

Reporte de prácticas profesionales

Salvador Alejandro Uribe
Castellanos



Objetivo general

El objetivo principal de este trabajo es analizar los datos arrojados de un electroencefalograma aplicado a tres tipos de personas, la primera son personas saludables del cerebro, la segunda padece déficit de atención y por último, son personas que han sufrido daños cerebrales. Lo anterior se hizo con la intención de realizar un clasificador, el cual pueda segmentar a los 3 tipos de personas.

Para poder llevar a cabo el proyecto se utilizaran diferentes tipos de filtros y de clasificadores y de la misma manera se harán diferentes combinaciones con los mimos.

Este clasificador se hizo con la intención de mejorar la funcionalidad que tiene un electroencefalograma hacia los pacientes y de esta manera identificar fácilmente si un paciente tiene problemas o no.

Descripción de la base de datos

La base de datos está compuesta por tres carpetas, la primera se llama “Controles” donde están todos los registros de los sujetos que se encuentran saludables del cerebro, la segunda es “PELF” y hace referencia a los registros con las personas que tuvieron daños cerebrales y por último “TDAH” que son los registros de los sujetos con déficit de atención. En cada carpeta se encuentran los registros de cada sujeto, donde a cada uno se le realizaron alrededor de 300 pruebas. En la Tabla 1 se presentó un resumen del número de sujetos y pruebas realizadas según al segmento que pertenecen.

Tabla 1

Datos

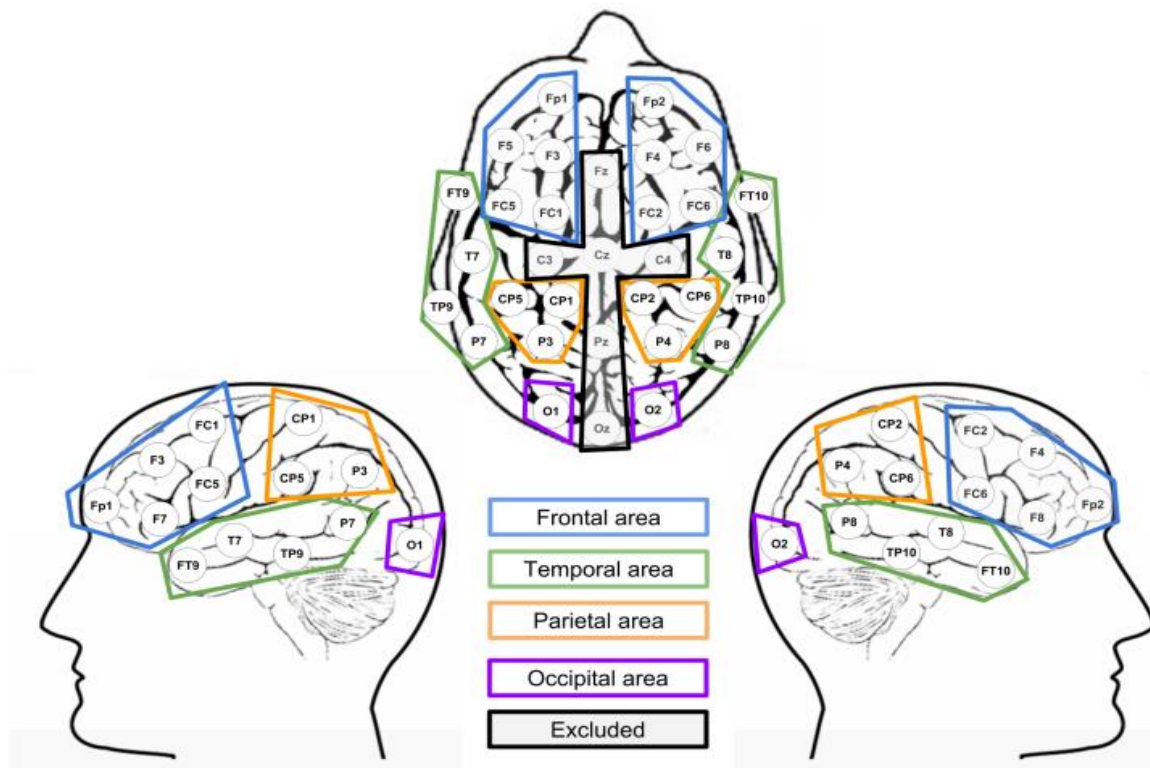
	# Sujetos	# Pruebas
Controles	15	5,512
PELF	9	3,268
TDAH	18	6,387
Total	42	15,167

Dentro de cada archivo csv se encuentra una matriz de 57X1,000, los cuales hacen referencia al comportamiento de los 57 electrodos a lo largo de 2 segundos pero solo se conectaron 32 durante las pruebas, así que, para este trabajo solo se utilizaron los primeros 32.

En la figura 1 se presentó la forma en que se asignaron los electrodos alrededor del cerebro.

