P.PORTO

ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

# Sistema de Deteção de Fake News: Abordagem Multi-Modelo

Trabalho Prático - Avaliação Contínua

Mestrado em Engenharia Informática

**Grupo 4**

Diogo Pereira – 8200594

Hugo Guimarães – 8220337

15 de janeiro de 2026

# Conteúdo

# Índice de Figuras

# Índice de Tabelas

# 1 Introdução

## 1.1 Contextualização

A democratização do acesso à Internet e a ascensão das redes sociais alteraram drasticamente o paradigma de produção e consumo de informação. Se, por um lado, estas plataformas permitem uma disseminação rápida de conhecimento, por outro, tornaram-se canais privilegiados para a propagação de desinformação, comummente designada por *Fake News* [1].

Este fenómeno não é apenas um problema tecnológico, mas um desafio social complexo com consequências tangíveis. Estudos recentes demonstram como a desinformação tem sido utilizada como ferramenta de manipulação política, com impacto observado em processos eleitorais e na polarização da opinião pública [2]. Além disso, em contextos de crise, como a pandemia COVID-19, a disseminação de notícias falsas representou um risco direto para a saúde pública [3].

O grande desafio reside no volume massivo de dados gerados diariamente (Big Data), o que torna a verificação manual de factos (*fact-checking*) uma tarefa impossível de realizar em tempo útil [4] . Consequentemente, torna-se imperativo o desenvolvimento de sistemas automáticos baseados em *Machine Learning* (ML) capazes de detetar, classificar e mitigar a propagação de conteúdos falsos, analisando não apenas o texto, mas também o contexto, a postura (*stance*) e a consistência semântica das notícias.

## 1.2 Objetivos

O principal objetivo deste trabalho é o desenvolvimento de uma arquitetura de *Machine Learning* capaz de classificar a veracidade de artigos noticiosos. Este projeto visa integrar os conhecimentos adquiridos na Unidade Curricular, aplicando algoritmos de aprendizagem supervisionada e não supervisionada para a resolução de um problema real.

De acordo com os requisitos propostos no enunciado, foram definidos os seguintes objetivos específicos:

- Realizar uma análise exploratória de dados em múltiplos *datasets* para compreender padrões de desinformação;
- Implementar e comparar diferentes algoritmos de classificação para tarefas distintas:

classificação de tópicos, deteção de *stance*, identificação de *clickbait* e análise de anomalias;

- Aplicar técnicas de aprendizagem não supervisionada para a deteção de padrões anómalos em notícias reais;
- Desenvolver um *Meta-Classificador* (modelo final) que agregue as previsões dos modelos parcelares para uma decisão final robusta;
- Construir uma interface gráfica que permita a um utilizador testar o modelo treinado de forma interativa;
- Avaliar a performance da solução utilizando métricas adequadas (Accuracy, Precision, Recall e F1-Score).

### 1.3 Estrutura do Relatório

O presente relatório encontra-se organizado em 6 capítulos, refletindo o fluxo de trabalho desenvolvido:

- A Secção 2 apresenta a arquitetura do modelo de *Machine Learning* treinado;
- A Secção 3 descreve os *datasets* selecionados e o processo de análise exploratória e tratamento dos dados;
- A Secção 4 detalha a metodologia de desenvolvimento, justificando a arquitetura modular e a escolha dos algoritmos para cada tarefa específica;
- A Secção 5 ilustra a implementação da interface de utilização;
- Por fim, a Secção 6 sintetiza as conclusões do trabalho e aponta linhas para desenvolvimento futuro.

# 2 Arquitetura da Solução

A solução proposta baseia-se numa arquitetura hierárquica modular, seguindo uma estratégia de *Stacking Ensemble*. Ao contrário de abordagens monolíticas, este sistema decompõe o problema da deteção de *Fake News* em sub-tarefas especializadas, cujos resultados alimentam um decisor final.

A estrutura da *pipeline*, ilustrada na Figura 1, divide-se em três fases principais: Pré-processamento, Nível 1 (Especialistas) e Nível 2 (Meta-Classificação).

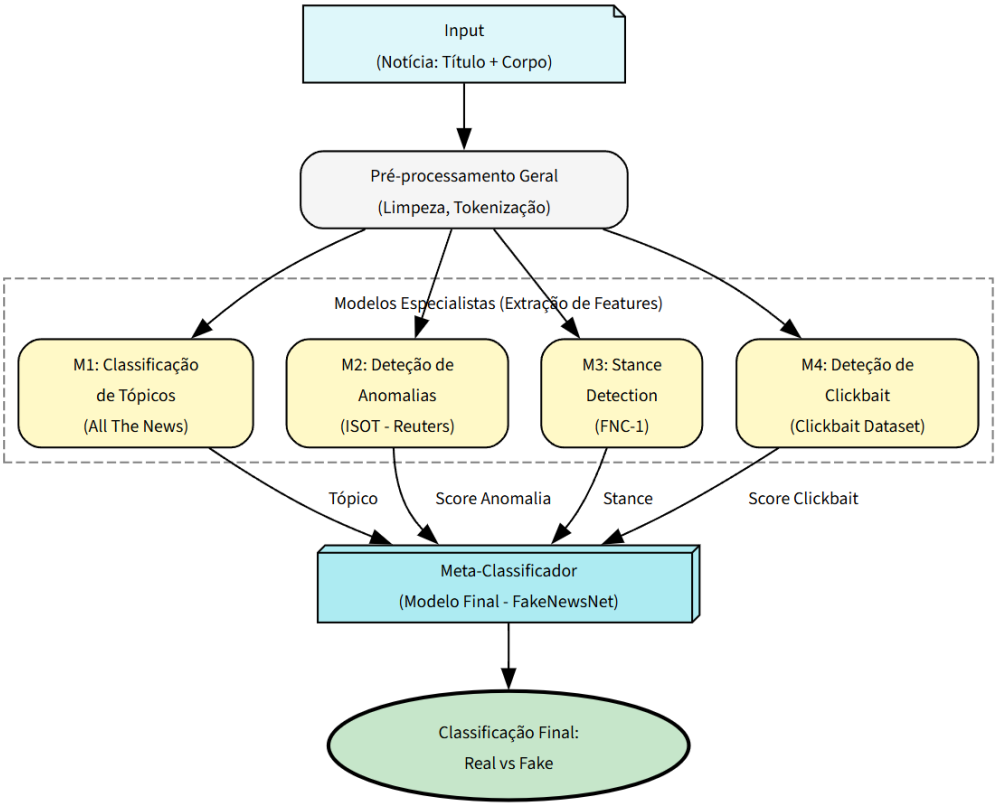


Figura 1: Diagrama da Arquitetura Hierárquica do Sistema

## 2.1 Fluxo de Dados e Processamento

**1. Entrada e Pré-processamento:** O sistema recebe como entrada o título e o corpo da notícia. Estes dados são submetidos a um processo de pré-processamento que executa a limpeza de texto (remoção de caracteres especiais, espaços em branco, lematização) e tokenização, preparando os dados para os modelos posteriores.

**2. Nível 1: Modelos Especialistas (Extração de *Features*):** Nesta camada, quatro modelos independentes analisam características distintas da notícia. Cada modelo foi

treinado num *dataset* específico para garantir especialização:

- **M1 - Classificação de Tópicos (All The News):** Identifica o contexto temático (ex: política, economia, saúde). O objetivo é fornecer contexto ao meta-classificador, visto que a linguagem de *fake news* varia consoante o tópico.
- **M2 - Deteção de Anomalias (ISOT - Reuters):** Analisa padrões estatísticos e linguísticos para detetar desvios da norma em notícias reais, gerando um *score* de anomalia.
- **M3 - *Stance Detection* (FNC-1):** Verifica a consistência entre o título e o corpo da notícia. Este modelo é crucial para detetar "títulos enganosos"onde o corpo da notícia não suporta a afirmação do título.
- **M4 - Deteção de *Clickbait* (Clickbait Dataset):** Avalia o sensacionalismo do título, atribuindo uma pontuação baseada em padrões de atração de cliques comuns em desinformação.

**3. Nível 2: Meta-Classificador:** As saídas dos quatro modelos especialistas (probabilidades, classes e *scores*) são concatenadas num vetor de *meta-features*. Este vetor serve de entrada para o Meta-Classificador (treinado no *dataset* WELFake).

Este modelo final aprende a ponderar a importância de cada especialista. Por exemplo, pode aprender que uma notícia com alto *score* de *clickbait* (M4) e inconsistência título-corpo (M3) tem uma probabilidade quase total de ser falsa, independentemente do tópico (M1). A saída final é a classificação binária: Real ou *Fake*.

# 3 Dados e Analise Exploratória dos Dados

## 3.1 Seleção dos Datasets

Para cumprir os objetivos do projeto e alimentar os diferentes modelos desenvolvidos, foi necessário recorrer a múltiplas fontes de dados. Como o sistema final depende de várias tarefas distintas (como detetar tópicos ou analisar títulos), não seria viável utilizar apenas um único dataset.

Abaixo apresenta-se a lista dos datasets escolhidos e a respetiva justificação:

- **All The News (Para Análise de Tópicos):**
  - *Origem:* Kaggle (David McKinley).
  - *Conteúdo:* Cerca de 143.000 artigos de publicações reais (ex: CNN, New York Times).
  - *Justificação:* Devido ao grande volume de notícias legítimas, é ideal para a análise de tópicos de notícias, permitindo o modelo aprender a classificar tópicos corretamente.
- **Fake News Challenge - FNC-1 (Para Stance Detection):**
  - *Origem:* Repositório oficial do desafio FNC-1.
  - *Conteúdo:* Pares de "Título"e "Corpo da Notícia"classificados quanto à concordância (concorda, discorda, discute, não relacionado).
  - *Justificação:* A maioria dos datasets não liga o título ao texto. Este foi escolhido especificamente para deteção de posição (*Stance*), pois permite treinar o algoritmo a perceber se o título está a mentir sobre o conteúdo do texto.
- **Clickbait Dataset (Para Deteção de Clickbait):**
  - *Origem:* Kaggle (Aman Anand Rai).
  - *Conteúdo:* Milhares de manchetes classificadas simplesmente como "Clickbait"ou "Não-Clickbait".
  - *Justificação:* Escolhido para a deteção de *clickbait* pois isola o problema do sensacionalismo. Permite que o sistema identifique títulos exagerados independentemente de a notícia ser falsa ou não.

- **ISOT Fake News Dataset (Para Análise de Anomalias):**
  - *Origem:* University of Victoria (ISOT Research Lab).
  - *Conteúdo:* Artigos verdadeiros (extraídos da Reuters) e artigos falsos (sinalizados pelo PolitiFact).
  - *Justificação:* Escolhido devido à qualidade da secção de notícias verdadeiras, provenientes da agência Reuters. Por serem textos com um padrão jornalístico rigoroso e consistente, constituem a base ideal para definir o que é uma notícia legítima e fiável.
- **WELFake (Para o Modelo Final):**
  - *Origem:* IEEE Transactions on Computational Social Systems (Verma et al., 2021).
  - *Conteúdo:* Uma agregação de quatro *datasets* (Kaggle, McIntire, Reuters e BuzzFeed Political), totalizando mais de 72.000 notícias com uma distribuição equilibrada entre reais e falsas.
  - *Justificação:* Foi selecionado por ser um dos maiores e mais diversos conjuntos de dados públicos. Ao combinar quatro fontes diferentes, evita o enviesamento típico de *datasets* singulares e previne o *overfitting*, garantindo que o Modelo Final é testado num cenário mais realista e genérico.

### 3.2 Análise e Tratamento de Dados

Previamente à fase de modelação, foi executado um rigoroso processo de tratamento e normalização de dados. Esta etapa é crítica para garantir a qualidade das *features* extraídas e, consequentemente, a performance dos algoritmos.

Foi realizada uma verificação preliminar da integridade dos dados em todos os *datasets*, não tendo sido detetados valores omissos (nulos) que comprometessem a análise. Relativamente a *outliers*, analisou-se a distribuição do comprimento dos textos, não se registando anomalias significativas (como textos vazios ou excessivamente longos resultantes de erros de recolha).

Embora cada módulo exija especificidades, definiu-se uma *pipeline* transversal de pré-processamento aplicado a todos os dados:

- **Divisão dos Dados:** Partição em conjuntos de treino (80%) e teste (20%), garantindo a representatividade das classes;
- **Normalização:** Conversão de todo o texto para minúsculas (*lowercase*) para reduzir a variabilidade do vocabulário;
- **Limpeza:** Remoção de sinais de pontuação e caracteres especiais;
- **Tokenização:** Segmentação do texto em unidades individuais (palavras/tokens);
- **Vetorização:** Conversão do texto para representação numérica (a técnica específica varia consoante o modelo, conforme detalhado abaixo).

Nas secções seguintes, detalham-se as estratégias de *Feature Engineering* específicas adotadas para cada tarefa.

### 3.2.1 Classificação de Tópicos (AllTheNews)

Este *dataset*, caracterizado por ter um elevado volume, exigiu estratégias focadas na eficiência computacional e escalabilidade:

- **Redução de Dimensionalidade (NMF):** Aplicação de *Non-Negative Matrix Factorization* para resumir o vasto vocabulário a um conjunto de tópicos principais, reduzindo o ruído e a complexidade do modelo;
- **Hashing Vectorization:** Em detrimento do tradicional TF-IDF (que armazena o vocabulário em memória), optou-se pela vetorização via *hashing*. Esta técnica torna o processamento mais rápido e "leve"para grandes volumes de dados textual.

### 3.2.2 Análise de Anomalias (ISOT)

O processamento deste conjunto de dados foi adaptado para suportar duas fases distintas de experimentação. Inicialmente, procedeu-se à fusão dos ficheiros originais (*True.csv* e *Fake.csv*) e à criação de uma coluna alvo (*label*) para identificar a veracidade.

Posteriormente, a amostragem dos dados variou consoante a estratégia de modelação:

- **Contaminação Controlada (Deteção de Anomalias):** Para os modelos não supervisionados, construiu-se um conjunto de treino composto pela totalidade das

notícias verdadeiras, contaminado deliberadamente com apenas 5% de notícias falsas. O objetivo foi simular se os algoritmos conseguiam isolar a minoria falsa como *outliers*;

- **Classificação Supervisionada:** Para os classificadores supervisionados (implementados após a falha da deteção de anomalias), utilizou-se a união completa de ambos os ficheiros, permitindo aos modelos aprender explicitamente as características distintivas de ambas as classes com a totalidade da informação disponível.

### 3.2.3 Análise de Stance (FNC-1)

O conjunto de dados FNC-1 apresenta a maior complexidade de pré-processamento, exigindo adaptações para capturar a relação semântica entre duas sequências de texto (Título e Corpo):

- **Tratamento de Stopwords:** Ao contrário da abordagem padrão, **não** foram removidas as *stop words*. Palavras de ligação e negação (ex: "not", "but", "however") são cruciais para inverter o sentido de uma frase e detetar desacordo (*disagree*);
- **Fusão de Atributos:** Criação de uma nova *feature* (*"combined_text"*) resultante da concatenação do título com o corpo da notícia;
- **Codificação de Variáveis:** Mapeamento das classes categóricas (*agree, disagree, discuss, unrelated*) para valores numéricos;
- **Gestão de Desbalanceamento:** Cálculo e aplicação de pesos às classes (*class weights*) durante o treino, de modo a mitigar o forte desequilíbrio entre a classe maioritária e as classes de concordância/discordância.

