

Instituto Tecnológico de Monterrey
Campus Guadalajara

ANÁLISIS DE DATOS DE ACTIVIDAD CEREBRAL

Modelación del aprendizaje con inteligencia artificial
Grupo 301

Alejandra Velasco Zárate A01635453

Junio 2023

1. Introducción y objetivo

El análisis de datos de actividad cerebral y la inteligencia artificial son dos campos científicos y tecnológicos que han transformado la forma en que comprendemos y utilizamos la información relacionada con el cerebro humano. El análisis de datos de actividad cerebral se enfoca en el estudio de los patrones de actividad neuronal registrados mediante diversas técnicas de medición, mientras que la inteligencia artificial se centra en el desarrollo de algoritmos y modelos capaces de emular el pensamiento humano.

2. Objetivo

Realizar una serie de experimentos de neuropsicología para adquirir datos cerebrales. Construir con esos datos modelos de clasificación, para comprender las dificultades que hay en utilizar modelos de aprendizaje computacional para tratar datos biomédicos.

3. ¿Qué es la electroencefalografía?

Un electroencefalograma (EEG) es un examen que evalúa la actividad eléctrica de las células nerviosas (neuronas) que están situadas en la corteza cerebral (tejido nervioso que recubre el cerebro). Esta prueba permite detectar la actividad cerebral de las neuronas de los dos hemisferios cerebrales y de sus cuatro lóbulos (frontal, temporal, parietal y occipital). Las neuronas cerebrales se comunican a través de impulsos eléctricos y están siempre activas, tanto si la persona está despierta como dormida. Esta actividad eléctrica se manifiesta en forma de ondas de distinta intensidad: baja, moderada y alta (Clínic Barcelona, 2023).

4. ¿Cómo se interpretan las señales de EEG?

Las señales EEG generalmente se detallan en términos de tipos de actividad que son rítmicas y transitorias. Las actividades rítmicas se dividen en ciertas bandas de frecuencia, las cuales se ha demostrado que tienen cierta importancia biológica o cierta distribución sobre el cuero cabelludo. Las bandas de las señales EEG son nombradas con cierta nomenclatura; según su oscilación (las que caen debajo de 1Hz y por encima de 20Hz son consideradas como artefactos).







| Tipo | Rango de frecuencia (Hz) | Forma de onda | Propiedades | Actividad mental |
|-------------------|--------------------------|--|---|--|
| Delta(Δ) | 1-3 |  | Frontalmente en adultos, posteriormente en niños; ondas de alta amplitud | Sueño de onda lenta (profundo) en bebés |
| Theta(Θ) | 4-7 |  | Línea media frontal (Fz a Cz) | Al ralentí, inconsciente, meditación y somnolencia |
| Alpha(α) | 7-12 |  | Regiones posteriores de la cabeza, ambos lados, mayor en amplitud en el lado dominante. | Relajación y concentración |
| Mu(μ) | 8-13 |  | Corteza Sensorimotora | La supresión indica que las neuronas motoras están funcionando |
| Beta(β) | 12-30 |  | Corteza sensoriomotora, entre C3 y C4, distribución simétrica, más evidente frontalmente ondas de baja amplitud | Alerta, pensamiento y concentración activa |
| Gamma(γ) | >30 |  | Corteza somatosensorial | Procesamiento somatosensorial se muestra durante la coincidencia de memoria a corto plazo de objetos reconocidos, sonidos o sensaciones táctiles |

Figura 1: Señales cerebrales

En la Figura 1, se comparan todas las bandas de EEG en términos de frecuencia, forma, propiedades y actividades mentales. Como se indica en la figura, hay seis tipos de señales que son Delta(δ), Theta(Θ), Alpha(α), Mu(μ), Beta(β) y Gamma(γ). Delta (δ) hace referencia al estado de relajación mental y Gamma (γ) de mucha actividad cerebral (Olvera, 2020).

5. Descripción de los datos (señales) obtenidos en el experimento

- Equipo: Unicorn Hybrid Black desarrollado por GTec
- Canales: 8
 1. Fz
 2. C3
 3. Cz
 4. C4

5. Pz

6. PO7

7. Oz

8. PO8

- Velocidad de muestreo: 250 muestras por segundo.
- Valores típicos: 100 Hz - 500 Hz
- Referencia de los potenciales: debajo de los lóbulos de las orejas.

6. Descripción de los experimentos

6.1. P300

En este experimento, el participante se sentaba frente a un ordenador donde había una interfaz gráfica de usuario (GUI), mostraba los estímulos (distribuidos uniformemente en la pantalla) y un cuadro de instrucciones que guiaba a los participantes en la ejecución del experimento. En la figura 2, se muestra la interfaz y el usuario interactuando.



Figura 2: Montaje experimental con un participante

En la Figura 2, se muestra cómo se divide el experimento de P300 en bloques por segundos. La duración de este bloque es de unos 40 s, y se repiten varios bloques consecutivos hasta que el estímulo objetivo parpadee al menos 280 veces. El ejercicio se repetía 7 veces, en cada ejercicio se tenía que contar 40 caritas felices, lo que al final quedaría un total de 280 estímulos de P300.

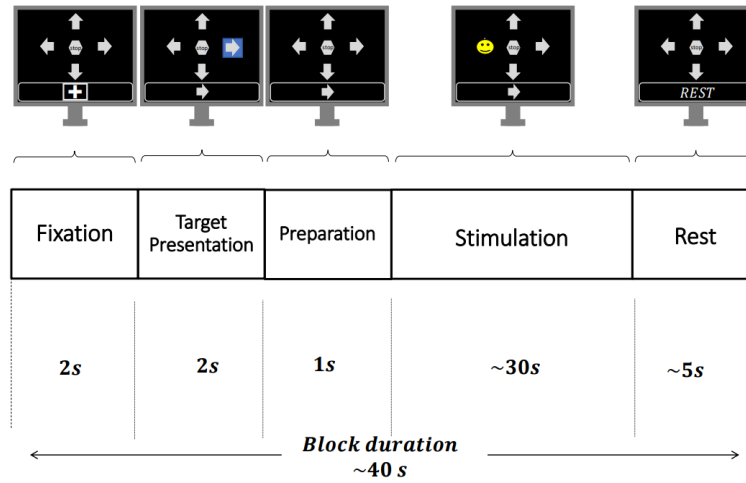


Figura 3: Fases del experimento

6.2. Tarea cognitiva

Este experimento constaba de 3 diferentes tipos de tareas cognitivas: imaginar nombres o cosas, realizar operaciones de aritmética y lectura. De cada tarea cognitiva se realizaban 4 pruebas, además de una prueba complementaria que trataba de ver un punto en la pantalla fijamente. En total el experimento trataría de 13 actividades. En la Figura 4 se puede observar la primera tarea cognitiva, sobre imaginar cosas, en la Figura 5 la tarea cognitiva de leer y la tarea cognitiva de matemáticas se visualiza en la Figura 6.



Figura 4: Tarea cognitiva 1

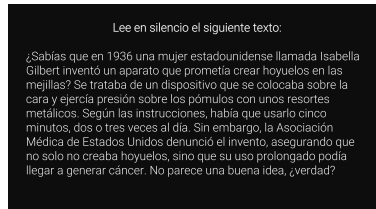


Figura 5: Tarea cognitiva 2

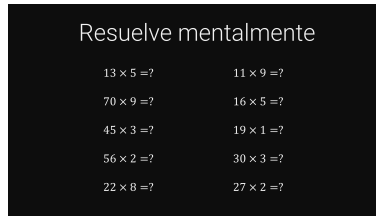


Figura 6: Tarea cognitiva 3

7. Extracción de características

7.1. P300

Se sabe que este experimento trata de ver si hay o no la onda cerebral P300, cuando la imagen de cara feliz aparece en el simulador. ¿Pero cómo se extraen los datos para poder comprobar esto? Después de que aparece la cara feliz, se toman 800 milisegundos de la actividad cerebral, es decir, 200 muestras (si por cada segundo se toman 250 muestras, por 800ms son 200). Estas 200 muestras por canal (o señal cerebral) pasan por un filtrado pasabanda, donde las muestras debajo de 4Hz y arriba de 20Hz no se toman en cuenta. Después pasan por un submuestreo que divide las muestras entre 4, dejando un total de 50 muestras por canal. Por último, pasa por un filtrado espacial, conocido como Análisis de Correlación Canónica (CCA), que transforma y combina los datos de una manera que solo existan 4 canales (señales cerebrales) de 50 muestras.

7.2. Tarea cognitiva

La tarea cognitiva es un experimento que consta de dos partes: si hay o no actividad cognitiva y cuál actividad cognitiva se está realizando. Para este experimento se tiene una cruz en la pantalla por 10 segundos, una tarea cognitiva por 15 segundos y descanso de 5 segundos. Por cada segundo hay una ventana de 250 muestras, por lo que, en el tiempo de la cruz hay 10 ventanas de 250 muestras, en la tarea cognitiva hay 15 ventanas y en el descanso no hay ventanas, es decir, no se recolectan datos. Por cada canal se hace un análisis de potencia espectral (PSD), que es técnica que permite descomponer una señal o serie de

tiempo (luminosa, sonora, sensorial, económica, etc) en componentes senoidales de diferentes frecuencias y amplitudes.

8. Clasificador XGBOOST

XGBOOST (Extreme Gradient Boosting) es un modelo de aprendizaje automático para problemas de aprendizaje supervisado donde se utilizan datos de entrenamiento para predecir la variable objetivo. Este clasificador es un modelo que permite predecir la clase a la que pertenece cada observación, basándose en variables explicativas que pueden ser cuantitativas y/o cualitativas (".^Extrême Gradient Boosting", 2023).

Este algoritmo se caracteriza por obtener buenos resultados de predicción con relativamente poco esfuerzo, en muchos casos equiparables o mejores que los devueltos por modelos más complejos computacionalmente, en particular para problemas con datos heterogéneos. La idea detrás del boosting es generar múltiples modelos de predicción "malos o débiles" secuencialmente, y que cada uno de estos tome los resultados del modelo anterior, para generar un modelo más "bueno o fuerte", con mejor poder predictivo y mayor estabilidad en sus resultados (Mendoza, 2020). Esto lo consigue utilizando un algoritmo de optimización llamado Descenso de gradiente (Gradient Descent), que busca encontrar el mínimo de una función mediante la iteración de ajustes en la dirección opuesta del gradiente, es decir, en la dirección de mayor descenso de la función.

Durante el entrenamiento, los parámetros de cada modelo débil son ajustados iterativamente tratando de encontrar el mínimo de una función objetivo, que puede ser la proporción de error en la clasificación, el área bajo la curva (AUC), la raíz del error cuadrático medio (RMSE) o algún otro método. Cada modelo es comparado con el anterior y si un nuevo modelo tiene mejores resultados, entonces se toma este como base para realizar modificaciones. Si, por el contrario, tiene peores resultados, se regresa al mejor modelo anterior y se modifica ese de una manera diferente ("XGBoost Tree, nodo", 2021).

Una parte importante de este clasificador es que el usuario tiene que especificar el hiperparámetros que establece que tan grandes son los ajustes de un modelo a otro. Este proceso se repite hasta llegar a un punto en el que la diferencia entre modelos consecutivos es insignificante, lo cual nos indica que hemos encontrado el mejor modelo posible, o cuando se llega al número de iteraciones máximas definido por el usuario (Mendoza, 2020). La principal razón para utilizar XGBoost es su capacidad para producir modelos altamente precisos y robustos, especialmente en conjuntos de datos complejos con características no lineales y de alta dimensionalidad. Además, XGBoost ofrece un control avanzado sobre los hiperparámetros, lo que permite ajustar el modelo de manera óptima para obtener los mejores resultados. Con su capacidad para manejar características faltantes y manejar automáticamente la selección de características,

XGBoost es una herramienta poderosa para resolver problemas de clasificación y regresión en el campo del aprendizaje automático.

