Uso de Inteligência Artificial para encontrar a probabilidade de tendências depressivas em frases da língua portuguesa

Use of Artificial Intelligence to find a probability of depressive tendencies in Portuguese Language sentences.

Lucas Rabelo Moreaux, Renan Soares da Silva,Carlos Eduardo Vasconcelos Silva, Guilherme Silva Carbonesi, Vitor Selloti de Souza[1]

Orientador: Professor Me. Gilberto Alves Pereira

Faculdade Impacta de Tecnologia
São Paulo, SP, Brasil
Setembro de 2022

RESUMO

Na sociedade contemporânea atual cada vez mais casos de depressão estão sendo diagnosticados, a partir disso foi pensando em um sistema de Inteligência Artificial que irá utilizar uma vertente da inteligência artificial que ajuda computadores a entender, interpretar e manipular a linguagem humana conhecida como PLN (Processamento de Linguagem Natural) para realizar a Análise de Sentimentos em textos, e identificar se o texto contém uma mensagem depressiva. Este trabalho tem o objetivo de desenvolver um protótipo de software que seja capaz de identificar padrões psicológicos depressivos em *textos livres da língua portuguesa. Para a realização deste trabalho, foi utilizada a linguagem Python com a biblioteca NLTK, a qual é alicerçada no teorema de Naive Bayes. Para criar a base de dados de teste, foram utilizados 2000 textos com tendências depressivas e 2000 textos randômicos coletados do Big Data do Twitter, que foram classificados manualmente. Em seguida foi realizado um tratamento dessa lista, composto pelas fases de remoção de palavras irrelevantes, palavras que não possuem peso na análise, e de extração de radical, para melhor aproveitamento do vocabulário. Com a base de treinamento já tratada, foi realizado o treinamento do algoritmo e obteve uma acurácia superior a 70% de probabilidades de tendências depressivas nos textos analisados

Palavras-chaves: Big Data. Inteligência Artificial. Análise de Sentimento. Processamento de Linguagem Natural.

ABSTRACT

The society that will use computers to understand and manipulate human language known as NLP (Natural Language Processing) to perform Sentiment Analysis on texts, and identify whether the text contains a depressive message. This work aims to develop that, when developing a software that can present a free text, it is able to identify consumption defects. To carry out this work, the Python language was used with the NLTK library, which is based on the Naive Bayes theorem. To create a database, 200 texts with pressive trends and 200 texts from random Twitter Big Data test texts were used, which were manually classified. In was carried out, a treatment consists of steps to remove irrelevant words, which do not have weight analysis, and then, for a better usable list. With the training base already treated, the training of the training of superior precision training was performed and we obtained the probability of tests of a test of 70% of probabilities of a test.

Keywords: 1. Big Data; 2.Artificial Intelligence; 3. Sentiment Analysis; 4 Natural Language Processing

1 Os autores podem ser contatados respectivamente pelos seus correios eletrônicos: carlos.eduardo@aluno.faculdadeimpacta.com.br, lucas.moreaux@aluno.faculdadeimpacta.com.br, guilherme.carbonesi@aluno.faculdadeimpacta.com.br, renan.silva@aluno.faculdadeimpacta.com.br, vitor.souza@aluno.faculdadeimpacta.com.br

1 INTRODUÇÃO

Com o aumento dos problemas de saúde mental entre os jovens, de acordo com Lopes, (2022), cresceu a procura por psicólogos, principalmente no período da pandemia, quando os problemas de saúde mental tiveram um aumento. A situação atual mostra que o volume de informações que são geradas diariamente cresceu tanto que se tornou necessário construir métodos, técnicas e ferramentas para conseguir minimamente aproveitar essa riqueza. Graças à internet e às redes sociais, as pessoas podem se expressar de uma maneira mais livre com imagens, textos e vídeos alcançando um número maior de pessoas, as quais compõem a maior parte do fluxo de dados que circula pela internet.

Por meio de algoritmos inteligentes, em particular na Mineração de Dados, é possível filtrar e fazer uma breve análise de informações relevantes para um objetivo específico dentro desses terabytes diários que são gerados. Esses dados trazem informações muito valiosas que podem ser trabalhadas de diversas maneiras: muitas empresas já utilizam essas informações para se beneficiar comercialmente, porém o uso delas para outras áreas pode ser muito bem aproveitado também, com um cunho social ou médico, por exemplo. (Duque, J. W. G. Raymundo, A. L., & Neto, 2018)

Para esse tipo de análise, faz-se necessário um processamento mais complexo, pois, para uma máquina realizar interpretações de sentimentos humanos, é necessária uma inteligência artificial (IA) bem treinada para isso utilizando de processamento de linguagem natural. O objetivo desta pesquisa é criar um algoritmo de processamento de linguagem natural que calcula a probabilidade de tendências depressivas de uma frase digitada na língua portuguesa.

2 APRENDIZAGEM DE MÁQUINA

A primeira pessoa que usou a frase “aprendizagem de máquina” foi Arthur Samuel, que desenvolveu um dos primeiros programas de computador para jogar damas.

Conforme Samuel (1959), “Tecnologia que dá aos computadores a capacidade de aprender sem serem explicitamente programados”.

