**1. Introduzione Decision\_tree\_classifier**

E’ stato quello di applicato al dataset un Decision Tree Classifier per la classificazione dei paesi in base al loro continente, utilizzando il dataset world\_population.csv. L'analisi ha previsto:

* Il preprocessing dei dati per garantirne la qualità e la compatibilità con il modello.
* Il tuning degli iperparametri, focalizzandosi su max\_depth e max\_leaf\_nodes.
* La valutazione del modello, utilizzando cross-validation, accuratezza e matrice di confusione.
* La visualizzazione dell’albero decisionale, per interpretare le regole apprese dal modello.

**2. Preprocessing dei Dati**

Per garantire che il dataset fosse pronto per la classificazione, sono state effettuate diverse operazioni di pre-elaborazione:

1. Selezione delle feature numeriche e della variabile target
   * Sono state scelte caratteristiche legate alla popolazione storica, alla densità e al tasso di crescita.
   * La variabile target è stata definita come Continent, ovvero la categoria in cui classificare i paesi.
2. Gestione dei valori mancanti
   * Tutti i valori NaN sono stati sostituiti con 0, per evitare problemi durante l’addestramento del modello.
3. Normalizzazione delle feature
   * Le caratteristiche numeriche sono state standardizzate usando StandardScaler(), per garantire che le variabili avessero lo stesso peso nel modello.
4. Conversione della variabile target
   * I nomi dei continenti sono stati trasformati in numeri interi utilizzando LabelEncoder(), rendendo la variabile compatibile con il classificatore.
5. Suddivisione del dataset in training e test set
   * Il dataset è stato diviso in 75% training e 25% test, con stratificazione sulla variabile target, per mantenere proporzioni equilibrate tra le classi.

**3. Tuning degli Iperparametri**

Per migliorare le prestazioni del Decision Tree, sono stati ottimizzati due iperparametri chiave:

1. Profondità massima dell’albero (max\_depth)
   * Controlla quanto in profondità l’albero può suddividere i dati.
   * Un valore troppo alto porta a overfitting (il modello si adatta troppo ai dati di training).
   * Un valore troppo basso porta a underfitting (il modello non apprende bene i pattern).
2. Numero massimo di foglie (max\_leaf\_nodes)
   * Controlla il numero massimo di nodi terminali dell’albero.
   * Un valore elevato crea un modello complesso, mentre un valore basso lo semplifica.

Per trovare i valori ottimali, è stata applicata la Cross-Validation (CV) su 10 folds.  
L'algoritmo ha testato combinazioni di max\_depth e max\_leaf\_nodes, selezionando la configurazione con la migliore accuratezza media sui dati di validazione.

**4. Valutazione del Modello**

Dopo il tuning, il modello finale è stato allenato con i migliori iperparametri trovati e testato sul set di test.  
Sono state calcolate le seguenti metriche:

1. Accuratezza finale
   * accuracy\_score() è stato utilizzato per misurare la percentuale di predizioni corrette.
2. Report di Classificazione
   * classification\_report() ha mostrato precision, recall, F1-score e supporto per ogni classe.
3. Matrice di Confusione
   * confusion\_matrix() è stata utilizzata per visualizzare gli errori di classificazione.
   * È stato creato un heatmap con Seaborn, evidenziando le classi più confuse tra loro.
4. Visualizzazione dell’Albero Decisionale
   * plot\_tree() ha permesso di interpretare le decisioni prese dal modello, mostrando la struttura e le condizioni utilizzate per suddividere i dati.

**5. Confronto dei Risultati**

Per valutare il miglioramento dovuto al tuning:

* È stato generato un grafico delle performance, che mostra l’accuratezza in funzione della profondità dell’albero.
* È stato osservato che la cross-validation ha prevenuto l’overfitting, trovando un compromesso ottimale tra complessità e accuratezza.

**Il modello ha mostrato buone performance, con alcune aree di miglioramento.  
-La matrice di confusione evidenzia errori di classificazione tra Asia, Europa e Nord America.  
-L’uso del tuning degli iperparametri ha permesso di migliorare l’accuratezza del modello.**

