UNIVERSIDADE FEDERAL FLUMINENSE

ALEXANDRE DE CASTRO LUNARDI

CLASSIFICAÇÃO MULTICLASSE DE TEXTOS BASEADA EM DIVISÕES BINÁRIAS ADAPTADAS AO DOMÍNIO

Niterói

2016

UNIVERSIDADE FEDERAL FLUMINENSE

ALEXANDRE DE CASTRO LUNARDI

CLASSIFICAÇÃO MULTICLASSE DE TEXTOS BASEADA EM DIVISÕES BINÁRIAS ADAPTADAS AO DOMÍNIO

Dissertação apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Mestre. Área de Concentração: Inteligência Artificial.

Orientador:
Prof. Dr. JOSÉ VITERBO FILHO

Coorientadora:
Profª. Drª. FLÁVIA CRISTINA BERNARDINI

Niterói
2016

Ficha Catalográfica elaborada pela Biblioteca da Escola de Engenharia e Instituto de Computação da UFF

L961 Lunardi, Alexandre de Castro
Classificação multiclasse de textos baseada em divisões binárias adaptadas ao domínio / Alexandre de Castro Lunardi. – Niterói, RJ : [s.n.], 2016.
104 f.

Dissertação (Mestrado em Computação) - Universidade Federal Fluminense, 2016.
Orientadores: José Viterbo Filho, Flávia Cristina Bernardini.

1. Inteligência artificial. 2. Aprendizagem de máquina. 3. Mineração de opinião. I. Título.

CDD 006.3

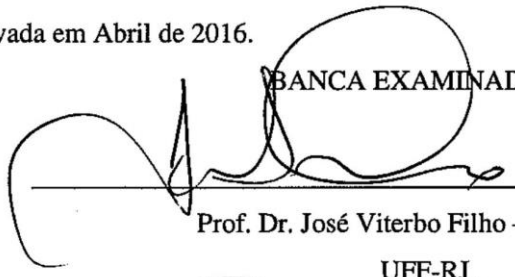
ALEXANDRE DE CASTRO LUNARDI

CLASSIFICAÇÃO MULTICLASSE DE TEXTOS BASEADA EM DIVISÕES BINÁRIAS ADAPTADAS AO DOMÍNIO

Dissertação apresentada ao Programa de Pós-Graduação em Ciência de Computação da Universidade Federal Fluminense, como requisito parcial para a obtenção do Grau de Mestre. Área de Concentração: Inteligência Artificial

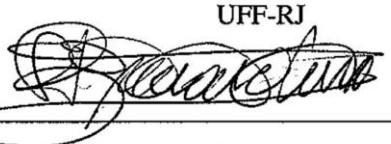
Aprovada em Abril de 2016.

BANCA EXAMINADORA



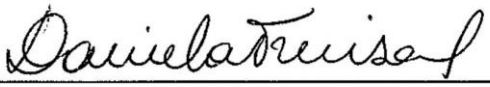
Prof. Dr. José Viterbo Filho – Orientador

UFF-RJ



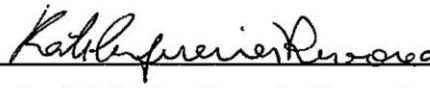
Profª. Drª. Flávia Cristina Bernardini – Coorientadora

UFF-RJ



Profª. Drª. Daniela Gorski Trevisan

UFF-RJ



Profª. Drª. Kate Cerqueira Revoredo

UNIRIO-RJ

Niterói

2016

À minha família e amigos.

AGRADECIMENTOS

Aos meus pais Marcos e Luísa, por toda a base e apoio que me concederam por todo esse período e à minha irmã Nicoli por demonstrarem que devemos lutar sempre pelo que acreditamos.

À minha família, em especial aos meus tios Yvone e Tadeu e a meus primos Yvonne Maria, Nuccio, Huguinho, Hugo e Caio da cidade do Rio de Janeiro que me receberam e me deram apoio total em todas as etapas do mestrado.

Ao meu orientador Viterbo pelos sábios conselhos e por transmitir seu conhecimento. Agradeço também por sua tranquilidade em orientar e incentivar o trabalho realizado. Agradeço também à professora Flávia com colaborações fundamentais na pesquisa.

A todos os funcionários do IC, principalmente aos secretários de pós-graduação.

Aos meus amigos de mestrado Leandro, Leonardo e Eider que de alguma forma colaboraram nesta fase.

Aos membros da banca pela disponibilidade e conselhos relativos a esse trabalho.

A CAPES pelo suporte financeiro em grande parte desse período.

“Talvez não tenha conseguido fazer o melhor, mas lutei para que o melhor fosse feito”.

Marthin Luther King

RESUMO

Desde o advento da Web 2.0, é cada vez mais comum encontrar valiosas opiniões ou comentários *online* relacionados a produtos, serviços, organizações, eventos e vários outros itens. Capturar e processar corretamente essas informações e descobrir o interesse do público em geral sobre algum item é de grande interesse para o mundo dos negócios e para os próprios consumidores. A informação agregada inferida a partir das opiniões de muitos usuários da web pode ser usada no processo de tomada de decisão por aqueles que prospectam um item. Os pesquisadores na área de análise de sentimentos vêm estudando técnicas e buscando desenvolver ferramentas para identificar as opiniões de usuários sobre determinados produtos ou serviços. Enquanto muitos trabalhos propõem a classificação binária de uma avaliação, algumas pesquisas focam na classificação multiclasse, como, por exemplo, a inferência de *ratings*. Neste caso, o objetivo principal é classificar cada opinião dentro de uma faixa múltipla de valores – tipicamente 1 a 5 estrelas – e não apenas como positivas ou negativas. Existem duas abordagens para a construção de um classificador multiclasse. A primeira consiste no emprego de algoritmos de aprendizagem construídos diretamente para problemas multiclasse, como Naïve Bayes, árvores de decisão, redes neurais, e assim por diante. A segunda baseia-se na decomposição do problema multiclasse inicial para uma combinação de problemas binários. Há duas técnicas clássicas para decompor o problema: (i) "um x um"; e (ii) "um x todos". Uma abordagem alternativa para decompor o problema é implementada pelo algoritmo *Nested Dichotomies* (divisões binárias em cascata), que constrói árvores de todas as combinações possíveis de divisões binárias do conjunto de classes. Nesta dissertação, propomos um modelo baseado em uma adaptação do algoritmo divisões binárias em cascata, visando construir uma única árvore de divisões binárias de classes, em que a primeira divisão é capaz de dividir as classes em dois conjuntos, recomendados e não recomendados. Esta divisão é determinada a partir de uma análise do domínio, com o auxílio de questionários para identificar a preferência dos usuários. Apresentamos um estudo de caso em que o modelo de classificação é aplicado a um conjunto de dados subjetivos extraídos do site *TripAdvisor*, contendo revisões classificadas com valores de 1 a 5. Os resultados do classificador proposto foram comparados com aqueles de outras técnicas de classificação multiclasse tradicionalmente utilizadas. Verificou-se que os resultados obtidos superaram muitos dos trabalhos anteriores em análise de sentimento, considerando a acurácia final da classificação multiclasse e, especialmente, a acurácia da primeira divisão, conforme as expectativas iniciais.

Palavras-chave: Aprendizado de Máquina, Análise de Sentimentos, Problema de Inferência de *Ratings*, Classificação Multiclasse.

ABSTRACT

Since the Web 2.0 advent, it is increasingly common to find valuable online opinions or reviews related to products, services, organizations, events and various other items. Properly capturing and processing such information, and discovering the interest of the general public on any item, is of great interest for the business world and also for the online customers in general. The aggregated information inferred from the opinions of many web users can be used in the decision-making process by those that prospect such items. The sentiment analysis community is thus studying techniques and developing tools to identify users' sentiments on reviewed products or services. While many works propose the classification of the polarity of a review (binary classification), some researches focus on the inference of ratings (multiclass classification) for the reviews. In this case, the main goal is to classify each review in a grade range – typically 1 to 5 stars – and not just as positive or negative. There are two approaches for multiclass classifier construction. The first one uses learning algorithms constructed for multiclass problems, such as Naïve Bayes, induction of Decision Trees, generalization of Neural Networks, and so on. The second one is based on decomposing the initial multiclass problem into a combination of binary problems. There are two classical techniques for transforming the problem: (i) one-vs-one; and (ii) one-vs-all. A newer approach for decomposing the problem is the one implemented by the Nested Dichotomies algorithms, which constructs trees of (all the) possible combinations of class binary splits. In this thesis, we propose an adaptation of the Nested Dichotomies algorithm, aiming at creating a single tree of binary divisions of classes, based on domain characteristics, making more efficient the construction of the classifier. That is, we propose a method for creating a model for multiclass text classification, defined by a tree of sequential binary splits, in which the first split separates the classes in recommended and not recommended sets. This split is determined by a domain analysis, with the help of questionnaires to identify the user's preferences. We present a case study in which the classification model is applied to a set of subjective data extracted from TripAdvisor website, containing reviews labeled with *ratings* from 1 to 5. The proposed model was implemented and tested with several different base-classifiers, and the best results were obtained with the Naïve Bayes algorithm. The results of the proposed classifier were compared with those of other multiclass classification techniques traditionally used. It was found that our results overcame many of the earlier works in sentiment analysis, considering the final accuracy of the multiclass classification and specially the accuracy of the first split, meeting our initial expectations.

Keywords: Machine Learning, Sentiment Analysis, Rating Inference Problem, Multiclass Classification

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

NLP: *Natural Language Processing*;

RIP: *Rating-Inference Problem*;

TEC'S: Técnicas de Extração de Características;

IG: *Information Gain*;

GR: *Gain Ratio*;

CHI: *Chi-Square*;

OvA: *One-vs-All*;

OvO: *One-vs-One;*

SVM: *Support Vector Machine;*

SMO: *Sequential Minimal Optimization;*

PQ: Programação Quadrática;

NBM: *Naive Bayes Multinomial;*

NB: Naive Bayes;

ND: *Nested Dichotomies;*

END: *Ensemble Nested Dichotomies;*

tfidf: *Term Frequency–Inverse Document Frequency*;

MaxEnt: *Maximum Entropy*;

kNN: *k-Nearest Neighbors*;

RI: Recuperação de Informação;

NDiST: *Nested Dichotomies Single Tree*;

SUMÁRIO

CAPÍTULO 1 – INTRODUÇÃO

Opiniões sempre foram úteis no que diz respeito à tomada de decisões dos seres humanos (CAMBRIA *et al.*, 2013). Nossas escolhas sempre foram, em certo grau, dependentes das opiniões e conselhos de outras pessoas (LIU, 2012). Além disso, é de grande importância para empresas conhecer o sentimento das pessoas em relação a um produto ou serviço, o que permite realizar previsões de mercado ou recomendações aos consumidores (TURNEY, 2002), tornando essas empresas mais próximas de seu público-alvo.

Com o advento da Web 2.0, é cada vez mais fácil encontrar opiniões valiosas relacionadas a produtos, serviços, organizações, indivíduos, eventos, pesquisas e vários outros domínios. Isso se deve ao crescente uso de redes sociais, blogs e, principalmente, ferramentas que permitem aos usuários deixar registrado seus comentários sobre algum produto ou serviço em sites de comércio eletrônico. Essa crescente disponibilização de dados é também conhecida como "web social" (CAMBRIA *et al.*, 2013). Isso pode ser notado em sites de comércio eletrônico como o *Booking.com*™[1] e a *Amazon*™[2], nos quais os clientes podem deixar seus comentários, revelando suas opiniões a respeito do produto ou serviço oferecido.

Com essa grande quantidade de informação disponível na Internet, analisar todo o conjunto de opiniões encontradas se tornou uma tarefa inviável para o ser humano. Com isso, capturar e processar de forma adequada essas informações por meio de técnicas computacionais – a chamada mineração de opiniões ou análise de sentimentos (CAMBRIA *et al.*, 2013) – se torna fundamental para permitir a identificação do real interesse do público sobre algum item.

A comunidade científica vem, dessa forma, desenvolvendo ferramentas que visam auxiliar na recuperação e tratamento de opiniões – ou avaliações – sobre produtos e serviços, disponíveis na web social. Dessa forma, pesquisas sobre mineração de opiniões e/ou análise de sentimentos são uma das áreas mais ativas e desafiantes, abordadas principalmente na área de Processamento de Linguagem Natural (NLP).

Análise de sentimentos ou *mineração de opiniões* são os principais termos empregados para descrever a análise automática de textos subjetivos, isto é, textos que contém não apenas fatos ou explicações técnicas sobre algo, mas alguma opinião a respeito de um item. A partir

[1] http://www.booking.com/
[2] http://www.amazon.com/

dessa análise, é possível identificar aspectos distintos de um item, por exemplo, a localização ou a limpeza de um hotel, a durabilidade ou a facilidade de uso de um eletrodoméstico. É possível realizar também uma análise agregada das avaliações sobre um determinado item, identificando o sentimento geral em relação a esse item. Com isso, poderemos saber, por exemplo, se um hotel é recomendado pelos consumidores que ali se hospedaram, de acordo com o conjunto de opiniões emitidas. Ou seja, considerando-se apenas uma escala binária, os aspectos específicos de um produto ou serviço podem ser classificados como bons ou ruins. Por exemplo, pode-se avaliar se a câmera de um celular é recomendável ou não, ou se a localização de um hotel é boa ou não. A partir da análise de diversos aspectos, chega-se a uma conclusão sobre o sentimento final sobre um item. Por exemplo, após avaliar atributos de um celular como a câmera, facilidade de uso, preço e vários outros aspectos, pode-se chegar a uma conclusão final sobre a recomendação do celular. Outra aplicação possível é construir classificadores que infiram *ratings* a partir de opiniões de usuários utilizando aprendizado supervisionado. Tais classificadores podem ser utilizados por sistemas de recomendação de ratings pelas empresas que vendem produtos e/ou serviços e coletam opiniões de seus usuários. Esse é o foco de aplicação deste trabalho.

Considerando que um dado texto – que representa a avaliação de um usuário – seja subjetivo, a classificação do mesmo pode ser binária ou multiclasse. Na classificação binária, o objetivo é rotular essa avaliação como boa ou ruim (positiva ou negativa), com relação ao sentimento expressado pelo usuário. Um exemplo de ferramenta criada para analisar o sentimento de uma opinião é a *sentiment140*[3], proposta por (GO; BHAYANI; HUANG, 2009). Nesse site é possível conhecer o sentimento em relação a uma entidade (empresa, produto, serviço...) utilizando *tweets* sobre a entidade em análise. Essa ferramenta seleciona os *tweets* de acordo com a palavra-chave informada pelo usuário e classifica os *tweets* encontrados como positivos ou negativos. Além disso, eles criam um gráfico com a porcentagem de *tweets* positivos e negativos. Outros sites também exemplificam o uso da análise de sentimentos, como o *NLTK Text Classification*[4], no qual um usuário digita uma opinião sobre algo e o sistema determina se é uma opinião e, caso positivo, se a polaridade dela é positiva ou negativa. Outro site é o *Skyttle*[5], no qual a opinião informada também é

[3] http://www.sentiment140.com
[4] http://text-processing.com/demo/sentiment/
[5] http://www.skyttle.com/demoin

classificada como boa ou ruim e, além disso, as frases com sentimento bom são marcadas de verde e as frases com sentimento ruim são marcadas como vermelho.

A classificação ou análise multiclasse analisa uma avaliação considerando várias (e mais de duas) escalas do sentimento, como por exemplo, “bom”, “neutro” ou “ruim”. Em diversos cenários, os produtos ou serviços são classificados em escalas de valores múltiplos. Pode-se citar como exemplo o site *Booking*, no qual os hotéis são classificados com notas que variam de 0 a 10. Assim sendo, uma forma típica de análise multiclasse é o Problema de Inferência de *Rating* (*Rating-Inference Problem* - RIP), baseada em escalas de *rating* que tipicamente variam de 1 a 5 estrelas (PAN e LEE, 2005). Em alguns casos, essa escala pode ser analisada como 4 classes, na qual a classe 3 (neutra) é desconsiderada, ou como 3 classes, nas quais as classes 1 e 2 são unidas, assim como as classes 4 e 5.

Escalas baseadas em *ratings* estão presentes em larga escala em ferramentas de avaliação disponíveis em serviços como *Amazon*™ e *Netflix*™[6], e projetos como o *GroupLens*™[7] (KONSTAN *et al.*, 1997) com avaliações utilizando opiniões rotuladas em uma escala 5-*ratings*. A importância da avaliação correta pode ser comprovada de acordo com a pesquisa do site *ComScore*[8], que mostra que os consumidores têm maior disposição para gastar entre 20% e 99% a mais em serviços que tenham uma classificação excelente (5 estrelas) do que um serviço classificado com 4 estrelas (Bom). Para o domínio de hotéis, esse percentual é de 38%. Obviamente, técnicas de classificação binária não permitiriam que essa divisão (4 e 5 estrelas) fosse identificada automaticamente em trabalhos em análise de sentimentos. Além disso, considerando a grande utilização de várias classes na análise de textos subjetivos, é de grande interesse analisar não somente a polaridade de um item, mas também avaliar os graus de positividade e negatividade por meio de *ratings* numéricos, baseados, por exemplo, na escala de Likert (LIKERT, 1932), variando de 1 a 5 estrelas.

Embora seja uma forma de classificação essencial devido ao grande uso de *ratings* e de grande importância para a comunidade e para empresas, o número de trabalhos disponíveis em análise de sentimentos multiclasse é muito inferior se comparado aos trabalhos com classificação binária. Mesmo em cenários tipicamente de escalas múltiplas, grande parte das análises de opiniões consideram apenas duas classes principais – agrupadas em recomendado

[6] http://www.netflix.com
[7] http://grouplens.org
[8] http://comscore.com

ou não recomendado –, em que, em uma escala de 1 a 5 estrelas, por exemplo, 4 ou 5 estrelas são consideradas recomendáveis e 1, 2 ou 3 estrelas não são recomendáveis.

Um possível emprego para as técnicas de análise multiclasse, seria permitir a classificação automática de comentários de usuários em sites de produtos ou serviços, diminuindo o chamado efeito manada (*herding effect*) (WANG e WANG, 2014). Esse efeito acontece na avaliação direta realizada pelos usuários quando estes se deixam influenciar pela avaliação da maioria. Por exemplo, um usuário pode ter achado um celular bom (4 estrelas), mas caso a maioria dos outros usuários tenham considerado excelente (5 estrelas), existe a possibilidade de que ele avalie o celular com base na média dos outros usuários e não no que o celular representou para ele. Além disso, a classificação automática de comentários baseada em análise multiclasse poderia simplesmente evitar os erros da classificação realizada pelo usuário, e casos em que um comentário não condiz o número de estrelas atribuídos, poderiam ser evitados. Mesmo com a divisão com *ratings* 4 e 5 consideradas recomendadas, de acordo com a pesquisa feita pelo site *PracticalECommerce*[9] a inferência de *ratings* seria de grande utilidade tendo em vista que uma estrela a mais ou a menos pode fazer a diferença no momento da compra de um item.

1.1 DEFINIÇÃO DO PROBLEMA

O termo Inferência de *Rating* (PANG e LEE, 2005) é utilizado para os problemas de classificação de uma opinião em uma escala de estrelas. Devido à grande importância desse tipo de análise, muitos trabalhos em análise multiclasse propõem o uso de algum modelo de classificação. Para construir tal modelo de classificação, podem ser utilizados algoritmos de aprendizado de máquina. Neste trabalho, entende-se que a associação de um número de estrelas a uma opinião é considerada um tipo de análise de sentimento, já que o sentimento de quem emite a opinião tem uma associação direta ao número de estrelas associado a essa opinião. Assim sendo, neste trabalho é considerada a abordagem de análise de sentimentos para inferência de *rating*.

Embora o problema de análise de sentimentos para inferência de ratings seja de grande importância, considerando o grande número de sistemas que utilizam e que ainda poderão utilizar esse tipo de classificação, o número de pesquisas que ataca esse problema utilizando

[9] http://www.practicalecommerce.com/articles/93017-Study-5-Star-Reviews-Not-Necessarily-Helpful

aprendizado supervisionado para problemas multiclasse é menor do que utilizando aprendizado para problemas binários.

Existem duas abordagens para a construção de um classificador multiclasse para inferência de *rating*[10]. A primeira consiste no emprego de algoritmos de aprendizagem construídos diretamente para problemas multiclasse, como Naive Bayes, Árvores de Decisão, Redes Neurais, e assim por diante. A segunda baseia-se na decomposição do problema multiclasse inicial para uma combinação de problemas binários. Há duas técnicas clássicas para decompor o problema: (i) "um *vs* um", em que um classificador binário é construído de modo a distinguir entre um par de classes e por isso o número de classificadores binários que compõem o classificador multiclasse é dado pela combinação de todos os pares de classes; e (ii) "um *vs* todos", em que um classificador binário é construído de modo a distinguir uma classe de todas as outras e o número de classificadores binários necessários corresponde ao número de classes. Estas abordagens são comumente utilizadas quando algoritmos SVM são indicados para o problema, por exemplo.

1.2 OBJETIVOS

Esse trabalho tem como principal objetivo propor um modelo eficiente para o problema de classificação de *rating* multiclasse de textos avaliativos, do ponto de vista da abordagem de análise de sentimentos. Para estudar esse problema, é utilizada uma escala de *rating* de 1 a 5 estrelas.

Com essas divisões binárias baseadas no algoritmo Nested Dichotomies (FRANK e KRAMER, 2004), além de aproveitar o grande número de trabalhos disponíveis na análise da polaridade de uma opinião, o modelo proposto é capaz de, em uma primeira divisão, separar as opiniões em recomendadas ou não, indo de encontro com pesquisas que demonstram que *ratings* entre 1 e 3 são menos influentes na compra de um item, ou seja, é adaptado ao domínio. Essa divisão, por exemplo, torna-se impossível se pensarmos em um classificador individual ou em algoritmos como o *one-vs-all* (OvA) sem que haja algum tipo de agrupamento prévio das opiniões.

[10] Um classificador multiclasse pode ser do tipo *crisp* (*crisp multiclass classifier*), para o qual não é considerada uma escala de ordem entre as classes, ou pode ser do tipo *rating* (*rating multiclass classifier*), para o qual é considerada uma escala de ordem entre as classes. Neste trabalho, são considerados classificadores multiclasse do tipo *rating*, pois há uma escala de avaliação de produtos e/ou serviços.

Logo, o modelo proposto não só aproveita os métodos de extração de características como também utiliza um novo método de divisões binárias, chamado Nested Dichotomies (ND), para o problema da classificação de sentimentos multiclasse.

Outra área importante a ser estudada é o pré-processamento textual. Conforme dito por Liu (LIU, 2012), uma fase essencial é de extração de características, fundamental no desempenho de algoritmos de aprendizado de máquina. Além disso, (PRUSA; KHOSHGOFTAAR; DITTMAN, 2015) demonstram a importância da fase de seleção de características na melhoria do desempenho dos algoritmos de aprendizado utilizados pelos autores. Dessa forma, essas técnicas devem ser muito bem exploradas e configuradas.

1.3 METODOLOGIA

Em uma primeira etapa da metodologia, um conjunto de opiniões sobre hotéis disponíveis na web foi analisado, a fim de avaliar a classificação multiclasse com o uso de algoritmos de aprendizado construídos para problemas multiclasse. Para o treinamento desses classificadores, é necessária uma fase inicial de tratamento textual. Assim, Técnicas de Extração de Características (TEC's) presentes na literatura foram utilizadas, como Ganho de Informação (IG), Ganho Médio (GR) e Chi-quadrado (CHI). Essas fases são essenciais, já que uma boa seleção de características para o treinamento está diretamente relacionada à acurácia do algoritmo de aprendizado (LIU, 2012).

A segunda etapa propõe o uso de um método que utiliza várias etapas de classificação binária a fim de classificar as opiniões da base de dados utilizada neste trabalho resultando em uma escala de *ratings* de 5 classes.

Para avaliar esse modelo de divisões binárias, as seguintes etapas foram realizadas:

- Configurar um estudo de caso baseado em avaliações retiradas do site *TripAdvisor*™[11], disponíveis em[12], que foi utilizado na pesquisa de (WANG; LU; ZHAI, 2010);
- Testar o desempenho do algoritmo de divisões binárias já que, embora este algoritmo seja o mais indicado, como será visto, caso a acurácia fosse muito inferior a outros métodos, esse seria um fator que impossibilitasse o uso de divisões binárias baseada no ND;
- Avaliar o desempenho em relação a acurácia final e a acurácia da primeira divisão.

[11] http://tripadvisor.com
[12] http://times.cs.uiuc.edu/~wang296/Data/

Em relação aos testes com algumas das técnicas presentes na literatura, esse trabalho exibe resultados para uma série de combinações de TEC's e algoritmos de aprendizado de máquina no domínio de hotéis.

1.4 ORGANIZAÇÃO

Capítulo 2: A Análise de Sentimentos e o Aprendizado Supervisionado. Esse Capítulo apresenta a definição de análise de sentimentos bem como a etapa de configuração das opiniões por meio das técnicas de extração de características até a vetorização para o uso dos algoritmos de aprendizado de máquina.

Capítulo 3: Trabalhos Relacionados. Capítulo com trabalhos relacionados com trabalhos das principais áreas de análise de sentimentos que estão descritas no Capítulo 2, com foco para as classificações binárias e multiclasse.

Capítulo 4: Avaliação dos Algoritmos Nativos. Capítulo com a metodologia de trabalho utilizada desde a etapa de pré-processamento das opiniões até os testes com os algoritmos de aprendizado.

Capítulo 5: Uma Proposta de um Modelo de Divisões Binárias. Capítulo com a proposta de um classificador multiclasse com etapas binárias. Neste Capítulo também são apresentadas outras técnicas adaptadas, bem como o algoritmo Nested Dichotomies.

Capítulo 6: Estudo de Caso. Este Capítulo apresenta um estudo de caso com avaliações feitas sobre hotéis, descrevendo o domínio e a importância dos *ratings*. Além disso, a implementação do modelo é descrita.

Capítulo 7: Avaliação Experimental do Classificador NDiST. Capítulo com os resultados da arquitetura proposta para a classificação multiclasse final e para a primeira divisão realizada.

Capítulo 8: Conclusões e Trabalhos Futuros.

CAPÍTULO 2 – A ANÁLISE DE SENTIMENTOS E O APRENDIZADO DE MÁQUINA

Um dos primeiros trabalhos a analisar o sentimento das pessoas através de dados da web foi discutido em (DAS e CHEN, 2001), e utilizou o termo *extração de sentimento* para capturar a influência da opinião de indivíduos no domínio de finanças. Já Pang *et al.* (PANG; LEE; VAITHYANATHAN, 2002), utilizam o termo *classificação de sentimentos* para avaliar documentos considerando o sentimento geral de uma opinião, classificando-as como positivas ou negativas. Outro trabalho inicial é o de Turney (TURNEY, 2002) que visa classificar opiniões como *recomendadas* ou *não recomendadas* (em inglês, *thumbs up* e *thumbs down*). Apenas em Nasukawa e Yi (NASUKAWA e YI, 2003) o termo *análise de sentimentos* é empregado, e assim como em (PANG; LEE; VAITHYANATHAN, 2002), os autores introduzem uma pesquisa para classificar uma opinião como positiva ou negativa.

De acordo com Cambria *et al.* (CAMBRIA et al., 2013), a mineração de opiniões pode ser agrupada em quatro campos, na qual a análise pode ser realizada por meio de:

- Palavras-chave e afinidade léxica: classifica o texto de acordo com a presença de palavras sem sentido ambíguo, tais como “feliz”, “triste” e “medo”. Além de detectar palavras óbvias, também atribui a outras palavras uma relação de afinidade com um sentimento, seja ele bom ou ruim. Um exemplo de aplicação é o *SentiWordNet*[13] 3.0 (BACCIANELLA; ESULI; SEBASTIANI, 2010), um recurso léxico criado a fim de orientar aplicações em mineração de opiniões.
- Aprendizado de máquina: utiliza modelos de aprendizado de máquina, como Naive Bayes e SVM, para classificar um texto. Nesse caso, o sistema, além de aprender a importância de uma palavra-chave óbvia, considera outras palavras que podem ser fundamentais, além da possibilidade de analisar a frequência ou a pontuação de um texto.

