Categorização de emoções em texto usando Processamento de Linguagem Natural (PLN)

Gabriel de Mello Cambuy Ferreira
RA: 142641
gabriel.cambuy@unifesp.br

Nicole Cristine de Faria Santos
RA: 156636
nicole.cristine@unifesp.br

***Abstract* — This article describes the development of a supervised learning NLP algorithm for a generalized sentiment analysis model.**

Keywords— Artificial Intelligence, NLP, Sentiment Analysis, Machine Learning.

I. Introdução e Motivação

No contexto contemporâneo, as mídias sociais se consolidaram como plataformas fundamentais para a comunicação e expressão de indivíduos ao redor do mundo. Diariamente, bilhões de usuários compartilham pensamentos, sentimentos e experiências através de textos publicados em redes como X, Pinterest, Filmow Facebook e Instagram. Esse vasto fluxo de informações textuais oferece um terreno fértil para a análise e compreensão do comportamento humano.

Nesse sentido, o reconhecimento de emoções em textos de mídias sociais, utilizando técnicas de Processamento de Linguagem Natural (PLN), emerge como um campo de pesquisa extremamente relevante. A habilidade de decifrar e categorizar emoções expressas em postagens pode trazer benefícios significativos em diversas áreas, desde o marketing e a publicidade até a saúde mental e a segurança pública. Por exemplo, empresas podem ajustar suas estratégias de comunicação com base nas emoções predominantes dos consumidores, enquanto instituições de saúde podem identificar sinais de depressão ou ansiedade através das publicações de indivíduos.

No entanto, é fundamental destacar que a linguagem natural é complexa, pois uma única palavra pode ter múltiplos significados dependendo do contexto em que é utilizada na frase, além de possuir um vocabulário muito vasto. Por esse motivo, é necessário um sistema robusto para interpretar adequadamente o significado das palavras em diferentes contextos e inferir emoções a partir do texto, sendo particularmente exigente em termos de gasto computacional.

Portanto, esse trabalho tem como objetivo criar um modelo de inteligência artificial para classificar postagens de mídias sociais de acordo com as emoções identificadas, utilizando técnicas de PLN e árvores de decisão.

II. Conceitos Importantes e Trabalhos Relacionados

A. Conceitos importantes

Para compreender o trabalho proposto, é necessário ter conhecimento sobre alguns conceitos fundamentais, como conjuntos de dados, processamento de linguagem natural (PLN), aprendizado supervisionado e análise semântica. O primeiro passo para resolver um problema de inteligência artificial é selecionar um conjunto de dados, que servirá como base para o desenvolvimento de um algoritmo capaz de interpretar a linguagem humana. O PLN, um dos ramos da IA, permite que o algoritmo entenda a linguagem humana por meio de um modelo estatístico que interpreta o texto. O PLN é amplamente utilizado em diversas aplicações, como assistentes virtuais e chatbots.

Neste trabalho, o modelo de aprendizado supervisionado será empregado, dividindo o conjunto de dados em duas partes: uma para treinamento e outra para teste. O algoritmo de PLN aprenderá os padrões do conjunto de treinamento, e sua eficiência será avaliada no conjunto de testes até que atinja um desempenho satisfatório. Para interpretar o texto de forma adequada, a IA precisa realizar uma análise semântica eficaz, o que pode representar um desafio, dado que um texto pode ter múltiplas interpretações e incluir figuras de linguagem como hipérbole, eufemismo e sarcasmo.

B. Trabalhos relacionados

O artigo Sentiment Analysis on Twitter by Using TextBlob for Natural Language Processing[1] publicado em 2020, utiliza a biblioteca TextBlob para fazer uma análise de sentimento em postagens feitas na rede social Twitter atribuindo um rótulo de sentimento para a publicação. A coleta de dados foi feita diretamente através de uma API do Twitter, e com isso, surgem alguns desafios como lidar com postagens de spam. O artigo The State-of-the-Art in Twitter Sentiment Analysis: A Review and Benchmark Evaluation publicado em 2018 desta o aumento no interesse de análise de sentimento de redes sociais devido ao alto volume de dados que é gerado todos os dias e cita os desafios como spam e figuras de linguagem que geram ruído nos dados e outros desafios como a informalidade e brevidade das postagens realizadas na rede social.

III. Objetivos

A partir da execução do algoritmo de PLN com a abordagem de aprendizagem supervisionada na base de dados do Twitter, será testada a eficiência do modelo criado e se o modelo apresenta um comportamento generalizado para a análise de sentimentos, ou seja, se é capaz de realizar a análise em outras bases de dados que não sejam exclusivamente de postagens em uma rede social, mas que podem, da mesma forma, serem classificados em positivo, negativo ou neutro

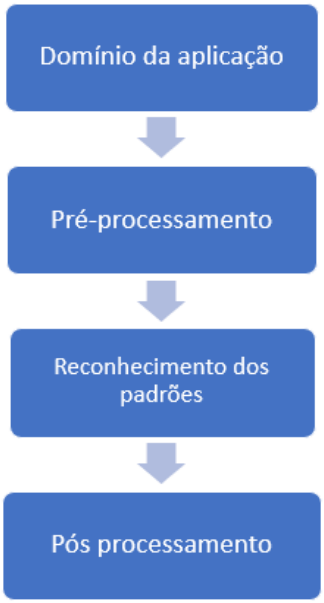
como uma base de dados de avaliações de produtos de um e-commerce e uma base de dados de avaliações de filmes..

IV. Metodologia Experimental

A metodologia experimental seguiu um conjunto estruturado de etapas que possibilitaram a análise dos dados e a construção de um modelo eficaz para a classificação das emoções presentes nas informações coletadas.

Os tweets coletados passam por um pré-processamento para limpeza dos dados. Usando métodos de seleção de características, as características importantes são extraídas do texto limpo. Os dados são divididos e rotulados manualmente como tweets negativos, positivos e neutros para a construção de um conjunto de treinamento. As características extraídas e o conjunto de treinamento rotulado são fornecidos como entrada para o classificador, que é utilizado para criar o conjunto de teste, conforme fluxograma abaixo:

Figura 1 - Fluxograma Processamento de dados



Fonte: Autoria própria.

A. *Bases Utilizadas*

Inicialmente, para a construção do modelo de aprendizagem, será utilizado uma base de dados que contém 8.199 linhas de postagens extraídas da **rede social Twitter**. Cada linha inclui a data de criação do tweet, as coordenadas de latitude e longitude, o nome de usuário de quem postou, a classificação do tweet (neutro, positivo ou negativo) e várias outras colunas que estão sem informações.Posteriormente, após a aplicação dos modelos de treinamento e a comparação entre eles, foram utilizadas duas outras bases de dados para validação.

As outras bases para testes do modelo de aprendizagem foram as seguintes:

- **Reviews de produtos em e-commerce, coletados do site Buscapé:** consiste em avaliações de produtos em português coletadas em 2013, com mais de 80.000 amostras do Buscapé, um site de busca de produtos e preços;
- **Reviews de filmes:** Essa é a maior base de dados considerada neste trabalho, com quase 2 milhões de avaliações, contém resenhas de filmes coletadas da Filmow, uma popular rede social de filmes,

B. *Exploração das Bases de dados*

Ao obter o conjunto de dados, foi necessário entender como eles estão estruturados. Esse processo foi feito através da exploração da base de dados com comandos que mostram o tipo dos dados de cada coluna, quantidade de valores nulos por coluna, exemplos de linhas do dataset e posteriormente será feito uma nuvem de palavras para entender as palavras mais comuns do dataset e determinar quais tratamentos serão necessários na etapa de pré-processamento.

C. *Pré-processamento*

Após a fase de exploração da base de dados de 8.199 linhas contendo postagens realizadas na rede social twitter foi necessária a limpeza da base de dados que contém 15 colunas vazias e uma coluna com apenas uma informação que não acrescentam na análise da base de dados..

Nessa etapa, cabe ressaltar que a mineração de dados no Twitter é um processo difícil, pois envolve dados brutos, sendo essencial limpá-los utilizando os seguintes métodos: Hashtags (#), retweets (RT) e identificadores de conta (@) precisam ser removidos. Símbolos, URLs, hiperlinks, dados não textuais e emoticons também são removidos, já que apenas os dados textuais são necessários. Palavras de parada como "ao", "os", "aquela", etc., não expressam emoções, portanto, são removidas para descomprimir o conjunto de dados. Palavras com letras repetidas, como "amorr", são comprimidas para "amor". Gírias como "c8" e "g9" são descomprimidas, pois são adjetivos ou substantivos que indicam o nível mais alto de sentimento. A remoção dessas palavras é essencial.

Assim, nessa etapa do projeto foram feitos alguns tratamentos em cada uma das bases de dados:

Base de dados Twitter:

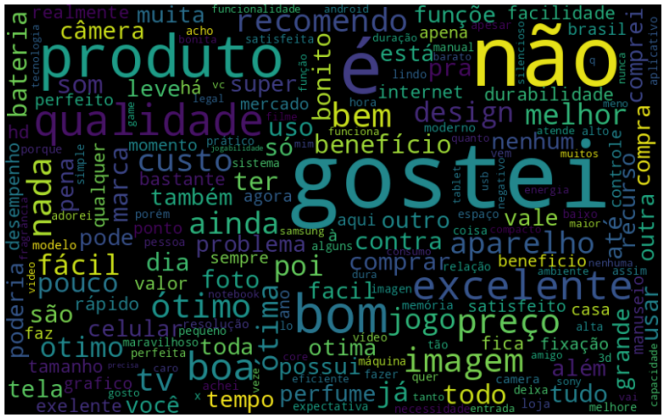
1) **Remoção de colunas nulas:** exclusão de diversas colunas com informações em branco, que não agregam em nada nas análises e modelagem
2) **Limpeza de pontuação e símbolos**: Remoção de caracteres especiais e pontuações que não contribuem para o conteúdo semântico dos tweets.
3) **Remoção de hashtags e menções:**: Descartar hashtags (#hashtag) e as menções a outros usuários em retweets;
4) **Remoção de Stopwords:** Exclusão de palavras comuns que não têm valor semântico significativo, como "e", "a", "o", "de", etc.

Bases de dados Reviews de produtos e filmes:

1) **Remoção de colunas desnecessárias:** Foram removidas 3 ou mais colunas que não agregaram nas análises:
2) **Remoção de valores nulas (NaN):** Na base de dados, há polaridades expressas como "NaN". Como não podemos assumir nenhuma definição nesses dados, eles foram retirados do dataset.
3) **Transformação da polaridade em classificação textual:** Em ambas as bases, a classificação preliminar nas emoções vieram em valores binários (0 ou 1), sendo que 0 representava as emoções negativas e 1 as positivas. Assim, foi necessário criar uma coluna adicional que as descrevessem como negativas e positivas.

Para explorar a bases de dados de texto em linguagem natural gráficos convencionais como barras ou de setores não são satisfatórios. Para conseguir explorar a base de dados e entender algum padrão nos dados foram feitas nuvens de palavras.

Figura 2 - Nuvem de palavras da base de dados de reviews de produtos



Fonte: Autoria própria

A nuvem de palavras acima está com um filtro de comentários classificados como positivos. Através da nuvem de palavras, já conseguimos identificar alguns produtos que potencialmente são avaliados na base pelas palavras: tv, celular, perfume e notebook. Além do entendimento geral do que essa base de dados possui de reviews, podemos visualizar alguns dos desafios de lidar com texto em linguagem natural como as palavras “funções” e “funçõe” que indicam o mesmo sentido porém frequentemente foram digitadas de forma incompleta.

D. Reconhecimento dos padrões

Nesta etapa do trabalho, foram escolhidos os algoritmos de aprendizado a serem empregados, assim como as bibliotecas necessárias para a aplicação dos métodos e para a visualização dos resultados obtidos.

1) Bibliotecas utilizadas:

- **NumPy**: Biblioteca fundamental para computação científica em Python. Fornece suporte para arrays multidimensionais e funções matemáticas eficientes.
- **Pandas**: Biblioteca para manipulação e análise de dados. Oferece estruturas de dados como DataFrames que facilitam a limpeza e análise de grandes conjuntos de dados.
- **Scikit-learn**: Biblioteca para aprendizado de máquina em Python. Inclui ferramentas para treinamento e avaliação de modelos, além de pré-processamento e seleção de características.
- **TextBlob**: Biblioteca para processamento de linguagem natural (PLN). Facilita a análise de sentimentos, tradução e outras tarefas de PLN com uma API simples.
- **re**: Módulo da biblioteca padrão de Python para manipulação de expressões regulares. Utilizado para busca e substituição de padrões em textos.
- **Matplotlib**: Biblioteca para criação de visualizações estáticas em Python. Permite a geração de gráficos e plots para análise de dados.
- **Seaborn**: Biblioteca de visualização baseada em Matplotlib que fornece uma interface de alto nível para criar gráficos estatísticos informativos e atraentes.
- **nltk**: Biblioteca para processamento de linguagem natural. Oferece ferramentas para tokenização, análise de sentimentos, e outras operações de PLN.
- **WordCloud**: Biblioteca para criar nuvens de palavras a partir de texto. Utilizada para visualização de frequência de palavras de maneira gráfica e intuitiva.

2) Separação dos dados em treino e teste:

Na etapa de divisão dos dados, utilizamos a função *train_test_split* da biblioteca *sklearn.model_selection* para separar o conjunto de dados em duas partes distintas: um conjunto de treinamento e um conjunto de teste. Este processo é essencial para avaliar a performance do modelo de aprendizado de máquina de maneira justa e robusta. O código realiza a seguinte divisão:

- **Conjunto de Treinamento (80%)**: Contém 80% dos dados originais e é utilizado para treinar o modelo.
- **Conjunto de Teste (20%)**: Contém os 20% restantes dos dados e é utilizado para avaliar a capacidade do modelo em generalizar para novos dados.

Após a divisão, o código exibe os conjuntos de treinamento e teste, permitindo uma visualização inicial dos dados que serão utilizados para treinar e avaliar o modelo. Essa etapa é crucial para garantir que o modelo tenha dados suficientes para aprender e também para verificar seu desempenho em dados não vistos.

3) Cross-validation;

Utilizou-se a técnica de validação cruzada com K-Fold, dividindo os dados em 5 subconjuntos (folds) para avaliar a robustez do modelo. O parâmetro *shuffle=True* foi utilizado para garantir que os dados sejam embaralhados antes de cada divisão, e o *random_state=42* assegura a reprodutibilidade dos resultados.

4) Naive Bayes:

O algoritmo **Naive Bayes** é uma técnica de classificação probabilística baseada no Teorema de Bayes, que assume que as características são independentes umas das outras. No contexto da classificação de sentimentos em tweets, o **Multinomial Naive Bayes** é particularmente adequado para lidar com dados textuais.

5) Regressão logística:

O método *LogisticRegression* é aplicado para treinar um modelo de regressão logística, que é uma técnica de aprendizado supervisionado utilizada para classificar dados em categorias discretas. O parâmetro *max_iter=1000* é ajustado para garantir que o modelo tenha tempo suficiente para convergir durante o treinamento.

V. RESULTADOS E DISCUSSÕES

Para avaliar o desempenho dos modelos de aprendizado, foram exploradas diversas combinações de etapas de processamento e algoritmos. No contexto da base de dados de tweets, foram testadas três combinações distintas, cada uma englobando uma abordagem específica para processamento e modelagem, sendo elas:

- Divisão em treino e teste + Algoritmo de Naive Bayes;
- Cross validation + Algoritmo de Naive Bayes;
- Cross validation + Regressão logística;

1. BASE DE DADOS TWITTER

a) Divisão em treino/ teste e Algoritmo de Naive Bayes

Ao testar a combinação do Algoritmo de Naive Bayes + Divisão preliminar dos dados em treino e teste, foi obtido a seguinte matriz de confusão:

Figura 3 - Matriz de confusão (Treino/teste x Naive Bayes)

```
Matriz de Confusão:
 [[182  16   1]
 [ 26 353  14]
 [  7  17 537]]
```

Fonte: Autoria própria

Assim, dos 1153 dados que estavam no conjunto de teste, 81 deles foram classificados incorretamente, resultado em uma acuracidade de aproximadamente, 93%, conforme imagem abaixo:

Figura 4 - Acurácia (Treino/teste x Naive Bayes)

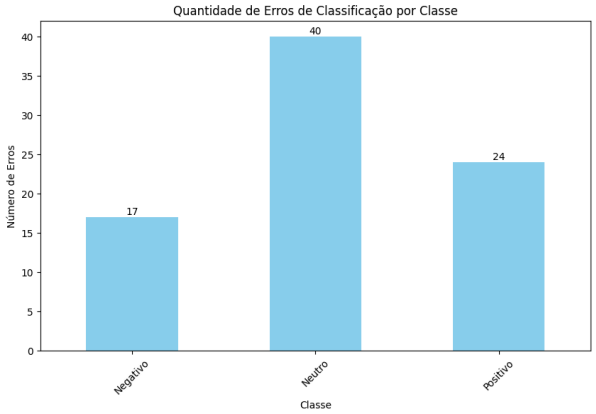
```
# Cálculo da acurácia
acuracia_TrainTest = metrics.accuracy_score(classes_test, predicoes)
print("Acurácia:", acuracia_TrainTest)

Acurácia: 0.9297484822202949
```

Fonte: Autoria própria

Em detalhes, temos que 17 tweets com classe negativa foram classificados incorretamente pelo modelo, 40 dos neutros e 24 dos positivos foram classificados incorretamente pelo modelo, conforme o gráfico abaixo:

Figura 5 - Gráfico de Erros por classe (Treino/teste x Naive Bayes)



Fonte: Autoria própria

b) Modelagem K-Folds e Algoritmo de Naive Bayes

Nessa etapa, ao testar a combinação de modelagem por K-folds combinado com o algoritmo de Naive Bayes, obtemos os seguinte matriz de confusão:

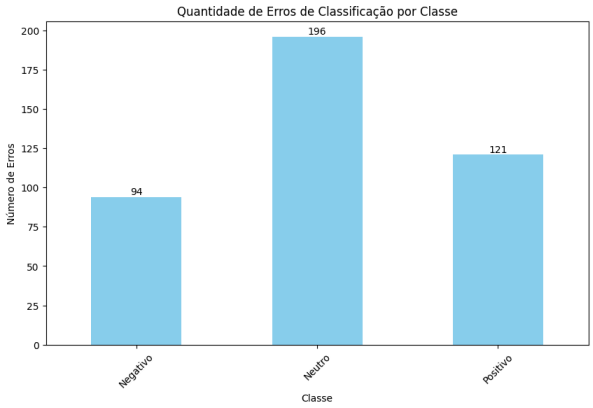
Figura 6 - Matriz de confusão (Modelagem K-Folds x Naive Bayes)

```
Matriz de Confusão:
 [[2719   19  102]
 [   5  857   89]
 [  75  121 1778]]
```

Fonte: Autoria própria

Pela matriz pode-se observar que, dos 5765 dados presentes na modelagem, 411 deles foram classificados incorretamente, resultando em uma acuracidade de 92,87%. O gráfico abaixo ilustra os erros segmentados por classe.

Figura 7 - Gráfico de erros por classe (Modelagem K-Folds x Naive Bayes)



Fonte: Autoria própria

c) Modelagem K-Folds e Regressão Logística

Por fim, a última combinação feita na base de dados retirada do twitter foi considerando o modelo de regressão logística e a divisão dos dados em K-Folds, obtendo a seguinte matriz de confusão:

Figura 8 - Matriz de confusão (Modelagem K-Folds x Regressão Logística)

```
Matriz de Confusão:
[[2762    7   71]
 [   8  848   95]
 [  33   64 1877]]
```

Fonte: Autoria Própria

Testando essa combinação aplicada aos dados, obtemos a maior acuracidade em comparação com as outras duas combinações, a **acuracidade** foi de **95%**, aproximadamente.

Figura 9 - Acurácia (Modelagem K-Folds x Regressão Logística)

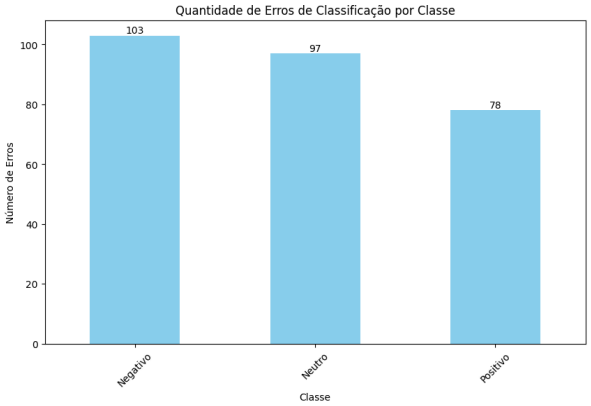
```
[ ]  # Calcular a acurácia
     acuracia = metrics.accuracy_score(classes, predicoes)
     print(f"Acurácia: {acuracia:.4f}")

     Acurácia: 0.9518
```

Fonte: Autoria própria

Assim, conforme expresso pelo gráfico abaixo, temos que dos 5.765 tweets, 103 de classe negativa foram classificados de forma errada, 92 neutros e 78 positivos foram classificados de forma errada.

Figura 10 - Gráfico de erros por classe (Modelagem K-Folds x Regressão Logística)



Fonte: Autoria própria

2. BASE DE DADOS REVIEWS PRODUTOS

Com a finalidade de verificar se a modelagem feita com os algoritmos e técnicas de pré-processamento funcionam de modo similar em outras bases de dados, aplicamos o algoritmo de Naive Bayes e Cross Validation na base de dados de review de produtos, obtendo os seguintes resultados:

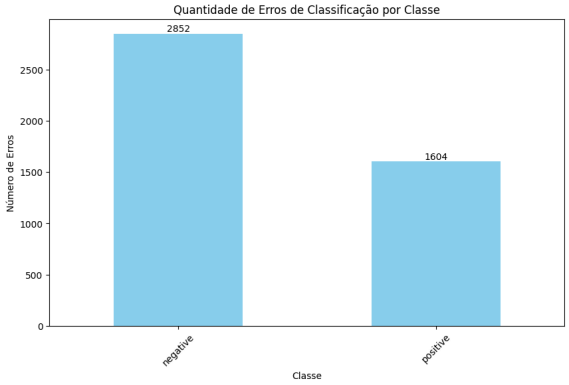
Figura 11 - Matriz de confusão (Modelagem K-Folds x Naive Bayes)

```
Matriz de Confusão:
[[65212  1604]
 [ 2852  3958]]
```

Fonte: Autoria Própria

Como nessa base de dados, só existiam classificações preliminares de avaliações positivas e negativas, foi possível observar que dos 73.626 dados, 4.456 foram previstos incorretamente, conforme representação gráfica abaixo:

Figura 12 - Gráfico de erros por classe (Modelagem K-Folds x Naive Bayes)



Fonte: Autoria Própria

Dessa forma, foi obtido uma **acuracidade** de aproximadamente, **94%.**

3. BASE DE DADOS REVIEWS FILMES

Nesta outra base de dados, também foi testado a combinação dos algoritmos de Naive Bayes e K-Folds. Dessa forma, é possível analisar que das 1.157.800 avaliações, 105.797 foram classificados incorretamente, obtendo-se uma **acuracidade de 91%**

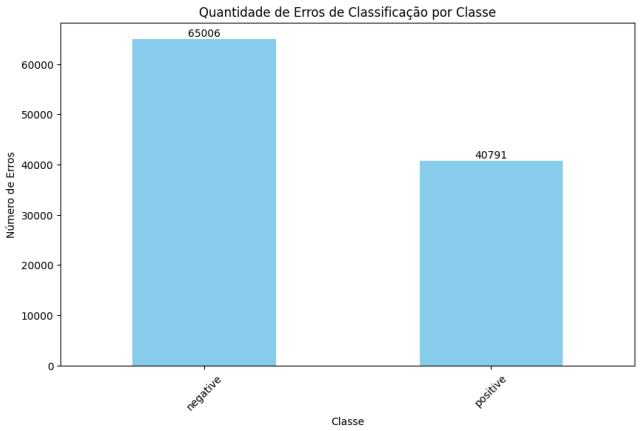
Figura 12 - Matriz de confusão (Modelagem K-Folds x Naive Bayes)

```
Matriz de Confusão:
 [[1011212   40791]
 [  65006   72533]]
```

Autoria própria

Por fim, de acordo com o gráfico abaixo, os erros foram segmentados da seguinte forma: 65.006 das avaliações negativas foram previstas incorretamente e 40.791 das positivas foram classificadas do mesmo modo.

Figura 13 - Gráfico de erros (Modelagem K-Folds x Naive Bayes)



Fonte: Autoria própria

CONCLUSÃO

Neste trabalho, abordamos a aplicação de técnicas de inteligência artificial para a classificação de sentimentos em tweets, explorando diferentes abordagens e algoritmos de aprendizado de máquina. Através da análise e pré-processamento dos dados, conseguimos preparar um conjunto robusto de informações para a construção e avaliação de modelos.

Os resultados revelaram que o Naive Bayes, quando combinado com validação cruzada, atingiu uma acurácia de 91% para as avaliações de filmes e 94% para as avaliações de produtos. Esses altos índices de acurácia indicam que o modelo é eficaz na classificação de sentimentos, tanto em avaliações de produtos quanto em filmes.

No entanto, uma análise mais detalhada dos erros revelou que a maior taxa de erros ocorreu nas classes negativas. Esse padrão sugere que o modelo teve dificuldades específicas em identificar e classificar corretamente os sentimentos negativos em ambas as bases de dados. Esse desafio pode estar relacionado à complexidade dos sentimentos expressos nos tweets ou à variabilidade nas expressões de negatividade em diferentes contextos.

A discrepância nas taxas de erro para a classe negativa, apesar da alta acurácia geral, destaca a necessidade de uma atenção especial na modelagem de sentimentos negativos. Futuras abordagens podem incluir técnicas avançadas de pré-processamento, como análise de sentimentos mais refinada e tratamento específico para dados negativos, para melhorar a precisão nessa categoria.

Em resumo, o uso do Naive Bayes com validação cruzada demonstrou ser uma abordagem sólida para a classificação de sentimentos, com uma acurácia robusta em diferentes conjuntos de dados. No entanto, os maiores erros na classificação de sentimentos negativos apontam para áreas de

melhoria que podem ser exploradas em trabalhos futuros, visando aprimorar a capacidade do modelo em lidar com emoções negativas de forma mais precisa.

Referencias Bibliográficas

[1] Hazarika, Ditiman & Konwar, Gopal & Deb, Shuvam & Bora, Dibya. (2020). Sentiment Analysis on Twitter by Using TextBlob for Natural Language Processing. 63-67. 10.15439/2020KM20.

[2] FREITAS, Abner. NLP Buscapé Data PT-BR: Sentiment Analysis. Kaggle, 2021. Disponível em: https://www.kaggle.com/code/abnerfreitas/nlp-buscape-data-ptbr-sentiment-analysis.

[3] DAVID, A. J. Portuguese NLP. GitHub, 2023. Disponível em: https://github.com/ajdavidl/Portuguese-NLP

[4] SILVA, Felipe. *Análise de Sentimentos: Aprenda de uma vez por todas como funciona utilizando dados do Twitter*. iMasters, 2023. Disponível em: https://imasters.com.br/desenvolvimento/analise-de-sentimentos-aprenda-de-uma-vez-por-todas-como-funciona-utilizando-dados-do-twitter