Fábio Assunção Berlim Camelo

Detecção Automática de Discursos de Ódio em Comentários de Jornais Online

Niterói, RJ, Brasil

2017

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

C181 Camelo, Fábio Assunção Berlim
Detecção automática de discursos de ódio em comentários de jornais online / Fábio Assunção Berlim Camelo. – Niterói, RJ : [s.n.], 2017.
86 f.

Projeto Final (Bacharelado em Ciência da Computação) – Universidade Federal Fluminense, 2017.
Orientador: Aline Marins Paes Carvalho.

1. Aprendizado de máquina. 2. Mineração de dados (Computação). 3. Minorias sexuais. 4. Mídia social. 5. Jornal eletrônico. I. Título.

CDD 006.31

Fábio Assunção Berlim Camelo

Detecção Automática de Discursos de Ódio em Comentários de Jornais Online

Trabalho submetido ao Curso de Bacharelado em Ciência da Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Universidade Federal Fluminense

Instituto de Computação

Departamento de Ciência da Computação

Orientador: Profa. Aline Marins Paes Carvalho

Niterói, RJ, Brasil

2017

Fábio Assunção Berlim Camelo

Detecção Automática de Discursos de Ódio em Comentários de Jornais Online

Trabalho submetido ao Curso de Bacharelado em Ciência da Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Trabalho aprovado. Niterói, RJ, Brasil, 12 de Dezembro de 2017:

Profa. Aline Marins Paes Carvalho
Orientadora – Universidade Federal Fluminense

Prof. Alexandre Plastino de Carvalho
Universidade Federal Fluminense

Profa. Flávia Cristina Bernardini
Universidade Federal Fluminense

Niterói, RJ, Brasil
2017

Agradecimentos

Os agradecimentos principais são direcionados ao meu pai, que sai para trabalhar às 4 da manhã e chega por volta das 8 da noite, sempre enfrentando muitas dificuldades para manter a mim e aos meus irmãos. À minha mãe que sempre me apoiou e me ensinou a ser uma pessoa decente e responsável. Aos meus irmãos, que sei que um dia eles seguirão os meus passos. A todos os meus amigos, especialmente Manuel e Suzana, que me acompanharam nos momentos mais difíceis da minha graduação. À FAPERJ e ao CNPQ, por apoiarem meus estudos com bolsas de iniciação científica. Agradeço à UFF, principalmente pela oportunidade de morar na moradia estudantil, que me propiciou um crescimento pessoal ao conviver com tantas pessoas diferentes. À professora Aline Paes, que sempre me orientou, me levando para o melhor caminho. Ao professor Dante Corbucci, pelos seus desafios que me fizeram superar a distância de conhecimento que tinha dos outros alunos. Ao professor Luis Antonio, que apresentava um senso de responsabilidade inspirador. Ao professor Diego Gimenez, que mostrou uma organização impecável se tornando um exemplo a ser seguido. Aos professores Flávia Bernardini e Alexandre Plastino, que gentilmente aceitaram fazer parte desta banca avaliadora.

"É verdade o que você disse.
Humanos são fracos e morrem facilmente.
Mas não importa o quão fraco,
não importa o quanto somos torturados,
não importa quanta dor nós sentimos,
nós ainda queremos viver."
(Gatts - Berserk)

Resumo

As mídias sociais estão cada vez mais interativas, incluindo ferramentas que permitem que o usuário colabore com a criação do conteúdo nela exposto. Muitos usuários se aproveitam dessa funcionalidade para divulgar conteúdo ilícito ou criminoso. No Brasil, nota-se uma dificuldade por parte das mídias sociais e das políticas públicas na remoção, e até mesmo na identificação, efetiva deste conteúdo. Caso não seja removido, este conteúdo será visto por cada vez mais pessoas e poderá ser propagado pela internet, atingindo um número maior de vítimas, e incentívando a ocorrência de outros crimes. Objetivando ajudar na redução da popularização de tal prática, esse trabalho se propõe a criar uma ferramenta implementada com técnicas de Aprendizado de Máquina, para detectar automaticamente um dos crimes mais comuns nessas mídias, o discurso de ódio. Esse tipo de crime geralmente é voltado aos grupos mais vulneráveis e marginalizados da sociedade, e seus efeitos nocivos incluem o aumento da exclusão social e da violência praticada contra esses grupos. A avaliação da ferramenta teve como foco a detecção de discursos de ódio inseridos em comentários de um jornal online, direcionados a pessoas que pertençam à comunidade Lésbicas, Gays, Bissexuais, Travestis, Transexuais e Transgêneros (LGBT). Resultados preliminares demonstraram que o detector conseguiu aprender os padrões de textos de algumas formas de discurso de ódio a ele apresentados. Porém, após testes com notícias atuais, com outros padrões de discursos de ódio, o comportamento da ferramenta se deteriora, o que indica a necessidade de atualização contínua do modelo. Contudo, foi possível concluir que a metodologia implementada funciona, e que pode servir de base para a criação de outros detectores de discursos de ódio em mídias digitais em geral.

Palavras-chaves: Aprendizado de máquina, LGBT, Discurso de ódio, Mineração de dados, Mídias sociais, Jornais Online.

Abstract

Social media is increasingly interactive, including tools that allow the user to collaborate and create content exposed in it. Many users take advantage of this feature to spread illicit or criminal content. In Brazil, we notice a difficulty on the part of social media and public policies in the removal, and even in the effective identification of this content. If not removed, this content will be seen by more and more people, and can be spread by the internet, reaching a greater number of victims, and encouraging the occurrence of other related crimes. Aiming at helping on the reduction of the popularization of such crimes, this work proposes a tool embedded with Machine Learning techniques to automatically detect one of the most common crimes in these media, the hate speech. This type of crime is generally targeted at the most vulnerable and marginalized groups in society, and its harmful effects include increasing social exclusion and violence against such groups. The evaluation of the tool was focused on the detection of hate speech in comments from an online newspaper, aimed at the people belonging to the lesbian, gay, bisexual, transvestite, transsexual and transgender (LGBT) community. Preliminary results showed that the detector was able to learn the text patterns of some forms of hate speech. However, after additional tests with new types of hate speech, the behavior of the tool deteriorates, which indicates the need for continuously updating the model. Nevertheless, it was possible to conclude that the implemented methodology is useful to detect some forms of hate speech and that it can serve as a basis for the development of other hate speech detectors in digital media in general.

Keywords: Machine learning, LGBT, Hate Speech, Data mining, Social media, Online Newspaper.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

SVM	*Support Vector Machines*
AM	Aprendizado de Máquina
LGBT	Lésbicas, Gays, Bissexuais, Travestis, Transexuais e Transgêneros.
TF-IDF	*Term Frequency–Inverse Document Frequency*
ROC	*Receiver Operating Characteristic*
BOW	*Bag Of Words*
IA	Inteligência Artificial
TED	*Technology, Entertainment, Design*
PIDCP	Pacto Internacional sobre Direitos Civis e Políticos
HTML	*Hypertext Markup Language*
HTTP	*HyperText Transfer Protocol*
JSON	*JavaScript Object Notation*
XML	*eXtensible Markup Language*
CSV	*Comma-Separated Values*
CRISP-DM	*Cross Industry Standard Process for Data Mining*
LDA	*Latent Dirichlet Allocation*
LSA	*Latent Semantic Analysis*
WEKA	*Waikato Environment for Knowledge Analysis*

Sumário

1 Introdução

O Brasil vive hoje uma turbulência política, onde alguns políticos lutam para manter algum poder, enquanto outros lutam para se salvar de denúncias de corrupção (CORRÊA, 2017),(BRETAS, 2017). No meio disso, temos uma população cada vez mais dividida e extremismos ideológicos cada vez mais presentes. Tal polarização, segundo Leandro Karnal (KARNAL, 2017), é caracterizada por pessoas que se insultam muito, se adjetivam muito, pensam muito pouco e negam ao outro o direito ao debate. Ou seja vemos um aumento no número de pessoas proferindo ódio contra quem pensa diferente de si.

Discutivelmente, esse comportamento é mais frequente quando o usuário se comunica através da internet, incentivado pela sensação de liberdade que o uso dela provê (REIS et al., 2016). Podemos imaginar a proporção deste problema no Brasil que, de acordo com os dados do projeto Digital in 2017(KEMP, 2017), possui 139,1 milhões de usuários de internet e possui a população que passa mais tempo na internet.

Podemos considerar que as pessoas que realizam esse tipo de discurso na internet pertencem a dois grupos: (1) aquelas que entram nas aplicações web e realizam o discurso de ódio através de ferramentas disponíveis da aplicação, (2) e aquelas que geram conteúdo, muitas vezes baseados em mentiras, com intuito de atrair mais pessoas ao seu modo de pensar, criando assim mais pessoas do primeiro grupo. Em ambos os casos, as vítimas desse tipo de discurso tem sua honra violada, o seu psicológico afetado, e podem até sofrer violência física no seu cotidiano, como consequência do ódio proferido a elas, e manifestado na Internet. Uma noticia recente(RJTV, 2017) exemplifica o efeito do discurso de ódio, o caso do assassinato da moradora de rua Fernanda Rodrigues dos Santos em Copacabana, realizado por um lutador de MMA e estudante de medicina, onde a polícia investiga se os dois presos estariam ligados a um grupo de extermínio de moradores de rua. Nesse tipo de grupo, a superioridade entre grupos de pessoas(SILVA et al., 2011) é doutrinada e a violência contra aqueles considerados como inferiores é incentivada. Essa suposta sensação de superioridade é vista nos mais diversos crimes, como quando travestis são mortas de forma brutal, e leitores da notícia postam comentários apoiando o crime, justificando com afirmações, por exemplo de que travestis são aberrações.[1]

O combate mais comum a esse tipo de crime é feito através de sistemas de denúncias, onde um usuário denuncia o conteúdo gerado por outro usuário. Muitos usuários desconhecem tal recurso, enquanto outros não conseguem se decidir sobre quais conteúdos devem ser de fato denunciados. Algumas pessoas conseguem utilizar tais sistemas de

[1] https://twitter.com/tibianmichael/status/875515842015965186

denúncias de forma maliciosa, reunindo muitos usuários para denunciar conteúdos que não ferem os termos de uso da aplicação, e que acabam sendo removidos pela quantidade de denúncias.

Com foco no problema de proliferação e divulgação maliciosa desse tipo de comentário, e a dificuldade de descobri-los de forma automática, propomos nesse trabalho uma ferramenta computacional cujo objetivo é auxiliar no combate de tais tipos de discurso, nomeados de discursos de ódio (BRUGGER, 2007). Para tanto, este trabalho utiliza técnicas de Aprendizado de Máquina(MITCHELL; HILL, 1997), cujo paradigma é bem diferente das técnicas de programação convencionais. Para aplicar um algoritmo de aprendizado, é requerido que sejam fornecidos dados, e a partir de tais dados, o algoritmo de aprendizado induzirá um modelo para a resolução do problema. No caso da detecção de discurso de ódio a solução envolvendo aprendizado de máquina é mais eficiente que uma solução canônica como busca por palavras chave, onde todo discurso que contém determinada palavra chave é considerado um possível candidato a discurso de ódio. Nesse caso, para desenvolver uma ferramenta canônica, o programador teria que estabelecer previamente ma lista com as possíveis palavras chaves, o que pode causar uma certa dificuldade, pois os termos que caracterizam um discurso de ódio estão sempre mudando. O algoritmo de Aprendizado de Máquina, por outro lado, aprenderia os novos termos a partir de novos dados. Outro problema que poderia surgir com a abordagem canônica é que não se saberia ao certo se o contexto em que a palavra-chave se encontra de fato constitui um discurso de ódio, e poderia ser difícil definir todas as particularidades envolvendo tal contexto.

Vale ressaltar que não esperamos com esse trabalho *resolver* o problema inerente associado ao discurso de ódio, mas apenas identificá-los de forma automática, isso porque uma redução efetiva de emissores de discurso de ódio só seria possível envolvendo muitos outros fatores, externos a sistemas computacionais, como uma melhor educação, onde fosse ressaltada a importância do respeito às diferenças, mudanças culturais drásticas, entre outros.

1.1 Objetivos

O objetivo deste trabalho é desenvolver uma aplicação capaz de classificar um comentário como contendo ou não um discurso de ódio. Para avaliar a solução, realizaremos testes com a ferramenta desenvolvida no trabalho, utilizando um (1) ambiente de proliferação de discurso de ódio específico e (2) um grupo específico a quem o discurso de ódio é proferido. No item (1), os dados utilizados nos experimentos deste trabalho foram extraídos de comentários postados por usuários diversos em notícias divulgadas em um jornal online bastante popular no Brasil, a saber, o Jornal O Globo. Foram coletadas diversas notícias e os comentários realizados na mesmas, a fim de obter exemplos de discurso

de ódio e o contexto no qual eles estão inseridos. Também avaliaremos alguns aspectos associados ao discurso de ódio, bem como as implicações de sua remoção automática, aspectos do jornalismo online e como o contexto da notícia ajuda a detectar o discurso de ódio. No item(2), coletaremos discursos de ódio direcionados a um grupo específico de pessoas, a saber, a comunidade LGBT (Lésbicas, Gays, Bissexuais, Travestis, Transexuais e Transgêneros), uma vez que os padrões e jargões usados em um comentário podem variar de acordo com o tipo de grupo ao qual ele é proferido. Além disso, a classificação de um comentário como contendo um discurso de ódio pode variar, dependendo do grupo de pessoas que está sendo ofendido.

A ferramenta desenvolvida deve ter a habilidade de analisar o contexto ao qual o comentário está associado, e determinar se o conteúdo do comentário é um discurso de ódio contra um grupo específico. As técnicas utilizadas para o desenvolvimento da ferramenta foram escolhidas dentre aquelas que são usadas com sucesso em aplicações de classificação de sentenças(TELES; SANTOS; SOUZA, 2016) (TRIVEDI et al., 2015).

1.2 Proposta de Solução

Para alcançar o objetivo apresentado, a primeira tarefa é a obtenção dos exemplos de discurso de ódio. Para isto, propomos realizar a coleta de comentários de jornais online, a partir de notícias relacionadas a uma possível situação envolvendo grupos discriminados. Dessa forma, aumentamos as possibilidades de obter comentários que contenham discurso de ódio.

A segunda tarefa envolve a anotação dos comentários em uma dada classe, dado que nem todos os comentários conterão de fato discursos de ódio. Para tanto, o ideal é que os anotadores humanos pertençam ao grupo para o qual o discurso de ódio está direcionado, uma vez que essas pessoas terão mais propriedade em definir o que as ofende ou não. Para facilitar o processo de anotação, foi desenvolvida uma interface web que auxilie pessoas do grupo alvo a classificarem os comentários de forma voluntária. Nesta interface um mesmo comentário pode ser anotado por vários usuários, um comentário anotado por dois o mais usuários possue uma confiabilidade maior. A ferramenta é capaz de decidir qual a classe final de um comentário dado todas as anotações que o mesmo recebeu. Para que o contexto também seja levado em consideração, além dos comentários, também é fornecido o título da notícia.

Finalmente, os comentários anotados são fornecidos como exemplos de treinamento, juntamente com o título da notícia, de forma que ambos os anotadores e o programa de aprendizado tenham acesso aos mesmos dados. Os comentários foram anotados com as seguintes classes 'discurso de ódio', 'discurso ofensivo' ou ' discurso neutro' (DAVIDSON et al., 2017). O discurso de ódio é motivado por aspectos passíveis de discriminação, que

seus alvos possuem; já o discurso apenas ofensivo não leva esses fatores em consideração.

Os dados coletados e anotados devem passar por técnicas de pré-processamento de textos, onde símbolos irrelevantes são removidos, e os dados são convertidos a um formato aceitável pela técnica de Aprendizado de Máquina(CAMILO; SILVA, 2009). Para a tarefa de classificação de texto realizada nesse trabalho, utilizamos o formato documento-termo, gerado pela técnica *Bag-of-Words* (BOW) (TAN; STEINBACH; KUMAR, 2015). As técnicas de Aprendizado de Máquinas escolhidas foram *Support Vector Machines* (SVM) (LORENA; CARVALHO, 2007) e *Naive Bayes*(HAN; KAMBER, 2006). Dos diversos *frameworks* que implementam estas técnicas, selecionamos o SciKit-learn(PEDREGOSA et al., 2011), por ser repleto de técnicas de Aprendizado de Máquina, tratamento dos dados e avaliação de resultados. Essa *framework* possui uma vasta documentação e é implementada em python, uma linguagem de programação simples e poderosa.

1.3 Organização do Texto

O restante deste trabalho está organizado como descrito a seguir. No Capítulo 2 será apresentada a fundamentação teórica relativa ao trabalho,contendo conceitos indispensáveis para o entendimento e validação da ferramenta desenvolvida. No Capítulo 3 descreve a metodologia empregada no desenvolvimento da ferramenta, bem como, aspectos importantes de sua implementação. No Capítulo 4 são apresentados os resultados experimentais obtidos a partir da ferramenta desenvolvida. Finalmente, no Capítulo 5 expõe as conclusões, limitações e possíveis trabalhos futuros acerca da solução proposta.

2 Fundamentação Teórica

Neste capítulo são abordados conceitos associados ao problema apresentado, a saber, conceitos de mídias digitais, mídias sociais e redes sociais na internet, com foco no jornalismo online, bem como conceitos relacionados a discursos de ódio. Para um melhor entendimento das técnicas utilizadas no desenvolvimento da ferramenta, apresentamos conceitos relacionados a Mineração de Dados e Aprendizado de Máquina. Por fim, são brevemente descritos trabalhos relacionados, apontado em como eles se diferem do apresentado no presente trabalho.

2.1 Mecanismos Online de Difusão de Informação e de Conexão de Pessoas

Hoje, a maior parte dos brasileiros está conectada à internet, seja pelo computador ou por meio de dispositivos móveis. O mundo digital já se tornou parte do dia a dia dos brasileiros, que gastam boa parte de suas horas vagas na internet(KEMP, 2017). Esse comportamento também é justificado pelo fato que atualmente a maioria das informações que queremos na internet, por meio de mecanismos de busca ou através dos outros usuários aos quais estamos conectados. Nesta seção analisamos os principais mecanismos de difusão de informações e de conexão de pessoas disponibilizados na internet. Para tanto, a seguir são apresentados os três mecanismos principais de difusão de informações na Internet, a saber, mídias digitais, mídias sociais e redes sociais, e a relação entre eles. Na Figura 1 são exibidos graficamente como estes três conceitos se relacionam, de forma geral. Nela podemos ver o relacionamento representados por conjuntos, onde redes sociais é um subconjunto de mídias sociais, ou seja todo rede social é uma mídia social, com características especificas.

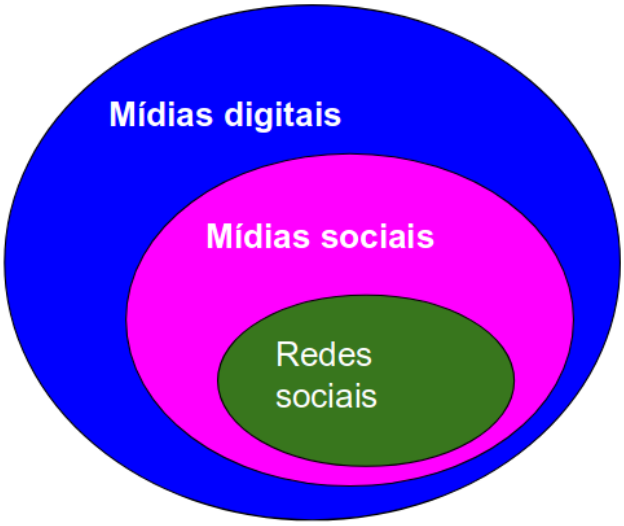


Figura 1 – Relacionamento entre Mídia digital, Mídia social e Rede social

2.1.1 Mídias Digitais

Mídias digitais são definidas como "qualquer meio de comunicação que se utilize de tecnologia digital"(SILVA; VIEIRA; SCHNEIDER, 2010), ou seja, qualquer conteúdo produzido por um computador, seja um texto, imagem, áudio, vídeo ou um página web. Por exemplo o resultado de digitalizar uma foto com um *Scanner* é classificado como uma mídia digital.

A mídia digital possui algumas vantagens em relação às antigas mídias. Para exemplificar essas vantagens, vamos considerar os jornais online, comparados aos jornais impressos. Através do jornal online, é possível que a imprensa compartilhe uma informação para um número muito maior de indivíduos ao mesmo tempo, do que com o jornal impresso. Seu conteúdo pode ser replicado e permanecer na internet por um longo período de tempo. Esse conteúdo pode agregar diversos formatos, como textos, vídeos, áudios, jogos, dentre outros, o que torna o jornal online mais atrativo que o impresso. Outra grande vantagem do jornal online é a hipertextualidade, isto é, links inseridos no texto que permitem ao usuário navegar por outras fontes de conteúdo. Além desse recurso, os jornais online também possuem mecanismos que permitem uma fácil atualização de conteúdo, produção de notícias com baixo custo e interação entre os usuários e as notícias(BARONI et al., 2013). Ao permitir, por exemplo, que um usuário comente em uma notícia, ou responda a uma enquete, tais mecanismos de interação permitem que a notícia seja construída de forma colaborativa.

2.1.2 Mídias Sociais

"As mídias sociais são plataformas na Internet construídas para permitir a criação colaborativa de conteúdo, a interação social e o compartilhamento de informações em diversos formatos"(TELLES, 2010). "As mídias sociais são 'um grupo de aplicações para Internet, construídas com base nos fundamentos ideológicos e tecnológicos da Web 2.0, e que permitem a criação e troca de Conteúdo Gerado pelo Utilizador'"(SILVA; VIEIRA; SCHNEIDER, 2010 apud KAPLAN; HAENLEIN, 2010).

Para criar um marco na evolução das aplicações web, alguns estudiosos se reuniram e criaram o termo web 2.0. Durante um *brainstorming*, eles dividiram diversas aplicações em 1.0 e 2.0, fizeram comparações entre aplicações que possuem propósitos semelhantes, uma de cada grupo, e com isso conseguiram definir quais características que as aplicações 1.0 e 2.0 possuem. O termo web 2.0 foi rapidamente difundido e citado em diversos artigos(O'REILLY, 2005). Segundo Ivan Gorski (GORSKI, 2009) algumas características da Web 2.0 são: transparência e verdade; vários emissores, vários receptores; possibilidade de interação entre os usuários do meio; várias fontes de opinião, conduzindo a mais confiança. Segundo Ana Neves(NEVES, 2007) as ferramentas web 2.0 geralmente possuem algumas funcionalidades que permitem a comunicação dela com outras ferramentas e a agregação de conteúdo por partes dos usuários. Uma das funcionalidades descritas por ela é a criação de blogs,"sites em forma de diário no qual os textos são apresentados por ordem cronológica inversa"(NEVES, 2007). Um exemplo de ferramenta 2.0 que possui essa funcionalidade é o Wordpress[1]. Nela, os usuários podem criar blogs, páginas e aplicativos. Outros exemplos de mídias sociais que podemos citar são: Youtube[2], cujo foco é divulgação de vídeos, Twitter[3]cujo foco é a divulgação de mensagens e SlideShare[4] cujo foco é o compartilhamento de apresentações. Os jornais online apresentados na Seção 2.1.1 possuem muitos dos recursos da Web 2.0. Porém, a definição não se encaixa fielmente a eles, pois o jornal online é o que inicia a emissão de conteúdo, só permitindo que os usuários colaborem agregando novas informações ao conteúdo exposto. Isso se difere na mídia social onde o usuário tem o "poder de difundir uma mensagem de forma descentralizada dos grandes meios de comunicação de massa"(ALTERMANN, 2010), porém é difícil para os órgãos fiscalizadores controlarem o conteúdo divulgado nessas mídias. Com isso, algumas mídias sociais são capazes de divulgar conteúdo falso ou criminoso com facilidade.

2.1.3 Redes Sociais

"Os sites de relacionamento ou redes sociais são ambientes que se focam em reunir pessoas, os chamados membros, que uma vez inscritos, podem expor seu perfil com dados

[1]https://br.wordpress.org/
[2]https://www.youtube.com/
[3]https://twitter.com/
[4]https://pt.slideshare.net/

como fotos pessoas, textos, mensagens e vídeos, além de interagir com outros membros, criando listas de amigos e comunidades."(TELLES, 2010) "Uma rede social é definida como um conjunto de dois elementos: atores(pessoas, instituições ou grupos; os nós da rede) e suas conexões(interações ou laços sociais)"(RECUERO, 2009 apud WASSERMAN; FAUST, 1994)(RECUERO, 2009 apud DEGENNE; FORSE, 1999)

Segundo Raquel Recuero (RECUERO, 2009), o ator em uma rede social na Internet é uma construção identitária no ciberespaço. Um ator pode ser representado por um weblog, fotolog, conta no Twitter[3], perfil do Facebook[5], etc. Os atores podem ser constituídos por mais de um indivíduo, como por exemplo uma página no Facebook[5], administrada por diversos outros usuários do Facebook[5]. Quando um dos administradores cria uma postagem para a página, a postagem referencia a página em si e não a conta especifica que a criou. Sendo assim, a página é um ator capaz de interagir com outros atores e criar conteúdo.

A conexão em uma rede social na internet é definida pela interação entre atores daquela rede. Por exemplo, no Facebook[5], uma conexão pode ser representada por um comentário em uma postagem de outro ator, conversar com outro ator por chat, etc.. A rede social na internet permite que um ator se conecte a um número muito maior de atores que uma rede social offline. Estas interações deixam rastros de informação e geram recursos que podem ser utilizados por outros atores(RECUERO, 2005).

Rede social é uma categoria de mídia social cujo foco é o relacionamento entre os usuários. Se a aplicação tem como foco a divulgação de conteúdo, ela fica caracterizada apenas como uma mídia social. Este foco também influencia a forma com que os usuários interagem entre si e com a mídia social, este e outros fatores que influenciam o comportamento do usuário serão descritos na próxima seção.

2.2 Uma Análise sobre o Comportamento do Usuário em Mídias Sociais

O usuário se comporta de forma diferente em uma mídia social e uma rede social. Um ator em uma rede social online pode ser visto como um conjunto de dados que podem ser manipulados, mobilizados, transportados, e até mesmo vendidos. O ator é um objeto dissociado dos indivíduos que o constroem e está em permanente processo de construção. Por exemplo, uma conta no Facebook[5] permite que o ator tenha uma página pessoal, um perfil por onde o mesmo pode expressar sua individualidade e personalidade. Tal perfil, por sua vez, contém fotos e informações sobre o indivíduo que o criou. O ator cria um perfil com intuito de auxiliar na percepção que os outros atores possuem dele, e na forma com

[5]https://www.facebook.com/

que esses atores irão interpretar suas interações na rede social(SOUSA; BRAGA, 2013).

Essas interações também fazem parte da construção do ator e levam em consideração as suas conexões na rede social. Por exemplo, no Facebook[5], os amigos de um usuário têm acesso aos seus comentários e compartilhamentos, o que leva este usuário a ponderar melhor sobre quais palavras ele vai utilizar para expressar sua opinião, e sobre qual conteúdo ele irá compartilhar. Por outro lado, o grau de anonimato presente nas demais mídias sociais, dada a quantidade menor ou até mesmo a ausência de conexões, permite que o usuário se expresse com mais liberdade.

Com o intuito de analisar o impacto do anonimato em comentários de notícias online, um grupo de pesquisadores da UFMG (REIS et al., 2016) realizou um experimento onde foram analisados dois grupos de comentários, aqueles feitos por usuários anônimos do Jornal Reuters[6] e aqueles feitos por usuários conhecidos do Facebook[5]. Foram utilizadas ferramentas de análise de sentimentos para detectar a polaridade dos textos, onde foi constatado que mais da metade dos comentários anônimos tinham polaridade negativa.

Outro exemplo de que as conexões podem alterar o comportamento do usuário foi observado em um estudo realizado na Finlândia (STRANDBERG; BERG, 2013), que analisou amostras de comentários pertencentes a notícias políticas retiradas de um famoso jornal online da região, o *Vasabladet*[7], com o objetivo de detectar a existência de debates democráticos nestes comentários. Apesar da maioria dos comentários apresentar algum tipo de raciocínio lógico, poucos possuíam validações baseadas em fontes externas: a maioria dos indivíduos se basearam em suas próprias perspectivas e valores. Também notou-se que as respostas direcionadas aos comentários eram em sua maioria superficiais, diferente das redes sociais, onde os debates estão cada vez mais aprofundados e repletos de links para fontes externas.

Segundo o neurocientista Miguel Nicolelis em entrevista ao jornal GGN(MILENA, 2016), a imersão contínua no mundo virtual tem moldado a mente das pessoas, onde pessoas se comportam cada vez mais como máquinas, perdendo peculiaridades analógicas como empatia, solidariedade e respeito à opinião alheia. De acordo com ele, a Internet tem sincronizado a mente humana, levando a uma fragmentação da sociedade em grupos de mesmo interesse. No Facebook[5], vemos comunidades de pessoas com os mesmos ideais, que passaram a cultivar intolerância com pessoas que pensam diferente de si. Isso faz com que as pessoas acreditem que sua forma de pensar é a mais correta e passam grande parte do tempo tentando provar isso as outras pessoas. Este comportamento é fomentado por algoritmos que filtram o conteúdo que o usuário irá acessar, impedindo que um usuário entre em contato com informações contrárias à sua opinião. Esse fenômeno é conhecido como Filtro de Bolha, descrito a seguir.

[6] http://www.reuters.com
[7] https://www.vasabladet.fi/

2.2.1 Filtro de Bolha

"O filtro de bolha é seu próprio, pessoal e único universo de informação com o qual você vive online. E o que está no seu filtro de bolha depende de quem você é, e do que você faz. Mas a questão é que você não define o que entra. E mais importante você não vê o que fica de fora."(PARISER, 2011) Segundo Eli Pariser o filtro de bolha é criado pela junção de todos os algoritmos de filtro de conteúdo encontrados na web, que tentam prever o que o usuário quer acessar. Esses algoritmos funcionam da seguinte forma: eles baseiam suas decisões sobre qual conteúdo deve ser visto pelo usuário no histórico de interações que o mesmo tem com as aplicações Web(FAVA; JÚNIOR, 2014).

Tais algoritmos têm limitado a forma com que os indivíduos percebem a realidade. Pariser argumenta em sua apresentação no TED[8] *talk* o quão ruim é não termos acesso a informações que nos desafiem e ampliem nossa visão do mundo, pois uma visão de mundo limitada pode acarretar em intolerância e perda de racionalidade. Entender que existem visões de mundo diferentes é o primeiro passo para se chegar a um entendimento entre pessoas com ideais distintos.

Exemplificando o uso desses filtros, temos o Google[9] que traz resultados diferentes para usuários diferentes em suas buscas. Por exemplo, se um usuário que viaja muito pesquisar sobre a França, irão aparecer resultados relacionados a preços de passagens de avião e quartos de hotel, enquanto um usuário que se interessa por política irá receber notícias de jornais sobre os acontecimentos políticos da França. O Facebook[5] também utiliza esse tipo de filtro, pois na *timeline* do Facebook[5] só aparecem as postagens de perfis com os quais o ator mais interagiu, seja por conversas ou por realizar comentários, curtidas e compartilhamentos de postagens.

Infelizmente, isso pode fazer com que o usuário só receba informações que vão de acordo com os seus ideais, e só confraternize com pessoas que pensem da mesma forma, fazendo com que o mesmo acredite que sua a visão de mundo é a mais correta, pois ele não tem acesso a informações e pessoas que invalidem a sua visão. Outro risco advindo de tais algoritmos de filtragem é a manipulação desses filtros para que usuário acredite em algo que pode não ser verdade. Em janeiro de 2012, o Facebook[5] realizou um experimento com 700 mil usuários durante uma semana (KRAMERA; GUILLORYB; HANCOCK, 2014), sem que os usuários tivessem conhecimento sobre o mesmo, uma vez que todo usuário do Facebook[5] aprova este tipo de prática ao concordar com os termos de uso do Facebook[5]. Nesse experimento, os perfis selecionados foram divididos em dois grupos: um deles recebia notícias de caráter mais positivo em seu *feed*, enquanto o outro grupo recebia notícias negativas. Como resultado, perceberam que o grupo que recebeu notícias boas tinha a propensão a publicar mais coisas positivas, e quem recebeu notícias sobre acontecimentos

[8] https://www.ted.com/
[9] https://www.google.com

ruins, acabava por publicar conteúdos de teor mais negativo.

Porém, apesar do poder nocivo desses algoritmos, atualmente a maior parte deles é utilizada com o intuito de indicar produtos aos usuários que possuam uma predisposição maior a comprá-los. Esse tipo de algoritmo se faz necessário, dada a infinidade de informações que a internet possui. Um possível caminho de solução pode ser o que Pariser apela às grandes organizações: que o funcionamento deste tipo de algoritmo não seja invisível ao usuário, de forma que ele próprio possa ter algum tipo de controle sobre a saída do filtro.

2.3 Discurso de Ódio

"[...]o discurso do ódio refere-se a palavras que tendem a insultar, intimidar ou assediar pessoas em virtude de sua raça, cor, etnicidade, nacionalidade, sexo ou religião, ou que têm a capacidade de instigar violência, ódio ou discriminação contra tais pessoas"(SILVA et al., 2011 apud BRUGGER, 2007). Ele também pode ser definido como "uma manifestação segregacionista, baseada na dicotomia superior (emissor) e inferior (atingido) e, como manifestação que é, passa a existir quando é dada a conhecer por outrem que não o próprio autor"(SILVA et al., 2011). Segundo Rosane Leal da Silva et al (SILVA et al., 2011), o "discurso de ódio"é caracterizado por dois atos. O primeiro consta em denegrir a dignidade de um determinado grupo de pessoas por conta de características que elas possuem em comum. O segundo ato seria direcionado a outras pessoas que não pertencem ao grupo dos atingidos pelo discurso, a fim de incentivá-los a participarem do mesmo, assim ampliando o alcance e as consequências desse discurso. Essas pessoas ao entrarem em contato com o discurso de ódio, podem criar uma visão distorcida em relação as vítimas. As vítimas desse tipo de discurso acabam envoltos em sentimentos negativos, aflitos e com pouca auto-estima, interiorizando para si a forma com que elas são vistas por essas pessoas. Por exemplo, em uma notícia online, um usuário faz um comentário dizendo que "um personagem homossexual citado na notícia é uma aberração, que deveria deixar de existir". Com grandes chances, qualquer pessoa homossexual que ler o comentário terá sua dignidade violada e se sentira inferior as outras pessoas. O mesmo comentário será difundido para um número grande de pessoas, potencializando o poder nocivo do discurso de ódio.

Em se tratando de mídias sociais, quanto maior o poder difusor da mídia na qual ele foi postado, mais grave o crime se torna. O discurso de ódio na Internet, onde uma informação pode se propagar muito rapidamente, atingindo uma quantidade muito grande de pessoas, tem um potencial muito alto de incitar violência. Por isso, muitos consideram o ato de postar um discurso de ódio uma violência em si.

Hoje, o discurso de ódio é proibido em diversas mídias digitais, o que é comumente especificado nos termos de uso da mídia digital. Para tornar efetiva essa proibição, a

reguladora da mídia pode remover o conteúdo e/ou punir o usuário que criou o discurso de ódio. Porém, o anonimato de algumas mídias digitais aumenta a sensação do usuário de que ele não será punido pelo seus feitos, o que acaba por incentivá-lo a uma prática nociva. Além disso, alguns usuários utilizam mecanismos para não serem detectados, como a criação de perfis falsos. Essas mídias carecem de mecanismos que ajudem a detectar esses crimes. Na próxima seção veremos algumas formas de combate ao discurso de ódio, utilizadas atualmente.

2.3.1 Combate ao Discurso de Ódio

A forma mais comum de detecção do discurso de ódio utilizada em diversas mídias sociais é o sistema de denúncias, onde os usuários avaliam o conteúdo gerado por outros usuários, verificando se esse conteúdo fere os termos de uso da mídia social. Além desse método, os usuários podem usar ferramentas disponíveis pela própria mídia social para expor a um número grande de pessoas o conteúdo contendo discurso de ódio, com objetivo de expor a atitude negativa e conscientizar as pessoas de que aquilo é errado. Geralmente, tal compartilhamento vem acompanhado de uma descrição da crítica a ser feita. Por exemplo, recentemente um juiz, permitiu através de uma liminar que psicólogos ofereçam terapia de "reversão sexual", popularmente chamada de "cura gay", sem que sofram punições do Conselho Federal de Psicologia (CFP) que condena essa prática. Essa notícia teve muita repercussão e em reposta a isso grupos sociais se uniram e começaram a postar tweets com a hashtag #HomofobiaÉDoença, esse tipo de manifestação, também chamada de ẗuitaço, envolve muitos usuários em prol de uma causa comum, que agem de forma coordenada e utilizam uma *hashtag* para demarcar que seus tweets pertencem a manifestação. (SZPACENKOPF; GUERRA; GRANDELLE, 2017).

Outra iniciativa de combate ao discurso de ódio parte das organizações não governamentais, como a Safernet[10] que recebem denúncias de diversos crimes cibernéticos em seu website e ainda oferecem ajuda psicológica para as vítimas. O emissor do discurso de ódio pode utilizar artifício para validar suas afirmações e trazer mais gente ao seu modo de pensar. Um desses artifícios é a criação de conteúdo falso, geralmente composto por sites com noticias falsas que incentivam o público a odiar determinado grupo. Ongs como a Safernet[10] ajudam a remover esse tipo de conteúdo da internet.

Algumas mídias hoje procuram formas de detectar automaticamente o discurso de ódio sem o envolvimento dos seus usuários, com o uso de algoritmos de Aprendizado de Máquina. O Wikipedia[11](WULCZYN; THAIN; DIXON, 2017) utiliza Aprendizado de Máquina para detectar ataques pessoais e tom agressivo nos comentários dos usuários e da página de conversa do artigo. O jornal *The New York Times*[12](COMPANY, 2016)

[10]http://new.safernet.org.br/
[11]https://www.wikipedia.org/
[12]fot:https://www.nytimes.com

utiliza Aprendizado de Máquina para auxiliar a sua equipe de 14 moderadores que revisam manualmente cerca de 11.000 comentários por dia, levando até eles os comentários com maior probabilidade de serem retirados.

Porém, os diversos algoritmos existentes hoje não são perfeitos, e a maioria é limitado a uma língua específica. Para o cientista de dados Brasileiro, um dos grandes desafios atuais é desenvolver aplicações capazes de reconhecer determinados tipos de textos escritos em português, devido a falta de recursos e estudos dessa área na linguá portuguesa. Na próxima seção veremos como o discurso de ódio é abordado pela legislação Brasileira e as dificuldades de julga-lo como crime.

2.3.2 Discurso de ódio X Liberdade de Expressão

Existe um grande desafio em classificar textos como discurso de ódio perante a lei, sem ferir o direito à liberdade de expressão. Infelizmente, pode ser necessário limitar a liberdade de expressão para proteger outros direitos fundamentais, como igualdade, privacidade e honra. Porém, segundo o autor Daniel Sarmento (SARMENTO, 2006), essas limitações não podem ser definidas pelas doutrinas majoritárias, mas sim serem neutras em relação aos pontos de vista existentes naquela sociedade e naquele momento histórico. Pode ser o caso de que, quando utilizada em um debate racional, uma ideia, mesmo que impopular, polêmica e controversa, possa trazer benefícios para a sociedade. Por exemplo, uma pintura sobre um símbolo religioso, rodeado de dinheiro, é uma crítica a comercialização da religião. Mesmo que ofensivo para um grupo, o intuito da mesma é fomentar o debate e a discussão de ideias. Já o discurso de ódio é caracterizado pelo intuito de apenas ofender pelos motivos mais vis, onde o emissor do discurso não possui nenhuma predisposição para ouvir a opinião contrária à dele e refletir sobre o assunto.

Segundo o Pacto Internacional sobre Direitos Civis e Políticos(PIDCP) (ONU, 1996) artigo 19 parágrafo 2, "Toda pessoa terá direito à liberdade de expressão; esse direito incluirá a liberdade de procurar, receber e difundir informações e ideias de qualquer natureza, independentemente de considerações de fronteiras, verbalmente ou por escrito, em forma impressa ou artística, ou qualquer outro meio de sua escolha". Entretanto, o artigo 19 parágrafo 3 do PIDCP determina responsabilidades no exercício da liberdade de expressão, e restrições em alguns casos, mas somente se a restrição for (a) prevista em lei, (b) necessária, (c) para proteção de um dos objetivos listados no artigo, quais sejam: assegurar o respeito do direito e reputação dos outros, a segurança nacional, a ordem, saúde ou moral pública. Assim, conforme o artigo 20 do PIDCP, os critérios para enquadrar um discurso como discurso do ódio, tornando-o passível de punição são:

- Severidade:

A ofensa deve ser "a mais severa e profunda forma de opróbrio".

- Intenção:
  Deve haver a intenção de incitar o ódio.
- Conteúdo ou forma do discurso: Devem ser consideradas a forma, estilo e natureza dos argumentos empregados.
- Extensão do discurso: O discurso deve ser dirigido ao público em geral ou à um número de indivíduos em um espaço público.
- Probabilidade de ocorrência de dano:
  o crime de incitação não necessita que o dano ocorra de fato, entretanto, é necessária a averiguação de algum nível de risco de que algum dano resulte de tal incitação.
- Iminência:
  o tempo entre o discurso e a ação (discriminação, hostilidade ou violência) não pode ser demasiado longo, de forma que não seja razoável imputar ao emissor do discurso a responsabilidade pelo eventual resultado.
- Contexto:
  O contexto em que é proferido o discurso é de suma importância para verificar se as declarações têm potencial de incitar ódio e gerar alguma ação.

.

O judiciário brasileiro possui uma grande dificuldade de julgar o discurso de ódio, uma vez que a maioria dos casos desse tipo são classificados como ofensa a honra, calúnia, difamação e injúria. Julgar um discurso de ódio na internet é um problema mais complexo, dado o grau de anonimato que os emissores possuem, sendo necessário uma investigação para descobrir quem foi o indivíduo por traz da conta que emitiu o discurso. Porém, atualmente, vemos no Brasil um avanço em relação ao julgamento de crimes baseados na internet, com a criação do marco civil da internet que melhor delimita as características deste tipo de crime. "O Marco Civil é a primeira legislação brasileira em matéria de Direito e internet, sendo pioneira no âmbito mundial. Trata-se, essencialmente, de instrumento legislativo que estabelece os direitos e deveres dos provedores e usuários da Internet no Brasil e, ainda, traça os contornos da responsabilidade civil destes últimos."(MARTINS; VILELA; SOARES, 2016 apud GNET, 2016) Dado a ineficácia do judiciário, soluções extra judiciais têm ganhado força, seja pelo Estado fazendo campanhas de conscientização e dando amparo as vítimas, ou por iniciativas da sociedade. Essas vítimas geralmente pertencem a grupos marginalizados pela sociedade (SILVA et al., 2016), na próxima seção veremos como o discurso de ódio afeta um desses grupos, a comunidade LGBT (Lésbicas, Gays, Bissexuais, Travestis, Transexuais e Transgêneros).

2.3.3 Discursos de Ódio Direcionados à Comunidade LGBT

Os usuários de jornais online no Brasil dão muito destaque a notícias cujo tema envolve o público LGBT (Lésbicas, Gays, Bissexuais, Travestis, Transexuais e Transgêneros), como visto em uma pesquisa que analisou o caso da publicação do Banco do Brasil em sua página no Facebook[5] a favor do casamento igualitário(LEITE; BATISTA; SOUZA, 2014). Grande parte dos comentários foram de aprovação. Porém, dos comentários contrários a publicação, foram avaliados usos de termos bíblicos em ofensas, e comentários que associavam o público LGBT a crimes como pedofilia, zoofilia entre outros. Esse exemplo mostra que algo simples como pedir um direito fundamental, algo que não prejudicaria ou afetaria a vida das outras pessoas é duramente atacado por usuários que querem pregar uma certa superioridade em relação a outros grupos.

A pesquisa mencionada no parágrafo anterior destacou dois comentários que geraram um debate construtivo, o que destaca a importância das empresas de publicidade que apresentam um ponto em determinados assuntos, para gerar reflexão. A demonstração de ódio contra o público LGBT na Internet pode ser considerada um reflexo do que acontece no dia a dias dessas pessoas. A quantidade de violência cometida contra LGBTs tem crescido a cada ano que passa, conforme podemos acompanhar em relatórios anuais fornecidos pelo governo (BRASIL, 2016).

2.4 Mineração de Dados

A prática de armazenar dados existe desde o surgimento dos sistemas computacionais, porém com o avanço das tecnologias tanto de hardware quanto de software, tornou-se possível o armazenamento de uma infinidade de dados. Extrair informações novas a partir dos dados armazenados se tornou uma prática vantajosa para diversas organizações. Por exemplo, nas redes sociais, cada interação do usuário com o sistema e com os outros usuários geram dados. Destes dados podemos extrair informações úteis para o direcionamento de *marketing* e assim aumentar a chance de venda de um produto(CAMILO; SILVA, 2009). "Mineração de dados é a exploração e a análise, por meio automático ou semi-automático, de grandes quantidades de dados, a fim de descobrir padrões e regras significativos"(BERRY; LINOFF, 1997).O potencial da Mineração de Dados é tão grande que atualmente algumas redes sociais disponibilizam aplicações capazes de extrair dados de seus sistemas com facilidade, incentivando um número maior de pessoas a realizaram estudos a partir de seus dados.

Com os dados em mãos, segue-se usualmente o passo a passo das etapas do modelo CRISP-DM de mineração de dados descritas a seguir (CAMILO; SILVA, 2009) e visualizadas na na Figura 2

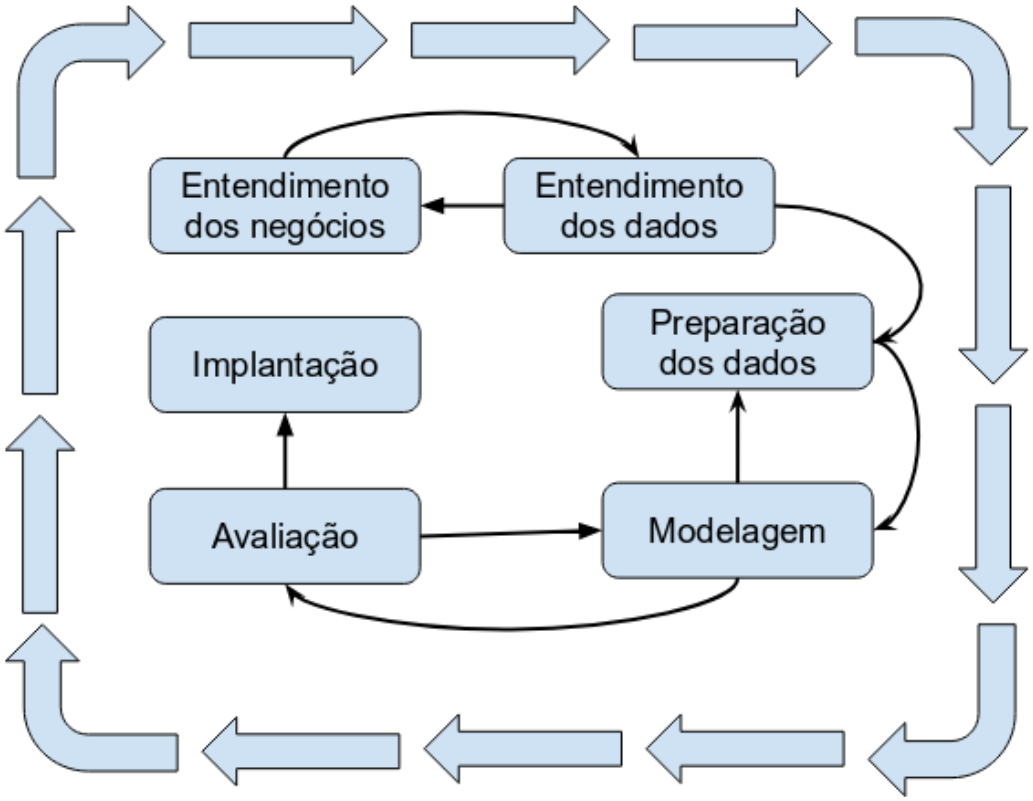


Figura 2 – Figura representando o processo CRISP-DM (LAROSE, 2005)

1. Entendimento dos Negócios:
   Definir qual o objetivo devemos atingir com a Mineração de Dados.

2. Entendimento dos Dados:
   Os dados podem vir de diversas fontes com formatos diferentes. É preciso entender as características e limitações dos dados. A partir disso, avaliar se eles são suficientes para a resolução do problema.

3. Preparação dos Dados:
   Adaptar os dados para um formato específico para o modelo de mineração de dados escolhido, selecionar os melhores dados para a solução do problema e utilizar técnicas para melhorar a qualidade dos dados.

4. Modelagem:
   Selecionar um algoritmo de Aprendizado de Máquina com maiores chances de solução do problema. Este algoritmo ira gerar um modelo treinado a partir dos dados preparados.

5. Avaliação:
   Realizar testes e validações para testar a confiabilidade do modelo gerado pelo algoritmo de Aprendizado de Máquina.

6. Implantação:
   Aplicar o modelo em dados diferentes dos utilizados no treinamento e exibir os resultados.

As etapas 4 a 6 da Figura 2 serão detalhadas na seção a seguir, sobre Aprendizado de Máquina.

2.5 Aprendizado de Máquina

Aprendizado é um processo de aquisição de novas formas de conhecimento. O ser humano utiliza diversos tipos de aprendizado, como o aprendizado indutivo, que utiliza inferência indutiva de fatos providos por um professor ou ambiente externo para adquirir novos conhecimentos(POZO, 2017). "O mais frequente tipo de aprendizado estudado é o aprendizado a partir de exemplos (chamado de aquisição de conhecimento), onde a tarefa é induzir descrições gerais de conceitos de instâncias específicas destes conceitos"(POZO, 2017).

Projetar um computador capaz de aprender como um ser humano é uma tarefa há muito tempo explorada pelos pesquisadores da área de Inteligência Artificial(RUSSELL; NORVIG, 2013). Estes modelaram o processo de aprendizagem em diversos algoritmos que hoje constituem a subárea da Inteligência Artificial chamada de Aprendizado de Máquina(MITCHELL; HILL, 1997). Os programas desenvolvidos nessa área, usando aprendizado indutivo, conseguem determinar regras de decisão e funções, tendo como base os dados fornecidos por especialistas. Além disso, os algoritmos desenvolvidos na área são capazes de detectar e retificar inconsistências, remover redundâncias, fechar lacunas e simplificar regras de decisão elicitadas por um especialista.

"Um sistema de aprendizado tem a função de analisar informações e generalizá-las, para a extração de um novo conhecimento. Para isso, usa-se um programa de computador, para automatizar o aprendizado"(MATOS et al., 2009 apud MONARD; BARANAUSKAS, 2003).

Os algoritmos tradicionais possuem sequências de instruções bem definidas para solucionar um ou mais problemas com características semelhantes. Assim, esses algoritmos têm um comportamento previsível, geralmente é bem definido que tipo de dados ele espera de entrada e qual tipo de saída que será retornada. No caso de algoritmos determinísticos, para uma mesma entrada sempre obteremos a mesma saída. Por exemplo, um algoritmo responsável por realizar impressões recebe de entrada o tipo de papel, o documento a ser impresso, a quantidade de impressões e a cor da impressão. O resultado obtido é a conclusão da impressão, ou em caso da falta de tinta ou folhas de papel a solicitação das mesmas.

Esse tipo de algoritmo não guarda um histórico e não utiliza informações de execuções anteriores para melhorar o desempenho das impressões. Já os algoritmos de Aprendizado de Máquina, segundo Adam Geitgey (GEITGEY, 2014), são genéricos e podem dizer-lhe algo interessante sobre um conjunto de dados, sem que você tenha que escrever qualquer código personalizado para o problema. Em vez disso, alimenta-se o algoritmo com dados, que por sua vez são utilizados pelo algoritmo para descobrir um padrão de resolução do problema.

Os algoritmos de Aprendizado de Máquina nem sempre retornam um resultado satisfatório sempre que executados. Porém, a sua eficiência para resolução do problema pode aumentar conforme o mesmo adquire experiência. A comparação entre o algoritmo tradicional e o algoritmo de aprendizado de máquina é melhor explicitada na Figura 3.

