Chat GPT News Headline Sentiment Analysis

Damaris Santiago, Sonali Singh, Susmitha Kusuma

12 May 2023

Big Data

Professor Rodriguez

Spring 2023

Introduction

In today's digital world, news headlines play a vital role in shaping public opinion and influencing decision-making processes. The advent of artificial intelligence (AI) and natural language processing (NLP) technologies has revolutionized the way news is created, disseminated, and consumed. ChatGPT, a new "state-of-the-art" language model developed by OpenAI, represents a significant milestone in AI-driven language generation. Launched on November 30th, 2022, ChatGPT has attracted widespread attention and sparked intense discussions across various domains.

This final project aims to investigate the sentiment of news headlines related to ChatGPT and how that sentiment has evolved since ChatGPT's launch. By analyzing the sentiment of these headlines, we can gain valuable insights into public perception and the overall attitude towards surrounding ChatGPT.

Problem Statement

The problem addressed in this final project is to analyze the sentiment of news headlines about ChatGPT and track how it has changed over time. The sentiment analysis of news headlines is important for understanding public perception and sentiment trends, which can have significant implications for decision-makers, businesses, and the wider AI community.

By leveraging the Python Google News API, GNews, we have obtained a comprehensive collection of news articles relevant to ChatGPT over a 5-month period from its release on November 30th, 2022 to April 30th, 2023. These articles were then processed using Kafka for efficient data ingestion and Spark Structured Streaming for robust data processing and sentiment analysis. The sentiment analysis allows us to classify the headlines into positive, neutral, or negative sentiments, providing valuable insights into the overall sentiment surrounding ChatGPT.

The primary objective of this project is to:

1. Analyze the sentiment of the news headlines relating to ChatGPT.
2. Identify and track sentiment trends and changes over time since the model's launch.
3. Analyze the breakdown of article sentiments (positive, neutral, negative) within each publishing / news media company.
4. Provide actionable insights and a comprehensive analysis of sentiment towards ChatGPT.

By achieving these objectives, this project aims to contribute to a deeper understanding of the overall sentiment dynamic surrounding ChatGPT.

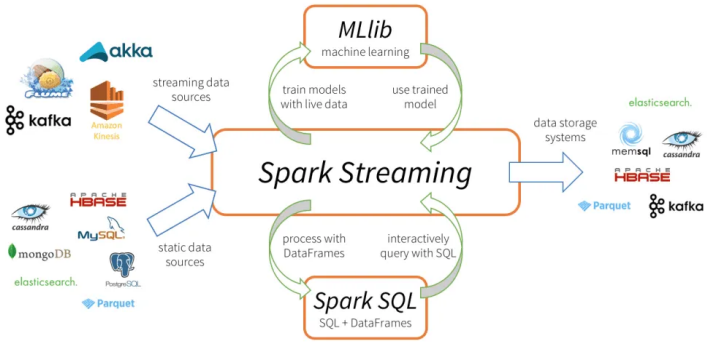
What is Sentiment Analysis?

A sentiment is an opinion of, or attitude towards, something which can be conveyed with words. Sentiment analysis is the process of analyzing text to determine if its emotional tone is positive, neutral, or negative. Sentiment analysis is a valuable technique that enables us to gauge the

emotions, opinions, and attitudes expressed within textual data. It involves using computational techniques to automatically classify and quantify the sentiment associated with a given text, allowing for actionable insights to be derived.

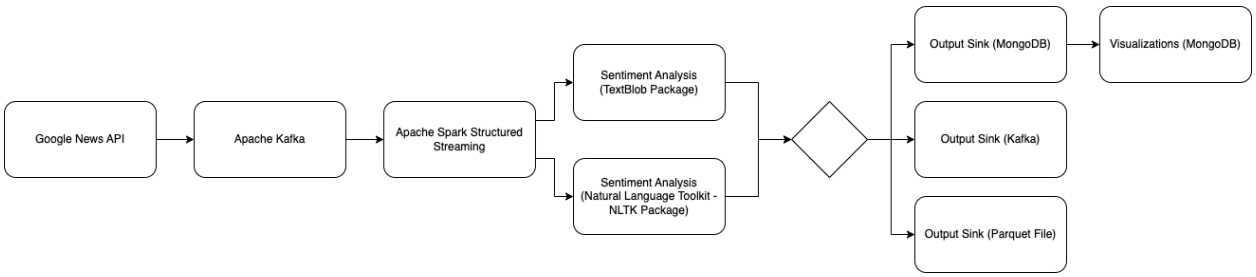
What is a Data Pipeline?

In order to successfully implement our project, we built a data pipeline. A data pipeline is a big data method or set of actions that ingests raw data from one or multiple sources, whether continuous, static, or both. This data is then processed, transformed, and stored into an output storage system.



Our Data Pipeline

The data pipeline begins with our streaming data source, GNews. GNews is a Google News Python API that can be queried to fetch articles, formatted as a JSON response. That JSON response acts as our raw data. Each time the API is queried, this streaming data is sent to Kafka which acts as a packaging and transportation vehicle for our data from the GNews API to Spark. Kafka serves this function because Spark cannot directly interact with GNews. Once the data is delivered to Kafka, Spark reads the streaming data from Kafka into a structured streaming dataframe that we can use to perform sentiment analysis in a distributed, parallel manner. In this pipeline, sentiment analysis is performed on the dataframe using two popular libraries, TextBlob and NLTK. Once sentiment analysis is complete, the transformed dataframe is stored in three different output sinks: MongoDB, Kafka, and Parquet Files. At the last stage in our data pipeline, visualizations are performed on the data in the output sink using MongoDB Atlas. Below is a visual representation of our data pipeline.



