Universidade de São Paulo
Instituto de Matemática e Estatística
Bachalerado em Ciência da Computação

Lucas Helfstein Rocha

Estudos em análise de sentimentos da corrida presidencial de 2018

São Paulo
Dezembro de 2018

Estudos em análise de sentimentos da corrida presidencial de 2018

Monografia final da disciplina
MAC0499 – Trabalho de Formatura Supervisionado.

Supervisora: Prof.ª Dr.ª Kelly Rosa Braghetto

São Paulo
Dezembro de 2018

Agradecimentos

Agradeço em primeiro lugar minha orientadora, professora Kelly por ter me ajudado ao longo deste trabalho, mesmo em situações adversas.

Agradeço minha família que me deu suporte em todos os anos de minha graduação, em especial minha avó que sempre esteve ao meu lado mesmo em momentos difíceis.

Agradeço à todos os meus amigos do IME-USP que me ajudaram nessa jornada neste ano, dando opiniões críticas e sinceras para o andamento de meu projeto.

Resumo

Com o crescimento da população brasileira na internet, o uso de redes sociais tem se tornado cada vez mais comum e se tornaram um lugar onde os usuários usam para opinar sobre diversos temas, grande parte das mobilizações que ocorrem em nosso cotidiano começam por conta da internet.

O trabalho tem como objetivo identificar as opiniões da população brasileira sobre os principais candidatos à presidência da República em 2018, durante o período das propagandas eleitorais utilizando técnicas de Processamento de Linguagem Natural ao lado de técnicas de Aprendizado de Máquina.

Foi estudado o problema de classificação textual supervisionada, aplicando esses os conceitos em textos de *tweets* sobre os principais candidatos à presidência da República. Os *tweets* foram coletados antes e durante a campanha eleitoral dos candidatos utilizando a API do *Twitter* e classificados manualmente através de uma ferramenta na internet.

Os utilizados dados foram preparados com técnicas de Processamento de Linguagem Natural para a extração de informações dos tweets e a classificação dos tweets foram feitas utilizando métodos e procedimentos de Aprendizado de Máquina.

Ao final do trabalho, foram classificados os *tweets* coletados durante a campanha e baseando-se no classificador desenvolvido ano final deste trabalho, é possível ter uma noção de que os comentários da população à respeito dos candidatos eram majoritariamente negativos.

Palavras-chave: análise de sentimentos, redes sociais, eleições.

Sumário

Capítulo 1

Introdução

1.1 Contextualização

As redes sociais são fundamentais nas discussões da sociedade moderna, elas são o local onde a maior parte das conversas e debates ocorrem atualmente. A população brasileira tem crescido em participação nas redes sociais, como apontado em *Global Digital Report 2018* (We Are Social, b).

2018 é um ano de eleições presidenciais no Brasil, é esperado que tanto a população quanto os candidatos comecem a difundir mais o tema e isso seja refletido nas redes sociais. Este comportamento já foi observado em países como Holanda e Reino Unido, como apontado em *Social media as beat: Tweets as news source during the 2010 British and Dutch elections* (Broersma e Graham, 2012).

Também já foram feitos estudos com foco no Twitter e o alinhamento político de eleitores com os partidos nas eleições presidenciais da Alemanha em 2009, *Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment* (Andranik Tumasjan, 2010).

1.2 Motivação

A motivação deste trabalho partiu de um interesse pessoal em observar a movimentação do cenário político nas redes sociais ao longo do ano e de conseguir prever opiniões da população brasileira neste contexto utilizando conceitos de Aprendizado de Máquina e Processamento de Linguagem Natural.

Processamento de Linguagem Natural é uma área da Linguística, Computação e Inteligência Artificial que trata da interação entre computadores e a linguagem natural humana. O foco principal é compreender como seres humanos lidam com a linguagem e, assim, desenvolver ferramentas computacionais que possam processar linguagem natural para se atingir as tarefas desejadas, *Natural Language Processing* (G. Chowdhury, 2003).

1.3 Objetivo

Este trabalho tem o objetivo de analisar a polaridade de mensagens em redes sociais sobre candidatos à presidência nas eleições brasileiras de 2018. A identificação automática de polaridade em mensagens textuais é uma combinação de um problema de Processamento de Linguagem Natural e uma tarefa de coleta de dados da opinião de uma parcela da população brasileira através de redes sociais.

Para a obtenção dos resultados apresentados neste trabalho, foi desenvolvida uma metodologia e um arcabouço computacional genéricos, que podem ser reaplicados para a análise de polaridade em contextos eleitorais futuros.

1.4 Estrutura da monografia

No capítulo 2 é explicado sobre a Análise de Sentimentos e como este campo pode se relacionar com as redes sociais. Em seguida, é feita uma breve análise da situação das redes sociais no Brasil e o crescimento da população brasileira com acesso a essas redes. É explicado também como é feita a coleta de *tweets*, descrevendo as particularidades encontradas em relação à estrutura dos *tweets* e à janela de tempo limitada que o *Twitter* oferece a desenvolvedores que não pagam pelo serviço.

No capítulo 3 são explicados diversos tópicos relacionados a área de aprendizado de máquina e é feita uma definição formal do problema de classificação textual que é foco deste trabalho. Também são dadas explicações breves sobre o funcionamento dos algoritmos utilizados nos experimentos deste trabalho, além de serem apontadas referências.

O capítulo 4 trata dos experimentos. São definidas as métricas são utilizadas para avaliação dos modelos e são feitos comentários sobre os dados coletados para os experimentos, indicando fatores exógenos que podem influenciar os classificadores que trabalham com dados de períodos eleitorais. É explicado os termos que são removidos na limpeza textual e o *pipeline* dos experimentos, além de serem feitas observações críticas sobre os resultados dos experimentos.

No capítulo 5 são feitas considerações finais sobre o trabalho, as contribuições feitas e os possíveis trabalhos futuros.

Capítulo 2

Análise de Sentimentos e redes sociais

2.1 Análise de Sentimentos

O campo de Análise de Sentimentos, também conhecido como Mineração de Opinião, usa o Processamento de Linguagem Natural, análise textual e Linguística Computacional para identificar, extrair, quantificar e estudar sentimentos de usuários de um sistema. A análise de sentimentos é muito usada em avaliações de produtos, pesquisas, redes sociais e diversas aplicações que podem ir de propaganda até serviços de saúde.

Zhang e Liu (2012, pp.10) dividem as formas de se analisar sentimentos nas seguintes categorias:

- Documento: nesse caso analisa-se o sentimento associado a um documento como um todo, o tratando de uma forma mais genérica em relação aos demais.
- Sentença: esse tipo de análise trabalha com o sentimento associado a cada sentença de um documento.
- Entidade: um produto ou pessoa sobre o qual o texto se refere, seja direta ou indiretamente. Nessa categoria se busca analisar se a opinião do texto em torno da entidade é positiva ou não.
- Aspecto: características sobre uma entidade. Uma entidade pode ser um produto, por exemplo, tendo como aspectos seus materiais de construção ou até mesmo seu preço. Nessa categoria se busca avaliar a opinião acerca de cada um dos aspectos.

Neste trabalho é feita uma análise de sentimentos em relação a uma entidade: o candidato à presidência da República. Essa classificação é feita através de técnicas de aprendizado de máquina apropriadas para lidar com o problema de classificação de textos.

Em *Sentiment analysis algorithms and applications: A survey* (Walaa Medhat, 2015) são apontadas as duas abordagens possíveis para se fazer uma análise de sentimentos. A primeira abordagem utiliza um dicionário léxico para fazer a análise se baseando

numa pontuação entre palavras positivas e negativas em uma sentença. A segunda abordagem, utilizada neste trabalho, é com aprendizado de máquina. Essa abordagem consiste no uso de algoritmos de classificação em conjunto com técnicas para processamento textual.

Em textos coletados de redes sociais, a análise das opiniões expressas está condicionada à identificação de diversos tipos de elementos, como abreviações, acontecimentos do mundo real, memes, gírias, palavrões, *emojis* e linguagem informal.

Segundo Zhang e Liu (2012, pp.16), a análise de sentimentos em *tweets* é uma tarefa mais fácil por serem mensagens com comprimento máximo definido, em que os usuários procuram se expressar de maneira mais direta a respeito do tema. É esperado que uma análise de sentimentos em *tweets* seja mais precisa.

2.2 Redes sociais no Brasil

A parcela da população brasileira que acessava redes sociais em 2014 era de 48%, como apontado em *Global Digital Report 2014* (We Are Social, a). Em 2018, aproximadamente 61% da população brasileira (130 milhões) acessa redes sociais, representando um crescimento de 7% em relação ao ano de 2017 e um crescimento de 13% da população em redes sociais no período entre duas eleições presidenciais. Dados retirados de *Global Digital Report 2018* (We Are Social, b).

A rede social mais utilizada no Brasil é o *Youtube*, com 60% dos usuários que utilizam redes sociais, seguida de perto pelo *Facebook*, que tem 59% dos mesmos usuários. O *Twitter* tem uma parcela de 27% dos usuários que utilizam redes sociais, o que representa cerca de 35 milhões de brasileiros.

Entre as redes sociais mais utilizadas no Brasil em 2018, o *Twitter* é a que disponibiliza postagens de usuários da maneira mais acessível. A seção 2.3 explica como funciona o mecanismo de transmissão de dados para coleta dos desenvolvedores e também dá um exemplo de postagem coletada.

É possível coletar publicações relevantes a determinados temas através das redes sociais, buscando por expressões e palavras que sejam relacionadas a esses temas. Através desses dados, pode-se analisar o que a população está dizendo a respeito de algum tema ou evento e adquirir mais conhecimento sobre a sociedade brasileira. Neste trabalho, são analisados dados relacionados às eleições gerais no Brasil em 2018 com foco nos candidatos à presidência da República.

As postagens de usuários no *Twitter* são chamadas de *tweets*. Os *tweets* disponibilizados pelo *Twitter* possuem diversas informações atreladas, como: texto, mídias contidas na postagem, nome do usuário que fez a postagem, quantidade de repostagens, horário da publicação, número de curtidas, número de interações, entre outras.

2.3 Obtendo dados do Twitter

Em 2018, um escândalo envolvendo a empresa britânica *Cambridge Analytica* e o uso indevido de dados dos usuários do *Facebook* foi manchete em diversos jornais ao redor do mundo, como em *50 million Facebook profiles harvested for Cambridge Analytica in major data breach* (Cadwalladr e Graham-Harrison, 2018). Este acontecimento serviu de incentivo para que as empresas de tecnologia revisassem suas políticas de privacidade e de controle de dados de seus usuários.

Com a atualização dos termos de privacidade, o *Twitter* busca oferecer aos usuários mais controle sobre os dados e assim acabou por disponibilizar uma quantidade menor de dados aos desenvolvedores através de sua API.

Uma API, do inglês *application programming interface*, é um conjunto de funções proporcionadas para que haja uma comunicação entre programas diferentes. Através de uma API, é possível obter dados de publicações em redes sociais com diversas informações, tais como interações de usuários, curtidas e geolocalização. Para um desenvolvedor acessar uma API de uma rede social, geralmente ele recebe um *token* de autenticação e o utiliza em seu programa. Esse *token* é uma chave única que distingue os usuários que estão acessando os dados da rede social.

A obtenção de dados do *Twitter* é feita através de uma API chamada de *Search API*. Nessa API são oferecidos três possíveis níveis de buscas: comum, *premium* e *enterprise*. As opções *premium* e *enterprise* são pagas. Para usuários comuns, a busca se limita a um conjunto de *tweets* dos 7 últimos dias e prioriza a relevância dos *tweets* sobre a completude dos *tweets* nesse intervalo de tempo, segundo informações disponíveis em Twitter (2018).

Isto significa que um usuário comum que pretende coletar informações acerca de um determinado tema precisa ter uma busca que seja executada continuamente, para evitar a perda de informações relevantes. Eventos importantes que ocorrem ao longo do ano podem ser perdidos pois a API comum não possibilita a recuperação de dados retrospectivos.

Para usar a *Search API* é necessário criar uma conta de desenvolvedor no *Twitter* e gerar um *token* de autenticação para que que a aplicação consiga acessar os dados da rede. Em particular, para a linguagem de programação *Python* há uma biblioteca chamada *Tweepy* que permite uma conexão fácil com o *Twitter*.

No *Tweepy*, basta utilizar o *token* de autenticação do *Twitter* e criar um *listener* para se acessar os dados que os usuários estão criando. Um *listener* é um método programado que fica obtendo dados coletados em JSON diretamente do *Twitter*. JSON é um acrônimo de *JavaScript Object Notation*, que é um formato textual, simples e compacto para a troca de dados entre sistemas. Mais informações podem ser obtidas em *Tweepy* (Tweepy, 2018).

Abaixo há um excerto de *tweet* em formato JSON, com algumas das informações que um *tweet* pode conter. A íntegra de um *tweet* pode ser vista no apêndice A.

```
{
   'contributors': None,
   'coordinates': None,
   'created_at': 'Thu Oct 25 12:59:31 +0000 2018',
   'entities': {
      'hashtags': [],
      'symbols': [],
      'urls': [],
      'user_mentions': [{'id': 3317555339,
         'id_str': '3317555339',
         'indices': [3, 19],
         'name': 'Nada Novo no Front - A Democracia Vencera',
         'screen_name': 'nadanovonofront'}]
    },
   'favorite_count': 0,
   'favorited': False,
   'filter_level': 'low',
   'lang': 'pt',
   'place': None,
   'quote_count': 0,
   'reply_count': 0,
   'retweet_count': 0,
   'retweeted': False,
   ...
}
```


Através do JSON de um *tweet* (como o mostrado acima) é possível extrair muitas informações relevantes, como horário da publicação, texto, se é uma repostagem (*retweet*), localização, plataforma de publicação da postagem, quantidade de seguidores do usuário, entre outras.

Para a coleta dos dados utilizados neste trabalho, foi programado um *listener* que buscou pelos nomes dos principais candidatos à presidência da República em 2018. Os termos pesquisados foram: 'Lula', 'Bolsonaro', 'Marina Silva', 'Ciro Gomes', 'Geraldo Alckmin' e 'Haddad'.

Capítulo 3

Classificação

3.1 Aprendizado de máquina

Aprendizado de Máquina é um ramo da área de Inteligência Artificial onde são estudados e desenvolvidos algoritmos nos quais o que é implementado é a maneira por meio da qual o algoritmo melhora seu desempenho para executar uma tarefa, ao invés de ser programado explicitamente para isso.

Existem dois tipos de algoritmos de Aprendizado de Máquina: os supervisionados, que necessitam tanto de dados quanto de suas respostas, e os algoritmos não supervisionados, que só precisam dos dados para identificar padrões que os permitam modelar uma função que seja pertinente em relação aos dados.

Ambos os tipos precisam passar por uma etapa de treinamento, que é quando os dados (e respostas, caso seja supervisionado) são fornecidos ao algoritmo para que ele, de acordo com cada modelo que seja utilizado, chegue em uma função que após o treinamento consiga predizer a resposta para dados não vistos.

Neste trabalho, o foco é dado para algoritmos de aprendizado supervisionado, usando como dados de entrada *tweets*. Para a construção de um conjunto de treinamento para o classificador a partir dos *tweets* coletados, foi adaptada e utilizada uma ferramenta de classificação manual de *tweets*, o CLAM, desenvolvido em *Django* por Lucas Romão Silva e disponível em *Github* (Romão, 2018).

3.2 Definições

Nesta seção, as definições são feitas da maneira mais genérica possível, de forma que se aplicam a qualquer tipo de texto. O texto que este trabalho tem como foco é o texto de qualquer *tweet* coletado. Será utilizado o termo *feature* como sinônimo de palavras de um determinado texto a ser analisado pelos classificadores e também para características do texto que não sejam palavras, como *n-gramas*.

3.2.1 *Features*

As *features* são as características que são levadas em consideração pelos algoritmos.

3.2.2 Corpus

O conjunto de todos os *tweets* coletados é chamado de corpus.

3.2.3 Vocabulário

O conjunto de todas as palavras distintas contidas nos textos do corpus.

3.2.4 Stop words

As *stop words* são palavras que podem ser removidas do vocabulário por não agregarem um grande valor para o propósito da análise de sentimentos que nos propomos a fazer. Em geral, são as palavras mais frequentes em uma linguagem, como artigos e preposições.

3.2.5 N-gramas

Um n-grama é conjunto de n palavras que aparecem na sequência em um documento. Um exemplo do funcionamento dos n-gramas pode ser visto na figura 3.3.

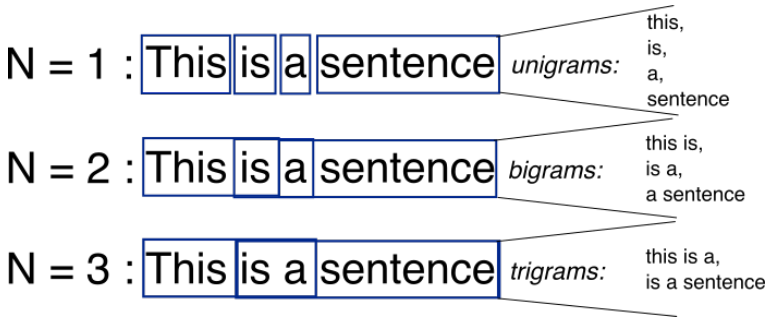

Figura 3.1: *Exemplo de n-gramas para n = 1, 2, 3.*

É comum chamar os 1-gramas de unigramas e 2-gramas de bigramas. Note que uma divisão em unigramas nada mais é do que uma divisão por palavras de um texto.

3.2.6 Representação do corpus em um espaço vetorial

Considere um espaço vetorial EV, definido como

$$EV = \{P_1, P_2, ..., P_i, ...P_n\}$$

onde cada P_i está associado a um n-grama presente no corpus.

Dessa forma, todos os n-gramas do corpus estão representadas em EV. Note que se $n = 1$, EV é uma representação do vocabulário.

3.2.7 Documento

Um documento pode ser representado como um vetor que pertence a EV e que possui a seguinte forma

$$D = \{P_{D_1}, P_{D_2}, ..., P_{D_i}, ...P_{D_n}\}$$

onde cada P_{D_i} representa um valor associado ao n-grama P_i.

Esse valor pode ser modelado de diversas formas. Neste trabalho, são usadas duas modelagens para representação de documentos definidas em *Text Analytics with Python* (Sarkar, 2016, chap. 4): o modelo saco de n-gramas e o valor TF-IDF.

3.2.7.1 O modelo saco de n-gramas

É um método simples e muito eficaz para a tarefa de classificação de texto. M modelo saco de n-gramas é uma representação das frequências absolutas dos n-gramas de um documento. Isto é, cada P_{D_i} possui o valor da frequência do n-grama P_i no documento D.

3.2.7.2 O valor TF-IDF

O modelo saco de n-gramas é bom, mas se baseia apenas nas frequências absolutas das palavras. Isso pode levar a problemas, pois alguns n-gramas que são mais interessantes para uma categorização de um documento tendem a ocorrer com menos frequência do que n-gramas mais comuns de nossa língua.

O valor TF-IDF tem a finalidade de corrigir esse problema. TF-IDF significa *term frequency - inverse document frequency*. É uma combinação das frequências de um termo e da inversa de sua frequência em documentos. O cálculo é feito da seguinte maneira:

$$TF(t) = \frac{\text{número de aparições do termo t no documento}}{\text{número de termos no documento}} \tag{3.1}$$

$$IDF(t) = \log\left(\frac{\text{número total de documentos}}{\text{número de documentos que contêm o termo t}}\right) \tag{3.2}$$

$$TF\text{-}IDF(t) = TF(t) * IDF(t) \tag{3.3}$$

Dessa forma, cada P_{D_i} possui o valor TF-IDF do n-grama P_i no documento D do corpus.

3.3 Classificação textual supervisionada

Com as definições dadas na seção anterior, um problema de classificação textual supervisionada consiste em um conjunto X de textos, onde cada texto precisa ser associado a uma categoria $c \in C$, onde C é o conjunto de categorias. Cada documento é

representado por um vetor de *features* $v \in EV$.

O objetivo é buscar uma função $f : EV \to C$ que corretamente atribua os vetores de EV a suas devidas categorias em C. Para isso, a função é treinada inicialmente com um conjunto de textos D previamente classificados, isto é, um conjunto de pares (v, c) tal que $c \in C$ e $v \in EV$.

É preciso tomar cuidado para que a função que realiza a classificação não tenha um bom desempenho apenas para o conjunto D de textos que é disponibilizado para seu treinamento. Dependendo da complexidade da função utilizada, é possível que os resultados sejam excelentes sobre o conjunto D, mas tenham um desempenho ruim sobre X. Essa situação é chamada de sobreajuste (ou, em inglês, *overfitting*).

3.4 Algoritmos para classificação

As seções a seguir fornecem descrições introdutórias sobre como os algoritmos utilizados nos experimentos deste trabalho funcionam.

3.4.1 Algoritmos probabilísticos

3.4.1.1 Naïve Bayes

O *Naïve Bayes* é um algoritmo probabilístico que funciona muito bem e obtém resultados competitivos em relação a outros métodos mais sofisticados, como apontado em *Tackling the Poor Assumptions of Naive Bayes Text Classifiers* (Rennle *et al.*, 2003).

A principal suposição que caracteriza este algoritmo como *naïve* (ingênuo) é a hipótese de independência das *features*. O algoritmo considera que a ocorrência delas é independente uma das outras dado o contexto de uma categoria. Outra suposição levada em conta é a de que a ordem das palavras dentro de um documento não tem importância. Isso é falso na prática, porém não causa grande efeito nos resultados.

O que gostaríamos de saber é a probabilidade de um documento pertencer a uma determinada classe $c \in C$ dadas suas *features*, isto é, dado algum $d \in EV$. O algoritmo faz uso do teorema de Bayes de probabilidades condicionais, para classificar um texto.

O teorema de Bayes é dado pela seguinte fórmula

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)},$$

onde A e B são eventos.
Para o caso do problema abordado neste trabalho especificamente, ele se torna

$$P(c \mid v) = \frac{P(v \mid c)\, P(c)}{P(v)}.$$

Temos uma equação que representa a probabilidade desejada, se possuirmos estimativas para o lado direito da equação, onde temos três termos. O denominador é

diferente de zero e é o mesmo para todas as categorias, então este termo não representa grande problema.

Na etapa de treinamento, há duas tarefas. A primeira é estimar os valores de $P(c_i)$, para todas as categorias, isto é, para todo $c_i \in C$. Para estimar cada $P(c_i)$ é feita uma contagem, ou seja, para cada categoria, entre todos os documentos do conjunto de treinamento $D \subseteq EV$, determina-se quantos documentos pertencem a ela e obtém-se uma frequência relativa da categoria:

$$P(c_i) = \frac{\mid \{v \in D \mid cat(v) = c_i\} \mid}{\mid D \mid}$$

A segunda tarefa é calcular $P(v \mid c_i)$ para todo par $v \in EV$ e $c_i \in C$. Lembremos que v é um vetor com as *features* e a hipótese do algoritmo é de que elas são independentes. Podemos escrever essa probabilidade como um produto das probabilidades das *features*, dado um c_i:

$$P(v \mid c_i) = P(f_1, f_2, ..., f_n \mid c_i)$$

Sendo assim, o problema a ser resolvido é saber, para cada par de *feature* f_k e categoria c_i, a probabilidade $P(f_k \mid c_i)$ e dessa forma poderemos obter $P(v \mid c_i)$, para qualquer par de v e c_i.

Foi definido anteriormente que os valores das *features* f_k de um documento podem ser tanto a frequência de um n-grama como o seu valor TF-IDF. O fato desses valores serem ou não inteiros não influencia o funcionamento do algoritmo.

Para obter $P(f_k \mid c_i)$, é contabilizada uma frequência relativa para cada *feature* em relação a cada uma das categorias. É uma uma divisão do somatório das vezes que uma determinada *feature* acontece em uma categoria pelo somatório das vezes que todas as *features* pertencentes àquela categoria acontecem.

$$P(f_k \mid c_i) = \frac{\sum_{v \in c_i} \mid f_k \in v \mid}{\sum_{v \in c_i} \left(\sum_{f_k \in v} \mid f_k \in v \mid \right)}$$

Agora temos que para um documento d representado em EV da forma $d = (f_1, f_2, ..., f_n)$ e uma categoria c_i:

$$P(v \mid c_i) = P(f_1, f_2, ..., f_n \mid c_i) = P(f_1 \mid c_i) \cdot P(f_2 \mid c_i) \cdot ... \cdot P(f_n \mid c_i)$$

Com isso, conseguimos calcular a probabilidade de um documento pertencer a cada uma das categorias. Dessa forma, basta atribuir a categoria de maior probabilidade ao documento.

3.4.2 Algoritmos lineares

3.4.2.1 Regressão logística

A regressão logística é uma técnica de aprendizado de máquina muito popular para resolver um problema de classificação binária, isto é, um problema onde o conjunto

de classes é composto de apenas duas categorias.

A regressão logística funciona aplicando a função sigmóide (também chamada de função logística) ao produto interno de um vetor de *features* $v \in EV$, com um vetor de pesos θ da mesma dimensão de v, obtendo assim um valor no intervalo $[0, 1]$.

