Pontifícia Universidade Católica de Minas Gerais

Curso de Especialização – Ciência de dados e Big data:

Processamento de Linguagem Natural

Análise de sentimentos - Tweeter

Trabalho apresentado ao Instituto de Educação Continuada (IEC) da pós-graduação em Ciência de dados e Big data da PUC Minas, como trabalho na disciplina de Processamento de Linguagem Natural.

Aluna: Ana Carolina de Albuquerque Santos

Professora : Bárbara Silveira Fraga

Abril de 2022

Introdução

Análise de sentimentos ou mineração de opiniões refere-se ao uso de NLP (processamento de linguagem natural) cujo objetivo é identificar, extrair e quantificar a polaridade expressada nos dados. Através dos textos somos capazes de expressar alguns sentimentos como: amor, raiva, tristeza, alegria e medo por exemplo. Combinando NLP e machine learning podemos identificar se o sentimento que foi expressado em uma determinada rede social é positiva, negativa ou neutra, boa ou ruim, satisfatório ou insuficiente.

Com todos os tweets circulando a cada segundo, é difícil dizer se o sentimento por trás de um tweet específico impactará a marca de uma empresa ou de uma pessoa por ser viral (positivo) ou devastar o lucro porque atinge um tom negativo. Capturar o sentimento na linguagem é importante nestes tempos em que decisões e reações são criadas e atualizadas em segundos.

Objetivo

O objetivo deste trabalho foi classificar a observação utilizando metodologias dadas em aula, em um conjunto de classes discretas do aprendizado de máquina supervisionado regressão logística, a fim de gerar um modelo de análise de sentimentos. O objetivo do algoritmo foi aprender a mapear observações conhecidas para saídas corretas.

Base de dados

Este trabalho utilizou os dados: **https://www.kaggle.com/c/tweet-sentiment-extraction/overview .** Os dados foram extraídos da plataforma Data for Everyone da Figure Eight. O conjunto de dados é intitulado análise de sentimento: identifica emoção em tweets de texto com rótulos de sentimento existentes. Abaixo a pré-visualização do banco de dados (teste+treinamento) utilizado no trabalho:

```
train.append(test)
```

	textID	text	selected_text	sentiment
0	cb774db0d1	I'd have responded, if I were going	I'd have responded, if I were going	neutral
1	549e992a42	Sooo SAD I will miss you here in San Diego!!!	Sooo SAD	negative
2	088c60f138	my boss is bullying me...	bullying me	negative
3	9642c003ef	what interview! leave me alone	leave me alone	negative
4	358bd9e861	Sons of ****, why couldn`t they put them on t...	Sons of ****,	negative
...	...	...	...	...
3529	e5f0e6ef4b	its at 3 am, im very tired but i can`t sleep ...	NaN	negative
3530	416863ce47	All alone in this old house again. Thanks for...	NaN	positive
3531	6332da480c	I know what you mean. My little dog is sinkin...	NaN	negative
3532	df1baec676	_sutra what is your next youtube video gonna b...	NaN	positive
3533	469e15c5a8	http://twitpic.com/4woj2 - omgssh ang cute n...	NaN	positive

31015 rows × 4 columns

Nos dados de entrada temos um conjunto de dados de observações e seus respectivos rótulos. Cada entrada (*text*) está associada a alguma saída correta (*sentiment*) categorizada conforme abaixo:

```
target = train['sentiment'].replace(['neutral', 'negative',
'positive'],[0, 1, 2])
```

Metodologia

1. **Os dados foram pré-processados da seguinte forma:**.

- Tokenização: O texto foi dividido em uma lista de tokens;
- Stopwords: foram retiradas as palavras e termos frequentes que não tem relevância nos dados;
- Removido links, pontos, vírgulas, ponto e vírgulas dos tweets;
- Foram removidos acentos e transformados para minúscula as palavras.

```
def pre_processamento_texto(corpus):
  #print("#Tokenização")
  corpus_alt = re.findall(r"\w+(?:'\w+)?|[^\w\s]", corpus)
  #lowcase
  corpus_alt = [ t.lower() for t in corpus_alt ]
  #print('remove stopwords')
  english_stops = stopwords.words('english')
  corpus_alt = [t for  t in corpus_alt if t not in
english_stops]
  #print('remove numeros')
  corpus_alt = [re.sub(r'\d','',t) for t in  corpus_alt]
```

```
    #print('remove pontuação')
    corpus_alt = [t for  t in corpus_alt if t not in
string.punctuation]
    #print('remove acentos')
    corpus_alt = [unidecode(t) for t in  corpus_alt]
    corpus_alt = ' '.join(corpus_alt).lower()
    return corpus_alt


train['text_clean'] = train['text'].progress_apply(lambda
x:pre_processamento_texto(str(x)))
```

A visualização da etapa de pré processamento foi feita por word-cloud.

