Processamento de Língua Natural - EP1

Gabriel A. D. de Oliveira[1] **– 9845235**
Gabriel M. de Camargo[1] **– 10284455**
Matheus Vinicius L. Martins[1] **– 9845214**
Victor G. O. M. Nicola[1] **– 9844881**

[1]Escola de Artes, Ciências e Humanidades – Universidade de São Paulo (USP)

1. Introdução

O presente trabalho apresenta um estudo de diferentes classificadores aplicados a um conjunto de dados formado por textos extraídos do twitter que foram rotulados de acordo com o posicionamento político dos autores [Santos and Paraboni 2019]. Dentre os tópicos disponíveis, foram utilizados apenas os textos referentes à aborto e controle de armas e os posicionamentos possíveis para cada texto são: a favor, contra ou neutro. Os textos foram representados como ***bag of words*** formados por **n-gramas de palavras** com diversos tamanhos e diferentes pré-processamentos aplicados, conforme detalhado na seção 2. Para a classificação, foram selecionados quatro algoritmos: ***Multinomial Nayve Bayes***, **Regressão Logística**, ***Support Vector Machine*** **(SVM)** e redes neurais ***multilayer perceptron*** (MLP).

2. Pré-processamento

De forma semelhante a outras áreas de processamentos de dados e aprendizado de máquina, no processamento de língua natural é necessário preparar os dados a serem usados para que estes possam ser lidos de uma melhor maneira pelos algoritmos. Tal preparo se mostra uma das partes mais importantes para garantir bons resultados.

Assim, algumas técnicas de pré-processamento foram aplicadas ao corpora. Inicialmente, os textos foram normalizados, passando os caracteres para minúsculos, sendo removidas as *stopwords* para a língua portuguesa, pois após uma análise de nuvem de palavras percebeu-se que artigos, preposições e outras *stopwords* eram muito frequentes em todos os textos. Foram retiradas tanto palavras muito frequentes (max-df), como palavras pouco frequentes (min-df), de acordo com os pontos de corte apresentados na tabela 1. Os textos foram representados por ngramas com permutações de 1 a 3, totalizando 6 combinações. Além disso, aplicou-se um algoritmo para seleção das mais relevantes utilizando ANOVA F1 (k-best) como função de score.

Tabela 1. Parâmetros utilizados no pré-processamento.

Parâmetro	Valores
min-df	1, 2, 3 e 4
max-df	50% e 75%
n-gram	1, 2, 3
k-best	300, 500, 1000, 1200, 1500

3. Parâmetros dos classificadores

Esta seção descreve os parâmetros explorados nos algoritmos de classificação utilizados. Toda a implementação foi feita em python 3.7. Cada parâmetro é apresentado na tabela 12 conforme o nome na documentação da biblioteca utilizada. Para os algoritmos Multinomial Naive Bayes (MNB), Support Vector Machines (SVM), regressão logística (RL) e redes neurais multilayer perceptron (MLP) foi utilizada a biblioteca scikit-learn [1].

Tabela 2. Parâmetros utilizados em cada classificador.

Classificador	Parâmetro	Valores
MNB	alpha	{0.001, 0.01, 0.1, 1.0}
	fit_prior	{1e-2, 1e-3, 1e-4, 1e-5}
SVM	C	{0.001, 0.005, 0.001, 0.05, 0.1, 0.5, 1, 5, 10, 50}
	kernel	{'poly', 'linear','sigmoid','rbf'}
MLP	hidden_layer_sizes	{(50,50), (25,25), (100,100)}
	activation	{'relu','tanh'}
	solver	{'adam'}
	alpha	{0.001, 0.00001, 0.000001}
	batch_size	{64}
	learning_rate	{'constant','invscaling','adaptive'}
	learning_rate_init	{0.001, 0.0001, 0.00001}
	early_stopping	{True}
RL	C	{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50}
	max_iter	{7000}
	solver	{'lbfgs'}

