

**باسمه تعالی**

**پروژه پایانی درس سیستم‌های توزیع شده**

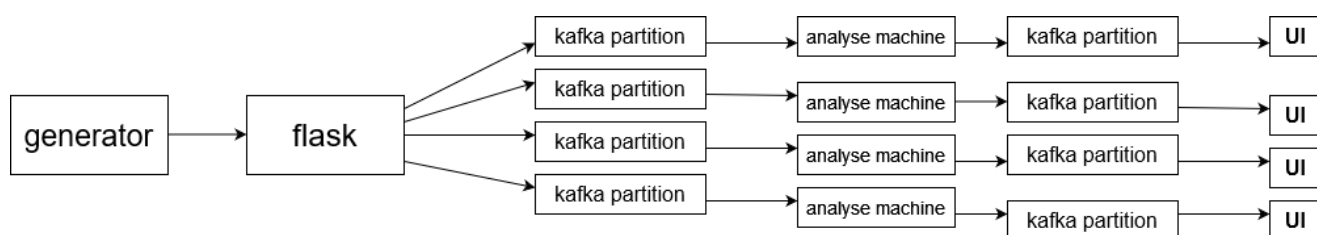
**استاد درس: دکتر شریفی**

**دانشجو: امیرحسین درخشان**

**شماره دانشجویی: ۴۰۲۷۲۳۲۰۹**

## معماری به کار رفته

جهت انجام این پروژه از معماری شکل ۱ استفاده شده است. ابتدا generator داده‌ها را به آدرس و پورت مورد نظر که در آن یک سرور flask آماده است ارسال می‌کند. سپس flask داده‌هایی که شامل قیمت یک سهام می‌باشند را تشخیص داده و به پارتیشن‌های مختلف از یک تاپیک کافکا می‌فرستد. در سمت مقابل analyse machine ها منتظر دریافت داده از تاپیک کافکا هستند و با دریافت داده شروع به پردازش آن و محاسبه ۳ اندیکاتور می‌کنند و نتایج پردازش را به یک تاپیک دیگر کافکا ارسال می‌کنند و در نهایت بخش UI که منتظر دریافت داده از تاپیک دوم است با رسیدن داده‌ها شروع به نمایش آن می‌کند. در ادامه هر بخش به طور مجزا توضیح داده می‌شود.



رسم توضیحی ۱ معماری به کار رفته

## بخش Gnerator

این بخش تنها جهت تولید داده‌ها می‌باشد و بدون هیچ تغییری با دستور

Python generator.py

اجرا شده است.

## بخش flask

این بخش هر داده را از generator دریافت می کند و با استفاده از اینکه این داده مربوط به کدام سهام است آن را به partition مورد نظر از تاپیک stock\_price ارسال می کند.

## بخش stock\_price kafka topic

این تاپیک به منظور ذخیره سازی موقتی قیمت سهام ها پیاده سازی شده است. این تاپیک به تعداد سهام ها به پارتیشن های جداگانه تقسیم شده است و اطلاعات هر سهام در یک پارتیشن مجزا قرار می گیرد.

## بخش analyse machine

این بخش با استفاده از consume کردن اطلاعات موجود در تاپیک stock\_price و استخراج قیمت سهام ها سه معیار میانگین، میانگین توانی (EMA) و RSI را محاسبه می کند. لازم به ذکر است جهت محاسبه میانگین از دو بازه ۲۰ روزه و ۵۰ روزه استفاده شده است. یعنی میانگین در ۲۰ قیمت اخیر و ۵۰ قیمت اخیر محاسبه شده اند. و این مورد برای میانگین توانی نیز صادق است. و در ادامه RSI نیز محاسبه شده است. در نهایت تمام این معیارها به همراه قیمت close و سیگنال تولید شده در یک دیکشنری قرار داده میشوند و به صورت سریالایز شده به پارتیشن مربوط از analysed\_topic ارسال میشوند.

لازم به ذکر است با توجه به این که یکی از اشتباهات در الگوریتم های trading مشکل look-ahead bias است، از قیمت high و low استفاده نشده است و تنها برای محاسبه معیارها به مانند بیشتر الگوریتم های trading از قیمت close استفاده شده است.

## بخش analysed\_topic

در این بخش نیز به تعداد سهام‌ها تایپیک مورد نظر به پارتیشن بندی شده است. و معیارها و قیمت close به طور موقتی ذخیره می‌شوند تا رابط کاربری از آن‌ها جهت بصری‌سازی استفاده کند.

