Detecção de Fake News: Comparação de Modelos Clássicos e Deep Learning

Eric Ribeiro Alves

[a] *Universidade Estadual Paulista (UNESP), Engenharia de Controle e Automação, Sorocaba, 18087-180, SP, Brasil*

Abstract

Este trabalho aborda a detecção de *fake news* por meio da análise de dados textuais, utilizando o conjunto de dados ISOT Fake News Dataset. Quatro modelos foram implementados: *Logistic Regression*, *Decision Trees*, *BERT* e uma arquitetura combinada de *Passive Aggressive Classifier* com *Deep Neural Network (DNN)*. O modelo BERT foi incorporado para capturar dependências semânticas complexas. O desempenho dos modelos foi avaliado utilizando as métricas de acurácia, *F1-Score*, e *ROC-AUC*, permitindo uma comparação robusta entre abordagens clássicas e de aprendizado profundo. Os resultados destacam a diferença e eficácia dos modelos baseados em aprendizado profundo e aprendizado de maquina, tendo como linha exploratória a para a tarefa proposta.

Keywords: Fake News, Machine Learning, Deep Learning, Comparação de Modelos

1. Introdução

O crescimento exponencial da disseminação de *fake news* é um problema crítico para a sociedade moderna. A identificação automatizada dessas notícias tornou-se essencial, dada a escala e a velocidade da propagação. Este trabalho investiga métodos para classificação de *fake news* utilizando o ISOT Fake News Dataset, amplamente adotado em estudos de detecção de notícias falsas.

Para abordar este desafio, foram explorados diferentes modelos de aprendizado de máquina e aprendizado profundo, avaliando suas performances com base em métricas padronizadas. A principal contribuição deste trabalho é a análise comparativa entre abordagens tradicionais, como regressão logística e

árvores de decisão, e modelos modernos baseados em aprendizado profundo, como BERT e uma combinação de *Passive Aggressive Classifier* com redes neurais profundas (*DNN*).

2. Análise de Dados

O conjunto de dados ISOT Fake News Dataset é uma coleção que contém artigos classificados em duas categorias: notícias falsas e notícias verdadeiras. Esses dados foram obtidos a partir de fontes reais, como o site Reuters.com para as notícias verdadeiras, e fontes de notícias falsas identificadas e marcadas pelo Politifact e pela Wikipedia, fontes reconhecidas por sua verificação de fatos. O dataset apresenta uma variedade de artigos de diferentes tópicos, com uma predominância de notícias sobre política e eventos mundiais.

A estrutura do dataset é composta por dois arquivos CSV: o arquivo `True.csv`, com mais de 12.600 artigos provenientes de Reuters.com, e o arquivo `Fake.csv`, que também contém mais de 12.600 artigos de sites de notícias falsas. Cada artigo inclui informações como título, texto, tipo (verdadeiro ou falso) e a data de publicação. A coleta de dados foi focada principalmente em artigos de 2016 a 2017, a fim de alinhar-se com um dataset similar disponibilizado no Kaggle. A limpeza dos dados foi realizada, porém, as pontuações e erros encontrados nas notícias falsas foram mantidos no texto.

Os artigos estão divididos em várias categorias, com as seguintes quantidades de artigos por categoria:

Tipo de Notícia	**Categoria**	**Quantidade de Artigos**
Notícias Verdadeiras	World-News	10.145
	Politics-News	11.272
Notícias Falsas	Government-News	1.570
	Middle-East	778
	US News	783
	Left-News	4.459
	Politics	6.841
	News	9.050

Table 1: Distribuição de Artigos por Tipo e Categoria

Este conjunto de dados é fundamental para a classificação de notícias, sendo utilizado em diversos modelos de aprendizado de máquina para a de-

tecção de notícias falsas. A seguir, detalharemos as etapas de pré-processamento, análise exploratória e a aplicação de diferentes técnicas de classificação.

