EXPERIMENT NO- 2

AIM: To study data collection.

RESOURCES REQUIRED: Windows/MAC/Linux O.S, Compatible version of Python.

THEORY:

Social media data is any type of data that can be gathered through social media. In general, the term refers to social media metrics and demographics collected through analytics tools on social platforms. Social media data can also refer to data collected from content people post publicly on social media. Social media data collection can help you customize your social media marketing strategy for each social network. Even more specifically, you can customize your strategy by location or demographics.

Some of the most important raw data you can collect through social media:

- Engagement: Clicks, comments, shares, etc.
- Reach
- Impressions and video views
- Follower count and growth over time
- Profile visits
- Brand sentiment
- Social share of voice
- Demographic data: age, gender, location, language, behaviors, etc.

1. Web Scraping:

Web Scraping is a technique used to extract a large amount of data from websites and then saving it to the local machine in the form of XML, excel or SQL. The tools used for web scraping are known as web scrapers. On the basis of the requirements given, they can extract the data from any website in a fraction of time. This automation of tasks is very helpful for developing data for machine learning and other purpose. They work in four steps:

- Sending the request to the target page.
- Getting response from the target page.
- Parsing and extracting the response.
- Download the data.

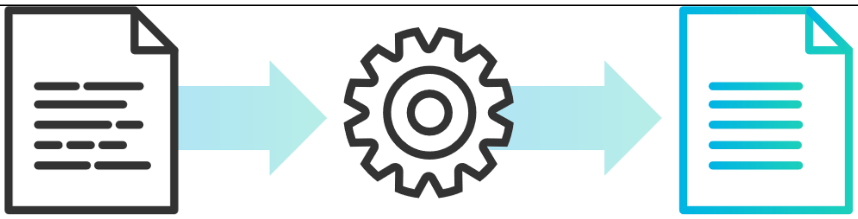
Some of the popular web scraping tools are ProWebScraper, Webscraper.io, etc.

2. Web Crawling:

Web Crawling is analogous to a spider crawling but the place of crawling here is the web!. It basically visits a website and read web pages for the purpose of building entries for search engine index. The tools that are used for web crawling are known as web crawlers or spiders. A series of web pages are analyzed and links to the pages on them are then followed for even more links thus it does a deep search for extracting of information. Famous search engines such as Google, Yahoo and Bing do web crawling and use this information for indexing web pages. Examples are Scrapy and Apache nut.

3. Data Parsing:

Data parsing is converting data from one format to another. Widely used for data structuring, it is generally done to make the existing, often unstructured, unreadable data more comprehensible.



Data parsing is a widely used method for data structuring; thus, you may discover many different descriptions while trying to find out what exactly it is. To make understanding this concept easier, we've put it into a simple definition.

What is data parsing?

Data parsing is a method where one string of data gets converted into a different type of data. So let's say you receive your data in raw HTML, a parser will take the said HTML and transform it into a more readable data format that can be easily read and understood.

CONCLUSION: Hence we have successfully studied social media data collection.

In [136…

```
import praw
import pandas as pd
```

In [137…

```
reddit = praw.Reddit(
    client_id='#',
    client_secret='#',
    user_agent='assignment',
)
```

In [138…

```
reddit.read_only
```

Out[138]:

```
True
```

Collecting post and comments from r/python

In [139…

```
posts = reddit.subreddit('python').top(time_filter='week',limit=10)
```

In [142…

```
from praw.models import MoreComments

post_list, comment_list = [],[]

def fetch_comments(comments_obj,post_id):

    q = []
    for i,comment in enumerate(comments_obj):
        if i>20:
            break
        if isinstance(comment,MoreComments):
            continue

        q.append((comment,0))

    while(q):
        comment,lvl = q.pop()

        comment_list.append({
            'post_id':post_id,
            'comment_id':comment.id,
            'parent_id':comment.parent_id,
            'body':comment.body,
            'created_on':comment.created_utc,
            'upvotes':comment.score,
            'author_id':comment.author.id if comment.author and hasattr(c
            'author_name':comment.author.name if comment.author and hasat
        })

        if lvl > 0:
            continue

        for i,neigh_comment in enumerate(comment.replies):
            if isinstance(neigh_comment,MoreComments):
                continue
            if i >10:
                break
            q.append((neigh_comment,lvl+1))




def fetch_posts(subreddit,post_types = ['rising','hot','top week'],limit=
    visited = set()
    for post_type in post_types:
        if len(post_type.split(' '))==2 and post_type.split(' ')[0]=='top
            posts = reddit.subreddit(subreddit).top(time_filter=post_type
        elif post_type =='top':
            posts = reddit.subreddit(subreddit).top(time_filter='week',li
        elif post_type=='rising':
            posts = reddit.subreddit(subreddit).rising(limit=limit)
        elif post_type=='hot':
            posts = reddit.subreddit(subreddit).hot(limit=limit)

        for post in posts:
            if post.id in visited:
                continue
            visited.add(post.id)
            post_list.append({
                'id':post.id,
                'author':post.author.name if post.author and hasattr(post
                'author_id':post.author.id if post.author and hasattr(pos
                'total_comments':post.num_comments,
                'upvote':post.score,
                'post_type':post_type,
                'title':post.title,
                'downvote':post.score*(1-post.upvote_ratio),
                'url': post.url,
                'created_on':post.created_utc,
                'subreddit':subreddit,
                'subreddit_id':post.subreddit_id
            })

            fetch_comments(post.comments,post.id)
```

In [143…

```
fetch_posts('python',['top'])
```

In [144…

```
post_list
```

```
Out[144]: [{'id': '12glkw4',
            'author': 'TheBodyPolitic1',
            'author_id': 'von3w6y2',
            'total_comments': 319,
            'upvote': 588,
            'post_type': 'top',
            'title': "Why didn't Python become popular until long after its creati
          on?",
            'downvote': 35.28000000000003,
            'url': 'https://www.reddit.com/r/Python/comments/12glkw4/why_didnt_pyt
          hon_become_popular_until_long_after/',
            'created_on': 1681051909.0,
            'subreddit': 'python',
            'subreddit_id': 't5_2qh0y'},
           {'id': '12hj9oc',
            'author': 'MetonymyQT',
            'author_id': 'shnqm',
            'total_comments': 73,
            'upvote': 510,
            'post_type': 'top',
            'title': 'Free course: Build a modern API with FastAPI and Python',
            'downvote': 25.50000000000002,
            'url': 'https://www.reddit.com/r/Python/comments/12hj9oc/free_course_b
          uild_a_modern_api_with_fastapi_and/',
            'created_on': 1681134311.0,
            'subreddit': 'python',
            'subreddit_id': 't5_2qh0y'},
           {'id': '12egsoz',
            'author': '2broke2code',
            'author_id': 'rs4dqilj',
            'total_comments': 105,
            'upvote': 465,
            'post_type': 'top',
            'title': "I trained a RoastBot on >120,000 faces and >0.5 million comm
          ents and it's a menace 😈.",
            'downvote': 32.549999999999976,
            'url': 'https://www.reddit.com/r/Python/comments/12egsoz/i_trained_a_r
          oastbot_on_120000_faces_and_05/',
            'created_on': 1680863778.0,
            'subreddit': 'python',
            'subreddit_id': 't5_2qh0y'},
           {'id': '12ffsif',
            'author': 'midnitte',
            'author_id': '3gad9',
            'total_comments': 64,
            'upvote': 385,
            'post_type': 'top',
            'title': 'EP 684: A Per-Interpreter GIL Accepted',
            'downvote': 11.55000000000001,
            'url': 'https://discuss.python.org/t/pep-684-a-per-interpreter-gil/195
          83/42',
            'created_on': 1680942397.0,
            'subreddit': 'python',
            'subreddit_id': 't5_2qh0y'},
           {'id': '12fzdu2',
            'author': 'aeluro1',
            'author_id': '88efmodhn',
            'total_comments': 12,
            'upvote': 367,
            'post_type': 'top',
          rious number of reasons, though more recently it seems that they've beco
          me another small part of [the pile](https://pile.eleuther.ai), a dataset
          that quite a few language models are trained on.",
            'created_on': 1680646662.0,
            'upvotes': 20,
            'author_id': '8um8szao',
            'author_name': 'pointmetoyourmemory'},
           {'post_id': '12bl2uj',
            'comment_id': 'jeywttj',
            'parent_id': 't3_12bl2uj',
            'body': 'your github link is not working as of 1612 central time\n\nEd
          it: 1900 central. Still getting 404\n\nEdit: this works -> https://githu
          b.com/SuperflowsAI/enron-sentiment-analysis\n\nCredit to u/ShadowDocke
          t',
            'created_on': 1680642784.0,
            'upvotes': 22,
            'author_id': '89syl1dx',
            'author_name': 'WhyDoIHaveAnAccount9'},
           {'post_id': '12bl2uj',
            'comment_id': 'jf0v8f8',
            'parent_id': 't1_jeywttj',
            'body': 'Whoops! Sorry about that. Looks like you found the right one!
          \n\nAccidently sent the private repo I was working from. Have edited the
          post to update this',
            'created_on': 1680678914.0,
            'upvotes': 2,
            'author_id': 'alzl8hco',
            'author_name': 'Ok-Craft-9908'},
           {'post_id': '12bl2uj',
            'comment_id': 'jf00txb',
            'parent_id': 't1_jeywttj',
            'body': '[deleted]',
            'created_on': 1680660368.0,
            'upvotes': 8,
            'author_id': 'deleted',
            'author_name': 'deleted'}]
```

In [146…

```python
from tinydb import TinyDB

post_db, comment_db = TinyDB('post.json'), TinyDB('comment.json')
```

In [147…

```python
post_db.insert_multiple(post_list)
comment_db.insert_multiple(comment_list)
```