Universidade Estadual de Maringá
Centro de Tecnologia
Departamento de Informática
Curso de Ciência da Computação

# UMA COMPARAÇÃO ENTRE MODELOS DE ANÁLISE DE SENTIMENTO SOBRE NOTÍCIAS FINANCEIRAS

Diogo Felipe Soares da Silva

**Trabalho de Conclusão de Curso – 2025**

Maringá
2025

**Universidade Estadual de Maringá – Departamento de Informática**
**Av. Colombo, 5790 – Bloco C56, Zona 7 – CEP: 87020-900**
**Fone: (44) 3011-4324/3011-4219**
**Maringá – PR**

Universidade Estadual de Maringá
Centro de Tecnologia
Departamento de Informática
Curso de Ciência da Computação

# UMA COMPARAÇÃO ENTRE MODELOS DE ANÁLISE DE SENTIMENTO SOBRE NOTÍCIAS FINANCEIRAS

Diogo Felipe Soares da Silva

**Trabalho de Conclusão de Curso – 2025**

Trabalho de Conclusão de Curso apresentado à Universidade Estadual de Maringá, como parte dos requisitos necessários à obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Wagner Igarashi
Banca: Prof. Dr. Yandre Maldonado
Banca: Prof. Dr. Franklin César Flores

Maringá
2025

**Universidade Estadual de Maringá – Departamento de Informática**
**Av. Colombo, 5790 – Bloco C56, Zona 7 – CEP: 87020-900**
**Fone: (44) 3011-4324/3011-4219**
**Maringá – PR**

# Agradecimentos

Agradeço a Deus, a minha família e aos meus amigos por tornarem a experiência da graduação mais leve. Também sou grato a todos os professores que contribuíram para a minha formação e, em especial, ao meu orientador Prof. Dr. Wagner Igarashi, cuja orientação foi fundamental para o desenvolvimento deste trabalho. Obrigado!

# Resumo

A constante publicação de notícias financeiras em portais especializados representa uma fonte rica de informações que pode influenciar diretamente as decisões de investimento no mercado de ações. Diante da volatilidade característica do mercado brasileiro, torna-se essencial que os investidores contem com ferramentas capazes de auxiliá-los na interpretação e antecipação de movimentos de ativos. Com os recentes avanços em inteligência artificial e no processamento de linguagem natural (PLN), abre-se espaço para o desenvolvimento de métodos automatizados capazes de extrair valor de grandes volumes de dados textuais, como as notícias financeiras. Neste contexto, a Análise de Sentimentos, também conhecida como Sentiment Analysis (SA), destaca-se como uma técnica relevante para identificar e classificar textos ligados ao mercado de capitais. Essa abordagem pode ser implementada por meio de modelos baseados em dicionários de termos ou utilizando algoritmos de aprendizado de máquina (Machine Learning), que aprendem a partir de grandes volumes de dados previamente rotulados. A presente pesquisa tem como objetivo desenvolver, avaliar e comparar essas duas abordagens de análise de sentimentos aplicadas às notícias financeiras relacionadas às ações PETR3 e PETR4, da Petrobras. Para tanto, foi utilizada uma base de notícias coletadas em diversos portais financeiros nacionais, abrangendo quatro anos distintos, organizados em dois períodos: pandemia (2020 e 2021) e pós-pandemia (2023 e 2024). A partir dessa análise, foi investigado a correlação entre as polaridades sentimentais atribuídas às notícias e sua associação com o comportamento do preço das ações da Petrobras ao longo do tempo. Os experimentos indicaram que o modelo de aprendizagem de máquina (XLM-roBERTa) foi superior tanto nas classificações das notícias quanto no nível de correlação com o mercado financeiro. Apesar do modelo de dicionário Loughran e McDonald (L&M) não ter apresentado resultados equilibrados, sua implementação exigiu menos poder computacional e, também, obteve maior velocidade nas classificações, além de ter superado o modelo XLM-roBERTa no ano de 2021. Com isso, este estudo busca contribuir para o avanço de soluções computacionais mais eficientes no suporte à tomada de decisões por profissionais do mercado financeiro, disponibilizando recursos que viabilizem análises mais estratégicas, precisas e automatizadas.

**Palavras-chave:** Análise de Sentimento; Aprendizado de Máquina; Inteligência Artificial, Mercado Financeiro, Mercado de Ações.

# Abstract

The constant publication of financial news on specialized portals represents a rich source of information that can directly influence investment decisions in the stock market. Given the characteristic volatility of the Brazilian market, it is essential for investors to rely on tools capable of assisting them in interpreting and anticipating asset movements. With recent advances in artificial intelligence and natural language processing (NLP), there is room for the development of automated methods capable of extracting value from large volumes of textual data, such as financial news. In this context, Sentiment Analysis (SA) stands out as a relevant technique for identifying and classifying texts related to the capital market. This approach can be implemented through dictionary-based models of terms or by using machine learning algorithms, which learn from large volumes of previously labeled data.This research aims to develop, evaluate, and compare two sentiment analysis approaches applied to financial news concerning Petrobras' PETR3 and PETR4 stocks. To this end, a database of news articles collected from various national financial portals was used, covering four distinct years organized into two periods: pandemic (2020 and 2021) and post-pandemic (2023 and 2024). From this analysis, the correlation between the sentimental polarities assigned to news articles and their association with Petrobras's stock price behavior over time was investigated. The experiments indicated that the machine learning model (XLM-roBERTa) outperformed both in news classification and in the level of correlation with the financial market. Although the Loughran and McDonald (L&M) dictionary model did not show balanced results, its implementation required less computational power and achieved faster classification speeds. Moreover, it outperformed the XLM-roBERTa model in 2021. Therefore, this study aims to contribute to the advancement of more efficient computational solutions to support decision-making by financial market professionals, providing tools that enable more strategic, accurate, and automated analyses.

**Keywords:** Sentiment Analysis; Machine Learning; Artificial Intelligence; Financial Market; Stock Market.

# Lista de ilustrações

# Lista de Tabelas e Quadros

# Lista de Blocos de Código

# Sumário

# 1 Introdução

Diariamente, notícias sobre o mercado financeiro são publicadas nos principais portais do mundo. Dessa forma, profissionais da área permanecem constantemente atentos a essas publicações com o objetivo de tomarem decisões de investimento mais embasadas. E levando em consideração um mercado emergente e altamente volátil como o brasileiro, é importante que o investidor possua ferramentas adequadas que o auxiliem em seu ofício tanto na escolha do ativo quanto na previsão de sua movimentação no mercado.

Nesse contexto, considerando o recente avanço da inteligência artificial voltada para o processamento de linguagem natural (PLN) e com a disponibilidade dos recursos computacionais atuais, torna-se factível o desenvolvimento de um estudo que correlaciona o grande volume de notícias financeiras disponíveis com a cotação de um determinado ativo no mercado financeiro.

Uma das técnicas utilizadas é a Análise de Sentimentos (AS), conhecida também como *Sentiment Analysis (SA)*, de textos de notícias financeiras. Estes textos são classificados como positivos ou negativos de forma automatizada pelo computador (TABOADA, 2016). Além disso, a classificação do texto pode ocorrer por diferentes abordagens: seja mediante a contagem de palavras presentes em um dicionário de termos, seja por meio de técnicas de aprendizagem de máquina.

Segundo a Hipótese dos Mercados Eficientes, as variações no mercado financeiro ocorrem pela percepção de novidades externas (FAMA, 1970). Portanto, é interessante aplicar a AS em notícias financeiras de certo ativo com o intuito de obter uma correlação que, embora não necessariamente alta, seja estatisticamente significativa e operacionalmente útil, capaz de auxiliar o investidor na tomada de decisão e potencialmente melhorar sua rentabilidade.

A técnica de AS está intimamente ligada com a análise fundamentalista do mercado financeiro, visto que tenta encontrar o valor intrínseco do ativo, isto é, quando o seu real valor difere daquele que está sendo negociado no momento. E essa análise é feita a partir do estudo de fatores macro-econômicos (PETRUSHEVA; JORDANOSKI, 2016), que no escopo deste trabalho serão as notícias financeiras.

Diversos estudos têm sido desenvolvidos no âmbito da aplicação da AS em notícias financeiras. Entre eles, destaca-se o trabalho de Patil, Sharma e Sinha (2024), que realizaram a coleta de títulos de notícias relacionadas a ações como a Tesla (TSLA) e ao índice norte-americano S&P 500. Após a obtenção dos dados, utilizaram a ferramenta *NLTK (Natural Language Toolkit)* para a classificação da polaridade desses títulos. Além disso, aplicaram modelos de aprendizagem de máquina, como o *Random Forest* e o *Long Short-Term*

*Memory (LSTM)*, para aprimorar a análise dos sentimentos. Os resultados indicaram que o modelo *LSTM* apresentou desempenho superior ao *Random Forest* em termos de precisão na correlação.

Outro estudo, conduzido por Maqbool et al. (2023), utilizou bibliotecas de PLN para atribuir polaridade às notícias financeiras coletadas. Nesse trabalho, foi proposto um modelo de aprendizagem de máquina denominado *MLB-Regressor*, que utiliza como entrada tanto as notícias financeiras previamente classificadas quanto o histórico de preços do ativo analisado, com o objetivo de prever o movimento futuro da ação. O histórico de cotações das quatro ações consideradas no estudo (Reliance, Tata Motors, Tata Steel e HDFC) foi obtido por meio do *Yahoo Finance*. Os resultados demonstraram que o modelo proposto alcançou uma precisão de 0,90 tanto na previsão das tendências atuais quanto na previsão das tendências futuras em um período de 10 dias.

Uma comparação entre modelos de aprendizagem de máquina foi realizado no estudo de Agrawal e Mukherjee (2025), onde destacou-se a superioridade do modelo *BERT (Bidirecional Encoder Representations from Transformers)* em relação aos demais. Além disso, o estudo demonstrou que modelos tradicionais baseados exclusivamente em indicadores técnicos ficam em desvantagem dos modelos que incluem dados sentimentais em suas previsões.

Além dos trabalhos citados no âmbito internacional, tem-se desenvolvido alguns estudos de iniciação científica na Universidade Estadual de Maringá, relacionados ao tema de análise de sentimentos no mercado de ações. Um desses estudos é o de Silva (2020), que se baseou no modelo de Dicionário de Termos para fazer sua pesquisa. Diferentemente dessa abordagem, a presente pesquisa amplia a análise ao incorporar, além da técnica lexicográfica, a aplicação de um modelo de Aprendizagem de Máquina, também conhecido como *Machine Learning (ML)*, para a análise de sentimentos em notícias financeiras, proporcionando uma comparação entre as duas metodologias.

Diante do cenário apresentado, esta pesquisa tem como objetivo avaliar e comparar o desempenho de dois modelos de análise de sentimentos, um baseado em dicionário de termos e outro fundamentado em técnicas de aprendizagem de máquina, aplicados a textos de notícias financeiras. Busca-se identificar qual modelo apresenta maior acurácia classificatória com base na interpretação das características linguísticas contextuais do texto, bem como examinar o grau de similaridade entre as classificações produzidas por ambos. Adicionalmente, pretende-se estimar o nível de correlação entre as classificações de cada modelo com a movimentação dos ativos analisados no mercado financeiro, com a finalidade de verificar o potencial preditivo de cada abordagem. Por fim, o estudo visa discutir as vantagens e limitações inerentes a cada método no contexto da análise de sentimentos aplicada a notícias financeiras.

## 1.1 Justificativa e Motivação

Com a explosão do volume de informações geradas diariamente na internet a partir dos anos 2000, especialmente em redes sociais, fóruns e plataformas de avaliação, tornou-se cada vez mais estratégico extrair dessas fontes os dados que expressem as opiniões, percepções e sentimentos dos usuários. A análise de sentimentos surge nesse contexto como uma ferramenta poderosa para transformar esse conteúdo textual desestruturado em conhecimento útil, (LIU, 2011).

No mercado financeiro brasileiro, marcado por grande volatilidade, a previsão de suas movimentações torna-se uma tarefa muito árdua. Nesse contexto, notícias financeiras exercem grande influência sobre os investidores e os preços dos ativos, pois refletem e, muitas vezes, moldam as expectativas do mercado. Estudos de AS sobre essas notícias são essenciais para captar o tom das opiniões predominantes (positivo, negativo ou neutro) sobre um ativo, revelando, assim, suas tendências de movimentação antes que elas realmente aconteçam. Dessa forma, a aplicação de instrumentos automatizados no estudo do mercado financeiro é fundamental para a construção de uma base sólida de informações, permitindo que os investidores estejam mais preparados na tomada de suas decisões.

## 1.2 Organização do trabalho

O desenvolvimento deste trabalho está estruturado da seguinte forma: Seção 1, a presente seção de caráter introdutório; Seção 2, onde discute sobre a fundamentação teórica adotada para a elaboração deste trabalho; Seção 3, em que descreve sobre a metodologia e as ferramentas utilizadas no estudo; Seção 4, onde são elucidados os detalhes do desenvolvimento; Seção 5, no qual são apresentados os resultados dos experimentos; Seção 6, em que são delineadas as conclusões e as possibilidades de estudos futuros; e, por fim, são descritas as referências bibliográficas e apêndices.

# 2 Fundamentação Teórica

Este estudo apresenta uma fundamentação baseada em conceitos da área de ciência da computação e, também, do mercado financeiro. Apesar de ambas as áreas serem de suma importância para esta pesquisa, o enfoque será dado para as definições envolvendo a análise de sentimentos, visto que essa técnica constitui o núcleo metodológico para a geração dos resultados. Entretanto, é imprescindível que os fundamentos do mercado financeiro sejam igualmente considerados, garantindo uma compreensão sólida do contexto em que os dados estão inseridos.

## 2.1 Mercado Financeiro

Segundo Mishkin e Eakins (1995), o mercado financeiro pode ser definido como o conjunto de instituições responsáveis por intermediar a transferência de recursos de agentes que possuem excedente de capital para aqueles que, embora não o detenham em abundância, têm potencial para utilizá-lo de forma produtiva. Um mercado financeiro funcional é um indicador chave no crescimento econômico de uma nação. Dentre seus segmentos, o mercado de ações se destaca pela ampla visibilidade nos portais de notícias, razão pela qual será o foco deste estudo. Portanto, é preciso estabelecer previamente os termos e conceitos que nortearão esta pesquisa.

### 2.1.1 Mercado de Ações

Uma ação representa uma parcela da propriedade de uma empresa. Isto é, um investimento financeiro que confere direitos sobre os lucros e ativos da corporação. A emissão de ações e sua venda ao público é uma maneira das empresas captarem o capital necessário para financiar suas atividades e, com isso, continuar seu desenvolvimento (MISHKIN; EAKINS, 1995).

Um investidor pode obter retorno sobre ações de duas maneiras principais: pela valorização do preço do ativo ou pelo recebimento de dividendos distribuídos pela empresa. Muitos investidores, buscando alcançar maior lucro com seus títulos, tentam prever as movimentações dos preços de suas ações. No entanto, antecipar os movimentos do mercado é uma tarefa complexa, uma vez que diversos fatores internos e externos influenciam a direção dos preços. Por essa razão, o mercado de ações é frequentemente descrito como um lugar onde pessoas podem enriquecer e empobrecer rapidamente.

No presente trabalho, será desenvolvida uma análise do mercado de ações com foco

nos ativos PETR3 e PETR4, que correspondem às ações ordinárias [1] e preferenciais [2] da empresa Petrobras, respectivamente.

Fundada em 1953, durante o governo Getúlio Vargas, a Petrobras é considerada uma das maiores empresas do Brasil e uma das maiores companhias petrolíferas do mundo e detém o objetivo de garantir a autossuficiência energética do país. Ela é uma empresa de capital aberto, sendo o governo brasileiro o principal acionista. Os cidadãos brasileiros podem investir na Petrobras mediante a compra de suas ações na Bolsa de Valores de São Paulo, também conhecida como B3 [3], (Petrobras, 2025).

### 2.1.2 Análise Fundamentalista

De acordo com Murphy (1999), todos os fatores externos e relevantes que afetam o preço de um ativo são examinados pela análise fundamentalista, visando determinar o seu valor intrínseco. Os fundamentalistas acreditam que quando o valor intrínseco de uma ação está abaixo de seu valor de mercado, significa a supervalorização desta ação e, portando, deve ser vendida. Entretanto, quando o valor intrínseco está acima do valor de mercado, a ação está subvalorizada e deve ser feita a compra.

Neste contexto, é possível entender o valor intrínseco de um ativo como o seu valor real, baseado na lei de oferta e demanda. Assim, para um investidor estimar o valor intrínseco de um ativo, é essencial que ele analise todos os fatores econômicos que o envolvem. Isso inclui analisar o segmento da empresa, as empresas concorrentes, os relatórios financeiros divulgados e todas as notícias financeiras recentes. Tudo isso para compreender sua performance atual e conseguir incluir todas essas informações no valor real do ativo, (PETRUSHEVA; JORDANOSKI, 2016).

Uma premissa central desta análise estabelece que, no curto prazo, o preço das ações não corresponde ao seu valor real, mas ao longo do tempo ele tende a se corrigir sozinho. Portanto, convém dizer que esta análise está atrelada a investimentos de longo prazo. Um bom exemplo de investidor fundamentalista seria Luiz Barsi Filho [4], o maior investidor pessoa física do Brasil.

Além da análise fundamentalista, também existe a análise técnica. Ambas as abordagens possuem o mesmo objetivo: prever a movimentação de preços das ações. No entanto, diferem quanto ao método utilizado. Enquanto a análise técnica preocupa-se em estudar os efeitos visíveis do movimento do mercado, com o auxílio de gráficos e

[1] Uma ação ordinária é um título que confere direito a voto nas assembleias da empresa.
[2] Uma ação preferencial é um título que confere participação no capital social e nos lucros da empresa.
[3] A B3 é a bolsa de valores oficial do Brasil, onde são negociados ativos como ações, derivativos e títulos públicos.
[4] Luiz Barsi Filho é um dos maiores investidores da bolsa de valores do Brasil, conhecido por sua estratégia de longo prazo focada em ações que pagam dividendos

indicadores, a análise fundamentalista foca na causa desse movimento, (MURPHY, 1999).

No contexto desta pesquisa, o uso da análise técnica não será abortada em profundidade, dando lugar para a análise fundamentalista como principal instrumento de avaliação dos resultados deste estudo.

## 2.2 Inteligência Artificial

O termo Inteligência Artificial (IA) surgiu em 1955, pelo renomado cientista da computação John McCarthy [5]. Ele definiu IA como uma forma de programas simularem a inteligência humana em suas ações. E de acordo com o dicionário online Infopédia (2025), o termo inteligência significa: "Conjunto de todas as funções mentais que permitem o pensamento, o entendimento, a aprendizagem, o raciocínio e a interpretação".

Segundo o trabalho de Russell e Norvig (2010), é possível classificar a IA em quatro estratégias distintas, cada uma com métodos próprios e que se complementam.

**Figura 1** – *Definições de Inteligência Artificial*

| **Pensando como um humano** | **Pensando racionalmente** |
|---|---|
| "O novo e interessante esforço para fazer os computadores pensarem (...) *máquinas com mentes*, no sentido total e literal." (Haugeland, 1985)<br>"[Automatização de] atividades que associamos ao pensamento humano, atividades como a tomada de decisões, a resolução de problemas, o aprendizado..." (Bellman, 1978) | "O estudo das faculdades mentais pelo uso de modelos computacionais." (Charniak e McDermott, 1985)<br>"O estudo das computações que tornam possível perceber, raciocinar e agir." (Winston, 1992) |
| **Agindo como seres humanos** | **Agindo racionalmente** |
| "A arte de criar máquinas que executam funções que exigem inteligência quando executadas por pessoas." (Kurzweil, 1990)<br>"O estudo de como os computadores podem fazer tarefas que hoje são melhor desempenhadas pelas pessoas." (Rich and Knight, 1991) | "Inteligência Computacional é o estudo do projeto de agentes inteligentes." (Poole *et al.*, 1998)<br>"AI... está relacionada a um desempenho inteligente de artefatos." (Nilsson, 1998) |

**Fonte:** Russell e Norvig (2010).

No presente estudo, o foco está na última abordagem, chamada "Agindo Racionalmente", que define a IA como o estudo de agentes inteligentes.

Agentes inteligentes podem ser definidos como entidades capazes de perceber seu ambiente e agir de forma adequada para alcançar o melhor resultado possível. Em cenários de incerteza, devem ser capazes de adaptar-se a fim de atingir o melhor resultado esperado.

[5] <http://jmc.stanford.edu/general/index.html>

Existem diversos tipos de agentes inteligentes, que variam conforme sua complexidade e capacidade de atuação. Entre os principais, destacam-se: agentes reativos simples, agentes reativos em modelos, agentes baseados em objetos, agentes baseados na utilidade e agentes com aprendizagem.

Os agentes com aprendizagem serão importantes para esta pesquisa e eles possuem 4 componentes teóricos em sua estrutura: crítico, gerador de problemas, elemento de desempenho e elemento de aprendizado. Os dois últimos são os mais importantes, sendo responsáveis, respectivamente, pela seleção de ações externas e pelo aperfeiçoamento contínuo do programa.

A inteligência artificial será usada como ferramenta neste estudo, como uma das formas de automatizar a análise das notícias financeiras coletadas.

### 2.2.1 Mineração de Dados na *Web*

Mineração de Dados é o processo de descoberta de padrões e conhecimentos úteis a partir de diversas fontes, como bancos de dados, textos, imagens e a *Web*. Esses padrões devem ser válidos, compreensíveis e ter potencial de aplicação. Ademais, também é uma área interdisciplinar, envolvendo campos como aprendizado de máquina, estatística, banco de dados e visualização de informações. Seu objetivo é transformar dados brutos em conhecimento valioso que possa apoiar decisões, prever tendências e entender comportamentos, (LIU, 2011).