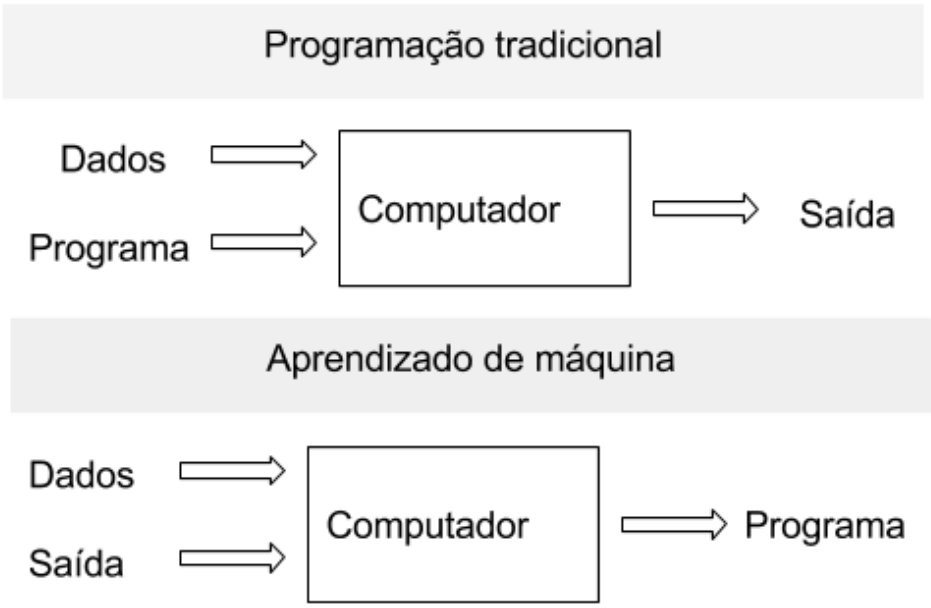


Figura 3 – Comparação entre a programação tradicional e o aprendizado de máquina (BROWNLEE, 2015)

Por exemplo, um algoritmo de recomendação de produtos possui uma infinidade de fatores que influenciam o gosto do usuário. Assim, a tarefa de desenvolver um programa específico para esse problema é difícil, visto que com o passar do tempo o usuário muda e esses fatores também, fazendo com que o programa específico para aquele usuário se torne obsoleto. A melhor escolha para este caso é um algoritmo de Aprendizado de Máquina que se adaptaria a essas mudanças através dos dados de entrada, que, por sua vez, implicitamente possuem informações sobre as mudanças no gosto do usuário.

Com objetivo de exemplificar algumas aplicações que utilizam Aprendizado de Máquina, tomemos como exemplo o Facebook[5], uma rede social que é utilizada por grande

parte da população brasileira. O Facebook[5] revelou que adquire em média 500 terabytes de dados novos por dia e os armazena em um disco Hadoop[13] de 100 petabytes de dados (cada petabyte equivale a 1.048.576 gigabytes)(TAM, 2012). Para analisar essa quantidade imensa de dados gerados por uma mídia social, o Facebook[5] se utiliza de técnicas de Mineração de Dados e Aprendizado de Máquina. O Facebook[5] tem mais de 1,5 bilhões de agentes de Inteligência Artificial (HIGGINBOTHAM, 2016), um para cada usuário. O agente monitora todas as atividades do usuário e armazena dados relevantes que serão utilizados para diversos objetivos, como detectar o rosto desse usuário em fotos, descobrir que eventos esse usuário gostaria de ir, sugerir quais pessoas ele poderia querer adicionar, exibir anúncios de produtos que ele gostaria de comprar, e selecionar que conteúdos ele gostaria de ver 2.2.1.

2.5.1 Funcionamento dos Algoritmos de Aprendizado de Máquina

Após a fase de preparação dos dados da Mineração de Dados, um algoritmo de Aprendizado de Máquina recebe os dados convertidos para o formato específico com o qual ele é capaz de trabalhar. Um dos formatos mais utilizados é o formato vetorial, onde o exemplo obtido a partir da base de dados é modelado em um vetor. O valor associado a um atributo pode ser categorizado em dois tipos, qualitativo e quantitativo. Os dados quantitativos são compostos por valores numéricos. Estes ainda podem ser divididos em dois grupos: dados discretos e contínuos. Os dados qualitativos são compostos por valores nominais e ordinais (categóricos) (CAMILO; SILVA, 2009). Na Tabela 1 exemplificada essa modelagem.

Tabela 1 – Exemplos de representação vetorial

Características:	**volume(ml)**	**material**	**cor**	**preço(R$)**	**classe**
Copo A:	200	vidro	vermelho	2,00	*barato*
Copo B:	500	vidro	azul	5,00	*barato*
Copo C:	200	plastico	verde	5,00	*caro*

Os tipos mais comuns de algoritmos de Aprendizado de Máquina são o supervisionado, não supervisionado e aprendizado por reforço. Nas próximas seções o aprendizado de máquina supervisionado e não supervisionado que foram utilizados nos experimentos desse trabalho

2.5.1.1 Aprendizado Supervisionado

"Os algoritmos de aprendizado supervisionados fazem previsões com base em um conjunto de exemplos."(ERICSON; OLPROD; OPENLOCALIZATIONSERVICE, 2017).

[13] http://hadoop.apache.org/

Uma das tarefas realizadas pelos algoritmos de Aprendizado de Máquina supervisionados é a de classificação, onde o vetor que representa o elemento possui um atributo discreto que simboliza a classe a qual ele pertence. O algoritmo de Aprendizado deve ser capaz de analisar o exemplo que ele recebeu e, de alguma forma, associar as suas características com a classe a qual ele pertence. Pode-ser ver essa tarefa como a geração de um modelo preditivo (classificador), representado por uma função $F(X) = y$, que, dada uma entrada de dados X, retorna a classe y(ABU-MOSTAFA; MAGDON-ISMAIL; LIN, 2012). A representação da função a ser aprendida varia de algoritmo para algoritmo. Na Figura 4 podemos ver o processo de transformação dos dados rotulados usado no treinamento no modelo preditivo.

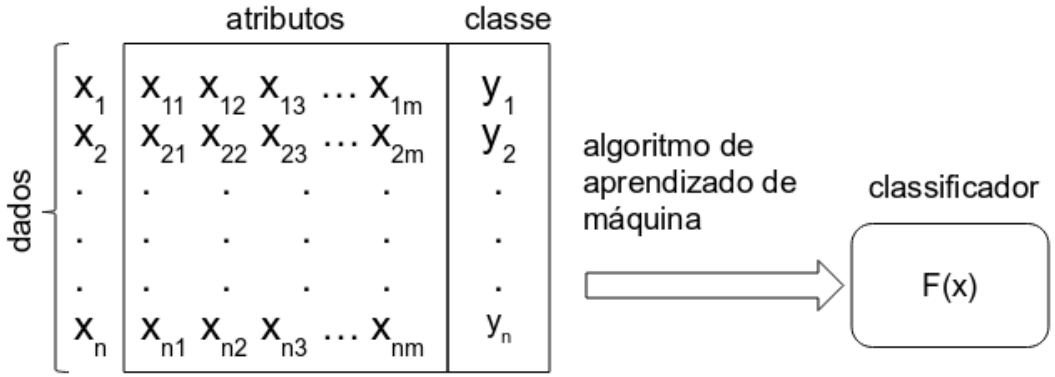


Figura 4 – Gerando um classificador com aprendizado de máquina(LORENA; CARVALHO, 2007)

2.5.1.2 Aprendizado não Supervisionado

"No aprendizado não supervisionado, os pontos de dados não têm rótulos associados a eles. Em vez disso, a meta de um algoritmo de aprendizado sem supervisão é organizar os dados de alguma forma ou descrever sua estrutura"(ERICSON; OLPROD; OPENLOCALIZATIONSERVICE, 2017). Uma das tarefas mais comuns realizadas por algoritmos de aprendizado não supervisionados são tarefas de agrupamento. Um algoritmo de agrupamento busca semelhanças entre as características dos elementos da coleção de dados e atribui grupos a eles com base em um critério pré-definido. Esse critério geralmente trata-se de uma função de dissimilaridade, que recebe dois objetos e analisa as características dos mesmos, retornando a distância entre eles. Como no exemplo anterior, visto na tabela 1, caso se queira dividir os copos em dois grupos, podemos utilizar um método que compare os vetores de forma simples, que a cada atributo igual some 1 ao valor de saída. Dessa forma, temos $F(CopoA, CopoB) = 2$, $F(CopoB, CopoC) = 1$, $F(CopoA, CopoC) = 1$. O Resultado seria o Grupo 1 = {Copo A, Copo B} e Grupo 2 = {Copo C}

2.5.2 Classificação de Textos

> "O objetivo dos algoritmos de aprendizado de máquina é aprender, generalizar, ou ainda extrair padrões ou características das classes das coleções com base nos documentos textuais e rótulos (identificadores de classe) dos documentos informados por um usuário ou especialista de domínio"(ROSSI, 2015).

Técnicas de Aprendizado de Máquina podem ser utilizadas para classificação automática de textos, como, por exemplo, a tarefa de classificar e-mails como spam ou não spam, e para isso precisamos de exemplos dessas duas classes de e-mail. A quantidade de exemplos influencia a capacidade de classificar corretamente um novo e-mail, e é preferível uma quantidade semelhante de exemplos de ambas as classes.

O mapeamento de um texto para um vetor é diferente de um mapeamento de um objeto para o vetor, como visto na Tabela 1, onde cada característica do objeto se torna um atributo. Ao mapear um texto para um vetor utilizamos os termos nele presente como atributos. Uma representação vetorial computada a partir de um texto muito utilizada atualmente é a *Bag-of-words* (BROWNLEE, 2017). Essa técnica transforma a coleção de N documentos com M termos em uma matriz documento-termo, exemplificada na tabela 2, onde o valor de uma célula w_{d_i,t_j} na matriz representa um valor ou um peso de um termo t_j em um documento d_i. Para cada documento d_i, pode ser definido um atributo adicional c_di, para representar a classe do documento. Por exemplo, em um problema de análise de sentimentos (BENEVENUTO; RIBEIRO; ARAÚJO, 2015), um documento pode pertencer à classe de polaridade positiva, indicando sentimento positivo ou à classe de polaridade negativa, indicando sentimento negativo.(ROSSI, 2015 apud TAN; STEINBACH; KUMAR, 2015)

Tabela 2 – Tabela documento-termo

	t_1	t_2	t_3	...	t_{M-2}	t_{M-1}	t_M	$classe$
d_1	$w_{d_1t_1}$	$w_{d_1t_2}$	$w_{d_1t_3}$	...	$w_{d_1t_{M-2}}$	$w_{d_1t_{M-1}}$	$w_{d_1t_M}$	c_{d_1}
d_2	$w_{d_2t_1}$	$w_{d_2t_2}$	$w_{d_2t_3}$	...	$w_{d_2t_{M-2}}$	$w_{d_2t_{M-1}}$	$w_{d_2t_M}$	c_{d_2}
d_3	$w_{d_3t_1}$	$w_{d_3t_2}$	$w_{d_3t_3}$	...	$w_{d_3t_{M-2}}$	$w_{d_3t_{M-1}}$	$w_{d_3t_M}$	c_{d_3}
...	...	...	...	...	...	...	...	...
d_{N-2}	$w_{d_{N-2}t_1}$	$w_{d_{N-2}t_2}$	$w_{d_{N-2}t_3}$	...	$w_{d_{N-2}t_{M-2}}$	$w_{d_{N-2}t_{M-1}}$	$w_{d_{N-2}t_M}$	$c_{d_{N-2}}$
d_{N-1}	$w_{d_{N-1}t_1}$	$w_{d_{N-1}t_2}$	$w_{d_{N-1}t_3}$	...	$w_{d_{N-1}t_{M-2}}$	$w_{d_{N-1}t_{M-1}}$	$w_{d_{N-1}t_M}$	$c_{d_{N-1}}$
d_N	$w_{d_Nt_1}$	$w_{d_Nt_2}$	$w_{d_Nt_3}$	...	$w_{d_Nt_{M-2}}$	$w_{d_Nt_{M-1}}$	$w_{d_Nt_M}$	c_{d_N}

Um dos problemas desta representação é a alta dimensionalidade oriunda do conjunto de termos, dado que uma coleção de documentos pode possuir milhares de termos. Um outro problema é a alta esparsidade, dado que alguns termos aparecem em apenas poucos documentos, fazendo com que a matriz fique com muitos valores iguais a zero,

que não agregam informação, mas somente prejudicam o desempenho. Para diminuir o problema da alta dimensionalidade e melhorar o resultado do programa, podemos utilizar técnicas de pré processamento de textos(VIJAYARANI; ILAMATHI; NITHYA, 2015)(BIRD; KLEIN; LOPER, 2009) durante a fase de preparação dos dados vista na Figura 2 que representa o processo de CRISP-DM. A seguir serão descristas algumas destas técnicas.

- *Stemmer*
  Reduz a palavra a sua forma de raiz. Por exemplo, as palavras "boca"e "bocas"seriam reduzidas à palavra "boc".

- Remoção de *stop words*.
  Remove palavras que podem ser consideradas irrelevantes para a classificação. Exemplos: as, e, os, de, para.

- Padronizar o texto. Remover acentos, caracteres especiais, deixar o texto com letras minúsculas somente, e remover qualquer anormalidade do texto, a fim de duas palavras antes diferentes serem contabilizadas como um mesmo atributo. Por exemplo mãe! e mae? = mae

A Tabela 3 apresenta um exemplo com uma coleção de quatro documentos, usados na explicação das próximas técnicas. A explicação de cada técnica utilizará um ou mais destes documentos.

Tabela 3 – Exemplos de documentos

Documento A:	Três pratos de trigo para três tigres tristes
Documento B:	O doce respondeu pro doce, Qual o doce mais doce, Do que o doce de batata doce, É o doce de batata doce.
Documento C:	brinquedo de cachorro
Documento D:	cachorro de brinquedo
Documento E:	quero mais doce

Uma das técnicas usuais utilizada para o preenchimento da tabela documento-termo 2 é computar a frequência dos termos, ou seja w_{d_i,t_j} da tabela será preenchido com a frequência absoluta do termo t_j no documento d_i. A Tabela 4 exemplifica o uso desta técnica.

Tabela 4 – Técnica de Bag of Words a partir da frequência dos termos, com os documentos A e B de entrada

	batata	**doce**	**pratos**	**pro**	**respondeu**	**tigres**	**tres**	**trigo**	**tristes**
A:	0	0	1	0	0	1	2	1	1
B:	2	8	0	1	1	0	0	0	0

Um problema que surge ao usar os termos do documento isoladamente é a perda da informação do relacionamento entre as palavras. Por exemplo, para um modelo treinado usando *Bag-of-Words* e a frequência dos termos como apresentado anteriormente, as frases "brinquedo de cachorro"e "cachorro de brinquedo"teriam o mesmo significado, visto que a contagem das frequências dessas palavras seriam iguais. Para evitar esse tipo de problema, pode ser utilizada outra representação chamada *n-gram*, que agrupa sequências de tamanho n de palavras no texto, fazendo com que tais sequências passem a ser os atributos, como visto na tabela 5.

Tabela 5 – Resultado do n-gram nas frases:brinquedo de cachorro e cachorro de brinquedo

	(brinquedo cachorro)	**(cachorro brinquedo)**
C:	1	0
D:	0	1

Outra técnica para o preenchimento da tabela documento-termo 2 de grande utilidade é o $TF-IDF$ *Term Frequency–Inverse Document Frequency*. Ele efetua a equação 2.1 para cada célula da matriz documento-termo, onde N é o número de documentos na coleção. O valor w_{d_i,t_j} da matriz aumenta proporcionalmente em relação a tf_{d_i,t_j} (numero de ocorrências de t_j em d_i). No entanto, esse valor é equilibrado pelo df_{t_j} (numero de documentos que contém t_j). Isso auxilia a distinguir o fato da ocorrência de algumas palavras serem mais comuns que outras.

$$w_{d_i,t_j} = tf_{d_i,t_j} \chi \log\left(\frac{N}{df_{t_j}}\right) \tag{2.1}$$

As Tabelas 6 e 7 exemplificam o uso da técnica do TF-IDF. Nota-se que na tabela 7 foi utilizado um documento a mais que também possui o termo 'doce' fazendo com que o valor de $w_{d_A,t_{doce}}$ diminua, pois o termo 'doce' se tornou mais comum entre os documentos.

Tabela 6 – Bag of words usando TF-IDF, com os documentos de A e B

	batata	**doce**	**pratos**	**pro**	**respondeu**	**tigres**	**tres**	**trigo**	**tristes**
A:	0.00	0.00	0.35	0.00	0.00	0.35	0.71	0.35	0.35
B:	0.24	0.96	0.00	0.12	0.12	0.00	0.00	0.00	0.00

Tabela 7 – Bag of words usando TF-IDF, com os documentos de A, B e E

	batata	**doce**	**pratos**	**pro**	**quero**	**respondeu**	**tigres**	**tres**	**trigo**	**tristes**
A:	0.00	0.00	0.35	0.00	0.00	0.00	0.35	0.71	0.35	0.35
B:	0.30	0.93	0.00	0.15	0.00	0.15	0.00	0.00	0.00	0.00
E:	0.00	0.60	0.00	0.00	0.79	0.00	0.00	0.00	0.00	0.00

Em problemas de alta dimensionalidade, para obter um melhor desempenho do algoritmo de Aprendizado de Máquina, podemos utilizar métodos de decomposição de atributos, de forma a diminuir a quantidade de atributos a serem processados, ao criar novos atributos a partir da combinação dos atributos originais. Por exemplo, o método LSA (*Latent Semantic Analysis*)(BHAGWANT, 2011) analisa os documentos que recebe de entrada, procurando os significados ou conceitos implícitos que eles possuem. Esse método é utilizado para revolver problemas relacionados ao significado das palavras, como o problema das palavras polissêmicas. Uma palavra polissêmica é uma palavra que reúne vários significados. Por exemplo, nas frases "Nós chegamos a um acordo"e "Eu acordo de manhã", na primeira a palavra 'acordo' refere-se a um entendimento, e na segunda frase, 'acordo' refere-se a conjugação do verbo acordar. Algumas técnicas como a Tabela 2, preenchida a partir da frequência dos termos são incapazes de reconhecer essa diferença de significados.

Com objetivo de simplificar a resolução do problema, o LSA utiliza a tabela termo-documento em sua análise, assume que as palavras possuem um único significado e define 'conceito' como padrões de palavras que costumam aparecer em documentos. Por fim, o método do LSA gera agrupamentos rotulados por palavras, esses agrupamentos representam os conceitos e serão utilizados como os novos atributos da representação vetorial de um texto.

Outra opção para diminuir a quantidade de atributos é a remoção de alguns deles, usando um processo conhecido como seleção de atributos (TANG; ALELYANI; LIU, 2016). Existem diversos métodos desenvolvidos com o objetivo de remover os atributos que menos influenciam na classificação. Por exemplo, o método $selectKBest$ seleciona os k atributos que possuem uma maior pontuação de acordo com um método avaliador escolhido, como o "Qui Quadrado", simbolizado por X^2, e um teste de hipóteses, que se destina a encontrar um valor da dispersão para duas variáveis nominais, avaliando a associação existente entre variáveis qualitativas"(CONTI, 2009). O objetivo desse método é calcular as divergências entre as frequências observadas e esperadas para um certo evento. Com o resultado desse método, o SelectKBest consegue definir quais termos são os mais prováveis de serem independentes da classe e, portanto, irrelevantes para a classificação.

Outro método de seleção de atributos é o método $selectFromModel$, que utiliza um classificador como base, para definir pesos para os atributos, e remover os atributos que tenham um peso menor do que o valor pré-determinado.

Decidido o formato em que o texto será transformado para servir de entrada ao algoritmo de Aprendizado de Máquina, devemos escolher qual algoritmo utilizar. Nas próximas seções descreveremos dois algoritmos que vêm apresentando bons resultados em classificação de texto (KHAN et al., 2010), são o Naive Bayes (RASCHKA, 2014) e

Support Vector Machines (JOACHIMS, 1998).

2.5.2.1 Naive Bayes

O Naive Bayes é uma técnica estatística que utiliza probabilidade condicional, baseada no teorema de Thomas Bayes (ELLINOR et al., 2017). O teorema de Bayes é uma fórmula que descreve como atualizar as probabilidades de hipóteses quando há evidências. Ele segue simplesmente dos axiomas da probabilidade condicional, mas possui uma ampla gama de uso.

Dadas uma hipótese H e evidências E, o teorema de Bayes afirma que a relação entre a probabilidade da hipótese antes de obter a evidência e a probabilidade da hipótese depois de obter a evidência é dada pela equação 2.2.

$$[H]P(H|E) = \frac{P(E|H)P(H)}{P(E)} \tag{2.2}$$

Esse algoritmo é simples, mas possui uma boa capacidade de predição em diversos problemas. O seu princípio é de que não existe relação de dependência entre os atributos. Han e Kamber Matos et al. (2009 apud HAN; KAMBER, 2006) apresentam o processo de classificação do Naïve Bayes dividido em cinco passos:

1. Seja D um conjunto de treinamento de sentenças distribuídas nas respectivas classes. Cada sentença é representada por um vetor de termos n-dimensional, $X = (\ X_1,X_2,\ ...\ ,X_N\ )$ e cada termo está relacionado à sentença, respectivamente, por $A_1,A_2,\ ...,\ A_N$.

2. Suponha que há m classes $C_1,C_2,...,C_N$ dada uma sentença X, o classificador irá predizer que X pertence a classe que tiver a maior probabilidade posterior, condicionada a X. Isto é, a sentença X pertence a classe C_i se e somente se

$$[H]P(C_i|X) > P(C_j|X) \text{ para } i \leq j \leq m, j \neq i \tag{2.3}$$

   Portanto, pelo teorema de Bayes, para $P(C_i|X)$ a classe C_i é maximizada pela Equação 2.4:

$$[H]P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)} \tag{2.4}$$

3. Como $P(X)$ é constante para todas as classes, somente $P(X|C_i)P(C_i)$ necessita ser maximizada. Se a probabilidade prévia da classe não é conhecida, então é comumente assumido que as classes têm probabilidades iguais, isto é, $P(C_1)=P(C_2)=...P(C_N)$. Portanto, somente é necessário maximizar $P(X|C_i)$. Caso contrário, é maximizado $P(X|C_i)P(C_i)$. Note que a probabilidade prévia da classe pode ser estimada por

$P(C_1) = |C_{i,D}| \div |D|$ onde $|C_{i,d}|$é o número de sentenças de treinamento da classe C_i em D.

4. Considere um conjunto de dados com muitos termos. Seria computacionalmente caro calcular $P(X|C_i)$ para cada termo. A hipótese simples de independência condicional da classe é usada a fim de reduzir o custo computacional para avaliar $P(X|C_i)$. Presume-se que os valores dos termos são condicionalmente independentes uns dos outros. Assim, pela Equação 2.5 tem-se a probabilidade de X condicionada a classe C_i.

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k, C_i) = P(x_1, C_i)\chi P(x_2, C_i)\chi \ ... \ \chi P(x_n, C_i) \tag{2.5}$$

   Lembrando que x_k refere-se ao valor do termo A_k da sentença X. Para cada termo, computa-se $P(X|C_i)$ da seguinte forma: x_k dividido por $|C_{i,D}|$, e $|D|$ é o tamanho do conjunto de sentenças.

5. $P(X|C_i)P(C_i)$ é avaliada para cada classe C_i a fim de predizer a qual classe a sentença X pertence. O classificador prediz a classe C_i para a sentença X para a classe que tiver a probabilidade mais alta, Equação 2.3

2.5.2.2 Vector Machines

"As SVMs lineares com margens rígidas definem fronteiras lineares a partir de dados linearmente separáveis. Seja T um conjunto de treinamento com n dados $x_i \in X$ e seus respectivos rótulos $y_i \in Y$, em que X constitui o espaço dos dados e $Y = \{+1,-1\}$. T é linearmente separável se é possível separar os dados das classes +1 e -1 por um hiperplano"(LORENA; CARVALHO, 2007 apud SCHöLKOPF; SMOLA, 2002).

O algoritmo do SVM utiliza-se da equação do hiperplano, que divide o espaço de valores de X em duas regiões, onde cada uma dessas regiões representa uma classe. Para obter a melhor divisão possível, procura-se escolher uma região onde o hiperplano ficaria o mais distante possível dos pontos de cada classe. Esse processo é representado na figura 5.

Para obter tal distância, são utilizados vetores de suporte paralelos ao vetor do hiperplano, que passam a representar a margem do classificador. O objetivo do algoritmo é tentar maximizar essa margem, dado que quanto mais próximo um ponto estiver do hiperplano, maiores as chances de erro na classificação. Essa técnica tem obtido bons resultados na classificação, porém uma de suas desvantagens é o tempo alto que ela leva para treinar o classificador.

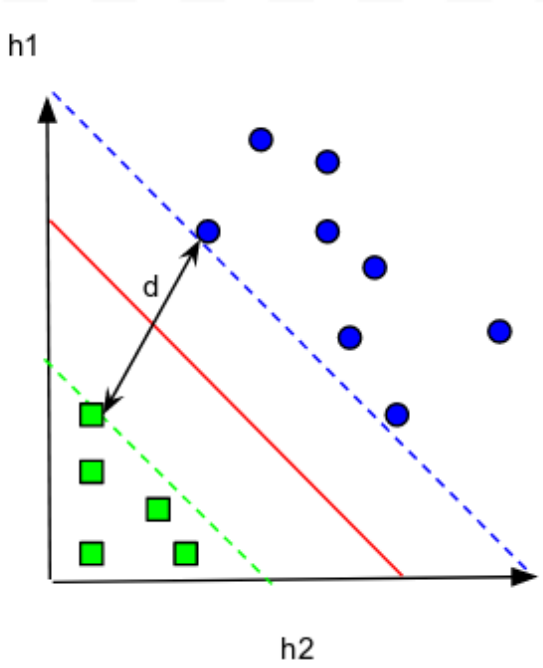


Figura 5 – SVM Linear (Support Vector Machines) baseada na figura 6 de (LORENA; CARVALHO, 2007)

2.5.2.3 Ensemble

Também é possível combinar as previsões de vários classificadores construídos com um determinado algoritmo de aprendizado, a fim de obter a melhor generalização e robustez do que um único classificador. Esse método é chamado de *ensemble*(DIETTERICH, 2000).

A seguir, descrevemos as técnicas de Ensemble mais conhecidas, segundo Jason Brownlee (BROWNLEE, 2016) :

- *Bagging*: cria vários modelos (tipicamente do mesmo tipo) treinados com diferentes subamostras do conjunto de dados de treinamento. Normalmente, as predições dos diferentes modelos erados são combinadas por votação.
- *Boosting*: cria vários modelos (tipicamente do mesmo tipo), cada um dos quais aprende a corrigir os erros de previsão de um modelo anterior na cadeia.
- *Stacking*: cria vários classificadores distintos e utiliza as predições dos classificadores para construir uma nova base de dados. Um novo classificador é treinado a partir de tal base, para finalmente definir o valor final de classe de um exemplo.