Figura 1

Esquematzación de los electrodos



Como se aprecia en la Figura 1, el cerebro fue segmentado en 5 zonas; frontal, temporal, parietal, occipetal y la excluida, donde la zona excluida solo sirvió como referencia para los demás electrodos y por lo tanto no se tomó en cuenta para este trabajo.

De acuerdo a lo comentado en las diferentes sesiones realizadas se espera que la zona frontal arroje los mejores resultados ya que es la zona en la que se puede obtener mejor información.

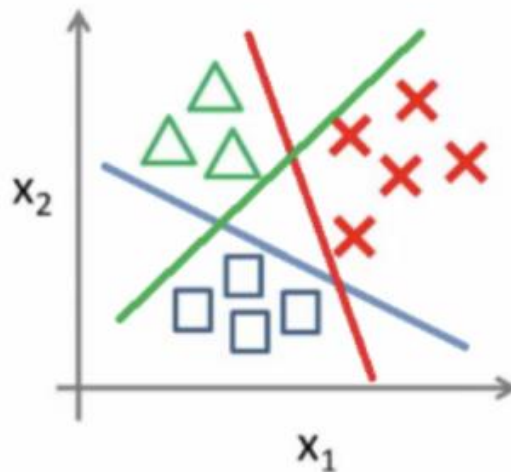
Procesos utilizados

Para realizar un clasificador en donde existen tres tipos de muestras se puede hacer en más de una manera, pero en este caso sólo se utilizaron dos, que son;

Multiclase. Este consiste en elaborar un solo clasificador en el cual a los datos de entrenamiento se le asignen 3 o más tipos de muestras, es decir en esta caso le asigné 1 a controles, 2 a TDAH y 3 a PELF. En la Figura 2 se presentó un ejemplo gráfico de este tipo de clasificador.

Figura 2

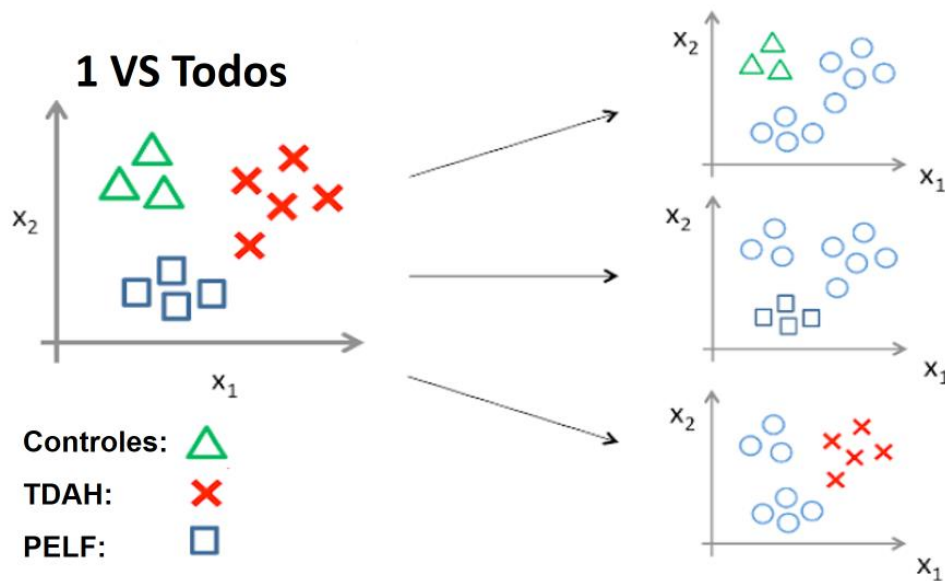
Multiclase



1VSTodos. Desde mi punto de opinión este tipo de clasificador es un proceso, ya que se deben de realizar varios pasos. El primero está en realizar un clasificador por cada tipo de muestra y como en este trabajo se tienen tres, se realizaron los tres clasificadores, así como se presentó en la Figura 2, donde en la gráfica de la izquierda se tienen los tres segmentos agrupados y después se segmentan en cada uno, por ejemplo para hacer el primer clasificador que se hizo respecto de controles, los datos de entrenamiento que eran asignados como 1 (correcto) fueron los de controles y los datos de PELF y TDAH eran 0 (incorrecto).

Figura 3

1VSTodos



Por último se realiza una predicción de cada clasificador con los mismos registros en cada uno y para que al final el resultado se cuente como correcto solo el clasificador de la muestra correcta debe de arrojar un uno, es decir, si sabemos que la muestra es del registro de Controles, entonces el clasificador de controles debe de arrojar un uno y los otros dos ceros, en caso contrario se tomó como si fuera un error.

Para comparar los clasificadores entre sí se utilizó la métrica **exactitud**, la cual hace referencia al total de aciertos correctos que tuvo el clasificador entre el total de pruebas.

A continuación se describieron los procesos utilizados para la construcción de los filtros y los clasificadores. De la misma manera en la sección *Anexos* se encuentran los códigos utilizados en Python (spyder).

