Pós-Graduação em Ciência da Computação

Douglas Álisson Marques de Sá Vitório

Avaliando Estratégias de Seleção de *Active Learning* para Mineração de Opinião com Fluxos Contínuos de Dados

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
http://cin.ufpe.br/~posgraduacao

Recife
2020

Douglas Álisson Marques de Sá Vitório

Avaliando Estratégias de Seleção de *Active Learning* para Mineração de Opinião com Fluxos Contínuos de Dados

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Área de Concentração: Inteligência Computacional
Orientador: Dr. Adriano Lorena Inacio de Oliveira
Co-orientadora: Dra. Ellen Polliana Ramos Souza

Recife
2020

Catalogação na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

V845a Vitório, Douglas Álisson Marques de Sá
Avaliando estratégias de seleção de *active learning* para mineração de opinião com fluxos contínuos de dados / Douglas Álisson Marques de Sá Vitório. – 2020.
105 f.: il., fig., tab.

Orientador: Adriano Lorena Inacio de Oliveira.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2020.
Inclui referências e apêndices.

1. Inteligência computacional. 2. Mineração de opinião. I. Oliveira, Adriano Lorena Inacio de (orientador). II. Título.

006.31 CDD (23. ed.) UFPE - CCEN 2020 - 90

Douglas Álisson Marques de Sá Vitório

"**Avaliando Estratégias de Seleção de *Active Learning* para Mineração de Opinião com Fluxos Contínuos de Dados**"

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Aprovado em: 20/02/2020.

BANCA EXAMINADORA

Prof. Dr. George Darmiton da Cunha Cavalcanti
Centro de Informática / UFPE

Prof. Dr. George Gomes Cabral
Departamento de Computação / UFRPE

Prof. Dr. Adriano Lorena Inácio de Oliveira
Centro de Informática / UFPE

Dedico este trabalho a meus pais, pois sem eles eu não estaria aqui; e a meus amigos, sem os quais a vida não teria cor.

AGRADECIMENTOS

Agradeço primeiramente aos meus pais, **Iraneide** e **Fernandes**, por terem me proporcionado as oportunidades e o apoio necessários para que eu chegasse aqui, além de todo o carinho, amor e atenção que eles puderam dar. Amo vocês.

Agradeço também às minhas primas, **Isabela** e **Ingrid**, as quais eu sempre considerei irmãs e se tornaram ainda mais ao me proporcionarem um lar em Recife durante os últimos dois anos. Isto tornou tudo bem mais fácil e divertido, pois ter uma família que vai além do sangue é essencial. Amo vocês.

Agradeço a **Brunna**, **Clariana** e **Danielly** por todos os momentos juntos, todos os abraços e todos os cuidados, e por me aturarem durantes esses mais de três anos de amizade. Mesmo com a distância, eu sempre soube que, quando voltasse, elas estariam lá para mim. Amo vocês.

Agradeço a **Jessyca**, a qual não permitiu que eu ficasse sozinho durante as aulas do mestrado, quando eu não conhecia mais ninguém, e sei que não me deixará sozinho pelo resto da vida. Daquelas amizades que começam "por acaso", por conta de um trabalho em grupo da disciplina de Aprendizagem de Máquina, mas que marcam sua vida de todas as formas imagináveis. A colega que se transformou em uma das melhores amigas e foi companheira durante toda a jornada do mestrado. Amo você.

Agradeço a **Andréa** e **Mariana**, minhas companheiras e confidentes em Recife. Nosso "trio" que começou nas ladeiras de Olinda em pleno sábado de carnaval, e proporcionou incontáveis momentos felizes durante todo o ano de 2019. Além de tudo, são uma razão para que eu sempre volte a Recife. Amo vocês.

Como sempre, agradeço a **Clarissa**, à qual serei eternamente grato, pois me ajudou no momento em que eu mais precisei, se fazendo presente quando ninguém mais se fez. E, mesmo com a distância, se faz presente até hoje. Amo você.

Agradeço a meu orientador **Adriano** e minha coorientadora **Ellen**, que me aceitaram e me deram a oportunidade de fazer este mestrado, me aconselhando e me orientando sempre. Sem eles não haveria esta dissertação.

Por fim, agradeço a **todos** os que me deram pelo menos um abraço, trazendo, assim, mais cor à vida.

RESUMO

Mineração de Opinião, também conhecida como Análise de Sentimento, é a área de estudo que analisa computacionalmente os sentimentos e opiniões das pessoas acerca de entidades, como produtos e serviços, expressos de forma não estruturada, como em texto, por exemplo. Entretanto, as abordagens mais comuns de Mineração de Opinião não estão aptas a lidar com as características e os desafios trazidos pelo processamento de fluxos contínuos de dados (*data streams*), devido, principalmente, ao fato de estes terem uma natureza evolutiva, requerendo atualizações constantes do modelo, e aquelas serem fortemente baseadas em Aprendizagem Supervisionada; dessa forma, uma alternativa é a utilização de técnicas Semi-supervisionadas, como a de *Active Learning*, a qual visa rotular apenas dados selecionados, em vez de rotular todo o conjunto de dados. A abordagem de *Active Learning* requer a escolha de uma estratégia para selecionar as instâncias mais relevantes para atualização do modelo de aprendizagem; contudo, nenhum estudo realizou uma análise com o objetivo de identificar as melhores estratégias para Mineração de Opinião. Nesta pesquisa, portanto, essa análise é realizada com base em oito estratégias de seleção: seis delas encontradas na literatura e duas propostas pelo autor; e utilizando 20 conjuntos de dados oriundos de quatro corpora com *data streams*: dois deles construídos especificamente para esta pesquisa e contendo dados do Facebook e do Twitter acerca da Eleição Presidencial no Brasil em 2018. As estratégias foram avaliadas em três cenários diferentes e com três tipos de classificadores. Com base nos resultados e considerando os 20 conjuntos de dados utilizados, pôde-se perceber que a técnica *Entropy* é a mais indicada, em termos de *f-measure*, para o maior número de situações; porém, esta estratégia seleciona um número muito grande de documentos, na maioria dos casos selecionando o dobro das outras, não sendo recomendável para casos nos quais não há a possibilidade de rotular um grande volume de dados. Nestes cenários, a estratégia *Variable Entropy*, proposta neste trabalho, se mostrou uma opção mais viável.

Palavras-chaves: Mineração de Opinião. Análise de Sentimento. *Active Learning*. Fluxos contínuos de dados.

ABSTRACT

Opinion Mining, also known as Sentiment Analysis, is the field of study that analyzes people's sentiments and opinions about entities, such as products and services, expressed in an unstructured form, e.g., in textual input. However, the most common Opinion Mining approaches are not able to deal with the characteristics and challenges brought by the processing of continuous data streams, mainly due to the evolutive nature of the streams, and due to the fact that these approaches are strongly based on Supervised Learning; so, an alternative is the use of semi-supervised techniques such as Active Learning, which aims to label only selected data, rather the entire data set. The Active Learning approach requires the choice of a sampling strategy to select the most valuable instances to update the learning model; nevertheless, no study has performed an analysis in order to identify the best strategies for Opinion Mining. Therefore, in this study, this analysis is made based on eight sampling strategies: six of them found in the literature and two proposed by the author; and using 20 data sets from four data streams corpora, two of them specially built for this research and containing Facebook and Twitter data about the 2018 Presidential Election in Brazil. The strategies were evaluated in three different scenarios and with three kinds of classifiers. According to the results and considering the 20 data sets used, it could be observed that the Entropy is the most indicated strategy, in terms of accuracy, for most cases; however, this strategy selects a large number of instances, in most cases sampling a number twice as large as the others, not being recommended for scenarios in which there is no possibility of labeling a lot of data. In these cases, the Variable Entropy strategy, proposed in this work, proved to be the most viable choice.

Keywords: Opinion Mining. Sentiment Analysis. Active Learning. Data streams.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE QUADROS

LISTA DE ABREVIATURAS E SIGLAS

AL	*Active Learning*
AM	Aprendizagem de Máquina
CD	*Critical Difference*
CGU	Conteúdo Gerado pelo Usuário
MD	Mineração de Dados
MNB	*Multinomial Naïve Bayes*
MO	Mineração de Opinião
MT	Mineração de Texto
NLTK	*Natural Language Toolkit*
PLN	Processamento de Linguagem Natural
RL	Regressão Logística
SMC	Sistemas de Múltiplos Classificadores
SVM	*Support Vector Machine*

LISTA DE SÍMBOLOS

θ	Parâmetro teta, que corresponde ao limiar usado pelas estratégias de seleção

SUMÁRIO

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

O crescimento recente e exponencial das mídias sociais e do Conteúdo Gerado pelo Usuário (CGU) na Internet, potencializado por meio dessas mídias, vem fornecendo uma grande quantidade de informação que possibilita a descoberta dos sentimentos, das opiniões e das experiências de usuários e clientes. O volume desse tipo de dado tem crescido de terabytes a petabytes nos últimos anos (MARINE-ROIG; CLAVÉ, 2015).

As mídias sociais são ricas fontes de informação em tempo real, fazendo com que muitas entidades, como companhias e figuras políticas, tenham demonstrado interesse em conhecer as opiniões das pessoas acerca de produtos e serviços.

Dentre as mídias sociais mais utilizadas atualmente, destacam-se o Facebook e o Twitter como importantes fontes de CGU (SOUZA et al., 2018). Enquanto a primeira é menos utilizada em Mineração de Texto (MT) por geralmente conter dados não-textuais, como imagens; a segunda possui mensagens, denominadas *tweets*, com um limite atual de 280 caracteres, o que torna seu conteúdo mais objetivo.

Compreender o que as pessoas estão pensando e suas opiniões é fundamental para a tomada de decisão, principalmente em cenários nos quais essas pessoas exprimem seus comentários voluntariamente (ALVES et al., 2014). Entretanto, é impossível para seres humanos interpretar completamente essa grande quantidade de dados gerados pelo usuário em uma quantidade razoável de tempo, o que criou a necessidade de desenvolver sistemas capazes de extrair informação desse tipo de dado de forma automática.

As abordagens mais comuns para lidar com esse problema são baseadas em Mineração de Opinião (MO), que também é conhecida na literatura como Análise de Sentimento e é, segundo Liu e Zhang (2012), a área de estudo que analisa os sentimentos, opiniões, avaliações, atitudes e emoções das pessoas acerca de entidades, como produtos, serviços, organizações, indivíduos, problemas, eventos, tópicos e seus atributos, expressos de forma não-estruturada, como em texto, por exemplo. Esta análise é realizada através da classificação de opinião de um documento, sentença ou característica em três principais categorias: *positiva*, *negativa* e *neutra*.

Dessa forma, a Mineração de Opinião tem colaborado com os objetivos de companhias e organizações, auxiliando-as a observar as reações do público e a satisfação dos clientes. Além disso, esse entendimento das opiniões também já se mostrou importante em outros domínios, como na Política (PANG; LEE, 2008). Em sua pesquisa, Souza et al. (2018) mostraram um aumento no interesse pelo uso de aplicações de MO em campanhas políticas, principalmente usando dados escritos em Português; e estima-se que as mídias sociais vêm tendo um grande impacto em eleições, como na Eleição Presidencial nos Estados Unidos

em 2016 (ALLCOTT; GENTZKOW, 2017).

1.2 DESCRIÇÃO DO PROBLEMA E MOTIVAÇÃO

Porém, mesmo sendo as mais utilizadas para resolver os problemas supracitados, as aplicações de Mineração de Opinião têm tido seu foco em domínios estáticos e bem-conhecidos, como o de avaliações de filmes, não estando aptas a lidar eficientemente com as características dos fluxos de dados contínuos (*data streams*) e os desafios trazidos por eles (GUERRA; MEIRA JR.; CARDIE, 2014). A análise de *data streams* oriundos de mídias sociais, também conhecidos como *social streams*, é importante pelo fato de que as opiniões das pessoas acerca de um determinado assunto ou entidade podem mudar com a chegada de novas informações (WANG et al., 2013).

Em Aprendizagem de Máquina, uma mudança que ocorre em um determinado conceito ao longo do tempo é conhecida como *concept drift* (WIDMER; KUBAT, 1996). De forma semelhante e de acordo com Wang et al. (2013), uma mudança em uma opinião que ocorre com o passar do tempo pode ser considerada um *opinion drift*, sendo a detecção desses *drifts* importante para os resultados de MO, pois a opinião geral acerca de uma entidade pode mudar. Já Silva et al. (2011) denominaram essa mudança como *sentiment drift* e afirmaram que tanto as características associadas a certos sentimentos quanto a distribuição deles podem mudar, tornando as predições menos precisas com o passar do tempo.

Aplicações que lidam com *data streams* e que são sensíveis a *drifts* precisam enfrentar dois principais obstáculos: 1) a disponibilidade limitada de dados rotulados, devido à velocidade com que os dados podem chegar e à alta latência de verificação (*verification latency*) (MARRS; HICKEY; BLACK, 2010) que pode existir nesses casos; e 2) a necessidade de atualização constante do modelo de aprendizagem, devido à natureza evolucionária dos *data streams* e à ocorrência de *drifts*. Logo, o fato de que os modelos mais utilizados de MO são fortemente baseados em Aprendizagem Supervisionada (RAVI; RAVI, 2015) se torna um problema, já que este tipo de abordagem requer uma grande quantidade de dados rotulados; enquanto que o segundo desafio se encontra nas mudanças de vocabulário que ocorrem no CGU (SALEIRO et al., 2017) e, principalmente, na supracitada natureza dos *data streams* (GUERRA; MEIRA JR.; CARDIE, 2014).

Devido aos problemas apresentados, uma alternativa é o uso de Aprendizagem Semi-supervisionada, a qual não necessita que todo o conjunto de dados seja rotulado, mas apenas uma parte dele. Uma abordagem semi-supervisionada comum em Mineração de Dados é a que usa *Active Learning* (AL) (ZHU et al., 2007), a qual, ao rotular apenas os dados mais valiosos em vez de todo o conjunto de dados, consegue lidar de maneira eficaz com problemas para os quais dados rotulados são custosos de obter (ZIMMERMANN; NTOUTSI; SPILIOPOULOU, 2015). Porém, é necessária a escolha de uma estratégia de

seleção (*sampling strategy*) para decidir quais instâncias dos dados terão seus rótulos solicitados, afim de que elas sejam utilizadas para alimentar o modelo de treinamento.

Os trabalhos encontrados na literatura que utilizam *Active Learning* em aplicações de Mineração de Opinião (SMAILOVIĆ et al., 2014; ZIMMERMANN; NTOUTSI; SPILIOPOULOU, 2015; KRANJC et al., 2015; ALDOĞAN; YASLAN, 2017; LI et al., 2019) fazem uso de diferentes estratégias de seleção para escolher as instâncias, e eles não realizam uma comparação entre as estratégias utilizadas. Portanto, não é possível identificar quais as melhores técnicas para MO.

Ademais, especificamente para a Língua Portuguesa, sabe-se que há uma falta de bases de dados para serem utilizadas como *benchmark*, conforme relatado por Souza et al. (2018). E não foram encontrados corpora disponíveis publicamente contendo *data streams* escritos em Português.

1.3 OBJETIVOS

1.3.1 Objetivo geral

Assim, o objetivo geral deste trabalho é avaliar estratégias de seleção de *Active Learning*, comparando-as em cenários de Mineração de Opinião com *data streams* extraídas de mídias sociais.

1.3.2 Objetivos específicos

Para alcançar o objetivo geral, os seguintes objetivos específicos foram estabelecidos:

- Construir corpora contendo *data streams* em Língua Portuguesa com dados de mídias sociais;
- Implementar e avaliar estratégias de seleção de *Active Learning* dentro de cenários de Mineração de Opinião.

1.4 METODOLOGIA

Este trabalho avalia seis estratégias de seleção de instâncias de *Active Learning* encontradas na literatura e cujo uso foi percebido em aplicações de MT e MO: *Random Sampling*, *Uncertainty*, *Variable Uncertainty*, *Variable Randomized Uncertainty*, *Information Gain* e *Entropy*; além de duas novas estratégias propostas pelo autor: *Variable Entropy* e *Variable Randomized Entropy*, as quais foram baseadas nas estratégias da literatura.

Para isso, diferentes cenários de Mineração de Opinião foram pensados e construídos utilizando como conjuntos de dados *data streams* oriundos de mídias sociais e pertencentes a quatro corpora: dois publicamente disponíveis na internet e escritos em Língua Inglesa (*Sentiment140* e *Sanders*), que contêm dados do Twitter, e dois construídos pelo autor

utilizando opiniões em Língua Portuguesa referentes à Eleição Presidencial no Brasil em 2018, um contendo dados do Twitter e outro do Facebook. No total, 20 conjuntos de dados foram extraídos dos quatro corpora e utilizados para avaliar as estratégias.

Cada conjunto de dado foi utilizado para treinar e testar cada uma das oito estratégias de seleção, separando-se uma parte do conjunto de treinamento para treinar o primeiro estágio do modelo de aprendizagem e as demais instâncias sendo avaliadas pela *sampling strategy* a fim de serem escolhidas ou não para atualizar o modelo.

Três cenários foram idealizados para realizar uma comparação mais completa das estratégias: 1) utilizando uma abordagem iterativa; 2) selecionando todas as instâncias antes de atualizar o modelo de aprendizagem; e 3) determinando um número fixo de instâncias a serem selecionadas. Na abordagem iterativa, cada vez que uma instância é selecionada, o modelo é atualizado, e o processo continua até que todas as instâncias do conjunto de treinamento tenham sido avaliadas. Ademais, como a maioria das estratégias possui um parâmetro que representa o limiar pelo qual é decidido se uma instância será selecionada ou não, diferentes valores deste parâmetro, denominado θ, também foram avaliados neste trabalho.

Como algoritmos de aprendizagem, três classificadores bastante utilizados em MO foram avaliados: *Multinomial Naïve Bayes* (MNB), *Support Vector Machine* (SVM) e Regressão Logística (RL). Assim, este trabalho também visa avaliar quais as melhores estratégias para cada um dos algoritmos.

Por fim, uma análise comparativa dos resultados foi realizada utilizando a medida de *f-measure* e testes estatísticos, com o objetivo de determinar quais estratégias de AL são mais adequadas para aplicações de Mineração de Opinião com *data streams* oriundos de mídias sociais.

1.5 ESCOPO NEGATIVO

Durante o desenvolvimento desta pesquisa, foram propostas duas novas estratégias de seleção de AL; entretanto, não é o objetivo principal deste trabalho que o desempenho destas estratégias supere o desempenho daquelas existentes na literatura, mas sim avaliar todas as estratégias em cenários de MO, identificando quais as melhores estratégias para cada cenário.

Também não é objetivo desta dissertação avaliar as estratégias utilizando um grande número de algoritmos de aprendizagem, mas apenas três mais utilizados em Mineração de Opinião: *Multinomial Naïve Bayes*, SVM e de Regressão Logística. Além de não constituir o escopo deste estudo avaliar técnicas de pré-processamento de dados (tendo sido utilizadas as técnicas mais simples), mesmo sabendo que estas podem impactar diretamente nos resultados das aplicações de MO, pois o foco do estudo se encontra na etapa de processamento dos dados.

Por fim, este trabalho não visa realizar a detecção e identificação de *drifts* nas bases de dados utilizadas, nem mesmo nos dois corpora propostos; em vez disso, optou-se por uma abordagem implícita para lidar com os possíveis *drifts* existentes nelas, partindo do princípio que a probabilidade destes ocorrerem é alta e atualizando constantemente o modelo de aprendizagem.

1.6 ESTRUTURA DO TRABALHO

O restante desta dissertação está estruturado da seguinte forma:

- No Capítulo 2, é discutido o referencial teórico desta dissertação: explanando sobre *data streams*; Mineração de Opinião e os algoritmos de classificação utilizados; e *Active Learning*. Além dos trabalhos relacionados.
- No Capítulo 3, as estratégias de seleção de *Active Learning* são explicadas.
- No Capítulo 4, a metodologia experimental adotada para este trabalho é detalhada: as bases de dados utilizadas, bem como o processo de criação dos corpora acerca da Eleição Presidencial 2018; o pré-processamento dos dados; as etapas de processamento e avaliação; e a configuração dos experimentos.
- No Capítulo 5, os resultados são apresentados e discutidos.
- Por fim, no Capítulo 6, são realizadas as conclusões deste estudo, destacando suas limitações e contribuições, ideias para trabalhos futuros e as considerações finais.

2 REFERENCIAL TEÓRICO

2.1 *DATA STREAMS*

Um **fluxo contínuo de dados** (*data stream*) é definido como uma sequência ordenada, e possivelmente ilimitada, de dados que são coletados durante um certo período de tempo. Este meio dinâmico de obtenção de bases de dados acrescenta novos desafios para a Aprendizagem de Máquina (AM) e para a Mineração de Dados (*data mining*), já que os métodos tradicionais costumam ser desenvolvidos para bases de dados estáticas (KRAWCZYK et al., 2017).

A obtenção de fluxos contínuos de dados difere da obtenção de dados convencionais pelo fato de, nessa nova forma, os dados poderem ser transitórios (GAMA, 2010), isto é, pode-se não ter acesso aos dados sempre que preciso. Além disso, segundo Krawczyk et al. (2017), os *data streams*, por si só, diferem das bases de dados estáticas por diversos motivos:

- os itens no fluxo aparecem sequencialmente com o tempo;
- não há controle sobre o ordenamento com que os itens chegam e o sistema de processamento desses dados deve estar preparado a reagir a qualquer hora;
- o volume dos dados pode ser enorme, sendo talvez impossível de armazenar todos os dados na memória;
- geralmente apenas uma varredura nos dados pode ser realizada;
- a velocidade com que os itens chegam pode ser muito alta;
- os *data streams* são suscetíveis a mudanças, conhecidas como *concept drifts*;
- e a anotação dos dados pode ser muito custosa (em alguns casos, até impossível) e pode não ser imediata.

Mineração de Dados é realizada tradicionalmente em cenários nos quais os algoritmos podem acessar os dados muitas vezes, contudo, nem todos os itens de um fluxo contínuo de dados podem estar carregados ao mesmo tempo na memória (BIFET, 2009). Por conta disso, os problemas existentes ao se lidar com *data streams* se localizam na limitação de recursos computacionais: tempo, memória e armazenamento.

Outro desafio presente ao se lidar com fluxos contínuos de dados é sua suscetibilidade a mudanças (*drifts*), que ocorrem nos chamados fluxos de dados não-estacionários.

Concept drift é definido como uma mudança que ocorre em um determinado conceito ao longo do tempo (WIDMER; KUBAT, 1996). Essas mudanças fazem com que o modelo

de aprendizagem necessite de atualização, que pode ser realizada de forma contínua (de maneira implícita) ou apenas no momento da detecção de um *drift* (de maneira explícita). A forma implícita de se lidar com *concept drifts* parte do princípio que há uma grande possibilidade da ocorrência de *drifts*. Assim, entende-se que *concept drifts* já são esperados, então o modelo precisa ser adaptativo (ŽLIOBAITĖ et al., 2011).

De maneira similar aos *concept drifts*, Wang et al. (2013) apontaram a existência do que eles denominam ***opinion drifts***, os quais correspondem às mudanças que ocorrem na distribuição de sentimentos em uma base de dados que contenha opiniões textuais; essas mudanças estão associadas à ocorrência de eventos reais que podem afetar a opinião de um grupo acerca de uma determinada entidade. Já Silva et al. (2011) denominaram essas mudanças ***sentiment drifts*** e explicaram que tanto a distribuição de sentimentos quanto as características associadas a cada sentimento podem mudar, tornando as predições menos precisas com o passar do tempo.

Sendo assim, pode-se lidar com os *opinion drifts* (ou *sentiment drifts*) de forma semelhante aos *concept drifts*. Além disso, os problemas trazidos por ambos partem do mesmo princípio: a necessidade de atualização do modelo de aprendizagem.

2.2 MINERAÇÃO DE OPINIÃO

Mineração de Opinião (MO), também conhecida na literatura como Análise de Sentimento, é uma subárea da Mineração de Texto (MT), sendo considerada uma atividade de classificação de texto, já que discrimina a orientação da opinião, ou sentimento, de um determinado documento textual em duas ou mais classes. No que diz respeito à quantidade de classes, ela tem sido realizada de várias formas: binária (duas classes), ternária (três classes), n-ária na forma de estrela (n classes), entre outras (RAVI; RAVI, 2015). Pesquisadores também têm levado em consideração vários tipos de emoção, como as seis emoções universais: *raiva*, *desgosto*, *medo*, *felicidade*, *tristeza* e *surpresa* (PANG; LEE, 2008), para definir a quantidade de classes.

Porém, o tipo mais comum de classificação de texto é a classificação de rótulo único, na qual cada texto pertence a exatamente uma classe ou categoria (CARDOSO-CACHOPO; OLIVEIRA, 2007). E, na Mineração de Opinião, os textos, ou seja, as opiniões, são classificadas geralmente em categorias como *positiva*, *negativa* e *neutra*. Este tipo de classificação é referida na literatura como "polaridade de sentimento" ou "classificação de polaridade".

De acordo com o levantamento realizado por Pang e Lee (2008), o termo "mineração de opinião" (*opinion mining*) apareceu pela primeira vez no estudo de Dave, Lawrence e Pennock (2003). Para estes autores, a ferramenta de MO ideal deveria processar um conjunto de resultados de pesquisa para um determinado item, gerando uma lista de atributos e agregando opiniões acerca de cada um desses atributos. Muito da pesquisa subsequente auto-intitulada como Mineração de Opinião se encaixa nesta definição, porém

o termo também tem sido interpretado mais amplamente, para incluir outros tipos de análise de texto (PANG; LEE, 2008).

Mineração de Opinião costuma ser realizada em três níveis de análise diferentes: documento, sentença ou aspecto (FELDMAN, 2013). A classificação a nível de documento determina a classe à qual a opinião de todo o documento pertence, por exemplo se a opinião do documento é *positiva*, *negativa* ou *neutra*. A classificação a nível de sentença determina a classe de uma sentença específica dentro do documento. E a classificação a nível de aspecto, ou a nível de característica, foca em todas as expressões de sentimentos presentes em um dado documento e o aspecto ao qual elas se referem.

Revisões da literatura sugerem que a maioria das aplicações de MO pode ser classificada em quatro categorias: avaliações de produtos, avaliações de filmes, extração de orientação política e predições do mercado de ações (RAVI; RAVI, 2015). Mídias sociais, como o Facebook e o Twitter, são importantes fontes de opiniões, sendo bastante utilizadas em Mineração de Opinião; porém o Facebook é menos usado em Mineração de Texto porque geralmente contém dados não-textuais, como imagens, o que torna a análise apenas do texto não muito efetiva (PAK; PAROUBEK, 2010; EVANGELISTA; PADILHA, 2014).

As abordagens de MO mais frequentemente utilizadas são as baseadas em técnicas supervisionadas de Aprendizagem de Máquina (AM) e as baseadas em léxicos; mas também há abordagens híbridas, que utilizam uma combinação de ambas (PANG; LEE, 2008; MEDHAT; HASSAN; KORASHY, 2014; RAVI; RAVI, 2015). Os métodos que utilizam AM aplicam algoritmos de classificação para aprender padrões subjacentes a partir de opiniões de exemplo, com o objetivo de classificar novas opiniões (BALAZS; VELÁSQUEZ, 2016). Para estes métodos, são necessários dois conjuntos de dados anotados: um para treinamento e outro para teste. Um bom número de algoritmos de Aprendizagem de Máquina Supervisionada vem sendo utilizado para classificar opiniões e textos de modo geral, dentre eles se destacam os Bayesianos, como o *Multinomial Naïve Bayes* (MNB), e o *Support Vector Machine* (SVM) (RAVI; RAVI, 2015; SOUZA et al., 2016; SOUZA et al., 2018).

Já a abordagem baseada em léxicos, também conhecida como semântica ou simbólica, faz uso de palavras de opinião positivas, usadas para expressar estados desejados, e palavras de opinião negativas, usadas para expressar estados indesejados. Existem também frases de opinião e expressões idiomáticas que, juntas, constituem os chamados léxicos de opinião. Três abordagens principais são utilizadas para construir léxicos de opinião: a abordagem manual, embora demande muito tempo; a abordagem baseada em dicionários, na qual um conjunto inicial, construído manualmente, é incrementado procurando os seus antônimos e sinônimos em corpora como a *WordNet* e *thesaurus*; e aquela baseada em corpus, que começa com uma lista semente de palavras de opinião a fim de encontrar outras em um grande corpus com orientações específicas de contexto (MEDHAT; HASSAN; KORASHY, 2014).

2.2.1 Algoritmos utilizados

Neste trabalho, utilizou-se uma abordagem de Mineração de Opinião baseada em algoritmos supervisionados de Aprendizagem de Máquina: *Multinomial Naïve Bayes* (MNB), *Support Vector Machine* (SVM) e o classificador de Regressão Logística (RL).

2.2.1.1 *Multinomial Naïve Bayes* (MNB)

O *Multinomial Naïve Bayes* (MNB) é um classificador probabilístico baseado no teorema de Bayes, e que usa a probabilidade condicional para classificar os dados em classes predeterminadas. Esta abordagem é chamada *naïve* (ingênua) porque assume a independência entre os vários valores dos atributos (KUMARI, 2014).

O MNB usa a informação da frequência de palavras nos documentos para análise. Cada documento é considerado um conjunto ordenado de palavras obtido a partir de um vocabulário V, e a probabilidade de ocorrência de uma palavra é independente do contexto da palavra e de sua posição no documento (MCCALLUM; NIGAM, 1998).

Também é assumido que o comprimento dos documentos é independente de classe e cada documento d_i é desenhado a partir de uma distribuição multinomial de palavras, com tantas tentativas diferentes quanto o tamanho do documento d_i. Então, sendo N_{it} a contagem do número de vezes que a palavras w_t aparece no documento d_i, a probabilidade de um dado documento pertencer a uma classe é dada pela equação:

$$P(d_i|c_j;\theta) = P(|d_i|)|d_i|! \prod_{t=1}^{|V|} \frac{P(w_t|c_j;\theta)^{N_{it}}}{N_{it}!}, \tag{2.1}$$

na qual $P(d_i|c_j;\theta)$ é a probabilidade de d_i pertencer a classe c_j, $P(|d_i|)$ é a probabilidade do documento d_i e $P(w_t|c_j;\theta)$ é a probabilidade da ocorrência da palavra w_t na classe c_j (MCCALLUM; NIGAM, 1998).

