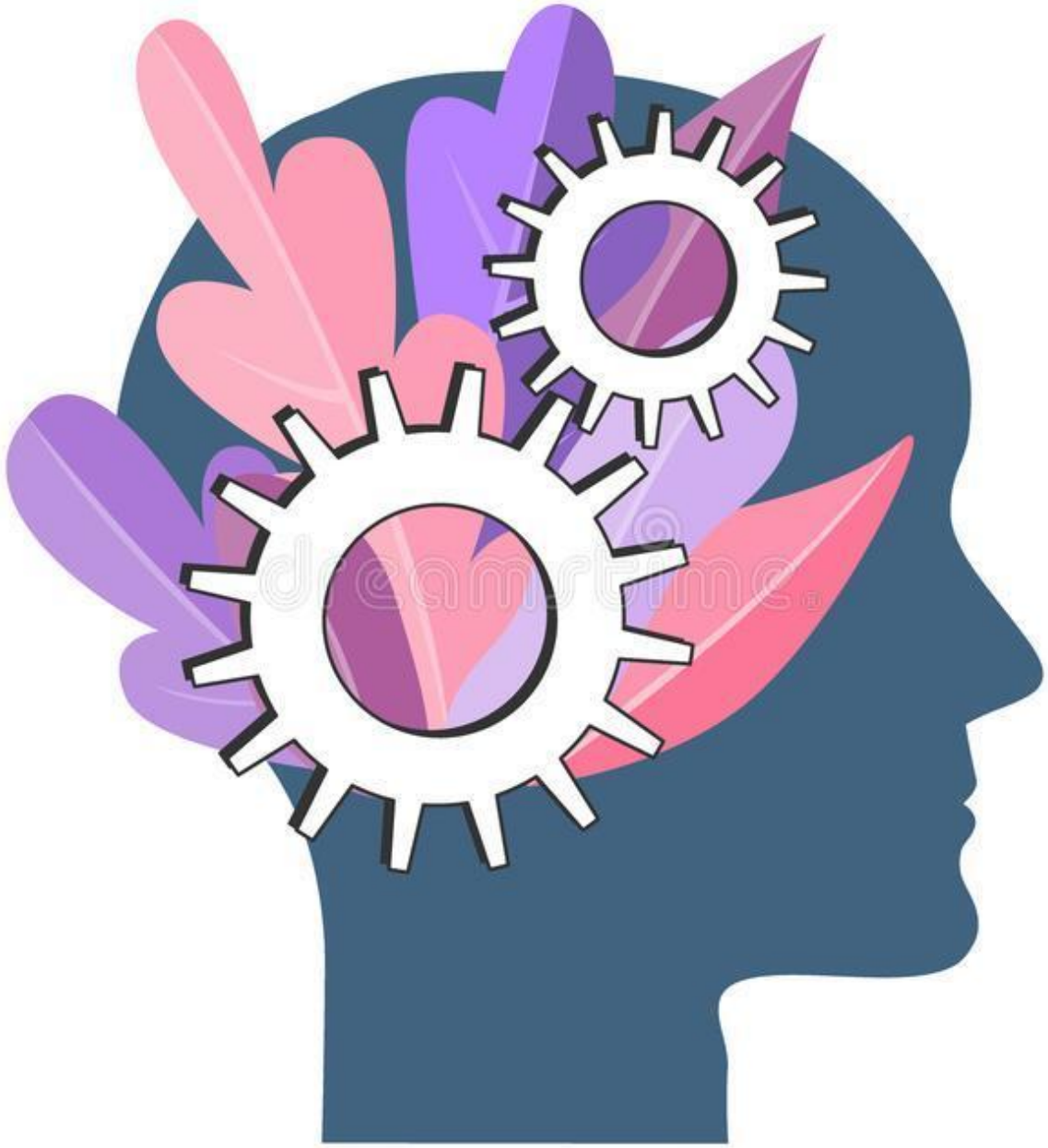
Work Culture Project

Week 4 Updates

Feb 9 2023

Presenter:

Mitanshi Vyas (MS Student)



Overview

- Conducted a detailed analysis of a worker interacting with Toloka Tasks Tab, to provide an estimate duration of task tab navigation for each of those workers
- Visualizations to compare the average time spent by each worker just browsing through the tasks on the website
- Analysis of patterns between task with which the workers just interacted one time versus the task with which the workers interacted multiple times
- Time intervals for which a worker worked on a task