A hipótese para um determinado vetor v é dada pela equação:

$$h_\theta(v) = sig(z)$$

onde

$$z = \theta^t v$$

e sig é a função sigmóide:

$$sig(z) = \frac{1}{1 + e^{-z}}$$

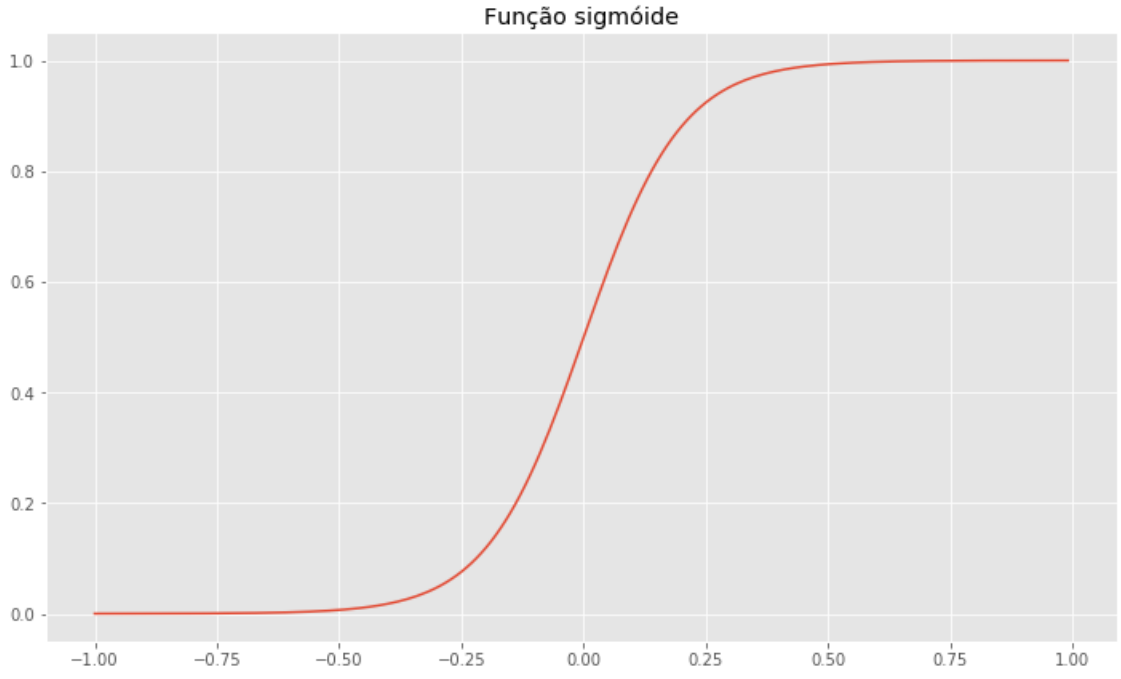

Figura 3.2: *Gráfico do comportamento da função sigmóide.*

A partir de um valor determinado, geralmente 0.5, se considera que o documento representado por f tem uma polaridade positiva. Caso contrário, tal documento é considerado com polaridade negativa.

$$h_\theta(v) \geq 0.5 \rightarrow \text{documento de polaridade positiva}$$

$$h_\theta(v) < 0.5 \rightarrow \text{documento de polaridade negativa}$$

A etapa de treinamento da regressão logística se resume em como estimar o vetor θ de pesos. Este processo não é explicado neste trabalho. Tal explicação pode ser encontrada em *Learning from Data* (Abu-Mostafa *et al.*, 2010, Capítulo 3).

3.4.2.2 SVM

Máquina de vetores de suporte, frequentemente chamado de SVM (do inglês *support vector machine*), é um tipo de algoritmo de aprendizado supervisionado utilizado para classificação, regressão e detecção de anomalias ou *outliers*.

Considerando um problema de classificação binária, a ideia do SVM é a de encontrar uma separação linear entre as categorias dos dados classificados, com uma certa margem de tamanho máximo entre a separação de cada uma das instâncias de cada categoria.

Dados um vetor de *features* $v \in EV$, um vetor de pesos de mesma dimensão w e um número real b, a ideia do algoritmo é criar o seguinte hiperplano:

$$w^T \cdot v + b = 0$$

Os vetores de suporte que dão nome ao algoritmo são margens do hiperplano que separam as categorias onde apenas alguns elementos do conjunto de treino se encontram.

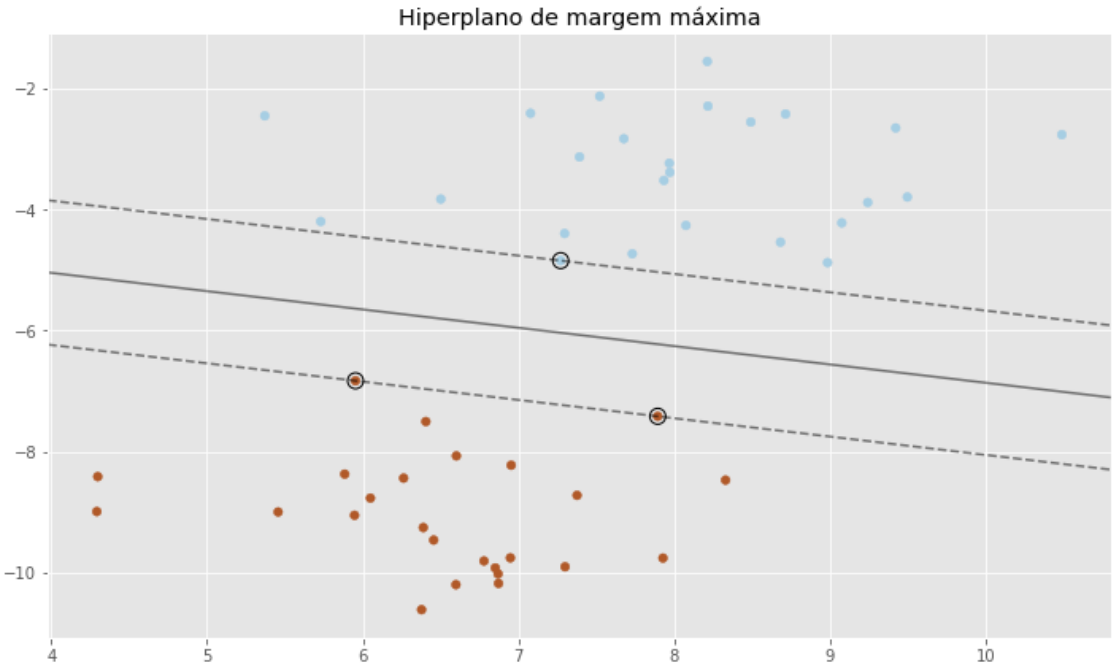

Figura 3.3: *Separação de dados bidimensional. Hiperplano de margem máxima (linha contínua) e vetores de suporte (linhas tracejadas).*

O problema resolvido pelo algoritmo é o de encontrar um hiperplano ótimo, para o problema. Isto é, um hiperplano que tenha a maior margem de separação possível entre as classes. Os valores de w e b são obtidos pela resolução de um problema de otimização quadrática. SVM também pode ser utilizado para casos onde os dados não são linearmente separáveis, com algumas modificações chamadas de *kernel tricks*.

Mais detalhes sobre o problema de otimização do algoritmo e sobre os *kernels* podem ser encontrados em *Training Algorithm for Optimal Margin Classifiers* (Boser *et al.*,

1992).

Capítulo 4

Resultados

4.1 Avaliação e métricas

Para os experimentos realizados, consideramos os seguintes resultados para predições do classificador:

- Verdadeiro positivo (VP): se o *tweet* continha um teor positivo sobre um determinado candidato e a predição foi positiva.
- Verdadeiro negativo (VN): se o *tweet* continha um teor negativo sobre um determinado candidato e a predição foi negativa.
- Falso positivo (FP): se o *tweet* continha um teor negativo sobre um determinado candidato e a predição foi positiva.
- Falso negativo (FN): se o *tweet* continha um teor positivo sobre um determinado candidato e a predição foi negativa.

A seguir são explicadas as métricas utilizadas para avaliar a performance dos classificadores.

4.1.1 Acurácia

Acurácia é uma medida muito utilizada em problemas de classificação pois sua formulação e interpretação são simples. A expressão para seu cálculo é a seguinte:

$$A = \frac{| VP | + | VN |}{| VP | + | VN | + | FP | + | FN |}$$

É a razão entre as amostras classificadas corretamente e o número total de amostras classificadas. Dessa forma, se $A = 1$, todas as amostras foram classificadas corretamente. Se $A = 0$, nenhuma amostra foi corretamente classificada.

4.1.2 Precisão

A precisão de uma classe é o número de verdadeiros positivos dividido pelo número total de amostras estimadas como positivas. É dada pela expressão:

$$P = \frac{| VP |}{| VP | + | FP |}$$

É possível notar facilmente que a precisão está entre zero e um. Para os casos extremos, podemos interpretá-la da seguinte forma: se $P = 1$, então $\mid FP \mid = 0$. Desta forma, todas as amostras classificadas como positivas são realmente positivas, não havendo nenhum falso positivo. Se $P = 0$, então $\mid VP \mid = 0$. Nesse caso, todas as amostras classificadas como positivas na verdade são negativas.

4.1.3 Revocação

Revocação de um classe é o número de verdadeiros positivos dividido pelo número total de amostras positivas. É dada pela expressão:

$$R = \frac{\mid VP \mid}{\mid VP \mid + \mid FN \mid}$$

Note que a revocação também está entre zero e um. Quando $R = 1$, todos os itens que são positivos foram estimados corretamente como positivos pelo classificador. Já para o caso $R = 0$, a interpretação é a mesma de $P = 0$.

4.1.4 *F1-score*

É uma média harmônica entre precisão e revocação, dada pela seguinte expressão:

$$F = \frac{2PR}{P + R}$$

Assim como a precisão e revocação, a *f1-score* assume valores entre zero e um. Note que $F = 1$ se e somente $P = 1$ e $R = 1$.

4.1.5 *ROC Curves*

A análise de curvas de Característica de Operação do Receptor (*ROC Curves*, em inglês) é utilizada para se visualizar, organizar e selecionar classificadores baseando-se em suas respectivas performances. Os gráficos de característica de operação do receptor são gráficos em duas dimensões, onde o eixo x representa a taxa de falsos positivos, enquanto o eixo y representa a taxa de verdadeiros positivos.

Um classificador, ao receber instâncias, produz resultados que podem ser interpretados como pontos neste gráfico. Informalmente, a tendência é que os classificadores que se encontrem na parte superior esquerda sejam melhores, pois têm uma taxa de verdadeiros positivos maior e uma taxa de falsos positivos menor.

Classificadores que se mantêm mais próximos do eixo y no lado esquerdo tendem a ser mais conservadores, classificadores que aparecem mais para o canto superior direito tendem a ser mais liberais. Um classificador que se mantém na linha $y = x$ não é uma boa alternativa, já que tem um comportamento próximo a decisões aleatórias.

A área sob uma curva também pode ser usada para julgar um classificador, ela representa a probabilidade de que um modelo considere um exemplo como positivo numa amostra aleatória . A área sob a curva de um classificador minimamente realista deve ter mais do que 0,5, que seria a área sob a curva de um classificador totalmente

aleatório.

4.2 Sobre os dados coletados

4.2.1 Período de coleta

A coleta dos dados utilizados neste trabalho foi feita em 47 dias distintos. Foram coletados dados entre os dias 21 de julho de 2018 e 28 de outubro de 2018, quando ocorreu o segundo turno das eleições presidenciais.

Os dados foram coletados em janelas de tempo de 10 minutos na maioria dos dias. Em situações especiais, como em datas de debates na televisão, foram coletados *tweets* durante toda a duração dos programas.

Os *tweets* de ocasiões especiais contêm textos que contrastam com a normalidade dos que foram colocados para classificação, por conterem situações momentâneas e específicas, como memes que surgem devido aos debates. Por esse motivo, foram deixados de lado dos testes realizados com os classificadores.

4.2.2 Particularidades do corpus

Durante a construção do corpus coletado, diversos fatores foram notados como consideravelmente importantes para a corrida eleitoral dos candidatos à presidência monitorados para este trabalho. Esta seção trata desses fatores.

4.2.2.1 Inelegibilidade do ex-presidente Lula

Até a votação do TSE sobre a inelegibilidade do candidato Luís Inácio Lula da Silva, ocorrida no dia 31/08/18, o corpus estava sendo construído buscando pelo termo 'Lula'. Após essa decisão, o corpus passou a considerar o candidato Fernando Haddad, representado pelo termo 'Haddad'.

4.2.2.2 Manifestações ocorridas na construção do corpus

Durante o período que antecedeu as eleições gerais no Brasil em 2018, ocorreram diversos fatos que geraram muitas manifestações da população brasileira no *Twitter*. Coletar texto próximo de momentos cruciais em relação às eleições pode provocar um viés no corpus total de *tweets*, além de adicionar memes, expressões e comentários relevantes especificamente a uma data ou acontecimento.

