Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus, Querétaro

Inteligencia Artificial Avanzada para la Ciencia de Datos - (Gpo 103)

Módulo 2

Análisis y Reporte sobre el desempeño del modelo

Estudiante:

Samantha Guanipa Ugas A01703936

Profesor:
Adrián Rodríguez Rocha

Fecha de entrega:
10 de septiembre de 2023

I. Introducción

Según el Centro de Diagnóstico e Intervención Neurocognitiva, el Alzheimer es una enfermedad neurodegenerativa crónica, que afecta la memoria y el comportamiento de una persona, usualmente de edad avanzada. Suele ser una causa de demencia para los adultos mayores. Entre sus síntomas se encuentran: problemas de memoria, desafíos en la resolución de problemas, confusión con tiempo y lugar, y la pérdida de habilidades cognitivas. “El análisis de la escritura puede ser un buen indicativo indirecto para ver cómo va avanzando la enfermedad, a pesar que externamente no se vean cambios evidentes.”(Juan Moisés de la Serna, 2021). De acuerdo con la idea de detectar el Alzheimer en una etapa temprana con base en la escritura de los posibles pacientes, se decidió utilizar el dataset "Handwriting Data to Detect Alzheimer’s Disease" obtenido de Kaggle, por el autor Francesco Fontanella, que cuenta con una licencia “Commons Attribution 4.0 International (CC BY 4.0)”.

Este dataset cuenta con 450 features o características que presentan 174 adultos mayor (que padecen o no Alzheimer) al momento de escribir, con características como cuántas veces el bolígrafo tocó la hoja, promedio de la presión, el tiempo que le llevó escribir, entre otras.

Este proyecto se centrará en determinar si un paciente padece o no Alzheimer a partir de los datos obtenidos de su manera de escribir, siendo este un problema de clasificación binaria.

II. Análisis del desempeño del modelo “Random Forest Classifier” para el set de datos

Al ser el Alzheimer una enfermedad compleja con múltiples factores, se necesita un modelo de clasificación que pueda manejar conjuntos de datos con muchas características, por lo que se eligió el modelo “Random Forest Classifier” para realizar esta tarea de clasificación binaria.

Para desarrollar este modelo, se utilizaron distintos frameworks como “sklearn.preprocessing” y “matplotlib”. Además de librerías como “Pandas”, y “Numpy”.

Se debe conocer que para este caso que la columna "class" del paciente contendrá "P"para determinar que el paciente padece Alzheimer o "H" si no lo padece. Por lo

tanto, la columna “class” será el target que se desea, y sus valores “P” tomarán el valor de 1, y los valores “H” que estén presentes tomarán el valor de 0.

El seleccionar las variables para este modelo es muy importante, debido a que el dejar muchas características o features puede llevar a un sobreajuste del modelo, u overfitting, y hacer que este no generalice bien, sino que memorice las respuestas que estará dando. Por lo tanto, se hará un análisis de las variables presentes, para seleccionar las que mejor predicen si el paciente padece Alzheimer.

Primeramente, se hará un análisis estadístico, implementando un modelo de regresión lineal múltiple utilizando RStudio para todas las variables o features almacenados en el dataframe que se cargó desde el csv descargado desde Kaggle, con el fin de observar si existe o no singularidad.

```
Call:
lm(formula = class ~ . - class, data = df)

Residuals:
ALL 174 residuals are 0: no residual degrees of freedom!

Coefficients: (277 not defined because of singularities)
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)          -1.050e+02        NaN     NaN      NaN
air_time1             3.253e-04        NaN     NaN      NaN
disp_index1           3.975e+05        NaN     NaN      NaN
gmrt_in_air1         -8.944e-03        NaN     NaN      NaN
gmrt_on_paper1       -7.470e-03        NaN     NaN      NaN
max_x_extension1     -3.473e-03        NaN     NaN      NaN
max_y_extension1     -1.806e-03        NaN     NaN      NaN
mean_acc_in_air1     -3.919e+01        NaN     NaN      NaN
mean_acc_on_paper1   -2.169e+02        NaN     NaN      NaN
mean_gmrt1                   NA         NA      NA       NA
mean_jerk_in_air1     2.436e+02        NaN     NaN      NaN
mean_jerk_on_paper1   1.487e+03        NaN     NaN      NaN
mean_speed_in_air1   -1.561e+00        NaN     NaN      NaN
mean_speed_on_paper1  9.448e+00        NaN     NaN      NaN
num_of_pendown1       7.782e-03        NaN     NaN      NaN
paper_time1          -9.254e-04        NaN     NaN      NaN
pressure_mean1       -3.894e-02        NaN     NaN      NaN
pressure_var1        -1.275e-05        NaN     NaN      NaN
total_time1                  NA         NA      NA       NA
air_time2             3.143e-03        NaN     NaN      NaN
disp_index2          -4.035e+04        NaN     NaN      NaN
gmrt_in_air2          2.571e-02        NaN     NaN      NaN
gmrt_on_paper2        7.501e-03        NaN     NaN      NaN
max_x_extension2     -1.600e-03        NaN     NaN      NaN
max_y_extension2      2.519e-04        NaN     NaN      NaN
mean_acc_in_air2      9.037e+01        NaN     NaN      NaN
mean_acc_on_paper2   -2.213e+02        NaN     NaN      NaN
mean_gmrt2                   NA         NA      NA       NA
mean_jerk_in_air2    -5.087e+02        NaN     NaN      NaN
mean_jerk_on_paper2  -1.533e+03        NaN     NaN      NaN
mean_speed_in_air2    1.378e+00        NaN     NaN      NaN
mean_speed_on_paper2 -3.510e+00        NaN     NaN      NaN
num_of_pendown2      -4.261e+00        NaN     NaN      NaN
paper_time2          -1.184e-03        NaN     NaN      NaN
pressure_mean2        3.831e-02        NaN     NaN      NaN
pressure_var2         8.996e-05        NaN     NaN      NaN
total_time2                  NA         NA      NA       NA
```

Figura 1. Resultado del modelo de regresión lineal

Se puede observar que existe singularidad y que los residuales son todos de 0, además se obtuvo un Multiple R-squared de 1, por lo que si se utilizan todas las variables para la creación e implementación del modelo existirá un overfitting, y el modelo no logrará generalizar, sino únicamente memorizar las respuestas. Esto se genera debido a que hay

muchas variables predictoras que se correlacionan entre sí, por lo que se debe analizar cuáles son estas, y eliminarlas.

Para evitar esta situación y obtener variables predictoras útiles para generalizar, se procederá a hacer el cálculo de una matriz de correlación entre las variables predictoras, e identificar las que se correlacionan entre sí. Si existen variables predictoras altamente correlacionadas, se evaluará cuál de las dos tiene una mayor correlación con "class", que sería el diagnóstico, y se guardará solamente en un dataframe las variables con mayor correlación con el target.

Habiendo hecho esto, quedó un dataframe con únicamente 93 features (antes contaba con 450), y se volvió a implementar el modelo de regresión lineal múltiple para observar si aún existe colinealidad entre las variables.

```
mean_speed_on_paper15 -1.206e-01  1.265e-01  -0.953   0.3434
mean_jerk_on_paper15   8.530e+00  1.657e+01   0.515   0.6081
mean_jerk_in_air15     4.075e-01  9.066e-01   0.449   0.6542
air_time16             1.257e-05  6.453e-06   1.948   0.0548 .
num_of_pendown16      -8.829e-03  1.434e-02  -0.616   0.5396
paper_time16          -2.006e-05  2.861e-05  -0.701   0.4851
disp_index16           1.349e+04  2.621e+04   0.515   0.6082
total_time16                  NA         NA      NA       NA
mean_gmrt16            1.267e-03  6.995e-04   1.811   0.0737 .
mean_jerk_in_air16    -5.786e-01  5.392e-01  -1.073   0.2863
mean_gmrt17           -1.005e-03  1.712e-03  -0.587   0.5589
mean_acc_in_air17     -1.012e-01  1.139e-01  -0.888   0.3768
mean_speed_in_air17    5.029e-02  6.783e-02   0.741   0.4605
max_x_extension18     -9.604e-05  8.035e-05  -1.195   0.2353
mean_gmrt18           -1.861e-06  5.204e-04  -0.004   0.9972
mean_speed_on_paper18 -1.234e-01  1.248e-01  -0.989   0.3253
gmrt_on_paper18        2.191e-03  2.347e-03   0.933   0.3532
mean_gmrt19            2.972e-03  2.849e-03   1.043   0.2998
mean_speed_in_air19   -8.697e-01  3.348e-01  -2.598   0.0111 *
gmrt_in_air19          1.656e-02  7.626e-03   2.172   0.0326 *
mean_gmrt20            5.355e-04  1.153e-03   0.464   0.6437
gmrt_on_paper20        2.400e-03  3.723e-03   0.645   0.5208
mean_speed_on_paper20 -1.145e-01  1.861e-01  -0.615   0.5402
mean_gmrt25           -1.283e-03  2.227e-03  -0.576   0.5660
gmrt_in_air25         -2.099e-03  3.756e-03  -0.559   0.5777
mean_speed_in_air25    1.379e-01  1.954e-01   0.706   0.4822
air_time21             9.486e-06  6.321e-06   1.501   0.1371
num_of_pendown21      -2.002e-04  6.137e-03  -0.033   0.9741
total_time21          -4.063e-06  2.603e-06  -1.561   0.1223
mean_jerk_in_air21     5.685e-02  1.998e-01   0.285   0.7766
mean_gmrt22           -2.191e-04  1.848e-03  -0.119   0.9059
gmrt_on_paper22        2.678e-03  3.395e-03   0.789   0.4323
mean_speed_on_paper22 -1.481e-01  1.948e-01  -0.760   0.4492
mean_speed_in_air23   -1.445e-02  1.581e-01  -0.091   0.9274
gmrt_in_air23         -2.363e-03  3.430e-03  -0.689   0.4927
mean_gmrt23            1.476e-03  1.662e-03   0.888   0.3770
mean_speed_in_air24   -7.264e-02  1.806e-01  -0.402   0.6885
gmrt_in_air24          2.340e-03  4.052e-03   0.578   0.5651
mean_gmrt24           -1.011e-03  1.580e-03  -0.640   0.5242
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3415 on 85 degrees of freedom
Multiple R-squared:  0.7719,    Adjusted R-squared:  0.5358
F-statistic: 3.269 on 88 and 85 DF,  p-value: 5.035e-08
```

Figura 2. Resultado del modelo de regresión lineal con el nuevo dataframe

Teniendo en cuenta que esta vez se tuvo un valor de Adjusted R-squared de 0.53, se puede inferir que el modelo de regresión lineal múltiple con esas variables seleccionadas puede explicar el 53% de la variabilidad en la variable de respuesta, ajustado por el número de predictores en el modelo. Aunque ahora se observa que existe underfitting, ya que sólo explica la mitad de las predicciones del modelo con estas

variables. Por lo tanto, se utilizará el "criterio de información de Akaike" para obtener la menor cantidad de variables que puedan explicar la variabilidad en la variable de respuesta, con la ayuda de RStudio.

```
Call:
lm(formula = class ~ gmrt_on_paper1 + paper_time2 + mean_gmrt4 +
    air_time5 + air_time6 + paper_time8 + mean_gmrt6 + mean_gmrt7 +
    mean_speed_on_paper7 + total_time10 + mean_gmrt10 + gmrt_on_paper10 +
    mean_gmrt11 + pressure_mean10 + air_time18 + total_time18 +
    total_time13 + gmrt_on_paper14 + mean_speed_on_paper14 +
    mean_speed_on_paper15 + air_time16 + num_of_pendown16 + mean_gmrt16 +
    mean_acc_in_air17 + max_x_extension18 + mean_speed_on_paper18 +
    gmrt_on_paper18 + mean_speed_in_air19 + gmrt_in_air19 + gmrt_in_air25 +
    mean_speed_in_air25 + air_time21 + total_time21 + gmrt_on_paper22 +
    mean_speed_on_paper22 + gmrt_in_air23 + mean_gmrt23 + paper_time9 +
    gmrt_on_paper15, data = selected_df2)
```

Figura 3. Variables a utilizar para la predicción

Las variables seleccionadas por el criterio de información de Akaike se observan en la Figura 3. Para comprobar que se tiene un mejor rendimiento y economía de las variables predictoras con los features seleccionados, se volvió a calcular el modelo de regresión lineal múltiple pero con las variables predictoras seleccionadas.

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2911 on 134 degrees of freedom
Multiple R-squared:  0.7387,    Adjusted R-squared:  0.6627
F-statistic: 9.716 on 39 and 134 DF,  p-value: < 2.2e-16
```

Figura 4. Regresión lineal con nuevas variables

Se puede observar en la figura 4 que con las 39 variables predictoras seleccionadas, el "Adjusted R-Squared" es de 0.6627 y es mayor que cuando se tenían 95 features para el modelo de regresión lineal múltiple calculado por RStudio, y a su vez no se genera un overfitting, se puede inferir que estas variables son las correctas para continuar con el modelo de clasificación binaria.

A partir de esto se procederá a crear el modelo de clasificación "Random Forest Classifier" con las variables predictoras seleccionadas, utilizando Python.
Inicialmente se crea un dataframe "X" que contendrá las variables predictoras o features que fueron seleccionadas por medio del criterio de información de Akaike.
Además, se creará un dataframe para "y" que tendrá la clase o diagnóstico del paciente con valores de 0 o 1 como se mencionó previamente.

a) Separación y evaluación del modelo con un conjunto de prueba y un conjunto de validación

Teniendo tanto el dataframe para los features como el dataframe para el target, se dividieron los datos, 90% para entrenar y 10% para probar, y se implementó la función "stratify= y" para mantener las mismas proporciones de clases en los conjuntos divididos. Además, se utilizó un random seed de 10 un valor o semilla como el punto de inicio para la función random.

Posteriormente, se creó una instancia de bosque aleatorio que utilizó la función "RandomForestClassifier" de "sklearn", utilizando gini, un número de estimadores de 5 y 2 núcleos para mejorar la eficiencia al procesar el modelo.
Luego, se entrenó el modelo y se midió el performance del mismo obteniendo el accuracy para train y para test. Se probó el modelo, pero ahora contando con un set de validación mediante el uso de "cross validation". Para este caso se utilizó K-Folds, que iteró 10 veces, permitiendo una evaluación más robusta del modelo utilizando múltiples divisiones de datos y proporcionando una estimación más confiable de cómo el modelo se comportará si se le ingresan nuevos datos que no ha visto.
Se midió el accuracy obtenido sobre las muestras apartadas, lo que quiere decir que se hicieron 10 entrenamientos independientes y el accuracy final fue el promedio de las 10 accuracies anteriores, permitiendo observar qué tan bueno es el modelo en la práctica.

```
Train accuracy: 0.955
Test accuracy: 0.722
Cross validation: 0.8091666666666667
Precision: 72.50%
Recall: 72.22%
```

Figura 5. Resultados obtenidos para la evaluación del modelo con un conjunto de prueba y un conjunto de validación

Se puede observar en la figura 5 que para el entrenamiento se tuvo un accuracy del 95.5%, lo cuál es bastante bueno. Sin embargo, para el conjunto de pruebas se obtuvo un accuracy del 72.2% (que es una diferencia significativa con el conjunto de entrenamiento), que habla de que es posible que exista un overfitting, y que el modelo no está generalizando realmente bien. Para el conjunto de validación se obtuvo un 80.9% en el promedio de los puntajes obtenidos en el cross validation, lo que significa que el modelo generaliza de una buena manera con datos que no había visto antes.

Con estos valores se puede notar que el modelo está captando el ruido y memorizando, no generalizando realmente.

Se obtuvo además un buen desempeño para precisión y recall, por lo que existen una cantidad no muy grande de falsos negativos y falsos positivos, se observa equilibrado, lo que permite inferir que el modelo no está sesgado a hacer predicciones positivas o negativas incorrectas o falsas.

b) Diagnóstico y explicación el grado de bias o sesgo

Para poder visualizar el grado de bias se realizó una gráfica con la función "learning curve" de sklearn, utilizando el modelo creado, un set de validación de KFolds que iteró 10 veces con fracciones del conjunto de datos total, desde el 10% hasta el 100%.
Se fueron calculando y almacenando los puntajes para calcular las medias y desviaciones estándar de los mismos y se obtuvo una gráfica de la curva de aprendizaje.

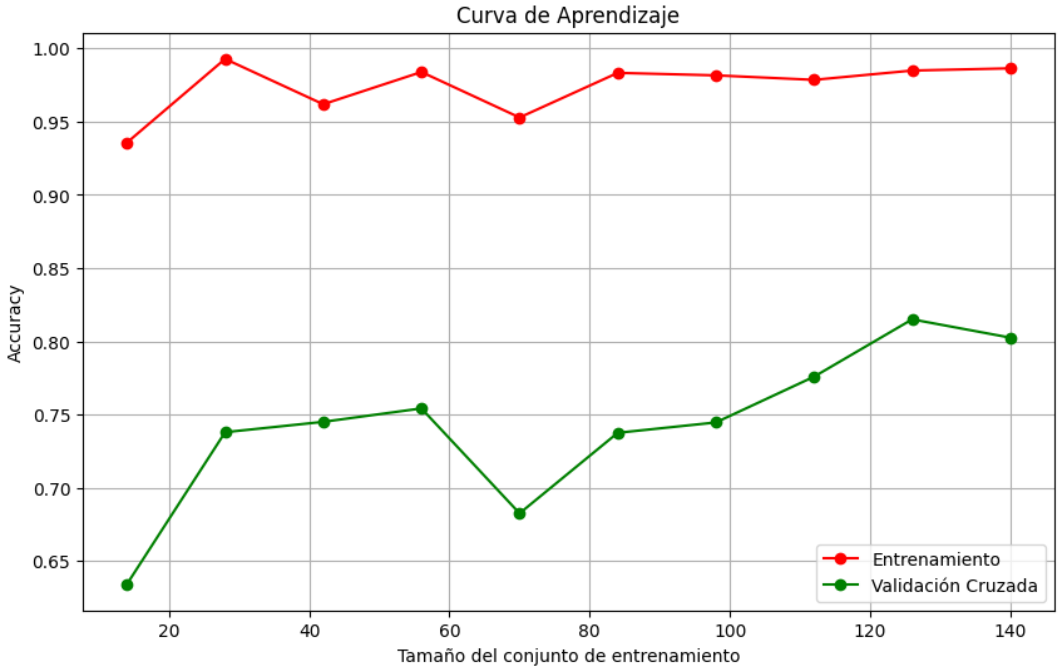

Figura 6. Curva de aprendizaje del modelo

A partir de la gráfica observada en la figura 6, ambas líneas (tanto para el conjunto de entrenamiento como para el conjunto de validación) se observan que están por encima de 0.5 (la mitad del accuracy), por lo que el bias cuenta con un grado medio, ya que por ejemplo, cuando el set de entrenamiento está en 1 de accuracy, el set de validación se encuentra en 0.70 de accuracy, y su máximo valor de accuracy fue de 0.82 lo que implica que el modelo ha capturado ciertos patrones del conjunto de entrenamiento pero no generaliza todo correctamente, ya que existe una diferencia notoria entre el valor predicho y el valor real, pero no exagerado o menor al 60%.

c) Diagnóstico y explicación el grado de varianza

Tomando como referencia la figura 6, el accuracy para el set de entrenamiento no alcanzó el valor de 1, sino que únicamente se aproximó a este. Para el set de validación su máximo valor de accuracy fue de 82%, además de que los valores para el set de entrenamiento estuvieron en 95.5% de accuracy, y 72.2% para el set de pruebas, por lo que con esto se infiere que la varianza posee un grado medio, y está capturando ruido en los datos, pero sin memorizar completamente.

d) Diagnóstico y explicación el nivel de ajuste del modelo

Como se mencionó anteriormente, el set de entrenamiento solo alcanzó a estar muy cercano de 1 en una ocasión, pero no en todo el proceso. A su vez, el set de validación no obtuvo valores muy bajos para accuracy como para considerar underfit, sino que está en un rango entre 70% a 80%, por lo que se considera que logró ubicar patrones y realizar predicciones a nuevos datos. Sin embargo, como existe una diferencia significativa entre el accuracy para el set de entrenamiento y para el set de pruebas, se puede inferir que existe overfitting.

e) Técnicas de regularización o ajustes de parámetros

Para lograr disminuir este overfitting y lograr que el modelo generalice de una buena manera se implementó la función "Grid Search" con el fin de ajustar los hiperparámetros del modelo. Este genera todas las posibles combinaciones de valores de hiperparámetros a partir del espacio de búsqueda definido en "grid_space".

Para "max_depth", se utilizaron los valores "30, 40, 50, None", y para "n_estimators", se utilizaron los valores de "200, 300, 400", para min_samples_leaf se utilizaron los valores "1, 2", y por último para "min_samples_split" se utilizaron los valores de "5,10,15". Para cada combinación de hiperparámetros, se entrenó un modelo de Random Forest utilizando estos en un conjunto de datos de entrenamiento y se evaluó su rendimiento mediante cross validation con Kfolds (de 10 iteraciones), para ver el desempeño del modelo con datos que no ha aprendido.

```
Inicializando grid space
Calculando grid search
Modelo calculado, entrenando modelo
Best hyperparameters are: {'max_depth': 30, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 300}
Best score is: 0.8143790849673203
```

Figura 7. Resultados para Grid Search

Tomando en cuenta la figura 7, después de realizar el Grid Search se encontraron los valores óptimos para los hiperparámetros, con un puntaje para el conjunto de validación del 81.43%, teniendo un "max_depth" de 30, un "min_sample_leaf" de 1, un "min_sample_split" de 5 y un valor de 300 para "n_estimators".

f) Mejora del desempeño del modelo

Posteriormente, se agregaron estos hiperparámetros al modelo "Random Search Classifier", se volvió a entrenar el modelo con estos nuevos valores otorgados por el Grid Search, y se obtuvieron los siguientes resultados.

```
Train accuracy: 1.000
Test accuracy: 0.833
Cross validation: 0.8779166666666667
Precision: 83.75%
Recall: 83.33%
```

Figura 8. Resultados para el modelo con nuevos hiperparámetros

Con estos nuevos hiperparámetros el accuracy para el set de entrenamiento aumentó al 100%, y el accuracy para el set de pruebas aumentó significativamente al 83%.

Para el set de validación se obtuvo un buen rendimiento por lo que se puede inferir que el modelo está generalizando bien y no está captando ruido como lo hacía previamente, ni tampoco memorizando las respuestas que da. La precisión y el recall aumentaron a su vez significativamente, pero siguen estando equilibradas, dando una certeza ambos de un poco más del 83% de datos correctamente predichos, indicando con el "recall" que el modelo es efectivo para identificar la mayoría de los casos positivos verdaderos en el conjunto de datos, y con "precision" teniendo un bajo número de casos positivos

falsos, lo que es bastante importante para este caso que se requiere una buena predicción para diagnosticar a los pacientes que puedan padecer Alzheimer.

Para observar el comportamiento del modelo con los nuevos hiperparámetros se generó una nueva gráfica de la curva de aprendizaje.

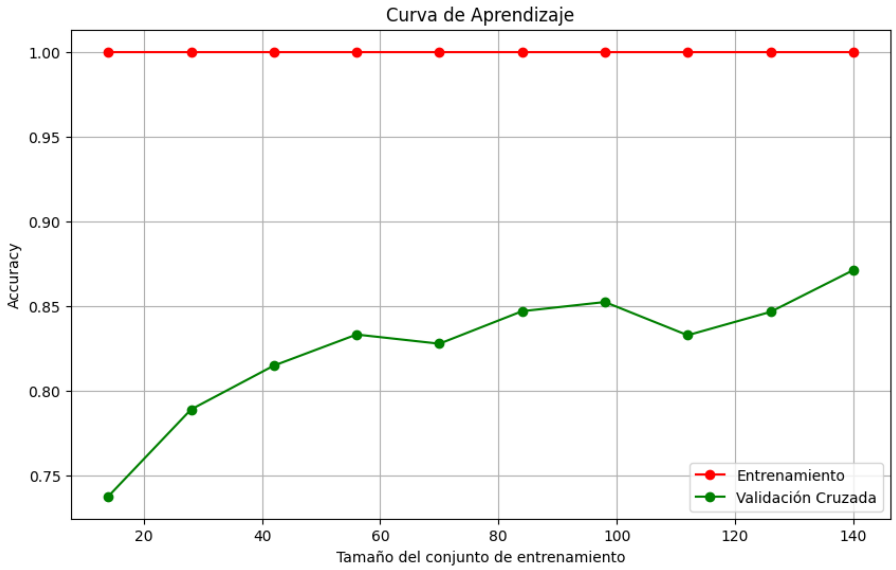

Figura 9. Curva de aprendizaje para el modelo con nuevos hiperparámetros

Se puede notar que para el set de entrenamiento se obtuvo un accuracy perfecto en las predicciones, y que para el set de cross validation poco a poco fue aumentando el accuracy, aunque inició mucho más alto que anteriormente, en aproximadamente un 73% de accuracy, lo que habla de la mejora del modelo.

Para el set de validación el puntaje fue aumentando progresivamente. Como los valores para el set de entrenamiento estuvieron en todo momento en 100% de accuracy y para el set de validación cruzada estuvieron en gran parte mayores que 80% de accuracy, entonces se observa que hay un bias medio (la diferencia entre la predicción media del modelo y el valor real no es alta, pero tampoco es mínima), y que el modelo está aprendiendo bien de los datos de entrenamiento, generalizando de manera efectiva nuevos datos. Por otra parte, al tener valores tanto para train como para test muy buenos (mayores al 80%), se puede inferir que existe una baja varianza, logrando el objetivo de tener valores bajos o medios para varianza y bias, y tener predicciones reales. Con todo esto se puede inferir que el modelo no sufre de overfitting, que generaliza bien, y que los parámetros encontrados fueron buenos para ajustar el modelo, por lo que se considera que el modelo cuenta con un buen fit.

III. Conclusión

A lo largo de este proyecto se hicieron distintas pruebas, iniciando con un análisis estadístico para el preprocesamiento de las variables predictoras o features, permitiendo

la reducción de la cantidad de las mismas para evitar el sobreajuste del modelo. Se encontraron las variables predictoras que ayudaron a identificar los mejores patrones para el diagnóstico de un paciente, y ayudar a conocer si este sufre o no de Alzheimer a partir de su manera de escribir en un papel. Posteriormente se generó un modelo de clasificación "Random Forest Classifier" que con los primeros parámetros que se le dieron generaba un sobreajuste y no generalizó bien, por lo que no era un buen modelo para un caso como el planteado, del sector salud.

A pesar de esto, se hizo un ajuste de hiperparámetros con la ayuda del "Grid Search", logrando mejorar el modelo a que tuviera un bias medio y una baja varianza, concluyendo que el modelo cuenta con un buen fit para el caso planteado.

Referencias

Juan Moisés de la Serna. (2021). Cómo afecta el Alzheimer a la escritura. Recuperado de https://juanmoisesdelaserna.es/como-afecta-el-alzheimer-a-la-escritura/

CDinc Barcelona. (2023). Enfermedad de Alzheimer y otras enfermedades neurodegenerativas. Recuperado de https://cdincbarcelona.com/es/especialidades-cdinc/enfermedad-alzheimer-enfermedades-neurodegenerativas/