Análise de Sentimentos em Críticas de Filmes por meio de Algoritmos Clássicos de Aprendizado de Máquina

Ana Júlia de Oliveira Bellini

Orientadora: Profa. Drª. Lilian Berton

Trabalho de Conclusão de Curso II

Bacharelado em Ciência da Computação

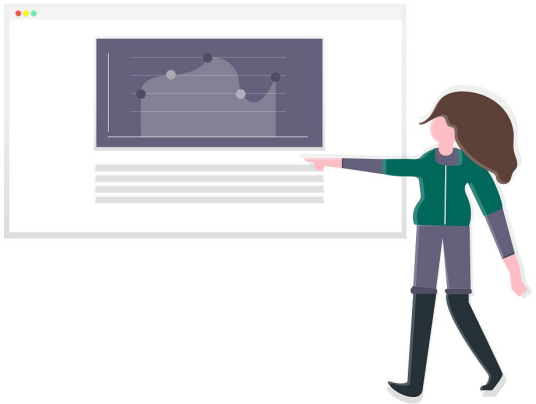
Universidade Federal de São Paulo

05 de Dezembro de 2019

Roteiro

- Introdução;
- Análise de sentimentos;
- Processamento e representação de texto;
- Aprendizado de máquina;
- Algoritmos de classificação;
- Metodologia;
- Resultados;
- Discussões;
- Considerações finais.

Introdução



Motivação

- Crescimento no acesso à Internet e uso de *smartphones*;
- Consumidores buscam opiniões sobre produtos e lugares nas mais diversas fontes *online*;
- Grande impacto na decisão de compra e em outros posicionamentos do cliente.

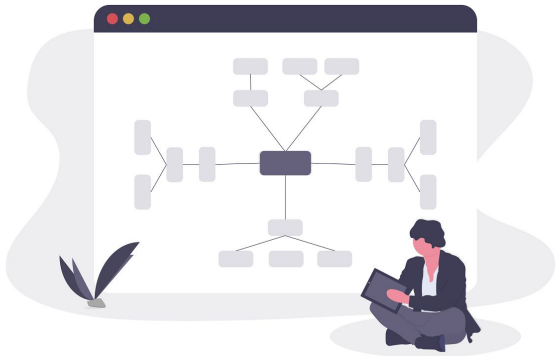
Análise de Filmes

- *Sites* especializados em críticas de filmes - IMDb [1], *Rotten Tomatoes* [2] -, além de redes sociais no geral;
- Espectadores procuram por *reviews* de críticos ou outros usuários para decidir se assistem, ou não, a um determinado filme;
- *Feedback* importante para produtoras;
- Aplicação de análise de sentimentos sobre opiniões.

Objetivos

- Estudar e aplicar análise de sentimentos em conjuntos de dados contendo críticas de filmes, utilizando:
  - Técnicas de processamento de texto;
  - Algoritmos de aprendizado de máquina para classificação de dados;
- Analisar o desempenho para fins de comparação entre as técnicas implementadas.

Fundamentação Teórica



> Análise de sentimentos é o estudo computacional de opiniões, avaliações e emoções das pessoas, em relação a outros indivíduos, eventos e produtos [3].

Análise de Sentimentos

- Subárea do processamento de linguagem natural;
- Trabalhos precursores dos anos 1970;
- Mais pesquisada a partir dos anos 2000.

- Aplicações:
  - Sistemas de recomendação;
  - Opinião de cidadãos sobre propostas de governo;
  - Inteligência de negócios.

- Desafios:
  - Conjunto base de palavras;
  - Pontos positivos e negativos juntos;
  - Fatos + opiniões;
  - Ironia e sarcasmo [4].

Processamento de Textos

- Algoritmos de classificação geram modelos matemáticos;
  - Textos precisam ser representados matematicamente.
- Diversas técnicas para processar textos antes da classificação do sentimento dos mesmos.

Técnicas de Processamento

Caixa baixa e pontuações

Se uma palavra aparecer em caixa baixa e alta, o algoritmo pode entender que são duas palavras distintas. Por isso, todas são convertidas para caixa baixa.

Stop words

Palavras sem significado intrínseco podem confundir o classificador (pronomes, artigos, conjunções, entre outras), sendo importante sua remoção.

Lematização

Palavras flexionadas são convertidas em sua forma básica (exemplo: “vendido” passa a ser “vender”).

Técnicas de Processamento

Stemming

Palavras reduzidas ao seu radical, removendo seus sufixos (exemplo: “vendido” e “vender” passam a ser “vend”).

POS Tagging

Cada palavra recebe uma *tag* morfossintática de acordo com o contexto (“vend” recebe *tag* de verbo, “meia” recebe *tag* de numeral).

Representando Textos

- Textos representados como vetores:
  - Cada palavra é uma posição;
  - Valores podem ser dados pela frequência das palavras no texto ou em todo o conjunto de dados.

