

## ▼ Step 1: Importing Necessary Libraries

In this initial step, we import the Python libraries required for our project. Each library serves a specific purpose, enabling us to fetch and process Reddit data efficiently.

- **Pandas (pd):** The `pandas` library is used to manipulate and analyze data. It provides data structures like `DataFrames`, which are ideal for handling tabular data such as posts or comments retrieved from Reddit.
- **Time:** The `time` module enables us to introduce delays in our code using `time.sleep()`. This is critical when interacting with Reddit's API to prevent overwhelming the server and to comply with rate limits.
- **TQDM:** The `tqdm` library is used to create progress bars for loops. It helps visualize the progress of tasks such as downloading posts or extracting comments, making it easier to track long-running operations.
- **PRAW (Python Reddit API Wrapper):** The `praw` library simplifies interactions with Reddit's API. It provides an intuitive interface to retrieve posts, comments, and other Reddit data.
- **PRAW Exceptions:** The `praw.exceptions` module includes error classes for handling specific issues that might arise during API interactions, such as rate limits or connection errors.

```
import pandas as pd
import time
from tqdm.auto import tqdm
```

```
!pip install -q praw
import praw
import praw.exceptions
```



189.3/189.3 kB 3.6 MB/s eta 0:00:00

## ▼ Step 2: Create a Reddit application and obtain API credentials

### 1. Go to Reddit's App Preferences:

[Click here to open Reddit App Preferences.](#)

### 2. Create a New Application:

- Ensure you're **logged in** to your Reddit account.
- Scroll down to the section titled **"Developed Applications"**.
- Click on the **"Create App"** or **"Create Another App"** button.

### 3. Fill in Application Details:

- **Name:** Enter a name for your application, such as "Reddit Data Downloader".
- **App Type:** Select **"script"** (for personal, non-distributed use).
- **Description:** Write a short description (e.g., "An app to download Reddit data").
- **About URL:** Leave this blank unless you have a website for your app.
- **Redirect URI:** Enter a redirect URI (e.g., `http://localhost:8080`).
- **Permissions:** Leave as default.

### 4. Create the Application:

- After filling in all fields, click **"Create app"**.

### 5. Retrieve Your Credentials:

- **client\_id:** This alphanumeric string is located directly under the application name.
- **client\_secret:** Found in the application details and labeled as "secret".

Save these credentials securely, as they are needed to authenticate your API requests.

### 6. Authenticate with the Reddit API:

To access Reddit's data programmatically, we need to authenticate our Python application using the `praw` library.

`reddit = praw.Reddit(...)`: creates a Reddit API client object named `reddit`. This object is used for making authenticated requests to the Reddit API. The constructor takes the following parameters:

- **client\_id:** as from step 5; it's used to identify your application when making API requests.
- **client\_secret:** as from step 5; it's a secret key that, when combined with the client ID, allows your application to securely authenticate with the Reddit API.
- **user\_agent:** The user agent is a string that identifies your application and its purpose. It's important to provide a user agent that follows Reddit's guidelines, typically including the name of your application and a version number. For personal projects, you can include your Reddit username or any other descriptive information.

With this authenticated reddit object, we can now access various Reddit data and perform operations like fetching posts, comments, and more, which will be an essential part of our project. ```

```
reddit = praw.Reddit(
    client_id = '9AkNcQ17Z5pi_zo36Qrr6g',
    client_secret = 'bTQxJR7g2NvrYQZ1kNT1iipeMIGckA',
    user_agent = 'Dry_Try8800',
    check_for_async = False
)
```

### ✓ Step 3: Search for Subreddits Related to the Hashtag

The `reddit.subreddits.search_by_name` method is used to search for subreddits based on the user-provided hashtag or keyword. The following parameters are used:

- `hashtag`: The keyword entered by the user to find relevant subreddits.
- `include_nsfw=False`: Ensures that NSFW (Not Safe for Work) subreddits are excluded from the results.
- `exact=False`: Allows for partial matching, which makes the search more flexible by including subreddits that contain the keyword in their name.

The list of subreddits returned by the search is converted into a Python list, and the top 5 results are selected using slicing (`[ :5 ]`).

- If no results are found, the program exits with a message indicating that no subreddits matched the input.
- If results are found, the script iterates through the top 5 subreddits using Python's `enumerate` function to display the subreddit names and descriptions in a numbered list format. This allows the user to view the best matches and select their preferred subreddit for further processing.

```
keyword = "Italy" # Keyword to search for related subreddits

print("\nSearching for subreddits related to the hashtag...")
related_subreddits = reddit.subreddits.search_by_name(keyword, include_nsfw=False, exact=False)
# Search for subreddits matching the keyword (excluding NSFW ones)

top_subreddits = list(related_subreddits)[:5] # Get the top 5 matching subreddits
if not top_subreddits: # If no subreddits are found, exit the program
    print(f"No subreddits found related to '{keyword}'. Exiting.")
    exit()

print("\nTop 5 subreddits related to your keyword:")
for i, subreddit in enumerate(top_subreddits):
    # Display the top 5 subreddits with their names
    print(f"{i + 1}. {subreddit.display_name}")
```



Searching for subreddits related to the hashtag...

```
Top 5 subreddits related to your keyword:
1. italy
2. ItalyTravel
3. ItalyInformatica
4. italygames
5. ItalyMotori
```

### ✓ Revised Explanation for Step 4: Downloading Reddit Posts and Saving to CSV

This code retrieves a specific number of top-rated posts from the `ItalyTravel` subreddit and saves them to a CSV file. The file contains detailed information about each post, including its title, author, and other metadata.

#### 1. Define the Search Settings:

- **Subreddit**: The `subreddit_name` variable is set to `'ItalyTravel'`, focusing the search on posts related to travel within Italy.
- **Total Posts**: The `total_posts_to_retrieve` variable specifies the maximum number of posts to collect, which in this case is 1,000.
- **Time Filter**: The `time_filter` variable is set to `'year'`, limiting the search to posts from the past year.

#### 2. Initialize an Empty List for Storing Data:

- An empty list named `all_posts` is created to store the collected data.
- Each post's details will be stored as a dictionary in this list, making it easy to convert into a pandas DataFrame later.

#### 3. Retrieve and Store Each Post's Data:

- The code uses the `subreddit.top()` function to fetch posts based on their upvote score. The `time_filter` and `limit` parameters control the time range and the number of posts retrieved, respectively.

- For each post retrieved, important details are collected, including:
  - **subreddit**: The name of the subreddit where the post was published.
  - **selftext**: The body text of the post.
  - **author\_fullname**: The full name of the post's author (or 'N/A' if unavailable).
  - **title**: The title of the post.
  - **upvote\_ratio** and **ups**: The ratio of upvotes and the total number of upvotes the post received.
  - **created** and **created\_utc**: The post's creation time in standard and UTC formats.
  - **num\_comments**: The total number of comments on the post.
  - **author**: The username of the post's author (or 'N/A' if unavailable).
  - **id**: The unique identifier of the post.
- The collected data for each post is appended to the `all_posts` list.

#### 4. Alternative Search Options:

- The code provides flexibility to retrieve posts using different methods:
  - **hot()**: Retrieves currently trending posts.
  - **search()**: Searches for posts containing specific keywords (e.g., 'Rome') and allows sorting by relevance, top, new, etc.
  - Note: The `search()` function has a practical limit of 250 posts.

#### 5. Save the Data to a CSV File:

- The collected data is converted into a pandas DataFrame using `pd.DataFrame(all_posts)`.
- To ensure uniqueness, duplicate posts are removed based on their IDs using `df.drop_duplicates(subset='id')`.
- The cleaned DataFrame is saved to a CSV file named `'ItalyTravel_top_posts.csv'`.