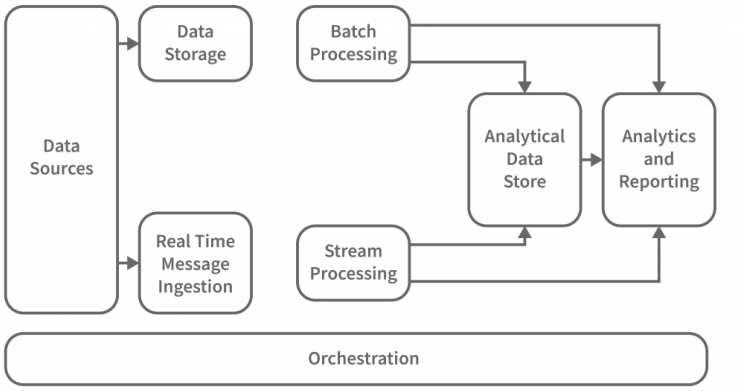
What is Big Data?

"*What typically defines big data is the need for new techniques and tools in order to be able to process it. In order to use big data, you need programs which span multiple physical and/or virtual machines working together in concert in order to process all of the data in a reasonable span of time.*"

- Professor Rodriguez, Lecture 1

Why is this a Big Data Project?

Our project follows the Big Data Architecture and has several layers and components to achieve this. Our data source is the GNews API from which we retrieve the article headlines relating to ChatGPT. We use Kafka for our real time message ingestion and we use Spark for the data stream processing. The processed data is then stored in our Data Storage/Analytical Data Store in MongoDB Atlas.



Tools and Technologies

The tools and technologies used in this project include Google Colaboratory, GNews, Apache Kafka, Apache, Spark, TextBlob, Natural Language Toolkit, and MongoDB. They will be discussed in more depth in this section.

Google Colaboratory

This project was built using Google Colaboratory. Google Colaboratory is a Python notebook that allows users to combine Python code, text, and images into a single executable document. Google Colaboratory also makes it very easy to configure virtual python environments that can easily be reconfigured and reset. Last but not least, unlike Jupyter Notebook, Google Colaboratory notebooks can be shared with and edited by multiple people.

GNews (Python Google News API)

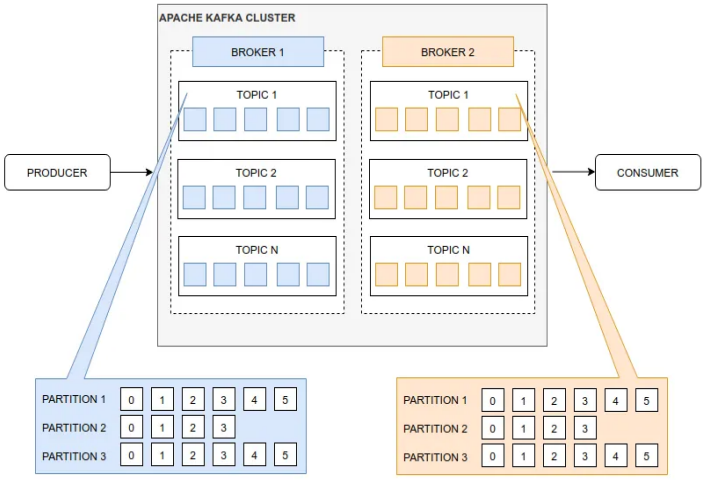
As mentioned previously, GNews is a Google News Python API that retrieves articles that match a user-defined keyword. This API can be queried to retrieve a maximum of 100 articles per request. In our project, that keyword is “GPT.” In this slide, you can see an example of the JSON response retrieved from GNews. The following is an example of the JSON response:

```
{'description': 'The Amazing Ways Duolingo Is Using AI And GPT-4 Forbes',
'published date': 'Fri, 28 Apr 2023 06:31:28 GMT',
'publisher': {'href': 'https://www.forbes.com','title': 'Forbes'},
'title': 'The Amazing Ways Duolingo Is Using AI And GPT-4 - Forbes'
'url': 'https://news.google.com/rss/articles/...'}
```

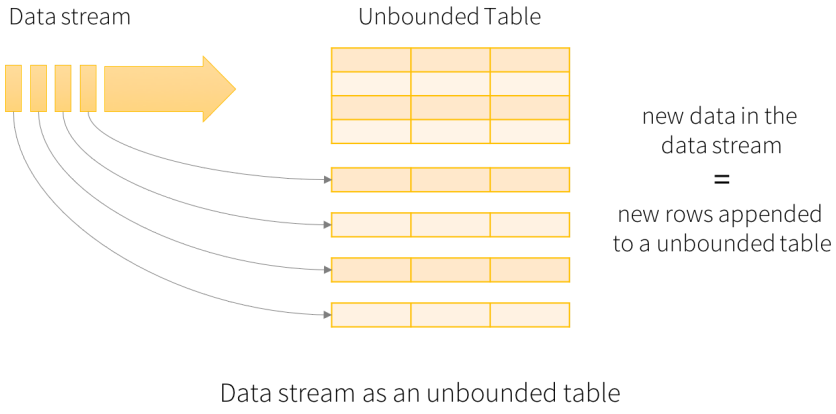
The most important field in this JSON response is “title” which is used to perform sentiment analysis. The other fields include the article's description, publish date, publisher, and website link. In addition to specifying keywords in the GNews API query, the dates between which the article was published can be specified as well. For this project, we want to track the sentiment of ChatGPT-related articles since its launch date of November 30th over a 5-month period ending on April 30th. To retrieve this data from GNews, we sent a request to the GNews API for each day in the 5-month period and incremented the start and end date with each query to the API. After each query, the JSON response was sent to Kafka.

Apache Kafka

Kafka is a distributed messaging system consisting of 4 main components: brokers, topics, producers, and components, the first three being the most important to this project. A Kafka broker is one of many servers in a Kafka cluster. A topic, contained within a broker, acts as a categorical bucket that stores data entries relating to that topic. Topics are divided into partitions that are split across brokers, allowing parallelization of the data. In this project, the Kafka topic was defined as “google-news”. In order to send data from the GNews API to the Kafka topic, we used a producer, which sends each streaming event, or batch or articles, to the "google-news" topic.



Apache Spark Structured Streaming



Data stream as an unbounded table

Spark is an important technology because it is scalable, fault-tolerant, has low latency, and performs operations on the data in distributed batches. As mentioned previously, Kafka acts as a link between GNews and Spark since they cannot interact directly. Once the articles are sent to Kafka, the Kafka data stream is read by Spark into a structured streaming dataframe. Essentially, Spark reads the latest available data from Kafka, updates and appends to the streaming dataframe with each incoming batch of articles, and discards the source data.

Sentiment Analysis: TextBlob and Natural Language Toolkit (NLTK)

When performing sentiment analysis, we explored two possible solutions - using the Textblob library and the Natural Language Toolkit (NLTK) library. Textblob is a library that is built on top of NLTK, offering a simple and intuitive API for common natural language processing tasks. The NLTK library is a comprehensive library for building Natural Language Processing (NLP) applications in Python, offering a wide range of tools and resources for text processing, including tokenization, stemming, lemmatization, and sentiment analysis.

We performed sentiment analysis in the following steps:

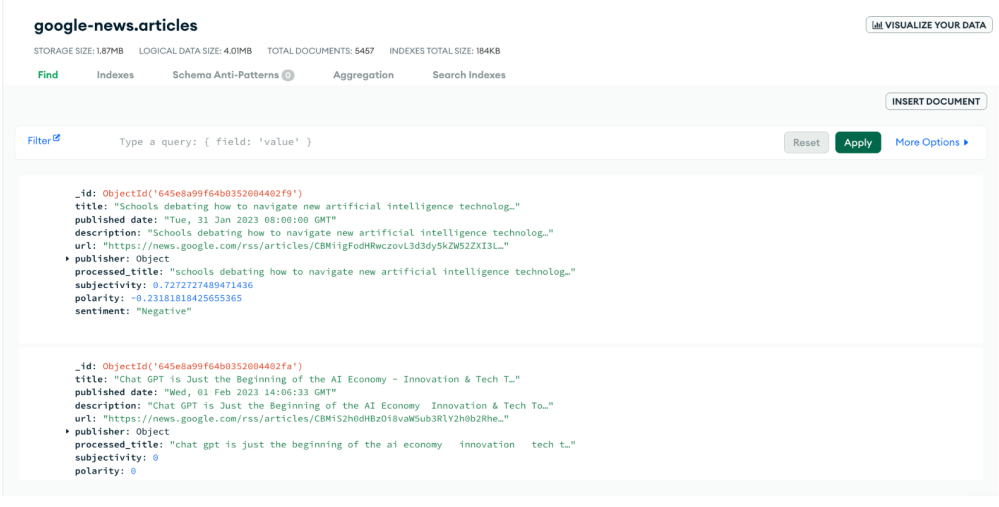
1. Text Preprocessing: The input text (news headlines) were preprocessed to remove noise and irrelevant information. This step involved processes including tokenization, lowercasing, removing punctuation and other unnecessary characters.

2. Sentiment Scoring: Textblob and NLTK provide methods to assign sentiment scores or labels to the preprocessed text. Textblob utilizes a polarity score ranging from [-1, 1] where -1 is a negative sentiment score, 0 is a neutral sentiment score, and 1 is a positive sentiment score. NLTK offers a polarity score as well as several other classifiers. In this project we made use of the polarity score classifier that works similarly to the Textblob classification.

3. Interpretation and Visualization: Once polarity scores and sentiments have been assigned to the new headlines we send it to our output sink in MongoDB Atlas. Visualizations were then derived from our data for further analysis.

Output Sinks: MongoDB Atlas, Kafka, and Parquet Files

To store our transformed data, we used three different output sinks. The first and primary output sink is MongoDB, a document-oriented NoSQL database used for high-volume data storage. In a MongoDB database, each article is represented by a document. Documents are stored within a collection and a collection is stored within a database. In this project, our MongoDB database is called "google-news," the same as the Kafka topic, and the collection name is "articles" because each document in the collection is an article initially retrieved from GNews and later processed with sentiment analysis. Below is a view of the MongoDB Atlas database. The second output sink used was Kafka, which was discussed earlier, and the final output sink used was parquet file storage. A parquet file is a column-oriented data file format that uses a data compression and encoding scheme for efficient data storage and retrieval. It is important to note that parquet files are not human-readable.



Visualizations and Analysis

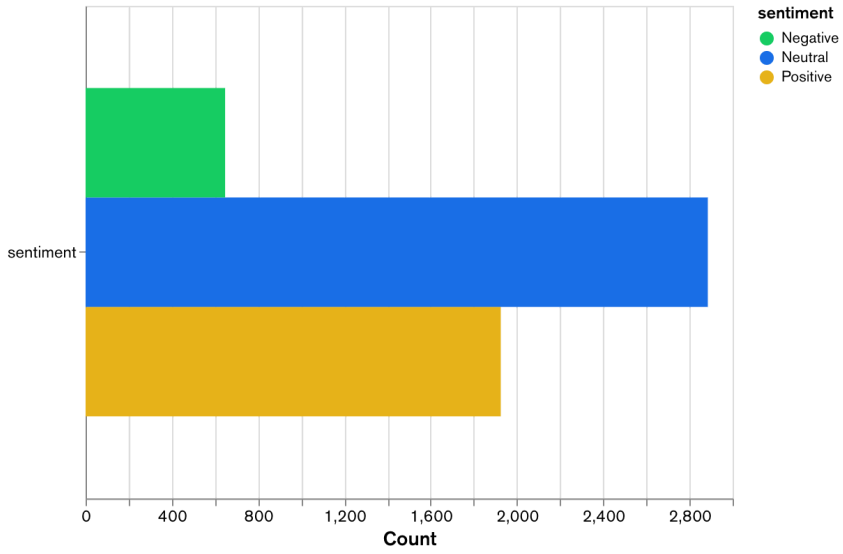
The following visualizations were made using MongoDB Atlas. These visualizations automatically update hourly to account for new articles that may be inserted into the collection. The visualizations shown below can be directly accessed using this link.

There are a total of 5,457 Documents in the MongoDB Atlas collection, 1,925 documents have neutral sentiment: 2885, and 647 documents have negative sentiment.

