PROJECT WHITE PAPER

Real-Time Big Data Streaming with Kafka

Abstract

In today's data-driven world, data streaming has emerged as an essential element for businesses worldwide.

Edris Safari

DSC680 Project2, Milestone3

Contents

Figures

Business Problem

To stay competitive, companies must use data to gauge the health of their business and find ways to improve it. The business problems that lie data streaming can solve are numerous. Here are a few:

1. Real-time decision making: Live data streaming enables businesses to make informed decisions in real-time based on the latest data. For example, a stock trading company can use real-time streaming data to make quick decisions on which stocks to buy or sell.
2. Improved customer engagement: Live data streaming can help businesses to better engage with their customers by providing them with real-time updates and personalized recommendations. For instance, a sports streaming platform can offer real-time scores, personalized recommendations, and related content to keep the viewers engaged. It can also be used to strategize the game by the coaching staff.
3. Predictive analytics: Live data streaming can be used for predictive analytics to identify trends and patterns in the data as they occur in real-time. This can help businesses to make more accurate forecasts, reduce risk, and improve operational efficiency.
4. Fraud detection and prevention: Live data streaming can be used to detect and prevent fraud in real-time by monitoring transactions and identifying any anomalies or suspicious activity. For example, if a shopper purchases something very expensive and something that doesn't match previous purchases.
5. IoT device management: Live data streaming is essential for managing IoT devices, which generate large amounts of data in real-time. Streaming data enables businesses to monitor and manage their IoT devices in real-time, identify any issues, and optimize performance.

Overall, live data streaming enables businesses to stay agile, make informed decisions, and respond quickly to changing market conditions, giving them a competitive edge.

Background/History

The goal of this project is to demonstrate a data streaming environment where data is live fed to an application that will consume the received data.

Datasets

The dataset from yfinance will have values for Date, Open, High, Low, Close, Adjusted Close, and Volume. The flight data set has over 40 columns from source to destination to departure delay, arrival delay, duration of flight, flight number, etc.

Data Explanation

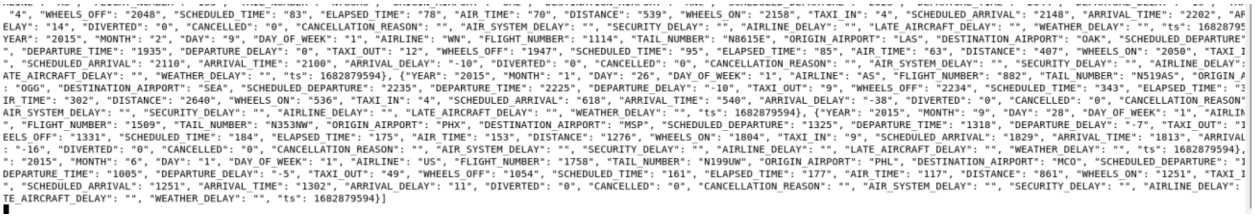
For this phase of the project, we will use two datasets. One is historical data augmented with live data from the stock market. The second dataset is flight data obtained from Kaggle. The producer sends flight data sent from csv files to consumers. The consumer of flight data computes the number of flights per day for each airline it receives data for. We set up several consumers against each producer.

Data Preparation

Data is packaged from a data frame which is tabular to a dictionary which is name/value in by the producer of data. The receiver transforms the dictionary into a data frame and processes the data.

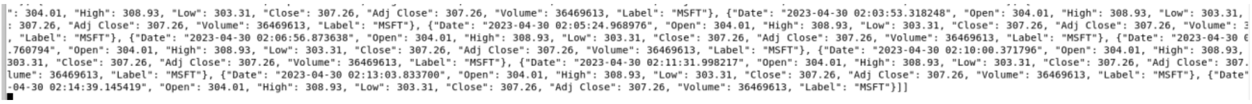
Methods

We will use Python as the primary programming language. We will use Matplotlib, Seaborn, Keras, Pandas, yfinance and other necessary libraries. Apache Kafka will be used to facilitate live data streaming. We compute and display technical indicators for individual stocks. We display flight data for the flight data dataset. Using a local Kafka server, we will send data and monitor the server during the operation. Pictures below show the activity on the server. Each of the Stock and Flight data streams use their respective Topics. Topics are names by which the kafka application distinguishes which stream to forward to which topic. Consumers connect to topics and producers send to topic.



```
 "4", "WHEELS_OFF": "2048", "SCHEDULED_TIME": "83", "ELAPSED_TIME": "78", "AIR_TIME": "70", "DISTANCE": "539", "WHEELS_ON": "2158", "TAXI_IN": "4", "SCHEDULED_ARRIVAL": "2148", "ARRIVAL_TIME": "2202", "AF
ELAY": "14", "DIVERTED": "0", "CANCELLED": "0", "CANCELLATION_REASON": "", "AIR_SYSTEM_DELAY": "", "SECURITY_DELAY": "", "AIRLINE_DELAY": "", "LATE_AIRCRAFT_DELAY": "", "WEATHER_DELAY": "", "ts": 16828795
YEAR": "2015", "MONTH": "2", "DAY": "9", "DAY_OF_WEEK": "1", "AIRLINE": "WN", "FLIGHT_NUMBER": "1114", "TAIL_NUMBER": "N8615E", "ORIGIN_AIRPORT": "LAS", "DESTINATION_AIRPORT": "OAK", "SCHEDULED_DEPARTURE"
", "DEPARTURE_TIME": "1935", "DEPARTURE_DELAY": "0", "TAXI_OUT": "12", "WHEELS_OFF": "1947", "SCHEDULED_TIME": "95", "ELAPSED_TIME": "85", "AIR_TIME": "63", "DISTANCE": "407", "WHEELS_ON": "2050", "TAXI_I
", "SCHEDULED_ARRIVAL": "2110", "ARRIVAL_TIME": "2100", "ARRIVAL_DELAY": "-10", "DIVERTED": "0", "CANCELLED": "0", "CANCELLATION_REASON": "", "AIR_SYSTEM_DELAY": "", "SECURITY_DELAY": "", "AIRLINE_DELAY":
ATE_AIRCRAFT_DELAY": "", "WEATHER_DELAY": "", "ts": 1682879594}, {"YEAR": "2015", "MONTH": "1", "DAY": "26", "DAY_OF_WEEK": "1", "AIRLINE": "AS", "FLIGHT_NUMBER": "882", "TAIL_NUMBER": "N519AS", "ORIGIN_A
: "OGG", "DESTINATION_AIRPORT": "SEA", "SCHEDULED_DEPARTURE": "2235", "DEPARTURE_TIME": "2225", "DEPARTURE_DELAY": "-10", "TAXI_OUT": "9", "WHEELS_OFF": "2234", "SCHEDULED_TIME": "343", "ELAPSED_TIME": "3
IR_TIME": "302", "DISTANCE": "2640", "WHEELS_ON": "536", "TAXI_IN": "4", "SCHEDULED_ARRIVAL": "618", "ARRIVAL_TIME": "540", "ARRIVAL_DELAY": "-38", "DIVERTED": "0", "CANCELLED": "0", "CANCELLATION_REASON"
AIR_SYSTEM_DELAY": "", "SECURITY_DELAY": "", "AIRLINE_DELAY": "", "LATE_AIRCRAFT_DELAY": "", "WEATHER_DELAY": "", "ts": 1682879594}, {"YEAR": "2015", "MONTH": "9", "DAY": "28", "DAY_OF_WEEK": "1", "AIRLIN
", "FLIGHT_NUMBER": "1509", "TAIL_NUMBER": "N353NW", "ORIGIN_AIRPORT": "PHX", "DESTINATION_AIRPORT": "MSP", "SCHEDULED_DEPARTURE": "1325", "DEPARTURE_TIME": "1318", "DEPARTURE_DELAY": "-7", "TAXI_OUT": "1
EELS_OFF": "1331", "SCHEDULED_TIME": "184", "ELAPSED_TIME": "175", "AIR_TIME": "153", "DISTANCE": "1276", "WHEELS_ON": "1804", "TAXI_IN": "9", "SCHEDULED_ARRIVAL": "1829", "ARRIVAL_TIME": "1813", "ARRIVAL
: "-16", "DIVERTED": "0", "CANCELLED": "0", "CANCELLATION_REASON": "", "AIR_SYSTEM_DELAY": "", "SECURITY_DELAY": "", "AIRLINE_DELAY": "", "LATE_AIRCRAFT_DELAY": "", "WEATHER_DELAY": "", "ts": 1682879594},
": "2015", "MONTH": "6", "DAY": "1", "DAY_OF_WEEK": "1", "AIRLINE": "US", "FLIGHT_NUMBER": "1758", "TAIL_NUMBER": "N199UW", "ORIGIN_AIRPORT": "PHL", "DESTINATION_AIRPORT": "MCO", "SCHEDULED_DEPARTURE": "1
DEPARTURE_TIME": "1005", "DEPARTURE_DELAY": "-5", "TAXI_OUT": "49", "WHEELS_OFF": "1054", "SCHEDULED_TIME": "161", "ELAPSED_TIME": "177", "AIR_TIME": "117", "DISTANCE": "861", "WHEELS_ON": "1251", "TAXI_I
", "SCHEDULED_ARRIVAL": "1251", "ARRIVAL_TIME": "1302", "ARRIVAL_DELAY": "11", "DIVERTED": "0", "CANCELLED": "0", "CANCELLATION_REASON": "", "AIR_SYSTEM_DELAY": "", "SECURITY_DELAY": "", "AIRLINE_DELAY":
TE_AIRCRAFT_DELAY": "", "WEATHER_DELAY": "", "ts": 1682879594}]
```

Figure 1- Flight data streaming activity



```
": 304.01, "High": 308.93, "Low": 303.31, "Close": 307.26, "Adj Close": 307.26, "Volume": 36469613, "Label": "MSFT"}, {"Date": "2023-04-30 02:03:53.318248", "Open": 304.01, "High": 308.93, "Low": 303.31,
: 307.26, "Adj Close": 307.26, "Volume": 36469613, "Label": "MSFT"}, {"Date": "2023-04-30 02:05:24.968976", "Open": 304.01, "High": 308.93, "Low": 303.31, "Close": 307.26, "Adj Close": 307.26, "Volume": 3
, "Label": "MSFT"}, {"Date": "2023-04-30 02:06:56.873638", "Open": 304.01, "High": 308.93, "Low": 303.31, "Close": 307.26, "Adj Close": 307.26, "Volume": 36469613, "Label": "MSFT"}, {"Date": "2023-04-30 0
.760794", "Open": 304.01, "High": 308.93, "Low": 303.31, "Close": 307.26, "Adj Close": 307.26, "Volume": 36469613, "Label": "MSFT"}, {"Date": "2023-04-30 02:10:00.371796", "Open": 304.01, "High": 308.93,
303.31, "Close": 307.26, "Adj Close": 307.26, "Volume": 36469613, "Label": "MSFT"}, {"Date": "2023-04-30 02:11:31.998217", "Open": 304.01, "High": 308.93, "Low": 303.31, "Close": 307.26, "Adj Close": 307.
lume": 36469613, "Label": "MSFT"}, {"Date": "2023-04-30 02:13:03.833700", "Open": 304.01, "High": 308.93, "Low": 303.31, "Close": 307.26, "Adj Close": 307.26, "Volume": 36469613, "Label": "MSFT"}, {"Date"
-04-30 02:14:39.145419", "Open": 304.01, "High": 308.93, "Low": 303.31, "Close": 307.26, "Adj Close": 307.26, "Volume": 36469613, "Label": "MSFT"}]]
```

Figure 2-Stock data streaming activity

Analysis

Below is the description of the results we want to demonstrate.

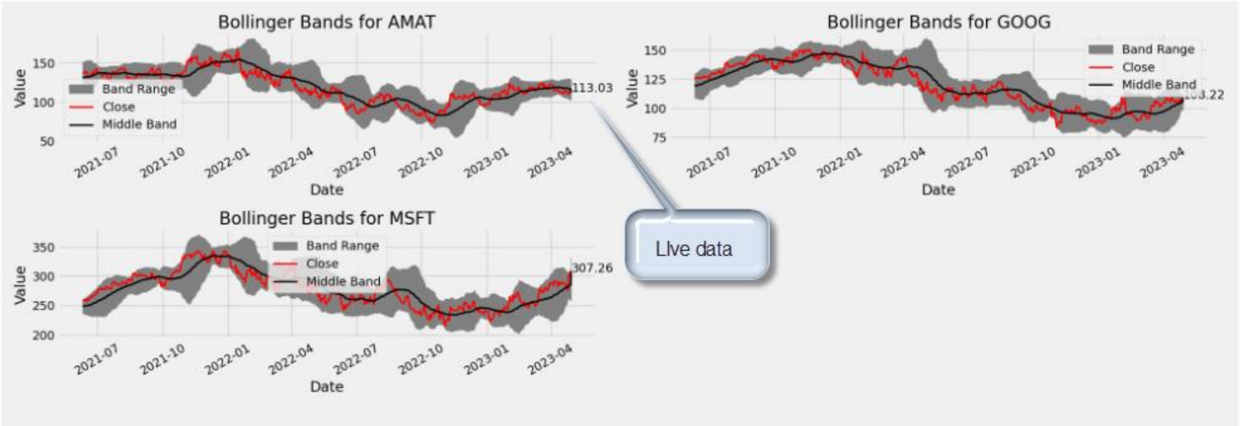
Stock Consumers

Technical Indicators

Technical indicators are intrinsically pattern-based because they are based on time. They are signals that are produced by the price, volume, and other parameters. Shown graphically, these signals will show spike, declines, gradual/sharp decrease, or gradual/sharp increase, etc. These visuals will help technical analysist predict future price movements.

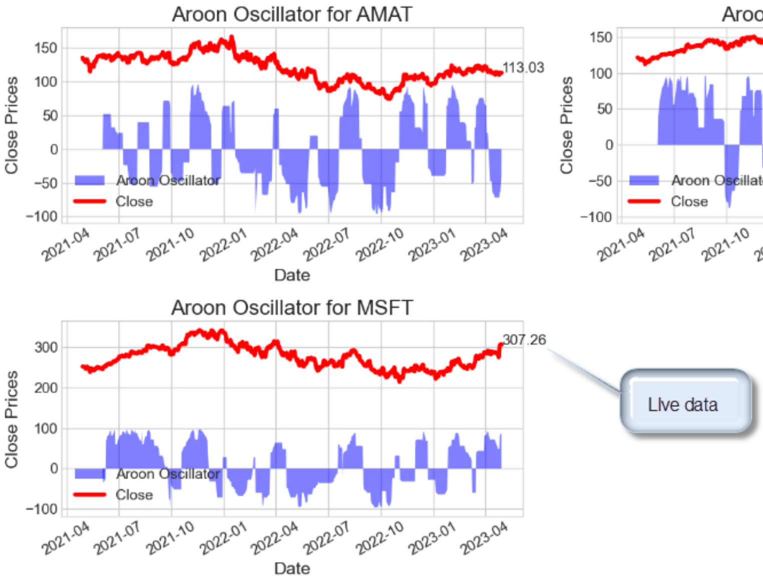
For this demonstration, we created 2 technical indicators. The Bollinger band indicator and Aroon Oscillator. We created two consumers one will plot the Bollinger Bands and the other plots the Aroon Oscillator. This is to demonstrate the architecture's distributed nature.

Pictures below show each consumer's view. The live data is refreshed every 30 seconds.



LIve data

Figure 3- Stock Consumer1



LIve data

Figure 4-Stock Consumer2

Flight Data

Flight data consumer shows number of flights by 3 airlines. The figure below refreshes every 5 seconds.

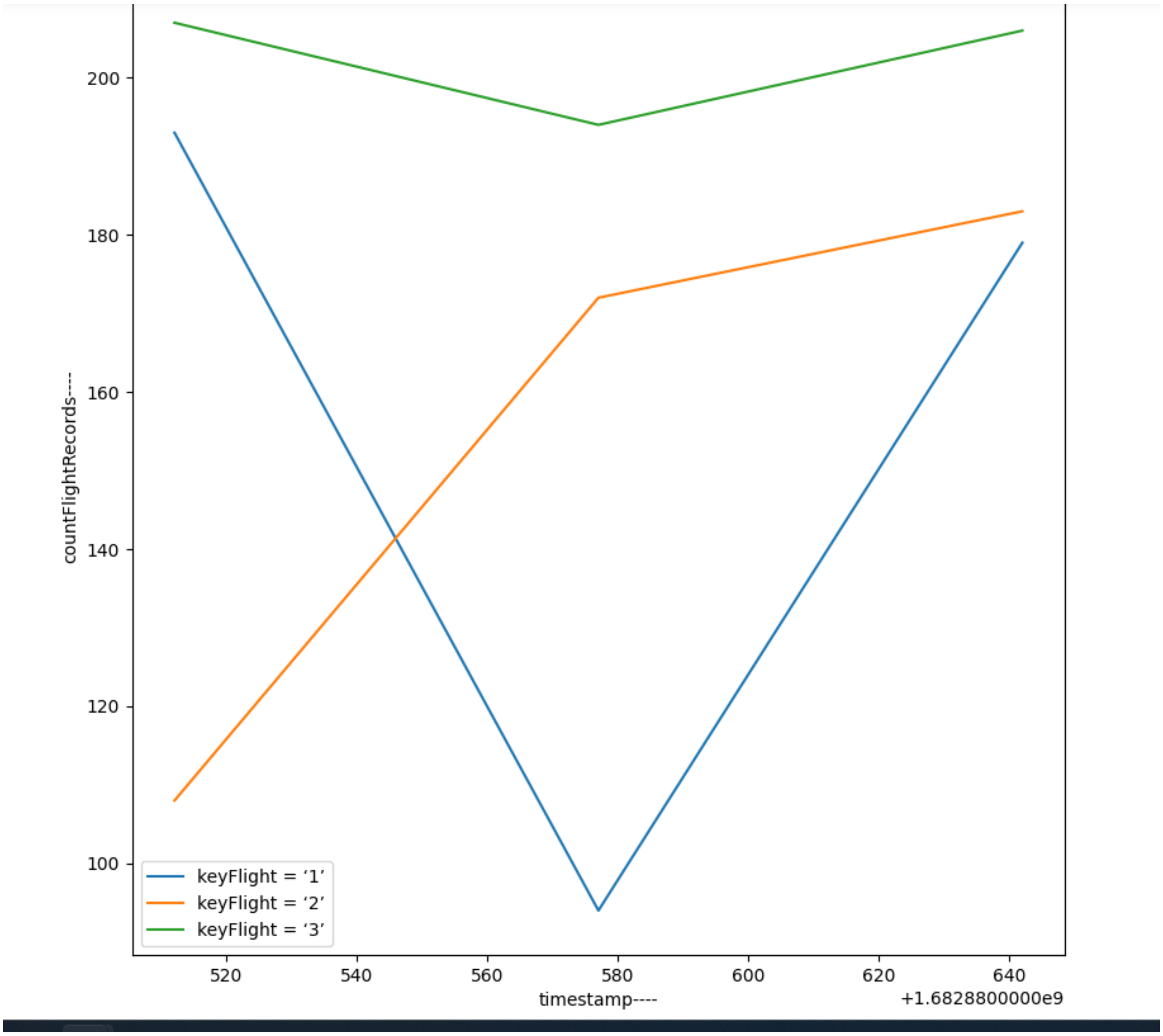


Figure 5-Flight Data Consumer

Kafka

We installed Kafka on a local PC running Linux Ubuntu. We then started the server and started coding the producer and consumer. We monitored the data streaming on the kafka server and verified the receipt of data by the consumer.

Conclusion

We showed that we can use live streaming software package Kafka to package and send data to two receivers-each processing the data to their own requirement. One receiver shows a different graph than the other.

Assumptions

Limitations

One major limiting factor to this project is data quality. We must ensure that the quality of data is monitored and always maintained. Limitations arising from rising volume and speed issues must be considered and mitigated. Another limitation was that during busy hours yfinance would drop connection. Their not-for-free service is uninterruptable.

Challenges

Challenges and issues that we could face in this project must be considered and addressed. This is part of risk assessment that all projects must go through. The table below lists the risks and their mitigation.

Risk	Mitigation
Data Quality	Ensure data quality by performing a preliminary analysis
Data Security	Ensure data is secure both incoming and outgoing. Enable/utilize security measures.
Data Availability	Ensure data is available without interruption or delay that may affect the performance of the analysis
Technical Challenges	Enforce fault tolerance, redundancy, connection integrity
Ethical Violations	Ensure procedures are put in place that will reduce and remove risk of ethical violations by all parties involved.

Future Uses & Product Roadmap

The following improvements will be put on the roadmap.

Increase speed a volume of data

Test the server with increased speed and volume of data. Create a benchmark for hardware sizing.

Use pyspark data streaming

Install and use a local pyspark server and make comparisons with kafka.

Other algorithms.

Use XBoost algorithm for live stock proce prediction.

Recommendations

Use Apache Kafka to send live streaming data and process them. Use more features of Kafka such as segmentation, and control in timing and volume of data.

Implementation Plan

To implement this project, we will perform the following main tasks:

1. Gather and prepare dataset.
2. Prepare Design Document.
3. Code, and test
4. Present
5. Deploy

Ethical Assessment

Based on the ethical considerations we listed in our proposal; we have taken the following measures to address them.

Data privacy: Work with the legal team to ensure that the data collected is obtained in a legal and ethical manner. We will also work with the IT team to ensure that personal information for clients as well as employees is safe and secure.

Bias and fairness: Provide appropriate training about ethics involved in this field to people involved in all aspects of the product.

Transparency: Work with engineering to ensure that the data sources, analysis methods, and findings are transparent and easily understandable to all stakeholders.

Security: We will work with IT on security issues and data privacy as mentioned above.

Informed consent: Work with legal team to make sure that legal documents are in place to align with the customers on topics requiring consent.

Impact on society: Perform a thorough evaluation of the results, perform risk analysis, measure accuracy, and any metrics that can help ascertain minimal negative impact.

References

1. https://blog.logrocket.com/apache-kafka-real-time-data-streaming-app/
2. https://kafka.apache.org/intro
3. https://analyticshut.com/kafka-producer-and-consumer-in-python/
4. https://www.youtube.com/watch?v=tFlYoEJsT2k
5. https://www.youtube.com/watch?v=LjjPjT6R9Bg
6. https://medium.com/@alipazaga07/setting-up-your-first-apache-kafka-development-environment-in-google-cloud-in-15-minutes-ffdb1623e125
7. https://www.learningjournal.guru/courses/kafka/kafka-foundation-training/kafka-in-gcp/
8. https://www.youtube.com/watch?v=RYC-7wECMds&list=PLa7VYi0yPIH14oEOfwbcE9_gM5lOZ4ICN
9. https://spark.apache.org/docs/latest/structured-streaming-programming-guide.html
10. https://phoenixnap.com/kb/install-spark-on-ubuntu

11. https://www.makeuseof.com/pyscript-python-visualizations-web/
12. François Chollet. Deep Learning with Python (Kindle Locations 1504-1508). Manning Publications Co.. Kindle Edition.
13. Project Proposal-Project2_Milestone1_EdrisSafari.pdf