This step generates a structured file containing metadata for up to 1,000 posts from the `ItalyTravel` subreddit, ready for further analysis.

```
subreddit_name = 'ItalyTravel' # select subreddit name
total_posts_to_retrieve = 1000 # select number of posts,
time_filter = 'year' # select among "all", "day", "hour", "month", "week",
                    # or "year", only for functions 'top' and 'search'

all_posts = []
subreddit = reddit.subreddit(subreddit_name)

for post in tqdm(subreddit.top(limit=total_posts_to_retrieve, time_filter=time_filter),
                  # as an alternative to the 'top' function, you can also use the 'hot' function
                  total=total_posts_to_retrieve, desc='Reddit posts'):
    all_posts.append({
        'subreddit': post.subreddit.display_name,
        'selftext': post.selftext,
        'author_fullname': post.author_fullname if post.author else 'N/A',
        'title': post.title,
        'upvote_ratio': post.upvote_ratio,
        'ups': post.ups,
        'created': post.created,
        'created_utc': post.created_utc,
        'num_comments': post.num_comments,
        'author': str(post.author) if post.author else 'N/A',
        'id': post.id
    })


df = pd.DataFrame(all_posts) # build dataframe
df.drop_duplicates(subset='id', inplace=True) # drop potential duplicates
df.to_csv('ItalyTravel_top_posts.csv', index=False) # export to csv
```



Reddit posts: 100%

1000/1000 [00:12<00:00, 83.57it/s]

```
df = pd.DataFrame(pd.read_csv('ItalyTravel_top_posts.csv')) # read dataframe
df # display dataframe
```



|     | subreddit   | selftext  | author_fullname | title   | upvote_ratio | ups  | created      | created_utc  | num_comments | author  |
|-----|-------------|---|-----------------|---|--------------|------|--------------|--------------|--------------|---------|
| 0   | ItalyTravel | - Did not get robbed\nDid not eat at tourist...   | t2_16tm5w       | Went to Italy twice this year... and nothing bad... | 0.94         | 1622 | 1.722175e+09 | 1.722175e+09 | 378          | prisuke |
| 1   | ItalyTravel | I loved how packed it was, I loved whenever it... | t2_16j0pv1qo1   | There isnt a thing I dont miss about italy          | 0.95         | 1029 | 1.730755e+09 | 1.730755e+09 | 257          | KarlVan |
| 2   | ItalyTravel | I'm ending my two weeks in Italy with my famil... | t2_vm2cg806     | Italian Law Enforcement                             | 0.98         | 1002 | 1.729447e+09 | 1.729447e+09 | 42           | daddee  |
| 3   | ItalyTravel | Taking the train from Venezia to Ferrara with ... | t2_ebho8        | Funny dumb scammers on Trenitalia                   | 0.99         | 975  | 1.721908e+09 | 1.721908e+09 | 179          | titan   |
| 4   | ItalyTravel | I'm sad to report that during my 3 weeks in It... | t2_kzz0t        | Spent 3 weeks in Italy and nothing bad happene...   | 0.92         | 933  | 1.720899e+09 | 1.720899e+09 | 164          | brenDae |
| ... | ...         | ...   | ...             | ...   | ...          | ...  | ...          | ...          | ...          | ...     |
| 995 | ItalyTravel | I'm in Rome currently and was thinking            | t2_15ny6x       | A speculation about ...                             | 0.59         | 7    | 1.718366e+09 | 1.718366e+09 | 19           | M       |

▼ Revised Explanation for Step 5: Extracting Comments from Reddit Posts

This code extracts comments from Reddit posts retrieved earlier. The process includes handling potential rate limits from Reddit's API and saving the extracted comments to a CSV file. Below is a step-by-step explanation:

1. Initialize an Empty List:
- An empty list, `comments_list`, is created to store the extracted comment data.
2. Loop Through Posts:
- The loop iterates over the first `num_posts` posts in the DataFrame `df`. For each post, its unique ID (`post_id`) is retrieved, which is used to fetch the corresponding submission object from Reddit.
3. Handle Comments with try-except :
- A `try-except` block ensures that potential exceptions (e.g., rate limit errors) during comment extraction are handled gracefully.