Há uma área subárea dentro da mineração de dados chamada de mineração de dados na *Web*, ou conhecido em inglês como *Web data mining*. Esse é o modelo de mineração que será empregado na fase inicial da parte empírica deste estudo, pois a coleta das notícias financeiras necessárias para a AS acontecerão a partir da extração do conteúdo de páginas *Web*. Portanto, compreender seus conceitos são de suma importância.

De acordo com Liu (2011), o processo de mineração de dados na *Web* consiste em encontrar informações ou conhecimentos relevantes a partir da estrutura de *hyperlinks* da *Web*, do conteúdo das páginas e dos dados de uso. No presente estudo, será necessário selecionar fontes adequadas para coletar os dados de interesse que, neste caso, são as notícias financeiras.

O processo de mineração de dados tradicional apresenta 3 fases principais que também são aplicadas na mineração da *Web*, sendo elas:

- Pré-processamento: Nessa etapa, realiza-se a limpeza dos dados brutos, uma vez que eles não estão adequados para a mineração por diversos motivos. Os dados coletados podem ser muito extensos e conter atributos irrelevantes para o processo

de mineração. Além disso, podem apresentar ruídos e anomalias, exigindo, portanto, um tratamento prévio;

- Mineração de dados: Os dados filtrados são processados por algum algoritmo de mineração. Ao final, é gerado padrões ou conhecimento;
- Pós-processamento: Essa etapa é encarregada de identificar os padrões produzidos que serão úteis na resolução do problema desejado. Diversas técnicas de avaliação e visualização são aplicadas para ajudar na tomada de decisão.

A mineração de dados é um processo frequentemente iterativo, tendo que repeti-lo várias vezes até que se obtenha um resultado satisfatório, capaz de ser aplicado a problemas reais.

Entretanto, mesmo a mineração de dados da *Web* apresenta muitas técnicas em comum com a mineração tradicional, ela também utiliza algoritmos próprios, devido à natureza semi-estruturada e não estruturada dos dados presentes na *Web*, (LIU, 2011). Essa etapas são classificadas nas seguintes categorias:

- Mineração da estrutura da *Web*: Esta categoria tem como objetivo extrair informações relevantes a partir dos links que compõem a estrutura da *Web*. Por meio desses links, é possível identificar outras páginas de destaque, o que constitui uma das técnicas fundamentais dos mecanismos de busca. Além disso, há como detectar comunidades de usuários com interesses em comum, algo que a mineração de dados tradicional não é capaz de fazer.
- Mineração do conteúdo da *Web*: O objetivo desta categoria é extrair dados significativos do conteúdo das páginas *Web*. É possível fazer a classificação automática das páginas de acordo com seu tópico. Isso ajuda na identificação de padrões importantes para a formação de um conhecimento útil e aplicável.
- Mineração de uso da *Web*: Esta categoria corresponde à descoberta de padrões de acesso do usuário utilizando *Web logs* [6], os quais registram cada clique realizado por eles. Um dos principais desafios dessa abordagem está na etapa de pré-processamento do fluxo de cliques, necessária para filtrar os dados e torná-los adequados à mineração.

Como base nas categorias descritas, esta pesquisa concentra-se na vertente da mineração do conteúdo da *Web*, onde serão extraídas as informações de notícias financeiras de forma automatizada, de modo a identificar o sentimento subjacente a estas.

[6] *Web logs* são arquivos que registram automaticamente todas as atividades realizadas em um site.

### 2.2.2 Análise de Sentimentos

Análise de sentimentos é um campo que está inserido no contexto de processamento de linguagem natural, que conforme Russell e Norvig (2010) tem o objetivo de ganhar conhecimento, mesmo que parcialmente, sobre a linguagem usada por um ser humano. A dificuldade encontra-se em sua ambiguidade e constante mutação.

Segundo Taboada (2016), pode-se definir que:

> "A análise de sentimentos (AS) é um campo em crescimento na intersecção entre a linguística e a ciência da computação, que busca determinar automaticamente o sentimento contido no texto. O sentimento pode ser caracterizado como uma avaliação positiva ou negativa expressa por meio da linguagem.", (TABOADA, 2016)

O campo de AS teve seu crescimento a partir dos anos 2000, pois esse período coincide com a disseminação de várias plataformas online que deram poder às pessoas de expressarem suas opiniões na internet. Essas plataformas, também conhecidas como redes sociais, foram responsáveis, pela primeira vez na história da humanidade, pela acumulação de um grande volume de dados gravados digitalmente. Nesse contexto, a AS tornou-se uma das áreas de pesquisa mais ativas dentro do campo de PLN, (LIU, 2012).

Isso só se tornou possível devido ao fato de que a AS possui um campo de aplicabilidade muito amplo, podendo ser explorada praticamente em qualquer domínio. Além disso, apresenta muitos desafios que ainda não haviam sido estudados até então.

O valor sentimental extraído de textos digitais tem relevância para as entidades públicas e privadas, em virtude da utilidade dessas informações no apoio à tomada de decisão. Por essa razão, a AS não é mais estudado pela ciência da computação de forma exclusiva, mas também se espalhou para as ciências sociais, econômicas, política e de gestão.

De acordo com o trabalho de Liu (2012), a análise de sentimentos também pode ser chamada de mineração de opinião, pois seu campo de atuação envolve a análise das opiniões das pessoas, o que inclui sentimentos, atitudes, avaliações e emoções, sobre uma determinada entidade, como produtos, serviços ou organizações.

Liu (2012) também relata que a AS pode ser classificada em 3 níveis conforme o grau de profundidade da análise feita: Nível de Documento, Nível de Sentença, Nível de Entidade e Aspecto.

No Nível de Documento, a classificação ocorre no nível de documento completo, com o objetivo de identificar se o sentimento expressado é positivo ou negativo. Além disso, a opinião deve estar direcionada a uma única entidade (por exemplo, um único produto).

No Nível de Sentença a análise é feita em cada sentença do texto, podendo identificar uma opinião positiva, negativa ou neutra. Esse nível de análise está fortemente relacionado à classificação de subjetividade, sendo encarregado de separar frases que expressam informações objetivas daquelas que expressam visões subjetivas e opiniões.

As duas análises citadas anteriormente não são capazes de descobrir exatamente o que as pessoas gostam. Assim, no Nível de Entidade e Aspecto, há um maior nível de detalhe em sua análise. A classificação acontece em nível de opinião e baseia-se no fato de que uma opinião consiste de um sentimento (positivo ou negativo) e seu alvo (entidades e/ou seus diferentes aspectos). O objetivo é descobrir o sentimento de entidades e suas diferentes características. O exemplo "*A qualidade das fotos do IPhone é boa, mas sua bateria é curta*" analisa a qualidade das fotos e bateria (aspectos ou alvos da opinião) da entidade *IPhone*. O resultado dessa análise é a produção de dados estruturados que podem ser usados para análises quantitativas quando qualitativas.

Segundo Guellil e Boukhalfa (2015), o desenvolvimento da AS acontece em 4 etapas: extração dos dados, pré-processamento dos dados coletados, classificação e identificação do conhecimento útil. A etapa de extração de dados pode ocorrer de três formas principais:

1. Extração com base em pesquisas e textos já existentes (*corpus existente*);
2. Extração direta das fontes de dados disponíveis, como as redes sociais e sites de notícias (*corpus manual*);
3. Extração direta da fonte de dados, mas com o auxílio de uma ferramenta automatizada (*corpus automático*) chamada (API) (*Application Programming Interface*).

A etapa de pré-processamento é incumbida de realizar ajustes nos dados brutos coletados, com o intuito de torná-los adequados para classificação. Há diversos tratamentos que podem ser aplicados, sendo que o trabalho de Patil, Sharma e Sinha (2024) aborda os mais utilizados na AS:

- Limpeza do texto: Ocorre a remoção de caracteres irrelevantes, como símbolos desnecessários e caracteres especiais (por exemplo "#", "@", "&"). Aqui também ocorre a *tokenização* [7] do texto e sua conversão para letras minúsculas;
- Remoção de *stopwords*: Ocorre a remoção de palavras que não carregam valor sentimental, como artigos definidos e preposições. Por exemplo, palavras como "o", "a", "de", "por", "e", "em";
- *Stemming* ou *Lemmatization*: técnicas responsáveis por reduzir palavras à sua forma base (ou lema). *Stemming* realiza essa redução por meio de regras simples, o que

[7] Divisão do texto em unidades menores chamadas *tokens*

pode resultar em palavras truncadas e que não pertencem formalmente ao idioma (por exemplo, "correndo" torna-se "corr"). Já *Lemmatization* utiliza um dicionário linguístico para identificar o lema correto, garantindo que a palavra reduzida seja gramaticalmente válida na língua (por exemplo, "correndo" torna-se "correr").

A fase de classificação é a principal e mais importante tarefa na AS e pode ser realizada por três tipos distintos de abordagens automáticas, sendo elas:

1. Supervisionada: utiliza dados rotulados (com sentimentos previamente classificados como positivos, negativos ou neutros) para treinar algoritmos de aprendizado de máquina.

2. Não supervisionada: Não utiliza dados rotulados. Baseia-se em técnicas como métodos baseados em léxicos (dicionários de palavras com polaridades conhecidas).

3. Híbrida: combina elementos das abordagens supervisionada e não supervisionada para melhorar a precisão.

Concluindo, tem-se a geração de conhecimento útil, que pode ser de natureza geral ou técnica. Com esse conhecimento em mãos, entidades públicas e privadas tornam-se capazes de tomar melhores decisões.

### 2.2.3 Dicionários

No contexto de PLN, dicionários representam listas de palavras já polarizadas (por exemplo, "excelente" é positivo, "horroroso" é negativo) onde são a base para a análise léxica de textos. Nesse modelo, o sentimento de um texto é derivado do agregado do sentimento das palavras individuais contidas naquele texto. Assim, ao receber um texto de entrada, o modelo verifica a compatibilidade das palavras no texto com as do dicionário e, com isso, agrega seus valores positivos e negativos para produzir o valor semântico do texto, (TABOADA, 2016).

Um dicionário muito conhecido e utilizado na análise de textos financeiros é o dicionário desenvolvido por Tim Loughran e Bill McDonald, professores da Universidade de Notre Dame, nos Estados Unidos.

Esse dicionário, popularmente abreviado como L&M (*Loughran and McDonald Dictionary*), foi desenvolvido visando a criação de um modelo léxico voltado para a área financeira, visto que uma fonte comum na época, muito utilizada para a análise textual, era o dicionário de psicologia de Harvard, chamado de H4N.

De acordo com Loughran e McDonald (2011), o H4N apresentava muitas inconsistências para a classificação de palavras financeiras. Portanto, os pesquisadores, com o

intuito de refutar o H4N, analisaram múltiplos registros 10-K [8] de empresas americanas, no período de 1994 a 2008, e constataram que cerca de três quartos das palavras identificadas como negativas pelo H4N, no contexto financeiro, não apresentavam conotação negativa. Desse modo, criou-se um novo dicionário que representa de forma mais adequada os termos utilizados na área financeira.

O dicionário L&M apresenta uma lista de palavras da área financeira categorizadas em 6 tipos de sentimentos. Sendo eles: positivo, negativo, incerteza, litigioso, modal (forte e fraco) e restritivo.

No contexto deste estudo, para utilizar o dicionário L&M no modelo léxico, será necessário traduzi-lo, haja vista que sua versão original está em inglês e as notícias coletadas estão em português.

### 2.2.4 Aprendizagem de Máquina

Após discorrer sobre o tema AS, também será essencial discutir os conceitos de aprendizagem de máquina, uma vez que esta é uma ferramenta fundamental para a análise das notícias financeiras examinadas nesta pesquisa.

Segundo Mitchell (1997), aprendizagem de máquina pode ser definida como qualquer programa de computador que seja capaz de melhorar sua performance em alguma tarefa por meio da experiência. Exemplos dessa definição são fáceis de se encontrar atualmente, como, por exemplo, o sistema de reconhecimento facial, assistentes virtuais, carros autônomos e entre outros.

Russell e Norvig (2010) descrevem que qualquer agente consegue melhorar seu desempenho quando ele aprende utilizando dados. E essa melhoria depende de quatro fatores, sendo eles: qual componente do agente vai ser melhorado, qual conhecimento prévio o agente já possui, qual é a representação de dados utilizada e qual é o *feedback* disponível para o aprendizado do agente.

Além disso, é possível fazer a classificação de três tipos principais de aprendizagem de acordo com o *feedback* utilizado. O primeiro tipo é chamado de aprendizagem não supervisionada, pois agentes conseguem aprender padrões importantes a partir de uma entrada, mesmo ela não contendo nenhum *feedback* explícito. Dentro dessa forma de aprendizagem, a tarefa mais comum é o agrupamento, já que ele consegue detectar grupos de dados potencialmente relevantes para uso.

O segundo tipo é a aprendizagem por reforço. Nessa categoria, o agente aprende mediante recompensas e punições. Ou seja, quando mais recompensas o agente recebe, sabe-se que está executando algo correto, e caso receba punições, sabe-se que suas ações

[8] O 10-K é um relatório financeiro anual obrigatório das empresas de capital aberto americanas.

não estão de acordo com o objetivo. E a responsabilidade de detectar qual ação gerou a recompensa ou a punição fica a cargo do próprio agente.

O terceiro tipo é denominado de aprendizagem supervisionada. Aqui o agente observa diversos exemplos compostos por pares de entrada e saída e, por meio desse estudo, ele aprende uma função capaz de associar uma entrada à sua respectiva saída. Dessa forma, ao receber uma nova entrada, a função será capaz de prever corretamente a sua saída, (RUSSELL; NORVIG, 2010).

#### 2.2.4.1 Processamento de Linguagem Natural (PLN)

A área de PLN pode ser compreendida como um campo interdisciplinar que combina ciência da computação, linguística e inteligência artificial para construir sistemas capazes de realizar tarefas úteis com linguagem humana, seja para interações humano-computador, tradução, extração de informações ou outras finalidades práticas, (JURAFSKY; MARTIN, 2019).

#### 2.2.4.2 Redes Neurais

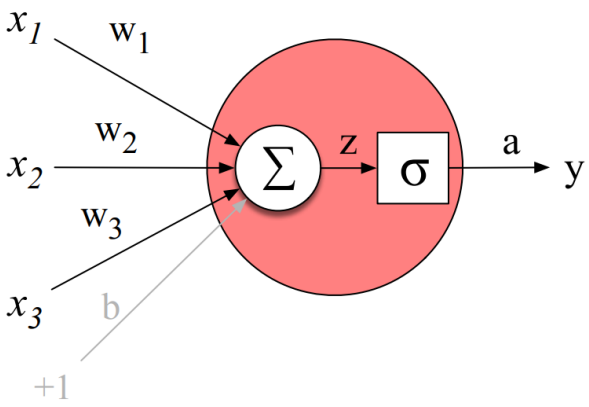
Uma rede neural é uma ferramenta fundamental no estudo de PLN. O termo "Neural" surgiu devido aos estudos do chamado Neurônio de McCulloch-Pitts (MCCULLOCH; PITTS, 1943), na qual foi desenvolvido um modelo simplificado do neurônio biológico como um elemento computacional que pode ser descrito em termos de lógica proposicional. Entretanto, o processamento de linguagem moderno não se baseia mais nessas inspirações biológicas, (JURAFSKY; MARTIN, 2025).

Atualmente, uma rede neural pode ser compreendida como uma rede de pequenas unidades computacionais, conhecidas como neurônios artificiais, em que cada uma dessas unidades recebe um vetor de valores como entrada e produz um único valor como saída.

Na imagem acima, o neurônio artificial possui 3 valores de entrada: $x1, x2$ e $x3$. Sendo que cada entrada possui seu respectivo peso: $w1, w2$ e $w3$. O valor $z$ indica o resultado do somatório das entradas multiplicadas por seus pesos e pelo termo de ajuste $b$, conhecido como *bias*. O resultado $z$ é introduzido em uma função de ativação $\sigma$, culminando na ativação $a$ desta unidade computacional. Como neste exemplo não há mais unidades para serem conectadas, o resultado da rede neural $y$ é o mesmo da função de ativação $a$.

Com o conhecimento da unidade básica de uma rede neural, o próximo passo é entender as estruturas de conexão de um neurônio artificial com outro para se formar uma rede. De acordo com Russell e Norvig (2010), há duas formas principais distintas se fazer isso: a primeira, chamada de rede de alimentação direta (*feed forward network*) e a segunda, conhecida como redes recorrentes.

**Figura 2** – *Unidade computacional de uma rede neural*




**Fonte:** Jurafsky e Martin (2025).

Na rede de alimentação direta, as unidades são conectadas em apenas uma direção, isto é, uma unidade recebe suas entradas de unidades em uma camada superior e devolve sua saída para unidades em camadas inferiores.

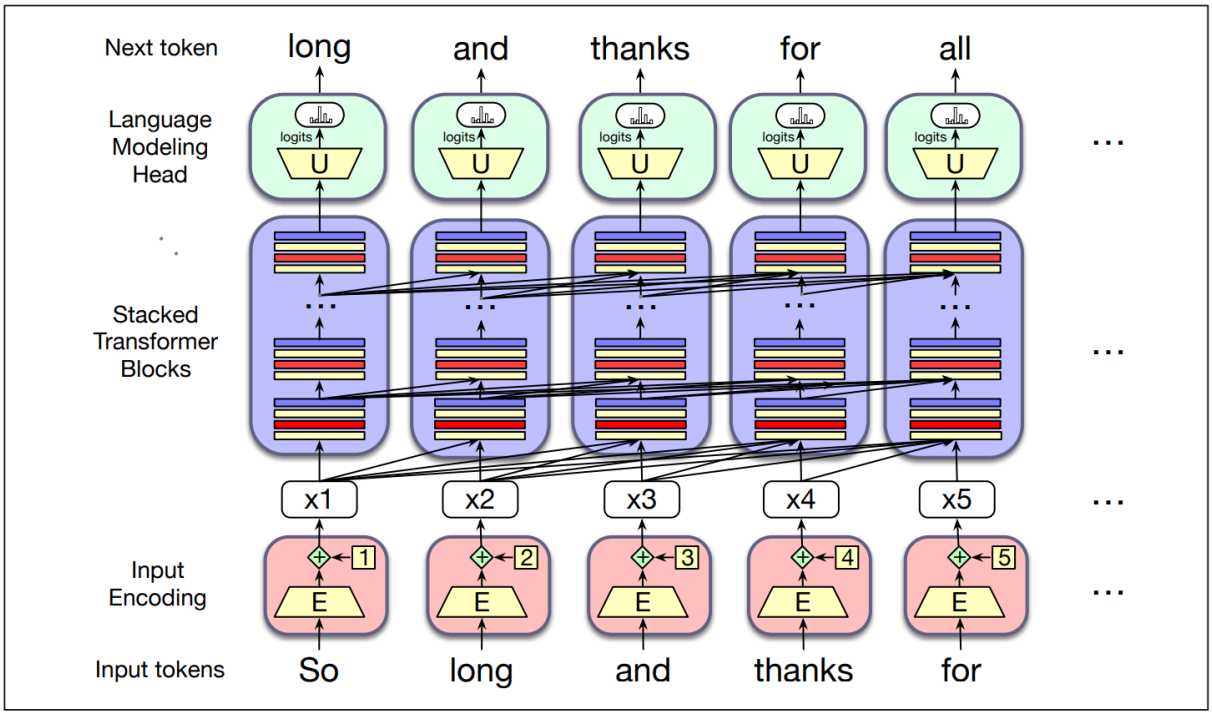
Por outro lado, as redes recorrentes alimentam suas entradas com o próprio resultado de sua saída. Esse comportamento recursivo faz que a resposta da rede para um dado valor de entrada seja dependente de seu estado inicial, que também pode ser dependente de entradas anteriores. Desse modo, as redes recorrentes são capazes de manter uma memória de curto prazo, tornando-se um modelo mais interessante e, também, de mais difícil compreensão, (RUSSELL; NORVIG, 2010).

#### 2.2.4.3 Arquitetura Transformer

A arquitetura transformer é uma rede neural com uma estrutura específica que inclui um mecanismo chamado de *self attention* ou *multi-head attention*. Esse mecanismo consiste em construir uma representação contextual do significado de um token por meio da integração de informações de tokens vizinhos, ajudando o modelo a aprender como esses tokens relacionam-se uns com os outros em entradas relativamente extensas, (JURAFSKY; MARTIN, 2025).

A arquitetura transformer é um modelo de linguagem autorregressivo, também chamado de causal, pois dado uma sequência de tokens de entrada, a predição dos tokens de saída é feita uma por uma, da esquerda para a direita, condicionado no contexto anterior. A figura 3 ilustra essa característica.

**Figura 3** – *Arquitetura de um Transformer autorregressivo*




**Fonte:** Jurafsky e Martin (2025).

De acordo com a acima, a arquitetura transformer possui 3 componentes principais. O primeiro é chamado de *input enconding* e ele é responsável por processar os tokens de entrada e transformá-los em representações vetoriais contextuais. Dessa forma, ocorre a codificação de palavras, como ***thanks***, em vetores numéricos para serem processados pelos componentes subsequentes do transformer. Essa codificação acontece mediante duas etapas:

1. O token de entrada passa por uma matriz de *embedding* $E$, convertendo a forma textual do token para sua representação vetorial de tamanho fixo;
2. Um mecanismo de codificação da posição do token é incorporado ao vetor criado na etapa anterior.

O segundo componente, como é possível visualizar na imagem acima, é conhecido como *Stacked Transformer Blocks* ou blocos empilhados do transformer. Cada bloco é uma rede neural que apresenta uma camada para o mecanismo de *self attention*, camadas de rede de alimentação direta e camadas de normalização. Esse blocos são responsáveis por mapear um vetor de entrada $x_i$ na coluna $i$ (correspondente ao token de entrada $i$) para um vetor de saída $h_i$.

Após o vetor de entrada $x_i$ passar por todos os blocos empilhados desse componente, o vetor resultante é chamado de *embedding output*. No último componente do transformer, denominado de *Language Modeling Head*, o vetor resultante passa por duas etapas:

1. Uma matriz de *unembedding U* é responsável por projetar o *embedding output* de volta ao espaço de vocabulário, produzindo, assim, um vetor denominado de *logits*;
2. Aplica-se a camada de *softmax* sobre o vetor *logits*. O retorno é uma distribuição de probabilidades sobre o vocabulário.

Dessa maneira, a arquitetura transformer pode escolher a palavra com maior probabilidade para ser a sucessora do respectivo token de entrada. Isso demonstra a capacidade preditiva dessa arquitetura e o seu potencial em aplicações de PLN.

#### 2.2.4.4 Exemplos de Modelos Transformer

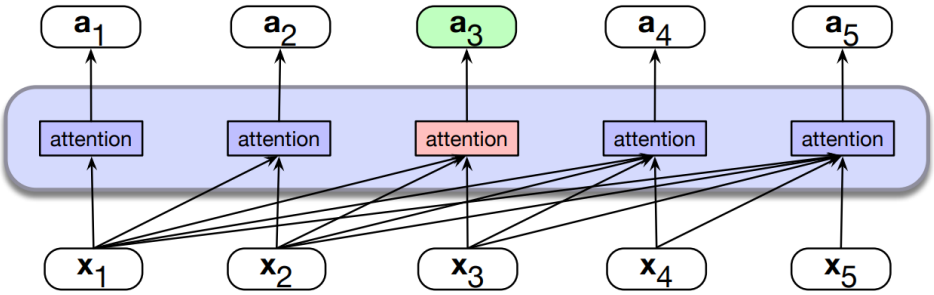
A arquitetura transformer foi a base para a criação de inúmeros modelos de PLN que estão presentes em aplicações de inteligência artificial em todo o mundo. Por isso, é essencial compreender os principais exemplos de modelos transformer existentes hoje.

Um desses modelos é conhecido como BERT (*Bidirectional Encoder Representations from Transformers*). Segundo Devlin et al. (2018), diferentemente dos modelos unidirecionais desenvolvidos até então, esse modelo é capaz de realizar a codificação contextual de forma bidirecional, isto é, permite que cada *token* observe simultaneamente seus contextos à esquerda e à direita. Essa característica viabiliza maior compreensão semântica da sentença.

Em sua arquitetura, o BERT é composto por diversas camadas (*Enconders*), sendo que cada uma dessas camadas apresenta o mecanismo de *multi-head attention*, que quando empilhadas, são capazes de processar representações distribuídas de forma paralela e eficiente. Isso possibilita o aprendizado de padrões linguísticos complexos, fundamentais para tarefas de PLN.

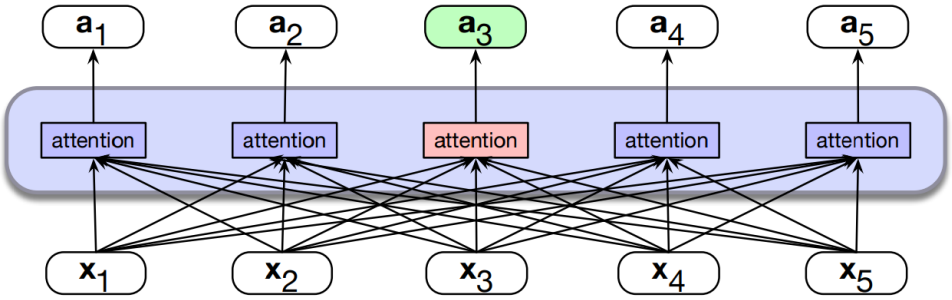
Segundo Jurafsky e Martin (2025), o modelo BERT se difere do Transformer autorregressivo em dois pontos principais. O primeiro é no mecanismo de atenção, visto que no BERT esse mecanismo é capaz de computar o contexto de tokens anteriores e posteriores ao token atual. Já no modelo autorregressivo, o contexto é computado somente pelos tokens antecessores ao atual, ignorando informações potencialmente úteis localizada à direta do token sendo processado.

**Figura 4** – *Camada de Atenção do Transformer autorregressivo*




**Fonte:** Jurafsky e Martin (2025).

**Figura 5** – *Camada de Atenção do BERT*




**Fonte:** Jurafsky e Martin (2025).

De acordo com o estudo de Devlin et al. (2018), o modelo BERT apresenta duas etapas principais, sendo elas o pré-treinamento e o *fine-tuning*. A primeira etapa utiliza duas abordagens não supervisionadas para pré-treinar o modelo. Sendo elas:

- *Masked Language Modeling (MLM)*: Nesta fase cerca de 15% dos tokens em cada sequência são aleatoriamente mascarados com um token especial $[MASK]$, de modo que o modelo deve prever os tokens originais a partir do contexto bidirecional;
- *Next Sentence Prediction (NSP)*: Essa fase consiste em treinar o modelo a partir de duas sentenças A e B. O objetivo do treinamento é descobrir se a sentença B realmente é ou não a próxima sentença de A. Essa fase é importante para tarefas de *Question Answering (QA)* e *Natural Language Inference (NLI)*.

O pré-treinamento do BERT foi realizado com duas bases de dados de grande volume, sendo o *BookCorpus* e *English Wikipedia* que juntos somam cerca de 16GB de dados não compactados de texto.

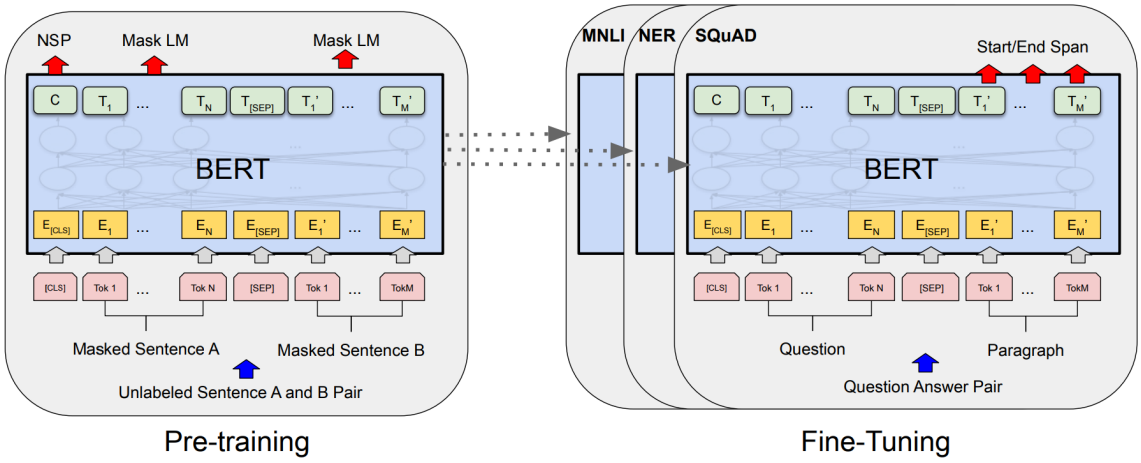
Para a etapa de *fine-tuning*, o modelo é inicializado com os parâmetros do pré-treinamento e, em seguida, todos os parâmetros são ajustados (*fine-tuned*) de acordo com

dados rotulados da tarefa específica que ele deve executar. O objetivo dessa fase é treinar o modelo para executar tarefas específicas de PLN.

Em comparação com o pré-treinamento, essa etapa é considerada sem custos, pois os resultados de seu processamento podem ser obtidos em poucas horas com o uso de uma GPU. O artigo de Devlin et al. (2018) fez o *fine-tunning* por meio da entrada de pares de sentenças $A$ e $B$ considerando a tarefa a ser executada pelo modelo.

Por exemplo, para a tarefa de implicação, a sentença $A$ é a hipótese e a sentença $B$ é a premissa. Já para tarefas de resposta a perguntas, a sentença $A$ seria a pergunta e a sentença $B$ seria sua resposta. Dessa forma, o modelo é capaz de ser ajustado para atingir objetivos específicos de PLN.

**Figura 6** – *Fases de Pré-Treinamento e Fine tuning do BERT*




**Fonte:** Devlin et al. (2018).

Outro exemplo de modelo Transformer é o *Robustly Optimized BERT Pretraining Approach*, também conhecido como RoBERTa. Segundo o estudo de Liu et al. (2019), foram feitas modificações na fase de pré-treinamento do modelo BERT que contribuíram significativamente para um melhor desempenho em relação ao modelo BERT original. Dentre dessas modificações, pode-se citar:

1. Treinar o modelo com um maior volume de dados e por mais tempo. Para isso usou-se um conjunto de dados de 160GB de texto não compactado. Ou seja, um conjunto 10 vezes maior que o usado no BERT original;

2. Treinar o modelo com sequências maiores de tokens (conhecido como *batch size*);

3. Remover a fase de *Next Sentence Prediction* do pré-treinamento;

4. Utilização de máscaras dinâmicas na fase de *Masked Language Modeling*, pois, assim, a cada nova leitura de um texto, palavras diferentes são mascaradas, melhorando a diversidade no treinamento.

Portanto, por meio do ajuste de certas características na fase de pré-treinamento, foi possível criar um modelo de maior desempenho em tarefas específicas de PLN.

Nesta pesquisa, o modelo de aprendizagem que máquina que será utilizado para fazer a análise de sentimentos das notícias financeiras será um modelo RoBERTa multilíngue denominado de XLM-RoBERTa.

De acordo com o artigo de Conneau et al. (2020), esse modelo foi treinado com 2,5TB de texto usando o corpus *CommonCrawl Multilingual* (CC100), que inclui grandes quantidades de dados *Web* em 100 idiomas diferentes. Além disso, ele foi projetado para funcionar bem sem ajuste específico por idioma, sendo eficaz em tarefas de PLN multilíngue.

Pelo fato de ser um modelo multilíngue e funcional para o idioma português, as notícias coletadas não precisarão ser traduzidas antes de serem introduzidas no modelo. Dessa forma, as inconsistência presentes na tradução serão evitadas.

## 2.3 Estudos Correlatos

No trabalho de Patil, Sharma e Sinha (2024) a coleta dos dados financeiros foi feita utilizando vários portais de notícias da área. Entre eles, é possível citar o Finviz, que foi utilizado para a coleta de títulos de notícias financeiras por meio de *Web Scraping* [9], usando a biblioteca Python chamada *Beautiful Soup*. Os títulos coletados foram de ações como Tesla (TSLA), Google (GOOG) e o índice norte-americado S&P 500.

O sistema proposto no estudo foi a criação de uma base de dados que inicialmente inclui: o histórico dos dados das ações, os títulos das ações a serem analisadas e, também, uma rotulação inicial (positivo, negativo ou neutro) desses títulos feita pela ferramenta *NLTK (Natural Language Toolkit)*. Depois, toda essa base de dados foi incluída nos modelos de aprendizagem de máquina estudados na pesquisa. Dentre eles temos o *LSTM (Long Short-Term Memory)*, que é um modelo pertencente à classe de Redes Neurais Recorrentes (RNN) muito usado em previsão de séries temporais.

Além disso, o trabalho aborda o modelo *ARIMA (Autoregressive Integrated Moving Average)* e o modelo *Random Forest*. Os resultados mostram que tanto o modelo *LSTM* quanto *ARIMA* alcançaram um alto nível de assertividade nas previsões das movimentações de Tesla e S&P 500, superando o modelo *Random Forest* com ampla vantagem.

[9] *Web Scraping* é a técnica de extração automática de dados de sites da internet por meio de scripts ou ferramentas especializadas.

De acordo com a pesquisa de Maqbool et al. (2023), um modelo de aprendizagem de máquina chamado de *MLP-Regressor* foi proposto com o intuito de prever as tendências futuras de quatro ativos alvos: Reliance, Tata Motors, Tata Steel e HDFC. Esse modelo consegue fazer a previsão da tendência do próximo dia usando as pontuações de preço e sentimento do dia anterior e, em seguida, comparando os preços do dia seguinte com os do dia anterior. Também consegue prever tendências futuras, comparando os preços das ações de um dia com o preço após n dias. Para o modelo fazer essas comparações ele tem como entrada uma base de dados semelhante a do estudo anterior, ou seja, uma união dos sentimentos das notícias financeiras coletadas com os dados históricos da ação analisada.

Os sentimentos foram atribuídos às notícias por meio de 3 ferramentas automatizadas: *VADER (Valance Aware Dictionary and Sentiment Reasoner)* que faz parte do *NLTK*, *TextBlob* e *Flair*. E os dados históricos foram extraídos do *Yahoo Finance*. O estudo também usou métricas para avaliar o desempenho do modelo proposto, como o cálculo de acurácia, precisão e o do *MAPE (Mean Absolute Percentage Error)*, que expressa o erro absoluto como uma porcentagem do valor real. As métricas foram analisadas em períodos de 10, 30 e 100 dias para todas as combinações das ferramentas de sentimento. Os resultados demonstram que o uso de sentimentos de notícias financeiras juntamente com o *MLP-Regressor* permite prever o preço das ações com uma precisão de 0,90 e mostra uma alta correlação entre o preço das ações e as notícias financeiras.

Além disso, o trabalho de Agrawal e Mukherjee (2025) afirma que métodos antigos, baseados apenas em indicadores técnicos para a previsão dos preços de ações, apresentam desempenho inferior em comparação com aqueles que consideram o sentimento contido em notícias financeiras e opiniões de investidores. Dessa forma, a pesquisa integrou os sentimentos extraídos de notícias financeiras e redes sociais com modelos avançados de aprendizagem de máquina afim de aprimorar a previsão da tendência do mercado acionário.

Um dos objetivos da pesquisa é comparar cinco desses modelos, sendo eles: *LSTM*, *Random Forest*, *Logistic Regression*, *BERT* e *SVM (Support Vector Machine)*, visando investigar suas diferenças e o seu poder preditivo em um conjunto de dados aumentado por sentimento. Para isso, utilizou-se múltiplas bases de dados diferentes, como: *Combined_News_DJIA*, *RedditNews* e *DJIA Stock Prices*. Todos os dados coletados passaram por pré-processamento, processamento em um dos cinco algoritmos citados, engenharia de recursos, ajuste de parâmetros e otimização de modelos.

A comparação foi feita por meio do uso de métricas avaliativas (acurácia, precisão, *F1 score* e *Recall*) e testes estatísticos. Os resultados mostram que o modelo *BERT* sobressaiu-se em relação aos demais, atingindo uma acurácia de 98,92%. O modelo *Logistic Regression*, por ser o mais simples, teve o menor desempenho, mas ainda conseguiu atingir resultados robustos como uma acurácia de 93,94%.

Na sequência, no quadro 1, são elencados de maneira resumida, os autores, as fontes

de dados utilizadas, os modelos e objetivos dos estudos correlatos analisados nesta seção.

**Quadro 1** – *Comparação entre estudos correlatos e esta pesquisa*

| Estudo | Fonte dos dados | Modelos utilizados | Objetivo |
|---|---|---|---|
| Patil, Sharma e Sinha (2024) | Títulos de notícias financeiras (Finviz), histórico de ações (TSLA, GOOG, S&P 500) | LSTM, ARIMA, Random Forest | Criar uma base de dados combinando notícias e preços de ações para prever tendências com modelos de ML |
| Maqbool et al. (2023) | Yahoo Finance + sentimentos de notícias financeiras (Reliance, Tata Motors, etc.) | MLP-Regressor | Prever tendências futuras de ações com base em sentimentos e preços anteriores |
| Agrawal e Mukherjee (2025) | Combined News DJIA, RedditNews, DJIA Stock Prices | BERT, LSTM, Random Forest, SVM, Logistic Regression | Comparar modelos de ML baseados em sentimento para prever movimentos do mercado acionário |
| Esta Pesquisa | Infomoney, Seu dinheiro, Suno notícias, Money Times, Uol Economia + Yahoo Finance | Diocionário L&M, XLM-RoBERTa | Comparar os dois modelos utilizados com base na correlação de seus resultados com a movimentação dos preços dos ativos PETR3 e PETR4 no mesmo período |

Analisando os estudos apresentados no quadro 1, o estudo desta pesquisa diferencia-se dos estudos supracitados, pois seu enfoque está na comparação entre dois modelos de AS, um baseado no dicionário L&M e outro que faz uso de técnicas avançadas de aprendizagem de máquina. Embora todos os modelos utilizados nos estudos correlatos são de aprendizagem de máquina, esta pesquisa visa mostrar as diferenças classificatórias e de correlação entre dois modelos bem distintos de AS.

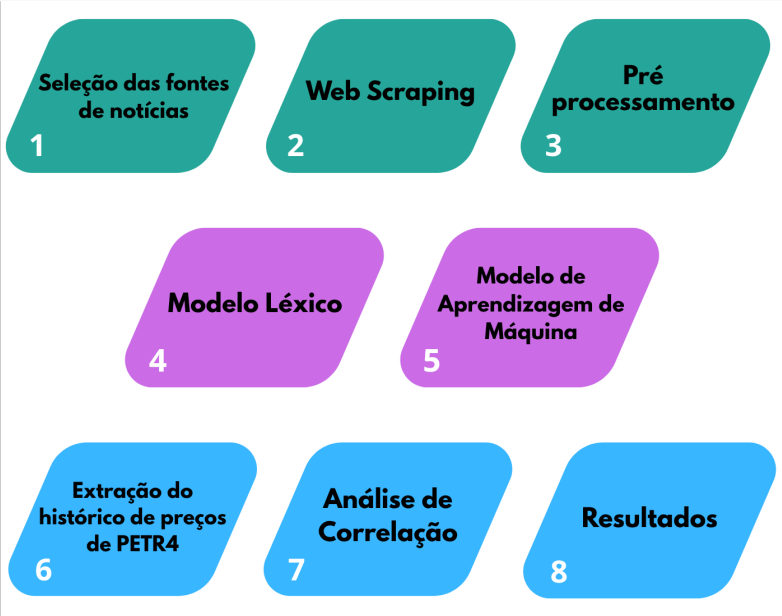
Além disso, esta pesquisa assemelha-se às demais por também fazer comparações entre modelos de AS, buscando evidenciar a influência e o potencial preditivo das notícias financeiras sobre o mercado acionário.

# 3 Métodos e Ferramentas

## 3.1 Metodologia

A metodologia adotada por esta pesquisa foi criada a partir da junção dos conceitos fundamentais de AS e Mineração de Dados na *Web*. As etapas foram ilustradas (figura 7) para facilitar a compreensão das ações realizadas ao longo do desenvolvimento.

**Figura 7** – *Etapas de desenvolvimento da pesquisa*




**Fonte:** Elaborado pelo autor.

A enumeração das etapas do desenvolvimento presente nos pacotes do diagrama da figura 7 representam apenas a ordem cronológica em que as ações foram implementadas.

Pode-se considerar que a presente pesquisa terá 3 fases principais em seu desenvolvimento. A primeira fase está marcada em verde na figura 7 e consiste nas seguintes etapas:

- Seleção das fontes de notícias: Etapa responsável pela escolha dos noticiários online onde serão extraídos as notícias financeiras relacionadas à Petrobras;
- *Web Scraping*: Etapa dedicada à coleta propriamente dita das notícias selecionadas de forma automatizada;
- Pré-processamento: Etapa onde serão aplicadas técnicas para o tratamento dos dados coletados. As técnicas aplicadas seguirão a seguinte ordem: remoção de

caracteres irrelevantes e símbolos desnecessários, remoção de *stopwords* e, por fim, *lemmatization*.

A segunda fase está marcada pela cor rosa na figura 7 aborda tanto o desenvolvimento quanto a aplicação dos modelos de AS nos dados já previamente ajustados. Pode-se dividir essa fase em duas etapas, sendo cada uma dedicada para um modelo específico, sendo elas:

- Modelo Léxico baseado no dicionário L&M;
- Modelo de Aprendizagem de Máquina pré-treinado chamado de XLM-roBERta.

Por fim, a terceira fase, marcada pela cor azul na figura 7, corresponde às três etapas finais do desenvolvimento desta pesquisa, sendo elas:

- Extração do histórico de preços dos ativos analisados: Etapa encarregada de coletar o histórico de preços da cotação dos ativos PETR3 e PETR4 na faixa de tempo analisada. Usou-se a plataforma do *Yahoo Finance* para atender esse requisito;
- Análise de correção dos modelos de AS: Etapa cuja finalidade é comparar os sentimentos das notícias e o preço do ativo no mesmo período de tempo, com intuito de verificar a influência que as notícias têm sobre a mudança do valor negociado da ação. Isto é, verificar qual foi a capacidade preditiva das notícias com base em seus sentimentos.
- Análise dos Resultados: Etapa incumbida de analisar os resultados da correlação apresentada por ambos os métodos, com o intuito de comparar suas performances e, assim, chegar a conclusões embasadas sobre a eficácia dos métodos utilizados.

## 3.2 Ferramentas

Aqui serão descritos todos os materiais necessários para o desenvolvimento desta pesquisa. Isso inclui os elementos físicos onde foi implementada a análise (hardware) e, também, os componentes de software utilizados, como linguagem de programação, bibliotecas e ambiente de desenvolvimento.

### 3.2.1 Componentes de Hardware

#### 3.2.1.1 Computador utilizado

A implementação dos modelos de AS requer uma máquina com capacidade para executar essas tarefas. Sendo assim, toda a pesquisa será desenvolvida em um computador com as seguintes especificações:

- Processador Rysen 5 5600G
- Memória RAM de 32GB
- Armazenamento SSD de 512 GB
- Sistema Operacional Windows 10 de 64 bits

### 3.2.2 Componentes de Software

#### 3.2.2.1 Linguagem de Programação Python

Python é uma linguagem de programação de alto nível amplamente utilizada por sua simplicidade e legibilidade. Ela possui uma biblioteca padrão ampla e uma comunidade ativa que constantemente contribuem com novos módulos e *frameworks*.

Ela é uma linguagem interpretada, multiparadigma, dinamicamente tipada, pois não exige a declaração de tipos e, também, fortemente tipada, visto que não permite operações entre tipos incompatíveis sem conversão explícita. É uma linguagem muito favorável para trabalhar com aplicações envolvendo inteligência artificial devido a quantidade de bibliotecas de qualidade disponíveis.

#### 3.2.2.2 Principais Bibliotecas Utilizadas

No decorrer do desenvolvimento deste trabalho, algumas bibliotecas foram utilizadas para facilitar atividades de extração e manipulação de dados, treinamento e execução de modelos de PLN. Abaixo estão descritas as principais bibliotecas adotadas neste pesquisa.

**BeautifulSoup** é uma biblioteca em Python usada para extração de dados de páginas HTML, ou seja, *Web Scraping*. Ela fornece ferramentas simples para navegar pela estrutura do documento, buscar elementos por *tags*, classes ou atributos e manipular o conteúdo de forma prática. O uso dessa ferramenta facilitou a extração dos dados dos diversos portais financeiros.

**Pandas** é uma biblioteca voltada para análise e manipulação de dados em Python. Ela oferece estruturas de dados eficientes, como os *DataFrames*, que facilitam o trabalho com tabelas de dados.

**spaCy** é uma biblioteca de PLN desenvolvida para ser rápida e prática em aplicações reais. Ela vem com modelos de linguagem já treinados que aplicam diversas funcionalidades de PLN. No contexto desta pesquisa, as funcionalidades utilizadas foram inclinadas para realizar o pré-processamento. Assim, esta biblioteca foi responsável por realizar a *tokenização* e a *lemmatização* do texto bruto previamente coletado.

A **Hugging Face** é uma empresa e comunidade de tecnologia voltada para inteligência artificial. Ela desenvolveu várias bibliotecas que aceleram o desenvolvimento de aplicações em IA e que foram utilizadas nesta pesquisa. A *Transformers* permite o carregamento modelos pré-treinados prontos para uso e *fine-tuning*, enquanto a *Datasets* facilita o acesso e manipulação de grandes conjuntos de dados. Já a *Tokenizers* oferece ferramentas rápidas e eficientes para preparar texto antes do treino, e a *Evaluate* ajuda a calcular métricas padronizadas de avaliação do treinamento.

A **Matplotlib** é uma biblioteca do Python voltada para a criação de visualizações gráficas, permitindo gerar desde gráficos simples, como linhas e barras, até representações mais complexas, como histogramas, dispersões e figuras tridimensionais. Dessa forma, essa biblioteca foi amplamente utilizada na criação dos gráficos refente a seção de resultados (seção 5).

#### 3.2.2.3 *Google Colaboratory*

O *Google Colaboratory* é uma plataforma de desenvolvimento gratuita fornecida pelo Google que permite a execução de notebooks Jupyter hospedados. Ou seja, os usuários podem usar recursos computacionais diretamente pela nuvem, sem precisar dispor dos recursos de sua própria máquina para desenvolver suas aplicações.

Ela é uma plataforma ótima para estudantes, pois permite integrar código e texto no mesmo ambiente, permitindo clareza nas atividades desenvolvidas. Além disso, é amplamente usado em projetos de ciência de dados e aprendizado de máquina e, por isso, foi escolhido para integrar essa pesquisa.

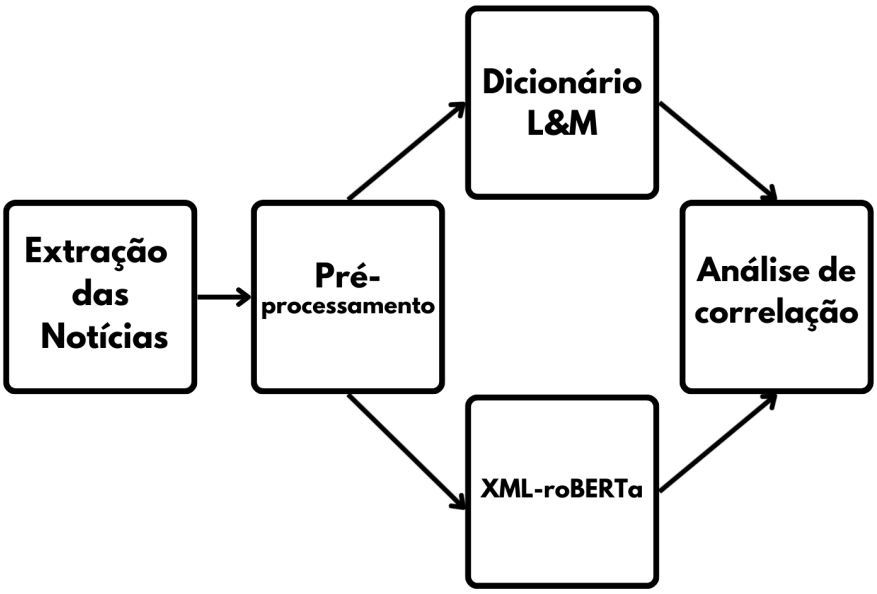
#### 3.2.2.4 *Yahoo Finance*

O Yahoo Finance é uma plataforma online que oferece informações financeiras atualizadas sobre inúmeros ativos do mercado acionário. Ele será utilizado para coletar os dados necessários das ações PETR3 e PETR4 na presente pesquisa.

# 4 Desenvolvimento

A execução do desenvolvimento aconteceu em 4 fases majoritárias. Descritas pelo *pipeline* apresentado na figura 8.

**Figura 8** – *Pipeline de implementação do desenvolvimento*




**Fonte:** Elaborado pelo autor.

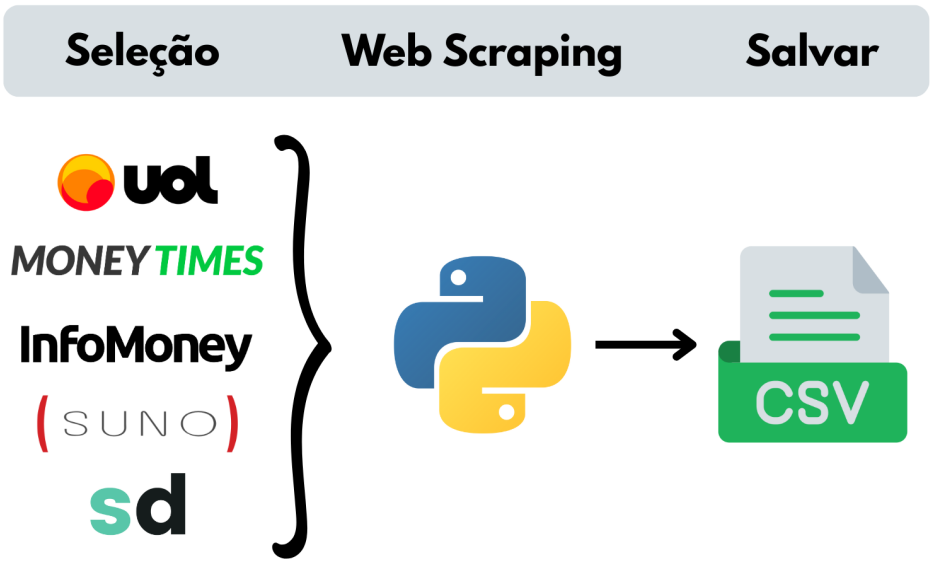
De acordo com o *pipeline* de execução dos experimentos (figura 8), na primeira fase, foram coletadas as notícias referentes à empresa Petrobras (PETR3 e PETR4) de diversos portais financeiros diferentes. Adiante, implementou-se o pré-processamento do conteúdo dessas notícias, de modo a adequar o texto para os modelos seguintes. As técnicas de AS vieram na sequência, rotulando o conteúdo textual com os seus sentimentos correspondentes (positivo, negativo e neutro). E, por fim, a análise de correlação foi encarregada de comparar o sentimento atribuído às notícias com os preços do ativo no mercado de ações. Estas 4 fases são descritas na sequência.

## 4.1 Extração das notícias

Três etapas foram incluídas nessa fase inicial do desenvolvimento: Seleção das notícias, Filtragem das notícias por período de tempo e o *Web Scraping* (figura 9). Esta última pode ser considerada a extração propriamente dita, realizada por meio da criação

de *scripts*, com o uso da biblioteca *BeautifulSoup*, capazes de retirar o conteúdo presente nas *tags* HTML de páginas *Web*.

**Figura 9** – *Pipeline da fase de extração*




**Fonte:** Elaborado pelo autor.

### 4.1.1 Seleção

Esta etapa consistiu em selecionar os principais sites financeiros nacionais que trazem notícias relevantes sobre a Petrobras e seus ativos PETR3 e PETR4. Dentre os sites escolhidos, foram extraídas 2.909 notícias no total.

Diversos portais financeiros foram analisados nesta etapa, contudo não foi possível incluir uma maior quantidade devido ao fato de serem sites por assinatura, o que impediu a captura de uma quantidade maior de notícias.

A tabela abaixo especifica os sites explorados para realizar a extração das notícias e, também, informa a quantidade extraída de cada site por ano de análise.

**Tabela 2** – *Quantidade de notícias extraídas de cada site*

| **Sites Financeiros** | **2020** | **2021** | **2023** | **2024** |
|---|---|---|---|---|
| Infomoney | 10 | 24 | 42 | 93 |
| Seu Dinheiro | 177 | 198 | 113 | 104 |
| Suno Notícias | 270 | 304 | 292 | 385 |
| Money Times | 238 | 220 | 221 | 165 |
| Uol Economia | 25 | 6 | 22 | 0 |

### 4.1.2 Filtragem

Como visto na tabela 2, foram extraídas notícias de quatro anos diferentes: 2020, 2021, 2023 e 2024. Essa escolha baseou-se no fato de investigar como as notícias da Petrobras impactaram a movimentação de suas ações em dois períodos diferentes: pandemia e pós-pandemia.

Nos anos de 2020 e 2021, o mundo sofreu com a pandemia da Covid-19 e, com isso, o mercado financeiro teve comportamentos únicos que foram influenciados pela crise vivenciada, (SCHRANK, 2024). Já nos anos de 2023 e 2024, o Brasil vivenciou um período relativamente estável e sem tantas turbulências quando comparado ao período anterior.

Dessa forma, aplicou-se a AS em dois períodos economicamente distintos com o intuito de avaliar como as notícias extraídas impactaram o mercado dos ativos PETR3 e PETR4.

Feita a apuração dos portais de notícias e estabelecido o intervalo a ser analisado, prosseguiu-se com o filtro temporal na etapa de captura e extração das notícias.

Nesse contexto, como foi possível visualizar na tabela 2, os sites nacionais com o maior número de notícias extraídas foram: *Suno Notícias*, *Money Times* e *Seu Dinheiro*. Isso ocorreu devido ao fácil acesso que eles oferecem aos usuários para encontrar notícias de anos anteriores.

Cada um desses sites conta com um menu destinado a empresas e negócios. Neste menu, são publicadas notícias de várias corporações, incluindo a Petrobras. Além disso, o que permitiu localizar as notícias de anos anteriores foi a paginação presente em sua *URL*. Isto é, o menu é separado em várias *URLs* indexadas, como é possível ver no exemplo do site *Suno Notícias*:

```
https://www.suno.com.br/noticias/negocios/page/2/#ultimas-noticias
```

A numeração que está na frente do termo *page* representa a indexação das páginas. Desta maneira, ao modificar seu valor, as notícias puderam ser localizadas no período de interesse.

Entretanto, para os sites *Uol Economia* e *Infomoney* foi encontrada uma dificuldade em fazer a filtragem de tempo dentro de suas plataformas, visto que o menu destinado às empresas e negócios não possuem paginação. Assim, todas as notícias ficam presentes em um mesma *URL*, o que tornou a tarefa de localizar as notícias inviável.

Entretanto, uma solução adotada para esse problema foi fazer uma busca no próprio *Google Browser* com o seguinte formato:

```
Petrobras site:domínio_site after:data_início before:data_fim
```

O termo principal dessa busca é *Petrobras*. Isto é, o *Google* irá retornar somente resultados de sites que contenham a palavra *Petrobras* em seu conteúdo.

Já o campo *site:domínio_site* está restringindo a pesquisa a somente um domínio específico, como por exemplo, *site:infomoney.com.br* indica que a busca somente retornará notícias vindas do portal *infomoney.com.br*.

Os últimos dois campos da busca: *after* e *before* filtram os resultados para mostrar apenas páginas publicadas após a data indicada em *after* e antes da data indicada em *before*. O formato de data aplicado nessa busca foi: *YYYY-MM-DD* (ano-mês-dia).

Nesse contexto, quando juntou-se todos esses elementos, a pesquisa retornou notícias sobre a *Petrobras* dentro de um intervalo de tempo específico, vindas apenas de um site escolhido.

## 4.2 *Web Scraping*

Com a utilização das duas formas de filtragem, fez-se uma seleção manual das *URLs* das notícias mais relevantes sobre a Petrobras. Essa etapa foi fundamental, pois permitiu excluir matérias em que a Petrobras não era o foco principal, já que frequentemente eram abordadas diversas empresas em um mesmo conteúdo. Assim, para evitar inconsistências nas fases de atribuição de sentimentos e nas análises subsequentes, foram consideradas apenas as notícias em que a Petrobras figurava como tema central.

Além disso, criou-se uma lista de *URLs* para cada site financeiro analisado. Isso foi necessário, uma vez que os sites financeiros não tem a mesma estrutura de *tags* HTML. Dessa maneira, foi preciso criar *scripts* específicos para cada portal.

Os *scripts* de *Web Scraping* foram implementados com o uso da biblioteca *BeautifulSoup*. Para cada notícia (*URL*), os seguintes atributos foram extraídos na forma de um dicionário em Python: URL, Título, Conteúdo, Data de Publicação e Nome do site.

Nesse contexto, para implementar a extração de todas as notícias selecionadas, criou-se uma lista vazia (*noticias = []*) que agregou todo o conjunto de notícias de cada portal. Assim, para o primeiro portal, percorreu-se sua lista de *URLs* e, para cada *URL* foi aplicado a função de extração de dados (*extrair_dados_site(url: str)*). Ao final, o conteúdo de todas as notícias desse portal estava presente na lista *noticias*.

Esse processo foi executado para todos os portais existentes. Dessa forma, a lista agregadora armazenou os principais atributos da totalidade das notícias. Além disso, a lista foi convertida em um *Dataframe* pandas e, finalmente, salva como um arquivo CSV. O bloco A, presente na seção de apêndices, demostra como foi realizada a extração de notícias do site *infomoney*.

## 4.3 Pré-processamento

Essa fase, como elucidado na seção 2.2.2, possui três abordagens principais para o tratamento do texto de entrada, sendo elas: Limpeza Textual, Remoção de *stopwords* e *Lemmatização*.

Todas as abordagens supracitadas foram implementadas com o auxílio das seguintes bibliotecas:

- **re**: Biblioteca de expressões regulares. Ela foi aplicada para fazer a limpera textual, mediante a remoção de quebras de linhas e tabs, remoção de múltiplos espaços e caracteres especiais;
- **spaCy**: Biblioteca responsável por fazer a *tokenização* e *lemmatização*. Isso foi feito por meio do carregamento do modelo *pt_core_news_sm*.
- **pandas**: Os dados a serem pré-processados foram lidos de um arquivo CSV e convertidos para um *Dataframe*, onde foram manipulados e salvos. Nesse ínterim, torna-se indispensável o uso desta ferramenta.

A etapa de remoção de *stopwords* foi feita manualmente, haja visto que as listas de *stopwords* fornecidas pelas bibliotecas de PLN contém palavras importantes que, quando removidas, prejudicariam a análise. Por isso, criou-se uma lista própria, contendo somente termos cuja semântica não tem importância alguma. Por exemplo, pronomes, artigos, preposições e conjunções.

O próximo passo envolveu a implementação da classe *NewsPreprocessor*. Dentro dela há uma função de inicialização (*def ___init___(self)*) onde o *spaCy* carregou o modelo designado e, também, existe a função de limpeza do textal (*def clean_text(text)*). Entretando, a função *preprocess_text(text)* foi a responsável por fazer efetivamente o pré-processamente, visto que fez uso da função de limpeza textual, realizou a remoção de *stopwords* e, também, executou a *lemmatização*.

Para finalizar, foi feito a conversão do arquivo CSV das notícias coletas para um *Dataframe*. Depois, percorreu-se a coluna "*conteudo*" do arquivo e, para cada linha, aplicou-se a função *preprocess_text(text)* e o seu retorno foi armazenado em outras duas colunas, chamadas de: *conteudo_tokens* e *conteudo_processado*. Finalmente, o *Dataframe* modificado foi salvo como outro arquivo CSV. A implementação da função de pré-processamento pode ser visualizada no apêndice B.

## 4.4 Modelos de AS

### 4.4.1 Modelo de Dicionário L&M

Como já explicado na seção 2.2.3, para utilizar o dicionário L&M em notícias nacionais, foi necessário fazer a sua tradução. Para isso, utilizou-se a API de tradução *DeepL*. Essa API oferece um plano gratuito para desenvolvedores e apresentou um resultado mais eficiente na tradução do que a API gratuita do Google Tradutor.

Em primeiro plano, foi preciso filtrar as colunas *positivas* e *negativas* do dicionário original. Depois, notou-se que muitas das palavras presentes nessas duas colunas são derivações. Por exemplo, as palavras *accomplished*, *accomplishes*, *accomplishing* e *accomplishment* são derivações da palavra *accomplish*.

Por causa das possíveis inconsistências na tradução dessas derivações, o processo de *lematização* foi incorporado nas listas de palavras positivas e negativas do dicionário original. Ao final, o número total de palavras positivas foi reduzido de 354 para 232 e de 2355 para 1382 no total de palavras negativas.

O próximo passo foi usar a API da *DeepL* para realizar a tradução. Na sequência, o novo dicionário foi salvo como um arquivo CSV. A primeira coluna deste novo dicionário contém a palavra original em inglês, a segunda coluna contém a palavra traduzida e a terceira o sentimento da palavra (positivo ou negativo).

**Figura 10** – *Exemplo do dicionário traduzido*

| A | B | C |
|---|---|---|
| solve | solucionar | positivo |
| winner | vencedor | positivo |
| decreased | diminuiu | negativo |
| disturbance | perturbação | negativo |

**Fonte:** Elaborado pelo autor.

A segunda fase da implementação consistiu em fazer a pontuação de sentimento das notícias extraídas mediante a contagem de palavras.

Primeiramente, a partir da importação do dicionário traduzido, uma lista das palavras positivas presentes no dicionário foi criada, juntamente com outra lista contendo as palavras negativas. Depois, o arquivo das notícias pré-processadas foi utilizado para fazer a contagem da quantidade de palavras positivas e negativas presente em cada notícia

(bloco 1).

**Bloco 1** – *Implementação da contagem das palavras positivas*

```
news_df["num_pos"] = news_df["conteudo_tokens"].apply(
    lambda tokens: sum(1 for token in tokens if token in
       palavras_positivas))
```

A métrica empregada para calcular a pontuação de sentimentos foi desenvolvida a partir da normalização das palavras identificadas com sentimentos positivos e negativos. A fórmula 4.1 exemplifica essa métrica:

$$P = \frac{POSITIVAS - NEGATIVAS}{POSITIVAS + NEGATIVAS} \tag{4.1}$$

Finalmente, a partir do cálculo da pontuação $P$, o sentimento (positivo, negativo ou neutro) foi atribuído à notícia correspondente por meio da comparação com um valor de corte ($CUTOFF$).

Dessa forma, os valores de $P$ maiores que o $CUTOFF$ resultou em notícias positivas. Se $P$ esteve no intervalo $[-CUTOFF, CUTOFF]$, a notícia assumiu o valor neutro, um sentimento adicional que não estava presente no dicionário original. Já quando $P$ foi menor que $-CUTOFF$, o sentimento atribuído a respectiva notícia foi negativo.

O valor de $CUTOFF = 0,3$ foi atribuído para reduzir classificações incorretas na região próxima ao neutro, criando uma zona neutra mais robusta, segundo a lógica apresentada no estudo de Iqbal, Karim e Kamiran (2015).

Com isso, esse método garantiu a correta rotulação das notícias nas três classes definidas: positivo, negativo e neutro.

**Bloco 2** – *Implementação da pontuação de sentimentos*

```
CUTOFF = 0.3
news_df["sentimento"] = news_df["P"].apply(
    lambda x: "positivo" if x > CUTOFF else "negativo" if x < -
       CUTOFF else "neutro")
```

A figura 11 mostra um exemplo de resultado da aplicação do modelo de dicionário no arquivo das notícias pré-processadas. As colunas *conteudo_tokens* e *conteudo_processado* foram geradas na etapa de pré-processamento e as demais colunas foram geradas pela execução do modelo. A coluna *num_token* mostra a quantidade de *tokens* presente na notícia, já *num_pos* e *num_neg* contém o somatório de palavras positivas e negativas que foram correlacionadas ao dicionário, respectivamente. A coluna *lm_score* apresenta o resultado do cálculo da pontuação de sentimento e *lm_sentimento* aplica a rotulação da respectiva notícia.

**Figura 11** – *Arquivo das notícias processadas pelo modelo L&M*

| conteudo_tokens | conteudo_processado | num_token | num_pos | num_neg | lm_diff | lm_score | lm_sentimento |
|---|---|---|---|---|---|---|---|
| [crise, coronavírus, reduzir, drasticamente, d... | crise coronavírus reduzir drasticamente demand... | 171 | 7 | 15 | -8 | -0.363636 | negativo |
| [uma, vez, aversão, risco, mercado, ofuscar, t... | uma vez aversão risco mercado ofuscar totalmen... | 451 | 8 | 25 | -17 | -0.515152 | negativo |
| [correção, contrário, informado, anteriormente... | correção contrário informado anteriormente açã... | 632 | 21 | 39 | -18 | -0.300000 | neutro |
| [petrobras, estar, recorrer, tecnologia, conti... | petrobras estar recorrer tecnologia continuar ... | 298 | 10 | 17 | -7 | -0.259259 | neutro |

**Fonte:** Elaborado pelo autor.

### 4.4.2 Modelo XML-roBERTa

O modelo de aprendizagem de máquina implementado nesta pesquisa (XML-roBERTa[1]) é disponibilizado pela plataforma *Hugging Face* que mediante sua grande variedade de bibliotecas, tornou-se factível o seu uso no escopo deste trabalho.

