Riportiamo i grafici delle feature importance, nel caso con e senza principal component analysis. Più un attributo è in alto in classifica, più è significativo il suo utilizzo per le predizioni.

Senza PCA, caso arousal:

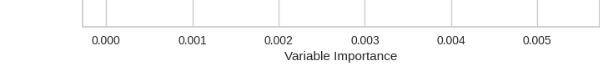
Immagine che contiene tavolo

Descrizione generata automaticamente

Immagine che contiene tavolo

Descrizione generata automaticamente

………….



Senza PCA, caso valence:

Immagine che contiene tavolo

Descrizione generata automaticamenteImmagine che contiene tavolo

Descrizione generata automaticamente

Per la parte con pca otteniamo:

Immagine che contiene tavolo

Descrizione generata automaticamente

Dove “P\_3” indica il terzo elettrodo, “p\_28” indica l’elettrodo numero 28 e così via.

Il codice nel caso senza PCA è:

from pycaret.classification import \*

import pandas as pd

import io

import numpy as np

# importo il dataset

dataset = pd.read\_csv(io.BytesIO(uploaded['EEG\_MANHOB\_w8\_s2.csv']))

# elimino le colonne che non mi servono

dataset = dataset.drop('video', axis=1)

dataset = dataset.drop('window', axis=1)

dataset = dataset.drop('subject', axis=1)

dataset = dataset.drop('dominance', axis=1)

dataset = dataset.drop('valence', axis=1)

dataset = dataset.drop('emotion', axis=1)

dataset = dataset.drop('predictance', axis=1)

data = dataset.sample(frac=0.9).reset\_index(drop=True)

data\_unseen = dataset.drop(data.index).reset\_index(drop=True)

# eseguo la classificazione su "valence", oppure "arousal"

exp\_reg101 = setup(data = data, target = 'arousal', session\_id=123)

# uso extraTreeClassifier

et = create\_model('et')

# valuto il modello

evaluate\_model(et)

predict\_model(et, data = data\_unseen);

final\_et = finalize\_model(et)

print(final\_et)

L’accuratezza, sia nel caso di valence, sia in quello di arousal, si aggira intorno al 90%.

**IMPORTANZA CANALE**

Per fare un confronto equo tra i due approcci, con e senza PCA, sarebbe utile capire quanto spesso le feature di un certo canale appaiono in alto sulla classifica della feature importance (nel caso senza PCA, in quanto il caso con PCA è già stato rappresentato in precedenza). Con “canale” intendiamo un determinato elettrodo: per esempio, nel grafico prima citato del caso con PCA, al primo posto si ha il canale “3”, al secondo il canale “28” e cosi via. Questo ci permetterà di mettere a confronto i due grafici in maniera più diretta, avendo in entrambi i casi un ordinamento basato sull' importanza di ciascun canale. Potremmo ottenere facilmente questa misura come:

IMPORTANZA CANALE N = media ((tot\_features - posizione\_attuale\_nel\_ordinamento(feature\_canale\_N))/tot\_features)

Esempio: dati 3 soli canali e 2 sole features (alpha e beta) un ordinamento di feature importance del tipo

Posizione 1) alpha\_canale\_2

Posizione 2) beta\_canale\_3

Posizione 3) beta\_canale\_2

Posizione 4) alpha\_canale\_1

Posizione 5) alpha\_canale\_3

Posizione 6) beta\_canale\_1

genera le seguenti importanze per canale

IMPORTANZA CANALE 1 = ( ((6 - 4)/6) + ((6 - 6)/6) ) / 2 = 0.16

IMPORTANZA CANALE 2 = ( ((6 - 1)/6) + ((6 - 3)/6) ) / 2 = 0.66

IMPORTANZA CANALE 3 = ( ((6 - 2)/6) + ((6 - 5)/6) ) / 2 = 0.41

In questo modo il canale 3, che appare più spesso in alto nella classifica, ha un valore di importanza maggiore. Al contrario il canale 1.

Eseguendo il codice, notiamo che la classifica viene all’incirca identica a quella del caso con PCA stilata precedentemente. La similarità della spiegazione potrebbe essere proprio relativa al dataset in questione: vuol dire che le dinamiche che osservate sono abbastanza astabili e che tutti i metodi di spiegazione sono concordi su cosa sia più o meno importante.

L’array restituito è:



Riassumiamo in una tabella i due risultati, con e senza PCA:

|  |  |
| --- | --- |
| con PCA | Senza PCA |
| 3 | 3 |
| 28 | 28 |
| 18 | 14 |
| 30 | 18 |
| 1 | 32 |
| 2 | 11 |
| 14 | 2 |
| 4 | 30 |
| 32 | 21 |
| 24 | 1 |
| 11 | 29 |
| 21 | 27 |
| 23 | 5 |
| 5 | 15 |
| 15 | 24 |
| 16 | 23 |
| 27 | 19 |
| 29 | 13 |
| 17 | 16 |
| 7 | 26 |
| 8 | 9 |
| 6 | 17 |
| 22 | 20 |
| 13 | 10 |
| 25 | 8 |
| 10 | 4 |
| 31 | 6 |
| 9 | 25 |
| 19 | 12 |
| 12 | 31 |
| 20 | 7 |
| 26 | 22 |

Il codice è:

import pandas as pd

import matplotlib.pyplot as plt

import io

import numpy as np

from sklearn.ensemble import ExtraTreesClassifier

from sklearn.model\_selection import train\_test\_split

from sklearn import metrics

from sklearn.metrics import r2\_score

dataset = pd.read\_csv(io.BytesIO(uploaded['EEG\_MANHOB\_w8\_s2.csv']))

y = dataset.iloc[:,0:1]

X = dataset.iloc[:,5:133]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X,y, test\_size=0.2)

etc = ExtraTreesClassifier(n\_estimators = 5)

etc.fit(X\_train,y\_train)

# Computing the importance of each feature

feature\_importance = etc.feature\_importances\_

importanza\_canale = [None]\*32

# adesso devo implementare importanza\_canale[i] = ( (128 - posizione\_classifica[1])/128 + (128 - posizione\_classifica[2])/128 + (128 - posizione\_classifica[3])/128 + (128 - posizione\_classifica[4])/128 )/4

for i in range(0,32):

  importanza\_canale[i] = 0

  for j in range(1,4):

    importanza\_canale[i] += ( (128 - feature\_importance[4\*i + j] )/128 )

  importanza\_canale[i] = importanza\_canale[i]/4

print(np.argsort(importanza\_canale) + 1)

y\_pred = etc.predict(X\_test)

print(r2\_score(y\_test, y\_pred))