Classificação de Posicionamento Político em Tweets no Cenário Brasileiro

Victor Hugo de Andrade Landin[201611009] victor.landin@estudante.ufla.br

Departamento de Ciência da Computação - Universidade Federal de Lavras - UFLA, Lavras, CEP 37200-000, Brasil
http://www.ufla.br

Keywords: Posicionamento político · Mineração de dados · Mineração de opinião · Politica no Twitter.

1 Introdução

Com o crescente avanço da tecnologia, a participação de dispositivos eletrônicos no cotidiano das pessoas, tais como smartphones, têm se tornado cada vez maior. Associado à isso, têm se popularizado o papel das mídias sociais, que vem exercendo um grande papel na comunicação social. Como exemplo disso, o *Twitter*[1], que é uma rede social e um servidor para *microblogging*, que permite que usuários enviem e recebam atualizações pessoais de outros usuários em todo o planeta.

No Brasil, os usuários gastam cerca de 9 horas e 14 minutos navegando na internet[2] e o Twitter tem cerca de 40 milhões de contas vinculadas à usuários brasileiros, segundo estudo de uma empresa de Mídias Sociais chamada Semiocast [3].

Com esse grande número de usuários, o Twitter tem se tornado uma grande base de dados sobre conteúdos diversos acerca de seus usuários. Dentre esses conteúdos, está incluso opiniões sobre política, religião e sexualidade [1].

Por ser uma ferramenta em tempo real e aberta ao público, o Twitter permite uma grande aproximação das pessoas com seus representantes políticos. Dessa forma, esses representantes têm cada dia mais adotado a plataforma[4], levando o aumento do número de discussões com viés político. Neste cenário, pessoas têm expressado seus posicionamentos políticos, dos quais podem ter diferentes correntes ideológicas, como a esquerda e a direita[5], além dos posicionamentos

[1] Disponível em: www.twitter.com.
[2] Disponível em: https://www.techtudo.com.br/noticias/2018/02/10-fatos-sobre-o-uso-de-redes-sociais-no-brasil-que-voce-precisa-saber.ghtml.
[3] Disponível em: https://exame.abril.com.br/negocios/dino/62-da-populacao-brasileira-esta-ativa-nas-redes-sociais.
[4] Disponível em: https://politica.estadao.com.br/noticias/geral,twitter-permite-mais-proximidade-com-representantes,70003000258.
[5] Disponível em: https://veja.abril.com.br/blog/felipe-moura-brasil/esquerda-x-direita-entenda-de-uma-vez/.

mais neutros ou também chamados de centro. Estes posicionamentos podem caracterizar os usuários em grupos distintos, dos quais muitas vezes geram bolhas sociais, onde usuários interagem cada vez menos com outros usuários fora de sua bolha[6].

Várias aplicações têm surgidos a fim de extrair opiniões existentes em conteúdos como esse. Um exemplo disso são técnicas usadas na mineração de opiniões e detecção de posicionamentos, consistindo em identificar o posicionamento do autor de determinado texto em relação a algum propósito [2]. A maioria dessas aplicações se baseia em técnicas de mineração de texto para interpretar essa grande quantidade de dados das quais, geralmente, não seriam processadas manualmente em um tempo possível[3].

O meio mais natural de se interpretar conteúdos em forma de textos é por meio da linguagem natural, como a linguagem utilizada na escrita deste texto, o Português. A área de Processamento da Linguagem Natural (PLN), no campo da Ciência da Computação, tem como alvo a capacitação do computador na interpretação automática das línguas naturais[4].

Neste trabalho, propõe-se a investigação e implementação de um sistema para inferir o posicionamento político de pequenas sentenças de texto em português, dos quais serão classificados em um dos 2 tipos de posicionamentos políticos diferentes, o posicionamento de esquerda e o de direita. Há ainda uma terceira opção para classificar sentenças onde não há um posicionamentos político, o qual chamamos de posicionamento nulo.

A base de dados é extraída por meio da API do Twitter[7], uma ferramenta desenvolvida e disponibilizada pelo próprio Twitter que permite a extração de *tweets*.

Os *tweets* extraídos utilizando a API constituem a base de dados necessária para classificar um modelo de Machine Learning utilizando uma abordagem semi-supervisionada, que consiste em rotular alguns desses elementos para auxiliar no treinamento do modelo. Estima-se que para este problema, a base de dados seja maior que algumas dezenas de milhares. Uma parte da base de dados é dividida e rotulada manualmente, para assim gerar uma porção de dados de treinamento, que aumenta conforme o conceito do aprendizado semi-supervisionado. Dessa forma, têm-se números crescentes de elementos para treinar o modelo a cada treinamento que se passa. Uma porção dos dados é separada para testar a precisão do modelo. Além disso, alguns modelos serão utilizados para o treinamento a fim de comparar a precisão destes.

Espera-se que as soluções produzidas neste trabalho possam auxiliar trabalhos futuros, como por exemplo: detecção de bolhas sociais e políticas, detecção de tendências em opiniões de usuários, estudos sobre a relação de tendências no Twitter e resultados de eleições, entre outros possíveis benefícios.

O restante deste texto está organizado da seguinte forma. A seção 2 apresenta o referencial teórico sobre os conceitos utilizados neste trabalho. A seção

[6] Disponível em: https://temas.folha.uol.com.br/gps-ideologico/as-bolhas-na-rede-social/bolha-politica-da-direita-no-twitter-e-mais-fechada-que-a-da-esquerda.shtml.

[7] Disponível em: https://developer.twitter.com/en/docs.

3 descreve os trabalhos relacionados. A seção 4 detalha a metodologia utilizada para chegar a solução do problema apresentado acima. A seção 5 apresenta análise e discussão dos resultados obtidos. A seção 6 contém conclusões e direções para trabalhos futuros.

2 Referencial Teórico

Esta seção apresenta os principais conceitos acerca da descoberta de conhecimentos em bases de dados, da mineração de dados e outros conceitos utilizados neste trabalho.

2.1 Descoberta de conhecimento em base de dados

Segundo Galvão et al. [5], a descoberta de conhecimento em bases de dados (KDD) pode ser definida como o processo de extração de informação a partir de dados registrados numa base de dados, um conhecimento implícito, previamente desconhecido, potencialmente útil e compreensível. Sendo definida em Fayyad [6] como uma tentativa de solucionar o problema da sobrecarga de dados causado pela chamada "era da informação", além de ser "um processo não trivial de identificação de novos padrões válidos, úteis e compreensíveis", sendo a última uma das definições mais aceitas, segundo Camilo et al. [7].

Ainda em Camilo, não se tem um concenso a cerca da definição dos termos KDD e mineração de dados, para alguns ambos são considerados sinônimos e para outros não.

Fayyed diferencia os termos de forma que, o KDD se refere ao processo geral de descoberta de conhecimento útil a partir de dados e a mineração de dados se refere a uma etapa específica desse processo.

Em um nível mais abstrato, o campo KDD se preocupa com o desenvolvimento de métodos e técnicas para entender os dados. O problema básico abordado pelo processo KDD é o mapeamento de dados de baixo nível (que geralmente são muito volumosos para entender e digerir facilmente) para outras formas que podem ser mais compactas (por exemplo, um relatório curto), mais abstratas (por exemplo , uma aproximação ou modelo descritivo do processo que gerou os dados) ou mais útil (por exemplo, um modelo preditivo para estimar o valor de casos futuros). No centro do processo está a aplicação de métodos específicos de mineração de dados para descoberta e extração de padrões [6].

Assim, o processo de KDD utiliza conceitos de base de dados, métodos estatísticos, ferramentas de visualização e técnicas de inteligência artificial, dividindo-se nas etapas de seleção, pré-processamento, transformação, DM e avaliação/ interpretação. Dentre essas etapas, a mais importante é a mineração de dados, que comprova o pressuposto da transformação de dados em informação, e posteriormente em conhecimento, o que torna a técnica imprescindível para o processo de tomada de decisão [5].

2.2 Mineração de Dados

Como definido em Galvão, expressão Mineração de Dados (DM) surge inicialmente, como um sinônimo de KDD, porém é apenas uma das etapas da descoberta de dados no processo global do KDD.

O conhecimento adquirido através da DM tem se mostrado bastante útil nas mais diversas áreas, como medicina, finanças, comércio, marketing, telecomunicações, meteorologia, agropecuária, bioinformáticas, entre outras. Sendo a Mineração de Dados uma das alternativas mais eficazes para extrair conhecimento a partir de grandes volumes de dados, possuindo alguns objetivos como descobrir relações ocultas, padrões e gerar regras para predizer e correlacionar dados, que podem ajudar instituições e corporações nas tomadas de decisões mais rápidas ou até atingir um grau de confiança maior em relação aos resultados [5].

Segundo Jothi et al. [8], a mineração de dados é fundamentada em disciplinas como o Aprendizado de Maquina, a Inteligência Artificial, a Probabilidade e a Estatística.

A mineração de dados possui várias etapas, entre elas estão presentes a definição clara do problema; a seleção de todas as fontes internas e externas de dados e a preparação dos dados, que inclui o pré-processamento, reformatação dos dados e análise dos resultados obtidos no processo de DM [5].

Dessa forma a DM não é um processo trivial, consistindo essa na habilidade de identificar, nos dados, os padrões válidos, novos, potencialmente utéis e compreensíveis, envolvendo métodos estatísticos, ferramentas de visualização e técnicas de inteligência artificial [5].

Os processos de desenvolvimento da mineração de dados envolvem diversas tarefas, métodos e algoritmos com objetivo de possibilitar a extração de novos conhecimentos [10].

2.2.1 Mineração de Texto

A mineração de texto, também definida como descoberta de conhecimento a partir de texto, refere-se ao processo de extrair informações em dados no formato de texto[16], ou seja dados semi estruturados ou não estruturados, como por exemplo paragráfos e/ou pequenas sentenças em blogs, portais de noticias, entre outros. Ela abrange um amplo conjunto de tópicos e algoritmos relacionados à análise de texto como recuperação de informações, processamento de linguagem natural, mineração de dados, aprendizado de máquina [15].