Representando Textos

- *Bag-of-Words*: considera a frequência de cada termo dentro de uma crítica (*Term Frequency - TF*).
  - TF = total de ocorrências da palavra / total de palavras.
- *Term Frequency-Inverse Document Frequency* (TF-IDF): normalização do vetor *Bag-of-Words*.
  - TF-IDF = TF * ln (número de textos / número de textos com a palavra).

no meio do caminho tinha uma pedra
tinha uma pedra no meio do caminho
tinha uma pedra
no meio do caminho tinha uma pedra

nunca me esquecerei desse acontecimento
na vida de minhas retinas tão fatigadas
nunca me esquecerei que no meio do caminho
tinha uma pedra
tinha uma pedra no meio do caminho
no meio do caminho tinha uma pedra

mei	caminh	pedr	nunc	esquec	dess	acontec	vid	retin	tão	fatig
6	6	7	1	1	1	1	1	1	1	1

Exemplo de transformação de texto em representação por *Bag-of-Words*

Aprendizado de Máquina

- Indução de hipóteses ou funções, a partir de experiências passadas, para problemas de descrição de objetos ou previsão de valores [5].
- Tipos de aprendizado:
  - Supervisionado;
  - Não supervisionado;
  - Semissupervisionado;
  - Por reforço.

Classificação de Dados

- Problema onde aprendemos/prevemos a classe de um determinado objeto, com base em exemplos já classificados/rotulados (aprendizado supervisionado);
- Feita com duas (“positiva” e “negativa”) ou mais classes.
- Pode ser utilizada para analisar a polaridade dos sentimentos.

Técnicas de Classificação

Naïve Bayes

Baseado no Teorema de Bayes, onde o exemplo é rotulado com a classe de maior probabilidade.

Support Vector Machine (SVM)

Objetos mapeados em *N* dimensões, onde cada classe é separada por um hiperplano, e o exemplo é rotulado de acordo com sua posição neste espaço.

Árvore de Decisão

Nós intermediários realizam testes sobre alguns dos atributos mais significativos, retornando uma decisão sobre a classificação em suas folhas.

Técnicas de Classificação

Random Forest

Coleção de árvores de decisão, onde a classificação do exemplo é dada pela classe mais votada entre todas as árvores.

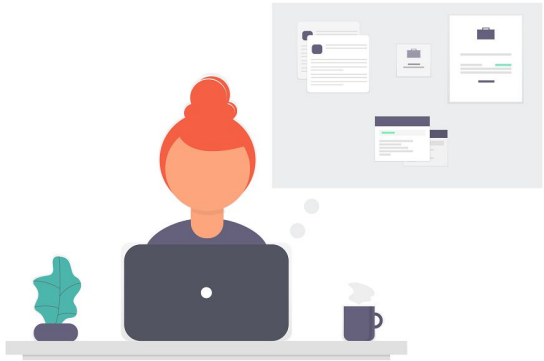
k-Nearest Neighbours (k-NN)

Exemplo é classificado de acordo com os *k* vizinhos mais próximos; a classe predominante entre eles é a escolhida.

Multilayer Perceptron (MLP)

Rede neural com várias camadas, que ajusta seus pesos sinápticos para aprender com o conjunto de dados fornecido.

Metodologia



Conjuntos de Dados

- Para treino e teste dos modelos de classificação, foram selecionados dois conjuntos de dados, com críticas retiradas de *sites* especializados.

Rotten Tomatoes

Conjunto de dados com 55 mil críticas, divididas entre positivas e negativas.

IMDb

Dados coletados por Maas et al. (2011), com 25 mil críticas, também classificadas como positivas e negativas.

Bibliotecas

Matplotlib

Plotagem de gráficos 2D, para visualização dos dados na análise exploratória.

Natural Language Toolkit (NLTK)

Processamento de texto em linguagem humana.

NumPy

Uso de arranjos, vetores e matrizes de *N* dimensões.

Pandas

Estruturas de dados específicas e operações de manipulação para análise de dados.

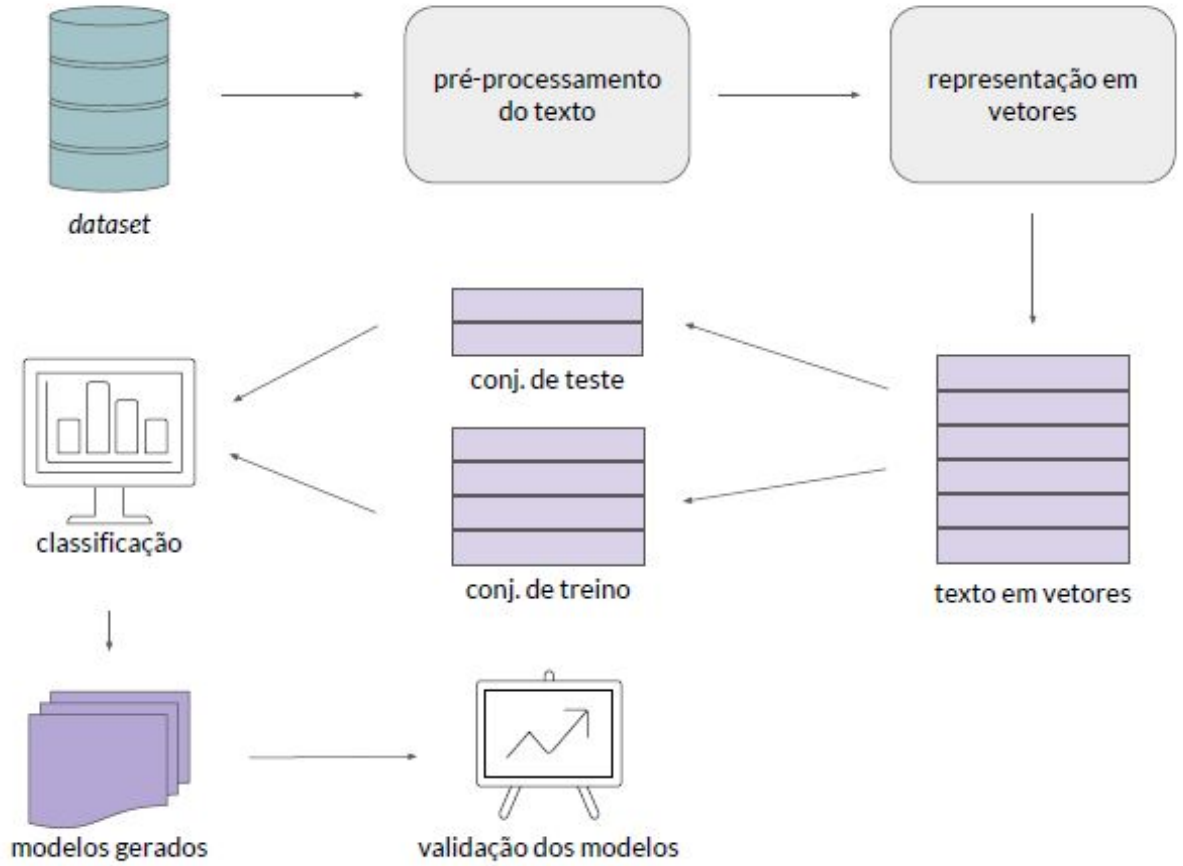
Scikit-learn

Aprendizado de máquina, mineração e análise de dados, com métodos para implementar os algoritmos vistos.

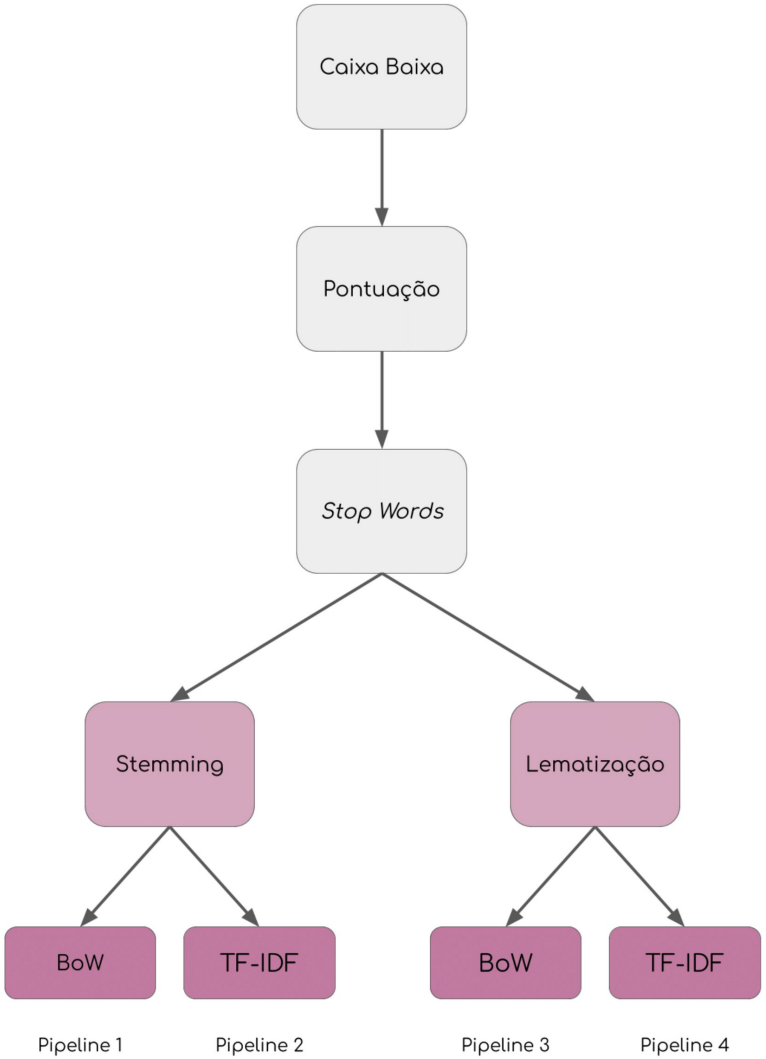
Etapas


1
Processamento dos Textos
Aplicação das técnicas, junto com representação em vetores.
2
Classificação
Vetores utilizados para classificar polaridade de cada crítica, em cada algoritmo.
3
Validação
Análise de desempenho dos algoritmos e comparação entre os mesmos.

