

GTAA-Backtesting Tool von Lukas Florian Richter ad Nemanja Cerovac

Lukas Florian Richter, Nemanja Cerovac

Carl-Friedrich-Gauß-Gymnasium Frankfurt (Oder)

13. März 2024

Inhaltsverzeichnis

1	Problemstellung und Planung	1
2	Analyse	3
3	Implementierung	7
4	Testläufe	8

1 Problemstellung und Planung

Der Handel mit Aktien ist häufig sehr risikobehaftet. Während die Märkte im langfristigen Trend tendenziell steigen und Verluste durch längere Haltedauern ausgeglichen werden können, sind kurz- und mittelfristig enorme Schwankungen möglich. Diese Volatilität stellt insbesondere für private Anleger mit begrenztem Informationszugang im Vergleich zu institutionellen Großinvestoren und dem somit auch psychologischen Faktor der Risikowagnis und Ausdauer ein Problem dar.

Perfekt wäre somit eine Strategie, welche mit geringerem Risiko dennoch annähernd identische Gewinne wie die Buy-and-Hold-Taktik (nachfolgend B&H) liefert. Oder, um es wie Meb Faber auszudrücken, „equity-like returns with bond-like volatility“. Dieser gilt als Erfinder der GTAA - Global Tactical Asset Allocation (siehe hierzu https://papers.ssrn.com/sol3/papers.cfm?abstract_id=962461). Deren Grundlage ist es einen Pool aus möglichst geringfügig korrelierenden (taktisch) Wertpapieren verschiedener Klassen (Global) und Regionen oder Sektoren derart zusammenzustellen, dass möglichst in jedem wirtschaftlichen Umfeld eines dieser Wertpapiere gut performt.

Es wird eine Verteilung des Gesamtinvestments auf diese Wertpapiere festgelegt

(Allocation) und periodisch überprüft, ob ihr aktueller Preis über einem festgelegten dynamischen Mittel (Simple Moving Average, SMA) liegt, beispielsweise der 200-Tage-SMA oder 252-Tage-SMA (ein Jahr entspricht meist 252 Handelstagen). Jedes Asset, welches überhalb dieses Wertes notiert, wird seinem festgelegten Anteil am Gesamtinvestment entsprechend gekauft und bis zum Ende der Periode im Portfolio behalten, wo wiederum erneut gewichtet wird. Für alle Investments, die unter oder auf dem Durchschnitt liegen, wird der zugewiesene Betrag bis zur nächsten Periode nicht investiert.

Dieses Konzept basiert darauf, dass der sogenannte Moment die einzige langfristig andauernde Börsenanomalie und Ineffizienz der Märkte sei, wonach historisch erfolgreiche Investments auch in absehbarer Zukunft ihre Tendenz beibehalten (siehe hierzu auch https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2042750). Gleichzeitig ist der Anwender während längerer Verlustperioden durch das Unterschreiten des dynamischen Durchschnittes nicht investiert und verkürzt sowie verringert hierdurch seine Drawdowns. Die Strategie eliminiert so die psychologische Nervositäts-Komponente, indem durch ein automatisiertes Entscheidungssystem die Unsicherheit des Menschen ausgeschaltet wird.

Gleichzeitig ruft dieses starre System eine Inflexibilität, welche bei inkorrekten Grundannahmen zur Sinnhaftigkeit der Allokation zu finanziellem Schaden führen kann. Umso wichtiger ist es also, diese mit Hilfe historischer Daten zu untermauern, also einen Backtest durchzuführen.

Ziel dieses Projektes soll also ein Programm sein, welches:

- (1) den Vergleich der GTAA-Strategie mit der Buy- and Hold-Strategie
- (2) den Vergleich zweier GTAA-Strategien miteinander

erlaubt. Dabei soll der Benutzer über ein graphisches Interface eine Zusammenstellung von Wertpapieren mit selbst gewählter Wichtung erstellen können sowie in der Lage sein, den SMA selbst auszuwählen. Aus statistischen Gründen und zur Vermeidung des Selection Bias, wobei für die Untersuchung zufällig ein vorteilhafter Zeitraum gewählt wird, wird für die Betrachtung stets der größtmögliche Zeitraum betrachtet, für welchen die notwendigen Daten zu den Wertpapieren verfügbar sind. Ein anderer Bias, der Survivorship Bias, kann nicht eliminiert werden, da die notwendigen Daten von der yahooFinance-API stammen und diese nur Wertpapiere führt, welche noch existieren. Das heißt, insolvente Unternehmen oder anderweitig „erfolglose“ Wertpapiere können im Backtest nicht vorkommen und diesen somit positiv verfälschen.