Orientação semântica: esses métodos calculam a orientação semântica (por exemplo, para o problema binário saber a polaridade da palavra) de uma palavra baseada na coocorrência da mesma com palavras que possuem a mesma orientação. O principal trabalho que propõem um método que calcule essa orientação semântica é o algoritmo proposto por Turney, 2002 (TURNEY, 2002). O algoritmo Pointwise Mutual Information and Information Retrieval (PMI-IR) é utilizado a fim de medir a similaridade de pares de palavras ou frases. A

[13] http://www.sentiwordnet.isti.cnr.it

orientação é calculada pela comparação da similaridade de uma palavra em relação aos sentimentos positivo e negativo.

- Baseado em conceitos: usam ontologias ou redes de palavras-chave para realizar a análise textual. Podem analisar expressões que não possuem uma emoção explícita, mas estão relacionadas a um sentimento implicitamente. No trabalho realizado por Kontopoulos *et al* (KONTOPOULOS *et al.*, 2013), é proposto o uso de ontologias a fim de melhorar o desempenho da análise de sentimentos no *Twitter*™.

Como pode ser notado, existem várias técnicas de análise de sentimentos, entretanto, o foco dessa dissertação está na utilização de modelos de aprendizado de máquina juntamente com técnicas de extração de características a fim de treinar e classificar um conjunto de opiniões, de acordo com o esquema exibido pela Figura 1.

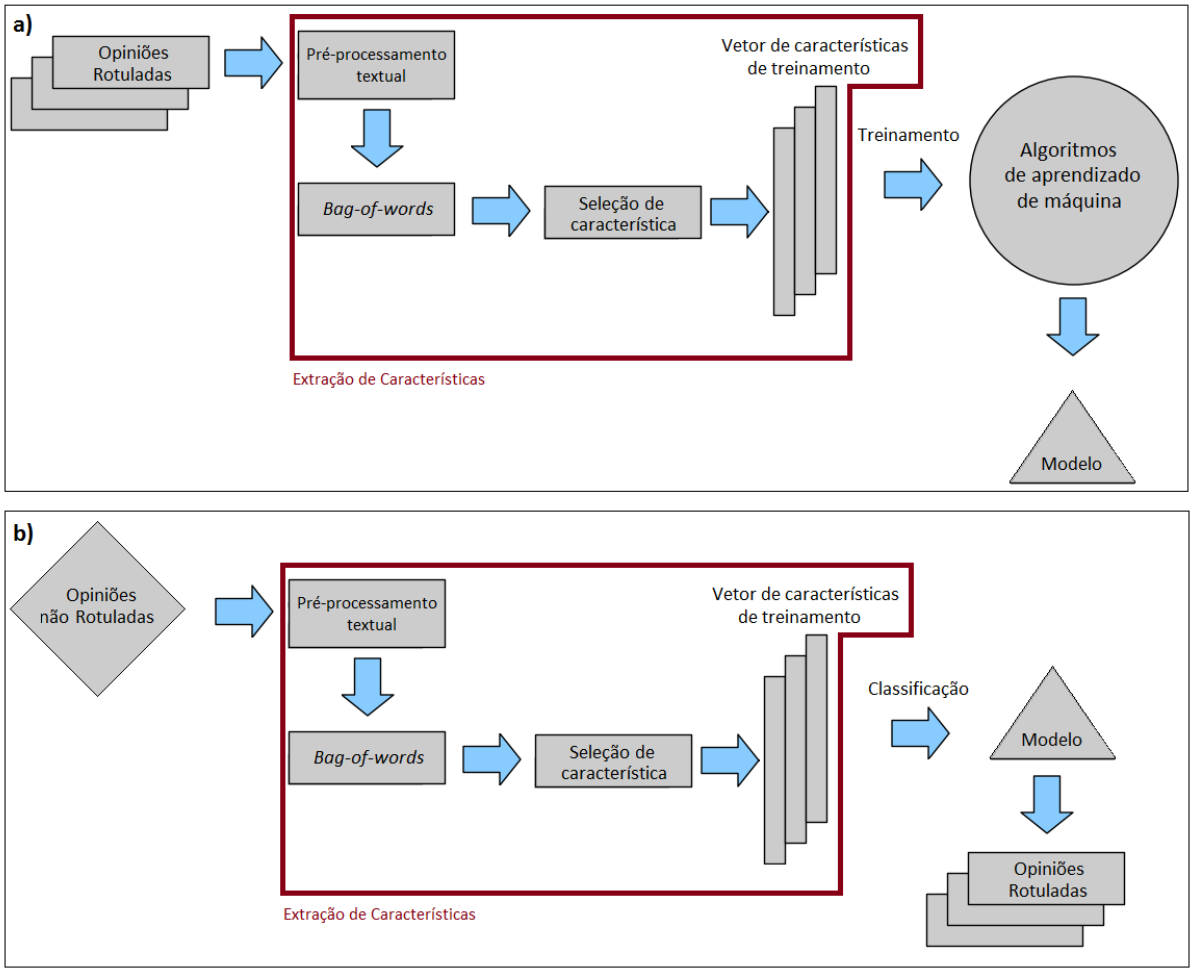


Figura 1. Processo de análise textual com extração de características e aprendizado de máquina. (a) Processo de treinamento. (b) Processo de classificação

Na parte a), o processo de extração de características e o treinamento dos algoritmos de aprendizado são descritos. Após a seleção de uma base de dados com opiniões previamente rotuladas, a fase de extração de características é dividida em quatro etapas. A primeira etapa, de pré-processamento textual, consiste na retirada de caracteres especiais, *stopwords* e tratamento da negação. A segunda etapa, *bag-of-words*, transforma cada opinião da base de dados em um conjunto de unigramas e bigramas. A terceira etapa é a fase de seleção de características que consiste na escolha dos melhores n-gramas para o treinamento dos algoritmos de classificação. A última etapa é a de vetorização que transforma a base de dados em documentos que são mais facilmente compreendidos pelos algoritmos de aprendizado. Por fim, esses algoritmos criam modelos de classificação que podem ser utilizados para categorizar opiniões sem rótulos.

A parte b) representa o modelo de classificação de novas instâncias. As opiniões não rotuladas selecionadas passam pelo mesmo processo de extração de características descrito na parte a). Após a extração de características, as opiniões são classificadas através de um modelo criado na parte a).

Logo, na Seção 2.1, a definição do termo “análise de sentimentos” é apresentada. As etapas de pré-processamento, *bag-of-words*, seleção de características e vetorização de características para treinamento estão descritas na Seção 0. Essas técnicas estão presentes em alguns dos principais trabalhos que tem como objetivo a análise de sentimentos por meio do aprendizado de máquina. Na Seção 2.3, os principais algoritmos de classificação utilizados em análise de sentimentos são apresentados.

Embora existam centenas de técnicas e métodos de análise de sentimentos presentes na literatura, como pode ser notado no Capítulo 3, que discute os trabalhos relacionados, alguns passos comuns e técnicas bem utilizadas foram selecionados baseado na importância dos trabalhos e no bom desempenho dos métodos existentes. Além disso, as medidas avaliativas e alguns problemas existentes no processo de mineração de dados são descritos nas seções 2.4 e 2.5, respectivamente.

2.1 DEFINIÇÕES

Segundo (LIU, 2012), a “análise de sentimentos” ou “mineração de opiniões” é o campo de estudo que analisa as atitudes, emoções, sentimentos e as opiniões das pessoas em relação a entidades - como produtos, serviços, organizações, eventos, tópicos - e os atributos

dessas entidades. Ela é um campo desafiador na área de Processamento de Linguagem Natural já que trata várias questões de PLN como a tratamento de negação e retirada de palavras-chave e pode cobrir muitos problemas, desde a classificação em relação à polaridade de uma opinião até o processo de sumarização do sentimento geral sobre algo.

Seja um texto *d*, a tarefa inicial na mineração de opiniões consiste em determinar se *d* é subjetivo, ou seja, expressa um sentimento. Seja tal texto considerado subjetivo, formalmente, uma opinião pode ser representada como uma 5-tupla (LIU, 2012) O = (e, a, s, h, t), na qual:

- *e* é o nome da entidade ou objeto ao qual uma opinião se refere;
- *a* é o atributo específico da entidade;
- *s* é o sentimento do autor da opinião em relação a um atributo ou entidade;
- *h* é o autor da opinião, e;
- *t* é a data na qual a opinião foi criada.

Como exemplo, temos a seguinte opinião sobre o hotel *H* retirada do site TripAdvisor[1]:

> *User: DesDeeMona (h)*
> *Title: A magnicent building of fading grandeur, redolent of earlier times*
> *Rating: 4*
> *Date: April 28, 2015 (t)*
> *Review: "Ground floor lobbies and suites with art deco design are impressive. Rooms (a) are spacious (s) with high ceilings and plenty of room in the en suite shower. Yes, lifts are a little slow, the paint is peeling, the plaster cracking, there are stains on the carpet - but hey, everything works, the sheets are well laundered and beds are comfortable".*

Nesse exemplo, pode-se observar que o hotel é a entidade e um dos atributos são os quartos, destacados com a letra *(a)* no documento. Eles são classificados pelo autor como "espaçosos", como demarcado acima pela letra *(s)*. Para (DAVE *et al.*, 2003), a tarefa ideal na análise de sentimentos deveria processar um conjunto de opiniões sobre certa entidade, gerando uma lista de atributos para a mesma e agregar opiniões sobre os atributos da entidade. Entretanto, outros autores consideram apenas a entidade da opinião e o sentimento final, como feito em (PANG; LEE; VAITHYANATHAN, 2002).

De forma resumida, a taxonomia de análise de sentimentos está presenta na Figura 2. A ideia básica para o problema de análise de sentimentos é que um usuário emita uma opinião, também chamada de avaliação ou revisão. Essa avaliação pode ser sobre uma entidade ou item (como um hotel, por exemplo) ou pode ser relativa a um aspecto ou atributo específico de um item (a localização do hotel). Esse sentimento geralmente pode ser classificado em duas ou mais classes. A forma mais comum de classificação é a classificação binária, que diz se uma opinião é positiva ou negativa. Além disso, outras formas de classificação merecem destaque como a classificação por meio de *ratings* (presentes no site da *Amazon*) ou por meio de notas (*Booking.com*).

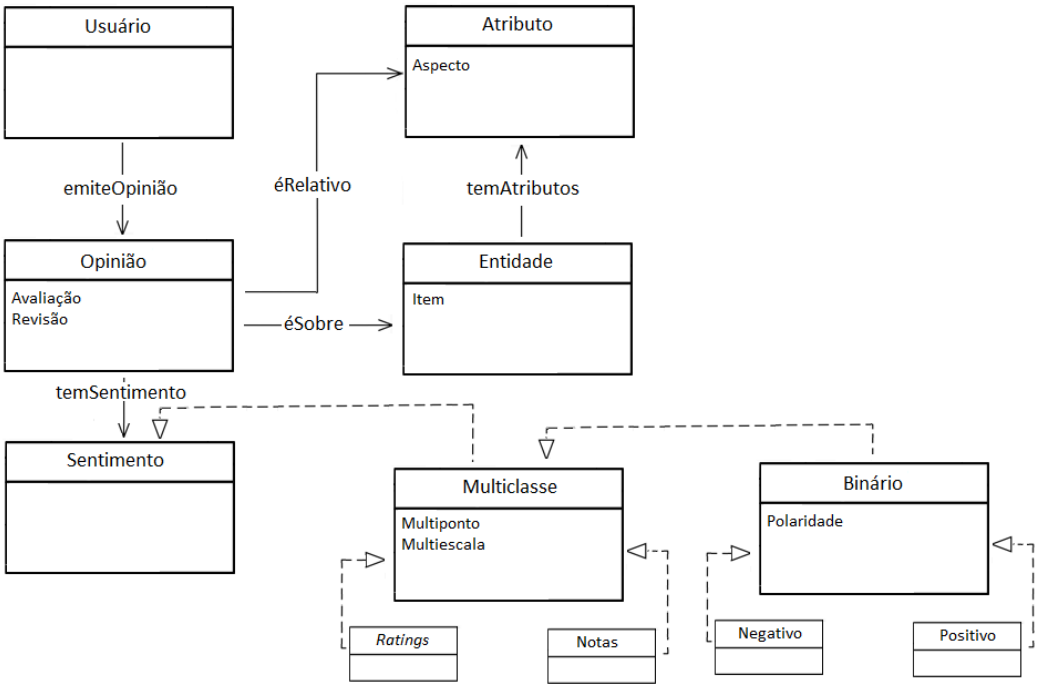


Figura 2. Ontologia com a análise de sentimentos

2.2 EXTRAÇÃO DE CARACTERÍSTICAS

Seja um conjunto *D* de opiniões selecionadas, algumas fases são essenciais no pré-processamento textual. Embora as Técnicas de Extração de Características (TEC's) descritas a seguir não agreguem todas as formas de análise disponíveis, por meio destes passos é possível configurar um bom documento que possa ser compreendido por um algoritmo de aprendizado de máquina, cuja análise é o foco deste trabalho.

2.2.1 PRÉ-PROCESSAMENTO TEXTUAL

O primeiro passo para a construção de um documento compreensível para os algoritmos de classificação é selecionar uma base de dados com textos avaliativos, isto é, textos que possuam um sentimento em relação a um item.

Com as opiniões a serem analisadas devidamente selecionadas, o próximo passo do pré-processamento é a *tokenization*, que consiste na retirada de caracteres como vírgulas, acentos e pontuações. Em alguns trabalhos, alguns caracteres, como pontos de exclamação ou *emoticons* podem ser utilizados como característica de treinamento (GO; BHAYANI; HUANG, 2009) ou como forma de seleção de opiniões para a criação de uma base de dados (PAK e PAROUBEK, 2010). Em casos nos quais as opiniões são extraídas diretamente de páginas web, a retirada de *tags* em HTML também deve ser realizada como feito em (BEINEKE *et al.*, 2004) e (KANG; YOO; HAN, 2012). Alguns exemplos de caracteres especiais estão presentes na Tabela 1.

Tabela 1. Exemplos de caracteres especiais

Descrição	**Token**
Acentos	´ ~ ^
Pontuação	‘ ’ , . ; : ? !
Especiais	@ # * ( ) &
Emoticons	:) ;) :D :(:(;(
HTML	 <p>

Com a retirada desses caracteres especiais das opiniões, o próximo passo é o da normalização textual. Nesta etapa, estão incluídas a retirada de radicais, retirada de letras repetidas em algumas palavras e a correção ortográfica. A etapa de correção ortográfica pode ser notada em trabalhos como (KOULOUMPIS; WILSON; MOORE, 2011). Embora seja indiscutível a necessidade desse passo, poucos trabalhos que tem o foco na classificação de sentimentos citam a normalização textual na fase de extração de características. Esses passos podem ser melhor estudadas em livros como (MANNING e RAGHAVAN, 2009) que, além de apresentarem uma boa introdução sobre recuperação de informação e o pré-processamento textual, mostram a utilização de algoritmos de aprendizado para a classificação textual.

Outro passo importante na parte de tratamento das opiniões é a retirada de palavras consideradas com pouco ou nenhum sentimento, as chamadas *stopwords*[14]. O objetivo é

[14] Lista de *stopwords* que será utilizada: http://www.ranks.nl/stopwords

diminuir a quantidade de palavras que possam ser usadas no treinamento, retirando palavras que pouco influenciam na determinação do sentimento final de um texto.

Outro passo importante foi o tratamento de opiniões com palavras que expressam negação (PANG e LEE, 2008). Desta forma, frases como "*This is not bad*" ou "*That is not good*" tem seu sentimento invertido pelo *token* "*not*". A fim de tratar esse problema, palavras que tem como precedentes os modificadores *no, not ou nothing* podem ser transformadas em uma única palavra. Como exemplo, "*not good*" é representado pelo *token "not_good"* que é similar ao *token* "*bad"*.

2.2.2 N-GRAMAS – *BAG OF WORDS*

Com as opiniões normalizadas, cada palavra de uma opinião corresponde a um unigrama, como pode ser observado no trabalho de Pang *et al.*(PANG; LEE; VAITHYANATHAN, 2002). Além de unigramas, essas palavras podem ser agrupadas formando bigramas (duas palavras) ou n-gramas (duas ou mais palavras). Unigramas e bigramas são as principais formas de representação de *tokens* e possuem bons resultados na análise de sentimentos (LIU, 2012), tanto na classificação binária (PANG; LEE; VAITHYANATHAN, 2002) como multiclasse (PANG e LEE, 2005).

Seja a frase "*This cell phone is amazing*". Na Tabela 2, é exibido um exemplo da representação desta frase em unigramas e bigramas, sem que haja a retirada de nenhuma das palavras em etapas anteriores. Cada n-grama está separado por vírgulas na tabela e a união dos mesmos está representada pelo caractere "_". Nota-se que a ordem das palavras foi mantida em relação à estrutura da frase inicial e nem todos os n-gramas possíveis estão representados.

Tabela 2. Exemplo de bag-of-words com n-gramas

Unigrama	This, cell, phone, is, amazing
Bigrama	This_cell, cell_phone, phone_is, is_amazing

2.2.3 TÉCNICAS DE SELEÇÃO DE CARACTERÍSTICAS

Após a etapa de normalização textual, a fase de Seleção de Características é fundamental para a escolha dos n-gramas para o treinamento de algoritmos de aprendizado (LIU, 2012). Como demonstrado por (PRUSA; KHOSHGOFTAAR; DITTMAN, 2015) na

análise de dados recolhidos do *Twitter*™[15], a seleção de características pode melhorar significativamente o desempenho da classificação. Esta etapa consiste na escolha de n-gramas que serão utilizadas como atributos de treinamento.

Três métodos de seleção de características foram testados e analisados: Ganho de Informação, Ganho Médio e Chi-quadrado e estão descritas nas subseções posteriores. Trabalhos como (TANG; TAN; CHENG, 2009), (SHARMA e DEY, 2012) e (PRUSA; KHOSHGOFTAAR; DITTMAN, 2015) fazem uso de alguma técnica de extração de característica.

2.2.3.1 GANHO DE INFORMAÇÃO

O Ganho de Informação ou Information Gain (IG) é uma redução esperada na entropia causada pela divisão dos exemplos de acordo com um atributo qualquer *x*, na qual entropia é definida como o valor esperado de uma informação (HARRINGTON, 2012), considerando-se *z* o número de classes possíveis que uma informação pode assumir dada pela Equação 1:

$$H = \sum_{i=1}^{z} p(x_i) \log_2 p(x_i) \quad (1).$$

Ele mede o número de bits obtidos por meio da predição de uma classe através da presença ou falta de um termo em um documento. Seja *t* um n-grama, o ganho de informação de um termo é calculado como na Equação 2:

$$IG(t) = -\sum_{i=1}^{z} P(c_i) \log P(c_i) + P(t) \sum_{i=1}^{z} P(c_i|t) \log P(c_i|t) + P(\bar{t}) \sum_{i=1}^{z} P(c_i|\bar{t}) \log P(c_i|\bar{t}) \quad (2),$$

onde $P(c_i)$ denota a probabilidade de uma classe *i* ocorrer; P(t) é a probabilidade de um n-grama (atributo) *t* ocorrer; e $P(\bar{t})$ a é a probabilidade de um n-grama *t* não ocorrer (TAN e ZHANG, 2008).

Em análise de sentimentos, dado um conjunto de n-gramas de uma base de dados com opiniões, na qual duas classes (positivo ou negativo) existem, o IG para cada *token* é calculado com base na Equação 2. Para o problema de inferência de *ratings* com 5 classes, *i* varia de 1 a 5.

De acordo com a metodologia utilizada, apenas alguns n-gramas são utilizados para treinamento. Estes são escolhidos de acordo com a maior variação do ganho de informação, tanto para classes negativas quanto para classes positivas, isto é, palavras que expressam

[15] http://www.twitter.com

sentimento negativo, por exemplo, tem maior tendência a serem utilizadas em opiniões nas quais o autor não recomendaria um item.

2.2.3.2 GANHO MÉDIO DE INFORMAÇÃO

O Ganho Médio de Informação ou Gain Ratio (GR) aprimora o resultado do ganho de informação normalizando a contribuição de todas as características na decisão da classificação final para um documento. Na Equação 3, os valores de normalização ou *Split Information* são calculados por meio da informação obtida pela divisão de um documento de treinamento P em *v* partes, na qual *v* corresponde ao número de atributos (SHARMA, A.; DEY, 2012):

$$\text{SplitInfo(t)} = -\sum_{j=1}^{v} \frac{|P_j|}{|P|} \log \frac{|P_j|}{|P|} \text{ (3).}$$

Por fim, a Equação 4 define o ganho médio como:

$$\text{Gain Ratio(t)} = \text{Information Gain(t)/SplitInfo(t) (4).}$$

Assim como no IG, essa fórmula tem como objetivo selecionar palavras que possuem algum sentimento, seja ele positivo ou negativo, e os n-gramas com maior ganho médio são utilizados como atributos.

2.2.3.3 CHI-QUADRADO

O modelo Chi-quadrado ou Chi-square (CHI) consiste em retirar os n-gramas mais comuns ou os que sejam mais próximos de palavras como “bom” ou “ruim” de um texto. A partir disso, vetores podem ser criados com palavras separadas (unigramas), duas palavras (bigramas) ou n-gramas.

Ele representa a associação entre uma característica e a classe correspondente por meio da Equação 5:

$$\text{CHI}(t, c_i) = \frac{N*(AD-BE)^2}{(A+E)*(B+D)*(A+B)*(E+D)} \text{ and } \text{CHI}_{max} = \max_i(\text{CHI}(t, c_i)), \text{ (5),}$$

onde *t* é um n-grama e c_i a classe. A é o número de vezes que *t* e c_i ocorrem simultaneamente; B é o número de vezes que *t* ocorre sem c_i; E é o número de vezes que c_i ocorre sem *t*; D é o número de vezes que nem c_i nem *t* ocorrem e; N é o total de documentos (TAN e ZHANG, 2008).

Para cada classe, a associação entre um atributo e uma classe é calculada, entretanto, apenas o valor máximo CHI_{max} é utilizado, selecionando a classe com maior relação. Na análise textual, *t* é representado por um n-grama e *c* são as classes positivo ou negativo na

classificação binária ou são as classes referentes as estrelas presentes no problema de inferência de *ratings*.

2.2.4 VETORIZAÇÃO

Com os n-gramas selecionados pelos métodos de extração de características citados na Seção anterior, a próxima etapa consiste em transformar uma frase em um vetor de características, onde os atributos correspondem aos n-gramas selecionados. Estes atributos são configurados de acordo com frequência dos mesmos em relação a uma opinião. Como exemplo, podemos notar o texto no quadro abaixo.

Opinião: *Great Hotel, lovely staff, great location.8 of us stayed here for 2 nights on a hen party, hotel is close to all bars night clubs, shopping, would definitely stay here again.Hotel is clean and security is great, rooms are really nice and comfortable and have great tv, kitchenette is very handy*. *Rating*: 5.

Words (15): *great, lovely, worst, location, stay, close, shop, terrible, clean, security, nice, comfortable, handy, bad, good.*

Matriz de representação

4	1	0	1	2	1	1	0	1	1	1	1	0	0	**5**

Nesse caso, cada posição do vetor corresponde exclusivamente a uma palavra e seu valor é dado pela frequência em uma determinada opinião. As *words* utilizadas acima são apenas exemplos, mas em uma aplicação real, essas palavras são selecionadas pelas TEC's citadas na Seção 2.2.3.

Utilizando o exemplo acima, notamos que a palavra *great* está na primeira posição do vetor. Sua frequência é dada pelo número de vezes que a palavra aparece na frase, neste caso o número 4. A última posição do vetor corresponde à classe inicial (*rating*) da opinião. Toda a base de dados deve ser configurada seguindo este modelo a fim de criar um grande grupo de exemplos para o treinamento dos algoritmos de aprendizado de máquina supervisionados.

Além da frequência, outro valor pode ser utilizado para preencher cada posição do vetor. O modelo TF-IDF configura os vetores com um peso w_t para um termo *t* de acordo com a Equação 6:

$$w_t = f_t . idf_t = f_t . \log \frac{N}{df_t} \qquad (6),$$

onde f_t é o número de vezes que *t* ocorre em uma opinião *d;* idf_t é a frequência inversa em um documento do termo *t; N* é o total de opiniões e df_t é o número de opiniões que contém *t* (PALTOGLOU e THELWALL, 2010). Além do trabalho de Paltoglou e Thelwall, que testa várias variantes deste modelo, esta fórmula de representação apresenta bons resultados no trabalho de Martineau e Finin (MARTINEAU e FININ, 2009).

2.3 MODELOS E ALGORITMOS DE CLASSIFICAÇÃO

Com todo o processo de seleção de características finalizado, criando, por fim, arquivos com atributos quantitativos que são mais facilmente compreendidos e executados por algoritmos de aprendizado, a próxima Seção apresenta alguns dos principais modelos e algoritmos nativos utilizados em análise de sentimento para resolver problemas multiclasse. Além disso, ela apresenta dois métodos que utilizam uma forma de classificação binária: *one-versus-one* (OvO) e o *one-versus-all* (OvA), métodos conhecidos como multiclasse adaptado. Além desses, o algoritmo de divisões binárias Nested Dichotomies será apresentado no Capítulo 5. Os trabalhos que utilizam alguns desses algoritmos estão bem descritos em (LUNARDI; VITERBO; BERNARDINI, 2015) e no Capítulo 3 dessa dissertação.

2.3.1 NAIVE BAYES

O algoritmo Naive Bayes é uma variação da teoria de decisão Bayesiana. A probabilidade Bayesiana habilita o conhecimento inicial e a lógica a serem aplicados em declarações desconhecidas (HARRINGTON, 2012). Formalmente, pode-se calcular a probabilidade condicional como na Equação 7:

$$P\,(c \mid d) = \frac{P(c)P(d \mid c)}{P(d)} \qquad (7)$$

Uma variação a teoria bayesiana, o modelo multinomial captura a frequência de uma palavra no conjunto de opiniões (MCCALLUM e NIGAM, 1998). Para associar a um novo exemplo *t* uma classe c_i, a classe com maior probabilidade $c^* = \operatorname{argmax} P(c_i|t)$ é considerada. Na Equação 8 é mostrado como o cálculo das probabilidades para cada classe $c_i \in c$ é realizado.

$$P_{NB}(c_i \mid t) = P(c_i)(\prod_{j=1}^{D} P(t_j|c_i) \qquad (8),$$

no qual *t* é um termo, *i* é o número da classe e *D* é o conjunto de opiniões.

A partir de um conjunto de termos *t* de uma opinião, representado pelo vetor *w*, a distribuição das probabilidades é dada pela Equação 9:

$$P_{NB}(c_i \mid w) = P_{NB}(c_i \mid t_1 .. t_n) = \frac{P(w|c_i)P(c_i)}{P(w)} \qquad (9),$$

no qual *n* é o número de termos em um vetor *w*.

Um dos trabalhos iniciais de análise de sentimentos (PANG; LEE; VAITHYANATHAN, 2002) utiliza, além do SVM e da Máxima Entropia, o Naive Bayes já que este demonstrava bons resultados no problema de categorização de textos. Apesar de simples, o Naive Bayes apresentou bons resultados, superando os outros algoritmos quando treinado com unigramas. Para o problema de multiclasse, Long *et. al* (LONG; ZHANG; ZHUT, 2010) utilizam tanto um modelo de regressão quanto o Naive Bayes, sendo estes treinados com características retiradas de opiniões com uma técnica baseada na complexidade Kolmogorov. Assim como em (PANG; LEE; VAITHYANATHAN, 2002), o resultado é satisfatório, chegando a atingir cerca de 12,5% de melhoria de desempenho com os classificadores testados em relação aos trabalhos anteriores.

2.3.2 SVM

Dado um conjunto de dados linearmente separável, caso exista uma linha em um plano que possa separar o conjunto de dados, a linha é chamada de hiperplano separador. A ideia é encontrar o hiperplano que esteja o mais próximo possível dos pontos, sendo que esses pontos estejam o mais distante possível do hiperplano a fim de garantir a melhor robustez do classificador. Isso é chamado de margem. Os pontos mais próximos da margem são chamados de vetores de suporte (HARRINGTON, 2012), como pode ser visto na Figura 3.

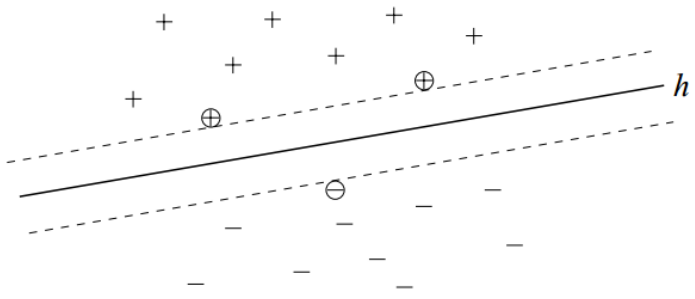


Figura 3. Hiperplano *h* encontrado, separando dados de treinamento positivos e negativos. Dados circulados são vetores de suporte (JOACHIMS, 1998)

A ideia principal do modelo de máquina de vetores de suporte é encontrar as margens ótimas em relação a um hiperplano separador *h*. Essa distância é calculada pela fórmula *u* =

$\vec{w}.\vec{x} - b$, no qual $\vec{w}$ é o vetor normal para o hiperplano, $\vec{x}$ é o vetor de entrada e b é uma constante.

Para o caso linear, a margem é definida pela distância do hiperplano em relação ao vizinho mais próximo dos exemplos positivos e negativos. Maximizar esta margem pode ser expresso por um problema de otimização, no qual a maximização $\frac{2}{||\vec{w}||^2}$ é equivalente a minimizar o problema, conforme a Equação 10:

$$L(w) = \frac{||\vec{w}||^2}{2} \quad (10),$$

sujeito a $y_n(\vec{w}.\vec{x} - b) \geq 1$, $\forall n$ no qual x_n é o *n*-ésimo exemplo de treinamento e y_n é a saída correta do SVM para o *n*-ésimo exemplo de treinamento.

Para problemas com a margem suave, variáveis de relaxamento são utilizadas para flexibilizar as restrições do problema de otimização descrito na Equação 910Essas variáveis ξ medem o local de uma amostra em relação as margens. Nesse caso, a Equação 10 fica sujeita a $y_n(\vec{w}.\vec{x} - b) \geq 1 - \xi$.

Para problemas não lineares, nem sempre é possível encontrar um hiperplano H para o problema. Para esse tipo de problema, é preciso encontrar uma transformação φ(x) que não seja linear, de acordo com a Equação 11:

$$\varphi(x) = \varphi_1(x), \ldots, \varphi_m(x) \quad (11),$$

dado *m* o número de dimensões do problema. Nesse caso, os padrões $\vec{x}$ passam a ser linearmente separáveis e o SVM fica sujeito as restrições $y_n(\vec{w}.\varphi(\vec{x}) - b) \geq 1$. Para um conjunto de *n* padrões $\varphi(\vec{x}_n)$, multiplicadores de Lagrange podem ser utilizados. A solução depende apenas do produto $\varphi(\vec{x}_i)\varphi(\vec{x}_j)$, que pode ser obtido por meio de funções conhecidas como Kernels, como o polinomial exibido na Equação 12:

$$K(\vec{x}_i, \vec{x}_j) = (\delta(\vec{x}_i.\vec{x}_j) + k)^d \quad (12).$$

Para problemas multiclasses, são necessários vários classificadores binários que podem ser construídos por meio de técnicas adaptadas, descritas na Seção 2.3.5. Em muitos trabalhos, algumas variantes deste modelo são utilizadas. Isso pode ser notado em (BROOKE, 2009) e em (PANG e LEE, 2005), no qual o algoritmo Sequential Minimal Optimization (SMO) é o mais indicado para resolver o problema de análise de sentimentos multiclasse já que ele é utilizado para resolver problemas de regressão a partir do SVM. O SMO divide o problema de programação quadrática (PQ) existente no SVM simples, criando soluções menores para o problema de PQ sem utilizar uma matriz de armazenamento extra (PLATT,

1998). Além do SMO, variações do pacote LibSVM[16] com a função linear sendo utilizada podem ser adaptadas para a classificação multiclasse. Isto se deve ao fato de que, segundo (DUMAIS *et al.*, 1998) e (KAESTNER, 2013), o modelo linear é o mais adequado para análise de texto.

Assim como dito em (LIU, 2012), o problema de inferência de *rating* também pode ser considerado um problema de regressão. Isso faz com que variantes do SVM, como o SMO, estejam presentes em trabalhos como (PANG e LEE, 2005), (LONG; ZHANG; ZHUT, 2010) e (DE ALBORNOZ *et al.*, 2011), trabalhos estes que possuem bons resultados e são referências na área de inferência de *rating* ou classificação multiclasse.

2.3.3 kNN

Os k-vizinhos mais Próximos ou k-Nearest Neighbors (kNN) é um método baseado em instâncias que aprende com o simples armazenamento dos dados de treinamento. Quando uma nova instância surge, ele recupera os dados armazenados e classifica essa nova instância (MITCHELL, 1997). A partir dos k vizinhos mais parecidos, ele escolhe o dado com os k mais similares com o que será classificado e atribui uma nova classe a ele (HARRINGTON, 2012). A proximidade dos vizinhos pode ser definida, por exemplo, de acordo com a distância Euclidiana (MITCHELL, 1997) demonstrada na Equação 13 para dois vizinhos:

$$u = \sqrt{(xA_0 - xB_0)^2 + (xA_1 - xB_1)^2} \qquad (13).$$

Em Tan e Zhang (TAN e ZHANG, 2008), considerando *d* um documento de teste, a tarefa está em encontrar os *k* vizinhos entre os outros documentos de treinamento. Na Equação 14, a similaridade *sim*() entre o item *d* e os outros vizinhos é usada como o peso das classes dos documentos mais próximos, calculado como:

$$\text{score}(d, c_i) = \sum_{d_j \in KNN(d)} sim(d, d_j)\delta(d_j, c_i) \qquad (14),$$

no qual KNN(*d*) representa o conjunto de vizinhos do documento *d* e c_i uma classe. A função sim(d, d_j) representa a similaridade entre um documento *d* o documento de treino d_j. Se d_j pertence a c_i, $\delta(d_j, c_i)$ é igual a 1, senão, é igual a 0. Logo, o documento *d* deve pertencer à classe que ele possui o maior *score*.

Além de Tan e Zhang, entre os trabalhos relacionados, apenas em Sharma e Dey (SHARMA e DEY, 2012) o kNN é utilizado para o problema de classificação de sentimentos binária. Em ambos trabalhos, o kNN apresenta resultado bem inferior quando comparado com

[16] https://www.csie.ntu.edu.tw/~cjlin/libsvm

os algoritmos SVM e Naive Bayes. Para o RIP, em nenhum dos trabalhos citados nesta pesquisa foi utilizado este algoritmo, o que serviu como motivação para avaliar o desempenho do mesmo neste trabalho. Na ferramenta utilizada nesta dissertação, o algoritmo kNN é conhecido como IBk (Instance-Based Learning with Paramater k).

2.3.4 ÁRVORES DE DECISÃO

As árvores de decisão são um dos principais métodos de inferência indutiva utilizadas. Elas consistem em um método de aproximação discreta do alvo, na qual a função de aprendizado é representada por uma árvore de decisão, que podem ser representadas como um conjunto de regras *if-then* (MITCHELL, 1997).

A tarefa de construir uma árvore de indução consiste em criar uma regra de classificação que pode determinar a classe de um exemplo de treinamento a partir dos valores dos seus atributos. Essa regra de classificação pode ser expressa por meio de uma árvore de decisões. As folhas de uma árvore são as classes existentes do problema e os nós internos são os atributos escolhidos no treinamento. A classificação de um novo exemplo começa na raiz e para cada atributo uma decisão é tomada a fim de chegar em um novo atributo. Esse processo continua até que a classe apropriada seja encontrada (QUINLAN, 1986).

O algoritmo ID3, uma implementação de uma árvore de decisões, foi criado para problema nos quais existem muitos atributos e o conjunto de treinamento possui vários objetos. A ideia básica deste algoritmo é iterativa. Um subconjunto de treinamento é escolhido aleatoriamente, no qual um atributo é escolhido a fim de representar a melhor divisão entre os dados de treinamento. A partir disso, uma árvore é criada a partir do atributo do subconjunto.

Analisando um problema binário, para escolher um atributo que divida a árvore, o objetivo é escolher o atributo que possui o maior ganho (*gain*). Esse ganho é definido por meio da redução da entropia, dada pela Equação 15:

$$\text{Entropia(P)} = (-(P_+ \log_2 P_+) - (P_- \log_2 P_-)) \qquad (15),$$

sendo P o conjunto de treinamento, P_+ é a fração dos exemplos positivos de treinamento e P_- a fração dos exemplos negativos. Com a entropia definida, o ganho é dado pela Equação 16:

$$\text{Ganho(P,}t\text{)} = \text{Entropia (P)} - \sum_j^{valores(n)} \frac{P_j}{P} Entropia(P_j) \qquad (16),$$

no qual P é o conjunto de treinamento, *t* um atributo qualquer e *n* é o conjunto de termos.

O restante dos exemplos de treinamento é classificado por meio da árvore inicial. Se o restante do conjunto for corretamente classificado, o processo de construção é finalizado. Senão, um conjunto de exemplos que não foram corretamente classificados são adicionados ao subconjunto inicial e uma nova árvore é criada. Esse processo pode ser finalizado encontrando uma árvore classifique todos os dados de treinamento corretamente. Além do ID3, as variações C4.5 e J48 são utilizadas em análise de sentimentos.

Para o problema de análise de sentimentos multiclasse, os atributos dos nós internos são as características de treinamento selecionados pelos métodos de seleção da Seção 2.2.3 e as folhas representam as classes (para o problema de inferência de *ratings*, cada folha é o valor numérico das estrelas). Nesse caso, a entropia é calculada por meio da Equação 17:

$$\text{Entropia(P)} = \sum_{i=1}^{c} - P_i \log_2 P_i \qquad (17).$$

Na pesquisa de Albornoz *et al.* (DE ALBORNOZ *et al.*, 2011), um modelo de árvore (Functional Tree - FT) é utilizado a fim avaliar o vetor de intensidade de características proposto pelos autores, juntamente com o LibSVM e o algoritmo Logistic. Eles utilizam estes algoritmos com o intuito de prever o *rating* final de uma opinião, atingindo a acurácia de 43,7% para o modelo FT. Entre todos os algoritmos utilizados, a FT obteve o pior desempenho, sendo 3,2% inferior ao modelo Logistic (46,9%).

Chen *et al.* (CHEN *et al.*, 2006) criam uma análise visual de opiniões sobre o livro *O Código da Vinci* inspirados em um modelo semelhante a uma árvore de decisões. Além disso, eles utilizam os algoritmos C4.5, SVM e Naive Bayes a fim de selecionar bons termos para a categorização das opiniões utilizadas.

2.3.5 MODELOS MULTICLASSE ADAPTADOS

Nessa Seção, dois dos principais métodos para resolver problemas multiclasse por meio de divisões binárias são descritos: o *One-vs-One* (OvO) e o *One-vs-All* (OvA). O método OvA cria *n* divisões, na qual cada etapa do aprendizado é feito comparando um a classe a todas as outras classes. No modelo OvO, cada classe c_i é comparada com outra classe c_k, onde k, $i = 1..n$ e $i \neq k$, dado que n é o número de classes (HSU e LIN, 2002).

Em um modelo OvA, a partir da escolha de um classificador (SVM, por exemplo), *n* classificadores são construídos na etapa de treinamento, isto é, para cada comparação entre uma classe e as demais, um classificador é construído. Para o *i*-ésimo classificador, os

exemplos positivos são todos os pontos da classe c_i e os exemplos negativos são todos os pontos que não estão na classe c_i.

Na fase de classificação, um novo exemplo é rotulado por todos os classificadores f_i. Após a predição, na qual f_i seja o *i*-ésimo classificador, a classificação é dada por meio da Equação 18:

$$f(x) = \arg\max_i f_i(x) \quad (18),$$

isto é, a classe final é obtida selecionando a classe com maior número de votos. Um processo de votação que pode ser utilizado é o de votação distribuída (PIMENTA, 2004). Por exemplo, para um classificador que utilize a classe c_i contra todas, se uma classe predita for igual a *i*, a classe c_i recebe um ponto. Caso contrário, cada classe c_k recebe um valor dado como $\frac{1}{(i-1)}$.

Para o classificador OvO, um modelo classificador também é escolhido, entretanto, cada classe *j* é comparada com outra classe *i.* Seja f_{ij} o classificador no qual as classes i são exemplos positivos e as classes *j* são exemplo negativos. Assumindo que $f_{ij} = -f_{ij}$, a classificação será feita por meio da Equação 19:

$$f(x) = \arg\max_i \left(\sum_j f_{ij}(x)\right). \quad (19).$$

Nesse caso, para decidir qual o melhor classificador, uma estratégia de votação direta pode ser utilizada (HSU e LIN, 2002). Se a função f_{ij} diz que *x* está em uma classe *i*, um voto para a *i*-ésima classe é computado. Caso contrário, um voto é acrescentado para a *j*-ésima classe. Por fim, a classe de *x* é definida pela classe com maior número de votos.

Em relação ao número de divisões binárias necessárias em cada um desses métodos, no classificador *one-vs-all* ele é dado por *i*, onde *i=n*, sendo *n* o número de classes. Já para o algoritmo *one-vs-one*, o número de etapas para a classificação é dado por $\frac{n(n-1)}{2}$, onde *n* é o número de classes (ALY, 2005). Para um problema com 4 classes, o modelo OvA é configurado com 4 classificadores. Já para o modelo OvO, com as mesmas 4 classes, 6 classificadores são construídos. Para o problema multiclasse, os algoritmos SVM citados na Seção 2.3.2 (SMO e LibSVM) utilizam o modelo OvO.

2.4 MEDIDAS AVALIATIVAS

Para medir o desempenho dos algoritmos e técnicas citados anteriormente, as medidas avaliativas citadas abaixo foram utilizadas, baseando-se na matriz de confusão. Essas medidas são as mais utilizadas em outros trabalhos como (PANG; LEE; VAITHYANATHAN, 2002),

(TAN e ZHANG, 2008), (GOLDBERG e ZHU, 2006) e (GO; BHAYANI; HUANG, 2009), seja para a análise multiclasse ou binária.

Uma matriz de confusão para um problema *n*-classes é uma matriz *n* x *n* (GODBOLE e SARAWAGI, 2004) onde o elemento M_{ij} é, para *i=j*, o número de opiniões pertencentes a uma classe *i* que foram corretamente classificadas e, para *i≠j*, o número de opiniões de uma classe *i* que foram erroneamente classificadas em outra classe *j*. Na Tabela 3 é apresentado um exemplo de uma matriz que foi criada a partir dos testes realizados nesse trabalho e exibidos nos Capítulos Capítulo 4, Capítulo 6 e Capítulo 7 em que as letras *a-e* correspondem à escala de *ratings* utilizada (1-5 estrelas), respectivamente.

2.4.1 ACURÁCIA, PRECISÃO E RECALL

Com base na matriz de confusão, as medidas descritas abaixo são muito utilizadas a fim de medir o desempenho da precisão dos algoritmos, principalmente a acurácia do modelo que mede quão próximo *T* é para *S*, onde *T* o conjunto inicial e *S* o conjunto com as predições criadas para uma base de dados.

Tabela 3. Exemplo de uma matriz de confusão para o problema 5-classes

a	b	c	d	e	Total	← classificado como
1140	276	61	10	13	1500	a=1
497	502	380	96	25	1500	b=2
132	260	773	281	54	1500	c=3
47	97	228	648	480	1500	d=4
19	36	45	267	1133	1500	e=5
1835	**1171**	**1487**	**1302**	**1705**	7500	

A acurácia é calculada como:

$$\text{A} = \frac{TP+TN}{P+N} = \frac{número\ de\ exemplos\ classificados\ corretamente}{total\ de\ exemplos},$$

no qual TP (*true-positive*) e TN (*true-negative*) são os exemplos classificados corretamente (JAPKOWICZ e SHAH, 2011). Analisando a Tabela 3, a acurácia final é dada pelo número de exemplos corretamente classificados (1140+502+773+648+1133) dividido pelo número total de exemplos (7500). Desta forma, a acurácia é dada por 0,5594.

A precisão é dada por:

$$\text{P} = \frac{TP}{TP+FP} = \frac{número\ de\ corretas\ predições\ positivas}{número\ de\ predições\ positivas},$$

no qual FP (*false-positive*) são exemplos classificados como uma classe c_i mas que no treinamento eram de outra classe c_j. A precisão para a classe *a* é dada pelo número de corretas

predições positivas (1140) dividido pelo número de predições positivas (1835), isto é, o número de objetos classificados como *a* e que inicialmente eram rotulados como *a* dividido pelo número de exemplos classificados como *a*, sejam eles inicialmente *a* ou não. Desta forma, a precisão é dada pelo valor 0,6212.

O *recall* é dado pela seguinte fórmula:

$$\mathrm{R} = \frac{TP}{TP+FN} = \frac{número\ de\ corretas\ predições\ positivas}{número\ de\ exemplos\ positivos},$$

no qual TP (*true-positive*) + FN (*false-negative*) são os exemplos positivos. O *recall* para a classe *a* é dada pelo número de corretas predições positivas (1140) dividido pelo número de exemplos positivos (1500), isto é, o número de objetos classificados como *a* e que inicialmente eram rotulados como *a* dividido pelo número de exemplos inicialmente rotulados como *a*. Desta forma, a precisão é dada pelo valor 0,76.

2.4.2 ACURÁCIA APROXIMADA

O cálculo da acurácia aproximada é definido por Brooke (BROOKE, 2009), e esta medida considera aceitável quando uma opinião é classificada com a classe exata ou com a(s) classe(s) vizinhas à classe exata, considerando a escala de *ratings* (1 a 5). A Tabela 4 estende a Tabela 3 para incluir os valores da acurácia aproximada para cada classe. Analisando a classe *b*, tanto opiniões classificadas como *a* ou *c* são aceitáveis e as opiniões inicialmente rotuladas como *b* e classificadas como *d* e *e* são consideradas como erro. Desta forma, se notarmos a acurácia exata da classe *b* (0,335) e a acurácia próxima, concluímos que muitas opiniões da classe *b* foram classificadas como *a* (497) ou *c* (380). Logicamente, o valor da acurácia aproximada é sempre mais elevado do que a acurácia exata.

Tabela 4. Matriz de confusão com acurácia exata e próxima

a	b	c	d	e	Acurácia		← classificado como
					Exata	Aproximada	
1140	276	61	10	13	0,76	0,944	a=1
497	502	380	96	25	0,335	0,919	b=2
132	260	773	281	54	0,515	0,876	c=3
47	97	228	648	480	0,432	0,904	d=4
19	36	45	267	1133	0,755	0,933	e=5
1835	**1171**	**1487**	**1302**	**1705**	**0,559**	**0,915**	-

Essa medida avaliativa também foi utilizada em (PALTOGLOU e THELWALL, 2013), no qual os autores consideram não só a acurácia e o erro quadrático médio, mas

também exibem o valor do erro absoluto médio e a acurácia aproximada, onde a distância máxima analisada é de uma classe para a classe correta.

2.5 DIFICULDADES DA ANÁLISE DE SENTIMENTOS

A grande maioria dos trabalhos de mineração de opiniões existentes tem como foco a mineração de opiniões no idioma inglês. Embora raros, trabalhos como o de Ortigosa *et al.* (ORTIGOSA; MARTÍN; CARRO, 2014) e Tang e Zhang (TAN e ZHANG, 2008) utilizam os idiomas espanhol e mandarim, respectivamente. Essa falta de trabalhos em alguns idiomas pode dificultar a análise já que os idiomas têm processos de construção diferentes.

Em relação ao pré-processamento textual, nota-se uma dificuldade na escolha de palavras para treinamento. Isso porque a forma de escrever pode mudar para cada pessoa. Uma expressão que possa indicar um sentimento muito bom para uma pessoa, pode indicar um sentimento nem sempre bom para outra (LIU, 2012). Da mesma forma, uma palavra pode ser utilizada em qualquer classe, logo, o contexto deve ser analisado para compreender o sentimento da mesma. Um exemplo disso são as ironias, muito presentes em avaliações políticas, por exemplo.

Além disso, outra dificuldade está na análise de opiniões que apresentam poucas palavras ou expressões que indiquem algum sentimento. Esse é um caso estudado principalmente em *tweets*. Isso se deve ao fato de um *tweet* ter o número de caracteres limitado a 140. Em alguns casos, como feito em (GO; BHAYANI; HUANG, 2009), a utilização de *emoticons* no treinamento é uma opção a fim de melhorar o desempenho da mineração de opiniões. Outro problema está na existência de *herding effects* (WANG e WANG, 2014) para a RIP. Isso se deve ao fato de muitas vezes o *rating* final de um usuário não condizer com o comentário. Isso pode acontecer, por exemplo, pelo fato do *rating* ser baseado na média de notas existente no site e não avaliado em relação à opinião por si só. Muitas vezes, pode-se notar que um comentário possui uma avaliação que poderia ter nota máxima (5 estrelas) mas tem o *rating* 4, por exemplo.

CAPÍTULO 3 – TRABALHOS RELACIONADOS

Como foi discutido no Capítulo anterior, os principais métodos de análise de sentimentos podem ser divididos em quatro grandes áreas, de acordo com um modelo semelhante ao de (CAMBRIA *et al.*, 2013):

- afinidade léxica;
- aprendizado de máquina;
- orientação semântica, e;
- conceitos ou ontologias.

O aprendizado de máquina ou métodos estatísticos consistem na utilização de algoritmos como Naive Bayes e Máquina de Vetores de Suporte a fim de treinar um corpo textual e, a partir do treinamento, classificar novas opiniões. Esses métodos foram anteriormente abordados no Capítulo 2, já que este trabalho tem como foco a proposta de uma técnica que utilize estes algoritmos na análise de sentimentos. Desta forma, esta Seção apresenta uma discussão dos principais trabalhos que utilizam algoritmos de classificação para a análise de sentimentos.

Estes trabalhos foram escolhidos com base na importância dos mesmos para a área de análise de sentimentos, levando em consideração os resultados obtidos e as técnicas de extração e algoritmos utilizados. Em alguns casos, algumas destas técnicas e algoritmos não foram citados no Capítulo anterior devido ao grande número de técnicas disponíveis, sendo inviável que todas sejam descritas. Em relação às formas de utilização das opiniões, os principais trabalhos se distribuem em três campos que merecem destaque:

- a classificação em relação à objetividade ou subjetividade;
- a classificação binária, e;
- a classificação multiclasse.

Esses campos serão descritos nas seções abaixo, com destaque para os trabalhos de classificação binária e multiclasse.

3.1 CLASSIFICAÇÃO EM TEXTO OBJETIVO OU SUBJETIVO

A primeira etapa para realizar a classificação de textos é saber se eles são subjetivos, isto é, contém algum tipo de opinião em relação a uma entidade. Desta forma, tendo uma base de dados que não garanta que existam apenas textos com opiniões subjetivas, uma primeira

etapa a ser realizada no processo de análise de sentimentos deve ser separar tais textos em relação à objetividade ou subjetividade. Wiebe e Rillof (WIEBE e RILLOF, 2005) desenvolveram um classificador subjetivo usando textos não rotulados para o treinamento. A pesquisa inicia com um processo de busca que utiliza um dicionário de palavras subjetivas para criar os dados de treinamento automaticamente. Esses dados são utilizados para criar um modelo de extração de características e um classificador probabilístico. Finalmente, eles adicionam um mecanismo de autotreinamento que providencia um auxílio aos classificadores, enquanto eles ainda dependem de dados não anotados.

Yu e Hatzivassiloglou (YU e HATZIVASSILOGLOU, 2003) utilizaram a similaridade entre sentenças e um classificador Naive Bayes para classificar um texto como subjetivo ou objetivo, baseando-se na afirmativa de que opiniões são mais similares a outras opiniões do que a textos factuais. Eles utilizaram um sistema chamado SIMFINDER para medir a similaridade entre as palavras e frases utilizadas nas diversas sentenças de treinamento. Para realizar a classificação final (objetivo ou subjetivo), os autores utilizaram técnicas de extração como n-gramas, marcadores POS e palavras que possuam algum sentimento. Além disso, a proposta também realizou a classificação binária de uma sentença classificada como subjetiva.

3.2 CLASSIFICAÇÃO BINÁRIA

Muitos dos trabalhos existentes na área de análise de sentimentos têm como principal objetivo avaliar o desempenho de um ou mais algoritmos de aprendizado, comparando o resultado final, seja por meio da acurácia, tempo ou outras medidas avaliativas. Para isso, são utilizadas bases de dados com avaliações disponíveis na web, com o intuito de avaliar os melhores algoritmos e as melhores técnicas de extração de características. O principal objetivo destes estudos é a classificação em relação à polaridade de uma opinião, isto é, saber se ela é negativa ou positiva; boa ou ruim; recomendada ou não recomendada.

Essa Seção discute a grande maioria dos trabalhos referenciados em nossa pesquisa, muitos das quais serviram de base para a metodologia utilizada. Isso se deve ao fato de o problema de análise de sentimentos ser geralmente considerado como um problema de classificação binária (LIU, 2012)

Pang *et al.* (PANG; LEE; VAITHYANATHAN, 2002) tinham como principal objetivo determinar se uma avaliação é positiva ou negativa utilizando algoritmos de aprendizado. Os

autores compararam o desempenho destes algoritmos no problema de mineração de opiniões com o desempenho na classificação feita por humanos e na categorização baseada em tópicos. Os autores mostraram que os algoritmos são melhores na classificação do que humanos, mas seu desempenho não é melhor do que tradicionais métodos de categorização baseado em tópicos (classificação por assunto). Eles utilizaram uma base de dados de avaliações de filmes e pediram para que dois estudantes criassem uma seleção de palavras que indicavam a positividade ou negatividade de uma avaliação. Baseado nessa lista, eles criaram novos vetores de palavras que serão utilizadas pelos algoritmos Naive Bayes, SVM e Máxima Entropia. O desempenho alcançado foi melhor do que as bases formadas por humanos, mas em relação à acurácia de 90% da categorização baseada em tópicos, nenhum dos algoritmos, mesmo quando combinados com bigramas, POS ou a posição de um n-grama no texto conseguiu atingir tal desempenho. O melhor classificador foi o SVM, enquanto a utilização de unigramas mostrou-se mais efetiva em relação às características.

Kang *et al.* (KANG; YOO; HAN, 2012) propuseram um novo método para a análise de sentimentos de opiniões sobre restaurantes apresentando duas melhorias no algoritmo Naive Bayes a fim de resolver o problema de balanceamento das acurácias das classificações positivas e negativas. Eles combinaram técnicas de unigramas e bigramas (que incluem tratamento de palavras negativas e utilização de advérbios intensivos) com o algoritmo SVM, o Naive Bayes e as melhorias do Naive Bayes propostas pelos autores. Os autores demonstraram que o Naive Bayes proposto, quando implementado usando bigramas e unigramas, diminui a distância entre a acurácia positiva e a acurácia negativa para 3.6% comparada ao Naive Bayes original e em até 28% em relação ao SVM para opiniões sobre restaurantes.

Xia *et al.* (XIA; ZONG; LI, 2011) fizeram um estudo sobre a efetividade do agrupamento de técnicas para tarefas de classificação binária, focando no agrupamento de conjuntos de características e algoritmos de classificação. Eles projetam dois esquemas utilizando POS e dependência sintática e, para cada esquema, utilizam NB, SVM e a Entropia Máxima (MaxEnt) para a classificação, utilizando a base de dados de filmes disponíveis em Cornell[17] e o Multi-Domain Sentiment Dataset[18] com avaliações sobre produtos da *Amazon*™.

[17] Disponível em www.cs.cornell.edu/people/pabo/movie-review-data/.
[18] Disponível em www.cs.jhu.edu/~mdredze/datasets/sentiment/

Tan e Zhang (TAN e ZHANG, 2008) fizeram um trabalho que apresenta um estudo sobre análise de sentimentos que não usa a língua inglesa, mas sim a chinesa. Os autores utilizam quatro métodos de seleção de características (Informação Mútua, IG, DF e CHI) e cinco algoritmos de aprendizado de máquina (kNN, Naive Bayes, SVM, Winnow e o classificador centroide, estes dois últimos não citados nesta dissertação) em uma base de dados que contém opiniões sobre três domínios: educação, filmes e eletrodomésticos. Considerando todos os algoritmos de aprendizado, o melhor método de seleção de característica é o Ganho de Informação, que atinge uma média de 88.6% de acurácia. Considerando os métodos de seleção de características, em relação aos algoritmos de aprendizado, o SVM produz a melhor acurácia: 86.8%. Em um dos testes, os autores realizaram o treinamento do SVM em um domínio de eletrodomésticos e utilizaram o conhecimento adquirido para classificar opiniões no domínio de educação. Os autores surpreendentemente obtiveram 0,899 para o valor do MacroF1 para o SVM treinado, ilustrando a possibilidade do uso de modelos treinados em um domínio serem utilizado em outros.

Matsumoto *et al.* (MATSUMOTO; TAKAMURA; OKUMURA, 2005) analisaram o desempenho do SVM para realizar a classificação binária de avaliações sobre filmes, utilizando dois conjuntos de dados. Os autores extraíram unigramas, bigramas, frequentes subsequências de palavras e subárvores dependentes, e usaram tais características para o treinamento de um classificador SVM. Entre os vários testes, eles atingiram 88.3% de acurácia para a primeira base de dados utilizando bigramas, unigramas e árvores de dependência, e 93.7% para o segundo conjunto, utilizando o SVM com bigramas, unigramas, palavras subsequentes e árvores de dependência.

Paltoglou and Thelwall (PALTOGLOU e THELWALL, 2010) mostraram que funções de peso adaptadas da Recuperação de Informação (RI) baseadas no cálculo da *tf.idf* [25] e adaptadas para uma configuração particular da análise de sentimentos podem aumentar significativamente o desempenho da classificação. Os autores mostraram que a utilização do SVM adaptado como algoritmo de aprendizado e com essas funções de peso no processo de vetorização, os resultados atingiram até 96% de acurácia. Esse resultado está entre os melhores desempenhos entre os trabalhos relacionados para classificação binária utilizando um algoritmo de aprendizado.

Sharma e Dey (SHARMA e DEY, 2012) exploraram cinco métodos de seleção de características em mineração de dados e sete algoritmos de aprendizado de máquina para análise de sentimento em um conjunto de avaliações on-line de filmes. Entre os melhores resultados, o método GR, uma variação de IG, foi o que apresentou os melhores resultados. Já em relação aos algoritmos de aprendizado, o SVM possuiu a melhor média de desempenho, considerando as cinco estratégias de seleção, mas o melhor resultado é apresentado pelo Naive Bayes atingindo 90,9% com GR.

Como pode ser observado, muitas das aplicações exploraram novas configurações e novos métodos para melhorar o desempenho dos algoritmos de aprendizado. Xia *et al.* (XIA; ZONG; LI, 2011) exploraram métodos agrupados: regras fixas e métodos treinados a fim de melhorar o desempenho dos algoritmos de aprendizado. Sharma and Dey (SHARMA e DEY, 2012) fizeram um estudo sobre vários métodos de seleção de características e algoritmos de aprendizado. Paltoglou and Thelwall (PALTOGLOU e THELWALL, 2010) utilizaram várias variações do inverso da frequência e atingem acurácia superior a 95%.

Pode ser notado também que alguns trabalhos utilizaram diversos algoritmos de aprendizado, combinados com diversas TEC's, mostrando que em muitos trabalhos houve algum tipo de comparação a fim de obter o melhor algoritmo para o(s) domínio(s) em estudo. Entre as principais TEC's destacaram-se as que analisaram termos e sua frequência em uma opinião. Entre os principais métodos de análise textual estão DF, IG, CHI, unigramas e n-gramas.

Os unigramas e n-gramas foram usados juntamente com outra técnica de extração em algumas pesquisas, com o intuito de selecionar os n-gramas mais importantes e calcular a frequência dos mesmos. Por exemplo, em (PAK e PAROUBEK, 2010) foram usados n-gramas para representar palavras que foram obtidas através da análise da frequência de tais palavras chaves, além de marcadores POS. Em (PANG; LEE; VAITHYANATHAN, 2002) foram utilizados unigramas, bigramas, POS e adjetivos, considerando em alguns casos a frequência, e em outras a presença de uma palavra. Em [19] foram testados cinco TEC's e sete algoritmos de aprendizado.

Embora exista um grande número de TEC's, essa dissertação considera apenas os algoritmos e as TEC's mais utilizados, que foram descritos no Capítulo 2. Esses métodos geralmente apresentaram bons resultados em outros trabalhos relacionados e aparecem em

trabalhos de grande importância na área de análise de sentimentos utilizando algoritmos de aprendizado. Um resumo de todos esses trabalhos está presente na Tabela 5.

Tabela 5. Resumo dos principais trabalhos em análise de sentimento binária

Autores	Domínio	Seleção de Características	Algoritmos	Acurácia (%)
Pang *et al.* 2002	Filmes	POS, unigramas, bigramas, posição, adjetivos	NB, MaxEnt e SVM	82.9 (SVM + unigramas)
Mak *et al* 2003	Filmes	IG e DF	Decision Tree, kNN e NB	65 (DT + DF)
Matsumoto *et al.* 2005	Filmes	Unigramas, bigramas, frequentes subsequências de palavras e sub-árvores dependentes	SVM	93.7 (SVM + unigramas + bigramas, frequentes subsequências de palavras
Tan e Zhang 2008	Educação, filmes e eletrodomésticos	IG, DF and CHI	Classificador centroide, kNN, NB, Winnow e o SVM	*90.6 (SVM + IG) – Medida Macro F1
Go *et al.* 2009	*Tweets*	Palavras com sentimento, bigramas and unigramas	NB, MaxEnt e SVM	83.0 (MaxEnt com unigramas + bigramas)
Paltoglou e Thelwall 2010	Filmes	Unigramas e DF – variantes do *tfidf*	SVM	96.9 (SVM + BM25 *tf + variante* BM25 delta *idf*)[b]
Kang *et al.* 2011	Restaurantes	Unigramas and bigramas	NB, SVM e NB adaptado	81.2 (NB adaptado + unigramas + brigramas)
Xia *et al.*, 2011	Livros, eletrônicos, DVD's e artigos de cozinha	POS and dependência sintática (Word Relation - WR)	NB, SVM e MaxEnt	Filme – 86.85 (MaxEnt + POS) Cozinha – 88.65 (NB + WR)
Sharma e Dey 2012	Filmes	IG, GR, MI, CHI e Belief	NB, SVM, MaxEnt, DT, kNN, Adaboost e Winnow	90.9 (NB + GR)
Ortigosa *et al.* 2014	*Posts* no Facebook	Classificação léxica	J48, NB e SVM	83.27 (SVM + classificador léxico)

3.3 CLASSIFICAÇÃO MULTICLASSE

Os problemas de classificação multiclasse agregam trabalhos que analisam problemas que podem ser divididos em 3 ou mais classes. O problema de inferência de *ratings* é considerado um problema multiclasse, seja em uma escala com 3, 4, 5 ou mais estrelas. Esses problemas também são conhecidos como problemas de escala de multiponto (PANG; LEE, 2008).

Em (PANG e LEE, 2005), os autores avaliaram a acurácia de humanos em relação à tarefa de determinar o *rating* de um comentário e, posteriormente, eles aplicaram um algoritmo baseado em *metric labing* que, em alguns casos, pode superar o desempenho de

algumas versões do SVM e a *baseline* de humanos na classificação de sentimentos em dados com três ou quatro classes.

Goldberg e Zhou (GOLDBERG e ZHU, 2006) apresentaram um algoritmo semisupervisionado baseado em grafos a fim de inferir *ratings*, utilizando, em parte, dados não classificados, isto é, não rotulados. Para cada opinião não classificada *x,* esta foi conectada com outras *k* vizinhas previamente classificadas. Além disso, a opinião *x* também foi conectada com suas vizinhas *k'* não rotuladas. Esse grafo criado com tais relações foi utilizado como treinamento para algoritmos de aprendizado, onde a função *f(x)* foi utilizada para suavizar o grafo. Como experimento, eles usaram cinco algoritmos de aprendizado baseados em regressão e em *metric labeling*, demonstrando o benefício em utilizar opiniões não rotuladas no problema de inferência de *rating*.

Analisando dados do *Twitter*, Pak e Paroubek (PAK e PAROUBEK, 2010) coletaram microtextos e os separaram em três classes: sentimento positivo, sentimento negativo e textos objetivos. Esses *tweets* foram selecionados a partir de *emoticons* que apresentassem uma relação com os sentimentos "felizes" ou "tristes". As TEC's utilizadas para o treinamento foram n-gramas e a frequência dos mesmos nos *tweets* selecionados. Entretanto, para o treinamento do classificador utilizado (Naive Bayes), eles utilizaram, além de n-gramas, marcadores POS. Como resultado final, eles demonstram que o melhor resultado foi utilizando bigramas, com acurácia chegando a 85%.

Qu *et al.* (QU; IFRIM; WEIKUM, 2010) introduziram um novo tipo de *bag-of-opinions*. Seja uma opinião composta de várias frases, cada frase foi assinalada com um *score* e o *rating* foi inferido agregando os resultados dos *scores*. Para determinar o *score*, um método de regressão foi utilizado, no qual o modelo foi inferido baseando-se nos valores de todas as frases por meio de um modelo de n-gramas proposto. Este modelo avalia o *score* de cada unigrama e, por fim, gera um *score* final para uma frase, no qual um unigrama é o foco da frase (raiz), seguido de n-gramas modificadores e negadores. Os autores mostraram que esta técnica supera todos os trabalhos anteriores em uma margem significante.

Long *et al.* (LONG; ZHANG; ZHUT, 2010) propuseram uma nova pesquisa em seleção de opiniões a fim de estimar os *ratings* para serviços em sites utilizando a distância de informação das opiniões por meio da complexidade Kolmogorov. O modelo Kolmogorov associa um valor numérico a cada *string* binária e induz um conceito de similaridade entre tais *strings*. Neste trabalho, a inferência do *rating* foi feita em relação a um atributo do serviço.

Isto é, seja um item A com vários atributos $a_1, a_2, \ldots, a_n$. Para inferir o *rating* para A, os autores utilizaram uma combinação dos valores inferidos para cada atributo a por meio de classificadores de redes Bayesianas. Este método produziu bons resultados para o problema de análise de sentimentos multiclasse usando qualquer tipo de opiniões, sejam elas compreensíveis (quando estão relacionadas especificamente sobre os atributos de uma entidade) ou não (quando algum atributo não possui uma opinião) em relação aos atributos utilizados: preço, serviço, quartos e limpeza. Quando o resultado foi estimado para opiniões compreensíveis, a acurácia, entretanto, não chega a 60%.

Albornoz *et al.* (DE ALBORNOZ *et al.*, 2011) analisaram o impacto de diferentes características de um produto e o *rating* final. O objetivo é inferir o *rating* com base no *rating* que cada atributo que um produto recebeu em uma determinada avaliação. Para isso, os autores criaram um vetor com a *intensidade dos atributos*, que foi baseado na polaridade e na força da opinião expressada e em outras opiniões associadas a ela, utilizado para o treinamento dos algoritmos de aprendizado. Em relação aos resultados, o algoritmo Logistic (disponível na ferramenta Weka) apresentou o melhor resultado, atingindo 46,9% de acurácia em relação à 5 classes.

Embora também tenham como principal objetivo a classificação multiclasse, Paltoglou e Thelwall (PALTOGLOU e THELWALL, 2013) exploraram outros dois tipos de dimensão afetiva para classificar as opiniões: a valência e a excitação. Eles construíram os vetores de características por meio de *tokens* extraídos considerando as duas dimensões afetivas citadas e utilizam um modelo de regressão e uma variação do algoritmo SVM (OVA) para classificar uma opinião em uma escala de sentimento (escala 1-5).

Na Tabela 6, os trabalhos estão organizados destacando o domínio, as TEC’s e os algoritmos empregados e a acurácia final. Como citado por Pang e Lee (PANG e LEE, 2008), o problema de multiclasse pode ser resolvido por meio da regressão, já que os *rating* são ordinais. Isso pode justificar a escolha da grande maioria dos autores pela utilização do SVM e outros modelos de regressão.

3.4 APLICAÇÕES DA ANÁLISE DE SENTIMENTOS

Além das pesquisas voltadas para a comparação entre técnicas de aprendizado e seleção de características, pode-se encontrar trabalhos que, a partir do uso das mesmas,

apresentam também uma aplicação final. Nos exemplos abaixo, destaque para trabalhos voltados para as áreas de educação e serviços.

Tabela 6. Resumo dos principais trabalhos em análise de sentimento multiclasse

Autores	Domínio	Técnicas de Extração de Características	Algoritmos	Acurácia (%)
Pang e Lee 2005	Filmes	Frequência de um termo	SVM One-vs-all, Regression and Metric label	59,4 (SVM + vetor de palavras + regressão)
Goldberg e Zhou 2006	Filmes	Modelo semi-supervisionado baseado em grafos	Regressão (SVM), Metric Labeling e PSP	59,2 (regressão+PSP ou regressão)
Pak e Paroubek	*Tweets*	Frequência, n-gramas e POS	NB	60-80 (NB + bigramas)
Qu *et al.*2010	Produtos da Amazon	Bag of opinions	Regressão	-*mostra apenas o erro quadrático médio
Long *et al.* 2010	Hotéis	Complexidade Kolmogorov + rede bayesiana	SVM	73,1 – 57,3 (Kolmogorov+SVM baseado em atributos)
Albornoz *et al.*, 2011	Hotéis	Vetor de intensidade das características	Regressão logística, SVM e Árvore funcional	46,9 (vetor+regressão)
Paltoglou e Thelwall 2013	Notícias	Palavras que expressem valência e excitação	Regressão de Vetor de Suporte, SVM (OvA)	51,8 (Excitação + SVM (OvA))

Chen *et al.* (CHEN *et al.*, 2006) criam uma análise visual de opiniões positivas e negativas do livro “The Da Vinci Code”. Eles utilizam uma ferramenta visual, o TermWatch, para construir uma rede multicamada de termos baseada em associações sintáticas, semânticas e estatísticas. A fim de avaliar os termos que foram selecionados anteriormente, eles utilizam um modelo preditivo baseado no SVM. Como característica para o treinamento, um conjunto de opiniões positivas e negativas é utilizado. Neste caso, uma opinião é decomposta em três componentes que refletem a presença de termos positivos, negativos e comuns em ambas as categorias.

Mak *et al.* (MAK; KOPRINSKA; POON, 2003) criaram um sistema de recomendação web utilizando categorização textual de sinopses de filmes armazenadas no IMDB, selecionados do EachMovie database. Primeiramente, eles adaptaram as opiniões a fim de serem utilizadas nos algoritmos, representando-as em vetores, retirando palavras que não possuem informação útil, utilizando valores para cada palavra restante e ranqueando as características do corpus resultante através de três TECs: IG, DF e Informação Mútua. Com essa primeira etapa finalizada, eles utilizaram três algoritmos para construir um classificador para um usuário do sistema: kNN, Decisions Trees e o Naive Bayes. O desempenho final dos

algoritmos foi em torno de 60 a 65%, com as árvores de decisão apresentando o melhor resultado, entretanto, a diferença entre os três é pouco significativa.

Em (GO; BHAYANI; HUANG, 2009), Go *et al.* utilizaram *emoticons* a fim de treinarem opiniões retiradas do *Twitter*, utilizando algoritmos de aprendizado. Além dos *emoticons*, palavras-chave presentes no site Twittratr[19] que tenham sentimento positivo ou negativo foram utilizadas no treinamento como unigramas e bigramas. Após testes, eles atingiram cerca de 83% de acurácia com o algoritmo Naive Bayes configurado tanto com unigramas como bigramas. Por fim, os autores também disponibilizaram um site, o *sentiment140*[20], onde é possível saber o sentimento sobre algo em relação aos *tweets* existentes. O site cria uma lista de *tweets* positivos, negativos e neutros, além de gráficos que mostram qual sentimento é predominante.

Na área de educação, Ortigosa *et al.* (ORTIGOSA; MARTÍN; CARRO, 2014) construíram um modelo para avaliar postagens no *Facebook*™[21] e, a partir da detecção do sentimento habitual do usuário, verificar mudanças emocionais. A aplicação é chamada de SentBuk. Essa informação foi utilizada em sistemas e-learning a fim de recomendar atividades mais adequadas em relação ao humor do estudante em determinado período. Eles construíram um classificador léxico e, quando um grande número de *posts* foi classificado, eles usaram essas mensagens como entrada de treinamento para o algoritmo de aprendizado de máquina. Para realizar os testes eles utilizaram os algoritmos J48, Naive-Bayes e SVM (radial e sigmoide), onde o melhor resultado foi utilizando o algoritmo SVM (sigmoide) com 83% de acurácia.

[19] http://twitrratr.com/
[20] http://www.sentiment140.com/
[21] http://facebook.com

CAPÍTULO 4 – AVALIAÇÃO DOS ALGORITMOS NATIVOS

Segundo (LIU, 2012), a análise de sentimentos é sensível em relação ao domínio utilizado no treinamento, no qual um conjunto de técnicas de extração bem como um classificador treinado para um determinado conjunto de dados não possui bom desempenho para outro domínio. Isso porque palavras ou expressões utilizadas para expressar uma opinião podem ser diferentes para cada domínio e, portanto, cada etapa de extração de características e algoritmo de aprendizado apresenta resultados dependentes do domínio.

Dessa forma, o objetivo desse Capítulo é descrever os passos da análise explorativa realizada a fim de avaliar as técnicas de extração de características e os algoritmos de aprendizado descritos no Capítulo 2, utilizando uma base de dados com opiniões sobre hotéis. As melhores técnicas de extração de características e algoritmos serão utilizados na avaliação do modelo de divisões binárias proposto para o mesmo domínio de hotéis, modelo que está descrito no Capítulo 6.

O domínio de hotéis está descrito no Capítulo 6 e apenas a base de dados é apresentada na Seção 4.1, juntamente com as ferramentas utilizadas para realizar os testes. A Seção 4.2 descreve os passos realizados na etapa de pré-processamento, como a retirada de caracteres especiais e *stopwords.* Além disso, descreve a criação do bag-of-words utilizando unigramas e bigramas. Na Seção 4.3, o processo de seleção de características e vetorização das opiniões é exibido. A Seção 4.4 contém os resultados da avaliação dos algoritmos testados com os arquivos criados por meio da etapa de extração de características. Por fim, a Seção 4.5 apresenta uma discussão dos resultados em relação a outros trabalhos existentes, comparando, quando possível, os resultados obtidos nesse estudo com outras pesquisas existentes.

4.1 CONFIGURAÇÕES DOS TESTES E A BASE DE DADOS UTILIZADA

Nessa pesquisa, as linguagens Python e Java foram utilizadas a fim de analisar o sentimento geral das opiniões da base de dados. Para o tratamento textual, seleção de características e configuração dos arquivos para a análise dos algoritmos, o pacote Anaconda[22] foi utilizado. A ferramenta Weka[23] foi utilizada a fim de avaliar o desempenho dos algoritmos de aprendizado descritos no Capítulo 2. Para os testes, uma máquina Samsung

[22] https://www.continuum.io/downloads
[23] http:// www.cs.waikato.ac.nz/ml/weka/

com processador Intel Core i5 2.6 GHz, com 4MB de memória e sistema operacional Windows 10 foi utilizado.

Para realizar a análise de sentimentos, um conjunto de dados conhecido foi selecionado a fim de analisar diversas opiniões com escalas variando de 1 a 5. Dessa forma, parte da base de dados referente às opiniões do *TripAdvisor*™ foi utilizado, onde o mesmo contém avaliações em inglês. Essa base de dados foi utilizada na pesquisa de Wang *et al.* (WANG; LU; ZHAI, 2010) a fim de analisar opiniões expressas sobre uma entidade em relação aos aspectos da mesma, descobrindo a latência da opinião de cada indivíduo bem como a importância de cada aspecto na formação do *rating* final.

Como a base de dados possui um grande número de opiniões divididas por hotéis, 7500 foram selecionadas aleatoriamente, respeitando a divisão de 1500 opiniões para cada classe da escala de Likert (1-5). Cada opinião foi tratada a fim de separar apenas dois campos principais:

- Opinião;
- *Rating*.

Outros dados como título, autor e data da criação foram desconsiderados já que o intuito está na análise das opiniões, embora outros trabalhos, como (MUKHERJEE; BASU; JOSHI, 2014), analisem a preferência do autor. Além disso, esta análise considera apenas ao sentimento final em relação a um hotel, diferentemente de (LONG; ZHANG; ZHUT, 2010) que analisam quatro características (preço, quarto, serviços e limpeza) e calculam o *rating* com base na média desses atributos.

4.2 PRÉ-PROCESSAMENTO TEXTUAL E *BAG-OF-WORDS*

Com a seleção das 7500 opiniões da base de dados inicial, o próximo passo foi o tratamento textual retirando caracteres especiais (ver Capítulo 2 com exemplos de caracteres especiais) e *tags* em HTML. Após essa normalização textual, as opiniões foram lidas e cada palavra foi salva como um unigrama ou bigrama a fim de contabilizar o número de ocorrências de uma dada palavra em uma classe. Além disso, palavras consideradas de com pouco ou nenhum sentimento, como *a*, *an*, *the*, *on*, *in*, *at*…entre outras também foram excluídas. Uma lista com as *stopwords* está presente no APÊNDICE A.

Outro passo importante foi o tratamento de opiniões com palavras que expressam negação. Desta forma, frases como "*even bed sheets were not clean*" tem seu sentimento

invertido pelo token “not”, como feito em (DAS e CHEN, 2001). A fim de tratar esse problema, palavras que tem os modificadores *no*, *not* ou *nothing* foram transformadas em um único unigrama. Como exemplo, *not clean* é representado pelo token *not_clean* que, analisando gramaticalmente a expressão, pode ser sinônimo a *dirty*.

Embora as etapas de tratamento e normalização textual sejam de grande importância, nem todas as etapas citadas no Capítulo 2 foram realizadas, como a retirada de palavras grafadas incorretamente e retirada de radicais, já que grande parte dos trabalhos não utilizam todos esses tratamentos textuais. Essa etapa deve ser melhor estudada, ficando como trabalho futuro a fim de comparar a importância do pré-processamento da linguagem natural. Isso se deve pelo fato de que, apenas com os tratamentos textuais realizados previamente, os resultados já foram bons se considerados outros trabalhos com análise de sentimento.

Um ponto que não foi considerado é a importância do autor das opiniões, colocando pesos em relação à confiança de dado usuário. Em grande parte das opiniões desta base de dados, o autor não pode ser identificado o que impossibilitou uma análise em termos da confiabilidade do mesmo. Nesses casos, a identificação era como “A TripAdvisor Member”.

Com esses dados tratados, cada palavra de uma opinião forma um unigrama e duas palavras um bigrama, assim como feito no trabalho de (PANG; LEE; VAITHYANATHAN, 2002) formando o chamado *bag-of-words*. Unigramas e bigramas são as principais formas de representação de características e possuem bons resultados na análise de sentimento (LIU, 2012), tanto na classificação binária (PANG; LEE; VAITHYANATHAN, 2002) como multiclasse (PANG e LEE, 2005). Esses n-gramas foram retirados e armazenados, bem como o número de ocorrência de cada n-grama em uma determinada classe. N-gramas que apareciam com pouca frequência foram desconsiderados já que estes podem diminuir a precisão dos algoritmos, servindo como ruído para a classificação. Um limite de 16 repetições foi colocado.

4.3 TÉCNICAS DE SELEÇÃO DE CARACTERÍSTICAS

Essa etapa de seleção de características é fundamental para a fase de extração de características (LIU, 2012) e, como demonstrado por (PRUSA; KHOSHGOFTAAR; DITTMAN, 2015) em dados recolhidos do *Twitter*™, pode melhorar significativamente a performance da classificação.

Com os n-gramas selecionados e armazenados a partir da base de dados do *TripAdvisor*™, como dito anteriormente na Seção 4.2, o próximo passo consiste na seleção dos principais n-gramas, isto é, selecionar as palavras ou *tokens* que possuem relação com o sentimento expresso em uma opinião utilizando algoritmos de extração que possuem bons resultados comprovados na análise binária. Para efeito organizacional, dois vetores foram criados: uma com unigramas e outra com n-gramas (compostos por unigramas+bigramas) juntamente com o número de ocorrências de dado n-grama em uma determinada classe.

As três fórmulas de seleção de características descritas no Capítulo 2 foram utilizadas: IG, GR e Chi. Os n-gramas selecionadas são palavras em inglês que tipicamente aparecem em opiniões que avaliam hotéis presentes em sites como *TripAdvisor*™, *Booking.com*™[24], entre outros, onde cada n-grama pode estar presente em qualquer classe. Uma lista com alguns n-gramas selecionados por cada fórmula encontra-se no APÊNDICE B.

O grande ganho da seleção de palavras associadas com um sentimento está em relação à probabilidade de uma palavra negativa aparecer em uma frase positiva, por exemplo. Desta forma, avaliando essa pequena probabilidade, palavras que aparecem mais vezes em uma determinada classe tem mais peso na definição de uma classe final no desafio da inferência *ratings*.

Para cada TEC, uma lista de palavras que possuem relação com o sentimento dado a uma opinião foi criada, nesse caso relacionado com o *rating* referenciado. Tendo essa lista de n-gramas, o processo de vetorização foi realizado seguindo o modelo disponível pela biblioteca *SciKit Learn* em Python e descrito na Seção 2.2.4, transformando todas as opiniões em um grande arquivo onde cada coluna é representada por um n-grama que foi selecionado por um dos métodos de extração e cada linha é representada por uma opinião que foi vetorizada. Cada célula M_{ij} representa o número de ocorrências de um n-grama em *j* em uma opinião *i.* Além da frequência, onde um número inteiro é utilizado, a fórmula *tfidf*, disponível no pacote *SciKit*, também foi utilizada para a criação dos arquivos de treinamento. Além disso, arquivos que utilizam apenas unigramas e arquivos que utilizam n-gramas (unigramas incluídos) também foram criados.

Para cada TEC foram testadas 250, 500, 1000, 1500, 2000 e 2500 n-gramas a fim de comparar o desempenho de cada algoritmo de aprendizado na classificação multiclasse bem como o número de características utilizadas no treinamento. Com isso, a queda ou o

[24] http://www.booking.com

crescimento dos resultados pode ser avaliado, como será exibido por meio de gráficos e tabelas na Seção 4.4. Esse número de características pode ser observado no exemplo da Seção 2.2.4, onde o número entre parênteses (15) é o número de n-gramas.

Esses modelos serão compreendidos pelos algoritmos de classificação como o SVM e o Naive Bayes, como visto na próxima Seção utilizando uma representação de características baseada na frequência ou no peso de cada n-grama.

4.4 AVALIAÇÃO DE ALGORITMOS DE CLASSIFICAÇÃO

Com os arquivos para treinamento criados conforme descrito na Seção anterior, a ferramenta Weka foi utilizada a fim de testar o desempenho dos algoritmos de aprendizado supervisionado em relação às opiniões sobre hotéis. Com base nos modelos descritos na Seção 2.3, os seguintes algoritmos multiclasse nativos foram testados:

- Naive Bayes Multinomial;
- Naive Bayes;
- Árvore de decisão - J48;
- kNN - IBk.

Além desses algoritmos nativos, os modelos multiclasse adaptados descritos na Seção 2.3.5 também foram avaliados, utilizando alguns dos modelos descritos na Seção 2.3. Esses algoritmos foram selecionados com base nos resultados obtidos nos trabalhos relacionados e nos testes realizados com os algoritmos nativos nessa dissertação, que estão descritos Capítulo 3. Desta forma, os seguintes modelos adaptados foram avaliados:

- OvO - SMO;
- OvO - LibSM;
- OvA – Naive Bayes Multinomial.

O treinamento e o desempenho de cada algoritmo nativos e técnica de multiclasse adaptada pode ser conferido abaixo, com a precisão, a acurácia e o recall baseados na matriz de confusão disponível no final do treinamento.

Na Tabela 7 pode-se notar o resultado da execução dos algoritmos com o tempo contabilizado. O algoritmo Naive Bayes Multinomial precisou de menos de dois décimos de segundo em todos os arquivos de teste para criar um modelo enquanto o algoritmo SMO gastou entre 10,72 (250) e 414,43 (2500) segundos para criar um modelo em uma das

execuções. Nesse caso, o SMO está configurado com um modelo de divisão *one-vs-one* tendo em vista que o algoritmo SVM é utilizado principalmente para problemas com duas classes.

Tabela 7. Tabela com o tempo em segundos do NBM e do SMO utilizando o Chi como método de extração de características

Naive Bayes Multinomial						
Bag-of-words	**Número de unigramas ou bigramas**					
	250	500	1000	1500	2000	2500
Unigrama	**0,02**	0,03	0,05	0,08	0,1	0,14
N-grama	**0,02**	0,04	0,07	0,09	0,12	0,16
SMO						
Unigrama	**13,8**	38,81	112,72	192,09	257,07	414,43
N-grama	**10,72**	32,33	97,51	136,64	256,36	328,74

Na Tabela 8, os desempenhos dos dois algoritmos são exibidos evidenciando a utilização de unigramas ou n-gramas (onde, nesse caso, n-gramas são considerados formados por unigramas e bigramas). Para o algoritmo Naive Bayes, os melhores resultados foram obtidos com unigramas em 4 dos 6 arquivos utilizados nos testes, entretanto a maior acurácia foi com n-gramas, onde 2500 tokens foram utilizados, atingindo 60,45% de acurácia, embora esse resultado seja muito semelhante à acurácia com unigramas (60,3%).

Tabela 8. Tabela com a acurácia dos algoritmos NBM e SVM

Naive Bayes Multinomial						
Bag-of-words	**Número de unigramas ou bigramas**					
	250	500	1000	1500	2000	2500
Unigrama	**56,2**	**57,8**	**59,02**	59,36	**60,1**	60,3
N-grama	55,21	57,74	58,81	**59,43**	59,89	**60,45**
SMO						
Unigrama	55,52	56,85	56,24	55,12	54,88	55,74
N-grama	**55,78**	**57,07**	**56,68**	**56,06**	**55,3**	**56,05**

O algoritmo SMO, uma variante do SVM, apresentou os melhores resultados quando testado com n-gramas. Em todos os seis arquivos de teste o desempenho dos n-gramas superou o resultado dos unigramas, mesmo que a diferença de acurácia tenha sido bem pequena.

Na Tabela 9 são apresentados os testes que exibem o desempenho das técnicas de seleção de características utilizadas nesse trabalho com arquivos de 1000 n-gramas para o treinamento. Embora apenas essa tabela seja exibida, testes com 250, 500, 1500, 2000 e 2500 também foram realizados.

Tabela 9. Tabela com a acurácia dos métodos de seleção utilizados nesta pesquisa

Técnica de seleção: Chi-quadrado				
Bag-of-words	**Algoritmo – Número de unigramas ou bigramas**			
	NBM – 1000	LibSVM - 1000	J48 – 1000	kNN – 1000
Unigrama	**59,02**	53,17	41,32	34,37
N-grama	58,81	**55,17**	41,09	**36,92**
Técnica de seleção: Ganho de Informação				
	NBM – 1000	LibSVM - 1000	J48 – 1000	kNN – 1000
Unigrama	49,44	49,21	42,36	36,18
N-grama	43	46,45	**42,64**	35,2
Técnica de seleção: Ganho Médio				
	NBM – 1000	LibSVM - 1000	J48 – 1000	kNN – 1000
Unigrama	32,25	35,84	27,78	30,98
N-grama	26,76	29,94	-	28,64

O método Chi-quadrado teve melhor desempenho em quatro dos algoritmos testados, tanto com unigrama como com unigramas+bigramas. Ele só foi superado pelo IG quando o algoritmo de Árvores de Decisão J48 foi executado. Em relação ao maior desempenho, ele foi alcançado pelo método CHI com o Naive Bayes Multinomial, atingindo 59,02% de acurácia.

Na Tabela 10 são apresentados alguns dos resultados que demonstram que a utilização da frequência de um n-grama superou o modelo de pesos *tfidf*. Abaixo, dois algoritmos com os melhores resultados, bem como o melhor algoritmo de seleção de características CHI tem sua acurácia exibida. Esse resultado difere de trabalhos como (PALTOGLOU e THELWALL, 2010) que citam a melhoria do desempenho quando a fórmula *tfidf* ou variantes desta fórmula são utilizadas. Entretanto, nesses trabalhos, variantes do modelo *tfidf* também foram testadas.

A Tabela 11 apresenta a relação técnica de extração/classificador que apresentou a melhor acurácia exata: o Naive Bayes Multinomial com arquivos de entrada configurados com 2500 n-gramas e características selecionadas por meio do Chi-quadrado. Além disso, a acurácia aproximada é exibida evidenciando que a classificação, embora não tenha uma acurácia exata tão alta, ela é aceitável considerando que os altos valores indicados pela acurácia próxima atingindo 93,7%.

Tabela 10. Tabela com a acurácia da frequência versus modelo *tfidf*

Naive Bayes Multinomial – CHI						
Vetorização	**Número de unigramas + bigramas**					
	250	500	1000	1500	2000	2500
Frequência	**55,21**	**57,74**	**58,81**	**59,43**	**59,89**	**60,45**
TFIDF	48,79	50,61	51,8	52,04	51,92	52,84
SMO – CHI						
Frequência	**55,78**	**57,07**	**56,68**	**56,06**	**55,3**	**56,05**
TFIDF	48,3	48,69	48,28	47,29	47,63	46,83

Tabela 11. Matriz de confusão para o algoritmo NBM: melhor configuração encontrada

a	b	c	d	e	Acurácia		← classificado como
					Exata	Próxima	
1108	297	69	17	9	0,739	0,937	a=1
382	641	383	85	9	0,427	0,937	b=2
75	284	845	270	26	0,563	0,933	c=3
32	77	193	858	340	0,572	0,927	d=4
21	22	33	342	1082	0,721	0,949	e=5
1618	1321	1523	1572	1466	**0,605**	**0,937**	-

A Tabela 12 apresenta o resultado dos algoritmos de aprendizado utilizando a melhor combinação de técnicas de seleção de características demonstradas nas tabelas anteriores em relação às três medidas descritas anteriormente no Capítulo 2: acurácia, precisão e recall. Como pode ser notado, o algoritmo Naive Bayes Multinomial apresenta acurácia, precisão e recall superiores a praticamente todos os modelos em relação à acurácia, com exceção do primeiro arquivo de testes com 250 características tem o SMO como melhor algoritmo.

4.5 ANÁLISE DOS RESULTADOS

Após finalizada a etapa de testes com algoritmos de classificação para as opiniões relacionadas a hotéis presentes no *TripAdvisor*, notou-se que o Naive Bayes Multinomial apresentou o melhor desempenho em termos de acurácia (A), precisão (P) e recall (R) superando os dois modelos do SVM testados: o SMO e o LibSVM. Muitos dos trabalhos

referenciados em análise de sentimentos multiclasse, inclusive, não apresentam nenhum teste com o Naive Bayes.

Tabela 12. Resultados utilizando Chi-quadrado, N-gramas e vetor de frequência – Melhor Resultado

Naive Bayes Multinomial														
Número de unigramas+bigramas														
250			500			1000			1500			2000		
A	*P*	*R*	*A*	*P*	*R*	*A*	*P*	*R*	*A*	*P*	*R*	*A*	*P*	*R*
0,552	0,546	0,552	**0,575**	0,571	0,575	**0,588**	0,585	0,588	**0,594**	0,592	0,595	**0,599**	0,591	0,599
Naive Bayes														
250			500			1000			1500			2000		
A	*P*	*R*	*A*	*P*	*R*	*A*	*P*	*R*	*A*	*P*	*R*	*A*	*P*	*R*
0,519	0,512	0,519	0,527	0,524	0,527	0,522	0,519	0,522	0,513	0,511	0,513	0,514	0,513	0,514
Árvores de Decisão – J48														
250			500			1000			1500			2000		
A	*P*	*R*	*A*	*P*	*R*	*A*	*P*	*R*	*A*	*P*	*R*	*A*	*P*	*R*
0,417	0,416	0,417	0,408	0,409	0,408	0,411	0,418	0,419	0,42	0,419	0,42	0,416	0,414	0,416
kNN – Ibk														
250			500			1000			1500			2000		
A	*P*	*R*	*A*	*P*	*R*	*A*	*P*	*R*	*A*	*P*	*R*	*A*	*P*	*R*
0,418	0,42	0,418	0,376	0,379	0,376	0,369	0,399	0,369	0,345	0,38	0,345	0,333	0,373	0,333
SVM – SMO														
250			500			1000			1500			2000		
A	*P*	*R*	*A*	*P*	*R*	*A*	*P*	*R*	*A*	*P*	*R*	*A*	*P*	*R*
0,557	0,558	0,553	0,571	0,568	0,571	0,567	0,565	0,567	0,561	0,561	0,561	0,553	0,554	0,553
SVM – SMVLib (linear)														
250			500			1000			1500			2000		
A	*P*	*R*	*A*	*P*	*R*	*A*	*P*	*R*	*A*	*P*	*R*	*A*	*P*	*R*
0,549	0,545	0,549	0,559	0,557	0,559	0,552	0,553	0,552	0,53	0,532	0,53	0,532	0,535	0,532
OvA – NBM														
250			500			1000			1500			2000		
A	*P*	*R*	*A*	*P*	*R*	*A*	*P*	*R*	*A*	*P*	*R*	*A*	*P*	*R*
0,554	0,554	0,543	0,575	0,565	0,577	0,59	0,578	0,59	0,597	0,587	0,597	0,6	0,59	0,6

Outro fato importante, conforme identificado na Tabela 9, é que o uso da técnica de seleção de características Chi-quadrado apresenta resultados superiores ao do Ganho de Informação. Nota-se que o Ganho de Informação é o mais utilizado e em trabalhos como (TAN e ZHANG, 2008), apresenta melhor resultado na classificação de um conjunto de opiniões em três tipos de domínios: educação, filmes e casa. Isso evidencia que é necessário selecionar técnicas adequadas para cada tipo de domínio, isto é, para cada domínio, um

conjunto de técnicas deve ser testado para que as melhores configurações sejam escolhidas. Além disso, nossa pesquisa exibe bons resultados em relação a outros estudos para problemas multiclasse, já que nenhum dos trabalhos citados utiliza alguma das fórmulas de extração de características utilizadas (Chi, IG e GR). Mesmo não analisando o mesmo domínio ou técnica, a acurácia dos modelos em alguns casos supera outros estudos que utilizam menos classes, como o trabalho inicial de Pang e Lee (PANG e LEE, 2005).

Analisando a matriz de confusão representada na Tabela 11, nota-se que, embora a acurácia exata seja de 60,5%, a acurácia próxima atinge 93,7%, o que equivale a resultados obtidos em bons trabalhos de análise binária de sentimento como (MATSUMOTO; TAKAMURA; OKUMURA, 2005) e (PALTOGLOU e THELWALL, 2010). Outro fato a ser analisado é a alta acurácia para os extremos (1 e 5 estrelas), ambas com mais de 70% de acurácia, fator que pode ser notado em (PALTOGLOU e THELWALL, 2013), no qual a acurácia destas duas classes supera em quase 10% a acurácia das outras, atingindo valores superiores a 52%. Quando analisada a acurácia utilizando os vizinhos mais próximos (±1) como corretos, essa disparidade de classificação diminui bruscamente, com todos os valores superando a marca de 90%.

Assim como em (PAK e PAROUBEK, 2010), a utilização de bigramas juntamente com unigramas teve melhor desempenho do que apenas analisando unigramas separadamente. Entretanto, esse resultado não condiz com o que foi encontrado em (PANG; LEE; VAITHYANATHAN, 2002), no qual o uso de unigramas superam bigramas+unigramas.

Embora o domínio de hotéis seja bem utilizado para o estudo da classificação multiclasse, na classificação binária ele é raramente utilizado. Para a análise de sentimentos binária, opiniões sobre filmes são mais utilizadas, assim como avaliações de produtos da *Amazon*. Analisando, então, os estudos que utilizam uma base sobre comentários sobre hotéis, como feito em (LONG; ZHANG; ZHUT, 2010) e (DE ALBORNOZ *et al.*, 2011), nota-se que os resultados obtidos nesse trabalho são satisfatórios, mesmo que a base de dados seja diferente. Essa comparação será feita posteriormente, incluindo os resultados do próximo Capítulo.

Considerando que muitos trabalhos têm como principal objetivo a análise binária e que entre os trabalhos relacionados citados não existe nenhum que utilize opiniões sobre hotéis, o próximo Capítulo apresenta uma variação da técnica de dicotomias aninhadas que utilize

algoritmos binários para o problema multiclasse, e que possa ser adaptado para qualquer domínio.

CAPÍTULO 5 – PROPOSTA DE UM MODELO DE DIVISÕES BINÁRIAS

De acordo com discussão dos trabalhos relacionados no Capítulo 3, os resultados obtidos pela classificação binária são naturalmente melhores do que qualquer tipo de classificação multiclasse na área de análise de sentimentos. Da mesma forma, os resultados obtidos no Capítulo anterior para uma base de dados multiclasse também apresentam baixa acurácia quando comparados à classificação binária.

Entretanto, a classificação multiclasse é de grande importância, principalmente devido a utilização de escalas como os *ratings* e as notas em sites *e-commerce* para avaliar algum item. Além disso, de acordo com uma pesquisa realizada pelo site *PracticalECommerce*, uma estrela a mais ou a menos pode ser essencial no momento de decidir sobre um compra.

Dessa forma, esse Capítulo propõe a utilização de uma técnica alternativa aos modelos de classificação multiclasse discutidos na Seção 2.3.5, que também permita a utilização divisões binárias para o problema de inferência de *rating*. Na Seção 5.1, um exemplo de divisões binárias com os modelos OvO e OvA são apresentados. O algoritmo de divisões binárias, o algoritmo Nested Dichotomies (ND), é apresentado na Seção 5.2. Na Seção 5.3, uma discussão em relação as divisões binárias propostas e a hierarquia das classes é feita. E na Seção 5.4, um modelo de análise de sentimentos multiclasse que utilize o algoritmo proposto NDiST é apresentado.

5.1 MODELOS MULTICLASSE ADAPTADO

Os métodos multiclasse adaptados baseiam-se na decomposição do problema multiclasse inicial para uma combinação de problemas binários. As duas principais técnicas de divisão binárias foram apresentadas na Seção 2.3.5:

i. *one-vs-one (OvO)*;
ii. *one-vs-all (OvA).*

Para um problema com 4 classes {1, 2, 3, 4}, o OvO cria 6 classificadores (1-2, 1-3, 1-4, 2-3, 2-4, 3-4). Para a criação de cada classificador binário, as instâncias de treinamento possuem o rótulo correspondente a cada classificador, isto é, para uma divisão binária 3-4, apenas exemplos de treinamento classificados como 3 ou 4 são utilizados.

Na fase de classificação, a classe escolhida é baseada em uma votação direta dada pelo maior valor de acordo com a Equação 19 da Seção 2.3.5, selecionando a classe com maior número de votos. Exemplificando, dado uma nova instância *a*, a Tabela 13 e a Tabela 14 mostram uma predição para esse novo dado.

Tabela 13. Votos de cada classificador do modelo OvO

Classificador	f(*a*)=
1-2	2
1-3	1
1-4	1
2-3	2
2-4	2
3-4	3

Tabela 14. Contagem dos votos para cada classe

	Votos para cada classe			
Classe	1	2	3	4
Número de votos	2	**3**	1	0

Nesse exemplo, a classe 2 é a escolhida como rótulo da nova instância *a*. Em caso de empata, a escolhe é feita aleatoriamente (PIMENTA, 2004).

Avaliando o modelo OvA para o mesmo número de classes, 4 classificadores são criados (1 *vs* {234}, 2 *vs* {134}, 3 *vs* {124}, 4 *vs* {123}). Nesse caso, na fase de treinamento, todas as classes são utilizadas em cada divisão. No processo de classificação, dada uma nova instância *a*, a predição é dada por meio de Equação 18 da Seção 2.3.5, utilizando uma votação direta distribuída, conforme exibido na Tabela 15.

Tabela 15. Votos de cada classificador do modelo OvA

f(*a*)		Votos			
Classificador		1	2	3	4
1 *vs* {234}	1	0	0,333	0,333	0,333
2 vs {134}	Outra	0	1	0	0
3 vs {124}	Outra	0,333	0,333	0	0,333
4 vs {123}	Outra	0,333	0,333	0,333	0
Total		0,666	**1,999**	0,666	0,666

Nesse exemplo, a classe 2 recebe o maior valor (1,999). Dessa forma, a instância *a* é classificada como 2.

Estas abordagens são comumente utilizadas quando algoritmos SVM são indicados para o problema, por exemplo. Para a avaliação multiclasse realizada no Capítulo 4, os algoritmos SVM citados acima (SMO e LibSVM) utilizaram o modelo de divisões OvO. Já o modelo OvA foi utilizado tendo o Naive Bayes Multinomial como classificador.

5.2 NESTED DICHOTOMIES

Uma alternativa para os dois modelos discutidos na Seção anterior, o algoritmo Nested Dichotomies (ND) ou Dicotomias Aninhadas é representado como uma árvore de divisões binárias, na qual uma classe A é associada com dois nós B e C mutuamente exclusivos. Dessa forma, a raiz contém todas as classes para um problema qualquer multiclasse e as folhas correspondem ao número de classes. Dessa forma, *n-1* divisões são necessárias para criar uma árvore binária.

Para construir um classificador baseado na estrutura de uma árvore proposta, os seguintes processos devem ser feitos: na etapa de treinamento, para cada nó interno, somente as instâncias pertencentes às classes associadas com os nós são armazenadas; após isso, as classes pertencentes ao nó são agrupadas em dois subconjuntos, de modo que cada subconjunto contenha as classes associadas com exatamente um dos nós do nó inicial. Finalmente, classificadores binários para cada nó existente são criados.

A probabilidade de uma classe para o classificador ND é dada pelo produtório (RODRÍGUEZ; GARCÍA-OSORIO; MAUDES, 2010) na Equação 20:

$$p(c = C|x)\ \prod_{i=1}^{n-1} (I(c\ \in\ C_{i1})p(c\ \in\ C_{i1}|x, c\ \in\ C_i) +\ I(c\ \in\ C_{i2})p(c\ \in\ C_{i2}|x, c\ \in\ C_i)), \qquad (20)$$

onde C_{i1} e C_{i2} são dois subconjuntos de uma divisão de um conjunto de classes C_i em um nó interno i da árvore, e seja p(c $\in$ C_{i1} | x,c $\in$ C_i) e p(c $\in$ C_{i2} | x,c $\in$ C_i) a distribuição da probabilidade condicional estimada pelo modelo de duas classes do nó i em para uma instância x.

Para um conjunto de classes *C* e seja cada classe existente *c*, um custo unitário e_c para classificar uma classe *c* é definido de acordo com a Equação 21:

$$E_C = \sum_{c' \in C} P(c')e_c(c') \qquad (21).$$

Dessa forma, um custo E_C é calculado para cada classe e a classe *c* de maior custo é inferida para uma nova instância. O custo e_c de cada classe pode ser calculado, por exemplo, por meio da similaridade estatística entre as classes (YANG et al., 2014).

Analisando a Figura 4, o nó inicial (raiz) possui todas as classes para um problema com 4 classes. A partir disso, uma primeira divisão mutuamente exclusiva é realizada, separando as classes {1,2} em um nó e as classes {3, 4} em outro nó. A partir disso, cada nó filho também é dividido, chegando, nesse caso, nas folhas. Para cada divisão, um classificador binário é construído.

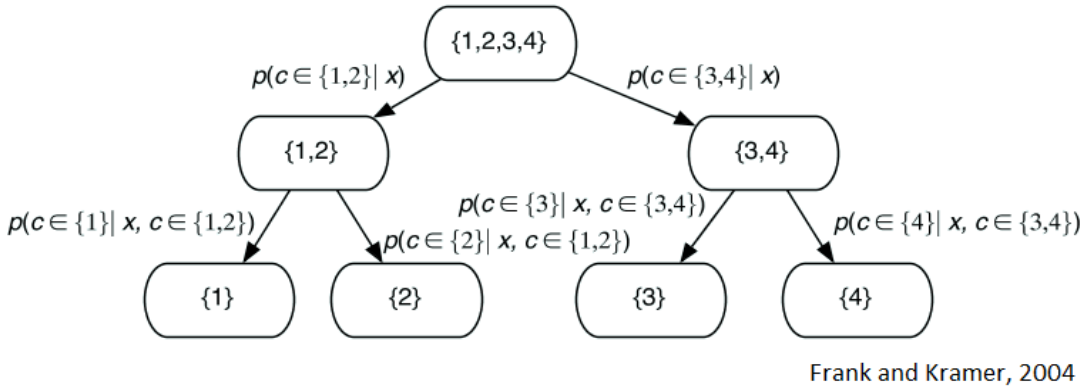


Figura 4. Um exemplo de árvore com divisões binárias para o problema de 4 classes

Para realizar uma inferência de uma classe, a função representada pela Equação 19 é utilizada. Dada uma instância não classificada *n* que esteja previamente rotulada como {1,2}, na qual a probabilidade seja igual para cada classe (0,25) e o custo para a classe {1} seja 2 e o custo para a classe {2} seja 3. Dessa forma, o custo $e_1 = 0{,}25 \,.\, 2 = 0{,}5$ e o custo $e_2 = 0{,}25 \,.\, 3 = 0{,}75$. Nesse caso, a classe inferida será a classe 2.

O algoritmo ND, com base nos trabalhos pesquisados, não apresenta nenhuma pesquisa na área de análise de sentimentos, seja binária ou multiclasse. Isso permite explorar este algoritmo que, como demonstrado por (FRANK e KRAMER, 2004), supera outros métodos multiclasse, como o *one-vs-one*, para os domínios testados.

5.3 DISCUSSÃO DOS MODELOS ADAPTADOS

Para um problema que possui uma escala de ordem entre as classes, os modelos OvA e OvO apresentam divisões que não respeitam tal hierarquia. Por exemplo, analisando o modelo OvA para 4 classes descrito na Seção 5.1, uma possível divisão poderia construir um classificador que compara a classe 2 com as demais (2 *vs* {1,3,4}). Considerando que existe

uma hierarquia entre essas classes, essa divisão não respeitaria tal hierarquia. Da mesma forma, um classificador OvO pode construir um classificador para as classes 1 e 4, também desrespeitando a hierarquia existente.

Analisando o algoritmo ND, algumas árvores de divisão podem criar classificadores respeitando a hierarquia de divisões, como por exemplo a árvore exibida na Figura 4. Um exemplo de árvore com divisões binárias para o problema de 4 classes da Seção 5.2. Analisando o problema de classificação multiclasse de *ratings* e, considerando que existe uma hierarquia entre as 5 classes do problema, o algoritmo ND é uma alternativa para o problema de inferência de *ratings*. Além disso, a primeira divisão da árvore poderia representar quais estrelas são mais importantes para um consumidor, conforme a divisão apresentada na pesquisa do site *Practical E-commerce*, na qual itens avaliados com 4 ou 5 estrelas são mais procurados.

5.4 O MODELO PROPOSTO

Considerando os problemas discutidos na Seção 5.3, o modelo proposto nesta Seção tem como principal objetivo se adaptar ao domínio, isto é, após uma análise de um conjunto de dados e com a seleção das melhores técnicas de extração de características, um modelo baseado em uma árvore com dicotomias aninhadas é criado. Esse classificador foi chamado de Nested Dicotomies Single Tree (NDiST).

Após escolhido um domínio para ser analisado, sendo este um conjunto de opiniões sobre filmes, hotéis, produtos eletrônicos, entre outros vários que podem ser observados no Capítulo 3, uma análise sobre a relevância dos *ratings* na compra de um produto ou serviço é feita. A partir dessa análise, uma árvore é proposta com base em uma primeira divisão ótima: qual a faixa de *ratings* é prioritária na escolha de um item.

Paralelamente, um processo de extração de características, descrito na Seção 2.2, deve ser criado a fim de escolher tanto a combinação de técnicas de extração bem como o algoritmo de aprendizado utilizado nas divisões binárias, como mostra o esquema da Figura 5.

A primeira etapa consiste na retirada de caracteres especiais, *stopwords* e o tratamento da negação de cada opinião presente na base de dados, de acordo com a Seção 2.2.1. Após isso, um *bag-of-words* com unigramas e bigramas de cada opinião é criado, conforme descrito na Seção 2.2.2. Desse *bag-of-words*, cada método de seleção de características é utilizado a fim de retirar quais n-gramas estão mais relacionados com determinada classe, isto é, quais

características são mais influentes em uma determinada classe. Essas listas podem ser criadas utilizando um número *x* de n-gramas. Por exemplo, nos testes do Capítulo 4, foram criadas listas somente de unigramas e listas de unigramas e bigramas. Para cada uma dessas listas, foram utilizados 250, 500, 1000, 1500, 200 e 2500 n-gramas para o treinamento.

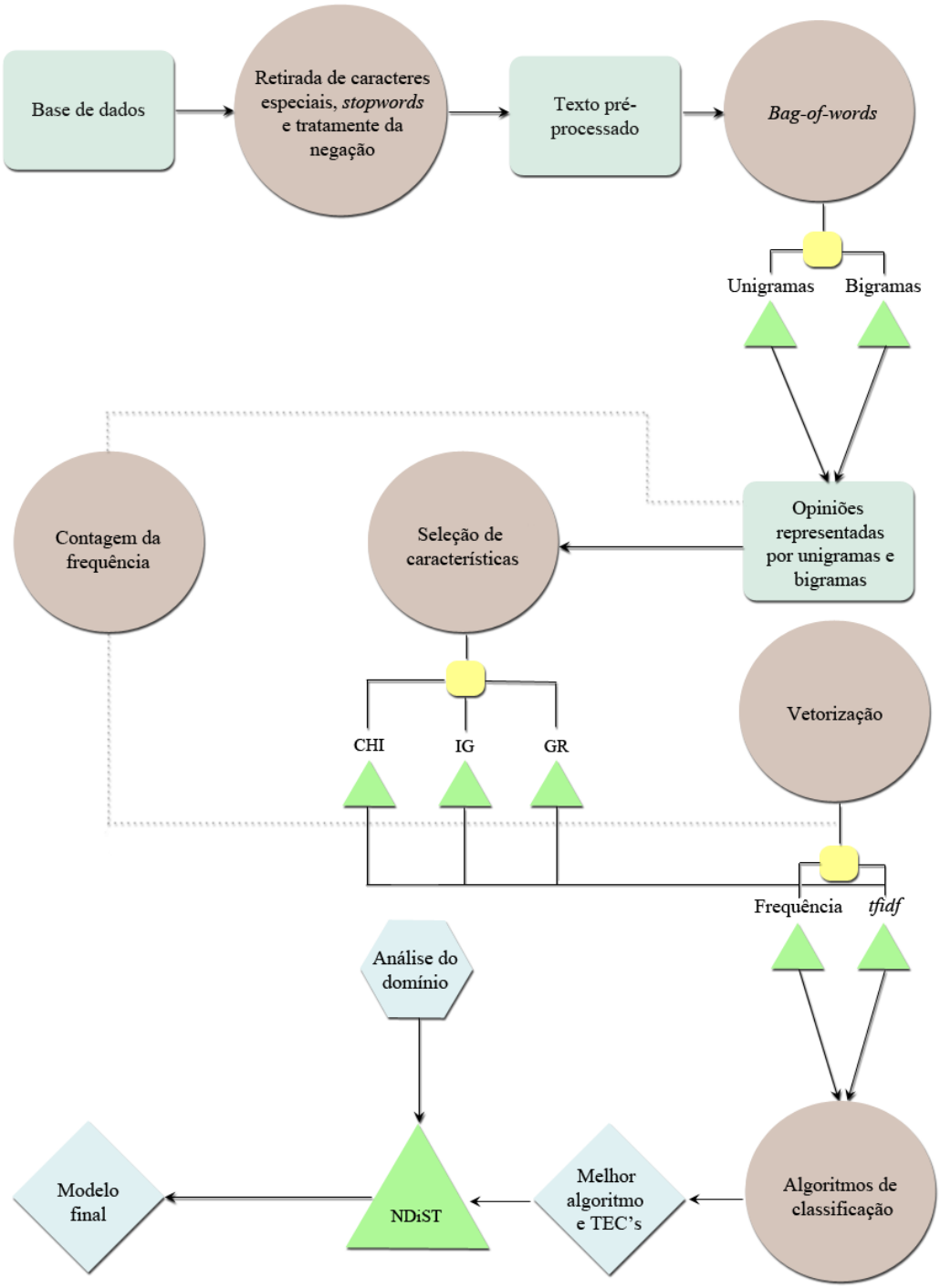


Figura 5. Modelo de análise de sentimentos proposto

Com os n-gramas selecionados e com várias listas de n-gramas criadas, as opiniões da base de dados, representadas por unigramas e bigramas, passam por um processo de vetorização, tanto utilizando a frequência de um n-grama como utilizando um peso dado pelo inverso da frequência (*tfidf*) calculado pela Equação 6, ambos descritos na Seção 2.2.4. A última etapa consiste em avaliar os algoritmos de aprendizado de máquina, selecionando também quais técnicas de extração apresentaram os melhores desempenhos.

Com as melhores técnicas escolhidas para uma base de dados e com a análise do domínio, uma árvore com dicotomias pode ser criada, sendo, nesse caso, a única árvore do algoritmo. Como mostrado no Algoritmo 1 – NDiST: Nested Dichotomies sem escolha aleatória, o algoritmo NDiST proposto, ao invés de selecionar uma árvore e a combinação dos nós aleatoriamente, constrói uma única árvore na qual a primeira divisão é baseada no estudo de um domínio, isto é, a árvore é adaptada ao domínio. Uma lista com as classes é criada a partir dos exemplos de treinamento por meio do método *criaListaClasse()*.

Algoritmo 1 – NDiST: Nested Dichotomies sem escolha aleatória

```
1: importaDados();
2: deletaDadosNãoRotulados ();
3: criaListaClasses();
4: insereClasseNó ();
5: para i=1até i<numNós
6: {
7:         constroiClassificadorNó (classificador);
8: }
9: calculaValidacaoCruzada (10);
```

As classes são inseridas em cada nó de acordo com a configuração de divisão proposta através de *insereClasseNó()*. Para cada nó da árvore, um classificador binário é configurado com algum algoritmo de aprendizado de máquina através do método *constroiClassificadorNó(classificador)*, no qual esse algoritmo foi escolhido com base na avaliação inicial dos modelos que foi descrita anteriormente.

Definindo estas etapas, o modelo passa por uma validação cruzada 10-fold, disponível na ferramenta Weka. Esse modelo divide randomicamente o conjunto de testes em 10 partes (KOHAVI, 1995) e cada parte é treinada e testada 10 vezes. Por fim, a acurácia, o recall e a precisão são calculadas de acordo com o desempenho de cada conjunto 10-fold.

CAPÍTULO 6 - ESTUDO DE CASO

Na Seção anterior, um modelo de análise de sentimentos é proposto, utilizando o algoritmo NDiST como modelo de classificação. Neste Capítulo, um estudo de caso foi criado utilizando opiniões referentes a hotéis, na qual uma árvore com divisões binárias será proposta de acordo com a análise do domínio.

Na Seção 6.1, o domínio de hotéis é apresentado, exibindo o resultado de um questionário feito em nossa pesquisa, evidenciando quais estrelas são mais relevantes para os consumidores. Na Seção 6.2, a influência dos *ratings* é discutida com base nos resultados encontrados na seção 6.1. Na Seção 6.3, a árvore binária é descrita e os resultados do método proposto são apresentados, exibindo quais as melhores técnicas de extração e quais os melhores algoritmos.

6.1 O DOMÍNIO DE HOTÉIS

O domínio de hotéis é bem utilizado em trabalhos que estudam o problema de classificação multiclasse e na inferência de *ratings*. Embora a base de dados utilizada tenha sido utilizada a fim de descobrir a latência da opinião de um indivíduo em relação a aspectos de uma entidade e a relação com o *rating* final, outros trabalhos como (DE ALBORNOZ *et al.*, 2011), que utilizam avaliações disponíveis no site *booking.com*, propõem algum método a fim de inferência de *ratings*.

A fim de conhecer o domínio e a influência dos *ratings* na escolha de um hotel, um questionário foi criado a fim de verificar a preferência de usuários no momento da reserva de um hotel. Este questionário foi respondido por 131 pessoas e o mesmo encontra-se no APÊNDICE C .

Analisando o nível de escolaridade dos participantes, 42% declararam possuir nível superior, 23,4% possuíam o grau de mestre e 22,4% possuíam o grau de doutor. Os outros 12,1% possuíam o ensino médio completo. Esses dados são apresentados na Figura 6 (a).

Em relação à faixa etária dos participantes, presente na Figura 6 (b), 52,3% tinham entre 20 e 29 anos, 31,8% tinham entre 30 e 39 anos, 11,2% tinham entre 40 e 49 anos. Apenas 2,8% tinham mais do que 50 anos e somente 1,9% tinham menos de 19 anos.

Quando perguntados sobre frequência com que verificam opiniões e os respectivos *ratings* deixados por outros usuários nos sites de reserva, os participantes, que declararam que

utilizam alguns sites de reserva de hotéis, sempre utilizam estas medidas avaliativas (68,4%), conforme a Figura 7. 23,3% informaram que verificam com frequências as avaliações. Outros 4,5% verificam algumas vezes, 3% raramente verificam e apenas 0,8% não utilizam as avaliações disponíveis.

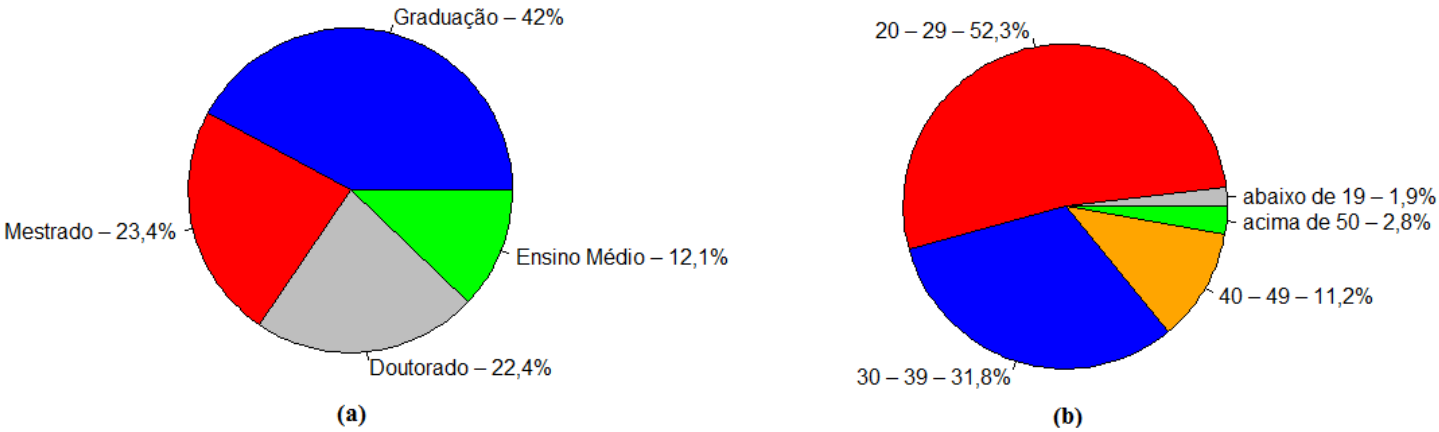


Figura 6. (a) Escolaridade dos participantes do questionário. (b) Faixa etária dos participantes.

Na escolha de um hotel, 65,4% dos participantes informou que o *rating* é a principal medida utilizada no momento de decidir sobre a reserva de um hotel, enquanto outros 34,6% avaliam outros aspectos, como localização, preço e serviços disponíveis. Esse resultado pode ser notado no gráfico de Figura 8.

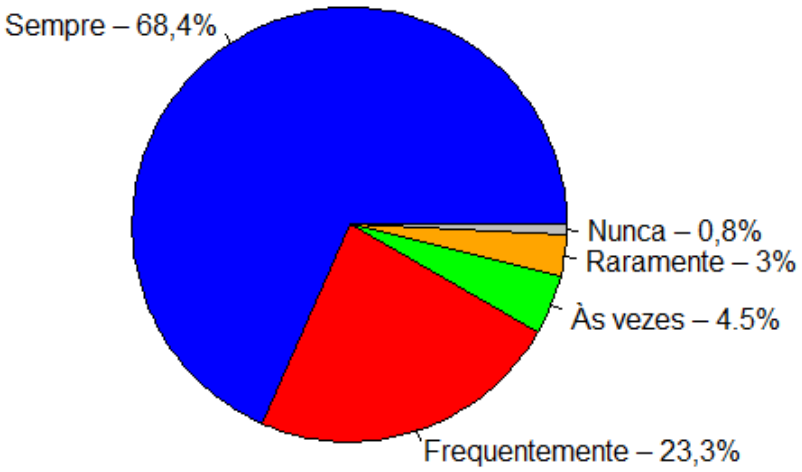


Figura 7. Frequência que os usuários verificam os *ratings* quando vão fazer uma reserva

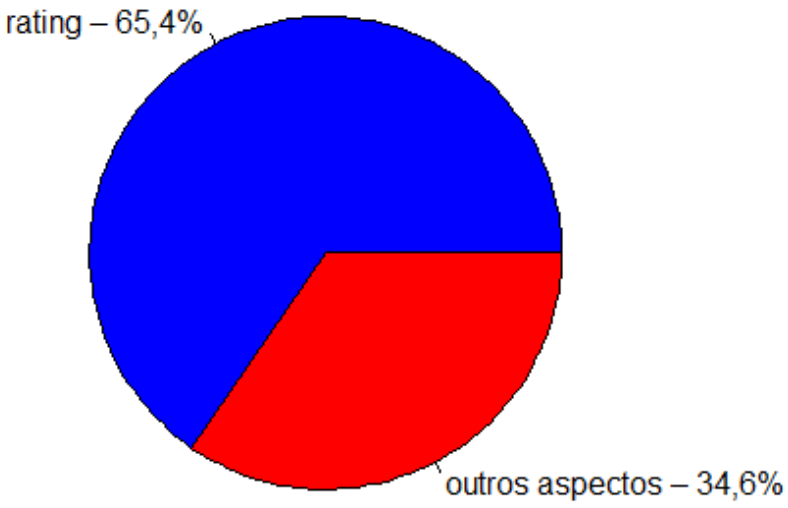


Figura 8. Importância dos *ratings* no momento da reserva de um hotel

Quando analisado a importância da escala de *ratings* baseada em avaliações de usuários, a maioria dos participantes (46%) prefere reservar hotéis que tenham 4 ou mais estrelas, mas que podem avaliar outros aspectos. Outros 39,1% escolhem hotéis com 3 ou mais estrelas, enquanto 14,9% só reservam hotéis com 5 estrelas. Esse resultado vai de encontro com a pesquisa presente no site *PracticalECommerce* e está presente na Figura 9.

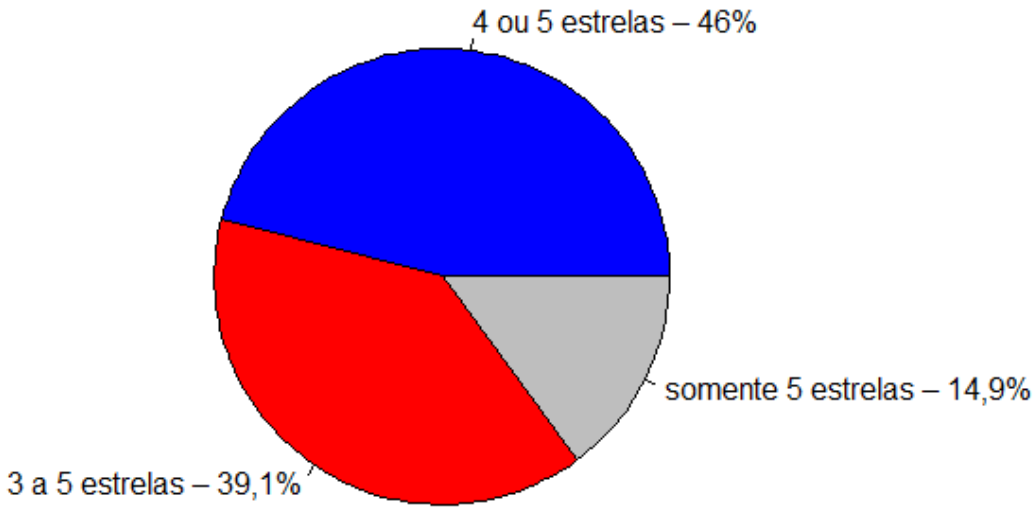


Figura 9. *Ratings* mais importantes na escolha de um hotel

6.2 INFLUÊNCIA DE *RATINGS* NO DOMÍNIO DE HOTÉIS

Com base nas pesquisas citadas na Seção anterior, um processo de divisão pode ser explorado por técnicas de divisão, como o ND. Embora o processo de divisões binárias seja mais custoso em relação ao tempo de execução, já que cada nó é configurado com um algoritmo, a melhor acurácia nos nós intermediários configurados com algoritmos binários pode ser um fator viável quando se trata de inferir notas às opiniões. Isso porque a classificação em várias classes (*ratings*, por exemplo) tem seu uso presente em sites como *Amazom*™, *IMDb*[25]™ e *TripAdvisor*™, sendo, portanto, fundamental inferir *ratings* baseados em opiniões em outros domínios.

Além disso, considerando que duas classes podem ser consideradas muito próximas, como 2 e 3, na qual a acurácia nesta divisão é menor, pode-se considerar que o erro na classificação não seja um fator influente na escolha de um hotel. Isso porque opiniões com 1-3 estrelas, por exemplo, podem não ser consideradas de grande utilidade na análise de um usuário como demonstrado na pesquisa do site *PracticalECommerce* e no questionário realizado neste estudo**.** Essa é uma vantagem em relação ao modelo multiclasse, já que apenas uma etapa é executada e não se tem a garantia de alcançar tais divisões. Logicamente, para cada domínio, essa configuração de árvores binárias (dicotomias) deve ser estudada a fim de se determinar quais opiniões são importantes e, a partir disso, o mesmo modelo de inferência pode ser empregado. Para cada domínio ou dependendo da necessidade do usuário, um modelo de divisões binárias pode ser empregado.

Esse modelo de inferência de *ratings*, além de separar em uma primeira divisão as opiniões recomendadas e as não recomendadas, tem, como objetivo final a classificação de uma opinião com relação à uma classe dentro da escala de *ratings*. Esse processo de divisão pode ser um fator decisivo a fim de evitar os chamados *herdding effects* já que a inferência de um *rating* feita por um usuário pode ser realizada de forma assistida, na qual cada divisão do algoritmo possa sugerir qual a melhor estrela.

6.3 IMPLEMENTAÇÃO DO MODELO

Após a análise feita do domínio de hotéis e da seleção das melhores técnicas de características realizadas no Capítulo 4, uma nova arquitetura para o problema de inferência de *rating* que divida um único classificador em várias etapas de classificação binária baseado

[25] http://www.imdb.com

em dicotomias aninhadas (Nested Dichotomies) [13] é proposta, semelhante à árvore exibida na Figura 10.

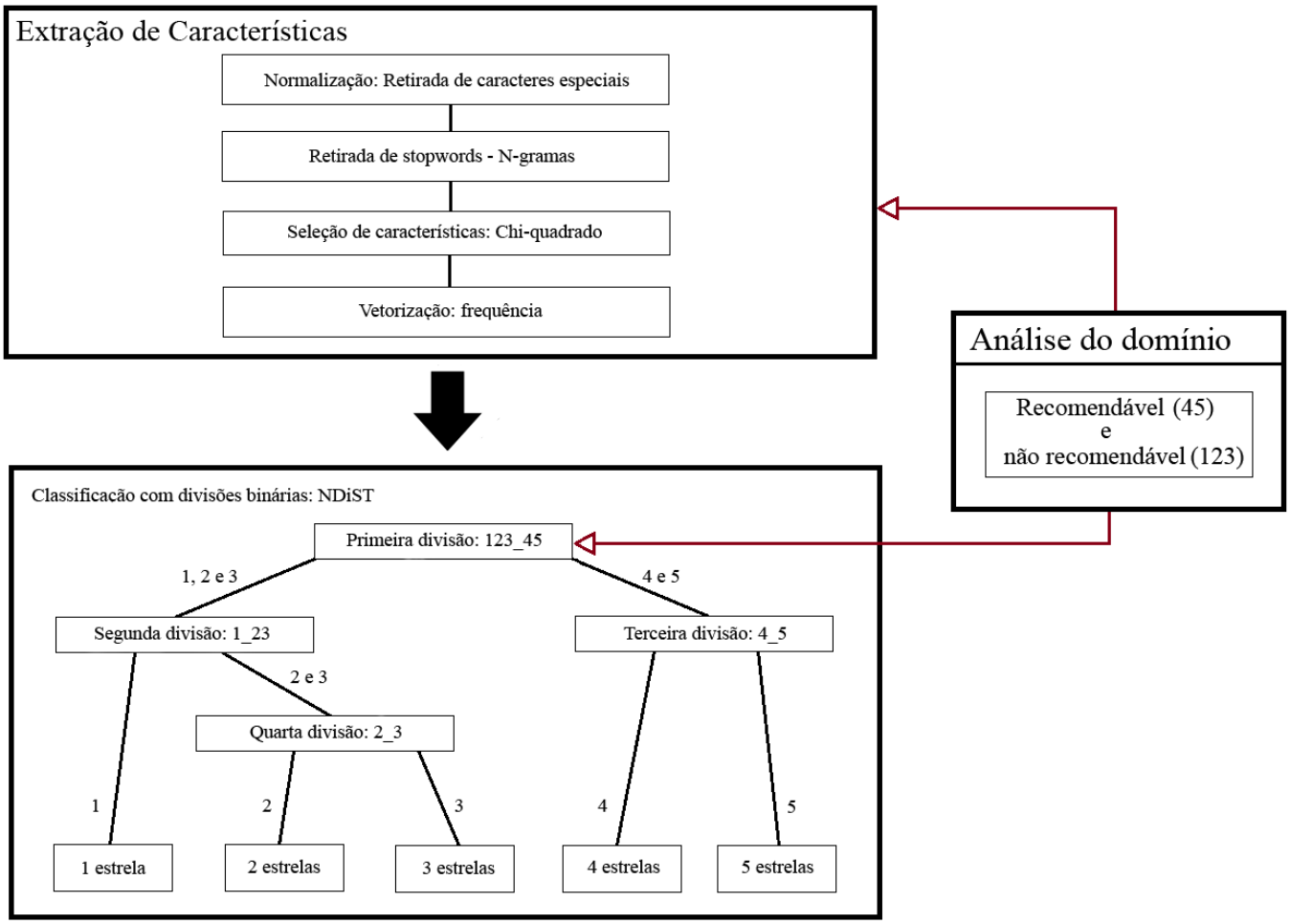


Figura 10. Modelo de divisões binária para o Problema de Inferência de *Ratings*

Na primeira etapa do algoritmo, as opiniões são analisadas a fim de verificar se alguma delas está sem classificação inicial. A partir disso, uma lista das classes existentes é criada e elas são inseridas nos nós da árvore proposta.

Primeiramente, as opiniões são separadas em duas classes principais: recomendadas e não recomendadas. A partir desta classificação, a classe rotulada como não recomendada é dividida em duas categorias: muito ruins e regulares. A classe regular passa por um novo processo de divisão binária gerando classificações ruins e neutras. Já em relação à classe classificada como recomendada, esta será dividida como boa e muito boa. Com isso, cinco classes finais para as opiniões são criadas: muito boas, boas, neutras, ruins e muitos ruins, na qual cada uma corresponde a uma estrela na escala de Likert (1-5).

A fim de comparar os resultados da técnica de dicotomias aninhadas e os resultados obtidos no Capítulo 4, os arquivos criados conforme descrito nas seções 4.2 e 4.3 também serão utilizados. Como será visto, assim como os resultados dos algoritmos nativos, os modelos que utilizavam n-gramas, chi-quadrado e frequência apresentaram melhores resultados do que outras configurações testadas nesta pesquisa.

Na Tabela 16, os n-gramas foram superiores na maioria dos arquivos criados para realizar os testes. Apenas quando o NDiST foi configurado com o SMO nos nós da árvore e o arquivo com 250 características de treinamento foi utilizado, o resultado dos unigramas foi superior aos n-gramas.

Tabela 16. Tabela com a acurácia dos algoritmos NBM e SVM

Naive Bayes Multinomial						
Bag-of-words	**Número de unigramas ou bigramas**					
	250	500	1000	1500	2000	2500
Unigrama	50,1	49,91	51,4	52,63	51,96	56,11
N-grama	**51,56**	**51,23**	**52,97**	**54,33**	**54,37**	**56,6**
SMO						
Unigrama	**51,45**	51,92	53,13	52,49	51,68	51,91
N-grama	50,65	**53,13**	**54,3**	**54,32**	**52,61**	**53,43**

Em relação aos algoritmos de aprendizado, o Naive Bayes Multinomial foi superior em 4 dos 6 arquivos utilizados, sendo superados pelo SMO apenas quando 500 ou 1000 n-gramas foram utilizados. Em relação às técnicas de extração, o Chi também teve resultado superior, como pode ser notado na Tabela 17.

Tabela 17. Tabela com a acurácia dos métodos de seleção utilizados o NDiST

Técnica de seleção: Chi-quadrado				
Bag-of-words	**Algoritmo – Número de unigramas ou bigramas**			
	NBM - 1000	SVM - 1000	J48 – 1000	kNN – 1000
Unigrama	51,4	53,13	40,12	32,69
N-grama	**52,97**	**54,3**	40,6	**34,76**
Técnica de seleção: Ganho de Informação				
	NBM - 1000	SVM - 1000	J48 – 1000	kNN – 1000
Unigrama	46,63	46,16	**41,47**	33,96
N-grama	41,6	42,81	40,57	34,45
Técnica de seleção: Ganho Médio				
	NBM - 1000	SVM - 1000	J48 – 1000	kNN – 1000
Unigrama	34,96	32,75	24,39	30,48
N-grama	28,78	28,41	21,01	29,27

O desempenho desta técnica teve desempenho superior de quase 12% em relação ao ganho de informação, como pode ser notado nos resultados do SVM configurado com n-gramas. Entretanto, para o algoritmo J48, o melhor resultado foi com IG quando configurado com unigramas. Já a técnica de ganho médio de informação apresentou resultados inferiores em todos os algoritmos.

Para avaliar a significância estatísticas dos resultados da Tabela 17, o teste de Friedman (JAPKOWICZ e SHAH, 2011) com o grande de 95% de confiança para a seguinte hipótese H1:

H1: os três métodos de seleção são compráveis. Avaliando os valores encontrados através da ferramenta R, no qual o Chi-quadrado = 13, df = 2, p-value = 0.001503, pode-se concluir que, de acordo com a tabela de Friedman, existe uma diferença significativa entre os três métodos de seleção de características e que a hipótese nula H1 é rejeitada, já que p-value < 0,05. Com o teste de Friedman indicando uma significância estatística, de acordo com (DEMSAR, 2006), o teste de Nemenyi pode ser utilizado para a comparação dos métodos de seleção, utilizando o pacote PMCMR[26].

Avaliando a Tabela 19 gerada, conclui-se que o método Ganho Médio difere significativamente dos outros métodos (CHI e IG), já que 0,0014 e 0,0332 < 0,05.

Tabela 18. Comparações utilizando o teste Nemenyi

	CHI	**IG**
IG	0,5768	-
GR	0,0014	0,0332

Assim como feito no Capítulo 4, a utilização da frequência foi comparada com a técnica *tfidf* que cria pesos para os n-gramas utilizados. Utilizando o resultado dos dois melhores algoritmos de classificação, os arquivos de teste com a frequência dos n-gramas foram superiores em todos os testes, mesmo que esta diferença tenha sido de próxima, como nos arquivos com 500 e 1000 n-gramas, no qual o NBM foi utilizado. Os resultados da acurácia dos arquivos testado estão na Tabela 18.

Na Tabela 20, a matriz com o melhor resultado utilizando a técnica de NDiST é exibida. Assim como no Capítulo anterior, o melhor resultado foi obtido com o Naive Bayes configurado com n-gramas e frequência de termos, utilizando a técnica de extração chi-quadrado. Como pode ser notado, embora a acurácia final seja de 56,6%, a acurácia próxima ultrapassa 90% (92,3%) assim como no uso de algoritmos nativos. Outro detalhe está na alta

[26] https://cran.r-project.org/web/packages/PMCMR/index.html

acurácia de das classes extremas, 1 e 5 estrelas, ultrapassando 75% de acurácia. Isso pode ser justificado pela dificuldade em analisar classes intermediárias, devido a semelhança entre essas classes.

Tabela 19. Tabela com a acurácia da frequência versus modelo *tfidf*

Naive Bayes Multinomial – CHI						
Vetorização	**Número de unigramas + bigramas**					
	250	500	1000	1500	2000	2500
Frequência	**51,56**	**51,23**	**52,97**	**54,33**	**54,37**	**56,6**
TFIDF	48,05	49,13	50,73	48,61	48,65	49,05
SMO – CHI						
Frequência	**50,65**	**53,13**	**54,3**	**54,32**	**52,61**	**53,43**
TFIDF	45,12	46,6	47,17	46,6	45,95	45,44

Tabela 20. Matriz de confusão para a técnica NDiST com o algoritmo NBM: melhor configuração encontrada com 2500 n-gramas

a	b	c	d	e	Acurácia		← classificado como
					Exata	Próxima	
1140	272	61	17	10	0,760	0,941	a=1
419	546	431	91	13	0,364	0,931	b=2
145	364	731	201	59	0,487	0,864	c=3
30	65	202	699	504	0,466	0,937	d=4
22	27	50	283	1128	0,752	0,941	e=5
					0,566	**0,923**	-

A Tabela 21 apresenta a acurácia entre os nós para a divisão binária dentro do modelo de Nested Dichotomies proposto na Figura 7, utilizando os arquivos com 1000 e 2000 características. Como pode ser notado, a melhor divisão é a primeira cuja as opiniões são separadas como recomendadas ou não recomendadas. A partir disso, o desempenho do classificador diminui à medida que as classes agrupadas são divididas. Isso se deve à semelhança entre opiniões com *ratings* próximos, como nas divisões 2 e 3 representadas pelo classificador binário[3]. Todas as etapas atingem acurácia superior a 65%, superando a etapa geral de um classificador multiclasse individual, entretanto, a média final apresentada na tabela não é uma boa medida comparativa tendo em vista que a acurácia final do classificador é dada pelo produtório [57] na Equação 20 da Seção 5.2. Esse resultado serve apenas para exibir o desempenho dos nós da árvore binária que apresenta melhor resultado médio com uma versão do algoritmo Naive Bayes Multinomial utilizado como classificador em cada nó existente.

Tabela 21. Tabela com acurácia de cada nó da árvore proposta

Algoritmo	Binária[1]		Binária[2]		Binária[3]		Binária[4]		Média
	Número de unigramas + bigramas								
	1000	2000	1000	2000	1000	2000	1000	2000	
Naive Bayes	83,41	82,79	74,62	73,36	70,57	67,6	74,28	69,9	73,13
NBM	**87,91**	**88,85**	**80,44**	**80,91**	**73,47**	**71,27**	**78,24**	**73,83**	**77,11**
SVM Linear	87,59	87,17	80,38	77,84	69,9	65,9	76,21	67,87	74,79
SMO	85,11	86,92	80,56	79,22	70,47	66,8	76,79	67,9	75,39

Para a árvore proposta na Figura 10, por exemplo, a probabilidade da classe 2 para uma instância x é dada por:

$$P(c=2|x) = p(c \in \{1,2,3\}|x).p(c \in \{2,3\}|x, c \in \{1,2,3\}).p(c \in \{2\}|x, c \in \{2,3\}).$$

Um dos ganhos deste modelo está em qual nó o resultado já é útil em termos de classificação e inferência de *rating*. Por exemplo, dividir um conjunto de opiniões com 2 e 3 estrelas pode não ser viável tendo em vista a semelhança entre as opiniões e a importância que essa diferença pode fazer na escolha de um hotel. Isso por que, além das estrelas, outros atributos podem ser considerados na escolha e nem sempre uma estrela a mais ou a menos pode ser determinante na escolha final de uma hospedagem. Alguns trabalhos, como (PANG; LEE, 2005) e (GOLDBERG, A. B.; ZHU, 2006) utilizam apenas 4 classes finais.

Mesmo com as técnicas selecionadas no Capítulo 4, os testes anteriores também avaliaram quais técnicas de extração de características e algoritmos de aprendizado foram melhores, entretanto, para o algoritmo NDiST. Pode-se concluir que as mesmas técnicas foram escolhidas, enfatizando a relação entre o domínio e as técnicas de extração e algoritmos escolhidos.

No Capítulo seguinte, os resultados dos algoritmos nativos serão comparados aos resultados obtidos através da técnica proposta NDIST, exibindo gráficos comparativos em relação à acurácia dos mesmos. A partir disso, a metodologia será comparada com os principais resultados obtidos por outros pesquisadores e com os resultados dos algoritmos testados anteriormente no Capítulo 4. Isso pode ser de grande utilidade para a comunidade a fim de gerar *ratings* mais consistentes para os produtos e, para os usuários, indicar como os produtos são realmente avaliados.

CAPÍTULO 7 – AVALIAÇÃO EXPERIMENTAL DO CLASSIFICADOR NDiST

Com a definição do algoritmo com divisões binárias e com os resultados da técnica NDiST descritos nas seções anteriores, esse Capítulo apresenta um comparativo, utilizando gráficos e tabelas dos testes realizados nos dois Capítulos anteriores, com o intuito de avaliar as técnicas de extração e, principalmente, a técnica de dicotomias aninhadas.

As próximas Seções estão divididas em resultados para a acurácia final do modelo, destacando as técnicas que apresentaram o melhor desempenho, e em resultados que destacam o resultado da primeira e principal divisão em relação aos algoritmos nativos, multiclasse adaptados e outras árvores criadas com o NDiST.

7.1 COMPARATIVO DA ACURÁCIA FINAL DOS MODELOS TESTADOS

O primeiro comparativo apresentado na Figura 11 e na Tabela 22 é em relação à acurácia dos algoritmos nativos em relação à árvore proposta no Capítulo anterior. Embora mais algoritmos tenham sido testados, apenas os dois melhores algoritmos foram exibidos. Em relação ao número de n-gramas utilizados no treinamento, os melhores resultados foram com arquivos que utilizam 2500 n-gramas para o modelo proposto. Outro detalhe está no crescimento da acurácia com o aumento dos n-gramas, que pode ser notado em quase todos os algoritmos. Apenas o SMO não apresentou uma linha crescente.

Tabela 22. Acurácia do NDiST e dos melhores algoritmos nativos - CHI – N-gramas

Algoritmo	**Número de unigramas + bigramas**					
	250	500	1000	1500	2000	2500
SMO – NDiST	50,65	53,15	54,3	54,32	52,61	53,45
NBM – NDiST	51,56	51,23	51,4	54,33	54,37	**56,60**
SMO	55,78	57,05	56,68	56,06	55,3	55,74
NBM	55,21	57,54	58,81	59,43	59,89	**60,45**

Como pode ser notado, a acurácia do Naive Bayes Multinomial é a melhor, atingindo 60,45%, sendo superior em quase 4% em relação a melhor acurácia do modelo ND proposto. Embora normalmente inferior, o NDiST chega a superar o modelo SMO com 2500 n-gramas.

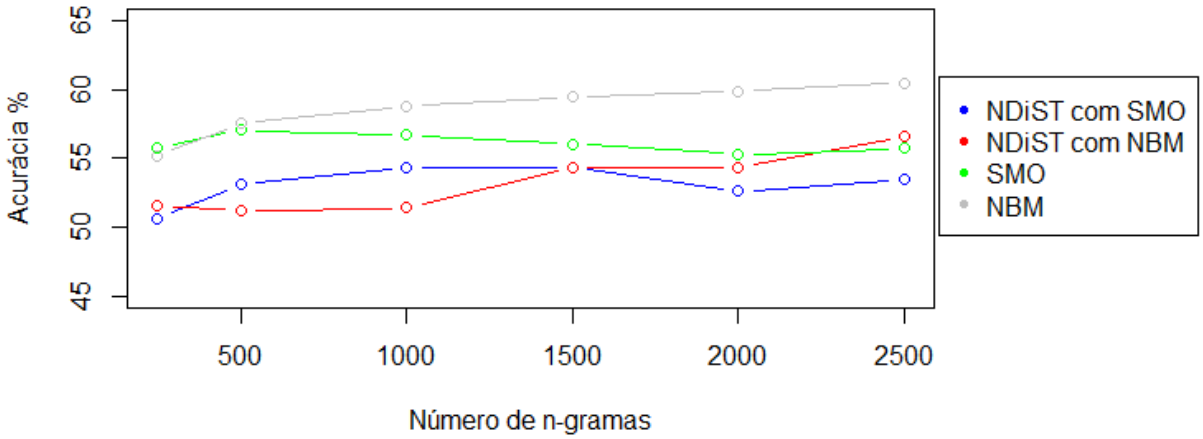


Figura 11. Comparativo da acurácia entre o NDiST e os algoritmos nativos - CHI – N-gramas

A Figura 11 mostra o desempenho final da variação do algoritmo NDiST, comparado com os resultados dos algoritmos nativos testados no Capítulo 4. Embora o algoritmo NDiST seja mais adequado, baseando-se em pesquisas que demonstrem a importância da primeira divisão, algoritmos nativos 1-etapa possuem acurácia final superior quando comparada ao modelo de divisões proposto.

Da mesma forma, embora o CHI sendo quase sempre superior aos outros dois métodos utilizados (IG e GR), o IG foi testado a fim de verificar tanto o desempenho dos algoritmos com a técnica NDiST, bem como a acurácia atingida em relação ao número de características de treinamento, conforme mostra a Tabela 23 e o gráfico da Figura 12. O melhor desempenho foi atingido pelos algoritmos nativos, assim como analisado anteriormente. Os resultados dos algoritmos testados foram bem semelhantes, entretanto, a maior acurácia foi alcançada pelo SMO, atingindo 50,09%.

Tabela 23. Acurácia do NDiST e os algoritmos nativos – IG – n-gramas

Algoritmo	**Número de unigramas + bigramas**				
	250	500	1000	1500	2000
SMO – NDiST	29,6	37,7	42,81	45,71	47,29
NBM – NDiST	24,05	36,96	41,6	47,27	**48,36**
SMO	29,17	40,09	46,45	49,05	50,09
NBM	28,56	34,21	43	49,06	**52,21**

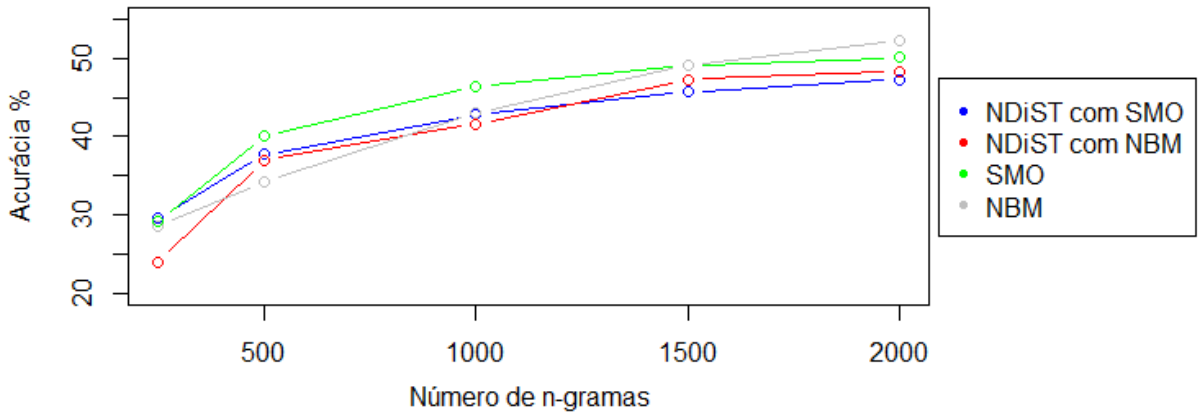


Figura 12. Comparativo da acurácia entre o NDiST e os algoritmos nativos - IG – N-gramas

Na Tabela 24 e na Figura 13, o método de divisões baseado em uma árvore binária é comparado com dois outros modelos de divisão binária: *one-vs-one* e *one-vs-all*. Da mesma forma quando comparado aos algoritmos de classificação nativos, a variação do ND proposta é superada por algum algoritmo, nesse caso pelo modelo OvA, ambos configurados com o Naive Bayes Multinomial. Com o modelo *one-vs-all*, a melhor acurácia foi atingida em todos os testes realizados com os arquivos já citados: 61,01 % com 2500 n-gramas selecionados com o método chi-quadrado.

Em relação ao algoritmo OvO, quando utilizados 2500 n-gramas para o treinamento, o algoritmo NDiST tem a acurácia superior. Outro detalhe está na diferença entre os dois melhores métodos cuja maior valor atinge cerca de 7%, quando 1000 n-gramas são usados no treinamento.

O ganho das divisões baseadas em uma árvore binária está em relação ao número de divisões possíveis para cada modelo. Para o problema 5-classes, 4 iterações são utilizadas no ND, enquanto OvA e OvO utilizam 5 e 10 iterações, respectivamente.

Tabela 24. Acurácia do NDiST e métodos de classificação multiclasse adaptado: OVA e OVO

Algoritmo	**Número de unigramas + bigramas**					
	250	500	1000	1500	2000	2500
NDiST – NBM	51,56	51,23	51,4	54,33	54,37	**56,60**
OvA – NBM	55,44	57,53	58,97	59,72	60,03	**61,01**
OvO – SMO	55,78	**57,05**	56,68	56,06	55,3	55,74

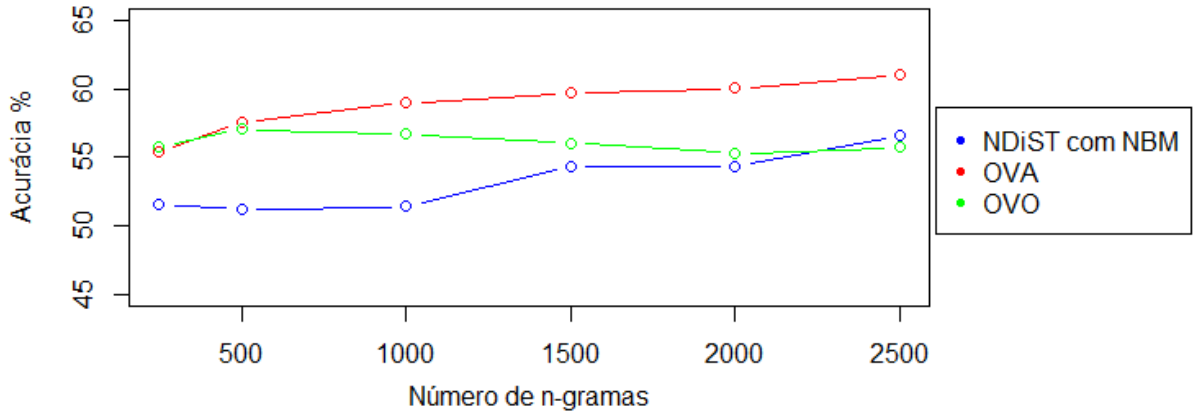


Figura 13. Comparativo da acurácia entre o NDiST e métodos de classificação multiclasse adaptado: OVA e OVO

Na Tabela 25, o algoritmo NDiST utilizando o Naive Bayes Multinomial é comparado com duas técnicas de ensemble: o Bagging e o AdaBoost em relação ao tempo e à acurácia, ambos disponíveis na ferramenta Weka. Esses resultados foram alcançados com o arquivo com 2000 n-gramas e mostra que, assim como verificado nos testes comparativos com outros algoritmos, a acurácia do algoritmo de divisões baseadas em ND é inferior à ambos os modelos aqui exibidos. Apenas em relação ao tempo do AdaBoost o algoritmo com divisões binárias é superior.

Tabela 25. Comparativo da acurácia entre o NDiST e métodos ensemble

Algoritmo	**Tempo**	**Recall**	**Precisão**	**Acurácia**
ND com NBM	0,84	0,566	0,556	56,6
Bagging com NBM	**0,28**	0,597	0,594	59,69
AdaBoost	3,14	**0,599**	**0,597**	**59,89**

Para avaliar a significância estatística entre o algoritmo NDiST e os outros algoritmos, o teste de Wilcoxon (JAPKOWICZ e SHAH, 2011) foi realizado, com base na Hipótese 2:

H2: Utilizar ou não o algoritmo NDiST com o método de seleção CHI apresenta resultados comparáveis em relação aos algoritmos NBM e ao SVM. Nesse caso, a acurácia foi avaliada com um nível de confiança de 95%. Analisando a Tabela Wilcoxon para n = 12, V = 17 e V = 13. Nesse caso, os valores da tabela são maiores que o $T_{Wilcoxon}$ encontrado $T_{Wilcoxon}$ = 0, rejeitando a hipótese H2 de que o algoritmo NDiST é comparável aos algoritmos NBM e SMO.

A Tabela 26 foi utilizada para realizar o teste de Wilcoxon.

Tabela 26. Resultados da Tabela 22 usados no teste Wilcoxon

Domínio	NDiST	Não-NDiST	NDiST – Não-NDiST	Não-NDiST - NDiST	Rank	±Rank
SMO250	50,65	55,78	-5,13	5,13	9	-9
NBM250	51,56	55,21	-3,65	3,65	5	-5
SMO500	53,15	57,05	-3,9	3,9	7	-7
NBM500	51,23	57,54	-6,31	6,31	11	-11
SMO1000	54,3	56,68	-2,38	2,38	3	-3
NBM1000	51,4	58,81	-7,41	7,41	12	-12
SMO1500	54,32	56,06	-1,74	1,74	1	-1
NBM1500	54,33	59,43	-5,1	5,1	8	-8
SMO2000	52,61	55,3	-2,69	2,69	4	-4
NBM2000	54,37	59,89	-5,52	5,52	10	-10
SMO2500	53,45	55,74	-2,29	2,29	2	-2
NBM2500	56,60	60,45	-3,85	3,85	6	-6

7.2 COMPARATIVO DA ACURÁCIA DA PRIMEIRA DIVISÃO DOS MODELOS TESTADOS

Como visto na Seção anterior, a acurácia final do classificador proposto foi inferior à grande parte dos algoritmos e técnicas testadas. Entretanto, quando a primeira e mais importante divisão foi analisada, divisão esta que é utilizada em grande parte dos trabalhos de classificação em relação à polaridade das opiniões, a acurácia do modelo supera, em alguns casos, alguns algoritmos nativos e outros modelos multiclasse adaptados. Isto vai de encontro com as pesquisas que demonstram a importância desta divisão no contexto de análise de sentimentos.

O primeiro comparativo apresentado é em relação à acurácia da primeira e mais importante divisão, tanto de algoritmos nativos quanto da técnica NDiST, utilizando arquivos de treinamento com 1000 e 2000. Nesse caso, três modelos de classificação binária foram

utilizados: o modelo binário¹, no qual as classes 1, 2 e 3 são consideradas não recomendadas e as classes 4 e 5 são recomendadas; o modelo binário², cujas classes 1 e 2 são ruins; e um modelo binário (binário³) representado por um algoritmo nativo onde, nesse caso, a acurácia da divisão é calculada de acordo com a matriz de confusão final. No modelo binário³, o intuito é medir a acurácia em relação ao número de opiniões que foram consideradas recomendas e o número que foi classificada como não recomendada.

Na Tabela 27, quando apenas os algoritmos configurados com a divisão binária (binária¹ e binária²) são comparados, nota-se que o agrupamento utilizando a classe 3 como não recomendada apresenta maior acurácia do que o modelo com a classe 3 considerada recomendada, indo de encontro com a pesquisa do site *PracticalECommerce* e com a maioria dos participantes do questionário de nossa pesquisa. Em relação aos algoritmos nativos, os resultados são bem semelhantes. Para os algoritmos Naive Bayes e Naive Bayes Multinomial, os algoritmos nativos apresentam a acurácia um pouco superior, não chegando a mais de 1% de diferença. Avaliando o SVM, quando a técnica proposta utiliza o modelo linear, os resultados são superiores ao dos algoritmos nativos, ficando um pouco acima de 1% em alguns casos. Para o SMO, o resultado do modelo proposto fica um pouco abaixo do algoritmo nativo. Para cada algoritmo, os resultados exibidos correspondem ao uso de n-gramas e da técnica chi-quadrado.

Tabela 27. Comparativo entre a primeira divisão binária e o modelo multiclasse

Algoritmo	**Binária¹**		**Binária²**		**Binária³**	
	Número de unigramas + bigramas					
	1000	2000	1000	2000	1000	2000
Naive Bayes	83,4	82,8	80,8	80	83,7	83
Naive Bayes Multinomial	**87,9**	**88,8**	**85,9**	**86,1**	**88,4**	**89,1**
SVM Linear	87,6	87,1	84,6	85,3	86,3	86,4
SMO	86,9	86,9	85,1	84,2	87,2	87,4

O modelo de divisões binárias proposto é o mais adequado em relação à influência que *ratings* com poucas estrelas tem, separando, em uma primeira etapa, opiniões boas e ruins. A Tabela 28 e a Figura 14 mostram o desempenho da árvore proposta em relação a outras árvores que tiverem melhor acurácia final, chamadas *BestTree*. Como pode ser notado, mesmo quando comparado com árvores que, posteriormente tem a acurácia superior, na primeira e mais importante divisão, os resultados da árvore proposta são bem semelhantes e,

em alguns casos, chegam a superar as *BestTrees*. Analisando o algoritmo Naive Bayes Multinomial, o algoritmo proposto supera em todos os casos a árvore com melhor acurácia final. Estes resultados foram obtidos utilizando o Chi-quadrado juntamente com n-gramas e a frequência dos mesmos.

Tabela 28. Comparativo entre acurácia do modelo NDiST e das *BestTrees*

Algoritmo	**Número de unigramas + bigramas**					
	250	500	1000	1500	2000	2500
SMO – NDiST	85,6	86,3	86,9	87,9	86,9	86,8
NBM – NDiST	**86,6**	**86,7**	**87,9**	**88,5**	**88,8**	**89,6**
SMO – BestTree	83	85,9	85,9	85,6	85,9	85,8
NBM – BestTree	85,2	87,2	87,8	87,7	88,8	88,3

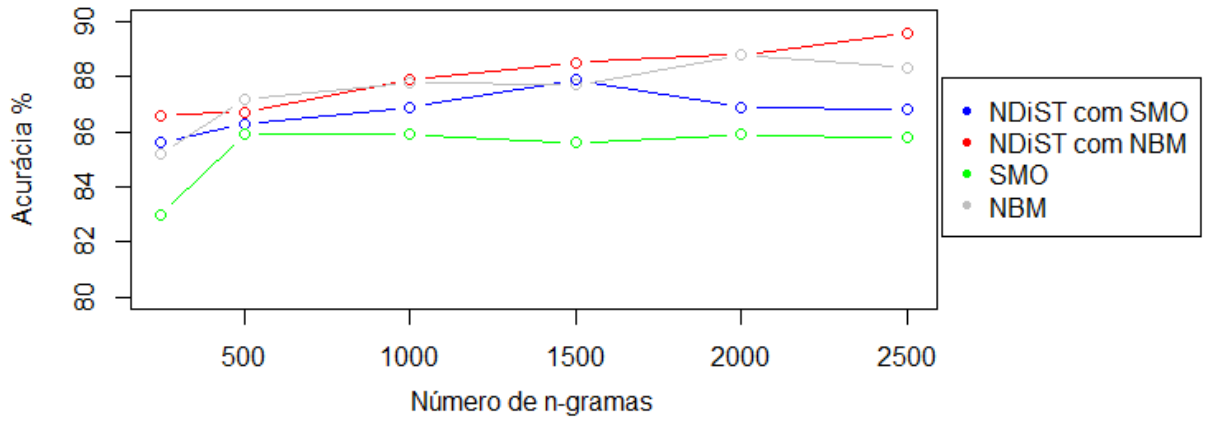


Figura 14. Comparativo entre acurácia na primeira divisão do modelo NDiST e das *BestTrees*

Na Tabela 29 e na Figura 15, o NDiST supera em alguns casos os algoritmos nativos. Isso mostra que, mesmo com a acurácia final sendo inferior, essa primeira divisão atinge um bom desempenho, conforme analisado no questionário proposto e nos resultados de outras pesquisas. Isso mostra que a maior dificuldade está na divisão de classes consideradas próximas, como as classes 2 e 3, por exemplo. Analisando o NBM, para os arquivos com 250 e 2500 características de treinamento, o modelo proposto supera o algoritmo nativo.

Da mesma forma, quando os resultados da primeira divisão são comparados com outras técnicas multiclasse adaptadas, alguns resultados são superiores. Esses resultados estão descritos na Tabela 30 e na Figura 16, na qual o algoritmo Naive Bayes foi utilizado com as

técnicas OvA e OvO. Embora as acurácias sejam muito semelhantes, isso mostra que, para esta primeira divisão, o uso da técnica de dicotomias proposta é viável.

Tabela 29. Comparativo entre primeira divisão do NDiST e dos algoritmos nativos

Algoritmo	**Número de unigramas + bigramas**					
	250	500	1000	1500	2000	2500
SMO – NDiST	85,6	86,3	86,9	87,9	86,9	86,8
NBM - NDiST	**86,6**	86,7	87,9	**88,5**	88,8	**89,6**
SMO	85,9	87	87,2	86,9	87,4	87,9
NBM	86,1	**87,5**	**88,4**	**88,5**	**89,1**	89,4

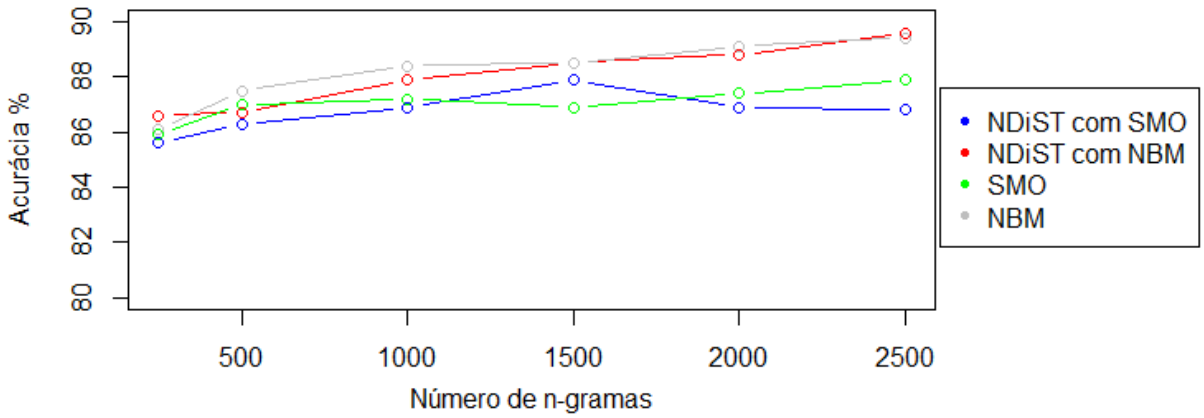


Figura 15. Comparativo entre primeira divisão do NDiST e dos algoritmos nativos

A Figura 17 mostra os resultados da primeira divisão e da acurácia final aproximada, evidenciando que o resultado da primeira divisão fica bem próximo da acurácia final que considera as classes vizinhas *n*-1 como corretas. Essa alta acurácia aproximada indica que, em muitos dos casos, existe um erro pequeno de classificação. Esse fato está relacionado às dificuldades da análise de texto, já que cada usuário tem uma maneira de escrever.

Tabela 30. Comparativo entre primeira divisão do NDiST e dos métodos multiclasse adaptados

Algoritmo	**Número de unigramas + bigramas**					
	250	500	1000	1500	2000	2500
NDiST	**86,6**	86,7	87,9	88,5	88,8	**89,6**
OVA	85,8	**87,4**	**89,7**	**88,8**	**89,1**	89,4
OVO	85,8	87,3	88,3	**88,8**	**89,1**	89,4

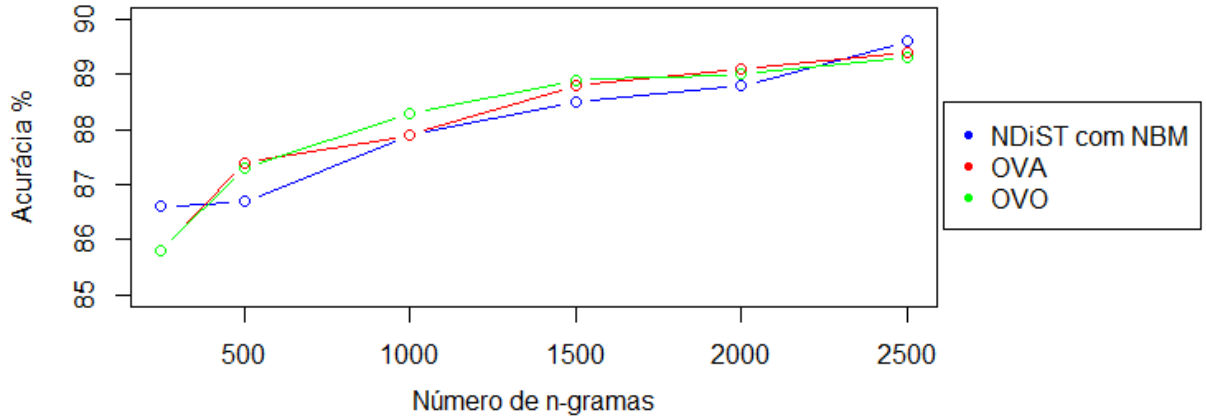


Figura 16. Comparativo entre primeira divisão do NDiST e dos métodos multiclasse adaptados

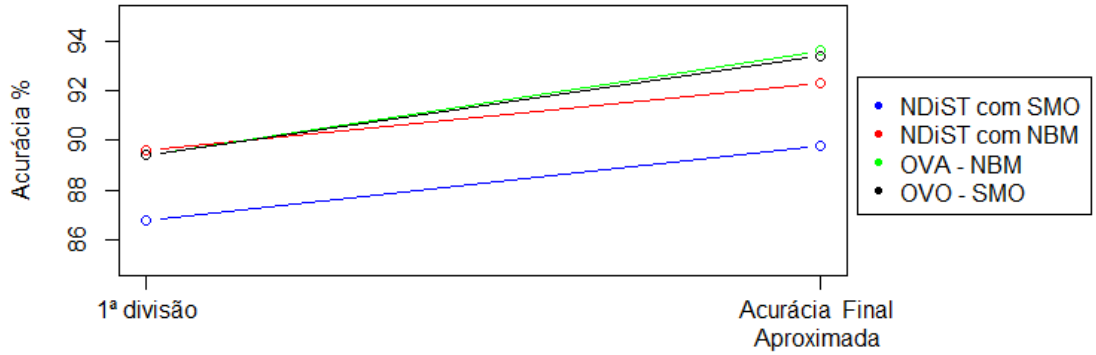


Figura 17. Comparativo entre a primeira divisão e a acurácia final aproximada da árvore proposta e os algoritmos nativos

Para avaliar a significância estatística entre o algoritmo NDiST e os outros algoritmos, o teste de Wilcoxon (JAPKOWICZ e SHAH, 2011) foi realizado, com base na Hipótese 3:

H3: Utilizar ou não o algoritmo NDiST com o método de seleção CHI apresenta resultados comparáveis em relação aos algoritmos NBM e ao SVM. Nesse caso, a acurácia da primeira divisão foi avaliada com um nível de confiança de 95%. Os resultados utilizando a ferramenta R foram *p*-value = 0.1671 e $T_{Wilcoxon}$ = 17. Analisando a tabela de Wilcoxon para n = 12-1=11, tem-se o valor crítico V = 13 e V = 10. Como $T_{Wilcoxon} > V$, a H3 é aceita, ou seja, o algoritmo NDiST pode ser utilizado em relação aos algoritmos NBM e SVM.

A Tabela 31 foi utilizada para realizar o teste de Wilcoxon.

Tabela 31. Resultados da Tabela 29 usados no teste Wilcoxon

Domínio	NDiST	Não-NDiST	NDiST – Não-NDiST	Não-NDiST - NDiST	Rank	±Rank
SMO250	85,6	85,9	-0,3	0,3	4	-4
NBM250	86,6	86,1	0,5	0,5	2	+2
SMO500	86,3	87	-0,7	0,7	9	-9
NBM500	86,7	87,5	-0,8	0,8	10	-10
SMO1000	86,9	87,2	-0,3	0,3	5	-5
NBM1000	87,9	88,4	-0,5	0,5	7	-7
SMO1500	87,9	86,9	1	1	1	+1
NBM1500	88,5	88,5	0	0	Remove	Remove
SMO2000	86,9	87,4	-0,5	0,5	8	-8
NBM2000	88,8	89,1	-0,3	0,3	6	-6
SMO2500	86,8	89,4	-3,4	3,4	11	-11
NBM2500	89,6	89,4	0,2	0,2	3	+3

CAPÍTULO 8 – CONCLUSÕES E TRABALHOS FUTUROS

Essa pesquisa apresentou um estudo do uso de análise de sentimentos e algoritmos de aprendizado de máquina para problemas de classificação de *rating* relacionadas a opiniões sobre produtos e/ou serviços. Ainda, apresentou uma proposta da utilização de um algoritmo de divisões binárias para resolver o problema de classificação multiclasse que, de acordo com algumas pesquisas com problemas multiclasse, se mostra mais adequada do que processos de divisões binárias como *one-vs-all* (OvA) e *one-vs-one* (OvO) em relação ao número de divisões necessárias, embora a acurácia final do modelo OvA seja superior. Embora os resultados finais nem sempre superem os resultados de outros modelos, o algoritmo proposto NDiST, uma variante do algoritmo *Nested Dichotomies* (ND), em determinados conjuntos de dados, supera alguns modelos de classificação multiclasse (FRANK e KRAMER, 2004).

Para o problema de inferência de *ratings*, caso a acurácia final seja considerada, isto é, apenas a inferência de *rating* final, os algoritmos nativos e o modelo OvA são mais indicados por terem maior desempenho. Entretanto, para uma classificação passo a passo, indicando qual a divisão é a melhor por meio de divisões binárias, o algoritmo proposto é capaz de dividir o processo de forma que, em um primeiro agrupamento, as opiniões já sejam, no mínimo, separadas como recomendáveis ou não recomendáveis. Essa divisão é a mais utilizada pelos trabalhos em classificação binária, na qual as opiniões normalmente são agrupadas com 3 classes (1, 2 e 3) consideradas não recomendadas e duas (4 e 5) como recomendadas. Além disso, a utilização de divisões binárias também pode ser útil no aproveitamento dos principais algoritmos propostos nos trabalhos relacionados sem que haja algum tipo de modificação a fim de adaptar um modelo ao problema multiclasse. Apenas os arquivos de treinamento são modificados e isso pode ser realizado por meio do algoritmo NDIST.

8.1 CONTRIBUIÇÕES

Essa pesquisa apresenta um modelo de extração de características baseados nos principais métodos de seleção de características utilizados como referência e que foram selecionados de acordo com a importância e relevância dos trabalhos. Na etapa de extração de características, fundamental na análise de sentimentos de acordo com (LIU, 2012) e (PRUSA; KHOSHGOFTAAR; DITTMAN, 2015), são utilizadas técnicas que foram previamente

empregadas na análise de polaridade, como ganho de informação e o chi quadrado e que obtiveram resultado muito bom na análise multiclasse.

Como principal contribuição, esse trabalho também propõe um modelo de divisão e análise que possa ser utilizado em qualquer domínio de análise multiclasse de opiniões com algoritmos de aprendizado de máquina supervisionados, sendo para 5-classes (*ratings*) ou 10-classes (notas)[27]. Para isso, além de todas as etapas comuns no problema de inferência de *rating* até a utilização de algum algoritmo de aprendizado, um estudo deve ser realizado em torno do domínio a fim de encontrar quais são as melhores divisões possíveis em termos de agrupamento das classes, bem como quais são as melhores TEC's para dado domínio.

8.2 TRABALHOS FUTUROS

Como pode ser visto, o foco deste trabalho estava na proposta de uma nova forma de utilizar algoritmos de aprendizado para o problema de inferência de *ratings*. A fim de melhorar o desempenho dos mesmos, algumas etapas na fase de extração de características foram desconsideradas e foi dada maior importância na área de aprendizado do que no processamento da linguagem natural. Como exemplo de melhorias na fase de processamento textual, pode-se citar a correção de palavras e retiradas de radicais que não alterem o sentido de uma frase como etapas que podem aumentar a acurácia final do modelo de divisão binária.

Outro fator essencial e que pode contribuir para a melhoria do desempenho é analisar a credibilidade do autor dos comentários, detalhe que não foi possível devido à falta de identificação de alguns autores na base de dados utilizada. Isso pode fazer com que um n-grama tenha um peso diferente na execução dos algoritmos de aprendizado. Outras medidas que podem ser feitas são a análise das opiniões em relação aos atributos de uma entidade e melhorias nas configurações do algoritmo NDiST, como a utilização de diferentes algoritmos nos nós da árvore. Além do NDiST, outros algoritmos podem ser modificados, tendo em vista que a ferramenta Weka² foi utilizada apenas com os algoritmos já existentes.

Em trabalhos futuros, esse modelo pode ser testado em diferentes bases de dados a fim de medir o desempenho em relação à outras técnicas e algoritmos já utilizados na literatura. Além disso, esse modelo pode ser utilizado em um sistema de recomendação de *ratings*. Nesse caso, um sistema on-line poderia utilizar o conhecimento adquirido e, em cada etapa, sugerir qual a melhor divisão e indicar o motivo da escolha de tal divisão, apontando quais

[27] http://www.booking.com

palavras foram fundamentais para a sugestão de tal divisão. Isso pode ser de grande utilidade para um usuário onde este poderia verificar quais palavras ou conjunto delas foram essenciais na formação do *rating* para sua opinião. Esse pode ser um passo a fim de evitar que erros de classificação ocorram, onde, em muitos casos, textos subjetivos que poderiam ser analisados como recomendações excelentes são classificadas com estrelas inferiores devido à influência do *overall rating* de um item avaliado. Esse problema, conhecido como *herding effects*, pode ser evitado se uma recomendação de estrelas for feita baseada nas palavras utilizadas em uma opinião.

REFERÊNCIAS

AKSHI KUMAR, T. M. S. **Sentiment Analysis on Twitter.** IJCSI International Journal of Computer Science Issues v. 9, n. 4, p. 372–378, 2012.

ALY, M. **Survey on Multiclass Classification Methods Extensible algorithms.** Neural Networks, n. November, p. 1–9, 2005.

ANDREEVSKAIA, A.; BERGLER, S. **Mining WordNet for fuzzy sentiment: Sentiment tag extraction from WordNet glosses**. Proceedings of EACL, v. 6, p. 209–216, 2006.

BACCIANELLA, S.; ESULI, A.; SEBASTIANI, F. **SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining**. Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10), v. 0, n. November, p. 2200–2204, 2010.

BEINEKE, P.; HASTIE, T.; MANNING, C.; VAITHYANATHAN, S. **Exploring Sentiment Summarization**. Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text Theories and Applications, v. 07, p. 1–4, 2004.

BROOKE, J. **A Semantic Approach to Automated Text Sentiment Analysis**. Simon Fraser University, v. 26, n. 4, p. 118, 2009.

CAMBRIA, E. et al. **New Avenues in Opinion Mining and Sentiment Analysis**. IEEE Intelligent Systems, n. April, p. 15–21, 2013.

CHEN, C.; IBEKWE-SANJUAN, F.; SANJUAN, E.; WEAVER, C.. **Visual Analysis of Conflicting Opinions.** IEEE Symposium on Visual Analytics Science and Technology 2006, VAST 2006 - Proceedings, p. 59–66, 2006.

DAS, S. R; CHEN, M. Y. **Yahoo! For Amazon: Opinion Extraction from Small Talk on the Web**. Proceedings of the 8th Asia Pacific Finance Association Annual Conference, v. XXXIII, n. 2, p. 81–87, 2001.

DAVE, K.; LAWRENCE, S.; PENNOCK, D.. **Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews.** Proceedings of the 12th international conference on World Wide Web, p. 519–528, 2003.

DE ALBORNOZ, J. C.; PLAZA, L.; GERVÁS, P.; DÍAZ, A. **A Joint Model of Feature Mining and Sentiment Analysis for Product Review Rating**. Advances in information retrieval, p. 55–66, 2011.

DEMŠAR, J. **Statistical comparisons of classifiers over multiple data sets.** The Journal of Machine Learning Research, v. 7, p. 1-30, 2006.

DUMAIS, S.; PLATT, J.; HECKERMAN, D.; SAHAMI, M. **Inductive Learning Algorithms and Representations for Text Categorization**. CIKM '98: Proceedings of the seventh international conference on Information and knowledge management, p. 148–155, 1998.

FRANK, E.; KRAMER, S. **Ensembles of Balanced Nested Dichotomies for Multi-Class Problems**. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), v. 3721 LNAI, p. 84–95, 2004.

GO, A.; BHAYANI, R.; HUANG, L. **Twitter Sentiment Classification Using Distant Supervision**. Processing CS224N Project Report, Stanford, v. 150, n. 12, p. 1–6, 2009.

GODBOLE, S.; SARAWAGI, S. **Discriminative Methods for Multi-Labeled Classification**. Advances in Knowledge Discovery and Data, v. LNCS3056, p. 22–30, 2004.

GOLDBERG, A. B.; ZHU, X. **Seeing Stars When There Aren't Many Stars: Graph-Based Semi-Supervised Learning for Sentiment Categorization**. Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing., 2006.

HARRINGTON, P. **Machine Learning in Action**. Manning, 2012.

INFORMATIK, F.; JOACHIMS, T. **Text Categorization with Suport Vector Machines: Learning with Many Relevant Features**. Proceedings of the 10th European Conference on Machine Learning ECML '98, p. 137–142, 1998.

JAPKOWICZ, N.; SHAH, M.. **Evaluating Learning Algorithms: a Classification Perspective**. Cambridge University Press, 2011.

KAESTNER, C. A. A. **Support Vector Machines and Kernel Functions for Text Processing**. Revista de Informática Teórica e Aplicada, p. 1–7, 2013.

KANG, H.; YOO, S. J.; HAN, D. **Senti-Lexicon and Improved Naïve Bayes Algorithms for Sentiment Analysis of Restaurant Reviews**. Expert Systems with Applications, v. 39, n. 5, p. 6000–6010, 2012.

KOHAVI, R. **A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection**. International Joint Conference on Artificial Intelligence, p. 7, 1995.

KONSTAN, J. A.; MILLER, B. N.; MALTZ, D.; HERLOCKER, J. L.; GORDON, L. R.; RIEDL, J. **Grouplens: Applying Collaborative Filtering to Usenet News**. Communications of the ACM, v. 40, n. 3, p. 73–75, 1997.

KONTOPOULOS, E. et al. **Ontology-based Sentiment Analysis of Twitter Posts**. Expert Systems with Applications, v. 40, n. 10, p. 4065–4074, 2013.

KOULOUMPIS, E.; WILSON, T.; MOORE, J. **Twitter Sentiment Analysis : The Good the Bad and the OMG !** Proceedings of the Fifith International AAAI Conference on Weblogs and Social Media, p. 538–541, 2011.

LIKERT, R. **A Technique for the Measturement of Attittudes**. Archives of Psychology, v. 22, n. 140, p. 1–55, 1932.

LIU, B. **Sentiment Analysis and Opinion Mining**. Morgan and Claypool Publishers, n. May, 2012.

LONG, C.; ZHANG, J.; ZHUT, X. **A Review Selection Approach for Accurate Feature Rating Estimation**. Proceedings of the 23rd International Conference on Computational Linguistics: Posters, n. August, p. 766–774, 2010.

LUNARDI, A. D. C.; VITERBO, J.; BERNARDINI, F. C. **Um Levantamento do Uso de Algoritmos de Aprendizado Supervisionado em Mineração de Opiniões**. ENIAC - Natal, RN, 2015.

MAK, H.; KOPRINSKA, I.; POON, J. **INTIMATE: a Web-Based Movie Recommender Using Text Categorization**. Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003), p. 2–5, 2003.

MANNING, C. D.; RAGHAVAN, P. **An Introduction to Information Retrieval**. Cambridge: Cambridge University, p. 1, 2009.

MARTINEAU, J.; FININ, T. **Delta TFIDF : an Improved Feature Space for Sentiment Analysis**. ICWSM, May, p. 258–261, 2009.

MATSUMOTO, S.; TAKAMURA, H.; OKUMURA, M. **Sentiment Classification Using Word Sub-Sequences and Dependency Sub-Trees**. Proceedings of the 9th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, v. 05the9, p. 301–311, 2005.

MCCALLUM, A.; NIGAM, K. **A Comparison of Event Models for Naive Bayes Text Classification**. AAAI/ICML-98 Workshop on Learning for Text Categorization, p. 41–48, 1998.

MITCHELL, T. M. **Machine Learning**. 1997.

MUKHERJEE, S.; BASU, G.; JOSHI, S. **Joint Author Sentiment Topic Model**. Proceedings of the 14th International Conference on Data Mining (SDM), p. 370–378, 2014.

NASUKAWA, T.; YI, J. **Sentiment Analysis : Capturing Favorability Using Natural Language Processing Definition of Sentiment Expressions**. 2nd International Conference on Knowledge Capture, p. 70–77, 2003.

ORTIGOSA, A.; MARTÍN, J. M.; CARRO, R. M. **Sentiment analysis in Facebook and its Application to e-learning**. Computers in Human Behavior, v. 31, p. 527–541, 2014.

PAK, A.; PAROUBEK, P. **Twitter as a Corpus for Sentiment Analysis and Opinion Mining**. LREC, p. 1320–1326, 2010.

PALTOGLOU, G.; THELWALL, M. **A Study of Information Retrieval Weighting Schemes for Sentiment Analysis**. Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, n. July, p. 1386–1395, 2010.

PALTOGLOU, G.; THELWALL, M. **Seeing Stars of Valence and Arousal in Blog Posts**. IEEE Transactions on Affective Computing, v. 4, n. 1, p. 116–123, 2013.

PANG, B.; LEE, L. **Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect oo Rating Scales**. In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (pp. 115-124). Association for Computational Linguistics. v. 3, n. 1, 2005.

PANG, B.; LEE, L. **Opinion Mining and Sentiment Analysis**. Foundations and Trends® in Information Retrieval, v. 2, n. 1, p. 1–135, 2008.

PANG, B.; LEE, L.; VAITHYANATHAN, S. **Thumbs Up? Sentiment Classification Using Machine Learning Techniques**. Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10, n. July, p. 79–86, 2002.

PIMENTA, E. **Abordagens para Decomposição de Problemas Multiclasse : os Códigos de Correcção de Erros de Saída**. Universidade do Porto, 2004

PLATT, J. C. **Fast Training of Support Vector Machines Using Sequential Minimal Optimization**. Advances in kernel methods, p. 185 – 208, 1998.

PRUSA, J. D.; KHOSHGOFTAAR, T. M.; DITTMAN, D. J. **Impact of Feature Selection Techniques for Tweet Sentiment Classification**. The Twenty-Eighth International Flairs Conference, p. 299–304, 2015.

QU, L.; IFRIM, G.; WEIKUM, G. **The Bag-of-Opinions Method for Review Rating**

Prediction from Sparse Text Patterns. Coling, n. August, p. 913–921, 2010.

QUINLAN, J. R. **Induction of Decision Trees**. Machine Learning, v. 1, n. 1, p. 81–106, 1986

RODRÍGUEZ, J. J.; GARCÍA-OSORIO, C.; MAUDES, J. **Forests of Nested Dichotomies**. Pattern Recognition Letters, v. 31, n. 2, p. 125–132, 2010.

RUGGIERI, S. Efficient **C4. 5 [Classification Algorithm]**. Knowledge and Data Engineering, IEEE Transactions on, v. 14, n. 2, p. 438–444, 2002.

SHARMA, A.; DEY, S. **A Comparative Study of Feature Selection and Machine Learning Techniques fo Sentiment Analysis**. RAC'S 2012, p. 1–7, 2012.

TAN, S.; ZHANG, J. **An Empirical Study of Sentiment Analysis for Chinese Documents**. Expert Systems with Applications, v. 34, n. 4, p. 2622–2629, 2008.

TANG, H.; TAN, S.; CHENG, X. **A Survey on Sentiment Detection of Reviews**. Expert Systems with Applications, v. 36, n. 7, p. 10760–10773, 2009.

TURNEY, P. D. **Thumbs Up or Thumbs Down ? Semantic Orientation Applied to Unsupervised Classification of Reviews**. Proceedings of the 40th annual meeting on association for computational linguistics, 2002.

WANG, H.; LU, Y.; ZHAI, C. **Latent Aspect Rating Analysis on Review Text Data**. Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '10, p. 783, 2010.

WANG, T.; WANG, D. **Why Amazon's Ratings Might Mislead You: The Story of Herding Effects**. Big Data, v. 2, n. 4, p. 196–204, 2014.

WIEBE, J. M.; RILLOF, E. **Creating Subjective and Objective Sentence Classifiers from Unannotated Texts**. Computational Linguistics and Intelligent Text Processing, v. 3406, p. 486–497, 2005.

YANG, G.; DESTERCKE, S.; MASSON, M. **Nested Dichotomies with Probability Sets for Multi-class Classification.** ECAI. 2014.

XIA, R.; ZONG, C.; LI, S. **Ensemble of Feature Sets and Classification Algorithms for Sentiment Classification**. Information Sciences, v. 181, n. 6, p. 1138–1152, 2011.

APÊNDICE A – STOPWORDS

a	couldn’t	her	more	she’ll	through	whom
about	did	here	most	she’s	to	why
above	didn’t	here’s	mustn’t	should	too	why’s
after	do	hers	my	shouldn’t	under	with
again	does	herself	myself	so	until	won’t
against	doesn’t	him	no	some	up	would
all	doing	himself	nor	such	very	wouldn’t
am	don’t	his	not	than	was	you
an	down	how	of	that	wasn’t	you’d
any	during	how’s	off	that’s	we	you’ll
are	each	I	on	the	we’d	you’re
aren’t	few	i’d	once	their	we’ll	you’ve
as	for	i’ll	only	theirs	we’re	your
at	from	i’m	or	them	we’ve	yours
be	further	i’ve	other	themselves	were	yourself
because	had	If	ought	then	weren’t	yourselves
been	hadn’t	In	our	there	what	
before	has	into	ours	there’s	what’s	
being	hasn’t	Is	ourselves	these	when	
below	have	isn’t	out	they	when’s	
between	haven’t	It	over	they’d	where	
both	having	it’s	own	they’ll	where’s	
but	he	Its	same	they’re	which	
by	he’d	itself	shan’t	they’ve	while	
can’t	he’ll	let’s	she	this	who	
could	he’s	me	she’d	those	who’s	

APÊNDICE B – EXEMPLOS DE N-GRAMAS PARA O TREINAMENTO

Ganho de Informação	Chi-quadrado	Ganho Médio
disgusting	dirty	parisian
not_impressed	worst	loved
smelly	horrible	divine
filthy	great	pristine
failed	rude	disappointed
worst	told	hotel_great
unacceptable	wonderful	no_hesitation
not_return	terrible	loved
dingy	loved	strolling
drawing	dump	been_more
parisian	awful	excellence
divine	excellent	moment
loved	perfect	rusty
perfect	manager	terrific
great	fantastic	definitely
fantastic	never	small_room
treasure	location	be_back
excellent	disgusting	magical
wonderful	ok	sophisticated
incredible	filthy	exquisite
awesome	clean	rue
favorite	finally	perfect

APÊNDICE C – QUESTIONÁRIO

A influência da opinião de outros usuários na seleção de hotéis em sites de reserva

Um recurso comum para auxiliar usuários de sites de reserva de hotéis a selecionar o item mais apropriado para as suas necessidades é o registro e exibição das opiniões de outros usuários sobre a qualidade do serviço oferecido por hotéis em que já se hospedaram. Essas opiniões podem ser representadas nas seguintes formas:

a) avaliações padronizadas (notas, estrelas ou ratings); e
b) comentários livres, descrevendo impressões positivas ou negativas sobre um hotel.

Essa pesquisa tem como objetivo avaliar a importância que usuários de sites de reserva de hotéis dão às opiniões de outros usuários (seja na forma de avaliações padronizadas ou comentários), no momento em que selecionam um hotel para sua hospedagem.

A pesquisa tem 12 perguntas e serão necessários apenas cerca de 3 minutos para o seu preenchimento. As informações recolhidas serão destinadas estritamente ao estudo descrito e não será solicitado nenhum dado pessoal como nome, telefone, e-mail ou qualquer número de documento como RG, CPF ou passaporte. A divulgação dos resultados obtidos pauta-se no respeito-à privacidade dos participantes, cujo anonimato será preservado. Qualquer dúvida poderá ser encaminhada ao e-mail alexandre.lunardi2@gmail.com para esclarecimento.

*Obrigatório

Para prosseguir, assinale o campo abaixo: *

☐ Concordo com os termos para a realização desta pesquisa

A influência da opinião de outros usuários na seleção de hotéis em sites de reserva

*Obrigatório

Parte A: Perfil do participante

1 - Qual a sua faixa etária? *

- 19 anos ou menos
- 20 a 29 anos
- 30 a 39 anos
- 40 a 49 anos
- 50 a 59 anos
- 60 anos ou mais

2 - Qual o seu maior nível de escolaridade (completo)? *

- Ensino Fundamental I (1º ao 5º ano)
- Ensino Fundamental II (6º ao 9º ano)
- Ensino Médio (1º ao 3ºano)
- Ensino Superior
- Especialização
- Mestrado
- Doutorado

A influência da opinião de outros usuários na seleção de hotéis em sites de reserva

*Obrigatório

Parte B: A importância das avaliações padronizadas (notas, estrelas ou ratings)

3 - Ao selecionar um hotel em um site de reserva, você costuma verificar as avaliações padronizadas (notas, estrelas ou ratings) dos hotéis que pretende reservar? *

- Sempre
- Quase sempre
- Às vezes
- Raramente
- Nunca

4 - Considerando a escala de 1 a 5 estrelas, que é utilizada na maioria dos sites para a avaliação da qualidade de um hotel, como você se enquadra? *

- Só reservo hotéis avaliados com 5 estrelas
- Prefiro reservar hotéis avaliados com 5 estrelas, mas posso optar por inferiores, dependendo de outros aspectos
- Só reservo hotéis avaliados com 4 ou 5 estrelas
- Prefiro reservar hotéis avaliados com com 4 ou 5 estrelas, mas posso optar por inferiores, dependendo de outros aspectos
- Só reservo hotéis avaliados com 3, 4 ou 5 estrelas
- Prefiro reservar hotéis avaliados com 3, 4 ou 5 estrelas, mas posso optar por inferiores, dependendo de outros aspectos
- Reservo preferencialmente hotéis com boas avaliações, mas levo em consideração outros atributos antes de decidir
- Levo em consideração outros atributos para decidir, mas verifico as avaliações dos hotéis em que pretendo me hospedar
- Não verifico as avaliações dos hotéis

5 - Uma estrela a mais ou a menos na classificação de um hotel faz diferença na sua escolha? *

- Sim
- Não
- Às vezes, dependendo de outros atributos

A influência da opinião de outros usuários na seleção de hotéis em sites de reserva

*Obrigatório

Parte C: A importância dos comentários individuais de outros usuários

6 - Ao selecionar hotéis em um site de reserva, você costuma ler os comentários individuais de outros usuários que já se hospedaram nos hotéis que você pretende reservar? *

- Sempre
- Quase sempre
- Às vezes
- Raramente
- Nunca

7 - As opiniões expressas nos comentários dos outros usuários costumam influenciar a sua decisão sobre reservar ou não um determinado hotel? *

- Sempre
- Quase sempre
- Às vezes
- Raramente
- Nunca

8 - Com relação às avaliações dos hotéis e os comentários individuais dos usuários, como você se posiciona? *

- Percebo que os hotéis bem avaliados sempre recebem comentários positivos, e vice-versa
- Percebo que alguns hotéis bem avaliados às vezes recebem comentários negativos, e vice-versa
- Percebo que muitos hotéis bem avaliados recebem comentários negativos, e vice-versa
- Não vejo relação entre as avaliações e os comentários dos usuários

A influência da opinião de outros usuários na seleção de hotéis em sites de reserva

*Obrigatório

Parte D: Outros atributos importantes na seleção de um hotel

9 - Quais são os outros atributos e informações importantes que você leva em consideração na escolha de um hotel? *

(selecione todos os itens que achar necessário)

- [ ] Nenhum
- [ ] Preço
- [ ] Localização
- [ ] Disponibilidade de wifi
- [ ] Qualidade do marca/rede de hotéis
- [ ] Outras facilidades (fitness, piscina, etc)
- [ ] Outro:

A influência da opinião de outros usuários na seleção de hotéis em sites de reserva

*Obrigatório

Parte E: Questões complementares

10 - Em quantos sites, em média, você busca um hotel antes de realizar uma reserva? *

- ○ Dois ou menos
- ○ Entre três e quatro
- ○ Cinco ou mais

11 - Quais sites você costuma utilizar para reservar hotéis? *

- ☐ Booking
- ☐ CVC
- ☐ Decolar
- ☐ Hoteis.com
- ☐ Submarino Viagens
- ☐ TripAdvisor
- ☐ Trivago
- ☐ Outro: _____

12 - Qual o site mais utilizado por você no momento de reservar um hotel? *

- ○ Booking
- ○ CVC
- ○ Decolar
- ○ Hoteis.com
- ○ Submarino Viagens
- ○ TripAdvisor
- ○ Trivago
- ○ Outro: _____

Se desejar fazer algum comentário adicional, utilize o campo abaixo:

_____