UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

ROBERTA SOUSA PIRES

Análise de Depressão em *Tweets*

Goiânia
2020

UNIVERSIDADE FEDERAL DE GOIÁS

INSTITUTO DE INFORMÁTICA

AUTORIZAÇÃO PARA PUBLICAÇÃO DE TRABALHO DE CONCLUSÃO DE CURSO EM FORMATO ELETRÔNICO

Na qualidade de titular dos direitos de autor, **AUTORIZO** o Instituto de Informática da Universidade Federal de Goiás – UFG a reproduzir, inclusive em outro formato ou mídia e através de armazenamento permanente ou temporário, bem como a publicar na rede mundial de computadores (*Internet*) e na biblioteca virtual da UFG, entendendo-se os termos "reproduzir" e "publicar" conforme definições dos incisos VI e I, respectivamente, do artigo 5º da Lei nº 9610/98 de 10/02/1998, a obra abaixo especificada, sem que me seja devido pagamento a título de direitos autorais, desde que a reprodução e/ou publicação tenham a finalidade exclusiva de uso por quem a consulta, e a título de divulgação da produção acadêmica gerada pela Universidade, a partir desta data.

Título: Análise de Depressão em *Tweets*

Autor(a): Roberta Sousa Pires

Goiânia, 11 de Agosto de 2020.

Roberta Sousa Pires – Autor

Nádia Félix – Orientador

ROBERTA SOUSA PIRES

Análise de Depressão em *Tweets*

Trabalho de Conclusão apresentado à Coordenação do Curso de Ciência da Computação do Instituto de Informática da Universidade Federal de Goiás, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Área de concentração: Ciência da Computação.

Orientador: Prof. Nádia Félix

Goiânia
2020

ROBERTA SOUSA PIRES

Análise de Depressão em *Tweets*

Trabalho de Conclusão apresentado à Coordenação do Curso de Ciência da Computação do Instituto de Informática da Universidade Federal de Goiás como requisito parcial para obtenção do título de Bacharel em Ciência da Computação, aprovada em 11 de Agosto de 2020, pela Banca Examinadora constituída pelos professores:

Prof. Nádia Félix
Instituto de Informática – UFG
Presidente da Banca

Prof. Dr. Thierson Couto Rosa
Instituto de Informática – UFG

Roberta Sousa Pires

Iniciou a graduação na Universidade Federal de Goiás no segundo semestre de 2015 no curso de Ciências da Computação. Durante o andamento do curso foi estagiária do Centro Cultural - UFG por um ano (2018-2019) atuando na parte de desenvolvimento e manutenção do *website* informativo do local e auxiliando em serviços de suporte técnico e administrativos.

À minha mãe que sempre me incentivou e apoiou durante toda a graduação.

Agradecimentos

Agradeço, primeiramente, à Deus por me permitir chegar até aqui e por todas as pessoas e recursos que Ele colocou em meu caminho para que me ajudassem durante a realização deste trabalho. Entre essas pessoas, meus pais que se dedicaram tanto para que eu tivesse uma educação de qualidade e que sempre me apoiaram e incentivaram. À minha irmã Fernanda, namorado Daniel Lucas e amigos Deborah, Jefferson e Jucicleia pelo apoio e por tornarem meus dias mais leves e fáceis de continuar com disposição e alegria. Aos meus avós e tios que sempre acreditaram e se dispuseram a me ajudar quando foi preciso. Aos meus amigos e colegas de curso que me ajudaram e passaram por momentos especiais comigo.

Não basta ensinar ao homem uma especialidade. Porque se tornará uma máquina utilizável, mas não uma personalidade. É necessário que adquira um sentimento, um senso prático daquilo que vale a pena ser empreendido, daquilo que é belo, do que é moralmente correto. A não ser assim, ele se assemelhará, com seus conhecimentos profissionais, mais a um cão ensinado do que a uma criatura harmoniosamente desenvolvida. Deve aprender a compreender as motivações dos homens, suas quimeras e suas angústias para determinar com exatidão seu lugar exato em relação a seus próximos e à comunidade.

Albert Einsten,
Como Vejo o Mundo (1953), página 29.

Resumo

Pires, Roberta Sousa. **Análise de Depressão em *Tweets***. Goiânia, 2020. 68p. Relatório de Graduação. Instituto de Informática, Universidade Federal de Goiás.

O presente trabalho teve como objetivo construir um classificador através da aprendizagem de máquina para trabalhar com dados que contenham indícios de depressão e dados que não contenham tais indícios. Para obter um classificador de bom desempenho, estudou-se modelos, bem como algoritmos usados para classificar os dados. Para representar os textos de modo estruturado, utilizou-se o *bag of words*. Os algoritmos usados foram K-vizinhos mais próximos, Naive Bayes, *Support Vector Machine*, Árvore de Decisão, Random Forest e Perceptron. Algumas métricas bem conhecidas como a acurácia, o *recall*, F1 *score* e a precisão foram utilizadas para medir o desempenho de cada um dos algoritmos. Gráficos de matriz de confusão e curva de aprendizagem também foram utilizados para ilustração e melhor compreensão do problema e seus resultados. Para tais finalidades, foram utilizadas bibliotecas da linguagem Python que facilitaram a manipulação dos dados e a classificação destes por algoritmos já previamente implementados. Por fim, o melhor desempenho obtido foi através do classificador *Support Vector Machine*, sendo que este apresentou uma acurácia de 88% na classificação dos dados.

Palavras–chave

Processamento de Linguagem Natural, Classificação, Aprendizagem de Máquina Supervisionada

Abstract

Pires, Roberta Sousa. **Depression Analysis in Tweets**. Goiânia, 2020. 68p. Relatório de Graduação. Instituto de Informática, Universidade Federal de Goiás.

The present work aimed to build a classifier through machine learning to work with data that contain evidence of depression and data that do not contain such evidence. To obtain a good performance classifier, implementation models were studied, as well as algorithms used to classify the data. To display texts in a structured way, we use bag of words model. The algorithms used were K-Nearest Neighbors, Naive Bayes, Support Vector Machines, Decision Tree, Random Forest and Perceptron. Some well-known metrics such as accuracy, recall, F1 score and precision were used to measure the performance of each algorithm. Confusion matrix and learning curve graphs were also used to illustrate and better understand the problem and its results. For such purposes, Python libraries were used to facilitate data manipulation and data classification by previously implemented algorithms. Finally, the best performance obtained was through the Support Vector Machine classifier, which showed an accuracy of 88% in the data classification.

Keywords

Natural language processing, Classification, Supervised Machine Learning

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviações

AD Árvore de Decisão

API Interface de Programação de Aplicações

BoW Bag of Words

CBoW Bag of Words Contínuo

FN Falso Negativo

FP Falso Positivo

KNN K vizinhos mais próximos

LDA Alocação de Diretórios Latentes

LIWC Investigação Linguística e Contagem de Palavras

LR Regressão Logística

LSVC Classificação Vetorial de Suporte Linear

MDL Aprendizagem de Dicionário Depressivo Multimodal

MSNL Aprendizagem Múltipla de Redes Sociais

NB *Naive Bayes*

OMS Organização Mundial de Saúde

PTSD Estresse Pós Traumático

RNC Redes Neurais Convolutivas

RNR Redes Neurais Recorrentes

SVM *Support Vector Machine*

TFIDF Frequência do Termo–Inverso da Frequência nos Documentos

VN Verdadeiro Negativo

VP Verdadeiro Positivo

WDL Aprendizagem de Dicionário Wasserstein

CAPÍTULO 1

Introdução

Depressão é uma doença mental comum. Segundo a OMS (Organização Mundial de Saúde), estima-se que há mais de 300 milhões de pessoas no mundo que sofrem com a doença e que essa é a principal causa de incapacidade mundial. Ainda segundo a OMS, mais de 800 mil pessoas morrem todo ano devido ao suicídio, ação que pode ser, em alguns casos, responsiva à depressão. Sabe-se também que o suicídio foi a segunda maior causa de morte entre jovens de 15 a 19 anos no mundo em 2015 [Organization 2017].

Embora seja uma doença tratável, menos da metade das pessoas que sofrem deste mal recebem o tratamento adequado. Muitas delas nem reconhecem os sintomas da depressão, tais como: tristeza, perda de energia, mudança no apetite, aumento ou redução do sono, ansiedade, culpas ou desesperança, entre outros [OPAS 2018].

Ainda segundo a Organização Mundial de Saúde, é importante medir quantas pessoas morrem e por quê elas morrem todos os anos. Isso, além de avaliar a eficácia do sistema de saúde de um país, avalia também como as doenças estão afetando as pessoas. De posse das estatísticas de doenças, autoridades podem determinar o foco de ações de saúde pública. Assim, pode-se destinar mais recursos para políticas que ajude no combate ou na prevenção das mesmas [World Health Organization 2018].

Sabe-se ainda que o mundo digital está evoluindo continuamente ao redor do mundo. Cerca de 3,1 bilhões de pessoas no mundo são usuárias ativas de redes sociais [We Are Social 2018]. Dentre essas, o Twitter é uma das mais populares.

As redes sociais podem ser consideradas por muitas pessoas como um lugar de entretenimento, informação, notícias, para dar opiniões pessoais e também falar sobre sentimentos. Pensando nisso, pode-se analisar perfis em redes sociais para detectar casos de depressão em usuários e, assim, identificar possíveis eventos da doença.

Diante desta possibilidade, o presente estudo foi feito utilizando a Inteligência Artificial para trabalhar com dados coletados do Twitter e aplicar o método de Aprendizagem de Máquina com Processamento de Linguagem Natural para analisar os mesmos. Com esses recursos, pode-se predizer se um *tweet* tem indícios depressivos.

Neste trabalho, utiliza-se dados já coletados do Twitter [1]. Os *tweets* (mensagens publicadas no Twitter que podem conter textos) publicados por alguns usuários foram selecionados para fazer o treinamento e teste do modelo desenvolvido.

1.1 Objetivos

1.1.1 Objetivo Geral

O objetivo principal deste trabalho é identificar indícios de depressão em dados do Twitter utilizando processamento de linguagem natural e aprendizagem de máquina supervisionada.

1.1.2 Objetivos Específicos

Obter um conjunto de dados que contenha exemplos de *tweets* com indícios de depressão e outros *tweets* que não possuam este sinal. Tratar os dados da melhor forma para eliminar partes consideradas ruins e que possam atrapalhar os resultados finais para que esses sejam bem utilizados no momento de classificação e permitam uma boa predição dos dados. Extrair características de cada texto do conjunto de dados e utilizar diferentes algoritmos para o processo de classificação e, por fim, analisar os resultados obtidos e compará-los.

1.2 Contribuições

O estudo dos trabalhos anteriores realizado e apresentado nesta pesquisa contribui para uma revisão sistemática para o assunto em questão. Além disso, este presente estudo pode favorecer o estado da arte atual, uma vez que obteve resultados satisfatórios. Todos os processos realizados no conjunto de dados, desde a limpeza, classificação e valores métricos para avaliar os resultados obtidos, estão disponíveis em um repositório no GitHub [2] para visualização de como foram feitos os processos descritos no trabalho. Isto pode contribuir para a replicação e melhoramento da pesquisa.

[1] http://twitter.com
[2] https://github.com/robertasousap/DeteccaodeDepressaoTweets

CAPÍTULO 2

Fundamentação Teórica

A Inteligência Artificial [Russell e Norvig 2009] é um ramo da ciência da computação que busca expressar, em máquinas, habilidades similares às habilidades humanas. Esse ramo vem se desenvolvendo ao longo dos anos e contribuindo para melhores soluções de problemas reais. A aprendizagem de máquina (conhecida por *Machine Learning*) [Mitchell 1997] é uma alternativa para chegar em uma Inteligência Artificial. A aprendizagem de máquina é um método que pratica o uso de algoritmos para aprendizado, predição e análise de dados sobre algum assunto. É conhecido, no mundo de *Machine Learning*, dois tipos de aprendizagem: aprendizagem não supervisionada e aprendizagem supervisionada. Esta última, foi utilizada no presente trabalho.

2.1 Aprendizagem Supervisionada

Esse método utiliza dados prévios para ensinar à máquina padrões, ou seja, a máquina é treinada a partir de dados conhecidos [Goodfellow, Bengio e Courville 2016]. Assim, pode-se tomar decisões mais precisas quando novos dados são recebidos. Para tal fim, um conjunto de dados deve ser utilizado como entrada para um algoritmo. Esse conjunto deve ser separado em uma parte para treino e outra para teste. No momento de treino, são conhecidas as verdadeiras respostas e a aprendizagem supervisionada trata de predizer e corrigir as respostas incorretas para que o algoritmo aprenda e lide melhor com dados futuros (dados de teste). Quando o algoritmo termina o treino, é feita a etapa de teste, onde os verdadeiros rótulos são escondidos durante o processo. O algoritmo, então, faz predições baseadas no aprendizado adquirido com os dados de treino. Depois disso, faz-se uma comparação entre os dados preditos e as verdadeiras respostas para que haja uma análise do comportamento do algoritmo para a tarefa.

Na aprendizagem supervisionada, tem-se variáveis de entrada e essas variáveis são mapeadas em uma função para se obter uma saída [Silveira e Bullock]. O objetivo, nesse ponto de vista, é ajustar muito bem a função de mapeamento para que quando forem recebidos novos dados, a saída seja produzida de forma correta.

Problemas de aprendizagem supervisionada podem ser separadas em dois grupos diferentes: regressão e classificação.

2.1.1 Regressão

Um problema de regressão ocorre quando a saída da função é um valor contínuo. Como, por exemplo, predizer o valor de algum produto, predizer altura ou peso de determinada coisa. Nesse tipo de problema, o conjunto de dados pode ter rótulos diferentes para cada entrada do conjunto e o algoritmo pode dar como saída para algum novo dado, um valor que não esteja presente nesse grupo de rótulos que foi colocado como parâmetro para a máquina.

Um bom exemplo de aprendizagem de máquina com a abordagem de regressão é o algoritmo chamado de Regressão Linear [Berger, Pietra e Pietra 1996] que pode ser considerado um dos algoritmos mais simples de aprendizagem de máquina.

Dispõe-se de variáveis (atributos), consideradas independentes, para calcular sobre os dados com seus rótulos. O objetivo nesse modelo de regressão é encontrar um conjunto de pesos ótimo para esses atributos de acordo com alguma função de custo. A forma mais usual do método de regressão linear é com a abordagem dos mínimos quadrados. Tal abordagem procura encontrar o melhor ajuste para um conjunto de dados e tenta minimizar a soma dos quadrados das diferenças entre o valor estimado e os dados observados.

2.1.2 Classificação

Um problema de classificação ocorre quando a saída corresponde a um valor não-contínuo [Pacheco e Pereira 2018]. Por exemplo, predizer se um certo dado pertence à uma classe ou não. Essa resposta sempre será binária e o valor de saída será correspondente a um dos valores existentes nos rótulos dos dados de entrada. Alguns algoritmos/modelos de classificação são apresentados a seguir.

K vizinhos mais próximos [Cunningham e Delany 2007]

Nesse algoritmo, conhecido como KNN (sigla referente à *K-Nearest Neighbors*), a função é aproximada localmente e todo o cálculo é adiado até a etapa de classificação. A classificação é baseada na maioria dos votos dos k vizinhos mais próximos. Isso quer dizer que o algoritmo é sensível à estrutura local dos dados [Dasarathy 1991]. Uma técnica que pode ser usada para melhorar o método é fazer a ponderação entres os vizinhos. Assim, o vizinho mais próximo pode contribuir mais com a média do que um vizinho um pouco mais distante entre os k escolhidos e o rótulo que é mais frequente entre os

vizinhos é atribuído ao dado referente. Outra técnica para ajudar em classificação (este algoritmo pode ser usado para regressão também), pode ser escolher um k de número ímpar para que não haja empate nos votos dos vizinhos. Uma desvantagem desse modelo é o desequilíbrio entre as classes do conjunto, se houver mais dados pertencentes à uma classe do que outras, esta tenderá a dominar a previsão do novo exemplo. Porém, esse problema pode ser contornado usando técnicas de ponderação de classificação.

