



# PROGETTO IA

di Giulio Vignati, Matteo Losa &  
Alessandro Pieragostini



# INTRODUZIONE

La nostra idea nasce dalla volontà di approfondire l'argomento dell'apprendimento automatico.

Questo programma Prolog implementa un sistema di apprendimento automatico per la classificazione degli infortuni dei giocatori in un campionato di basket. Utilizza tecniche di apprendimento supervisionato per creare un albero di decisione basato su un set di dati di addestramento, e poi utilizza questo albero per classificare nuovi oggetti (ovvero nuovi casi di giocatori).

# SOLUZIONE

- Pulizia Dataset iniziale
- Algoritmo di apprendimento automatico in PROLOG
- Induzione dell'albero di decisione con 2 criteri: Gini o Entropia
- Potatura dell'albero di decisione per valutare le variazioni dell'accuratezza
- Interazione tramite interfaccia Python per la classificazione

# PULIZIA DATASET

Eliminazione attributi considerati non influenti ai fini dell'apprendimento, ad esempio:

- Nome giocatore
- Nome squadra
- Anno stagione
- Ingressi in area/p
- Passaggi in area
- Tiri in sospensione da 3 punti/p
- Tocchi metà campo avversaria/p
- Tocchi su angolo
- Durata infortunio

# **PULIZIA DATASET**

**Nel dataset iniziale, per ogni record era indicato (se presente) il tipo specifico di infortunio subito. Per i fini del progetto si è semplificato unendo i vari tipi di infortuni in un'unica classe.**

**Per ogni attributo i valori sono stati classificati in intervalli.**

**Gli intervalli sono stati generati tramite funzione Python impostando il numero di “percentili” da utilizzare**

# APPRENDIMENTO

- Il predicato apprendi(Classe) raccoglie tutti gli esempi (istanze) disponibili e inizia a generare una descrizione (regola) per la classe specificata.
- Se non ci sono più esempi della classe data, il processo termina. Altrimenti, tramite il predicato apprendi\_cong(Esempi, Classe, Cong) genera una congiunzione di attributi-valori (Cong) che copre alcuni esempi della classe senza includere esempi di altre classi.

# APPENDIMENTO

- `scegli_cond(Esempi, Classe, AttVal)` seleziona una coppia attributo-valore (AttVal) tra tutte le possibili, assegnando un punteggio basato su quante volte riesce a discriminare la classe target dagli altri esempi.
- Viene scelto l'attributo-valore con il punteggio migliore.



# APPENDIMENTO

- Gli esempi che soddisfano la congiunzione attuale vengono filtrati dal set di esempi rimanenti tramite la il predicato `filtra(Esempi, Cond, NuoviEsempi)` filtra gli esempi in base alla condizione data.
- `rimuovi(Esempi, Cong, RestoEsempi)` rimuove gli esempi già coperti dalla congiunzione trovata, lasciando solo quelli non ancora coperti.



# INDUZIONE ALBERO

Per l'induzione dell'albero sono stati utilizzati due criteri differenti:

- Gini
- Entropia



# INDUZIONE ALBERO

$$Gini(S) = 1 - \sum_{i=1}^n p_i^2$$

- **Definizione:** È una misura della probabilità che un elemento scelto dal dataset sia classificato erroneamente se fosse etichettato in base alla distribuzione delle classi nel nodo.
- **Interpretazione:** Valori vicini a 0 indicano che tutti gli esempi appartengono alla stessa classe (purezza massima), mentre valori vicini a 0.5 indicano una distribuzione uniforme tra le classi (impurezza max).

# INDUZIONE ALBERO

$$Entropia(S) = - \sum_{i=1}^n p_i \log_2 p_i$$

- **Definizione:** L'entropia misura l'incertezza o impurità in un sistema. Rappresenta la quantità di imprevedibilità nella distribuzione delle classi all'interno di un nodo.
- **Interpretazione:** Valori vicini a 0 indicano bassa incertezza (alta purezza), mentre valori più alti indicano maggiore incertezza (impurezza).

# INDUZIONE ALBERO

## Differenze:

- **Intervallo di valori:** Il criterio di Gini varia tra 0 e 0.5 per due classi, mentre l'entropia varia tra 0 e 1.
- **Sensibilità:** L'impurezza di Gini è meno severa rispetto all'entropia. Questo significa che l'entropia tende a preferire nodi più puri.
- **Calcolo:** Il criterio di Gini è più semplice da calcolare rispetto all'entropia poiché non coinvolge logaritmi.

# POTATURA ALBERO

È stata effettuata la potatura dell'albero per ridurre la complessità dell'albero e vedere gli effetti sull'accuratezza.