9. Resultados

9.1. Evaluación de algoritmos de clasificación

En esta parte del trabajo se evaluaron 5 clasificadores diferentes por cada base de datos con cross validation k-folds, donde $k = 5$. Estos clasificadores son:

- Support Vector Machine (SVM)
- K-Nearest Neighbor(KNN)
- Multilayer perceptron (MLP) de 5 capas.
- Decision Tree classifier
- XGBOOST classifier

Después se evaluaron hiperparámetros de algunos clasificadores para así poder observar cuál es el valor óptimo de estos mismo. De los 5 clasificadores lo hiperparámetros son:

- **SVM:** C es el parámetro que controla la relación el tamaño del margen (valor grande de C ocasiona un margen pequeño, y un valor grande de C conduce a un margen grande).
- **KNN:** k representa el número de vecinos más cercanos que se utilizan para la clasificación. Seleccionar un valor adecuado para k es importante, ya que un valor bajo puede llevar a un modelo demasiado sensible al ruido y un valor alto puede llevar a un modelo demasiado generalizado.
- **MLP:** *hidden_layer_sizes* especifica la arquitectura de la red neuronal, es decir, el número de neuronas en cada capa oculta.
- **Decision Tree Classifier:** *max_depth* determina la profundidad máxima del árbol de decisión. *criterion* es el encargado de medir la uniformidad de los nodos, uniformidad quiere decir que las cosas que son similares deben estar juntas y las que son diferentes deben separarse y distinguirse claramente unas de otras. se puede elegir *gini*, *entropy*, *log_loss* para clasificación. *min_samples_split* es el número mínimo de datos requeridos por un nodo para realizar una división.
- **XGBoost Classifier:** *n_estimators* representa el número de árboles que se utilizarán en el modelo. Cuanto mayor sea este valor, más complejo será el modelo y más tiempo requerirá el entrenamiento. Sin embargo, un número demasiado alto puede llevar a un sobreajuste. *booster* es el tipo de modelo de clasificación usado, por defecto gbtrees. *max_depth* es la 'profundidad' o número de nodos de bifurcación de los árboles de decisión usados en el entrenamiento.

Por último, se aplica un método de Wrapper feature selection para cada base de datos del experimento P300, llamado **Sequential Feature Selector**, que añade (selección hacia delante) o elimina (selección hacia atrás) características para formar un subconjunto de características de forma codiciosa. En cada etapa, este estimador elige la mejor característica para añadir o eliminar en función de la puntuación de validación cruzada de un estimador (scikit-learn, 2023). Este método de reducción de dimensionalidad se hizo con el fin de ver cuántas características son suficientes para obtener un clasificador óptimo de SVM base radial. En cada clasificador, se evalúa este método de Wrapper desde $k = 1$ hasta $k = 10$, donde k representa la cantidad de características seleccionadas para el modelo.

Para las bases de datos de los experimentos cognitivos, se utilizará el método de filtrado **Select K Best**, que selecciona las características según las k puntuaciones más altas. Este método de reducción de dimensionalidad se hizo con el fin de ver cuántas características son suficientes para obtener un clasificador óptimo de SVM base radial. En cada clasificador, se evalúa este método de filtrado desde $k = 1$ hasta $k = 10$, donde k representa la cantidad de características seleccionadas para el modelo.

9.1.1. Clasificación de datos P300

En el experimento P300, se recolectaron dos bases de datos, una del sujeto identificado como Laura y otra del sujeto identificado como Frank. En ambas bases de datos, se encontraba un desbalance en los datos, esto significa que la distribución de las clases objetivo en el conjunto de datos no es uniforme, es decir, hay una discrepancia significativa en el número de ejemplos pertenecientes a cada clase. En este caso contamos con dos clases: 1 y 2, en donde en ambas bases de datos había casi 4 veces más observaciones de la clase 2 que de la clase 1. Esto podría llevar a que el modelo de clasificación esté sesgado y en la mayoría de los casos clasifique por default una nueva observación como clase 2, además que las métricas de evaluación de rendimiento *recall*, *precisión* y *F1-Score* de la clase 1 tienen valores por debajo del .5, lo cual no es bueno en un modelo de clasificación. Por esta razón, se balancearon los datos reduciendo el número de observaciones de la clase 2, si bien esta reducción significa una disminución en el *accuracy* del clasificador con los datos obtenidos, representa una mejora para el clasificador cuando sea utilizado con nuevas observaciones de nuevos sujetos.

Datos Laura

El mejor modelo de clasificación para los datos de Laura fue el **SVM base radial** con unas métricas de evaluación:

| Clase | Precisión | Recall | Puntaje F1 | Soporte |
|--------------|--------------------|--------------------|--------------------|---------|
| 1 | 0.7969348659003831 | 0.7938931297709924 | 0.7954110898661567 | 262 |
| 2 | 0.7946768060836502 | 0.7977099236641222 | 0.7961904761904763 | 262 |
| macro avg | 0.7958058359920166 | 0.7958015267175573 | 0.7958007830283165 | 524 |
| weighted avg | 0.7958058359920166 | 0.7958015267175572 | 0.7958007830283165 | 524 |

Accuracy = 0.7958015267175572

Figura 7: Mejor clasificador Laura P300

En la Figura 7, se observa que el *recall*, *precisión* y *F1-Score* de las clase 1 y 2 están arriba de 0,80, lo cuál, en un modelo de bases de datos de fenómenos de la vida real, es muy bueno. Complementando la exactitud del modelo que fue de 0,7958.

En este caso, el hiperparámetro que se evaluó fue en el clasificador **KNN**, donde se evaluó este modelo con $k = 10$ hasta $k = 20$. En este caso, se observó que el óptima valor de k es 16, ya que la exactitud del modelo fue de 0,7557 y las demás métricas de evaluación de rendimiento por cada clase está arriba de 0,70.

Finalmente, utilizando el método wrapper antes mencionado se visualizó que la mejor cantidad de características en el modelo de clasificación SVM es de 8, ya que tiene un accuracy de 0,7557 con las métricas de evaluación por clase arriba de 0,70. En la Figura 8, se puede observar la gráfica del método, donde el eje y representa la exactitud del modelo y el eje x el número de características.