Naive Bayes [Friedman, Geiger e Goldszmidt 1997]

O *Naive Bayes*, é um algoritmo usado para classificação. O teorema de Bayes [Bayes 1763], usado para provar a existência de Deus, foi usado como base para o classificador *Naive Bayes*. Esse classificador considera os atributos independentes, ou seja, ele não considera a correlação entre duas ou mais variáveis. É um algoritmo considerado simples, rápido e de boa eficiência para problemas de classificação de textos ou previsões em tempo real. A ideia básica do classificador é prever uma classe para determinado dado, observando o histórico de outros dados. O *Naive Bayes* é representado por probabilidades e essas probabilidades serão usadas para fazer previsões em novos dados. Essas são: a probabilidade de cada classe no conjunto de treino e a probabilidade condicional de cada atributo dado a classe. O Teorema de Bayes é dado pelo seguinte:

$$P(h|o) = \frac{P(o|h) \cdot P(h)}{P(o)}$$

onde P(h|o) é a probabilidade da hipótese "h"dado a observação "o". Isso é conhecido como probabilidade posterior (o que se quer encontrar). P(o|h) é a probabilidade da observação "o"dado que a hipótese "h"é verdadeira. P(h) é a probabilidade da hipótese "h"ser verdadeira (independentemente dos dados). Isso é conhecido como probabilidade anterior de "h". P(o) é a probabilidade da observação "o"(independentemente da hipótese). O objetivo é encontrar "h"dado "o". Então, depois de calcular a probabilidade de várias hipóteses "h"(várias classes), devemos selecionar aquela com a maior probabilidade e associá-la a observação.

Support Vector Machine [JOACHIMS 1999]

Support Vector Machine, mais conhecido como SVM, é um classificador binário e não probabilístico e é considerado uma abordagem geométrica para o problema de classificação. Esse classificador faz um representação dos dados no plano, como se cada exemplo do conjunto fosse um ponto no espaço [Cortes e Vapnik 1995]. Esses pontos são mapeados de maneira que os exemplos de cada classe sejam divididos por um espaço claro, ou seja, que separe as classes de forma bem clara, onde um dado não se confunda

entre os espaços de cada uma das classificações. Um novo dado é mapeado no mesmo espaço e predito como pertencente a uma classe baseado em qual lado do espaço ele foi colocado. A Figura 2.1 representa um plano com duas classes diferentes. Os vetores de suporte são os que estão mais próximos à margem e destacados (linha pontilhada). A linha de separação (linha contínua presente entre as duas linhas pontilhadas) das classes é tal que tem a maior distância dos pontos de dados mais próximos de qualquer classe. As margens, idealmente, tem que ser maximizadas para diminuir o erro de generalização do classificador.

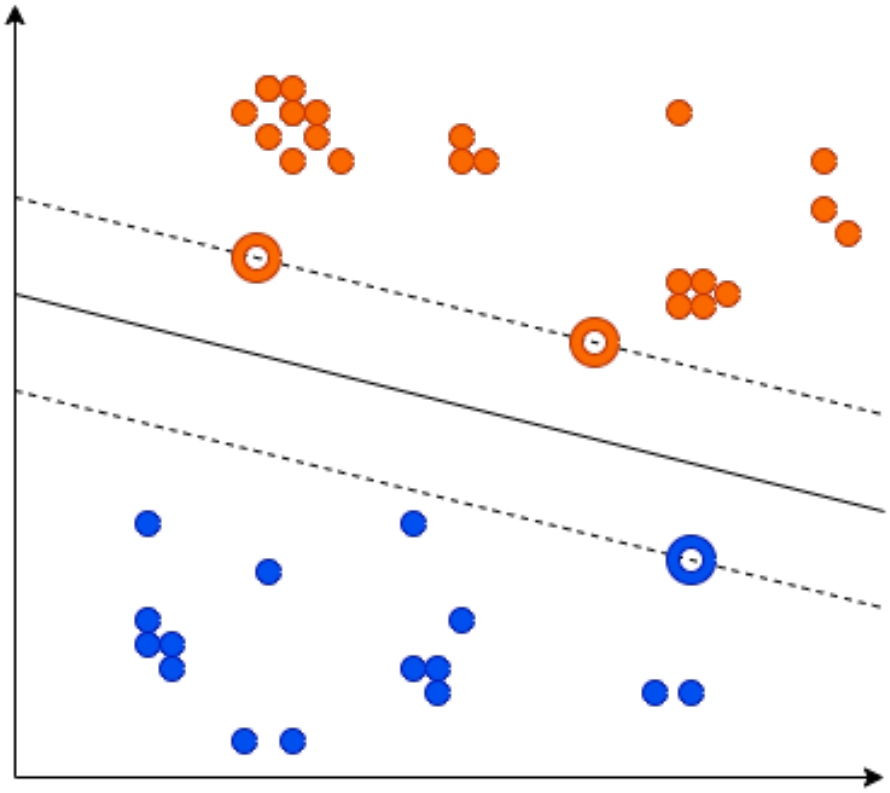

Figura 2.1: Plano SVM - Imagem Adaptada [Cortes e Vapnik 1995]

Árvores de Decisão [Quinlan 1986]

A árvore de decisão relaciona o conjunto de atributos e o rótulo da classe dos dados de entrada. A estrutura da árvore consiste em nós e arestas direcionadas [Bento 2017]. Árvore de decisão é um modelo baseado em regras de decisão que podem ser expressas em linguagem natural e pode facilitar a interpretação da análise dos dados. O funcionamento acontece percorrendo a árvore desde a raiz até um nó-folha, que corresponde à uma classe do conjunto de dados. Seguindo estes passos, passa-se pelos nós internos da árvores, onde se tem uma "pergunta"(regra de decisão) e a resposta decidirá, então, o novo caminho a seguir (aresta). Faz-se isso até chegar um nó que irá prever a classe do dado. A Figura 2.2 representa uma parte da árvore de decisão sobre um conjunto

de dados contendo exemplos de flores. Na raiz, a pergunta é sobre o comprimento da pétala, caso ela for menor ou igual a 2.45cm, o caminho percorrido é o da esquerda que representa "verdadeiro" à regra de decisão. Se ela for maior, percorre-se o caminho da aresta da direita, que corresponde à "falso". Esse novo nó representa uma outra regra de decisão. E assim, percorre-se toda a árvore até chegar a um nó folha correspondente à uma classe do conjunto.

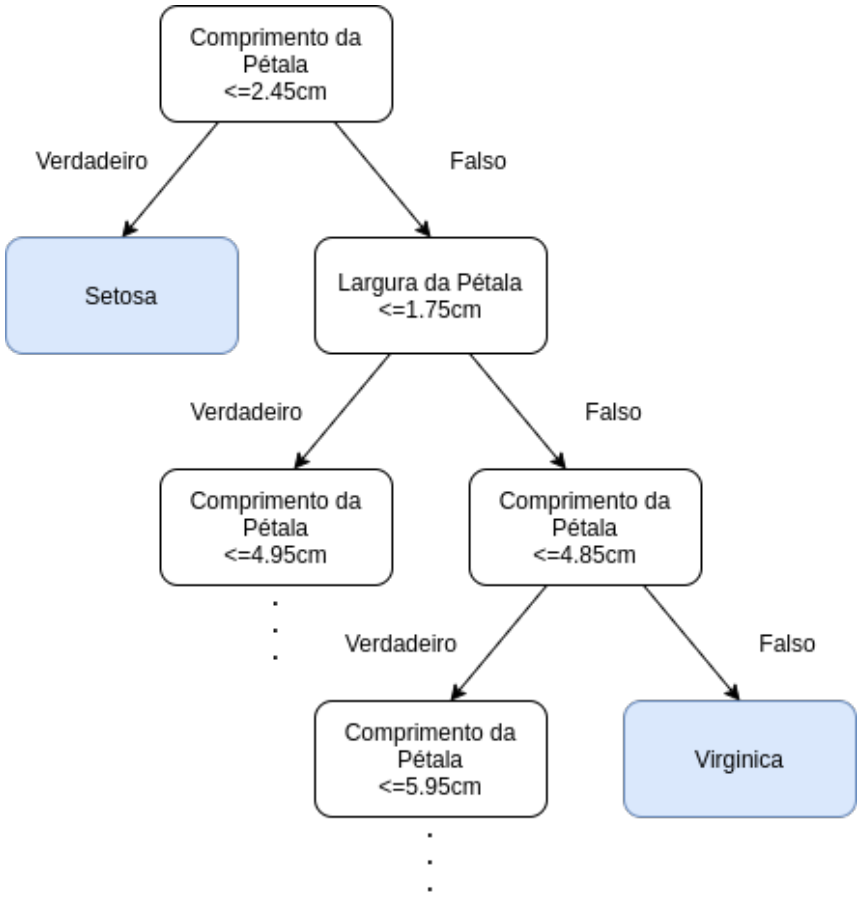

Figura 2.2: Árvore de Decisão - Imagem Adaptada [Pedregosa et al. 2011]

Perceptron [Rosenblatt 1960]:

Perceptron é uma rede neural bem simples, mas que em certos problemas consegue obter bons resultados. Para compreender melhor, é necessário entender o que é uma rede neural. Essa pode ser conceituada como uma estrutura complexa interligada por neurônios, que são elementos de processamento simples que podem operar paralelamente. Essa técnica é baseada em uma rede neural biológica, onde há comunicação (por meio de sinapses) entre as células neurais, através de seus ramos: os dendritos, que recebem e repassam informações, e o axônio, que faz esse repasse [Ferneda 2006]. O modelo de Rosenblatt baseia-se nesse conceito. O Perceptron é um algoritmo de classificação binária [Freund e Schapire 1998]. O modelo é de camada única, recebe várias entradas e

produz uma única saída binária. Na Figura 2.3, vê se um modelo Perceptron recebendo x1, x2, x3, x4 e x5 como entrada. Essas entradas terão w1, w2, w3, w4 e w5 como pesos, respectivamente, que expressam a importância de tais entradas. O valor de saída, 0 ou 1, depende da soma ponderada dos pesos multiplicados pelas entradas. Se este, comparado ao limiar escolhido (número real constante), for maior, então a saída é 1. Caso contrário, a saída é 0.

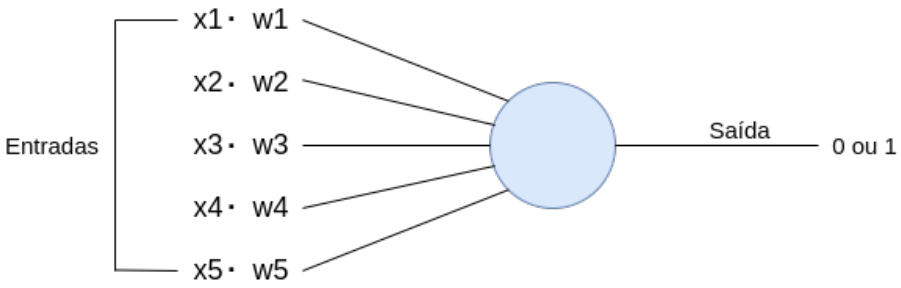

Figura 2.3: Modelo Perceptron - Imagem Adaptada [Saporito 2019]

2.2 Pré-Processamento e Limpeza de Dados

Outro aspecto importante no processo de solução do problema é a forma como os dados são tratados antes de serem utilizados como entrada para o classificador. Essa escolha pode refletir em resultados e tempo computacional. Por exemplo, têm-se um conjunto de dados com muitos ruídos, que são dados que não favorecem o padrão da classificação, se esses forem retirados ou tratados de alguma forma, poderão contribuir para uma predição melhor de novos dados, pois não atrapalham o aprendizado do algoritmo. Pode-se afirmar também que se estes dados ruidosos forem retirados, não será necessário computá-los. Dessa forma, o tempo computacional também pode ser ligeiramente menor.

A limpeza de dados não é, necessariamente, a mesma para todos os problemas. Há problemas em que certos aspectos são considerados ruídos e em outros problemas, não.

2.2.1 *Stop Words*

As *stop words* são palavras que não adicionam significado ao texto, pois são usadas para dar coesão e contexto, porém, se olhadas separadamente, não têm sentido específico [Leskovec, Rajaraman e Ullman 2014]. As *stop words* são, geralmente, palavras comuns no idioma. Na biblioteca NLTK (*Natural Language Toolkit*) [Loper e Bird 2002] do *Python*, existem algumas *stop words* já pré-definidas para utilização no processamento

de linguagem natural, porém essas palavras podem ser alteradas e definidas conforme a necessidade do problema. Para o inglês tem-se o padrão das palavras relacionadas na Tabela 2.1.

Tabela 2.1: *Stop Words* em Inglês

i	me	my	myself	we	our	ours	ourselves
you	your	yours	yourself	yourselves	he	him	his
himself	she	her	hers	herself	it	its	itself
they	them	their	theirs	themselves	what	wich	who
whom	this	that	these	those	am	is	are
was	were	be	been	being	have	has	had
having	do	does	did	doing	a	an	the
and	but	if	or	because	as	until	while
of	at	by	for	with	about	against	between
into	through	during	before	after	above	below	to
from	up	down	in	out	on	off	over
under	again	further	then	once	here	there	when
where	why	how	all	any	both	each	few
more	most	other	some	such	no	nor	not
only	own	same	so	than	too	very	s
t	can	will	just	don	should	now	

A remoção de *stop words* em uma sentença pode ser melhor representada no exemplo a seguir, considerando o conjunto padrão de *stop words* da língua inglesa.

["the depression is the century's disease"]

["depression century' disease"]

2.2.2 Tokenização

A tokenização é feita para obter os termos individuais de um documento [Pyle 1999]. Um documento pode ser considerado cada texto de um conjunto de dados. Então, cada palavra é tida como um *token* em um texto. Considera-se que um espaço em branco separa cada palavra em um documento. Essa é uma das formas de tokenização. Existe também a tokenização por caracteres, por exemplo. Nessa, cada caractere é um *token* do texto. No exemplo abaixo pode-se entender melhor como funciona a tokenização de palavras em um texto:

["A depressão é a doença do século"]

["A", "depressão", "é", "a", "doença", "do", "século"]

2.2.3 Normalização

A normalização de um texto trata-se de converter todas as letras em caixa baixa e retirar acentos e cedilhas [Witten, Frank e Hall 2011]. Esse processo é útil para impedir que palavras iguais sejam tratadas diferentes apenas porque uma está escrita em letras maiúsculas e a outra em minúsculas. Exemplifica-se o processo de normalização:

["A Depressão é a DOENÇA do século"]

["a depressao e a doenca do seculo"]

2.2.4 Stemização

O processo de stemização é o processo de remover os sufixos das palavras para que apenas o radical se mantenha no documento [Manning, Raghavan e Schütze 2008]. Isso é viável porque, via de regra, o radical mantém o significado da palavra e o sufixo apenas concorda com o tempo e pessoa para que a frase tenha o sentido correto e esteja em concordância. Sufixos como "*ing*"e "*ly*"da língua inglesa que correspondem aos sufixos "ando", "endo", "indo"para "*ing*"e "mente"para "*ly*"são retirados no processo de stemização. Um exemplo de redução de algumas palavras para o mesmo radical pode ser visto abaixo:

["**estud**ar", "**estud**ou", "**estud**o", "**estud**ando"]

Todas essas palavras têm o mesmo radical, portanto serão removidos os sufixos e mantidos apenas a parte que é necessário para o significado das mesmas: "**estud**"

O mesmo processo sendo feito em um documento em inglês pode ser exemplificado da seguinte forma:

["***walk****ing*", "***walk****s*", "***walk****ed*"]

A parte mantida das palavras após a stemização, será o radical "***walk***"

2.2.5 Lemmatização

A lematização é semelhante ao processo de stemização. Porém, a lematização converte a palavra em seu significado raiz, em vez de apenas remover os sufixos [Manning, Raghavan e Schütze 2008]. O processo faz uso do vocabulário e faz uma análise morfológica para obter a palavra raiz. A Figura 2.4 ilustra esse processo em três palavras da língua portuguesa, assim como a Figura 2.5 ilustra o processo em três palavras da língua inglesa.

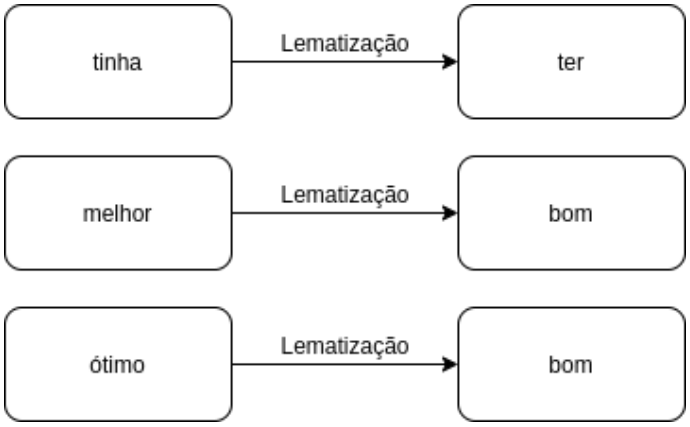

Figura 2.4: Processo de Lematização - Língua Portuguesa

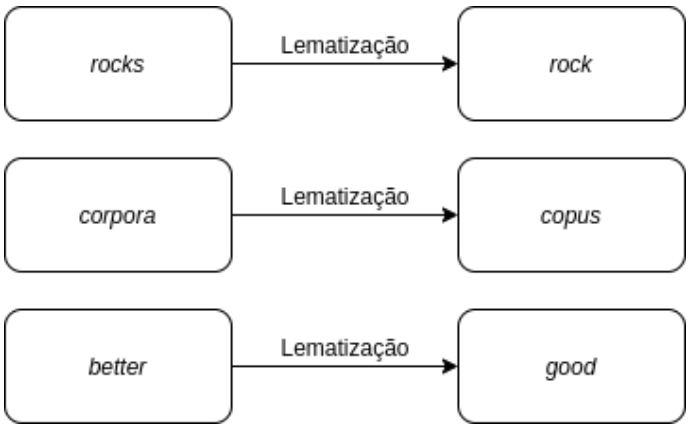

Figura 2.5: Processo de Lematização - Língua Inglesa

Para melhor entendimento dessas duas técnicas no processamento de linguagem natural apresenta-se a Tabela 2.2 que mostra o processo de stemização e lematização em algumas palavras da língua inglesa.

Tabela 2.2: Lematização x Stemização

palavra	lematização	stemização
policy	*policy*	*polici*
dictionaries	*dictionary*	*dictionari*
are	*be*	*ar*
studying	*study*	*studi*
separated	*separete*	*separ*
at	*at*	*at*

2.2.6 Remoção de Palavras Frequentes e Raras

As técnicas de remoção de palavras frequentes pode se justificar na ideia de que se as palavras são muito frequentes, elas podem não ser uma boa variável na

hora da diferenciação das classes, uma vez que a maioria dos documentos as têm [Manning, Raghavan e Schütze 2008].

Da mesma forma, assim como pode-se remover as palavras mais comuns, as palavras que raramente ocorrem também podem sofrer remoção no texto. Por serem tão raras, a associação entre elas e outras palavras é dominada pelo ruído [Hardeniya 2015].

Essas técnicas dependem do problema e há a necessidade de verificação de quais são essas palavras no conjunto de dados, pois cada conjunto é diferente e, por isso, deve ser feito um levantamento.

2.2.7 Remoção de Pontuação

As pontuações em um texto raramente vão acrescentar conteúdo significativo [Witten, Frank e Hall 2011]. Por isso, na maioria dos casos faz-se a remoção das pontuações e também caracteres especiais. Dessa forma, o tamanho dos dados pode ser reduzido e pode haver, também, um ganho computacional.

2.3 Extração de Atributos

Para o processamento de linguagem natural há algumas etapas a serem seguidas. Entre elas, a extração de atributos se faz necessária, pois um classificador, muitas vezes, espera como entrada um vetor de atributos numéricos em vez de documentos de texto bruto. Sabendo que os classificadores não podem ser alimentados diretamente com os dados brutos, deve-se oferecer como entrada os atributos referentes ao conjunto de dados. Esses atributos podem ser escolhidos de acordo com a necessidade do problema e do que se quer como solução [Sammons et al. 2016]. Podem ser atributos, por exemplo, o número de palavras em um documento, número de caracteres, comprimento médio das palavras, quantidade de *stop words* em cada texto, número de caracteres especiais, quantidade de números presentes no texto, número de palavras em letras maiúsculas, entre outros. Uma importante forma de representação da extração de atributos é o *bag of words*.

2.3.1 *Bag of Words*

Esse método, mais conhecido como BoW (sigla para *Bag of Words*), faz o processo de vetorização que, em geral, transforma uma coleção de documentos de texto em vetores numéricos [McTear, Callejas e Griol 2016]. No BoW, especificamente, transforma-se cada *token* do conjunto de dados em um atributo. Esses *tokens*, como descritos anteriormente, podem ser cada uma das palavras presentes no conjunto de dados [Zhang, Jin e Zhou 2010].

É construída uma matriz pelo processo de vetorização do conjunto de dados. Essa vetorização é tal que cada documento do conjunto de dados é representado por uma linha da matriz e cada *token* é uma coluna da matriz. O que preenche o corpo da matriz é a quantidade de vezes que determinado *token* aparece em determinado documento. Sendo assim, para a entrada [0][7] da matriz, por exemplo, encontra-se a quantidade de vezes que a palavra localizada na coluna 7 aparece no texto do documento da linha 0. Para um melhor entendimento, exemplifica-se a seguir.

Sejam os documentos presentes no conjunto de dados:

["Depressão é um problema sério e precisa de sério cuidado",
"Depressão é muitas vezes silenciosa"]

Ao aplicar o processo de vetorização pelo *Bag of Words* tem-se:

	cuidado	de	depressão	muitas	precisa	problema	sileciosa	sério	um	vezes
0	1	1	1	0	1	1	0	2	1	0
1	0	0	1	1	0	0	1	0	0	1

Figura 2.6: *Bag of Words*

Assim como citado, o conteúdo da célula [0][7] da matriz informa que por 2 vezes apareceu a palavra "sério"no documento presente na linha 0: "Depressão é um problema sério e precisa de sério cuidado".

Esse exemplo não conteve nenhum tipo de limpeza dos dados antes da construção do *bag of words*. Os dados brutos foram usados para extrair os atributos necessários.

Pode-se resumir o processo de construção de um *bag of words* sobre um conjunto de dados pelo fluxograma da Figura 2.7.

Figura 2.7: Fluxograma da Construção de um *Bag of Words*

Devem ser lembradas algumas características que apresentam vantagens e outras que apresentam desvantagem do modelo *Bag of Words* [Brownlee 2017]:

Vantagens

- É um modelo bem simples de se construir e de fácil entendimento.
- Oferece uma boa flexibilização para trabalhar com diferentes conjuntos de dados.
- Tem gerado bons resultados com problemas de predição e de classificação de documentos.
- Dependendo do tamanho do conjunto de dados, não requer grande capacidade computacional.

Desvantagens

- Os documentos são descritos por ocorrências de palavras, ignorando completamente as informações de posição relativa às palavras no documento.
- A abordagem básica do *BoW* não considera o significado da palavra no documento. Ignora completamente o contexto em que é usado.
- Por apresentar cada palavra existente no conjunto de dados como um item na coluna da matriz, pode-se afetar o custo computacional (tempo e espaço), uma vez que, se o conjunto for muito grande, haverá muitas colunas e linhas. Isso requer um *design* cuidadoso dos dados, como a limpeza, por exemplo, para que o tamanho da estrutura seja o menor possível.

- Ainda se tratando de uma matriz, pode-se observar que, as células são preenchidas de acordo com a contagem de quantas vezes a palavra aparece em um texto. Isso, consequentemente, gera uma matriz esparsa. Em outras palavras, muitos zeros estarão presentes na representação de um conjunto de dados por *Bag of Words* e isso deve ser considerado.

2.3.2 *N-grams*

O conceito de *n-grams* atua na consideração de contexto em um documento [Jurafsky e Martin 2009]. Por exemplo, um *Bag of Words* sem considerar nenhum tipo de contexto é o que utiliza a técnica de 1-gram. Isto é, observa cada palavra individualmente como um atributo, não inserindo contexto na análise. Porém, se houver necessidade de uma observação além da individualidade de cada palavra, pode-se usar a técnica de *n-grams*, onde "n"representa um número natural que, para ser considerado como análise de contexto, deve ser maior que 1.

N-grams pode ser considerado como uma sequência contígua de n palavras. Assim, cada uma dessas sequência será considerada como atributo do conjunto de dados em uma representação de *bag of words* com parâmetros de *n-grams* [Silge e Robinson 2017]. A Figura demonstra este tipo de abordagem. Tem-se os mesmos documentos do conjunto de dados do exemplo anterior:

["Depressão é um problema sério e precisa de sério cuidado",
"Depressão é muitas vezes silenciosa"]

Se a escolha da construção do *bag of words* for baseada em 1-gram, a matriz construída será exatamente igual à exposta anteriormente na Figura 2.6. Porém se a escolha for de 2-gram, observa-se que as palavras serão juntadas duas a duas e cada uma dessas junções serão tratadas como *tokens* no BoW (Figura 2.8).

	de sério	depressão muitas	depressão um	muitas vezes	precisa de
0	1	0	1	0	1
1	0	1	0	1	0

	problema sério	sério cuidado	sério precisa	um problema	vezes sileciosa
0	1	1	1	1	0
1	0	0	0	0	1

Figura 2.8: *Bag of Words com 2-grams*

Quanto maior o número "n", maior o contexto inserido na análise. Dependendo do problema, inteirar mais contexto à análise pode dar bons resultados, mas em outros, pode se tornar um dificultador. Não necessariamente um *bag of words* terá a análise baseada em apenas junções de duas palavras. Essa técnica pode ser combinada. Por exemplo, analisar as palavras de forma individuais (1-gram), mas também analisar com junções de duas a duas ou ainda, três a três, e assim por diante [Majumder, Mitra e Chaudhuri 2002]. Todas essas escolhas vão depender do problema e do tipo de solução que se deseja. Ainda devem ser observados fatores como quais tipos de dados estão sendo analisados e poder computacional.

2.4 Hiper Parametrização

O ajuste (ou otimização) de hiper parâmetros é feito para conseguir um melhor resultado com os algoritmos escolhidos. Dessa forma, faz-se diferentes combinações entre os possíveis parâmetros dos classificadores e escolhe-se aquele que gera um melhor resultado [Claesen e Moor 2015]. Os parâmetros são ajustados ao modelo conforme seu desempenho.

2.5 Métricas

As métricas são métodos para avaliar o desempenho de um algoritmo no processo de classificação. Cada uma delas mede um aspecto do desempenho e possui uma importância para a análise geral da contribuição do estudo.

2.5.1 Matriz de Confusão

Um dado pode ser classificado corretamente ou não. Sendo assim, ele pode ser dito como pertencente à uma classe e realmente pertencer. É o que chamamos de resultado verdadeiro. Se o dado for classificado erroneamente, esse resultado será chamado de falso. Seja, por exemplo, um problema que possui duas classes: positivo e negativo. Podemos separar as classificações feitas em quatro grupos.

- Verdadeiro Positivo (VP): o dado foi corretamente predito na classe positivo, ou seja, era positivo e foi classificado como positivo.
- Falso Positivo (FP): o dado foi incorretamente predito na classe positivo, ou seja, era negativo e foi classificado como positivo.
- Verdadeiro Negativo (VN): o dado foi corretamente predito na classe negativo, ou seja, era negativo e foi classificado como negativo.

- Falso Negativo (FN): o dado foi incorretamente predito na classe negativo, ou seja, era positivo e foi classificado como negativo.

Esses quatro tipos de resultados sobre um conjunto podem ser dispostos em uma matriz, chamada de matriz de confusão, para que se possa visualizar melhor as informações de quantos dados estão sendo preditos corretos e incorretos em cada classe existente no conjunto de dados [Ting 2010]. A Tabela 2.3 reproduz a matriz de confusão de um conjunto de dados contendo 600 exemplos.

Tabela 2.3: Predições

	Predito Negativo	Predito Positivo
Rótulo Negativo	203 (VN)	111 (FP)
Rótulo Positivo	106 (FN)	180 (VP)

2.5.2 Acurácia

A acurácia é uma métrica bem simples e usual, além de ter fácil interpretação. Essa métrica é, simplesmente, a divisão entre todos os acertos do classificador pelo número total dos dados, ou seja, é a porcentagem que o modelo classifica corretamente [Sammut e Webb 2010]. Para calcular a acurácia, podemos utilizar a fórmula:

$$Acurácia = \frac{VP + VN}{VP + FP + VN + FN}$$

De acordo com o conjunto da Tabela 2.3 tem-se:

$$Acurácia = \frac{180 + 203}{180 + 111 + 203 + 106} = 0.63$$

Apesar de simples, pode ser uma medida enganadora. Se o conjunto de dados não for equilibrado, pode ser que a acurácia não seja uma métrica muito confiável do desempenho do classificador. Suponha um conjunto que tenha 80% dos dados com rótulo positivo. Se a acurácia for de 0.80 não quer dizer que o algoritmo seja bom, quer dizer apenas que há uma classe dominadora no conjunto e que isto está sendo refletido na acurácia. Por isso, é necessário o uso de algumas outras métricas mais completas juntamente com a acurácia para se ter certeza do desempenho do algoritmo.

2.5.3 Precisão

Quantos dos dados preditos como sendo de uma classe específica realmente são pertencentes à essa classe? A precisão de um classificador em relação a uma determinada categoria é uma métrica dada a partir da razão entre os verdadeiros positivos e os totais

preditos para aquela categoria [Ting 2010]. Assim sendo, podemos derivar a precisão da seguinte fórmula:

$$Precisão(P) = \frac{VP}{VP + FP}$$

Do exemplo anterior, tem-se:

$$Precisão(P) = \frac{180}{180 + 111} = 0.61$$

A principal utilização desta métrica é para modelos onde é preciso minimizar os falsos positivos.

2.5.4 *Recall*

A questão respondida pelo *recall* é: "De todos os dados que realmente são de uma certa classe, qual percentual é identificado corretamente pelo modelo?" Com essa resposta podemos saber o quanto o classificador está identificando os dados de uma certa classe corretamente [Powers 2011]. O *recall* de determinada categoria é dado pela razão entre os verdadeiros positivos e os totais reais daquela categoria (VP e FN). O *recall* é bastante útil quando é preciso minimizar os falsos negativos.

$$Recall(P) = \frac{VP}{VP + FN}$$

Do exemplo anterior, tem-se:

$$Recall(P) = \frac{180}{180 + 106} = 0.62$$

2.5.5 F1 *Score*

A medida de F *Score* combina as métricas de *recall* e precisão [Brownlee 2014]. Há algumas variações dessa medida. A mais comum é a F1, onde atribui ao *recall* e à precisão a mesma importância nos cálculos. É feita uma média harmônica entre os resultados das duas métricas anteriores da seguinte forma:

$$F1Score(P) = \frac{2 \times Precisão(P) \times Recall(P)}{Precisão(P) + Recall(P)}$$

Do exemplo anterior, tem-se:

$$F1Score(P) = \frac{2 \times 0.61 \times 0.62}{0.61 + 0.62} = 0.64$$

2.5.6 Validação Cruzada

A validação cruzada é uma técnica para medir o poder de generalização de um classificador [Kohavi 1995]. A validação cruzada possui algumas variações, sendo que a mais comum delas é a chamada *k-fold* [Brownlee 2018] [Schaffer 1993].

