

## Análise de Sentimentos: Material de Aula

### Introdução

A análise de sentimentos é uma área do Processamento de Linguagem Natural (PLN) que busca identificar e extrair opiniões e emoções expressas em textos. Ela tem se tornado cada vez mais relevante em diversas áreas, como marketing, pesquisa de opinião, análise de redes sociais e atendimento ao cliente.

### Conceitos Básicos

- **Sentimento:** Refere-se à polaridade de uma opinião, que pode ser positiva, negativa ou neutra.
- **Opinião:** É a expressão de um ponto de vista, crença ou sentimento sobre um determinado assunto.
- **Polaridade:** Indica a direção do sentimento, ou seja, se é positivo, negativo ou neutro.
- **Subjetividade:** Refere-se à presença de opiniões e emoções em um texto.
- **Objetividade:** Refere-se à presença de fatos e informações neutras em um texto.

### Técnicas de Análise de Sentimentos

Existem diversas técnicas para realizar análise de sentimentos, que podem ser divididas em três categorias principais:

1. **Baseadas em léxico:** Utilizam dicionários de palavras com pontuações de sentimento predefinidas para calcular a polaridade de um texto.
2. **Baseadas em aprendizado de máquina:** Treinam modelos de aprendizado de máquina com dados rotulados para classificar o sentimento de novos textos.
3. **Baseadas em aprendizado profundo:** Utilizam redes neurais profundas para capturar padrões complexos e sutilezas da linguagem, como o contexto e a ironia.

### Aplicações da Análise de Sentimentos

- **Monitoramento de redes sociais:** Para acompanhar a reputação de marcas e produtos.
- **Análise de feedback de clientes:** Para identificar pontos fortes e fracos de produtos e serviços.

- **Pesquisa de opinião:** Para entender a opinião pública sobre temas políticos e sociais.
- **Deteção de fake news:** Para identificar textos com forte carga emocional e baixa credibilidade.
- **Análise de tendências de mercado:** Para prever o comportamento dos consumidores.

### Desafios da Análise de Sentimentos

- **Ambiguidade da linguagem:** Palavras e frases podem ter diferentes significados dependendo do contexto.
- **Ironia e sarcasmo:** Difíceis de detectar e interpretar pelos algoritmos.
- **Linguagem informal e gírias:** Variam muito e podem não estar presentes nos dicionários de sentimentos.
- **Multilinguismo:** A análise de sentimentos em diferentes idiomas pode ser complexa devido às diferenças culturais e linguísticas.

# Importando as bibliotecas necessárias

```
import nltk
```

```
from nltk.sentiment import SentimentIntensityAnalyzer
```

# Baixando os recursos do NLTK (necessário apenas na primeira vez)

```
nltk.download('vader_lexicon')
```

# Criando um analisador de sentimentos

```
analyzer = SentimentIntensityAnalyzer()
```

# Função para analisar o sentimento de um texto

```
def analisar_sentimento(texto):
```

```
    pontuacao = analyzer.polarity_scores(texto)
```

```
    if pontuacao['compound'] >= 0.05:
```

```
        return 'positivo'
```

```
elif pontuacao['compound'] <= -0.05:
```

```
    return 'negativo'
```

```
else:
```

```
    return 'neutro'
```

```
# Exemplo de uso
```

```
texto = "Adorei o novo filme! A história é incrível e os atores são ótimos."
```

```
sentimento = analisar_sentimento(texto)
```

```
print(f"O sentimento do texto é: {sentimento}")
```

```
Linguagem R
```

```
# Instalando e carregando os pacotes necessários
```

```
install.packages(c("tidytext", "syuzhet", "dplyr"))
```

```
library(tidytext)
```

```
library(syuzhet)
```

```
library(dplyr)
```

```
# Função para analisar o sentimento de um texto
```

```
analisar_sentimento <- function(texto) {
```

```
    sentimento <- get_sentiment(texto, method = "syuzhet")
```

```
    if (sentimento > 0) {
```

```
        return("positivo")
```

```
    } else if (sentimento < 0) {
```

```
        return("negativo")
```

```
    } else {
```

```
        return("neutro")
```

```
    }
```

```
}
```

```
# Exemplo de uso
```

```
texto <- "Adorei o novo filme! A história é incrível e os atores são ótimos."
```

```
sentimento <- analisar_sentimento(texto)
```

```
print(paste("O sentimento do texto é:", sentimento))
```

A análise de sentimentos é uma área em constante evolução, com novas técnicas e aplicações surgindo a cada dia. É importante lembrar que os resultados da análise de sentimentos devem ser interpretados com cautela, levando em consideração os desafios e limitações da área.

