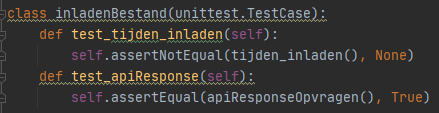
Brian, Damian, Edwin en Pelle

Testrapport

IPBDAM5

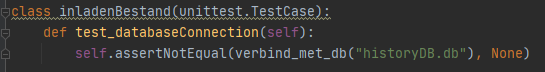
We hebben unit testen uitgevoerd voor de code die we gemaakt hebben. De belangrijkste processen zijn verdeeld in 3 python bestanden, namelijk het data ophalen, de geschiedenis trades uitvoeren en het bestand waar die van alle trades berekend hoeveel winst we per verschillend patroon hebben verdiend of hoeveel verlies we hebben per patroon.

Om te beginnen kijken we naar het data ophalen, in dit python bestand worden alle candlesticks van de afgelopen 3 jaar opgehaald. Hierin gebeuren een aantal stappen en 2 daarvan hebben we getest zoals hieronder te zien:

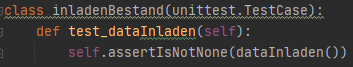


Hier testen we de funtie tijden\_inladen(). In deze functie laadt die het tijden bestand op van alle timestamps die we willen ophalen en die splitst die op in een list en geeft die dan weer terug als variabel. Als het inladen van het bestand niet lukt dan geeft die niks terug of een error, in de meeste gevallen geeft die een None bestand terug. Waar wij op testen is dat die geen None waarde teruggeeft. Als die dus iets anders dan een None waarde teruggeeft is de test gelukt, in dit geval zou die alleen een list kunnen teruggeven. Als die None teruggeeft is de test gefaald. Dit hebben we getest door het bronbestand weg te halen waardoor die het bestand dus ook niet kan laden. In dit geval zal die dus een None type variabel maken en zal de test falen.

Het is ook belangrijk dat we de verbinding met de API van Binance testen voordat we de vervolgstappen uitvoeren. Met deze unittest kijkt die of we een juiste API response krijgen in de vorm van een HTTP code. Deze code wordt omgezet naar een True of False, het is True als de response goed is en False als de response slecht is, in de unittest kijkt die dus naar de response en of de response True is. Als die True is dan is de unittest gelukt, anders is die gefaald.



In de historyTrades is het belangrijkste dat we de connectie met de database testen. In deze unittest roepen we de verbinden met database functie aan met de sleutel naar de database. Deze functie geeft de verbinding terug die we kunnen gebruiken om SQL query’s uit te voeren. Het moet dus niet een None variabel zijn. Hierop testen we en als die dus geen None variabel teruggeeft is de unittest gelukt, anders is die gefaald.



In het bestand winstPerPatroon wordt bij alle trades gekeken hoeveel winst we per patroon hebben gemaakt in totaal. Hierbij hebben we een functie waar alle data ingeladen wordt en gecheckt wordt en dit wordt teruggegeven voor de volgende functie die de data weer gebruikt om berekeningen mee te doen. Het is dus de bedoeling dat de variabel niet leeg is omdat we hier niks mee kunnen. In deze unittest wordt dus gekeken of er wel een variabel wordt teruggeven waar echt iets in staat en dus niet van het type None is. Als het een None is zal de unittest falen, als het geen None type variabel is zal de test slagen en heeft hij de data dus goed kunnen inladen.