Progetto di Data Mining 2.5
Clustering

Gruppo G2.5
Componenti:
Cocco Alessandro 65378
Cosseddu Marco 65367

Introduzione:

Questo progetto mira a condurre un'analisi approfondita del dataset "`pendigits`", un insieme di dati in cui ogni oggetto rappresenta l'immagine di una cifra scritta a mano. I dati appartengono a 10 categorie diverse (le cifre da 0 a 9). L'obiettivo primario è applicare tecniche di data mining per clusterizzare appropriatamente l'intero dataset.

Note sul dataset: il dataset `pendigits` si può scaricare dal repository openml (https://openml.org/), mentre le informazioni relative si trovano al link: https://openml.org/search?type=data&id=32

Fasi del Progetto:

1. Analisi del Dataset

Caricare il dataset tramite la libreria *scikit-learn*. Il dataset può essere scaricato mediante la funzione **`fetch_openml`**:

```
fetch_openml(name = 'pendigits')
```

N.B. la funzione restituisce un oggetto contenente i dati e le informazioni sul dataset. In particolare l'oggetto avrà i seguenti attributi:

- `data`: array contenente i dati;
- `target`: array contenente le label del dataset.
- `feature_names`: i nomi di ciascuna feature.

Inoltre, la descrizione completa del dataset è memorizzata nell'attributo `DESCR` dell'oggetto in questione.

Organizzare i dati scaricati in un DataFrame di **`pandas`**, eliminando le feature categoriche, e

visualizzare le prime righe per acquisire una panoramica delle variabili disponibili. Analizzare il dataset, discutere le caratteristiche dei dati e le aspettative del modello descrittivo.

2. Preprocessing

Esplorare il dataset per individuare dati mancanti o outlier. Eseguire la normalizzazione o standardizzazione delle variabili per garantire risultati più accurati durante la fase di clustering. Inoltre, proiettare i dati in uno spazio ridotto secondo diverse tecniche. In particolare, considerare i seguenti dataset, derivanti da quello iniziale:

- Il dataset originale, normalizzato/standardizzato;
- Il dataset trasformato mediante PCA, con un numero di componenti determinato dalla regola *elbow*, ordinando le componenti in base alla loro varianza e plottando in un apposito grafico;

Per ogni dataset trasformato, plottare la distribuzione dei dati in uno spazio bidimensionale considerando le prime due componenti, indicando la classe di appartenenza con opportuno colore o marker.

3. Clustering

3.1) Utilizzare i seguenti modelli di clustering, impostando come output 10 cluster per i primi 2 in elenco:

- k-Means;
- Hierarchical Agglomerative Clustering (HAC);
- DBSCAN (utilizzare i parametri di default);

Plottare, per ogni modello, le distribuzioni dei cluster nello spazio bidimensionale, analizzando e comparando a livello visivo i risultati, confrontandoli con la distribuzione originale. Inoltre, per valutare quantitativamente i cluster ottenuti da ogni modello, definire due funzioni, una che, dato un cluster (array di punti) restituisce il centroide (dove ogni coordinata è il valore medio della coordinata), e una che determina l'SSE (Sum of Squared Error), secondo la formula seguente:

$$SSE = \sum_{i} \sum_{x \in C_i} (x - c_i)^2$$

dove $\boldsymbol{C_i}$ è un dato cluster, $\boldsymbol{c_i}$ il suo centroide, e $\boldsymbol{x}$ sono i punti contenuti nel cluster. Riportare la valutazione per i tre modelli, analizzando e discutendo i risultati.

3.2) Considerando il modello DBSCAN, indicando con N il numero dei cluster, far variare N da 2 a 15, e riportare in un grafico l'SSE per ogni valore di N. Verificare, mediante la regola *elbow*, se il numero reale dei cluster (10) corrisponde al numero ottimale per rispettare tale regola.

Consegna del Progetto:

Gli studenti dovranno produrre il seguente materiale:

1. Il codice sorgente in linguaggio Python.
2. Una breve relazione che spiega le scelte di preprocessing, i modelli di clustering, e l'interpretazione dei risultati ottenuti.
3. Una presentazione (10-15 slide max), da presentare ai docenti in una sessione dedicata, che riassuma l'intero progetto.

In dettaglio, il materiale dovrà rispettare le seguenti caratteristiche.

1. Codice Sorgente

Il codice sorgente dovrà possibilmente essere ben commentato o essere incluso in un notebook Jupyter. Fare in modo che lo script visualizzi le informazioni relative a ciascuna fase descritta in precedenza, comprendendo i grafici generati.

2. Report Finale

Il report dovrà includere i seguenti paragrafi:

1. *Introduzione*. Comprende descrizione del problema e obiettivi del task assegnato.
2. *Dataset*. Descrizione del dataset e principali caratteristiche.
3. *Preprocessing*. Descrizione dei processi di elaborazione e trasformazione del dataset effettuati, riportando risultati ed eventuali grafici associati.
4. *Clustering*. Descrivere metodologie e modelli utilizzati, motivando eventualmente le scelte effettuate, e riportare i risultati ottenuti (compresi di eventuali grafici).
5. *Discussione e conclusioni*. Analisi e discussione dei risultati, includendo un paragrafo riassuntivo che descriva le considerazioni finali sui task e sui risultati ottenuti.

3. Presentazione

L'obiettivo è preparare una serie di slide (max 10-15) che riassumano e mostrino tutti i punti chiave del report. Includere una slide introduttiva e una conclusiva.