# RagFeed

Jonathan Sada

## Project Overview

## Objective

Create an RSS reader powered by RAG and GenAI that provides the user:

- An overview of the most relevant topics on the today articles.
- The possibility to ask for specific topics and get and overview and the most relevant related articles.

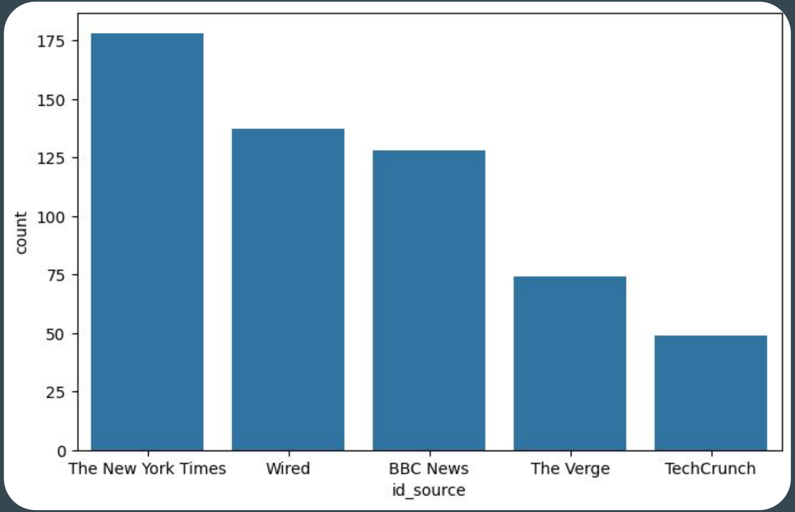
## Why?

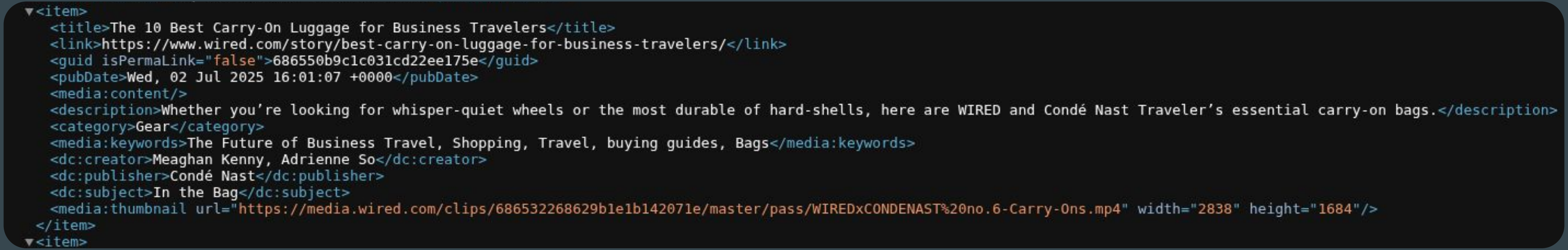
- RSS Sources publish tens of articles each day.
- Some Sources cover many different topics (not always interesting).
- Interesting topics might be lost in between all the daily articles.
- To have "the algorithm" under control.

## Data Source

English RSS found in Best RSS Feeds in 2025:

- The New York Times
- Wired
- BBC News
- The Verge
- TechCrunch
- Ars Technica

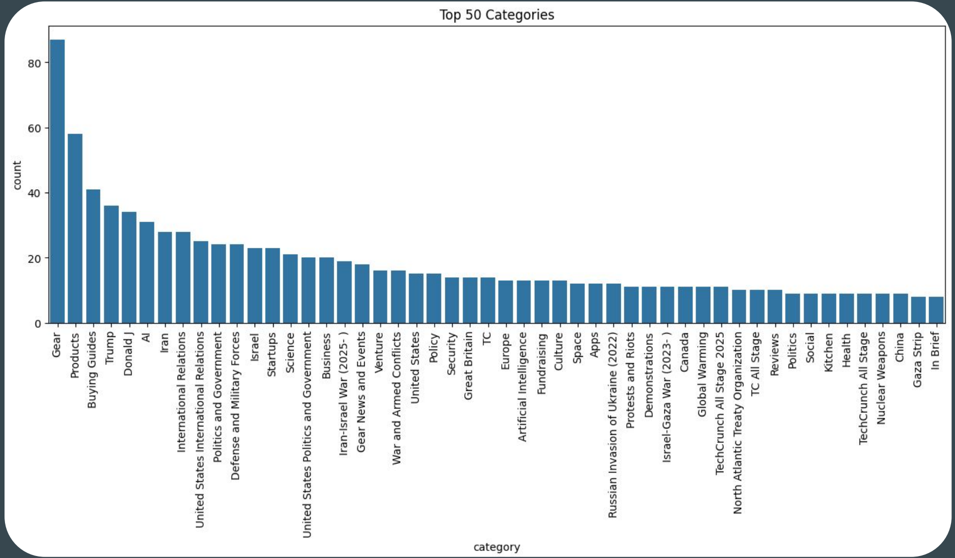


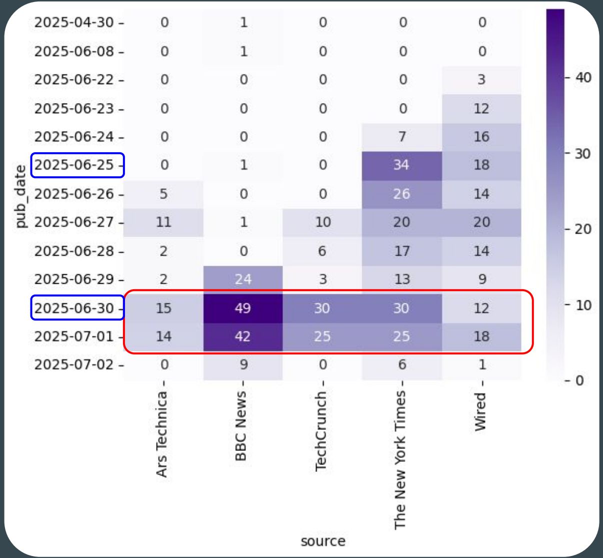


```
▼<item>
   <title>The 10 Best Carry-On Luggage for Business Travelers</title>
   <link>https://www.wired.com/story/best-carry-on-luggage-for-business-travelers/</link>
   <guid isPermaLink="false">686550b9c1c031cd22ee175e</guid>
   <pubDate>Wed, 02 Jul 2025 16:01:07 +0000</pubDate>
   <media:content/>
   <description>Whether you're looking for whisper-quiet wheels or the most durable of hard-shells, here are WIRED and Condé Nast Traveler's essential carry-on bags.</description>
   <category>Gear</category>
   <media:keywords>The Future of Business Travel, Shopping, Travel, buying guides, Bags</media:keywords>
   <dc:creator>Meaghan Kenny, Adrienne So</dc:creator>
   <dc:publisher>Condé Nast</dc:publisher>
   <dc:subject>In the Bag</dc:subject>
   <media:thumbnail url="https://media.wired.com/clips/686532268629b1e1b142071e/master/pass/WIREDxCONDENAST%20no.6-Carry-Ons.mp4" width="2838" height="1684"/>
 </item>
▼<item>
```

# Exploratory Data Analysis

- More than 1200 categories in for 5 sources in 13 days* of information.
- Around 130 articles per day just with 5 sources (in peak collection days*)

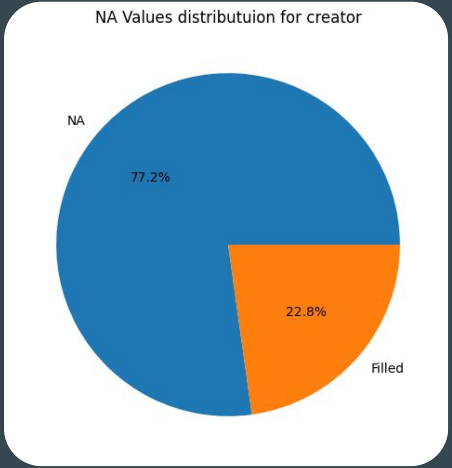


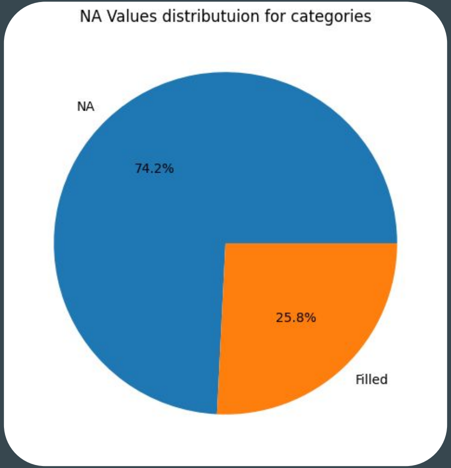


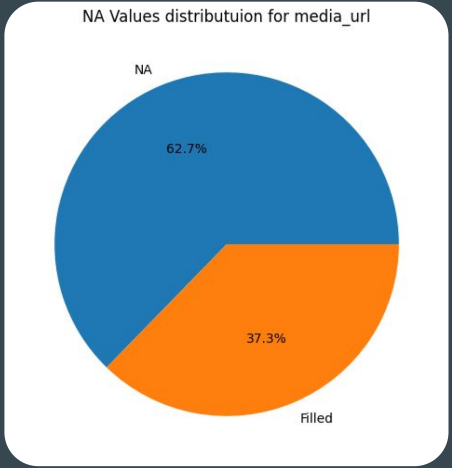
| pub_date | Ars Technica | BBC News | TechCrunch | The New York Times | Wired |
|---|---|---|---|---|---|
| 2025-04-30 | 0 | 1 | 0 | 0 | 0 |
| 2025-06-08 | 0 | 1 | 0 | 0 | 0 |
| 2025-06-22 | 0 | 0 | 0 | 0 | 3 |
| 2025-06-23 | 0 | 0 | 0 | 0 | 12 |
| 2025-06-24 | 0 | 0 | 0 | 7 | 16 |
| 2025-06-25 | 0 | 1 | 0 | 34 | 18 |
| 2025-06-26 | 5 | 0 | 0 | 26 | 14 |
| 2025-06-27 | 11 | 1 | 10 | 20 | 20 |
| 2025-06-28 | 2 | 0 | 6 | 17 | 14 |
| 2025-06-29 | 2 | 24 | 3 | 13 | 9 |
| 2025-06-30 | 15 | 49 | 30 | 30 | 12 |
| 2025-07-01 | 14 | 42 | 25 | 25 | 18 |
| 2025-07-02 | 0 | 9 | 0 | 6 | 1 |

* Data collection started with TNYT on 25/06 the rest of the sources were added on 30/06 and finished on 02/07 morning
Articles previous to these dates were provided by the RSS sources during the mentioned period.

# EDA - Other Learnings







The RSS parser logic needs to be improved:

- Main fields (title, link, pubDate) are standard.
- But there are alternative implementations (media:thumbnail / media:content, description / summary)
- And some are not always defined (category, dc:creator, content,...)

## EDA - Other Learnings

Selective data cleaning:

- Same fields in different sources might have different formats.
- The format is not always explicit.

```
<summary type="html">
  <![CDATA[ Amazon’s early Prime Day sale
  started off slow, but as the event draws
  closer to its official kickoff on July 8th,
  it’s beginning to pick up steam. Right now,
  for example, Prime members can grab an Amazon
  Echo Hub at Amazon for just $119.99 ($60 off),
  marking a new low price. As you might expect,
  [&#8230;] ]]>
</summary>
```

```
<description>Weather agencies have
warned of a third day of high
temperatures that in some places have
climbed well above 100 degrees
Fahrenheit, or more than 40 degrees
Celsius.</description>
```

```
<description><![CDATA[TikTok Shop US is
conducting its third round of layoffs in as
many months.]]></description>
```

## Implementation Details

## 1. Obtain Data and Feed RAG

- Use requests() to pull xml from source.
- Store xml into a file per source (file was replaced in each update)
- UnstructuredXMLLoader to feed RAG and “snowflake-arctic-embed2” as embedder.
- OllamaChat to process rag requests with “Llama3.1:8B”

### Results and Learnings:

- UnstructuredXMLLoader was not properly loading de documents.
- XML files were slow and I was losing older articles (in every update).
- VectorStore data management was not possible.
- RAG wasn’t the proper solution for creating top topics.

## 2. Custom RSS Loader and SQLite

- Implemented a custom algorithm to read and parse RSS to langchain Documents.
- Implemented a SQLite Database to store the articles and not older articles with each update.
- Used articles id in database to manage content on ChromaVS in order to avoid duplicated data.
- Started splitting the code in modular classes and implemented the initial controllers.

### Results and Learnings:

- The process speed increased with the database.
- RAG results started to be relevant.

## 3. Top Topics and Streamlit

- Use SQLite database to send the articles of the day directly to the LLM in order to get the Top Topics.
- Streamlit client showing the list of articles and sources in the database and top topics as Trending Topics.
- Worked with the prompt in order to obtain a valid JSON from the model.
- Switched from OllamaChat to OllamaLLM.
- Switched model to Llama3.2:3b.

### Results and Learnings:

- Llama3.2:3B provide similar results than the previous model with less resources.
- Getting a JSON as answer was easy, ensuring it was correct not.
- I found Streamlit a bit limited in components but I managed to use the default ones.
- Reduced prompt complexity.

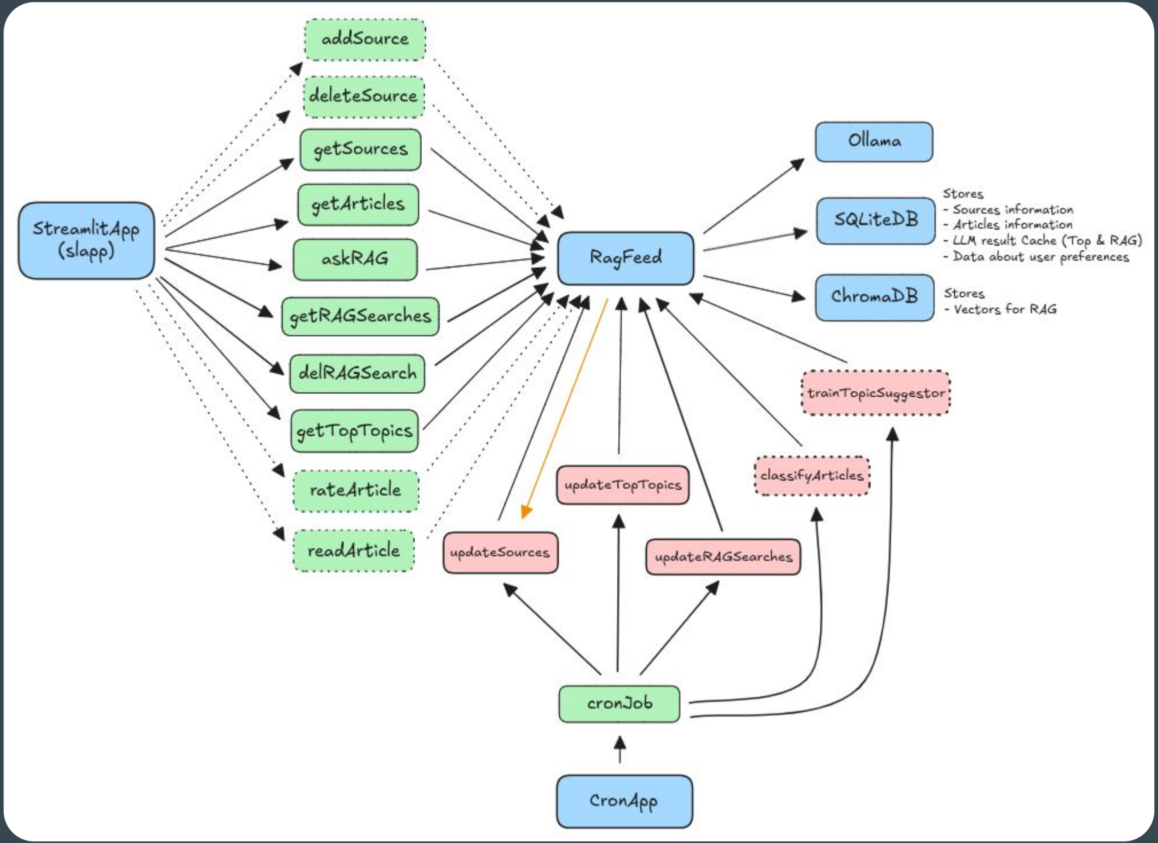
## 4. Integrating RAG Search and LLM GuardRails

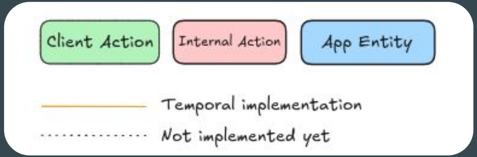
- Modified the prompt of the RAG functionality to return a JSON.
- Created the interface in Streamlit, changing the general search for a “follow topic” system.
- Using SQLite database as cache to reduce number of inferences when not needed (and user wait time).
- Implemented a prompt to correct wrong JSON formats returned by RAGSearch and TopTopics.
- Implemented a cron to update RagSearches and TopTopics out of the client.

### Results and Learnings:

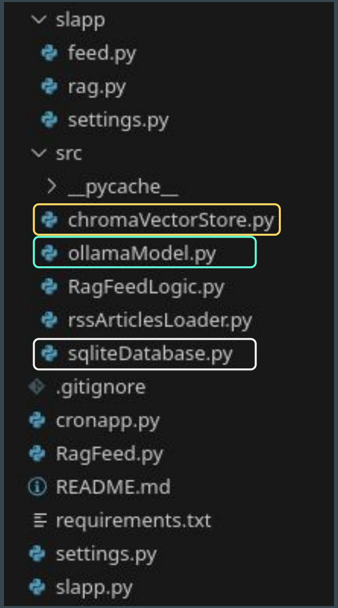
- Improved app usability.
- Improved reliability of the model results.

# App structure and endpoints

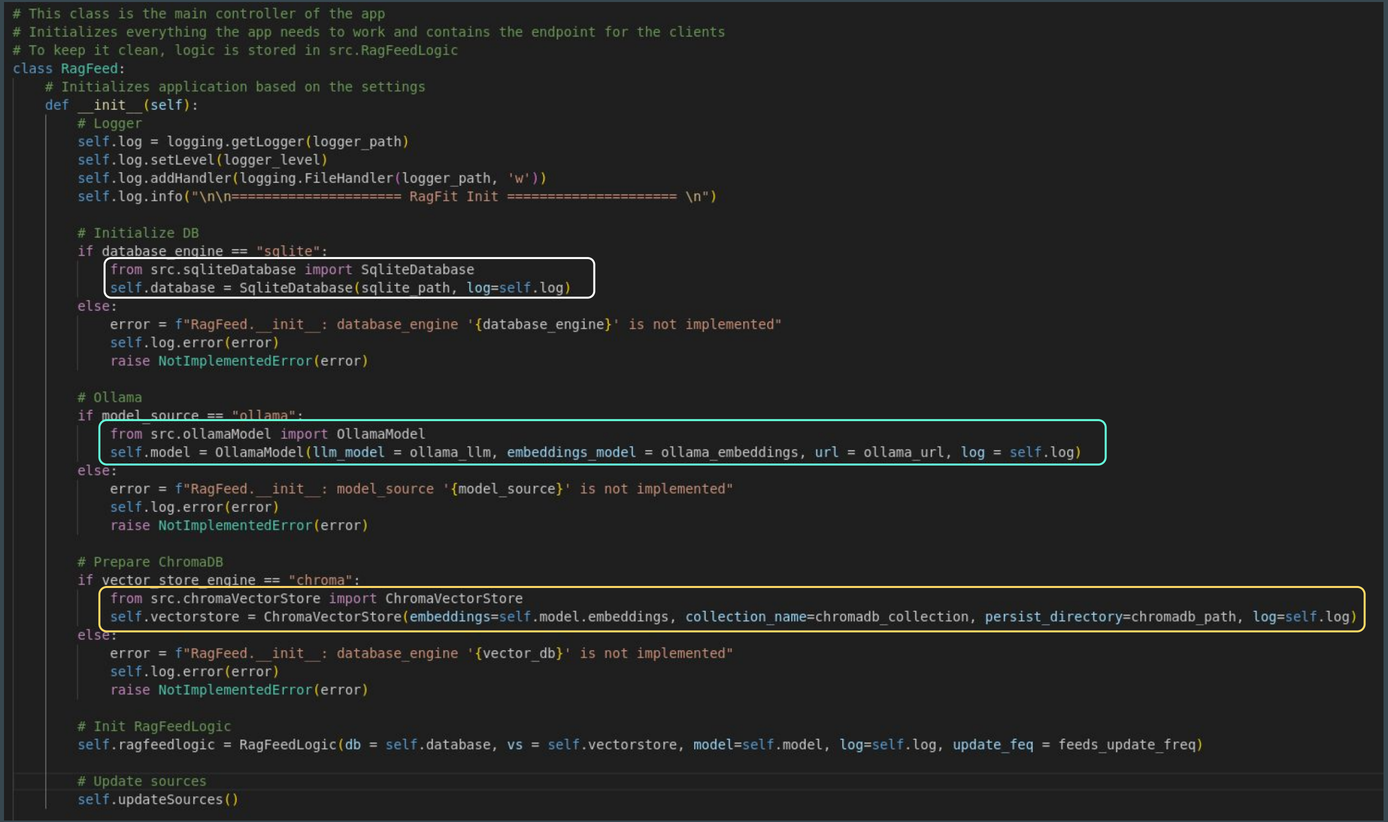




# Modular Implementation

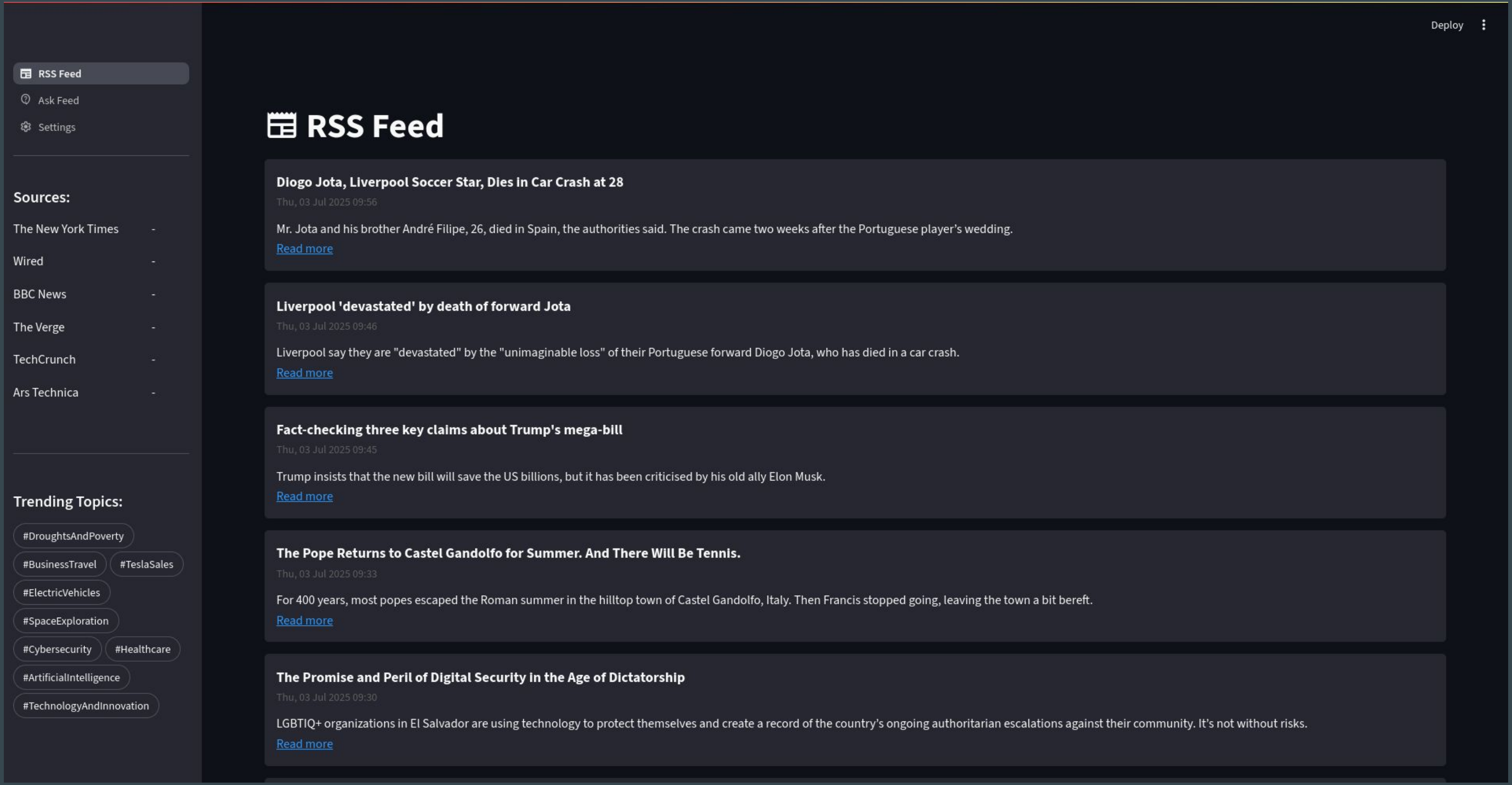


```
∨ slapp
    feed.py
    rag.py
    settings.py
∨ src
  > __pycache__
    chromaVectorStore.py
    ollamaModel.py
    RagFeedLogic.py
    rssArticlesLoader.py
    sqliteDatabase.py
  .gitignore
  cronapp.py
  RagFeed.py
  README.md
  requirements.txt
  settings.py
  slapp.py
```



```python
# This class is the main controller of the app
# Initializes everything the app needs to work and contains the endpoint for the clients
# To keep it clean, logic is stored in src.RagFeedLogic
class RagFeed:
    # Initializes application based on the settings
    def __init__(self):
        # Logger
        self.log = logging.getLogger(logger_path)
        self.log.setLevel(logger_level)
        self.log.addHandler(logging.FileHandler(logger_path, 'w'))
        self.log.info("\n\n===================== RagFit Init ===================== \n")

        # Initialize DB
        if database_engine == "sqlite":
            from src.sqliteDatabase import SqliteDatabase
            self.database = SqliteDatabase(sqlite_path, log=self.log)
        else:
            error = f"RagFeed.__init__: database_engine '{database_engine}' is not implemented"
            self.log.error(error)
            raise NotImplementedError(error)

        # Ollama
        if model_source == "ollama":
            from src.ollamaModel import OllamaModel
            self.model = OllamaModel(llm_model = ollama_llm, embeddings_model = ollama_embeddings, url = ollama_url, log = self.log)
        else:
            error = f"RagFeed.__init__: model_source '{model_source}' is not implemented"
            self.log.error(error)
