# HEART DISEASE PREDICTION

## INDUZIONE DI REGOLE E ALBERI DECISIONALI CON PROLOG: GINI E ENTROPIA AL VARIARE DELLA GRANULARITÀ

Studente: Federico Beni (S1123835)
Docente: Aldo Franco Dragoni
Corso: Intelligenza Artificiale A.A. 2025/2026

# DATASET

Il dataset in questione si chiama *UCI Heart Disease Data*;

- Presenta un totale di **920 record** e di **14 attributi**.
- L'obiettivo è determinare se un paziente è malato o meno osservando le diagnosi relative ai sintomi, di fatti la **variabile target** è **Num**, convertita a valore binario per una classificazione, appunto, binaria (*0=no; 1,2,3,4=yes*).
- Nella fase di **preprocessing**, tutti i valori mancanti ed i valori pari a 0 anomali sono stati sostituiti con **'unknown'.**

Per i successivi test effettuati, il dataset è stato suddiviso come segue:

- Training set: 644 record
- Test set: 276 record.

| Attributo | Valore |
|---|---|
| Age | *Valori compresi tra 28 e 77* |
| Sex | *male, female* |
| Cp | *typical angina, atypical angina, non-anginal, asymptomatic* |
| trestbps | *Valori da 80 a 200* |
| chol | *Valori da 85 a 603* |
| Fbs | *true, false* |
| Restecg | *normal, stt abnormality, lv ipetrophy* |
| Thalach | *Valori da 60 a 202* |
| Exang | *true, false* |
| Oldpeak | *Valori da -26 a 62* |
| Slope | *upsloping, flat, downsloping* |
| Ca | *0, 1, 2, 3* |
| Thal | *normal, fixed defect, reversible defect* |
| Num | *0, 1, 2, 3, 4* |

# SCRIPT UTILIZZATI

- discretizza1/2/3.pl e discretizza4.py
- crea_heart_dataset.pl
- heart_RulesInduction.pl
- heart_Gini_TreeInduction.pl
- heart_Entropia_TreeInduction.pl

# BASELINE: APPROCCIO SENZA DISCRETIZZAZIONE

```
        49 ==> ???
        50 ==> ???
        51 ==> ???
        52 ==> ???
        53 ==> ???
        54 ==> ???
        55 ==> ???
        56 ==> ???
        57 ==> yes
        58 ==> ???
        59 ==> ???
        60 ==> ???
        61 ==> ???
        62 ==> ???
        63 ==> ???
        64 ==> ???
        65 ==> ???
        66 ==> ???
        67 ==> ???
        68 ==> ???
        69 ==> ???
        70 ==> ???
        71 ==> ???
        72 ==> ???
        73 ==> ???
        74 ==> ???
        75 ==> ???
        76 ==> ???
        77 ==> ???
149 ==> yes
150 ==> yes
151 ==> ???
152 ==> ???
153 ==> ???
154
  age
        28 ==> ???
        29 ==> ???
```

Albero generato con Gini.

```
                    192 ==> ???
                    200 ==> ???
                    unknown ==> ???
                1 ==> ???
                2 ==> yes
                3 ==> ???
                4 ==> ???
                5 ==> ???
                6 ==> ???
                7 ==> ???
                8 ==> ???
                9 ==> ???
                11 ==> ???
                12 ==> ???
                13 ==> ???
                14 ==> ???
                15 ==> ???
                16 ==> ???
                17 ==> ???
                18 ==> ???
                19 ==> ???
                21 ==> ???
                22 ==> ???
                23 ==> ???
                24 ==> ???
                25 ==> ???
                26 ==> ???
                28 ==> ???
                29 ==> ???
                31 ==> ???
                32 ==> ???
                34 ==> ???
                35 ==> ???
                36 ==> ???
                37 ==> ???
                38 ==> ???
                42 ==> ???
                44 ==> ???
                56 ==> ???
                62 ==> ???
                unknown ==> ???
            upsloping ==> yes
atypical angina ==> yes
```

Albero generato con Entropia.

```
Albero = t(chol, [85:l(no), 100:t(age, [28:null, 29:null, 30:nu
ll, ... : ...|...]), 117:l(yes), 126:l(no), 129:l(no), 131:null
, 132:l(...), ... : ...|...]) .
```