## بخش UI

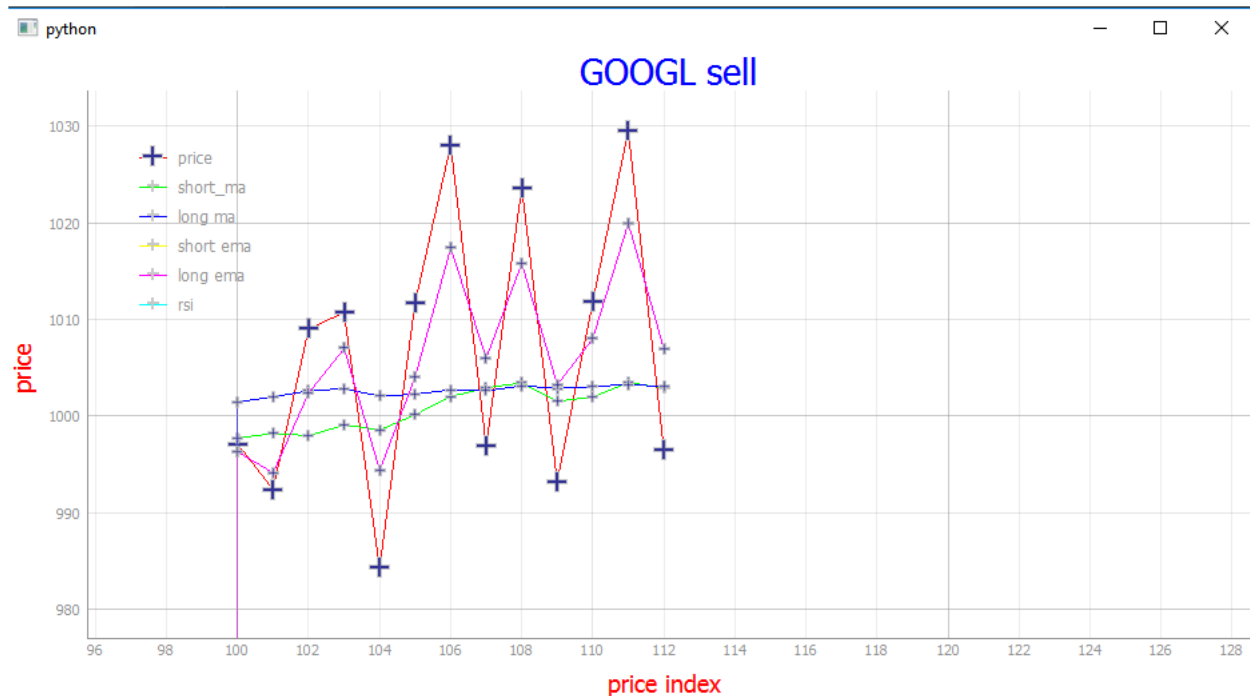
این بخش با استفاده از `pyqt framework` پیاده‌سازی شده است. این بخش با `consume` کردن داده‌های موجود در پارتیشن موردنظر از `analysed_topic` کار نمایش قیمت هر سهام و معیارهای به دست‌آمده را انجام می‌دهد. همچنین در زمانی که سیگنال خرید یا فروش تولید شده باشد با نمایش `notification` کاربر را آگاه می‌کند.

## استراتژی به کار رفته

جهت تولید سیگنال خرید یا فروش از استراتژی `moving average crossover` استفاده شده است. به این صورت که اگر میانگین کوتاه مدت میانگین بلند مدت را روبه بالا قطع کند سیگنال خرید و اگر میانگین کوتاه مدت میانگین بلند مدت را روبه پایین قطع کند سیگنال فروش تولید می‌شود. اما می‌دانیم که استفاده از یک استراتژی به تنهایی کافی نیست بنابراین از دو معیار دیگر نیز استفاده شده است. یعنی به شرط خرید این نکته هم اضافه شده است که میانگین توانی کوتاه مدت بالای میانگین بلند مدت باشد و `RSI` کوچکتر از ۳۰ باشد و به شرط فروش پایین بودن میانگین توانی کوتاه مدت از بلندمدت و `RSI` بزرگتر از ۷۰ اضافه شده‌اند.

## تصویری از محیط برنامه

در ادامه تصویری از رابط کاربری نمایش داده میشود که به کاربر پیام فروش نمایش داده شده است.



رسم توضیحی ۲ نمونه اجرای برنامه

آدرس گیتهاب پروژه

[https://github.com/ahderakhshan/DS\\_project.git](https://github.com/ahderakhshan/DS_project.git)