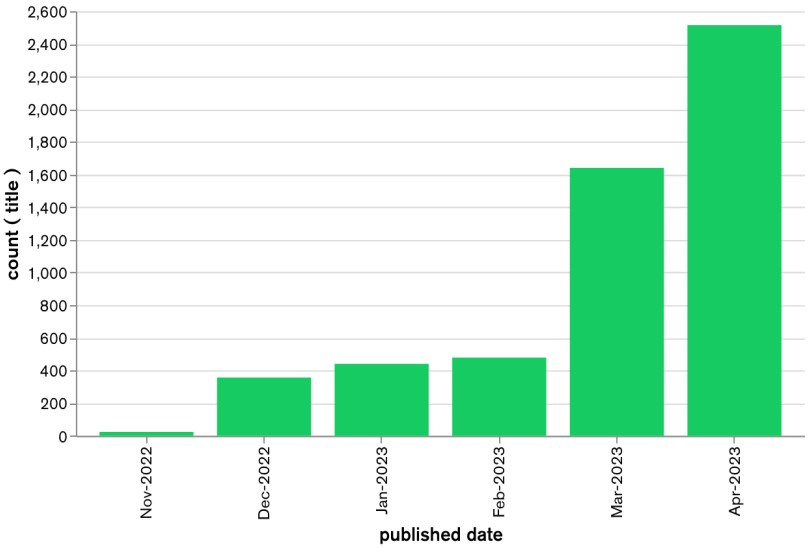
It is important to note that depending on the keyword (topic) used in GNews to extract related articles, the size of the dataset will change. Since we chose to retrieve articles based on ChatGPT, our dataset is limited to 5,457 records. However, if we chose a more popular and older topic, the size of our dataset would change. For example, it is safe to assume that if we were to perform sentiment analysis on Twitter related articles, we would have a higher number of news articles and a more sizable dataset to work with.

Article Sentiment Distribution

This visualization displays the number of article headlines published between November 30th, 2022 and April 30th, 2023 that were classified as having positive, neutral, or negative sentiment. Based on the graph, Chat-GPT related article headlines have predominantly neutral sentiment. There are almost three times the number of positive sentiment article headlines as there are negative sentiment article headlines.



