# Progetto 6: I mammiferi depongono uova?

Il dataset Zoo fornisce una serie di dati relativi a diverse specie animali al fine di classificarle in 7 diverse classi, ovvero:

1. mammiferi
2. uccelli
3. rettili
4. pesci
5. anfibi
6. insetti
7. invertebrati

Seguendo un approccio non supervisionato, ovvero senza osservare la classe di ogni specie animale, il progetto mira a confrontare le diverse specie e raggrupparle utilizzando diversi algoritmi di clustering. Confrontando poi il risultato di ogni algoritmo, si intende mostrare quale algoritmo di clustering approssimi meglio le classi fornite dal dataset. Si richiede pertanto non solo di definire una metodologia per confrontare i risultati del clustering con la classificazione attesa, ma anche di descrivere in modo sintetico le caratteristiche distintive di ciascun cluster di specie prodotto dall'algoritmo oggetto della valutazione.

# Esplorazione del dataset

Procediamo con l’importazione del dataset (salvato in formato .csv) importandolo sotto forma di dataframe per poi successivamente analizzare per ogni feature:

* il tipo (category, float, etc)
* presenza di dati mancanti
* valori presenti nel dataset per ogni features
* La correlazione tra le features e ogni singola classe (specie) target

Nella cartella dataset ci sono:

* zoo.data: dataset contenente in ogni riga la descrizione di ogni animale come 18 attributi espressi sotto forma di colonne e nella colonna “type” la specie di appartenenza.

Sono per facilità le classi “originali” sono state ridotte di uno nella loro corrispondenza numerica così da poter far partire la numerazione da 0 come nel linguaggio di programmazione python

* class.csv: il dataset delle classi o file dei risultati, contiene al suo interno la suddivisione nelle 7 diverse classi dei vari elementi del dataset "zoo.data". Di ogni classe espressa sotto forma di riga sono riportate le seguenti informazioni:

- Class\_Number

- Number\_Of\_Animal\_Species\_In\_Class

- Class\_Type

-Animal\_Names

Text

Description automatically generated with medium confidence

Graphical user interface, text, application

Description automatically generated

Il dataset è composto dalle seguenti features:

* animal name: unico per ogni elemento
* hair: Boolean
* feathers: Boolean
* eggs: Boolean
* milk: Boolean
* airborne: Boolean
* aquatic: Boolean
* predator: Boolean
* toothed: Boolean
* backbone: Boolean
* breathes: Boolean
* venomous: Boolean
* fins: Boolean
* legs: Numeric (insieme di valori: {0,2,4,5,6,8})
* tail: Boolean
* domestic: Boolean
* catsize: Boolean
* type: Numeric (variabile obbiettivo, intero nel range [0,6])

Procediamo quindi alla verifica di eventuali valori mancanti all’interno del dataset e del numero di possibili valori che può assumere ogni variabile:

Graphical user interface, text

Description automatically generated



Procediamo quindi a costruire manualmente (in italiano) il dizionario che ci permetterà di identificare ogni specie tramite il suo numero di riferimento [0,6]

species\_dict =dict(list(enumerate(['mammiferi', 'uccelli', 'rettili', 'pesci', 'anfibi', 'insetti','invertebrati'])))

print(species\_dict)

# Analisi del Dataset

Per capire se ci sono feature che presentano una maggiore correlazione con alcune specie costruiamo una matrice di correlazione e la grafichiamo tramite heatmap (che omettiamo in questo pdf ma è presente nel file main.jpyter) e barplot.

A screenshot of a computer

Description automatically generated with low confidence

Possiamo osservare come alcune features abbiano una elevata correlazione con determinate classi:

## 0. mammiferi

La classe presenta la maggiore correlazione per le feature:

- milk (Latte)

- hair (pelo)

- eggs (uova)

Tutti i mammiferi sono provvisti di ghiandole mammarie, di peli (nei mammiferi marini essi sono solo accennati e solo durante lo sviluppo embrionale) e solo due specie tra loro depongono uova (ornitorinco e echidna)

Chart, bar chart

Description automatically generated

## 1. Uccelli

La classe presenta la maggiore correlazione per le feature:

- feathers (piume)

- airbone (volante)

- toothed (dentati)

La maggiorparte degli uccelli hanno piume, volano e presentano una dentatura più o meno marcata nel becco

Chart, bar chart

Description automatically generated

## 3. Pesci

La classe presenta la maggiore correlazione per le feature:

- fins (pinne)

- bresthes (respira)

- legs (gambe)

I pesci sono un gruppo eterogeneo di organismi vertebrati fondamentalmente acquatici, dotati di pinne, che respirano attraverso le branchie.

