

# Processamento de Língua Natural - EP1

Gabriel A. D. de Oliveira<sup>1</sup> – 9845235

Gabriel M. de Camargo<sup>1</sup> – 10284455

Matheus Vinicius L. Martins<sup>1</sup> – 9845214

Victor G. O. M. Nicola<sup>1</sup> – 9844881

<sup>1</sup>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

Algumas técnicas de pré-processamento foram aplicadas ao corpora e esta seção descreve o procedimento para tal. Inicialmente, os textos foram normalizados, passando os caracteres para caixa baixa, 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 palavras 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
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

2 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 <sup>1</sup>.

**Tabela 2. Parâmetros utilizados em cada classificador.**

Classificador	Parâmetro	Valores
<b>MNB</b>	alpha	{0.001, 0.01, 0.1, 1.0}
	fit_prior	{1e-2, 1e-3, 1e-4, 1e-5}
<b>SVM</b>	C	{0.001, 0.005, 0.001, 0.05, 0.1, 0.5, 1, 5, 10, 50}
	kernel	{'poly', 'linear', 'sigmoid', 'rbf'}
<b>MLP</b>	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}
<b>RL</b>	C	{0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50}
	max_iter	{7000}
	solver	{'lbfgs'}

## 4. Valores ótimos

### 4.1. Aborto

**Tabela 3. Melhores resultados obtidos com cada classificador**

Classificador	Parâmetro	Valores	Média de F1-score (d.p)
<b>MNB</b>	alpha	0,1	0,825 (0,042)
	fit_prior	0,01000	
<b>SVM</b>	C	5	0,740 (0,080)
	kernel	'rbf'	
<b>MLP</b>	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	
<b>RL</b>	C	0,001	0,688 (0,076)
	max_iter	7000	
	solver	'lbfgs'	

<sup>1</sup><https://scikit-learn.org/stable/>

## 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 cruzada. Todos os classificadores utilizaram os mesmos subconjuntos de dados. Em seguida, foi realizado o pré-processamento dos textos com os parâmetros escolhidos e, por fim, ocorreu o treino e a validação dos modelos.

A parametrização do modelo e do pré-processamento foi feita em duas etapas: exploratória e direcionada. Na etapa exploratória, decidimos quais valores de pré-processamento e parâmetros dos classificadores seriam variados e com laços encadeados foram geradas todas as combinações possíveis. A segunda etapa consistiu em explorar mais os resultados parâmetros que tiveram maior impacto no desempenho dos classificadores 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, conforme explicado na seção 7. Tais tabelas estão separadas 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

[illegible]

### Tabela 6. Resultados de SVM

[illegible]

### 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

[illegible]

## 6.2. Armas

### Tabela 9. Resultados do Multinomial Nayve Bayes

[illegible]

### Tabela 10. Resultados de SVM

[illegible]

**Tabela 11. Resultados de MLP**

min_ngram	max_ngram	max_df	min_df	hidden_layer_sizes	activation	alpha	learning_rate	learning_rate_init	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**

[illegible]

## 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 os códigos são: `sklearn`, `pandas`, `numpy` e `nlTK`.

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

```
raiz
├── 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
│   └── ...
├── PLN-aborto.xlsx
├── PLN-Armaz.xlsx
├── 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 apresentada na árvore acima, mostrando a qual execução o arquivo de resultados se refere. Basta executar cada um da maneira preferida, seja via linha de comando seja em um IDE.

## 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.