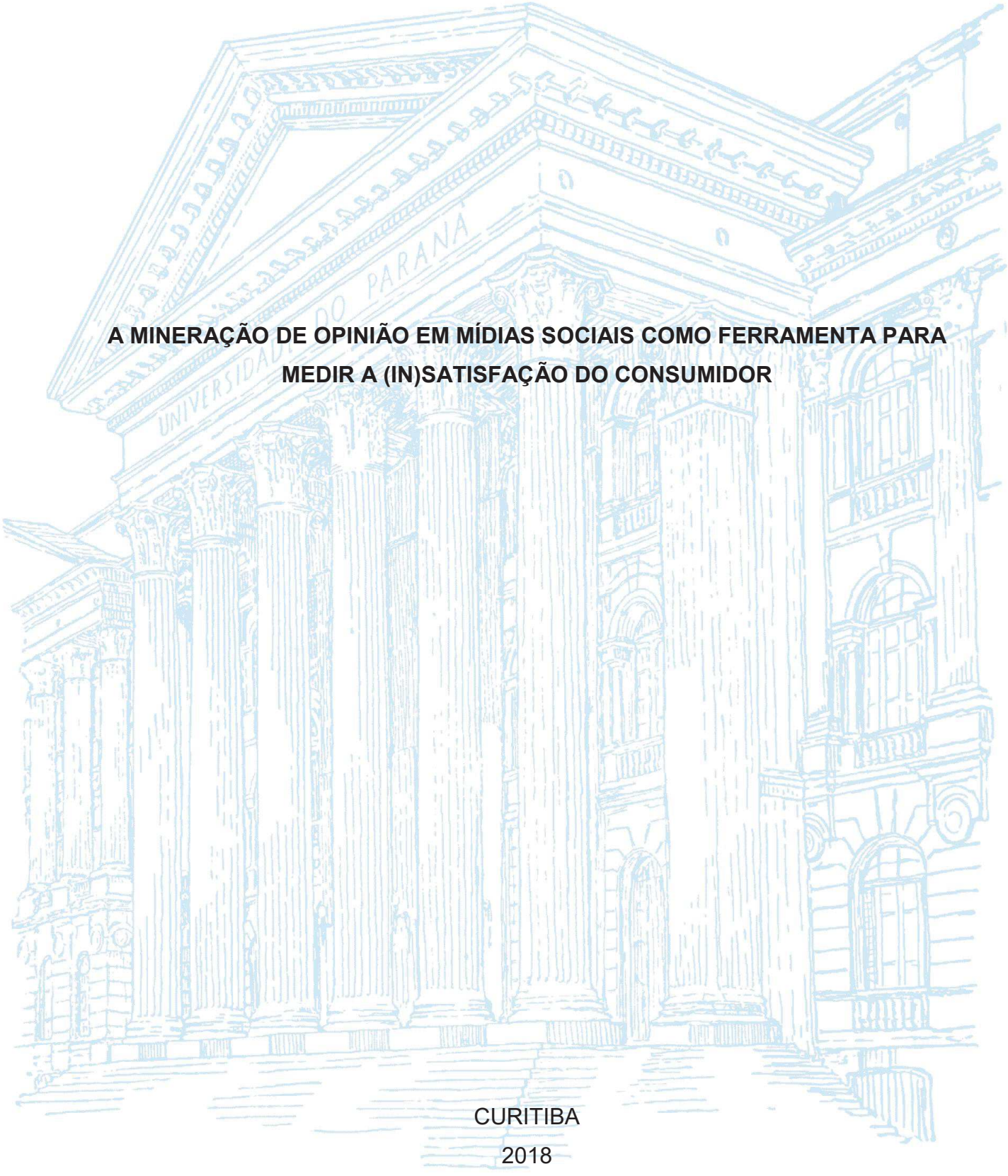
UNIVERSIDADE FEDERAL DO PARANÁ

LUIS SANCLIMENT IGLESIAS



A MINERAÇÃO DE OPINIÃO EM MÍDIAS SOCIAIS COMO FERRAMENTA PARA MEDIR A (IN)SATISFAÇÃO DO CONSUMIDOR

CURITIBA

2018

LUIS SANCLIMENT IGLESIAS

A MINERAÇÃO DE OPINIÃO EM MÍDIAS SOCIAIS COMO FERRAMENTA PARA MEDIR A (IN)SATISFAÇÃO DO CONSUMIDOR

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção de grau de Bacharel no Curso de Gestão da Informação, Departamento de Ciência e Gestão da Informação, do Setor de Ciências Sociais Aplicadas, da Universidade Federal do Paraná.

Orientadora: Prof.ª Dr.ª Denise Fukumi Tsunoda

CURITIBA

2018

“O que não se define não se pode medir,
o que não se mede não se pode melhorar,
o que não se melhora vai degradar sempre.”

William Thomson

RESUMO

Estudo de natureza quantitativa que objetiva explorar a contribuição da mineração de opinião em bases de dados SAC 2.0 extraídas do Facebook para a medição da (in)satisfação dos consumidores. Visa analisar e propor ferramentas que auxiliem nas etapas do processo de descoberta de conhecimento em texto e selecionar as mais adequadas para a mineração de opinião a nível de sentença, onde se analisa o sentimento positivo, negativo e neutro. Propõe desenvolver uma metodologia de análise de opinião, iniciando com a seleção da ferramenta NetVizz para extrair a base de dados do Facebook, seguido do uso do Microsoft Excel®, para seleção e redução de dados, códigos em Python para a fase de limpeza e transformação e a ferramenta Semantria como instrumento de análise de texto. Submete-se para a mineração de opinião a base de dados extraída com quatro tratamentos de pré-processamento. Utilizam-se os algoritmos *Naive Bayes*, SMO e J48 na ferramenta Weka para a etapa de processamento. Apresenta resultados satisfatórios na mineração de opinião com melhor taxa de acerto obtida usando o algoritmo SMO. Propõe trabalhos futuros em bases de dados SAC com a aplicação desta metodologia desenvolvida e estudos de descobrimento das causas de (in)satisfação dos consumidores encontradas em bases de dados SAC e SAC 2.0.

Palavras-chave: Análise de Sentimentos. Mineração de Texto. SAC 2.0. Satisfação do Consumidor.

ABSTRACT

A quantitative study that aims to explore the contribution of opinion mining for customer service 2.0 databases extracted from Facebook in order to measure consumer (in)satisfaction. It aims to analyze and propose tools that assist in the steps of Knowledge Discovery in Text and select the appropriate one for the opinion mining at sentence level, where the positive, negative and neutral feeling is analyzed. It proposes to develop a methodology of opinion analysis, starting with the selection of NetVizz tool to extract a database from Facebook, followed by the use of Microsoft Excel®, for selection and reduction of data, codes in Python for cleaning and transformation step and the Semantria tool as an instrument of text analysis. Submitted for opinion mining test the database extracted with four pre-processing treatments. Naive Bayes, SMO and J48 algorithms are used in the Weka tool for the processing step. It presents satisfactory results in opinion mining with the best-hit rate obtained using the SMO algorithm. It proposes future work in Customer Services databases with the application of this methodology developed and studies on discovery of consumer (in)satisfaction causes found in customer services and customer service 2.0 databases.

Keywords: Sentiment Analysis. Text Mining. Customer Service 2.0. Customer Satisfaction.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

API - *Application Programming Interface*

BDTD – Biblioteca Digital Brasileira de Teses e Dissertações

DCT – Descoberta de Conhecimento em Texto

GI – Gestão da Informação

PLN – Processamento de Linguagem Natural

SAC – Serviço de Atendimento ao Consumidor

SMO – *Sequential Minimal Optimization*

UFPR - Universidade Federal do Paraná

SUMÁRIO

1 INTRODUÇÃO

Com o avanço das tecnologias de comunicação e a globalização mundial em proporções cada vez maiores, as mídias sociais são ferramentas que permitem a interação entre as pessoas (conhecidas ou não). Essas interações acontecem seja para as pessoas poderem se pronunciar quanto a algum tema que tenha despertado seu interesse ou para expressar suas opiniões e sentimentos a respeito dos mais variados assuntos. Estas opiniões podem por exemplo, expressar (in)satisfação sobre os produtos ou serviços consumidos e até mesmo questionar ou buscar informações para a tomada de decisão na aquisição (ou não) de produtos e serviços ainda não conhecidos.

Assim, cria-se uma verdadeira “mina inesgotável” de dados advindos das mídias sociais que são valiosíssimas para as organizações. Com isso, forma-se quase que de maneira natural um canal direto e não tão oneroso como manter um setor de serviço de atendimento ao consumidor para que as empresas que estão atentas e presentes nas diferentes mídias sociais se beneficiem deste “filão” e possam aproveitá-lo e aproximar-se mais de seus consumidores, permitindo não somente medir o grau de (in)satisfação dos mesmos, bem como responder mais rapidamente às solicitações, questionamentos e reclamações, bem como o crescimento das possibilidades de alavancar e conquistar novos clientes.

Para essa nova forma de interação cliente-empresa é que surgiu o SAC 2.0 (Serviço de Atendimento ao Consumidor 2.0) que segundo a agência wck (2018) é uma evolução do SAC tradicional que visa um atendimento mais completo, tendo o consumidor maior interação e voz ativa, de maneira que o problema dele possa ser resolvido com maior rapidez e de forma otimizada. A aplicação da ferramenta SAC 2.0 é mais popular nas mídias sociais, e embora as empresas ainda utilizam o SAC tradicional, o consumidor moderno já está acostumado com uma forma mais rápida de interação.

Uma vez que a quantidade de dados que circula nas mídias sociais é de grande proporção, faz-se necessária a utilização de ferramentas que auxiliem na captura de texto em linguagem natural, para posterior uso da mineração de opinião, e com isso viabilizar a extração de dados relevantes de forma automatizada, sumarizando em resultados que irão permitir uma melhor visualização e tomada de decisão.

Esta pesquisa pretende investigar a eficácia da utilização da mineração de opinião como ferramenta que permita mensurar as manifestações de consumidores quanto às suas opiniões e sentimentos demonstrados em mídias sociais, nas suas expressões de (in)satisfação quanto aos produtos ou serviços oferecidos pelas empresas.

1.1 PROBLEMA DE PESQUISA

A satisfação do consumidor é relevante para as empresas e não resta dúvida que alcançar a fidelidade deles tem impacto direto na existência e perenidade do negócio e é fato que a sua insatisfação é fator preponderante para que as organizações os percam. Para Gerson (2001), fornecer serviços com excelência pode fazer uma diferença crítica no sucesso da organização e em poder manter os seus clientes. Segundo Kotler (2002), para as empresas que estão focadas no cliente, a satisfação dos mesmos é uma meta e também uma ferramenta de marketing. As organizações precisam estar preocupadas com o nível de satisfação do cliente, principalmente porque nos dias de hoje a internet permite que os consumidores opinem e divulguem para o mundo seus elogios e reclamações.

Segundo Schiessl e Bräscher (2011), existem “informações ricas” passadas pelos consumidores de forma textual nas bases de dados de SAC, e com metodologias adequadas, é possível extrair informações úteis dos consumidores ainda não identificadas.

É fato que se encontra uma grande quantidade de dados relevantes e valiosos a respeito dos consumidores nas opiniões e comentários que circulam a todo momento nas mídias sociais. Também se sabe que pela enorme quantidade de informações textuais acumuladas no dia a dia, torna-se praticamente impossível de serem extraídas de forma visual humana sem a intervenção de mecanismos eletrônicos automáticos que permitam agrupar, classificar e interpretar as informações disponibilizadas pelos clientes nas mídias sociais, de forma a poderem ser utilizadas para identificar as principais causas que provocam a (in)satisfação dos consumidores.

Por conseguinte, a questão desta pesquisa é verificar: **como a mineração de opinião em bases de dados extraídas de mídias sociais pode contribuir para a medição da (in)satisfação dos consumidores?**

1.2 OBJETIVOS

Para poder responder à questão de pesquisa exposta, há a necessidade de traçar objetivos geral e específicos, os objetivos específicos são derivados do objetivo geral, e eles darão o norte para se alcançar o objetivo final.

1.2.1 Objetivo Geral

Levando em consideração que a questão de pesquisa é o que se pretende investigar, o objetivo geral é verificar como a mineração de opinião em bases de dados extraídas de mídias sociais pode contribuir na medição da (in)satisfação dos consumidores.

1.2.2 Objetivos Específicos

Os objetivos específicos para poder cumprir com o objetivo geral são:

a) estudar a mineração de opiniões e propor ferramentas que auxiliem na extração e pré-processamento de bases de dados extraídas de mídias sociais;
b) escolher e aplicar ferramentas que possam ser utilizadas para a mineração de opiniões proposta;
c) registrar a metodologia de análise de opiniões utilizada, sumarizando etapas, ferramentas e métodos.

1.3 JUSTIFICATIVAS

Para melhor fundamentar os motivos pelos quais foi selecionada e desenvolvida esta pesquisa, a justificativa foi dividida em cinco diferentes aspectos, conforme apresentado na sequência.

1.3.1 Científica

Do ponto de vista científico este estudo pode contribuir com o tema abordado devido à escassez de artigos e trabalhos encontrados nas bases pesquisadas. Em

pesquisa realizada em 11 de outubro de 2018 na base de periódicos CAPES envolvendo os termos “mineração de opinião”, “mineração de sentimentos”, “análise de opiniões” e “análise de sentimentos” aliados com os termos “mídias sociais”/“redes sociais” e refinado por idioma “português” e tipo de recurso “artigo”, foram encontrados apenas 11 (onze) artigos relacionados, sendo que quatro deles não foram listados pois estavam repetidos na busca dos termos “análise de opiniões” e “análise de sentimentos”. Os resultados obtidos, bem como os temas abordados e o ano de publicação dos trabalhos encontrados podem ser visualizados na Tabela 1.

TABELA 1 – RESULTADOS DE TRABALHOS ENCONTRADOS NA BASE CAPES

Termos Pesquisados	Artigos Encontrados	Temas Abordados nos Artigos
"mineração de opinião" AND "mídias sociais" OR "redes sociais"	1	- Métodos de agrupamento de otimização de enxame para mineração de opinião (2018)
"mineração de sentimentos" AND "mídias sociais" OR "redes sociais"	0	-
"análise de opiniões" AND "midias sociais" OR "redes	4	- Análise de opiniões expressas em redes sociais de filmes lançados nos Estados Unidos (2011) - Proposta de modelo de *e-learning* implementado sobre uma rede social (2015) - Análise de opiniões *online* sobre restaurantes (2014) - Avaliação de privacidade de redes sociais virtuais (2012)
"análise de sentimentos" AND "mídias sociais" OR "redes sociais"	2	- Análise de sentimentos no Twitter através de uma escala psicométrica (2015) - Estudo sobre construção da opinião a partir de um corpus textual extraído do YouTube (2017)

FONTE: O autor (2018).

Foi realizada também em 11 de outubro de 2018 uma pesquisa de teses e dissertações na Biblioteca Digital Brasileira de Teses e Dissertações (BDTD) envolvendo os termos “mineração de opinião”, “mineração de sentimentos”, “análise de opiniões” e “análise de sentimentos” aliados com os termos “mídias sociais”/”redes sociais”. Os resultados encontrados foram 21 (vinte e um) trabalhos relacionados, sendo que quatro deles não foram listados pois estavam repetidos na busca dos termos “mineração de opinião” e “análise de sentimentos”. Os resultados obtidos, bem como os temas abordados nos trabalhos encontrados podem ser visualizados na Tabela 2.

TABELA 2 – RESULTADOS DE TRABALHOS ENCONTRADOS NA BASE BDTD

Termos Pesquisados	Trabalhos Encontrados	Temas Abordados nos Trabalhos
"mineração de opinião" AND "mídias sociais" OR "redes sociais"	8	- Relevância da tradução de textos de português para inglês no processo de classificação binária de sentimentos em redes sociais (Dissertação de 2016) - Aplicação da mineração de opinião no planejamento turístico em opiniões extraidas do twitter e facebook (Dissertação de 2016) - Modelo social de relevância de opinião para usuários de redes sociais em domínios de jogos eletrónicos (Tese de 2014) - Experimentos comparativos combinando aprendizado de máquina e tradução automática para mineração de emoções em textos de diferentes idiomas (Dissertação de 2016) - métodos de análise de sentimentos com ferramentas de processamento de linguagem natural combinadas com algoritmos de aprendizagem de máquina (Dissertação de 2016) - Método para classificação de sentimentos em nível de característica baseado em pares (Dissertação de 2013) - Processo para classificação de sentimentos no twitter utilizando termos factuais e retweets em fonte de opiniões do debate político polarizado (Dissertação de 2014) - Processo de análise de sentimento em debates polarizados, com foco nos debates não ideológicos (Dissertação de 2014)
"mineração de sentimentos" AND "mídias sociais" OR "redes sociais"	0	-
"análise de opiniões" AND "mídias sociais" OR "redes sociais"	1	- normalização e classificação de polaridade de textos opinativos nas mídias sociais (Dissertação de 2015)
"análise de sentimentos" AND "mídias sociais" OR "redes sociais"	8	- Análise de sentimentos e afetividade aplicada em um sistema de recomendação de músicas (Tese de 2015) - Análise do uso de agregadores de classificadores para explorar a diversidade e a potencialidade de várias abordagens supervisionadas quando atuam em conjunto (Tese de 2015) - Exploração de informação relacional pela análise de opiniões para predição de preferência política de usuários do Twitter (Tese de 2016) - Análise de sentimento e desambiguação no contexto da tv social nas midias sociais (Dissertação 2012) - Mineração de opiniões aplicada a mídias sociais (dissertação de 2012) - Mineração de mídias sociais como ferramenta para a análise da tríade da persona virtual (Tese 2016) - Análise de sentimentos nas mídias sociais sobre o mercado cinematográfico americano (Dissertação de 2017) - Análise adaptatativa de fluxo de sentimentos nas mídias sociais (Dissertação de 2012)

FONTE: O autor (2018).

Pode-se verificar que dos 24 trabalhos encontrados, 18 foram publicados nos últimos 5 anos e destes 13 são dissertações ou teses. De todos os trabalhos, aquele que apresentou alguma relação com esta proposta devido ao estudo da análise de opiniões em mídias sociais foi o trabalho “Análise de opiniões expressas nas redes sociais” (TEIXEIRA; AZEVEDO, 2011).

1.3.2 Social

A finalidade social desta pesquisa é ser útil para empresas e profissionais que tenham interesse na exploração de informações textuais registrados em comentários

e opiniões dos consumidores nas mídias sociais, por meio da aplicação de técnicas de mineração de opinião, e poder aproveitá-las para gerar resultados de avaliação e sentimentos dos consumidores quanto a sua (in)satisfação dos produtos e/ou serviços oferecidos e com isso auxiliar nas estratégias e tomadas de decisão, na implementação de programas de melhorias, e na gestão de políticas de atendimento ao consumidor.

1.3.3 Econômica

Como agente contribuinte na preparação do custo da não-qualidade com as potenciais perdas de consumidores, dependendo do grau de insatisfação que eles demonstram nas reclamações registradas nas mídias sociais. Isto representa dados que se utilizam para custear os negócios que podem deixar se ser efetuados, em aquisição de produtos ou serviços ao terem suas compras interrompidas por consumidores insatisfeitos que não pretendem mais ser consumidores da organização.

1.3.4 Curso de Gestão da Informação (GI)

A contribuição desta pesquisa para o curso de GI está na abrangência da sua interdisciplinaridade, peculiar dos três pilares do curso, sendo: na área de Tecnologia da Informação, compreendendo a utilização de algoritmos de mineração de dados, na área de Ciência da Informação com a aplicação de tratamento das linguagens naturais para estabelecer padrões em formatos adequados à mineração de texto, e na área de Administração de Empresas com a utilização do conhecimento extraído nas bases textuais para poder auxiliar na aplicação de melhores tomadas de decisão.

Em pesquisa realizada em 03 de maio de 2018 no Repositório Digital Institucional da UFPR sobre trabalhos de monografia elaborados no curso de Gestão da Informação envolvendo os termos “mineração de opinião”, “mineração de sentimentos”, “análise de opiniões” e “análise de sentimentos”, foram encontrados 5 (cinco) trabalhos. Acrescentando-se um filtro na pesquisa com o termo “redes sociais”, encontra-se um único trabalho e com um filtro na pesquisa com o termo “Facebook”, também um único trabalho. Utilizando-se os termos “satisfação do cliente” e “satisfação do consumidor” encontram-se 3 (três) trabalhos. Por conseguinte, não

foram encontrados trabalhos de monografia no curso de graduação de Gestão de Informação da UFPR com o tema de mineração de opinião em mídias sociais como ferramenta para medir (in)satisfação do consumidor.

1.3.5 Pessoal

A finalidade da pesquisa do ponto de vista pessoal está em poder ampliar o conhecimento adquirido pelo pesquisador durante a sua vida profissional, quando trabalhava na área de atendimento ao consumidor, aplicando outras técnicas para aquisição de informações e assim efetuava a avaliação de satisfação dos consumidores. As técnicas que o pesquisador pretende utilizar na pesquisa vão ao encontro de uma proposta mais ágil e automatizada de medição da satisfação do consumidor.

1.4 DELIMITAÇÃO DA PESQUISA

A presente pesquisa pretende abranger as fases de extração de texto em uma base construída de SAC 2.0 na rede social Facebook, juntamente com a criação e validação de padrões para estarem alinhados ao processo de mineração, aplicando algoritmos/técnicas de mineração e com as informações obtidas poder apresentar os resultados. A pesquisa não intenciona abranger situações no que tange à gestão dos dados para práticas de programas de melhoria ou boas práticas da qualidade.

1.5 ESTRUTURA DO DOCUMENTO

Na sua estrutura, o documento está dividido em cinco partes, a primeira é a introdução, onde se apresenta o problema, o objetivo geral e os específicos, a justificativa da pesquisa e a delimitação do escopo da mesma.

A segunda seção se refere à revisão de literatura, relativo à abordagem da pesquisa quanto à satisfação do consumidor, mídias sociais, técnicas de processamento em linguagem natural, mineração de opinião e indicadores de satisfação.

A seção três trata dos encaminhamentos metodológicos que foram utilizados na extração das bases de dados, no pré-processamento e na mineração de opinião para levar a cabo os objetivos estipulados na pesquisa.

A quarta seção abrange os resultados finais, a apresentação dos resultados obtidos e as análises realizadas.

Na quinta seção serão apresentadas as considerações finais, as contribuições do estudo realizado e a possível continuidade que pode ser dada ao trabalho em estudos futuros.

2 REVISÃO DE LITERATURA

O referencial teórico a seguir visa a busca dos conceitos requeridos para poder dar fundamentação à pesquisa, e, por conseguinte dar sustento e direção aos objetivos do estudo proposto, a fundamentação teórica foi baseada nos pontos chave da pesquisa que são: a (in)satisfação do consumidor e o tratamento da linguagem natural para o formato adequado à mineração.

2.1 SATISFAÇÃO DO CONSUMIDOR

A satisfação do cliente é primordial para as organizações, independentemente do ramo de atuação, por conseguinte, para conseguirem que seus produtos e serviços sejam consumidos, é preciso primeiramente alcançar a satisfação de seus clientes. Os hábitos dos consumidores sofrem mudanças frequentes e com elas a forma com que eles pensam também muda, com isso, os hábitos de compra estão em constante modificação. O consumidor da atualidade, têm por diversas vezes dificuldades na hora de escolher por um produto ou serviço, principalmente devido à quantidade de opções ofertadas e à diversidade de fornecedores disponíveis.

Segundo Kotler (2012), A satisfação tanto pode consistir em um sentimento de prazer como de desapontamento, este sentimento resulta da comparação entre o desempenho de um produto e as expectativas existentes no consumidor. Quando o desempenho do produto vai além das expectativas o consumidor ficará altamente satisfeito. A fidelidade dos consumidores para com uma marca fica diretamente ligada às percepções a respeito de um produto de uma marca à qual eles constroem sentimentos favoráveis.

Para Cobra (2009), não é porque um cliente está satisfeito com um produto ou serviço que ele se tornará leal com a empresa, pois um cliente nunca está totalmente satisfeito. Conseguir alcançar a satisfação dos consumidores, é uma tarefa árdua e implica em ter plena consciência do que ele deseja ou espera. Para isso, é preciso estar constantemente medindo sua satisfação e buscar a melhoria contínua do desempenho e qualidade dos produtos ou serviços. A busca da satisfação total, tem como resultado cativar e reter o cliente buscando a sua lealdade.

Kotler (2012), afirma que quando o desempenho do produto vai além das expectativas, o consumidor ficará altamente satisfeito. A fidelidade dos consumidores

para com uma marca fica diretamente ligada às percepções a respeito de um produto de uma marca à qual eles constroem sentimentos favoráveis. Embora a empresa centrada no consumidor busque criar um alto nível de satisfação, essa não pode ser a principal meta, pois se aumentar a satisfação do cliente unicamente melhorando seus serviços ou reduzindo seus preços, o resultado é ficar com lucros menores. Uma forma da empresa satisfazer seus consumidores sem precisar abrir mão de sua lucratividade, é por exemplo melhorando seus processos ou investindo em pesquisa e desenvolvimento.

Para Gerson (2001), o atendimento a clientes se paga de várias formas, a principal é poder mantê-lo perenemente. Algumas organizações entendem o custo de conquistar um cliente, porém não tem ideia do quanto custa perdê-lo. Conquistar um novo cliente é cinco a seis vezes mais custoso do que negociar com um cliente antigo. Quando os clientes reclamam é porque estão se sentindo lesados de alguma forma, se forem tratados e atendidos de forma adequada e seus problemas forem resolvidos, 50 a 74 % desses clientes voltarão a negociar, caso contrário comentarão com 20 pessoas sua insatisfação. Quanto à satisfação e retenção de clientes, Gerson (2001, p. 85) afirma que:

> O melhor meio de satisfazer e manter seus clientes é conhecer o máximo possível sobre os mesmos. Você deveria saber o que eles gostam ou não, seus históricos de compras, suas necessidades e desejos e tudo o mais que possa ajudá-lo a mostrar-se mais atraente para eles. Sua meta é sempre manter sua lealdade e retê-los como clientes. (GERSON, 2001, p. 85).

Segundo Kotler (2012), Frederick Herzberg desenvolveu uma teoria que envolve dois fatores: os satisfatores, que são os fatores que causam satisfação e os insatisfatores, fatores que causam insatisfação. A falta de insatisfatores não é o suficiente para que haja motivação para uma compra, precisam haver fatores de satisfação também. A teoria implica em que devem em primeiro lugar ser evitados os insatisfatores, que mesmo não sendo elementos que vendem um produto, podem impedir que seja vendido. Uma segunda implicação é que devem-se identificar os principais satisfatores que motivam a compra e fornecê-los.

2.1.1 SAC 2.0

Com o surgimento da Internet e das mídias sociais, as empresas ante a necessidade de atender melhor o consumidor e como uma opção alternativa ao SAC tradicional por meio de telefone, se propuseram a procurar novas formas de poder dar suporte ao cliente, ouvir seus elogios, dúvidas, comentários, sugestões e reclamações. Com isso, as organizações se deram conta que as mídias sociais poderiam ser uma boa oportunidade para o atendimento aos consumidores. Foi onde surgiu uma inovação denominada SAC 2.0, que nada mais é que um SAC voltado às mídias sociais, onde grande parte de consumidores as utiliza. Teixeira (2012, p. 11) destaca que:

> De acordo com o IBOPE, no primeiro trimestre de 2012 o número de usuários de internet no Brasil com mais de 16 anos era de 82,4 milhões de pessoas, ou seja, cerca de 43% da população brasileira tem acesso à internet, isso fez com que os clientes buscassem nessa nova ferramenta uma maneira alternativa de comunicação com as empresas. (TEIXEIRA, 2012, p. 11).

Segundo Gonsalves e First (2013), o SAC 2.0 constitui o serviço de atendimento ao consumidor nas mídias sociais, principalmente no Twitter e Facebook. Pelo fato da internet estar presente no dia a dia das pessoas, o SAC 2.0 bem sendo mais buscado pelos consumidores pois o ambiente proporciona rapidez e agilidade. O grande diferencial além da velocidade é que diferente do SAC tradicional que é de mão única, o SAC 2.0 é um canal de duas vias, onde existe uma interação entre os usuários e as organizações. Gonsalves e First (2013, p.49) concluem que:

> Ao procurar um SAC 2.0, as pessoas buscam eficiência por parte da organização, ou seja, pretendem que a empresa responda adequadamente e atenda às suas necessidades. Caso contrário, podem reclamar da empresa nas próprias redes sociais – o que pode ser altamente prejudicial à sua imagem – ou procurar uma empresa concorrente. Em ambos os casos, a empresa em questão se prejudica. Saber como se posicionar nas redes sociais, portanto, é fundamental para que o relacionamento entre a empresa e seus públicos se mantenha satisfatório. O SAC 2.0 é um canal informal, porém não menos importante, de modo que não deve ser deixado de lado no planejamento estratégico da organização. Afinal, ele pode auxiliar na tomada de decisão, na redução de crises e na manutenção da confiança do consumidor, assim como as outras ações de relações públicas. (GONSALVES e FIRST, 2013, p. 49).

2.2 INDICADORES DE SATISFAÇÃO

Saber o quanto os clientes estão satisfeitos são medidas de grande utilidade para as organizações, elas auxiliam a descobrir o grau em que um produto ou serviço está cumprindo com as expectativas do cliente. Através do uso dos indicadores de satisfação, é possível determinar quais são os pontos favoráveis e onde precisam haver melhorias.

Para Slack e Lewis (2002), existem basicamente cinco objetivos de desempenho que tem por finalidade atender as exigências dos clientes e tem significado para qualquer tipo de operação, obedecendo prioridades diferentes, dependendo de cada situação:

- qualidade: está relacionado com as especificações do produto ou serviço e estar em conformidade com as expectativas do consumidor;
- rapidez de resposta: refere-se ao tempo entre a solicitação e o recebimento do produto ou serviço pelo cliente;
- confiabilidade: tem relação com o cliente receber o produto ou serviço no momento necessário ou quando prometido;
- flexibilidade: refere-se à capacidade de alterar de alguma maneira o que, como e quando uma operação pode fazer algo;
- custo: é o objetivo mais importante para as organizações que competem diretamente com preço. Quanto menor o custo de produção menor pode ser o preço aos consumidores.

Segundo Hill, Self e Roche (2002), a lealdade do cliente vai se conseguindo com o tempo e a satisfação do cliente é baseada em atender ou superar as exigências de clientes. Para isso ser alcançado, é necessário que a empresa tenha foco no cliente, de forma a fazer o melhor para ele. Um programa de medição de satisfação do cliente, irá fornecer as informações necessárias para adquirir-se os efeitos financeiros de se ter a fidelidade e a satisfação do cliente. Os autores complementam que a medição de satisfação do cliente, permite identificar com maior precisão os requisitos dos clientes; entender como eles percebem a organização e se o desempenho atende suas demandas; identificar as prioridades para as melhorias; saber as áreas em que as melhorias de desempenho produzirão melhor ganho em

índice de satisfação do cliente; identificar onde a organização está falhando no entendimento de atender as necessidades do cliente; definir metas para melhorias de serviço; monitoramento do progresso em relação ao índice de satisfação do cliente e aumentar os lucros por meio de uma maior fidelização e retenção dos clientes.

2.3 MÍDIAS SOCIAIS

As mídias sociais são uma nova forma de comunicação por meio de estruturas que permitem que pessoas e empresas possam interagir e trocar informações entre si de forma rápida e em tempo real, seja por algum interesse mútuo ou até mesmo por querer demonstrar sua opinião.

Telles (2010), afirma que diversas pessoas se confundem com os termos “mídias sociais” e “redes sociais” e por várias vezes os usam de forma confusa. O autor afirma que os termos não têm o mesmo significado, já que redes sociais são uma categoria de mídias sociais.

Segundo a Post Digital (2018), “pode-se dizer que uma rede social é focada na criação ou manutenção de relacionamentos entre as pessoas, enquanto uma mídia social é mais focada no compartilhamento do conteúdo”.

Para Kaplan e Haenlein (2010), as mídias sociais são um grupo de aplicações baseadas na Internet que foram concebidas sobre as fundações ideológicas e técnicas da Web 2.0, e possibilitam geração e troca de conteúdo criado pelo usuário.

O fator dificultador de se interpretar dados obtidos nas mídias sociais através de mineração de opinião é a linguagem natural utilizada nesses meios, pois não é padronizada como os textos registrados nos sistemas SAC. Com isso, torna-se mais complexo o pré-processamento, principalmente quando se trata de filtrar dados relevantes e não relevantes, e mesmo que o nível de análise de opiniões buscado seja positivo, negativo e neutro, há necessidade de criar-se agrupamentos devido à grande quantidade de opiniões registrada nas mídias sociais e ao fato de existir um grau de subjetividade intenso, o que dificulta ainda mais a interpretação pela máquina. Segundo Santos (2014), pelo fato de alguns textos não serem estruturados e não terem uma formalidade composta por sujeito e predicado é possível se deparar com dificuldades, já que grande parte dos textos extraídos da internet apresentam erros gramaticais, duplo sentido, abreviações, sarcasmos, ironias e gírias.

Com as mídias sociais, pode-se chegar rapidamente a diversas informações, sem mesmo ter-se a necessidade de usar motores de busca. A comunicação das mídias sociais flui em todos os sentidos, já que quando alguém publica uma opinião ou faz alguma pergunta, outros podem responder e dar suas opiniões. Além do uso na comunicação e no entretenimento, as mídias sociais apresentam vantagens que as organizações aproveitam para poder mostrar sua imagem, seus negócios e seus produtos de forma econômica e com grande alcance e difusão ao público. As organizações também podem usar as mídias sociais para captar as percepções de (in)satisfação que os consumidores têm a respeito das marcas, dos seus produtos e serviços. Da mesma forma os consumidores podem utilizar as mídias sociais para expressar suas opiniões a respeito das empresas, marcas e produtos. De acordo com Johnson (2014), o mundo dos consumidores mudou e está mudando a cada dia. O consumismo passivo ficou para trás e os consumidores querem ter a sua voz e expressar sua opinião a respeito dos produtos e serviços. Este fato levou as empresas a estar em um clima competitivo, na busca ativa de formas de se alavancarem por meio da inovação.

Segundo Franco (2018), as mídias sociais Youtube e Facebook estão entre as primeiras colocações no ranking dos sites mais usados no Brasil, conforme a empresa Alexa, afiliada à Amazon.

2.3.1 Facebook

Para Johnson (2014), o Facebook é uma espécie de serviço de rede social que foi lançado em 2004. Para que possa ser utilizado, faz-se necessário que o usuário se registre e crie seus perfis. Uma vez criados os perfis, os usuários podem fazer troca de mensagens, publicar seu *status*, publicar fotos, e conversar com outros usuários via *chat*, entre outras funcionalidades. Com o passar do tempo a rede social que era simples, converteu-se em uma rede mais profissional com as possibilidades de uso que apareceram para as organizações.

