Análisis y Clasificación del Estado del Ojo a partir de Señales EEG

Modelos y Simulación de Sistemas II

Andrés Felipe Giraldo Yusti
Dept. de Ingeniería en Sistemas
Universidad de Antioquia
Medellín
andres.giraldoy@udea.edu.co

John Edison Zapata Ramirez
Dept. de Ingeniería en Sistemas
Universidad de Antioquia
Medellín
john.zapata1@udea.edu.co

***Abstract*—Este trabajo presenta el desarrollo inicial de un sistema de clasificación basado en técnicas de aprendizaje de máquina para la detección automática del estado ocular (verificar si el ojo está abierto o cerrado) a partir de señales registradas durante un estudio de electroencefalogramas (EEG).**
Se utilizará la base de datos *EEG Eye State*, la cual contiene 14 canales de EEG registrados mediante el dispositivo Emotiv Neuroheadset durante 117 segundos continuos, con un total de 14 980 instancias. Cada muestra está asociada a una etiqueta binaria que indica el estado del ojo: abierto (0) o cerrado (1). En esta primera etapa del proyecto se realiza la descripción y análisis del problema y del conjunto de datos, identificando su estructura, variables y propiedades estadísticas, todo esto con el fin de diseñar modelos de clasificación de manera efectiva y eficiente. Los algoritmos previstos para su evaluación incluyen regresión logística, *k*-vecinos más cercanos (k-NN), árboles de decisión, máquinas de vectores de soporte (SVM) y redes neuronales artificiales. El objetivo final es comparar el desempeño de estos modelos para establecer la capacidad predictiva del sistema y su aplicabilidad en contextos de monitoreo de fatiga. Esto permitirá establecer una base para las siguientes etapas, como la reducción de la dimensión y la optimización del modelo.

I. Introducción

El análisis de señales cerebrales mediante electroencefalografía (EEG) ha adquirido gran relevancia en los últimos años debido a su capacidad para registrar la actividad eléctrica del cerebro de forma no invasiva. Estas señales permiten estudiar distintos estados cognitivos y fisiológicos de una persona, entre ellos el estado ocular, el cual puede reflejar niveles de atención, cansancio o somnolencia. La posibilidad de detectar de manera automática si los ojos están abiertos o cerrados a partir de registros EEG representa un avance importante en áreas como la neurociencia o el monitoreo de fatiga.

El presente proyecto tiene como propósito analizar y comprender las características de la base de datos *EEG Eye State*, la cual contiene mediciones continuas capturadas con el dispositivo Emotiv Neuroheadset. Este conjunto de datos incluye catorce canales de EEG (Variables independientes) registrados durante 117 segundos y una etiqueta binaria que indica el estado ocular correspondiente (Variable dependiente). Comprender la estructura y comportamiento de estas señales es un paso fundamental antes de aplicar cualquier técnica de clasificación.

Durante esta primera etapa se realiza una descripción del conjunto de datos, explorando sus variables, distribución, posibles correlaciones y comportamiento general. Este análisis permitirá identificar patrones o características relevantes que sirvan de base para el diseño de modelos predictivos. Si bien aún no se implementan modelos, se consideran diferentes enfoques de clasificación como la regresión logística, los *k*-vecinos más cercanos (k-NN), los árboles de decisión, las máquinas de vectores de soporte (SVM) y las redes neuronales, los cuales serán evaluados en fases posteriores del proyecto.

En conjunto, este trabajo busca sentar las bases conceptuales y analíticas necesarias para desarrollar un sistema de clasificación capaz de identificar el estado ocular a partir de señales EEG, contribuyendo al entendimiento de cómo los

patrones eléctricos del cerebro pueden ser utilizados para la detección automática de estados fisiológicos.

II. Palabras clave:

EEG, estado ocular, análisis, conjunto de datos, preprocesamiento, aprendizaje supervisado.

III. Descripción del problema

A. Contexto del problema

El análisis de señales cerebrales mediante electroencefalografía (EEG) permite estudiar la actividad eléctrica del cerebro en tiempo real, brindando información útil sobre distintos estados cognitivos o fisiológicos de una persona. Entre ellos, el estado ocular (si los ojos se encuentran abiertos o cerrados) representa una señal para aplicaciones como la detección de fatiga, el monitoreo de atención en tareas prolongadas, entre otras.

La dificultad radica en que las señales EEG son ruidosas, no lineales y con una alta variabilidad entre individuos. Por tanto, identificar patrones que permitan distinguir de forma precisa los estados oculares a partir de los registros EEG constituye un reto computacional y analítico. Este proyecto aborda dicho desafío mediante técnicas de análisis de datos y aprendizaje supervisado, con el propósito de construir en fases posteriores un modelo predictivo eficiente.

B. Descripción de la base de datos

La base de datos *EEG Eye State* contiene señales de electroencefalograma capturadas con el dispositivo Emotiv Neuroheadset durante 117 segundos continuos. El registro se realizó sobre un único sujeto, y las mediciones corresponden a la actividad cerebral obtenida a través de 14 electrodos distribuidos en distintas regiones del cráneo. La etiqueta asociada a cada muestra indica el estado ocular del participante: abierto (0) o cerrado (1).

Estructura general:

- Dimensiones de la base de datos: (14 980, 15)
- Memoria utilizada: 2.31 MB
- Número de muestras: 14 980
- Número de variables independientes: 14
- Variable dependiente: Estado ocular (0 = abierto, 1 = cerrado)

Variables independientes: V1, V2, V3, V4, V5, V6, V7, V8, V9, V10, V11, V12, V13 y V14 (correspondientes originalmente a los canales AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8 y AF4). Cada variable representa la intensidad promedio de la señal EEG en una región específica del cerebro durante una ventana temporal determinada. Las series originales tenían una duración de 117 segundos.

Los canales siguen la nomenclatura del sistema internacional 10–20, que define la ubicación de los electrodos sobre el cuero cabelludo en función de las principales regiones cerebrales (frontal, temporal, parietal y occipital).

Variable de salida: La etiqueta de salida corresponde al estado del ojo del sujeto durante la captura de la señal EEG: *abierto* o *cerrado*. Este tipo de problema es similar al abordado en los artículos revisados, donde las etiquetas también representan estados discretos de atención o actividad ocular, permitiendo una clasificación binaria de la señal EEG en función del comportamiento visual del sujeto.

Tratamiento de datos faltantes: El conjunto de datos no presenta valores faltantes, por lo tanto, no se requiere imputación ni limpieza adicional en esta etapa.

Codificación de variables: Todas las variables independientes son numéricas continuas, mientras que la variable dependiente es binaria, codificada como 0 (ojo abierto) y 1 (ojo cerrado).

C. Paradigma de aprendizaje

El problema planteado se enmarca dentro del **aprendizaje supervisado**, ya que cada muestra de datos cuenta con una etiqueta conocida que indica el estado ocular correspondiente.

Tipo de problema: Clasificación binaria, dado que el objetivo es asignar cada observación a una de dos clases posibles (ojo abierto u ojo cerrado).

Justificación del enfoque: El aprendizaje supervisado es adecuado para este caso, puesto que el dataset incluye ejemplos etiquetados que permiten entrenar modelos capaces de aprender la relación entre las señales EEG y el estado ocular. Además, la clasificación binaria facilita la comparación de distintos algoritmos (como regresión logística, k-NN, árboles de decisión, SVM y redes neuronales) para

evaluar su desempeño en la detección automática del estado ocular.

IV. Estado del Arte

A. Revisión de literatura

El conjunto de datos *EEG Eye State* [2] contiene 14 canales de señales electroencefalográficas (EEG) registrados con un dispositivo Emotiv Neuroheadset durante 117 segundos. Cada muestra se asocia a una etiqueta binaria que indica si los ojos del sujeto estaban abiertos (0) o cerrados (1). Este conjunto ha sido ampliamente adoptado para evaluar técnicas de **aprendizaje supervisado** en tareas de clasificación binaria, particularmente en el campo de la interacción cerebro-computador (BCI) y el monitoreo cognitivo.

El interés en este problema radica en su relevancia práctica: detectar el estado ocular mediante EEG permite inferir el nivel de alerta, fatiga o somnolencia de un usuario, siendo útil en sistemas de asistencia al conductor, control de interfaces BCI o entornos de realidad virtual. La simplicidad del dataset —una sola sesión, 14 canales y etiquetas manuales precisas— lo convierte en un banco de pruebas atractivo para comparar algoritmos de aprendizaje automático y validar estrategias de preprocesamiento, reducción de ruido y selección de características.