Estandarizado-Logit (Multiclase)

El primer pre-procesamiento aplicado a los datos fue la estandarización, para que todos los valores tuvieran media cero, esto lo hice porque en ocasiones anteriores en otras bases de datos me había funcionado, de la misma manera por el conocimiento que tenía en ese momento preferí utilizar el clasificador logit ya que lo comprendía del todo. Por lo tanto en este método estandaricé los datos e hice un clasificador multiclase logit.

SVM-Logit (Multiclase)

Después de investigar un poco más sobre los electroencefalogramas, encontré que la máquina de vectores de soporte es muy utilizada en esta área como filtro en los datos, así que, opté por utilizarlo y como clasificador volví a usar el multiclase logit.

Fourier-Logit (Multiclase / 1VSTodos)

Cuando el doctor Gregorio entró a las sesiones me recomendó que utilizara como filtro la transformada de Fourier porque tiene mucha sinergia con el clasificador logit y también porque tiene un buen funcionamiento en esta área.

Resultados

En este subcapítulo se presentaron todos los resultados obtenidos de los clasificadores por medio de la métrica exactitud, en donde fueron segmentados por zona y por número de sujetos, esto se hizo con la intención de saber cuál zona es la que aporta mayor información y también para saber cómo se comporta con respecto del número de sujetos.

En la sección se multiclase en algunos métodos solo se hizo con respecto de la zona frontal por la suposición de que esta era la mejor, pero al analizar los resultados de mis compañeros nos dimos cuenta que dichos resultados pueden variar según el método que se le apliqué y se concluyó que lo mejor sería hacerlo con respecto de las tres principales zonas, que son; frontal, temporal y parietal.

Figura 4

Multiclase

Multiclase				
Método	# Sujetos	Frontal	Temporal	Parietal
Estandarización Logit	3	39.28%		
	9	36.02%		
	Todas	34.67%		
SVM Logit	3	42.39%		
	9	33.92%		
	Todas	35.49%		
Fourier Logit	3	80.59%	94.25%	81.36%
	9	65.54%	62.40%	61.47%
	Todas	58.50%	62.59%	56.78%

Figura 5

1 VS Todos

1 VS Todos				
Método	# Sujetos	Frontal	Temporal	Parietal
Fourier Logit	3	82.11%	96.12%	83.81%
	9	55.76%	54.44%	52.89%
	Todas	46.13%	39.92%	50.73%

Conclusiones

Los clasificadores para este tipo de estudios suelen ser un poco complicados por diversas razones como el procesamiento y en escoger el clasificador adecuado, así que, de primera instancia solo se deben de escoger los clasificadores y filtros que generalmente se utilizan en esta área, ya que, generalmente de esta manera se podrán obtener los mejores resultados posibles. Otra dificultad que se tuvo en este trabajo fue el manejo de base de datos, ya que, existen procesos que se aplicaron y que fueron relativamente pesados para la computadora y aún más cuando se manejaron cantidades grandes de datos.

Para poder obtener un clasificador con un buen rendimiento se esperaría que al menos tuviera el 70% de exactitud, lamentablemente los únicos clasificadores que tuvieron ese porcentaje o mayor fue para 3 o 2 sujetos y con esto no tendríamos la suficiente prueba para poder segmentar a cualquier sujeto.

Por último se obtuvo que el mejor clasificador aplicado a todos los sujetos fue el multiclase Fourier logit en la zona temporal con el 62.59%.

A continuación se presentaron los códigos utilizados para la obtención de los resultados presentados en este trabajo, cabe recalcar que sólo se presentaron los trabajos para la zona frontal ya que se utilizaron las mismas líneas de código para las demás zonas tomando en cuenta que para ello solo se cambiaron los electrodos.

SVM – Logit (Multiclase)

```
# Importo todas las librerias
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import precision_score
import pandas as pd
import glob
import os
import pywt

# Adjunto todos los archivos "Controles"
os.chdir(r'C:\Users\reysi\Desktop\Salvador\Prácticas\Conjunto de datos\Datos
Segmentados\Controles')

carpetas = glob.glob('*')

df=pd.DataFrame()

# Realizo un for para tomar en cuenta todos los pacientes de "Controles"
for i in carpetas:

    os.chdir(r'C:\Users\reysi\Desktop\Salvador\Prácticas\Conjunto de datos\Datos
Segmentados\Controles\' +i )

    archivos = glob.glob('*')

    # Creo un DataFrame solamente con los electrodos de la Zona frontal derecha
    for leer in archivos:

        columna = pd.read_csv(leer, header=None)
```

```

columna = pd.DataFrame(columna)

fc6 = columna.iloc[26]
fc2 = columna.iloc[27]
f4 = columna.iloc[28]
f8 = columna.iloc[29]
fp2 = columna.iloc[30]
fp1=columna.iloc[0]
f3= columna.iloc[2]
f7=columna.iloc[3]
fc5=columna.iloc[5]
fc1=columna.iloc[6]

promedio=(fc6 + fc2 + f4 + f8 + fp2+fp1+f3+f7+fc5+fc1) / 10

df=df.append(promedio,ignore_index=(True))

Base_R1=df

Base_R1=Base_R1.drop(3472,axis=0)

Base_R1.reset_index()

del promedio

Base_R1=Base_R1.iloc[:,0:Base_R1.shape[1]-2]


# Adjunto todos los archivos "TDAH"

os.chdir(r'C:\Users\reysi\Desktop\Salvador\Prácticas\Conjunto de datos\Datos
Segmentados\TDAH')

carpetas = glob.glob('*')

df=pd.DataFrame()

# Realizo un for para tomar en cuenta todos los pacientes de "Controles"

for i in carpetas:

    os.chdir(r'C:\Users\reysi\Desktop\Salvador\Prácticas\Conjunto de datos\Datos
Segmentados\TDAH\' +i)

    archivos = glob.glob('*')

    # Creo un DataFrame solamente con los electrodos de la Zona frontal derecha

