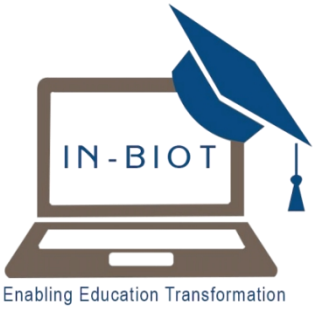
FINAL PROJECT REPORT

On

Scraping details of Engineering colleges in India

Submitted by

Name	**:**	**Kushal Sharan**
College	**:**	**ITER-Siksha 'O' Anusandhan**
Passout Batch	**:**	**2022**
Organisation	**:**	**IN-BIOT Pvt. Ltd.**
Course Title	**:**	**Machine Learning Using Python**
Intern ID	**:**	**p112@industryskills.org**
Email ID	**:**	**sharankushal0@gmail.com**



IN-BIOT Private Limited

Bangalore, Karnataka

Acknowledgement

I would like to express my sincere thanks and gratitude to

IN-BIOT Pvt. Ltd. for providing this wonderful opportunity to learn and work on much sought after skill in software industry-

Web Scraping.

I would specially like to thank **Anjali ma'am** for mentoring me throughout my journey with this project as well as during internship. I also want to express my gratitude to **Anuradha ma'am** and **Alok Sir** for making this whole journey a smooth flow.

Kushal Sharan

Table of Contents

S.No.	Title	Page Number
1.	Aim of Project	4
2.	Scope of Project	4
3.	Technologies Used	4
4.	What is Web Scraping?	4-5
6.	Why Python for Web Scraping?	5-6
7.	My Approach	6-14
8.	Visualization using Matplotlib	14-16
9.	Conclusion	16
10.	References	17

Aim of Project:

- In the times of pandemic, IN-BIOT Pvt ltd. aims at enabling remote learning in large scale. It provides technical setup, support and training to it's clients.
- The objective of this project is to scrape available details of Engineering colleges in India from the web and present a report to marketing team, which would be then used for running a marketing campaign.

Scope of project:

- The project is designed to extract website and emails of AICTE approved colleges in India, and scrape their contact details.

Technologies used:

- Python 3.7.10 64-bit
- conda 4.10.3
- Visual Studio Code 1.59.1(user setup)
- WPS Office 11.2.0.10233

What is Web Scraping?

Web scraping is the process of collecting structured web data in an automated fashion. It’s also called web data extraction. Some of the main use cases of web scraping include price monitoring, price intelligence, news monitoring, lead generation, and market research among many others.

In general, web data extraction is used by people and businesses who want to make use of the vast amount of publicly available web data to make smarter decisions.

If we ever copy and pasted information from a website, we've performed the same function as any web scraper, only on a microscopic, manual scale. Unlike the mundane, mind-numbing process of manually extracting data, web scraping uses intelligent automation to retrieve hundreds, millions, or even billions of data points from the internet's seemingly endless frontier.

Why Python for Web Scraping?

- Ease of Use: Python is simple to code. We do not have to add semi-colons ";" or curly-braces "{}" anywhere. This makes it less messy and easy to use.
- Large Collection of Libraries: Python has a huge collection of libraries such as Numpy, Matlplotlib, Pandas etc., which provides methods and services for various purposes. Hence, it is suitable for web scraping and for further manipulation of extracted data.
- Dynamically typed: In Python, we don't have to define datatypes for variables, you can directly use the variables wherever required. This saves time and makes your job faster.
- Easily Understandable Syntax: Python syntax is easily understandable mainly because reading a Python code is very similar to reading a statement in English. It is expressive and easily readable, and the indentation used in Python also helps the user to differentiate between different scope/blocks in the code.
- Small code, large task: Web scraping is used to save time. But what's the use if we spend more time writing the code? Well, we don't have to. In Python, you can write small codes to do large tasks. Hence, we save time even while writing the code.
- Community: What if we get stuck while writing the code? we don't have to worry. Python community has one of the biggest and most active communities, where we can seek help from.

