



## **Análisis de datos de actividad cerebral**

**José Carlos Yamuni Contreras A01740285**

**Profesor Omar Mendoza Montoya**

**TC2034.201**

**Zapopan, Jalisco 13/06/2023**

### **¿Qué es la electroencefalografía?**

La electroencefalografía, o EEG, es un examen neurológico no invasivo que sirve principalmente para la detección de afecciones neurológicas. Es comúnmente utilizada para diagnosticar trastornos convulsivos, metabólicos, infecciosos y de sueño. Además, es útil para monitorear la actividad cerebral de una persona al estar anestesiada o haber perdido el conocimiento, siendo también capaz de confirmar la muerte cerebral.

La electroencefalografía monitorea la actividad cerebral a través del cráneo midiendo potenciales eléctricos. Para esto se adhieren una serie de electrodos al cuero cabelludo con la ayuda de una pasta conductora especial, siendo estos lo que funge como terminal positiva ya que es donde se da el mayor potencial eléctrico. Además, se adhieren otros electrodos donde el potencial eléctrico es más débil, siendo esta la terminal negativa.

Las señales detectadas por los sensores viajan a través de hilos conductores a una máquina, la cual interpreta las señales como una serie de oscilaciones en el tiempo. Una forma común de interpretar las señales es con los rangos de frecuencias:

- Delta: 0.1 Hz a 4 Hz
- Teta: 4 Hz a 8 Hz
- Alfa: 8 Hz a 12 Hz
- Beta: 12 Hz a 30 Hz
- Gamma: 30 Hz en adelante

Un alto nivel de energía en las bajas frecuencias indica un estado mental relajado, mientras que un alto nivel de energía en las altas frecuencias indica que se está realizando una tarea cognitiva de alto nivel. A pesar de que esta interpretación es común, no es una verdad absoluta y puede haber casos en los que no sirva.

## Características de señales adquiridas

El equipo utilizado para el proceso de encefalografía fue el Unicorn Hybrid Black lanzado en 2019 desarrollado por la compañía Gtec. Esta interfaz evalúa la actividad de EEG a una velocidad de muestreo de 250 Hz en ocho canales distintos. Tomando como referencia el sistema 10-20, los canales evaluados son los siguientes

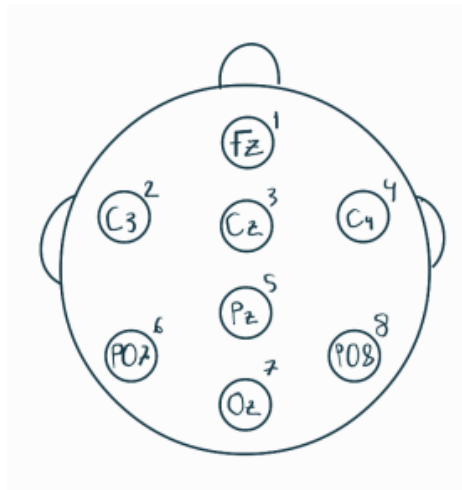


Figura 1. Canales medidos: Fz, C3, Cz, C4, Pz, PO7, Oz, PO8

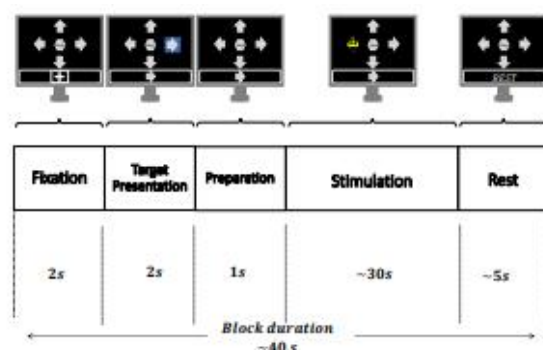
Además, como referencias se adhieren otros dos electrodos debajo de los lóbulos de las orejas.

