UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

Rotsen Diego Rodrigues de Albuquerque

Estudo Comparativo de Algoritmos de Classificação Supervisionada para Classificação de Polaridade em Análise de Sentimentos

Recife

2019

Rotsen Diego Rodrigues de Albuquerque

Estudo Comparativo de Algoritmos de Classificação Supervisionada para Classificação de Polaridade em Análise de Sentimentos

Monografia apresentada ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Universidade Federal Rural de Pernambuco – UFRPE

Departamento de Estatística e Informática

Curso de Bacharelado em Sistemas de Informação

Orientador: Gabriel Alves

Recife

2019

Dados Internacionais de Catalogação na Publicação
Universidade Federal Rural de Pernambuco
Sistema Integrado de Bibliotecas
Gerada automaticamente, mediante os dados fornecidos pelo(a) autor(a)

R849e Albuquerque, Rotsen
Estudo Comparativo de Algoritmos de Classificação Supervisionada para Classificação de Polaridade em Análise de Sentimentos / Rotsen Albuquerque. - 2019.
44 f. : il.

Orientador: Gabriel Alves.
Inclui referências e apêndice(s).

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal Rural de Pernambuco, Bacharelado em Sistemas da Informação, Recife, 2019.

1. Análise de Sentimentos. 2. Aprendizado de Maquina. 3. Mineração de Dados. 4. Classificação de Texto. 5. Mineração de opinião. I. Alves, Gabriel, orient. II. Título

CDD 004

ROTSEN ALBUQUERQUE

ESTUDO COMPARATIVO DE ALGORITMOS DE CLASSIFICAÇÃO SUPERVISIONADA PARA CLASSIFICAÇÃO DE POLARIDADE EM ANÁLISE DE SENTIMENTOS

Monografia apresentada ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Aprovada em: 13 de Dezembro de 2019.

BANCA EXAMINADORA

_____

Gabriel Alves (Orientador)
Departamento de Estatística e Informática
Universidade Federal Rural de Pernambuco

_____

Rafael Ferreira
Departamento de Estatística e Informática
Universidade Federal Rural de Pernambuco

Agradecimentos

Agradeço a Deus, pelo dom da vida e por todos os momentos, principalmente os mais difíceis, em que estive estudando para a concretização desse sonho. A minha mãe, pelo amor, incentivo e apoio incondicional que me ajudou todo esse período em que estive ocupado com os meus estudos. Ela nunca mediu esforços para me ajudar. Ao meu orientador, professor Gabriel Pereira, por ter acreditado prontamente no meu potencial, pelo suporte no pouco tempo que lhe coube, bem como, dando me total apoio sobre o tema escolhido, na orientação do desenvolvimento de todo o trabalho de pesquisa e elaboração dessa monografia. Ao meu amigo Ademir, que com toda correria foi um ombro amigo e paciente para acreditar nesse sonho. Aos grande amigos do BSI, Ingrid, Tancicleide, Jessica que se tornaram acima de tudo amigos. A minha noiva que sempre esteve ao meu lado em todas as situações e que insistentemente procurou me incentivar e lembrar de tudo que eu sou capaz. A UFRPE e às pessoas com quem convivi e me relacionei nesses espaços ao longo desses anos que não foram poucos. A experiência de uma produção compartilhada na comunhão com os meus amigos nesses espaços que me foram a melhor experiência da minha formação acadêmica.

“A persistência é o caminho do êxito.”
(Charles Chaplin)

Resumo

Com o grande aumento de dados na internet, mostra-se uma rica fonte para a avaliação da opinião pública sobre uma entidade específica. Consequentemente, o número de opiniões disponíveis torna impossível uma tomada de decisão se for necessário ler e analisar todas as opiniões. Como o uso de *Machine Learning* tem sindo bastante usado, irei apresentar um estudo comparativo de dois algoritmos para classificar os comentários usando técnicas de processamento de linguagem natural e Análise de Sentimentos. O dados obtidos foram obtidos manualmente onde através do site de competições chamado *Kaggle* temos cerca de 50.000 comentários sobre diversos filmes. Este estudo tem por finalidade usar também os conceitos da ciência de dados e *Machine Learning*, processamento de linguagem natural e analises de sentimentos para agregar mais informação sobre a industria de entretenimento e cinema. Por isso esses algoritmos foram criados para que seja possível mostrar os resultados para esse domínio nos *reviews* de filmes registrados no site da grande industria cinematográfica o famoso IMDB. Após a aplicação dos treinos e testes, a máquina teve uma Acurácia de 86% sobre a predição de textos comentados de filmes.

Palavras-chave: Análise de Sentimentos, Aprendizado de Maquina, Mineração de Dados, Classificação de Texto, Mineração de opinião, Linguagem Natural

Abstract

The huge increase of data on the Internet, it is a rich source for public opinion assessment of a specific subject. Consequently, the number of opinions available makes decision-making impossible if it is necessary to read and analyze all opinions. Since the use of Machine Learning has been widely used, I will present a comparative study of two algorithms for classifying movie comments using techniques of natural language processing and Sentiment Analysis. The data obtained were obtained manually where through the competition site called Kaggle where we have about 50,000 comments on various films. The purpose of this study is also to use the concepts of data science and Machine Learning, natural language processing and sentiment analysis to add more information about the entertainment and film industry. That is why these algorithms were created so that it is possible to show the results for this domain in the of movies comments registered in one big site/platform of movie industry, the famous IMDB. After training and testing, the machine had an accuracy of 86 % on predicting sentiments on commented text from movies.

Keywords: Sentiment Analysis, Machine Learning, Data Mining, Natural Processing Language, Text Classifing, Opinion Mining,

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

OP	Opinion mining
DL	Deep Learning
AS	Análise de Sentimentos
AM	Aprendizado de máquina
PLN	Processamento de Linguagem Natural

Sumário

1 Introdução

Quando um consumidor tem interesse por um produto ou serviçoo´e comum que ele procure referencias ou opiniões. Empresas que vendem produtos e disponibilizam serviços também são motivadas a ter conhecimento das opiniões dos consumidores, tendo que procurar formas de analisar essas informações para conduzir ações de marketing e tomada de decisão.

Nesse contextom, pesquisadores da´area de Processamento de Linguagen Natural (PLN) tem buscado extrair informações´uteis de dados n ao estruturados. O termo *big datase* refere ao massivo conjunto de informações digitais coletadas. Todos os dias são criados 2.5 quinquilhões de bytes de dados, tanto que 90% dos dados do mundo hoje foram criados apenas nos´últimos dois anos. Essa grande quantidadede dados faz com que a análise manual se torne uma tarefa impossıvel, sendo necessária a criacão de métodos automáticos para analisar os dados (LIU et al., 2010).

Os estudos em análise automáticas de documentos vêm permitindo esses avanços no reconhecimento de aspectos subjetivos. Dentre ele está a classificação de polaridade do texto, ou melhor, o quão positiva e negativamente são as opiniões descritas por pessoas. Assim, esse trabalho tem como objetivo o estudo e a aplicação de uma ferramenta que, conterá dois algoritmos de classificação de sentimentos, sendo ele capaz de avaliar a polaridade de comentário extraído de uma base de dados de comentários de filmes do IMDB, baseando-se em técnicas de mineração de textos.

A análise de sentimentos pode ser usada em algum sistema ou como ferramenta de apoio a decisão. Isso devido a propagação de opiniões, comentários e avaliações, sobre produtos,filmes e empresas. Essa identificação de sentimentos em textos é uma das áreas mais ativas na atualidade (LIU et al., 2010). O Sentimento ë uma atitude, pensamento ou julgamento sobre uma emoção. A análise de sentimento também conhecida como mineração de opinião, estuda esse sentimento em relação a determinada entidade ou contexto.

Com a abertura dos usuários em cada vez mais exprimir suas experiências ou expectativas sobre todos os assuntos, a análise de sentimentos começou a ter seu valor social muito grande. Note na figura abaixo, que a tendência de buscas no Google pelo termos *”Sentiment Analysis”* está em crescimento desde 2004.

Usando uma grande massa de dados, um problema fundamental na análise de sentimentos seria a categorização ou classificação em escala de milhares de dados. (PANG; LEE et al., 2008). Ou seja, baseado no escopo de um texto, a análise de sentimentos parece ter um forte fundamento com suporte a esses dados massivos,

Figura 1 – Gráfico de Pesquisa do Termo ”Sentiment Analysis”no Google Trends.

extraindo no nível de sentença, a qual categoria uma opinião pertence. Outra oportunidade desafiadora seria o exemplo de aplicação em um sistema de recomendações onde leva em consideração de *reviews* de filmes baseados em informação de itens de texto, sabendo por exemplo qual a reação do público ao lançamento de um filme, ou se alguma discussão está sendo negativa ou positiva, em algum fórum sobre filmes. Tudo isso ajudaria a tornar mais inteligente decisões de negócio.

Assim nesse trabalho tivemos o desafio de trabalhar com uma grande massa de dados feitos no site de um site especializado em Filmes. Eles textos dos comentários já foram extraídos do plataforma onde ao todo temos 50 mil em língua inglesa. Por serem extraídos de uma plataforma online onde os próprios usuários fazem os comentários e os mesmos não são revisados, há uma grande incidência de erros ortográficos, gírias, abreviações e jargões, o que tende a dificultar a análise quando comparado a um texto formalmente bem escrito.

Dessa forma, diferentes fases de pré-processamento são aplicadas na base com o intuito de melhorar a performance dos algoritmos de classificação utilizados. Dada a grande quantidade de texto analisada, ferramentas de mineração de dados mais robustas se fazem necessárias. Assim, modelos de aprendizagem de máquina são criados utilizando-se do processamento mais robusto.

1.1 Justificativa e Motivação

A indústria do cinema está cada vez mais agressiva. Serviços como o Amazon Prime Video, Disney+ e a Netflix estão projetando um investimento de quase 500 Milhões de reais em produções e licenciamentos brasileiros. Só a Netflix por exemplo registrou alta de 65% no lucro líquido do terceiro trimestre de 2019 ante o mesmo período do ano passado. O lucro da companhia entre julho e setembro foi de US$ 665,2 milhões (cerca de R$ 2,76 bilhões). No terceiro trimestre do ano passado, foram registrados US$ 403 milhões. Diante disso essas empresas estão encorajando ainda mais os usuários a contribuir com opiniões e experiencias assistindo suas obras. Isto por-

que quando uma pessoa escreve um comentário, ela provavelmente comenta vários aspectos do produto, além disso, opiniões sobre os aspectos não são iguais, alguns aspectos recebem comentários negativos, outros recebem positivos e alguns recebem neutros. Essas opiniões são úteis para usuários e fabricantes e produtores do cinema. Como resultado, a área de mineração de opinião e análise de sentimentos desfrutou uma atividade de pesquisa estourada recentemente.

