

Kunnen we snel rijk worden met AI?

In het kader van mijn zoektocht naar antwoord op de vraag **AI magie of gewoon wat regels Python code?** Dacht ik waarom wagen we niet een poging om de heilige graal te ontdekken/ontwikkelen in de aandelenhandel. Met al die beschikbare AI kennis die voor het oprapen ligt moet dat toch te doen zijn?!

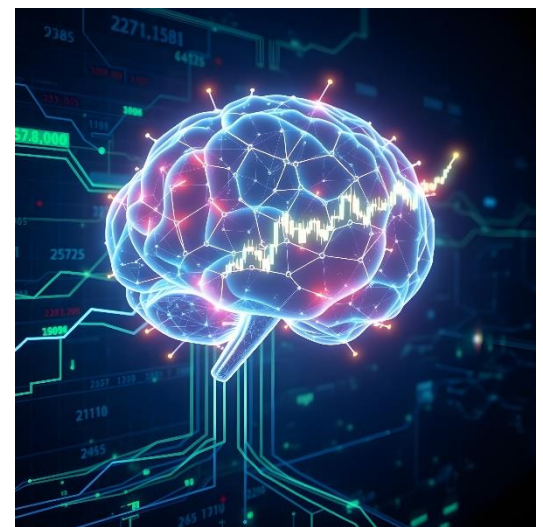
A Holy Grail trading strategy refers to a trading approach that is believed to be infallible and consistently profitable. It's a strategy that traders might perceive as a "magic bullet" for making riches in the stock market. The hope is that it will offer big returns with no risk.

Het idee

Met behulp van Python, AI-modellen, gratis beschikbare data en wat coderen de aandelenkoers van morgen voorspellen voor een specifiek aandeel.

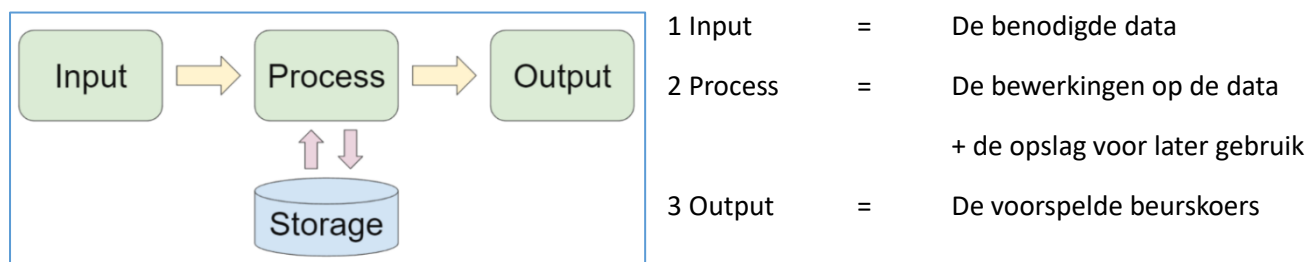
De Ingrediënten

- Een Python programma
- Gratis beschikbaarheid van aandelen data
- Gratis beschikbaarheid van aanvullende economische data
- Open source AI-modellen
- Aanvullende AI-hulp (Co-pilot, Deepseek, ChatGPT, Claude 3.5 Sonnet)
- Wat kennis van bovenstaande
- ...



De grote lijn

Zoals met alle op te lossen IT-problemen delen we het idee op in 3 delen:



De data

Om het leuk te houden willen we alle gebruikte data gratis beschikbaar hebben. Mocht, in een later stadium, het model het gewenste effect hebben en de euro's binnen stromen dan is betalen voor de data natuurlijk geen probleem!

De historische aandelen data halen we op vanaf [financialmodeling](#). Deze levert 5 jaar historische koersdata. Daarnaast willen we zoveel als mogelijk aanvullende economische data toevoegen om het model met zoveel mogelijk relevante gegevens te 'voeden'. Dit kan met de [Python Technical Analysis Library](#). Een aantal voorbeelden die we inzetten in ons model zijn:

- **On-Balance Volume (OBV)** → Een indicator die het volume relateert aan prijsveranderingen.
- **Williams %R** → Een momentumindicator die overkochte en oversold condities meet.
- **Rate of Change (ROC)** → Meet de procentuele verandering in prijs over een bepaalde periode.

- **Relative Strength Index (RSI)** → Een momentumindicator die de snelheid en verandering van prijsbewegingen meet.
- **Moving Average Convergence Divergence (MACD)** → Een trendvolgende momentumindicator.
- **Bollinger Bands** → Een volatiliteitsindicator die een boven- en onderband rond een voortschrijdend gemiddelde plaatst.
- **Average Directional Index (ADX)** → Meet de sterkte van een trend.
- **Chaikin Money Flow (CMF)** → Combineert prijs en volume om de geldstroom te meten.
- **Commodity Channel Index (CCI)** → Meet de afwijking van de prijs ten opzichte van het statistisch gemiddelde.
- **Stochastic Oscillator** → Een momentumindicator die de sluitingsprijs vergelijkt met het prijsbereik over een bepaalde periode.
- **Average True Range (ATR)** → Meet de volatiliteit door het bereik van prijsbewegingen te analyseren.
- **Aroon Indicator** → Identificeert trendveranderingen en de sterkte van een trend.
- **Keltner Channel High** → Een volatiliteitsindicator die een bovenband rond een voortschrijdend gemiddelde plaatst, gebaseerd op de gemiddelde true range (ATR).
- **Keltner Channel Low** → Een volatiliteitsindicator die een onderband rond een voortschrijdend gemiddelde plaatst, gebaseerd op de gemiddelde true range (ATR).
- **Average True Range (ATR)** → Een volatiliteitsindicator die het gemiddelde van de true range over een bepaalde periode meet.

Het proces

1. Als we de data binnen hebben kan het echte werk beginnen! Voor het gemak en latere visuele validatie slaan we de data op in een CSV. Als blijkt dat het model goed bruikbaar is dan volgt in een volgende versie het vastleggen van de data in een database zodat we zelf nog langere historie kunnen gaan vormen.
2. Voor het gebruik binnen de applicatie plaatsen we de data in een Pandas Dataframe vergelijkbaar met een tabel met rijen en kolommen.
3. Dan splitsen we de data in een deel voor training en een deel, van 100 dagen, voor testen.
4. We schalen de data zodat het bruikbaar is voor de computer. Oftewel we normaliseren de waarden binnen een bepaald bereik (standaard tussen 0 en 1).

Waarom schalen inzetten?

- Verbetering van machine learning prestaties
 - Vermijden van dominantie door grote getallen
 - Nuttig voor neurale
5. Nu gaan we het AI-model definiëren om onze voorspelling vorm te geven. In ons geval is dat een **neuronaal netwerk** met behulp van **Keras** en **TensorFlow**, specifiek een **Long Short Term Memory (LSTM)-gebaseerd model** voor tijdreeksvoorspellingen, zoals het voorspellen van aandelenkoersen.

Wat doet het?

- Het neemt tijdreeksen als input (bijvoorbeeld historische aandelenprijzen).
 - Het gebruikt een LSTM-laag om patronen en trends in de tijdreeks te leren.
 - Het geeft een voorspelling als output, bijvoorbeeld de aandelenprijs van morgen.
6. Het bovenstaande model zetten we in om het te trainen met de train-dataset. Doel is de patronen in de gegevens te onderkennen en ervan te leren.
 - Het doet dit 50 keer (aantal epochs).
 - In elke epoch verwerkt het model een bepaald aantal batches (steps_per_epoch).
 - Na elke epoch wordt de validatiedataset (test_dataset) gebruikt om te zien hoe goed het model presteert.
 7. Nu gaan we met het test-deel van de data bekijken hoe goed ons model is in het voorspellen van de koers.
 8. De uitkomst van deze voorspelling test wordt weergegeven in 2 waarden als mate van afwijking: **Mean Squared Error (MSE)** is een maat voor de gemiddelde kwadratische afwijking tussen de voorspellingen van je model en de werkelijke waarden; **0.4970483243207762**

Mean Absolute Error (MAE) berekent de gemiddelde absolute afwijking tussen de voorspellingen en de werkelijke waarden, zonder dat grotere fouten zwaarder worden gewogen zoals bij de MSE;

0.5616767144227245

9. In ons geval is de koers van **AHOLD** genomen als training en testdata. We weten dat de koers momenteel zo rond de \$35,00 schommelt. Als we deze koers gebruiken om onze afwijking te berekenen dan zien we het volgende:

De **MSE** van 0.497 betekent dat de gemiddelde kwadratische afwijking tussen de voorspelde waarden en de werkelijke waarden \$0.50 is. Dit is een kleine afwijking ten opzichte van de aandelenprijs van \$35,00.

Vergelijking: Als de aandelenprijs gemiddeld rond de \$35,00 ligt, dan is een **MSE** van 0.497 relatief laag, wat betekent dat het model goed presteert. Bijvoorbeeld: een fout van \$0.50 betekent dat het model gemiddeld een fout heeft van ongeveer 1.4% van de voorspelde prijs.

De **MAE** van 0.56 betekent dat de gemiddelde absolute afwijking tussen de voorspelde en werkelijke waarden \$0.56 EUR is. Vergelijking: Een **MAE** van \$0.56 is ook relatief klein als we kijken naar een aandelenkoers van \$35,00, wat betekent dat de voorspellingen meestal binnen ongeveer 1.6% van de werkelijke waarde liggen.

10. We slaan het model op zodat we het kunnen gebruiken bij onze dagelijkse voorspelling

De Output

De output van het idee is dus de heilige graal voor ons als 'newbee day-trader'. **DE** manier om snel rijk te worden met een minimale inspanning. Oftewel vandaag de beurskoers van morgen voorspellen met ons AI-model en kopen die stocks.

(donderdag 13 feb) Het model geeft aan dat de slotprijs voor morgen **\$32,94** is.

Best een flinke afwijking van de koers van vandaag **\$35,02**.

Het goede nieuws is dat als het klopt we snel veel geld (hadden) kunnen verdienen 😊

Als extraatje kunnen we de voorspelde prijs ook in een grafiek (Matplot) bekijken tov de prijs over laatste 60 dagen

```
1/1 ————— 0s 312ms/step
=====
Predicted closing price for tomorrow: 32.938515083789824
=====
Last 60 actual closing prices:
[33.85 34.45 34.63 34.49 33.86 34.31 34.42 34.12 34.67 34.56 34.36 34.48
 34.49 35.11 34.21 33.84 33.94 33.57 33.67 33.9 33.64 33.44 32.65 32.57
 32.62 32.48 32.5 32.54 32.75 32.46 32.59 32.56 33.04 32.84 33.44 33.35
 33.19 33.03 33.44 34.01 34.41 34.88 35.79 35.19 35.37 35.02 35.57 35.64
 35.64 35.7 35.31 35.16 36.2 36.32 36.36 36.25 36.38 36.95 35.02 35.02]
```