2.1. Pré-Processamento de Dados

O pré-processamento do conjunto de dados é uma etapa crucial para preparar as informações para o treinamento de modelos de aprendizado de máquina. As principais etapas realizadas incluem:

2.1.1. Limpeza de Texto

Os textos dos artigos foram submetidos a um processo de limpeza que envolveu a remoção de caracteres especiais, links, números e palavras irrelevantes, também conhecidas como *stopwords* (artigos, conjuntivos, etc). Este processo visa deixar o texto mais simples e eficiente para o modelo, eliminando elementos que não agregam valor na tarefa de classificação.

2.1.2. Tokenização e Embeddings

Para converter os textos em uma forma que pudesse ser processada por modelos de aprendizado, foi aplicada a tokenização, que transforma os textos em sequências de tokens (geralmente palavras ou subpalavras). A tokenização pode ser feita de várias maneiras, como por palavras, subpalavras ou até caracteres, dependendo da tarefa. Para os modelos clássicos, como a regressão logística e a árvore de decisão, foi utilizada a técnica de **TF-IDF (Term Frequency-Inverse Document Frequency)**, que transforma os textos em representações vetoriais baseadas na frequência dos termos nos documentos. A técnica é composta por duas métricas principais: **Term Frequency (TF)** e **Inverse Document Frequency (IDF)**.

A fórmula do **TF** é dada por:

$$\text{TF}(t, d) = \frac{\text{Número de vezes que o termo } t \text{ aparece no documento } d}{\text{Total de termos no documento } d}$$

E a fórmula do **IDF** é:

$$\text{IDF}(t) = \log\left(\frac{N}{\text{Número de documentos que contêm o termo } t}\right)$$

onde N é o número total de documentos no corpus. Combinando essas duas métricas, obtemos o **TF-IDF**:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t)$$

Já para o modelo BERT, utilizaram-se *embeddings* próprios, que são representações vetoriais densas que capturam o significado semântico das palavras no contexto do texto. Em vez de depender apenas da frequência de palavras, como o TF-IDF, os embeddings são mais eficazes para tarefas complexas, como a detecção de fake news, pois compreendem melhor as relações e o contexto em que as palavras são usadas. O modelo BERT utiliza embeddings contextuais, ou seja, as representações das palavras variam de acordo com as palavras ao seu redor no texto.

2.2. Modelagem e Avaliação

Após o pré-processamento, o conjunto de dados foi dividido em conjuntos de treino e teste (70%/30%). Diversos modelos foram treinados e avaliados utilizando diferentes abordagens, incluindo **regressão logística**, **árvore de decisão**, **BERT**, **Passive-Aggressive Classifier (PAC)** e **Deep Neural Networks (DNN)**. Cada modelo foi ajustado com base nas características do conjunto de dados e nas especificidades do processo de treinamento.

- **Regressão Logística:** Este modelo foi treinado após a transformação dos textos utilizando a técnica de **TF-IDF**, que representa os documentos como vetores numéricos baseados na frequência dos termos e na importância de cada termo no contexto do corpus. A regressão logística, um modelo linear de classificação, calcula a probabilidade de um dado texto pertencer a uma classe específica (fake ou real) com base nas características extraídas dos dados. A fórmula para a regressão logística é dada por:

$$p(y = 1|x) = \sigma(w^T x + b)$$

onde $\sigma(z) = \frac{1}{1+e^{-z}}$ é a função sigmoide, w é o vetor de pesos, x é o vetor de características (nesse caso, o vetor TF-IDF), e b é o viés. O modelo é treinado para minimizar a função de perda logarítmica (log-loss) entre a saída predita e a verdadeira:

$$L = -\sum_{i=1}^{n} y_i \log(p_i) + (1 - y_i) \log(1 - p_i)$$

A principal limitação da regressão logística é que ela não captura relações não-lineares entre as variáveis, mas ela é eficiente para problemas simples e pode servir como um bom ponto de partida para comparação com modelos mais complexos.