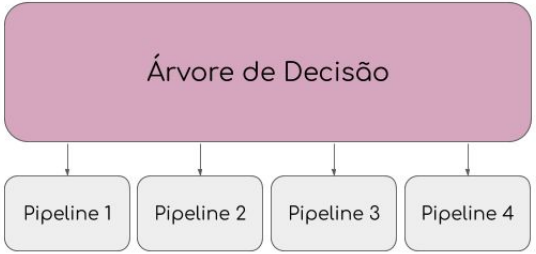
Fluxo de Trabalho

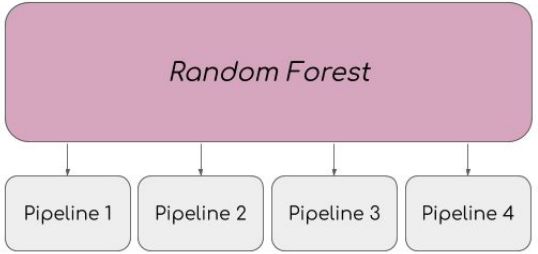


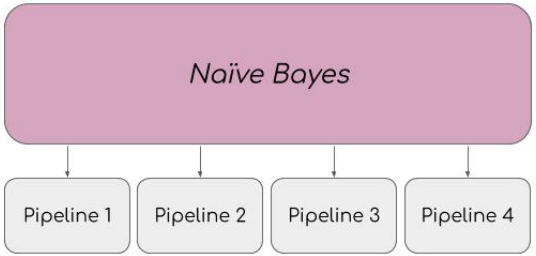
Processando os Textos

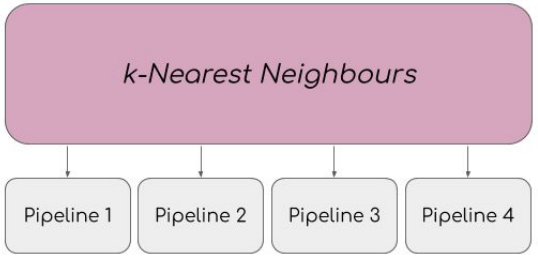


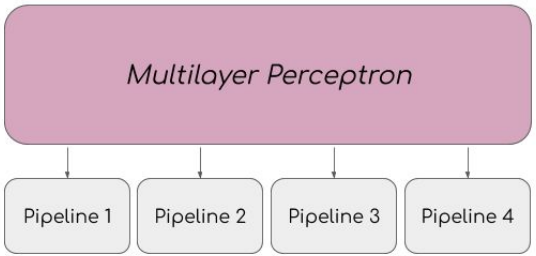
Modelos

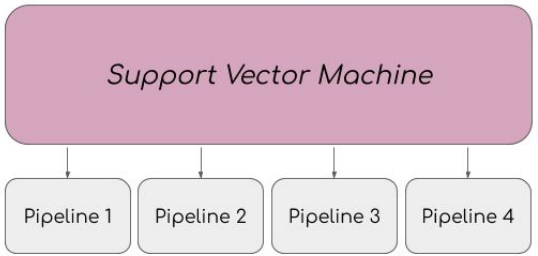

Árvore de Decisão
Pipeline 1
Pipeline 2
Pipeline 3
Pipeline 4


Random Forest
Pipeline 1
Pipeline 2
Pipeline 3
Pipeline 4


Naïve Bayes
Pipeline 1
Pipeline 2
Pipeline 3
Pipeline 4


k-Nearest Neighbours
Pipeline 1
Pipeline 2
Pipeline 3
Pipeline 4


Multilayer Perceptron
Pipeline 1
Pipeline 2
Pipeline 3
Pipeline 4


Support Vector Machine
Pipeline 1
Pipeline 2
Pipeline 3
Pipeline 4

Métricas

- *k-Fold cross validation*;
- Matriz de confusão;
- **Acurácia;**
- Precision;
- Recall;
- **F1 Score.**

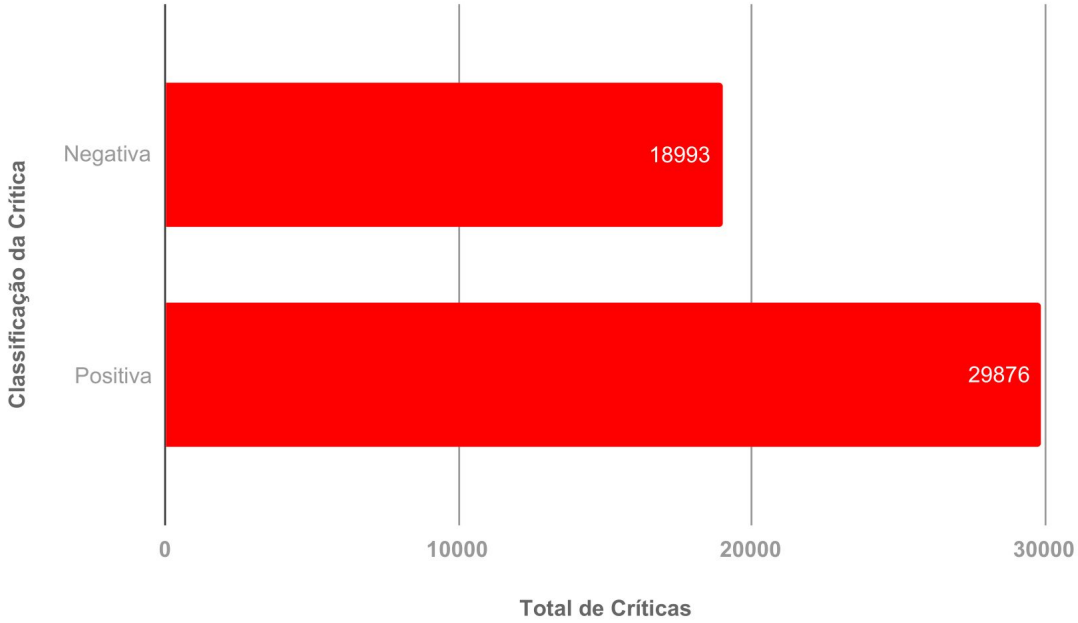
Resultados

Análise Exploratória

- Análise da distribuição de classes.
- Foram geradas nuvens de palavras, para verificar os termos mais usados em cada *site*.
- Por fim, foi feita uma análise da frequência dos termos contidos em cada conjunto de dados.