2.2.1.2 *Support Vector Machine* (SVM)

O algoritmo *Support Vector Machine* (SVM) (em Português, Máquina de Vetor de Suporte) foi proposto por Cortes e Vapnik (1995) e usa hiperplanos como limites de decisão. Ele foi desenvolvido para classificações binárias usando um hiperplano de separação ótimo entre as duas classes, pela maximização da margem entre os pontos mais próximos de cada classe, estes denominados "vetores de suporte".

Dado um conjunto de treino com pares de dados anotados $(\mathbf{x}_i, y_i)$, no qual $i = \{1, 2, 3, \ldots, l\}$, $\mathbf{x}_i \in R^n$ e $y_i \in \{1, -1\}^l$, o SVM precisa resolver o problema de otimi-

zação da Equação 2.2:

$$\begin{aligned} \min_{\mathbf{x},b,\xi} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i \\ sujeito\ a \quad & y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \end{aligned} \tag{2.2}$$

na qual $\mathbf{w}$ é o vetor de pesos atribuído às variáveis, ξ_i é a folga, ou correção de erro, adicionada e C é o fator de regularização. O SVM mapeia os vetores de treinamento $\mathbf{x}_i$ em um espaço dimensional mais alto (talvez infinito) pela função ϕ, então encontra o hiperplano separador com a maior margem neste novo espaço dimensional (HSU; CHANG; LIN, 2003).

2.2.1.3 Regressão Logística (RL)

O algoritmo de classificação baseado em Regressão Logística (RL) já se mostrou bastante útil em diversas áreas, como nas de categorização de documentos e Processamento de Linguagem Natural (PLN) (YU; HUANG; LIN, 2011). Ele é um algoritmo linear probabilístico que utiliza um vetor de pesos para construir um hiperplano de separação entre as classes e modela a probabilidade condicional de um instância pertencer a uma classe seguindo a Equação 2.3:

$$P_w(y = \pm 1|\mathbf{x}) \equiv \frac{1}{1 + e^{-y\mathbf{w}^T\mathbf{x}}}, \tag{2.3}$$

na qual $\mathbf{x}$ é a instância, y é o rótulo da classe e $\mathbf{w} \in \mathbb{R}^n$ é o vetor de pesos.

Em um cenário com duas classes e l instâncias, por exemplo, no qual os dados de treino seguem $\{\mathbf{x}_i, y_i\}_{i=1}^{l}$, $\mathbf{x}_i \in \mathbb{R}^n$, $y_i \in \{1, -1\}$, a Regressão Logística minimiza o seguinte logaritmo de verossimilhança:

$$P(\mathbf{w}) = C\sum_{i=1}^{l} log(1 + e^{-y_i\mathbf{w}^T\mathbf{x}_i}) + \frac{1}{2}\mathbf{w}^T\mathbf{w}, \tag{2.4}$$

na qual $C > 0$ é um parâmetro de penalidade (YU; HUANG; LIN, 2011).

2.3 *ACTIVE LEARNING*

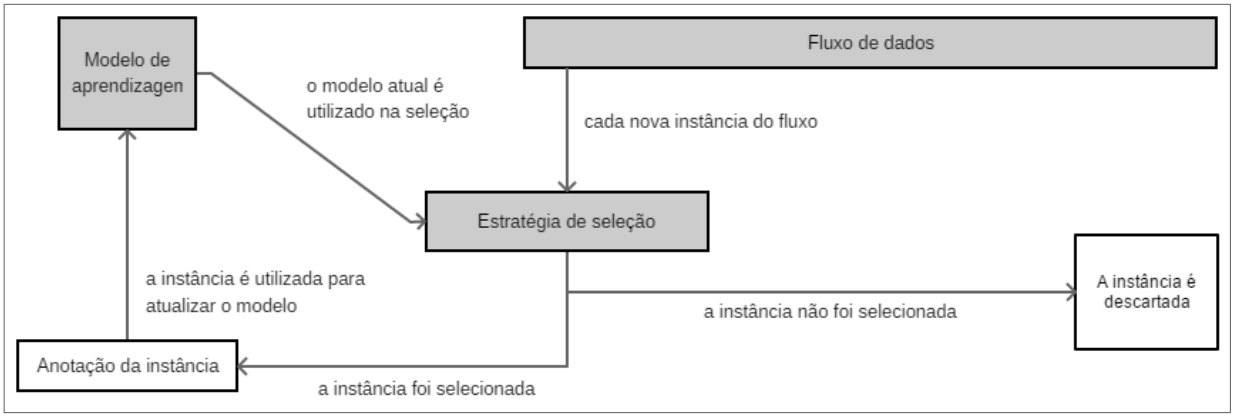
As restrições de recursos de memória e tempo, explanadas na Seção 2.1, resultaram no desenvolvimento e no uso de novos tipos de abordagens para lidar com *data streams* (KRAWCZYK et al., 2017). Alguns deles são baseados em estratégias de seleção (*sampling*), como o *Active Learning* (AL).

Active Learning é uma abordagem semi-supervisionada comumente utilizada para Mineração de Dados com *data streams* (ZHU et al., 2007), devido à sua competência em lidar

de maneira eficaz com problemas nos quais dados rotulados são custosos de obter (ZIMMERMANN; NTOUTSI; SPILIOPOULOU, 2015).

O método AL usa um conjunto inicial de instâncias rotuladas como o primeiro conjunto de treinamento do modelo. Então, uma estratégia de seleção é utilizada para solicitar o rótulo de novas instâncias a fim de atualizar o modelo de aprendizagem. Estas novas instâncias são aquelas consideradas pela estratégia de seleção mais adequadas para o problema (ZIMMERMANN; NTOUTSI; SPILIOPOULOU, 2015). A Figura 1 traz o processo de *Active Learning* para um processamento online e iterativo, isto é, esse processo é repetido para cada nova instância do fluxo de dados.

Figura 1 – Processo de *Active Learning* para processamento online.



Fonte: próprio autor (2020).

Portanto, o principal desafio na área de *Active Learning* é identificar a estratégia de seleção que alcança o melhor desempenho, ao passo que mantém o conjunto de treinamento o menor possível, isto é, que seleciona apenas as melhores instâncias (ZHU et al., 2007). Além disso, essa estratégia deve estar apta a balancear o conjunto de dados rotulados através do tempo e perceber a ocorrência de *drifts*, preservando a distribuição dos dados de entrada para detectá-los (ŽLIOBAITĖ et al., 2011).

Como estratégias de seleção, é comum a utilização de algumas heurísticas (ou regras) para selecionar as instâncias mais valiosas (ZHU et al., 2010). No Capítulo 3, são explanadas as estratégias utilizadas neste trabalho.

2.4 TRABALHOS RELACIONADOS

Duas estratégias de seleção de *Active Learning* foram propostas por Žliobaitė et al. (2011). Elas são baseadas na incerteza do classificador, na alocação dinâmica de esforços ao longo do tempo e na aleatorização do espaço de busca, o que, segundo os autores, consegue lidar explicitamente com *concept drifts*, além de balancear o conjunto de treinamento. Conjuntos de dados de *benchmark* contendo *data streams*, além de dados textuais oriundos do IMDb e da Reuters, foram utilizados para realizar experimentos e avaliar as técnicas

propostas, concluindo que estas estratégias são eficazes, principalmente quando o conjunto anotado é pequeno; porém, os autores não realizaram Análise de Sentimento.

Já Smailović et al. (2014) usaram o classificador SVM e combinaram as vantagens de duas estratégias de seleção: *Uncertainty* e *Random Sampling*, além de utilizá-las separadamente. Eles realizaram Mineração de Opinião utilizando *data streams* financeiros oriundos do Twitter acerca de oito companhias (Apple, Amazon, Baidu, Cisco, Google, Microsoft, Netflix e RIM), em dois conjuntos de experimentos: 1) considerando apenas as classes *positiva* e *negativa*, e 2) considerando também a classe *neutra*. O objetivo dos autores era encontrar a melhor estratégia de consulta para *Active Learning* neste cenário de Análise de Sentimento, além de investigar se o *feed* do Twitter é adequado para a análise preditiva do mercado de ações. Com os experimentos, eles puderam concluir que, usando a abordagem de AL, o poder de predição do classificador de sentimentos para esta aplicação foi melhorado e que realizar MO com dados do Twitter é uma boa opção para prever o movimento de preços de ações.

A estratégia *Uncertainty* também foi utilizada por Zimmermann, Ntoutsi e Spiliopoulou (2015) para selecionar os documentos que iriam atualizar o modelo de aprendizagem de um classificador de polaridade e as palavras que iriam ser adicionadas a um vocabulário de apoio. Além disso, eles propuseram uma estratégia baseada no ganho de informação provido pelo documento ao modelo. Essas duas técnicas de *Active Learning* foram comparadas com a estratégia *Random Sampling*, com uma abordagem incremental que necessita do rótulo de todos os documentos e com um método não-adaptativo, todos eles utilizando o classificador MNB. Os resultados mostraram que a utilização de AL apresenta um bom desempenho e um baixo número de rótulos solicitados. Além disso, a estratégia *Information Gain* (ganho de informação) obteve bons resultados em ambas as bases de dados utilizadas (*TwitterSentiment* e *StreamJi*), considerando a métrica *kappa statistic* e o número de rótulos solicitados; enquanto a *Uncertainty* apresentou um desempenho inferior.

Kranjc et al. (2015) criaram um *framework* denominado *ClowdFlows*, o qual possui extensões para análise de *data streams* e *Active Learning*, além de permitir a análise de dados em tempo real. No seu trabalho, eles apesentaram um caso de uso de Análise de Sentimento com dados financeiros oriundos do Twitter, utilizando um classificador SVM linear e combinando as estratégias *Uncertainty* e *Random Sampling*, variando a proporção de dados aleatórios e de dados próximos da fronteira de decisão (dados com maior nível de incerteza); além de comparar o desempenho dos modelos que utilizam AL ao desempenho de um modelo estático. Os resultados alcançados pela combinação de 75% de dados aleatórios com 25% de dados "incertos" foram levemente superiores, ao passo que pode-se afirmar que a estratégia *Random Sampling* (com 100% de dados aleatórios) obteve um desempenho melhor que a *Uncertainty* (100% de dados com alto nível de incerteza).

Já Aldoğan e Yaslan (2017) incorporaram a um *framework* existente, uma metodologia de *Active Learning* baseada na estratégia *Query By Committee* (QBC), a qual mantém um comitê de diferentes algoritmos, todos treinados no mesmo conjunto de dados, e seleciona, para atualização do modelo, as instâncias para as quais os algoritmos mais discordaram. Os autores avaliaram três abordagens para construção do comitê: escolha aleatória, *Shannon Entropy* e *Maximum Disagreement*; realizando experimentos utilizando corpora de avaliações de filmes e avaliações de produtos e concluindo que a *Shannon Entropy* se apresentou como a melhor abordagem para QBC nos cenários avaliados.

Por fim, Li et al. (2019) também propuseram um *framework* para classificação de sentimentos. Eles consideraram três técnicas semi-supervisionadas: *Self-learning*, a qual anota automaticamente dados não rotulados, adicionando as instâncias com maior nível de certeza ao conjunto de treinamento; *Co-training*, que utiliza modelos treinados com duas visões do mesmo conjunto de dados, na qual os exemplos classificados com maior certeza por um classificador são adicionadas ao conjunto de treinamento do outro; e *Active Learning*. As três técnicas foram combinadas em um processo iterativo para atualização do modelo e este método foi comparado com diversas outras abordagens, incluindo uma baseada em léxicos e uma utilizando apenas *Active Learning* com a estratégia *Uncertainty*; sendo o método proposto superior a todos os outros em cinco bases de dados de avaliações de livros, DVDs, eletrônicos, artigos para cozinha e filmes.

Embora não se possa comparar os resultados dos trabalhos apresentados nesta seção, já que estes realizaram experimentos em cenários distintos, o Quadro 1 apresenta uma sumarização deles, comparando as estratégias de *Active Learning* utilizadas, os algoritmos adotados e a fonte de suas bases de dados.

Fonte: próprio autor (2020).

Trabalho relacionado	**Estratégias de AL**	**Algoritmos**	**Fonte de dados**	**Resumo**
Žliobaitė et al. (2011)	*Random Sampling*, *Uncertainty*, *Variable Uncertainty* e *Variable Randomized Uncertainty*	Naïve Bayes	IMDb (filmes) e Reuters (artigos jurídicos)	Propuseram duas novas estratégias de AL baseadas em incerteza, usaram bases textuais, porém sem realizar MO.
Smailović et al. (2014)	*Random Sampling* e *Uncertainty*	SVM	Twitter (mercado financeiro)	Realizaram Mineração de Opinião com uma abordagem de AL incremental para predizer preços no mercado de ações.
Zimmermann, Ntoutsi e Spiliopoulou (2015)	*Random Sampling*, *Uncertainty* e *Information Gain*	MNB	TwitterSentiment (múltiplos domínios) e StreamJi (avaliações de produtos)	Compararam estratégias de AL a um método incremental e a um estático, para classificação de polaridade.
Kranjc et al. (2015)	*Random Sampling* e *Uncertainty*	SVM	Twitter (mercado financeiro)	Criaram um *framework* que possibilita a análise de sentimento combinando técnicas de AL, o qual foi avaliado em um caso de uso offline.
Aldoğan e Yaslan (2017)	*Query by Committee*	MNB, SMO, Voted Perceptron, k-NN, Random Forest...	Cornell MR (avaliações de filmes) e Amazon (avaliações de produtos)	Avaliaram três métodos para formação do comitê de algoritmos da estratégia de *Active Learning* QBC.
Li et al. (2019)	*Uncertainty*	SVM, Maximum Entropy	Weibo (múltiplos domínios), Hotel review (avaliações de hotéis), Amazon (avaliações de produtos) e Cornell MR (avaliações de filmes)	Combinaram três estratégias semi-supervisionadas, incluindo *Active Learning*, para análise de sentimento em dados de avaliações de diversos domínios.

Quadro 1 – Sumarização dos trabalhos relacionados.

3 ESTRATÉGIAS DE SELEÇÃO DE *ACTIVE LEARNING*

3.1 ESTRATÉGIAS ENCONTRADAS NA LITERATURA

3.1.1 *Random Sampling*

A estratégia de seleção ***Random Sampling*** seleciona as instâncias aleatoriamente, em vez de decidir quais delas são mais relevantes para o modelo. Ela se baseia em uma probabilidade B, a qual representa o *budget* e determina a porcentagem do fluxo de dados que será selecionada (ŽLIOBAITĖ et al., 2011). Neste trabalho, a implementação desta estratégia seguiu o Algoritmo 1.

Algoritmo 1: *Random Sampling*

Entrada: classificador C, conjunto semente S, fluxo de dados F, *budget* B

```
1 treina o classificador C com o conjunto S;
2 foreach d in F do
3     n = gera número aleatório entre 0 e 1;
4     if n <= B then
5         atualiza o classificador C com o documento d;
6 return C;
```

Fonte: próprio autor (2020).

3.1.2 *Uncertainty*

A estratégia ***Uncertainty***, proposta por Lewis e Gale (1994), é uma das mais utilizadas para *Active Learning* (SETTLES, 2009). Sua ideia é rotular as instâncias para as quais o classificador é menos confiante, ou seja, as instâncias para as quais ele esteja mais incerto de sua classe (ŽLIOBAITĖ et al., 2011).

Considerou-se o Algoritmo 2 para implementação desta estratégia, no qual a incerteza para uma instância pode ser calculada pela Equação 3.1:

$$U(x) = 1 - P(\hat{x}|x), \tag{3.1}$$

na qual x é a instância, $\hat{x}$ é a predição para x com o maior nível de certeza e $P(\hat{x}|x)$ é a probabilidade de x pertencer à classe predita (DANKA; HORVATH, 2018b).

O parâmetro θ determina o limiar de certeza, então as instâncias que tiverem uma incerteza maior ou igual a 1 - θ, isto é, que não atingiram o limiar, são selecionadas. Sendo assim, quanto maior θ, mais instâncias tendem a ser escolhidas.

Algoritmo 2: *Uncertainty*

Entrada: classificador C, conjunto semente S, fluxo de dados F, limiar θ

```
1 treina o classificador C com o conjunto S;
2 foreach d in F do
3     uncertainty = calcula a incerteza do classificador C para o documento d;
4     if uncertainty >= 1 - θ then
5         atualiza o classificador C com o documento d;
6 return C;
```

Fonte: próprio autor (2020).

3.1.3 *Variable Uncertainty*

Žliobaitė et al. (2011) propuseram duas alterações na estratégia *Uncertainty*. A primeira delas, denominada ***Variable Uncertainty*** altera o valor do limiar θ utilizando um passo de ajuste s, com o objetivo de adaptá-lo aos dados que chegam com o fluxo, expandindo-o e contraindo-o. Quando uma instância é selecionada, o limiar diminui; ao passo que, caso se passe muito tempo sem que alguma instância seja escolhida, o limiar tende a aumentar.

A implementação seguiu o Algoritmo 3, o qual foi adaptado do trabalho original de Žliobaitė et al. (2011) e utilizando a medida de incerteza dada pela Equação 3.1.

Algoritmo 3: *Variable Uncertainty*

Entrada: classificador C, conjunto semente S, fluxo de dados F, limiar θ, passo de ajuste s

```
1 treina o classificador C com o conjunto S;
2 foreach d in F do
3     uncertainty = calcula a incerteza do classificador C para o documento d;
4     if uncertainty >= 1 - θ then
5         atualiza o classificador C com o documento d;
6         θ = θ(1 − s);                          /* o limiar diminui */
7     else
8         θ = θ(1 + s);                  /* aumenta a região de incerteza */
9 return C;
```

Fonte: adaptado de Žliobaitė et al. (2011).

3.1.4 *Variable Randomized Uncertainty*

A segunda estratégia proposta por Žliobaitė et al. (2011), denominada ***Variable Randomized Uncertainty***, utiliza, além do passo de ajuste, uma ferramenta de aleatoriza-

ção do limiar, por meio de sua multiplicação por uma variável aleatória oriunda de uma distribuição normal que segue $\mathcal{N}(1,\ \delta)$. O Algoritmo 4 apresenta o pseudocódigo desta estratégia, a qual também utiliza a medida de incerteza apresentada na Equação 3.1.

Os autores afirmam que tanto a estratégia *Variable Uncertainty* quanto a *Variable Randomized Uncertainty* conseguem reagir bem a *drifts* que ocorram em qualquer lugar do espaço de instâncias, o que as torna úteis para *data streams*.

Algoritmo 4: *Variable Randomized Uncertainty*

Entrada: classificador C, conjunto semente S, fluxo de dados F, limiar θ, passo de ajuste s, variância da aleatorização δ

```
1  treina o classificador C com o conjunto S;
2  foreach d in F do
3      θ_random = θ × η;      /* onde η ∈ N(1,δ) é um multiplicador aleatório */
4      uncertainty = calcula a incerteza do classificador C para o documento d;
5      if uncertainty >= 1 - θ_random then
6          atualiza o classificador C com o documento d;
7          θ = θ(1 − s);                                  /* o limiar diminui */
8      else
9          θ = θ(1 + s);                        /* aumenta a região de incerteza */
10 return C;
```

Fonte: adaptado de Žliobaitė et al. (2011).

3.1.5 *Information Gain*

A estratégia ***Information Gain*** foi proposta por Zimmermann, Ntoutsi e Spiliopoulou (2015) e é específica para Mineração de Texto, já que utiliza a distribuição de palavras/classe e mantém um vocabulário. Nela, os documentos que proverem um ganho de informação ao modelo são selecionados.

O ganho de informação é calculado considerando a distribuição, observada até o momento, de palavras/classe das palavras de um determinado documento e a distribuição após considerar o rótulo de classe predito pelo modelo para aquele documento, seguindo a Equação 3.2:

$$IG(d) = \sum_{w_i \in d \wedge \in V} H(N_{i+}, N_{i-}) - H(N_{i+} + 1, N_{i-}), \tag{3.2}$$

neste exemplo com duas classes (+, -), a classe predita para o documento d foi a positiva (+). w_i representa cada palavra contida em d, V é o vocabulário, N_{i+} é a quantidade de ocorrências da palavra w_i na classe +, N_{i-} é a quantidade de ocorrências da palavra w_i

na classe - e $H(a, b)$ é o cálculo de entropia de dois valores positivos $a, b \in \mathbb{N}$, explanado na Equação 3.3 (ZIMMERMANN; NTOUTSI; SPILIOPOULOU, 2015):

$$H(a,b) = -\left[\frac{a}{a+b} * log_2(\frac{a}{a+b}) + \frac{b}{a+b} * log_2(\frac{b}{a+b})\right] \tag{3.3}$$

O Algoritmo 5 apresenta a sequência de passos para seleção das instâncias e atualização do modelo de classificação por meio desta estratégia.

Algoritmo 5: *Information Gain*

Entrada: classificador C, conjunto semente S, fluxo de dados F

```
1  treina o classificador C com o conjunto S;
2  V = extrai todas as palavras contidas em S;
3  foreach d in F do
4      pred = rótulo do documento d atribuído pelo classificador C;
5      informationGain = calcula o ganho de informação para o modelo
         considerando o rótulo pred;
6      if informationGain > 0 then
7          c = rótulo correto do documento d;
8          foreach p ∈ d do
9              if p ∈ V then
10                 atualiza a contagem de ocorrências da palavra p na classe c;
11             else
12                 V = V ∪ p;                    /* incrementa o vocabulário */
13         atualiza o classificador C com o documento d;
14 return C;
```

Fonte: adaptado de Zimmermann, Ntoutsi e Spiliopoulou (2015).

3.1.6 *Entropy*

A medida de entropia pode ser utilizada como uma medida de incerteza em estratégias de AL (DANKA; HORVATH, 2018b; YANG; LOOG, 2018). Assim, neste estudo, a estratégia ***Entropy*** foi utilizada seguindo a implementação de Lewis e Gale (1994) e de forma bem similar à estratégia *Uncertainty*, apenas substituindo uma medida pela outra.

Neste caso, uma instância é selecionada caso a entropia calculada para ela seja maior ou igual a 1 - θ, ou seja, a estratégia seleciona as instâncias com maior entropia (*maximum entropy*), como pode-se observar no Algoritmo 6. Nele, a entropia pode ser calculada pela Equação 3.4:

$$H(x) = -\sum_k p_k \log(p_k), \tag{3.4}$$

na qual, x é a instância e p_k é a probabilidade de x pertencer à classe k (DANKA; HORVATH, 2018b).

Algoritmo 6: *Entropy*

Entrada: classificador C, conjunto semente S, fluxo de dados F, limiar θ

```
1 treina o classificador C com o conjunto S;
2 foreach d in F do
3     entropy = calcula a entropia do classificador C considerando o documento d;
4     if entropy >= 1 - θ then
5         atualiza o classificador C com o documento d;
6 return C;
```

Fonte: próprio autor (2020).

Embora não tenham sido encontrados trabalhos que utilizassem a estratégia *Entropy* em cenários de Mineração de Opinião com *data streams*, optou-se pela avaliação desta estratégia devido ao seu bom desempenho no trabalho de Yang e Loog (2018), os quais a utilizaram em substituição à estratégia *Uncertainty*.

3.2 ESTRATÉGIAS PROPOSTAS

Como apontado anteriormente, Žliobaitė et al. (2011) afirmam que a variação do limiar θ utilizando um mecanismo de aleatorização e/ou um passo de ajuste faz com que a estratégia *Uncertainty* se adapte melhor aos dados do fluxo, reagindo com eficiência à ocorrência de *drifts* e sendo adequada para mineração com *data streams*. Além disso, no trabalho de Yang e Loog (2018), a seleção de instâncias com *maximum entropy*, considerada pelos autores uma medida de incerteza, se mostrou a mais promissora, superando outras técnicas de seleção bem mais complexas.

Por conta disso, pensou-se em duas questões:

- ***Os benefícios da variação do limiar, em termos de desempenho, também estão presentes utilizando a medida de entropia?***
- ***A utilização da medida de entropia pode melhorar o desempenho das estratégias baseadas na variação do limiar?***

Assim, propôs-se a utilização das estratégias apresentadas por Žliobaitė et al. (2011) com a medida de máxima entropia (Equação 3.4) substituindo a medida comum de incerteza (Equação 3.1). Já que a estratégia *Entropy* possui um limiar semelhante ao da estratégia *Uncertainty*, a variação deste limiar tende a ser benéfica para a entropia da mesma forma que foi para a medida de incerteza: adaptando-o aos novos dados e reagindo a possíveis mudanças de conceito. Além disso, espera-se que essas mudanças balan-

ceiem a quantidade de instâncias selecionadas, escolhendo apenas aquelas cujo alto grau de entropia seja benéfico para o modelo.

Estas combinações deram origem a duas novas estratégias de seleção de *Active Learning*, as quais são propostas nesta dissertação com o objetivo de responder às duas questões levantadas.

3.2.1 *Variable Entropy*

A primeira delas, denominada ***Variable Entropy***, combina a variação de θ utilizando o passo de ajuste (s), trazido pela estratégia *Variable Uncertainty*, com a medida de entropia, e é implementada de acordo com o Algoritmo 7.

Algoritmo 7: *Variable Entropy*

Entrada: classificador C, conjunto semente S, fluxo de dados F, limiar θ, passo de ajuste s

```
1 treina o classificador C com o conjunto S;
2 foreach d in F do
3     entropy = calcula a entropia do classificador C considerando o documento d;
4     if entropy >= 1 - θ then
5         atualiza o classificador C com o documento d;
6         θ = θ(1 − s);                          /* o limiar diminui */
7     else
8         θ = θ(1 + s);                  /* aumenta a região de entropia */
9 return C;
```

Fonte: próprio autor (2020).

Dessa forma, o limiar de entropia é modificado para cada nova instância: caso a instância tenha sido selecionada, o limiar diminui; do contrário, o limiar aumenta, adaptando-se assim a mudanças na distribuição dos dados. Quanto mais instâncias tenham sido escolhidas, mais dificilmente um novo exemplo que não possua uma entropia muito alta será selecionado; e vice versa.

3.2.2 *Variable Randomized Entropy*

Por fim, a segunda estratégia proposta nesta dissertação chama-se ***Variable Randomized Entropy*** e combina a aleatorização do limiar θ apresentado pela estratégia *Variable Randomized Uncertainty* (ŽLIOBAITĖ et al., 2011) com a medida de entropia (Equação 3.4), sendo explanada pelo Algoritmo 8.

A diferença desta técnica para a *Variable Entropy* é que, nesta, além da utilização do passo de ajuste (s), o limiar de entropia sofre um processo de aleatorização a cada novo

documento que chega no fluxo de dados: multiplica-se o limiar atual por uma variável aleatória, gerando um θ_{random}, o qual é utilizado para definir se a instância será selecionada ou não. O valor de θ é, então, modificado da mesma forma que na estratégia *Variable Entropy*, dependendo se a instância foi escolhida ou descartada.

Algoritmo 8: *Variable Randomized Entropy*

Entrada: classificador C, conjunto semente S, fluxo de dados F, limiar θ, passo de ajuste s, variância da aleatorização δ

```
1  treina o classificador C com o conjunto S;
2  foreach d in F do
3      θ_random = θ × η;      /* onde η ∈ N(1, δ) é um multiplicador aleatório */
4      entropy = calcula a entropia do classificador C considerando o documento d;
5      if entropy >= 1 - θ_random then
6          atualiza o classificador C com o documento d;
7          θ = θ(1 − s);                                  /* o limiar diminui */
8      else
9          θ = θ(1 + s);                        /* aumenta a região de entropia */
10 return C;
```

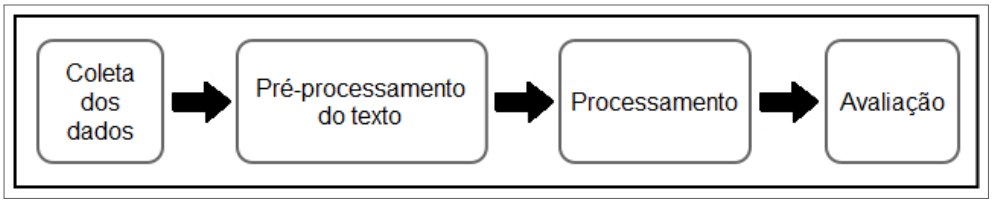
Fonte: próprio autor (2020).

4 METODOLOGIA EXPERIMENTAL

Para esta dissertação, foi realizada análise de sentimento a nível de documento, uma vez que foram utilizadas bases de dados que, embora sejam constituídas por textos considerados curtos, possuem documentos que podem conter mais de uma sentença.

A Figura 2 mostra o método utilizado para a aplicação de mineração apresentada neste trabalho. Ele é dividido em quatro etapas: *Coleta de dados*, na qual os dados, neste caso documentos de texto, são obtidos e anotados; *Pré-processamento do texto*, na qual os dados textuais são estruturados antes do processamento; *Processamento*, na qual ocorre o treinamento dos algoritmos e a posterior classificação dos documentos; e *Avaliação*, na qual obtém-se os resultados das classificações.

Figura 2 – Método de mineração utilizado.



Fonte: próprio autor (2020).

Este método constituído por quatro etapas costuma ser o mais utilizado para Mineração de Texto (FELDMAN; SANGER, 2006). Com relação à primeira etapa, neste estudo foram utilizados corpora já coletados e prontos para o uso, porém também foi realizada uma coleta própria, com o objetivo de aumentar o número de bases de dados nas quais os experimentos viriam a ser avaliados. O detalhamento desta coleta é feito na seção seguinte.

4.1 BASES DE DADOS

Ao todo, foram utilizados 20 conjuntos de dados, oriundos de quatro corpora diferentes contendo *data streams* de mídias sociais. Dois dos corpora foram encontrados na Internet, estando publicamente disponíveis para uso; enquanto os outros dois foram construídos pelo próprio autor contendo dados acerca da Eleição Presidencial no Brasil em 2018, tendo sido também disponibilizados publicamente[1].

4.1.1 Bases da Eleição Presidencial 2018

Devido à falta de bases de dados para a Língua Portuguesa, principalmente no que se diz respeito a *data streams*, e à importância da Mineração de Opinião para o domínio

[1] http://miningbrgroup.com.br/index.php/resources/

político, optou-se pela construção de dois corpora acerca do segundo turno da Eleição Presidencial Brasileira de 2018, o qual envolveu os candidatos Jair Bolsonaro (PSL) e Fernando Haddad (PT): um corpus contendo *tweets* e o outro comentários do Facebook.