Albero generato con Gini (struttura compatta).

```
Albero = t(fbs, [false:t(sex, [female:t(exang, [false:t(..., ..
.), ... : ...]), male:t(exang, [... : ...|...])]), true:t(sex,
[female:t(exang, [... : ...|...]), male:t(exang, [...|...])])])
 .
```

Albero generato con Entropia (struttura compatta).

```
?- stampa_matrice_di_confusione.
Test effettuati :272
Test non classificati :147
Veri Negativi  29   Falsi Positivi 19
Falsi Negativi 28   Veri Positivi  49
Accuratezza: 0.624
Errore: 0.376
true
```

Matrice di confusione generata con Gini.

```
?- stampa_matrice_di_confusione.
Test effettuati :272
Test non classificati :83
Veri Negativi  72   Falsi Positivi 18
Falsi Negativi 17   Veri Positivi  82
Accuratezza: 0.8148148148148148
Errore: 0.18518518518518523
true .
```

Matrice di confusione generata con Entropia.

- L'albero ad Entropia riduce drasticamente i casi non classificati (83) rispetto a Gini (147), dimostrando una miglior capacità di **copertura** del Test Set nella Baseline.
- Già dalla baseline l'Entropia raggiunge una buona accuratezza a differenza di Gini

# BASELINE: APPROCCIO SENZA DISCRETIZZAZIONE

```
?- apprendi(yes).

yes<==
   [oldpeak=25]
   [trestbps=95]
   [trestbps=144]
   [chol=282]
   [thalach=108]
   [trestbps=115,sex=male]
   [trestbps=136]
   [trestbps=200]
   [thalach=119]
   [oldpeak=28]
   [thalach=105,ca=unknown]
   [thalach=102]
   [thalach=103]
   [thalach=117]
   [thalach=123]
```

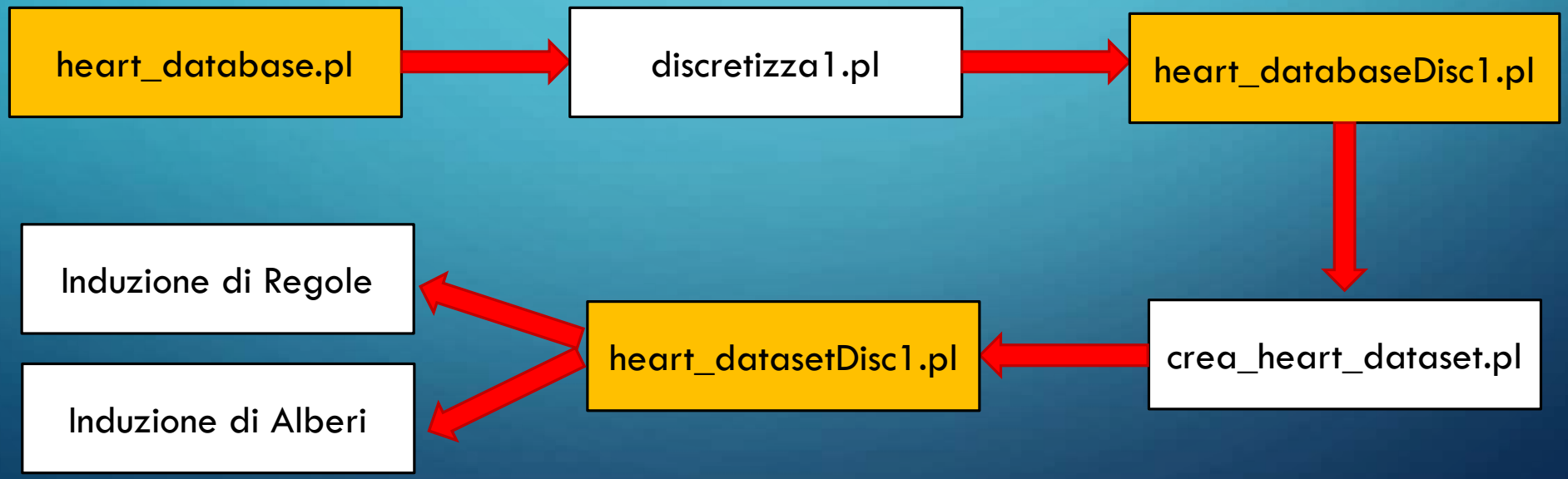
```
?- apprendi(no).

no<==
   [thalach=172]
   [cp='atypical angina',slope=unknown,exang=false]
   [chol=240]
   [chol=220]
   [chol=308]
   [thalach=178]
   [thalach=179]
   [chol=182]
   [chol=250]
   [chol=271]
   [age=45,sex=female]
   [trestbps=104]
```

```
?- stampa_matrice_di_confusione.
Test effettuati :272
Test non classificati :94
Veri Negativi  58   Falsi Positivi 21
Falsi Negativi 17   Veri Positivi  82
Accuratezza: 0.7865168539325843
Errore: 0.2134831460674157
true.
```

