Data Analysis and Collection

Author:Sai Krishna

Introduction:

The importance of efficient data collection and analysis cannot be overstated. In today's fast-paced environment, businesses and organizations need to have access to accurate and reliable data in order to make informed decisions. That is where tools such as XML parsers, API collectors, web scrapers, and data analysis come into play. This explores these tools by taking a closer look at examples of their usage, specifically SitemapParser for XML parsing, CovidDataCollector for API Data Collection, GuardianJobsScraper for web scraping, and finally, data analysis tool for analyzing collected datasets. By understanding how these tools work together seamlessly towards providing meaningful insights from different types of structured/unstructured datasets that range across various sources, one can appreciate their value in modern-day research projects or business strategies alike.

```
from xml_parser import SitemapParser
from api_data_collector import CovidDataCollector
from web_scraper import GuardianJobsScraper
from data_analysis import DataAnalysis
```

XML Parsing with SitemapParser

```
sitemap_parser = SitemapParser("https://www.bbc.com")
sitemap_df = sitemap_parser.download_sitemaps()

sitemap_df
```

```
                                                   loc  \
0        https://www.bbc.com/marathi/topics/czp1xz28xrqt
1      https://www.bbc.com/serbian/lat/topics/cwr97gj...
2        https://www.bbc.com/persian/topics/c00gmr40p29t
3        https://www.bbc.com/marathi/topics/c00gpx7wmgrt
4        https://www.bbc.com/persian/topics/clxnrqjqdk5t
...                                                  ...
```

```
42756  http://www.bbc.co.uk/zhongwen/trad/china/2014/...
42757  http://www.bbc.co.uk/zhongwen/trad/china/2014/...
42758  http://www.bbc.co.uk/zhongwen/trad/china/2014/...
42759  http://www.bbc.co.uk/zhongwen/trad/multimedia/...
42760  http://www.bbc.co.uk/zhongwen/trad/world/2014/...

                          lastmod
0
1
2
3
4
...                           ...
42756  2014-09-22T13:47:20+00:00
42757  2014-09-22T13:21:06+00:00
42758  2014-09-22T13:02:29+00:00
42759  2014-09-22T12:53:23+00:00
42760  2014-09-22T11:57:57+00:00

[42761 rows x 2 columns]
```

Justification:

The intricate and complex process of XML parsing involves the arduous analysis of an XML document's variegated structure and content. In point of fact, this perplexing process extracts meaningful information from a vast quantity of data. To achieve such a feat with finesse, Python library enthusiasts have developed the highly capable SitemapParser, which impressively enables developers to parse sitemaps using various methods. Upon initialization, one must initialize an instance of SitemapParser with the URL "https://www.bbc.com." By instituting this connection between our script and BBC's website, that can enormously benefit by utilizing their formidable sitemap data. It is indeed worth noting that following a successful connection establishment, our meticulously crafted code endeavors to download the site map for this specific URL into what I refer to as a 'panda's DataFrame' object, affectionately named 'sitemap_df.'

API Data Collection with CovidDataCollector

```
covid_collector =
CovidDataCollector("https://api.covid19india.org/state_district_wise.j
```

```
son")
covid_df = covid_collector.collect_data()

covid_df
```

```
               District
notes  \
0   Railway Quarantine

1   Airport Quarantine

2          Other State

3              Ariyalur

4         Chengalpattu

5                Chennai  [July 22]: 444 backdated deceased entries
adde...
6            Coimbatore

7              Cuddalore

8            Dharmapuri

9              Dindigul

10                Erode

11        Kallakurichi

12        Kancheepuram

13         Kanyakumari

14                Karur

15         Krishnagiri

16              Madurai

17        Nagapattinam

18             Namakkal

19             Nilgiris

20           Perambalur

21         Pudukkottai

22      Ramanathapuram
```

```
23              Ranipet
24                Salem
25            Sivaganga
26              Tenkasi
27            Thanjavur
28                Theni
29          Thiruvallur
30           Thiruvarur
31         Thoothukkudi
32     Tiruchirappalli
33          Tirunelveli
34           Tirupathur
35             Tiruppur
36      Tiruvannamalai
37              Vellore
38           Viluppuram
39         Virudhunagar
40      Mayiladuthurai

   active confirmed migratedother deceased recovered  \
0       0       428             0        0       428
1       8      2098             0        2      2088
2       0         0             0        0         0
3     237     16037             0      240     15560
4    1152    163274             0     2413    159709
5    2048    540300             0     8345    529907
6    2259    231863             0     2201    227403
7     777     61220             0      821     59622
8     316     26415             0      237     25862
9     157     32322             0      627     31538
10   1710     95559             0      641     93208
11    468     29521             0      199     28854
```