            raise NotImplementedError(error)

        # Prepare ChromaDB
        if vector_store_engine == "chroma":
            from src.chromaVectorStore import ChromaVectorStore
            self.vectorstore = ChromaVectorStore(embeddings=self.model.embeddings, collection_name=chromadb_collection, persist_directory=chromadb_path, log=self.log)
        else:
            error = f"RagFeed.__init__: database_engine '{vector_db}' is not implemented"
            self.log.error(error)
            raise NotImplementedError(error)

        # Init RagFeedLogic
        self.ragfeedlogic = RagFeedLogic(db = self.database, vs = self.vectorstore, model=self.model, log=self.log, update_feq = feeds_update_freq)

        # Update sources
        self.updateSources()
```

# RagFeed - RSS Feed



Deploy

RSS Feed
Ask Feed
Settings

**Sources:**

The New York Times -
Wired -
BBC News -
The Verge -
TechCrunch -
Ars Technica -

**Trending Topics:**

#DroughtsAndPoverty #BusinessTravel #TeslaSales #ElectricVehicles #SpaceExploration #Cybersecurity #Healthcare #ArtificialIntelligence #TechnologyAndInnovation

## RSS Feed

**Diogo Jota, Liverpool Soccer Star, Dies in Car Crash at 28**
Thu, 03 Jul 2025 09:56

Mr. Jota and his brother André Filipe, 26, died in Spain, the authorities said. The crash came two weeks after the Portuguese player's wedding.
Read more

**Liverpool 'devastated' by death of forward Jota**
Thu, 03 Jul 2025 09:46

Liverpool say they are "devastated" by the "unimaginable loss" of their Portuguese forward Diogo Jota, who has died in a car crash.
Read more

**Fact-checking three key claims about Trump's mega-bill**
Thu, 03 Jul 2025 09:45

Trump insists that the new bill will save the US billions, but it has been criticised by his old ally Elon Musk.
Read more

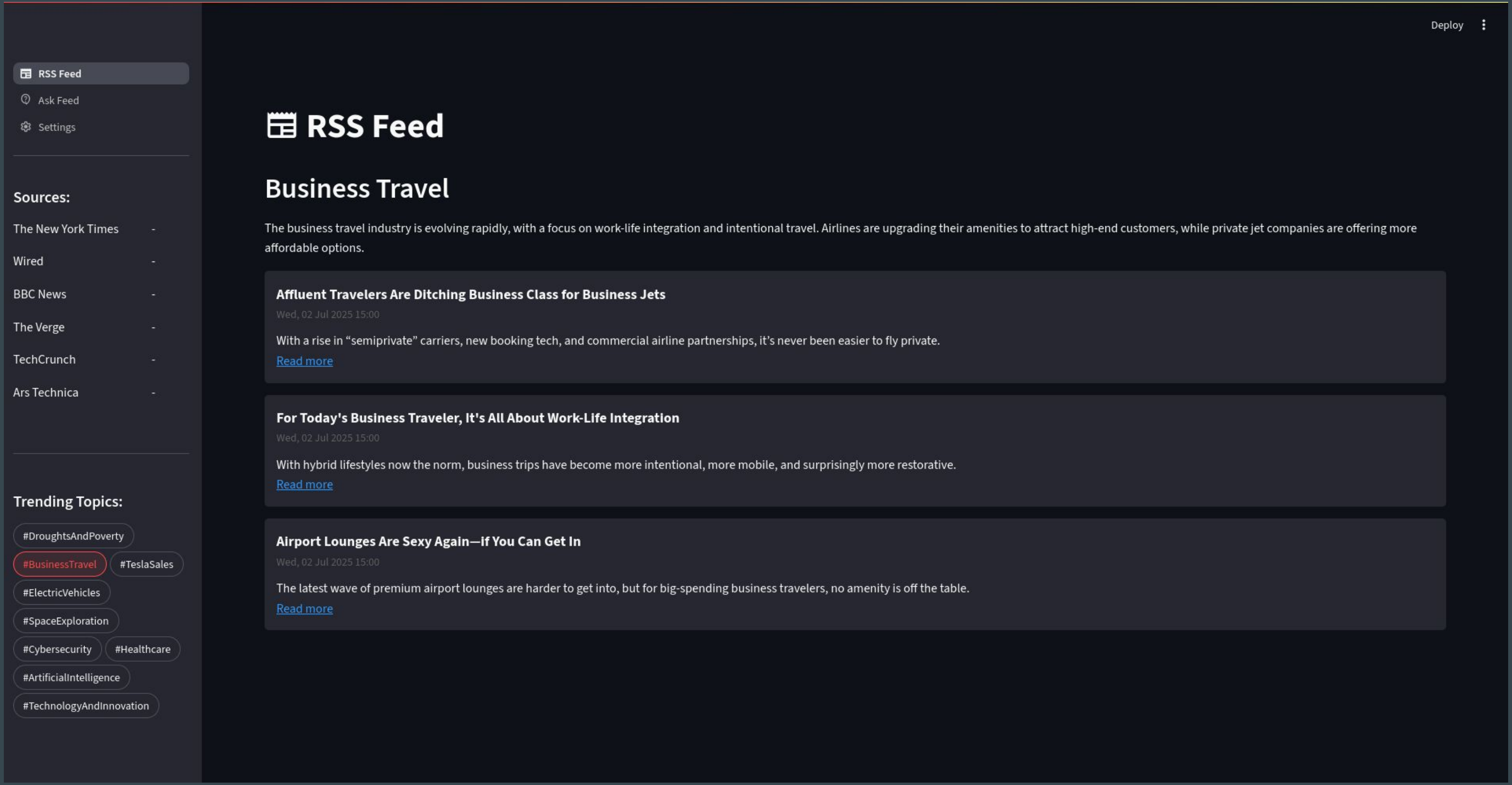
**The Pope Returns to Castel Gandolfo for Summer. And There Will Be Tennis.**
Thu, 03 Jul 2025 09:33

For 400 years, most popes escaped the Roman summer in the hilltop town of Castel Gandolfo, Italy. Then Francis stopped going, leaving the town a bit bereft.
Read more

**The Promise and Peril of Digital Security in the Age of Dictatorship**
Thu, 03 Jul 2025 09:30

LGBTIQ+ organizations in El Salvador are using technology to protect themselves and create a record of the country's ongoing authoritarian escalations against their community. It's not without risks.
Read more

# RagFeed - Trending Topics



**Sources:**

| | |
|---|---|
| The New York Times | - |
| Wired | - |
| BBC News | - |
| The Verge | - |
| TechCrunch | - |
| Ars Technica | - |

**Trending Topics:**

#DroughtsAndPoverty #BusinessTravel #TeslaSales #ElectricVehicles #SpaceExploration #Cybersecurity #Healthcare #ArtificialIntelligence #TechnologyAndInnovation

## RSS Feed

### Business Travel

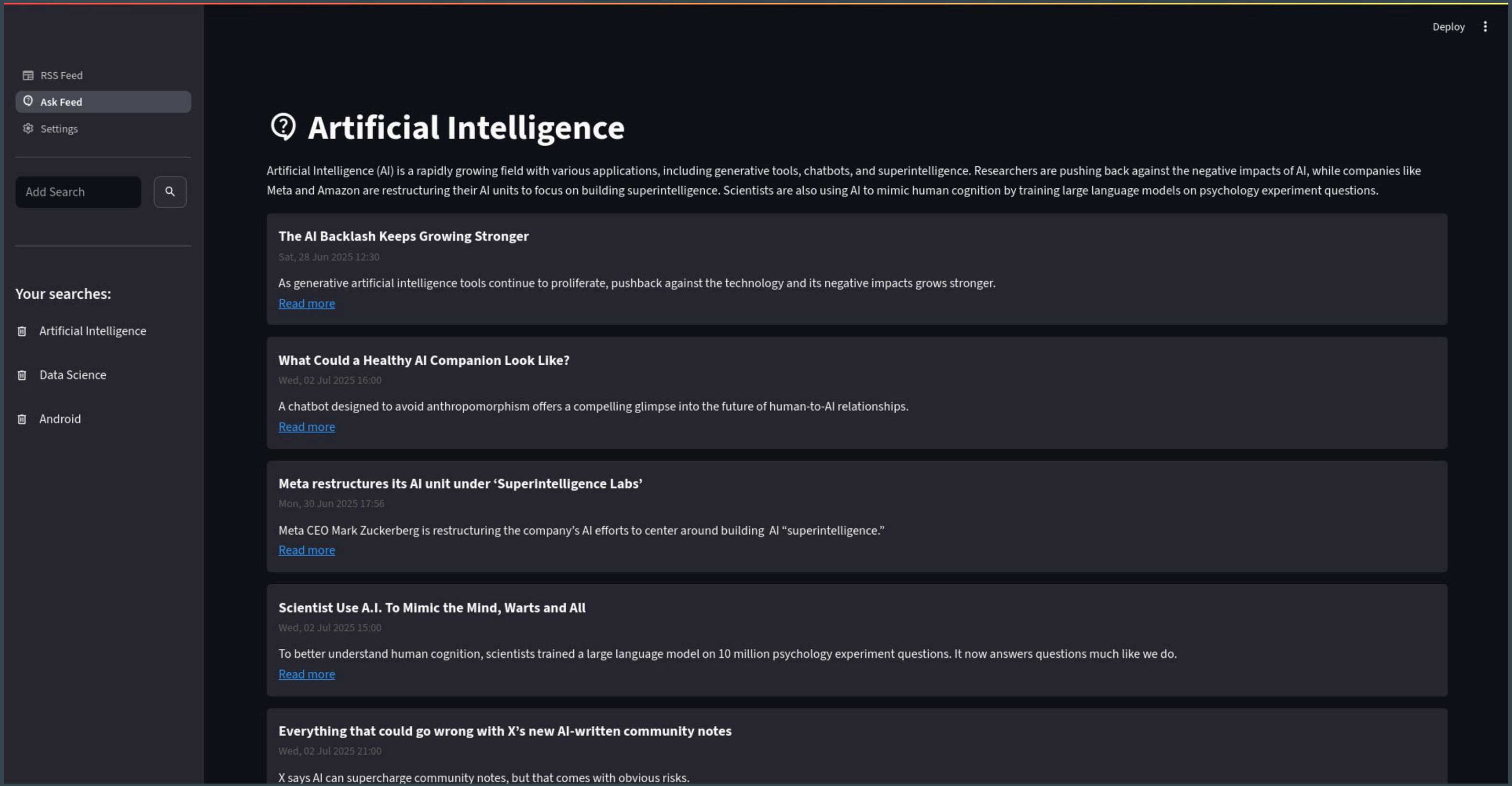
The business travel industry is evolving rapidly, with a focus on work-life integration and intentional travel. Airlines are upgrading their amenities to attract high-end customers, while private jet companies are offering more affordable options.

**Affluent Travelers Are Ditching Business Class for Business Jets**
Wed, 02 Jul 2025 15:00
With a rise in "semiprivate" carriers, new booking tech, and commercial airline partnerships, it's never been easier to fly private.
Read more

**For Today's Business Traveler, It's All About Work-Life Integration**
Wed, 02 Jul 2025 15:00
With hybrid lifestyles now the norm, business trips have become more intentional, more mobile, and surprisingly more restorative.
Read more

**Airport Lounges Are Sexy Again—If You Can Get In**
Wed, 02 Jul 2025 15:00
The latest wave of premium airport lounges are harder to get into, but for big-spending business travelers, no amenity is off the table.
Read more

# RagFeed - Ask Rag



Your searches:

- Artificial Intelligence
- Data Science
- Android

## Artificial Intelligence

Artificial Intelligence (AI) is a rapidly growing field with various applications, including generative tools, chatbots, and superintelligence. Researchers are pushing back against the negative impacts of AI, while companies like Meta and Amazon are restructuring their AI units to focus on building superintelligence. Scientists are also using AI to mimic human cognition by training large language models on psychology experiment questions.

**The AI Backlash Keeps Growing Stronger**
Sat, 28 Jun 2025 12:30
As generative artificial intelligence tools continue to proliferate, pushback against the technology and its negative impacts grows stronger.
Read more

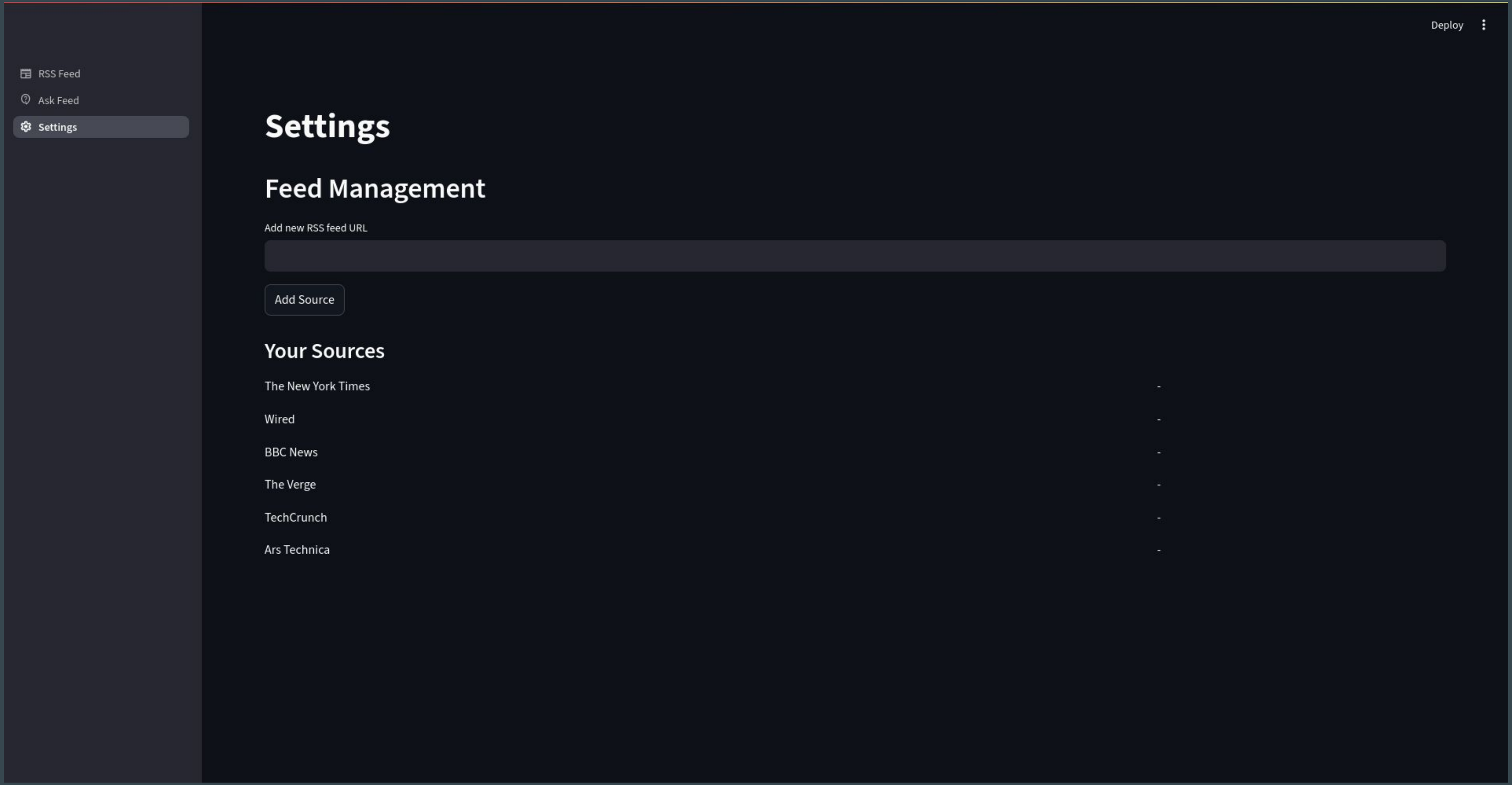
**What Could a Healthy AI Companion Look Like?**
Wed, 02 Jul 2025 16:00
A chatbot designed to avoid anthropomorphism offers a compelling glimpse into the future of human-to-AI relationships.
Read more

