UNIVERSIDADE FEDERAL DO ABC

TRABALHO DE GRADUAÇÃO EM

ENGENHARIA DE INFORMAÇÃO

Estudo sobre avaliações de produtos de *e-commerce* com uso de ferramentas de processamento de língua natural e mineração de dados

Gabriel Sena de Queiroz

Santo André, SP

2021

Gabriel Sena de Queiroz

Estudo sobre avaliações de produtos de *e-commerce* com uso de ferramentas de processamento de língua natural e mineração de dados

Monografia apresentada ao curso de Engenharia de Informação da Universidade Federal do ABC como parte dos requisitos para a obtenção do grau de Engenheiro de Informação.

UNIVERSIDADE FEDERAL DO ABC

Orientador:
Prof. Dr. André Kazuo Takahata

Santo André, SP

2021

Sumário

Resumo

Este projeto foi desenvolvido como trabalho de graduação do curso de engenharia de informação e tem como objetivo verificar hipóteses relacionadas à capacidade de predição de satisfação de clientes através de algoritmos de aprendizado de máquina e analisar dados relacionados a avaliações de consumidores no *e-commerce* do site americanas com utilizando ferramentas de processamento de língua natural e mineração de dados. Para tal foi utilizada uma base de dados de domínio do grupo B2W disponibilizada publicamente, na qual implementamos algoritmo baseado em árvore de decisão e o classificador Naive Bayes Multinomial para realizar predições de satisfação dos consumidores em determinados produtos. Observamos uma acurácia de 84% no algoritmo que utiliza os dados de *reviews* dos clientes. Já no algoritmo baseado em árvore de decisão, utilizando os dados do cliente e do produto, os resultados apontaram para uma não correlação entre tais dados e a satisfação dos clientes.

Palavras-chave: Ciência de Dados; Satisfação; Naive Bayes; Classificador de Textos.

CAPÍTULO 1

Introdução

1.1 Motivação

A partir do trabalho e dos dados disponibilizados em [Real et al., 2019] nos propomos a aplicar ferramentas de Processamento de informação em língua natural e análise de sentimento para entender melhor o comportamento de consumidores em compras de *marketplaces* e a aplicação de ferramentas de tecnologia de informação para análise e predição de satisfação dos clientes. Tais ferramentas são importantes pois nos permitem realizar predições sobre o comportamento dos consumidores a partir de dados pré-processados, permitindo a análise das tendências comportamentais dos consumidores através dos dados.

1.2 Objetivos

Para entender a efetividade das ferramentas que serão utilizadas e as soluções buscadas para a classificação das compras, iremos estabelecer algumas hipóteses e um plano de verificação para cada uma.

Hipótese 1: .
"Obtemos uma acurácia semelhante ao tentar predizer a satisfação de uma compra processando o *review* do cliente com ferramentas de PLN (Processamento de Língua Natural) e processando as informações de dados do cliente e do produto através do algoritmo *Extra Trees Classifier*."

Hipótese 2: .

"Pode-se predizer a satisfação do cliente através de ferramentas de PLN e Análise de Sentimento com relevante acurácia (>80%)"

Decidimos utilizar tal valor mencionado de modo subjetivo. Em trabalhos futuros, porém, pode-se analisar com mais profundidade tal valor.
Para a verificação das hipóteses apresentadas têm-se que utilizar algumas estratégias de classificação das compras realizadas. Cada compra realizada no site que compos o Dataset utilizado apresenta a possibilidade de classificação da compra realizada com uma avaliação de 1 a 5 estrelas, de modo que quanto maior este número, maior a satisfação do cliente com a compra realizada. Para uma melhor compreenção do sentimento do cliente em relação à compra, uma estratégia apresentada já no trabalho de origem [Real et al., 2019] é reclassificar as estrelas em 3 novas classes: Negativo, Neutro e Positivo.

A partir dessas classificações, podemos treinar os modelos, que serão apresentados nos capíutlos a seguir, e tentar predizer as classes das avaliações de compras.

