

UM PROCESSO DE CLASSIFICAÇÃO DE TEXTO: ANÁLISE DE SENTIMENTO DAS OPINIÕES NO TRIPADVISOR® SOBRE A ATRAÇÃO OKTOBERFEST BLUMENAU

Marcio Crescencio¹;

Alexandre Leopoldo Gonçalves²;

José Leomar Todesco³;

***Abstract:** The aim of this article was to develop an experiment with a sentiment analysis of user opinions on TripAdvisor® about the Oktoberfest Blumenau attraction using Data Mining and Machine Learning through a Knowledge Discovery in Data process. Two supervised sentiment classification approaches were implemented in Python®, based-model probabilistic on the Multinomial Naïve Bayes and the vector representation of words model using Word2Vec. The performance of models was evaluated and compared with measurements: Accuracy, Precision, Recall and F-score. The probabilistic model achieved an accuracy of 90%, while the recurrent neural network model LSTM was 92%. The feeling of the opinions is positive for the features of traditional German party, variety and number of drinks, and music and animation. Opposite for the queues in bathrooms and overcrowding on Saturday nights.*

***Keywords:** Sentiment analysis; Machine Learning; Classification; Knowledge Discovery in Data process.*

Resumo: O objetivo desse artigo foi desenvolver um experimento com a análise de sentimento das opiniões no TripAdvisor® sobre a atração Oktoberfest Blumenau utilizando a Mineração de Dados e Aprendizado de Máquina através de um processo de Descoberta de Conhecimento em Dados. Duas abordagens de classificação supervisionada de sentimento foram implementadas em Python®, o modelo probabilístico baseado no algoritmo Multinomial Naïve Bayes e o

¹ Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento – Universidade Federal de Santa Catarina (UFSC) Florianópolis – Brasil. Correo eletrônico: marcio.crescencio@posgrad.ufsc.br

² Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento – Universidade Federal de Santa Catarina (UFSC) Florianópolis – Brasil. Correo eletrônico: a.l.goncalves@ufsc.br

³ Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento – Universidade Federal de Santa Catarina (UFSC) Florianópolis – Brasil. Correo eletrônico: jose.todesco@ufsc.br

modelo de representação vetorial de palavras usando o Word2Vec. O desempenho dos modelos foi avaliado e comparado usando as métricas: Acurácia, Precisão, Recall e F-score. O modelo probabilístico alcançou acurácia de 90%, enquanto o modelo de rede neural recorrente LSTM foi de 92%. O sentimento nas opiniões é positivo para as características da festa típica alemã, a variedade e quantidade de bebidas, as músicas e a animação. O sentimento é negativo para filas nos banheiros e a superlotação aos sábados à noite.

Palavras-chave: Análise de Sentimento; Mineração de Dados; Classificação; Processo de Descoberta de Conhecimento em Dados.

1 INTRODUÇÃO

A análise de sentimento ou mineração de opinião ganhou atenção devido à grande quantidade de opiniões e outros textos de contribuições nas plataformas sociais. É uma área de pesquisa que estuda principalmente um conjunto de opiniões subjetivas que expressam ou implicam sentimentos positivos ou negativos em um domínio de aplicação, tais como: notícias, *sites* de compras, postagens em mídias sociais ou fóruns de discussão, e outros (Aggarwal, 2018; Liu, 2015).

Na literatura acadêmica recente existe uma variedade muito grande de pesquisa nessa área com diferentes tendências e abordagens. Assim como os métodos disponíveis, os desafios também são abrangentes, portanto, a etapa mais crítica da análise de sentimento é a escolha da técnica apropriada para classificar os sentimentos. Avaliar e comparar o desempenho entre diferentes métodos em termos de acurácia, precisão, pontos fortes e fracos é um caminho viável para a tomada de decisão (Cambria *et al.*, 2017; Devika *et al.*, 2016; Giatsoglou *et al.*, 2017; Hemmatian & Sohrabi, 2019; Rossi, 2019; Singh *et al.*, 2017).

O objetivo desse estudo foi conduzir um experimento de análise de sentimento de uma atração do TripAdvisor® utilizando as técnicas de mineração de dados e aprendizado de máquina a partir de um processo de Descoberta de Conhecimento em Dados. A atração cultural Oktoberfest Blumenau foi escolhida por se tratar da maior festa alemã do Brasil e receber mais de 500 mil visitantes em cada edição. As fases do processo consistiram da coleta de dados (*web scraping* do *site*), pré-processamento do texto, classificação, interpretação e avaliação.

As opiniões foram separadas em sentenças e rotuladas manualmente como positivas ou negativas. Duas abordagens de classificação supervisionada de sentimento foram empregadas, o modelo probabilístico baseado no algoritmo *Multinomial Naive Bayes* e o modelo de representação vetorial de palavras com o algoritmo *Word2Vec*. O desempenho dos modelos foi avaliado e comparado usando as métricas: Acurácia, Precisão, Recall e F-score.