Segundo Oliver e Fernandez (2012), o Facebook é a rede social que tem maior impacto internacional. Desde o seu lançamento, a ferramenta de comunicação social criada por Mark Zuckerberg na Universidade de Harvard, oferece serviços baseados no conceito da “amizade”, onde os usuários se comunicam entre os já existentes na sua conta, adicionar novas amizades no seu círculo social e virtual e podem inclusive

procurar pessoas com as que já perderam contato há muito tempo. Para tal finalidade o servidor do Facebook proporciona ferramentas de busca e sugestões de pessoas conhecidas. Estes precisam estar registrados na rede e aceitar o convite enviado para que sejam adicionados à conta de quem convidou.

Oliver e Fernandez (2012), relatam que o Facebook proporciona a possibilidade de utilizar “grupos e páginas”, que permitem com que pessoas com interesses comuns possam estar participando. Há também organizações públicas e privadas que criam páginas e grupos com a finalidade de divulgar conteúdos científicos e eventos de interesse da comunidade acadêmica.

2.3.2 Youtube

O Youtube é considerado uma mídia social onde os usuários podem postar, ver e compartilhar vídeos. Segundo Johnson (2014), o Youtube é um serviço web que foi fundado por empregados da PayPal, foi lançado em 2005 e desde esse então a popularidade tem crescido de forma constante e chegou a ser reconhecido como o site web com maior tráfego de internet. Em 2006, foi comprado pelo Google por $ 1.600 milhões de dólares e a partir disso os aportes financeiros advindos da publicidade crescem “exponencialmente”, bem como os usuários.

Para Oliver e Fernandez (2012), O Youtube habilita o acesso aos usuários para poderem visualizar o conteúdo que outros usuários publicaram sem a necessidade de estarem registrados. O *website* do Youtube permite hospedar vídeos e compartilhá-los de forma simples. É um recurso cada vez mais usado por escolas e universidades, também em reportagens, entrevistas, videoaulas e outros.

Segundo Johnson (2014), O Youtube proporciona uma plataforma de marketing rentável para as organizações e é um canal barato e efetivo de publicidade, onde as empresas de várias indústrias o estão utilizando como forma de construir relações com os seus clientes, aproveitar a lucratividade e como estratégia de marketing indireto. Além disso o Youtube pode ser utilizado para informação, educação e entretenimento.

2.3.3 Twitter

O Twitter é uma rede social baseada em tecnologia *microblogging* que permite aos usuários enviar e ler textos denominados *tweets* com um comprimento máximo de 140 caracteres.

Segundo Johnson (2014), O Twitter foi criado por Jack Dorsey em 2006 e tem sido denominado de “SMS da Internet”. O Twitter proporciona uma mistura de formas de comunicação de textos, fotos, músicas e vídeos e envolve as experiências do dia a dia de usuários. Através de “*hashtags*”, que são palavras chave utilizadas para identificar conteúdos através das mídias sociais, os usuários podem localizar mensagens sobre assuntos específicos, publicar e ver atualizações, bem como, seguir outros usuários e enviar respostas públicas ou privadas. Com o tempo o Twitter cresceu permitindo aos usuários poder procurar pessoas, notícias e outros temas.

Os usos mais frequentes do Twitter são o seguimento de eventos ao vivo, troca de opiniões entre usuários, comentários e debates sobre diversos assuntos, entre outros. Conforme Oliva e Fernandez (2012), é frequente participantes de eventos científicos utilizarem o Twitter para fazerem a divulgação em tempo real de informação relevante e resumida para seguidores interessados.

2.4 DESCOBERTA DE CONHECIMENTO EM TEXTO (DCT)

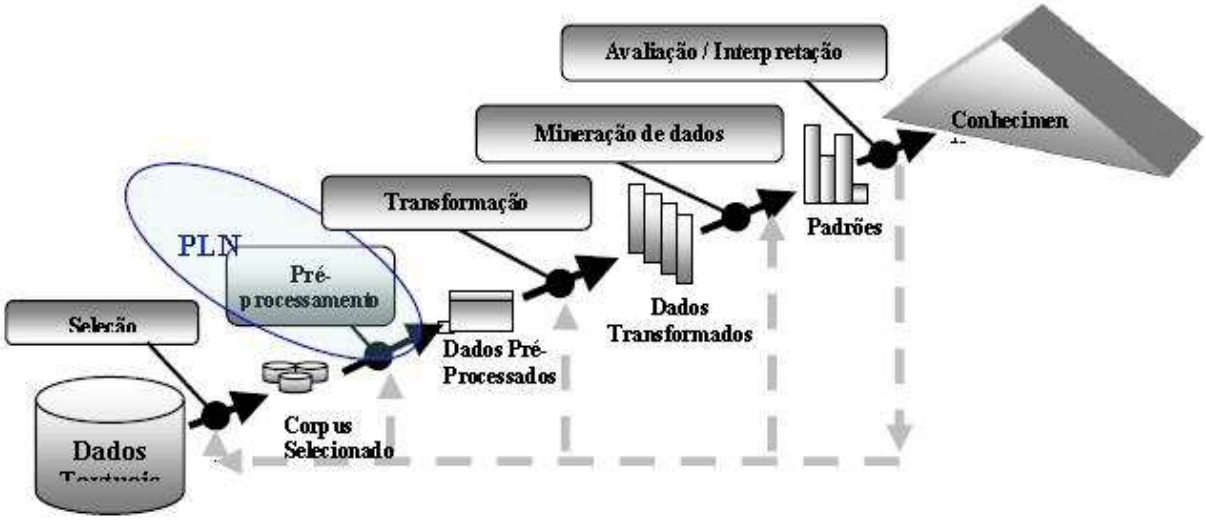
As mídias sociais estão cheias de informações, dados, notícias, imagens, vídeos e em alguns casos mais específicos como o Facebook, há um conteúdo bem significativo em texto, tanto de pessoas como de organizações que escrevem mensagens contendo opiniões, comentários e reclamações dos mais variados assuntos. O grande dificultador é que estes textos se encontram em linguagem natural, isto é, na linguagem razoavelmente inteligível para o ser humano, mas não para as máquinas. Para que possa haver uma interpretação desses textos pelas máquinas, faz-se necessário que seja realizado uma etapa de pré-processamento prévia à mineração denominada Descoberta de Conhecimento em Texto (DCT).

Para Schiessl e Bräscher (2011), devido à complexidade da linguagem natural para a interpretação direta das máquinas, é necessário fazer uma extração de conhecimento das bases textuais e criar agrupamentos e modelo de classificação automatizada para que possa ser interpretado por computadores.

Segundo Schiessl e Bräscher (2011), a Descoberta de Conhecimento em Texto (DCT), é constituída pela conexão das técnicas de Processamento de Linguagem Natural (PLN) e a Descoberta de Conhecimento em Dados (DCD). O DCT tem o objetivo de automatizar o processo de transformação de dados textuais em informações que permitem adquirir conhecimento.

Schiessl e Bräscher (2011), demonstram na Figura 1 o ciclo do processo de descoberta de conhecimento de texto por meio de uma consolidação de conceitos de autores, através da adaptação ao modelo que foi proposto por Fayyad, Piatetsky-Shapiro e Smyth (1996).

FIGURA 1 - PROCESSO DE DESCOBERTA DE CONHECIMENTO EM TEXTO



FONTE: ADAPTADO DE FAYYAD, PIATETSKY-SHAPIRO E SMYTH (1996) POR SCHIESSL E BRÄSCHER (2011).

2.5 MINERAÇÃO DE OPINIÃO

Segundo Liu (2015), A mineração de opinião, é o estudo computacional das opiniões, sentimentos, atitudes e emoções das pessoas. A mineração de opinião é dirigida principalmente a opiniões que exprimem sentimentos positivos ou negativos. Deve-se considerar também expressões que não denotam nenhum sentimento, as denominadas expressões neutras. Além da opinião e do sentimento, existem os conceitos de afeto, emoção e humor, que vem a ser os estados psicológicos mentais.

Para Chen e Zimbra (2010), a mineração de opinião é uma subdivisão da mineração de dados (*data mining*) e consiste em métodos que classificam,

processam, extraem e analisam as diversas opiniões que podem ser encontradas em diferentes locais da internet e mídias sociais.

Segundo Castro e Ferrari (2016), classificar um objeto é outorgar rótulos que se denominam classes, conforme a categoria a que os mesmos pertencem. Desta forma um algoritmo de classificação se usa na construção de modelos chamados de classificadores, que se constroem com base em um conjunto de treinamento de dados rotulados.

Os autores, afirmam que existem diversos algoritmos de classificação e podem ser categorizados conforme a sua estrutura em:

- baseados em conhecimento: modelo de algoritmo classificador que se baseia em um conjunto de regras que se utilizam para outorgar certas classes a um objeto se o mesmo satisfazer condições predefinidas. A estrutura deste modelo pode ser visualizada na Figura 2.

FIGURA 2 – ESTRUTURA DE ALGORITMOS BASEADOS EM CONHECIMENTO

FONTE: Castro e Ferrari (2016, p.165).

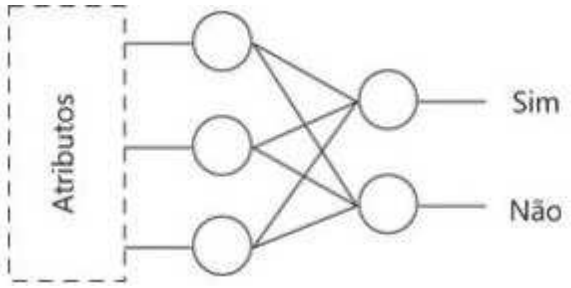
- baseados em árvores: comumente utilizados em mineração. Nos processos de classificação. A classificação ocorre de maneira que o nó raiz e os nós intermediários das árvores apresentam os testes de um atributo, os galhos da árvore são os resultados e as folhas são os rótulos de classe. A estrutura do modelo baseado em árvores pode ser vista na Figura 3.

FIGURA 3 – ESTRUTURA DE ALGORITMOS BASEADOS EM ÁRVORE

FONTE: Castro e Ferrari (2016, p.166).

– conexionistas: modelos baseados em nós interconectados. Estes sistemas são um tipo de grafo, os modelos conexionistas mais comuns são as redes neurais artificiais, muito embora existam outros modelos conexionistas diferentes. A estrutura do modelo conexionista pode ser visualizada na Figura 4.

FIGURA 4 – ESTRUTURA DE ALGORITMOS CONEXIONISTAS



FONTE: Castro e Ferrari (2016, p.166).

– baseados em distância: estes modelos são classificados através do cálculo da distância entre o objeto ao qual se deseja conhecer a classe e um ou mais objetos rotulados. Assim, a classe do objeto que ainda não é conhecido passa a ser a mesma classe dos que estão próximos a ele. A estrutura do modelo baseado em distância pode ser vista na Figura 5.

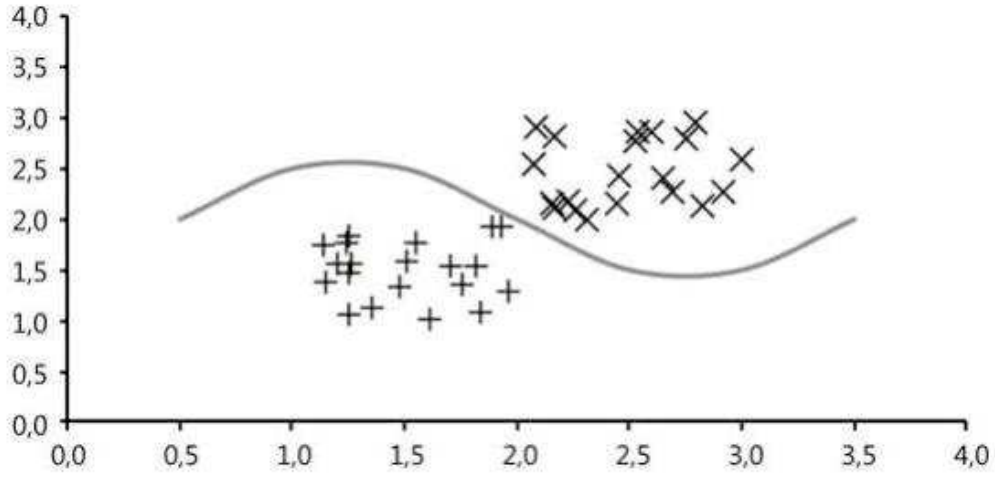
FIGURA 5 - ESTRUTURA DE ALGORITMOS BASEADOS EM DISTÂNCIA

FONTE: Castro e Ferrari (2016, p.166).

– baseados em função: modelos paramétricos baseados em funções definidas onde os seus parâmetros são adequados no processo de treinamento. Logo após esse processo, um novo objeto cuja classe é desconhecida se apresenta à função e por sua vez o valor desta é calculado e representa de alguma forma

a classe desse objeto. A estrutura do modelo baseado em distância pode ser visualizada na Figura 6.

FIGURA 6 - ESTRUTURA DE ALGORITMOS BASEADOS EM FUNÇÃO



FONTE: Castro e Ferrari (2016, p.167).

– probabilísticos: estes modelos comportam atribuir a probabilidade de um objeto poder pertencer a uma ou mais classes. A estrutura deste modelo pode ser vista na Figura 7.

FIGURA 7 – ESTRUTURA DE ALGORITMOS PROBABILÍSTICOS

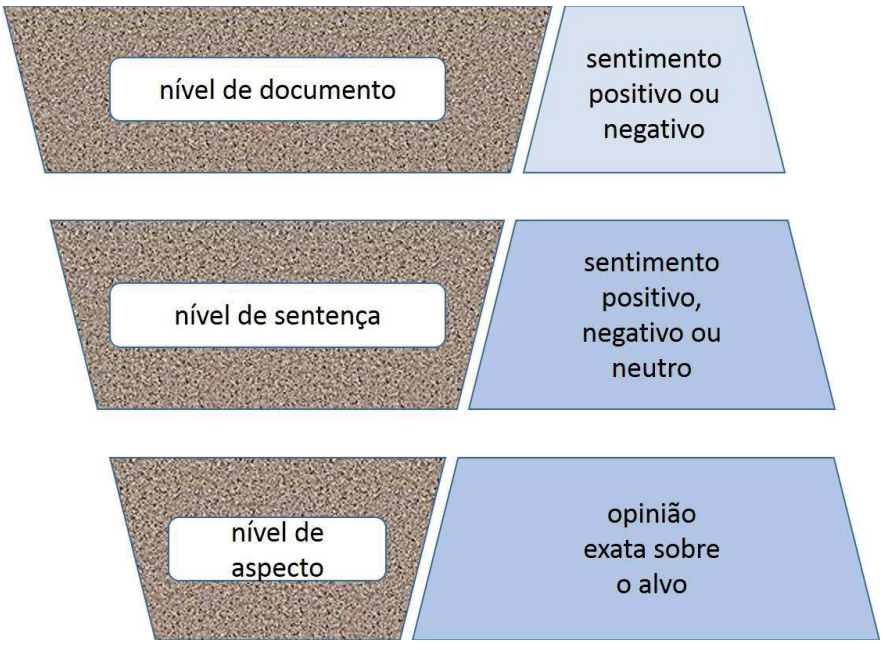
FONTE: Castro e Ferrari (2016, p.167).

Nas mídias sociais são encontradas diversas opiniões sobre produtos, contextos geopolíticos e sociais, permitindo fazer extrações de sentimentos e a possibilidade de classificá-los. A análise do sentimento se usa com frequência na mineração de opinião para poder identificar sentimentos, afetos, subjetividade e demais estados emocionais no texto online. Uma dificuldade encontrada na mineração

de opinião é a complexidade de sentimentos expressos por um grupo grande de participantes, pois com a diversidade de resultados impede de que haja um consenso entre as opiniões.

Conforme Liu (2015), a análise de sentimentos se conduz em três níveis. O primeiro nível denominado nível de documento (*document level*), onde se classifica primeiramente todo o documento para saber se está expressando um sentimento positivo ou negativo. O segundo nível é denominado nível de sentença (*sentence level*), onde se analisa se cada frase expressa uma opinião positiva, negativa ou neutra. Normalmente a opinião neutra quer dizer sem opinião. Para este segundo nível de análise se classifica a subjetividade, que faz distinção entre frases que expressam sentenças objetivas de frases que expressam visões e opiniões subjetivas ou frases subjetivas. Ainda Liu (2015), afirma que a subjetividade não é o mesmo que um sentimento ou uma opinião, pois muitas sentenças objetivas podem implicar sentimentos ou opiniões. O terceiro nível denomina-se nível de aspecto (*aspect level*), onde diferentemente dos níveis um e dois, onde em nenhum caso as análises denotam se as pessoas gostam precisamente ou não, no nível três a análise consegue diferenciar. O nível de aspecto, que era anteriormente chamado de nível de recurso, examina diretamente a opinião e seu alvo, como pode ser visto na Figura 8.

FIGURA 8 – NÍVEIS DE ANÁLISE DE SENTIMENTOS



FONTE: Adaptado de Liu (2015) pelo autor (2018).

Existem ainda segundo Liu (2015), além dos três níveis de análise de sentimentos, dois tipos diferentes de opiniões, as opiniões regulares e as opiniões comparativas. As opiniões regulares, são as mais comuns e expressam um sentimento sobre uma entidade em particular ou de um certo aspecto da entidade em si. As opiniões comparativas comparam várias entidades baseadas em seus aspectos de preferências.

Após a discussão dos principais termos relacionados, a próxima seção apresenta os encaminhamentos metodológicos que foram adotados no desenvolvimento deste trabalho.

3 ENCAMINHAMENTOS METODOLÓGICOS

Neste capítulo estão descritos os encaminhamentos metodológicos a serem realizados para levar a cabo a pesquisa. Tais procedimentos levam em consideração o objetivo geral e os objetivos específicos que foram definidos neste trabalho.

3.1 CARACTERIZAÇÃO DA PESQUISA

Quanto ao nível desta pesquisa, pode ser classificada como **exploratória**, já que visa ampliar o estudo para futuros trabalhos que tenham objetivos semelhantes e poder proporcionar possíveis perspectivas sobre oportunidades ou hipóteses sobre problemas afins. Para Gil (2002), a pesquisa exploratória tem como finalidade uma maior aproximação com o problema, de maneira a torná-lo mais claro ou até mesmo criar-se novas hipóteses. O planejamento da pesquisa exploratória segundo o autor, é bastante flexível por permitir considerar vários aspectos relacionados com o tema estudado. Várias vezes, pesquisas exploratórias podem constituir a primeira parte de uma pesquisa mais detalhada (GIL, 2009).

Quanto à natureza da pesquisa, é classificada como **quantitativa** por abranger uma forma estruturada de coletar e analisar dados quantitativos e possibilitar o estudo da relação entre as variáveis quantificadas, permitindo uma melhor interpretação dos resultados.

Os métodos indicadores do meio técnico a serem utilizados nesta pesquisa são: o método **observacional** e **experimental**, por permitirem se complementar. Segundo Gil (2009), o método observacional outorga um maior grau de precisão no campo das ciências sociais e permite observar o que acontece ou aconteceu. O método experimental permite ao pesquisador analisar os resultados de uma variável no objeto estudado e verificar o que vem a seguir.

A técnica a ser utilizada na pesquisa para a obtenção de dados é a de **observação simples**, onde, segundo Gil (2009, p. 101) é “aquela em que o pesquisador, permanecendo alheio à comunidade, grupo ou situação que pretende estudar, observa de maneira espontânea os fatos que aí ocorrem”. Para Lakatos e Marconi (1992), as técnicas que correspondem à parte prática da coleta de dados podem ser classificadas em: documentação indireta, que compreende a pesquisa documental e bibliográfica e a documentação direta, que por sua vez pode ser dividida

em intensiva e extensiva. Dentro da documentação direta intensiva, há a técnica de observação, onde Lakatos e Marconi (1992, p.107) definem que:

> [...] utiliza os sentidos na obtenção de determinados aspectos da realidade. Não consiste apenas em ver e ouvir, mas também em examinar fatos ou fenômenos que se deseja estudar. Pode ser: Sistemática, Assistemática; Participante, Não participante; Individual, em Equipe; na Vida Real, em Laboratório (LAKATOS e MARCONI, 1992, p.107).

Quanto ao envolvimento do pesquisador nesta pesquisa, foi seguido o **modelo clássico**, que, segundo Gil (2009) ocorre quando a possibilidade de distorção do estudo pelo pesquisador é mínima, em outras palavras, as opiniões do pesquisador tem pouca chance de interferirem na pesquisa.

Quanto à finalidade da pesquisa, pode ser apontada como de **ordem prática**. Segundo Gil (2002), existem diversos motivos que podem levar à execução de uma pesquisa. Podem ser classificadas em dois grupos: de ordem intelectual e de ordem prática. A causa de ordem intelectual é originada para se satisfazer o desejo de conhecer. O motivo de ordem prática se relaciona com o desejo de conhecer, mas com a intenção de realizar algo mais eficiente ou efetivo.

A Figura 9 apresenta o esquema resumido da caracterização da pesquisa.

FIGURA 9 - ESQUEMA DA CARACTERIZAÇÃO DA PESQUISA

FONTE: O autor (2018).

De acordo com Gil (2009), as pesquisas diferem entre si devido a cada uma delas ter procedimentos e objetivos diferentes. Por isso é impossível serem definidas de forma esquematizada as etapas de uma pesquisa. Para esta pesquisa, as principais etapas previstas são:

a) formulação do problema;
b) determinação de objetivos geral e específicos;
c) justificativa da pesquisa;
d) seleção de base de dados a pesquisar;
e) pré-processamento dos dados;
f) mineração dos dados;
g) análise e interpretação dos resultados;
h) considerações parciais e finais.

3.2 MATERIAIS E MÉTODOS

Seguindo a classificação de Liu (2015), o nível de análise de sentimento que foi adotado neste trabalho é o segundo nível denominado nível de sentença, que visa a análise das frases de consumidores com opiniões de subjetividade positiva, negativa e neutra. Com este nível de análise, intencionasse poder classificar as opiniões dos consumidores sobre produtos e serviços, considerando a satisfação em opiniões positivas e a insatisfação em opiniões negativas. Para as opiniões em que os consumidores não expressam claramente uma opinião, serão consideradas como opiniões neutras.

3.2.1 Base de dados

A base de dados utilizada para este trabalho foi retirada da página oficial da empresa Ford Brasil no Facebook, especificamente dos comentários da postagem de 30 de julho de 2018, sobre a nova linha Ford Ka 2018 que contava quando foi capturada a base com 8 mil visualizações, 64 mil curtidas, 1.169 compartilhamentos e 1.114 comentários. O motivo de ter sido selecionada a base de dados sobre a postagem mencionada da empresa de automóveis Ford, é por contemplar nas postagens opiniões positivas, negativas e neutras de forma equilibrada, relevante para que os objetivos deste estudo sejam atingidos, uma vez que visam mensurar tanto as

opiniões de satisfação como as de insatisfação. Outro motivo de ter sido selecionada uma empresa automobilística é por ela abranger não somente as opiniões do produto em si, mas também as de outros serviços e atendimentos agregados que estão envolvidos, como o de vendas, pós-vendas, revisões, garantia, entre outros.

Outras bases de dados retiradas das páginas de empresas no Facebook foram avaliadas tais como empresas de telefonia, televisão por assinatura, eletrônicos, eletrodomésticos, bancos, planos de saúde, e varejo eletrônico. Porém a que melhor atendeu aos critérios de seleção desta pesquisa por contemplar maior gama de serviços avaliados e homogeneidade quanto a sentimentos de satisfação e insatisfação foi a da empresa Ford Brasil.

3.2.1.1 Facebook

A extração da base de dados foi efetuada no Facebook, o fato de ter-se escolhido esta rede social para a pesquisa, deve-se a que a mesma consta em segundo lugar em mídias sociais no ranking de sites mais utilizados no Brasil, segundo a empresa Alexa (2018) que é afiliada a Amazon, empresa de comércio eletrônico.

Este ranking é obtido pela média de visitantes que diariamente acessam um certo site e o número de visualizações deste mesmo site no período do último mês. (ALEXA, 2018).

O Facebook segundo o ranking está atrás apenas do Youtube.com, que não foi utilizado na pesquisa por ser uma plataforma de compartilhamento de vídeos e por não estar no escopo desta pesquisa a análise de vídeos. Ainda, os comentários postados em relação aos vídeos apresentam, em sua grande maioria, menor quantidade quando comparados aos comentários no Facebook. Inclusive para o vídeo utilizado para fins desta pesquisa, o lançamento do Ford Ka 2018, apresenta um bloqueio de comentários no Youtube.com[1].

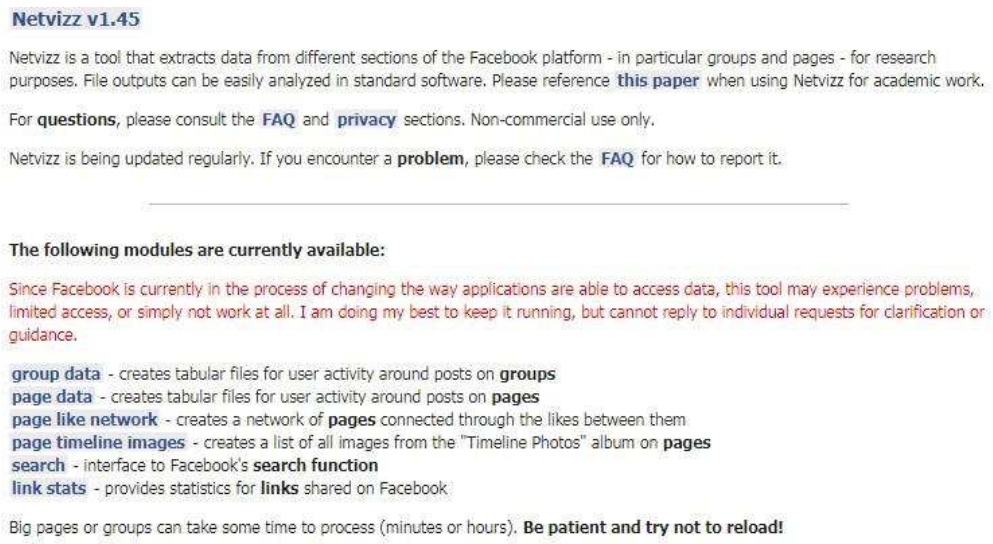
[1] Ford Ka Família. Disponível em: <https://www.youtube.com/watch?v=2g34tayfiTg>. Acesso em: 23 out. 2018.

3.2.1.2 NetVizz

Na coleta de dados foi utilizada uma aplicação denominada NetVizz que é uma ferramenta com várias funções, criada por Bernhard Rieder em 2009 e permite que seja executada dentro do próprio Facebook, extraindo dados de seções da plataforma gerando arquivos com extensão “tab” e possibilitando que sejam analisados para fins de pesquisa. Segundo Rieder (2013), a ferramenta inicialmente se desenvolveu com o intuito de estudar uma interface de programação de aplicações (API) para o Facebook como um novo objeto de mídia e para a avaliação de métodos nativos digitais. Depois, a aceitação do NetVizz foi tão positiva que acabou sendo utilizado como uma ferramenta para extração de dados do Facebook.

Para poder se utilizar a ferramenta, basta ter uma conta no Facebook pois a aplicação utiliza a plataforma como interface para permitir fazer a extração de dados. Nos últimos anos a aplicação está sendo restringida quanto à extração de dados. Em 2015 o Facebook, obedecendo a política de acesso à informação restringiu a extração de dados pessoais de usuários. Na tela inicial da aplicação, há um alerta do criador do NetVizz, informando que o Facebook está em processo de alterar a forma de como os aplicativos podem acessar os dados, e alertando que a ferramenta pode ter problemas, acesso limitado ou até não funcionar. A Figura 10 apresenta a tela inicial do NetVizz com as opções disponíveis e o alerta do criador da ferramenta quanto às limitações da ferramenta.

FIGURA 10 – TELA INICIAL DA APLICAÇÃO *NETVIZZ*

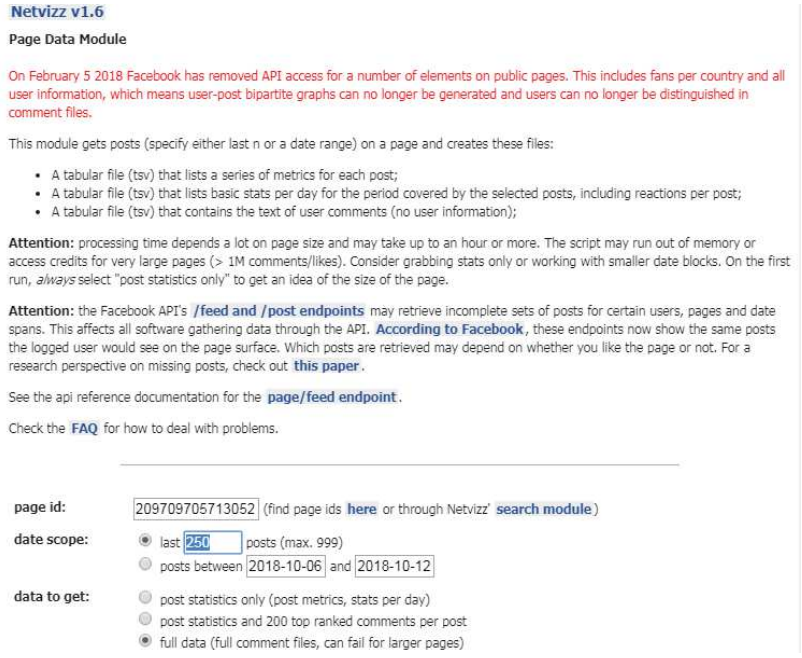


FONTE: NetVizz (2018).

Dentro das opções disponíveis na tela principal, existem as seguintes alternativas de extração de dados: o módulo *group data* que permite a extração de dados dentro dos grupos do Facebook, o módulo *page data* que corresponde às postagens que são efetuadas em páginas do Facebook. Há também o módulo *page timeline images* que permite extrair fotos da linha do tempo de páginas do Facebook e o módulo *link stats* que traz estatísticas dos *links* compartilhados no Facebook. Os módulos *page like network* e *search module* não estão mais em funcionamento.

O módulo dentro do NetVizz selecionado para extrair a base de dados para este estudo foi o módulo *page data*, pois é onde se encontram as postagens por página. Foi necessário primeiramente identificar qual é a página do Facebook da Ford Brasil. Uma vez acessado este módulo, foi preciso preencher os campos "*page id*" (código da página). No campo "*date scope*", deve-se optar pelo número das últimas postagens que se deseja extrair (sendo o limite máximo de 999) ou as postagens de um período correspondente entre duas datas. É preciso selecionar as opções de somente extrair a parte estatística das postagens, as estatísticas e as 200 postagens mais comentadas por postagem, ou ainda, há a opção de trazer todos os dados. É possível extrair as postagens por página ou por página e usuário. A tela do módulo *page data,* onde devem ser preenchidas as informações, pode ser observada na Figura 11 com as seleções que foram feitas para a extração dos dados.

FIGURA 11 – TELA INICIAL DO MÓDULO "*PAGE DATA"*



FONTE: NetVizz (2018).

Para poder-se adquirir o código da página e preencher com o mesmo no campo "*page id*" do módulo "*page data*", é necessário acessar o link http://lookup-id.com/# e copiar o endereço da página do Facebook que se deseja, pressionando em seguida "*lookup*". No caso o link buscado foi o www.facebook.com/FordBrasil/ conforme pode ser visto na Figura 12.

FIGURA 12 – TELA DE SOLICITAÇÃO DO CÓDIGO "*PAGE ID*"

FONTE: Facebook (2018).

Após o procedimento apresentado na Figura 12, o resultado do código "*page id*" foi gerado, conforme pode ser observado na Figura 13.

FIGURA 13 – TELA DE AQUISIÇÃO DO CÓDIGO "*PAGE ID*"

FONTE: Facebook (2018).

Em seguida, depois de se preencher todos os campos necessários na tela inicial do módulo *page data* e selecionar extrair as postagens por página ou por página e usuário, o aplicativo procede com a extração dos dados e geração do arquivo "*zip*" conforme pode ser observado na Figura 14.

FIGURA 14 – TELA DE EXTRAÇÃO DE DADOS NO *NETVIZZ*

FONTE: NetVizz (2018).

Em qualquer uma das opções que for selecionada no módulo *page data*, é feita a extração e gerado um arquivo "*zip"*, contendo 3 arquivos com extensão "*tab*", sendo um com os comentários das postagens, outro a estatística por dia e um terceiro arquivo contendo toda a estatística. Nos três arquivos gerados na extração, o arquivo *fullstats.tab* contém toda a estatística das postagens, comentários, compartilhamentos e reações dos usuários. As descrições dos metadados do arquivo *fullstats.tab* podem ser visualizadas na Tabela 3.

TABELA 3 - DESCRIÇÕES DOS METADADOS DO ARQUIVO *FULLSTATS.TAB*

(Continua)

Metadado	Descrição
type	Classificação da postagem
by	Código da postagem da página
post_id	Identificação da postagem
post_link	Link direto da postagem
post_message	Texto da Postagem
picture	Link da foto incluida na postagem
full_picture	Imagem da postagem tamanho grande
link	Link da postagem
link_domain	Link do domínio da postagem
post_published	Data de publicação da postagem
post_published_unix	Data de publicação da postagem em formato Unix

(Conclusão)

post_published_sql	Data de publicação da postagem em formato SQL
likes_count_fb	Número de curtidas da postagem
comments_count_fb	Número de comentários da postagem
reactions_count_fb	Número de reações e curtidas da postagem
shares_count_fb	Número de compartilhamentos da postagem
engagement_fb	Número de cometários, reações e de ações
comments_retrieved	Número de comentários recuperados
comments_base	Número de comentários no nível básico
comments_replies	Número de respostas nos comentários
comment_likes_count	Número de "curtidas" nos comentários da postagem
rea_LOVE	Reação de amor
rea_WOW	Reação de surpresa
rea_HAHA	Reação de alegria
rea_SAD	Reação de tristeza
rea_ANGRY	Reação de raiva
rea_THANKFUL	Reação de agradecimento

FONTE: NetVizz (2018).

O arquivo *statsperday.tab* contém toda a estatística por dia das postagens, comentários e compartilhamentos dos usuários. As descrições dos metadados do arquivo *statsperday.tab* podem ser visualizados na Tabela 4.

TABELA 4 - DESCRIÇÕES DOS METADADOS DO ARQUIVO *STATSPERDAY.TAB*

Metadado	Descrição
day	Data da postagem
posts	Número de postagens
likes	Número de curtidas
reactions	Número de reações
comments	Número de comentários
shares	Número de compartilhamentos

FONTE: NetVizz (2018).

No arquivo *comments.tab*, que foi o selecionado para extrair a base de dados para este estudo, pode-se observar o tipo de publicação, o link de acesso, o conteúdo da mensagem, o link da imagem quando publicada, o link de acesso externo, a data da publicação e as curtidas, comentários e compartilhamentos dos usuários. As descrições dos metadados do arquivo *comments.tab* podem ser observados na Tabela 5.

TABELA 5 - DESCRIÇÕES DOS METADADOS DO ARQUIVO *COMMENTS.TAB*

Metadado	Descrição
position	Posição do comentário dentro da postagem
post_id	Código de identificação da postagem
post_by	Código de identificação do autor da postagem
post_text	Texto da postagem
post_published	Data de publicação da postagem
comment_id	Código de identificação do comentário
comment_by	Código de identificação do autor do comentário
is_reply	Resposta do comentário (0=não;1=sim)
comment_message	Mensagem do comentário
comment_published	Data de publicação do comentário
comment_like_count	Número de curtidas do comentário
attachment_type	Tipo de anexo do comentário
attachment_url	Link do anexo do comentário

FONTE: NetVizz (2018).

A base de dados foi extraída do arquivo comments.tab, este arquivo foi importado no programa Microsoft Excel®, utilizando o assistente de importação de texto, com a opção delimitado e os delimitadores tabulação e ponto e vírgula, a opção “meus dados possuem cabeçalho” selecionada e origem do arquivo no formato “65001: Unicode (UTF-8)”, para trazer o texto completo com todos os caracteres, seguido de avançar e concluir.

Na Tabela 6 podem ser visualizados os dados que foram extraídos do arquivo “page_209709705713052_2018_09_03_20_27_30_comments.tab” já importados no programa Microsoft Excel®, nela são exibidos dados brutos sem nenhuma formatação. A seguir foi salvo o arquivo como “Ford_Comments_Main.xlsx”.

TABELA 6 - DADOS EXTRAÍDOS DO NETVIZZ E IMPORTADOS NO MICROSOFT EXCEL

post_pi	comme	comme	is_repl	comment_message
2018-07-3	230151953	da39a3ee5	0	Não comprem! Estamos com um Ford Ka 2018 com problema de alto consumo de combustível e nem a concessionária, nem a Ford resolvem o problema. Estamos nessa brig
2018-07-3	230151953	da39a3ee5	0	Tenho um fusion com 60milkm rodado .. Simplesmente levei na concessionária por q ele apresentou uns tranco na troca de marcha . Custo 20 mil pra arrumar .. Estão de bri
2018-07-3	230151953	da39a3ee5	0	Tenho um ford ka 2015 que apresenta defeito de fabrica, um barulho na ré desde o primeiro dia que saiu da concessionária. Já foram trocados dois kit de embreagem e nad
2018-07-3	230151953	da39a3ee5	0	Desde dia 13/08 sem o ford ka 2015, um defeito de fábrica um barulho insuportável na ré e quando descola o carro do lugar, 2 kit de embreagem trocado, carro na autorizad
2018-07-3	230151953	da39a3ee5	0	Péssima experiência com os carros da Ford , estou com a segunda caminhonete ranger e as duas apresentam problemas e sofri a um acidente a três meses onde consegui re
2018-07-3	230151953	da39a3ee5	0	Tenho um Ka 2015, muito bom carro, nunca me deixou na mão, a Ford está de parabéns, agora já estou pensando no modelo 2019 automático, e o visual está ficou ainda me
2018-07-3	230151953	da39a3ee5	0	Na verdade não comprem o Ka eu tenho um que tirei zero com 12 mil km ele faz muito barulho nas portas não pode nem encostar o braço nela , as vezes quem está atrás qu
2018-07-3	230151953	da39a3ee5	0	Primeiro vamos concertar esse Powershift Ford , pelo amor de Deus ! Já fiz a reprogramação do câmbio na concessionária e ainda mesma coisa .
2018-07-3	230151953	da39a3ee5	0	Estou muito chateada acabei de adquirir um Ford Ka 2015 e nessa semana de chuva entrou muita água dentroooo estou passada sem saber o quê fazer e pior sem saber pir
2018-07-3	230151953	da39a3ee5	0	Em novembro de 2015 comprei um Ford Ka e faço todas as revisões. Meu carro já está com mais de 70000km. Hoje viajando furei o pneu e parei para trocar. Quando fui usar
2018-07-3	230151953	da39a3ee5	0	Não faça consórcio em empresa prestadora da Ford nem a Ford tem telefone para fazer reclamações e nem as empresas tem
2018-07-3	230151953	da39a3ee5	0	Tenho uma Ford Ranger 2017 que está com problema no sistema de navegação, ja fui DUAS vezes no distribuidor(Ford Horizonte Suzano) , com horario marcado. O senhor
2018-07-3	230151953	da39a3ee5	0	Não me pega nunca mais, menos de 6 meses um carro de mais de 80 mil reais já com problema é um absurdo, sem falar no péssimo atendimento do pós vendas.
2018-07-3	230151953	da39a3ee5	0	To com nome limpo ! Estou doido para comprar um carro, mais ficou muito burocrático . Ta louco!
2018-07-3	230151953	da39a3ee5	0	Tenho um Ford Ka 2018! Tem muito ruído no Painel e nas colunas já levei na concessionária não resolve nada. 😩
2018-07-3	230151953	da39a3ee5	0	estou muitíssimo chateado o meu carro além do problema com um cabo de acionamento do lado do motorista passando fora do conduite de segurança agora apresenta prc
2018-07-3	230151953	da39a3ee5	0	Tenho o novo Ford ka desde 2015 e estou muito satisfeito com ele. Pretendo trocá-lo apenas em 2020, mas apenas por outro!
2018-07-3	230151953	da39a3ee5	0	Eu tenho um Ford Ka sedan e o famoso que eu vi no comercial foi o Raul lemos
2018-07-3	230151953	da39a3ee5	0	Tenho um Ford Ka + ano 2018 excelente super economico, até agora não tenho que rechamar.
2018-07-3	230151953	da39a3ee5	0	Eu andei no Ka 2017....88000km pensa no super carro.... parabéns Ford

FONTE: O autor (2018).

3.2.2 Descoberta de Conhecimento em Texto

O método utilizado neste trabalho é baseado no processo de Descoberta de Conhecimento em Texto (DCT) que tem como objetivo a aquisição de conhecimento através das informações obtidas na transformação de forma automática de dados textuais. Seguindo o fluxo apresentado na revisão de literatura, Figura 1, página 28, após a pré-seleção da base de dados (apresentada na seção anterior) são realizadas as etapas de pré-processamento, mineração de dados (para fins desta pesquisa, mineração de opiniões) e avaliação (pós-processamento), conforme detalhado nesta seção.

Segundo Lunardi, Viterbo e Bernardini (2015), a mineração de opiniões por meio de técnicas de aprendizado de máquina supervisionado são as que têm maior utilidade para a definição de modelos para classificação de opiniões. Para os autores, a pesquisa de mineração de opinião em análise de sentimentos pode se classificar em:

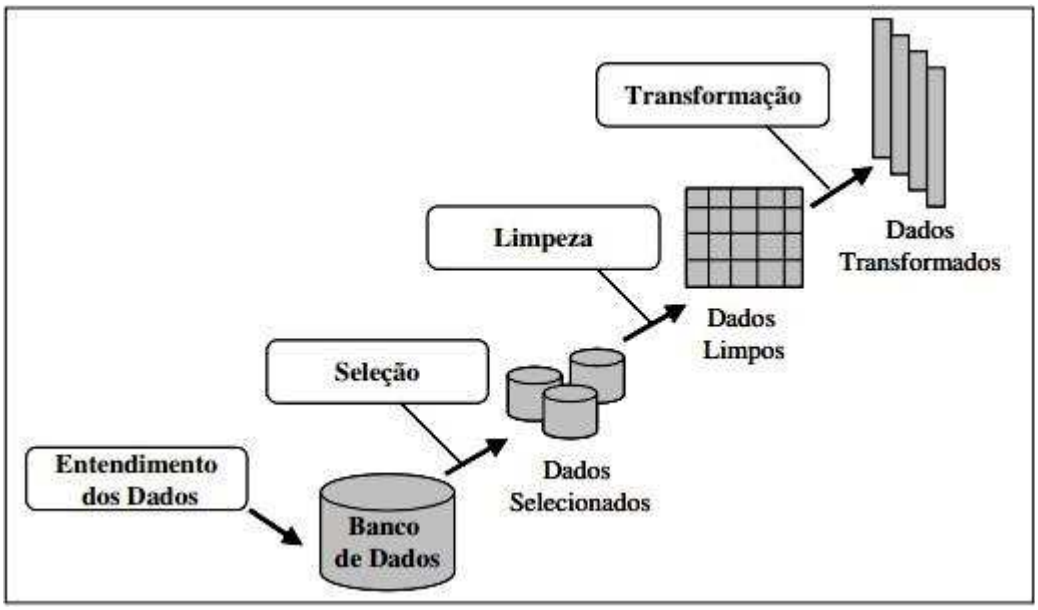
- palavras-chave e afinidade léxica: a classificação do texto é efetuada com palavras unívocas, ou seja, que não dão margem a outras interpretações, com a detecção de palavras obvias e a atribuição de afinidade com um sentimento para outras palavras;
- aprendizado de máquina: com a utilização de algoritmos de aprendizado de máquina tais como *Naive Bayes* e SVM, utilizados para classificação de textos;
- métodos estatísticos: calculam a polaridade de palavras com base na ocorrência simultânea de uma palavra com outras que tenham a mesma orientação;
- baseado em conceitos: pelo uso de ontologias para fazer a análise de texto, por meio da análise de expressões que não contenham emoções explícitas, porém que se relacionem a um sentimento de forma implícita.

Para esta pesquisa foram usados algoritmos que utilizam o aprendizado de máquina supervisionado, com o método de processamento de linguagem natural relacionado com análise de sentimentos dos consumidores quanto a produtos e serviços nos conceitos positivo, negativo e neutro, conforme apresentado na próxima seção.

3.2.2.1 Pré-processamento

A fase do pré-processamento de dados visa uma prévia preparação dos dados para depois serem submetidos ao processo de mineração de opinião. Segundo Neves (2003), o pré-processamento se divide em quatro etapas: entendimento dos dados, seleção, limpeza e transformação. A etapa de entendimento compreende a análise dos dados coletados e a definição da sua importância e significado. A seleção abrange a definição de quais dados deverão ser utilizados para serem submetidos à mineração e quais deverão ser descartados. A limpeza dos dados consiste em elevar a qualidade dos dados para níveis propícios para o processo de mineração. A etapa de transformação visa preparar os dados de forma a estarem adequados para a técnica de mineração que foi utilizada, as operações envolvidas nesta etapa são a normalização, conversão de valores, discretização e composição dos atributos (NEVES, 2003). A divisão do pré-processamento de dados pode ser observada na Figura 15.

FIGURA 15 – DIVISÃO DAS ETAPAS DE PRÉ-PROCESSAMENTO DE DADOS



FONTE: Neves (2003).

Castro e Ferrari (2016), afirmam que as principais tarefas de pré-processamento são:

- limpeza: para introdução de valores ausentes, retirada de ruídos e correção de inconsistências;

- integração: para poder agrupar dados de várias fontes em um só local;
- redução: para diminuir o tamanho da base de dados, seja agrupando ou excluindo-se atributos redundantes, sumarização da base ou redução de objetos da base;
- transformação: para permitir que sejam padronizados dados de maneira a ficarem com o formato adequado para a aplicação de distintas formas de mineração;
- discretização: para possibilitar o uso de da base em metodologias que utilizam somente atributos nominais, e assim, ampliar a empregabilidade para um conjunto maior de problemas. Além de permitir que a quantidade de valores de um determinado atributo seja reduzida.

Na Figura 16 podem ser vistas as etapas do processo de preparação da base de dados, incluindo o pré-processamento.

FIGURA 16 – ETAPAS DO PROCESSO DE PREPARAÇÂO DA BASE DE DADOS

FONTE: Castro e Ferrari (2016, p. 35).

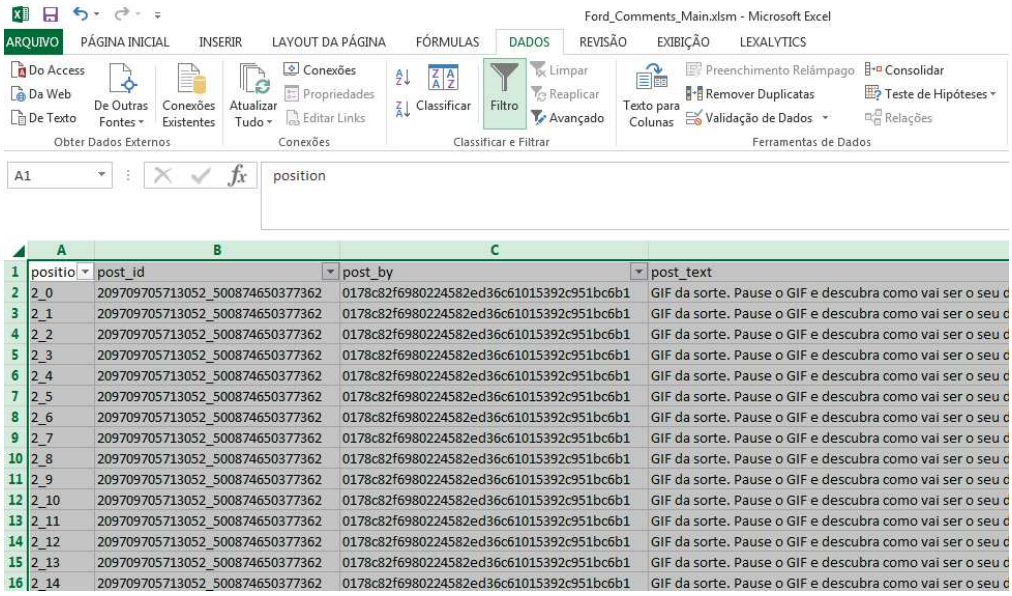
Para a etapa de pré-processamento deste estudo foram adotadas as seguintes ferramentas:

- **Microsoft Excel©:** foi utilizado inicialmente na etapa do pré-processamento o Microsoft Excel® para fazer a **seleção** dos dados e a **redução** da quantidade de objetos da base que será utilizada neste estudo, bem como uma **limpeza** prévia de registros (linhas) em branco.

  Primeiramente foi aberto o arquivo “Ford_Comments_Main.xlsx”, preparado anteriormente na etapa de coleta de dados no NetVizz. Esta base originalmente contava com 26.589 registros (linhas) e 13 atributos (colunas). Foram

selecionados na planilha todos os registros e atributos e em seguida na opção “DADOS” do Microsoft Excel® foi habilitada a opção filtro para permitir que fossem selecionados somente os registros (linhas) a utilizar neste trabalho. Esta opção pode ser visualizada na Figura 17.

FIGURA 17 – SELEÇÃO DE DADOS NO EXCEL PARA FILTRAGEM DE REGISTROS

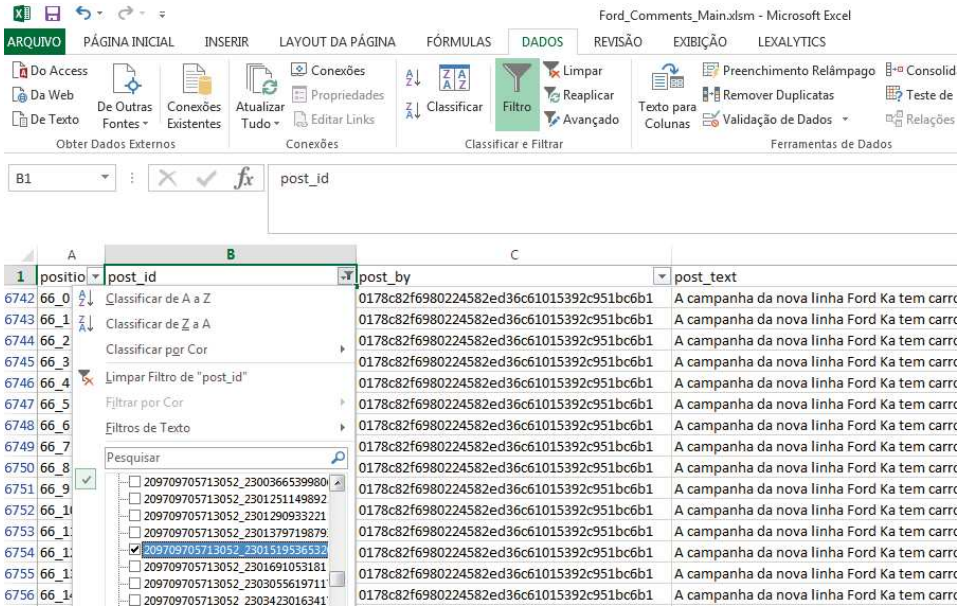


	A	B	C	
1	positio	post_id	post_by	post_text
2	2_0	209709705713052_500874650377362	0178c82f6980224582ed36c61015392c951bc6b1	GIF da sorte. Pause o GIF e descubra como vai ser o seu d
3	2_1	209709705713052_500874650377362	0178c82f6980224582ed36c61015392c951bc6b1	GIF da sorte. Pause o GIF e descubra como vai ser o seu d
4	2_2	209709705713052_500874650377362	0178c82f6980224582ed36c61015392c951bc6b1	GIF da sorte. Pause o GIF e descubra como vai ser o seu d
5	2_3	209709705713052_500874650377362	0178c82f6980224582ed36c61015392c951bc6b1	GIF da sorte. Pause o GIF e descubra como vai ser o seu d
6	2_4	209709705713052_500874650377362	0178c82f6980224582ed36c61015392c951bc6b1	GIF da sorte. Pause o GIF e descubra como vai ser o seu d
7	2_5	209709705713052_500874650377362	0178c82f6980224582ed36c61015392c951bc6b1	GIF da sorte. Pause o GIF e descubra como vai ser o seu d
8	2_6	209709705713052_500874650377362	0178c82f6980224582ed36c61015392c951bc6b1	GIF da sorte. Pause o GIF e descubra como vai ser o seu d
9	2_7	209709705713052_500874650377362	0178c82f6980224582ed36c61015392c951bc6b1	GIF da sorte. Pause o GIF e descubra como vai ser o seu d
10	2_8	209709705713052_500874650377362	0178c82f6980224582ed36c61015392c951bc6b1	GIF da sorte. Pause o GIF e descubra como vai ser o seu d
11	2_9	209709705713052_500874650377362	0178c82f6980224582ed36c61015392c951bc6b1	GIF da sorte. Pause o GIF e descubra como vai ser o seu d
12	2_10	209709705713052_500874650377362	0178c82f6980224582ed36c61015392c951bc6b1	GIF da sorte. Pause o GIF e descubra como vai ser o seu d
13	2_11	209709705713052_500874650377362	0178c82f6980224582ed36c61015392c951bc6b1	GIF da sorte. Pause o GIF e descubra como vai ser o seu d
14	2_12	209709705713052_500874650377362	0178c82f6980224582ed36c61015392c951bc6b1	GIF da sorte. Pause o GIF e descubra como vai ser o seu d
15	2_13	209709705713052_500874650377362	0178c82f6980224582ed36c61015392c951bc6b1	GIF da sorte. Pause o GIF e descubra como vai ser o seu d
16	2_14	209709705713052_500874650377362	0178c82f6980224582ed36c61015392c951bc6b1	GIF da sorte. Pause o GIF e descubra como vai ser o seu d

FONTE: O autor (2018).

A seguir foi efetuada a filtragem no atributo (coluna) “post_id” com o código que identifica exclusivamente a postagem de 30 de julho de 2018, sobre a nova linha Ford Ka 2018, que foi a base de estudo deste trabalho. O código “209709705713052_2301519536532048” é o que corresponde à postagem desejada. Com essa filtragem o número de registros (linhas) foi reduzido para 1007 registros (linhas). A filtragem aplicada pode ser vista na Figura 18.

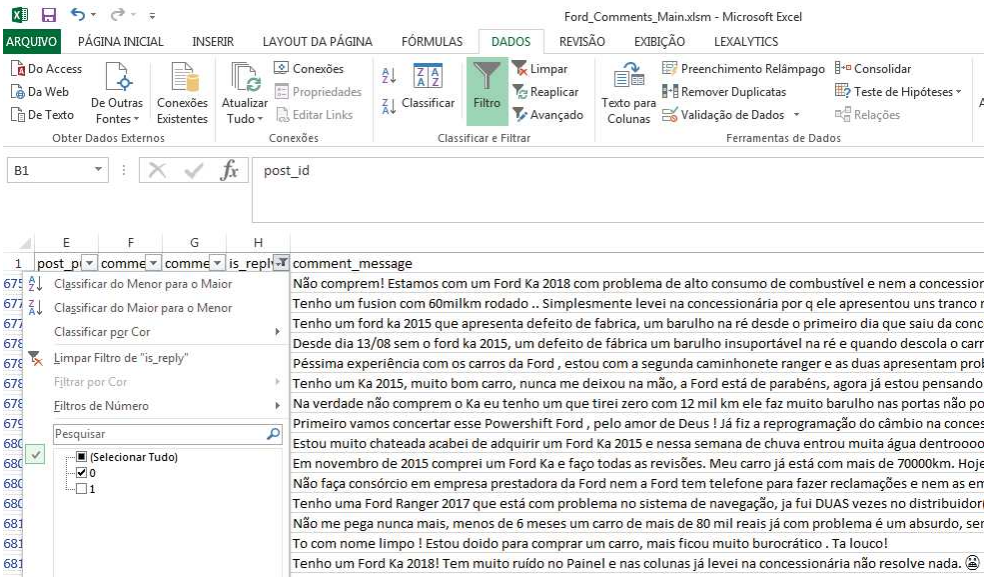
FIGURA 18 – SELEÇÃO DA POSTAGEM NOVA LINHA FORD KA 2018



FONTE: O autor (2018).

A seguir, foi feita uma filtragem no atributo (coluna) "*is_reply*", selecionando-se apenas o valor "0" que identifica somente os comentários feitos e não a resposta dos comentários que é o valor "1". Esta seleção é devida a que a base de estudo neste trabalho são os comentários feitos pelas pessoas e não as respostas efetuadas pela empresa. Com essa nova filtragem o número de registros (linhas) foi reduzido para 500 registros (linhas). O detalhe desta seleção pode ser visualizado na Figura 19.

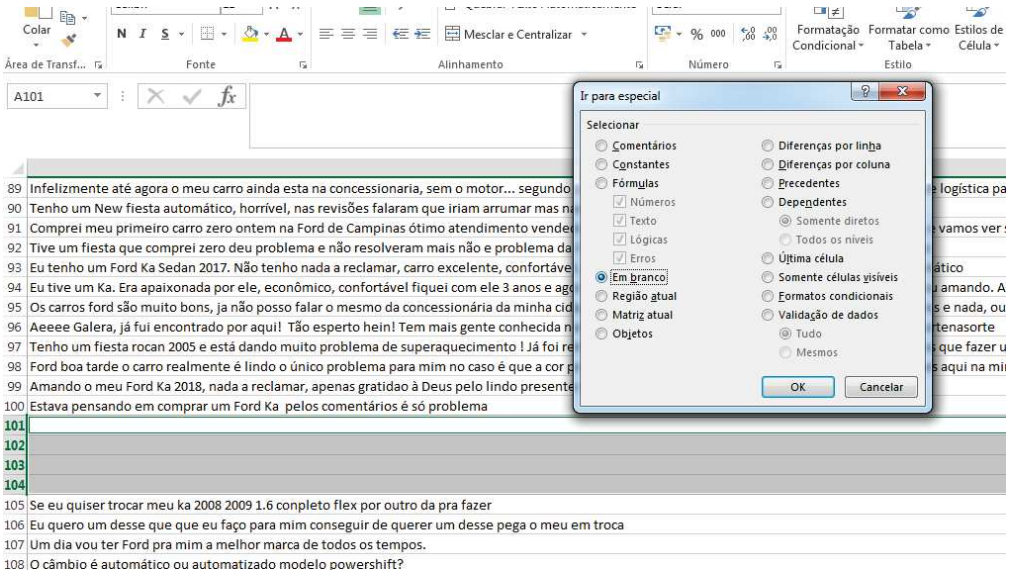
FIGURA 19 – SELEÇÃO DOS COMENTÁRIOS NO ATRIBUTO "IS_REPLY"



FONTE: O autor (2018).

Uma vez feita a filtragem desejada, foram excluídos os atributos (colunas), que não foram utilizadas no estudo. Desta forma, dos 13 atributos (colunas) iniciais da base, foi mantido unicamente o atributo (coluna) "*comment_message*" que é o atributo onde está registrada a mensagem do comentário efetuado pelas pessoas. Em seguida foi efetuada uma limpeza prévia da base, excluindo-se todas os registros (linhas) com valores ausentes (linhas em branco). Para este procedimento foram usadas as opções no Microsoft Excel®: "página inicial", "localizar e selecionar", "ir para especial", "em branco" e comando "excluir". Conforme pode ser visualizado na figura 20.

FIGURA 20 – LIMPEZA DE REGISTROS (LINHAS) EM BRANCO



FONTE: O autor (2018).

Após o pré-processamento no Microsoft Excel®, a base de dados ficou com 298 registros (linhas) e 1 atributo (coluna).

– **Python**: após o uso da ferramenta Microsoft Excel® na primeira etapa de pré-processamento, foi utilizado um código de programação desenvolvido em linguagem Python versão 3 por Rodrigues (2017). Este código com o nome "preprocessamento.py" pode ser visualizado nos anexos A, B e C e tem a finalidade de efetuar: **limpeza** e **transformação** de dados, primeiramente por padronização, que segundo Castro e Ferrari (2016), faz-se necessário para solucionar diferenças de unidades e escalas dos dados. Um dos casos dentro

da padronização é o de capitalização, que consiste em padronizar os dados nominais que podem surgir em maiúsculo, minúsculo ou ambos os casos. Para este estudo foi definido que todos os textos da base seriam transformados em minúsculo. Segundo Castro e Ferrari (2016), essa padronização de capitalização é necessária para evitar inconsistências com ferramentas que possuem sensibilidade a letras com capitalização variada.

Outra padronização necessária e efetuada pelo código “preprocessamento.py” é a de retirada de caracteres especiais e de acentuação, que segundo Castro e Ferrari (2016), podem entrar em conflito com ferramentas de mineração com conjuntos de determinado idioma. Para este estudo adicionalmente foram também retirados os caracteres numéricos para obter-se melhores resultados.

Segundo Rodrigues (2017), é necessário a transformação de abreviações para palavras que expressam o seu sentido, pois ao transformar-se as mesmas em palavras melhora a base de dados, já que se forem mantidas, aumenta a chance de serem apagadas por remoção de palavras indesejadas. Para melhor adequar as abreviações da base a ser estudada foram retiradas do código as seguintes: “migs”, “oq” e “eua”. A relação definida por Rodrigues (2017) pode ser vista na tabela 7.

TABELA 7 – ABREVIAÇÕES E CORRESPONDÊNCIAS

Abreviação	Representação
blz	beleza
flw	tchau
vlw	obrigado
ta	esta
mt	muito
q	que
n	não
s	sim
pq	porque
ok	beleza
vcs	voces
vc	voce
amr	amor
migo	amigo
migs	amigo
okz	beleza
oks	beleza

FONTE: Rodrigues (2017).

Para Rodrigues (2017), no pré-processamento, procura-se não apagar informações que possam vir a ser úteis, com isso o autor incluiu no código “preprocessamento.py” a transformação de *emoticons* contidos na base para as respectivas palavras correspondentes. O autor menciona que a função em Python utilizada para executar este processo foi a ()*replace* e que foram criadas duas listas de *strings* e uma variável de controle, com isso quando o programa encontra o conjunto de caracteres que configura um *emoticon* contido na primeira lista, o mesmo é substituído pela palavra de seu significado correspondente na segunda lista. A relação de *emoticons* com a respectiva correspondência definida por Rodrigues (2017), pode ser vista na tabela 8.

TABELA 8 – TRANSFORMAÇÃO DE *EMOTICONS* EM PALAVRAS CORRESPONDENTES

(:	sorrindo
:)	sorrindo
=	sorrindo
(=	sorrindo
:D	sorrindo
D:	surpreso
;)	piscando
(;	piscando
xd	sorrindo
:O	surpreso
:P	lingua de fora
<2	amor
<3	amor
><	gostei
s2	amor
sz	amor
u.u	prevalecido
:@	bravo
:/	indeciso
:'(	chorando
:9	gostando
:x	aborrecido
_	gostando

FONTE: Rodrigues (2017).

Segundo Rodrigues (2017), a medida que vão ocorrendo transformações nas bases, podem aparecer alguns espaços em branco desnecessários e que por tal motivo foi criada a função de retirada de espaços em branco que segue o mesmo princípio das demais transformações do código, fazendo que quando houverem espaços duplos, triplos ou quádruplos sejam reduzidos a um espaço simples.

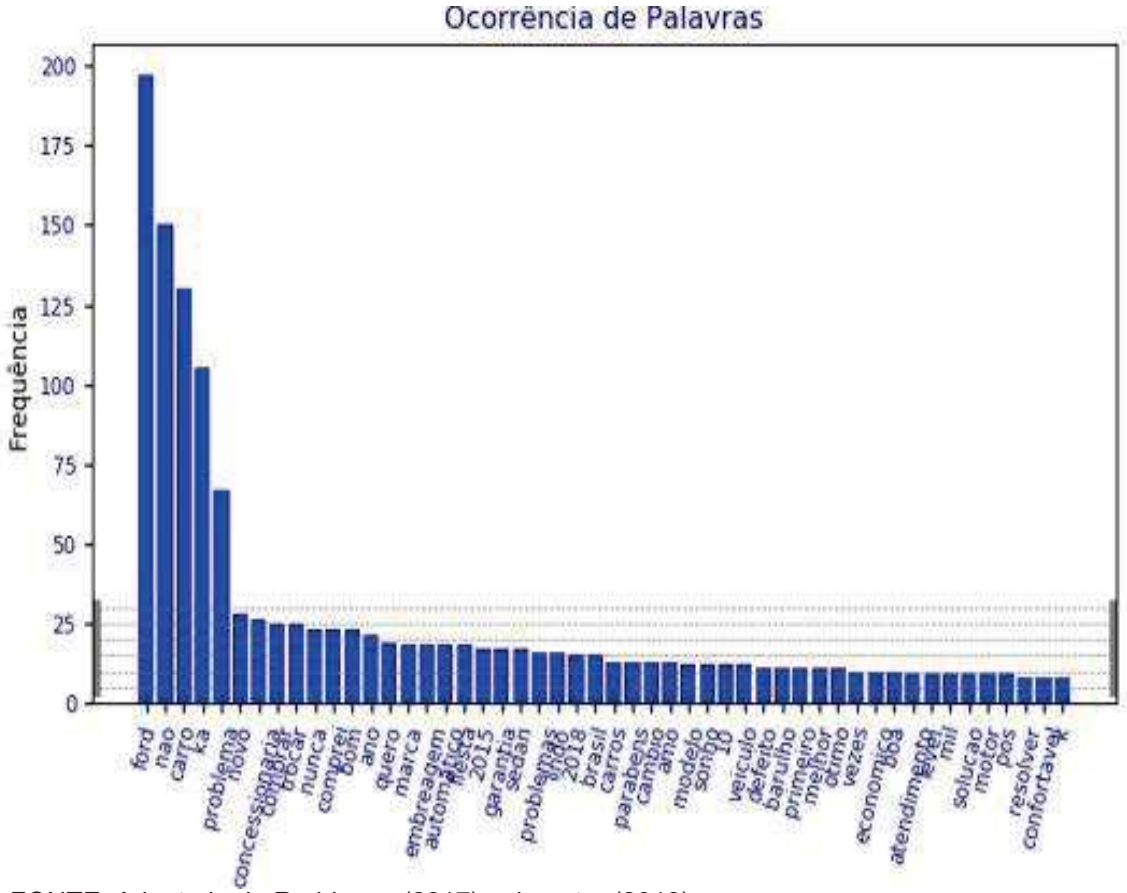
Conforme Rodrigues (2017), para a remoção de *stopwords* (palavras irrelevantes), foi utilizada a biblioteca “NTLK.CORPUS” com a lista de *stopwords*. Para este trabalho, foram alteradas e adicionadas algumas novas *stopwords* de acordo com a necessidade por meio do método “*stop_words.update*”.

Para poder submeter a base de dados gerada anteriormente em Microsoft Excel® com extensão “CSV” ao código em Python desenvolvido por Rodrigues (2017), foi necessário abrir previamente o arquivo com extensão “CSV” com o uso do programa “Bloco de Notas” que faz parte do Sistema Operacional Microsoft Windows® e salvar o arquivo como texto e extensão “txt” com o nome “processar.txt”. Este nome de arquivo precisa ser obedecido pois o código internamente abre o arquivo “processar.txt” e após o processamento gera o arquivo “resultado.txt” já devidamente processado com a limpeza e transformação programadas no código.

De forma a poder avaliar melhor a performance da ferramenta Semantria (utilizada na última fase de pré-processamento), foram usados os códigos dos anexos A, B e C que são variantes do código desenvolvido por Rodrigues (2017). A diferença está em que o código do anexo A retira todas as *stopwords* programadas exceto o “nao”, o código do anexo B retira todas as *stopwords* programadas inclusive o “nao” e o anexo C não retira as *stopwords*. Com esses três arquivos “txt” gerados mais um arquivo “txt” sem nenhum tipo de tratamento do código Python foram submetidos ao processo de análise de texto na ferramenta Semantria. Para poder-se verificar graficamente a frequência de palavras obtida em cada um dos quatro arquivos “txt”, foi utilizado um código em linguagem Python versão 3, constante no anexo D e desenvolvido por Rodrigues (2017). A diferença de frequência de palavras obtidas pode ser observada nas Figuras 21 a 24.

Com a exclusão dos *stopwords* da base excetuando a palavra “nao”, a ocorrência de palavras com maior relevância se destaca. A palavra “nao” foi mantida por dar sentidos diferentes às frases. Por exemplo: “comprem este carro” e “nao comprem este carro”. Na Figura 21 pode-se observar uma frequência maior das palavras: “ford”, “nao”, “carro”, “ka” e “problema”.

