Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Spatio-Temporal Trend Analysis of the Brazilian Elections based on Twitter Data

Bruno Justino Garcia Praciano

Monografia apresentada como requisito parcial
para conclusão do Curso de Engenharia da Computação

Orientador
Prof. Dr. -Ing João Paulo Lustosa da Costa

Coorientador
MSc. João Paulo de Abreu Maranhão

Brasília
2018

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Spatio-Temporal Trend Analysis of the Brazilian Elections based on Twitter Data

Bruno Justino Garcia Praciano

Monografia apresentada como requisito parcial
para conclusão do Curso de Engenharia da Computação

Prof. Dr. -Ing João Paulo Lustosa da Costa (Orientador)
ENE/UnB

Prof. Dr. Rafael Timóteo de Sousa Júnior
ENE/UnB

Dr. -Ing Ricardo Kerhle Miranda
ENM/UnB

Prof. Dr. José Edil Guimarães de Medeiros
Coordenador do Curso de Engenharia da Computação

Brasília, 13 de novembro de 2018

Dedicatória

Na *dedicatória* o autor presta homenagem a alguma pessoa (ou grupo de pessoas) que têm significado especial na vida pessoal ou profissional. Por exemplo (e citando o poeta): *Eu dedico essa música a primeira garota que tá sentada ali na fila. Brigado!*

Agradecimentos

Nos *agradecimentos*, o autor se dirige a pessoas ou instituições que contribuíram para elaboração do trabalho apresentado. Por exemplo: *Agradeço aos gigantes cujos ombros me permitiram enxergar mais longe. E a Google e Wikipédia.*

Resumo

O *resumo* é um texto inaugural para quem quer conhecer o trabalho, deve conter uma breve descrição de todo o trabalho (apenas um parágrafo). Portanto, só deve ser escrito após o texto estar pronto. Não é uma coletânea de frases recortadas do trabalho, mas uma apresentação concisa dos pontos relevantes, de modo que o leitor tenha uma ideia completa do que lhe espera. Uma sugestão é que seja composto por quatro pontos: 1) o que está sendo proposto, 2) qual o mérito da proposta, 3) como a proposta foi avaliada/validada, 4) quais as possibilidades para trabalhos futuros. É seguido de (geralmente) três palavras-chave que devem indicar claramente a que se refere o seu trabalho. Por exemplo: *Este trabalho apresenta informações úteis a produção de trabalhos científicos para descrever e exemplificar como utililzar a classe LaTeX do Departamento de Ciência da Computação da Universidade de Brasília para gerar documentos. A classe `UnB-CIC` define um padrão de formato para textos do CIC, facilitando a geração de textos e permitindo que os autores foquem apenas no conteúdo. O formato foi aprovado pelos professores do Departamento e utilizado para gerar este documento. Melhorias futuras incluem manutenção contínua da classe e aprimoramento do texto explicativo.*

Palavras-chave: Big Data, Aprendizado de Máquina Supervisionado, Análise de Sentimentos, Máquina de Vetor de Suporte

Abstract

O *abstract* é o resumo feito na língua Inglesa. Embora o conteúdo apresentado deva ser o mesmo, este texto não deve ser a tradução literal de cada palavra ou frase do resumo, muito menos feito em um tradutor automático. É uma língua diferente e o texto deveria ser escrito de acordo com suas nuances (aproveite para ler `http://dx.doi.org/10.6061%2Fclinics%2F2014(03)01`). Por exemplo: *This work presents useful information on how to create a scientific text to describe and provide examples of how to use the Computer Science Department's LaTeX class. The `UnB-CIC` class defines a standard format for texts, simplifying the process of generating CIC documents and enabling authors to focus only on content. The standard was approved by the Department's professors and used to create this document. Future work includes continued support for the class and improvements on the explanatory text.*

Keywords: Big Data, Supervised Machine Learning, Sentiment Analysis, Support Vector Machine

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

AM Aprendizado de Máquina.

API Aplication Programming Interface.

IBGE Instituto Brasileiro de Geografia e Estatística.

ICDM International Conference on Data Mining.

IEEE Instituto de Engenheiros Eletricistas e Eletrônicos.

PLN Processamento de Linguagem Natural.

SVM Support Vector Machine.

TM Text Mining.

Capítulo 1

Introdução

Com a popularização da internet tem revolucionado as sociedades com o passar do tempos, pois agora é possível conectar várias pessoas, e realizar trocar de informações em tempo real e com o custo muito baixo em relação aos veículos tradicionais de mídia [1].

As notícias tem sido compartilhadas de maneira muito rápida e eficente e com a utilização massiva das redes de relacionamento, as pessoas podem trocar ideias e opiniões acerca de determinado assunto e com isso facilitar o acesso de todos. Com o passar dos anos as redes sociais já fazem parte da vida de várias pessoas, e com isso as relações interpessoais modificaram-se e esse mundo tem gerado muitos dados de fácil e livre acesso [2].

Com as redes sociais é possível comunicar-se com pessoas de diversas nacionalidades e características, com a grande amplitude que essas alcançam, o volume de dados é algo imensurável e também é uma fonte de dados inesgotável. Com todo esse volume de informações, o ambiente torna-se atrativo para aplicar técnicas de aprendizado de máquina e outros tipos de análises.

1.1 Motivação

De acordo com o IBGE, no Brasil mais de 116 milhões de pessoas tem acesso a internet, ou seja, grande parte da população está expressando suas ideias de forma livre nas redes sociais. E como as eleições em um evento muito importante em qualquer democracia, realizar análise de sentimentos nos textos provenientes de redes sociais se tornam cada vez mais atrativos. O Brasil ocupa a 4ª no ranking de países com a quantidade de pessoas com acesso a internet[3].

Na Figura 1.1 é possível visualizar que grande parte do conteúdo disponível na internet está em inglês, ou seja, é por esse motivo que existem dicionários léxicos para atividades

de TM, como é o caso do WordNet [4], uma das maiores bases de dados do mundo para essa atividade.

As pesquisas utilizando TM com o idioma português ainda são recentes e poucos exploradas, pois a maioria das ferramentas são desenvolvidas para atender o idioma inglês. Mas a análise de sentimento em comentários de redes sociais, pode ser útil em diversas áreas como venda de um produto, avaliação de um estabelecimento, marketing e até mesmo realizar predições de eleições que é o objetivo desse trabalho.