My Approach:

1. For extracting details of college we must already have one base list which we can use for further extraction of details. For the same purpose, I collected the one base list of colleges which is AICTE approved. That contains AICTE ID, Name, Address, District, State, Institute information of more than 5700 colleges in India. This was imported using Pandas read_csv method as a dataframe. A Data frame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns.

AICTE ID	NAME	ADDRESS	DISTRICT	INSTITUTE TYPE	State
1-7003786958	GOVERNMENT POLYTECHNIC MAYEM BICHOLIM	MAYEM BICHOLIM GOA	NORTH GOA	Government	Goa
1-7006364088	DON BOSCO COLLEGE OF ENGINEERING	DON BOSCO, FATORDA,	SOUTH GOA	Private-Self Fin	Goa
1-7006458223	INSTITUTE OF SHIPBUILDING TECHNOLOGY	BOGDA, VASCO-DA-GAMA, GOA 403802	SOUTH GOA	Govt aided	Goa
1-7006524807	GOA COLLEGE OF ENGINEERING (GOVERNMENT OF GOA)	FARMAGUDI, PONDA-GOA PIN-403401	NORTH GOA	Government	Goa
1-7010374121	SHREE RAYESHWAR INSTITUTE OF ENGINEERING AND INFO	SHIV SHAIL, KARAI, SHIRODA, GOA 403103	NORTH GOA	Private-Self Fin	Goa
1-7011315312	HQ 2 SIGNAL TRAINING CENTRE	C/O 56 APO	NORTH GOA	Government	Goa

```
import numpy as np
import pandas as pd
pd.options.mode.chained_assignment = None  # default='warn'
```

[1] ✓ 1.6s

```
df=pd.read_csv('C:/Users/MyPC/OneDrive/Documents/Inbiot VS/AICTE Combined List 1.csv')
df.head()
```

[2] ✓ 0.2s

	AICTE ID	NAME	ADDRESS	DISTRICT	INSTITUTE TYPE	State
0	1-7003786958	GOVERNMENT POLYTECHNIC MAYEM BICHOLIM	MAYEM BICHOLIM GOA	NORTH GOA	Government	Goa
1	1-7006364088	DON BOSCO COLLEGE OF ENGINEERING	DON BOSCO, FATORDA,	SOUTH GOA	Private-Self Financing	Goa
2	1-7006458223	INSTITUTE OF SHIPBUILDING TECHNOLOGY	BOGDA, VASCO-DA-GAMA, GOA 403802	SOUTH GOA	Govt aided	Goa
3	1-7006524807	GOA COLLEGE OF ENGINEERING (GOVERNMENT OF GOA)	FARMAGUDI, PONDA-GOA PIN-403401	NORTH GOA	Government	Goa
4	1-7010374121	SHREE RAYESHWAR INSTITUTE OF ENGINEERING AND I...	SHIV SHAIL, KARAI, SHIRODA, GOA 403103	NORTH GOA	Private-Self Financing	Goa

2. Next step is to extract websites of corresponding colleges. A general apprach would be to take the names of college individually, perform a google search, copy the website and paste it in website column. Why not automate this process? For this I used googlesearch library in python. googlesearch uses requests and BeautifulSoup4 to scrape Google.
3. The idea is to iterate through each row of our base list(dataframe), perform a googlesearch, extract the topmost website of search result and store it in website column. Though this idea has one drawback- We can instruct googlesearch to extract topmost result, hence this result may contain thir party websites like collegedunia.com, siksha.com, etc. which are not the actual websites of that particular college. Thus filtering of data is requried

after extraction proccess. This extraction was time taking (4-5 hours), as it had to iterate 5700+ rows. The resulted dataframe was then exported to csv file.

```
df["WEBSITE"]=np.nan
[3] ✓ 0.5s
                                                                    Python

for i in range(0,df.shape[0]):
    query=df.iloc[i,1]
    for j in search(query, num=1, stop=1, pause=1):
        df.iloc[i,6]=j
        print(df.iloc[i,1],'\t',j)
[6] ⊗ 33.8s
                                                                    Python
...  GOVERNMENT POLYTECHNIC MAYEM BICHOLIM    https://gpb.goa.gov.in/
     DON BOSCO COLLEGE OF ENGINEERING         https://www.dbcegoa.ac.in/
     INSTITUTE OF SHIPBUILDING TECHNOLOGY     http://www.isbt.ac.in/
     GOA COLLEGE OF ENGINEERING (GOVERNMENT OF GOA)   http://www.gec.ac.in/
     SHREE RAYESHWAR INSTITUTE OF ENGINEERING AND INFORMATION TECHNOLOY       https://ritgoa.ac.in/
     HQ 2 SIGNAL TRAINING CENTRE      https://www.indgovtjobs.in/2021/07/2-Signal-Training-Centre-Goa-Recruitment.html
     PADRE CONCEICAO COLLEGE OF ENGINEERING    https://pccegoa.edu.in/
```

