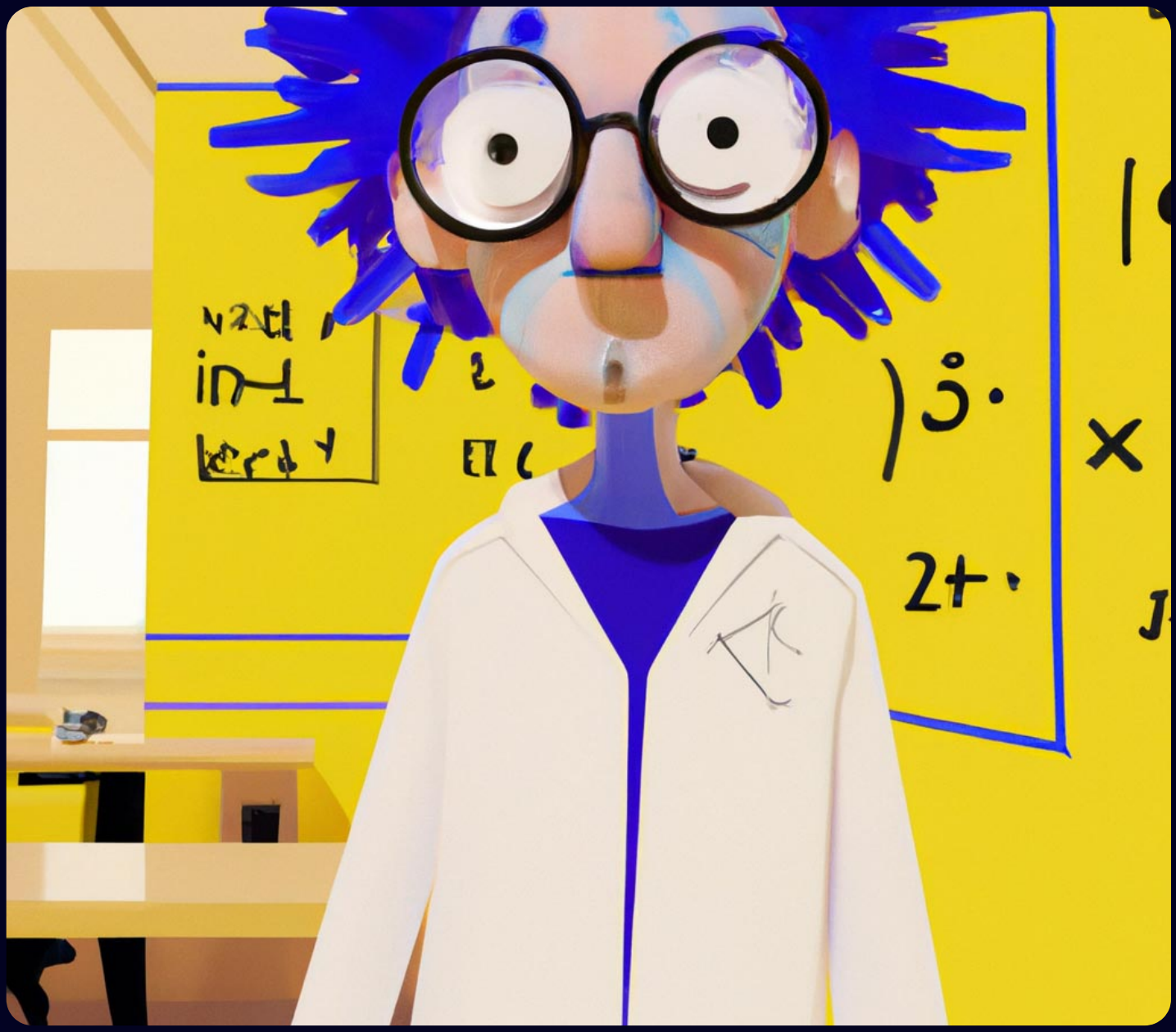
Real-time sentiment analysis with Python libraries

February 23, 2022 · 3 min read



Dr. Deephaven
Deep Data Doc @Deephaven

Support Chat

```
# Real-time streams, AI and batch data converge in
# Deephaven tables
```

With the uptick in machine learning (ML) and artificial intelligence (AI) capabilities, sentiment analysis is certainly nothing new to the developer and data science communities. This work, though, has largely been performed on static data sets.

What if you could perform sentiment analysis on Twitter or Reddit, with Python in real time? That is to say, take all of the historical batch data on a topic, feed real-time updates into this set, and then analyze the whole model (streaming analytics and batch data alike) in one dynamic table.

With Deephaven, all of that is possible since the Python API seamlessly integrates Deephaven's real-time capabilities with the most powerful Python sentiment analysis libraries, including TensorFlow, NLTK, PyTorch, and others. The result is a two-way data transfer that allows real-time ML/AI calculations to occur within tables that are updating continuously.

In terms of sentiment analysis, you could immediately glean current public opinion of trending topics in news and social media by analyzing the information, moments after it becomes available. You could predict the potential Cinderella stories of March Madness, for example. Virtually, any sentiment analysis project is possible in real time, with Deephaven analyzing the tweets, comments, and posts of users around the internet.