```

```

for leer in archivos:

    columna = pd.read_csv(leer, header=None)
    columna = pd.DataFrame(columna)
    fc6 = columna.iloc[26]
    fc2 = columna.iloc[27]
    f4 = columna.iloc[28]
    f8 = columna.iloc[29]
    fp2 = columna.iloc[30]
    fp1=columna.iloc[0]
    f3= columna.iloc[2]
    f7=columna.iloc[3]
    fc5=columna.iloc[5]
    fc1=columna.iloc[6]

    promedio=(fc6 + fc2 + f4 + f8 + fp2+fp1+f3+f7+fc5+fc1) / 10

    df=df.append(promedio,ignore_index=(True))

Base_R2=df

del promedio


# Adjunto todos los archivos "PELF"

os.chdir(r'C:\Users\reysi\Desktop\Salvador\Prácticas\Conjunto de datos\Datos
Segmentados\PELF')

carpetas = glob.glob('*')

df=pd.DataFrame()

# Realizo un for para tomar en cuenta todos los pacientes de "PELF"

for i in carpetas:

    os.chdir(r'C:\Users\reysi\Desktop\Salvador\Prácticas\Conjunto de datos\Datos
Segmentados\PELF\' +i)

    archivos = glob.glob('*')

    # Creo un DataFrame solamente con los electrodos de la Zona frontal derecha

    for leer in archivos:

```

```

columna = pd.read_csv(leer, header=None)

columna = pd.DataFrame(columna)

fc6 = columna.iloc[26]

fc2 = columna.iloc[27]

f4 = columna.iloc[28]

f8 = columna.iloc[29]

fp2 = columna.iloc[30]

fp1=columna.iloc[0]

f3= columna.iloc[2]

f7=columna.iloc[3]

fc5=columna.iloc[5]

fc1=columna.iloc[6]

promedio=(fc6 + fc2 + f4 + f8 + fp2+fp1+f3+f7+fc5+fc1) / 10

df=df.append(promedio,ignore_index=(True))

Base_R3=df

del promedio

```

```

# Aplico wavelet a las 3 bases

cA,cD= pywt.wavedec(Base_R1,'db4',level = 1, mode='periodic')

Base_R1_W=pd.DataFrame(cA)

listaa=[]

for uno1 in range(len(Base_R1_W.axes[0])):

    listaa.append(0)

Base_R1_W['y']=listaa

```

```

cA,cD= pywt.wavedec(Base_R2,'db4',level = 1, mode='periodic')

Base_R2_W=pd.DataFrame(cA)

listaa=[]

for uno1 in range(len(Base_R2_W.axes[0])):

```

```

listaa.append(1)
Base_R2_W['y']=listaa

cA,cD= pywt.wavedec(Base_R3,'db4',level = 1, mode='periodic')
Base_R3_W=pd.DataFrame(cA)
listaa=[]
for uno1 in range(len(Base_R3_W.axes[0])):
    listaa.append(2)
Base_R3_W['y']=listaa

# Junto los datos de ambas secciones con su bandera respectiva
datos = Base_R1_W
datos =datos.append(Base_R2_W,ignore_index=(True))
datos=datos.append(Base_R3_W,ignore_index=(True))
val_fal=datos.isnull().sum()

# Segmentamos la base de datos y dividimos la base de datos en 2
# El 80% de los datos se utilizará para entrenar al modelo
# El 20% restante para evaluarlo
y = datos.y
X=datos.drop('y',axis=1)
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=42,stratify=y)

# Aplicamos el modelo
from sklearn.linear_model import LogisticRegression
model = LogisticRegression()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