# DISCRETIZZAZIONE: PIPELINE DI LAVORO

La pipeline di lavoro eseguita per ogni discretizzazione è la seguente:



# DISCRETIZZAZIONE1

paziente(63,male,'typical angina',145,233,true,'lv hypertrophy',150,false,23,downsloping,0,'fixed defect',0).

discretizza1.pl

pd(old,male,'typical angina',high,high,true,'lv hypertrophy',high,false,depression,downsloping,0,'fixed defect',no.)

crea_heart_dataset.pl

pd(no,[age=old, sex=male, cp='typical angina', trestbps=high, chol=high, fbs=true, restecg='lv hypertrophy', thalach=high, exang=false, oldpeak=depression, slope=downsloping, ca=0, thal='fixed defect']).

# DISCRETIZZAZIONE1

## discretizza1.pl

```
:- ensure_loaded('heart_database.pl').

start :-
    tell('heart_databaseDisc1.pl'), % Apre il file di output
    elabora_fatti,                  % Avvia il ciclo di elaborazione
    told.                           % Chiude il file

elabora_fatti :-
    % Legge un fatto dal database grezzo
    paziente(Age, Sex, Cp, Trestbps, Chol, Fbs, Restecg, Thalach, Exang, Oldpeak, Slope, Ca, Thal, Num),

    %DISCRETIZZAZIONE DEGLI ATTRIBUTI

    % 1. Età : young (<35), adult (35-60), old (>60)
    discretizza_eta(Age, AgeGroup),

    % 2. Pressione: optimal (<120), normal (120-139), high (>=140)
    discretizza_bp(Trestbps, BP_Level),

    % 3. Colesterolo: normal (<200), high (200-239), very_high (>=240)
    discretizza_chol(Chol, Chol_Level),

    % 4. Battito Cardiaco: low (<150), high (>=150)
    discretizza_talach(Thalach, Talach_Level),

    % 5. Depressione ST (Oldpeak): no_depression (0), depression (>0)
    discretizza_oldpeak(Oldpeak, Depression),

    % 6. Target (Num): 0 -> no (Sano), >0 -> yes (Malato)
    discretizza_target(Num, Target),

    % Gli attributi categorici (Sex, Cp, Fbs, Restecg, Exang, Slope, Ca, Thal)
    % vengono passati direttamente o gestiti solo per 'unknown'.
```

```
    %SCRITTURA DEL FATTO ELABORATO
    write('pd('),
    writeq(AgeGroup), write(','),
    writeq(Sex), write(','),
    writeq(Cp), write(','),
    writeq(BP_Level), write(','),
    writeq(Chol_Level), write(','),
    writeq(Fbs), write(','),
    writeq(Restecg), write(','),
    writeq(Talach_Level), write(','),
    writeq(Exang), write(','),
    writeq(Depression), write(','),
    writeq(Slope), write(','),
    writeq(Ca), write(','),
    writeq(Thal), write(','),
    writeq(Target), writeln(').'),

    fail. % Forza il backtracking per trovare il prossimo paziente
elabora_fatti. % Termina quando non ci sono più fatti

%REGOLE DI DISCRETIZZAZIONE (inclusa gestione 'unknown')

% ETA
discretizza_eta('unknown', 'unknown') :- !.
discretizza_eta(A, 'young') :- A < 35, !.
discretizza_eta(A, 'adult') :- A =< 60, !.
discretizza_eta(_, 'old').

% PRESSIONE (Trestbps)
discretizza_bp('unknown', 'unknown') :- !.
discretizza_bp(P, 'optimal') :- P < 120, !.
discretizza_bp(P, 'normal')  :- P =< 139, !.
discretizza_bp(_, 'high').

% COLESTEROLO (Chol)
discretizza_chol('unknown', 'unknown') :- !.
discretizza_chol(C, 'normal')    :- C < 200, !.
discretizza_chol(C, 'high')      :- C =< 239, !.
discretizza_chol(_, 'very_high').
```

# DISCRETIZZAZIONE1

crea_heart_dataset.pl

```
:- ensure_loaded('heart databaseDisc1.pl').

start :-
    tell('heart_datasetDisc1.pl'),

    %GENERAZIONE DOMINI (METADATI)

    % Per ogni attributo, uso setof per trovare tutti i valori unici presenti nel DB.
    % La sintassi A^B^... serve a dire "ignorando le altre variabili".

    % 1. Age (Eta)
    setof(Age, Sex^Cp^Bp^Chol^Fbs^Ecg^Hr^Ex^Dep^Slp^Ca^Thal^Tar^pd(Age,Sex,Cp,Bp,Chol,Fbs,Ecg,Hr,Ex,Dep,Slp,Ca,Thal,Tar), DomAge),
    write('a(age,'), writeq(DomAge), writeln(').'),

    % 2. Sex
    setof(Sex, Age^Cp^Bp^Chol^Fbs^Ecg^Hr^Ex^Dep^Slp^Ca^Thal^Tar^pd(Age,Sex,Cp,Bp,Chol,Fbs,Ecg,Hr,Ex,Dep,Slp,Ca,Thal,Tar), DomSex),
    write('a(sex,'), writeq(DomSex), writeln(').'),

    % 3. Cp (Chest Pain)
    setof(Cp, Age^Sex^Bp^Chol^Fbs^Ecg^Hr^Ex^Dep^Slp^Ca^Thal^Tar^pd(Age,Sex,Cp,Bp,Chol,Fbs,Ecg,Hr,Ex,Dep,Slp,Ca,Thal,Tar), DomCp),
    write('a(cp,'), writeq(DomCp), writeln(').'),

    % 4. Trestbps (Pressione)
    setof(Bp, Age^Sex^Cp^Chol^Fbs^Ecg^Hr^Ex^Dep^Slp^Ca^Thal^Tar^pd(Age,Sex,Cp,Bp,Chol,Fbs,Ecg,Hr,Ex,Dep,Slp,Ca,Thal,Tar), DomBp),
    write('a(trestbps,'), writeq(DomBp), writeln(').'),
```

# DISCRETIZZAZIONE1

crea_heart_dataset.pl

```
scrivi_esempi :-
    % CASO POSITIVO (MALATO - 'yes')
    pd(Age,Sex,Cp,Bp,Chol,Fbs,Ecg,Hr,Ex,Dep,Slp,Ca,Thal, 'yes'),
    write('e(yes,['),
    write('age='), writeq(Age), write(', '),
    write('sex='), writeq(Sex), write(', '),
    write('cp='), writeq(Cp), write(', '),
    write('trestbps='), writeq(Bp), write(', '),
    write('chol='), writeq(Chol), write(', '),
    write('fbs='), writeq(Fbs), write(', '),
    write('restecg='), writeq(Ecg), write(', '),
    write('thalach='), writeq(Hr), write(', '),
    write('exang='), writeq(Ex), write(', '),
    write('oldpeak='), writeq(Dep), write(', '),
    write('slope='), writeq(Slp), write(', '),
    write('ca='), writeq(Ca), write(', '),
    write('thal='), writeq(Thal), writeln(']).'),
    fail.

scrivi_esempi :-
    % CASO NEGATIVO (SANO - 'no')
    pd(Age,Sex,Cp,Bp,Chol,Fbs,Ecg,Hr,Ex,Dep,Slp,Ca,Thal, 'no'),
    write('e(no,['),
    write('age='), writeq(Age), write(', '),
    write('sex='), writeq(Sex), write(', '),
    write('cp='), writeq(Cp), write(', '),
    write('trestbps='), writeq(Bp), write(', '),
    write('chol='), writeq(Chol), write(', '),
    write('fbs='), writeq(Fbs), write(', '),
    write('restecg='), writeq(Ecg), write(', '),
    write('thalach='), writeq(Hr), write(', '),
    write('exang='), writeq(Ex), write(', '),
    write('oldpeak='), writeq(Dep), write(', '),
    write('slope='), writeq(Slp), write(', '),
    write('ca='), writeq(Ca), write(', '),
    write('thal='), writeq(Thal), writeln(']).'),
    fail.

scrivi_esempi :- told.
```

# DISCRETIZZAZIONE1

La prima discretizzazione prevede la suddivisione nelle seguenti fasce di valori:

- **Età (Age):** 3 fasce – *Young* (<35), *Adult* (35-60), *Old* (>60).
- **Pressione (Trestbps):** 3 livelli – *Optimal* (<120), *Normal* (120-139), *High* (≥140).
- **Colesterolo (Chol):** 3 categorie – *Normal* (<200), *High* (200-239), *Very High* (≥240).
- **Frequenza Cardiaca (Thalach):** Binarizzazione – *Low* (<150), *High* (≥150).
- **Depressione ST (Oldpeak):** Binarizzazione– *No Depression* (0), *Depression* (>0).
- **Target (Num):** Classificazione binaria – *No* (0), *Yes* (>0).

