PROJECT TITLE

Personalized News Aggregator Using AI Agents

PROJECT DESCRIPTION

Questions to Answer / Problems to Solve

1. How can I stay updated on topics I care about without spending hours sifting through irrelevant information?
2. How can AI agents make it easier to process and summarize news from multiple sources efficiently?

Relevance to Business

- Question 1: Staying informed on specific topics like technology and leadership is crucial for personal growth and professional decision-making. A personalized news tool saves users time and ensures they don't miss important updates in their field.
- Question 2: Automating the process of finding and summarizing relevant news allows for quicker insights, making it easier for individuals and businesses to focus on actions rather than information gathering.

Four-Step Plan and Expected Results

1. Understand User Preferences
    - Result: A user interest profile that guides the system on what type of news articles to fetch (e.g., topics like AI, management strategies).
2. Gather News Content
    - Result: Relevant news articles are collected from trusted websites using RSS feeds and APIs.
3. Summarize Articles
    - Result: Each article is summarized into a short, easy-to-read format that captures the main points.
4. Deliver Personalized News

    - Result: A personalized feed is created and updated daily or on-demand, offering users a simple way to stay informed.

SOFTWARE

- Programming Language: Python
- AI and NLP Libraries:
    - spaCy: For natural language processing tasks.
    - Hugging Face Transformers: For article summarization using pre-trained models.
- Web Libraries:
    - feedparser: For retrieving RSS feed data.
    - BeautifulSoup: For scraping articles when RSS feeds aren't available (if allowed by terms of service).
- Platform: crewAI for managing AI agents.
- Database: SQLite or a lightweight JSON file for storing user preferences and article metadata.
- Interface Framework: Flask for building a simple user interface.

All tools are free and openly available. No special licenses are required, and setup instructions can be shared with colleagues in the course.

DATASET DESCRIPTION

- Volume and Format:
    - Data includes news articles and metadata from RSS feeds in XML format, typically lightweight and manageable.
    - Articles are updated daily or hourly depending on the source.
- Existing Data:
    - Publicly available RSS feeds from websites like TechCrunch, Wired, Harvard Business Review, and similar sources.
- Additional Data Needed:
    - User preferences collected during system setup.
    - Interaction data (e.g., which articles users read or dismiss) to refine the interest profile over time.
- How to Access Data:
    - RSS feeds are accessed using the feedparser library.

    - APIs (if available) can be used to fetch additional articles.
    - Websites without RSS feeds can be scraped (if allowed), but this is not a primary data source.
- Starting Points:
    - Example RSS feeds: TechCrunch, Wired, and Harvard Business Review

---

RESTRICTIONS

- License and Terms of Use:
    - RSS feeds are generally free for personal and non-commercial use, but specific terms of service from each source will be respected.
    - Storing or redistributing full article content will be avoided unless permitted.
- Simulated or Anonymized Data:
    - If necessary, sample articles and synthetic datasets can be created for testing and demonstration purposes.
    - Article metadata (titles and descriptions) from RSS feeds can be used to simulate summaries without accessing the full content.

HARDWARE

- Requirements:
    - No specialized hardware is required for this project.
- Infrastructure:
    - Data will be processed locally, and cloud services are not required.