16. Classificatori e regressori

Corso di Python per il Calcolo Scientifico

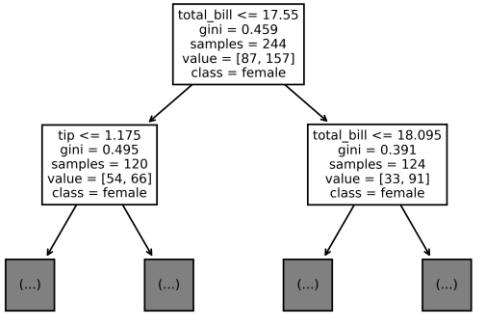
Angelo Cardellicchio
angelo.cardellicchio@stiima.cnr.it

Outline

- Alberi decisionali
  - Addestramento di un albero decisionale
  - Generalizzazione, overfitting e pruning
- Foreste decisionali
- Multilayer perceptron

Alberi decisionali

- Effettuano una predizione a partire da regole di tipo booleano
- Usano il **recursive partitioning**
- **Facili da interpretare e visualizzare**
- Soggetti ad **overfitting**; danno una predizione **lineare a tratti**
- In Scikit Learn sono implementati mediante le classi `DecisionTreeClassifier()` e `DecisionTreeRegressor()`



Addestramento di un albero decisionale

- Supponiamo di avere il seguente dataset.

Identificativo	Zampe (numero)	Occhi (numero)
Chihuahua	4	2
Rottweiller	4	2
Moroseta	2	2
Malmignatta	8	8
Ragno violino	8	6
Sussex	2	2

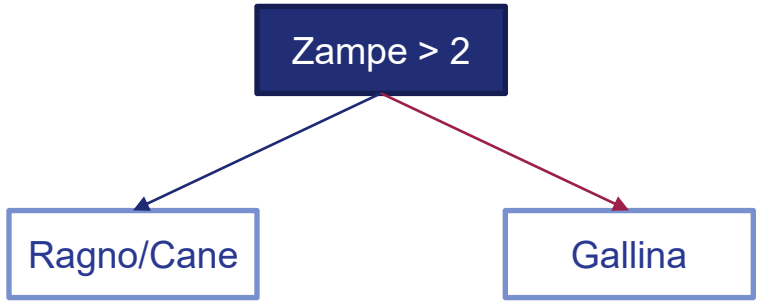
Addestramento di un albero decisionale

- **Step 1: scelta del nodo radice**
- Avviene sulla base di una **Attribute Selective Measure (ASM)**
- Esistono due ASM principali:
  - **Gini impurity:** misura la probabilità che ad un campione casuale sia classificato in modo sbagliato nel caso la classe sia assegnata in modo casuale seguendo la distribuzione di probabilità delle label nel dataset originario. L'obiettivo è scegliere l'attributo che **minimizza** questo indice.
  - **Information gain:** misura il contributo informativo di una determinata feature sulla base del concetto di entropia dell'informazione. L'obiettivo è scegliere l'attributo che **massimizza** questo indice.

Zampe

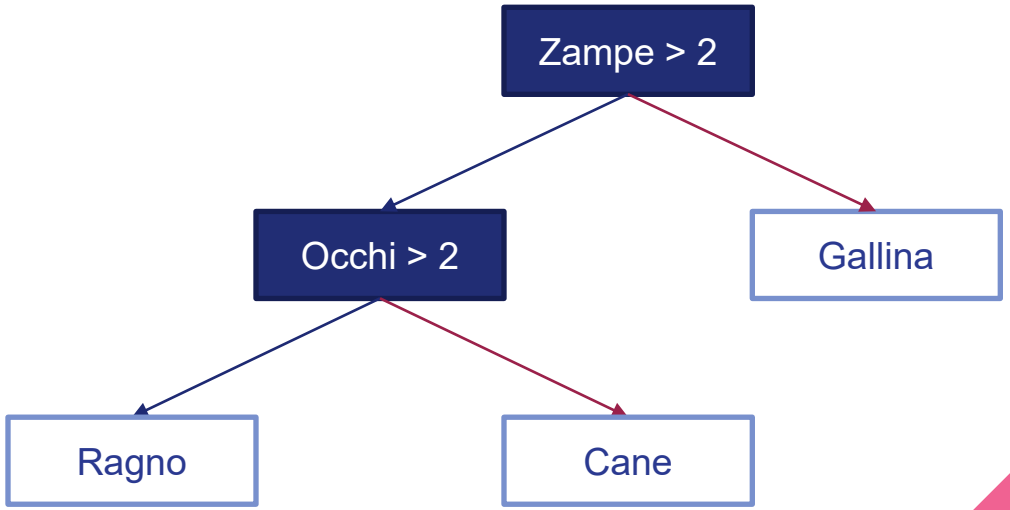
Addestramento di un albero decisionale

- **Step 2: accrescimento del nodo radice**
- Il nodo radice viene accresciuto sulla base di come suddivide i dati.
- In questo caso, se **Zampe** è maggiore di 2, abbiamo campioni appartenenti alle classi Ragno e Cane, mentre se è minore o uguale a due abbiamo solo campioni appartenenti alla classe Gallina.



Addestramento di un albero decisionale

- **Step 3: accrescimento del nodi foglia**
- Esploriamo per primo il nodo Gallina, cercando di suddividerlo ulteriormente sulla base delle informazioni in nostro possesso; dato che questo non è possibile, avremo un nodo foglia.
- Potremo invece suddividere ulteriormente il nodo Ragno/Cane utilizzando l'attributo Occhi.



Generalizzazione, overfitting e pruning

- Gli alberi decisionali sono soggetti ad **overfitting**
  - Ciò comporta problemi di **generalizzazione**
- Per ridurre l'incidenza di questo fenomeno, è possibile:
  - Inserire una **profondità massima**, in modo tale che l'albero non crei dei percorsi troppo lunghi.
  - Impostare un **numero minimo di campioni** per ciascuna foglia, in modo che non insorgano percorsi troppo specifici, con regole che portano ad un limitato numero di campioni.
  - Usare una procedura di **pruning**, convertendo alcuni nodi in foglie sulla base di criteri specifici.

Foreste decisionali

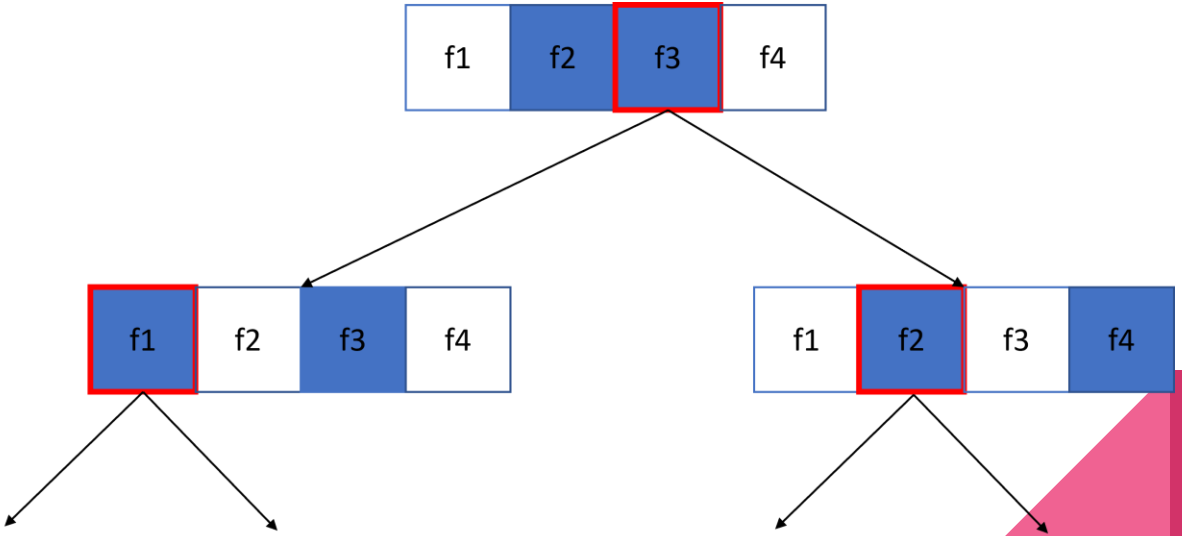
- I **metodi ensemble** combinano i risultati di un insieme (per l'appunto, *ensemble*) di diversi predittori.
  - Nel caso delle **foreste decisionali**, i singoli predittori sono degli alberi decisionali.
- Il tipo più conosciuto di foresta decisionale è chiamato **random forest**.
  - Un random forest inserisce delle componenti di casualità nell'addestramento di un insieme di alberi decisionale.
  - Questa componente casuale permette di aumentare le capacità di generalizzazione della foresta.
- In Scikit Learn sono implementati mediante le classi `RandomForestClassifier()` e `RandomForestRegressor()`

Addestramento di una foresta decisionale

- Esistono due metodi per addestrare una foresta decisionale.
- Nel **bagging**, ogni albero della foresta è addestrato su un sottoinsieme casuale dei dati presenti nel dataset.
  - Viene usata una tecnica chiamata **replacement training**, che prevede che ogni albero sia addestrato su un insieme di dati numericamente pari a quelli del dataset iniziale, ma selezionando soltanto il 67% dei possibili valori.
- Nell'**attribute sampling**, ogni albero viene addestrato selezionando, ad ogni espansione, un sottoinsieme casuale di feature.

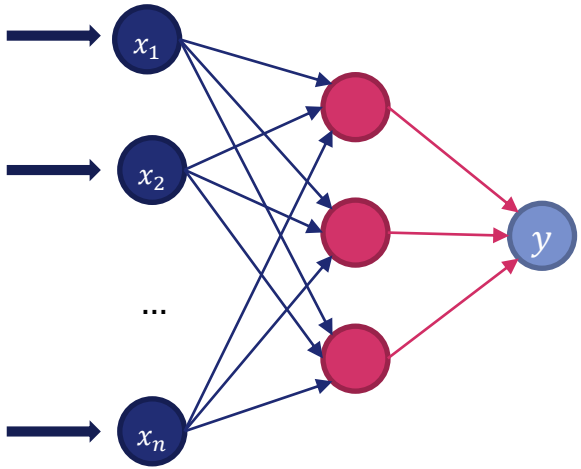
Addestramento di una foresta decisionale

- Se F è il numero totale di feature, sono testate $\frac{F}{3}$ feature in caso di regressione e $\sqrt{F}$ in caso di classificazione.



Multilayer perceptron

- Il **multilayer perceptron** è tra i più semplici modelli di rete neurale esistenti
- Mappa un input ad m feature su un output ad o dimensioni
- È composto da diversi neuroni, ognuno dei quali è un **sommatore lineare** seguito da una **funzione di attivazione non lineare**
- In Scikit Learn sono implementati mediante le classi `MLPClassifier()` e `MLPRegressor()`



Domande?

42