4.1.1.1 Extração dos dados

Durante o segundo turno da Eleição Presidencial, seis debates televisionados entre os dois candidatos estavam agendados: para os dias 11, 14, 15, 17, 21 e 26 de Outubro de 2018. O planejamento inicial seria coletar *tweets* e comentários do Facebook durante a realização dos debates, porém estes não ocorreram.

Entretanto, ainda assim decidiu-se realizar a coleta nos dias agendados para os debates. No Twitter, foram coletadas opiniões que contivessem menções aos candidatos e/ou *hashtags* com os nomes deles. Já do Facebook, foram extraídos os comentários de notícias acerca dos candidatos, totalizando 20000 opiniões. Ambos os processos de coleta foram realizados automaticamente: utilizando a biblioteca Python Tweepy[2] para o Twitter, e a bilbioteca Requests[3] para acessar a Graph API do Facebook.

4.1.1.2 Anotação manual

Dentre os dados coletados, foram selecionados aleatoriamente 2000 comentários do Facebook e 1500 *tweets*. Após a seleção, os documentos foram classificados de acordo com o candidato ao qual se referiam: Fernando Haddad (ou ao partido PT), Jair Bolsonaro (ou ao partido PSL) ou a ambos. Por fim, cada comentário teve sua opinião referente a cada candidato classificada em *positiva* ou *negativa*; ou seja, os documentos que possuíam opiniões acerca de ambos foram classificados com duas polaridades. O Quadro 2 traz amostras de comentários anotados de acordo com a polaridade de cada candidato.

A classificação de polaridade foi realizada por três anotadores: o autor deste trabalho e dois outros membros do grupo de pesquisa MiningBR[4], com o objetivo de minimizar classificações erradas, de forma que dois anotadores diferentes se responsabilizaram por cada documento. Nos casos em que houve discordância entre os dois anotadores, isto é, nos quais cada anotador classificou o documento com uma polaridade diferente, eles discutiram a anotação para chegar a um consenso. Além disso, foi computado o coeficiente Kappa de Cohen (COHEN, 1960) para cada base criada, visando a descoberta do nível de concordância entre os anotadores.

Por fim, foram obtidas seis bases de dados, que diferiam entre si pela fonte dos dados: Twitter ou Facebook; e pelo candidato a qual se referiam: Jair Bolsonaro, Fernando Haddad ou Ambos. Entretanto, os documentos que se referiam a ambos os candidatos ao mesmo tempo foram excluídos das bases *Ambos*, pelo fato de o objetivo deste trabalho

[2] https://www.tweepy.org
[3] https://requests.readthedocs.io/en/master/
[4] http://miningbrgroup.com.br

Quadro 2 – Exemplos de comentários anotados.

Comentário	**Pol. Bolsonaro**	**Pol. Haddad**
Só o @jairbolsonaro pra fazer isso kkkk, PT perdendo suas origens. #PTNunaMais #bolsonaro	positiva	negativa
#Haddad defende o trabalhador com direito a 13º, férias e muito mais. O 17 não. Por isso voto 13.	negativa	positiva
Minha solidariedade com o povo brasileiro ! O #fascismo será derrotado. #FascimoNão #BolsonaroFujão #Bolsonaro	negativa	-
#Bolsonaro Presidente do Brasil	positiva	-
Partido dos Trabalhadores quesito hipocrisia nota 10. #pt #haddad #manueladavila #abortonao #hipocrisia	-	negativa
É #Haddad e #Manu para presidente!	-	positiva

Fonte: próprio autor (2020).

não ser realizar uma classificação multi-classe; sendo assim, estas bases foram constituídas de documentos acerca de Jair Bolsonaro e documentos acerca de Fernando Haddad, porém não possuindo nenhum documento referente aos dois ao mesmo tempo. A Tabela 1 apresenta os detalhes das bases de dados construídas.

Fonte: próprio autor (2020).

#	**Candidato**	**Fonte**	**Quant**	**#positiva**	**#negativa**	**Kappa**
1.	Jair Bolsonaro	Facebook	1,047	817	230	0.9439
2.	Jair Bolsonaro	Twitter	861	484	377	0.7404
3.	Fernando Haddad	Facebook	1,046	440	606	0.9528
4.	Fernando Haddad	Twitter	762	298	464	0.8370
5.	Ambos	Facebook	1,907	1,167	740	0.9579
6.	Ambos	Twitter	1,377	675	702	0.7805

Tabela 1 – Detalhes dos corpora criados.

Dois pontos interessantes que pode-se observar com base na Tabela 1 são: 1) o coeficiente Kappa maior (em torno de 0.95) para as bases do Facebook mostra que os documentos oriundos desta mídia social foram mais fáceis de anotar, o que pode ter um impacto no desempenho dos classificadores para essas bases, já que elas podem ser mais fáceis de classificar também para os modelos de Aprendizagem de Máquina; 2) o fato de o candidato Jair Bolsonaro ter tido mais comentários positivos que negativos, principalmente no Facebook, ao contrário do candidato Fernando Haddad, que apresentou uma rejeição maior. Este segundo ponto refletiu o resultado final da Eleição, na qual o candidato do PSL foi eleito, mostrando o potencial das mídias sociais para previsão do resultado das eleições.

4.1.2 *Sentiment140* e *Sanders*

Além daqueles criados para este projeto, fez-se uso de outros dois corpora de *benchmark*, amplamente utilizados para a Língua Inglesa, contendo *data streams* do Twitter: *Sentiment140* (GO; BHAYANI; HUANG, 2009) e *Sanders* (SANDERS, 2011). Os estudos de Smailović et al. (2014), Wagner et al. (2015) e Zimmermann, Ntoutsi e Spiliopoulou (2015) realizaram Mineração de Opinião com *data streams* utilizando o primeiro corpus, enquanto Aston, Liddle e Hu (2014) e Aston et al. (2014) utilizaram o segundo.

O conjunto de treinamento do corpus *Sentiment140* contém 1600000 *tweets* anotados automaticamente, o que seria extremamente custoso de processar; então foram selecionados subconjuntos de 1000, 2500, 5000 e 10000 *tweets* de dois pontos do fluxo de dados (os conjuntos *Sentiment140_1000_1*, *Sentiment140_2500_1*, *Sentiment140_5000_1* e *Sentiment140_10000_1* foram extraídos do primeiro ponto do fluxo; já *Sentiment140_1000_2*, *Sentiment140_2500_2*, *Sentiment140_5000_2* e *Sentiment140_10000_2* foram extraídos do segundo ponto), com o objetivo de avaliar as estratégias de seleção de *Active Learning* em bases de diferentes tamanhos e possivelmente sob efeito de diferentes *drifts*. O segundo ponto do fluxo foi inspirado pelo trabalho de Zimmermann, Ntoutsi e Spiliopoulou (2015). Além disso, também foi utilizado o conjunto de "treino" do *Sentiment140* (*Sentiment140_test*), o qual contém 497 *tweets* anotados manualmente.

Já o corpus *Sanders* contém *tweets* acerca de quatro organizações: Apple, Google, Microsoft e Twitter. Sendo assim, utilizou-se cada subconjunto referente a cada empresa como um conjunto de dados distinto (*Sanders_apple*, *Sanders_google*, *Sanders_microsoft* e *Sanders_twitter*), além de ter sido utilizado o corpus completo (*Sanders_all*). Essa partição proporcionou um número maior de bases para avaliação das estratégias; ademais, um *data stream* geralmente se refere a apenas uma entidade.

Assim, além das seis bases construídas com os dados da Eleição Presidencial, foram selecionados e utilizados mais 14 conjuntos de dados (nove oriundos do *Sentiment140* e cinco do *Sanders*), os quais são detalhados na Tabela 2, que traz a quantidade total de *tweets* por base, a quantidade de *tweets* positivos (#PO), negativos (#NG) e neutros (#NT), e o intervalo do fluxo do corpus *Sentiment140* do qual os dados foram extraídos. Como pode-se observar, as bases se diferem por tamanho, número de classes, domínio e nível de desbalanceamento.

Fonte: próprio autor (2020).

#	Base	Quant	#PO	#NG	#NT	Intervalo
7.	Sentiment140_test	497	182	177	138	-
8.	Sentiment140_10000_1	10,000	5,812	4,188	-	25,000-35,000
9.	Sentiment140_5000_1	5,000	2,970	2,030	-	35,000-40,000
10.	Sentiment140_2500_1	2,500	1,461	1,039	-	40,000-42,500
11.	Sentiment140_1000_1	1,000	579	421	-	42,500-43,500
12.	Sentiment140_10000_2	10,000	5,602	4,398	-	1,235,000-1,245,000
13.	Sentiment140_5000_2	5,000	2,821	2,179	-	1,245,000-1,250,000
14.	Sentiment140_2500_2	2,500	1,421	1,079	-	1,250,000-1,252,500
15.	Sentiment140_1000_2	1,000	572	428	-	1,252,500-1,253,500
16.	Sanders_apple	1,002	164	316	522	-
17.	Sanders_google	838	202	57	579	-
18.	Sanders_microsoft	864	91	132	641	-
19.	Sanders_twitter	719	62	67	590	-
20.	Sanders_all	3,423	519	572	2,332	-

Tabela 2 – Detalhes dos corpora utilizados.

4.2 PRÉ-PROCESSAMENTO

A implementação da aplicação de Mineração de Opinião foi realizada com a linguagem de programação Python, tendo sido utilizadas as bibliotecas *Natural Language Toolkit* (NLTK)[5] e *scikit-learn*[6] para a etapa de pré-processamento dos dados textuais.

Para estruturar os dados em um Modelo de Espaço Vetorial, fez-se uso do método *CountVectorizer* da biblioteca *scikit-learn*, pela sua simplicidade; enquanto o *TweetTokenizer* da biblioteca NLTK foi utilizado para a tokenização dos documentos, isto é, para a divisão do texto em segmentos menores denominados *tokens* (WEISS et al., 2005).

A tokenização divide o documento utilizando um delimitador geral, o qual pode ser os espaços em branco entre as palavras; o *TweetTokenizer*, por sua vez, e por ser específico para o Twitter, separa partes específicas dos *tweets*, como *hashtags*, identificadores de usuários, pontuações e *emoticons*. E, como o Facebook incorporou diversas características do Twitter (*hashtags*, por exemplo), optou-se por utilizar o *TweetTokenizer* também para os textos dessa mídia social. O Quadro 3 traz uma demonstração deste processo em um comentário extraído do Facebook.

De forma geral, foram utilizadas as técnicas de pré-processamento mais simples possíveis, ou seja, apenas as técnicas essenciais para o processamento de documentos de texto. Já que o foco deste estudo se concentra na avaliação das estratégias de *Active Learning*.

[5] https://www.nltk.org
[6] https://scikit-learn.org/stable/

Quadro 3 – Exemplo do processo de tokenização.

Entrada	[“avante #bolsonaro17. brasil acima de tudo e deus acima de todos!!”]
Saída	[‘avante’, ‘#bolsonaro17’, ‘.’, ‘brasil’, ‘acima’, ‘de’, ‘tudo’, ‘e’, ‘deus’, ‘acima’, ‘de’, ‘todos’, ‘!’, ‘!’]

Fonte: próprio autor (2020).

4.3 PROCESSAMENTO

Os algoritmos de Aprendizagem de Máquina Supervisionada escolhidos para este trabalho foram aqueles descritos na Seção 2.2.1: *Multinomial Naïve Bayes* (MNB), *Support Vector Machine* (SVM) e de Regressão Logística. Para os três algoritmos, suas respectivas implementações oriundas do *scikit-learn* foram utilizadas: *MultinomialNB()*, *SVC()* e *LogisticRegression()*, sem mudanças nos parâmetros padrão, exceto para o SVM, para o qual optou-se por um *kernel* linear, utilizado nos trabalhos de Smailović et al. (2014) e Kranjc et al. (2015).

Já para implementar as estratégias de seleção de *Active Learning*, contou-se com o auxílio da biblioteca Python *modAL* (DANKA; HORVATH, 2018a), que é própria para AL e possui a implementação das medidas de incerteza e entropia. Todas as estratégias foram implementadas seguindo seus respectivos algoritmos originais, detalhados no Capítulo 3.

Para esta pesquisa, optou-se por uma abordagem implícita como forma de lidar com os *drifts*, utilizando a atualização constante do modelo de aprendizagem para este fim. Portanto, não foi utilizado nenhum mecanismo para detectar explicitamente *drifts* nas bases de dados.

4.4 AVALIAÇÃO

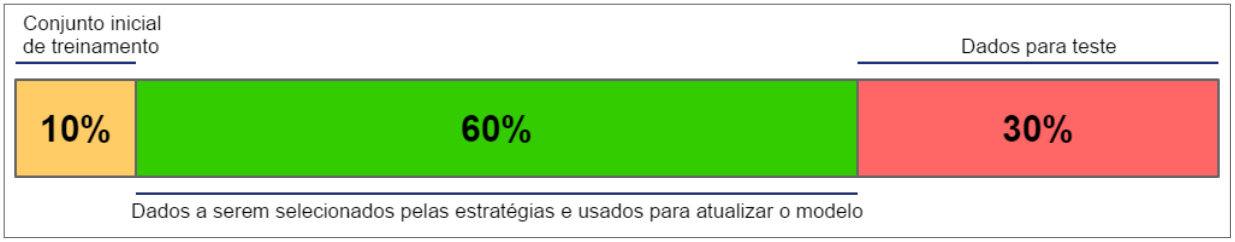
Devido à natureza contínua e sequencial dos *data streams*, optou-se por não utilizar a técnica de validação cruzada *k-fold*, a qual é bastante utilizada para Mineração de Texto (SOUZA et al., 2018). Como a técnica *k-fold* aleatoriza os dados, optou-se pela técnica *holdout* que permite realizar a divisão de base de dados em treinamento e teste de forma a manter a sequencialidade do fluxo de dados.

O método utilizado pode ser visto na Figura 3, no qual separou-se 70% dos dados de cada base para treinamento e os demais 30% para teste. E, dos dados selecionados para treinamento, uma parte (10% da base inteira) foi utilizado para treinar o classificador no seu estágio inicial, e o restante passou pelo processo de seleção das estratégias de AL.

A medida de desempenho utilizada para obter os resultados dos classificadores foi a *f-measure*, pelo fato de as bases utilizadas serem desbalanceadas. Por fim, como forma de avaliar os resultados de uma maneira estatisticamente significante, foram aplicados o teste de Friedman (FRIEDMAN, 1937) e o pós-teste de Nemenyi (NEMENYI, 1963), ambos

considerando um nível de significância de 95%. O primeiro foi utilizado com o intuito de observar se os desempenhos das estratégias avaliadas apresentaram alguma diferença significativa; e o segundo para, caso houvesse diferença, identificar quais estratégias a apresentaram. As bibliotecas Python SciPy[7] e Orange[8] foram utilizadas para este fim.

Figura 3 – Técnica utilizada para divisão do conjunto de dados.



Fonte: próprio autor (2020).

4.5 CONFIGURAÇÃO DOS EXPERIMENTOS

Visando uma avaliação completa das estratégias de seleção de *Active Learning*, realizou-se experimentos em três cenários diferentes, além de ter sido feita uma análise de seus dois parâmetros principais: B e θ.

Os parâmetros s, que corresponde ao tamanho do passo de ajuste utilizado nas estratégias *Variable Uncertainty*, *Variable Entropy*, *Variable Randomized Uncertainty*, e *Variable Randomized Entropy*; e δ, que representa a variância da distribuição normal utilizada nas duas últimas, não foram avaliadas neste trabalho pelo fato de os autores que primeiro propuseram essas estratégias (ŽLIOBAITĖ et al., 2011) já terem estabelecido seus valores: respectivamente, 0.01 e 1.

Como citado anteriormente, para este trabalho não foi utilizada a técnica de validação cruzada *k-fold*, então, para todos os cenários, cada experimento foi executado apenas uma vez em cada base de dados. Isso se deve ao fato de que, com exceção das estratégias *Random Sampling*, *Variable Randomized Uncertainty* e *Variable Randomized Entropy*, as quais possuem algum mecanismo de aleatorização, cada estratégia de seleção sempre selecionará, em um mesmo cenário, as mesmas instâncias.

4.5.1 Avaliação dos parâmetro B e θ

O parâmetro B, que representa o tamanho do *budget* na estratégia *Random Sampling*, corresponde à porcentagem dos dados que tendem a ser selecionados pela estratégia. Desta forma, optou-se por variar este parâmetro em cinco valores: 0.1, 0.2, 0.3, 0.4 e 0.5; que correspondem, respectivamente, à seleção de, em torno de, 10%, 20%, 30%, 40% e 50% dos

[7] https://www.scipy.org/scipylib/index.html
[8] https://docs.biolab.si//3/data-mining-library/

dados. Logo, quanto maior o valor do parâmetro, mais dados são selecionados. Žliobaitė et al. (2011) realizaram a mesma avaliação no seu trabalho, justificando a escolha dos valores. Além disso, optou-se por não utilizar valoras maior que 0.5, os quais corresponderiam à seleção de mais da metade das bases de dados.

Já o parâmetro θ, que corresponde ao limiar de certeza na estratégias *Uncertainty*, *Variable Uncertainty* e *Variable Randomized Uncertainty* e ao limiar de entropia nas estratégias *Entropy*, *Variable Entropy* e *Variable Randomized Entropy*, foi avaliado com os valores de 0.5, 0.6, 0.7, 0.8 e 0.9. Para este parâmetro, a quantidade de instâncias que tendem a ser selecionadas também é diretamente proporcional ao seu valor. Também não foram utilizados valores maiores que 0.9, pois valores mais próximos a 1 corresponderiam à seleção de praticamente todas as instâncias.

As avaliações para os parâmetros foram realizadas considerando as bases de dados completas, sem a utilização de um subconjunto de validação. E os valores de cada parâmetro correspondentes aos melhores resultados nesta análise foram utilizados para avaliar os cenários posteriores.

4.5.2 Cenário I: Processo iterativo

O primeiro cenário avaliado nesta dissertação pode ser considerado o cenário "padrão", o qual reflete o processamento de um *data stream* em tempo real. Ele consiste na escolha iterativa das instâncias pela estratégia de seleção, de forma que, cada vez que uma instância é escolhida, ela é adicionada ao conjunto de treinamento e o modelo de aprendizagem é reajustado.

Neste cenário, após o treinamento inicial do classificador com o conjunto semente, o processo de *Active Learning* ocorre da seguinte forma (cujo processo se repete até que todas as instâncias disponíveis para seleção tenham sido avaliadas):

1. uma nova instância é avaliada pela estratégia de seleção utilizando o modelo no seu estado atual;

2. caso a instância não tenha sido escolhida, esta é descartada e parte-se para avaliação da instância seguinte; caso a instância tenha sido escolhida, esta é adicionada ao conjunto de treinamento, o qual é utilizado para reajustar o modelo, retreinando-o.

A atualização do modelo com cada nova instância selecionada foi realizada, nesta dissertação, utilizando o método *teach()* da biblioteca *modAL* (DANKA; HORVATH, 2018a), o qual reajusta o modelo com o conjunto de dados atualizado sem necessitar que ele seja retreinado do zero, diminuindo o custo computacional de retreinamento.

4.5.3 Cenário II: Seleção anterior à atualização

O segundo cenário consiste na seleção de todas as instâncias antes que o modelo seja atualizado. A diferença para o cenário anterior é que, quando um novo documento é escolhido, ele é adicionado a um conjunto intermediário, em vez de ser diretamente incluso no conjunto de treinamento do classificador. Assim, quando todas as instâncias forem analisadas pela estratégia de seleção é que ocorre a atualização do modelo com o conjunto de instâncias selecionadas. Portanto, o algoritmo só é retreinado uma vez, ao final do processo.

Neste caso, após o treinamento inicial do classificador com o conjunto semente, o processo de *Active Learning* ocorre da seguinte forma (cujo processo também se repete até que todas as instâncias disponíveis para seleção tenham sido avaliadas):

1. uma nova instância é avaliada pela estratégia de seleção utilizando o modelo no seu estado atual;

2. caso a instância não tenha sido escolhida, esta é descartada e parte-se para avaliação da instância seguinte; caso a instância tenha sido escolhida, esta é adicionada a um conjunto intermediário, o qual será utilizado para retreinar o modelo quando todos os documentos tiverem sido avaliados.

Para este cenário, é necessário observar que as estratégias de seleção utilizam o modelo em seu estado atual para avaliar novas instâncias: por exemplo, a estratégia *Uncertainty* utiliza o classificador para rotular o novo exemplo, calculando seu nível de incerteza, o qual é utilizado para a tomada de decisão. Sendo assim, os Cenários I e II apresentam comportamentos distintos e o classificador obtido ao final do processo tende a ser bastante diferente nos dois casos, embora o mesmo conjunto tenha sido avaliado, pois os documentos selecionados em cada cenário são diferentes.

Isso se deve ao fato de que, enquanto no Cenário I o modelo utilizado para avaliar as instâncias está sempre sofrendo atualizações, no Cenário II todos os exemplos são avaliados de acordo com o modelo no seu estado inicial, isto é, treinado com apenas o conjunto semente e podendo ser considerado um classificador fraco. Dessa forma, a mesma instância é avaliada no primeiro cenário tendo como base um modelo bastante diferente daquele utilizado para avaliá-la no segundo cenário.

Entretanto, este cenário tem como objetivo avaliar as estratégias em situações em que não é possível o retreinamento constante do modelo. Em situações nas quais o *data stream* é dividido em *batches*, por exemplo, todo um conjunto de dados é avaliado antes que haja a atualização do classificador.

4.5.4 Cenário III: Número fixo de instâncias

Este é outro cenário em que o retreinamento do modelo só ocorre uma vez. Neste caso, o processo não é iterativo, as instâncias são todas avaliadas e ordenadas pela medida utilizada por cada estratégia: aleatoriedade, incerteza, entropia ou ganho de informação, e os documentos que estiverem no topo da lista são selecionados e utilizados para atualização do modelo.

Ou seja, calcula-se a medida utilizada (entropia, por exemplo) para todas as instâncias, as quais são ordenadas de forma decrescente e seleciona-se os X primeiros elementos da lista. Para isso, é necessário que todos os dados do fluxo estejam armazenados e disponíveis para consulta ao mesmo tempo, diferentemente dos cenários anteriores, que avaliavam uma instância por vez. E duas porcentagens (valores de X) diferentes dos dados foram utilizadas: 33% e 50%, isto é, um terço e metade dos documentos disponíveis para seleção.

Por este cenário não utilizar um processo iterativo de avaliação dos documentos, as estratégias *Variable Uncertainty*, *Variable Randomized Uncertainty*, *Variable Entropy* e *Variable Randomized Entropy* não podem ser utilizadas nele, já que elas requerem uma escolha iterativa das instâncias para que o limiar θ possa ser alterado. Isso restringe a avaliação deste cenário a apenas quatro técnicas: *Random Sampling*, *Uncertainty*, *Information Gain* e *Entropy*.

Desta forma, pode-se avaliar as estratégias considerando que elas selecionaram a mesma quantidade de instâncias, o que pode ser útil em situações nas quais há um limite de instâncias que possam ser selecionadas. Porém, neste caso é necessário que o fluxo de dados esteja totalmente armazenado, já que a avaliação não ocorre *on the fly*.

5 RESULTADOS E DISCUSSÃO

As tabelas com os resultados alcançados nos cenários propostos e avaliados nesta dissertação encontram-se nos Apêndices A, B, C e D. Em cada uma delas, os maiores resultados por base de dados estão em negrito.

As subseções a seguir apresentam a discussão acerca dos resultados para cada cenário.

5.1 AVALIAÇÃO DOS PARÂMETROS B E θ

A seguir, é realizada a discussão do impacto do valor do parâmetro B para estratégia *Random Sampling*. Já para o parâmetro θ, posteriormente são apresentadas as discussões para cada uma das seis estratégias nas quais ele está presente: *Uncertainty*, *Variable Uncertainty*, *Variable Randomized Uncertainty*, *Entropy*, *Variable Entropy* e *Variable Randomized Entropy*.

5.1.1 *Random Sampling*

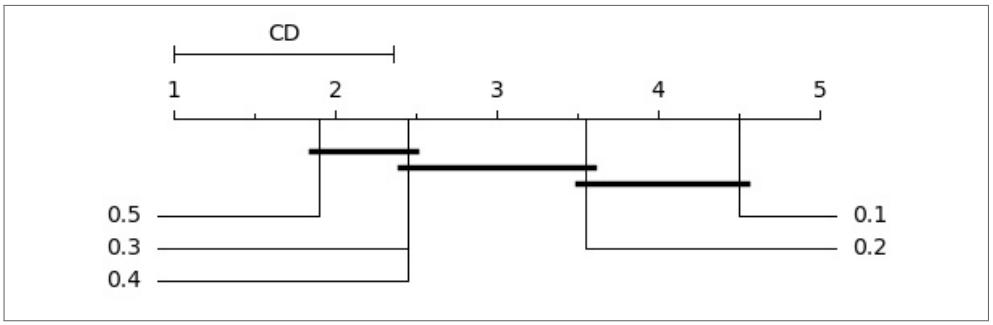
As Tabelas 3 e 4 apresentam as *f-measures* alcançadas pela estratégia *Random Sampling* em cada uma das bases, considerando os cinco valores de B avaliados e utilizando, respectivamente, os classificadores *Multinomial Naïve Bayes* (MNB) e Regressão Logística (RL).

Pelos resultados apresentados, pode-se notar que a estratégia *Random Sampling* utilizando $B = 0.5$ obteve os melhores desempenhos para os dois classificadores. Em ambos os casos, a configuração com o maior valor para este parâmetro atingiu a maior média, o menor ranking médio e a *f-measure* mais alta na maioria das bases. Isso pode ser explicado pelo fato de que, quanto maior o valor de B, maior a quantidade de instâncias selecionadas pela *Random Sampling*, o que impactou diretamente o desempenho dos classificadores. Para esta estratégia, o aumento na quantidade de documentos selecionados tende a ser benéfico, já que ela os seleciona de forma aleatória.

O teste estatístico de Friedman demonstrou que, para ambos os classificadores, há uma diferença significativa entre os resultados: o teste retornou um *p-value* de 3.834×10^{-7} para o MNB e de 1.328×10^{-8} para o RL. Porém, como pode-se observar pelos diagramas de *Critical Difference* (CD) apresentados nas Figuras 4 e 5, os resultados alcançados pelos valores de $B = 0.5$, 0.4 e 0.3 são estatisticamente semelhantes (no diagrama, os valores conectados por uma barra não apresentam diferença significativa).

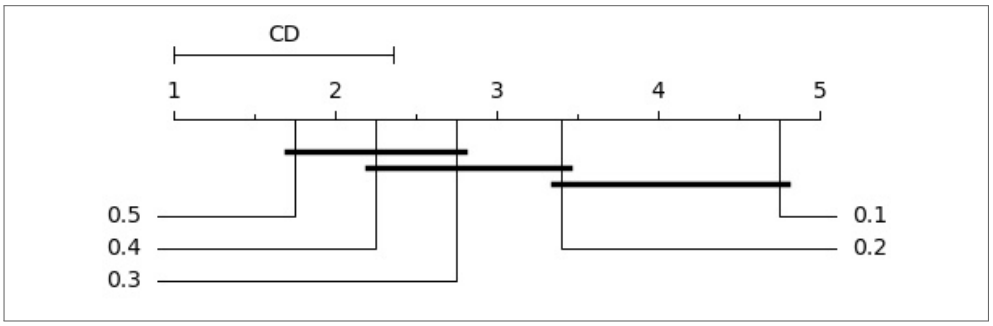
Como os resultados para ambos os classificadores foram bastante semelhantes (como pode-se notar nas Figuras 4 e 5) e, pelo fato de o SVM ser um classificador cujo processo de treinamento é demorado, a análise do parâmetro B não foi realizada com este algoritmo, pois despenderia muito tempo, principalmente nas bases de dados maiores.

Figura 4 – Resultado do pós-teste Nemenyi para o parâmetro B com o classificador MNB.



Fonte: próprio autor (2020).

Figura 5 – Resultado do pós-teste Nemenyi para o parâmetro B com o classificador RL.

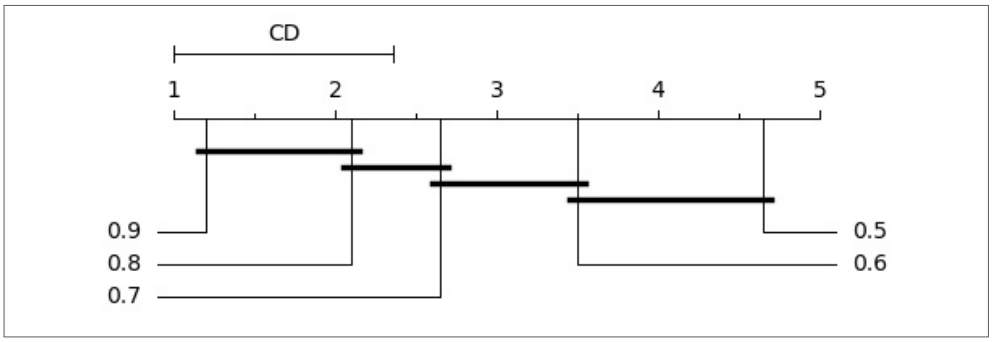


Fonte: próprio autor (2020).

5.1.2 *Uncertainty*

A Tabela 5 traz os resultados alcançados pela estratégia *Uncertainty* por base de dados ao variar o valor do parâmetro θ, considerando o classificador *Multinomial Naïve Bayes*; enquanto a Tabela 6 os traz considerando o classificador de Regressão Logística. E as Figuras 6 e 7 apresentam os respectivos diagramas CD obtidos após o pós-teste Nemenyi.

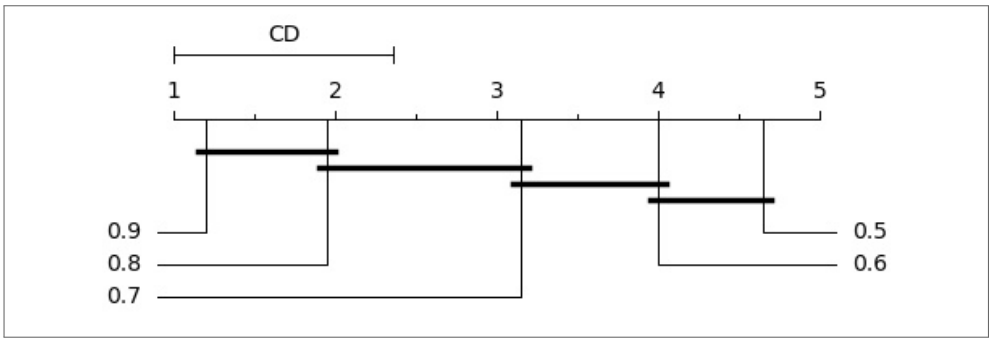
Figura 6 – Resultado do pós-teste Nemenyi para o parâmetro θ da estratégia *Uncertainty* com o classificador MNB.