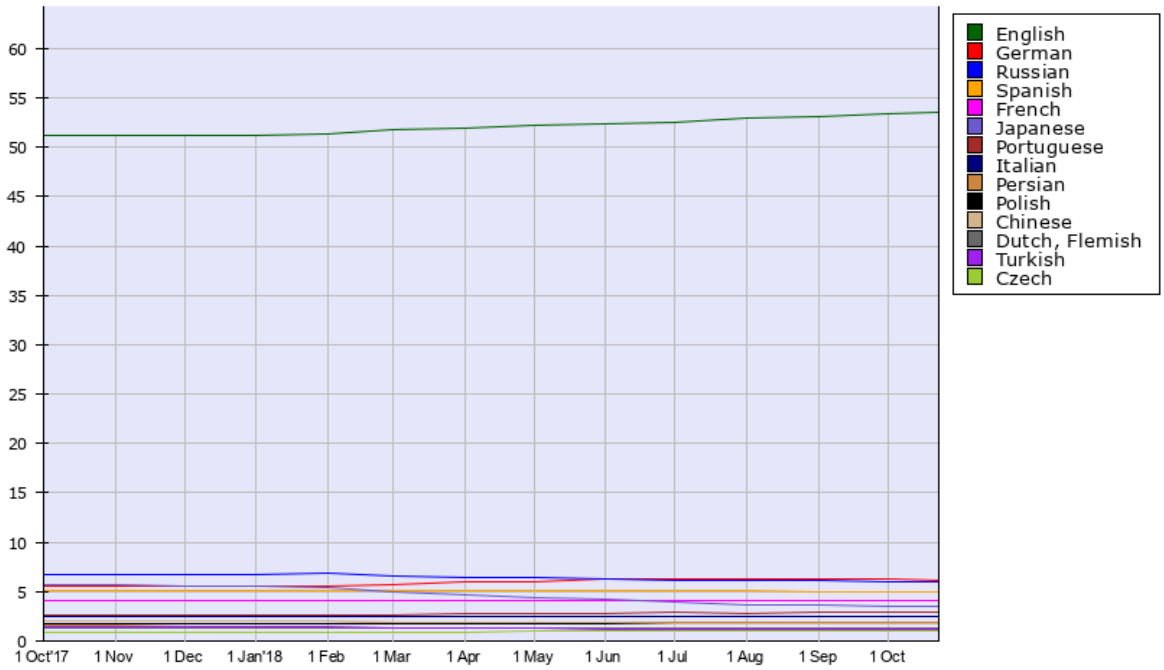

Figura 1.1: Idiomas utilizados em conteúdos disponíveis na internet [5].

1.2 Problemas

Existem várias redes sociais em funcionamento, e cada uma tem o foco diferente, por exemplo, o *Twitter* é uma rede social com o intuito de formação de opinião, pois grande parte dos usuários a utilizam para compartilhar texto pequenos de até 140 caracteres e também apresenta uma API aberta, mas essa possui limitação em relação ao número de requisições e também ao período que pode efetuar uma busca, que atualmente são de 14 dias [6].

A rede social mais utilizado no Brasil é o *Facebook*, que tem mais de 120 milhões de usuários ativos no Brasil[7], mas após os escândalos das eleições americanas que a envolveram [8] foram adotadas inúmeras medidas de segurança para evitar a extração de

dados da plataforma, atualmente para utilizar a API da empresa é necessário ser aprovado e também conta com o número limitado de requisições que podem ser feitas por cada token de segurança.

É importante citar que os dicionários em língua portuguesa para realizar esse tipo de atividades ainda são pouco desenvolvidos, para esse trabalho foi utilizado dois dicionários em conjunto para melhorar os resultados. O OpLexicon [9] foi combinado com o Sentilex [10], pois atualmente são os melhores dicionários abertos em português para realizar análise de sentimentos, também foi utilizada uma biblioteca chamada TextBlob [11] que é necessário realizar a tradução para o inglês, o que acaba ocasionando um viés na etapa de processamento dos dados.

1.3 Objetivos

O objetivo desse trabalho é criar uma forma de predição e um ambiente que expresse a opinião dos usuários da rede social *Twitter* aplicando em textos curtos técnicas de AM, o intuito principal desse trabalho é a utilização do framework proposto em textos que falam sobre políticos que estão concorrendo a cargos eletivos, mas com a inserção de outros textos na fase de treinamento do modelo de AM é possível ampliar as opções e realizar diversas análises para distintas áreas.

1.4 Trabalho Publicado

Durante o desenvolvimento desse trabalho de graduação, foi desenvolvido um artigo científico cujo o intuito era validar a ideia e verificar se apresentava algum tipo de inovação, onde foi apresentado os primeiros resultados do trabalho.

O artigo publicado é entitulado *Spatio-Temporal Trend Analysis of the Brazilian Elections based on Twitter Data* foi aceito em uma das maiores conferências de mineração de dados do mundo, o IEEE ICDM, e foi aceito para apresentação oral na seção de análise de sentimentos.

B. J. G. Praciano, J. P. C. L. da Costa, J. P. A. Maranhão, J. B. Prettz, R. T. de Sousa Jr. e F. Mendonça, "Spatio-Temporal Trend Analysis of the Brazilian Elections based on Twitter Data," 2018 IEEE International Conference on Data Mining (ICDM).

1.5 Trabalhos Relacionados

Em [12] os autores definiram o conceito de análise de sentimento em vários níveis e também utilizaram algoritmos que realizam o reconhecimento de entidades, que é a

detecção de nomes próprios e com isso remover esses substantivos da análise de sentimento para que o os resultados sejam melhores. O dicionário léxico utilizado para a classificação dos textos provenientes do *Twitter* foi o SentiWordNet, e as polaridades utilizadas nesse trabalho foram três: Positivo, Negativo e Neutro. Foi abordado duas paradigmas de AM, o supervisionado e o não-supervisionado e com o isso o melhor resultado foi de 90% na utilização de algoritmos supervisionados.

No artigo [13] foi utilizado um filtro baseado na mineração de opiniões, que é uma das áreas de análise de sentimentos. Foi utilizadas técnicas de decomposição tensorial para capturar interações intrínsecas, pois como o dado é multidimensional, o autor dividiu entre usuários, filmes e outros aspectos, com a aplicação dessas técnicas, houve uma grande redução no esforço computacional do computador na parte de análise de sentimentos, pois o *dataset* utilizado foi reduzido após a decomposição tensorial.

Em [1] os autores aplicaram TM nos dados provenientes do *Twitter* que citavam as eleições presidencias de 2012 da Coréia do Sul. Foram utilizadas distintas técnicas: *topic modeling* para acompanhar as mudanças nos assuntos mais falados do rede sociais, técnicas de análise de rede foram utilizadas para verificar quais pessoas eram citadas e por quem. Os resultados sugeriram que o *Twitter* pode ser um aliado para detectar as mudanças no contexto social enquanto são analisados o texto de quem escreveu.

Em [14] é proposto um sistema para acompanhar as eleições francesas através de tópicos escritos no *Twitter* através da análise de sentimentos. Os resultados obtidos convergiram com os resultados divulgados pelas autoridades da França e foram associados as mudanças de popularidade dos candidatos após a eleição.

Em [15], o dataset utilizado nesse trabalham foi o da eleição colombiana de 2014, técnicas de aprendizado supervisionado foram implementadas e também foram rotuladas previamente usuários que seriam spam. Foi implementado um sistema com o objetivo de investigar o potencial que uma rede social tem de interferir em uma votação, e de acordo com os resultados obtidos, foi possível afirmar que os dados utilizados não foram consistentes.