Após o primeiro mês de campanha, os candidatos com o maior número de interações, isto é, que tinham a maior quantidade de comentários e curtidas, eram: Jair Bolsonaro, Fernando Haddad e Marina Silva. Todos os candidatos tiveram aumentos significativos em datas de debates na televisão. O candidato Ciro Gomes teve um crescimento no número de interações e passou Marina Silva apenas nos últimos dias de campanha, informações retiradas de DAPP-FGV (a).

O atentado contra Jair Bolsonaro no dia 06/09/2018 foi o evento de maior repercussão imediata no *Twitter* desde as eleições de 2014, com uma média de 11,8 mil postagens por minuto, segundo relatório da Diretoria de Análise de Políticas Públicas (DAPP) da FGV. Cerca de 40% das postagens questionavam a veracidade do atentado e muitas postagens associavam o atentado a outros candidatos, como o ex-presidente Lula, segundo DAPP-FGV (e).

As eleições gerais de 2018 foram o evento que teve o maior impacto nas redes sociais no Brasil nos últimos 4 anos, com 4,8 milhões de tuítes. Nos dias que antecederam a eleição, os candidatos mais citados foram Jair Bolsonaro e Ciro Gomes DAPP-FGV (c).

Os movimentos pró e anti-Bolsonaro também provocaram grande mobilização nas ruas do Brasil e nas redes sociais, evidenciando mais uma vez a polarização da política nacional. Houve cerca de 3,6 milhões de menções no Twitter somando as manifestações de ambos os lados, de sábado (29 de setembro) até as 11h da segunda-feira (1º de outubro), 1,4 milhão destacaram as *hashtags* contra Bolsonaro, enquanto 1,05 milhão mobilizaram apoio ao candidato, segundo informações de DAPP-FGV (d).

4.2.2.3 Outras formas de menção aos candidatos

Outro fenômeno que dificultou a construção do corpus foi o de usuários que resolveram não utilizar nas postagens o nome de determinado candidato, para evitar que ele ficasse nos assuntos mais comentados do *Twitter* ou até mesmo para se referir a ele de modo carinhoso ou irônico.

Foram usados anagramas e apelidos para quase todos os candidatos, como relatado por Hous (2018). Considerando somente os candidatos usados na construção do corpus que foram para o segundo turno da eleição, alguns dos seguintes termos poderiam estar sendo usados para se referir a algum dos candidatos: “mito”, “capitão”, “bonoro”, “bozo”, “bolso”, “andrade”, “poste”, “prefeitão”. Nenhum desses termos foram considerados na construção do corpus, para se manter o discurso mais próximo da neutralidade de opiniões e não causar impacto muito grande na classificação.

4.2.2.4 Robôs

Nem todo *tweet* pode ser considerado como escrito por um ser humano, dado o grande volume de postagens feitas por robôs nas redes sociais atualmente. Tais robôs são criados com a intenção de veicular mensagens em torno de um certo tema, dando a ele um volume irreal, influenciando os usuários indecisos sobre o tema e também fortalecendo os usuários mais radicais no debate orgânico, considerando o posicionamento frequente dos robôs nos extremos do debate político.

Identificar a presença de robôs e os assuntos que eles geram é muito importante para saber quais situações estão sendo manipuladas e quais são reais no ambiente virtual. Em abril 2017, durante a greve geral, mais de 20% das curtidas e *retweets* ocorridas no Twitter entre os usuários a favor da greve eram provocadas por contas

de *bots*. Já nas eleições presidenciais de 2014, essas contas correspondiam a cerca de 10%. Análise de DAPP-FGV (b).

A utilização de robôs nas eleições gerais de 2018 ficou evidenciada nos casos onde diversos perfis respondiam a postagens que continham “bolovo” e “bolso” com mensagens defendendo o candidato Jair Bolsonaro, como relatado em Folha (2018).

4.2.3 Características textuais

Neste trabalho, a classificação textual é baseada em textos de *tweets*. *Tweets* são limitados em tamanho, o que faz com que usuários abreviem muitas palavras. Os usuários também utilizam muito de linguagem informal, emojis e memes.

Também é necessário levar em conta que nos últimos anos, o Brasil passou por um período turbulento politicamente. Os discursos sobre a situação política chegaram a uma situação de polarização extrema, os textos coletados refletem esse contexto e contêm grandes quantidades de ironia e discursos que beiram o extremismo em alguns casos.

Como exemplo de ironia, é possível citar as frases *“Lula ladrão, teu lugar é na prisão”* e *“Lula ladrão que roubou meu coração”*. Há uma estrutura textual muito parecida, que representa dois sentimentos completamente diferentes.

4.2.4 Termos mais frequentes

Unigramas e bigramas mais frequentes no conjunto de *tweets* disponibilizado para classificação podem ser vistas nas nuvens de palavras da figura 4.1, quanto maior a palavra, maior é sua frequência no corpus.

É de se notar que os nomes dos candidatos aparecem com bastante frequência nas postagens. Na figura 4.2, vemos os unigramas e bigramas mais frequentes após excluirmos os nomes dos candidatos.

Figura 4.1: *Nuvem de unigramas e bigramas mais frequentes dos dados pré-eleições.*

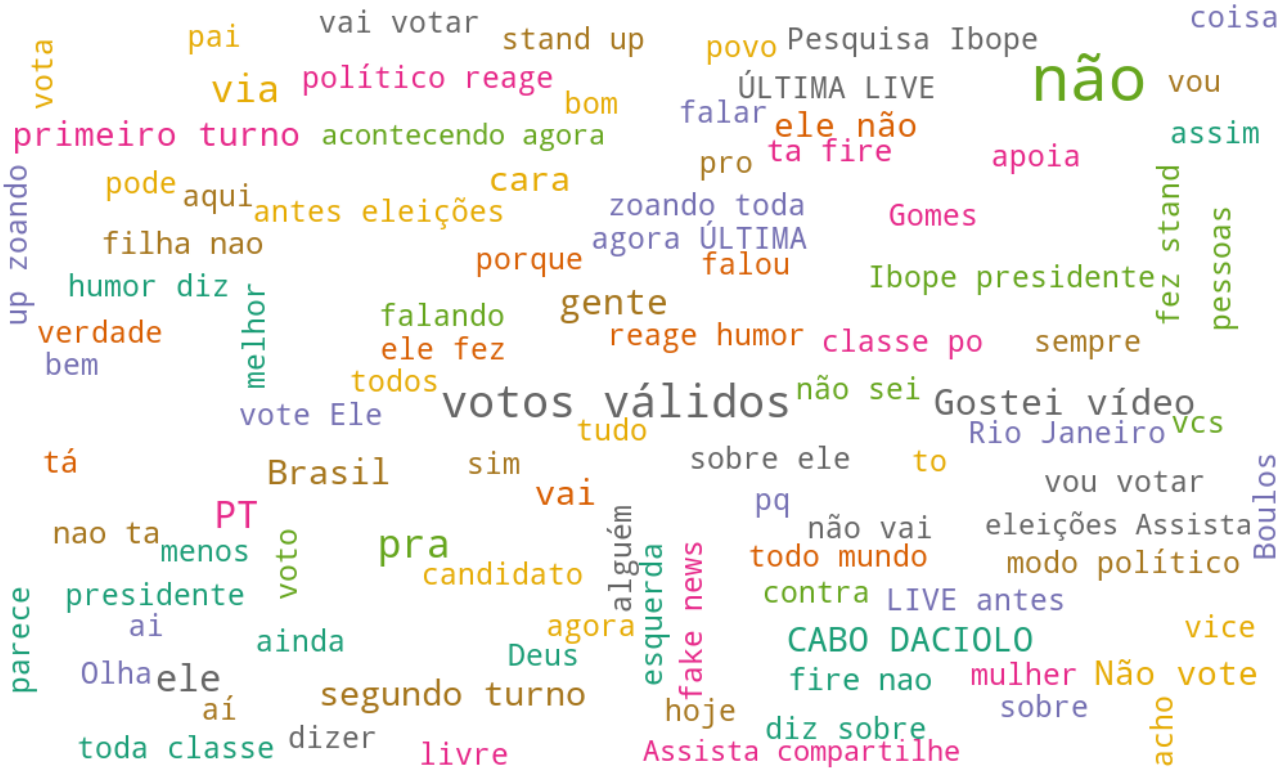

Figura 4.2: *Nuvem de unigramas e bigramas mais frequentes dos dados pré-eleições, sem o nome dos candidatos pesquisados.*

As nuvens de palavras também evidenciaram algumas *stop words* que devem ser levadas em consideração no classificador, como: *RT*, *pq*, *vou*, entre outras.

4.3 Sobre os dados classificados

Foram selecionados *tweets* do período entre 21 de julho de 2018 e 27 de agosto de 2018. Dentro deste período, foram escolhidos aleatoriamente 6400 *tweets* para ficarem na plataforma de classificação. Este limite de *tweets* se deve a uma particularidade técnica do servidor onde a ferramenta ficou hospedada. Foram feitas alterações na ferramenta para que os *tweets* selecionados para classificação fossem pegos de forma aleatória entre os coletados, permitindo assim a escolha de exemplares de diferentes datas do período de coleta.

O período dos *tweets* classificados foi até 4 dias antes do início do horário eleitoral gratuito na televisão e rádio. Essa foi uma decisão de projeto para que as opiniões classificadas fossem mais moderadas e permitissem uma compreensão melhor da opinião geral da população antes do início do horário eleitoral.

Além disso, uma classificação manual feita exclusivamente por um único indivíduo levaria o modelo do classificador a ter um desempenho de acordo com a ideologia de quem o classificou. Para contornar esse problema, a ferramenta de classificação manual de *tweets* foi divulgada para vários usuários, para que a classificação geral tivesse um teor mais imparcial. Os usuários foram instruídos sobre como utilizar a ferramenta e orientados a classificar de maneira apartidária as postagens sobre os candidatos.

Foi considerado também oferecer um mesmo conjunto fixo de *tweets* para diversos usuários e retirar uma média ponderada da polaridade de cada *tweet*. Essa ideia foi descartada pois demandava a contribuição de diversas pessoas não relacionadas ao trabalho e também causaria uma diminuição considerável no tamanho do conjunto rotulado.

A distribuição dos *tweets* classificados manualmente se encontra na tabela 4.1.

Categoria	**Quantidade**
Positivo	225
Neutro	379
Negativo	408

Tabela 4.1: *Distribuição das categorias dos tweets rotulados manualmente.*

4.3.1 Reamostragem

A distribuição dos *tweets* positivos e negativos não está balanceada. Para contornar este problema, é possível aplicar técnicas de reamostragem sobre o conjunto de testes para observar se a performance dos classificadores tem alguma melhora.

Há duas formas de se fazer reamostragem, com sobreamostragem (do inglês, *oversampling*) e sub-amostragem (do inglês, *undersampling*). Na sobreamostragem, integrantes aleatórios da classe com menor número de integrantes são selecionados e duplicados. O ponto negativo neste caso é que uma grande quantidade de elementos replicados pode levar à situação de sobreajuste.

Com sub-amostragem, são removidos elementos aleatórios da classe com maior número de elementos classificados. Neste caso, o ponto negativo é uma perda de informação para o treinamento do classificador.

4.4 Limpeza dos dados

Após a classificação manual, os *tweets* foram submetidos a um processo de limpeza textual onde foram removidos:

- *emojis*
- *usernames*
- *hashtags*
- *stop words*
- *links*

O conjunto dos textos limpos foi então usado como conjunto de treinamento para diferentes algoritmos de classificação a fim de avaliar qual teria o melhor desempenho.

4.5 Experimentos

4.5.1 Tecnologias utilizadas

As implementações dos algoritmos utilizadas neste trabalho são da biblioteca de aprendizado de máquina para *Python* chamada *scikit-learn*. Apenas *tweets* pertencentes às categorias positiva e negativa foram utilizados para o treinamento do classificador. Todos os experimentos aqui descritos foram feitos em um *Jupyter notebook* e podem ser acessados no *GitHub* (Helfstein, 2018).

