Vet Schools Contact Information Extraction

Abstract:

This a project, based on Web Scrapping. A list of 5455 schools throughout Europe was given. However, in that list, the contact information was missing. The task was to extract all the ‘Emails’ and ‘Social Media Accounts’ from the website list. The codes first read websites and extract all the text information from a website's link. Then, it matches the pattern of the ‘Emails’ and ‘Social Media Accounts.’ If the pattern matches, it is added to the new column and automatically goes to the next website's link. Reading a list of 5455 links one by one is difficult for any machine; unfortunately, my machine does not have that much memory. So, I distributed the code among the team member and took their help. Each of the members took 1200 websites and divided them into 12 data frames. After receiving the extracted information, I merged all data frames and created the final list of ‘Emails’ and ‘Social Media Accounts.’ Then, the data set was cleaned and wrangled according to the needs of the client.

In [ ]:
```
# pip install requests
# pip install beautifulsoup4
```

In [ ]:
```
# Read and Prepare the data

import pandas as pd
df = pd.read_excel('WS.xlsx')
df = df[['Webpage']]

# Add 'http://' to links that don't have it

df['Webpage'] = df['Webpage'].astype(str).apply(lambda x: 'http://' + x if not x.startswith('http') else
df
```

Shadi = 1201-2400, Rosemary = 2401-3600, Thomas = Rest

In [ ]:
```
# Spliting the data set into 12 data frame. 4 people should cover 5500 observation.
# Each people 1200 observation. then again the data will be marged

df1 = df[0:100]
df2 = df[101:200]
df3 = df[201:300]
df4 = df[301:400]
df5 = df[401:500]
df6 = df[501:600]
df7 = df[601:700]
df8 = df[701:800]
df9 = df[801:900]
df10 = df[901:1000]
df11 = df[1001:1100]
df12 = df[1101:1200]
```

Emails

In [ ]:
```
# Function to extract emails
import requests
from bs4 import BeautifulSoup
import re
import pandas as pd

def scrape_and_extract_emails(url_list):
    data = {'Website': [], 'Emails': []}

    for url in url_list:
        try:
            # Send a GET request to the specified URL
            response = requests.get(url)
```

```
            # Check if the request was successful (status code 200)
            if response.status_code == 200:
                # Parse the HTML content of the page
                soup = BeautifulSoup(response.text, 'html.parser')

                # Extract all the text from the page
                all_text = soup.get_text()

                # Define the regular expression pattern for extracting email addresses
                email_pattern = r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b'

                # Use re.findall to find all email addresses in the text
                emails = re.findall(email_pattern, all_text)

                # Append the data to the dictionary
                data['Website'].append(url)
                data['Emails'].append(emails)

            else:
                print(f"Error: Unable to fetch content from {url}. Status Code: {response.status_code}")

        except Exception as e:
            print(f"An error occurred for {url}: {e}")

    # Create a DataFrame from the collected data
    df = pd.DataFrame(data)

    return df
```

Extracting the emails and saving the file to the directory

In [ ]:
```
email_list1 = scrape_and_extract_emails(df1['Webpage'])
email_list1.to_excel('email_list1.xlsx', index=False)
```

In [ ]:
```
email_list2 = scrape_and_extract_emails(df2['Webpage'])
email_list2.to_excel('email_list2.xlsx', index=False)
```

In [ ]:
```
email_list3 = scrape_and_extract_emails(df3['Webpage'])
email_list3.to_excel('email_list3.xlsx', index=False)
```

In [ ]:
```
email_list4 = scrape_and_extract_emails(df4['Webpage'])
email_list4.to_excel('email_list4.xlsx', index=False)
```

In [ ]:
```
email_list5 = scrape_and_extract_emails(df5['Webpage'])
email_list5.to_excel('email_list5.xlsx', index=False)
```

In [ ]:
```
email_list6 = scrape_and_extract_emails(df6['Webpage'])
email_list6.to_excel('email_list6.xlsx', index=False)
```

In [ ]:
```
email_list7 = scrape_and_extract_emails(df7['Webpage'])
email_list7.to_excel('email_list7.xlsx', index=False)
```

In [ ]:
```
email_list8 = scrape_and_extract_emails(df8['Webpage'])
email_list8.to_excel('email_list8.xlsx', index=False)
```

In [ ]:
```
email_list9 = scrape_and_extract_emails(df9['Webpage'])
email_list9.to_excel('email_list9.xlsx', index=False)
```