[1] https://scikit-learn.org/stable/

4. Valores ótimos

4.1. Aborto

Tabela 3. Melhores resultados obtidos com cada classificador

Classificador	Parâmetro	Valores	Média de F1-score (d.p)
MNB	alpha	0,1	0,825 (0,042)
	fit_prior	0,01000	
SVM	C	5	0,740 (0,080)
	kernel	'rbf'	
MLP	hidden_layer_sizes	(100,100)	0,740 (0,106)
	activation	'tanh'	
	solver	'adam	
	alpha	0,00001	
	batch_size	64	
	learning_rate	'constant'	
	learning_rate_init	0,001	
	early_stopping	True	
RL	C	0,001	0,688 (0,076)
	max_iter	7000	
	solver	'lbfgs'	

4.2. Armas

Tabela 4. Melhores resultados obtidos em cada classificador

Classificador	Parâmetro	Valores	Média de F1-score (d.p)
MNB	alpha	0,001	0,812 (0,065)
	fit_prior	0,010	
SVM	C	10	0,667 (0,077)
	kernel	'rbf'	
MLP	hidden_layer_sizes	(100,100)	0,718 (0,081)
	activation	'tanh'	
	solver	'adam	
	alpha	0,00001	
	batch_size	64	
	learning_rate	'constant'	
	learning_rate_init	0,001	
	early_stopping	True	
RL	C	0,001	0,626 (0,057)
	max_iter	7000	
	solver	'lbfgs'	

5. Procedimento

Esta seção descreve a metodologia aplicada para treinar os modelos e avaliar os resultados. Inicialmente, o conjunto de dados foi dividido em dez partes para a validação cru-

zada. Todos os classificadores utilizaram os mesmos subconjuntos para treino e validação.

é realizado o pré-processamento dos textos com os parâmetros escolhidos. Em seguida, o modelo é parametrizado e o treino é feito com a validação cruzada em 10 partes.

A parametrização foi feita em duas etapas: exploratória e direcionada. Na etapa exploratória decidimos quais valores de pré-processamento e parâmetros do classificadores seriam variados e com laços encadeados foram geradas todas as combinações possíveis. A segunda etapa consistiu em explorar mais os resultados parâmetros que mais impactaram na primeira etapa.

6. Resultados

Para todo o corpora foram realizados milhares de testes e, portanto, as tabelas apresentadas são apenas as primeiras linhas das tabelas finais de resultados enviadas com os arquivos de código.Tais tabelas estão separads em 2 arquivos .xlsx (PLN-aborto.xlsx e PLN-Armas.xlsx), contendo todos os testes realizados para cada classificador.

6.1. Aborto

Tabela 5. Resultados do Multinomial Nayve Bayes

min_ngram	max_ngram	max_df	min_df	alpha	fit_prior	kbest_value	avg_f1_macro	std_f1_macro
1	1	0,75	1	0,1	1,00E-02	1000	0,825	0,042
1	1	0,75	1	0,1	1,00E-03	1000	0,825	0,042
1	1	0,75	1	0,1	1,00E-04	1000	0,825	0,042
1	1	0,75	1	0,1	1,00E-05	1000	0,825	0,042
1	1	0,75	1	0,1	1,00E-02	1000	0,825	0,042
1	1	0,75	1	0,1	1,00E-03	1000	0,825	0,042
1	1	0,75	1	0,1	1,00E-04	1000	0,825	0,042
1	1	0,75	1	0,1	1,00E-05	1000	0,825	0,042
1	1	0,75	1	0,1	1,00E-02	1000	0,825	0,042
1	1	0,75	1	0,1	1,00E-03	1000	0,825	0,042
1	1	0,75	1	0,1	1,00E-04	1000	0,825	0,042
1	1	0,75	1	0,1	1,00E-05	1000	0,825	0,042
1	1	0,5	1	0,1	1,00E-02	1000	0,820	0,040
1	1	0,5	1	0,1	1,00E-03	1000	0,820	0,040
1	1	0,5	1	0,1	1,00E-04	1000	0,820	0,040
...	...	...	...	...	...	...	...	...