12	448	72144	0	1213	70483
13	339	60451	0	1022	59090
14	188	22836	0	351	22297
15	309	41638	0	325	41004
16	228	73729	0	1147	72354
17	427	19075	0	295	18353
18	574	47771	0	458	46739
19	493	31048	0	186	30369
20	109	11579	0	225	11245
21	356	28533	0	372	27805
22	94	20110	0	351	19665
23	216	42169	0	745	41208
24	843	94328	0	1597	91888
25	229	19022	0	199	18594
26	112	26928	0	484	26332
27	995	68972	0	864	67113
28	120	43038	0	514	42404
29	941	114461	0	1766	111754
30	432	38327	0	376	37519
31	196	55274	0	398	54680
32	749	73209	0	979	71481
33	252	48151	0	430	47469
34	178	28410	0	604	27628
35	857	88754	0	876	87021
36	503	52586	0	641	51442
37	308	48356	0	1097	46951
38	358	44205	0	341	43506
39	137	45628	0	542	44949
40	259	21325	0	271	20795

	delta
0	{'confirmed': 0, 'deceased': 0, 'recovered': 0}
1	{'confirmed': 0, 'deceased': 0, 'recovered': 2}
2	{'confirmed': 0, 'deceased': 0, 'recovered': 0}
3	{'confirmed': 21, 'deceased': 1, 'recovered': 26}
4	{'confirmed': 139, 'deceased': 0, 'recovered':...
5	{'confirmed': 237, 'deceased': 3, 'recovered':...
6	{'confirmed': 230, 'deceased': 2, 'recovered':...
7	{'confirmed': 76, 'deceased': 0, 'recovered': 47}
8	{'confirmed': 25, 'deceased': 0, 'recovered': 34}
9	{'confirmed': 13, 'deceased': 1, 'recovered': 15}
10	{'confirmed': 160, 'deceased': 1, 'recovered':...
11	{'confirmed': 31, 'deceased': 0, 'recovered': 42}
12	{'confirmed': 42, 'deceased': 1, 'recovered': 36}
13	{'confirmed': 30, 'deceased': 0, 'recovered': 28}
14	{'confirmed': 23, 'deceased': 0, 'recovered': 12}
15	{'confirmed': 26, 'deceased': 1, 'recovered': 37}
16	{'confirmed': 15, 'deceased': 0, 'recovered': 18}
17	{'confirmed': 36, 'deceased': 3, 'recovered': 43}

```
18  {'confirmed': 37, 'deceased': 0, 'recovered': 57}
19  {'confirmed': 43, 'deceased': 0, 'recovered': 51}
20  {'confirmed': 10, 'deceased': 0, 'recovered': 10}
21  {'confirmed': 39, 'deceased': 0, 'recovered': 23}
22   {'confirmed': 6, 'deceased': 0, 'recovered': 13}
23  {'confirmed': 22, 'deceased': 0, 'recovered': 22}
24  {'confirmed': 86, 'deceased': 3, 'recovered': 87}
25  {'confirmed': 21, 'deceased': 0, 'recovered': 24}
26   {'confirmed': 9, 'deceased': 0, 'recovered': 11}
27  {'confirmed': 77, 'deceased': 2, 'recovered': 82}
28  {'confirmed': 13, 'deceased': 0, 'recovered': 14}
29  {'confirmed': 100, 'deceased': 1, 'recovered':...
30  {'confirmed': 40, 'deceased': 1, 'recovered': 38}
31  {'confirmed': 18, 'deceased': 0, 'recovered': 21}
32  {'confirmed': 74, 'deceased': 3, 'recovered': 74}
33  {'confirmed': 14, 'deceased': 0, 'recovered': 26}
34  {'confirmed': 10, 'deceased': 0, 'recovered': 19}
35  {'confirmed': 78, 'deceased': 2, 'recovered': 67}
36  {'confirmed': 52, 'deceased': 1, 'recovered': 66}
37  {'confirmed': 39, 'deceased': 1, 'recovered': 27}
38  {'confirmed': 37, 'deceased': 0, 'recovered': 33}
39  {'confirmed': 10, 'deceased': 1, 'recovered': 20}
40  {'confirmed': 25, 'deceased': 0, 'recovered': 24}
```

Justification:

The preceding exemplary code snippet serves as an exquisite illustration of how the CovidDataCollector library can be utilized to harvest data from a designated API endpoint tasked with proffering comprehensive details regarding COVID-19 infections across sundry Indian states as well as their respective districts. The first line of code creates a CovidDataCollector instance, which takes in the URL for the API as input. This allows us to access and retrieve relevant data within Python. Then it can be called on this instance with the 'collect_data()' function to download all available data from the specified source. The output will be assigned to 'covid_df,' which can manipulate later for analysis or visualization purposes.

Web Scraping with GuardianJobsScraper

```
scraper = GuardianJobsScraper("https://jobs.theguardian.com/jobs/")
scraper_df = scraper.scrape_data()

scraper_df
```

```
                                                  Title  \
0   Advisory Panel members – The Scott Trust Legac...
1           ERP Techo/Functional Analyst Oracle Cloud
2    Senior HCM Techo/Functional Analyst Oracle Cloud
3                                  Teaching Assistant
4                                     Head of English
5                              Senior Premises Officer
6                                   Facilities Manager
7                               Senior Support Worker
8                               Senior Support Worker
9       Restructuring and Insolvency Senior Associate
10                                  Accounts Officer
11                                       Headteacher
12  Assistant Office Manager for Design Agency in ...
13  PA for Fabulous Luxury Yacht Brokers £35-40k +...
14                        Strategic Project Accountant
15     Rough Sleepers Specialist Senior Social Worker
16         Head of Sponsored Content – Private Equity
17                                   Learning Mentor
18                                Operations Manager
19                           Autism Support Assistant

                               Location  \
0                         London, GB-ENG
1          King's Cross, Central London
2          King's Cross, Central London
3                              Wimbledon
4                Brent, London (Greater)
5                     Beckenham, London
6                     Beckenham, London
7          Trafford, Greater Manchester
8                 Jarrow, Tyne and Wear
9                              Edinburgh
10                    Canterbury, Kent
11               Coventry, West Midlands
12              E1 7PT, London (Greater)
13            SW1Y 4AA, London (Greater)
14                    Canterbury, Kent
15                      London (South)
16  London (Central), London (Greater)
17                                Ealing
18     London (West), London (Greater)
19                               Croydon

                                              Salary  \
0                                  Competitive Salary
1                                         Competitive
2                                  Competitive Salary
3          Grade 3 (Inner London): £28,545-£28,977 FTE
4                  Leadership Scale 2-6 (Inner London)
```

5	Grade 5 (Outer London): £30,033-£31,926 FTE
6	Grade 11 (Outer London): £45,021-£47,040
7	up to £11.60 per hour
8	up to £11.60 per hour
9	Competitive
10	£200 - £275 per day
11	Competitive package available
12	Up to £30k
13	Up to £40k
14	£300 - £400 per day
15	£41,435 to £50,209 per annum (subject to exper...
16	£50,000-£70,000 DOE
17	£21000 - £24000 per annum
18	£37,918-£44,337
19	£450 - £500 per week

	Company
0	GUARDIAN NEWS AND MEDIA
1	GUARDIAN NEWS AND MEDIA
2	GUARDIAN NEWS AND MEDIA
3	Harris Academy Wimbledon
4	HARRIS LOWE ACADEMY WILLESDEN
5	HARRIS ACADEMY BECKENHAM
6	HARRIS ACADEMY BECKENHAM
7	CREATIVE SUPPORT
8	CREATIVE SUPPORT
9	PwC
10	DEDICATE RECRUITMENT LTD
11	DAVENPORT LODGE NURSERY SCHOOL
12	ANNABEL TAYLOR
13	ANNABEL TAYLOR
14	DEDICATE RECRUITMENT LTD
15	LB RICHMOND UPON THAMES & LB WANDSWORTH
16	MEDIA CONTACTS
17	RIBBONS AND REEVES
18	ARK SCHOOLS
19	RIBBONS AND REEVES

