**\*\*\* فایل Result که به تبلو میدیم نباید تفکیک نماد داشته باشه، چون اینجوری نحوه‌ی میانگین‌گیریش به هم میخوره.**

**روش Sparse در بهینه‌سازی سبد سرمایه‌گذاری**

بهینه‌سازی سبد سرمایه‌گذاری همواره یکی از موضوعات مهم در مدیریت مالی و اقتصاد بوده است. روش‌های مختلفی برای بهینه‌سازی سبد سرمایه‌گذاری وجود دارد که هر کدام به نوعی تلاش می‌کنند تا بازدهی پرتفوی را به حداکثر و ریسک را به حداقل برسانند. یکی از این روش‌ها، استفاده از تکنیک‌های sparse است که در این بخش به تشریح آن می‌پردازیم.

**مفهوم Sparse**

در بهینه‌سازی، اصطلاح sparse به حالتی اشاره دارد که بسیاری از عناصر یک بردار یا ماتریس صفر باشند. در زمینه بهینه‌سازی سبد سرمایه‌گذاری، این مفهوم به دنبال ایجاد یک پرتفوی است که فقط شامل تعداد محدودی از دارایی‌ها باشد. این کار باعث می‌شود که پرتفوی ساده‌تر و مدیریت آن آسان‌تر شود و هزینه‌های معامله و پیگیری نیز کاهش یابد.

**ضرورت استفاده از روش Sparse**

روش‌های سنتی بهینه‌سازی مانند روش مارکویتز ممکن است منجر به انتخاب پرتفوی‌هایی شوند که شامل تعداد زیادی از دارایی‌ها هستند. این امر می‌تواند منجر به مشکلات عملی در مدیریت پرتفوی و هزینه‌های بالای معاملاتی شود. روش‌های sparse تلاش می‌کنند تا با محدود کردن تعداد دارایی‌های انتخاب شده در پرتفوی، این مشکلات را کاهش دهند و یک پرتفوی بهینه با ساختاری ساده‌تر و قابل مدیریت‌تر ایجاد کنند.

**روش‌های رایج در بهینه‌سازی Sparse**

* Lasso Regression: یکی از روش‌های معروف برای دستیابی به پراکندگی (sparsity) در پرتفوی استفاده از Lasso Regression است. این روش، یک نوع رگرسیون خطی است که یک جریمه L1L1 به مدل اضافه می‌کند. این جریمه باعث می‌شود بسیاری از وزن‌های پرتفوی به صفر میل کنند و در نتیجه، تعداد دارایی‌های انتخاب شده کاهش یابد.
* Elastic Net: ترکیبی از جریمه‌های L1L1 و L2L2 است که از مزایای هر دو بهره می‌برد. این روش، علاوه بر دست‌یابی به پراکندگی، به حفظ ثبات و دقت مدل نیز کمک می‌کند.
* Subset Selection: در این روش، به جای بهینه‌سازی پرتفوی با تمام دارایی‌ها، زیرمجموعه‌ای از دارایی‌ها انتخاب می‌شود که بهترین عملکرد را دارند. این کار معمولاً از طریق الگوریتم‌های جستجوی حریصانه (greedy algorithms) یا الگوریتم‌های فراابتکاری مانند ژنتیک یا تبرید شبیه‌سازی شده انجام می‌شود.

**مراحل اجرای بهینه‌سازی Sparse**

* تعیین معیارهای بهینه‌سازی: ابتدا باید معیارهای بهینه‌سازی مشخص شود، مانند بازده مورد انتظار، ریسک و میزان پراکندگی مطلوب.
* انتخاب مدل مناسب: بر اساس نیازها و ویژگی‌های داده‌ها، یکی از روش‌های sparse مانند Lasso، Elastic Net یا Subset Selection انتخاب می‌شود.
* محاسبه و بهینه‌سازی: مدل انتخابی با استفاده از داده‌های تاریخی و معیارهای مشخص شده، محاسبه و بهینه‌سازی می‌شود. هدف این است که پرتفویی با تعداد محدودی از دارایی‌ها و عملکرد بهینه به دست آید.
* ارزیابی و بازنگری: پرتفوی بهینه شده با استفاده از معیارهای ارزیابی مختلف مانند نسبت شارپ، نسبت‌ترینور و غیره ارزیابی می‌شود و در صورت نیاز، بازنگری و بهبود می‌یابد.