Tabela 6. Resultados de SVM

min_ngram	max_ngram	max_df	min_df	C	kernel	kbest_value	avg_f1_macro	std_f1_macro
3	3	0,5	1	5	rbf	500	0,740	0,080
3	3	0,5	1	10	rbf	500	0,740	0,080
3	3	0,5	1	50	rbf	500	0,740	0,080
3	3	0,75	1	5	rbf	500	0,740	0,080
3	3	0,75	1	10	rbf	500	0,740	0,080
3	3	0,75	1	50	rbf	500	0,740	0,080
3	3	0,5	1	1	rbf	500	0,734	0,072
3	3	0,75	1	1	rbf	500	0,734	0,072
3	3	0,5	1	0,5	rbf	500	0,726	0,072
3	3	0,75	1	0,5	rbf	500	0,726	0,072
3	3	0,5	1	5	rbf	300	0,720	0,078
3	3	0,5	1	10	rbf	300	0,720	0,078
3	3	0,5	1	50	rbf	300	0,720	0,078
3	3	0,75	1	5	rbf	300	0,720	0,078
3	3	0,75	1	10	rbf	300	0,720	0,078
...	...	...	...	...	...	...	...	...

Tabela 7. Resultados de MLP

min_ngram	max_ngram	max_df	min_df	hidden_layer_sizes	activation	alpha	learning_rate	learning_rate_ini	kbest_value	avg_f1_macro	std_f1_macro
1	3	0,5	2	(100, 100)	tanh	1,00E-05	constant	1,00E-03	300	0,740	0,106
1	1	0,5	3	(25, 25)	tanh	1,00E-06	constant	1,00E-03	500	0,733	0,101
1	2	0,75	1	(50, 50)	tanh	1,00E-04	invscaling	1,00E-03	300	0,733	0,083
1	3	0,75	1	(50, 50)	tanh	1,00E-04	constant	1,00E-03	300	0,729	0,075
1	3	0,75	2	(50, 50)	tanh	1,00E-05	constant	1,00E-03	300	0,728	0,069
1	2	0,5	2	(50, 50)	tanh	1,00E-05	constant	1,00E-03	300	0,728	0,074
1	2	0,75	2	(100, 100)	tanh	1,00E-04	constant	1,00E-03	300	0,728	0,041
1	3	0,75	2	(100, 100)	tanh	1,00E-05	constant	1,00E-03	300	0,728	0,049
1	3	0,5	2	(50, 50)	tanh	1,00E-04	constant	1,00E-03	300	0,727	0,065
1	3	0,75	2	(100, 100)	tanh	1,00E-04	adaptive	1,00E-03	500	0,727	0,069
1	3	0,5	2	(50, 50)	tanh	1,00E-06	invscaling	1,00E-03	300	0,724	0,063
1	3	0,75	2	(100, 100)	tanh	1,00E-05	invscaling	1,00E-03	300	0,723	0,077
1	2	0,5	1	(100, 100)	tanh	1,00E-06	invscaling	1,00E-03	300	0,722	0,071
1	3	0,75	1	(100, 100)	tanh	1,00E-04	invscaling	1,00E-03	500	0,722	0,089
1	3	0,75	1	(25, 25)	tanh	1,00E-05	constant	1,00E-03	300	0,719	0,081
...	...	...	...	...	...	...	...	...	...	...	...

Tabela 8. Resultados da regressão logística

min_ngram	max_ngram	max_df	min_df	C	max_iter	kbest_value	avg_f1_macro	std_f1_macro
1	1	0,75	2	0,001	7000	900	0,688	0,076
1	1	0,75	2	0,001	7000	1000	0,687	0,076
1	1	0,75	2	0,005	7000	1000	0,687	0,076
1	1	0,75	2	0,01	7000	1000	0,687	0,076
1	1	0,75	2	0,05	7000	1000	0,687	0,076
1	1	0,75	2	0,1	7000	1000	0,687	0,076
1	1	0,75	2	0,5	7000	1000	0,687	0,076
1	1	0,75	2	1	7000	1000	0,687	0,076
1	1	0,75	2	5	7000	1000	0,687	0,076
1	1	0,75	2	10	7000	1000	0,687	0,076
1	1	0,75	2	50	7000	1000	0,687	0,076
1	1	0,75	2	100	7000	1000	0,687	0,076
1	1	0,75	2	0,001	7000	1000	0,687	0,076
1	2	0,5	3	0,001	7000	500	0,687	0,102
1	2	0,5	3	0,005	7000	500	0,687	0,102
...	...	...	...	...	...	...	...	...