2 ANÁLISE DE SENTIMENTO

A análise de sentimento ou mineração de opinião é definido por Liu (2015) como um campo de estudo que analisa as opiniões, sentimentos das pessoas em relação a entidades como produtos, serviços, organizações, indivíduos, questões, eventos, tópicos e seus atributos. Atualmente, ela é considerada um subconjunto do processo de mineração de conteúdo da Web (Hemmatian & Sohrabi, 2019). O objetivo dessa área é a identificação de informações subjetivas e não triviais em textos não estruturados (Giatsoglou *et al.*, 2017). Por isso, requer muitas tarefas de processamento de linguagem natural para um sistema entender aspectos positivos ou negativos das entidades ou tópicos alvos (Cambria *et al.*, 2017).

Segundo Liu (2015), o básico de uma opinião consiste de dois componentes-chaves: um alvo g e um sentimento s a respeito do alvo:

(g, s) onde,

- g pode ser uma entidade ou aspecto sobre o qual a opinião foi expressa;
- s um sentimento positivo, negativo ou neutro, ou ainda, uma avaliação numérica expressando ponto de força/intensidade de um sentimento, por exemplo, estrelas de 1 a 5.

Na prática, a identificação completa do alvo pode ser complexa e nem aparecer na mesma frase. Neste caso, o alvo pode ser decomposto e descrito de maneira estruturada com vários níveis, o que facilita a mineração de sentimento e o uso posterior dos resultados das opiniões extraídas.

O problema análise de sentimento, na concepção de Liu (2015), representa um conjunto de subproblemas inter-relacionados. O nível de dificuldade também possui diferentes dimensões, dependendo do domínio de aplicação. Quanto ao nível de granularidade, pode ser em nível de documento, sentença ou aspecto. No nível de documento, a suposição implícita é

que o documento expressa uma única opinião sobre um alvo específico (previamente conhecido). Na abordagem tradicional, o documento é tratado como um conjunto de palavras (*bag-of-words*), a tarefa é descobrir a sua polaridade. No entanto, a simplicidade do modelo ignora as informações estruturais em um documento, que podem ser críticas para detecção de emoções sociais (Tang *et al.*, 2019). Portanto, uma nova abordagem está se tornando popular para este nível: o *Word Embeddings*. A incorporação de palavras é uma representação vetorial das palavras, na qual palavras com significados semelhantes são mapeadas mais próximas umas das outras. Essa abordagem apresenta níveis de precisão mais altos do que as abordagens tradicionais (Rudkowsky *et al.*, 2018).

No nível de sentença, cada frase é analisada separadamente, são classificadas conforme a sua polaridade positiva, negativa ou neutra e sua força pode ser expressada numericamente. Em alguns casos, é difícil definir como uma sentença deve ser tratada. Geralmente, as sentenças precisam primeiro ser classificadas como subjetivas ou objetivas (Wang *et al.*, 2017). As frases subjetivas normalmente contêm muito adjetivos e frases emocionais, enquanto as frases objetivas contêm declaração de fato. Portanto, um problema importante neste nível é a classificação da subjetividade. Muitas frases podem conter opinião que não estão se referindo a entidade. Isso implica que é muito importante especificar o alvo da opinião para tornar a mineração realmente útil. A análise de opinião em nível de entidade e do aspecto visam as características refinadas de uma entidade (Aggarwal, 2018). Devido ao seu refinamento, a classificação em nível de palavras é a que pode resultar na melhor classificação de opinião (Wang *et al.*, 2017).

Segundo Zheng *et al.* (2018), os métodos utilizados para a classificação da análise de sentimento geralmente são: a abordagem orientada a semântica e a abordagem estatística de aprendizado de máquina. A abordagem semântica determina o sentimento de um documento baseado na extração de palavras (aspectos) e frases (sentenças) de sentimentos. Enquanto a abordagem estatística determina o sentimento de um documento baseado na extração de características sentimentais e aprendizado de máquina. Esse último será o foco desse trabalho.

2.1 APRENDIZADO DE MÁQUINA PARA CLASSIFICAÇÃO DE SENTIMENTO

O problema de classificação é suportado pela mineração de dados (*Data Mining*) e aprendizado de máquina (*Machine Learning*). A mineração de dados é um processo de extração de conhecimentos novos, válidos, úteis e compreensíveis. Na classificação, o *corpus* (conjunto de documentos) é particionado em *classes* (p.ex.: positivo, negativo). Exemplos treinados são fornecidos associando pontos de dados a rótulos que indicam sua participação na classe. Quando os rótulos não estão disponíveis, o objetivo é determinar a classe com o uso de um modelo supervisionado construído utilizando o conjunto de treinamento (Aggarwal, 2018; Aghdaie, 2017).