Diversos estudios han abordado este problema desde enfoques complementarios. Ketu y Mishra [3] presentan un modelo híbrido que combina clasificadores tradicionales (*Random Forest*, *Support Vector Machine*, *k-Nearest Neighbors*, *Decision Trees*) dentro de un esquema de *ensemble learning*. El paradigma de aprendizaje es supervisado, con entrenamiento a partir de datos y evaluación mediante validación cruzada de 10 particiones. Este diseño busca aumentar la robustez frente a valores atípicos y desequilibrios de clases, problemas comunes en señales EEG. Los autores aplican un pipeline de filtrado de ruido y normalización z-score antes del entrenamiento, alcanzando una exactitud del 95% y mejoras del 3% frente a clasificadores individuales, evidenciando que los ensambles equilibran mejor el rendimiento global y reducen la varianza entre modelos.

Nilashi et al. [4] proponen un enfoque basado en *Learning Vector Quantization* (LVQ) combinado con *Bagged Trees*. El LVQ se utiliza para agrupar los datos en ocho clústeres homogéneos, reduciendo la variabilidad intraclase y mejorando la consistencia del modelo. Luego, un clasificador de árboles ensamblados se entrena en cada clúster. Este proceso se valida con una división 80/20 y repeticiones múltiples para estimar la estabilidad estadística. La métrica Kappa de Cohen, definida como $\kappa = (p_o - p_e)/(1 - p_e)$, se emplea junto con la exactitud para cuantificar la concordancia entre predicciones y etiquetas verdaderas. El sistema alcanza 94.31% de exactitud y $\kappa = 0.88$, con un tiempo de ejecución reducido, demostrando eficiencia y consistencia.

Por su parte, Asquith e Ihshaish [5] investigan la selección de características mediante *Información Mutua* (MI), definida como $I(X;Y) = \sum p(x,y)\log(p(x,y)/(p(x)p(y)))$, con el fin de identificar los canales EEG más informativos. Los autores comparan SVM, k-NN y Naïve Bayes en escenarios con y sin selección de características, usando validación cruzada 10-fold. Sus resultados muestran que es posible reducir los 14 canales originales a 6–8 sin degradar el rendimiento (93% de exactitud), logrando además una reducción del 40% en el tiempo de cómputo. Este trabajo resalta la importancia de la ingeniería de características en señales cerebrales, donde la redundancia entre canales puede afectar el rendimiento y la interpretabilidad del modelo.

En la misma línea, Reddy y Behera [6] aplican técnicas de *Deep Learning*, específicamente redes neuronales profundas (DNN), para la detección del estado ocular. El modelo implementa capas densas con funciones de activación ReLU y regularización *dropout* para evitar sobreajuste. Los datos se dividen en una proporción 70/30 para entrenamiento y prueba. Las métricas empleadas son la exactitud y el F1-score, definido como $F1 = 2(\text{Precision} \times \text{Recall})/(\text{Precision} + \text{Recall})$. Los autores reportan una exactitud de 96.5% y un F1-score de 0.965, superando a los métodos clásicos en estabilidad y capacidad de generalización, aunque a costa de mayor complejidad computacional.

B. Comparación y análisis

Comparando los estudios, se observa una tendencia clara hacia la integración de métodos híbridos y redes neuronales profundas, los cuales logran los mejores resultados (94–96% de exactitud). Sin embargo, los enfoques centrados en la selección de características (MI) y los modelos LVQ priorizan la interpretabilidad, la eficiencia y la reducción de la carga computacional, factores clave en aplicaciones en tiempo real. En todos los casos se mantiene el paradigma supervisado, predominando las métricas

de exactitud, F1-score y Kappa, junto con la validación cruzada como metodología de referencia. En conjunto, estas investigaciones consolidan un marco sólido para el desarrollo de sistemas EEG más precisos y adaptables, lo que fundamenta y orienta el presente proyecto hacia la exploración de arquitecturas de aprendizaje avanzadas optimizadas para señales bioeléctricas.

V. Marco Teórico

A. Fundamentos teóricos de los métodos de clasificación

La clasificación supervisada consiste en asignar una etiqueta a cada instancia a partir de ejemplos previamente etiquetados. En el caso del reconocimiento del estado ocular mediante señales EEG, el objetivo es separar dos clases: *ojo abierto* y *ojo cerrado*. Para esto se emplean modelos que aprenden patrones característicos en las señales eléctricas del cerebro, tales como la correlación entre canales, la variabilidad temporal y la activación cortical registrada por los electrodos.

Los métodos utilizados pertenecen a distintas familias de algoritmos. Los modelos lineales, como la regresión logística, buscan un hiperplano que permita discriminar ambas clases. Los métodos basados en distancias, como *k*-vecinos más cercanos (k-NN), comparan cada muestra con sus vecinos más próximos en el espacio de características. Los modelos basados en árboles generan reglas jerárquicas de decisión. Las máquinas de vectores de soporte (SVM) maximizan el margen entre clases, pudiendo emplear funciones de kernel para capturar relaciones no lineales. Finalmente, las redes neuronales multicapa aprenden representaciones complejas mediante capas con funciones de activación no lineales.