Functions

Below are several similar functions that I have created to fetch these insights with their respective description and parameters:

1. **avg_navigate_time(user_id, threshold)** : The function will take the user id and the a threshold and will calculate avg time taken by the user to navigate through the tasks
2. **task_tab_navigation(worker_id)** : The function will take the user_id and would find the average navigation time on task tab using a different strategy
3. **interacted_once(user_id):** This function will take user_id as a parameter to return the lists of tasks which were interacted with just once, with their pool_ids and titles
4. **interacted_most(user_id, n):** This function takes two arguments. One being the user_id and second is a 'n' which defines how many mostly interacted tasks do we want the function to return
5. **find_time(user_id):** This function takes user_id as an argument and then provides with a list of tasks attempted by the worker. It then takes the pool_id as an input from the user and then finally returns with the time intervals that the worker has worked on that task.

Snippet of the Log Data

uid	event	timestamp	url
1699065816	TABUPDATED	1668886728408	https://toloka.yandex.com/tasks
1699065816	WEBREQUEST	1668886728510	https://toloka.yandex.com/tasks
1699065816	TABUPDATED	1668886738800	https://toloka.yandex.com/tasks
1699065816	TABUPDATED	1668886888745	https://toloka.yandex.com/tasks
1699065816	TABUPDATED	1668886942803	https://toloka.yandex.com/tasks
1699065816	TABUPDATED	1668886974590	https://toloka.yandex.com/tasks
1699065816	TABUPDATED	1668887067760	https://toloka.yandex.com/tasks
1699065816	WEBREQUEST	1668887068121	https://toloka.yandex.com/tasks
1699065816	TABUPDATED	1668887105018	https://toloka.yandex.com/tasks
1699065816	TABUPDATED	1668887210613	https://toloka.yandex.com/tasks
1699065816	TABUPDATED	1668887259012	https://toloka.yandex.com/tasks
1699065816	TABUPDATED	1668887283278	https://toloka.yandex.com/tasks
1699065816	TABUPDATED	1668887320119	https://toloka.yandex.com/tasks
1699065816	WEBREQUEST	1668887401632	https://toloka.yandex.com/tasks
1699065816	TABUPDATED	1668887455650	https://toloka.yandex.com/tasks
1699065816	TABUPDATED	1668887471809	https://toloka.yandex.com/tasks
1699065816	WEBREQUEST	1668887473691	https://toloka.yandex.com/tasks
1699065816	TABUPDATED	1668889086228	https://toloka.yandex.com/tasks
1699065816	TABUPDATED	1668889100139	https://toloka.yandex.com/tasks
1699065816	TABUPDATED	1668889307237	https://toloka.yandex.com/tasks

Several Results from Functions

Function Name: `avg_navigate_time(user_id, threshold)`

Methodology: While querying the time duration between events for each user, I was receiving a large amount of duration in minutes. So it becomes very difficult to exactly understand if these time durations are real or not. Since, it is not possible to have such large duration of task navigation time, I used a different strategy in which I used a threshold of duration to limit the maximum duration of task navigation and find the average task navigation time for each user.

Outputs:

```
[48] avg_navigate_time(1716528528, 40)

     Average time taken by user 1716528528 to navigate Task tab: 21.153921568627453 Minutes
```

```
[14] avg_navigate_time(1458049517, 10)

     The user only has one log entry for Task Tab, hence we cannot find their average navigation time.
```

Several Results from Functions

Function Name: `task_tab_navigation(worker_id)`

Methodology: In the previous methodology, we we were receiving large durations of time for task navigation, which was very unrealistic. So this time I tried utilizing the 'url' column to check where exactly the user is switching the tab. And with the help of that I was able to get more precise and realistic results. I also incorporated the 'event' column but in a slightly different way.

Outputs:

```
task_tab_navigation(1718146027)
```

```
Average Task tab navigation duration for user 1718146027 is 7.766666666666667 minutes.
```

```
task_tab_navigation(1718416044)
```

```
Average Task tab navigation duration for user 1718416044 is 5.791666666666667 minutes.
```

```
task_tab_navigation(1699065816)
```

```
Average Task tab navigation duration for user 1699065816 is 11.406666666666666 minutes.
```

Several Results from Functions

Function Name: `find_time(user_id)`

Methodology: This function takes user_id as an argument and then provides with a list of tasks attempted by the worker. It then takes the pool_id as an input from the user and then finally returns with the time intervals that the worker has worked on that task.

