Notebook 2

1. Introducción

El dogma central de la biología molecular describe el proceso de dos pasos, transcripción y traducción, mediante el cual la información genética se convierte en proteínas: ADN → ARN → proteína. Las proteínas pueden modificarse posteriormente para regular la función celular. Los procesos de transcripción y traducción se regulan de diversas maneras. Comprender estas regulaciones y cómo se alteran en los tumores es crucial para avanzar en la investigación y el tratamiento del cáncer **(Alfaro et al., 2014)**.

La actividad desregulada de las proteínas, incluida la señalización por cinasas y la acetilación de la cromatina, se evalúa más directamente con mediciones de las proteínas y sus modificaciones postraduccionales. Por lo tanto, la proteómica tiene un importante valor complementario a la caracterización genómica y transcriptómica de los tumores.

La relación entre los transcritos de ARNm y las proteínas es fundamental para nuestra comprensión y aplicación de la biología molecular **(Crick, 1958)**. Para mejorar el rendimiento en el uso del número de copias de genes y los niveles de transcripción en la predicción de los niveles de proteínas y fosforilación, se lanzó una competencia colaborativa basada en la comunidad **(Saez-Rodriguez et al., 2016)**.

Se trata del desafío proteogenómico NCI-CPTAC DREAM en noviembre de 2017. En este desafío, los participantes aplicaron diferentes métodos computacionales a los datos proteogenómicos generados por el consorcio de análisis proteómico clínico de tumores (CPTAC) para predecir los niveles de proteínas y fosforilación basados en datos genómicos y transcriptómicos. Los participantes tuvieron acceso a mediciones cuantitativas del número de copias de genes, transcritos, proteínas y niveles de fosforilación para miles de genes en dos cohortes de cáncer. Además de los datos ómicos, los participantes fueron invitados a utilizar información previa de bases de datos existentes, como interacciones proteína-proteína y propiedades fisicoquímicas, para mejorar el rendimiento.

Este proyecto que tenemos delante ahora mismo, se centra en la clasificación multicategoría del cáncer, cuyo objetivo es la identificación precisa de diversos tipos específicos de cáncer, contribuyendo significativamente al avance y mejora de los métodos diagnósticos y terapéuticos en oncología.

El ADN libre en sangre, bajo condiciones normales, se encuentra en muy bajas concentraciones, asociado a los procesos de muerte celular que ocurren normalmente en el recambio celular. Las concentraciones de ADN libre en sangre pueden verse alteradas por distintas circunstancias, como daño a tejidos, inflamación, embarazo, infecciones, cáncer y trauma. En pacientes con cáncer, la concentración de ADN libre disminuye en respuesta a quimioterapia y se incrementa cuando hay diseminación del tumor o metástasis.

Existe gran divergencia entre valores de referencia de la concentración de ADN libre en voluntarios sanos. Para Wu y colaboradores **(Wu et al.)**, la concentración de ADN libre en individuos sanos es de 57,1 ± 30,6 ng/mL; otros estudios muestran concentraciones de 0 hasta 35,2 µg/mL; y para otros autores, los valores normales son de 10 a 30 ng/mL con una media de 13 ng/mL. Adicionalmente, la concentración de este ADN aumenta en pacientes con cáncer, excediendo los 100 ng/mL hasta aproximadamente 180 ng/mL. En pacientes con cáncer de pulmón se han reportado medias de 318 ng/mL.

Es necesario establecer los valores normales de la cuantificación de ADN libre para poder establecer comparaciones precisas con pacientes que padezcan alguna de las patologías mencionadas. La divergencia en estos valores reportados podría resultar en errores al realizar comparaciones, obteniendo valores de cuantificación equívocos que puedan resultar en informes deficientes a la hora de determinar enfermedades mediante esta técnica.

Este estudio tuvo como objetivo determinar la concentración de ADN libre en personas sanas en la población bogotana, para establecer un rango normal o de referencia mediante la técnica de PCR en tiempo-Real sin realizar pasos de extracción y purificación de ADN.

Comenzamos cargando los datos, seleccionando del archivo las tablas más significativas para la tarea: las tablas S4 y S6. Este paso es crucial para iniciar el análisis de los datos utilizados en nuestro proyecto de clasificación multicategoría del cáncer, cuyo objetivo es identificar diversos tipos específicos de cáncer y contribuir al avance de los métodos diagnósticos y terapéuticos en este campo.

2. Exploratory Data Analysis (EDA)

Procedemos limpiando estas tablas de datos nulos y convirtiéndolas en DataFrames, denominados `df4` y `df6`, respectivamente. Este proceso de limpieza implementa un pipeline de análisis exploratorio de datos (EDA) utilizando las librerías `pandas`, `seaborn` y `matplotlib`.

El EDA completo incluye:

- Carga y limpieza de datos
- Análisis descriptivo
- Visualización de distribuciones numéricas y categóricas
- Visualización de correlaciones
- Comparación de características por grupos

2.1. Tabla S4

	Patient ID #	Tumor type	AJCC Stage	Histopathology	Plasma volume (mL)	Plasma DNA concentration (ng/mL)	CancerSEEK Logistic Regression Score	CancerSEEK Test Result
0	CRC 455	Colorectum	I	Adenocarcinoma	5.0	6.08	0.938	Positive
1	CRC 456	Colorectum	I	Adenocarcinoma	4.0	46.01	0.925	Positive
...	...	...	...	...	...	...	...	...
1812	PAPA 1353	Ovary	I	Epithelial carcinoma	3.5	6.55	0.980	Positive
1813	PAPA 1354	Ovary	I	Epithelial carcinoma	3.5	22.83	0.999	Positive
1814	PAPA 1355	Ovary	III	Epithelial carcinoma	3.5	64.51	1.000	Positive
1815	PAPA 1356	Ovary	II	Epithelial carcinoma	3.5	13.71	1.000	Positive
1816	PAPA 1357	Ovary	III	Epithelial carcinoma	3.5	19.81	1.000	Positive

1817 rows × 8 columns

Fig. 1

Notas

- Esta tabla muestra un extracto de las primeras y últimas filas de la tabla completa de 1817 filas y 10 columnas.
- *Para acceder a la tabla completa, consulte el archivo original.*

2.1.1. Descripción Estadística de Tabla S4

	Age	Plasma volume (mL)	Plasma DNA concentration (ng/mL)	CancerSEEK Logistic Regression Score
count	1817	1817	1817	1817
mean	56.81	7.37	8.92	0.55
std	17.31	0.62	15.18	0.37
min	17.00	2.00	0.00	0.06
25%	47.41	7.50	2.31	0.20
50%	60.00	7.50	4.38	0.46
75%	69.43	7.50	8.20	0.99
max	93.00	7.50	157.48	1.00

Fig. 2

Notas

- Los valores presentados son resúmenes estadísticos de algunas columnas seleccionadas de la tabla completa para simplificar la visualización.
- **Plasma DNA concentration**: Concentración de ADN en plasma
- **CancerSEEK Logistic Regression Score**: Puntaje de regresión logística de CancerSEEK

Para acceder a la tabla completa, consulte el archivo original.

2.1.2.Observaciones del Análisis de la Tabla S4

- **Balance de Datos**: Hay un desbalance significativo en algunas categorías (por ejemplo, Raza, Tipo de Tumor y Resultado del Test CancerSEEK).
- **Distribuciones**: Algunas variables muestran distribuciones sesgadas (por ejemplo, la concentración de ADN en plasma), lo que puede requerir transformaciones para un análisis adecuado.
- **Correlaciones**: Identificamos correlaciones moderadas entre algunas variables clave, lo que puede guiar el desarrollo de modelos predictivos.
- **Análisis de Componentes Principales (PCA)**: Indica alta variabilidad y dispersión en los datos, sugiriendo que múltiples factores contribuyen a las diferencias observadas.

Estos datos permiten el análisis de marcadores biológicos en relación con el diagnóstico y el pronóstico del cáncer, ofreciendo una visión integral para estudios oncológicos detallados y personalizados.

2.2. Tabla S6

Index	Patient ID #	Tumor type	AJCC Stage	AFP (pg/ml)	CA-125 (U/ml)	CA 15-3 (U/ml)	CA19-9 (U/ml)	CEA (pg/ml)	CancerSEEK Test Result
0	CRC 455	Colorectum	I	1583.450	5.090	19.08	16.452	6832.07	Positive
1	CRC 456	Colorectum	I	715.308	7.270	10.04	40.910	5549.47	Positive
2	CRC 457	Colorectum	II	4365.530	4.854	16.96	16.452	3698.16	Negative
3	CRC 458	Colorectum	II	715.308	5.390	8.31	16.452	5856.00	Negative
4	CRC 459	Colorectum	II	801.300	4.854	11.73	16.452	5447.93	Negative
...	...	...	...	...	...	...	...	...	...
1812	PAPA 1353	Ovary	I	879.498	24.820	10.30	42.390	5390.31	Positive
1813	PAPA 1354	Ovary	I	1337.330	5.580	9.80	16.440	7951.03	Positive
1814	PAPA 1355	Ovary	III	879.498	30.480	8.48	16.440	2396.36	Positive
1815	PAPA 1356	Ovary	II	879.498	1469.450	23.74	62.260	3079.81	Positive
1816	PAPA 1357	Ovary	III	879.498	1428.310	836.85	37.900	3967.55	Positive

1817 rows × 10 columns

Fig. 3

Notas

- Esta tabla muestra un extracto de las primeras y últimas filas de la tabla completa de 1005 filas y 47 columnas.
- **AFP**: Alfa-fetoproteína
- **CA-125**: Antígeno del cáncer 125
- **CEA**: Antígeno carcinoembrionario

Para acceder a la tabla completa, consulte el archivo original.

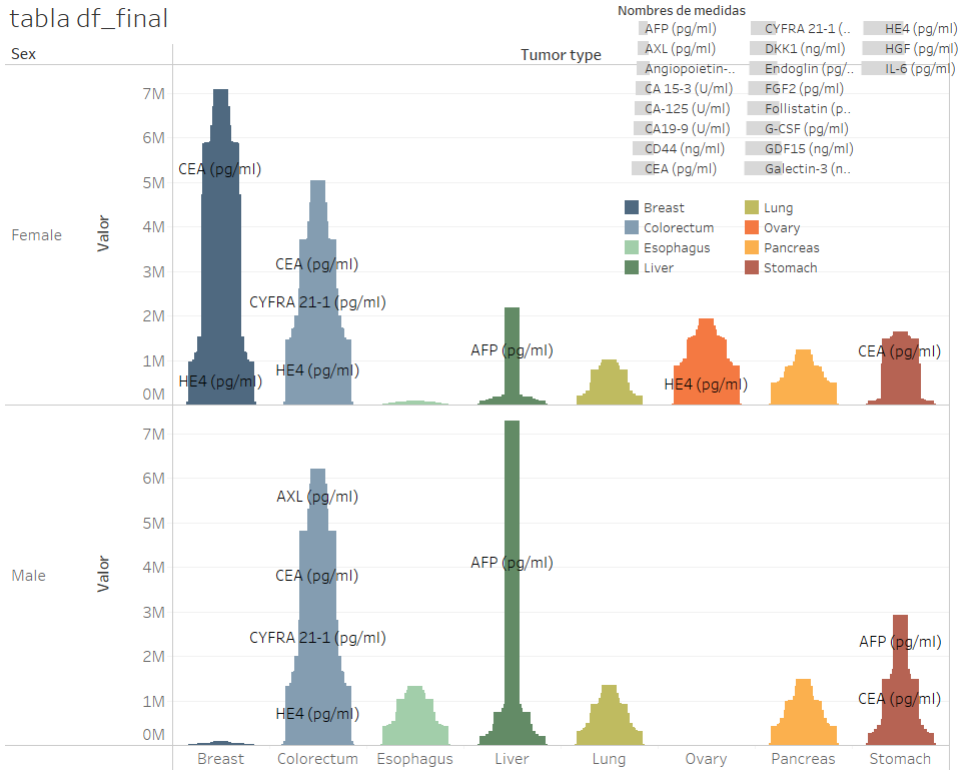

Fig. 4 **TABLA S6, valores niveles proteínas según tipo tumor y sexo.**

2.2.1. Descripción Estadística de Tabla S6

	AFP (pg/ml)	Angiopoietin-2 (pg/ml)	CA-125 (U/ml)	CEA (pg/ml)	CancerSEEK Logistic Regression Score
count	1817	1817	1817	1817	1817
mean	589.75	241.92	33.26	3703.97	0.63
std	1396.07	578.08	90.78	7383.15	0.39
min	0.00	0.00	0.00	0.00	0.06
25%	130.48	31.59	5.60	571.85	0.21
50%	715.31	94.29	11.00	2242.57	0.48
75%	879.50	221.80	19.92	4756.55	0.99
max	16236.47	13608.49	1469.45	65236.36	1.00

Fig. 5

Notas

- Los valores presentados son resúmenes estadísticos de algunas columnas seleccionadas de la tabla completa para simplificar la visualización.

Para acceder a la tabla completa, consulte el archivo original.