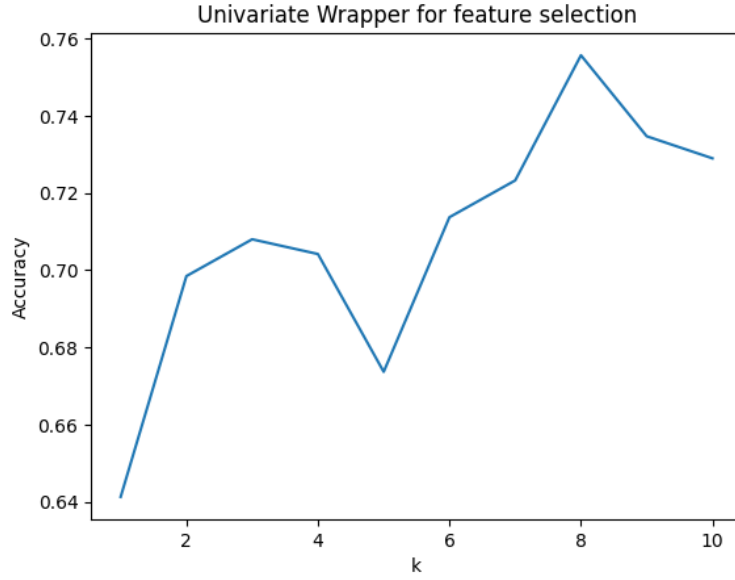


Figura 8: Wrapper Laura

Datos Frank

El mejor modelo de clasificación para los datos de Frank también fue el **SVM base radial** con unas métricas de evaluación:

| Clase | Precisión | Recall | Puntaje F1 | Soporte |
|--------------|--------------------|--------------------|--------------------|---------|
| 1 | 0.7683823529411765 | 0.7545126353790613 | 0.761384335154827 | 277 |
| 2 | 0.7588652482269503 | 0.7725631768953068 | 0.7656529516994633 | 277 |
| macro avg | 0.7636238005840634 | 0.7635379061371841 | 0.7635186434271451 | 554 |
| weighted avg | 0.7636238005840634 | 0.7635379061371841 | 0.7635186434271451 | 554 |

Accuracy = 0.7635379061371841

Figura 9: Mejor clasificador Frank P300

En la Figura 9, se observa que el *recall*, *precisión* y *F1-Score* de las clase 1 y 2 están arriba de 0,75, lo cuál, en un modelo de bases de datos de fenómenos de la vida real, es muy bueno. Complementando la exactitud del modelo que fue de 0,7635.

Para el caso de Frank, el hiperparámetro evaluado fue del clasificador **Decision Tree Classifier**, *criterion*. Para *criterion* = *gini*, la exactitud fue de 0,6353, para *criterion* = *entropy*, la exactitud fue de 0,6263 y para *criterion* = *log-loss*, la exactitud fue de 0,6119.

Finalmente, para el caso de Frank, a comparación de Laura, la cantidad de características para obtener el rendimiento optimo del clasificador SVM es de 9, con una exactitud de 0,7202, aunque el clasificador inicial que utiliza todas las características, tiene mejor rendimiento, por lo cual, se llega a la conclusión que es mejor utilizar mas de 9 características. En la Figura 10, se observa de una manera más visual dicho análisis.

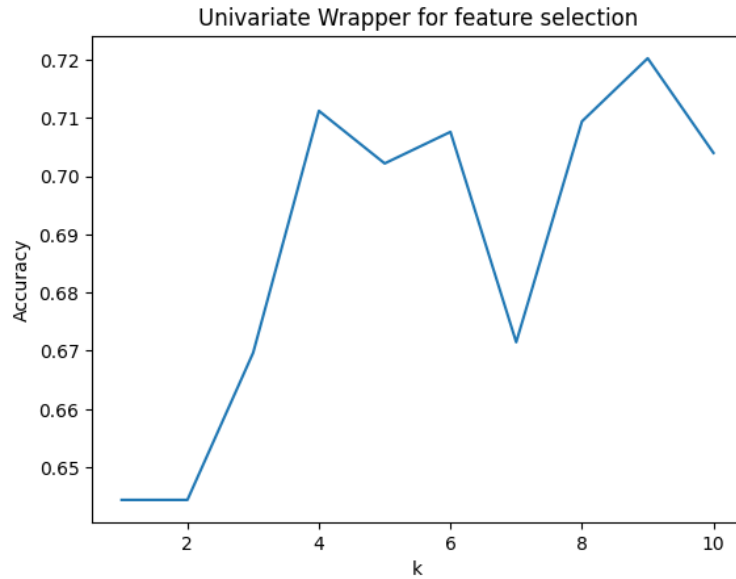


Figura 10: Wrapper Frank

9.1.2. Clasificación de datos Cognitivo (tipo de tarea)

La idea principal de esta parte del proyecto, era contar con 3 archivos del experimento cognitivo: uno de Laura, uno de Frank y uno de Ale. Sin embargo, hubo un problema con la conexión del material y los datos del experimento cognitivo de Ale no se obtuvieron. Por lo que solo se harán los análisis con los datos de Laura y de Frank. En la naturalidad de los datos, las clases eran etiquetas del 1 al 13, donde se agruparon las actividades cognitivas del 1 al 4 como 1, del 5 al 8 como 2, del 9 al 12 como 3 y la actividad 13 como 4.

Datos Cognitivos Laura

En este caso del experimento cognitivo, los datos de Laura obtuvieron como mejor modelo el clasificador **XGBoost Classifier** con una accuracy de 0,8782 y todas las métricas de evaluación por clase arriba de 0,78.

| Clase | Precisión | Recall | Puntaje F1 | Soporte |
|--------------|--------------------|--------------------|--------------------|---------|
| 1 | 0.8723404255319149 | 0.7884615384615384 | 0.8282828282828283 | 52 |
| 2 | 0.9074074074074074 | 0.9423076923076923 | 0.9245283018867925 | 52 |
| 3 | 0.8545454545454545 | 0.9038461538461539 | 0.8785046728971962 | 52 |
| macro avg | 0.8780977624949257 | 0.8782051282051282 | 0.8771052676889389 | 156 |
| weighted avg | 0.8780977624949257 | 0.8782051282051282 | 0.8771052676889389 | 156 |

Accuracy = 0.8782051282051282

Figura 11: Mejor clasificador Cognitivo Laura

Para este caso, se evaluó el hiperparámetro del clasificador KNN, el valor de k . Se observó el mejor valor para k es de 10, porque obtiene una exactitud de 0,8653 y métricas de evaluación por clase arriba de 0,70. Este es un modelo muy bueno y es aún mejor que el modelo KNN inicial que se evaluó con $k = 13$. Véase Figura 12 para observar los resultados.

Para $k = 10$ los resultados son:
Resultados del clasificador:

| Clase | Precisión | Recall | Puntaje F1 | Soporte |
|--------------|--------------------|--------------------|--------------------|---------|
| 1 | 0.9473684210526315 | 0.6923076923076923 | 0.7999999999999999 | 52 |
| 2 | 0.8909090909090909 | 0.9423076923076923 | 0.9158878504672897 | 52 |
| 3 | 0.7936507936507936 | 0.9615384615384616 | 0.8695652173913043 | 52 |
| macro avg | 0.877309435204172 | 0.8653846153846154 | 0.8618176892861981 | 156 |
| weighted avg | 0.8773094352041719 | 0.8653846153846154 | 0.861817689286198 | 156 |

Accuracy = 0.8653846153846154

Figura 12: Hiperparámetro de KNN, k

Finalmente, el método de filtrado para este grupo de datos evaluó que el óptimo número de características para obtener el mejor modelo es de 10, con una exactitud de 0,7821 y unas métricas de evaluación arriba de 0,65. Este modelo obtenido no es malo, pero podría mejorar cambiando el clasificador evaluado y aumentando características.

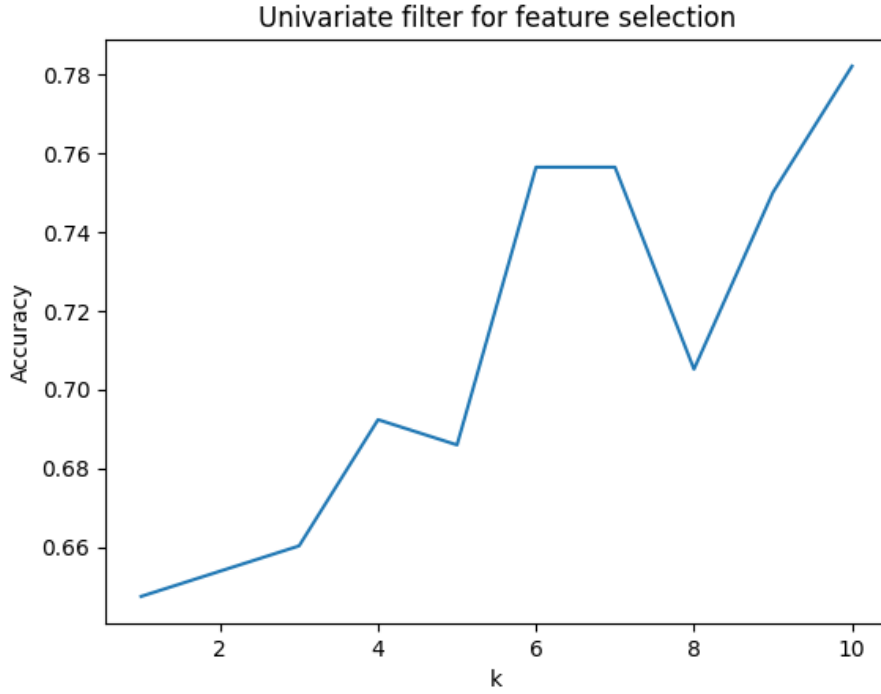


Figura 13: Filter Cognitivo Laura

Datos Cognitivos Frank

Con los datos cognitivos de Frank, el mejor modelo de clasificación fue **SVM base radial** con una exactitud de 0,8653. Además, todas las métricas de evaluación por clase tienen valores altos. Véase la Figura 14, para mejor visualización de los resultados.

| Clase | Precisión | Recall | Puntaje F1 | Soporte |
|--------------|--------------------|--------------------|--------------------|---------|
| 1 | 0.8809523809523809 | 0.7115384615384616 | 0.7872340425531914 | 52 |
| 2 | 0.9272727272727272 | 0.9807692307692307 | 0.9532710280373831 | 52 |
| 3 | 0.7966101694915254 | 0.9038461538461539 | 0.8468468468468469 | 52 |
| macro avg | 0.8682784259055446 | 0.8653846153846154 | 0.862450639145807 | 156 |
| weighted avg | 0.8682784259055446 | 0.8653846153846154 | 0.862450639145807 | 156 |

Accuracy = 0.8653846153846154

Figura 14: Mejor clasificador Cognitivo Frank

Para este conjunto de datos, el hiperparámetro evaluado fue C para el modelo de Support Vector Machine. Se empezó la evaluación con $C = 0,01$ y se evaluó

hasta $C = 1$. En este caso se observó que el mejor modelo es cuando $C = ,83$ con una exactitud de 0,8718 que es incluso mejor que los primeros resultados obtenidos en el clasificador SVM, en la Figura 14.

C = 0.83
Resultados del clasificador:

| Clase | Precisión | Recall | Puntaje F1 | Soporte |
|--------------|--------------------|--------------------|--------------------|---------|
| 1 | 0.8636363636363636 | 0.7307692307692307 | 0.7916666666666666 | 52 |
| 2 | 0.9444444444444444 | 0.9807692307692307 | 0.9622641509433962 | 52 |
| 3 | 0.8103448275862069 | 0.9038461538461539 | 0.8545454545454546 | 52 |
| macro avg | 0.8728085452223383 | 0.8717948717948718 | 0.8694920907185058 | 156 |
| weighted avg | 0.8728085452223383 | 0.8717948717948718 | 0.8694920907185059 | 156 |

Accuracy = 0.8717948717948718

Figura 15: Hiperparámetro C para SVM, óptimo resultado

Finalmente, para los datos cognitivos de Frank, la cantidad de características que obtiene un modelo óptimo con mejores resultados, es cuando se seleccionan 7 con una exactitud de 0,7564, pero este valor es menor que el clasificador inicial de SVM y del de óptimo hiperparámetro C . Esto conluye en que se necesitan más características de las evaluadas en este método de reducción de dimensionalidad filtrado.

