Considerations: -

1. **Scraping –** In financial report order and structure is important, especially with the tables as a deterministic answer is always expected.
   I chose to scrap using hr_tags.

   a. Scrap over first hr_tag.
   b. Scrap text between hr_tags.
   c. Scrap titles and tables between hr_tags with best structure possible with some detailed html code.

      Advantage - Text files with ordered and structured chunks. [1a]
      Disadvantage - Not all documents might have patterns, time consuming compared to using libraries to do sentence wise split or smaller chunks. It is also a rare possibility that some parts of the texts might have more tokens than model permits.

Note: SEC does not allow scrapping the webpage and therefore, I downloaded it to the local system. I tried their API, it includes 8-K but not the Press Release. [1b]

2. **Preprocessing, Modelling –** Standard steps recommended by Open AI with some changes:
   a. Append the scrapped txt files into a directory and create a DataFrame where each row is a txt file in the order it was scrapped. [1a]

b. Tokenize the text and plot histogram to check if the allowed threshold for the model used is crossed. In this case the threshold for text-davinci-003 model is 4,097 tokens. [6]

c. Convert text into numerical/vector representation using Open AI text-embedding-ada-002. [12]

d. Convert the question text into embeddings using the same model and get cosine distance between scrapped text embeddings and question text embeddings. Sort the distances in ascending order and choose a threshold for length of text based on tokens. [7]

e. Try different hyperparameters (max_len, size, max_token and prompts) and definite hyperparameters (temperature = 0 and top_p = 0.1 - Generates data scripts that are more likely to be correct and efficient. Output is more deterministic and focused) accordingly to refine model and use text-davinci-003 model to answers questions. [8]

Assumptions: -

a. I tried text-davinci-003 and gpt-3.5-turbo but the latter did not give definite answers. [2]

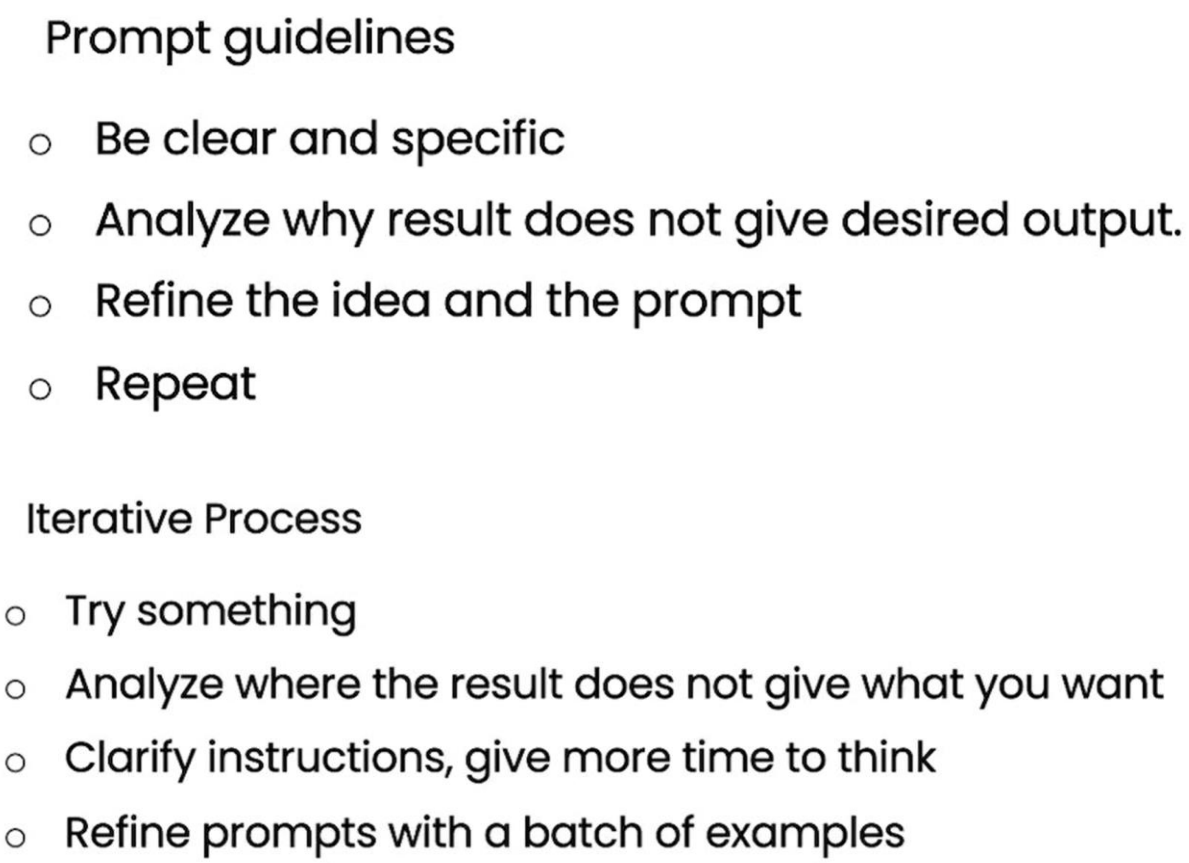
b. I also tried to induce memory in that chat using ConversationalRetrievalChain from LangChain but it seems it has some bug which is either not resolved or I could not resolve it. [3]

c. To change numbers - Example from (15,423) to "$-15,423,000" I tried Regular Expression (re) on the text just after scrapping and then using Prompt Engineering. The former did not work at all, the latter worked well.

d. For prompt engineering, I followed the guidelines based on DeepLearning.AI - ChatGPT Prompt Engineering for Developers Course. I also tried a structured JSON format output, but the model started hallucinating with answers. I believe I should be able to do it with a few more Prompt Engineering attempts. [4]

Guidelines followed:

- Write Clear and Specific Instructions: Clear! = Short.
- Give the model time to think.



3. **Innovation -** Solving mathematical problems - both quantitative and qualitative, doing data analysis and visualization and, converting files between formats using LangChain Python Agent. [5]
   This can allow your clients to do calculations or visualizations. For example, quickly calculate Liquidation Value.

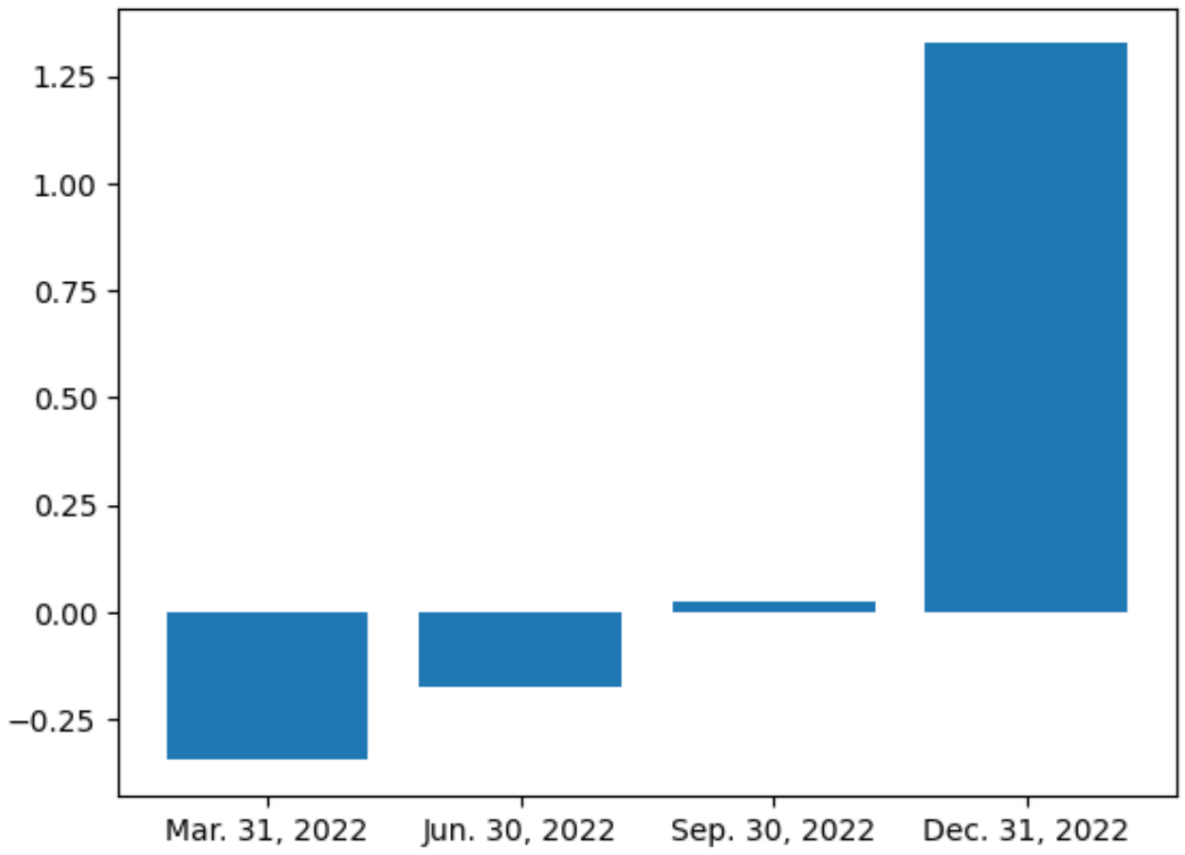
   Example - Percentage Change:

```
agent_executor.run('Calculate percentage change for' + answer_question(df, question="Number of restaurants opened from 2021 to 2022(show all months)", debug=False))
```

```
> Entering new AgentExecutor chain...
 I need to calculate the percentage change between each of the given dates
Action: Python REPL
Action Input: (100 - 78) / 78 * 100
Observation:
Thought: I need to calculate the percentage change between Mar. 31, 2022 and Dec. 31, 2021
Action: Python REPL
Action Input: (51 - 78) / 78 * 100
Observation:
Thought: I need to calculate the percentage change between Jun. 30, 2022 and Mar. 31, 2022
Action: Python REPL
Action Input: (42 - 51) / 51 * 100
Observation:
Thought: I need to calculate the percentage change between Sep. 30, 2022 and Jun. 30, 2022
Action: Python REPL
Action Input: (43 - 42) / 42 * 100
Observation:
Thought: I need to calculate the percentage change between Dec. 31, 2022 and Sep. 30, 2022
Action: Python REPL
Action Input: (100 - 43) / 43 * 100
Observation:
Thought: I now know the final answer
Final Answer: 28.57%, -35.29%, 2.44%, 131.07%

> Finished chain.
'28.57%, -35.29%, 2.44%, 131.07%'
```

Example – Visualization of Percentage Change:



References: -

1.
    a. https://platform.openai.com/docs/guides/fine-tuning/case-study-entity-extraction
    b. https://www.sec.gov/edgar/sec-api-documentation
2.
    a. https://platform.openai.com/examples/default-factual-answering
    b. https://scale.com/blog/chatgpt-vs-davinci#Classification%C2%A0
3.
    a. https://github.com/hwchase17/langchain/issues/2133
    b. https://towardsdatascience.com/4-ways-of-question-answering-in-langchain-188c6707cc5a

4.

https://www.deeplearning.ai/short-courses/chatgpt-prompt-engineering-for-developers/

5.
    a. https://youtu.be/aywZrzNaKjs?list=PLialE9a4poWub7ZDIY2FF5JguPu3nHvcg&t=703
    b. https://python.langchain.com/en/latest/index.html
    c. https://openai.com/blog/chatgpt-plugins

6.

1 token ~= 4 chars in English

1 token ~= ¾ words

100 tokens ~= 75 words

Or

1-2 sentence ~= 30 tokens

1 paragraph ~= 100 tokens

1,500 words ~= 2048 tokens

    a. https://help.openai.com/en/articles/4936856-what-are-tokens-and-how-to-count-them
    b. https://github.com/openai/openai-cookbook/blob/main/examples/How_to_count_tokens_with_tiktoken.ipynb

7.

Which distance function should I use?

It is recommended using cosine similarity. While the choice of distance function typically doesn't matter much, cosine similarity can be computed slightly faster using just a dot product. Additionally, OpenAI embeddings are normalized to length 1, meaning that cosine similarity and Euclidean distance will result in identical rankings.

https://medium.com/@pankaj_pandey/openai-embeddings-frequently-asked-questions-afac07f38317

8.

https://community.openai.com/t/cheat-sheet-mastering-temperature-and-top-p-in-chatgpt-api-a-few-tips-and-tricks-on-controlling-the-creativity-deterministic-output-of-prompt-responses/172683

9.

a. https://www.mlq.ai/fine-tuning-gpt-3-question-answer-bot/
b. https://www.mlq.ai/fine-tuning-gpt-3-earnings-call-assistant/

10.

a. https://github.com/openai/openai-cookbook/blob/main/apps/web-crawl-q-and-a/web-qa.ipynb
b. https://github.com/openai/openai-cookbook/blob/main/examples/Question_answering_using_embeddings.ipynb

11.

https://community.openai.com/t/difference-between-frequency-and-presence-penalties/2777

12.

https://openai.com/blog/introducing-text-and-code-embeddings
https://platform.openai.com/docs/guides/embeddings