In [ ]:
```
email_list10 = scrape_and_extract_emails(df10['Webpage'])
email_list10.to_excel('email_list10.xlsx', index=False)
```

In [ ]:
```
email_list11 = scrape_and_extract_emails(df11['Webpage'])
email_list11.to_excel('email_list11.xlsx', index=False)
```

In [ ]:
```
email_list12 = scrape_and_extract_emails(df12['Webpage'])
email_list12.to_excel('email_list12.xlsx', index=False)
```

In [ ]:
```
# Loading and Marging the Email list
email_list1 = pd.read_excel('email_list1.xlsx')
email_list2 = pd.read_excel('email_list1.xlsx')
email_list3 = pd.read_excel('email_list1.xlsx')
email_list4 = pd.read_excel('email_list1.xlsx')
email_list5 = pd.read_excel('email_list1.xlsx')
email_list6 = pd.read_excel('email_list1.xlsx')
email_list7 = pd.read_excel('email_list1.xlsx')
email_list8 = pd.read_excel('email_list1.xlsx')
email_list9 = pd.read_excel('email_list1.xlsx')
email_list10 = pd.read_excel('email_list1.xlsx')
email_list11 = pd.read_excel('email_list1.xlsx')
email_list12 = pd.read_excel('email_list1.xlsx')

# Marge the lists
email_list = [email_list1,
              email_list2,
              email_list3,
              email_list4,
              email_list5,
              email_list6,
              email_list7,
              email_list8,
              email_list9,
              email_list10,
              email_list11,
              email_list12]

# Concatenate the data frames column-wise
email_list_final = pd.concat(email_list, axis=0)
email_list_final.to_excel('email_list_tipu.xlsx', index=False)
```

Marge The Files

In [1]:
```
import pandas as pd

#Uploading the lists
df0 = pd.read_excel('WS.xlsx')
df1 = pd.read_excel('email_list_tipu.xlsx')
df2 = pd.read_excel('email_list_shadi.xlsx')
df3 = pd.read_excel('email_list_rosemary.xlsx')
df4 = pd.read_excel('email_list_tomas.xlsx')


# Marge the lists
email_list = [df1,df2,df3,df4]

# Concatenate the data frames column-wise
email_list_final = pd.concat(email_list, axis=0)
email_list_final

# Putting the same name of the dataframes
df0.rename(columns = {'Webpage':'Website'}, inplace = True)
# Add 'http://' to links that don't have it
df0['Website'] = df0['Website'].astype(str).apply(lambda x: 'http://' + x if not x.startswith('http') el

# Merge DataFrames on the 'website' column
df = pd.merge(df0, email_list_final, on='Website', how='left')

# Fill missing values in the 'email' column with an empty string or any other desired value
df['Emails'] = df['Emails'].fillna('')
df.to_excel('WebScrappingProjectFinalFileEmail.xlsx', index=False)
df
```

Out[1]:

	Proposal	Erasmus		Organisation	Postal

	Number	Code	PIC	OID	Legal Name	Street	Code	City	Country	Websit
0	101099009	E ALMERIA06	949571995	NaN	IES ABDERA	C/ MARISMA, 6	4770	ADRA	Spain	http://www.iesabdera.cor
1	101099009	E ALMERIA06	949571995	NaN	IES ABDERA	C/ MARISMA, 6	4770	ADRA	Spain	http://www.iesabdera.cor
2	101099009	E ALMERIA06	949571995	NaN	IES ABDERA	C/ MARISMA, 6	4770	ADRA	Spain	http://www.iesabdera.cor
3	101099009	E ALMERIA06	949571995	NaN	IES ABDERA	C/ MARISMA, 6	4770	ADRA	Spain	http://www.iesabdera.cor
4	101099009	E ALMERIA06	949571995	NaN	IES ABDERA	C/ MARISMA, 6	4770	ADRA	Spain	http://www.iesabdera.cor
...	...	...	...	...	...	...	...	...	...	.
6343	101012284	N BERGEN14	916768923	E10002190	HOGSKULEN PA VESTLANDET	INNDALSVEIEN 28	5020	BERGEN	Norway	http://www.hvl.no
6344	101011552	E PALMA34	944834418	E10002067	Colegio Nuestra Señora de Montesión	C/Montesión, 24	07001	Palma	Spain	http://www.colegiomontesion.e
6345	101010243	F DAX08	923046181	E10001844	Lycée Charles Despiau	637 avenue du houga	40000	MONT DE MARSAN	France	http://na
6346	101048606	E SEVILLA84	916717610	E10001368	IES EL CARMEN	CALLE SANTA CLARA S/N	41370	CAZALLA DE LA SIERRA	Spain	http://www.ieselcarmen cazalla.e
6347	101010166	E	910511162	E10001284	IES MARIA	Calle: San Vicente Ferrer,	50011	ZARAGOZA	Spain	http://www.ies-mariamoliner.e

