Classificação de Avaliações de Produtos

Projeto NLP - Deep Learning

Lucas Accioly

Especialização
Deep Learning

Centro de Informática
UFPE

Universidade Federal de Pernambuco

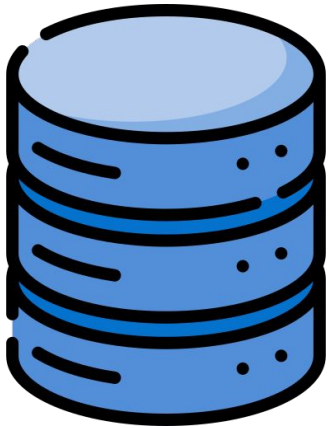
Objetivo

Automatizar e classificar os reviews de produtos em scores.

Introdução

- Base de Dados da Olist de avaliações de produtos comercializados (Kaggle)
- Olist é uma startup Brasileira de tecnologia para varejo
  - Gestão de Lojas offline e online (e-commerce)
  - MercadoLivre, Americanas e Amazon
  - Loja Única
  - 30 mil lojistas
  - 2 milhões de consumidores únicos a cada ano
- Avaliações de produtos de telefonia

Base de Dados

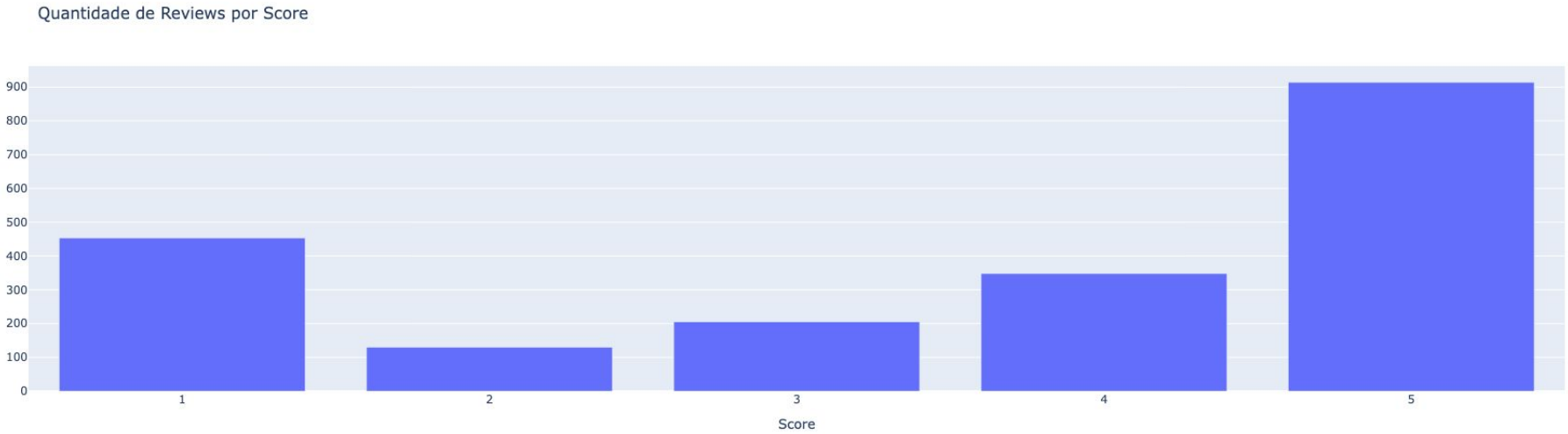


- Join 4 bases
  - Itens por pedido
  - Reviews de pedidos
  - Pedidos
  - Produtos
- Filtrar categoria Telefonia
- 2050 Reviews de produtos de Telefonia

- Limpeza e eliminação comentários Null e espaço vazio

```
Data columns (total 5 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   review_id               2050 non-null   object
 1   product_category_name   2050 non-null   object
 2   review_score            2050 non-null   int64
 3   review_comment_title    440 non-null    object
 4   review_comment_message  2050 non-null   object
```

```
Score :  1
Quantidade de Reviews com score 1:  453
Percentual de Reviews com score 1:  22.097560975609756  %
Exemplo de Reviews :  A película não é de gel. É de plástico e a capa é de plástico de um material péssimo.

Tive que jogar no lixo.

----------


----------
Score :  2
Quantidade de Reviews com score 2:  130
Percentual de Reviews com score 2:  6.341463414634147  %
Exemplo de Reviews :  Comprei dois produtos e veio só um
----------


----------
Score :  3
Quantidade de Reviews com score 3:  205
Percentual de Reviews com score 3:  10.0  %
Exemplo de Reviews :  solicitei a devolução mesmo assim me mandarão e mandarão errado
----------


----------
Score :  4
Quantidade de Reviews com score 4:  348
Percentual de Reviews com score 4:  16.975609756097562  %
Exemplo de Reviews :  Produto foi encontrado com um bom preço no site e entregue bem rapido.
----------


----------
Score :  5
Quantidade de Reviews com score 5:  914
Percentual de Reviews com score 5:  44.58536585365854  %
Exemplo de Reviews :  Produto de excelente qualidade
----------


Quantidade total de reviews : 2050
```