CAPÍTULO 2

Ferramentas e Base de Dados

2.1 Ferramentas

Iremos utilizar algumas ferramentas de processamento de informação para pré-processar, processar e analisar os dados. A principal ferramenta a ser utilizada será a linguagem Python e a partir dessa, algumas outras bibliotecas e APIs nos ajudarão a realizar os processamentos e análises. Algumas ferramentas nos permitem analisar informações em língua natural, quais sejam os *reviews* de compras presentes na base. A seguir apresentamos as principais ferramentas utilizadas:

Biblioteca NLTK - *Natural Language Tool Kit*

A biblioteca contêm uma grande quantidade de dados, códigos e documentação sobre processamento de informação em língua natural. Algumas ferramentas nos ajudarão na fase de pré-processamento dos dados de *review*:

- Tokenização: Quebra da sequência de caracteres de um texto localizando as delimitações de cada palavra ou símbolo [Palmer, 2010].

- Remoção de Stopwords: Remoção de palavras ou símbolos com pouco conteúdo semântico considerados irrelevantes para o modelo preditivo. [BARBOSA, et al., 2017]

Naive Bayes Multinomial

O algoritmo Naive Bayes é um classificador probabilístico que se baseia no Teorema de Bayes, criado por Thomas Bayes (1701 - 1761). Atualmente é amplamente utilizado em

projetos de aprendizado de máquina. [RENNIE, et al., 2003]

O Algoritmo parte da formula de probabilidade condicional, do teorema de Bayes, onde calcula-se a probabilidade de uma classe c_i dado um documento d:

$$P(c_i \mid d) = \frac{P(c\,e\,d_i)}{p(d)}$$

A partir de então, se busca obter estimativas da probabilidade de c_i dado d, para N features, de modo a retornar a classe que maximiza a função de probabilidade:

$$\hat{c} = \underset{c_i}{\operatorname{argmax}} P(c_i \mid d), \text{ i=1, 2, 3, ..., N}$$

Reescrevendo, temos pelo teorema de Bayes que:

$$\hat{c} = \underset{c_i}{\operatorname{argmax}} P(c_i \mid d) = \underset{c_i}{\operatorname{argmax}} \frac{P(c_i)P(d|c_i)}{P(d)} .$$

Porém, sabe-se que *P(d)* não se altera com c_i, portanto:

$$\hat{c} = \underset{c_i}{\operatorname{argmax}} P(c_i)\, P(d \mid c_i) ,$$

Assim, ao representar d como um conjunto de características f_1,f_2, ..., f_m, temos:

$$\hat{c} = \underset{c_i}{\operatorname{argmax}} P(c_i)\, P\left(f_1, f_2, \ldots, f_M \mid c_i\right) ,$$

O algoritmo naive Bayes utiliza a hipótese simplificadora de que as características são condicionalmente independentes

$$\hat{c} = \underset{c_i}{\operatorname{argmax}} P(c_i)\, P\left(f_1 \mid c_i\right) P\left(f_2 \mid c_i\right) \ldots P\left(f_M \mid c_i\right)$$

Neste projeto, trabalharemos com 3 classes: "Positivo", "Negativo"e "Neutro". Em particular, utilizamos uma variação do Naive Bayes denominada Naive Bayes Multinomial, em que a distribuição das palavras de um texto é modelada pela distribuição multinomial (Rennie et al., 2013). A implementação utilizada foi a disponibilizada pela biblioteca scikit-learn.

ExtraTreesClassifier

É um algoritmo de machine learning presente na biblioteca scikit learn que utiliza árvores de decisão aleatórias para a criação de um modelo preditivo a partir dos dados relacionados à um determinado grupo de classes [GEURTS et al., 2006]. Tal modelo foi escolhido para o projeto por sua boa performance na aplicação proposta (atributos com grande variabilidade).

Árvores de decisão são modelos utilizados em inferência indutiva. Estas árvores são treinadas de acordo com um conjunto de treino (exemplos previamente classificados) e posteriormente, outros exemplos são classificados de acordo com essa mesma árvore. Observe na figura abaixo a ilustração de uma árvore de decisão com valores fictícios, onde se têm uma árvore treinada para clientes e suas probabilidades de pagar um empréstimo de acordo com dados cadastrais:

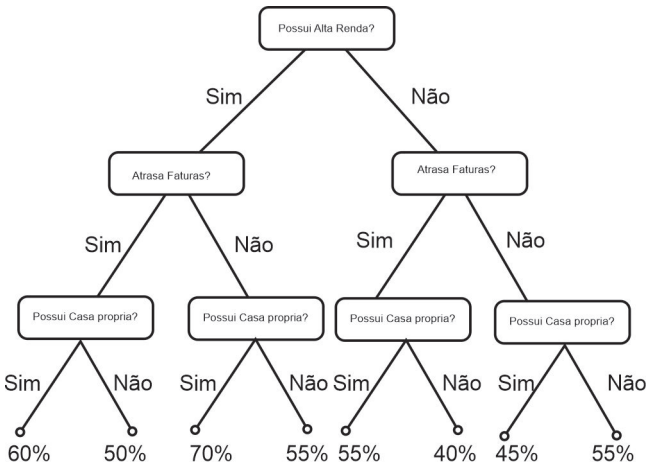

Figura 1: Ilustração - Árvore de decisão

Esse algoritmo será utilizado para os dados do cliente e dos produtos, pois nele será possível associar estes dados com as classes de satisfação, e comparar com a acurácia do modelo baseado nos *reviews*, tornando possível a verificação da hipótese 1.

VADER Sentiment Analyzer

VADER (Valence Aware Dictionary and Sentiment Reasoner) é uma ferramenta de análise de sentimento baseada em em anotações manuais de polaridade e valência (intensidade) de sentimento. É totalmente *open source* sob a Licença do *MIT (Massachusetts Institute of Technology)*. O algoritmo é baseado em valência, e tem curadoria humana, ou seja, para a construção do modelo foram utilizados rótulos que atribuiram ,por meio de anotação humana, os sentimentos associados aos textos.

O algoritmo é capaz de, com entradas de textos em língua inglesa, atribuir um coeficiente de -1 a 1 que nos fornece o sentimento associado da seguinte forma:

maior ou igual a -1 e menor que -0,05: Negativo
entre -0,05 a 0,05: Neutro
maior que -0,05 e menor ou igual a 1: Positivo

Iremos utilizar essa ferramenta para análise dos *reviews* e comparação com as classes

dos *reviews*, para a verificação dos sentimentos atribuídos pelo algoritmo em cada classe (Negativo, Neutro e Positivo).

2.2 Base de Dados

A base de dados disponibilizada em [Real et al., 2019] contêm 132.373 *reviews* de compra submetidos ao site Americanas.com entre Janeiro e Maio de 2018. Não se pode considerar essa amostra como representativa do comportamento do brasileiro nos *reviews* de compras online, mas sim como representativa do momento em questão e dos consumidores desse site. Todavia, a eficiência das ferramentas apresentadas e as conclusões poderão ser expandidas para aplicação em outras bases de dados de avaliação de compras.

A base conta com 14 atributos, quais sejam:

- `submission_date`
- `reviewer_id`
- `product_id`
- `product_name`
- `product_brand`
- `site_category_lv1`
- `site_category_lv2`
- `review_title`
- `overall_rating`
- `recommend_to_a_friend`
- `review_text`
- `reviewer_birth_year`
- `reviewer_gender`
- `reviewer_state`

Tais dados caracterizam a compra a partir de dados do consumidor e do produto. Torna-se interessante, portanto, analisar o perfil dos consumidores afim de entender a representatividade do presente trabalho. A partir do atributo de data de nascimento, criamos um novo campo de "faixa de idade", onde dividimos os consumidores em faixas de 10 em 10 anos, a partir de 0 anos, até um campo onde juntamos todos com mais de 80 anos. Observe na figura 1 a distribuição dos clientes nas faixas:

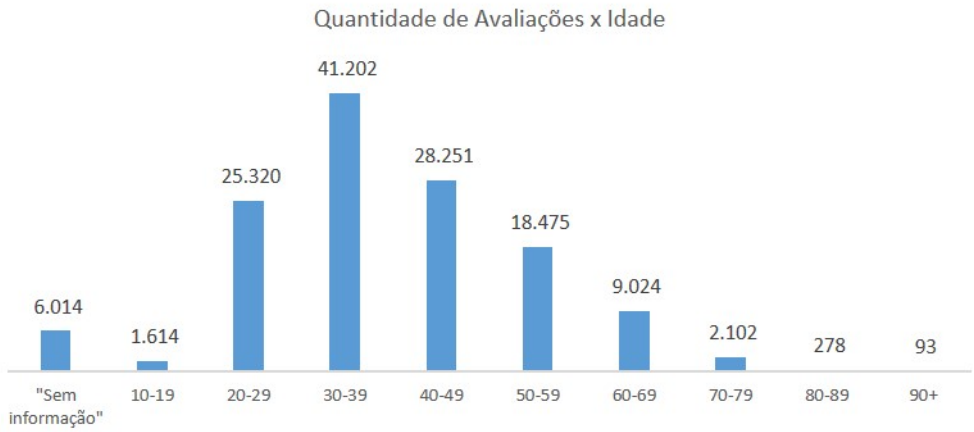

Figura 1: Faixa de Idade - Consumidores - Base

Analisando as notas em "estrelas"dadas por faixa de idade, observa-se que não há uma tendência de aumento ou diminuição do valor médio de avaliação em função das faixas de idade, conforme na Figura 2:

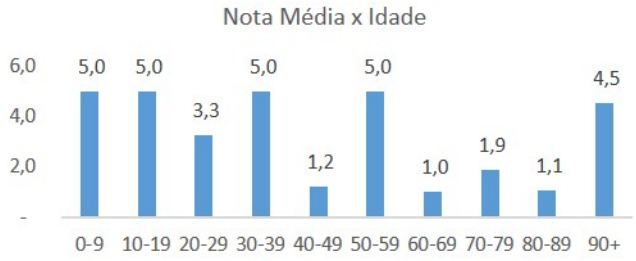

Figura 2: Categorias - Produtos - Base

Observando-se os campos que caracterizam os produtos a partir da categoria, analisamos, também, a distribuição dos *reviews* nas categorias para entender a representatividade das categorias no trabalho. Na figura 3 observamos a distribuição dos *reviews* pelas categorias de produtos:

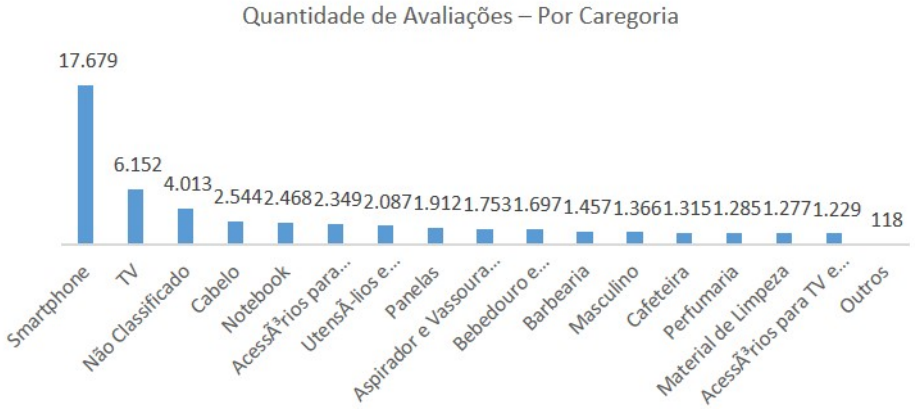

Figura 3: Categorias - Produtos - Bases

A classificação da compra se dá nos campos review_text e overall_rating onde temos a avaliação textual em língua portuguesa e a nota em "estrelas"discutida anteriormente.

A classificação dos textos se dará a partir do campo overall_rating onde iremos reclassificar os dados em 3 classes: Negativo, Neutro e Positivo a partir da classificação original já apresentada em [Real et al., 2019] onde se classifica:

- 1 a 2 estrelas: Negativo

- 3 a 4 Estrelas: Neutro

- 5 Estrelas: Positivo

Posteriormente discutiremos com mais profundidade reclassificação.

Já com os campos que caracterizam o cliente e o produto, separamos os campos a serem utilizados no modelo *Extra Trees Classifier* disponível na biblioteca Sickit-Learn do Python.

Os campos selecionados para esse modelo foram:

- `product_brand`
- `site_category_lv2`
- `reviewer_birth_year`
- `reviewer_state`

Tais campos foram escolhidos, pois são os campos da base que trazem as informações mais relevantes sobre o produto e o cliente. Tal determinação foi efetuada analisando os campos que de fato agregavam informação, eliminando campos que eram subcategorias dos campos atuais e que não afetaravam o resultado do algoritmo, como campos de *id* (identificação do cliente) e data de submissão. Houve tentativas de se incluir outros campos, porém em nada alterou a capacidade preditiva do modelo. Conforme discutido, a partir do atributo de data de nascimento foi criado um novo atributo de faixa de idade, sendo faixas de 10 em 10 anos a partir de 0 e uma faixa acima de 80 anos.

A partir do campo de avaliação textual (`review_text`) utilizaremos os modelos de PLN e análise de sentimento para a predição das classes definidas para verificação da hipótese 2 apresentada. Já comparando a acurácia dos dois principais tipos de modelos implementados, será possível a verificação da hipótese 1 apresentada.

CAPÍTULO 3

Pré Processamento e Processamento

3.1 Rotulagem dos Dados

Na base, temos rotuladas as estrelas dadas pelos clientes no campo `overal_review` que são nossas classes iniciais. Conforme explicado, foram definidas 3 classes, a partir das 5 estrelas a fim de simplificar o entendimento e o processamento dos dados:

- 1 a 3 estrelas: Negativo
- 3 a a estrelas: Neutro
- 5 estrelas: Positivo

Essa rotulagem servirá para o treinamento dos dois modelos (a partir dos *reviews* e dos dados do cliente/produto) bem como para a análise frente aos resultados obtidos pelo VADER

3.2 Balanceamento das classes

A partir da rotulagem foram obtidos 35.758 *reviews* positivos, 48.660 *reviews* neutros e 47.955 *reviews* negativos. Para se balancear as classes, foi feita subamostragem aleatória de modo que foram utilizados 35.758 *reviews* para cada as classe (Negativo, Neutro e Positivo.)

Utilizamos 70% dos dados para treinamento e 30% para o teste dos modelos, em todos os processamentos.

3.3 Seleção de Entradas

Conforme já discutido, para cada tipo de processamento foram selecionados os dados a serem usados como entradas para os modelos. Para o modelo que utiliza o NLTK e o Naive Bayes foram utilizados os apenas o campo de *reviews*. Já para o modelo que utiliza o *Extra Trees Classifier* foram utilizados todos os principais dados do avaliador (mencionados anteriormente) e do produto, já mencionados.

3.4 Ajustes de Atributos

Alguns ajustes foram necessários para o processamento dos dados das avaliações de compra. Para o caso dos *reviews* foram aplicados tratamentos comumente utilizados para processamento de informação em língua natural, a saber:

- Limpeza de caracteres (pontos, vírgulas e demais caracteres que não acrescentem informação relevante)
- Tokenização utilizando ferramenta da NLTK
- Remoção de StopWords utilizando ferramenta da NLTK

Para os dados a serem utilizados para o modelo com o Extra Trees Classifier apenas criamos, a partir do atributo que nos informa o ano de nascimento do cliente, um campo com a faixa de idade, conforme discutido.

3.5 Tradução para Língua Inglesa

Para a utilização do VADER, tornou-se necessária a tradução dos *reviews* para a língua inglesa. Para isso foi utilizada uma API de tradução da empresa norte-americana Google: *Google Translate API*.

3.6 Processamento - *Reviews*

A partir dos dados pré-processados, realizamos o treinamento do modelo classificador baseado em Naive Bayes na base de treino, composta por 30% do dataset já balanceado, utilizando os dados rotulados. Para este modelo foi utilizado o algoritmo NaiveBayes Multinomial utilizando os dados rotulados e também o VADER sem a rotulação de dados, pois este último dispõe de um dicionário próprio. Assim tornou-se possível a comparação dos resultados de ambos os modelos, bem como a verificação de como se comportam as classes

pré-definidas em um modelo que não é treinado a partir das mesmas, ou seja, se o algoritmo VADER Classifica como Negativas, Neutras e Positivas de forma semelhante à reclassificação proposta inicialmente.

3.7 Processamento - Dados

De modo semelhante ao processo utilizando o Naive Bayes Multinomial , realizamos o treinamento na base de treino, composta por 30% do dataset já balanceado, utilizando os dados rotulados. Dessa vez, porém, utilizamos o algoritmo *Extra Trees Classifier*.

CAPÍTULO 4

Resultados e Conclusão

4.1 Resultados | *Reviews*

Para os algoritmos em que realizamos treinamento a partir de *reviews* (Naive Bayes e VADER) obtivemos resultados que de fato expressam um potencial preditivo, porém não com a reclassificação incicial proposta:

- Acurácia de predição a partir dos *reviews* com (Naive Bayes Multinomial): 63%

De fato essa acurácia se mostrou não satisfatória, pois não atingiu o valor de acurácia definido incialmente como satisfatório. Porém ao estudarmos e nos aprofundarmos nos casos em que houveram erros, observamos que grande parte dos erros de predição advinham do fato de que a Classe "4 estrelas"reclassificada como "Neutro"(que chamaremos de Reclassificação 1) diminui a acurácia do modelo e é menos adequada do que utilizando a classe "4 estrelas"como "Positivo"(que chamaremos de Reclassificação 2). Realizamos, portanto, o mesmo procedimento, porém com a Reclassificação 2:

1 a 2 estrelas: Negativo
3 estrelas: Neutro
4 a 5 estrelas: Positivo

Dessa maneira, obtivemos uma acurácia substancialmente maior, aumentando para 84%. Observe na Tabela 1 abaixo os resultados obtidos para o verificador na matriz de confusão e

na tabela 2 de Precisão e Recall:

Tabela 1: Matriz de Confusão - Naive Bayes Multinomial

	Matriz de Confusão	Classe Real		
		Positivo	Neutro	Negativo
Saida do Verificador Naive Bayes Multinomial	Positivo	11.425	1.537	1.361
	Negativo	1.171	13.855	1.880
	Neutro	105	5.235	20.807

Tabela 2: Precisão e Recall - Naive Bayes Multinomial

Classe	Precisão	Recall
Positivo	85%	90%
Negativo	82%	67%
Neutro	81%	87%

Assim, podemos verificar como verdadeira a hipótese 2 apresentada, pois podemos predizer com razoável acurácia a satisfação do cliente a partir de uma análise automatizada dos *reviews* de compra pelos algoritmos.

Os termos predominantes em cada classe são descritos na Tabela 3.

Tabela 3: Termos Predominantes - Algoritmo Naive Bayes Multinomial

Termos Predominantes	
Classe	Termos
Negativo	produto' 'recebi' 'defeito' 'não' 'pessimo' 'entrega' 'ainda' 'ruim
Neutro	produto' 'gostei' 'custo' 'beneficio' 'atende' 'preço' 'porem' 'qualidade'
Positivo	produto' 'excelente' 'gostei' 'otimo' 'recomendo ' 'qualidade' 'adorei' 'otima'

Tendo em vista uma maior acurácia para o modelo com essa última reclassificação proposta, utilizamos o VADER para entender como se comportam as classes do Vader Sentiment em relação ao atributo número de estrelas de modo a verificarmos se o resultado corrobora com a nova reclassificação e entendermos se, de fato, essa nova classificação se torna mais adequada para uma predição em 3 classes, posto que a rotulação dos dados foi proposta a partir das estrelas.

Na Tabela 4, podemos comparar os resultados do VADER com o número de estrelas atribuidas pelos clientes:

Tabela 4: Classificação Vader x Classificação Estrelas

		Estrelas				
		1	2	3	4	5
Vader	Negativo	61%	15%	12%	6%	5%
	Neutro	35%	12%	16%	18%	19%
	Positivo	6%	3%	12%	31%	48%

Observa-se que, dado que o modelo VADER não é treinável e possui sistemas de regras próprio, as classes correspondentes ao número de estrelas não se concentram totalmente nas reclassificações propostas, porém observa-se que, conforme se esperava, a classificação 4 estrelas se apresenta mais próxima de 5 estrelas do que de 3 estrelas em termos de respostas positivas. Isso corrobora com a proposta de reclassificação discutida, onde a classe "Positivo"abarca tanto 4 quanto 5 estrelas.

Observe nas Tabelas 5 e 6 a relação dos resultados obtidos no VADER (Linhas) comparados com os resultados do Naive Bayes Multinomial (colunas) com as reclassificações testadas. Nas tabelas analisamos como o Naive Bayes Classificou os dados, postas as saídas do VADER, fazendo com que se some 100% os valores das linhas. Observe, também, o índice Kappa que também nos ajuda a analisar a concordância entre os modelos.

Tabela 5: Classificação Vader x Naive Bayes (Reclassificação 1)

		Naive Bayes Multinomial (Reclassificação 1)		
		Negativo	**Neutro**	**Positivo**
VADER	**Negativo**	76%	12%	11%
	Negativo	47%	16%	37%
	Positivo	9%	12%	79%

Coeficiente Kappa Vader x Naive Bayes (Reclassificação 1): 25%

Tabela 6: Classificação Vader x Naive Bayes (Reclassificação 2)

		Naive Bayes Multinomial (Reclassificação 2)		
		Negativo	**Neutro**	**Positivo**
VADER	**Negativo**	76%	18%	5%
	Negativo	50%	33%	17%
	Positivo	9%	43%	48%

Coeficiente Kappa Vader x Naive Bayes (Reclassificação 1): 43%

Observe que, na Reclassificação 2 proposta, o VADER concentra a intersecção das classes "Positivas"muito mais do que na Reclassificação 1. Observamos, também, que o coeficiente Kappa, corrobora para uma maior concordância entre os modelos com a Reclassificação 2 proposta.

