

AI per la classificazione dei tumori

Esegui il seguente codice per capire quali valori assegnare alle variabili **A**, **B** e **C** per il tuo gruppo (usa il Group_Number assegnato).

```
from itertools import product

i = <Group_Number>

data = ["version_1.csv", "version_2.xlsx", "version_3.txt", "version_4.json",
"version_5.tsv"]
tech = ["Random Subsampling", "K-fold Cross Validation", "Leave-one-out Cross
Validation", "Leave-p-out Cross Validation", "Stratified Cross Validation",
"Stratified Shuffle Split", "Bootstrap"]

A, B, C = list(product(data, tech, tech))[i]
print("A:", A, "B:", B, "C:", C)
```

Descrizione:

In questo progetto esplorerete un dataset che contiene informazioni su varie caratteristiche delle cellule tumorali e la loro corrispondente classificazione come benigne o maligne. L'obiettivo è sviluppare un modello di apprendimento automatico e verificarne le prestazioni per classificare i tumori in base alle caratteristiche fornite.

Dataset:

Viene fornito un set di dati **A** contenente le 9 variabili indipendenti (features) e le corrispondenti label di classe delle cellule tumorali:

ID

Sample code number

Features:

Clump Thickness: 1 – 10
Uniformity of Cell Size: 1 – 10
Uniformity of Cell Shape: 1 – 10
Marginal Adhesion: 1 – 10
Single Epithelial Cell Size: 1 – 10
Bare Nuclei: 1 – 10
Bland Chromatin: 1 – 10
Normal Nucleoli: 1 – 10
Mitoses: 1 – 10

Class label:

Class: (2 per benigno, 4 per maligno)

Task:

Il vostro task è quello di sviluppare un programma che addestri e valuti un classificatore di machine learning per classificare i tumori come benigni o maligni in base alle caratteristiche fornite. L'obiettivo non è quello di sviluppare un modello ad alte prestazioni, ma di sviluppare una pipeline che addestri e valuti un modello in base alle caratteristiche specificate dall'utente.

Molto Importante

1. Leggete tutti gli step prima di iniziare a scrivere codice.

2. Come ogni attività di analisi dei dati, non esiste un modo unico e corretto. Poiché i risultati che otterrete possono dipendere dalle scelte che farete durante l'analisi, è molto importante (e necessario) che descriviate ogni singola decisione che prenderete e tutti i passi che farete.
3. Per facilitare lo svolgimento, potete iniziare con un sotto insieme del dataset, e poi estenderlo al dataset completo.
4. Il vostro codice deve essere il più generico possibile. Deve essere in grado di funzionare per ogni tipologia di input specificata dall'utente (Specifiche).
5. Non è consentito utilizzare librerie esterne che implementano direttamente gli algoritmi richiesti (ad esempio, Sklearn). Tuttavia, è possibile impiegare librerie come NumPy, Pandas e Matplotlib per supportare lo sviluppo e l'analisi del codice.

Seguire questi passaggi:

1. Data Preprocessing:

- Caricare il dataset.
- Gestire tutte le peculiarità in modo appropriato (scegliete il modo che preferite, non c'è una risposta corretta).
- Dividere il dataset in features (variabili indipendenti variables) and target label (class).

2. Model Development:

- Sviluppare un classificatore **k**-nn da scratch (non utilizzare librerie esterne che abbiano già implementato il classificatore **k**-nn):
 - Per ogni campione dell'insieme di test, calcolare la distanza da tutti i campioni del set di training e identificare i **k** campioni più vicini. Utilizzare la distanza euclidea.
 - Classificare ogni campione dell'insieme di test scegliendo la label più comune per i **k** vicini. Se c'è un pareggio tra le label, si sceglie in modo casuale.

3. Model Evaluation:

- Split del dataset in nei set di training e testing:
 - In **Holdout**, seguendo le **percentuali** specificate.
 - In **B** e **C**, seguendo i **K** esperimenti specificati: per ogni esperimento, valutare le prestazioni del modello sui dati di test e calcolare la media della metrica attraverso gli esperimenti.
- Utilizzare le metriche di validazione specificate.
- Una volta finita la validazione del modello. Salvate le performance su un file (es. excel) e con un plot (es. Confusion matrix, ROC curve, distribuzione performance delle sulle folds).

Specifiche:

L'uso del programma deve consentire all'utente di specificare:

- **k** (numero di vicini da utilizzare per il classificatore)
- come valutare il modello: in **Holdout**, in **B** o in **C**.
 - Se l'utente sceglie **Holdout**, può specificare la **percentuale** di dati da utilizzare nei set di training e di test.
 - Se l'utente sceglie **B** o **C**, può specificare **K** (il numero di esperimenti).
- quali **metriche** devono essere validate:
 - Accuracy Rate
 - Error Rate
 - Sensitivity
 - Specificity
 - Geometric Mean

- Area Under the Curve
- All the above

N.B.: **k** e **K** sono due parametri distinti.

Consegna:

Commentare per bene tutto il codice!

Tutto il codice dovrà essere versionato su GitHub.

Seguire il document **Guida_allo_sviluppo_di_progetti.pdf** come guida per la configurazione e lo sviluppo di un progetto.

Scrivere report sul file README descrivendo:

- Come eseguire il codice, con le varie opzioni di input che l'utente può specificare
- Come visualizzare ed interpretare i risultati ottenuti.