A aprendizagem de máquina é um subconjunto da inteligência artificial, o segmento da ciência da computação que se concentra na criação de computadores que pensam de forma semelhante aos humanos. Em outras palavras, todos os sistemas de aprendizado de máquinas são sistemas de Inteligência Artificial (IA), mas nem todos os sistemas de IA possuem capacidades de aprendizado de máquina (INDÚSTRIA 4.0 2019).

2.1 Tipos de inteligência artificial

De acordo com Ludermir (2021), IA pode ser caracterizada em três tipos: IA Focada, IA Generalizada e IA Superinteligente.

A IA Focada, também conhecida como IA Fraca, consiste de algoritmos especializados em resolver problemas em uma área e/ou um problema específico. Aqui os sistemas armazenam uma grande quantidade de dados e os algoritmos são capazes de realizar tarefas complexas, porém sempre focadas no objetivo para o qual foram desenvolvidos (LUDERMIR, 2021).

Na IA Generalizada, também conhecida como IA Forte, os algoritmos desenvolvidos se tornam tão capazes quanto humanos em várias tarefas e, em geral, os algoritmos usam técnicas de Aprendizado de Máquina como ferramenta (LUDERMIR, 2021).

Na IA Superinteligente, os algoritmos são significantemente mais capazes que humanos em praticamente todas as tarefas. Ainda não existem sistemas com IA Superinteligente (LUDERMIR, 2021).

2.2 Tipos de aprendizagem de máquina

Existem quatro tipos principais de *Machine Learning* (Aprendizado de Máquina): Supervisionado, Não Supervisionado, Semi-Supervisionada e por Reforço (LUDERMIR, 2021).

No Aprendizado Supervisionado, requer um programador ou "professor" que ofereça exemplos de quais entradas se alinham com os resultados (ou seja, um rótulo informando a que classe o exemplo pertence, no caso de um problema de classificação de imagens, por exemplo, como distinguir imagens de cachorros e de gatos). Cada exemplo é descrito por um vetor de valores (atributos) e pelo rótulo da classe associada. O objetivo do algoritmo é construir um classificador que possa determinar corretamente a classe de novos exemplos ainda não rotulados (LUDERMIR, 2021).

No Aprendizado Não Supervisionado, os exemplos são fornecidos ao algoritmo sem rótulos, exigindo que o sistema desenvolva suas próprias conclusões. O algoritmo agrupa os exemplos pelas similaridades dos seus atributos. O algoritmo analisa os exemplos fornecidos e tenta determinar se alguns deles podem ser agrupados de alguma maneira, formando agrupamentos ou clusters. Após a determinação dos agrupamentos, em geral, é necessária uma análise para determinar o que cada agrupamento significa no contexto/problema sendo analisado (LUDERMIR, 2021).

No Aprendizado Semi-Supervisionado, é uma combinação de aprendizado supervisionado e não supervisionado. Voltando ao exemplo do Aprendizado Supervisionado, imagine que você tenha um grande número de imagens, algumas das quais foram rotuladas como "gato", "cachorro" e algumas outras imagens sem rótulos. Um sistema de aprendizagem semi-supervisionado usaria as imagens rotuladas para fazer inferências sobre qual das imagens não marcadas inclui gatos e cachorros. As melhores suposições seriam então devolvidas ao sistema para ajudá-lo a melhorar suas capacidades e o ciclo continuaria (LUDERMIR, 2021).

No Aprendizado por Reforço, o algoritmo não recebe a resposta correta, mas recebe um sinal de reforço, de recompensa ou punição. O algoritmo faz uma hipótese baseado nos exemplos e determina se essa hipótese foi boa ou ruim. Um exemplo clássico de aprendizagem de reforço (como se aplica à aprendizagem de máquina) é um agente aprendendo a jogar um game. O objetivo é vencer o game e o agente vai sendo recompensado ou punido de acordo com seus erros e acertos, até atingir seu objetivo (LUDERMIR, 2021).

De acordo com Ludermir (2021), o uso de *Machine Learning* para solucionar problemas não é fácil, sendo necessário alguns requisitos:

a) Conjunto de exemplos. Uma boa base de exemplos, sendo atualizada constantemente.

b) Técnicas que melhorem a qualidade dos dados, já que nem sempre os dados são bons.

c) Seleção dos conjuntos de algoritmos apropriadas para o problema focado.

d) Parâmetros dos algoritmos (por exemplo, o número de camadas de uma Rede Neural).

e) Revisão dos resultados, para verificar se o problema está sendo resolvido.

f) Atualizações periódicas, mudanças nos dados podem fazer com que o sistema deixe de funcionar (LUDERMIR, 2021).

2.3 Rede Neural

Todos realizam várias tarefas diariamente as quais exigem diversos processamentos por parte do cérebro, responsável por administrar tudo o que pensam ou sentem. Baseado no desempenho do sistema nervoso, McCulloch e Pitts (1943) desenvolveram as redes neurais artificiais, que simulam o funcionamento do cérebro humano, sendo capaz de processar e aprender com os dados obtidos.

A rede neural (humana) é composta por diversas unidades, dentre elas os neurônios, que têm como função transmitir e processar dados.

Já a rede neural artificial realiza diversas conexões e se desenvolve através de treinamentos, sendo essencial em diversas áreas como na robótica e análise de dados (RITTER et al., 2017).