Quantidade de Reviews por Score
900
800
700
600
500
400
300
200
100
0
1
2
3
4
5
Score

Pré-Processamento

- Emoji, caracteres especiais e palavras escritas abreviadas
- Normalização palavras
- Maiúscula nomes próprios, começo de frase e acrônimos
- Retirar Stop Words, pontuação, dígitos, moedas e símbolos especiais.
- Manter o não devido a importância

Análise Descritiva e Exploratória

WordCloud

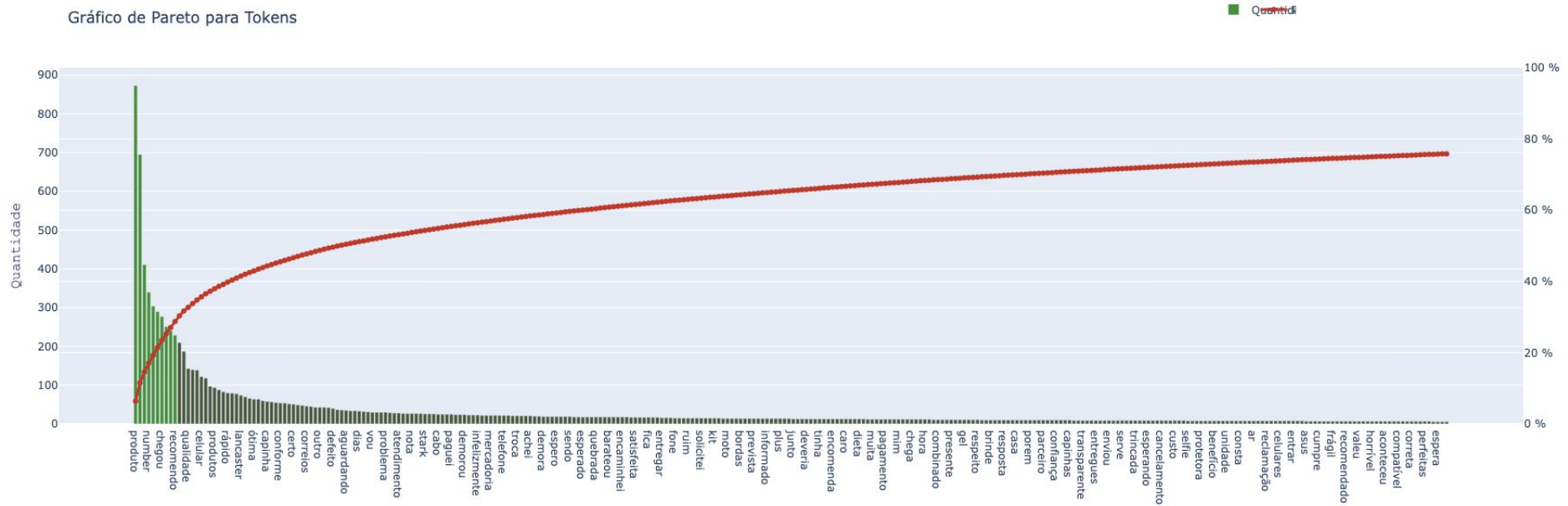
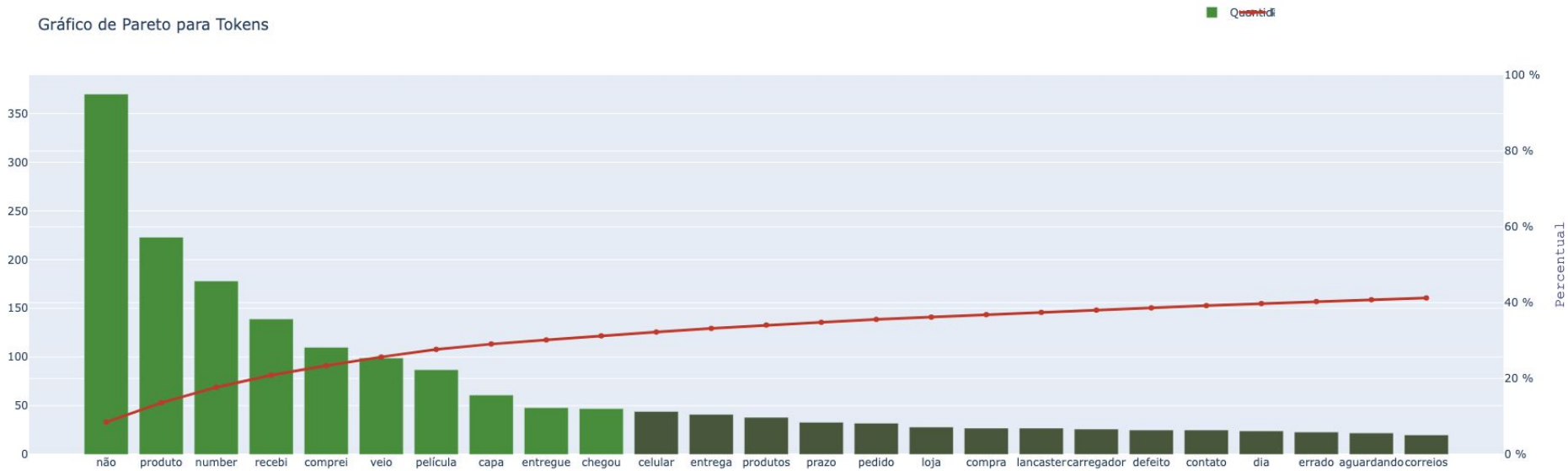
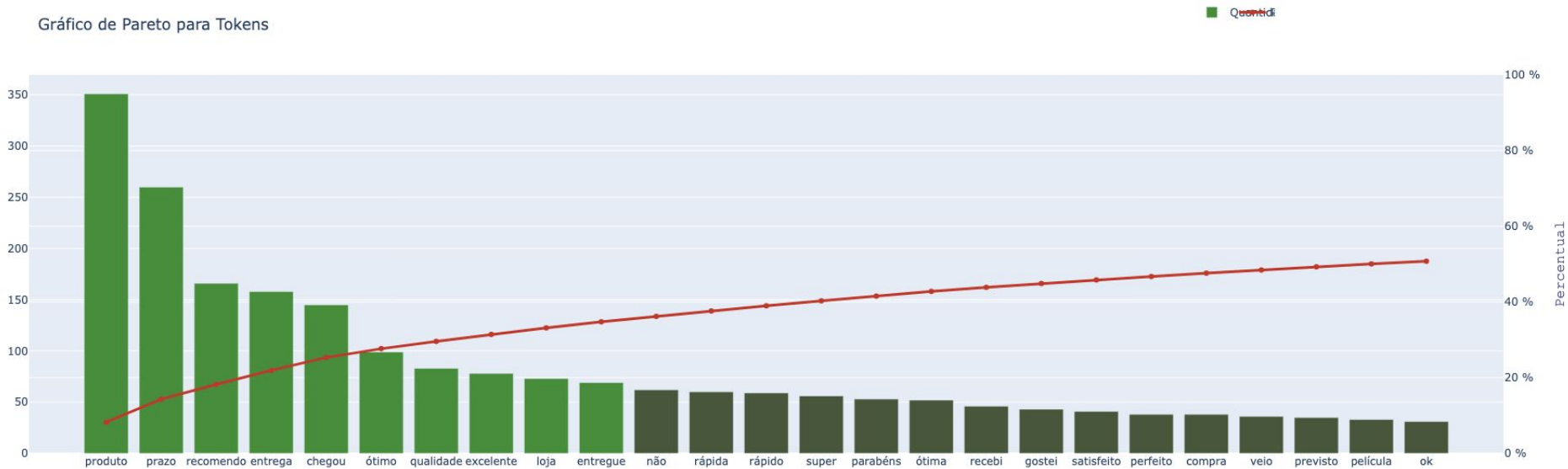