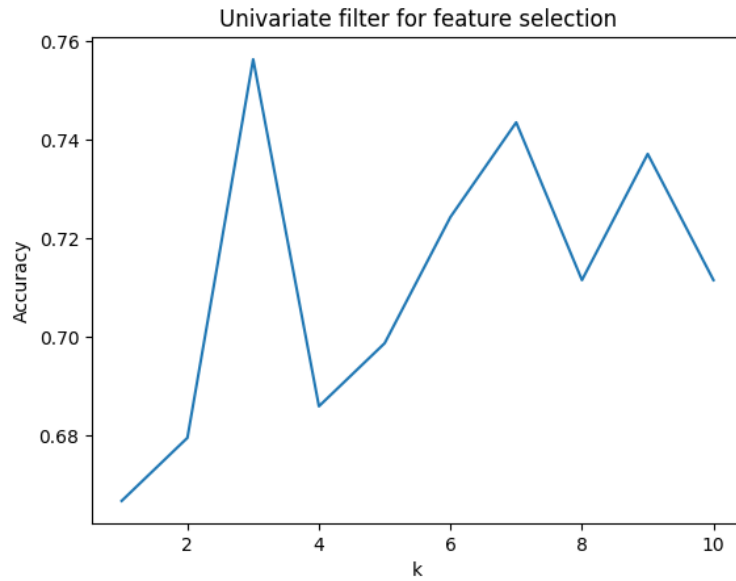


Figura 16: Filtrado de características

9.1.3. Clasificación de datos Cognitivo (tarea vs no tarea)

La idea principal de esta parte del proyecto, era contar con 3 archivos del experimento cognitivo: uno de Laura, uno de Frank y uno de Ale. Sin embargo, hubo un problema con la conexión del material y los datos del experimento cognitivo de Ale no se obtuvieron. Por lo que solo se harán los análisis con los datos de Laura y de Frank.

Datos Cognitivos Laura

Con los datos cognitivos de Laura, el mejor modelo de clasificación fue **SVM base radial** con una exactitud de 0,8182. Además, todas las métricas de evaluación por clase tienen valores altos. Véase la Figura 17, para mejor visualización de los resultados.

| Clase | Precisión | Recall | Puntaje F1 | Soporte |
|--------------|--------------------|--------------------|--------------------|---------|
| 1 | 0.8089887640449438 | 0.9230769230769231 | 0.8622754491017964 | 156 |
| 2 | 0.84 | 0.6494845360824743 | 0.7325581395348837 | 97 |
| macro avg | 0.8244943820224719 | 0.7862807295796987 | 0.7974167943183401 | 253 |
| weighted avg | 0.8208784473953014 | 0.8181818181818182 | 0.8125419351571699 | 253 |

Accuracy = 0.8181818181818182

Figura 17: Mejor clasificador Cognitivo Laura

Para este caso, se evaluó el hiperparámetro del clasificador KNN, el valor de k . Se observó el mejor valor para k es de 17, porque obtiene una exactitud de 0,7826 y métricas de evaluación por clase arriba de 0,70. Este es un modelo bueno, ya que son datos de un fenómeno de la vida diaria y es aún mejor que el modelo KNN inicial que se evaluó con $k = 13$. Véase Figura 18 para observar los resultados.

Para $k = 17$ los resultados son:
Resultados del clasificador:

| Clase | Precisión | Recall | Puntaje F1 | Soporte |
|--------------|--------------------|--------------------|--------------------|---------|
| 1 | 0.7821229050279329 | 0.8974358974358975 | 0.835820895522388 | 156 |
| 2 | 0.7837837837837838 | 0.5979381443298969 | 0.6783625730994152 | 97 |
| macro avg | 0.7829533444058583 | 0.7476870208828972 | 0.7570917343109016 | 253 |
| weighted avg | 0.7827596846299785 | 0.782608695652174 | 0.7754514991784024 | 253 |

Accuracy = 0.782608695652174

Figura 18: Hiperparámetro de KNN, k

Finalmente, para los datos cognitivos de Laura, la cantidad de características

que obtiene un modelo óptimo con mejores resultados, es cuando se seleccionan 7 con una exactitud de 0,7589, pero este valor es menor que el clasificador inicial de SVM. Esto conluye en que se necesitan más características de las evaluadas en este método de reducción de dimensionalidad filtrado.

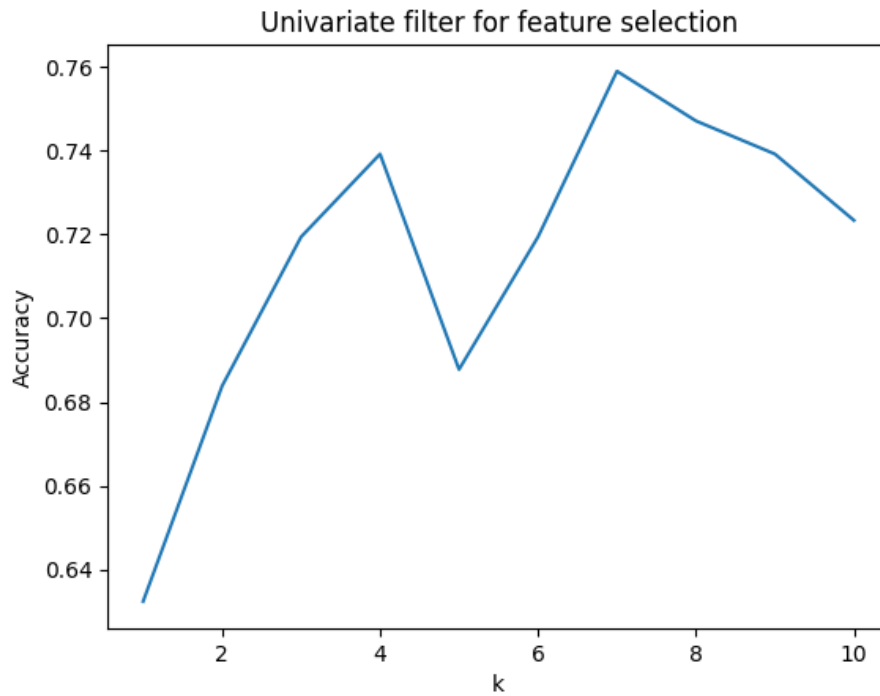


Figura 19: Hiperparámetro de KNN, $k=7$

Datos Cognitivos Frank

Con los datos cognitivos de Frank, el mejor modelo de clasificación fue **SVM base radial** con una exactitud de 0,8175. Además, todas las métricas de evaluación por clase tienen valores altos. Véase la Figura 20, para mejor visualización de los resultados.

