Bootcamp: Cientista de Dados

Desafio

Módulo 3	CDD – Coleta e Obtenção de Dados

Objetivos

- ✓ Realizar análise de sentimento dos tweets coletados com Python/tweepy.
- ✓ Realizar scraping e crawling em páginas web com Python/scrapy.

Enunciado

A coleta de dados na web é hoje uma das grandes necessidades das empresas. Duas técnicas comuns de coleta é a utilização de APIs para coletar dados em redes sociais e a raspagem/rastreamento de dados na web (web crawling e web scraping). O desafio consiste em duas grandes práticas de coleta de dados na web:

A primeira será uma implementação em Python para a análise de sentimento, envolvendo textos de *tweets* coletados via API e o pacote *tweepy*. A coleta de tweets e a análise de polaridade (sentimento) utilizando o Python foi apresentada na videoaula “Aula 4.4. Coleta de dados no Twitter: Exemplo utilizando a linguagem Python”.

A segunda será uma implementação em Python para realizar a raspagem (scraping) de dados na página de notícias do Governo Federal e a realização de algumas técnicas de mineração de textos. As práticas de web crawling e web scraping foram apresentadas nas videoaulas:

- Aula 4.7.1. Web Crawling: Exemplo com a biblioteca Scrapy da linguagem Python
- Aula 4.7.2. Web Crawling: Exemplo com a biblioteca Scrapy da linguagem Python
- Aula 4.8.1. Web Crawling: Exemplo de script utilizando a linguagem Python

- Aula 4.8.2. Web Crawling: Exemplo de script utilizando a linguagem
- Aula 4.8.3. Web Crawling: Exemplo de script utilizando a linguagem Python

Atividades

Os alunos deverão desempenhar as seguintes atividades:

Atividade 1: Implementar um script utilizando a linguagem Python e seu pacote *tweepy* para coletar um conjunto de *tweets* no idioma português por meio de API do Twitter. Opcionalmente, você pode salvar os tweets coletados em um arquivo csv ou json, ou ainda em um banco de dado MongoDB. Para coletar os tweets, devem ser usadas as seguintes configurações:

a) Palavras chave e/ou hashtags: 'home office' ou 'trabalho remoto' ou #homeoffice ou #trabalhoremoto.

b) Texto completo do tweet com 280 caracteres. (tweet_mode='extended').

c) Quantidade de tweets coletados: 18000.

d) Tweets mais populares. (result_type="popular").

Observação 1: Deve-se ter uma conta no Twitter com acesso de desenvolvedor. Além de criar sua aplicação e gerar as credenciais de acesso. (Vide Trabalho Prático)

Observação 2: A API free só permite recuperar os tweets dos últimos 7 dias e limita o número de *tweets* recuperados a 100 tweets por chamada, apesar de definirmos o número para 18000, não teremos este número de *tweets*. (Detalhes na aula interativa)

Atividade 2: Utilizando a linguagem Python e os *tweets* coletados na atividade anterior, implementar no script anterior ou criar um novo script, uma análise de sentimento para visualizar as seguintes análises:

- Tweets mais curtidos e tweets mais retweetados (compartilhados).
- Identificar a fonte de origem (dispositivo) dos tweets.

