Università degli Studi di Verona

“Human Activity Recognition with Smartphones”

Fondamenti di Machine Learning

A.A. 2021/2022

Giulio Cappelletti – VR478827

Presentazione del Progetto

Questo progetto è valido per il corso di Fondamenti di Machine Learning tenuto presso l'Università degli Studi di Verona.

E' stato utilizzato il dataset "Human Activity Recognition with Smartphones", il quale presenta il seguente problema di classificazione: "Classificare uno o più record in una delle sei attività svolte". Il dataset è disponibile sulla piattaforma Kaggle al seguente link:

Human Activity Recognition with Smartphones | Kaggle

Diamo una panoramica delle parti che compongo lo sviluppo del progetto

- Risoluzione di un problema di classificazione, in quanto rappresenta il fulcro dell'elaborato.
- Sezione aggiuntiva con la risoluzione di un problema di Clustering supponendo che il dataset sia non etichettato, l'obiettivo diventa quello di andare a raggruppare correttamente le attività.
- Le principali tecniche utilizzate sono state implementate 'from scratch' in Python:
    - Principal Component Analysis
    - Fisher's Linear Discriminant Analysis
    - K Neirest Neighbor
    - K Means

Descrizione del Dataset

Come citato sul sito Kaggle dove è disponibile il Dataset:

"ll database di riconoscimento dell'attività umana è stato costruito dalle registrazioni di 30 partecipanti allo studio che svolgevano attività di vita quotidiana (ADL) mentre trasportavano uno smartphone montato in vita con sensori inerziali incorporati."

Le attività da riconoscere sono le seguenti:

- Standing (in piedi)
- Laying (in posa)
- Sitting (seduto)
- Walking (Camminare)
- Walking Downstairs (Camminare scendendo le scale)
- Walking Upstairs (Camminare salendo le scale)

Il Dataset è composto da due file che indicano rispettivamente **training set** e **testing set**, così strutturati :

- Le righe identificano i singoli record
- Le prime 561 colonne identificano le features
- La colonna 562, ovvero la colonna 'subject' indica il soggetto che ha svolto una determinata attività
- La colonna 563, ovvero la colonna 'Activity' indica l'attività svolta

Classificazione

Il codice per la Classificazione è disponibile nel file 'Classifier.ipynb'.

Nel corso dello svolgimento del progetto ho seguito una specifica struttura e pipeline:

1. **Data Collection & Data Management**
   In questa fase ho caricato il Dataset disponibile. Successivamente ho eseguito alcune operazioni di Exploratory Data Analysis con il fine di 'interrogare' il set di dati, in particolare analizzandone i parametri si è potuto notare che il dataset presenta media e deviazione standard notevolmente differenti tra le features:

	tBodyAcc-mean()-X	tBodyAcc-mean()-Y	tBodyAcc-mean()-Z	tBodyAcc-std()-X	tBodyAcc-std()-Y	tBodyAcc-std()-Z	tBodyAcc-mad()-X	tBodyAcc-mad()-Y	tBodyAcc-mad()-Z	tBodyAcc-max()-X
count	7352.000000	7352.000000	7352.000000	7352.000000	7352.000000	7352.000000	7352.000000	7352.000000	7352.000000	7352.000000
mean	0.274488	-0.017695	-0.109141	-0.605438	-0.510938	-0.604754	-0.630512	-0.526907	-0.606150	-0.468604
std	0.070261	0.040811	0.056635	0.448734	0.502645	0.418687	0.424073	0.485942	0.414122	0.544547

Inoltre ho effettuato la suddivisione dei dataset dalle label e dall'id l'id associato ai soggetti dal set dati, in modo da avere solamente le 561 features come colonne della matrice.

Infine, Poiché come visto al punto 1.2 la media e la deviazione standard per le features erano su scale differenti ho effettuato la standardizzazione del dataset, in modo da avere la media a 0 e la deviazione standard a 1.

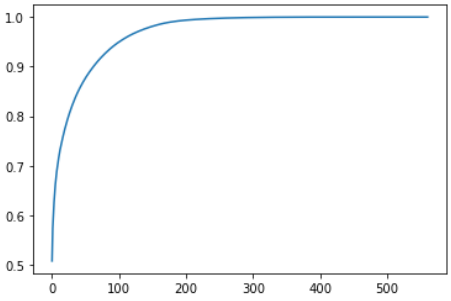
	tBodyAcc-mean()-X	tBodyAcc-mean()-Y	tBodyAcc-mean()-Z	tBodyAcc-std()-X	tBodyAcc-std()-Y	tBodyAcc-std()-Z	tBodyAcc-mad()-X	tBodyAcc-mad()-Y	tBodyAcc-mad()-Z	tBodyAcc-max()-X
count	7.352000e+03	7.352000e+03	7.352000e+03	7.352000e+03	7.352000e+03	7.352000e+03	7.352000e+03	7.352000e+03	7.352000e+03	7.352000e+03
mean	8.381882e-15	1.360597e-16	-5.324570e-15	-1.979555e-15	-2.678262e-15	-3.624866e-15	-4.231276e-15	2.603829e-15	2.975676e-15	3.792759e-16
std	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00	1.000000e+00

2. Features Exctractions

Dato il notevole numero di features (561), al fine di ridurre la dimensionalità, ho applicato ed implementato le seguenti tecniche:

1. PCA – Principal Component Analysis

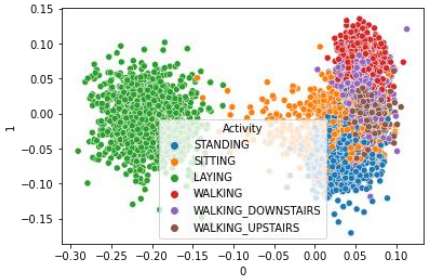
Nell'implementazione di PCA, si è tenuto conto della explained variance per decidere quante componenti principali selezionare



Il risultato di questa fase è stato quello di selezionare le prime 100 componenti principali, riducendo quindi il numero delle features da 561 a 100.

2. FLDA – Fisher's Linear Discriminant Analysis

In questa fase l'obiettivo è stato quello di discriminare al meglio i dati, che come si può notare dal plot precedente risultano sovrapposti.
E' stata quindi implementata FLDA con 6 classi, di conseguenza il risultato di questa fase è stato quello di avere un dataset con 5 features, permettendo una miglior distinzione delle classi.



3. Model Selection

In questa fase si è scelto di implementare il classificatore K Neirest Neighbor (KNN), valutando anche le sue diverse configurazioni in merito ai differenti valori di K.
E' stata poi applicata la tecnica di Cross Validation Leave One Out per testare KNN, ed effettuando numerose prove ho riscontrato si è riscontrato che in fase di validation i migliori risultati sono stati ottenuti con valori di k più alti di 5, nello specifico tra i k testati, k=24 è il valore del parametro che restituisce la miglior accuratezza. Di seguito sono state riportate solamente alcune prove:

```
#k=5:  0.7774755168661589
#k=6:  0.7808759521218716
#k=7:  0.786588683351469
#k=9:  0.7914853101196954
#k=24: 0.7995103373231773
```

4. **Testing**
   In questa fase ho applicato il classificatore KNN al test set

5. **Model Evaluation**
   Nella fase di testing il classificatore è stato valutato utilizzando le Matrice di confusione, da cui sono state poi estratte le seguenti metriche:
   - **Precision** = True Positive / (True Positive + False Positive)
   - **Recall** = True Positive / (True Positive + False Negative)

	precision	recall	f1-score	support
0	1.000000	0.996276	0.998134	537
1	0.789583	0.771894	0.780639	491
2	0.790826	0.810150	0.800371	532
3	0.702277	0.808468	0.751640	496
4	0.493671	0.371429	0.423913	420
5	0.586000	0.622081	0.603502	471
accuracy			0.744825	2947
macro avg	0.727059	0.730050	0.726366	2947
weighted avg	0.738745	0.744825	0.739802	2947

Dalle seguenti metriche si può notare che il classificatore classifica con una precisione:

- Precisione del 100% i soggetti che compiono l'attività 'Standing'
- Mentre la precisione si riduce nel caso in cui si considerano le classi 'Sitting' e 'Laying'
- Mentre si riduce drasticamente nella classificazione tra le attività di movimento come 'Walking', 'Walking Downstairs' 'Walking upstairs'

Clustering

E' stato affrontato anche un problema di Clustering prendendo in esame il solo training set e supponendolo non etichettato. Di conseguenza rispetto all'iter di classificazione è stato utilizzato solamente PCA per la features exctractions, in quanto tecnica di apprendimento non supervisionato.

1. **Clustering**
   E' stato scelto di implementare l'algoritmo K Means in quanto una tecnica di Hard Clustering ed adatta allo scopo di assegnare ad ogni punto una label.
   Il risultato sarà quindi un Dataset etichettato secondo le label generate da K Means .
   Di seguito è riportato il confronto tra un plot dei dati con le label generate da K Means (immagine a sinistra) ed il plot di dati dopo PCA con le etichette reali.