4. Following step 3, I was able to scrape all websites (mix of actual and third party websites). Next step is to extract email information. For emails, apporach is the websites retirieved from previous step, we need to visit that website using urllib.request library and search for email pattern using regular expression. If the match is found, email is stored in email column. This whole process needs to be in try except block, to handle errors and making sure program doesn't termintate abruptly.

```
import pandas as pd
import re
from matplotlib import pyplot as plt
import urllib.request
import numpy as np
df=pd.read_csv('C:/Users/MyPC/OneDrive/Documents/Inbiot VS/FINAL/Final after website/AICTE Combined List FINAL.csv'
                                                                    Python

df.head()
                                                                    Python
```

	Unnamed: 0	AICTE ID	NAME	ADDRESS	DISTRICT	INSTITUTE	STATE	WEBSITE
0	0	1-7003786958	GOVERNMENT POLYTECHNIC MAYEM BICHOLIM	MAYEM BICHOLIM GOA	NORTH GOA	Government	Goa	http://gpb.goa.gov.in/
1	1	1-7006364088	DON BOSCO COLLEGE OF ENGINEERING	DON BOSCO, FATORDA,	SOUTH GOA	Private-Self Financing	Goa	https://www.dbcegoa.ac.in/
2	2	1-7006458223	INSTITUTE OF SHIPBUILDING TECHNOLOGY	BOGDA, VASCO-DA-GAMA, GOA 403802	SOUTH GOA	Govt aided	Goa	http://www.isbt.ac.in/
3	3	1-7006524807	GOA COLLEGE OF ENGINEERING (GOVERNMENT OF GOA)	FARMAGUDI, PONDA-GOA PIN-403401	NORTH GOA	Government	Goa	http://www.gec.ac.in/

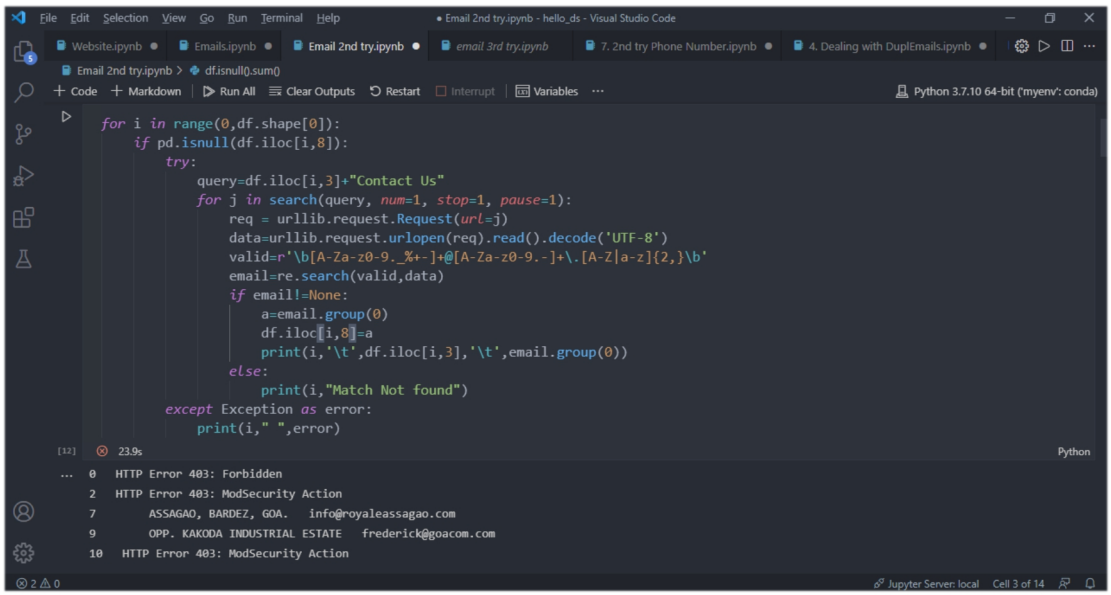
```
df['EMAIL']=np.nan
✓ 0.9s

for i in range(0,df.shape[0]):
    try:
        weburl=urllib.request.urlopen(df.iloc[i,7])
        data=weburl.read().decode('utf-8')
        valid=r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b'
        email=re.search(valid,data)
        if email!=None:
            a=email.group(0)
            df.iloc[i,8]=a
            print(i,df.iloc[i,2],'\t',email.group(0))
        else:
            print(i,'\t', "Match not found")
    except Exception as error:
        print(i,"\t",error)
↻ 26.2s

0       Match not found
1       HTTP Error 406: Not Acceptable
2       HTTP Error 403: ModSecurity Action
3 GOA COLLEGE OF ENGINEERING (GOVERNMENT OF GOA)        enquiry@gec.ac.in
4 SHREE RAYESHWAR INSTITUTE OF ENGINEERING AND INFORMATION TECHNOLOY     principal.ritgoa@gmail.com
```