Vamos desenvolver um código de análise de sentimentos que pode ser aplicado a diversas aplicações, como análise de avaliações de produtos, comentários de redes sociais ou feedback de clientes. Usaremos Python e a biblioteca nltk para realizar a análise.

```
import nltk
```

```
from nltk.sentiment import SentimentIntensityAnalyzer
```

```
import pandas as pd
```

```
# Baixando os recursos do NLTK (necessário apenas na primeira vez)
```

```
nltk.download('vader_lexicon')
```

```
def analisar_sentimento(texto):
```

```
    """Analisa o sentimento de um texto e retorna a polaridade."""
```

```
    analyzer = SentimentIntensityAnalyzer()
```

```
    pontuacao = analyzer.polarity_scores(texto)
```

```
    if pontuacao['compound'] >= 0.05:
```

```
        return 'positivo'
```

```
    elif pontuacao['compound'] <= -0.05:
```

```
        return 'negativo'
```

```
else:
```

```
    return 'neutro'
```

```
def analisar_sentimentos_dataframe(dataframe, coluna_texto):
```

```
    """Analisa os sentimentos de um dataframe e adiciona uma coluna com a polaridade."""
```

```
    dataframe['sentimento'] = dataframe[coluna_texto].apply(analisar_sentimento)
```

```
    return dataframe
```

```
# Exemplo de uso com um DataFrame do pandas
```

```
dados = {
```

```
    'texto': [
```

```
        "Adorei o produto, superou minhas expectativas!",
```

```
        "O serviço foi péssimo, não recomendo.",
```

```
        "É um produto ok, nada de especial.",
```

```
        "Estou muito feliz com a compra!",
```

```
        "Que decepção, perdi meu dinheiro."
```

```
    ]
```

```
}
```

```
df = pd.DataFrame(dados)
```

```
df_com_sentimentos = analisar_sentimentos_dataframe(df, 'texto')
```

```
print(df_com_sentimentos)
```

```
# Exemplo de uso com um arquivo CSV
```

```
# Suponha que você tenha um arquivo 'avaliacoes.csv' com uma coluna 'comentario'
```

```
# df_csv = pd.read_csv('avaliacoes.csv')
```

```
# df_csv_com_sentimentos = analisar_sentimentos_dataframe(df_csv,
'comentario')

# print(df_csv_com_sentimentos)
```

### **Explicação:**

1. **Importação das bibliotecas:** Importamos nltk, SentimentIntensityAnalyzer e pandas.

2. **Função analisar\_sentimento(texto):**

- Recebe um texto como entrada.
- Cria um objeto SentimentIntensityAnalyzer.
- Calcula a pontuação de polaridade do texto usando polarity\_scores().
- Retorna a polaridade do texto ('positivo', 'negativo' ou 'neutro') com base na pontuação composta.

3. **Função analisar\_sentimentos\_dataframe(dataframe, coluna\_texto):**

- Recebe um dataframe do pandas e o nome da coluna que contém os textos a serem analisados.
- Aplica a função analisar\_sentimento() a cada texto na coluna especificada.
- Adiciona uma nova coluna chamada 'sentimento' ao dataframe com a polaridade de cada texto.
- Retorna o dataframe com a nova coluna de sentimentos.

4. **Exemplo de uso:**

- Criamos um dataframe de exemplo com alguns textos.
- Chamamos a função analisar\_sentimentos\_dataframe() para analisar os sentimentos dos textos no dataframe.
- Imprimimos o dataframe com a nova coluna de sentimentos.
- Foi comentado um exemplo de uso com arquivo CSV, onde seria possível realizar a leitura de um arquivo CSV, e realizar o mesmo processo de análise de sentimentos.

### **Como usar:**

### 1. Instale as bibliotecas:

Bash

```
pip install nltk pandas
```

### 2. Execute o código Python.

### 3. Adapte o código para sua aplicação:

- Se você tiver dados em um arquivo CSV, use `pd.read_csv()` para carregar os dados em um dataframe.
- Se você tiver dados em outro formato, adapte o código para carregar os dados em um dataframe do pandas.
- Ajuste o nome da coluna que contém os textos a serem analisados na chamada da função `analisar_sentimentos_dataframe()`.

