Toolbox de processamento de linguagem natural com *scikit-learn*: comparativo de algoritmos e técnicas de pré-processamento

Ewerton Carlos de Araujo Assis

***Resumo*— Técnicas de processamento de linguagem natural têm sido bastante utilizadas nos dias atuais a fim de automatizar a interação entre usuários e máquinas, a partir da "compreensão", ainda que limitada, de linguagem natural. Diferentes abordagens e técnicas têm sido utilizadas, dependendo do contexto e das características de interação e dos dados disponíveis. O presente trabalho tem por finalidade comparar modelos de mineração de dados para processamento de linguagem natural e métodos de pré-processamento de dados (textuais) para alimentar estes modelos. Os modelos utilizados são: *Naive Bayes* (NB), *Decision Tree* (DT) e *Support-Vector Machine* (SVM). As técnicas de pré-processamento envolvem o uso de *Count Vectorizer*, *tf–idf* e *N-grams*, com eliminação de *stopwords* em alguns experimentos. Os modelos e técnicas são comparados em termos de acurácia a partir da medida *F1 Score*. Duas bases de dados são utilizadas para trabalhar com a classificação de textos: (1) para realizar análise de sentimentos; e (2) para classificar produtos a partir de sua descrição.**

***Palavras-Chave*— Processamento de linguagem natural; classificação de dados textuais; classificação; mineração de dados.**

***Abstract*— Nowadays, natural language processing has been used to automatize the interaction between human users and machines, from a limited "comprehension" of natural languages by computer systems. Different approaches and techniques have been used to address the interaction between human users and machines, based on each context and its characteristics and the data available for that matter. The present work aims to compare data mining models and pre-processing techniques. The following models are used: Naive Bayes (NB), Decision Tree (DT), and Support-Vector Machine (SVM). The pre-processing techniques used are based on the Count Vectorizer, tf–idf and N-grams, eliminating stopwords for some experiments. The models and techniques are compared using the accuracy provided by the F1 Score. Two datasets are used to classify texts and provide experiment cases: (1) one for sentiment analysis; and (2) one for classifying products according to their description.**

***Keywords*— Natural language processing; textual data classification; classification; data mining.**

I. Introdução

Processamento de linguagem natural (PLN) [8] tem sido bastante utilizada nos dias atuais com a finalidade de analisar bases de dados textuais e extrair não apenas outros dados, mas também informação — ou até mesmo conhecimento, quando se usa modelos mais complexos, como aqueles baseados em *deep learning* [9]. Aplicações comuns que são construídas a partir de PLN são a classificação e tradução de textos; transcrição de áudios (*speech-to-text*); análise de sentimento; e simular a comunicação entre um agente automatizado (sistema computacional) com um agente humano (usuário) [9] [10].

A partir de duas bases de dados textuais, o presente trabalho tem por finalidade comparar três modelos de aprendizado supervisionada e o impacto que o pré-processamento tem na qualidade (acurácia) destes modelos. Os seguintes algoritmos são utilizados: *Decision Tree* (DT), *Naive Bayes* (NB) e *Support-Vector Machine* (SVM) [2] [7]. Estes são geralmente utilizados para realizar a classificação de dados a partir de classes pré-definidas (pela definição de aprendizado supervisionado). Para realizar o pré-processamento das bases de dados, técnicas baseadas em *Count Vectorizer*, **tf–idf** e *N-grams* são utilizadas, com a remoção de *stopwords* em parte dos experimentos. Por fim, a acurácia de cada algoritmo é medida a partir do cálculo do *F1 score*.

Os experimentos foram realizados a partir de um Python Notebook (Jupyter) que encontra-se disponível em `https://github.com/earaujoassis/ufabc-data-mining-project`. Os algoritmos (modelos de aprendizado supervisionado) utilizados neste projeto são aqueles encontrados através da biblioteca Python *scikit-learn* (`https://scikit-learn.org/`) [1].

Nas próximas sessões, as bases de dados e seus objetos são definidos; os parâmetros dos experimentos são estabelecidos; e uma análise dos resultados é apresentada. Em seguida, as conclusões a partir dos experimentos são delineadas.

II. Base de Dados

Duas bases de dados foram utilizadas para realizar os experimentos deste trabalho:

- base de dados textuais em português brasileiro com revisões (análises) de filmes do IMDb [6]. Os objetos nesta base de dados serão classificados como "positivo" ou "negativo" (classificação com base na análise de sentimentos). Estão disponíveis 49.459 objetos; e
- base de dados com a descrição de produtos e a classificação destes dentre quatro categorias: 'livro', 'game', 'maquiagem' e 'brinquedo'. Estão disponíveis 2.916 objetos.

III. Experimentos

As duas bases de dados foram analisadas utilizando os mesmos parâmetros para os experimentos. Cada um dos modelos

Ewerton Carlos de Araujo Assis, Centro de Matemática, Computação e Cognição (CMCC), Universidade Federal do ABC (UFABC), Pós-graduação em Ciência da Computação (mestrado). Santo André-SP. E-mail institucional: carlos.assis@ufabc.edu.br. E-mail pessoal: earaujoassis@gmail.com.

de aprendizado supervisionado foram configurados com as mesmas técnicas de pré-processamento, conforme a seguir:

- uso do *Count Vectorizer* [3], o qual tem por finalidade analisar cada objeto da base de dados e realizar a contagem de palavras que aparecem neste documento, criando assim um vetor de relevância das palavras no corpus (ou conjunto total dos objetos textuais da base de dados);
- uso do **tf–idf** (ou *term frequency–inverse document frequency*) [4], o qual, como o nome sugere, cria uma matriz de frequência de palavras no corpus. A frequência (ou peso) de um termo (palavra) é obtida a partir dos objetos e do corpus, e uma frequência inversa é realizada para dar maior relevância a termos únicos/distintos, pouco frequentes, no corpus. Desta forma, embora algumas palavras possam ser bastante comum no corpus (como a classe de artigos definidos ou indefinidos da língua portuguesa), estas palavras não terão grande peso ao ser completada a análise do corpus;
- uso em conjunto do *Count Vectorizer* com **tf–idf**; e
- *N-grams* [5] com as variações *1-gram*, *2-grams* e *3-gram*: basicamente, tendo por exemplo a frase ”o rato roeu a roupa do rei de roma”, o corpus será analisado tendo como parâmetro o número de variações de uma frase no corpus. Assim, no caso do *1-gram*, será feita a contagem básica de palavras; no caso do *2-grams*, será feita a contagem de frases com até duas palavras, respeitando a ordem em que as palavras aparecem, como em ”o rato”, ”rato roeu”, ”roeu a”, ”a roupa”, ”roupa do”, ”do rei” e assim por diante; no caso do *3-grams*, seguirá a mesma lógica.

Assim, temos um total de seis configurações possíveis para cada pré-processamento. Para cada uma delas, foi realizada a manutenção ou a remoção das *stopwords*. As *stopwords* são palavras comuns que não acrescentam sentido às frases por si só, tendo relevância apenas para a construção das frases: as palavras ”bom” e ”ruim” têm maior relevância para a análise de sentimentos, comparadas às palavras ”um” ou ”que”. Para cada modelo de aprendizado supervisionado, temos 12 experimentos de pré-processamento realizados. Como foram selecionados três modelos, temos um total de 36 experimentos para cada base de dados, totalizando 72 experimentos neste trabalho.

A análise de acurácia foi realizada utilizando a métrica *F1 score*, disponível no próprio *scikit-learn*. Esta métrica calcula quão assertivos os modelos de aprendizado foram ao classificar os objetos da base de dados, no grupo de teste, para cada classe do problema.

IV. Resultados

A seguir são apresentados os resultados para a classificação dos objetos, respeitando cada base de dados. Os gráficos a seguir são organizados da seguinte forma:

- para cada base de dados, e respeitando cada modelo de aprendizado (DT, NB e SVM), são apresentados dois gráficos: um referente à manutenção das *stopwords* e outro referente à remoção destas;
- nos gráficos, cada linha representa um tipo de experimento com pré-processamento, conforme a legenda a seguir:
  - `Experimento 1` ou pré-processamento com *Count Vectorizer*;
  - `Experimento 2` ou pré-processamento com **tf–idf**;
  - `Experimento 3` ou pré-processamento com *Count Vectorizer* e **tf–idf**;
  - `Experimento 4` ou pré-processamento com *unigram* (1-gram);
  - `Experimento 5` ou pré-processamento com *bigram* (2-gram); e
  - `Experimento 6` ou pré-processamento com *trigram* (3-gram).

A. Classificação de produtos

Os gráficos com os resultados dos modelos DT, NB e SVM, com manutenção e remoção das *stopwords*, para classificação de produtos a partir de sua descrição, são apresentados a seguir:

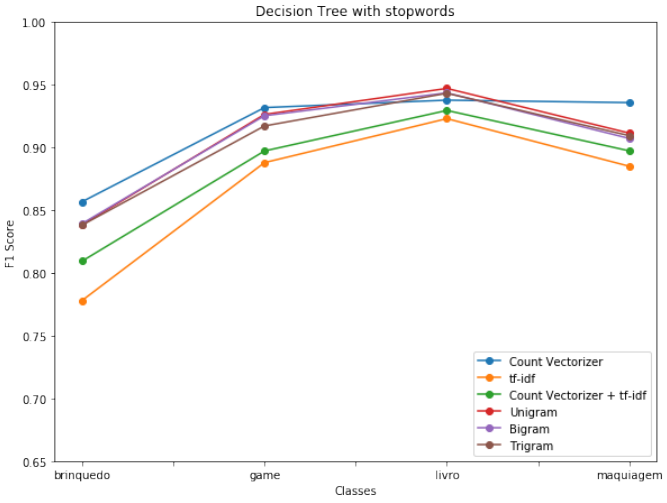

Fig. 1. Classificação de produtos com *Decision Tree* (DT) e manutenção das *stopwords*

A partir da análise dos gráficos das figuras 1, 2, 3, 4, 5 e 6, podemos verificar que a solução baseada no SVM e com o pré-processamento a partir do **tf–idf** (`Experimento 2`) obteve o melhor resultado. É interessante notar, na figura 3, o quanto o `Experimento 3`, utilizando *Count Vectorizer* e **tf–idf**, foi prejudicado. A remoção das *stopwords*, em geral, trouxe ganhos interessantes para a acurácia da solução, no caso desta base de dados.

B. Análise de sentimentos em revisões de filmes do IMDb

Os gráficos com os resultados dos modelos DT, NB e SVM, com manutenção e remoção das *stopwords*, para análise de sentimentos de revisões de filmes da base IMDb, são apresentados a seguir:

A partir da análise dos gráficos das figuras 7, 8, 9, 10, 11 e 12, podemos verificar que a solução baseada no SVM e NB,

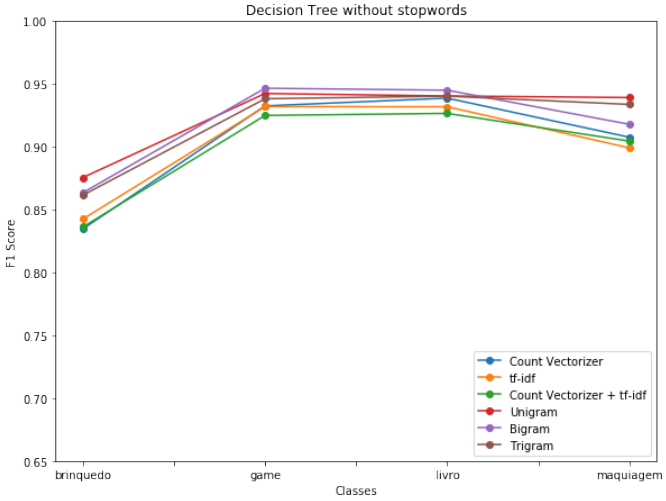

Fig. 2. Classificação de produtos com *Decision Tree* (DT) e remoção das *stopwords*

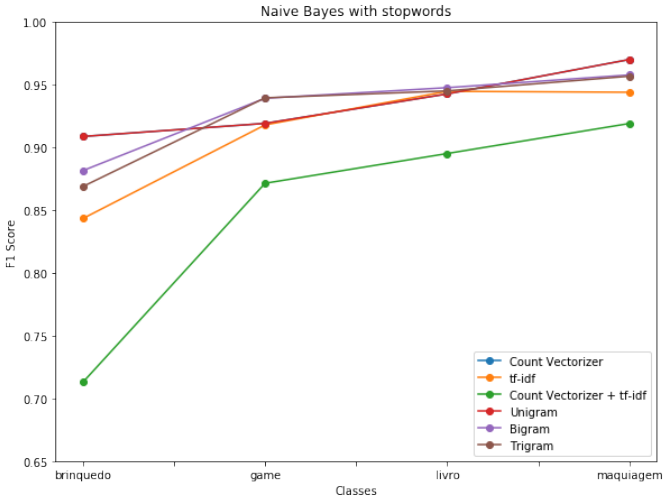

Fig. 3. Classificação de produtos com *Naive Bayes* (NB) e manutenção das *stopwords*

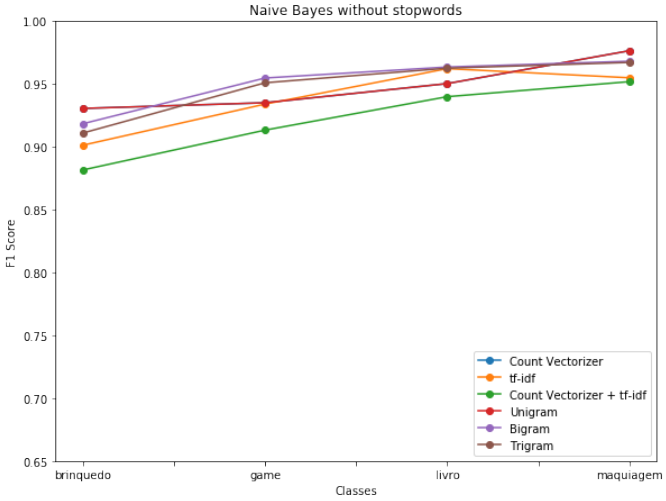

Fig. 4. Classificação de produtos com *Naive Bayes* (NB) e remoção das *stopwords*

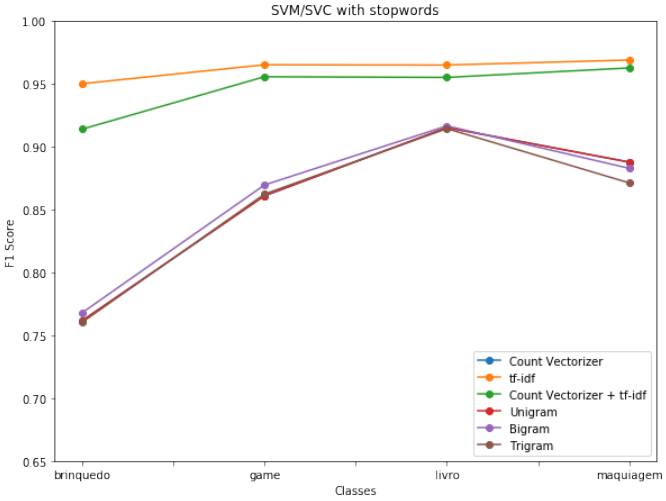

Fig. 5. Classificação de produtos com *Support-Vector Machine* (SVM) e manutenção das *stopwords*