Gráfico de Pareto para Tokens
Quantidade
Percentual
0
100
200
300
400
500
600
700
800
900
0 %
20 %
40 %
60 %
80 %
100 %
produto
number
chegou
recomendo
qualidade
celular
produtos
rápido
lancaster
ótima
capinha
conforme
certo
correios
outro
defeito
aguardando
dias
vou
problema
atendimento
nota
stark
cabo
paguei
demorou
infelizmente
mercadoria
telefone
troca
achei
demora
espero
sendo
esperado
quebrada
barateou
encaminhei
satisfeita
fica
entregar
fone
ruim
solicitei
kit
moto
bordas
prevista
informado
plus
junto
deveria
tinha
encomenda
caro
dieta
muita
pagamento
mim
chega
hora
combinado
presente
gel
respeito
brinde
resposta
casa
porem
parceiro
confiança
capinhas
transparente
entregues
enviou
serve
trincada
esperando
cancelamento
custo
selfie
protetora
benefício
unidade
consta
ar
reclamação
celulares
entrar
asus
cumpre
frágil
recomendado
valeu
horrível
aconteceu
compatível
correta
perfeitas
espera

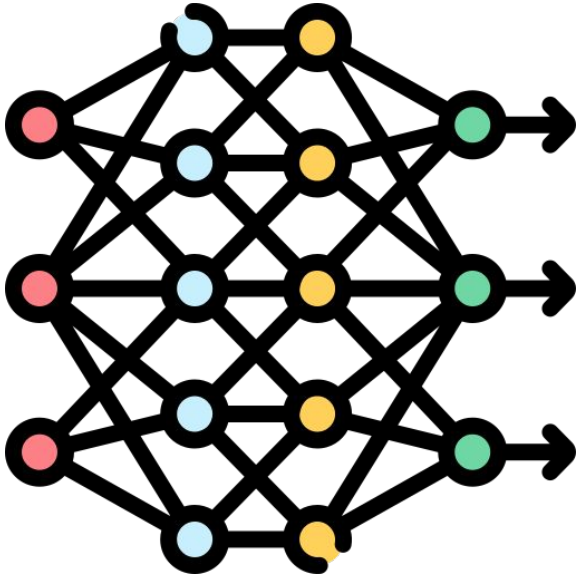
Pareto Score 1



Pareto Score 5

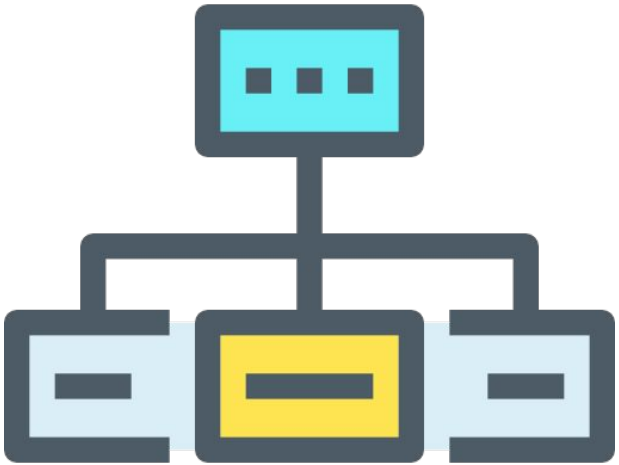


Modelos



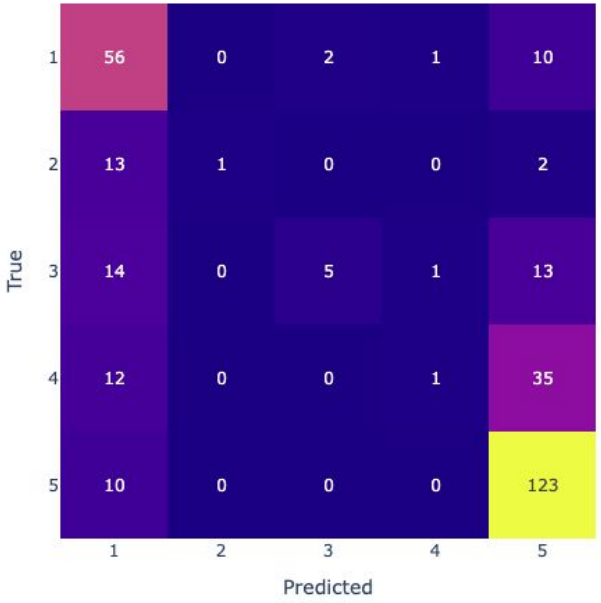
Treino, Validação e Teste

- 70% Treino
- 15% Validação
- 15% Teste



SVM + BoW

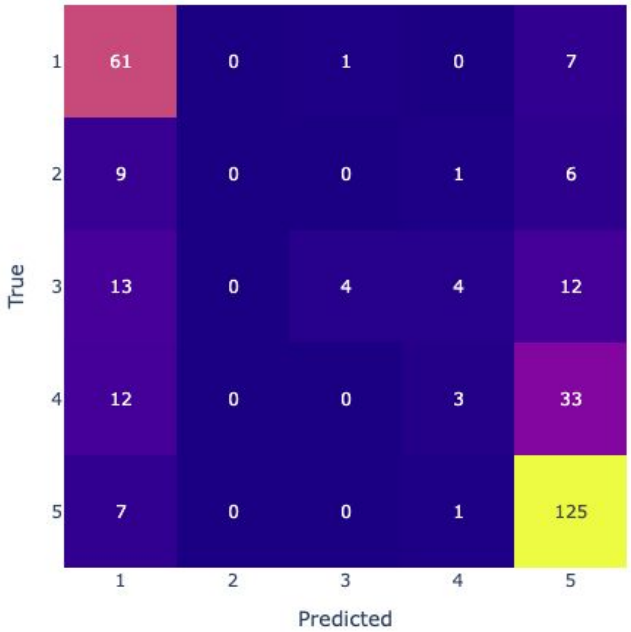
- CountVectorizer



	precision	recall	f1-score	support
1	0.53	0.81	0.64	69
2	1.00	0.06	0.12	16
3	0.71	0.15	0.25	33
4	0.33	0.02	0.04	48
5	0.67	0.92	0.78	133
accuracy			0.62	299
macro avg	0.65	0.39	0.37	299
weighted avg	0.61	0.62	0.54	299

SVM + TFIDF Unigrams

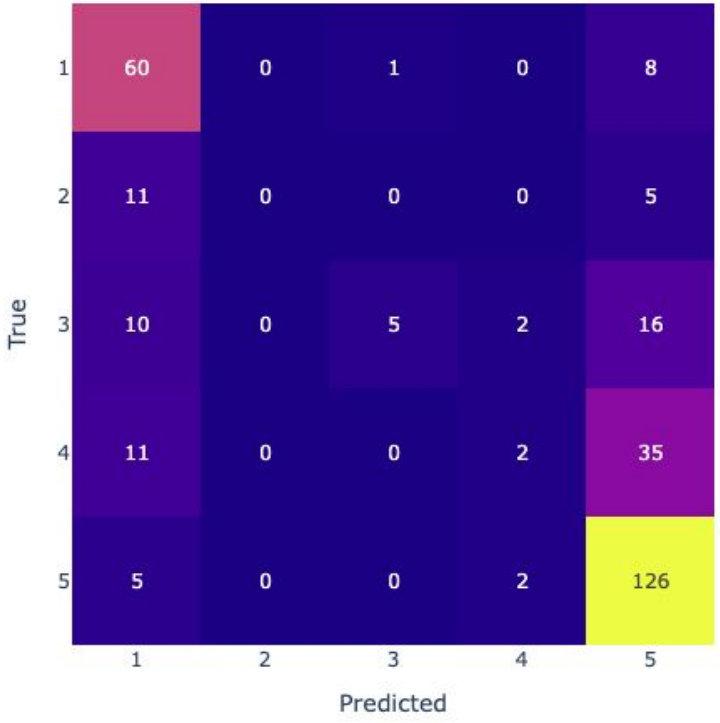
- TfidfVectorizer Unigrams



	precision	recall	f1-score	support
1	0.60	0.88	0.71	69
2	0.00	0.00	0.00	16
3	0.80	0.12	0.21	33
4	0.33	0.06	0.11	48
5	0.68	0.94	0.79	133
accuracy			0.65	299
macro avg	0.48	0.40	0.36	299
weighted avg	0.58	0.65	0.56	299