2.O bag of Words foi utilizado para representar o texto:

```
vect_bag = CountVectorizer()

X_bag = vect_bag.fit_transform(train['text'])
```

3.A base de dados foi dividida em dados em treino e teste:

```
X_train_bow, X_test_bow, y_train_bow, y_test_bow =
train_test_split(X_bag, target,random_state=123)
```

4. Foram aplicados os algoritmos de classificação de regressão logística e Naive Bayes:

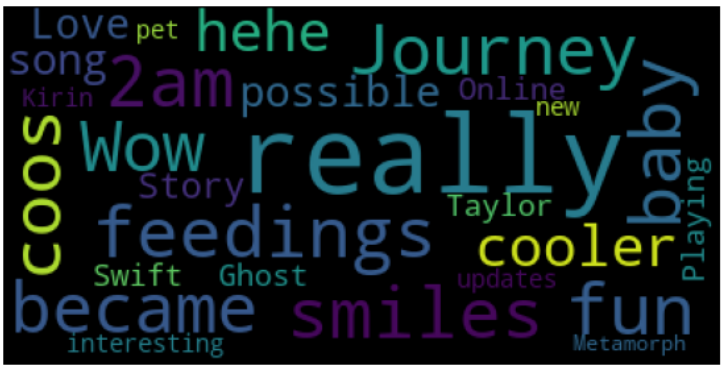
```
modelo_bow = LogisticRegression(intercept_scaling=0)

modelo_bow.fit(X_train_bow,  y_train_bow)


modelo_NB = MultinomialNB()

modelo_NB.fit(X_train_bow,  y_train_bow)
```

5. O modelo foi avaliado com base nas métricas de precisão e recall:

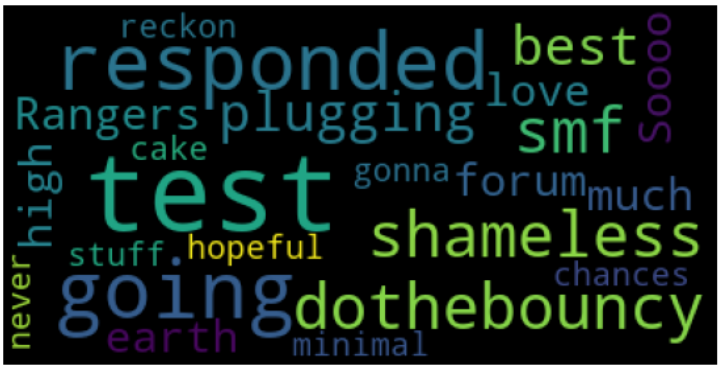
```
y_predi = modelo_bow.predict(X_test_bow)
print(classification_report(y_test_bow, y_predi))
```

Visualização dados dados

Após a aplicação das técnicas foi criado um Word-Cloud para a visualização do pré-processamento de cada classificação (positivo, negativo, neutro) dos twitters na base de dados.
Word-Cloud para twitters classificados com sentimento positivo:

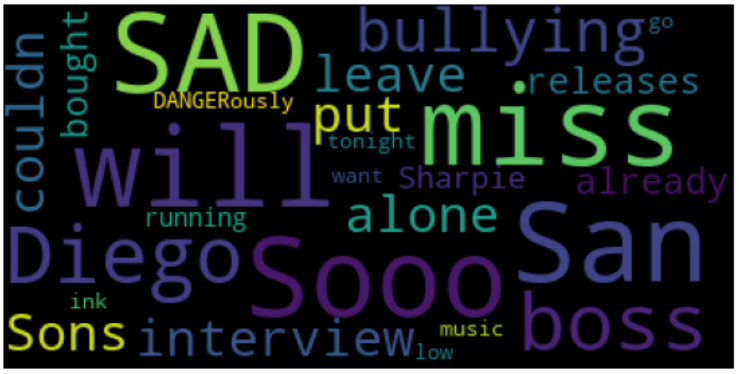


Word-Cloud para twitters classificados com sentimento neutro:



:

Word-Cloud para twitters classificados com sentimento negativo:



Resultados:

Resultado para a Regressão Logística:

		precision	recall	f1-score	support
	0	0.64	0.73	0.68	2785
	1	0.70	0.62	0.66	1935
	2	0.78	0.72	0.75	2150
accuracy				0.70	6870
macro avg		0.71	0.69	0.70	6870
weighted avg		0.70	0.70	0.70	6870

Resultado para a Naive Bayes:

		precision	recall	f1-score	support
	0	0.59	0.70	0.64	2785
	1	0.67	0.56	0.61	1935
	2	0.74	0.67	0.70	2150
accuracy				0.65	6870
macro avg		0;67	0.64	0.65	6870
weighted avg		0.66	0.65	0.65	6870

O algoritmo *Logistic Regression* obteve a maior acurácia, precisão e recall em todas as classes de sentimento analisadas em comparação ao Naive Bayes. Com um recall de 73% para sentimento neutro, 62% para *tweets* negativos e 72% para *tweets* positivos.

Considerações finais

No trabalho foi utilizado o método de aprendizagem *Logistic Regression* para classificar os *tweets*. Esse método é independente do idioma e pode ser utilizado para classificar texto poluído como é característico dos *tweets*. Os resultados obtidos acerca do pré-processamento, treino e testes demonstram resultados satisfatórios quando considerados aspectos de classificação realizada por humanos.

Anexo:

https://colab.research.google.com/drive/1RS3s9KljKBtoEjua9rEJv1RbjH40WnBe#scrollTo=FCf06OENqTKx

Referências:

https://minerandodados.com.br/analise-de-sentimentos-utilizando-dados-do-twitter/