A abordagem de aprendizado de máquina para classificação exige dois conjuntos de documentos: um para treinamento e outro de teste. O conjunto de treino serve para aprender os indícios da diferença dos documentos usados pelo classificador automático. O conjunto de testes valida o desempenho do classificador automático usado pelas técnicas de aprendizado de máquina para classificar o número de revisões adotadas. Essa abordagem contém vários tipos de algoritmos classificadores, os mais comumente utilizados em textos curtos são *Naïve Bayes*, *Support Vector Machine – SVM* e os *Neural Networks* (Bhonde *et al.*, 2015; Ceci *et al.*, 2016; Giatsoglou *et al.*, 2017). Nesse trabalho, o experimento foi conduzido com a aplicação dos algoritmos: *Multinomial Naive Bayes* e *Word2Vec*, com o objetivo de comparar os seus resultados.

O *Multinomial Naive Bayes* é um método de aprendizado probabilístico supervisionado, no qual a probabilidade de um documento d estar na classe c é computada como:

$$P(c|d) \propto P(c) \prod_{1 \leq k \leq n_d} P(t_k|c)$$

onde, $P(t_k|c)$ é uma probabilidade condicional do termo t_k ocorrer no documento da classe c . $P(t_k|c)$ é interpretado como uma medida da quantidade de evidência que t_k contribui para que c seja a classe correta. Se os termos de um documento não fornecerem uma evidência clara versus outra, é escolhido uma com probabilidade anterior mais alta. $(t_1, t_2, \dots, t_{n_d})$ são os tokens em d que são partes de um vocabulário usado para classificação e n_d é o número de tokens buscados em d (Manning *et al.*, 2008).

O *Word2Vec* é um método para representação distribuída de palavras em um espaço vetorial de dados de textos não estruturados. A representação de palavras não é recente, mas

este método foi aperfeiçoado por Mikolov *et al.* (2013) usando uma técnica de deslocamento de palavras onde são executadas operações algébricas simples nos vetores de palavras. O *Word2Vec* trata cada palavra como um único elemento do modelo de representação vetorial, atribuindo uma representação distinta para cada palavra no corpus do texto ignorando estruturas internas da própria palavra (Kambali *et al.*, 2018). Em textos mais longos, a frequência do termo geralmente transmite evidências suficientes para lidar com decisões simples de aprendizado de máquina, como a classificação binária (Aggarwal 2018). Dois modelos predominam no treinamento de *word embeddings*: o *continuous bag-of-words* (CBOW) e *skip-gram* (Rong, 2014). No modelo *skip-gram*, o contexto C é predito a partir da palavra alvo w , enquanto no *CBOW* acontece o contrário, a palavra alvo w é predita, dado um contexto C .

Na análise de sentimento, o algoritmo *Word2Vec* pode ser usado para contar a frequência de cada palavra na sentença e vincular essa contagem a coleção de texto. Esse processo obtém um vocabulário, ou seja, uma lista de palavras recorrentes no texto, em que cada palavra terá seu próprio índice. Isso permite criar um vetor para cada sentença. Utilizando esse vetor numérico é possível determinar o espaço geométrico formado pelos vetores e então obter a similaridade.

3 PROCEDIMENTOS METOLÓGICOS

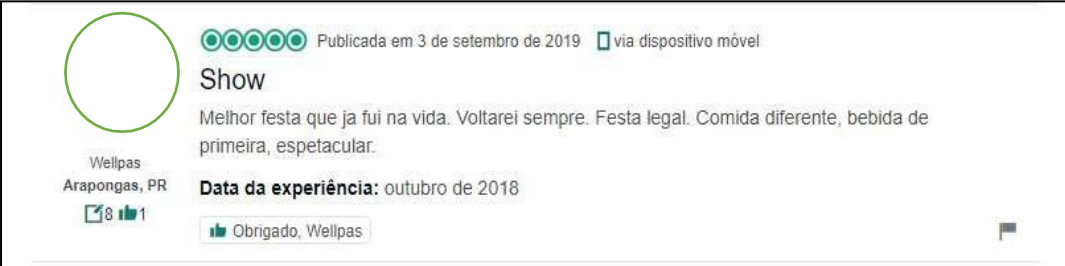
A proposta desse estudo foi realizar uma análise de sentimento ou opinião de uma atração turística aplicando técnicas de mineração de dados e aprendizado de máquina através de um processo conhecido por *Knowledge Discovery in Data* – KDD, ou processo de descoberta de conhecimento em dados (Klösgen, 1996; Kodratoff, 1999). Este processo possui várias fases dependendo do domínio de aplicação. Nesse trabalho serão detalhadas as fases de coleta dos dados, pré-processamento, classificação (*Data Mining*), interpretação e avaliação. Todo o processo foi realizado usando a linguagem de programação Python[®], seguindo modelos de códigos disponibilizados nos tutoriais das bibliotecas utilizadas no notebook do projeto⁴.

⁴ O Notebook Jupyter é um ambiente computacional interativo para execução de código, plotagens, entre outros. Para mais detalhes consulte o site <<https://jupyter.org/>>.

A coleta dos dados foi realizada utilizando o básico de uma técnica automatizada de coleta de dados conhecida como *web scraping* (Mitchell, 2018). Essa fase consistiu na localização das opiniões das pessoas sobre a atração turística Oktoberfest Blumenau no site TripAdvisor®. A extração dos dados foi codificada utilizando a biblioteca *Beautiful Soup* ou BS4 que serve para extrair dados de arquivos HTML e XML. Os elementos (*id*, *css*, *hyperlinks*, *tags*, etc.) que continham os dados das revisões foram localizados na página HTML utilizando a biblioteca *Selenium*, convertidos e guardados em uma estrutura de arquivo *JSON*, que nada mais é do que uma coleção de objetos contendo vários *dicts* (coleção de dados), onde cada *dict* representa uma opinião sobre a atração. Na Figura 1, um exemplo de opinião: a) original do site TripAdvisor® e b) no formato *dict*.

Figura 1 - a) uma opinião no formato original; b) um dict python dessa opinião.

a)



b)

```
57 {
58     "IdRevisao": "706196082",
59     "NomeUsuario": "Wellpas",
60     "LocalUsuario": "Arapongas, PR",
61     "Titulo": "Show",
62     "TextoRevisao": "Melhor festa que ja fui na vida. Voltarei sempre. Festa legal. Comida diferente, bebida de primeira,
63     espetacular.",
64     "DataExperiencia": "Data da experiência:",
65     "Pontos": 5.0
66 }
```

Fonte: Elaborado pelos autores.

Conforme dito antes, o problema da análise de sentimento de texto não estruturado necessita de processamento da linguagem natural porque o texto coletado é contaminado com sinais, símbolos e caracteres, palavras ambíguas, preposições, conjunções, pronomes e artigos que são consideradas *stop words*, tipicamente inadequadas para a classificação, pois adicionam uma grande quantidade de ruído. Quanto mais sujo o texto, pior serão os resultados da classificação. Portanto, a próxima etapa do processo é a de preparação dos dados ou pré-processamento do texto para a classificação. Primeiro, utilizando o submódulo *Tokenize* da biblioteca NLTK, os textos foram divididos em sentenças e armazenados em uma lista.

Devido a especificidade dos dados coletados, optou-se por rotular manualmente as sentenças para analisar o desempenho da classificação do texto. Com isso, um arquivo com extensão *csv* foi criado contendo duas colunas, sendo uma para o texto e outra para o rótulo, no qual o dígito 1 corresponde a classe dos positivos e o dígito 0 corresponde a classe dos negativos.

Em seguida, conforme a Figura 2, percorrendo a coleção de sentenças (linha 4), as palavras foram normalizadas em caixa baixa, removidos os tokens não alfabéticos e as palavras de parada (linha 9), removido a acentuação e as palavras reduzidas até a sua raiz/base (linha 14), técnica conhecida por *stemming*. Como exemplo, uma sentença em seu formato normal: [A Oktoberfest é uma festa extraordinária para todos os públicos!], depois do pré-processamento: ['oktoberfest', 'fest', 'extraordin', 'tod', 'publ']. Com isso, finalizou-se a etapa de pré-processamento dos dados.

Figura 2 - Parte do código sobre a etapa de pré-processamento do texto.

```
1 # pre-preparação das sentenças para classificação
2 all_words = []
3 sentence_words = []
4 for item in sentences:
5     #print(item)
6     # separa em palavras
7     tokens = word_tokenize(item)
8     # remove todos os tokens que não são alfabético
9     words = [word for word in tokens if word.isalpha()]
10    # remove as stopwords
11    words = [word for word in words if not word in STOP_WORDS]
12    # Stemming essa função serve para diminuirmos a palavra até a sua raiz/base
13    # "correr" e "corrida" == 'corr'
14    words = [rslpsstemmer.stem(word) for word in words]
15    # add todas as palavras na lista
16    [all_words.append(word) for word in words]
17
18    sentence_words.append(words)
19
```

Fonte: Elaborado pelos autores.

A etapa de classificação teve início com a verificação de como estava o balanceamento das classes. Conjunto de dados desbalanceados prejudicam o desempenho porque a maioria dos algoritmos de classificação são preparados para fornecer o máximo de precisão e redução de erros. Sendo assim, se o conjunto de dados possui quantidade muito maior de uma classe, haverá uma tendência do modelo aprender pontos de dados específicos, ocasionando *overfitting* (excesso de ajustes) e pouca generalização nos dados de teste (Aggarwal, 2018). O

desbalanceamento do conjunto de dados foi confirmado, pois continha 1.957 sentenças rotuladas como positiva e somente 265 rotuladas como negativa. Diante disso, duas estratégias de reamostragem foram testadas. A primeira, consistiu no descarte de amostras da classe majoritária (*under-sampling*), a segunda, adicionando mais exemplos na classe minoritária (*over-sampling*).

