# PREDICCIÓN DE PATOLOGÍAS ASOCIADAS A LA TIROIDES

**POR:**

Sara Camila Guarín Castillo, Carlos Hernan Molina Bustos

**MATERIA:**

Introducción a la inteligencia artificial

**PROFESOR:**

Raul Ramos Pollan


UNIVERSIDAD DE ANTIOQUIA

FACULTAD DE INGENIERÍA

MEDELLÍN 2022

## *1. Introducción*

*Planteamiento del problema*

Según la Organización Mundial de la Salud (OMS), más de 750 millones de personas en el mundo padecen algún tipo de patología relacionada con la tiroides y se cree que aproximadamente el 60% lo desconoce.

Se sabe que el funcionamiento inadecuado de la glándula tiroidea puede generar las siguientes patologías: hipotiroidismo, hipertiroidismo, nódulo único sólido o quístico, bocio multinodular, tiroiditis, o cáncer de tiroides. Si el paciente se hace un diagnóstico temprano y conoce su condición, un tratamiento adecuado puede determinar un buen pronóstico en la mayoría de los casos[1]; por tal motivo, se desea desarrollar un modelo que permita calcular la probailidad de que una persona pueda llegar a padecer alguna patología relacionada a la enfermedad de la tiorides (hipertiroidismo o hipotiroidismo) de acuerdo a los siguientes factores de riesgo: edad, sexo y niveles de TSH.

*Dataset*

El dataset a utilizar para el desarrollo del modelo proviene de una competencia de Kaggle, el cual se creó mediante la conciliación de un conjuntos de datos de enfermedades de la tiroides proporcionado por UCI Machine Learning Repository, los cuales fueron extraídos directamente de Garvan Institute of Medical Research. Este dataset consta de un archivo llamado **thyroidDF.csv** el cual proporciona la información personal y médica necesaria para la creación del modelo, dentro de la cual se tiene la edad, el sexo e información relevante sobre la tiroides[2].

La descripción detallada del dataset se muestra a continuación:

- **age** – edad del paciente (int)
- **sex** – sexo del paciente (str)
- **on_thyroxine** – si el paciente está tomando tiroxina (bool)
- **query on thyroxine** - *si el paciente está tomando tiroxina (Consulta) (bool)
- **on antithyroid meds** - si el paciente está tomando medicamentos antitiroideos (bool)
- **sick** - si el paciente está enfermo (bool)
- **pregnant** - si la paciente está embarazada (bool)
- **thyroid_surgery** - si el paciente se ha sometido a una cirugía de tiroides (bool)
- **I131_treatment** - si el paciente está en tratamiento con I131 (bool)
- **query_hypothyroid** - si el paciente cree que tiene hipotiroidismo (bool)
- **query_hyperthyroid** - si el paciente cree que tiene hipertiroidismo (bool)
- **lithium** - si el paciente * litio (bool)
- **goitre** - si el paciente tiene bocio (bool)
- **tumor** - si el paciente tiene tumor (bool)
- **hypopituitary** - si el paciente * glándula hiperpituitaria (float)
- **psych** - si paciente * psych (bool)

- **TSH_measured** - si se midió la TSH en la sangre (bool)
- **TSH** - nivel de TSH en la sangre del análisis de laboratorio (float)
- **T3_measured** - si se midió T3 en la sangre (bool)
- **T3** - Nivel de T3 en la sangre del trabajo de laboratorio (float)
- **TT4_measured** - si se midió TT4 en la sangre (bool)
- **TT4** - Nivel de TT4 en sangre del trabajo de laboratorio (float)
- **T4U_measured** - si se midió T4U en la sangre (bool)
- **T4U** - nivel de T4U en la sangre del análisis de laboratorio (float)
- **FTI_measured** - si FTI se midió en la sangre (bool)
- **FTI** - nivel de FTI en la sangre del análisis de laboratorio (float)
- **TBG_measured** - si se midió TBG en la sangre (bool)
- **TBG** - nivel de TBG en la sangre del análisis de laboratorio (float)
- **referral_source** - (str)
- **target** - diagnóstico médico de hipertiroidismo (str)
- **patient_id** - identificación única del paciente (str)

Como se puede apreciar, el dataset contiene 31 columnas y alrededor de 9172 observaciones, cumpliendo con los requisitos solicitados al escoger el dataset; así mismo, contiene al menos el 10% de columnas categóricas y se simularán aleatoriamente un 5% de datos faltantes en al menos 3 columnas.

*Métricas*

Para la métrica de machine learning se desea implementar *Logloss*, que se calcula mediante la siguiente expresión:

$$Logloss = -\frac{1}{N}\sum_{i=1}^{N}[y_i ln p_i + (1 - y_i)\, ln(1 - p_i)]$$

Donde *N* es el número de observaciones, *i* es la observación dada, *y* es el valor actual positivo, *p* es la probabilidad de predicción y *ln* es el logaritmo natural.

*Logloss* indica qué tan cerca está la probabilidad de predicción del valor real/verdadero correspondiente. Cuanto más diverge la probabilidad predicha del valor real, mayor será el valor de pérdida logarítmica[3].

Adicionalmente, se plantea emplear accuracy y *f1_score*; donde *accuracy* mide la exactitud del modelo (% de casos que el modelo ha acertado) y *f1_score* combina la precisión y la exhaustividad(*recall*) en un solo valor y se calcula haciendo la media armónica entre la precisión y la exhaustividad. Para el modelo se tendrán en cuenta el *query_hypothyroid* y el *query_hyperthyroid* que contienen información de interés sobre las personas que presentan o presentaron alguna vez enfermedades de la tiroides, hipotiroidismo o hipertiroidismo, respectivamente.

*Desempeño*

En cuanto a la métrica de negocio, se tiene como interés que las predicciones sean precisas en la detección de hipotiroidismo e hipertiroidismo en pacientes; para ello

se espera que en la evaluación del dataset según la métrica de *Logloss* la pérdida logarítmica sea cercana a 0, de esta manera se puede garantizar que el funcionamiento del modelo es el esperado e identificar qué poblaciones son las que tienen mayor riesgo y así poder comparar estos resultados con los estudios realizados anteriormente[4][5].