Zudem ist es wünschenswert, neben der einfachen Performance auch weitere Metriken und Indikatoren, wie die Volatilität oder Sharpe Ratio der Strategie und ihrer Benchmark zu berechnen, um eine optimale Informationsgrundlage zur Wahl der passenden Strategie zu schaffen. Diese Metriken sowie die Performance sollten mit Hilfe von passenden Diagrammen und Tabellen in einem Spreadsheet als eigene Da-

tei dargestellt werden, sodass der User wiederholt einfach darauf Zugriff hat.

2 Analyse

Diese Anforderung lassen sich technische folgendermaßen ausdrücken:

Es müssen Informationen über vom User gewählte Wertpapiere erlangt werden. Diese müssen hierzu eindeutig identifiziert werden können. Dafür bietet sich deren Ticker-Symbol an. Dieses ist zu jedem gegebenen Zeitpunkt eindeutig definiert, wobei es historisch bereits von anderen Wertpapieren belegt gewesen sein kann. Da der Survivorship Bias aber in Kauf genommen wird, ist die Beschränkung auf aktuell gelistete Wertpapiere akzeptabel.

Als Quelle von Daten möglichst vieler Ticker-Symbole eignet sich yahooFinance, auf dessen APIs per yfinance-Library in Python zugegriffen werden kann. Die Informationen werden dann als Pandas Dataframe zurückgegeben und anschließend aufbereitet. Verfügbar sind somit Tageshöchst-, -tief-, -abschluss- und -anfangskurse, wovon mit Hilfe des Abschlusskurse der SMA berechnet wird. Da bspw. für einen 200d-SMA 200 initiale Kurse benötigt werden, können die ersten 200 Einträge nicht mit in die Performance-Berechnung einfließen, da hier der vollständige SMA fehlt. Die Performance-Berechnung findet auf zwei Ebenen statt.

Einerseits werden für jedes einzelne Wertpapier einer Strategie einzeln die Daten von yfinance abgerufen und die Performance dieses einen Wertpapiers berechnet. Ob in einer Periode investiert wird, oder nicht, wird zu Periodenbeginn über den Vergleich Tagesabschlusskurses mit dem eigens berechneten SMA, welcher dem Pandas Dataframe als Spalte angefügt wird, bestimmt. Für nicht-investierte Zeiträume wird die Wertpapierperformance als stagnant angesehen. Nicht-Handelstage werden mit dem Vortageskurs „aufgefüllt“. Die Performance selbst ist keine neue Spalte des Dataframes, da diese durch Nicht-Handelstage Daten enthält, für welche das Dataframe keine Zeilen enthält. Stattdessen wird sie mittels einer Datums- und einer Performance-Liste getrackt, die am Ende der Berechnung zu einer Pandas Series zusammengefügt werden.

Andererseits müssen für die Performance der Gesamt-Strategie die Entwicklungen aller einzelnen Investments mit ihren jeweiligen Gewichtungen zusammengefügt werden, was über Multiplikation der jeweiligen Pandas Series mit ihrem Gewicht und anschließende Addition zueinander geschieht. Insgesamt werden hierfür also zwei Klassen benötigt:

Entry für die einzelnen Wertpapiere. Jeder Entry benötigt als Attribute einen validen Ticker sowie eine ID, mit welcher wiederum im Portfolio dieser Entry identifiziert werden kann. Wohlgermerkt beschreibt ein Entry kein Wertpapier, sondern den Eintrag eines Wertpapiers in einem bestimmten Portfolio. Es handelt sich also um eine Komposition. Außerdem muss es eine Funktion zur Performance-Berechnung des

Entrys geben.

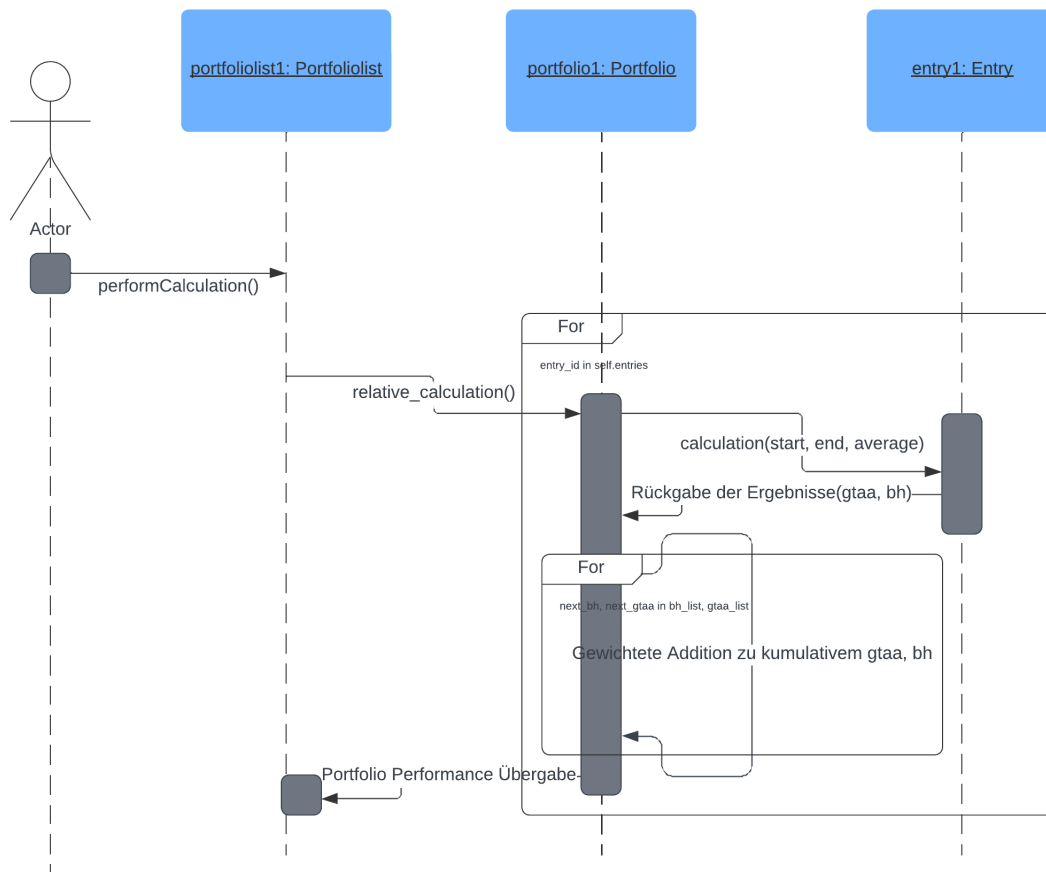
Das Portfolio wiederum enthält ein Dictionary mit Entries, deren Keys Integer sind. Der Hintergrund hierfür ist die Möglichkeit, in einer späteren Erweiterung des Projektes auch nachträglich noch die Ticker der Entries in einem Portfolio zu bearbeiten, zu löschen oder neue Entries hinzuzufügen. Gleiches gilt für die in einem weiteren Dictionary mit identischen Schlüsseln gespeicherten Gewichte der Entrys. Die Gewichte sind kein Attribut der Entrys, da sie innerhalb deren eigener Performance-Berechnung irrelevant sind und erst bei der Berechnung der Gesamtperformance des Portfolios ins Spiel kommen.

Zuletzt existiert eine Klasse Portfoliolist, deren einziges Attribut ein read-only Dictionary portfolios ist, in welchem sich mit dem Namen des Portfolios als Key Null oder mehr Portfolios befinden. Diese Klasse ist aktuell nicht zwingend notwendig, das Dictionary würde auch alleinstehend bereits genügen, jedoch stellt sie die Grundlage für zukünftige Erweiterungen des Projektes dar, welches letztlich online zugänglich gemacht werden und User-Profile mit eigenen Portfolios anbieten soll. Das Interface wird der Einfachheit halber mit Tkinter programmiert und dient bis zur Veröffentlichung als Lösung für lokale Tests und als Prototyp. Zwischenzeitlich wurde bereits eine aufwendigere online Streamlit-Implementierung versucht, aufgrund der Komplexität aber vorerst verworfen. Durch die erlebten Schwierigkeiten und aufgrund des Formats der Analyse-Spreadsheet wird zukünftig eher ein eigenes HTML-Frontend genutzt werden.

Die Analysen werden für den User nämlich als HTML-Datei bereitgestellt. Dies hat den Hintergrund, dass diese einfach automatisiert mit Hilfe der webbrowser-Library in Python geöffnet werden können. Gleichzeitig bietet die quantstats-Library mehrere Funktionen zur Berechnung wichtiger Metriken, welche vor Allem in der technischen Aktienanalyse nützlich sind. Darüber hinaus lassen sich Performance-Daten damit visualisieren und ausführliche Spreadsheets im HTML-Format erstellen.

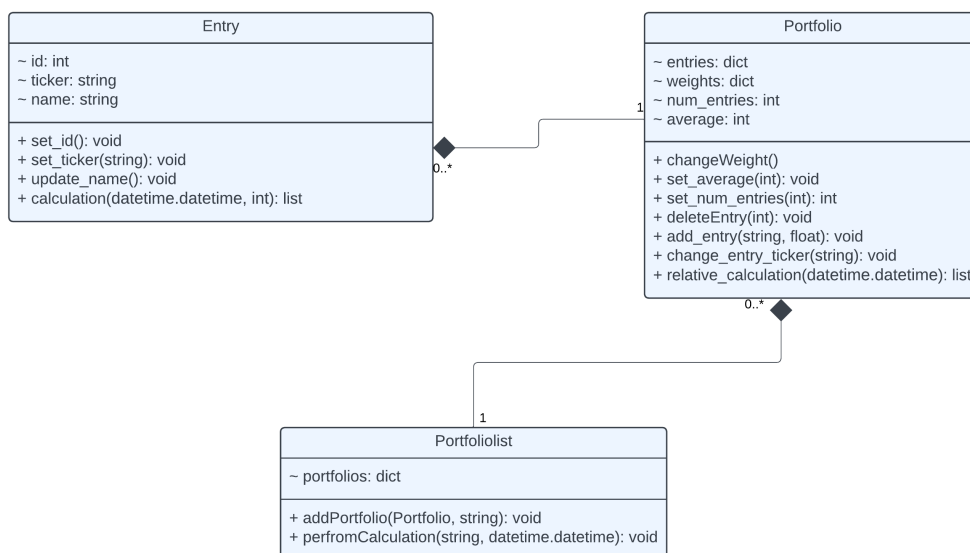
Insgesamt wird dem User also ein Tkinter-Interface bereitgestellt, welches ihn erst in einem Fenster zur Erstellung eines Portfolios aufruft und dabei die Korrektheit der Eingaben (Gewichte nichtnegativ und in Summe 100%, Portfolio-Namen einmalig zur Unterscheidung, Ticker valide, SMA mindestens 2 und ganzzahlig) überprüft. Wurde ein nichtleeres Portfolio erstellt, kann der User entweder weitere Portfolios erstellen, oder eine erste Analyse in einem zweiten Fenster genießen lassen. Die Dateinamen der Analysen basieren auch den Namen der Strategien. Bei Anklicken eines Berechnungs-Buttons wird die gewünschte Analyse erstellt, lokal im selben Ordner wie die frontend.py-Datei gespeichert und automatisch im Browser geöffnet.

Der Ablauf der Performance-Berechnung eines Portfolios in einer Portfoliolist kann also mit folgendem Sequenzdiagramm vereinfacht dargestellt werden:



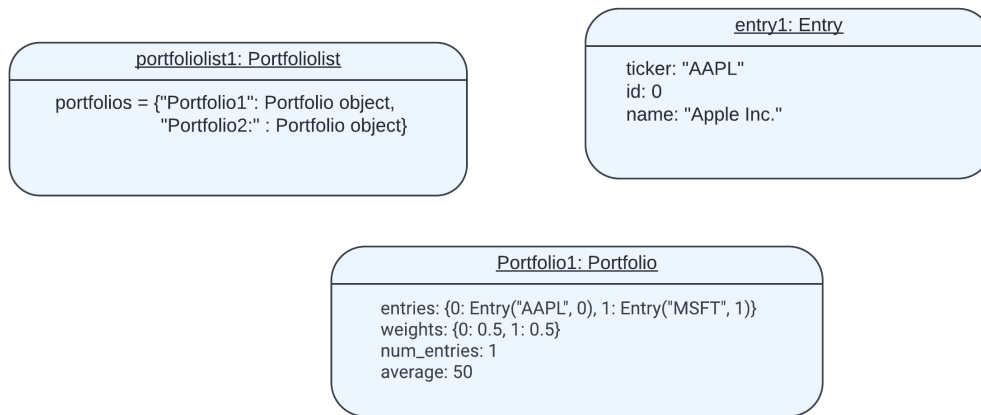
Das dazugehörige Klassendiagramm ist das Folgende:

Klassendiagramm

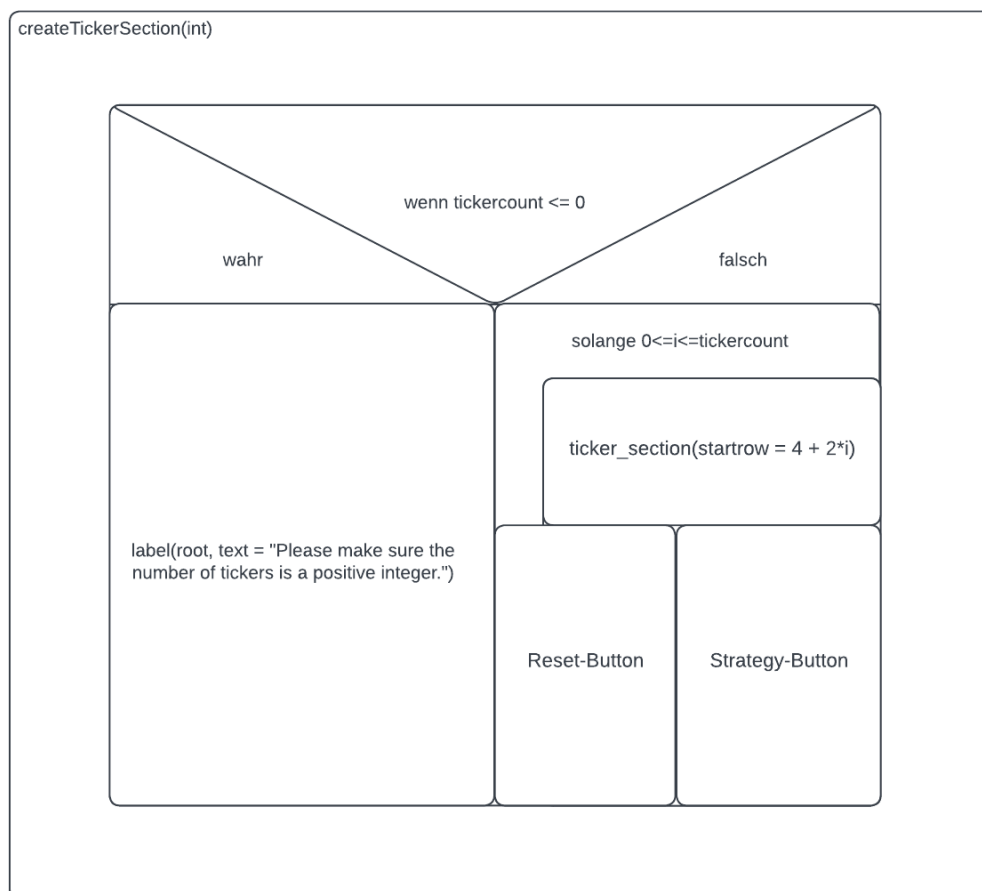


Und letztlich lassen sich die Klassen mit diesem Objektdiagramm veranschaulichen:

Objektdiagramm



Für die Generierung des Abschnittes im Interface, wo der Benutzer zuerst die Anzahl der Ticker für sein Portfolio angeben soll, sei hier beispielsweise ein Struktogramm angegeben:



Dabei wird erst sichergestellt, dass das Portfolio mindestens einen Ticker enthalten soll und anschließend für die Anzahl der gewünschten Ticker eine Sektion mit entsprechend vielen Eingabefeldern, ein Button zum Zurücksetzen der Seite sowie ein

Button zur Generierung des Portfolios mit den Eingaben des Users in den Eingabefeldern erzeugt.

3 Implementierung

Hier wird nicht auf alle Funktionen im Detail eingegangen. Wichtig sind vor Allem die Funktionen zur Performance-Berechnung.

Alle Zeilenangaben beziehen sich nun auf die `classes.py`-Datei.

In Zeile 51 wird mit der `calculation()`-Methode die Berechnung der Performance eines einzelnen Wertpapiers durchgeführt. Zum besseren Verständnis wird bereits der Quellcode ausführlich kommentiert, dennoch soll dieser hier in Teilen erklärt werden.

In Zeile 59 werden die historische Preisdaten zum Wertpapier von yahooFinance abgerufen und das erhaltene Pandas Dataframe auf dessen Close-Spalte als Pandas Series reduziert. Anschließend werden die Daten für den späteren Vergleich mit `dt.date`-Objekten in Zeitzone-naive `dt.date`-Objekten (von zuvor Zeitzone-bewussten `numpy.datetime64`-Objekten) konvertiert, wobei ein Umweg über die `Pandas.Timestamp`-Klasse gegangen werden muss. Anschließend wird die Index-Spalte der Pandas Serie durch die neuen Daten ersetzt. Die Pandas Serie wird jetzt wieder zu einem Dataframe umgewandelt, um eine weitere Spalte für den SMA hinzufügen zu können.

Wichtig sind jedoch nur die Einträge ab dem Datum, welches als Startdatum der Berechnung festgelegt wurde, sodass dieses in den Zeilen 73 bis 76 ermittelt und die Serie darauf zugeschnitten wird. Im Anschluss werden Ausgangsinvestment für GTAA- und B&H-Strategie auf 1 normiert und der letzte bekannte Abschlusskurs als `previous` festgehalten. Die aktuelle Periode beginnt mit dem aktuellen Datum und die Kaufentscheidung wird zum ersten Datum mit bekannten Kursen durchgeführt.

Nun werden immer bei Kaufentscheidung `True`, also Kaufen, die täglichen Veränderungen der `gtaa`-Strategie mit jenen des Wertpapiers gleichgesetzt, oder an Nicht-Handelstagen als `stagnant` angenommen. Dies wird fortgesetzt, bis das Datum der nächsten Kaufentscheidung (beziehungsweise des ersten bekannten Kurses danach) erreicht wird. Jeder neue bekannte Kurs wird ab Erreichen von dessen Datum als letzter bekannter Kurs `previous` für die Performance-Berechnung abgespeichert. Nach jedem Hinzufügen eines Datums erfolgt eine erneute Erhöhung des aktuellen Datums. Für die B&H-Strategie werden generell alle Kursveränderungen des Wertpapiers beachtet, bei der GTAA-Strategie nur jede in Kauf-Perioden.

Zum Schluss werden bis zum festgelegten Enddatum fehlende Performance-Daten noch mit einer stagnanten Entwicklung aufgefüllt und die Listen zum Tracking von Datum und Performance der Strategien je in einer Pandas Series mit dem Datum als

Index-Spalte zusammengefasst.

In der Portfolio-Klasse wird in den Zeilen 192 bis 211 für jedes Investment (Klasse Entry) in dem Portfolio die Berechnung durchgeführt, ein gemeinsames Start- und Enddatum auf Basis der Überschneidung verfügbarer Daten festgelegt und die Pandas Serien der Entrys mit ihrer jeweiligen Gewichten zusammengefügt, um gemeinsam für die beiden Strategien zurückgegeben zu werden.

Die Portfoliolist-Klasse greift lediglich auf ein Portfolio in ihrem Dictionary mit einem bestimmten Namen als Key zu und gibt dessen Performance zurück.

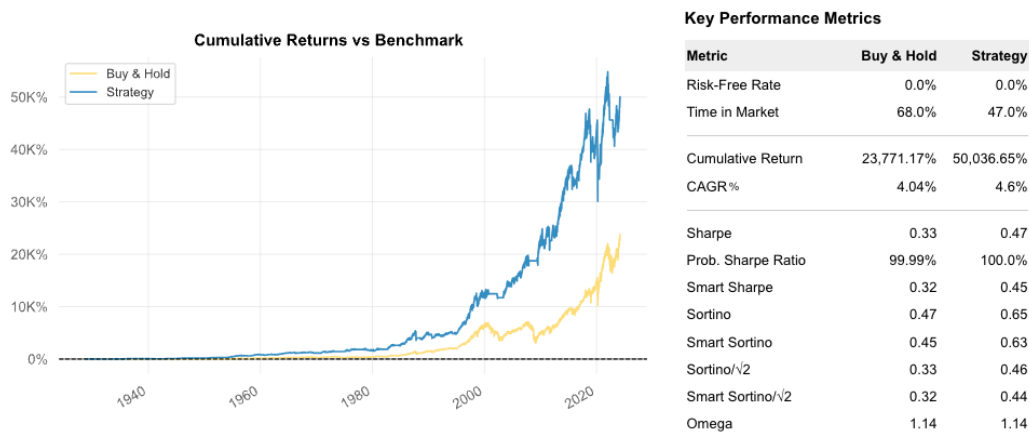
Viele der anderen Funktionen und teilweise auch Attribute der Portfolio- und Entry-Klassen werden nicht benötigt, sondern stammen noch von einer früheren, aufwändigeren Version des Projektes mit erweiterter Funktionalität, wurden aufgrund damit verbundener Schwierigkeiten jedoch verworfen. Vorhanden sind diese Funktionen weiterhin, um sie in Zukunft in einem neuen Anlauf der Implementierung neuer Features noch zu verwenden.

