

Relatório 3 IA: *Machine Learning*

Erick Grilo¹, Max Fratane¹, Vitor Araujo¹, Vítor Lourenço¹

¹Instituto de Computação – Universidade Federal Fluminense (UFF)
Niterói, Rio de Janeiro – Brazil

{simas-grilo,mfratane,vitoraraujo,vitorlourenco}@id.uff.br

Resumo.

1. Introdução

O que é pensado pelas pessoas sempre foi uma informação importante para seres humanos para o processo de tomada de decisão. Com o advento da *World Wide Web*, cresceu o acesso à quantidade de opiniões e experiências sobre determinados assuntos que são de pessoas que não conhecemos e nem são profissionais especialistas no assunto. Dessa forma, é possível obter informações de pessoas com os mais variados sentimentos acerca de algum assunto.

Nesse espectro, surge a área de análise de sentimentos (ou mineração de opiniões), que é responsável por fazer o processamento de linguagem natural, usando técnicas de análise textual e linguística computacional a fim de identificar, extrair e estudar opiniões, estados afetivos e informação subjetiva. Dessa forma, é possível extrair opiniões de consumidores acerca de um determinado produto, por exemplo. Tal mineração é extremamente útil, pois como é visto em [Pang et al. 2008], influencia bastante em tópicos como a aquisição de serviços: a cada 2000 americanos, dentre os leitores de resenhas on-line de restaurantes, hotéis e outros serviços, como viagens, escolas, médicos e cursos, de 73% à 87% dos entrevistados disseram que tais resenhas tiveram uma influência significativa na aquisição desses serviços [Zhu and Zhang 2010].

Tal abordagem também é útil para outras finalidades: além da compra de serviços e produtos, as revisões de outros usuários online também são úteis na busca de opiniões políticas (tanto acerca de empresas e organizações quanto acerca de políticos): muitas pessoas buscam atualmente informações de outras acerca de políticos, por exemplo, para confirmar se a opinião dele é condizente com a sua, ou até mesmo buscam na internet opiniões que divergem das suas a fim de enriquecer o debate [Gil de Zúñiga et al. 2009].

Com o advento de plataformas na *web*, tais como blogs, fóruns de discussão, redes *peer-to-peer* e outros tipos de *social media*, tais como o *Facebook* e o *Twitter*. consumidores têm uma quantidade de informação e uma facilidade de expor sua opinião sem precedentes, sejam elas negativas ou positivas. sobre qualquer produto ou serviço. Nesse âmbito, grandes companhias (bancos, restaurantes, agências de viagem, redes de *fast-food* e muitas outras companhinhas dos mais diversos ramos) buscam ler desse "apelo" informações relevantes para satisfazer as opiniões dos potenciais clientes; em outras palavras, essas opiniões podem exercer uma influência enorme na formação de opiniões de outros usuários, formando a "lealdade" à marca, o público consumidor, podendo alavancar ou condenar um determinado produto ou até mesmo a imagem de uma empresa [Hoffman 2008].

O seguinte experimento visa fazer uso de ferramentas como o *NLTK* [Bird 2006], uma ferramenta em Python que permite a construção de programas em Python que permite a e o *scikit* [Pedregosa et al. 2011], um módulo em Python que possui uma ampla gama de algoritmos de aprendizado de máquina para detectar o estado afetivo de textos, criando classificadores binários (positivo e negativo), ternários (positivo, negativo e neutro) e quaternários (positivo, negativo, neutro e irrelevante) usando diferentes classificadores e técnicas de seleção.

2. Metodologia de Pesquisa

A metodologia abordada foi dividida em três partes. A primeira parte consiste na comparação da abordagem de redução de dimensionalidade PCA [Jolliffe 2002] e seleção de atributos RFE [Guyon and Elisseeff 2003] aplicados no classificador SVM [Michalski et al. 2013]. A segunda parte trata da seleção de parâmetros utilizando a técnica de Grid Search [Snoek et al. 2012]. Por fim, a terceira parte confere a execução dos classificadores Naive Bayes, SVM, Decision Tree e Random Forest [Michalski et al. 2013] em cima da base de dados selecionada.

A base de dados utilizada é formada por tweets sobre os produtos e serviços fornecidos pela Apple e fornecida pela Carnegie Mellon University^{1 2}. A base foi tratada segundo a abordagem de *Bag of Words* utilizando a ferramenta NLTK³. A seleção da base baseou-se na proximidade das informações nela presente com o conhecimento de mundo dos integrantes do grupo.

Os algoritmos supracitados e técnicas foram implementadas conforme a ferramenta de aprendizado de máquina SciKit-Learn⁴ e a linguagem utilizada foi Python⁵ *release 3.5.2*.