Además, el rango de señales de EEG que se espera obtener de los experimentos a realizar es de -100 microvoltios a 100 microvoltios, todo lo que esté fuera de ese rango, es posiblemente ruido y no sirve para entrenar un clasificador. Por lo tanto, antes de comenzar los experimentos se debe verificar que las señales registradas sean estables y responsivas ante acciones realizadas por el individuo.

## Descripción de experimento P300



(a)



(b)

Figura 2. Experimento P300

1. Se muestra un símbolo de cruz en la pantalla por dos segundos en la parte inferior de la interfaz que te indica que debes estar preparado pero relajado.
2. Uno de los símbolos se remarca, lo cual indica que el sujeto debe prestar atención a ese símbolo.
3. Durante treinta segundos una carita feliz hace múltiples apariciones fugaces sobre los cinco símbolos. El sujeto debe contar las veces que apareció sobre el símbolo que le fue indicado anteriormente.
4. El sujeto tiene cinco segundos de descanso.
5. Se repite este proceso hasta que la carita feliz aparezca al menos 280 veces.

### **Descripción de experimento de Tareas Cognitivas**

1. Se muestra un símbolo de cruz en la pantalla por diez segundos que te indica que debes estar preparado pero relajado.
2. Se despliega una de las doce tareas cognitivas a realizar. Esta puede ser de recordar nombres, lectura, o ejercicios de cálculo. Esta fase también tiene una duración de diez segundos.
3. El sujeto tiene 5 segundos de descanso.
4. Se repite este proceso hasta que aparezcan las doce tareas cognitivas.

### **Extracción de características**

La extracción de características es un proceso mediante el cual se busca transformar la matriz de características original, a una con componentes cuyas propiedades sean útiles para obtener información de mayor calidad.

#### **P300:**

1. En el momento que la carita feliz hace una aparición, se hace un corte desde ese punto hasta 800 milisegundos después. Tomando en cuenta que se cuenta con una velocidad de muestreo de 250 Hz, se sabe que dentro de este corte se contará con 200 muestras por canal.
2. Se aplica un proceso de filtrado de pase de banda para así eliminar frecuencias fuera de una banda de frecuencias seleccionada, para así poder separar y analizar las frecuencias dentro de esa banda. En este caso, se aplica el filtro de 4 Hz a 20 Hz.

3. Se aplica un submuestreo para analizar menos mediciones. En este caso se hace un submuestreo entre cuatro, es decir, de tener 200 muestras en cada canal pasamos a tener solo 50.
4. Por último, se aplica una técnica de filtrado espacial llamada análisis de correlación canónica. De forma muy superficial, lo que hace esta técnica es combinar las señales de forma promediada, para después obtener cuatro nuevas señales de las cuales se esperan propiedades mejoradas, por ejemplo: menos ruido. Estas cuatro señales finales con cincuenta muestras cada una son las características finales que recibe el clasificador.

### **Tarea cognitiva:**

- Contamos con tres fases de experimentales: no tarea cognitiva con una duración de 10 segundos, tarea cognitiva con una duración de 10 segundos y descanso con una duración de cinco segundos.
- Para la extracción de características descartamos la sesión de descanso, y dividiremos cada muestra de las fases de no tarea cognitiva y tarea cognitiva en 10 ventanas de análisis con duración de un segundo y 250 muestras cada una.
- Para cada ventana, en todos los canales se realiza un análisis de potencia espectral (PSD). El cual analiza el nivel de energía aportado por cada frecuencia. A mayor oscilación, mayor es el nivel de energía aportado por la frecuencia dentro del segundo que se está analizando dentro del canal en cuestión.
- La información obtenida se concatena y así se obtiene el vector de características con el que se entrenará al clasificador.

### **Proceso de clasificación**

Para clasificar las observaciones se utilizaron cinco algoritmos de clasificación de la librería Scikit-Learn de Python: Support Vector, k-Nearest Neighbors, Multilayer Perceptron, Random Forest y Gradient Boosting.