```

```
#Matriz de confusion
```

```
from sklearn.metrics import confusion_matrix
```

```
matriz = confusion_matrix(y_test,y_pred)
```

```
print(matriz)
```

```
#Cálculo precision del modelo
```

```
from sklearn.metrics import precision_score
```

```
precision = precision_score(y_test,y_pred,average='micro')
```

```
print(precision)
```

```
#Cálculo exactitud del modelo
```

```
from sklearn.metrics import accuracy_score
```

```
exactitud = accuracy_score(y_test,y_pred)
```

```
print(exactitud)
```

```
#Cálculo sensibilidad
```

```
from sklearn.metrics import recall_score
```

```
sensibilidad = recall_score(y_test,y_pred,average='micro')
```

```
print("Sensibilidad del modelo " + str(sensibilidad))
```

Fourier – Logit (Multiclase)

```
# Importo todas las librerias
```

```
from sklearn.model_selection import train_test_split
```

```
from sklearn.metrics import confusion_matrix
```

```
from sklearn.metrics import precision_score
```

```
import pandas as pd
```

```
import glob
```

```

import os

import pywt

# Adjunto todos los archivos "Controles"

os.chdir(r'C:\Users\reysi\Desktop\Salvador\Prácticas\Conjunto de datos\Datos
Segmentados\Controles')

carpetas = glob.glob('*')

df=pd.DataFrame()

# Realizo un for para tomar en cuenta todos los pacientes de "Controles"
for i in carpetas:

    os.chdir(r'C:\Users\reysi\Desktop\Salvador\Prácticas\Conjunto de datos\Datos
Segmentados\Controles\' +i )

    archivos = glob.glob('*')

    # Creo un DataFrame solamente con los electrodos de la Zona frontal derecha
    for leer in archivos:

        columna = pd.read_csv(leer, header=None)

        columna = pd.DataFrame(columna)

        fc6 = columna.iloc[26]

        fc2 = columna.iloc[27]

        f4 = columna.iloc[28]

        f8 = columna.iloc[29]

        fp2 = columna.iloc[30]

        fp1=columna.iloc[0]

        f3= columna.iloc[2]

        f7=columna.iloc[3]

        fc5=columna.iloc[5]

        fc1=columna.iloc[6]

        promedio=(fc6 + fc2 + f4 + f8 + fp2+fp1+f3+f7+fc5+fc1) / 10

        df=df.append(promedio,ignore_index=(True))

Base_R1=df

```



```
Base_R1=Base_R1.drop(3472,axis=0)
```

```
Base_R1.reset_index()
```

del promedio

```
# Modificado#####3
```

```
#Esta linea es buena #
```

```
Base_R1=Base_R1.iloc[:,0:Base_R1.shape[1]-2]
```

```
# Adjunto todos los archivos "TDAH"
```

```
os.chdir(r'C:\Users\reysi\Desktop\Salvador\Prácticas\Conjunto de datos\Datos  
Segmentados\TDAH')
```

```
carpetas = glob.glob('*')
```

```
df=pd.DataFrame()
```

```
# Realizo un for para tomar en cuenta todos los pacientes de "Controles"
```

```
for i in carpetas:
```

```
    os.chdir(r'C:\Users\reysi\Desktop\Salvador\Prácticas\Conjunto de datos\Datos  
Segmentados\TDAH\' +i)
```

```
    archivos = glob.glob('*')
```

```
# Creo un DataFrame solamente con los electrodos de la Zona frontal derecha
```

```
for leer in archivos:
```

```
    columna = pd.read_csv(leer, header=None)
```

```
    columna = pd.DataFrame(columna)
```

```
    fc6 = columna.iloc[26]
```

```
    fc2 = columna.iloc[27]
```

```
    f4 = columna.iloc[28]
```

```
    f8 = columna.iloc[29]
```

```
    fp2 = columna.iloc[30]
```

```
    fp1=columna.iloc[0]
```

```
    f3= columna.iloc[2]
```

```
    f7=columna.iloc[3]
```

```

fc5=columna.iloc[5]

fc1=columna.iloc[6]

promedio=(fc6 + fc2 + f4 + f8 + fp2+fp1+f3+f7+fc5+fc1) / 10

df=df.append(promedio,ignore_index=(True))

Base_R2=df

del promedio


# Adjunto todos los archivos "PELF"

os.chdir(r'C:\Users\reysi\Desktop\Salvador\Prácticas\Conjunto de datos\Datos
Segmentados\PELF')

carpetas = glob.glob('*')

df=pd.DataFrame()

# Realizo un for para tomar en cuenta todos los pacientes de "PELF"

for i in carpetas:

    os.chdir(r'C:\Users\reysi\Desktop\Salvador\Prácticas\Conjunto de datos\Datos
Segmentados\PELF\' +i)

    archivos = glob.glob('*')

    # Creo un DataFrame solamente con los electrodos de la Zona frontal derecha

    for leer in archivos:

        columna = pd.read_csv(leer, header=None)

        columna = pd.DataFrame(columna)

        fc6 = columna.iloc[26]

        fc2 = columna.iloc[27]

        f4 = columna.iloc[28]

        f8 = columna.iloc[29]

        fp2 = columna.iloc[30]

        fp1=columna.iloc[0]

        f3= columna.iloc[2]

        f7=columna.iloc[3]

        fc5=columna.iloc[5]