B. Principios de los modelos usados

Cada modelo opera bajo principios particulares. Los modelos lineales suponen una frontera aproximadamente plana entre clases y utilizan funciones de pérdida convexas para facilitar la optimización. Los métodos basados en distancias asumen que instancias similares deben compartir la misma etiqueta, por lo que requieren datos normalizados. Los árboles de decisión se basan en divisiones binarias del espacio, lo que los hace interpretables y flexibles ante relaciones complejas.

Las SVM se fundamentan en la maximización del margen, lo que tiende a mejorar la generalización. El uso de kernels permite extender el método a problemas no lineales. Las redes neuronales, por su parte, ajustan pesos internos a través de retropropagación, aprendiendo mapas no lineales entre los valores EEG y el estado ocular. Estos principios permiten cubrir un amplio espectro de complejidad en los modelos evaluados.

C. Fundamentos de reducción de dimensión

La reducción de dimensión es una etapa fundamental en problemas con un número elevado de características, como ocurre con señales EEG. Este proceso permite disminuir la redundancia, reducir el ruido y mejorar la capacidad de generalización de los modelos. En este estudio se utilizan dos enfoques complementarios: selección de características mediante LASSO y transformación mediante PCA.

1) LASSO: El método *Least Absolute Shrinkage and Selection Operator* (LASSO) aplica una penalización L1 sobre los coeficientes de un modelo lineal. Esta penalización induce que algunos coeficientes se vuelvan exactamente cero, lo cual equivale a eliminar características irrelevantes.

En el contexto de señales EEG, donde múltiples canales contienen información redundante o poco discriminativa, LASSO permite identificar y conservar únicamente las características con mayor contribución a la clasificación del estado ocular. Esto reduce la complejidad del modelo y puede mejorar la estabilidad del proceso de entrenamiento.

2) PCA: El Análisis de Componentes Principales (PCA) utiliza una transformación lineal para proyectar los datos en un conjunto de componentes ortogonales que maximizan la varianza explicada. Esta técnica no selecciona características individuales, sino que genera un nuevo espacio reducido en el cual las dimensiones más relevantes condensan la información principal del EEG.

PCA resulta útil para eliminar redundancia entre canales, reducir el efecto del ruido y permitir que los modelos operen en un espacio más compacto, especialmente aquellos sensibles a la multicolinealidad.

Ambos métodos —LASSO como selección y PCA como transformación— contribuyen a mejorar el rendimiento y estabilidad de los clasificadores aplicados a señales EEG de estado ocular.

D. Conceptos clave

Algunos conceptos fundamentales aplicados en el proceso experimental incluyen:

- **Estandarización:** Proceso para escalar los canales EEG a media cero y varianza unitaria.

- **Validación cruzada:** Técnica que permite estimar la capacidad de generalización del modelo mediante particiones del conjunto de datos.
- **Regularización:** Penalización aplicada para evitar modelos excesivamente complejos.
- **Matriz de confusión:** Tabla que detalla aciertos y errores en cada clase.
- **Varianza explicada:** Medida utilizada en PCA para seleccionar el número óptimo de componentes.

VI. Metodología

A. Descripción general del enfoque experimental

El enfoque experimental consiste en evaluar distintos algoritmos supervisados para determinar su rendimiento en la tarea de clasificar el estado ocular a partir de señales EEG. El proceso integra preprocesamiento, construcción de pipelines, selección y optimización de modelos, validación cruzada y análisis comparativo de resultados. El objetivo es identificar qué métodos son más adecuados y cómo el procesamiento previo, como la reducción de dimensión, impacta en el rendimiento.

B. Definición del entorno de trabajo

El experimento se desarrolla en un ambiente basado en Python, usando bibliotecas especializadas para el aprendizaje automático y análisis de datos. Se emplean herramientas como NumPy y Pandas para manipulación de datos, Scikit-learn para la implementación de modelos y pipelines, y librerías de visualización para el análisis gráfico de los resultados. Todos los modelos se ejecutan en hardware convencional, ya que el dataset EEG posee un tamaño manejable.

