

# بسم الله الرحمن الرحيم



## سیستم تحلیل مالی و معاملات بلادرنگ با هوش مصنوعی

پروژه نهایی درس تحلیل‌ها و سیستم‌های داده حجیم



استاد درس: دکتر حسن نادری

دانشجویان: علی احمدی، امیر کریمی، مهدی شکاری و اهورا امینی

بهمن ۱۴۰۳

## فهرست مطالب

2.....	مقدمه
2.....	هدف پروژه
2.....	اهمیت پروژه
2.....	چالش‌ها
3.....	معماری سیستم
4.....	API Nobitex
4.....	Data Ingestion Service
4.....	Apache Kafka
4.....	Stream Data Processing Service (Apache Spark)
5.....	محاسبه شاخص‌ها
5.....	تولید سیگنال‌ها
5.....	QuestDB
5.....	Aggregation Service

## مقدمه

### هدف پروژه

پروژه تحلیل داده‌های مالی در بازارهای مالی مانند رمزارز و نمادهای سهام مختلف به منظور تجزیه و تحلیل داده‌ها در زمان واقعی (Real-time) طراحی و پیاده‌سازی شده است. این سیستم با استفاده از تکنولوژی‌های توزیع شده و پردازش بلادرنگ داده‌ها، امکان تحلیل سریع و تصمیم‌گیری به موقع برای معامله‌گران بازارهای مالی را فراهم می‌کند.

بازارهای مالی به دلیل نوسانات شدید و لحظه‌ای قیمت‌ها، نیاز به ابزارهایی دارند که بتوانند به سرعت و به صورت دقیق، اطلاعات را پردازش کرده و سیگنال‌های مناسب برای خرید و فروش ارائه دهند. در این پروژه، هدف اصلی این است که با استفاده از پردازش‌های پیچیده و بلادرنگ، سیگنال‌های دقیق و به موقع برای معامله‌گران فراهم شود تا بتوانند تصمیمات بهتری اتخاذ نمایند.

### اهمیت پروژه

تحلیل داده‌های مالی از اهمیت بسیار بالایی برخوردار است. با افزایش حجم داده‌ها و نیاز به پردازش بلادرنگ، استفاده از سیستم‌های مقیاس‌پذیر و توزیع شده مانند Apache Spark، Apache Kafka و QuestDB به ویژه در تحلیل‌های مالی اهمیت پیدا می‌کند. این پروژه نه تنها بر مقیاس‌پذیری تمرکز دارد، بلکه توانایی پردازش داده‌ها در لحظه را نیز فراهم می‌آورد.

به طور ساده، این پروژه یک سیستم بلادرنگ است که به صورت خودکار، داده‌ها را از بازارهای مالی دریافت کرده، پردازش کرده و سپس سیگنال‌های مناسب خرید و فروش و همچنین با استفاده از هوش مصنوعی پیشبینی روند را نیز به کاربران نمایش می‌دهد.

این توانایی می‌تواند برای کسانی که در بازارهای مالی فعالیت می‌کنند، به ویژه برای کسانی که به دنبال تصمیم‌گیری‌های سریع و بهینه هستند، بسیار ارزشمند باشد.

### چالش‌ها

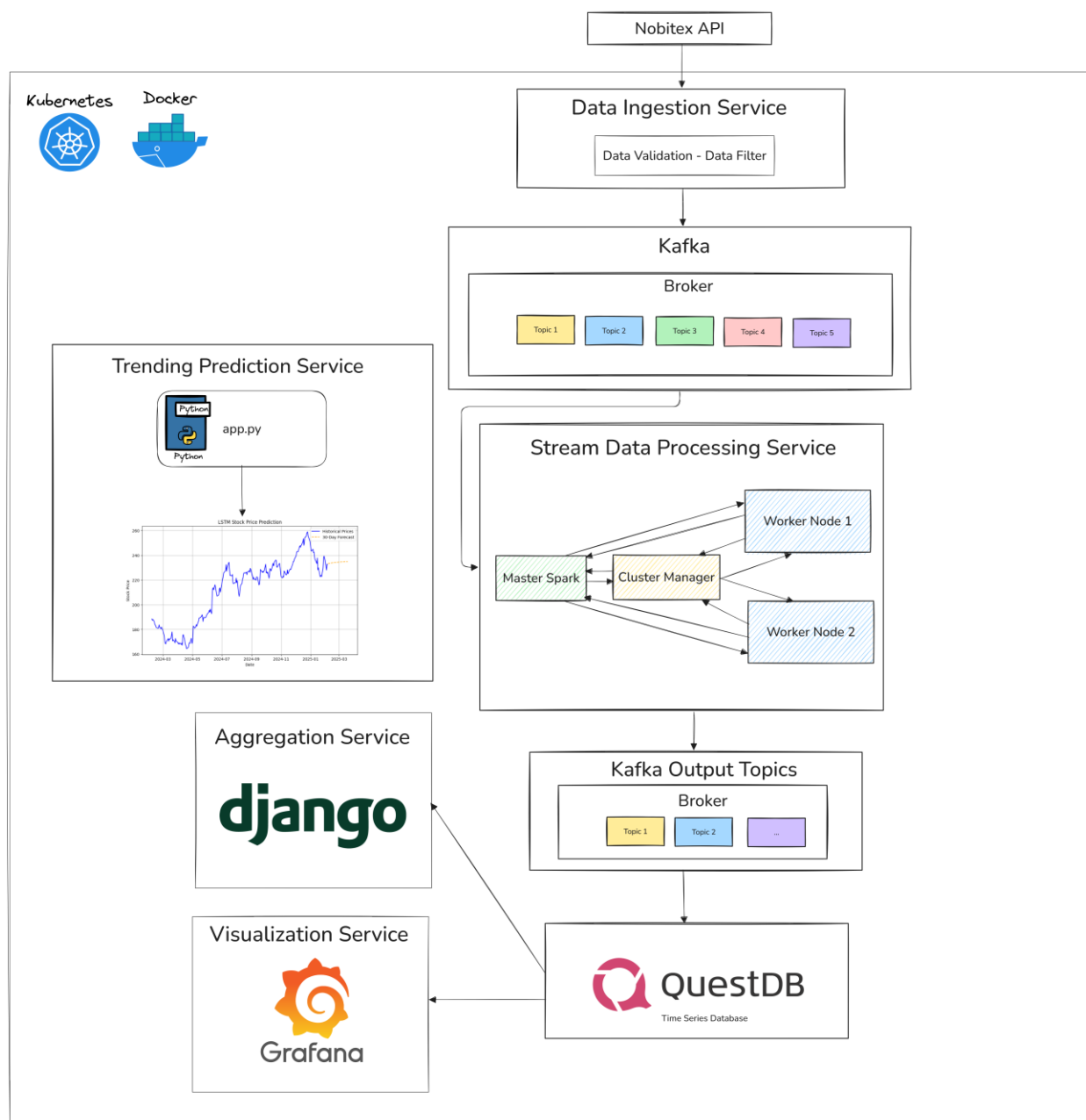
در این پروژه، برخی از چالش‌ها به شرح زیر هستند:

- حجم بالای داده‌ها: با توجه به حجم عظیم داده‌های موجود در بازارهای مالی، پردازش سریع داده‌ها به صورت بلادرنگ نیازمند استفاده از سیستم‌های توزیع شده است.
- تأخیر در پردازش (Latency): سیستم باید قادر باشد تا داده‌ها را با کمترین تأخیر پردازش کرده و سیگنال‌های دقیق و به موقع برای کاربران ارسال نماید.
- مقیاس‌پذیری: با توجه به اینکه داده‌ها به صورت لحظه‌ای وارد سیستم می‌شوند، مقیاس‌پذیری سیستم برای مدیریت تعداد زیاد درخواست‌ها و داده‌ها ضروری است.
- پیشبینی روند: استفاده از هوش مصنوعی به دلیل پیچیدگی ذاتی داده‌های مالی و همچنین پیشبینی آن‌ها، استفاده از الگوریتم مناسب برای تحلیل روند در آینده بسیار چالش برانگیز و محل بحث است.

هدف از این پروژه این است که با استفاده از معماری‌های مقیاس‌پذیر، پردازش‌های توزیع شده و استفاده از هوش مصنوعی (یادگیری عمیق) این چالش‌ها را پشت سر گذاشته و سیستمی کارآمد و سریع برای تحلیل داده‌های بازارهای مالی ارائه دهد.

## معماری سیستم

معماری سیستم به شرح زیر است:



این معماری یک سیستم توزیع شده برای پردازش داده‌ها به صورت بلادرنگ است که شامل اجزای مختلفی می‌باشد. در ادامه توضیحات معماری ارائه شده به تفصیل شرح داده خواهد شد.

## API Nobitex

این API به عنوان منبع داده‌های رمز ارز عمل می‌کند. داده‌ها شامل اطلاعات معاملات، قیمت‌ها، حجم بازار و سایر معیارهای مالی مرتبط با رمزارزها می‌باشد. ساختار داده‌های دریافتی به شرح زیر است:

نام متغیر	نوع متغیر	توضیحات
stock_symbol	object	نماد سهام
local_time	timestamps	تاریخ و ساعت دقیق دریافت داده
open_prices	float	قیمت باز شدن (بر اساس کندل)
high_prices	float	بالاترین قیمت
low_prices	float	پایین‌ترین قیمت
close_prices	float	قیمت بسته شدن
volumes	int	حجم معامله بازار

## Data Ingestion Service

وظیفه این سرویس جمع‌آوری و انتقال داده از API Nobitex است. در این فرآیند، داده‌ها مورد اعتبارسنجی (Validation) و فیلترگذاری قرار می‌گیرند تا فقط داده‌های معتبر به مرحله بعدی ارسال شوند. در صورت صحت داده‌ها، توسط کد producer هر داده جدید بر اساس فیلد stock\_symbol به یک Topic در Kafka ارسال می‌شود.

برای مثال داده‌های مربوط به BTCIRT به تاپیک btcirt\_topic ارسال خواهد شد.

```
2025-01-25 18:35:44 INFO:root:Data sent to topic=shibirt_topic, partition=0, offset=7, data={'stock_symbol': 'SHIBIRT', 'local_time': '2025-01-25 18:34:00', 'open': 1680.2, 'high': 1684.6, 'low': 1680.1, 'close': 1680.1, 'volume': 811.0, 'topic': 'shibirt_topic'}
2025-01-25 18:35:44 INFO:root:payload: {'stock_symbol': 'SHIBIRT', 'local_time': '2025-01-25 18:34:00', 'open': 1680.2, 'high': 1684.6, 'low': 1680.1, 'close': 1680.1, 'volume': 811.0, 'topic': 'shibirt_topic'} processed and forwarded to Kafka topic: shibirt_topic
2025-01-25 18:36:45 INFO:root:Data sent to topic=btctrt_topic, partition=0, offset=7, data={'stock_symbol': 'BTCIRT', 'local_time': '2025-01-25 18:36:00', 'open': 8697888889.0, 'high': 8697888889.0, 'low': 8697888889.0, 'close': 8697888889.0, 'volume': 5.22939e-05, 'topic': 'btctrt_topic'}
2025-01-25 18:36:45 INFO:root:payload: {'stock_symbol': 'BTCIRT', 'local_time': '2025-01-25 18:36:00', 'open': 8697888889.0, 'high': 8697888889.0, 'low': 8697888889.0, 'close': 8697888889.0, 'volume': 5.22939e-05, 'topic': 'btctrt_topic'} processed and forwarded to Kafka topic: btctrt_topic
2025-01-25 18:36:53 INFO:root:Data sent to topic=usdtirt_topic, partition=0, offset=8, data={'stock_symbol': 'USDIRT', 'local_time': '2025-01-25 18:36:00', 'open': 83597.0, 'high': 83598.0, 'low': 83552.0, 'close': 83553.0, 'volume': 2718.0373713987, 'topic': 'usdtirt_topic'}
2025-01-25 18:36:53 INFO:root:payload: {'stock_symbol': 'USDIRT', 'local_time': '2025-01-25 18:36:00', 'open': 83597.0, 'high': 83598.0, 'low': 83552.0, 'close': 83553.0, 'volume': 2718.0373713987, 'topic': 'usdtirt_topic'} processed and forwarded to Kafka topic: usdtirt_topic
2025-01-25 18:36:54 INFO:root:Data sent to topic=ethirt_topic, partition=0, offset=8, data={'stock_symbol': 'ETHIRT', 'local_time': '2025-01-25 18:36:00', 'open': 277734600.0, 'high': 277734600.0, 'low': 277170000.0, 'close': 277734600.0, 'volume': 0.2178, 'topic': 'ethirt_topic'}
2025-01-25 18:36:54 INFO:root:payload: {'stock_symbol': 'ETHIRT', 'local_time': '2025-01-25 18:36:00', 'open': 277734600.0, 'high': 277734600.0, 'low': 277170000.0, 'close': 277734600.0, 'volume': 0.2178, 'topic': 'ethirt_topic'} processed and forwarded to Kafka topic: ethirt_topic
2025-01-25 18:36:57 INFO:root:Data sent to topic=etctrt_topic, partition=0, offset=8, data={'stock_symbol': 'ETCIRT', 'local_time': '2025-01-25 18:36:00', 'open': 2261992.0, 'high': 2261992.0, 'low': 2261992.0, 'close': 2261992.0, 'volume': 0.0, 'topic': 'etctrt_topic'}
2025-01-25 18:36:57 INFO:root:payload: {'stock_symbol': 'ETCIRT', 'local_time': '2025-01-25 18:36:00', 'open': 2261992.0, 'high': 2261992.0, 'low': 2261992.0, 'close': 2261992.0, 'volume': 0.0, 'topic': 'etctrt_topic'} processed and forwarded to Kafka topic: etctrt_topic
```

## Apache Kafka

داده‌های تصحیح شده توسط سرویس Data Ingestion به Kafka به عنوان یک سیستم مقیاس‌پذیر و پایدار وارد می‌شوند. امکان انتقال داده‌ها به اجزای مختلف را به شیوه‌ای بسیار کارآمد فراهم می‌کند.

Broker: مسئول توزیع داده‌ها در قالب Topic‌های مختلف است. در معماری سیستم چندین Topic نمایش داده شده است که هر کدام ممکن است یک جزئی از داده را نمایش دهند.

## Stream Data Processing Service (Apache Spark)

داده‌های ارسال شده از Kafka در اینجا پردازش می‌شوند. Apache Spark به عنوان یک محیط پردازش موازی برای انجام عملیات پیچیده روی داده‌ها عمل می‌کند.

Cluster Spark و Master Spark: مدیریت عملیات پردازش توزیع شده را برعهده دارند که داده‌ها میان Worker Node1 و Worker Node2 تقسیم می‌شوند.

Kafka Output Topics: داده‌های پردازش شده مجدداً به Kafka ارسال می‌شوند، جایی که موضوعات خروجی جدید برای این داده‌ها ایجاد می‌شود.

این سرویس شامل چندین بخش است که به طور کلی برای پردازش و داده‌های استریم از Kafka، محاسبه شاخص‌های معاملاتی یا همان اندیکاتورها (SMA, EMA, RSI) و ارسال داده به Kafka و مدیریت میکروبیج‌ها با Spark طراحی شده است.

### محاسبه شاخص‌ها

این بخش برای هر گروه از داده‌ها شاخص‌های زیر را محاسبه می‌کند:

- SMA (5): میانگین ساده در دوره‌های ۵ دقیقه‌ای
- EMA (10): میانگین نمایی دوره‌های ۱۰ دقیقه‌ای
- RSI (10): شاخص قدرت نسبی در دوره‌های ۱۰ دقیقه‌ای
- Average gain & Average Loss: در دوره‌های ۱۰ دقیقه‌ای

### تولید سیگنال‌ها

در این بخش سیگنال‌های خرید، فروش یا نگه داشتن بر اساس شرایط زیر تولید می‌شود:

- سیگنال خرید: اگر  $EMA(10) > SMA(5)$  باشد و  $RSI(10)$  نیز کمتر از ۷۰ باشد.
- سیگنال فروش: اگر  $EMA(10) < SMA(5)$  باشد و  $RSI(10)$  بیشتر از ۳۰ باشد.
- در غیر این صورت سیگنال «نگه داشتن» یا همان hold است.

### QuestDB

داده‌های پردازش شده در QuestDB ذخیره می‌شوند. QuestDB یک پایگاه داده سری زمانی است که برای انجام تحلیل‌های پیچیده و بلادرنگ استفاده می‌شوند. این پایگاه داده برای داده‌های بازارهای مالی نیز بسیار مناسب است و stock\_symbol را به صورت خاص index گذاری می‌کند که به صورت سریع بتواند داده‌های مربوط به هر نماد سهام را در کوئری‌ها برگرداند.

### Aggregation Service

داده‌ها از QuestDB گرفته شده و در یک واسط کاربری با استفاده از Grafana به نمایش درمی‌آیند. این واسط کاربری به تحلیل‌گران و کاربران اجازه می‌دهد تا داده‌ها را به صورت نمودارهای مناسب بررسی کرده و سیگنال مناسب را نیز دریافت نمایند.