Classificação de Fake News com Aprendizado de Máquina: Comparação entre MLP, Random Forest e Naive Bayes

Gabriel D. Azevedo[1]

[1]Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (CEFET-RJ)
Petrópolis- RJ- Brasil

gabriel.dias@aluno.cefet-rj.br

Abstract. *With the popularization of the internet and social networks, there is an unprecedented circulation of information, both in volume and speed. This environment facilitates the creation and dissemination of content by any user, significantly increasing the risk of spreading false information, especially due to the lack of systematic verification mechanisms. This phenomenon represents a threat to democracy, justice, public health and society as a whole, which makes the problem particularly relevant. Given this context, this work aims to classify news as true or false using three machine learning models: Multi-Layer Perceptron, Random Forest, and Multinomial Naive Bayes.*

Resumo. *Com a popularização da internet e das redes sociais, observa-se uma circulação sem precedentes de informações, tanto em volume quanto em velocidade. Esse ambiente facilita a criação e disseminação de conteúdos por qualquer usuário, ampliando significativamente o risco de propagação de informações falsas, sobretudo pela ausência de mecanismos de verificação sistemática. Esse fenômeno representa uma ameaça à democracia, à justiça, à saúde pública e à sociedade como um todo, o que torna o problema particularmente relevante. Diante desse contexto, este trabalho tem como objetivo classificar notícias como verdadeiras ou falsas por meio de três modelos de aprendizado de máquina: Multi-Layer Perceptron, Random Forest, e Multinomial Naive Bayes.*

1. Introdução

A popularização da internet e das redes sociais transformou profundamente a dinâmica de produção, distribuição e consumo de informação. Se, por um lado, esses meios ampliaram o acesso ao conhecimento e democratizaram a comunicação, por outro, criaram um ambiente no qual conteúdos são disseminados em grande escala e em alta velocidade, muitas vezes sem qualquer filtragem ou verificação prévia. Nesse contexto, a propagação de informações falsas (*fake news*) tornou-se um fenômeno global, com impactos diretos em processos democráticos, na confiança institucional, na saúde pública e na própria coesão social.

A facilidade em publicar conteúdos online, aliada à ausência de mecanismos robustos de validação automática, cria condições favoráveis para que informações enganosas se espalhem rapidamente, alcancem grandes audiências e influenciem comportamentos. As notícias falsas tendem a se disseminar mais rapidamente do que notícias verdadeiras, em especial quando exploram emoções como medo, indignação ou surpresa

[Vosoughi et al. 2018]. Como consequência, governos, organizações e pesquisadores têm direcionado esforços significativos para compreender e mitigar esse fenômeno.

Nesse cenário, métodos computacionais que automatizem a detecção de notícias falsas se tornam ferramentas estratégicas. Técnicas de aprendizado de máquina (*machine learning*) se mostram promissoras nesse domínio, pois permitem identificar padrões linguísticos e estruturais difíceis de serem percebidos manualmente. A partir de bases de dados rotuladas, modelos podem aprender a distinguir conteúdos verdadeiros e falsos, oferecendo suporte a plataformas digitais, agências de checagem e usuários em geral [Masciari et al. 2019] [Sudhakar and Kaliyamurthie 2018].

Diante desse contexto, o presente trabalho tem como objetivo desenvolver e comparar modelos de aprendizado de máquina para a classificação automática de notícias como verdadeiras ou falsas. Para isso, são avaliados três algoritmos amplamente utilizados na literatura: *Multilayer Perceptron*, *Random Forest*, e *Multinomial Naive Bayes*. A análise busca identificar não apenas o desempenho individual de cada método, mas também suas vantagens, limitações e adequação ao problema em questão.

2. Descrição dos dados

O conjunto de dados utilizado neste trabalho é o *Fake News Detection Datasets* [Emineyetm 2022]. Ele inclui notícias falsas (*fake news*), coletadas a partir de diversos sites considerados não confiáveis, segundo o Politifact e a Wikipedia, e notícias reais (*real news*), obtidas por meio de extração de dados via *crawlers* do site Reuters.