SVM + TFIDF BiGrams

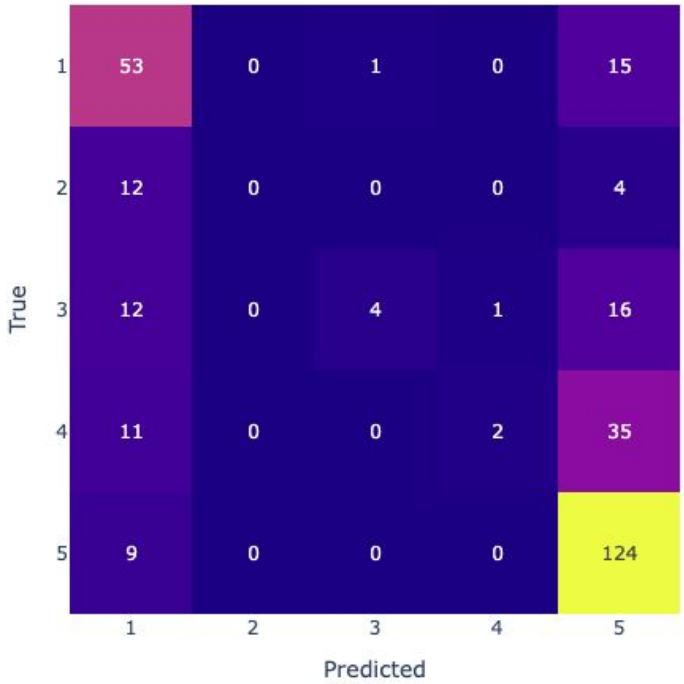
- TfidfVectorizer Bigrams



	precision	recall	f1-score	support
1	0.62	0.87	0.72	69
2	0.00	0.00	0.00	16
3	0.83	0.15	0.26	33
4	0.33	0.04	0.07	48
5	0.66	0.95	0.78	133
accuracy			0.65	299
macro avg	0.49	0.40	0.37	299
weighted avg	0.58	0.65	0.55	299

SVM + CBOW 300

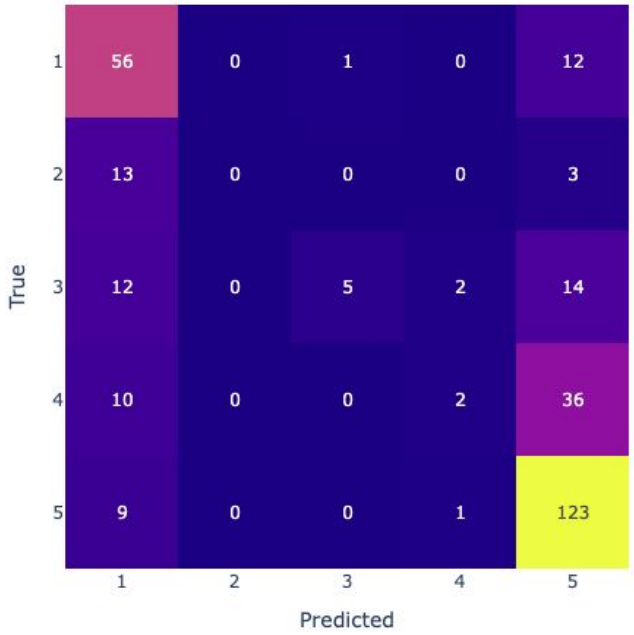
- Embedding Word2Vector CBOW 300 (NILC - USP)



	precision	recall	f1-score	support
1	0.55	0.77	0.64	69
2	0.00	0.00	0.00	16
3	0.80	0.12	0.21	33
4	0.67	0.04	0.08	48
5	0.64	0.93	0.76	133
accuracy			0.61	299
macro avg	0.53	0.37	0.34	299
weighted avg	0.61	0.61	0.52	299

SVM + SKIP 300

- Embedding Word2Vector Skip-Gram 300 (NILC - USP)



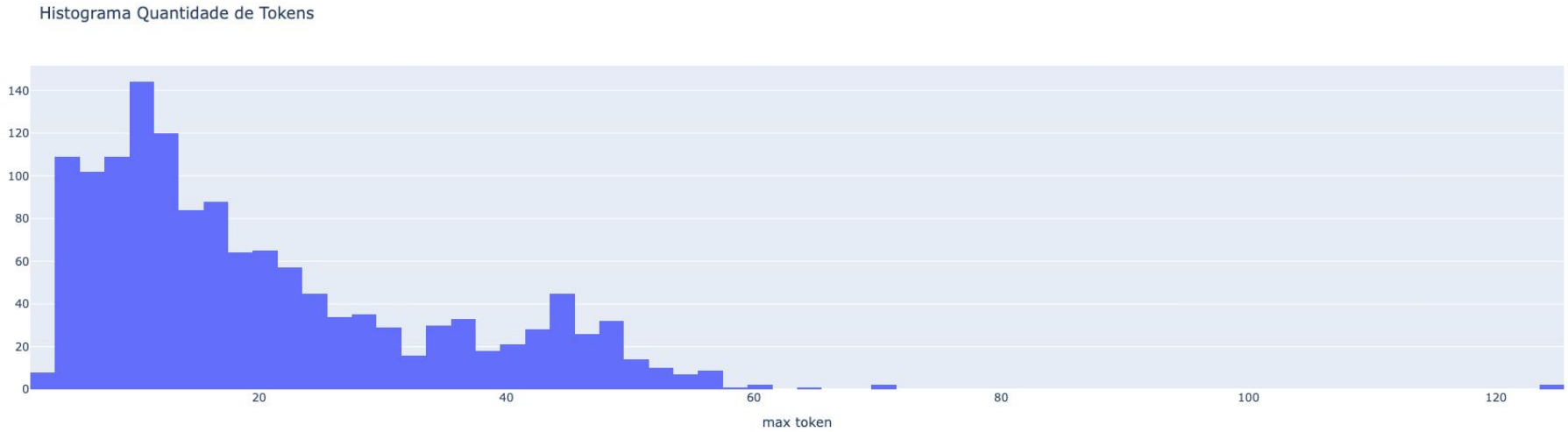
	precision	recall	f1-score	support
1	0.56	0.81	0.66	69
2	0.00	0.00	0.00	16
3	0.83	0.15	0.26	33
4	0.40	0.04	0.08	48
5	0.65	0.92	0.77	133
accuracy			0.62	299
macro avg	0.49	0.39	0.35	299
weighted avg	0.58	0.62	0.53	299