Em [16] foi usado um dicionário léxico e apenas o algoritmo Naïve Bayes para calcular o sentimento de *tweets* que foram coletados 100 dias antes da eleição americanas de 2016. Os autores classificaram manualmente os textos extraídos do *Twitter*. Os resultados obtids sugerem que essa rede social pode ser considerada ao realizar trabalhos com esse intuito.

1.6 Descrição dos capítulos

O presente trabalho é apresentado com a seguinte estrutura:

- Capítulo 2: Conceitos em Machine Learning e Mineração de Texto. Apresenta o conjunto de técnicas e metologias que foram necessárias para o desenvolvimento desse trabalho.
- Capítulo 3: Framework proposto. Discorre sobre a metodologia empregada no trabalho e ilustra todos os passos seguidos e necessários para entendidmento do modelo.
- Capítulo 4: Resultados. Nesse capítulo são apresentados os resultados obtidos com a utilização do modelo de aprendizado de máquina proposto e também uma breve justificativa a escolha das ferramentas.
- Capítulo 5: Conclusão. As conclusões sobre o tema são expostas.
- Capítulo 6: Trabalhos Futuros. Na última seção são discutidos quais temas que foram levantados que podem ser aprofundados.

Capítulo 2

Conceitos em Aprendizado de Máquina e Mineração de Texto

Neste capítulo é feita uma introdução sobre o AM, tema principal dessa monografia. E está dividido da seguinte maneira na seção 2.1 são apresentados os conceitos básicos para o entendimento inicial do tema. Na seção 2.2 são apresentados os conceitos de mineração de texto. Na seção 2.3 é apresentado o conceito de análise de sentimentos e a sua aplicação.

2.1 Aprendizado de Máquina

2.1.1 Conceitos Básicos

Com o alto volume de informações geradas no dia a dia, a utilização de algoritmos que sejam capazes de identificar padrões tornam-se cada vez mais necessários, pois em diversas empresas e órgãos do governo não são capazes de analisar todas as informações existentes ou entregar de maneira célere [17].

Portanto para realizar esse tipo de atividade é necessário entender o problema existente e também ter um volume de dados razoável para realizar o treinamento do modelo, e com a técnica de AM é possível automatizar a construção de sistemas inteligentes que podem ser ajustados de acordo com a necessidade de cada tarefa [18].

Em outras palavras o AM é um conjunto de regras, que possibilitam uma máquina a tomar decisões baseadas em experiências passadas ao invés de um software que teve que ser definido anteriormente como tratar cada tipo de regra, também existe

a possibilidade desses modelos serem desenvolvidos e melhorarem quando expostos a novos dados [19].

O conceito de AM pode ser sintetizado como a capacidade de um programa de computador aprender com a experiencia (E) relacionada a alguma classe de tarefas (T), baseada em uma medida de desempenho (P). Dessa forma, o desempenho em tarefas (T), quando medido por (P), melhora com a experiencia em (E) [20]

Essas técnicas tem sido utilizadas amplamente para resolução de diversos problemas, atualmente o assunto está em alta e existem diversas oportunidade para colocar em prática a matemática que foi desenvolvida para a criação desses algoritmos. Gigantes da indústria tem investido severamente para o desenvolvimento de novas tecnologias, como os veículos autônomos, robôs advogados, veículos não tripulados e na identificação de doenças.

Existem distintos paradigmas para ensinar uma máquina, esse trabalho irá focar apenas nos tipos de AM que estão sendo utilizados no desenvolvimento do framework, portanto não será abordados temas como aprendizado estatístico ou redes bayesianas.

2.1.2 Paradigmas de Aprendizado de Máquina

Os principais tipos de AM utilizados atualmente serão detalhados nas 3 subseções a seguir:

2.1.2.1 Aprendizado Supervisionado

O aprendizado supervisionado é uma das formas em ensinar uma tarefa para a máquina, onde existe uma entrada e uma saída desejada que já foi anteriormente rotulada, esse processo pode ser feito de forma manual ou utilizar de dicionários, quando a informação de entrada é um texto. Tendo os dados detalhados a máquina é capaz de classificar novas entradas a partir de experiências antigas [20]. Na Figura 2.1 é possível visualizar a forma que é realizada etapa de treinamento do modelo, onde é fornecida uma informação, juntamente com o que ela significa, pois esse tipo de aprendizado é necessário rotular todas as informações que serão utilizadas para que a máquina consiga distinguir o que é gato e o que é cachorro.

Na Figura 2.2 é possível visualizar o modelo preditivo em funcionamento, onde o usuário fornece uma informação de entrada e o sistema o responde com a classe que foi identificada através da entrada.

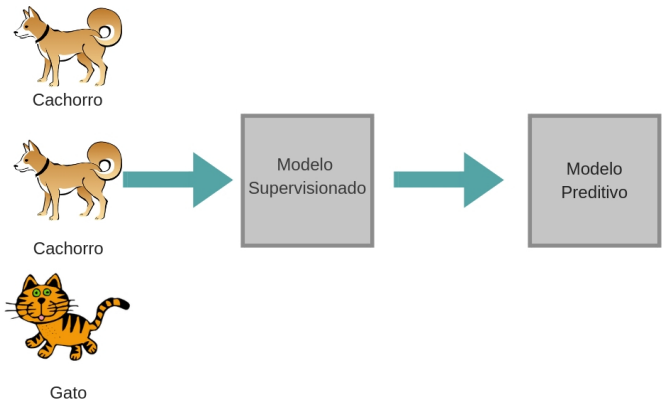

Figura 2.1: Treinamento de um modelo de aprenzidado de máquina supervisionado.

Figura 2.2: Utilização do modelo para realizar predição a partir de classes previamente treinadas.

Algum dos algoritmos mais utilizados para esse paradigma serão detalhados durante essa subseção.

2.1.2.1.1 SVM A máquina de vetor de suportes, do inglês support vector machine (SVM), faz parte do aprendizado supervisionado, com ele é possível classificar grupos de dados separando as suas margens. Essas margens são delineadas pela fração dos dados de treinamento, são chamadas de vetores de suporte [21].

As vantagens de utilizar SVM são [22]:

- Efetivos em espaços multidimensionais
- Continua eficaz em casos onde o número de dimensões é maior que o número de amostras.
- Utiliza os vetores de suporte, que otimiza o uso de memória do computador.
- É possível utilizar diversos kernels para a função de decisão.

Como desvantagens, temos [22]:

The disadvantages of support vector machines include:

- Se o número de entradas for muito maior que o número de amostras, pois existe a possibilidade de overfitting.

O SVM é construido em um hiper-plano ou em vários hiper-planos em um espaço de infinitas dimensões, que podem ser usadas classificação ou regressão. Na Figura 2.3 é detalhado um hiper-plano que mostra uma boa separação das variáveis que possui a maior distância até os pontos dos dados de treinamento mais próximos de qualquer classe, pois é conhecido no SVM que quanto maior for a margem, menor será o erro do classificador[23].