2.4 Mineração de dados

A mineração de dados é o processo de descoberta de informações acionáveis em grandes conjuntos de dados Microsoft (2022), quando relacionada a Granatyr (2017) que afirma que existem duas formas de abordar a mineração de dados, sendo elas estatística e a de Processamento de Linguagem Natural. A primeira é relevante a frequência em que os termos começam a aparecer e a partir disso, gera uma informação, ignorando a sintática e semântica do texto. Porém na segunda, a interpretação da semântica e sintática dos textos é a parte mais relevante, fazendo o algoritmo entender e aprender textos em linguagens. Neste projeto os dois tipos de mineração serão abordados, cada um em seu processo.

2.5 Algoritmos

Algoritmo é uma sequência ordenada e finita de instruções ou operações para a solução de um problema computacional. ALGORITMO é uma sequência ordenada e finita de instruções ou operações para a solução de um problema computacional. (Pierro, Fapesp, 2018)

Dentre os algoritmos que existem em operações computacionais, existem também algoritmos que são utilizados para a criação de machine learning e inteligência artificial, que já são feitos para conseguir otimizar e ter uma base para se trabalhar como o Naive Bayes.

2.6 Algoritmo Naive Bayes

Naive Bayes é um algoritmo probabilístico que normalmente é usado para problemas de classificação. Naive Bayes é simples, intuitivo e, no entanto, funciona surpreendentemente bem em muitos casos. Por exemplo, filtros de spam que os aplicativos de e-mail usam são criados no Naive Bayes (Granatyr, 2017).

Muito utilizado no meio acadêmico estatístico, seu racional é baseado nos estudos de Thomas Bayes, realizando uma análise do algoritmo para a base de dados. (A. Gusmão, 2022)

A função do naive bayes é classificar e gerar tabelas de probabilidades, em cima da sua lógica aplicada, assumindo que a presença de uma característica específica não se relaciona com outro recurso também contido na base de dados analisada. Permitindo fazer o aprendizado da máquina contemplar análises com diferentes dados de forma integrada ou separados. (A. Gusmão, 2022)

2.7 Processamento de linguagem natural e Text Analytics

O processamento de Linguagem Natural (PLN) é uma parte da ciência da computação responsável por utilizar inteligência artificial para melhorar a interação entre humanos e máquinas com o uso de linguagem natural. (Flavio, 2020)

Linguagem Natural é a forma de comunicação usada no dia a dia para nos comunicar.

Processamento de Linguagem Natural é um processo que consiste na transformação de textos e falas em conjuntos de dados capazes de serem interpretados para o desenvolvimento de análises ou algoritmos de aprendizagem de máquina. Textos e falas são fontes de dados não estruturados, que precisam ser tratados com técnicas de mineração de textos.

2.8 Mineração de Texto

A mineração de textos é um paradigma de programação criado para resolver este problema, sendo capaz de entender a linguagem natural dos documentos de texto e conseguindo lidar com a sua imprecisão e incerteza. A mineração de textos envolve várias áreas da informática, como mineração de dados, aprendizado de máquina, recuperação de informação, estatística e linguagem computacional, para conseguir transformar o texto em algo que um computador consiga entender (MACHADO, 2010).

O principal objetivo da mineração de textos é encontrar termos relevantes em documentos de texto com grande volume de dados e estabelecer padrões e relacionamentos entre eles com base na frequência e temática dos termos encontrados (SERAPIÃO, 2010).

A tecnologia de mineração de textos não é um mecanismo de busca, pois a mineração ajuda o usuário a descobrir informações previamente desconhecidas, enquanto na busca o usuário já sabe o que deseja procurar. Além disso, a mineração 59 também é diferente de robôs de conversação (chatterbot), pois ela não tenta simular o comportamento humano (ARANHA; PASSOS, 2006).

3 Iniciativas na Área

De acordo com Graham (2019) ‘À medida que as técnicas de IA continuam a ser refinadas e aprimoradas, será possível ajudar os profissionais de saúde mental a redefinir as doenças mentais de forma mais objetiva do que atualmente é feito.

Em Suma, a utilização de inteligências artificiais na área de saúde, com destaque a áreas relacionadas à saúde mental vem se aprimorando e diversos artigos foram publicados, entre eles um artigo publicado pela nature (Predictive modeling of depression and anxiety using electronic health records and a novel machine learning approach with artificial intelligence, Scientific Reports, 2021) é uma pesquisa ao qual se utilizou de registros eletrônicos de saúde (EHRs) junto de um novo pipeline de aprendizado de máquina para analisar dados de um estudo observacional para promover a detecção de transtorno de ansiedade generalizada (TAG) e transtorno depressivo maior (TDM) em ambientes de atenção prioritária. a fim de trazer uma identificação rápida dos transtornos e garantir um tratamento oportuno.

4 METODOLOGIA

O projeto foi realizado para calcular a probabilidade de tendências depressivas em textos com mais de 100 caracteres. Realizando a divisão do experimento em 3 etapas.

A primeira etapa, retira-se a base de dados contendo textos já classificados em 3 tipos, neutros, depressivos e randômicos, já classificada e realizado novas separações entre elas mesmas, para conseguir treinar o algoritmo diversas vezes com a mesma base, concluindo com um data mining entre todas as bases classificadas e processando os dados novamente no algoritmo para verificar a classificação da probabilidade obtida em cima da classificação manual.

