MINERAÇÃO DE TEXTO

Parte 3 - ANÁLISE DE SENTIMENTOS COM CLASSIFICAÇÃO BINÁRIA

Importação das bibliotecas:

In [1]:
```
import nltk
import matplotlib.pyplot as plt
import pandas as pd
import re

# Importando o Corpora e funções do NLTK...
from nltk.corpus import stopwords
from nltk.corpus import opinion_lexicon
from nltk.tokenize import word_tokenize

nltk.download('stopwords', quiet=True)
nltk.download('opinion_lexicon', quiet=True)
nltk.download('punkt', quiet=True)

# ... e do Matplotlib
plt.style.use('ggplot')
```

Importação do texto de Stevenson:

In [2]:
```
df = pd.read_csv('data/TheAmateurEmigrant.txt',sep='\t')\
    .dropna()

df.head()
```

Out[2]:

	text
0	THE AMATEUR EMIGRANT
1	THE SECOND CABIN
2	I first encountered my fellow-passengers on th...
3	Thence we descended the Clyde in no familiar s...
4	on each other as on possible enemies. A few S...

Preparação do texto (limpeza e tokenização):

In [3]:
```
def clean_text(text):
    text = text.lower()
    text = text.replace("'", '')
    text = re.sub(r'[^\w]', ' ', text)
    text = re.sub(r'\s+', ' ', text)
    text = text.strip()
    return text

df['text'] = df['text'].map(clean_text)
df['text'] = df['text'].map(word_tokenize)
```

```
df.head()
```

Out[3]:

	text
0	[the, amateur, emigrant]
1	[the, second, cabin]
2	[i, first, encountered, my, fellow, passengers...
3	[thence, we, descended, the, clyde, in, no, fa...
4	[on, each, other, as, on, possible, enemies, a...

In [4]:

```
df = df.text.explode().to_frame('token')
df.head()
```

Out[4]:

	token
0	the
0	amateur
0	emigrant
1	the
1	second

In [5]:

```
df.token.value_counts().head()
```

Out[5]:

```
the    1548
and    1079
of      860
a       808
to      687
Name: token, dtype: int64
```

In [6]:

```
stopwords = set(stopwords.words('english'))
```

In [7]:

```
df = df[~df.token.isin(stopwords)]
```

In [8]:

```
df.token.value_counts().head()
```

Out[8]:

```
one      129
man       94
like      70
would     67
said      56
Name: token, dtype: int64
```

Classificação dos sentimentos com o Opinion Lexicon

In [9]:

```
sentiment_lexicon = {
    **{w: 'positivo' for w in opinion_lexicon.positive()},
    **{w: 'negativo' for w in opinion_lexicon.negative()}
}
```

```
df['sentiment'] = df['token'].map(sentiment_lexicon)
df = df[~df.sentiment.isna()]

df.head()
```

Out[9]:

	token	sentiment
3	askance	negativo
4	enemies	negativo
5	friendly	positivo
6	suspicion	negativo
7	supreme	positivo

In [10]:
```
df.token.value_counts().head()
```

Out[10]:
```
like      70
good      46
well      32
work      27
better    23
Name: token, dtype: int64
```

Sumarizando as palavras por sentimentos

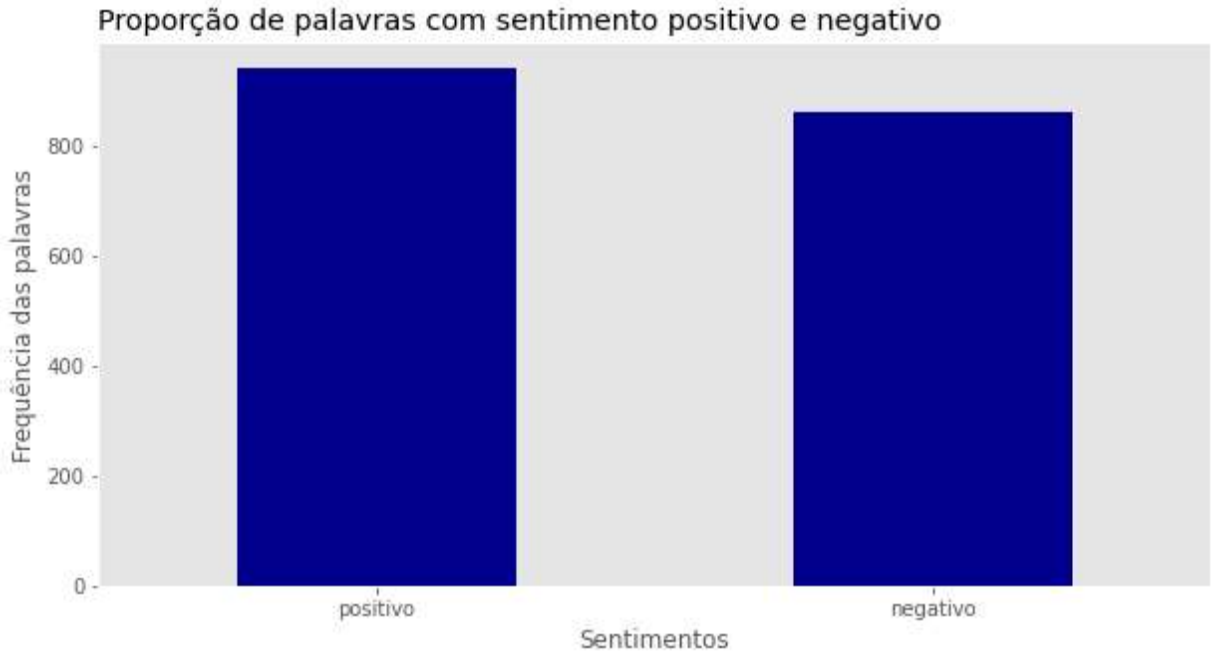
In [11]:
```
summary_df = df.sentiment.value_counts().to_frame('n')
summary_df['prop'] = summary_df['n'] / summary_df.n.sum()

summary_df.round(3)
```

Out[11]:

	n	prop
positivo	942	0.522
negativo	863	0.478

Visualização gráfica da Classificação Binária dos sentimentos

In [12]:
```
summary_df.n.plot.bar(legend=False, figsize=(10, 5), grid=False, color='darkblue')
plt.xlabel('Sentimentos')
plt.ylabel('Frequência das palavras')
plt.title('Proporção de palavras com sentimento positivo e negativo', loc='left')
plt.xticks(rotation=0);
```



O texto de "The Amateur Emigrant" mostra um certo equilíbrio entre a quantidade de palavras **positivas** (**52,2%**) e **negativas** (**47,8%**), o que é refletido no gráfico acima. Por se tratar de uma história real, ambientada num navio cuja tripulação era composta, em sua maioria, por homens desempregados que estavam em busca de melhores condições de vida na América, esse resultado pode causar surpresa em quem esperava que o tom da história fosse mais negativo. No entanto, Stevenson soube dosar a sua narrativa com fatos sobre como esses viajantes se divertiam, a camaradagem que existia entre eles e a esperança que depositavam no Novo Mundo.