Nesse método, o conjunto de dados é divido em k subconjuntos. A cada iteração da validação cruzada, treina-se os k-1 subconjuntos e o teste é realizado sobre o subconjunto restante. Esse processo é realizado k vezes. Os k subconjuntos sempre serão alocados de forma diferente para ter uma melhor precisão, pois se os dados do conjunto de teste forem favoráveis à predições corretas, os resultados serão bons, mas não significa que o classificador consegue gerar os mesmos resultados para todas possibilidades de conjuntos de teste.

Ao final das k iterações, obtém-se a acurácia das predições. Dessa forma, pode-se obter uma medida mais confiável sobre a capacidade de generalização do classificador. O diagrama da Figura 2.9 demonstra esse processo.

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Iteração 1	Treino	Treino	Treino	Treino	Teste
Iteração 2	Treino	Treino	Treino	Teste	Treino
Iteração 3	Treino	Treino	Teste	Treino	Treino
Iteração 4	Treino	Teste	Treino	Treino	Treino
Iteração 5	Teste	Treino	Treino	Treino	Treino

Figura 2.9: Validação Cruzada *K-fold*, k=5 - Imagem Adaptada [Lima e Batista 2018]

2.5.7 Curva de Aprendizagem

Uma curva de aprendizagem é um gráfico que mostra a quantidade de dados no eixo x e o aprendizado ou a melhoria no eixo y. Esses gráficos fornecem uma representação matemática do processo de aprendizagem [Brownlee 2019]. O comportamento do aprendizado pode ser observado conforme o tempo (ou quantidade de amostras) e a curva de aprendizagem nos mostra isso.

Um modelo de aprendizagem de máquina pode ser avaliado no conjunto de dados de treinamento para dar uma ideia de quão bem o modelo está “aprendendo” [Anzanello e Fogliatto 2011]. Também pode ser avaliado em um conjunto de dados de validação de espera que não faz parte do conjunto de dados de treinamento. A avaliação no conjunto de dados de validação fornece uma ideia de quão bem o modelo está generalizando.

Duas curvas podem ser analisadas no comportamento do modelo:

- Curva de aprendizagem de treinamento: calculada a partir do conjunto de dados de treinamento que fornece uma ideia de quão bem o modelo está aprendendo.
- Curva de aprendizagem de teste: calculada a partir de um conjunto de dados de teste que fornece uma ideia de quão bem o modelo está generalizando.

É comum criar curvas de aprendizado duplo para um modelo de aprendizado de máquina nos conjuntos de dados de treinamento e teste. Perante um gráfico de curvas de aprendizagem de um modelo de classificação, pode-se analisar o comportamento do aprendizado e determinar se os dados estão com um bom ajuste ao modelo e assim, predizendo corretamente novos dados. A Figura 2.10 apresenta um exemplo de uma curva de aprendizagem em que a linha vermelha representa o aprendizado da máquina e a linha verde os acertos obtidos na validação dos dados.

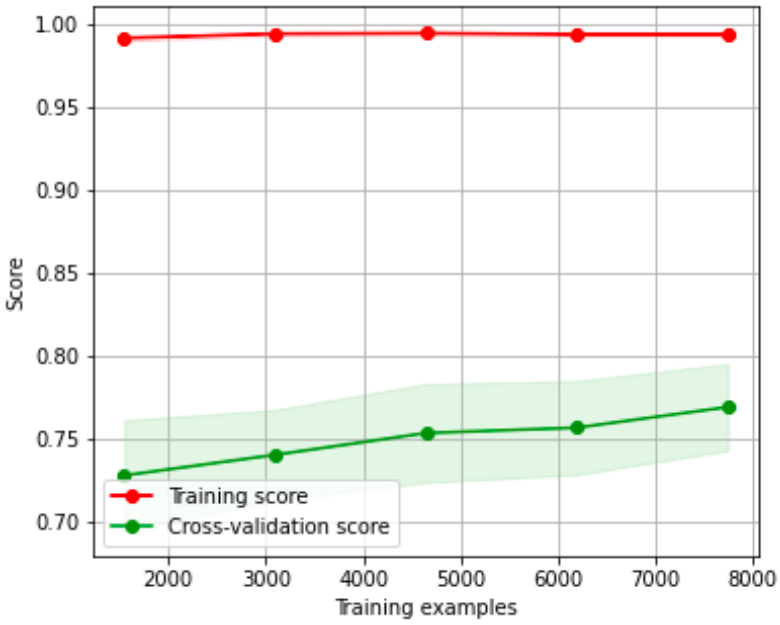

Figura 2.10: Curva de Aprendizagem

CAPÍTULO 3

Trabalhos Relacionados

Durante o processo de revisão sistemática sobre o tema, alguns trabalhos relacionados ao assunto foram estudados para um melhor entendimento do problema e possíveis soluções para tal. Apresenta-se os tópicos principais de cada artigo que contribuiu para o desenvolvimento deste.

A Aprendizagem Profunda, um conhecido ramo da aprendizagem de máquina, foi usado em um dos trabalhos analisados para detecção de depressão em usuários do Twitter [Orabi et al. 2018]. O modelo utiliza as Redes Neurais Convolutivas (RNC) e Redes Neurais Recorrentes (RNR) para o processo de classificação. O primeiro conjunto de dados utilizado [Coppersmith et al. 2015] foi analisado por um anotador humano que avaliou cada declaração para garantir que não houvessem declarações falsas ou irônicas. Cada usuário do *dataset* é rotulado por gênero e idade e depois em classes de depressão, controle e estresse pós traumático (PTSD). O segundo conjunto de dados foi construído com usuários que publicaram *tweets* em uma campanha de conscientização e combate dos estigmas de doenças mentais [Bell Let's Talk 2015]. Os usuários foram selecionados e para cada um deles, até os 3000 *tweets* públicos mais recentes foram incluídos no conjunto de dados. Nesse, os usuários são rotulados em depressão ou controle e não recebem rótulos de idade ou gênero. O *tweet* que revela explicitamente a condição depressiva do usuário, é removido para que não haja viés a partir da amostragem dos dados. No entanto, em outros *tweets*, esses mesmos usuários podem falar sobre o diagnóstico. O pré-processamento dos dados também foi feito para realizar as etapas de treinamento e teste. Foram removidos os *retweets*, as menções, urls, todos caracteres não alfa-numéricos e algumas *stopwords*. A biblioteca NLTK [Loper e Bird 2002] foi usada para tokenizar as mensagens e construir o vocabulário. O modelo desenvolvido tem o objetivo de identificar usuários com risco de depressão a partir do seu perfil na rede social. Para isso, foi utilizada a abordagem de mapeamento das palavras do vocabulário para vetores de números reais, as *word embeddings*. Duas formas foram estudadas: a otimização das *words embeddings* e o treinamento aleatório dessas. Depois, usando o método estatístico *Word2Vec* [Mikolov et al. 2013] para aprender o vetor das *word embeddings*, foram estudadas duas formas: *skip-gram* e *continuous bag of words* (CBoW). Na primeira

forma, deduz-se as palavras contextuais próximas a partir de uma palavra central. Já na segunda (*bag of words* contínuo), faz-se o contrário: a partir de palavras de um contexto, deduz-se a palavra central. Foram comparadas as quatro abordagens citadas, juntamente com as redes neurais recorrentes e convolutivas, mudando parâmetros de cada uma para comparar e avaliar qual modelo gera um resultado melhor para a solução exigida. A avaliação apresenta que as redes neurais convolutivas apresentam um melhor resultado para o primeiro e o segundo conjuntos de dados no referente estudo.

Com o mesmo conjunto de dados utilizado pelo estudo citado anteriormente, [Coppersmith et al. 2015] foi desenvolvido um algoritmo para identificar depressão através de mídias sociais [Nadeem 2016]. Para cada usuário selecionado, os 3000 *tweets* mais recentes publicados foram incluídos no conjunto de dados. Para a extração de atributos do conjunto, foi utilizado a abordagem de *Bag of Words*, onde cada palavra existente no conjunto de dados é considerado como um atributo. E, para cada *tweet*, conta-se quantas vezes a palavra aparece. Depois da extração dos atributos feita, os algoritmos de classificação foram utilizados. Os classificadores escolhidos foram Árvores de Decisão (AD) [Quinlan 1986], Classificação Vetorial de Suporte Linear (LSVC) [Joachims 2006], *Naive Bayes* (NB) [Friedman, Geiger e Goldszmidt 1997], a Regressão Logística (LR) [Berger, Pietra e Pietra 1996] e, por fim, Ridge [Aljouie e Roshan 2015]. Diante de todos os resultados avaliados por métricas usuais, conclui-se que a melhor acurácia foi ganha com o algoritmo *Naive Bayes*, com uma taxa de 0.86.

Sabe-se que a Inteligência Artificial vem colaborando com muitas questões do mundo, inclusive científicas. Porém, a Inteligência Artificial ainda não é capaz de substituir totalmente um olhar humano especializado sobre algum problema. A intenção é que se desenvolvam máquinas cada vez mais inteligentes para que consigam agregar ainda mais para a solução de problemas. Comparar os índices de acertos de uma máquina com os índices de acertos humano, pode ser um bom auxílio para saber como melhorar modelos e aproximá-los de uma eficiência humana.

Um estudo desenvolveu um classificador automatizado de computador que pudesse replicar a acurácia de classificadores humanos para detectar indícios de suicídio em dados do Twitter [O'Dea et al. 2015]. Este estudo detecta o nível de preocupação com as postagens individuais no Twitter que fizeram referências textuais ou audiovisuais, de formas diretas ou indiretas, ao suicídio. Primeiramente, foi feita a coleta dos dados do Twitter. Palavras e/ou frases que remetiam ao suicídio foram escolhidas como palavras-chaves para que a coleta fosse restringida a um certo grupo de *tweets*. Após esse processo, codificadores humanos classificaram os textos dos *tweets* em classes diferentes: fortemente preocupante, possivelmente preocupante e seguro ignorar. A classe "fortemente preocupante" inclui *tweets* de pessoas que transmitem um desejo sério e pessoal de cometer suicídio. A classe "seguro ignorar", inclui *tweets* que não sugerem o risco de suicídio. E,

por último, "possivelmente preocupante"é a classe padrão de todos os *tweets* coletados, ou seja, para sair dessa classe, o *tweet* deve apresentar conteúdo referente a alguma das outras duas classes e, caso não apresentasse, deveria ser mantido em "possivelmente preocupante". Para classificar os *tweets*, os anotadores humanos foram separados em duas duplas e, caso a dupla tivesse uma opinião diferente referente a um *tweet*, um terceiro anotador faria a classificação. Para a segunda etapa do estudo vários métodos de aprendizagem de máquina foram utilizados para classificar automaticamente os *tweets* nas três classes diferentes. Cada palavra contida no conjunto de dados foi tratada como um atributo e para explorar esses atributos, utilizou-se três métodos: a frequência simples que calcula a frequência de cada palavra em um documento, a frequência do termo–inverso da frequência nos documentos (TFIDF) que calcula a importância de uma palavra em um *tweet* em relação ao conjunto de todos os *tweets* e o filtro que remove as palavras que ultrapassam um limite determinado para a frequência do documento. Para a classificação automática, dois classificadores foram escolhidos, o *Support Vector Machine* (SVM) [JOACHIMS 1999] e a Regressão Logística (LR) [Berger, Pietra e Pietra 1996]. Utilizando métricas para comparar resultados, notou-se que houve uma concordância moderada a boa entre os anotadores humanos com uma taxa de 0.76, e, para os classificadores de máquina, a taxa de concordância geral foi a mesma, de 0.76.

O objetivo de identificar sinais de outras doenças mentais além da depressão, como a bipolaridade, o estresse pós-traumático e o transtorno afetivo sazonal também foi estudado em textos do Twitter [Coppersmith, Dredze e Harman 2014]. Para esse fim, *tweets* em que os autores explicitavam o diagnóstico de alguma das quatro doenças citadas foram coletados, esse foi o grupo denominado de Grupo Diagnosticado. O rótulo dessas quatro classes inclusas no Grupo Diagnosticado foi feito de forma manual para que todo tipo de ironia ou falsas declarações fossem removidas. Os *tweets* mais recentes de cada usuário pertencente às classes, foram incluídos no conjunto de dados. Para equilibrar esse conjunto, *tweets* aleatórios também foram coletados e rotulados como Controle. Os dados foram analisados para identificar características existentes no conjunto de dados de cada uma das doenças do Grupo Diagnosticado, demonstrando que existe um modelo linguístico similar entre os textos de cada uma das classes estudadas, ou seja, palavras de significados parecidos aparecem com uma certa frequência na maioria dos textos de uma certa classe (a palavra tristeza na classe depressão, por exemplo). Para essa análise, foi feito o uso da Investigação Linguística e Contagem de Palavras (LIWC) [Pennebaker, Francis e Booth 2001] . A análise realizou-se de duas formas diferentes: utilizando a análise da palavra sem contexto, a chamada técnica 1-gram e também utilizando a análise da palavra com contexto em uma janela de 5 palavras, 5-gram. Conclui-se que há sinais quantificáveis e relevantes nos dados do Twitter para o transtorno bipolar, depressivo maior, transtorno de estresse pós-traumático e transtorno afetivo

sazonal.

Sabe-se que o comportamento de pessoas em redes sociais pode ser multifacetado, ou seja, o comportamento de um mesmo usuário pode ser observado em publicações, fotos de perfil, informações pessoais expostas, entre outras. Através de um modelo criado pôde-se estudar as características dos usuários do Twitter de diferentes perspectivas e capturar a relação entre tais modalidades e, dessa forma, revelar a discrepância de comportamento entre usuários deprimidos e não deprimidos nas redes sociais.[Shen et al. 2017]. Para esse fim, criou-se três conjuntos de dados. A coleta dos dados foi baseada em um *tweet* âncora (o *tweet* principal da coleta para cada usuário. Este *tweet* possuía as condições exigidas para ser coletado, como as palavras chaves) com critérios específicos para cada conjunto e os demais *tweets* postados em até um mês após a coleta do *tweet* âncora também foi incluído no respectivo conjunto. O primeiro deles, chamado de D1 incluiu usuários que postaram um *tweet* revelando explicitamente o diagnóstico depressivo. O segundo conjunto de dados (D2) possui *tweets* de usuários que nunca mencionaram em algum deles a palavra "*depress*"(depressão). O D3 contem *tweets* de usuários que escreveram *tweets* em que a palavra "*depress*"(depressão) aparece. Os dois primeiros conjuntos são rotulados como depressivo e não depressivo, respectivamente. O terceiro conjunto não possui rótulo. Foi feito um pré-processamento dos dados para diminuir a dificuldade da análise de sentimentos. O pré-processamento contou com a stemização de palavras, processamento de palavras irregulares com o modelo *word2vec* [Mikolov et al. 2013] e o processamento de *emojis*. A extração de atributos baseou-se nos chamados comportamentos *offline* dos usuários. Seis grupos de atributos foram levados em consideração:

- Atributos de rede social: inclui o número de *tweets* postados, número de seguidores e horário em que os *tweets* foram postados.
- Atributos do perfil do usuário: gênero, idade, *status* de relacionamento e nível educacional dos usuários.
- Atributos visuais: extraiu-se da imagem de *avatar* dos usuários as cinco combinações de cores, brilho, saturação, proporção de cores frias e proporção de cores nítidas.
- Atributos emocionais: palavras que demonstram sentimentos positivos e negativos foram consideradas por meio da Investigação Linguística e Contagem de Palavras [Pennebaker, Francis e Booth 2001], extração de sentimentos de *emojis* (três anotadores votam para o sentimento de cada *emoji*) e, por último, utilizou-se normas afetivas para palavras em inglês [Bradley et al. 1999] para extrair características de valência, domínio e nível que são eficazes para explicar emoções humanas.
- Atributos de Nível de tópico: assuntos discutidos por pessoas depressivas são diferentes de pessoas não-depressivas. Foi feito o uso da alocação de diretórios latentes (LDA) [Blei, Ng e Jordan 2003] para extrair a distribuição de tópicos nos

documentos.

- Atributo de domínio específico: foram extraídas palavras que compunham um léxico antidepressivo e também palavras-chaves que correspondiam à sintomas da doença.

O modelo apresentado foi o de Aprendizagem de Dicionário Depressivo Multimodal (MDL) para que fosse possível aprender a representação dos usuários pela aprendizagem de dicionário e capturar os padrões comuns. Validou-se a efetividade do modelo MDL e três outros classificadores foram usados para treinar e detectar usuários deprimidos com os recursos aprendidos. Os classificadores escolhidos para compor o estudo foram o *Naive Bayes* (NB) [Friedman, Geiger e Goldszmidt 1997], Aprendizagem Múltipla de Redes Sociais (MSNL) [Song et al. 2015] e Aprendizagem de Dicionário Wasserstein (WDL) [Rolet, Cuturi e Peyré 2016]. Entre as escolhas, avaliou-se cada uma e conclui-se que o modelo criado para o problema, que combina a estratégia multimodal e a estratégia de aprendizado de dicionário, é eficaz na detecção de depressão, pois esse alcançou o melhor resultado entre as avaliações.

Para fim de comparação do estado da arte atual do assunto tratado, resume-se os trabalhos que contribuíram para esta pesquisa na Tabela 3.1.

Tabela 3.1: Resumo da revisão sistemática

	Algoritmos	Dados	Classes	Anotação	Atributos
1	RNR RNC	CLPsych 2015 shared task *Tweets* da campanha Bell Let's Talk	PTSD Depressão Controle	Anotador humano	Word embeddings Skip-gram CBoW
2	AD LSVC NB LR Ridge	CLPsych 2015 shared task	PTSD Depressão Controle	Anotador humano	BoW
3	SVM LR	*Tweets* que têm palavras relacionada ao suicídio	Suicídio Controle Não Suicídio	Anotador humano	Frequência TFIDF Filtro
4	LIWC	*Tweets* contendo declaração de diagnóstico de alguma doença mental e *tweets* aleatórios	PTSD Depressão Controle SAD	Anotador humano	N-grams Contagem
5	MDL NB MSNL WDL	*Tweets* contendo declaração de diagnóstico de de depressão, *tweets* com e sem a palavra "depressão"	Depressão Não Depressão	Palavras chaves	Word embbedings Dicionário

A primeira coluna refere-se aos trabalhos mencionados neste capítulo, a segunda aos classificadores e modelos utilizados em cada um, a terceira coluna refere-se aos conjuntos

de dados utilizados e/ou coletados, a quarta são as classes de rótulo, a quinta é reservada para o método de coleta e a última coluna para descrever os atributos utilizados em cada estudo.

O presente trabalho a ser apresentado, diferentemente dos demais revisados neste capítulo, busca conseguir bons resultados mediante a um bom pré-processamento dos dados usando técnicas simples, mas que podem se mostrar bem eficientes no tratamento de textos com linguagem natural. Essa pesquisa também investe em uma boa hiper-parametrização dos algoritmos usados, podendo aumentar o poder de classificação dos mesmos.

CAPÍTULO 4

Metodologia

O objetivo desta pesquisa é identificar incidências de depressão através de dados coletados na rede social Twitter. Para isso, um conjunto de dados publicado em um estudo anterior [Shen et al. 2017] foi utilizado na presente pesquisa. Entretanto, o estudo referente ao conjunto de dados aqui utilizado, analisou, em diferentes aspectos, as informações obtidas de um perfil no Twitter, como foto do perfil, quantidade de seguidores, horário de publicação dos *tweets*, entre outros. Dessa forma, não havendo análise relacionada unicamente ao processamento de linguagem natural dos textos publicados pelos usuários presentes no *dataset*, propõe-se a análise deste tópico. Algumas etapas foram realizadas até o objetivo final que se resume em fazer predições em novos dados com uma boa confiabilidade.

As etapas a serem realizadas são:

- Estudo dos dados
- Pré-processamento dos textos
- Separação do conjunto de dados em treino e teste
- Treinamento do classificador com os dados de treino
- Teste da confiabilidade do classificador com os dados de treino
- Utilização de métricas para avaliar os resultados da classificação
- Análise das métricas

Todos os processos e experimentos realizados foram na linguagem *Python* nas versões 2.7 [Rossum e Jr 1995] e 3.6 [Rossum e Drake 2009]. Os dados foram obtidos através de uma interface de programação de aplicações (API) disponibilizada pelo *Python* para coletar dados do Twitter. Foi utilizado também a plataforma Colab [Bisong 2019] disponibilizada pela Google para rodar o processamento dos dados. Do conjunto de dados obtido, foram selecionados apenas os *tweets* de língua inglesa para que houvesse uma simplificação e padronização dos processos. Os atributos foram extraídos por meio de um *bag of words*.

CAPÍTULO 5

Experimentos

5.1 *Corpus*

Os conjuntos de dados obtidos [Shen et al. 2017] foram originalmente coletados do Twitter. Para isso, foram empregados métodos heurísticos baseados em regras para construir dois conjuntos de dados: de depressão e não depressão. O método de coleta foi baseado em obter um *tweet* âncora para cada usuário de cada um dos conjuntos de dados. Como as pessoas devem ser observadas por um período de tempo, de acordo com a experiência clínica, todos os outros *tweets* publicados dentro de um mês a partir do *tweet* âncora também são obtidos. Informações sobre os perfis dos usuários incluídos em cada um dos *datasets* também foram coletados.

- D1: o grupo D1, chamado de conjunto de dados com depressão inclui *tweets* âncoras contendo frases como (*I'm/ I was/ I am/ I've been*) *diagnosed depression*"que, em uma tradução para língua portuguesa, significa "(Eu estou/ Eu fui/ Eu estive) diagnosticado(a) com depressão". O conjunto contém *tweets* publicados entre 2009 e 2016. Esse recebeu o rótulo de depressão.
- D2: chamado de conjunto de dados sem depressão, inclui *tweets* em que os usuários nunca publicaram um *tweet* contendo a palavra "*depress*"(depressão). Todos os *tweets* selecionados foram publicados em dezembro de 2016.
- D3: diferentemente dos conjuntos D1 e D2, D3 é um conjunto não rotulado que inclui usuários, também de dezembro de 2016, em que seus *tweets* âncoras contém a palavra "*depress*"(depressão).

Foram excluídos dos conjuntos usuários que possuíam mais de 15mil seguidores, pois esse poderia ser um indício de um perfil organizacional ou famoso. Também foram excluídos usuários em que os *tweets* âncoras narravam uma história, uma citação e que, consequentemente, não era um *tweet* que expunha a condicional pessoal do usuário. Com esta mesma ideia, *retweets* foram excluídos, pois poderiam não estar afirmando a mesma condição pessoal do *tweet* original.

5.2 Pré-Processamento

5.2.1 Construção do Conjunto de Dados

Para o fim desta pesquisa, foi construído um único *dataset* a partir dos conjuntos de dados rotulados originais, D1 e D2. Assim sendo, o conjunto D3 não foi incluído no estudo.

Para a construção do novo conjunto de dados para a classificação proposta, os conjuntos com depressão e não depressão foram tratados individualmente e depois unificados em um único *dataset*, cada um com seu rótulo.

Limpeza dos Textos

Visando conseguir um melhor resultado, os dados foram pré-processados, eliminando possíveis ruídos que atrapalhassem uma boa predição dos mesmos. A primeira etapa foi incluir apenas os *tweets* âncoras da cada usuário presente nos dois conjuntos. Sendo assim, nenhuma informação adicional do perfil do usuário foi utilizado, apenas o texto do *tweet* âncora. Essa seleção restringiu os *tweets* que continham apenas textos em inglês. Nos dois conjuntos, foram feitos os mesmo processos de limpeza dos dados, exceto por uma etapa exclusiva ao conjunto com depressão.

Primeiramente, foram removidas as url's dos textos, uma vez que elas não trazem significado ao *tweet*. Tem-se, a seguir, uma exemplificação com o texto de um dos *tweets* utilizando todas as funções de limpeza dos textos para facilitar a interpretação de todo o processo.

Tweet original:

["@23weeksocks 2day I spend 3 hours in my psychiatry session. I was diagnosed with depression amp anxiety ☹☹ ☹ #MatExpHour http://t.co/PD47ciOCd4"]

Tweet após remoção de url's:

["@23weeksocks 2day I spend 3 hours in my psychiatry session. I was diagnosed with depression amp anxiety ☹☹ ☹ #MatExpHour "]

Em seguida, são removidas as menções feitas no *tweet*, pois elas acrescentam apenas ruídos aos dados:

["2day I spend 3 hours in my psychiatry session. I was diagnosed with depression amp anxiety ☹☹ ☹ #MatExpHour "]

Depois, os caracteres especiais foram removidos. Nessas funções houve uma atenção especial ao notar que, no estudo de análise de sentimentos em redes sociais, os *emojis* podem fazer grande diferença justamente por experessarem emoções. Portanto, houve um cuidado ao remover caracteres especiais sem retirar os *emojis* presentes no texto.

["2day I spend 3 hours in my psychiatry session. I was diagnosed with depression amp anxiety ☹☹ ☹ MatExpHour "]

Ainda sobre emojis, faz-se um tratamento específico nos textos que possuem esses tais valores. Pode se considerar que cada emoji expressa uma emoção. Como a extração de atributos será feita por *bag of words*, deve-se ter um cuidado para que cada emoji seja analisado separadamente. Suponha que uma pessoa escreva um *tweet* com três emojis tristes. Cada um deles representa uma emoção triste. Se esses não forem separados por espaços, os três juntos serão considerados como um único *token*. Se for feita a separação de cada um deles, eles serão tratados individualmente e, dessa forma, contados três vezes no documento. A frequência irá aumentar, e consequentemente, a análise poderá ser mais correta. Por este motivo, fez-se a separação por espaços de cada emoji encontrado em um texto.

["2day I spend 3 hours in my psychiatry session. I was diagnosed with depression amp anxiety ☹ ☹ ☹ MatExpHour "]

Devido ao tipo de arquivo em que os dados foram remontados para uma nova base, a formatação transformou os "&"("e"comercial) existentes em um cadeia de caracteres igual a "&amp". Para que o "&"e "and"tenham os mesmos significados, foi feita a substituição necessária:

["2day I spend 3 hours in my psychiatry session. I was diagnosed with depression and anxiety ☹ ☹ ☹ MatExpHour "]

Foi feita a remoção de algarismos, que não trazem contribuição para a análise de sentimentos dos *tweets*.

["day I spend hours in my psychiatry session. I was diagnosed with depression and anxiety ☹ ☹ ☹ MatExpHour "]

E, por fim, foi feita a lematização dos textos.

["day I spend hour in my psychiatry session. I was diagnosed with depression and anxiety ☹ ☹ ☹ MatExpHour "]

O processo exclusivo ao conjunto de dados com depressão é a remoção das palavras/frases chaves que foram usadas para a coleta dos dados. Se essas chaves não forem removidas, um viés pode ser criado no modelo. Assim, o modelo irá ficar super ajustado aos dados [Hawkins 2004], dando uma falsa impressão de que a modelagem está ótima, quando na verdade, ele apenas aprendeu que quando certas palavras chaves estão em um texto, ele tem um rótulo específico. O texto do exemplo é um *tweet* do conjunto de dados com depressão e, portanto, sofreu este processo.

["day I spend hour in my psychiatry session. and anxiety ☹ ☹ ☹ MatExpHour "]

As chaves removidas estão listadas na Tabela 5.1

Tabela 5.1: Chaves Removidas

I'm diagnoised depression
I was diagnoised depression
I am diagnoised depression
I've been diagnoised depression
I'm diagnoised with depression
I was diagnoised with depression
I am diagnoised with depression
I've been diagnoised with depression
I have been diagnoised depression
I have been diagnoised with depression
Im diagnoised depression
Im diagnoised with depression

Depois da limpeza feita em todos os dados de cada um dos dois conjuntos, estes foram unidos em um só. Os rótulos também foram modificados para binários para facilitar a computação dos classificadores. Dessa forma, o conjunto não depressivo (negativo) foi rotulado com o número 0, e o conjunto depressivo, com o número 1. Na Tabela 5.2 tem-se uma pequena amostra do conjunto de dados construído com o D1 e D2 e também já limpo.

Tabela 5.2: Amostra do Conjunto de Dados

0	Ive just watched episode SE of Black Mirror blackmirror
0	Im way too understanding for people to ever need a reason to bullshit or avoid me
0	but im sleepy so nvm
0	LJasmin AdeleGorgeousxx I am funny girl i love new experience i love dancing talking
0	Im l on CounterStrikeGlobalOffens wizebot twitch ⋆
1	Im a severe case of paranoia and
1	What I wouldve told myself the day I via themightysite
1	Part Ive been anxiety and been through just about every kind
1	PartyCyclops I used to smoke a lot of weed but stopped when I two year ago
1	I in the early s and have been on every antidepressant available some two te around

O conjunto de dados formado tem as seguintes características:

- 6482 *tweets* positivos (classe depressiva)
- 4267 *tweets* negativos (classe não depressiva)
- 10749 *tweets* no total
- A proporção desse conjunto é de aproximadamente 60% de *tweets* positivos e 40% de *tweets* negativos, o que pode ser considerado um conjunto equilibrado.

5.2.2 Extração de Atributos

A próxima etapa consiste em manipular o conjunto de dados de forma que ele seja fornecido como entrada para algum classificador. Para este fim, primeiramente deve haver uma separação do conjunto em uma parte para treino (para o classificador aprender como predizer novos dados) e uma parte para teste (para validar o aprendizado do classificador).

A divisão do conjunto foi feita na proporção de 80% dos dados, incluindo os rótulos 0 e 1, para treino e 20% dos dados, também incluindo os rótulos 0 e 1, para teste. Dessa maneira, o conjunto de treino possui 8599 *tweets* e o conjunto de teste possui 2150.

Para conseguir a extração de atributos efetivamente, faz-se a tokenização e a vetorização dos dados. Isto é, transformar cada palavra presente no conjunto de dados em um *token* e, a cada documento, contar quantas vezes este *token* aparece. Esse é o método *bag of words* que será utilizado neste modelo.

Durante o processo de transformação dos dados em um *bag of words*, tem-se a opção de realizar mais algumas mudanças no conjunto de dados, para que, apenas palavras realmente necessárias façam parte do conjunto. Sendo assim, as duzentas palavras mais raras no vocabulário do conjunto foram removidas, pois elas não possuíam um significado claro e que pudesse contribuir para a classificação.

As palavras do *stop word* padrão para língua inglesa da biblioteca NLTK também foram removidas, em exceção às palavras "*I*"e "*me*". A exceção à essas palavras foi feita pois há indícios que elas podem contribuir para a predição correta dos dados, uma vez que pessoas que têm traços depressivos usam mais os pronomes de primeira pessoa [Pennebaker 2011].

Abaixo, uma visualização das dez palavras mais frequentes antes da remoção das *stop words* e das palavras mais raras (Tabela 5.3).

Tabela 5.3: Palavras Mais Frequentes

and	3328
to	2041
im	1905
been	1865
the	1718
have	1646
my	1453
it	1349
anxiety	1219
you	1112

Após a remoção das *stop words* e das duzentas palavras mais raras, as dez palavras mais frequentes podem ser visualizadas na Tabela 5.4

Tabela 5.4: Palavras Mais Frequentes Após Remoção de Palavras

anxiety	1219
years	561
depression	523
like	510
dont	493
ago	473
know	469
clinical	422
severe	403
disorder	376

Ainda no processo de vetorização, foi feita a remoção de pontuações e a normalização dos textos, convertendo todas letras para caixa baixa. Depois de todos os processos realizados antes da vetorização e durante a vetorização, o *bag of words* final foi construído. A Figura 5.1 ilustra o *bag of words* do conjunto de treinamento.

	aaaaaaaaaaah	aaaaaah	aaah	aaand	aabitw	...	zotlam	zozeebo	zs	zwar	zy
0	0	0	0	0	0	...	0	0	0	0	0
1	0	0	0	0	0	...	0	0	0	0	0
2	0	0	0	0	0	...	0	0	0	0	0
3	0	0	0	0	0	...	0	0	0	0	0
4	0	0	0	0	0	...	0	0	0	0	0
...	...	...	...	...	...	...	...	...	..	...	..
8594	0	0	0	0	0	...	0	0	0	0	0
8595	0	0	0	0	0	...	0	0	0	0	0
8596	0	0	0	0	0	...	0	0	0	0	0
8597	0	0	0	0	0	...	0	0	0	0	0
8598	0	0	0	0	0	...	0	0	0	0	0

[8599 rows x 12415 columns]

Figura 5.1: *BoW* do Conjunto de Treinamento

Pode-se observar que a matriz, visivelmente esparsa, possui um tamanho de 8599 x 12415. Isso significa que, como mostrado anteriormente, 8599 *tweets* estão presentes

nesse conjunto e que, no total, tem-se 12415 *tokens* extraídos no processo de tokenização realizado.

5.3 Bibliotecas Usadas

Todos os processos realizados, desde a obtenção dos conjuntos de dados até a análise de resultados após a classificação, foram feitos através de bibliotecas do Python.

Os arquivos originais estavam em formato *.json* e, para conseguir exportá-lo para a plataforma Colab, usou-se a bilioteca disponível pelo Python para trabalhar com arquivos deste formato [Rossum e Drake 2009] e transformou-se cada um deles em um arquivo csv [Altis Dave Cole 2003] usando também a biblioteca csv do Python.

Depois, os dados foram colocados no formato de um *DataFrame* existente na biblioteca do Pandas [McKinney et al. 2010], pois assim, os processos realizados nos dados, poderiam ser feitos de uma forma mais clara e padronizada.

A biblioteca NLTK [Loper e Bird 2002] foi muito usada para o pré-processamento e limpeza dos textos. Processos como tokenização, lematização, escolha das *stop words* foram realizados através de funções da biblioteca.

Para outros processos de limpeza de dados, como a remoção de caracteres especiais, usou-se a biblioteca *Regex* [Rossum e Drake 2009] para trabalhar com expressões regulares e conseguir identificar qual tipo de texto deveria ser removido.

A biblioteca de *Emojis* do Python [Kim 2014] foi usada para reconhecimento dos padrões de *emojis* nos textos.

A *scikit-learn* [Pedregosa et al. 2011] é uma biblioteca muito utilizada na aprendizagem de máquina. Esta biblioteca foi usada na extração de atributos com o método *bag of words* e contagem da frequência de cada palavra em um documento através da função *CountVectorizer()*. Para a classificação, os algoritmos usados também foram importados da *scikit-learn*, assim como uma função para ajuste do hiperparâmetro e funções métricas usadas para avaliar o modelo.

Como durante o processo trabalhou-se com uma matriz multidimensional, preferiu-se utilizar a biblioteca *NumPy* [Oliphant 2006–] que proporciona suporte para trabalhar com esse tipo de estrutura.

Os gráficos usados para melhor visualização e interpretação dos dados foram possíveis através da biblioteca *MatPlot* [Hunter 2007].

5.4 Classificadores

Neste estudo foram utilizados, ao total, seis algoritmos para classificar o conjunto de dados construído. Descreve-se, a seguir, o modo como cada um deles foi usado no

modelo e como deu-se a hiper-parametrização dos classificadores.

5.4.1 *Naive Bayes* (NB)

O melhor ajuste de parâmetro neste classificador foi escolhida de tal forma que não são conhecidas as classes das probabilidades de palavras anteriores e isso não influencia no cálculo probabilidade da próxima palavra (*token)*, assim como a probabilidade de classes anteriores não serem conhecidas. O foco desse algoritmo é trabalhar no pré-processamento dos dados, pois o ajuste de parâmetro obteve pequenas diferenças entre suas combinações.

5.4.2 K-vizinhos mais próximos (KNN)

O número de vizinhos do classificador KNN foi definido como três, um número ímpar para impedir empates e também é um número pequeno que não distancia muito a busca. Outro parâmetro definido foi a função de peso. Neste modelo utilizou-se a função que calcula pontos de peso pelo inverso de sua distância, isto é, quanto mais próximo, mais influencia terá o voto do vizinho. O melhor algoritmo para calcular quais são os três vizinhos mais próximos, dentro deste modelo, foi o *ball tree* que é uma estrutura de dados de particionamento espacial para organizar pontos em um espaço multidimensional. Nesta árvore, as folhas tem tamanho 30 que é o tamanho de folha que melhor administra o custo computacional, velocidade e resultado da classificações. A métrica definida no *ball tree* é a distância padrão euclidiana (distância entre dois pontos). Todos os processos foram realizados de forma paralela.

5.4.3 *Support Vector Machine* (SVM)

Neste, o algoritmo foi otimizado duplamente, pois esta opção funciona melhor quando tem-se mais atributos que amostras para se treinar um classificador. A penalidade, que é um parâmetro para o cálculo do hiperplano onde as classes serão separadas, foi baseado no coeficiente dos vetores esparsos. E a função de perda foi fixado como o quadrado da perda, o que ajuda no ganho de resultados melhores.

5.4.4 Árvore de Decisão (AD)

Na árvore de decisão, o critério utilizado para a qualidade da divisão dos nós foi o ganho de informações, que representa a informação aprendida sobre os rótulos (entropia). A estratégia para divisão foi escolher a melhor taxa de divisão dada o atributo e não de forma aleatória e, o número mínimo de amostras necessárias para dividir um nó interno, foi três.

5.4.5 *Random Florest* (RF)

O Random Forest é um algoritmo que cria uma floresta aleatória. Essa floresta é uma combinação de árvores de decisão. Este algoritmo combina tais árvores para obter um melhor resultado.

A hiper-parametrização desse foi feita de forma que todo o conjunto de dados fosse usado para construir cada uma das árvores. Também, em cada uma, foi usado o critério de ganho de informações. Foram, no total, 100 árvores utilizadas para se criar e combinar uma floresta aleatória.

5.4.6 Perceptron

Neste classificador, o número de épocas foi definido como o padrão (1000), sendo que, a cada época, os dados são embaralhados (fase de treinamento onde cada amostra é apresentada). As soluções anteriores não afetam nas soluções que ainda serão realizadas.

CAPÍTULO 6

Resultados

Todos os seis classificadores listados anteriormente foram treinados e testados no conjunto de dados. A avaliação de cada um deles realizou-se por meio de métricas, sendo elas a acurácia, precisão, *recall* e *F1 Score*. Também foram plotados gráficos da matriz de confusão e da curva de aprendizagem da modelagem para cada algoritmo.

Nas tabelas e figuras a seguir, poderá ser feita uma comparação entre os seis, sendo que, o melhor resultado em cada categoria, será destacado para que haja uma melhor visualização.

Tabela 6.1: Resultados

	Acurácia	Precisão	*Recall*	*F1 Score*
KNN	0.78465	0.86258	0.75956	0.80780
NB	0.87162	0.89912	0.88368	0.89133
SVM	**0.88046**	**0.91626**	0.87978	**0.89765**
AD	0.83581	0.84017	0.89461	0.86654
RF	0.86837	0.86908	**0.91725**	0.89251
Perceptron	0.86093	0.85891	**0.91725**	0.88712

Percebe-se, a partir dessa tabela, que o algoritmo *Support Vector Machine* (SVM) obteve um melhor resultado em quase todos os aspectos. Destacam-se também os classificadores *Random Forest* e Perceptron com bons resultados em *recall*. O pior algoritmo, no geral, foi o k-vizinhos mais próximos que obteve resultados considerados razoáveis, mas não ótimos para o problema, uma vez que tentar classificar pessoas sendo depressivas ou não, tem um grande peso.

Na validação cruzada, onde se pode ter uma ideia melhor da generalização, observa-se o melhor resultado do classificador SVM, bem próximo dos resultados anteriores. Isso mostra que não houve um super ajuste, ou ainda um sobre ajuste dos dados e que as divisões entre treino e teste não influenciaram, de forma negativa ou positiva, nos resultados. A Tabela 6.2 mostra os resultados de uma validação cruzada com 10 *folds* tendo como foco a acurácia do modelo, fazendo a média entre as 10 iterações da métrica.

Tabela 6.2: Validação Cruzada 10-*folds*

	Validação Cruzada
KNN	0.73467
NB	0.87329
SVM	**0.88231**
AD	0.84621
RF	0.86929
Perceptron	0.86454

As próximas tabelas mostram as matrizes de confusão dos seis classificadores modelados ao conjunto de dados com a média entre as 10 iterações da validação cruzada. Nota-se que 0 refere-se ao conjunto de dados não depressivos e 1 ao conjunto de dados depressivos.

Tabela 6.3: Matriz de Confusão - KNN

Predito	0	1	Total
Real			
0	3994	323	4267
1	2527	3955	6482
Total	6471	4278	10749

Tabela 6.4: Matriz de Confusão - NB

Predito	0	1	Total
Real			
0	3690	577	4267
1	785	5697	6482
Total	4475	6274	10749

Tabela 6.5: Matriz de Confusão - SVM

Predito	0	1	Total
Real			
0	3767	500	4267
1	765	5717	6482
Total	4532	6217	10749

Tabela 6.6: Matriz de Confusão - AD

Predito	0	1	Total
Real			
0	3299	968	4267
1	685	5797	6482
Total	3984	6765	10749

Tabela 6.7: Matriz de Confusão - RF

Predito	0	1	Total
Real			
0	3433	834	4267
1	571	5911	6482
Total	4004	6745	10749

Tabela 6.8: Matriz de Confusão - Perceptron

Predito	0	1	Total
Real			
0	3399	868	4267
1	588	5894	6482
Total	3987	6762	10749

A seguir, os gráficos das matrizes normalizadas, isto é, com a porcentagem de cada acerto e erro de cada classe. A cor também é um método interpretativo nos gráficos. Dessa forma, as cores mais escuras indicam um maior número de amostras.

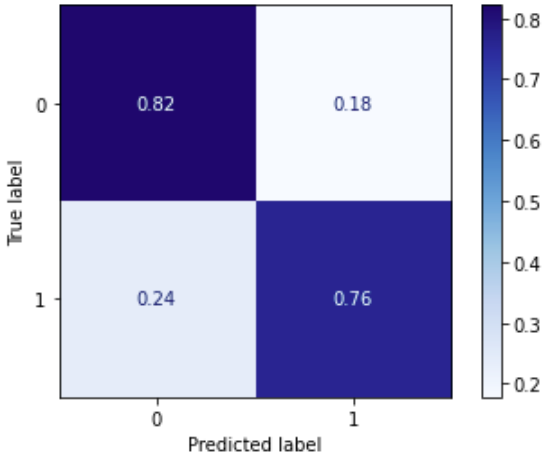

Figura 6.1: Matriz de Confusão Normalizada - KNN

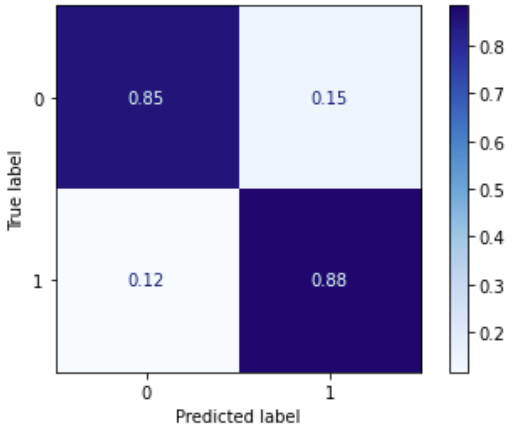

Figura 6.2: Matriz de Confusão Normalizada - NB

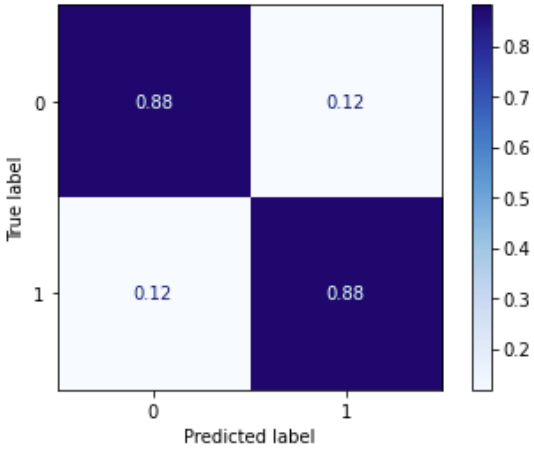

Figura 6.3: Matriz de Confusão Normalizada - SVM

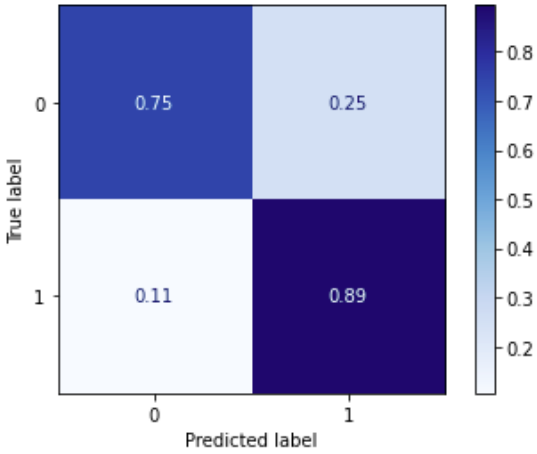

Figura 6.4: Matriz de Confusão Normalizada - AD

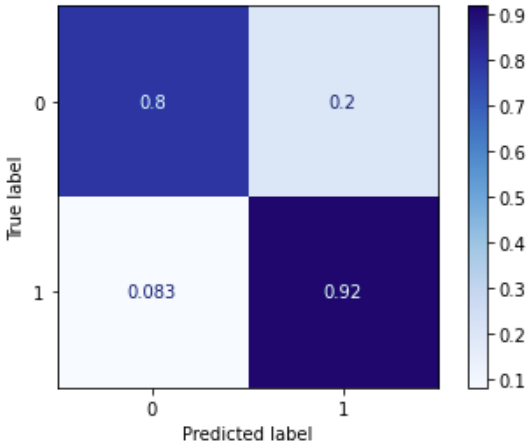

Figura 6.5: Matriz de Confusão Normalizada - RF

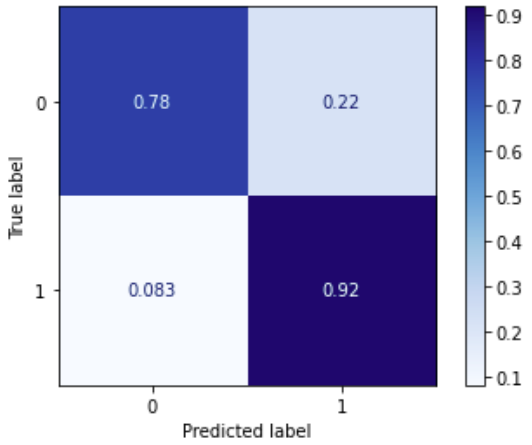

Figura 6.6: Matriz de Confusão Normalizada - Perceptron

De forma geral, o modelo teve mais dificuldade em predizer corretamente os não depressivos, mas a classe depressiva foi a que obteve mais acertos, a exceção desses resultados é do algoritmo KNN que realizou o contrário. Isso deve-se ao fato de que os dados depressivos continham, frequentemente, palavras comuns. Como, por exemplo, ansiedade (*anxiety*), transtorno (*disorder*) e clínico (*clinical*) que são palavras que se relacionam com a depressão e seu diagnóstico, uma vez que a depressão pode caminhar junto com a ansiedade [Castillo et al. 2000].

As curvas de aprendizagem também foram plotadas em um gráfico para observação do comportamento do modelo. Elas são apresentadas abaixo:

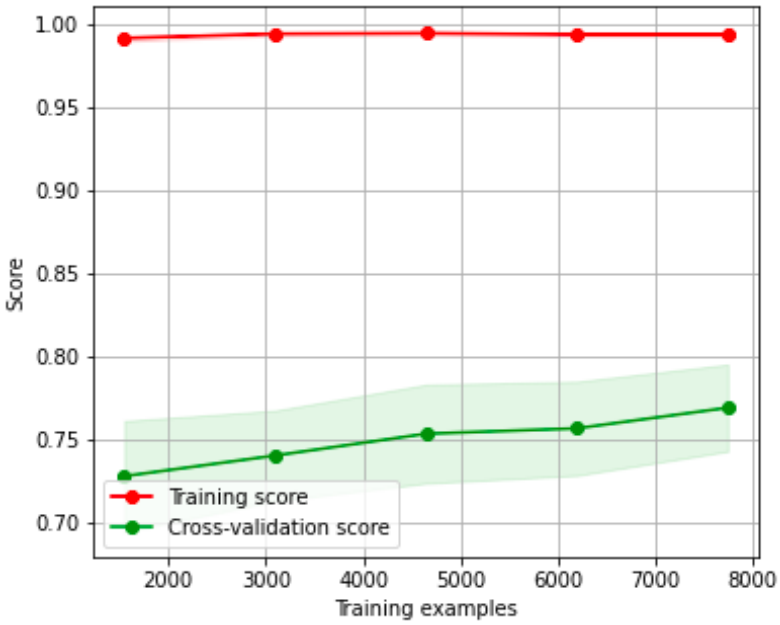

Figura 6.7: Curva de Aprendizagem - KNN

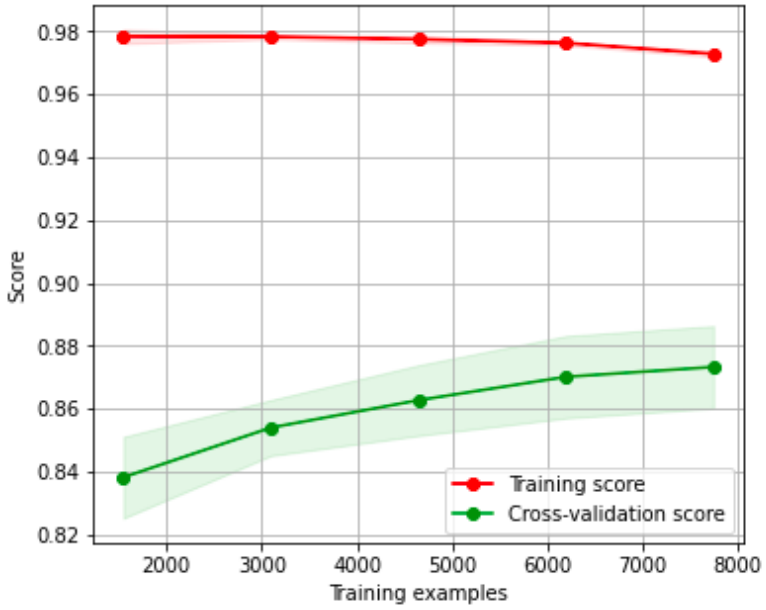

Figura 6.8: Curva de Aprendizagem - NB

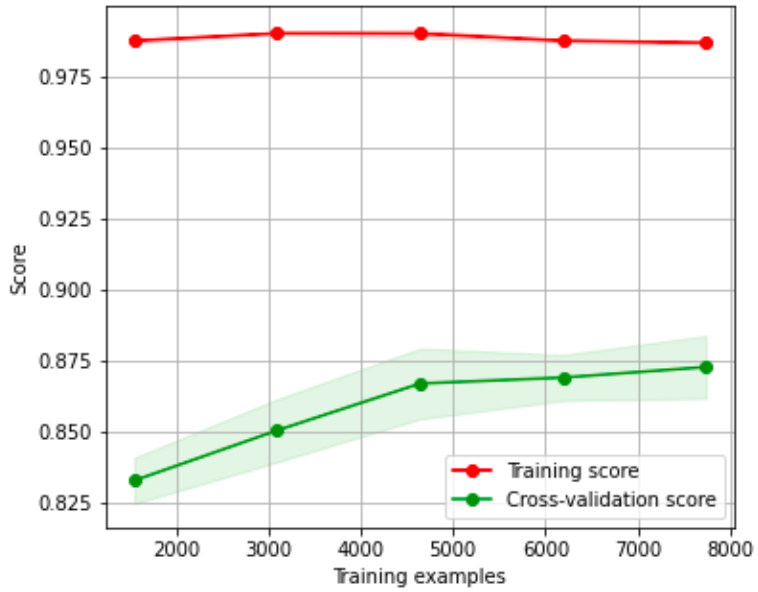

Figura 6.9: Curva de Aprendizagem - SVM

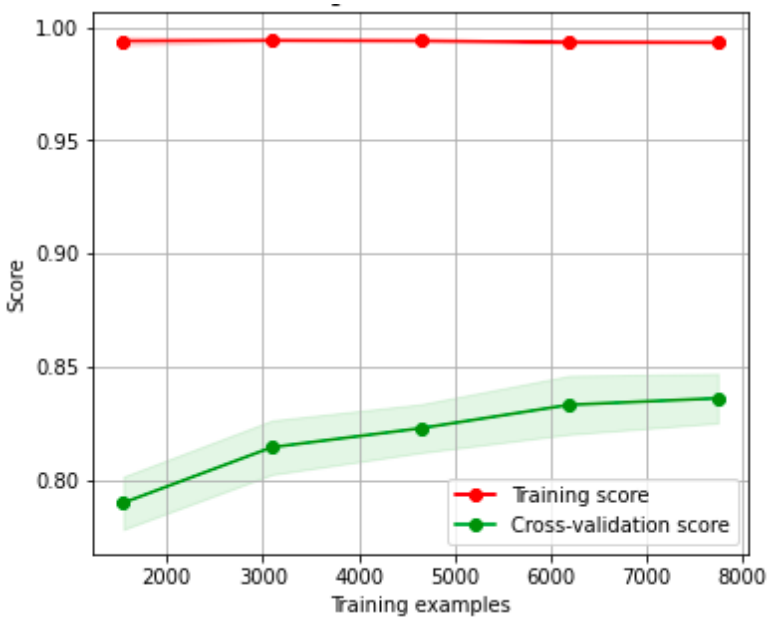

Figura 6.10: Curva de Aprendizagem - AD

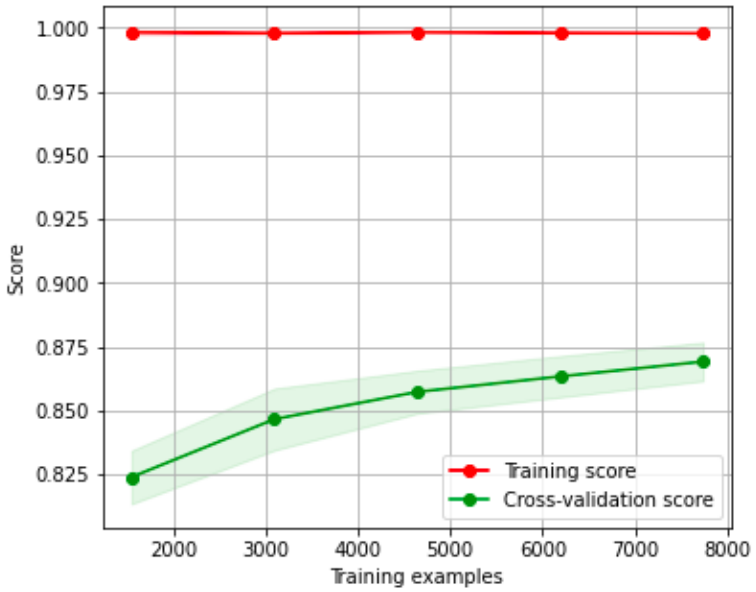

Figura 6.11: Curva de Aprendizagem - RF

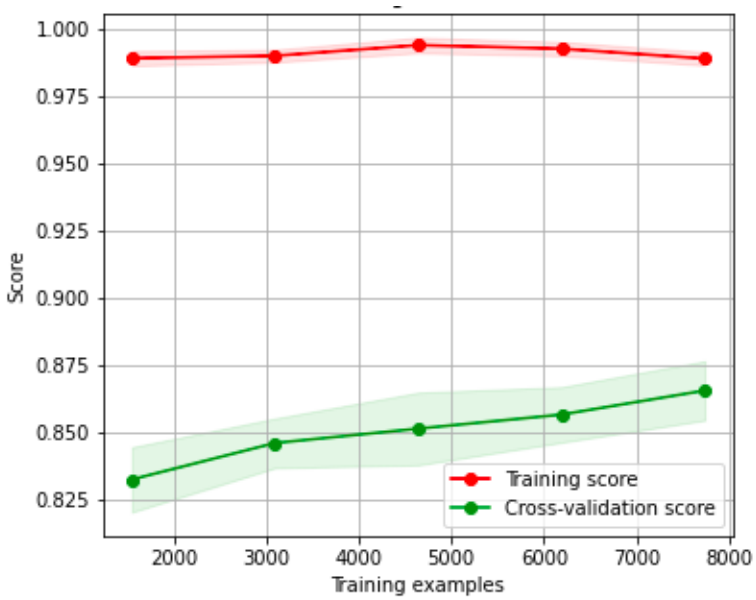

Figura 6.12: Curva de Aprendizagem - Perceptron

As curvas de aprendizagem são, relativamente, parecidas. Isso demonstra que em todos os classificadores, o conjunto de dados comportou-se de uma forma parecida ao modelo, seja com mais acertos ou menos, ainda sim, possuindo uma variância similar de acordo com a quantidade de amostras.

6.1 Análise

Com base nos resultados, percebe-se que o modelo se comportou bem com o conjunto de dados obtido e construído mesmo em classificadores simples como o SVM. O resultado de 88% de acurácia é, certamente, um resultado considerado bom.

O fato de o modelo ter apresentado bons resultados deve-se ao conjunto de dados bem coletados e rotulados e, também, pela forma com que foi feita a limpeza e pré-processamento dos dados.

Apesar de bons resultados, neste caso de estudo, uma máquina que tem 88% de acerto poderia contribuir para estudos na área, mas não substituir um profissional em um diagnóstico da doença, dada a seriedade da depressão.

CAPÍTULO 7

Conclusão

Este estudo teve como objetivo utilizar a aprendizagem de máquina para fazer previsões sobre dados coletados do Twitter, classificando-os em depressivos ou não. Recursos teóricos (como funcionamento de algoritmos) e práticos (como bibliotecas importadas do Python) foram importantes e muito utilizados durante o desenvolvimento da pesquisa. Os presentes recursos foram abordados para melhor compreensão e mais facilidade na construção do modelo.

O modelo construído deu-se através de dados já coletados no Twitter e separados conforme convinha ao propósito da pesquisa. O conjunto construído foi tratado com limpeza e pré-processamento dos dados e, posteriormente, servido como entrada para algoritmos de classificação. Seis algoritmos conhecidos foram utilizados para o modelo e seus desempenhos foram comparados através de métricas extraídas dos resultados obtidos.

Notou-se que houve um bom desempenho mesmo em algoritmos que são conhecidos pela simplicidade. O resultado pode ser considerado bom, apesar de que, neste caso, esses classificadores não poderão substituir profissionais capacitados a distinguir com firmeza um indivíduo acometido pela depressão dos que não possuem a doença.

Apesar disso, o estudo realizado pode contribuir para atividades de monitoramento dos índices de depressão em redes sociais, por exemplo. Dessa forma, pode-se pensar em campanhas nas próprias redes sociais, ou fora delas, para ajudar pessoas que apresentam indícios depressivos em seus meios comunicativos na internet.

Para trabalhos futuros, espera-se melhorar este modelo criado obtendo uma maior confiabilidade no processo de classificação. Este objetivo pode ser alcançado experimentando outras técnicas que não foram abordadas neste trabalho. Técnicas como a *Word2Vec*, comumente usada para extração de atributos em textos [Wang, Ma e Zhang 2016], pode ser uma boa opção na análise de indícios depressivos em *tweets*. Outra possível opção seria a representação de palavras por vetores (GloVe) [Pennington, Socher e Manning 2014] também muito utilizada em classificação de textos [Carvalho 2018]. Saber identificar nos textos quais tipos de palavras (advérbio, adjetivo, substantivo) são mencionadas, pode ser uma forma de melhorar o modelo. Isso é chamado

de *part of speech* e também é uma boa técnica usada durante o processo de classificação de textos [DeRose 1988]. Seria viável, ainda, usar outros algoritmos para fins de comparação e escolha de qual se comporta melhor neste cenário podendo assim, contribuir melhor para os propósitos do estudo.

Referências Bibliográficas

[Aljouie e Roshan 2015]ALJOUIE, A.; ROSHAN, U. Prediction of continuous phenotypes in mouse, fly, and rice genome wide association studies with support vector regression snps and ridge regression classifier. In: LI, T. et al. (Ed.). *ICMLA*. IEEE, 2015. p. 1246–1250. ISBN 978-1-5090-0287-0. Disponível em: <http://dblp.uni-trier.de/db/conf/icmla/icmla2015.htmlAljouieR15>.

[Altis Dave Cole 2003]ALTIS DAVE COLE, A. M. S. M. C. W. K. Csv file api. 2003.

[Anzanello e Fogliatto 2011]ANZANELLO, M. J.; FOGLIATTO, F. S. Learning curve models and applications: Literature review and research directions. *International Journal of Industrial Ergonomics*, v. 41, n. 5, p. 573 – 583, 2011. ISSN 0169-8141. Disponível em: <http://www.sciencedirect.com/science/article/pii/S016981411100062X>.

[Bayes 1763]BAYES, T. An essay towards solving a problem in the doctrine of chances. *Phil. Trans. of the Royal Soc. of London*, v. 53, p. 370–418, 1763.

[Bell Let's Talk 2015]Bell Let's Talk. *Bell Let's Talk*. 2015. Disponível em: <https://letstalk.bell.ca/en/, acessado em abril de 2020>.

[Bento 2017]BENTO, B. M. P. Avaliação de classificadores por árvore de decisão e árvore de regressão em cenas urbanas do sensor worldview-2. INPE, São José dos Campos, 2017. Disponível em: <http://mtc-m21b.sid.inpe.br/col/sid.inpe.br/mtc-m21b/2016/10.24.18.16/doc/publicacao.pdf>.

[Berger, Pietra e Pietra 1996]BERGER, A. L.; PIETRA, S. A. D.; PIETRA, V. J. D. A maximum entropy approach to natural language processing. *Computational Linguistics*, v. 22, n. 1, p. 39–71, 1996. Disponível em: <https://www.aclweb.org/anthology/J96-1002>.

[Bisong 2019]BISONG, E. Google colaboratory. In: ____. *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*. Berkeley, CA: Apress, 2019. p. 59–64. ISBN 978-1-4842-4470-8. Disponível em: <https://doi.org/10.1007/978-1-4842-4470-8$_7$>.

[Blei, Ng e Jordan 2003]BLEI, D. M.; NG, A. Y.; JORDAN, M. I. Latent dirichlet allocation. *J. Mach. Learn. Res.*, JMLR.org, v. 3, p. 993–1022, 2003. ISSN 1532-4435. Disponível em: <http://portal.acm.org/citation.cfm?id=944937>.

[Bradley et al. 1999]BRADLEY, M. M. et al. *Affective Norms for English Words (ANEW): Instruction manual and affective ratings*. 1999.

[Brownlee 2014]BROWNLEE, J. Classification accuracy is not enough: More performance measures you can use. Machine Learnign Mastery, 3 2014. Disponível em: <https://machinelearningmastery.com/classification-accuracy-is-not-enough-more-performance-measures-you-can-use/, acessado em maio de 2020>.

[Brownlee 2017]BROWNLEE, J. A gentle introduction to the bag-of-words model. Machine Learnign Mastery, 10 2017. Disponível em: <https://machinelearningmastery.com/gentle-introduction-bag-words-model, acessado em maio de 2020>.