Métodos de Aprendizado de Máquina Usados

1. *Naive Bayes*: O classificador Naïve Bayes é um classificador probabilístico baseado no teorema de Bayes com forte independência entre as *features* [Han 2005].
2. *SVM*: O classificador *Support Vector Machines* são modelos de aprendizado de máquina supervisionado que criam modelos de associação através dos exemplos, onde esses são mapeados e é definido uma "linha" entre os conjuntos de dados. A partir desse modelo os novos dados são categorizados nos grupos existentes.
3. *Decision Tree*: O classificador Árvore de Decisão a partir de tuplas treinadas é uma das categorias de árvores de decisão. Essa árvore é uma árvore de estrutura similar a um fluxograma, onde cada nó não folha denota um teste em um atributo, cada ramo representa um resultado do teste e cada nó folha representa a classe rótulo [Han 2005].
4. *Random Forest*: O classificador *Random Forest* é um método *ensemble*. Cada classificador existente no *ensemble* é um classificador do tipo árvore de decisão e sua coleção é chamada de floresta. Por fim, as árvores de decisão individuais são geradas selecionando atributos aleatórios de cada nó [Han 2005].

¹<http://boston.lti.cs.cmu.edu/classes/95-865-K/HW/HW3/twitter-sanders-apple2.zip>

²<http://boston.lti.cs.cmu.edu/classes/95-865-K/HW/HW3/twitter-sanders-apple3.zip>

³<http://www.nltk.org/>

⁴<http://scikit-learn.org/stable/>

⁵<https://www.python.org/>

PCA vs. RFE

PCA é uma técnica que converte um conjunto de possíveis variáveis correlatas em um conjunto de variáveis não correlatas chamada de 'componentes principais', usando transformação ortogonal. Ela é uma técnica que visa encontrar as features mais importantes para a variação dos dados. É usado justamente para reduzir a dimensionalidade de um conjunto de dados muito grande. Em poucas palavras, o PCA pode ser representado pela seguinte pergunta: *Existe algum subconjunto menor de parâmetros, 30% por exemplo, que consegue explicar 70% ou mais da variação do dado?*

Feature Selection, também conhecido como *Variable Selection* ou *Attribute Selection*, é o processo de seleção, do conjunto de treinamento, de um subconjunto de features mais relevantes. Tem como objetivo facilitar o classificador, tornando-o mais eficiente, pois diminuirá o número de *features*. E isso é muito importante para classificadores no qual o número de *features* afeta no tempo de treinamento. E, em segundo, *Feature Selection*, normalmente, aumenta a precisão, pois as *features* que desviam do conjunto padrão e que podem causar uma piora na precisão são eliminadas.

Feature Selection é diferente da Redução de Dimensionalidade. Ambos métodos tem como objetivo diminuir o número de *features* da base de dados, mas a redução de dimensionalidade realiza esse trabalho combinando os atributos, enquanto a *feature selection* inclui e exclui atributos presentes na base.

Grid Search

O *Grid Search* é o método tradicional para a otimização de hiperparâmetro que é a busca exaustiva por um subconjunto dos hiperparâmetros de um algoritmo de aprendizado. Este método é necessário ser executado com auxílio de alguma métrica de performance que, normalmente, pode ser mensurado pela validação cruzada do conjunto de teste ou avaliação do conjunto de validação [wei Hsu et al. 2010].

3. Avaliação Experimental

As tabelas que seguem exibem a execução de cada um dos experimentos feitos: a matriz de confusão resultante de cada experimento (que estima a performance do algoritmo, mostrando o comparativo entre os valores previstos pelo algoritmo e os valores reais) acompanhadas de uma tabela que mostra a precisão do algoritmo em cada caso.

Para a execução dos classificadores binários: Decision tree puro, Naive Bayes puro, Random Forest puro, SVM puro, SVM com PCA e SVM com RFE, foram usados os arquivos twitter-sanders-apple3.csv para treino e para testes foi usado o arquivo twitter-sanders-apple2.csv, onde tanto o tamanho dos dados de teste quanto o tamanho dos dados de treino foram 479 *tweets* cada.

Para a execução dos classificadores ternários: Decision tree puro, Naive Bayes puro, Random Forest puro, SVM puro e SVM com PCA foram usados os arquivos full-corpus.csv para treino, com 3428 *tweets* e o arquivo twitter-sanders-apple3.csv, com 988 *tweets*.

Para a execução dos classificadores quaternários: Decision tree puro, Naive Bayes com GridSearch, Random Forest puro, SVM puro e SVM com PCA foram usados os arquivos full-corpus.csv, full_training_dataset.csv e twitter-sanders-apple2.csv para treino,

com um total de 27066 *tweets* e para testes, foram usados os arquivos irrelevantTest.csv e twitter-sanders-apple3.csv com um total de 1073 *tweets*.

O GridSearch foi experimentado usando o classificador Naive Bayes. Usando os parâmetros 'vect__ngram_range': [(1, 1), (1, 2)], 'tfidf__use_idf': (True, False) e 'clf__alpha': (1e-2, 1e-3) (pode ser visto em predict.py). Ao executá-lo, a seguinte combinação de parâmetros encontrada pelo GridSearch que maximizou a precisão: Pipeline(steps=[('vect', CountVectorizer(analyzer='word', binary=False, decode_error='strict', dtype=class 'numpy.int64', encoding='utf-8', input='content', lowercase=True, max_df=1.0, max_features=None, min_df=1, ngram_range=(1, 2), preprocessor=None, stop_words=None, strip...lse, use_idf=False)), ('clf', MultinomialNB(alpha=0.01, class_prior=None, fit_prior=True))]), onde a precisão foi de 93,9%. Tal resultado pode ser encontrado na pasta Comparação GridSearch.

Teoricamente, a *feature selection* deveria aumentar a precisão do classificador. Porém, ao executarmos o classificador binário (SVM binário), o resultado encontrado não foi esperado, que era o aumento da precisão: a precisão encontrada foi de 0.810020876827, com o RFE. Agora, sem o RFE, obtivemos um resultado melhor: uma precisão de 0.993736951983. Uma explicação para isso pode estar no tamanho do *dataset* usado como treino, que é muito pequeno no caso do teste binário, uma vez que o RFE atua melhor em um ambiente com muitos *features*, onde quanto mais *features*, maior a quantidade de coisas inúteis que ele pode remover. Tais *logs* de teste podem ser encontrados no diretório Comparação PCA vs RFE/BINÁRIO.

Na execução do PCA com o SVM binário, obtivemos um resultado que também diminuiu a precisão, comparado com o SVM puro binário. Uma possível explicação se encontra na suposição de reta que o PCA faz pode ter agrupado *features* que não são muito correlatas no resultado final, prejudicando a precisão.

**Tabela 1. Matriz de Confusão
Binária: Naïve Bayes**

Atual\Previsto	positivo	negativo
positivo	142	21
negativo	0	316

Tabela 2. Medidas da Matriz de Confusão

	precisão	recall	f1-score
positivo	1.00	0.87	0.93
negativo	0.94	1.00	0.97
média	0.96	0.96	0.96
acurácia	0.956158663883		

**Tabela 3. Matriz de Confusão
Binário: SVM**

Atual\Previsto	positivo	negativo
positivo	160	3
negativo	0	316

Tabela 4. Medidas da Matriz de Confusão

	precisão	recall	f1-score
positivo	1.00	0.98	0.99
negativo	0.99	1.00	1.00
média	0.99	0.99	0.99
acurácia	0.993736951983		

Tabela 5. Matriz de Confusão
Binária: *Decision Tree*

Atual\Previsto	positivo	negativo
positivo	163	0
negativo	0	316

Tabela 6. Medidas da Matriz de Confusão

	precisão	recall	f1-score
positivo	1.00	1.00	1.00
negativo	1.00	1.00	1.00
média	1.00	1.00	1.00
acurácia	1.0		

Tabela 7. Matriz de Confusão
Binária: *Random Forest*

Atual\Previsto	positivo	negativo
positivo	162	1
negativo	3	313

Tabela 8. Medidas da Matriz de Confusão

	precisão	recall	f1-score
positivo	0.98	0.99	0.99
negativo	1.00	0.99	0.99
média	0.99	0.99	0.99
acurácia	0.991649269311		

Tabela 9. Matriz de Confusão
Ternária: *Naïve Bayes*

Atual\Previsto	positivo	negativo	neutro
positivo	16	1	146
negativo	0	99	217
neutro	0	0	509

Tabela 10. Medidas da Matriz de Confusão

	precisão	recall	f1-score
positivo	1.00	0.10	0.18
negativo	0.99	0.31	0.48
neutro	0.58	1.00	0.74
média	0.78	0.63	0.56
acurácia	0.631578947368		

Tabela 11. Matriz de Confusão
Ternário: *SVM*

Atual\Previsto	positivo	negativo	neutro
positivo	130	4	29
negativo	2	289	25
neutro	2	4	503

Tabela 12. Medidas da Matriz de Confusão

	precisão	recall	f1-score
positivo	0.97	0.80	0.88
negativo	0.97	0.91	0.94
neutro	0.90	0.99	0.94
média	0.94	0.93	0.93
acurácia	0.933198380567		

Tabela 13. Matriz de Confusão
Ternária: *Decision Tree*

Atual\Previsto	positivo	negativo	neutro
positivo	162	0	1
negativo	1	314	1
neutro	3	2	504