2.2.2 Processamento de Linguagem Natural

O Processamento de linguagem natural é um subárea da ciência da computação, inteligência artificial e lingüística, que visa a compreensão, pelo computador e sistemas computacionais, da linguagem natural[15], ou seja, visa a capacitação dos computadores na interpretação automática das línguas naturais [4].

2.2.3 Pré-Processamento

Com a base de dados definida, é necessário explorar os dados a fim de adquirir conhecimento sobre os mesmos e, além disso, encontrar possíveis fenômenos que possam comprometer sua qualidade, tais como: valores em branco ou nulo, valores viciados, váriaveis duplicadas e informações que possam interferir no processo. Á medida em que problemas vão surgindo e o entendimento vai sendo obtido, ocorre a preparação dos dados para que se possa aplicar os algoritmos de mineração de dados [7].

Segundo Han et al.[12], a preparação dos dados, também chamada de pré-processamento, é um processo que consiste no tratamento dos dados de modo à identificar e solucionar possíveis problemas relacionados a consistência dos mesmos, além de transformar os dados em entradas possíveis para alguns algoritmos. Entre as etapas do pré-processamento, destacam-se a limpeza dos dados, transformação dos dados e redução dos dados.

A **limpeza dos dados** visa eliminar problemas de inconsistências para que estes não influenciem no resultado final. Existem varias técnicas a serem utilizadas nesta etapa, que vão desde a remoção de registros com problema, passando pela atribuição de valores padrões e aplicação de técnicas de agrupamento para auxiliar na descoberta de melhores valores [12].

A **transformação dos dados** é uma etapa que consiste em transformar os dados para uma entrada correspondente em uma representação diferente, pois alguns algoritmos trabalham apenas com representações específicas. Por exemplo, alguns algoritmos trabalham apenas com valores numéricos, outros com valores categóricos e outros com valores vetorizados [7].

Diversas técnicas podem ser usadas na transformação de acordo com os objetivos pretendidos. Algumas dessas técnicas são: suavização (remove valores errados dos dados), agrupamento (agrupa valores em faixas sumarizadas), generalização (converte valores muito específicos para valores mais genéricos) e normalização (colocar as variáveis em uma mesma escala)[7].

A **redução dos dados** consiste em converter o volume original dos dados em um volume menor, sem perder a representatividade dos dados originais. As técnicas de redução de dados são necessárias quando se tem um volume muito grande de dados, de modo que isso torne impraticável o processo de análise dos dados. Esta redução permite que os algoritmos de mineração de dados sejam executados com mais eficiência e mantendo a qualidade dos resultados.[7]

2.2.4 Modelos

Como definido em Kantardzic [9], existem dois tipos de modelos em mineração de dados: o modelo preditivo e o modelo descritivo. O modelo preditivo, geralmente, é aplicado em funções de aprendizado supervisionado, em que se rotula os dados, com o objetivo de classificar ou prever valores desconhecidos ou prever valores futuros de variavéis de interesse. Já o modelo descritivo, é aplicado em aprendizados não supervisionados, em que não se rotula os dados,

para encontrar padrões que descrevem os dados que podem ser interpretados em humanos.

2.2.5 Tarefas

Na DM, são definidas tarefas e algoritmos que serão utilizados para obter uma resposta para o problema, de acordo com os objetivos [11].

Entre as várias tarefas, destacam-se algumas que são as mais utilizadas: associação, classificação, regressão, clusterização e sumarização. Neste trabalho será desenvolvido uma tarefa de classificação [5].

A classificação consiste na predição de uma variável categórica, ou seja, descobrir uma função que mapeie um conjunto de registros em um conjunto de classes predefinidas [5].

O modelo analisa o conjunto de registros fornecidos, com cada registro já contendo a indicação à qual classe pertence, a fim de 'aprender' como classificar um novo registro (aprendizado supervisionado) [7], ou seja, tomam como entrada uma coleção de casos, cada um pertencendo a um pequeno número de classes e descrito por seus valores para um conjunto fixo de atributos, e produz um classificador que pode prever com precisão a classe à qual um novo caso pertence [14]. Dessa forma, a função pode ser aplicada em novos registros, para prever a classe em que tais registros se enquadram, os classificando.

2.2.6 Métodos

Os métodos são tecnologias existentes que são utilizados para se transformar dados em conhecimento util. No caso deste trabalho, onde temos uma tarefa de classificação, vários algoritmos podem ser empregados, como as Redes Neurais, Classificadores Bayesianos, Árvore de Decisão, Algoritmos Genéticos, entre outros [5]. Os algoritmos utilizados neste trabalho, a fim de classificar os dados mencionados, serão Árvore de Decisão (DT) e Rede Neural Artificial.