Chart, bar chart

Description automatically generated

## 5. Insetti

La classe insetti presenta la maggiore correlazione per le feature

- backbone (spina dorsale)

- tail (coda)

- legs (gambe)

Essa è la classe più ampia e popolosa nel mondo reale, ritroviamo tale correlazione con alcune caratteristiche descrittive della specie in quanto essi hanno uno scheletro esterno (esoscheletro), più zampe e molti di loro presentano un allungamento dell'esoscheletro descrivibile come coda

Chart, histogram

Description automatically generated

## 6. Invertebrati

La classe presenta la maggiore correlazione per le feature:

- backbone (spina dorsale)

- tail (coda)

- toothed (dentati)

Gli animali invertebrati sono gli animali privi di colonna vertebrale e di scheletro interno, molti di loro presentano un allungamento dell'esoscheletro descrivibile come coda e non presentano denti

Chart, histogram

Description automatically generated

Le altre due classi non presentano una spiccata correlazione tra le features, esso può essere dovuto al basso numero di campioni

Chart, histogram

Description automatically generated

# Dataset Composition

Valutiamo ora la composizione del dataset espressa come:

- composizione percentuale delle varie specie

- numero campioni per ogni specie

Questi due numeri sono molto vicini in quanto abbiamo un dataset composto da 101 elementi

Chart

Description automatically generated

Per ogni classe osserviamo anche la distribuzione delle varie feature

A picture containing text, stationary

Description automatically generated

A picture containing text, stationary

Description automatically generated

# Evaluation Metrics for Clustering Algorithms

Al fine di poter valutare e confrontare i vari modelli di clustering useremo differenti indici di performance.

## Confusion Matrix

Il confronto tra la classi a cui appartengono realmente i singoli elementi del dataset e quella loro attribuita dall'algoritmo, la cui label è assegnata tramite majority voting, può essere riassunto tramite una matrice di confondimento:

- essa ci mostra per ogni item la sua classe originale e la classe assegnata dal modello

- sulla diagonale principale troviamo gli elementi "clusterizzati" correttamente

Faremo uso di questa matrice per riassumere in modo grafico insieme allo scatterplot (specie originale vs specie assegnata) il comportamento dell’algorimo di clusterizzazione

## Rand index (RI)

Statistica che quantifica la tendenza al raggruppamento di un certo dataset a valle di una procedura di clustering, esso si basa sui principi:

- Ogni elemento appartiene ad uno ed un solo cluster

- Ogni cluster è definito sia dagli elementi che contiene che da quelli che non contiene

- ogni elemento ha la stessa importanza nel definire la qualità di una partizione (pari peso tra gli elementi)

Quindi se una coppia di elementi è inclusa in uno stesso cluster da due distinte clusterizzazioni (label originali e label assegnate dall'algoritmo di clustering) vuol dire che c'è similarità altrimenti se la coppia viene "mappata" in due cluster differenti c'è dissimilarità (non considera le coppie che risultino scompagnate in entrambe le classificazioni)

Diagram

Description automatically generated with low confidence

Il rand index varia tra 0 e 1 (clustering perfettamente coincidenti)

## Mutual Information (MI, NMI)

Le mutual information (MI, NMI, AMI) misurano l'"accordo" tra le assegnazioni del cluster. Un punteggio più alto indica una somiglianza maggiore.

Il grado di accordo tra i cluster è calcolato da probabilità congiunte e marginali, espresse tramite gli indici:

- informazioni reciproche (MI)

- informazioni reciproche normalizzate (NMI), è la MI divisa per le entropie medie dei cluster

## V-Measure

Misura la correttezza delle assegnazioni del cluster usando l'analisi dell'entropia condizionale. Un punteggio più alto indica una somiglianza maggiore.

Si possono usare due metriche per valutare la correttezza delle assegnazioni del cluster:

- Omogeneità (HS): ogni cluster contiene solo membri di una singola classe (un po'come la "precision")

Text

Description automatically generated with medium confidence- Completezza (CS):tutti i membri di una determinata classe sono assegnati allo stesso cluster (un po' come la "recall")

Text

Description automatically generated with medium confidence

- La V-Measure è la media armonica di omogeneità e misura di completezza, simile al f-score è una media armonica di precision e recall.

Text

Description automatically generated with medium confidence

## Silhouette Coefficient

Il coefficiente di silhouette misura la distanza between-cluster vs la distanza within-cluster. Un punteggio più alto significa cluster meglio definiti.

per ogni singolo campione il coefficiente di silhouette misura la distanza media del campione da tutti gli altri elementi dei cluster più vicini (between-cluster) contro tutti gli altri elementi del suo stesso cluster (within-cluster).

Se il risultato di tale rapporto è "alto" significa che il singolo cluster è "lontano" dagli altri cluster lui vicini e che il cluster è ben definito.

Il coefficiente di silhouette di un dataset prende il coefficiente di silhouette medio per ciascun elemento

Graphical user interface, text

Description automatically generated

Il coefficiente di Silhouette va da -1 (modello molto scarso) a 1 (modello eccellente), al fine di poterlo usare successivamente con i restanti indici che vivono tra [0,1] applichiamo la funzione normalizzazione MIN-MAX con minimo scala a 0 e massimo scala a 1.

ovvero:

## Model\_comparison dataframe

Procederemo quindi alla valutazione dei vari algoritmi di clustering tramite gli indici selezionati.

Per ogni algoritmo di clustering saranno valutate le performace per diversi iperparametri, tutti i valori saranno inseriti nel "model\_comparison" dataframe il quale avrà la seguente struttura:

model\_comparison =[model\_type, model, ri, MI, NMI, HS, CS, V, siluetteN]

# Algoritmi di Clustering

Di seguito una veloce panoramica sui i vari algoritmi di clustering presi in esame, omettiamo il codice utilizzato per valutarli al variare dei specifici iperparametri per esso rimandiamo al file jupiter main.

## Affinity Propagation (AP)

Algoritmo di clustering basato sul concetto di “passaggio di messaggi” tra i campioni.

Esso non richiede che siano stabili a priori il numero dei cluster ma cerca tramite la misura di somiglianza Standar (negativo delle distanze euclidee al quadrato) i campioni "rappresentativi" per l’intero set di dati.

L'iperparametro che andremo a valutare è il “damping”(“smorzamento”) tra 0,5 e 1 .

## Agglomerative Clustering

Il clustering agglomerativo appartiene agli algoritmi di “clustering gerarchico”, ovvero partendo da dall'associazione di un ckluster ad ogni elemento del dataset esegue la "fusione" di dei vari cluster fino a raggiungere il numero desiderato di k-gruppi.

Si è allenato il modello Agglomerative Clustering con l'iperparametro k con valore in [4,10].

## BIRCH

Balanced Iterative Reducing and Clustering using Hierarchies, algoritmo di clustering che visualizza i dati in una struttura ad albero da cui vengono estratti i centroidi dei cluster.

Tale algoritmo integra il clustering gerarchico (utilizzato nello stadio iniziale di microclustering) con altri metodi di clustering quali il partizionamento iterativo (utilizzato nello stadio di macroclustering successivo).

Raggruppare in modo incrementale e dinamico i punti di dati metrici multidimensionali, con l’obiettivo di ottenere un clustering migliore in condizioni limitate, come una scarsa memoria o un tempo ridotto. Supera così le due principali difficoltà dei metodi di clustering agglomerativo (scalabilità e l’incapacità di annullare ciò che è stato fatto nei passi precedenti).

Si è allenato il modello BIRCH con l'iperparametro th (threshold) e k (n\_clusters) usati per fornire una stima del numero di cluster.

## DBSCAN

Density-Based Spatial Clustering of Applications with Noise, ricerca aree ad alta densità di campioni nello spazio multidimensionale e l’espansione di quelle aree come cluster, basandosi sulle loro caratteristiche.

Tale algoritmo è stato specificamente ideato per scoprire cluster di forma arbitraria. Basta avere un parametro di input e DBSCAN è in grado di aiutarti a determinare il suo esatto valore.

Si è allenato il modello DBSCAN con l'iperparametro “eps” costante (0.75) e n (min\_samples) che varia da 1 a 10 con passo 2.

## OPTICS

Ordering Points To Identify the Clustering Structure è una versione modificata del DBSCAN. Tale algoritmo differentemente dai precedenti non produce l’analisi di cluster di un set di dati, ma mostra la struttura di clustering basata sulla densità di un database.

Si è allenato il modello OPTICS con l'iperparametro “eps” costante (0.75) e n (min\_samples) che varia

## K-means

Tale algoritmo richiede di conoscere a priori il numero di cluster totali, classifica i punti dell’input nei vari cluster.

Aggiusta iterativamente la dimensione dei cluster, spostando i punti nei cluster più vicini a loro ad ogni iterazione, aggiorna i centroidi dei cluster finché non si raggiunge un criterio di ottimalità (minimizzare la varianza totale intra-gruppo).

Si è allenato il modello k-means con l'iperparametro k con valore in [4,10].

## Mini-Batch K-Means

Mini-Batch K-Means è una versione modificata di k-means che aggiorna i centroidi dei cluster usando dei piccoli batch dei campioni invece dell’intero set di dati, tale escamotage elimina il rumore e riduce i tempo di addestramento sui grandi dataset.

Si è allenato il modello Mini-Batch K-Means con l'iperparametro k con valore in [4,10].

## Mean shift

Il clustering Mean shift adatta i centroidi in base alla densità dei campioni nello spazio delle caratteristiche. È utilissimo per rilevare le varie modalità di densità per i dati discreti, perché attua uno spostamento medio ricorsivo al punto stazionario più vicino all’interno di una funzione di densità.

## Spectral Clustering

Algoritmo di clustering di spettro, prevede di usare i primi autovettori di una matrice derivata dalla distanza tra i punti.

Si è allenato il modello Spectral Clustering con l'iperparametro k (n\_clusters) con valore in [4,10].

## Gaussian Mixture

Il modello a miscela gaussiana, usa il principio della funzione di densità di probabilità multivariata con una miscela di distribuzioni di probabilità gaussiane.

Si è allenato il modello Gaussian Mixture con l'iperparametro k (n\_clusters) con valore in [4,10].

# Best Model Selection

Avendo esplorato diversi algoritmi di clustering e per quasi ogni uno diversi possibili iperparametri si è scelto di estrarre per ogni algoritmo di clustering il best-model basato sull'indice ottenuto dalla media degli score index:

* rand index
* normalized mutual nfo
* v\_measure il quale racchiude in se le informazioni degli indici (homogeneitye completeness)
* siluetteN

La scelta dell'uso dell'indice medio è dovuta all'osservazione che ogni indice osserva un aspetto specifico del risultato di clustering, siccome tutti vivono in un range tra [0,1] essi sono già normalizzati e quindi nessuno di essi "pesa" di più degli altri.

A livello di codice si è proceduto con il calcolo del valore medio degli indici osservati (resume\_index) per ogni rida del dataframe model\_comparison e a raggruppare gli elementi per tipologia di modello per poter estrarre per ogni gruppo i "migliori" rappresentanti

A screenshot of a computer

Description automatically generated

Per ogni "campione" di ogni algoritmo di clustering andremo ad analizzare le performance tramite:

* matrice di confusione
* scatterplot che presenta:
  + asse x: la specie originale dell'elemento
  + asse y: la specie assegnata all'elemento tramite il metodo del majority voting (ovvero per ogni cluster assegnamo l'etichetta degli elementi a massima rappresentanza all'interno di essa)
  + colore diverso per ogni cluster generato dall'algoritmo (riportato il leggenda)
* BarPlot della correlazione tra le features e il cluster assegnato dall'algoritmo comparato nel grafico speculare sottostante) con la correlazione tra le features e la specie (proveniente dal majority voting del cluster) all'interno del dataset originale

## AffinityPropagation k=0.5

Text

Description automatically generatedChart, scatter chart, bubble chart

Description automatically generated