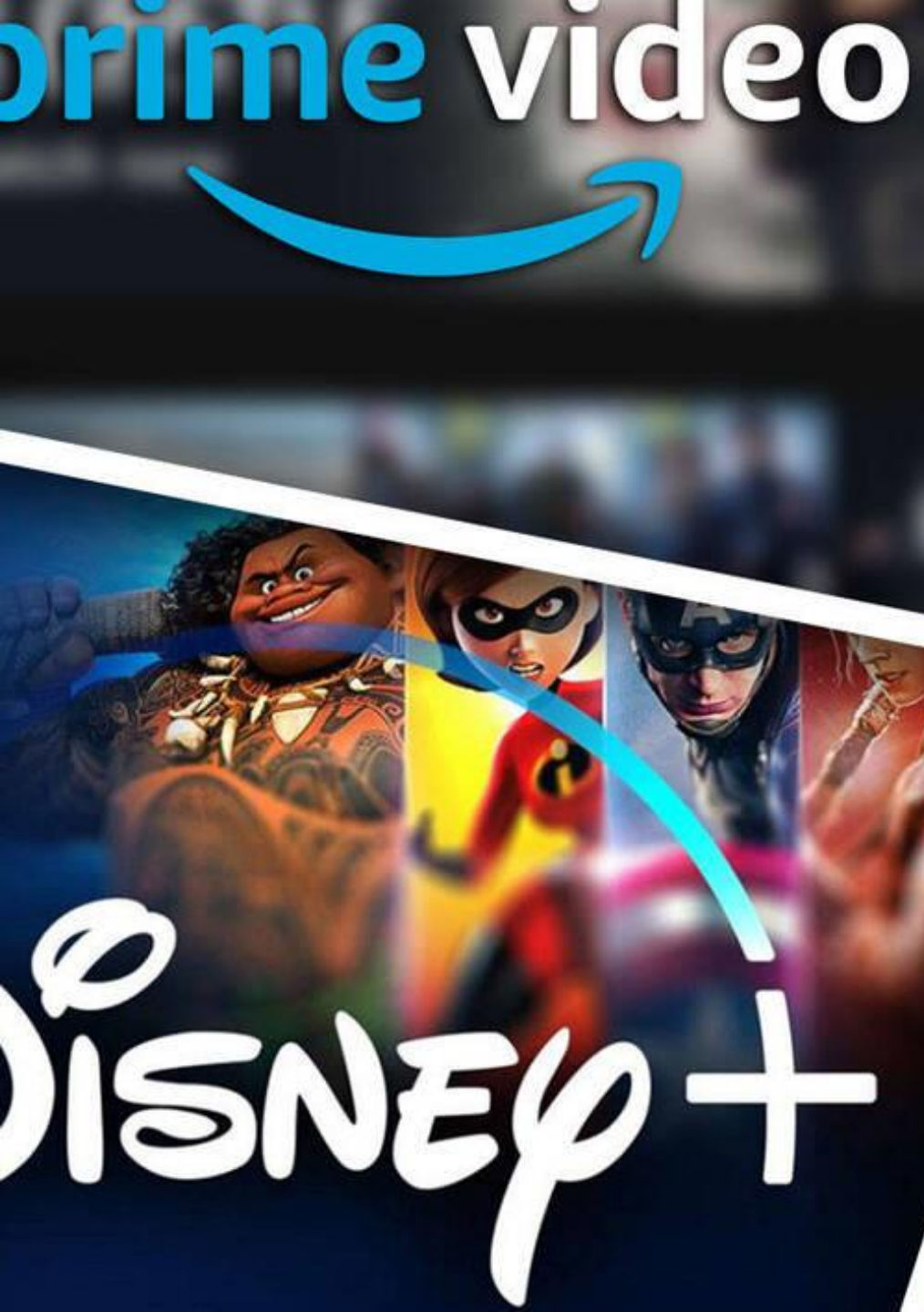


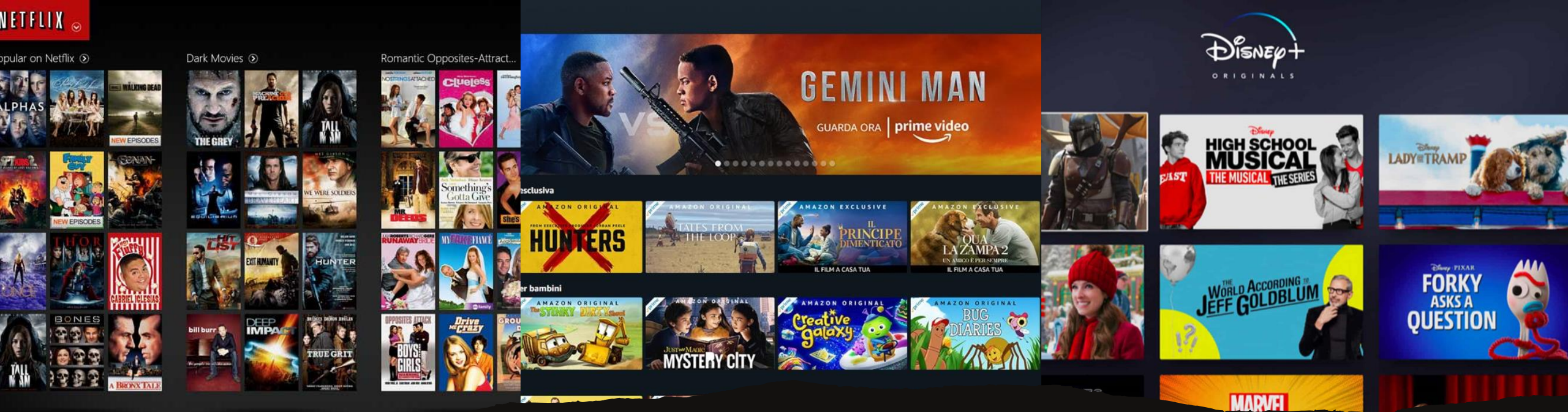
[illegible]



NETFLI

INTRODUZIONE

NOW T



- Nell'ultimo periodo la popolarità delle serie tv è aumentata in maniera esponenziale, anche grazie alla fruibilità sulle piattaforme streaming.
- Il problema di queste piattaforme risiede nel sistema di suggerimento dei contenuti limitati ai prodotti presenti su di esse.
- La risposta a questo problema è TV SERIES' WORLD!



-
- TV SERIES' WORLD è un'applicazione che si pone l'obiettivo di soddisfare a pieno le necessità degli utenti che non si limitano a fruire di un'unica piattaforma streaming.
 - Tramite l'inserimento delle serie già viste, il sistema sarà in grado di suggerire prodotti simili per garantire un'esperienza audiovisiva completa e portare il telespettatore alla scoperta di mondi fantastici ed ancora sconosciuti.

SPECIFICA PEAS

PERFORMANCE: La misura di performance dell'ambiente è la capacità di consigliare le serie tv che l'utente intende vedere.

ENVIRONMENT: L'ambiente è il frame con cui l'utente interagisce. Esso è:

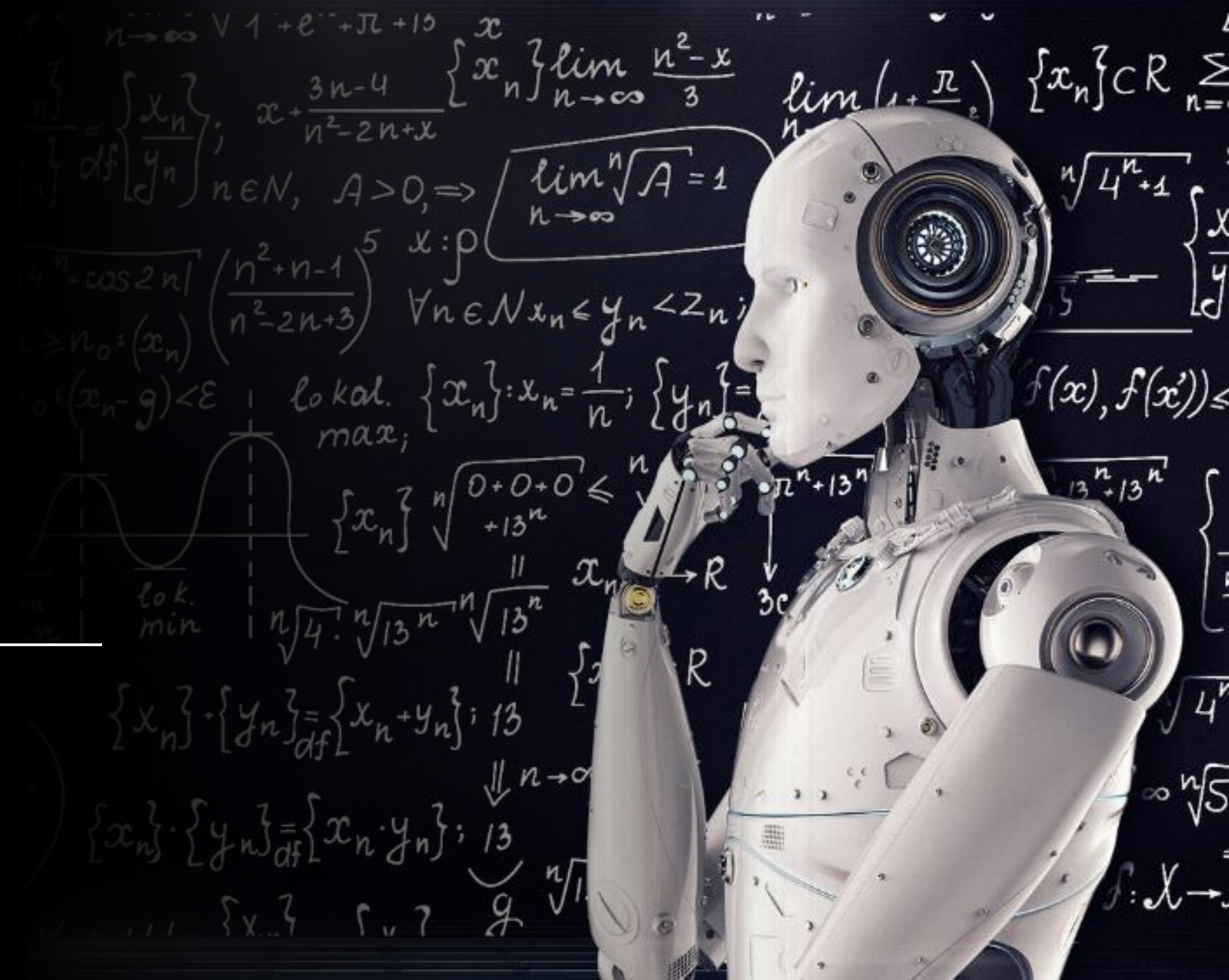
- Statico: l'agente agisce soltanto sull'insieme di telefilm corrente;
- Completamente osservabile: in ogni momento l'agente conosce le caratteristiche delle serie tv;
- Singolo: un unico agente opera nell'ambiente preso in esame;
- Episodico: le decisioni future dell'agente non sono influenzate dalle scelte passate dell'utente;
- Stocastico: lo stato successivo non è possibile determinarlo in quanto sono presenti più componenti di probabilità, oltre ad elementi randomici;
- Discreto: l'ambiente fornisce un numero limitato di percezioni ed azioni;

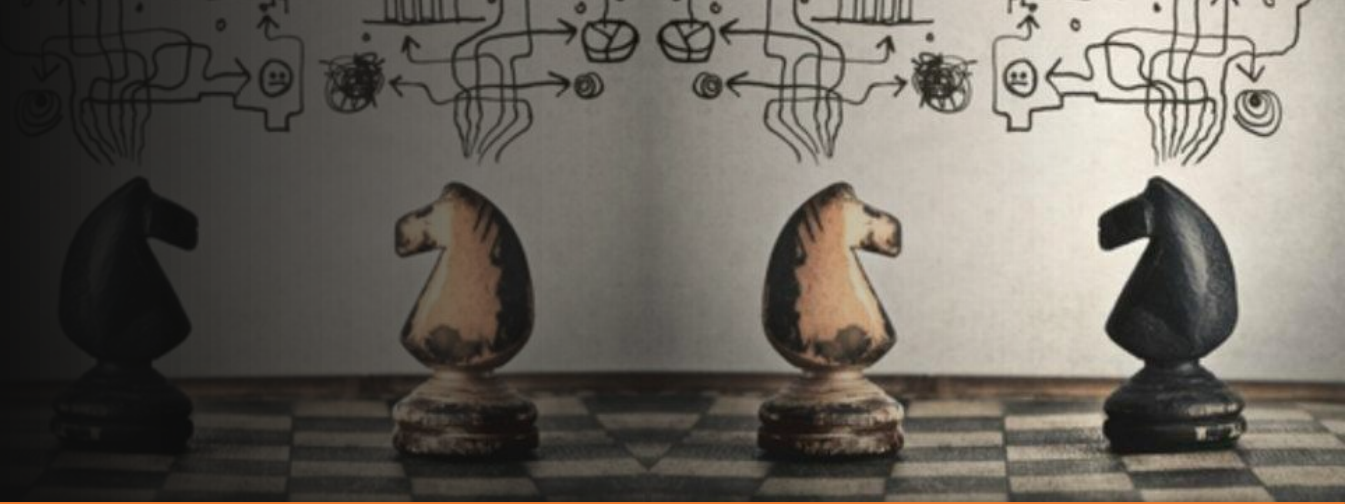
ACTUATORS: Gli attuatori dell'agente consistono nella lista dei prodotti consigliati sulla base delle preferenze;

SENSORS: i sensori consistono nella lista in input delle serie tv già viste, oltre al bottone "Invia" presente nella versione grafica del progetto.



ANALISI DEL PROBLEMA





- Individuato il problema abbiamo valutato diverse opzioni prima di procedere all'implementazione vera e propria.
- Innanzitutto abbiamo pensato ad un modello ML dato che le principali piattaforme di streaming lo utilizzano per il loro sistema di suggerimenti, ma la mancanza di un dataset adattabile a questo problema ci ha portato a scartare questa opzione.
- Ci siamo diretti quindi verso un algoritmo di ottimizzazione capace di risolvere il nostro problema.
- Avendo un ambiente singolo è stata scartata l'ipotesi di una soluzione mediante le tecniche presentate dalla teoria dei giochi.
- Inoltre, la mancanza di chiare relazioni tra gli elementi del nostro dataset ci ha portato a scartare l'ipotesi di rappresentare il problema come un grafo.
- Quindi abbiamo deciso di implementare una soluzione tramite un algoritmo genetico.





RACCOLTA DEI DATI



Per la scelta implementativa fatta, era necessario l'uso di un dataset e le possibili strade da seguire erano due:


1. **Creare** un dataset da zero, inserendo manualmente quanti più prodotti audiovisivi possibili con le relative caratteristiche;
2. **Cercare** sulla rete un dataset già formato e adeguarlo alle nostre esigenze;

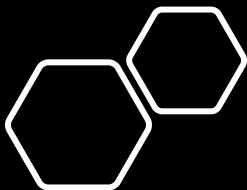
La prima opzione era soggetta a limitazioni non trascurabili:

- Non affidabilità dei dati;
- Lentezza nel creare dati per avere un numero di prodotti considerevole per il testing dell'algoritmo;

Si è deciso, pertanto, di procedere con la seconda soluzione, cercando in rete un dataset già pronto.

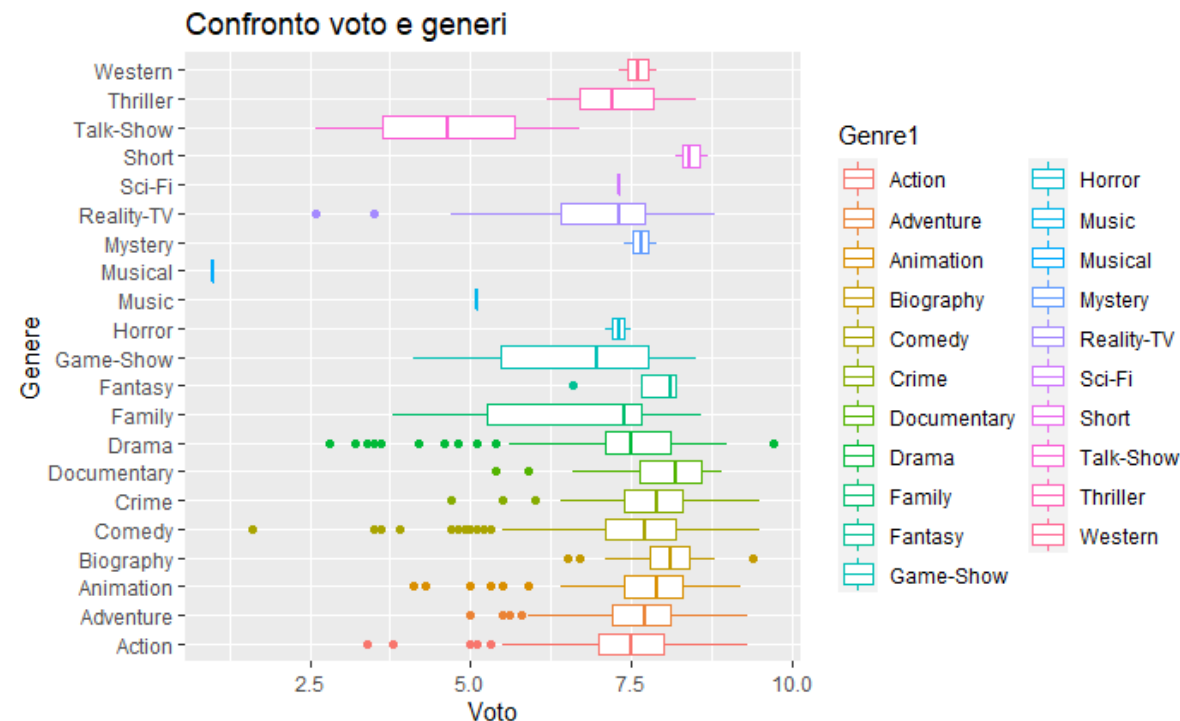


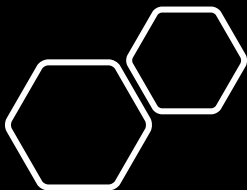
- 
- The background image shows a hand reaching towards a glowing digital interface. The interface features several data visualization elements: a bar chart on the left with blue bars, a central area with a large blue bell curve and a yellow arrow pointing upwards and to the right, and a right-hand panel with three distinct boxes. The top box shows '65 %' with a small bar chart and the text 'Growth'. The middle box shows '+8.3 %' with a bar chart and the text 'Sales'. The bottom box shows '8276' and 'Profit ea'. The overall aesthetic is high-tech and data-driven.
- Una volta ottenuto il dataset *IMDB TV Series* l'abbiamo caricato in una sessione R in modo da studiarne le variabili e gettare le basi, tramite statistiche descrittive, per la progettazione della funzione di valutazione del nostro algoritmo genetico.
 - Il dataset, in formato CSV, conteneva 1283 osservazioni e 17 variabili.
 - Una volta splittata la variabile Genre in Genre1, Genre2 e Genre3 ed eliminati i record con valori NA, abbiamo prodotto statistiche descrittive (sia grafiche che non) per valutare la correlazione tra le variabili.



CONFRONTO TRA VOTO E GENERE

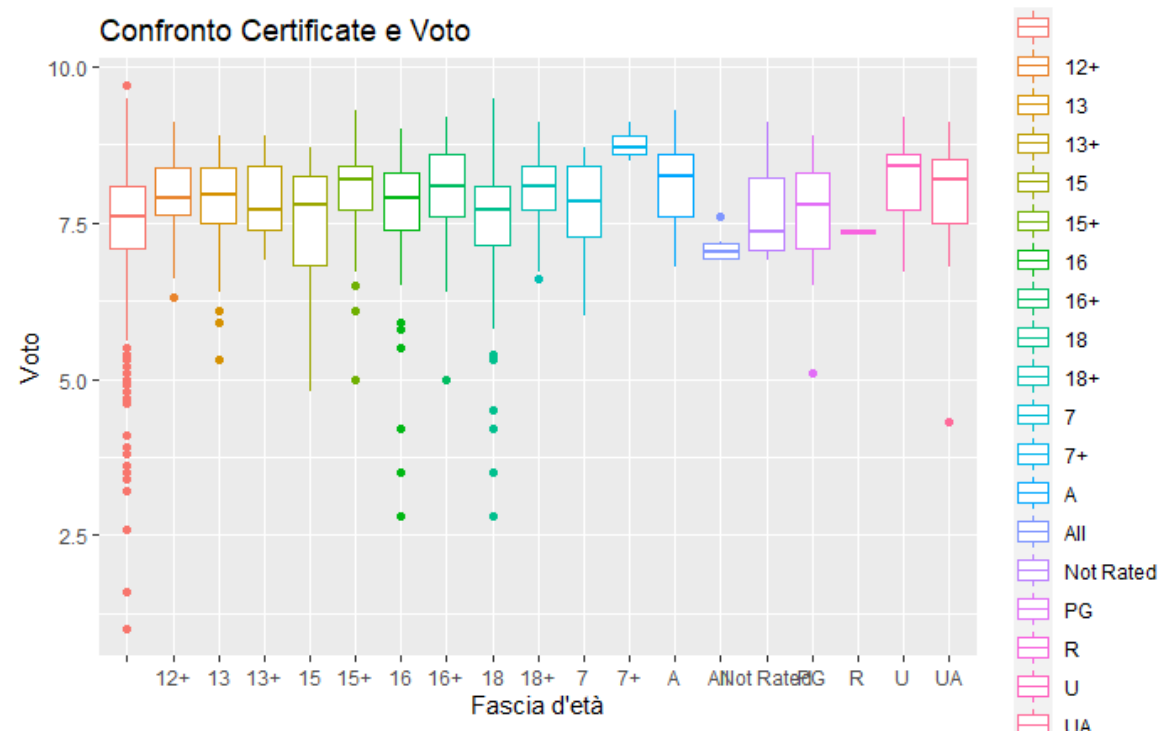
- Innanzitutto, abbiamo calcolato l'indice di Cramèr per queste due variabili ed è risultato pari a 0.33 indicando una correlazione tutto sommato forte come visibile anche dal boxplot prodotto.
- Inoltre, il grafico evidenzia molti punti distanti dalla mediana della distribuzione ma questo era previsto in quanto, parlando di serie tv, i voti possono variare di molto in base al telefilm in questione.





CONFRONTO TRA VOTO E CERTIFICAZIONE

- La variabile Certificate indica la fascia di età consigliata per la fruizione del prodotto in questione.
- Abbiamo quindi pensato che questa potesse avere un peso sull'indice di gradimento dell'audience ma siamo stati subito smentiti dai dati.
- L'indice di Cramèr per queste variabili è pari a 0.19, correlazione che sussiste ma è troppo bassa per essere rilevante come si nota anche nel grafico.





ALGORITMO GENETICO

PASSI DELL'ALGORITMO

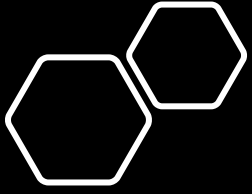


1. Prendere in input serie tv inserite dall'utente;
2. Costruire una popolazione iniziale;
3. Codificare la popolazione iniziale in una lista di 0 e 1, in cui 0 equivale a *"non consigliare"* e 1 equivale a *"consigliare"*;
4. Calcolare il valore di fitness;
5. Selezionare un sottoinsieme di individui in base al loro punteggio di fitness;
6. Gli individui selezionati vengono fatti accoppiare con una determinata probabilità di crossover (pari a 0.8);
7. Gli individui ottenuti vengono fatti mutare con una probabilità di mutazione pari a 0.01 e si ritorna al passo 4;
8. Quando la dimensione della popolazione è minore di un certo valore oppure allo scadere del tempo di esecuzione l'algoritmo termina e restituisce la generazione attuale che rappresenta la migliore.



INIZIALIZZAZIONE E CODIFICA

- La versione finale del progetto inizializza la popolazione estraendo un sottoinsieme di serie tv con i generi uguali a quelli delle serie inserite in input dall'utente. Sulla base di questo si scelgono tra le 3 e le 15 serie tv in modo casuale e si passano alla funzione di fitness.
- In seguito la popolazione verrà codificata casualmente in una lista di 0 e 1.



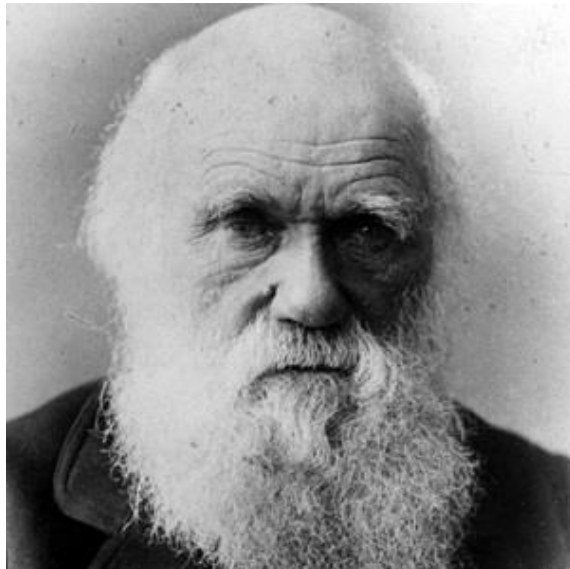
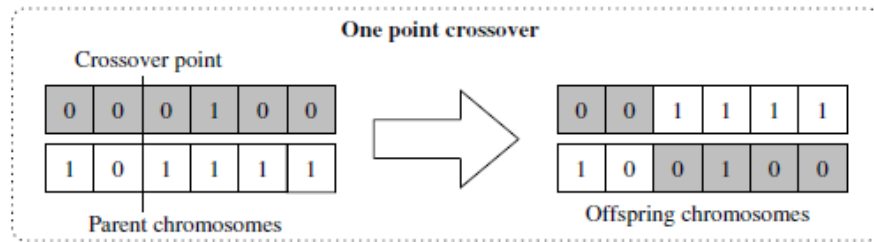
FUNZIONE DI FITNESS

La funzione di fitness a cui si è giunti dopo svariati tentativi si basa su:

- **Generi simili:** abbiamo predisposto un dizionario python per tenere traccia dei generi simili tra loro. La chiave è un genere univoco del dataset e i corrispondenti valori sono i generi simili ad essa. Se i generi della serie presa in considerazione NON rientrano tra i generi simili a quelli delle serie dell'utente, il punteggio di fitness viene ridotto del 50%;
- **IMBD Rating:** la funzione di fitness aggiunge al punteggio di fitness 0.3 se il voto della serie presa in considerazione è maggiore di quello della serie inserita dall'utente;
- **Overview:** grazie ad una lista di parole chiave, la funzione di fitness confronta le sinossi delle 2 serie e per ogni keyword in comune aggiunge al punteggio di fitness 0.5.

```
def fitness(lista_utente, lista_non_codificata, lista_codificata):  
    lista_fit=[]  
    global keywords  
    for z in range(len(lista_codificata)):  
        fit=0  
        for i in range(len(lista_non_codificata)):  
            first=False  
            if lista_codificata[z][i]!=0:  
                for j in range(len(lista_utente)):  
                    if lista_non_codificata[i][5]>=lista_utente[j][5] and first==False:  
                        fit+=0.3  
                        first=True  
                for k in keywords:  
                    if k in lista_non_codificata[i][-1].lower() and k in lista_utente[j][-1].lower():  
                        fit+=0.5  
            genere_consiglio=lista_non_codificata[i][4].split(", ")  
            genere_utente=lista_utente[j][4].split(", ")  
            for x in genere_consiglio:  
                for y in genere_utente:  
                    if x not in consiglia_genre[y]:  
                        fit+=(fit*50)/100  
            lista_fit.append([lista_codificata[z], fit])  
    return lista_fit
```


SELEZIONE E CROSSOVER

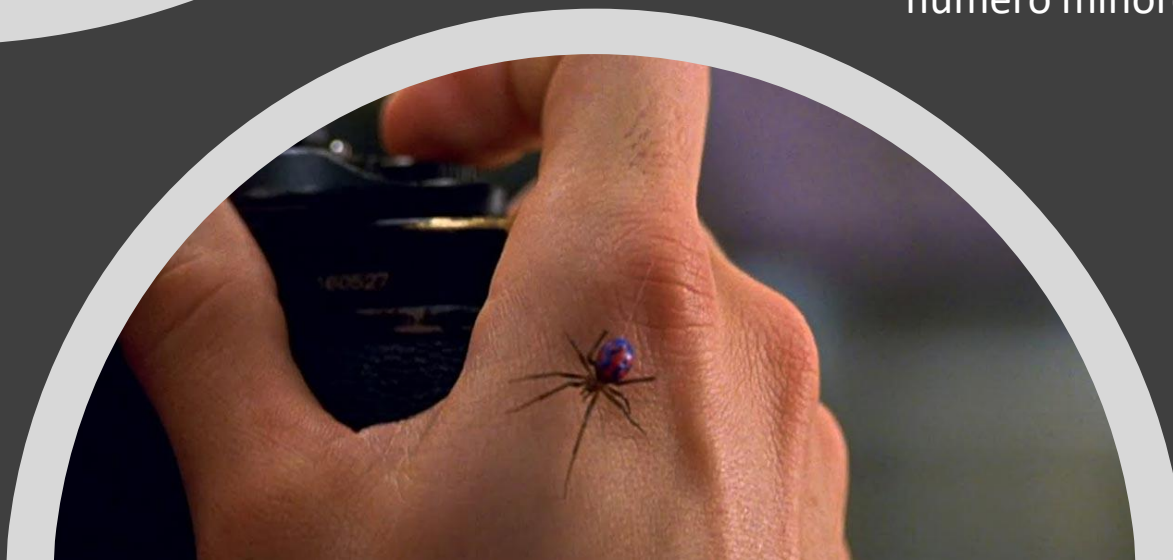


Per scelta progettuale, la selezione avviene tramite *Two Tournament* mentre il crossover attraverso il *Single Point Crossover* con probabilità pari all' 80%.



MUTAZIONE E STOPPING CONDITION

- Per semplicità abbiamo deciso di implementare il tipo di mutazione *Random Resetting*, con probabilità di mutazione pari a 1%.
- Infine, basandoci sull'immediata risposta che l'utente vuole ricevere, l'algoritmo termina entro massimo 2 secondi oppure quando la selezione *Two Tournament* riduce i partecipanti al "torneo" ad un numero minore o uguale di 2.



TvSeriesWorld

Inserisci qui il titolo di una serie tv che hai visto!

CONSIGLIA

INVIA

Le serie che hai inserito sono:
[Arrow', '(2012–2020)', 'A', '42 min', 'Action, Adventure, Crime', 7.5, 'Stephen Amell', 'Katie Cassidy']
[Daredevil', '(2015–2018)', '18+', '54 min', 'Action, Crime, Drama', 8.6, 'Charlie Cox', 'Vincent D'Onofrio']
[Supergirl', '(2015–2021)', '12+', '43 min', 'Action, Adventure, Drama', 6.3, 'Melissa Benoist', 'Chyler Leigh']

I consigli forniti da WorldSeriesTV sono:
Titolo:Tokyo Ghoul Genere: Animation, Action, Drama
Sinossi:
A Tokyo college student is attacked by a ghoul, a superpowered human who feeds on human flesh. He survives, but has become part ghoul and becomes a fugitive on the run.

Titolo:Flames Genere: Comedy, Drama, Romance
Sinossi:
A teenage romance from The Timeliners that aims straight from the heart. This web-series is the story of a young romance unfolding as a chemical reaction.


Titolo:Fate/stay night: Unlimited Blade Works Genere: Animation, Action, Fantasy
Sinossi:
A group of seven mages gets chosen to become masters of seven classes of heroic spirits, in order to fight and win the Holy Grail.

Titolo:Alex Rider Genere: Action, Adventure, Family
Sinossi:
Alex Rider is an ordinary teenager enlisted to work on behalf of MI6, where he uses skills he didn't know he had to become an extraordinary spy.

Titolo:Knightfall Genere: Action, Adventure, Drama
Sinossi:
A look at the final days of the Knights Templar during the 14th century.

A presto!

ESEMPIO DI UTILIZZO

The background image shows a human hand on the left and a white robotic hand on the right, both reaching towards the center. The background is dark with a grid pattern and several bright, diagonal light rays. A small orange rectangle is in the top left corner.

Concludiamo dicendo che siamo soddisfatti del lavoro svolto e nonostante i risultati dell'algoritmo non siano particolarmente brillanti, la soluzione offerta raggiunge gli obiettivi che ci eravamo inizialmente prefissati