

Verslag – Voorspellen met monte carlo - FA

IN DIT VERLSAG WORD ER UITGELEGT HOE DE FA VAN
VOORSPELLEN MET MONTE CARLO GEREALISEERT IT
CEYHUN ÇAKIR 1784480

Table of Contents

<i>Inleiding</i>	2
<i>Probleemstelling</i>	2
<i>Voetbal tournament Voorspellen met Monte Carlo</i>	2
Hoe gebruik je de simulatie?	2
Wat voor informatie kan ik eruit halen?	3
<i>Mersenne Twister</i>	3
Hoe werkt het?	3
Wat kan je der mee?	3
Wat zijn de voordelen van de Mersenne Twister	3
Wat zijn de nadelen van de Mersenne Twister	3
Hoe heb ik het gebruikt	3
<i>Uitkomsten</i>	4

Inleiding

in dit verslag vertel ik hoe ik met de Mersenne Twister algoritme een Monte Carlo voorspelling heb gemaakt over een voetbal tournament. Verder ga ik uitleggen hoe ik de Mersenne Twister algoritme heb geïmplementeerd en hoe je het kan gebruiken. Ook leg ik uit hoe ik de simulatie heb gebouwd en hoe je uiteindelijk het kan gebruiken om voorspellingen te krijgen van jouw team. Uiteindelijk toon ik de resultaten die uit de voetbal simulatie is gekomen.

Probleemstelling

Voor de opdracht moest ik een Monte Carlo Simulatie realiseren die meerdere keren (duizenden) de competitie speelt met gebruik van een tabel met kansen. Hoe vaker je een competitie speelt, hoe nauwkeuriger je voorspelling wordt moet worden. Het moest bijhouden bij hoeveel punten elk team scoort in de competitie en uiteindelijk een ranking maken van alle teams in de tournament.

Voetbal tournament Voorspellen met Monte Carlo

Voor het voorspellen van het voetbal tournament ben ik begonnen met het bekijken wat ik nodig had kwa classes. Door veel nagedacht te hebben kwam ik op een idee uit om een team klassen te hebben die een team kan aanmaken voor het tournament. Ook kwam ik op het idee om uiteraard ook een tournament klassen nodig had die een tournament kon aanmaken en teams konden accepteren voor het tournament.

Na dat was ik begonnen met het bouwen van het team klassen en functie ingevoegd om bijvoorbeeld kansen op te slaan het ook het team naam op te slaan zodat dit later in het tournament gebruikt kan worden.

Voor het tournament klassen had ik voornamelijk de functionaliteiten nodig waar ze tegen elkaar konden spelen en dat ik ook de uitslagen kon bijhouden. Dit was voornamelijk nodig aangezien we voor elke tournament uiteraard de uitslagen wouden opslaan. Door dit (10000) keer te executeren konden we een accurate uitslag krijgen welke team met hoeveel kans op een bepaald positie zou komen.

Om de resultaten te kunnen tonen heb ik de python module tabulate gebruikt aangezien dit makkelijk te gebruiken is en makkelijk headers en tables kan invoegen.

Hoe gebruik je de simulatie?

Door gemakkelijk het project vanuit mijn GitHub repository te downloaden kan je gemakkelijk het programma executeren door de volgende commando te schrijven (python main.py). Door dit commando te executeren zou de simulatie in gang moeten komen.

Wat voor informatie kan ik eruit halen?

Uit het programma zou er een tabel moeten komen met alle kansen voor elke positie voor elk team. Deze kansen geven aan wat de kans is dat het bepaalde team op dit positie zou kunnen eindigen binnen het tournament.

Mersenne Twister

Voor mijn opdracht heb ik de mersenne twister algoritme gebruikt. Aangezien dit algoritme beter kan zijn dan een true random number generator had ik als gedachte om dit algoritme te gebruiken.

De mersenne twister is een psuedorandom number generator (PRGN). Het kan van een w -bit integers genereren van (0, tot 2^w-1). Dit algoritme is gebaseerd op een matrix linear recurrence over een finite binary field.

Hoe werkt het?

Door een seed waarde te geven kan je een reeks random getallen genereren. Dit zorgt ervoor dat je in en instantie komt te zitten. Als je elke keer een ander random nummer wilt krijgen moet je de state transformeren in een one way function.

Wat kan je der mee?

Wat zijn de voordelen van de Mersenne Twister

- Je mag het overal gebruiken in je projecten aangezien er geen copyright op zit.
- De implementatie van de Mersenne Twister creëert random nummers sneller dan true random nummer functies.

Wat zijn de nadelen van de Mersenne Twister

- De buffer kan redelijk groot zijn
- Meerdere instanties door andere seed waardes
- Heeft een minder goede diffusie
- Het is niet zo crypto grafisch beveiligt

Hoe heb ik het gebruikt

Door de pseudo code van Wikipedia goed te bestuderen. Kon ik het van c naar python vertalen door naar de constraints te bekijken. Door deze functies te implementeren zoals (seed, twist etc..) kon ik een random nummer genereren die ik kon gebruiken voor een match tussen twee teams.

Door de onderstaande afbeelding te bekijken kunnen we zien wat de resultaten zijn van alle teams binnen de competitie gebaseerd op hun statistieken.

We kunnen hier bijvoorbeeld zien dat Ajax met een percentage van (71.32%) eerst positie zou kunnen eindigen. De percentages die we hier zien zijn de indicaties van elk team waar ze kunnen eindigen.