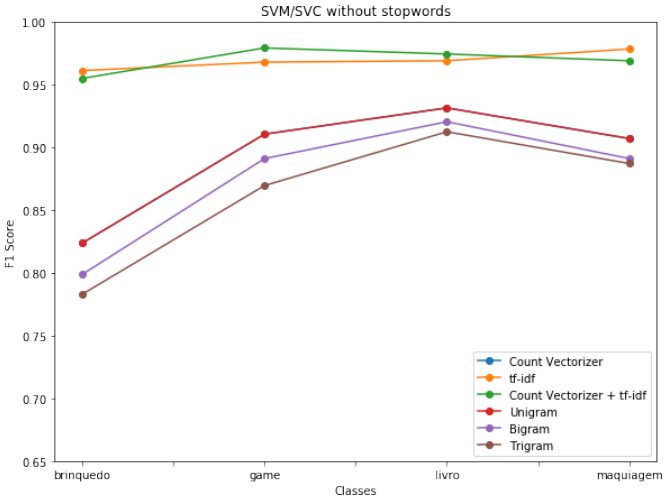

Fig. 6. Classificação de produtos com *Support-Vector Machine* (SVM) e remoção das *stopwords*

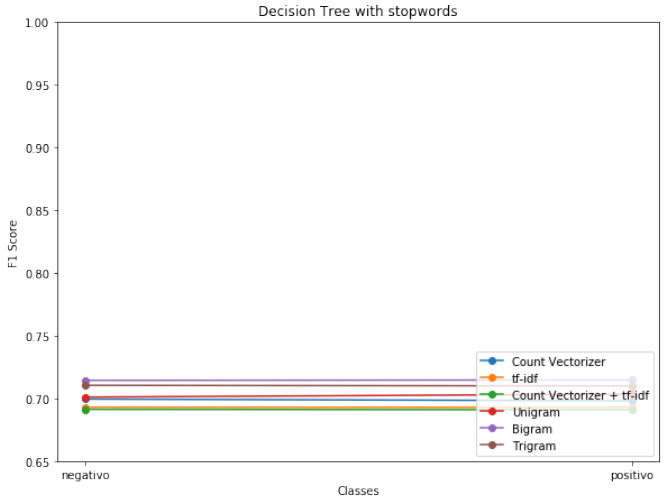

Fig. 7. Análise de sentimentos com *Decision Tree* (DT) e manutenção das *stopwords*

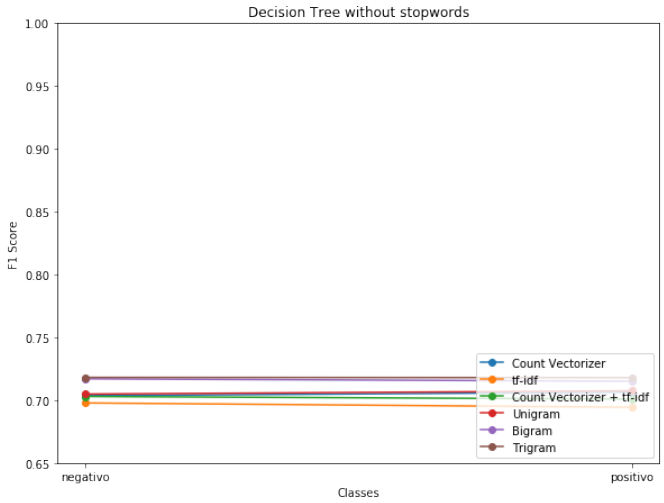

Fig. 8. Análise de sentimentos com *Decision Tree* (DT) e remoção das *stopwords*