A segunda parte dessa etapa foi criar uma partição aleatória dos conjuntos de treinamento e teste para o classificador *Multinomial Naive Bayes* aprender o modelo e para a abordagem de validação cruzada (*cross-validation*). Na sequência, a coleção de textos foi convertida em uma matriz de ocorrência de *tokens*, utilizando a classe *CountVectorizer* da biblioteca *Sklearn*. Isto é uma tarefa necessária porque os algoritmos de aprendizado de máquina não trabalham corretamente com dados brutos (uma sequência de símbolos), a maioria deles esperam vetores de recursos numéricos com tamanho fixo⁵, em vez de texto bruto com comprimento variável.

Os conjuntos de treinamento e teste foram transformados (*fit_transform*, *transform*) em matriz do tipo termo-documento. Com isso, o conjunto de dados ficou pronto para o classificador *MultinomialNB()*, a predição da validação cruzada, a medição da acurácia e a construção da matriz de confusão.

O algoritmo *Word2Vec* do pacote *gensim* foi usado para criar um modelo próprio baseado em redes neurais para o treinamento do conjunto de textos (linha 4 da Figura 3). O resultado do modelo criado é um conjunto de vetores de palavras. O objeto *wv* (linha 8) contém essencialmente o mapeamento entre as palavras e a incorporação.

Figura 3 - Parte do código que cria o modelo vetorial de palavras.

```
1  ## WORD2VEC
2
3  # train model
4  model = Word2Vec(sentence_words, min_count=1, sg=0)
5  # summarize vocabulary
6  words = list(model.wv.vocab)
7
8  word_vectors = model.wv
```

Fonte: Elaborado pelos autores.

⁵ Fonte: <https://scikit-learn.org/stable/modules/feature_extraction.html#text-feature-extraction>.

A classe *Tokenizer* do pacote *Keras* foi utilizada novamente para vetorizar o corpus de texto em preparação para a classificação *word embeddings*, a partir do modelo CBOW. Para criar a camada de incorporação foi utilizado um tipo de Rede Neural Recorrente chamado *Long Short-Term Memory Units* – LSTM. Este tipo de rede processa a entrada em sequência, iterando pelos elementos e produzindo uma sequência de saída. O LSTM aprende e decide armazenar informações por intervalos de tempos prolongados através da retropropagação recorrente (Hochreiter & Schmidhuber, 1997).

A capacidade da rede neural pode ser controlada por dois aspectos de um modelo: o número de nodos e de camadas. A precisão que será obtida nos resultados da classificação depende desses números que devem ser definidos considerando o tamanho do conjunto de dados. Nesse experimento, optou-se por 32 nodos (*units*) porque o conjunto de dados é pequeno. O modelo foi compilado utilizando o otimizador (*adam*) e a função de perda (*loss*). Para treinar o modelo foi declarado 10 *epochs* (ponto de corte arbitrário). A etapa de interpretação e avaliação dos dados foi conduzida na sessão a seguir.

4 ANÁLISE E INTERPRETAÇÃO DOS DADOS

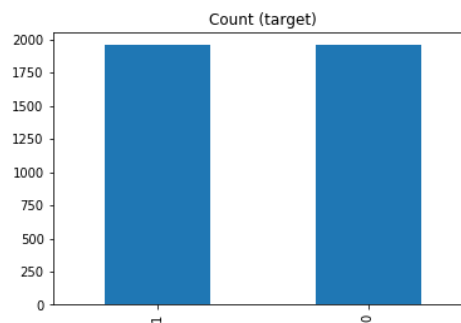
No *site* TripAdvisor® as pessoas podem expressar seu sentimento quanto a atração através de uma pontuação de 1 a 5, sendo 5 pontos a mais positiva. No processo de obtenção dos dados, foram capturados os dados referentes a essa pontuação. O sentimento nas revisões em termos de pontuação é positivo em 94% (4 e 5 pontos), neutro em 4% (3 pontos) e negativo em aproximadamente 2% (1 e 2 pontos). Considerando essa pontuação, é incontestável que a atração é muito bem avaliada, mas os textos podem indicar sentimento dúbio. Conforme Medhat *et al.* (2014), os detentores de opinião podem prover opiniões diferentes para diferentes aspectos da mesma entidade. A tarefa de detectar a subjetividade de um texto e então classificar sua polaridade é o problema da análise de sentimento.

No total foram coletadas 565 revisões exclusivas sobre a atração turística Oktoberfest Blumenau em outubro de 2019. Um conjunto de dados relativamente pequeno para tarefa de classificação, por essa razão este trabalho deve ser considerado um experimento de análise de sentimento. No pré-processamento do texto, as revisões foram subdivididas em 2.222 sentenças e pré-classificadas como: 1.957 positivas e 265 negativas. As técnicas de aprendizado de

máquina necessitam de uma amostra do conjunto de dados rotuladas para treinar o modelo de classificação. Para evitar o desbalanceamento do conjunto de dados foi utilizada a técnica *over-sampling*, adicionando mais exemplos na classe minoritária - opiniões negativas. Conforme a Figura 4, as classes ficaram balanceadas ambas em 1.957 exemplos.

