

گزارش کار پروژه پیش‌بینی روند نزولی یا صعودی قیمت سهام

دانشجویان: علی صادقی (۹۴۰۱۲۲۶۹۰۳۰)، صدرا صمدی (۹۳۱۲۲۶۸۱۲۲)، محمد زیارتی (۹۴۰۱۲۲۶۹۰۳۲)

استاد: مهندس تورانی

پیش‌بینی روند قیمت‌های بورس همیشه مورد توجه فعالین بازار سرمایه و محققان بوده است.

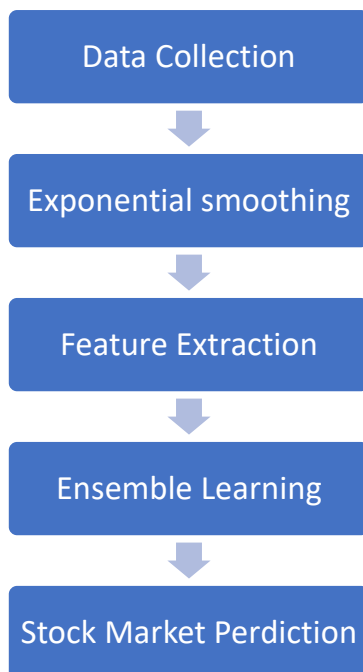
پیش‌بینی قیمت سهام با توجه به تاثیراتی که وضعیت پیچیده جهان بر روی اقتصاد دارد، کاری بس دشوار است که پارامترهای مختلفی روی آن تاثیر دارد.

مسئله پیش‌بینی صعود و یا نزول قیمت یک سهم در n روز آینده را می‌توان به صورت یک مسئله Classification در نظر گرفت.

یکی از تحلیل‌های رایج بازار، تحلیل تکنیکال یا فنی است. تحلیل تکنیکال با توجه به سابقه قیمت، الگوها، اندیکاتورها و... به پیش‌بینی وضعیت و قیمت سهم در آینده می‌پردازد.

در ادامه از اندیکاتورهای RSI، Stochastic، Williams، MACD، PROC و OBV استفاده میکنیم.

متدولوژی و آنالیز



الگوریتم یادگیری‌ای که در این برنامه استفاده می‌کنیم Logistic Regression است. مقادیر اندیکاتورها با توجه به قیمت‌های پایانه محاسبه می‌شوند. این اندیکاتورها بعداً برای train کردن استفاده می‌شوند. در ادامه هر بخش به صورت جداگانه توضیح داده می‌شود.

Data Preprocessing

در این بخش، دیتای ورودی (در اینجا قیمت پایانی) به صورت نمایی Smooth می‌شود. Exponential Smoothing به داده‌های جدید وزن بیشتری داده و به مشاهدات قبلی وزن کمتری می‌دهد. این کار باعث از بین بردن نوسان‌های ریز می‌شود و اطلاعات به نوعی همسان سازی شده و فاصله‌ها کمتر می‌شوند. این کار با استفاده از فرول زیر انجام می‌شود.

$$S_0 = Y_0$$

$$\text{for } t > 0, S_t = \alpha * Y_t + (1 - \alpha) * S_{t-1}$$

آلفا فاکتور اسموسینگ است که بین ۰ و ۱ تغییر می‌کند. هرچه مقدار آلفا بزرگتر باشد، میزان اسموسینگ کمتر می‌شود و اگر آلفا برابر ۱ شود، داده‌ها هیچ تغییری نمی‌کنند.

این کار با از بین بردن نویزها و نوسان‌ها ریز به ما در پیش بینی روند میان مدت و بلند مدت بازار کمک می‌کند.



آبی: قیمت اصلی - سبز: اسموسینگ کاستوم قرمز: اسموسینگ کتابخانه پایتون

این اعداد به دست آمده برای محاسبه مقادیر اندیکاتورهای استفاده می‌شود و سپس از مقادیر اندیکاتورهای برای ساخت feature matrix استفاده می‌شود.

قیمت هدف در روز t از فرمول زیر محاسبه می‌شود.

$$\text{Target}(i) = \text{Sign}(\text{close}(i+d) - \text{close}(i))$$

وقتی مقدار target برابر ۱ شود، روند صعودی و اگر برابر -۱ شود، روند نزولی خواهد بود.

Feature Extraction

اندیکاتورهای پارامترهای مهمی هستند که با توجه به قیمت‌ها محاسبه می‌شوند. اندیکاتورهای پیش‌بینی وضعیت بازار استفاده می‌شوند.

RSI

این اندیکاتور از طریق فرمول زیر به دست می‌آید:

$$RSI = 100 - \frac{100}{1 + RS}, RS = \frac{\text{Average gain over past 14 days}}{\text{Average loss over past 14 days}}$$

اندیکاتور RSI یکی از معروفترین اندیکاتورهای بازار سهام است که با توجه به قیمت، ناحیه‌هایی که سهام بیش از اندازه خریداری شده و قیمت رشد کرده است را با **overbought** و ناحیه‌ای که قیمت افت کرده و با مقدار کمتری معامله می‌شود با **oversold** نشانه گذاری می‌کند. این اندیکاتور بین ۰ و ۱۰۰ تغییر می‌کند و بالای ۷۰ **overbought** و زیر ۳۰ **oversold** است.

Stochastic Oscillator

این اندیکاتور از طریق فرمول زیر به دست می‌آید:

$$\%K = 100 * \frac{(C - L14)}{(H14 - L14)}$$

C برابر قیمت پایانی فعلی است

L14 برابر آخرین قیمت در ۱۴ روز گذشته است

H14 برابر بالاترین قیمت در ۱۴ روز گذشته است

استوکستیک، سرعت قیمت را نمایش می‌دهد.

Williams %R

$$\%R = \frac{(H14 - C)}{(H14 - L14)} * -100$$

ویلیامز بین -۱۰۰ و ۰ نوسان می‌کند. وقتی مقدار از ۲۰- بیشتر شود نشان دهنده سیگنال فروش و وقتی زیر ۸۰- است نشان دهنده سیگنال خرید است.

MACD

این اندیکاتور از تفاضل دو EMA به دست می‌آید. EMA یک میانگین متحرک نمایی است.

$$MACD = EMA12(C) - EMA26(C)$$

C قیمت پایانی کنونی است

Price Rate of Change

$$PROC(t) = 100 * \frac{C(t) - C(t - n)}{C(t - n)}$$

این اندیکاتور تغییرات قیمت را محاسبه می‌کند.

On Balance Volume

این اندیکاتور از معبود اندیکاتورهای است که از حجم به جای قیمت استفاده می‌کند.

$$OBV(t) = \begin{cases} OBV(t-1) + Vol(t) & \text{if } C(t) > C(t-1) \\ OBV(t-1) - Vol(t) & \text{if } C(t) < C(t-1) \\ OBV(t-1) & \text{if } C(t) = C(t-1) \end{cases}$$

OBV(t) نشان دهنده مقدار اندیکاتور در زمان t است

Vol(t) نشان دهنده حجم معامله در زمان t است

C(t) نشان دهنده قیمت در زمان t است.

Keras

کراس از کتابخانه‌های پیاده سازی شبکه‌های عصبی در پایتون است. از این کتابخانه برای پیاده سازی یک شبکه عصبی با ۳ لایه میانی استفاده شده است.

Logistic Regression

این پیش بینی بر اساس هر ۹۰ روز انجام می‌شود.

تابع Sigmoid

فرض کنید می‌خواهیم داده های ورودی خود را به دو دسته مثبت (۱) و منفی (۰) تقسیم کنیم. به این منظور از تابع زیر استفاده میکنیم که به تابع sigmoid معروف است:

$$g(z)=1/(1+e^{-z})$$

این تابع هر مقداری را به بازه ۰ تا ۱ تبدیل می کند، اگر ما ضرب ورودی و وزن ها را به این تابع بدهیم، می توانیم یک عدد احتمالی به دست بیاوریم که هرچه به یک نزدیکتر بود یعنی داده ورودی جزء دسته مثبت است و بالعکس.

Gradient Function

برای به روزرسانی وزن ها ابتدا از طریق تابع زیر مقدار انحراف را محاسبه کرده و سپس با یک ظریبی به وزن قبلی اضافه میکنیم:

$$\frac{\partial J(\theta)}{\partial \theta} = \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)}) x^{(i)}$$

نتیجه

برای اندازه گیری بازدهی این برنامه از چهار مقدار Accuracy, Precision و Recall استفاده می کنیم.

Accuracy: چند درصد مقادیر پیش بینی شده، درست پیش بینی شده اند.

Precision: اگر پیش بینی مثبت بود، چند درصد درست تشخیص داده شده است.

Recall: مواقعی که مقدار واقعی مثبت است، چند درصد آن درست تشخیص داده شده است.

ابتدا Confusion Matrix را تشکیل داده و سپس مقادیر را طبق فرمول های زیر حساب می کنیم.

$$\text{Accuracy} = (tp + tn) / (tp + tn + fp + fn)$$

$$\text{Precision} = tp / (tp + fp)$$

$$\text{Recall} = tp / (tp + fn)$$

```
➡ Accuracy: 0.772222222222223  
Precision: 0.845360824742268  
Recall: 0.5899280575539568
```