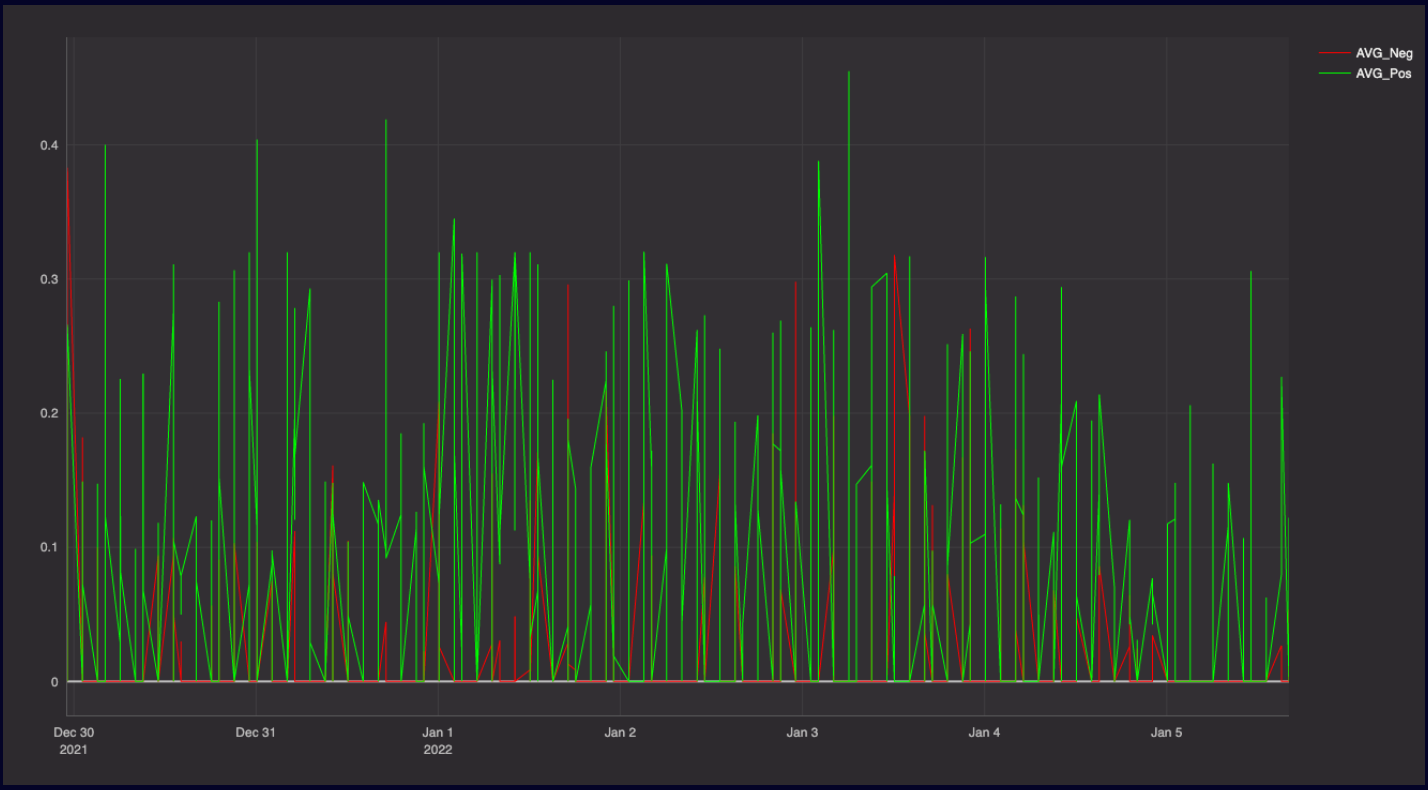
And did we mention crypto? We've included an example for analysis of Twitter sentiment about DOGE coin below. We've also included an additional Twitter sentiment project that leverages TensorFlow, and one that analyzes Reddit subtopics in real time.

Read through the below examples to learn how this real-time AI is possible. And then think about what else you could do with this powerful capability—real-time AI—for the next great sentiment analysis project.

Sentiment analysis examples

Twitter sentiment analysis for meme stocks

Let's expand our Reddit example above onto Twitter. In this scenario, we demonstrate how Python and TensorFlow's natural language processing libraries and Deephaven's stream-and-batch table engine combine to analyze Twitter sentiment of DOGE coin. This project also leverages Python's NLTK to pull recent tweets and analyze sentiment in real time. The article then goes on to explain in great detail how to configure your functions, clean the tweets, and then assess sentiment. Finally, we put our data into a table and analyze. Furthermore, we can leverage Deephaven's plotting methods to create a data visualization of the resultant table.



Reddit sentiment analysis for meme stocks

In the example provided, we leveraged the Python feedparser package to pull RSS feed data from the WallStreetBets subreddit into Deephaven. We used Deephaven's

DynamicTableWriter to transform this real-time data into a table that we can write to. Once all of this data is writing to our Deephaven table, we then employ the Python NLTK (Natural Language Toolkit) package for sentiment analysis of the complete table.

Real-time sentiment analysis using an LSTM network in TensorFlow

Sticking with the Twitter example, this article explains how to leverage Python machine learning (i.e., TensorFlow) for analyzing the sentiment of a GOP debate from 2016. We're leveraging a long short-term memory (LSTM) network, which is typically used for time-series forecasting and speech/image recognition, and can also greatly facilitate sentiment analysis.

Our data is historical, but we also simulate the real-time streams here (in addition to the batch data from the Kaggle project) to demonstrate that analysis on both batch and real-time data is possible. Reference the full project on GitHub here.

Furthermore, it's particularly interesting to see how we train the AI model, which in Deephaven is performed within a function.

AI/ML Python Dr Deephaven

Get Started

Use our tutorial to learn Deephaven. Set up your Docker environment with three CLI commands to install Deephaven from pre-built images.

Quick Start →