Trabalhar com analise de texto é um trabalho bem oneroso e difícil. Temos centenas de idiomas no nosso planeta e o processamento de linguagem natural ainda está muito novo quanto ao desenvolvimento de software e inteligencia artificial. Por tanto a iniciativa de mais e estudo em estimar qual a eficacia de dois algoritmos no processo de classificação se faz necessária nesse trabalho.

1.2 Objetivo

Este trabalho tem como objetivo, analisar e processar textos com comentários de filmes (*reviews*) para classificá-los como positivos e negativos.

Dos objetivos específicos deste trabalho, podemos destacar os seguintes pontos:

- Implementar uma ferramenta que a partir das interações de *reviews* de filmes evidenciando automaticamente qual o sentimento dos usuário em relação ao que foi escrito.
- Contribuir com os estudos de análise de sentimento apresentando uma alternativa para o contexto de filmes.
- Contribuir com os estudos de análise de sentimento apresentando uma alternativa para o contexto de filmes.
- Utilização de processamento de linguagem natural para realização das classificações dos textos.

1.3 Estrutura do Trabalho

Este trabalho está dividido em seis capítulos. O primeiro capítulo 1 apresenta a introdução do trabalho, com uma breve motivação, justificativa e objetivo definidos. No Capítulo 2 apresentamos alguns trabalhos que buscam mostrar como esta o andamento das pesquisas na área de analise de sentimentos, mineração de opinião,processamento de linguagem natural e como foram seus resultados e objetivos principais O terceiro é dedicado à apresentação dos métodos de Aprendizado de Máquina, assim como

também qual o *framework* foi utilizado. No Capítulo 4 descrevemos como foi realizado o experimento, como está organizado a estrutura dos arquivos, como foi realizado a escolha e parâmetros das principais funcionalidades para a analise dos dados. O penúltimo capitulo, o quinto, este é designado para a apresentação dos resultados e quais métricas foram escolhidas para a avaliação. No Capítulo 6 discutimos as conclusões, quais foram as limitações e como isso pode gerar os trabalhos futuros para o tema.

2 Trabalhos Relacionados

Nesta sessão abordaremos vários trabalhos relacionados ao tema de Análise de sentimentos, aprendizado de máquina e processamentos de linguagem natural. Todos eles possuem pontos interessantes que são em comum para este trabalho, importantes para definição das técnicas e métodos usados.

Apesar de existirem muitos trabalhos com o estado da arte bem avançado na literatura, cada desafio foi importante para generalizar e analisar as aplicações para o domínio em que estamos tratando no contexto do cinema. Além disso tivemos como buscar de maneira mais sucinta bons resultados para os conceitos acadêmicos.

São vários os estudos onde a extração de sentimento de características em comentários de consumidores ́e realizada através de Mineração de Opinião abrangendo técnicas de PLN (LIU; LI; GUO, 2012),(ARAVINDAN; EKBAL, 2014) propõem técnicas baseadas em mineração de dados e aprendizado de máquina para analisar comentários automaticamente, extraindo as características do produto e determinando a polaridade de cada características. Seus experimentos mostraram que o classificador teve uma precisão de 79.67% quando aplicados a cinco produtos populares.

Quanto ao contexto de analise de comentários livros e filmes, (COELHO; LIMA; OMAR, 2017) obteve uma acurácia de 59.3% na base de dados da Amazon, e 72.09% no Twitter. Esse trabalho se diferenciou devido a utilização de um framework chamado PAFRA.

No trabalho de (LONGHI; BERCHT; BEHAR, 2007) e colaboradores é apresentado uma estudos que introduzem a mineração de texto como uma ferramenta para o uso de informática para educação. A partir do momento que as máquinas e as novas tecnologias de comunicação passaram a mediar significadamente as relações sociais tem-se percebido as mudanças nas formas de comunicação. Assim através dos fóruns no AVA (ambiente de avaliação e aprendizagem) buscou-se demonstrar a viabilidade de reconhecer os diversos estados de ânimos nos textos disponibilizados no Fórum do AVA. Para os casos demonstrados, os autores considerando 38 mensagens postadas em um fórum sobre planejamento do trabalho final de uma disciplina o tempo de execução foi de 1,84s. Naturalmente faz-se necessário aplicar a metodologia em um numero muito maior de situações com o intuito de verificar a viabilidade dos algoritmos e também a veracidade da pesquisa.

(MENGER et al., 2018) aplicaram os algoritmos de *(*Deep Learning) e clássicos para predizer incidentes de violência entre pacientes de um hospital. Basicamente os dados para serem processados foram os textos da analises clinicas e histórico dos

pacientes. Essa classificação tem como saída Y qual a probabilidade do paciente apresentar um comportamento agressivo em sua admissão na clínica. A estratégia usada para representação textual foi a mesma em que estaremos usando nesse trabalho. A clássica *bag-of-words* onde uma sentença é representada através das duas palavras, desconsiderando a gramatica ou sequência das palavras. A parti desse saco de palavras foi usado a técnica de *tokenization* (tokenização) onde as palavras ganham valores chaves para poder ter uma representação em uma matriz e assim ser consumida pelos algoritmos. Depois de vários ajustes o melhor resultado foi com o algoritmo de redes neurais recorrentes. Nesse artigo vimos que existe uma melhoria muito pequena entre os algoritmos de *Deep Learning* e os clássicos. Nos confirmamos a hipótese de que para muitos casos onde existe uma lacuna de dados para usar técnicas de *Deep Learning*, os clássicos são muitos bem vindos.

Em 2014, Almatrafi, Parack e Chavan identificaram as intenções de voto nas eleições da Índia usando o Twitter. O grande volumes de dados textuais que o twitter tem se torna um grande atrativo para *opinion mining* (mineração de opinião) e SA. Ao total foram usados 650.000 tweets para entender a opinião do eleitor e identificar seus padrões. Um diferencial desse artigo foi a integração com um banco de dados geográficos. Assim foi possível entender como os sentimentos das regiões mudavam. O resultado final para avaliação foi de uma precisão de 70%.

Já o estudo feito por Zuo, em 2018, trouxe um dos conceitos mais lucrativos dessa décadas, games. Dois algoritmos supervisionado, Árvore de decisão e *Naive Bayes* obtiveram 75%. Considera-se um bom resultado devido a quantidade de lixo existe em fóruns de games. Por um lado também é importante a existência desses dados para o modelo não decorar as emoções. Para esses dados não balanceados onde a classificação foi feita na manualmente foi usada a validação cruzada. A proporção de dados para treino e teste foi de 75% e 25% respectivamente.

Algumas pesquisas buscando entender as opiniões que os usuários do rede social Twitter exprimem já foram conduzidas. No ano de 2009, (BOLLEN; DAVIS, 2009) desenvolveu um dos primeiros trabalhos com enfoque de classificar emoções presentes em tweets. O objetivo do estudo realizado foi analisar a flutuação de sentimento dessas mensagens e buscar eventos sócio-econômicos e políticos que pudessem estar relacionados às oscilações identificadas. Para isso, os autores utilizaram um instrumento chamado POMS ex, que classifica os tweets em 6 categorias diferentes de humor, a partir da comparação dos termos que compõem a mensagem com termos, definidos previamente, associados a cada uma das categorias. Em 2010, (PAK; PAROUBEK, 2010) utilizou também um classificador Naïve-Bayes para categorizar tweets em positivo ou negativo, com base em N-gramas e na classificação gramatical de partes do texto.

Já (DAVIDOV; TSUR; RAPPOPORT, 2010) buscou categorizar os tweets utilizando apenas as mensagens com hashtags – palavras precedidasde que representam o sentimento e/ou o assunto alvo daquela mensagem – e emoticons contidos no texto para treinar o classificador utilizado e categorizar tweets entre diversas emoções. Além disso, aspectos como pontuação e as palavras utilizadas também foram consideradas para a extração de sentimento. Outra vertente da análise do conteúdo gerado no Twitter é a extração de informação a partir do estudo de posts publicados nessa rede. (SAKAKI; OKAZAKI; MATSUO, 2010) e (ACHREKAR et al., 2011) promoveram pesquisas cujo objetivo era detectar acontecimentos de forma mais rápida através da análise das mensagens divulgadas. Nesses casos, a intenção era criar uma ferramenta capaz de detectar terremotos e epidemias de gripe, respectivamente, antes mesmo de tais eventos serem anunciados oficialmente.

Já em (LIU; LI; GUO, 2012), seu objetivo era utilizar as opiniões compartilhadas no Twitter para auxiliar empresas a tomarem decisões em suas campanhas de marketing. A ideia foi utilizar a análise de sentimento de (**??**) para tornar possível o acompanhamento da opinião dos clientes em relação a serviços e produtos e permitir que empresas tomassem decisões antes mesmo de os clientes chegarem até elas com reclamações e sugestões. Dentre os trabalhos citados, podemos verificar que nenhum deles buscou analisar sentimento relacionado a fatos divulgados na mídia. Na verdade, a estratégia sempre foi inversa: analisar os sentimentos e buscar ocorrências que pudessem estar associadas a eles.

Neste trabalho, o foco é fomentar ainda mais os estudos na analise de sentimentos. Onde fizemos de diferente uma analise individual das palavras através da tokenização, conhecendo assim a polaridade de cada uma das palavras, porém para a avaliação do algoritmo, usamos a abordagem de sentenças.

3 Referencial Teórico

Neste capítulo é apresentado todo o referencial teórico deo que será utilizado para o desenvolvimento do trabalho.

3.1 *CRISP-DM 2.0*

O CRIPS-DM é a abreviação de Cross Industry Standard Process for Data Mining, que pode ser traduzido como Processo Padrão Inter-Indústrias para Mineração de Dados. No processo de mineração de dados existe um uma metodologia que auxilia no entendimento do negócio, preparação dos dados, modelagem, avaliação e implantação do modelo escolhido. Especialistas utilizam essa abordagem frenquentemente em mineração de dados para atacar problemas.

