

HA1 - Social Network Influencers

Your task is to write a program for **finding the influencers in a social network**.

The solution should allow for implementation of multiple strategies to calculate the *influence score* of each user. Upon execution, the user chooses one strategy which will be applied.

Input

The input data is one or both of:

- followers table - containing info about who follows who on this social network
- activity table - containing all posts, likes and shares made by the users

More information can be found in *data_explanation.xlsx*

Output

The output should be a table of users, with columns `influence_score` and `influence_category`.

- `influence_score` is a real number, its definition depends on strategy.
- `influence_category` is a string with values "*high*", "*mid*", "*low*" and is calculated from `influence_score` so that:
 - "*high*" = top 10% of users
 - "*mid*" = top 10-50% of users
 - "*low*" = bottom 50% of users

Solution Requirements & Evaluation

1. Implement 1 simple strategy that ranks the users by number of followers. In this case the `influence_score` will be number of followers.
2. Implement 1 more complicated strategy for ranking the users in a network of followers, using both info about followers and user activity. In this scenario, you define the `influence_score`.
3. Make the program configurable. As an input argument program should take some kind of config file in which the user can specify the following: desired strategy for finding influencers, paths to inputs and output table, optionally some strategy-specific parameters.

Your solution should save the output table and print out the distribution of users across categories and top 5 influencers. Implement `__str__` method for strategies to return it's basic info: name and number of users that are being ranked.

The solution should:

- be well documented
- have a file listing all packages required for the program execution
- enable easy implementation of new strategies (in a format you specify) without having to modify other parts of code
- optional: have a few unit tests

How to design the code architecture and input is entirely up to you, but please provide **clear instructions** how to run the code.

While evaluating the solution we will also be taking into account:

- the idea and implementation of 2nd strategy
- python best practices
- code readability and cleanliness
- memory and time optimization
- attention to detail

Example of Solution

Example tables are provided in *activity_example.feather* and *followers_example.feather* files.

	user_id	user_id_follower
0	11	22
1	11	33
2	11	44
3	11	55
4	22	11
5	22	44
6	22	66
7	33	55
8	33	66
9	55	11
10	55	66

	post_id	post_user_id	post_date	is_like	liked_post_id	liked_user_id	is_share	shared_post_id	shared_user_id
0	1	11	2022-04-20	False	NaN	NaN	False	NaN	NaN
1	2	22	2022-04-20	True	1	11	False	NaN	NaN
2	3	33	2022-04-20	False	NaN	NaN	True	1	11
3	4	22	2022-04-20	False	NaN	NaN	False	NaN	NaN
4	5	44	2022-04-20	True	1	11	False	NaN	NaN
5	6	11	2022-04-20	False	NaN	NaN	True	4	22

In this example users 22,33,44,55 follow user 11. User 11 posted an original post 1, that was liked by users 22 and 44 and shared by user 33. Also user 22 posted an original 4, that was shared by user 11.

Ranking of influencers that only counts the number of followers would output this:

	user_id	influence_score	influence_category
0	11	4	high
1	22	3	mid
2	33	2	low
3	55	2	low

Note that only users from user_id column in followers are included, although other users appear in activity table (44), and user_id_follower column (44, 66).