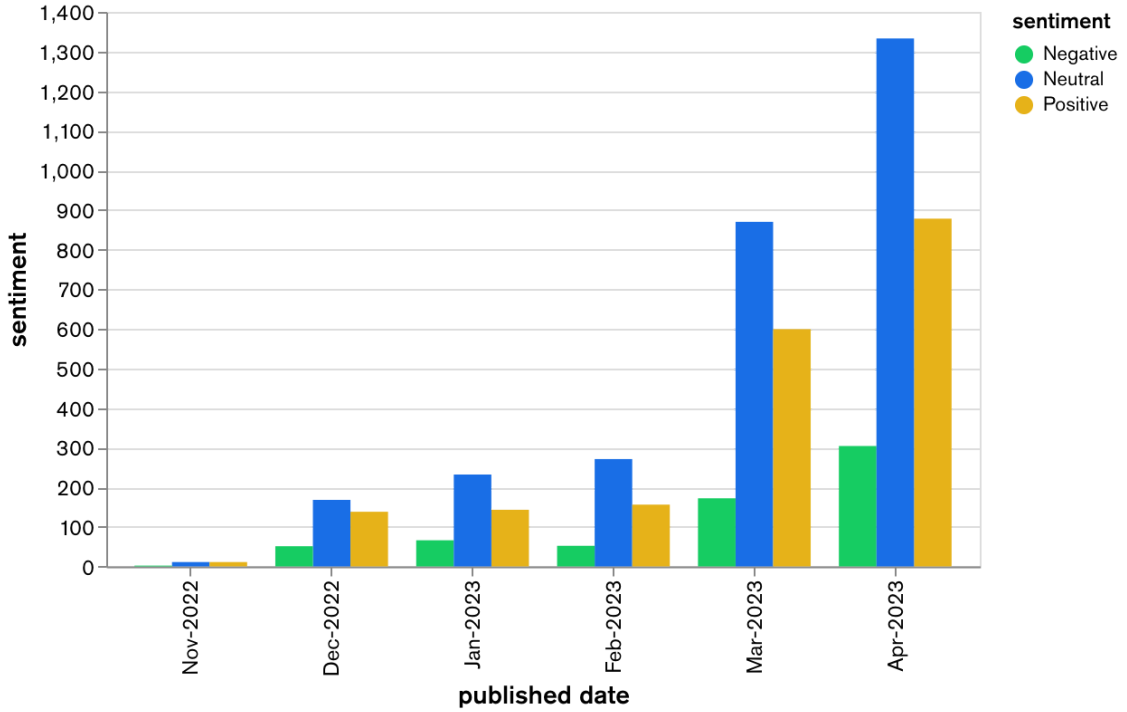
Monthly Article Publishing Distribution

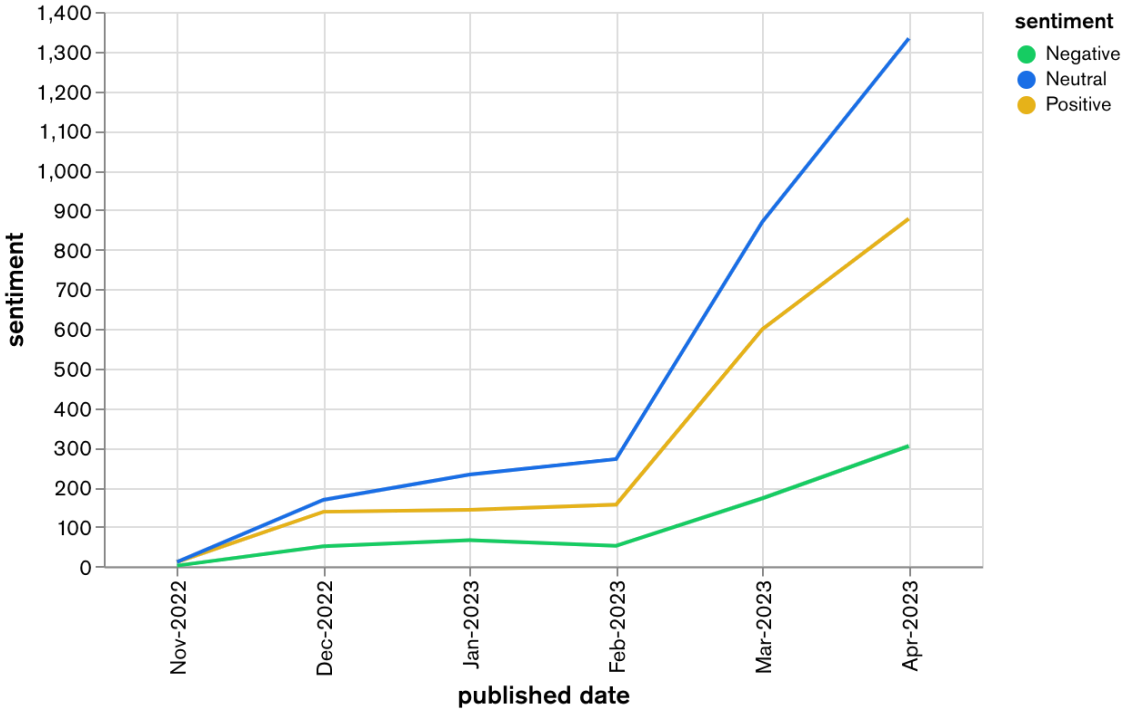


This visualization displays the number of articles published each month between November 30th, 2022 and April 30th, 2023. There was a drastic increase in the number of ChatGPT related articles published in March and April 2023, reflecting ChatGPT's rapid gain in popularity. Conversely, there were very few ChatGPT related articles published in November 2022, which is to be expected because ChatGPT was launched on the last day of November.

Monthly Article Sentiment Distribution

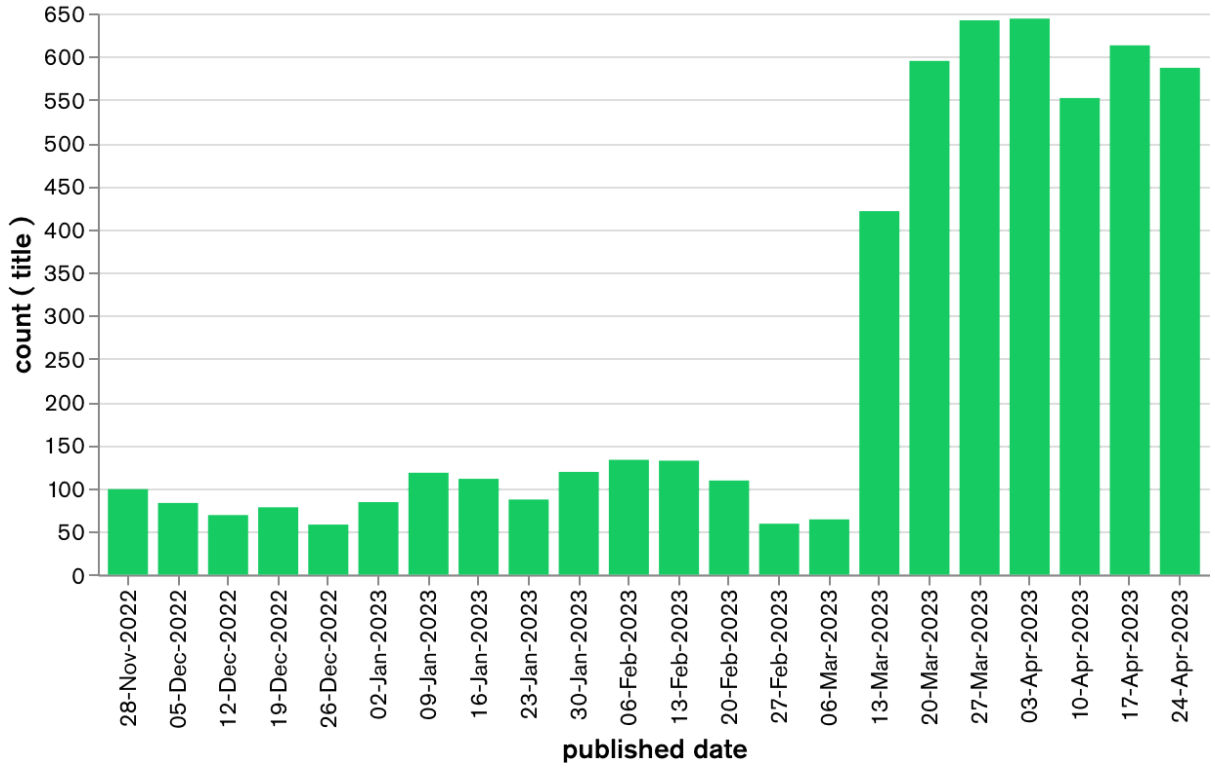
These visualizations display the number of articles published each month between November 30th, 2022 and April 30th, 2023, grouped by sentiment (positive, neutral, negative). The ratio of negative to neutral to positive article sentiment has remained relatively steady over time, despite the gradual increase in the number of articles published. The highest sentiment in article headlines published each month is neutral, followed by positive, and then negative.