Para mejorar los resultados de los clasificadores se realizó submuestreo para equilibrar las muestras de los conjuntos de datos ya que estas estaban demasiado desbalanceadas. Esto representó un cambio muy positivo en las sensibilidades de las clases. Para hacer esto se utilizó el método imbalanced-learn, implementado en Scikit-Learn como RandomUnderSampler.

Para cada uno de estos algoritmos se realizó selección de hiperparámetros utilizando búsqueda exhaustiva con la función GridSearchCV de Scikit-Learn. Por cuestiones de tiempo y costo computacional no se tomaron en cuenta todos los hiperparámetros de los modelos.

Probando distintos valores, estos fueron los hiperparámetros evaluados en cada algoritmo:

- Support vector:  $C = \{0.1, 1, 10\}$ , Función kernel = {linear, radial}
- k-Nearest Neighbors: Número de vecinos = {3, 5, 7}, Pesos = {uniforme, distancial}, Métrica = {euclidiana, manhattan}
- Multilayer Perceptron: Capas y neuronas por capa = {(2, 2), (3, 3)}
- Random Forest: Número de árboles = {450, 500}
- Gradient Boosting: Etapas de incremento = {150, 200}

Por encima de todo esto, se implementó la validación estratificada k-Fold con cinco pliegues, la cual está también implementada en Scikit-Learn.

## Resultados

Se realizaron tres clasificadores, uno para el experimento P300 y dos para el experimento de tareas cognitivas. Para cada clasificador se utilizaron dos conjuntos de datos de los siguientes sujetos de prueba: José Carlos Yamuni Contreras y Michel Emiliano Bureau Romo.

### Clasificador P300 vs. No P300

Es un problema de clasificación binaria en el cual se busca clasificar si el sujeto localiza o no a la carita feliz en la flecha indicada.

A continuación, se muestran las tablas de resultados de las muestras de Michel Bureau:

- Support Vector:

	precision	recall	f1-score	support
P300	0.41	0.74	0.52	204
NoP300	0.92	0.74	0.82	826
accuracy	0.74	1030		
macroavg	0.66	0.74	0.67	1030
weightedavg	0.82	0.74	0.76	1030

- K-Nearest Neighbors:

	precision	recall	f1-score	support
P300	0.32	0.8	0.46	204
NoP300	0.92	0.58	0.71	826
accuracy	0.62	1030		
macroavg	0.62	0.69	0.58	1030
weightedavg	0.8	0.62	0.66	1030

- Multilayer Perceptron:

	precision	recall	f1-score	support
P300	0.37	0.76	0.5	204
NoP300	0.92	0.68	0.78	826
accuracy	0.7	1030		
macroavg	0.64	0.72	0.64	1030
weightedavg	0.81	0.7	0.73	1030

- Random Forest:

	precision	recall	f1-score	support
P300	0.43	0.74	0.54	204
NoP300	0.92	0.76	0.83	826
accuracy	0.75	1030		
macroavg	0.67	0.75	0.68	1030
weightedavg	0.82	0.75	0.77	1030

- Gradient Boosting:

	precision	recall	f1-score	support
P300	0.41	0.76	0.53	204
NoP300	0.93	0.73	0.81	826
accuracy	0.73	1030		
macroavg	0.67	0.75	0.67	1030
weightedavg	0.82	0.73	0.76	1030

A continuación, se muestran las tablas de resultados de las muestras de José Yamuni:

- Support Vector:

	precision	recall	f1-score	support
P300	0.36	0.65	0.47	269
NoP300	0.89	0.72	0.79	1079
accuracy	0.7	1348		
macroavg	0.63	0.68	0.63	1348
weightedavg	0.79	0.7	0.73	1348

- K-Nearest Neighbors:

	precision	recall	f1-score	support
P300	0.27	0.54	0.36	269
NoP300	0.85	0.63	0.73	1079
accuracy	0.61	1348		
macroavg	0.56	0.59	0.54	1348
weightedavg	0.73	0.61	0.65	1348