Fonte: próprio autor (2020).

Para a estratégia *Uncertainty*, pode-se observar que, assim como na avaliação do parâmetro B, o maior valor do parâmetro θ, que também corresponde à maior quantidade

Figura 7 – Resultado do pós-teste Nemenyi para o parâmetro θ da estratégia *Uncertainty* com o classificador RL.



Fonte: próprio autor (2020).

de instâncias selecionadas, obteve os resultados mais expressivos na grande maioria das bases de dados. Desta vez, o valor de $\theta = 0.9$ foi alcançou a *f-measure* mais alta em 80% das bases, para ambos classificadores, além de ter atingido também a maior média geral nos dois casos. Entretanto, os diagramas CD mostram que os resultados obtidos por $\theta = 0.9$ não chegam a ser estatisticamente superiores aos alcançados pela utilização de $\theta = 0.8$.

Mesmo assim, torna-se claro que, quanto maior o valor de θ, mais instâncias são selecionadas pela estratégia *Uncertainty*, o que tende a melhorar o desempenho do modelo de aprendizagem, fazendo-o reagir com mais eficiência a *drifts*. Outro ponto interessante de se notar se encontra no fraco desempenho desta técnica de seleção quando utilizando $\theta = 0.5$; isto se deve ao fato de que, com esse limiar, um número extremamente baixo de documentos foi selecionado: em 16 das 20 bases para o MNB e em 15 delas para o RL, a estratégia sequer selecionou alguma instância, fazendo com que o modelo permanecesse no seu primeiro estágio, isto é, treinado apenas com os 10% iniciais do *data stream*.

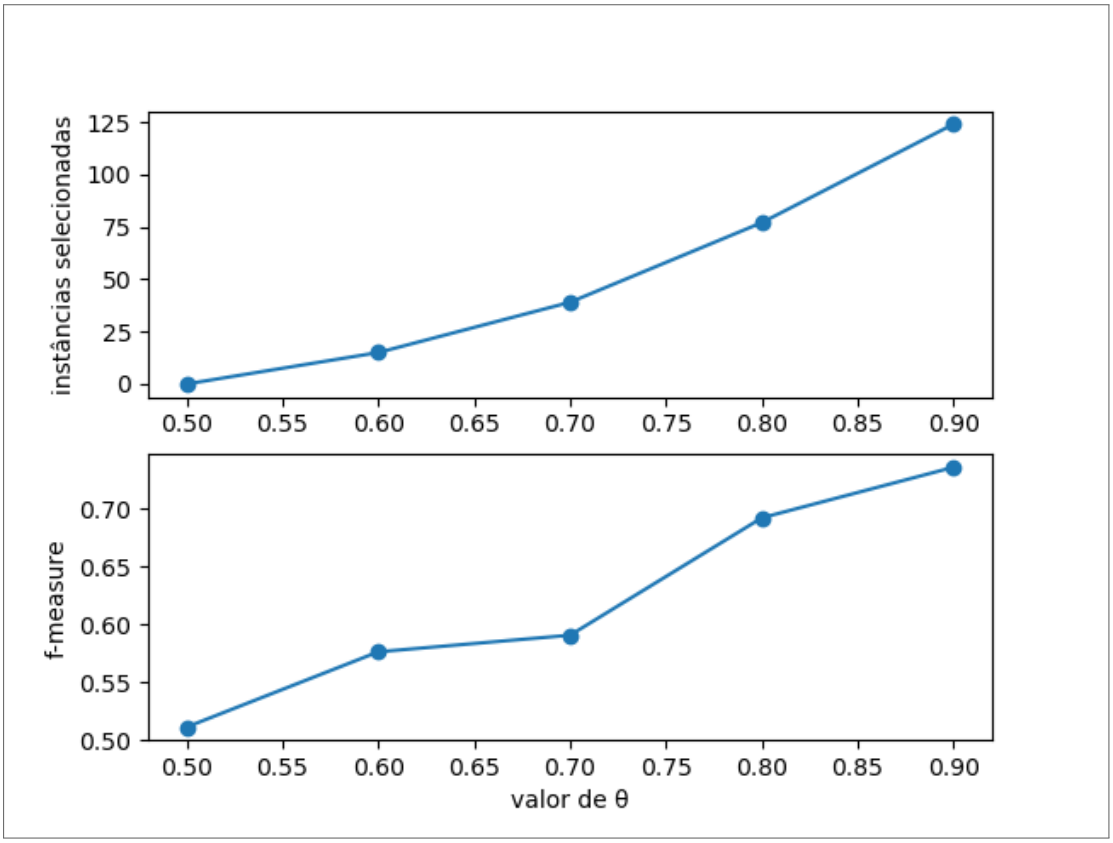
A Figura 8 apresenta gráficos que relacionam os valores avaliados de θ com a quantidade de instâncias selecionadas pela estratégia *Uncertainty* na base de dados *Haddad_twitter*, bem como com a *f-measure* alcançada por essa estratégia no mesmo cenário. Os gráficos mostram uma correlação entre o aumento do valor de θ, o aumento da quantidade de instâncias escolhidas e a melhora da *f-measure* do classificador, pelo menos no contexto analisado.

Dessa forma, pode-se afirmar que o impacto da escolha do valor de θ na estratégia *Uncertainty* é alto, e que, quanto mais documentos são selecionados por essa estratégia, melhor tende a ser o desempenho do classificador, em termos de *f-measure*.

5.1.3 *Variable Uncertainty*

Já para a estratégia *Variable Uncertainty*, não houve diferença estatística entre os resultados alcançados utilizando diferentes valores do parâmetro θ: o teste de Friedman

Figura 8 – Relação entre os valores de θ e a quantidade de instâncias selecionadas (acima) e a *f-measure* (abaixo) da estratégia *Uncertainty* na base de dados *Haddad_twitter*.



Fonte: próprio autor (2020).

retornou *p-values* de 0.4282 para o MNB e de 0.7598 para o classificador RL.

Sendo assim, pode-se dizer que, para as bases avaliadas, a mudança do valor de θ não apresentou muita significância para essa estratégia. Isso pode decorrer do mecanismo de variação do limiar presente nela, fazendo com que a escolha inicial do valor do parâmetro não tenha muito impacto nos resultados. As Tabelas 7 e 8 apresentam os resultados alcançados pelos experimentos com a *Variable Uncertainty*.

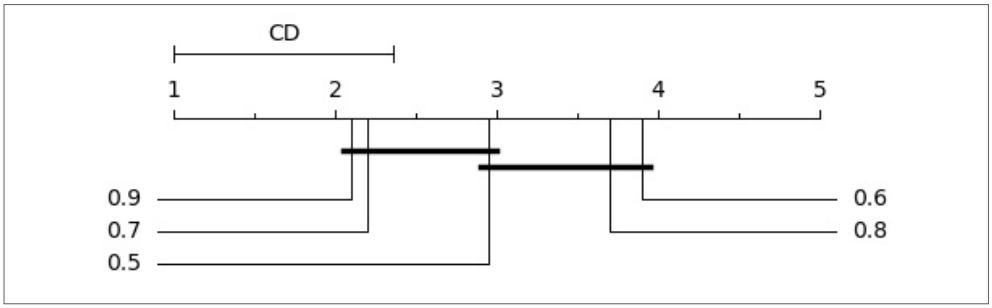
Além disso, é interessante ressaltar que esta estratégia não sofre do mesmo problema da *Uncertainty* ao se utilizar o valor $\theta = 0.5$, já que o processo de variação do limiar o faz aumentar quando um número baixo de instâncias está sendo selecionado.

5.1.4 *Variable Randomized Uncertainty*

Para a *Variable Randomized Uncertainty*, os resultados apresentados nas Tabelas 9 e 10 também mostram que o valor inicial do limiar θ não tem tanto impacto no desempenho do classificador. O teste de Friedman demonstrou que não há diferença estatística nos resultados considerando o classificador de Regressão Logística (*p-value* de 0.9884) e que há uma pequena diferença considerando o *Multinomial Naïve Bayes* (*p-value* de 0.0002).

Porém, o pós-teste de Nemenyi apontou que o maior valor de θ (0.9) obteve resultados estatisticamente semelhantes ao menor valor (0.5), como pode ser visto no diagrama CD da Figura 9. Portanto, é possível afirmar que a relação entre o aumento do valor do parâmetro e a melhora no desempenho do classificador, observado para a estratégia *Uncertainty*, não se repetiu para a *Variable Randomized Uncertainty*, nem mesmo utilizando o MNB. Isso pode decorrer do mecanismo de aleatorização presente nesta estratégia, fazendo com que ela não seja tão sensível a mudanças no valor de θ.

Figura 9 – Resultado do pós-teste Nemenyi para o parâmetro θ da estratégia *Variable Randomized Uncertainty* com o classificador MNB.



Fonte: próprio autor (2020).

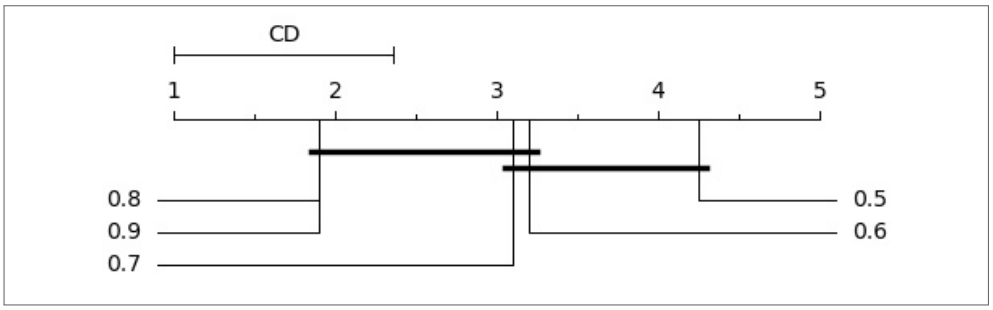
5.1.5 *Entropy*

As *f-measures* obtidas pela estratégia *Entropy* podem ser encontradas nas Tabelas 11, utilizando o *Multinomial Naïve Bayes*, e 12, utilizando o algoritmo de Regressão Logística. Nelas, é possível notar que os valores de θ de 0.8 e 0.9 se sobressaíram aos demais, de maneira semelhante ao que foi observado para a estratégia *Uncertainty* ao se utilizar o classificador MNB. Porém, para o classificador RL, a diferença entre os desempenhos dos maiores valores de θ e dos menores não foi tão grande: por exemplo, o valor de 0.9, apesar de ter obtido a maior média, não alcançou o maior número de vitórias, nem o menor ranking médio.

O teste de Friedman confirma o apontado no parágrafo anterior: enquanto ele retornou um *p-value* de 4.0755×10^{-7} para os experimentos com o MNB, mostrando uma grande certeza na diferença estatística, para os experimentos com o RL, o *p-value* retornado foi de 0.0274, mostrando apenas uma pequena certeza. E, apesar de haver diferença estatística nos dois casos, de acordo com os diagramas CD das Figuras 10 e 11, os valores de θ de 0.6, 0.7, 0.8 e 0.9 apresentaram resultados estatisticamente semelhantes para ambos os classificadores.

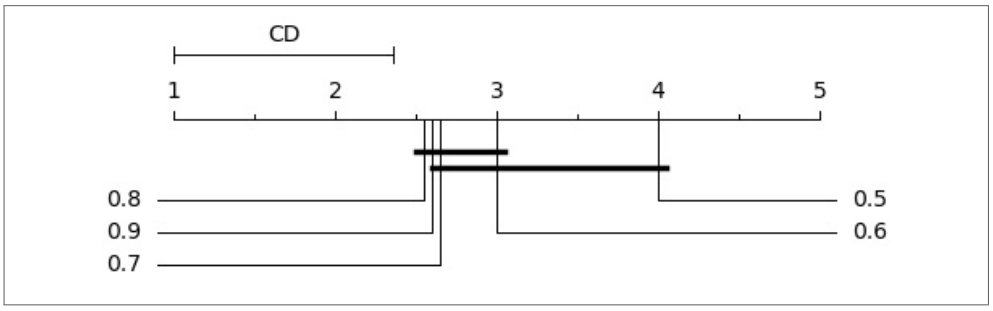
Sendo assim, pode-se indicar a existência de um impacto na escolha do valor do limiar de entropia para a estratégia *Entropy*, já que ao se utilizar $\theta = 0.8$ e $\theta = 0.9$ obteve-se resultados estatisticamente superiores à utilização de $\theta = 0.5$. Entretanto, este impacto não é tão grande quanto o observado para a *Uncertainty*.

Figura 10 – Resultado do pós-teste Nemenyi para o parâmetro θ da estratégia *Entropy* com o classificador MNB.



Fonte: próprio autor (2020).

Figura 11 – Resultado do pós-teste Nemenyi para o parâmetro θ da estratégia *Entropy* com o classificador RL.



Fonte: próprio autor (2020).

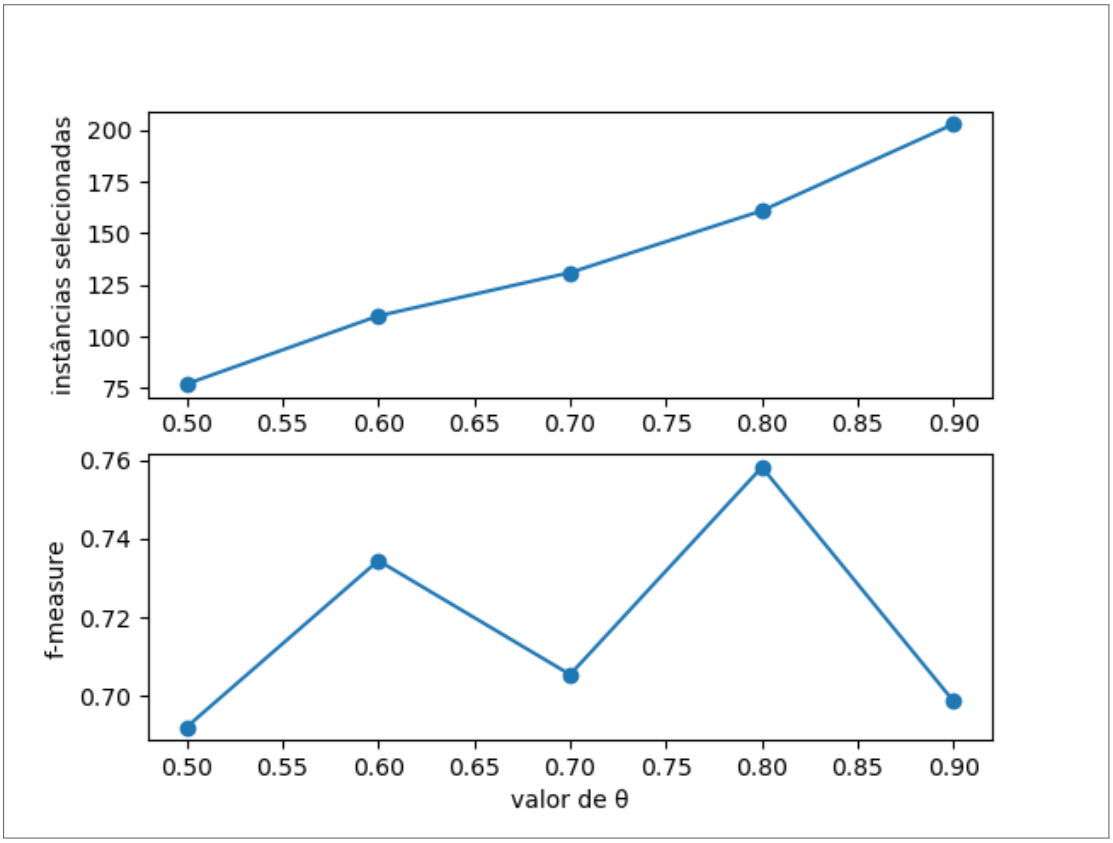
Enquanto que, para o método baseado em incerteza, o aumento do número de instâncias selecionadas quase sempre indicou uma melhora no desempenho, para aquele baseado em entropia esta relação não se mostrou direta, como pode ser observado nos gráficos da Figura 12 ao compará-los com os gráficos da Figura 8, a qual mostrou as mesmas relações, na mesma base de dados, porém utilizando a *Uncertainty*. Isso pode indicar que, em alguns casos, a seleção de uma quantidade maior de instâncias pela *Entropy* se torna maléfica para o modelo, possivelmente pela seleção de instâncias que podem ser consideras ruído.

5.1.6 *Variable Entropy*

Da mesma forma que para a estratégia *Variable Uncertainty*, os resultados alcançados pelos diferentes valores do parâmetro θ também não apresentaram significância para a *Variable Entropy*. O teste de Friedman mostrou isto ao retornar *p-values* de 0.2713 e 0.6707 para, respectivamente, os algoritmos *Multinomial Naïve Bayes* e de Regressão Logística, ou seja, afirmando que não houve diferença significativa entre os resultados para nenhum dos classificadores.

As Tabelas 13 e 14 apresentam os resultados obtidos, mostrando que os valores do parâmetro θ tiveram desempenhos bem semelhantes, não sendo possível apontar qual a

Figura 12 – Relação entre os valores de θ e a quantidade de instâncias selecionadas (acima) e a *f-measure* (abaixo) da estratégia *Entropy* na base de dados *Haddad_twitter*.



Fonte: próprio autor (2020).

melhor escolha levando em consideração os dois classificadores analisados.

A explicação para este fato tende a ser a mesma que para o método baseado em incerteza variável: o mecanismo de variação do limiar faz com que a escolha inicial do seu valor não tenha tanta importância, independente da medida escolhida.

5.1.7 *Variable Randomized Entropy*

Por fim, a escolha do parâmetro θ também se mostrou sem importância para a estratégia *Variable Randomized Entropy* nos experimentos com ambos os algoritmos de classificação. Os *p-values* de 0.1423 e 0.6482 provaram que não há diferença estatística nos resultados, considerando, respectivamente, o MNB e o RL.

Esta conclusão é semelhante àquela encontrada ao se analisar a estratégia *Variable Randomized Uncertainty*, porém, naquele caso, descobriu-se um ligeiro impacto causado pela escolha do valor de θ para o algoritmo *Multinomial Naïve Bayes*. Entretanto, cabe ressaltar que este pequeno impacto pode ser fruto do fator de aleatorização presente em ambas as técnicas de seleção.

As Tabelas 15 e 16 trazem as *f-measures* da avaliação do parâmetro θ para a *Variable*

Randomized Uncertainty.

5.2 CENÁRIO I: PROCESSO ITERATIVO

Pela análise realizada e detalhada na Seção 5.1, escolheu-se os seguintes valores para os parâmetros: $B = 0.5$ e $\theta = 0.9$, os quais foram utilizados nos três cenários cujos resultados são apresentados a seguir. Embora o parâmetro θ tenha se mostrado importante apenas para a estratégia baseada em incerteza fixa, optou-se pela escolha do mesmo valor para todas as outras, como forma de equipará-las, e, já que o valor de 0.9 se mostrou o melhor para a *Uncertainty* e sem representar prejuízo para as outras, ele foi o selecionado.

As Tabelas 17 e 18 apresentam os resultados alcançados por cada uma das estratégias: *Random Sampling* (RS), *Uncertainty* (UN), *Variable Uncertainty* (VU), *Variable Randomized Uncertainty* (VRU), *Information Gain* (IG), *Entropy* (EN), *Variable Entropy* (VE) e *Variable Randomized Entropy* (VRE) para os algoritmos *Multinomial Naïve Bayes* e de Regressão Logística, respectivamente.

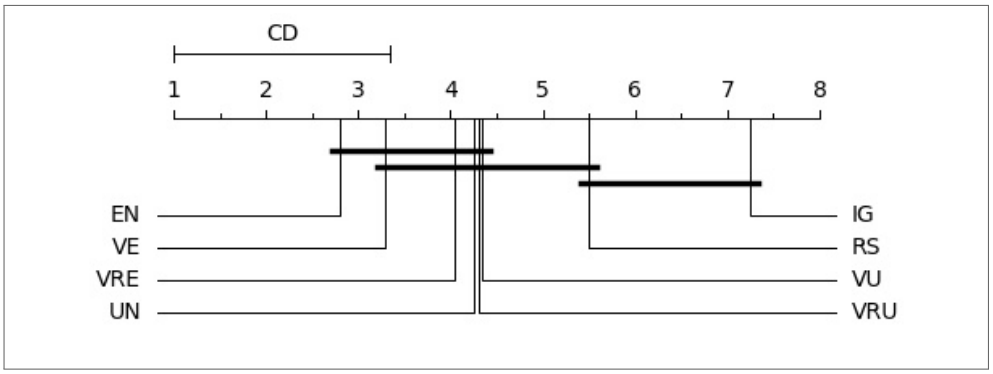
Primeiramente, é preciso apontar que os modelos obtiveram os melhores resultados nas bases sobre as Eleições Presidenciais 2018 com dados do Facebook, as quais foram construídas para esta dissertação. As *f-measures* superiores a 85%, alcançadas por algumas configurações, podem ser explicadas pelo fato de que estas bases também apresentaram um coeficiente Kappa de Cohen elevado (em torno de 95%), como pode ser visto no Quadro 2, mostrando que em uma base de dados na qual não houve dificuldades para humanos anotarem, os classificadores também tendam a alcançar bons resultados.

Comparando as estratégias de seleção, pode-se observar que a técnica *Entropy* obteve os melhores resultados numéricos para ambos os classificadores, atingindo a maior *f-measure* média e o maior número de vitórias: 17 no total. Este achado é similar ao encontrado por Yang e Loog (2018), os quais realizaram uma comparação de estratégias de AL em bases de dados estáticas e não-textuais para regressão logística, e mostraram que a utilização de entropia como medida de incerteza pode produzir o melhor resultado em um grande número de *data sets*.

Já de forma diferente aos trabalhos publicados anteriormente com base nesta pesquisa (VITÓRIO; SOUZA; OLIVEIRA, 2019a; VITÓRIO; SOUZA; OLIVEIRA, 2019b), nos quais a estratégia *Uncertainty* obteve um desempenho muito baixo para o classificador MNB utilizando o valor de $\theta = 0.7$, nesta dissertação, com o valor de $\theta = 0.9$, ela alcançou resultados bem mais expressivos nas mesmas bases de dados, mostrando novamente a importância da escolha do valor do limiar de incerteza para esta estratégia.

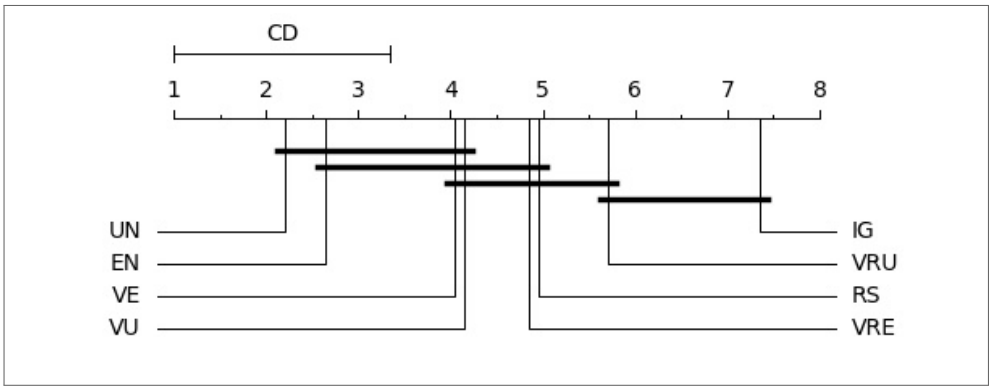
Essa melhora no desempenho pode ser observada mais claramente pelos diagramas CD das Figuras 13 e 14, nos quais a *Uncertainty* obteve o quarto melhor ranking para o classificador MNB (nos trabalhos anteriores, ela havia obtido os piores rankings para este algoritmo), sendo estatisticamente semelhante à estratégia *Entropy*, e o melhor para o classificador RL.

Figura 13 – Resultado do pós-teste Nemenyi para o Cenário I com o classificador MNB.



Fonte: próprio autor (2020).

Figura 14 – Resultado do pós-teste Nemenyi para o Cenário I com o classificador RL.



Fonte: próprio autor (2020).

Outra diferença nos resultados deste cenário para aqueles publicados em Vitório, Souza e Oliveira (2019a) e Vitório, Souza e Oliveira (2019b) é que, considerando $\theta = 0.9$, as variações da *Uncertainty* propostas por Žliobaitė et al. (2011) não apesentaram uma grande melhora no desempenho em comparação com a estratégia de original, ao contrário do que pôde ser observado com $\theta = 0.7$. Isso pode ser explicado pelo fato de que o parâmetro θ, embora seja essencial para a *Uncertainty*, não apresenta um impacto grande nas estratégias *Variable Uncertainty* e *Variabel Randomized Uncertainty*; portanto, a melhora observada na performance da primeira com o aumento do limiar não refletiu nas *f-measures* das outras duas.

Embora a estratégia *Information Gain* tenha alcançado os melhores resultados no estudo que a propôs (ZIMMERMANN; NTOUTSI; SPILIOPOULOU, 2015), ela se mostrou a pior no cenário desta dissertação, sendo inferior estatisticamente à maioria das outras técnicas. A explicação para essa diferença pode estar no tamanho das bases de dados utilizadas: enquanto Zimmermann, Ntoutsi e Spiliopoulou (2015) realizaram experimentos com um corpus contendo 250000 *tweets*, a maior base usada neste trabalho continha 10000 *tweets*.

Além disso, a *Information Gain* também apresenta um ponto negativo pelo fato de ser a mais complexa dentre as estratégias avaliadas, já que necessita manter um vocabulário armazenado, elevando seu custo computacional.

Outra análise que pode ser feita é a comparação entre as técnicas de *Active Learning* com uma *baseline* dos classificadores sem utilizar AL, isto é, tendo sido treinados com o conjunto inteiro de documentos. A Tabela 19 traz a comparação entre a *f-measure* obtida pela melhor estratégia de seleção para cada base e a *f-measure* alcançada pelo classificador sem AL, ou seja, com um processo de aprendizagem totalmente supervisionado.

Assim, é possível verificar que, na maioria das bases de dados, o modelo treinado utilizando *Active Learning* atingiu um resultado superior ao classificador treinado com toda a base; isto mostra que as estratégias de AL, além de diminuírem a quantidade de dados a serem anotados, também conseguem superar em desempenho o modelo tradicional de Aprendizagem de Máquina Supervisionada, pelo menos em aplicações de Mineração de Opinião com *data streams*. Ao escolherem apenas as instâncias mais relevantes para o modelo, as estratégias de seleção tendem a melhorar a *f-measure* do classificador, além de a sua natureza iterativa fazer com que ele reaja bem à ocorrência de *drifts*.

O teste de Wilcoxon (WILCOXON, 1945) foi utilizado para comparar estatisticamente os resultados apresentados na Tabela 19. Considerando um nível de significância de 95%, pôde-se observar que não houve uma diferença significativa entre elas: *p-value* de 0.3702 para o classificador *Multinomial Naïve Bayes* e de 0.0761 para o classificador de Regressão Logística. Porém, esse desempenho superior numericamente e equiparável estatisticamente da técnica de *Active Learning* pode indicar uma capacidade de eliminar ruídos por parte das estratégias de seleção.

Já as Tabelas 20 e 21 apresentam a quantidade de instâncias selecionadas por cada estratégia, mostrando também a quantidade disponível de documentos para seleção em cada base de dados.

As quantidades correspondentes ao melhor desempenho (em termos de *f-measure*) por base de dados estão em negrito, apontando que a seleção de uma maior quantidade de instâncias não implica necessariamente na melhoria do desempenho do modelo de aprendizagem. Contudo, as estratégias que, no geral, obtiveram os melhores resultados neste primeiro cenário também foram as que, em média, selecionaram mais instâncias para atualização do modelo: *Entropy* e *Uncertainty*. Esta última, inclusive, apresentou uma melhor performance com o classificador RL, para o qual também selecionou um número maior de documentos, do que com o classificador MNB.

Ademais, a estratégia *Information Gain*, considerada a pior em relação à *f-measure*, também foi a responsável por escolher a menor quantidade média de instâncias. Sendo assim, pode-se estabelecer uma relação entre o desempenho do modelo e a quantidade de vezes que ele é atualizado com novos dados, porém esta não é uma regra.

E, conforme apresentado anteriormente, a estratégia *Entropy*, responsável pelo melhor

desempenho em termos de *f-measure*, escolheu um número muito grande de instâncias: em alguns casos este número foi bastante próximo da quantidade total de dados disponíveis para seleção, enquanto os outros métodos selecionaram cerca da metade desta quantidade. Este fato constitui uma grande desvantagem para a *Entropy* em relação às outras estratégias, já que manter o conjunto de treinamento o menor possível é uma característica importante da abordagem de *Active Learning* (ZHU et al., 2007).

Por conta disso, a escolha da entropia fixa como medida a ser utilizada para seleção de documentos em *Active Learning* pode não ser a melhor opção em situações nas quais, por exemplo, os recursos disponíveis não permitem a anotação de um grande volume de dados. Nesses casos, uma opção mais viável é a estratégia *Variable Entropy*, a qual sempre alcançou resultados estatisticamente semelhantes à *Entropy*, mas mantendo o conjunto de treinamento com um tamanho bem menor.

Por fim, o classificador *Support Vector Machine* não foi utilizado neste cenário pelo fato de seu custo computacional de treinamento ser muito elevado. Como o cenário apesentado é iterativo, requerendo uma atualização do modelo de aprendizagem a cada nova instância selecionada, o tempo despendido pelo SVM para executar essa tarefa faz com que sua utilização não tenha sido viável neste caso, principalmente considerando as bases de dados com mais de 2000 documentos.