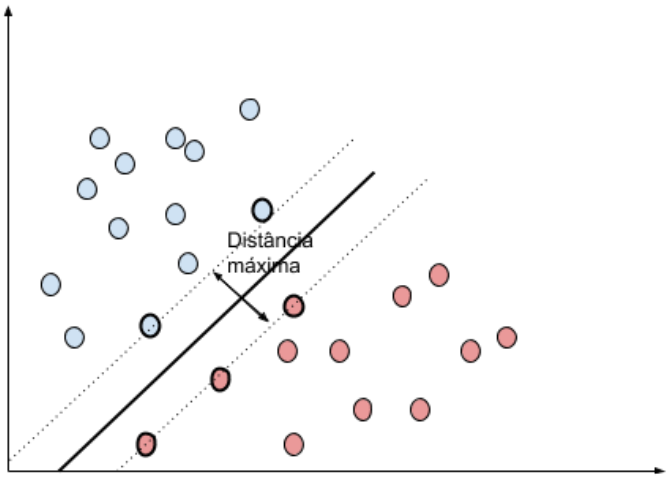

Figura 2.3: Exemplo de vetores de suporte com dimensão 2.

Nesse trabalho foi utilizado esse algoritmo para classificar dados multi-classes, pois foi utilizadas três classes distintas, para desenvolvimento do modelo: Positivo, Negativo e Neutro.

Tendo como os dados de treinamento $x_i \in R^p$, onde i=1,$\cdots$, n.*Onde* y $\in \{1, -1\}^n$, na Equação 2.1 é possível visualizar a solução matemática para esse algoritmo.

$$\begin{aligned} &\min_{\omega,\beta,\zeta} \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{n}\zeta_i \\ &\text{sujeito a } y_i\left(\omega^T\phi\left(x_i\right) + b\right) \geq 1 - \zeta_i, \\ &\zeta_i \geq 0, i = 1, \cdots, n \end{aligned} \tag{2.1}$$

A partir do espaço de Hilbert, temos que o dual de um espaço de Hilbert é um espaço de Hilbert [24],

na Equação 2.2, temos:

$$\begin{aligned} &\min_{\alpha} \frac{1}{2}\alpha^T Q\alpha - e^T\alpha \\ &\text{sujeito } ay^T\alpha = 0 \\ &0 \leq \alpha_i \leq C, i = 1 \cdots, n \end{aligned} \tag{2.2}$$

onde e é o vetor com todos os valores, $C > 0$ é o limite superior, Q é uma matriz semidefinida positiva de tamanho n por n, $Q_{ij} \equiv y_i y_j K(x_i, x_j)$, onde $K(x_i, x_j) = \phi(x_i)^T\phi(x_j)$ é a função kernel do SVM. Onde os vetores de treinamento são definidos em um espaço dimensional maior pela função ϕ. E por fim na Equação 2.3, temos a função de decisão:

$$sgn(\sum_{i=1}^{n} y_i\alpha_i K(x_i, x) + \rho) \tag{2.3}$$

Onde o termo $y_i\alpha_i$ representa o coeficiente dual, $K(x_i, x)$ são os vetores de suporte e ρ é um termo independente.

2.1.2.1.2 Naive Bayes O Naive Bayes é um outro algoritmo AM supervisionado, a sua formulação matemática é sustentada pelo teorema estatístico de Bayes com um pouco de ingenuidade, pois assume-se que a suposição de independência condicional entre cada par de recursos, dado o valor da variável de classe [25]. O teorema de Bayes clássico segue a seguinte relação, onde um termo independente y e o vetor x_1 até x_n, na Equação 2.4

$$P\left(y \mid x_1, \cdots, x_n\right) = \frac{P(y)P(x_1, \cdots, x_n \mid y)}{P(x_1, \cdots, x_n)} \tag{2.4}$$

Na Equação 2.5 é desconsiderado completamente a correlação entre as variáveis,

$$P\left(x_i \mid y, x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n\right) = P(x_i|y), \tag{2.5}$$

Na Equação 2.6, para todo i, é possível simplificar através da relação:

$$P(y|x_1, \cdots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1, \cdots, x_n)}, \tag{2.6}$$

Sabendo que $P(x_1, \cdots, x_n)$ é constante dada a entrada, podemos usar a seguinte regra de classificação, na Equação 2.7:

$$\widehat{y} = \arg\max_{y} P(y) \prod_{i=1}^{n} P(x_i|y), \tag{2.7}$$

O classificador Naive Bayes pode ser extremamente rápido em comparação a outros algoritmos mais sofisticados, pois existe o desacoplamento das distribuições de características condicionais de classe que significa que cada uma pode ser estimada independentemente com uma distribuição unidimensional[26].

2.1.2.1.3 Árvores de Decisão A árvore de decisão funciona quando as classes presentes no conjunto de dados de treinamento se dividem, ou seja, esse algoritmo seleciona um problema mais complexo e divide em subproblemas mais simples e de forma recursiva a mesma estratégia é a aplicada a cada subproblema [27].

Na Figura 2.4, é um exemplo de como uma árvore de decisão poderia realizar uma análise de crédito de forma rápida.

Tendo como entrada os vetores de treinamento $x_i \in R^n$, $i = 1, \cdots, n$, e o vetor com os rótulos de saída $y \in R^n$, a árvore de decisão divide recursivamente o espaço de forma que as amostras com os mesmos rótulos sejam agrupadas[28].

Considerando os dados no nó m que pode ser representado pela letra Q. Para cada candidato é dividido $\theta = (j, t_m)$ consistindo a entrada como j e o *threshold* t_m, dividindo os dados entre $Q_{left}(\theta)$ e $Q_{right}(\theta)$ e esses subconjuntos pode ser vistos na Equação 2.8

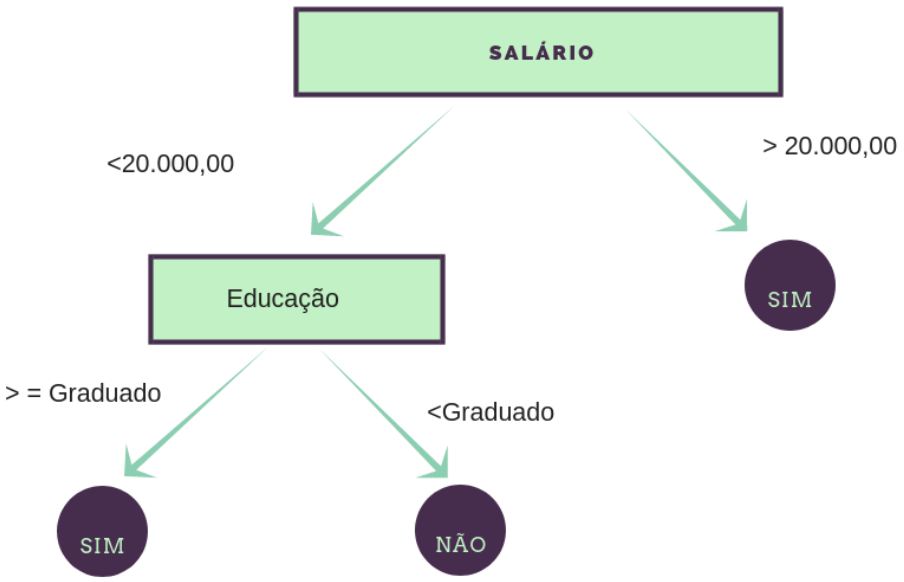

Figura 2.4: Árvore de decisão para análise de crédito de um cliente que deseja empréstimo alto no banco, levando em consideração a educação e faixa salarial.

$$\begin{aligned} Q_{left}(\theta) &= (x, y)|x_j \leq t_m \\ Q_{right}(\theta) &= Q \backslash Q_{left}(\theta) \end{aligned} \tag{2.8}$$

A impureza no nó apresentado é calculada utilizando uma função de impureza denotada de $K()$, que pode ser escolhida dependendo se o problema for de classificação ou regressão. Na Equação 2.9 é calculada essa impureza:

$$G(Q, \theta) = \frac{n_{left}}{N_m} K(Q_{left}(\theta)) + \frac{n_{right}}{N_m} K(Q_{right}(\theta)) \tag{2.9}$$

Por fim, na Equação 2.10 são selecionados os parâmetros para diminuir a impureza,

$$\theta = \arg\min_{\theta} G(Q, \theta) \tag{2.10}$$

É possível gerar outros nós a partir dos subconjuntos de dados gerados para direita e para esquerda, até que a profundidade máxima permitida seja atingida, $N_m < \min_{samples}$ ou $N_m = 1$.

2.1.2.1.4 Regressão Logística Apesar do nome, é um modelo de classificação linear e não de regressão como sugere o seu nome. Neste modelo, os dados são descritos, através de um único teste e são modeladas a partir de uma função logística [17].

Seja a probabilidade P(X), na Equação 2.11

$$P(x) = \frac{1}{1 + e^{-(B_0 + B_1 x)}} = \frac{e^{B_0 + B_1} x}{1 + e^{B_0 + B_1} x}$$

$$\frac{p(x)}{1 - p(x)} = e^{B_0 + B_1 x} \tag{2.11}$$

Onde, $_i$ são calculados através da função de máxima verossimilhança.

2.1.2.2 Aprendizado não-supervisionado

Nesse tipo de aprendizado não existe a necessidade de um vetor contendo os rótulos de cada medida, ou seja, não é necessário avisar a máquina o tipo de cada dado. A essa não inserção de rótulos faz com que o algoritmo aprenda de acordo com as características dos dados de entrada. Esse tipo de paradigma não tem certos benefícios que existem no aprendizado supervisionado, para o seu funcionamento, o modelo propõe hipóteses a partir do que foi inserido na entrada [18]. Esse tipo de algoritmo é muito interessante em base de dados que ainda não foram exploradas e que necessita-se visualizar os dados agrupados.

A Figura 2.5 é ilustrado o exemplo que foi exposto ao falar sobre aprendizado supervisionado, mas agora levando em consideração o não-supervisionado. E é possível ver que o modelo foi capaz de agrupar as características semelhantes que os cachorros têm e agrupá-los em um único cluster e também foi formado um outro cluster contendo apenas o gato, pois não tem características semelhantes aos cães.

2.1.2.2.1 K-means O algoritmo mais utilizado no aprendizado não-supervisionado é o *k-means*, pois a sua implementação é muito simples, pois basta comprovar o processo de minimizacão da distância quadrática total de cada ponto de um grupo, em

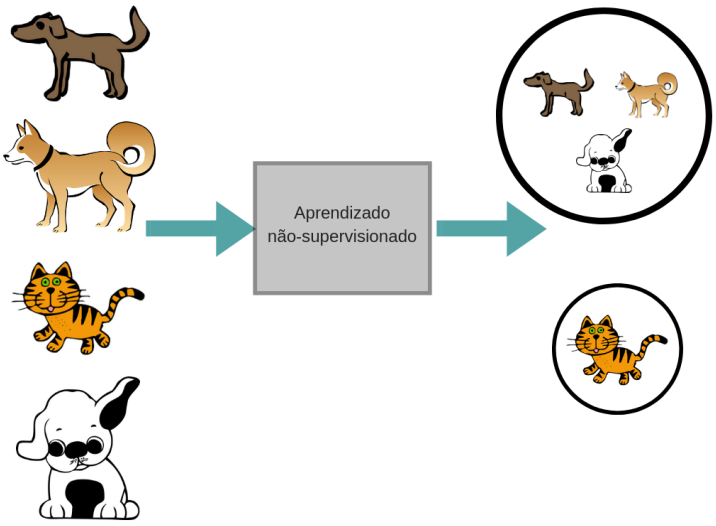

Figura 2.5: Treinamento de um modelo de aprendizado de máquina não-supervisionado.

relação ao centróide de referência [29]. Na Equação 2.12 é possível visualizar como é realizada a seleção de clusters através do número de amostras X, descrito pela média μ_j de cada amostra nos clusters.

$$\sum_{i=0}^{n} \min_{\mu_j \in C}(||x_i - \mu_j||^2) \tag{2.12}$$

2.1.2.3 Aprendizado por reforço

O último paradigma é baseado na forma em que o modelo reage de acordo com o que o ambiente passa para ele, é possível aprender continuamente. Esse aprendizado é realizado através das ações que são executadas, caso seja a decisão correta o modelo ganha uma recompensa, caso contrário pode ser imposta algum tipo de penalidade [30]. Esse paradigma de aprendizado é necessário ajustar os parâmetros de forma contínua com o intuito de maximar ou minimizar um determinado índice, e sempre há um "treinador"nessa fase verificando se os estímulos proporcionados estão tendo a saída correta, pois caso esses dados não estejam balanceados, eles podem alterar a performance do algoritmo [31].

Na aprendizagem por reforço, o modelo só aprende com uma série de estímulos, que podem ser recompensas ou punições no decorrer da aprendizagem, um exemplo do mundo real seria, avaliar um motorista com nota baixa em um aplicativo de locomoção urbana, nesse caso ele irá saber que existiu alguma condunta ruim. Um

caso em que recebe um estímulo positivo pode ser dar um biscoito a um cachorro após fazer de forma correta o que foi pedido.

Na Figura 2.6 é possível ver um fluxograma de treinamento de um modelo utilizando o reforço.

Figura 2.6: Fluxograma de treinamento de um modelo de aprendizado de máquina por reforço.

