CMPE 272 - Apr 18th 2024

Teammates: SJSU ID

Satwik Upadhyayula:017423796

Vineet Samudrala : 017426253

Bala Supriya Vanaparthi:017464135

Shreekar Kolanu : 017406493

Stock Market Prediction Engine Proposal

Overview

Our group aims to build a robust stock market prediction engine. This document outlines the high-level approach, including the types of data to be used, data storage strategies, and the architecture of the prediction engine.

1. Data Acquisition

Types of Data:

- Historical Stock Prices: Daily open, high, low, close, and adjusted close prices.

- Volume Data: Daily trading volume.

- Fundamental Data: Quarterly and annual financial statements, including income statements, balance sheets, and cash flow statements.

- Sentiment Data: News articles and social media sentiment (e.g., Twitter sentiment analysis).

- Economic Indicators: Interest rates, inflation rates, GDP growth, etc.

- Technical Indicators: Calculated values such as moving averages, Relative Strength Index (RSI), and Moving Average Convergence Divergence (MACD).

Sources of Data:

- Financial APIs: Alpha Vantage, Yahoo Finance, IEX Cloud for historical and real-time data.
- Web Scraping: Tools like BeautifulSoup and Scrapy for extracting sentiment data from news and social media.
- Financial Databases: Bloomberg Terminal, Reuters for comprehensive fundamental data.

2. Data Storage

Stock Selection:

- Number of Stocks: Focus on the S&P 500 index, covering 500 large-cap U.S. stocks.
- Selection Criteria: Market cap, industry, and trading volume.

Data to Store for Each Stock:

- Price Data: Daily open, high, low, close, and adjusted close prices.
- Volume Data: Daily trading volume.
- Fundamental Data: Quarterly and annual financial statements.
- Sentiment Data: Daily aggregated sentiment scores.
- Technical Indicators: Daily calculated values (moving averages, RSI, MACD).

Storage Duration:

- Historical Data: At least 10-15 years of historical data to train long-term models.
- Real-Time Data: Continuously updated with new data as it becomes available.

Storage Solutions:

- Databases:
  - SQL Databases: PostgreSQL, MySQL for structured data (e.g., historical prices, volume).
  - NoSQL Databases: MongoDB, Elasticsearch for unstructured data (e.g., sentiment data).

- Data Warehousing:

  - Cloud-Based Solutions: AWS Redshift, Google BigQuery for scalable storage and efficient querying.

3. Data Processing

ETL (Extract, Transform, Load):

- Tools: Apache Airflow, AWS Glue to automate data extraction, transformation, and loading processes to ensure data consistency and cleanliness.

Feature Engineering:

- Technical Indicators: Calculate moving averages, RSI, MACD, Bollinger Bands.

- Sentiment Analysis: Use Natural Language Processing (NLP) techniques and libraries like NLTK, SpaCy to analyze and score sentiment from news and social media data.

- Economic Indicators: Integrate macroeconomic data.

4. Prediction Engine

Model Selection:

- Algorithms:

  - Traditional Machine Learning: Linear Regression, Decision Trees, Random Forests.

  - Deep Learning: Long Short-Term Memory (LSTM) networks for time series forecasting.

  - Ensemble Models: Combining predictions from multiple models (e.g., using stacking, bagging) to improve accuracy.

Training the Model:

- Data Split: Split data into training (70%), validation (15%), and test sets (15%).

- Hyperparameter Tuning: Use grid search and cross-validation to optimize model parameters.

- Model Evaluation: Evaluate models using metrics like RMSE (Root Mean Square Error), MAE (Mean Absolute Error), and R-squared. Use confusion matrices for classification models.

5. Deployment

Model Deployment:

- API Development: Develop an API using Flask or FastAPI to serve predictions.

- Real-Time Predictions: Implement real-time data pipelines using Kafka, Spark Streaming to feed data into the model for live predictions.

- Model Serving: Use Docker and Kubernetes for containerization and orchestration of model serving.

- Monitoring: Implement monitoring tools like Prometheus and Grafana to track model performance and detect drift.

6. Data Access

Read/Write Access:

- Read-Only: Historical data, fundamental data, and precomputed technical indicators will be read-only.

- Read/Write: Real-time data ingestion and model retraining processes will require read/write access.

7. Security and Compliance

Data Security:

- Encryption: Use AES-256 encryption for data at rest and TLS for data in transit.

- Access Control: Implement role-based access control (RBAC) and audit logging.

Compliance:

- Regulations: Adhere to financial regulations and data privacy laws, including GDPR and CCPA.

High-Level Architecture

1. Data Ingestion Layer:
   - Sources: Financial APIs, Web Scraping, News Feeds.
   - Tools: Kafka for real-time data ingestion, AWS Lambda for serverless data processing.

2. Data Storage Layer:
   - Databases: SQL/NoSQL Databases (PostgreSQL, MongoDB).
   - Data Warehouse: Cloud-based solutions (AWS Redshift, Google BigQuery).

3. Data Processing Layer:
   - ETL Pipelines: Apache Airflow, AWS Glue.
   - Feature Engineering: Calculation of technical indicators, sentiment analysis, integration of economic indicators.

4. Modeling Layer:
   - Model Training: Use of Jupyter Notebooks, TensorFlow, PyTorch.
   - Ensemble Models: Combine multiple models for improved accuracy.

5. Prediction Layer:
   - API Development: Flask/FastAPI for serving predictions.
   - Real-Time Prediction Pipeline: Kafka, Spark Streaming for live data.

6. Monitoring and Maintenance:
   - Performance Monitoring: Prometheus, Grafana.

- Retraining and Updating Models: Automated retraining pipelines using Jenkins, GitLab CI/CD.

By following this structured and detailed approach, our stock market prediction engine will leverage a wide variety of data sources and advanced machine learning techniques to provide accurate and reliable predictions.