BREAST CANCER REPORT

Origen de los datos

Nuestro dataset de Breast Cancer Wisconsin está recogido de Kaggle, en el siguiente enlace podrás acceder a él.

Breas Cancer Winsconsin (Diagnostic)

Problemas de negocio

Descripción del problema

El cáncer de mama es una de las principales causas de mortalidad entre las mujeres a nivel mundial. La detección temprana del cáncer es crucial para que las probabilidades de supervivencia del paciente se vean incrementadas y también reducir los altos costos del tratamiento.

Aún detectado el tumor a tiempo, la intervención médica se prolonga hasta determinar si es benigno o maligno, debido principalmente a la interpretación de las imágenes de las mamografías de los radiólogos.

Impacto en el Negocio

Los principales desafíos a los que se enfrentan las instituciones médicas en la detección tardía del cáncer de mama son los siguientes:

- **Costos de tratamiento elevado**
- **Tasa de supervivencia reducida**
- **Errores de diagnóstico**

En nuestro proyecto nos centramos en eliminar los errores de diagnóstico de los médicos que, indirectamente, causan atrasos que conllevan a la detección tardía del cáncer de mama, aumentando los costos del tratamiento y haciendo más difícil la supervivencia de este.

Aplicación de vuestra solución

Para lograr nuestro objetivo hemos entrenado modelos de clasificación para ayudar a los médicos a minimizar el error y a una rápida detección.

Descripción de las variables

Nuestro dataset se compone de 30 columnas de datos reales a la hora de aplicarse en la detección de cáncer de mama, las cuales todas ellas contienten valores de tipo Float, y una columna target que representa si el túmor analizado es benigno o maligno, siendo esta la única columna con valores objeto.

Feature engineering

Distribución de variables

La distribución de variables de nuestro dataset muestra una tendencia a la asimetría negativa.

Correlaciones

Estudiando las correlaciones existentes entre nuestras variables, podemos observar que entre ellas guardan muchas correlaciones, y centrándonos en la columna de diagnóstico, la mitad de las variables tienen una correlación del 50% y un cuarto de las restantes entre 40% y 20%

Reducción de dimensionalidad

Con la valoración inicial que nos arroja el estudio de las correlaciones, podemos afirmar que la reducción de dimensionalidad no es viable en nuestro caso, puesto que al no tener demasiadas dimensiones con las que trabaja el modelo, realizarla es casi seguro que comprometerá la información, maximizará la perdida de información y aumentará el bias del modelo, lo que puede desembocar en un sobreajuste del modelo.

Tras esa valoración, decidimos no utilizar ninguna técnica de reducción de dimensionalidad.

Balanceo de datos

Como no tenemos un gran número de datos y solo alrededor del 30% de nuestro dataset representa las muestras de tumores malignos, las técnicas de undersapling no son posibles. Tras esto hicimos pruebas de oversapling y otras técnicas que implicaban generación de datos sintéticos, pero estas resultaron que perjudicaban al modelo.

Al final, como los datos que manejamos son sensibles y críticos, hemos decido no comprometer la información y evitar agregar ruido que pueda perjudicar el desempeño del modelo.

Transformación

El dataset sobre el que estamos trabajando trae ya datos limpios y dispuestos a trabajar con ellos directamente, salvo por la columna Diagnóstico, por lo que la única transformación que hemos hecho ha sido mapear esta columna sustituyendo los valores por 0 o 1, siendo 1 si el tumor es Maligno.

Modelos

RandomForest

A este modelo en un primero momento le añadimos un ***threshold*** de 0.4, este hiperparámetro modifica las probabilidades de pertenencia a una clase, en este caso el umbral nos ayudará a clasificar con más frecuencia los falsos negativos.

Luego hicimos una búsqueda de hiperparámetros para mejorar su rendimiento, y el resultado arrojado es el siguiente:

XGBoost

Parámetros

NUM_BOOST_ROUND: el modelo realiza **400** iteraciones ya que es a esa altura donde el aprendizaje del modelo se estabiliza y ya no tiene sentido seguir haciendo más.

MAX_DEPTH. Escogimos **3** como la profundidad máxima de los árboles ya que fue el valor que mejores resultados obtuvo y medidas más altas podrían llevar a overfitting.

ETA: al ser un dataset pequeño podemos trabajar con un learning rate más alto como 0.1.

SUBSAMPLE: **0.5.** Como nuestro dataset es reducido un subsample bajo hace que cada árbol vea solo una parte de los datos y permite mejorar la generalización del modelo.

COLSAMPLE_BYTREE: 0.8. Lo mismo que con el subsample pero en vez de con las filas con las columnas.

GAMMA Y MIN_CHILD_WEIGHT: despues de probar con variaciones de estos hiperparámetros el valor **1** fue el que nos dió mejores resultados en ambos.

LAMBDA Y ALPHA: Regularizaciones L2 y L1, le damos un valor de **1** a ambas para reducir un posible sobreajuste.

Red Neuronal

Arquitectura

La Red Neuronal que hemos diseñado cuenta con 2 capas ocultas, a las cuales se les aplica la función de activación ***Relu***, que hace que retorne 0 si el valor de entrada es negativo e introduce la *no linealidad*.

La segunda capa oculta también cuenta con un *dropout* con probabilidad de 20% de apagar la neurona, con esta operación reducimos las posibilidades de que el modelo sufra de *overfitting*.

La capa de salida tiene por función de activación ***Sigmoid***, está mapea cualquier valor real a un rango entre 0 y 1, lo que para un problema de clasificación binaria como este es el más adecuado.

Función de pérdida

La función de pérdida utilizada es Binary Cross-Entropy Loss, la cual es comunmente utilizada en problemas de clasificación binaria, cuando se requiere averiguar si un valor pertenece a una clase o no.

Optimizador

El optimizador que utilizamos es Root Mean Square Propagation, que calcula la tasa de aprendizaje de manera adaptativa para cada parámetro siguiendo las siguientes fórmulas:

Acumulador de los cuadrados gradientes

$$E[g^2]_t = \rho E[g^2]_{t-1} + (1-\rho) g_t^2$$

Actualización de los pesos

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t$$

Resultados

El tiempo de entrenamiento es similar entre los dos modelos, así que la diferencia fundamental radica en sus resultados:

Random Forest

Random Forest Classifier:
Precisión: 0.96

	precision	recall	f1-score	support
0	0.95	1.00	0.97	107
1	1.00	0.91	0.95	64
accuracy			0.96	171
macro avg	0.97	0.95	0.96	171
weighted avg	0.97	0.96	0.96	171

XGBoost

XGBoost Classifier:
Precisión: 0.99

	precision	recall	f1-score	support
0.0	0.98	1.00	0.99	107
1.0	1.00	0.97	0.98	64
accuracy			0.99	171
macro avg	0.99	0.98	0.99	171
weighted avg	0.99	0.99	0.99	171

Red Neuronal

Red Neuronal:
Precisión: 0.98

	precision	recall	f1-score	support
0.0	0.97	1.00	0.99	107
1.0	1.00	0.95	0.98	64
accuracy			0.98	171
macro avg	0.99	0.98	0.98	171
weighted avg	0.98	0.98	0.98	171

Por: Vanesa Blanco, Pedro Mera y Pablo Pérez