```

```

fc1=columna.iloc[6]

promedio=(fc6 + fc2 + f4 + f8 + fp2+fp1+f3+f7+fc5+fc1) / 10

df=df.append(promedio,ignore_index=(True))

Base_R3=df

del promedio

```

```

# Aplico la transformada de fourier

```

```

import scipy.fftpack as fourier

Base_R1_F=fourier.fft(Base_R1)

Base_R1_F=abs(Base_R1_F)

```

```

Base_R2_F=fourier.fft(Base_R2)

Base_R2_F=abs(Base_R2_F)

```

```

Base_R3_F=fourier.fft(Base_R3)

Base_R3_F=abs(Base_R3_F)

```

```

#Asignamos las banderas

```

```

Base_R1_F=pd.DataFrame(Base_R1_F)

listaa=[]

for uno1 in range(len(Base_R1_F.axes[0])):

    listaa.append(0)

Base_R1_F['y']=listaa

```

```

Base_R2_F=pd.DataFrame(Base_R2_F)

listaa=[]

for uno1 in range(len(Base_R2_F.axes[0])):

    listaa.append(1)

Base_R2_F['y']=listaa

```

```
Base_R3_F=pd.DataFrame(Base_R3_F)
```

```
listaa=[]
```

```
for uno1 in range(len(Base_R3_F.axes[0])):
```

```
    listaa.append(2)
```

```
Base_R3_F['y']=listaa
```

```
# Junto los datos de ambas secciones con su bandera respectiva
```

```
datos = Base_R1_F
```

```
datos =datos.append(Base_R2_F,ignore_index=(True))
```

```
datos=datos.append(Base_R3_F,ignore_index=(True))
```

```
val_fal=datos.isnull().sum()
```

```
# Segmentamos la base de datos y dividimos la base de datos en 2
```

```
# El 80% de los datos se utilizará para entrenar al modelo
```

```
# El 20% restante para evaluarlo
```

```
y = datos.y
```

```
X=datos.drop('y',axis=1)
```

```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=42,stratify=y)
```

```
# Aplicamos el modelo
```

```
from sklearn.linear_model import LogisticRegression
```

```
model = LogisticRegression()
```

```
model.fit(X_train, y_train)
```

```
y_pred = model.predict(X_test)
```

```
#Matriz de confusion
```

```
from sklearn.metrics import confusion_matrix  
matriz = confusion_matrix(y_test,y_pred)  
print(matriz)
```

#Cálculo precision del modelo

```
from sklearn.metrics import precision_score  
precision = precision_score(y_test,y_pred,average='micro')  
print(precision)
```

#Cálculo exactitud del modelo

```
from sklearn.metrics import accuracy_score  
exactitud = accuracy_score(y_test,y_pred)  
print(exactitud)
```

#Cálculo sensibilidad

```
from sklearn.metrics import recall_score  
sensibilidad = recall_score(y_test,y_pred,average='micro')  
print("Sensibilidad del modelo " + str(sensibilidad))
```

Fourier – Logit (1VTodos)

Importo todas las librerias

```
from sklearn.model_selection import train_test_split  
from sklearn.linear_model import LogisticRegression  
from sklearn.metrics import accuracy_score  
from sklearn.metrics import confusion_matrix  
import pandas as pd  
import glob  
import os
```

CONTROLES

```

os.chdir(r'C:\Users\Salvador Uribe\Desktop\Salvador\Prácticas\Conjunto de datos\Datos
Segmentados\Controles')

carpetas= glob.glob('*')

df=pd.DataFrame()

for i in carpetas:

    os.chdir(r'C:\Users\Salvador Uribe\Desktop\Salvador\Prácticas\Conjunto de datos\Datos
Segmentados\Controles\\' +i )

    archivos = glob.glob('*')

    # Creo un DataFrame solamente con los electrodos de la Zona frontal derecha

    for leer in archivos:

        columna = pd.read_csv(leer, header=None)

        columna = pd.DataFrame(columna)

        fc6 = columna.iloc[26]

        fc2 = columna.iloc[27]

        f4 = columna.iloc[28]

        f8 = columna.iloc[29]

        fp2 = columna.iloc[30]

        fp1=columna.iloc[0]

        f3= columna.iloc[2]

        f7=columna.iloc[3]

        fc5=columna.iloc[5]

        fc1=columna.iloc[6]

        promedio=(fc6 + fc2 + f4 + f8 + fp2+fp1+f3+f7+fc5+fc1) / 10

        df=df.append(promedio,ignore_index=(True))


Base_R1=df.drop(3472,axis=0)

Base_R1.reset_index()

del promedio

Base_R1=Base_R1.iloc[:,0:Base_R1.shape[1]-2]