2.6 Avaliação

Uma forma simples de avaliar a qualidade de um classificador é feita através de um teste, onde o especialista monta um conjunto de dados previamente anotados, e utiliza o modelo de aprendizado de máquina para tentar prever as classes dos dados pertencentes a esse conjunto. Com as classes reais e as classes preditas pelo modelo podemos extrair as métricas, descritas a seguir.

2.6.1 Métricas

Como descrito anteriormente, tanto a classificação quanto as outras tarefas realizadas com Aprendizado de Máquina, devem ser avaliadas quanto à sua qualidade. Para avaliar a qualidade do modelo gerado, existem algumas métricas importantes, descritas abaixo (MANNING; RAGHAVAN; SCHÜTZE, 2008)

- TP (*(*True positive)): número de exemplos positivos que foram classificados como positivos
- FP (*(*False positive): número de exemplos negativos que foram classificados como positivos
- TN(*(*True negative)): número de exemplos negativos que foram classificados como negativos
- FN (*(*False negative)): número de exemplos positivos que foram classificados como negativos
- Precisão(*(*Precision)): fração de exemplos corretamente classificados como positivo em relação a todos os exemplos classificados como positivos.

$$\text{Precisão}(\textit{(Precision)}) = \frac{TP}{TP + FP} \tag{2.6}$$

- Revocação(Recall):fração de exemplos corretamente classificados como positivo em relação a todos os exemplos que possuem a classe positiva.

$$\text{Revocação} = \frac{TP}{TP + FN} \tag{2.7}$$

- Medida-F(F- measure): é a média harmônica entre a precisão e a revocação.

$$\text{Medida-F} = \frac{2x(\text{Precisão} + \text{Revocação})}{\text{Precisão} + \text{Revocação}} \tag{2.8}$$

- Acurácia(Accuracy): porcentagem de amostras positivas e negativas classificadas corretamente sobre a soma de amostras positivas e negativas.

$$\text{Acurácia} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.9}$$

Uma forma de exibir os diferentes valores produzidos é a matriz de confusão, exibida na Tabela 8, onde PP é o número de exemplos preditos como positivo, PN números de exemplos preditos como negativos, POS é número de exemplos positivos , NEG é o número de exemplos negativos e N é o número de exemplos.

Tabela 8 – Matriz de Contingência para modelos de classificação usando frequência absoluta baseada na tabela 1 do artigo (PRATI; BATISTA; MONARD, 2008)

	Preditos		
Reais	TP	FN	POS
	FP	TN	NEG
	PP	PN	N

Uma tabela equivalente a Tabela 8 pode ser criada usando probabilidades condicionais, conforme exemplificado na Tabela8. Lá, X é uma variável aleatória da classe real positiva e Y é uma variável de classe predita positiva. $\overline{X}$ e $\overline{Y}$ o oposto de X e Y. $P(X, Y)$ é dada pela equação 2.10.

$$P(X, Y) = P(X|Y)P(Y) = P(Y|X)P(X) \tag{2.10}$$

Tabela 9 – Matriz de Confusão para modelos de classificação usando probabilidade conjunta baseada na tabela 1 do artigo (PRATI; BATISTA; MONARD, 2008)

	Y	$\overline{Y}$	
X	$P(X, Y)$	$P(X, \overline{Y})$	$P(X)$
$\overline{X}$	$P(\overline{X}, Y)$	$P(\overline{X}, \overline{Y})$	$P(\overline{X})$
	$P(Y)$	$P(\overline{Y})$	1

$P(X|Y)$ é a probabilidade condicional de X dado Y, ou seja, probabilidade de que a classe real seja positiva dado que a classe predita pelo modelo foi positiva. $P(Y|X)$ é probabilidade da predição ser positiva dado observações de exemplos de classe real positiva. A relação entre a probabilidade de classificar um exemplo como positivo ou não é dada

pela equação 2.11

$$P(Y|X) = 1 - P(\overline{Y}|X) \tag{2.11}$$

Em um conjunto de exemplos, as probabilidade condicionais podem ser estimadas como proporções, como visto nas equações 2.12 e 2.13

$$P(X|Y) = \frac{P(X,Y)}{P(Y)} = \frac{TP}{PP} \tag{2.12}$$

$$P(Y|X) = \frac{P(X,Y)}{P(X)} = \frac{TP}{POS} \tag{2.13}$$

A classe a qual uma probabilidade representa é dado por uma parâmetro do modelo preditivo chamado limiar. Em um problema de classificação binária a limiar L é dada uma porcentagem, onde todos os exemplos que tiverem com $P(Y|X) > L$ pertencem a classe positiva e $P(Y|X) < L$ pertencem a classe negativa. O uso de gráficos ajuda a visualizar as características do problema de avaliação de um modelo preditivo. Para avaliarmos o modelo podemos gerar um gráfico ROC, "Receiver Operating Characteristic é um método gráfico para avaliação, organização e seleção de sistemas de diagnóstico e/ou predição."(PRATI; BATISTA; MONARD, 2008). Esse gráfico exibe a relação entre a taxa de verdadeiros positivos $TPR = P(Y|X)$ e a taxa de falsos positivos $FPR = P(Y|\overline{X})$, onde TPR representa o eixo y e FPR representa o eixo x. Um modelo preditivo é um ponto no espaço do gráfico ROC(HAND, 2009), que utiliza a matriz de confusão para definição de suas coordenadas (FPR,TPR). Um exemplo está representado na Figura 6

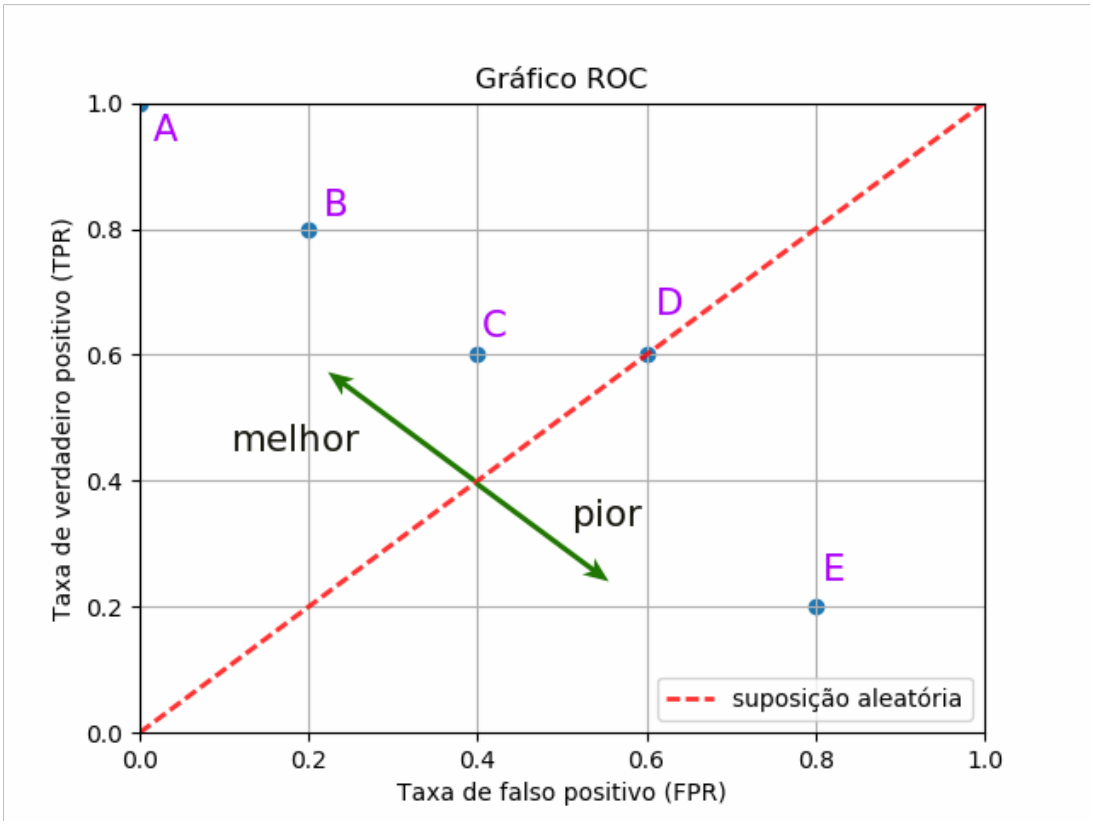


Figura 6 – Exemplo de gráfico de ROC

Alguns elementos do gráfico merecem destaque:

- O ponto (0.0,0.0) representa os modelos que nunca classificam exemplos como positivos.
- O ponto (1.0,1.0) representa modelos que sempre classificam os exemplos como positivos.
- O ponto (0.0,1.0) representa os modelos que classificam corretamente todos os exemplos.
- O ponto (1.0,0.0) representa os modelos que classificam incorretamente todos os exemplos.
- A linha de (0.0,0.0) a (1.0,1.0) , onde os pontos pertencentes ao triângulo superior esquerdo da representam os modelos que possuem desempenho melhor que o aleatório e os do triangulo inferior direito são os que apresentam desempenho pior do que o aleatório.

Duas principais vantagens da análise utilizando o gráfico de ROC é que a análise pode ser feita independente de algumas condições, tais como a distribuição das classes

entre os exemplos, o limiar de classificação e os custos relacionados à classificações errôneas. A segunda vantagem é a utilização desse método para calibração e ajuste de modelos de predição. Existem outras formas de avaliar um modelo utilizando o gráfico ROC. É possível, por exemplo, utilizar um mesmo modelo para criar uma curva no gráfico ROC, vista na Figura 7, ao alterar o valor do parâmetro de limiar, desde seu valor mais restritivo até o seu valor mais liberal, onde cada configuração de modelo e limiar representam um ponto diferente no gráfico.

A maneira mais eficiente de gerar esta curva é ordenando todos os casos de teste de acordo com o valor contínuo predito pelo modelo. Para cada caso desse conjunto dá-se um passo de tamanho $\frac{1}{POS}$, na direção y se o exemplo for positivo e $\frac{1}{NEG}$, na direção x se o exemplo for negativo. Quanto mais afastada a curva estiver da diagonal principal no sentido do ponto (0.0,1.0), melhor será o desempenho do algoritmo naquele domínio. Outra métrica importante retirada dessa curva é a AUC(*Area Under Curve*)(FLACH; HERNÁNDEZ-ORALLO; FERRI, 2011) que é numericamente igual a probabilidade que, dado dois exemplos de classes distintas, o exemplo de classe positiva seja ordenado primeiramente que um exemplo negativo.

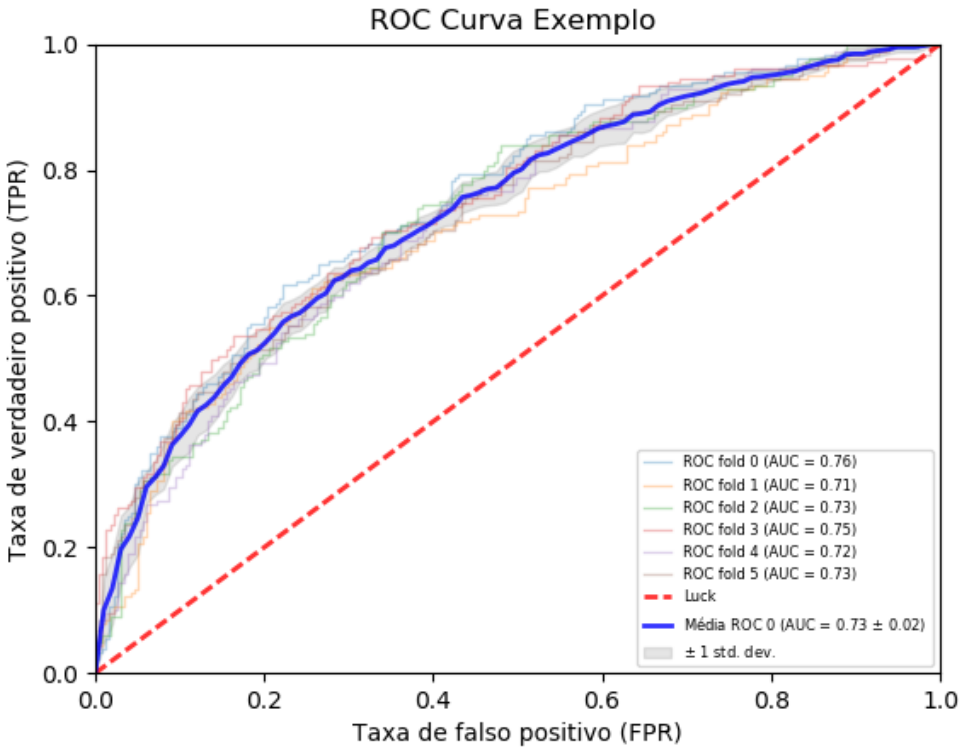


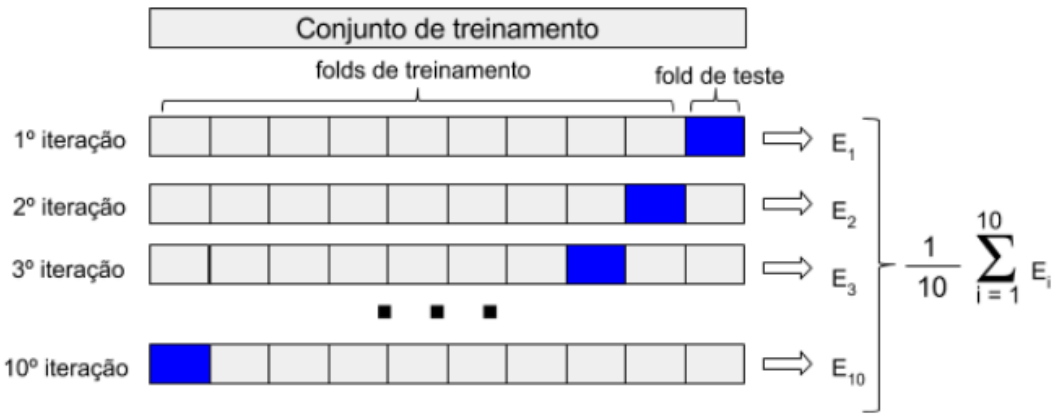
Figura 7 – Exemplo de curva ROC

2.6.1.0.1 Definição dos conjuntos de treinamento e teste

Tendo as métricas, devemos definir que dados serão utilizados para a avaliação do modelo aprendido. Uma das técnicas mais simples que podemos utilizar é a divisão dos exemplos de entrada em dois conjuntos(CAMILO; SILVA, 2009): O conjunto de treinamento (*training set*), que é conjunto de exemplos utilizados para criação do modelo e o conjunto de testes (*test set*), que é o conjunto de exemplos que serão utilizados para testar o modelo construído. Podemos escolher a proporção de quanto dos dados serão de treinamento e quantos serão de teste. O modelo irá classificar os exemplos de teste e, dependendo da classificação retornada e da classe original do exemplo, é possível extrair os valores de TN, FN, TP, FP para o calculo das métricas.

Outra técnica de validação consiste em dividir o conjunto de exemplos de forma aleatória em k subconjuntos, e então, em k iterações, utilizar um dos conjuntos como teste e os demais como conjunto de treinamento. Esse processo se repete até que todos os subconjuntos tenham sido usados como conjunto de teste. Esse processo, chamado de *K-Fold Cross-Validation*, é representado pela Figura 8.

Figura 8 – Técnica *K-Fold Cross-Validation*



Stratified k-Fold Cross-Validation é uma variação da técnica do *K-Fold Cross-Validation* que busca criar os *folds* com uma proporção de exemplos de cada classe semelhante.

2.7 Trabalhos Relacionados

- *Pespective*[14]
  Recentemente, o Google[9] e Jigsaw (WULCZYN; THAIN; DIXON, 2017) lançaram um projeto chamado *Perspective*, cujo objetivo é detectar automaticamente insultos

[14] https://www.perspectiveapi.com/

online, assédio e discurso abusivo nos servidores do Google[9]. Atualmente esta API (Application Programming Interface) somente suporta o idioma inglês. Os dados utilizados para o aprendizado dessa ferramenta foram 160 mil comentários do Wikipédia[11] retirados de sua talk page, onde os usuários podem fazer discussões sobre os artigos. 5000 pessoas classificaram os comentários e cada comentário foi classificado por pelo menos 10 pessoas. Apesar de possuir uma base de treinamento robusta essa API possui algumas falhas, segundo Violet Blue (BLUE, 2017), a API *Perpective*[14], classificava comentários com sentido aposto aos discursos tóxicos, como tóxicos. por exemplo "eu tenho orgulho de ser negro"era classificado como 56% tóxico. Provavelmente termos usados para representar os alvos de discursos tóxicos ganharam uma importância maior na classificação, sendo associados ao discurso tóxico, para lidar com este problema eles poderiam balancear sua base de treinamento inserindo mais exemplos positivos que usa-sem esses termos.

- *Nohomophobes*[15]
  Esse site é projetado como um espelho social, objetivando mostrar a prevalência da homofobia casual em nossa sociedade. Através da coleta de "tweets", contendo as palavras como "faggot", "dyke", "homo"e "so gay", essas palavras são utilizadas casualmente na linguagem cotidiana, apesar de promover a continuada alienação, isolamento e, em alguns casos, trágicos suicídios de pessoas pertencentes ao grupo LGBT. A página que exibe as estatística do site podem ser vistas na Figura 9

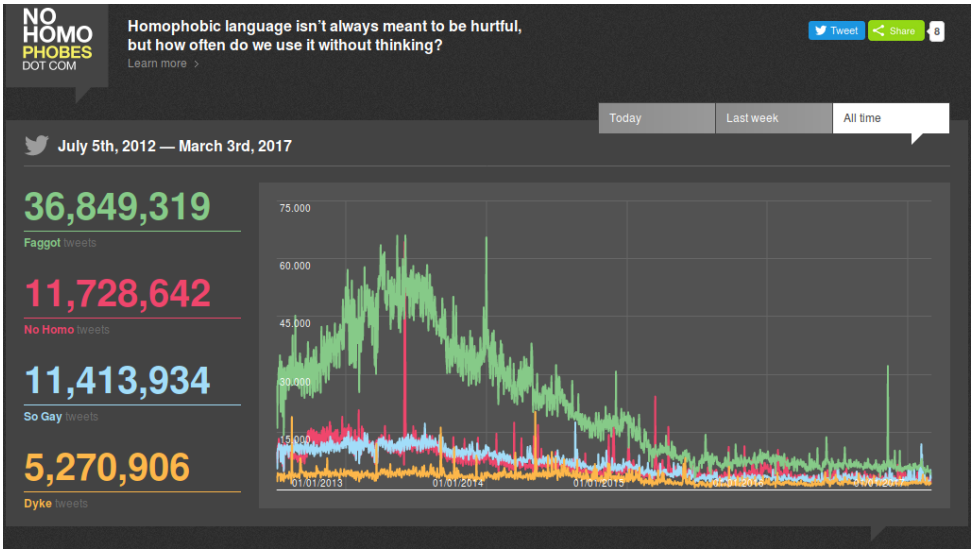


Figura 9 – Site do Nohomophobes (WELLS, 2012)

- *Attitude Buzz: Homophobia* [16]
  Neste trabalho (COHN; KUNTZ; BIRNBAUM, 2015) foi desenvolvido um modelo

[15] http://www.nohomophobes.com/
[16] http://attitudebuzz.infolab.northwestern.edu/

capaz de predizer se um *tweet* possui sentimentos positivos ou negativos voltados ao público LGBT. Para treinar esse modelo, foi construído dois tipos de corpus, um formado por tweets que continham termos referentes a identidade sexual originários de regiões onde as pessoas são historicamente simpatizantes da causa LGBT, esse tweets foram rotulados como não pejorativo. Os tweets vindo de outras regiões aleatórias foram rotulados como pejorativos. Com isso eles treinaram um modelo usando *Suport Vector Machine* e passaram a classificar tweets de diversas regiões. Por fim, criaram um site para mapear os resultados, visto na Figura 10. Esse site mostra um mapa mundial onde usuário pode consultar por locais específicos e visualizar a densidade de tweets pejorativos realizados naquela região.

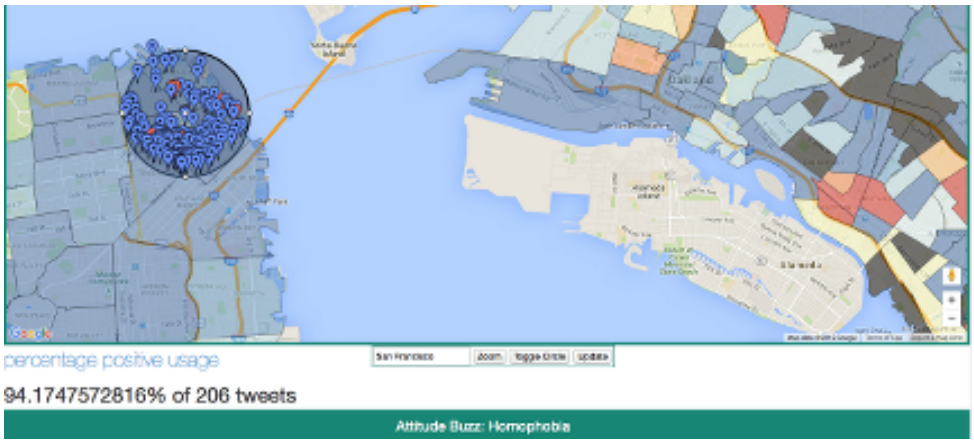


Figura 10 – Site do Attitude Buzz

- *Hate Speech, Machine Classification and Statistical Modelling of Information Flows on Twitter*[3]*: Interpretation and Communication for Policy Decision Making* (BURNAP; WILLIAMS, 2014)
  Com o objetivo de auxiliar as decisões da polícia, nesse trabalho foi construído um classificador para decidir, entre outras coisas, se um *tweet* continha uma mensagem odiosa e/ou antagônica a cerca de raça, etnia, ou religião Dos 450.000 tweets coletados, foram amostrados 2.000 para serem classificados por humanos. Para classificar cada tweet foi feita a pergunta: "este texto é ofensivo ou antagônico em termos de raça, etnia ou religião?". Eles usaram *k-Fold Cross-Validation* com k = 10 para avaliar suas combinações de algoritmos de aprendizado de máquina e técnicas como n-gram. Nesse trabalho eles ressaltam a dificuldade de seres humanos classificarem manualmente um número grande de tweets e aceitam as limitações que o modelo preditivo gerado por eles possue, afirmando que existem muitas combinações e permutações dos termos utilizados em discursos de ódio, o que dificulta a classificação. Por fim eles desenvolveram um modelo preditivo de aprendizado de máquina supervisionado cujo propósito era detectar conteúdo odioso e antagônico para auxiliar a policia na tomada de decisões.

- *Analyzing the Targets of Hate in Online Social Media* (SILVA et al., 2016)
  Nesse trabalho foram coletados em torno de 530 milhões de de postagens em inglês de duas redes socais, o Twitter[3] e o Whisper[17], durante um ano. Dessas, foram selecionadas aquelas que continham a seguinte estrutura .
  I < intensity >< userintent >< Any word>

  o campo <userintent> é formado pela palavra ódio ou os sinônimos da mesma tirados de um dicionario online[18], o campo <intensity> recebe os qualificadores usados pelos usuários para expressar suas emoções, por exemplo advérbios. Manualmente eles removeram dos comentários filtrados por essa estrutura , aqueles que apresentavam o sentido negativo, como i don't hate you. Para descobrir os valores para o campo <hatetarget>, ao tentar definir que todas as palavras presentes após o inicio da estrutura fossem as representações dos alvos.
  I < intensity >< userintent >< any word>
  Porém esse método retorna muitas informações irrelevantes e então tentaram usar uma estrutura simples que define como alvo do discurso de ódio a primeira palavra que aparece após a estrutura.
  I < intensity >< userintent >< one word>
  mas com isso foram perdidos os casos como i hate any people. Após essas tentativas eles utilizaram uma base de dados[19] com 1078 palavras de ódio que abrangem oito categorias:arcaico, classe, deficiência, etnia, gênero, nacionalidade,religião orientação sexual e sexual. E cada palavra dessa base vem com uma pontuação de ofensividade que vai de 0 a 100. Nesse trabalho eles utilizaram somente as que possuíam 50 ou mais pontos. Essas palavras foram utilizadas no campo <hatetarget>. Por fim como resultado deste trabalho, eles conseguiram definir quais eram os alvos mais frequentes nessas redes sociais, dividiram esses alvos em grupos e descobriram a frequência com que cada grupo se tornava alvo dos discursos de ódio.

2.7.1 Análise dos trabalhos

O projeto mais semelhante ao proposto por este trabalho foi o Pespective[14] do Google[9], porém ele foi um projeto em larga escala feito somente em inglês, e ainda hoje ele classifica como tóxicos frases sem contexto negativo com facilidade. Basta apenas ter algum termo que referencie um grupo marginalizado para que ele classifique como tóxico. Para evitar este tipo de problema tentamos adicionar mais informações que remetam ao contexto do comentário. Nohomophobes[15] é popular porém usa uma classificação bem ingênua por palavras chave, sendo que o público LGBT costuma remover o sentido negativo de algumas

[17] http://whisper.sh/
[18] http://www.thesaurus.com/browse/hate/verb
[19] https://www.hatebase.org/

palavras a fim de diminuir o poder nocivo dos discursos de ódio. Por exemplo, a palavra "bicha"antes só usada em contextos negativos agora é usada em frases de empoderamento como "arrazo! bicha". O AttitudeBuzz[16] utilizou uma forma bem diferente de rotular os tweets, usando as regiões ele conseguiu uma grande quantidade de tweets rotulados porém dessa forma as classificações não ficam precisas. O trabalho que desenvolveu o modelo preditivo para auxilio da decisão humana escolheu um caminho seguro, pois a responsabilidade de julgar como um crime ou não o conteúdo, torna o risco associado ao programa muito alto. Para evitar o problema explicitado por eles de que os seres humanos possuem dificuldade para classificar quantidades grandes de texto. Foi desenvolvida uma interface web que permite muitos usuários classificarem cada um alguns comentários. Por fim o trabalho sobre os alvos do discurso de ódio criou uma espécie de filtro, que é limitado a aquela estrutura, aumentando a dificuldade de modificação da forma de detecção, problema inexistente no aprendizado de máquina que utiliza os dados de entrada e desenvolve sozinho a lógica. Á ultima grande diferença entre os trabalhos apresentados e o algoritmo proposto e a linguagem na qual ele foi desenvolvido, este trabalho está sendo desenvolvido com exemplos em português mantendo o traço cultural que a língua possue e detectando discursos de ódio específicos do português. No próximo capitulo veremos a forma com que esse algoritmo foi desenvolvido.