Como medida de comparação, calculou-se o tempo de execução dos três classificadores no cenário de avaliação com os parâmetros $B = 0.1$ e $\theta = 0.9$ para a base *Sentiment140_1000_1*, concluindo que, numa base relativamente pequena contendo 1000 *tweets*, o SVM necessitou de um tempo (17678 segundos) cem vezes maior que os algoritmos MNB (187 segundos) e de Regressão Logística (167 segundos). Isso faz com que o SVM não seja adequado para Mineração de Opinião de forma iterativa com fluxos de dados, nos quais os dados tendem a chegar com uma velocidade muito alta, porque representaria um atraso na classificação dos novos documentos.

5.3 CENÁRIO II: SELEÇÃO ANTERIOR À ATUALIZAÇÃO

Para o segundo cenário, no qual todas as instâncias foram selecionadas antes que houvesse qualquer atualização do modelo, os resultados são apresentados nas Tabelas 22, 23 e 24. Nesta última, são mostrados os resultados para o classificador SVM nas 14 bases de dados com as quais foi viável executar os experimentos.

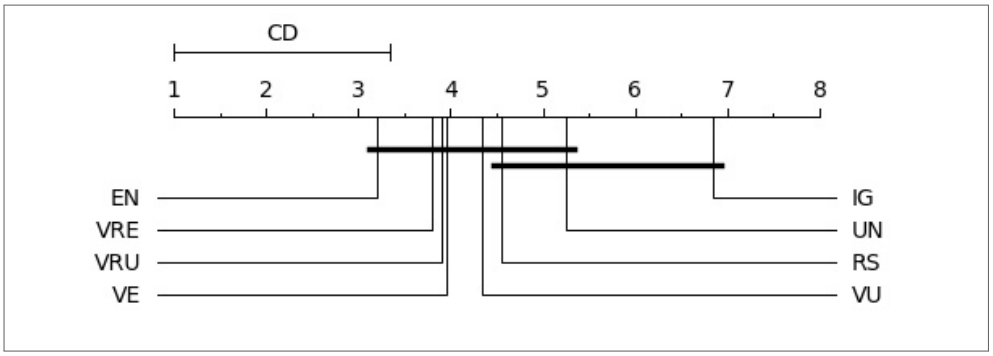
O primeiro ponto que se pode analisar é que, comparando o desempenho dos dois cenários apresentados até o momento, para o classificador *Multinomial Naïve Bayes* o Cenário I apresentou resultados estatisticamente superiores ao Cenário II, de acordo com o teste de Wilcoxon com um nível de significância de 95%. Utilizando este teste para comparar as melhores *f-measures* alcançadas em cada cenário, obteve-se um *p-value* de 0.0125, indicando uma diferença estatística.

Já ao analisar os resultados do classificador de Regressão Logística, o *p-value* retornado pelo teste de Wilcoxon foi de 0.7945, mostrando que, para esse algoritmo, os dois cenários são estatisticamente semelhantes. Sendo assim, pode-se concluir que a escolha do processo de *Active Learning*, se com uma atualização iterativa do modelo ou atualizando-o apenas ao final da seleção, pode depender do classificador utilizado, além do cenário.

Com base nas tabelas, observa-se que, de modo geral, o desempenho das estratégias nos dois cenários foi semelhante. Porém, a *Uncertainty* apresentou uma queda no seu desempenho no Cenário II com o MNB em comparação com o Cenário I: de acordo com a Tabela 22, ela alcançou resultados muito baixos em algumas bases, fazendo com que ela obtivesse a segunda pior *f-measure* média.

Essa queda no desempenho da *Uncertainty* é melhor observada no diagrama CD da Figura 15 e pode indicar que a falta de atualização constante do modelo, que faz com que o classificador permaneça fraco e sem capacidade de reagir de maneira eficaz a *drifts* durante toda a etapa de seleção, torna a medida de incerteza fixa, a qual é sempre avaliada utilizando o modelo atual, não muito recomendada para este cenário utilizando o classificador MNB, possivelmente pela natureza probabilística deste algoritmo.

Figura 15 – Resultado do pós-teste Nemenyi para o Cenário II com o classificador MNB.



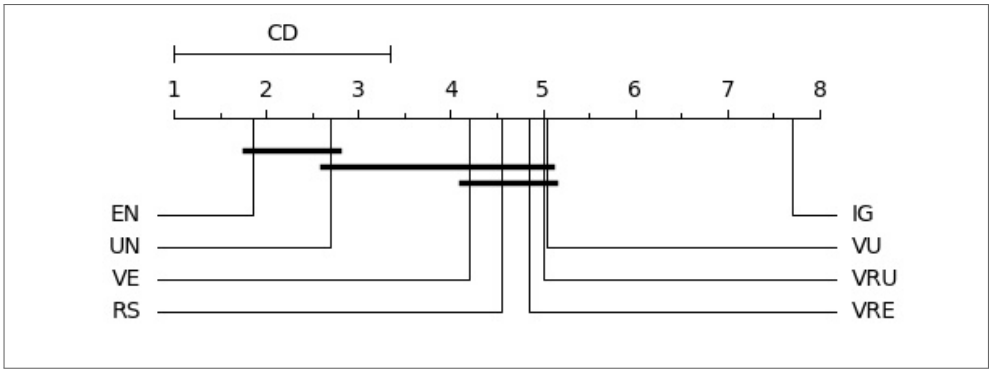
Fonte: próprio autor (2020).

Contudo, também pode-se notar na Figura 15 que, para este segundo cenário, as estratégias apresentaram performances bem semelhantes entre si, estatisticamente falando, utilizando o classificador *Multinomial Naïve Bayes*: a melhor estratégia (*Entropy*) se mostrou estatisticamente superior apenas à pior (*Information Gain*). Este resultado difere do visualizado no Cenário I, no qual a diferença entre as *f-measures* das estratégias foi mais significativa, como é possível perceber pelos *p-values* do teste de Friedman: 1.9942×10^{-7} para o primeiro cenário e 0.0001 para o segundo.

Enquanto isso, com relação ao classificador de Regressão Logística, a estratégia *Entropy* obteve resultados ainda melhores que para o cenário anterior: alcançando a melhor *f-measure* em 11 das 20 bases de dados e sendo estatisticamente superior a seis das sete outras estratégias de seleção, conforme apresentado pelo diagrama CD da Figura 16. Esses

resultados reiteram o achado descrito na seção anterior: de que, em termos de *f-measure*, a estratégia *Entropy* se sobressai às demais nos casos avaliados neste estudo.

Figura 16 – Resultado do pós-teste Nemenyi para o Cenário II com o classificador RL.



Fonte: próprio autor (2020).

A *Uncertainty* também manteve, utilizando o algoritmo RL, a boa performance apesentada no Cenário I, mostrando que as diferenças presentes na natureza de cada tipo de classificador podem afetar o uso de estratégias de *Active Learning*, já que a queda de desempenho observada ao se utilizar o MNB não se repetiu.

E analisando, com o teste de Wilcoxon, os melhores resultados alcançados pelos dois classificadores em cada base de dados, pode-se ver que para o Cenário I não houve diferença significativa (*p-value* de 0.1353), enquanto que para o Cenário II houve (*p-value* de 0.0206). Ou seja, é possível apontar que, pelo menos para o cenário em que a atualização do modelo só ocorre ao final do processo de seleção, o classificador de Regressão Logística é mais indicado, em termos de *f-measure*, que o classificador *Multinomial Naïve Bayes*.

Já ao se considerar o uso do classificador SVM, primeiro é preciso apontar que não foi viável realizar experimentos com ele em todas as 20 bases de dados. Nas bases com mais de 2000 instâncias, o custo computacional de treinamento desse algoritmo se tornou um grande empecilho: para a base *Sentiment140_2500_1*, por exemplo, o experimento de avaliação das oito estratégias de seleção durou em torno de 6 dias, ao passo que, para o MNB e o algoritmo de Regressão Logística, o tempo necessário foi de menos de 30 minutos para cada um.

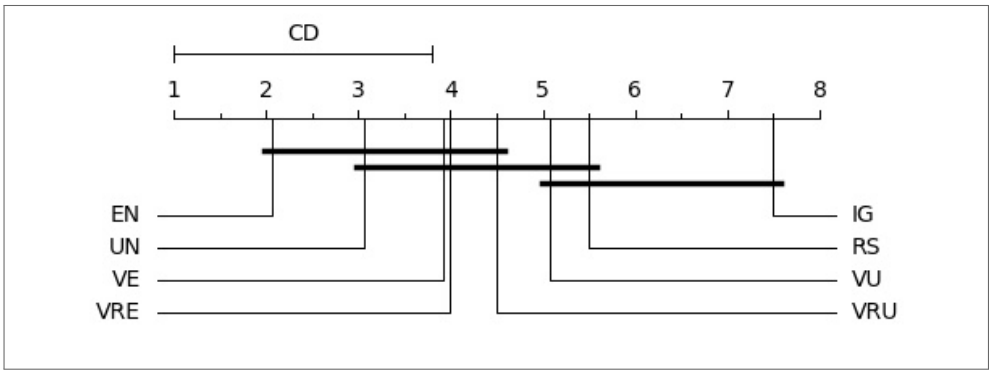
Este alto custo computacional de treinamento do SVM já o tornava não recomendável para processos iterativos de *Active Learning*. Agora, após a percepção de que o custo se mantém elevado em situações nas quais ocorre apenas um retreinamento do modelo e considerando que as técnicas de AL tornam o conjunto de treinamento menor, pode-se afirmar que o SVM não é indicado para Mineração de Opinião com *data streams*, principalmente para bases de dados grandes e ao compará-lo com outros classificadores.

Além disso, utilizando o teste de Wilcoxon, percebe-se que o SVM não se sobressai estatisticamente nem ao MNB (*p-value* de 0.1240), nem ao RL (*p-value* de 0.4630); para este último, inclusive, ocorre o contrário: ele alcançou *f-measures* superiores ao SVM na

maioria das bases de dados. Ou seja, o elevado custo computacional ao se utilizar o SVM não é compensado por uma melhora no desempenho da classificação.

Portanto, nesta pesquisa são reportados experimentos com o SVM em 14 bases de dados, com os quais pode-se perceber que a estratégia *Entropy* também se sobressaiu, numericamente falando, utilizando este classificador, enquanto que a *Uncertainty* alcançou o segundo melhor desempenho. Já considerando a análise estatística dos testes de Friedman e Nemenyi, nota-se que os resultados apresentaram uma diferença significativa, como pode-se observar no diagrama CD da Figura 17.

Figura 17 – Resultado do pós-teste Nemenyi para o Cenário II com o classificador SVM.



Fonte: próprio autor (2020).

Outro ponto confirmado pelos experimentos deste cenário é que a estratégia *Information Gain* apresenta, para as configurações avaliadas nesta dissertação, um desempenho muito aquém do apontado no estudo de Zimmermann, Ntoutsi e Spiliopoulou (2015).

E, como última análise realizada no Cenário II, as Tabelas 25, 26 e 27 trazem, respectivamente, a quantidade de instâncias selecionadas neste cenário por cada estratégia utilizando os classificadores *Multinomial Naïve Bayes*, de Regressão Logística e SVM. Nelas, as quantidades de instâncias selecionadas que correspondem às maiores *f-measures* por *data set* estão em negrito.

Novamente, assim como para o Cenário I, as duas estratégias que atingiram as maiores *f-measures* (*Entropy* e *Uncertainty*) também foram as responsáveis por selecionar, em média, o maior número de documentos; enquanto que a *Information Gain* se manteve como a pior técnica de seleção em termos de desempenho e como a que seleciona o menor número de instâncias. O que confirma que, na maioria dos casos, há uma relação direta entre a quantidade de instâncias adicionadas ao modelo e o desempenho da classificação.

Comparando-se o número de documentos selecionados pelas estratégias nos Cenários I e II, observa-se que, para o MNB, houve uma diminuição da quantidade média de instâncias no segundo cenário, enquanto que, para o RL ocorreu um aumento deste número; essa diferença pode ser observada principalmente para a *Uncertainty*, o que pode explicar a queda no desempenho desta estratégia considerando o Cenário II ao se utilizar o classificador *Multinomial Naïve Bayes*.

E considerando o algoritmo SVM, percebe-se que, na maioria das bases, tanto a *Entropy* quanto a *Uncertainty* selecionaram todas ou quase todas as instâncias disponíveis. Isso se deu pelo faço de que, ao se retardar a atualização do modelo, este permanece fraco durante todo o processo de seleção das instâncias; logo, todas elas tendem a apresentar um grau de incerteza e entropia altos, sendo selecionadas por essas duas estratégias.

Essa característica de as estratégias *Entropy* e *Uncertainty* selecionarem um número demasiadamente grande de documentos faz com que a *Variable Entropy*, proposta neste trabalho, se apresente como uma boa opção para Mineração de Opinião com *data streams*. Já que ela obteve resultados estatisticamente semelhantes às outras duas em todos os cenários avaliados, enquanto mantém o conjunto de treinamento em um tamanho bem menor.

5.4 CENÁRIO III: NÚMERO FIXO DE INSTÂNCIAS

Por fim, o Cenário III tem como objetivo comparar as quatro medidas utilizadas pelas estratégias de seleção: aleatoriedade, incerteza, ganho de informação e entropia, fixando a quantidade de instâncias selecionadas para todas as técnicas. As Tabelas 28, 29 e 30 trazem os resultados deste cenário para a seleção de 33% das bases de dados, enquanto as Tabelas 31, 32 e 33 os trazem considerando a seleção de uma quantidade de instâncias correspondente a 50% de cada base.

Primeiramente, pode-se observar que, utilizando os classificadores MNB e RL, as estratégias *Uncertainty* e *Entropy* obtiveram *f-measures* extamaente iguais em todas as bases de dados com duas classes. Sendo assim, percebe-se que, nessas situações, ambas as estratégias selecionaram as mesmas instâncias, apontando o quão semelhantes são as medidas de incerteza e entropia. Já para as bases de dados com três classes e utilizando o classificar SVM, esse padrão não se repetiu, indicando que os documentos selecionados foram diferentes.

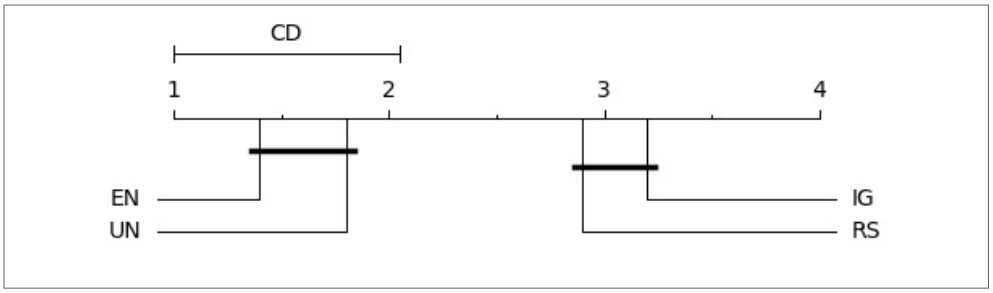
Isso mostra que as medidas de incerteza e entropia se comportam exatamente da mesma forma ao se lidar com bases de duas classes utilizando classificadores probabilísticos. Então, com essas configurações e em um cenário no qual tem-se uma quantidade fixa de instâncias a serem selecionadas, pode-se afirmar que não há diferenças na utilização de uma ou de outra.

Entretanto, conforme apresentado nas seções anteriores, a diferença da técnica *Entropy* para a *Uncertainty* reside no fato de que a primeira seleciona um número maior de instâncias e obtém, de maneira geral, resultados superiores. Então, para os casos em que não há um limite de documentos a serem selecionados, a estratégia baseada em entropia ainda pode ser considerada a melhor em termos de *f-measure*.

O teste de Friedman apontou que apenas as configurações utilizando o *Multinomial Naïve Bayes* com 33% dos dados (*p-value* de 0.3669) e utilizando o SVM com 50% (*p-value* de 0.0893) não apresentaram uma diferença significativa nos resultados. E, como

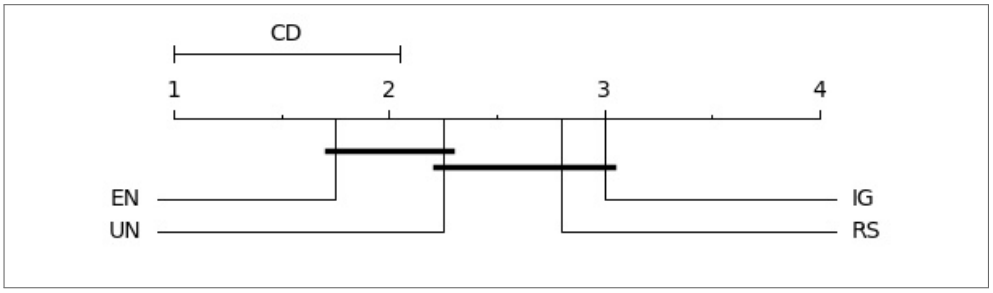
pode ser visto nos diagramas CD das Figuras 18, 19, 20 e 21, pelos outros casos, os quais apresentaram diferença estatística, nota-se que a estratégia *Entropy* apresentou resultados numericamente superiores às demais e, em três deles, só não superou estatisticamente a estratégia *Uncertainty*.

Figura 18 – Resultado do pós-teste Nemenyi para o Cenário III com o classificador RL e selecionando 33% dos dados.



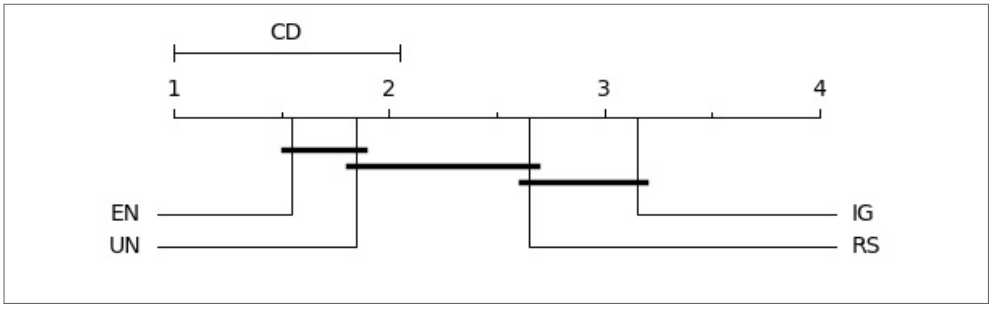
Fonte: próprio autor (2020).

Figura 19 – Resultado do pós-teste Nemenyi para o Cenário III com o classificador SVM e selecionando 33% dos dados.



Fonte: próprio autor (2020).

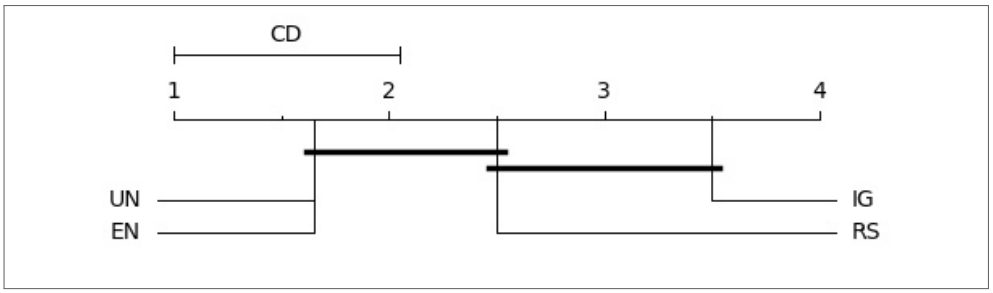
Figura 20 – Resultado do pós-teste Nemenyi para o Cenário III com o classificador MNB e selecionando 50% dos dados.



Fonte: próprio autor (2020).

O fraco desempenho da *Random Sampling*, corroborado pela sua performance nos cenários anteriores, mostra que a escolha de uma medida para avaliação das instâncias em um processo de *Active Learning* é importante, e que a seleção realizada de forma

Figura 21 – Resultado do pós-teste Nemenyi para o Cenário III com o classificador RL e selecionando 50% dos dados.



Fonte: próprio autor (2020).

puramente randômica não é útil. E como, mais uma vez, a estratégia *Information Gain* apresentou resultados inferiores às outras, torna-se claro que as medidas de incerteza e entropia, e suas variantes, são as melhores opções para *Active Learning* em aplicações de Mineração de Opinião.

Caso a quantidade de instâncias a serem rotuladas não seja um empecilho, a estratégia *Entropy* desponta como a mais indicada. Porém, caso não haja recurso suficiente para anotar uma grande quantidade de dados, a *Variable Entropy* pode ser mais recomendável.

6 CONCLUSÕES

Neste estudo, foi realizada uma comparação de estratégias de seleção de instâncias para *Active Learning* em aplicações de Mineração de Opinião a nível de documento com fluxos contínuos de dados (*data streams*), com o objetivo de avaliar quais estratégias são mais indicadas para cada cenário avaliado. Para isso, foram selecionadas seis técnicas de seleção oriundas da literatura, cujos usos foram reportados em aplicações de Mineração de Opinião e Mineração de Texto: *Random Sampling*, *Uncertainty*, *Variable Uncertainty*, *Variable Randomized Uncertainty*, *Information Gain* e *Entropy*; além de terem sido propostas outras duas técnicas baseadas naquelas encontradas na literatura: *Variable Entropy* e *Variable Randomized Entropy*.

Além disso, dois corpora contendo *data streams* foram construídos, disponibilizados e tiveram seus processos de construção relatados nesta dissertação. Os dois corpora, que contêm dados oriundos do Twitter e Facebook acerca da Eleição Presidencial no Brasil em 2018, foram utilizados para avaliar as estratégias de seleção juntamente com mais duas bases de dados encontradas na Internet: *Sentiment140* e *Sanders*.

Com base no objetivo desta pesquisa e nos resultados alcançados com os experimentos, algumas conclusões puderam ser obtidas, de acordo com os cenários avaliados:

1. Primeiramente pôde-se notar que a utilização de estratégias de *Active Learning* conseguiu obter desempenhos superiores ao modelo treinado com todo o conjunto de dados, mostrando que a abordagem de AL, além de reduzir a quantidade de instâncias que necessitam ser rotuladas, também está apta a superar o modelo tradicional de Aprendizagem de Máquina Supervisionada.

2. Também observou-se que as estratégias que selecionam em média mais instâncias são aquelas que obtêm resultados melhores, ao passo que aquelas que escolhem uma pequena quantidade de documentos apresentam um desempenho inferior. Isso pode indicar uma relação diretamente proporcional entre a quantidade de padrões selecionados e a *f-measure* do modelo.

3. Ao se analisar o parâmetro θ, presente em seis das oito estratégias comparadas, pôde-se notar que a escolha do valor deste só apresenta um impacto significativo no desempenho da estratégia *Uncertainty*, a qual se mostrou bastante dependente dessa escolha.

4. A estratégia *Entropy* se mostrou a melhor em termos de *f-measure* em praticamente todos os cenários, confirmando o apontado por Yang e Loog (2018): que a utilização da entropia como medida de incerteza consegue produzir o melhor desempenho em uma grande quantidade de bases de dados. E, ainda considerando a *f-measure*,

pode-se apontar a *Uncertainty* como a segunda melhor estratégia para Mineração de Opinião com *data streams*.

5. Ambas as medidas utilizadas nas estratégias que apresentaram os melhores desempenhos (incerteza e entropia) obtiveram resultados praticamente iguais considerando o cenário no qual o número de instâncias foi fixada para todas as estratégias. Sendo assim, observou-se que a diferença entre as duas técnicas de seleção está na quantidade de documentos escolhidos: a *Entropy* apresenta um desempenho superior por selecionar mais padrões a serem rotulados.

6. Porém, o fato de a *Entropy* selecionar uma quantidade muito grande de documentos constitui uma desvantagem para esta estratégia, já que manter o conjunto de treinamento o menor possível é uma característica importante da abordagem de AL (ZHU et al., 2007). Por conta disso, este método de seleção pode não ser uma escolha muito boa em situações nas quais não se pode realizar a anotação de um grande volume de dados. Nestes casos, mais recomendável é a estratégia *Variable Entropy*, a qual requer uma menor quantidade de dados e apesentou resultados estatisticamente semelhantes à *Entropy*.

7. A técnica *Information Gain*, embora tenha se mostrado a melhor escolha no estudo de Zimmermann, Ntoutsi e Spiliopoulou (2015), os quais a propuseram, obteve os piores resultados em todos os cenários desta dissertação. E o desempenho alcançado pela estratégia *Random Sampling*, a qual não apresentou resultados expressivos na grande maioria das bases e cenários, mostra que a escolha aleatória de instâncias também não é útil. Sendo assim, essas observações corroboram com a afirmação de que as estratégias baseadas em incerteza e entropia constituem as melhores escolhas para MO com *Active Learning*.

8. O cenário no qual ocorria uma atualização constante e iterativa do modelo de aprendizagem apresentou resultados estatisticamente superiores àquele em que o classificador só era atualizado ao final do processo de seleção das instâncias, considerando o algoritmo *Multinomial Naïve Bayes*. Isto indica que, para este classificador, um processo iterativo de AL é o mais recomendável. Já para o classificador de Regressão Logística, esta diferença não foi observada.

9. A escolha do tipo de classificador pode ter um impacto na utilização de estratégias de *Active Learning*, já que, a depender do cenário avaliado, as técnicas de seleção obtiveram resultados distintos para cada algoritmo de classificação: apresentando desempenhos superiores com algum classificador e inferiores com outro. Porém, não é possível indicar com precisão qual o melhor algoritmo de Aprendizagem de Máquina para se utilizar com AL.

10. Por fim, o classificador SVM não se mostrou aconselhável para utilização nos cenários de Mineração de Opinião com fluxos de dados aqui avaliados. Seu elevado custo computacional de treinamento o torna inviável para ser utilizado em um cenário no qual a atualização do modelo é constante: o algoritmo necessita de um tempo elevado para ser treinado, ao passo que, ao se lidar com *data streams*, a velocidade com que os dados chegam pode requerer uma atualização rápida do modelo; além disso, o desempenho deste algoritmo não se mostrou superior aos resultados alcançados pelo MNB e pelo RL, sendo inclusive inferior a este último na maioria das bases de dados.

6.1 LIMITAÇÕES

Uma das limitações desta pesquisa se encontra no fato de terem sido utilizadas bases de dados oriundas de poucos corpora. Esta limitação se deu pela existência de poucos *data sets* publicamente disponíveis para uso que contenham *data streams*, e foi mitigada pela criação de dois corpora adicionais e pela utilização de diferentes subconjuntos, oriundos de diferentes pontos dos fluxos de dados, dos corpora encontrados.

Outra limitação está presente no fato de que neste trabalho não foram avaliadas técnicas de pré-processamento dos dados. É possível que a adoção de pré-processamentos diferentes tenha um impacto na utilização de *Active Learning* em Mineração de Opinião; porém optou-se por fazer uso das técnicas mais simples possíveis e poupar tempo para uma análise mais detalhadas do foco desta pesquisa: as estratégia de seleção de AL.

A implementação em Python das estratégias de seleção também pode ser considerada uma limitação deste trabalho e uma ameaça à sua validade, já que esta foi realizada de forma manual, o que pode ter ocasionado prejuízo ao desempenho das estratégias, principalmente daquelas propostas por Žliobaitė et al. (2011) e Zimmermann, Ntoutsi e Spiliopoulou (2015). Porém, procurou-se reduzir o impacto desta ameaça seguindo estritamente os pseudo-códigos destas estratégias apresentados nos seus respectivos artigos e detalhados no Capítulo 3.

Por fim, pode-se julgar que a comparação de estratégias que usam praticamente as mesmas medidas para seleção das instâncias constitui outra limitação. Entretanto, conforme apontado durante toda esta dissertação, a escolha das estratégias se deu pela sua utilização em aplicações de Mineração de Opinião ou de Mineração de Texto. Sendo assim, não foi encontrada uma grande variedade de técnicas de seleção que cumprissem esses requisitos.

6.2 TRABALHOS FUTUROS

Como trabalho futuro, e baseando-se em uma das limitações apresentadas anteriormente, pode-se apontar que se faz necessária a realização das análises aqui propostas e feitas em um conjunto mais variado de bases de dados. Porém, esta análise é refém da

construção ou descoberta de novos corpora com *data streams* de redes sociais. Além disso, pode ser importante a realização das análises considerando técnicas de pré-processamento diferentes e mais robustas, visando demonstrar se as estratégias de seleção de AL se comportam da mesma maneira com pré-processamentos distintos.

Outro trabalho futuro que pode ser realizado é a análise dos corpora de fluxos de dados utilizados, com o objetivo de detectar a ocorrência de *concept drifts* e *opinion drifts*; já que neste trabalho utilizou-se uma abordagem implícita para lidar com a ocorrência de *drifts*, não sendo necessária a identificação deles.

E, conforme apresentado no trabalho de Vitório, Souza e Oliveira (2019c), cuja ideia surgiu a partir desta dissertação, a utilização de estratégias de *Active Learning* pode ser uma abordagem útil para geração de modelos de classificação a serem utilizados em Sistemas de Múltiplos Classificadores (SMC) para aplicações de Mineração de Opinião. Além disso, esta abordagem pode ser utilizada em muitas subáreas de SMC, abrindo novas possibilidades de utilização das técnicas de AL e requerendo, portanto, o desenvolvimento de mais pesquisas.

6.3 CONTRIBUIÇÕES CIENTÍFICAS

O desenvolvimento desta dissertação resultou na publicação de dois *full papers*, contendo parte do trabalho aqui discutido, em conferências: uma nacional, o BraSNAM 2019, e outra internacional, o EPIA 2019. Além disso, outro artigo cuja ideia se originou nesta pesquisa, porém dentro de outra área de pesquisa não englobada neste trabalho, foi submetido, aceito e apresentado no BRACIS 2019.