Como já foi explicado na seção 2.2.4.4, os modelos roBERTa já são pré-treinados com uma grande quantidade de dados, mas é importante fazer o processo de *fine-tuning* para especializá-lo na tarefa de PLN desejada, que neste escopo é a classificação de sentimentos de notícias financeiras.

Para tal finalidade, recorreu-se a uma base de notícias previamente rotulada quantos aos sentimentos: positivo, negativo e neutro. O site utilizado para o *download* da base foi o Kaggle[2]. Além disso, o conteúdo da base apresenta notícias do seguimento financeiro traduzidas para o português, visto que seu texto original está em inglês. As notícias abordam assuntos financeiros de diversas empresas de seguimentos diferentes.

Ademais, detectou-se como um desafio o elevado desbalanceamento entre as três classes de sentimentos presentes nessa base. Seu conjunto totaliza 4.845 notícias, das quais 59% têm sentimento neutro, 28% positivo e 12% negativo.

Inicialmente, o modelo foi treinado sem nenhum ajuste quanto ao desequilíbrio da base de dados. Isso resultou em um modelo com uma acurácia muito baixa, cerca de 60%.

[1] Link do modelo: https://huggingface.co/FacebookAI/xlm-roberta-base
[2] Plataforma online onde usuários podem baixar *Datasets* e treinar modelos de ML

Para mitigar essa problemática, foi adotada uma estratégia que aplicou pesos para cada classe de sentimento de forma a compensar o desalinhamento inicial.

O peso da classe de sentimento $j$ foi calculado pela divisão do número total de amostras do *Dataset* pela multiplicação do número de classes diferentes (neste caso são três) com o número total de amostras da classe $j$. A fórmula 4.2 exemplifica esse cálculo:

$$PESO_j = \frac{n_{\text{total}}}{n_{\text{classes}} \times n_{\text{total},j}} \tag{4.2}$$

Na implementação, a operação acima foi executada com o auxílio da função *compute_class_weight* da biblioteca *Scikit-learn*, facilitando o processo. Abaixo está a representação das classes com seus respectivos pesos já calculados:

- Positivo: 1.19
- Negativo: 2.67
- Neutro: 0.56

Outras estratégias também foram testadas para mitigar o desbalanceamento da base de treinamento. Dentre elas, pode-se citar o uso de *Data Augmentation* para aumentar a diversidade dos dados de treinamento para as classes com um menor número de amostras (positivo e negativo).

Nesse contexto, foi implementado técnicas de *Data augmentation* por substituição de sinônimos e paráfrase do texto original. Entretanto, os resultados de treinamento obtidos com a utilização destas técnicas não teve ganhos significativos. Portanto, manteve-se somente a técnica de pesos das classes previamente elucidada.

Para iniciar o processo de *fine-tuning*, importou-se todas as bibliotecas necessárias e a base de dados de treinamento foi carregada como um *Dataframe* pandas. Depois, foi necessário mapear as classes de sentimentos para valores numéricos, visto que o modelo não aceita rótulos com tipo *string* (bloco 3). Após essa modificação, o *Dataframe* foi convertido para a estrutura de dados *Dataset*. Essa é a estrutura requisitada pelas bibliotecas do *Hugging Face* para fazer o treinamento dos modelos.

**Bloco 3** – *Configurações de Mapemento e conversão*

```
# Mapeando sentimentos para valores inteiros
labels_map = {"negativo": 0, "neutro": 1, "positivo": 2}
dataframe["label"] = dataframe["sentimento"].map(labels_map)

# Convertendo Dataframe para dataset do Hugging Face
```

```
dataset = Dataset.from_pandas(dataframe[["texto", "label"]])
```

Na sequência, o *Dataset* foi dividido em dois conjuntos de dados: 80% deles destinados ao treinamento e os outros 20% para testes. Essa divisão foi estratificada com o intuito de manter a proporcionalidade dos dados de cada classe em ambos os conjuntos. Essa técnica é conhecida como *hold-out* estratificado.

**Bloco 4** – *Divisão da base de dados*

```
dataset = dataset.train_test_split(test_size=0.2, seed=42,
    stratify_by_column="label")
```

A próxima etapa foi realizar as configurações de treinamento do modelo. Isso incluiu inicialmente o carregamento do modelo e do *Tokenizer*, que foi responsável em transformar o conteúdo das notícias em números para ser processado pelo modelo.

A classe responsável por carregar o *Tokeziner* foi a *AutoTokenizer* e ela possui o método **from_pretrained** que recebeu como parâmetro o nome do modelo e retornou o seu respectivo *Tokenizer*. A mesma função existe na classe *AutoModelForSequenceClassification*, que ficou encarregada de carregar o modelo propriamente dito. A diferença está na passagem de parâmetros adicionais à função, tais como o número de rótulos (*num_labels*) e o tipo do problema (*problem_type*), possibilitando o carregamento do modelo especificado (bloco 5).

**Bloco 5** – *Carregamento inicial do modelo*

```
# Configurando o modelo
model_name = "xlm-roberta-base"

# Carregar tokenizer
tokenizer = AutoTokenizer.from_pretrained(model_name)

# Carregar o modelo
model = AutoModelForSequenceClassification.from_pretrained(
    model_name,
    num_labels=3,
    problem_type="single_label_classification"
)
```

Na sequência, o *Dataset* de entrada foi processado por uma função de *tokenização* que converteu o texto bruto em conjuntos de no máximo 512 *tokens* (chamados de *batchs*).

O bloco 6 mostra que a função **tokenize_function** converteu o texto presente na coluna "texto" em números (identificadores e máscaras de atenção) e também garantiu

que os *batchs* não excedessem seu máximo valor de 512 *tokens*, produzindo um *Dataset* pronto para o treinamento do modelo.

**Bloco 6** – *Tokenizando os dados de entrada*

```
def tokenize_function(examples):
    return tokenizer(examples["texto"], truncation=True, padding=
       False, max_length=512, return_attention_mask=True
    )

tokenized_data = dataset.map(tokenize_function, batched=True,
   remove_columns=["texto"])
```

Foram exploradas as seguinte métricas para avaliar a performance do processo de *fine-tuning*: acurácia, precisão, revocação (*recall* em inglês) e a medida F1 *score*. Elas foram carregadas com o uso da biblioteca *evaluate* e computadas na função **compute_metrics** (bloco 7).

**Bloco 7** – *Exemplo de implementação da métrica de acurácia*

```
# Carregando a metrica com "evaluate"
accuracy = evaluate.load("accuracy")

def compute_metrics(eval_pred):
    # Computando metrica
    acc = accuracy.compute(predictions=predictions, references=
       labels)
    return {"accuracy": acc["accuracy"]}
```

A etapa final que precedeu a aplicação do *fine-tuning* consistiu na definição dos hiperparâmetros de treinamento utilizando a classe *TrainingArguments*. Esta classe serviu como a configuração central para o treinamento, sendo que foram definidos os seguintes parâmetros:

- Número de épocas (*num_train_epochs*) = 5
- Taxa de Aprendizado (*learning_rate*) = $2 \times 10^{-5}$
- Tamanho do *batch* (*per_device_train_batch_size*) = 16
- Carregar melhor modelo ao final (load_best_model_at_end) = *True*
- Estratégias de salvamento e avaliação (*save_strategy*, *eval_strategy*) = "*steps*"

Outros parâmetros adicionais também foram incluídos na implementação, contudo é correto considerar os parâmetros previamente elucidados como principais. Também é importante destacar que o valor da taxa de aprendizado teve alta influência sobre o desempenho do treinamento. Foram testados diferentes valores para esse parâmetro, entretanto o melhor resultado foi obtido com a instância de $2 \times 10^{-5}$.

Além disso, os parâmetros *save_steps* e *eval_steps* determinaram a frequência, em número de passos de treinamento, com que o modelo foi salvo e avaliado, respectivamente. Cada passo (*step*) correspondeu a uma atualização dos pesos do modelo após o processamento de um *batch* de dados. Neste caso, foi atribuído o valor de 100 a ambos os parâmetros, indicando que o modelo foi salvo e avaliado a cada 100 atualizações durante o treinamento.

A classe *WeightedTrainer* foi a responsável pelo treinamento propriamente dito. Ela escondeu a complexidade do processo com o auxílio de uma interface simples, ajudando no fluxo de treinamento. Dentre suas principais funcionalidades, pode-se destacar a gerência dos *Datasets* de treino e validação, cálculo das métricas e o salvamento de *checkpoints* durante o treinamento (bloco 8).

**Bloco 8** – *Treinando o modelo*

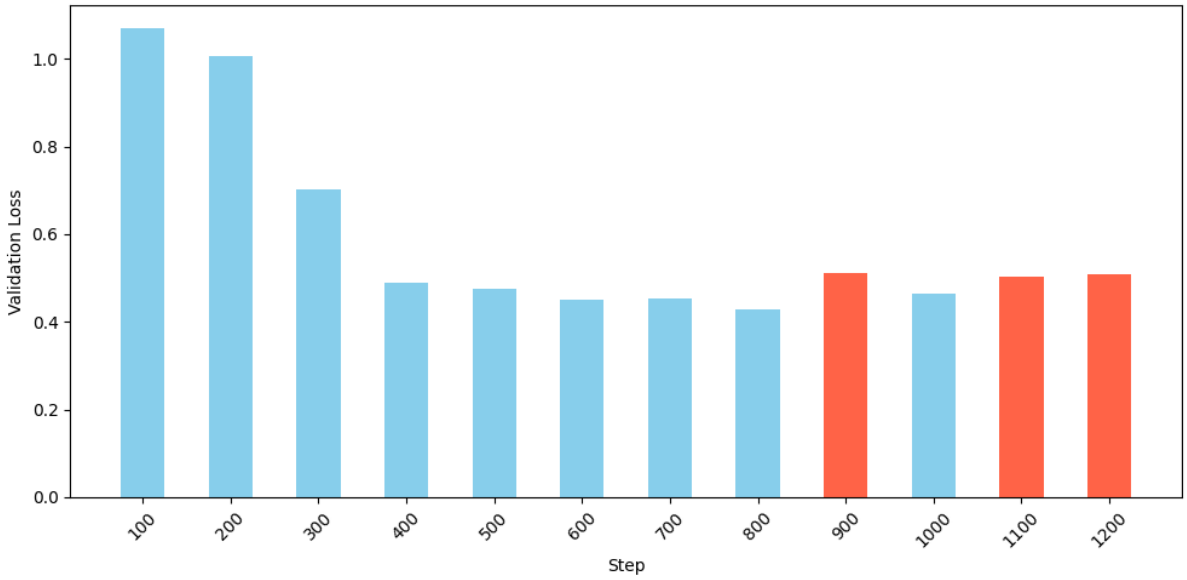
```
trainer = WeightedTrainer(
    model=model,
    args=training_args,
    train_dataset=tokenized_data["train"],
    eval_dataset=tokenized_data["validation"],
    tokenizer=tokenizer,
    data_collator=data_collator,
    compute_metrics=compute_metrics,
)

trainer.train()
```

A tabela 3 mostra o resultado das métricas de treinamento do modelo em diferentes *steps*. É observável que o valor de *Training Loss* caiu de 1,1072 no *step* 100 até valores muitos baixos, como 0,1073 no *step* 1100, indicando que o modelo apresentou um bom aprendizado com os dados de treino.

A figura 12 mostra que os valores da coluna de *Validation Loss* também tiveram uma queda constante (de 1,0691 no *step* 100 até 0,4287 no *step* 800). Entretanto, do *step* 900 em diante o seu valor mostrou muita oscilação. Isso sugere que o modelo pode ter começado a apresentar *overfitting*, visto que, nos *steps* 900, 1100 e 1200, seu valor aumentou (representado pela barra vermelha na figura 12), em vez de continuar decrescendo

gradualmente[3].

**Figura 12** – *Resultado de desempenho da métrica de Validation Loss*




**Fonte:** Elaborado pelo autor.

Como o argumento *metric_for_best_model="eval_loss"* foi definido na classe *TrainingArguments*, o estado do modelo que foi salvo ao final do treinamento é o que apresentou o melhor valor para a métrica de *Validation Loss*. Logo, o estado do modelo que foi utilizado neste estudo é o que está presente no *step* 800 da tabela 3. Nesse ínterim, o modelo utilizado possui acurácia de $85,86\%$ e um *F1-score* de $86\%$.

As métricas utilizadas para mensurar o desempenho do modelo foram *accuracy, precision, recall e F1 score*. A avaliação conjunta dessas métricas é fundamental no contexto da classificação de notícias financeiras, pois não basta verificar apenas o desempenho global do modelo. É necessário compreender como ele se comporta em cada classe, identificando a proporção de acertos, os tipos de erros cometidos e o nível de confiabilidade de suas previsões.

A métrica de acurácia (*acurracy*) fornece uma visão geral do quanto o modelo acerta como um todo, enquanto a precisão (*precision*) mostra o quão confiáveis são as previsões positivas, evitando falsos alarmes que prejudicam muito uma análise financeira. Já a métrica de revocação (*recall*) complementa a análise de desempenho ao indicar a capacidade do modelo de encontrar todos os casos relevantes, reduzindo a chance de escapar exemplos importantes.

Por última, a métrica de *F1-score* equilibra precisão e revocação em uma única medida, permitindo avaliar o modelo de forma robusta mesmo quando há desequilíbrio

[3] Overfitting é quando um modelo aprende tão bem os dados de treinamento que perde a capacidade de generalizar para novos dados.

entre classes, como é o caso desse treinamento. Assim, essas métricas juntas garantem uma análise mais sólida e informativa sobre a qualidade real do modelo de classificação.

Além disso, os resultados das métricas de desempenho *accuracy, precision, recall e F1 score* cresceram de forma consistente até por volta do *step* 800-1000 e atingiram valores estáveis dentro do intervalo $[0,86; 0,87]$ (tabela 3).

**Tabela 3** – *Resultados do treinamento do modelo em diferentes steps.*

| Step | Training Loss | Validation Loss | Accuracy | Precision | Recall | F1 score |
|---|---|---|---|---|---|---|
| 100 | 1,107200 | 1,069120 | 0,369453 | 0,507468 | 0,369453 | 0,361782 |
| 200 | 1,056600 | 1,005706 | 0,598555 | 0,552495 | 0,598555 | 0,552873 |
| 300 | 0,929000 | 0,702822 | 0,691434 | 0,708447 | 0,691434 | 0,692313 |
| 400 | 0,592200 | 0,488453 | 0,818369 | 0,826928 | 0,818369 | 0,816977 |
| 500 | 0,535000 | 0,475480 | 0,816305 | 0,836299 | 0,816305 | 0,820311 |
| 600 | 0,435300 | 0,449947 | 0,843137 | 0,844162 | 0,843137 | 0,843450 |
| 700 | 0,388900 | 0,452026 | 0,849329 | 0,860444 | 0,849329 | 0,851550 |
| 800 | 0,280000 | 0,428716 | 0,858617 | 0,865091 | 0,858617 | 0,860084 |
| 900 | 0,282500 | 0,510445 | 0,842105 | 0,853246 | 0,842105 | 0,844731 |
| 1000 | 0,193200 | 0,463799 | 0,856553 | 0,864815 | 0,856553 | 0,858384 |
| 1100 | 0,107300 | 0,502875 | 0,866873 | 0,871393 | 0,866873 | 0,867919 |
| 1200 | 0,216600 | 0,507510 | 0,860681 | 0,865658 | 0,860681 | 0,861835 |

Após a etapa de *fine-tuning*, o modelo ficou pronto para ser utilizado. A criação do classificador foi realizada de forma bastante intuitiva por meio da função *pipeline* da biblioteca *Hugging Face*, exigindo apenas o fornecimento do modelo e do *Tokenizer* como parâmetros.

A entrada do classificador consiste nos textos das notícias coletadas, e sua saída corresponde ao sentimento associado a cada notícia. Durante esta etapa do desenvolvimento, observou-se que o desempenho do classificador foi superior quando utilizado o conteúdo original das notícias, sem pré-processamento.

Essa ocorrência decorre do fato de que o pré-processamento eliminou palavras relevantes para o contexto, comprometendo a compreensão semântica do modelo. E uma vez que o modelo XLM-RoBERTa depende fortemente do contexto para realizar inferências precisas, preservar o conteúdo original do texto de entrada contribuiu significativamente para a acurácia de suas classificações.

O bloco de código 9 mostra como o classificador (*sentiment_classifier*) aplicou o modelo de classificação para cada notícia presente na coluna *conteúdo* do *Dataframe* **df**. O classificador extraiu o rótulo do sentimento (*label*), como positivo, negativo ou neutro e o armazenou em uma nova coluna chamada de *xlm_sentimento*. Ademais, outra extração foi feita para a pontuação do sentimento (*score*) que ficou armazenada na coluna *xlm_score*.

**Bloco 9** – *Utilizando o modelo XLM-roBERTa para classificação de notícias*

```
df["xlm_sentimento"] = df["conteudo"].apply(lambda x:
    sentiment_classifier(str(x))[0]['label'])
df["xlm_score"] = df["conteudo"].apply(lambda x:
    sentiment_classifier(str(x))[0]['score'])
```

O novo arquivo, que reúne o rótulo do sentimento e sua respectiva pontuação gerada pelo modelo XLM-roBERTa, foi exportado como CSV para posterior análise de correlação.

## 4.5 Análise de Correlação

Para alcançar o objetivo deste trabalho, que é comparar dois modelos de AS, foi de fundamental importância empregar um método estatístico capaz de mensurar o grau de associação entre os sentimentos gerados pelos modelos e os preços dos ativos PETR3 e PETR4 no mercado acionário. Para isso, utilizou-se o coeficiente de correlação de Pearson:

$$r = \frac{\Sigma_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\Sigma_{i=1}^{n}(x_i - \bar{x})^2(y_i - \bar{y})^2}} \tag{4.3}$$

Nesta pesquisa, a correlação foi feita por meio do uso da função *pearsonr*, disponibilizada pela biblioteca *scipy.stats*. Esse método calcula o índice de correlação entre duas variáveis, correspondentes ao $x$ e $y$ da fórmula 4.3.

O índice de correlação retornado é um valor no intervalo de $[-1; 1]$, indicando a força e a direção da relação linear entre as variáveis: valores próximos de 1 representam uma correlação positiva forte, valores próximos de -1 indicam correlação negativa forte, e valores próximos de 0 sugerem ausência de correlação linear.

Além do coeficiente de correlação, a função também retorna o *valor-p*, que permite avaliar a significância estatística da correlação observada. Caso o *valor-p* retornado seja menor que $0,05$, significa que a correlação é estatisticamente significativa. Entretanto, caso seu valor seja maior que $0,05$, não é possível afirmar se a correlação é realmente significativa.

A correlação dos dados foi realizada separadamente para cada um dos 4 anos em que foram coletadas as notícias (tabela 2). Além disso, para cada ano, duas correlações foram executadas, uma para cada modelo de AS.

Uma das variáveis usadas na correlação foi o percentual dos retornos diários dos ativos da Petrobras. Com o uso da biblioteca *yfinance* foi possível fazer o *download* desses dados de forma muito simples.

A outra variável utilizada na correlação foi a média de sentimentos diários das notícias, na qual foi construída a partir do agrupamento das datas de publicação das notícias com as respectivas datas de pregão[4] dos ativos analisados (PETR3 e PETR4).

Durante esse agrupamento, as notícias que foram publicadas depois do último dia de pregão do ano de análise foram descartadas. Observa-se na tabela 4 que somente os anos de 2020 e 2023 tiveram pequenas reduções na quantidade total das notícias coletadas. Logo, o impacto gerado por esse agrupamento foi mínimo.

**Tabela 4** – *Quantidade total de notícias coletadas por ano antes e após o agrupamento para análise de correlação*

| **Ano de coleta** | **Valor inicial** | **Valor final** |
|---|---|---|
| 2020 | 720 | 709 |
| 2021 | 752 | 752 |
| 2023 | 690 | 647 |
| 2024 | 747 | 747 |

Além disso, o modo de operação do agrupamento converteu previamente os sentimentos extraídos das notícias em valores numéricos (positivo = 1, negativo = -1 e neutro = 0), o que possibilitou calcular a média diária desses sentimentos. Dessa forma, para cada dia de negociação na bolsa, foi obtido um valor representativo do sentimento médio associado às notícias divulgadas no respectivo dia.

Com a obtenção das duas variáveis necessárias para a correlação, calculou-se o valor de seu índice para intervalos trimestrais e mensais dentro de cada ano. Aliás, também foi considerado um parâmetro de defasagem dos dados (*lag*), com o intuito de verificar qual foi o impacto gerado pelo sentimento das notícias atuais sobre o retorno dos preços das ações no dia seguinte. O apêndice C mostra a função de correlação que foi implementada para este estudo.

[4] Data de pregão corresponde ao dia em que ocorre negociações oficiais de ativos na bolsa de valores

# 5 Resultados

## 5.1 Resultados de classificação

Nesta seção, são apresentados os resultados obtidos pelos dois modelos de AS estudados aplicados na tarefa de classificação de notícias financeiras e, também, os resultados da correlação de ambos com a cotação dos preços das ações PETR3 e PETR4 no mercado financeiro.

A comparação entre os sentimentos gerados por cada modelo permitiu avaliar sua precisão, consistência e possíveis divergências na classificação das notícias.

A tabela 5 apresenta a distribuição dos sentimentos (positivo, negativo e neutro) ao longo dos 4 anos de análise. É possível notar que para o modelo de dicionário L&M, grande parte das notícias foram classificadas como negativas. Isso pode ser explicado pelo próprio desbalanceamento do dicionário L&M, induzindo o modelo a um viés negativo.

Já o modelo XML-roBERTa apresentou uma distribuição mais balanceada entre sentimentos neutros e negativos, com predominância de neutros. Em todos os anos, o número de classificações neutras foi maior que dos outros sentimentos. Além disso, em todos os anos, as classificações positivas foram mais numerosas quando comparado com o modelo de dicionário L&M. Isso sugere um modelo menos polarizado e que foi capaz de reconhecer detalhes no texto que tornou sua classificação equilibrada.

**Tabela 5** – *Distribuição dos sentimentos ao longo dos anos para os modelos L&M e XLM-roBERTa.*

| **Sentimento** | **Modelo L&M** | | | | **Modelo XLM-roBERTa** | | | |
|---|---|---|---|---|---|---|---|---|
| | **2020** | **2021** | **2023** | **2024** | **2020** | **2021** | **2023** | **2024** |
| **Positivo** | 58 | 62 | 65 | 86 | 155 | 203 | 197 | 195 |
| **Negativo** | 399 | 413 | 312 | 330 | 209 | 165 | 145 | 203 |
| **Neutro** | 263 | 277 | 313 | 331 | 356 | 384 | 348 | 349 |

A tabela 6 mostra o nível de concordância entre as classificações de sentimentos de ambos os modelos. A coluna *Concordância total* mostra a porcentagem de notícias em que ambos os modelos atribuíram o mesmo sentimento a elas. É possível observar que o ano de 2020 apresentou maior concordância ($44,31\%$), ou seja, neste ano os modelos tiveram interpretações mais próximas, mesmo que ainda de forma moderada. Já o ano de 2021 teve menor concordância entre os modelos, apontando uma divergência maior na percepção de sentimentos entre eles.

A coluna *Concordância por sentimento* detalha a quantidade de notícias em que os

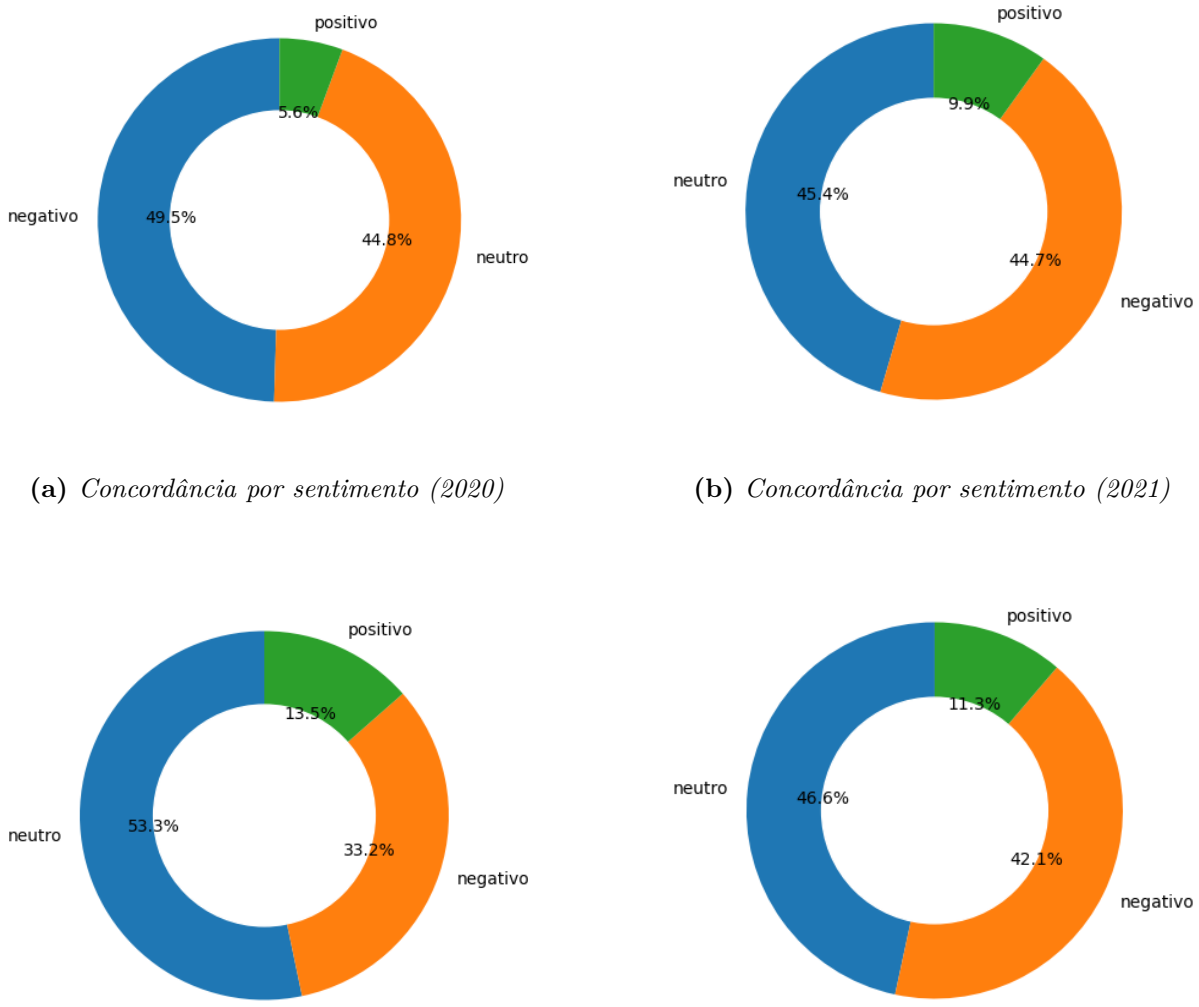
modelos coincidiram no mesmo tipo de sentimento. É notável que os modelos apresentam maior alinhamento nas classificações neutras e negativas, possivelmente devido ao seu maior volume. A baixa concordância em sentimentos positivos pode ser atribuída ao viés negativo do modelo de dicionário L&M, que influenciou sua concordância com o modelo XLM-roBERTa.

**Tabela 6** – *Distribuição anual do número de notícias coincidentes entre os modelos por tipo de sentimento*

| **Ano** | **Concordância total** | **Concordância por sentimento** | | |
|---|---|---|---|---|
| | | **Positivo** | **Negativo** | **Neutro** |
| 2020 | 44,31% | 18 | 158 | 143 |
| 2021 | 34,84% | 26 | 117 | 119 |
| 2023 | 37,57% | 35 | 86 | 138 |
| 2024 | 41,63% | 35 | 131 | 145 |

A figura 13 mostra os gráficos obtidos no estudo da concordância entre os modelos. Observa-se que, apenas em 2020 (figura 13a), o sentimento negativo superou o neutro, registrando a maior taxa de concordância. Ademais, no ano de 2023 (figura 13c), o sentimento positivo atingiu seu pico em relação aos demais anos ($13,5\%$), embora sua proporção ainda seja inferior à dos sentimentos neutro e negativo.

**Figura 13** – *Gráficos da porcentagem de concordância dos modelos por sentimento*




**(a)** *Concordância por sentimento (2020)*

**(b)** *Concordância por sentimento (2021)*

**(c)** *Concordância por sentimento (2023)*

**(d)** *Concordância por sentimento (2024)*

Dessa maneira, é evidenciado uma concordância moderada entre os modelos, visto que eles nem sempre interpretaram o sentimento das notícias da mesma forma, refletindo suas diferenças arquiteturais e, também, de captar sensibilidades linguísticas.

## 5.2 Resultados de correlação

Agora será analisado os resultados da correlação dos dois modelos com o valor de mercado das ações PETR3 e PETR4. A correlação foi analisada de forma trimestral e mensal para cada ano do período de estudo.

A análise trimestral oferece uma visão mais ampla dos dados, permitindo identificar padrões de correlação que se manifestam em períodos específicos do ano. Essa abordagem

foi útil para captar movimentos sazonais ou tendências que se tornam menos evidentes em análises mais granulares.

Por outro lado, a análise mensal proporcionou uma perspectiva mais detalhada e sensível às variações de curto prazo, revelando flutuações que podem estar associadas a eventos pontuais, decisões estratégicas da empresa ou mudanças no cenário macroeconômico.

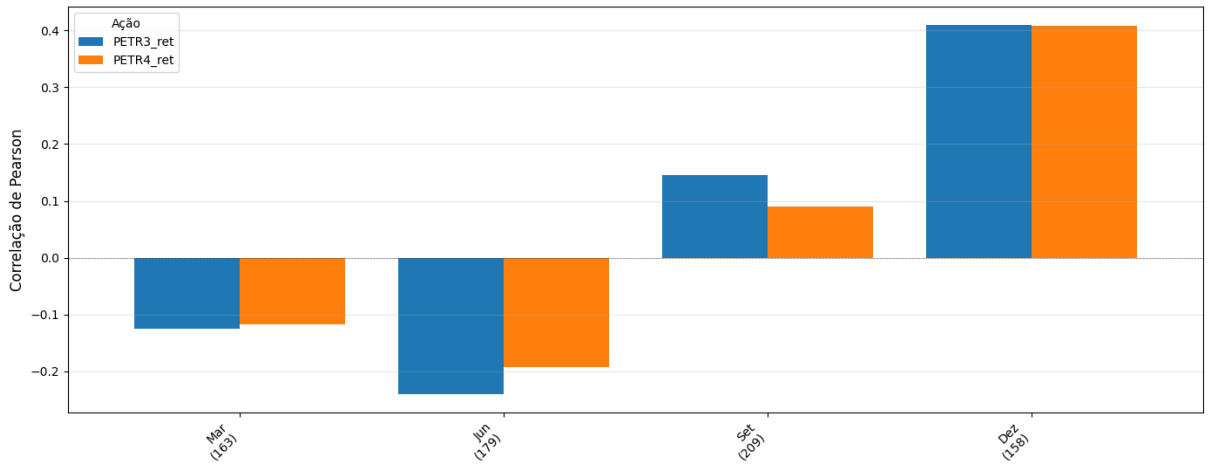
A função que calcula a correlação por período (apêndice C) agrupou todos os pares de valores de média de sentimento e retornos diários existentes no intervalo especificado (mês ou trimestre) e calculou diretamente o índice de correlação.

Ademais, é importante interpretar que uma correlação positiva indica que, à medida que o sentimento das notícias tornam-se mais positivos, o retorno da ação também tente a aumentar no mercado financeiro. Esse resultado sugere que o modelo foi eficaz ao prever corretamente a direção do movimento do ativo.

Em contrapartida, uma correlação negativa significa que, mesmo com o aumento do sentimento positivo nas notícias, o retorno da ação tende a diminuir, o que é um resultado ruim para o modelo. Neste ínterim, os resultados gráficos obtidos foram analisados levando em consideração essas propriedades.

A figura 14 mostra que, para o ano de 2020, o modelo de L&M não teve resultados consideráveis de correlação nos 3 primeiros trimestres do ano. Entretanto, melhorou seu desempenho no último trimestre, chegando a um valor de índice de correlação de 0,4 para ambos os ativos da Petrobras.

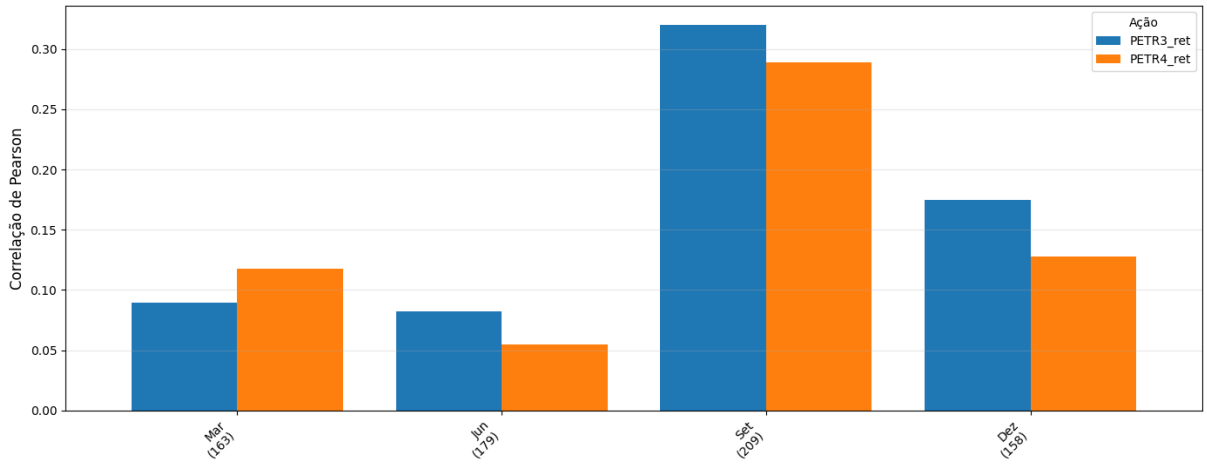
**Figura 14** – *Correlação trimestral do modelo de dicionário L&M em 2020*




**Fonte:** Elaborado pelo autor.

Já o modelo XLM-roBERTa (figura 15) performou melhor no 3 trimestre com um pico no valor do índice de correlação de 0,3. Além disso, ao longo de todo o ano de 2020, obteve mais acertos que o modelo de dicionário L&M, visto que em seu gráfico é possível observar a ausência de correlações negativas.

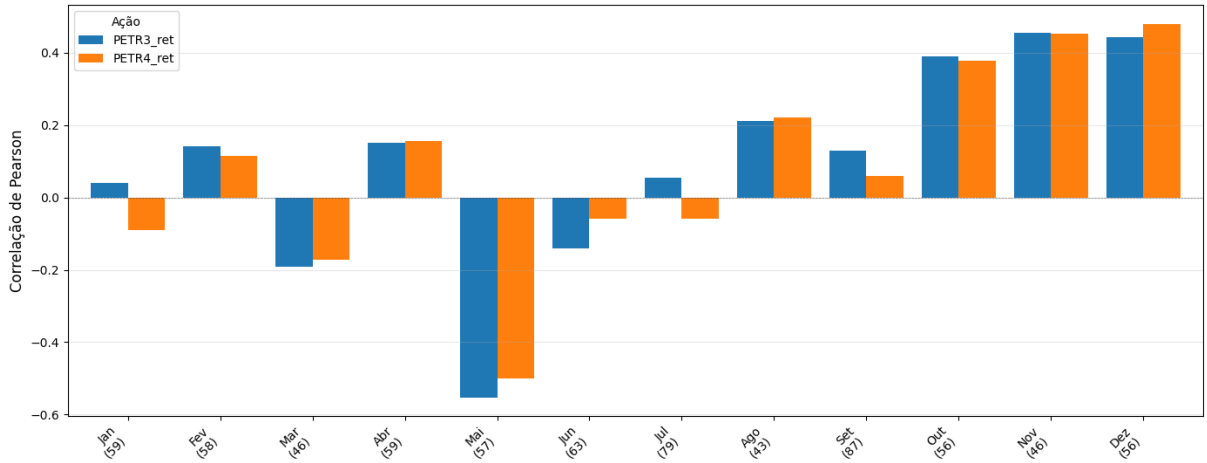
**Figura 15** – *Correlação trimestral do modelo XLM-roBERTa em 2020*




**Fonte:** Elaborado pelo autor.

No gráfico de correlação mensal do modelo de dicionário L&M (figura 16), é observável que no primeiro semestre do ano (de janeiro a junho) as correlações foram baixas e instáveis, variando entre positivas e negativas. O mês de maio apresentou uma forte correlação negativa, indicando que neste mês a classificação do modelo foi inversamente proporcional ao movimento dos preços das ações PETR3 e PETR4. No segundo semestre, a correlação mantém-se positiva de maneira constante, tornando-se mais acentuada a partir de outubro, com índices próximos ou acima de 0,4.

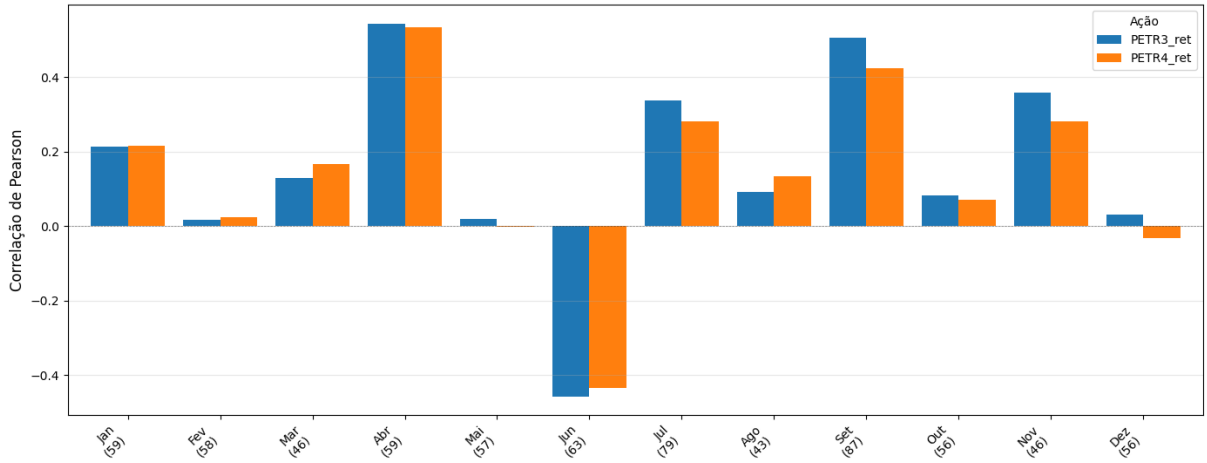
**Figura 16** – *Correlação mensal do modelo de dicionário L&M em 2020*




**Fonte:** Elaborado pelo autor.

O gráfico de correlação mensal do modelo XLM-roBERTa também apresentou uma correlação baixa e instável no primeiro semestre do ano, exceto pelo mês de abriu, onde a correlação foi positiva com um valor acima de 0,4 (figura 17). No restante do ano, o modelo obteve resultados bons para os meses de julho, setembro e novembro. Já para os meses de agosto, outubro e dezembro, o valor do índice não foi significativo.

**Figura 17** – *Correlação mensal do modelo XLM-roBERTa em 2020*



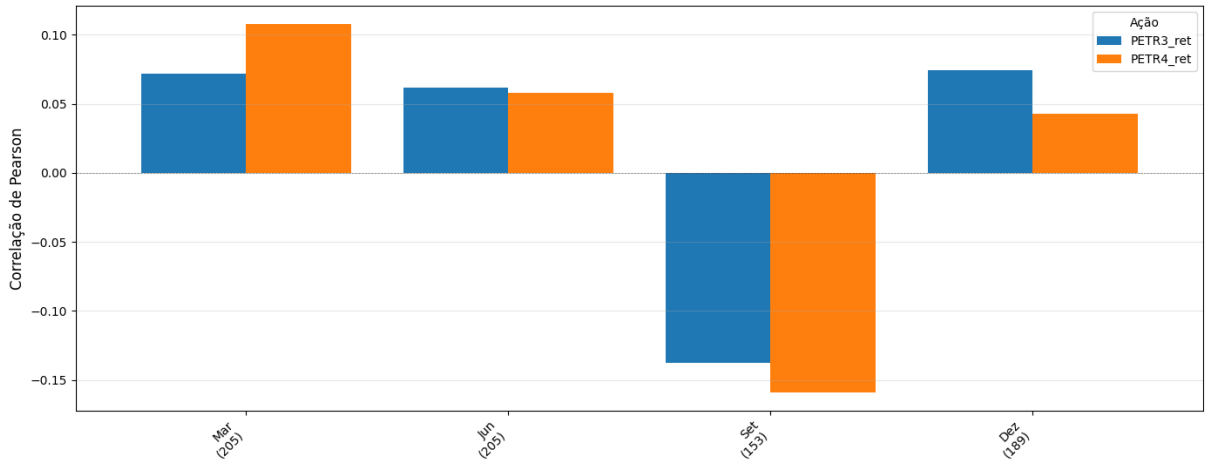

**Fonte:** Elaborado pelo autor.

A instabilidade das correlações observada no primeiro semestre do ano, em ambos os modelos, pode ser explicada pelo início da pandemia de Covid-19 no Brasil, que gerou forte turbulência no mercado financeiro nacional. Nesse contexto, o surgimento de notícias positivas não foi suficiente para reverter o movimento descendente das ações da Petrobras, influenciado pela crise macroeconômica do país.

Os preços dos ativos da Petrobras no ano de 2021 sofreram grandes oscilações. Os principais motivos foram a troca do CEO da estatal que impactou negativamente o valor de seus ativos no mês de fevereiro e, também, o aumento do preço do petróleo no mercado internacional no mês de junho. Essa fato fez a Petrobras ter o maior lucro já registrado na história da empresa até o momento e, com isso, sua ações tiveram um grande ganho no mercado.

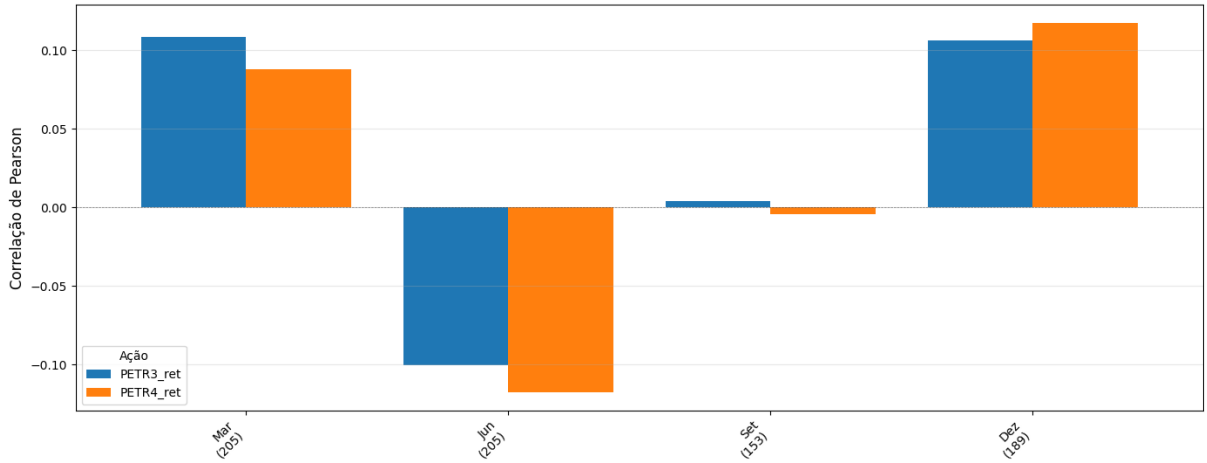
As figuras 18 e 19 apresentam as correlações trimestrais de ambos os modelos ao longo de 2021. Os resultados foram inferiores aos observados no ano anterior, com o pico de correlação positiva em ambos os gráficos não ultrapassando 0,1. Esse desempenho pode indicar um aumento nos erros de classificação dos modelos ou, ainda, que a elevada volatilidade dos ativos da Petrobras naquele período reduziu a capacidade das notícias de influenciar o mercado, comprometendo, assim, a previsibilidade dos modelos.