◦ If an exception occurs, the program waits for 1 second (`sleep(1)`) before continuing.
4. Retrieve All Comments:
- The `submission.comments.replace_more(limit=None)` method ensures all comments are retrieved, even those hidden behind Reddit's "load more comments" feature.
5. Extract Comment Data:
- For each comment in the post, the following details are extracted:

▪ `comment_id`: The unique identifier for the comment.

▪ `parent_id`: The ID of the parent post or comment to which this comment is replying.

▪ `post_id`: The ID of the original post to which the comment belongs.

▪ `comment_body`: The text content of the comment.
6. Append Data to List:
- The extracted data for each comment is added to `comments_list` as a dictionary. This ensures that all comments from multiple posts are stored in a single structure.
7. Convert to DataFrame:
- After processing all posts, the collected comments are converted into a pandas DataFrame named `comments_df`. Each row represents a single comment with its corresponding metadata.
8. Save to CSV:
- The `comments_df` DataFrame is saved as a CSV file named `'ItalyTravel_top_comments.csv'` without an index column.

```
# here we run it for the first five posts only
# replace 5 -> len(df) for the all the posts

num_posts = 5

comments_list = []

for i in tqdm(range(num_posts), desc='Reddit comments'):
    post_id = df.iloc[i]['id']
    submission = reddit.submission(id=post_id)
    try:
        submission.comments.replace_more(limit=None)
        for comment in submission.comments.list():
            comments_list.extend([{'comment_id': comment.id,
                                   'parent_id': comment.parent_id,
                                   'post_id': post_id,
                                   'comment_body': comment.body,
                                   }])

    except PossibleExceptions:
        print("Handling replace_more exception")
        sleep(3) # create delay

comments_df = pd.DataFrame(comments_list) # build dataframe
comments_df.to_csv('ItalyTravel_top_comments.csv', index=False) # export to csv
```

 Reddit comments: 100%

5/5 [00:05<00:00, 1.00it/s]

```
comments_df = pd.DataFrame(pd.read_csv('ItalyTravel_top_comments.csv')) # read dataframe

comments_df # display dataframe
```



|      | comment_id | parent_id  | post_id | comment_body                                      |
|------|------------|------------|---------|---|
| 0    | lfc3qm0    | t3_1ee7uu2 | 1ee7uu2 | Ciao! Welcome to r/ItalyTravel. While you wait... |
| 1    | lfc9mzy    | t3_1ee7uu2 | 1ee7uu2 | I am sorry your travel did not meet up expecta... |
| 2    | lfc1wt     | t3_1ee7uu2 | 1ee7uu2 | I go every year, usually twice, popping down f... |
| 3    | lfc44xn    | t3_1ee7uu2 | 1ee7uu2 | Of course. Italy is a civilized country with f... |
| 4    | lfc60if    | t3_1ee7uu2 | 1ee7uu2 | You have to be a regular on this Reddit to und... |
| ...  | ...        | ...        | ...     | ...   |
| 1003 | ld26gx5    | t1_ld1v6ys | 1e2ijur | Your post or comment was removed because it vi... |
| 1004 | ld1vmvf    | t1_ld1vg2n | 1e2ijur | Lol, whatever dude. Have fun going to migrant...  |
| 1005 | ld3dluj    | t1_ld1vmvf | 1e2ijur | What like Venice, Sorrento, and Capri? You've...  |
| 1006 | ld3u05a    | t1_ld3dluj | 1e2ijur | Not saying where else I went cause I don't wan... |
| 1007 | ld3y0z6    | t1_ld3u05a | 1e2ijur | You honestly think somewhere would get popular... |

1008 rows x 4 columns