

Proyecto introduccion a la inteligencia artificial

Juan sebastian Rodriguez Solarte- Juan pablo Restrepo Mancilla

Dataset

para utilizar el dataset, nos dirigimos al enlace de la competencia y se llenó el formulario, posteriormente llegó al correo el dataset

[\[top \]](#)

Download of data sets

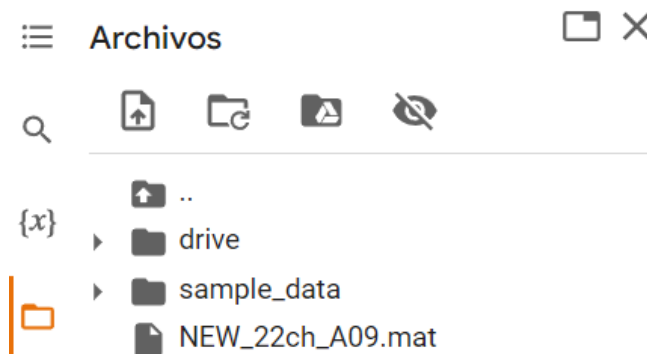
Each participant has to agree to give reference to the group(s) which recorded the data and to cite (one of) the paper listed in the respective description in each of her/his publications where one of those data sets is analyzed. Furthermore, we request each author to report any publication involving BCI Competition data sets to us for including it in our [list](#).

After filling out the form and pushing the "I Agree" button an automatic e-mail will be generated containing location and access information for the data set download area.

Name:

E-mail:

posteriormente se descargaron y se subieron directamente a coolab, debido al insuficiente espacio que hay en el drive



Descripción progreso alcanzado

Se descargo el dataset, obtuvimos sus características en donde vimos que utilizaban diccionarios, posteriormente obtuvimos sus llaves , para poder extraer la información importante, posteriormente miramos la dimensión de la señal que se quiere analizar.

```
#cargamos el dataset
mat_contents = sio.loadmat('NEW_22ch_A09.mat')

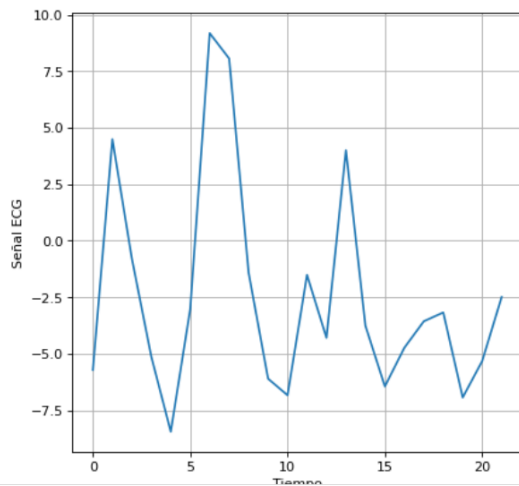
#los datos se cargan como un diccionario, se puede evaluar los campos que contiene
print("Los campos cargados son: " + str(mat_contents.keys()))

#queremos extraer solamente la señal eeg
data=sio.loadmat("NEW_22ch_A09.mat")["X"]
#Mostramos las dimensiones de la señal
dims_control = [sio.loadmat("NEW_22ch_A09.mat")["X"].shape ]
```

Una vez con la información lista , graficamos partes de las señal, para mejorar la comprensión de esta, sin embargo nos dimos cuenta que manejar una información tan

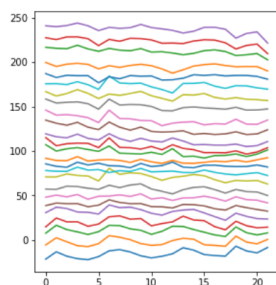
grande puede ser complicado, dado que cada dataset contiene más de 10000 datos.

```
data1=data[0,:,0]
#data2=data1.reshape(1,11000)
plt.figure(figsize=(6, 6), dpi=80) #se establece un tamaño del figure con el fin de que la grafica se vea de manera adecuada
plt.plot(data1) #se grafica la señal que se desea observar
plt.xlabel('Tiempo') #se nombra el eje x
plt.ylabel('Señal ECG') #se nombra el eje y
plt.grid() # se activa la cuadrícula en la grafico
```



```
#creamos el lienzo
fig = plt.figure(figsize=(10, 5))
ax = fig.add_subplot(1, 2, 2)
#graficamos la cantidad de canales que queremos
for canal in range(int(data.shape[0]/20)):
    ax.plot( data[canal,0:22,1] + canal*10)

plt.show()
```



Por tanto optamos por realizar un programa con interfaz gráfica que nos permitiera recorrer más fácilmente toda la señal de diferentes personas, por tanto se procedió a realizarlo mediante el paradigma orientado a objetos y utilizando MVC(modelo vista controlador), aparte de ello se sacaron parámetros importantes en la señales como máximos , mínimos, desviación estándar, valor cuadrático medio (RMS) Sample entropy, que se utiliza para evaluar la complejidad de las señales fisiológicas de series temporales y diagnosticar estados patológicos, estas características nos van a servir a la hora de crear el modelo de machine learning, por otra parte un reto que tenemos es identificar cuales son las partes de la señal que nos aportan más información

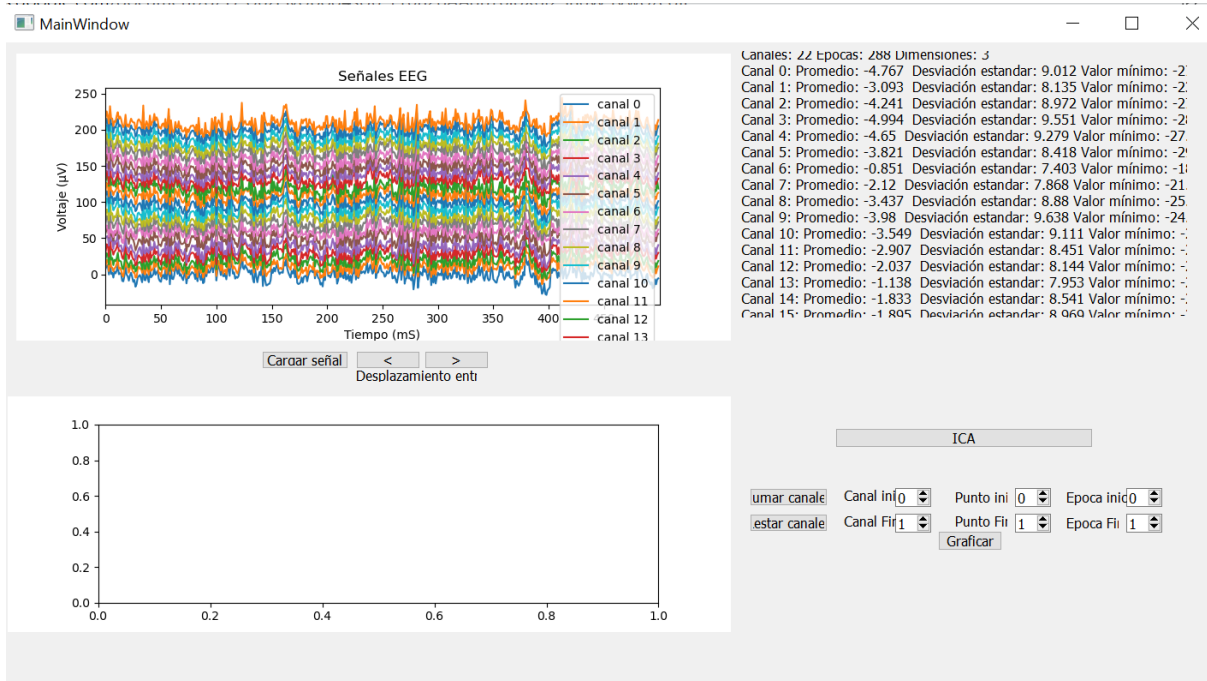
```

for i in range(canales):
    c = self.ica2[i, :]
    promedio = round(np.mean(c),3)
    desviacion = round(np.std(c),3)
    minim = round(np.min(c),3)
    maxim = round(np.max(c),3)
    kur = round(kurtosis(c,fisher=True),3)
    sk = round(abs(skew(self.ica2[i])),3)
    se = round(abs(float(ent.sample_entropy(self.ica2[i], 1, 0.2*desviacion))),3)

    # print(round(float(kur), 3))
    #print(kur)

    t = "\nCanal " + str(k) + ":" + " Promedio: " + str(promedio) + " Desviación estandar: " + str(desviac
    datos = datos + t
    k = k + 1
self.info_senal_ica.setText("Canales: " + str(canales) + " Dimensiones: " + str(dimen) + datos)

```



Se implementaron métodos ICA, los cuales se consideran como condición primordial para la independencia estadística. Buscando dentro de espacios multidimensionales los componentes principales, Se pretende que si las señales originales son estadísticamente independientes, las señales recuperadas también deben de serlo.

```
def devolverIca(self,data,epoca):
    S = []
    A = []

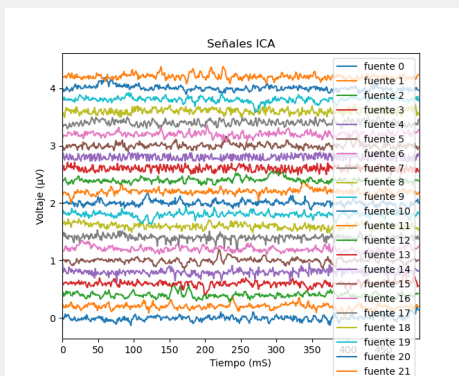
    ica = FastICA(n_components=len(data[0]), max_iter=1000, tol=0.1)
    S_ = ica.fit_transform(data[epoca].T) # Reconstruct signals
    A_ = ica.mixing_ # Get estimated mixing matrix

    S.append(S_)
    A.append(A_)
    # print("Modelo 65")
    # print(S[0].shape)

    # print("Hola" + str(S))
    # print("Es a:" + str(A))
    return S
```

Dialog

Graficar



Canales: 22 Dimensiones: 2

Canal 0: Promedio: 0.005 Desviación estándar: 0.047 Valor mínimo: -0.066 Valor máximo: 0.066 Kurtosis: 3.094 Skewness: 1.449 Sample Entropy: 2.224

Canal 1: Promedio: 0.002 Desviación estándar: 0.038 Valor mínimo: -0.065 Valor máximo: 0.065 Kurtosis: -0.735 Skewness: 0.239 Sample Entropy: 2.184

Canal 2: Promedio: -0.005 Desviación estándar: 0.04 Valor mínimo: -0.081 Valor máximo: 0.081 Kurtosis: 0.492 Skewness: 0.457 Sample Entropy: 1.859

Canal 3: Promedio: 0.0 Desviación estándar: 0.043 Valor mínimo: -0.111 Valor máximo: 0.111 Kurtosis: 0.375 Skewness: 0.456 Sample Entropy: 2.609

Canal 4: Promedio: 0.006 Desviación estándar: 0.031 Valor mínimo: -0.045 Valor máximo: 0.045 Kurtosis: -0.427 Skewness: 0.422 Sample Entropy: 2.351

Canal 5: Promedio: 0.005 Desviación estándar: 0.031 Valor mínimo: -0.038 Valor máximo: 0.038 Kurtosis: -0.896 Skewness: 0.455 Sample Entropy: 2.184

Canal 6: Promedio: 0.003 Desviación estándar: 0.037 Valor mínimo: -0.074 Valor máximo: 0.074 Kurtosis: -0.596 Skewness: 0.067 Sample Entropy: 2.498

Canal 7: Promedio: 0.001 Desviación estándar: 0.043 Valor mínimo: -0.109 Valor máximo: 0.109 Kurtosis: 1.336 Skewness: 0.129 Sample Entropy: 2.351

Canal 8: Promedio: 0.003 Desviación estándar: 0.046 Valor mínimo: -0.083 Valor máximo: 0.083 Kurtosis: 0.526 Skewness: 0.344 Sample Entropy: 2.11

Canal 9: Promedio: 0.008 Desviación estándar: 0.039 Valor mínimo: -0.059 Valor máximo: 0.059 Kurtosis: -0.829 Skewness: 0.045 Sample Entropy: 2.609

Canal 10: Promedio: 0.011 Desviación estándar: 0.039 Valor mínimo: -0.105 Valor máximo: 0.105 Kurtosis: 1.603 Skewness: 1.106 Sample Entropy: 2.264

Canal 11: Promedio: -0.008 Desviación estándar: 0.043 Valor mínimo: -0.132 Valor máximo: 0.132 Kurtosis: 0.876 Skewness: 1.098 Sample Entropy: 2.11

Canal 12: Promedio: -0.013 Desviación estándar: 0.056 Valor mínimo: -0.198 Valor máximo: 0.198 Kurtosis: 3.611 Skewness: 1.168 Sample Entropy: 2.075

Canal 13: Promedio: 0.002 Desviación estándar: 0.052 Valor mínimo: -0.143 Valor máximo: 0.143 Kurtosis: 1.216 Skewness: 0.74 Sample Entropy: 2.351

Canal 14: Promedio: -0.005 Desviación estándar: 0.044 Valor mínimo: -0.127 Valor máximo: 0.127 Kurtosis: 1.068 Skewness: 0.795 Sample Entropy: 2.11

Canal 15: Promedio: -0.004 Desviación estándar: 0.038 Valor mínimo: -0.093 Valor máximo: 0.093 Kurtosis: -0.074 Skewness: 0.411 Sample Entropy: 2.447

Canal 16: Promedio: -0.016 Desviación estándar: 0.043 Valor mínimo: -0.102 Valor máximo: 0.102 Kurtosis: 1.068 Skewness: 0.795 Sample Entropy: 2.11

Link dataset cargado

https://drive.google.com/drive/folders/1P2YI_d_Y0GUqp2zOynKhoHbJ9OvSI9xk?usp=sharing

Link video

<https://www.youtube.com/watch?v=9SrTk5bmWu0>