[Brownlee 2018]BROWNLEE, J. A gentle introduction to k-fold cross-validation. Machine Learnign Mastery, 5 2018. Disponível em: <https://machinelearningmastery.com/k-fold-cross-validation, acessado em maio de 2020>.

[Brownlee 2019]BROWNLEE, J. How to use learning curves to diagnose machine learning model performance. Machine Learnign Mastery, 2 2019. Disponível em: <https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance, acessado em maio de 2020>.

[Carvalho 2018]CARVALHO, M. H. de. Estudo comparativo dos métodos de word embedding na análise de sentimentos. 2018.

[Castillo et al. 2000]CASTILLO, A. R. G. et al. Transtornos de ansiedade. *Brazilian Journal of Psychiatry*, scielo, v. 22, p. 20 – 23, 12 2000. ISSN 1516-4446. Disponível em: <http://www.scielo.br/scielo.php?script=sci$_a$rttextpid = S1516 – 44462000000600006nrm = iso>.

[Claesen e Moor 2015]CLAESEN, M.; MOOR, B. D. Hyperparameter search in machine learning. *CoRR*, abs/1502.02127, 2015. Disponível em: <http://arxiv.org/abs/1502.02127>.

[Coppersmith, Dredze e Harman 2014]COPPERSMITH, G.; DREDZE, M.; HARMAN, C. Quantifying mental health signals in twitter. In: *Proceedings of the Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality*. Baltimore, Maryland, USA: Association for Computational Linguistics, 2014. p. 51–60. Disponível em: <https://www.aclweb.org/anthology/W14-3207>.

[Coppersmith et al. 2015]COPPERSMITH, G. et al. CLPsych 2015 shared task: Depression and PTSD on twitter. In: *Proceedings of the 2nd Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality*. Denver, Colorado: Association for Computational Linguistics, 2015. p. 31–39. Disponível em: <https://www.aclweb.org/anthology/W15-1204>.

[Cortes e Vapnik 1995]CORTES, C.; VAPNIK, V. Support-vector networks. *Mach. Learn.*, Kluwer Academic Publishers, USA, v. 20, n. 3, p. 273–297, set. 1995. ISSN 0885-6125. Disponível em: <https://doi.org/10.1023/A:1022627411411>.

[Cunningham e Delany 2007]CUNNINGHAM, P.; DELANY, S. J. *k-Nearest Neighbour Classifiers*. 2007.

[Dasarathy 1991]DASARATHY, B. *Nearest neighbor (NN) norms: nn pattern classification techniques*. IEEE Computer Society Press, 1991. (IEEE Computer Society Press tutorial). ISBN 9780818659300. Disponível em: <https://books.google.com.br/books?id=k2dQAAAAMAAJ>.

[DeRose 1988]DEROSE, S. J. Grammatical category disambiguation by statistical optimization. *Comput. Linguist.*, MIT Press, Cambridge, MA, USA, v. 14, n. 1, p. 31–39, jan. 1988. ISSN 0891-2017.

[Ferneda 2006]FERNEDA, E. Redes neurais e sua aplicação em sistemas de recuperação de informação. *Ciência da Informação*, v. 35, n. 1, 2006. ISSN 1518-8353. Disponível em: <http://revista.ibict.br/ciinf/article/view/1149>.

[Freund e Schapire 1998]FREUND, Y.; SCHAPIRE, R. E. Large margin classification using the perceptron algorithm. In: *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*. New York, NY, USA: Association for Computing Machinery, 1998. (COLT' 98), p. 209–217. ISBN 1581130570. Disponível em: <https://doi.org/10.1145/279943.279985>.

[Friedman, Geiger e Goldszmidt 1997]FRIEDMAN, N.; GEIGER, D.; GOLDSZMIDT, M. Bayesian network classifiers. *Mach. Learn.*, Kluwer Academic Publishers, USA, v. 29, n. 2–3, p. 131–163, nov. 1997. ISSN 0885-6125. Disponível em: <https://doi.org/10.1023/A:1007465528199>.

[Goodfellow, Bengio e Courville 2016]GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. MIT Press, 2016. Disponível em: <http://www.deeplearningbook.org, acessado em maio de 2020>.

[Hardeniya 2015]HARDENIYA, N. *NLTK Essentials*. [S.l.]: Packt Publishing, 2015. ISBN 1784396907.

[Hawkins 2004]HAWKINS, D. M. The problem of overfitting. *Journal of Chemical Information and Computer Sciences*, v. 44, n. 1, p. 1–12, 2004. PMID: 14741005. Disponível em: <https://doi.org/10.1021/ci0342472>.

[Hunter 2007]HUNTER, J. D. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, IEEE COMPUTER SOC, v. 9, n. 3, p. 90–95, 2007.

[JOACHIMS 1999]JOACHIMS, T. Making large-scale support vector machine learning practical, advances in kernel methods. *Support Vector Learning*, MIT Press, 1999. Disponível em: <https://ci.nii.ac.jp/naid/10011961265/en/>.

[Joachims 2006]JOACHIMS, T. Training linear svms in linear time. In: *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2006. (KDD '06), p. 217–226. ISBN 1595933395. Disponível em: <https://doi.org/10.1145/1150402.1150429>.

[Jurafsky e Martin 2009]JURAFSKY, D.; MARTIN, J. H. *Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition*. Upper Saddle River, N.J.: Pearson Prentice Hall, 2009. ISBN 9780131873216 0131873210. Disponível em: <http://www.amazon.com/Speech-Language-Processing-2nd-Edition/dp/0131873210/ref=pd$_b$xgy$_{bi}$mg_y>.

[Kim 2014]KIM, K. W. T. *Emoji for Python*. [S.l.: s.n.], 2014.

[Kohavi 1995]KOHAVI, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995. (IJCAI'95), p. 1137–1143. ISBN 1558603638.

[Leskovec, Rajaraman e Ullman 2014]LESKOVEC, J.; RAJARAMAN, A.; ULLMAN, J. D. *Mining of Massive Datasets*. 2nd. ed. USA: Cambridge University Press, 2014. ISBN 1107077230.

[Lima e Batista 2018]LIMA, A.; BATISTA, E. J. S. Uma analise de ambientes de programação em blocos com base em recomendações de interação criança-computador. In: ____. [S.l.: s.n.], 2018. p. 87–96. ISBN 978-85-7669-459-5.

[Loper e Bird 2002]LOPER, E.; BIRD, S. Nltk: The natural language toolkit. In: *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*. [S.l.: s.n.], 2002.

[Majumder, Mitra e Chaudhuri 2002]MAJUMDER, P.; MITRA, M.; CHAUDHURI, B. B. N-gram: a language independent approach to ir and nlp. 2002.

[Manning, Raghavan e Schütze 2008]MANNING, C. D.; RAGHAVAN, P.; SCHüTZE, H. *Introduction to Information Retrieval*. USA: Cambridge University Press, 2008. ISBN 0521865719.

[McKinney et al. 2010]MCKINNEY, W. et al. Data structures for statistical computing in python. In: AUSTIN, TX. *Proceedings of the 9th Python in Science Conference*. [S.l.], 2010. v. 445, p. 51–56.

[McTear, Callejas e Griol 2016]MCTEAR, M.; CALLEJAS, Z.; GRIOL, D. *The Conversational Interface: Talking to Smart Devices*. 1st. ed. [S.l.]: Springer Publishing Company, Incorporated, 2016. ISBN 3319329650.

[Mikolov et al. 2013]MIKOLOV, T. et al. Distributed representations of words and phrases and their compositionality. In: BURGES, C. J. C. et al. (Ed.). *Advances in Neural Information Processing Systems 26*. Curran Associates, Inc., 2013. p. 3111–3119. Disponível em: <http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf>.

[Mitchell 1997]MITCHELL, T. M. *Machine Learning*. New York: McGraw-Hill, 1997. ISBN 978-0-07-042807-2.

[Nadeem 2016]NADEEM, M. Identifying depression on twitter. *CoRR*, abs/1607.07384, 2016. Disponível em: <http://arxiv.org/abs/1607.07384>.

[O'Dea et al. 2015]O'DEA, B. et al. Detecting suicidality on twitter. *Internet Interventions*, v. 103, 04 2015.

[Oliphant 2006–]OLIPHANT, T. *NumPy: A guide to NumPy*. 2006–. USA: Trelgol Publishing. [Online; accessed <today>]. Disponível em: <http://www.numpy.org/>.

[OPAS 2018]OPAS, O. P. de S. Technical documents, *Folha Informativa - Depressão*. [S.l.]: Organização Pan-Americana de Saúde, 2018.

[Orabi et al. 2018]ORABI, A. H. et al. Deep learning for depression detection of twitter users. In: *Proceedings of the Fifth Workshop on Computational Linguistics and Clinical Psychology: From Keyboard to Clinic*. New Orleans, LA: Association for Computational Linguistics, 2018. p. 88–97. Disponível em: <https://www.aclweb.org/anthology/W18-0609>.

[Organization 2017]ORGANIZATION, W. H. *Depression and other common mental disorders: global health estimates*. [S.l.], 2017. 24 p. p.

[Pacheco e Pereira 2018]PACHECO, C.; PEREIRA, N. Deep learning conceitos e utilização nas diversas Áreas do conhecimento. v. 2, p. 34–49, 12 2018.

[Pedregosa et al. 2011]PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011.

[Pennebaker 2011]PENNEBAKER, J. *The Secret Life of Pronouns: What Our Words Say About Us*. Bloomsbury Publishing, 2011. ISBN 9781608194971. Disponível em: <https://books.google.com.br/books?id=OWfMNeXw13oC, acessado em maio de 2020>.

[Pennebaker, Francis e Booth 2001]PENNEBAKER, J.; FRANCIS, M.; BOOTH, R. Linguistic inquiry and word count (liwc): Liwc2001. v. 71, 01 2001.

[Pennington, Socher e Manning 2014]PENNINGTON, J.; SOCHER, R.; MANNING, C. D. Glove: Global vectors for word representation. In: *In EMNLP*. [S.l.: s.n.], 2014.

[Powers 2011]POWERS, D. M. W. Evaluation: From precision, recall and f-measure to roc., informedness, markedness & correlation. *Journal of Machine Learning Technologies*, v. 2, n. 1, p. 37–63, 2011.

[Pyle 1999]PYLE, D. *Data Preparation for Data Mining*. 1st. ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999. ISBN 1558605290.

[Quinlan 1986]QUINLAN, J. R. Induction of decision trees. *Mach. Learn.*, Kluwer Academic Publishers, USA, v. 1, n. 1, p. 81–106, mar. 1986. ISSN 0885-6125. Disponível em: <https://doi.org/10.1023/A:1022643204877>.

[Rolet, Cuturi e Peyré 2016]ROLET, A.; CUTURI, M.; PEYRé, G. Fast dictionary learning with a smoothed wasserstein loss. In: GRETTON, A.; ROBERT, C. C. (Ed.). *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*. Cadiz, Spain: PMLR, 2016. (Proceedings of Machine Learning Research, v. 51), p. 630–638. Disponível em: <http://proceedings.mlr.press/v51/rolet16.html>.

[Rosenblatt 1960]Rosenblatt, F. Perceptron simulation experiments. *Proceedings of the IRE*, v. 48, n. 3, p. 301–309, March 1960. ISSN 2162-6634.

[Rossum e Drake 2009]ROSSUM, G. V.; DRAKE, F. L. *JSON encoder and decoder*. Scotts Valley, CA: CreateSpace, 2009. ISBN 1441412697.

[Rossum e Drake 2009]ROSSUM, G. V.; DRAKE, F. L. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009. ISBN 1441412697.

[Rossum e Drake 2009]ROSSUM, G. V.; DRAKE, F. L. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009. ISBN 1441412697.

[Rossum e Jr 1995]ROSSUM, G. V.; JR, F. L. D. *Python reference manual*. [S.l.]: Centrum voor Wiskunde en Informatica Amsterdam, 1995.

[Russell e Norvig 2009]RUSSELL, S.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. 3rd. ed. USA: Prentice Hall Press, 2009. ISBN 0136042597.

[Sammons et al. 2016]SAMMONS, M. et al. EDISON: Feature extraction for NLP, simplified. In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. Portorož, Slovenia: European Language Resources Association (ELRA), 2016. p. 4085–4092. Disponível em: <https://www.aclweb.org/anthology/L16-1645>.

[Sammut e Webb 2010]Accuracy. In: SAMMUT, C.; WEBB, G. I. (Ed.). *Encyclopedia of Machine Learning*. Boston, MA: Springer US, 2010. p. 9–10. ISBN 978-0-387-30164-8. Disponível em: <https://doi.org/10.1007/978-0-387-30164-8$_3$>.

[Saporito 2019]SAPORITO, G. What is a perceptron? In: . [s.n.], 2019. Disponível em: <https://towardsdatascience.com/what-is-a-perceptron-210a50190c3b, acessado em maio de 2020>.

[Schaffer 1993]SCHAFFER, C. Selecting a classification method by cross-validation. In: *Machine Learning*. [S.l.: s.n.], 1993. p. 135–143.

[Shen et al. 2017]SHEN, G. et al. Depression detection via harvesting social media: A multimodal dictionary learning solution. In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*. [s.n.], 2017. p. 3838–3844. Disponível em: <https://doi.org/10.24963/ijcai.2017/536>.

[Silge e Robinson 2017]SILGE, J.; ROBINSON, D. *Text Mining with R: A Tidy Approach*. 1st. ed. [S.l.]: O'Reilly Media, Inc., 2017. ISBN 1491981652.

[Silveira e Bullock]SILVEIRA, G.; BULLOCK, B. *Machine Learning: Introdução à classificação*. Casa do Código. ISBN 9788594188199. Disponível em: <https://books.google.com.br/books?id=XL46DwAAQBAJ, year=2017, acessado em maio de 2020>.

[Song et al. 2015]SONG, X. et al. Multiple social network learning and its application in volunteerism tendency prediction. In: *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: Association for Computing Machinery, 2015. (SIGIR '15), p. 213–222. ISBN 9781450336215. Disponível em: <https://doi.org/10.1145/2766462.2767726>.

[Ting 2010]TING, K. M. Confusion matrix. In: ____. *Encyclopedia of Machine Learning*. Boston, MA: Springer US, 2010. p. 209–209. ISBN 978-0-387-30164-8. Disponível em: <https://doi.org/10.1007/978-0-387-30164-8$_1$57>.

[Ting 2010]TING, K. M. Precision. In: ____. *Encyclopedia of Machine Learning*. Boston, MA: Springer US, 2010. p. 780–780. ISBN 978-0-387-30164-8. Disponível em: <https://doi.org/10.1007/978-0-387-30164-8$_6$51>.

[Wang, Ma e Zhang 2016]Wang, Z.; Ma, L.; Zhang, Y. A hybrid document feature extraction method using latent dirichlet allocation and word2vec. In: *2016 IEEE First International Conference on Data Science in Cyberspace (DSC)*. [S.l.: s.n.], 2016. p. 98–103.

[We Are Social 2018]We Are Social. *DIGITAL IN 2018: WORLD'S INTERNET USERS PASS THE 4 BILLION MARK*. 2018. Disponível em: <https://wearesocial.com/blog/2018/01/global-digital-report-2018, acessado em abril de 2020>.

[Witten, Frank e Hall 2011]WITTEN, I. H.; FRANK, E.; HALL, M. A. *Data Mining: Practical Machine Learning Tools and Techniques*. 3. ed. Amsterdam: Morgan Kaufmann, 2011. (Morgan Kaufmann Series in Data Management Systems). ISBN 978-0-12-374856-0. Disponível em: <http://www.sciencedirect.com/science/book/9780123748560>.

[World Health Organization 2018]World Health Organization. *The top 10 causes of death*. 2018. Disponível em: <https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death, acessado em abril de 2020>.

[Zhang, Jin e Zhou 2010]ZHANG, Y.; JIN, R.; ZHOU, Z.-H. Understanding bag-of-words model: A statistical framework. *International Journal of Machine Learning and Cybernetics*, v. 1, p. 43–52, 12 2010.