4.5.2 *Pipeline*

Para todos os experimentos realizados, foram seguidos os seguintes passos:

1. Limpeza textual
2. Retirada das *stop words*
3. Conversão em *features* n-gramas com alguma modelagem
4. Treinamento do classificador
5. Avaliação dos resultados

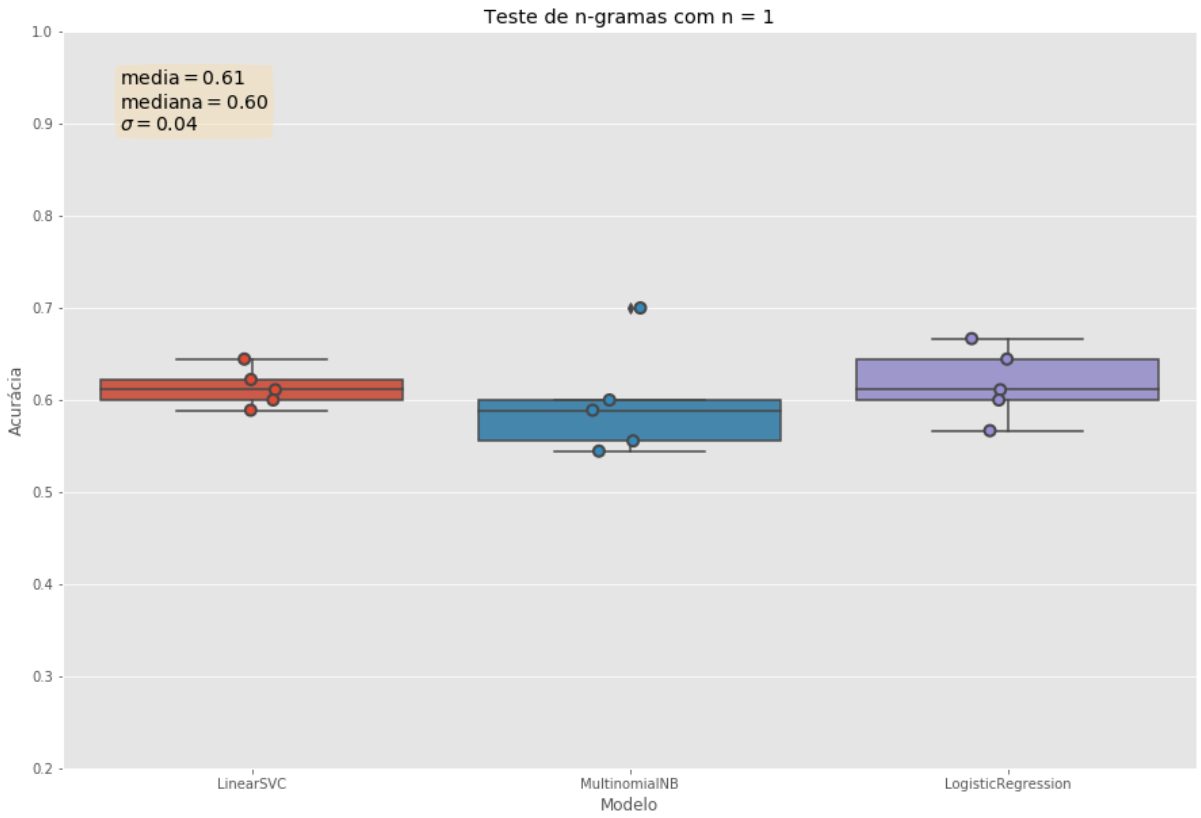

Figura 4.3: *Resultado de acurácias obtidas com unigramas e saco de n-gramas*

4.5.3 Primeiro experimento: determinando um n para os n-gramas

Foram testados valores distintos de n para os n-gramas a fim de determinar qual obteria uma maior acurácia média entre todos os testados. Foram testadas as modelagens baseadas no modelo saco de n-gramas e na valoração TF-IDF. Esses testes foram realizados utilizando validação cruzada do tipo *k-fold* com $k = 5$ no conjunto de treinamento.

Os resultados de acurácia obtidos usando o modelo saco de n-gramas para cada um dos três algoritmos de classificação testados são mostrados nos box-plots das figuras de 4.3 a 4.5. Já os resultados com a modelagem usando os valores TF-IDF são mostrados nas figuras de 4.6 a 4.8.

A partir desses resultados, os unigramas e o modelo saco de n-gramas foram escolhidos para serem utilizados nos demais experimentos por proporcionarem, em média, uma maior acurácia.

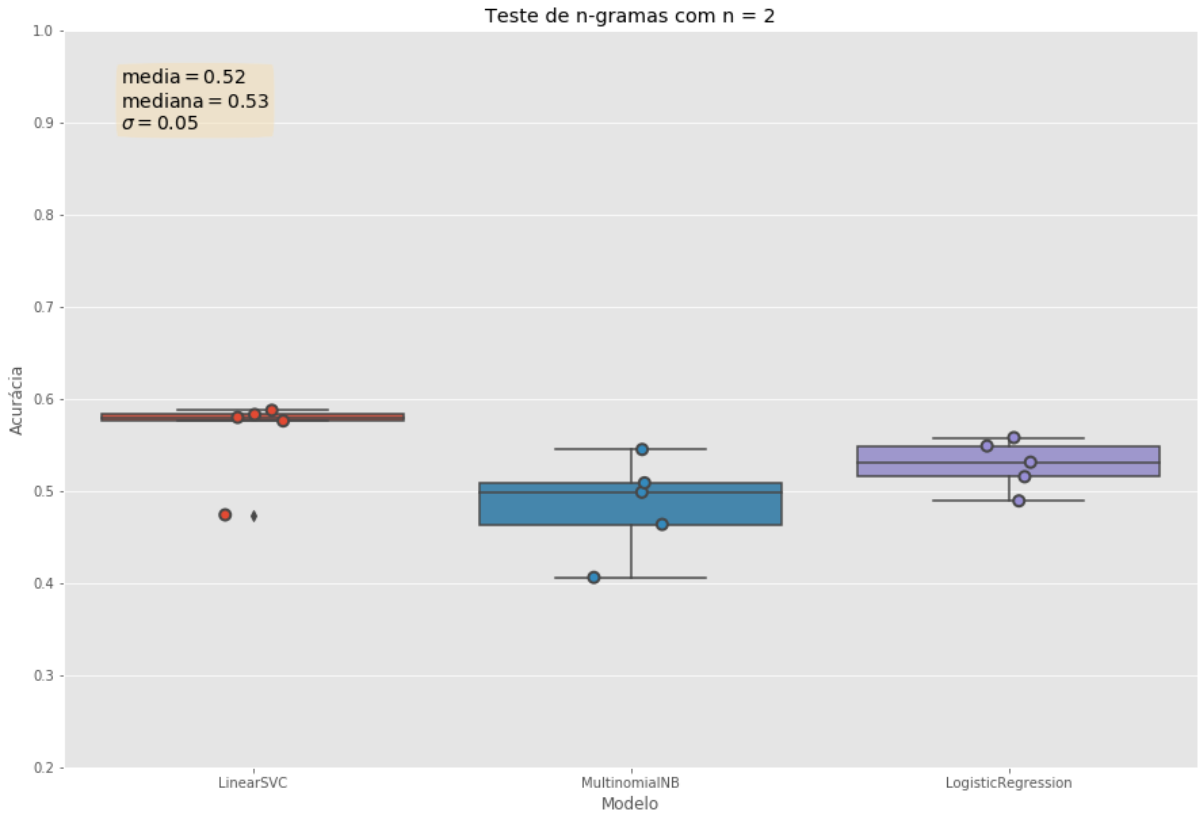

Figura 4.4: *Resultado de acurácias obtidas com bigramas e saco de n-gramas*

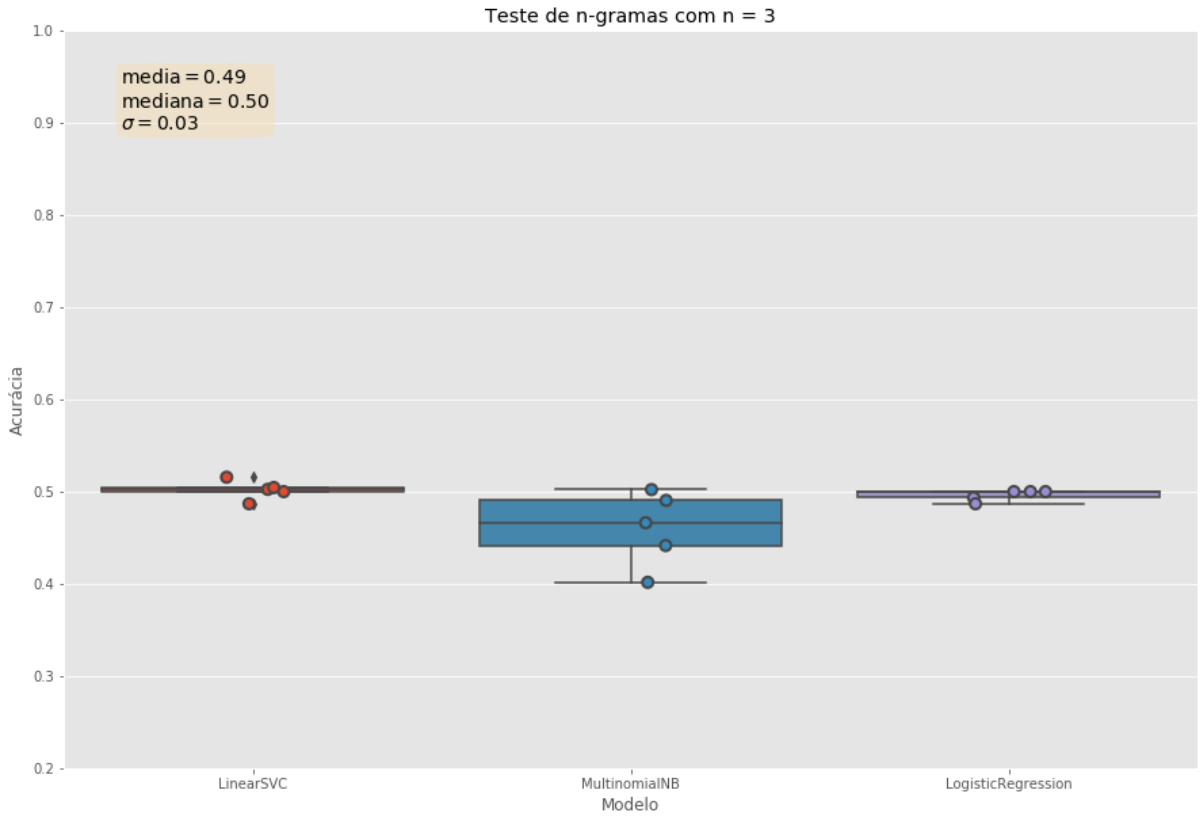

Figura 4.5: *Resultado de acurácias obtidas com trigramas e saco de n-gramas*

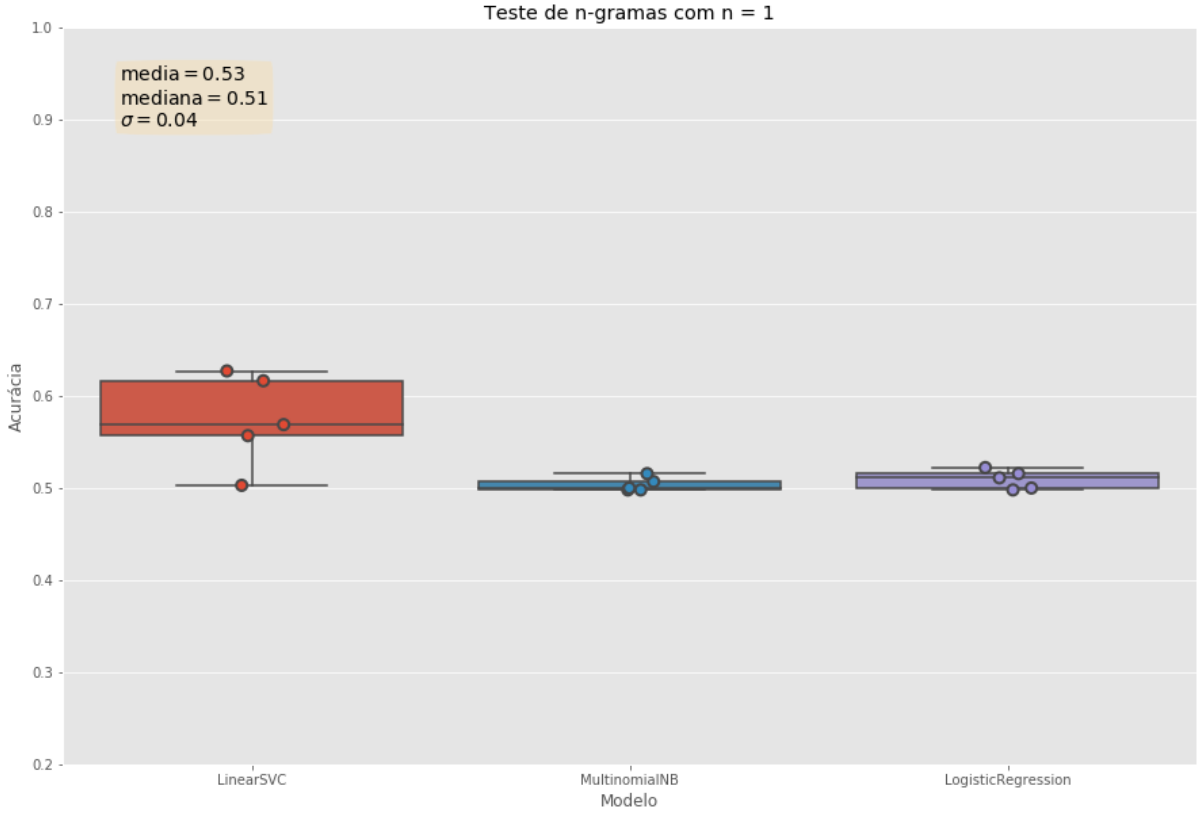

Figura 4.6: *Resultado de acurácias obtidas com unigramas e valoração TF-IDF*

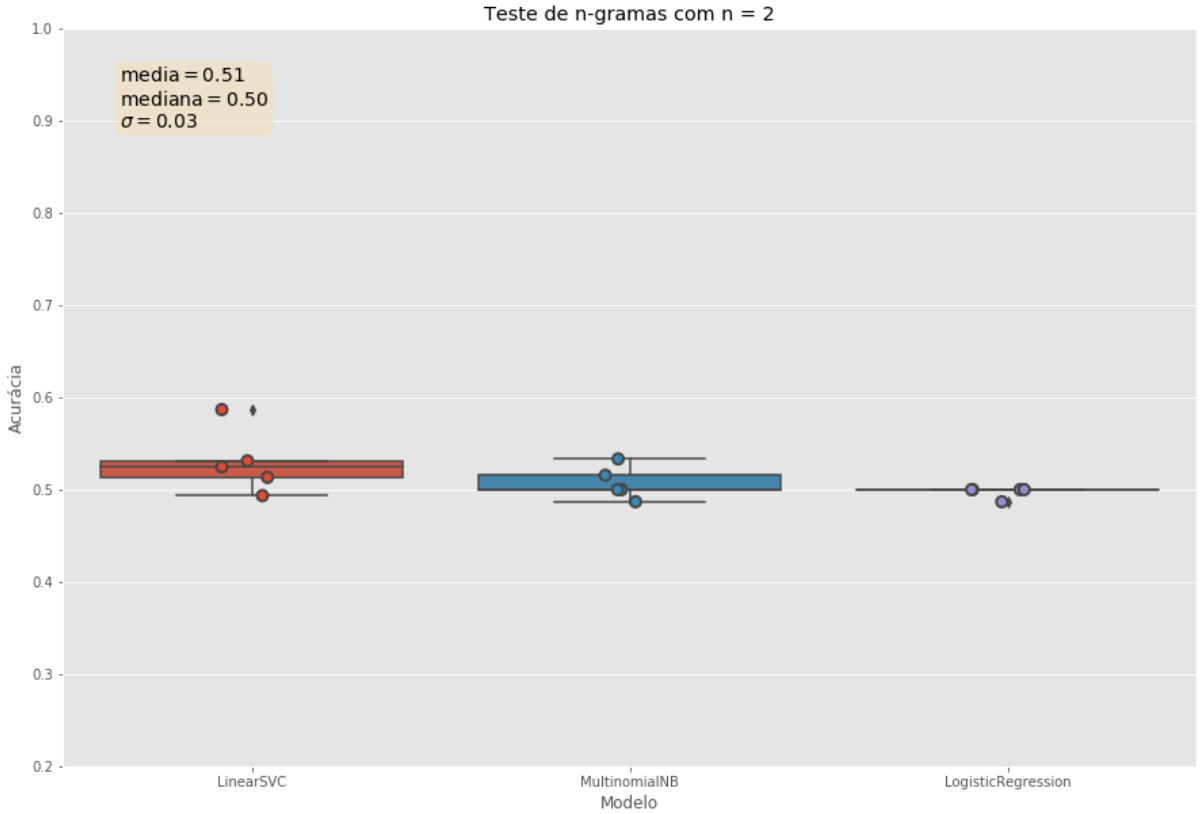

Figura 4.7: *Resultado de acurácias obtidas com bigramas e valoração TF-IDF*

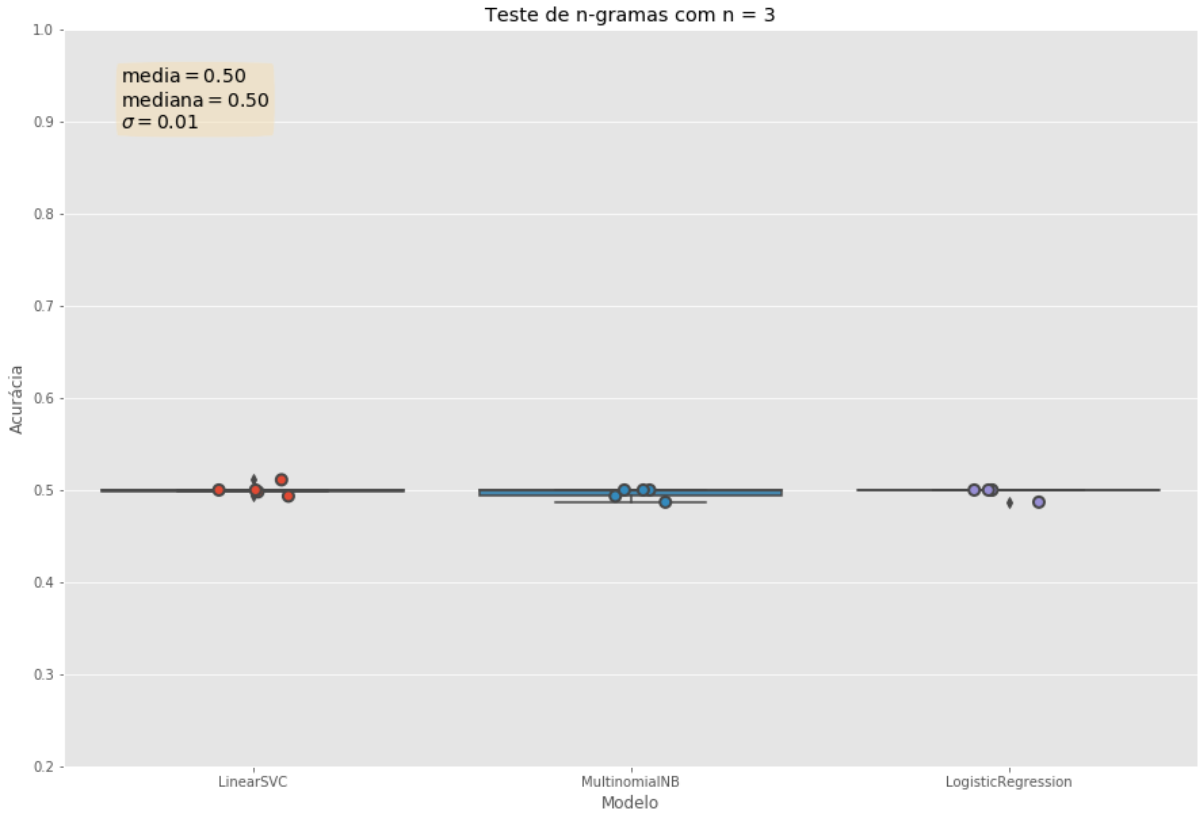

Figura 4.8: *Resultado de acurácias obtidas com trigramas e valoração TF-IDF*

4.5.4 Segundo experimento: busca de hiperparâmetros

É possível fazer uma busca de parâmetros conhecida como *grid-search* para otimizar os parâmetros dos modelos baseados em SVM e regressão logística. Essa otimização foi feita utilizando validação cruzada sobre os dados coletados para evitar a situação de sobreajuste.

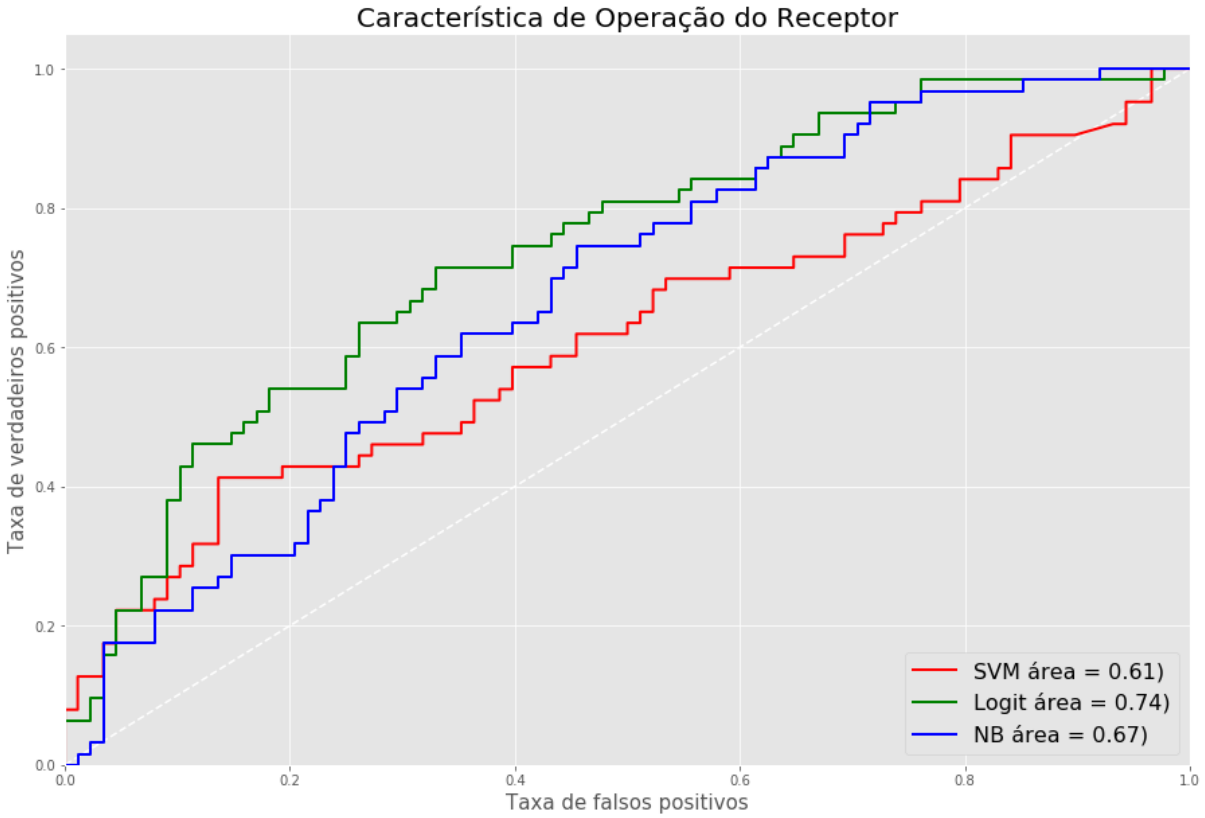

Figura 4.9: *ROC para os algoritmos otimizados*

Resultados das métricas para os algoritmos com os hiperparâmetros otimizados:

Classificador	Precisão	Revocação	F1-Score
SVM	0.71	0.62	0.51
Logit	0.72	0.71	0.69
Naïve Bayes	0.64	0.64	0.64

Tabela 4.2: *Desempenho para os algoritmos básicos*

4.5.5 Terceiro experimento: reamostragem

O baixo desempenho dos algoritmos provavelmente está relacionado com o baixo número de *tweets* classificados e o desbalanceamento das classes como visto na seção 4.3. Para resolver este problema, podemos aplicar as técnicas de reamostragem descritas em 4.3.1 sobre o conjunto de testes para observar se há alguma alteração na performance dos classificadores tem alguma melhora.

Resultados das métricas para os algoritmos com sobreamostragem e sub-amostragem podem ser vistas nas tabelas 4.3 e 4.4, respectivamente.

