### 

### 

### 

### 

### **Python fördjupning EC Utbildning**

### **Kunskapskontroll 2**

Isaac Högfeldt

2025-09-22

### **1. Inledning**

I detta projekt har jag valt att fördjupa mig inom web scraping, datalogging och interaktiv visualisering med Python – ett område som är centralt inom Data Science, särskilt när det gäller att samla in och analysera realtidsdata. Projektet fokuserar på att hämta och logga fulltids oddset från fotbollsmatcher från spelbolaget Unibet, spara dessa i en CSV-fil, och sedan visualisera oddsens utveckling under matchdagen fram till matchstart.

### **2. Syfte och mål**

Syftet med projektet är att:

* Självständigt bygga ett system som automatiskt hämtar och loggar oddsdata från ett externt API.
* Strukturera och spara datan på ett sätt som möjliggör vidare analys.
* Skapa ett användarvänligt gränssnitt för att visualisera oddsens förändring över tid.

Projektet är relevant för Data Science-yrkesrollen då det kombinerar flera viktiga kompetenser: datahämtning, datarengöring, visualisering och användarinteraktion.

### **3. Genomförande**

#### **Kodstruktur**

Projektet består av tre huvudfiler:

* main.py: Hämtar dagens fotbollsmatcher från fem europeiska ligor via Unibets API, extraherar "Full Time"-odds och loggar dessa till full\_time\_log.csv.
* plot\_data.py: Läser in CSV-filen och erbjuder ett GUI med radioknappar för matchval samt en interaktiv graf som visar oddsens utveckling.
* config.py: Innehåller konfigurationer för logging, HTTP headers och ligaval.

#### **Automatisering**

En batchfil (run\_scraper\_10\_min.bat) kör main.py var tionde minut, vilket möjliggör kontinuerlig insamling av oddsdata under dagen.

#### **Dataformat**

Oddsdata loggas med följande fält:

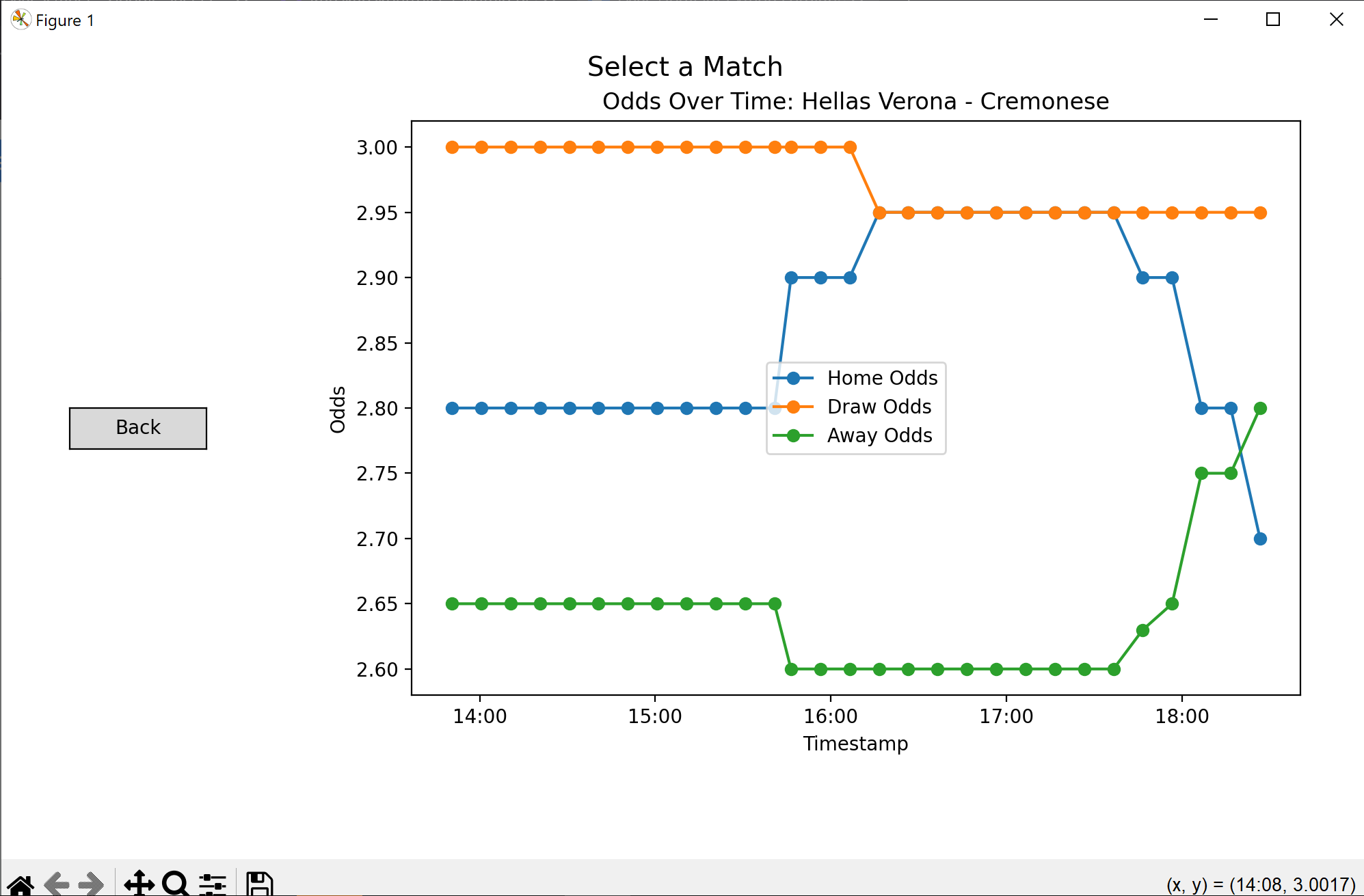
* event\_id, match\_name, event\_start, time\_stamp, home\_odds, even\_odds, away\_odds

#### **Visualisering**

Användaren väljer en match via radioknappar. Därefter visas en linjeplot med tre kurvor:

* Home Odds (orange)
* Draw Odds (blå)
* Away Odds (grön)

Exempel på graf:

* 

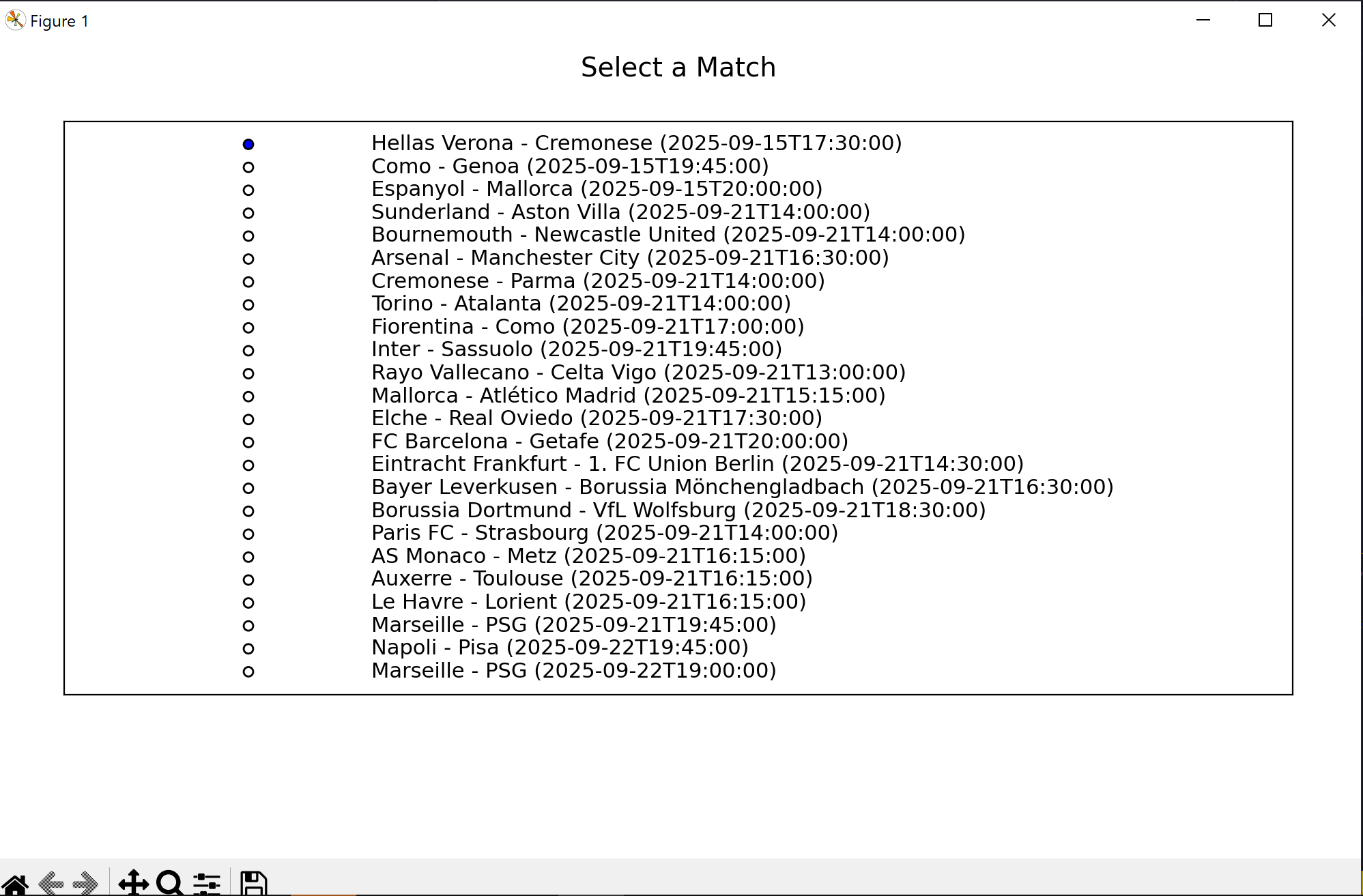
Hellas Verona - Cremonese Odds för hemmavinst låg stabilt runt 2.8, medan oavgjort sjönk och bortalaget steg kraftigt mot matchstart.

### 

### **4. Resultat**

Projektet har resulterat i ett fungerande system som:

* Hämtar och loggar oddsdata korrekt.
* Bevarar metadata och undviker duplicering.
* Visualiserar oddsens utveckling.
* Är modulärt och lätt att vidareutveckla.



### **5. Reflektion & vidare utveckling**

#### **Vad fungerade bra**

* API-hanteringen var stabil och responsiv.
* CSV-loggning med bevarad historik fungerade som tänkt.
* GUI:t med matplotlib.widgets gav en ok användarupplevelse men kan definitivt förbättras, framförallt när fler matcher läggs till.

#### **Utmaningar**

* API:et saknar dokumentation, vilket krävde manuell reverse engineering.
* Vissa matcher saknade oddsdata vid vissa tidpunkter (t.ex. Tillfälligt borttaget från Unibet) – hanterades med fallback-logik.
* Mycket tid las på reverse engineering, vilket ledde till att jag inte hade tid till att analysera vidare odds datan.

#### **Möjlig vidareutveckling**

* Implementera oddsförändringsdetektion och notifieringar.
* Koppla till en databas för mer robust lagring.
* Utöka GUI:t med filter för liga, tid eller oddsintervall.
* Analysera frekvensen på “större” oddsförändringar under matchdag. Detta kan ge en intressant inblick i hur snabbt oddsmarknaden kan reagera på nyheter osv.