2.2.2. Observaciones del Análisis de la Tabla S6

- **Variabilidad entre Tipos de Tumores**: Muchos biomarcadores muestran variaciones significativas entre diferentes tipos de tumores, lo que puede ser útil para la clasificación y predicción del tipo de cáncer.
- **Valores Atípicos**: Existen varios valores atípicos en los datos, indicando que algunos pacientes tienen niveles extremadamente altos o bajos de ciertos biomarcadores.
- **Valores Faltantes**: La presencia de valores faltantes en algunos biomarcadores debe ser considerada y abordada mediante técnicas de imputación o exclusión de datos.
- **Distribución de Edad**: La mayoría de los pacientes tienen edades que oscilan entre 40 y 70 años.
- **Distribución de Sexo**: Hay más mujeres que hombres en el conjunto de datos.
- **Distribución del Tipo de Tumor**: El tipo de tumor más común en el conjunto de datos es el colorrectal, seguido por otros tipos como el de pulmón y el de mama.
- **Distribución del Volumen de Plasma**: La mayoría de los volúmenes de plasma están entre 4 y 7 mL.
- **Distribución de la Concentración de ADN en Plasma**: La concentración de ADN en plasma varía ampliamente, pero la mayoría de las muestras tienen concentraciones bajas.
- **Distribuciones Sesgadas**: Se observan distribuciones sesgadas en varios biomarcadores, particularmente en AFP y CEA, lo que sugiere la necesidad de transformaciones de datos.
- **Correlaciones**: Algunas correlaciones son fuertes entre ciertos marcadores, lo que puede indicar relaciones biológicas subyacentes.
- **Implicaciones Clínicas**: Las concentraciones elevadas de marcadores como CA-125 y CEA tienen importantes implicaciones clínicas y pueden ser útiles en la estratificación del riesgo y en el monitoreo de la enfermedad.

El análisis exhaustivo de estas tablas proporciona una base robusta para el desarrollo de modelos predictivos y la identificación de biomarcadores críticos en el cáncer.

Antes de preprocesar la tabla S6, eliminamos el valor "Normal" (se refiere a ausencia de tumor) de la columna Tipo de Tumor y obtenemos las siguientes métricas:

- **Omega score (0.70% de Valores Faltantes)**:
  - Recomendación: Dado el pequeño porcentaje de valores faltantes, la imputación con la mediana es adecuada para mantener la robustez ante posibles valores atípicos.
- **G-CSF (pg/ml) (0.10% de Valores Faltantes)**:
  - Recomendación: La imputación con la mediana también es adecuada aquí debido al muy bajo porcentaje de valores faltantes.

2.3. Exploración de los Modelos Predictivos

El objetivo principal de este análisis es identificar los factores clave que influyen en los resultados del test CancerSEEK. Para lograr esto, evaluamos la precisión predictiva de varios modelos de machine learning, incluidos Regresión Logística, Random Forest y XGBoost. El modelo con mejor desempeño se selecciona para el análisis final y la validación en un conjunto de datos separado. Este enfoque garantiza una comprensión profunda de los mecanismos biológicos subyacentes y una precisión diagnóstica mejorada.

A continuación se presenta el flujo de trabajo del análisis, desde la carga de los datos hasta la evaluación de los modelos predictivos:

- **Carga de Datos**: Importación y preprocesamiento de los datos.
- **Exploración Inicial**: Análisis exploratorio para comprender las características clave.
- **Limpieza y Transformación**: Eliminación de datos nulos y normalización de variables.
- **Modelado Predictivo**: Entrenamiento y evaluación de modelos de machine learning.
- **Validación**: Evaluación del modelo en un conjunto de datos de prueba.
- **Interpretación**: Análisis de los resultados y determinación de los factores predictivos más importantes.

Este flujo de trabajo garantiza un análisis riguroso y sistemático de los datos, proporcionando información valiosa para el diagnóstico y tratamiento del cáncer.

2.3.1. Preprocesamiento de datos

Pasos seguidos en la creación del DataFrame a usar.

Aquí estamos tratando la aplicación de las funciones creadas para preprocesar los datos del DataFrame df6 sin los datos correspondientes al tipo de tumor "Normal" (sin indicios de la existencia de algún tipo de tumor), dando lugar a la creación del DataFrame que se usará a continuación bajo el nombre *df_final*, detallando dicho proceso a continuación:

	AFP (pg/ml)	Angiopoietin-2 (pg/ml)	AXL (pg/ml)	CA-125 (U/ml)	CA 15-3 (U/ml)	CA19-9 (U/ml)	CD44 (ng/ml)	CEA (pg/ml)	CYFRA 21-1 (pg/ml)
0	-0.162730	-0.399967	0.161177	-0.155628	-0.208773	-0.126586	0.100087	-0.177265	-0.118556
1	-0.161972	-0.412345	-0.862979	-0.155856	-0.139035	-0.126858	-0.260427	-0.162648	-0.117902
2	-0.155496	-0.507819	-0.284533	-0.155194	-0.255232	0.053126	-0.630079	-0.184200	-0.115457
3	-0.150629	-0.584907	-0.928337	-0.155787	-0.230770	-0.106049	0.047755	-0.181969	-0.011969
4	-0.147157	-0.464835	1.144186	-0.131803	0.317763	-0.085735	2.054673	-0.192859	-0.119066

Fig. 6

Notas

- Los valores presentados son resúmenes estadísticos de algunas columnas seleccionadas de la tabla completa para simplificar la visualización.
- Las columnas DKK1 (ng/ml), sHER2/sEGFR2/sErbB2 (pg/ml), sPECAM-1 (pg/ml), TGFa (pg/ml), Thrombospondin-2 (pg/ml), TIMP-1 (pg/ml), TIMP-2 (pg/ml), Omega score, AJCC Stage, Sex_Male y Tumor type no están presentes en esta versión de la tabla, para una mejor visualización en el documento escrito.

`*Para acceder a la tabla completa, consulte el archivo original.*`

1. **Eliminar Columnas No Deseadas**: Se eliminaron las siguientes columnas del DataFrame original ya que no son necesarias para el análisis:

- **ID del Paciente**: No aporta información relevante para la clasificación.
- **ID de la Muestra**: Similar al ID del Paciente, es irrelevante para el análisis.
- **Resultado del Test CancerSEEK**: Esta columna es redundante porque estamos interesados en la clasificación a partir de los biomarcadores.
- **Puntuación de Regresión Logística de CancerSEEK**: Se elimina porque queremos centrarnos en los biomarcadores específicos y no en una puntuación compuesta.

2. **Identificación de Características**:
   - **Características Numéricas**: Después de eliminar las columnas no deseadas, identificamos las columnas que contienen datos numéricos.
   - **Características Categóricas**: De manera similar, identificamos las columnas que contienen datos categóricos.
3. **Preprocesamiento de las Características Numéricas**:
   - **Imputación con la Mediana**: Para manejar valores nulos en características numéricas.
   - **Estandarización**: Para escalar las características numéricas a una distribución estándar.
4. **Preprocesamiento de las Características Categóricas**:
   - **Codificación Ordinal para *AJCC Stage***: Dado que tiene un orden intrínseco.
   - **Codificación Binaria para *Sex***: Dado que es una variable con dos categorías (Hombre y Mujer).
5. **Combinación de los Transformadores**: Se combinó todo el preprocesamiento en un solo ColumnTransformer para aplicar las transformaciones adecuadas a cada tipo de característica.
6. **Separación de Características y Variable Objetivo**:
   - **Características (X)**: Se eliminaron las columnas seleccionadas para quedarse con las características.
   - **Variable Objetivo (y)**: Tipo de Tumor - *Tumor type*.
7. **División en Conjuntos de Entrenamiento y Prueba**:
   - **Entrenamiento (70%) y Prueba (30%)**: Los datos se dividieron aleatoriamente en conjuntos de entrenamiento y prueba usando una semilla aleatoria (*random_state*=42) para reproducibilidad.

8. **Ajuste y Transformación de los Datos**:
    - Se ajustaron los datos de entrenamiento con el preprocesador configurado.
    - Se transformaron los datos de entrenamiento y prueba aplicando las mismas transformaciones.

Uno de los resultados clave del preprocesamiento de datos, como se muestra en la Fig. 6, es la transformación de la variable objetivo, *Tumor type*, a un formato numérico. Este paso es crucial por varias razones:

- **Compatibilidad**: Convertir la variable objetivo a formato numérico asegura su compatibilidad con una amplia gama de algoritmos de machine learning. La mayoría de estos algoritmos están diseñados para trabajar con datos numéricos, por lo que esta conversión es esencial para el correcto funcionamiento del modelo.
- **Eficiencia**: Los algoritmos de aprendizaje automático suelen funcionar más eficientemente con variables numéricas. Las operaciones matemáticas subyacentes en estos algoritmos son más rápidas y precisas cuando se utilizan números en lugar de cadenas de texto.
- **Precisión**: La conversión a formato numérico evita posibles errores de procesamiento que pueden ocurrir con variables categóricas en formato string. Al trabajar con números, se minimiza el riesgo de errores de codificación y se mejora la consistencia de los resultados.

Este enfoque de preprocesamiento no solo mejora la compatibilidad y eficiencia del modelo, sino que también asegura la precisión en la predicción de los distintos tipos de tumores.

Seguidamente, desarrollamos un código de transformación y combinación de datos con el objetivo de crear un nuevo DataFrame, que utilizaremos posteriormente para calcular la probabilidad total de predicción de los distintos tipos de tumores.

Este código realiza una serie de operaciones de preprocesamiento sobre un conjunto de datos de entrenamiento y prueba, utilizando librerías de Python como `numpy`, `pandas` y `sklearn`. El proceso incluye:

1. **Transformación y Combinación de Datos**: Utilizamos la función `combine_transformed_data_full` para combinar las características transformadas y la variable objetivo en un único DataFrame. Este enfoque nos permite trabajar con un conjunto de datos unificado que facilita las etapas posteriores de análisis y modelado.
2. **Preprocesamiento de Características**:
    - **Estandarización de Características Numéricas**: Las características numéricas se estandarizan para ajustar sus valores a una distribución con media cero y desviación estándar uno, lo cual es crucial para mejorar el rendimiento de muchos algoritmos de machine learning.
    - **Codificación de Características Categóricas**: Las características categóricas se transforman mediante técnicas como la codificación ordinal y la codificación one-hot, asegurando que estas variables puedan ser interpretadas adecuadamente por los modelos de machine learning.
3. **Transformación de la Variable Objetivo**: La variable objetivo, que representa el tipo de tumor, se convierte a valores numéricos utilizando `LabelEncoder`. Esto es fundamental por varias razones:
    - **Compatibilidad**: Asegura que la variable objetivo sea compatible con una amplia gama de algoritmos de machine learning.
    - **Eficiencia**: Los algoritmos de aprendizaje suelen funcionar más eficientemente con variables numéricas.
    - **Precisión**: Evita posibles errores de procesamiento que pueden ocurrir con variables categóricas en formato string.

Finalmente, el DataFrame combinado se guarda en un archivo Excel y se muestra una vista previa del mismo. Este proceso garantiza que los datos estén en el formato adecuado para el análisis posterior y la construcción de modelos predictivos, facilitando la predicción precisa de los distintos tipos de tumores.

	AFP (pg/ml)	Angiopoietin-2 (pg/ml)	AXL (pg/ml)	CA-125 (U/ml)	CA 15-3 (U/ml)	CA19-9 (U/ml)	CD44 (ng/ml)	CEA (pg/ml)	CYFRA 21-1 (pg/ml)	DKK1 (ng/ml)	...	sHER2/sEGFR2/sErbB2 (pg/ml)	sPECAM-1 (pg/ml)	TGFa (pg/ml)	Thrombospondin-2 (pg/ml)	TIMP-1 (pg/ml)	TIMP-2 (pg/ml)	Omega score	AJCC Stage	Sex_Male	Tumor type
0	-0.162730	-0.399967	0.161177	-0.155628	-0.208773	-0.126586	0.100087	-0.177265	-0.118556	-0.748740	...	0.613042	-0.739088	-0.190133	-0.501581	-0.709657	0.506555	-0.168524	0	1	1
1	-0.161972	-0.412345	-0.862979	-0.155856	-0.139035	-0.126858	-0.260427	-0.162648	-0.117902	-0.268988	...	-0.502477	-0.480916	-0.191864	-0.504778	-0.669592	-0.306917	-0.239250	1	1	4
2	-0.155496	-0.507819	-0.284533	-0.155194	-0.255232	0.053126	-0.630079	-0.184200	-0.115457	0.867268	...	-0.710243	-0.461172	-0.182823	-0.367130	-0.404068	-1.120939	-0.243808	1	1	1
3	-0.150629	-0.584907	-0.928337	-0.155787	-0.230770	-0.106049	0.047755	-0.181969	-0.011969	0.210765	...	-0.584788	-0.635555	-0.198597	-0.488231	-0.154741	1.505864	-0.042227	1	0	1
4	-0.147157	-0.464835	1.144186	-0.131803	0.317763	-0.085735	2.054673	-0.192859	-0.119066	-1.203243	...	0.827342	1.377542	-0.207830	-0.195666	-0.340628	1.733127	-0.215178	0	1	6

Fig. 7

A continuación, se presenta un análisis de la cantidad de datos disponibles para cada tipo de tumor. La siguiente tabla resume la distribución de los diferentes tipos de tumores en el conjunto de datos:

Tipo de Tumor	Cantidad
Colorectum	388
Breast	209
Lung	104
Pancreas	93
Stomach	68
Ovary	54
Esophagus	45
Liver	44

Fig. 8

La distribución de los datos, representada en la Fig. 8, revela un claro desbalance entre las distintas categorías de tumores. Observamos que ciertas categorías, como *Colorectum* y *Breast*, tienen una cantidad significativamente mayor de ejemplos en comparación con otras, como *Esophagus* y *Liver*. Este desbalance en los datos puede impactar negativamente el rendimiento de los modelos de machine learning, ya que tienden a sesgarse hacia las clases más representadas. Es crucial abordar este desbalance durante el preprocesamiento y la construcción de los modelos para asegurar una evaluación justa y precisa de todas las clases.

2.3.2. Evaluación de Modelos con Estrategias de Balanceo

En esta sección del proyecto, se procede a la evaluación y comparación del rendimiento de varios algoritmos de machine learning para la clasificación de tipos de tumores, utilizando técnicas de balanceo de datos para abordar conjuntos desbalanceados. El objetivo es seleccionar el modelo más adecuado que logre la mejor precisión y generalización posible.