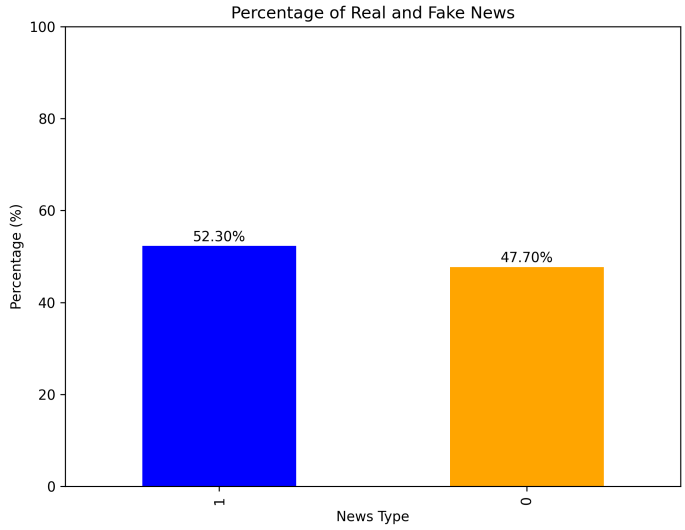

Figura 1. Distribuição percentual de notícias verdadeiras e falsas no conjunto de dados.

Os dados estão organizados em dois arquivos: *Fake.csv*, contendo 21.417 amostras, e *True.csv*, com 23.481 amostras. Ambos compartilham as mesmas características e, devido à pequena diferença no número de amostras entre notícias reais e falsas, não foi necessário aplicar técnicas específicas de balanceamento de classes. As características presentes em ambos os arquivos são:

- ***title***: título da notícia;
- ***text***: corpo da notícia;
- ***subject***: assunto abordado pela notícia;
- ***date***: data de publicação da notícia.

Para consolidar o conjunto de dados, os dois arquivos CSV foram concatenados, e uma coluna adicional chamada *target* foi incluída, indicando se a notícia é verdadeira ou falsa.

3. Descrição da Solução

Para abordar o problema de classificação de notícias como verdadeiras ou falsas, adotou-se uma estratégia de aprendizado supervisionado envolvendo a comparação de três modelos: *Multi-Layer Perceptron*, *Random Forest*, e *Multinomial Naive Bayes*. O processo foi estruturado em várias etapas, destacando-se a concatenação e o pré-processamento dos dados, a transformação textual em representação vetorial por meio de *TF-IDF*, a seleção dos modelos e o manejo de possíveis desequilíbrios entre as classes. A implementação foi realizada utilizando as bibliotecas `NumPy` [Walt et al. 2011], `Pandas` [McKinney 2010] e `scikit-learn` [Pedregosa et al. 2011].

3.1. Pré-processamento e Vetorização com TF-IDF

Antes de treinar os modelos, foi feita a etapa de pré-processamento nos dados de texto. A principal técnica utilizada para transformar os textos em uma representação numérica adequada ao modelo foi a *TF-IDF* (Term Frequency-Inverse Document Frequency). Essa técnica calcula um peso para cada termo com base em duas medidas: a frequência do termo no documento (*Term Frequency*) e a raridade do termo no conjunto de dados (*Inverse Document Frequency*). Dessa forma, palavras mais relevantes para o conteúdo do documento recebem maior peso, enquanto termos muito comuns, mas pouco informativos, como artigos, preposições e conjunções (*stopwords*), têm seu impacto reduzido.

A transformação *TF-IDF* foi aplicada sobre a junção das colunas `title` e `text`, enquanto a coluna `subject` foi descartada, uma vez que poderia fornecer pistas diretas sobre a veracidade da notícia e introduzir viés nos modelos. Foram utilizados como parâmetros `max_features=100`, limitando a representação vetorial às 100 palavras mais relevantes do corpus, e *stopwords* em inglês, compatíveis com o idioma do dataset. O resultado foi uma representação vetorial densa que serviu como entrada para os modelos de classificação.

3.2. Tratamento da coluna *Target*

Além disso, na coluna *target*, as notícias falsas foram codificadas com o valor 1, enquanto as notícias verdadeiras receberam o valor 0. A utilização de valores numéricos para representar as classes é necessária para que os algoritmos de aprendizado de máquina possam processar e interpretar corretamente os dados categóricos, uma vez que a maioria dos modelos de classificação exige entradas numéricas para realizar cálculos de similaridade, probabilidades ou otimização de funções de perda.