Figura 4 - Balanceamento do conjunto de dados.

```
Random under-sampling:
1    1957
0    1957
Name: Tipo, dtype: int64
```



Fonte: Elaborado pelos autores.

Para avaliar o desempenho da classificação de sentimentos são considerados quatro parâmetros: Acurácia, Precisão, Recall e F1-Score (Bhonde *et al.*, 2015; Hemmatian & Sohrabi, 2019). A matriz de confusão da Figura 5, representa a classificação com o modelo probabilístico *Multinomial Naive Bayes*. Os erros do modelo são representados por 56 falso-negativos e 141 falso-positivos dos exemplos classificados. O valor da Acurácia de 0,909 indica que 90% dos dados treinados foram corretamente classificados, bem como, a métrica de Precisão 95% corresponde a proporção dos classificados como positivos que efetivamente eram exemplos positivos.

Figura 5 - Métricas do modelo MultinomialNB.

Acurácia: 0.9097					
	precision	recall	f1-score	support	
0	0.88	0.96	0.91	1471	
1	0.95	0.86	0.91	1464	
micro avg	0.91	0.91	0.91	2935	
macro avg	0.91	0.91	0.91	2935	
weighted avg	0.91	0.91	0.91	2935	
Matriz de Confusão do modelo MultinomialNB					
Predito	0	1	All		
Esperado					
0	1415	56	1471		
1	141	1323	1464		
All	1556	1379	2935		

Fonte: Elaborado pelos autores.

O modelo baseado em redes neurais *Word2Vec* resultou em 1.971 vetores de palavras. A Figura 6, representa um teste de similaridade das palavras que foi aplicado para verificar o funcionamento do modelo com as palavras positivas ‘oktoberfest’ e ‘chop’ e a palavra negativa ‘fil’. Este método calcula a similaridade de cosseno entre uma média simples dos vetores de peso da projeção das palavras fornecidas e os vetores para cada palavra no modelo. Apesar do conjunto de vetores ser pequeno, o resultado do teste foi satisfatório. Relembrando que no pré-processamento as palavras passaram por um processo de redução à raiz (*stemming*), portanto nesse exemplo, a palavra ‘fil’ pode significar fila ou filas.

Figura 6 - Teste de similaridade de palavras no modelo treinado.

```
In [13]: 1 word_vectors.most_similar(  
2     positive=['oktoberfest', 'chop'],  
3     negative=['fil'])  
  
Out[13]: [('alem', 0.9994899034500122),  
( 'alemanh', 0.9994856715202332),  
( 'tipic', 0.9994617700576782),  
( 'cervej', 0.9994420409202576),  
( 'danc', 0.9994394779205322),  
( 'boa', 0.9994301795959473),  
( 'music', 0.9994286298751831),  
( 'band', 0.9994219541549683),  
( 'brasil', 0.9994202852249146),  
( 'gastronom', 0.9994168281555176)]
```

Fonte: Elaborado pelos autores.

Na Figura 7, o desempenho do modelo de rede neural apresentou uma acurácia de 0,99 do conjunto de treinamento e de 0,92 para o conjunto de teste. A métrica de Precisão do modelo foi de 93%. Mesmo com uma pequena amostra, o modelo de rede neural apresentou um resultado mais satisfatório que o modelo probabilístico. Segundo Hemmatian e Sohrabi (2019), o método de rede neural fornece bons resultados contra ruídos nos dados.

Figura 7 - Plotagem das métricas do modelo de rede neural.

```
Training Accuracy: 0.9934  
Testing Accuracy: 0.9285  
  
Recall: 0.9295  
Precision: 0.9368  
F1_Score: 0.9293  
  
=== Confusion Matrix ===  
[[483  3]  
 [ 66 427]]
```

Fonte: Elaborado pelos autores.

A diferença entre a acurácia do conjunto de treinamento e a parcela do texto de validação é ilustrado na Figura 8. O ideal seria que não fosse observada uma diferença significativa. O gráfico da Figura 9 representa que o valor de perda ficou mais alto, indicando que haverá uma grande diferença entre as previsões do modelo treinado e a produção real. O ideal seria que a perda tendesse a zero, o que significaria não haver diferença entre o que o modelo aprende e o que realmente deve ser. Isto indica que é preciso trabalhar com conjunto de dados maiores para obter resultados mais confiáveis.

Figura 8 - Acurácia do treinamento.