Segue o modelo hierárquico na figura abaixo:

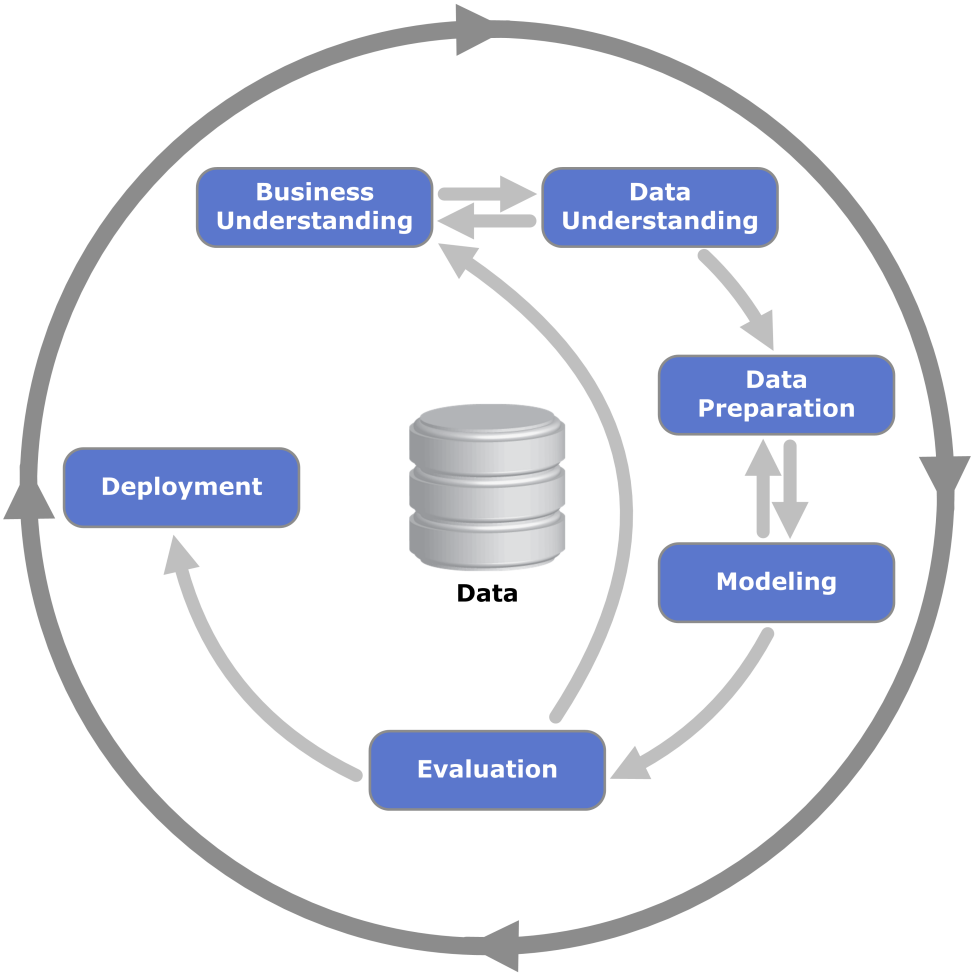

Figura 2 – Fases do modelo CRISP-DM
(LORENA; CARVALHO, 2007)

O CRISP-DM define um projeto como um processo cíclico, em que várias iterações podem ser usadas para permitir resultados finais mais sintonizado com os objetivos do negócio.

Depois de identificar o objetivo (fase de entendimento do negócio), os dados precisam ser analisado (compreensão de dados) e processado (Preparação de dados). Segundo (MORO; LAUREANO; CORTEZ, 2011), dados possuem conceitos (o que precisa ser aprendido), instâncias (registros independentes relacionados a uma ocorrência) e atributos (que caracterizam um aspecto específico de um determinado instância). A fase de modelagem constrói o modelo que representa o conhecimento aprendido (por exemplo, dada uma instância, o modelo pode ser usado para prever o valor alvo que representa a meta definida). A seguir, o modelo é analisado na fase avaliação, no que se refere a sua *peformance*. Por exemplo, nas tarefas de classificação (prever um alvo discreto), métricas comuns são a matriz de confusão (PROVOST; KOHAVI, 1998) e a Curva ROC (*Receiver Operating Characteristic*). Se o obtido modelo não é bom o suficiente para ser usado para apoiar negócios, uma nova iteração para o CRISP-DM é definida. Senão, modelo é implementado em um ambiente de tempo real (Fase de implantação).

As fases do CRISP-DM, em tradução livre, são: entendimento do negócio, entendimento dos dados, preparação dos dados, modelagem, avaliação, e utilização. Cada uma das fases, bem como suas tarefas, será explicada nas sessões adiante:

Segundo (MORO; LAUREANO; CORTEZ, 2011), dados possuem conceitos portanto a fase inicial, de entendimento do negócio (Business Understanding) se refere objetivos do projeto e dos requisitos necessários para a sua realização na perspectiva do domínio do projeto. Nessa fase o projeto é mapeado como um problema de mineração de dados e é traçado um plano que guiará o projeto durante o seu desenvolvimento. São feitos cronogramas, listas de requisitos, seleção de ferramentas e técnicas de mineração de dados e etc.

A fase de entendimento dos dados (Data Understanding) se baseia na familiarização com os dados. Essa fase dá inicio a um processo de extração, verificação e análise dos dados coletados. Os dados devem ser verificados quanto à sua qualidade e se serão realmente úteis ao que é proposto pelo projeto. Nessa fase são feitos relatórios de descrição dos dados, da qualidade, da coleta inicial e etc.

A próxima fase é a de preparação dos dados (Data Preparation). Uma vez que os dados foram avaliados e considerados satisfatórios pela fase de entendimento, aqui os dados coletados passam por um processo de preparação que transforma os dados brutos iniciais em dados prontos para serem aplicados às ferramentas de mineração de dados. Nessa fase são realizadas tarefas como seleção de atributos, limpeza, construção e formatação dos dados.

Na fase de modelagem (Modelling) são selecionadas técnicas de modelagem para aplicação dos dados. Nessa fase também são escolhidos valores ótimos de parametrização para serem utilizados nessas técnicas de forma a obter os melhores resultados possíveis.

Na fase de avaliação (Evaluation), por exemplo, nas tarefas de classificação (prever um alvo discreto), métricas comuns são a matriz de confusão (PROVOST; KOHAVI, 1998) e a Curva ROC (*Receiver Operating Characteristic*) são as mais usadas nessa faa=se. Os modelos são avaliados quanto a sua qualidade utilizando dados de teste diferentes dos dados utilizados para o treinamento e os resultados são verificados de acordo com os critérios adotados no entendimento do negócio.

A última fase é a de colocação em uso (Deployment). Nessa fase os resultados obtidos são utilizados por um analista para auxiliar em tomadas de decisões. Além disso, o modelo desenvolvido pode ser utilizado em outras bases de dados.

3.2 Aprendizado de Máquina

As técnicas de AM tem o princípio de inferência denominado indução, no qual obtém-se as conclusões genéricas a partir de um conjunto particular de exemplos(LORENA; CARVALHO, 2007).

No aprendizado supervisionado tem-se a figura de um professor externo, o qual apre-senta o conhecimento do ambiente por conjuntos de exemplos na forma: entrada, saída desejada . O algoritmo de AM extrai a representação do conhecimento a partir desses exemplos. O objetivo é que a representação gerada seja capaz de produzir saídas corretas para novas entradas não apresentadas previamente.

Na análise não supervisionada, não existe a presença de um professor, ou seja, os exemplos para treinamento do modelo da maquina, não sabem o resultado dos exemplos. Para tal abordagem os algoritmos usados usam outras técnicas como agrupamento por semelhanças segundo algum critério pre selecionado.

Neste estudo iremos usar o trabalho com dados supervisionado. O nosso banco da dados já possui toda a entrada de dados dos comentários classificado previamente.

O processo do estudo entre os classificadores (algoritmos de aprendizagem de maquina) é que a partir de uma amostra de dados (texto) ele pode ser visto como uma função que recebe um dados x e fornece uma predição y.

Os rótulos ou classes representam o fenômeno de interesse sobre a qual estamos fazendo previsões. Neste caso nosso y representa valores de 1...0. Tem-se um problema de classificação. Denominado binário.

O modelo abaixo descreve que dado o tempo Xi ele possui m atribulos (pala-

vras) ou seja Cada dado xi possui m atributos, ou seja, $Xi = (Xi1, ..., Xim)$. As variáveis yi representam as classes. A partir dos exemplos e as suas respectivas classes, o algoritmo de AM extrai um classificador.

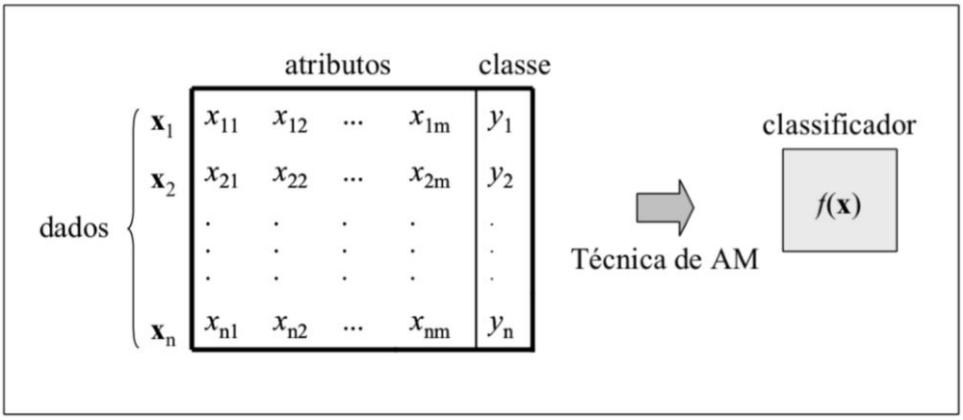

Figura 3 – Indução de classificador em aprendizado supervisionado. (LORENA; CARVALHO, 2007)

3.2.1 Naive Bayes

Um classificador Naive bayes é um modelo probabilístico simples baseado na regra de Bayes ao longo com uma forte suposição de independência. O modelo Naïve Bayes envolve uma suposição de independência condicional simplificadora. Isso é dado uma classe (positiva ou negativa), as palavras são condicionalmente independentes uma da outra. Essa suposição não afeta a precisão na classificação do texto muito, mas cria algoritmos de classificação realmente rápidos aplicáveis ao problema. Rennie et al discutem o desempenho de Naïve Bayes em tarefas de classificação de texto em seu artigo de 2003. [6] No nosso caso, a probabilidade máxima de probabilidade de uma palavra pertencente a uma classe específica é dada pela expressão:

$$P(B/A) = \frac{QntdeVezesqueapareceB}{QnttotaldeBnodocumento}$$

As contagens de frequência das palavras são armazenadas em tabelas de hash durante a fase de treinamento. De acordo com a Regra de Bayes, a probabilidade de um documento específico pertencer a uma classe B é dada por,

$$P(B/A) = \frac{P(B) * P(A/B)}{P(A)}$$

- P(B/A) : Probabilidade A acontecer dado B

- P(B) : Probabilidade de B acontecer
- P(A/B) : Probabilidade B acontecer dado A
- P(A) : Probabilidade A acontecer

Se usarmos a suposição simplificada de independência condicional, que dada uma classe (positivo ou negativo), as palavras são condicionalmente independentes uma da outra. Devido a Nesta suposição simplificadora, o modelo é denominado como ”ingênuo”.

$$P(B/A) = \frac{P(B) * P(A/B)}{P(A)}$$

- P(B/A) : Probabilidade A acontecer dado B
- P(B) : Probabilidade de B acontecer
- P(A/B) : Probabilidade B acontecer dado A
- P(A) : Probabilidade A acontecer

Aqui os $x_i s$ são as palavras individuais do documento. O classificador gera o classe com a probabilidade posterior máxima. 196 V. Narayanan, I. Arora e A. Bhatia Também removemos palavras duplicadas do documento, elas não adicionam mais em formação; esse tipo de algoritmo ingênuo de bayes é chamado Bernoulli Naïve Bayes. Incluir apenas a presença de uma palavra em vez da contagem foi encontrado para melhorar marginalmente, quando houver um grande número de exemplos de treinamento.

3.2.2 Random Forest

É um algoritmo de aprendizagem de máquina baseado em árvore de decisão. Uma árvore de decisão classifica uma instância baseada em variáveis decisórias seguindo o caminho da raiz até as folhas(BREIMAN, 2001). Variáveis decisórias (ou atributos) são usadas para tomar decisões, e suas respostas formam caminhos específicos em uma árvore.

Em análise de sentimentos podemos classificar cada atributo como uma *feature*, e cada resposta seria uma possível característica dessa *feature*. Uma árvore de decisão então poderia ser construída a partir de um subconjunto de dados já classificados. Infelizmente, essa abordagem oferece algumas desvantagens. Como sabemos que a linguagem natural pode variar muito, e que uma mesma palavra que aparece em classes positivas também pode aparecer em classes negativas. Então, construir apenas

uma árvore não é o suficiente para a criação de um modelo eficiente. Assim sendo, o algoritmo Random Forest propõe a criação de várias árvores de decisão baseadas em subconjuntos aleatórios de uma base de dados. Dessa forma, podemos classificar um documento baseado não só em uma árvore, mas, sim em uma "floresta". O *overfitting* também pode acontecer para essa técnica. Nesse caso, o que acontece é que uma árvore de decisão pode ficar tão profunda, ou seja, com tantos atributos, que nenhuma instância de teste possa ser corretamente classificada. Para esse problema, a solução é "podar" a árvore em determinados pontos, aumentando a generalidade de sua decisão. Pode parecer que podar uma árvore de decisão é jogar fora informações importantes e diminuir sua precisão, mas esse é um problema estatístico em que a generalização é preferida para que possamos maximizar nossos resultados.

Segue exemplo abaixo:

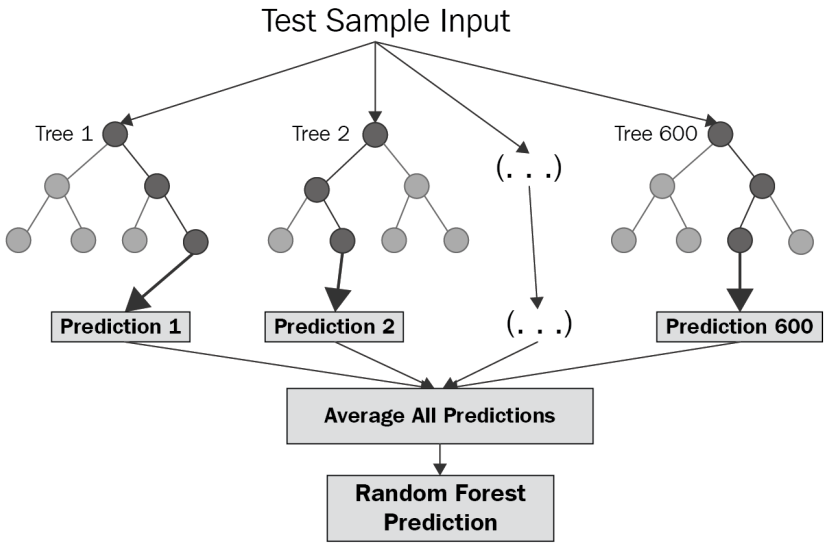

Figura 4 – Árvore de Decisão
(LORENA; CARVALHO, 2007)

3.3 Processamento de Linguagem Natural

Nessa sessão será abordado como foi criado os algoritmos de classificação, quais os processos envolvidos no trabalho, quais foram os dados necessários para ser usado no contexto de análise de texto no domínio de filmes e cinema.

Nosso banco de dados é composto por 25 mil comentários sobre filmes do site IMDB. Os textos estão divididos em comentários positivos e negativos. Cada comentário possui uma nota de satisfação que vai de 1 estrela para um filme ruim, até a nota 10 para um filme muito bom. Em relação a quantidade de *reviews*, o *dataset* encontra-se balanceado, com 12.500 textos para cada classificação. Foi empregado técnicas de mineração de textos sobre os comentários dos filmes. A figura abaixo a quantidade de

textos que cada nota de classificação possui. O texto é classificado como negativo se sua nota no comentário for menor que 5. Os demais comentários acima da nota 5 são considerados positivos.

Para realização dos experimentos utilizamos o *framework Sckilearn* da linguagem *python*.

Abaixo segue uma figura de todo o fluxo do trabalho

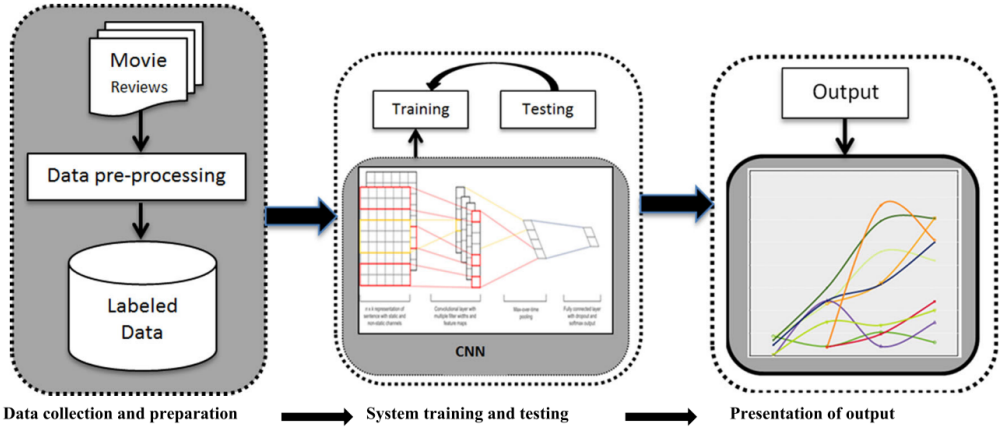

Figura 5 – Sentiment analysis process on product reviews.
(ALMATRAFI; PARACK; CHAVAN, 2015)

3.4 Processamento do Texto

Para um computador, processar *strings* ou dados que contenham caracteres de um alfabeto conhecido não é uma tarefa complexa, pois o mesmo possui capacidade de computar este tipo de informação. Entretanto é necessária uma técnicaque, na eventualidade desse tipo de dado ter um *reasoningde* alto nível, ou seja, que o mesmo seja qualitativo, a IA seja capaz de entender o significado aplicado a um contexto. Humanos começam a associação entre entidades e palavras quando ainda são crianças e bebês, atravésda interação com o ambiente.

Segundo Vygotsky (2000), gestos, palavras, linguagem corporal e outros traços característicos de um indivíduo são absorvidos e ao longo do tempo transformados em conhecimento, um processo gradual.

O processo de aprendizado humano é diferente em relação um computador. O computador toma um rumo distinto a partir do momento que se torna possível importar um dicionário inteiro em um robô, a fim de que ele conheça as palavras da língua e consiga criar conexões linguísticas baseado em um conjunto de sinônimos e antônimos. Entretanto existem algumas situações subjetivas que necessitam de maior atenção quando se trata de processamento de linguagem natural. O processamento da lingua-

gem segue uma cadeia de ações, conforme a Figura 2, que se iniciam na leitura de arquivos brutos de texto, selecionando dados de interesse a serem observados e minerando informações através da pré-seleção realizada. Então são utilizados padrões obtidos através da etapa anterior para obter conhecimento específico, que neste trabalho são os sentimentos contidos em textos.

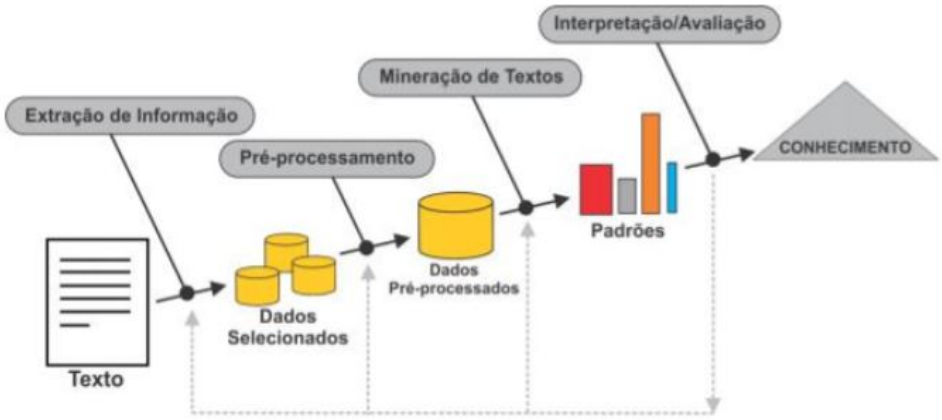

Figura 6 – Etapas de Processamento de Linguagem Natural (GONÇALVES, 2001)

O método mais comum para tratamento de texto é o de tokenização. Onde consiste em quebrar um texto em pequenos pedaços é conhecido esses pedaços são denominados *tokens*, sendo esses termos individuais detectados no processamento do texto (PERKINS, 2010) . Os textos podem ser *tokenizados* em sentenças, isto é gerada uma lista de sentenças, a partir do texto original, onde cada sentença é considerada um token. Tokenização de sentenças consiste em dividir as frases em palavras individuais. Todo esse processo de criação de lista de palavras advindas de uma sentença é uma parte essencial para todo o processamento de texto.

