Web scraping

Introduce: Web scraping is an automated method used to extract large amounts of data from websites. The data on the websites are unstructured. Web scraping helps collect these unstructured data and store it in a structured form. There are different ways to scrape websites such as online Services, APIs or writing your own code. In this article, we'll see how to implement web scraping with python.

1. Import libraries:

```
In [1]: #imports here
        from selenium import webdriver
        from selenium.webdriver.common.keys import Keys
        from selenium.webdriver.support import expected_conditions as EC
        from selenium.webdriver.common.by import By
        from selenium.webdriver.support.wait import WebDriverWait
        from selenium import webdriver
        from selenium.webdriver.common.by import By
        import requests
        import io
        from PIL import Image
        import time
```

2. Specify the path to chromedriver.exe:

```
In [2]: #specify the path to chromedriver.exe (download and save on your computer)
        PATH = "C:\\Users\\Administrator\\Desktop\\chromedriver.exe"

        wd = webdriver.Chrome(PATH)
```

3. Open scraping page in chromedriver:

```
In [3]: chrome_options = webdriver.ChromeOptions()
        prefs = {"profile.default_content_setting_values.notifications" : 2}
        chrome_options.add_experimental_option("prefs",prefs)
        wd = webdriver.Chrome(chrome_options=chrome_options)
        wd.get("https://www.youtube.com/watch?v=N6EHKn6SK7k")
```

4. Pulling data out of HTML and XML files:

```
In [5]: import pandas as pd
        import requests
        from bs4 import BeautifulSoup
        #wd.get('https://www.youtube.com/watch?v=N6EHKn6SK7k')

        for x in range(1, 4):
            wd.execute_script("window.scrollTo(0,document.body.scrollHeight)")
            time.sleep(5)

        soup = BeautifulSoup(wd.page_source, 'html.parser')

        #titles=soup.find_all('div',attrs={'class':'style-scope ytd-expander'})
        comments=soup.find_all('div',attrs={'class':'style-scope ytd-expander'})

        #titleloop=[title.text for title in titles]
        commentloop=[comment.text for comment in comments]
```

5. Make a dictionary:

```
[8]: data={'comment':commentloop}

[9]: data
```

6. Create data frame:

```
In [11]: data1=pd.DataFrame(data,columns=['comment'])
         data1
```

7. Remove new line:

```
In [13]: import re
         def replace_new_line(text):
             return re.sub(r'(\n)', ' ', text)

In [14]: data1['comment']= data1['comment'].apply(lambda x: replace_new_line(x))
```

8. Remove under line:

```
[5]: def replace_under_line(text):
         return re.sub(r'_', '', text)

[6]: data1['comment']= data1['comment'].apply(lambda x: replace_under_line(x))
```

9. Import nltk Library:

```
In [18]: import nltk
         nltk.download('punkt')
         nltk.download('wordnet')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\Administrator\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\Administrator\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
```

10. Tokenize comment column data:

```
In [19]: data1['comment']= data1['comment'].apply(lambda x: nltk.word_tokenize(x))
```

11. Remove stop word:

```
In [20]: stopwords=nltk.corpus.stopwords.words('english')

In [21]: def remove_stopWord(text):
             output=[i for i in text if i not in stopwords]
             return output
```

12. Lemmatize the comment data:

```
In [23]: from nltk.stem import WordNetLemmatizer
         wordnet= WordNetLemmatizer()
```

```
In [24]: def lemmatizer(text):
             lemm_text = [wordnet.lemmatize(word) for word in text]
             return lemm_text
```

```
In [25]: data1['comment']=data1['comment'].apply(lambda x: lemmatizer(x))
```

13. Join the lemmatize data:

```
In [26]: data1['lemmatized'] = data1.comment.apply(lambda x: ' '.join(x))
```

14. Import IMBD data set:

```
In [34]: data=pd.read_csv('IMDB Dataset.csv')
```

15. Preprocess the IMBD data set:

```
In [28]: def remove_punctuation(text):
             punctuationfree="".join([i for i in text if i not in string.punctuation])
             return punctuationfree
```

```
In [29]: def lower(text):
             return text.lower()
```

```
In [30]: def tokenization(text):
             tokens =  nltk.word_tokenize(text)
             return tokens
```

```
In [31]: def remove_stopWord(text):
             output=[i for i in text if i not in stop]
             return output
```

```
In [32]: def lemmatizer(words):
             return [wordnet.lemmatize(word) for word in words]
```

```
In [33]: def preprocess(text):
             text1=remove_punctuation(text)
             text2=lower(text1)
             word=tokenization(text2)
             words=remove_stopWord(word)
             final=lemmatizer(words)
             return final
```

16. TFIDFI vectorize data:

```
In [39]: from sklearn.feature_extraction.text import TfidfVectorizer
         from sklearn.model_selection import train_test_split
         from sklearn.svm import LinearSVC
         from sklearn.metrics import classification_report
```

```
In [40]: tfidf=TfidfVectorizer(max_features=10000)
```