2.1.3 Medidas de avaliação

Ao utilizar um modelo AM é necessário que existam parâmetros para demonstrar se o modelo está correto ou não. Pois existem possibilidade do modelo ter um *overfitting* que é quando o modelo se ajusta perfeitamente ao conjunto de dados de teste, mas não é capaz de prever novos resultados, já no *underfitting* é quando não há dados suficientes ou com baixa qualidade e não é possível treinar o modelo da maneira correta [32].

Na validação cruzada, os dados são divididos de forma aleatória em n grupos de tamanho aproximadamente iguais. E este processo é realizado por várias vezes, e em cada rodada do teste é considerado um novo valor. O seu desempenho é calculado a partir da média dos desempenhos calculados em cada uma das divisões, o número que n pode assumir é variável, depende da situação, os valores mais utilizados são 5 e 10[33]

A matriz de confusão é outra forma de verificar a qualidade do modelo, pois ela fornece métricas importantes para medir o desempenho. Essa é baseada em classes

relacionadas à assertividade da previsão das classes. Os positivos verdadeiros (VP) e verdadeiros negativos (VN) que são os valores corretos que o sistema previu de forma correta, e também existe outras duas medidas que são os falsos negativos (FN) e falsos positivos (FP) que são as entradas que foram classificadas de forma errada e tiveram sua saída trocada [34]. Na Figura 2.7 é possível visualizar as classes de uma matriz de confusão.

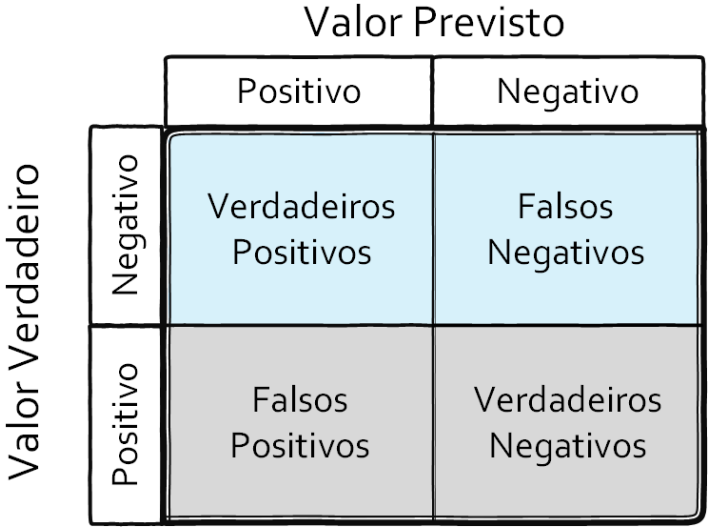

Figura 2.7: Matriz de confusão [34].

Com os resultados extraídos da matriz de confusão é possível calcular outras métricas para verificar o desempenho de cada algoritmo, na Equação 2.13 é calculada a acurácia do modelo, na Equação 2.14 é calculada a precisão do modelo, na Equação 2.15 é calculada a frequência que o modelo encontra os exemplos de uma determinada classe e por fim na Equação 2.16, temos o F1-score que é uma métrica que indica a qualidade geral do modelo.

$$\text{Acurácia} = \frac{VP + VN}{VP + VN + FP + FN} \tag{2.13}$$

$$\text{Precisão} = \frac{VP}{VP + VN + FP + FN} \tag{2.14}$$

$$\text{Recall} = \frac{VP}{VP + FN} \tag{2.15}$$

$$\text{F1-Score} = \frac{2 * \text{Precisão} * \text{Recall}}{\text{Precisão} + \text{Recall}} \tag{2.16}$$

2.2 Mineração de Texto

A TM representa uma parcela da mineração de dados, também pode ser descrita como a forma de descobrir padrões em textos, que vão desde identificar palavras até mesmo sumarizar o texto ou realizar análise de sentimentos de determinada informação. A sua análise tenta recolher o maior número de informações possíveis para encontrar padrões úteis e também pode ser utilizada juntamente com algoritmos de AM. Os textos geralmente são mais difíceis de obter análises, pois em grande parte não são apresentados de forma estruturada, portanto para que o resultado seja satisfatório é necessário uma etapa de pré-processamento dos dados, para que de alguma forma seja possível estruturar o texto.

O corpus é o conjunto de frases que foram rotulados com a sua saída que será utilizada em um modelo de AM, ou seja ele é a base de dados que contém todos os textos disponíveis de forma rotulada.

2.2.1 Processamento de Linguagem Natural

O PLN é uma área de pesquisa que preocupa-se em explorar como um computador pode entender um texto escrito por humanos, com esse tipo de ferramenta é possível realizar análises morfológicas, sintáticas ou léxicas no texto e também extrair informações de um determinado documento [35].

2.2.1.1 Análise de Sentimentos

A análise de sentimentos é um campo que envolve outras ciências além da computação, como a psicologia e linguística, pois é necessário a construção de dicionários para facilitar a classificação de textos, quando por exemplo a intenção é apenas identificar a polaridade de uma palavras, ela poderá ter três saídas conhecidas: Positivo, Negativo e Neutro [14].

Existem diversas formas de fazer o uso da análise de sentimentos, a realização dessas tarefas no idioma em português ainda é muito díficil devido não ter dicionários grandes para realizar o tratamento com um grande volume de dados, também é possível verificar a polaridade de cada frase através da bibliotca [11], mas para isso é necessário traduzir toda a frase para o idioma inglês através da API aberta da *Google*, o que pode implicar um viés durante o processo de análise de sentimentos.

É possível utilizar um modelo de AM para realizar a análise de sentimentos, onde faz o uso do modelo previamente treinado para realizar a predição da polaridade de novas frases que não existiam durante a fase de treinamento [36].

2.2.1.2 Pré-Processamento

O pré-processamento dos dados é a parte mais importante da análise, pois é a etapa onde irá ser tratada as informações e com isso facilitará o processo de aprendizado da máquina. Onde será indentificado padrões para facilitar a limpeza dos dados, onde pode ser definido que todo o texto deverá estar escrito em letra minúscula, remoção de pontuação e acentos, nessa fase também é possível retirar os textos que não estão no padrão desejado, por exemplo apresentam *emotions* ao invés de texto, remoção de números, ou seja, o interesse dessa etapa é filtrar o texto de forma que a máquina obtenha apenas informações que irão ser utilizadas para treinamento [37].

2.2.1.3 Tokenização

Acontece na etapa de pré-processamento e a sua utilização tem como finalidade extrair pequenas informações de um texto livre. Recebe esse nome, pois cada divisão é nomeada de *token* e que na maioria das vezes é uma palavra que está inserida no texto, mas em outros casos podem ser separados em letras ou duplas ou trincas de palavras[38]. Nesse trabalho foi utilizado o *token* em sua forma clássica utilizando a linguagem de programação *Python* e a biblioteca *NLTK* [39].

Por exemplo, caso tenhamos a frase *"Obina é melhor do que Neymar!"*, teremos sete tokens, conforme exemplo abaixo.