# DISCRETIZZAZIONE1

```
ca
  0 ==> yes
  1 ==> yes
  2 ==> ???
  3 ==> yes
  unknown
    thal
      fixed defect ==> ???
      normal ==> ???
      reversable defect ==> yes
      unknown
        cp
          asymptomatic ==> yes
          atypical angina ==> yes
          non-anginal ==> no
          typical angina ==> ???
unknown ==> ???
no_depression
  cp
    asymptomatic
      ca
        0 ==> ???
        1 ==> ???
        2 ==> yes
        3 ==> ???
        unknown
          thalach
            high ==> ???
            low
              chol
                high ==> yes
```

```
                                        normal
                                          slope
                                            downsloping ==> ???
                                            flat ==> ???
                                            unknown ==> ???
                                            upsloping ==> [no,yes]
                                        st-t abnormality ==> ???
                                    unknown ==> ???
                                true ==> ???
                            true ==> ???
                    old ==> ???
                    young ==> ???
            1 ==> ???
            2 ==> ???
            3 ==> ???
            unknown ==> no
    unknown ==> ???
reversable defect
  chol
    high
      oldpeak
        depression ==> yes
        no_depression ==> no
        unknown ==> ???
    normal ==> no
    unknown ==> ???
    very_high ==> no
unknown
  slope
    downsloping ==> ???
    flat
```

```
?- apprendi(yes).
false.

?- apprendi(no).
false.
```

```
?- stampa_matrice_di_confusione.
false.
```

(sinistra Gini, destra Entropia)

- Presenza di **foglie impure** [no, yes]: l'ambiguità dei dati impedisce una classificazione univoca, causando il fallimento logico nel calcolo delle matrici e dell'apprendimento delle regole

# DISCRETIZZAZIONE2

- **Età (Age):** *5* fasce – *Young* (<35), *Early Adult* (35-48), *Late Adult* (49-60), *Senior* (61-72), *Elderly* (>72).
- **Pressione (Trestbps):** *5* livelli – *Optimal* (<120), *Normal* (120-129), *High Normal* (130-139), *Grade 1 Hyp* (140-159), *Grade 2 Hyp* (≥160).
- **Colesterolo (Chol):** *5* categorie – *Desirable* (<180), *Borderline* (180-219), *High* (220-259), *Very High* (260-299), *Extreme* (≥300).
- **Frequenza Cardiaca (Thalach):** 4 livelli – *Very Low* (<90), *Low* (90-129), *Moderate* (130-169), *High* (≥170).
- **Depressione ST (Oldpeak):** 3 fasce – *None* (0), *Mild* (0.1-2.0), *Severe* (>2.0).

I risultati ottenuti dal secondo tentativo risultano identici al primo, senza alcun tipo di miglioramento; l'incremento di così poche fasce di valori mantiene i dati ancora troppo generici e la granularità ancora troppo bassa per gli algoritmi.

# DISCRETIZZAZIONE3

- **Età (Age):** 10 fasce da 5 anni ciascuna (es. *age1* 25-29, ..., *age10* ≥70).
- **Pressione (Trestbps):** 10 fasce da 12 mmHg (es. *bp1* 80-91, ..., *bp10* ≥188).
- **Colesterolo (Chol):** 10 fasce da 50 mg/dl (es. *ch1* 85-134, ..., *ch10* ≥535).
- **Frequenza Cardiaca (Thalach):** 10 fasce da 15 bpm (es. *talach1* 60-74, ..., *talach10* ≥195).
- **Depressione ST (Oldpeak):** 10 fasce da 10 unità (da -30 a +60).