Classificador	Precisão	Revocação	F1-Score
SVM	0.85	0.81	0.80
Logit	0.73	0.71	0.71
Naïve Bayes	0.71	0.70	0.70

Tabela 4.3: *Desempenho para os algoritmos com sobreamostragem*

Classificador	Precisão	Revocação	F1-Score
SVM	0.63	0.63	0.63
Logit	0.61	0.59	0.59
Naïve Bayes	0.68	0.68	0.68

Tabela 4.4: *Desempenho para os algoritmos com sub-amostragem*

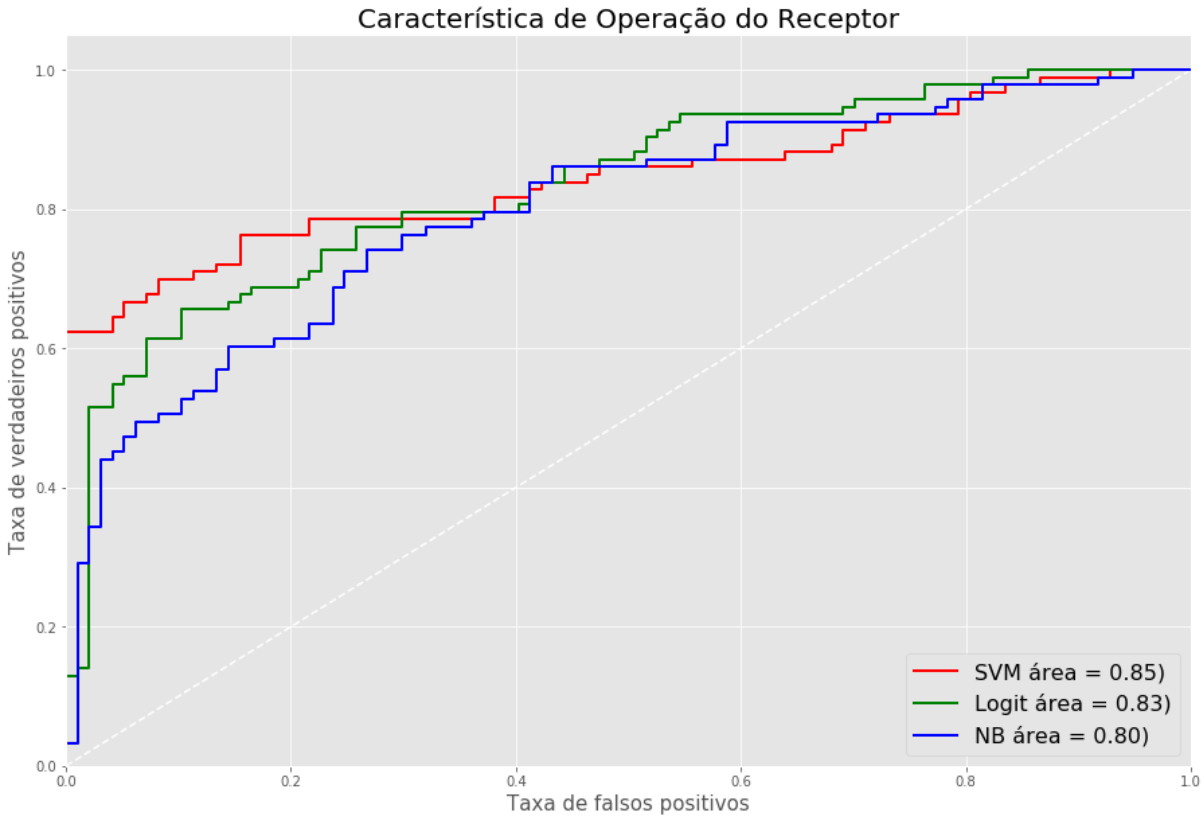

Figura 4.10: *ROC para os algoritmos otimizados e com sobreamostragem*

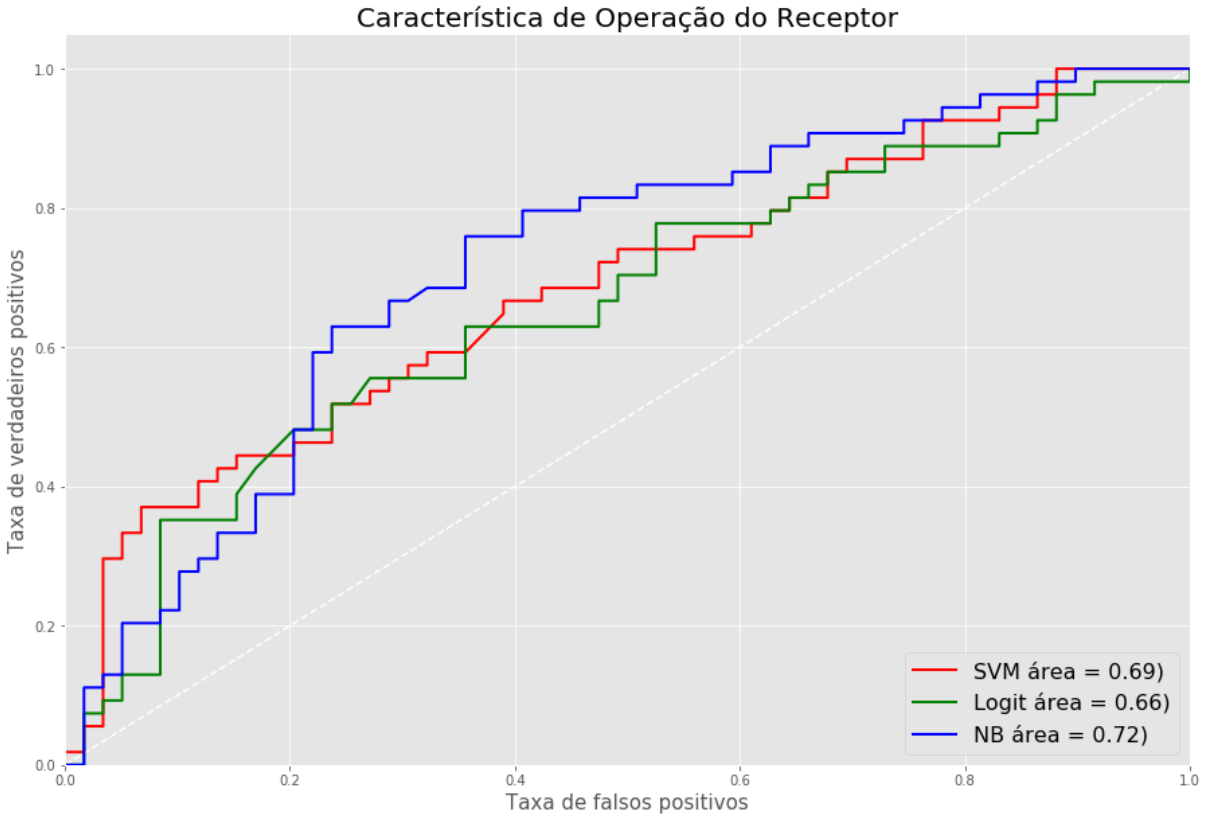

Figura 4.11: *ROC para os algoritmos otimizados e com sub-amostragem*

4.6 Testes fora do corpus

Os algoritmos foram testados em 29602 *tweets* de fora do corpus. Esses *tweets* correspondem ao período da corrida eleitoral até o primeiro turno. Não foram testados *tweets* coletados para o segundo turno das eleições, devido ser um contexto de polarização extrema.

4.6.1 Considerando todos os tweets e algoritmos

Nas tabelas 4.5, 4.6 e 4.7 são mostrados quantos *tweets* foram classificados com opiniões positivas e negativas, considerando algoritmos treinados com o conjunto de treino original e também considerando os conjuntos com reamostragem.

Classificador	Positivos	Negativos
SVM	25466	4136
Logit	20118	9484
Naïve Bayes	19490	10112

Tabela 4.5: *Predições dos algoritmos treinados com dados sem reamostragem*

Classificador	Positivos	Negativos
SVM	25291	4311
Logit	10546	19056
Naïve Bayes	10607	18995

Tabela 4.6: *Predições dos algoritmos treinados com dados que passaram por sobreamostragem*

Classificador	Positivos	Negativos
SVM	8270	21332
Logit	5998	23604
Naïve Bayes	10465	19137

Tabela 4.7: *Predições dos algoritmos treinados com dados que passaram por sub-amostragem*

4.6.2 Análise da opinião sobre os candidatos

Para esta seção, foram separados os *tweets* que continham o nome dos candidatos e foi feita uma predição das polaridade das opiniões a respeito de cada um dos candidatos. Para a classificação, foram utilizados os algoritmos treinados com o conjunto de dados com sobreamostragem, por terem apresentado um desempenho melhor no geral.

Candidato	Positivos	Negativos
Bolsonaro	547	6651
Lula	118	2026
Haddad	64	1897
Marina Silva	7	430
Geraldo Alckmin	1	445
Ciro Gomes	14	901

Tabela 4.8: *Sentimentos identificados sobre os candidatos à presidência da República utilizando SVM*

Candidato	Positivos	Negativos
Bolsonaro	3238	3960
Lula	1053	1091
Haddad	701	1260
Marina Silva	332	105
Geraldo Alckmin	178	268
Ciro Gomes	672	243

Tabela 4.9: *Sentimentos identificados sobre os candidatos à presidência da República utilizando Logit*

Candidato	Positivos	Negativos
Bolsonaro	3392	3806
Lula	1064	1080
Haddad	1019	942
Marina Silva	333	104
Geraldo Alckmin	253	193
Ciro Gomes	680	235

Tabela 4.10: *Sentimentos identificados sobre os candidatos à presidência da República utilizando naïve Bayes*

Nas tabelas 4.8, 4.9 e 4.10 podemos notar que os sentimentos classificados são em grande maioria negativos.

Na classificação utilizando SVM, que tem a melhor performance em todas as métricas, todos os candidatos tem uma proporção desbalanceada entre as classificações positivas e negativas. Esse comportamento do algoritmo, evidencia que um problema de sobreajuste sobre os dados sobreamostrados, como foi alertado na seção 4.3.1.

Na classificação utilizando naïve Bayes e regressão logística a maior parte das predições é negativa, no entanto, a classificação é muito mais balanceada e representa um comportamento mais próximo da realidade. Considerando as métricas e a curva de característica de operação, o algoritmo que apresentou o melhor desempenho para essa situação com dados sobreamostrados foi o algoritmo de regressão logística.

Capítulo 5

Conclusões

Neste trabalho, foi estudado o problema de classificação textual supervisionada envolvendo *tweets* sobre os candidatos à presidência da República.

Para este objetivo ser cumprido, aprendeu-se sobre como é possível coletar dados do *Twitter* e como armazená-los de forma a serem utilizáveis para análise de sentimentos e também como lidar com o problema de ter uma API que fornece uma quantidade limitada de *tweets*.

Durante o período de coleta dos dados, foram notados momentos importantes relacionados às eleições que poderiam interferir na qualidade do conjunto de treinamento. Além disso, também foi notado o problema de que uma classificação manual viesada acarreta em um conjunto de treino e classificador viesado ideologicamente, é necessário que mais de uma pessoa classifique os dados.

Foram feitas definições sucintas sobre os algoritmos de regressão logística, naïve Bayes e SVM. Também foram dadas definições sobre as métricas utilizadas para a avaliação dos classificadores baseados em cada um dos algoritmos.

Ao final deste trabalho, foi realizada uma classificação de *tweets* coletados ao longo do período das campanhas eleitorais. Embora essa tenha sido feita com um conjunto de treinamento pequeno, a classificação apresentou uma boa predição para os sentimentos da população. Foram testados os classificadores e o algoritmo que teve o melhor resultado, baseado nas métricas definidas e considerado o problema de sobreajuste, foi o de regressão logística.

Para melhorar os resultados deste trabalho no futuro, seria interessante buscar uma extração de *features* específicas a cada uma das categorias rotuladas. Uma melhor representação em *features* consequentemente traria melhores resultados nos classificadores.

