Machine Learning
Esercitazione di laboratorio sul confronto di vari classificatori

1) MATERIALE NECESSARIO (contenuto nel file .ZIP scaricabile dal sito del corso)

- common.py
- Amianto_ParamOpt.py
- Amianto_TrainingData.txt
- Amianto_TestData.txt (fornito durante l'esercitazione)

2) CONOSCENZE UTILI

La libreria scikit learn mette a disposizione numerose classi e funzionalità specifiche per il *machine learning*.

- Classificatori - tra cui `KNeighborsClassifier` (*k*-NN), `SVC` (SVM), `RandomForestClassifier` (Random Forest) e `AdaBoostClassifier` (AdaBoost).
- Cross-Validation:
  - `cross_val_score(estimator,X,y,cv)` – stima l'accuratezza di un classificatore (`estimator`) tramite Cross-Validation (con un numero di fold pari a `cv`). I pattern (`X`) sono memorizzati per righe in un Numpy array bidimensionale mentre le etichette (`y`) in un Numpy array monodimensionale.

3) TRAINING E VALIDATION

Utilizzare i classificatori messi a disposizione dalla libreria scikit learn per segmentare le zone con copertura in amianto (*segmentation*). I dati provengono da un set di immagini aeree acquisite con un drone professionale. Si dovranno individuare le combinazioni di iperparametri che permettono di massimizzare l'accuratezza dei vari classificatori utilizzando la tecnica K-fold Cross-Validation vista a lezione (pag. 20 delle dispense "Fondamenti").

a. Il file Amianto_TrainingData.txt contiene 20000 pattern 25-dimensionali etichettati (10000 amianto e 10000 non-amianto). Ogni pattern è relativo ad una posizione spaziale (x,y) in un'immagine aerea e codifica attraverso 25 feature le caratteristiche di colore e tessitura in un intorno 32×32 centrato in x,y.

b. Il file Amianto_ParamOpt.py è incompleto e contiene puntini di sospensione "…" laddove è richiesto di completare il codice.
c. Assegnare alle variabile `datasetFilePath` il percorso del file di traininig.
d. Assegnare alla variabile `foldCount` il numero di fold da utilizzare nella Cross-Validation (`cv`, K nelle dispense). Per questa esercitazione si consiglia di utilizzare un valore pari a 5.
e. Assegnare alla variabile `outputFolderPath` il percorso della cartella in cui salvare i risultati.
f. Assegnare alla variabile `classifierName` il nome del classificatore utilizzato. La variabile verrà impiegata per dare il nome al file contenente i risultati.
g. Analizzare l'esempio di ottimizzazione del *k*-NN effettuato utilizzando una *grid search* sull'iperparametro $k = \{1,3,5,7,9,11\}$. Ad ogni iterazione la funzione `cross_val_score` suddividerà i pattern in `foldCount` partizioni e addestrerà il classificatore `clf` (inizializzato con l'iperparametro *k* corrente) utilizzando di volta in volta `foldCount-1` partizioni come training e la restante come validation. Infine restituirà un Numpy array (`accuracies`) contenente le `foldCount` accuratezze di classificazione ottenute.
h. Calcolare l'accuratezza media (`mean`) delle `accuracies` ottenute ad ogni sessione di Cross-Validation.
i. Completare la tabella sottostante con il miglior valore di *k* individuato e la corrispondente accuratezza media. Per fare ciò, si consiglia di aprire il file dei risultati (memorizzato nella cartella `outputFolderPath` con il nome `classifierName.txt`) con Excel, suddividere il testo in colonne (Dati→Testo in colonne) e ordinare le righe per accuratezza media decrescente.

	k-NN	SVM Lineare	SVM RBF	Random Forest	AdaBoost
Iperparametri	K=	C=	C= gamma=	max_depth= n_estimators= max_features=	n_estimators= learning_rate=
Accuratezza Cross-Validation					
Accuratezza Test					

j. Ripetere l'ottimizzazione con i seguenti classificatori (per una descrizione dettagliata dei parametri dei vari classificatori si rimanda alla guida di scikit learn) e completare la tabella di conseguenza:
   - *SVM Lineare* – ottimizzare l'iperparametro `C` (costo). Pag. 9, 10 delle dispense "Classificazione (2)".
   - *SVM RBF* – ottimizzare gli iperparametri `gamma` (opera in modo inverso rispetto a σ presentato nelle dispense) e `C` (costo). Pag. 11, 12 delle dispense "Classificazione (2)".
   - *Random Forest* – ottimizzare gli iperparametri `max_depth`, `n_estimators` e `max_features` (d' nelle dispense). Pag. 28-30 delle dispense "Classificazione (2)".
   - *AdaBoost* – ottimizzare gli iperparametri `n_estimators` e `learning_rate`. Pag. 31-33 delle dispense "Classificazione (2)".
k. Cercare di evitare *overfitting* degli iperparametri, questo potrebbe portare a scarse capacità di generalizzazione (pag. 22, 23 delle dispense "Fondamenti").

4) <u>TEST</u>

Misurare l'accuratezza ottenuta dai classificatori (utilizzando gli iperparametri trovati al punto precedente) sul dataset di Test (scaricabile dal sito del corso durante l'esercitazione) per verificarne l'effettiva capacità di generalizzazione.

a. Il file Amianto_TestData.txt contiene 20000 pattern 25-dimensionali etichettati (10000 amianto e 10000 non-amianto).
b. Completare la tabella soprastante con l'accuratezza di classificazione ottenuta sul dataset di test. Per fare ciò si consiglia di prendere spunto dal codice dell'esercitazione precedente (kNN_Svm.py).