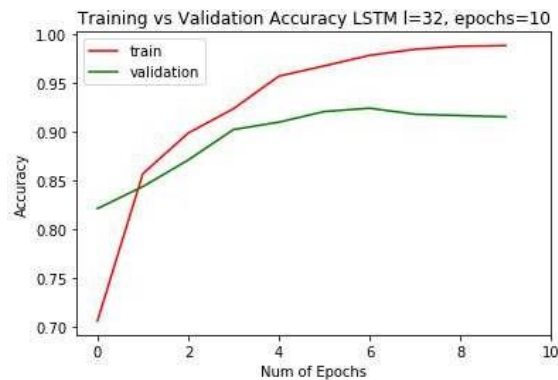
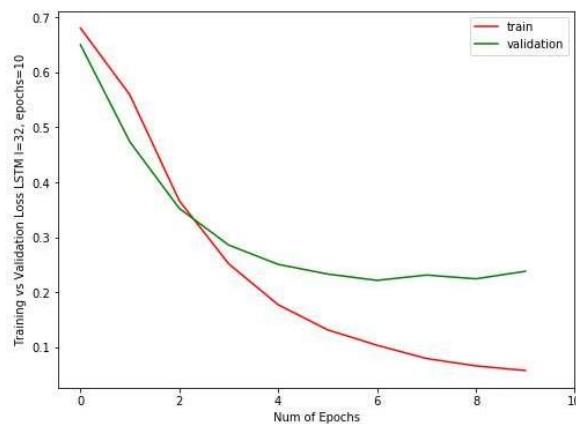


Figura 9 - Valor de perda entre as previsões.



Fonte: Elaborado pelos autores.

Para concluir a fase de análise, duas nuvens de palavras foram criadas a partir das palavras mais recorrentes no texto. No lado esquerdo da Figura 10 estão as palavras presentes nas sentenças classificadas como positivas, destacam-se as palavras: comida, todo, dia, chopp,

muita, cerveja, típica, alemã. No lado direito estão as palavras das sentenças classificadas como negativas, as palavras mais significativas foram: sábado, noite, não ir, fica lotado, fila, banheiro.

Figura 10 - No lado esquerdo, as 30+ Palavras em sentenças positivas. No direito, as 30+ Palavras em sentenças negativas.



Fonte: Elaborado pelos autores.

5 CONCLUSÕES

O interesse no processo de mineração de conteúdo da Web e análise de sentimento das opiniões das pessoas sobre uma atração turística motivou a produção desse estudo. As opiniões foram coletadas do *site* TripAdvisor®. A atração escolhida foi a Oktoberfest Blumenau, uma festa típica alemã considerada um dos maiores eventos culturais do Brasil. Foram coletadas 565 opiniões disponíveis no site em outubro de 2019, processadas e transformadas no conjunto de dados para a classificação. Os textos das opiniões foram separados em sentenças e pré-classificadas manualmente de acordo com a polaridade positiva ou negativa.

Na etapa de avaliação da classificação foram utilizados o algoritmo do modelo probabilístico *Multinomial Naive Bayes* e o algoritmo do modelo de redes neurais *Word2Vec*. Os resultados da validação cruzada apresentaram acurácia e precisão superiores a 90% nos dois modelos. Apesar do tamanho do conjunto de dados e a classificação manual, este estudo serviu para a compreensão de alguns métodos utilizados na análise de sentimento de opiniões coletadas da Web.

Para trabalhos futuros, a proposta é realizar a classificação de sentimento automática utilizando recursos baseados em léxico no idioma português brasileiro. Encontrar dicionário

léxico em idiomas diferentes do inglês tem sido um grande desafio para os pesquisadores. Os arquivos do código e o conjunto de dados do projeto foram publicados no seguinte endereço: <https://github.com/marcrescencio/proj_sent_analysis>.

