# Sistema di Money Management basato su D'Alembert con EV e logica statistica

# INPUT MANUALI (puoi sostituirli con input dinamici o da file CSV)

frequenza\_storica = 0.39 # es. 39% di segno X in 1000 partite

ritardo\_attuale = 16

ritardo\_medio = 11

ritardo\_massimo = 18

quota = 3.20

unit\_base = 5 # puntata base in euro

storico\_esiti = [] # lista di boolean: True = vinta, False = persa

# 1. Calcolo probabilità stimata

def calcola\_probabilita(frequenza\_storica, ritardo\_attuale, ritardo\_massimo):

p = frequenza\_storica

if ritardo\_attuale >= ritardo\_massimo \* 0.90:

p \*= 1.05 # incremento del 5% se vicino al massimo

return min(p, 1.0)

# 2. Calcolo del valore atteso (EV)

def calcola\_ev(p, quota):

return p \* (quota - 1) - (1 - p)

# 3. Verifica se giocare o meno

def should\_play(ritardo\_attuale, ritardo\_medio, ritardo\_massimo, ev):

return (

ritardo\_attuale >= ritardo\_medio \* 1.10 or

ritardo\_attuale >= ritardo\_massimo \* 0.90

) and ev >= 0

# 4. Calcolo puntata con D'Alembert

def calcola\_puntata\_dalembert(esiti, unit\_base):

step = sum(1 if not esito else -1 for esito in esiti)

step = max(step, 0)

return unit\_base \* (1 + step)

# --- Simulazione del processo ---

p = calcola\_probabilita(frequenza\_storica, ritardo\_attuale, ritardo\_massimo)

ev = calcola\_ev(p, quota)

giocare = should\_play(ritardo\_attuale, ritardo\_medio, ritardo\_massimo, ev)

print("\n--- DECISIONE ---")

print(f"Probabilità stimata: {p:.2f}")

print(f"Valore Atteso (EV): {ev:.3f}")

print(f"Condizioni OK per giocare? {giocare}")

if giocare:

puntata = calcola\_puntata\_dalembert(storico\_esiti, unit\_base)

print(f"Puntata da fare: {puntata:.2f} euro")

else:

print("Non è consigliato scommettere in questo momento.")

# Per test: inserisci manualmente gli esiti delle scommesse giocate

# storico\_esiti.append(True) oppure storico\_esiti.append(False)