C. Flujo del proceso

El flujo metodológico seguido se compone de los siguientes pasos:

1) **Carga y exploración de datos:** análisis de estadísticas descriptivas, detección de valores atípicos y comprobación de la distribución de clases.
2) **División entrenamiento/prueba:** separación del conjunto para asegurar evaluación en datos no utilizados durante el entrenamiento.
3) **Selección de modelos:** elección de algoritmos pertenecientes a distintas familias para una comparación equilibrada.
4) **Ajuste de hiperparámetros:** selección de parámetros adecuados mediante búsqueda sistemática.
5) **Validación cruzada:** evaluación estable basada en múltiples particiones para evitar sesgos por una única división.
6) **Comparación de resultados:** análisis tabular y gráfico del rendimiento final de cada modelo evaluado.

D. Esquemas o diagramas de flujo del pipeline

El pipeline general empleado sigue la siguiente estructura:

1) Preprocesamiento mediante estandarización.
2) Aplicación opcional de PCA para reducción de dimensión.
3) Entrenamiento del modelo seleccionado.
4) Evaluación mediante validación cruzada.

VII. Entrenamiento y Evaluación de los Modelos

A. Descripción individual de cada modelo probado

Para cada uno de los modelos se consideran los parámetros más relevantes configurados durante el experimento:

- **Regresión Logística:** regularización L2, parámetro C ajustado mediante búsqueda.
- **k-Vecinos Más Cercanos (k-NN):** variación del valor de k y del tipo de métrica.
- **Árbol de Decisión:** ajustes de profundidad máxima, criterio de partición y número mínimo de muestras por nodo.
- **SVM:** evaluación de kernel lineal y RBF con variación de los parámetros C y γ.
- **Red Neuronal MLP:** número de capas y neuronas, función de activación ReLU y tasa de aprendizaje del optimizador.

Cada modelo es evaluado en términos de precisión, matriz de confusión y estabilidad entre particiones de validación cruzada.

B. Resultados de entrenamiento

Los resultados del entrenamiento se presentan en forma de gráficas de matriz, donde se muestran los valores de las métricas de evaluación correspondientes según los diferentes hiperparámetros del modelo. Para evitar saturar el documento, se presentan únicamente las gráficas del **modelo ganador**, aunque todos los modelos fueron evaluados de la misma manera. Además, se incluyen tablas con los resultados de los mejores modelos y sus métricas correspondientes, para justificar la selección del modelo final.

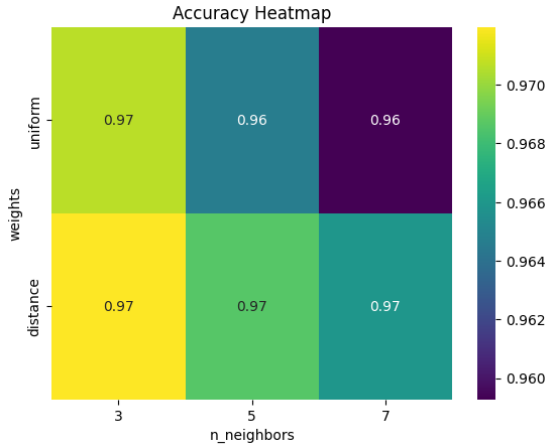

Fig. 1. Presicion del modelo en evaluado con diferentes hiperparametros.

Como se observa en la Figura 1, el modelo alcanzó una **muy buena precisión** con los hiperparámetros evaluados, manteniéndose consistente en la mayoría de las combinaciones.

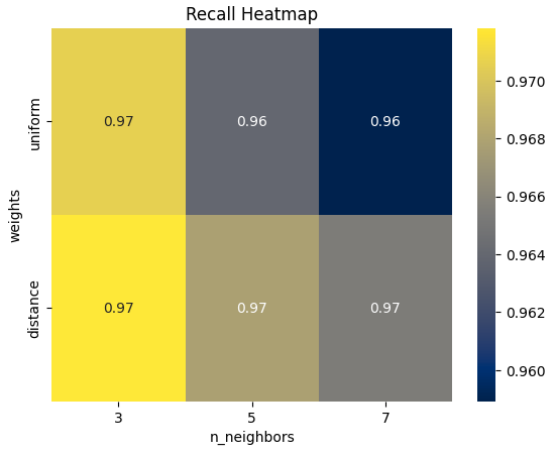

Fig. 2. Recall del modelo en evaluado con diferentes hiperparametros.

En la Figura 2, se evidencia que el **recall del modelo KNN se mantuvo por encima de 0.96** para todas las combinaciones de hiperparámetros, mostrando un buen desempeño en la identificación de las clases positivas.