```

```
# TDAH #
```

```
# Adjunto todos los archivos "TDAH"
```

```
os.chdir(r'C:\Users\Salvador Uribe\Desktop\Salvador\Prácticas\Conjunto de datos\Datos  
Segmentados\TDAH')
```

```
carpetas = glob.glob('*')
```

```
df=pd.DataFrame()
```

```
# Realizo un for para tomar en cuenta todos los pacientes de "Controles"
```

```
for i in carpetas:
```

```
    os.chdir(r'C:\Users\Salvador Uribe\Desktop\Salvador\Prácticas\Conjunto de datos\Datos  
Segmentados\TDAH\' +i)
```

```
    archivos = glob.glob('*')
```

```
    # Creo un DataFrame solamente con los electrodos de la Zona frontal derecha
```

```
    for leer in archivos:
```

```
        columna = pd.read_csv(leer, header=None)
```

```
        columna = pd.DataFrame(columna)
```

```
        fc6 = columna.iloc[26]
```

```
        fc2 = columna.iloc[27]
```

```
        f4 = columna.iloc[28]
```

```
        f8 = columna.iloc[29]
```

```
        fp2 = columna.iloc[30]
```

```
        fp1=columna.iloc[0]
```

```
        f3= columna.iloc[2]
```

```
        f7=columna.iloc[3]
```

```
        fc5=columna.iloc[5]
```

```
        fc1=columna.iloc[6]
```

```
        promedio=(fc6 + fc2 + f4 + f8 + fp2+fp1+f3+f7+fc5+fc1) / 10
```

```
        df=df.append(promedio,ignore_index=(True))
```

```
Base_R2=df
```

del promedio

```
# PELF #

# Adjunto todos los archivos "PELF"

os.chdir(r'C:\Users\Salvador Uribe\Desktop\Salvador\Prácticas\Conjunto de datos\Datos
Segmentados\PELF')

carpetas = glob.glob('*')

df=pd.DataFrame()

# Realizo un for para tomar en cuenta todos los pacientes de "PELF"

for i in carpetas:

    os.chdir(r'C:\Users\Salvador Uribe\Desktop\Salvador\Prácticas\Conjunto de datos\Datos
Segmentados\PELF\' + i)

    archivos = glob.glob('*')

    # Creo un DataFrame solamente con los electrodos de la Zona frontal derecha

    for leer in archivos:

        columna = pd.read_csv(leer, header=None)

        columna = pd.DataFrame(columna)

        fc6 = columna.iloc[26]

        fc2 = columna.iloc[27]

        f4 = columna.iloc[28]

        f8 = columna.iloc[29]

        fp2 = columna.iloc[30]

        fp1=columna.iloc[0]

        f3= columna.iloc[2]

        f7=columna.iloc[3]

        fc5=columna.iloc[5]

        fc1=columna.iloc[6]

        promedio=(fc6 + fc2 + f4 + f8 + fp2+fp1+f3+f7+fc5+fc1) / 10

        df=df.append(promedio,ignore_index=(True))

Base_R3=df
```


del promedio

#Transformada de FOURIER#

```
import scipy.fftpack as fourier
```

```
Base_R1=fourier.fft(Base_R1)
```

```
Base_R1=abs(Base_R1)
```

```
Base_R1=pd.DataFrame(Base_R1)
```

```
Base_R2=fourier.fft(Base_R2)
```

```
Base_R2=abs(Base_R2)
```

```
Base_R2=pd.DataFrame(Base_R2)
```

```
Base_R3=fourier.fft(Base_R3)
```

```
Base_R3=abs(Base_R3)
```

```
Base_R3=pd.DataFrame(Base_R3)
```

Controles VS Todos

```
Base_R1_controles=pd.DataFrame(Base_R1)
```

```
listaa=[]
```

```
for uno1 in range(len(Base_R1_controles.axes[0])):
```

```
    listaa.append(1)
```

```
Base_R1_controles['y']=listaa
```

```
Errores_1 = Base_R2
```

```
Errores_1 = Errores_1.append(Base_R3)
```

```
listaa=[]
```

```
for ceroo in range(len(Errores_1.axes[0])):
```

```
    listaa.append(0)
```

```
Errores_1['y']=listaa
```

```
Base_R1_controles = Base_R1_controles.append(Errores_1,ignore_index=(True))  
val_fal_1=Base_R1_controles.isnull().sum()
```

```
y_1 = Base_R1_controles.y  
X_1=Base_R1_controles.drop('y',axis=1)  
X_train_1,X_test_1,y_train_1,y_test_1=  
train_test_split(X_1,y_1,test_size=0.2,random_state=42,stratify=y_1)
```

```
model_1 = LogisticRegression()  
model_1.fit(X_train_1, y_train_1)  
y_pred_1 = model_1.predict(X_test_1)  
y_test_1=pd.DataFrame(y_test_1)  
y_test_1.reset_index(drop=True)
```

```
predicciones_1 = pd.DataFrame(y_pred_1)  
predicciones_1['REAL_1'] =y_test_1.reset_index(drop=True)  
#Matriz  
matriz_1 = confusion_matrix(y_test_1,y_pred_1)  
print(matriz_1)  
#Exactitud  
exactitud_1 = accuracy_score(y_test_1,y_pred_1)  
print(exactitud_1)
```