**Figura 18** – *Correlação trimestral do modelo de dicionário L&M em 2021*




**Fonte:** Elaborado pelo autor.

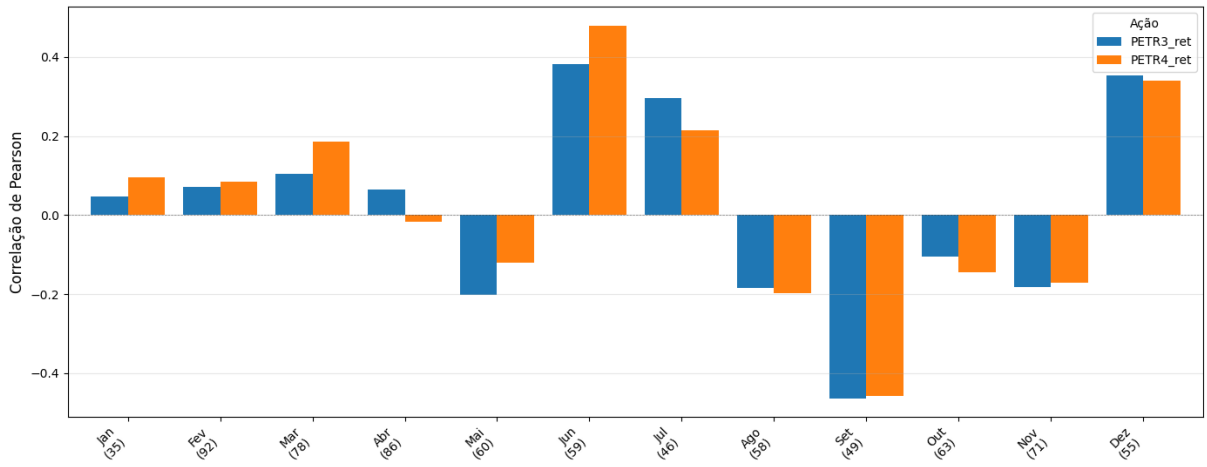
**Figura 19** – *Correlação trimestral do modelo XLM-roBERTa em 2021*




**Fonte:** Elaborado pelo autor.

Analisando os gráficos mensais de correlação do ano de 2021 foi possível identificar que o modelo de dicionário L&M teve resultados melhores que o modelo XLM-roBERTa.

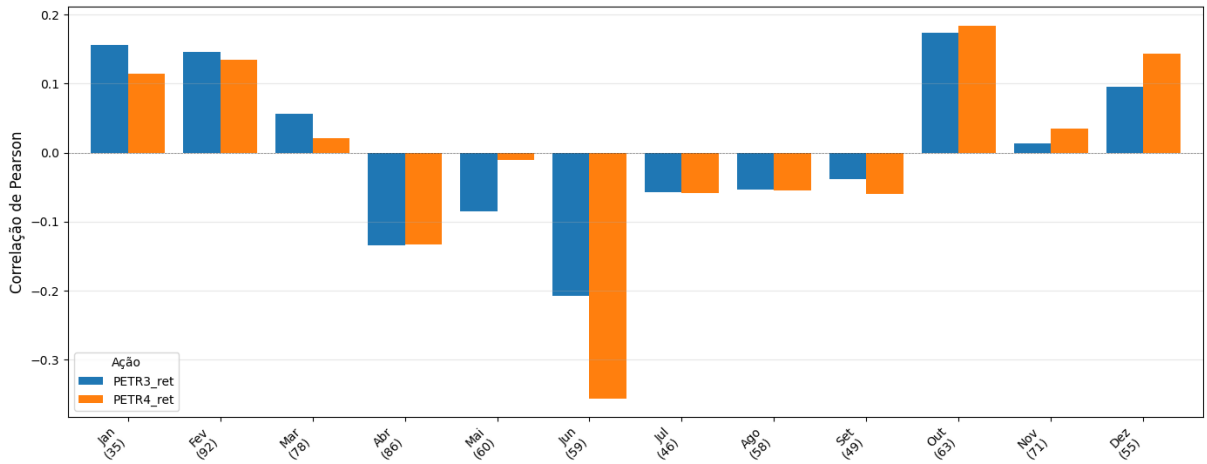
**Figura 20** – *Correlação mensal do modelo de dicionário L&M em 2021*




**Fonte:** Elaborado pelo autor.

Desse modo, ainda em 2021, no modelo XLM-roBERTa (figura 21), as correlações positivas que foram observadas no primeiro e último trimestre do ano não ultrapassaram o valor de 0,2, sugerindo que o modelo XLM-roBERTa produziu menos acertos de classificação que o modelo de dicionário L&M e, desse modo, os impactos das notícias sobre a movimentação dos ativos da Petrobras não foram detectados de forma consistente.

**Figura 21** – *Correlação mensal do modelo XLM-roBERTa em 2021*




**Fonte:** Elaborado pelo autor.

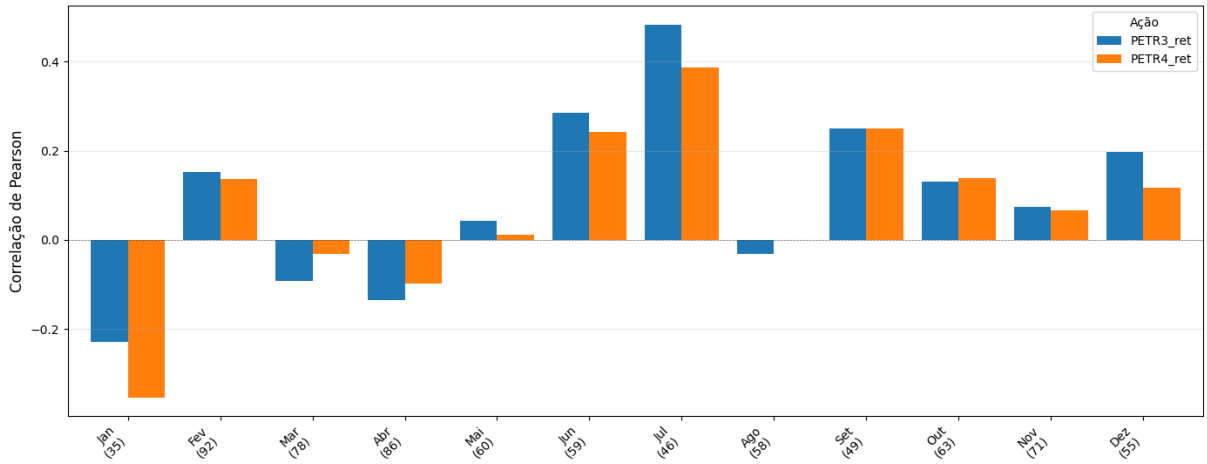
Como foi possível observar, ambos os modelos não apresentaram acertos relevantes no dia em que a notícia foi publicada. Entretanto, ao analisar os resultados da correlação com defasagem de um dia, ou seja, os impactos que os sentimentos atuais das notícias tiveram sobre os preços dos ativos no dia seguinte, é possível identificar pontos de acerto mais significativos.

Os gráficos das figuras 22 e 23 mostram as correlações com defasagem de ambos os modelos para o ano de 2021. Um ponto relevante a ser observado foi que, no primeiro trimestre do ano (de janeiro a março), os gráficos com defasagem apresentaram desempenho

inferior em relação aos modelos analisados sem defasagem, uma vez que as correlações inicialmente positivas tornaram-se negativas.

Ao analisar o gráfico do modelo de dicionário, com defasagem de um dia, em 2021 (figura 22), verificou-se que os resultados dos meses de junho e dezembro foram menores em comparação ao gráfico sem defasagem (figura 20). Entretanto, o modelo melhorou seus acertos em julho e, também, nos meses de setembro a novembro, quando as correlações tornaram-se todas positivas.

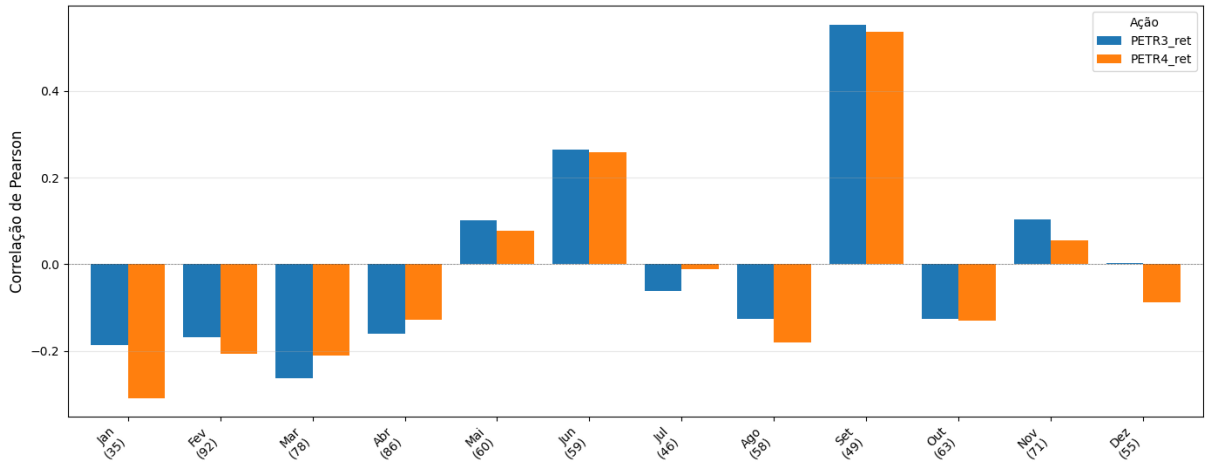
**Figura 22** – *Correlação mensal com defasagem do modelo de dicionário L&M em 2021*




**Fonte:** Elaborado pelo autor.

Já o modelo XLM-roBERTa, com defasagem de um dia (figura 23), apresentou várias correlações negativas, mas apontou notáveis melhoras nos meses de junho, com valores acima de 0,2 e em setembro, com valores acima de 0,4. Entretanto, o desempenho do índice de correlação caiu nos meses de outubro e dezembro.

**Figura 23** – *Correlação mensal com defasagem do modelo XLM-roBERTa em 2021*

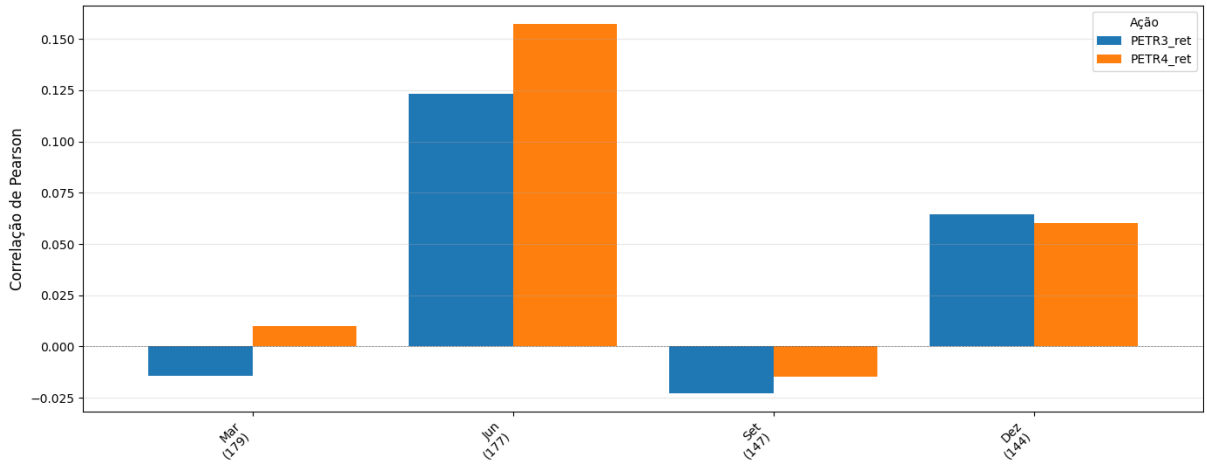



**Fonte:** Elaborado pelo autor.

Para o ano de 2023, a correlação trimestral do modelo de dicionário L&M (figura

24) teve um pico discreto no segundo trimestre e apesar do último trimestre também apresentar correlação positiva, o seu valor não foi expressivo (menor que 0,1).

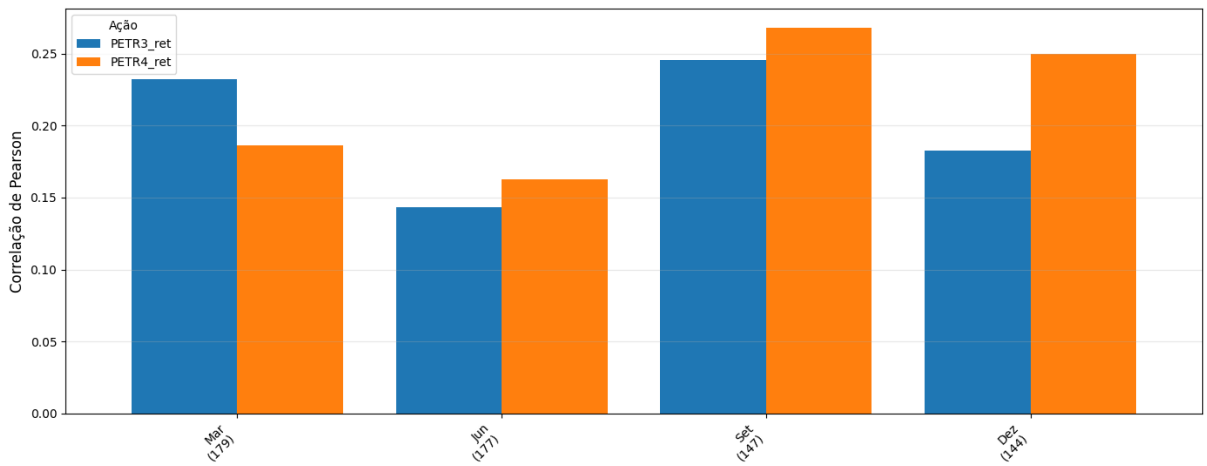
**Figura 24** – *Correlação trimestral do modelo de dicionário L&M em 2023*




**Fonte:** Elaborado pelo autor.

Já o modelo XLM-roBERTa (figura 25) mostrou consistência nas correlação ao longo de todo o ano de 2023, indicando que percepções mais otimistas nas notícias estiveram associadas a desempenhos positivos das ações. Além disso, os ativos PETR3 e PETR4 apresentaram comportamento semelhante em seus resultados, reforçando a coerência entre eles.

**Figura 25** – *Correlação trimestral do modelo XLM-roBERTa em 2023*



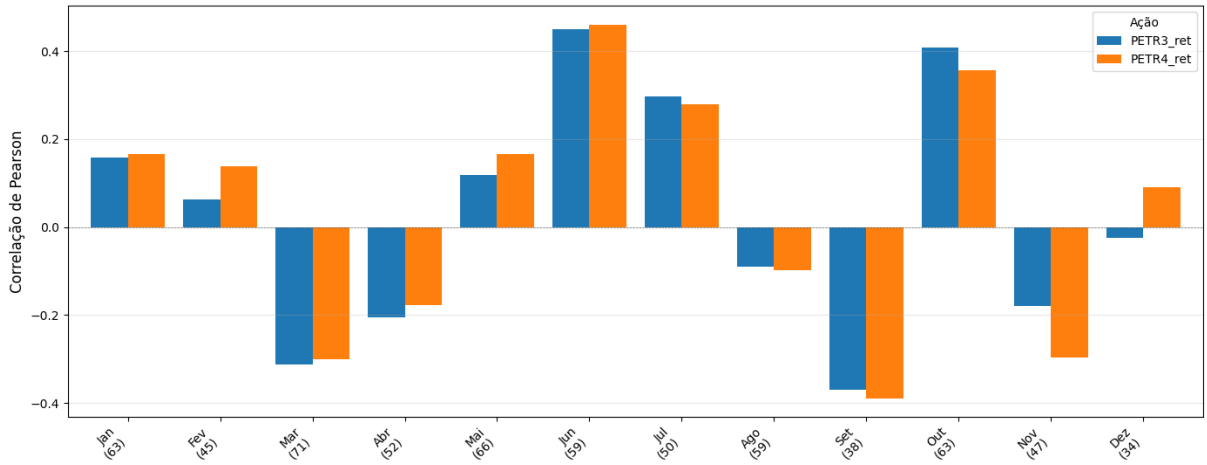

**Fonte:** Elaborado pelo autor.

Os resultados mensais dos modelos no ano de 2023 (figuras 26 e 27) foram mais semelhantes que em anos anteriores. Embora a porcentagem de concordância entre suas classificações ser de $37,57\%$ (tabela 6), os índices de correlação sinalizam que ambos os modelos conseguiram identificar os mesmos padrões de associação entre os sentimentos das notícias e a movimentação dos ativos da Petrobras.

Nos meses de junho, julho e outubro, os modelos capturaram adequadamente o tom das notícias de forma compatível com os movimentos de preço e por isso, foram os meses com maiores índices de correlação positiva.

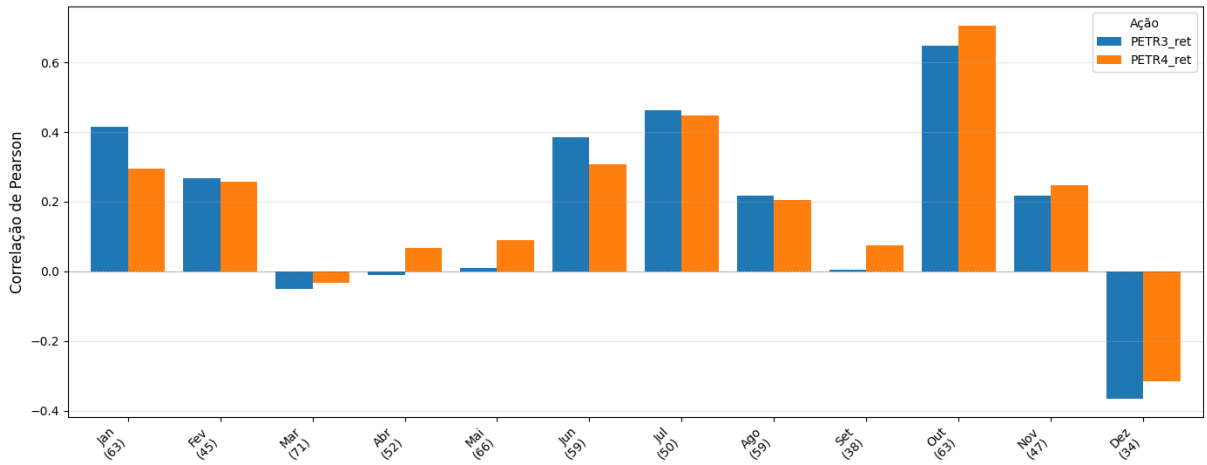
No caso do modelo de dicionário L&M (figura 26), houve uma maior oscilação entre as correlações positivas e negativas em comparação com o modelo XLM-roBERTa (figura 27), que se manteve mais estável ao longo do ano de 2023 e se destacou, especialmente em outubro, ao atingir um índice de correlação superior a 0,6 para ambos os ativos da Petrobras.

**Figura 26** – *Correlação mensal do modelo de dicionário L&M em 2023*




**Fonte:** Elaborado pelo autor.

**Figura 27** – *Correlação mensal do modelo XLM-roBERTa em 2023*



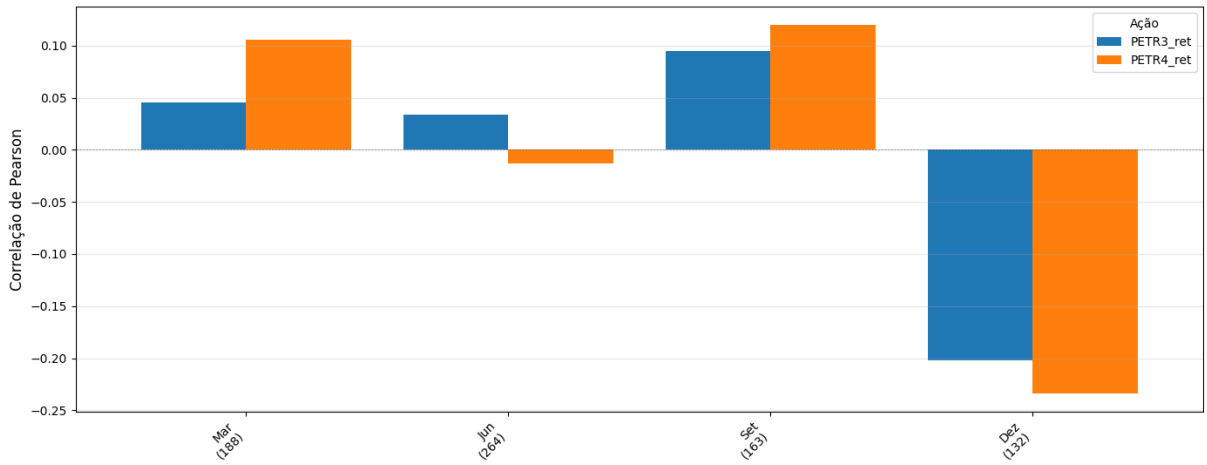

**Fonte:** Elaborado pelo autor.

Considerando os quatro anos avaliados nesta pesquisa (2020, 2021, 2023 e 2024), o ano de 2024 apresentou a maior estabilidade na variação dos preços dos ativos PETR3 e PETR4.

Desse modo, ao analisar os gráficos das correlações trimestrais para o ano de 2024, evidenciou-se a superioridade do modelo XLM-roBERTa (figura 29) ao modelo de dicionário L&M (figura 28).

No modelo de dicionário L&M (figura 28), as correlações trimestrais positivas tiveram poucos resultados, ficando um pouco acima do valor de 0,1 para o ativo PETR4 no primeiro e terceiro trimestre do ano.

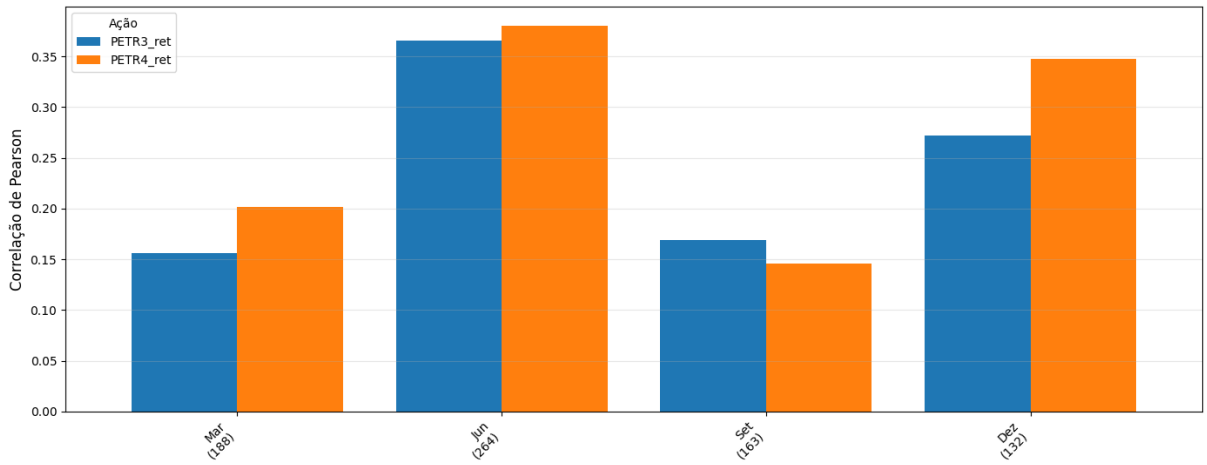
**Figura 28** – *Correlação trimestral do modelo de dicionário L&M em 2024*




**Fonte:** Elaborado pelo autor.

Por outro lado, o modelo XLM-roBERTa (figura 29) teve correlações trimestrais positivas ao longo de 2024, destacando-se no segundo trimestre com valores acima de 0,35 para ambos os ativos.

**Figura 29** – *Correlação trimestral do modelo XLM-roBERTa em 2024*



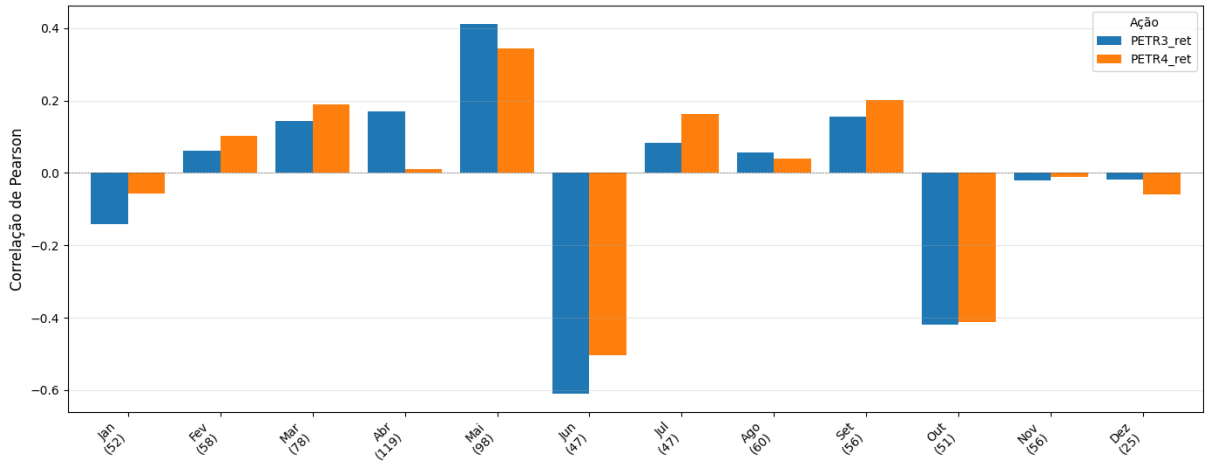

**Fonte:** Elaborado pelo autor.

Para o gráfico de correlação mensal do modelo de dicionário L&M (figura 30), foi possível notar instabilidade, variando entre correlações positivas e negativas ao longo de 2024.

Nos meses de fevereiro a maio, foi observado um forte aumento das correlações positivas, com pico em maio, quando as correlações ultrapassam 0,4 para o ativo PETR3. Esse período refletiu maior alinhamento entre o tom das notícias e o desempenho das ações. Entretanto, em junho, ocorre uma inversão brusca, com correlações negativas acentuadas (em torno de -0,5 e -0,6).

Os meses de julho a setembro voltaram a apresentar valores positivos moderados até outubro, quando ocorreu outra forte queda. Nos meses finais (novembro e dezembro), observou-se neutralidade pela falta de correlação linear.

**Figura 30** – *Correlação mensal do modelo de dicionário L&M em 2024*



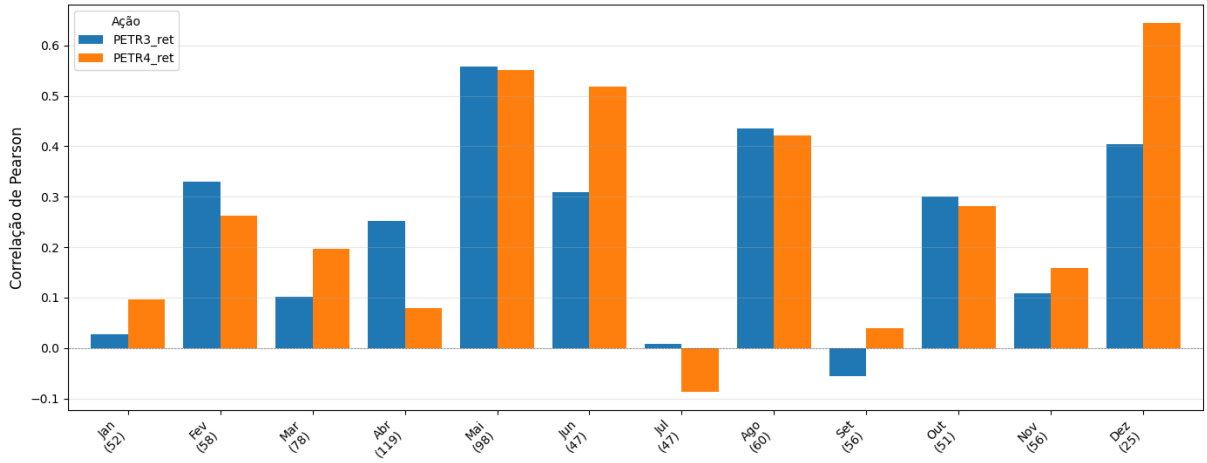

**Fonte:** Elaborado pelo autor.

De forma geral, para o modelo XLM-roBERta (figura 31) as correlações se mantiveram positivas na maior parte do ano de 2024, indicando que os sentimentos positivos das notícias tiveram sucesso em acompanhar os períodos de valorização das ações.

No primeiro trimestre do ano (de janeiro a junho), houve uma tendência de forte correlação positiva, com destaque para maio, quando ambas as ações alcançam o pico do período (em torno de 0,55). Os meses de julho e setembro apresentaram neutralidade, indicando momentos de desconexão entre os sentimentos e o valor de retorno das ações.

A partir de outubro, as correlações voltam a subir, atingindo novos máximos em dezembro, com valores acima de 0,6 para PETR4 e 0,4 para PETR3.

**Figura 31** – *Correlação mensal do modelo XLM-roBERTa em 2024*




**Fonte:** Elaborado pelo autor.

Em comparação com o modelo de dicionário L&M, o modelo XLM-RoBERTa exibiu resultados mais consistentes e predominantemente positivos, sugerindo que este modelo captou melhor a relação entre o tom das notícias e o comportamento do mercado.

As correlações com defasagem para os anos de 2020, 2023 e 2024 não mostraram resultados melhores que as suas respectivas correlações sem defasagem. Isso demonstrou que o impacto das notícias na movimentação das ações da Petrobras no dia de sua publicação foi superior ao do dia seguinte.

Ademais, por esta pesquisa ter sido desenvolvida no âmbito de uma universidade pública, os sites utilizados para a coleta de notícias não foram os mesmos amplamente utilizados por analistas da área financeira, uma vez que esses exigem assinatura para acesso ao conteúdo. Nesse contexto, optou-se por sites nacionais gratuitos, que frequentemente republicam informações provenientes de fontes mais influentes. Em razão disso, as correlações obtidas podem não ter alcançado resultados mais expressivos.

No panorama geral dos anos analisados, o modelo XLM-roBERTa apresentou desempenho superior ao modelo de dicionário L&M, especialmente por ter obtido uma quantidade maior de correlações positivas estatisticamente significativas.

Essa vantagem foi observada em todos os anos, com valores de correlação mais elevados e consistentes, exceto para o ano de 2021. A significância estatística foi avaliada com base no *valor-p* retornado pela função de correlação *pearsonr*, evidenciando a robustez do modelo XLM-roBERTa frente ao modelo de dicionário L&M.

# 6 Conclusão

Os experimentos realizados nesta pesquisa evidenciam a relevância do desenvolvimento de técnicas de PLN voltadas à construção de métodos para prever a movimentação do mercado financeiro com base em dados de notícias financeiras, com o objetivo de auxiliar profissionais da área na tomada de decisões de investimento mais embasadas.

Foram implementados dois modelos de AS com o objetivo de comparar seus desempenhos na classificação de textos de notícias financeiras relacionadas à empresa Petrobras e, adicionalmente, correlacionar seus resultados com a movimentação dos ativos PETR3 e PETR4. Essa abordagem investigou qual dos modelos apresentou maior capacidade preditiva em relação às variações de preço dos ativos ao longo do período analisado.

Em termos de capacidade de classificação, o modelo XLM-roBERTa se mostrou mais robusto que o modelo de dicionário L&M em captar sutilezas nos textos coletados e, portanto, apresentou uma distribuição equilibrada de sentimentos entre as notícias, com destaque para o sentimento neutro (tabela 6).

Por outro lado, o modelo de dicionário L&M exibiu um viés negativo em suas classificações devido ao desbalanceamento do próprio dicionário L&M. Desse modo, o modelo teve mais dificuldade na classificação de notícias com sentimento positivo, resultando em perdas de desempenho na previsibilidade do modelo.

Quanto as correlações, os resultados do período da pandemia (2020 e 2021) revelaram maior instabilidade dos modelos em prever corretamente a movimentação dos ativos. Diante desse cenário, destacou-se, em 2021, o modelo de dicionário L&M que teve melhor previsibilidade do que o modelo XLM-roBERTa.

No período pós pandemia (2023 e 2024), com maior estabilidade no cenário macroeconômico nacional, destacou-se a superioridade do modelo XLM-roBERTa, que apresentou bons índices de correlação positiva nesse período.

No geral, ficou evidente a superioridade do modelo XLM-roBERTa que, por se tratar de um modelo de aprendizagem de máquina com arquitetura *transformer*, sua eficiência na tarefa de classificação de notícias financeiras foi eminente e, consequentemente, exibiu melhores resultados de índice de correlação.

Apesar das limitações do modelo de dicionário L&M, ele apresentou a vantagem de ser mais ágil que o modelo XLM-roBERTa, uma vez que o seu desempenho está atrelado apenas ao pré-processamento dos textos de entrada e ao dicionário de palavras utilizado. Por outro lado, o modelo de XLM-roBERTa requer uma etapa prolongada de treinamento e ajuste de hiperparâmetros, além de demandar maior capacidade computacional.

Assim, conclui-se que o estudo atingiu seu objetivo ao implementar e comparar os modelos XLM-roBERTa e o modelo baseado em dicionário L&M, avaliando suas capacidades tanto na classificação de sentimentos quanto na associação com os movimentos das ações PETR3 e PETR4 no mercado financeiro.

Em estudos futuros, o modelo baseado em dicionário pode ser aprimorado com o desenvolvimento de um novo dicionário, com balanceamento entre as classes sentimentais e, também, adaptado para o contexto do mercado financeiro nacional. Dessa forma, o problema do viés negativo observado neste estudo seria mitigado e, com isso, as classificações mostrariam resultados mais equilibrados e consistentes.

Em relação ao modelo XLM-RoBERTa, o processo de *fine-tuning* pode ser realizado com uma base de dados mais ampla e balanceada, composta preferencialmente por notícias nacionais rotuladas, em vez de traduções para o português, como adotado nesta pesquisa. Essa alteração na base de treinamento tem potencial para promover melhorias significativas no desempenho do modelo

Além disso, recomenda-se ampliar a coleta de notícias financeiras para incluir portais de maior relevância e influência, pois essas fontes tendem a exercer maior impacto sobre os movimentos do mercado. Com isso, os modelos analisados podem alcançar maior previsibilidade em relação às variações dos ativos PETR3 e PETR4.

# Referências


AGRAWAL, M. et al. Predicting stock market trends using machine learning and sentiment analysis. In: **SoutheastCon 2025**. [S.l.: s.n.], 2025. p. 1001–1006.

CONNEAU, A. et al. Unsupervised cross-lingual representation learning at scale. **arXiv preprint arXiv:1911.02116**, 2020. Disponível em: <https://arxiv.org/abs/1911.02116>.

DEVLIN, J. et al. Bert: Pre-training of deep bidirectional transformers for language understanding. **arXiv preprint arXiv:1810.04805**, 2018. Disponível em: <https://arxiv.org/abs/1810.04805>.

FAMA, E. F. Efficient capital markets: A review of theory and empirical work. **The Journal of Finance**, Wiley, v. 25, n. 2, p. 383–417, 1970. Accessed: 2025-04-08. Disponível em: <https://www.jstor.org/stable/2325486>.

FAVERO, L. P. et al. **Manual de Análise de Dados: Estatística e Modelagem Multivariada com Excel, SPSS e Stata**. 1. ed. Rio de Janeiro: Elsevier, 2017.

GUELLIL, I. et al. Social big data mining: A survey focused on opinion mining and sentiments analysis. In: **2015 12th International Symposium on Programming and Systems (ISPS)**. [S.l.: s.n.], 2015. p. 1–10.

Infopédia. **Inteligência**. 2025. Acesso em: 5 maio 2025. Disponível em: <https://www.infopedia.pt/dicionarios/lingua-portuguesa/inteligÃłncia>.

IQBAL, M. et al. Bias-aware lexicon-based sentiment analysis. In: **Proceedings of the 30th Annual ACM Symposium on Applied Computing**. New York, NY, USA: Association for Computing Machinery, 2015. (SAC '15), p. 845–850. ISBN 9781450331968. Disponível em: <https://doi.org/10.1145/2695664.2695759>.

JURAFSKY, D. et al. **Speech and Language Processing**. 2. ed. [S.l.]: Prentice Hall, 2019. ISBN 9780131873216.

JURAFSKY, D. et al. **Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models**. 3. ed. [s.n.], 2025. Online manuscript released January 12, 2025. Disponível em: <https://web.stanford.edu/~jurafsky/slp3>.

LIU, B. **Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data**. 2. ed. Berlin, Heidelberg: Springer, 2011. ISBN 978-3-642-19459-7.

LIU, B. **Sentiment Analysis and Opinion Mining**. Morgan & Claypool Publishers, 2012. Disponível em: <https://www.cs.uic.edu/~liub/FBS/SentimentAnalysis-and-OpinionMining.pdf>.

LIU, Y. et al. Roberta: A robustly optimized bert pretraining approach. **arXiv preprint arXiv:1907.11692**, 2019. Disponível em: <https://arxiv.org/abs/1907.11692>.

LOUGHRAN, T. et al. When is a liability not a liability? textual analysis, dictionaries, and 10-ks. **The Journal of Finance**, Wiley Online Library, v. 66, n. 1, p. 35–65, 2011.

MAQBOOL, J. et al. Stock prediction by integrating sentiment scores of financial news and mlp-regressor: A machine learning approach. **Procedia Computer Science**, v. 218, p. 1067–1078, 2023. ISSN 1877-0509. International Conference on Machine Learning and Data Engineering. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1877050923000868>.

MCCULLOCH, W. S. et al. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, Springer, v. 5, n. 4, p. 115–133, 1943.

MISHKIN, F. S. et al. **Financial Markets and Institutions**. Boston: Addison Wesley, 1995.

MITCHELL, T. M. **Machine Learning**. New York: McGraw-Hill, 1997. ISBN 9780070428072.

MURPHY, J. J. **Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications**. 1st. ed. New York: New York Institute of Finance, 1999.

PATIL, A. et al. Sentiment analysis of financial news and its impact on the stock market. In: **2024 2nd World Conference on Communication & Computing (WCONF)**. [S.l.: s.n.], 2024. p. 1–5.

Petrobras. **Perfil - Quem Somos**. 2025. <https://petrobras.com.br/quem-somos/perfil>. Acesso em: 6 maio 2025.

PETRUSHEVA, N. et al. Comparative analysis between the fundamental and technical analysis of stocks. **Journal of Process Management. New Technologies**, v. 4, n. 2, p. 26–31, 2016.

RUSSELL, S. J. et al. **Artificial Intelligence: A Modern Approach**. 3. ed. Upper Saddle River, NJ: Prentice Hall, 2010. ISBN 978-0136042594.

SCHRANK, J. The impact of a crisis on monetary policy's influence on financial markets: Evidence from the covid-19 pandemic. **Cogent Economics & Finance**, Cogent OA, v. 12, n. 1, p. 2322874, 2024. Disponível em: <https://doi.org/10.1080/23322039.2024.2322874>.

SILVA, W. D. C. **Correlação da variação de preço de ações a partir da análise de sentimento de notícias financeiras**. Trabalho de Conclusão de Curso (Graduação) — Universidade Estadual de Maringá (UEM), 2020.

TABOADA, M. Sentiment analysis: An overview from linguistics. **Annual Review of Linguistics**, Annual Reviews, v. 2, n. Volume 2, 2016, p. 325–347, 2016. ISSN 2333-9691. Disponível em: <https://www.annualreviews.org/content/journals/10.1146/annurev-linguistics-011415-040518>.

# Apêndice A - Código simplificado da extração de notícias do site infomoney.com

```
lista_url_infomoney = ["url1, url2, ..., urln"]

def extrair_infomoney(url: str) -> dict:
    # Extraindo Titulo
    titulo = soup.find("h1")
    # Extraindo Data
    data = soup.find("time")
    # Extraindo Conteudo
    conteudo_div = soup.find("article")
    if conteudo_div:
        paragrafos = conteudo_div.find_all("p")
        conteudo = "\n".join(p.get_text(" ", strip=True) for p in
            paragrafos)

    return {
        "url": url,
        "titulo": titulo,
        "data": data,
        "conteudo": conteudo,
        "site": "Infomoney"
    }

# Agregando a lista "noticias"
noticias = []
for url in lista_url_infomoney:
    dados_noticia = extrair_dados_infomoney(url)
    if dados_noticia:
    noticias.append(dados_noticia)
```

# Apêndice B - Função de pré-processamento

```python
def preprocess_text(self, text) -> Tuple[List[str], str]:
    # Limpa o texto
    cleaned_text = self.clean_text(text)
    # Processa com spaCy (transformacao do cleaned_text em objeto Doc do spacy)
    doc = self.nlp(cleaned_text)
    # Extrai tokens: remove stop words, pontuacao, espacos e faz lematizacao
    processed_tokens = []
    for token in doc:
        # Ignora stopwords, pontuacao, espacos e tokens muito curtos
        if (token.text.lower() not in stopwords_set and
        not token.is_punct and
        not token.is_space and
        len(token.text) > 2 and
        token.text.isalpha()):
        # Adiciona o lema em minusculo
        processed_tokens.append(token.lemma_.lower())

    return processed_tokens, ' '.join(processed_tokens)
```

# Apêndice C - Função de correlação dos dados por período

```python
def pearson_report(x, y):
    mask = x.notna() & y.notna()
    if mask.sum() < 3:
        return np.nan, np.nan
    r, p = pearsonr(x[mask], y[mask])
    return r, p

def correlation_by_period(df, sentiment_col, return_cols, period='Q'):
    """
    Calcula correlacao por periodo

    Parametros:
    -----------
    df : DataFrame
    sentiment_col : nome da coluna de sentimento
    return_cols : lista de colunas de retorno
    period : 'Q' (trimestre), 'M' (mes), 'Y' (ano)
    lag: numero do periodo de defasagem
    """
    results = []

    for col in return_cols:
        for period_label, group in df.groupby(pd.Grouper(freq=period)):
            # aplica a defasagem
            shifted_returns = group[col].shift(-lag) if lag != 0 else group[col]

            # remove NaNs criados pelo shift
            valid = group[[sentiment_col]].join(shifted_returns).dropna()

            # precisa de pelo menos 2 pontos para correlacao
            if len(valid) > 1:
            r, p = pearson_report(valid[sentiment_col], valid[col])
            else:
            r, p = None, None
```

```
35
36              results.append({
37                  'periodo': period_label,
38                  'acao': col,
39                  'correlacao': r,
40                  'p_valor': p,
41                  'n_observacoes': len(group),
42                  'num_noticias': group['count_news'].sum(),
43                  'lag': lag
44              })
45
46      return pd.DataFrame(results)
```