3.3. Treinamento dos modelos

Para a classificação de notícias como verdadeiras ou falsas, foram utilizados três algoritmos de aprendizado supervisionado: *Multi-Layer Perceptron* (MLP), *Random Forest* e *Multinomial Naive Bayes* [Scikit-learn Developers]. Estes modelos foram implementados utilizando a biblioteca *scikit-learn* [Pedregosa et al. 2011], que fornece classes específicas para cada algoritmo:

- **MLPClassifier**: classe para redes neurais do tipo *feedforward* com uma ou mais camadas ocultas, capaz de aprender padrões complexos a partir de dados tabulares ou vetorizados;
- **RandomForestClassifier**: classe para construção de florestas aleatórias de árvores de decisão, combinando múltiplas árvores independentes para melhorar a generalização e reduzir o risco de sobreajuste;
- **MultinomialNB**: classe para o modelo Naive Bayes multinomial, adequada para dados discretos, especialmente vetores de contagem ou representações TF-IDF de texto.

Cada modelo passou por um processo de otimização de hiperparâmetros utilizando *Grid Search* com validação cruzada de 5 folds (*GridSearchCV*) [Pedregosa et al. 2011]. O *GridSearchCV* é uma ferramenta do *scikit-learn* que permite testar automaticamente múltiplas combinações de parâmetros, avaliando o desempenho do modelo em cada uma delas de forma consistente por meio de *k-fold cross-validation*. Essa abordagem garante que a seleção de hiperparâmetros não dependa apenas de uma divisão específica dos dados, aumentando a robustez da escolha final.

O *Multinomial Naive Bayes* foi incluído por ser particularmente eficiente para dados de texto representados como vetores de contagem ou TF-IDF [Scikit-learn Developers]. Este modelo assume independência entre as características e segue uma distribuição multinomial, permitindo lidar eficientemente com a alta dimensionalidade típica de textos, além de apresentar baixo custo computacional e fornecer resultados robustos mesmo em conjuntos de dados relativamente pequenos.

As grades de hiperparâmetros testadas incluíram, para o MLP:

- **hidden_layer_sizes** $(50, 25), (100, 50), (150, 75)$: define o número de neurônios em cada camada oculta da rede, controlando sua capacidade de aprendizado;
- **learning_rate_init** $0.01, 0.1, 0.25$: taxa de aprendizado inicial, que determina a velocidade de atualização dos pesos durante o treinamento;
- **activation** (*relu*): função de ativação que introduz não-linearidades no modelo;
- **alpha** 0.0001: parâmetro de regularização L2, evitando sobreajuste penalizando pesos muito grandes;
- **max_iter** 120: número máximo de iterações permitidas para o treinamento.

Para a Random Forest, os hiperparâmetros avaliados foram:

- **n_estimators** $100, 200$: número de árvores na floresta, aumentando a robustez do modelo;
- **max_depth** (None, 20): profundidade máxima das árvores, controlando o detalhamento das divisões;
- **min_samples_leaf** $1, 2$: número mínimo de amostras por folha, evitando que as árvores se tornem excessivamente específicas.

No Naive Bayes, foi testado o parâmetro:

- **alpha** $0.1, 1.0$: parâmetro de suavização Laplace, que evita probabilidades zero para palavras não observadas no treinamento.

O desempenho de cada configuração foi avaliado utilizando a acurácia nos dados de treino e validação, permitindo comparar o comportamento dos modelos e identificar as melhores combinações de hiperparâmetros. A visualização das métricas por meio de gráficos mostrou a acurácia média para cada configuração testada, facilitando a análise do desempenho. Ao final, os melhores modelos, com suas respectivas combinações de parâmetros otimizadas, foram selecionados para a avaliação final, incluindo cálculo de precisão, recall e F1-score.

