به نام خدا

امیرمحمد کمیجانی ۹۹۵۲۲۰۳۲

آرین عبداللهی ثابت نژاد ۹۹۵۲۱۴۴۲

نکات:

فایل ارسال شده شامل ۲ فایل کد میباشد؛یک فایل برای تاریخ ۱ نوامبر ۲۰۲۲ تا ۱ نوامبر ۲۰۲۳ و فایل دیگر از تاریخ ۲ نوامبر در همین سال ها میباشد.

در این فایل ، ابتدا توضیحات کد و در انتها نمایش و مقایسه نتایج وجود دارد.

منابع کمکی در صورت بهره گیری ؛ بالای هر سِل آورده شده اند

توضیحات کد:

در سِل اول ، کتابخانه های مورد نیاز را ایمپورت کرده ایم.

در سِل دوم ، متغیرهای مورد نیاز در سوال تعریف شده اند.
که شامل رمزارز های مورد نظر،تاریخ ابتدا و انتها و مقدار سرمایه اولیه که طبق داک ۱۰۰۰ دلار تعریف شده است.

در سِل سوم ، دیتای مورد نظر را از یاهو فایننس فِچ کرده ایم و در قالب یک دیتافریم قرار داده ایم.(قیمت کلوز را برای فچ کردن مدنظر گرفتیم و از تایم فریم ۱ ساعته استفاده کرده ایم)

در سِل چهارم ، توابع مربوط به هر یک از معیارهای مسئله یعنی sharpe ratio و sortino ratio , net profit را پیاده سازی میکنیم.

برای تابع sharp ratio مقدار ریسک بر اساس منبعی که در بالای سِل ذکر کردم قرار دادم.

reward, how much risk am I taking on?

$$Sharpe\ Ratio = \frac{R_p - R_f}{\sigma_p}$$

where:

R_p = return of portfolio

R_f = risk-free rate

σ_p = standard deviation of the portfolio's excess return

مقدار ریترن بالانس اکانت خود را محاسبه میکنیم سپس میانگین و انحراف معیار میگیریم و با توجه به مقدار ریسک معیار sharpe ratio را محاسبه میکنیم.

برای تابع sortino ratio روزهایی که ریترن منفی داریم را مدنظر قرار میدهیم.

$$\text{Sortino Ratio} = \frac{R_p - r_f}{\sigma_d}$$

where:

R_p = Actual or expected portfolio return

r_f = Risk-free rate

σ_d = Standard deviation of the downside

که شبیه به شارپ ریشیو محاسبه میشود فقط برای انحراف معیار شرط منفی بودن ریترن را در نظر میگیریم.

برای محاسبه net profit یا همان سود خالص مقداری که در انتها در بالانس اکانت خود داشته ایم را از مقدار سرمایه اولیه که وارد کردیم یعنی ۱۰۰۰ دلار کم میکنیم.

در قسمت بعدی استراتژی buy and hold را پیاده سازی کرده ایم.

ابتدا مقدار ریترن یا درصد تغییرات روزانه را محاسبه میکنیم(با استفاده از تابع pct_change()) سپس ریترن تجمعی را محاسبه میکنیم.و درنهایت وزن ها را در ریترن تجمعی ضرب میکنیم و مقدار بالانس را بر اساس آن تغییر میدهیم.

Date	BTC-USD	ATOM-USD	USDT-USD	DOGE-USD
2022-11-02 00:00:00+00:00	0.984097	0.942505	1.000072	0.896581
2022-11-03 00:00:00+00:00	0.986562	0.957671	1.000083	0.859842
2022-11-04 00:00:00+00:00	1.032314	1.066089	1.000152	0.885709
2022-11-05 00:00:00+00:00	1.038926	1.091571	1.000195	0.872439
2022-11-06 00:00:00+00:00	1.021538	1.026547	1.000166	0.803830
...	...	...	...	...
2023-10-27 00:00:00+00:00	1.655326	0.498401	1.000333	0.476107
2023-10-28 00:00:00+00:00	1.664101	0.509315	1.000329	0.483963
2023-10-29 00:00:00+00:00	1.686015	0.518712	1.000463	0.486474
2023-10-30 00:00:00+00:00	1.684252	0.574713	1.000648	0.488515
2023-10-31 00:00:00+00:00	1.692327	0.562498	1.000391	0.478885

تصویر دیتافریم بعد از اعمال ریترن تجمعی یا culmulative return

```
weights = [0.3 , 0.25 , 0.4 , 0.05]
```

که در این وزن ها ضرب میشود.