Rotten Tomatoes

Conjunto desbalanceado, com cerca de 60% de exemplos positivos e 40% negativos.

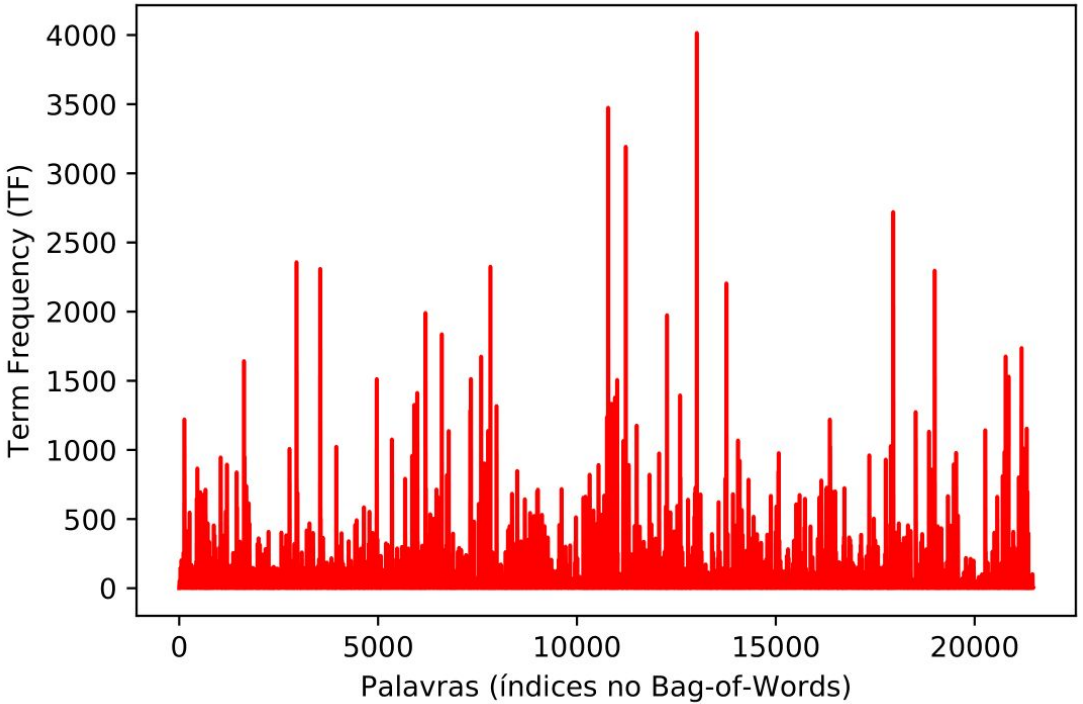


Rotten Tomatoes

Foco nas performances, enredo, gênero e emoções dos espectadores durante o filme.

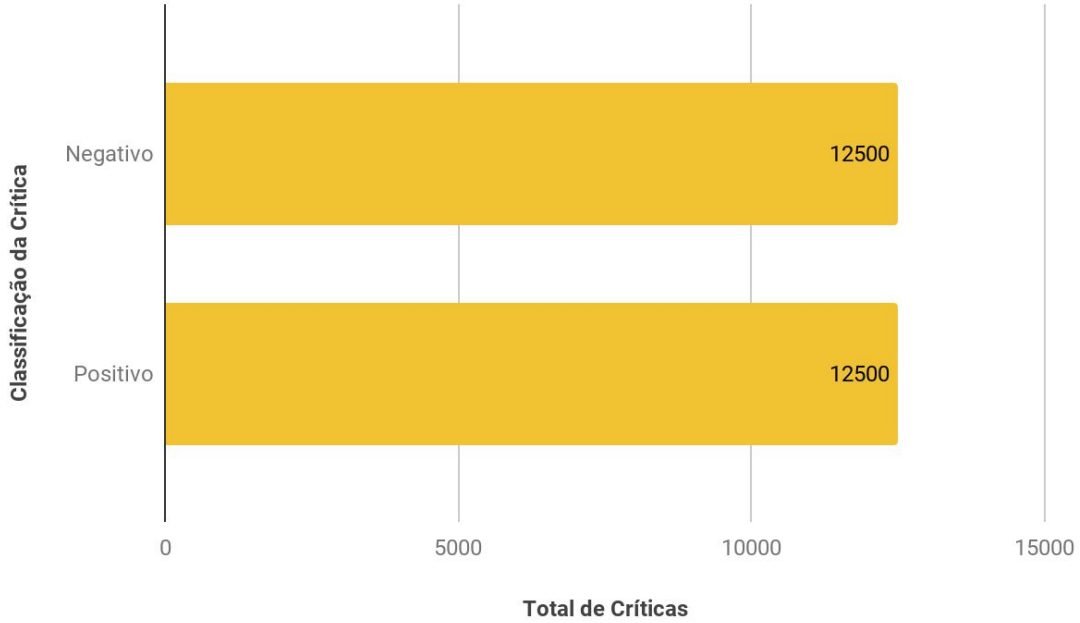
Rotten Tomatoes

A maior parte das palavras têm frequência mais baixa, e são as mais importantes [6].



