بسمه تعالی



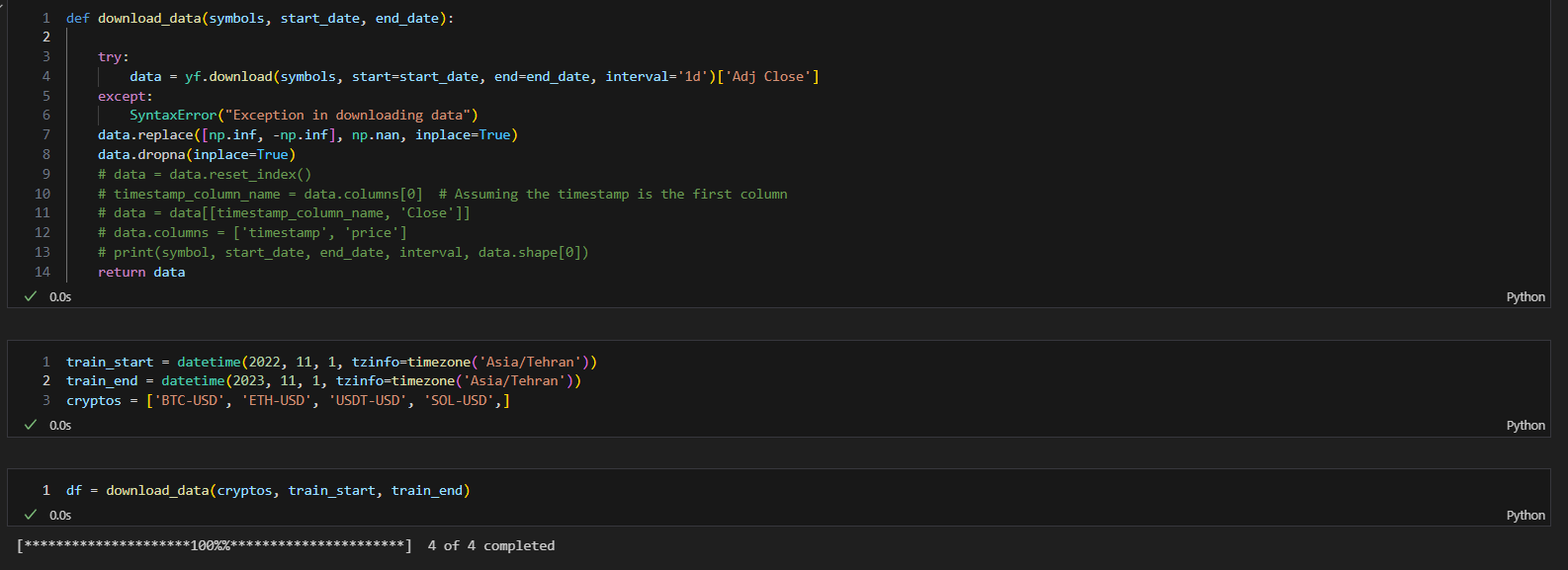
پروژه درس الگوریتم های معاملاتی

دکـتـر انتظاری

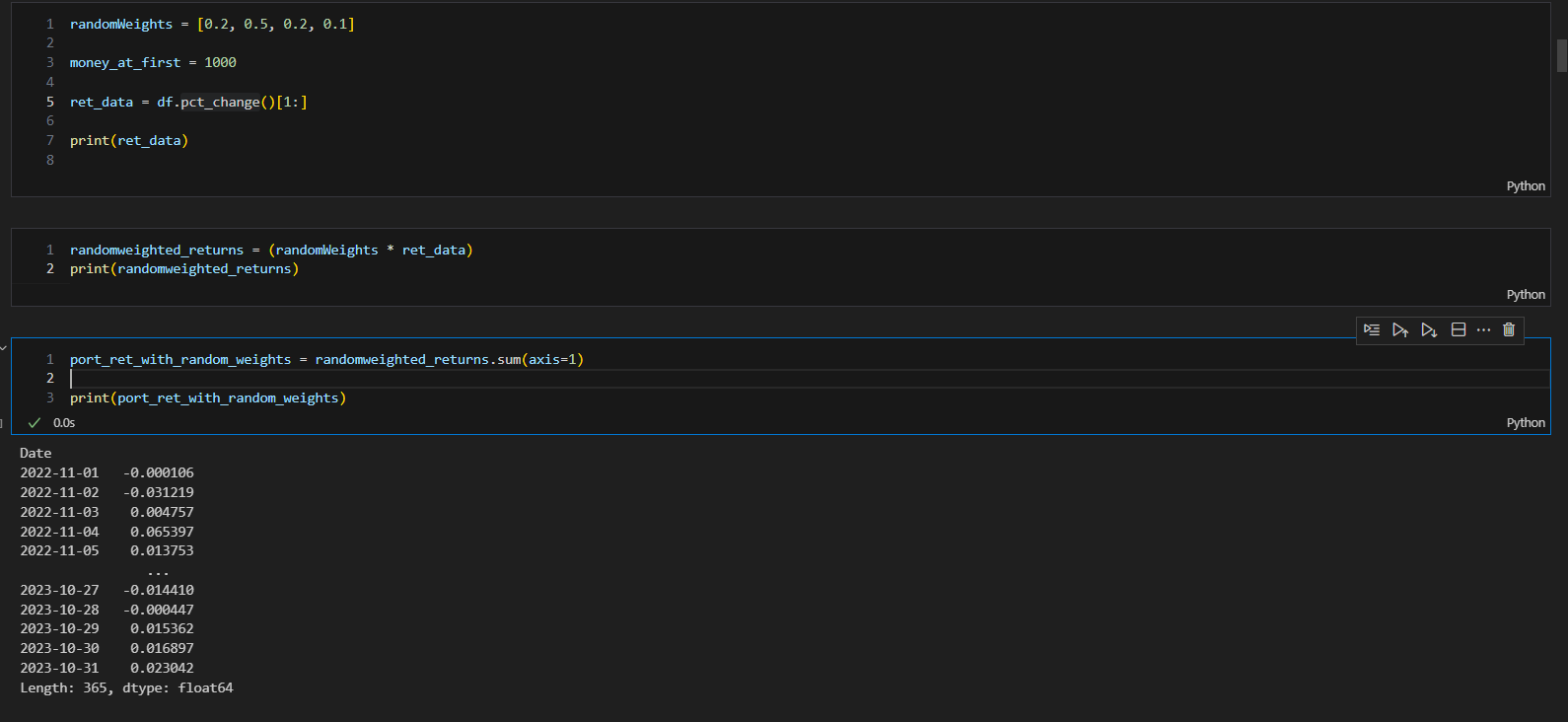
مـاهــان مــحـمودی - 98542436

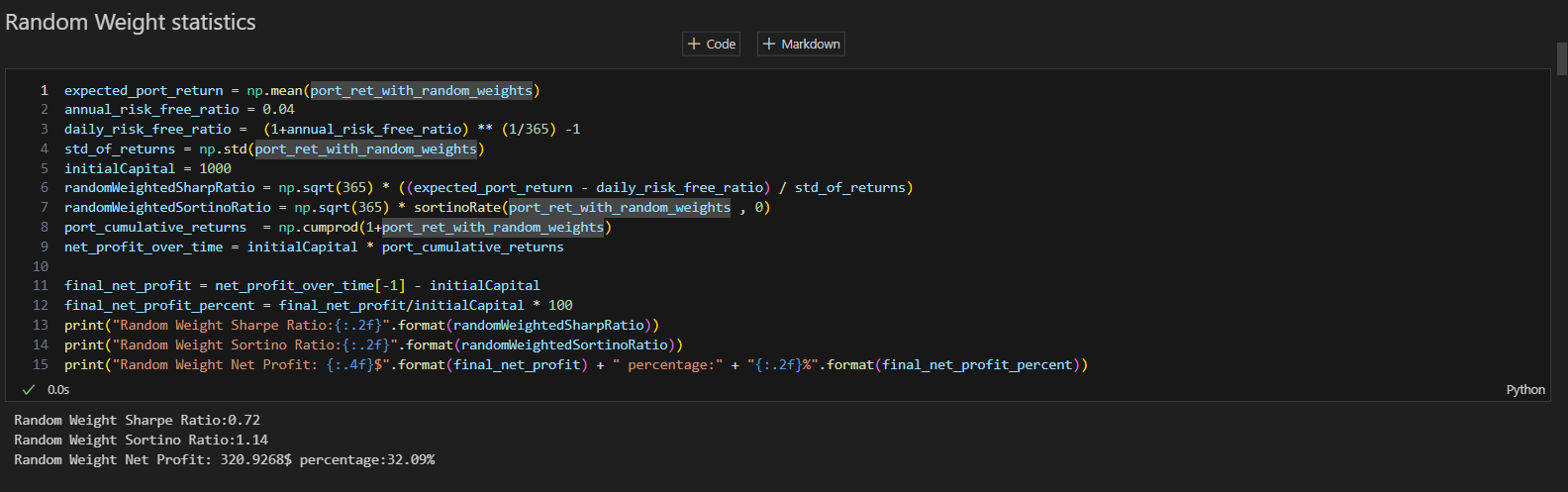
