


Exercício - Aula 3

Como já sabemos, a base de dados de "produto" [1] foi utilizada para modelos de classificação supervisionados e sabemos qual foi a configuração que gerou um bom resultado. Que tal aplicarmos essas técnicas de pré-processamento que gerou um bom modelo e compararmos? Faça um teste de análise de tópicos com 4 tópicos conforme as seguintes configurações:

- Remover registros com valores nulos;
- Concatenar as colunas de nome e descrição;
- Aplicar lematização em verbos;
- Amostra de 20% para teste e random_state = 42;
- Vetorização de contagem dos termos em unigramas removendo stopwords (NLTK + Spacy).

[1] <https://dados-ml-pln.s3-sa-east-1.amazonaws.com/produtos.csv>

```
!python -m spacy download pt_core_news_sm --quiet
```



13.0/13.0 MB 31.1 MB/s eta 0:00:00

✓ Download and installation successful

You can now load the package via `spacy.load('pt_core_news_sm')`

⚠ Restart to reload dependencies

If you are in a Jupyter or Colab notebook, you may need to restart Python in order to load all the package's dependencies. You can do this by selecting the 'Restart kernel' or 'Restart runtime' option.


```
import pandas as pd

df = pd.read_csv("https://dados-ml-pln.s3-sa-east-1.amazonaws.com/produtos.csv", delimiter=";", encoding='utf-8')

df.dropna(inplace=True)

df["texto"] = df['nome'] + " " + df['descricao']

df.describe()
```



	nome	descricao	categoria	texto
count	2916	2916	2916	2916
unique	2584	2460	4	2646
top	Mais Escuro - Cinquenta Tons Mais Escuros Pel...	JOGO ORIGINAL. NOVO. LACRADO. PRONTA ENTREGA. ...	livro	Boneco Dragon Ball Z Son Gokou Produto novo
fren	20	39	838	20

```

import spacy
nlp = spacy.load('pt_core_news_sm')

def lemmatizer_verbs(text):
    sent = []
    doc = nlp(text)
    for word in doc:
        if word.pos_ == "VERB":
            sent.append(word.lemma_)
        else:
            sent.append(word.text)
    return " ".join(sent)

df['text_lemma_verbs'] = df.texto.apply(lemmatizer_verbs)

from sklearn.model_selection import train_test_split

df_train, df_test = train_test_split(df,
    test_size = 0.2,
    random_state = 42
)

from sklearn.feature_extraction.text import CountVectorizer
import nltk
nltk.download('stopwords')
import spacy

nlp = spacy.load('pt_core_news_sm')

# stopwords do SpaCy e NLTK combinadas
stops = list(set(nlp.Defaults.stop_words).union(set(nltk.corpus.stopwords.words('portuguese'))))


from sklearn.feature_extraction.text import CountVectorizer

# verorização
vect = CountVectorizer(ngram_range=(1,1), stop_words=stops)
vect.fit(df_train.text_lemma_verbs)
text_vect_train = vect.transform(df_train.text_lemma_verbs)

from sklearn.decomposition import LatentDirichletAllocation

# modelo
LDA = LatentDirichletAllocation(n_components=4, random_state=42)
LDA.fit(text_vect_train)

```

 [nltk_data] Downloading package stopwords to /root/nltk_data...
 [nltk_data] Unzipping corpora/stopwords.zip.