**Meta restructures its AI unit under 'Superintelligence Labs'**
Mon, 30 Jun 2025 17:56
Meta CEO Mark Zuckerberg is restructuring the company's AI efforts to center around building AI "superintelligence."
Read more

**Scientist Use A.I. To Mimic the Mind, Warts and All**
Wed, 02 Jul 2025 15:00
To better understand human cognition, scientists trained a large language model on 10 million psychology experiment questions. It now answers questions much like we do.
Read more

**Everything that could go wrong with X's new AI-written community notes**
Wed, 02 Jul 2025 21:00
X says AI can supercharge community notes, but that comes with obvious risks.

# RagFeed - Settings



Settings

Feed Management

Add new RSS feed URL

Your Sources

| | |
|---|---|
| The New York Times | - |
| Wired | - |
| BBC News | - |
| The Verge | - |
| TechCrunch | - |
| Ars Technica | - |

## Tools and Technologies

## Tools and Technologies

- GenAI: Obtain TopTopics, RagSearch and fix JSON
- RAG: Feed RagSearch
- Ollama: Manage the LLM and Embedder
- LangChain: Manage Documents and Chroma
- ChromaVS: Store the documents for RAG
- SQLite: Store the Articles and Sources data and cache llm answers.
- BeautifulSoup: Parse RSS XML to a usable format.
- Streamlit: User front-end.
- Transformers: Count Tokens

## Major Obstacle

## Ask RAG for Top Topics

- I was expecting a list of topics related to the current date.
- RAG returns documents related to the words “top”, “topics” and “today”.

### Solution and improvements:

- Feed the LLM with data in SQLite to generate TopTopics.
- Won more control on what today mean (last 24h)

## UnstructuredXMLLoader documents parse.

- Documents created had no description or link.
- RAG answer were not good and wasn’t able to provide the the source.

### Solution and improvements:

- Use BeautifulSoup to parse XML.
- Organize the information in the SQLite database.
- Use langchain_core.documents to create documents
  - Decided what data add in page_content and metadata.
  - Decided the format of the data added in VS.

## Manage data in Vector Store

- I wasn't able to know what articles were inside Chroma.
- I had to recreate vector store everytime I delete or update articles.
- I was duplicating information (feeding VS directly from RSS).

### Solution and improvements:

- Use SQLite as source of documents.
- Use article id as id in vector store so I can delete documents.
- Have a boolean indicator in DB to know if article is in VS

## Model inference time

- Ollama requires some time to execute the inference.
- The time vary depending on the length of the prompt.
- Client App (streamlit) might seem freeze while waiting.

### Solution and improvements:

- Implemented a cron task to periodically update articles.
- Cron also updates TopTopics and RagSearches when articles were updated.
- SQLite stores the last result so the app don’t need to call the model every time.
- Rollback to Llama3.2:8b since 3.1:3b was causing ollama to freeze.

## JSON from model inference.

- Model results where good as markdown.
- Transform the answer to a JSON was easy.
- Get a valid JSON (parseable) in the proper format reliably was challenging.

### Solution and improvements:

- Reduce model temperature.
- Improved prompting.
- Implemented GuardRails
    - Prompt focused on fix JSON structure.
    - Adapted the the code to be flexible to the prompt structure.
    - Implemented an alternative path to avoid the app to crash.

## Conclusions and Insights

# Conclusions and Insights

- Developing an application using GenAI as part of the backend is fast.

- Get reliable results from LLM, specially when structured formats are required.

- Working with local models:
    - Start with small models and change to bigger ones if needed (when the prompt is well optimized).
    - Good for an MVP (focus in make the app work first and later optimize it).
    - Makes app hard to deploy.

- RAG is powerful and seems easy but proper document formatting and data management is key.

- Do not focus on getting perfect prompts before integrating with the client.

- Iterate and be flexible with the ideas.

- Implement a logging system soon in your app.

## Next Steps

- Improve RSS parse algorithm.
- Source management flow.
- Allow to mark articles as Read/Undead.
- Allow to rate articles interest.
- Internal reader and track number of times an article is open.
- Use the information about RagSearch, interest rate and num times open to train an “Interest Classifier” ML Algorithm.
- Improve app look and feel (especially in light mode).
- Async tasks instead of a cron for updating sources, TopTopic and RagSearch.
- Train an Unsupervised Learning Algorithm to find better categories or articles relations.

Thank you.