REFERÊNCIAS

- Aggarwal, C. C. (2018). Opinion Mining and Sentiment Analysis. *Machine Learning for Text*. Cham: Springer International Publishing, pp. 413–434.
- Aghdaie, M. H. (2017). Data mining group decision-making with FAHP: An application in supplier evaluation and segmentation. In: Emrouznejad, Ali; Ho, William (ed.). *Fuzzy Analytic Hierarchy Process*. CRC Press.
- Bhonde, R., Bhagwat, B., Ingulkar, S. and Pande, A. (2015). Sentiment Analysis Based on Dictionary Approach. *International Journal of Emerging Engineering Research and Technology*, v. 3, n. 1, pp. 51-55.
- Cambria, E., Poria, S., Gelbukh, A. and Thelwall, M. (2017). Sentiment Analysis Is a Big Suitcase. *IEEE Intelligent Systems*, v. 32, n. 6, pp. 74-80. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8267597&isnumber=8267591>>. Acesso em 17 out. 2019.
- Ceci, F., Leopoldo Goncalves, A. and Weber, R. (2016). A model for sentiment analysis based on ontology and cases. *IEEE Latin America Transactions*, vol. 14, no. 11, pp. 4560-4566.
- Devika, M. D., Sunitha, C. and Ganesh, A. (2016). Sentiment Analysis: A Comparative Study on Different Approaches. *Procedia Computer Science*, v. 87, pp. 44-49. Disponível em: <<https://doi.org/10.1016/j.procs.2016.05.124>>. Acesso em 17 out. 2019.
- Giatsoglou, M., Vozalis, M. G., Diamantaras, K., et al. (2017). Sentiment analysis leveraging emotions and word embeddings. *Expert Systems with Applications*, v. 69, pp. 214-224. Disponível em: <<https://doi.org/10.1016/j.eswa.2016.10.043>>. Acesso em 21 out. 2019.
- Hemmatian, F. and Sohrabi, M. K. (2019). A survey on classification techniques for opinion mining and sentiment analysis. *Artificial Intelligence Review*, v. 52, n. 3, pp. 1495-1545. Disponível em: <<https://doi.org/10.1007/s10462-017-9599-6>>. Acesso em 17 out. 2019.
- Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, v. 9, n. 8, pp. 1735-1780. Disponível em: <<https://doi.org/10.1162/neco.1997.9.8.1735>>. Acesso em 25 out 2019.
- Kambali, P. S., Suri, S. and Sagar, B. M. (2018). Distributed Representation of Words in Vector Space for Kannada Language. In *Proceedings 2018 3rd International Conference on Computational Systems and Information Technology for Sustainable Solutions, CSITSS 2018*, Bengaluru, India, pp. 54- 58. DOI: 10.1109/CSITSS.2018.8768761.
- Klösgen, W. (1996). Knowledge discovery in databases and data mining. In: Raś Z.W., and Michalewicz M. (eds). Foundations of Intelligent Systems. *Lecture Notes in Computer*

- Science*, v. 1079, Springer, Berlin, Heidelberg.
- Kodratoff, Y. (1999). Knowledge discovery in texts: A definition, and applications. In: Ras Z.W., and Skowron A. (eds) Foundations of Intelligent Systems. ISMIS 1999. *Lecture Notes in Computer Science*, v. 1609, Springer, Berlin, Heidelberg.
- Liu, B. (2015). *Sentiment Analysis*. Cambridge: Cambridge University Press.
- Manning, C. D., Raghavan, P. and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, pp. 234-265. Disponível em: <<https://nlp.stanford.edu/IR-book/html/htmledition/naive-bayes-text-classification-1.html>>. Acesso em 20 out. 2019.
- Medhat, W., Hassan, A. and Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, v. 5, n. 4, pp. 1093-1113. Disponível em: <<https://doi.org/10.1016/j.asej.2014.04.011>>. Acesso em 23 out 2019.
- Mikolov, T., Chen, K., Corrado, G. and Dean, J. (2013). Efficient estimation of word representations in vector space. In: *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*.
- Mitchell, R. (2018). *Web Scraping with Python, 2nd Edition*.
- Rong, Xin. (11 nov 2014). word2vec Parameter Learning Explained. arXiv. Disponível em: <<https://arxiv.org/pdf/1411.2738v3.pdf>>. Acesso em 24 out. 2019.
- Rossi, R. H. P. (18 sep 2019). Análise de sentimentos para o auxílio na gestão das cidades inteligentes. *Tese (Doutorado em Sistemas Digitais)*, Biblioteca Digital de Teses e Dissertações da Universidade de São Paulo.
- Rudkowsky, E., Haselmayer, M., Wastian, M., et al. (2018). More than Bags of Words: Sentiment Analysis with Word Embeddings. *Communication Methods and Measures*, v. 12, n. 2-3, pp. 140-157. Disponível em: <<https://doi.org/10.1080/19312458.2018.1455817>>. Acesso em 27 set. 2019.
- Singh, J., Singh, G. and Singh, R. (2017). Optimization of sentiment analysis using machine learning classifiers. *Human-centric Computing and Information Sciences*, pp. 7-32. Disponível em: <<https://hcis-journal.springeropen.com/track/pdf/10.1186/s13673-017-0116-3>>. Acesso em 18 out. 2019.
- Tang, D., Zhang, Z., He, Y., Lin, C. and Zhou, D. (2019). Hidden topic–emotion transition model for multi-level social emotion detection. *Knowledge-Based Systems*, v. 164, pp. 426-435. Disponível em: <<https://doi.org/10.1016/j.knosys.2018.11.014>>. Acesso em 27 set. 2019.
- Wang, X., Ding, C., Zheng, W. and Wu, M. (2017). Sentiment Analysis based on Specific Dictionary and Sentence Analysis. International Conference on Economics and Management, Education, Humanities and Social Sciences (EMEHSS). Disponível em: <<https://dx.doi.org/10.2991/emehss-17.2017.2>>. Acesso em 25 set. 2019.
- Zheng, L., Wang, H. and Gao, S. (2018). Sentimental feature selection for sentiment analysis of Chinese online reviews. *International Journal of Machine Learning and Cybernetics*, v. 9, n. 1, pp. 75-84. Disponível em: <<https://doi.org/10.1007/s13042-015-0347-4>>.