

EduProfiler

Progetto per il corso di **Fondamenti di Intelligenza Artificiale (FIA)** della facoltà di informatica dell'università degli studi di Salerno.



Un progetto di **Battaglia Daniel** e **Pennarella Fabio**

# Introduzione

- L'**intelligenza artificiale (IA)** è diventata una componente fondamentale non solo nel campo informatico, ma del nostro quotidiano. È importante conoscerne le potenzialità, così da poterla utilizzare efficientemente. Il nostro progetto include l'uso dell'IA per soddisfare il problema che successivamente descriveremo.
- Lo scopo del progetto è creare **un'applicazione desktop** che permette all'utente di inserire i dati di uno studente, e di calcolarne un "indice accademico" in base al suo andamento.

## Specifiche PEAS

- Le specifiche **PEAS** (Performance measure, Environment, Actuators, Sensors) sono un modello utilizzato in ambito **IA** per descrivere in modo chiaro ed efficiente un problema. In altre parole, le PEAS sono un insieme di specifiche che definiscono il comportamento e le caratteristiche di un agente intelligente. Ogni elemento del modello PEAS serve a specificare un aspetto fondamentale di come l'agente percepisce l'ambiente e come interagisce con esso.

## Performance Measure (Misure di prestazione)

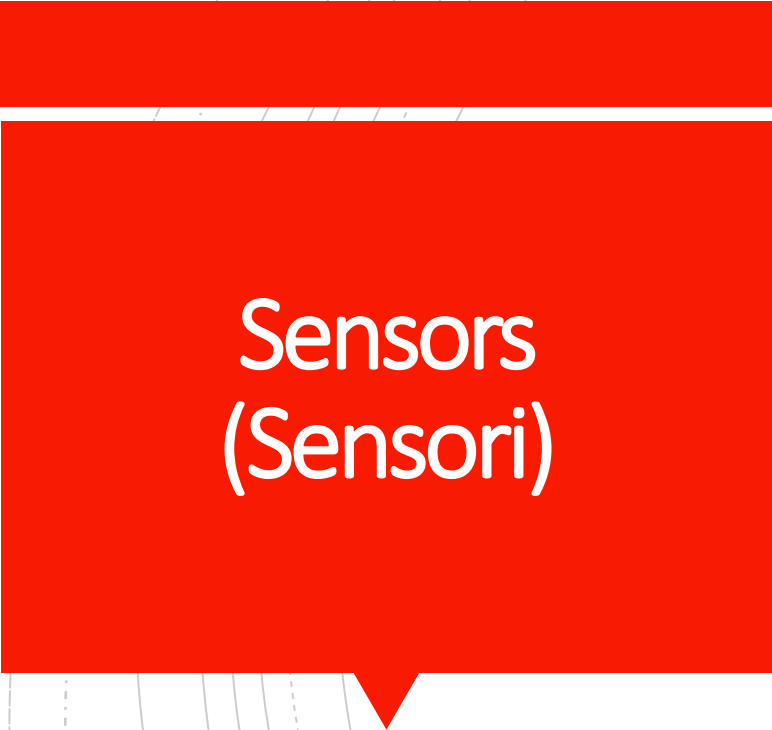
- Come misura di prestazione del nostro applicativo si tiene in conto della sua abilità di dare un risultato corretto rispetto al risultato atteso, e che il tutto venga svolto dall'applicativo in un tempo ragionevole.

## Environment (Ambiente)

- Nel caso del nostro applicativo ci troviamo in un ambiente che è:
  - **Completamente osservabile**, in quanto l'agente ha accesso a tutte le informazioni rilevanti sull'ambiente (quali: media voti di ogni studente, media ore di studio di ogni studente, numero attività extra di ogni studente).
  - **Deterministico**, in quanto l'esito di una previsione è completamente prevedibile e non dipende da fattori casuali.
  - **Statico**, in quanto l'ambiente non cambia mentre l'agente prende decisioni.
  - **Episodico**, in quanto le azioni dell'agente sono indipendenti, e ogni decisione riguarda un singolo episodio senza influenzare i successivi.
  - **Agente singolo**, in quanto c'è un solo agente che interagisce con l'ambiente.

## Actuators (Attuatori)

- Gli **attuatori** rappresentano il meccanismo attraverso cui l'agente restituisce una risposta o influenza l'ambiente. Gli attuatori del sistema sono:
- **Output del risultato:**
  - Il sistema apprende l'indice accademico del nuovo studente e lo mostra su linea di comando.

A red speech bubble graphic with a white outline, pointing downwards. It contains the text 'Sensors (Sensori)' in white. The background of the slide features faint, curved, concentric lines in the top-left and bottom-right corners.

## Sensors (Sensori)

- I **sensori** rappresentano il meccanismo attraverso cui l'agente acquisisce informazioni sull'ambiente. I sensori del sistema sono:
- **Dataset:**
  - Al sistema viene reso disponibile un dataset di studenti etichettati (media voti, media ore di studio settimanali, numero attività extra, indice accademico) per poter così predire l'indice accademico di un nuovo studente.
- **Interfaccia Utente:**
  - I dati di un nuovo studente di cui dovrà essere predetto l'indice accademico vengono inseriti dall'utente tramite un'interfaccia.

## Analisi del problema

- Il progetto è caratterizzato da un unico agente, il cui scopo è quello di apprendere l'indice accademico di un nuovo studente, in base ai valori inseriti dall'utente, ed apprendendo da un dataset attraverso un **algoritmo di machine learning**, ovvero la componente **IA**.
- Ogni studente possiede tre caratteristiche:
  - **Media voti;**
  - **Media ore di studio settimanali;**
  - **Numero di attività extra-curricolari che svolge.**



## Analisi del problema (2)

- Ogni caratteristica ha associata al suo effettivo valore una sotto-categoria da tre possibili valori: “bassa”, “media” e “alta”. Per ogni sotto-categoria ci sono dei **range**:
  - Media voti:
    - bassa:  $\geq 18$  &  $\leq 22$ ;
    - media:  $\geq 23$  &  $\leq 26$ ;
    - alta:  $\geq 27$  &  $\leq 30$ ;
  - Media ore di studio:
    - bassa:  $\geq 1$  &  $\leq 10$ ;
    - media:  $\geq 11$  &  $\leq 20$ ;
    - alta:  $\geq 21$ ;
  - Numero di attività extra-curricolari:
    - bassa:  $\geq 0$  &  $\leq 2$ ;
    - media:  $\geq 3$  &  $\leq 5$ ;
    - alta:  $\geq 6$ .

## Analisi del problema (3)

- Quindi, ogni studente, oltre ai suoi 3 attributi principali, avrà una categoria, il cui valore è una combinazione tra gli elementi di questo insieme {"bassa", "media", "alta"}.
- Il **dataset** sarà composto da **27 studenti**, numero sufficiente per gestire tutte le combinazioni ( $3^3$ ).

# L'indice accademico

- L'**indice accademico** ha lo scopo di indicare l'efficienza accademica di un singolo studente in base ai valori dei suoi attributi (media voti, media ore di studio settimanali e numero attività extra-curricolari). Per ogni studente nel dataset, l'indice accademico verrà calcolato attraverso i seguenti criteri:
- Ogni attributo ha un punteggio:
  - bassa: 1;
  - media: 2;
  - alta: 3;
- La somma dei punteggi di ogni attributo indicherà il valore dell'indice accademico (ex:  $3+1+1$ ):
  - basso:  $\leq 3$ ;
  - medio:  $\leq 6$ ;
  - alto:  $\geq 7$ .
- Per i nuovi studenti, l'indice accademico non verrà calcolato, ma predetto attraverso l'**IA**.

# Machine Learning

- Quello che stiamo trattando è un **problema di classificazione**, la cui definizione è la seguente:

*“Task in cui l’obiettivo è predire il valore di una variabile categorica, chiamata variabile dipendente, target, o classe, tramite l’utilizzo di un training set, ovvero un insieme di osservazioni per cui la variabile target è nota”.*

Che tipo di  
problema stiamo  
affrontando?

- Nel nostro caso ci troviamo d'avanti ad un problema di classificazione, in quanto l'obiettivo dell'agente intelligente è quello di predire quale potrebbe essere l'indice accademico di uno studente sulla base delle sue caratteristiche, e quindi quello di classificare uno studente secondo le 3 possibili classi dell'indice accademico, ovvero:
  - «Basso»
  - «Medio»
  - «Alto»

# Classificazione

- I problemi di classificazione sono istanze di problemi di apprendimento supervisionato, e quindi di ***machine learning***, dove l'apprendimento avviene attraverso un **training set** (un dataset di dati etichettati).
- L'**etichetta** determina la variabile dipendente, ovvero quello che l'agente dovrà apprendere. Nel nostro contesto, l'etichetta è l'indice accademico, e l'agente ne deve apprendere il valore.

## Quale algoritmo abbiamo scelto?

- L'algoritmo usato è l'**albero decisionale**, il quale mira a creare un albero i cui nodi rappresentano gli attributi/caratteristiche, e i cui archi ne rappresentano i valori (decisioni). Ecco il suo svolgimento:
  - La miglior caratteristica del training set viene posizionata alla radice;
  - Il training set viene diviso in sotto-insiemi, ognuno composto da valori simili per una certa caratteristica;
  - I primi due step vengono ripetuti fino a quando non viene raggiunto un nodo foglia in ogni sotto-albero.

## Perché un algoritmo di apprendimento?

- Abbiamo considerato, secondo il problema da risolvere, più adatto un algoritmo di apprendimento invece che di ricerca. Ecco alcuni vantaggi:
  - **Generalizzazione:** è in grado di generalizzare i risultati, ovvero di trovare soluzioni di dati non ancora esistenti, attraverso appunto l'apprendimento del dataset;
  - **Scalabilità:** Gli algoritmi di machine learning sono progettati per funzionare bene con dataset di qualsiasi dimensione. Se in futuro il dataset cresce o vengono introdotti nuovi attributi, l'algoritmo di apprendimento può essere riaddestrato per integrare i nuovi dati senza dover riscrivere l'intera logica;
  - **Efficienza computazionale:** Una volta addestrato, il modello è molto rapido nel fare predizioni. Non ha bisogno di calcolare soluzioni da zero ogni volta che viene presentato un nuovo input.



## Scelta dell'attributo

- Per scegliere quale attributo debba dividere il dataset è stato utilizzato l'**Information Gain**, ovvero: *"la misura che indica il grado di purezza di un attributo, ovvero quanto un certo attributo sarà in grado di dividere adeguatamente il dataset"*.
- Alla base dell'information Gain è presente l'**entropia**, che nella teoria dell'informazione indica in che misura un messaggio è ambiguo e difficile da capire.

# Information Gain

- L'**Information Gain** è una misura utilizzata negli alberi decisionali per determinare quale attributo utilizzare come nodo in un punto specifico dell'albero.
- Nel caso del nostro applicativo, che calcola la previsione dell'indice accademico di uno studente tramite l'utilizzo di un albero decisionale, si ha bisogno di scegliere quali degli attributi di uno studente (scelto tra: media voti, media ore di studio settimanali, numero attività extra) dovrà essere utilizzato come nodo radice dell'albero o di un sottoalbero. Dato però che i vari valori contrastanti degli attributi all'interno del dataset appaiono con la stessa frequenza, l'Information Gain risulterà uguale per tutti.

$$Gain(D, A) = H(D) - \sum_{v \in values(A)} \frac{|D_v|}{|D|} \cdot H(D_v)$$

dove

- $D$  è l'entropia del dataset;
- $D_v$  è il sottoinsieme di  $D$  per cui l'attributo  $A$  ha valore  $v$ ;
- $|D_v|$  è il numero di elementi di  $D_v$ ;
- $|D|$  è il numero di elementi del dataset.

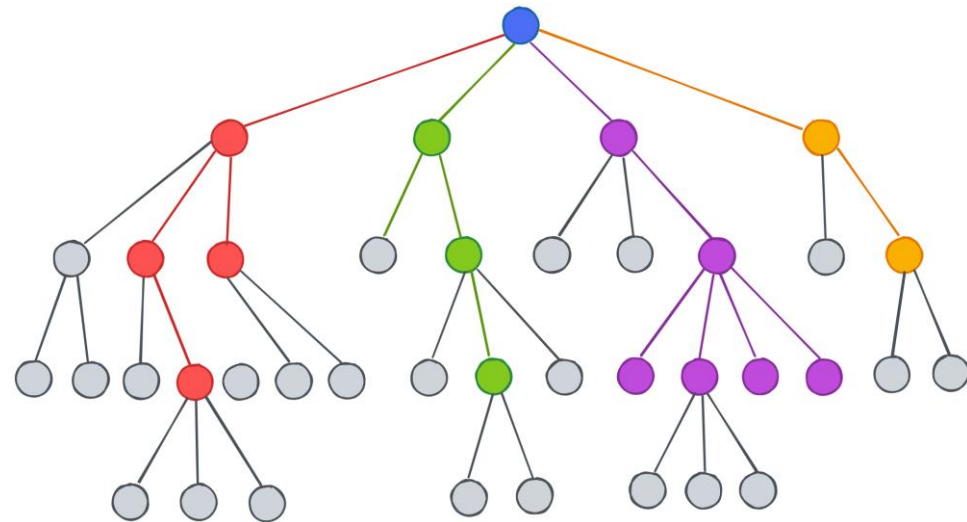
# Entropia

- L'**Entropia** serve per misurare il livello di “disordine” o incertezza in un insieme di dati. In una classificazione, l'entropia quantifica quanto i dati sono misti in termini delle loro etichette.
- Nel nostro caso si vuole calcolare l'entropia del dataset iniziale dato in input alla nostra applicazione, ossia un insieme di dati su 27 studenti, etichettati tramite l'indice accademico.

$$H(D) = - \sum_c p(c) \cdot \log_2 p(c) \quad \text{dove } p(c) \text{ è la proporzione della classe } c \text{ nel dataset } D.$$