Tabela 14. Medidas da Matriz de Confusão

	precisão	recall	f1-score
positivo	0.98	0.99	0.98
negativo	0.99	0.99	0.99
neutro	1.00	0.99	0.99
média	0.99	0.99	0.99
acurácia	0.991902834008		

4. Conclusão

Referências

Bird, S. (2006). Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pages 69–72. Association for Computational Lin-

Tabela 15. Matriz de Confusão
Ternária: *Random Forest*

Atual\Previsto	positivo	negativo	neutro
positivo	163	0	5
negativo	0	316	7
neutro	3	1	505

Tabela 16. Medidas da Matriz de Confusão

	precisão	recall	f1-score
positivo	0.96	0.97	0.96
negativo	1.00	0.97	0.98
neutro	0.98	0.99	0.98
média	0.98	0.98	0.98
acurácia	0.97975708502		

Tabela 17. Matriz de Confusão
Quartenário: *Naïve Bayes*

Atual\Previsto	positivo	negativo	neutro	irrelevante
positivo	131	20	12	0
negativo	3	310	3	0
neutro	29	134	344	2
irrelevante	16	28	14	27

Tabela 18. Medidas da Matriz de Confusão

	precisão	recall	f1-score
positivo	0.73	0.80	0.77
negativo	0.63	0.98	0.77
neutro	0.92	0.68	0.78
irrelevante	0.93	0.32	0.47
média	0.81	0.76	0.75
acurácia	0.756756756757		

Tabela 19. Matriz de Confusão
Quartenário: *Naïve Bayes c/ GridSearch*

Atual\Previsto	positivo	negativo	neutro	irrelevante
positivo	156	0	7	0
negativo	0	311	5	0
neutro	0	6	503	0
irrelevante	15	19	13	38

Tabela 20. Medidas da Matriz de Confusão

	precisão	recall	f1-score
positivo	0.91	0.96	0.93
negativo	0.93	0.98	0.95
neutro	0.95	0.99	0.97
irrelevante	1.00	0.45	0.62
média	0.94	0.94	0.93
acurácia	0.939422180801		

Tabela 21. Matriz de Confusão
Quaternário: *SVM*

Atual\Previsto	positivo	negativo	neutro	irrelevante
positivo	140	6	17	0
negativo	3	296	17	0
neutro	8	6	493	2
irrelevante	10	11	32	32

Tabela 22. Medidas da Matriz de Confusão

	precisão	recall	f1-score
positivo	0.87	0.86	0.86
negativo	0.93	0.94	0.93
neutro	0.88	0.97	0.92
irrelevante	0.94	0.38	0.54
média	0.90	0.90	0.89
acurácia	0.895619757689		

Tabela 23. Matriz de Confusão
Quartenário: *Decision Tree*

Atual\Previsto	positivo	negativo	neutro	irrelevante
positivo	163	0	0	0
negativo	0	316	0	0
neutro	4	2	503	0
irrelevante	19	8	50	8

Tabela 24. Medidas da Matriz de Confusão

	precisão	recall	f1-score
positivo	0.88	1.00	0.93
negativo	0.97	1.00	0.98
neutro	0.91	0.99	0.95
irrelevante	1.00	0.09	0.17
média	0.93	0.92	0.89
acurácia	0.922646784716		

guistics.

Gil de Zúñiga, H., Puig-i Abril, E., and Rojas, H. (2009). Weblogs, traditional sources

Tabela 25. Matriz de Confusão
Quartenário: *Random Forest*

Atual\Previsto	positivo	negativo	neutro	irrelevante
positivo	163	0	0	0
negativo	0	316	0	0
neutro	0	1	508	0
irrelevante	7	3	58	17

Tabela 26. Medidas da Matriz de Confusão

	precisão	recall	f1-score
positivo	0.96	1.00	0.98
negativo	0.99	1.00	0.99
neutro	0.90	1.00	0.95
irrelevante	1.00	0.20	0.33
média	0.94	0.94	0.92
acurácia	0.922646784716		

- online and political participation: An assessment of how the internet is changing the political environment. *New media & society*, 11(4):553–574.
- Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182.
- Han, J. (2005). *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- Hoffman, T. (2008). Online reputation management is hot—but is it ethical. *Computerworld, February*, pages 1–4.
- Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer.
- Michalski, R. S., Carbonell, J. G., and Mitchell, T. M. (2013). *Machine Learning: An Artificial Intelligence Approach*. Springer Publishing Company, Incorporated.
- Pang, B., Lee, L., et al. (2008). Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830.
- Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. pages 2960–2968.
- wei Hsu, C., chung Chang, C., and jen Lin, C. (2010). A practical guide to support vector classification.
- Zhu, F. and Zhang, X. (2010). Impact of online consumer reviews on sales: The moderating role of product and consumer characteristics. *Journal of marketing*, 74(2):133–148.