[*Obina*] [é] [*melhor*] [*do*] [*que*] [*Neymar*] [!]

2.2.1.4 Stemming

Essa técnica serve para evitar que palavras que tenham o mesmo radical sejam classificadas de maneira distintas, ou seja, essa técnica tem por objetivo reduzir até a menor parte com significado da palavra que no caso é o radical, portanto no processo de *steeming*, palavras como caminhar, caminhada, caminharam, caminharão resultem na mesma palavra final: caminh. A necessário atentar-se na escolha do algoritmo correto para realizar *steeming*, pois essa etapa está diretamente ligada com o idioma em que o texto está escrito.

2.2.1.5 Stop Words

São os termos não relevantes para a análise do texto, são palavras que na maioria das vezes são conectivos de frases ou preposições, que podem ser consideradas como palavras vazias, não existe uma lista universal de *Stop Words*, mas por padrão é utilizada a lista presente na biblioteca NLTK [39]. Na Tabela 2.1 encontra-se algumas das *Stop Words* utilizadas nesse trabalho.

de	os	tua	tem	estão	da	lhes	essas
e	é	foi	nossas	muito	o	se	tuas
tu	por	as	sua	aquele	entre	não	ele
delas	minhas	às	nos	pela	havia	me	como
ser	aqueles	nossa	vocês	eu	ter	tenho	suas
está	isso	pelos	estes	tinha	depois	foram	este
para	só	quem	deles	isto	um	eles	do
vos	mais	mesmo	num	dele	será	minha	a
no	teus	à	você	em	meus	esses	pelas
com	ao	dela	há	que	na	nosso	te
aos	dos	ou	aquela	era	uma	das	esta
teu	nem	já	até	seja	esse	mas	quando
aquelas	nossos	têm	também	seus	lhe	meu	seu
ela	elas	estas	nós	sem	essa	fosse	qual
		pelo	nas	numa	aquilo		

Tabela 2.1: *Stop Words* utilizadas nesse trabalho

2.2.2 Bag of Words

Os dados utilizados nesse trabalho são textos, ou seja, é um formato que não existe um padrão conhecido, o que dificulta a extração de informações. Por isso é necessário alguma técnica que organize esse conjunto de dados e extraia informações relevantes deles[40].

Portanto é necessário utilizar o *bag of words* para realizar a estruturação do texto que será analisado. No caso mais simples, seria uma matriz que apenas indica se determinado termo existe ou não naquela frase, dessa forma seria uma abordagem booleana. Onde temos duas frases que serão transformadas na matriz a ser utilizada nas próximas etapas.

D_1 = José gosta de assistir os jogos do Flamengo.
D_2 = Maria gosta de assistir filmes

$$\begin{bmatrix} \text{José} & \text{gosta} & \text{assistir} & \text{jogos} & \text{Flamengo} & \text{Maria} & \text{filmes} \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

2.2.3 Term Frequency

A medida de *Term Frequency* é uma outra forma de representar os dados na matriz do modelo estudado, ele considera a quantidade de ocorrência que um determinado termo n_t, o que difere drásticamente do primeiro tipo de *bag of words*[41]. Esse modelo tem uma performance superior ao booleano que foi apresentado na subseção anterior, a sua fórmula matemática é descrita na Equação 2.17.

$$TF_t = \frac{n_t}{N}, \tag{2.17}$$

onde n_t é a frequência que o termo n apareceu dentro do *Corpus* e N é a quantidade total de termos.

2.2.4 Term Frequency - Inverse Document Frequency (TF-IDF)

Uma abordagem mais eficaz é a TF-IDF, pois quando se utiliza a *Term Frequency* ele considera todas as palavras tendo a mesma importância, portanto é necessário normalizar esses valores, para diminuir a influência que esses termos irão ter no processo de análise textual [42]. Para isso é necessário utilizar a equação descrita por [42], onde é apresentada uma função inversa que irá calcular a importância de cada termo. É possível visualizar a Equação 2.18 que é descrita por [42].

$$IDF = \log \frac{N}{d}, \tag{2.18}$$

Onde N é o número total de frases e d é a quantidade de documentos nos quais o termo aparece.

A fórmula final do TF-IDF é obtida a partir do produto da Equação 2.17, que é a frequência de cada termo com a Equação 2.18, que é o a taxa de normalização dos valores. Na Equação 2.19 é possível ver que esse tipo de normalização irá desconsiderar palavras que aparecem em todas as frases do *Corpus*, pois considera que esses termos não são úteis para as etapas seguintes.

$$Tf - IDF = TF * IDF, \tag{2.19}$$

Capítulo 3

Framework Proposto

3.1 Extração de Dados

3.2 Pré-processamento de dados

3.3 Dicionário Léxico

3.4 Classificação dos Sentimentos

3.5 Visualização dos Dados

Capítulo 4

Resultados

4.1 Avaliação da Performance dos Algoritmos

4.2 Validação Cruzada N-Fold

4.3 Avaliação do erro utilizando o modelo

4.4 Análise Espaço-Temporal

4.5 Resultado das Eleições

Capítulo 5

Conclusão

Este documento serve de exemplo da utilização da classe `UnB-CIC` para escrever um texto cujo objetivo é apresentar os resultados de um trabalho científico. A sequência de ideias apresentada deve fluir claramente, de modo que o leitor consiga compreender os principais conceitos e resultados apresentados, bem como encontrar informações sobre conceitos secundários.

Referências

[1] Song, M., M. C. Kim e Y. K. Jeong: *Analyzing the political landscape of 2012 Korean presidential election in Twitter*. IEEE Intelligent Systems, 29(2):18–26, 2014. 1, 4

[2] Araniti, G., I. Bisio e M. De Sanctis: *Towards the reliable and efficient interplanetary internet: A survey of possible advanced networking and communications solutions*. Em *2009 First International Conference on Advances in Satellite and Space Communications*, páginas 30–34, July 2009. 1

[3] Stats, Internet Live: *Elaboration of data by international telecommunication union (itu), united nations population division, internet mobile association of india (iamai), world bank.* `http://www.internetlivestats.com/internet-users-by-country/`. 1

[4] Miller, G. A.: *Wordnet: a lexical database for English*. Communications of the ACM, 38(11):39–41, 1995. 2

[5] W3Techs: *Historical trends in the usage of content languages for websites.* `https://w3techs.com/technologies/history_overview/content_language`. 2

[6] Twitter: *Twitter developer platform*. `https://developer.twitter.com`. 2

[7] Comunicação, Empresa Brasil de: *Facebook chega a 127 milhões de usuários no brasil*. `http://agenciabrasil.ebc.com.br/economia/noticia/2018-07/facebook-chega-127-milhoes-de-usuarios-no-brasil`. 2

[8] País, El: *Cambridge analytica, empresa pivô no escândalo do facebook, é fechada.* `https://brasil.elpais.com/brasil/2018/05/02/internacional/1525285885_691249.html`. 2