### 3.2.4 Análise de Clickbait (Clickbait Dataset)

Dada a natureza sintética e direta das manchetes presentes neste *dataset*, o pré-processamento foi mantido intencionalmente minimalista. A estrutura linguística dos títulos *clickbait* (ex: uso de imperativos, exageros) é capturada eficazmente pela *pipeline* padrão de tokenização e vetorização, não justificando engenharia de atributos adicional que pudesse introduzir ruído desnecessário.

# 4 Desenvolvimento dos Modelos (Metodologia)

O presente capítulo detalha a metodologia adotada para o desenvolvimento do sistema de deteção de *Fake News*. Dada a natureza multidimensional da desinformação, optou-se por uma arquitetura modular hierárquica (abordagem inspirada em *Stacking Ensemble*), em vez de um único modelo monolítico.

Para tal, foram desenvolvidos modelos especialistas independentes, treinados em *datasets* distintos, cujo objetivo é capturar diferentes nuances linguísticas e estruturais das notícias. As saídas probabilísticas destes modelos funcionam como *features* de alto nível (meta-features) para o classificador final.

A arquitetura proposta compreende os seguintes módulos:

- **Classificação de Tópicos:** Contextualização temática do artigo (ex: Política, Saúde, Tecnologia);
- **Análise de Anomalias:** Identificação de padrões nos textuais em notícias verdadeiras de modo a detetar anomalias;
- **Deteção de Stance (Postura):** Análise da concordância entre o título e o corpo da notícia;
- **Deteção de Clickbait:** Análise de padrões sensacionalistas nos títulos;
- **Meta-Classificador (Modelo Final):** Agregação das saídas anteriores para a previsão final de veracidade.

Nas subsecções seguintes, é descrito o ciclo de vida de desenvolvimento para cada um destes componentes, abrangendo desde o pré-processamento específico e engenharia de atributos (*Feature Engineering*), até à justificação da escolha dos algoritmos.

## 4.1 Modelo 1: Classificação de Tópicos

Inicialmente, a abordagem explorada consistiu numa *pipeline* simples de *Clustering* baseada em TF-IDF (*Term Frequency-Inverse Document Frequency*), seguindo o fluxo: *Pré-processamento* $\rightarrow$ *TF-IDF* $\rightarrow$ *Clustering*. Contudo, rapidamente tornou-se evidente a necessidade de reduzir a dimensionalidade da matriz resultante da vetorização para obter resultados mais robustos.

Numa segunda iteração, optou-se pela utilização de *Feature Hashing* (unsigned) como técnica de vetorização, devido à sua rapidez e otimização de memória em comparação com o TF-IDF tradicional. Para a redução de dimensionalidade, foram testadas e comparadas duas abordagens: LSA (*Latent Semantic Analysis*) e NMF (*Non-Negative Matrix Factorization*), resultando no fluxo: *Pré-processamento* → *Hashing* → *NMF/LSA* → *Clustering*.

A análise dos resultados desta segunda abordagem revelou uma segmentação ineficiente dos *clusters*. Frequentemente, os algoritmos identificavam apenas dois grandes grupos: um "Tópico A"muito específico (como desporto ou política) e um "Tópico B"que englobava o resto dos dados. Além disso, verificou-se que a otimização baseada no *Silhouette Score* não era ideal, visto ser uma métrica puramente geométrica que não captura necessariamente a coerência semântica dos tópicos gerados. A Tabela 1 resume os resultados quantitativos obtidos nestas experiências, comparando as diferentes configurações testadas.

Tabela 1: Comparação das métricas de qualidade obtidas nas diferentes abordagens testadas para classificação de tópicos.

| **Redução** | **Modelo** | **Métrica** | **Score** |
|---|---|---|---|
| LSA | KMeans | Silhouette | 0.3843 |
| | HDBSCAN | Silhouette | 0.4490 |
| | GMM | Silhouette | 0.5504 |
| NMF | KMeans | Silhouette | 0.5413 |
| | HDBSCAN | Silhouette | 0.3772 |
| | GMM | Silhouette | **0.5536** |
| *Modelagem de Tópicos (Sem redução)* | | | |
| – | NMF (TF-IDF) | Coerência | **0.7279** |

Consequentemente, a metodologia final evoluiu para o uso de NMF diretamente sobre TF-IDF como técnica de modelação de tópicos, em vez de apenas redução para clustering (*Pré-processamento* → *TF-IDF* → *NMF*). A escolha do NMF justifica-se pela sua capacidade de criar "clusters probabilísticos", onde um documento não pertence apenas a um grupo de forma binária, mas possui um peso de pertença (ex: Tópico A: 0.9, Tópico B: 0.1), o que é mais representativo da realidade das notícias.

Para a avaliação e otimização final, substituiu-se o critério geométrico (*Silhouette Score*) pelo *Coherence Score*, uma métrica mais indicada para avaliar a qualidade semântica e a interpretabilidade humana dos tópicos extraídos. A distribuição final dos tópicos, resultante desta abordagem otimizada, pode ser visualizada na Figura 2.

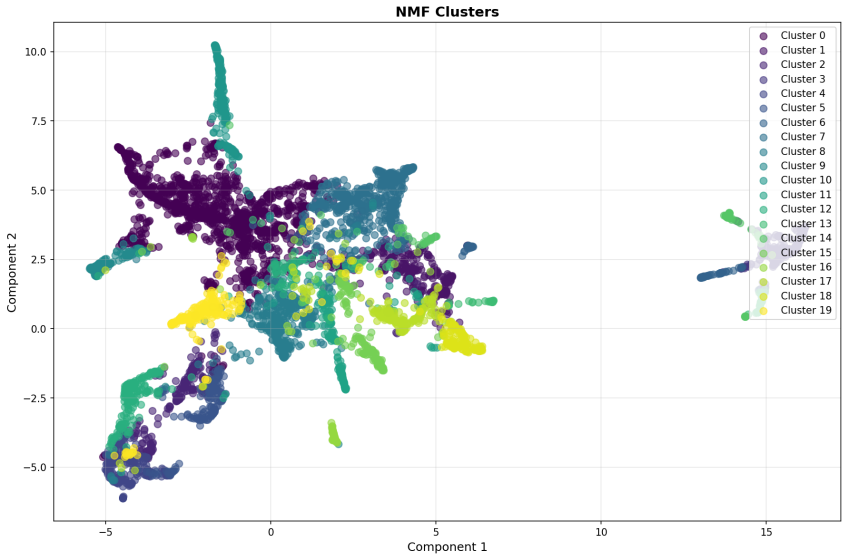


Figura 2: Visualização da distribuição dos tópicos obtidos com a abordagem final utilizando NMF.

### 4.2 Modelo 2: Análise de Anomalias

Para esta componente, formulou-se inicialmente a hipótese de que as notícias falsas constituiriam "anomalias"estatísticas, divergindo significativamente das notícias verdadeiras sem a necessidade de etiquetas explícitas (*Unsupervised Learning*).

Numa primeira fase experimental, foram aplicados algoritmos de deteção de anomalias (incluindo *Isolation Forest*, *One-Class SVM* e *Autoencoders*) no *dataset* ISOT. Contudo, os resultados refutaram a hipótese inicial: as notícias falsas partilhavam demasiadas semelhanças estruturais e vocabulares com as verdadeiras, resultando numa incapacidade dos modelos em separar as classes (F1-Scores inferiores a 0.10). Concluiu-se que, neste contexto, as notícias desinformativas não são tão diferentes das verdadeiras.

Face a esta limitação, alterou-se a estratégia para uma abordagem de **Aprendizagem Supervisionada**. O problema foi reestruturado utilizando um *dataset* equilibrado (50% notícias reais, 50% falsas), permitindo aos algoritmos aprenderem as características discriminatórias de cada classe. Foram treinados três modelos distintos: *Random Forest*, *Support Vector Machine* (SVM) e uma *Neural Network*.

Ao contrário da tentativa não supervisionada, esta abordagem obteve resultados satisfatórios, conforme demonstrado na Tabela 2.

A eficiência desta nova estratégia é demonstrada na Figura 3. Enquanto a abordagem anterior apresentava uma sobreposição total, os novos modelos supervisionados consegui-

Tabela 2: Comparação de desempenho (F1-Score) após a transição para Aprendizagem Supervisionada (Dataset Equilibrado).

| **Modelo** | **F1-Score** |
|---|---|
| Random Forest | 0.9720 |
| SVM | 0.9774 |
| Neural Network | **0.9896** |

ram criar fronteiras de decisão claras, separando eficazmente as notícias verdadeiras das falsas.

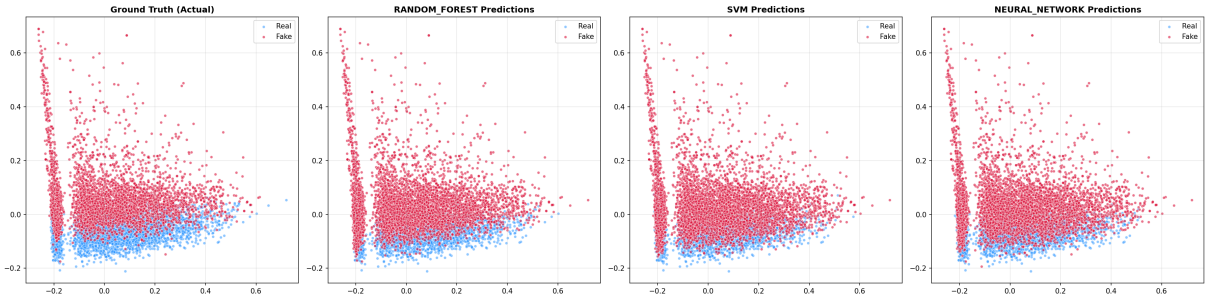


Figura 3: Visualização dos resultados da abordagem supervisionada

Desta análise comparativa, selecionou-se a Rede Neuronal (F1: 0.9896) como o modelo final, por ter uma capacidade maior de generalização na distinção de padrões complexos em notícias.

### 4.3 Modelo 3: Deteção de Stance (Postura)

O objetivo deste modelo é identificar a relação de concordância entre o título da notícia e o seu corpo, assumindo que notícias falsas apresentam frequentemente incoerências ou títulos "clickbait"que não correspondem ao conteúdo textual.

O principal desafio encontrado no desenvolvimento deste modelo foi o desequilíbrio do *dataset*. A classe maioritária ("unrelated"ou neutro) dominava as restantes, enviesando o modelo. Para mitigar este problema, aplicou-se uma técnica de *undersampling*, reduzindo aleatoriamente o número de exemplos da classe dominante até se obter uma distribuição equilibrada entre as classes, permitindo ao modelo aprender padrões distintivos de todas as categorias.

Na fase de vetorização, optou-se por aumentar a complexidade da representação TF-IDF para capturar melhor o contexto semântico. O parâmetro `max_features` foi aumentado para expandir o vocabulário considerado, e o intervalo de *n-grams* foi configurado

para $(1, 3)$. Isto significa que o modelo passou a analisar não apenas palavras isoladas (unigramas), mas também sequências de duas e três palavras (bigramas e trigramas), capturando expressões compostas essenciais para determinar a postura do texto.

Contrariamente aos módulos anteriores, não foi aplicada qualquer redução de dimensionalidade. As experiências realizadas demonstraram que a compressão do espaço vetorial resultava numa perda significativa de qualidade, indicando que a dispersão original dos dados era necessária para uma classificação eficaz. Dada a simplicidade e a alta dimensionalidade dos dados resultantes, comparou-se o desempenho de uma *Support Vector Machine* (SVM) e de uma *Random Forest*.

Tabela 3: Comparação do desempenho (F1-Score Macro) para a deteção de Stance.

| **Modelo** | **F1-Score (Macro)** |
|---|---|
| SVM | **0.8464** |
| Random Forest | 0.51 |

Os resultados, apresentados na Tabela 3, revelam uma grande diferença de desempenho. A SVM obteve um F1-Score de 0.8464, demonstrando uma capacidade muito superior para lidar com a alta dimensionalidade dos vetores de texto gerados pelos *n-grams*, enquanto o *Random Forest* revelou-se ineficaz para este nível de complexidade vetorial.

### 4.4 Modelo 4: Deteção de Clickbait

A última componente foca-se na análise dos títulos das notícias, com o objetivo de identificar o uso de técnicas de sensacionalismo, exagero ou omissão deliberada de informação (*clickbait*) para atrair a atenção do leitor.

O conjunto de dados utilizado para o treino foi o *ClickbaitDataset*. Considerando a natureza deste problema (classificação binária baseada em títulos curtos) e a especificidade do vocabulário tipicamente utilizado neste tipo de chamadas, optou-se por não aplicar técnicas de redução de dimensionalidade. As experiências preliminares indicaram que a compressão do espaço vetorial tendia a eliminar nuances linguísticas e palavras-chave determinantes, não justificando o ganho computacional face à relativa simplicidade do *dataset*.

Para esta tarefa de classificação, foram selecionadas e comparados dois algoritmos dis-

tintos: *XGBoost* (*Extreme Gradient Boosting*) e *CNN* (*Convolutional Neural Network*). O *XGBoost* foi escolhido por ser robusto e eficiente, servindo como uma linha de base sólida baseada em árvores de decisão. Por outro lado, a *CNN* foi selecionada por ser capaz de detetar padrões locais e sequenciais (como *n-grams* ou expressões específicas bastante utilizadas por títulos sensacionalistas) independentemente da sua posição na frase, uma característica teórica vantajosa para análise de texto curto.

Os resultados quantitativos, apresentados na Tabela 4, validam a hipótese inicial sobre a superioridade das redes neuronais para este contexto específico.

Tabela 4: Comparação do desempenho (F1-Score Binary) para a deteção de Clickbait.

| **Modelo** | **F1-Score (Default - Binary)** |
|---|---|
| XGBoost | 0.8462 |
| CNN | **0.9509** |

A CNN apresentou um desempenho significativamente superior ao XGBoost (F1-Score de 0.9509 contra 0.8462). Este resultado demonstra que a deteção de *clickbait* beneficia consideravelmente da extração de características espaciais e padrões locais que as camadas convolucionais conseguem isolar melhor, superando a abordagem de *gradient boosting* na captura da estrutura sintática e semântica de títulos sensacionalistas.

### 4.5 Modelo Final: Fake News Meta-Classifier

O último componente da arquitetura é o Meta-Classificador. Ao contrário dos modelos anteriores, este modelo não analisa o texto da notícia diretamente. Ele simplesmente recebe as opiniões (previsões) de todos os modelos especialistas e toma a decisão final.

Para treinar este modelo, foi necessário construir um novo conjunto de dados a partir do *dataset* WELFake. O processo foi o seguinte:

1. Cada notícia do WELFake passou pelos quatro modelos anteriores (Tópicos, Anomalias, Stance e Clickbait);

2. As previsões de cada modelo foram guardadas como novas colunas (features);

3. Criou-se sim uma tabela onde as colunas representam a "opinião"de cada especialista e a variável alvo é a veracidade da notícia (Real ou Fake).

Com este novo *dataset*, foram testados cinco algoritmos de classificação para encontrar o que melhor conseguia combinar estas informações. Os resultados de desempenho (F1-Score) encontram-se na Tabela 5.

Tabela 5: Comparação de desempenho dos algoritmos para o Meta-Classificador.

| **Algoritmo** | **F1-Score** |
|---|---|
| Logistic Regression | 0.8770 |
| Neural Network | 0.8404 |
| SVM | 0.8604 |
| Random Forest | 0.8790 |
| **XGBoost** | **0.8794** |

A análise dos resultados mostra que os modelos baseados em árvores de decisão (*Random Forest* e *XGBoost*) obtiveram os melhores desempenhos, superando a rede neuronal e o SVM neste tipo de dados tabulares.