امیرعلی فرازمند – 99522329

دیتای روزانه را اینگونه از yahoo finance دانلود میکنیم

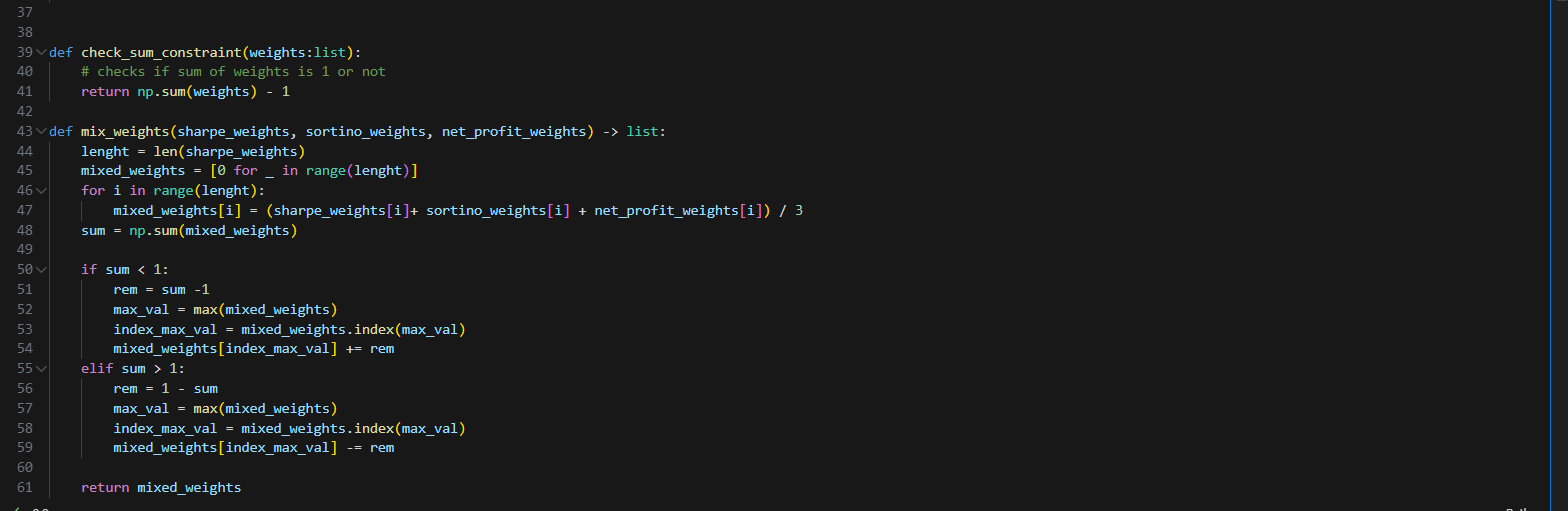
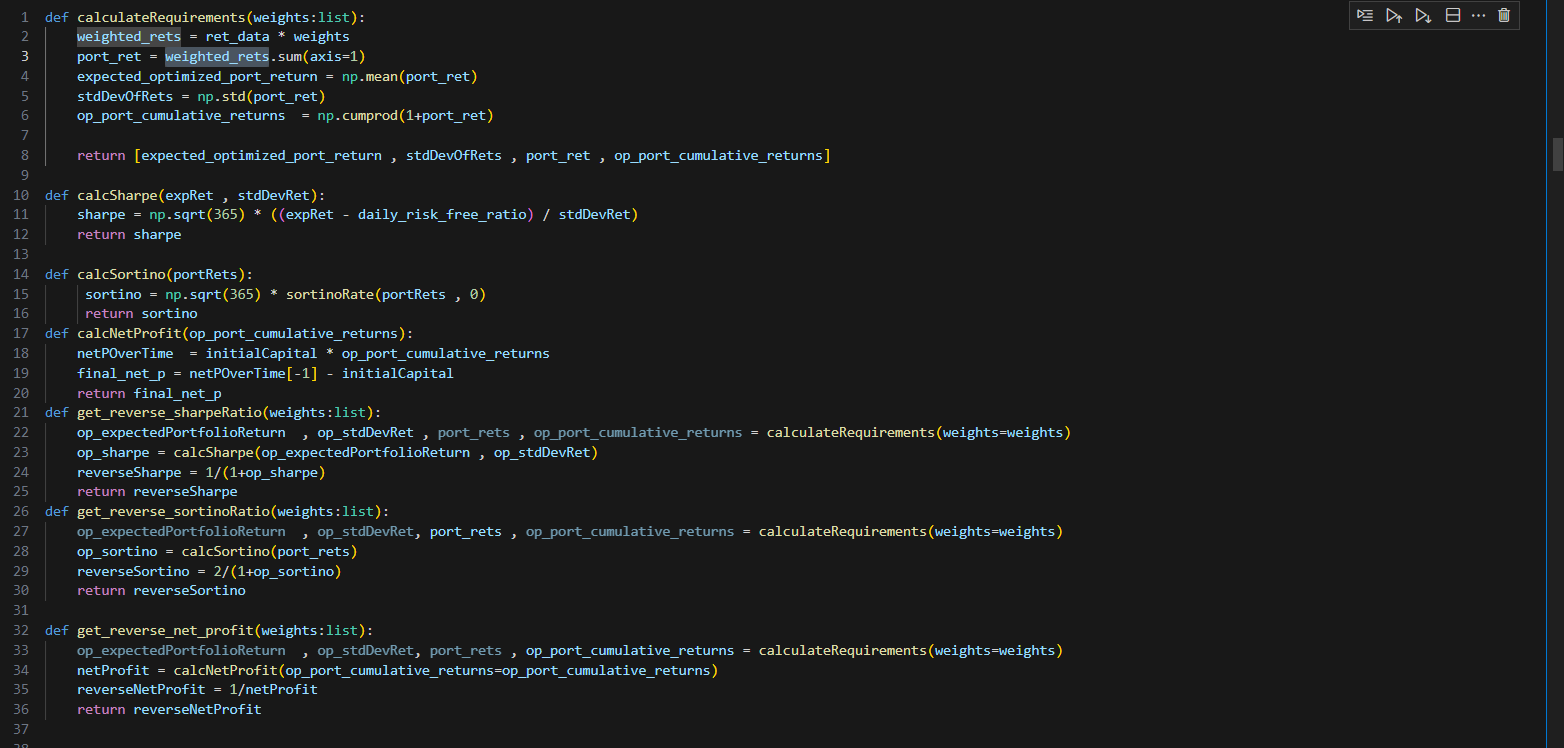


یک سری وزن رندوم خودمان به دلخواه تعریف میکنیم. دیتایمان را اینگونه تغیر میدهیم که از روز دوم به بعد برای close هر رمزارز نسبت به روز قبلی اش percent change میگیریم و در وزنش ضرب میکنیم. همچنین برای هر روز حساب میکنیم که در مجموع چند درصد صود یا ضرر کردیم.



معیار های گفته شده را به ازای همان وزن های رندوم حساب میکنیم تا بعدا با معیار هایی که بدست می آیند مقایسه کنیم. 

توابع تعریف شده برای پیدا کردن وزن های بهینه.

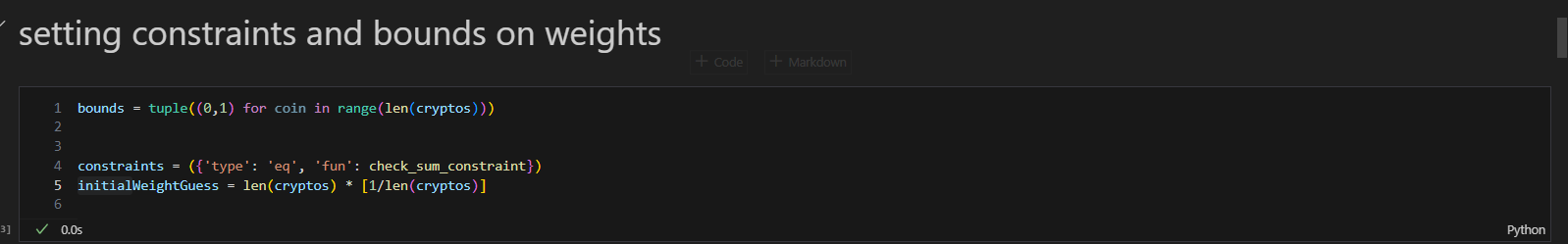
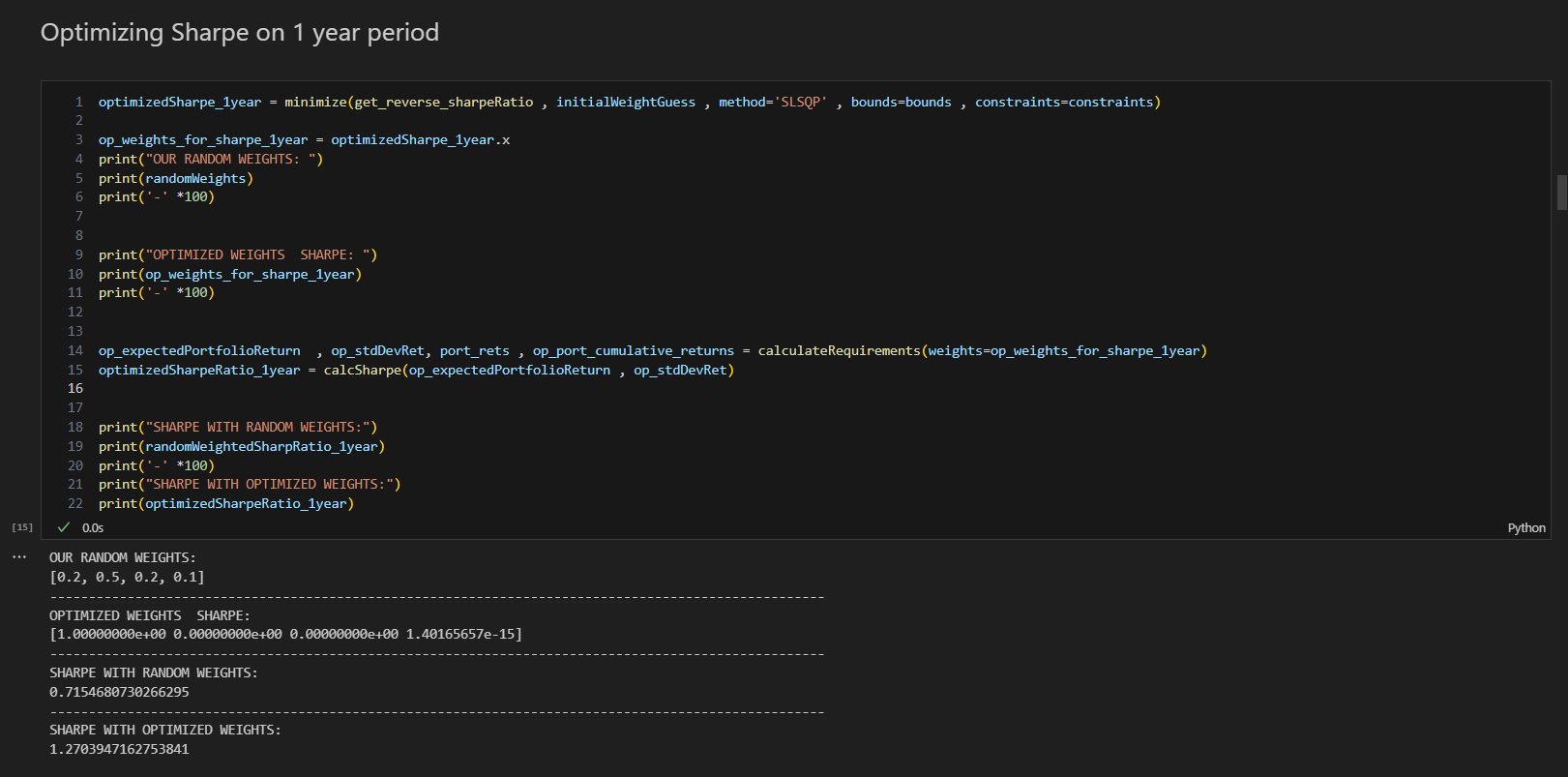


در تابع calculateRequirements ما اعدادی مثل انحراف معیار، return پورتفولیو را به ازای وزن های داده شده بر میگردانیم.

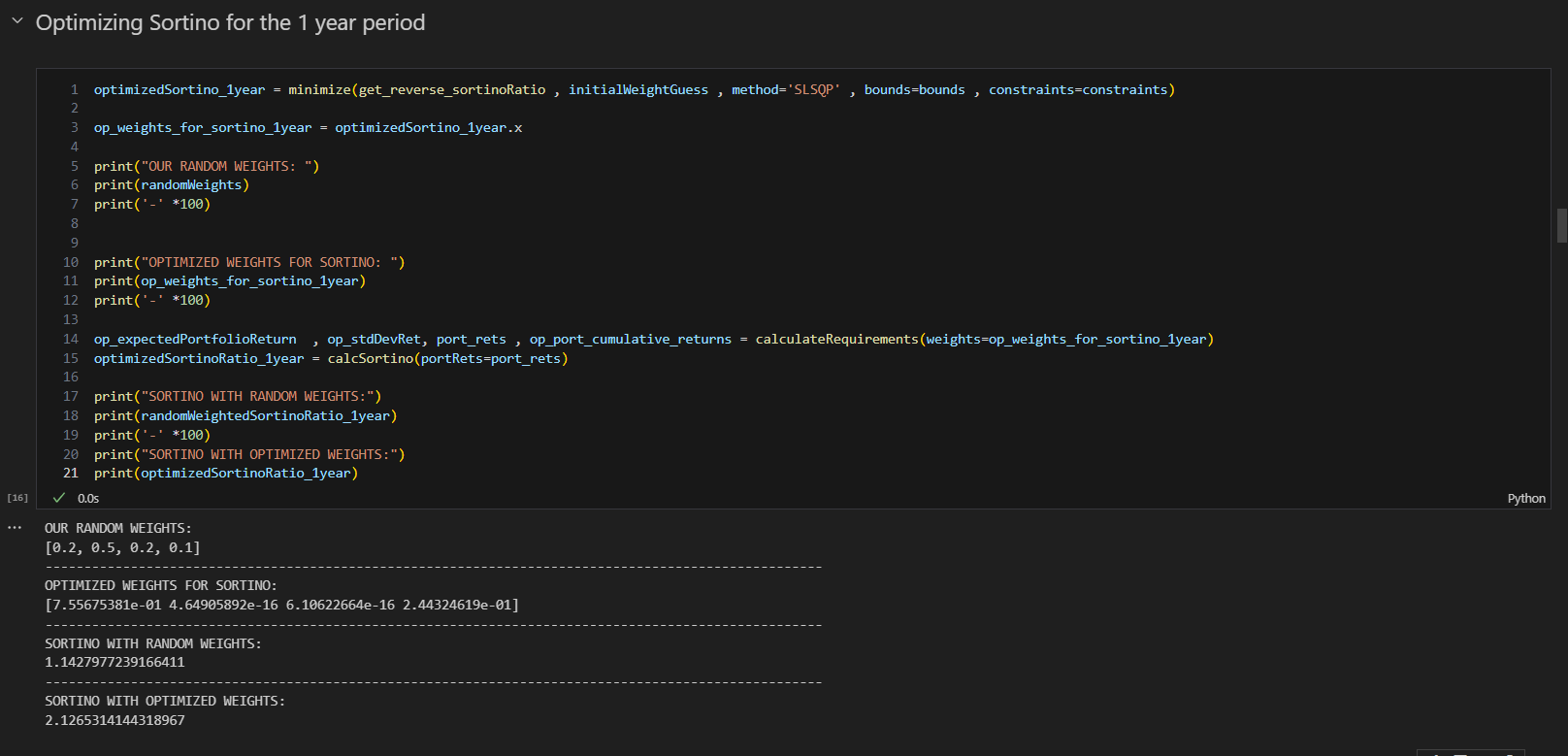
ما میخواهیم متریک های sharpe ratio, sortino ratio, net profit را ماکسیمایز کنیم.این کار را باید به کمک متد scipy.optimize.minimize وزن ها را طوری بدست بیاوریم که 1 / (1+metric) مینیمایز شود.

حال وزن های بهینه را برای متریک های مختلف در بازه ی 1 ساله ی گفته شده بدست می اوریم.

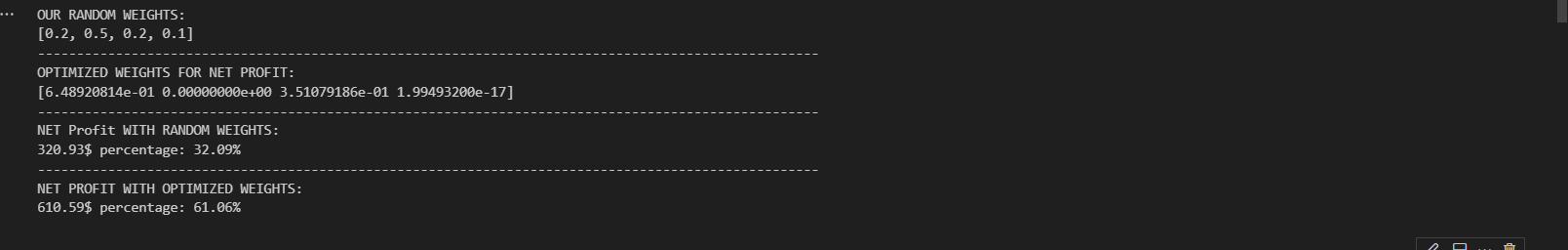
Sharpe ratio

Sortino ratio

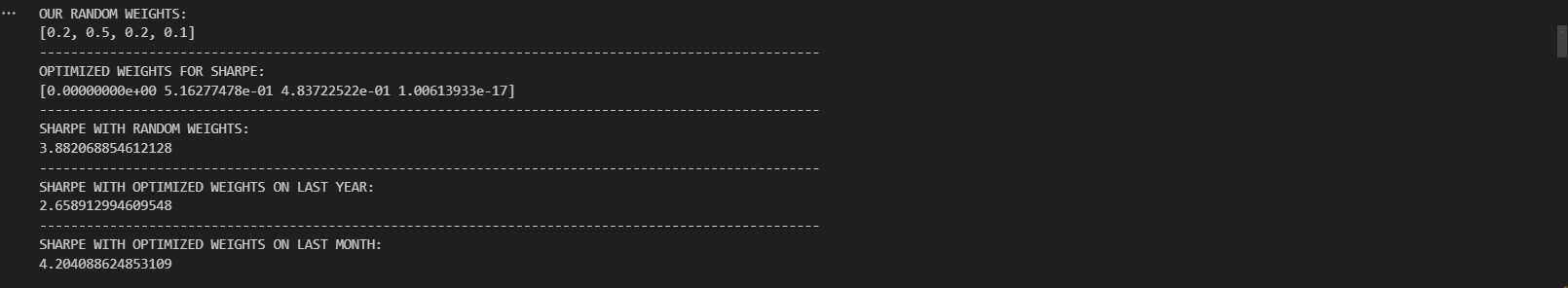


Net profit

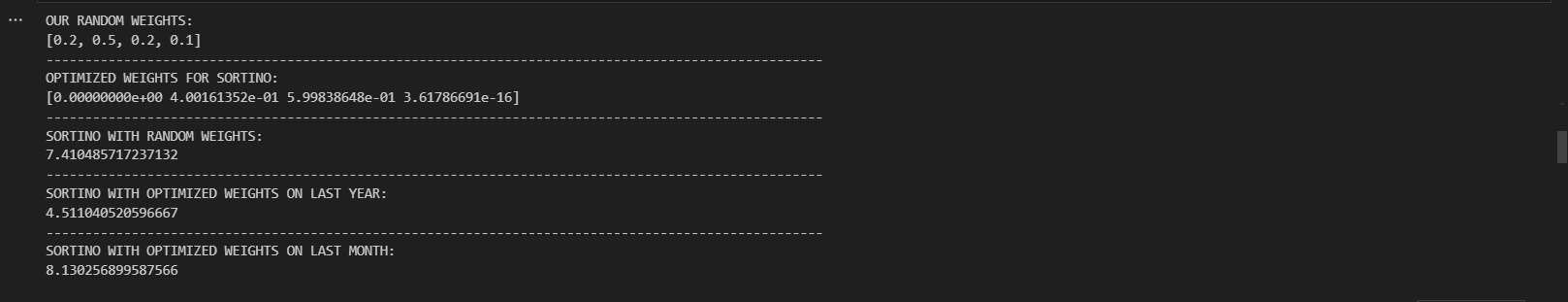


حال برای 1ماه گفته شده هم همین پروسه را تکرار میکنیم. با این تفاوت که این دفعه متریک ها را هم برای وزن های رندوم، هم برای وزن هایی که در یکسال قبلش بدست آورده ایم(از ترکیب وزن های بهینه) و هم وزن های بهینه ی صرفا همان یکماه که بدست می‌آیند بررسی میکنیم.