Para implementar esta evaluación, se emplean bibliotecas como *scikit-learn* e *imblearn*, que proporcionan herramientas para la clasificación y técnicas de balanceo de *undersampling*. Se han definido funciones específicas para calcular métricas clave como precision, recall, F1-score y matriz de confusión, así como una puntuación global ponderada (*Global Score*) que facilita la comparación entre modelos. Cerramos esta fase del proyecto, guardando los datos obtenidos en un archivo formato excel, *model_results.xlsx* .

Los modelos de clasificación seleccionados para esta evaluación incluyen:

- **Logistic Regression**: Utilizado para problemas de clasificación binaria.
- **Decision Tree Classifier**: Ideal para clasificación multiclase con control de la profundidad del árbol.
- **Random Forest Classifier**: Implementa un bosque aleatorio para mejorar la precisión en clasificación multiclase.
- **K-Nearest Neighbors (KNN)**: Basado en vecinos más cercanos con métrica de distancia Manhattan.
- **AdaBoost Classifier**: Utiliza boosting para mejorar la precisión en clasificación multiclase.
- **Gradient Boosting Classifier**: Otra técnica de boosting para optimizar la clasificación multiclase.

Además de la selección de modelos, se incorporan estrategias de balanceo como parte del proceso de evaluación. Estas estrategias incluyen:

1. **RandomUnderSampler**

- **Propósito**: Esta técnica reduce el número de ejemplos en la clase mayoritaria seleccionando aleatoriamente muestras sin reemplazo para equilibrar el número de ejemplos en la clase minoritaria.
- **Ventaja**: Es sencillo y rápido de implementar, especialmente efectivo con conjuntos de datos grandes. Desventaja: Puede eliminar ejemplos importantes de la clase mayoritaria, lo que podría resultar en la pérdida de información relevante.
- **Tratamiento**: Undersampling.

2. **NearMiss**

- **Propósito**: NearMiss es una técnica de submuestreo que selecciona ejemplos de la clase mayoritaria basados en su proximidad a los ejemplos de la clase minoritaria.
- **Ventaja**: Conserva ejemplos representativos de la clase mayoritaria cerca de la frontera de decisión del clasificador.
- **Desventaja**: Similar al RandomUnderSampler, puede eliminar ejemplos significativos y puede ser costoso computacionalmente en conjuntos de datos grandes.
- **Tratamiento**: Undersampling.

3. **CondensedNearestNeighbour (CNN)**

- **Propósito**: CNN elimina ejemplos redundantes de la clase mayoritaria mientras conserva un subconjunto que representa bien la frontera de decisión del conjunto de datos.
- **Ventaja**: Reduce el tamaño del conjunto de datos sin perder ejemplos críticos para la clasificación.
- **Desventaja**: Puede ser computacionalmente costoso, especialmente con conjuntos de datos de alta dimensionalidad, y su eficacia puede variar.
- **Tratamiento**: Undersampling.

Estas estrategias de submuestreo (undersampling) se centran en reducir el número de muestras de la clase mayoritaria para equilibrar el conjunto de datos, mejorando así el rendimiento de los modelos al mitigar el sesgo hacia las clases dominantes.

```

Modelo: Logistic Regression
Cross-validated accuracy: 0.6182
Model performance for Training - Logistic Regression
- Accuracy: 0.7212
- F1 score: 0.7054
- Precision: 0.7429
- Recall: 0.7212
Confusion Matrix:
 [[118  22   0   0   2   1   2   1]
 [ 20 241   0   0   5   0   3   2]
 [  1  19   7   0   3   0   0   1]
 [  2   9   1  17   1   0   0   1]
 [  7  31   0   0  35   0   0   0]
 [  1  12   0   0   0  25   0   0]
 [  3  12   0   0   0   0  50   0]
 [  2  23   1   0   8   0   0  14]]
Model performance for Test - Logistic Regression
- Accuracy: 0.6722
- F1 score: 0.6513
- Precision: 0.6830
- Recall: 0.6722
Confusion Matrix:
 [[47 12  0  0  3  0  1  0]
 [12 99  0  0  2  1  0  3]
 [ 1  7  4  0  0  0  1  1]
 [ 2  6  0  5  0  0  0  0]
 [ 3 13  0  0 15  0  0  0]
 [ 0  5  0  1  0 10  0  0]
 [ 0  6  0  0  1  0 21  0]
 [ 2 13  0  1  1  0  1  2]]

Modelo: Decision Tree
```

Fig. 9

2.3.3. Interpretación de modelos

Cargamos el archivo *dataframe model_results* que generamos anteriormente y detectamos los 3 mejores modelos en la fase de prueba. Para ello ordenamos usando la puntuación obtenida en el parámetro *Global Score*:

	Model	Set	Accuracy	Precision	Recall	F1-Score	Global Score
1	Logistic Regression	Test	0.672185	0.683021	0.672185	0.651319	66.97

	Model	Set	Accuracy	Precision	Recall	F1-Score	Global Score
11	Gradient Boosting	Test	0.639073	0.619969	0.639073	0.615271	62.83
7	KNN	Test	0.609272	0.632833	0.609272	0.585354	60.92

Fig. 10

2.3.4. Exploración de Variables Relevantes en la Clasificación de Tipos de Tumores

Para identificar las variables más influyentes en la clasificación de tipos de tumores, se empleó un modelo *Random Forest* sobre un conjunto de datos preprocesado. Inicialmente, se aseguró la integridad de los datos eliminando cualquier instancia con valores faltantes del conjunto combinado y transformado.

Luego, se procedió a separar las características (variables predictoras) de la etiqueta de destino, que en este caso es el tipo de tumor. El conjunto de datos se dividió en datos de entrenamiento y prueba, utilizando una proporción del 80% para entrenamiento y 20% para prueba.

El modelo *Random Forest* se configuró con 250 árboles y se ajustó a los datos de entrenamiento. Posteriormente, se calculó la importancia de cada característica utilizando el atributo `feature_importances_` del modelo. Estas importancias se ordenaron de manera descendente y se visualizaron mediante un gráfico de barras utilizando la biblioteca *seaborn*, facilitando la comprensión de la contribución relativa de cada variable en la clasificación.

Finalmente, los resultados se almacenaron en un archivo CSV llamado *feature_importances.csv*, proporcionando una referencia para análisis posteriores y facilitando la comunicación de los hallazgos. Este enfoque no solo permite identificar las variables más relevantes en la clasificación de tumores, sino que también mejora la interpretación y visualización de los resultados mediante gráficos claros y detallados.

A continuación se muestra la tabla que presenta las 20 características más importantes según el modelo Random Forest:

	Feature	Importance
31	sFas (pg/ml)	0.044544
33	sHER2/sEGFR2/sErbB2 (pg/ml)	0.038370
4	CA 15-3 (U/ml)	0.034747
5	CA19-9 (U/ml)	0.033765
3	CA-125 (U/ml)	0.033653
38	TIMP-2 (pg/ml)	0.032779
35	TGFa (pg/ml)	0.032361
41	Sex_Male	0.032141
21	Leptin (pg/ml)	0.031827
19	IL-8 (pg/ml)	0.031686
18	IL-6 (pg/ml)	0.028046
0	AFP (pg/ml)	0.027086
15	GDF15 (ng/ml)	0.026415
29	Prolactin (pg/ml)	0.026332
17	HGF (pg/ml)	0.025399
6	CD44 (ng/ml)	0.025003
23	Midkine (pg/ml)	0.024707
36	Thrombospondin-2 (pg/ml)	0.024402
37	TIMP-1 (pg/ml)	0.023657
16	HE4 (pg/ml)	0.023151

Fig. 11

Este enfoque metodológico proporciona una manera efectiva de identificar y visualizar qué variables son más relevantes para la clasificación de tumores, promoviendo una mejor comprensión y análisis de los datos biomédicos.

2.3.5. Entrenamiento y Evaluación de Modelos XGBoost y LightGBM para la Detección de Tipos de Cáncer con Objetivo Desbalanceado

XGBoost (Extreme Gradient Boosting) y *LightGBM* (Light Gradient Boosting Machine) son poderosos algoritmos de boosting ampliamente utilizados en la detección de tipos de cáncer, incluso en conjuntos de datos con desequilibrio de clases. Aquí se detallan las características que hacen que estos modelos sean efectivos:

XGBoost

1. **Boosting por Gradiente**:

XGBoost utiliza un proceso secuencial de construcción de árboles, donde cada árbol corrige los errores del anterior, optimizando así la capacidad predictiva.

2. **Manejo de Datos Desbalanceados**:

Incorpora parámetros como *scale_pos_weight* para ajustar el peso de las clases y abordar el desbalance de datos, lo cual es crucial en la detección de cáncer donde las clases pueden estar desproporcionadamente representadas.

3. **Regularización**:

Implementa regularización L1 y L2 para evitar el sobreajuste, mejorando la generalización del modelo en nuevos datos.

4. **Eficiencia Computacional**:

Optimizado para operaciones eficientes, aprovechando características de hardware y software para manejar grandes volúmenes de datos con rapidez.

5. **Importancia de Características**:

Proporciona medidas de importancia de características que ayudan a identificar biomarcadores relevantes en la clasificación de tipos de cáncer.

6. **Rendimiento y Análisis del Rendimiento del Modelo XGBoost**:

El rendimiento del modelo XGBoost para la detección de tipos de cáncer en un conjunto de datos desbalanceado muestra resultados prometedores. A continuación, se presenta un análisis detallado de las métricas obtenidas:

- **Accuracy**: 0.6468

La métrica accuracy del modelo XGBoost es de 0.6468, lo que indica que aproximadamente el 64.68% de las predicciones fueron correctas. Esta métrica proporciona una visión general de la capacidad del modelo para clasificar correctamente los casos. Aunque es una métrica útil, puede no reflejar completamente el rendimiento del modelo en un contexto de datos desbalanceados.

- **F1 score**: 0.6174

El F1 score es de 0.6174, lo que refleja un equilibrio entre la precision y el recall. Este valor sugiere que el modelo tiene un rendimiento razonable, considerando tanto su capacidad para identificar correctamente los casos positivos (recall) como la proporción de predicciones positivas correctas (precision). En escenarios de desbalance de clases, el F1 score es una métrica crucial, y un valor de 0.6174 indica un desempeño satisfactorio del modelo XGBoost.

- **Precision**: 0.6233

La métrica precision del modelo es de 0.6233, lo que significa que el 62.33% de las predicciones positivas del modelo fueron correctas. Una alta precisión es importante en la detección de cáncer para reducir los falsos positivos, evitando diagnósticos erróneos y tratamientos innecesarios. Un valor de 0.6233 indica que el modelo tiene una precisión moderada-alta, lo que es positivo, aunque aún hay espacio para mejorar.

- **Recall**: 0.6468

El recall, también conocido como sensibilidad, es de 0.6468. Este valor indica que el modelo identificó correctamente el 64.68% de los casos positivos reales. En la detección de cáncer, un alto valor de recall es esencial para minimizar los falsos negativos, asegurando que la mayoría de los casos de cáncer se detecten. Un recall de 0.6468 sugiere que el modelo XGBoost tiene una buena capacidad para detectar casos positivos, aunque también indica que aún hay un 35.32% de casos positivos que no fueron detectados.

7. **Conclusión** El análisis del rendimiento del modelo XGBoost revela un desempeño bastante sólido en la detección de tipos de cáncer en un conjunto de datos desbalanceado. Con una precisión del 64.68%, un F1 score de 0.6174, una precisión de 0.6233 y un recall de 0.6468, el modelo muestra que puede diferenciar entre las clases con un nivel de exactitud razonable. Estos resultados indican que XGBoost es eficaz para esta tarea, aunque se pueden realizar mejoras adicionales para aumentar su rendimiento. Técnicas como el ajuste de hiperparámetros, el uso de métodos de preprocesamiento avanzados y la combinación con otros enfoques de machine learning podrían ayudar a mejorar aún más estos resultados. Este análisis sugiere que XGBoost es una herramienta valiosa para la detección de tipos de cáncer, proporcionando una base sólida para trabajos futuros.

LightGBM

1. **Boosting por Hojas**:

LightGBM adopta un enfoque basado en hojas en lugar de niveles, lo que le permite manejar mejor los datos desbalanceados al centrarse en las hojas con mayores errores.

2. **Tratamiento del Desbalanceo**:

Al igual que XGBoost, ofrece opciones como *is_unbalance* y *scale_pos_weight* para ajustar automáticamente el modelo y tratar eficazmente el desbalance de clases.

3. **Velocidad y Eficiencia**:

Conocido por su rapidez y eficiencia en conjuntos de datos grandes, utiliza métodos como el aprendizaje basado en histogramas para acelerar el entrenamiento.

4. **Escalabilidad**:

Capaz de manejar conjuntos de datos de alta dimensionalidad y grandes volúmenes gracias a su diseño optimizado para memoria y capacidad de paralelización.

5. **Reducción del Overfitting**:

Incorpora estrategias avanzadas de regularización y ajuste de hiperparámetros para mitigar el sobreajuste, mejorando así la precisión en datos de prueba.

6. **Rendimiento y Análisis del Rendimiento del Modelo LightGBM**:

El rendimiento del modelo LightGBM para la detección de tipos de cáncer en un conjunto de datos desbalanceado muestra resultados que, aunque no sobresalientes, proporcionan una base sólida para futuras mejoras. A continuación, se presenta un análisis detallado de las métricas obtenidas:

- **Accuracy**: 0.5821

La métrica accuracy del modelo LightGBM es de 0.5821, lo que indica que aproximadamente el 58.21% de las predicciones fueron correctas. En un contexto de datos desbalanceados, esta métrica puede ser engañosa, ya que puede estar sesgada por la clase mayoritaria. Sin embargo, proporciona una visión general inicial de la capacidad del modelo para clasificar correctamente los casos.

