📘 Report Tecnico: Sistema di Money Management Avanzato

🎯 Obiettivo

Creare un sistema automatico di gestione delle scommesse (Money Management) che:

Usa il metodo D’Alembert come strategia di progressione.

Si basa su dati statistici reali (frequenza, ritardo medio/massimo/attuale).

Valuta ogni scommessa tramite il calcolo del valore atteso (EV).

Decide se scommettere o meno in base a regole logiche + EV.

---

🔧 DATI NECESSARI

Per ciascun segno (1, X, 2):

Dato Descrizione

Frequenza storica % di uscite in 1000 partite

Ritardo attuale Numero di partite senza uscita del segno

Ritardo medio Media dei ritardi storici

Ritardo massimo Massimo ritardo storico

Quota attuale Quota decimale offerta dal bookmaker

Capture rate (presa) % di successo reale del sistema

---

📐 LOGICA DECISIONALE

1. Calcolo della probabilità stimata

var p = frequenzaStorica // es. 0.39

if (ritardoAttuale >= ritardoMassimo \* 0.90) {

    p \*= 1.05 // aumento per “pressione” statistica

}

2. Calcolo del valore atteso (EV)

val EV = p \* (quota - 1) - (1 - p)

3. Regole per entrare nel ciclo D’Alembert

val sogliaMedia = ritardoMedio \* 1.10

val sogliaAlta = ritardoMassimo \* 0.90

val trigger = (ritardoAttuale >= sogliaMedia || ritardoAttuale >= sogliaAlta) && EV >= 0

---

📊 STRATEGIA DI PUNTATA: D’Alembert

fun calcolaPuntata(perditeConsecutive: Int, unitàBase: Double): Double {

    return unitàBase \* (1 + perditeConsecutive)

}

📌 Aumenta la puntata dopo ogni perdita di 1 unità, torna indietro dopo ogni vincita.

---

🧭 ROADMAP DI SVILUPPO

Fase 1: Analisi e Modellazione

[ ] Raccogliere dati storici completi (1000+ partite)

[ ] Calcolare: frequenza, ritardo medio/massimo/attuale per ogni segno

[ ] Calcolare quote medie disponibili

[ ] Validare capture rate delle strategie

---

Fase 2: Costruzione logica e moduli

[ ] Funzione calcolaProbabilitàStimata(...)

[ ] Funzione calcolaEV(p, quota)

[ ] Funzione shouldPlay(...) → ritorna true/false

[ ] Funzione calcolaPuntata(...) con D’Alembert

---

Fase 3: Interfaccia

[ ] UI per inserimento dati segno/ritardo/quote

[ ] Calcolo live del valore atteso e decisione

[ ] Storico scommesse

[ ] Progresso bankroll

[ ] Formule automatiche EV e decisione

[ ] Tabella storico eventi

[ ] Grafico ritardi

[ ] Simulatore bankroll

---

Fase 4: Sicurezze & Gestione Rischio

[ ] Definizione Stop-Loss (es. -6 step D’Alembert)

[ ] Target di uscita (es. profitto ≥ X)

[ ] Notifica quando ritardo supera massimo

---

Fase 5: Test & Ottimizzazione

[ ] Backtest su storico reale

[ ] Validazione efficienza EV

[ ] Monitoraggio capture rate aggiornato

[ ] Simulazione bankroll per 500 eventi

---

📎 Deliverables

Interfaccia (con UI interattiva, storico puntate, barra di avanzamento)

---

🔐 Regole d’oro del sistema

Situazione Azione

Ritardo ≥ 110% della media Giocabile

Ritardo ≥ 90% del massimo Giocabile, pressione

EV ≥ 0 Giocabile matematicamente

Entrambe vere Avvio D’Alembert

EV < 0 NON giocare

Ritardo > massimo Valutazione extra rischio

---

🔚 Conclusione

Con questa logica, un programmatore può sviluppare un sistema automatico e intelligente che combina:

Probabilità reali,

Logica statistica storica,

Progressione matematica (D’Alembert),

Calcolo EV per la sostenibilità a lungo termine.