SCALA PROJECT: Spring 2020

Hedge Fund Application:
Real Time Risk Analysis

Team 11:
Amit Pingale
Mayank Gangrade

Goal

Building reactive application for portfolio management and risk analysis

Leveraging:

- Kafka + Spark streaming
- Spark Analysis engine
- MongoDB for maintaining historic records + batch processing

Why Real-time Big Data Pipeline Important

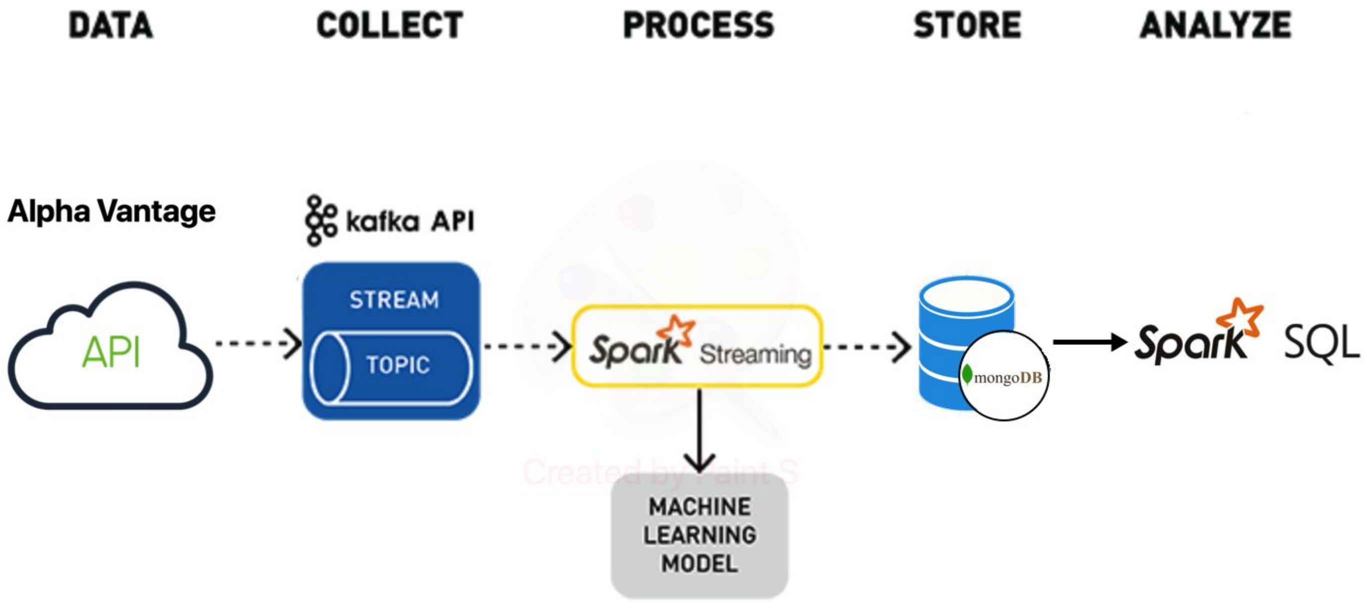
It is estimated that by 2020 approximately 1.7 megabytes of data will be created every second. This results in an increasing demand for real-time and streaming data analysis. For historical data analysis descriptive, prescriptive, and predictive analysis techniques are used. On the other hand, for real-time data analysis, streaming data analysis is the choice. The main benefit of real-time analysis is one can analyze and visualize the report on a real-time basis.

Technology and Tools:

1. Alpha Vantage API for Real Time Data
2. Kafka for capturing Real Time Data
3. Spark Streaming For Consuming Data
4. Spark + Scala for performing Analysis on the data
5. Spark Machine Learning Library
6. MongoDb*
7. Tableau
8. Jupyter Notebook - Scala kernel

* Dumping historic and predicted data on NoSQL database like MongoDB

Architecture



Risk Analysis Process

1. Fetch real-time stock data from Alpha Vantage API
2. Perform cointegration test for pairs trading strategy
3. Develop ensemble of machine learning models to predict the momentum of the asset
4. Analyse stock and predicted value to make a decision
5. Calibrate portfolio to minimise risk

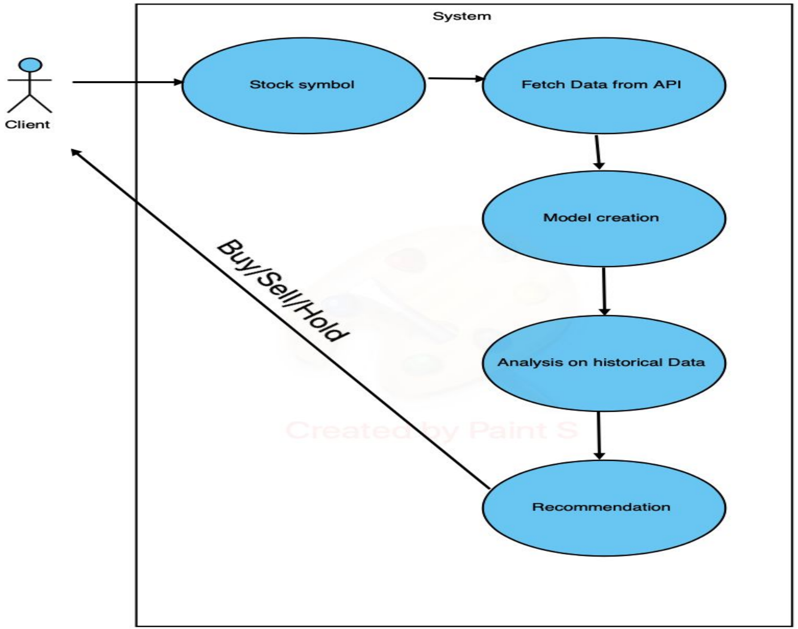
Ensemble Models

1. Linear Regression
2. Decision Tree Regression
3. Random forest Regression
4. Gradient Boosting Regression

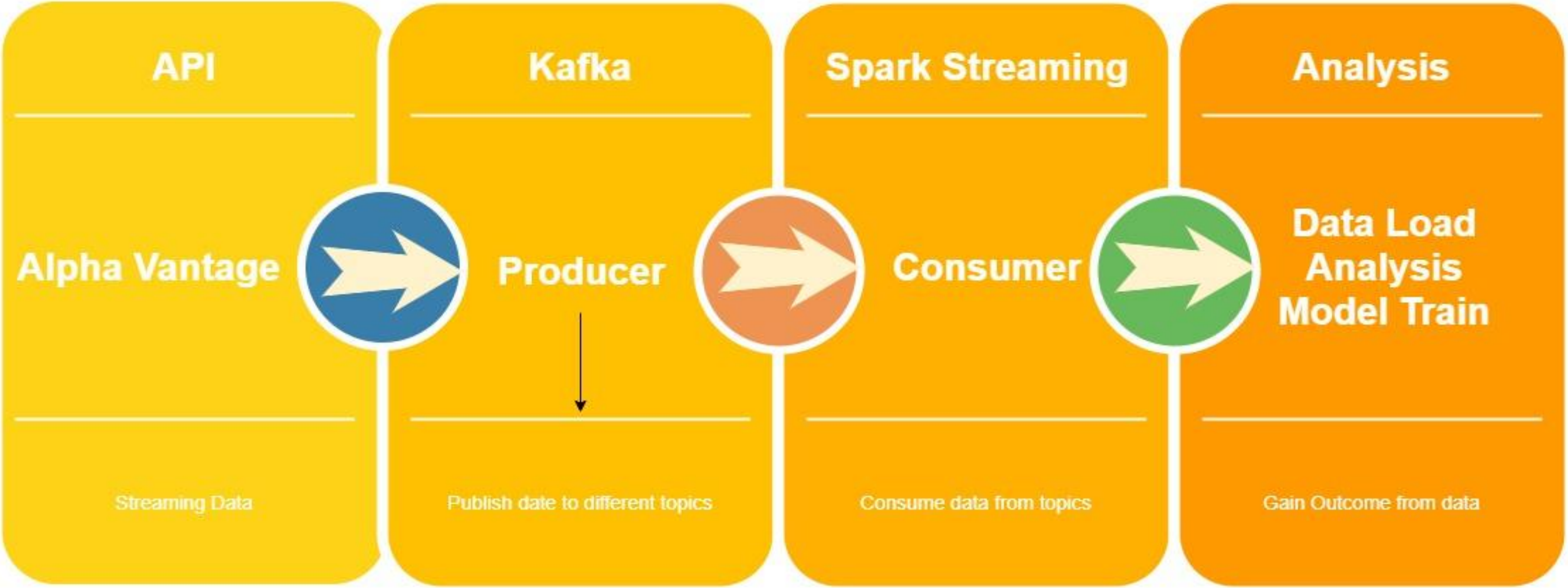
Data features: Stocks

1. Open (Independent feature)
2. High
3. Close
4. Adj Open
5. Volume
6. Low

Use Case Diagram



High Level Architecture



Project Plan

Week 1: (23rd March - 29th March)

- Setting up new git repo
- Setting up whole ecosystem (Spark + Scala + Kafka + API Keys)

Week 2: (30th March - 5th April)

- Improve the code base
- Handle Exceptions and Errors

Project Plan

Week 3: (6th April - 12th April)

- Create test cases and verify all the functions
- Make the code more modular and remove repetition

Week 4: (13 April - 15th April)

- Check the robustness the code by running on longer time
- Visualize the data using a visualization tool

Acceptance criteria

- Data from API should be fetched in every **5 min** and published with in **5 Seconds** on Kafka Topics
- Consume the data from Kafka topics with in **5 seconds** as it arrived, validate each data and load correct data into database
- Selecting best model depending upon RMSE value (**RMSE < 0.7**)
- Update data in real time interactive dashboards with maximum lag of **2 mins**

Thank You