- **Árvore de Decisão:** A árvore de decisão foi treinada de forma semelhante ao modelo de regressão logística, com a transformação de texto em vetores TF-IDF. A árvore de decisão utiliza uma estrutura hierárquica, onde cada nó representa uma decisão baseada em uma característica (neste caso, uma palavra ou combinação de palavras), dividindo recursivamente os dados em subsets cada vez mais homogêneos. A função de perda utilizada para dividir os dados é geralmente a *Gini Impurity* ou a *Entropia*, e é dada por:

$$Gini(D) = 1 - \sum_{i=1}^{c} p_i^2$$

onde p_i é a proporção de exemplos da classe i no conjunto D, e c é o número de classes. A árvore continua a dividir os dados até que a impureza ou a entropia seja mínima.

- **BERT (Bidirectional Encoder Representations from Transformers):** O BERT é um modelo pré-treinado de aprendizado profundo baseado na arquitetura Transformer. Ele foi projetado para entender o contexto de uma palavra em um texto considerando tanto as palavras à esquerda quanto à direita (bidirecional). A tokenização foi feita utilizando o `BertTokenizer`, que converte o texto em tokens específicos para o modelo BERT. O funcionamento do BERT é baseado no mecanismo de atenção, cuja fórmula é dada por:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

onde Q, K, e V representam as matrizes de consulta, chave e valor, respectivamente, e d_k é a dimensionalidade das chaves. A função de atenção permite ao modelo capturar relações contextuais entre palavras em uma frase, o que é essencial para tarefas de NLP, como a detecção de fake news.

Durante o treinamento, o modelo ajusta os pesos das representações vetoriais (embeddings) para minimizar o erro na previsão das classes. O BERT realiza a pré-processamento de dados através de um embedding contextual, onde cada token é transformado em um vetor denso que representa o significado da palavra em seu contexto específico dentro da frase.

- **Passive-Aggressive Classifier (PAC):** O PAC é um modelo de aprendizado online, eficiente para tarefas de classificação em tempo real. Ele é chamado de "passivo-agressivo" porque, durante o treinamento, o modelo tenta ser o mais conservador possível (passivo) quando está correto, mas ajusta rapidamente seus parâmetros (agressivo) quando comete um erro. O PAC é treinado com o objetivo de minimizar a margem de erro. A atualização dos parâmetros w para uma amostra errada é dada por:

  $$w_{t+1} = w_t + \eta y_t x_t$$

  onde η é a taxa de aprendizado, y_t é o rótulo da amostra, x_t é o vetor de características e w_t é o vetor de pesos no tempo t.

- **Deep Neural Networks (DNN):** As redes neurais profundas foram configuradas com múltiplas camadas ocultas, permitindo a captura de representações não-lineares dos dados. Cada camada da rede é composta por múltiplos neurônios, onde cada neurônio aplica uma função de ativação f (geralmente ReLU ou sigmoide) para transformar a entrada antes de passá-la para a próxima camada. A fórmula geral de um neurônio é dada por:

  $$a = f(Wx + b)$$

  onde W são os pesos, x é a entrada (ou a saída da camada anterior), b é o viés e a é a ativação. Durante o treinamento, o modelo ajusta os pesos da rede por meio do processo de retropropagação, minimizando a função de perda L, que pode ser a função de erro quadrático médio (MSE) ou a entropia cruzada, dependendo da tarefa. A retropropagação é dada por:

  $$\Delta w = -\eta \frac{\partial L}{\partial w}$$

  onde η é a taxa de aprendizado e $\frac{\partial L}{\partial w}$ é o gradiente da função de perda em relação aos pesos.

Esses modelos foram avaliados com base em métricas de desempenho como precisão, F1-score e ROC-AUC, permitindo uma comparação direta entre as abordagens. As métricas quantitativas fornecem uma visão clara da eficácia de cada modelo na tarefa de classificação de fake news, ajudando a identificar o modelo mais adequado para o conjunto de dados ISOT.

3. Modelos Propostos

Esta seção descreve os modelos utilizados para a classificação de fake news, detalhando suas características e como foram adaptados ao problema. Cada modelo foi escolhido com base em sua capacidade de lidar com diferentes aspectos dos dados textuais, como linearidade, hierarquia, relações contextuais e padrões complexos.

3.1. Logistic Regression

O modelo de regressão logística é uma abordagem linear amplamente utilizada em problemas de classificação binária. Ele opera calculando uma probabilidade associada a cada classe com base nas características de entrada, utilizando a função sigmoide como ativação. Por ser um modelo linear, ele é eficiente para problemas de alta dimensionalidade, especialmente quando combinado com métodos de vetorização como TF-IDF (*Term Frequency-Inverse Document Frequency*), que transforma os textos em representações numéricas. A regularização é incorporada ao modelo para evitar problemas de sobreajuste, tornando-o robusto mesmo em datasets ruidosos.

3.2. Decision Trees

As árvores de decisão são modelos que particionam recursivamente os dados com base em regras de decisão simples, criando uma estrutura hierárquica que facilita a interpretação. Cada nó da árvore representa uma condição baseada em uma característica específica, enquanto os ramos correspondem aos resultados possíveis dessa condição. Para este problema, foi configurada uma profundidade máxima, o que limita a complexidade da árvore e reduz o risco de sobreajuste. Esse modelo é particularmente eficaz em capturar relações não lineares entre as características, tornando-o adequado para datasets onde as classes possuem padrões claramente separados.

3.3. Passive Aggressive Classifier com Deep Neural Network (DNN)

Esta abordagem híbrida combina o Passive Aggressive Classifier (PAC), um algoritmo projetado para lidar com dados de fluxo contínuo, com uma Rede Neural Profunda (DNN) para capturar padrões mais complexos nos dados.

O PAC é um classificador linear eficiente que atualiza os pesos apenas quando ocorre um erro de classificação, garantindo rapidez e adaptabilidade a mudanças nos dados. No entanto, por si só, ele pode ser limitado na

identificação de padrões mais sutis. Para superar essa limitação, as previsões do PAC são combinadas com representações vetorizadas dos textos (obtidas pelo TF-IDF) e utilizadas como entrada para a DNN.

A DNN é composta por múltiplas camadas densas (*fully connected layers*) que aprendem representações hierárquicas dos dados. Camadas intermediárias com funções de ativação não lineares ($ReLU$) e técnicas como *dropout* foram incorporadas para reduzir o risco de sobreajuste e melhorar a capacidade de generalização. A camada de saída utiliza a função sigmoide, adequada para classificação binária. Essa integração entre PAC e DNN permite explorar tanto a eficiência do PAC quanto a flexibilidade da DNN para identificar padrões complexos.

3.4. BERT (Bidirectional Encoder Representations from Transformers)

O modelo BERT é uma arquitetura baseada em *Transformers*, projetada para capturar relações contextuais profundas em sequências de texto. Diferentemente de abordagens tradicionais, o BERT utiliza um mecanismo de atenção bidirecional, o que significa que ele considera o contexto tanto antes quanto depois de cada palavra em uma frase, proporcionando uma compreensão mais rica e precisa do texto.

No pré-processamento, os textos são tokenizados em subtokens utilizando o *WordPiece Tokenizer*, e cada subtoken é mapeado para uma representação numérica densa em um espaço de alta dimensão. Essas representações são passadas através de várias camadas transformadoras, onde o mecanismo de autoatenção modela as interações entre as palavras em todo o texto.

O modelo foi adaptado para o problema de classificação binária ao incluir uma camada densa na saída, responsável por prever as classes de fake news. O treinamento foi realizado com um otimizador baseado no algoritmo Adam e uma taxa de aprendizado ajustada para preservar o conhecimento pré-treinado do BERT enquanto adapta o modelo às características específicas do dataset. Essa abordagem é particularmente eficaz para capturar nuances linguísticas e identificar padrões semânticos complexos nos textos.

4. Avaliação de Desempenho

Para comparar a performance dos modelos, foram utilizadas as seguintes métricas de avaliação. Cada uma delas mede diferentes aspectos da eficácia dos modelos, proporcionando uma visão abrangente sobre sua capacidade de realizar a tarefa de classificação de fake news.

