

Algorithmisches Handeln von Crypto-Währungen

Der algorithmische Handel beschreibt den automatisierten Handel mittels eines Computerprogramms, welches einem definierten Satz von Anweisungen folgt, um einen Handel zu platzieren. Der Handel kann theoretisch Gewinne in einer Geschwindigkeit und Häufigkeit erzielen, die für einen menschlichen Händler unmöglich sind. Die definierten Anweisungssätze basieren auf Zeitpunkt, Preis, Menge oder irgendeinem mathematischen Modell. Der algorithmische Handel hält Einzug in fast alle Bereiche des Handels und der gesamten Investmentbranche und ist somit keine Neuheit. Im Internet werden bereits zahlreiche überbezahlte Systeme für den automatisierten Handel mit Aktien oder Crypto-Währungen wie Bitcoin, Ethereum oder ähnlichem angeboten. An letzteres will auch ich mit einem algorithmischen System anknüpfen, um herauszufinden, ob der automatisierte Handel mit Crypto-Währungen, durch ein von mir selbst entwickeltes System, erfolgreich sein kann. Dabei ergeben sich einige, zu lösende Herausforderungen:

1. Trading Strategien

Ohne eine geeignete Trading Strategie, die entscheidet, wann Einkäufe und Verkäufe zu tätigen sind, kann wohl kein System Erfolg haben. Somit bilden die Einbindung und die Verwendung geeigneter Strategien einer der wichtigsten Faktoren des Systems.

Ein Beispiel einer sehr einfachen Trading Strategie stellt die Moving Average Strategie dar. Der **Moving Average (MA)** ist ein Trendindikator und eine Trading Strategie, dargestellt durch eine kurvige Linie und wird auf Basis der Preisdaten berechnet. So werden beispielsweise stündliche Preisinformationen eines bestimmten Zeitraums in eine Durchschnittsberechnung eingebracht und ergeben somit eine Linie, welche der Preisentwicklung des Zeitraums folgt. Anschließend wird die Linie zur Bestätigung von Trends in der zukünftigen Preisentwicklung zu Hand genommen. Auf dem folgenden Bild lässt sich der Moving Average als rote Linie erkennen.



Implementieren ließe sich die Strategie beispielsweise wie folgt:

- **Kaufe**, sobald sich der Preis einen bestimmten Faktor x **unter** der Moving Average befindet
- **Verkaufe**, sobald sich der Preis einen bestimmten Faktor y **über** der Moving Average befindet

Der Fokus dieses Projekts soll jedoch weniger der Findung und Konzeption neuer Trading Strategien folgen, da diese bereits in einer Vielzahl im Internet zu finden sind und theoretisch nur in Code umgesetzt werden müssen. Vielmehr soll es um die Gestaltung und Entwicklung eines Systems gehen, in welches diese Trading Strategien möglichst einfach und modular „einzuhängen“ sind. Das heißt, das System soll hierbei relativ generisch gehalten werden und es soll egal sein, welche Trading Strategie nun für den Handel ausgewählt wurde. So soll es möglich sein, neu implementierte Strategien ohne viel Aufwand in das System einbinden zu können.

2. Auswahl einer Handelsplattform

Der Handel von Crypto-Währungen mittels eines algorithmischen Systems erfordert, genauso wie beim Handeln von Hand, eine Plattform, auf der man ein sogenanntes Crypto-Wallet (Depot) anlegt. Sozusagen der virtuelle Geldbeutel auf dem sich die gesammelte Crypto-Währung befindet. Ist dieses einmal angelegt, ist man nun in der Lage verschiedene Tätigkeiten wie z.B. Einkäufe und Verkäufe zu tätigen. Im Vergleich zum Handeln von Hand, benötigt das Trading System jedoch eine Schnittstelle (API), an die es anknüpfen kann, um ebenfalls in der Lage zu sein, Einkäufe und Verkäufe zu tätigen. Solche APIs werden zum Teil kostenlos von einigen Plattformen angeboten. Da die Anzahl der möglichen Aktionen von den Features abhängt, die per API bereitgestellt werden, ist die Auswahl einer geeigneten Trading Plattform der erste Schritt. Jedoch kann es durchaus vorkommen, dass sich nach einiger Zeit eine andere, bessere Plattform findet. Daher soll auch hier auf Modularität gesetzt werden. Dies wird durch die Implementierung einer geeigneten Schnittstelle an die Funktionen der API erreicht und beeinträchtigt beim Austausch dieser nicht die anderen Komponenten des Systems.

3. Backtest

Backtesting bzw. Rückvergleich bezeichnet den Prozess, eine Strategie zu evaluieren, indem die Strategie auf historische Daten angewendet wird. Findet man beispielsweise eine neue, vielversprechende Trading Strategie und möchte diese, nach der Umsetzung in Code, nun im Trading System testen, ist es sehr riskant, die Strategie auf Echtzeitdaten des Markest anzuwenden. Sollte die Trading Strategie nämlich doch nicht so vielversprechend sein, läuft man Gefahr, große Verluste durch schlecht platzierte Ein- und Verkäufe hinnehmen zu müssen. Der Backtest macht diese Gefahr zunichte und stellt somit eine der wichtigsten Komponenten des Systems dar. Mit ihm soll es möglich sein, ausgewählte Strategien auf historische Marktdaten verschiedener Crypto-Währungen anzuwenden und somit Daten über

das Verhalten dieser zu sammeln. Die Simulation soll dabei mit bestimmten Parametern, wie z.B. einem fiktiven Startkapital beginnen und letztendlich Auskunft darüber geben, wie viel Gewinn oder Verlust die Strategie in vergangenen Marktszenarien hätte generieren können.

4. Konsolen Anwendung

Alle bereits aufgeführten Komponenten sollen Teil einer Konsolenanwendung werden, welche diese dann als System vereint. Hierbei soll es möglich sein, unterschiedliche „Trading Bots“ zu erstellen, welche verschiedenen Parametern wie Auswahl der Crypto-Währung, Strategie, Kapital, Kaufbetrag usw. zu Grunde liegen. Diese sollen dann in der Lage sein, echten Handel mit Crypto-Währungen zu betreiben. Die Anwendung soll außerdem einen Backtest starten können und die erzeugten Ergebnisse dem Nutzer in lesbarer Form bereitstellen.

5. Datenbank

Die vom Backtest erzeugten Ergebnisse, sowie die unterschiedlichen Konfigurationen der Trading Bots sollen unter Verwendung einer geeigneten Datenbank gespeichert und wieder abrufbar gemacht werden. Auch hier soll auf Modularität geachtet werden, da es bei einer Weiterentwicklung durchaus von Vorteil sein kann, die Datenbank austauschen zu können, ohne dabei das gesamte System zu gefährden.