**مزایا**

* سادگی و قابلیت مدیریت: پرتفوی‌های ایجاد شده ساده‌تر و قابل مدیریت‌تر هستند.
* کاهش هزینه‌های معاملاتی: به دلیل تعداد محدود دارایی‌ها، هزینه‌های معاملاتی کاهش می‌یابد.
* کارایی بالا: در بسیاری از موارد، این روش‌ها می‌توانند به پرتفویی با بازدهی مطلوب و ریسک کمتر دست یابند.

**معایب**

* پیچیدگی محاسباتی: برخی از روش‌های sparse به محاسبات پیچیده نیاز دارند.
* حساسیت به انتخاب مدل: انتخاب مدل مناسب و تنظیمات بهینه می‌تواند چالش‌برانگیز باشد.

**تحلیل کد**

تابع، ورودی‌ای از نوع DataFrame می‌گیرد که شامل اطلاعات نماد (symbols)، بازدهی (returns)، و ریسک (risks) است. هدف این تابع ایجاد یک سبد سرمایه‌گذاری بهینه است که تعدادی محدود از دارایی‌ها را انتخاب کند و ریسک و بازدهی آن را محاسبه نماید.

**تعریف توابع داخلی**

* project\_simplex(w): این تابع یک بردار وزن‌ها را به داخل یک سیمپلکس (مجموعه‌ای که در آن مجموع عناصر برابر ۱ است و همه عناصر غیرمنفی‌اند) فرافکن می‌کند.
* threshold(w, k): این تابع k عنصر بزرگتر در بردار وزن‌ها را نگه می‌دارد و بقیه را صفر می‌کند. این دقیقاً همان مفهوم sparsity را پیاده‌سازی می‌کند، زیرا تنها تعداد محدودی از دارایی‌ها وزن مثبت خواهند داشت.

**پارامترها و مقداردهی اولیه**

* k = 10: تعداد دارایی‌های انتخاب شده.
* r = 0.001: حداقل بازدهی مورد انتظار.
* w: بردار وزن‌های اولیه که به طور یکنواخت تقسیم شده است.
* سایر پارامترها مانند نرخ یادگیری (alpha)، تحمل خطا (tol)، و حداکثر تعداد تکرارها (max\_iter).

**حلقه بهینه‌سازی**

* وزن‌ها به روزرسانی می‌شوند.
* وزن‌ها به سیمپلکس فرافکن می‌شوند.
* تنها k عنصر بزرگتر نگه داشته می‌شوند.
* اگر بازدهی پرتفوی کمتر از حداقل بازدهی باشد، نرخ یادگیری نصف می‌شود و وزن‌ها بازگردانی می‌شوند.
* حلقه تا زمانی که تغییر وزن‌ها کمتر از تحمل خطا باشد یا حداکثر تعداد تکرارها انجام شود، ادامه می‌یابد.

**نتیجه‌گیری**

این کد در دو جا از مفهوم sparse بهره می‌برد:

* تابع threshold(w, k): تنها k عنصر بزرگتر در بردار وزن‌ها را نگه می‌دارد و بقیه را صفر می‌کند. این دقیقاً همان روشی است که در subset selection توضیح دادیم.
* بهینه‌سازی وزن‌ها با استفاده از فرافکنی سیمپلکس و به روزرسانی وزن‌ها.

بنابراین، این تابع به روش Subset Selection برای ایجاد یک پرتفوی sparse از میان دارایی‌ها اشاره دارد. این روش از این لحاظ مشابه است که تنها تعداد محدودی از دارایی‌ها در پرتفوی نهایی وزن غیرصفر خواهند داشت.