| Clase | Precisión | Recall | Puntaje F1 | Soporte |
|--------------|--------------------|--------------------|--------------------|---------|
| 1 | 0.8089887640449438 | 0.9230769230769231 | 0.8622754491017964 | 156 |
| 2 | 0.8378378378378378 | 0.6458333333333334 | 0.7294117647058824 | 96 |
| macro avg | 0.8234133009413909 | 0.7844551282051282 | 0.7958436069038395 | 252 |
| weighted avg | 0.8199788873946178 | 0.8174603174603174 | 0.8116607121890673 | 252 |

Accuracy = 0.8174603174603174

Figura 20: Mejor clasificador Cognitivo Frank

Para este conjunto de datos, el hiperparámetro evaluado fue *booster* para el modelo de XGBoost. Este parámetro tiene 3 posibles valores: 'gbtree', 'gblinear', 'dart' y se evaluó con cada uno de ellos. En este caso se observó que el mejor modelo es cuando *booster* = *gbtree* con una exactitud de 0,8056 que es incluso mejor que los primeros resultados obtenidos en el clasificador XGBOOST, en la Figura 21.

Para booster gbtree:
Resultados del clasificador:

| Clase | Precisión | Recall | Puntaje F1 | Soporte |
|--------------|--------------------|--------------------|--------------------|---------|
| 1 | 0.8242424242424242 | 0.8717948717948718 | 0.8473520249221185 | 156 |
| 2 | 0.7701149425287356 | 0.6979166666666666 | 0.7322404371584699 | 96 |
| macro avg | 0.7971786833855798 | 0.7848557692307692 | 0.7897962310402942 | 252 |
| weighted avg | 0.803622431208638 | 0.8055555555555556 | 0.8034999914883475 | 252 |

Accuracy = 0.8055555555555556

Figura 21: Hiperparámetro booster para XGBOOST, óptimo resultado

Finalmente, para los datos cognitivos de Frank, la cantidad de características que obtiene un modelo óptimo con mejores resultados, es cuando se seleccionan 7 con una exactitud de 0,7659 y unas métricas de evaluación por clase arriba de 0,75.

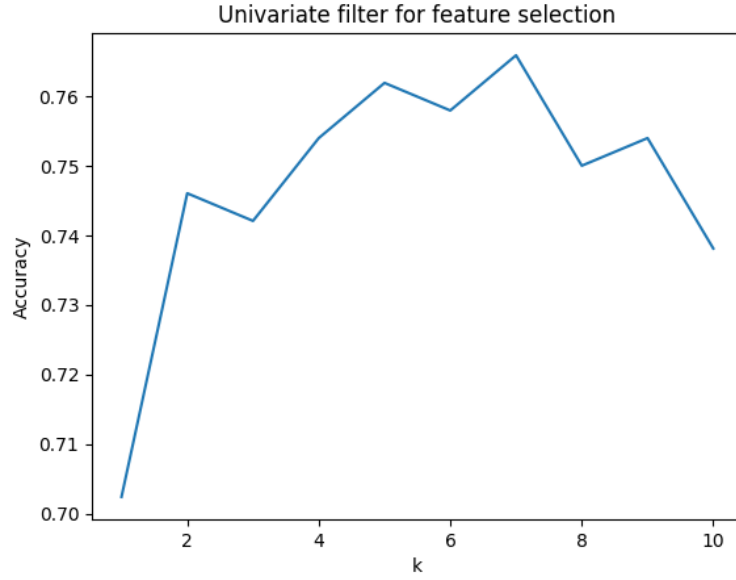


Figura 22: Filtrado de características

Después de analizar todos los clasificadores para todas las bases de datos obtenidas, se llega a la conclusión que el mejor modelo para clasificar las ondas cerebrales de una persona es el de Support Vector Machine (SVM) de base radial, ya que mantenía la óptima exactitud sin descuidar el *recall*, *precisión* y *F1-Score* por clase.

9.2. Aprendizaje por transferencia

Esta parte del proyecto, consiste en evaluar 3 clasificadores distintos, donde se entrene con un conjunto de datos del Sujeto Laura y se prueben con el conjunto de datos de Frank en el experimento P300.

- Support Vector Machine: *exactitud* = 0,7818

| Clase | Precisión | Recall | Puntaje F1 | Soporte |
|--------------|---------------------|---------------------|---------------------|---------|
| 1 | 0.34831460674157305 | 0.11191335740072202 | 0.16939890710382513 | 277 |
| 2 | 0.8113496932515337 | 0.9480286738351255 | 0.8743801652892562 | 1116 |
| macro avg | 0.5798321499965534 | 0.5299710156179237 | 0.5218895361965407 | 1393 |
| weighted avg | 0.7192745181163872 | 0.7817659727207465 | 0.7341936552265395 | 1393 |

Precisión del modelo SVM: 0.7817659727207465

Figura 23: SVM, entrenado con Laura probado con Frank

- MLP: *exactitud* = 0,7308

| Clase | Precisión | Recall | Puntaje F1 | Soporte |
|--------------|--------------------|---------------------|--------------------|---------|
| 1 | 0.28125 | 0.22743682310469315 | 0.251497005988024 | 277 |
| 2 | 0.8169375534644996 | 0.8557347670250897 | 0.8358862144420132 | 1116 |
| macro avg | 0.5490937767322498 | 0.5415857950648915 | 0.5436916102150186 | 1393 |
| weighted avg | 0.7104153335724204 | 0.7307968413496052 | 0.7196796022799493 | 1393 |

Precisión del modelo MLP: 0.7307968413496052

Figura 24: MLP, entrenado con Laura probado con Frank

- XGBOOST: $exactitud = 0,7753$

| Clase | Precisión | Recall | Puntaje F1 | Soporte |
|--------------|--------------------|---------------------|---------------------|---------|
| 1 | 0.3732394366197183 | 0.19133574007220217 | 0.25298329355608595 | 277 |
| 2 | 0.820943245403677 | 0.9202508960573477 | 0.8677651035065482 | 1116 |
| macro avg | 0.5970913410116977 | 0.5557933180647749 | 0.5603741985313171 | 1393 |
| weighted avg | 0.7319167163059336 | 0.7753050969131371 | 0.7455148799916321 | 1393 |

Precisión del modelo XGBOOST: 0.7753050969131371

Figura 25: XGBOOST, entrenado con Laura probado con Frank

¿Es posible entrenar un modelo de clasificación con los datos de un sujeto y obtener buenos resultados con los datos de otro sujeto? ¿Por qué?