No conjuntos dos nosso *corpus* (vocabulário do texto) algumas palavras podem se apresentar com muita frequência, porém apresentam baixa relevância para expressar o significado de um sentença. Essas palavras são chamadas de *stopword* (HAPKE; LANE; HOWARD, 2019).

Normalmente esses artigos, conjunções, preposições, interjeições, verbos auxiliares e palavras muito repetidas na linguagem natural que compõe esse grupo. Tais palavras são retiradas dos textos após a tokenização visando reduzir o esforço computacional, quando se quer extrair informações de um texto. Cabe salientar que em alguns casos como processamento de textos curtos, a retirada dos *stopwords* pode levar à perda de informações relevantes para o significado do texto. O vocabulário representa o conjunto de palavras (*tokens*) que será usado no processamento do texto

O *Stemming* é uma técnica usada para remover de variações de plural, gerúndio

e sufixos temporais, substituindo por suas raízes morfológica. Por exemplo as palavras *"analysis","analyzed", "analyzing"* iram ser reduzidas para a palavra *analysi*.O maior propósito é melhorar o processamento das palavras, não recomendado para nomear ou analisar as entidades nomeadas como nome próprio. (HAPKE; LANE; HOWARD, 2019). Assim o tamanho do vocabulário pode implicar diretamente na complexidade computacional e na memória requerida para o devido processamento. O uso de técnicas que reduzam o vocabulário é imprescindível para o ganho de performance bem como pode proferir maior generalidade ao processamento. Tais técnicas buscam transformar diversas palavras com significados semelhantes em uma só. Uma dessas técnicas é converter todas as letras do texto para minúsculas. Por ser muito comum palavras iniciadas com letra maiúscula, ter o mesmo significado da mesma com a letra inicial em minúsculo.

3.5 Representação Textual

Nessa sessão será abordado o passo mais importante no processamento de linguagem natural (PLN). A representação do texto precisa ser adequada ao algoritmos, possibilidando e facilitando a extração das informações de entrada para o modelo.

A ação mais importante no Processamento de Linguagens Naturais (PLN) é a conversão do texto, em uma representação legível aos algoritmos computacionais, possibilitando a extração de informação contidas nos textos, por meio de ferramentas tecnológicas.

Uma forma de representar o texto é por meio do Modelo de Espaço Vetorial (Vector Space Model - VSM), proposto por (SALTON; WONG; YANG, 1975), consiste em representar um documento por meio de um vetor formado por um ou mais índices de *tokens* (termos), quer dizer cada *token* é representado por uma posição no vetor (índice), e o conteúdo da posição é preenchido com pesos, ou somente 0 e 1, onde zero significa que o termo não está presente no documento. A dimensionalidade do vetor de representação do documento se dá na quantidade de termos, logo o tamanho dos vetores é a mesma da quantidade de palavras contidas no vocabulário.

Esse modelo de representação também conhecido como "saco de palavras" *(Bag of Words - BOW)*, não leva em consideração a posição em que os termos ocorrem, não trazendo informações de relacionamento semântico entre palavras. Pois é formada uma matriz, onde cada linha representa um documento e as colunas representam os tokens. Uma forma de atribuir pesos às palavras é contando quantidade de ocorrência (frequência) de uma palavra em um documento (Term Frequency - TF), de forma simplificada quer dizer quanto maior a frequência de uma palavra mais relevante ela é para o significado do texto.

Essa abordagem simples de contagem de palavras pode atribuir relevância a palavras muito comuns que não contribuem para o significado do texto, que naturalmente possuem uma frequência muito alta, embora a retirada das stopwords mitigue esse problema. Uma outra abordagem é usar um fator que reduza o grau de relevância (peso) de um token com base na análise da frequência do termo em uma coleção de documentos (SALTON; BUCKLEY, 1988), fator esse conhecido como inverso da frequência do documento (Inverse Document Frequence - IDF). Esse fator varia inversamente de acordo o número de documentos n que contém o termo e relação ao total de documentos da coleção.

$$idf_{i,d} = \lg \frac{n}{N} \tag{3.1}$$

Neste contexto, se obtém relevância do termo (tf-idf), por meio do produto da frequência do termo (TF) e o fator IDF, assim os termos mais relevantes serão aqueles com maior ocorrência em um documento e menor ocorrência no restante da coleção de documentos N.

$$tfidf_i = tf_i * \lg \frac{n}{N} \tag{3.2}$$

Representação do texto usando o modelo de espaço vetorial são adequadas para classificação de texto, pois representam matematicamente os documentos em vetores, onde os sistemas computacionais são capazes aferir similaridade entre documentos por meio do ângulo formado entre pares de vetores. Possibilitando o uso de medidas de similaridades como cosseno, ou são acessadas diretamente por diversos algoritmos de aprendizado de máquinas, que fazem uso de medidas de similaridades próprias.

3.6 Representação Distribuída

Esse tipo de representação foi mostrada de uma forma que os dos textos podessem condiderar as frequências das palavras presentes nos documentos para composição de vetores. Tais vetores desconsideram a ordem das palavras. A reapresentação distribuída gera um vetor de representação de uma palavra com base no contexto em que ela é inserida, palavras que ocorrem em contextos similares tendem a possuir significados próximos. Em 2013 foi desenvolvido o algoritmo word2vec (MIKOLOV et al., 2013) que cria representações vetoriais distribuídas chamadas *word embedding*, capazes de representar palavras, considerando as relações sintáticas e semânticas de cada palavra em relação ao vocabulário e independentemente do idioma dos textos, dando mais flexibilidade ao processamento dos dados (AGUIAR, 2016). Tais vetores

são gerados usando redes neurais que fazem uso de uma camada oculta do algoritmo *backpropagation* para atualizar os pesos dessa camada, que dizer gera um vetor por meio de aprendizagem de máquina capaz de capturar propriedades linguísticas indiretamente.

A implementação do algoritmo gera dois modelos de representação do vetor. Um produz o vetor com as dimensões pré-definidas, considerando o contexto, buscando informar qual seria a palavra faltante, este é conhecido com Continuous Baf-of-Word (Cbow) (AGUIAR, 2016) enquanto o *Continuous Skip-Gram*por meio de uma palavra, busca informar qual seria o contexto (MIKOLOV et al., 2013).

Quando usamos o texto com o (WILSON; WIEBE; HOFFMANN, 2005) é apresentado uma maior gama de informações quando comparadas com representações que fazem uso de contagem de frequência de palavras, sendo verdade até mesmo em comparação com outros modelos que usam parâmetros de compensação para efeitos de frequência das palavras.

3.6.1 Análise de Sentimentos

A análise de sentimentos é um campo da NPL (Natural Processing Language) responsável por ligar o estado emocional de um sujeito sobre um algo assunto (ALMATRAFI; PARACK; CHAVAN, 2015). Também conhecida como Mineração de Opiniões, é uma subárea da mineração de texto que processa computacionalmente as emoções, atitudes em relação a um domínio. Alguns autores também chamam opinião, mineração de sentimentos, análise de subjetividade, análise de afeto, análise de emoção, revisão de mineração. Logo é bastante comum mencionar a palavra opinião no lugar se sentimento. Ambos nesse contexto possuem o mesmo significado. Diante do nosso cotidiano temos regularmente problemas de escolha, como restaurantes, hotéis ou até mesmo escolher um filme. Assim o objetivo final da AS é encontrar as opiniões, identificar sentimentos que possam expressar e classificar binariamente. Assim como monstra a figura abaixo.

Conforme a figura acima, temos a AS considerado como um processo de classificação.

Podemos classificar a AS em 3 níveis de granularidade: no nível do documento, observando o sentimento global do texto, nível de sentença, classificando a polaridade de cada sentença no texto e no aspecto ou característica, onde se baseia nos atributos do objeto. No nível de documento tem como objetivo classificar um documento de opinião como expressando uma opinião ou sentimento positivo ou negativo. Ele considera todo o documento uma unidade básica de informação. Ao nível de sentença classificamos os sentimentos expressos em cada sentença. O primeiro passo é identificar se

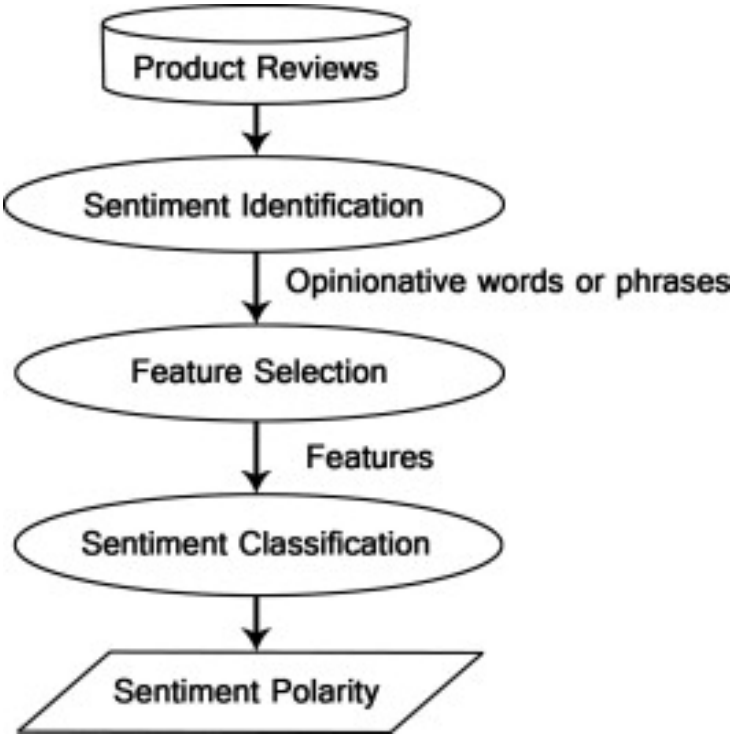

Figura 7 – Análise de Sentimentos no processo de reviews de produtos.
(MEDHAT; HASSAN; KORASHY, 2014)

a sentença é subjetiva ou objetiva. Se a sentença for subjetiva, a AS no nível da sentença determinará se a sentença expressa opiniões positivas ou negativas. (WILSON; WIEBE; HOFFMANN, 2005) apontaram que expressões de sentimentos não são necessariamente de natureza subjetiva. No entanto, não há diferença fundamental entre as classificações em nível de documento e sentença, porque as sentenças são apenas documentos curtos. A classificação do texto no nível do documento não fornece os detalhes necessários, opiniões necessárias sobre todos os aspectos da entidade, necessários em muitos aplicativos. Para obter esses detalhes precisamos ir para o nível do aspecto. No nível de aspecto, para classificar o sentimento em relação aos aspectos específicos das entidades, o primeiro passo é identificar as entidades e seus aspectos. Os detentores de opinião podem dar opiniões diferentes para diferentes aspectos da mesma entidade, como esta frase ”O conforto desse carro é excelente, mas seu consumo é muito alto”.

Um problema também é o fator cultural, nuances linguísticas e contextos diferentes tornam extremamente difícil transformar uma sequência de texto escrito em um sentimento com viés negativo ou positivo. Abaixo segue as técnicas para a classificação na análise de sentimentos.

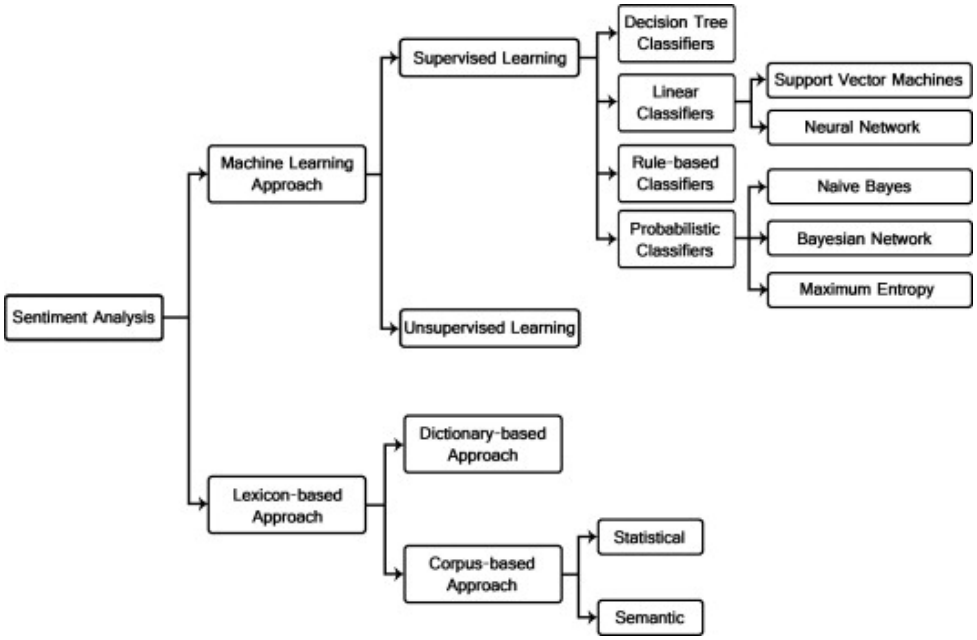

Figura 8 – Técnicas em Análise de Sentimentos.
(MEDHAT; HASSAN; KORASHY, 2014)

3.7 Métodos de Avaliação

As medidas mais comuns para a avaliação dos resultados da classificação de texto são: abrangência (recall), precisão (precision), exatidão (accuracy) e error. Sendo as medidas de precisão e abrangência as mais usuais e indicadas para classificação de texto.

Precisão é a razão entre as observações de verdadeiros positivos previstos corretamente e o total de observações julgadas como positivas (VP + FP), ou seja:

$$Preciso = \frac{VP}{VP + FP}$$

- VP : Verdadeiro Positivo
- FP : Falso Positivo

A acurácia é a medida de desempenho mais intuitiva (BIRD; KLEIN; LOPER, 2009) é simplesmente uma proporção do resultado previsto corretamente com o total de observações do conjunto. Quanto maior nossa *accuracy*, melhor estará nosso modelo. Esta é uma ótima referencia ao nosso *dataset* uma vez que ele se encontra balanceado com os falsos positivos e os falsos negativos. Segue abaixo equação representativa:

$$Acurcia = \frac{VP + VN}{VP + FP + FN + VN}$$

- VP : Verdadeiro Positivo
- FP : Falso
- VN : Verdadeiro Negativo
- FN : Falso Negativo

Também chamado de Sensibilidade nos casos para classificação binário, o *recall* se define como medir a capacidade do modelo em saber a proporção de VP (verdadeiros positivos) que conseguimos acertar de todas os documentos da classe real, em outras palavras, quão bom meu modelo é para prever apenas os positivos.

Portanto é definido como a razão entre verdadeiros positivos sobre a soma de verdadeiros positivos com negativos falsos.

$$Recall = \frac{VP}{VP + FN}$$

- VP : Verdadeiro Positivo
- FN : Falso Negativo

Mesmo com as medidas de precisão e recall sendo de grande utilidade para avaliações de desempenho de classificadores, às vezes se faz necessária uma medida única, para que se possa comparar de forma direta dois ou mais classificadores. E essa é a ideia da medida Fβ que é uma medida baseada na média harmônica dos valores de precisão e recall.

$$F\beta = \frac{(1 + \beta) * Preciso * Recall}{\beta 2 * Preciso + Recall} \tag{3.3}$$

Na fórmula apresentada percebe-se que dependendo do valor da constante β, a medida Fβ estará dando maior importância para a medida de precisão (0 < β < 1) ou para a medida de recall (β > 1). Um caso particular e muito utilizado dessa medida F é a medida F1, com β = 1, isto é, a medida F que dá a mesma importância para Precisão e Recall, resultando na fórmula:

$$F1 = \frac{2 * Preciso * Recall}{Preciso + Recall}$$

4 Metodologia

A partir da revisão bibliográfica e trabalhos relacionados, foi adquirido conhecimento para planejar e executar todo o método e pesquisa adotado por esse trabalho.

Resumidamente temos as trés etapas apresentados abaixo:

1. Definição do banco de dados experimental. Com definição dos parâmetros, linguagem de programação e as variáveis.
2. Escolha dos algoritmos para modelagem de treinamento e teste.
3. Realização dos experimentos e análise dos resultados registrados.

Essa solução propõe ajudar as empresas no processo de análise de sentimentos e opinião sobre os seus produtos e serviços.Todo o sistema desenvolvido é capas de extrair as informação do banco de dados, processar o texto oriundo do site IMDB e alimentar o algoritmo de aprendizagem de maquina, previamente treinado e sugerindo uma classificação ao texto. O problema é de classificação de texto. E o modelo de aprendizado foi supervisionado.Na primeira etapa a linguagem Python se mostra na atualidade como uma boa alternativa de bibliotecas para aprendizado de maquina. Em novo caso iremos usar o pandas e a sklearn.

O projeto desenvolvido está basicamente divido em duas parte: Preprocessamento e Treinamento.

No pre processamento é feita toda a leitura dos *reviews* onde para o estudo, devido a quantidades de arquivos, foi dividido por lotes *batchs*. Assim para estudo houve um processamento parcial. Na assinatura da função está definido como *data size*.

Segue função abaixo na figura 9:

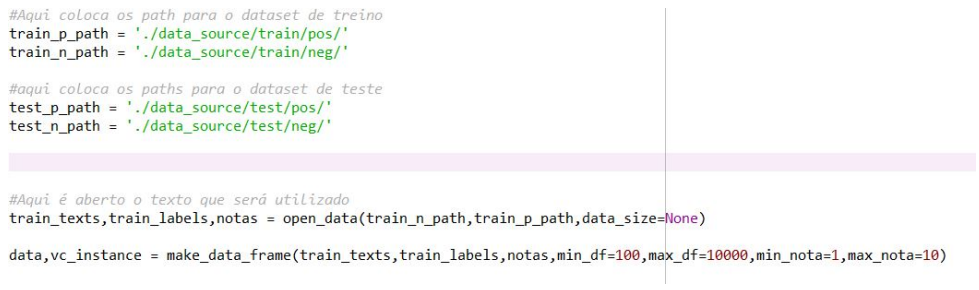

Figura 9 – Função de separação do dados de Treino e Teste

Assim, devido ao método de criação do *dataframe* no pandas, foi definido mais umas variável, para definir a quantidade de arquivos que será processado no treinamento. O dados de texto estão balanceados, essa definição será a mesma para o grupo de treinamento e teste.

4.1 O Corpus

O banco de dabos usado para o estudo (MAAS et al., 2011a) possui binariamente as *labels* de sentimentos polarmente associados aos *reviews*. Com objetivo de servir como referência para classificação de intenções, opiniões e emoções.

O corpus é composto por esses comentários de 50.000 análises divididas igualmente em um conjuntos de testes, 25.000 negativos e 25.000 positivos. A distribuição dos dados já está balanceada quanto a polaridade e todos pertencem a um mesmo domínio de conhecimento.

Rotulamos negativamente todos os comentários de treino para teste que tivesse um $score(nota) \leq 4$ no total de 10. Assim as análises mais neutras não estão incluídas nos conjuntos de treino de teste. Uma avaliação de um usuário com uma pontuação >= 7 no total de 10 será considerada positiva.

Na figura 10 abaixo temos a distribuição de todas as notas do banco de dados.

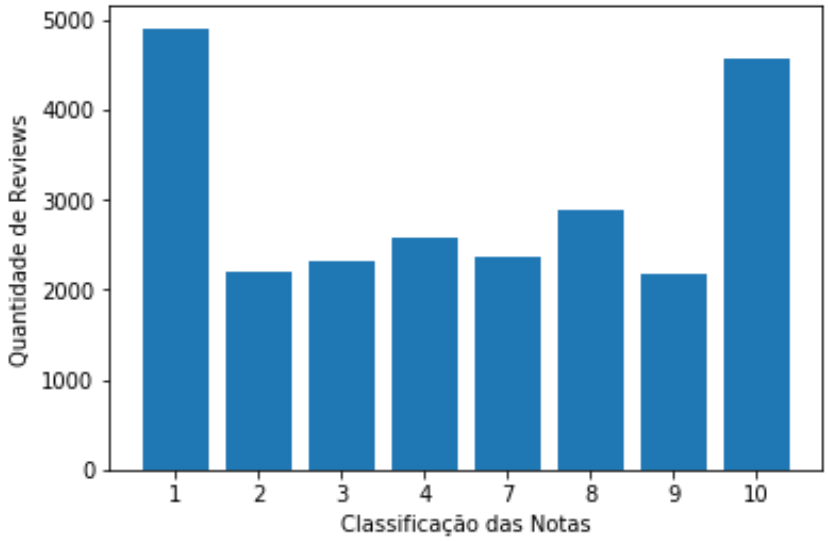

Figura 10 – Gráfico das notas por classificação X Quantidade de *reviews*

Existem dois diretórios de nível superior [*train e test*] correspondentes a os conjuntos de treinamento e teste. Cada um contém diretórios [pos, neg] para as revisões dos *labels* binários positivas e negativas. Dentro destes diretórios, as revisões são armazenadas em arquivos de texto nomeados após da seguinte maneira [[id][*rating*].txt]

em que [id] é um ID exclusivo e [*rating*] é a classificação por estrelas para essa revisão em uma escala de 1 a 10.

A decisão foi baseada no trabalho de (MAAS et al., 2011b). Onde o mesmo usou um dataset do IMDB com cerca de 30 *reviews* onde o mesmo usou a metodologia de bag of word e SVM para classificar os documentos. Um *review* negativo tem um classificação de $\leq$ 4 de 10 e um *review* positivo tem um score de $\geq$ 7 de 10.

No nosso trabalho fico organizado da seguite forma: o arquivo [test / pos / 200_8.txt] é o texto para um conjunto de testes com etiqueta positiva exemplo com identificação única 200 e classificação por estrela 8/10 da IMDb.O diretório [[*train*] / unsup /] possui 0 para todas as classificações porque as classificações são omitido para esta parte do conjunto de dados.

Segue figura de como está disposto o ambiente de desenvolvimento quanto ao organização dos diretórios.

- data source
  - test
    - neg
    - pos
  - train
    - neg
    - pos

4.2 Pré-Processamento e Processamento Dos Dados

Como estamos tratando um conjunto de dados não estruturados essa etapa se torna importantíssima quando utilizamos texto pra isto. Este passo no nosso trabalho é relativamente simples devido a configuração do framework *Sklearn*, e a corpus na língua inglesa estarem programados no pacote do módulo.

No trecho de código abaixo temos o método principal para o a normalização dos nossos textos. Usamos a biblioteca do *CountVectorizer* que é onde se concentra a maior parte de nossos parâmetros de configuração do texto.

A saída dessa função é um arquivo de estrutura de *dataframe*. Esse **dataframe** esta composto nas colunas pelas *features*, que são as palavras, as linhas são os documentos. Assim na interjeição da matriz em [i][j] temos um contador palavra X comentário se existe ou não no *dataframe*. Ex: Na Figura 13, temos que o texto Número 40 possuem a palavra Xero e a nota classificada para ele foi 1. Sua Label é 0 ou seja, Negativa.

```
def make_data_frame(texts,labels,notas,min_df=100,max_df=10000,min_nota=1,max_nota=10,cv_istance=None):

    if cv_istance == None:
        vc = CountVectorizer(min_df=min_df,max_df=max_df)
        word_vectors = vc.fit_transform(texts).toarray()
    else:
        vc = cv_istance
        word_vectors = vc.transform(texts).toarray()

    df = pd.DataFrame(word_vectors,columns=vc.get_feature_names())
    df['labels'] = labels
    df['notas'] = notas


    if min_nota == None and max_nota != None:
        df = df[df['notas'] >= max_nota]
    elif max_nota == None and min_nota != None:
        df = df[df['notas'] <= min_nota]
    elif max_nota != None and  min_nota != None:
        df = df[(df['notas'] <= min_nota) | (df['notas'] >= max_nota)]
    else:
        pass

    return df,vc
```

Figura 11 – Funcão Data Frame Maker

Console 1/A

	000	10	100	11	12	13	...	zero	zombie	zombies	zone	labels	notas
3	0	0	0	0	0	0	...	0	0	0	0	0	1
4	0	0	0	0	0	0	...	0	0	0	0	0	1
8	0	0	0	0	0	0	...	0	0	0	0	0	1
10	0	0	0	0	0	0	...	0	0	0	0	0	1
14	0	0	0	0	0	0	...	0	0	0	0	0	1
15	0	0	0	0	0	0	...	0	0	0	0	0	1
26	0	0	0	0	0	0	...	0	0	0	0	0	1
28	0	0	0	0	0	0	...	0	0	0	0	0	1
30	0	1	0	0	0	0	...	1	0	0	0	0	1
32	0	1	1	0	0	0	...	0	0	0	0	0	1
34	0	0	0	0	0	0	...	0	0	0	0	0	1
37	0	0	0	0	0	0	...	0	0	0	0	0	1
38	0	0	0	0	0	0	...	0	0	0	0	0	1
39	0	0	0	0	0	0	...	0	0	0	0	0	1
40	0	0	0	0	0	0	...	1	0	0	0	0	1
41	0	0	0	0	0	0	...	0	0	0	0	0	1
43	0	2	0	0	0	0	...	0	0	0	0	0	1
46	0	0	0	0	0	0	...	0	0	0	0	0	1
47	0	0	0	0	0	0	...	0	0	0	0	0	1
48	0	0	0	0	0	0	...	0	0	0	0	0	1
49	0	0	1	0	0	0	...	0	0	0	0	0	1
50	0	0	0	0	0	0	...	0	0	0	0	0	1
51	0	0	0	0	0	0	...	0	0	0	0	0	1
52	0	0	0	0	0	0	...	0	0	0	0	0	1
54	0	0	0	0	0	0	...	0	0	0	0	0	1
60	0	0	0	0	0	0	...	1	0	0	0	0	1
63	0	1	0	0	0	0	...	0	0	0	0	0	1
64	0	0	0	0	0	0	...	0	0	0	0	0	1
67	0	0	0	0	0	0	...	0	0	0	0	0	1
69	0	0	0	0	0	0	...	0	0	0	0	0	1
70	0	0	0	0	0	0	...	0	0	0	0	0	1

Figura 12 – Matriz das *features* e Documentos

Outra análise que executamos foi relacionada com o *Corpus*, onde verificamos qual o acurácia dos algoritmos de acordo com as extremidades das notas dos arquivos.

Ex: Usando os dados apenas dos piores e melhores comentários do nosso *dataset*, irá automaticamente melhorar minha precisão nos dados?

Min Nota	Max Nota	Classificador	Score
1,2	9,10	NB	80
3,4	7,8	NB	81
1,2	9,10	RB	84
1,2,3,4,5	7,8,9,10	NB	88
1,2,3,4,5	7,8,9,10	RF	83

Tabela 1 – Tabela Análise de Notas e Classificador.
fonte: Produzida pelo Autor

Após essa análise executamos a exclusão das colunas *Labels* e notas. Ver trecho do codigo abaixo:

Ambos os algoritmos escolhidos, foram exemplos de base na pesquisa de (FERNÁNDEZ-DELGADO et al., 2014). Onde para o Naive Bayes o algoritmo alcançou uma acurácia de 92.3% e o Random Forest como o melhor resultado de 94.1%. Tudo isso sobre 121 *datasets*.

Com isto todo o processo se torna automático pois o Corpus (vocabulário) para uma análise mais robusta. O peso da matriz dos modelos podem ser usada para qualquer outro domínio.

A configuração final com as parametrizações do processamento do texto está descrita abaixo:

- *Stemização* é o processo de produção de variantes morfológicas de uma palavra raiz / base onde temos como ”Diminuir”as palavras. Para nosso Trabalho, verificamos que houve uma perda significativa no vocabulário certa de 1/3. E também houve perda de aproximadamente 2% na acuraria dos dois algoritmos. A variável *Vocab_* foi processada com a função de stemização. Segue abaixo o resultado final após a stemização de 2791 palavras. Optamos por não usar.

  In[102]: vocab.count()https://www.overleaf.com/project/5d51775e5442786f9d1ac387
  Out[103]: 3791

  In[102]: vocab.count() Out[103]: 2791

- O *Lowercase* - Está habilitado para *True* onde todas as palavras é transformada em caixa baixa para não haver distinção entre palavras.

- O *Ngram* se refere a quantidade e sequencia de palavras que será analisada no processamento. No estudo fizemos o *Ngram* (1,1). Ou seja, analisamos apenas uma palavra.
- *Stop Word* é uma lista de palavras que podem ser excluídas do treinamento através desse parâmetro. Apesar do uso dessa técnica, ela não apresentou diferença significativa para a precisão do nosso modelo.

```
Out[49]:
CountVectorizer(analyzer='word', binary=False, decode_error='strict',
                dtype=<class 'numpy.int64'>, encoding='utf-8', input='content',
                lowercase=True, max_df=10000, max_features=None, min_df=100,
                ngram_range=(1, 1), preprocessor=None, stop_words=None,
                strip_accents=None, token_pattern='(?u)\\b\\w\\w+\\b',
                tokenizer=None, vocabulary=None)
```

Figura 13 – Parâmetros do processamento de Texto

Para o treinamento do vector de texto segue abaixo a imagem 18 sobre o processo com o Random Forest.

train_p_path = ’./data_source/train/pos/’ train_n_path = ’./data_source/train/neg/’

test_p_path = ’./data_source/test/pos/’ test_n_path = ’./data_source/test/neg/’

train_texts,train_labels,notas = open_data(train_n_path,train_p_path,data_size=12500)

data,vc_instance = make_data_frame(train_texts,train_labels,notas,min_df=100,max_df=1

y_data = data[’labels’] x_data = data.drop([’notas’,’labels’],axis=1)

rf = RandomForestClassifier(n_estimators=200,criterion=’gini’) rf.fit(x_data,y_data)

test_texts,test_labels,test_notas = open_data(test_n_path,test_p_path,data_size=12500)

test_data,cv_instance_2 = make_data_frame(test_texts,test_labels,test_notas, max_df=0.85,min_nota=None,max_nota=None, cv_istance=vc_instance)

y_test = test_data[’labels’] x_test = test_data.drop([’notas’,’labels’],axis=1)

preds = rf.predict(x_test) preds_2 = rf.predict(x_data)

Usamos o X_data e o Y_data para treinar os dados. Todos eles ja passaram pelo processo de tokenização e pela função ”to_array”, para ser tranformado em vetores. A função ”predict”linha 53 está com parâmetro as informações em X Test. Onde os dados não estão classificados.

O processo é bem semelhante quando ao algorimo do Naive Bayes. Nele temos as mesmas funções *”fit()”* (treinamento) e *””predict()”* (estimar).

gnb = GaussianNB() gnb.fit(x_data,y_data)

```
test_texts,test_labels,test_notas = open_data(test_n_path,test_p_path,data_size=12500)

test_data,cv_instance_2 = make_data_frame(test_texts,test_labels,test_notas,min_df=100
max_df=10000,min_nota=None,max_nota=None, cv=vc_instance)

y_test = test_data['labels'] x_test = test_data.drop(['notas','labels'],axis=1)

preds = gnb.predict(x_test)

preds_2 = gnb.predict(x_data)

train_vectors = cv_instance_2.transform(train_texts).toarray()

train_vectors,val_vectors,train_labels,val_labels = train_test_split(train_vectors,train_labels

test_vectors = cv_instance_2.transform(test_texts).toarray()

vocab = pd.Series(cv_instance_2.vocabulary_)

preds = gnb.predict(test_vectors)
```

5 Resultados

Como apresentado nos capitulos anteriores seguem os indicadores de performance para os algoritmos de aprendizagem abaixo:

O resultado do score para o Naive Bayes foi de 84% de Acuracia. O algoritmo Naive Bayes teve um resultado inferior, aproximadamente 76%

classe	precision	recall	f1-score	support
0	0.84	0.83	0.84	12500
1	0.84	0.84	0.84	12500

Tabela 2 – Matriz de Confusão.
fonte: Produzida pelo Autor

Quanto a precisão dos resultados a variação foi pequena. O pior resultado foi a precisão dos resultados para a classe 0 (negativa) no algoritmo de Navie Bayes com 74%.

classe	precision	recall	f1-score	support
0	0.74	0.81	0.77	12500
1	0.79	0.71	0.75	12500

Tabela 3 – Matriz de Confusão.
fonte: Produzida pelo Autor

Nas Matrizes de confusão vemos que o Modelo de Random Forest possui certa de 20% das frequencias que erram a classe predita. Ex: O maior número foi de 2057 de casos onde a classe era 0 (negativa) e o modelo previu quer era da classe 1, ou seja verdadeiro positiva.

Assim, temos algumas observação nosso modelo no Random Florest:

- O modelo previu 10443 vezes a categoria ”0”(Sentimento negativo).
- O modelo previu 10474 vezes a categoria do dominio ”1”(Sentimento positivo)

$$\begin{vmatrix} 10443 & 2057 \\ 2026 & 10474 \end{vmatrix}$$

Tal estrutura é importante para verificação dos falsos positivos e falsos negativos no contexto do problema considerado. No caso da Naive Bayes quantidade dos FP e FN é mais relevante quanto ao algoritmo anterior. Segue abaixo:

$$\begin{vmatrix} 101443 & 2356 \\ 3595 & 8905 \end{vmatrix}$$

- O modelo previu 3595 vezes a categoria ”1”, (Sentimento positivo). Quando na verdade deveria ser a categoria 0 (Sentimento negativo)
- O modelo previu 2356 vezes a categoria 0 (Sentimento Negativo) onde o correto deveria ser da classe positiva

6 Conclusão

Neste trabalho estudamos alguns modelos para a classificação. Nossas investigações consistiu em dois modelos com resultados satisfatórios. O random forest e o Naive Bayes. Em ambos os casos aplicamos a técnica de bag of words e o modelo word2vec para representar as palavras numericamente. O grande desafio é agregar as palavras os *word vector* (vetores de palavras) em uma *feature* vector para cada *review*(comentário).

Concluímos também que técnicas de mineração de texto e análise de dados se mostraram eficientes para classificar textos com relação à sua orientação, uma vez que foram atingidas medidas de desempenho similares à outros trabalhos da área. Vale ressaltar que a análise de sentimento em textos em português é ainda pouco estudada e por isso considera-se que este foi um estudo relevante para a literatura.

As principais contribuições deste trabalho são:

1. A baixa eficacia do pre-processamento de texto na análise das opiniões desse domínio
2. Obtenção de um corpus na língua portuguesa
3. A descoberta de algoritmos no processo de analise de processamento de linguagem natural

6.1 Trabalhos Futuros

1. Analisar o contexto usando um Ngram maior. Ex Ngram (2,1) ou (3,1)
2. Investigar o motivo pelo qual o pré-processamento não demonstrou significativa melhora na performance dos classificadores.
3. A descoberta de algoritmos no processo de analise de processamento de linguagem natural
4. Analisar e executar um estudo na área do seleção de palavras para saber o real peso e influencia real delas.

Referências

ACHREKAR, H. et al. Predicting flu trends using twitter data. In: IEEE. *2011 IEEE conference on computer communications workshops (INFOCOM WKSHPS)*. [S.l.], 2011. p. 702–707. Citado na página 17.

AGUIAR, E. M. d. *Aplicação do Word2vec e do Gradiente descendente dstocástico em tradução automática*. Tese (Doutorado), 2016. Citado 2 vezes nas páginas 27 e 28.

ALMATRAFI, O.; PARACK, S.; CHAVAN, B. Application of location-based sentiment analysis using twitter for identifying trends towards indian general elections 2014. In: ACM. *Proceedings of the 9th international conference on ubiquitous information management and communication*. [S.l.], 2015. p. 41. Citado 2 vezes nas páginas 24 e 28.

ARAVINDAN, S.; EKBAL, A. Feature extraction and opinion mining in online product reviews. In: IEEE. *2014 International Conference on Information Technology*. [S.l.], 2014. p. 94–99. Citado na página 15.

BIRD, S.; KLEIN, E.; LOPER, E. *Natural language processing with Python: analyzing text with the natural language toolkit*. [S.l.]: ”O’Reilly Media, Inc.”, 2009. Citado na página 30.

BOLLEN, K. A.; DAVIS, W. R. Causal indicator models: Identification, estimation, and testing. *Structural Equation Modeling: A Multidisciplinary Journal*, Taylor & Francis, v. 16, n. 3, p. 498–522, 2009. Citado na página 16.

BREIMAN, L. Random forests. *Machine learning*, Springer, v. 45, n. 1, p. 5–32, 2001. Citado na página 22.

COELHO, U. M.; LIMA, A. C. E.; OMAR, N. Analisador de expressões positivas e negativas aplicado em comentários de livros e filmes. In: *Simposio Argentino de GRANdes DAtos (AGRANDA)-JAIIO 46 (Córdoba, 2017)*. [S.l.: s.n.], 2017. Citado na página 15.

DAVIDOV, D.; TSUR, O.; RAPPOPORT, A. Enhanced sentiment learning using twitter hashtags and smileys. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the 23rd international conference on computational linguistics: posters*. [S.l.], 2010. p. 241–249. Citado na página 17.

FERNÁNDEZ-DELGADO, M. et al. Do we need hundreds of classifiers to solve real world classification problems? *The Journal of Machine Learning Research*, JMLR. org, v. 15, n. 1, p. 3133–3181, 2014. Citado na página 36.

GONÇALVES, L. P. F. Avaliação de ferramentas de mineração de dados como fonte de dados relevantes para a tomada de decisão: aplicação na rede unidão de supermercados, são leopoldo-rs. 2001. Citado na página 25.

HAPKE, H. M.; LANE, H.; HOWARD, C. *Natural language processing in action*. [S.l.]: Manning, 2019. Citado 2 vezes nas páginas 25 e 26.

LIU, B. et al. Sentiment analysis and subjectivity. *Handbook of natural language processing*, v. 2, n. 2010, p. 627–666, 2010. Citado na página 11.

LIU, K.-L.; LI, W.-J.; GUO, M. Emoticon smoothed language models for twitter sentiment analysis. In: *Twenty-sixth aAAI conference on artificial intelligence*. [S.l.: s.n.], 2012. Citado 2 vezes nas páginas 15 e 17.

LONGHI, M. T.; BERCHT, M.; BEHAR, P. A. Reconhecimento de estados afetivos do aluno em ambientes virtuais de aprendizagem. *RENOTE: revista novas tecnologias na educação [recurso eletrônico]. Porto Alegre, RS.*, 2007. Citado na página 15.

LORENA, A. C.; CARVALHO, A. C. de. Uma introdução às support vector machines. *Revista de Informática Teórica e Aplicada*, v. 14, n. 2, p. 43–67, 2007. Citado 4 vezes nas páginas 18, 20, 21 e 23.

MAAS, A. L. et al. Learning word vectors for sentiment analysis. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, 2011. p. 142–150. Disponível em: <http://www.aclweb.org/anthology/P11-1015>. Citado na página 33.

MAAS, A. L. et al. Learning word vectors for sentiment analysis. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*. [S.l.], 2011. p. 142–150. Citado na página 34.

MEDHAT, W.; HASSAN, A.; KORASHY, H. Sentiment analysis algorithms and applications: A survey. *Ain Shams engineering journal*, Elsevier, v. 5, n. 4, p. 1093–1113, 2014. Citado 2 vezes nas páginas 29 e 30.

MENGER, V. et al. Deduce: A pattern matching method for automatic de-identification of dutch medical text. *Telematics and Informatics*, Elsevier, v. 35, n. 4, p. 727–736, 2018. Citado na página 15.

MIKOLOV, T. et al. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. Citado 2 vezes nas páginas 27 e 28.

MORO, S.; LAUREANO, R.; CORTEZ, P. Using data mining for bank direct marketing: An application of the crisp-dm methodology. In: EUROSIS-ETI. *Proceedings of European Simulation and Modelling Conference-ESM'2011*. [S.l.], 2011. p. 117–121. Citado na página 19.

PAK, A.; PAROUBEK, P. Twitter based system: Using twitter for disambiguating sentiment ambiguous adjectives. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the 5th International Workshop on Semantic Evaluation*. [S.l.], 2010. p. 436–439. Citado na página 16.

PANG, B.; LEE, L. et al. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, Now Publishers, Inc., v. 2, n. 1–2, p. 1–135, 2008. Citado na página 11.

PERKINS, J. *Python text processing with NLTK 2.0 cookbook*. [S.l.]: Packt Publishing Ltd, 2010. Citado na página 25.

PROVOST, F.; KOHAVI, R. On applied research in machine learning. *MACHINE LEARNING-BOSTON-*, Citeseer, v. 30, p. 127–132, 1998. Citado 2 vezes nas páginas 19 e 20.

SAKAKI, T.; OKAZAKI, M.; MATSUO, Y. Earthquake shakes twitter users: real-time event detection by social sensors. In: ACM. *Proceedings of the 19th international conference on World wide web*. [S.l.], 2010. p. 851–860. Citado na página 17.

SALTON, G.; WONG, A.; YANG, C.-S. A vector space model for automatic indexing. *Communications of the ACM*, ACM, v. 18, n. 11, p. 613–620, 1975. Citado na página 26.

WILSON, T.; WIEBE, J.; HOFFMANN, P. Recognizing contextual polarity in phrase-level sentiment analysis. In: *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. [S.l.: s.n.], 2005. Citado 2 vezes nas páginas 28 e 29.