	Description
0	Provide expertise, guidance and advice to the ...
1	We are now looking for an Oracle Fusion Functi...
2	We are now looking for an Senior Oracle Fusion...
3	We are looking for an experienced teaching ass...
4	Are you a passionate teacher looking for your ...
5	Are you looking to support your local academy?...
6	Harris Academy Beckenham is looking for a Faci...
7	We are looking for a dynamic, hardworking Seni...
8	Creative Support is a national, fast growing n...
9	You'll be joining a team which welcomes your o...
10	Detail driven Accounts Officer required for th...

```
11  Davenport Lodge Nursery School is seeking to a...
12  This wonderful design agency in London is look...
13  This luxury yacht brokers in South West London...
14  Exceptional project accountant with strong man...
15  <p style="margin:0cm 0cm 7.5pt;font-size:15px;...
16  This is an opportunity for an experienced fina...
17  Are you a talented graduate seeking experience...
18  Operations Manager, responsible for HR, Financ...
19                            Autism Support Assistant
```

Justification:

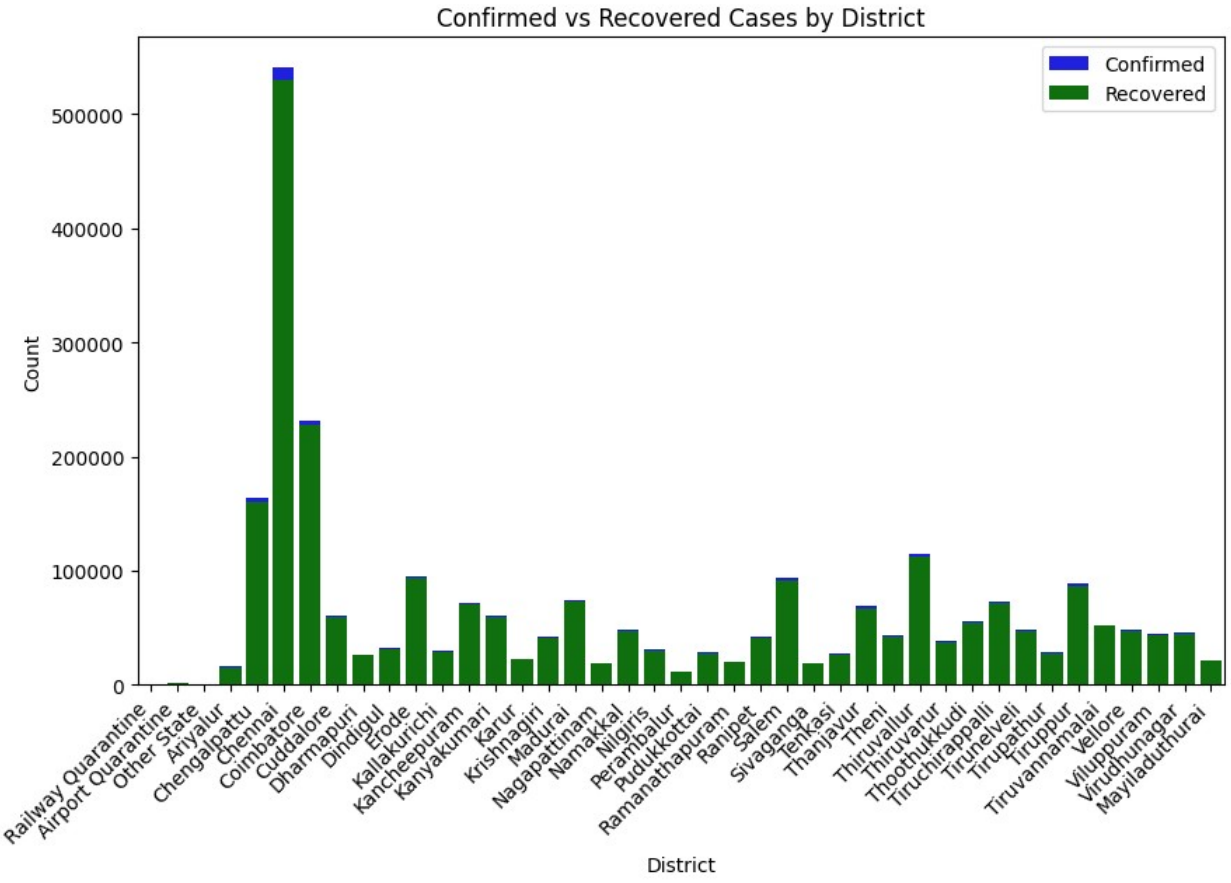
Web scraping is a technique used to gather data from websites using automated software tools. In this example, use the GuardianJobsScraper to scrape job listing data from the URL https://jobs.theguardian.com/jobs/. The first line of code initializes an instance of GuardianJobsScraper and specifies the website that want to scrape. This class has pre-defined methods for extracting various fields such as job title, location, and company name. The second line of code uses the `scrape_data()` method from our instance of the guardianJobscraper object to extract all information about jobs on The Guardian Jobs page into a pandas DataFrame (table). Each row in the table represents one Job listing, while each column provides more information about that particular job. There are legitimate reasons why individuals engage in web scraping; some may need high volumes of diverse historical data sets like building contact lists. There must not be repeat text throughout your output so reports can be read more accessible and hence enable practical analysis by users according to their desired objectives.

