Stock Data Streaming and Analysis

Electric Vehicle Stock

Emmitt Hasty
Data Science
University of Missouri-Kansas City
Missouri
emhrv3@umsystem.edu

Amulya Sara
Computer Science
University of Missouri-Kansas City
Missouri
asc79@umsystem.edu

Gautham Sidhardh Dommeti
Computer Science
University of Missouri-Kansas City
Missouri
gsddbc@umsystem.edu

Snehalatha Mallavarapu
Computer Science
University of Missouri-Kansas City
Missouri
smfyp@umsystem.edu

Naladala Sai Karthik
Computer Science
University of Missouri-Kansas City
Missouri
sndzm@umsystem.edu

ABSTRACT

Data analytics is the process of examining enormous amounts of data to find patterns, insights, and trends. Although data analytics technologies can be applied to a wide range of fields, including healthcare, politics, retail, banking, and government agencies, they are essential for survival in the competitive financial markets of the modern era. Data analytics are essential in finance to understand the ups and downs of capital markets. Great financial data science gives traders and investment advisors the assurance they need to decide for themselves whether to purchase, sell, or hold a certain security. They can use it to manage a portfolio according to short-, mid-, or long-term goals. This live streaming of data from a stock analysis is taken as source data. the data from Forex and Electric Vehicles trading API is streamed through Kafka. After streaming, the data stored in topics is analyzed. The analysis of data is visualized in Jupyter notebook for various use cases. Stocks for Electric vehicle brands are complicated, and insight can be gained from many things, as electric vehicles need special batteries, and valuable insight may be gained from looking into the stock of these companies that make the special batteries. These are companies like Amaraj, or HBL Power. As these stocks become more valuable it can become an indicator that the market for EV's is increasing and will attribute to an increase in stock value for EV's across the board in India. For this paper, we studied three company types, Auto Manufacturing, Battery, and Charging Point. In our study, we predict that increases in the volume of shares per day correlates to an increase in stock prices related to EV's and can be useful for stock trading and investing purposes. Using Kafka, daily streaming data was ingested from AlphaVantage, showing stock data relevant to the daily prices. Then using Spark, the data is loaded into a Jupyter Notebook, with which we can do data analysis on. Plotting the volume of stocks moving day to day can then be used for insight into long term or short-term goals.

INTRODUCTION

There's a Silent EV Revolution Coming in the world.

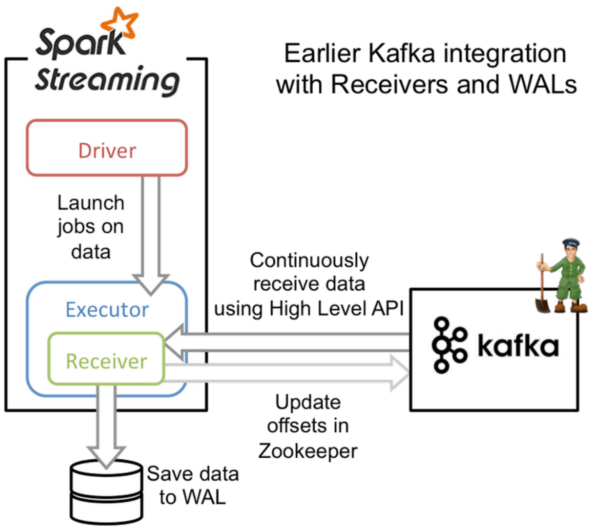
Over the world, electric vehicles are becoming more and more popular. The mass adoption of electric vehicles is being fueled by alarming emission rates and surging gasoline costs, making EV stocks the hottest trend.

It might be a business that manufactures lithium batteries, charging stations, associated components, and more. Due to the industry's youth and quick expansion, investing correctly in such businesses could result in a substantial return.

TOOLS & TECHNOLOGIES

- Requests library,
- Pandas datareader library,
- API : Tiingo API,Alpha Vantage API.
- data visualization tool: matplotlib,Seaborn
- Spark
- Kafka
- Pyspark

ARCHITECTURE

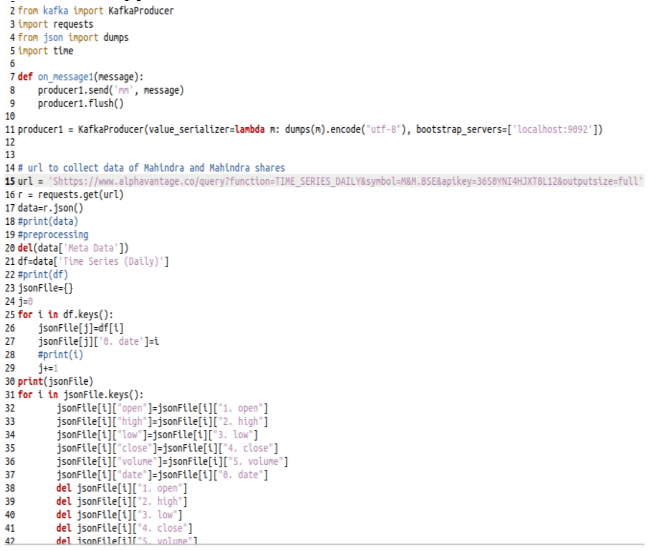

Spark Streaming
Earlier Kafka integration with Receivers and WALs
Driver
Launch jobs on data
Executor
Receiver
Continuously receive data using High Level API
kafka
Update offsets in Zookeeper
Save data to WAL

KEYWORDS

EV: Electric Vehicle

1.1 KAFKA PRODUCER AND CONSUMER

Using Kafka, we have acquired daily stock reports detailing the price at market open, the high for the day, the low for the day, and the price at market close, as well as the time stamp of the day of the data. The data we gathered is the most recent 100 days, but the data can be altered to consider all data since 1999. The data is preprocessed to declare an index, as well as convert the date, which was the original index, to become a value of the data instead. Some minor changes are then made to the data for aesthetic purposes, and then the original columns are then removed from the data. This data is then loaded into the target cluster, to a topic declared 'mm.' Another topic, 'mm1' is also declared within the Kafka cluster, which is then used to load the data from the cluster into the target source,theJupyterNotebook.

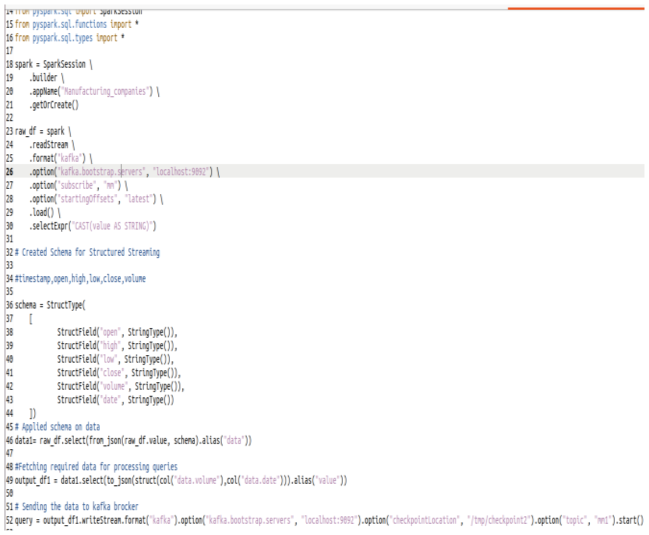


```
from kafka import KafkaProducer
import requests
from json import dumps
import time

def on_message1(message):
    producer1.send('mm', message)
    producer1.flush()

producer1 = KafkaProducer(value_serializer=lambda m: dumps(m).encode("utf-8"), bootstrap_servers=['localhost:9092'])


# url to collect data of Mahindra and Mahindra shares
url = 'Shttps://www.alphavantage.co/query?function=TIME_SERIES_DAILY&symbol=M&M.BSE&apikey=36S0YNI4HJXT8L12&outputsize=full'
r = requests.get(url)
data=r.json()
#print(data)
#preprocessing
del(data['Meta Data'])
df=data['Time Series (Daily)']
#print(df)
jsonFile={}
j=0
for i in df.keys():
    jsonFile[j]=df[i]
    jsonFile[j]['0. date']=i
    #print(i)
    j+=1
print(jsonFile)
for i in jsonFile.keys():
        jsonFile[i]["open"]=jsonFile[i]["1. open"]
        jsonFile[i]["high"]=jsonFile[i]["2. high"]
        jsonFile[i]["low"]=jsonFile[i]["3. low"]
        jsonFile[i]["close"]=jsonFile[i]["4. close"]
        jsonFile[i]["volume"]=jsonFile[i]["5. volume"]
        jsonFile[i]["date"]=jsonFile[i]["0. date"]
        del jsonFile[i]["1. open"]
        del jsonFile[i]["2. high"]
        del jsonFile[i]["3. low"]
        del jsonFile[i]["4. close"]
        del jsonFile[i]["5. volume"]
```

Figure 1.1: Source code from the Kafka Producer, to ingest the data and add it to the Kafka Cluster.

1.2 SPARK PROCESSING

Spark is used to prepare the data for analyzing. Using readstream, Spark subscribes to the mm topic, then converts the entire data field to a string, and allowing the data to be structured. This is done by creating a data model type, and then casting the type onto the data. Once the data is prepared, it can then be sent back to the Kafka broker using writestream, and prepared for the Notebook.



```
from pyspark.sql import SparkSession
from pyspark.sql.functions import *
from pyspark.sql.types import *

spark = SparkSession \
    .builder \
    .appName("Manufacturing_companies") \
    .getOrCreate()

raw_df = spark \
    .readStream \
    .format("kafka") \
    .option("kafka.bootstrap.servers", "localhost:9092") \
    .option("subscribe", "mm") \
    .option("startingOffsets", "latest") \
    .load() \
    .selectExpr("CAST(value AS STRING)")

# Created Schema for Structured Streaming

#timestamp,open,high,low,close,volume

schema = StructType(
    [
            StructField("open", StringType()),
            StructField("high", StringType()),
            StructField("low", StringType()),
            StructField("close", StringType()),
            StructField("volume", StringType()),
            StructField("date", StringType())
    ])
# Applied schema on data
data1= raw_df.select(from_json(raw_df.value, schema).alias("data"))

#Fetching required data for processing queries
output_df1 = data1.select(to_json(struct(col("data.volume"),col("data.date"))).alias("value"))

# Sending the data to kafka brocker
query = output_df1.writeStream.format("kafka").option("kafka.bootstrap.servers", "localhost:9092").option("checkpointLocation", "/tmp/checkpoint2").option("topic", "mm1").start()
```

Figure 1.2: Code of the Spark Processing

1.3 JUPYTER NOTEBOOK

Once the data is sent to the Notebook, it can be loaded into a dataframe, and then analysis can begin. The data is loaded from the Kafka cluster, and then cleaned to prepare a plot of the volume vs data. For convenience's sake, only the latest 30 days are plotted to have a legible graph. This allows the data to be analyzed for trends in volume change, which is beneficial knowledge due to its effect on stock price. Using matplotlib, and kafka-python allows for the gathering of data from the topic and moving it to the Jupyter Notebook.

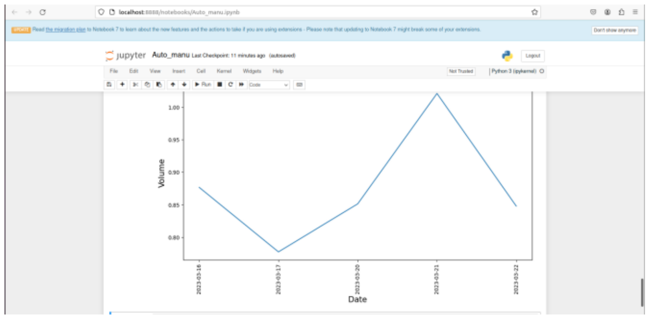
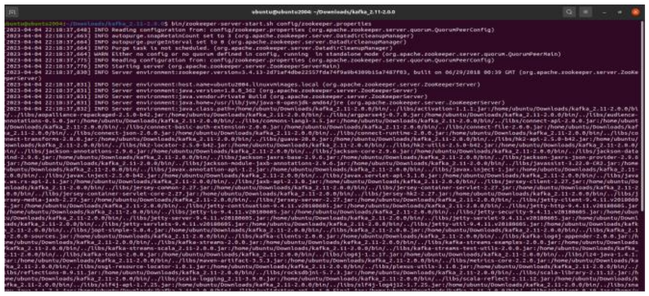


Figure 1.3: The plot that comes from the notebook. This is a small selection of the data, only spanning six days but showing good trend perspective.

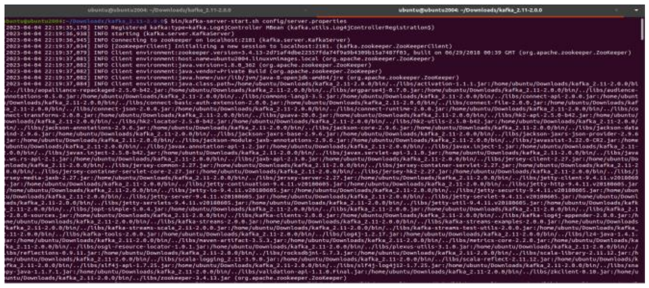
2 INITIATION

Zookeeper is used as a cluster manager and must be initiated before the system starts. Starting it will begin the Kafka session, allowing for clusters to be maintained. After that, Kafka itself must then be initiated. Once Kafka is opened, the topics can then be created, mm and mm1, respectively. These topics are within the brokers, which are within the clusters. Running the mm file will then complete the process to ingest the data to the cluster, and then running mm1 will preprocess and send the data to the target source. The data is now ready for the notebook and can begin data analysis.



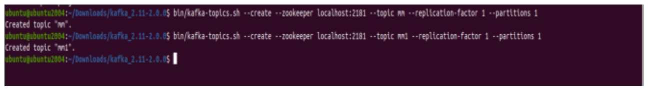
```
ubuntu@ubuntu2004:~/Downloads/kafka_2.11-2.0.0$ bin/zookeeper-server-start.sh config/zookeeper.properties
[2023-04-04 22:18:37,640] INFO Reading configuration from: config/zookeeper.properties (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2023-04-04 22:18:37,663] INFO autopurge.snapRetainCount set to 3 (org.apache.zookeeper.server.DatadirCleanupManager)
[2023-04-04 22:18:37,664] INFO autopurge.purgeInterval set to 0 (org.apache.zookeeper.server.DatadirCleanupManager)
[2023-04-04 22:18:37,664] INFO Purge task is not scheduled. (org.apache.zookeeper.server.DatadirCleanupManager)
[2023-04-04 22:18:37,664] WARN Either no config or no quorum defined in config, running  in standalone mode (org.apache.zookeeper.server.quorum.QuorumPeerMain)
[2023-04-04 22:18:37,775] INFO Reading configuration from: config/zookeeper.properties (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2023-04-04 22:18:37,776] INFO Starting server (org.apache.zookeeper.server.ZooKeeperServerMain)
[2023-04-04 22:18:37,830] INFO Server environment:zookeeper.version=3.4.13-2d71af4dbe22557fda74f9a9b4309b15a7487f03, built on 06/29/2018 00:39 GMT (org.apache.zookeeper.server.ZooKeeperServer)
[2023-04-04 22:18:37,831] INFO Server environment:host.name=ubuntu2004.linuxvmimages.local (org.apache.zookeeper.server.ZooKeeperServer)
[2023-04-04 22:18:37,831] INFO Server environment:java.version=1.8.0_362 (org.apache.zookeeper.server.ZooKeeperServer)
[2023-04-04 22:18:37,831] INFO Server environment:java.vendor=Private Build (org.apache.zookeeper.server.ZooKeeperServer)
[2023-04-04 22:18:37,831] INFO Server environment:java.home=/usr/lib/jvm/java-8-openjdk-amd64/jre (org.apache.zookeeper.server.ZooKeeperServer)
[2023-04-04 22:18:37,832] INFO Server environment:java.class.path=/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/activation-1.1.1.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/aopalliance-repackaged-2.5.0-b42.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/argparse4j-0.7.0.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/audience-annotations-0.5.0.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/commons-lang3-3.5.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/connect-api-2.0.0.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/connect-basic-auth-extension-2.0.0.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/connect-file-2.0.0.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/connect-json-2.0.0.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/connect-runtime-2.0.0.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/connect-transforms-2.0.0.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/guava-20.0.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/hk2-api-2.5.0-b42.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/hk2-locator-2.5.0-b42.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/hk2-utils-2.5.0-b42.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jackson-annotations-2.9.6.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jackson-core-2.9.6.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jackson-databind-2.9.6.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jackson-jaxrs-base-2.9.6.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jackson-jaxrs-json-provider-2.9.6.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jackson-module-jaxb-annotations-2.9.6.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/javassist-3.22.0-CR2.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/javax.annotation-api-1.2.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/javax.inject-1.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/javax.inject-2.5.0-b42.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/javax.servlet-api-3.1.0.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/javax.ws.rs-api-2.1.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jaxb-api-2.3.0.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jersey-client-2.27.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jersey-common-2.27.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jersey-container-servlet-2.27.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jersey-container-servlet-core-2.27.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jersey-hk2-2.27.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jersey-media-jaxb-2.27.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jersey-server-2.27.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jetty-client-9.4.11.v20180605.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jetty-continuation-9.4.11.v20180605.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jetty-http-9.4.11.v20180605.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jetty-io-9.4.11.v20180605.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jetty-security-9.4.11.v20180605.jar:/home/ubun
[illegible]
```

Figure 2.1



```
ubuntu@ubuntu2004:~/Downloads/kafka_2.11-2.0.0$ bin/kafka-server-start.sh config/server.properties
[2023-04-04 22:19:35,170] INFO Registered kafka:type=kafka.Log4jController MBean (kafka.utils.Log4jControllerRegistration$)
[2023-04-04 22:19:36,938] INFO starting (kafka.server.KafkaServer)
[2023-04-04 22:19:36,945] INFO Connecting to zookeeper on localhost:2181 (kafka.server.KafkaServer)
[2023-04-04 22:19:37,034] INFO [ZooKeeperClient] Initializing a new session to localhost:2181. (kafka.zookeeper.ZooKeeperClient)
[2023-04-04 22:19:37,079] INFO Client environment:zookeeper.version=3.4.13-2d71af4dbe22557fda74f9a9b4309b15a7487f03, built on 06/29/2018 00:39 GMT (org.apache.zookeeper.ZooKeeper)
[2023-04-04 22:19:37,081] INFO Client environment:host.name=ubuntu2004.linuxvmimages.local (org.apache.zookeeper.ZooKeeper)
[2023-04-04 22:19:37,082] INFO Client environment:java.version=1.8.0_362 (org.apache.zookeeper.ZooKeeper)
[2023-04-04 22:19:37,082] INFO Client environment:java.vendor=Private Build (org.apache.zookeeper.ZooKeeper)
[2023-04-04 22:19:37,082] INFO Client environment:java.home=/usr/lib/jvm/java-8-openjdk-amd64/jre (org.apache.zookeeper.ZooKeeper)
[2023-04-04 22:19:37,084] INFO Client environment:java.class.path=/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/activation-1.1.1.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/aopalliance-repackaged-2.5.0-b42.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/argparse4j-0.7.0.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/audience-annotations-0.5.0.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/commons-lang3-3.5.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/connect-api-2.0.0.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/connect-basic-auth-extension-2.0.0.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/connect-file-2.0.0.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/connect-json-2.0.0.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/connect-runtime-2.0.0.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/connect-transforms-2.0.0.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/guava-20.0.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/hk2-api-2.5.0-b42.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/hk2-locator-2.5.0-b42.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/hk2-utils-2.5.0-b42.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jackson-annotations-2.9.6.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jackson-core-2.9.6.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jackson-databind-2.9.6.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jackson-jaxrs-base-2.9.6.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jackson-jaxrs-json-provider-2.9.6.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jackson-module-jaxb-annotations-2.9.6.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/javassist-3.22.0-CR2.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/javax.annotation-api-1.2.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/javax.inject-1.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/javax.inject-2.5.0-b42.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/javax.servlet-api-3.1.0.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/javax.ws.rs-api-2.1.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jaxb-api-2.3.0.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jersey-client-2.27.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jersey-common-2.27.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jersey-container-servlet-2.27.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jersey-container-servlet-core-2.27.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jersey-hk2-2.27.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jersey-media-jaxb-2.27.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jersey-server-2.27.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jetty-client-9.4.11.v20180605.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jetty-continuation-9.4.11.v20180605.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jetty-http-9.4.11.v20180605.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jetty-io-9.4.11.v20180605.jar:/home/ubuntu/Downloads/kafka_2.11-2.0.0/bin/../libs/jetty-security-9.4.11.v20180605.jar:/home/ubun
[illegible]
untu/Downloads/kafka_2.11-2.0.0/bin/../libs/zookeeper-3.4.13.jar (org.apache.zookeeper.ZooKeeper)
```

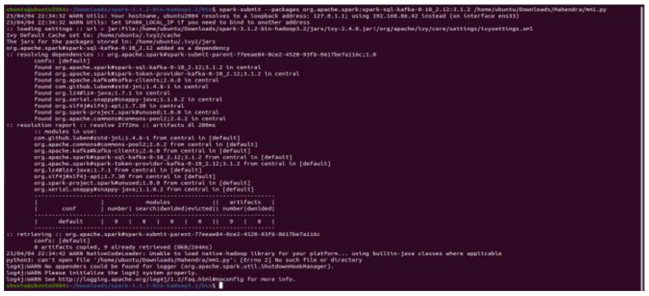
Figure 2.2



```
ubuntu@ubuntu2004:~/Downloads/kafka_2.11-2.0.0$ bin/kafka-topics.sh --create --zookeeper localhost:2181 --topic mm --replication-factor 1 --partitions 1
Created topic "mm".
ubuntu@ubuntu2004:~/Downloads/kafka_2.11-2.0.0$ bin/kafka-topics.sh --create --zookeeper localhost:2181 --topic mm1 --replication-factor 1 --partitions 1
Created topic "mm1".
ubuntu@ubuntu2004:~/Downloads/kafka_2.11-2.0.0$
```

Figure 2.3

Figure 2.1,2.2,2.3: These images show the code and processes of the Zookeeper initiation, the Kafka initiation, and then the topic declaration.



```
ubuntu@ubuntu2004:~/Downloads/spark-3.1.2-bin-hadoop3.2/bin$ spark-submit --packages org.apache.spark:spark-sql-kafka-0-10_2.12:3.1.2 /home/ubuntu/Downloads/Mahendra/mm1.py
23/04/04 22:34:32 WARN Utils: Your hostname, ubuntu2004 resolves to a loopback address: 127.0.1.1; using 192.168.86.42 instead (on interface ens33)
23/04/04 22:34:32 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
:: loading settings :: url = jar:file:/home/ubuntu/Downloads/spark-3.1.2-bin-hadoop3.2/jars/ivy-2.4.0.jar!/org/apache/ivy/core/settings/ivysettings.xml
Ivy Default Cache set to: /home/ubuntu/.ivy2/cache
The jars for the packages stored in: /home/ubuntu/.ivy2/jars
org.apache.spark#spark-sql-kafka-0-10_2.12 added as a dependency
:: resolving dependencies :: org.apache.spark#spark-submit-parent-77eeae84-0ce2-4520-93f6-0617be7a116c;1.0
        confs: [default]
        found org.apache.spark#spark-sql-kafka-0-10_2.12;3.1.2 in central
        found org.apache.spark#spark-token-provider-kafka-0-10_2.12;3.1.2 in central
        found org.apache.kafka#kafka-clients;2.6.0 in central
        found com.github.luben#zstd-jni;1.4.8-1 in central
        found org.lz4#lz4-java;1.7.1 in central
        found org.xerial.snappy#snappy-java;1.1.8.2 in central
        found org.slf4j#slf4j-api;1.7.30 in central
        found org.spark-project.spark#unused;1.0.0 in central
        found org.apache.commons#commons-pool2;2.6.2 in central
:: resolution report :: resolve 2772ms :: artifacts dl 200ms
        :: modules in use:
        com.github.luben#zstd-jni;1.4.8-1 from central in [default]
        org.apache.commons#commons-pool2;2.6.2 from central in [default]
        org.apache.kafka#kafka-clients;2.6.0 from central in [default]
        org.apache.spark#spark-sql-kafka-0-10_2.12;3.1.2 from central in [default]
        org.apache.spark#spark-token-provider-kafka-0-10_2.12;3.1.2 from central in [default]
        org.lz4#lz4-java;1.7.1 from central in [default]
        org.slf4j#slf4j-api;1.7.30 from central in [default]
        org.spark-project.spark#unused;1.0.0 from central in [default]
        org.xerial.snappy#snappy-java;1.1.8.2 from central in [default]
        ---------------------------------------------------------------------
        |                  |            modules            ||   artifacts   |
        |       conf       | number| search|dwnlded|evicted|| number|dwnlded|
        ---------------------------------------------------------------------
        |      default     |   9   |   0   |   0   |   0   ||   9   |   0   |
        ---------------------------------------------------------------------
:: retrieving :: org.apache.spark#spark-submit-parent-77eeae84-0ce2-4520-93f6-0617be7a116c
        confs: [default]
        0 artifacts copied, 9 already retrieved (0kB/264ms)
23/04/04 22:34:42 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
python3: can't open file '/home/ubuntu/Downloads/Mahendra/mm1.py': [Errno 2] No such file or directory
log4j:WARN No appenders could be found for logger (org.apache.spark.util.ShutdownHookManager).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
ubuntu@ubuntu2004:~/Downloads/spark-3.1.2-bin-hadoop3.2/bin$
```

Figure 2.4: Shows the code and processes of the Spark submittal process.

3. PRELIMINARY RESULTS

Initial results have shown promising increases in the volume of shares being traded, which coinciding with the increase in stock prices, indicates the correlation between the two. Continued preparation of the data is expected to follow the trends of the volume plotting. This can become a valuable asset for stockbrokers to know when to buy and sell stocks, and for the average citizen to know when to invest in order to make capital gains.

4.CONTRIBUTIONS

Emmitt Hasty

- Streaming Data Pipeline
- Install and configured Zookeeper, Kafka

Snehalatha Mallavarapu

- Install and configured Jupyter Notebook
- Spark Streaming Jobs

Amulya Sara

- Streaming Data Pipeline
- Install and Configured Zookeeper, Kafka

Sai Karthik Naladala

- Spark Streaming Jobs
- Data Pre-processing
- Kafka Producer

Gautham Sidhardh Dommeti

- Data Pre-processing
- Kafka Producer

5.REFERENCES

[1] https://www.digitalocean.com/community/tutorials/how-to-set-up-jupyter-notebook-with-python-3-on-ubuntu-20-04-and-connect-via-ssh-tunneling
[2] https://sparkbyexamples.com/kafka/apache-kafka-cluster-setup/
[3] https://www.alphavantage.co/documentation/