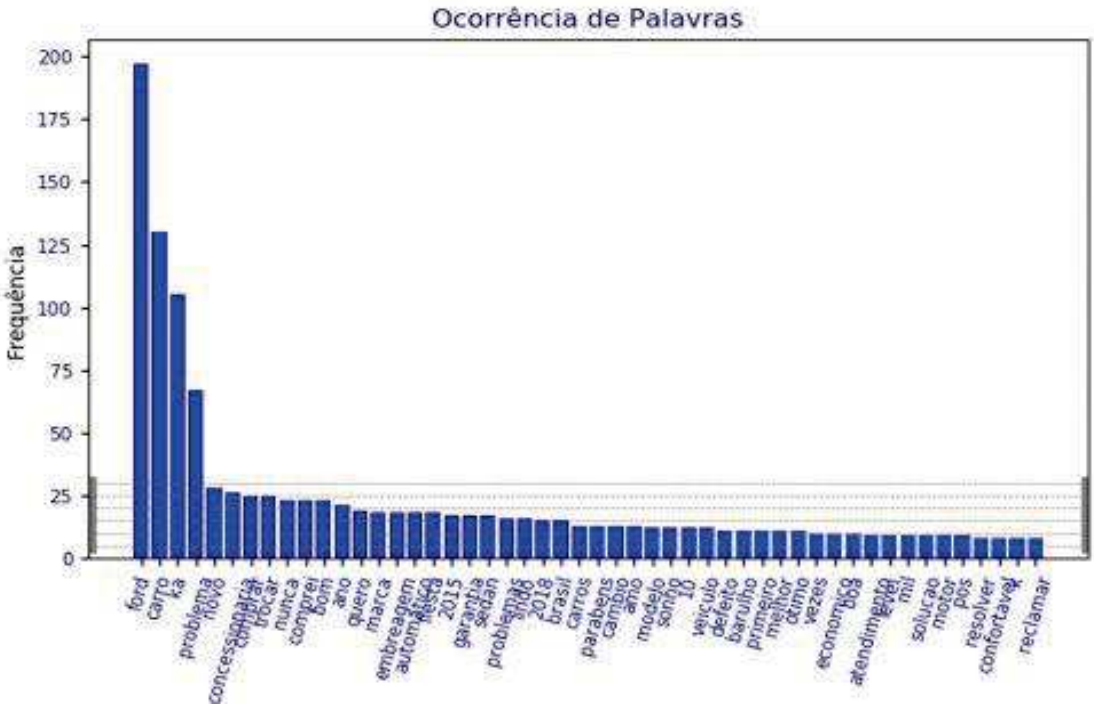
FIGURA 21 – FREQUÊNCIA DE PALAVRAS RETIRANDO *STOPWORDS* EXCETO O “NAO”



FONTE: Adaptado de Rodrigues (2017) pelo autor (2018).

Para a opção de exclusão de todos os *stopwords* da base incluindo a palavra “nao”, percebe-se uma frequência das palavras semelhante ao da Figura 21 com o surgimento da palavra “reclamar” como pode ser observado na Figura 22.

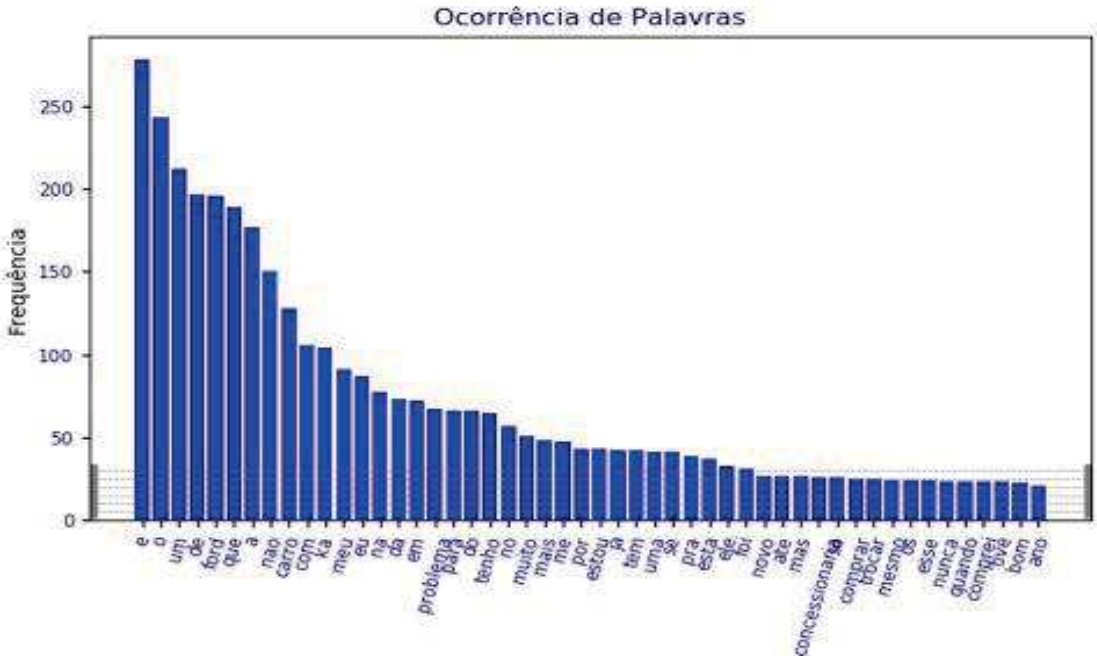
FIGURA 22 - FREQUÊNCIA DE PALAVRAS RETIRANDO TODOS OS STOPWORDS



FONTE: Adaptado de Rodrigues (2017) pelo autor (2018).

Para a opção da não retirada das *stopwords* da base, há uma frequência elevada de palavras tais como: “e”; “o”, “um”, “de”, “que”, “a”, etc. Essa ocorrência elevada de *stopwords* pode ser vista na Figura 23.

FIGURA 23 - FREQUÊNCIA DE PALAVRAS SEM RETIRADA DE *STOPWORDS*



FONTE: Adaptado de Rodrigues (2017) pelo autor (2018).

Para a opção de nenhum tratamento na base, é possível perceber no gráfico de frequência além das *stopwords* o surgimento de alguns caracteres especiais como: “$$”, “0”, “ ‘ “ e palavras acentuadas como: “está” e “já”. O surgimento de *stopwords* e caracteres especiais pode ser observado na Figura 24.

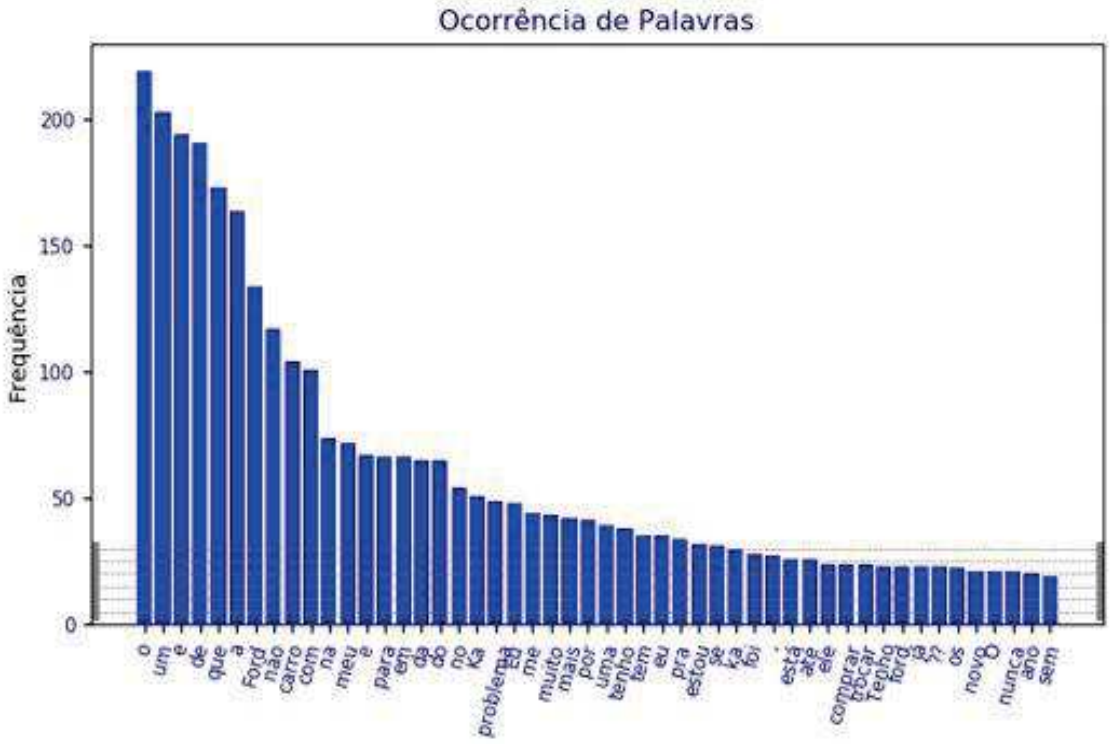


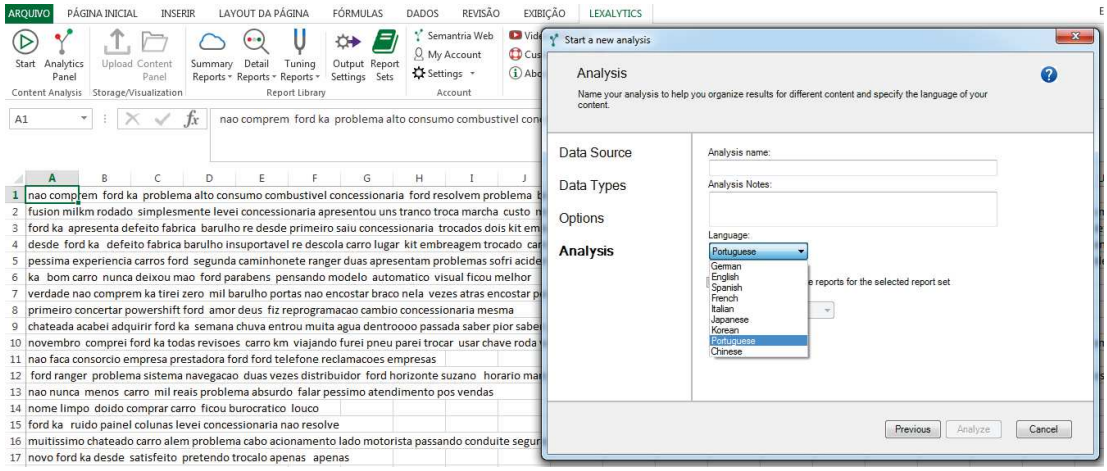
FIGURA 24 – FREQUÊNCIA DE PALAVRAS DOS DADOS BRUTOS
FONTE: Adaptado de Rodrigues (2017) pelo autor (2018).

- **Semantria:** para efetuar o processo de análise de texto foi utilizada a ferramenta Semantria na versão 2013 x64 6.0.102 que é uma solução de análise de texto e sentimento para pesquisas, mídias sociais e análises, desenvolvida pela empresa Lexalytics. Segundo a Lexalytics (2018), a análise de sentimentos é baseada no processo de determinar se partes textuais são positivas, negativas ou neutras. O site descreve que um sistema de análise de sentimentos para análise de texto combina o processamento de linguagem natural (PLN) e técnicas de aprendizado de máquina para atribuir pontuações de sentimento ponderadas às entidades, tópicos, temas e categorias dentro de uma sentença ou uma frase. A ferramenta Semantria é disponibilizada pela Lexalytics via *Application Programming Interface* (API) e em *plugin* para o Microsoft Excel®. Ambas as versões são pagas, porém é possível utilizar a

versão teste para avaliação que permite fazer um total de 10 análises completas por um período de 14 dias, mediante o cadastro de um usuário e senha com o uso de um e-mail corporativo.

Uma vez instalada a ferramenta, a opção Lexalytics é incorporada no menu do Microsoft Excel®. A seguir deve ser aberta a base a ser analisada em formato planilha. Para dar início à análise deve ser acessada a opção Lexalytics e a opção “Start” que abrirá uma janela para que seja informada a base a ser analisada, os atributos (colunas) e a quantidade de registros (linhas). Deve ser informado o idioma em que a base será analisada e finalmente deve ser acessada a opção “*Analyze*” (analisar). Conforme pode ser visto na Figura 25.

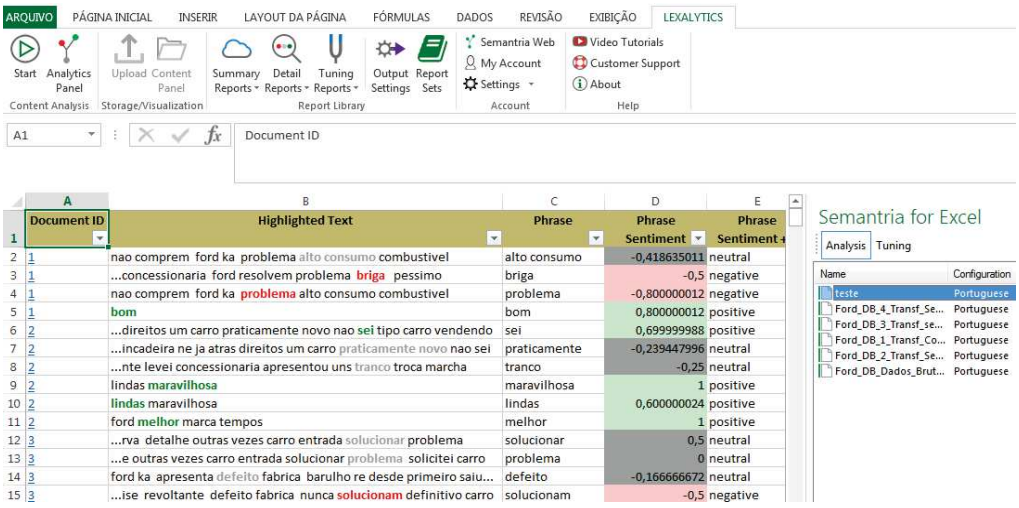
FIGURA 25 – TELA INICIAL PARA ANÁLISE SEMANTRIA



FONTE: Semantria (2018).

Após a execução da análise, para visualização dos resultados, é necessário acessar a opção “*Analytics Panel*” e selecionar o nome da análise desejada. As opções “*Summary Reports”* e “*Detail Reports*” fazem com que as diferentes análises sejam geradas. Este processo pode ser visualizado na Figura 26.

FIGURA 26 – TELA PARA GERAÇÃO DE RESULTADOS NO SEMANTRIA



	A	B	C	D	E
1	Document ID	Highlighted Text	Phrase	Phrase Sentiment	Phrase Sentiment
2	1	nao comprem ford ka problema alto consumo combustivel	alto consumo	-0,418635011	neutral
3	1	...concessionaria ford resolvem problema briga pessimo	briga	-0,5	negative
4	1	nao comprem ford ka problema alto consumo combustivel	problema	-0,800000012	negative
5	1	bom	bom	0,800000012	positive
6	2	...direitos um carro praticamente novo nao sei tipo carro vendendo	sei	0,699999988	positive
7	2	...incadeira ne ja atras direitos um carro praticamente novo nao sei	praticamente	-0,239447996	neutral
8	2	...nte levei concessionaria apresentou uns tranco troca marcha	tranco	-0,25	neutral
9	2	lindas maravilhosa	maravilhosa	1	positive
10	2	lindas maravilhosa	lindas	0,600000024	positive
11	2	ford melhor marca tempos	melhor	1	positive
12	3	...rva detalhe outras vezes carro entrada solucionar problema	solucionar	0,5	neutral
13	3	...e outras vezes carro entrada solucionar problema solicitei carro	problema	0	neutral
14	3	ford ka apresenta defeito fabrica barulho re desde primeiro saiu...	defeito	-0,166666672	neutral
15	3	...ise revoltante defeito fabrica nunca solucionam definitivo carro	solucionam	-0,5	negative

FONTE: Semantria (2018).

3.2.2.2 Processamento

A etapa de processamento é constituída pela mineração de opinião, que visa extrair conhecimento através do descobrimento de padrões e regras que possam atribuir significado nos dados que estão sendo submetidos. Para este estudo optou-se pela utilização do *software* Weka, com aplicação de algoritmos de classificação, por serem mais conhecidos e adequados na associação de dados com conjuntos de objetos que tem definição prévia de classes.

O programa selecionado para a mineração na base de dados estudada neste trabalho foi o Weka versão 3.8.1. Castro e Ferrari (2016) afirmam que existem diversos *software* para a mineração de dados, sendo alguns pagos e outros gratuitos. Segundo os autores, o Weka está entre os mais utilizados, é gratuito e de código aberto. O *software* foi desenvolvido em linguagem Java e é mantido pela Universidade de Waikato localizada na Oceania.

O Weka permite que o usuário execute tarefas de pré-processamento, classificação, agrupamento e visualização dos dados através de sua interface gráfica. Castro e Ferrari (2016) afirmam ainda que é possível executar análises mais complexas através da criação de fluxogramas que concatenam tarefas de mineração de dados.

Como a base de dados utilizada para a mineração é composta por um atributo do tipo *PhraseSentiment* (numérico) e por um atributo do tipo *string* (sequência de caracteres) alguns algoritmos de mineração não podem ser utilizados diretamente

pois são incompatíveis com bases que contenham atributos com estas características. Por esse motivo os algoritmos considerados neste trabalho foram executados no *software* Weka com o uso de filtros, como é o caso do "*FilteredClassifier*" que habilita a discretização de atributos nominais vazios, atributos do tipo *string*, valores ausentes, atributos relacionais, atributos binários, numéricos, nominais, entre outros. Também foi utilizado o filtro não supervisionado "*StringToWordVector*" que converte os atributos *string* em um conjunto de atributos que representam informações de ocorrências de palavras. Adicionalmente também foi utilizado um "*tokenizer*" que é uma operação que permite dividir uma sequência de *strings* em partes como: frases, palavras, símbolos e outros elementos denominados "*tokens*" para aprimorar a mineração de opinião.

Os algoritmos escolhidos para a mineração da base selecionada dentro do *software* Weka foram o *Naive Bayes* que usa o modelo de classificação probabilístico, o SMO (*Sequential Mininal Optimization*) que utiliza o modelo baseado em função e o J48 que é um algoritmo baseado em árvores. Estes algoritmos foram selecionados devido a serem mais comumente utilizados e mais adequados ao tipo de base de dados que está sendo analisada.

O processo de configuração e execução dos algoritmos no Weka foi efetuado primeiramente selecionando a base em formato "arff" conforme modelo do apêndice A, já previamente pré-processada no *software* Semantria.

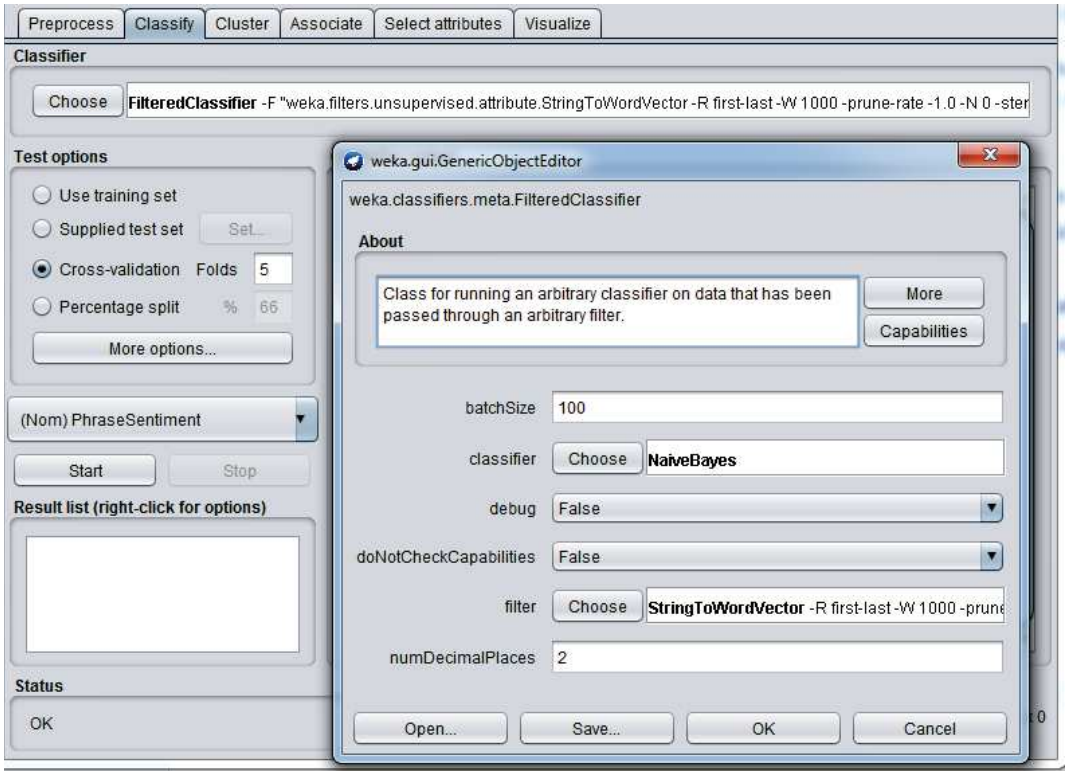
Na opção "*Classify*" optou-se pelos filtros "*FilteredClassifier*" e "*StringToWordVector*" conforme justificado nesta seção, aliado à escolha do primeiro algoritmo selecionado que foi o *Naive Bayes*.

O *Naive Bayes*, segundo Castro e Ferrari (2016), é um algoritmo classificador estatístico (probabilístico), utilizado para poder estimar a probabilidade de um objeto pertencer a uma dada classe, onde o efeito do valor de um determinado atributo em uma certa classe independe dos valores dos outros atributos (independência condicional da classe). Os autores afirmam que estes algoritmos simples de classificação bayesiana possuem desempenho comparável a redes neurais artificias e a árvores de decisão além de apresentarem alta acuracidade e velocidade de processamento em grandes bases de dados.

Como opção de teste foi decidido pela validação cruzada, que segundo Castro e Ferrari (2016), tem como objetivo dividir a base de dados em grupos de treinamento e teste de forma que os dados de treinamento sejam usados para que os parâmetros livres do modelo possam ser ajustados e os dados para teste possam gerar uma

estimativa para que o modelo generalize dados não usados no treinamento. Os autores afirmam que uma forma bastante comum de validação cruzada em mineração é a denominada “validação cruzada em k-pastas”, que compreende a divisão dos dados em subconjuntos (pastas) que são usadas para treinamento e uma dessas pastas é usada para teste. O número de pastas corresponde ao número de “rodadas” de treinamento que será efetuado para se estimar o erro. O número de pastas mais usual em validação cruzada segundo os autores é de 10 que é a quantidade padrão em termos práticos. As configurações descritas podem ser visualizadas na Figura 27.

FIGURA 27 – CONFIGURAÇÕES INICIAIS NO *SOFTWARE* WEKA



FONTE: Weka (2018).

As mesmas configurações foram adotadas para os demais métodos selecionados SMO e J48.

O SMO é um algoritmo baseado em função. Segundo Cristianini e Shawe-Taylor (2013) o SMO é um algoritmo heurístico que utiliza duas variáveis em cada iteração, assim gera uma solução analítica na iteração. O objetivo da heurística SMO é obter os valores dos multiplicadores de Lagrange para que os seus erros tendam a zero, isto acontece quando um dos multiplicadores é atualizado, então o outro é

ajustado para manter a condição verdadeira. A atualização dos valores é efetuada de forma analítica.

O J48 é um algoritmo de árvore de decisão que implementa o C4.5 proposto por Quinlan (1993). Segundo o autor o C4.5 tem a capacidade de agrupar valores de atributos, de forma que em casos que um atributo tenha diversos valores que pertençam a uma mesma classe, cria-se um ramo para esses valores.

Segundo Castro e Ferrari (2016), a árvore de decisão é uma estrutura que tem forma de árvore, onde cada “nó interno” está ligado a um teste de determinado atributo, cada “ramo” corresponde a um resultado do teste, e os “nós folhas” correspondem às classes ou às distribuições das classes. O nó que estiver mais acima na árvore é denominado “nó raiz”, onde cada “caminho” do “nó raiz” até o “nó folha” corresponde a uma regra de classificação.

3.2.2.3 Pós-processamento

A fase de pós-processamento visa explicar as formas de avaliação dos resultados obtidos na análise efetuada com o *software* Weka com os três algoritmos (seção 3.2.2.2). Os resultados que foram utilizados neste estudo são:

- modelo de classificação *(classifier model)*: apresenta os resultados e as representações textuais dos modelos de classificação que foram usados nos dados do treinamento em cada um dos algoritmos. Nas Figuras 28 a 30 podem ser visualizados a forma de cálculo de cada algoritmo.

  O SMO embora seja um algoritmo simples baseado em função, o seu cálculo é de difícil interpretação nos resultados gerados no modelo de classificação. Segundo Castro e Ferrari (2016), são modelos baseados em funções predefinidas e os seus parâmetros se ajustam durante o treinamento, posteriormente apresenta-se à função um novo objeto de classe desconhecida, esse novo valor é calculado e representa a classe do novo objeto. Na Figura 28 podem ser visualizados os valores gerados no treinamento já normalizados.

FIGURA 28 – MODELO DE CLASSIFICAÇÃO SMO

```
Classifier Model
SMO

Kernel used:
  Linear Kernel: K(x,y) = <x,y>

Classifier for classes: positive, neutral

BinarySMO

Machine linear: showing attribute weights, not support vectors.

         0.3992 * (normalized) a
 +      -0.0897 * (normalized) abraco
 +      -0.0078 * (normalized) abs
 +      -0.2189 * (normalized) acabamento
 +      -0.3383 * (normalized) acabou
 +      -0.2936 * (normalized) acessivel
 +       0.0652 * (normalized) acha
 +       0.0356 * (normalized) achando
 +      -0.2424 * (normalized) aconteceu
 +      -0.2338 * (normalized) adorei
 +      -0.045  * (normalized) adoro
 +      -0.1104 * (normalized) adquiri
 +      -0.0882 * (normalized) adquirindo
 +      -0.0607 * (normalized) afirmo
 +      -0.2385 * (normalized) aguardand
 +      -0.0712 * (normalized) ajudar
 +      -0.0616 * (normalized) alegria
 +      -0.2757 * (normalized) alo
 +       0.0088 * (normalized) alto
 +      -1.053  * (normalized) amando
```

FONTE: Weka (2018).

A interpretação dos cálculos gerados no modelo de classificação probabilístico do algoritmo *Naive Bayes* é mais intuitiva quando comparada à saída do método SMO. Segundo Castro e Ferrari (2016), os modelos probabilísticos atribuem a probabilidade de um objeto pertencer a uma ou mais classes possíveis por meio de medidas estatísticas. Na Figura 29 podem ser vistos os resultados estatísticos para cada classe em relação a cada atributo, ocorridos durante o treinamento.

FIGURA 29 – MODELO DE CLASSIFICAÇÃO *NAIVE BAYES*

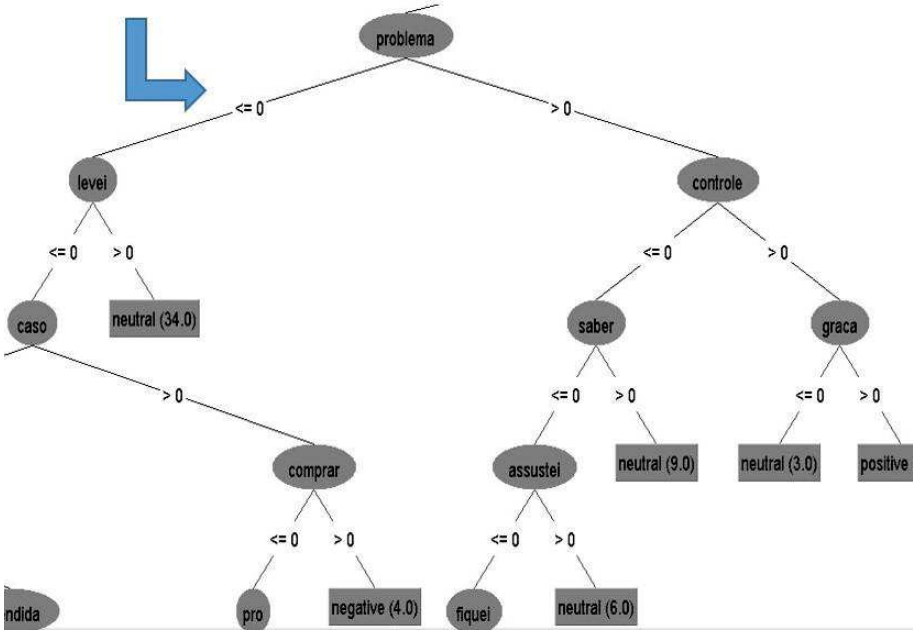
```
Classifier Model
Naive Bayes Classifier

                          Class
Attribute               positive  neutral negative
                          (0.22)   (0.53)   (0.25)
=============================================================
a
  mean                    0.0063    0.032   0.0169
  std. dev.               0.1667    0.176   0.1667
  weight sum                 158      375      177
  precision                    1        1        1

abraco
  mean                    0.0127   0.0107        0
  std. dev.               0.1667   0.1667   0.1667
  weight sum                 158      375      177
  precision                    1        1        1

abs
  mean                    0.0063   0.0107   0.0056
  std. dev.               0.1667   0.1667   0.1667
  weight sum                 158      375      177
  precision                    1        1        1

acabamento
  mean                    0.0127        0        0
  std. dev.               0.1667   0.1667   0.1667
  weight sum                 158      375      177
  precision                    1        1        1

acabou
  mean                    0.0063   0.0027   0.0226
  std. dev.               0.1667   0.1667   0.1667
  weight sum                 158      375      177
  precision                    1        1        1
```

FONTE: Weka (2018).

Para o algoritmo J48, que é um modelo de classificação baseado em árvore, torna-se mais fácil a interpretação que a dos outros dois algoritmos utilizados neste estudo. Isto se deve a que podem ser melhor visualizadas suas regras de classificação, ainda que, algumas vezes, a visualização da árvore completa seja prejudicada pela quantidade de atributos previsores, após a extração da regra de classificação da árvore gerada, a interpretação desta é mais intuitiva. Segundo Castro e Ferrari (2016), na classificação dos modelos baseados em árvore o nó (ou nodo) raiz e os nós (ou nodos) intermediários das árvores representam testes sobre um atributo, sendo que os ramos (ou galhos) significam os resultados de tais testes e os nós (ou nodos) folhas os rótulos das classes. Na Figura 30 é possível visualizar os resultados obtidos na formação da árvore e das classes geradas para cada atributo, ocorridos durante o treinamento.

FIGURA 30 – MODELO DE CLASSIFICAÇÃO J48

```
cambio <= 0
|   problema <= 0
|   |   levei <= 0
|   |   |   caso <= 0
|   |   |   |   embreagem <= 0
|   |   |   |   |   defeito <= 0
|   |   |   |   |   |   protetor <= 0
|   |   |   |   |   |   |   barulho <= 0
|   |   |   |   |   |   |   |   falta <= 0
|   |   |   |   |   |   |   |   |   otimo <= 0
|   |   |   |   |   |   |   |   |   |   resposta <= 0
|   |   |   |   |   |   |   |   |   |   |   comprei <= 0
|   |   |   |   |   |   |   |   |   |   |   |   pos <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   grande <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   solucao <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   minivan <= 0
```



FONTE: Weka (2018).

- estatística de Kappa (*Kappa statistic*): medida estatística que tem por finalidade verificar o grau de confiabilidade intermediária, o seu valor indica quanto os dados coletados são representações corretas das variáveis, segundo Simon (2005), uma possível interpretação dos valores seria:
  - deficiente: menor que 0,20;
  - justo: de 0,20 a 0,40;
  - moderado: de 0,40 a 0,60;
  - bom: de 0,60 a 0,80
  - muito bom: de 0,80 a 1,00

- taxa de acerto (*classified instances*): quantidade e percentual de instâncias corretamente e incorretamente classificadas pelo algoritmo.
- matriz de confusão (*confusion matrix*): matriz que demonstra o número de classificações reais comparadas com as classificações preditas de cada classe. Assim pode ser verificado quantas foram classificadas de forma correta e quantas de forma incorreta para cada classe. O número de instâncias classificadas de forma correta pode ser verificado somando-se os valores na diagonal principal da matriz conforme pode ser visualizado na Figura 31. A soma das demais instâncias que não estão na diagonal principal é o total das que foram classificadas incorretamente.

FIGURA 31 – EXEMPLO DE RESULTADO DE ANÁLISE EFETUADO NO WEKA

```
Time taken to build model: 0.72 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         578               68.5647 %
Incorrectly Classified Instances       265               31.4353 %
Kappa statistic                          0.4764
Mean absolute error                      0.2255
Root mean squared error                  0.4006
Relative absolute error                 57.9991 %
Root relative squared error             90.8704 %
Total Number of Instances              843

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,699    0,100    0,626      0,699    0,661      0,575    0,900     0,734     positive
               0,704    0,275    0,771      0,704    0,736      0,425    0,795     0,819     neutral
               0,632    0,151    0,567      0,632    0,598      0,464    0,812     0,666     negative
Weighted Avg.  0,686    0,211    0,694      0,686    0,688      0,463    0,819     0,766

=== Confusion Matrix ===

   a   b   c   <-- classified as
 114  34  15 |   a = positive
  66 337  82 |   b = neutral
   8  66 127 |   c = negative
```

FONTE: Weka (2018).

> Segundo a Google Developers (2018), um verdadeiro positivo (*true positive*) é um resultado em que o modelo prevê de forma correta a classe positiva. Da mesma forma, um verdadeiro negativo (*true negative*) é um resultado em que o modelo prevê corretamente a classe negativa. Já os falsos positivos (*false positive*) e os falsos negativos (*false negative*) são resultados que prevêem incorretamente as classes positiva e negativa respectivamente.

Após a apresentação dos encaminhamentos metodológicos aplicados neste trabalho, a próxima seção aborda os resultados obtidos.

4 RESULTADOS

Nesta seção são apresentados os resultados alcançados com a aplicação dos algoritmos selecionados nas bases de dados pré-processadas para este trabalho.

4.1 ANÁLISE DAS BASES DE DADOS

Foram executadas no Semantria as 4 (quatro) bases de dados geradas após o pré-processamento no código Python (conforme detalhamento na seção 3.2.2.1). O Semantria efetuou diversas análises conforme alguns modelos que podem ser visualizados no apêndice B. Dos resultados compatíveis com o estudo deste trabalho, foi selecionado o da planilha "*EntityThemeDetail*" pois categoriza a frase analisada, separando a "entidade" o "tipo de entidade" e o segmento da frase que caracteriza o sentimento, para depois classificá-lo como negativo, positivo e neutro. As colunas "*Highlighted Text" e "Entity Theme Sentiment"* foram utilizadas para gerar as bases com extensão "arff", que foram submetidas para análise no *software* Weka na etapa de processamento. Os detalhes da planilha "EntityThemeDetail" e das colunas que foram utilizadas para compor a base de dados para o processamento de mineração podem ser visualizadas na Figura 32.

FIGURA 32 – TELA DE ANÁLISE DO SEMANTRIA "ENTITYTHEMEDETAIL"

Document ID	Highlighted Text	Entity	Entity Type	Entity Theme	Entity Theme Sentiment	Entity Theme Sentiment +/-	Entity Theme Sentiment Evidenc
1	nao comprem ford ka problema alto consumo combustivel	ford	veículo	ka problema	-0,800000012	negative	4
1	...d ka problema alto consumo combustivel concessionaria ford	ford	veículo	concessionaria ford	-0,572878301	negative	4
1	...onsumo combustivel concessionaria ford resolvem problema	ford	veículo	resolvem problema	-0,800000012	negative	4
1	...onaria ford resolvem problema briga pessimo atendimento	ford	veículo	pessimo atendimento	-0,572878301	negative	4
2	...direitos um carro praticamente novo nao sei tipo carro vendendo	ford	veículo	sei tipo	0,699999988	positive	4
2	... custo mil arrumar brincadeira ne ja atras direitos um carro	ford	veículo	atras direitos	0,070183992	neutral	4
2	...esmente levei concessionaria apresentou uns tranco troca marcha	ford	veículo	uns tranco	-0,25	neutral	4
2	ford melhor marca tempos	ford	veículo	ford melhor	1	positive	4
2	ford melhor marca tempos	ford	veículo	marca tempos	1	positive	4
3	ford ka apresenta defeito fabrica barulho re desde primeiro saiu	ford	veículo	fabrica barulho	-0,295238107	neutral	4
3	...to fabrica nunca solucionam definitivo carro devolver preciso	ford	veículo	carro devolver	-0,295238107	neutral	4
3	...r horarios estabelecidos pasmem falei possibilidade carro	ford	veículo	possibilidade carro	-0,295238107	neutral	4
3	...bilidade carro reserva o ford disseram servico garantia contratual	ford	veículo	servico garantia	-0,295238107	neutral	4
3	...ma solicitei carro reserva nunca dado problema garantia	ford	veículo	problema garantia	0,800000012	positive	4
3	...rantia contratual nao fornecido antes sei prejuizo diariamente	ford	veículo	sei prejuizo	-0,699999988	negative	4
3	... carro apresenta defeito fabrica achar real causa insatisfacao	ford	veículo	real causa	-0,295238107	neutral	4
3	... fabrica achar real causa insatisfacao revolta define numero	ford	veículo	revolta define	-0,600000024	negative	4

FONTE: Semantria (2018).

Outros resultados gerados pela ferramenta Semantria foram as nuvens de palavras que apresentam a análise de frequência dos termos para cada uma das bases, conforme o tratamento a que foram submetidas. Na nuvem de palavras

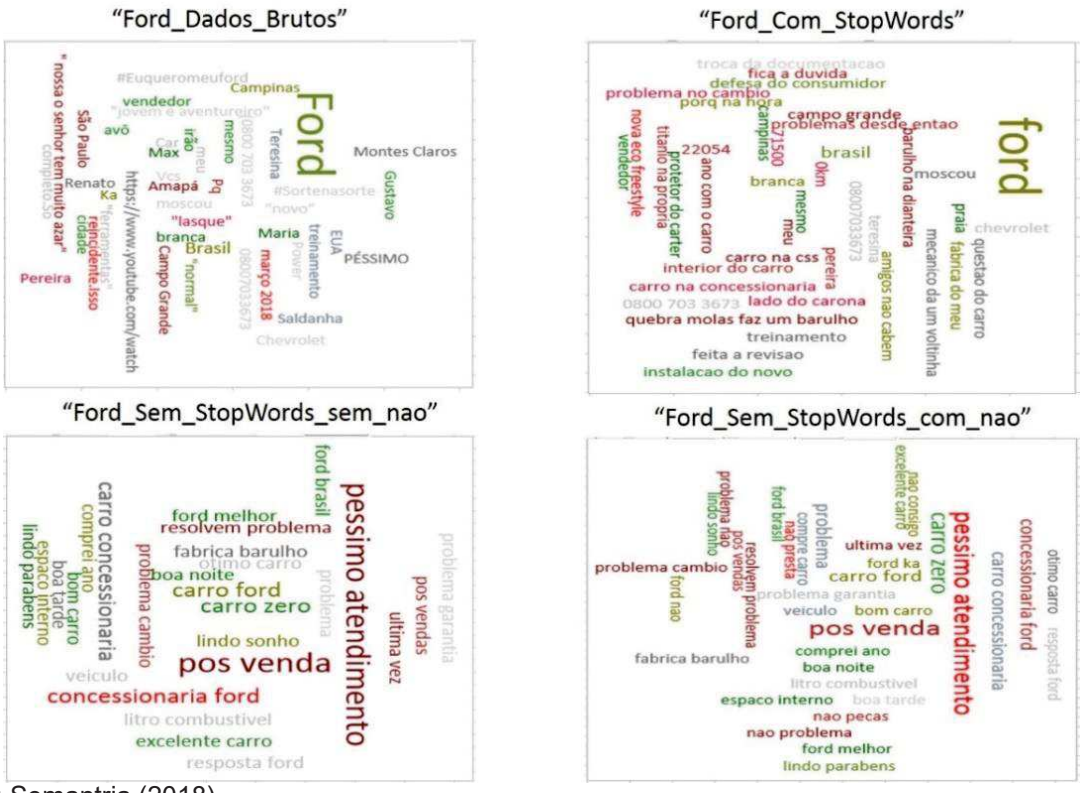
formada para a base “Ford_Dados_Brutos”, sem qualquer limpeza com o pré-processamento, podem ser vistas palavras acentuadas, abreviações como “vcs”, caracteres especiais tais como “#”, letras maiúsculas, e um maior destaque pela frequência à palavra “Ford”.

Na nuvem formada para a base “Ford_Com_StopWords, que foi tratada com códigos Python, pode ser percebida a ausência de palavras acentuadas, abreviações, caracteres especiais e letras maiúsculas, porém ainda se mantém nas frases os *stopwords*, há uma maior densidade de palavras que na nuvem da base “Ford_Dados_Brutos” e continua sendo destaque de frequência a palavra “ford”.

Nas nuvens de palavras formadas para as bases “Ford_Sem_StopWords_sem_nao” e “Ford_Sem_StopWords_com_nao”, pode ser visualizado que além dos tratamentos para a retirada de palavras acentuadas, abreviações, caracteres especiais e letras maiúsculas, também foram retirados os *stopwords* e as nuvens são semelhantes a exceção que na primeira manteve-se a palavra “nao” e na segunda esta palavra foi retirada.

Na Figura 33 é possível visualizar as nuvens geradas de cada uma das bases, refletindo o resultado no tratamento que foi dado às mesmas.

FIGURA 33 – NUVENS DE PALAVRAS GERADAS NO SEMANTRIA



FONTE: Semantria (2018).

Da base original extraída do Facebook foram geradas 4 bases de dados com diferentes tratamentos de pré-processamento com os códigos Python constantes nos anexos A, B e C, e explicado na seção 3.2.2.1 deste trabalho. Na tabela 9 podem ser observados os nomes das bases de dados, as descrições, o número de instâncias (linhas) e atributos (colunas) de cada uma das bases. O fato das bases possuírem quantidade de instâncias diferentes é devido à análise que o software Semantria executou em cada base com a influência do pré-processamento efetuado a cada uma delas. Todas as bases contêm dois atributos, sendo o primeiro do tipo string (texto a analisar) e o segundo do tipo *phrasesentiment* (positivo, neutro ou negativo).

TABELA 9 – CARACTERÍSTICAS DAS BASES DE DADOS ANALISADAS

Base de Dados	Descrição da Base de Dados	Nº de Instâncias	Nº de Atributos
Ford_Dados_Brutos	Base de dados sem nenhum tratamento dos dados.	430	2
Ford_Com_StopWords	Base de dados com padronização de minúsculas, abreviações, emoticons, retirada de caracteres especiais e de acentuação. Mantendo-se os StopWords.	1241	2
Ford_Sem_StopWords_sem_nao	Base de dados com padronização de minúsculas, abreviações, emoticons, retirada de caracteres especiais e de acentuação. Retirando-se os StopWords, inclusive o "nao".	806	2
Ford_Sem_StopWords_com_nao	Base de dados com padronização de minúsculas, abreviações, emoticons, retirada de caracteres especiais e de acentuação. Retirando-se os StopWords, exceto o "nao".	843	2

FONTE: O autor (2018).

4.2 RESULTADOS DAS BASES ANALISADAS

Os experimentos em todas as bases analisadas foram realizados com validação cruzada de 10 participações, com a utilização dos filtros “*FilteredClassifier*” e “*StringToWordVector*” e com o uso de “*tokenizer*”, (seção 3.2.2.2).

4.2.1 RESULTADOS PARA A BASE “Ford_Dados_Brutos”

A base de dados “Ford_Dados_Brutos” contém 430 instâncias e 2 atributos. Esta base não foi submetida a nenhum pré-processamento com os códigos Python que retiram caracteres especiais, acentuação, *stopwords* etc.

Os resultados do modelo de classificação desta base para os três algoritmos podem ser vistos na Figura 34. Para o algoritmo *Naive Bayes* as probabilidades das classes (23% para positivo, 45% para neutro e 32% para negativo), com as respectivas probabilidades relativas de cada atributo que está na figura. Para o algoritmo SMO os resultados normalizados que constam na figura de alguns dos atributos da base (por exemplo: -0.2615 para “#ford” e 0,0711 para “0”), o número de avaliações foi de 53.199. Como saída do algoritmo J48, a árvore gerada apresentou 71 folhas e tamanho 141.

FIGURA 34 – MODELO DE CLASSIFICAÇÃO BASE “FORD_DADOS_BRUTOS”

```
Classifier Model
Naive Bayes Classifier

                        Class
Attribute          positive  neutral negative
                     (0.23)   (0.45)   (0.32)
=================================================
#ford
  mean                0.0103        0        0
  std. dev.           0.1667   0.1667   0.1667
  weight sum              97      196      137
  precision                1        1        1

0
  mean                0.0412    0.051        0
  std. dev.           0.1988     0.22   0.1667
  weight sum              97      196      137
  precision                1        1        1

0km
  mean                0.0103        0        0
  std. dev.           0.1667   0.1667   0.1667
  weight sum              97      196      137
  precision                1        1        1
```

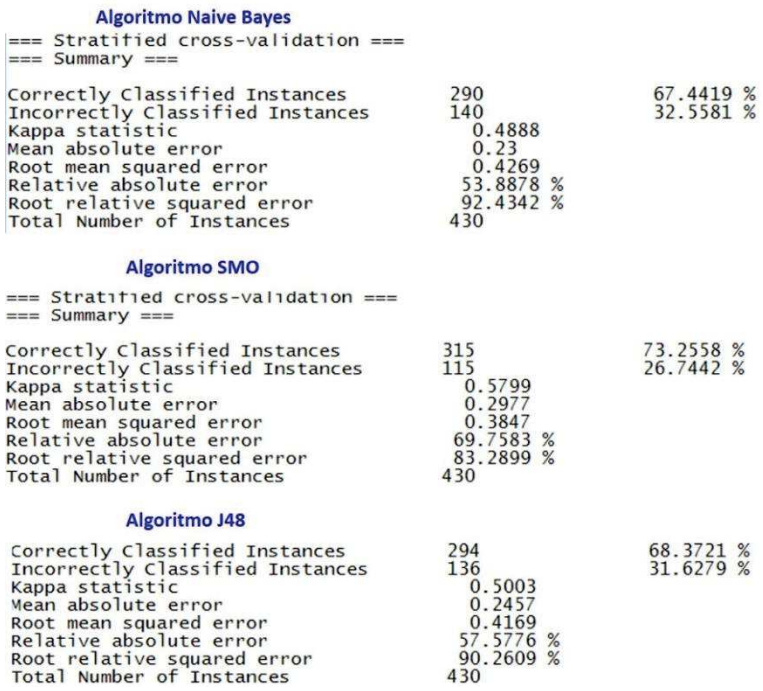
```
Classifier Model
SMO
Kernel used:
  Linear Kernel: K(x,y) = <x,y>

Classifier for classes: positive, neutral
BinarySMO

Machine linear: showing attribute weights,
not support vectors.
        -0.2615 * (normalized) #ford
 +       0.0711 * (normalized) 0
 +      -0.0388 * (normalized) 0km
 +       0.0405 * (normalized) 1
 +      -0.1749 * (normalized) 2001
 +      -0.0157 * (normalized) 2004
 …
Number of kernel evaluations: 53199 (95.257% cached)
```

```
Classifier Model
J48 pruned tree
------------------

minivan <= 0
|   q <= 0
|   |   jÃ¡ <= 0
|   |   |   Brasil <= 0
|   |   |   |   marca <= 0
|   |   |   |   |   jovem <= 0
|   |   |   |   |   |   levei <= 0
|   |   |   |   |   |   |   der <= 0
|   |   |   |   |   |   |   |   0 <= 0
|   |   |   |   |   |   |   |   |   defeito <= 0
|   |   |   |   |   |   |   |   |   |   som <= 0
|   |   |   |   |   |   |   |   |   |   |   pois <=
|   |   |   |   |   |   |   |   |   |   |   |   fui
|   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |   |   |   |   |

Number of Leaves  :     71

Size of the tree :     141
```

FONTE: Weka (2018).

Os resultados de taxa de acerto obtidos nesta base para os três algoritmos podem ser visualizados na Figura 35. Pode-se observar que o algoritmo SMO apresentou uma taxa de acertos superior aos outros dois algoritmos, sendo 315 instâncias corretamente classificadas e 115 instâncias classificadas incorretamente, o que corresponde respectivamente a um percentual de 73,26% e 26,74%.

FIGURA 35 – TAXA DE ACERTO BASE “FORD_DADOS_BRUTOS”

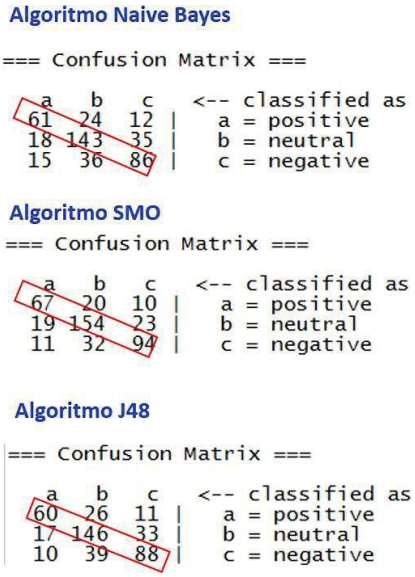


FONTE: Weka (2018).

Na Figura 35 também é possível verificar os resultados da estatística Kappa para os 3 (três) algoritmos. Os valores constantes apresentam grau de confiabilidade intermediária “moderado” (valores entre 0,40 e 0,60), conforme explicado na seção 3.2.2.3.

Os resultados da matriz de confusão desta base para os três algoritmos podem ser visualizados na Figura 36. Os acertos para cada matriz podem ser observados na diagonal principal (área demarcada por retângulo) e os erros no restante. Para o algoritmo *Naive Bayes* foram classificadas corretamente 61 instâncias positivas, 143 neutras e 86 negativas. Para o algoritmo SMO foram classificadas corretamente 67 instâncias positivas, 154 neutras e 94 negativas. Para o algoritmo J48 foram classificadas 60 instâncias positivas, 146 neutras e 88 negativas. O algoritmo SMO teve maior número de acertos nas três classificações. É possível perceber também que 196 instâncias (aproximadamente 46% das classificações) são neutras, independentemente da taxa de acertos dos três algoritmos.

FIGURA 36 - MATRIZ DE CONFUSÃO BASE “FORD_DADOS_BRUTOS”



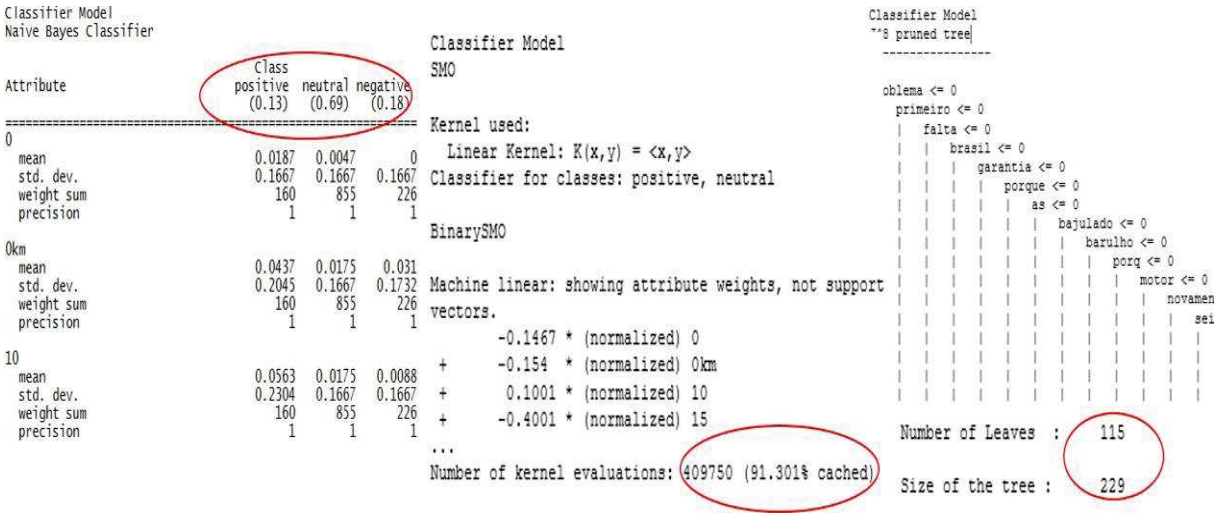
FONTE: Weka (2018).

4.2.2 RESULTADOS PARA A BASE “Ford_Com_StopWords”

A base de dados “Ford_Com_StopWords” contém 1241 instâncias e 2 atributos. Esta base foi submetida ao código Python do anexo C, que retira caracteres especiais, acentuação, etc., porém mantém os *stopwords* na base.

Os resultados do modelo de classificação desta base para os três algoritmos podem ser vistos na Figura 37. Para o algoritmo *Naive Bayes* as probabilidades das classes (13% para positivo, 69% para neutro e 18% para negativo), com as respectivas probabilidades relativas de cada atributo que está na figura. Para o algoritmo SMO os resultados normalizados que constam na figura de alguns dos atributos da base (por exemplo: -0.1467 para “0” e -0,0154 para “0km”), o número de avaliações foi de 409.750. Como saída do algoritmo J48, a árvore gerada apresentou 115 folhas e tamanho 229.

FIGURA 37 – MODELO DE CLASSIFICAÇÃO BASE “FORD_COM_STOPWORDS”



```
Classifier Model
Naive Bayes Classifier

                          Class
Attribute                positive  neutral negative
                           (0.13)   (0.69)   (0.18)
=============================================================
0
  mean                     0.0187   0.0047        0
  std. dev.                0.1667   0.1667   0.1667
  weight sum                  160      855      226
  precision                     1        1        1

0km
  mean                     0.0437   0.0175    0.031
  std. dev.                0.2045   0.1667   0.1732
  weight sum                  160      855      226
  precision                     1        1        1

10
  mean                     0.0563   0.0175   0.0088
  std. dev.                0.2304   0.1667   0.1667
  weight sum                  160      855      226
  precision                     1        1        1
```

```
Classifier Model
SMO

Kernel used:
  Linear Kernel: K(x,y) = <x,y>
Classifier for classes: positive, neutral

BinarySMO

Machine linear: showing attribute weights, not support
vectors.
       -0.1467 * (normalized) 0
 +     -0.154  * (normalized) 0km
 +      0.1001 * (normalized) 10
 +     -0.4001 * (normalized) 15
...
Number of kernel evaluations: 409750 (91.301% cached)
```

```
Classifier Model
J48 pruned tree
------------------

oblema <= 0
  primeiro <= 0
  |   falta <= 0
  |   |   brasil <= 0
  |   |   |   garantia <= 0
  |   |   |   |   porque <= 0
  |   |   |   |   |   as <= 0
  |   |   |   |   |   |   bajulado <= 0
  |   |   |   |   |   |   |   barulho <= 0
  |   |   |   |   |   |   |   |   porq <= 0
  |   |   |   |   |   |   |   |   |   motor <= 0
  |   |   |   |   |   |   |   |   |   |   novamen
  |   |   |   |   |   |   |   |   |   |   |   sei
  |   |   |   |   |   |   |   |   |   |   |   |
  |   |   |   |   |   |   |   |   |   |   |   |
  |   |   |   |   |   |   |   |   |   |   |   |

Number of Leaves  :     115

Size of the tree :     229
```

FONTE: Weka (2018).

Os resultados de taxa de acerto obtidos nesta base para os três algoritmos podem ser visualizados na Figura 38. Pode-se observar que o algoritmo SMO apresentou uma taxa de acerto superior aos outros dois algoritmos, sendo 1159 instâncias corretamente classificadas e 82 instâncias classificadas incorretamente, o que corresponde respectivamente a um percentual de 93,39% e 6,61%.

FIGURA 38 - TAXA DE ACERTO BASE “FORD_COM_STOPWORDS”

Algoritmo Naive Bayes

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         946               76.2288 %
Incorrectly Classified Instances       295               23.7712 %
Kappa statistic                          0.5415
Mean absolute error                      0.1732
Root mean squared error                  0.3684
Relative absolute error                 54.588  %
Root relative squared error             92.5217 %
Total Number of Instances             1241
```

Algoritmo SMO

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1159               93.3924 %
Incorrectly Classified Instances        82                6.6076 %
Kappa statistic                          0.8602
Mean absolute error                      0.2387
Root mean squared error                  0.3007
Relative absolute error                 75.2097 %
Root relative squared error             75.529  %
Total Number of Instances             1241
```

Algoritmo J48

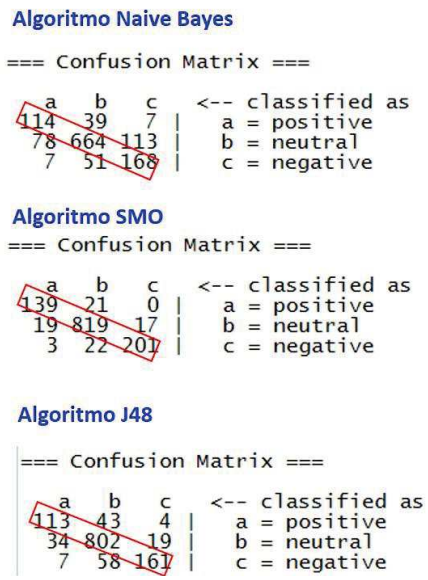
```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1076               86.7043 %
Incorrectly Classified Instances       165               13.2957 %
Kappa statistic                          0.7082
Mean absolute error                      0.1019
Root mean squared error                  0.2802
Relative absolute error                 32.1073 %
Root relative squared error             70.3674 %
Total Number of Instances             1241
```

FONTE: Weka (2018).

Na Figura 38 também é possível verificar os resultados da estatística Kappa para os 3 algoritmos. Os valores constantes apresentam grau de confiabilidade intermediária “muito bom” para o algoritmo SMO, “bom” para o algoritmo J48 e “moderado” para o algoritmo *Naive Bayes*.

Os resultados da matriz de confusão desta base para os três algoritmos podem ser visualizados na Figura 39. Os acertos para cada matriz podem ser observados na diagonal principal (área demarcada por retângulo) e os erros no restante. Para o algoritmo *Naive Bayes* foram classificadas corretamente 114 instâncias positivas, 664 neutras e 168 negativas. Para o algoritmo SMO foram classificadas corretamente 139 instâncias positivas, 819 neutras e 201 negativas. Para o algoritmo J48 foram classificadas 113 instâncias positivas, 802 neutras e 161 negativas. O algoritmo SMO teve maior número de acertos nas três classificações. Percebe-se também que 855 instâncias (aproximadamente 69% das classificações) são neutras, independentemente da taxa de acertos dos três algoritmos.

FIGURA 39 – MATRIZ DE CONFUSÃO BASE “FORD_COM_STOPWORDS”



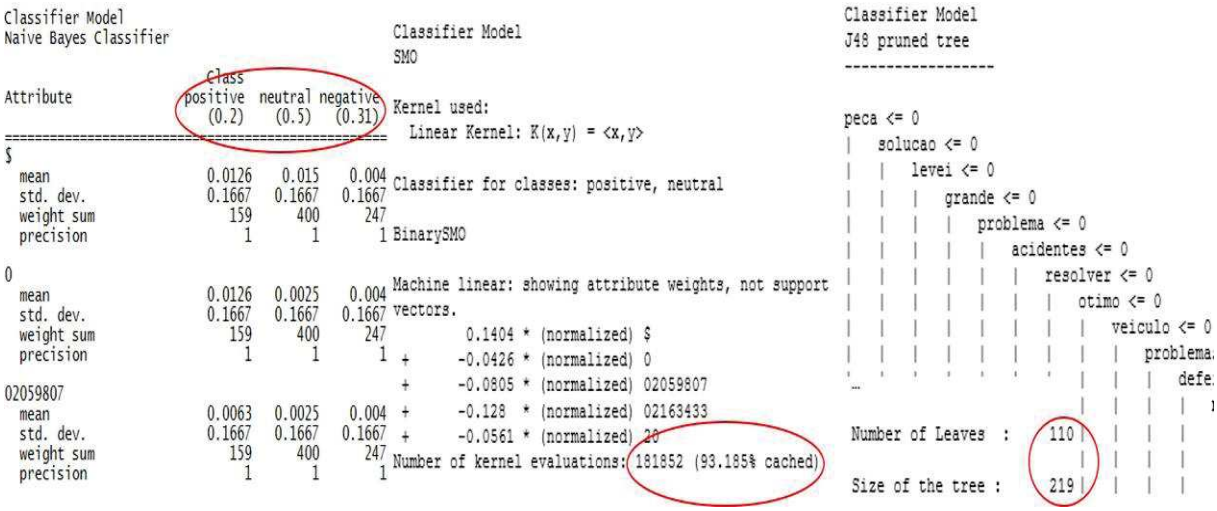
FONTE: Weka (2018).

4.2.3 RESULTADOS PARA A BASE “Ford_Sem_StopWords_sem_nao”

A base de dados “Ford_Sem_StopWords_sem_nao” contém 806 instâncias e 2 atributos. Esta base foi submetida ao código Python do anexo B, que retira caracteres especiais, acentuação, etc., e os *stopwords* da base inclusive o “nao”.

Os resultados do modelo de classificação desta base para os três algoritmos podem ser vistos na Figura 40. Para o algoritmo *Naive Bayes* as probabilidades das classes (20% para positivo, 50% para neutro e 30% para negativo), com as respectivas probabilidades relativas de cada atributo que está na figura. Para o algoritmo SMO os resultados normalizados que constam na figura de alguns dos atributos da base (por exemplo: -0.1404 para “$” e -0,0426 para “0”), o número de avaliações foi de 181.852. Como saída do algoritmo J48, a árvore gerada apresentou 110 folhas e tamanho 219.

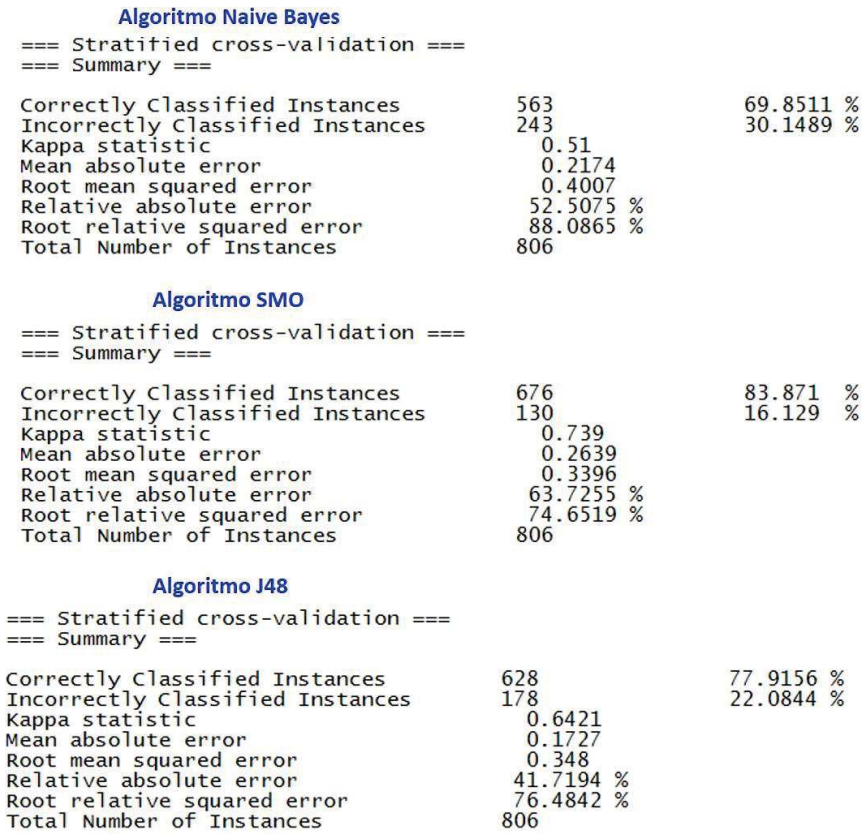
FIGURA 40 – MODELO DE CLASSIFICAÇÃO BASE “FORD_SEM_STOPWORDS_SEM_NAO”



FONTE: Weka (2018).

Os resultados de taxa de acerto obtidos nesta base para os três algoritmos podem ser visualizados na Figura 41. Pode-se observar que o algoritmo SMO apresentou uma taxa de acertos superior aos outros dois algoritmos, sendo 676 instâncias corretamente classificadas e 130 instâncias classificadas incorretamente, o que corresponde respectivamente a um percentual de 83,87% e 16,13%.

FIGURA 41 - TAXA DE ACERTO BASE “FORD_SEM_STOPWORDS_SEM_NAO”

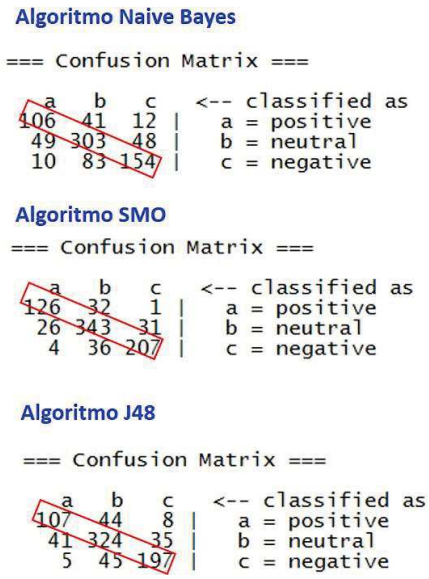


FONTE: Weka (2018).

Na Figura 41 também é possível verificar os resultados da estatística Kappa para os 3 algoritmos. Os valores constantes apresentam grau de confiabilidade intermediária “bom” para o algoritmo SMO, “bom” para o algoritmo J48 e “moderado” para o algoritmo *Naive Bayes*.

Os resultados da matriz de confusão desta base para os três algoritmos podem ser visualizados na Figura 42. Os acertos para cada matriz podem ser observados na diagonal principal (área demarcada por retângulo) e os erros no restante. Para o algoritmo *Naive Bayes* foram classificadas corretamente 106 instâncias positivas, 303 neutras e 154 negativas. Para o algoritmo SMO foram classificadas corretamente 126 instâncias positivas, 343 neutras e 207 negativas. Para o algoritmo J48 foram classificadas 107 instâncias positivas, 324 neutras e 197 negativas. O algoritmo SMO teve maior número de acertos nas três classificações. Também é perceptível que 400 instâncias (aproximadamente 50% das classificações) são neutras, independentemente da taxa de acertos dos três algoritmos.

FIGURA 42 – MATRIZ DE CONFUSÃO BASE “FORD_SEM_STOPWORDS_SEM_NAO”



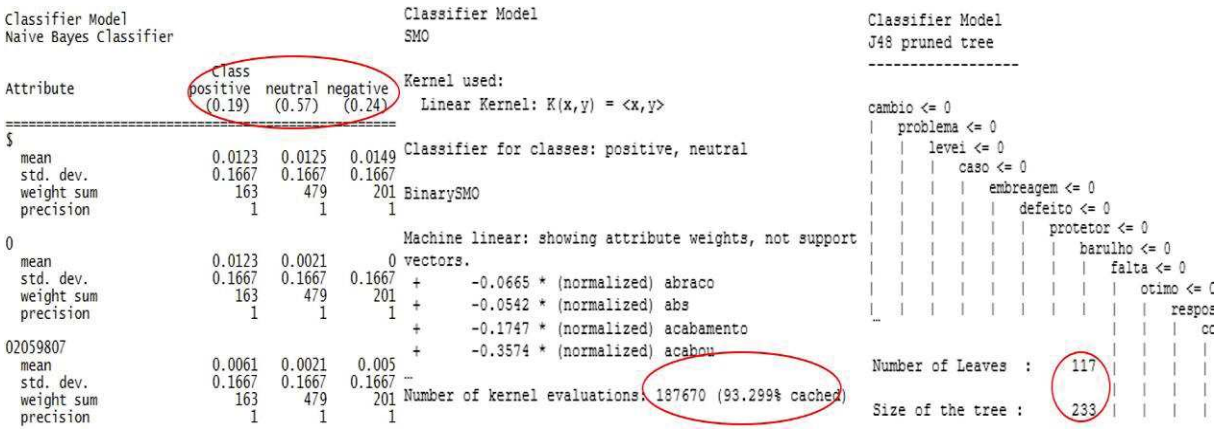
FONTE: Weka (2018).

4.2.4 RESULTADOS PARA A BASE “Ford_Sem_StopWords_com_nao”

A base de dados “Ford_Sem_StopWords_com_nao” contém 843 instâncias e 2 atributos. Esta base foi submetida ao código Python do anexo A, que retira caracteres especiais, acentuação etc., e os *stopwords* da base exceto o “nao”.

Resultados do modelo de classificação desta base para os três algoritmos podem ser vistos na Figura 43. Para o algoritmo *Naive Bayes* as probabilidades das classes (19% para positivo, 57% para neutro e 24% para negativo), com as respectivas probabilidades relativas de cada atributo que está na figura. Para o algoritmo SMO os resultados normalizados que constam na figura de alguns dos atributos da base (por exemplo: -0.0665 para “abraco” e -0,0542 para “abs”), o número de avaliações foi de 187.670. Como saída do algoritmo J48, a árvore gerada apresentou 117 folhas e tamanho 233.

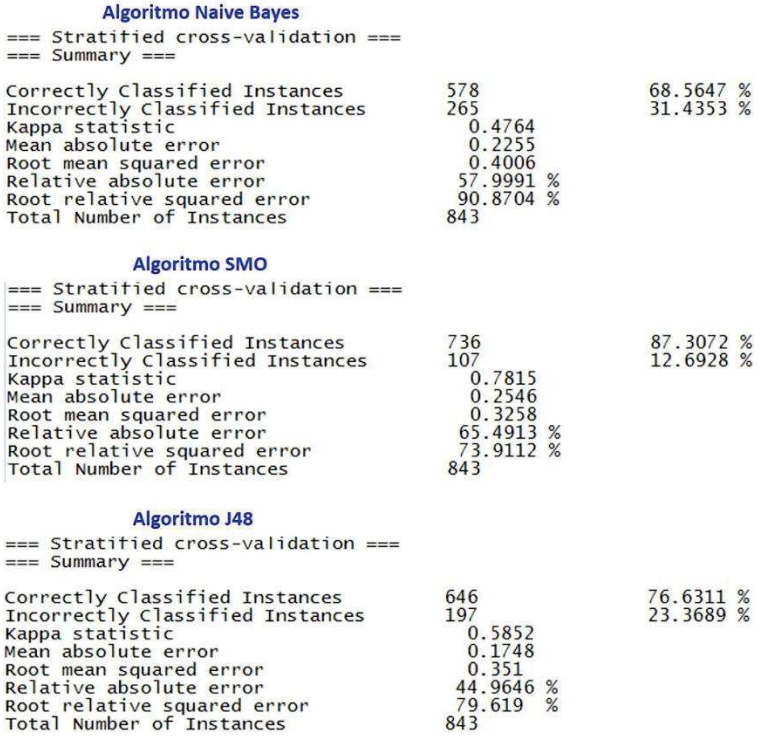
FIGURA 43 - MODELO DE CLASSIFICAÇÃO BASE “FORD_SEM_STOPWORDS_COM_NAO”



FONTE: Weka (2018).

Os resultados de taxa de acerto obtidos nesta base para os três algoritmos podem ser visualizados na Figura 44. Pode-se observar que o algoritmo SMO apresentou uma taxa de acertos superior aos outros dois algoritmos, sendo 736 instâncias corretamente classificadas e 107 instâncias classificadas incorretamente, o que corresponde respectivamente a um percentual de 87,31% e 12,69%.

FIGURA 44 - TAXA DE ACERTO BASE “FORD_SEM_STOPWORDS_COM_NAO”

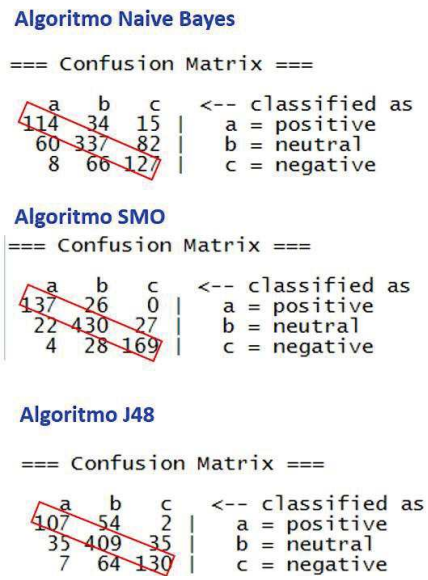


FONTE: Weka (2018).

Na Figura 44 também é possível verificar os resultados da estatística Kappa para os 3 algoritmos. Os valores constantes apresentam grau de confiabilidade intermediária “bom” para o algoritmo SMO, “moderado” para o algoritmo J48 e “moderado” para o algoritmo *Naive Bayes*, conforme explicado na seção 3.2.2.3.

Os resultados da matriz de confusão desta base para os três algoritmos podem ser visualizados na Figura 45. Os acertos para cada matriz podem ser observados na diagonal principal (área demarcada por retângulo) e os erros no restante. Para o algoritmo *Naive Bayes* foram classificadas corretamente 114 instâncias positivas, 337 neutras e 127 negativas. Para o algoritmo SMO foram classificadas corretamente 137 instâncias positivas, 430 neutras e 169 negativas. Para o algoritmo J48 foram classificadas 107 instâncias positivas, 409 neutras e 130 negativas. O algoritmo SMO teve maior número de acertos nas três classificações. É possível perceber também que 479 instâncias (aproximadamente 57% das classificações) são neutras, independentemente da taxa de acertos dos três algoritmos.

FIGURA 45 - MATRIZ DE CONFUSÃO BASE “FORD_SEM_STOPWORDS_COM_NAO”



FONTE: Weka (2018).

4.3 COMPARAÇÃO DOS RESULTADOS ENTRE AS BASES

Para os resultados obtidos nos modelos de classificação com o algoritmo *Naive Bayes*, pode-se observar que as probabilidades das classes (positivo, neutro e negativo) são semelhantes para as bases “Ford_Sem_StopWords_sem_nao” (20%,

50% e 31%) e “Ford_Sem_StopWords_com_nao”, (19%, 57% e 24%), para a base “Ford_Dados_Brutos” (23%, 45% e 32%) e para a base “Ford_Com_StopWords” (13%, 69% e 18%) com maior probabilidade da classe neutra.

Quanto ao número de avaliações geradas no algoritmo SMO para as 4 bases há apenas 53.199 para a base “Ford_Dados_Brutos” se comparado com 409.750 geradas na base “Ford_Com_StopWords” e 181.852 e 187.670 respectivamente nas bases “Ford_Sem_StopWords_sem_nao” e “Ford_Sem_StopWords_com_nao”.

Quanto ao número de folhas e tamanho da árvore geradas nas análises do algoritmo J48 nas bases, a que apresentou menor quantidade foi a base “Ford_Dados_Brutos” com 71 folhas e tamanho 141. As demais bases tiveram resultados semelhantes variando o número de folhas de 110 a 117 e tamanho de 219 a 233.

Com relação aos resultados obtidos entre as bases, é possível verificar que o algoritmo SMO obteve melhores resultados de taxas de acertos nas 4 bases submetidas para análise. Para a base bruta o percentual de taxa de acerto de instâncias classificadas corretamente foi de 73,3% utilizando o algoritmo SMO, 68,4% com o algoritmo J48 e 67,4% com o algoritmo *Naive Bayes*.

Para a base em que se mantiveram os *stopwords* o percentual de taxa de acerto para o algoritmo SMO foi de 93,4%, 86,7% com o algoritmo J48 e 76,2% com o algoritmo *Naive Bayes*. Para a base à qual foram retirados os *stopwords* inclusive o “nao” os percentuais de taxa de acerto foram de 83,9% utilizando o algoritmo SMO, 77,9% com o algoritmo J48 e 69,9% com o algoritmo *Naive Bayes*.

Para a base à qual foram retirados os *stopwords* exceto o “nao” os percentuais de taxas de acerto foram 87,3% utilizando o algoritmo SMO, 76,6% com o algoritmo J48 e 68,6% utilizando o algoritmo *Naive Bayes*.

Como pode ser visto na Tabela 10, o algoritmo SMO teve maior percentual de classificação correta em todas as situações, seguido do algoritmo J48. O algoritmo *Naive Bayes* foi o que apresentou menores percentuais de taxa de acerto em todos os experimentos realizados.

TABELA 10 – PERCENTUAL DE CLASSIFICAÇÃO CORRETA ENTRE ALGORITMOS

Base de Dados	Percentual de Classificações Corretas (%)		
	Algoritmo Naive Bayes	Algoritmo SMO	Algoritmo J48
Ford_Dados_Brutos	67,4%	73,3%	68,4%
Ford_Com_StopWords	76,2%	93,4%	86,7%
Ford_Sem_StopWords_sem_nao	69,9%	83,9%	77,9%
Ford_Sem_StopWords_com_nao	68,6%	87,3%	76,6%

FONTE: O autor (2018).

A respeito dos resultados de grau de confiabilidade intermediária atribuído pela estatística Kappa, para a base de dados bruta o grau encontrado foi “moderado” para os 3 algoritmos. Para a base em que se manteve os *stopwords* o grau foi “muito bom” para o algoritmo SMO, “bom” para o algoritmo J48 e “moderado para o algoritmo *Naive Bayes*. Para a base que teve a retirada de *stopwords* inclusive o “nao” o grau encontrado foi “bom” para os algoritmos SMO e J48 e “moderado” para o algoritmo *Naive Bayes*. Para a base que teve a retirada de *stopwords* excetuando o “não”, o grau foi “bom” unicamente para o algoritmo SMO e “moderado” para os algoritmos J48 e *Naive Bayes*.

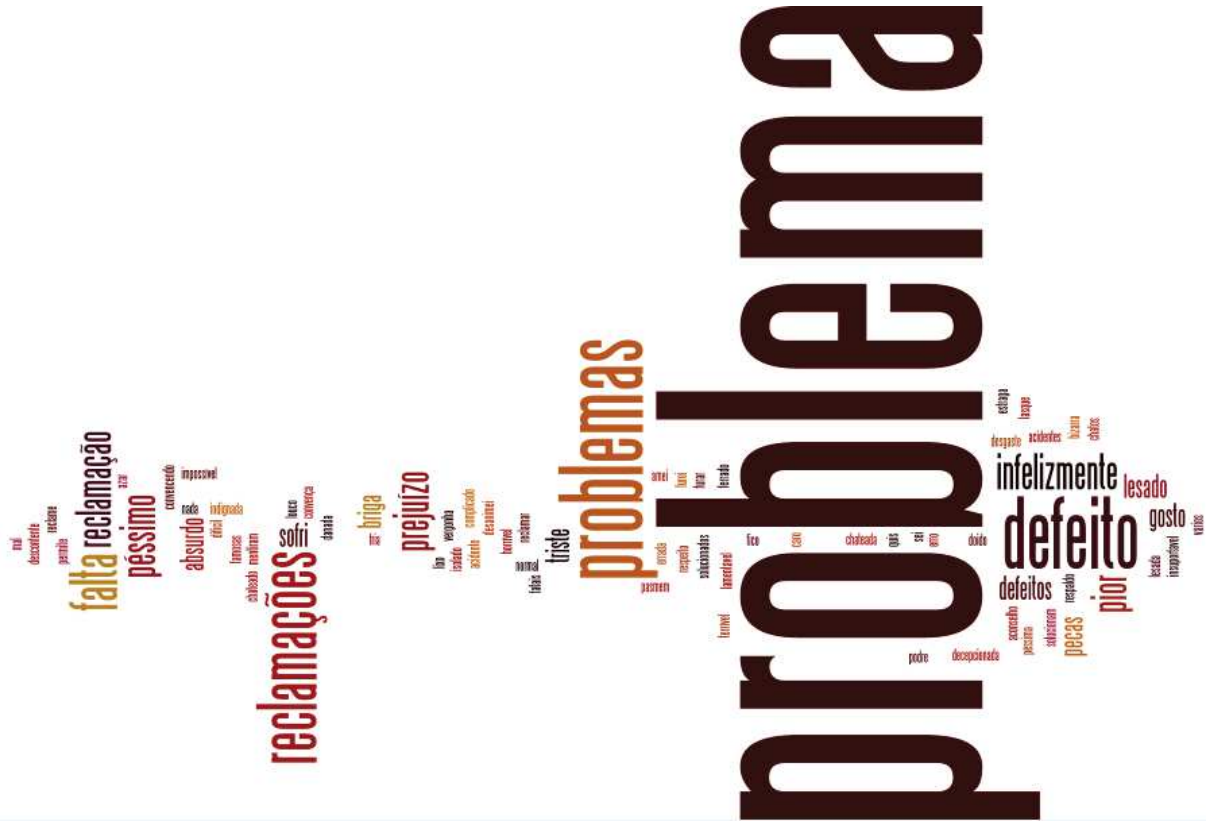
4.4 INDICADORES DE SATISFAÇÃO

Utilizando os termos negativos e positivos classificados na análise do Semantria, é possível verificar a frequência das palavras e a presença de alguns objetivos de desempenho quanto a indicadores de satisfação do consumidor (conforme explicado nas seções 2.1 e 2.2).

Tomando como exemplo a Base Ford_Dados_Brutos que gerou 430 instâncias, sendo 137 negativas, 196 neutras e 97 positivas, é possível visualizar na Figura 46 a nuvem gerada para os 137 termos negativos. As maiores densidades são encontradas nas palavras: “problema” com 27 ocorrências (19,7%), “problemas” com 7 ocorrências (5%), “defeito” com 6 ocorrências (4%), “reclamações” com 5 ocorrências (3%), “falta” com 4 ocorrências (3%), “péssimo” com 3 ocorrências (2%) e “prejuízo” com 3 ocorrências (2%). Destes 7 (sete) termos com maior frequência, 2 deles denotam relação direta com objetivos de desempenho: **“defeito”**, atrelado ao objetivo

qualidade, que se relaciona com a não conformidade de especificações de produto e **"prejuízo"**, que pode estar relacionado com **custo**, que também é um dos cinco objetivos de desempenho nos indicadores de satisfação.

FIGURA 46 – NUVEM DE TERMOS NEGATIVOS GERADOS NO SEMANTRIA



FONTE: Semantria (2018).

Considerando que, conforme explicado na seção 2.1, devem ser evitados os insatisfatores mas também fortalecidos os satisfatores que motivam a compra, foi realizada a análise dos termos positivos. Assim, com respeito aos 97 termos positivos classificados pelo Semantria, utilizando também a base Ford_Dados_Brutos, foi gerada uma nuvem de palavras, das quais os 7 (sete) termos com maior densidade foram: "lindo" com 16 ocorrências (6%), "parabéns" com 11 ocorrências (4%), "melhor" com 10 ocorrências (4%), "sonho" com 10 ocorrências (4%), ótimo com 8 ocorrências (3%), "gostei" com 7 ocorrências (3%) e "confortável" com 7 ocorrências (3%). A nuvem de termos positivos, pode ser visualizada na Figura 47.

FIGURA 47 – NUVEM DE TERMOS POSITIVOS GERADOS NO SEMANTRIA

FONTE: Semantria (2018).

Uma vez que teoria afirma que a **qualidade** está relacionada às expectativas do consumidor, a constatação dos termos “lindo”, “bom”, “confortável” e “conforto”, dentre outros destacados na Figura 47 pode indicar a satisfação do consumidor com os produtos e serviços da empresa em questão.

Finalmente, a **fidelidade** dos consumidores para com uma marca, diretamente relacionada às percepções a respeito de um produto de uma marca à qual eles constroem sentimentos favoráveis (KOTLER, 2012) pode ser observada pela presença de termos “bom”, “lindo”, “sonho”, “maravilhoso” e “amo”, dentre outros.

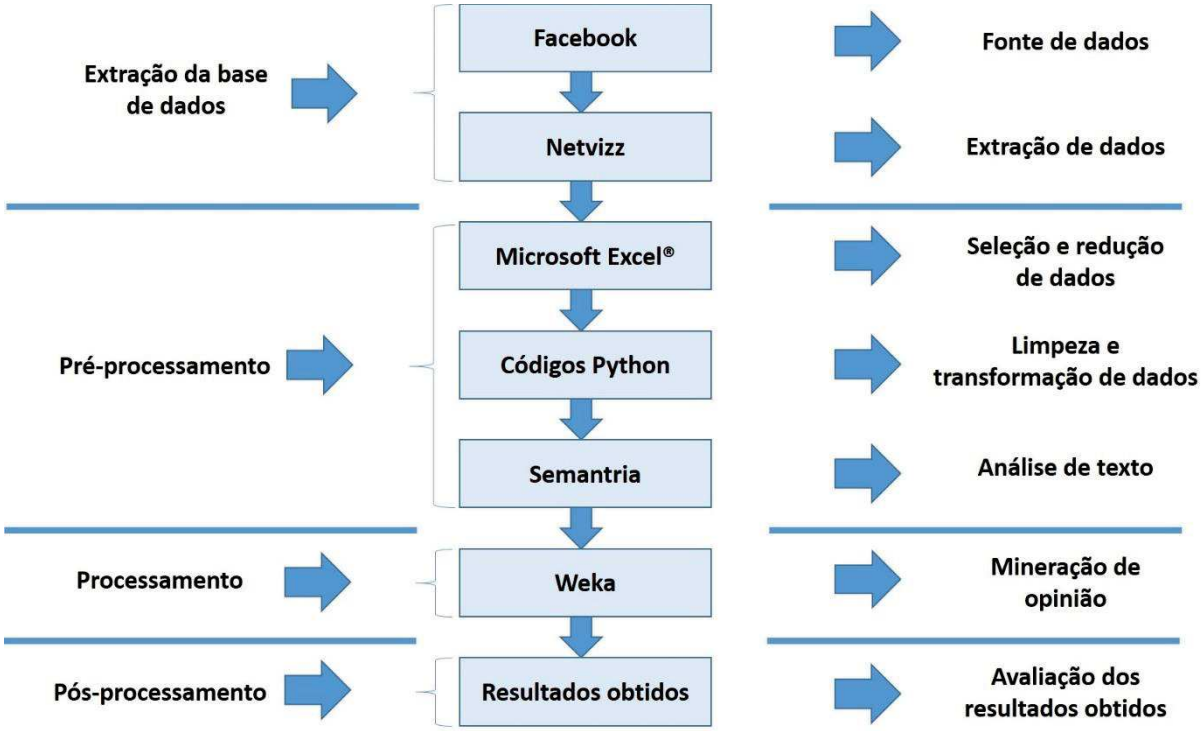
4.5 CONTRIBUIÇÃO PARA A METODOLOGIA DE ANÁLISE DE OPINIÕES

Um dos objetivos específicos definidos neste estudo foi o de contribuir com a metodologia de análise de opiniões, visando maximizar a automatização das etapas envolvidas no processo de extração da base de dados, pré-processamento, processamento e pós-processamento. Para criar uma visão geral da metodologia de

todo o processo compreendido neste estudo (seção 3.2), foi elaborado um fluxo abrangendo as etapas abordadas.

Na Figura 48 pode ser visualizado o fluxo com todas as etapas envolvidas no processo de mineração de opinião, bem como, as ferramentas utilizadas para cada etapa e as suas principais funções dentro do processo.

FIGURA 48 – FLUXO DA METODOLOGIA UTILIZADA PARA ANÁLISE DE OPINIÕES



FONTE: O autor (2018).

Após a apresentação e discussão dos resultados, a próxima seção apresenta as considerações finais e contempla a verificação dos objetivos, as principais contribuições e trabalhos futuros.

5 CONSIDERAÇÕES FINAIS

Inicialmente o projeto previa a utilização de uma base de dados SAC de organizações que possuíssem este serviço. Foram feitas tentativas com três empresas para se conseguir a base, porém, todas as empresas contatadas alegaram não ser possível a divulgação da mesma, por tratar-se de informações consideradas confidenciais e estratégicas para as companhias. O pesquisador deste projeto contatou dois professores do departamento de Administração de Empresas da UFPR que trabalham com a linha de comportamento de clientes e também confirmaram a dificuldade de se obter bases SAC.

A preferência de se utilizar uma base de dados SAC, seria pelo fato de possuir uma “linguagem” mais padronizada, já que normalmente estas bases são preenchidas por profissionais treinados, que possuem um padrão de escrita de registros mais calibrado entre elas. Assim, haveria menor probabilidade de se encontrar nestas bases termos como por exemplo o uso de gírias, abreviações tais como: “blz”, “flw”, “vlw” e emoticons como: “:)”, “:(“ e “:x”. Com isso, o esforço dispendido no pré-processamento da base poderia ser menor, sem a necessidade de um tratamento de texto mais complexo para adequar à mineração de opinião.

Como meio alternativo de fonte de dados, optou-se pela utilização do SAC 2.0, que se apresenta como uma outra opção de se conseguir a construção de uma base com opiniões, reclamações e sugestões de usuários nas mídias sociais, mais especificamente no Facebook que tem uma ampla utilização por parte dos usuários.

A base de dados utilizada foi extraída da página oficial da empresa Ford Brasil no Facebook, nos comentários efetuados na postagem de lançamento da nova linha Ford Ka 2018. A seleção desta base foi devido a que a mesma contém nas postagens opiniões positivas, negativas e neutras de maneira equilibrada, o que foi ideal para os objetivos deste estudo de avaliar tanto opiniões de satisfação como de insatisfação. O fato de ter sido selecionada uma empresa do ramo automobilístico é por ela compreender não somente as opiniões do produto em si, mas também de outros serviços e atendimentos envolvidos, como vendas, pós-vendas, revisões e garantia.

O grande desafio do estudo foi encontrar a forma adequada para proceder com o tratamento de texto ideal para transformar a linguagem natural utilizada nas mídias sociais em agrupamentos de textos em forma de padrões, que pudessem ser interpretados e adequados à mineração de opinião.

5.1 VERIFICAÇÃO DOS OBJETIVOS PROPOSTOS

Para poder alcançar o objetivo da verificação de como a mineração de opinião em bases de dados extraídas de mídias sociais podem contribuir na medição da (in)satisfação dos consumidores, houve a necessidade do atingimento de 3 objetivos específicos.

O primeiro objetivo de estudar a mineração de opiniões e propor ferramentas que auxiliem na extração e pré-processamento de bases de dados extraídas de mídias sociais foi atingido com êxito. As ferramentas Microsoft Excel®, os códigos em Python e o Semantria complementaram-se no sentido de cumprir cada uma com as diferentes tarefas de pré-processamento tais como selecionar e reduzir dados, limpeza e tratamento dos mesmos e concluindo com a análise de texto, para poder classificar automaticamente os comentários extraídos do Facebook em positivos, neutros e negativos. O NetVizz, embora tenha limitações pelas restrições de extração de dados ocasionadas pelas políticas estabelecidas pelo Facebook, demonstrou bastante eficiência para a finalidade deste estudo.

O segundo objetivo de escolher e aplicar ferramentas que podem ser utilizadas para a mineração de opiniões, também foi alcançado. O Weka é um software bastante utilizado em estudos e trabalhos que envolvem a mineração de opinião e atendeu perfeitamente à finalidade, em conjunto com os métodos de mineração de opinião que foram utilizados: algoritmos de classificação *Naive Bayes*, *Sequential Minimal Optimization* (SMO) e o J48 baseado em árvore de decisão.

O terceiro objetivo específico de contribuir com a metodologia de análise de opiniões, sumarizando as ferramentas e métodos utilizados, também foi atingido com sucesso. A partir da definição do local onde seria extraída a base e filtrando-se a postagem que seria estudada, foram selecionadas as ferramentas para as devidas funções em cada uma das etapas de extração de dados, pré-processamento, processamento e pós-processamento. Cada etapa do processo foi efetuada com pouca intervenção humana e a grande maioria das tarefas foi executada de maneira automática e semiautomática. À medida que se conseguia sucesso nas etapas, foi possível verificar que cada ferramenta selecionada atingiu o seu papel e preparou a base para a próxima ferramenta atuar. Ao término do processo foi possível criar-se um fluxo com a sequência e a atuação de cada uma das ferramentas em todas as etapas do processo de análise de opiniões.

Uma vez que a quantidade de informações contidas na *Web* tem um crescimento constante e diante desse enorme volume de dados a dificuldade de pessoas poderem explorá-las é cada vez maior, a utilização de máquinas e processamento de linguagem natural por meio do uso da mineração de opinião pode ser uma alternativa muito útil para mensurar a (in)satisfação dos consumidores de forma mais ágil.

Finalmente, este estudo constatou que com a adoção de técnicas de pré-processamento é possível preparar bases de dados de forma adequada para a mineração de opinião, que por sua vez irá criar agrupamentos de opiniões positivas, neutras e negativas e gerar a descoberta de padrões por meio de treinamentos efetuados por algoritmos de mineração adequados, e com isso auxiliar no processo de gestão e tomada de decisão. Com os resultados obtidos neste trabalho é possível afirmar que a mineração de opinião em bases de dados extraídas de mídias sociais pode contribuir na mensuração da (in)satisfação dos consumidores, considerando-se assim o objetivo alcançado.

5.2 CONTRIBUIÇÕES

Considera-se que, para fins acadêmicos, as bases utilizadas e os diversos procedimentos que envolveram o pré-processamento, a mineração de opinião e a avaliação dos resultados obtidos, poderá destacar a importância do monitoramento do grau de (in) satisfação dos clientes quanto aos produtos e serviços fornecidos pelas organizações. Ao mesmo tempo, com a descoberta de conhecimento em dados pela mineração de opinião e os padrões encontrados nas análises efetuadas, geraram informações que podem ser de utilidade e valor para as empresas, tais como: a busca de tendências de mercado, requisitos de clientes, aprimoramentos e melhorias, bem como recomendações e sugestões que os consumidores deixam registrados nas mídias sociais.

Além deste trabalho poder contribuir no processo de identificação das principais causas e níveis de (in)satisfação dos consumidores frente a produtos e serviços das empresas, pode contribuir também para os processos de avaliação da velocidade e acuracidade na identificação de resultados ao serem processados e classificados de forma automática pelo(s) algoritmo(s) de mineração, sem intervenção de recursos

humanos, bem como a avaliação e mensuração dos erros advindos dos processos de pré-processamento, mineração e pós-processamento.

5.3 TRABALHOS FUTUROS

Como sugestão para trabalhos futuros, recomenda-se aplicar este estudo utilizando-se uma base de dados SAC, onde existem algumas características distintas das encontradas em bases SAC 2.0 como linguagens mais padronizadas e especializadas (linguagens documentais) e registros textuais que permitam um menor esforço de pré-processamento por serem efetuados por profissionais especializados e treinados para conduzir este trabalho. Dessa forma, poderiam haver comparações entre ambos os resultados dos estudos das bases e assim retirarem-se conclusões mais específicas e com elas a criação de metodologias mais aprimoradas.

Outra sugestão é a de dar sequência a este estudo com pesquisas sobre o descobrimento de padrões das diferentes causas de (in)satisfação dos consumidores encontradas em bases de dados SAC e SAC 2.0, assim como análises de sugestões, questionamentos, elogios, estudos de comportamento do consumidor e possíveis descobertas e predições encontradas nos padrões estudados, que possam ser validados por especialistas que atestem a qualidade dos agrupamentos e dos padrões obtidos em textos e com o auxílio da linguística computacional, que permita a construção de vocabulários especializados com a finalidade de facilitar o reconhecimento de informações em linguagem natural.

REFERÊNCIAS

AGÊNCIA WCK. **O que é SAC 2.0 e por que a sua empresa deve ficar atenta a ele?** 2016. Disponível em: <https://agenciawck.com.br/o-que-e-sac-2-0-e-por-que-a-sua-empresa-deve-ficar-atenta-a-ele/>. Acesso em: 05 mai. 2018.

ALEXA. **Alexa top sites**: detailed description. 2018. Disponível em: <https://www.alexa.com/topsites/countries/BR>. Acesso em 13 mai. 2018.

CASTRO, L. N.; FERRARI, D. G. **Introdução à mineração de dados:** conceitos básicos, algoritmos e aplicações. São Paulo: Saraiva, 2016.

CHEN, H.; ZIMBRA, D. AI and opinion mining. **IEEE intelligent systems**, v. 10, p. 74–80, 2010. Disponível em: <https://pdfs.semanticscholar.org/42ca/bbf853768c1a57586e7963dc0f5ff02fbfbd.pdf>. Acesso em: 15 mai. 2018.

COBRA, M. **Administração de marketing no Brasil.** 3. ed. Rio de Janeiro: Elsevier, 2009.

CRISTIANINI, N.; SHAWE-TAYLOR, J. **An introduction to support vector machine and other kernel-based learning methods.** United Kingdom: Cambridge, 2013.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. From data mining to knowledge discovery in databases. **AI magazine**, v. 17, n. 3, p. 37–54, 1996. Disponível em: <http://www.aaai.org/ojs/index.php/aimagazine/article/view/1230/1131>. Acesso em: 20 mai. 2018.

FRANCO, A. H. C. **Inteligência coletiva:** manifestações nos ambientes digitais. 2018. 141 f. Tese (Doutorado em Ciência da Informação) - Faculdade de Filosofia e Ciências, Universidade Estadual Paulista, 2018. Disponível em: <https://repositorio.unesp.br/bitstream/handle/11449/152741/franco_ahc_dr_mar.pdf?sequence=3>. Acesso em: 13 mai. 2018.

GERSON, R. F. **A excelência no atendimento a clientes:** mantendo seus clientes por toda a vida. Rio de Janeiro: Qualitymark, 2001.

GIL, A. C. **Como elaborar projetos de pesquisa.** 4. ed. São Paulo: Atlas, 2002.

GIL, A. C. **Métodos e técnicas de pesquisa social.** 6. ed. São Paulo: Atlas, 2008.

GONSALVES, A; FIRST, L. **SAC 2.0 como parte do planejamento estratégico de comunicação.** 56 f. Trabalho de Graduação (Disciplina Tema Final) - Curso de Comunicação Social, Setor de Artes, Comunicação Social e Design, Universidade Federal do Paraná, Curitiba, 2013. Disponível em: <https://acervodigital.ufpr.br/bitstream/handle/1884/52858/TCC_sac_2.0_como_part

e_do_planejamento_estrategico_de_comunicacao.pdf?sequence=1>. Acesso em: 15 mai. 2018.

GOOGLE DEVELOPERS. **Classification**: true vs. false and positive vs. negative. 2018. Disponível em: < https://developers.google.com/machine-learning/crash-course/classification/true-false-positive-negative>. Acesso em 03 nov. 2018.

JOHNSON, A. J. **Su éxito en redes sociales**. 2014. E-book. Disponível em: <https://pt.scribd.com/document/313463715/Su-Exito-en-Redes-Sociales-Amanda-J-Johnson>. Acesso em 13 mai. 2018.

KAPLAN, A.M.; HAENLEIN, M. Users of the world, unite: the challenges and opportunities of social media. In: **Business Horizons**. v. 53, n. 1, 2010. p. 59-68. Disponível em: <http://michaelhaenlein.eu/Publications/Kaplan,%20Andreas%20-%20Users%20of%20the%20world,%20unite.pdf>. Acesso em: 24 out. 2018.

KOTLER, P. **Administração de marketing.** 10. ed. São Paulo: Afiliada, 2002.

LAKATOS, M. E.; MARCONI, M. de A. **Metodologia do trabalho cientifico.** 4. ed. São Paulo: Atlas, 1992.

HILL, N.; SELF, B.; ROCHE, G. **Customer satisfaction measurement for ISO 9000:2000.** Oxford: Butterworth-Heinemann, 2002.

LIU, B. **Sentiment analysis**: mining opinions, sentiments and emotions. New York: Cambridge University, 2015.

LUNARDI, A. C.; VITERBO, J.; BERNARDINI, F. C. Um levantamento do uso de algoritmos de aprendizado supervisionado em mineração de opiniões. PROCEEDINGS OF XII ENCONTRO NACIONAL DE INTELIGÊNCIA ARTIFICIAL E COMPUTACIONAL, 12., 2015, Natal. **Anais...** Natal: ENIAC, 2015. p. 262-269. Disponível em: <http://www.lbd.dcc.ufmg.br/colecoes/eniac/2015/039.pdf>. Acesso em 22 out. 2018.

NEVES, R. C. D. **Pré-processamento no processo de descoberta de conhecimento em banco de dados.** 2003. 137 f. DISSERTAÇÃO (Mestrado) - Programa de Pós-graduação em Computação, Instituto de Informática, Universidade Federal do Rio Grande do Sul. Disponível em: < http://www.lume.ufrgs.br/bitstream/handle/10183/2701/000375412.pdf?sequence=1> . Acesso em: 28 mai. 2018.

PMEAN. **What is a Kappa coefficient?** Disponível em: <http://www.pmean.com/definitions/kappa.htm>. Acesso em: 26 out. 2018.

QUINLAN, J. R. **C4.5:** Programs for Machine Learning. Morgan Kaufmann Publishers, 1993.

RIEDER, B. Studying Facebook via data extraction: the Netvizz application. In: ANNUAL ACM WEB SCIENCE CONFERENCE, 5., 2013, New York. **Proceedings…** New York: ACM, 2013. p. 346-355. Disponível em:

<http://thepoliticsofsystems.net/permafiles/rieder_websci.pdf>. Acesso em: 27 mai. 2018.

RODRIGUES, A. C. F. **Modelo para análise de sentimentos no Facebook:** um estudo de caso na página do senado federal brasileiro. 83 f. TCC (Graduação) – Curso de Gestão da Informação, Universidade Federal do Paraná, Curitiba, 2017. Disponível em: <https://acervodigital.ufpr.br/bitstream/handle/1884/54864/Alan%20Cristian%20Falcoski%20Rodrigues.pdf?sequence=1&isAllowed=y>. Acesso em: 11 out. 2018.

OLIVA, A. D.; FERNANDEZ, A. D. Empole-o de los medios sociales en educación superior: una nueva competencia docente en ciernes. **Revista de docência universitária,** v. 10, n. 2, p. 365-379, mai./ago. 2012. Disponível em: <http://red-u.net/redu/documentos/vol10_n2_completo.pdf>. Acesso em: 14 mai. 2018.

POST DIGITAL. **Qual a diferença entre rede social e mídia social?** 2018. Disponível em: <http://www.postdigital.cc/blog/artigo/qual-a-diferenca-entre-rede-social-e-midia-social#04>. Acesso em: 24 out. 2018.

SANTOS, L. M. et al. Twitter, análise de sentimento e desenvolvimento de produtos: quanto os usuários estão expressando suas opiniões? **Revista PRISMA.COM**, n. 13, 2010 p. 1-1. Disponível em: < http://revistas.ua.pt/index.php/prismacom/article/view/790/722>. Acesso em 06 jun. 2018.

LEXALYTICS. **Semantria for Excel.** 2018. Disponível em: <https://www.lexalytics.com/semantria/excel>. Acesso em: 08 out. 2018.

SCHIESSL, M.; BRÄSCHER, M. Descoberta de conhecimento em texto aplicada a um sistema de atendimento ao consumidor. **Revista Ibero-Americana de Ciência da Informação,** v. 4, n. 2, 2011. Disponível em: <http://periodicos.unb.br/ojs311/index.php/RICI/article/view/1682/1481>. Acesso em: 22 abr. 2018.

SLACK, N; LEWIS, M. **Operations strategy.** Harlow: Pearson Education, 2002.

TEIXEIRA, D.; AZEVEDO, I. Análise de opiniões expressas nas redes sociais. **Revista Ibérica de Sistemas e Tecnologias de Informação – Risti**, Porto, n. 8, p. 53-65, dez. 2011. Disponível em: <http://www.scielo.gpeari.mctes.pt/pdf/rist/n8/n8a06.pdf>. Acesso em: 11 out. 2018.

TEIXEIRA, M. B. **SAC 2.0 - uma análise das estratégias de relacionamento com o cliente no ponto frio.** 45 f. Trabalho de Graduação (Bacharelado em Comunicação Social), Universidade Católica de Brasília, Brasília, 2014. Disponível em:<https://repositorio.ucb.br/jspui/bitstream/10869/4535/1/Mariana%20Brasil%20Teixeira.pdf>. Acesso em: 14 mai. 2018.

TELLES, A. **A revolução das mídias sociais**: cases, conceitos, dicas e ferramentas. São Paulo, M.Books do Brasil Editora Ltda, 2010.

APÊNDICE A – MODELO DE BASE DE DADOS EM FORMATO “ARFF”

```
% Ford - Base de dados bruta

@relation fordbaseBruta

@attribute HighlightedText string
@attribute PhraseSentiment {positive, neutral, negative}

@data
'...mos com um Ford Ka 2018 com problema de alto consumo de combustível e nem a concessionária, ...',neutral
'...Ford resolvem o problema. Estamos nessa briga há 7 meses. Péssimo atendimento!',negative
'...omprem! Estamos com um Ford Ka 2018 com problema de alto consumo de combustível e nem a ...',negative
'...oblema. Estamos nessa briga há 7 meses. Péssimo atendimento!',negative
'Muito bom..',positive
'Um dia vou ter Ford pra mim a melhor marca de todos os tempos.',positive
'...stou atrás dos meus direitos ..um carro praticamente novo .não sei que tipo de carro vcs estão ve...',neutral
'...concessionária por q ele apresentou uns tranco na troca de marcha . Custo 20 mil pra a...',neutral
'...eitos ..um carro praticamente novo .não sei que tipo de carro vcs estão vendendo .n...',negative
'Lindas maravilhosa',positive
'Lindas maravilhosa',positive
'...mático ou automatizado modelo powershift?',neutral
'...tratual, pq não foi me fornecido antes? Sei que estou no prejuízo diariamente pagan...',positive
'...tras vezes que o carro deu entrada para solucionar esse problema, solicitei o carro reserv...',neutral
'...ontratual, pq não foi me fornecido antes? Sei que estou no prejuízo diariamente p...',neutral
'Tenho um ford ka 2015 que apresenta defeito de fabrica, um barulho na ré desde o pr...',negative
'... que é um defeito de fábrica, que nunca solucionam em definitivo e que ficam com o carro s...',negative
'...abalho para o outro, pois sem o mesmo é impossível cumprir os horários estabelecidos. E pas...',negative
'...el cumprir os horários estabelecidos. E pasmem, hoje quando falei da possibilidade de ...',negative
'... carro deu entrada para solucionar esse problema, solicitei o carro reserva e nunca me f...',negative
'...oi me fornecido antes? Sei que estou no prejuízo diariamente pagando táxi, com um carro q...',negative
.
```

APÊNDICE B – MODELOS DE RESULTADOS DO *SOFTWARE* SEMANTRIA

- *PhraseDetail*

nao comprem ford ka problema alto consumo combustivel	neutral
...concessionaria ford resolvem problema briga pessimo	negative
nao comprem ford ka problema alto consumo combustivel	negative
bom	positive
...direitos um carro praticamente novo nao sei tipo carro vendendo	positive
...incadeira ne ja atras direitos um carro praticamente novo nao sei	neutral
...nte levei concessionaria apresentou uns tranco troca marcha	neutral
lindas maravilhosa	positive
lindas maravilhosa	positive
ford melhor marca tempos	positive
...rva detalhe outras vezes carro entrada solucionar problema	neutral
...e outras vezes carro entrada solucionar problema solicitei carro	neutral
ford ka apresenta defeito fabrica barulho re desde primeiro saiu...	neutral
...ise revoltante defeito fabrica nunca solucionam definitivo carro	negative
...ossivel cumprir horarios estabelecidos pasmem falei	negative
... fabrica achar real causa insatisfacao revolta define numero	negative
...rantia contratual nao fornecido antes sei prejuizo diariamente	negative
... serie consumidores ford ka merecemos respeito assistencia	positive
...r carro comprei paguei obrigacao achar solucionar definitivo	neutral

- *ThemeDetail*

carros ford sao bons nao falar concessionaria cidade pedi falar	concessionaria	1	0,150000006	neutral	4	concessionaria	concessionaria
...ores falta treinamento parece insistir comprar parece nao vender	comprar parece	1	0,150000006	neutral	4	comprar parece	comprar parece
kaamo queria carro alto	kaamo queria	1	0	neutral	4	kaamo queria	kaamo queria
aeeee galera encontrado tao esperto hein conhecida filme	esperto hein	1	0,800000012	positive	4	esperto hein	esperto hein
...e galera encontrado tao esperto hein conhecida filme acharem	conhecida filme	1	0,800000012	positive	4	conhecida filme	conhecida filme
aeeee galera encontrado tao esperto hein conhecida filme ach...	galera	1	0,800000012	positive	4	galera	galera
fiesta rocan dando problema superaquecimento retirado acabar	dando problema	1	-0,800000012	negative	4	dando problema	dando problema
...lema superaquecimento retirado acabar problema motor novo	problema motor	1	-0,800000012	negative	4	problema motor	problema motor
carro sao bom cd economico	carro sao	1	0,800000012	positive	4	carro sao	carro sao
ford boa tarde carro realmente lindo unico problema ca...	ford boa tarde	1,5	0,100000001	neutral	4	ford boa tarde	ford boa tarde
...ntes claros talvez nao consiga resolver comentario pediria favor	comentario	1,5	0,100000001	neutral	4	comentario	comentario
...olver comentario pediria favor liberado dessas cores faturamento	dessas cores	1,5	0,100000001	neutral	4	dessas cores	dessas cores
...d boa tarde carro realmente lindo unico problema caso cor prata	problema caso	1	-0,800000012	negative	4	problema caso	problema caso
...rro realmente lindo unico problema caso cor prata cinza moscou	cor prata	1	0,100000001	neutral	4	cor prata	cor prata
...caso cor prata cinza moscou versao nao disponiveis cidade	disponiveis	1	0,100000001	neutral	4	disponiveis	disponiveis
a moscou versao nao disponiveis cidade montes claros talvez nao	montes claros	1	0,100000001	neutral	4	montes claros	montes claros

- *EntityThemeDetail*

Document ID	Highlighted Text	Entity	Entity Type	Entity Theme	Entity Theme Sentiment	Entity Theme Sentiment +/-	Entity Theme Sentiment Evidenc
1	nao comprem ford ka problema alto consumo combustivel	ford	veículo	ka problema	-0,800000012	negative	4
1	...d ka problema alto consumo combustivel concessionaria ford	ford	veículo	concessionaria ford	-0,572878301	negative	4
1	...onsumo combustivel concessionaria ford resolvem problema	ford	veículo	resolvem problema	-0,800000012	negative	4
1	...onaria ford resolvem problema briga pessimo atendimento	ford	veículo	pessimo atendimento	-0,572878301	negative	4
2	...direitos um carro praticamente novo nao sei tipo carro vendendo	ford	veículo	sei tipo	0,699999988	positive	4
2	... custo mil arrumar brincadeira ne ja atras direitos um carro	ford	veículo	atras direitos	0,070183992	neutral	4
2	...esmente levei concessionaria apresentou uns tranco troca marcha	ford	veículo	uns tranco	-0,25	neutral	4
2	ford melhor marca tempos	ford	veículo	ford melhor	1	positive	4
2	ford melhor marca tempos	ford	veículo	marca tempos	1	positive	4
3	ford ka apresenta defeito fabrica barulho re desde primeiro saiu	ford	veículo	fabrica barulho	-0,295238107	neutral	4
3	...to fabrica nunca solucionam definitivo carro devolver preciso	ford	veículo	carro devolver	-0,295238107	neutral	4
3	...r horarios estabelecidos pasmem falei possibilidade carro	ford	veículo	possibilidade carro	-0,295238107	neutral	4

- *EntityDetail*

Document ID	Highlighted Text	Entity	Entity Type	Entity Theme	Entity Theme Sentiment	Entity Theme Sentiment +/-	Entity Theme Sentiment Evidenc
1	nao comprem ford ka problema alto consumo combustivel	ford	veículo	ka problema	-0,800000012	negative	4
1	...d ka problema alto consumo combustivel concessionaria ford	ford	veículo	concessionaria ford	-0,572878301	negative	4
1	...onsumo combustivel concessionaria ford resolvem problema	ford	veículo	resolvem problema	-0,800000012	negative	4
1	...onaria ford resolvem problema briga pessimo atendimento	ford	veículo	pessimo atendimento	-0,572878301	negative	4
2	...direitos um carro praticamente novo nao sei tipo carro vendendo	ford	veículo	sei tipo	0,699999988	positive	4
2	... custo mil arrumar brincadeira ne ja atras direitos um carro	ford	veículo	atras direitos	0,070183992	neutral	4
2	...esmente levei concessionaria apresentou uns tranco troca marcha	ford	veículo	uns tranco	-0,25	neutral	4
2	ford melhor marca tempos	ford	veículo	ford melhor	1	positive	4
2	ford melhor marca tempos	ford	veículo	marca tempos	1	positive	4
3	ford ka apresenta defeito fabrica barulho re desde primeiro saiu	ford	veículo	fabrica barulho	-0,295238107	neutral	4
3	...to fabrica nunca solucionam definitivo carro devolver preciso	ford	veículo	carro devolver	-0,295238107	neutral	4

- *QueryCategoryDetail*

Document ID	Highlighted Text	Query Category	Query Category Sentiment	Query Category Sentiment +/-	Query Category Relevan
5	...ram nenhum carro menos transporte filha escola nao aconselho	Educação	0,5	neutral	1
11	nao faca consorcio empresa prestadora ford ford telefone	Negócios	0	neutral	1
12	faltando famoso ford k	Cultura Pop	0,699999988	positive	1
18	ford ka sedan famoso comercial raul lemos	Cultura Pop	0,699999988	positive	1
41	...forma nao possivel solucao diretamente empresa levei veiculo	Negócios	-0,5	negative	1
47	...i concessionaria nao aprovada garantia empresa lixo	Negócios	-0,600000024	negative	1
48	pecas autorizada acha baba internet preco alto bom carro ford	Tecnologia	0,300000012	neutral	1
50	...rentemente resolvido voltou procurei internet assustei numero	Tecnologia	-0,800000012	negative	1
57	ford brasil conceitosa empresa automilitica tecnologia inovacao	Negócios	2,200000048	positive	1
60	...ndedor mundo cliente satisfeito vende empresa marca produto	Negócios	1,799999952	positive	1
70	...rd nao propagando proprios clientes famosas usam carro marca	Cultura Pop	0,699999988	positive	1

- *ConceptTopicDetail*

Document ID	Source Text	Concept Topic	Concept Topic Sentiment	Concept Topic Sentiment +/	Concept Topic Strength
1	nao comprem ford ka problema alto consumo combustivel	Bebidas	-0,572878301	negative	0,806327641
1	nao comprem ford ka problema alto consumo combustivel	Economia	-0,572878361	negative	0,723573864
1	nao comprem ford ka problema alto consumo combustivel	Comida	-0,572878361	negative	0,567705393
1	nao comprem ford ka problema alto consumo combustivel	Automotiva	-0,572878301	negative	0,543137312
2	fusion milkm rodado simplesmente levei concessionaria	Automotiva	0,070183992	neutral	0,747153163
2	ford melhor marca tempos	Automotiva	1	positive	0,459207922
3	ford ka apresenta defeito fabrica barulho re desde primeiro saiu	Automotiva	-0,295238107	neutral	0,798477829
3	ford ka apresenta defeito fabrica barulho re desde primeiro saiu	Trabalho	-0,295238107	neutral	0,55402267
3	ford ka apresenta defeito fabrica barulho re desde primeiro saiu	Setor Imobiliário	-0,295238107	neutral	0,50632
3	ford ka apresenta defeito fabrica barulho re desde primeiro saiu	Tecnologia	-0,295238107	neutral	0,500597298
3	cambio automatico automatizado modelo powershift	Automotiva	0	neutral	0,684233665
4	desde ford ka defeito fabrica barulho insuportavel re descola carro	Automotiva	0,008333336	neutral	0,761067331
4	gostaria comprar ford ka desanimei	Automotiva	0,150000006	neutral	0,66334033

- *DocumentDetail*

Document ID	Status	Source Text	Summary	Detected Language	Detected Language Scor	Document Sentiment	Document Sentiment +/
1	PROCESSED	nao comprem ford ka problema alto consumo	nao comprem ford ka problema alto consumo	Portuguese	0,4681958	-0,572878301	negative
1	PROCESSED	quero desse conseguir querer desse troca	quero desse conseguir querer desse troca	Portuguese	0,8898652	0	neutral
1	PROCESSED	bom	bom	Portuguese	0,5743513	0,800000012	positive
2	PROCESSED	fusion milkm rodado simplesmente levei	fusion milkm rodado simplesmente levei	Portuguese	0,50492436	0,070183992	neutral
2	PROCESSED	ford melhor marca tempos	ford melhor marca tempos	Portuguese	0,9789812	1	positive
2	PROCESSED	lindas maravilhosa	lindas maravilhosa	Unknown	1	0,800000012	positive
3	PROCESSED	ford ka apresenta defeito fabrica barulho re	ford ka apresenta defeito fabrica barulho re desde	Portuguese	0,6886528	-0,295238107	negative
3	PROCESSED	cambio automatico automatizado modelo	cambio automatico automatizado modelo	Spanish	0,7082999	0	neutral
3	PROCESSED	show show	show show	English	0,5483819	0	neutral
4	PROCESSED	desde ford ka defeito fabrica barulho	desde ford ka defeito fabrica barulho insuportavel	Portuguese	0,6034285	0,008333336	neutral
4	PROCESSED	gostaria comprar ford ka desanimei	gostaria comprar ford ka desanimei	Portuguese	0,6963219	0,150000006	neutral
4	PROCESSED	ansiosa conhecer novo ka	ansiosa conhecer novo ka	Portuguese	0,999995	-0,150000006	negative
5	PROCESSED	pessima experiencia carros ford segunda	pessima experiencia carros ford segunda	Portuguese	0,4885809	-0,578571439	negative
5	PROCESSED	humilde ford ka tornar kkkk mds	humilde ford ka tornar kkkk mds	Unknown	1	0	neutral

APÊNDICE C – RESULTADO BASE "FORD_DADOS_BRUTOS" COM ALGORITMO "*NAIVEBAYES*"

Scheme: weka.classifiers.meta.FilteredClassifier -F "weka.filters.unsupervised.attribute.StringToWordVector -R first-last -W 1000 -prune-rate -1.0 -N 0 -stemmer weka.core.stemmers.NullStemmer -stopwords-handler weka.core.stopwords.Null -M 1 -tokenizer \"weka.core.tokenizers.WordTokenizer -delimiters \\\" \\\\r\\\\n\\\\t.,;:\\\\\\\'\\\\\\\"()?!\\\"\"" -W weka.classifiers.bayes.NaiveBayes
Relation: Ford_Dados_Brutos
Instances: 430
Attributes: 2
HighlightedText
PhraseSentiment
Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===
FilteredClassifier using weka.classifiers.bayes.NaiveBayes on data filtered through weka.filters.unsupervised.attribute.StringToWordVector -R 1 -W 1000 -prune-rate -1.0 -N 0 -stemmer weka.core.stemmers.NullStemmer -stopwords-handler weka.core.stopwords.Null -M 1 -tokenizer "weka.core.tokenizers.WordTokenizer -delimiters \" \\r\\n\\t.,;:\\\'\\\"()?!\""
Filtered Header
@relation 'Ford_Dados_Brutos-weka.filters.unsupervised.attribute.StringToWordVector-R1-W1000-prune-rate-1.0-N0-stemmerweka.core.stemmers.NullStemmer-stopwords-handlerweka.core.stopwords.Null-M1-tokenizerweka.core.tokenizers.WordTokenizer -delimiters \" \\r\\n\\t.,;:\\\'\\\"()?!\"'

```
@attribute PhraseSentiment {positive,neutral,negative}
@attribute #ford numeric
@attribute 0 numeric
@attribute 0km numeric
@attribute 1 numeric
@attribute 2001 numeric
…

Classifier Model
Naive Bayes Classifier
                        Class
Attribute           positive  neutral negative
                      (0.23)   (0.45)   (0.32)
================================================
#ford
  mean                0.0103        0        0
  std. dev.           0.1667   0.1667   0.1667
  weight sum              97      196      137
  precision                1        1        1

0
  mean                0.0412    0.051        0
  std. dev.           0.1988     0.22   0.1667
  weight sum              97      196      137
  precision                1        1        1

0km
  mean                0.0103        0        0
  std. dev.           0.1667   0.1667   0.1667
  weight sum              97      196      137
  precision                1        1        1
…
```

```
Time taken to build model: 0.15 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         290               67.4419 %
Incorrectly Classified Instances       140               32.5581 %
Kappa statistic                          0.4888
Mean absolute error                      0.23
Root mean squared error                  0.4269
Relative absolute error                 53.8878 %
Root relative squared error             92.4342 %
Total Number of Instances              430

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area
Class
                 0,629    0,099    0,649      0,629    0,639      0,536    0,861     0,737
positive
                 0,730    0,256    0,704      0,730    0,717      0,472    0,811     0,805
neutral
                 0,628    0,160    0,647      0,628    0,637      0,471    0,836     0,698
negative
Weighted Avg.    0,674    0,190    0,673      0,674    0,674      0,486    0,830     0,755

=== Confusion Matrix ===

   a   b   c   <-- classified as
  61  24  12 |   a = positive
  18 143  35 |   b = neutral
  15  36  86 |   c = negative
```

APÊNDICE D – RESULTADO BASE “FORD_DADOS_BRUTOS” COM ALGORITMO “SMO”

=== Run information ===

Scheme: weka.classifiers.meta.FilteredClassifier -F "weka.filters.unsupervised.attribute.StringToWordVector -R first-last -W 1000 -prune-rate -1.0 -N 0 -stemmer weka.core.stemmers.NullStemmer -stopwords-handler weka.core.stopwords.Null -M 1 -tokenizer \"weka.core.tokenizers.WordTokenizer -delimiters \\\" \\\\r\\\\n\\\\t.,;:\\\\\\\'\\\\\\\"()?!\\\"\"" -W weka.classifiers.functions.SMO -- -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -E 1.0 -C 250007" -calibrator "weka.classifiers.functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4"
Relation: Ford_Dados_Brutos
Instances: 430
Attributes: 2
HighlightedText
PhraseSentiment
Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

FilteredClassifier using weka.classifiers.functions.SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -E 1.0 -C 250007" -calibrator "weka.classifiers.functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4" on data filtered through weka.filters.unsupervised.attribute.StringToWordVector -R 1 -W 1000 -prune-rate -1.0 -N 0 -stemmer weka.core.stemmers.NullStemmer -stopwords-handler weka.core.stopwords.Null -M 1 -tokenizer "weka.core.tokenizers.WordTokenizer -delimiters \" \\r\\n\\t.,;:\\\'\\\"()?!\""

Filtered Header
@relation 'Ford_Dados_Brutos-weka.filters.unsupervised.attribute.StringToWordVector-R1-W1000-prune-rate-1.0-N0-stemmerweka.core.stemmers.NullStemmer-stopwords-handlerweka.core.stopwords.Null-M1-tokenizerweka.core.tokenizers.WordTokenizer -delimiters \" \\r\\n\\t.,;:\\\'\\\"()?!\"'

```
@attribute PhraseSentiment {positive,neutral,negative}
@attribute #ford numeric
@attribute 0 numeric
@attribute 0km numeric
@attribute 1 numeric
…
Classifier Model
SMO
Kernel used:
  Linear Kernel: K(x,y) = <x,y>

Classifier for classes: positive, neutral
BinarySMO

Machine linear: showing attribute weights,
not support vectors.
        -0.2615 * (normalized) #ford
 +       0.0711 * (normalized) 0
 +      -0.0388 * (normalized) 0km
 +       0.0405 * (normalized) 1
 +      -0.1749 * (normalized) 2001
 +      -0.0157 * (normalized) 2004
 …
Number of kernel evaluations: 53199 (95.257% cached)
```

```
Time taken to build model: 0.19 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         315               73.2558 %
Incorrectly Classified Instances       115               26.7442 %
Kappa statistic                          0.5799
Mean absolute error                      0.2977
Root mean squared error                  0.3847
Relative absolute error                 69.7583 %
Root relative squared error             83.2899 %
Total Number of Instances              430

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area
Class
                 0,691    0,090    0,691      0,691    0,691      0,601    0,826     0,608
positive
                 0,786    0,222    0,748      0,786    0,766      0,562    0,786     0,690
neutral
                 0,686    0,113    0,740      0,686    0,712      0,586    0,837     0,643
negative
Weighted Avg.    0,733    0,157    0,732      0,733    0,732      0,578    0,811     0,656

=== Confusion Matrix ===

   a   b   c   <-- classified as
  67  20  10 |   a = positive
  19 154  23 |   b = neutral
  11  32  94 |   c = negative
```

APÊNDICE E – RESULTADO BASE “FORD_DADOS_BRUTOS” COM ALGORITMO “J48”

=== Run information ===

Scheme: weka.classifiers.meta.FilteredClassifier -F "weka.filters.unsupervised.attribute.StringToWordVector -R first-last -W 1000 -prune-rate -1.0 -N 0 -stemmer weka.core.stemmers.NullStemmer -stopwords-handler weka.core.stopwords.Null -M 1 -tokenizer \"weka.core.tokenizers.WordTokenizer -delimiters \\\" \\\\r\\\\n\\\\t.,;:\\\\\\\'\\\\\\\"()?!\\\\\"\"" -W weka.classifiers.trees.J48 -- -C 0.25 -M 2
Relation: Ford_Dados_Brutos
Instances: 430
Attributes: 2
HighlightedText
PhraseSentiment
Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

FilteredClassifier using weka.classifiers.trees.J48 -C 0.25 -M 2 on data filtered through weka.filters.unsupervised.attribute.StringToWordVector -R 1 -W 1000 -prune-rate -1.0 -N 0 -stemmer weka.core.stemmers.NullStemmer -stopwords-handler weka.core.stopwords.Null -M 1 -tokenizer "weka.core.tokenizers.WordTokenizer -delimiters \" \\r\\n\\t.,;:\\\'\\\"()?!\""

Filtered Header
@relation 'Ford_Dados_Brutos-weka.filters.unsupervised.attribute.StringToWordVector-R1-W1000-prune-rate-1.0-N0-stemmerweka.core.stemmers.NullStemmer-stopwords-handlerweka.core.stopwords.Null-M1-tokenizerweka.core.tokenizers.WordTokenizer -delimiters \" \\r\\n\\t.,;:\\\'\\\"()?!\"'

```
@attribute PhraseSentiment {positive,neutral,negative}
@attribute #ford numeric
@attribute 0 numeric
@attribute 0km numeric
@attribute 1 numeric
...
Classifier Model
J48 pruned tree
------------------

minivan <= 0
|   q <= 0
|   |   jÃ¡ <= 0
|   |   |   Brasil <= 0
|   |   |   |   marca <= 0
|   |   |   |   |   jovem <= 0
|   |   |   |   |   |   levei <= 0
|   |   |   |   |   |   |   der <= 0
|   |   |   |   |   |   |   |   0 <= 0
|   |   |   |   |   |   |   |   |   defeito <= 0
|   |   |   |   |   |   |   |   |   |   som <= 0
|   |   |   |   |   |   |   |   |   |   |   pois <= 0
|   |   |   |   |   |   |   |   |   |   |   |   fui <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   garantia <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   problemas <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   Tive <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   caso <= 0
```

```
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   nem <=
0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   ano
<= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
por <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   sou <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   Mandei <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   criticas <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   |   pelo <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   |   |   ford <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   resposta <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   isso <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   |
...

Number of Leaves  :     71

Size of the tree :      141


Time taken to build model: 2.53 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         294               68.3721 %
Incorrectly Classified Instances       136               31.6279 %
Kappa statistic                          0.5003
Mean absolute error                      0.2457
Root mean squared error                  0.4169
Relative absolute error                 57.5776 %
Root relative squared error             90.2609 %
Total Number of Instances              430

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area
Class
                 0,619    0,081    0,690      0,619    0,652      0,559    0,802     0,590
positive
                 0,745    0,278    0,692      0,745    0,717      0,465    0,771     0,735
neutral
                 0,642    0,150    0,667      0,642    0,654      0,497    0,797     0,641
negative
Weighted Avg.    0,684    0,193    0,683      0,684    0,683      0,497    0,786     0,672

=== Confusion Matrix ===

   a   b   c   <-- classified as
  60  26  11 |   a = positive
  17 146  33 |   b = neutral
  10  39  88 |   c = negative
```

APÊNDICE F – RESULTADO BASE “FORD_COM_STOPWORDS_NB” COM ALGORITMO “*NAIVE BAYES*”

=== Run information ===

Scheme: weka.classifiers.meta.FilteredClassifier -F "weka.filters.unsupervised.attribute.StringToWordVector -R first-last -W 1000 -prune-rate -1.0 -N 0 -stemmer weka.core.stemmers.NullStemmer -stopwords-handler weka.core.stopwords.Null -M 1 -tokenizer \"weka.core.tokenizers.WordTokenizer -delimiters \\\" \\\\r\\\\n\\\\t.,;:\\\\\\'\\\\\\\"()?!\\\"\"" -W weka.classifiers.bayes.NaiveBayes
Relation: Ford_Com_StopWords
Instances: 1241
Attributes: 2
HighlightedText
PhraseSentiment
Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

FilteredClassifier using weka.classifiers.bayes.NaiveBayes on data filtered through weka.filters.unsupervised.attribute.StringToWordVector -R 1 -W 1000 -prune-rate -1.0 -N 0 -stemmer weka.core.stemmers.NullStemmer -stopwords-handler weka.core.stopwords.Null -M 1 -tokenizer "weka.core.tokenizers.WordTokenizer -delimiters \" \\r\\n\\t.,;:\\\'\\\"()?!\""
Filtered Header
@relation 'Ford_Com_StopWords-weka.filters.unsupervised.attribute.StringToWordVector-R1-W1000-prune-rate-1.0-N0-stemmerweka.core.stemmers.NullStemmer-stopwords-handlerweka.core.stopwords.Null-M1-tokenizerweka.core.tokenizers.WordTokenizer -delimiters \" \\r\\n\\t.,;:\\\'\\\"()?!\"'

```
@attribute PhraseSentiment {positive,neutral,negative}
@attribute 0 numeric
@attribute 0km numeric
@attribute 10 numeric
@attribute 15 numeric
...
Classifier Model
Naive Bayes Classifier
                                    Class
Attribute                         positive  neutral negative
                                    (0.13)   (0.69)   (0.18)
============================================================
0
  mean                              0.0187   0.0047        0
  std. dev.                         0.1667   0.1667   0.1667
  weight sum                           160      855      226
  precision                              1        1        1

0km
  mean                              0.0437   0.0175    0.031
  std. dev.                         0.2045   0.1667   0.1732
  weight sum                           160      855      226
  precision                              1        1        1

10
  mean                              0.0563   0.0175   0.0088
  std. dev.                         0.2304   0.1667   0.1667
  weight sum                           160      855      226
...
```

```
Time taken to build model: 0.52 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         946               76.2288 %
Incorrectly Classified Instances       295               23.7712 %
Kappa statistic                          0.5415
Mean absolute error                      0.1732
Root mean squared error                  0.3684
Relative absolute error                 54.588  %
Root relative squared error             92.5217 %
Total Number of Instances             1241

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0,713    0,079    0,573      0,713    0,635      0,579    0,922     0,604     positive
               0,777    0,233    0,881      0,777    0,825      0,515    0,839     0,927     neutral
               0,743    0,118    0,583      0,743    0,654      0,572    0,899     0,679     negative
Weighted Avg.  0,762    0,192    0,787      0,762    0,770      0,534    0,861     0,840

=== Confusion Matrix ===

   a   b   c   <-- classified as
 114  39   7 |   a = positive
  78 664 113 |   b = neutral
   7  51 168 |   c = negative
```

APÊNDICE G – RESULTADO BASE “FORD_COM_STOPWORDS_NB” COM ALGORITMO “SMO”

=== Run information ===

Scheme: weka.classifiers.meta.FilteredClassifier -F "weka.filters.unsupervised.attribute.StringToWordVector -R first-last -W 1000 -prune-rate -1.0 -N 0 -stemmer weka.core.stemmers.NullStemmer -stopwords-handler weka.core.stopwords.Null -M 1 -tokenizer \"weka.core.tokenizers.WordTokenizer -delimiters \\\" \\\\r\\\\n\\\\t.,;:\\\\\\'\\\\\\"()?!\\\"\"" -W weka.classifiers.functions.SMO -- -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -E 1.0 -C 250007" -calibrator "weka.classifiers.functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4"
Relation: Ford_Com_StopWords
Instances: 1241
Attributes: 2
HighlightedText
PhraseSentiment
Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

FilteredClassifier using weka.classifiers.functions.SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -E 1.0 -C 250007" -calibrator "weka.classifiers.functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4" on data filtered through weka.filters.unsupervised.attribute.StringToWordVector -R 1 -W 1000 -prune-rate -1.0 -N 0 -stemmer weka.core.stemmers.NullStemmer -stopwords-handler weka.core.stopwords.Null -M 1 -tokenizer "weka.core.tokenizers.WordTokenizer -delimiters \" \\r\\n\\t.,;:\\\'\\\"()?!\""

Filtered Header
@relation 'Ford_Com_StopWords-weka.filters.unsupervised.attribute.StringToWordVector-R1-W1000-prune-rate-1.0-N0-stemmerweka.core.stemmers.NullStemmer-stopwords-handlerweka.core.stopwords.Null-M1-tokenizerweka.core.tokenizers.WordTokenizer -delimiters \" \\r\\n\\t.,;:\\\'\\\"()?!\"'

```
@attribute PhraseSentiment {positive,neutral,negative}
@attribute 0 numeric
@attribute 0km numeric
@attribute 10 numeric
@attribute 15 numeric
@attribute 2 numeric
...
Classifier Model
SMO

Kernel used:
  Linear Kernel: K(x,y) = <x,y>
Classifier for classes: positive, neutral

BinarySMO

Machine linear: showing attribute weights, not support
vectors.
        -0.1467 * (normalized) 0
 +      -0.154  * (normalized) 0km
 +       0.1001 * (normalized) 10
 +      -0.4001 * (normalized) 15
...
Number of kernel evaluations: 409750 (91.301% cached)
```

```
Time taken to build model: 1.28 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1159               93.3924 %
Incorrectly Classified Instances        82                6.6076 %
Kappa statistic                          0.8602
Mean absolute error                      0.2387
Root mean squared error                  0.3007
Relative absolute error                 75.2097 %
Root relative squared error             75.529  %
Total Number of Instances             1241

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area
Class
                 0,869    0,020    0,863      0,869    0,866      0,846    0,957     0,791
positive
                 0,958    0,111    0,950      0,958    0,954      0,851    0,923     0,939
neutral
                 0,889    0,017    0,922      0,889    0,905      0,885    0,955     0,854
negative
Weighted Avg.    0,934    0,082    0,934      0,934    0,934      0,856    0,933     0,905

=== Confusion Matrix ===

   a   b   c   <-- classified as
 139  21   0 |   a = positive
  19 819  17 |   b = neutral
   3  22 201 |   c = negative
```

APÊNDICE H – RESULTADO BASE “FORD_COM_STOPWORDS_NB” COM ALGORITMO “J48”

=== Run information ===

Scheme: weka.classifiers.meta.FilteredClassifier -F "weka.filters.unsupervised.attribute.StringToWordVector -R first-last -W 1000 -prune-rate -1.0 -N 0 -stemmer weka.core.stemmers.NullStemmer -stopwords-handler weka.core.stopwords.Null -M 1 -tokenizer \"weka.core.tokenizers.WordTokenizer -delimiters \\\" \\\\r\\\\n\\\\t.,;:\\\\\\\'\\\\\\\"()?!\\\\\"\"" -W weka.classifiers.trees.J48 -- -C 0.25 -M 2
Relation: Ford_Com_StopWords
Instances: 1241
Attributes: 2
HighlightedText
PhraseSentiment
Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

FilteredClassifier using weka.classifiers.trees.J48 -C 0.25 -M 2 on data filtered through weka.filters.unsupervised.attribute.StringToWordVector -R 1 -W 1000 -prune-rate -1.0 -N 0 -stemmer weka.core.stemmers.NullStemmer -stopwords-handler weka.core.stopwords.Null -M 1 -tokenizer "weka.core.tokenizers.WordTokenizer -delimiters \" \\r\\n\\t.,;:\\\'\\\"()?!\""

Filtered Header
@relation 'Ford_Com_StopWords-weka.filters.unsupervised.attribute.StringToWordVector-R1-W1000-prune-rate-1.0-N0-stemmerweka.core.stemmers.NullStemmer-stopwords-handlerweka.core.stopwords.Null-M1-tokenizerweka.core.tokenizers.WordTokenizer -delimiters \" \\r\\n\\t.,;:\\\'\\\"()?!\"'

```
@attribute PhraseSentiment {positive,neutral,negative}
@attribute 0 numeric
@attribute 0km numeric
@attribute 10 numeric
@attribute 15 numeric
@attribute 2 numeric
...
Classifier Model
J48 pruned tree
------------------

problema <= 0
|   primeiro <= 0
|   |   falta <= 0
|   |   |   brasil <= 0
|   |   |   |   garantia <= 0
|   |   |   |   |   porque <= 0
|   |   |   |   |   |   as <= 0
|   |   |   |   |   |   |   bajulado <= 0
|   |   |   |   |   |   |   |   barulho <= 0
|   |   |   |   |   |   |   |   |   porq <= 0
|   |   |   |   |   |   |   |   |   |   motor <= 0
|   |   |   |   |   |   |   |   |   |   |   novamente <= 0
|   |   |   |   |   |   |   |   |   |   |   |   sei <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   mas <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   automatico <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   comprou <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   ir <= 0
```

```
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   aqui <=
0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   ou
<= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
sem <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   anos <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   so <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   defeito <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   |   por <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   |   |   problemas <= 0
...

Number of Leaves  :     115

Size of the tree :      229


Time taken to build model: 10.44 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        1076               86.7043 %
Incorrectly Classified Instances       165               13.2957 %
Kappa statistic                          0.7082
Mean absolute error                      0.1019
Root mean squared error                  0.2802
Relative absolute error                 32.1073 %
Root relative squared error             70.3674 %
Total Number of Instances             1241

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area
Class
                 0,706    0,038    0,734      0,706    0,720      0,679    0,903     0,679
positive
                 0,938    0,262    0,888      0,938    0,912      0,703    0,893     0,927
neutral
                 0,712    0,023    0,875      0,712    0,785      0,749    0,921     0,815
negative
Weighted Avg.    0,867    0,189    0,866      0,867    0,864      0,709    0,899     0,875

=== Confusion Matrix ===

   a   b   c   <-- classified as
 113  43   4 |   a = positive
  34 802  19 |   b = neutral
   7  58 161 |   c = negative
```

APÊNDICE I – RESULTADO BASE “FORD_SEM_STOPWORDS_SEM_NAO” COM ALGORITMO “*NAIVE BAYES*”

=== Run information ===

Scheme: weka.classifiers.meta.FilteredClassifier -F "weka.filters.unsupervised.attribute.StringToWordVector -R first-last -W 1000 -prune-rate -1.0 -N 0 -stemmer weka.core.stemmers.NullStemmer -stopwords-handler weka.core.stopwords.Null -M 1 -tokenizer \"weka.core.tokenizers.WordTokenizer -delimiters \\\" \\\\r\\\\n\\\\t.,;:\\\\\\\'\\\\\\\"()?!\\\"\"" -W weka.classifiers.bayes.NaiveBayes
Relation: Ford_Sem_StopWords_sem_nao
Instances: 806
Attributes: 2
HighlightedText
PhraseSentiment
Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

FilteredClassifier using weka.classifiers.bayes.NaiveBayes on data filtered through weka.filters.unsupervised.attribute.StringToWordVector -R 1 -W 1000 -prune-rate -1.0 -N 0 -stemmer weka.core.stemmers.NullStemmer -stopwords-handler weka.core.stopwords.Null -M 1 -tokenizer "weka.core.tokenizers.WordTokenizer -delimiters \" \\r\\n\\t.,;:\\\'\\\"()?!\""

Filtered Header
@relation 'Ford_Sem_StopWords_sem_nao-weka.filters.unsupervised.attribute.StringToWordVector-R1-W1000-prune-rate-1.0-N0-stemmerweka.core.stemmers.NullStemmer-stopwords-handlerweka.core.stopwords.Null-M1-tokenizerweka.core.tokenizers.WordTokenizer -delimiters \" \\r\\n\\t.,;:\\\'\\\"()?!\"'

```
@attribute PhraseSentiment {positive,neutral,negative}
@attribute $ numeric
@attribute 0 numeric
@attribute 02059807 numeric
@attribute 02163433 numeric
@attribute 04 numeric
@attribute 10 numeric
...
Classifier Model
Naive Bayes Classifier

                          Class
Attribute              positive  neutral negative
                          (0.2)    (0.5)   (0.31)
=====================================================
…

confianca
  mean                  0.0063     0.01   0.0364
  std. dev.             0.1667   0.1667   0.1874
  weight sum               159      400      247
  precision                  1        1        1

confio
  mean                  0.0063   0.0025        0
  std. dev.             0.1667   0.1667   0.1667
  weight sum               159      400      247
  precision                  1        1        1
```

```
confo
  mean                     0.0063        0        0
  std. dev.                0.1667   0.1667   0.1667
  weight sum                  159      400      247
  precision                     1        1        1
...

Time taken to build model: 0.32 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         563               69.8511 %
Incorrectly Classified Instances       243               30.1489 %
Kappa statistic                          0.51
Mean absolute error                      0.2174
Root mean squared error                  0.4007
Relative absolute error                 52.5075 %
Root relative squared error             88.0865 %
Total Number of Instances              806

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area
Class
                 0,667    0,091    0,642      0,667    0,654      0,568    0,891     0,720
positive
                 0,758    0,305    0,710      0,758    0,733      0,453    0,801     0,766
neutral
                 0,623    0,107    0,720      0,623    0,668      0,539    0,834     0,740
negative
Weighted Avg.    0,699    0,202    0,699      0,699    0,697      0,502    0,829     0,749
```

=== Confusion Matrix ===

```
a   b   c   <-- classified as
 106  41  12 |   a = positive
  49 303  48 |   b = neutral
  10  83 154 |   c = negative
```

APÊNDICE J – RESULT. BASE “FORD_SEM_STOPWORDS_SEM_NAO” COM ALGORITMO “SMO”

=== Run information ===

Scheme: weka.classifiers.meta.FilteredClassifier -F "weka.filters.unsupervised.attribute.StringToWordVector -R first-last -W 1000 -prune-rate -1.0 -N 0 -stemmer weka.core.stemmers.NullStemmer -stopwords-handler weka.core.stopwords.Null -M 1 -tokenizer \"weka.core.tokenizers.WordTokenizer -delimiters \\\" \\\\r\\\\n\\\\t.,;:\\\\\\\'\\\\\\\"()?!\\\"\"" -W weka.classifiers.functions.SMO -- -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -E 1.0 -C 250007" -calibrator "weka.classifiers.functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4"
Relation: Ford_Sem_StopWords_sem_nao
Instances: 806
Attributes: 2
HighlightedText
PhraseSentiment
Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

FilteredClassifier using weka.classifiers.functions.SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -E 1.0 -C 250007" -calibrator "weka.classifiers.functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4" on data filtered through weka.filters.unsupervised.attribute.StringToWordVector -R 1 -W 1000 -prune-rate -1.0 -N 0 -stemmer weka.core.stemmers.NullStemmer -stopwords-handler weka.core.stopwords.Null -M 1 -tokenizer "weka.core.tokenizers.WordTokenizer -delimiters \" \\r\\n\\t.,;:\\\'\\\"()?!\""

Filtered Header
@relation 'Ford_Sem_StopWords_sem_nao-weka.filters.unsupervised.attribute.StringToWordVector-R1-W1000-prune-rate-1.0-N0-stemmerweka.core.stemmers.NullStemmer-stopwords-handlerweka.core.stopwords.Null-M1-tokenizerweka.core.tokenizers.WordTokenizer -delimiters \" \\r\\n\\t.,;:\\\'\\\"()?!\"'

```
@attribute PhraseSentiment {positive,neutral,negative}
@attribute $ numeric
@attribute 0 numeric
...
@data


Classifier Model
SMO

Kernel used:
  Linear Kernel: K(x,y) = <x,y>

Classifier for classes: positive, neutral

BinarySMO

Machine linear: showing attribute weights, not support
vectors.
         0.1404 * (normalized) $
 +      -0.0426 * (normalized) 0
 +      -0.0805 * (normalized) 02059807
 +      -0.128  * (normalized) 02163433
 +      -0.0561 * (normalized) 20
Number of kernel evaluations: 181852 (93.185% cached)
```

```
Time taken to build model: 0.57 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         676               83.871  %
Incorrectly Classified Instances       130               16.129  %
Kappa statistic                          0.739
Mean absolute error                      0.2639
Root mean squared error                  0.3396
Relative absolute error                 63.7255 %
Root relative squared error             74.6519 %
Total Number of Instances              806

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area
Class
                 0,792    0,046    0,808      0,792    0,800      0,751    0,931     0,733
positive
                 0,858    0,167    0,835      0,858    0,846      0,690    0,845     0,790
neutral
                 0,838    0,057    0,866      0,838    0,852      0,788    0,921     0,801
negative
Weighted Avg.    0,839    0,110    0,839      0,839    0,839      0,732    0,885     0,782

=== Confusion Matrix ===

   a   b   c   <-- classified as
 126  32   1 |   a = positive
  26 343  31 |   b = neutral
   4  36 207 |   c = negative
```

APÊNDICE K – RESULT. BASE “FORD_SEM_STOPWORDS_SEM_NAO” COM ALGORITMO “J48”

=== Run information ===

Scheme: weka.classifiers.meta.FilteredClassifier -F "weka.filters.unsupervised.attribute.StringToWordVector -R first-last -W 1000 -prune-rate -1.0 -N 0 -stemmer weka.core.stemmers.NullStemmer -stopwords-handler weka.core.stopwords.Null -M 1 -tokenizer \"weka.core.tokenizers.WordTokenizer -delimiters \\\" \\\\r\\\\n\\\\t.,;:\\\\\\\'\\\\\\\"()?!\\\"\"" -W weka.classifiers.trees.J48 -- -C 0.25 -M 2
Relation: Ford_Sem_StopWords_sem_nao
Instances: 806
Attributes: 2
HighlightedText
PhraseSentiment
Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

FilteredClassifier using weka.classifiers.trees.J48 -C 0.25 -M 2 on data filtered through weka.filters.unsupervised.attribute.StringToWordVector -R 1 -W 1000 -prune-rate -1.0 -N 0 -stemmer weka.core.stemmers.NullStemmer -stopwords-handler weka.core.stopwords.Null -M 1 -tokenizer "weka.core.tokenizers.WordTokenizer -delimiters \" \\r\\n\\t.,;:\\\'\\\"()?!\""

Filtered Header
@relation 'Ford_Sem_StopWords_sem_nao-weka.filters.unsupervised.attribute.StringToWordVector-R1-W1000-prune-rate-1.0-N0-stemmerweka.core.stemmers.NullStemmer-stopwords-handlerweka.core.stopwords.Null-M1-tokenizerweka.core.tokenizers.WordTokenizer -delimiters \" \\r\\n\\t.,;:\\\'\\\"()?!\"'

```
@attribute PhraseSentiment {positive,neutral,negative}
@attribute $ numeric
@attribute 0 numeric
@attribute 02059807 numeric
@attribute 02163433 numeric
@attribute 04 numeric
@attribute 10 numeric
@attribute 15 numeric
...
Classifier Model
J48 pruned tree
------------------

peca <= 0
|   solucao <= 0
|   |   levei <= 0
|   |   |   grande <= 0
|   |   |   |   problema <= 0
|   |   |   |   |   acidentes <= 0
|   |   |   |   |   |   resolver <= 0
|   |   |   |   |   |   |   otimo <= 0
|   |   |   |   |   |   |   |   veiculo <= 0
|   |   |   |   |   |   |   |   |   problemas <= 0
|   |   |   |   |   |   |   |   |   |   defeito <= 0
|   |   |   |   |   |   |   |   |   |   |   minivan <= 0
|   |   |   |   |   |   |   |   |   |   |   |   protetor <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   embreagem <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   totalmente <= 0
```

```
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   graca <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   comprei <=
0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   duas <=
0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
resposta <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
lindo <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   trocar <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   o <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   compre <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   |   ano <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   |   |   disseram <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   inovacao <= 0
…

Number of Leaves  :     110

Size of the tree :      219


Time taken to build model: 5.95 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         628               77.9156 %
Incorrectly Classified Instances       178               22.0844 %
Kappa statistic                          0.6421
Mean absolute error                      0.1727
Root mean squared error                  0.348
Relative absolute error                 41.7194 %
Root relative squared error             76.4842 %
Total Number of Instances              806

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area
Class
                 0,673    0,071    0,699      0,673    0,686      0,611    0,885     0,702
positive
                 0,810    0,219    0,785      0,810    0,797      0,591    0,857     0,843
neutral
                 0,798    0,077    0,821      0,798    0,809      0,727    0,902     0,824
negative
Weighted Avg.    0,779    0,146    0,779      0,779    0,779      0,636    0,876     0,810

=== Confusion Matrix ===

   a   b   c   <-- classified as
 107  44   8 |   a = positive
  41 324  35 |   b = neutral
   5  45 197 |   c = negative
```

APÊNDICE L – RES. BASE "FORD_SEM_STOPWORDS_COM_NAO_NB" COM ALGORITMO "*NAIVE BAYES*"

=== Run information ===

Scheme: weka.classifiers.meta.FilteredClassifier -F "weka.filters.unsupervised.attribute.StringToWordVector -R first-last -W 1000 -prune-rate -1.0 -N 0 -stemmer weka.core.stemmers.NullStemmer -stopwords-handler weka.core.stopwords.Null -M 1 -tokenizer \"weka.core.tokenizers.WordTokenizer -delimiters \\\" \\\\r\\\\n\\\\t.,;:\\\\\\\'\\\\\\\"()?!\\\\\"\"" -W weka.classifiers.bayes.NaiveBayes
Relation: Ford_Sem_StopWords_com_nao
Instances: 843
Attributes: 2
HighlightedText
PhraseSentiment
Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

FilteredClassifier using weka.classifiers.bayes.NaiveBayes on data filtered through weka.filters.unsupervised.attribute.StringToWordVector -R 1 -W 1000 -prune-rate -1.0 -N 0 -stemmer weka.core.stemmers.NullStemmer -stopwords-handler weka.core.stopwords.Null -M 1 -tokenizer "weka.core.tokenizers.WordTokenizer -delimiters \" \\r\\n\\t.,;:\\\'\\\"()?!\""

Filtered Header
@relation 'Ford_Sem_StopWords_com_nao-weka.filters.unsupervised.attribute.StringToWordVector-R1-W1000-prune-rate-1.0-N0-stemmerweka.core.stemmers.NullStemmer-stopwords-handlerweka.core.stopwords.Null-M1-tokenizerweka.core.tokenizers.WordTokenizer -delimiters \" \\r\\n\\t.,;:\\\'\\\"()?!\"'

@attribute PhraseSentiment {positive,neutral,negative}
@attribute $ numeric
@attribute 0 numeric
@attribute 02059807 numeric
@attribute 02163433 numeric
@attribute 04 numeric
@attribute 10 numeric
@attribute 15 numeric
...

```
Classifier Model
Naive Bayes Classifier

                          Class
Attribute              positive  neutral negative
                         (0.19)   (0.57)   (0.24)
=================================================
$
  mean                   0.0123   0.0125   0.0149
  std. dev.              0.1667   0.1667   0.1667
  weight sum                163      479      201
  precision                   1        1        1

0
  mean                   0.0123   0.0021        0
  std. dev.              0.1667   0.1667   0.1667
  weight sum                163      479      201
  precision                   1        1        1
```

```
02059807
  mean                     0.0061   0.0021    0.005
  std. dev.                0.1667   0.1667   0.1667
  weight sum                  163      479      201
  precision                     1        1        1
...

Time taken to build model: 0.34 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         578               68.5647 %
Incorrectly Classified Instances       265               31.4353 %
Kappa statistic                          0.4764
Mean absolute error                      0.2255
Root mean squared error                  0.4006
Relative absolute error                 57.9991 %
Root relative squared error             90.8704 %
Total Number of Instances              843

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area
Class
               0,699    0,100    0,626      0,699    0,661      0,575    0,900     0,734
positive
               0,704    0,275    0,771      0,704    0,736      0,425    0,795     0,819
neutral
               0,632    0,151    0,567      0,632    0,598      0,464    0,812     0,666
negative
Weighted Avg.  0,686    0,211    0,694      0,686    0,688      0,463    0,819     0,766

=== Confusion Matrix ===

   a   b   c   <-- classified as
 114  34  15 |   a = positive
  60 337  82 |   b = neutral
   8  66 127 |   c = negative
```

APÊNDICE M – RES. BASE “FORD_SEM_STOPWORDS_COM_NAO_NB” COM ALGORITMO “SMO”

=== Run information ===

Scheme: weka.classifiers.meta.FilteredClassifier -F "weka.filters.unsupervised.attribute.StringToWordVector -R first-last -W 1000 -prune-rate -1.0 -N 0 -stemmer weka.core.stemmers.NullStemmer -stopwords-handler weka.core.stopwords.Null -M 1 -tokenizer \"weka.core.tokenizers.WordTokenizer -delimiters \\\" \\\\r\\\\n\\\\t.,;:\\\\\\'\\\\\\"()?!\\\"\"" -W weka.classifiers.functions.SMO -- -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -E 1.0 -C 250007" -calibrator "weka.classifiers.functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4"
Relation: Ford_Sem_StopWords_com_nao
Instances: 843
Attributes: 2
HighlightedText
PhraseSentiment
Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

FilteredClassifier using weka.classifiers.functions.SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.PolyKernel -E 1.0 -C 250007" -calibrator "weka.classifiers.functions.Logistic -R 1.0E-8 -M -1 -num-decimal-places 4" on data filtered through weka.filters.unsupervised.attribute.StringToWordVector -R 1 -W 1000 -prune-rate -1.0 -N 0 -stemmer weka.core.stemmers.NullStemmer -stopwords-handler weka.core.stopwords.Null -M 1 -tokenizer "weka.core.tokenizers.WordTokenizer -delimiters \" \\r\\n\\t.,;:\\\'\\\"()?!\""

Filtered Header
@relation 'Ford_Sem_StopWords_com_nao-weka.filters.unsupervised.attribute.StringToWordVector-R1-W1000-prune-rate-1.0-N0-stemmerweka.core.stemmers.NullStemmer-stopwords-handlerweka.core.stopwords.Null-M1-tokenizerweka.core.tokenizers.WordTokenizer -delimiters \" \\r\\n\\t.,;:\\\'\\\"()?!\"'

```
@attribute PhraseSentiment {positive,neutral,negative}
@attribute $ numeric
@attribute 0 numeric
@attribute 02059807 numeric
...

Classifier Model
SMO

Kernel used:
  Linear Kernel: K(x,y) = <x,y>

Classifier for classes: positive, neutral

BinarySMO

Machine linear: showing attribute weights, not support
vectors.
 +      -0.0665 * (normalized) abraco
 +      -0.0542 * (normalized) abs
 +      -0.1747 * (normalized) acabamento
 +      -0.3574 * (normalized) acabou
…
Number of kernel evaluations: 187670 (93.299% cached)
```

```
Time taken to build model: 0.69 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         736               87.3072 %
Incorrectly Classified Instances       107               12.6928 %
Kappa statistic                          0.7815
Mean absolute error                      0.2546
Root mean squared error                  0.3258
Relative absolute error                 65.4913 %
Root relative squared error             73.9112 %
Total Number of Instances              843

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area
Class
                 0,840    0,038    0,840      0,840    0,840      0,802    0,944     0,774
positive
                 0,898    0,148    0,888      0,898    0,893      0,751    0,877     0,857
neutral
                 0,841    0,042    0,862      0,841    0,851      0,806    0,919     0,781
negative
Weighted Avg.    0,873    0,102    0,873      0,873    0,873      0,774    0,900     0,823

=== Confusion Matrix ===

   a   b   c   <-- classified as
 137  26   0 |   a = positive
  22 430  27 |   b = neutral
   4  28 169 |   c = negative
```

APÊNDICE N – RES. BASE “FORD_SEM_STOPWORDS_COM_NAO_NB” COM ALGORITMO “J48”

=== Run information ===

Scheme: weka.classifiers.meta.FilteredClassifier -F "weka.filters.unsupervised.attribute.StringToWordVector -R first-last -W 1000 -prune-rate -1.0 -N 0 -stemmer weka.core.stemmers.NullStemmer -stopwords-handler weka.core.stopwords.Null -M 1 -tokenizer \"weka.core.tokenizers.WordTokenizer -delimiters \\\" \\\\r\\\\n\\\\t.,;:\\\\\\\'\\\\\\\"()?!\\\\\"\"" -W weka.classifiers.trees.J48 -- -C 0.25 -M 2
Relation: Ford_Sem_StopWords_com_nao
Instances: 843
Attributes: 2
HighlightedText
PhraseSentiment
Test mode: 10-fold cross-validation

=== Classifier model (full training set) ===

FilteredClassifier using weka.classifiers.trees.J48 -C 0.25 -M 2 on data filtered through weka.filters.unsupervised.attribute.StringToWordVector -R 1 -W 1000 -prune-rate -1.0 -N 0 -stemmer weka.core.stemmers.NullStemmer -stopwords-handler weka.core.stopwords.Null -M 1 -tokenizer "weka.core.tokenizers.WordTokenizer -delimiters \" \\r\\n\\t.,;:\\\'\\\"()?!\""

Filtered Header
@relation 'Ford_Sem_StopWords_com_nao-weka.filters.unsupervised.attribute.StringToWordVector-R1-W1000-prune-rate-1.0-N0-stemmerweka.core.stemmers.NullStemmer-stopwords-handlerweka.core.stopwords.Null-M1-tokenizerweka.core.tokenizers.WordTokenizer -delimiters \" \\r\\n\\t.,;:\\\'\\\"()?!\"'

```
@attribute PhraseSentiment {positive,neutral,negative}
@attribute $ numeric
@attribute 0 numeric
@attribute 02059807 numeric
@attribute 02163433 numeric
@attribute 04 numeric
@attribute 10 numeric
@attribute 15 numeric
...

Classifier Model
J48 pruned tree
------------------

cambio <= 0
|   problema <= 0
|   |   levei <= 0
|   |   |   caso <= 0
|   |   |   |   embreagem <= 0
|   |   |   |   |   defeito <= 0
|   |   |   |   |   |   protetor <= 0
|   |   |   |   |   |   |   barulho <= 0
|   |   |   |   |   |   |   |   falta <= 0
|   |   |   |   |   |   |   |   |   otimo <= 0
|   |   |   |   |   |   |   |   |   |   resposta <= 0
|   |   |   |   |   |   |   |   |   |   |   comprei <= 0
|   |   |   |   |   |   |   |   |   |   |   |   pos <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   grande <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   solucao <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   minivan <= 0
```

```
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   motor <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   graca <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   duas <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
conforto <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
atendimento <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
nenhum <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   comprar <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   desgaste <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   fusion <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   |   brasil <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   |   |   marca <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   fabrica <= 0
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   lindo <= 0
…

Number of Leaves  :     117

Size of the tree :      233


Time taken to build model: 6.48 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         646               76.6311 %
Incorrectly Classified Instances       197               23.3689 %
Kappa statistic                          0.5852
Mean absolute error                      0.1748
Root mean squared error                  0.351
Relative absolute error                 44.9646 %
Root relative squared error             79.619  %
Total Number of Instances              843

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area
Class
                 0,656    0,062    0,718      0,656    0,686      0,616    0,915     0,762
positive
                 0,854    0,324    0,776      0,854    0,813      0,542    0,848     0,865
neutral
                 0,647    0,058    0,778      0,647    0,707      0,630    0,889     0,755
negative
Weighted Avg.    0,766    0,210    0,765      0,766    0,763      0,577    0,871     0,818

=== Confusion Matrix ===

   a   b   c   <-- classified as
 107  54   2 |   a = positive
  35 409  35 |   b = neutral
   7  64 130 |   c = negative
```

ANEXO A – CÓDIGO DE PRÉ-PROCESSAMENTO EM PYTHON COM RETIRADA DE *STOPWORDS* MANTENDO-SE A PALAVRA “NAO”

```
from unicodedata import normalize
import nltk
nltk.download('stopwords')
nltk.download('punkt')
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

arquivo = open('processar.txt', 'r')
resultado = open('resultado.txt', 'w')

for linha in arquivo:
    # ----Transformando caracteres em letras MINUSCULAS----#
      linha = linha.lower()

    # ----Transformando ABREVIAÇÕES----#
      i = 0
      abv = ['blz', 'flw', 'vlw', ' ta ', ' mt ', ' q ', ' n ', ' pq ', '
ok ',
      ' vcs ', ' vc ', ' amr ', ' migo ', 'hj', 'tmb', ' br ']
      abv2 = ['beleza', 'tchau', 'obrigado', ' esta ', ' muito ', ' que ',
' nao ', ' porque ', ' entendi ',
      ' voces ', ' voce ', ' amor ', ' amigo ', 'hoje', 'tambem', ' brasil
']
      while i < len(abv):
         linha = linha.replace(abv[i], abv2[i])
         i = i + 1

   # ----Transformando EMOTICONS em palavras----#
      i = 0
      emt = ['(:', ':)', '=)', '(=', ':D', 'D:', ';)', '(;', ' xd ', ':O',
      ':P', '<2', '<3', '><', ' s2 ', ' sz ', 'u.u',
      ':@', ':/', ":'(", ':9', ':x', '*-*']
      emtC = ['sorrindo', 'sorrindo', 'sorrindo', 'sorrindo', 'sorrindo',
      'surpreso', 'piscando', 'piscando', 'sorrindo',
      'surpreso', 'lingua de fora', 'amor', 'amor', 'gostei', 'amor',
      'amor', 'prevalecido', 'bravo', 'indeciso',
      'chorando', 'gostando', 'aborrecido', 'gostando']
      while i < len(emt):
         linha = linha.replace(emt[i], emtC[i])
         i = i + 1

   # ----Removendo ACENTUAÇÃO ----#
      linha = normalize('NFKD',
      linha).encode('ASCII','ignore').decode('ASCII')

   # ---- STOPWORDS ----#
      stop_words = set(stopwords.words("portuguese"))
      stop_words.update(['voces', 'ficam', 'tirar', 'sobre', 'quer',
      'querem', 'vou', 'vamos', 'ir', 'gente', 'fazer', 'cada', 'acho',
'pode',
      'cara', 'bem', 'pois', 'voce', 'ninguem', 'ainda', 'mae', 'deve',
'estado', 'pai',
      'filhos', 'filho','porque', 'pais', 'anos', 'casa',
      'pessoas','nessa', 'algum', 'algumas', 'nesse', 'aqui', 'coisa',
'seria',
      'pros', 'poxa', 'ser', 'assim', 'dar', 'fez', 'quiser', 'posso',
```

```
'todo',
      'toda', 'nada', 'todos', 'etc', 'ter', 'la', 'ate', 'faz',
      'ficar','ai', 'vai', 'pega', 'vao', 'e', 'pra', 'sim', 'ta', 'to',
'vi', 'vem',
      'pro', 'tambem', 'hoje', 'pede', 'agora', 'dia', 'outro', 'meses',
'km', 'deu',
      '2', 'ja', 'so', 'dias', 'outros', 'uso', '15', 'tudo', 'mim', 'ha',
'estao', 'faco', 'ver'])

   ##descomente a linha abaixo para ver a lista de stopwords##
   # print(stop words)
      words = word_tokenize(linha)
    ##descomente a linha abaixo para ver os tokens criados##
    # print(words)
      linha_limpa = [w for w in words if not w in stop_words]
      linha = str(''.join(str(e + ' ') for e in linha_limpa))

    # ----Removendo CARACTERES ESPECIAIS E NÚMEROS----#
      i = 0
      caracteres = [', ', '. ', '? ', '! ', ',', '.', '?', '!', '`', "'",
'"', '@', '*', '#', '%', '¨', '-',
      '_', '+', '-', '=', ')', '(', '[', ']', '.', '&', ':', '>', '<', '/
', '/', '$', '0', '1', '2', '3', '4',
      '5' ,'6', '7', '8', '9']
      caracteres2 = [' ', ' ', ' ', ' ', '', '', '', '', '', '', '', '',
'', '', '', '', '',
      '', '', '', '', '', '', '', '', '', '', '', '', '', ' ', '', '', '',
'', '', '', '',
      '', '', '', '', '']
      while i < len(caracteres):
         linha = linha.replace(caracteres[i], caracteres2[i])
         i = i + 1

    #---- ESCREVE linha processada em resultado.txt ----#
      resultado.write(linha+'\n')

arquivo.close()
resultado.close()
```

ANEXO B – CÓDIGO DE PRÉ-PROCESSAMENTO EM PYTHON COM RETIRADA DE *STOPWORDS* EXCLUINDO A PALAVRA “NAO”

```
from unicodedata import normalize
import nltk
nltk.download('stopwords')
nltk.download('punkt')
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

arquivo = open('processar.txt', 'r')
resultado = open('resultado.txt', 'w')

for linha in arquivo:
    # ----Transformando caracteres em letras MINUSCULAS----#
      linha = linha.lower()

    # ----Transformando ABREVIAÇÕES----#
      i = 0
      abv = ['blz', 'flw', 'vlw', ' ta ', ' mt ', ' q ', ' n ', ' pq ', '
ok ',
      ' vcs ', ' vc ', ' amr ', ' migo ', 'hj', 'tmb', ' br ']
      abv2 = ['beleza', 'tchau', 'obrigado', ' esta ', ' muito ', ' que ',
' nao ', ' porque ', ' entendi ',
      ' voces ', ' voce ', ' amor ', ' amigo ', 'hoje', 'tambem', ' brasil
']
      while i < len(abv):
         linha = linha.replace(abv[i], abv2[i])
         i = i + 1

   # ----Transformando EMOTICONS em palavras----#
      i = 0
      emt = ['(:', ':)', '=)', '(=', ':D', 'D:', ';)', '(;', ' xd ', ':O',
      ':P', '<2', '<3', '><', ' s2 ', ' sz ', 'u.u',
      ':@', ':/', ":'(", ':9', ':x', '*-*']
      emtC = ['sorrindo', 'sorrindo', 'sorrindo', 'sorrindo', 'sorrindo',
      'surpreso', 'piscando', 'piscando', 'sorrindo',
      'surpreso', 'lingua de fora', 'amor', 'amor', 'gostei', 'amor',
      'amor', 'prevalecido', 'bravo', 'indeciso',
      'chorando', 'gostando', 'aborrecido', 'gostando']
      while i < len(emt):
         linha = linha.replace(emt[i], emtC[i])
         i = i + 1

   # ----Removendo ACENTUAÇÃO ----#
      linha = normalize('NFKD',
      linha).encode('ASCII','ignore').decode('ASCII')

   # ---- STOPWORDS ----#
      stop_words = set(stopwords.words("portuguese"))
      stop_words.update(['nao', 'voces', 'ficam', 'tirar', 'sobre', 'quer',
      'querem', 'vou', 'vamos', 'ir', 'gente', 'fazer', 'cada', 'acho',
'pode',
      'cara', 'bem', 'pois', 'voce', 'ninguem', 'ainda', 'mae', 'deve',
'estado', 'pai',
      'filhos', 'filho','porque', 'pais', 'anos', 'casa',
      'pessoas','nessa', 'algum', 'algumas', 'nesse', 'aqui', 'coisa',
'seria',
      'pros', 'poxa', 'ser', 'assim', 'dar', 'fez', 'quiser', 'posso',
```

```
'todo',
      'toda', 'nada', 'todos', 'etc', 'ter', 'la', 'ate', 'faz',
      'ficar','ai', 'vai', 'pega', 'vao', 'e', 'pra', 'sim', 'ta', 'to',
'vi', 'vem',
      'pro', 'tambem', 'hoje', 'pede', 'agora', 'dia', 'outro', 'meses',
'km', 'deu',
      '2', 'ja', 'so', 'dias', 'outros', 'uso', '15', 'tudo', 'mim', 'ha',
'estao', 'faco', 'ver'])

   ##descomente a linha abaixo para ver a lista de stopwords##
   # print(stop words)
      words = word_tokenize(linha)
    ##descomente a linha abaixo para ver os tokens criados##
    # print(words)
      linha_limpa = [w for w in words if not w in stop_words]
      linha = str(''.join(str(e + ' ') for e in linha_limpa))

    # ----Removendo CARACTERES ESPECIAIS E NÚMEROS----#
      i = 0
      caracteres = [', ', '. ', '? ', '! ', ',', '.', '?', '!', '`', "'",
'"', '@', '*', '#', '%', '¨', '-',
      '_', '+', '-', '=', ')', '(', '[', ']', '.', '&', ':', '>', '<', '/
', '/', '$', '0', '1', '2', '3', '4',
      '5' ,'6', '7', '8', '9']
      caracteres2 = [' ', ' ', ' ', ' ', '', '', '', '', '', '', '', '',
'', '', '', '', '',
      '', '', '', '', '', '', '', '', '', '', '', '', '', ' ', '', '', '',
'', '', '', '',
      '', '', '', '', '']
      while i < len(caracteres):
         linha = linha.replace(caracteres[i], caracteres2[i])
         i = i + 1

    #---- ESCREVE linha processada em resultado.txt ----#
      resultado.write(linha+'\n')

arquivo.close()
resultado.close()
```

ANEXO C – CÓDIGO DE PRÉ-PROCESSAMENTO EM PYTHON SEM RETIRADA DE *STOPWORDS*

```
from unicodedata import normalize
import nltk
nltk.download('stopwords')
nltk.download('punkt')
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

arquivo = open('processar.txt', 'r')
resultado = open('resultado.txt', 'w')

for linha in arquivo:
    # ----Transformando caracteres em letras MINUSCULAS----#
      linha = linha.lower()

    # ----Transformando ABREVIAÇÕES----#
      i = 0
      abv = ['blz', 'flw', 'vlw', ' ta ', ' mt ', ' q ', ' n ', ' pq ', '
ok ',
      ' vcs ', ' vc ', ' amr ', ' migo ', 'hj', 'tmb', ' br ']
      abv2 = ['beleza', 'tchau', 'obrigado', ' esta ', ' muito ', ' que ',
' nao ', ' porque ', ' entendi ',
      ' voces ', ' voce ', ' amor ', ' amigo ', 'hoje', 'tambem', ' brasil
']
      while i < len(abv):
         linha = linha.replace(abv[i], abv2[i])
         i = i + 1

   # ----Transformando EMOTICONS em palavras----#
      i = 0
      emt = ['(:', ':)', '=)', '(=', ':D', 'D:', ';)', '(;', ' xd ', ':O',
      ':P', '<2', '<3', '><', ' s2 ', ' sz ', 'u.u',
      ':@', ':/', ":'(", ':9', ':x', '*-*']
      emtC = ['sorrindo', 'sorrindo', 'sorrindo', 'sorrindo', 'sorrindo',
      'surpreso', 'piscando', 'piscando', 'sorrindo',
      'surpreso', 'lingua de fora', 'amor', 'amor', 'gostei', 'amor',
      'amor', 'prevalecido', 'bravo', 'indeciso',
      'chorando', 'gostando', 'aborrecido', 'gostando']
      while i < len(emt):
         linha = linha.replace(emt[i], emtC[i])
         i = i + 1

    # ----Removendo ACENTUAÇÃO ----#
      linha = normalize('NFKD',
      linha).encode('ASCII','ignore').decode('ASCII')

    # ----Removendo CARACTERES ESPECIAIS E NÚMEROS----#
      i = 0
      caracteres = [', ', '. ', '? ', '! ', ',', '.', '?', '!', '`', "'",
'"', '@', '*', '#', '%', '¨', '-',
      '_', '+', '-', '=', ')', '(', '[', ']', '.', '&', ':', '>', '<', '/
', '/', '$', '0', '1', '2', '3', '4',
      '5' ,'6', '7', '8', '9']
      caracteres2 = [' ', ' ', ' ', ' ', '', '', '', '', '', '', '', '',
'', '', '', '', '',
      '', '', '', '', '', '', '', '', '', '', '', '', '', ' ', '', '', '',
'', '', '', '',
```

```
        '', '', '', '', '']
        while i < len(caracteres):
            linha = linha.replace(caracteres[i], caracteres2[i])
            i = i + 1

    #---- ESCREVE linha processada em resultado.txt ----#
        resultado.write(linha)

arquivo.close()
resultado.close()
```

ANEXO D – CÓDIGO EM PYTHON DE FREQUÊNCIA DE PALAVRAS

```
from collections import Counter
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import *

ocurrences = []

with open('resultado.txt') as f:
    occurrences = Counter(f.read().split()).most_common(50)

label = [i[0] for i in occurrences]
value = [i[1] for i in occurrences]

x_axis = label
y_axis = value

ind = np.arange(len(x_axis))
#print(ind)

my_dpi = 100
plt.figure(figsize=(800/my_dpi, 500/my_dpi), dpi=my_dpi)
plt.bar(ind, y_axis, color='Blue', )
plt.xticks(ind, x_axis, rotation='75', size='small', color = 'navy')
plt.yticks(color='navy', size='small')
plt.subplots_adjust(bottom=0.2) #ajusta parte de baixo do gráfico na tela
plt.tick_params(width=1) #traço nos labels
plt.title("Ocorrência de Palavras", color='navy')
plt.xlabel('Palavras')
plt.axhline(30, color="gray", linestyle='--', marker='s', linewidth=0.5)
plt.axhline(25, color="gray", linestyle='--', marker='s', linewidth=0.5)
plt.axhline(15, color="gray", linestyle='--', marker='s', linewidth=0.5)
plt.axhline(20, color="gray", linestyle='--', marker='s', linewidth=0.5)
plt.axhline(10, color="gray", linestyle='--', marker='s', linewidth=0.5)
plt.axhline(5, color="gray", linestyle='--', marker='s', linewidth=0.5)
plt.ylabel('Frequência')
plt.show()
```