	ZARAGOZ56	MOLINER	S/Nº

6348 rows × 13 columns

Social Media

In [ ]:
```
import requests
from bs4 import BeautifulSoup
import re
import pandas as pd

def extract_social_media_links(url):
    try:
        # Fetch the HTML content of the webpage
        response = requests.get(url)
        response.raise_for_status()

        # Parse the HTML content using BeautifulSoup
        soup = BeautifulSoup(response.text, 'html.parser')

        # Define a regular expression to match social media links
        social_media_regex = re.compile(r'https?://(www\.)?(facebook|twitter|instagram|linkedin)\.com/[^

        # Find all links in the HTML content
        all_links = soup.find_all('a', href=True)

        # Extract social media links based on the regular expression
        social_media_links = [link['href'] for link in all_links if social_media_regex.match(link['href'

        return social_media_links

    except requests.RequestException as e:
```

```
        print(f"Error fetching the webpage {url}: {e}")
        return None

# Function to extract social media links from a list of websites
def extract_social_media_links_from_list(websites):
    data = []

    for website in websites:
        social_media_accounts = extract_social_media_links(website)
        data.append({'Website': website, 'Social Media Accounts': social_media_accounts})

    return data
```

In [ ]:
```
social_media_list1 = extract_social_media_links_from_list(df1['Webpage'])
social_media_list1 = pd.DataFrame(social_media_list1)
social_media_list1.to_excel('social_media_list1.xlsx', index=False)
```

In [ ]:
```
social_media_list2 = extract_social_media_links_from_list(df2['Webpage'])
social_media_list2 = pd.DataFrame(social_media_list2)
social_media_list2.to_excel('social_media_list2.xlsx', index=False)
```

In [ ]:
```
social_media_list3 = extract_social_media_links_from_list(df3['Webpage'])
social_media_list3 = pd.DataFrame(social_media_list3)
social_media_list3.to_excel('social_media_list3.xlsx', index=False)
```

In [ ]:
```
social_media_list4 = extract_social_media_links_from_list(df4['Webpage'])
social_media_list4 = pd.DataFrame(social_media_list4)
social_media_list4.to_excel('social_media_list4.xlsx', index=False)
```

In [ ]:
```
social_media_list5 = extract_social_media_links_from_list(df5['Webpage'])
social_media_list5 = pd.DataFrame(social_media_list5)
social_media_list5.to_excel('social_media_list5.xlsx', index=False)
```

In [ ]:
```
social_media_list6 = extract_social_media_links_from_list(df6['Webpage'])
social_media_list6 = pd.DataFrame(social_media_list6)
social_media_list6.to_excel('social_media_list6.xlsx', index=False)
```

In [ ]:
```
social_media_list7 = extract_social_media_links_from_list(df7['Webpage'])
social_media_list7 = pd.DataFrame(social_media_list7)
social_media_list7.to_excel('social_media_list7.xlsx', index=False)
```

In [ ]:
```
social_media_list8 = extract_social_media_links_from_list(df8['Webpage'])
social_media_list8 = pd.DataFrame(social_media_list8)
social_media_list8.to_excel('social_media_list8.xlsx', index=False)
```

In [ ]:
```
social_media_list9 = extract_social_media_links_from_list(df9['Webpage'])
social_media_list9 = pd.DataFrame(social_media_list9)
social_media_list9.to_excel('social_media_list9.xlsx', index=False)
```

In [ ]:
```
social_media_list10 = extract_social_media_links_from_list(df10['Webpage'])
social_media_list10 = pd.DataFrame(social_media_list10)
social_media_list10.to_excel('social_media_list10.xlsx', index=False)
```

In [ ]:
```
social_media_list11 = extract_social_media_links_from_list(df11['Webpage'])
social_media_list11 = pd.DataFrame(social_media_list11)
social_media_list11.to_excel('social_media_list11.xlsx', index=False)
```

In [ ]:
```
social_media_list12 = extract_social_media_links_from_list(df12['Webpage'])
social_media_list12 = pd.DataFrame(social_media_list12)
social_media_list12.to_excel('social_media_list12.xlsx', index=False)
```

In [ ]:
```
# Loading and Marging the Email list
social_media_list1 = pd.read_excel('social_media_list1.xlsx')
social_media_list2 = pd.read_excel('social_media_list2.xlsx')
social_media_list3 = pd.read_excel('social_media_list3.xlsx')
social_media_list4 = pd.read_excel('social_media_list4.xlsx')
social_media_list5 = pd.read_excel('social_media_list5.xlsx')
social_media_list6 = pd.read_excel('social_media_list6.xlsx')
social_media_list7 = pd.read_excel('social_media_list7.xlsx')
social_media_list8 = pd.read_excel('social_media_list8.xlsx')
social_media_list9 = pd.read_excel('social_media_list9.xlsx')
social_media_list10 = pd.read_excel('social_media_list10.xlsx')
social_media_list11 = pd.read_excel('social_media_list11.xlsx')
social_media_list12 = pd.read_excel('social_media_list12.xlsx')


# Marge the lists
social_media_list = [social_media_list1,
             social_media_list2,
             social_media_list3,
             social_media_list4,
             social_media_list5,
             social_media_list6,
             social_media_list7,
             social_media_list8,
             social_media_list9,
             social_media_list10,
             social_media_list11,
             social_media_list12]

# Concatenate the data frames column-wise
social_media_list_final = pd.concat(social_media_list, axis=0)
social_media_list_final.to_excel('social_media_list_tipu.xlsx', index=False)
```

Marge The Files

In [3]:
```
import pandas as pd

#Uploading the lists
df0 = pd.read_excel('WebScrappingProjectFinalFileEmail.xlsx')
df1 = pd.read_excel('social_media_list_tipu.xlsx')
df2 = pd.read_excel('social_media_list_shadi.xlsx')
df3 = pd.read_excel('social_media_list_rosemary.xlsx')
df4 = pd.read_excel('social_media_list_tomas.xlsx')


# Marge the lists
social_media_list = [df1,df2,df3,df4]

# Concatenate the data frames column-wise
social_media_list_final = pd.concat(social_media_list, axis=0)



df = pd.merge(df0, social_media_list_final, on='Website', how='left')
df = df[df['Website'] != 'http://nan']

df.to_excel('WebScrappingProjectFinalFile(Email+SocialMedia).xlsx', index=False)
df
```

Out[3]:

	Proposal Number	Erasmus Code	PIC	OID	Organisation Legal Name	Street	Postal Code	City	Country	Webs
0	101099009	E ALMERIA06	949571995	NaN	IES ABDERA	C/ MARISMA, 6	4770	ADRA	Spain	http://www.iesabdera.c

1	101099009	E ALMERIA06	949571995	NaN	IES ABDERA	C/ MARISMA, 6	4770	ADRA	Spain	http://www.iesabdera.c
2	101099009	E ALMERIA06	949571995	NaN	IES ABDERA	C/ MARISMA, 6	4770	ADRA	Spain	http://www.iesabdera.c
3	101099009	E ALMERIA06	949571995	NaN	IES ABDERA	C/ MARISMA, 6	4770	ADRA	Spain	http://www.iesabdera.c
4	101099009	E ALMERIA06	949571995	NaN	IES ABDERA	C/ MARISMA, 6	4770	ADRA	Spain	http://www.iesabdera.c
...	...	...	...	...	...	...	...	...	...	
32214	101014538	F PAPEETE03	944838104	E10002357	Lycée hôtelier de Tahiti	PK 7,8 côté Montagne	98717	PUNAAUIA	France	http://www.des
32215	101012284	N BERGEN14	916768923	E10002190	HOGSKULEN PA VESTLANDET	INNDALSVEIEN 28	5020	BERGEN	Norway	http://www.hvl.r
32216	101011552	E PALMA34	944834418	E10002067	Colegio Nuestra Señora de Montesión	C/Montesión, 24	07001	Palma	Spain	http://www.colegiomontesion
32358	101048606	E SEVILLA84	916717610	E10001368	IES EL CARMEN	CALLE SANTA CLARA S/N	41370	CAZALLA DE LA SIERRA	Spain	http://www.ieselcarme cazalla
32359	101010166	E ZARAGOZ56	910511162	E10001284	IES MARIA MOLINER	Calle: San Vicente Ferrer, S/Nº	50011	ZARAGOZA	Spain	http://www.ies-mariamoliner