3 Uma Ferramenta para Auxiliar a Detecção de Discurso de Ódio em Mídias Sociais

Nesse capítulo apresentamos a ferramenta desenvolvida no trabalho, cujo objetivo é coletar e classificar notícias de jornais online como sendo discurso de ódio ou não, usando técnicas de Aprendizado de Máquina (MITCHELL; HILL, 1997). São apresentados os pseudo códigos que serviram como base para a implementação da ferramenta, que foi desenvolvida na linguagem de programação Python[1] e utiliza ferramentas da biblioteca de Aprendizado de Máquina de código livre e aberto, chamada Scikit-learn[2](PEDREGOSA et al., 2011).

Os dados utilizados no trabalho foram extraídos de uma plataforma de jornal online e salvos no banco de dados NoSQL MongoDB[3]. Dentre os dados extraídos temos as notícias e os comentários introduzidos por leitores do jornal. Tais comentários são manualmente anotados por usuários voluntários, onde são rotulados como contendo discurso de ódio, discurso ofensivo ou um discurso neutro. Esse passo é necessário para a criação da base de dados utilizada no processo de treinamento de um classificador, usando aprendizado supervisionado.

O processo de treinamento do classificador foi dividido em X etapas, como pode ser observado na Figura 11. Cada etapa que necessite de uma explicação mais aprofundada será descrita nas próximas seções.

Após o treinamento, a ferramenta pode ser acoplada como um *plugin* no jornal, de forma a detectar automaticamente a partir dos comentários aqueles que contém ou não um discurso de ódio.

[1] https://www.python.org/
[2] http://scikit-learn.org/
[3] https://www.mongodb.com

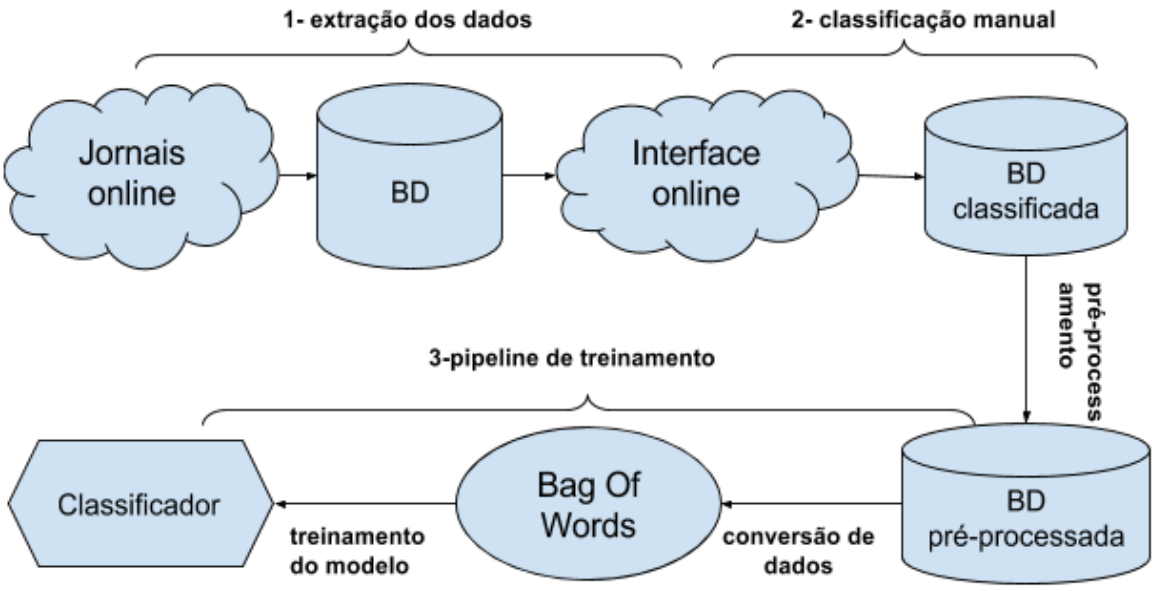


Figura 11 – Etapas do processo de Criação do classificador

3.1 Extração dos Dados

Algumas mídias digitais disponibilizam uma *Application Programming Interface* (API) de coleta de dados, para facilitar o trabalho dos pesquisadores e desenvolvedores de ferramentas que utilizem tais dados. Dentre elas, temos o Facebook, cuja a API possui algumas limitações de acesso, retornando apenas dados marcados como públicos por seus usuários. O Twitter, por outro lado, possui limitações na quantidade diária de acessos, porém possui diversas ferramentas úteis para efetuar buscas de dados mais precisas. Dados oriundos desse tipo de fonte geralmente são bem estruturados e possuem identificadores da própria mídia social, como o identificador do usuário e o identificador da postagem.

Em sites que não possuem esse tipo de API, a alternativa é fazer uma raspagem de dados (*web scrapping*). Nesse caso, é necessário desenvolver uma aplicação própria que navegue entre as páginas coletando informação do próprio código HTML e dos scripts da página, efetuando requisições HTTP que retornem dados dispostos nas páginas, e salvando tais dados em um formato (preferencialmente) estruturado, como JSON, XML ou CSV. Contudo, cada aplicação de *web scrapping* deve ser projetada de acordo com o HTML da página da qual ela precisa extrair dados, e quanto mais bem estruturada a página, mais fácil será a extração e melhor a qualidade dos dados. Uma vantagem deste método em relação às APIs é o acesso a uma quantidade maior de dados.

Nesse trabalho utilizamos a técnica da raspagem de dados na plataforma de

jornalismo online G1[4]. O G1[4] é um importante portal de notícias online das organizações Globo, que disponibiliza serviços como cadastro de usuários e compartilhamento em redes sociais. Nele, os usuários podem interagir com a notícia e com outros usuários a partir de comentários.

Esses comentários são fornecidos para a página através de um serviço web, que retorna uma lista de comentários e outras informações no formato *JSON* (JavaScript Object Notation)(MEDEIROS, 2012). Os resultados das requisições serão lidos e salvos pela ferramenta em um banco de dados NoSQL chamado MongoDB[3]. Esse banco de dados foi escolhido pois armazena e consulta dados no formato JSON, facilitando o armazenamento direto dos dados extraídos do G1[4]. Após a coleta e armazenamento dos blocos de comentários de diversas notícias do G1[4], a ferramenta de extração analisa essas informações e seleciona os dados necessários para criar as entidades, ou seja, a notícia e os comentários, no formato JSON, e as salva no MongoDB[3]. Na figura 12 é mostrado um exemplo de notícia e comentário pertencente aquela noticia no formato JSON.

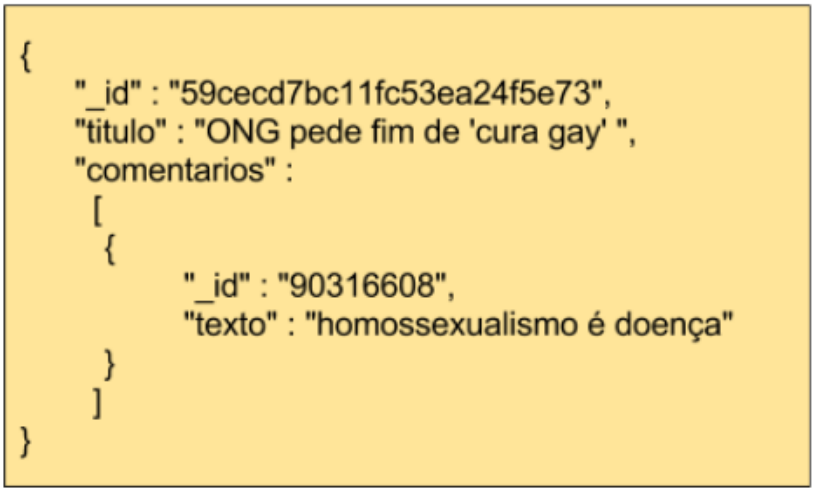


Figura 12 – Exemplo de noticia no formato JSON

No Algoritmo 1 é exibido o procedimento utilizado para extrair os dados. A aplicação responsável pelo *Web scrapping* foi desenvolvida em Python[1], com destaque para a biblioteca *BeautifulSoup*[5], responsável pela leitura da página e pela busca de elementos HTML na mesma. Para requisições HTTP, foi utilizada a biblioteca do Python [1]chamada *urllib*, e para a leitura e escrita de objetos json foi utilizada a biblioteca do Python[1] chamada *json*. Para salvar os dados extraídos no MongoDB[3] foi utilizada a biblioteca feita em Python[1] chamada PyMongo[6].

[4]http://g1.globo.com/
[5]https://www.crummy.com/software/BeautifulSoup/bs4/doc/
[6]https://api.mongodb.com/python/current/

Algoritmo 1 Processo de coleta de Notícias e Comentários

```
1:  função COLETAR(nPaginas,termo)
2:      para i ← 1 até nPaginas faça
3:          urlBusca ⇐ construirUrlBusca(i, termo)
4:          links ⇐ buscaLinksNoticias(urlBusca)
5:          para j ← 1 até links.tamanho faça
6:              dadosRequest ⇐ coletarDados(links[j])
7:              pagina ⇐ 0
8:              noticia ⇐ criaNoticia()
9:              arquivoJson ⇐ getComentarios(dadosRequest, pagina)
10:             enquanto arquivoJson! = null faça
11:                 adicionaComentarios(noticia, arquivoJson)
12:                 pagina ⇐ pagina + 1
13:                 arquivoJson ⇐ getComentarios(dadosRequest, pagina)
14:             fim enquanto
15:             salvarNoticia(noticia)
16:         fim para
17:     fim para
18: fim função
```

O site do G1[4] possui uma ferramenta de busca que, ao digitar um texto no campo de busca e clicar em buscar, o usuário é redirecionado para uma página que contém diversos links para notícias. Quando a busca retorna muitos resultados, o G1[4] utiliza um mecanismo de paginação, dividindo o total de resultados em N páginas. O Algoritmo 1 recebe a variável $nPaginas$, que representa a quantidade de vezes que o algoritmo irá avançar de página na paginação dos resultados, e a variável $termo$, que é o texto utilizado na busca.

Seguindo o algoritmo, na linha 3 é Utilizado o método $construirURL(i, termo)$, que criar a URL do site que contém a página de resultados da busca. Por exemplo, a URL: <http://g1.globo.com/busca/?q=gay&order=recent&species=noticiascias&page=1>

Na linha 4 é executada uma requisição com GET a partir da URL gerada, e capturar o código HTML referente à página solicitada, esse código é analisado e dele são extraídos os links de notícias existentes na página, e, usando uma repetição, são capturados os comentários a partir de cada URL. Para tanto, coleta-se informações especificas da noticia com elas, se estabelece uma conexão com o servidor de comentários do G1[4] que retorna uma página de comentários a cada requisição. Por isso utilizamos outra repetição para que todas as páginas de comentários de uma notícia sejam buscadas. O resultado dessa requisição não vem de acordo com as entidades do sistema , por isso na linha 11 usa-se um método para extrair as informações do arquivo JSON e as adiciona no objeto notícia. Quando não houver mais comentários a serem buscados o objeto notícia é salvo no mongoDB[3]. Após capturar todas as páginas de noticias, avança-se para a próxima página de busca e repete-se o processo.

3.2 Processo de Anotação Manual dos Dados para o Treinamento

Como não foram encontradas exemplos de treinamento na língua portuguesa, com foco em discurso de ódio inseridos em mídias sociais, foi necessário recorrer a um método de anotação usando voluntários humanos. Embora existam bases de dados na língua inglesa, os termos utilizados em discursos de ódio não são traduzíveis e não possuem os traços culturais relacionados ao cidadãos brasileiros.

Para fazer a anotação manual dos comentários coletados do G1, foi criada uma interface web em Java, usando o framework *Spring Boot*[7](AFONSO, 2012). Spring Boot[7] é um projeto da *Spring*[8], que foi criado para facilitar a criação de aplicações web, facilitando o uso de demais frameworks desenvolvidos pela empresa. Para tanto, eles desenvolveram um método que permite que o programador selecione os módulos do Spring que ele precisa utilizar através de um único arquivo, além de possuir configurações padrão para cada um desses módulos, e permitir configurações personalizadas de forma simples também através de um único arquivo.

O *implantação* da interface, que nessa caso trata-se de instalar a aplicação em um servidor e permitir o acesso aos usuários online, foi feita na plataforma do *Heroku*[9] que possui integração com o Spring Boot[7]. Foi utilizado o plano gratuito que possui varias limitações, como o tamanho máximo do Banco de Dados, que suporta no máximo 10 mil linhas. Essa interface permite o acesso simultâneo de vários usuários para que os mesmos possam classificar os comentários de forma paralela.

O público alvo que espera-se que vá usar a ferramenta são pessoas pertencentes ao grupo LGBT, com um grau de ensino médio ou superior, e que se voluntariaram para classificar os comentários. Os comentários são divididos em pacotes de forma aleatória de forma que cada usuário pode escolher um pacote para anotar, uma vez que pode ser custoso anotar muitos comentários. Segue abaixo a descrição das telas da interface:

- Tela de login
  Tela o usuário insere seu e-mail para identificar-se e ter acesso ao seu progresso na interface. Ou seja os comentários anotados por ele não são exibidos novamente para anotação. Não é necessário cadastro e nem senha, para facilitar a aderência de novos usuários.

- Tela de seleção do pacote de notícias
  Após o login, o usuário deve selecionar um pacote de notícia entre os X pacotes nos quais foram divididos os comentários filtrados.

[7]https://projects.spring.io/spring-boot/
[8]https://spring.io/
[9]https://www.heroku.com/

- Tela de seleção de notícias
  Essa tela apresenta as notícias em uma tabela, para que o usuário escolha a ordem em que quer classificar as notícias. Aquelas que forem anotadas são removidas da lista, para ficar evidente o progresso do usuário.

- Tela de classificação de comentários
  Ao selecionar um item na tela de seleção de notícias, o usuário é redirecionado para uma tela que apresenta o título da noticia e os comentários para classificar. Observe que essa também será a entrada do classificador, ou seja, o título da notícia concatenado aos comentários, para solucionar o problema da falta de contexto que o comentário possui. Os comentários são exibidos em uma tabela onde a primeira coluna contém o texto do comentário, a segunda contém o botão de classificação como discurso de ódio, a terceira o botão de classificação como ofensivo e a quarta com o botão de classificação como neutro. Todos os três botões possuem cores diferentes e textos, para deixar clara a diferença entre eles. Ao manter o mouse por cima de cada botão, uma pequena janela se abre acima do mesmo, com a descrição da classe a qual ele representa. Ao clicar no botão, o comentário é retirado da tela e a classificação do usuário é salva no Banco de Dados. Essa tela é vista na Figura 13

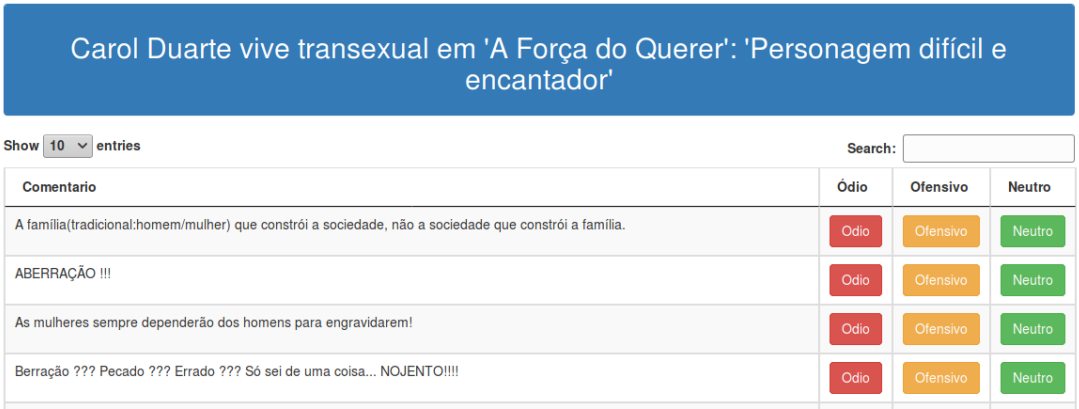


Figura 13 – Interface de classificação online

3.3 Pré-processamento dos Dados

A fim de reduzir a quantidade de ruídos nos dados, que podem acabar por interferir na qualidade do classificador, os comentários coletados devem passar por um pipeline de processamento, conforme mostrado no Algoritmo2

As etapas de um a quatro do algoritmo padronizam o texto como descrito no capítulo 2. Inicialmente, os caracteres são convertidos para minúsculo. A seguir, as palavras mascaradas com espaço entre as letras são tratadas. Por exemplo, h o m o s s e x u a l i s m o é transformada para homossexualismo. O método verifica se a próxima palavra é

composta de uma única letra e a une com a palavra anterior.
A seguir, são removidas letras repetidas de palavras. Por exemplo, a palavra kkkkkk é transformado em k, letras do português que podem aparecer em pares como s ou r, são reduzidas a duas letras, por exemplo a palavra pessssoas é transformada em pessoa. Na etapa 4, utiliza-se um método que remove caracteres especiais, dado que alguns usuários, a fim de burlar os sistemas de denúncia, escrevem discursos de ódio mascarados por esse caracteres. A pontuação no final das palavras é mantida. Por exemplo "Ga.y.s?"se transforma em "Gays?".
Na etapa 5, utiliza-se um método que substitui as pontuações que não foram removidas na linha 4 por tags, para que as mesmas possam ser usadas como atributos da base de dados. Os pontos finais são substituídos pela tag "_F", os pontos de interrogação são substituídos pela tag "_Q"e os pontos de exclamação são substituídos por "_X".

As etapas 6 e 7 compreendem as técnicas de remoção de *stop words* e de *stemming*, descritas no capitulo 2. Essas técnicas são disponibilizadas no conjunto de bibliotecas de processamento de texto do NLTK(*Natural Language Toolkit*)[10](BIRD; KLEIN; LOPER, 2009).

Algoritmo 2 Pré processamento de dados

1: **função** PROCESSA(texto)
2: $texto \Leftarrow transformarMinusculo(texto)$
3: $texto \Leftarrow removerEspacosBrancos(texto)$
4: $texto \Leftarrow removerLetrasRepetidas(texto)$
5: $texto \Leftarrow removerCaracteresEspeciais(texto)$
6: $texto \Leftarrow transformarPontuacaoTag(texto)$
7: $texto \Leftarrow removerPalavrasPorTamanho(texto, min, max)$
8: $texto \Leftarrow removerStopWords(texto)$
9: $texto \Leftarrow removerRadicais(texto)$
10: **devolve** texto
11: **fim função**

Na Tabela 10 é mostrado o passo a passo do Algoritmo 2 com entrada "Ga.y.s t e m q.u.e morrrr;er?"

[10]http://www.nltk.org/

Tabela 10 – Exemplo de passo a passo do algoritmo com a entrada Ga.y.s t e m q.u.e morrrr;er?

Etapa	**Descrição**	**Resultado**
1	Transforma texto em minusculo	ga.y.s t e m q.u.e morrrr;er?"
2	Remove os espaços em branco do texto	ga.y.s tem q.u.e morrrr;er?
3	Remove letras repetidas	ga.y.s tem q.u.e morr;er?
4	Remove caracteres especias	gays tem que morrer?
5	Transforma pontuação em tags	gays tem que morrer _Q
6	Remove as *stop words*	gays morrer _Q
7	Remove os radicais	gays morr _Q"

Na Tabela 11 é exemplificado o resultado do Algoritmo 2 em um comentário extraído de uma noticia.

Tabela 11 – Exemplos de pré processamento de texto

	Entrada	**Entrada pré processada**
TÍtulo	Parada Gay reúne multidão na Praia de Copacabana	par gay reun multida pra copacaban
Comentário	Uma criança criada por essas aberrações não vai ter escolha vai virar aberração	crianc cri aberraco nao vai ter escolh vai vir aberraca

3.4 Pipeline de Treinamento

Algoritmo 3 Pipeline de treinamento

```
1:  função CONSTROIPIPELINE(dados,tags,opts)
2:      dados ⇐ processarDados(dados)
3:      extOp, descOp, selOp, clasOp, tagsOp ⇐ opts
4:      tags ⇐ refatoraTags(tags, tagsOp)
5:      extrator ⇐ criarExtrator(extOp)
6:      dados ⇐ extrator.transforma(dados)
7:      decompositor ⇐ criarDecompositor(descOp)
8:      dados ⇐ decompositor.transforma(dados)
9:      selecionador ⇐ criarSelecionador(selOp)
10:     dados ⇐ selecionador.transforma(dados)
11:     classificador ⇐ criarClassificador(clasOp)
12:     modelo ⇐ classificador.treina(dados, classes)
13:     devolve modelo
14: fim função
```

No algoritmo 3 é descrito o funcionamento do componente responsável pelo processo de treinamento do classificador. Tal componente possui quatro sub-componentes: (1) criação dos atributos; (2) redução de dimensionalidade por fatorização de matriz; (3) seleção de atributos; (4) treinamento do classificador. As etapas (2) e (3) são opcionais, e normalmente

são exclusivas. O algoritmo recebe parâmetros que indicam quais técnicas ele irá utilizar em cada sub-componente. Isso é feito por meio da implementação de métodos que recebem esses parâmetros e retornam instâncias de objetos configurados. Por exemplo {"BOW-TF-UNIGRAM": CountVectorizer(1,1),"BOW-TF-BIGRAM":CountVectorizer(1,2),"BOW-TFIDF-BIGRAM":TfidfVectorizer((1, 2))}, no Sckit-Learn[11] CountVectorizer é o método que representa o bag-of-words com frequência de termos e ele possui um parâmetro no construtor que representa a técnica do N-GRAM, esse parâmetro como (1,1) representa a a configuração padrão onde cada termo será um atributo, e (1,2) representa a técnica do BI-GRAM. Já o TfidfVectorizer representa o uso da técnica TF-IDF para preencher o bag-of-words, também recebe o N-GRAM como parâmetro. o que permite realizar várias combinações de técnicas e comparações entre resultados.

Os objetos utilizados nesses métodos são todos implementações da Sckit-Learn[11]. As ferramentas de extração de atributos são encontradas no pacote $sklearn.feature\_extraction.text$, as ferramentas de decomposição são encontradas no pacote $sklearn.decomposition$, as ferramentas de seleção são encontradas no pacote $sklearn.feature\_selection$. As ferramentas que geram o modelo preditivo são encontradas em diversos pacotes cujo nome representa o tipo de algoritmo por exemplo $sklearn.ensemble$, $sklearn.svmesklearn.naive_bayes$.

A seguir é apresentada um descrição detalhada de cada etapa do Algoritmo 3. Na etapa 2, executa-se um método que transforma a lista de textos em uma lista de textos pré processados pelo algoritmo 2.
Na etapa 3, preenche-se as variáveis de configuração com os valores contidos no vetor de configurações *opts*.
Na linha 4, transforma-se ou não as classes do texto, para codificar uma classe nominal em numérica. Por exemplo, {"Odio":0,"Ofensivo":1,"Neutro",2} . Outro mapeamento efetuado é unir duas classes em uma só, para a execução dos testes, como veremos no próximo capítulo. Por exemplo, {"Odio":"Odio","Ofensivo":"Outro","Neutro","Outro"}, transformando uma classificação ternária em binária.

Na etapa 5, cria-se o extrator configurado através da variável de configuração *extOp*. Por exemplo, podemos passar como parâmetro uma configuração que faça com que o método $criarExtrator()$ crie uma bag-of-words a partir a partir do método do TF-IDF. Na etapa 6, transforma-se os dados através do extrator.

Na etapa 7, cria-se (ou não) o decompositor de atributos, configurado através da variável de configuração *descOp*. Por exemplo, podemos passar como parâmetro uma configuração que faça com que o método $criarDecompositor()$ crie um decompositor do tipo LDA. Na etapa 8, utiliza-se o decompositor de atributos criado na etapa anterior para decompor os dados.

[11] http://scikit-learn.org/

Na etapa 9, cria-se (ou não) o selecionador de atributos configurado através da variável de configuração *selOp*. Por exemplo, podemos passar como parâmetro uma configuração que faça com que o método $criarSelecionador()$ crie um selecionador do tipo $SelectKBest$. Na etapa 10, utiliza-se o selecionador de atributos criado na etapa anterior para selecionar os dados que passarão para a próxima etapa.

Na etapa 11, cria-se o classificador, usando um algoritmo de aprendizado de máquina supervisionado, configurado através da variável de configuração *clasOp*. Por exemplo, podemos passar como parâmetro uma configuração que faça com que o método $criarClassificador()$ crie um classificador do tipo Naive Bayes ou até mesmo um do tipo *Ensemble*, formado por outros classificadores.

Finalmente, na etapa 12, utiliza-se o classificador criado na etapa anterior para treinar um modelo preditivo.

No próximo capitulo veremos o uso de técnicas de avaliação no auxilio da escolha dos parâmetros do algoritmo 3 e a escolha final de configurações do pipeline, que irá gerar o modelo capaz de classificar em tempo de execução. Dessa forma, ele poderia ser implantado em outros jornais online, com os devidos ajustes no processo de captura e tratamento dos dados, como um plugin.

4 Estudo de Caso: Classificando Comentários de Reportagens no Jornal G1

Nesse capítulo apresentaremos os resultados dos testes e avaliações realizados com a ferramenta desenvolvida nesse trabalho, englobando o resultado das combinações de técnicas testadas e a qualidade do classificador para cada conjunto de dados coletado. Esses dados são oriundos da raspagem de dados realizada no portal de jornalismo online G1[4], onde os dados extraídos foram modelados em duas entidades: os comentários e as notícias. Toda notícia possui uma lista de comentários. No processo de anotação manual realizado na interface web, cada anotação em um comentário gera uma entidade do tipo classificação. Para um mesmo comentário contamos com uma ou mais classificações. A maior vantagem de se utilizar comentários de notícias online para treinar o classificador é a informação sobre o contexto que é agregada ao comentário através da notícia.

4.1 Metodologia Experimental

Nessa seção, descrevemos as bases de dados coletadas como parte desse trabalho, bem como o formato da metodologia experimental empregada na validação das possíveis técnicas a serem usadas para a geração do classificador.

4.1.1 Bases de Dados

Para que fosse possível utilizar as técnicas de aprendizado supervisionado, tornou-se necessário que os comentários associados a uma reportagem fossem manualmente classificados como pertencentes a uma das três classes mencionadas anteriormente, a saber: (1) discurso de ódio; (2) ofensivo; (3) neutro. Para prosseguir com tal anotação, estabelecemos dois requerimentos necessários ao processo de anotação manual das classes dos comentários. O primeiro requerimento era que os comentários fossem classificados apenas por pessoas que se declarassem como pertencentes à comunidade LGBT, uma vez que pessoas da comunidade LGBT são alvos frequentes desse tipo de discurso. Suas experiências, e as experiências compartilhadas pelas pessoas de seu convívio, podem auxiliar na anotação dos comentários. O segundo requerimento é que todos os comentários coletados a partir do jornal online G1[4] e dispostos na aplicação web fossem anotados pelo menos uma vez. Porém, para obter uma anotação mais precisa, o ideal é que mais pessoas anotem um mesmo comentário, e a classe final seja obtida por um método de agregação, por exemplo, a votação majoritária.

Inicialmente, foram coletados 6000 comentários de 241 notícias distintas, a fim de recuperar notícias relacionadas ao público LGBT. Os comentários foram filtrados utilizando de palavras chaves presentes em discurso de ódios, de acordo com pessoas LGBT em grupos do Facebook[5]. As palavras utilizadas foram { imoral, queimar, isla, pecado, nojo, cancer, sodoma, indecente, dst, aids, praga, arabia, nojento, dizimar, lixo, excretor, espancar, morrer, matar, matando, baitola, boiola, biba, bicha, bichinha, viado, sapatao, traveco, homo, aberracao, gay, doente, doenca, perversao}. A filtragem desses comentários resultou no total de 1597 dos 6000 comentários da base.

Os comentários selecionados foram disponibilizados na aplicação web para a anotação manual. Dos 1597 comentários coletados, apenas 456 foram anotados por três ou mais pessoas. Um usuário sozinho classificou todos os comentários, de forma que o segundo requerimento acima foi atendido. O autor do trabalho anotou outros 934 comentários para poder testar a classificação de comentários de fora do conjunto de treinamento.

O processo de anotação dos comentários na interface web ocorreu durante o período de um mês. Para realizar as anotações contamos com a ajuda de trinta voluntários,que aceitaram um pedido de ajuda amplamente divulgado nas redes sociais. Esse pedido foi feito em grupos de pessoas LGBTs existentes no Facebook[5], tais como, *Diversitas UFF, LGBT Niterói, e GDN - Grupo Diversidade Niterói*. Segundo auto declarações dos voluntários, o grupo de voluntários era composto por pessoas cis (se identifica pelo gênero que o gênero que lhe registraram quando nasceu) e trans (oposto de cis), pessoas homossexuais e bissexuais. Após o processo de anotação, dos comentários coletados, organizamos as classificações geradas em 3 bases de treinamento, conforme descrito na Tabela 12.

Observa-se que as bases de dados B1 e B2 possuem intercessão, dado que as anotações do usuário que classificou todos os comentários contam no cálculo de quantidade de vezes que um comentário foi anotado.

Tabela 12 – Descrição das bases de dados de classificações utilizadas como conjuntos de treinamento

Identificador	**Descrição**	**Neutro**	**Ódio**	**Ofensivo**	**Total**
B1	classificações do usuário que completou todos os comentários	753	383	461	1597
B2	classificações dos comentários que tiveram 3 ou mais classificações	592	566	612	1770
B3	classificações feitas em outras notícias de fora do aplicativo	348	225	361	934

Tabela 13 – Descrição dos pipelines utilizados nos testes

ID	Extrator	Bi-Gram	Seletor	Decompositor	Classificador
P1	BOW+TF	não	não	não	Naive Bayes
P2	BOW+TF	não	não	não	SVM
P3	BOW+TF	sim	não	não	Naive Bayes
P4	BOW+TF	sim	não	não	SVM
P5	BOW+TF-IDF	não	não	não	Naive Bayes
P6	BOW+TF-IDF	não	não	não	SVM
P7	BOW+TF	não	KBEST	não	Naive Bayes
P8	BOW+TF	não	KBEST	não	SVM
P9	BOW+TF	não	FROM MODEL	não	Naive Bayes
P10	BOW+TF	não	FROM MODEL	não	SVM
P11	BOW+TF	não	não	não	Bagging(Naive Bayes)
P12	BOW+TF	não	não	não	Bagging(SVM)
P13	BOW+TF	não	não	LSA	SVM

4.1.2 Treinamento dos classificadores

Na Tabela 13 as colunas representam a entrada de parâmetros do algoritmo 3 e cada linha representa uma configuração de pipeline. Os nomes e atributos foram baseados nas implementações do Sckit-Learn[11]. Foram escolhidas algumas configurações que permitiram verificar a mudança de resultado para cada técnica em relação ao resultado de um pipeline que foi escolhido como ponto de referência, no processo de treinamento desses pipelines foram utilizadas a união das bases de dados $B1$ e $B2$ descritas na tabela 12, e o processo de avaliação desses pipelines foi realizado, a partir do processo de análise dos gráficos ROC e PR. Foram criados um gráfico ROC (*Receiver Operating Characteristic*) para cada combinação de classe e técnica de avaliação usadas para gerar as métricas utilizadas na construção do gráfico. Nesses gráficos se desenhou uma curva ROC para cada pipeline da tabela 13. Deve-se levar em consideração que a classe escolhida para cada gráfico ROC era considerada positiva e as restantes eram consideradas negativas, estratégia necessária pois esse processo avaliador só suporta classificação binária. O algoritmo que gera a curva foi retirado do tutorial do próprio Sckit-Learn[11] e adaptado para gerar no mesmo gráfico a curva de todos os pipelines. O mesmo processo foi utilizado no desenvolvimento dos gráficos PR.

4.2 Resultados Experimentais

Nesta seção veremos o resultado das comparações realizadas entre os pipelines, analisaremos os gráficos gerados pelas diferentes técnicas de avaliação. As principais comparações realizadas foram entre os subgrupos de pipelines que representam cada técnica e o subgrupo de pipelines formados pelos pipelines básicos usados como referência. E as comparações feitas entre os pipelines de um mesmo subgrupo a fim de definir qual deles

se adaptou melhor a técnica que aquele subgrupo representa. Os subgrupos selecionados foram:

- (P1,P2)
  Subgrupo formado pelos pipelines básicos compostos da técnica do bag-of-words a partir da frequência de termos e do algoritmo de aprendizado de máquina.
- (P3,P4)
  Subgrupo formado pelos pipelines que representam o uso da técnica BI -GRAM.
- (P5,P6)
  Subgrupo formado pelos pipelines que representam o uso da técnica TF-IDF.
- (P7,P8)
  Subgrupo formado pelos pipelines que representam o uso da técnica SELECT K BEST. Com parâmetro K = 1500
- (P9,P10)
  Subgrupo formado pelos pipelines que representam o uso da técnica SELECT FROM MODEL utilizando SVM como base. Utilizando SVM com o parâmetro "penalidade"= "l1"
- (P11,P12)
  Subgrupo formado pelos pipelines que representam o uso da técnica Bagging. Com parâmetro "n_estimators"=100
- (P13)
  Subgrupo formado pelo pipeline que representa o uso da técnica LSA. Com parâmetro "n_components"=1000

Os resultados dessas comparações são representados em tabelas, usando o identificador do pipeline e um símbolo característico. As comparações que verificam a eficácia das técnicas utilizadas na criação do modelo utilizam a pipeline básica PB, que é formada apenas por um Bag-of-Words a partir da frequência de termos, e um algoritmo de Aprendizado de Máquina. As representações são PX_-, que indica que a configuração do pipeline PX gera um modelo inferior ao modelo gerado por PB, PX_+ a configuração do pipeline PX gera um modelo superior ao modelo gerado por PB, $PX_=$ a configuração do pipeline PX gera um modelo semelhante ao modelo gerado por PB. Os parâmetros utilizados em cada técnica fora selecionados através de tentativa e erro, utilizando o método chamado *GridSearch* que recebe conjunto de parâmetros, e testa todas as combinações de parâmetros, onde cada combinação gera um classificador treinado através de uma mesma base de dados. Os classificadores gerados são avaliados e as configurações do classificador que obteve os melhores resultados são selecionadas.

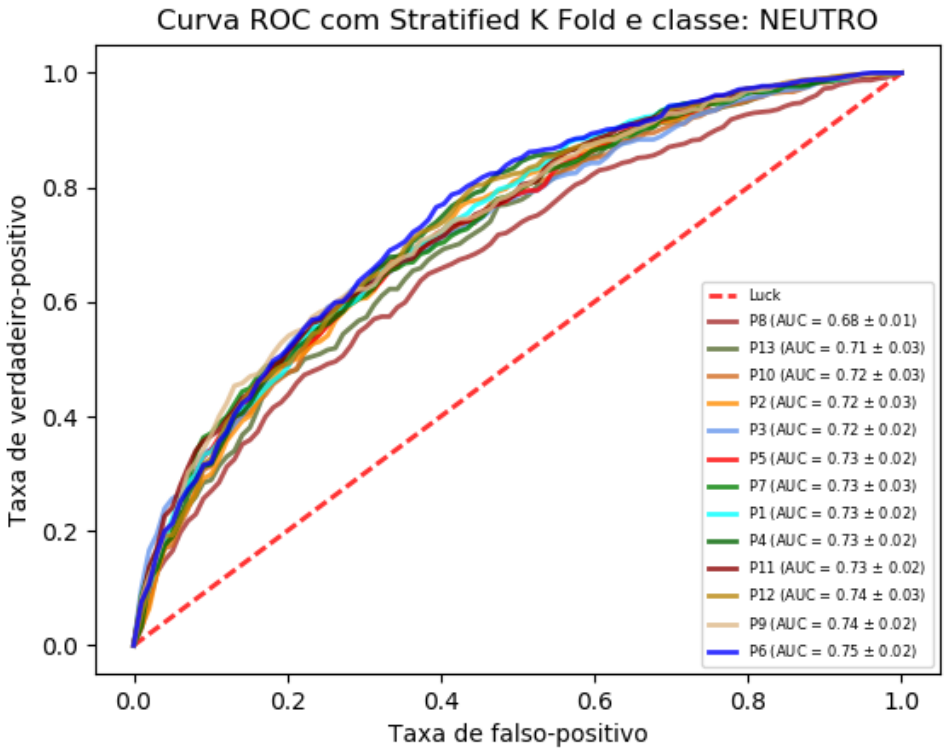


Figura 14 – Curva ROC(Receiver operating characteristic) para classe neutro

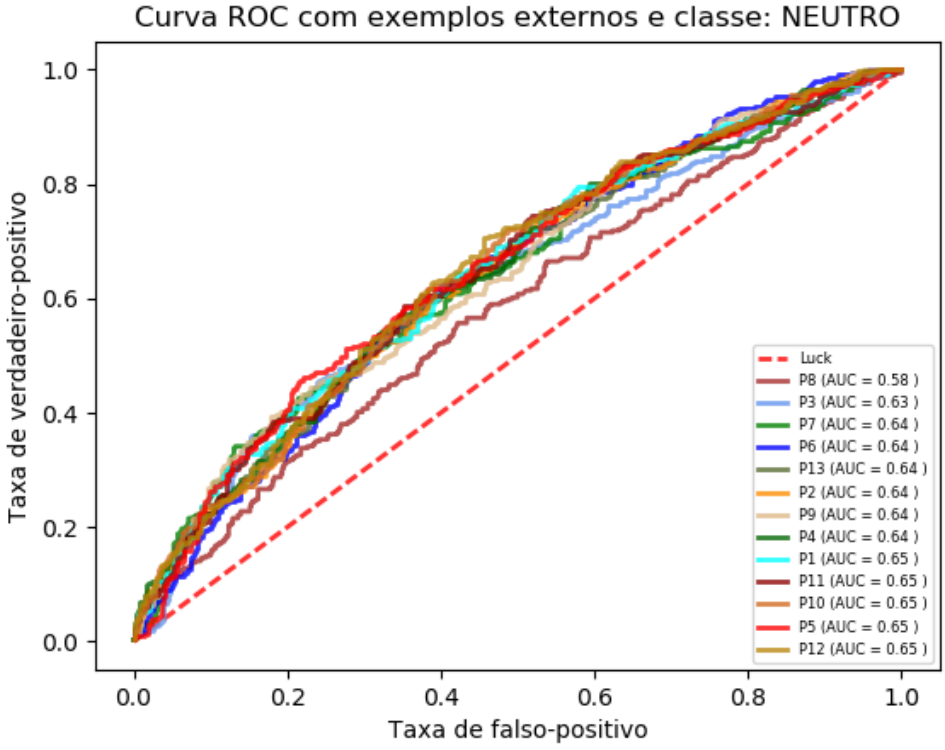


Figura 15 – Curva ROC(Receiver operating characteristic) para classe neutro

Tabela 14 – Analise dos Gráficos ROC da classe neutro

StratifiedKFold		**Externos**	
NB	SVM	NB	SVM
$P3_{-}$	$P4_{+}$	$P3_{-}$	$P4_{=}$
$P5_{=}$	$P6_{+}$	$P5_{=}$	$P6_{=}$
$P7_{=}$	$P8_{-}$	$P7_{-}$	$P8_{-}$
$P9_{+}$	$P10_{=}$	$P9_{-}$	$P10_{+}$
$P11_{=}$	$P12_{+}$	$P11_{=}$	$P12_{+}$
não	$P13_{-}$	não	$P13_{=}$

A Tabela 14 é preenchida através dos valores dos gráficos 14 e 15.

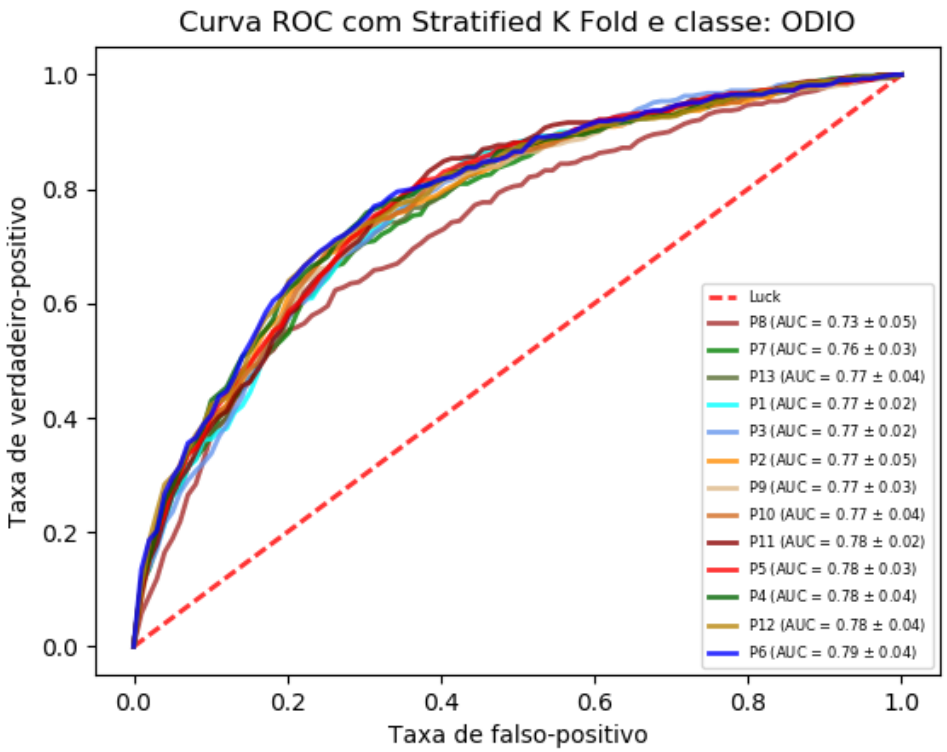


Figura 16 – Curva ROC(Receiver operating characteristic) para classe ódio

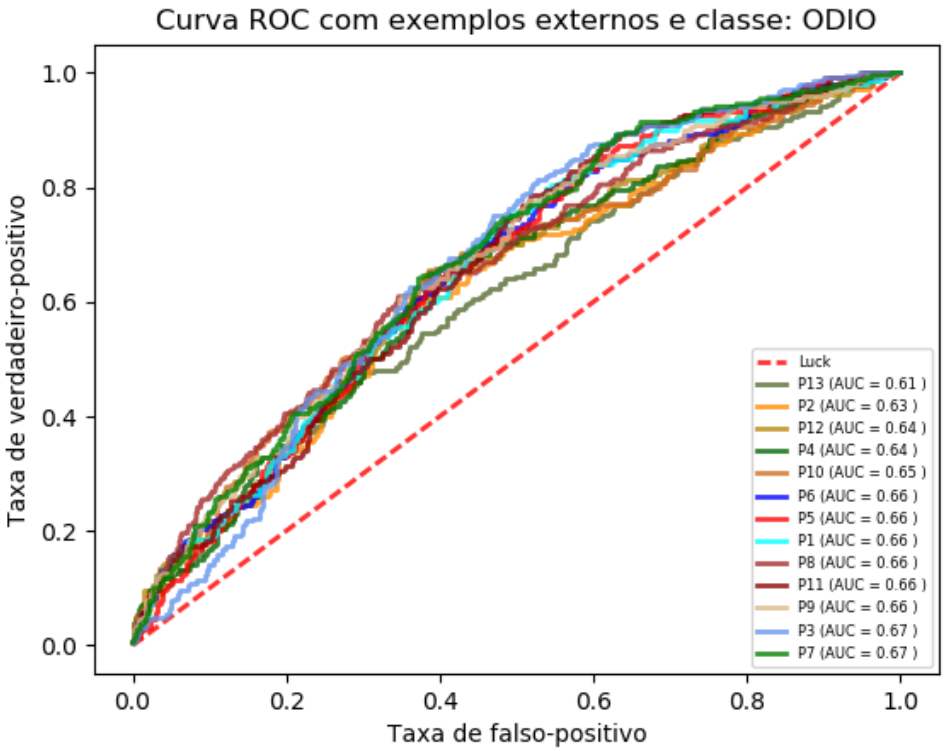


Figura 17 – Curva ROC(Receiver operating characteristic) para classe ódio

Tabela 15 – Analise dos Gráficos ROC classe ódio

StratifiedKFold		**Externos**	
NB	SVM	NB	SVM
$P3_{=}$	$P4_{+}$	$P3_{+}$	$P4_{+}$
$P5_{+}$	$P6_{+}$	$P5_{=}$	$P6_{+}$
$P7_{-}$	$P8_{-}$	$P7_{+}$	$P8_{+}$
$P9_{=}$	$P10_{=}$	$P9_{=}$	$P10_{+}$
$P11_{+}$	$P12_{+}$	$P11_{=}$	$P12_{+}$
não	$P13_{=}$	não	$P13_{-}$

A Tabela 15 é preenchida através dos valores dos gráficos 16 e 17.

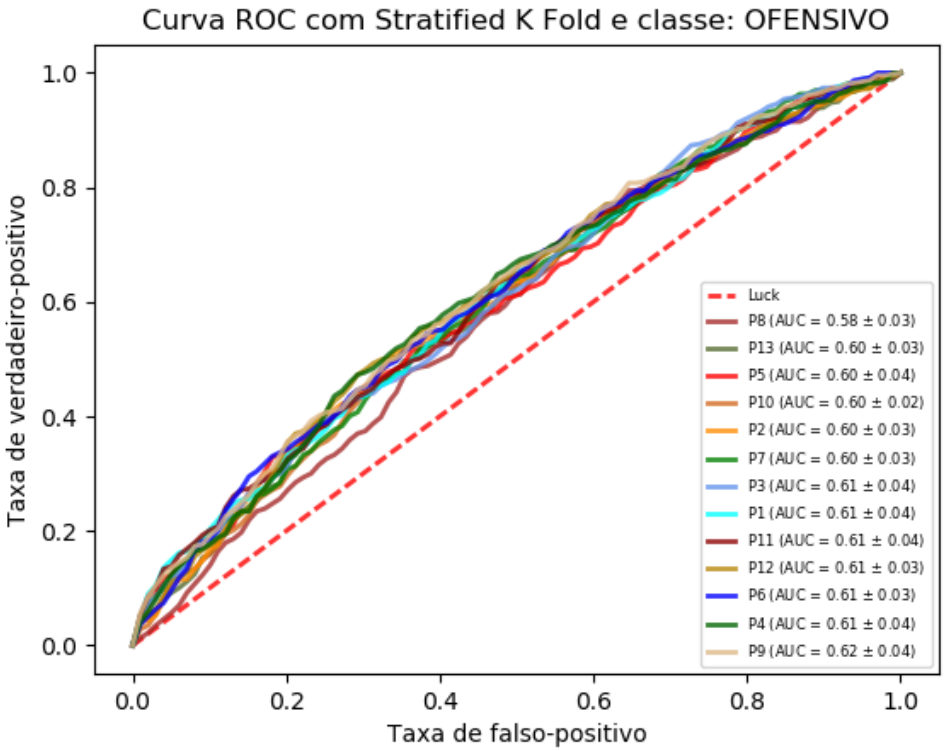


Figura 18 – Curva ROC(Receiver operating characteristic) para classe ódio

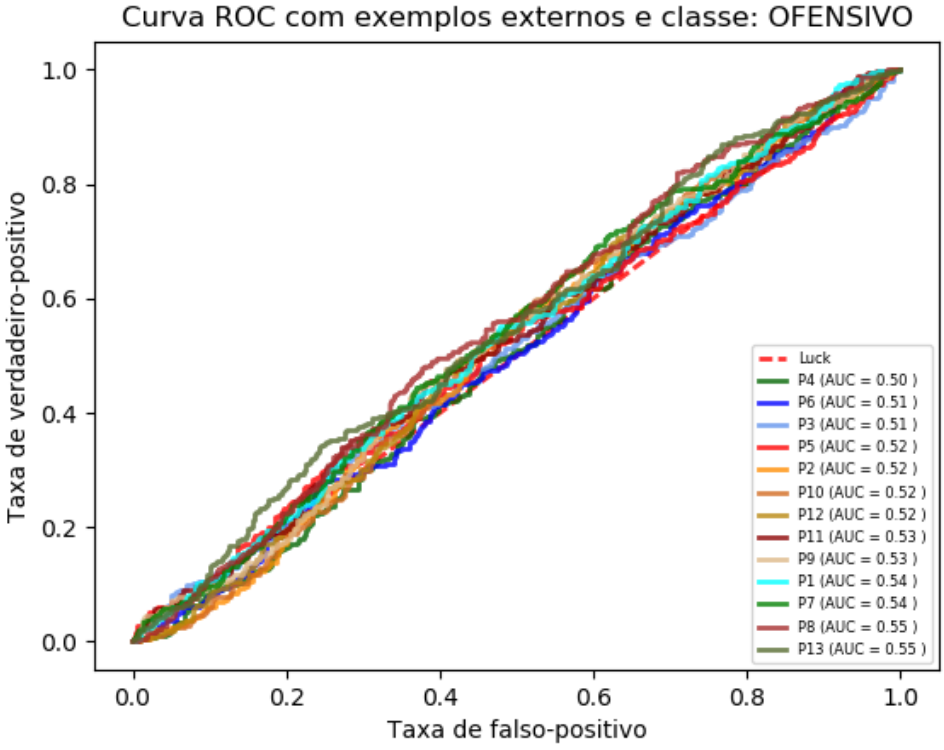


Figura 19 – Curva ROC(Receiver operating characteristic) para classe ofensivo

Tabela 16 – Analise dos Gráficos ROC classe ofensivo

StratifiedKFold		**Externos**	
NB	SVM	NB	SVM
$P3_{=}$	$P4_{+}$	$P3_{-}$	$P4_{-}$
$P5_{-}$	$P6_{+}$	$P5_{-}$	$P6_{-}$
$P7_{-}$	$P8_{-}$	$P7_{=}$	$P8_{+}$
$P9_{+}$	$P10_{=}$	$P9_{-}$	$P10_{=}$
$P11_{=}$	$P12_{+}$	$P11_{-}$	$P12_{=}$
não	$P13_{=}$	não	$P13_{+}$

A tabela 16 é preenchida através dos valores dos gráficos 18 e 19.

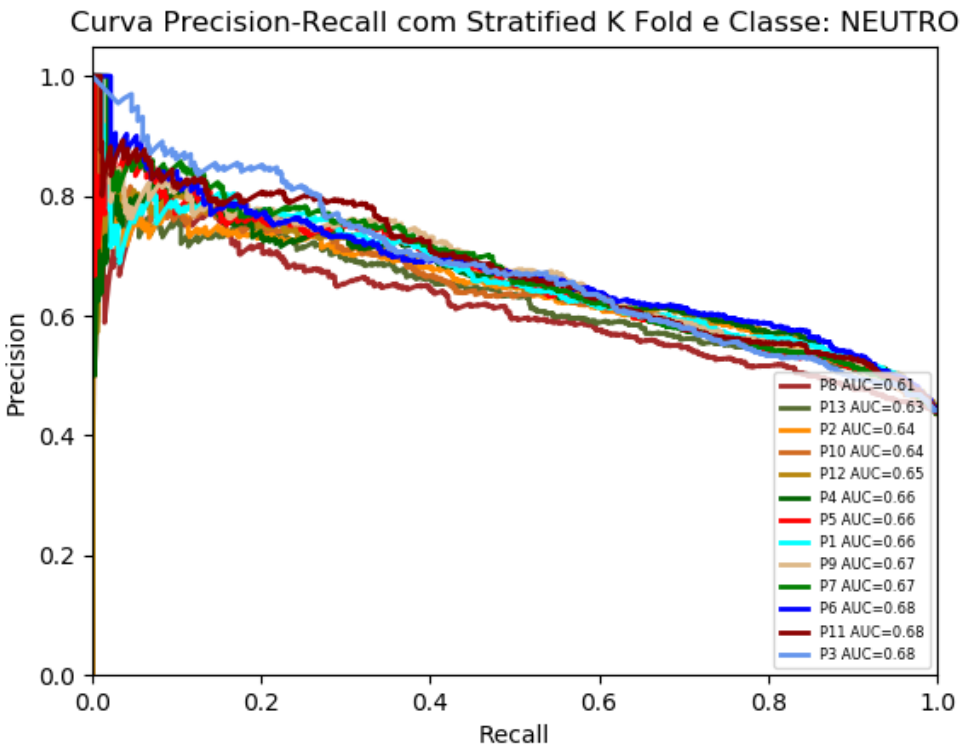


Figura 20 – Curva PR(Precision Recall) para classe neutro

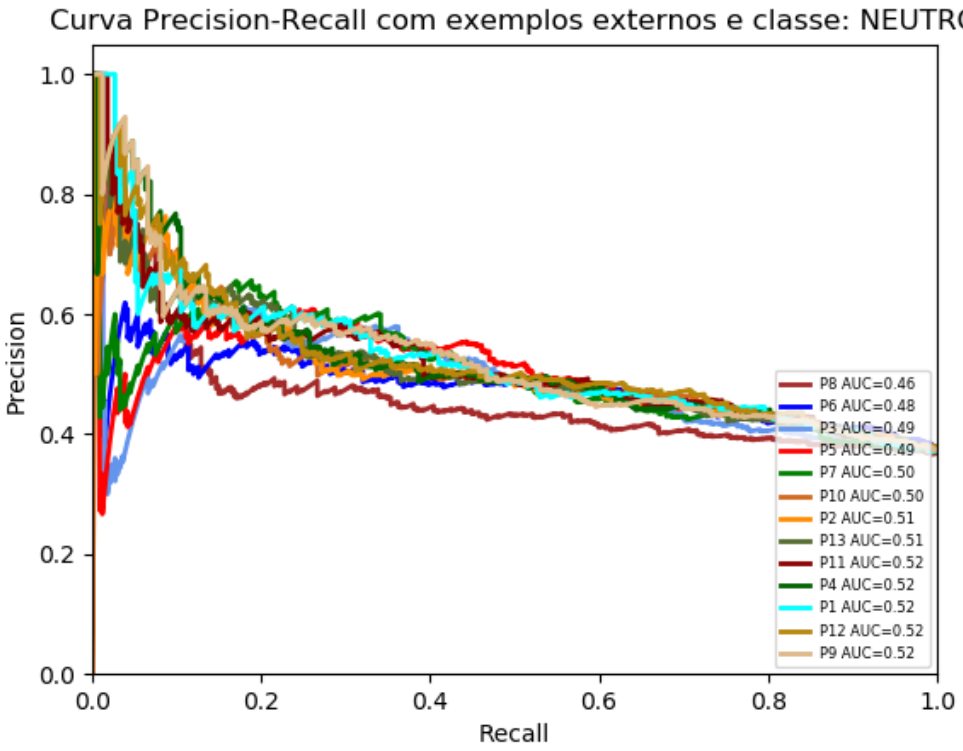


Figura 21 – Curva PR(Precision Recall) para classe neutro

Tabela 17 – Analise dos Gráficos PR classe neutro

StratifiedKFold		**Externos**	
NB	SVM	NB	SVM
$P3_+$	$P4_+$	$P3_-$	$P4_+$
$P5_=$	$P6_+$	$P5_-$	$P6_-$
$P7_+$	$P8_-$	$P7_-$	$P8_-$
$P9_+$	$P10_=$	$P9_=$	$P10_-$
$P11_+$	$P12_+$	$P11_=$	$P12_+$
não	$P13_-$	não	$P13_=$

A tabela 17 é preenchida através dos valores dos gráficos 20 e 21.

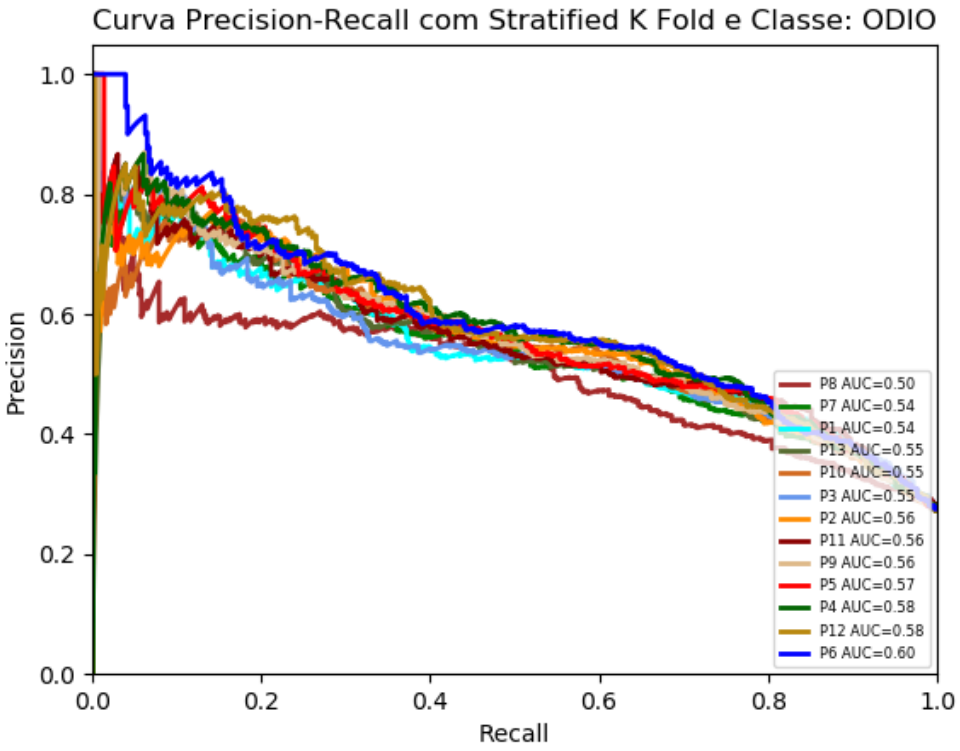


Figura 22 – Curva PR(Precision Recall) para classe ódio

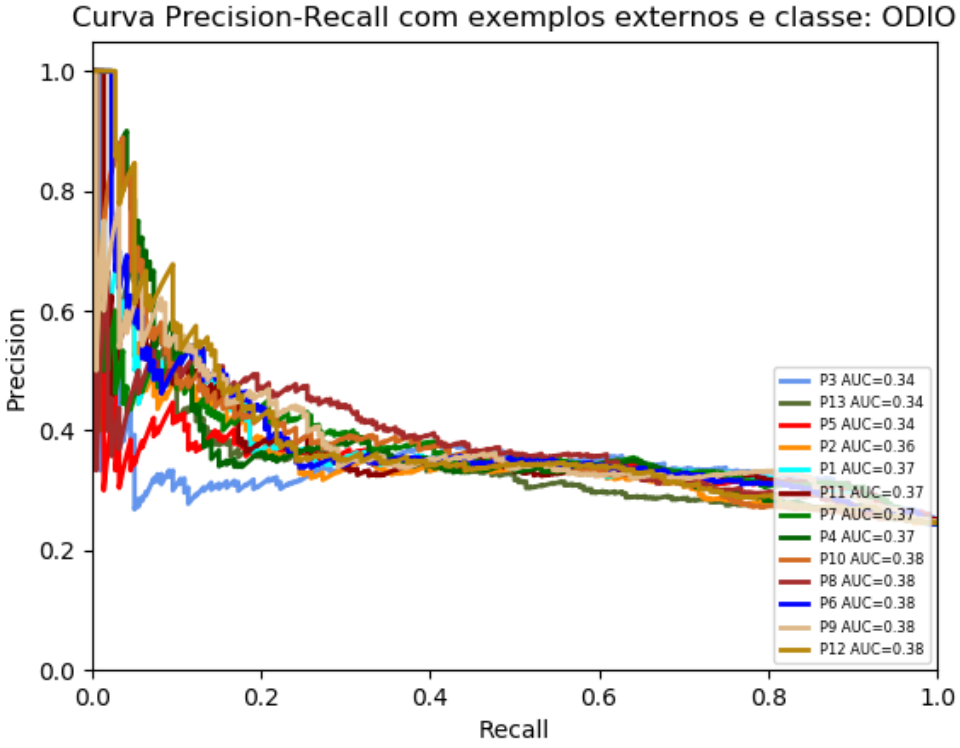


Figura 23 – Curva PR(Precision Recall) para classe ódio

Tabela 18 – Analise dos Gráficos PR classe ódio

StratifiedKFold		**Externos**	
NB	SVM	NB	SVM
$P3_+$	$P4_+$	$P3_-$	$P4_+$
$P5_+$	$P6_+$	$P5_-$	$P6_+$
$P7_=$	$P8_-$	$P7_=$	$P8_+$
$P9_+$	$P10_-$	$P9_+$	$P10_+$
$P11_+$	$P12_+$	$P11_=$	$P12_+$
não	$P13_-$	não	$P13_-$

A tabela 18 é preenchida através dos valores dos gráficos 22 e 23.

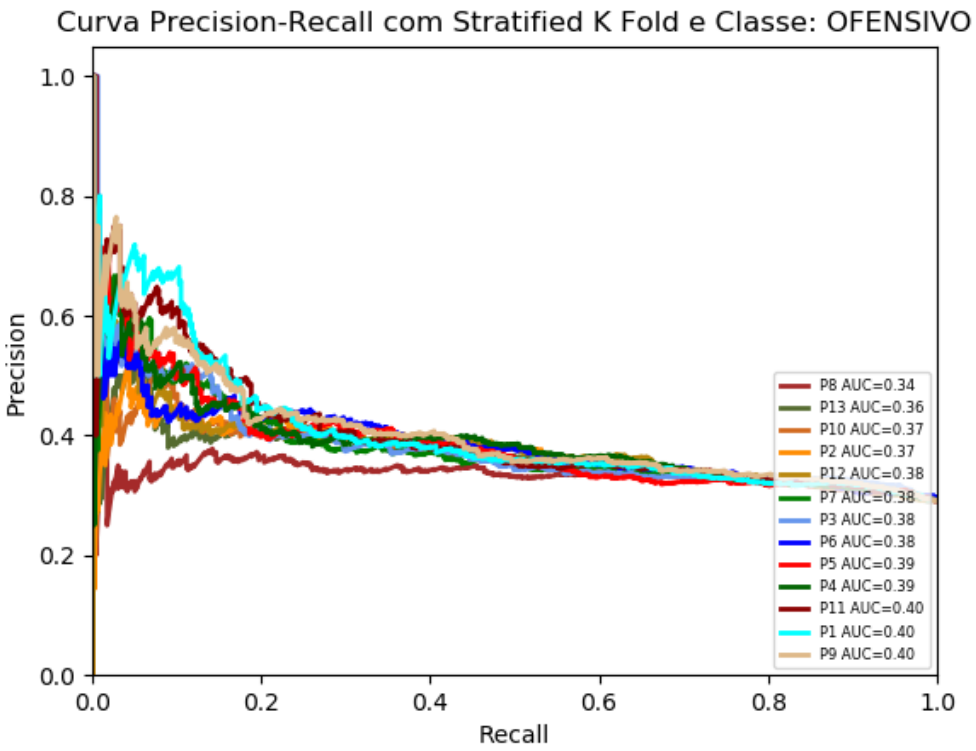


Figura 24 – Curva PR(Precision Recall) para classe ódio

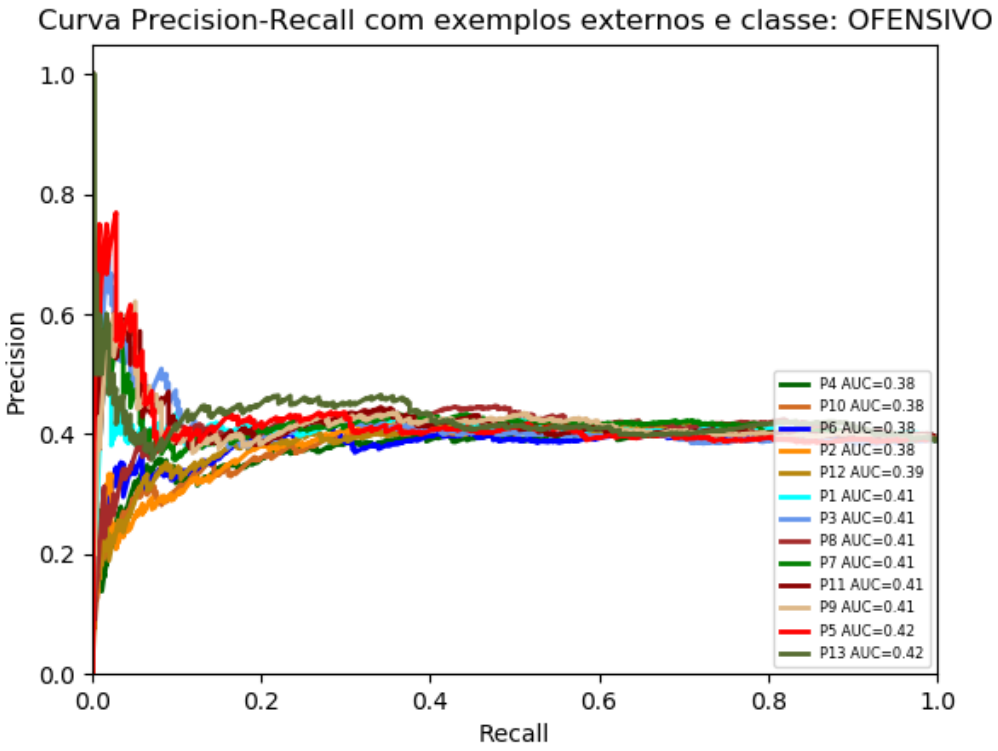


Figura 25 – Curva PR(Precision Recall) para classe ofensivo

Tabela 19 – Analise dos Gráficos PR classe ofensivo

StratifiedKFold		**Externos**	
NB	SVM	NB	SVM
$P3_{-}$	$P4_{+}$	$P3_{=}$	$P4_{=}$
$P5_{-}$	$P6_{+}$	$P5_{+}$	$P6_{=}$
$P7_{-}$	$P8_{-}$	$P7_{=}$	$P8_{+}$
$P9_{=}$	$P10_{=}$	$P9_{=}$	$P10_{=}$
$P11_{=}$	$P12_{+}$	$P11_{=}$	$P12_{+}$
não	$P13_{-}$	não	$P13_{+}$

A tabela 19 é preenchida através dos valores dos gráficos 24 e 25.

Tabela 20 – Resolução das analises dos gráficos ROC e PR

	NB			**SVM**			**Total**		
Técnica	PX_-	PX_+	$PX_=$	PX_-	PX_+	$PX_=$	PX_-	PX_+	$PX_=$
BI-GRAM	6	3	3	1	9	2	7	12	5
TF-IDF	5	3	4	2	8	2	7	11	6
SELECTK	5	2	5	8	4	0	13	6	5
FROM MODEL	2	6	4	2	3	7	4	9	11
BAGGING	1	3	8	0	11	1	1	14	9
LSA	não	não	não	6	2	4	6	2	4

Na tabela 20 podemos ver a frequência de vezes que um determinado símbolo apareceu, envolvendo uma técnica e um algoritmo de aprendizado de máquina. Nota-se que as técnicas que obtiveram mais indicações de melhorias foram BIGRAM, TF-IDF, SELECT FROM MODEL e BAGGING. O Algoritmo de Aprendizado de Máquina que possui a maior quantidade de PX_+ relacionado a essas técnicas foi o SVM. Essas técnicas e esse algoritmo compõem o pipeline utilizado neste trabalho, que será chamado de pipeline Resultante, os resultados desse pipeline podem ser vistos nas Figuras 26 e 27. A fim de demonstrarmos a eficácia do modelo gerado por este pipeline, vemos alguns exemplos de comentários corretamente classificados como discurso de ódio na Tabela 21 e incorretamente classificados como discurso de ódio na Tabela 22. Nota-se que os classificados incorretamente possuem a probabilidade $P(odio)$ pequena.

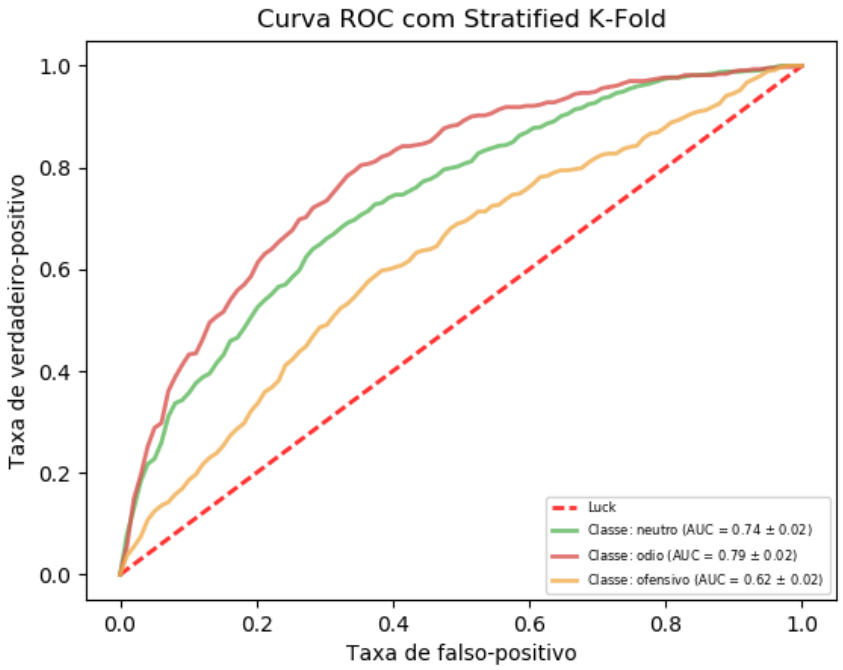


Figura 26 – Curva ROC(Receiver operating characteristic) do pipeline Resultante para as classes neutro, ódio e ofensivo.

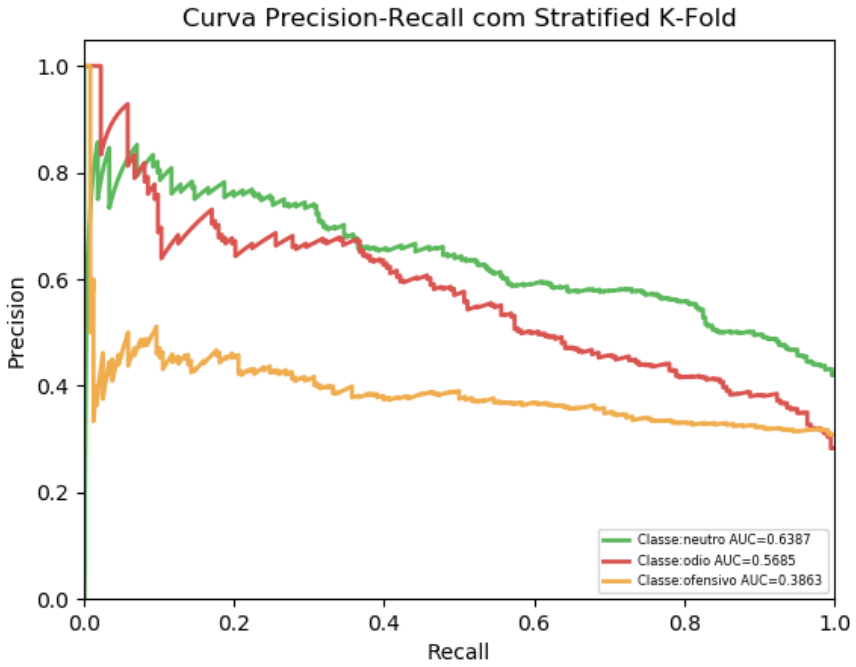


Figura 27 – Curva PR(Precision Recall) do pipeline Resultante para as classes neutro, ódio e ofensivo.

Tabela 21 – Exemplos de textos classificados corretamente como discurso de ódio

P(neutro)	P(odio)	P(ofensivo)	Título	Comentário
6%	81%	13%	Deputado aciona CNJ contra decisão de juiz federal do DF que autoriza 'cura gay'	Aberrações tem que ser tratadas.
25%	61%	14%	'Cura gay': juiz, OAB e advogado explicam resolução e liminar que libera tratamento	Ga'ys são guias cegos de satanás, duvida? Olhe as ações do diabo e vejam as ações dos g a y s.
3%	94%	3%	SP terá casamento coletivo gay; inscrições vão até 11 de outubro	Se deus quiser, todo o lugar vai explodir, e boa parte dessa rassa vai pro kintu dus inferno!
11%	71%	18%	'Fiquei em choque', diz Miss Amapá Gay após ter coroa arrancada por concorrente	Tinha que exterminar essas pragas

Tabela 22 – Exemplos de textos classificados incorretamente como discurso de ódio

P(neutro)	P(odio)	P(ofensivo)	Título	Comentário
24%	48%	28%	'Fiquei em choque', diz Miss Amapá Gay após ter coroa arrancada por concorrente	Plagiando um acontecimento, para aparecer. A que ponto chega, a mediocridade!
14%	55%	31%	Após vida marcada por preconceito, travesti negra conquista título de doutora na UFPR	Você escolhe ser policial...
37%	48%	15%	Grupos protestam contra liminar da 'cura gay' em mobilização no Centro de Teresina	Comentários parecem estar proibidos. Se sua consciência lhe acusa, lembre-se ser Deus maior que sua consciência
33%	40%	37%	Transexual recifense é a primeira do Norte e Nordeste a poder usar nome social na carteira da OAB	Melhor comentário.

Com o pipeline escolhido em mãos podemos realizar alguns testes de validação, como o Stratified K fold cross-validation, visto na Tabela 23 Nessa tabela também o possível ver o resultado do treinamento do pipeline a partir da união de todas as bases. Porém é preciso observar que os dados da base B3 não possuem o mesmo grau de confiança que a base B1 e B2.

Tabela 23 – Teste Stratified K fold Cross-Validation usando o pipeline Resultante

	Neutro			Odio			Ofensivo			Média		
	precision	recall	F1	precision	recall	F1	precision	recall	F1	precision	recall	F1
B1 U B2	0.608	0.642	0.626	0.58	0.598	0.59	0.406	0.362	0.382	0.54	0.55	0.544
B1 U B2 U B3	0.618	0.656	0.634	0.56	0.554	0.554	0.442	0.41	0.424	0.544	0.552	0.546

4.2.1 Outros Experimentos

Agrupamento de palavras em comentários de ódio

A fim de verificar como as palavras pertencentes aos comentários se comportam em relação à similaridade foi utilizada a técnica k-means na união das bases de classificação B1 e B2. Esse algoritmo constrói K clusters com textos semelhantes entre si, neste trabalho utilizamos k = 50 a fim de formar pequenos clusters para descobrirmos padrões de comentários. Esses clusters foram salvos no mongoDB junto a suas classificações. Com isso foi possível determinar quais clusters possuiam mais comentários anotados como discurso de ódio e verificar o padrão existente neles. Para cada cluster o k-means seleciona palavras chaves pertencentes aos textos contidos no cluster. Ao extrairmos essas palavras conseguimos notar alguns padrões de comentários, como visto na tabela 24.

Tabela 24 – Palavras chaves de clusters formados por maioria de comentários classificados como ódio

cluster	palavras chaves
14	homem,mulher,nao,ser,deus,existe,vai,criou
16	aberracoes,conta,dessas,criou,familias,correndo,risco,acabar,deus,festa
17	doenca,nao,cura,ser,homossexualismo,so,eh,assim,realmente
20	triste,vice,deveria,nao,versa,mudar,igreja,homem
22	aberracao,homossexualismo,ser,sus,natureza,infernos,escora
32	ver,gays,bater,dois,rua,beijando,nao,frente,pra,quero
38	queimar ,inferno,vao,marmore,vai,fogo,aberracoes,deus,rosca,nao

A Tabela 25 mostra o quanto os resultados são alterados quando utilizamos o título da notícia junto ao comentário. Essa tabela foi gerada através do método chamado classification report implementado pelo Scikit Learn que pertence ao pacote metrics, dado a lista de classes reais e a lista de classes preditas pelo modelo ele retorna as métricas precision, recall e f1 score. Nessa implementação eles calculam o f1 usando o score F-beta. Como entrada desse método utilizamos os exemplos externos a base de treinamento e o pipeline desenvolvido para o trabalho.

Tabela 25 – Comparação entre as classificações usando Stratified K-Fold, utilizando o pipeline Resultante, treinado com e sem o titulo da notícia agregado ao comentário

	Neutro			Odio			Ofensivo			Média		
	precision	recall	F1	precision	recall	F1	precision	recall	F1	precision	recall	F1
C/titulo	0.608	0.642	0.626	0.58	0.598	0.59	0.406	0.362	0.382	0.54	0.55	0.544
S/Titulo	0.604	0.656	0.628	0.462	0.476	0.466	0.414	0.34	0.374	0.506	0.516	0.508

4.3 Discussão

"A capacidade humana de avaliar corretamente a subjetividade de um texto varia de 72% a 85% ". (TELES; SANTOS; SOUZA, 2016 apud GOLDEN, 2011) (TELES; SANTOS; SOUZA, 2016 apud WIEBE; WILSON; CARDIE, 2005)

As métricas de avaliação da qualidade do modelo de Aprendizado de Máquina obtidas no processo de validação cruzada, com exemplos de validação oriundos da mesma base de dados, apresentaram um resultado razoável, o que prova que as classificações e os exemplos têm um bom grau de concordância entre si. Porém, quando a base de testes era de exemplos adicionais os resultados apresentam uma qualidade abaixo da esperada. Um dos problemas encontrados no processo foram os títulos das notícias, que por aparecerem em mais de uma classificação acabaram ganhando um peso maior na classificação dos comentários. Um comentário com uma palavra apenas, por exemplo "ola"em uma noticia com titulo "jovens gays foram assassinados na saída de baile funk"acabam sendo classificados como discurso de ódio dado o título.

Também nota-se uma perda pequena nos resultados como visto na tabela 25. Por outro lado, o título dá um contexto ao discurso de ódio, deixando mais fácil a sua detecção. Sem ele a maioria dos comentários sem uma citação direta aos alvos de discurso de ódio, não seriam reconhecidos. Um comentário "vocês todos tem que morre"retirado de uma notícia sobra a parada gay não poderia ser classificado como discurso de ódio apenas pelo texto do comentário, pois não é visível o grupo a quem ele se direciona.

Outro problema que pode ter contribuído para os resultados piores com os testes pode ter sido a amplitude do tema discurso de ódio, pois mesmo especificando discurso de ódio voltado ao público LGBT, o contexto em que o discurso de ódio pode acontecer são muitos, inclusive o uso de sarcasmo nesses casos é muito alto. O classificador probabilístico traz consigo o problema evidenciado por Violet Blue (BLUE, 2017) sobre o projeto semelhante do Google, que acaba classificando frases com contextos positivos ou neutros como frases tóxicas, dado que os termos que referenciam estes alvos ganham um peso maior na decisão do classificador, o que acaba inferiorizando mais ainda o público que tentamos ajudar.

O processo de anotação pela interface web, onde as anotações eram obtidas através de vários usuários diferentes se mostrou válida. A escolha da classe de comentário utilizava o conjunto de anotações que o mesmo recebeu da interface e escolhia a mais frequente, esse recurso permitiu utilizar o conhecimento de vários usuários para tomar a melhor decisão sobre a classe dos comentários. Dados obtidos dessa forma são mais confiáveis.

O pipeline resultante apresentou bons resultados em comparação aos outros pipelines da Tabela 13, pois utiliza as técnicas que apresentaram nos testes as maiores frequências de PX_+.

5 Conclusão

No âmbito do Brasil hoje, a sociedade encontra-se cada vez mais dividida, por ideais políticos, cultivados principalmente através da internet. O ódio acaba se tornando uma arma política, uma ferramenta para promover a si mesmo e adquirir adeptos de seus ideais. Assim, o combate ao discurso de ódio se torna cada vez mais importante.

Inicialmente o foco do trabalho era mais amplo, foi discutida a possibilidade de detecção de discurso de ódio em mídias sociais. Após estudar os aspectos dessas mídias, verificamos quais desses aspectos poderiam aumentar as chances de obter o discurso de ódio e ficou claro a necessidade de um tipo de mídia que possua um grau maior de anonimato, onde seus usuários tem liberdade maior para se expressar sem o julgamento de pessoas que o conhecem. Outro aspecto importante é a quantidade de informação que essa mídia disponibiliza, por exemplo os textos curtos extraídos(tweets) do twitter[3] geralmente não possuem informação suficiente para caracterização de um discurso de ódio. Os tweets realizados no Twitter[3] possuem um limite de quantidade de palavras que podem ser utilizadas. Por isso muitos dos usuários omitem informações que identifiquem os alvos do discurso de ódio para economizar palavras. Levando em conta esses dois aspectos foi decidido que o foco desse trabalho deveria ser direcionado aos jornais online. Outro fator que sustenta a escolha pelos jornais online foi o poder de difusão de informações e de formar opiniões que os mesmos possuem, fatores que intensificam o poder nocivo do discurso de ódio. A detecção do discurso de ódio voltado a todos os grupos marginalizados pela sociedade se tornou inviável pela quantidade de termos pertencentes a este tema que é muito amplo e está sempre em evolução. Levando isso em consideração, foi preciso escolher um grupo ao qual focar, no caso desse trabalho o grupo escolhido foi a comunidade LGBT. O foco em relação ao público LGBT serviu tanto para melhorar a classificação fazendo um classificador mais específico, quanto para trazer um alerta ao ódio crescente que a eles se direciona hoje, onde grupos supremacistas espalham notícias falsas sobre tal comunidade.

Os resultados mostraram que o sistema desenvolvido foi capaz de acertar em média apenas 55% dos comentários da base de comentários externa a utilizada no treinamento. É possível que utilizando o mesmo método com mais exemplos, cuja a qualidade seja determinada por uma equipe de especialistas em diversas áreas como psicologia e linguística, tenha grandes chances de melhorar os resultados e criar uma ferramenta de detecção mais eficiente e abrangente do que a atual. A precisão para a classe ódio ficou em torno de 60% um valor bom, mas que ainda oferece um risco de classificar erroneamente um comentário como discurso de ódio, para esta ferramenta este seria o pior caso, por exemplo em um debate o usuário afirma que tem orgulho de ser gay e a ferramenta remove-se o seu comentário achando que era um discurso de ódio, esse usuário se sentiria humilhado e

oprimido. Dado essas limitações a ferramenta de detecção deve ser utilizada como auxílio a um moderador humano, que teria a palavra final na decisão de remover ou não aquele conteúdo. Esse auxílio seria dado da forma de um filtro, onde milhares de comentários são reduzidos a poucos casos, que possuem uma chance maior de conter o discurso de ódio. E posteriormente esses casos filtrados devem ser analisados por seres humanos que decidem se ele será removido ou não. As métricas precision e recall, ficaram bem equilibradas no modelo criado.

5.1 Limitações e Ameaças à Validade dos Resultados

Mesmo com 30 voluntários a classificação de 1597 comentários demorou para ser realizada. Muitos dos voluntários desistiram no meio do caminho por falta de motivação e tempo. Dentre os 1597 só 456 possuem um bom grau de confiança pois foram classificados por 3 ou mais usuários. Os comentários que tiveram uma única anotação possuem uma chance maior de terem sido anotados com a classe errada. Algumas técnicas que demandam mais desempenho do computador não puderam ser testadas corretamente, devido ao maquinário disponível para esse estudo. Por exemplo o método de LDA(latent dirichlet allocation) demorava horas para emitir o resultado. Algumas técnicas que utilizam redes neurais não iniciavam no computador.

5.2 Trabalhos Futuros

Um dos maiores problemas enfrentados nesse trabalho foi a anotação dos dados. Visto isto, como trabalhos futuros, alguns meios de resolver este problema devem ser estudados como, promover uma anotação em larga escala, através de campanhas de anotação realizadas por Organizações Não Governamentais (ONGs) de prestigio com o público LGBT(Lésbicas, Gays, Bissexuais, Travestis, Transexuais e Transgêneros), com isso conseguiríamos um número maior de pessoas engajadas. Também é possível obter dados de Organizações Não Governamentais (ONGs) que trabalham recebendo denúncias de crimes cibernéticos. Outra alternativa é uso do aprendizado semi-supervisionado, onde utilizaremos o próprio modelo gerado para anotar dados sem anotação, ou com anotação de um único usuário, com isso podemos realimentar a base de dados e treinar outro modelo. Podemos utilizar também a técnica de Active Learning, na qual o modelo escolhe os dentre diversos dados não anotados, aqueles que ele julgue ter uma importância maior para realizar uma determinada tarefa e pede ajuda de um ser humano para anota-lo.

Outra alternativa é aproveitar os dados, dos diversos estudos estudos relacionados a detecção de discurso de ódio, geralmente feitos em inglês, como a base de dados de termos de ódio[19]. Para fazer uso dos exemplos em inglês, podemos treinar um modelo com

esses exemplos e transferir o que foi aprendido por ele para a linguá portuguesa, usando o método de Transfer Learning.

Com mais dados é possível utilizar técnicas mais avançadas como Deep Learning que requerem uma quantidade muito maior de exemplos do que o que tínhamos disponível.

5.3 Considerações Finais

Este trabalho é de grande importância para sociedade, pois esse tipo de iniciativa promove o bem estar social dos grupos aos quais o discurso de ódio é direcionado. Pois ele ajuda a encontrar esse tipo de conteúdo nocivo, impedindo que o mesmo se propague pela internet, afetando um número muito maior de pessoas. Espera-se que este trabalho sirva de incentivo à criação de tecnologias voltadas a ajudar grupos ainda discriminados pela sociedade, a fim de promover igualdade social. Os resultados obtidos a partir deste trabalho podem servir de fomento para outras pesquisas computacionais com foco em causas sociais.

Referências

ABU-MOSTAFA, Y. S.; MAGDON-ISMAIL, M.; LIN, H.-T. *Learning From Data*. [S.l.]: amlbook, 2012. Citado na página 20.

AFONSO, A. *O que é Spring Boot?* 2012. Http://blog.algaworks.com/spring-boot/. Online; acessado dia 27 de Novembro de 2017. Citado na página 42.

ALTERMANN, D. *Qual a diferença entre redes sociais e mídias sociais?* 2010. Http://www.midiatismo.com.br/qual-a-diferenca-entre-redes-sociais-e-midias-sociais. Online; acessado dia 20 de Novembro de 2017. Citado na página 7.

BARONI, D. et al. O gênero textual notícia: do jornal impresso ao on-line. In: *9º Encontro Nacional de História da Mídia*. Faculdade Anhanguera de Taubaté / São Paulo: [s.n.], 2013. p. 01–11. Citado na página 6.

BENEVENUTO, F.; RIBEIRO, F.; ARAÚJO, M. *Métodos para Análise de Sentimentos em mídias sociais*. 2015. Http://homepages.dcc.ufmg.br/ fabricio/download/webmedia-short-course.pdf. Online; acessado dia 21 de Novembro de 2017; Minicurso. Citado na página 21.

BERRY, M. J. A.; LINOFF, G. *Data mining techniques: for marketing, sales and customer support*. USA: Wiley Computer Publishing, 1997. Citado na página 15.

BHAGWANT. *Latent Semantic Analysis (LSA) Tutorial*. 2011. Https://technowiki.wordpress.com/2011/08/27/latent-semantic-analysis-lsa-tutorial/. Online; acessado dia 22 de Novembro de 2017. Citado na página 24.

BIRD, S.; KLEIN, E.; LOPER, E. *Natural Language Processing with Python*. [S.l.]: O'Reilly Media, 2009. Citado 2 vezes nas páginas 22 e 44.

BLUE, V. *Google's comment-ranking system will be a hit with the alt-right*. 2017. Https://www.engadget.com/2017/09/01/google-perspective-comment-ranking-system/. Online;Acessado no dia 14 de Novembro de 2017. Citado 2 vezes nas páginas 34 e 65.

BRASIL. *Relatório de Violência Homofóbica no Brasil: ano 2013*. 2016. Http://www.mdh.gov.br/assuntos/lgbt/dados-estatisticos/Relatorio2013.pdf. Online; acessado no dia 16 de Novembro de 2017, Secretaria Especial de Direitos Humanos do Ministério das Mulheres, da Igualdade Racial e dos Direitos Humanos. Citado na página 15.

BRETAS, V. *Os números que resumem os 3 anos de Operação Lava Jato*. 2017. Https://exame.abril.com.br/brasil/os-numeros-que-resumem-os-3-anos-de-operacao-lava-jato/. Online; acessado no dia 18 de Novembro de 2017. Citado na página 1.

BROWNLEE, J. *Basic Concepts in Machine Learning*. 2015. Https://machinelearningmastery.com/basic-concepts-in-machine-learning/. Online; acessado no dia 14 de Novembro de 2017. Citado 2 vezes nas páginas viii e 18.

BROWNLEE, J. *How to Build an Ensemble Of Machine Learning Algorithms in R (ready to use boosting, bagging and stacking)*. 2016. Https://machinelearningmastery.com/machine-learning-ensembles-with-r/. Online; acessado dia 15 de Novembro de 2017. Citado na página 27.

BROWNLEE, J. *A Gentle Introduction to the Bag-of-Words Model*. 2017. Https://machinelearningmastery.com/gentle-introduction-bag-words-model/. Online; acessado dia 04 de dezembro de 2017. Citado na página 21.

BRUGGER, W. Proibição ou proteção do discurso do Ódio? algumas observações sobre o direito alemão e o americano. *Trad. Maria Angela Jardim de Santa Cruz Oliveira. Revista de Direito*, 2007. Citado 2 vezes nas páginas 2 e 11.

BURNAP, P.; WILLIAMS, M. L. Hate speech, machine classification and statistical modelling of information flows on twitter: Interpretation and communication for policy decision making. *Cardiff School of Computer Science & Informatics*, 2014. Citado na página 35.

CAMILO, C. O.; SILVA, J. C. da. *Mineração de Dados: Conceitos, Tarefas, Métodos e Ferramentas*. [S.l.], 2009. Citado 4 vezes nas páginas 4, 15, 19 e 33.

COHN, J.; KUNTZ, A.; BIRNBAUM, L. Attitudebuzz: Using social media data to localize complex attitudes. In: *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. [S.l.: s.n.], 2015. p. 1569–1570. Citado na página 34.

COMPANY, T. N. Y. T. *The Times is Partnering with Jigsaw to Expand Comment Capabilities*. 2016. Https://www.nytco.com/the-times-is-partnering-with-jigsaw-to-expand-comment-capabilities/. Online; acessado dia de Novembro de 2017. Citado na página 12.

CONTI, F. *Biometria Qui Quadrado*. 2009. Http://www.ufpa.br/dicas/biome/biopdf/bioqui.pdf. Online; acessado dia 25 de Novembro de 2017. Citado na página 24.

CORRÊA, M. *Turbulência política deve reduzir crescimento no ano que vem*. 2017. Https://oglobo.globo.com/economia/turbulencia-politica-deve-reduzir-crescimento-no-ano-que-vem-21372753. Online; acessado dia 18 de Novembro de 2017. Citado na página 1.

DAVIDSON, T. et al. Automated hate speech detection and the problem of offensive language. In: *Proceedings of the Eleventh International AAAI Conference on Web and Social Media (ICWSM 2017)*. [S.l.: s.n.], 2017. p. 512–515. Citado na página 3.

DEGENNE, A.; FORSE, M. *Introducing Social Networks*. [S.l.]: SAGE Publications Ltd, 1999. Citado na página 8.

DIETTERICH, T. G. Ensemble methods in machine learning. In: *First International Workshop on Multiple Classifier Systems*. [S.l.: s.n.], 2000. Citado na página 27.

ELLINOR, A. et al. *Bayes Theorem and Conditional Probability*. 2017. Https://brilliant.org/wiki/bayes-theorem/. Online; acessado dia 25 de Novembro de 2017. Citado na página 25.

ERICSON, G.; OLPROD; OPENLOCALIZATIONSERVICE. *Como escolher algoritmos de Aprendizado de Máquina do Microsoft Azure*. 2017. Https://docs.microsoft.com/pt-br/azure/machine-learning/studio/algorithm-choice. Online; acessado dia 14 de Novembro de 2017. Citado 2 vezes nas páginas 19 e 20.

FAVA, G. P.; JÚNIOR, C. P. Filtros bolha nos algoritmos do facebook: Um estudo de caso nas e leições para reitoria da ufjf. In: *XXXVII Congresso Brasileiro de Ciências da Comunicação*. Universidade Federal de Juiz de Fora , Juiz de Fora, MG: [s.n.], 2014. Citado na página 10.

FLACH, P.; HERNÁNDEZ-ORALLO, J.; FERRI, C. A coherent interpretation of AUC as a measure of aggregated classification performance. In: *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. [S.l.: s.n.], 2011. p. 657–664. Citado na página 32.

GEITGEY, A. *Machine Learning is Fun!* 2014. Https://medium.com/@ageitgey/machine-learning-is-fun-80ea3ec3c471. Online; acessado dia 13-Novembro-2017. Citado na página 18.

GNET. *Curso de Introdução ao Direito de Internet*. 2016. Http://irisbh.com.br/wp-content/uploads/2017/07/Curso-de-IntroduGrupo de Estudos Internacionais em Propriedade Intelectual, Internet e Inovação - GNet Online; acessado dia 13-Novembro-2017. Citado na página 14.

GOLDEN, P. *Write here, write now*. 2011. Https://www.research-live.com/article/features/write-here-write-now/id/4005303. Online; acessado dia 15 de Novembro de 2017. Citado na página 64.

GORSKI, I. *Caracteristicas WEB 2.0: Afinal, o que é Web 2.0 ?* 2009. Https://ivangorski.wordpress.com/2009/01/15/caracteristicas-web-20-afinal-o-que-e-web-20/. Online; acessado no dia 17 de Novembro de 2017. Citado na página 7.

HAN, J.; KAMBER, M. Classification and prediction. In: *Data Mining: Concepts and Techniques*. second. [S.l.]: CA: Morgan Kaufmann, 2006. cap. 6, p. 311–313. Citado 2 vezes nas páginas 4 e 25.

HAND, D. J. Measuring classifier performance: A coherent alternative to the area under the ROC curve. *Machine Learning*, v. 77, p. 103–123, jun 2009. Citado na página 30.

HIGGINBOTHAM, S. *Inside Facebook's Biggest Artificial Intelligence Project Ever*. 2016. Http://fortune.com/facebook-machine-learning/. Online; acessado dia 13-Novembro-2017. Citado na página 19.

JOACHIMS, T. Text categorization with support vector machines: Learning with many relevant features. In: *European Conference on Machine Learning*. [S.l.: s.n.], 1998. p. 137–142. Citado na página 25.

KAPLAN, A. M.; HAENLEIN, M. Users of the world, unite! the challenges and opportunities of social media. *Business Horizons*, v. 53, n. 1, p. 58–68, 2010. Citado na página 7.

KARNAL, L. *O mundo dos PETRALHAS e COXINHAS*. 2017. Https://www.youtube.com/watch?v=lMHuITTKPQ4. Online; acessado dia 18 de Novembro de 2017. Citado na página 1.

KEMP, S. *Digital in 2017: Global Overview*. 2017. Https://wearesocial.com/special-reports/digital-in-2017-global-overview. Online; acessado dia 19 de Novembro de 2017. Citado 2 vezes nas páginas 1 e 5.

KHAN, A. et al. A review of machine learning algorithms for text-documents classification. *JOURNAL OF ADVANCES IN INFORMATION TECHNOLOGY*, v. 1, n. 1, p. 4–20, feb 2010. Citado na página 24.

KRAMERA, A. D. I.; GUILLORYB, J. E.; HANCOCK, J. T. Experimental evidence of massive-scale emotional contagion through social networks. *Proceedings of the National Academy of Sciences of the United States of America*, 2014. Citado na página 10.

LAROSE, D. T. *Discovering Knowledge in Data: An Introduction to Data Mining* . [S.l.]: John Wiley and Sons, Inc, 2005. Citado 2 vezes nas páginas viii e 16.

LEITE, F.; BATISTA, L. L.; SOUZA, J. T. de. Contribuições da publicidade online para os debates sociais brasileiros sobre o casamento civil igualitário. *Animus. Revista Interamericana de Comunicação Midiática*, v. 13, n. 26, 2014. Citado na página 15.

LORENA, A. C.; CARVALHO, A. C. P. L. F. de. Uma introdução às Support Vector Machines. *Revista de Informática Teórica e Aplicada - RITA*, v. 14, n. 2, p. 43–67, 2007. Citado 5 vezes nas páginas viii, 4, 20, 26 e 27.

MANNING, C. D.; RAGHAVAN, P.; SCHÜTZE, H. *Introduction to Information Retrieval*. [S.l.]: Cambridge University Press, 2008. Citado na página 28.

MARTINS, A. F.; VILELA, G. O.; SOARES, M. A. As perspectivas da jurisprudência sobre o discurso de ódio nas redes sociais. In: *IX ENCONTRO DA ANDHEP - GT08 - Direitos Humanos, Comunicação e Novas Tecnologias*. [S.l.: s.n.], 2016. p. 47–67. Citado na página 14.

MATOS, P. F. et al. *Conceitos sobre Aprendizado de Máquina*. [S.l.], 2009. Citado 2 vezes nas páginas 17 e 25.

MEDEIROS, H. *Conhecendo JSON*. 2012. Http://www.linhadecodigo.com.br/artigo/3623/conhecendo-json.aspx. Online; acessado dia 26 de Novembro de 2017. Citado na página 40.

MILENA, L. *'Internet molda o cérebro das pessoas', diz Nicolelis*. 2016. Https://jornalggn.com.br/noticia/Online; acessado dia 20 de Novembro de 2017. Citado na página 9.

MITCHELL, T.; HILL, M. *Machine Learning*. first. [S.l.]: McGraw-Hill Science and Engineer ing and Math, 1997. Citado 3 vezes nas páginas 2, 17 e 38.

MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. In: REZENDE, S. O. (Ed.). *Sistemas Inteligentes: Fundamentos e Aplicações*. Barueri, SP: Editora Manole Ltda, 2003. cap. 4, p. 89 – 114. Citado na página 17.

NEVES, A. *Web 2.0: Definição, Características e Exemplos*. 2007. Https://kmol.pt/artigos/2007/07/01/web-20-definicao-caracteristicas-e-exemplos/. Online; acessado dia 20 de Novembro de 2017. Citado na página 7.

ONU. *Pacto Internacional sobre Direitos Civis e Políticos*. [S.l.], 1996. Citado na página 13.

O'REILLY, T. *What Is Web 2.0 Design Patterns and Business Models for the Next Generation of Software*. 2005. Http://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html. Online; acessado dia 20 de Novembro de 2017. Citado na página 7.

PARISER, E. *Tenha cuidado com os 'filtros-bolha' online.* 2011. $https://www.ted.com/talks/eli_pariser_beware_online_filter_bubbles?language=pt-br$. Online; acessado no dia 17 de Novembro de 2017. Citado na página 10.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011. Citado 2 vezes nas páginas 4 e 38.

POZO, A. T. R. *Uma teoria e metodologia de aprendizado indutivo*. 2017. Http://www.inf.ufpr.br/aurora/tutoriais/aprendizadomaq/. Online; acessado dia 24 de Novembro de 2017. Citado na página 17.

PRATI, R. C.; BATISTA, G. E. A. P. A.; MONARD, M. C. Curvas ROC para avaliação de classificadores. *EEE LATIN AMERICA TRANSACTIONS*, v. 6, n. 2, p. 215–222, jun 2008. Citado 3 vezes nas páginas ix, 29 e 30.

RASCHKA, S. *Naive Bayes and Text Classification – Introduction and Theory*. 2014. $http://sebastianraschka.com/Articles/2014_naive_bayes_1.html$. Online; acessado dia 25 de Novembro de 2017. Citado na página 24.

RECUERO, R. Um estudo do capital social gerado a partir de redes sociais no orkut e nos weblogs. *FAMECOS*, Porto Alegre, n. 28, 2005. Citado na página 8.

RECUERO, R. *Redes sociais na Internet*. Porto Alegre: Ed. Sulina, 2009. Citado na página 8.

REIS, J. et al. Uma análise do impacto do anonimato em comentários de notícias online. In: *XXXVI Congresso da Sociedade Brasileira de Computação*. Universidade Federal de Minas Gerais (UFMG) – Brasil: [s.n.], 2016. p. 1290–1304. Citado 2 vezes nas páginas 1 e 9.

RJTV. *Polícia investiga se presos podem estar ligados a um grupo de extermínio de moradores de rua*. 2017. Https://g1.globo.com/rio-de-janeiro/noticia/policia-investiga-se-presos-podem-estar-ligados-a-um-grupo-de-exterminio-de-moradores-de-rua.ghtml. Online; acessado dia 19 de Novembro de 2017. Citado na página 1.

ROSSI, R. G. *Classificação automática de textos por meio de aprendizado de máquina baseado em redes*. Tese (Doutorado) — USP, São Carlos, 2015. Citado na página 21.

RUSSELL, S.; NORVIG, P. *Inteligência Artificial*. third. [S.l.]: Campus, 2013. Citado na página 17.

SARMENTO, D. A liberdade de expressão e o problema do 'hate speech'. *Livres e iguais : estudos de Direito Constitucional. Rio de Janeiro: Lumen Juris*, 2006. Citado na página 13.

SCHöLKOPF, B.; SMOLA, A. J. *Learning with Kernels*. Cambridge, Massachusetts: The MIT Press, 2002. Citado na página 26.

SILVA, A. L. da; VIEIRA, E. S.; SCHNEIDER, H. N. O uso das redes sociais como método alternativo de ensino para jovens: Análise de três projetos envolvendo comunidades virtuais. In: *IV Colóquio Educação e Contemporaneidade*. [S.l.: s.n.], 2010. p. 1–13. Citado 2 vezes nas páginas 6 e 7.

SILVA, L. et al. Analyzing the targets of hate in online social media. In: *THE 10TH INTERNATIONAL AAAI CONFERENCE ON WEB AND SOCIAL MEDIA (ICWSM-16)*. Cologne, Germany: [s.n.], 2016. p. 687–690. Citado 2 vezes nas páginas 14 e 36.

SILVA, R. L. da et al. Discurso de ódio em redes sociais: jurisprudência brasileira. *Revista DireitoGV*, 2011. Citado 2 vezes nas páginas 1 e 11.

SOUSA, P. V.; BRAGA, V. Self, identidade, redes sociais: definições e relações entre a psicologia social e a comunicação em tempos de redes sociotécnicas. In: *Sétimo Simpósio Nacional da Associação da Associação Brasileira de Cibercultura*. [S.l.: s.n.], 2013. p. 01–15. Citado na página 9.

STRANDBERG, K.; BERG, J. Comentários dos leitores dos jornais online: Conversa democrática ou discursos de opereta virtuais? *Comunicação e Sociedade*, v. 23, p. 110–131, 2013. Citado na página 9.

SZPACENKOPF, M.; GUERRA, R.; GRANDELLE, R. *Especialistas repudiam liminar que trata homossexualidade como doença*. 2017. Https://oglobo.globo.com/sociedade/especialistas-repudiam-liminar-que-trata-homossexualidade-como-doenca-21840633. Online; acessado dia 21 de Novembro de 2017. Citado na página 12.

TAM, D. *Facebook processes more than 500 TB of data daily*. 2012. Https://www.cnet.com/news/facebook-processes-more-than-500-tb-of-data-daily/. Online; acessado dia 21 de Novembro de 2017. Citado na página 19.

TAN, P.-N.; STEINBACH, M.; KUMAR, V. *Introduction to Data Mining*. [S.l.]: Addison-Wesley, 2015. Citado 2 vezes nas páginas 4 e 21.

TANG, J.; ALELYANI, S.; LIU, H. *Feature Selection for Classification: A Review*. 2016. *https : //web.archive.org/web/20160314145552/http : //www.public.asu.edu/ jtang20/publication/feature*$_s$*election*$_f$*or*$_c$*lassification.pdf*. Online; acessado dia 24 de Novembro de 2017. Citado na página 24.

TELES, V.; SANTOS, D.; SOUZA, E. Uma análise comparativa de técnicas supervisionadas para mineração de opinião de consumidores brasileiros no twitter. In: *XIII Encontro Nacional de Inteligência Artificial e Computacional*. [S.l.: s.n.], 2016. p. 217 – 228. Citado 2 vezes nas páginas 3 e 64.

TELLES, A. *A revolução das mídias sociais*. São Paulo: M. Books, 2010. Citado 2 vezes nas páginas 7 e 8.

TRIVEDI, M. et al. Comparison of text classification algorithms. *International Journal of Engineering Research & Technology (IJERT)*, v. 4, n. 2, p. 334–336, 2015. Citado na página 3.

VIJAYARANI, S.; ILAMATHI, J.; NITHYA. Preprocessing techniques for text mining - an overview. *International Journal of Computer Science & Communication Networks*, v. 5, n. 1, p. 7–16, 2015. Citado na página 22.

WASSERMAN, S.; FAUST, K. *Social Network Analysis: Methods and Applications*. [S.l.]: Cambridge University Press, 1994. Citado na página 8.

WELLS, K. *nohomophobes*. 2012. Http://www.nohomophobes.com/. Online; acessado dia 15 de Novembro de 2017. Citado 2 vezes nas páginas viii e 34.

WIEBE, J.; WILSON, T.; CARDIE, C. Annotating expressions of opinions and emotions in language. *Kluwer Academic Publishers*, 2005. Citado na página 64.

WULCZYN, E.; THAIN, N.; DIXON, L. Ex machina: Personal attacks seen at scale. In: *WWW2017 Session 7H: Web Mining*. Australia: [s.n.], 2017. p. 1391–1399. Citado 2 vezes nas páginas 12 e 33.