▾ LatentDirichletAllocation ⓘ ?
 LatentDirichletAllocation(n_components=4, random_state=42)

```
# top palavras dos tópicos
```

```
terms = vect.get_feature_names_out()
```

```
for index, topic in enumerate(LDA.components_):
```

```
    terms_comp = zip(terms, topic)
```

```
    sorted_terms = sorted(terms_comp, key= lambda x:x[1], reverse=True)[:15]
```

```
    print("THE TOP 15 WORDS FOR TOPIC # "+str(index)+": ")
```

```
    #print(sorted_terms)
```

```
    print([t[0] for t in sorted_terms])
```

```
    print('\n')
```



```
THE TOP 15 WORDS FOR TOPIC # 0:
```

```
['mercado', 'produto', 'prazo', 'pagamento', 'produtos', 'entrega', 'envio', 'frete', 'enviar',
```

```
THE TOP 15 WORDS FOR TOPIC # 1:
```

```
['produto', 'edição', 'livro', 'páginas', 'vida', 'mundo', 'anos', '00', 'história', 'origem',
```

```
THE TOP 15 WORDS FOR TOPIC # 2:
```

```
['pincel', 'cm', 'maquiagem', 'kit', 'cores', 'cílios', 'produto', 'profissional', 'maleta', 't
```

```
THE TOP 15 WORDS FOR TOPIC # 3:
```

```
['jogo', 'the', 'digital', 'compra', 'original', 'frete', 'of', 'vendedor', 'jogar', 'online',
```



```
...
```

```
0 - brinquedo
```

```
1 - livro
```

```
2 - maquiagem
```

```
3 - game
```

```
...
```



```
'\n0 - brinquedo\n1 - livro\n2 - maquiagem\n3 - game\n'
```

```
{'brinquedo': 0, 'livro': 1, 'maquiagem': 2, 'game': 3 }
```



```
{'maquiagem': 2, 'game': 3, 'livro': 1, 'brinquedo': 0}
```

```
text_vect_test = vect.transform(df_test.text_lemma_verbs)
```

```
results = LDA.transform(text_vect_test)
```

```
df_test['topico'] = results.argmax(axis=1)
```

```
df_test.head()
```



	nome	descricao	categoria	texto	text_lemma_verbs	topico
2700	Estojo Duo Iluminador E Bronzer Belle Angel M...	DUO ILUMINADOR E BRONZER BELLE ANGEL - B025O D...	maquiagem	Estojo Duo Iluminador E Bronzer Belle Angel M...	Estojo Duo Iluminador E Bronzer Belle Angel ...	2
1297	Patrulha Canina Carrinhos De Fricção 6	Kit carrinhos da Patrulha canina Tamanho: Cerc...	brinquedo	Patrulha Canina Carrinhos De Fricção 6	Patrulha Canina Carrinhos De Fricção 6 Perso...	0

```
import numpy as np
```

```
df_test['categ_cod'] = df_test.categoria.map(  
    {'brinquedo': 0, 'livro': 1, 'maquiagem': 2, 'game': 3 }  
)
```

```
df_test.head()
```



	Unidade nome	Escova Co... descricao	categoria	alta qualidade texto	esta alta qualida... text_lemma_verbs	topico	categ_cod
2700	Estojo Duo Iluminador E Bronzer Belle Angel M...	DUO ILUMINADOR E BRONZER BELLE ANGEL - B025O D...	maquiagem	Estojo Duo Iluminador E Bronzer Belle Angel M...	Estojo Duo Iluminador E Bronzer Belle Angel ...	2	2
1297	Patrulha Canina Carrinhos De Fricção	Kit carrinhos da Patrulha canina Tamanho:	brinquedo	Patrulha Canina Carrinhos De Fricção	Patrulha Canina Carrinhos De Fricção 6 Perso...	0	0

```
from sklearn.metrics import accuracy_score, confusion_matrix
```

```
print(accuracy_score(df_test.categ_cod, df_test.topico))
```



0.789383561643896	3ds Midia Fisica ...	100% ORIGINALS# ...	3ds Midia Fisica ...	Midia Fisica...	-	-
-------------------	----------------------	---------------------	----------------------	-----------------	---	---

```
import seaborn as sns  
import matplotlib.pyplot as plt
```

```
matrix = confusion_matrix(df_test.categ_cod, df_test.topico)
```

```
df_cm = pd.DataFrame(matrix, index=list('0123'),  
                      columns=list('0123'))
```

```
plt.figure(figsize = (10,7))  
sns.heatmap(df_cm, annot=True)
```

 <Axes: >