1. **Título:** *Opinion Mining and Active Learning: a Comparison of Sampling Strategies* (VITÓRIO; SOUZA; OLIVEIRA, 2019b)
   Autores: Douglas Vitório, Ellen Souza e Adriano L. I. Oliveira
   Local de publicação: *VIII Brazilian Workshop on Social Network Analysis and Mining* (BraSNAM 2019) – Belém-PA, Brasil (**Qualis B5** em Ciência da Computação)
   Resumo: *There are two main problems when performing Opinion Mining (OM) with data streams: the lack of labeled data and the need to update the learning model. The most used OM techniques cannot deal well with these challenges, so, an alternative is to use semi-supervised methods, such as the Active Learning, which is a method to label only selected data rather than the entire data set; however, it requires the choice of a sampling strategy to select the data to be labeled. In this paper, we evaluated eight strategies in ten data sets, in order to identify the best ones for OM with Twitter streams. According to our experiments, the Entropy strategy showed the best results, but it selects a large number of instances to be labeled, requiring further investigation.*

2. **Título:** *Evaluating Active Learning Sampling Strategies for Opinion Mining in Brazilian Politics Corpora* (VITÓRIO; SOUZA; OLIVEIRA, 2019a)
   Autores: Douglas Vitório, Ellen Souza e Adriano L. I. Oliveira
   Local de publicação: *19th EPIA Conference on Artificial Intelligence* (EPIA 2019) – Vila Real, Portugal (**Qualis B1** em Ciência da Computação)
   Resumo: *Politics is a commonly used domain in Opinion Mining applications, in which opinions may change over time. Nevertheless, the usual approaches for Opinion Mining are not able to deal with the characteristics and the challenges brought by continuous data streams; so, an alternative is the use of techniques such as Active Learning, which labels selected data rather than the entire data set. The Active Learning approach requires the choice of a sampling strategy to select the most valuable instances. However, no study has performed an analysis in order to identify the best strategies for Opinion Mining. In this sense, we evaluated eight Active Learning sampling strategies, from which Entropy achieved the best results. In addition, due to the lack of publicly available stream data sets written in Portuguese, we created and evaluated corpora from Twitter and Facebook about the 2018 Brazilian presidential elections.*

3. **Título:** *Using Active Learning Sampling Strategies for Ensemble Generation on Opinion Mining* (VITÓRIO; SOUZA; OLIVEIRA, 2019c)
   Autores: Douglas Vitório, Ellen Souza e Adriano L. I. Oliveira
   Local de publicação: *8th Brazilian Conference on Intelligent Systems* (BRACIS 2019) – Salvador-BA, Brasil (**Qualis B2** em Ciência da Computação)
   Resumo: *The lack of labeled data and the need to update the learning model are the two main challenges when performing Opinion Mining with data streams. Usual Opinion Mining approaches are not able to deal with these challenges, nor with the characteristics brought by this kind of data. Moreover, the occurrence of changes (drifts) in the concepts and/or opinions is another issue. Possible alternative solutions to these problems are: the use of Semi-supervised learning, such as Active Learning, which labels selected data rather than the entire data set; or Multiple Classifiers Systems, which combines different classifiers and are well suited to deal with drifts. In this study, we combined these two approaches, proposing the use of eight Active Learning sampling strategies as a generation method for Multiple Classifiers Systems. The Active Learning approach requires the choice of a strategy to select the instances, and each strategy results in the creation of a distinct classification model. Our method was evaluated in 14 Twitter stream data sets and the results showed that it can be better for Opinion Mining with data streams than popular ensemble generation methods present in the literature, such as Bagging and AdaBoost.*

REFERÊNCIAS

ALDOĞAN, D.; YASLAN, Y. A comparison study on active learning integrated ensemble approaches in sentiment analysis. *Computers & Electrical Engineering*, v. 57, p. 311 – 323, 2017.

ALLCOTT, H.; GENTZKOW, M. Social media and fake news in the 2016 election. *Journal of economic perspectives*, v. 31, n. 2, p. 211–36, 2017.

ALVES, A. L. F.; BAPTISTA, C. d. S.; FIRMINO, A. A.; OLIVEIRA, M. G. d.; PAIVA, A. C. d. A comparison of svm versus naive-bayes techniques for sentiment analysis in tweets: A case study with the 2013 fifa confederations cup. In: *Proceedings of the 20th Brazilian Symposium on Multimedia and the Web*. Nova York, NY, EUA: ACM, 2014. (WebMedia '14), p. 123—130.

ASTON, N.; LIDDLE, J.; HU, W. Twitter sentiment in data streams with perceptron. *Journal of Computer and Communications*, v. 2, n. 03, p. 11, 2014.

ASTON, N.; MUNSON, T.; LIDDLE, J.; HARTSHAW, G.; LIVINGSTON, D.; HU, W. Sentiment analysis on the social networks using stream algorithms. *Journal of Data Analysis and Information Processing*, v. 2, n. 02, p. 60, 2014.

BALAZS, J. A.; VELÁSQUEZ, J. D. Opinion Mining and Information Fusion: A survey. *Information Fusion*, v. 27, p. 95–110, 2016.

BIFET, A. *Adaptive Learning and Mining for Data Streams and Frequent Patterns*. Tese (Doutorado) — Universitat Politècnica de Catalunya, 2009.

CARDOSO-CACHOPO, A.; OLIVEIRA, A. L. Semi-supervised single-label text categorization using centroid-based classifiers. In: *Proceedings of the 2007 ACM Symposium on Applied Computing*. New York, NY, USA: ACM, 2007. (SAC '07), p. 844–851. ISBN 1-59593-480-4.

COHEN, J. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, v. 20, n. 1, p. 37–46, 1960.

CORTES, C.; VAPNIK, V. Support-vector networks. *Machine learning*, Springer, v. 20, n. 3, p. 273–297, 1995.

DANKA, T.; HORVATH, P. modAL: A modular active learning framework for Python. 2018. Available on arXiv at <https://arxiv.org/abs/1805.00979>. Disponível em: <https://github.com/cosmic-cortex/modAL>.

DANKA, T.; HORVATH, P. *Uncertainty sampling*. 2018. Disponível em: <https://modal-python.readthedocs.io/en/latest/content/query_strategies/uncertainty_sampling.html>. Acesso em: outubro de 2019.

DAVE, K.; LAWRENCE, S.; PENNOCK, D. M. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In: *Proceedings of the 12th International Conference on World Wide Web*. Nova York, NY, EUA: ACM, 2003. (WWW '03), p. 519–528.

EVANGELISTA, T. R.; PADILHA, T. P. P. Monitoramento de posts sobre empresas de e-commerce em redes sociais utilizando análise de sentimentos. In: SBC. *Anais do III Brazilian Workshop on Social Network Analysis and Mining*. [S.l.], 2014. p. 152–163.

FELDMAN, R. Techniques and applications for sentiment analysis. *Commun. ACM*, ACM, Nova York, NY, EUA, v. 56, n. 4, p. 82–89, abr. 2013. ISSN 0001-0782.

FELDMAN, R.; SANGER, J. *Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. New York, NY, USA: Cambridge University Press, 2006. ISBN 0521836573, 9780521836579.

FRIEDMAN, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the american statistical association*, Taylor & Francis, v. 32, n. 200, p. 675–701, 1937.

GAMA, J. *Knowledge Discovery from Data Streams*. 1st. ed. [S.l.]: Chapman & Hall/CRC, 2010. ISBN 1439826110, 9781439826119.

GO, A.; BHAYANI, R.; HUANG, L. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, v. 1, n. 12, 2009.

GUERRA, P. C.; MEIRA JR., W.; CARDIE, C. Sentiment analysis on evolving social streams: How self-report imbalances can help. In: *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*. [S.l.: s.n.], 2014. p. 443–452. ISBN 978-1-4503-2351-2.

HSU, C.-W.; CHANG, C.-C.; LIN, C.-J. *A practical guide to support vector classification*. 2003.

KRANJC, J.; SMAILOVIĆ, J.; PODPEČAN, V.; GRČAR, M.; ŽNIDARŠIČ, M.; LAVRAČ, N. Active learning for sentiment analysis on data streams: Methodology and workflow implementation in the clowdflows platform. *Information Processing & Management*, v. 51, n. 2, p. 187 – 203, 2015. ISSN 0306-4573.

KRAWCZYK, B.; MINKU, L. L.; GAMA, J.; STEFANOWSKI, J.; WOŹNIAK, M. Ensemble learning for data stream analysis: A survey. *Information Fusion*, v. 37, p. 132 – 156, 2017. ISSN 1566-2535.

KUMARI, A. Study on naive bayesian classifier and its relation to information gain. *International Journal on Recent and Innovation Trends in Computing and Communication*, v. 2, p. 601–603, 2014.

LEWIS, D. D.; GALE, W. A. A sequential algorithm for training text classifiers. In: SPRINGER. *SIGIR'94*. [S.l.], 1994. p. 3–12.

LI, Y.; LV, Y.; WANG, S.; LIANG, J.; LI, J.; LI, X. Cooperative hybrid semi-supervised learning for text sentiment classification. *Symmetry*, Multidisciplinary Digital Publishing Institute, v. 11, n. 2, p. 133, 2019.

LIU, B.; ZHANG, L. A survey of opinion mining and sentiment analysis. In: _____. *Mining Text Data*. Boston, MA, EUA: Springer US, 2012. p. 415–463. ISBN 978-1-4614-3223-4.

MARINE-ROIG, E.; CLAVÉ, S. A. Tourism analytics with massive user-generated content: A case study of barcelona. *Journal of Destination Marketing and Management*, v. 4, n. 3, p. 162–172, 2015.

MARRS, G. R.; HICKEY, R. J.; BLACK, M. M. The impact of latency on online classification learning with concept drift. In: SPRINGER. *International Conference on Knowledge Science, Engineering and Management*. [S.l.], 2010. p. 459–469.

MCCALLUM, A.; NIGAM, K. A comparison of event models for naive bayes text classification. In: CITESEER. *Proceedings of the AAAI-98 Workshop on Learning for Text Categorization*. [S.l.], 1998. v. 752, p. 41–48.

MEDHAT, W.; HASSAN, A.; KORASHY, H. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, v. 5, n. 4, p. 1093 – 1113, 2014.

NEMENYI, P. *Distribution-free multiple comparisons*. Tese (Doutorado) — Princeton University, 1963.

PAK, A.; PAROUBEK, P. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In: *Proceedings of the 7th International Conference on Language Resources and Evaluation*. Paris, França: European Language Resources Association (ELRA), 2010. (LREC '10), p. 1320–1326.

PANG, B.; LEE, L. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, v. 2, n. 1-2, p. 1–135, 2008.

RAVI, K.; RAVI, V. A survey on opinion mining and sentiment analysis: Tasks, approaches and applications. *Knowledge-Based Systems*, v. 89, p. 14–46, 2015.

SALEIRO, P.; SARMENTO, L.; RODRIGUES, E. M.; SOARES, C.; OLIVEIRA, E. Learning word embeddings from the portuguese twitter stream: A study of some practical aspects. In: *Progress in Artificial Intelligence*. [S.l.: s.n.], 2017. p. 880–891.

SANDERS, N. J. *Twitter Sentiment Corpus*. 2011.

SETTLES, B. *Active learning literature survey*. [S.l.], 2009.

SILVA, I. S.; GOMIDE, J.; VELOSO, A.; MEIRA JR., W.; FERREIRA, R. Effective sentiment stream analysis with self-augmenting training and demand-driven projection. In: *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*. [S.l.: s.n.], 2011. p. 475–484.

SMAILOVIĆ, J.; GRČAR, M.; LAVRAČ, N.; ŽNIDARŠIČ, M. Stream-based active learning for sentiment analysis in the financial domain. *Inf. Sci.*, v. 285, n. C, p. 181–203, nov. 2014. ISSN 0020-0255.

SOUZA, E.; COSTA, D.; CASTRO, D. W.; VITÓRIO, D.; TELES, I.; ALMEIDA, R.; ALVES, T.; OLIVEIRA, A. L. I.; GUSMÃO, C. Characterising text mining: a systematic mapping review of the portuguese language. *IET Software*, v. 12, n. 2, p. 49–75, 2018. ISSN 1751-8806.

SOUZA, E.; VITÓRIO, D.; CASTRO, D.; OLIVEIRA, A. L. I.; GUSMÃO, C. Characterizing opinion mining: A systematic mapping study of the portuguese language. In: _____. *Computational Processing of the Portuguese Language: 12th International Conference, PROPOR 2016, Tomar, Portugal, July 13-15, 2016, Proceedings*. Cham: Springer International Publishing, 2016. p. 122–127. ISBN 978-3-319-41552-9.

VITÓRIO, D.; SOUZA, E.; OLIVEIRA, A. L. I. Evaluating active learning sampling strategies for opinion mining in brazilian politics corpora. In: *19th EPIA Conference on Artificial Intelligence*. Cham: Springer, 2019. p. 695–707.

VITÓRIO, D.; SOUZA, E.; OLIVEIRA, A. L. I. Opinion mining and active learning: a comparison of sampling strategies. In: *Anais do VIII Brazilian Workshop on Social Network Analysis and Mining*. Porto Alegre: SBC, 2019. p. 61–70.

VITÓRIO, D.; SOUZA, E.; OLIVEIRA, A. L. I. Using active learning sampling strategies for ensemble generation on opinion mining. In: *8th Brazilian Conference on Intelligent Systems (BRACIS)*. [S.l.]: IEEE, 2019. p. 114–119.

WAGNER, S.; ZIMMERMANN, M.; NTOUTSI, E.; SPILIOPOULOU, M. Ageing-based multinomial naive bayes classifiers over opinionated data streams. In: *Proceedings of the 2015th European Conference on Machine Learning and Knowledge Discovery in Databases - Volume Part I*. [S.l.: s.n.], 2015. p. 401–416.

WANG, D.; FENG, S.; WANG, D.; YU, G. Detecting opinion drift from chinese web comments based on sentiment distribution computing. In: *Web Information Systems Engineering – WISE 2013*. [S.l.: s.n.], 2013. p. 72–81. ISBN 978-3-642-41230-1.

WEISS, S.; INDURKHYA, N.; ZHANG, T.; DAMERAU, F. *Text Mining: Predictive Methods for Analyzing Unstructured Information*. Nova York, NY, EUA: Springer-Verlag, 2005. ISBN 0387954333.

WIDMER, G.; KUBAT, M. Learning in the presence of concept drift and hidden contexts. *Mach. Learn.*, v. 23, n. 1, p. 69–101, abr. 1996.

WILCOXON, F. Individual comparisons by ranking methods. *Biometrics Bulletin*, [International Biometric Society, Wiley], v. 1, n. 6, p. 80–83, 1945. ISSN 00994987.

YANG, Y.; LOOG, M. A benchmark and comparison of active learning for logistic regression. *Pattern Recognition*, v. 83, p. 401 – 415, 2018.

YU, H.-F.; HUANG, F.-L.; LIN, C.-J. Dual coordinate descent methods for logistic regression and maximum entropy models. *Machine Learning*, Springer, v. 85, n. 1-2, p. 41–75, 2011.

ZHU, X.; ZHANG, P.; LIN, X.; SHI, Y. Active learning from data streams. In: *Seventh IEEE International Conference on Data Mining (ICDM 2007)*. [S.l.: s.n.], 2007. p. 757–762.

ZHU, X.; ZHANG, P.; LIN, X.; SHI, Y. Active learning from stream data using optimal weight classifier ensemble. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, v. 40, n. 6, p. 1607–1621, Dec 2010.

ZIMMERMANN, M.; NTOUTSI, E.; SPILIOPOULOU, M. Incremental active opinion learning over a stream of opinionated documents. *arXiv preprint arXiv:1509.01288*, 2015.

ŽLIOBAITĖ, I.; BIFET, A.; PFAHRINGER, B.; HOLMES, G. Active learning with evolving streaming data. In: *Machine Learning and Knowledge Discovery in Databases*. [S.l.: s.n.], 2011. p. 597–612.

APÊNDICE A – RESULTADOS DA AVALIAÇÃO DOS PARÂMETROS B E θ

Fonte: próprio autor (2020).

#	**Base**	$B = \mathbf{0.1}$	$B = \mathbf{0.2}$	$B = \mathbf{0.3}$	$B = \mathbf{0.4}$	$B = \mathbf{0.5}$
1.	Bolsonaro_facebook	0.6121	0.7761	**0.8447**	0.7679	0.7534
2.	Bolsonaro_twitter	0.5525	0.6169	0.7042	0.7284	**0.7538**
3.	Haddad_facebook	0.6218	0.8449	0.8355	0.8188	**0.8944**
4.	Haddad_twitter	0.5602	0.4594	0.6343	0.5655	**0.6587**
5.	Ambos_facebook	0.6697	0.7984	0.8584	0.8720	**0.8780**
6.	Ambos_twitter	0.6643	0.7486	0.6812	0.7408	**0.7523**
7.	Sentiment140_test	0.3717	0.4529	0.5960	0.5570	**0.5982**
8.	Sentiment140_10000_1	0.6815	0.6926	0.7009	**0.7129**	0.7066
9.	Sentiment140_5000_1	0.6038	0.6206	0.6407	**0.6716**	0.6709
10.	Sentiment140_2500_1	0.6883	0.6684	0.7086	**0.7120**	0.6947
11.	Sentiment140_1000_1	0.4890	0.5731	0.5793	0.6138	**0.6275**
12.	Sentiment140_10000_2	0.6874	0.6892	**0.7177**	0.7136	0.7169
13.	Sentiment140_5000_2	0.6784	0.6531	0.6839	0.6975	**0.7081**
14.	Sentiment140_2500_2	0.5915	0.6152	0.6555	**0.6975**	0.6603
15.	Sentiment140_1000_2	0.4923	0.4291	0.4566	**0.6383**	0.6146
16.	Sanders_apple	0.4239	0.4193	0.4291	0.4374	**0.4581**
17.	Sanders_google	0.3655	0.4348	**0.4386**	0.4313	0.3927
18.	Sanders_microsoft	0.2697	0.3237	**0.3615**	0.3237	0.3480
19.	Sanders_twitter	**0.2884**	**0.2884**	0.2874	0.2864	0.2874
20.	Sanders_all	0.3580	0.3929	**0.4080**	0.3677	0.3897
-	**Média**	0.5300	0.5749	0.6111	0.6177	**0.6282**
-	**Ranking médio**	4.50	3.55	2.45	2.45	**1.90**
-	**Quantidade vitórias**	1	1	5	5	**9**

Tabela 3 – Resultados da avaliação do parâmetro B com o classificador MNB.

Fonte: próprio autor (2020).

#	**Base**	$B = \mathbf{0.1}$	$B = \mathbf{0.2}$	$B = \mathbf{0.3}$	$B = \mathbf{0.4}$	$B = \mathbf{0.5}$
1.	Bolsonaro_facebook	0.8404	0.8646	**0.8873**	0.8738	0.8544
2.	Bolsonaro_twitter	0.6375	0.6354	0.6950	0.6858	**0.7574**
3.	Haddad_facebook	0.8293	0.8306	0.8707	0.8679	**0.8737**
4.	Haddad_twitter	0.5803	0.5756	0.6822	**0.6853**	0.6798
5.	Ambos_facebook	0.7560	0.8488	**0.8700**	0.8686	0.8678
6.	Ambos_twitter	0.6598	**0.7206**	0.6833	0.6993	0.6741
7.	Sentiment140_test	0.4156	0.4899	0.5694	0.6110	**0.6771**
8.	Sentiment140_10000_1	0.6842	0.7042	0.7012	**0.7215**	0.7212
9.	Sentiment140_5000_1	0.6372	0.6566	0.6670	0.6764	**0.6801**
10.	Sentiment140_2500_1	0.6666	0.6943	0.7104	0.6732	**0.7276**
11.	Sentiment140_1000_1	0.5432	0.5933	0.6488	0.6298	**0.6805**
12.	Sentiment140_10000_2	0.6885	0.7043	0.7181	0.7209	**0.7212**
13.	Sentiment140_5000_2	0.6662	0.6738	0.7019	0.6970	**0.7076**
14.	Sentiment140_2500_2	0.5997	0.6151	0.6320	**0.6539**	**0.6539**
15.	Sentiment140_1000_2	0.4735	0.5428	0.4981	0.5698	**0.6253**
16.	Sanders_apple	0.4272	0.4591	**0.4896**	0.4798	0.4530
17.	Sanders_google	0.4298	0.4484	0.3968	0.4361	**0.4721**
18.	Sanders_microsoft	0.3431	0.3765	0.3166	**0.4127**	0.3960
19.	Sanders_twitter	0.3310	**0.3597**	0.3564	0.3349	**0.3597**
20.	Sanders_all	0.4023	0.4159	0.4157	0.4202	**0.4491**
-	**Média**	0.5806	0.6103	0.6255	0.6359	**0.6516**
-	**Ranking médio**	4.75	3.40	2.75	2.25	**1.75**
-	**Quantidade vitórias**	0	2	3	4	**13**

Tabela 4 – Resultados da avaliação do parâmetro B com o classificador RL.

Fonte: próprio autor (2020).

#	**Base**	θ **= 0.5**	θ **= 0.6**	θ **= 0.7**	θ **= 0.8**	θ **= 0.9**
1.	Bolsonaro_facebook	0.4570	0.4684	**0.7464**	0.7240	0.7276
2.	Bolsonaro_twitter	0.3098	0.3431	0.4751	0.6035	**0.6337**
3.	Haddad_facebook	0.3650	0.3719	0.5234	0.6916	**0.7080**
4.	Haddad_twitter	0.5113	0.5764	0.5907	0.6921	**0.7358**
5.	Ambos_facebook	0.4437	0.5942	0.8388	0.8224	**0.8663**
6.	Ambos_twitter	0.6658	0.7131	0.7083	**0.7562**	0.7440
7.	Sentiment140_test	0.2427	0.4696	0.5765	0.5898	**0.6419**
8.	Sentiment140_10000_1	0.6625	0.6900	0.7016	0.7105	**0.7164**
9.	Sentiment140_5000_1	0.5550	0.6423	0.6597	0.6855	**0.6931**
10.	Sentiment140_2500_1	0.6396	0.6895	0.7092	0.7036	**0.7271**
11.	Sentiment140_1000_1	0.3339	0.4715	0.5659	0.5925	**0.6382**
12.	Sentiment140_10000_2	0.6045	0.7025	0.7186	0.7149	**0.7351**
13.	Sentiment140_5000_2	0.6502	0.6754	0.7039	0.6939	**0.7111**
14.	Sentiment140_2500_2	0.5279	0.6041	0.6502	**0.6841**	0.6740
15.	Sentiment140_1000_2	0.3694	0.3694	0.3679	0.3679	**0.5023**
16.	Sanders_apple	0.3821	0.4475	0.4411	0.4586	**0.4844**
17.	Sanders_google	0.3265	0.3923	0.4158	**0.4345**	0.4316
18.	Sanders_microsoft	0.2851	**0.2999**	**0.2999**	**0.2999**	**0.2999**
19.	Sanders_twitter	**0.2884**	**0.2884**	**0.2884**	**0.2884**	**0.2884**
20.	Sanders_all	0.3606	0.3664	0.3887	0.3952	**0.4064**
-	**Média**	0.4491	0.5088	0.5685	0.5955	**0.6213**
-	**Ranking médio**	4.65	3.50	2.65	2.10	**1.20**
-	**Quantidade vitórias**	1	2	3	5	**16**

Tabela 5 – Resultados da avaliação do parâmetro θ para a estratégia *Uncertainty* com o classificador MNB.

Fonte: próprio autor (2020).

#	**Base**	θ **= 0.5**	θ **= 0.6**	θ **= 0.7**	θ **= 0.8**	θ **= 0.9**
1.	Bolsonaro_facebook	0.7536	0.8660	0.8735	**0.8919**	0.8861
2.	Bolsonaro_twitter	0.4590	0.6650	0.6849	0.7158	**0.7226**
3.	Haddad_facebook	0.5517	0.8672	0.8874	**0.8973**	0.8941
4.	Haddad_twitter	0.4881	0.6409	0.6272	0.6429	**0.6882**
5.	Ambos_facebook	0.6029	0.8057	0.8540	0.8634	**0.8723**
6.	Ambos_twitter	0.6335	0.6331	0.7019	0.7114	**0.7196**
7.	Sentiment140_test	0.5940	0.5904	0.6439	0.6598	**0.6886**
8.	Sentiment140_10000_1	0.6738	0.7012	0.7256	**0.7313**	0.7286
9.	Sentiment140_5000_1	0.6074	0.6552	0.6705	0.6897	**0.6959**
10.	Sentiment140_2500_1	0.6431	0.6786	0.7168	0.7256	**0.7289**
11.	Sentiment140_1000_1	0.4511	0.5873	0.6607	0.6656	**0.6704**
12.	Sentiment140_10000_2	0.6610	0.7165	0.7339	0.7353	**0.7444**
13.	Sentiment140_5000_2	0.6473	0.6998	0.7196	0.7228	**0.7255**
14.	Sentiment140_2500_2	0.5583	0.6479	0.6447	0.6575	**0.6813**
15.	Sentiment140_1000_2	0.4008	0.5292	0.5650	0.6093	**0.6557**
16.	Sanders_apple	0.4786	0.5003	0.5089	0.5323	**0.5702**
17.	Sanders_google	0.4709	0.4440	**0.5318**	0.4593	0.4844
18.	Sanders_microsoft	0.3305	0.3326	0.3257	0.4019	**0.4316**
19.	Sanders_twitter	0.3130	0.2884	0.2884	0.3139	**0.3809**
20.	Sanders_all	0.4089	0.4204	0.4264	0.4324	**0.4450**
-	**Média**	0.5364	0.6135	0.6395	0.6530	**0.6707**
-	**Ranking médio**	4.65	4.00	3.15	1.95	**1.20**
-	**Quantidade vitórias**	0	0	1	3	**16**

Tabela 6 – Resultados da avaliação do parâmetro θ para a estratégia *Uncertainty* com o classificador RL.

Fonte: próprio autor (2020).

#	**Base**	θ **= 0.5**	θ **= 0.6**	θ **= 0.7**	θ **= 0.8**	θ **= 0.9**
1.	Bolsonaro_facebook	0.7718	**0.8392**	0.7679	0.7978	0.8003
2.	Bolsonaro_twitter	0.7663	**0.7813**	0.7423	0.7443	0.7791
3.	Haddad_facebook	0.8839	0.8904	0.8770	0.8940	**0.9103**
4.	Haddad_twitter	0.7106	**0.7522**	0.7174	0.7034	0.7216
5.	Ambos_facebook	0.8906	0.8917	**0.9052**	0.8739	0.8954
6.	Ambos_twitter	**0.7500**	0.7421	0.7455	0.7370	0.7381
7.	Sentiment140_test	0.6009	0.6112	0.5856	0.6129	**0.6134**
8.	Sentiment140_10000_1	0.7100	0.7110	0.7152	**0.7159**	0.7123
9.	Sentiment140_5000_1	**0.6889**	0.6808	0.6885	0.6793	0.6878
10.	Sentiment140_2500_1	0.7056	0.7219	**0.7340**	0.6994	0.7174
11.	Sentiment140_1000_1	0.6496	**0.6760**	0.6275	0.6417	0.6310
12.	Sentiment140_10000_2	0.7248	0.7252	0.7173	0.7189	**0.7259**
13.	Sentiment140_5000_2	0.6986	0.6992	0.6982	**0.7014**	0.6955
14.	Sentiment140_2500_2	0.6682	0.6852	0.6788	0.6826	**0.6853**
15.	Sentiment140_1000_2	0.5124	0.4983	0.5429	**0.5893**	0.5515
16.	Sanders_apple	**0.4868**	0.4705	0.4711	0.4864	0.4710
17.	Sanders_google	**0.4438**	0.4126	0.4294	0.4345	0.4164
18.	Sanders_microsoft	0.3842	**0.4271**	0.3801	0.3913	0.4127
19.	Sanders_twitter	0.2864	0.2854	**0.2874**	**0.2874**	0.2864
20.	Sanders_all	0.4145	**0.4149**	0.3997	0.3997	0.4035
-	**Média**	0.6374	**0.6458**	0.6355	0.6396	0.6427
-	**Ranking médio**	3.10	2.70	3.45	3.05	**2.60**
-	**Quantidade vitórias**	4	**6**	3	4	4

Tabela 7 – Resultados da avaliação do parâmetro θ para a estratégia *Variable Uncertainty* com o classificador MNB.

Fonte: próprio autor (2020).

#	**Base**	θ **= 0.5**	θ **= 0.6**	θ **= 0.7**	θ **= 0.8**	θ **= 0.9**
1.	Bolsonaro_facebook	0.8968	0.8828	**0.8978**	**0.8978**	0.8939
2.	Bolsonaro_twitter	**0.7485**	0.7134	0.7073	0.7266	0.7229
3.	Haddad_facebook	**0.8942**	0.8908	0.8909	0.8843	0.8843
4.	Haddad_twitter	0.6752	0.6560	0.6612	0.6495	**0.6829**
5.	Ambos_facebook	**0.8738**	0.8666	0.8672	0.8652	0.8698
6.	Ambos_twitter	0.6858	0.7079	0.7007	**0.7119**	0.6973
7.	Sentiment140_test	0.6047	**0.6242**	0.6112	0.6111	0.6035
8.	Sentiment140_10000_1	0.7257	**0.7317**	0.7291	0.7276	0.7267
9.	Sentiment140_5000_1	0.6734	0.6755	0.6739	**0.6808**	0.6794
10.	Sentiment140_2500_1	0.7215	0.7186	0.7276	0.7280	**0.7284**
11.	Sentiment140_1000_1	0.6349	0.6309	0.6374	**0.6467**	0.6374
12.	Sentiment140_10000_2	0.7378	0.7387	0.7342	0.7353	**0.7388**
13.	Sentiment140_5000_2	**0.7262**	0.7190	0.7196	0.7254	0.7203
14.	Sentiment140_2500_2	0.6539	0.6668	0.6480	0.6663	**0.6700**
15.	Sentiment140_1000_2	0.5904	0.6037	0.5937	0.6110	**0.6192**
16.	Sanders_apple	**0.5456**	0.5106	0.5154	0.5322	0.5274
17.	Sanders_google	0.5059	0.4968	**0.5078**	0.5045	0.4546
18.	Sanders_microsoft	0.4319	**0.4506**	0.4068	0.4202	0.4388
19.	Sanders_twitter	**0.4335**	0.4087	0.3861	0.3859	0.3859
20.	Sanders_all	**0.4576**	0.4381	0.4314	0.4233	0.4365
-	**Média**	**0.6609**	0.6566	0.6524	0.6571	0.6559
-	**Ranking médio**	2.80	3.15	3.30	2.80	**2.75**
-	**Quantidade vitórias**	**7**	3	2	4	5

Tabela 8 – Resultados da avaliação do parâmetro θ para a estratégia *Variable Uncertainty* com o classificador RL.

Fonte: próprio autor (2020).

#	**Base**	θ **= 0.5**	θ **= 0.6**	θ **= 0.7**	θ **= 0.8**	θ **= 0.9**
1.	Bolsonaro_facebook	0.8017	0.8411	0.8514	0.8470	**0.8576**
2.	Bolsonaro_twitter	0.7823	0.7207	0.7316	0.7616	**0.7881**
3.	Haddad_facebook	0.8776	0.8686	0.8605	0.8489	**0.8913**
4.	Haddad_twitter	0.5837	0.5639	**0.6733**	0.6112	0.6123
5.	Ambos_facebook	0.8712	0.8554	**0.8855**	0.8382	0.8612
6.	Ambos_twitter	0.7485	0.7418	0.7435	0.7189	**0.7630**
7.	Sentiment140_test	0.5105	0.5531	**0.6079**	0.5524	0.5852
8.	Sentiment140_10000_1	**0.7107**	0.7080	0.7096	0.7055	0.7081
9.	Sentiment140_5000_1	**0.6904**	0.6457	0.6883	0.6638	0.6695
10.	Sentiment140_2500_1	0.7011	0.7060	**0.7207**	0.6947	0.7021
11.	Sentiment140_1000_1	0.6263	0.6358	**0.6467**	0.6411	0.6332
12.	Sentiment140_10000_2	0.7181	0.7174	**0.7261**	0.7233	0.7126
13.	Sentiment140_5000_2	0.6948	0.6799	0.6944	0.7077	**0.7140**
14.	Sentiment140_2500_2	0.6623	**0.6771**	0.6768	0.6530	0.6758
15.	Sentiment140_1000_2	0.5851	0.5403	0.5993	0.6151	**0.6210**
16.	Sanders_apple	0.4674	0.4582	0.4777	0.4590	**0.4916**
17.	Sanders_google	**0.4264**	0.4233	0.4179	0.4086	0.4239
18.	Sanders_microsoft	0.3704	0.3328	0.3480	0.3217	**0.3890**
19.	Sanders_twitter	**0.3346**	0.2854	0.2864	0.2864	0.2864
20.	Sanders_all	0.3958	0.4036	**0.4193**	0.4040	0.4082
-	**Média**	0.6280	0.6179	0.6382	0.6231	**0.6397**
-	**Ranking médio**	2.95	3.90	2.20	3.70	**2.10**
-	**Quantidade vitórias**	4	1	7	0	**8**

Tabela 9 – Resultados da avaliação do parâmetro θ para a estratégia *Variable Randomized Uncertainty* com o classificador MNB.

Fonte: próprio autor (2020).

#	**Base**	θ **= 0.5**	θ **= 0.6**	θ **= 0.7**	θ **= 0.8**	θ **= 0.9**
1.	Bolsonaro_facebook	0.8610	0.8635	0.8421	**0.8789**	0.8459
2.	Bolsonaro_twitter	0.6898	0.7197	0.7054	**0.7305**	0.7002
3.	Haddad_facebook	0.8806	0.8773	0.8839	**0.8935**	0.8901
4.	Haddad_twitter	0.5606	0.6234	**0.6420**	0.6326	0.6400
5.	Ambos_facebook	0.8515	0.8652	0.8561	**0.8809**	0.8743
6.	Ambos_twitter	0.6921	0.6717	0.7249	0.7117	**0.7363**
7.	Sentiment140_test	0.6086	0.6091	**0.6455**	0.5880	0.6426
8.	Sentiment140_10000_1	0.7202	**0.7238**	0.7127	0.7217	0.7191
9.	Sentiment140_5000_1	0.6879	**0.7051**	0.6915	0.6859	0.6752
10.	Sentiment140_2500_1	0.7121	0.6995	0.7109	0.7142	**0.7182**
11.	Sentiment140_1000_1	**0.6445**	0.6289	0.6329	0.6388	0.6425
12.	Sentiment140_10000_2	**0.7284**	0.7216	0.7239	0.7200	0.7122
13.	Sentiment140_5000_2	**0.7252**	0.7171	0.7058	0.7021	0.7128
14.	Sentiment140_2500_2	0.6768	0.6724	**0.6773**	0.6423	0.6637
15.	Sentiment140_1000_2	0.5887	0.6109	0.5953	0.6117	**0.6292**
16.	Sanders_apple	**0.5160**	0.5116	0.4790	0.4961	0.5141
17.	Sanders_google	**0.4613**	0.4309	0.4526	0.4204	0.4322
18.	Sanders_microsoft	0.4101	**0.4226**	0.4047	0.3988	0.3792
19.	Sanders_twitter	0.3119	**0.3831**	0.3366	0.3613	0.3365
20.	Sanders_all	0.4279	0.4096	0.4181	0.4294	**0.4316**
-	**Média**	0.6378	0.6433	0.6421	0.6429	**0.6448**
-	**Ranking médio**	3.05	3.05	3.10	2.95	**2.85**
-	**Quantidade vitórias**	**5**	4	3	4	4

Tabela 10 – Resultados da avaliação do parâmetro θ para a estratégia *Variable Randomized Uncertainty* com o classificador RL.

Fonte: próprio autor (2020).

#	**Base**	θ **= 0.5**	θ **= 0.6**	θ **= 0.7**	θ **= 0.8**	θ **= 0.9**
1.	Bolsonaro_facebook	0.7240	0.7978	0.8042	**0.8686**	0.8482
2.	Bolsonaro_twitter	0.6035	0.7153	0.7524	0.7534	**0.7874**
3.	Haddad_facebook	0.6916	0.6998	0.7080	0.8109	**0.8835**
4.	Haddad_twitter	0.6921	0.7344	0.7054	**0.7581**	0.6986
5.	Ambos_facebook	0.8224	0.8809	0.8634	**0.8830**	0.8704
6.	Ambos_twitter	0.7562	0.7212	0.7623	**0.7834**	0.7422
7.	Sentiment140_test	0.6249	0.6471	0.6274	0.6479	**0.6609**
8.	Sentiment140_10000_1	0.7128	0.7164	0.7175	0.7161	**0.7209**
9.	Sentiment140_5000_1	0.6881	0.6815	0.6950	0.6951	**0.7029**
10.	Sentiment140_2500_1	0.7073	0.7093	0.7285	**0.7401**	0.7348
11.	Sentiment140_1000_1	0.5925	0.5980	0.6484	0.6782	**0.6824**
12.	Sentiment140_10000_2	0.7181	0.7267	0.7323	**0.7382**	0.7363
13.	Sentiment140_5000_2	0.6939	0.7045	0.7079	0.7069	**0.7173**
14.	Sentiment140_2500_2	0.6841	**0.6904**	0.6895	0.6896	0.6788
15.	Sentiment140_1000_2	0.3679	0.3918	0.5277	0.5723	**0.6064**
16.	Sanders_apple	0.4499	0.4949	0.4918	0.5001	**0.5193**
17.	Sanders_google	**0.4365**	0.4354	0.4078	0.4134	0.4109
18.	Sanders_microsoft	0.2999	0.2999	0.2999	0.3141	**0.3674**
19.	Sanders_twitter	**0.2884**	**0.2884**	**0.2884**	**0.2884**	**0.2884**
20.	Sanders_all	0.3989	0.4099	0.3978	0.4067	**0.4122**
-	**Média**	0.5976	0.6172	0.6278	0.6482	**0.6533**
-	**Ranking médio**	4.25	3.20	3.10	**1.90**	**1.90**
-	**Quantidade vitórias**	2	2	1	7	**12**

Tabela 11 – Resultados da avaliação do parâmetro θ para a estratégia *Entropy* com o classificador MNB.

Fonte: próprio autor (2020).

#	**Base**	θ **= 0.5**	θ **= 0.6**	θ **= 0.7**	θ **= 0.8**	θ **= 0.9**
1.	Bolsonaro_facebook	**0.8919**	0.8910	0.8861	0.8910	0.8850
2.	Bolsonaro_twitter	0.7158	0.7155	0.7300	0.7270	**0.7382**
3.	Haddad_facebook	**0.8973**	0.8877	0.8876	0.8909	0.8907
4.	Haddad_twitter	0.6429	0.6790	0.6778	0.6971	**0.7196**
5.	Ambos_facebook	0.8634	**0.8743**	0.8652	0.8652	0.8632
6.	Ambos_twitter	0.7114	0.7216	**0.7239**	0.7131	0.7121
7.	Sentiment140_test	0.6610	0.6886	**0.7097**	0.6889	0.7027
8.	Sentiment140_10000_1	0.7316	0.7314	0.7296	**0.7355**	0.7342
9.	Sentiment140_5000_1	0.6909	0.6923	0.7027	**0.7104**	0.7052
10.	Sentiment140_2500_1	0.7321	0.7333	0.7223	**0.7354**	0.7252
11.	Sentiment140_1000_1	0.6656	0.6734	0.6786	0.7079	**0.7089**
12.	Sentiment140_10000_2	0.7355	0.7379	**0.7457**	0.7370	0.7374
13.	Sentiment140_5000_2	0.7221	0.7241	**0.7271**	0.7257	0.7264
14.	Sentiment140_2500_2	0.6624	**0.6796**	0.6784	0.6759	0.6793
15.	Sentiment140_1000_2	0.6093	**0.6488**	0.6378	0.6289	0.6349
16.	Sanders_apple	0.5620	0.5576	**0.5710**	0.5699	0.5709
17.	Sanders_google	**0.4842**	0.4333	0.4372	0.4251	0.4251
18.	Sanders_microsoft	0.4436	0.4401	**0.4712**	0.4506	0.4606
19.	Sanders_twitter	0.3376	**0.3828**	0.3564	0.3601	0.3601
20.	Sanders_all	0.4402	0.4478	0.4429	**0.4627**	0.4604
-	**Média**	0.6600	0.6670	0.6691	0.6699	**0.6720**
-	**Ranking médio**	4.00	3.00	2.65	**2.55**	2.60
-	**Quantidade vitórias**	3	4	**6**	4	3

Tabela 12 – Resultados da avaliação do parâmetro θ para a estratégia *Entropy* com o classificador RL.

Fonte: próprio autor (2020).

#	**Base**	θ **= 0.5**	θ **= 0.6**	θ **= 0.7**	θ **= 0.8**	θ **= 0.9**
1.	Bolsonaro_facebook	**0.8121**	0.8056	0.7978	0.7885	0.7609
2.	Bolsonaro_twitter	0.7646	0.7638	0.7828	0.7828	**0.7860**
3.	Haddad_facebook	0.8908	0.8908	0.8576	0.9003	**0.9006**
4.	Haddad_twitter	0.7425	0.7457	0.7136	**0.7730**	0.7615
5.	Ambos_facebook	0.8815	0.8867	0.8970	**0.8976**	0.8965
6.	Ambos_twitter	**0.7543**	0.7278	0.7295	0.7382	0.7390
7.	Sentiment140_test	0.6096	**0.6556**	0.5957	0.6315	0.6162
8.	Sentiment140_10000_1	**0.7183**	0.7153	0.7041	0.7089	0.7106
9.	Sentiment140_5000_1	0.6829	0.6782	0.6881	**0.6907**	0.6890
10.	Sentiment140_2500_1	0.7181	0.7072	0.6976	0.7044	**0.7191**
11.	Sentiment140_1000_1	0.6025	0.6541	0.6332	**0.6554**	0.6408
12.	Sentiment140_10000_2	0.7170	**0.7209**	0.7208	0.7192	0.7185
13.	Sentiment140_5000_2	0.6961	**0.7104**	0.7061	0.7051	0.7029
14.	Sentiment140_2500_2	0.6856	0.6850	**0.6893**	0.6849	0.6870
15.	Sentiment140_1000_2	0.5658	0.5869	**0.6110**	0.6038	0.6064
16.	Sanders_apple	0.4971	0.4611	0.4729	**0.5105**	0.4800
17.	Sanders_google	0.4279	0.4306	0.4361	**0.4407**	0.4198
18.	Sanders_microsoft	0.4128	0.4267	0.4155	**0.4409**	0.4131
19.	Sanders_twitter	0.2864	0.2854	0.2874	**0.3138**	0.2874
20.	Sanders_all	**0.4237**	0.4205	0.4059	0.4059	0.4200
-	**Média**	0.6445	0.6479	0.6421	**0.6548**	0.6478
-	**Ranking médio**	3.35	3.15	3.20	**2.35**	2.80
-	**Quantidade vitórias**	4	3	2	**8**	3

Tabela 13 – Resultados da avaliação do parâmetro θ para a estratégia *Variable Entropy* com o classificador MNB.

Fonte: próprio autor (2020).

#	**Base**	θ **= 0.5**	θ **= 0.6**	θ **= 0.7**	θ **= 0.8**	θ **= 0.9**
1.	Bolsonaro_facebook	0.8968	**0.9067**	**0.9067**	0.8939	0.8910
2.	Bolsonaro_twitter	0.7229	0.7337	**0.7374**	**0.7374**	**0.7374**
3.	Haddad_facebook	0.8844	**0.8941**	0.8843	0.8843	**0.8941**
4.	Haddad_twitter	0.6636	0.6675	0.6713	**0.6843**	**0.6843**
5.	Ambos_facebook	**0.8692**	0.8646	**0.8692**	**0.8692**	0.8666
6.	Ambos_twitter	0.7144	**0.7347**	0.7258	0.7252	0.7087
7.	Sentiment140_test	0.6560	0.6532	0.6534	0.6595	**0.6598**
8.	Sentiment140_10000_1	0.7275	0.7297	**0.7359**	0.7286	0.7300
9.	Sentiment140_5000_1	0.6843	0.6772	0.6826	0.6777	**0.6860**
10.	Sentiment140_2500_1	**0.7353**	0.7260	0.7260	0.7276	0.7260
11.	Sentiment140_1000_1	0.6488	**0.6587**	0.6368	0.6467	0.6537
12.	Sentiment140_10000_2	0.7393	0.7354	**0.7394**	0.7384	0.7366
13.	Sentiment140_5000_2	**0.7230**	0.7181	0.7169	0.7182	0.7228
14.	Sentiment140_2500_2	0.6670	**0.6783**	0.6734	0.6633	0.6636
15.	Sentiment140_1000_2	0.6151	**0.6187**	0.6012	0.6047	0.5895
16.	Sanders_apple	0.5151	**0.5275**	0.5270	0.5167	0.5273
17.	Sanders_google	0.4501	**0.5172**	0.4997	0.4624	0.4452
18.	Sanders_microsoft	0.4391	**0.4447**	0.4364	0.4259	0.4285
19.	Sanders_twitter	0.3828	**0.4046**	**0.4046**	0.3564	**0.4046**
20.	Sanders_all	**0.4413**	0.4340	0.4212	0.4266	0.4296
-	**Média**	0.6588	**0.6662**	0.6625	0.6574	0.6593
-	**Ranking médio**	2.95	**2.50**	2.70	3.25	2.80
-	**Quantidade vitórias**	4	**10**	6	3	6

Tabela 14 – Resultados da avaliação do parâmetro θ para a estratégia *Variable Entropy* com o classificador RL.

Fonte: próprio autor (2020).

#	**Base**	θ **= 0.5**	θ **= 0.6**	θ **= 0.7**	θ **= 0.8**	θ **= 0.9**
1.	Bolsonaro_facebook	**0.8592**	0.7846	0.8411	0.8017	0.8312
2.	Bolsonaro_twitter	0.7433	0.7374	0.7208	**0.7629**	0.7441
3.	Haddad_facebook	0.8745	0.8689	0.8778	0.8813	**0.8847**
4.	Haddad_twitter	**0.7279**	0.6268	0.6381	0.6418	0.6921
5.	Ambos_facebook	0.8855	**0.8906**	0.8626	0.8849	0.8760
6.	Ambos_twitter	0.7459	0.7509	0.7504	**0.7696**	0.7500
7.	Sentiment140_test	0.5641	0.4976	0.5448	**0.6097**	0.6095
8.	Sentiment140_10000_1	0.6969	0.7119	0.7156	0.7165	**0.7166**
9.	Sentiment140_5000_1	0.6623	0.6861	0.6727	**0.6893**	0.6831
10.	Sentiment140_2500_1	**0.7223**	0.7081	0.7207	0.6966	0.7080
11.	Sentiment140_1000_1	**0.6853**	0.6640	0.6012	0.6070	0.6368
12.	Sentiment140_10000_2	0.7223	0.7216	0.7218	**0.7272**	0.7231
13.	Sentiment140_5000_2	0.7028	0.7081	0.7019	0.7013	**0.7108**
14.	Sentiment140_2500_2	**0.6903**	0.6854	0.6757	0.6561	0.6660
15.	Sentiment140_1000_2	0.5703	**0.5997**	0.5596	0.5824	0.5967
16.	Sanders_apple	**0.4804**	0.4704	0.4765	0.4773	0.4645
17.	Sanders_google	0.4133	0.4413	0.3920	0.4221	**0.4795**
18.	Sanders_microsoft	0.3720	0.3635	0.3637	**0.4006**	0.3349
19.	Sanders_twitter	0.3128	0.2864	0.2854	**0.3138**	0.3118
20.	Sanders_all	0.3997	0.4006	0.4072	0.3875	**0.4303**
-	**Média**	0.6416	0.6302	0.6265	0.6365	**0.6425**
-	**Ranking médio**	2.75	3.25	3.70	**2.65**	**2.65**
-	**Quantidade vitórias**	6	2	0	**7**	5

Tabela 15 – Resultados da avaliação do parâmetro θ para a estratégia *Variable Randomized Entropy* com o classificador MNB.

Fonte: próprio autor (2020).

#	**Base**	θ **= 0.5**	θ **= 0.6**	θ **= 0.7**	θ **= 0.8**	θ **= 0.9**
1.	Bolsonaro_facebook	0.8687	0.8789	0.8879	**0.8939**	0.8792
2.	Bolsonaro_twitter	0.7097	**0.7575**	0.7205	0.6884	0.7007
3.	Haddad_facebook	0.8969	0.8814	0.8679	**0.9002**	0.8872
4.	Haddad_twitter	0.6383	0.5688	0.6481	**0.7178**	0.6881
5.	Ambos_facebook	0.8741	**0.8808**	0.8450	0.8664	0.8644
6.	Ambos_twitter	0.7072	0.6954	0.6933	**0.7370**	0.6849
7.	Sentiment140_test	**0.6625**	0.6427	0.6050	0.6419	0.6494
8.	Sentiment140_10000_1	0.7225	**0.7349**	0.7225	0.7221	0.7224
9.	Sentiment140_5000_1	0.6859	0.6823	**0.7026**	0.6977	0.6881
10.	Sentiment140_2500_1	0.7064	0.7142	0.7145	0.7089	**0.7279**
11.	Sentiment140_1000_1	0.6486	0.6455	**0.6824**	0.6515	0.6666
12.	Sentiment140_10000_2	0.7260	**0.7305**	0.7291	0.7284	0.7212
13.	Sentiment140_5000_2	0.7101	0.7109	0.7121	**0.7184**	0.7072
14.	Sentiment140_2500_2	**0.6792**	0.6539	0.6469	0.6553	0.6636
15.	Sentiment140_1000_2	0.5522	0.6181	0.6187	0.5887	**0.6349**
16.	Sanders_apple	0.5058	0.5011	0.5090	**0.5441**	0.5338
17.	Sanders_google	0.4524	**0.4715**	0.4414	0.4445	0.4397
18.	Sanders_microsoft	0.4055	0.3755	0.3861	**0.4393**	0.4134
19.	Sanders_twitter	0.3634	0.3365	**0.4503**	0.3554	0.3110
20.	Sanders_all	0.4285	0.4305	0.4391	0.4386	**0.4517**
-	**Média**	0.6472	0.6456	0.6511	**0.6569**	0.6518
-	**Ranking médio**	3.25	3.25	2.85	**2.60**	3.05
-	**Quantidade vitórias**	2	5	3	**7**	3

Tabela 16 – Resultados da avaliação do parâmetro θ para a estratégia *Variable Randomized Entropy* com o classificador RL.

APÊNDICE B – RESULTADOS DO CENÁRIO I

Fonte: próprio autor (2020).

#	Base	RS	UN	VU	VRU	IG	EN	VE	VRE
1.	Bolsonaro_facebook	0.7534	0.7276	0.8003	**0.8576**	0.5849	0.8482	0.7609	0.8312
2.	Bolsonaro_twitter	0.7538	0.6937	0.7791	**0.7881**	0.3564	0.7874	0.7860	0.7441
3.	Haddad_facebook	0.8944	0.7080	**0.9103**	0.8913	0.7377	0.8835	0.9006	0.8847
4.	Haddad_twitter	0.6587	0.7358	0.7216	0.6123	0.5058	0.6986	**0.7615**	0.6921
5.	Ambos_facebook	0.8780	0.8663	0.8954	0.8612	0.6228	0.8704	**0.8965**	0.8760
6.	Ambos_twitter	0.7523	0.7440	0.7381	**0.7630**	0.7246	0.7422	0.7390	0.7500
7.	Sentiment140_test	0.5982	0.6419	0.6134	0.5852	0.3575	**0.6609**	0.6162	0.6095
8.	Sentiment140_10000_1	0.7066	0.7164	0.7123	0.7081	0.6737	**0.7209**	0.7106	0.7166
9.	Sentiment140_5000_1	0.6709	0.6931	0.6878	0.6695	0.6276	**0.7029**	0.6890	0.6831
10.	Sentiment140_2500_1	0.6937	0.7271	0.7174	0.7021	0.6280	**0.7348**	0.7191	0.7080
11.	Sentiment140_1000_1	0.6275	0.6382	0.6310	0.6332	0.6165	**0.6824**	0.6408	0.6368
12.	Sentiment140_10000_2	0.7169	0.7351	0.7259	0.7126	0.6861	**0.7363**	0.7185	0.7231
13.	Sentiment140_5000_2	0.7081	0.7111	0.6955	0.7140	0.6686	**0.7173**	0.7029	0.7108
14.	Sentiment140_2500_2	0.6603	0.6740	0.6852	0.6758	0.6470	0.6788	**0.6870**	0.6660
15.	Sentiment140_1000_2	0.6146	0.5023	0.5515	**0.6210**	0.4273	0.6037	0.6064	0.5967
16.	Sanders_apple	0.4581	0.4844	0.4710	0.4916	0.3680	**0.5193**	0.4800	0.4645
17.	Sanders_google	0.3927	0.4316	0.4164	0.4239	0.4487	0.4109	0.4198	**0.4795**
18.	Sanders_microsoft	0.3480	0.2999	0.4127	0.3890	0.3278	0.3674	**0.4131**	0.3349
19.	Sanders_twitter	0.2874	0.2884	0.2864	0.2864	**0.3118**	0.2884	0.2874	**0.3118**
20.	Sanders_all	0.3897	0.4064	0.4035	0.4028	0.3505	0.4122	0.4200	**0.4303**
-	**Média**	0.6282	0.6213	0.6427	0.6397	0.5336	**0.6533**	0.6478	0.6425
-	**Ranking médio**	5.50	4.25	4.35	4.30	7.25	**2.80**	3.30	4.05
-	**Quantidade vitórias**	0	0	1	4	1	**8**	4	3

Tabela 17 – Resultados do Cenário I com o classificador MNB.

Fonte: próprio autor (2020).

#	Base	RS	UN	VU	VRU	IG	EN	VE	VRE
1.	Bolsonaro_facebook	0.8544	0.8861	**0.8939**	0.8459	0.8235	0.8850	0.8910	0.8792
2.	Bolsonaro_twitter	**0.7574**	0.7226	0.7229	0.7002	0.5442	0.7382	0.7374	0.7007
3.	Haddad_facebook	0.8737	**0.8941**	0.8843	0.8901	0.8123	0.8907	**0.8941**	0.8872
4.	Haddad_twitter	0.6798	0.6882	0.6829	0.6400	0.5045	**0.7196**	0.6843	0.6881
5.	Ambos_facebook	0.8678	0.8723	0.8698	**0.8743**	0.7250	0.8632	0.8666	0.8644
6.	Ambos_twitter	0.6741	0.7196	0.6973	**0.7363**	0.6467	0.7121	0.7087	0.6849
7.	Sentiment140_test	0.6771	0.6886	0.6035	0.6426	0.4063	**0.7027**	0.6598	0.6494
8.	Sentiment140_10000_1	0.7212	0.7286	0.7267	0.7191	0.7097	**0.7342**	0.7300	0.7224
9.	Sentiment140_5000_1	0.6801	0.6959	0.6794	0.6752	0.6606	**0.7052**	0.6860	0.6881
10.	Sentiment140_2500_1	0.7276	**0.7289**	0.7284	0.7182	0.6875	0.7252	0.7260	0.7279
11.	Sentiment140_1000_1	0.6805	0.6704	0.6374	0.6425	0.6052	**0.7089**	0.6537	0.6666
12.	Sentiment140_10000_2	0.7212	**0.7444**	0.7387	0.7122	0.7132	0.7374	0.7366	0.7212
13.	Sentiment140_5000_2	0.7076	0.7255	0.7203	0.7128	0.6822	**0.7264**	0.7228	0.7072
14.	Sentiment140_2500_2	0.6539	**0.6813**	0.6700	0.6637	0.6711	0.6793	0.6636	0.6636
15.	Sentiment140_1000_2	0.6253	**0.6557**	0.6192	0.6292	0.5650	0.6349	0.5895	0.6349
16.	Sanders_apple	0.4530	0.5702	0.5274	0.5141	0.3987	**0.5709**	0.5273	0.5338
17.	Sanders_google	0.4721	**0.4844**	0.4546	0.4322	0.4725	0.4251	0.4452	0.4397
18.	Sanders_microsoft	0.3960	0.4316	0.4388	0.3792	0.3399	**0.4606**	0.4285	0.4134
19.	Sanders_twitter	0.3597	0.3809	0.3859	0.3365	0.3148	0.3601	**0.4046**	0.3110
20.	Sanders_all	0.4491	0.4450	0.4365	0.4316	0.3903	**0.4604**	0.4296	0.4517
-	**Média**	0.6516	0.6707	0.6559	0.6448	0.5837	**0.6720**	0.6593	0.6518
-	**Ranking médio**	4.95	**2.20**	4.15	5.70	7.35	2.65	4.05	4.85
-	**Quantidade vitórias**	1	6	1	2	0	**9**	2	0

Tabela 18 – Resultados do Cenário I com o classificador RL.

Fonte: próprio autor (2020).

		MNB		RL	
#	**Base**	**Sem AL**	**Melhor AL**	**Sem AL**	**Melhor AL**
1.	Bolsonaro_facebook	**0.8896**	0.8576	0.8908	**0.8939**
2.	Bolsonaro_twitter	0.7766	**0.7881**	0.7345	**0.7574**
3.	Haddad_facebook	0.8913	**0.9103**	**0.8973**	0.8941
4.	Haddad_twitter	**0.7646**	0.7615	0.7079	**0.7196**
5.	Ambos_facebook	0.8752	**0.8965**	**0.8776**	0.8743
6.	Ambos_twitter	**0.7875**	0.7630	0.7235	**0.7363**
7.	Sentiment140_test	0.6490	**0.6609**	0.7025	**0.7027**
8.	Sentiment140_10000_1	0.7208	**0.7209**	0.7310	**0.7342**
9.	Sentiment140_5000_1	0.6971	**0.7029**	**0.7069**	0.7052
10.	Sentiment140_2500_1	0.7172	**0.7348**	**0.7370**	0.7289
11.	Sentiment140_1000_1	**0.7038**	0.6824	**0.7160**	0.7089
12.	Sentiment140_10000_2	0.7317	**0.7363**	0.7366	**0.7444**
13.	Sentiment140_5000_2	**0.7213**	0.7173	0.7182	**0.7264**
14.	Sentiment140_2500_2	0.6835	**0.6870**	**0.6862**	0.6813
15.	Sentiment140_1000_2	**0.6575**	0.6210	0.6349	**0.6557**
16.	Sanders_apple	0.5148	**0.5193**	0.5686	**0.5709**
17.	Sanders_google	0.4198	**0.4795**	0.4280	**0.4844**
18.	Sanders_microsoft	**0.4172**	0.4131	0.4602	**0.4606**
19.	Sanders_twitter	0.3061	**0.3118**	0.3601	**0.4046**
20.	Sanders_all	0.4102	**0.4303**	**0.4610**	0.4604
-	**Quantidade vitórias**	7	**13**	7	**13**

Tabela 19 – Comparação entre a *f-measure* das melhores estratégias do Cenário I e da *baseline* sem AL.

Fonte: próprio autor (2020).

#	**Base**	**#Disp**	**RS**	**UN**	**VU**	**VRU**	**IG**	**EN**	**VE**	**VRE**
1.	Bolsonaro_facebook	628	326	80	307	**307**	306	178	307	311
2.	Bolsonaro_twitter	517	263	109	253	**251**	184	254	257	262
3.	Haddad_facebook	627	311	124	**222**	224	359	203	225	232
4.	Haddad_twitter	457	236	90	308	317	129	189	**312**	317
5.	Ambos_facebook	1,144	590	238	564	560	441	441	**566**	569
6.	Ambos_twitter	826	415	356	408	**415**	223	535	414	417
7.	Sentiment140_test	298	142	187	153	148	79	**227**	180	166
8.	Sentiment140_10000_1	6,000	3,018	3,962	2,992	2,991	2,432	**5,164**	3,016	2,999
9.	Sentiment140_5000_1	3,000	1,527	1,812	1,496	1,496	1,112	**2,474**	1,515	1,510
10.	Sentiment140_2500_1	1,500	796	967	749	741	506	**1,288**	773	761
11.	Sentiment140_1000_1	600	286	369	306	306	190	**505**	329	326
12.	Sentiment140_10000_2	6,000	2,958	3,933	2,996	2,995	2,437	**5,092**	3,017	3,008
13.	Sentiment140_5000_2	3,000	1,530	1,939	1,499	1,497	979	**2,574**	1,518	1,519
14.	Sentiment140_2500_2	1,500	762	873	752	756	525	1,211	**773**	764
15.	Sentiment140_1000_2	600	303	101	302	**300**	225	325	323	319
16.	Sanders_apple	601	305	305	300	300	172	**441**	317	321
17.	Sanders_google	503	247	259	248	252	176	364	260	**260**
18.	Sanders_microsoft	518	246	25	253	251	227	106	**255**	261
19.	Sanders_twitter	431	224	5	210	209	**291**	43	210	**213**
20.	Sanders_all	2,054	1,009	761	1,019	1,017	560	1,190	1,022	**1,027**
-	**Média**	-	775	825	767	767	578	1,140	779	778

Tabela 20 – Quantidade de instâncias escolhidas pelas estratégias de seleção de AL no Cenário I com o classificador MNB.

Fonte: próprio autor (2020).

#	**Base**	**#Disp**	**RS**	**UN**	**VU**	**VRU**	**IG**	**EN**	**VE**	**VRE**
1.	Bolsonaro_facebook	628	328	244	**308**	316	320	418	312	317
2.	Bolsonaro_twitter	517	**265**	333	263	265	179	458	282	283
3.	Haddad_facebook	627	314	**307**	313	319	351	491	**327**	326
4.	Haddad_twitter	457	225	251	229	231	109	**391**	241	246
5.	Ambos_facebook	1,144	564	493	566	**563**	424	795	573	583
6.	Ambos_twitter	826	415	555	413	**407**	201	740	433	433
7.	Sentiment140_test	298	149	283	170	173	76	**296**	220	205
8.	Sentiment140_10000_1	6,000	3,029	4,326	2,991	2,985	2,178	**5,626**	3,012	3,013
9.	Sentiment140_5000_1	3,000	1,551	2,251	1,496	1,502	1,003	**2,827**	1,519	1,510
10.	Sentiment140_2500_1	1,500	757	**1,205**	753	758	433	1,444	778	779
11.	Sentiment140_1000_1	600	299	520	307	306	158	**595**	333	334
12.	Sentiment140_10000_2	6,000	2,998	**4,483**	2,996	2,995	2,228	5,583	3,019	3,015
13.	Sentiment140_5000_2	3,000	1,521	2,308	1,499	1,506	897	**2,863**	1,527	1,524
14.	Sentiment140_2500_2	1,500	764	**1,145**	755	759	483	1,372	778	774
15.	Sentiment140_1000_2	600	304	**520**	311	314	219	591	341	324
16.	Sanders_apple	601	296	527	311	313	153	**601**	355	355
17.	Sanders_google	503	258	**414**	254	258	142	501	288	288
18.	Sanders_microsoft	518	277	325	261	258	230	**501**	278	280
19.	Sanders_twitter	431	211	226	214	211	291	424	**236**	237
20.	Sanders_all	2,054	1,032	1,541	1,021	1,031	559	**2,027**	1,034	1,048
-	**Média**	-	778	1,113	772	774	532	1,427	794	794

Tabela 21 – Quantidade de instâncias escolhidas pelas estratégias de seleção de AL no Cenário I com o classificador RL.

APÊNDICE C – RESULTADOS DO CENÁRIO II

Fonte: próprio autor (2020).

#	Base	RS	UN	VU	VRU	IG	EN	VE	VRE
1.	Bolsonaro_facebook	**0.8682**	0.6159	0.8233	0.8470	0.5849	0.8251	0.7391	0.8312
2.	Bolsonaro_twitter	**0.7897**	0.3791	0.7489	0.7770	0.3758	0.5411	0.7438	0.7838
3.	Haddad_facebook	0.8593	0.6662	0.8739	**0.8911**	0.7299	0.7039	0.8564	0.8434
4.	Haddad_twitter	0.6033	0.7520	0.6829	0.6453	0.4999	0.6938	**0.7537**	0.6618
5.	Ambos_facebook	0.8553	0.7971	0.8419	0.8620	0.6107	0.8553	0.8633	**0.8698**
6.	Ambos_twitter	0.7352	0.7217	0.7255	0.7093	0.7054	**0.7595**	0.7431	0.7544
7.	Sentiment140_test	0.6110	0.5948	**0.6296**	0.5073	0.3430	0.6274	0.6076	0.5977
8.	Sentiment140_10000_1	0.7042	0.7120	0.7031	0.7150	0.6604	**0.7208**	0.7091	0.7048
9.	Sentiment140_5000_1	0.6863	0.6815	0.6929	0.6761	0.6307	**0.7063**	0.6861	0.6938
10.	Sentiment140_2500_1	0.7050	0.7094	0.6832	0.7030	0.6738	**0.7264**	0.6975	0.7006
11.	Sentiment140_1000_1	0.6130	0.5745	0.6083	**0.6527**	0.6350	0.6305	0.6039	0.6332
12.	Sentiment140_10000_2	0.7103	0.6912	0.7255	**0.7293**	0.6666	0.7185	0.7069	0.7146
13.	Sentiment140_5000_2	0.6860	0.7097	0.7015	0.6907	0.6695	**0.7215**	0.7013	0.7011
14.	Sentiment140_2500_2	0.6566	0.6924	0.6796	0.6754	0.6273	**0.6958**	0.6933	0.6640
15.	Sentiment140_1000_2	0.5741	0.3861	0.5199	**0.5952**	0.4294	0.4604	0.5073	0.5659
16.	Sanders_apple	0.4699	0.4425	0.4444	0.4582	0.3973	**0.4937**	0.4638	0.4562
17.	Sanders_google	0.4143	0.4301	**0.4429**	0.4223	0.4284	0.4131	0.4301	0.4159
18.	Sanders_microsoft	0.3221	0.2999	0.3653	0.3702	0.3278	0.3123	0.3890	**0.3958**
19.	Sanders_twitter	0.2854	0.2884	0.2854	0.2864	**0.3118**	0.2894	0.2874	0.2864
20.	Sanders_all	0.3991	0.4181	0.4055	0.4057	0.3478	**0.4271**	0.4083	0.4167
-	**Média**	0.6274	0.5781	0.6292	0.6310	0.5328	0.6161	0.6296	**0.6345**
-	**Ranking médio**	4.55	5.25	4.35	3.90	6.85	**3.20**	3.95	3.80
-	**Quantidade vitórias**	2	0	2	4	1	**8**	1	2

Tabela 22 – Resultados do Cenário II com o classificador MNB.

Fonte: próprio autor (2020).

#	Base	RS	UN	VU	VRU	IG	EN	VE	VRE
1.	Bolsonaro_facebook	0.8610	0.8839	0.8850	0.8482	0.8582	**0.8879**	0.8861	0.8529
2.	Bolsonaro_twitter	0.6650	0.7276	0.7286	0.7065	0.5422	**0.7345**	0.7134	0.7049
3.	Haddad_facebook	0.8843	0.8735	0.8663	0.8802	0.8123	**0.8877**	0.8866	0.8639
4.	Haddad_twitter	0.6197	0.6915	0.6940	**0.7293**	0.4709	0.7174	0.6814	0.6522
5.	Ambos_facebook	**0.8798**	0.8501	0.8456	0.8593	0.7238	0.8527	0.8363	0.8566
6.	Ambos_twitter	0.7019	**0.7272**	0.6998	0.6852	0.6646	0.7262	0.7026	0.7022
7.	Sentiment140_test	0.6165	0.6900	0.6479	0.6140	0.4113	**0.7027**	0.6628	0.6359
8.	Sentiment140_10000_1	0.7247	0.7302	0.7211	0.7180	0.7089	**0.7349**	0.7208	0.7194
9.	Sentiment140_5000_1	0.6885	0.6984	0.6778	0.6895	0.6623	**0.7052**	0.6725	0.6938
10.	Sentiment140_2500_1	0.7215	**0.7403**	0.7071	0.7207	0.6727	0.7309	0.7145	0.7077
11.	Sentiment140_1000_1	0.6497	0.6907	0.6575	0.6488	0.6325	**0.7120**	0.6486	0.6374
12.	Sentiment140_10000_2	0.7128	0.7387	0.7314	0.7234	0.6986	**0.7402**	0.7329	0.7264
13.	Sentiment140_5000_2	0.7102	0.7189	0.7101	0.6873	0.6782	**0.7299**	0.7129	0.7058
14.	Sentiment140_2500_2	0.6681	**0.6831**	0.6578	0.6649	0.6519	0.6830	0.6691	0.6615
15.	Sentiment140_1000_2	0.6067	**0.6636**	0.5804	0.6266	0.5627	0.6428	0.6081	0.5980
16.	Sanders_apple	0.5075	**0.5724**	0.5015	0.4985	0.3992	0.5709	0.5079	0.5654
17.	Sanders_google	0.4393	0.4363	0.4370	0.4351	0.4184	0.4251	0.4884	**0.5037**
18.	Sanders_microsoft	0.4272	**0.4606**	0.4298	0.3856	0.3526	**0.4606**	0.4298	0.4491
19.	Sanders_twitter	0.3383	0.3110	0.3129	**0.3823**	0.3169	0.3601	0.3554	0.3758
20.	Sanders_all	0.4425	0.4607	0.4132	0.4332	0.4153	**0.4621**	0.4103	0.4257
-	**Média**	0.6433	0.6674	0.6453	0.6468	0.5827	**0.6733**	0.6520	0.6519
-	**Ranking médio**	4.55	2.70	5.05	5.00	7.70	**1.85**	4.20	4.85
-	**Quantidade vitórias**	1	6	0	2	0	**11**	0	1

Tabela 23 – Resultados do Cenário II com o classificador RL.

Fonte: próprio autor (2020).

#	**Base**	**RS**	**UN**	**VU**	**VRU**	**IG**	**EN**	**VE**	**VRE**
1.	Bolsonaro_facebook	0.8072	0.8513	0.8490	0.8562	0.8251	0.8636	0.8648	**0.8784**
2.	Bolsonaro_twitter	0.7117	0.7049	**0.7436**	0.6510	0.5968	0.7307	0.7015	0.7155
3.	Haddad_facebook	0.8713	0.8655	0.8717	0.8742	0.8255	**0.8877**	0.8815	0.8813
4.	Haddad_twitter	0.6473	0.6940	0.6784	0.6264	0.6113	**0.7008**	0.6959	0.6660
5.	Ambos_facebook	0.8167	0.8264	0.8193	**0.8392**	0.7740	0.8271	0.8234	0.8098
6.	Ambos_twitter	0.6957	0.7168	0.7168	0.6967	0.6078	**0.7350**	0.7091	0.6879
7.	Sentiment140_test	0.6046	0.6678	0.5448	0.6229	0.4715	0.6678	0.6333	**0.6938**
10.	Sentiment140_2500_1	0.6998	0.7180	**0.7245**	0.6874	0.6689	0.7180	0.7204	0.6835
11.	Sentiment140_1000_1	0.6805	**0.6875**	0.6569	0.6701	0.6229	**0.6875**	0.6654	0.6794
15.	Sentiment140_1000_2	**0.6437**	0.6297	0.6037	0.6433	0.5815	0.6199	0.5952	0.6131
16.	Sanders_apple	0.5218	**0.5918**	0.5103	0.5309	0.4795	**0.5918**	0.5520	0.5455
17.	Sanders_google	0.4780	**0.4974**	0.4116	0.4480	0.4403	0.4538	0.4864	0.4800
18.	Sanders_microsoft	0.3408	**0.4840**	0.4067	0.4189	0.3861	**0.4840**	0.4326	0.4398
19.	Sanders_twitter	0.3327	0.4650	0.4394	**0.5140**	0.4683	0.4983	0.4150	0.4806
-	**Média**	0.6323	0.6714	0.6412	0.6485	0.5971	**0.6761**	0.6555	0.6610
-	**Ranking médio**	5.50	3.07	5.07	4.50	7.50	**2.07**	3.92	4.00
-	**Quantidade vitórias**	1	4	2	2	0	**6**	0	2

Tabela 24 – Resultados do Cenário II com o classificador SVM.

Fonte: próprio autor (2020).

#	**Base**	**#Disp**	**RS**	**UN**	**VU**	**VRU**	**IG**	**EN**	**VE**	**VRE**
1.	Bolsonaro_facebook	628	**299**	20	307	314	301	81	307	310
2.	Bolsonaro_twitter	517	**278**	27	251	252	185	71	251	256
3.	Haddad_facebook	627	291	48	307	**309**	351	89	308	302
4.	Haddad_twitter	457	224	107	223	226	127	170	**223**	234
5.	Ambos_facebook	1,144	561	240	564	565	432	406	564	**571**
6.	Ambos_twitter	826	402	376	411	415	217	**569**	422	**419**
7.	Sentiment140_test	298	150	122	**146**	142	75	**195**	154	158
8.	Sentiment140_10000_1	6,000	3,023	3,634	2,990	2,994	2,282	**4,959**	3,011	3,008
9.	Sentiment140_5000_1	3,000	1,508	1,442	1,490	1,489	1,074	**2,123**	1,498	1,502
10.	Sentiment140_2500_1	1,500	765	977	750	750	446	**1,309**	770	764
11.	Sentiment140_1000_1	600	309	202	293	**296**	131	325	296	302
12.	Sentiment140_10000_2	6,000	2,983	3,073	2,986	**2,985**	2,408	4,446	2,997	3,000
13.	Sentiment140_5000_2	3,000	1,514	1,764	1,494	1,492	922	**2,427**	1,514	1,518
14.	Sentiment140_2500_2	1,500	720	603	744	748	507	**931**	755	759
15.	Sentiment140_1000_2	600	294	44	293	**291**	222	119	294	296
16.	Sanders_apple	601	279	353	304	306	157	**493**	325	323
17.	Sanders_google	503	259	278	**253**	256	114	406	272	269
18.	Sanders_microsoft	518	274	21	253	255	229	78	254	**262**
19.	Sanders_twitter	431	204	6	210	209	291	**48**	210	207
20.	Sanders_all	2,054	1,026	1,276	1,033	1,030	**527**	1,695	1,061	1,051
-	**Média**	-	718	731	765	766	560	1,047	774	776

Tabela 25 – Quantidade de instâncias escolhidas pelas estratégias de seleção de AL no Cenário II com o classificador MNB.

Fonte: próprio autor (2020).

#	**Base**	**#Disp**	**RS**	**UN**	**VU**	**VRU**	**IG**	**EN**	**VE**	**VRE**
1.	Bolsonaro_facebook	628	287	352	316	318	293	**545**	335	333
2.	Bolsonaro_twitter	517	270	352	265	257	179	**492**	289	283
3.	Haddad_facebook	627	314	341	316	316	359	**509**	**334**	327
4.	Haddad_twitter	457	226	326	226	**231**	112	431	249	253
5.	Ambos_facebook	1,144	**587**	671	569	576	422	931	588	592
6.	Ambos_twitter	826	420	**672**	415	417	192	**783**	437	437
7.	Sentiment140_test	298	142	291	169	161	78	**298**	235	210
8.	Sentiment140_10000_1	6,000	3,034	4,911	2,994	2,983	2,086	**5,824**	3,021	3,013
9.	Sentiment140_5000_1	3,000	1,445	2,478	1,500	1,501	991	**2,940**	1,526	1,516
10.	Sentiment140_2500_1	1,500	765	**1,290**	758	761	411	1,473	787	778
11.	Sentiment140_1000_1	600	289	535	309	314	152	**591**	338	333
12.	Sentiment140_10000_2	6,000	2,987	4,998	2,999	3,001	2,190	**5,746**	3,025	3,019
13.	Sentiment140_5000_2	3,000	1,503	2,564	1,501	1,501	837	**2,913**	1,529	1,531
14.	Sentiment140_2500_2	1,500	736	1,220	756	754	467	**1,404**	783	776
15.	Sentiment140_1000_2	600	293	**462**	302	300	223	585	325	321
16.	Sanders_apple	601	301	587	317	322	149	601	378	**379**
17.	Sanders_google	503	246	489	267	257	126	503	329	**315**
18.	Sanders_microsoft	518	259	**429**	264	269	230	**513**	301	303
19.	Sanders_twitter	431	208	334	216	**220**	**291**	430	244	245
20.	Sanders_all	2,054	1,024	1,991	**1,043**	1,047	523	2,054	1,148	1,142
-	**Média**	-	767	1,265	775	775	516	1,478	810	805

Tabela 26 – Quantidade de instâncias escolhidas pelas estratégias de seleção de AL no Cenário II com o classificador RL.

Fonte: próprio autor (2020).

#	**Base**	**#Disp**	**RS**	**UN**	**VU**	**VRU**	**IG**	**EN**	**VE**	**VRE**
1.	Bolsonaro_facebook	628	308	397	319	316	285	607	340	**335**
2.	Bolsonaro_twitter	517	250	**443**	266	271	186	515	292	290
3.	Haddad_facebook	627	303	314	315	321	329	**556**	333	331
4.	Haddad_twitter	457	218	370	237	240	106	**447**	264	256
5.	Ambos_facebook	1,144	597	592	569	**575**	393	913	588	581
6.	Ambos_twitter	826	440	790	421	426	195	**823**	451	437
7.	Sentiment140_test	298	155	298	163	162	79	298	232	**206**
10.	Sentiment140_2500_1	1,500	748	1,496	**767**	758	412	1,499	796	801
11.	Sentiment140_1000_1	600	295	**599**	315	315	149	**599**	344	338
15.	Sentiment140_1000_2	600	**303**	566	304	303	219	599	327	327
16.	Sanders_apple	601	282	**601**	328	332	126	**601**	457	436
17.	Sanders_google	503	237	**482**	266	272	123	503	353	312
18.	Sanders_microsoft	518	247	**489**	265	261	227	**518**	306	308
19.	Sanders_twitter	431	205	377	220	**226**	284	428	245	242
-	**Média**	-	327	558	339	341	222	636	380	371

Tabela 27 – Quantidade de instâncias escolhidas pelas estratégias de seleção de AL no Cenário II com o classificador SVM.

APÊNDICE D – RESULTADOS DO CENÁRIO III

Fonte: próprio autor (2020).

#	Base	Quant	RS	UN	IG	EN
1.	Bolsonaro_facebook	244	0.7951	**0.8660**	0.8567	**0.8660**
2.	Bolsonaro_twitter	201	**0.6972**	0.6508	0.5220	0.6508
3.	Haddad_facebook	244	**0.8087**	0.7996	0.7863	0.7996
4.	Haddad_twitter	178	0.5884	**0.7138**	0.5458	**0.7138**
5.	Ambos_facebook	445	**0.8734**	0.8624	0.7295	0.8624
6.	Ambos_twitter	321	0.7323	0.7298	**0.7399**	0.7298
7.	Sentiment140_test	116	0.4827	0.5556	0.5003	**0.5786**
8.	Sentiment140_10000_1	2,333	0.6937	**0.6976**	0.6707	**0.6976**
9.	Sentiment140_5000_1	1,166	0.6636	**0.6763**	0.6320	**0.6763**
10.	Sentiment140_2500_1	583	**0.7049**	0.6519	0.6970	0.6519
11.	Sentiment140_1000_1	233	0.5758	0.6018	**0.6031**	0.6018
12.	Sentiment140_10000_2	2,333	**0.7084**	0.6649	0.7025	0.6649
13.	Sentiment140_5000_2	1,166	0.6958	**0.7040**	0.6713	**0.7040**
14.	Sentiment140_2500_2	583	**0.6703**	0.6639	0.6463	0.6639
15.	Sentiment140_1000_2	233	0.5110	**0.5882**	0.5134	**0.5882**
16.	Sanders_apple	234	**0.4592**	0.4200	0.4354	0.4211
17.	Sanders_google	195	0.4090	0.3957	**0.4393**	0.4010
18.	Sanders_microsoft	201	0.3493	**0.3958**	0.3290	0.3904
19.	Sanders_twitter	168	0.2864	**0.2874**	**0.2874**	**0.2874**
20.	Sanders_all	799	0.3854	**0.4075**	0.4063	0.3904
-	**Média**	-	0.6045	0.6167	0.5857	**0.6170**
-	**Ranking médio**	-	2.50	1.95	2.85	**1.90**
-	**Quantidade vitórias**	-	7	**9**	4	8

Tabela 28 – Resultados do Cenário III com o classificador MNB e porcentagem de 33% dos dados.

Fonte: próprio autor (2020).

#	Base	Quant	RS	UN	IG	EN
1.	Bolsonaro_facebook	244	0.8567	**0.8822**	0.8487	**0.8822**
2.	Bolsonaro_twitter	201	0.6741	**0.7226**	0.6037	**0.7226**
3.	Haddad_facebook	244	0.8384	**0.8770**	0.8676	**0.8770**
4.	Haddad_twitter	178	0.6346	**0.6683**	0.5867	**0.6683**
5.	Ambos_facebook	445	**0.8714**	0.8407	0.7807	0.8407
6.	Ambos_twitter	321	**0.6778**	0.6765	0.6464	0.6765
7.	Sentiment140_test	116	0.4690	**0.6130**	0.5687	0.5595
8.	Sentiment140_10000_1	2,333	0.7003	**0.7276**	0.7079	**0.7276**
9.	Sentiment140_5000_1	1,166	0.6681	**0.6810**	0.6568	**0.6810**
10.	Sentiment140_2500_1	583	0.7016	**0.7041**	0.6928	**0.7041**
11.	Sentiment140_1000_1	233	0.6010	**0.6494**	0.6099	**0.6494**
12.	Sentiment140_10000_2	2,333	0.7073	**0.7261**	0.7048	**0.7261**
13.	Sentiment140_5000_2	1,166	0.6706	**0.7074**	0.6956	**0.7074**
14.	Sentiment140_2500_2	583	0.6603	0.6505	**0.6697**	0.6505
15.	Sentiment140_1000_2	233	0.5605	0.5813	**0.5942**	0.5813
16.	Sanders_apple	234	0.4797	0.4563	0.4646	**0.4848**
17.	Sanders_google	195	0.4274	0.4295	0.4062	**0.4427**
18.	Sanders_microsoft	201	0.3763	0.3753	0.3694	**0.3789**
19.	Sanders_twitter	168	0.3376	0.3129	0.2874	**0.3597**
20.	Sanders_all	799	0.4145	0.3998	**0.4208**	0.4197
-	**Média**	-	0.6164	0.6342	0.6091	**0.6370**
-	**Ranking médio**	-	2.90	1.80	3.20	**1.40**
-	**Quantidade vitórias**	-	2	11	3	**14**

Tabela 29 – Resultados do Cenário III com o classificador RL e porcentagem de 33% dos dados.

Fonte: próprio autor (2020).

#	Base	Quant	RS	UN	IG	EN
1.	Bolsonaro_facebook	244	0.8304	0.8586	**0.8698**	0.8635
2.	Bolsonaro_twitter	201	0.6522	0.6859	0.6063	**0.7010**
3.	Haddad_facebook	244	0.8742	**0.8781**	0.8618	**0.8781**
4.	Haddad_twitter	178	0.6866	0.6920	0.6818	**0.6999**
5.	Ambos_facebook	445	0.8058	0.8101	0.7893	**0.8180**
6.	Ambos_twitter	321	0.6631	0.6605	0.6512	**0.7061**
7.	Sentiment140_test	116	0.5842	0.5655	**0.6367**	0.5158
8.	Sentiment140_10000_1	2,333	0.7054	0.7078	0.6970	**0.7091**
9.	Sentiment140_5000_1	1,166	**0.6832**	0.6696	0.6655	0.6689
10.	Sentiment140_2500_1	583	0.6931	0.7148	0.7010	**0.7213**
11.	Sentiment140_1000_1	233	0.5903	0.6274	**0.6388**	0.6274
12.	Sentiment140_10000_2	2,333	0.6859	0.6991	0.6892	**0.7013**
13.	Sentiment140_5000_2	1,166	0.6876	**0.6977**	0.6759	**0.6977**
14.	Sentiment140_2500_2	583	0.6559	**0.6601**	0.6586	0.6570
15.	Sentiment140_1000_2	233	**0.6260**	0.5712	0.5925	0.5712
16.	Sanders_apple	234	0.4807	0.4892	0.4995	**0.5017**
17.	Sanders_google	195	0.4367	**0.4710**	0.4671	0.4088
18.	Sanders_microsoft	201	0.3976	0.3740	0.3964	**0.4372**
19.	Sanders_twitter	168	0.4002	0.3927	0.3419	**0.4180**
20.	Sanders_all	799	**0.4474**	0.4282	0.3959	0.4357
-	**Média**	-	0.6293	0.6327	0.6260	**0.6369**
-	**Ranking médio**	-	2.80	2.25	3.00	**1.75**
-	**Quantidade vitórias**	-	3	4	3	12

Tabela 30 – Resultados do Cenário III com o classificador SVM e porcentagem de 33% dos dados.

Fonte: próprio autor (2020).

#	Base	Quant	RS	UN	IG	EN
1.	Bolsonaro_facebook	366	0.8465	0.8596	**0.8976**	0.8596
2.	Bolsonaro_twitter	301	0.7207	**0.7415**	0.5882	**0.7415**
3.	Haddad_facebook	366	**0.8746**	0.8459	0.8200	0.8459
4.	Haddad_twitter	267	0.6118	**0.7958**	0.6556	**0.7958**
5.	Ambos_facebook	667	0.8749	**0.8849**	0.8152	**0.8849**
6.	Ambos_twitter	482	0.7390	**0.7630**	0.7534	**0.7630**
7.	Sentiment140_test	174	0.5399	**0.6092**	0.5429	**0.6092**
8.	Sentiment140_10000_1	3,500	0.7041	**0.7073**	0.7024	**0.7073**
9.	Sentiment140_5000_1	1,750	0.6612	**0.6763**	0.6640	**0.6763**
10.	Sentiment140_2500_1	875	0.7045	0.6790	**0.7108**	0.6790
11.	Sentiment140_1000_1	350	**0.6563**	0.6161	0.6404	0.6161
12.	Sentiment140_10000_2	3,500	**0.7140**	0.6850	0.7113	0.6850
13.	Sentiment140_5000_2	1,750	**0.7037**	0.7023	0.6934	0.7023
14.	Sentiment140_2500_2	875	0.6634	**0.6653**	0.6625	**0.6653**
15.	Sentiment140_1000_2	350	0.5796	**0.6205**	0.5473	**0.6205**
16.	Sanders_apple	251	**0.4628**	0.4610	0.4495	0.4569
17.	Sanders_google	293	0.4304	0.4264	0.4327	**0.4373**
18.	Sanders_microsoft	302	0.3937	0.3789	0.3516	**0.3978**
19.	Sanders_twitter	252	0.3048	**0.3128**	0.3118	**0.3128**
20.	Sanders_all	1,198	0.4181	0.4176	0.4066	**0.4219**
-	**Média**	-	0.6304	0.6424	0.6179	**0.6439**
-	**Ranking médio**	-	2.65	1.85	3.15	**1.55**
-	**Quantidade vitórias**	-	5	10	2	**13**

Tabela 31 – Resultados do Cenário III com o classificador MNB e porcentagem de 50% dos dados.

Fonte: próprio autor (2020).

#	Base	Quant	RS	UN	IG	EN
1.	Bolsonaro_facebook	366	0.8529	**0.8762**	0.8585	**0.8762**
2.	Bolsonaro_twitter	301	**0.7399**	0.7102	0.6544	0.7102
3.	Haddad_facebook	366	0.8644	**0.8835**	0.8609	**0.8835**
4.	Haddad_twitter	267	0.6301	**0.6790**	0.6542	**0.6790**
5.	Ambos_facebook	667	**0.8708**	0.8557	0.8488	0.8557
6.	Ambos_twitter	482	**0.7030**	0.6977	0.6618	0.6977
7.	Sentiment140_test	174	0.5940	0.6204	0.5842	**0.6309**
8.	Sentiment140_10000_1	3,500	0.7231	**0.7248**	0.7202	**0.7248**
9.	Sentiment140_5000_1	1,750	**0.7016**	0.6740	0.6671	0.6740
10.	Sentiment140_2500_1	875	0.6984	**0.7235**	0.7032	**0.7235**
11.	Sentiment140_1000_1	350	0.6467	**0.6544**	0.6275	**0.6544**
12.	Sentiment140_10000_2	3,500	0.7176	**0.7329**	0.7242	**0.7329**
13.	Sentiment140_5000_2	1,750	0.7038	**0.7178**	0.7032	**0.7178**
14.	Sentiment140_2500_2	875	0.6542	**0.6722**	0.6682	**0.6722**
15.	Sentiment140_1000_2	350	**0.6009**	0.5852	0.5988	0.5852
16.	Sanders_apple	351	**0.5392**	0.5255	0.4840	0.5150
17.	Sanders_google	293	**0.5123**	0.4462	0.4113	0.4823
18.	Sanders_microsoft	302	0.3934	**0.4190**	0.3933	0.3884
19.	Sanders_twitter	252	0.3544	**0.3765**	0.3128	0.3755
20.	Sanders_all	1,198	0.4200	0.4098	0.4304	**0.4447**
-	**Média**	-	0.6460	0.6492	0.6284	**0.6512**
-	**Ranking médio**	-	2.50	**1.65**	3.50	**1.65**
-	**Quantidade vitórias**	-	7	**11**	0	**11**

Tabela 32 – Resultados do Cenário III com o classificador RL e porcentagem de 50% dos dados.

Fonte: próprio autor (2020).

#	Base	Quant	RS	UN	IG	EN
1.	Bolsonaro_facebook	366	0.8399	0.8586	**0.8698**	0.8393
2.	Bolsonaro_twitter	301	0.6892	0.7125	0.6609	**0.7320**
3.	Haddad_facebook	366	**0.8712**	0.8685	0.8526	0.8685
4.	Haddad_twitter	267	**0.7278**	0.6842	0.7198	0.6663
5.	Ambos_facebook	667	0.8158	0.8282	**0.8325**	0.8264
6.	Ambos_twitter	482	0.6978	**0.7035**	0.6680	0.6826
7.	Sentiment140_test	174	0.6153	0.6127	**0.6662**	0.6213
8.	Sentiment140_10000_1	3,500	0.7022	**0.7192**	0.7070	**0.7192**
9.	Sentiment140_5000_1	1,750	0.6655	0.6867	0.6715	**0.6871**
10.	Sentiment140_2500_1	875	0.7113	0.7217	0.6891	**0.7232**
11.	Sentiment140_1000_1	350	0.6427	0.6532	**0.6783**	0.6532
12.	Sentiment140_10000_2	3,500	0.7044	0.7105	0.7007	**0.7115**
13.	Sentiment140_5000_2	1,750	0.6872	**0.7116**	0.6893	0.7080
14.	Sentiment140_2500_2	875	0.6729	0.6759	0.6759	**0.6793**
15.	Sentiment140_1000_2	350	0.6102	**0.6253**	0.6180	**0.6253**
16.	Sanders_apple	351	0.5061	0.5207	**0.5225**	0.5063
17.	Sanders_google	293	**0.4719**	0.4372	0.4466	0.4528
18.	Sanders_microsoft	302	0.4218	0.4401	0.4007	**0.4454**
19.	Sanders_twitter	252	0.4398	**0.4838**	0.3567	0.4303
20.	Sanders_all	1,198	**0.4566**	0.4405	0.4304	0.4355
-	**Média**	-	0.6475	**0.6547**	0.6428	0.6507
-	**Ranking médio**	-	2.90	**2.00**	2.90	2.05
-	**Quantidade vitórias**	-	4	5	5	**8**

Tabela 33 – Resultados do Cenário III com o classificador SVM e porcentagem de 50% dos dados.