6557 rows × 14 columns

Adjust The Columns Of The File

In [10]:

```
df = pd.read_excel('WebScrappingProjectFinalFile(Email+SocialMedia).xlsx')
#df = df[['Social Media Accounts','Website' ]]


# Assuming df is your DataFrame
df['Social Media Accounts'] = df['Social Media Accounts'].astype(str).str.replace('[', '').str.replace('
df['Social Media Accounts'] = df['Social Media Accounts'].astype(str).str.replace('[', '').str.replace('
df['Emails'] = df['Emails'].astype(str).str.replace('[', '').str.replace(']', '')
df['Emails'] = df['Emails'].astype(str).str.replace('[', '').str.replace(']', '').str.replace("'", '')

# Define the platforms
platforms = ['facebook', 'instagram', 'linkedin', 'teligram', 'twitter']

# Create columns for each platform and initialize with NaN
for platform in platforms:
    df[platform] = None

# Split the 'Social Media Accounts' column and populate the respective platform columns
for index, row in df.iterrows():
    social_media_links = row['Social Media Accounts'].split(',')
    for link in social_media_links:
        for platform in platforms:
            if platform in link:
                df.at[index, platform] = link

# Drop the original 'Social Media Accounts' column
df = df.drop('Social Media Accounts', axis=1)
df = df.drop_duplicates(subset='Website')
df.to_excel('VetSchool.xlsx', index = False)
```

```
/var/folders/6n/77y4z1n500d_4p85zkpbc_yw0000gn/T/ipykernel_4774/2037213289.py:6: FutureWarning: The def
```

```
/var/folders/6n/77y4z1n500d_4p85zkpbc_yw0000gn/T/ipykernel_4774/2037213289.py:6: FutureWarning: The def
ault value of regex will change from True to False in a future version. In addition, single character r
egular expressions will *not* be treated as literal strings when regex=True.
  df['Social Media Accounts'] = df['Social Media Accounts'].astype(str).str.replace('[', '').str.replac
e(']', '')
/var/folders/6n/77y4z1n500d_4p85zkpbc_yw0000gn/T/ipykernel_4774/2037213289.py:7: FutureWarning: The def
ault value of regex will change from True to False in a future version. In addition, single character r
egular expressions will *not* be treated as literal strings when regex=True.
  df['Social Media Accounts'] = df['Social Media Accounts'].astype(str).str.replace('[', '').str.replac
e(']', '').str.replace("'", '')
/var/folders/6n/77y4z1n500d_4p85zkpbc_yw0000gn/T/ipykernel_4774/2037213289.py:8: FutureWarning: The def
ault value of regex will change from True to False in a future version. In addition, single character r
egular expressions will *not* be treated as literal strings when regex=True.
  df['Emails'] = df['Emails'].astype(str).str.replace('[', '').str.replace(']', '')
/var/folders/6n/77y4z1n500d_4p85zkpbc_yw0000gn/T/ipykernel_4774/2037213289.py:9: FutureWarning: The def
ault value of regex will change from True to False in a future version. In addition, single character r
egular expressions will *not* be treated as literal strings when regex=True.
  df['Emails'] = df['Emails'].astype(str).str.replace('[', '').str.replace(']', '').str.replace("'",
'')
```

In [11]:

Out[11]:

	Website	ECHE Start Date	ECHE End Date	Emails	facebook	
	http://www.iesabdera.com	2022-11-01	2029-12-31		None	
	undimediazionelinguistica.com/	2022-11-01	2029-12-31		None	
	ttp://www.lycee-saint-eloi.com	2022-11-01	2029-12-31	accueil@lycee-saint-eloi.com	https://www.facebook.com/sainteloiaix/	https://www.instagram

https://2a.gretacfa.corsica/	2022-11-01	2029-12-31		http://www.facebook.com/gretacfa.corsica	http://www.instagram.cor
http://www.iesja.es	2022-11-01	2029-12-31	secretaria.iesja@gmail.com, ies.josefinaaldeco...	None	
...	...	...	...	...	
http://www.des.pf	2021-01-01	2029-12-31	nan	None	
http://www.hvl.no/	2021-01-01	2029-12-31		https://www.facebook.com/hvl.no	https://www.ins
ttp://www.colegiomontesion.es	2021-01-01	2029-12-31	secretariasonmoix@colegiomontesion.es, secreta...	None	https://instagram.co
ɔ://www.ieselcarmen-cazalla.es	2021-11-01	2029-12-31	nan	None	
http://www.ies-mariamoliner.es	2021-01-01	2029-12-31	iesmmozaragoza@educa.aragon.es	None	

In [ ]: