Machine Learning
Esercitazione di laboratorio sui classificatori *k*-NN e SVM

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

- common.py
- kNN_Svm.py
- pendigits_tr.txt
- pendigits_va.txt
- pendigits_te.txt (fornito durante l'esercitazione)
- pendigits_tr_Pca_K2.txt
- pendigits_va_Pca_K2.txt
- pendigits_te_Pca_K2.txt (fornito durante l'esercitazione)

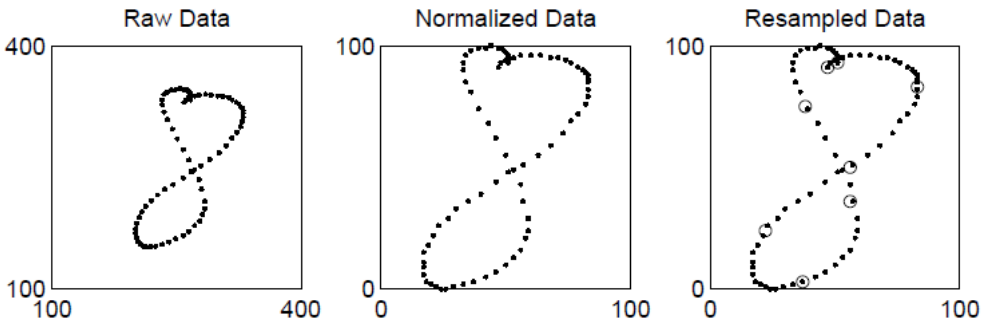
2) CONOSCENZE UTILI

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

- Classificatori – tra cui `KNeighborsClassifier` (*k*-NN) e `SVC` (SVM).
- I classificatori presenti nella libreria offrono una serie di metodi comuni per permettere al programmatore di astrarre dal classificatore utilizzato:
  - `fit(X,y)` – permette di addestrare il classificatore utilizzando un insieme di pattern etichettati. I pattern (`X`) sono memorizzati per righe in un Numpy array bidimensionale mentre le etichette (`y`) in un Numpy array monodimensionale.
  - `predict(T)` – esegue la classificazione dei pattern contenuti in `T` restituendo un Numpy array monodimensionale contenente le etichette delle classi dei singoli pattern.

3) ADDESTRAMENTO E SELEZIONE DEGLI IPERPARAMETRI

Utilizzare i classificatori *k*-NN e SVM per distinguere un dataset di cifre (0-9) scritte a mano. Si dovranno individuare le combinazioni di iperparametri che permettono di massimizzare l'accuratezza dei due classificatori sui dataset forniti.



Per ogni pattern sono presenti 16 feature costituite dalle coordinate *x*, *y* di 8 punti equispaziati.

a. I dataset pendigits_tr.txt (training) e pendigits_va.txt (validation) contengono entrambi 3665 pattern 16-dimensionali etichettati (cifre da 0 a 9). Viene fornita anche una versione bidimensionale (pendigits_tr_Pca_K2.txt e pendigits_va_Pca_K2.txt), ottenuta tramite riduzione della dimensionalità (mediante l'algoritmo *Principal Component Analysis*), utile per eseguire test iniziali visualizzando pattern e risultati.
b. Il file kNN_Svm.py è incompleto e contiene puntini di sospensione "…" laddove è richiesto di completare il codice.

c. Assegnare alle variabili `trFilePath e vaFilePath` i percorsi dei file di training e di validation che si vuole utilizzare. In prima istanza si consiglia di utilizzare la versione bidimensionale.
d. Assegnare alla variabile `featureCount` il numero di feature di ogni pattern. Questo valore deve essere coerente con i dataset impostati al punto precedente (16 o 2).
e. Implementare la funzione `compute_accuracy` che stimi l'accuratezza di classificazione (pag. 15 delle dispense "Fondamenti") di un generico classificatore (`classifier`) su un insieme di pattern (`data`) etichettato (`labels`).
f. Assegnare un nuovo classificatore alla variabile `clf` (*k*-NN o SVM) e addestrarlo utilizzando il dataset di training. Ogni classificatore ha una serie di iperparametri da impostare in fase di inizializzazione:
   - *k-NN* – ottimizzare l'iperparametro `k` (numero di elementi più vicini). Pag. 28 delle dispense "Classificazione (1)".
   - *SVM* – ottimizzare gli iperparametri `kernel` (Lineare o RBF), `C` (costo) ed eventualmente `gamma` (opera in modo inverso rispetto a σ presentato nelle dispense). Pag. 9, 12 delle dispense "Classificazione (2)".
g. Completare le tabelle sottostanti con i migliori iperparametri individuati e la corrispondente accuratezza di classificazione ottenuta.

2 dimensioni	k-NN	SVM
Iperparametri	K=	Kernel= C= gamma=
Accuratezza Training		
Accuratezza Validation		
Accuratezza Test		

16 dimensioni	k-NN	SVM
Iperparametri	K=	Kernel= C= gamma=
Accuratezza Training		
Accuratezza Validation		
Accuratezza Test		

h. Cercare di evitare *overfitting* degli iperparametri, questo potrebbe portare a scarse capacità di generalizzazione (pag. 22, 23 delle dispense "Fondamenti").

4) TEST

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

a. Il file pendigits_te.txt (pendigits_te_Pca_K2.txt) contiene 3662 pattern 16-dimensionali (bidimensionali) etichettati.
b. Completare le tabelle soprastanti con l'accuratezza di classificazione ottenuta sui dataset di test. E' sufficiente sostituire nel codice i riferimenti al validation set utilizzando il test set.