O **XGBoost** foi o algoritmo selecionado como Modelo Final, pois apresentou o F1-Score mais alto (0.8794). Embora a diferença para o *Random Forest* seja pequena, o XGBoost demonstrou ser ligeiramente mais eficaz a combinar as diferentes saídas dos modelos especialistas para distinguir notícias falsas de verdadeiras.

# 5 Interface de Utilização

De forma a cumprir o requisito de desenvolver uma aplicação que permita a interação com o modelo treinado, foi criada uma interface *web*, com o objetivo de permitir que qualquer utilizador verifique a veracidade de uma notícia e, simultaneamente, oferecer uma visão transparente sobre a performance dos modelos desenvolvidos.

A aplicação foi desenvolvida na linguagem Python (biblioteca *Streamlit*[1]) e encontra-se dividida em dois módulos principais.

## 5.1 Deteção em Tempo Real

Este é o módulo principal, onde o utilizador interage com o sistema final. O fluxo de utilização resume-se a três passos:

1. **Inserção do URL:** O utilizador introduz o *link* da notícia que deseja verificar;
2. **Extração Automática:** O sistema executa um algoritmo de *web scraping* que acede à página e extrai automaticamente o título e o corpo da notícia;
3. **Classificação:** O texto extraído é processado pelos modelos especializados e pelo Meta-Classificador, devolvendo o resultado final (Verdadeiro/Falso) e o nível de confiança.

## 5.2 Painel de Visualizações e Métricas

Para garantir que os resultados do projeto são transparentes e fáceis de analisar, foi adicionado um separador dedicado a "Visualizações". Nesta área, o utilizador pode consultar graficamente o desempenho de todos os modelos criados durante o projeto.

O painel inclui:

- **Comparação de Modelos:** Gráficos de barras que mostram os F1-Scores de todos os algoritmos testados, permitindo perceber rapidamente qual foi o "vencedor"em cada categoria;
- **Distribuição de Dados:** Gráficos de dispersão (*scatterplots*) que ilustram como os modelos separam as notícias verdadeiras das falsas.