Das Interface wird hier nicht weiter erläutert, da es vor Allem nur aus Objekten von Klassen der Tkinter-Library besteht, welche nur noch als Bausteine zu einem Interface zusammengefügt werden.

4 Testläufe

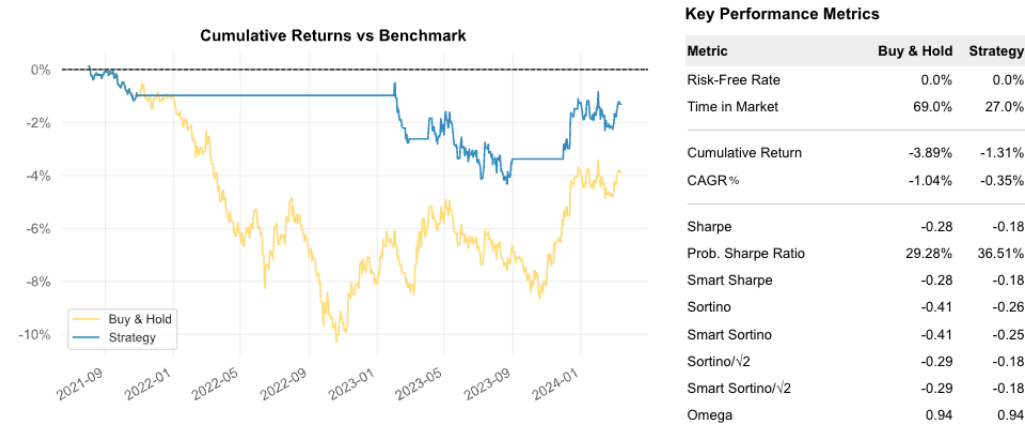
MyPortfolio vs. Buy & Hold 1 Nov, 1928 - 13 Mar, 2024

Benchmark is BUY & HOLD | Generated by [QuantStats](#) (v. 0.0.62)



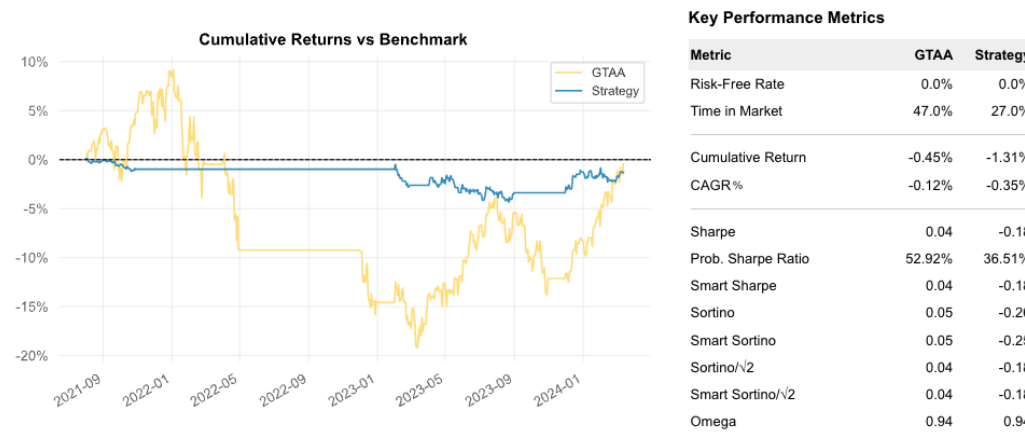
Oben zu sehen ist ein Teil des generierten Spreadsheets, bei welchem für ein Portfolio mit 100% Investment in den S&P500 die GTAA- und B&H-Strategie verglichen werden. Offenbar sind sowohl das Programm als auch die Strategie erfolgreich, wie vor Allem die 50,036.65% Gewinn per GTAA im Vergleich zu nur 23,771.17% mit B&H seit November 1928 bestätigen.