Outputs:

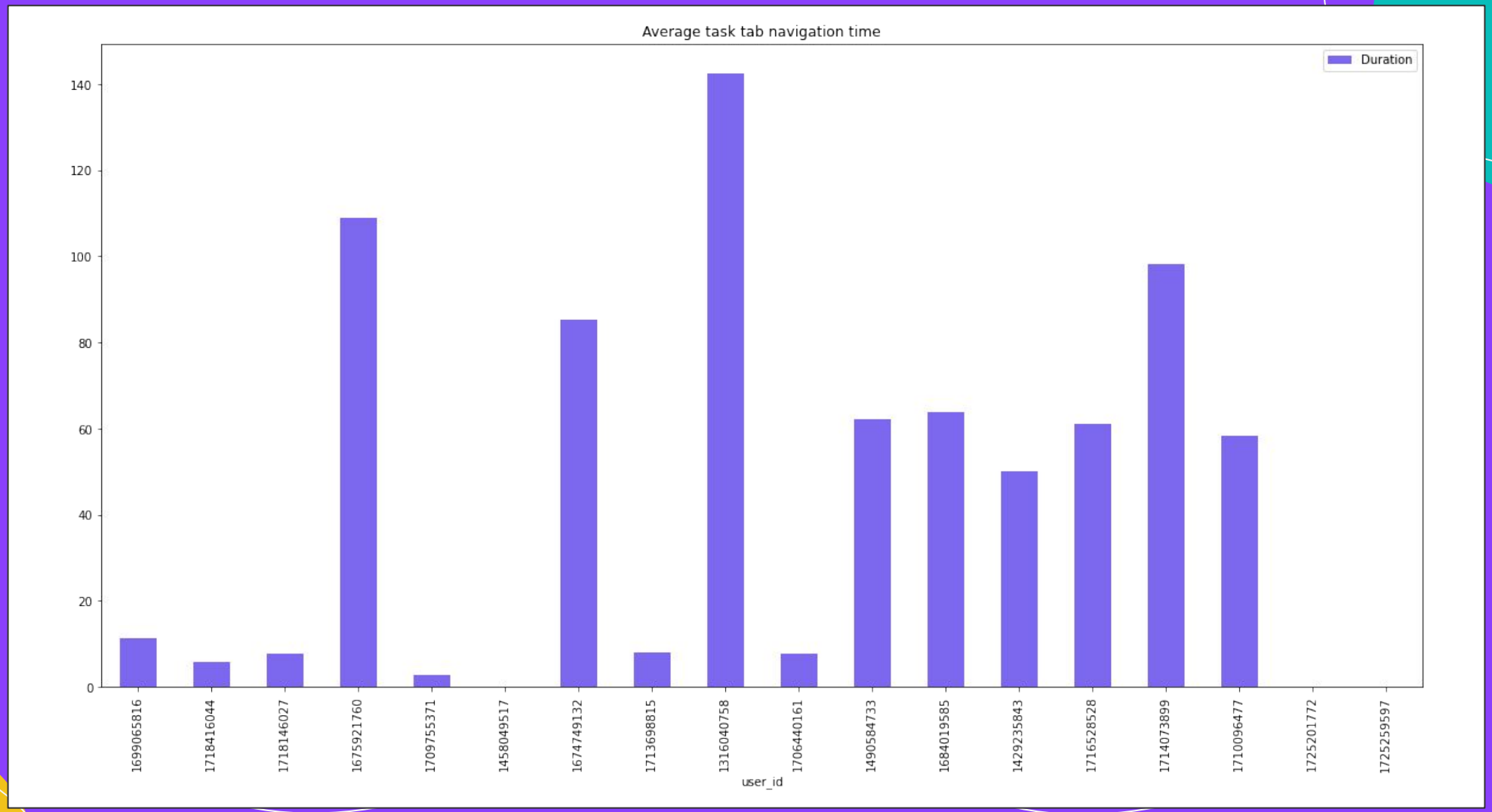
```
find_time(1316040758)

['36760642' '35992744']
0
```

```
find_time(1316040758)

['36760642' '35992744']
035992744
0
Duration for interval 0 : 52.65 minutes.
1
Duration for interval 1 : 62.766666666666666 minutes.
```