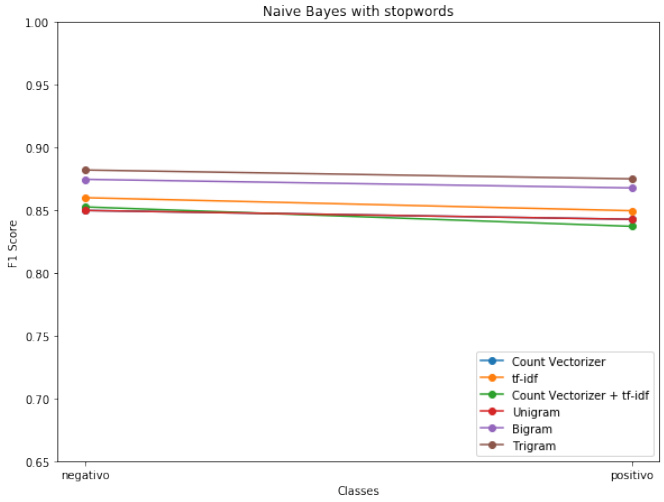

Fig. 9. Análise de sentimentos com *Naive Bayes* (NB) e manutenção das *stopwords*

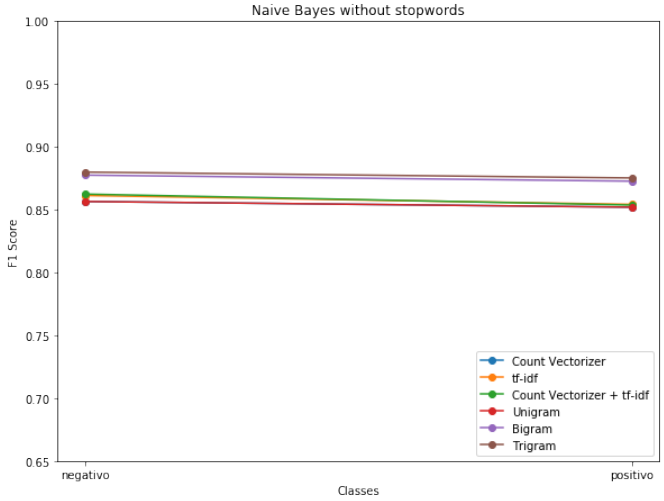

Fig. 10. Análise de sentimentos com *Naive Bayes* (NB) e remoção das *stopwords*

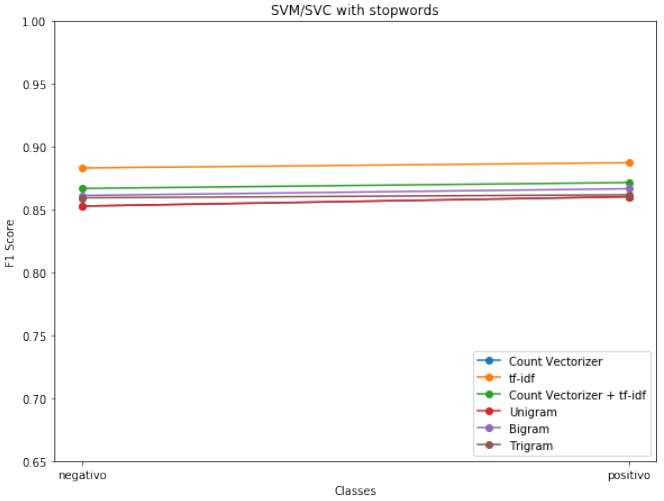

Fig. 11. Análise de sentimentos com *Support-Vector Machine* (SVM) e manutenção das *stopwords*

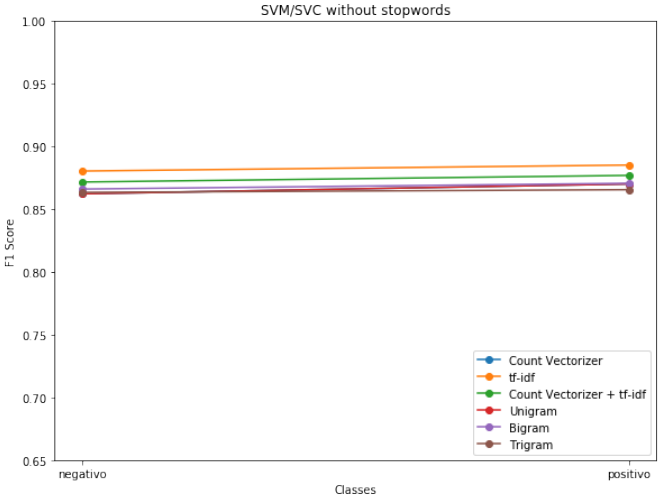

Fig. 12. Análise de sentimentos com *Support-Vector Machine* (SVM) e remoção das *stopwords*

com o pré-processamento a partir do **tf–idf** (`Experimento 2`), obtiveram os melhores resultados (scores acima de 85%). É interessante notar que experimentos diferentes tiveram resultados diferentes em cada modelo de aprendizado: nas figuras 7, 8, 9 e 10, os experimentos com *bigram* e *trigram* (`Experimento 5` e `Experimento 6`, respectivamente), proveram os melhores resultados; enquanto que nas figuras 11 e 12, os melhores experimentos foram aqueles utilizando **tf–idf**. A remoção das *stopwords*, no caso desta base de dados, não trouxeram ganhos relevantes para a acurácia da solução, ainda que a remoção mostre-se ligeiramente melhor nos gráficos.

V. Conclusões

A partir dos 72 experimentos realizados em ambas as bases de dados, tendo por base a análise dos resultados, podemos afirmar que, em geral, o pré-processamento com **tf–idf** e seu uso no modelo baseado em SVM promoveram

uma acurácia significativa para classificar objetos textuais. No entanto, os resultados não permitiram distinguir um único método de pré-processamento como a melhor alternativa para processamento de linguagem natural: os modelos de aprendizado terão, também, impacto significativo nos resultados obtidos, bem como qual método está sendo utilizado para um determinado modelo. A remoção das *stopwords*, em geral, mostra-se como um aspecto importante para obter-se bons resultados ao realizar a classificação dos textos.

Os resultados corroboram para a relevância de se avaliar as características de cada base de dados, como estes objetos da base devem ser pré-processados e quais modelos serão melhor ajustados para as características que emergem da base de dados e do pré-processamento.

REFERÊNCIAS

[1] *scikit-learn: machine learning in Python*. Disponível em: `https://scikit-learn.org`. Último acesso em 27 de Agosto de 2019.

[2] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, 2006.

[3] C. Manning, P. Raghavan e H. Schütze. "Scoring, term weighting, and the vector space model". In *Introduction to Information Retrieval*, p. 100-123. Cambridge University Press, 2008.

[4] A. Rajaraman e J. Ullman. "Data Mining". In *Mining of Massive Datasets*, p. 1-17. Cambridge University Press, 2011.

[5] A. Z. Broder, S. C. Glassman, M. S. Manasse e G. Zweig. "Syntactic clustering of the Web". In *Computer Networks and ISDN Systems*, p. 1157-1166, 29:8. 1997.

[6] *IMDb: Ratings and Reviews for New Movies and TV Shows*. Disponível em: `https://www.imdb.com/`. Último acesso em 27 de Agosto de 2019.

[7] S. Raschka e V. Mirjalili. *Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow*. 2ed. Packt, 2017.

[8] S. Bird, E. Klein e E. Loper. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media, 2009.

[9] D. L. Yse. *Your Guide to Natural Language Processing (NLP)*. Disponível em: `https://towardsdatascience.com/your-guide-to-natural-language-processing-nlp-48ea2511f6e1`. Último acesso em 27 de Agosto de 2019.

[10] S. Quarteroni. "Natural Language Processing for Industry". In *Informatik-Spektrum*, p. 105-112, 41:2. 2018.