### Observações:

- A biblioteca `nltk` utiliza um léxico predefinido para determinar a polaridade dos textos. Isso pode não ser perfeito para todos os casos, especialmente para textos com gírias, ironia ou sarcasmo.
- Para melhorar a precisão da análise de sentimentos, você pode treinar um modelo de aprendizado de máquina com seus próprios dados rotulados.
- É possível que o léxico utilizado, esteja em inglês, e que, portanto, traduções, ou palavras em português, não sejam analisadas corretamente.

Vamos criar um exemplo real de análise de sentimentos no Orange, utilizando avaliações de filmes como nosso conjunto de dados.

### 1. Preparação dos Dados:

#### • Conjunto de Dados:

- Suponha que temos um arquivo CSV chamado "avaliacoes\_filmes.csv" com duas colunas: "filme" (nome do filme) e "avaliacao" (texto da avaliação).
- Exemplo de dados:

Snippet de código

```
filme,avaliacao
```

```
"Vingadores: Ultimato","Filme incrível, efeitos especiais impressionantes!"
```

"O Poderoso Chefão","Uma obra-prima do cinema, atuações impecáveis."

"Transformers: O Último Cavaleiro","Filme horrível, roteiro confuso e atuações ruins."

"Parasita","Filme genial, uma crítica social poderosa."

"Minha Mãe é uma Peça 3","Muito engraçado, dei muita risada."

- **Importação:**

- Abra o Orange e arraste o widget "File" para a tela.
- Clique duas vezes no widget "File" e selecione o arquivo "avaliacoes\_filmes.csv".
- Certifique-se de que as colunas "filme" e "avaliacao" estejam corretamente identificadas.

## **2. Pré-processamento de Texto:**

- **Widget "Preprocess Text":**

- Arraste o widget "Preprocess Text" para a tela e conecte-o à saída do widget "File".
- Clique duas vezes no widget "Preprocess Text" e configure as seguintes etapas:
  - "Tokenization": "Words"
  - "Lowercase": marcado.
  - "Remove punctuation": marcado.
  - "Remove stopwords": marcado.
  - "Stemming": "Porter stemmer" (ou "Lemmatization").
- Aplique as alterações.

## **3. Análise de Sentimentos com Script Python:**

- **Widget "Python Script":**

- Arraste o widget "Python Script" para a tela e conecte-o à saída do widget "Preprocess Text".
- Clique duas vezes no widget "Python Script" e insira o seguinte código:



```
import pandas as pd

from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

analyzer = SentimentIntensityAnalyzer()

def analyze_sentiment(text):

    scores = analyzer.polarity_scores(text)

    compound_score = scores['compound']

    if compound_score >= 0.05:

        return 'positive'

    elif compound_score <= -0.05:

        return 'negative'

    else:

        return 'neutral'

data = in_data.copy()

data['sentimento'] = data['avaliacao'].apply(analyze_sentiment)

out_data = data
```

- **Observações:**

- O código utiliza a biblioteca vaderSentiment para calcular a polaridade do sentimento.
- A função analyze\_sentiment() retorna "positive", "negative" ou "neutral" com base na pontuação composta.
- O código cria uma nova coluna chamada "sentimento" no conjunto de dados.

#### **4. Visualização dos Resultados:**

- **Widget "Distributions":**

- Arraste o widget "Distributions" para a tela e conecte-o à saída do widget "Python Script".
- Clique duas vezes no widget "Distributions" e selecione a coluna "sentimento" para visualizar a distribuição dos sentimentos.
- **Widget "Data Table":**
  - Arraste o Widget "Data Table" para a tela e conecte-o a saída do Widget "Python Script".
  - Desta forma é possível visualizar os dados originais, e a coluna de sentimento gerada.
- **Widget "Scatter Plot":**
  - Caso haja dados numéricos nas avaliações, ou que sejam gerados, é possível utilizar o "Scatter Plot" para visualizar a relação entre dados numéricos e o sentimento.

## **5. Interpretação dos Resultados:**

- O widget "Distributions" mostrará a quantidade de avaliações positivas, negativas e neutras.
- O widget "Data Table" mostrará os dados da tabela, com a coluna de sentimento adicionada.
- Os gráficos gerados, auxiliam na interpretação dos dados.

## **Observações:**

- Este é um exemplo básico e pode ser adaptado para outros conjuntos de dados e aplicações.
- Você pode experimentar com diferentes etapas de pré-processamento e técnicas de análise de sentimentos para melhorar os resultados.
- Lembre-se de instalar as bibliotecas necessárias, dentro do orange, ou pelo Anaconda Prompt.