Bitcoin Test vs. Buy & Hold 2 Aug, 2021 - 13 Mar, 2024
Benchmark is BUY & HOLD | Generated by QuantStats (v. 0.0.62)

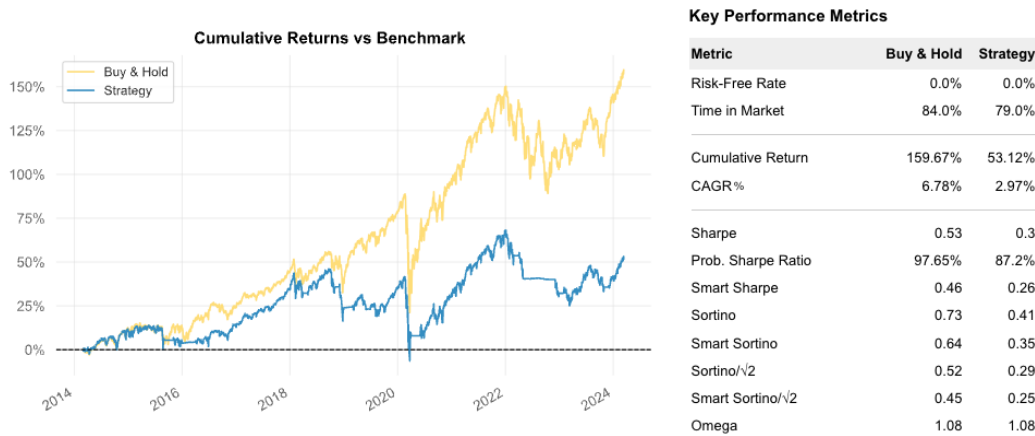


Ähnlich sieht es mit Bitcoin aus, wo zwar keine Strategie gewinnbringend ist, GTAA den Verlust durch Vermeidung der Mitnahme längerfristiger Kursverluste aber dennoch begrenzt.

Bitcoin Test vs. MyPortfolio 2 Aug, 2021 - 13 Mar, 2024
Benchmark is GTAA | Generated by QuantStats (v. 0.0.62)



Der erste Extremfall ist der Vergleich zwischen einem Investment mit Daten seit 1928 (S&P500) und einem mit Daten seit Ende 2020 (Bitcoin). Doch auch hier generiert das Programm ein Spreadsheet mit korrektem Zeitraum (die ersten 200 Tage ohne SMA sind logischerweise ausgeschnitten, daher der Beginn des Graphen in September 2021). Versagen würde das Programm dennoch bei Wertpapieren, deren Dataframe nach Abzug der ersten Zeilen für den SMA leer ist. Dabei würde eine Fehlermeldung der quantstats-Library hervorgerufen werden.



Der letzte Test ist die Analyse einer richtigen GTAA-Strategie. Hierzu werden:

- (1) der S&P500-Index zur Abbildung des US-Amerikanischen Aktienmarktes,
 - (2) der iShares 7-10 Year Treasury Bond ETF zur Abbildung des US-Amerikanischen Anleihemarktes,
 - (3) der iShares MSCI EAFE ETF zur Abbildung der Aktienmärkte nicht Nordamerikanischer Industrieländer,
 - (4) der EUWAX Gold-Index zur Abbildung der Rohstoffmärkte und
 - (5) der Realty Income-REIT zur Abbildung internationaler, vor Allem US-Amerikanischer Immobilienmärkte,
- verwendet und je mit 20% gewichtet.

Überraschender Weise ist der Ausgang hier für GTAA negativ, was an dem kurzen Zeitraum ohne länger anhaltende Krisen liegen könnte, wie es sie in dem Abschnitt nur mit der Corona-Pandemie gab. Vor Allem in Krisen holt die Strategie für gewöhnlich die Differenz auf, welche zuvor in Wachstumsphasen durch kurzfristige Schwankungen unterhalb des SMA trotz fortgesetztem Kurswachstum aufgebaut wird.

Wünschenswert wären zukünftig die Möglichkeit zur Auswahl eigener Betrachtungszeiträume, das langfristige Abspeichern von Portfolios - und nicht nur von deren Analysen - sowie der Vergleich von mehr als nur zwei Strategien gleichzeitig. Interaktive Diagramme wären ebenfalls hilfreich.

Dennoch wurden alle gestellten Anforderungen erfüllt und das Tool kann bereits jetzt uneingeschränkt nach Belieben für alle auf yahooFinance gelisteten Ticker verwendet werden.