- Multilayer Perceptron:

	precision	recall	f1-score	support
P300	0.32	0.69	0.44	269
NoP300	0.89	0.64	0.74	1079
accuracy	0.65	1348		
macroavg	0.61	0.67	0.59	1348
weightedavg	0.78	0.65	0.68	1348

- Random Forest:

	precision	recall	f1-score	support
P300	0.36	0.72	0.48	269
NoP300	0.91	0.68	0.77	1079
accuracy	0.68	1348		
macroavg	0.63	0.7	0.62	1348
weightedavg	0.8	0.68	0.71	1348

- Gradient Boosting:

	precision	recall	f1-score	support
P300	0.34	0.65	0.45	269
NoP300	0.89	0.68	0.77	1079
accuracy	0.67	1348		
macroavg	0.61	0.67	0.61	1348
weightedavg	0.78	0.67	0.7	1348

### **Clasificador Tarea Cognitiva vs. No Tarea Cognitiva**

Es un problema de clasificación binaria en el cual se busca clasificar si el sujeto está realizando o no una tarea cognitiva.



A continuación, se muestran las tablas de resultados de las muestras de Michel Bureau:

- Support Vector:

	precision	recall	f1-score	support
NoTask	0.51	0.61	0.55	75
Task	0.75	0.65	0.7	130
accuracy	0.64	205		
macroavg	0.63	0.63	0.63	205
weightedavg	0.66	0.64	0.64	205

- K-Nearest Neighbors:

	precision	recall	f1-score	support
NoTask	0.44	0.59	0.51	75
Task	0.71	0.58	0.64	130
accuracy	0.58	205		
macroavg	0.58	0.58	0.57	205
weightedavg	0.61	0.58	0.59	205

- Multilayer Perceptron:

	precision	recall	f1-score	support
NoTask	0.41	0.55	0.47	75
Task	0.67	0.54	0.6	130
accuracy	0.54	205		
macroavg	0.54	0.54	0.53	205
weightedavg	0.58	0.54	0.55	205

- Random Forest:

	precision	recall	f1-score	support
NoTask	0.48	0.59	0.53	75
Task	0.73	0.63	0.67	130
accuracy	0.61	205		
macroavg	0.6	0.61	0.6	205
weightedavg	0.64	0.61	0.62	205

- Gradient Boosting:

	precision	recall	f1-score	support
NoTask	0.41	0.51	0.45	75
Task	0.67	0.58	0.62	130
accuracy	0.55	205		
macroavg	0.54	0.54	0.54	205
weightedavg	0.57	0.55	0.56	205

A continuación, se muestran las tablas de resultados de las muestras de José Yamuni:

- Support Vector:

	precision	recall	f1-score	support
NoTask	0.48	0.67	0.56	60
Task	0.84	0.72	0.78	152
accuracy	0.7	212		
macroavg	0.66	0.69	0.67	212
weightedavg	0.74	0.7	0.71	212

- K-Nearest Neighbors:

	precision	recall	f1-score	support
NoTask	0.41	0.65	0.51	60
Task	0.82	0.64	0.72	152
accuracy	0.64	212		
macroavg	0.62	0.64	0.61	212
weightedavg	0.71	0.64	0.66	212

- Multilayer Perceptron:

	precision	recall	f1-score	support
NoTask	0.32	0.67	0.43	60
Task	0.77	0.43	0.55	152
accuracy	0.5	212		
macroavg	0.54	0.55	0.49	212
weightedavg	0.64	0.5	0.52	212

- Random Forest:

	precision	recall	f1-score	support
NoTask	0.38	0.6	0.47	60
Task	0.8	0.62	0.7	152
accuracy	0.61	212		
macroavg	0.59	0.61	0.58	212
weightedavg	0.68	0.61	0.63	212

- Gradient Boosting:

	precision	recall	f1-score	support
NoTask	0.31	0.48	0.38	60
Task	0.74	0.57	0.64	152
accuracy	0.55	212		
macroavg	0.52	0.53	0.51	212
weightedavg	0.62	0.55	0.57	212

### Clasificador Memoria vs. Lectura vs. Cálculo

Es un problema de clasificación multiclase en el cual se busca clasificar el tipo de tarea cognitiva que está realizando el sujeto.