IMDb

Ambas as classes possuem a mesma quantidade de críticas.

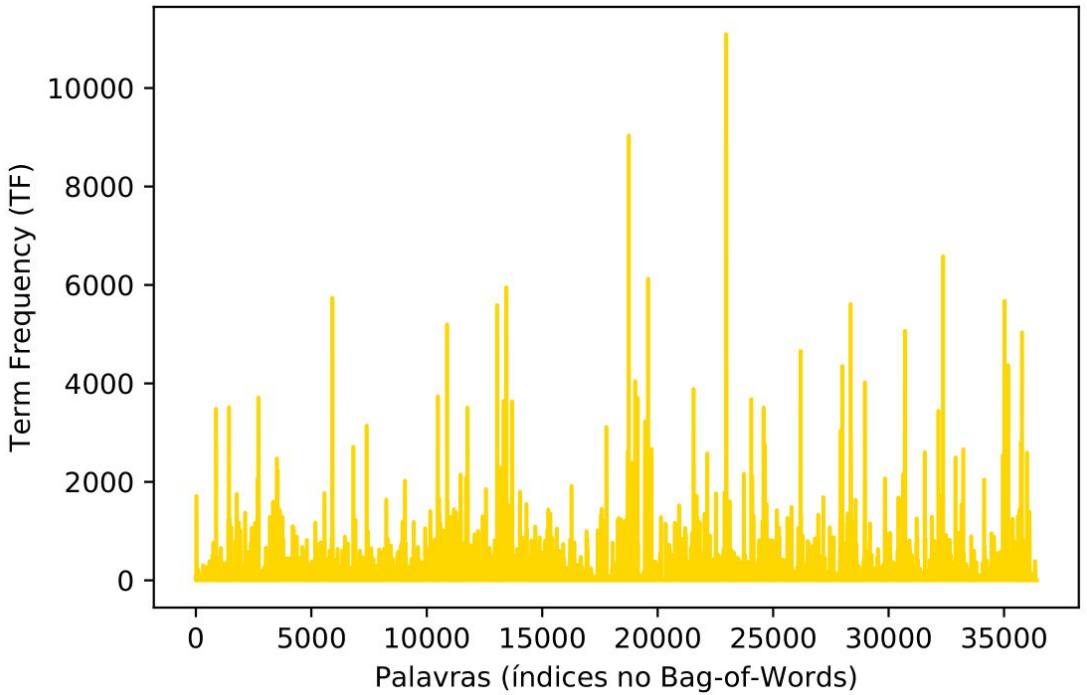


IMDb

Neste caso, foca-se mais em enredo do filme e nas atuações.

IMDb

Assim como para o *Rotten Tomatoes*, a maioria das palavras apresentam frequências menores.

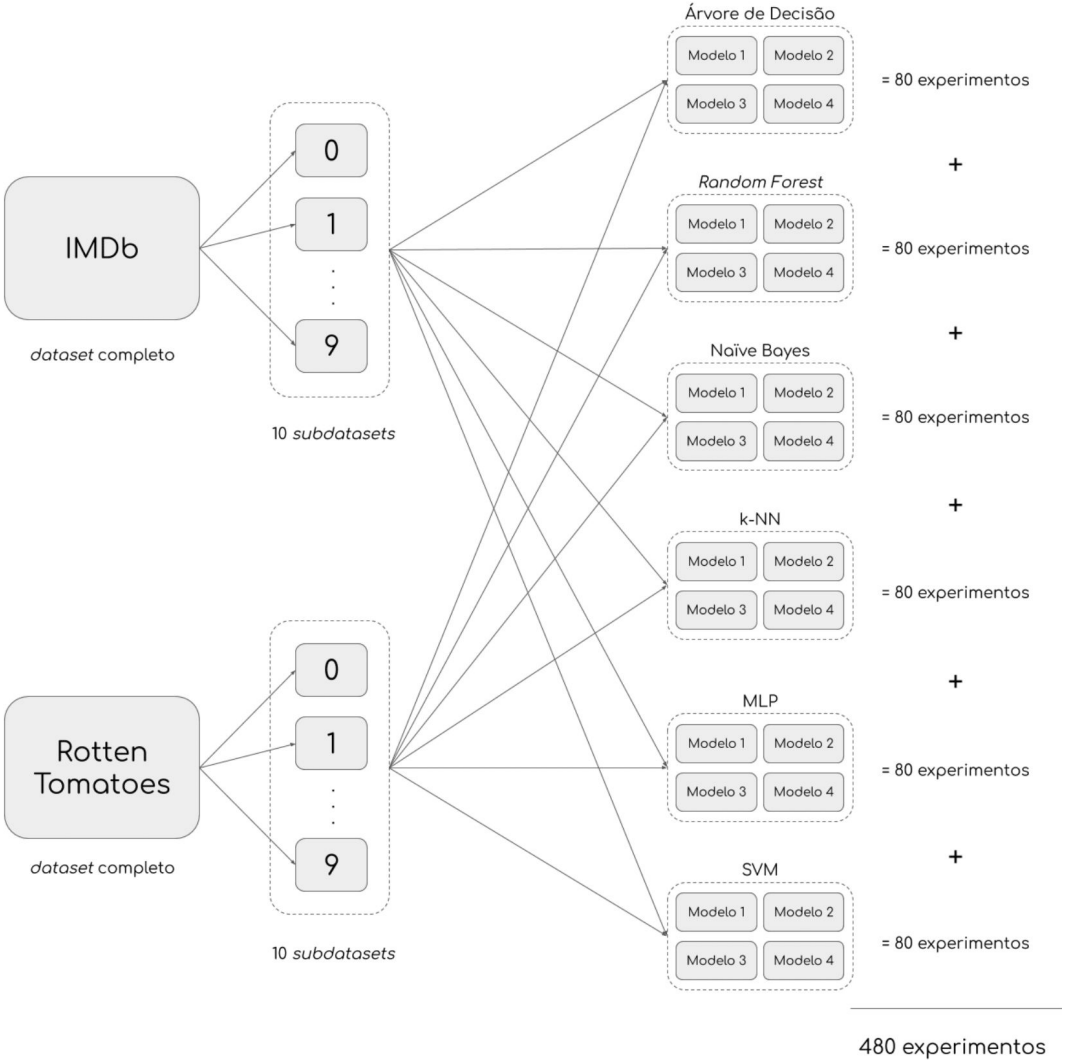


Experimentos

- Constatou-se que os dois conjuntos de dados possuíam muitos atributos, gerando custo computacional muito alto:
  - IMDb - cerca de 72 mil.
  - *Rotten Tomatoes* - aproximadamente 24,5 mil.
- Cada um deles foi dividido em 10 subconjuntos (não necessariamente disjuntos).
  - 10 mil exemplos selecionados aleatoriamente.

Experimentos

- Todos os 24 modelos foram submetidos a **10 experimentos com cada *dataset***, cada um sendo feito com um subconjunto diferente.
- Total de experimentos = **480** (com mais 20 extras).
- Resultados expressos como **média aritmética dos dez experimentos.**


IMDb
dataset completo
0
1
9
10 subdatasets
Rotten Tomatoes
dataset completo
0
1
9
10 subdatasets
Árvore de Decisão
Modelo 1
Modelo 2
Modelo 3
Modelo 4
= 80 experimentos
+
Random Forest
Modelo 1
Modelo 2
Modelo 3
Modelo 4
= 80 experimentos
+
Naïve Bayes
Modelo 1
Modelo 2
Modelo 3
Modelo 4
= 80 experimentos
+
k-NN
Modelo 1
Modelo 2
Modelo 3
Modelo 4
= 80 experimentos
+
MLP
Modelo 1
Modelo 2
Modelo 3
Modelo 4
= 80 experimentos
+
SVM
Modelo 1
Modelo 2
Modelo 3
Modelo 4
= 80 experimentos
480 experimentos

Naïve Bayes

Resultados semelhantes entre os quatro modelos.

IMDb		
Pipeline	*Acurácia*	*F1 Score*
Stemming + BoW	0,8402	0,8355
Stemming + TF-IDF	0,8463	**0,8406**
Lemat. + BoW	0,8403	0,8353
Lemat. + TF-IDF	0,8471	0,8405

Rotten Tomatoes		
Pipeline	*Acurácia*	*F1 Score*
Stemming + BoW	0,7489	**0,8022**
Stemming + TF-IDF	0,7064	0,7993
Lemat. + BoW	0,7478	0,8013
Lemat. + TF-IDF	0,7012	0,7971

SVM

A função *kernel* linear foi a mais adequada ao *dataset* do IMDb. Descarte de TFs baixos afetou negativamente o aprendizado.

IMDb, Kernel Radial			
TFs Descartados	*Acurácia*	*F1 Score*	*Atributos*
TF <= 4	0,5347	**0,3100**	13240,2
TF <= 5	0,5356	0,3094	12083,1

IMDb, Kernel Linear		
Pipeline	*Acurácia*	*F1 Score*
Stemming + BoW	0,8345	0,8344
Stemming + TF-IDF	0,8729	**0,8740**
Lemat. + BoW	0,8356	0,8356
Lemat. + TF-IDF	0,8726	0,8738

SVM

Modelo com *Stemming* e representação por *Bag-of-Words* foi destaque.

Rotten Tomatoes		
Pipeline	*Acurácia*	*F1 Score*
Stemming + BoW	0,7238	**0,7980**
Stemming + TF-IDF	0,6590	0,7783
Lemat. + BoW	0,7190	0,7956
Lemat. + TF-IDF	0,6498	0,7740

Árvore de Decisão

Resultados bem próximos, com *Bag-of-Words* sendo o melhor entre eles.

IMDb		
Pipeline	*Acurácia*	*F1 Score*
Stemming + BoW	0,7110	**0,7107**
Stemming + TF-IDF	0,7050	0,7038
Lemat. + BoW	0,7081	0,7074
Lemat. + TF-IDF	0,7052	0,7041

Rotten Tomatoes		
Pipeline	*Acurácia*	*F1 Score*
Stemming + BoW	0,6434	**0,7073**
Stemming + TF-IDF	0,6375	0,7044
Lemat. + BoW	0,6373	0,7027
Lemat. + TF-IDF	0,6295	0,6984

Random Forest

Resultados semelhantes para o IMDb. Com o *Rotten Tomatoes*, a representação por TF-IDF trouxe uma pequena melhora.

