

CS230 Project Report:

Design requirements:

Goal: The purpose of this project is to create an application that aims to assist a recruiting team finalize a selection procedure, centered around 12 finalist applicants of an influencer coordinator position in a social media company. The finalists were tasked to assume the role of an influencer, and the data collected reflects their ability to understand trends and manufacture engagement. To assist in the goal of assessing the finalists' performance, the application provides the recruiting team with 7 tools aimed towards the sorting, analysis, and display of provided data, through a concise menu interface.

Option 1: Displays the following data fields: Last Name, Followers, Posts, Views, Engagements, Likes, Comments, Shares.

Option 2: Calculates and displays per post data for all metrics.

Option 3: Prompts the user to input a lower limit of likes per post and then filters and displays the details and likes per post metric only for the applicants that scored above the limit. The user may choose to view the unfiltered data again by entering 505 as the limit. The user is not allowed to enter a limit above 500 since it is exceedingly high, zero since no filtering will take place, or below zero since all data are strictly positive. While the data will be originally presented unfiltered to the user to help him make his choice, they will be pre-sorted on the likes per post metric to provide further clarity on the choice.

Option 4: Calculates a new metric ($[(\text{likes}) \times (\text{comments}) \times (\text{shares})] / (\text{posts})$) and saves it in a new column, along with all other metrics, in a file called social_media_contest_processed_data.txt. Then, some of the data are read from that file and displayed on the screen sorted on the new metric in descending order.

Option 5: Calculates per post metrics as well as the new metric and allows the user to sort on any of them. Applicants with zero shares have been given one to lead to more logical calculations. The user is presented with a menu to make his choice and he can cycle through multiple iterations of the data without returning to the main menu. Furthermore, the user can choose the order of the sorting. Every time the user makes a different field selection, the data is sorted based on that field and then all per post metrics as well as the new metric are recalculated and displayed. The recalculation allows us to avoid calling many unnecessary parameters to the sorting function. Once he chooses to exit the option 5 submenu, the user is returned to the main menu. The final sorted arrangement of the user is saved in social_media_sorted_stats.txt.

Option 6: The data is sorted on the country of origin of the applicants in alphabetical order and displayed. An alteration of the sorting method is used to accommodate string values.

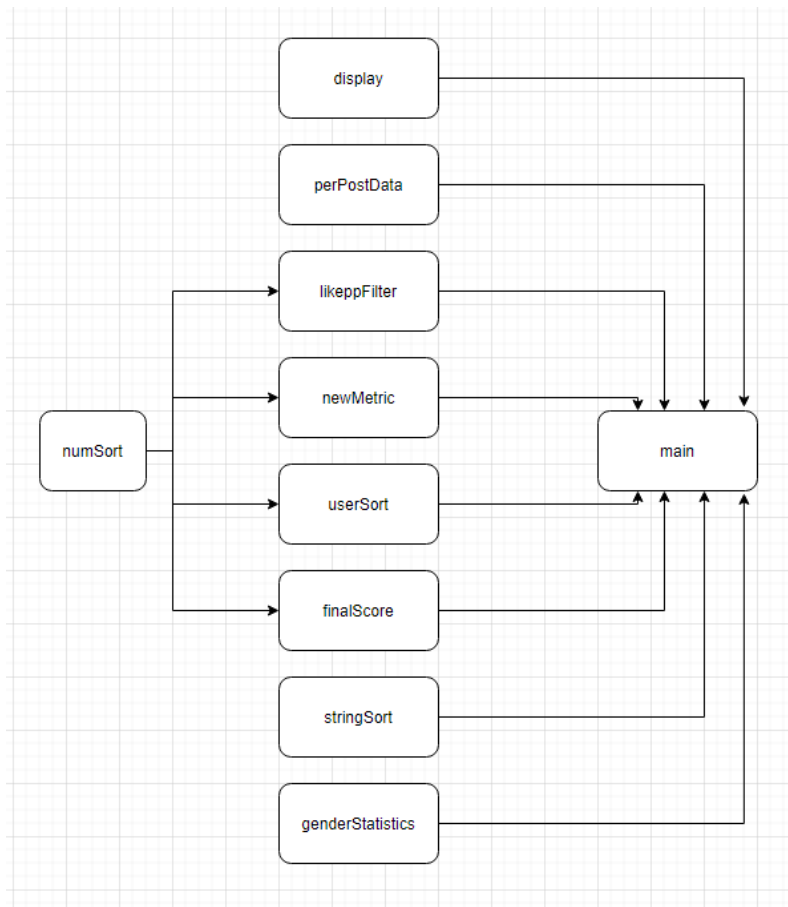
Option 7: The total sums of all metrics are calculated for each gender and displayed. We observe that with a little more activity, the female applicants outperformed the male by a large amount in all metrics. This result, according to Lokithasan, et. al (2019) is not indicative of anything other than the current trends benefit female dominated markets when it comes to activity and engagement, rather than male dominated ones. The outreach, thus, of the female participants overall tends to be larger than the male's in the market of the time this data was recorded.

Option 8: Calculates a final score for every candidate based on the performance, and displays all metrics sorted on that final score. A detailed explanation is given to the user to inform on the scoring process. The data are also saved in social_media_final_score.txt.

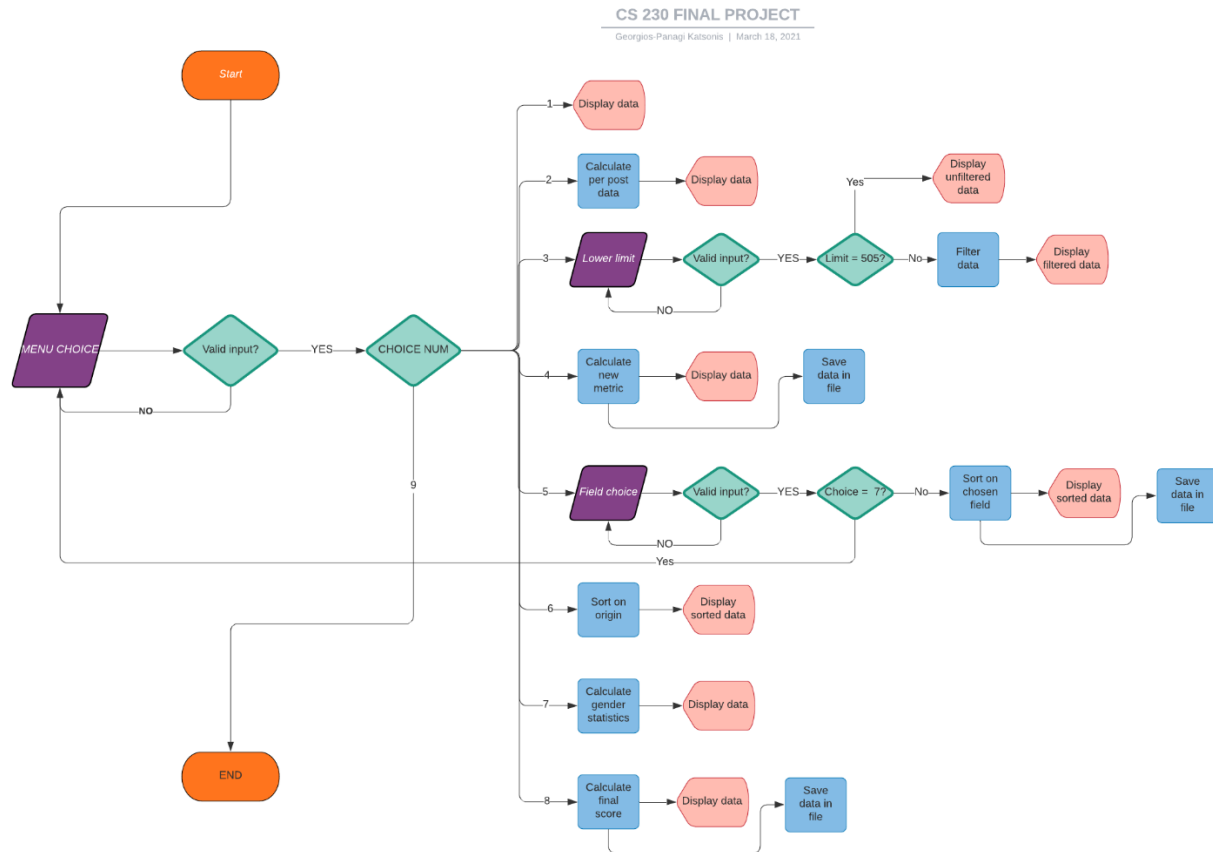
numSort: This method is used to sort the entire metrics table based on any number array, in either descending or ascending order. A global array is populated with the selected number array and then sorted using a simple insertion sort. At every sorting step the movements of the elements are copied to effectively sort all other metric arrays. Finally, the original number array is repopulated using the values of the global array.

stringSort: A copy of numSort altered to accommodate a string array as an input.

Function Interconnection diagram:



Flowchart:



Note: The flowchart will be also included as a separate pdf for easier viewing.