[9] Souza, Marlo e Renata Vieira: *Sentiment analysis on twitter data for portuguese language*. Em *Proceedings of the 10th International Conference on Computational Processing of the Portuguese Language*, PROPOR'12, páginas 241–247, Berlin, Heidelberg, 2012. Springer-Verlag, ISBN 978-3-642-28884-5. `http://dx.doi.org/10.1007/978-3-642-28885-2_28`. 3

[10] Neuenschwander, Bruna, Adriano C.M. Pereira, Wagner Meira, Jr. e Denilson Barbosa: *Sentiment analysis for streams of web data: A case study of brazilian financial markets*. Em *Proceedings of the 20th Brazilian Symposium on Multimedia and the Web*, WebMedia '14, páginas 167–170, New York, NY, USA, 2014. ACM, ISBN 978-1-4503-3230-9. `http://doi.acm.org/10.1145/2664551.2664579`. 3

[11] Loria, S.: *TextBlob: Simplified text processing.* `http://textblob.readthedocs.io/en/dev/index.html`. 3, 17

[12] Wagh, Rasika e Payal Punde: *Survey on sentiment analysis using twitter dataset.* Em *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, páginas 208–211. IEEE, 2018. 3

[13] Wang, Yuanhong, Yang Liu e Xiaohui Yu: *Collaborative filtering with aspect-based opinion mining: A tensor factorization approach.* Em *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, páginas 1152–1157. IEEE, 2012. 4

[14] Wegrzyn-Wolska, K. e L. Bougueroua: *Tweets mining for French presidential election.* Em *2012 Fourth International Conference on Computational Aspects of Social Networks (CASoN)*, páginas 138–143, Nov 2012. 4, 17

[15] Cerón-Guzmán, J. A. e E. León-Guzmán: *A sentiment analysis system of Spanish tweets and its application in Colombia 2014 presidential election.* Em *2016 IEEE International Conferences on Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom) (BDCloud-SocialCom-SustainCom)*, páginas 250–257, Oct 2016. 4

[16] Joyce, B. e J. Deng: *Sentiment analysis of tweets for the 2016 US presidential election.* Em *2017 IEEE MIT Undergraduate Research Technology Conference (URTC)*, páginas 1–4, Nov 2017. 4

[17] Nasrabadi, Nasser M: *Pattern recognition and machine learning.* Journal of electronic imaging, 16(4):049901, 2007. 6, 13

[18] Bonaccorso, Giuseppe: *Machine Learning Algorithms.* Packt Publishing Ltd, 2017. 6, 13

[19] Marks, II, Robert J., Jacek M. Zurada e Charles J. Robinson (editores): *Computational Intelligence: Imitating Life.* IEEE Press, Piscataway, NJ, USA, 1994, ISBN 0780311043. 7

[20] Mitchell, Thomas M.: *Machine Learning.* McGraw-Hill, Inc., New York, NY, USA, 1ª edição, 1997, ISBN 0070428077, 9780070428072. 7

[21] Chang, Chih Chung e Chih Jen Lin: *Libsvm: a library for support vector machines.* ACM transactions on intelligent systems and technology (TIST), 2(3):27, 2011. 8

[22] Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg *et al.*: *Scikit-learn: Machine learning in python.* Journal of machine learning research, 12(Oct):2825–2830, 2011. 9

[23] Vieira, Lucas Maciel, Clicia Grativol, Flavia Thiebaut, Thais G Carvalho, Pablo R Hardoim, Adriana Hemerly, Sergio Lifschitz, Paulo Cavalcanti Gomes Ferreira e Maria Emilia MT Walter: *Plantrna_sniffer: a svm-based workflow to predict long intergenic non-coding rnas in plants.* Non-coding RNA, 3(1):11, 2017. 9

[24] Lorena, Ana Carolina e André CPLF de Carvalho: *Uma introdução às support vector machines*. Revista de Informática Teórica e Aplicada, 14(2):43–67. 10

[25] McCallum, Andrew e Kamal Nigam: *A comparison of event models for naive bayes text classification*, 1998. 10

[26] Zhang, Harry: *The optimality of naive bayes*. AA, 1(2):3, 2004. 11

[27] Coelho, Vinícius Coutinho Guimarães: *Análise de logs de interação em ambiente educacional corporativo via mineração de dados educacionais*. 11

[28] Breiman, Leo: *Classification and regression trees*. Routledge, 2017. 11

[29] MacQueen, James *et al.*: *Some methods for classification and analysis of multivariate observations*. Em *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1 de *1*, páginas 281–297. Oakland, CA, USA, 1967. 14

[30] Zurada, Jacek M.: *Introduction to Artificial Neural Systems*. PWS Publishing Co., Boston, MA, USA, 1st edição, 1999, ISBN 053495460X. 14

[31] Mazurowski, Maciej A, Piotr A Habas, Jacek M Zurada, Joseph Y Lo, Jay A Baker e Georgia D Tourassi: *Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance*. Neural networks, 21(2-3):427–436, 2008. 14

[32] Aalst, Wil MP Van der, Vladimir Rubin, HMW Verbeek, Boudewijn F van Dongen, Ekkart Kindler e Christian W Günther: *Process mining: a two-step approach to balance between underfitting and overfitting*. Software & Systems Modeling, 9(1):87, 2010. 15

[33] Kohavi, Ron *et al.*: *A study of cross-validation and bootstrap for accuracy estimation and model selection*. Em *Ijcai*, volume 14, páginas 1137–1145. Montreal, Canada, 1995. 15

[34] Andreoni, Martin: *An intrusion detection and prevention architecture for software defined networking*, abril 2014. 16

[35] Liddy, Elizabeth D: *Natural language processing*. 2001. 17

[36] Golbeck, Jennifer, Justin M Grimes e Anthony Rogers: *Twitter use by the us congress*. Journal of the American Society for Information Science and Technology, 61(8):1612–1621, 2010. 18

[37] Ferreira, Lohann Paterno Coutinho, Mariza Miola Dosciatti e Emerson Cabrera: *Um método para o pré-processamento de textos na identificação automática de emoções para o português do brasil*. 18

[38] Mcnamee, Paul e James Mayfield: *Character n-gram tokenization for european language text retrieval*. Information retrieval, 7(1-2):73–97, 2004. 18

[39] Bird, Steven e Edward Loper: *Nltk: the natural language toolkit*. Em *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, página 31. Association for Computational Linguistics, 2004. 18, 19

[40] Wallach, Hanna M: *Topic modeling: beyond bag-of-words*. Em *Proceedings of the 23rd international conference on Machine learning*, páginas 977–984. ACM, 2006. 19

[41] Salton, Gerard e Christopher Buckley: *Term-weighting approaches in automatic text retrieval*. Information processing & management, 24(5):513–523, 1988. 20

[42] Sparck Jones, Karen: *A statistical interpretation of term specificity and its application in retrieval*. Journal of documentation, 28(1):11–21, 1972. 20

Apêndice A

A.1 Appendix