TDAH VS Todos

```
Base_R2_TDAH=pd.DataFrame(Base_R2)  
listaa=[]  
for uno1 in range(len(Base_R2_TDAH.axes[0])):  
    listaa.append(1)
```

```
Base_R2_TDAH['y']=listaa
```

```
Errores_2 = Base_R1
```

```
Errores_2 = Errores_2.append(Base_R3)
```

```
listaa=[]
```

```
for ceroo in range(len(Errores_2.axes[0])):
```

```
    listaa.append(0)
```

```
Errores_2['y']=listaa
```

```
Base_R2_TDAH = Base_R2_TDAH.append(Errores_2,ignore_index=(True))
```

```
val_fal_2=Base_R2_TDAH.isnull().sum()
```

```
y_2 = Base_R2_TDAH.y
```

```
X_2=Base_R2_TDAH.drop('y',axis=1)
```

```
X_train_2,X_test_2,y_train_2,y_test_2=  
train_test_split(X_2,y_2,test_size=0.2,random_state=42,stratify=y_2)
```

```
model_2 = LogisticRegression()
```

```
model_2.fit(X_train_2, y_train_2)
```

```
y_pred_2 = model_2.predict(X_test_2)
```

```
y_test_2=pd.DataFrame(y_test_2)
```

```
y_test_2.reset_index(drop=True)
```

```
predicciones_2 = pd.DataFrame(y_pred_2)
```

```
predicciones_2['REAL_2'] =y_test_2.reset_index(drop=True)
```

```
#Matriz
```

```
matriz_2 = confusion_matrix(y_test_2,y_pred_2)
```

```
print(matriz_2)
```

```
#Exactitud
```

```
exactitud_2 = accuracy_score(y_test_2,y_pred_2)
```

```
print(exactitud_2)
```

```
# PELF VS Todos #
```

```
Base_R3_PELF=pd.DataFrame(Base_R3)
```

```
listaa=[]
```

```
for uno1 in range(len(Base_R3_PELF.axes[0])):
```

```
    listaa.append(1)
```

```
Base_R3_PELF['y']=listaa
```

```
Errores_3 = Base_R1
```

```
Errores_3 = Errores_3.append(Base_R2)
```

```
listaa=[]
```

```
for cero0 in range(len(Errores_3.axes[0])):
```

```
    listaa.append(0)
```

```
Errores_3['y']=listaa
```

```
Base_R3_PELF= Base_R3_PELF.append(Errores_3,ignore_index=(True))
```

```
val_fal_3=Base_R3_PELF.isnull().sum()
```

```
y_3 = Base_R3_PELF.y
```

```
X_3=Base_R3_PELF.drop('y',axis=1)
```

```
X_train_3,X_test_3,y_train_3,y_test_3=
```

```
train_test_split(X_3,y_3,test_size=0.2,random_state=42,stratify=y_3)
```

```
model_3 = LogisticRegression()
```

```
model_3.fit(X_train_3, y_train_3)
```

```
y_pred_3 = model_3.predict(X_test_3)
```

```
y_test_3=pd.DataFrame(y_test_3)
```

```
y_test_3.reset_index(drop=True)
```

```
#Matriz
```

```
matriz_3 = confusion_matrix(y_test_3,y_pred_3)
```

```
print(matriz_2)
```

```
#Exactitud
```

```
exactitud_3 = accuracy_score(y_test_3,y_pred_3)
```

```
print(exactitud_3)
```

```
# Realizo las predicciones
```

```
Base = Base_R1.reset_index(drop=True)
```

```
Base = Base.append(Base_R2).reset_index(drop=True)
```

```
Base = Base.append(Base_R3).reset_index(drop=True)
```

```
val_fal=Base.isnull().sum()
```

```
Base=Base.drop(['y'],axis=1)
```

```
pred_1_1 = model_1.predict(Base)
```

```
pred_2_2 = model_2.predict(Base)
```

```
pred_2_2 = pd.DataFrame(pred_2_2)
```

```
pred_3_3 = model_3.predict(Base)
```

```
pred_3_3 = pd.DataFrame(pred_3_3)
```

```
# Importo todas las predicciones a un excel
```

```
os.chdir(r'C:\Users\Salvador Uribe\Desktop\Salvador\Prácticas\Programas\Final_Final')
```

```
predicciones_total = pd.DataFrame(pred_1_1)
```

```
predicciones_total['Pred_TDAH'] = pred_2_2
```

```
predicciones_total['Pred_PELF'] = pred_3_3  
predicciones_total['Conteo_Controles'] = len(Base_R1)  
predicciones_total['Conteo_TDAH'] = len(Base_R2)  
predicciones_total['Conteo_PELF'] = len(Base_R3)  
  
predicciones_total.to_csv('Predicciones_Todas.csv')
```