A **Árvore de Decisão** funciona como um fluxograma em formato de árvore, onde temos nós(galhos) internos que indicam possíveis decisões sobre uma variável, particionando os dados, e folhas que representam as categorias. Dessa forma, as ligações entre os nós representam os possíveis valores do teste do nó superior e as folhas, extremidade da árvore, indicam possíveis categorias a qual o registro pode pertencer. Após a árvore estar montada, para classificar um registro, percorre a árvore pelos nós, testando o registro e assim encontrando o caminho até uma folha que contém a classificação desse registro [7].

A **Rede Neural Artificial**(RNA) é um modelo computacional que constrói um modelo matemático baseado no sistema neural humano, possuindo capacidade de aprendizado, generalização, associação e abstração[5,13].

Ela é constituída por sistemas paralelos distribuídos em conjuntos de unidades simples de processamento. Essas unidades de processamento são camadas interligadas por um grande número de conexões. Na maioria dos modelos, es-

sas conexões estão associadas a pesos que, após o processo de aprendizagem, armazenam o conhecimento adquirido pela rede[13].

3 Trabalhos Relacionados

A detecção de posicionamento político em tweets tem sido abordada utilizando diferentes técnicas e contextos, segundo Araujo et al [2018], um dos primeiros trabalhos que utilizou dados do *Twitter* para análise política foi de Tumasjan [17], no qual é realizado uma análise utilizando dados do *Twitter* para prever as eleições federais na Alemanha. Nesse trabalho foram coletados *tweets* que continham nomes dos seis partidos representados no parlamento alemão e selecionaram políticos importantes relacionados a esses partidos. Utilizando um método baseado em regras para identificar os tweets como politicamente relevantes, eles declararam que o número de mensagens mencionando um partido reflete o resultado da eleição [3].

Em Araujo et al [2018] foi coletado um conjunto de 2.881 tweets holandeses e utilizado uma abordagem de aprendizado supervisionado para a classificação bidimensional de *tweets* em político e não politicos. O objetivo foi inferir se a classificação de conteúdo político do *Twitter* utilizando uma abordagem de aprendizado supervisionado supera um método baseado em regras. Vários algoritmos de aprendizado de máquina foram treinados utilizando o córpus etiquetado e as precisões foram comparadas [3].

Em Christhie et al [2018], foi desenvolvido um classificador para inferir automaticamente o posicionamento expresso em tweets escritos em português. Nesse trabalho foram coletados dados de *tweets* sobre possíveis presidenciavéis utilizando *hashtags* especificas para coletar dados favoravéis e contrários para cada um deles. A hipótese investigada era a de que o forte viés ideologico existente nessas hashtags poderia ser utilizado para etiquetar automaticamente um conjunto de *tweets* que possa ser utilizado para treinar um classificador [2].

4 Metodologia

4.1 Coleta dos dados

Para o desenvolvimento de uma solução para o problema, foi necessário obter uma base de dados(córpus) que contivesse as características necessárias para a classificação do modelo. Dessa forma, foi necessário obter diferentes sentenças nas quais ocorresse um dos fenômenos a seguir:

- Posicionamento político de esquerda.
- Posicionamento político de direita.

A solução desenvolvida consiste, à princípio, em levantar perfis de usuários populares no *Twitter*. Desses, alguns com posicionamento político de esquerda e

Tabela 1. Personalidades políticas populares no *Twitter*.

Usuário de Esquerda	Nome de Usuário	Usuário de Direita	Nome de Usuário
Fernando Haddad	@Haddad_Fernando	Jair Bolsonaro	@jairbolsonaro
Guilherme Boulos	@GuilhermeBoulos	Kim kataguiri	@kimpkat
Ciro Gomes	@cirogomes	Arthur do Val	@arthurmoledoval
Lula	@LulaOficial	Nando Moura	@moura_101
Gregorio Duvivier	@gduvivier	Joao Amoedo	@joaoamoedonovo
Manuela Davila	@ManuelaDavila	Fernando Holiday	@FernandoHoliday
Marcelo Freixo	@MarceloFreixo	Danilo Gentili	@DaniloGentili
PSOL	@psol50	PSL	@PSL_Nacional
Samia Bomfim	@samiabomfim	Major Olipio	@majorolimpio
Luciana Genro	@lucianagenro	Delegado Francischini	@Francischini
Randolfe Rodrigues	@randolfeap	Olavo de Carvalho	@OdeCarvalho

outros com posicionamento político de direita. As personalidades foram levantadas por pertencerem à grupos com posicionamentos políticos distintos. A lista desses usuários está ilustrada na tabela 1.

A partir desses usuários, utilizando a *API do Twitter*, foi levantado outros usuários que possuem ligação com algumas dessas personalidades politicas, no caso usuários que seguissem essas personalidades.

Dessa forma, foram selecionados perfis de usuários em que é possível encontrar *tweets* com posicionamentos politicos e também não políticos, pois nem todo seguidor de uma personalidade compartilha de seus posicionamentos e também leva-se em conta que um usuário poste *tweets* relacionados a outros assuntos além da política.

Após isso, foi extraído o córpus a partir desses usuários, utilizando a *API* para selecionar uma porção de seus *tweets*. A *API* permite buscar *tweets* filtrando por palavras-chave, como nome de usuário, hashtags, palavras e localização.