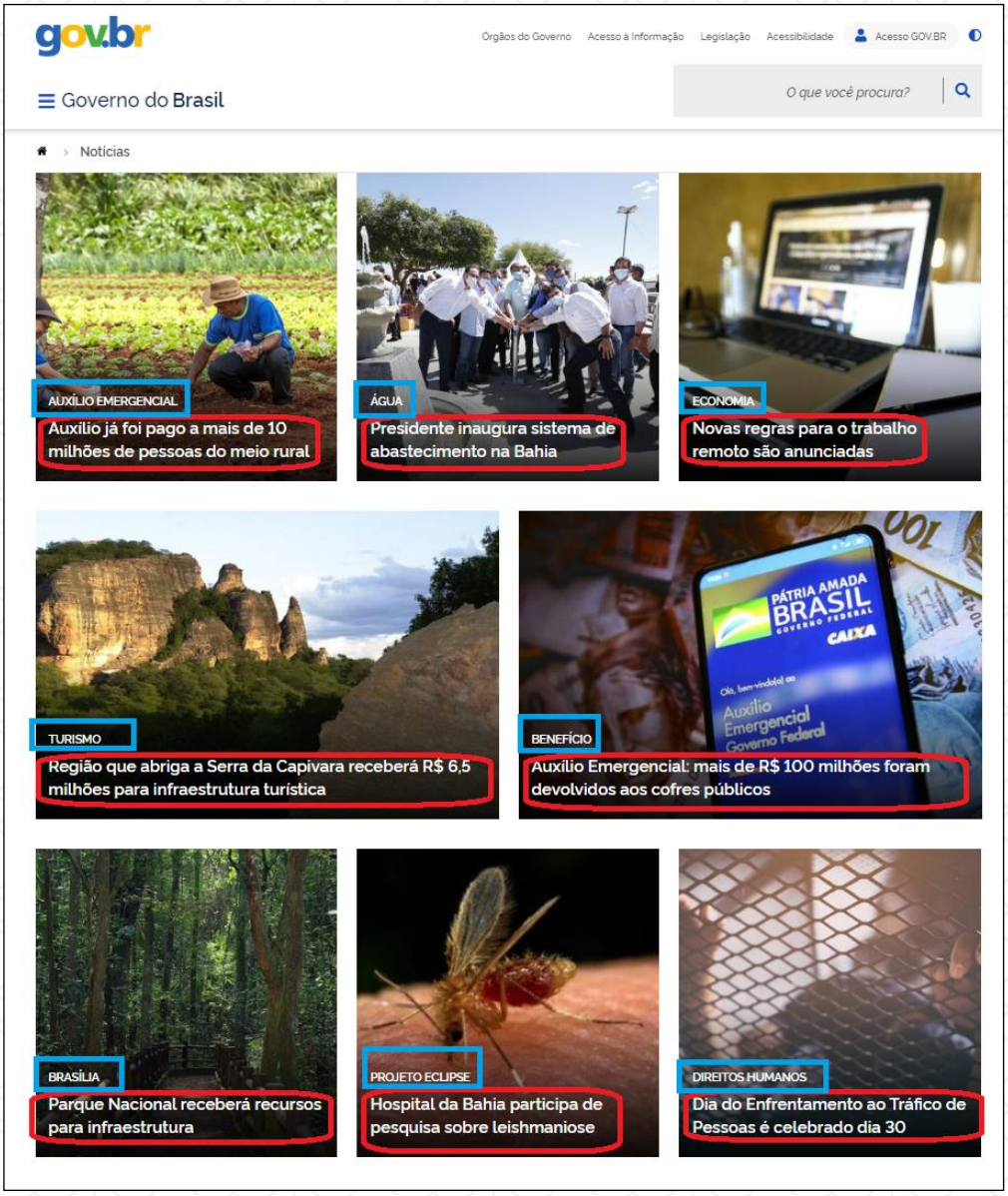
- Realizar a análise de polaridades dos tweets, categorizando-os como positivo, negativo e neutro, sendo um tweet que represente sentimento positivo (polaridade > 0), negativo (polaridade < 0) ou neutro (polaridade = 0).
- Criar uma nuvem de palavras com as palavras mais frequentes.
- Avaliar o volume de tweets publicados por data.
- Criar um mapa de calor dos tweets utilizando a localização declarada pelos usuários.

Observação 1: Aprenderemos na 2ª Aula Interativa como fazer cada um deste itens.

Atividade 3: Utilizando a linguagem Python e o pacote Scrapy, crie um projeto para realizar a raspagem de dados da página de notícias do governo federal e salvar em um arquivo csv. O endereço da página é: https://www.gov.br/pt-br/noticias. A Figura 1 apresenta um exemplo de como a página é organizada. As informações que devem ser raspadas da página inicial são:

- Título(s) e url da página inicial.
- Assunto de cada notícia (em destaque com o retângulo azul na Figura 1).
- Título de cada notícia (em destaque circulado em vermelho na Figura 1).
- URL de cada notícia.

Figura 1 - Exemplo da tela inicial da página de notícias do governo federal.



Atividade 4: Utilizando Python e Scrapy, altere o projeto da atividade 3 para realizar o rastreamento para as notícias selecionadas na página inicial. A Figura 2 abaixo apresenta um exemplo de como uma página de notícia é organizada. As informações, em destaque na Figura 2, que devem ser raspadas das páginas de cada notícia são:

- Título(s) da página;
- Url da página;
- Assunto de cada notícia;
- Título de cada notícia;
- Subtítulo de cada notícia;
- Data/hora da publicação;
- Texto de cada notícia;
- Autor da notícia;
- Categoria da notícia;
- Tags da notícia.

Observação: Salve todos os itens em um arquivo csv.

Figura 2: Exemplo da página de notícia rastreada.

Atividade 5: Utilizando Python e o pacote NLTK, faça a mineração do texto das notícias lidas na atividade 4, que foram salvas no arquivo csv. Serão realizadas as seguintes tarefas de mineração:

- Remoção de Stopwords.

- Remoção de valores numéricos.
- Tokenização: dividir o texto em uma lista de tokens/palavras.
- Gerar bag of words (saco de palavras) dos tokens.

Observação: A mineração de texto será demonstrada na Segunda Aula Interativa.

Respostas Finais

Os alunos deverão desenvolver a prática e, depois, responder às seguintes questões objetivas: