Stock Price Prediction

BIG DATA SPRING 2023

Yashika Khurana (yk2773) Raj Oza (ro2151)

Introduction

The stock market has been an integral part of the global economy for centuries. It serves as a platform for businesses to raise capital and for investors to earn returns on their investments. As such, predicting stock market trends has become a crucial area of interest for investors and financial analysts alike. It is heavily used for:

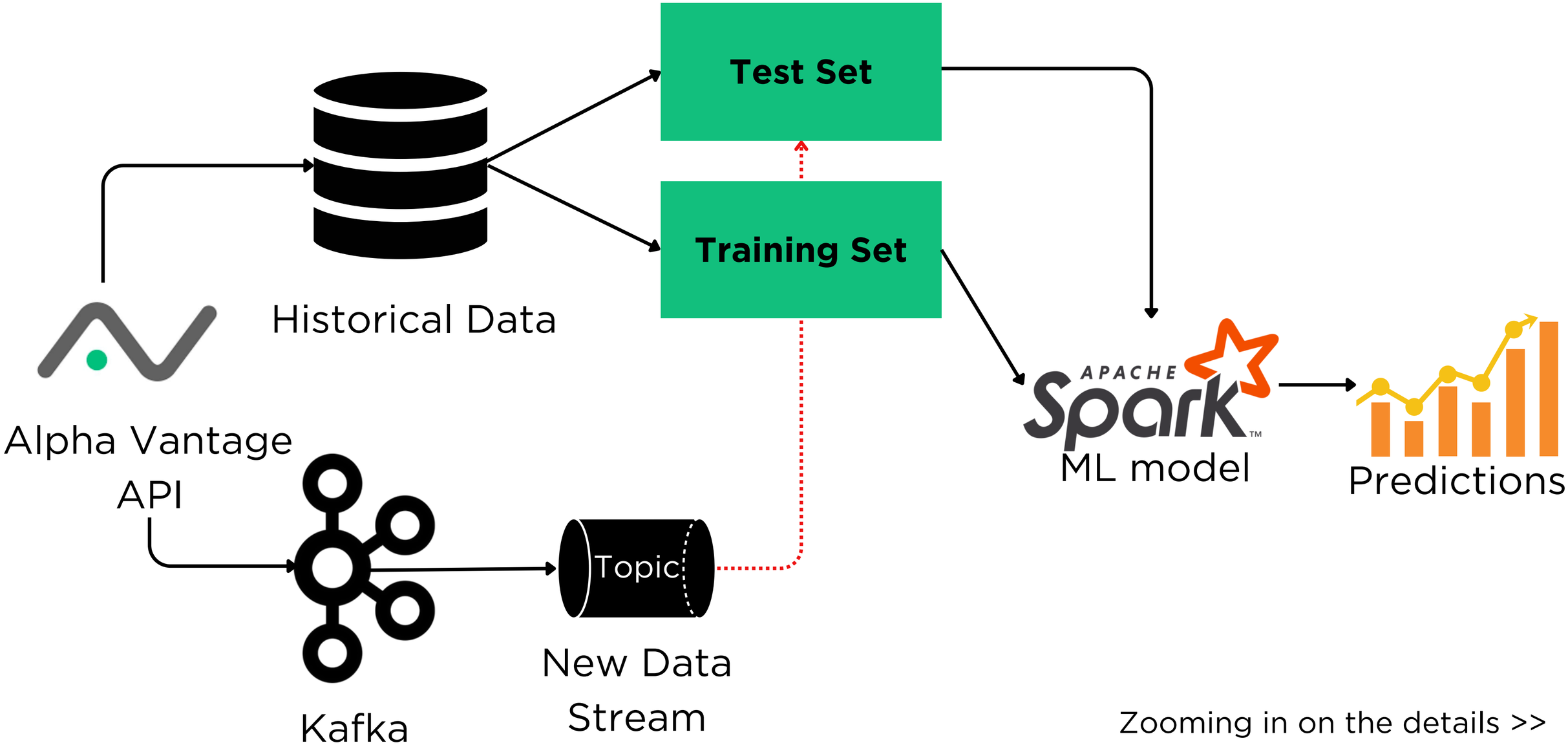
- making investment decisions
- strategising business decisions
- managing risk
- forecasting economy.

Objective

The primary goal of this project is to leverage **Big Data** technology to predict stock prices of 6 major tech. companies & thus curate a solution that can be scaled to use for regularly updated sophisticated and voluminous data, to help thrive in today's dynamic financial market.

The following slides discuss our data extraction methodology, data preprocessing, exploratory data analysis, machine learning model and most importantly, the complete big data pipeline we developed to create a scalable solution for stock prediction.

Pipeline



Zooming in on the details >>

Data

The data has been acquired from the Alpha Vantage API using Python scripts. For the purpose of this project, we collected the TIME_SERIES_DAILY_ADJUSTED data for 6 tech giants going back to 20 years. The data has the following features: timestamp, open, high, low, close, adjusted_close, volume, dividend_amount, split coefficient. Further, the data was split into 80:20 for training & testing.

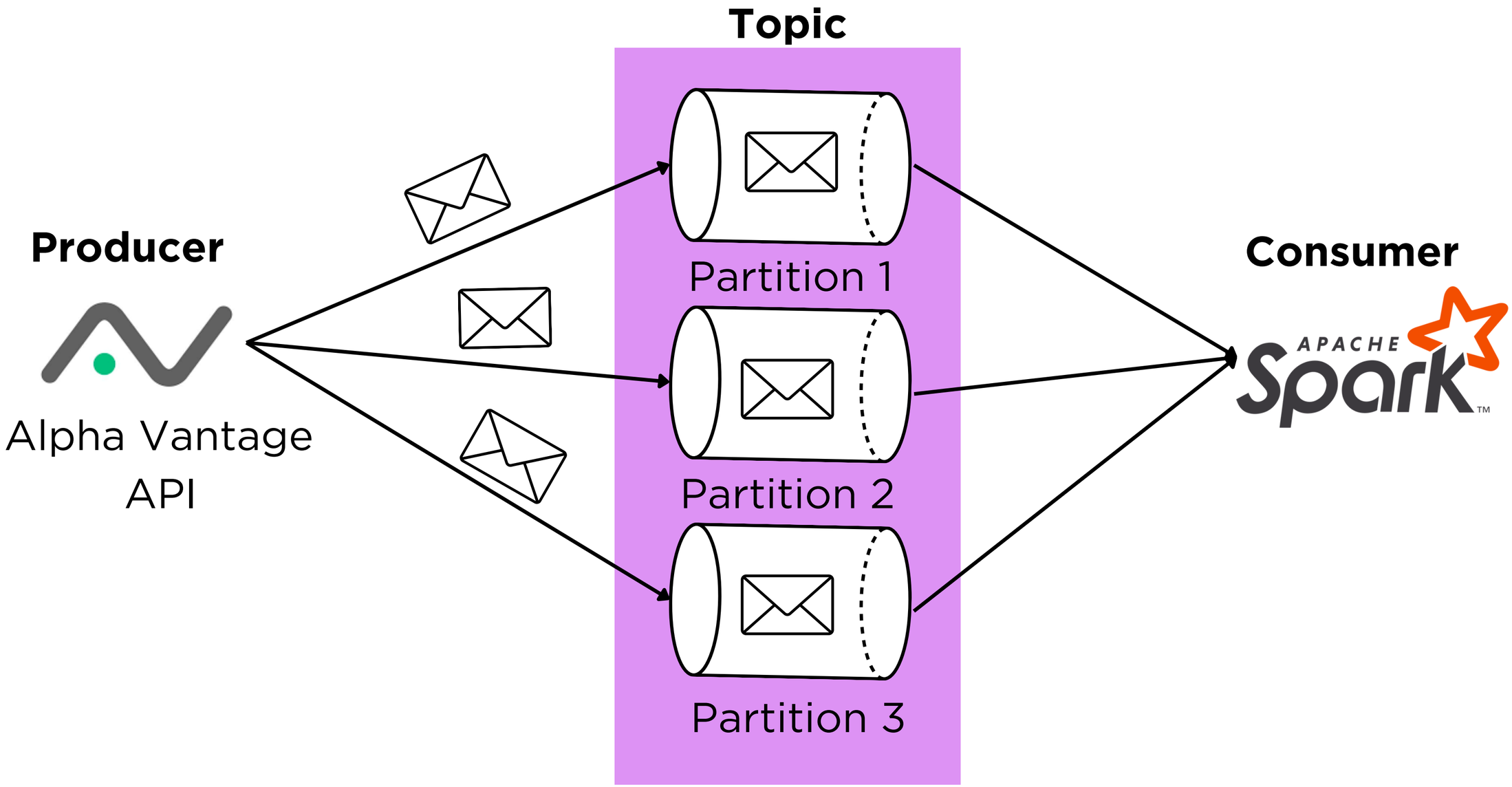


Data Streaming with Kafka

- We set up a Kafka producer that fetches data from the Alpha Vantage API and publishes it to a Kafka topic. This data is then consumed by our Spark ML application, which performs real-time a prediction.

- Kafka streaming allows for the processing of data as soon as it becomes available, without the need to wait for a large batch of data to accumulate. Thus, enabling real time predictions.

Kafka Streaming Architecture



Feature Engineering

We created the following features:

- **daily_return** which computes the percentage change in adjusted_close price from the previous row, which is the daily return.
- **price_range** computes the difference between the high price of the stock during a trading day and the lowest price of a stock during the same day.
- **prev_close** column is derived from the "adjusted_close" column, and represents the adjusted closing price of the stock on the previous day. The lag() function is used with a window specification created using the function to ensure that the lagged value is taken from the previous row in the DataFrame based on the ordering of the "timestamp" column.

Data Preprocessing

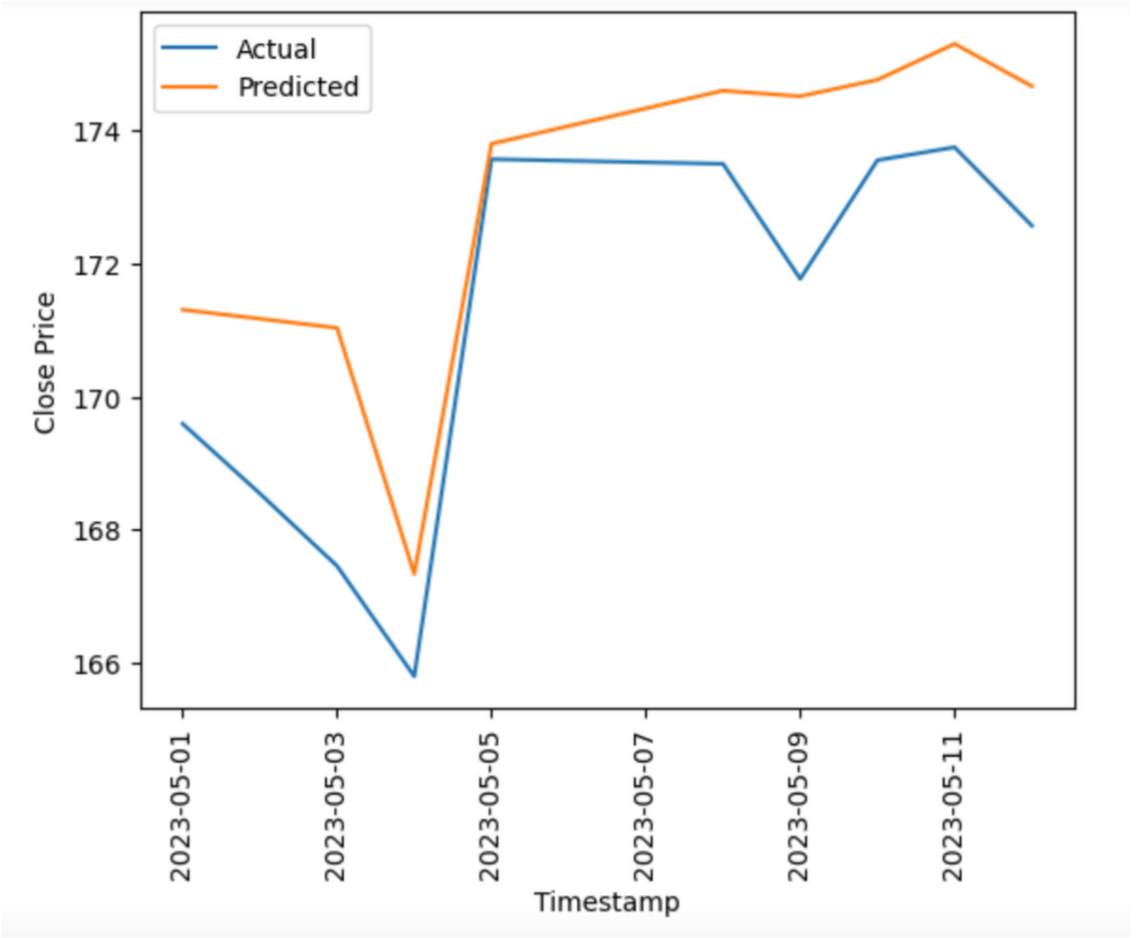
- We used PySpark's VectorAssembler to combine multiple columns in a DataFrame into a single feature vector column.

- In our case, the VectorAssembler is used to create a new column called "features", which is constructed by combining the values from the "open", "high", "low", "prev_close", "daily_return", and "price_range" columns.

- These columns represent different features or variables that are thought to be relevant for predicting the stock price.

Machine Learning with Spark

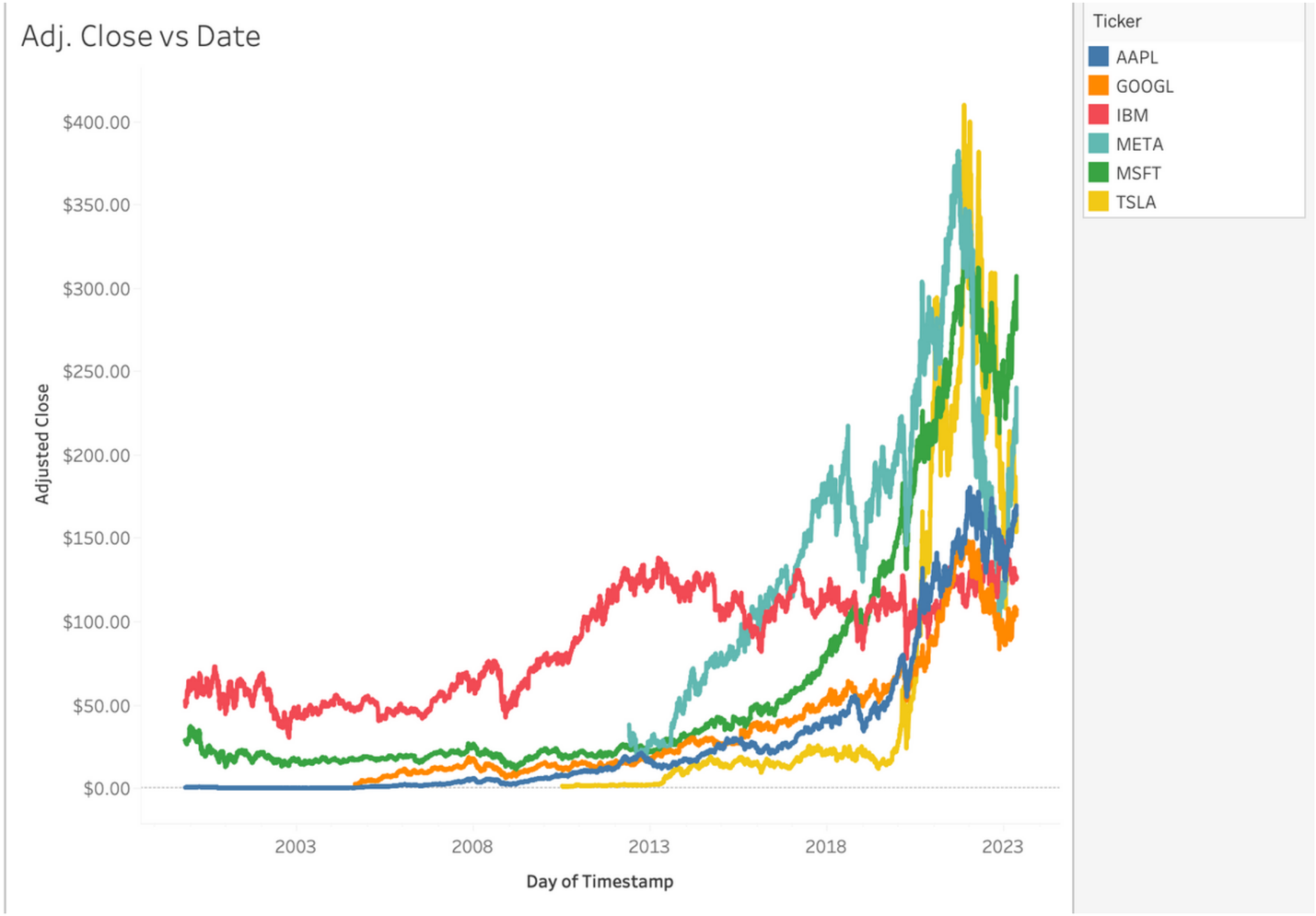
We used SparkML & performed a comparative analysis of various ML models such as linear regression, decision tree, random forests, gradient boosted tree regression, long-short term memory model and evaluated them on the basis of root mean square error and r2-score. Best performance was observed on the linear regression model so we deployed it in our system. We tested the model on streaming data & obtained the plot shown. rmse=2.05



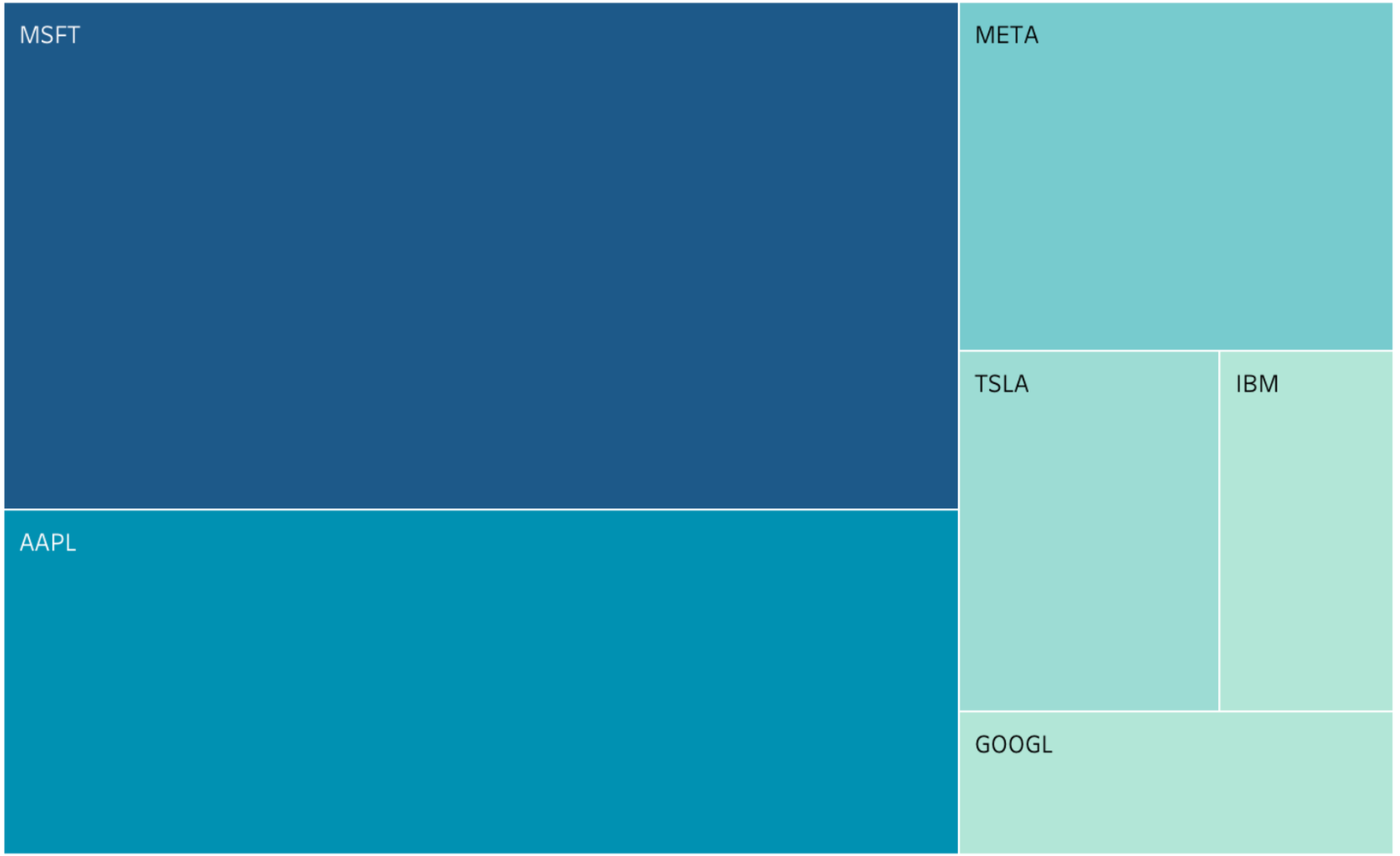
Predictions vs Actual

Exploratory Data Analysis with Tableau

We used Tableau for conducting EDA over data to understand the stock trends.

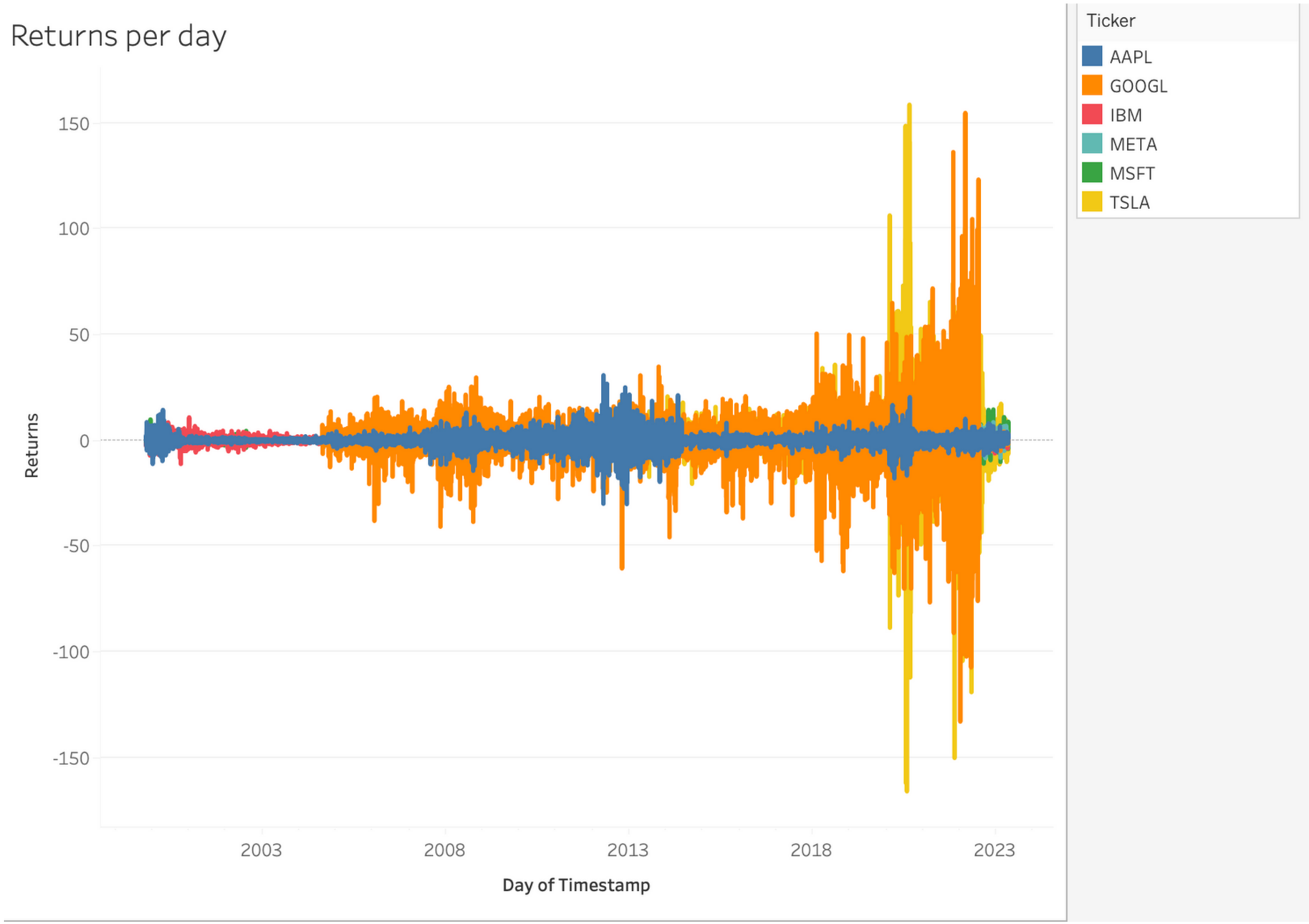


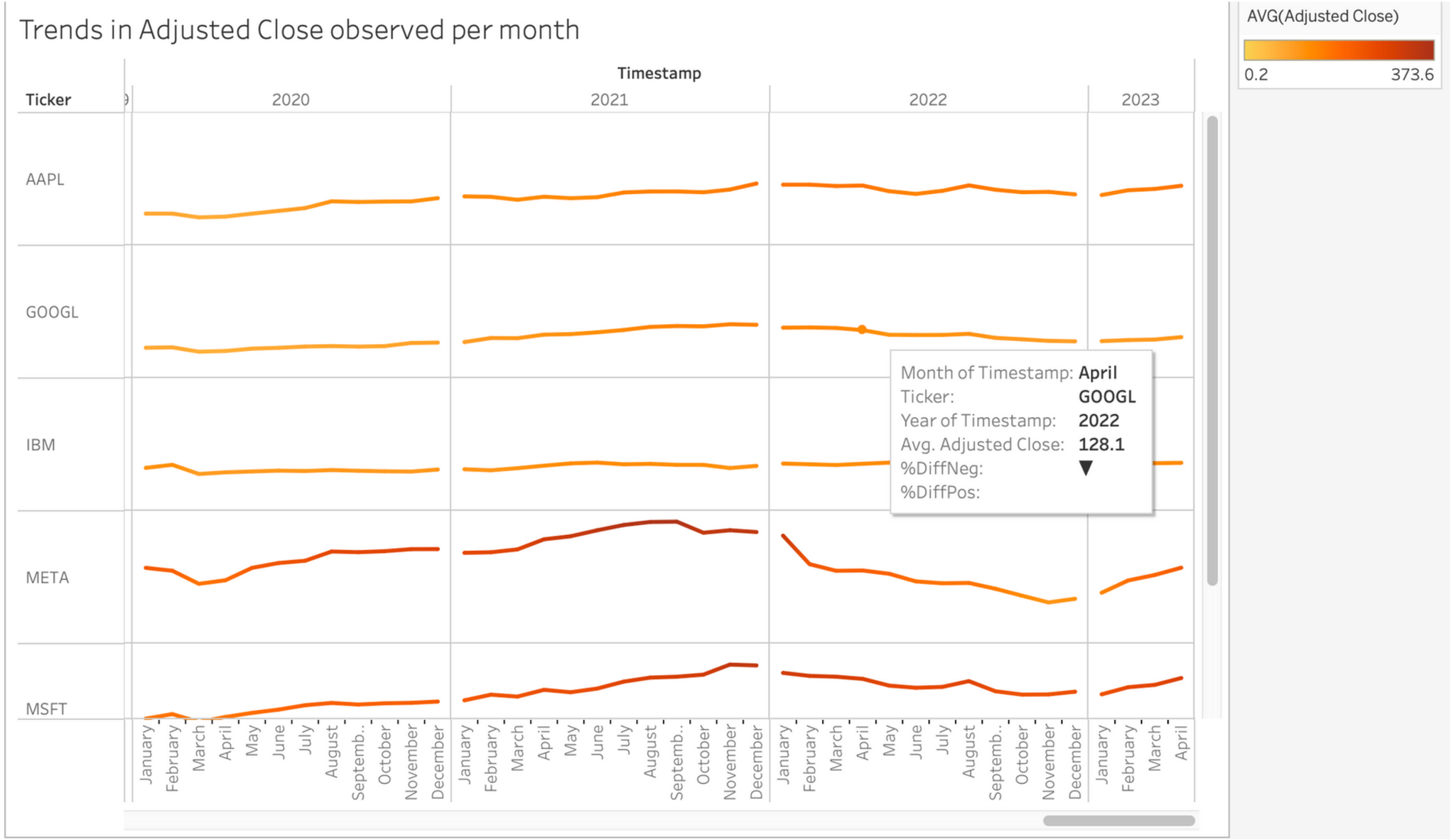
Volume of stocks

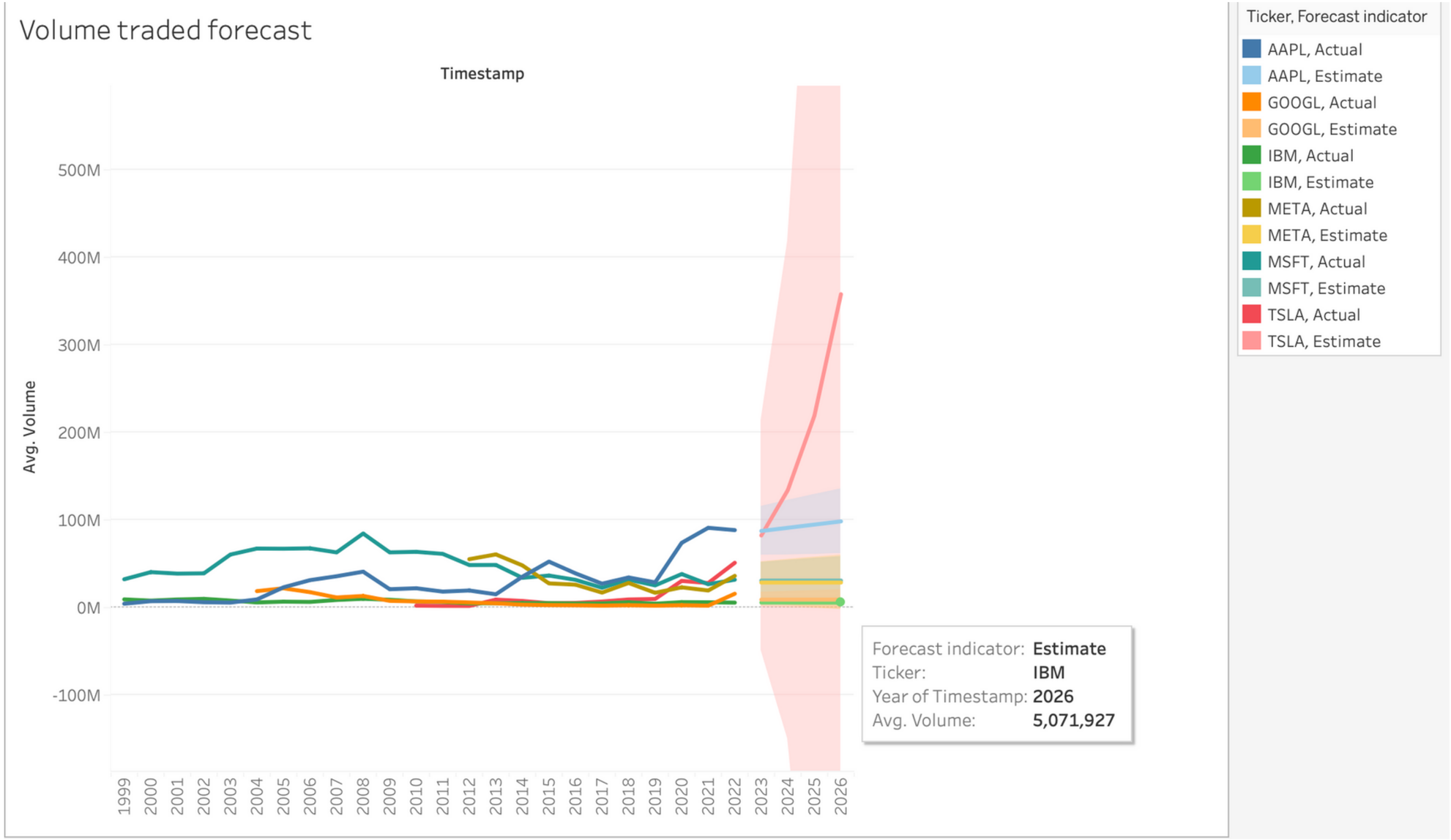


SUM(Volume)

35B 275B


Returns per day
Ticker
AAPL
GOOGL
IBM
META
MSFT
TSLA
150
100
50
0
-50
-100
-150
Returns
2003
2008
2013
2018
2023
Day of Timestamp


Trends in Adjusted Close observed per month
Timestamp
Ticker
2020
2021
2022
2023
AAPL
GOOGL
IBM
META
MSFT
January
February
March
April
May
June
July
August
Septemb..
October
November
December
Month of Timestamp: April
Ticker: GOOGL
Year of Timestamp: 2022
Avg. Adjusted Close: 128.1
%DiffNeg:
%DiffPos:
AVG(Adjusted Close)
0.2
373.6


Volume traded forecast
Timestamp
Avg. Volume
500M
400M
300M
200M
100M
0M
-100M
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
Forecast indicator: Estimate
Ticker: IBM
Year of Timestamp: 2026
Avg. Volume: 5,071,927
Ticker, Forecast indicator
AAPL, Actual
AAPL, Estimate
GOOGL, Actual
GOOGL, Estimate
IBM, Actual
IBM, Estimate
META, Actual
META, Estimate
MSFT, Actual
MSFT, Estimate
TSLA, Actual
TSLA, Estimate

Challenges

We faced certain challenging situations such as:

- Using Kafka with AWS using s3 buckets & Docker

Resolved by running Kafka locally

- Using LSTMs on limited data

Resolved by using alternate ML models

Lessons Learned

Post project completion, we feel confident in the application of:

- ETL pipelining
- Real Time Data Ingestion using APIs
- Local Kafka deployment for data streaming
- PySpark for Machine Learning
- Tableau for exploratory data analysis

Future Scope

We are excited about the prospect of :

- Ameliorating the existing Machine Learning model for an improved performance.
- Considering Twitter based sentiment analysis to predict trends in stocks.
- Providing trading suggestions to users by forecasting on the available data.

Thank you!

We express our sincere gratitude to ***Prof. Juan Rodriguez*** for his constant guidance & invaluable support throughout the course. We look forward to continuous experimentation in the exciting world of Big Data.

NYU