A continuación, se muestran las tablas de resultados de las muestras de Michel Bureau:

- Support Vector:

	precision	recall	f1-score	support
Recordar	0.4	0.47	0.43	43
Lectura	0.89	0.74	0.81	42
Calculo	0.47	0.47	0.47	45
accuracy	0.55	130		
macroavg	0.58	0.56	0.57	130
weightedavg	0.58	0.55	0.56	130

- K-Nearest Neighbors:

	precision	recall	f1-score	support
Recordar	0.46	0.51	0.48	43
Lectura	0.72	0.79	0.75	42
Calculo	0.61	0.49	0.54	45
accuracy	0.59	130		
macroavg	0.6	0.6	0.59	130
weightedavg	0.59	0.59	0.59	130

- Multilayer Perceptron:

	precision	recall	f1-score	support
Recordar	0.5	0.09	0.16	43
Lectura	0.44	0.83	0.57	42
Calculo	0.48	0.44	0.46	45
accuracy	0.45	130		
macroavg	0.47	0.46	0.4	130
weightedavg	0.47	0.45	0.4	130

- Random Forest:

	precision	recall	f1-score	support
Recordar	0.47	0.37	0.42	43
Lectura	0.71	0.86	0.77	42
Calculo	0.51	0.51	0.51	45
accuracy	0.58	130		
macroavg	0.56	0.58	0.57	130
weightedavg	0.56	0.58	0.56	130

- Gradient Boosting:

	precision	recall	f1-score	support
Recordar	0.39	0.42	0.4	43
Lectura	0.71	0.76	0.74	42
Calculo	0.44	0.38	0.4	45
accuracy	0.52	130		
macroavg	0.51	0.52	0.51	130
weightedavg	0.51	0.52	0.51	130

A continuación, se muestran las tablas de resultados de las muestras de José Yamuni:

- Support Vector:

	precision	recall	f1-score	support
Recordar	0.59	0.72	0.65	53
Lectura	0.72	0.6	0.65	52
Calculo	0.44	0.43	0.43	47
accuracy	0.59	152		
macroavg	0.59	0.58	0.58	152
weightedavg	0.59	0.59	0.58	152

- K-Nearest Neighbors:

	precision	recall	f1-score	support
Recordar	0.54	0.64	0.59	53
Lectura	0.7	0.67	0.69	52
Calculo	0.31	0.26	0.28	47
accuracy	0.53	152		
macroavg	0.52	0.52	0.52	152
weightedavg	0.52	0.53	0.53	152

- Multilayer Perceptron:

	precision	recall	f1-score	support
Recordar	0.36	0.49	0.42	53
Lectura	0.54	0.29	0.37	52
Calculo	0.33	0.36	0.34	47
accuracy	0.38	152		
macroavg	0.41	0.38	0.38	152
weightedavg	0.41	0.38	0.38	152

- Random Forest:

	precision	recall	f1-score	support
Recordar	0.66	0.7	0.68	53
Lectura	0.73	0.67	0.7	52
Calculo	0.48	0.49	0.48	47
accuracy	0.62	152		
macroavg	0.62	0.62	0.62	152
weightedavg	0.63	0.62	0.63	152

- Gradient Boosting:

	precision	recall	f1-score	support
Recordar	0.6	0.53	0.56	53
Lectura	0.76	0.67	0.71	52
Calculo	0.42	0.53	0.47	47
accuracy	0.58	152		
macroavg	0.59	0.58	0.58	152
weightedavg	0.6	0.58	0.59	152

## Análisis de resultados

### P300 vs. No P300

Los resultados en general de las muestras de Michel Bureau fueron mejores debido a que los datos eran de mejor calidad. El modelo dio un mejor rendimiento en general fue **Support Vector** y el peor fue k-Nearest Neighbors. El mejor clasificador fue **Random Forest** con las muestras de Michel Bureau.