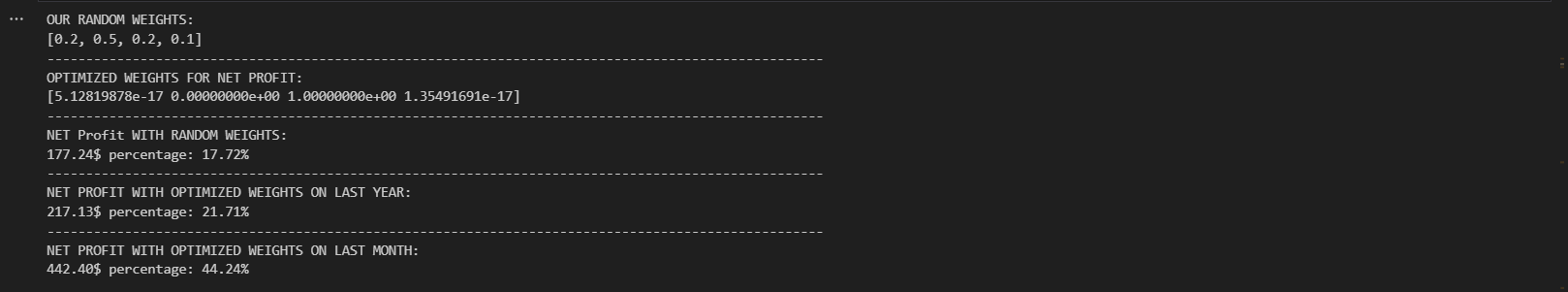
Sharpe ratio



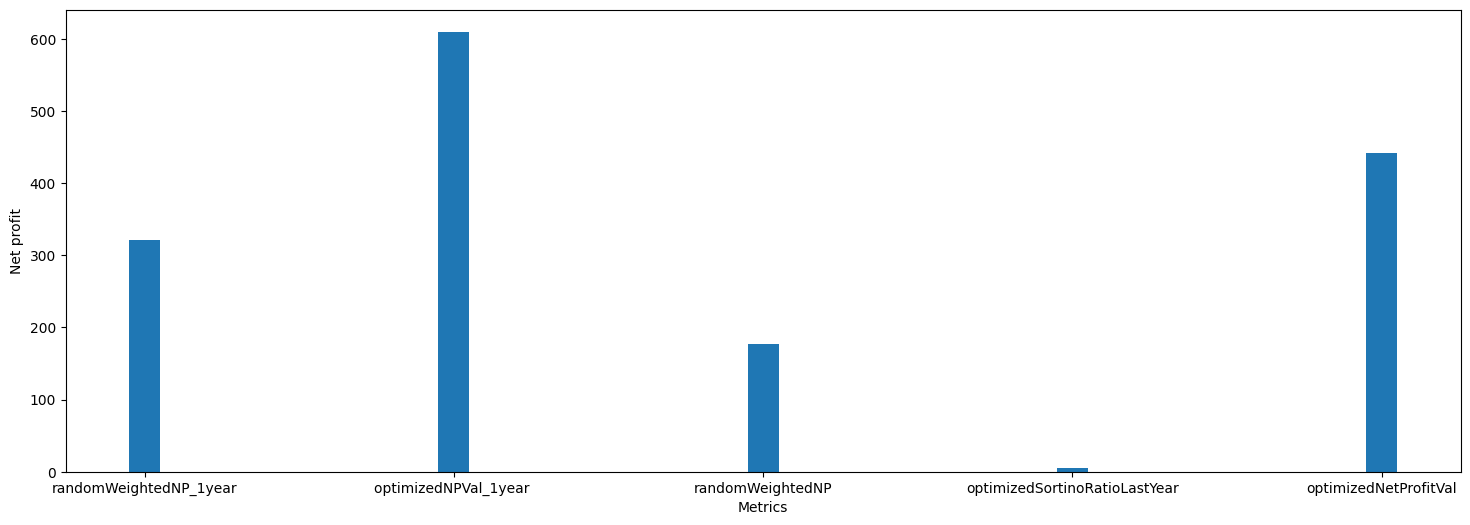
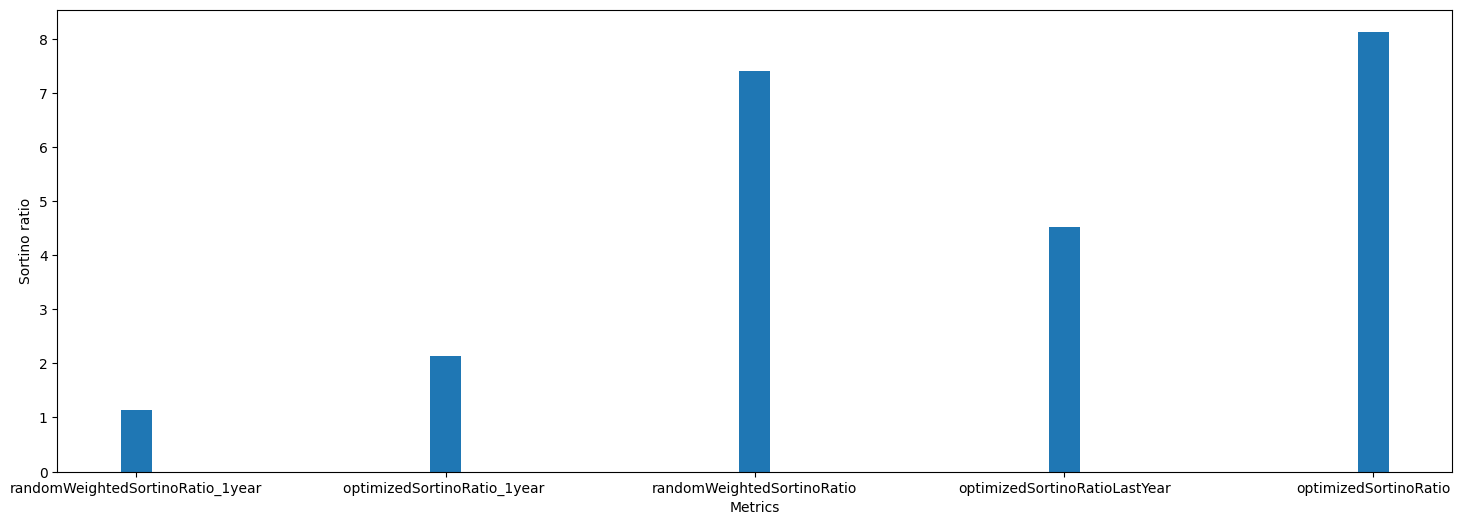
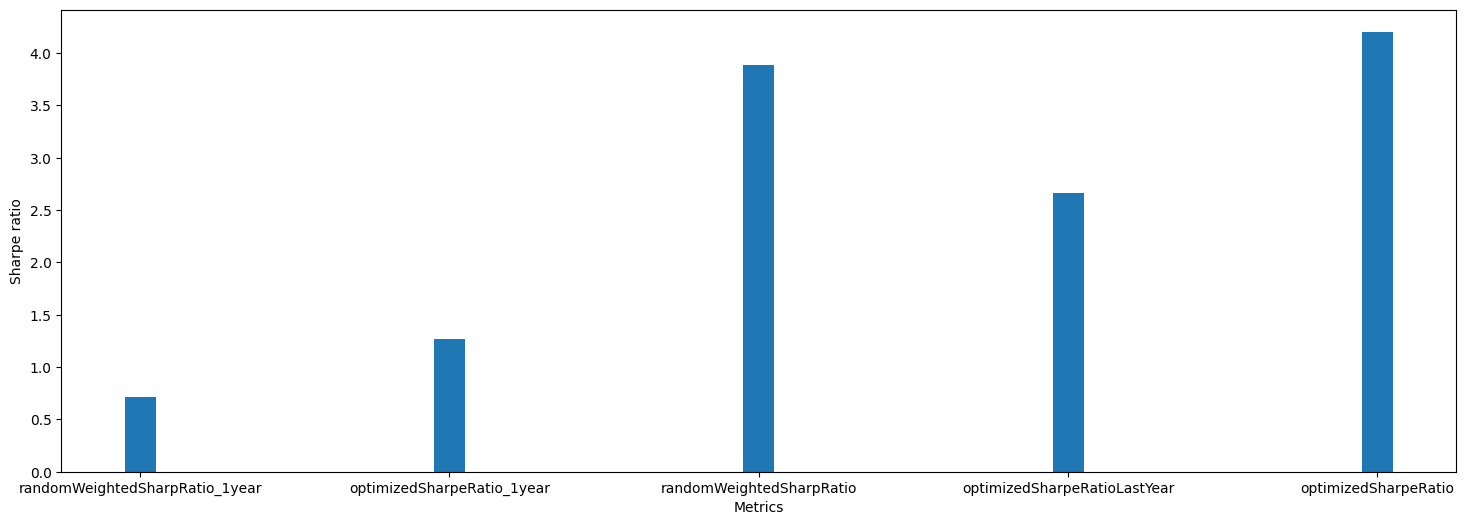
Sortino ratio