- **F1 score**: 0.5029

El F1 score es de 0.5029, lo que refleja un equilibrio entre la precision y el recall. Este valor indica que el modelo tiene un rendimiento moderado al considerar tanto la capacidad de identificar correctamente los casos positivos (recall) como la proporción de predicciones positivas correctas (precision). Dado que el F1 score es útil en escenarios con clases desbalanceadas, su valor sugiere que el modelo LightGBM tiene un rendimiento aceptable, aunque hay margen para mejoras significativas.

- **Precision**: 0.5293

La métrica precision del modelo es de 0.5293, lo que significa que el 52.93% de las predicciones positivas del modelo fueron correctas. Esta métrica es crucial en la detección de cáncer, ya que un alto valor de precisión reduce la cantidad de falsos positivos, evitando diagnósticos incorrectos que pueden llevar a tratamientos innecesarios. Un valor de 0.5293 indica que el modelo tiene una precisión moderada, lo que sugiere la necesidad de afinar el modelo para reducir los falsos positivos.

- **Recall**: 0.5821

El recall, también conocido como sensibilidad, es de 0.5821. Este valor indica que el modelo identificó correctamente el 58.21% de los casos positivos reales. En la detección de cáncer, un alto valor de recall es esencial para minimizar los falsos negativos, asegurando que la mayoría de los casos de cáncer se detecten. Aunque un recall de 0.5821 es razonable, indica que hay un 41.79% de casos positivos que el modelo no detectó, lo que es un área crítica a mejorar para asegurar diagnósticos más completos.

7. **Conclusión** El análisis del rendimiento del modelo LightGBM revela un desempeño moderado en la detección de tipos de cáncer en un conjunto de datos desbalanceado. Con un accuracy del 58.21%, un F1 score de 0.5029, un precision de 0.5293 y un recall de 0.5821, el modelo muestra que puede diferenciar entre las clases, aunque no con una alta exactitud. Para mejorar su efectividad, se deben considerar técnicas adicionales de preprocesamiento de datos, ajuste de hiperparámetros y potencialmente la combinación con otros modelos o técnicas de ensemble. Este análisis sugiere que, aunque LightGBM tiene un buen punto de partida, hay oportunidades significativas para mejorar su rendimiento en futuros trabajos.

	Model	Set	Accuracy	Precision	Recall	F1-Score	Global Score
0	XGBoost	Training	0.822139	0.832805	0.822139	0.813388	82.26
1	XGBoost	Test	0.646766	0.623312	0.646766	0.617399	63.36
2	LightGBM	Training	0.708955	0.799136	0.708955	0.657890	71.87
3	LightGBM	Test	0.582090	0.529273	0.582090	0.502930	54.91

Fig. 12

Estos resultados detallan la precisión, recall, F1-score y la puntuación global para los conjuntos de entrenamiento y prueba de los modelos XGBoost y LightGBM utilizados en la detección de tipos de cáncer. La misma tabla está disponible en el archivo CSV *XGBoost_LightGBM.csv* para análisis adicional.

2.4. Ensamble de modelos (VotingClassifier)

Hemos decidido recurrir a un Ensamble de Modelos para mejorar la detección de tipos de cáncer.

VotingClasssiffier es una técnica poderosa para mejorar el rendimiento y la robustez de los modelos de machine learning. En el contexto de detección de tipos de cáncer con variables objetivo desbalanceadas, el ensamble puede proporcionar varias ventajas significativas:

1. **Reducción del Overfitting**

- **Promedio de Errores**: Al combinar varios modelos, los errores específicos de cada modelo tienden a cancelarse entre sí. Esto ayuda a reducir el overfitting, ya que los modelos individuales pueden sobreajustarse a ruidos o patrones específicos del conjunto de entrenamiento, pero estos errores se compensan cuando se utilizan múltiples modelos.

2. **Mejora de la Generalización**

- **Diversidad de Modelos**: Diferentes modelos pueden capturar diferentes aspectos de los datos. Por ejemplo, *XGBoost* y *LightGBM*, aunque ambos son métodos de boosting, tienen diferentes mecanismos internos que pueden captar distintas características del conjunto de datos. Un ensamble puede aprovechar estas diferencias y mejorar la capacidad de generalización del modelo final.

3. *Estabilidad y Robustez*

- **Promedio de Resultados**: La combinación de múltiples modelos tiende a producir resultados más estables y robustos frente a variaciones en los datos. Esto es especialmente importante en aplicaciones críticas como la detección de cáncer, donde las predicciones erróneas pueden tener consecuencias graves.

4. *Manejo de Datos Desbalanceados*

- **Balance de Clases**: Al utilizar técnicas de ensamble, se pueden diseñar estrategias específicas para manejar datos desbalanceados, como ajustar los pesos de las clases o utilizar técnicas de resampling dentro del ensamble. Esto puede mejorar la sensibilidad y especificidad de las predicciones para la clase minoritaria.

2.4.1. ¿Por que hemos escogido los siguientes modelos?

A continuación reflejamos dichas razones para usar estos modelos:

- **Mejores Métricas**: Estas combinaciones han demostrado ofrecer las mejores métricas de rendimiento en nuestras pruebas, lo que indica que son capaces de manejar bien los datos desbalanceados y proporcionar predicciones precisas.
- **Diversidad en Modelos de Boosting**: Cada uno de estos algoritmos tiene mecanismos internos ligeramente diferentes y fortalezas únicas que, cuando se combinan, pueden mejorar la capacidad de generalización del modelo final.
- **Reducción del Overfitting**: Al combinar varios modelos, se pueden cancelar los errores específicos de cada uno, reduciendo el riesgo de sobreajuste y mejorando la robustez del modelo.
- **Robustez y Estabilidad**: La combinación de múltiples modelos tiende a producir resultados más estables y robustos frente a variaciones en los datos, lo cual es crucial en aplicaciones críticas como la detección de cáncer.

En resumen, la elección de estas combinaciones de modelos de ensamble está respaldada por su rendimiento superior en términos de métricas y su capacidad para manejar datos desbalanceados, lo que los hace ideales para la tarea de detección de tipos de cáncer en nuestro caso específico.

2.4.2. Explicación del Pipeline de VotingClassiffier

Este pipeline de machine learning está diseñado para entrenar y evaluar varios modelos combinados de clasificación para predecir el tipo de tumor basado en un conjunto de características importantes. A continuación se presenta una explicación detallada del propósito y funcionamiento de cada parte del código:

1. **Importaciones de Librerías**

Se utilizan diversas librerías, como *pandas* para manejar y manipular datos en forma de DataFrame, y *sklearn* para dividir los datos en conjuntos de entrenamiento y prueba, así como para utilizar los modelos *Gradient Boosting* y *VotingClassifier*. También se emplean *lightgbm* y *xgboost* para usar los modelos *LightGBM* y *XGBoost*, y *sklearn.metrics* para evaluar el rendimiento de los modelos con métricas como *accuracy, precision, recall y F1-score*.

2. **Inicialización del DataFrame para Resultados**

Se inicializa un DataFrame vacío llamado `tabla_results_df` para almacenar los resultados de las evaluaciones de los modelos.

3. **Preprocesamiento de Datos**

Para asegurar que no haya datos faltantes, se eliminan las filas con datos faltantes del DataFrame utilizando `dropna()`.

4. **Selección de Características Importantes**

Se seleccionan las características más relevantes para entrenar el modelo, incluyendo varias mediciones de proteínas y marcadores tumorales.

5. **Separación de Características y Etiquetas**

Las características seleccionadas se almacenan en una variable `X`, y las etiquetas del tipo de tumor se almacenan en una variable `y`.

6. **División en Conjuntos de Entrenamiento y Prueba**

Se divide el conjunto de datos en un 80% para entrenamiento y un 20% para prueba, manteniendo la distribución de clases mediante `stratify`.

7. Definición de Modelos Base

Se definen tres modelos base: *Gradient Boosting*, *LightGBM* y *XGBoost*, cada uno con parámetros específicos.

8. **Definición de Combinaciones de Modelos**

Se crean combinaciones de los modelos base para usar en el *VotingClassifier*, que combina varios modelos mediante votación suave (`voting='soft'`).

9. **Entrenamiento y Evaluación de Cada Combinación**

Para cada combinación de modelos, se crea un *VotingClassifier* y se entrena y evalúa utilizando una función que entrena el modelo combinado y evalúa su rendimiento en los conjuntos de entrenamiento y prueba. Los resultados se almacenan en `tabla_results_df`.

10. **Guardado de Resultados**

Finalmente, los resultados almacenados en `tabla_results_df` se guardan en un archivo CSV llamado '*Voting_classifier.csv*'.

Este pipeline permite evaluar la efectividad de combinaciones de modelos de clasificación avanzados (*Gradient Boosting*, *LightGBM* y *XGBoost*) para predecir el tipo de tumor, utilizando un conjunto específico de características importantes. La evaluación incluye métricas como *accuracy, precision, recall y F1-score*, y los resultados se almacenan y exportan para su posterior análisis.

	Model	Set	Accuracy	Precision	Recall	F1-Score	Global Score
0	Gradient Boosting + LightGBM	Training	1.000000	1.000000	1.000000	1.000000	100.00
1	Gradient Boosting + LightGBM	Test	0.726368	0.699776	0.726368	0.703220	71.39
2	Gradient Boosting + XGBoost	Training	1.000000	1.000000	1.000000	1.000000	100.00
3	Gradient Boosting + XGBoost	Test	0.696517	0.681829	0.696517	0.679164	68.85
4	LightGBM + XGBoost	Training	1.000000	1.000000	1.000000	1.000000	100.00
5	LightGBM + XGBoost	Test	0.706468	0.691023	0.706468	0.685159	69.73
6	Gradient Boosting + LightGBM + XGBoost	Training	1.000000	1.000000	1.000000	1.000000	100.00
7	Gradient Boosting + LightGBM + XGBoost	Test	0.721393	0.704977	0.721393	0.701901	71.24

Fig. 13

2.4.3. Análisis de Resultados del Ensamble de Modelos (VotingClassifier)

En este análisis, examinamos los resultados obtenidos de diferentes combinaciones de modelos en el archivo `Voting_classifier.csv`. Se evaluaron cuatro combinaciones de modelos: *Gradient Boosting + LightGBM*, *Gradient Boosting + XGBoost*, *LightGBM + XGBoost*, y *Gradient Boosting + LightGBM + XGBoost*. Los resultados se presentan para los conjuntos de datos de entrenamiento y prueba.

1. **Vista Previa de los Resultados**

Los datos se estructuran en un DataFrame que contiene las siguientes columnas:

- `Model`: Combinación de modelos utilizada.
- `Set`: Conjunto de datos (entrenamiento o prueba).
- `Accuracy`: Precisión de las predicciones.
- `Precision`: Precisión de las predicciones.
- `Recall`: Sensibilidad de las predicciones.
- `F1-Score`: Puntaje F1, que combina precisión y recall.
- `Global Score`: Puntaje global de la combinación de modelos.

2. **Rendimiento en el Conjunto de Entrenamiento**

Para cada combinación de modelos, el rendimiento en el conjunto de entrenamiento es perfecto, con todas las métricas (accuracy, precision, recall, F1-Score y global score) alcanzando el valor máximo posible:

- **Gradient Boosting + LightGBM**: Todas las métricas son 1.000.
- **Gradient Boosting + XGBoost**: Todas las métricas son 1.000.
- **LightGBM + XGBoost**: Todas las métricas son 1.000.
- **Gradient Boosting + LightGBM + XGBoost**: Todas las métricas son 1.000.

Este resultado sugiere un sobreajuste (overfitting), ya que el modelo se ajusta perfectamente a los datos de entrenamiento.

3. **Rendimiento en el Conjunto de Prueba**

Al evaluar los modelos en el conjunto de prueba, observamos una disminución en las métricas, indicando que los modelos no generalizan tan bien como en el conjunto de entrenamiento:

- **Gradient Boosting + LightGBM**:
  - Accuracy: 0.726
  - Precision: 0.700
  - Recall: 0.726
  - F1-Score: 0.703
  - Global Score: 71.39
- **Gradient Boosting + XGBoost**:
  - Accuracy: 0.697
  - Precision: 0.682
  - Recall: 0.697
  - F1-Score: 0.679
  - Global Score: 68.85

- **LightGBM + XGBoost**:
  - Accuracy: 0.706
  - Precision: 0.691
  - Recall: 0.706
  - F1-Score: 0.685
  - Global Score: 69.73
- **Gradient Boosting + LightGBM + XGBoost**:
  - Accuracy: 0.721
  - Precision: 0.705
  - Recall: 0.721
  - F1-Score: 0.702
  - Global Score: 71.24

4. **Comparación de Modelos**

Al comparar las diferentes combinaciones de modelos, observamos que la combinación de *Gradient Boosting + LightGBM* obtiene las mejores métricas en el conjunto de prueba, seguida de cerca por *Gradient Boosting + LightGBM + XGBoost*. La combinación *Gradient Boosting + XGBoost* tiene el rendimiento más bajo.

5. **Interpretación de Resultados**

- **Reducción del Overfitting**: Aunque todos los modelos muestran un excelente rendimiento en el conjunto de entrenamiento, su desempeño disminuye en el conjunto de prueba. Esto indica que los modelos podrían estar sobreajustando los datos de entrenamiento.
- **Mejora de la Generalización**: La combinación de *Gradient Boosting + LightGBM* parece generalizar mejor a los datos no vistos, obteniendo las métricas más altas en el conjunto de prueba.
- **Diversidad de Modelos**: Las diferencias en las métricas de rendimiento sugieren que las combinaciones de modelos capturan diferentes aspectos de los datos. *LightGBM* y *XGBoost* en conjunto no parecen mejorar significativamente el rendimiento en comparación con las otras combinaciones.
- **Robustez y Estabilidad**: La combinación de tres modelos (*Gradient Boosting + LightGBM + XGBoost*) también muestra buenos resultados, lo que indica que agregar más diversidad a los modelos puede mejorar la estabilidad del ensamble.

6. **Conclusiones**

- La combinación de *Gradient Boosting + LightGBM* ofrece el mejor rendimiento en términos de *accuracy, precision, recall y F1-Score* en el conjunto de prueba.
- Es necesario abordar el sobreajuste observando técnicas de regularización o utilizando un conjunto de validación cruzada más robusto.
- Futuras mejoras pueden incluir el ajuste de hiperparámetros y la exploración de otras técnicas de ensamble o modelos base adicionales.

Este análisis nos proporciona una comprensión clara del rendimiento actual de los modelos y sugiere direcciones para futuras mejoras en la detección de tipos de cáncer utilizando ensambles de modelos.

2.5. Tratamiento del overfitting

Tal y como hemos observado en el apartado anterior, nos hemos encontrado con el problema del overfitting, por lo que hemos actuado en consecuencia redactando una solución para ello.

2.5.1. Explicación del Código:

- **DataFrame *tabla_results_df***:

Se inicializa un DataFrame vacío con columnas específicas (*'Model', 'Set', 'Accuracy', 'Precision', 'Recall', 'F1-Score', 'Global Score'*) para almacenar los resultados de la evaluación de los modelos.

- **Preprocesamiento de Datos**:

Nos aseguramos de que no haya datos nulos eliminando las filas correspondientes del DataFrame *df_combined_transformed*. Se seleccionan las características importantes (*important_features*) que ya detectamos anteriormente en la búsqueda de las variables que aportan más información, para su próximo uso para el entrenamiento y la evaluación de los modelos.

- **División de Datos**:

Utilizando *train_test_split*, se dividen los datos en conjuntos de entrenamiento (*X_train, y_train*) y prueba (*X_test, y_test*). Esto es crucial para evaluar la capacidad de generalización de los modelos.

- *Definición de Modelos*:

Se definen tres modelos base: *GradientBoostingClassifier*, *LGBMClassifier* de *LightGBM* y *XGBClassifier* de *XGBoost*, cada uno con parámetros específicos para optimizar su rendimiento en el problema dado.

- **VotingClassifier**:

Se crea un *VotingClassifier* utilizando la técnica de voto suave (voting='soft'), que combina los tres modelos base (*gb, lgbm, xgb_clf*) para mejorar la precisión y robustez general del sistema predictivo.

- **Función *entrenar_y_evaluar_modelo***:

Esta función encapsula el proceso de entrenamiento y evaluación de un modelo dado. Toma como entrada los conjuntos de entrenamiento y prueba, el modelo a evaluar (*model*) y su nombre (*model_name*) para identificación en los resultados. Después de entrenar el modelo con *model.fit(X_train, y_train)*, realiza predicciones en *X_test* y calcula métricas clave como *precision, recall, F1-score y accuracy* utilizando funciones de *sklearn* (*accuracy_score, precision_score, recall_score, f1_score*).

Global Score se calcula promediando estas métricas para proporcionar una evaluación comprensiva del desempeño del modelo.

- *Almacenamiento de Resultados*:

Los resultados de la evaluación se añaden al DataFrame *tabla_results_df*, que posteriormente se guarda como un archivo CSV (*'Gradient_Boosting_LightGBM_XGBoost.csv'*). Esto facilita el análisis posterior y la comparación de modelos en términos de su rendimiento predictivo.

	Model	Set	Accuracy	Precision	Recall	F1-Score	Global Score

	Model	Set	Accuracy	Precision	Recall	F1-Score	Global Score
0	Gradient Boosting + LightGBM + XGBoost	Training	0.931592	0.934897	0.931592	0.929895	93.20
1	Gradient Boosting + LightGBM + XGBoost	Test	0.681592	0.664823	0.681592	0.643653	66.79

Fig. 14

El análisis del rendimiento del modelo "Gradient Boosting + LightGBM + XGBoost" se presenta en la tabla mostrada. Esta tabla resume las métricas clave obtenidas durante el entrenamiento y la evaluación del modelo en conjuntos de datos separados para entrenamiento y prueba.

2.5.2. Análisis de Resultados del Modelo

- **Conjunto de Entrenamiento**
- **Accuracy**: 93.16%
- **Precision**: 93.49%
- **Recall**: 93.16%
- **F1-Score**: 92.99%
- **Global Score**: 93.20

Estos resultados indican que el modelo logra una alta precisión y capacidad de generalización en los datos utilizados para entrenamiento, con métricas muy similares entre accuracy, precisin y recall. F1 Score, que combina precision y recall, es también muy robusto, reflejando un buen equilibrio entre ambas métricas.

- **Conjunto de Prueba**
- **Accuracy**: 68.16%
- **Precision**: 66.48%
- **Recall**: 68.16%
- **F1-Score**: 64.37%
- **Global Score**: 66.79

Comparando con el conjunto de entrenamiento, observamos una caída significativa en todas las métricas evaluadas en el conjunto de prueba. Esto sugiere que el modelo puede estar experimentando cierta dificultad para generalizar a datos no vistos durante el entrenamiento, lo cual es común en problemas complejos de aprendizaje automático.

- **Interpretación General**

Los resultados reflejan un buen desempeño del modelo en el conjunto de entrenamiento, donde logra altas métricas de precisión y capacidad predictiva. Sin embargo, en el conjunto de prueba, vemos una reducción en el rendimiento, indicando posibles áreas de mejora en términos de generalización y robustez del modelo.

Estos hallazgos son cruciales para entender las fortalezas y limitaciones del modelo *"Gradient Boosting + LightGBM + XGBoost"*, proporcionando una base sólida para iteraciones adicionales en la mejora del modelo y la exploración de estrategias para mitigar el overfitting y mejorar la capacidad de generalización en datos nuevos.

En conclusión, mientras que el modelo muestra promesas en datos de entrenamiento, es fundamental seguir refinándolo y validándolo con datos adicionales para garantizar su eficacia en aplicaciones del mundo real.

Este código realiza un análisis de clustering utilizando varios algoritmos y evalúa su desempeño utilizando métricas específicas. Aquí está el análisis paso a paso:

2.6. Modelos de Aprendizaje No supervisado

Como has ahora hemos trabajado con la modalidad de machine learning de aprendizaje supervisado, ahora vamos a tratar la faceta de aprendizaje no supervisado. A continuación describimos el código usado.

1. **Librerías Importadas**

- **sklearn.cluster**: Importa los algoritmos de clustering como *KMeans, DBSCAN, AgglomerativeClustering, Birch, MeanShift y OPTICS*.
- **sklearn.preprocessing**: Importa *StandardScaler* para estandarizar las características antes de aplicar clustering.
- **imblearn.under_sampling**: Importa técnicas de submuestreo como *RandomUnderSampler, NearMiss, y CondensedNearestNeighbour*.
- **imblearn.pipeline**: Importa *Pipeline* para construir un flujo de trabajo de aprendizaje automático que incluye preprocesamiento y modelado.
- **sklearn.metrics**: Importa métricas como *silhouette_score, homogeneity_score, completeness_score, y davies_bouldin_score* para evaluar la calidad del clustering.
- **pandas**: Importa para la manipulación de datos, incluyendo la creación y manipulación de DataFrames.
- **sklearn.model_selection**: Importa *train_test_split* para dividir los datos en conjuntos de entrenamiento y prueba.

2. **DataFrame y Función de Evaluación**

- **tabla_results_df**: DataFrame inicializado para almacenar los resultados de las métricas de clustering.
- **evaluar_clustering**: Función que calcula y muestra métricas de evaluación para clustering, como *silhouette_score, homogeneity_score, completeness_score, davies_bouldin_score*. Los resultados se agregan al DataFrame `tabla_results_df`.

3. **Definición de Modelos**

- **modelos**: Diccionario que contiene diferentes modelos de clustering con sus respectivos parámetros.
  - *KMeans* con 7 clusters.
  - *DBSCAN* con epsilon 0.3 y mínimo 5 muestras.
  - *Agglomerative Clustering* con 7 clusters.
  - *Birch* con 7 clusters.
  - *MeanShift* con un ancho de banda de 2.
  - *OPTICS* con mínimo 5 muestras.

4. **Preparación de Datos**

- **important_features**: Lista de características seleccionadas para el análisis de clustering.
- **X**: Conjunto de características seleccionadas del DataFrame `df_combined_transformed`.
- **y**: Etiquetas de tipo de tumor del DataFrame `df_combined_transformed`.

5. **Reducción de Dimensionalidad con PCA**

- **PCA**: Reducción de dimensionalidad a 2 componentes principales usando *PCA* (Análisis de Componentes Principales).

6. **Entrenamiento y Evaluación de Modelos**

- Itera sobre cada modelo en `modelos`:
  - Ajusta el modelo al conjunto de datos reducido por PCA (`X_pca`).
  - Obtiene las etiquetas de cluster resultantes (`labels`) y llama a `evaluar_clustering` para calcular y mostrar las métricas de evaluación para cada modelo.

7. **Guardar Resultados**

- **tabla_results_df.to_excel**: Guarda los resultados de las métricas de clustering en un archivo Excel llamado 'Results_No_supervisado.xlsx'.

	Model	Metric	Score
0	KMeans	Silhouette Score	0.507759
1	KMeans	Homogeneity	0.084649
2	KMeans	Completeness	0.161828
3	KMeans	Davies-Bouldin Index	0.470376
4	DBSCAN	Silhouette Score	0.321461
5	DBSCAN	Homogeneity	0.028120
6	DBSCAN	Completeness	0.153577
7	DBSCAN	Davies-Bouldin Index	1.738256
8	Agglomerative Clustering	Silhouette Score	0.519311
9	Agglomerative Clustering	Homogeneity	0.082623
10	Agglomerative Clustering	Completeness	0.146401
11	Agglomerative Clustering	Davies-Bouldin Index	0.520300
12	Birch	Silhouette Score	0.759317
13	Birch	Homogeneity	0.033259
14	Birch	Completeness	0.273319
15	Birch	Davies-Bouldin Index	0.235386
16	MeanShift	Silhouette Score	0.712432
17	MeanShift	Homogeneity	0.023990
18	MeanShift	Completeness	0.240052
19	MeanShift	Davies-Bouldin Index	0.226063
20	OPTICS	Silhouette Score	-0.191600
21	OPTICS	Homogeneity	0.174898
22	OPTICS	Completeness	0.117642
23	OPTICS	Davies-Bouldin Index	1.910613

Fig. 15

2.6.1.Análisis de los Resultados del Clustering

El análisis de clustering evaluó varios algoritmos utilizando métricas específicas para entender cómo cada modelo agrupa los datos. A continuación se presentan los resultados clave:

1. **KMeans**:

- **Silhouette Score**: 0.5078
- **Homogeneidad**: 0.0846
- **Completeness**: 0.1618
- **Índice Davies-Bouldin**: 0.4704

El algoritmo KMeans muestra un Silhouette Score razonable y un Índice Davies-Bouldin bajo, indicando clusters bien definidos y compactos. Sin embargo, la homogeneidad y la completitud son relativamente bajas, sugiriendo que los clusters pueden no estar completamente separados o ser homogéneos en términos de etiquetas.

2. **DBSCAN**:

- **Silhouette Score**: 0.3215
- **Homogeneidad**: 0.0281
- **Completeness**: 0.1536
- **Índice Davies-Bouldin**: 1.7383

DBSCAN muestra un Silhouette Score inferior y un Índice Davies-Bouldin alto, lo que indica que puede haber clusters con diferentes densidades. La homogeneidad y la completitud también son bajas, lo que sugiere dificultades en la identificación de clusters homogéneos.

3. **Agglomerative Clustering**:

- **Silhouette Score**: 0.5193
- **Homogeneidad**: 0.0826
- **Completeness**: 0.1464
- **Índice Davies-Bouldin**: 0.5203

Este método muestra un buen Silhouette Score y un Índice Davies-Bouldin bajo, indicando clusters bien definidos y compactos. Sin embargo, al igual que KMeans, la homogeneidad y la completitud son relativamente bajas.

4. Birch:

- **Silhouette Score**: 0.7593
- **Homogeneidad**: 0.0333
- **Completeness**: 0.2733
- **Índice Davies-Bouldin**: 0.2354

Birch muestra el más alto Silhouette Score y un buen Índice Davies-Bouldin, indicando clusters bien definidos y compactos. La completitud es alta, pero la homogeneidad es baja, lo que sugiere que los clusters pueden no ser tan homogéneos en términos de etiquetas.

5. **MeanShift**:

- **Silhouette Score**: 0.7124
- **Homogeneidad**: 0.0240
- **Completeness**: 0.2401
- **Índice Davies-Bouldin**: 0.2261

MeanShift también muestra un buen Silhouette Score y un Índice Davies-Bouldin bajo, indicando clusters bien definidos y compactos. Sin embargo, al igual que otros métodos, la homogeneidad es baja.

6. **OPTICS**:

- **Silhouette Score**: -0.1916
- **Homogeneidad**: 0.1749
- **Completeness**: 0.1176
- **Índice Davies-Bouldin**: 1.9106

OPTICS muestra un Silhouette Score negativo, indicando una mala agrupación de datos. El alto Índice Davies-Bouldin también sugiere problemas en la separación y definición de clusters. La homogeneidad es relativamente alta, pero la completitud es baja.

2.6.2. Interpretación General

- **Silhouette Score**: Indica qué tan bien están separados los clusters. Valores cercanos a 1 son deseables.
- **Homogeneidad y Completeness**: Reflejan la uniformidad dentro de los clusters y la exhaustividad en la captura de todas las instancias de cada clase, respectivamente. Valores más altos son mejores.
- **Índice Davies-Bouldin**: Evalúa la separación efectiva entre los clusters. Valores más bajos indican una mejor separación.