### Tarea Cognitiva vs. No Tarea Cognitiva

Los resultados en general de este experimento fueron peores que los del experimento anterior. El modelo dio un mejor rendimiento en general fue de nuevo **Support Vector** y los peores fueron Multilayer Perceptron y Gradient Boosting. El mejor clasificador fue **Support Vector** con las muestras de José Yamuni.

### Memoria vs. Lectura vs. Cálculo

En los resultados de las muestras de Michel Bureau se observa que clasifica muy bien la tarea de lectura y muy mal las otras, mientras que en los de José Yamuni clasifica bien las tareas de memoria y lectura y mal las de cálculo. El mejor clasificador fue **Random Forest** con las muestras de José Yamuni.

### Selección de características

A partir de un ejemplo de Gradient Boosting Classifier con las muestras de Michel Bureau del experimento P300 vs No P300, se obtienen las características más relevantes basadas en el parámetro de `feature_importances_`, obtenemos las n características que nos den el 80% de importancia, luego se utilizan estas características como los datos que entrenarán al modelo para hacer la validación cruzada, y obtenemos los resultados del modelo usando las n

características más relevantes. El modelo presentó una mejora de **dos centésimas** en la precisión, y **cinco centésimas** entre las sensibilidades de ambas clases.

### Aprendizaje por transferencia

En esta parte del proyecto se probó entrenar dos clasificadores del experimento P300 vs. No P300 con las muestras del sujeto Michel Bureau y evaluarlos con las muestras de José Yamuni. Se utilizaron los clasificadores que mejor rendimiento obtuvieron con las muestras de Michel Bureau en el experimento P300 vs. No P300.

A continuación, se muestran las tablas de resultados:

- Support vector: C = 10, Función kernel = radial

	precision	recall	f1-score	support
P300	0.15	0.26	0.19	269
NoP300	0.77	0.63	0.69	1079
accuracy	0.55	1348		
macroavg	0.46	0.44	0.44	1348
weightedavg	0.65	0.55	0.59	1348

- Random Forest: Número de árboles = 450

	precision	recall	f1-score	support
P300	0.12	0.2	0.15	269
NoP300	0.76	0.65	0.7	1079
accuracy	0.56	1348		
macroavg	0.44	0.42	0.43	1348
weightedavg	0.64	0.56	0.59	1348

Como se puede observar, los resultados que ambos clasificadores dan son bastante malos. Esto puede ser debido a que las condiciones en las que se realizó la adquisición de datos no fueron adecuadas, o a que los sujetos no hayan entendido bien lo que debían realizar en el experimento. Debido a que estamos analizando señales EEG, estas se pueden ver afectadas por factores externos al sujeto como interferencias o ruido ambiental. Incluso otros factores como fatiga o estado emocional pudieron haber repercutido negativamente en la adquisición de datos.

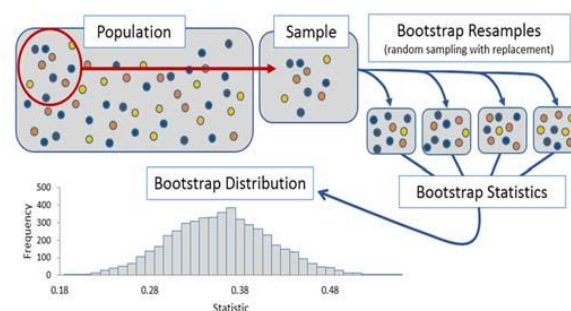


## Explicación detallada de la teoría detrás del algoritmo Random Forest

Para comprender como funciona el algoritmo Random Forest, es necesario saber primero qué es un árbol de decisión. Esto debido a que Random Forest no es más que un ensamble de árboles de decisión independientes.