IMDb		
Pipeline	*Acurácia*	*F1 Score*
Stemming + BoW	0,7481	**0,7271**
Stemming + TF-IDF	0,7431	0,7205
Lemat. + BoW	0,7487	**0,7271**
Lemat. + TF-IDF	0,7436	0,7213

Rotten Tomatoes		
Pipeline	*Acurácia*	*F1 Score*
Stemming + BoW	0,6840	0,7456
Stemming + TF-IDF	0,6824	**0,7514**
Lemat. + BoW	0,6758	0,7395
Lemat. + TF-IDF	0,6743	0,7462

k-NN

A representação por TF-IDF foi a mais adequada, em ambos os conjuntos de dados.

IMDb		
Pipeline	*Acurácia*	*F1 Score*
Stemming + BoW	0,6244	0,6267
Stemming + TF-IDF	0,7363	**0,7461**
Lemat. + BoW	0,6281	0,6214
Lemat. + TF-IDF	0,7349	0,7425

Rotten Tomatoes		
Pipeline	*Acurácia*	*F1 Score*
Stemming + BoW	0,5867	0,6477
Stemming + TF-IDF	0,6624	**0,7197**
Lemat. + BoW	0,5775	0,6317
Lemat. + TF-IDF	0,6634	0,7189

MLP

Para o IMDb, este algoritmo levou a alguns dos melhores resultados dentre todos os experimentos.

IMDb		
Pipeline	*Acurácia*	*F1 Score*
Stemming + BoW	0,8600	0,8608
Stemming + TF-IDF	0,8610	0,8615
Lemat. + BoW	0,8629	0,8634
Lemat. + TF-IDF	0,8632	**0,8636**

Rotten Tomatoes		
Pipeline	*Acurácia*	*F1 Score*
Stemming + BoW	0,7192	**0,7735**
Stemming + TF-IDF	0,7101	0,7676
Lemat. + BoW	0,7172	0,7717
Lemat. + TF-IDF	0,7113	0,7696

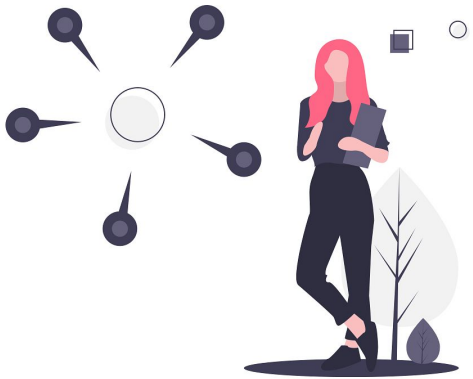
Discussão dos Resultados

- Representação de textos:
  - TF-IDF levou a métricas mais altas.
  - Para o IMDb, o *Bag-of-Words* foi mais adequado. Já para o *Rotten Tomatoes*, o melhor foi o TF-IDF.

Discussão dos Resultados

- Classificadores: SVM e MLP atingiram os melhores resultados, entre os seis experimentados.
- *Datasets*: acurácias maiores com o IMDb (balanceado).
- Quantidade de atributos: acurácias não tão distantes, entre um conjunto de dados e outro.

Considerações Finais



Referências

[1] IMDb. Disponível em: <https://www.imdb.com/>. Acesso em: 25 jun. 2019.

[2] Rotten Tomatoes. Disponível em: <https://www.rottentomatoes.com/>. Acesso em: 25 jun. 2019.

[3] LIU, B.; ZHANG, L. A Survey of Opinion Mining and Sentiment Analysis. In: AGGARWAL, C. C.; ZHAI, C. (Ed.). Mining Text Data. Boston, MA: Springer US, 2012. p. 415–463. ISBN 978-1-4614-3223-4. Disponível em: <http://www.cs.unibo.it/~montesi/CBD/Articoli/SurveyOpinionMining.pdf>. Acesso em: 25 jun. 2019.

[4] PANG, B.; LEE, L.; VAITHYANATHAN, S. Thumbs Up?: Sentiment Classification Using Machine Learning Techniques. In: Proceedings of the ACL-02 conference on Empirical methods in natural language processing - EMNLP '02. Morristown, NJ, USA: Association for Computational Linguistics, 2002. v. 10, n. July, p. 79–86. ISSN 0003-5696. Disponível em: <http://portal.acm.org/citation.cfm?doid=1118693.1118704>. Acesso em: 25 jun. 2019.

[5] FACELI, K. et al. Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina. Rio de Janeiro: LTC, 2011. 394 p. ISBN 9788521618805.

[6] RUNESON, P.; ALEXANDERSSON, M.; NYHOLM, O. Detection of duplicate defect reports using natural language processing. In: Proceedings of the 29th International Conference on Software Engineering. Washington, DC, USA: IEEE Computer Society, 2007. (ICSE '07), p. 499–510. ISBN 0-7695-2828-7. Disponível em: <https://doi.org/10.1109/ICSE.2007.32>. Acesso em: 04 dez. 2019.

Obrigada!

Alguma dúvida?

Contato:

- anajuliabell@gmail.com
- ana.bellini29@unifesp.br