Teammates:

Tisha Kathrani
Siya Jain
Ankita Khatri
Ruhi Aggarwal

ChatGPT Statement:

While we were debugging some of our code, we used ChatGpt to help us understand what the source of the bug was and suggestions for how to fix the errors we were getting. We found it useful because it actually explained what type of error was occurring and where in the code the issue may be. For example, often, small things like imports were missed and ChatGPT caught those for us. We also didn't fully understand how indexes worked two weeks ago when looking at the assignment and ChatGPT was able to explain it in further detail after we had looked at the lectures. Overall, it was useful for gaining knowledge about some concepts we were not fully sure on and helping us understand our bugs.

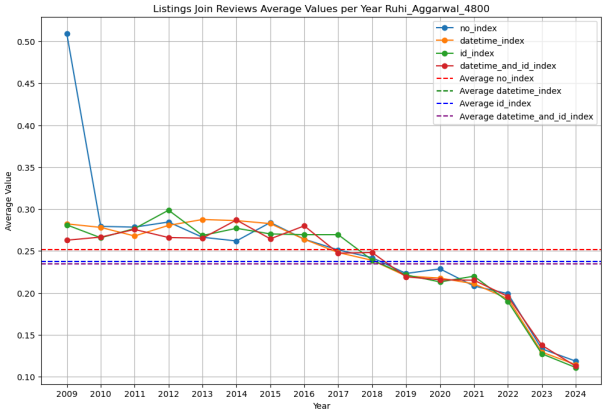
Who did what?

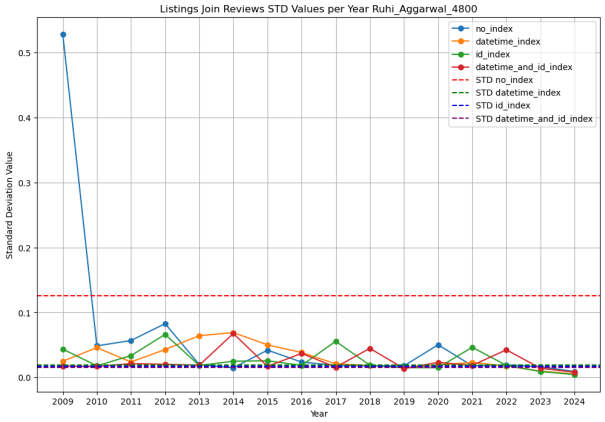
Ruhi: Siya and I worked on writing the code in the main jupyter notebooks to create the output of all 3 of the json files for steps 3a, 3b, and 3c. I also worked on writing the functions in the util file that were used to create the queries and run the code in the main notebook files. For the visualizations, I worked on the code in the step 4 jupyter notebook to create all of the graphs using the discussion 6 code as a reference, and adapted it to match the criteria for the new graphs we have to make.
Siya: I helped with writing the code in the main jupyter notebooks that created the 3 output json files for steps 3a, 3b, 3c. I helped with coming up with how to execute the different types of functions we needed for each step. I also helped with figuring out ways to split the data based on what type of graphs we wanted to visualize the output from each step.
Tisha: For this assignment, I worked on ensuring everyone's code was loaded in correctly. I ran into some difficulty with this so I attended office hours with the professor in order to understand the data better and therefore be able to interpret the data better with my team. For the coding section of this assignment we coded it all on one computer working together at the library as well as the MU. We effectively set up times to meet to do this. Siya and Ruhi used Jupyter Notebook while Ankita and I used VScode to do this assignment. Lastly I worked alongside Ankita to write the final report. We worked together to format everything and answer questions for Step 5.
Ankita: For this assignment, I worked on the planning aspects in the beginning and helping teammates get the data loaded in correctly. I zoomed a few times with the professor in order to understand the data loading process better in order to give that information to my team. I worked with my teammates to go through the prog-assignment 2 notebook to understand what was going on so we could get started on coding the assignment. I helped brainstorm for coding 3a and 3b. Because we coded on one computer, I was a part of the process of coding even though it was not

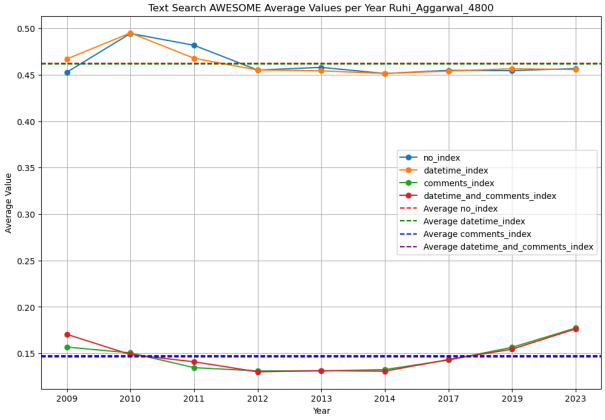
typed on my computer. We did this so we could develop our code together and then run it on our own machines using different softwares (I used vscode while some others used jupyter in the browser). Additionally, in the beginning, I tested the code on lower counts and to help us understand the long run times in order to optimize the code and my teammates were able to get us to 50 counts. I also aided in the visualization process when it was necessary. Additionally, I wrote around half of the report summarizing all of our team results and findings. Tisha and I wrote Step 5 and I used my numbers from my JSON files in order for me to solve question 4 in Step 5.

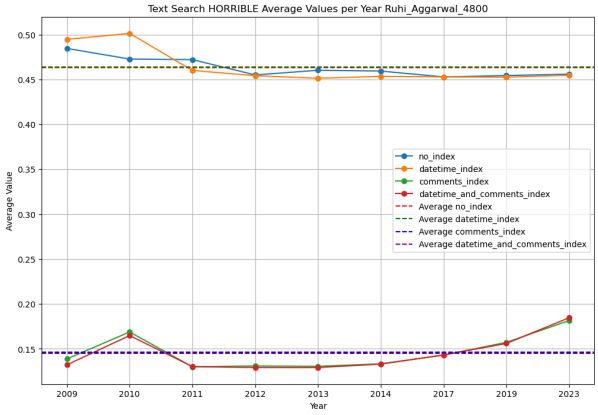
Visualizations:

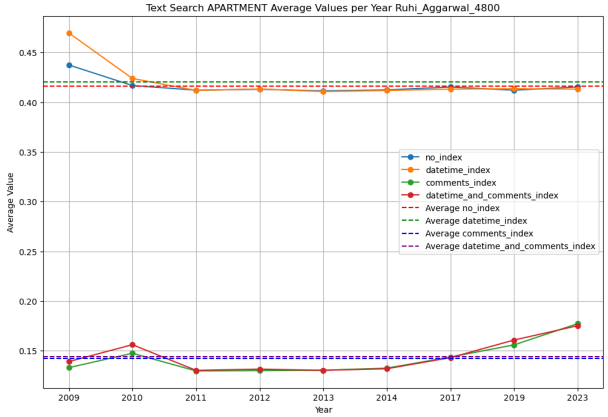
Ruhi:

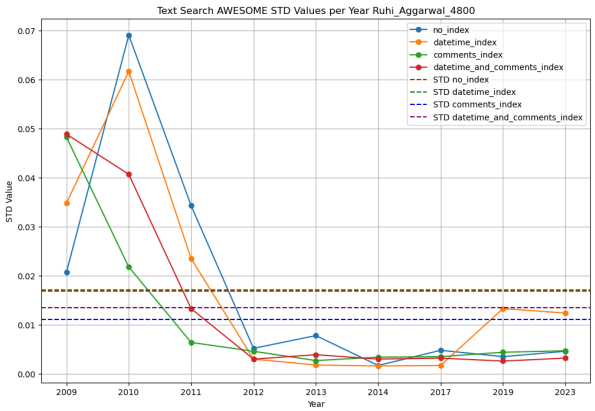


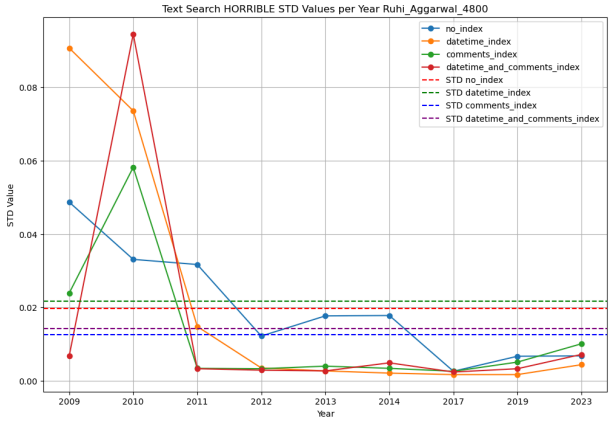


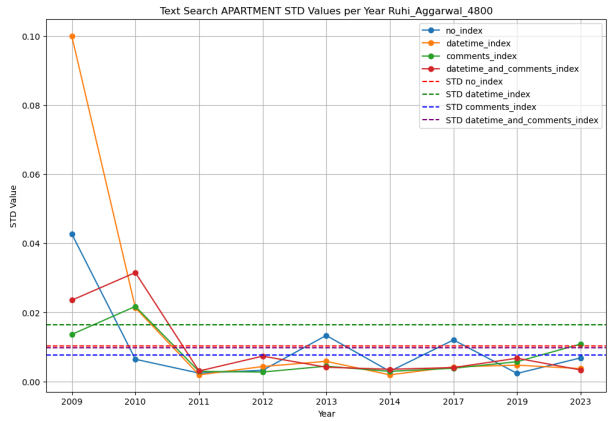


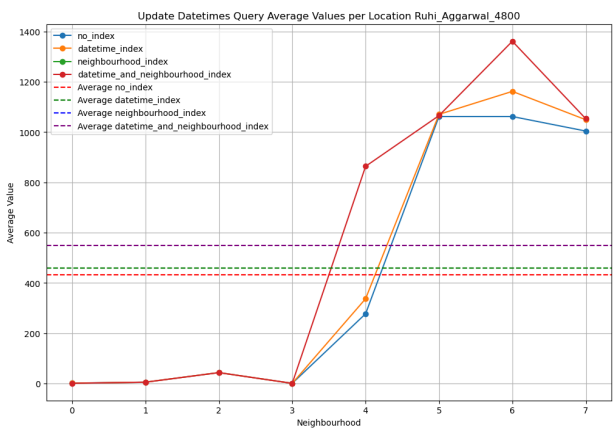


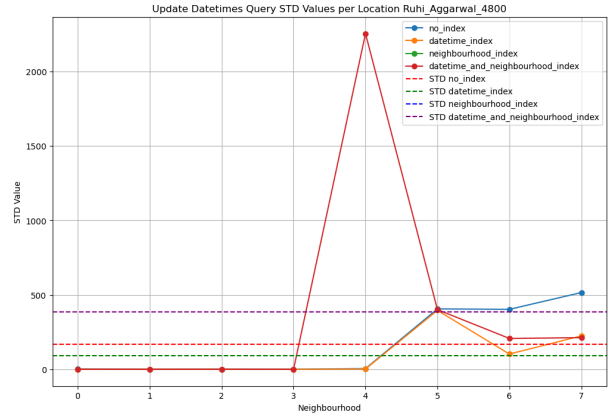

Text Search APARTMENT Average Values per Year Ruhi_Aggarwal_4800
Average Value
Year
no_index
datetime_index
comments_index
datetime_and_comments_index
Average no_index
Average datetime_index
Average comments_index
Average datetime_and_comments_index


Text Search AWESOME STD Values per Year Ruhi_Aggarwal_4800
STD Value
Year
no_index
datetime_index
comments_index
datetime_and_comments_index
STD no_index
STD datetime_index
STD comments_index
STD datetime_and_comments_index

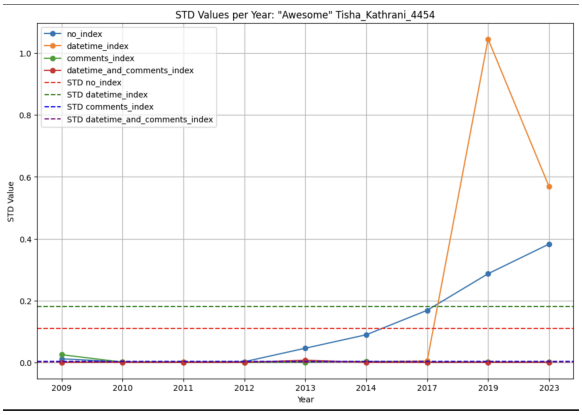

Text Search HORRIBLE STD Values per Year Ruhi_Aggarwal_4800
STD Value
Year
no_index
datetime_index
comments_index
datetime_and_comments_index
STD no_index
STD datetime_index
STD comments_index
STD datetime_and_comments_index

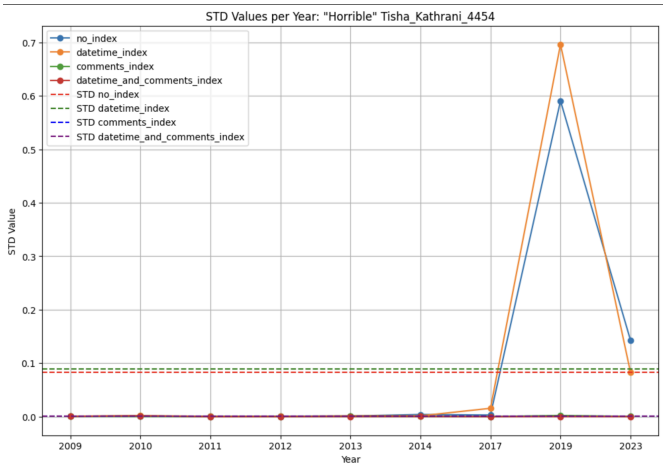

Text Search APARTMENT STD Values per Year Ruhi_Aggarwal_4800
STD Value
Year
no_index
datetime_index
comments_index
datetime_and_comments_index
STD no_index
STD datetime_index
STD comments_index
STD datetime_and_comments_index

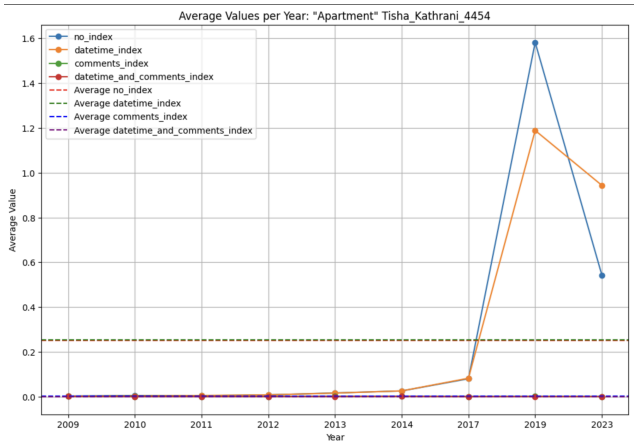

Update Datetimes Query Average Values per Location Ruhi_Aggarwal_4800
Average Value
Neighbourhood
no_index
datetime_index
neighbourhood_index
datetime_and_neighbourhood_index
Average no_index
Average datetime_index
Average neighbourhood_index
Average datetime_and_neighbourhood_index

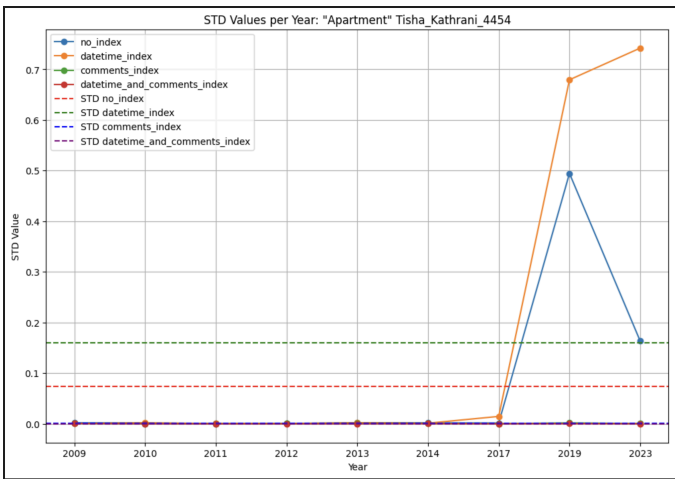

Update Datetimes Query STD Values per Location Ruhi_Aggarwal_4800
STD Value
Neighbourhood
no_index
datetime_index
neighbourhood_index
datetime_and_neighbourhood_index
STD no_index
STD datetime_index
STD neighbourhood_index
STD datetime_and_neighbourhood_index

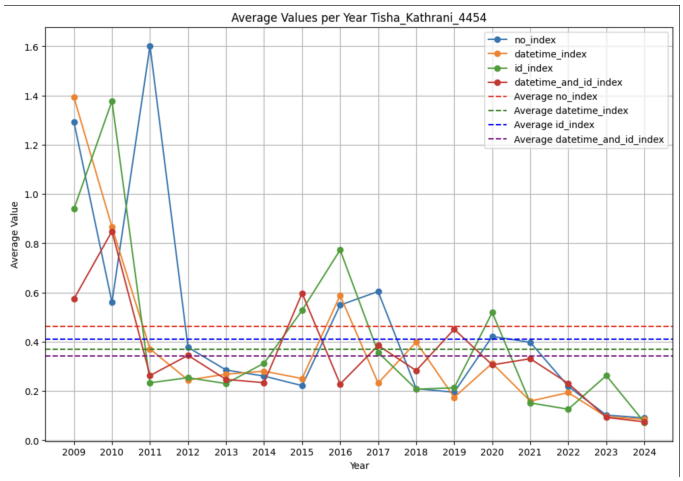
Tisha:

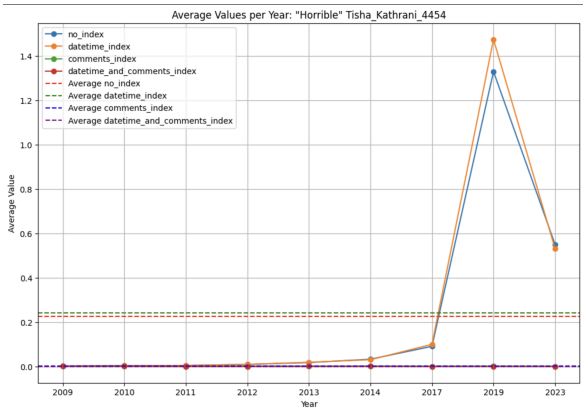

STD Values per Year: "Awesome" Tisha_Kathrani_4454
no_index
datetime_index
comments_index
datetime_and_comments_index
STD no_index
STD datetime_index
STD comments_index
STD datetime_and_comments_index
STD Value
Year

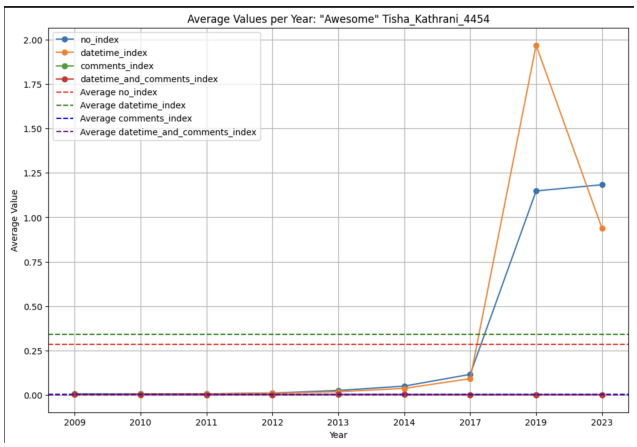

STD Values per Year: "Horrible" Tisha_Kathrani_4454
no_index
datetime_index
comments_index
datetime_and_comments_index
STD no_index
STD datetime_index
STD comments_index
STD datetime_and_comments_index
STD Value
Year

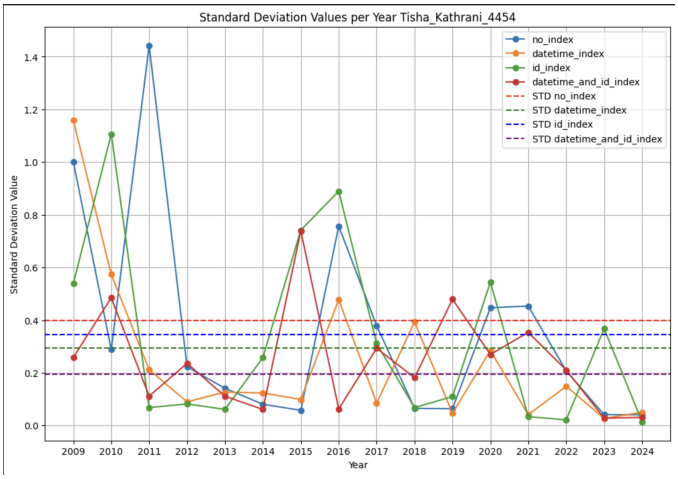

Average Values per Year: "Apartment" Tisha_Kathrani_4454
no_index
datetime_index
comments_index
datetime_and_comments_index
Average no_index
Average datetime_index
Average comments_index
Average datetime_and_comments_index
Average Value
Year

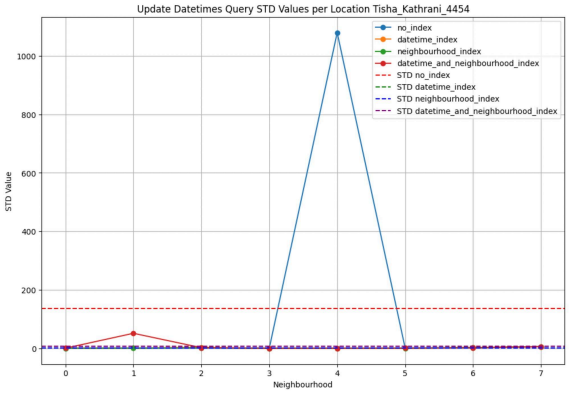

STD Values per Year: "Apartment" Tisha_Kathrani_4454
no_index
datetime_index
comments_index
datetime_and_comments_index
STD no_index
STD datetime_index
STD comments_index
STD datetime_and_comments_index
STD Value
Year

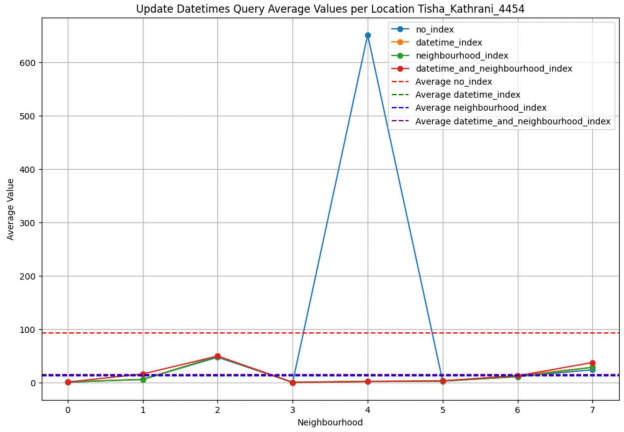

Average Values per Year Tisha_Kathrani_4454
no_index
datetime_index
id_index
datetime_and_id_index
Average no_index
Average datetime_index
Average id_index
Average datetime_and_id_index
Average Value
Year


Average Values per Year: "Horrible" Tisha_Kathrani_4454
no_index
datetime_index
comments_index
datetime_and_comments_index
Average no_index
Average datetime_index
Average comments_index
Average datetime_and_comments_index
Average Value
Year

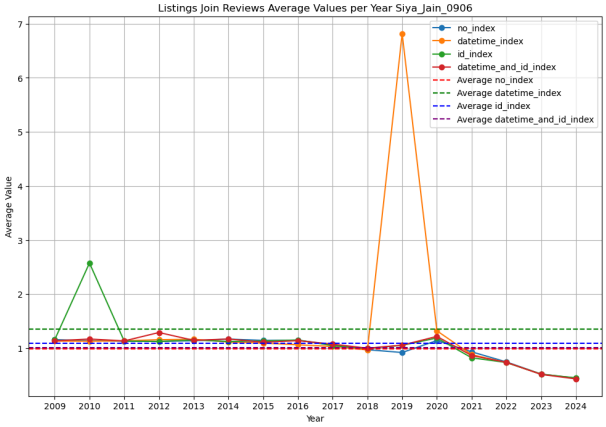

Average Values per Year: "Awesome" Tisha_Kathrani_4454
no_index
datetime_index
comments_index
datetime_and_comments_index
Average no_index
Average datetime_index
Average comments_index
Average datetime_and_comments_index
Average Value
Year

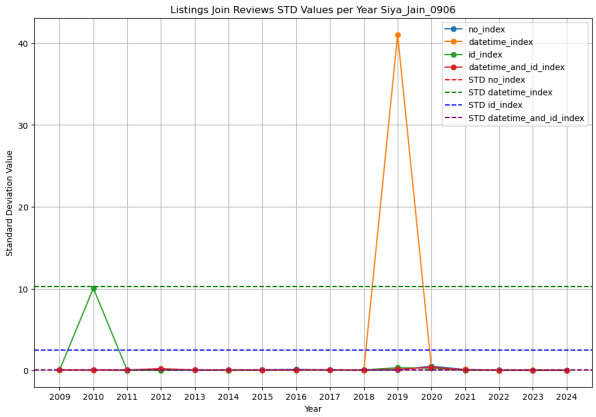

Standard Deviation Values per Year Tisha_Kathrani_4454
no_index
datetime_index
id_index
datetime_and_id_index
STD no_index
STD datetime_index
STD id_index
STD datetime_and_id_index
Standard Deviation Value
Year

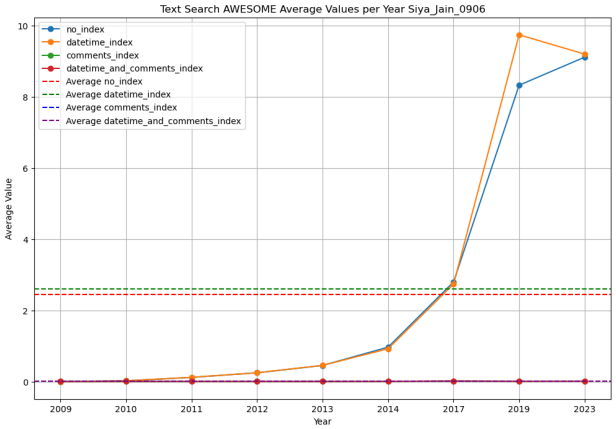

Update Datetimes Query STD Values per Location Tisha_Kathrani_4454
no_index
datetime_index
neighbourhood_index
datetime_and_neighbourhood_index
STD no_index
STD datetime_index
STD neighbourhood_index
STD datetime_and_neighbourhood_index
STD Value
Neighbourhood

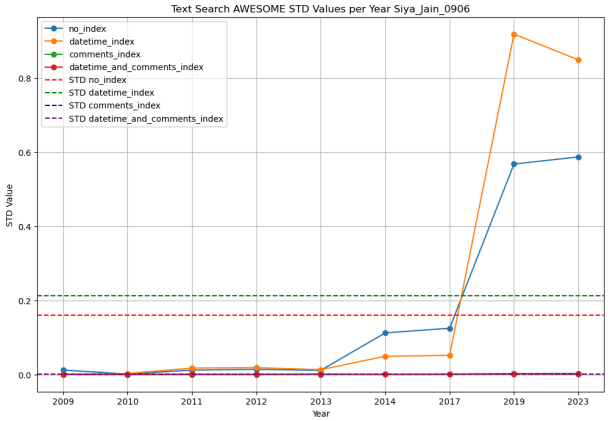

Update Datetimes Query Average Values per Location Tisha_Kathrani_4454
no_index
datetime_index
neighbourhood_index
datetime_and_neighbourhood_index
Average no_index
Average datetime_index
Average neighbourhood_index
Average datetime_and_neighbourhood_index
Average Value
Neighbourhood

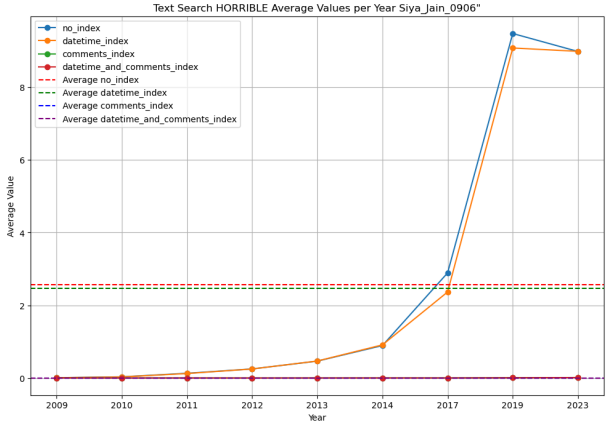
Siya:

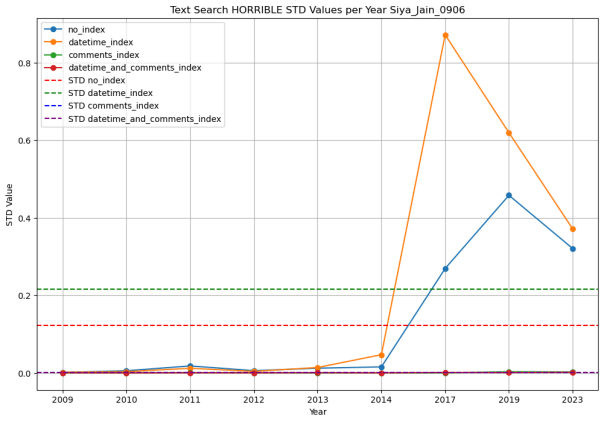

Listings Join Reviews Average Values per Year Siya_Jain_0906
no_index
datetime_index
id_index
datetime_and_id_index
Average no_index
Average datetime_index
Average id_index
Average datetime_and_id_index
Average Value
Year

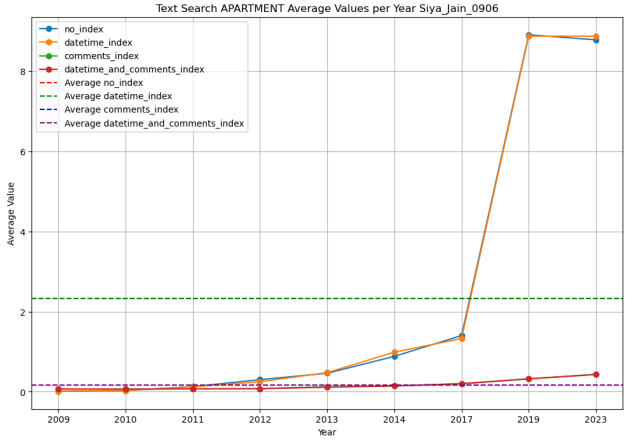

Listings Join Reviews STD Values per Year Siya_Jain_0906
no_index
datetime_index
id_index
datetime_and_id_index
STD no_index
STD datetime_index
STD id_index
STD datetime_and_id_index
Standard Deviation Value
Year

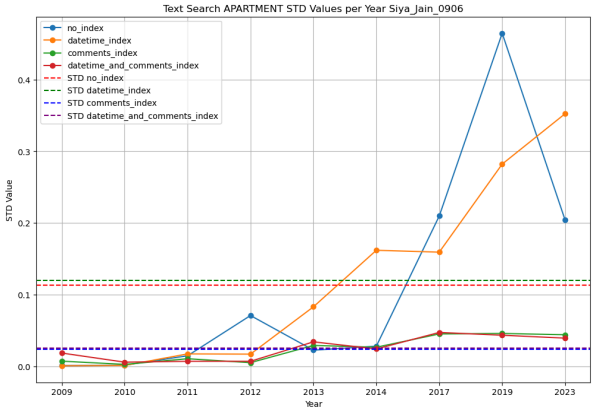

Text Search AWESOME Average Values per Year Siya_Jain_0906
no_index
datetime_index
comments_index
datetime_and_comments_index
Average no_index
Average datetime_index
Average comments_index
Average datetime_and_comments_index
Average Value
Year

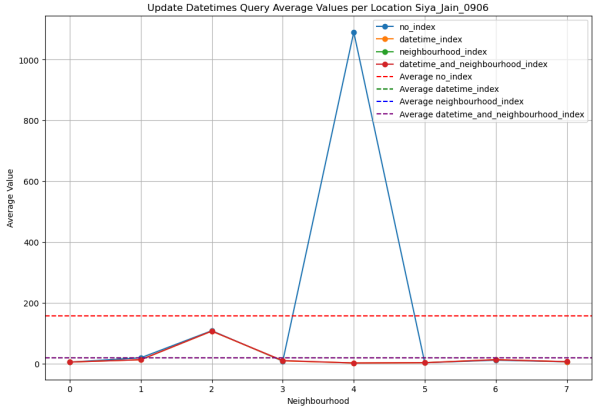

Text Search AWESOME STD Values per Year Siya_Jain_0906
no_index
datetime_index
comments_index
datetime_and_comments_index
STD no_index
STD datetime_index
STD comments_index
STD datetime_and_comments_index
STD Value
Year

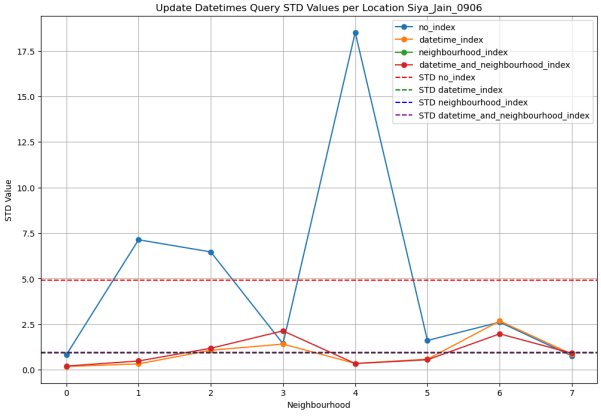

Text Search HORRIBLE Average Values per Year Siya_Jain_0906"
no_index
datetime_index
comments_index
datetime_and_comments_index
Average no_index
Average datetime_index
Average comments_index
Average datetime_and_comments_index
Average Value
Year


Text Search HORRIBLE STD Values per Year Siya_Jain_0906
no_index
datetime_index
comments_index
datetime_and_comments_index
STD no_index
STD datetime_index
STD comments_index
STD datetime_and_comments_index
STD Value
Year

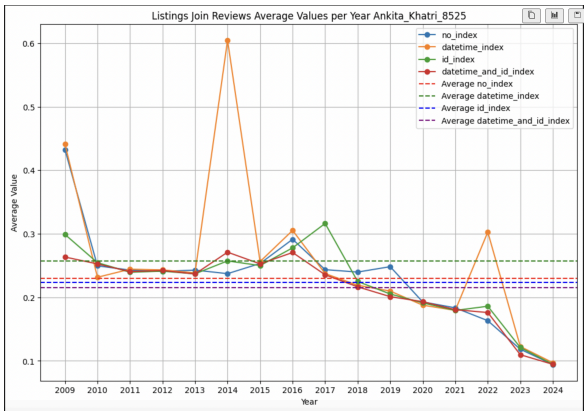

Text Search APARTMENT Average Values per Year Siya_Jain_0906
no_index
datetime_index
comments_index
datetime_and_comments_index
Average no_index
Average datetime_index
Average comments_index
Average datetime_and_comments_index
Average Value
Year

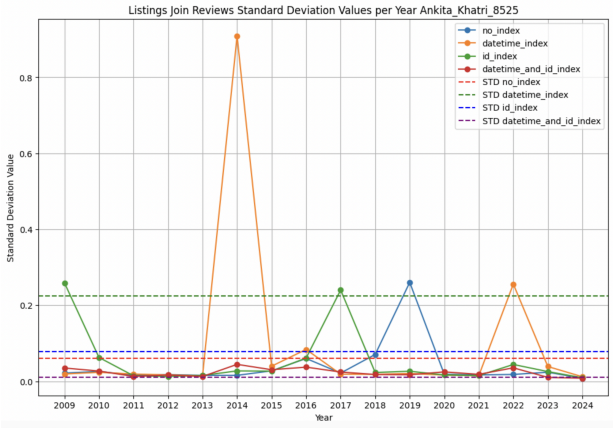

Text Search APARTMENT STD Values per Year Siya_Jain_0906
no_index
datetime_index
comments_index
datetime_and_comments_index
STD no_index
STD datetime_index
STD comments_index
STD datetime_and_comments_index
STD Value
Year

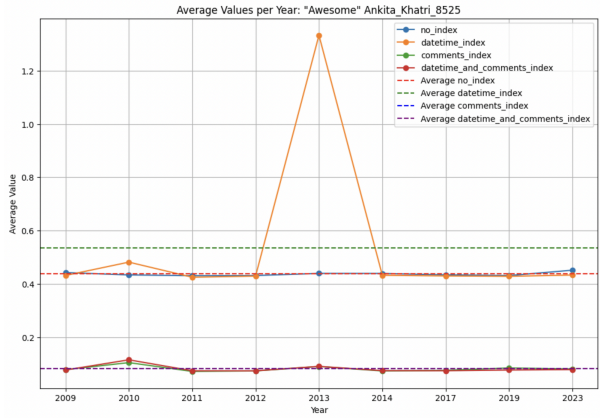

Update Datetimes Query Average Values per Location Siya_Jain_0906
no_index
datetime_index
neighbourhood_index
datetime_and_neighbourhood_index
Average no_index
Average datetime_index
Average neighbourhood_index
Average datetime_and_neighbourhood_index
Average Value
Neighbourhood


Update Datetimes Query STD Values per Location Siya_Jain_0906
no_index
datetime_index
neighbourhood_index
datetime_and_neighbourhood_index
STD no_index
STD datetime_index
STD neighbourhood_index
STD datetime_and_neighbourhood_index
STD Value
Neighbourhood

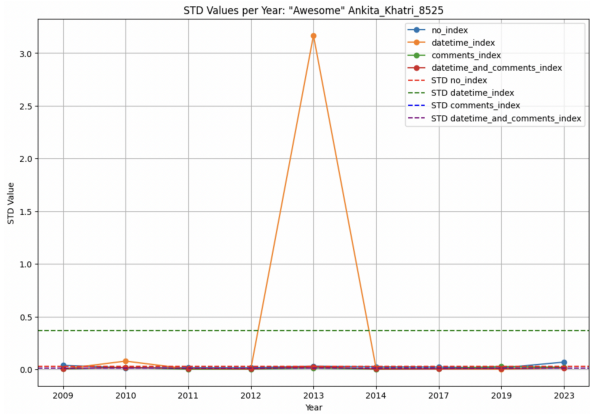
Ankita:

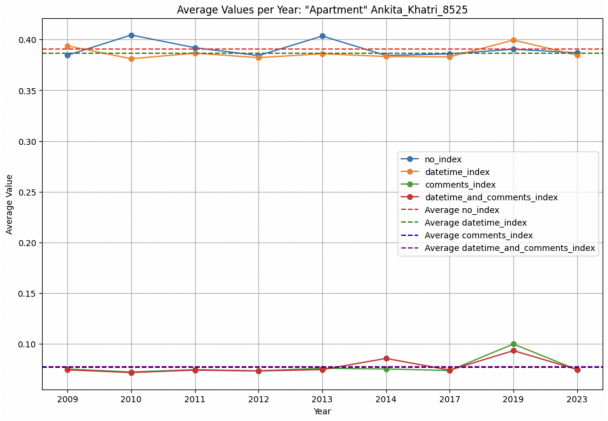

Listings Join Reviews Average Values per Year Ankita_Khatri_8525
no_index
datetime_index
id_index
datetime_and_id_index
Average no_index
Average datetime_index
Average id_index
Average datetime_and_id_index
Average Value
Year

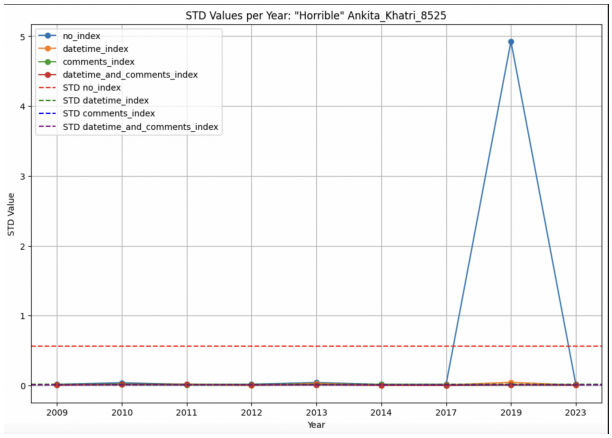

Listings Join Reviews Standard Deviation Values per Year Ankita_Khatri_8525
no_index
datetime_index
id_index
datetime_and_id_index
STD no_index
STD datetime_index
STD id_index
STD datetime_and_id_index
Standard Deviation Value
Year

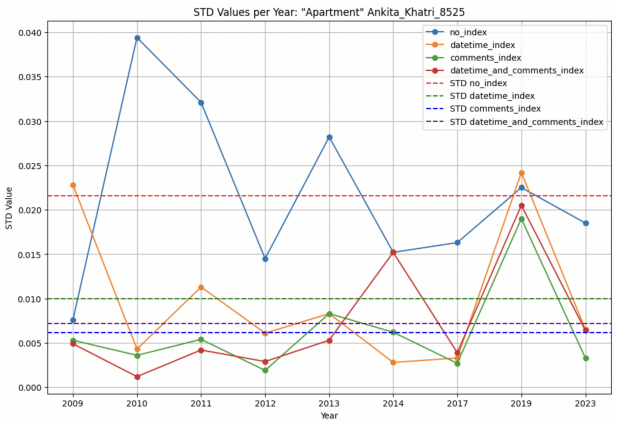

Average Values per Year: "Awesome" Ankita_Khatri_8525
no_index
datetime_index
comments_index
datetime_and_comments_index
Average no_index
Average datetime_index
Average comments_index
Average datetime_and_comments_index
Average Value
Year

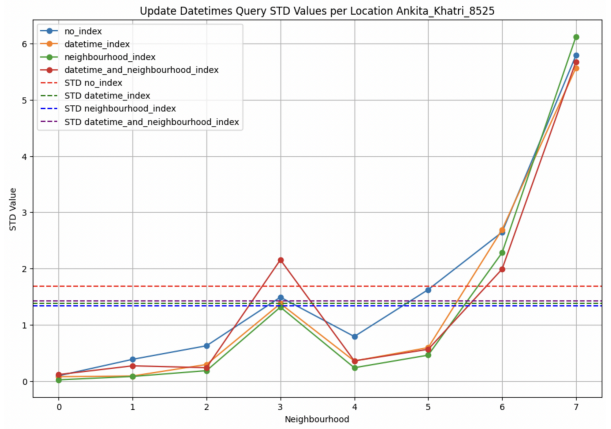

Average Values per Year: "Horrible" Ankita_Khatri_8525
no_index
datetime_index
comments_index
datetime_and_comments_index
Average no_index
Average datetime_index
Average comments_index
Average datetime_and_comments_index
Average Value
Year

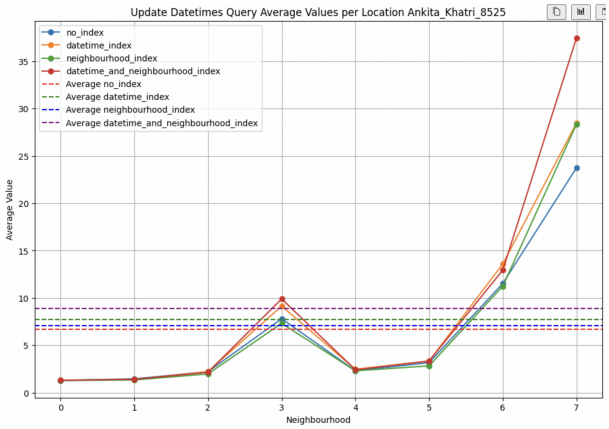

STD Values per Year: "Awesome" Ankita_Khatri_8525
no_index
datetime_index
comments_index
datetime_and_comments_index
STD no_index
STD datetime_index
STD comments_index
STD datetime_and_comments_index
STD Value
Year


Average Values per Year: "Apartment" Ankita_Khatri_8525
no_index
datetime_index
comments_index
datetime_and_comments_index
Average no_index
Average datetime_index
Average comments_index
Average datetime_and_comments_index
Average Value
Year







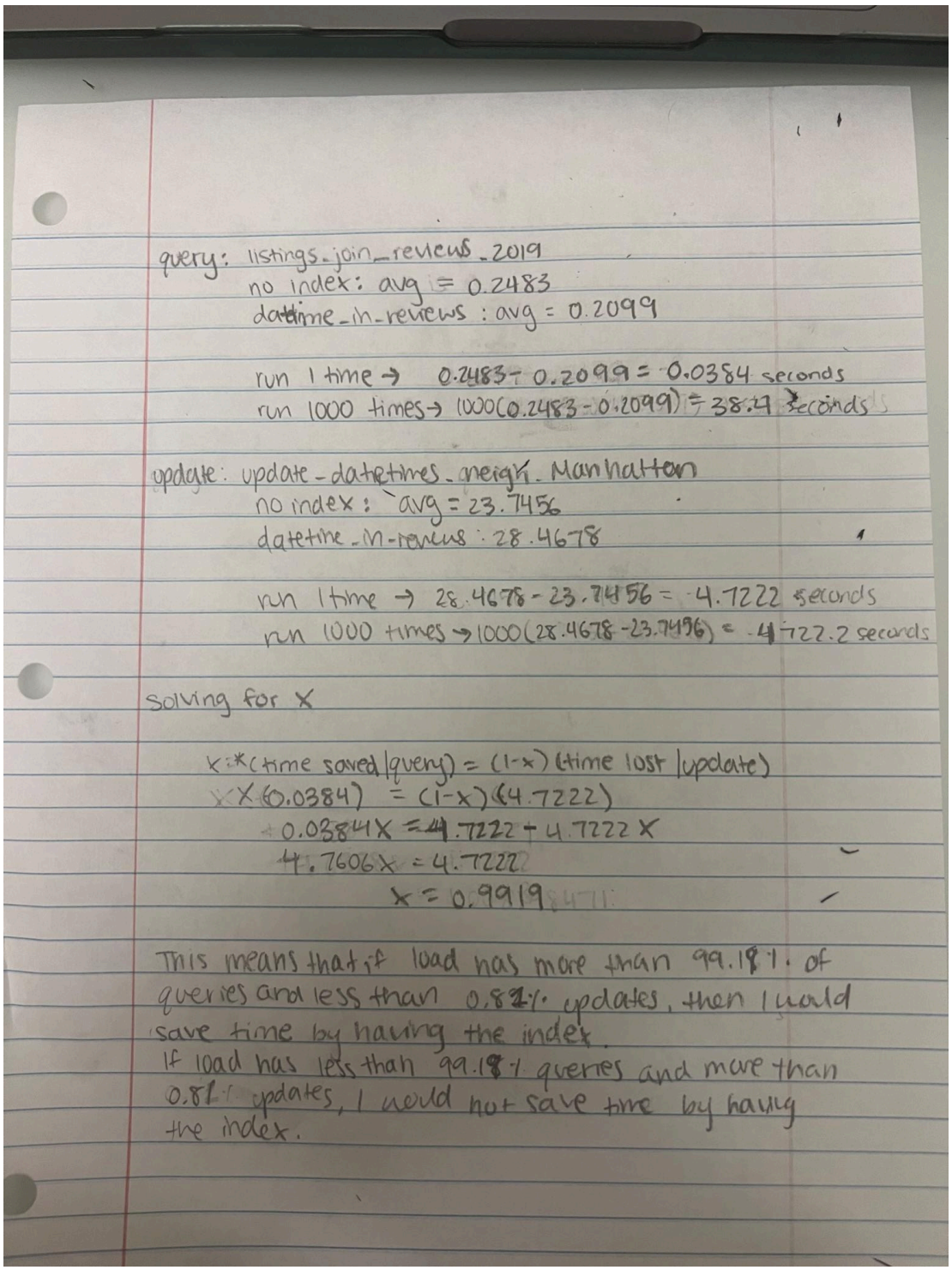


Explanations of Performance Results:

1. The most effective index for Step 3a would be __datetime_in_reviews__. This is because this specific index consistently showed the lowest average execution time across different years, indicating its efficiency in improving query performance when joining listings with reviews based on the year. The lower execution time is due to the index providing a faster lookup of datetime values, which is crucial for join operations that filter by specific time ranges.
2. It is significantly more effective to use indexes when the dataset contains a large volume of text data, and the queries involve searching through text fields. For example, the __comments_tsv_in_reviews__ index greatly improved query performance for text searches within reviews. The effectiveness is due to the index allowing quick text lookups and reducing the need for a full table scan, which is computationally expensive and time-consuming.
3. Yes, there is an expense to having the datetime_in_reviews index when updating the reviews table. The update operations take longer because the index needs to be maintained and updated alongside the table. The expense varies based on the complexity and the number of rows affected by the update. For instance, updates affecting a larger number of rows or involving additional indexed columns result in higher overhead and

longer execution times, as observed in different neighborhoods such as Fort Hamilton and Bedford-Stuyvesant.

4. For this question, we used the numbers from Ankita's JSON files. Here is the arithmetic:



References

We did not use any outside resources.