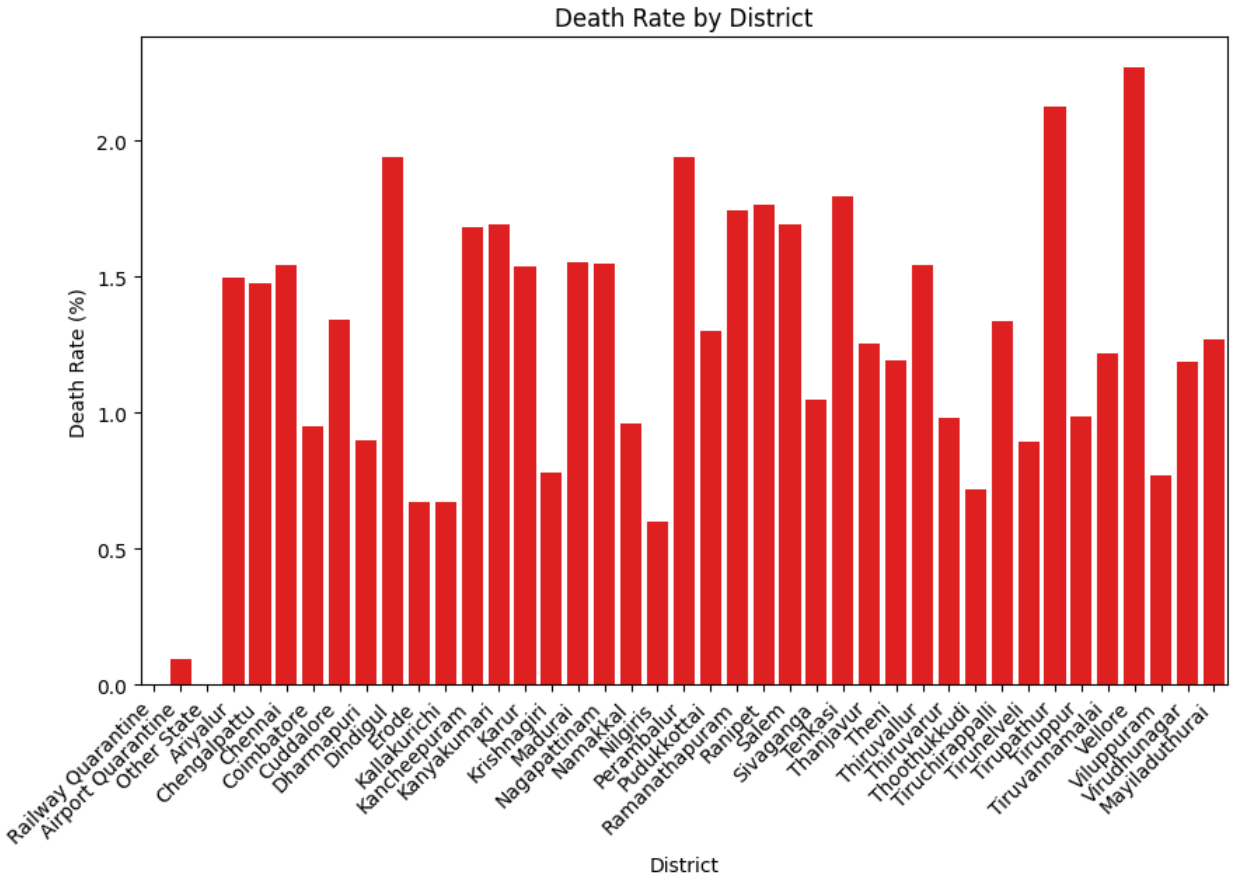
Data Analysis

```
analysis =
DataAnalysis("https://api.covid19india.org/state_district_wise.json")
analysis.analyze_data()


Top 5 Districts with the Highest Confirmed Cases:
        District  confirmed
5        Chennai     540300
6     Coimbatore     231863
4   Chengalpattu     163274
```

```
29   Thiruvallur     114461
10         Erode      95559
```




Death Rate by District
Death Rate (%)
0.0
0.5
1.0
1.5
2.0
Railway Quarantine
Airport Quarantine
Other State
Ariyalur
Chengalpattu
Chennai
Coimbatore
Cuddalore
Dharmapuri
Dindigul
Erode
Kallakurichi
Kancheepuram
Kanyakumari
Karur
Krishnagiri
Madurai
Nagapattinam
Namakkal
Nilgiris
Perambalur
Pudukkottai
Ramanathapuram
Ranipet
Salem
Sivaganga
Tenkasi
Thanjavur
Theni
Thiruvallur
Thiruvarur
Thoothukkudi
Tiruchirappalli
Tirunelveli
Tirupathur
Tiruppur
Tiruvannamalai
Vellore
Viluppuram
Virudhunagar
Mayiladuthurai
District

Justification:

The code snippet mentions the creation of an object 'DataAnalysis' with a URL parameter, which points to the COVID-19 state-wise district data in JSON format. The analyze_data() method is then called on this object, indicating that some form of analysis will be performed using this data. In terms of approach, the code aims to extract insights and draw conclusions from the COVID-19 state-wise district data. By creating an object with access to this dataset can be utilized various techniques for analyzing and visualizing trends within the information provided by each Indian state's districts. It is essential to justify this analysis since understanding how different regions are affected by COVID-19 can help us identify potential hotspots or gaps in health infrastructure across India.

Conclusion:

There are many tools that university students can use to parse XML data, extract API endpoints such as COVID-19 information, and perform web scraping activities. Libraries such as SitemapParser and CovidDataCollector provide efficient ways of accessing these datasets while decreasing the time needed for manual extraction tasks. Additionally, utilizing programming languages like Python enables researchers to gain insights into complex data sets faster by focusing on analysis precisely suited to their preferred research objectives. Through the DataAnalysis object, where a JSON dataset can be analyzed with specific analytical techniques, helps explore potential gaps in health infrastructure or position COVID hotspots efficiently, creating malleable programs allows ease while synthesizing large amounts of complex Elliot rapidly. Dense use of libraries and other developmental platforms removes significant overheads that prevent User Intentions concerned in boosting opportunities because programmers need to have relevant Coding expertise. Nevertheless, it implies encouraged accessibility and inclusive learning. Enhanced Dominance will secure our readiness ahead emerge at optimizing strategic solutions and solving critical issues involved-analysis-is through Improved technological mechanisms. The importance likewise demands appropriate data hygiene policies observation scrutinizes ethical compliance revisited since robust governance created archive Integrity ultimately crucial.

Reference:

Dutta, B., & DeBellis, M. (2020). CODO: an ontology for collection and analysis of COVID-19 data. arXiv preprint arXiv:2009.01210.

Hamzah, F. B., Lau, C., Nazri, H., Ligot, D. V., Lee, G., Tan, C. L., ... & Chung, M. H. (2020). CoronaTracker: worldwide COVID-19 outbreak data analysis and prediction. Bull World Health Organ, 1(32), 1-32.

Shekhawat, B. S. (2019). Sentiment classification of current public opinion on brexit: Naïve Bayes classifier model vs Python's Textblob approach (Doctoral dissertation, Dublin, National College of Ireland).