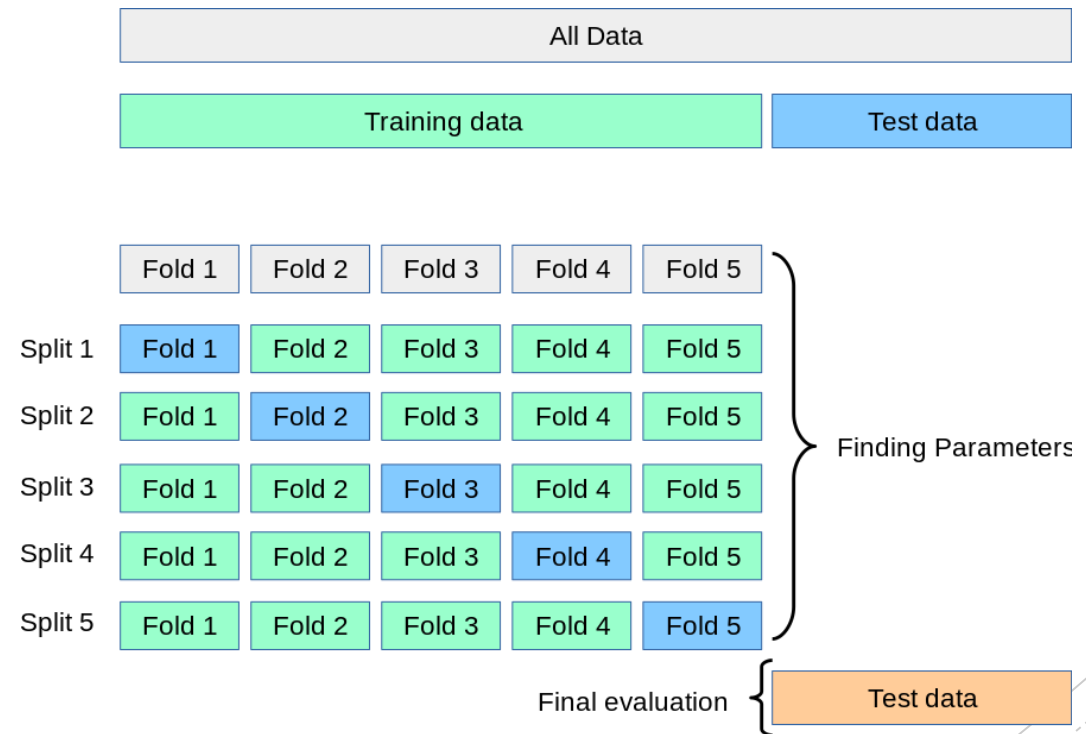
# Albero decisionale

- La costruzione dell'albero decisionale parte dalla generazione del nodo radice, costruendo in maniera ricorsiva i sottoalberi di ogni nodo, la cui fine è stabilita dai nodi foglia, ovvero i valori dell'indice accademico da apprendere. L'attributo di ogni nodo viene scelto attraverso il calcolo dell'information gain.
- Ogni nodo viene costruito ricorsivamente. Le liste sinistra, destra e centrale suddividono gli studenti in base al valore dell'attributo scelto (bassa, media, alta). Una volta analizzati i tre attributi di uno studente, e quindi i tre nodi verticali, viene generato il nodo foglia con il valore dell'indice accademico corrispettivo.



# KFoldCrossValidation

- Per completare la risoluzione del nostro problema e migliorare le performance del nostro modello, una delle ottimizzazioni più utili è l'implementazione della validazione incrociata (cross-validation). Questo processo è cruciale per valutare la capacità di generalizzazione del nostro modello, riducendo il rischio di overfitting e ottenendo stime più robuste delle sue prestazioni. In particolare, in questo contesto, utilizzeremo la **K-Fold Cross Validation**, che è una delle tecniche più comunemente adottate nel machine learning per validare i modelli predittivi.



## Considerazioni finali

- L'adozione **dell'albero decisionale** come algoritmo di apprendimento per il nostro problema si rivela la scelta più appropriata rispetto ad altre tecniche come il KNN o il Naive Bayes. Questo approccio si dimostra particolarmente adatto grazie alla natura delle variabili categoriali, che non sono numeriche ma altamente correlate tra loro, e al dataset composto da 27 studenti.
- Nonostante i numerosi vantaggi, è importante sottolineare che l'uso dell'albero decisionale implica un rischio potenziale di **overfitting**. Tuttavia, nel contesto specifico del nostro dataset, che è completo e rappresentativo di tutte le possibili informazioni, questo rischio risulta essere minimo.
- Al contrario, il rischio di **underfitting** è contenuto, poiché gli attributi presenti nel dataset sono sufficientemente esplicativi per estrarre regole valide, catturare le differenze tra le diverse classi e ottenere modelli in grado di interpretare adeguatamente i dati.

# Strumenti utilizzati

Linguaggio di programmazione	Java
IDE	IntelliJ
jdk	21
Librerie GUI	Java Swing e Java AWT
Framework Testing	<p>JUnit</p> <p>Inclusione di 4 librerie nella cartella <b>lib</b>:</p> <ul style="list-style-type: none"><li>- junit-jupiter-api-5.10.0.jar;</li><li>- junit-jupiter-engine-5.10.0.jar;</li><li>- junit-platform-commons-1.10.0.jar</li><li>- junit-platform-engine-1.10.0.jar</li></ul> <p>Download da: <a href="#">Maven Central Repository</a></p>
Tool generazione documentazione	javadoc

Demo



## EduProfiler - Predizione dell'indice accademico di uno studente

Media voti:

Media ore di studio settimanali:

Numero attività extra-curricolari:

Predici inidice accademico



# Source

- È possibile trovare tutto il materiale nella nostra Repository GitHub.
- All'interno della Repository sarà possibile trovare:
  - File PDF con report del progetto;
  - File PDF di presentazione del progetto;
  - Implementazione (codice) del progetto;
- Link repository GitHub:
  - **EduProfiler - GitHub**