# DISCRETIZZAZIONE3

```
            op10 ==> ???
            op2 ==> ???
            op3 ==> no
            op4 ==> ???
            op5 ==> no
            op6 ==> yes
            op7 ==> ???
            op8 ==> ???
            op9 ==> ???
            unknown ==> ???
typical angina
  chol
    ch1 ==> ???
    ch10 ==> ???
    ch2 ==> no
    ch3
      trestbps
        bp1 ==> ???
        bp10 ==> ???
        bp2 ==> no
        bp3 ==> no
        bp4 ==> no
        bp5 ==> yes
        bp6 ==> no
        bp7 ==> no
        bp8 ==> no
        bp9 ==> ???
        unknown ==> yes
    ch4
      thal
        fixed defect ==> no
        normal ==> no
        reversable defect ==> yes
```

Albero generato con Gini.

```
                    ch7 ==> ???
                    ch8 ==> ???
                    ch9 ==> ???
                    unknown ==> yes
                age2 ==> ???
                age3 ==> ???
                age4 ==> ???
                age5 ==> ???
                age6 ==> ???
                age7 ==> yes
                age8 ==> ???
                age9 ==> ???
            st-t abnormality ==> ???
            unknown ==> ???
        unknown
          slope
            downsloping ==> yes
            flat ==> yes
            unknown ==> no
            upsloping ==> ???
    op5 ==> yes
    op6 ==> ???
    op7 ==> yes
    op8 ==> ???
    op9 ==> ???
    unknown ==> ???
atypical angina ==> yes
non-anginal
  chol
    ch1 ==> ???
    ch10 ==> ???
    ch2 ==> ???
    ch3 ==> no
    ch4
      oldpeak
        op1 ==> ???
        op10 ==> ???
        op2 ==> ???
        op3 ==> ???
```

Albero generato con Entropia.

```
Albero = t(cp, [asymptomatic:t(ca, [0:t(thal, ['fixed d
efect':t(..., ...), ... : ...|...]), 1:l(yes), 2:t(thal
, [...|...]), 3:t(..., ...), ... : ...]), 'atypical ang
ina':t(slope, [downsloping:l(no), flat:t(chol, [...|...
]), unknown:t(..., ...), ... : ...]), 'non-anginal':t(a
ge, [age1:null, age10:t(..., ...), ... : ...|...]), 'ty
pical angina':t(chol, [ch1:null, ... : ...|...])]) .
```

Albero generato con Gini (struttura compatta)

```
Albero = t(fbs, [false:t(sex, [female:t(exang, [false:t
(..., ...), ... : ...]), male:t(exang, [... : ...|...])
]), true:t(sex, [female:t(oldpeak, [... : ...|...]), ma
le:t(exang, [...|...])])]) .
```

Albero generato con Entropia (struttura compatta).

# DISCRETIZZAZIONE3

```
?- stampa_matrice_di_confusione.
Test effettuati :276
Test non classificati :32
Veri Negativi  88   Falsi Positivi 35
Falsi Negativi 21   Veri Positivi  100
Accuratezza: 0.7704918032786885
Errore: 0.2295081967213115
```

Matrice di confusione generata con Gini.

```
?- stampa_matrice_di_confusione.
Test effettuati :276
Test non classificati :76
Veri Negativi  79   Falsi Positivi 25
Falsi Negativi 22   Veri Positivi  74
Accuratezza: 0.765
Errore: 0.235
```

Matrice di confusione generata con Entropia.

```
?- apprendi(yes).
false.

?- apprendi(no).
false.
```

- Netto miglioramenti per l'albero con Gini, sia per i non classificati che per l'accuratezza.
- L'Entropia riesce a diminuire i casi non classificati ma ottiene un'accuratezza minore rispetto alla baseline.
- Tuttavia il comando *apprendi* continua a fallire probabilmente per la presenza di ambiguità residue che impediscono una separazione netta tra le classi.

# DISCRETIZZA4

## discretizza4.py

```python
import re

# Funzione di discretizzazione base
def discretizza(valore, v_min, v_max, num_fasce=50):
    if valore == 'unknown': return 'unknown'              # Ritorna subito se unknown
    try:
        valore = float(valore)                            # Conversione a numero
        ampiezza = (v_max - v_min) / num_fasce            # Calcolo ampiezza fascia
        if valore <= v_min: return 1                      # Limite inferiore
        if valore >= v_max: return num_fasce              # Limite superiore
        return int((valore - v_min) // ampiezza) + 1      # Calcolo indice (1-50)
    except:
        return 'unknown'                                  # Fallback su errore

# Helper per gestire prefissi e unknown
def get_val(prefisso, val_raw, range_key):
    res = discretizza(val_raw, *ranges[range_key])        # Chiama discretizzazione
    return 'unknown' if res == 'unknown' else f"{prefisso}{res}"  #g gestione nknown

# Configurazione range (Min, Max)
ranges = {
    'age': (28, 77), 'trestbps': (80, 200),
    'chol': (85, 603), 'talach': (60, 202),
    'oldpeak': (-26, 62)
}

input_file = 'heart_database.pl'
output_file = 'heart_databaseDisc4.pl'
```

```python
with open(input_file, 'r') as f, open(output_file, 'w') as out:
    for riga in f:
        if not riga.startswith('paziente'): continue     # Filtra i fatti

        contenuto = re.search(r'\((.*)\)', riga).group(1)
        v = [x.strip().strip("'") for x in contenuto.split(',')] # Pulisce i dati

        # 1. Discretizzazione con controllo Unknown
        age_d = get_val("age", v[0], 'age')              # Fascia Age
        bp_d  = get_val("bp",  v[3], 'trestbps')         # Fascia Pressione
        ch_d  = get_val("ch",  v[4], 'chol')             # Fascia Colesterolo
        hr_d  = get_val("talach", v[7], 'talach')        # Fascia Freq. Cardiaca
        op_d  = get_val("op",  v[9], 'oldpeak')          # Fascia Oldpeak

        # 2. Conversione Target (0 -> no, 1-4 -> yes)
        target_final = 'no' if v[13].strip('.') == '0' else 'yes'

        # 3. Scrittura riga pd/14
        nuova_riga = (f"pd({age_d},{v[1]},'{v[2]}',{bp_d},{ch_d},{v[5]},"
                      f"'{v[6]}',{hr_d},{v[8]},{op_d},{v[10]},{v[11]},"
                      f"'{v[12]}',{target_final}).\n")

        out.write(nuova_riga)                            # Scrive su file

print(f"File {output_file} generato.")
```

# DISCRETIZZAZIONE4

```
?- apprendi(yes).

yes<==
   [thalach=talach12]
   [ca=3,cp=asymptomatic]
   [thalach=talach19,slope=flat]
   [trestbps=bp7]
   [thalach=talach16,ca=unknown]
   [oldpeak=op29,fbs=false]
   [oldpeak=op31]
   [trestbps=bp24]
   [thalach=talach14,restecg=normal]
   [trestbps=bp27]
   [oldpeak=op36]
   [thalach=talach17,exang=true]
   [thalach=talach23,oldpeak=op16]
   [ca=2,slope=flat]
   [age=age45]
   [trestbps=bp33]
   [trestbps=bp50]
   [thalach=talach5]
   [oldpeak=op27]
   [oldpeak=op33]
```

```
?- apprendi(no).

no<==
   [age=age10]
   [thalach=talach45]
   [age=age15,exang=false]
   [age=age18,exang=false]
   [cp='atypical angina',chol=ch12]
   [thalach=talach42,restecg=normal]
   [cp='atypical angina',slope=unknown,sex=female]
   [thalach=talach40,restecg=normal]
   [thalach=talach44]
   [ca=0,trestbps=bp21]
   [cp='atypical angina',age=age28]
   [age=age8,exang=false,sex=female]
   [thalach=talach33,cp='non-anginal']
   [cp='atypical angina',slope=unknown,restecg=normal,trestbps=bp17]
   [ca=0,thalach=talach37]
   [age=age12,thal=unknown]
   [age=age3]
   [trestbps=bp1]
   [trestbps=bp12,exang=false]
   [trestbps=bp8]
```

```
?- stampa_matrice_di_confusione.
Test effettuati :276
Test non classificati :70
Veri Negativi  80   Falsi Positivi 16
Falsi Negativi 15   Veri Positivi  95
Accuratezza: 0.8495145631067961
Errore: 0.15048543689320393
```

- Con 50 fasce di valori l'algoritmo riesce ad apprendere.
- Dalla matrice di confusione generata dall'induzione di regole, si nota, rispetto alla baseline, un incremento dell'accuratezza (dal 78% all'85%) e una diminuzione dei casi non classificati (da 94 a 70).