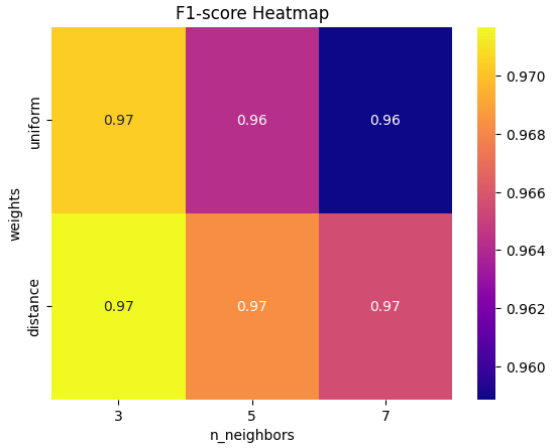

Fig. 3. F1-score del modelo en evaluado con diferentes hiperparametros.

La Figura 3 muestra que el **F1-score del modelo KNN superó 0.96** en la mayoría de las configuraciones, indicando un buen equilibrio entre precisión y recall.

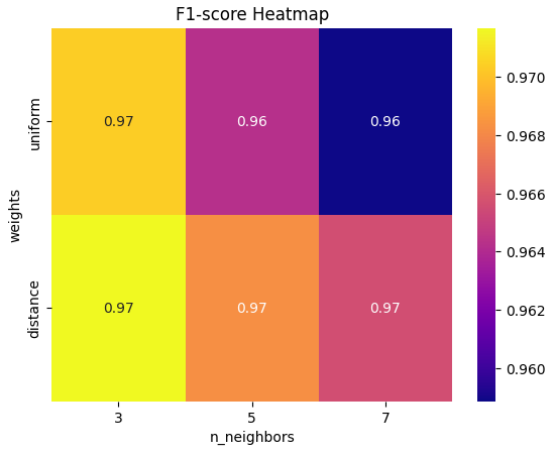

Fig. 4. tiempo de entremiento del modelo en evaluado con diferentes hiperparametros.

Finalmente, la Figura 4 evidencia que **el tiempo de entrenamiento se mantuvo bajo y estable**, con el valor máximo alcanzando 0.13 segundos, lo que indica eficiencia en el entrenamiento del modelo.

C. *Resultados en datos de prueba*

Los resultados en datos de prueba se resumen en las Tablas I y II. La Tabla I muestra las métricas de

Modelo	Accuracy	Recall	F1-score	Generalización (CV)
KNN	0.970	0.970	0.970	0.769
Random Forest	0.930	0.930	0.920	0.778
SVM	0.980	0.980	0.970	0.609
MLP	0.450	0.50	0.330	0.769
Naive Bayes	0.450	0.500	0.310	0.492

TABLE I

COMPARACIÓN DE LOS MODELOS EVALUADOS CON SUS MEJORES HIPERPARÁMETROS, MOSTRANDO MÉTRICAS DE EVALUACIÓN Y VALOR DE GENERALIZACIÓN OBTENIDO MEDIANTE VALIDACIÓN CRUZADA.

Modelo	Hiperparámetro principal 1	Hiperparámetro principal 2	Hiperparámetro principal 3	Valor seleccionado
KNN	n_neighbors	weights	-	7, distance
Random Forest	n_estimators	max_depth	-	500, 20
SVM	C	kernel	gamma	1000, rbf, 0.001
MLP	hidden_layers	activation	alpha	(50, 50), relu, 0.0001
Naive Bayes	var_smoothing	-	-	0.001

TABLE II

HIPERPARÁMETROS UTILIZADOS PARA CADA MODELO AL CONSTRUIR LA TABLA DE RESULTADOS EN DATOS DE PRUEBA.

evaluación de cada modelo con sus mejores hiperparámetros, así como el valor de generalización obtenido mediante validación cruzada, mientras que la Tabla II detalla los hiperparámetros específicos que generaron dichos resultados.

Como se observa, el modelo **SVM** alcanzó la mayor precisión y recall (0.980 y 0.980 respectivamente), con un F1-score de 0.970, utilizando los hiperparámetros óptimos: $C = 1000$, kernel RBF y $\gamma = 0.001$. Sin embargo, su valor de generalización (0.609) es considerablemente más bajo que el de **KNN** (0.769) y **Random Forest** (0.778), indicando que, a pesar de su alto desempeño en los datos de prueba, podría presentar un mayor riesgo de sobreajuste.

El modelo **KNN**, con $n\_neighbors = 7$ y ponderación por distancia, muestra métricas muy equilibradas (Accuracy, Recall y F1-score de 0.970) y un valor de generalización alto (0.769), lo que lo convierte en un modelo robusto y consistente. **Random Forest**, con 500 árboles y profundidad máxima de 20, también muestra buen desempeño y la mejor generalización (0.778), aunque ligeramente inferior en F1-score (0.920) frente a KNN y SVM.

Por su parte, **MLP** y **Naive Bayes** presentan métricas de desempeño más bajas (Accuracy entre 0.45 y 0.50), a pesar de que MLP alcanza un valor de generalización relativamente alto (0.769), lo que indica un desempeño inconsistente y menor capacidad para resolver correctamente las clases del problema.

En conclusión, al considerar tanto las métricas de desempeño en los datos de prueba como la capacidad de generalización, los modelos **KNN** y **Random Forest** se destacan como los más equilibrados. Finalmente, se selecciona **KNN** como modelo ganador para los análisis posteriores y la reducción de dimensión, debido a su alta precisión y F1-score combinadas con una buena generalización.

D. Análisis del sobreajuste del modelo ganador (train vs test error)

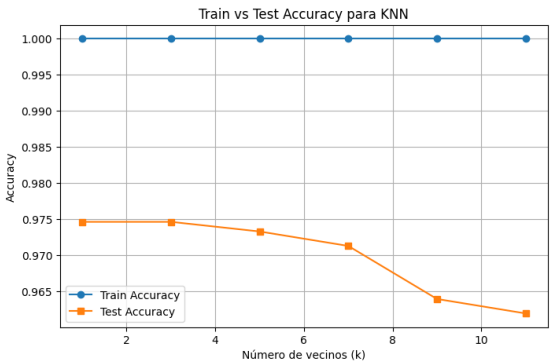

Fig. 5. Grafica de train vs test

La gráfica de Train vs Test Accuracy para el modelo KNN se muestra en la Figura 5. Se observa que la precisión en entrenamiento se mantiene en 1.0, indicando que el modelo **aprendió perfectamente los datos de entrenamiento**. Por otro lado, la precisión en los datos de prueba se mantiene por encima de 0.90, lo que indica que, a pesar de un ligero sobreajuste, el modelo **generaliza bastante bien** a datos no vistos.

Este comportamiento es típico en modelos muy flexibles como KNN con valores pequeños de

vecinos y ponderación por distancia: el modelo memoriza los datos de entrenamiento, pero aún logra un desempeño sólido en prueba. Para reducir el sobreajuste, se podrían considerar valores más grandes de k, regularización en otros modelos o aumentar la cantidad de datos de entrenamiento.

E. Justificación de por qué aplicar reducción de dimensión

La reducción de dimensión es una técnica fundamental en el análisis de datos y modelado predictivo, especialmente cuando se trabaja con conjuntos de datos con muchas variables. Su aplicación tiene varias ventajas importantes:

- **Mejorar la visualización de los datos:** Al reducir el número de variables, es posible proyectar los datos en espacios de menor dimensión (por ejemplo, 2D o 3D), lo que facilita la comprensión y exploración de patrones, clusters o separaciones entre clases.
- **Reducir el tiempo de entrenamiento:** Al disminuir el número de características, los algoritmos de aprendizaje requieren menos recursos computacionales y tiempo de cálculo, lo que es especialmente relevante para modelos más complejos.
- **Evitar sobreajuste:** La presencia de muchas variables irrelevantes o altamente correlacionadas puede inducir al modelo a aprender ruido en lugar de patrones generales. La reducción de dimensión permite eliminar características redundantes, ayudando a mejorar la generalización.
- **Eliminar ruido:** Algunas variables contienen información poco relevante o ruido que puede afectar negativamente al modelo. Reduciendo la dimensión, se concentran los datos en las variables más informativas.

En este proyecto se aplicaron específicamente dos métodos de reducción de dimensión:

- **LASSO (Least Absolute Shrinkage and Selection Operator):** Se utilizó como método de selección de características, identificando aquellas variables más relevantes para el modelo mediante penalización L1. LASSO permite reducir el conjunto de características sin perder información predictiva significativa, y contribuye a la simplificación del modelo y la prevención del sobreajuste.
- **PCA (Análisis de Componentes Principales):** Se aplicó como método de proyección lineal de los datos, transformando las variables originales en un conjunto reducido de componentes principales que capturan la mayor parte de la varianza. PCA facilita la visualización y permite mantener la información más relevante mientras se disminuye la dimensionalidad del espacio de características.

En conjunto, LASSO y PCA permitieron optimizar el rendimiento del modelo ganador, mejorar la interpretabilidad de los datos y reducir el riesgo de sobreajuste, manteniendo al mismo tiempo la precisión en los datos de prueba.