17. Split data feature and label:

```
In [41]: x=data["lemmatized"]
         y=data["sentiment"]
```

18. Train the model use IDMB data:

```
2]: x=tfidf.fit_transform(x)
    clf=LinearSVC()
    clf.fit(x,y)
```

19. Predict original data label:

```
In [43]: tf=data1["lemmatized"]
```

```
In [44]: vc=tfidf.transform(tf)
```

```
In [45]: y_pred=clf.predict(vc)
```

20. Make data frame with predict label:

```
In [46]: data2=pd.DataFrame(y_pred)
         data2
```

Out[46]:

	0
0	positive
1	negative
2	positive
3	positive
4	positive
...	...
81	negative
82	negative
83	negative
84	negative
85	positive

86 rows × 1 columns

21. Concat predict and old data set:

```
In [47]: data = pd.concat([data1, data2], axis=1)
         data
```

Out[47]:

	comment	lemmatized	0
0	[Checkout, full, review, Redmi, Note, 11, Pro,...	Checkout full review Redmi Note 11 Pro 5G smar...	positive
1	[89.9, %, comment, like, ;, never, fails, make...	89.9 % comment like ; never fails make laugh c...	negative
2	[3:30, Sir, ,, checked, comparison, Snapdragon...	3:30 Sir , checked comparison Snapdragon 695 v...	positive
3	[You, one, indian, tech, industry, uploading, ...	You one indian tech industry uploading good vi...	positive
4	[That, 's, Simple, Wow, Presentation, 👏🔥👏😍]	That 's Simple Wow Presentation 👏🔥👏😍	positive
...	...	...	...
81	[Voice, changer]	Voice changer	negative
82	[1frrst]	1frrst	negative
83	[I, n't, like, phone, 📱, 😡, Yr, price, dekho, ...	I n't like phone 📱 😡 Yr price dekho 👎👎👎👎	negative
84	[Dabba, phn, hai, 😡, 😡, 😡, 😡]	Dabba phn hai 😡 😡 😡 😡	negative
85	[Checkout, full, review, Redmi, Note, 11, Pro,...	Checkout full review Redmi Note 11 Pro 5G smar...	positive

22. Change predict column name:

```
In [64]: data.columns.values[2] = "sentiment"

In [65]: data
```

23. Delete author comment:

```
In [63]: data = data[data.Name != 'Jason Brownlee']

In [64]: data
```

Out[64]:

	Name	comments	clean_comment	lemmatized	sentiment
0	Gibachan	\nIf the deep learning is such great algorithm...	[If, deep, learning, great, algorithm, think, ...	If deep learning great algorithm think older a...	positive
2	Priyankar	\nCould you please give me some idea, how deep...	[Could, please, give, idea, deep, learning, ap...	Could please give idea deep learning applied s...	positive
4	Roman	\nMe too\n\nReply \n	[Me, Reply]	Me Reply	positive
5	sk	\ncan you tell what could be research for MS I...	[tell, could, research, MS, level, deep, learn...	tell could research MS level deep learning Reply	positive
7	Rita	\nif I want to reconstruct 3D object, which is...	[I, want, reconstruct, 3D, object, better, ANN...	I want reconstruct 3D object better ANN CNN de...	positive
...	...	...	...	...	...
282	deepika	\nwow its such a great post! im really into di...	[wow, great, post, im, really, digital, world,...	wow great post im really digital world learnin...	positive
283	James Carmichael	\nThank you for the feedback, Deepika!\n\nRepl...	[Thank, feedback, Deepika, Reply]	Thank feedback Deepika Reply	positive
284	Camila	\nCan tell us about the deep learning intervie...	[Can, tell, u, deep, learning, interview, ques...	Can tell u deep learning interview question Reply	positive
285	James Carmichael	\nHi Camila…the following looks like a great s...	[Hi, Camila…the, following, look, like, great,...	Hi Camila…the following look like great start ...	positive
286	Fatima	\nHi Dr. Jason, I have a dataset.\nit is a mul...	[Hi, Dr, Jason, I, dataset, multiclass, label,...	Hi Dr Jason I dataset multiclass label classif...	negative

24. Randomly select comment:

```
In [65]: data1 = data.sample(n=3)

In [66]: data1
```

Out[66]:

	Name	comments	clean_comment	lemmatized	sentiment
41	Sam Wilson	\nHi, thanks for the good overview. \nIn your ...	[Hi, thanks, good, overview, In, opinion, fiel...	Hi thanks good overview In opinion field CNN c...	positive
261	Saeid Rezaei	\nHello Jason, Thank you for your amazing blog...	[Hello, Jason, Thank, amazing, blog, I, chosen...	Hello Jason Thank amazing blog I chosen Deep L...	positive
278	Kofi Antwi	\nHow do we cite your 2015 Extract Conference?...	[How, cite, 2015, Extract, Conference, How, ci...	How cite 2015 Extract Conference How cite usef...	positive