

An aerial photograph of New York City, showing the Manhattan Bridge spanning the Hudson River, the dense urban landscape of Manhattan, and Central Park in the foreground. The title 'An Analysis of Nothing' is overlaid on a yellow oval in the center.

An Analysis of Nothing

Chandler Ault, Yamina Katariya, Yash Manne, Aditi Shrivastava



Background



What is Seinfeld:

- American sitcom that features four friends, Jerry Seinfeld, George Costanza, Elaine Benes, and Cosmo Kramer played by real-life comedians

Problem:

- Difficult for Seinfeld fans to search and discover certain episode and series characteristics when their only resource is manual web-searching

Solution:

- Interactive web page that covers the extensive capabilities of Wikipedia, IMDb, and Netflix all in one!

Data Used



Source	Collection	Description	Limitations
Seinfeld Chronicles (Kaggle)	Downloaded	Complete scripts for all 173 episodes	Doesn't adhere to Netflix/IMDb indexing, messy fields
IMDb Datasets	Downloaded	Episode production information and IMDb scores	Scores may be biased toward either ends.
IMDb Episode Pages	Scraped	Description, summaries, keywords	Time consuming. Truth is assumed

Use Cases



Episode Recommendation

1. Website:

Prompts user to enter number of desired recs and favorite episode titles

2. User:

Enters prompts

3. Website:

Returns n episodes with highest similarity

Episode Querying

1. User:

Enters general events/occurrences into search bar and selects desired filters

2. Website:

Displays a list of episodes that satisfy the query

3. User:

Clicks on an episode to view episode sentiment analytics

General Analysis

1. User:

Selects seasons of interest

2. Website:

Displays general analytics like number of lines spoken by main four and episode rating, by season

3. User:

Interacts with graphs (ex. zoom)

Tool Design



Data Manager

Data Cleaning + Pre-processing

- Cleaned all data to account for merging between IMDb and Kaggle data sources
- Precompute emotional sentiment for each dialogue
- Precompute feature embeddings from BERT on all lines of dialogue.



Data Manager Utility Tools

- Loads cleaned scripts & metadata and hands it other functions.

Episode Recommender

Episode Recommender Utility Tools

- Generate vector encodings for all episodes based on dialogue, key words, summaries, ratings, sentiment, and # of lines per character
 - Uses pretrained BERT embeddings for text
- Computes pairwise cosine similarities between input episode(s) and all other episodes, and returns top n episodes with the highest average cosine similarity score



Episode Query Web Page

- Display favorite bar for user input
- Display output table with recommended episodes and metadata: Episode ID, Rating, Title, Writers, & Directors
- Users click on any episode:
 - Can view it's IMDb episode description

Episode Query

Episode Query Utility Tools

- Generate vector encodings for all dialogue + input query, find closest matches via cosine similarity
- Create clickable AgGrid data frame from output
- Filter response by sidebar option (Season, Rating, Speaking Characters)
- Format data for interactive sentiment analysis visualizations



Episode Query Web Page

- Display search bar for user input, display sidebars for filtering
- Users click on any episode:
 - View it's IMDb episode description
 - Bar chart and sunburst plot depicting average episode sentiments, and average characters' sentiments

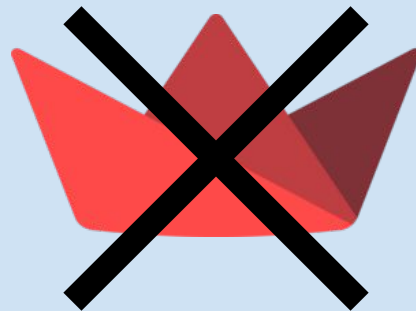
Lessons Learned and Future Work

Lessons Learned:

- Simplicity is not always the best option
- Planning and finalizing structure early on saves time

Future Work:

- Improve recommender algorithm and add additional filter fields
- Add specific dialog search to Episode Query
- Improve sentiment analysis



Website Demo: Local and Hosted

The screenshot shows a VS Code editor with a project named 'an analysis of nothing'. The Explorer sidebar on the left shows the project structure, including files like 'app.py', 'app_pages', and 'static'. The main editor window displays the 'app.py' file, which is a Python script using Streamlit to create a web application. The script imports 'os' and 'streamlit', sets the current working directory to the parent directory, and imports various functions from 'app_pages'. It then uses 'st.markdown' to display a title and a background image. The bottom panel shows the terminal output, which includes the command to run the application and the resulting URL: 'http://localhost:8501'.

```
File Edit Selection View Go Run Terminal Help
an analysis of nothing (WSL: Ubuntu)

EXPLORER
an analysis of nothing
  > .github
  > an_analysis_of_...
  > app
  > app_pages
  > _pycache_
  > _rnt_py
  > write_about_page.py
  > write_episode_que...
  > write_home_page.py
  > write_recommen...
  > static
  > tests
  > utils
  > app.py
  > README.md
  > requirements.txt
  > data_tools
  > doc
  > examples
  > scripts
  > gitignore
  > environment.yml
  > LICENSE
  > pylintrc
  > pyproject.toml
  > README.md

an analysis of nothing > app.py
1 """
2 Code that manages the theme of the website,
3 page navigation sidebar, and pages.
4 """
5 import os
6
7 import streamlit as st
8 from streamlit_option_menu import option_menu
9
10 from app_pages import write_home_page
11 from app_pages import write_recommender_page
12 from app_pages import write_about_page
13 from app_pages import write_episode_query
14
15 # Set current working directory to .
16 cwd = os.getcwd()
17 base_path = os.path.split("an_analysis_of_nothing")[0]
18 curr_path = base_path + \
19     'an_analysis_of_nothing/an_analysis_of_nothing/'
20 os.chdir(curr_path)
21
22 st.set_page_config(layout="centered")
23
24 # Set color and image theme of website
25 st.markdown("""style
26 {
27     [data-testid="stAppViewContainer"] > .main {
28         background-image: url("https://www.pixalstalk.net/wp-content/uploads/images1/free-download-central-park-backgrounds.jpg");
29     }
30 """)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
python - an analysis of nothing + - - - - - x
# ykatarly@ms-computer:~/an_analysis_of_nothing$ cd an_analysis_of_nothing/
# ykatarly@ms-computer:~/an_analysis_of_nothing/an_analysis_of_nothing$ ll
total 40
drwx-r-x-x 7 ykatarly ykatarly 4096 Mar 13 21:27 ./
drwx-r-x-x 9 ykatarly ykatarly 4096 Mar 13 22:00 ../
-rw-r--r-- 1 ykatarly ykatarly 1868 Mar 4 14:07 README.md
drwx-r-x-x 5 ykatarly ykatarly 4096 Mar 4 14:06 app/
-rw-r--r-- 1 ykatarly ykatarly 2075 Mar 13 22:16 app.py
drwx-r-x-x 3 ykatarly ykatarly 4096 Mar 13 21:27 app_pages/
-rw-r--r-- 1 ykatarly ykatarly 134 Mar 13 21:27 requirements.txt
drwx-r-x-x 4 ykatarly ykatarly 4096 Mar 13 21:27 static/
drwx-r-x-x 2 ykatarly ykatarly 4096 Mar 13 22:00 tests/
drwx-r-x-x 3 ykatarly ykatarly 4096 Mar 13 21:27 utils/
# ykatarly@ms-computer:~/an_analysis_of_nothing/an_analysis_of_nothing$
# ykatarly@ms-computer:~/an_analysis_of_nothing/an_analysis_of_nothing$
# ykatarly@ms-computer:~/an_analysis_of_nothing/an_analysis_of_nothing$ streamlit run app.py

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
```