- **Acurácia**: A acurácia é uma métrica fundamental para problemas de classificação. Ela calcula a proporção de previsões corretas (tanto positivas quanto negativas) sobre o total de previsões feitas. A fórmula da acurácia é dada por:

$$\text{Acurácia} = \frac{TP + TN}{TP + TN + FP + FN}$$

onde:

  - TP (True Positives) representa o número de exemplos classificados corretamente como positivos.
  - TN (True Negatives) representa o número de exemplos classificados corretamente como negativos.
  - FP (False Positives) representa o número de exemplos classificados incorretamente como positivos.
  - FN (False Negatives) representa o número de exemplos classificados incorretamente como negativos.

Embora a acurácia seja uma métrica simples, ela pode ser enganosa em conjuntos de dados desbalanceados, onde as classes não são igualmente representadas.

- **F1-Score**: O F1-score é a média harmônica entre a precisão e a revocação (recall), proporcionando uma medida balanceada que leva em consideração tanto os falsos positivos quanto os falsos negativos. A fórmula do F1-score é:

$$\text{F1-score} = 2 \cdot \frac{\text{Precisão} \cdot \text{Revocação}}{\text{Precisão} + \text{Revocação}}$$

onde:

$$\text{Precisão} = \frac{TP}{TP + FP}, \quad \text{Revocação} = \frac{TP}{TP + FN}$$

O F1-score é particularmente útil em problemas de classificação desbalanceada, onde uma simples acurácia pode não refletir adequadamente a performance do modelo.

- **ROC-AUC (Receiver Operating Characteristic - Area Under the Curve)**: A curva ROC é uma representação gráfica da performance de um modelo de classificação binária em diferentes limiares de

decisão. A AUC (Área Sob a Curva) quantifica a capacidade do modelo em distinguir entre as classes. A fórmula da AUC é dada por:

$$\text{AUC} = \int_0^1 \text{TPR}(x)\, d(\text{FPR}(x))$$

onde:

- TPR (True Positive Rate) é a taxa de verdadeiros positivos, também conhecida como revocação, e é dada por $\text{TPR} = \frac{TP}{TP+FN}$.
- FPR (False Positive Rate) é a taxa de falsos positivos, dada por $\text{FPR} = \frac{FP}{FP+TN}$.

A AUC varia entre 0 e 1, com 1 indicando que o modelo consegue distinguir perfeitamente entre as classes, e 0,5 indicando que o modelo está apenas fazendo classificações aleatórias. A AUC é uma métrica especialmente útil quando as classes são desbalanceadas, pois avalia o desempenho do modelo em todos os limiares de decisão possíveis.

Essas métricas forneceram uma avaliação completa do desempenho dos modelos, permitindo uma análise detalhada sobre sua capacidade de realizar a tarefa de classificação de fake news no conjunto de dados ISOT. A acurácia foi útil para avaliar a taxa geral de classificações corretas, o F1-score ofereceu uma visão equilibrada entre precisão e revocação, e a AUC permitiu avaliar a capacidade discriminativa dos modelos em diferentes limiares.

5. Resultados

Os resultados experimentais são apresentados na Tabela 2. O modelo BERT obteve o melhor desempenho geral, seguido pela abordagem combinada de *Passive Aggressive Classifier* com *DNN*.

Table 2: Desempenho dos Modelos na Detecção de Fake News

Modelo	**Acurácia**	**F1-Score**	**ROC-AUC**
Logistic Regression	85.2%	84.5%	0.86
Decision Trees	81.9%	80.7%	0.82
Passive Aggressive + DNN	91.3%	90.8%	0.92
BERT	94.7%	94.4%	0.96

6. Conclusão

Os resultados demonstram que os modelos baseados em aprendizado profundo, particularmente o BERT, apresentam desempenho superior na detecção de *fake news*, especialmente em tarefas onde o contexto semântico desempenha papel fundamental. Modelos clássicos, como regressão logística e árvores de decisão, continuam úteis para cenários com recursos computacionais limitados.

Como trabalhos futuros, planeja-se explorar técnicas de explicabilidade para os modelos baseados em *deep learning*, com o objetivo de tornar as decisões mais transparentes.

References

[1] Leslie Lamport, *LaTeX: a document preparation system*, Addison Wesley, Massachusetts, 2nd edition, 1994.