Il predicato `lancia_sottoalberi` calcola tutti i sottoalberi procedendo in profondità, ritornando una lista di tutti i sottoalberi.



# POTATURA ALBERO

Per ognuno di essi, lo toglie all'albero originale e tramite il test del chi quadrato verifica se sono stati rimossi nodi dell'albero che non fanno aumentare l'informazione in maniera statisticamente significativa.

In caso affermativo la potatura viene mantenuta.

$$\chi^2 = \sum_{i=1}^k \frac{(o_i - e_i)^2}{e_i}$$

# RISULTATI



Matrice di confusione con  
entropia senza potatura  
dell'albero:

```
Test effettuati :470
Test non classificati :45
Veri sani 159    Falsi infortuni 49
Falsi sani 42    Veri infortuni 175
Accuratezza: 0.7858823529411765
Errore: 0.21411764705882352
Precisione: 0.8064516129032258
```

# RISULTATI



Matrice di confusione con  
entropia con potatura  
dell'albero:

```
Test effettuati :470
Test non classificati :56
Veri sani 159      Falsi infortuni 42
Falsi sani 42      Veri infortuni 175
Accuratezza: 0.8
Errore: 0.2
Precisione: 0.81
```



# RISULTATI



Matrice di confusione con  
Gini senza potatura  
dell'albero:

```
Test effettuati :470
Test non classificati :44
Veri sani 149      Falsi infortuni 58
Falsi sani 47      Veri infortuni 172
Accuratezza: 0.7535211267605634
Errore: 0.24647887323943662
Precisione: 0.7853881278538812
```

# RISULTATI



Matrice di confusione con  
Gini con potatura dell'albero:

```
Test effettuati :470
Test non classificati :59
Veri sani 149    Falsi infortuni 47
Falsi sani 47    Veri infortuni 172
Accuratezza: 0.77
Errore: 0.23
Precisione: 0.79
```



Apprendimento intelligente NBA

Età	<input type="text"/>
Altezza	<input type="text"/>
Peso	<input type="text"/>
Partite giocate	<input type="text"/>
Minuti a partita	<input type="text"/>
Impiego attivo (%)	<input type="text"/>
Ritmo di gioco	<input type="text"/>
Possessi per stagione	<input type="text"/>
Tiri tentati a partita	<input type="text"/>
Attacchi al canestro a partita	<input type="text"/>
Miglia percorse a partita	<input type="text"/>
Velocità media	<input type="text"/>
Tiri in sospensione a partita	<input type="text"/>
Frequenza di palleggio	<input type="text"/>
Media dribbling per tocco	<input type="text"/>
Tocchi spalle a canestro	<input type="text"/>
Tocchi nel pitturato	<input type="text"/>

Costruisci la query

Potatura

Metodo di induzione:

Entropia

Entropia

Gini

# INTERFACCIA

## Funzionalità:

- Inserimento di valori numerici nelle caselle che verranno poi convertiti automaticamente in intervalli
- Menù a tendina per la scelta del metodo di induzione dell'albero





Apprendimento intelligente NBA

Età	<input type="text"/>
Altezza	<input type="text"/>
Peso	<input type="text"/>
Partite giocate	<input type="text"/>
Minuti a partita	<input type="text"/>
Impiego attivo (%)	<input type="text"/>
Ritmo di gioco	<input type="text"/>
Possessi per stagione	<input type="text"/>
Tiri tentati a partita	<input type="text"/>
Attacchi al canestro a partita	<input type="text"/>
Miglia percorse a partita	<input type="text"/>
Velocità media	<input type="text"/>
Tiri in sospensione a partita	<input type="text"/>
Frequenza di palleggio	<input type="text"/>
Media dribbling per tocco	<input type="text"/>
Tocchi spalle a canestro	<input type="text"/>
Tocchi nel pitturato	<input type="text"/>

Metodo di induzione:

Entropia

Entropia

Gini

# INTERFACCIA

## Funzionalità:

- **Bottone costruisci query:** lancia l'apprendimento e l'induzione dell'albero con il criterio selezionato
- **Bottone potatura:** lancia la potatura e mostra la matrice di confusione






# CONCLUSIONI

**Dalle matrici di confusione è emerso che, per il nostro modello, utilizzando l'entropia si hanno risultati migliori in termini di accuratezza rispetto all'utilizzo di Gini.**

**Utilizzando la potatura si guadagna circa il 2% in termini di accuratezza in entrambi i casi (entropia e Gini).**



# **GRAZIE**

**di Giulio Vignati, Matteo Losa &  
Alessandro Pieragostini**