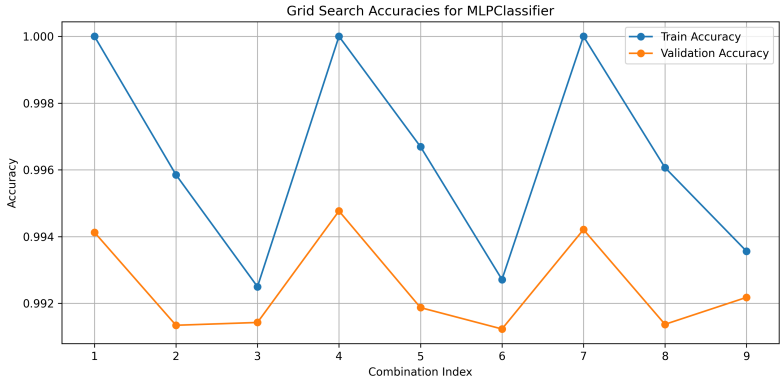

Figura 2. Evolução da acurácia dado a combinação de hiperparâmetros *Multi-Layer Perceptron*

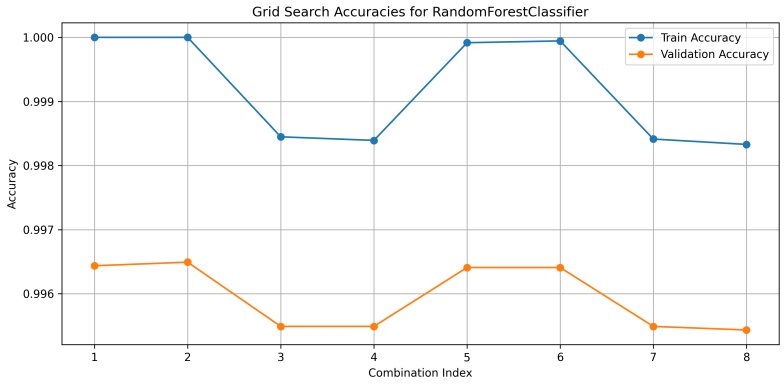

Figura 3. Evolução da acurácia dado a combinação de hiperparâmetros *Random Forest*

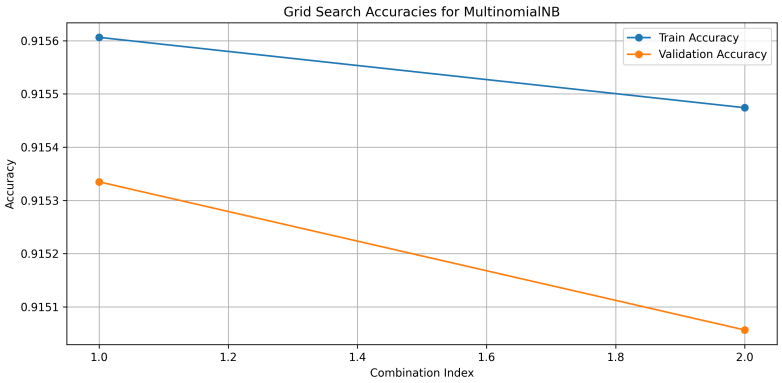

Figura 4. Evolução da acurácia dado a combinação de hiperparâmetros *Naive Bayes*

4. Resultados

Os modelos treinados foram avaliados utilizando 20% do conjunto de dados, reservado como conjunto de teste, considerando acurácia, precisão, recall e *F1-score*. As métricas obtidas para cada modelo estão detalhadas na Tabela 1.

O *Multi-Layer Perceptron* (MLP) apresentou uma acurácia de 0,9954 no conjunto de teste. A precisão para notícias reais foi de 0,9956 e para notícias falsas de 0,9953; o recall foi de 0,9949 para notícias reais e 0,9960 para notícias falsas. O *F1-score* médio ponderado foi de 0,9954, indicando excelente desempenho na classificação das duas classes. A matriz de confusão correspondente é apresentada na Figura 5.