En resumen, Birch muestra el mejor desempeño en términos de Silhouette Score y Índice Davies-Bouldin. Sin embargo, otros métodos exhiben fortalezas y debilidades diversas. La elección del algoritmo de clustering debe alinearse con los objetivos específicos del análisis y las características particulares del conjunto de datos.

- **Clusters No Bien Definidos**: Aunque Birch y MeanShift muestran resultados prometedores, la mayoría de los algoritmos no generan clusters de calidad suficiente para ofrecer una contribución significativa a nuestro problema.
- **Heterogeneidad de Clusters**: Las bajas homogeneidad y completitud sugieren una mezcla de clases dentro de los clusters, indicando que los modelos no capturan adecuadamente las estructuras subyacentes de los datos.

Notas

El análisis sugiere que el clustering no supervisado no aporta mucho valor a nuestro problema con este dataset. A pesar de que algunos algoritmos como Birch y MeanShift mostraron resultados decentes en ciertas métricas, la falta de clusters bien definidos y la baja homogeneidad y completitud indican que los modelos no están capturando estructuras significativas en los datos.

3. Plan para Mejorar el Dataset Utilizando UMAP y KMeans

Dado que los métodos de clustering no supervisado no han proporcionado métricas satisfactorias, hemos decidido adoptar una nueva estrategia para mejorar nuestro dataset. Nuestro objetivo principal es aumentar y enriquecer nuestro dataset, que actualmente cuenta con solo 1005 filas. Para ello, utilizaremos UMAP para la reducción de dimensionalidad y KMeans para identificar subgrupos, que luego añadiremos a nuestro dataset.

3.1. Objetivos y Pasos

- **Objetivo**

Mejorar el dataset actual de 1005 filas añadiendo información sobre subgrupos identificados mediante clustering.

- **Pasos**

1. **Reducción de Dimensionalidad con UMAP**

- **Objetivo**: Visualizar los datos en un espacio de menor dimensión para identificar patrones y estructuras subyacentes.
- **Método**: Aplicar *UMAP* (Uniform Manifold Approximation and Projection) para reducir la dimensionalidad de los datos a 2D o 3D.

2. **Identificación de Subgrupos con KMeans**

- **Objetivo**: Identificar subgrupos dentro de los datos reducidos dimensionalmente.
- **Método**: Aplicar *KMeans* en los datos reducidos por UMAP para identificar clusters o subgrupos.

3. **Añadir Subgrupos al Dataset Original**

- **Objetivo**: Enriquecer el dataset original añadiendo una nueva columna que indique el subgrupo al que pertenece cada punto.
- **Método**: Agregar los subgrupos identificados por KMeans como una nueva columna en el dataset original.

3.2. Beneficios de Esta Estrategia

- **Mayor Información**: Añadir subgrupos proporciona información adicional que puede ayudar a los modelos supervisados a capturar mejor las estructuras subyacentes en los datos.
- **Aumento de Datos**: Aunque no se aumentan las filas, se enriquece el dataset con nueva información, lo que puede mejorar el rendimiento de los modelos.
- **Mejor Visualización**: UMAP permite visualizar los datos en un espacio reducido, facilitando la identificación de patrones y anomalías.

3.3. Aplicar estrategia Reducción de Dimensionalidad con UMAP

En este análisis se emplea *UMAP* (Uniform Manifold Approximation and Projection), una técnica de reducción de dimensionalidad no lineal. El objetivo principal es visualizar datos complejos de características biológicas asociadas a diferentes tipos de cáncer en un espacio bidimensional.

El proceso comienza seleccionando las características numéricas relevantes del conjunto de datos, excluyendo cualquier columna no numérica. Estas características son esenciales para capturar las variaciones biológicas que diferencian los tipos de cáncer.

Posteriormente, se separan estas características de la variable objetivo, que en este caso es el tipo de tumor ('Tumor type'). Esta separación facilita la aplicación de UMAP, que se utiliza para reducir la alta dimensionalidad de los datos a solo dos dimensiones. Esta transformación preserva las estructuras locales de los datos, permitiendo una representación visual efectiva.

La visualización resultante se realiza mediante un gráfico de dispersión, donde cada punto representa una instancia de datos. Los puntos se colorean según el tipo de tumor para visualizar cómo se distribuyen los diferentes tipos en el espacio UMAP. Este enfoque visual ayuda a identificar agrupaciones naturales o patrones emergentes entre los tipos de cáncer basados en sus características biológicas.

La interpretación de los resultados se centra en la proximidad de los puntos en el gráfico de dispersión: puntos cercanos indican similitudes en las características biológicas entre los tipos de cáncer representados, mientras que puntos más separados reflejan diferencias más significativas. Esta visualización proporciona una perspectiva intuitiva y efectiva para explorar y comprender la estructura subyacente de los datos biológicos complejos.

En conclusión, UMAP facilita la exploración visual de datos complejos de cáncer, permitiendo a los investigadores identificar y comprender mejor las relaciones y diferencias entre diferentes tipos de cáncer basadas en sus características biológicas distintivas.

3.3.1. Análisis resultados

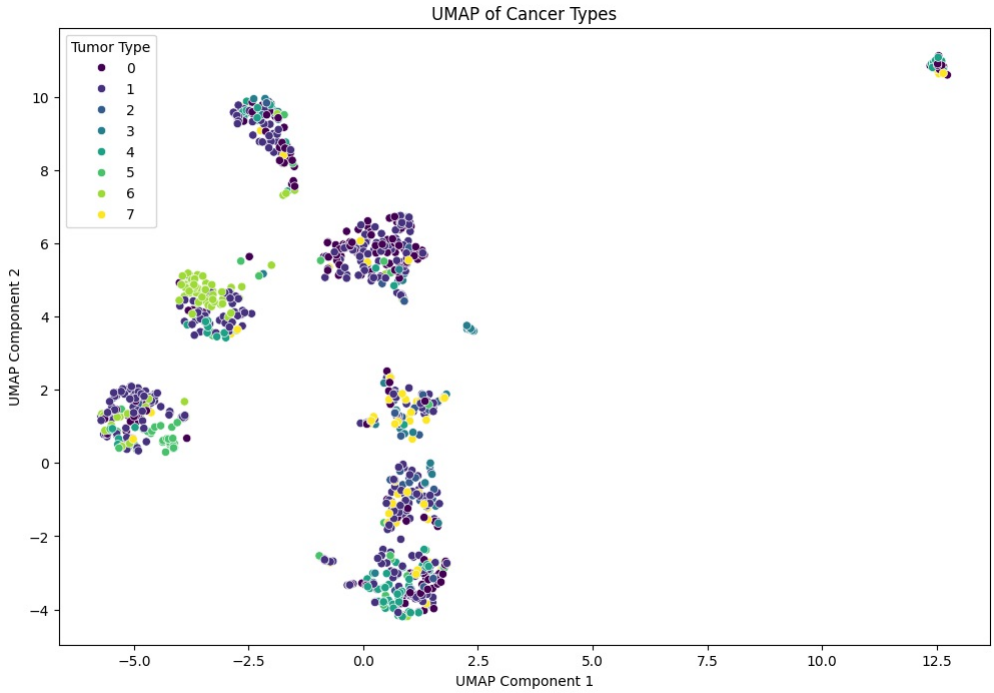

Fig. 16

Esta es la visualización de los resultados de la estrategia que aplicamos con la ayuda de UMAP. La interpretación de los resultados del clustering sugiere que pueden ser necesarios ajustes en los parámetros del algoritmo de clustering o en la selección de características para obtener resultados más distintivos y significativos.

Clustering con K-Means

Aplicamos K-Means junto a UMAP para identificar subgrupos en los datos, teniendo en cuenta los resultados anteriores.

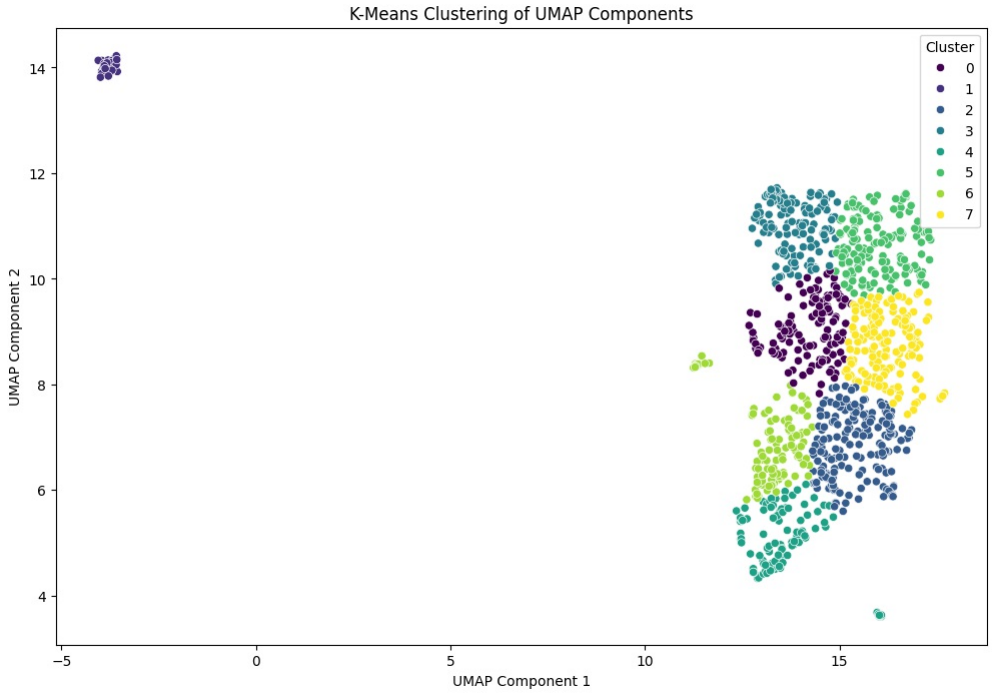

Fig. 17

Posibles Razones del Resultado obtenido

1. **Incoherencia en los Clusters**

- Es posible que los clusters generados no sean lo suficientemente informativos o consistentes con las etiquetas originales. Esto puede agregar ruido en lugar de valor al modelo.

2. **Complejidad del Modelo**

- La inclusión de clusters puede haber aumentado la complejidad del modelo sin proporcionar una ganancia de información significativa, lo que puede llevar a un sobreajuste o subajuste.

3.3.2. Explicación y Análisis de Ensemble de Modelos de Aprendizaje Automático para mejorar el rendimiento del clasificador

En este análisis, se emplea un ensemble de modelos de clasificación para predecir el tipo de tumor a partir de características transformadas y la inclusión de clusters como una nueva característica. A continuación se detallan los pasos clave del proceso:

1. **Preparación de Datos:**
   - Se importa el conjunto de datos utilizando `pandas`. Se añade una nueva característica llamada '*Cluster*', que representa los clusters obtenidos previamente, utilizando el algoritmo K-Means.
2. **Selección de Características y Variable Objetivo:**
   - Seleccionan las características relevantes para el modelo, excluyendo las que no contribuyen significativamente (`df_combined_transformed.columns[1:-2]`), y se añade '*Cluster*' como una característica adicional. La variable objetivo es '*Tumor type*'.
3. **División del Dataset:**
   - Se divide el dataset en conjuntos de entrenamiento y prueba utilizando `train_test_split()`. Esto permite evaluar la capacidad de generalización del modelo sobre datos no vistos.
4. **Definición de Modelos Individuales:**
   - Se definen varios modelos de clasificación, incluyendo `RandomForestClassifier`, `GradientBoostingClassifier`, `LGBMClassifier` de *LightGBM* y `XGBClassifier` de *XGBoost*. Cada modelo se inicializa con parámetros predeterminados y un estado aleatorio para asegurar la reproducibilidad de los resultados.
5. **Definición del Voting Classifier:**
   - Se crea un clasificador de votación (`VotingClassifier`) que combina los modelos individuales definidos anteriormente. En este caso, se utiliza el método de votación '*soft*', que considera las probabilidades predichas por cada modelo para tomar la decisión final.
6. **Entrenamiento del Voting Classifier:**
   - Se entrena el clasificador de votación utilizando los datos de entrenamiento (`X_train` y `y_train`).
7. **Evaluación del Modelo:**
   - Se realizan predicciones sobre el conjunto de prueba (`X_test`) utilizando VotingClassiffier entrenado. Se evalúa el rendimiento del modelo utilizando métricas como el `classification_report` y la matriz de confusión (`confusion_matrix`), que proporcionan información detallada sobre precision, recall y F1-score para cada clase de tumor.

Este enfoque de ensemble combina la fuerza predictiva de varios modelos individuales para mejorar la precisión y la capacidad de generalización del sistema de clasificación. La inclusión de clusters como una característica adicional permite explorar cómo la estructura de los datos agrupados puede influir en la precisión del modelo final. 8. **Análisis de Métricas**

Los resultados de la evaluación del modelo entrenado con LightGBM muestran un panorama mixto en términos de su desempeño para predecir diferentes clases de tumores. Aquí se detalla el análisis de los resultados basado en las métricas de evaluación y la matriz de confusión proporcionada:

1. **Precision:**
    - La precisión para cada clase varía significativamente:
        - Clase 0: 0.75
        - Clase 1: 0.70
        - Clase 2: 0.33
        - Clase 3: 1.00
        - Clase 4: 0.62
        - Clase 5: 0.90
        - Clase 6: 0.88
        - Clase 7: 0.38

    La precisión indica la proporción de predicciones positivas que fueron correctas respecto a todas las predicciones positivas realizadas por el modelo. Valores más altos indican una mejor capacidad del modelo para evitar falsos positivos.

