







☆ 0 stars 🍴 0 forks 👁 1 watching 🌿 1 Branch 🏷 0 Tags 📈 Activity 📄 Custom properties


🌐 Public repository


 main ▾


 1 Branch

 0 Tags





 Go to file

 t

Go to file

+

Add file ▾

Code

...

 avyfain	reduce dataset	last month	...	🕒
 .gitignore	initial commit	2 months ago		
 README.md	reduce dataset	last month		
 SP500.txt	reduce dataset	last month		

📖 README



Overview

In this assignment you will build a prototype of a cluster analysis tool to navigate financial statements.

Each company has a unique CIK (Central Index Key) that is used to identify it in the data. The CIK is a 10 digit number, and is the prefix of the file name for each company's 10-K report.

There are three main tasks in this assignment:

- To construct a CIK -> Company Name mapping from the provided data.
- To summarize each company's report into a few sentences.
- To cluster the companies into similar groups based on their financial statements.

The goal of this assignment is to demonstrate your ability to build a data pipeline to process unstructured data, and to use that data to build a simple clustering and summarizing tool whose output could be built into a more complex application. What we expect you to build are proofs of concept, and not production-ready models.

If you decide to use a paid API to solve the exercise, we will reimburse you for usage up to \$10.

Instructions

1. Clone (**please, don't fork!**) this repository and create a new branch for your development work
2. Create your implementation following the [Specification](#) below
3. Add instructions on how to run your implementation to the [Getting Started](#) section.
4. In the [follow-up questions](#) section below, respond inline to each of the questions.
5. Commit your implementation and answers to a new repo in your personal GH and give @avyfain access to the repo.

Guidelines:

- Do not spend longer than four hours on your implementation, a perfect implementation is not required or expected. Please move on to the [follow-up questions](#) after that.
- You may use any language or tools you like to complete this assignment. You can use any libraries or frameworks you like, including any existing clustering libraries. You can use any pre-trained language models you like.
- Ask questions if you have them. The business problem is intentionally vague, so you will need to make some assumptions. Document your assumptions in your code and in the follow-up questions.
- It's fine to use Google or StackOverflow to look up syntax or documentation. If you use ChatGPT or similar tools, please share the prompts you used in the follow-up questions.

Exercise Data

You can find a zip file with the required data in [this HuggingFace repo](#).

In the provided data zip file, you will find over 3000+ recent 10-K reports from publicly traded companies. These reports are HTML containing the financial statements for each company.

If you have a CIK, you can use it to access the corresponding company's data in the SEC's EDGAR database. For example, the CIK for Apple Inc. is 0000320193. You can find Apple's reports here:

<https://www.sec.gov/edgar/browse/?CIK=000320193>.

We do not expect you to download any additional data from the SEC's database, but you can find the full documentation for the EDGAR database here: <https://www.sec.gov/edgar/searchedgar/accessing-edgar-data.htm>

Specification

We expect you to build the following functionality:

- ☐ You will filter down the dataset to cluster companies that are in the S&P 500 index. You can find a recent list of CIKs for companies in the S&P 500 in the `SP500.txt` file.
- ☐ You will create a script that given a directory with report files can produce a `CIK -> Company Name` mapping in the shape of a CSV file with two columns: CIK and Company Name. Each row in this file will represent each file in the provided data. (hint: you don't need to throw an LLM at this problem)
- ☐ You will run your mapping script on the provided data, and include it in your response.
- ☐ You will write a data pipeline to process the provided HTML into an intermediate representation that can be used for clustering. One of the features in your intermediate representation should be a 1-paragraph summary of the report. You can use any pre-trained language model you like to generate the summary.
- ☐ You will use your pipeline to assign every company in the dataset into similar groups based on their financial statements.
- ☐ You will provide a Jupyter Notebook, a Streamlit app, or equivalent for users to inspect and interact with the results of your clustering and summarization. The visualization should allow the user to select a company and show other similar companies in the same cluster.

Getting Started

This is a placeholder for instructions on how to run your implementation.

Follow-Up Questions

1. Describe which task you found most difficult in the implementation, and why.
2. What led you to choose the libraries or frameworks you used in your implementation?
3. How did you evaluate whether the clusters and summaries created by your system were good or not?
4. If there were no time or budget restrictions for this exercise, what improvements would you make in the following areas:
 - Implementation
 - User Experience
 - Data Quality
5. If you built this using classic ML models, how would approach it if you had to build it with LLMs? Similarly, if you used LLMs, what are some things you would try if you had to build it with classic ML models?
6. If you had to build this as part of a production system, providing an inference API for previously unseen reports, how would you do it? What would you change in your implementation?

Evaluation Criteria

You will be evaluated out of a total of 50 points based on the following criteria.

- Learning Exercise (30 points total)
 - **Functionality (20 points):** is the requested functionality implemented as described?
 - **Code Quality (10 points):** is the code well structured and easily read?
 - **Bonus (3 maximum):** bonus points are awarded for anything that goes above and beyond the items in the specification. For example, additional .
- Follow Up Questions (20 points total)
 - Question 1 (2 points)
 - Question 2 (2 points)
 - Question 3 (3 points)
 - Question 4 (3 points)
 - Question 5 (5 points)

Releases

No releases published

Packages

No packages published