6.2. Armas

Tabela 9. Resultados do Multinomial Nayve Bayes

min_ngram	max_ngram	max_df	min_df	alpha	fit_prior	kbest_value	avg_f1_macro	std_f1_macro
1	2	0,75	1	0,001	0,01	200	0,812	0,065
1	2	0,75	1	0,001	0,001	200	0,812	0,065
1	2	0,75	1	0,001	0,0001	200	0,812	0,065
1	2	0,75	1	0,001	1.00E-05	200	0,812	0,065
1	2	0,75	2	0,001	0,01	200	0,812	0,065
1	2	0,75	2	0,001	0,001	200	0,812	0,065
1	2	0,75	2	0,001	0,0001	200	0,812	0,065
1	2	0,75	2	0,001	1.00E-05	200	0,812	0,065
1	2	0,75	1	0,01	0,01	200	0,806	0,062
1	2	0,75	1	0,01	0,001	200	0,806	0,062
1	2	0,75	1	0,01	0,0001	200	0,806	0,062
1	2	0,75	1	0,01	1.00E-05	200	0,806	0,062
1	2	0,75	2	0,01	0,01	200	0,806	0,062
1	2	0,75	2	0,01	0,001	200	0,806	0,062
1	2	0,75	2	0,01	0,0001	200	0,806	0,062
...	...	...	...	...	...	...	...	...

Tabela 10. Resultados de SVM

min_ngram	max_ngram	max_df	min_df	C	kernel	kbest_value	avg_f1_macro	std_f1_macro
2	2	0,50	1	10	rbf	500	0,667	0,077
2	2	0,50	1	50	rbf	500	0,667	0,077
2	2	0,75	1	10	rbf	500	0,667	0,077
2	2	0,75	1	50	rbf	500	0,667	0,077
2	3	0,50	1	10	rbf	500	0,667	0,084
2	3	0,50	1	50	rbf	500	0,667	0,084
2	3	0,75	1	10	rbf	500	0,667	0,084
2	3	0,75	1	50	rbf	500	0,667	0,084
2	2	0,50	1	5	rbf	500	0,659	0,077
2	2	0,75	1	5	rbf	500	0,659	0,077
2	3	0,50	1	5	rbf	500	0,657	0,067
2	3	0,75	1	5	rbf	500	0,657	0,067
1	2	0,75	2	5	rbf	500	0,631	0,092
1	3	0,75	2	5	rbf	500	0,629	0,102
2	2	0,50	1	10	rbf	300	0,628	0,079
...	...	...	...	...	...	...	...	...