# DISCRETIZZAZIONE4

```
        age40 ==> ???
        age41 ==> ???
        age42 ==> ???
        age43 ==> ???
        age44 ==> yes
        age45 ==> ???
        age46 ==> ???
        age47 ==> no
        age48 ==> ???
        age49 ==> ???
        age5 ==> ???
        age50 ==> ???
        age6 ==> ???
        age7 ==> ???
        age8 ==> no
        age9 ==> ???
typical angina
  age
    age1 ==> ???
    age10 ==> ???
    age11 ==> ???
    age12 ==> ???
    age13 ==> no
    age14 ==> ???
    age15 ==> ???
    age16
      chol
        ch1 ==> ???
        ch10 ==> ???
        ch11 ==> ???
        ch12 ==> ???
        ch13 ==> ???
        ch14 ==> no
        ch15 ==> ???
        ch16 ==> ???
        ch17 ==> ???
```

Albero generato con Gini.

```
        op36 ==> ???
        op37 ==> ???
        op39 ==> ???
        op40 ==> ???
        op47 ==> ???
        op50 ==> ???
        op7 ==> ???
        op9 ==> ???
        unknown ==> ???
unknown
  slope
    downsloping ==> yes
    flat ==> yes
    unknown
      oldpeak
        op1 ==> ???
        op10 ==> ???
        op11 ==> ???
        op12 ==> ???
        op14 ==> ???
        op15
          age
            age1 ==> ???
            age10 ==> ???
            age11 ==> ???
            age12 ==> ???
            age13 ==> ???
            age14 ==> ???
            age15 ==> ???
            age16 ==> ???
            age17 ==> ???
            age18 ==> ???
            age19 ==> ???
            age2 ==> ???
            age20 ==> no
            age21 ==> ???
            age22 ==> ???
            age23 ==> ???
            age24 ==> ???
```

Albero generato con Entropia.

```
Albero = t(cp, [asymptomatic:t(age, [age1:null, age10:l(no), age11:l(
yes), age12:t(..., ...), ... : ...|...]), 'atypical angina':t(age, [a
ge1:l(no), age10:l(no), age11:l(...), ... : ...|...]), 'non-anginal':
t(thalach, [talach1:null, talach10:l(...), ... : ...|...]), 'typical
angina':t(age, [age1:null, ... : ...|...])]) .
```

Albero generato con Gini (forma compatta).

```
Albero = t(fbs, [false:t(sex, [female:t(exang, [false:t(..., ...)
, ... : ...]), male:t(exang, [... : ...|...])]), true:t(sex, [fem
ale:t(exang, [... : ...|...]), male:t(exang, [...|...])])]) .
```

Albero generato con Entropia (forma compatta).

```
?- stampa_matrice_di_confusione.
Test effettuati :276
Test non classificati :111
Veri Negativi  61   Falsi Positivi 16
Falsi Negativi 18   Veri Positivi  70
Accuratezza: 0.793939393939394
Errore: 0.20606060606060606
```

Matrice di confusione generata con Gini.

```
?- stampa_matrice_di_confusione.
Test effettuati :276
Test non classificati :99
Veri Negativi  72   Falsi Positivi 23
Falsi Negativi 10   Veri Positivi  72
Accuratezza: 0.8135593220338984
Errore: 0.18644067796610164
```

Matrice di confusione generata con Entropia.

Dalle matrici di confusione si nota:

- Gini: non classificati diminuiti rispetto alla baseline, ma aumentati rispetto a Disc3; accuratezza aumentata in generale.
- Entropia: non classificati aumentati, sia rispetto alla baseline che alla Disc3, ma risalita del valore dell'accuratezza, che è praticamente la stessa della baseline.

# CONCLUSIONI E CONFRONTO FINALE

**Impatto della Granularità**

•**Bassa (Disc1/2): Underfitting**. Ambiguità e foglie impure [no, yes] bloccano l'apprendimento.

•**Media (Disc3): Massima Generalizzazione**. Punto di equilibrio con il minor numero di casi non classificati (NC).

•**Alta (Disc4): Massima Precisione**. L'accuratezza risale all'81%, ma aumenta la frammentazione (NC) rispetto a Disc3.

**Gini vs Entropia:**

•**Entropia:** Più robusta sui dati con granularità elevata (Baseline e Disc4, 84 e 99 NC). Tende a massimizzare l'accuratezza.

•**Gini:** Estremamente dipendente dal preprocessing; migliora nettamente e supera l'Entropia nei casi in cui i dati sono meno frammentati.

# GRAZIE PER L'ATTENZIONE!

Codice sorgente su GitHub