Foi extraído um conjunto de 9100 *tweets*, que estavam no formato JSON e foi selecionado os atributos identificador(id), texto extendido(full_text), urls presentes no texto e idioma(lang). Esses dados foram armazenados em um arquivo CSV em formato de dataframe da biblioteca Pandas[8] do Python[9].

4.2 Limpeza dos Dados

Após selecionar os atributos necessários, tem-se a necessidade de limpar os dados, removendo informações que possam influenciar negativamente a eficácia dos métodos de aprendizado de máquina. Abaixo é descrito as etapas utilizadas para realizar a limpeza dos dados.

À princípio, foi utilizado o atributo idioma (lang) para remover do córpus *tweets* que não estivessem em português. Com isso reduziu-se o córpus para 8040 *tweets*.

[8] Disponível em: https://pandas.pydata.org/pandas-docs/stable/.
[9] Disponível em: https://www.python.org/doc/.

O córpus continha dados duplicados de duas formas: *tweets* duplicados e *tweets* diferentes com o mesmo texto, ou seja, a duplicata estava no atributo texto extendido. Ambos as formas foram removidas, mantendo apenas a primeira ocorrência das duplicatas. Assim o córpus foi reduzido novamente, agora para 7650 *tweets*.

Em alguns textos continham nomes de usuários, em casos de usuários citados ou *retweets*. Removemos o nome de usuário ou *username* pois não é uma informação útil. Dessa forma podemos remover também *retweets*.

Outro problema, são os links encontrados dentro do atributo texto extendido, que podem influenciar os resultados deste trabalho, gerando ruídos. Por ocorrência disso, estes links foram removidos utilizando o módulo *Re*[10] do *Python*, que tratam expressões regulares, permitindo assim identificar dentro do textos as substrings referentes aos links e removê-las.

Após isso, houve a necessidade de remover caracteres duplicados, muito comum de serem utilizados para dar enfâse em algumas palavras, esses também foram removidos utilizando a biblioteca *Re*.

Além disso, todos os caracteres foram transformados para caracteres minúsculos e pontuações foram removidas, utilizando a biblioteca *nltk*[11] do Python. Por último, foram removidas duplicatas novamente, agora no texto normalizado e o córpus foi reduzido a 7517 *tweets*.

Com isso, o dataframe recebeu uma nova coluna, a coluna dos dados normalizados. O resultado da limpeza foi salvo dentro de um arquivo *csv*.

4.3 Classificação manual dos tweets

Após realizar a limpeza dos dados, o próximo passo é a classificação manual dos tweets para realizar o treinamento e o teste dos modelos. Com isso, o córpus foi separado em dois conjuntos, um para treinamento e um para teste.

Utilizando a biblioteca Numpy[12] do Python, foi selecionado aleatoriamente 23% das chaves do dataframe que contém todo o córpus e, com isso, foi extraído do córpus o conjunto de *tweets* para formar a base de treinamento. Esse conjunto foi salvo em um arquivo *.csv*.

O restante do córpus forma o conjunto de teste e também foi extraído e salvo em um arquivo *.csv*. Com isso, a base de dados foi divida em 1787 *tweets* para treinamento e 5730 *tweets* para teste.

Com o conjunto de treinamento selecionado, o próximo passo é a classificação ou rotulamento de cada tweet. Utilizando a ferramenta de edição de planilhas, LibreOffice Calc[13], foi posśível visualizar cada *tweet* e indicar a classe à qual pertence. Os dados foram rotulados em esquerda, neutro e direita, representados respectivamente por -1, 0 e 1.

[10] Disponível em: https://docs.python.org/3/library/re.html.
[11] Disponível em: https://www.nltk.org.
[12] Disponível em: https://numpy.org/devdocs/.
[13] Disponível em: https://pt-br.libreoffice.org/descubra/calc/.

4.4 Transformação dos dados e Treinamento dos modelos

Para que os dados possam ser utilizados pelos modelos, seja para treinar ou para testar, eles precisam estar dispostos de uma forma que os modelos possam processá-los.

A solução utilizada consiste em representar os textos na forma de um vetor, onde cada palavra é representada por um valor numérico. Para isso, foi utilizada a classe *tfidfvectorizer* para transformar esses dados em vetores.

Essa classe pertence a biblioteca *Scikit-learn*[14], que é uma biblioteca no *Python* que contém diversos métodos, modelos e classes para uso no aprendizado de máquina.

O primeiro passo para transformar os dados foi criar um dicionário que mapea cada palavra para um valor numérico único utilizando a função *fit* dessa classe. Foi necessário enviar como parametro todo o córpus, que é a junção dos dados de treinamento e de teste. Após isso, para transformar a base de treino e de teste foi utilizado a função *fit_ transform*.

Com isso, os dados estão preparados para serem utilizado nos modelos de aprendizado de máquina. Os modelos de Árvore de Decisão[15] e Rede Neural Artificial[16] utilizados estão presentes na biblioteca *Scikit-learn*.

Para realizar o treinamento foi utilizado o córpus de treinamento em forma de lista de sentenças e uma outra lista com os posicionamentos classificados anteriormente, de forma manual, de cada sentença da primeira lista.

5 Resultados

Nesta seção apresentamos os resultados obtidos nos modelos treinados, que estão divididos em classificação da base de teste e avaliação dos classificadores.

5.1 Classificação da base de teste

Os modelos treinados foram utilizados para classificar novos dados, que não contém uma classificação definida. Para isso foi utilizada a base de dados de teste definida na seção 4.3. Essa base de dados contém 5730 *tweets* e as classificações desses dados pelos modelos são mostradas abaixo.

Para o modelo de Árvore de Decisão, teve-se a classificação de 396 instâncias como direita, 501 como esquerda e 4833 como neutro. É possível visualizar esses dados na figura 1.

Diferente da Árvore de Decisão, para o modelo de Redes Neurais Artificiais, teve-se a classificação de 213 instâncias como direita, 274 como esquerda e 5243 como neutro. É possível visualizar esses dados na figura 2.

[14] Disponível em: https://scikit-learn.org/stable/.
[15] Disponível em: https://scikit-learn.org/stable/modules/tree.html.
[16] Disponível em: https://scikit-learn.org/stable/modules/neural_networks_supervised.html.

Figura 1. Classificação da base de dados pelo modelo de Árvore de Decisão

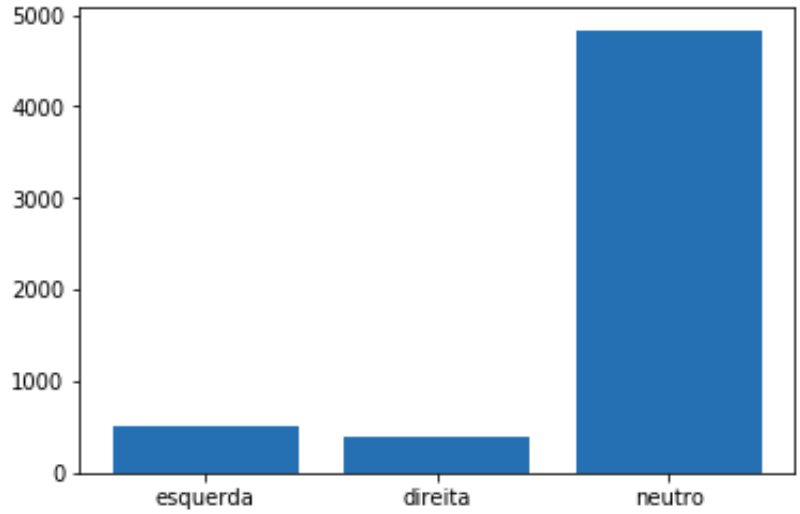
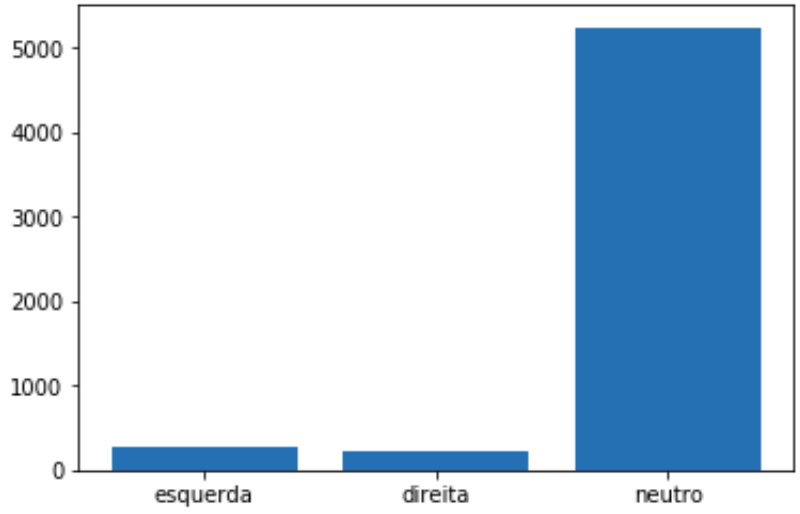


Figura 2. Classificação da base de dados pelo modelo de Árvore de Decisão



5.2 Avaliação dos Classificadores

Para avaliar os classificadores foi necessário utilizar dados dos quais eles não conheciam, dessa forma é possível ter uma noção de como estes vão se comportar em um sistema real.

Na seção 4.3 dividimos o córpus em treinamento e teste, dessa forma, na base de teste, temos instâncias das quais o modelo não conhece e podemos testá-lo utilizando a mesma. Porém, é possível ter-se uma desvantagem, pois caso os dados de teste e de treinamento forem muito semelhantes, a acurácia dos modelos será alta mesmo que não reflita a realidade na qual o modelo será usado. Neste caso, teremos uma análise não consistente.

Dessa forma, diferentes métricas serão utilizadas para definir a precisão dos modelos. Nas subsessões a seguir, veremos as técnicas de Croos Validation e Precisão dos Classificadores por Amostragem.