5. After step 4, for those cells where email value is null, meaning, the code was unsucessful to extract email due to various reason(match not found, HTTP error, decoding error etc.) next step is to take the name of college, perform a google search with:

   name of college + contact us string, visit the first website(usually contact us section of that particular college), and repeat step 4.



6. Practically I was able to reduce number of null values of email after step 5, one last thing of improvisation which striked my mind was, defining headers (which eliminates HTTP error(s)) and using ISO-8859-1 decoder instead of UTF-8 decoder, which futher reduced null values of emails.

Trying with Headers defined(for HTTP error code 403:Forbidden) and ISO-8859-1 decoding instead of UTF-8

```
header= {'User-Agent': 'Mozilla/5.0 (X11; Linux x86_64) '
        'AppleWebKit/537.11 (KHTML, like Gecko) '
        'Chrome/23.0.1271.64 Safari/537.11',
        'Accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8',
        'Accept-Charset': 'ISO-8859-1,utf-8;q=0.7,*;q=0.3',
        'Accept-Encoding': 'none',
        'Accept-Language': 'en-US,en;q=0.8',
        'Connection': 'keep-alive'}
```

```
for i in range(0,df.shape[0]):
    if pd.isnull(df.iloc[i,9]):
        try:
            query=df.iloc[i,3]+"Contact Us"
            for j in search(query, num=1, stop=1, pause=1):
                req = urllib.request.Request(url=j, headers=header)
                data=urllib.request.urlopen(req).read().decode('ISO-8859-1')
                valid=r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b'
                email=re.search(valid,data)
                if email!=None:
                    a=email.group(0)
                    df.iloc[i,9]=a
                    print(i,'\t',df.iloc[i,3],'\t',email.group(0))
                else:
```

```
df1.isnull().sum()

Unnamed: 0       0
AICTE ID         0
NAME             0
ADDRESS          0
DISTRICT         0
INSTITUTE        0
STATE            0
WEBSITE          0
EMAIL         1480
dtype: int64
```

7. Following step 6, the extraction process was now completed. Next steps were required for filtering the extracted data. Starting with websites, first thing was to count the occurences in two ways: 1. Occurence based on netloc value.(in https://www.example.com/ netloc value=www.example.com). This can be achieved using urlparse library which parses the url and would give me an idea of third party websites, since ussually they have more than one occurence. 2. Actual occurence. This is done by value_counts() method in Pandas and would give me an idea of duplicate websites like how many times a website is repeating.

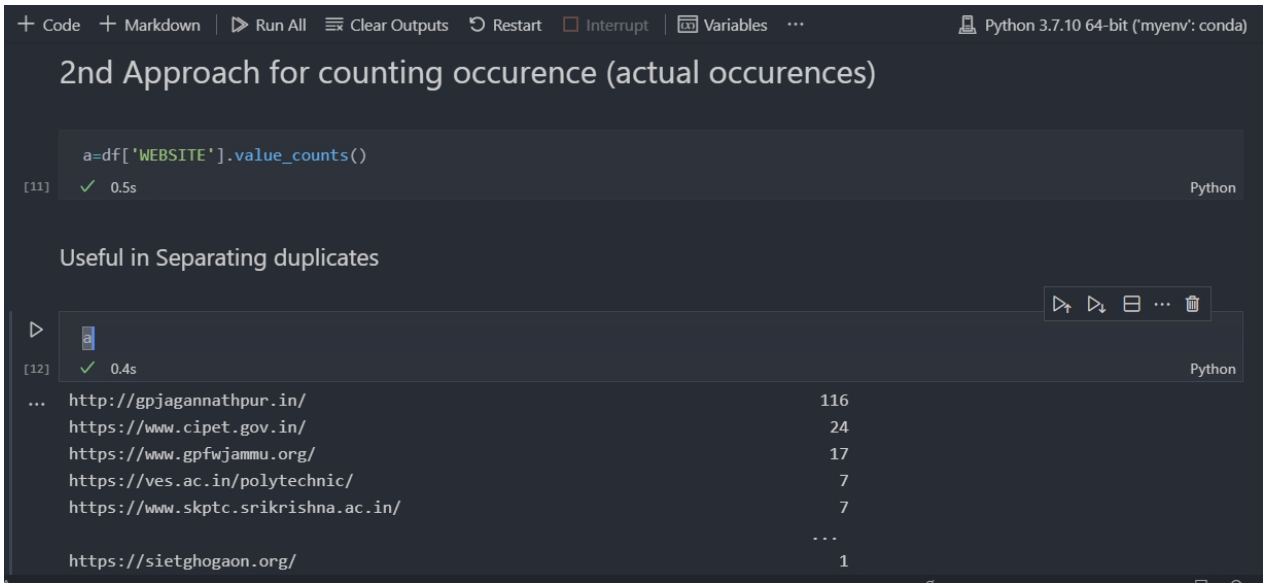
Counting website Occurences based on netloc value (varities of website)

+ Code + Markdown

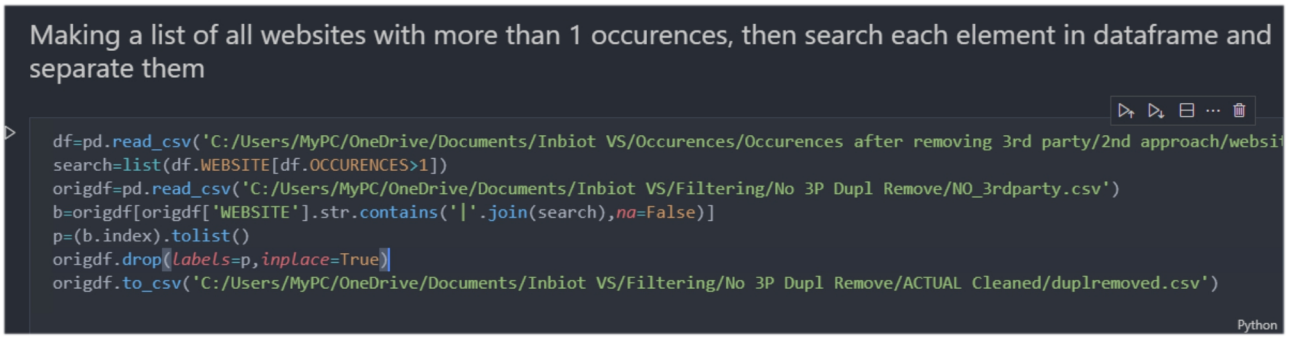
```
from urllib.parse import urlparse
from collections import Counter
ds = []
for i in range(0,df.shape[0]):
    string=df.iloc[i,6]
    o= urlparse(str(string))
    ds.append(o.netloc)
#Counter(ds).most_common()
a=Counter(ds).most_common()
#List-to-Dataframe
tempdf=pd.DataFrame(a)
#renamed the columns
tempdf.columns=['WEBSITE','OCCURENCES']
tempdf
```

	WEBSITE	OCCURENCES
0	www.knowyourcollege-gov.in	134
1	www.careers360.com	127
2	collegedunia.com	125
3	gpjagannathpur.in	116
4	www.collegedekho.com	95
...	...	...
3974	www.grdpoly.com	1
3975	www.bsmcoer.com	1
3976	phonicsedu.com	1
3977	rcppolytechnic.com	1
3978	www.itrroorkee.edu.in	1

3979 rows × 2 columns



8. Once I got the netloc occurence values, I can easily identify third party college, make a list of them, search all the rows which contained a substring indicating the names in third party list and separate those rows. Separating itself is a three way task-- Search, Save searched rows and finally delete them from original dataframe. These all tasks are done using various methods available in Pandas library. Same procedure was followed for dealing with duplicate websites (having occurence more than 1).



9. Once all third party and duplicate websites are separated, next step is to deal with duplicate, null and useless emails. Here useless is referred to cells of email column which are filled with unwanted strings like .png, .jpg (drawback of regular expression).

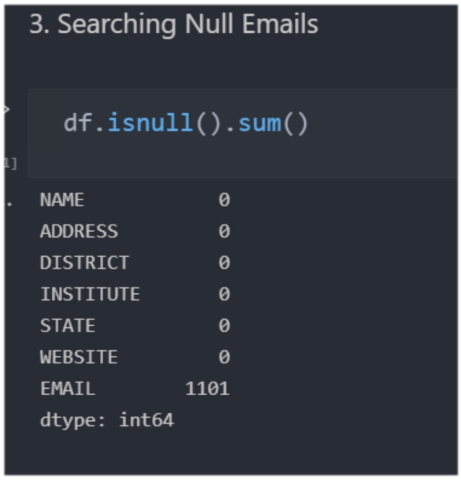
```
a=df.EMAIL.value_counts()
edf=a.to_csv('C:/Users/MyPC/OneDrive/Documents/Inbiot VS/Filtering/No 3P Dupl Remove/ACTUAL C:
edf.columns=['EMAIL','OCCURENCES']
edf.head(3)
```

[76] Python

	EMAIL	OCCURENCES
0	support@youth4work.com	36
1	14370afcdc17429f9e418d5ffbd0334a@sentry.wixpre...	22
2	info@example.com	8

```
search=list(edf.EMAIL[edf.OCCURENCES>1])
search+=['.jpg','.png']
```

[77] Python



10. For dealing with all such emails, I found email occurences using value_counts() method and useless mails using search of dataframe using str.contains method and separated them. For null emails, I used dropna() method to delete them.

5. Deleting Duplicates Emails

```
b=df[df['EMAIL'].str.contains('|'.join(search),na=False)]
b.head(4)
p=(b.index).tolist()
df.drop(labels=p,inplace=True)
```

6. Deleting Null Emails

```
df = df.dropna(axis=0, how='any')
```

```
df.shape[0]
```

11. Once all the data is filtered, next step was to filter the data based on the state information that is zones, Norh, South, East and West. This was again done using Pandas str.contains() method in which basically searching the name of the state and storing resulted rows to corresponding zones was done.

Northern Zone Colleges

```
search=["Himachal pradesh", "Punjab", "uttarakhand","Haryana","uttar pradesh"]
north=df[df['STATE'].str.contains('|'.join(search),na=False)]
north
```

```
#north.to_csv('C:/Users/MyPC/OneDrive/Documents/Inbiot VS/Zonal Distribution/NorthZone.csv')
```