A segunda etapa é a realização dos tratamentos das palavras para conseguir extrair os radicais, radicais são por si só o elemento básico que traz a significação a uma palavra, pela utilização da biblioteca NLTK 3.7 do Python 3.10.13, por que o NLTK é uma biblioteca da linguagem Python para Processamento de Linguagem Natural e Text Analytics, tornando-se a melhor opção para a realização dessa etapa.

Terceira etapa é onde se encontra toda a utilização das bases de dados juntamente com o tratamento da segunda etapa (aplicação do NLTK para extrair radicais das palavras), após os dados tratados se inicia a fase de treinamento, que seria a remoção das palavras classificadas como “stopwords” sendo elas palavras que não agregam valor algum para o algoritmo, como por exemplo palavras como “de”, “o”, “e” entre outras. Feito a remoção das “stopwords”, se inicia o treinamento da machine learning supervisionado utilizado para resolver problemas junto a um conjunto de dados categorizados para treinar o algoritmo para realizar a tarefa, utilizando o algoritmo NAIVE BAYES, que por si só é um algoritmo que gera uma tabela de probabilidades a partir de uma técnica de classificação de dados. A probabilidade de sua frase conter indicações depressivas utilizando machine learning.

5 BASE DE DADOS

A base de dados foi retirada do Twitter, Inc , extraindo de páginas criadas na rede social ou de textos postados pelos próprios usuários do Twitter, a extração foi realizada a partir do software FollowerWonk v1.3, software de análise do Twitter que auxilia empresas com pesquisas biográficas e segmentação de influenciadores, obtendo 2000 textos.

A primeira base a ser extraída foi de twitters depressivos ou com tendências depressivas, vindo de comunidades ou usuários que já classificam seus twitters (textos postados na rede social) como depressivos e melancólicos, após a extração desses textos já pré ditados como depressivos, foi realizada uma revisão na literatura para verificar os textos e ter a classificação exata dos textos. Para realizar essa classificação foram utilizados frases que contém palavras chaves já categorizados com cargas depressivas e melancólicas e comparadas com artigos que reforçam qual a sensação que o texto descrito traz para o leitor, deixando assim a interpretação do mesmo livre para o leitor classificar, levando em conta artigos ou sites do governo bvsms, SBIC, pepsic que informam sobre depressão e as suas causas.

A segunda base a extração é muito mais fácil e não necessariamente precisa de uma classificação, sendo elas twitters (textos) randômicos que não contém uma classificação pré ditada, sendo apenas textos sem nenhum peso ou informações úteis, textos casuais sobre o dia a dia ou comentários sobre qualquer assunto, como exemplo “ontem acordei cedo e fiz exercícios e tomei 1 café da manhã" ou “gostaria de comer 1 bolo de cenoura” entre outros tipos de textos sem nenhuma classificação. Entretanto foi realizada uma revisão nos textos para verificar se realmente não contém nenhuma tendência depressiva e melancólica

utilizando o mesmo método de classificação manual feita na primeira base, acima de artigos ou palavras chaves que já são atribuídas como depressivas.

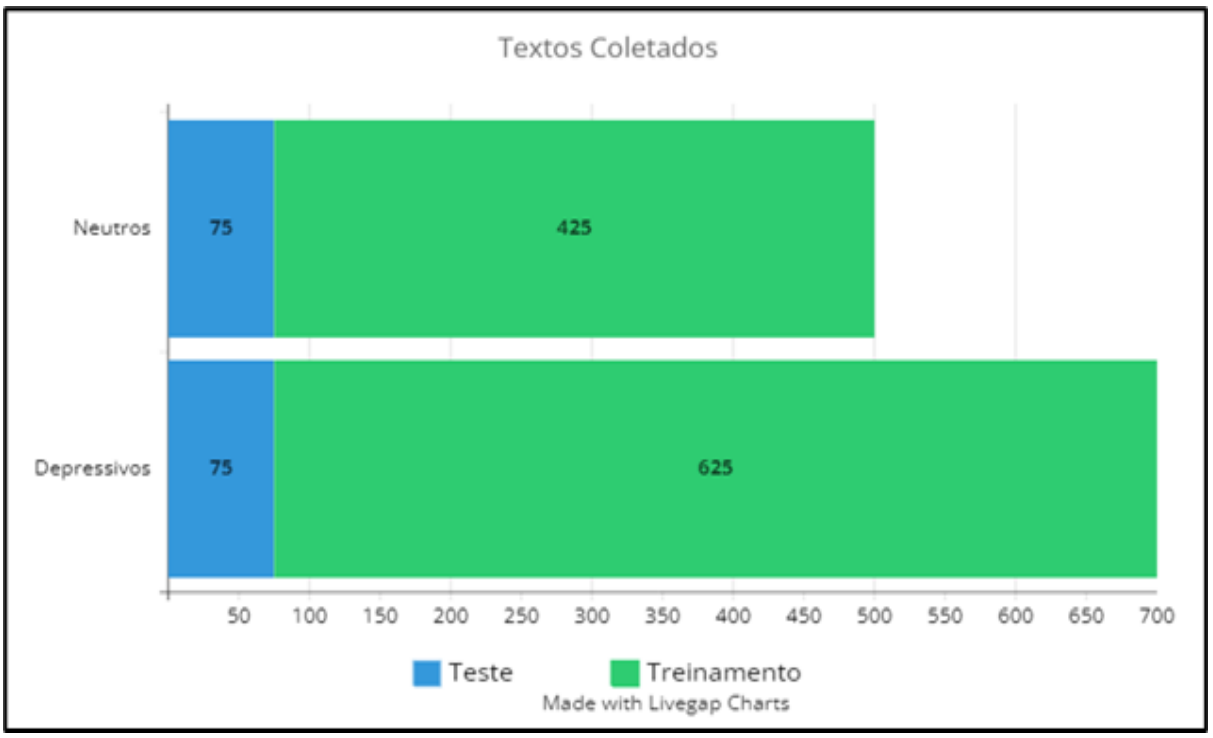
Realizado essa revisão nas bases de dados foi obtida uma base com 2000 textos sendo eles 1300 neutros e 700 depressivos.

5.1 Análise de realização

O objetivo do experimento é classificar textos com tendências depressivas. O projeto foi realizado dividindo ele em etapas, na primeira etapa, foram retirados 2000 textos em cima do big data já explicado como obtido, a dentre desses 2000 textos, 1300 são textos neutros (não tem indícios depressivos), dentre esses 1300, 1100 foram retirados para treinamento e 200 para testes do algoritmo. Os 700 textos restantes foram divididos em 625 textos com indícios depressivos e 75 para testes. A classificação foi realizada utilizando dicionários do Python. A segunda parte, foi a realização do tratamento das palavras, retirando palavras sem relevância na análise, e a terceira parte foi utilizar a biblioteca NLTK do Python. Para realização do treinamento do algoritmo, é necessário realizar a extração dos radicais das palavras, auxiliando o reconhecimento das palavras associadas.

Em seguida, foi realizado o tratamento das bases removendo stopwords (palavras que não têm relevância na análise), e extração dos radicais das palavras dos textos utilizando a biblioteca NLTK em Python. Para o treinamento do algoritmo inteligente, a extração dos radicais é importante, pois facilita o reconhecimento das palavras como por exemplo seria a palavra "sofrê", retirando o radical ficaria "sofr", criando um facilitador para encontrar palavras como sofrimento, sofrendo, sofria entre outras, na base de dados, que são classificadas como palavras depressivas, o critério para classificação das palavras depressivas foram retirados de artigos publicados online mencionados nas referências como bvsms, SBIC, pepsic. Após o treinamento do algoritmo inteligente, foi utilizada a base de testes para comparar os erros e acertos e medir a precisão que foi adquirida com a base de treinamento. Nesse projeto foi alcançado a precisão de 75% de acerto e 25% de erros: com o algoritmo NLTK, foi criada uma lista para que fossem enviadas todas as frases classificadas de maneira errada e comparadas com o que foi classificado manualmente. Com essa lista, é possível gerar uma matriz e ver a quantidade de frases que estão classificadas erradas. O classificador marcou 15 dos 75 textos neutros como depressivos e 15 dos 75 depressivos como neutros. A partir dessa lista foi possível observar as frases que foram classificadas erradas e aprimorar a base de dados para poder contornar esses erros. Como demonstrado na Figura 1:

Figura 1 - Gráfico de treinamento



Fonte: Os autores

5.2 Análise da Aplicação de PLN e Sentimentos

Processamento de linguagem natural (PLN) é uma vertente da inteligência artificial que ajuda computadores a entender, interpretar e manipular a linguagem humana.

Nesta seção apresenta-se a aplicação do processo de PLN através da fase de treinamento e validação, com base no algoritmo de Naive Bayes, no contexto de Análise de textos. Como aplicação das técnicas, tomou-se o tema “Depressão” para a construção das classes norteadoras do processo de PLN.

5.3 Fase de Treinamento

O primeiro passo foi criar duas bases de dados, uma para treinamento e outra para testes. A partir de uma coleta de textos em redes sociais, foi realizada a classificação desses textos para alimentar as bases de dados.

Segundo passo é o tratamento nas bases, retirando as “stopwords” que são basicamente palavras que não agregam valor nenhum para análise. Exemplo de “stopwords” “me” “de”, “e” , “a” , “que” dentro outros.

Na figura 2, apresenta-se os textos e classes realizadas para a remoção das stopwords.

Figura 2 - Classificação de textos e linguagem

```
1
2  import nltk
3
4  testes = [('preciso encontar uma nova forma de vida, pois so estou ficando triste atualemnte'
       ,'depressivo'),
5          ('eu nao sei o pq eu choro com tudo, parece que estou sempre triste','depressivo'),
6          ('se pensar dessa forma consigo estudar todos os dias','neutro'),
7          ('se voce tentar se formar na faculdade, talvez ajude a arrumar um emprego','neutro'),]
8
9  stopwordsnltk = nltk.corpus.stopwords.words('portuguese')
```

Fonte: Os autores

A próxima etapa foi retirar o radical das palavras, para evitar duplicidade de palavras com o mesmo radical na tabela probabilística gerada em cima do treinamento, como exemplo a palavra “sofrê” é possível analisar sofrimento, sofrido entre outras. Figura 3, apresenta a aplicação.