BERT

- BERTimbau (Pré-Treinado em Português) (NeuralMind)
- Tokenização
- Histograma Quantidade de Tokens

BERT

- Tokenização 165 tokens



BERT

- Arquitetura BERT (Pooling Embedding) (765) + 1 camada de 1500 + 1 camada de saída de 5
- DropOut de 30%
- BatchNormalization
- CrossEntropyLoss
- Softmax
- Penalização de Pesos no treinamento (Desbalanceamento)

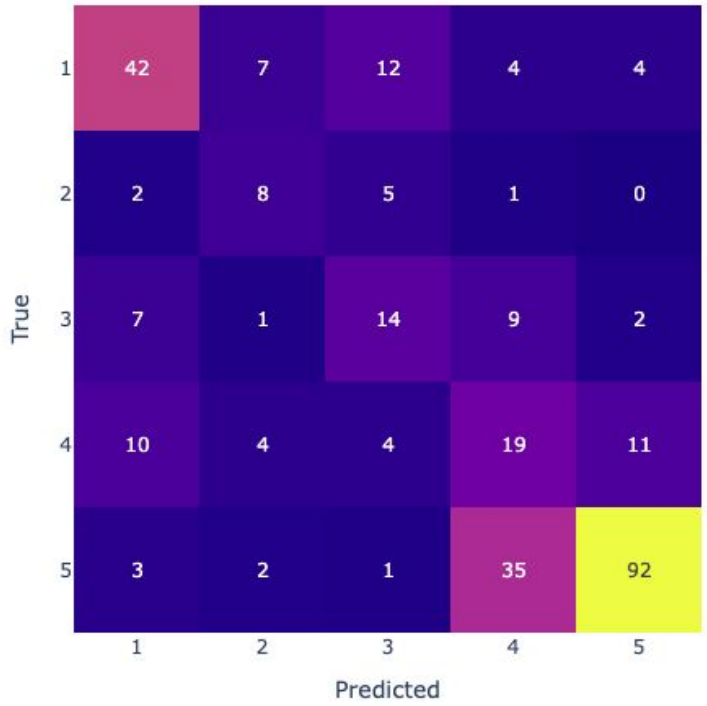
BERT

```
Bert(
  (bertimbau): BertModel(
    (embeddings): BertEmbeddings(
      (word_embeddings): Embedding(29794, 768, padding_idx=0)
      (position_embeddings): Embedding(512, 768)
      (token_type_embeddings): Embedding(2, 768)
      (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
      (dropout): Dropout(p=0.1, inplace=False)
    )
    (encoder): BertEncoder(
      (layer): ModuleList(
        (0-11): 12 x BertLayer(
          (attention): BertAttention(
            (self): BertSdpaSelfAttention(
              (query): Linear(in_features=768, out_features=768, bias=True)
              (key): Linear(in_features=768, out_features=768, bias=True)
              (value): Linear(in_features=768, out_features=768, bias=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
            (output): BertSelfOutput(
              (dense): Linear(in_features=768, out_features=768, bias=True)
              (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
          )
          (intermediate): BertIntermediate(
            (dense): Linear(in_features=768, out_features=3072, bias=True)
            (intermediate_act_fn): GELUActivation()
          )
          (output): BertOutput(
            (dense): Linear(in_features=3072, out_features=768, bias=True)
            (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
            (dropout): Dropout(p=0.1, inplace=False)
          )
        )
      )
    )
    (pooler): BertPooler(
      (dense): Linear(in_features=768, out_features=768, bias=True)
      (activation): Tanh()
    )
  )
  (dropout): Dropout(p=0.3, inplace=False)
  (batch_normalization): BatchNorm1d(768, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (head_classification): Sequential(
    (0): Linear(in_features=768, out_features=1500, bias=True)
    (1): Linear(in_features=1500, out_features=5, bias=True)
  )
  (loss): CrossEntropyLoss()
)
```