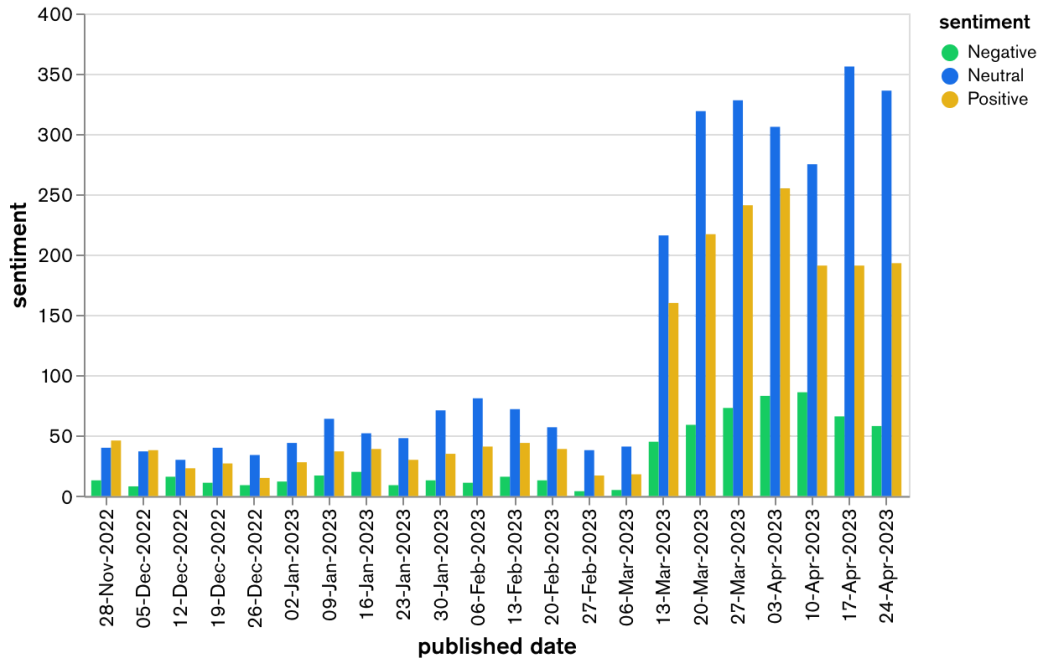
Weekly Article Publishing Distribution

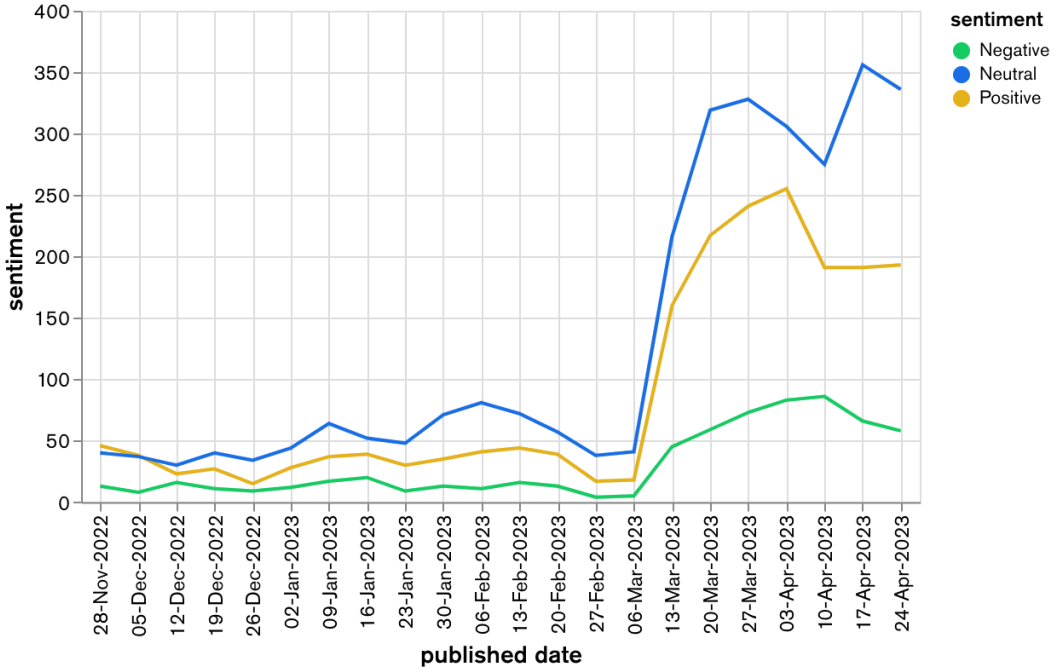
This visualization displays the number of articles published between November 30th, 2022 and April 30th, 2023. There was a drastic increase in the number of ChatGPT related articles published beginning on the Week of March 13th, 2023 and continuing through April 2023, reflecting ChatGPT's rapid gain in popularity. The rate at which articles were published in the previous weeks remained relatively steady.



Weekly Article Sentiment Distribution

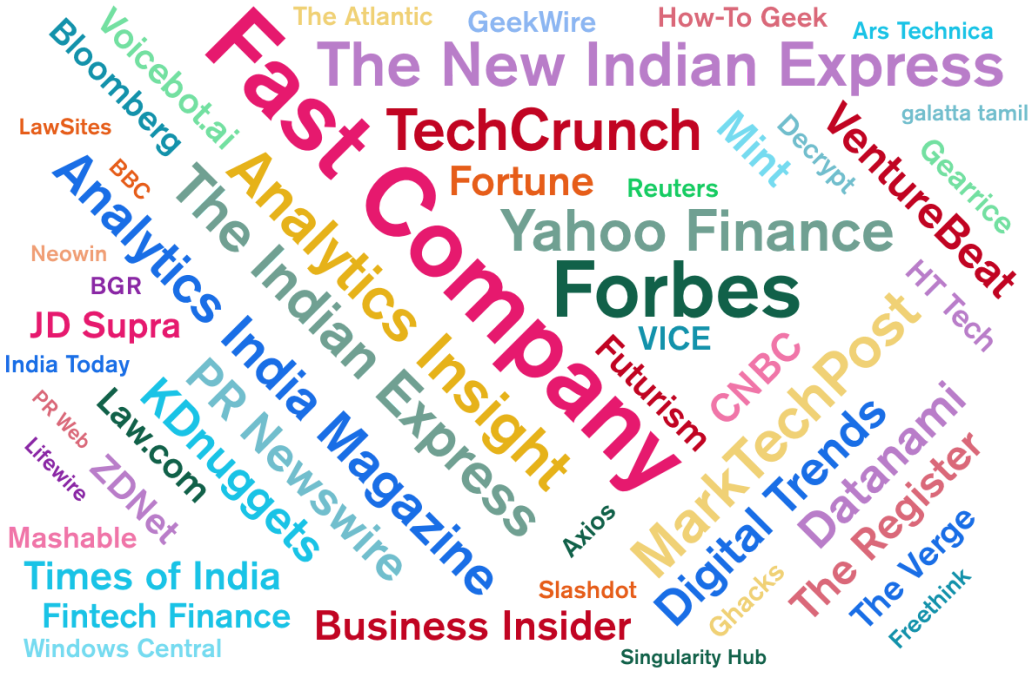
These visualizations display the number of articles published between November 30th, 2022 and April 30th, 2023, grouped by sentiment (positive, neutral, negative). The ratio of negative to neutral to positive article sentiment has remained steady over time, despite the gradual increase in the number of articles published. The highest sentiment in article headlines published each week is neutral, followed by positive, and then negative, except in the first two weeks of ChatGPT's release, November 28th, 2022 and December 5th, 2022, where the highest sentiment in article headlines published each week was positive, followed by neutral, and then negative.

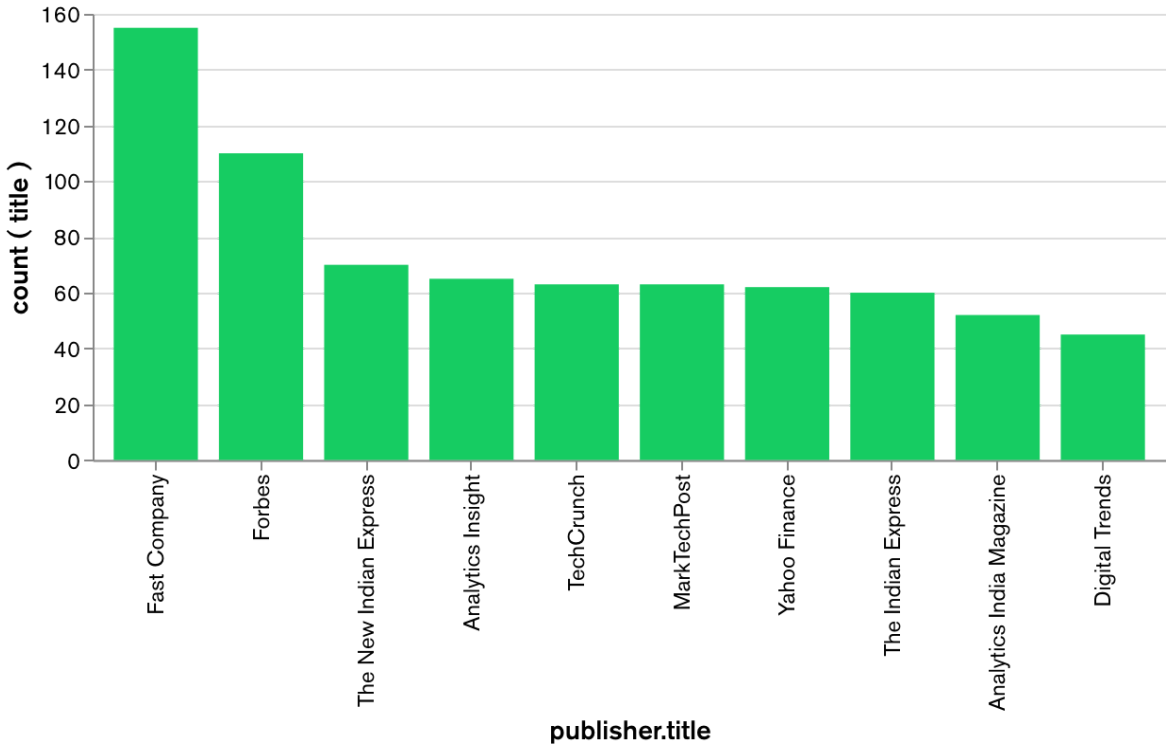




Most Frequent Publishers of ChatGPT Related Articles

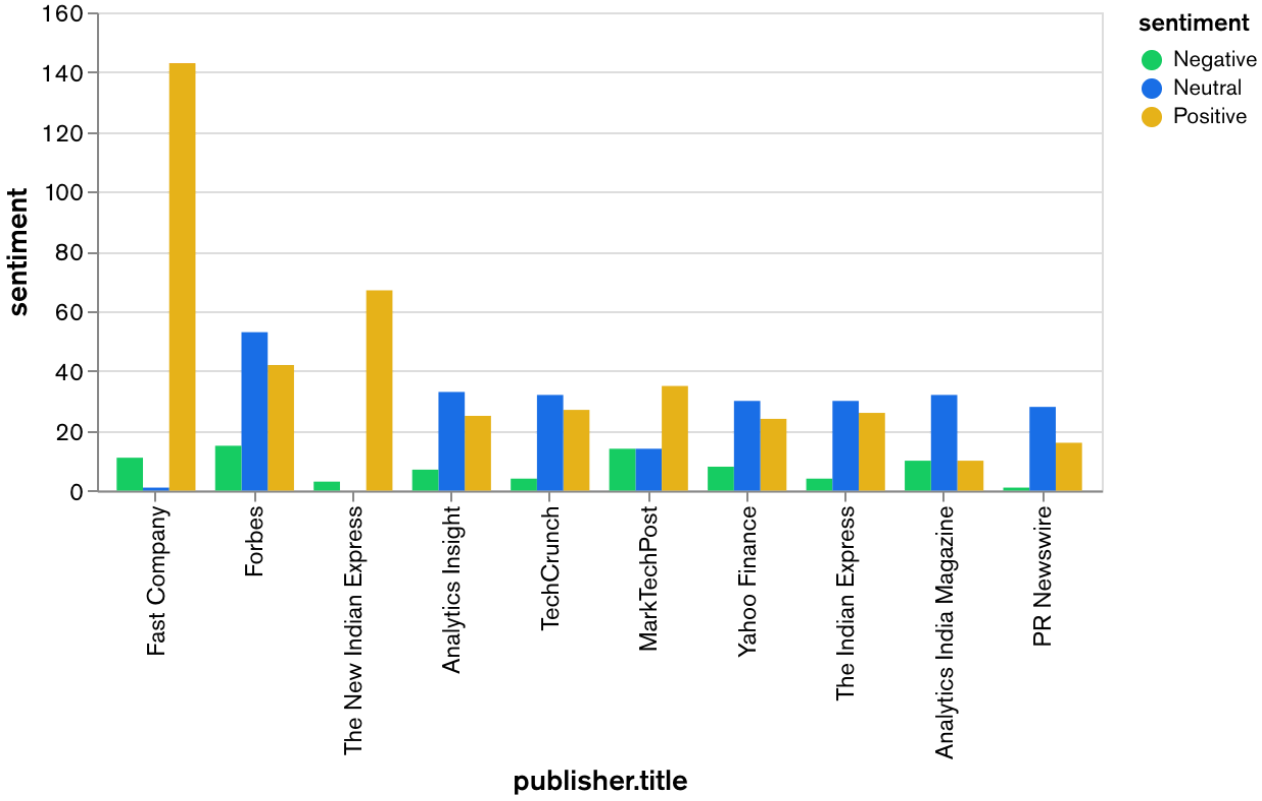
These visualizations display the companies who published articles related to ChatGPT most frequently. The two publishers who posted ChatGPT related articles most frequently were Fast Company and Forbes, respectively. Many of the most frequent publishers of ChatGPT related articles are in the finance, business, and tech industries.





Sentiment Distribution of Most Frequent Publishers of ChatGPT Related Articles

This visualization displays the companies who published articles related to ChatGPT most frequently, grouped by sentiment. The majority of headlines published by Fast Company and The New Indian Express, the first and third most frequent publisher of ChatGPT articles over the 5-month period, were of positive sentiment. The highest sentiment of the articles published by the remaining publishers was neutral.



Conclusion

Our visualizations have demonstrated that the highest sentiment of ChatGPT-related article headlines published over the course of the 5-month period between November 30th, 2022 and April 30th, 2023, is neutral, followed by positive, and then negative sentiment. Furthermore, the ratio of negative to neutral to positive article sentiment has remained steady over time, despite the gradual increase in the number of articles published. This gradual increase changed to a dramatic increase in March and April of 2023.

Overall, our project focuses on creating a big data architecture that is scalable and able to handle large amounts of data in an appropriate manner. By using the GNews API, we were able to attain a large amount of data and process it in a time-efficient way using several new technologies. Our project allows us to perform a sentiment analysis on any news topic and is not limited to ChatGPT, making it easier to visualize and derive insights to understand patterns and trends on a news subject of choice.

References

Ali, M. (2023). *NLTK Sentiment Analysis Tutorial for Beginners*. https://www.datacamp.com/tutorial/text-analytics-beginners-nltk

Apache Kafka. (n.d.). Apache Kafka. https://kafka.apache.org/intro

Cordon, T. (2022, March 30). Enabling streaming data with Spark Structured Streaming and Kafka. *Medium*. https://medium.com/data-arena/enabling-streaming-data-with-spark-structured-streaming-and-kafka-93ce91e5b435

Dominguez, H. R. (2022, May 14). Twitter sentiment analysis using Zookeeper, Kafka and PySpark live-streaming on Windows 10 in 2022. *Medium*. https://medium.com/mcd-unison/twitter-sentiment-analysis-using-zookeeper-kafka-and-pyspark-live-streaming-on-windows-10-in-2022-ada7757097a2

Gongang, L. (2022a, March 12). Apache Spark Structured Streaming - Lorena Gongang - Medium. *Medium*. https://medium.com/@lorenagongang/apache-spark-structured-streaming-69f06c490d8c

Gongang, L. (2022b, March 19). Sentiment analysis on streaming Twitter data using Kafka, Spark Structured Streaming & Python (Part 2). *Medium*. https://medium.com/@lorenagongang/sentiment-analysis-on-streaming-twitter-data-using-kafka-spark-structured-streaming-python-part-b27aecca697a

Gongang, L. (2022c, March 26). Sentiment analysis on streaming Twitter data using Kafka, Spark Structured Streaming & Python (Part 3). *Medium*. https://medium.com/@lorenagongang/sentiment-analysis-on-streaming-twitter-data-using-kafka-spark-structured-streaming-python-part-eaa9f0af076d

Gr, J. (2022, October 25). It’s time to understand streaming DataFrames ! - Javier Gr - Medium. *Medium*. https://medium.com/@JavierGr/its-time-to-understand-streaming-dataframes-f10624ceea19

Gr, J. (2023, April 2). Building a Streaming Data Pipeline With Kafka And Spark. *Medium*. https://blog.devgenius.io/building-a-streaming-data-pipeline-on-ubuntu-20-04-8fa9e6f9cced

Jaiswal, A. (2022). Spark Data Streaming with MongoDB. *Analytics Vidhya*. https://www.analyticsvidhya.com/blog/2022/04/spark-data-streaming-with-mongodb/

Kulhan, C. (2022, January 15). How to use PySpark Streaming with Google Colaboratory. *Medium*. https://che-kulhan.medium.com/how-to-use-pyspark-streaming-with-google-colaboratory-d08ded30cabf

MongoDB. (n.d.). *What Is Big Data Architecture?* https://www.mongodb.com/big-data-explained/architecture

Parmar, A. (2022, January 4). Handling real-time Kafka data streams using PySpark. *Medium*. https://medium.com/@aman.parmar17/handling-real-time-kafka-data-streams-using-pyspark-8b6616a3a084

Shaaban, A. (2021, December 25). Building a data pipeline using BeautifulSoup, Apache Kafka, Apache Spark Streaming and MySQL. *Medium*. https://ahmedshaaban1999.medium.com/building-a-data-pipeline-using-beautifulsoup-apache-kafka-apache-spark-streaming-and-mysql-403cd415d46c

Stamatelou, E. (2021, December 25). Sentiment analysis on streaming Twitter data using Spark Structured Streaming & Python. *Medium*. https://towardsdatascience.com/sentiment-analysis-on-streaming-twitter-data-using-spark-structured-streaming-python-fc873684bfe3

Structured Streaming Programming Guide - Spark 3.4.0 Documentation. (n.d.). https://spark.apache.org/docs/latest/structured-streaming-programming-guide.html

Walters, R. (2020, October 9). Getting started with MongoDB, PySpark, and Jupyter Notebook. *MongoDB*. https://www.mongodb.com/blog/post/getting-started-with-mongodb-pyspark-and-jupyter-notebook

Walters, R. (2022, May 5). *Streaming Data with Apache Spark and MongoDB | MongoDB*. https://www.mongodb.com/developer/languages/python/streaming-data-apache-spark-mongodb/