سپس از متد `sum(axis=1)` استفاده میکند که بعد از ضرب وزن ها در ریترن تجمعی هر روز ، جمع ریترن تجمعی وزن دار هر چهار کریپتو را محاسبه میکند و به مقدار بالانس ما اضافه میکند.

سپس استراتژی buy and hold را ران میکنیم و معیار های مورد نظر را بعد از اعمال این استراتژی بر روی بالانس اکانت خود محاسبه کرده ایم.

** وزن ها را به صورت رندوم و دستی وارد کرده ایم.

در قسمت آخر ، وزن ها را طوری یافتیم که برای هر معیار بهترین نتیجه ثبت شود.

برای توضیح تابع minimize و استفاده از متد SLSQP داخل آن از داک خود این کتابخانه استفاده کردیم:

Method SLSQP uses Sequential Least SQuares Programming to minimize a function of several variables with any combination of bounds, equality and inequality constraints. The method wraps the SLSQP Optimization subroutine originally implemented by Dieter Kraft [12]. Note that the wrapper handles infinite values in bounds by converting them into large floating values.

که روش محاسبه در آن توضیح داده شده و گفته شده که به سمت مقادیر بزرگتر میبریم که بهتر میشود.

روش کار با تابع minimize به صورتی است که باید متد های مربوط به سه معیار را یک بار دیگر تعریف کنیم و مقادیر منفی آنها را خروجی دهیم. و باید مقادیر وزن را تعریف کنیم.مقادیر وزن را همانند قبل تعریف کردیم و سپس اجازه دادیم که به مقدار optimal خود با استفاده از تابع ذکر شده برسند.

یک سری مقادیر مانند bounds هم داریم که نشان دهنده رنج مقادیر میباشد که انها

را به ۰ و ۱ محدود کرده ایم. و البته متغیر constraints که مشخص میکند مجموع مقادیر نباید بیشتر از ۱ باشد.

سپس بعد از به دست آوردن وزن ها مقادیر را با محاسبه ریترن تجمعی وزن دار مقادیر بهینه برای هر یک از معیار های sharp ratio , sortino ratio , net profit محاسبه میکنیم(با استفاده از fun)

** مجموع وزن هایی که به دست آوردیم باز هم برابر ۱ بود.

نتایج دو کد با تاریخ های متفاوت:

در هر دو کد ، با اعمال استراتژی گفته شده و به دست آوردن بهترین وزن ها ، توانستیم مقادیر بسیار بهتری برای معیارهای خواسته شده به دست آوردیم.

با اعمال تفاوت در تاریخ ها نتایج متفاوت شد و تفاوت بسیاری داشت ؛ مقادیر وزن ها دچار تفاوت بسیاری شدند و به طور کلی استفاده از این الگوریتم و متد در برابر تغییرات زمانی sensitive بودند و واکنش قابل توجهی نشان دادند.
همچنین مقادیر وزن ها در sharp , sortino ratio دچار تغییرات زیادی بودند چون نوسانات را هم حساب میکنند.

** نکته : میزان تاثیر اصلی بر روی نت پرافیت فعلی تا حد قابل توجهی به بیتکوین برمیگردد به دلیل حجم بالای آن که منطقی هم هست.در نتیجه اگر رمز ارز ها را از لحاظ حجم معاملاتی نزدیک به هم انتخاب کنیم شاید به ثبات بیشتری در نتایج برسیم و با تغییرات کوچک در زمان شاهد تغییرات زیاد در معیار هایمان نشویم.