## *2. Exploración descriptiva del Dataset*

Se realiza la carga de los datos del dataset desde Kaggle a un notebook de colab, a partir del cual se inicia el análisis exploratorio del dataset.

En el dataframe se identifican 9172 filas y 31 columnas con información relevante de pacientes que tienen o han presentado enfermedades asociadas a la tiroides y se realiza una observación de toda la información contenida en este. Se hace una revisión del número de datos faltantes en el dataframe y se confirma que existen 14.629 valores *NaN* correspondientes al 5,136% del total de valores del dataframe (275.160). Se identifican las columnas que poseen valores nulos, encontrándose valores nulos en 7 de las columnas; luego se hace un promedio la cantidad de valores faltantes en cada uno de ellos y se observa que la columna TBG tiene la mayoría de sus valores nulos (96, 823%), por lo que se procede a eliminar dicha columna. Según lo anterior, el dataset cumple con las condiciones requeridas en cuanto a número de registros y columnas para su estudio.

De acuerdo con la información contenida en el dataframe, se identifican dos columnas de interés correspondientes a *query_hypothyroid* y *query_hyperthyroid*, las cuales son los registros booleanos de pacientes que presentan o no una enfermedad asociada a hipo o hipertiroidismo, donde true indica la presencia de la enfermedad y false su ausencia. Conocidas estas columnas, se indaga sobre la cantidad de pacientes que presentan o no una de las enfermedades de la tiroides mencionadas y se tiene que 630 pacientes presentan hipotiroidismo y 651 presentan hipertioidismo.

Debido a que *query_hypothyroid* y *query_hyperthyroid* son datos categóricos, para facilitar su estudio, se desea convertirlos en datos numéricos; es por ello que, mediante un mapeo, se convierten los valores *false* en 0 y *true* en 1.
Cabe aclarar que constantemente se realiza conteo y suma de los datos que se procesan para verificar que estos no se pierden.

Se realizan conteos de otras variables que son importantes para identificar la presencia de enfermedades de tiroides como lo son el embarazo, de la cual se tienen 9065 pacientes en estado de embarazo.

Debido a que existe interés en conocer cuántas mujeres y hombres tienen hipo o hipertiroidismo, se realizan varias agrupaciones de la siguiente manera:

Para dar una mejor interpretación de las variables que se estudian, se realizan gráficos que ayudan visualmente a su comprensión.

En la *Figura 1* se observa que la mayoría de los registros corresponden a mujeres pues, en el dataframe se tiene un conteo de 6073 mujeres y 2792 hombres.

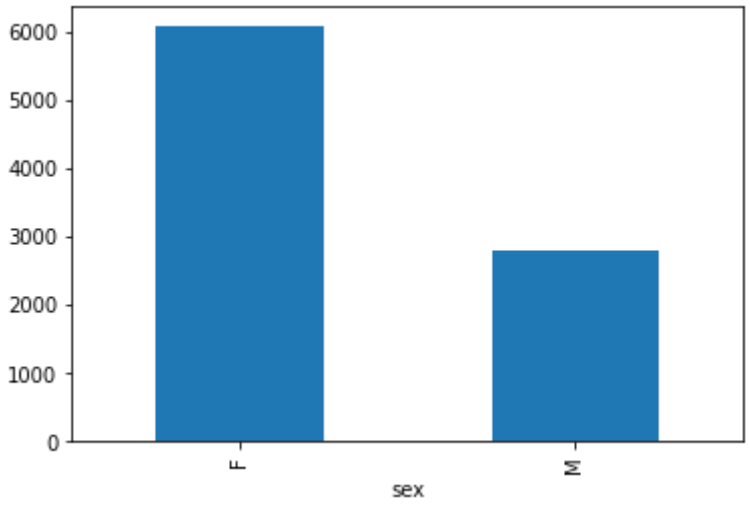


*Figura 1. Gráfico de barras indicando el sexo de los pacientes registrados en el dataframe.*

Además, se realiza un promedio de la edad según el sexo y como se observa en la *Figura 2* se tiene que el promedio de edad de las mujeres es aproximadamente 63 años y de los hombres es 99 años, indicando que los hombres elegidos para el estudio tienen una edad avanzada.

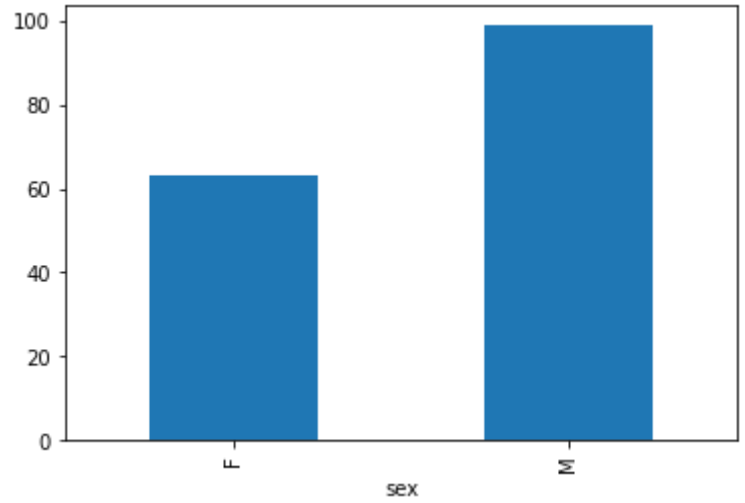


*Figura 2. Gráfico de barras indicando el promedio de edad de los pacientes registrados en el dataframe.*

Los registros de las columnas categóricas son reemplazados por valores binarios con el fin de hacer el tratamiento de estos datos en el modelado.

Ahora bien, las columnas con registros de tipo *object* son convertidos a datos numéricos.

Con los datos de las columnas, se visualizan histogramas para visualizar patrones o tendencias en los datos, como se observa en la *Figura 3*. Donde todas las columnas presentan datos recopilados en 0 y 1, excepto aquellos que tienen valores numéricos diferentes como *T3, TT4, T4U* y *FTI l*os cuales muestran una distribución normal. Sin embargo, la edad no se comportó igual, pues los valores estuvieron concentrados en una sola barra, variable para la cual se espera una distribución normal, ver la *Figura 4*.

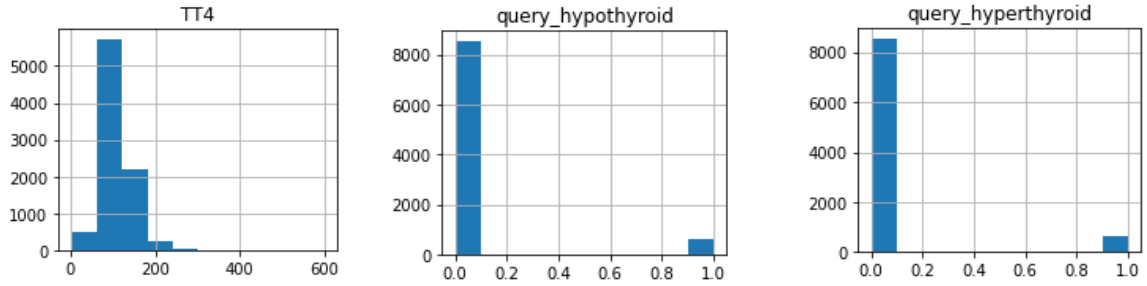


*Figura 3. Histogramas para sex, query_hypothyroid, query_hyperthyroid.*

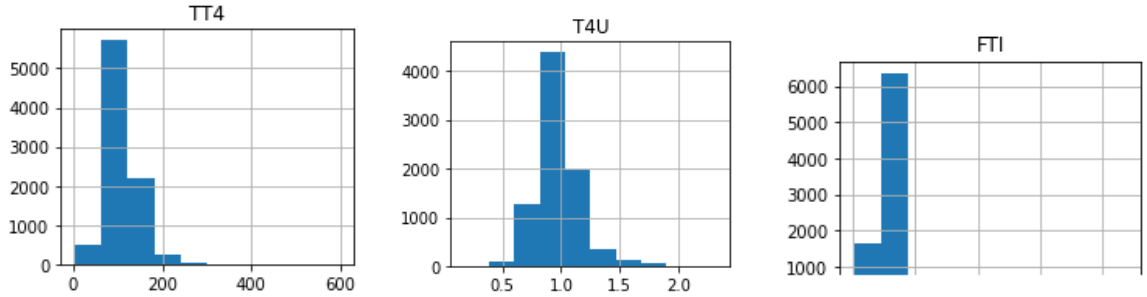


*Figura 4. Histogramas para TT4, T4U y FTI.*

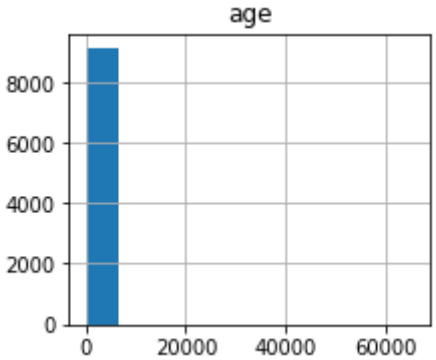

*Figura 5. Histograma para la edad.*

## 3. *Iteraciones de desarrollo*

*Modelo de predicción utilizando deep learning - Redes neuronales*

Para llevar a cabo este modelo*[6]*, se tiene en cuenta una consideración importante, ya mencionada anteriormente relacionada con la gráfica de la edad, la cual no mostraba una distribución normal. Para ello, se hace una búsqueda de edades mayores a los 100 años en el dataframe y se encuentran 4 datos que no tienen concordancia con el estudio, estos son edades de 455, 65511, 65512 y 65526 años y se procede a convertir estos valores en datos nulos.
En la figura se observa un rango de edades entre 1 y 100 años de edad, mediante una distribución normal de estos datos.

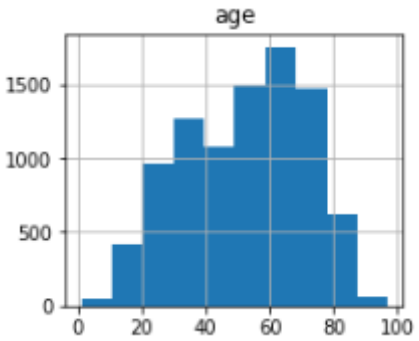


*Figura 6. Histograma para la edad.*

Cabe aclarar que el modelo descrito a continuación fue implementado incluyendo los valores de edad que no tenían concordancia con el estudio y los valores de las métricas mostraban un modelo perfecto, con accuracy de 100%, debido a esto se tomó la decisión de revisar el modelo y al anular estos valores de edad superiores a 100, se mostraron resultados con mayor acercamiento a la realidad del estudio realizado.
Una vez resuelto el problema de las edades, se procede a reemplazar los valores nulos por la media para cada columna de *age, T4U, TSH, T3, TT4, FTI*. Se realiza un chequeo y efectivamente no hay ningún dato nulo en las columnas, ahora los datos están limpios y con el formato correcto para iniciar el estudio.

Se realiza el modelo haciendo uso de deep learning, utilizando redes neuronales para predecir el hipotiroidismo(objetivo) e hipertiroidismo(objetivo) en función de los datos de entrada(características). Para ello, se seleccionan todas las columnas excepto *query hypothyroid* o *query_hyperthyroid* según el estudio sea para predecir hipotiroidismo o hipertiroidismo y a parte se selecciona únicamente la columna del target.

Para evaluar el modelo eficientemente, este se entrena con el 80% de los datos y se conserva el 20% restante para evaluar el modelo. Es por ello que se utiliza *train_test_split* que permite dividir el dataset en dos bloques, uno destinado al entrenamiento(train) y otro a la validación del modelo(test)
Con el objetivo de normalizar los datos dentro de un rango particular y acelerar los cálculos en el algoritmo, se escalan los datos de train y test mediante la clase *StandardScaler* de scikit-learn. El dataset tiene 27 columnas por lo que la dimensión de entrada es especificada como *input_dim = 27*, se utiliza la función de activación de ReLU en la capa oculta y la función sigmoide para la capa de salida y así devolver un valor binario de salida, es decir, hipotiroidismo (1) o negativo (0) o para el otro caso hipertiroidismo (1) o negativo (0).

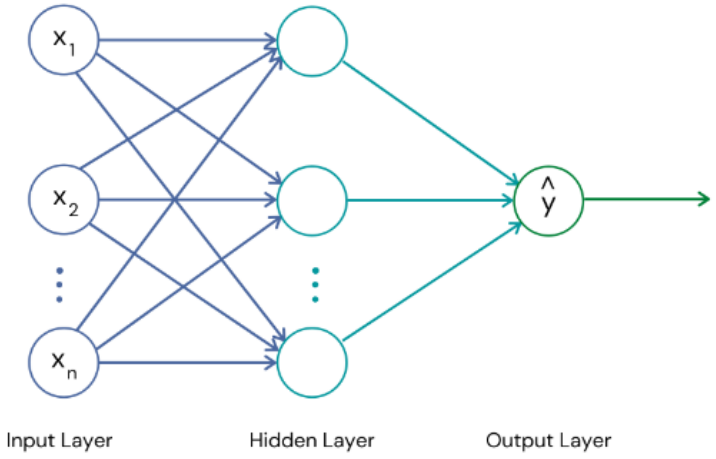


Figura 7. Redes neuronales.

Se procede a entrenar el modelo para predecir hipotiroidismo e hipertiroidismo y posteriormente se realiza la evaluación de este para obtener los valores de *loss*(pérdida) y *accuracy*(exactitud). De la evaluación con las métricas se tiene un *loss* de 37.39% y un accuracy de 91.95% para q*uery_hypothyroid* y un *loss* de 26.12%, *accuracy* de 91.88% para *query_hyperthyroid*.
Se calculan los porcentajes para *f1_score, recall y precision,* para los cuales se obtiene 4.91%, 2.63% y 36.36%, respectivamente para *query_hypothyroid* y 9.70%, 5.93% y 26.67%, respectivamente para *query_hyperthyroid*.

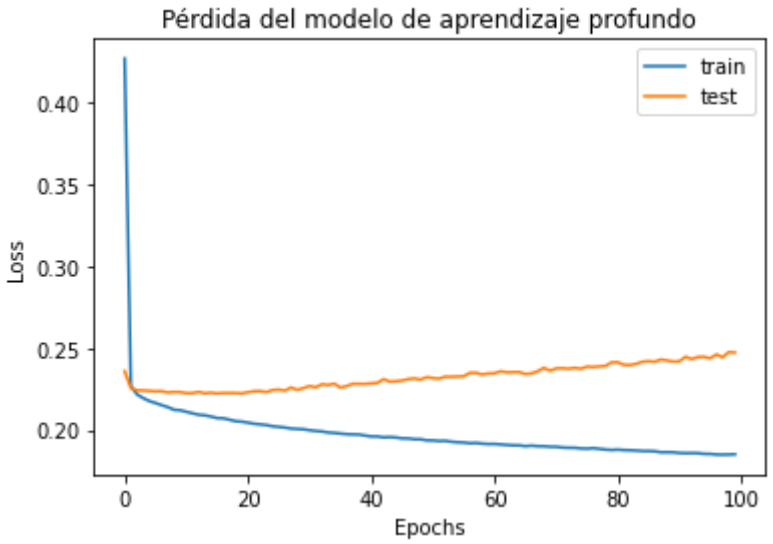


*Figura 8. Gráfica de pérdida del modelo para query_hypothyroid.*

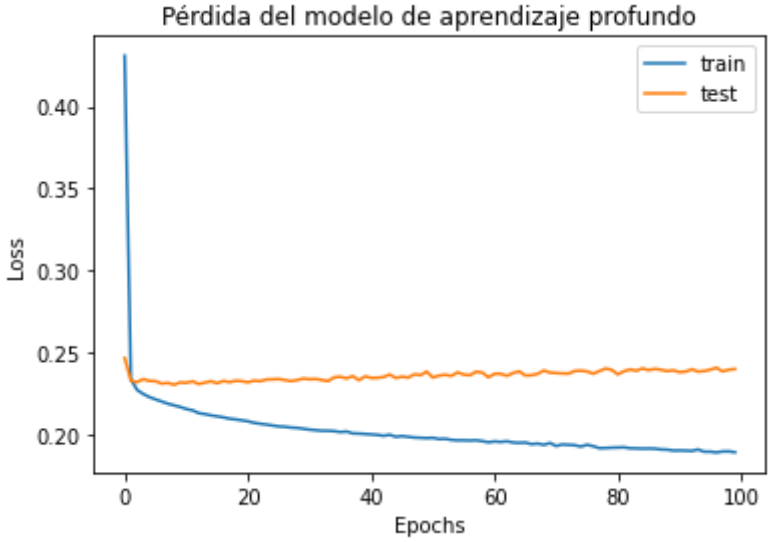


*Figura 9. Gráfica de pérdida del modelo para query_hyperthyroid.*

Buscando emplear un modelo que tenga un mejor rendimiento, se hace uso de XGBoost[7] el cual es uno de los algoritmos de machine learning de tipo supervisado más usados en la actualidad.

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[8] el cúal era el problema que se presentaba en los modelos de predicción de hipo e hipertiroidismo abordados anteriormente. Para el análisis se decide realizar las observaciones para los pacientes con diagnóstico negativo, hipotiroidismo e hipertiroidismo, considerando que estos son los datos más importantes para el estudio.

Se realiza un preprocesamiento de los datos, dentro del cual se eliminan los datos que son redundantes en el estudio y se hace una reasignación de valores al grupo de diagnóstico. Se realiza un remapeo para asignar el target al dataframe. Los valores de edad superiores a los 100 años son convertidos a valores nulos y se realizan múltiples gráficas de los valores numéricos de las hormonas vs el target como se observa en la siguiente figura.


Figura 10. Gráfica de atributos numéricos vs. Target.

Se procede a crear un diagrama de pares de las variables numéricas para detectar grupos que se formen entre las variables.
En las diagonales del gráfico de pares se pueden ver las distribuciones de cada variable numérica con respecto a la otra. Es evidente lo desequilibrado que está el conjunto de datos, con tantos "objetivos" negativos en comparación con hipotiroidismo o hipertiroidismo.

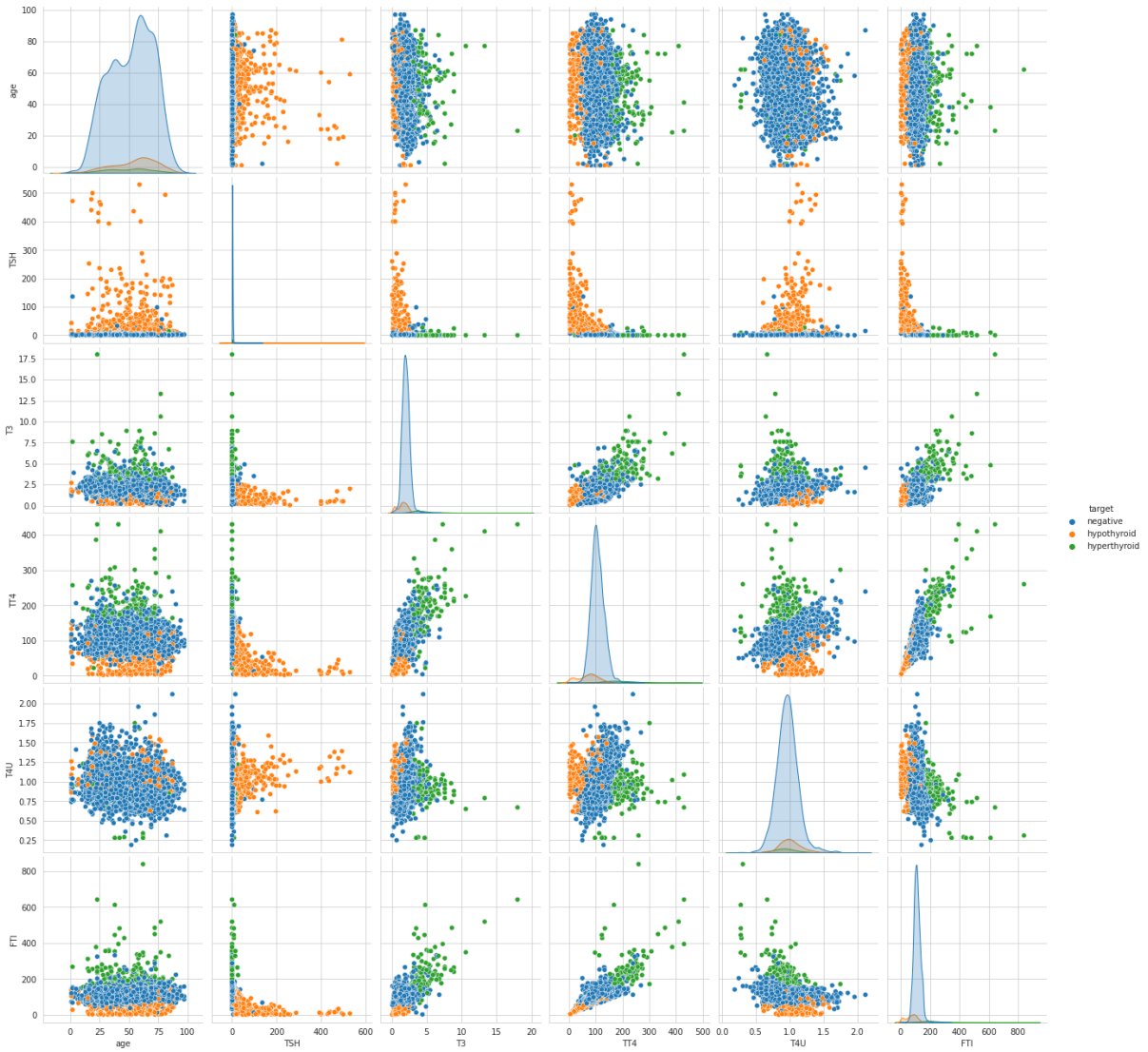


Figura 11. Diagrama de pares.

Se procede con la investigación de valores atípicos leves y severos dentro de los valores numéricos de cada una de las hormonas, mediante el código mostrado en la figura.

```
# TSH
Q1_TSH = thyroidDF['TSH'].quantile(0.25)
Q3_TSH = thyroidDF['TSH'].quantile(0.75)
IQR_TSH = Q3_TSH - Q1_TSH
under_TSH = thyroidDF['TSH'] < (Q1_TSH - 3 * IQR_TSH)
upper_TSH = thyroidDF['TSH'] > (Q3_TSH + 3 * IQR_TSH)
print('TSH:', 'Valores atipicos leves -', sum(under_TSH), ' | Valores atipicos severos -', sum(upper_TSH))
```

*Figura . Valores atípicos leves y severos de la hormona TSH.*

Se procede a crear boxplots que permiten visualizar los valores atípicos leves y severos de los valores numéricos, de los cuales se observa gran cantidad de valores atípicos, lo cual es normal debido a las condiciones médicas de los pacientes, los cuales pueden presentar alteraciones en los niveles de las hormonas.

Se identifica la cantidad de datos faltantes y se encuentra que *TBG* tiene muchos datos faltantes por lo cual no aporta mucha información al estudio, entonces se elimina del dataframe. Adicionalmente, las columnas con 3 o más datos faltantes son removidas del estudio. Posteriormente, se hace uso de XGBoost, un algoritmo de machine learning de tipo supervisado para manejar el desequilibrio severo entre las clases, pues *sample_weight* ayuda a contrarrestar este desequilibrio.
Se realiza un remapeo de los valores target del grupo diagnóstico y se crean las variables *train* y *test* para iniciar con el primer modelo.

*Modelo 1, ejecución base del modelo con hiperparámetros predeterminados - modelo supervisado.*

Se grafican el *mlogloss* y *merror* obtenidos mediante XGBoost.

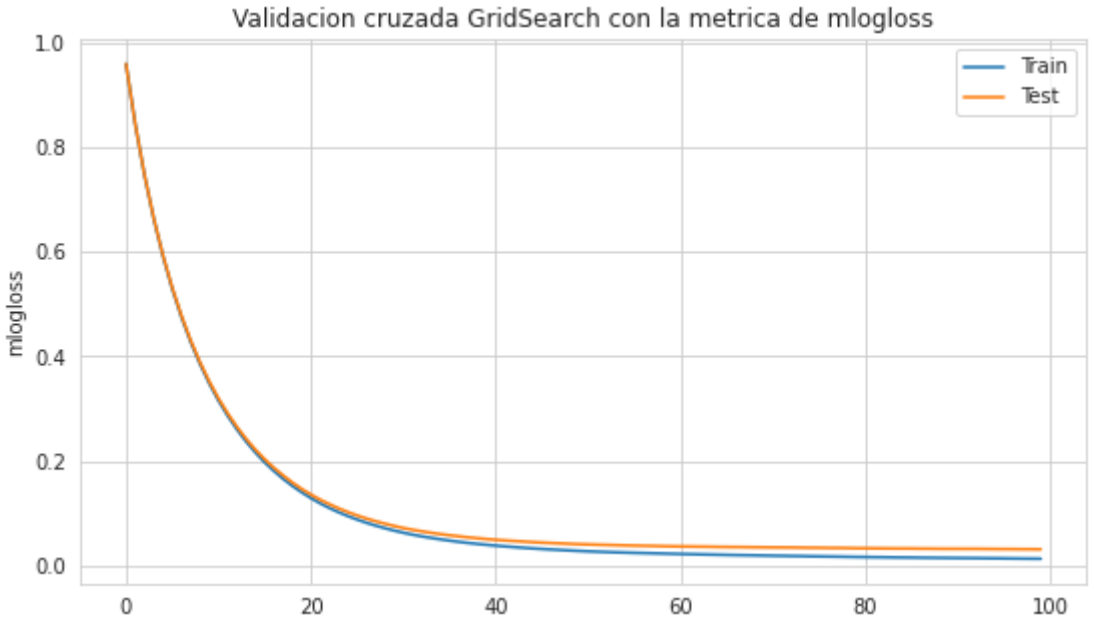


Figura 12 . Gráfica de mlogloss modelo 1 implementado mediante XGBoost.

Según la figura , los datos de train y test son datos similares, por lo cual el modelo está siendo capaz de predecir las patologías asociadas a la tiroides.
A continuación se muestra el reporte del modelo 1, en el cual se observan valores de las métricas *precision*, *recall* y *f1_score* para cada tipo de diagnóstico: negativo(0), hipotiroidismo(1) e hipertiroidismo(2).

```
---------Reporte del modelo -----------

              precision    recall  f1-score

           0       0.99      0.99      0.99
           1       0.97      1.00      0.98
           2       0.90      0.82      0.86
```

Figura 13. Reporte de las métricas del modelo 1 implementado mediante XGBoost.

Según el reporte anterior, para los diagnósticos 0, 1 y 2, los valores de *precision* son respectivamente 99%, 97% y 90%, lo cual indica que los pacientes del grupo 0 tendrá un 99% de probabilidad de no ser diagnosticados con hipo ni hipertiroidismo, los del grupo 1, tienen un 97% de ser diagnosticados con hiportiroidismo y el grupo 2, son pacientes con una probabilidad del 90% de ser diagnosticados con hipertiroidismo.
Para el caso del *recall*, los valores para los diagnósticos 0, 1 y 2 son respectivamente, 99%, 100% y 82%, los cuales indican los porcentajes de pacientes diagnosticados con ninguna enfermedad, con hipotiroidismo o hipertiroidismo en cada uno de los 3 grupos.

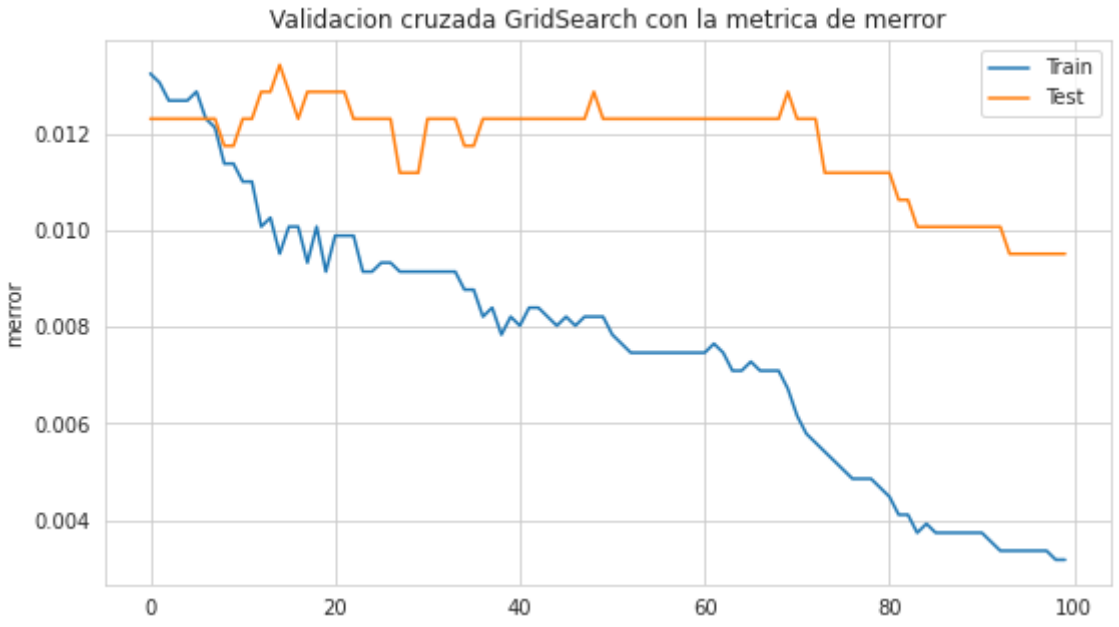


Figura 14. Gráfica de merror modelo 1 implementado mediante XGBoost.

*Modelo 2, ejecución con hiperparámetros optimizados - modelo supervisado*

Se grafican el *mlogloss* y *merror* obtenidos mediante XGBoost.

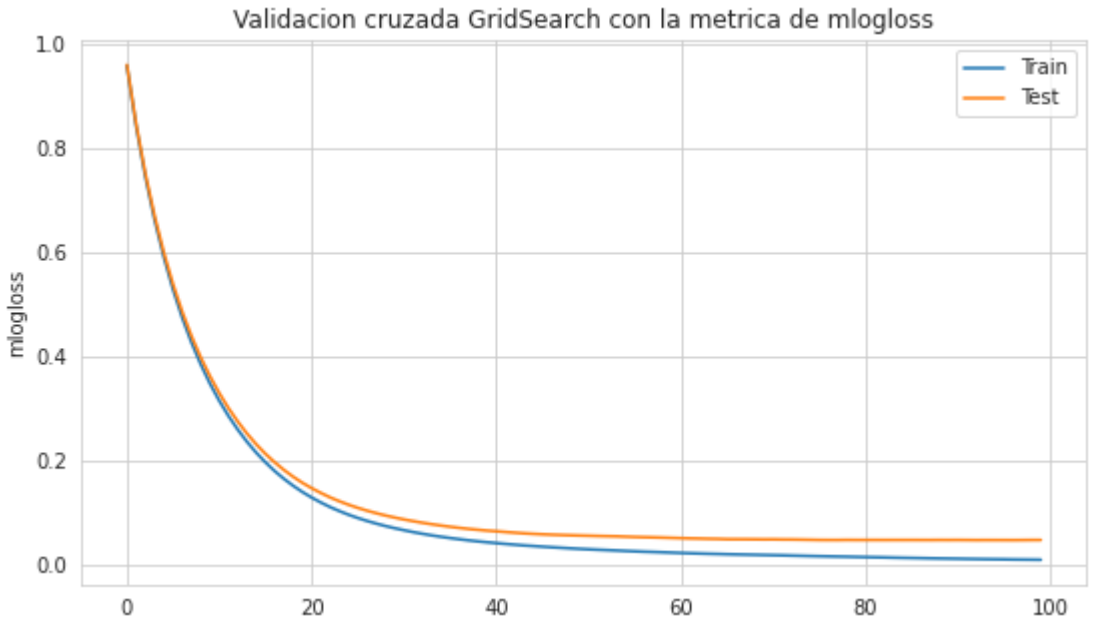


*Figura 15. Gráfica de mlogloss modelo 2* implementado mediante XGBoost.

Según la figura , los datos de train y test son datos similares, por lo cual el modelo está siendo capaz de predecir las patologías asociadas a la tiroides.
A continuación se muestra el reporte del modelo 1, en el cual se observan valores de las métricas *precision*, *recall* y *f1_score* para cada tipo de diagnóstico: negativo(0), hipotiroidismo(1) e hipertiroidismo(2).

| | precision | recall | f1-score |
|---|---|---|---|
| 0 | 1.00 | 0.99 | 0.99 |
| 1 | 0.95 | 1.00 | 0.98 |
| 2 | 0.77 | 0.93 | 0.85 |

Figura 16. Reporte de las métricas del modelo 2 implementado mediante XGBoost.

Según el reporte anterior, para los diagnósticos 0, 1 y 2, los valores de precision son respectivamente 100%, 95% y 77%, lo cual indica que los pacientes del grupo 0 tendrá un 100% de probabilidad de no ser diagnosticados con hipo ni hipertiroidismo, los del grupo 1, tienen un 95% de ser diagnosticados con hiportiroidismo y el grupo 2, son pacientes con una probabilidad del 77% de ser diagnosticados con hipertiroidismo.
Para el caso del recall, los valores para los diagnósticos 0, 1 y 2 son respectivamente, 99%, 100% y 93%, los cuales indican los porcentajes de pacientes diagnosticados con ninguna enfermedad, con hipotiroidismo o hipertiroidismo en cada uno de los 3 grupos.
Los porcentajes de *f1_score* son una combinación de las métricas *precision* y *recall*, dentro de la cual las dos métricas tienen igual importancia.

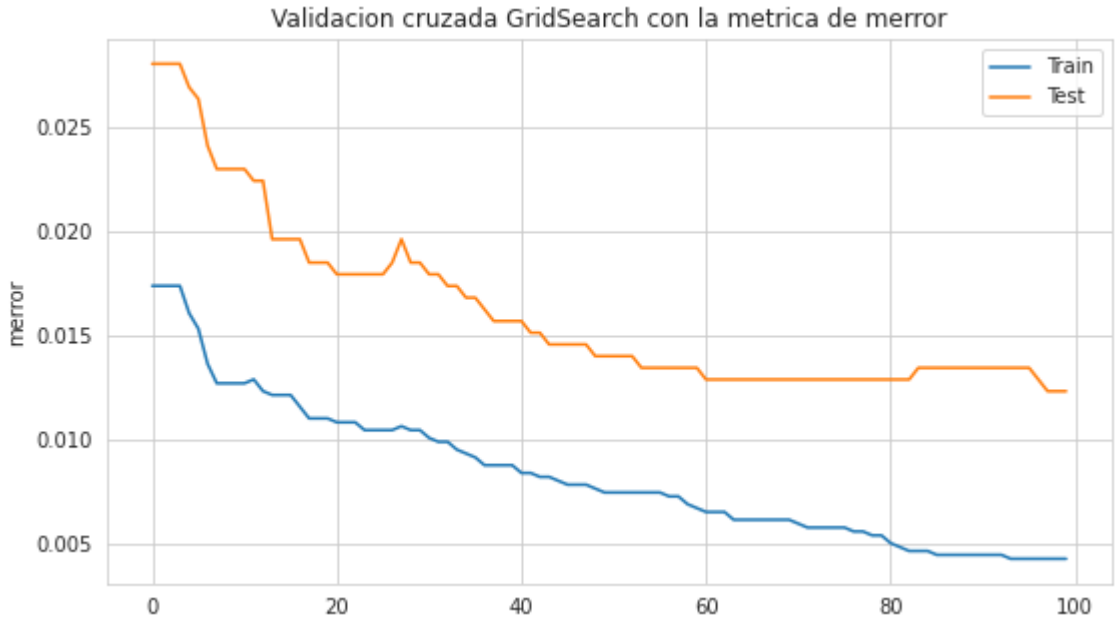


*Figura 17. Gráfica de merror modelo 2* implementado mediante XGBoost.

## *4. Conclusiones*

La identificación de la enfermedad tiroidea sigue siendo una tarea esencial pero difícil tanto en el diagnóstico clínico como en la clasificación estadística. El diagnóstico implica el uso de una gran cantidad de atributos de pacientes interrelacionados junto con grupos extremadamente desequilibrados que dan como resultado una relación complicada entre las características de entrada y las de salida.

La métrica accuracy (exactitud) no funciona bien cuando las clases están desbalanceadas como es en este caso. La mayoría de los pacientes presentan clases que están muy desbalanceadas entre sí debido a la naturaleza en la cual se encuentra el dataset, así que es muy fácil acertar diciendo que no lo van a estar. Para problemas con clases desbalanceadas es mucho mejor usar *precision*, *recall* y *f1_score*. Estas métricas dan una mejor idea de la calidad del modelo.

En la evaluación de los modelos de predicción utilizando deep learning para predecir hipotiroidismo e hipertiroidismo mediante la métricas *accuracy* y *loss*, se observaron valores de exactitud cercanos al 92% y valores de pérdida significativamente altos; por ello se implementaron tres métricas, precisión, recall y *f1_score,* la cual es una mezcla entre las dos métricas anteriores y se observó que los resultados para esta

métrica son demasiado bajos debido a que la precisión y recall son también muy bajos, por lo anterior se puede concluir que los datos están desbalanceados y por lo tanto *f1_score* no es la métrica adecuada para evaluar ninguno de los dos modelos para predecir hipotiroidismo e hipertiroidismo.

Los dos modelos que se trabajaron con el XGBoost, el modelo 1 que fue con los parámetros predeterminados y el modelo 2 que fue con los hiperparametros, arrojaron resultados similares en la validación cruzada para la métrica del *mlogloss*, mientras que para la validación cruzada con la métrica del *merror*, el modelo 2 con los hiperparametros es el modelo que mejor se ajusta y por tanto, se puede concluir que variar los hiperparametros y balancear los pesos de clase para nuestro target permite obtener un modelo preciso y capaz de predecir las patologías asociadas a la tiroides para nuestros grupos de diagnóstico.

## 5. Bibliografía


**[1]** Werr, E. F. (2022). Thyroid Disease Data. Kaggle.com.

**[2]** Jose Martinez Heras. (2019, November 17). *Precision, Recall, F1, Accuracy en clasificación - IArtificial.net*. IArtificial.net. https://www.iartificial.net/precision-recall-f1-accuracy-en-clasificacion/#:~:text=El%20valor%20F1%20se%20utiliza,la%20exhaustividad%20entre%20varias%20soluciones.&text=El%20valor%20F1%20asume%20que,la%20precisi%C3%B3n%20y%20la%20exhaustividad.

**[3]** Gaurav Dembla. (2020, November 17). Intuition behind Log-loss score - Towards Data
Science. Medium; Towards Data Science.
https://towardsdatascience.com/intuition-behind-log-loss-score-4e0c9979680a

**[4]** Las mujeres son ocho veces más propensas que los hombres de padecer enfermedades
en el tiroides | Médicos y Pacientes. (2022). Medicosypacientes.com.
http://www.medicosypacientes.com/articulo/las-mujeres-son-ocho-veces-mas-propensasque-los-hombres-de-padecer-enfermedades-en-el#:~:text=Los%20problemas%20de%20tir
oides%20afectan,a%20aumentar%20de%20modo%20gradual

**[5]** Félix, J., Corrales, B., & Soria, R. (2016). Factores de riesgo de las enfermedades
tiroideas. Hospital del Seguro Social Ambato. Revista de Ciencias Médicas de Pinar Del Río,
20(5), 113–128.
http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1561-3194201600050001

**[6]** merishnasuwal. (2020, November 10). *Diagnosing Hypothyroid disease using Deep Learning*. Kaggle.com; Kaggle. https://www.kaggle.com/code/merishnasuwal/diagnosing-hypothyroid-disease-using-deep-learning/notebook

**[7]** emmanuelfwerr. (2022, June 19). XGBoost Multi-class Classification. Kaggle.com; Kaggle. https://www.kaggle.com/code/emmanuelfwerr/xgboost-multi-class-classification

**[8]** Bosco, J. (2020, August 12). Tutorial: XGBoost en Python - Juan Bosco Mendoza Vega - Medium. Medium; Medium. https://medium.com/@jboscomendoza/tutorial-xgboost-en-python-53e48fc58f73