5.2.1 Cross Validation

A técnica de Croos Validation, ou validação cruzada, consiste em dividir os dados em K partes chamadas *folds*. Nessa divisão, uma parte da base será utilizada para teste e as restantes para treinamento. Esse passo é realizado repetidamente até que o modelo seja testado e treinado com todas as *folds*. A acurácia é o percentual de acertos em relação ao total de instâncias que o modelo obtiver.

Para cada um dos dois modelos foi utilizado essa técnica para verificar a acurácia de ambos, utilizando k igual a 10. Além disso o córpus de treinamento foi utilizado, sendo divido em 10 partes como mencionado acima. Os resultados são mostrados na tabela 2.

Tabela 2. Acurácia dos modelos utilizando a técnica Croos Validation

Modelos	**Acurácia**
`Árvore de Decisão`	0.701175
`Redes Neurais Artificiais`	0.725797

5.2.2 Precisão dos Classificadores por Amostragem

Para verificar informações sobre as precisões de cada classificador foi selecionado uma amostra de 150 instâncias dos resultados encontrados na seção 5.1 e com isso foi gerado duas novas bases de dados. Essas bases de dados contém o texto de cada *tweet* e suas predições em cada modelo. Dessa forma tem-se uma amostra dos resultados de cada modelo que podem ser utilizados para verificar se essas precisões foram corretas.

Após isso, as instâncias dessas amostras foram verificadas quanto a classificação dos modelos de forma a atestar se foi correta ou incorreta. As possíveis classificações são o Positivo (P), em que há uma classificação correta e o Negativo (N), em que há uma classificação incorreta e além disso, foi indicado a classe correta a qual cada instância pertence.

As tabelas 3 e 4 mostram a matriz de confusão para as instâncias de cada modelo. Essas matrizes mostram a quantidade de instâncias para cada classe e a distribuições das predições das mesmas em cada classificador.

Tabela 3. Matriz de confusão da amostra dos resultados do modelo DT

Real\Predição	Esquerda	Neutro	Direita	Total
`Esquerda`	4	10	4	18
`Neutro`	2	113	4	119
`Direita`	1	8	4	13
`Total`	7	131	12	150

Tabela 4. Matriz de confusão da amostra dos resultados do modelo NN

Real\Predição	Esquerda	Neutro	Direita	Total
`Esquerda`	4	14	0	18
`Neutro`	6	112	1	119
`Direita`	0	10	3	13
`Total`	10	136	4	150

A partir dos dados dessas matrizes, calculamos a precisão desses modelos utilizanda a métrica precisão (pr) representada na equação 1, que é calculada a partir da divisão do número de instâncias classificadas de forma corretas pela soma entre número de instâncias classificadas de forma correta e incorreta.

$$pr = \frac{P}{P + N} \tag{1}$$

O modelo de Árvore de Decisão obteve 122 instâncias corretas e 28 incorretas e o modelo de Redes Neurais Artificais obteve 119 instâncias corretas e 31 incorretas. As precisões calculadas são mostradas na tabela 5.

Tabela 5. Precisão dos modelos utilizando amostragem

Modelos	**Precisão**
`Árvore de Decisão`	0.813333
`Redes Neurais Artificiais`	0.793333

6 Discussão

Como mostrado na seção 4.3, grande parte do córpus de treinamento é composta por instâncias neutras, da mesma forma o córpus de teste, como mostrado nas matrizes de confusão nas tabelas 3 e 4. Esta grande diferenças entre as instâncias pode influênciar muito nos classificadores e nos resultados. Nesta seção, analisamos os resultados da seção anterior e apresentamos possíveis melhorias.

Segundo as métricas de avaliação dos classificadores definidos na seção 5.2, os modelos tiveram percentuais de acertos superiores a 70%, chegando ao seu maior número de acertos em 5.2.2, em que a precisão calculada foi de 81,33% para a Árvore de Decisão e 79% para as Redes Neurais Artificiais.

Analisando as matrizes de confusão no início desse mesmo capítulo, pode-se perceber que houve um número de acertos muito grande para instâncias neutras, sendo 113 acertos na Árvore de Decisão e 112 nas Redes Neurais Artificiais, ambos em 119 instâncias neutras. Porém, para as instâncias em que há posiciomento político, seja de esquerda ou de direita, o número de acertos não acompanha as precisões dos modelos. Em ambas as matrizes o número de acertos para esse tipo de instâncias é baixo. Para as 18 instâncias com posicionamento político de esquerda, o modelo da DT acertou apenas 4, mesmo número que o modelo NN acertou. Já para as 13 instâncias com posicionamento político de direita, o modelo DT acertou 4 e o modelo NN 3.

A partir desses dados, podemos inferir que ambos os modelos não tiveram bons resultados, levando em consideração que o objetivo principal era classificar instãncias em que houvessem posicionamento político de esquerda ou de direita. Esse fato pode ter ocorrido devido ao alto número de instâncias neutras no córpus inicial, fazendo com que os modelos tenham muitas informações sobre as características de instâncias neutras e informações insuficientes para instâncias das demais classes.

Espera-se que a obtenção de um córpus com mais instâncias com posicionamento político de esquerda e de direita, dando mais representatividade a base de dados e equilibrando o número de instâncias em cada classe, melhore os resultados dos classificadores. Para isso, sugerimos como melhoria a busca de *tweets* utilizando palavras chaves que caracterizem alguns tipos de vertentes e ideologias nos campos políticos no cenário brasileiro. Dessa forma, tornar o córpus mais abrangente e maior. Neste sentido, aumentar a porcentagem do córpus de treinamento faz com que os modelos aprendam mais sobre o córpus geral.

Como trabalhos futuros, esperamos utilizar os modelos treinados para auxiliar na detecção de bolhas políticas no Twitter, à princípio identificando posicionamento político em perfis de usuário e análisar os mesmos a fim de mapear os vínculos entre esses, para identificar ou não essas possíveis bolhas. Outro proposta futura é a detecção de tendências políticas, em que espera-se utilizar o modelo para classificar assuntos de cunho ideológico e, posteriormente, buscar identificar se o assunto é uma tendência, caracterizada por ser um assunto muito comentado ou os comentários sobre o assunto estão em crescente expansão.

7 Apêndice

A implementação prática deste trabalho se encontra em um repositório no seguinte link: https://github.com/vhal9/Classificacao-de-Posicionamento-Poltico-em-Tweets-no-cenario-brasileiro.

Referências

1. REIS, J. C., GONÇALVES, P., ARAUJO, M., PEREIRA, A. C., and BENEVENUTO, F. (2015). Uma abordagem multilıngue para analise de sentimentos. In BraSNAM - Brazilian Workshop on Social Network Analysis and Mining.
2. CHRISTHIE W., REIS,J.C.S., BENEVENUTO, F., MORO, M.M., ALMEIDA, V.: Detecção de Posicionamento em Tweets sobre Política no Contexto Brasileiro. Universidade Federal de Minas Gerais (UFMG) – Brasil. (2018).
3. Araújo, E. and Ebbelaar, D.: Detecting Dutch Political Tweets: A Classifier based on Voting System using Supervised Learning. Behavioural Informatics Research Group, VU Amsterdam. Amsterdam, The Netherlands. 2018.
4. Jurafsky, D.; James, H. M. (2009). Speech and Language Processing: An Introduction to Natural Language Processing, Speech Recognition, and Computational Linguistics. 2nd edition. Prentice-Hall.
5. Galvão, N.D; Marin, H.F.: Técnica de mineração de dados: uma revisão da literatura. Acta Paulista de Enfermagem, vol. 22, núm. 5, outubro, 2009, pp. 686-690. Escola Paulista de Enfermagem. São Paulo, Brasil. Disponivel em: https://www.redalyc.org/articulo.oa?id=307023846014
6. FAYYAD, U; PIATETSKY-SHAPIRO, G; SMYTH, P. From Data Mining to Knowledge Discovery in Databases. American Association for Artificial Intelligence, 1996.
7. Camilo, C. O.; da Silva, J.C.: Mineração de Dados: Conceitos, Tarefas, Métodos e Ferramentas. Instituto de Informática. Universidade Federal de Goiás. Goiás, Brasil. (2009).
8. Jothi, N.; Rashid, N. A. ; Husain, W.: Data Mining in Healthcare: A Review. The Third Information Systems International Conference. School of Computer Sciences, Universiti Sains Malaysia, 11800 Minden, Penang Malaysia.
9. Kantardzic, M.: Data Mining: Concepts, Models, Methods, and Algorithms, 2nd ed. Wiley-IEEE Press, 2011.
10. Cardoso O.N.P., Machado R.T.M.: Gestão do conhecimento usando data mining: estudo de caso na Universidade Federal de Lavras. Rev Adm Pública. 2008;42(3):495-528.
11. Matos, G.; Chalmeta, R.; Coltell, O.: Metodología para la extracción del conocimiento empresarial a partir de los datos. Inf Tecnol. 2006;17(2):81-8.
12. HAN, J; KAMBER, M. Data Mining: Concepts and Techniques. Elsevier, 2006.
13. Kovács ZL. Redes neurais artificiais: fundamentos e aplicações. 3a ed. rev. São Paulo: Livraria da Física; 2002.
14. X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, S. Y. Philip, et al. Top 10 algorithms in data mining. Knowledge and information systems, 14(1):1–37, 2008.
15. M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, and K. Kochut. A brief survey of text mining: Classification, clustering and extraction techniques. arXiv preprint arXiv:1707.02919, 2017.

16. Ronen Feldman and Ido Dagan. 1995. Knowledge Discovery in Textual Databases (KDT).. In KDD, Vol. 95. 112–117.
17. Tumasjan, A., Sprenger, T. O., Sandner, P. G., and Welpe, I. M. (2010). Predicting elections with twitter: What 140 characters reveal about political sentiment. ICWSM, 10(1):178–185.