CNN Data Intelligence: MLOps Eng Take Home Project

Hello! We're excited you're thinking of joining CNN Digital. Our team, Data Intelligence, builds products and features that understand our audience and our content. Hundreds of millions of people interact with CNN on a monthly basis, and we're just getting started building top-notch features to help the world understand what's happening around them.

We prepared this take-home project for candidates to help us get to know one another and give you the chance to show off your skills in data analysis, engineering and machine learning. We expect you to take no more than 4 hours on it in total.

Project: Suggest the News

Imagine a tab or sidebar on CNN.com or its apps that could show you the next 5 things you need to read. Would they be related to the article you're reading now? Or based on your personal history today? Or popular or breaking news items? Your choice! Let's build a feature with a simple API that given a user_id, returns the ids of the articles that the user should read next.

The data:
We are providing historical article view data for you to bootstrap your model in any way you see fit. In the .zip file provided, we've synthesized 50,000 users worth of lifelike triplets of [article_id, user_id, timestamp], roughly 1 million rows, in the file article_user.csv. We've also provided a file called article.csv with 1,455 triplets of [article_id, headline, timestamp] representing the articles read by these users.

The task:
Analyze the data, and develop an API that returns 5 new article_ids given a user_id. These articles should be related to the user_id or the piece of content they are viewing (defined as the latest article_id by timestamp for that user_id) in some way. The API you develop should respond over HTTP, accept user_id, and return an ordered list of article_ids (optionally, with "scores" indicating the fit between article_id and the user). Do not worry about creating a datastore; just return the article_id list from memory.

How to choose the articles for a user? It's up to you. A personalization system would use the user_ids history to first train a model that would inform which articles to show. A related content feature could use the text in the headlines. A "trending now" feature could use popularity and timestamps.

What to deliver:

- We'd love to see a short document to go along with code (attached in a zip file or private code repository). The document should show your thought process for the design of the algorithm and API, how you analyzed or trained on the data, and how you evaluated the results.
- Please use any major language you're comfortable with: Python, Go, JavaScript, C++ are all fine.
- Please instruct us how to run your code.
- Prioritize showing your work around how you chose an approach to recommending news over building the API. If you run short on time, it's better to spend less time on the API and provide instructions on how to accept a user_id and output article_ids via the command line than to spend less time on your approach to recommending articles in the first place.
- You don't have to write it down, but be prepared to discuss how you would operationalize this system (deployment, automated retraining, etc.).

Tips:

- Simple is best! There are thousands of ways to analyze this data, provide predictions, and deploy the model. But, we always look for solutions that we can introspect and iterate on as we grow. Some of the world's top personalization products are built on very simple approaches.
- Bring your passion! If you've ever been frustrated with current solutions, or had an idea that would help people understand the news better, give it a shot. We're always excited to bring in new approaches from the team, or be inspired by new academic or industrial research.
- Take advantage of others! We don't expect you to build everything from scratch. Use other models or libraries, do some light reading. We all work as a team here to learn from each other, so bring your own skills but don't try to do it all yourself.