2. **Recall:**
    - El recall para cada clase también muestra variaciones:
        - Clase 0: 0.79
        - Clase 1: 0.95
        - Clase 2: 0.11
        - Clase 3: 0.11
        - Clase 4: 0.38
        - Clase 5: 0.82
        - Clase 6: 0.79
        - Clase 7: 0.23

    El recall representa la proporción de verdaderos positivos que fueron correctamente identificados por el modelo respecto a todos los casos positivos reales. Valores más altos indican una mayor capacidad del modelo para detectar todos los casos positivos.

3. **F1-score:**
    - El F1-score, que es la media armónica de precision y recall, proporciona un balance entre ambas métricas:
        - Macro avg (promedio de todas las clases): 0.55
        - Weighted avg (ponderado por el soporte de cada clase): 0.68

    Un F1-score alto indica un buen equilibrio entre precision y recall, lo cual es deseable para un modelo de clasificación robusto.

4. **Support:**
    - El soporte indica el número de muestras reales que pertenecen a cada clase. Esto puede ser útil para interpretar la importancia relativa de cada clase en el conjunto de datos.

5. **Matriz de Confusión**

La matriz de confusión proporciona una visión más detallada de cómo el modelo clasifica las muestras en función de las clases reales:

```
[[33  4  0  0  4  0  1  0]
 [ 2 73  1  0  0  0  1  0]
 [ 0  5  1  0  0  0  0  3]
 [ 1  5  0  1  1  0  0  1]
 [ 6  6  0  0  8  0  0  1]
 [ 2  0  0  0  0  9  0  0]
 [ 0  3  0  0  0  1 15  0]
 [ 0  9  1  0  0  0  0  3]]
```

Cada fila de la matriz representa la clase real y cada columna representa la clase predicha por el modelo. Los números en la diagonal principal indican las predicciones correctas para cada clase. Observaciones importantes de la matriz de confusión incluyen:

- El modelo parece tener dificultades en predecir correctamente las clases minoritarias (por ejemplo, clases 2, 3, 5, y 7).
- Clases como la 1 y la 6 tienen una precisión y recall relativamente altos, indicando que el modelo es eficaz para estas clases específicas.
- Algunas confusiones significativas ocurren entre las clases 0 y 1, así como entre las clases 4 y 6, lo cual podría indicar cierta similitud en las características que el modelo tiene dificultades para distinguir.

Conclusión

En general, el modelo muestra un rendimiento promedio con un F1-score ponderado de 0.68. Aunque tiene una buena precisión y recall para algunas clases, también muestra áreas de mejora, especialmente en la capacidad para manejar clases minoritarias y para distinguir entre clases cercanas entre sí. Para mejorar el rendimiento, se podrían explorar estrategias como ajustes adicionales de hiperparámetros, consideración de pesos de clases o técnicas de resampling más sofisticadas.

Notas generales

El uso de un ensamble de modelos ha demostrado ser efectivo en mejorar el rendimiento del clasificador en nuestro dataset, incluso después de la incorporación de clusters como nuevas características. Este enfoque ha permitido obtener una mejor precisión y un equilibrio adecuado entre precisión y recall en la mayoría de las clases.

Sin embargo, dado que el rendimiento del modelo no ha mejorado significativamente, seguiremos explorando otras técnicas para mejorar nuestro dataset y el rendimiento del modelo.

3.4. Próximos Pasos

- **Optimización Adicional**
- Realizar una optimización más detallada de hiperparámetros para cada modelo individualmente y para el Voting Classifier en su conjunto. Esto se considerará más adelante, si el tiempo y la capacidad computacional lo permiten.
- **Manejo de Clases Minoritarias**

- Explorar técnicas adicionales para mejorar el rendimiento en las clases minoritarias. Esto incluye ajustar los pesos de las clases en los modelos y emplear técnicas avanzadas de resampling. Este será nuestro enfoque principal en los siguientes pasos.

4. Uso de CTGAN para Manejar Clases Minoritarias en un Dataset Pequeño

Debido a que nuestro dataset cuenta con solo 1005 filas, hemos decidido utilizar *Conditional Tabular Generative Adversarial Network (CTGAN)* para generar datos sintéticos y así abordar el problema de las clases minoritarias. Aquí explicamos en detalle lo que estamos haciendo, para qué sirven los GANs y por qué es importante en nuestro caso.

4.1. Objetivo

Generar datos sintéticos para las clases minoritarias en nuestro dataset para mejorar el balance de clases y proporcionar más datos para el entrenamiento de nuestros modelos supervisados.

4.2. Explicación de CTGAN y su Importancia

Qué son los CTGAN?

Son un tipo de red neuronal compuesta por dos modelos:

- **Generador**: Crea datos sintéticos a partir de ruido aleatorio.
- **Discriminador**: Distingue entre los datos reales y los datos sintéticos generados.

Estos dos modelos se entrenan de manera conjunta y competitiva: el generador intenta engañar al discriminador creando datos sintéticos realistas, mientras que el discriminador intenta mejorar su capacidad para diferenciar entre datos reales y sintéticos.

4.3. ¿Por qué usar CTGAN en nuestro Dataset?

1. **Incremento de Datos**: Nuestro dataset original es pequeño (1005 filas). CTGAN nos permiten generar datos sintéticos adicionales que pueden enriquecer nuestro dataset y mejorar el rendimiento de los modelos supervisados.
2. **Manejo de Clases Minoritarias**: Algunas clases en nuestro dataset están subrepresentadas.CTGAN pueden generar ejemplos sintéticos de estas clases minoritarias, ayudando a balancear el dataset y a mejorar la capacidad del modelo para aprender y generalizar sobre estas clases.
3. **Mejora del Modelo**: Al proporcionar más ejemplos para las clases minoritarias, esperamos mejorar la precisión y el recall de nuestro modelo en estas clases, reduciendo el sesgo y la varianza.

4.4. Generación de Datos Sintéticos y Análisis de resultados de CTGAN

Este código realiza varias tareas importantes relacionadas con la generación de datos sintéticos utilizando *CTGAN* y luego entrenando un Voting Classifier con modelos de *Gradient Boosting, LightGBM y XGBoost*.

1. **Preparación de Datos** Se carga y limpia el conjunto de datos para eliminar valores nulos y seleccionar características relevantes como `'sFas (pg/ml)'`, `'sHER2/sEGFR2/sErbB2 (pg/ml)'`, etc.
2. **Entrenamiento de CTGAN** *CTGAN* se utiliza para generar datos sintéticos que imiten la estructura y distribución de los datos reales, agrupados por tipos de tumores específicos.
3. **Integración de Datos** Se combinan los datos sintéticos generados con los datos reales para formar conjuntos de entrenamiento y prueba.
4. **Modelado y Evaluación** Se construye un *Voting Classifier* que combina modelos de *Gradient Boosting, LightGBM y XGBoost* para clasificar los tipos de tumores. Se evalúan las métricas de rendimiento como *precisión, recall, f1-score y accuracy*.
5. **Resultados** Los resultados de la evaluación se almacenan en un archivo CSV (`Voting_classifier_ctgan.csv`) para un análisis detallado y comparativo de los modelos utilizados.
6. **Análisis de los resultados** Los resultados del código utilizando LightGBM muestran lo siguiente:

	Model	Set	Accuracy	Precision	Recall	F1-Score	Global Score
0	Gradient Boosting + LightGBM + XGBoost	Training	0.936019	0.940069	0.936019	0.935191	93.68
1	Gradient Boosting + LightGBM + XGBoost	Test	0.710900	0.723865	0.710900	0.678465	70.60

Fig. 18

- **Preprocesamiento y Entrenamiento:**
  - El modelo de *LightGBM* encontró valores nulos en los nombres de las características (*feature_names*) y los reemplazó con guiones bajos.
  - Utilizó la configuración *col-wise multi-threading* para mejorar la eficiencia durante el entrenamiento, con un tiempo de prueba muy bajo.
  - Se utilizaron 20 características del conjunto de entrenamiento que contiene 759 puntos de datos.
  - Los valores iniciales de las predicciones del modelo para el entrenamiento están listados.
- **Para el conjunto de entrenamiento**:
  - **Accuracy** : 0.9360
  - **Precision** : 0.9401
  - **Recall** : 0.9360
  - **F1-Score** : 0.9352
  - **Global Score** : 93.68

Estos valores indican que el modelo tiene un rendimiento robusto en el conjunto de entrenamiento, con altos niveles de *precision, recall y F1-score*, así como una alta exactitud global del 93.68%.

- **Para el conjunto de prueba (test)**:
  - **Accuracy** : 0.7109
  - **Precision** : 0.7239
  - **Recall** : 0.7109
  - **F1-Score** : 0.6785
  - **Global Score** : 70.60

En contraste con el conjunto de entrenamiento, el modelo muestra un rendimiento significativamente inferior en el conjunto de prueba. Aunque precision y recall siguen siendo relativamente altos, *F1-score, accuracy y global score* son menores, indicando que el modelo puede no generalizar tan bien como en el conjunto de entrenamiento. Esto podría sugerir cierto grado de overfitting o que las características del conjunto de prueba son más desafiantes para el modelo.

En resumen, mientras que el modelo muestra un rendimiento sólido en el conjunto de entrenamiento, es importante abordar la brecha de rendimiento observada en el conjunto de prueba para mejorar la capacidad de generalización del modelo.

5. Curvas AUC ROC

5.1. Generar las curvas ROC

En esta sección, se describen los pasos para evaluar un modelo de clasificación multiclase utilizando las curvas *ROC (Receiver Operating Characteristic)* y el área bajo la curva (*AUC, Area Under the Curve*). Este enfoque permite visualizar y comparar la capacidad de discriminación del modelo para cada clase individual.

Primero, se binarizan las etiquetas de las clases tanto para el conjunto de entrenamiento como para el conjunto de prueba. La binarización transforma las etiquetas multicategoría en un formato binario necesario para calcular las curvas ROC. El número de clases se determina a partir del conjunto de datos binarizados.

Luego, se define un clasificador *OneVsRest* utilizando un *ensemble voting classifier*. Este clasificador ajusta el modelo al conjunto de entrenamiento y predice las probabilidades para el conjunto de prueba.

Para cada clase, se calcula la curva ROC y el área bajo la curva (*AUC*). La curva *ROC* se obtiene trazando la tasa de verdaderos positivos (*TPR*) contra la tasa de falsos positivos (*FPR*) en varios puntos de umbral. El *AUC* proporciona una medida agregada del rendimiento del modelo a través de todos los umbrales posibles.

Los nombres de las clases originales se recuperan y se utilizan para etiquetar las curvas *ROC* en la gráfica.

Finalmente, se representan en una gráfica las curvas *ROC* para cada clase en una sola figura. Cada curva se etiqueta con el nombre de la clase correspondiente y su correspondiente *AUC*, permitiendo una comparación visual del rendimiento del modelo entre diferentes clases. Se incluye una línea diagonal (representando una clasificación aleatoria) como referencia. La gráfica se configura con los límites adecuados para las tasas de falsos positivos y verdaderos positivos y se añaden etiquetas a los ejes y un título descriptivo. La leyenda se coloca en la esquina inferior derecha para facilitar la interpretación de la gráfica.

Este análisis ayuda a evaluar la efectividad del modelo en la clasificación de cada clase y proporciona una herramienta visual potente para identificar las áreas donde el modelo puede necesitar mejoras.

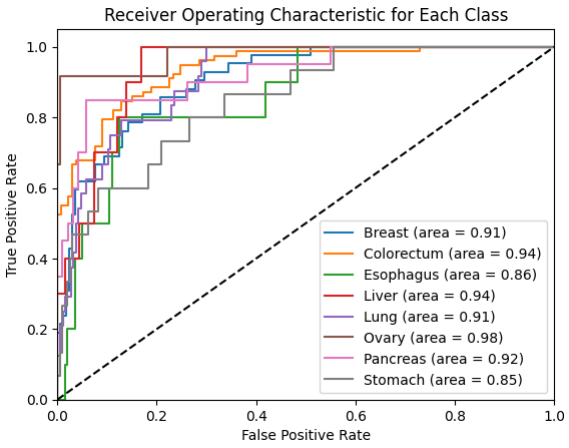

Fig. 19

5.2. Creación de AUC Promedio Global

En esta sección, se describen los pasos para evaluar un modelo de clasificación multiclase utilizando las curvas *ROC (Receiver Operating Characteristic)* y el área bajo la curva (*AUC, Area Under the Curve*) con validación cruzada de 10 pliegues. Este método permite una evaluación más robusta y generalizable del modelo.

Primero, se binarizan las etiquetas de las clases para todo el conjunto de datos. La binarización convierte las etiquetas multicategoría en un formato binario necesario para calcular las curvas ROC. El número de clases se determina a partir del conjunto de datos binarizados.

Luego, se define un *clasificador OneVsRest* utilizando un *ensemble voting classifier*. Este clasificador se entrenará y evaluará en cada pliegue de la validación cruzada.

Se realiza una validación cruzada estratificada de 10 pliegues, que divide el conjunto de datos en 10 partes, utilizando cada parte a su vez como conjunto de prueba y las restantes como conjunto de entrenamiento. En cada iteración, el modelo se entrena y se predicen las probabilidades para el conjunto de prueba.

Para cada clase, se calcula la curva *ROC* y el *AUC* en cada pliegue. Las tasas de verdaderos positivos (*TPR*) y las tasas de falsos positivos (*FPR*) se almacenan para cada clase y pliegue. Las curvas se interpolan para promediar las *TPR* a través de todos los pliegues, y se calcula el *AUC* medio y su desviación estándar.

Los nombres de las clases originales se recuperan y se utilizan para etiquetar las curvas *ROC* en la gráfica.

Finalmente, se grafican las curvas *ROC* promedio para cada clase en una sola figura. Cada curva se etiqueta con el nombre de la clase correspondiente y su *AUC* promedio, permitiendo una comparación visual del rendimiento del modelo entre diferentes clases. Se incluye una línea diagonal (representando una clasificación aleatoria) como referencia. La gráfica se configura con los límites adecuados para las tasas de falsos positivos y verdaderos positivos y se añaden etiquetas a los ejes y un título descriptivo. La leyenda se coloca en la esquina inferior derecha para facilitar la interpretación de la gráfica.