Tabela 11. Resultados de MLP

min_ngram	max_ngram	max_df	min_df	hidden_layer_sizes	activation	alpha	learning_rate	learning_rate_ini	kbest_value	avg_f1_macro	std_f1_macro
1	3	0,5	1	(100, 100)	tanh	1,00E-04	constant	1,00E-03	500	0,718	0,081
1	3	0,75	2	(50, 50)	tanh	1,00E-04	adaptive	1,00E-03	500	0,711	0,084
1	3	0,75	2	(100, 100)	tanh	1,00E-06	constant	1,00E-03	300	0,701	0,063
1	2	0,5	1	(100, 100)	tanh	1,00E-04	invscaling	1,00E-03	500	0,699	0,076
1	2	0,75	1	(50, 50)	tanh	1,00E-05	adaptive	1,00E-03	500	0,698	0,068
1	2	0,5	2	(100, 100)	tanh	1,00E-04	adaptive	1,00E-03	300	0,694	0,086
1	2	0,5	1	(100, 100)	relu	1,00E-05	constant	1,00E-03	300	0,694	0,069
1	3	0,5	2	(100, 100)	tanh	1,00E-05	adaptive	1,00E-03	500	0,694	0,108
1	2	0,5	1	(50, 50)	tanh	1,00E-05	invscaling	1,00E-03	500	0,692	0,056
1	2	0,5	2	(100, 100)	tanh	1,00E-04	constant	1,00E-03	300	0,690	0,084
1	3	0,5	2	(25, 25)	tanh	1,00E-05	adaptive	1,00E-03	500	0,689	0,082
1	2	0,5	1	(50, 50)	tanh	1,00E-04	adaptive	1,00E-03	300	0,688	0,055
1	2	0,5	3	(25, 25)	tanh	1,00E-06	invscaling	1,00E-03	500	0,688	0,072
1	2	0,5	1	(25, 25)	tanh	1,00E-04	constant	1,00E-03	300	0,688	0,075
1	3	0,75	2	(100, 100)	tanh	1,00E-04	invscaling	1,00E-03	500	0,688	0,091
...	...	...	...	...	...	...	...	...	...	...	...

Tabela 12. Resultados da regressão logística

min_ngram	max_ngram	max_df	min_df	C	max_iter	kbest_value	avg_f1_macro	std_f1_macro
1	2	0,75	3	0,001	7000	500	0,627	0,057
1	2	0,75	3	0,005	7000	500	0,627	0,057
1	2	0,75	3	0,01	7000	500	0,627	0,057
1	2	0,75	3	0,05	7000	500	0,627	0,057
1	2	0,75	3	0,1	7000	500	0,627	0,057
1	2	0,75	3	0,5	7000	500	0,627	0,057
1	2	0,75	3	1	7000	500	0,627	0,057
1	2	0,75	3	5	7000	500	0,627	0,057
1	2	0,75	3	10	7000	500	0,627	0,057
1	2	0,75	3	50	7000	500	0,627	0,057
1	2	0,75	3	100	7000	500	0,627	0,057
1	3	0,75	2	0,001	7000	500	0,624	0,054
1	3	0,75	2	0,005	7000	500	0,624	0,054
1	3	0,75	2	0,01	7000	500	0,624	0,054
1	3	0,75	2	0,05	7000	500	0,624	0,054
...	...	...	...	...	...	...	...	...

7. Instruções

Foram desenvolvidos diferentes códigos, um para cada classificador e conjunto de dados, totalizando 8 arquivos (e.g. RL_abortion). Foi realizada uma padronização nos parâmetros de pré-processamento para reproduzir os resultados. Isso permite a paralelização dos treinos e testes, bem como a análise dos resultados. As bibliotecas necessárias para executar um código são: *sklearn*, *pandas*, *numpy* e *nltk*.

A execução dos scripts devem então ser feitas separadamente, cada qual com o seu arquivo .py. Além disso, deve-se atentar para a localização dos scripts e todos os arquivos envolvidos (i.e. base de dados e resultados). Para tal, todos os arquivos foram organizados de acordo com a seguinte estrutura:

```
raíz
├── Dados
│   ├── ep1_abortion_train.csv
│   ├── ep1_gun-control_train.csv
│   ├── folds.npy
│   └── folds_armas.npy
├── Resultados
│   ├── abortion_resultados_multinomial_naive_bayes.csv
│   ├── gun_control_resultados_multinomial_naive_bayes.csv
│   └── ...
├── MNB_abortion.py
├── MNB_gun_control.py
└── ...
```

Assim, cada script ao ser executado extrai os dados da pasta `Dados` e, ao terminar sua execução, salva os resultados na pasta `Resultados` de acordo com uma nomenclatura padrão como se pode ver na árvore acima, detalhando a qual

Referências

Santos, W. and Paraboni, I. (2019). Moral stance recognition and polarity classification from Twitter and elicited text. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 1069–1075. INCOMA Ltd.