Un árbol de decisión es una estructura jerárquica en la cual cada nodo representa una prueba sobre un atributo o característica, cada rama representa el resultado de la prueba, y cada nodo hoja representa una etiqueta o resultado.

Para entrenar el ensamble, a cada árbol se le asigna una muestra aleatoria del conjunto de datos para ser entrenado. La selección de muestras se hace a través de un método de remuestreo conocido como Bootstrap.



*Figura 3. Proceso ilustrado de remuestreo Bootstrap*

Los árboles se construyen utilizando un algoritmo, un ejemplo común es el CART (Árboles de Regresión y Clasificación). Esto implica que para cada árbol se aplicará un muestreo de características. Esto quiere decir que solo se utilizará un subconjunto del vector de características. Esto con el objetivo de que haya mayor diferenciación entre árboles y se reduzca la correlación entre estos.

Una vez contruidos los árboles de decisión, las predicciones se realizan a través de un consenso, el método por el que se llega al consenso varía dependiendo del tipo de problema. Para un problema de clasificación binaria se llega por voto mayoritario, cada árbol realiza una predicción y aquella que tenga más votos es elegida como la definitiva. Para un problema de clasificación multiclase cada árbol realiza una votación ponderada, ya que cada árbol determina una probabilidad para cada clase y se realiza un voto final. Cabe resaltar que Random Forest asigna un nivel de importancia a cada característica de la siguiente manera. En cada uno de los árboles, la importancia se calcula midiendo la reducción de impureza, es decir, a cuantas clases reduce el nodo la característica en cuestión. Una medida muy utilizada para este algoritmo en particular es la ganancia de Gini.

### *Ganancia de Gini para dos clases*

$$1 - (p_1)^2 - (p_2)^2$$

$p_1$  = probabilidad de la clase 1

$p_2$  = probabilidad de la clase 2

Posteriormente para el ensamble en conjunto, la importancia de las características se evalúa permutando aleatoriamente los valores de cada característica. Una vez permutados los valores de una característica, se compara el rendimiento del modelo original y el permutado. Si los resultados no son muy diferentes, entonces se concluye que la característica en cuestión no es muy importante.

## **Conclusiones**

Los resultados presentados en este trabajo son bastante malos. Para obtener mejores resultados en este trabajo se deben mejorar las condiciones de los experimentos. Incluso se podría mejorar entrenando clasificadores más complejos y haciendo mejor la selección de hiperparámetros e incluso realizando reducción de dimensionalidad a los conjuntos de datos. Sin embargo, como se acaba de decir, el mayor problema está en la adquisición de datos. El monitoreo de la actividad cerebral humana es vital para el avance tecnológico en el área de interfaces cerebro-computadora, permitiendo a personas con discapacidades motoras comunicarse utilizando señales cerebrales. Además de tener otras áreas muy importantes de aplicación como el diagnóstico y tratamiento de trastornos neurológicos. El conocimiento sobre la electroencefalografía y el potencial que tiene si se hace uso de aprendizaje automático es algo extremadamente valioso de comprender.

## Bibliografía

- Antelis, J., Mendoza, O., Chailloux, J. (2020). *Single-Option P300-BCI Performance Is Affected by Visual Stimulation Conditions*. Sensors.
- Coffman C., Pontifex, M. (2023). Validation of the g.tec Unicorn Hybrid Black wireless EEG system. Wiley Online Library. Recuperado de: <https://onlinelibrary.wiley.com/doi/full/10.1111/psyp.14320>
- Institutos Nacionales de Salud (NIH). (s.f.). Pruebas y procedimientos de diagnóstico neurológico. Recuperado de: <https://espanol.ninds.nih.gov/es/trastornos/forma-larga/pruebas-y-procedimientos-de-diagnostico-neurologico>
- Scikit-Learn. (2023). Supervised learning. Recuperado de: [https://scikit-learn.org/stable/supervised\\_learning.html](https://scikit-learn.org/stable/supervised_learning.html)
- Joshi, P. (2017). Artificial Intelligence with Python. Packt. pp. 63-79.