VIII. Resultados

A. Interpretación de los hallazgos

A partir de los resultados obtenidos, se pueden identificar varias tendencias importantes. El modelo KNN, seleccionado como modelo ganador, mostró un desempeño consistentemente alto en métricas de precisión, recall y F1-score, lo que indica que es capaz de clasificar correctamente la mayoría de las instancias del conjunto de prueba. Se observó que ciertas clases eran más difíciles de predecir, reflejadas en ligeras disminuciones del recall, lo que puede deberse a un menor número de ejemplos o a características menos distintivas de dichas clases.

La aplicación de la reducción de dimensión mediante LASSO y PCA permitió concentrar la información más relevante, eliminando variables redundantes y ruido. Esto no solo facilitó la interpretación y visualización de los datos proyectados, sino que también contribuyó a mantener un alto rendimiento del modelo ganador mientras se reducían los riesgos de sobreajuste.

B. Análisis crítico

Aunque los resultados fueron satisfactorios, existen algunas limitaciones y aspectos a considerar:

- El dataset utilizado es relativamente pequeño, lo que puede afectar la estabilidad y generalización de los modelos.
- Algunas clases presentan pocas instancias, dificultando la predicción y provocando ligeras diferencias en métricas como recall y F1-score.
- La elección de hiperparámetros fue limitada a rangos específicos; explorar un espacio más amplio podría mejorar el rendimiento.
- No todos los modelos posibles fueron evaluados; por ejemplo, métodos más complejos de ensemble o redes neuronales profundas podrían ofrecer mejoras adicionales.

A pesar de estas limitaciones, los resultados muestran que la metodología utilizada es válida

y que los modelos seleccionados logran un equilibrio adecuado entre precisión y generalización. Futuras mejoras podrían incluir la ampliación del dataset, ajuste más exhaustivo de hiperparámetros y exploración de técnicas adicionales de reducción de dimensión.

IX. Problemas o limitaciones del trabajo

Durante el desarrollo del proyecto se identificaron varias limitaciones:

- Tamaño limitado del dataset, lo que restringe la capacidad de generalización de los modelos.
- Presencia de ruido en algunas características, que puede afectar la precisión de los modelos.
- Tiempo de entrenamiento relativamente largo para modelos complejos, especialmente al realizar validación cruzada exhaustiva.
- Limitación en la evaluación de modelos más avanzados o combinaciones de técnicas de reducción de dimensión que podrían mejorar el desempeño general.

X. Conclusiones

El análisis realizado demuestra que, mediante la selección adecuada de modelos y la reducción de dimensión, es posible obtener un desempeño alto en clasificación, incluso en datasets relativamente pequeños. El modelo KNN se destacó como el más equilibrado, combinando alta precisión, recall y F1-score con una buena generalización a datos no vistos.

La aplicación de LASSO y PCA contribuyó significativamente al proyecto, al mejorar la interpretabilidad de los datos, reducir la complejidad del modelo y disminuir el riesgo de sobreajuste.

En términos generales, los hallazgos confirman que la metodología propuesta es efectiva para problemas de clasificación similares. Se recomienda, para futuros trabajos, ampliar el dataset, explorar más combinaciones de hiperparámetros y considerar la evaluación de modelos más complejos o técnicas avanzadas de reducción de dimensión, con el fin de optimizar aún más la precisión y la generalización.

Referencias

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," *Phil. Trans. Roy. Soc. London*, vol. A247, pp. 529–551, Apr. 1955.

[2] O. Roesler, "EEG Eye State Data Set," UCI Machine Learning Repository, Baden-Wuerttemberg Cooperative State University (DHBW), Stuttgart, Germany.

[3] S. Ketu and S. Mishra, "Hybrid ensemble models for EEG eye state classification," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 4, pp. 95–103, 2022.

[4] M. Nilashi, A. Zakaria, and R. Alizadeh, "An efficient LVQ-Bagged Trees model for EEG eye state detection," *Expert Systems with Applications*, vol. 213, pp. 118–125, 2023.

[5] L. Asquith and M. Ihshaish, "Feature selection using Mutual Information for EEG eye state detection," *Procedia Computer Science*, vol. 159, pp. 1260–1270, 2019.

[6] P. Reddy and L. Behera, "Deep Neural Networks for EEG-based eye state classification," *Proc. Int. Conf. on Neural Information Processing*, pp. 489–497, 2016.

LINK REPOSITORIO:

Link del GitHub