INSIGHTS

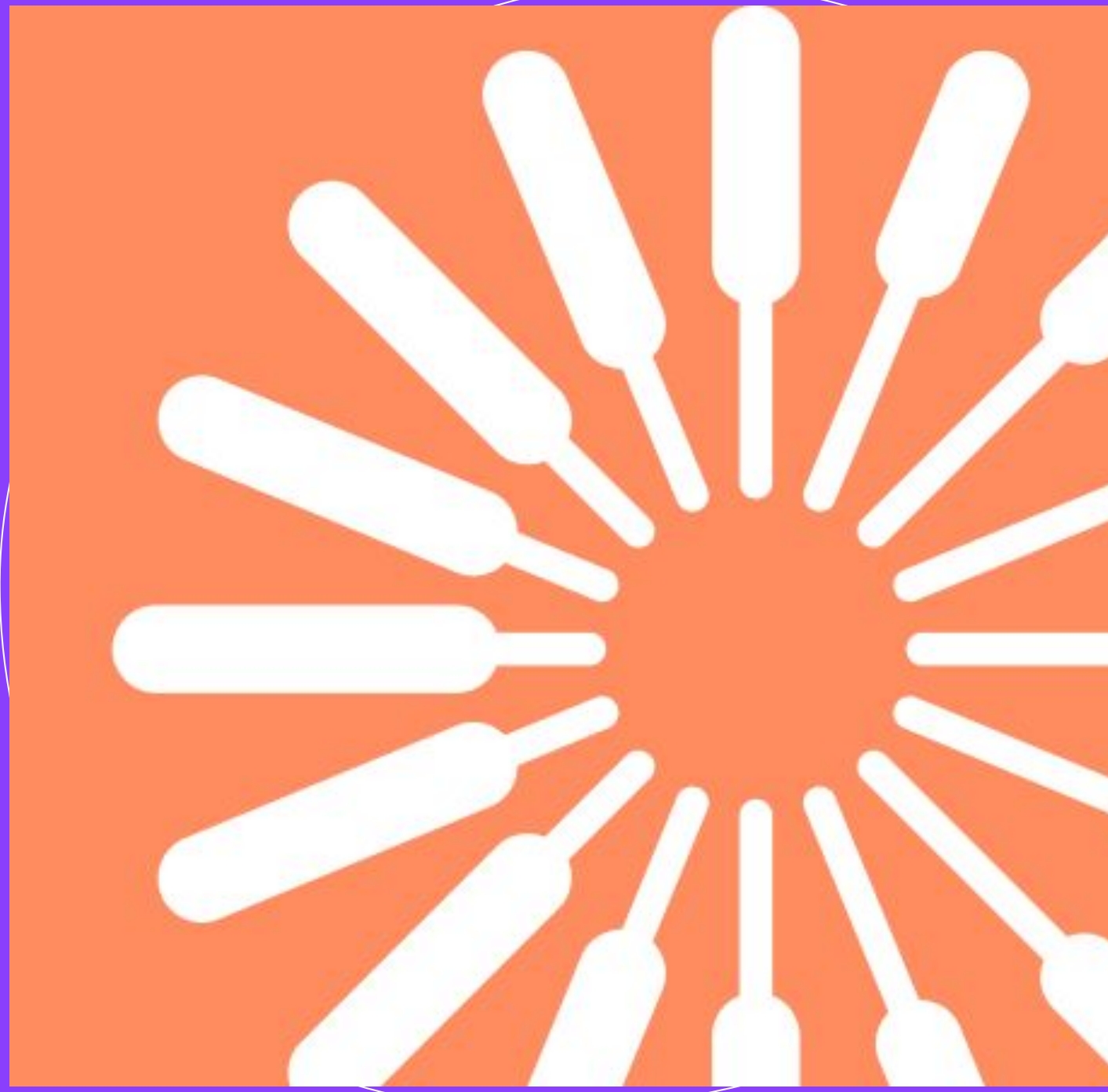

Average task tab navigation time
Duration
140
120
100
80
60
40
20
0
1699065816
1718416044
1718146027
1675921760
1709755371
1458049517
1674749132
1713698815
1316040758
1706440161
1490584733
1684019585
1429235843
1716528528
1714073899
1710096477
1725201772
1725259597
user_id

Appendix

Python Scripts:

https://colab.research.google.com/drive/14ma9u61W59pVucN0eQgLOqF1bJaMoWjX?usp=sharing

Thank You!



Future Approach

- For the future approach I am planning to consider ‘scroll count’ as a factor to further improve the accuracy of our navigation time calculation

- Make task duration more accurate by considering either ‘url’ column, cross checking with income_data to verify a task was completed