ENTREGA FINAL PROYECTO

Tapias, John y Cataño, Esteban

{absurdoeco, esteban.catanoe}@gmail.com

***Resumen*— En el presente trabajo se utilizan diferentes métodos de aprendizaje de máquinas para abordar un problema de clasificación del estado de los ojos a partir de la toma de un electroencefalograma, con el objetivo de determinar cuál es el mejor modelo que puede ser utilizado en aplicaciones de aprendizaje de máquinas para el problema tratado.**

***Palabras Claves*— Aprendizaje de Máquinas, Electroencefalograma, Estado de los Ojos.**

I. DESCRIPCIÓN DEL PROBLEMA

El problema que se aborda en este trabajo consiste en encontrar la relación entre los resultados de un electroencefalograma (EEG) y el estado en que se encuentra los ojos, ya sea abiertos o cerrados.

A cada persona cuyos resultados fueron incluidos en la base de datos se le realizó la toma del EEG y con una cámara se capturó el estado de los ojos, posteriormente se le agregó manualmente el estado tomado por la cámara a los resultados del EEG.

Un EEG consiste en medir la actividad eléctrica del cerebro, colocando pequeños electrodos (Discos metálicos con cables delgados) en el cuero cabelludo, estos envían las señales a una computadora que registra los datos.

El campo de aplicación de este problema puede estar centrado en ayudar a personas con discapacidad física y/o problemas de habla en determinar que decisiones desea tomar a partir de la actividad eléctrica cerebral.

El problema tratado consiste en un problema de clasificación en el que a partir de los resultados del EEG se quiere clasificar si los ojos se encuentran abiertos o cerrados.

Los datos de entrada son 14 variables numéricas en la que cada una contiene el resultado tomado por un electrodo específico y la variable de salida corresponde a cero si el ojo se encuentra abierto y uno si está cerrado.

II. REVISIÓN DE ARTÍCULOS

La base de datos utilizada para este problema no provee artículos que hayan utilizado estos mismos datos, por lo tanto se hace una búsqueda lo más completa posible en bases de datos como sciencedirect, ieexplore y google académico y solo se logra encontrar un artículo que utilice la misma base de datos, teniendo como inconveniente que es un artículo de conferencia, pero debido a la poca documentación relacionada al problema se decide utilizar este artículo.

En [1] trabajan con la misma base de datos utilizada en este proyecto. El estado del ojo es clasificado como una serie de tiempo de datos continuos, utilizando el enfoque Incremental Attribute Learning (IAL). Los resultados obtenidos son:

TABLE II. EEG EYE STATE TIME-SERIES CLASSIFICATION RESULTS

	Approaches	Classification ErrorRate (%)
1	AD with Feature Extraction where y'(t)=y(t+1)-y(t)	27.4573
2	Feature Extraction without Time Series, Original Ordering	27.4793
3	Batch-Training Method where y'(t)=y(t+1)-y(t)	29.5046
4	Conventional Batch-Training Method without Time Series	30.6328

En [2] utilizan 11 personas saludables para la toma de las muestras, con edad media en 21.6 y un rango de edades entre 20 y 23 años. A cada persona se le realizaron 3 sesiones de experimentos, la primera por 3 minutos con los ojos cerrados, luego otros 3 minutos con los ojos abiertos y una última toma por 10 minutos asignándole una tarea de atención visual. Utilizan la técnica mapas estadísticos no paramétricos para medir el poder de las bandas theta, alpha y beta (Bandas conectadas en el cuero cabelludo para la toma del electroencefalograma). El contexto de aplicación de este artículo es estudiar los cambios en los ritmos alpha cuando el ojo se encuentra cerrado o abierto, con el objetivo de encontrar cambios en el estado de atención visual.

En [3] realizan la toma de datos con 10 participantes con edad media de 24.4 años y desviación estándar 1.5. A cada persona le fueron realizadas 7 toma del electroencefalograma. Es utilizado el algoritmo de clasificación online, en el cual la secuencia de funciones Haar wavelet se utiliza para analizar la señal producida por el movimiento del ojo, utilizando la

primera sesión para calibración y las seis restantes para pruebas. Obtienen un porcentaje promedio de eficiencia del 85.21%. Este trabajo puede ser utilizado para construir software y hardware que pueda soportar la alta velocidad en el análisis de la señales de los EEG.

III. EXPERIMENTOS

La base de datos usada es un conjunto de datos compuesto por 14980 muestras donada el 10 de Junio de 2013, que se encuentra disponible en el UCI Machine Learning Repository. Puede ser consultado en la dirección web: https://archive.ics.uci.edu/ml/datasets/EEG +Eye+State.

Para el presente trabajo se utiliza un total de 13480 para entrenar y validar los modelos que se van a utilizar, determinando el mejor modelo para el problema trabajado. Luego se clasifican las 1500 muestras restantes con este modelo.

La metodología de validación usada es boostrapping en la cual se define un porcentaje del 90% para entrenar el modelo y el 10% restante para validarlo, repitiendo el procedimiento 10 veces. Los conjuntos de muestras de entrenamiento y validación son tomados aleatoriamente en cada repetición.

IV. EVALUACIÓN DE MODELOS

Se evalúan 8 modelos, para determinar cuál proporciona mejor resultado para el problema tratado.

A. Regresión Logística

Tasa de Aprendizaje	Grado del Polinomio	Eficiencia	IC	Sensibilidad	Especificidad
0.1	1	0.61869	+- 0.014688	0.62009	0.61637
	2	0.63501	+- 0.005245	0.64905	0.6074
	3	0.6224	+- 0.005245	0.62176	0.62384
	4	0.6224	+- 0.003147	0.6237	0.62
	5	0.61573	+- 0.015737	0.6252	0.59864

B. Máquinas de Soporte Vectorial

1) Kernel Lineal

Box Constraint	Eficiencia	IC	Sensibilidad	Especificidad
0.01	0.64169	+- 0.007343	0.64862	0.62687
0.1	0.63798	+- 0.001049	0.65557	0.60709
1	0.65467	+- 0.012065	0.65608	0.65164

2) Kernel Gaussiano

Box Constraint	Gamma	Eficiencia	IC	Sensibilidad	Especificidad
0,01	0.01	0.55415	+- 0010491	0.55415	NaN
	0.1	0.72181	+- 0.007343	0.67607	0.8574
	1	0.68954	+- 0.004721	0.68836	0.69166
0.1	0.01	0.55564	+- 0.017835	0.55352	NaN
	0.1	0.84792	+- 0.006294	0.84675	0.84967
	1	0.80675	+- 0.050882	0.81075	0.80114
1	0.01	0.611758	+- 0.10858	0.59607	0.9554
	0.1	0.89763	+- 0.006294	0.89906	0.89571
	1	0.82752	+- 0.016261	0.8296	0.82449

C. Modelo de Mezclas Gaussianas

Matriz de Covarianza	Mezclas	Eficiencia	IC	Sensibilidad	Especificidad
Completa	1	0.61409	+- 0.041804	0.90257	0.54271
	2	0.7862	+- 0.009237	0.83872	0.73453
	3	0.85178	+- 0.012093	0.86788	0.83441
Diagonal	1	0.47856	+- 0.043599	0.57563	NaN
	2	0.47507	+- 0.008624	0.70093	0.46167
	3	0.55497	+- 0.027421	0.56136	NaN
Esférica	1	0.4747	+- 0.041995	0.58615	0.45951
	2	0.52878	+- 0.043093	0.56984	0.49413
	3	0.5526	+- 0.016508	0.55793	NaN

D. K Vecinos más cercanos

Vecinos	Eficiencia	IC	Sensibilidad	Especificidad
1	0.91061	+- 0.062423	0.91604	0.056595
2	0.87648	+- 0.055079	0.84239	0.93538
3	0.92211	+- 0.050358	0.92546	0.9177
4	0.91024	+- 0.052456	0.88873	0.9409
5	0.86424	+- 0.007343	0.8746	0.85162
6	0.8572	+- 0.000524	0.83158	0.89516
7	0.91511	+- 0.050358	0.91024	0.90415
8	0.86387	+- 0.001573	0.91511	0.90415
9	0.86573	+- 0.015737	0.87362	0.85577
10	0.86499	+- 0.005245	0.86359	0.86701

E. Redes Neuronales Artificiales

Capas Ocultas	Neuronas	Eficiencia	IC	Sensibilidad	Especificidad
1	23	0.56157	+- 0.016366	0.56984	0.55723
	28	0.56528	+- 0.012489	0.57232	0.55965
	33	0.53932	+- 0.01605	0.53879	0.52989
2	23	0.55786	+- 0.012018	0.56983	0.54984
	28	0.56306	+- 0.013362	0.57784	0.56983
	33	0.5727	+-0.008372	0.58344	0.57563

F. Random Forest

Árboles	Eficiencia	IC	Sensibilidad	Especificidad
10	0.90282	+- 0.001049	0.88756	0.92538
15	0.91766	+- 0.013639	0.91221	0.92502
20	0.91728	+- 0.027148	0.90596	0.93305
25	0.9158	+- 0.012065	0.90764	0.92542
30	0.92841	+- 0.012065	0.92639	0.93078

G. Árboles de Decisión

Nivel Poda	Eficiencia	IC	Sensibilidad	Especificidad
Sin Poda	0.83605	+- 0.022032	0.85878	0.80764
1	0.8405	+- 0.005245	0.85763	0.82122
2	0.8342	+- 0.009966	0.84271	0.82276
3	0.82159	+- 0.008917	0.83233	0.80863
4	0.84199	+- 0.002098	0.85644	0.8238
5	0.84347	+- 0.008393	0.85993	0.82259

H. Ventana de Parzen

Ventana de Suavizado	Eficiencia	IC	Sensibilidad	Especificidad
0.05	0.91543	+- 0.054554	0.91615	0.91564
0.1	0.92953	+-0.051407	0.93766	0.922026
1	0.63687	+- 0.021507	0.6435	0.62935
10	0.56602	+- 0.028326	0.62978	0.53006

V. ANÁLISIS INDIVIDUAL

A. Coeficiente de Correlación de Pearson

Se utiliza el coeficiente de correlación de Pearson para analizar como está correlaccionadas las características y poder analizar cuales características pueden ser eliminadas.

```
***Coeficiente de correlación de Pearson***
La Característica #: 1 explica la característica #: 9 un 0.99995%
La Característica #: 1 explica la característica #: 13 un 0.99793%
La Característica #: 5 explica la característica #: 12 un 0.99965%
La Característica #: 5 explica la característica #: 14 un 0.99897%
La Característica #: 9 explica la característica #: 13 un 0.99811%
La Característica #: 12 explica la característica #: 14 un 0.99809%
```

De acuerdo a los resultados obtenidos se puede destacar que las características 9, 12, 13 y 14 son explicadas por otras características.

```
***Coeficiente de correlación de Pearson***
***Características candidatas a ser eliminadas***
Característica #: 9
Característica #: 12
Característica #: 13
Característica #: 14
```

B. Índice de Fisher

Se utiliza el índice Fisher para hacer un análisis de las características sin normalizar y posteriormente normalizadas.

```
Command Window
***Cociente discriminante de Fisher***
***Sin normalizar***

F =

    0.0418
  120.8256
  309.9804
  143.2596
    0.0569
  319.7656
  117.1195
  152.5317
    0.0243
  100.1585
  395.3794
    0.1033
    0.3469
    0.6901

***Cociente discriminante de Fisher***
***Normalizado***

F =

    0.0001
    0.3056
    0.7840
    0.3623
    0.0001
    0.8088
    0.2962
    0.3858
    0.0001
    0.2533
    1.0000
    0.0003
    0.0009
    0.0017
```

A partir de los resultados obtenidos, se resalta que las características 3, 6 y 11 son las que más aportan al modelo.

VI. SELECCIÓN DE CARACTERÍSTICAS

A. Método de Búsqueda Secuencial

Se decide utilizar una función tipo wrappers como función objetivo debido a que estas combinan la búsqueda en el espacio de atributos con el algoritmo de aprendizaje, evaluando los conjuntos de atributos y escogiendo el más adecuado. El inconveniente que presentan es que son más costosos que las funciones tipo filtro aunque suelen obtener mejores resultados.

Se utiliza como función objetivo el modelo de discriminantes gaussianos de tipo lineal, realizando 10 repeticiones al método de búsqueda secuencial, obteniendo los siguientes porcentajes de veces que aparece la característica en las arrojadas por el método.

```
Command Window
***Selección de características***
***Wrapper: % de importancia de cada característica***

fsAcumulado =

   20.0000
   25.0000
         0
         0
         0
   55.0000
   55.0000
   20.0000
   35.0000
   10.0000
   30.0000
   10.0000
  100.0000
  100.0000
```

A partir de los resultados obtenidos, se puede observar que las características 3,4 y 5 no fueron retornadas ninguna vez, por lo tanto se deciden eliminar.

VII. EXTRACCIÓN DE CARACTERÍSTICAS

Se utiliza el método PCA para realizar extracción de características, este método retorna los valores de los componentes principales, para los cuales se utilizan los componentes de mayor a menor hasta que acumulen el 90 % de la varianza. Se realizan 10 repeticiones con conjuntos de entrenamiento aleatorios.

Se vuelve a evaluar los 3 mejores modelos encontrados, obteniendo los siguientes resultados.

Método	Eficiencia	IC	Sensibilidad	Especificidad
1	0.68694	+- 0.013639	0.68512	0.6898
2	0.70364	+- 0.010519	0.72333	0.68213
3	0.67878	+- 0.019653	0.65574	0.72646

1: Random Forest con 30 árboles
2: Ventana de Parzen con h = 0.05
3: K vecinos con 6 vecinos.

VIII. DISCUSIÓN

Luego de aplicar técnicas de selección y extracción de características, se obtiene resultados muy bajos comparados con los de la primera simulación, por lo cual se decide trabajar con los datos originales, sin haberle aplicado ninguna técnica de selección ni extracción.

Con los resultados obtenidos en la primera simulación se observa claramente que los mejores modelos para el problema que se está trabajando son k vecinos, random forest, árboles de decisión y ventana de parzen.

Comparando los cuatro modelos que mejor resultados arrojó con los del artículo que trabajó con la misma base de datos, se puede destacar que estos fueron mucho mejor, obteniendo una eficacia por encima del 85%, en cambio en el artículo la eficiencia máxima la encontraron en 73%.

Se construyen gráficos para mirar el comportamiento de los cuatro modelos seleccionados al variar el valor de los parámetros.

IX. PREDICCIONES

A partir de los resultados obtenidos en la sección de discusión se toma la decisión de realizar la predicción utilizando dos modelos:

1. Random Forest con 30 árboles.
2. Ventana de Parzen con h = 0.05.

X. REFERENCIAS

- [1] Wang, T., Guan, S.-U., Man, K. L., & Ting, T. O. (n.d.). Time Series Classification for EEG Eye State Identification based on Incremental Attribute Learning. http://doi.org/10.1109/IS3C.2014.52
- [2] Li, L. (n.d.). The Differences among Eyes-closed, Eyes-open and Attention States: An EEG Study.
- [3] Belkacem, A. N., Shin, D., Kambara, H., Yoshimura, N., & Koike, Y. (2015). Online classification algorithm for eye-movement-based communication systems using two temporal EEG sensors. Biomedical Signal Processing and Control, 16, 40–47. http://doi.org/10.1016/j.bspc.2014.10.005