Southern Zone Colleges

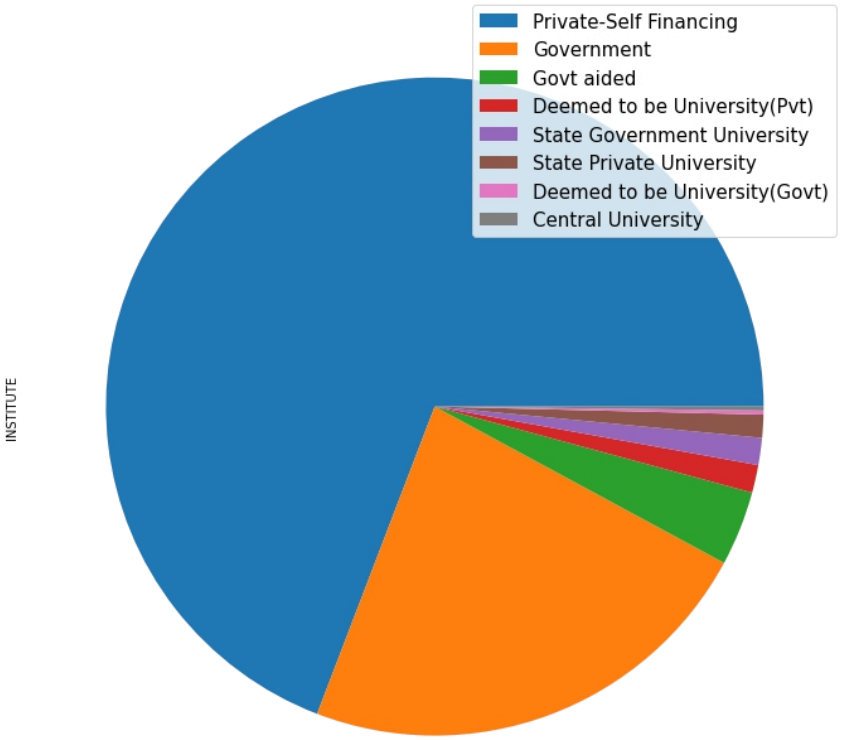
```
search=["Andhra pradesh", "karnataka", "kerala", "Tamil nadu"]
south=df[df['STATE'].str.contains('|'.join(search),na=False)]
south
```

```
#south.to_csv('C:/Users/MyPC/OneDrive/Documents/Inbiot VS/Zonal Distribution/SouthZone.csv')
```

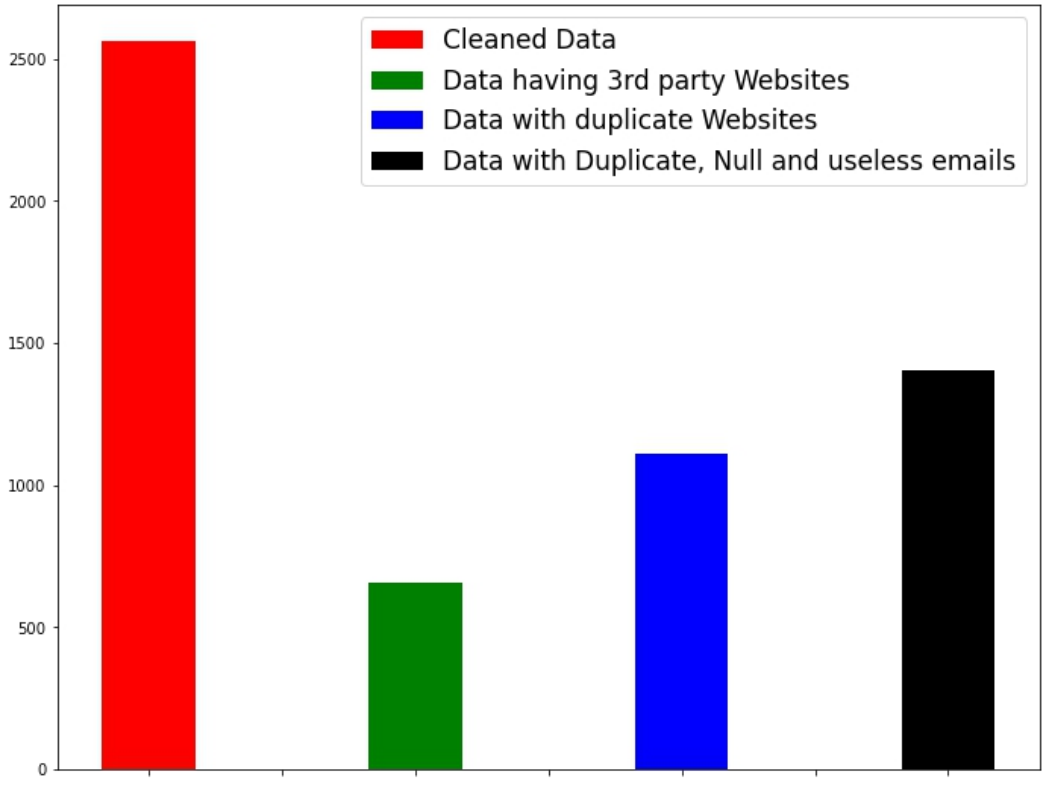
12. The final step is to visualize the extracted and cleaned data by plotting multiple plots using matplotlib.

Visualization using Matplotlib:

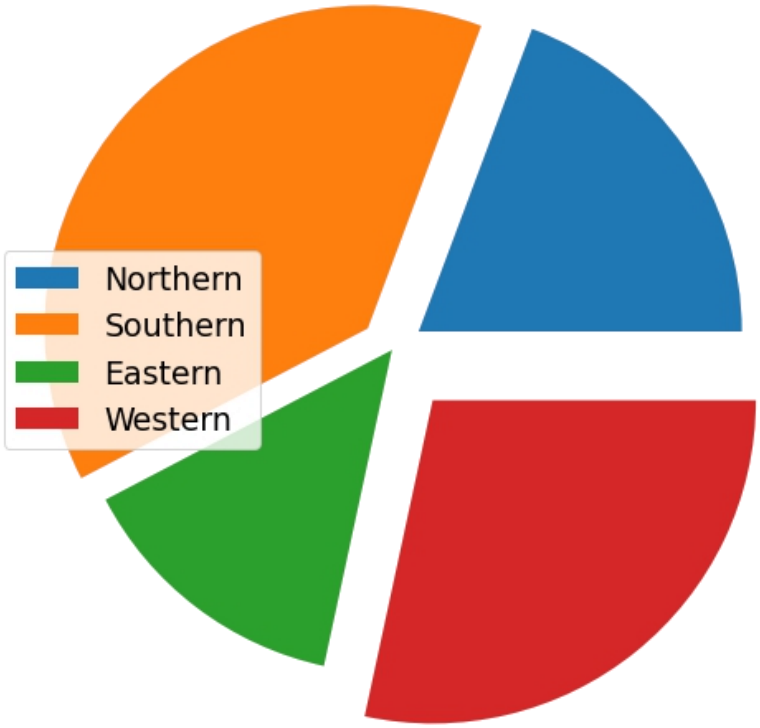
1. Types of Colleges based on Institute:



2. Categories of extracted Data:



3. Zonal Division of Colleges:



4. States belonging to different zones:

Zonal Divison			
Northern	Southern	Eastern	Western
Himachal pradesh	Andhra pradesh	Bihar	Rajasthan
Punjab	Karnataka	Sikkim	Madhya Pradesh
Uttarakhand	Kerala	Arunachal Pradesh	Gujarat
Uttar Pradesh	Tamil Nadu	Nagaland	Maharashtra
		Manipur	Goa
		Tripura	
		Meghalaya	
		Assam	
		WB	
		Jharkhand	
		Odisha	
		Chattishgarh	

Conclusion:

- Through this web scraping project I was able to extract Websites and contact Emails of 2564 colleges (Cleaned Data), which is almost 44.52% of colleges from base list.
- Throughout the project journey got to learn about various amazing and useful Python libraries like googlesearch, urllib.request, urlparse, Pandas, Matplotlib, Collections and many more.
- I also tried for extraction of phone numbers of Northern zone colleges first using regex matching which (didn't work properly) then using phonenumbers library. I was able to get around 21% of phonenumbers and due to deadline and limited knowledge I didn't proceed further with phone number extraction and concluded this as my final project. In future I will try to explore more about phonenumbers library.
- All the code files as well as output files will be uploaded along with this project report.

References:

- https://python-googlesearch.readthedocs.io/en/latest/
- https://docs.python.org/3/library/urllib.request.html
- https://github.com/MicrosoftDocs/ml-basics/blob/master/01%20-%20Data%20Exploration.ipynb
- https://docs.python.org/3/library/urllib.parse.html
- https://docs.python.org/3/library/collections.html
- https://www.geeksforgeeks.org/plot-a-pie-chart-in-python-using-matplotlib/