Net profit



در نمودار های زیر به ترتیب از چپ به راست متریک مورد نظر را برای: {(با وزن های رندوم برای سال قبل)، (با وزن های اپتیمایز شده برای سال قبل، روی دیتای سال قبل)، (با وزن های رندوم در ماه تست)، (با وزن های اپتیمایزشده بدست آمده در سال گذشته، بر روی ماه تست)، (اپتیمایز ترین حالت ممکن برای ماه تست) } را مشاهده میکنید.



پ.ن: تابع mix\_weights وزن های بهینه ای که بر اساس هر کدام از این متریک ها بدست میاوریم را بهش میدهیم و برای هر رمزارز وزن ها را میانگین میگیرد و در نهایت چک میکند که اگر جمع وزن ها برابر 1 نباشد از بیشترین وزن مقداری را کم یا زیاد میکند تا مجموع برابر با 1 شود. در آخر نوت بوک حالتی را گرفتیم که بجای اینکه تک تک متریک ها را با وزن هایی که ماکسیماز کردند آنهارا بررسی کنیم، با یک سری وزن که از این تابع گرفته میشوند بررسی کنیم. این بخش پیاده سازی شده اما ران نشده در نوت بوک، سل هایی هم که برای کار کردن نیاز بهشان داشت هم کامنت شده اند.

منابع:

* [پروژه ی مشابه](https://www.youtube.com/watch?v=D3O6riW4zDU&list=PLcFcktZ0wnNnqefRpFMS1k9_VlhVw7bzc&index=6) ([ریپو](https://github.com/areed1192/portfolio-optimization/blob/master/samples/portfolio_optimization.ipynb))
* [Optimizing Portfolio Sharpe Ratio and Weights Using SciPy Bounds, LinearConstraints & Minimize](https://medium.com/@adamchen564/test-22aef1296778)
* [Minimizing a function with scipy.optimize](https://www.einblick.ai/python-code-examples/minimizing-function-scipy-optimize-minimize/)
* کد های تمارین
* Chat GPT