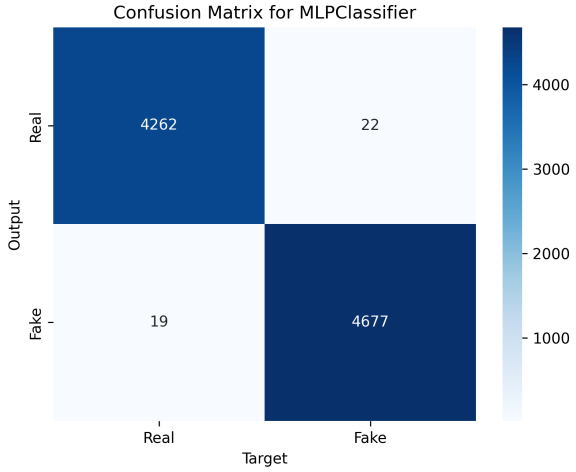

Figura 5. Matriz de confusão do modelo MLPClassifier

O *Random Forest* obteve a melhor performance, com acurácia de 0,9970. A precisão foi de 0,9960 para notícias reais e 0,9979 para notícias falsas, enquanto o recall foi de 0,9977 para notícias reais e 0,9964 para notícias falsas. O *F1-score* médio ponderado foi de 0,9970, demonstrando alta eficácia na classificação. A matriz de confusão correspondente é apresentada na Figura 6.

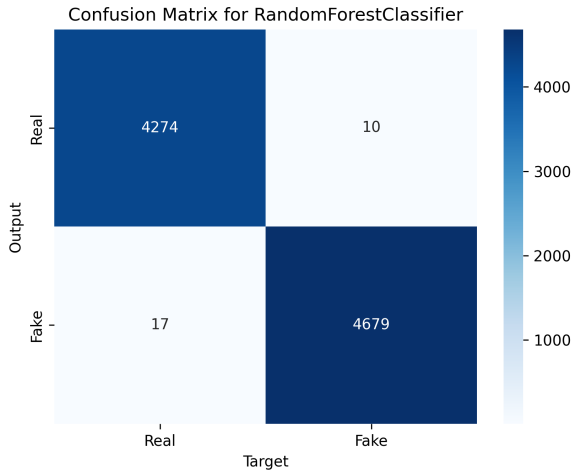

Figura 6. Matriz de confusão do modelo RandomForestClassifier

O *Multinomial Naive Bayes* apresentou desempenho inferior, com acurácia de 0,9161. A precisão para notícias reais foi de 0,8942 e para notícias falsas de 0,9380; o recall foi de 0,9349 para notícias reais e 0,8991 para notícias falsas. O *F1-score* médio ponderado foi de 0,9162, indicando que, embora o modelo consiga capturar boa parte dos padrões do texto, ele é menos eficaz que os modelos baseados em árvores ou redes neurais para esta tarefa. A matriz de confusão correspondente é apresentada na Figura 7.

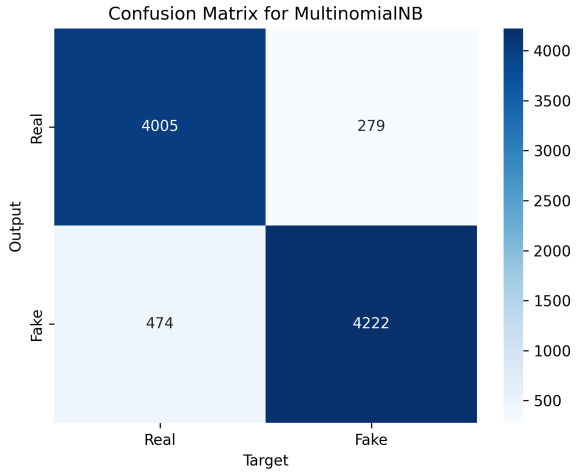

Figura 7. Matriz de confusão do modelo MultinomialNB

A comparação das métricas principais (acurácia, precisão e recall) dos três modelos está apresentada na Figura 8, e os detalhes completos por classe podem ser consultados na Tabela 1. Observa-se que tanto MLP quanto Random Forest apresentam desempenho elevado e consistente, enquanto o Naive Bayes é significativamente mais baixo, evidenciando que modelos com maior capacidade de aprendizado de padrões complexos se beneficiam da representação TF-IDF utilizada.