[1] https://streamlit.io

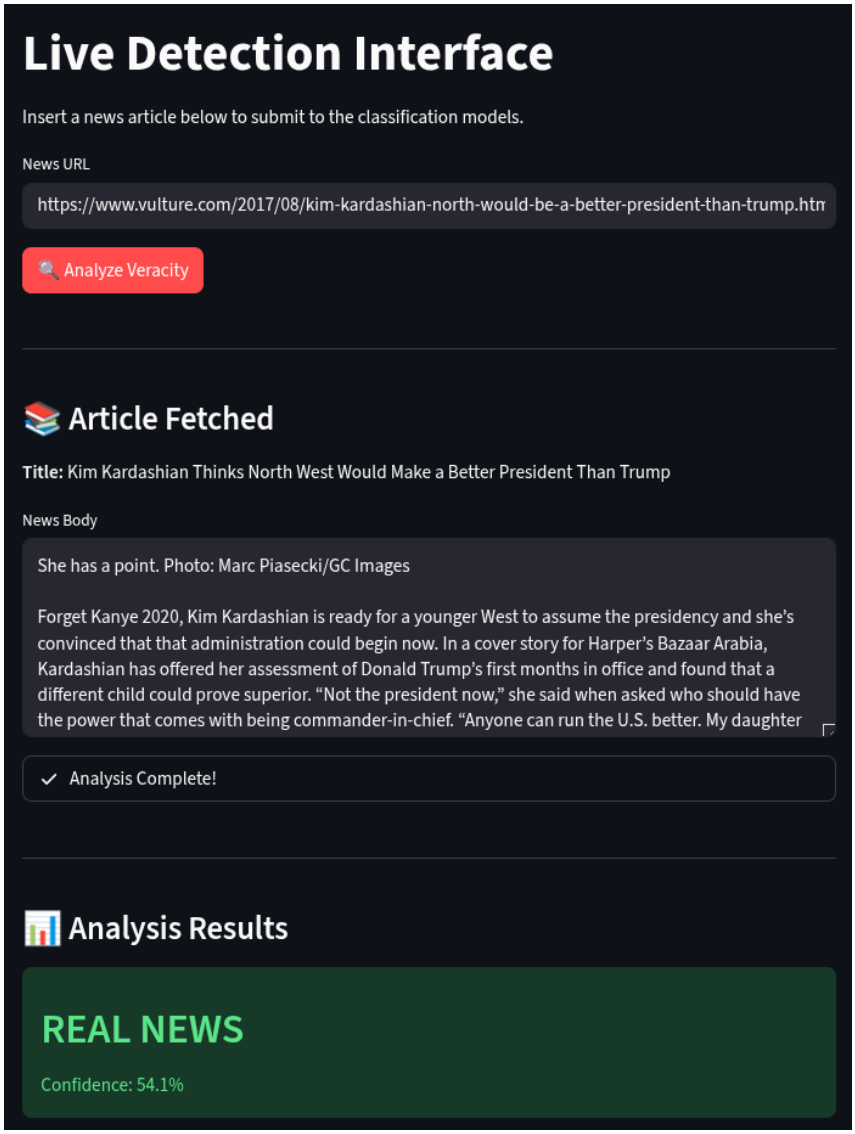


Figura 4: Interface de deteção de *Fake News*

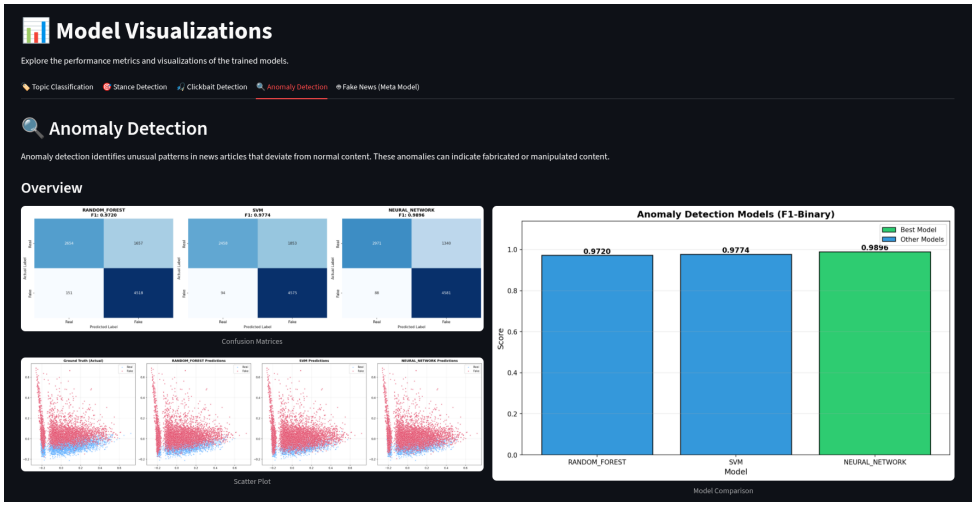


Figura 5: Painel de métricas da aplicação: Comparação do desempenho dos modelos (F1-Score) e visualização da distribuição das classes.

# 6 Conclusões e Trabalho Futuro

## 6.1 Reflexão Crítica dos Resultados

Este projeto serviu para provar que usar vários modelos a trabalhar em equipa funciona melhor do que tentar resolver tudo com um só modelo. A divisão do problema em várias partes, como analisar o tópico, o clickbait ou a coerência, permitiu olhar para as notícias falsas de várias formas diferentes.

Os resultados mostraram que o Modelo Final (baseado em XGBoost) conseguiu juntar as opiniões de todos os outros modelos e ter um bom desempenho, com uma precisão a rondar os 88% no conjunto de dados WELFake. Isto confirma que quando juntamos várias especialidades, a decisão final é mais acertada.

Uma lição importante que se tirou deste trabalho foi sobre a deteção de anomalias. Inicialmente, pensou-se que seria possível apanhar notícias falsas apenas por serem diferentes das reais. No entanto, os testes mostraram que isso não funciona, porque as notícias falsas são escritas para parecerem verdadeiras e imitam muito bem o estilo das notícias reais. Por isso, a abordagem teve de mudar para um método supervisionado (Rede Neuronal), onde se ensinou ao computador exemplos concretos do que é verdade e mentira, para poder obter resultados acima dos 98%.

## 6.2 Conclusões e Trabalho Futuro

Pode-se concluir que os objetivos do trabalho foram cumpridos. Criou-se um sistema completo que consegue receber uma notícia e dizer se é verdadeira ou falsa com base em factos e análises concretas. A criação da interface gráfica foi o passo final que permitiu tornar este sistema matemático numa ferramenta que qualquer pessoa consegue utilizar.

Para o futuro, e para melhorar este sistema, sugerem-se os seguintes passos:

- **Usar Modelos de Linguagem Modernos (LLMs):** Em vez de usar métodos simples de contagem de palavras, integrar modelos mais avançados como o BERT, que conseguem perceber o contexto e o significado das frases muito melhor.
- **Suporte para Português:** Atualmente o sistema só funciona em inglês. Seria importante treinar os modelos com notícias em português ou adicionar um tradutor automático para que possa ser usado em Portugal.

- **Explicação da Decisão:** Adicionar na interface uma explicação para o utilizador perceber o motivo da classificação. Por exemplo, dizer "Esta notícia parece falsa porque o título diz o contrário do texto".

## Referências

[1] M. Alves and E. Maciel, "O fenômeno das fake news: definição, combate e contexto," Feb. 2020.

[2] C. Tenove, "Protecting Democracy from Disinformation: Normative Threats and Policy Responses - Chris Tenove, 2020," May 2020.

[3] C. P. Galhardi, N. P. Freire, M. C. d. S. Minayo, and M. C. M. Fagundes, "Fato ou Fake? Uma análise da desinformação frente à pandemia da Covid-19 no Brasil," *Ciência & Saúde Coletiva*, vol. 25, pp. 4201–4210, 2020. Publisher: ABRASCO - Associação Brasileira de Saúde Coletiva.

[4] S. Mishra, P. Shukla, and R. Agarwal, "Analyzing Machine Learning Enabled Fake News Detection Techniques for Diversified Datasets," *Wireless Communications and Mobile Computing*, vol. 2022, no. 1, p. 1575365, 2022. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1155/2022/1575365.