# lab5_tracking

February 9, 2024

## 1 Lab 5: A tracking function

Let's say you're a sneaky developer, who wants to create a function to help people decide whether to join a subreddit or not. However, you also want to harvest data about their interests to sell to advertisers!

You need a function that will help collect as much data from them as possible. To do that, write a function that accepts a list of interests, as well as a target subreddit. Collect submissions from that subreddit to see whether any of those interests appear in the title or body of those submissions, and if they do, recommend that subreddit to the user.

However, you will also be storing those interests in a hidden variable, which you will send to yourself as a Reddit message.

First, go through the normal praw steps

[25]:
```
import praw
```

[26]:
```
%run reddit_keys.py
```

[27]:
```
reddit = praw.Reddit(
    username=username, password=password,
    client_id=client_id, client_secret=client_secret,
    user_agent="a custom python script for user /" + str(username)
)
```

If you want to check if a string contains a substring in python, it's helpful to make sure they're both lowercase, like this:

[28]:
```
"Hi There".lower()
```

[28]:
```
'hi there'
```

If you want to get the text from a submission on Reddit, you can use .title to get its title, or .selftext to get any text from its body (note that selftext is often just an empty string, if the post is just an image or link). For example:

[29]:
```
submissions = reddit.subreddit("sports").hot(limit=5)
submissions_list = list(submissions)
print(submissions_list[0].title)
```

```
print(submissions_list[0].selftext)
```

```
Join The Reddit Sports Discord Server!
 [**Join Our Discord Server!**](https://discord.gg/233aU5q)

**Welcome to** [**/r/sports**](https://old.reddit.com/r/sports)

We created a Discord server for our community and would like to invite all of
you to join! You'll be able to discuss sports with users around the world and
discuss events in real time!

There are separate channels for many sports you can opt in and out of,
including:

  Football
 Basketball
 Baseball
 Soccer
 Hockey
 Cricket
 Rugby
 Motorsports
 Fighting
 Golf
 Tennis
And many more!

[**Reddit Sports Discord Server**](https://discord.gg/233aU5q)
```

Now, define a function that takes two parameters: a list of interest, and a subreddit to check out. If any of the words in the list of interests appears in the title or body of submissions in that subreddit, tell the user they should join!

But remember, you also want to store their interests, and the subreddit, in a hidden variable. Get both their interests, and the subreddit they chose...

[38]:
```
user_data = {}

def should_join(interests, sub):
    global user_data

    user_data['interests'] = interests
    user_data['subreddit'] = sub
    recommended_submissions = []

    for submission in reddit.subreddit(sub).hot(limit=10):
        submission_title = submission.title.lower()
        submission_body = submission.selftext.lower()
```

```
        for interest in interests:
            if interest.lower() in submission_title or interest.lower() in␣
↪submission_body:
                recommended_submissions.append(submission.title)
                break
    if recommended_submissions:
        print("Based on your interests, we recommend joining the subreddit '{}'.
↪".format(sub))
        print("Here are some recommended submissions:")
        for submission_title in recommended_submissions:
            print("- " + submission_title)

    else:
        print("No relevant content found in the '{}' subreddit based on your␣
↪interests.".format(sub))

should_join(["campus", "sports"], "work")
```

```
No relevant content found in the 'work' subreddit based on your interests.
```

Now, modify your code from above, so that it sends you the data in a private message right before printing a message to the user. You can use the following function to send private messages on Reddit:

reddit.redditor(“”).message(subject=””, message=””)

[44]:
```
user_data = {}

def should_join(interests, sub):
    global user_data

    user_data['interests'] = interests
    user_data['subreddit'] = sub
    recommended_submissions = []

    for submission in reddit.subreddit(sub).hot(limit=10):
        submission_title = submission.title.lower()
        submission_body = submission.selftext.lower()

        for interest in interests:
            if interest.lower() in submission_title or interest.lower() in␣
↪submission_body:
                recommended_submissions.append(submission.title)
                break

    if recommended_submissions:
```

```
        print("Based on your interests, we recommend joining the subreddit '{}'.
↪".format(sub))
        print("Here are some recommended submissions:")
        for submission_title in recommended_submissions:
            print("- " + submission_title)

        message_subject = "User Interests and Subreddit"
        message_body = "Interests: {}\nSubreddit: {}".format(interests, sub)
        reddit.redditor("Flashy_Tea_1482").message(message_subject,␣
↪message_body)

    else:
        print("No relevant content found in the '{}' subreddit based on your␣
↪interests.".format(sub))

should_join(["python", "sports"], "learnmachinelearning")
```

```
Based on your interests, we recommend joining the subreddit
'learnmachinelearning'.
Here are some recommended submissions:
- Huge impact in training time by reducing the number of reading operations from
disk by using a cache in the Dataset object.
- Moving from MATLAB to Pytorch - Help; I feel like a newborn learning to walk

/tmp/ipykernel_282/371184371.py:27: DeprecationWarning: Positional arguments for
'MessageableMixin.message' will no longer be supported in PRAW 8.
Call this function with 'subject' and 'message' as keyword arguments.
  reddit.redditor("Flashy_Tea_1482").message(message_subject, message_body)
```

[ ]: