 sparse matrix of type '<class 'numpy.float64''>'
with 1180857 stored elements in Compressed Sparse Row format>
```

```
[58] #Aplicação do Naive Bayes
nb_classifier = MultinomialNB()
nb_classifier.fit(tfidf_train, y_train)

pred1 = nb_classifier.predict(tfidf_test)
```

## ▼ Pós-processamento

```
[62] #Métricas do Algoritmo

print(classification_report(y_test, pred1, target_names = ['Fake', 'True']))

acc_score = accuracy_score(y_test, pred1)

conf_mat = confusion_matrix(y_test, pred1)

print(acc_score)
print('\n')
print(conf_mat)
```

	precision	recall	f1-score	support
Fake	0.96	0.72	0.82	790
True	0.77	0.97	0.86	794
accuracy			0.84	1584
macro avg	0.87	0.84	0.84	1584
weighted avg	0.87	0.84	0.84	1584

0.8440656565656566

```
[[565 225]
 [ 22 772]]
```

```
[64] df1 = pd.DataFrame(X_test)
df1['Valor_Real'] = pd.DataFrame(y_test)
df1['Valor_Previsto'] = pred1

display(df1.head())

Falso_Negativo = df1[(df1.Valor_Real == 0) & (df1.Valor_Previsto == 1)]
Falso_Positivo = df1[(df1.Valor_Real == 1) & (df1.Valor_Previsto == 0)]

display(Falso_Negativo.head())
print('\n')
display(Falso_Positivo.head())
```

	Text	Valor_Real	Valor_Previsto
150	shutdown clash return force december notable n...	1	1
795	trey gowdy "what word loreta lynch talking co...	0	0
1306	paul christie embrace obamacare day j fred buz...	1	1
270	exclusive siege islamabad brig asif h raja oct...	0	0
6059	obama throw hillary clinton bus support direct...	0	0

	Text	Valor_Real	Valor_Previsto
4621	george laura bush celebrate birthday daily wir...	0	1
5012	assassination attempt reno trump rally assassi...	0	1
4605	trump see today final killer blunder channel I...	0	1
4506	sc gov nikki haley announces she's voting dona...	0	1
3320	trump voter listens carefully call national un...	0	1

	Text	Valor_Real	Valor_Previsto
2056	putin say abandon russian ukraine nationalist ...	1	0
3733	thanksgiving lesson syrian refugee debate pari...	1	0
7	'britain's schindler' dy czech stockbroker sav...	1	0
67	funny republican upset facebook column gop opp...	1	0
2070	dr manny water crisis flint tip iceberg water ...	1	0