Sí es posible entrenar un modelo de clasificación con los datos de un sujeto y probarlo con otro y obtener un buen resultado. En este caso el SVM Radial fue el mejor modelo con un accuracy de 0.78 que en la vida real es un muy buen valor para un modelo de clasificación. Además este modelo tiene un precisión y recall por la clase 2 muy buena. Esto es porque la cantidad de los datos que hay, muchos son observaciones de la clase dos y muy pocos de la clase 1. Eso nos dice que tenemos un desbalance en nuestro modelo, pero aun así se alcanza a obtener un buen clasificador.

¿Consideras que debes hacer ajustes cuando tengas datos de otro sujeto?

Si se tienen datos de otro sujeto se puede volver a entrenar con ambos datos y probar con los del sujeto restante, esto podría mejorar nuestro clasificador porque ya no solo se toma en cuenta los datos de un sujeto para entrenar el modelo, en su lugar se usan 2, lo que aportaría información valiosa al momento de entrenar el modelo.

Al observar los clasificadores, todos tienen una exactitud arriba de 0,70, lo cual es un modelo bueno cuando se trabaja con datos obtenidos de un experimentos que derivan a una hipótesis. Además, las métricas de evaluación por clase de cada clasificador son buenas, tomando en cuenta que estos datos de P300 no fueron balanceados. El mejor clasificador fue el de **Support Vector**

Machine base radial con una exactitud de 0,7818, casi 80 y es un buen clasificador. Esto se puede relacionar con la etapa de *Evaluación de Clasificadores*, porque en esa parte del proyecto se observa que la mayoría de los clasificadores que obtienen un óptimo modelo son los de **Support Vector Machine base radial**.

10. Conclusiones

Este proyecto tenía como principal objetivo aplicar la inteligencia artificial en el ámbito de salud, mediante diferentes experimentos donde se adquieren datos cerebrales de una persona para evaluar su funcionamiento. Esto se hizo mediante modelos clasificadores que pudieran predecir las actividades cerebrales adquiridas. Al analizar los modelos de clasificación de todos los datos adquiridos, se puede observar que para todos los casos, el mejor clasificador es el de Support Vector Machine, esto se debe a la naturalidad de los datos y de las señales cerebrales. Es observación solo fue posible mediante el análisis de los datos y de los temas relacionados a la inteligencia artificial.

El monitoreo de la actividad cerebral de una persona es beneficioso para su salud porque proporciona información valiosa sobre el funcionamiento del cerebro y puede ayudar en la detección temprana de problemas neurológicos, el seguimiento de enfermedades cerebrales y la personalización de tratamientos médicos. Otras aplicaciones del monitoreo de datos cerebrales que pueden ser beneficiosas para la salud, son la monitorización continua de signos vitales para la detección temprana de enfermedades, la detección de patrones anormales en electrocardiogramas para prevenir enfermedades cardíacas, el análisis de datos genéticos para identificar riesgos hereditarios y personalizar tratamientos, y la monitorización del sueño para mejorar la calidad y la salud del descanso. Estas aplicaciones pueden ayudar a prevenir enfermedades, optimizar tratamientos y promover un estilo de vida más saludable.

Por último, con estos dispositivos se podrían registrar variables como la frecuencia cardíaca, la temperatura corporal, la presión arterial, la actividad física y posiblemente otras señales biométricas como la saturación de oxígeno en sangre o el ritmo respiratorio. Trabajar en este proyecto de inteligencia artificial enfocado en las ondas cerebrales y la electroencefalografía fue fascinante debido a su potencial para aplicaciones clínicas en el diagnóstico de trastornos neurológicos, la complejidad de analizar datos cerebrales y la contribución al avance en nuestra comprensión del cerebro humano, lo que podría llevar a mejoras en la atención médica y el desarrollo de terapias innovadoras.

11. Referencias

Chailloux, J. D., Montoya, O. y Antelis, J. M. (2020). Single-Option P300-

BCI Performance Is Affected by Visual Stimulation Conditions.
Tecnologico de Monterrey, School of Engineering and Science, Monterrey.
[PDF]

Clínic Barcelona. (2021). *Universidad de Barcelona*. Recuperado de

<https://www.clinicbarcelona.org/asistencia/pruebas-y-procedimientos/electroencefalograma>

Extreme Gradient Boosting. (2023). *XLSTAT*. Recuperado de

<https://www.xlstat.com/es/soluciones/funciones/extreme-gradient-boosting-xgboost>

Mendoza, J. (2020). Tutorial: XGBoost en Python. *Medium*. Recuperado de

<https://medium.com/@jboscomendoza/tutorial-xgboost-en-python-53e48fc58f73>

Olvera, J. A. (2019). Análisis de espectrogramas de señales EEG.

Facultad de Ciencias de la Computación Universidad Autónoma de Puebla.
Recuperado de <https://repositorioinstitucional.buap.mx/bitstream/handle/20,500,12371/10364/20201117125934-2112-T.pdf>

sklearn.feature_selection.SequentialFeatureSelector. (2023). *Scikit-learn*.

Recuperado de https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SequentialFeatureSelector.html

XGBoost Tree, nodo. (2021). *IBM*. Recuperado de

<https://www.ibm.com/docs/es/spss-modeler/saas?topic=nodes-xgboost-tree-node>