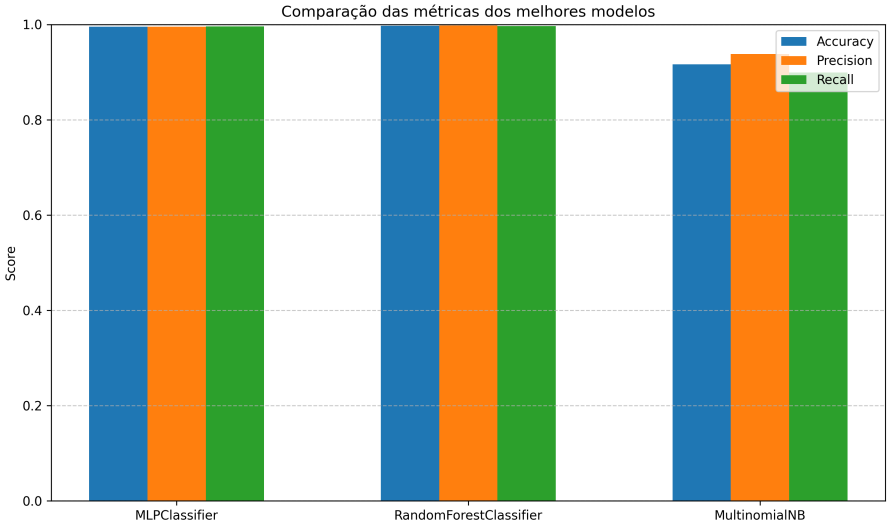

Figura 8. Comparação das métricas (acurácia, precisão e recall) dos melhores modelos

Tabela 1. Desempenho detalhado dos modelos no conjunto de teste

Modelo	Classe	Precisão	Recall	F1-score	Acurácia
MLPClassifier	Real	0,9956	0,9949	0,9952	0,9954
	Fake	0,9953	0,9960	0,9956	
RandomForestClassifier	Real	0,9960	0,9977	0,9969	0,9970
	Fake	0,9979	0,9964	0,9971	
MultinomialNB	Real	0,8942	0,9349	0,9141	0,9161
	Fake	0,9380	0,8991	0,9181	

5. Conclusão

6. Conclusão

Neste trabalho, classificamos notícias como verdadeiras ou falsas utilizando MLP, Random Forest e Multinomial Naive Bayes, com pré-processamento baseado em TF-IDF e otimização de hiperparâmetros via *GridSearchCV*.

Os modelos foram avaliados em 20% do conjunto de dados reservado para teste. MLP e Random Forest apresentaram excelente desempenho, com acurácias de 0,9954 e 0,9970, respectivamente, enquanto o Naive Bayes apresentou desempenho inferior (0,9161).

Os resultados mostram que modelos com maior capacidade de aprendizado de padrões complexos se beneficiam da representação TF-IDF, sendo o Random Forest ligeiramente superior para esta tarefa. Estes achados destacam a importância da escolha do modelo e da otimização de hiperparâmetros na detecção automática de notícias falsas.

Referências

[Emineyetm 2022] Emineyetm (2022). Fake news detection datasets. Disponível em: Kaggle. Acessado em: 19 out. 2025.

[Masciari et al. 2019] Masciari, E., Moscato, V., Picariello, A., and Sperlì, G. (2019). Leveraging machine learning for fake news detection. In *Proceedings of the International Conference on Advanced Computational Intelligence*.

[McKinney 2010] McKinney, W. (2010). Data structures for statistical computing in python. *Proceedings of the 9th Python in Science Conference*, pages 56–61.

[Pedregosa et al. 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., and Thirion, B. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830.

[Scikit-learn Developers] Scikit-learn Developers. sklearn.naive_bayes.multinomialnb. `https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html`. Acesso em: 25 out. 2025.

[Sudhakar and Kaliyamurthie 2018] Sudhakar, M. and Kaliyamurthie, K. (2018). Detection of fake news from social media using support vector machine learning algorithms. *International Journal of Pure and Applied Mathematics*, 118(20):1–9.

[Vosoughi et al. 2018] Vosoughi, S., Roy, D., and Aral, S. (2018). The spread of true and false news online. *Science*, 359(6380):1146–1151.

[Walt et al. 2011] Walt, S. v. d., Colbert, S. C., and Varoquaux, G. (2011). The numpy array: A structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30.