1. Solution

1.1. Trending market segments

In order to compute ranking of the top trending segments within crowd finance at the end of a given time period, one simple method is to compute the number of times the concepts occur in campaigns for each time segment, we then use this figure to rank the trending market segments. The limitation of this method is that it does not take into account of different concepts in time segments while this figure is important to market trending. Moreover, the number of concepts which occur does not necessarily reflect the market trending since it lacks informative insights. To solve this problem, we need to take individual concepts in each time segment into account. The solution can be described as follows:

We first split the given time period into month or day segments depending on the range of time period. For each time segment with start and end dates, we search for all campaigns that occur during that time segment and collect a list of concepts that are extracted from all resources (title, subtitle, category and description). The frequency of each concept will then be computed. We now compute a weight for each concept across the whole dataset, this can be done by computing the probability of a concept that occur in all campaigns. This probability is simply estimated by taking the frequency of a concept divided by the total number of time that concept occurs. Next, we compute a score for each time segment (s_i, e_i) as $r = \sum_c w(c)f(c)$, where s_i, e_i are the start and end dates, c is an individual concept and $w(c)$ and $f(c)$ are weight and frequency of concept c in that time segment. We then use this score to rank the trending market segments. In the case that there is no campaign that occurs during a specific time segment, we then rank this time segment as zero.

1.2. Market index

We can use value weighted index to calculate market index. For each time segment, we compute the total amount raised by all campaigns that end before

the end date of time segment (if there is no available campaign then we set the corresponding market index to zero). We set the market index of the first time point in given time period to default value 1000. The market index of current time point can be computed as $MI = 1000.\frac{r_c}{r_p}$, where r_c, r_v are the total amount raised by all campaigns of the current and the previous time points, respectively.

This is a good indicator of importance in market. However, this method sometimes favours big campaigns which are raising huge amounts of money that may dominate the index. This is because we did not take into account of relative ratio between the raised and goal amounts in each campaign. A better solution would be to replace the total amount raised by all campaigns by the ratio of the total raised amount and total goal amount. The greater this ratio is, the higher weight it gains.

1.3. Campaign search

In order to search for all campaigns that occur during a given time segment, we just need to filter all campaigns with time periods that overlap the given time segment. To do this procedure efficiently, we build a time interval tree where each node of tree is a time interval $(start, end)$. This data structure requires $O(n)$ storage space and $O(log(n)+m)$ query time with n being the total number of intervals and m being the number of reported results.

2. Discussion

The reason I chose these solutions is because they are simple and straightforward. For the trending market segments, the probabilities of concepts can easily be computed when extracting the data, the score of each time segment is therefore quickly computed without needing to build a separate function. As a result, it saves time and is computationally efficient. For the market index, it is also quick to compute market index for each time segment by ultilising the previously computed results. We can improve the solution by computing the ratio of raised amount and goal amount by all campaigns.

3. How to use solutions

The solutions were structured in following files:

- test.py: This file contains the main program.
- market_index_testcase.py: This file declares testcases used to test the function that computes market index.
- market_segment_ranking_testcase.py: This file declares testcases used to test the function that computes trending market segment.
- time_segment_split_testcase.py: This file declares testcases used to test the function that split a given time period into time segments.
- functest.py: This file contains functions which are used to test.

In order to test functions, you just need to run the corresponding test case files. I used file $test1.json$ as default data to test my code.

You can run file *test.py* to start doing the analysis. Once it starts, the program will first need to load the data from $projects.json$ file, this process may take some time. It then prompts you to enter a time period (start and end dates). You can use appropriate keys to enter start and end dates. The program will then start computing the trending market segment and market index and store the outputs into files named $segrank.json$ and $market\_index.json$, respectively. The json files are structured in two fields *time* and *rank* or *index* where *time* contains time segments and *rank* or *index* contains ranking scores or market indices, respectively.