Desse modo, sugere-se para trabalhos futuros a utilização dessa reclassificação em modelos de *reviews* de compras que se assemelhem ao presente trabalho.

Importante ressaltar que, na análise dos *reviews* em que o modelo errava a predição da classe, observou-se diversos casos em que o texto apresentava claras incoerências entre o número de estelas dadas e o *review* submetido. Em diversas avaliações de compra os textos de *review* se apresentavam incompreensíveis, com textos copiados de textos aleatórios ou com textos ininteligíveis.

4.2 Resultados | Cliente/Produto

Para o algoritmo em que foram utilizados os dados do Cliente e do Produto, também advindos do dataset, os resultados não expressaram um potencial preditivo.

- Acurácia de predição a partir dos dados do Cliente/Produto) com o *Extra Trees Classifier*: 34%

Observamos que o comportamento dos dados que caracterizam o Cliente e o Produto adquirido, na avaliação de compra, têm carater randômico, não ajudando a predizer a satisfação de compra, de modo que, no caso da análise da hipótese 1, a possibilidade de não se poder predizer a satisfação do cliente a partir dos dados do cliente e do produto (hipótese nula) não pôde ser rejeitada.

4.3 Conclusão

Podemos concluir que a Hipotese 1 não pôde ser aceita, posto que os dados do Cliente e do Produto não são relevantes para a predição da satisfação do cliente com a compra pela metodologia utilizada. Em contraste com esse fato os *reviews* textuais se mostraram absolutamente mais relevantes para a predição da satisfação.

Já a hipótese 2 apresentada, foi verificada verdadeira, na medida em que um modelo preditivo com acurácia de 84% foi obtido a partir da nova reclassificação proposta que se mostrou mais eficiente.

Tendo em vista que a base é formada por dados reais de compra e apresenta um nível razoável de incoerência entre as avaliações dadas em "estrelas"e o *review* apresentado, podemos concluir que os resultados foram satisfatórios e a ferramenta poderia ser implementada

para diversas aplicações tais como ofertas diferenciais para clientes com alta propensão de satisfação e melhora da experiência de Clientes com baixa propensão de satisfação.

Referências

Telles, R., Elias, M.B.S., Takahata A.K., Silva Júnior, L., Análise das relações entre disciplinas do Ensino Médio do Brasil por meio de questões de vestibular com uso de técnicas de PLN, 2019. In: Proceedings of VI Student Workshop on Information and Human Language Technology TILIC 2019, pp.354-357. 2019.

Real, L., Oshiro, M. e Mafra, A., B2WReviews01: An open product reviews corpus". In: Proceedings of XII Symposium in Information and Human Language Technology - (STIL 2019), pp. 200-208. 2019.

Barbosa J.L.N, Vieira J.P.A, Santos R.L.S., Junior, G.V.M,Moura, R.S., Introdução ao Processamento de LinguagemNatural usando Python, 2017.

Yokoi, L.M., Pinheiro, T.S., Oliveira, F.S., Santos, P.D., Takahata, A.K., Suyama, R., Simões, P.W., Modelo Preditor Bayesiano aplicado a Análise de Sobrevivência em Mulheres com Câncer de Mama. In: XVII Congresso Brasileiro de Informática em Saúde (CBIS' 20), 2020, Foz do Iguaçu, PR. Anais Estendidos do XVII Congresso Brasileiro de Informática em Saúde, p. 38-39, 2020.
Disponível em: <http://sbis.org.br/wp-content/uploads/2021/07/Anais_Estendidos_CBIS2020.pdf>. Acesso em: 30 de Novembro de 2021.

Geurtz, P., Ernst, D., Louis, W., Extremely randomized trees, 2006.

Rennie, J. D. M., Shih, L., Teevan, J., Karger, D.R., Tackling the Poor Assumptions of Naive Bayes Text Classifiers, 2003.

Palmer, D. D., Text preprocessing. In Handbook of natural lan- guage processing. Chapman and Hall/CRC, 2nd edition, 2010.
fancyhdr