Apêndice A

Tweet na íntegra

Abaixo, um exemplo de um JSON de *tweet* na íntegra:


```
{
   'contributors': None,
   'coordinates': None,
   'created_at': 'Thu Oct 25 12:59:31 +0000 2018',
   'entities': {
      'hashtags': [],
      'symbols': [],
      'urls': [],
      'user_mentions': [{'id': 3317555339,
         'id_str': '3317555339',
         'indices': [3, 19],
         'name': 'Nada Novo no Front - A Democracia Vencera',
         'screen_name': 'nadanovonofront'}]
    },
   'favorite_count': 0,
   'favorited': False,
   'filter_level': 'low',
   'geo': None,
   'id': 1055443739920031744,
   'id_str': '1055443739920031744',
   'in_reply_to_screen_name': None,
   'in_reply_to_status_id': None,
   'in_reply_to_status_id_str': None,
   'in_reply_to_user_id': None,
   'in_reply_to_user_id_str': None,
   'is_quote_status': False,
   'lang': 'pt',
   'place': None,
   'quote_count': 0,
   'reply_count': 0,
   'retweet_count': 0,
   'retweeted': False,
   'retweeted_status': {
      'contributors': None,
      'coordinates': None,
```

```
 'created_at': 'Thu Oct 25 01:46:04 +0000 2018',
 'entities': {
    'hashtags': [],
    'symbols': [],
'urls': [{
       'display_url': 'twitter.com/i/web/status/1055274259256410
          112',
       'expanded_url': 'https://twitter.com/i/web/status/1055274
          259256410112',
       'indices': [116, 139],
       'url': 'https://t.co/5U99pZlMSi'}],
    'user_mentions': []
 },
 'extended_tweet': {
   'display_text_range': [0, 279],
   'entities': {
       'hashtags': [],
       'symbols': [],
       'urls': [],
       'user_mentions': []
    },
    'full_text': 'O Bolsonaro fez outro texto falando do Kit gay
        porque ele so sabe falar disso. Eh nessa seara que ele
       joga. Nao tem proposta. Nao tem nenhuma ideia real pro
       pais. E ainda sai de "heroi" pro seu publico, por "
       desafiar o TSE", sabendo que nao vai acontecer nada. Ele
       eh esse lixo ai.'
 },
 'favorite_count': 1698,
 'favorited': False,
 'filter_level': 'low',
 'geo': None,
 'id': 1055274259256410112,
 'id_str': '1055274259256410112',
 'in_reply_to_screen_name': None,
 'in_reply_to_status_id': None,
 'in_reply_to_status_id_str': None,
 'in_reply_to_user_id': None,
 'in_reply_to_user_id_str': None,
 'is_quote_status': False,
 'lang': 'pt',
 'place': None,
 'quote_count': 16,
 'reply_count': 25,
 'retweet_count': 523,
 'retweeted': False,
 'source': '<a href="http://twitter.com" rel="nofollow">Twitter
    Web Client</a>',
 'text': 'O Bolsonaro fez outro texto falando do Kit gay porque
```

```
        ele so sabe falar disso. Eh nessa seara que ele joga. Nao
        tem proposta. Nao tem... https://t.co/5U99pZlMSi',
    'truncated': True,
    'user': {
        'contributors_enabled': False,
        'created_at': 'Wed Jun 10 16:24:13 +0000 2015',
        'default_profile': False,
        'default_profile_image': False,
        'description': 'Em uma cruzada contra o brasileiro que se
            acha predestinado.  Eu nÃ£o estou entendendo mais nada.
            FaÃ§o textÃ£o no Twitter. Minha capacidade de sÃntese Ã©
             ridÃcula.',
        'favourites_count': 9708,
        'follow_request_sent': None,
        'followers_count': 38302,
        'following': None,
        'friends_count': 1649,
        'geo_enabled': True,
        'id': 3317555339,
        'id_str': '3317555339',
        'is_translator': False,
        'lang': 'pt',
        'listed_count': 348,
        'location': 'Ilha da Queimada Grande',
        'name': 'Nada Novo no Front - A Democracia VencerÃi',
        'notifications': None,
        'profile_background_color': '000000',
        'profile_background_image_url': 'http://abs.twimg.com/images
            /themes/theme1/bg.png',
        'profile_background_image_url_https': 'https://abs.twimg.com
            /images/themes/theme1/bg.png',
        'profile_background_tile': False,
        'profile_banner_url': 'https://pbs.twimg.com/profile_banners
            /3317555339/1471019342',
        'profile_image_url': 'http://pbs.twimg.com/profile_images/74
            6360887452311552/JrPXQGQY_normal.jpg',
        'profile_image_url_https': 'https://pbs.twimg.com/
            profile_images/746360887452311552/JrPXQGQY_normal.jpg',
        'profile_link_color': '882E33',
        'profile_sidebar_border_color': '000000',
        'profile_sidebar_fill_color': '000000',
        'profile_text_color': '000000',
        'profile_use_background_image': False,
        'protected': False,
        'screen_name': 'nadanovonofront',
        'statuses_count': 122493,
        'time_zone': None,
        'translator_type': 'none',
        'url': 'http://www.facebook.com/nadanovonofront/',
```

```
          'utc_offset': None,
          'verified': False
          }
        },
      'source': '<a href="http://twitter.com/download/android" rel="
         nofollow">Twitter for Android</a>',
      'text': 'RT @nadanovonofront: O Bolsonaro fez outro texto
         falando do Kit gay porque ele so sabe falar disso. Eh nessa
         seara que ele joga. Nao tem proposta. Nao tem pro...',
      'timestamp_ms': '1540472371450',
      'truncated': False,
      'user': {
         'contributors_enabled': False,
         'created_at': 'Sat Sep 23 19:30:20 +0000 2017',
         'default_profile': True,
         'default_profile_image': False,
         'description': '[FanAccount]',
         'favourites_count': 18444,
         'follow_request_sent': None,
         'followers_count': 464,
         'following': None,
         'friends_count': 469,
         'geo_enabled': False,
         'id': 911674111197315075,
         'id_str': '911674111197315075',
         'is_translator': False,
         'lang': 'pt',
         'listed_count': 0,
         'location': 'TWICELAND',
         'name': '-',
         'notifications': None,
         'profile_background_color': 'F5F8FA',
         'profile_background_image_url': '',
         'profile_background_image_url_https': '',
         'profile_background_tile': False,
         'profile_banner_url': 'https://pbs.twimg.com/profile_banners/
            911674111197315075/1538225953',
         'profile_image_url': 'http://pbs.twimg.com/profile_images/104
            6012751968129026/HP9EXAZ7_normal.jpg',
         'profile_image_url_https': 'https://pbs.twimg.com/
            profile_images/1046012751968129026/HP9EXAZ7_normal.jpg',
         'profile_link_color': '1DA1F2',
         'profile_sidebar_border_color': 'C0DEED',
         'profile_sidebar_fill_color': 'DDEEF6',
         'profile_text_color': '333333',
         'profile_use_background_image': True,
         'protected': False,
         'screen_name': 'cmrninthedark',
         'statuses_count': 23052,
```

```
            'time_zone': None,
            'translator_type': 'none',
            'url': None,
            'utc_offset': None,
            'verified': False
        }
}
```

Referências Bibliográficas

Abu-Mostafa *et al.***(2010)** Yaser S. Abu-Mostafa, Malik Magdon-Ismail e Hsuan-Tien Lin. Learning From Data. *Learning from Data*, 21(4):479–481. ISSN 1044-3983. doi: 10.1097/EDE.0b013e3181e13328. Citado na pág. 12

Andranik Tumasjan(2010) Philipp G. Sandner Isabell M. Welpe Andranik Tumasjan, Timm O. Sprenger. Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment. https://www.aaai.org/ocs/index.php/ICWSM/ICWSM10/paper/viewFile/1441/1852//, 2010. Último acesso em 02/11/2018. Citado na pág. 1

Boser *et al.***(1992)** B E Boser, I Guyon e V Vapnik. A. Training Algorithm for Optimal Margin Classifiers. *In Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages:144–152. ISSN 0-89791-497-X. doi: 10.1145/130385.130401. Citado na pág. 13

Broersma e Graham(2012) Marcel Broersma e Todd Graham. Social media as beat: Tweets as news source during the 2010 British and Dutch elections. http://eprints.whiterose.ac.uk/113487/1/SOCIAL_MEDIA_AS_BEAT_Tweets_as_news_sour.pdf, 2012. Último acesso em 02/11/2018. Citado na pág. 1

Cadwalladr e Graham-Harrison(2018) Carole Cadwalladr e Emma Graham-Harrison. Revealed: 50 million Facebook profiles harvested for Cambridge Analytica in major data breach. https://www.theguardian.com/news/2018/mar/17/cambridge-analytica-facebook-influence-us-election, 2018. Último acesso em 25/10/2018. Citado na pág. 5

DAPP-FGV(a) DAPP-FGV. Primeiro mês de campanha nas redes é marcado por liderança isolada de Bolsonaro e crescimento de Haddad e Ciro. https://static.poder360.com.br/2018/09/19-09-1-mes-de-campanha-nas-redes.pdf, a. Último acesso em 09/10/2018. Citado na pág. 17

DAPP-FGV(b) DAPP-FGV. Robôs, redes sociais e política: Estudo da FGV/DAPP aponta interferências ilegítimas no debate público na web. http://dapp.fgv.br/robos-redes-sociais-e-politica-estudo-da-fgvdapp-aponta-interferencias-ilegitimas-no-debate-pub b. Último acesso em 09/10/2018. Citado na pág. 19

DAPP-FGV(c) DAPP-FGV. Com 4,85 milhões de tuítes, eleição presidencial é evento com maior impacto nas redes. https://observa2018.com.br/posts/com-4-8-milhoes-tuites-eleicao-presidencial-evento-maior-impacto-redes-brasil/, c. Último acesso em 09/10/2018. Citado na pág. 18

DAPP-FGV(d) DAPP-FGV. Atos contra Bolsonaro geram 1,4 milhão de menções; apoio, 1 milhão. https://observa2018.com.br/posts/

manifestacoes-contra-bolsonaro-provocam-14-milhao-de-mencoes-no-twitter-atos-de-apoio-ac
d. Último acesso em 09/10/2018. Citado na pág. 18

DAPP-FGV(e) DAPP-FGV. Ataque com faca a Jair Bolsonaro gera 3,2 milhões de menções em 16h. https://observa2018.com.br/posts/ataque-com-faca-a-jair-bolsonaro-gera-32-milhoes-de-mencoes-em-16-horas/, e. Último acesso em 15/9/2018. Citado na pág. 18

Folha(2018) Redação Folha. Tuítes da Folha sobre "bolso" e "bolovo" ativam ação de robôs, que saem em defesa de Bolsonaro. https://www.revistaforum.com.br/tuites-da-folha-sobre-bolso-e-bolovo-ativam-acao-de-robos-que-saem-em-defesa-de-bolsonaro 2018. Último acesso em 20/10/2018. Citado na pág. 19

G. Chowdhury(2003) Gobinda G. Chowdhury. Natural Language Processing. https://strathprints.strath.ac.uk/2611/1/strathprints002611.pdf, 2003. Último acesso em 02/11/2018. Citado na pág. 1

Helfstein(2018) Lucas Helfstein. Resultados. https://github.com/lucashelfs/mac0499/blob/master/c%C3%B3digos/classificador/Classificador%20.ipynb, 2018. Último acesso em 02/11/2018. Citado na pág. 22

Hous(2018) Débora Sögur Hous. Militantes usam apelidos para evitar encontros de 'bolhas' na internet. https://www1.folha.uol.com.br/poder/2018/09/militantes-usam-apelidos-para-evitar-encontros-de-bolhas-na-internet.shtml, 2018. Último acesso em 11/10/2018. Citado na pág. 18

Rennle *et al.***(2003)** Jason D M Rennle, Lawrence Shih, Jamie Teevan e David R Karger. Tackling the Poor Assumptions of Naive Bayes Text Classifiers. *Proceedings of the Twentieth International Conference on Machine Learning*, (1973):1–8. ISSN 14773155. doi: 10.1186/1477-3155-8-16. URL papers2://publication/uuid/1782FD93-5A92-46C7-92FF-A68D3637F224. Citado na pág. 10

Romão(2018) Lucas Romão. manual-classifier-helper. https://github.com/romaolucas/manual-classifier-helper, 2018. Último acesso em 02/11/2018. Citado na pág. 7

Sarkar(2016) Dipanjan Sarkar. *Text Analytics with Python*. ISBN 978-1-4842-2387-1. doi: 10.1007/978-1-4842-2388-8. URL http://link.springer.com/10.1007/978-1-4842-2388-8. Citado na pág. 9

Tweepy(2018) Tweepy. Streaming with tweepy. http://docs.tweepy.org/en/v3.4.0/streaming_how_to.html, 2018. Último acesso em 23/10/2018. Citado na pág. 5

Twitter(2018) Twitter. Search Tweets: Overview. https://developer.twitter.com/en/docs/tweets/search/overview, 2018. Último acesso em 30/6/2018. Citado na pág. 5

Walaa Medhat(2015) Hoda Korash Walaa Medhat, Ahmed Hassan. Sentiment analysis algorithms and applications: A survey. http://kt.ijs.si/markodebeljak/Lectures/Seminar_MPS/2012_on/Seminars_2015_16/Simon%20Brmez/Bibliography/%5B5%5D%20Sentiment%20analysis%20algorithms%20and%20applications%20A%20survey.pdf, 2015. Último acesso em 05/11/2018. Citado na pág. 3

We Are Social(a) We Are Social. Global digital report 2014. https://www.slideshare.net/wearesocialsg/social-digital-mobile-around-the-world-january-2014/53-JAN2014BRAZIL_DATA_SNAPSHOT20100962285l5TOTAL_POPULATIONURBANRURAL99357737INTERNET_USERS86000000ACTIVE, a. Último acesso em 20/10/2018. Citado na pág. 4

We Are Social(b) We Are Social. Global digital report 2018. https://digitalreport.wearesocial.com/, b. Último acesso em 20/10/2018. Citado na pág. 1, 4

Zhang e Liu(2012) Lei Zhang e Bing Liu. Sentiment Analysis and Opinion Mining. *Encyclopedia of Machine Learning and Data Mining*, (May):1–10. ISSN 1947-4040. doi: 10.1007/978-1-4899-7502-7_907-1. URL https://www.cs.uic.edu/~liub/FBS/SentimentAnalysis-and-OpinionMining.pdf. Citado na pág. 3, 4