Figura 3 - Aplicação de stemmer para remoção de radical

```
1
2  def aplicaStemmer(texto):
3      stemmer = nltk.stem.RSLPStemmer()
4      frasesStemming = []
5      for (palavras,emocao) in texto:
6          comStemming = [str(stemmer.stem(p)) for p in palavra.split() if p not in stopwordsnltk]
7          frasesStemming.append((comStemming, emocao))
8          return frasesStemming
9
10 frasesStemming = aplicaStemmer(testes)
```

Fonte: Os autores

Para melhor visualização das modificações feitas, foi implementada uma função para retornar todas as palavras da base de treinamento já tratadas. Uma outra função foi implementada para visualizar a frequência de que cada radical aparece na base de dados: a partir dessa função, foram exibidos todos os radicais sem repetições. Por fim, foi criada a função que gera a tabela probabilística do Naive Bayes, que é o treinamento propriamente dito, conforme apresentado a seguir na Figura 4:

Figura 4 - Treinamento de IA

```
def buscaPalavras(frases):
    todasPalavras = []
    for (palavras,emocao) in frases:
        todasPalavras.extend(palavras)
    return todasPalavras

palavras = buscaPalavras(frasesStemming)

def buscaFrequencia(palavras):
    palavras = nltk.FreqDist(palavras)
    return palavras

frequencia = buscaFrequencia(palavras)

def buscaPalavrasUnicas(frequencia):
    freq = frequencia.keus()
    return freq

palavrasUnicas = buscaPalavrasUnicas(frequencia)

def extraiPalavras(documeto):
    doc = set(documento)
    caracteristicas = {}
    for palavras in palavrasUnicas:
        caracteristicas['%s' % palavras] = palavras in doc
    return caracteristicas
```

Fonte: Os autores

5.4 Fase de Validação

Após o treinamento, é necessário validar o algoritmo. Nesse processo foi realizada uma comparação entre os resultados apresentados pelo algoritmo e os dados reais da base testada, criando assim uma lista contendo os textos classificados de forma equivocada pelo algoritmo, demonstrado na Figura 11. Com essa lista criada e a função de classificação accuracy do NLTK, é possível apresentar o resultado de precisão do algoritmo, que, no caso, foi de 75% mostrado na figura 5.

Figura 5 - Lista de erros

```
erros = []

for (frase,classe) in testedepressivos:
    resultado = classificador.classify(frase)
    if resultado != classe:
        erros.append((classe, resultado, frase))

nltk.classify.accuracy(classificador,basecompletatest)
```

Fonte: Os autores

Assim, para poder visualizar os resultados de acertos e erros do algoritmo, foi criada uma matriz, comparando dois tipos de dados: os resultados esperados, pertencentes à base de

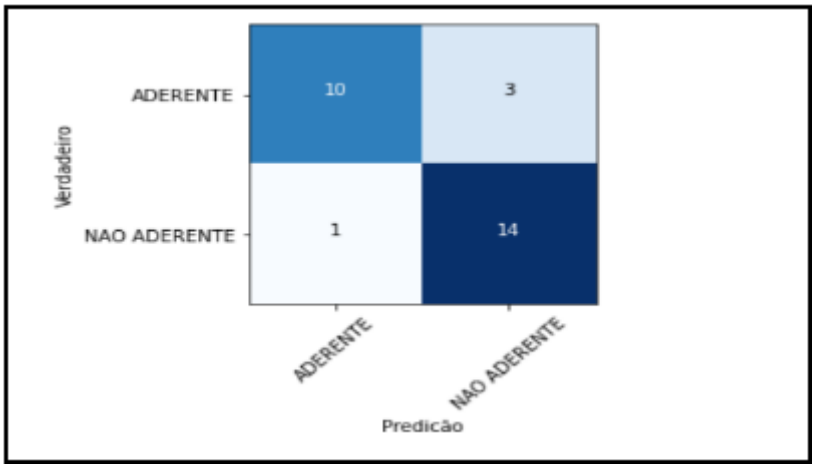
teste, previamente conhecidos, e os resultados previstos, realizados pelo classificador, mostrados na Figura 6, a matriz pode ser observada mais adiante, em que os resultados acertados pelo classificador se encontram entre o sinal '<>' na Figura 7.

Figura 6 - Criação da Matriz de Confusão

```
1  from nltk.metrics import ConfusionMatrix
2  esperado = []
3  previsto = []
4  for (frase, classe) in testesdepressivos:
5          resultado = classificador.classify(frase)
6          previsto.append(resultado)
7          esperado.append(classe)
8
9  matriz = ConfusionMatrix(esperado, previsto)
10 print(matriz)
```

Fonte: Os autores

Figura 7 - Matriz de Confusão do Naive Bayes



Fonte: Os autores

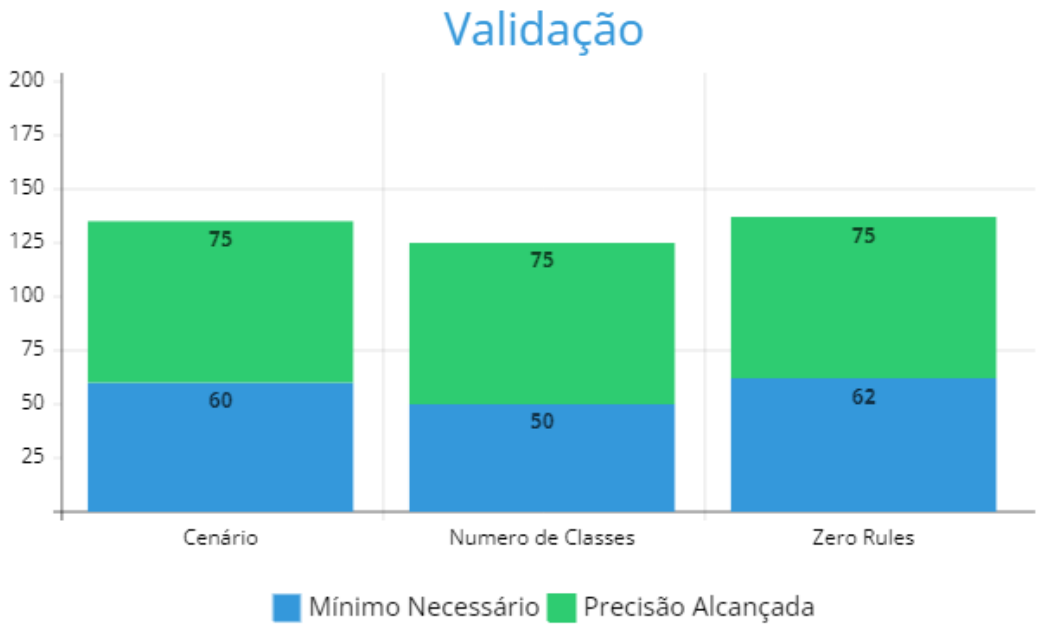
Figura 8 - Matriz de erros e acertos (classificador)

```
             |  d    |
             |  e    |
             |  p    |
             |  r    |
             |  e  n |
             |  s  e |
             |  s  u |
             |  i  t |
             |  v  r |
             |  o  o |
-------------+-------+
depressivo |<30>15  |
    neutro |  15<60>|
-------------+-------+
(row = reference; col = test)
```

Fonte: Os autores

O algoritmo treina a base de acordo com o exemplo abaixo de aprendizagem supervisionada, no qual são extraídas todas as palavras e classes da base, relacionada cada palavra da frase com a respectiva classe, gerando, assim, o classificador. Em seguida, a base de teste é analisada pelo classificador. Com o teste do algoritmo feito, foi realizada uma avaliação para decidir se o algoritmo é utilizável em alguma aplicação ou não, para isso ele deve passar em 3 condições: cenário, número de classes e Zero Rules. Indicadores para a verificação da aplicação do algoritmo naive bayes com performance e qualidade de uso. Na condição de cenário deve-se levar em conta em que situação está sendo implementado o algoritmo. Na próxima condição, deve-se considerar o número de classes, que, no caso, foram 2 (depressivo e neutro), nas quais são divididas por 100%, assim, o algoritmo precisaria possuir uma porcentagem de acerto maior que 50%, caso contrário, se apresentaria irrelevante, pois um gerador de resultados randômicos teria mais credibilidade do que a inteligência. E, na última condição, o Zero Rules, foi obtido o número da quantidade de frases da maior classe (neutro) e dividido pelo número total de frases da base de dados, multiplicando em seguida por 100, para obter a porcentagem. O algoritmo deveria apresentar um percentual acima de 62,5% de acerto para ser validado nessa condição, fato que também ocorreu. Nesse caso, se a inteligência não ultrapassasse o percentual de Zero Rules, o resultado da análise sempre seria a da classe de maior volume de dados, o que pode ser observado na Figura 8:

Figura 8 - Comparando condições e validação



Fonte: Os autores

6 CONSIDERAÇÕES FINAIS

A base de dados é o fator mais importante para o treinamento da machine learning, quanto maior a base e o mais amplo possível o vocabulário das frases na fase de treinamento, maior será a acurácia do classificador. A melhor possibilidade possível para aumentar a acurácia do classificador seria o próprio classificador acrescentar a sua própria base de dados novas frases para análise, sendo assim sempre expandindo o vocabulário da sua base, afetando diretamente a precisão dos acertos. A utilizar o algoritmo inteligente Naive Bayes, em todas as etapas e foi utilizado a biblioteca NLTK do Python, na parte da preparação da

base, retirando palavras que não agregam valor e na extração dos radicais, a biblioteca foi satisfatória na utilização do algoritmo, trazendo muitas funções que facilitam todo o processo. Algo que se tornou um tanto quanto trabalhoso foi a classificação da base de dados para o treinamento da inteligência, por ter sido toda feita manualmente, etapa essa que a biblioteca não apresentou ferramentas que auxiliassem no procedimento. Vale ressaltar, também, que existem alguns classificadores dessa espécie, porém em sua maioria se encontram na língua inglesa, portanto, escasso na língua portuguesa, sendo um diferencial a mais para o algoritmo inteligente do presente estudo. Com uma precisão maior da base de dados em reconhecimento de publicações depressivas em redes sociais, é possível prevenir suicídios e elaborar planos de tratamentos sociais para as pessoas que estiverem publicando textos depressivos com frequência, direcionando coisas mais positivas para a rede social dessa pessoa para que ela valorize mais a própria vida, ou ainda profissionais que possuem conhecimento de como agir em situações desse tipo, tenham a oportunidade de conversar com essa pessoa por meio da sua rede social e demonstrar que pessoas se importam com ela.

É sugerido para trabalhos futuros que o próprio classificador pudesse alimentar a sua base de dados com novas frases, classificadas por ele mesmo, para melhorar a precisão de acerto.

7 REFERÊNCIAS

ARANHA, Christian; PASSOS, Emmanuel. **A Tecnologia de Mineração de Textos.** 2006. Disponível em: <http://189.16.45.2/ojs/index.php/reinfo/article/view/171>. Acesso em 16 jun. 2022.

ALETHEIA, **Depressão numa contextualização contemporânea**, 2022. Disponível em: http://pepsic.bvsalud.org/scielo.php?script=sci_arttext&pid=S1413-03942006000300012 Acesso em 17 out .2022

IPQ. **Depressão**, 2022. Disponível em: https://bvsms.saude.gov.br/depressao-4/ Acesso em 17 out .2022

FLAVIO, **Técnicas de Processamento de Linguagem Natural Aplicadas à Gestão de Serviços de TI**, 2022. Disponível em: https://acervodigital.ufpr.br/bitstream/handle/1884/71061/R%20-%20E%20-%20FLAVIO%20AUGUSTO%20WEBER.pdf?sequence=1&isAllowed=y Acesso em 07 out. 2022

GRANATYR, Jones**. Mineração de Emoção em Textos com Python e NLTK**. Udemy, Disponível em: https://www.udemy.com/course/mineracao-de-emocao-em-textos-com-python-e-nltk/ Acesso em 16 jun. 2022

GRAHAM, Sarah et al. Artificial intelligence for mental health and mental illnesses: an overview. **Current psychiatry reports**, v. 21, n. 11, p. 1-18, 2019.

KHAN, Nighat Z.; JAVED, Muhammad Ali. **Use of artificial intelligence-based strategies for assessing suicidal behavior and mental illnes**s: a literature review. Cureus, v. 14, n. 7, 2022.

LOPES, Claudia de Souza. **Como está a saúde mental dos brasileiros? A importância dos cortes de nascimento para melhor compreensão do problema**. Cadernos de Saúde Pública, v. 36, 2020.

LUDERMIR, Teresa Bernarda. **Inteligência Artificial e Aprendizado de Máquina**: estado atual e tendências. Estudos Avançados, v. 35, p. 85-94, 2021.

LAMPROPOULOS, Aristomenis S.; TSIHRINTZIS, George A. **Machine Learning Paradigms: Applications in Recommender Systems**. Springer, 2015

MACHADO, Aydano P. et al. **Mineração de Texto em Redes Sociais Aplicada à Educação a Distância**
Disponível em: <http://pead.ucpel.tche.br/revistas/index.php/colabora/article/view/132>. Acesso em 19 out 2022

MICROSOFT, **Conceitos de mineração de dados.Microsoft Learn**, 2022. Disponível em: https://learn.microsoft.com/pt-br/analysis-services/data-mining/data-mining-concepts?view=asallproducts-allversions Acesso em 07 out. 2022

MCCULLOCH, Warren S. PITTS, Walter. **Uma Breve História das Redes Neurais Artificiais**. Disponível em: https://www.deeplearningbook.com.br/uma-breve-historia-das-redes-neurais-artificiais/ Acesso em 17 out. 2022

NEMESURE, Matthew D. et al. **Predictive modeling of depression and anxiety using electronic health records and a novel machine learning approach with artificial intelligence.** Scientific reports, v. 11, n. 1, p. 1-9, 2021. Disponível em: https://www.nature.com/articles/s41598-021-81368-4 Acesso em 17 out. 2022

Pierro, Bruno de Pierro, **O mundo mediado por algoritmos**
Disponível em: https://revistapesquisa.fapesp.br/o-mundo-mediado-por-algoritmos/ Acesso em 11 out. 2022

PRATES, Wlademir. **Introdução ao Processamento de Linguagem Natural (NLP)**, 2022. Disponível em: https://cienciaenegocios.com/processamento-de-linguagem-natural-nlp/ Acesso em 07 out. 2022

RITTER, Samuel et al. **Cognitive psychology for deep neural networks**: A shape bias case study. In: International conference on machine learning. PMLR, 2017. p. 2940-2949.

DUQUE, José Walmir Gonçalves; RAYMUNDO, Abner Lucas; NETO, Pedro Ferreira. Uma aplicação de big data para classificação de sentenças depressivas do twitter. **Revista H-TEC Humanidades e Tecnologia**, v. 2, n. 1, p. 82-95, 2018. Disponível em: https://revista.fateccruzeiro.edu.br/index.php/htec/article/view/74 Acesso em 10 out 2022

SBIC. **Detecção de perfis sintomáticos de depressão no Twitter utilizando aprendizado de máquina, 2022.** Disponível em: https://sbic.org.br/wp-content/uploads/2021/09/pdf/CBIC_2021_paper_17.pdf

SERAPIÃO, Paulo Roberto Barbosa et al. **Uso de mineração de texto como ferramenta de avaliação da qualidade informacional em laudos eletrônicos de mamografia**. Disponível em: http://www.scielo.br/pdf/rb/v43n2/a10v43n2.pdf Acesso em 22 out. 2022.