BERT

```
==========================================================================
Layer (type:depth-idx)                                   Param #
==========================================================================
├─BertModel: 1-1                                        (108,923,136)
├─Dropout: 1-2                                          --
├─BatchNorm1d: 1-3                                      1,536
├─Sequential: 1-4                                       1,161,005
├─CrossEntropyLoss: 1-5                                 --
==========================================================================
Total params: 110,085,677
Trainable params: 1,162,541
Non-trainable params: 108,923,136
==========================================================================
==========================================================================
Layer (type:depth-idx)                                   Param #
==========================================================================
├─BertModel: 1-1                                        (108,923,136)
├─Dropout: 1-2                                          --
├─BatchNorm1d: 1-3                                      1,536
├─Sequential: 1-4                                       1,161,005
├─CrossEntropyLoss: 1-5                                 --
==========================================================================
Total params: 110,085,677
Trainable params: 1,162,541
Non-trainable params: 108,923,136
==========================================================================
```

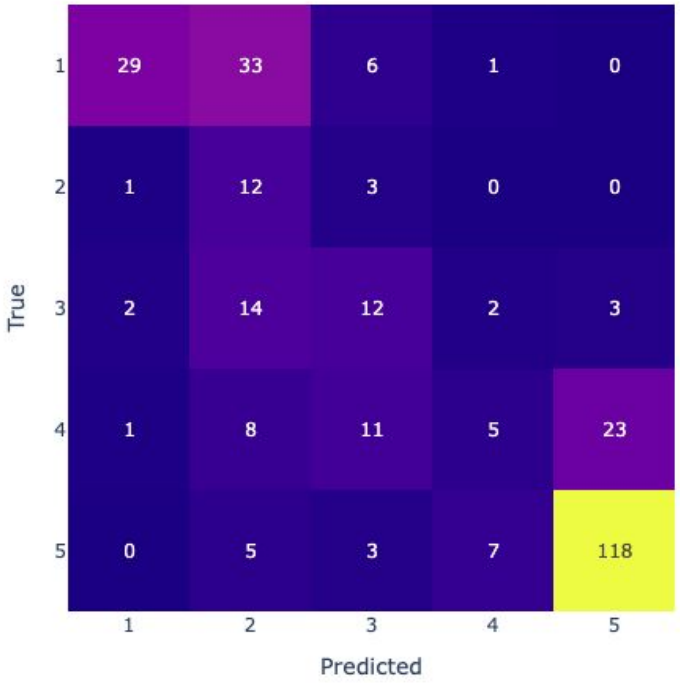
BERT



	precision	recall	f1-score	support
1	0.66	0.61	0.63	69
2	0.36	0.50	0.42	16
3	0.39	0.42	0.41	33
4	0.28	0.40	0.33	48
5	0.84	0.69	0.76	133
accuracy			0.59	299
macro avg	0.51	0.52	0.51	299
weighted avg	0.63	0.59	0.60	299

LLM GPT-4 Mini

- OpenAI
- Prompt Contexto + Exemplos



	precision	recall	f1-score	support
1	0.88	0.42	0.57	69
2	0.17	0.75	0.27	16
3	0.34	0.36	0.35	33
4	0.33	0.10	0.16	48
5	0.82	0.89	0.85	133
accuracy			0.59	299
macro avg	0.51	0.51	0.44	299
weighted avg	0.67	0.59	0.59	299

Resultados e Conclusão

	model	accuracy	f1_score
0	SVM + BoW	0.622074	0.365804
1	SVM + TFIDF unigrams	0.645485	0.364076
2	SVM + TFIDF bigrams	0.645485	0.366712
3	SVM + Word2Vector CBOW 300	0.612040	0.337184
4	SVM + Word2Vector SKIP 300	0.622074	0.352192
5	BERTimbau Base	0.585284	0.509269
6	LLM GPT-4 mini	0.588629	0.441002

Obrigado

Projeto NLP - Deep Learning

Lucas Accioly

Especialização
Deep Learning

Centro de Informática
UFPE

Universidade Federal de Pernambuco