Además, se calcula el *AUC promedio global* ponderado por la cantidad de muestras de cada clase, proporcionando una medida agregada del rendimiento del modelo a través de todas las clases y reflejando la importancia relativa de cada clase en el conjunto de datos.

Este análisis ayuda a evaluar la efectividad del modelo en la clasificación de cada clase y proporciona una herramienta visual potente para identificar las áreas donde el modelo puede necesitar mejoras, al tiempo que asegura que la evaluación del modelo sea robusta y generalizable mediante el uso de validación cruzada.

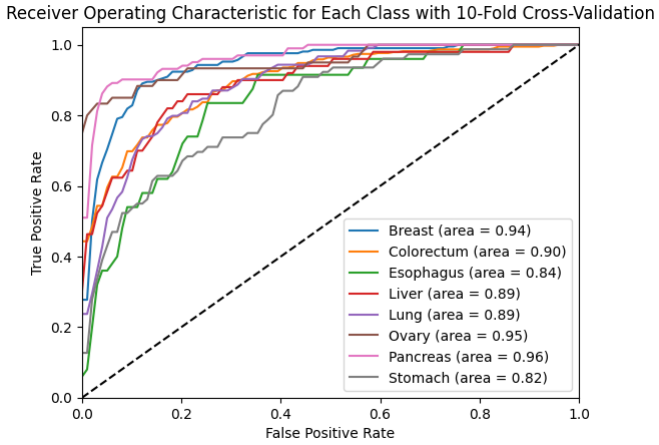

Fig. 20

6. Desarrollo de un Modelo Predictivo para el Diagnóstico de Tipos de Tumores

En esta fase, nos enfocamos en el desarrollo de un modelo predictivo avanzado que tiene como objetivo generar probabilidades precisas para el diagnóstico de diversos tipos de tumores presentes en nuestro conjunto de datos. Tal y como se mencionó en secciones anteriores, este componente del proyecto es fundamental para mejorar la precisión y eficacia en la detección y clasificación de tumores, utilizando información clínica y biomarcadores específicos.

- **Selección de Características Relevantes**

Para garantizar la efectividad del modelo predictivo, se llevó a cabo una cuidadosa selección de las características más relevantes del conjunto de datos. Se identificaron biomarcadores clave y variables clínicas, tales como niveles de diferentes proteínas y antígenos, que tienen un impacto significativo en la diferenciación de los tipos de tumores. Estas características fueron seleccionadas en base a su importancia clínica y su correlación con los diagnósticos de tumor, asegurando que el modelo se basara en datos con valor predictivo real.

- **Análisis de la Distribución de Tipos de Tumores**

Antes de proceder con la construcción del modelo, se realizó un análisis exhaustivo de la distribución de los tipos de tumores en el conjunto de datos. Este análisis incluyó la visualización de las frecuencias de cada tipo de tumor a través de gráficos de barras, lo que permitió identificar la prevalencia de cada categoría de tumor. Esta visualización no solo facilita la comprensión de la composición del conjunto de datos, sino que también es crucial para abordar cualquier desbalance en la distribución de clases que pueda afectar el rendimiento del modelo predictivo.

- **Clasificación de Tumores: Mayoritarios vs. Minoritarios**

Para manejar eficazmente la variabilidad en la prevalencia de los tipos de tumores, se implementó una clasificación adicional que distingue entre tumores mayoritarios y minoritarios. Utilizando un umbral predeterminado, se categorizan los tipos de tumor como mayoritarios si su frecuencia supera dicho umbral, y como minoritarios en caso contrario. Esta clasificación se incorpora en el conjunto de datos como una nueva variable, permitiendo al modelo ajustarse adecuadamente a las características específicas de cada grupo y mejorar la precisión de sus predicciones.

- **Implementación del Modelo Predictivo**

El desarrollo del modelo predictivo se basa en técnicas avanzadas de aprendizaje automático, específicamente diseñadas para manejar datos médicos complejos. El modelo no solo aprende a distinguir entre los diferentes tipos de tumores, sino que también calcula la probabilidad de cada diagnóstico posible, proporcionando una herramienta robusta y confiable para los profesionales médicos.

- **Conclusión**

Este capítulo detalla el enfoque metodológico y técnico adoptado para construir un modelo predictivo preciso y eficiente para el diagnóstico de tumores. A través de una cuidadosa selección de características, análisis de distribución y clasificación de tumores, y la implementación de técnicas avanzadas de aprendizaje automático, se busca mejorar significativamente la capacidad de diagnóstico, contribuyendo a un mejor manejo y tratamiento de los pacientes con cáncer.

6.1. Planteamiento jerárquico

	AFP (pg/ml)	Angiopoietin-2 (pg/ml)	AXL (pg/ml)	CA-125 (U/ml)	CA 15-3 (U/ml)	CA19-9 (U/ml)	CD44 (ng/ml)	CEA (pg/ml)	CYFRA 21-1 (pg/ml)	DKK1 (ng/ml)	...	sHER2/sEGFR2/sErbB2 (pg/ml)	sPECAM-1 (pg/ml)	TGFa (pg/ml)	Thr
0	-0.162730	-0.399967	0.161177	-0.155628	-0.208773	-0.126586	0.100087	-0.177265	-0.118556	-0.748740	...	0.613042	-0.739088	-0.190133	
1	-0.161972	-0.412345	-0.862979	-0.155856	-0.139035	-0.126858	-0.260427	-0.162648	-0.117902	-0.268988	...	-0.502477	-0.480916	-0.191864	
2	-0.155496	-0.507819	-0.284533	-0.155194	-0.255232	0.053126	-0.630079	-0.184200	-0.115457	0.867268	...	-0.710243	-0.461172	-0.182823	
3	-0.150629	-0.584907	-0.928337	-0.155787	-0.230770	-0.106049	0.047755	-0.181969	-0.011969	0.210765	...	-0.584788	-0.635555	-0.198597	
4	-0.147157	-0.464835	1.144186	-0.131803	0.317763	-0.085735	2.054673	-0.192859	-0.119066	-1.203243	...	0.827342	1.377542	-0.207830	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1000	-0.162529	-0.168476	-0.876422	-0.155536	-0.239253	-0.126601	-0.726438	-0.186948	-0.065832	-0.092237	...	-0.172675	-0.547599	-0.189364	

	AFP (pg/ml)	Angiopoietin-2 (pg/ml)	AXL (pg/ml)	CA-125 (U/ml)	CA 15-3 (U/ml)	CA19-9 (U/ml)	CD44 (ng/ml)	CEA (pg/ml)	CYFRA 21-1 (pg/ml)	DKK1 (ng/ml)	...	sHER2/sEGFR2/sErbB2 (pg/ml)	sPECAM-1 (pg/ml)	TGFa (pg/ml)	Thr
1001	-0.142995	-0.562578	-0.431811	-0.106853	-0.020372	-0.032200	0.319386	-0.146195	0.002598	-0.672990	...	0.559587	0.196013	0.507743	
1002	-0.148364	-0.353302	-0.605628	-0.155856	-0.237970	-0.126858	-0.399981	-0.201005	-0.117902	-1.051742	...	-0.210727	-0.548275	-0.191864	
1003	-0.161157	-0.206974	0.054145	-0.156015	-0.186974	-0.126786	-0.694872	-0.208696	-0.119163	-0.874991	...	-0.232628	-0.567034	-0.204752	
1004	-0.161577	-0.482293	-1.306079	-0.100463	-0.246552	-0.093165	-0.637555	-0.170552	-0.091464	-1.152742	...	-0.873070	-0.961481	-0.202700	

1005 rows × 43 columns

Fig. 21

Para comenzar con el desarrollo del modelo predictivo en esta fase del proyecto, vamos a utilizar el Dataframe *transformed_combined_dataframe*, que representamos en la figura *Fig. 21*.

La tabla muestra los valores estandarizados de varios biomarcadores en plasma, como AFP, Angiopoietin-2, y CA-125, entre otros, para 1005 muestras de pacientes. Además, incluye datos adicionales como el puntaje Omega, la etapa AJCC del cáncer, el sexo y el tipo de tumor. Cada fila representa una muestra de paciente con sus correspondientes valores de biomarcadores y características clínicas. La tabla tiene un total de 43 columnas, abarcando tanto los biomarcadores medidos en unidades específicas como información clínica de los pacientes.

Partiendo de este DataFrame, se genera un código que realiza un análisis de características importantes para la clasificación de tipos de tumores. Primero, selecciona un conjunto de biomarcadores relevantes y características clínicas, y los combina en un nuevo conjunto de datos. Luego, cuenta la frecuencia de cada tipo de tumor y genera un gráfico de barras que muestra la distribución de los tipos de tumor en el conjunto de datos. Finalmente, clasifica los tipos de tumor en mayoritarios y minoritarios, aplicando un umbral para distinguirlos. Esta clasificación se almacena en una nueva columna denominada *"Majority_Minority"*, donde los tipos de tumor que superan el umbral de frecuencia se marcan como mayoritarios (1) y los que no, como minoritarios (0).

Su resultado se refleja en la figura *Fig. 22*.

Para poder detectar los tipos de tumores específicos representados en la figura de más abajo, recuerdo el contenido del diccionario *tumor_mapping*:

Label	Tumor type
0	Breast
1	Colorectum
2	Esophagus
3	Liver
4	Lung
5	Ovary
6	Pancreas
7	Stomach

Fig. 22

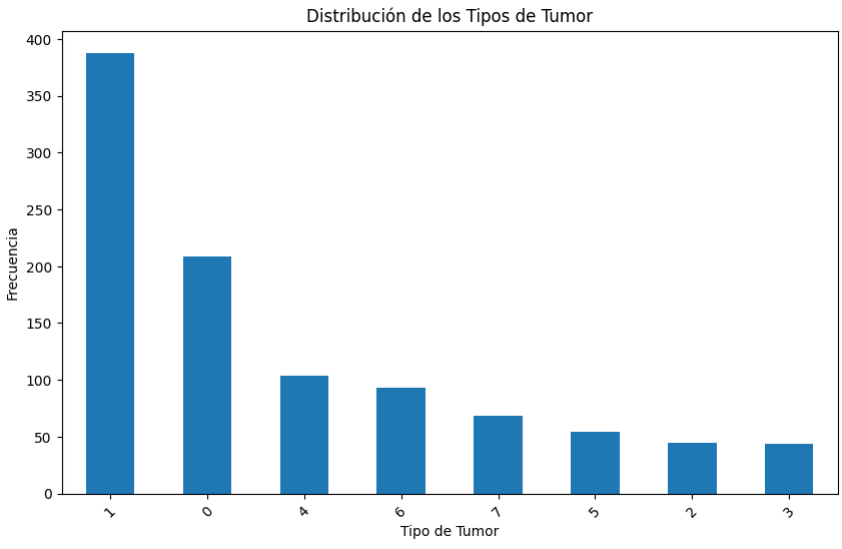

Fig. 23

6.2. Análisis y Modelado Predictivo de Tipos de Tumores: Clasificación Mayoritaria y Minoritaria

Vamos a empezar a generar el propio proceso de análisis y modelado predictivo orientado a la clasificación de tumores en categorías mayoritarias y minoritarias, utilizando técnicas avanzadas de aprendizaje automático.

6.2.1. PASO 1

1. **Carga y Preprocesamiento de Datos**

Inicialmente, se cargan los datos del Dataframe *df_final*, que contiene información detallada sobre diversos biomarcadores y características clínicas de los pacientes. Posteriormente, se analizan las frecuencias de los distintos tipos de tumores presentes en el conjunto de datos, identificando así la distribución de cada tipo de tumor.

2. **División en Características y Objetivo**

Para el proceso de modelado, se divide el conjunto de datos en dos partes: las características independientes y la variable objetivo. La variable objetivo en este caso es una nueva columna denominada "*Majority_Minority*", la cual categoriza los tumores en mayoritarios y minoritarios en función de un umbral de frecuencia previamente definido.

3. **Entrenamiento y Evaluación del Modelo Inicial**

Se procede a dividir los datos en conjuntos de entrenamiento y prueba, y se entrena un modelo de *Random Forest* para predecir la categorización de los tumores como mayoritarios o minoritarios. Este modelo se evalua utilizando un reporte de clasificación, el cual proporciona métricas detalladas de *precision, recall y f1-score* para ambas categorías.

4. **Modelado Específico para Clases Mayoritarias y Minoritarias**

Para abordar las diferencias entre los tumores mayoritarios y minoritarios, se implementan dos modelos de *Random Forest* separados. Los datos se divididen en dos subconjuntos: uno para los tumores mayoritarios y otro para los minoritarios. Cada subconjunto se utiliza para entrenar y evaluar un modelo específico, permitiendo así una mayor precisión en la predicción de cada tipo de tumor dentro de sus respectivas categorías.

5. **Evaluación de Modelos y Reportes de Clasificación**

Los modelos específicos para tumores mayoritarios y minoritarios se evaluan por separado, generando reportes de clasificación que detallan el rendimiento del modelo en términos de *precision, recall y f1-score* para cada tipo de tumor dentro de las categorías mayoritarias y minoritarias. Los resultados de estos reportes proporcionan una visión clara de la efectividad de los modelos en la clasificación precisa de los distintos tipos de tumores, resultado representado en la figura *Fig. 24*.

Dicho resultado, muestra un rendimiento sólido en la clasificación de tumores mayoritarios, con una precisión global del 79% y f1-scores altos en cáncer de mama (0.88) y colorectal (0.79). Sin embargo, el rendimiento es menor para los tumores de pulmón (f1-score de 0.48). Para las clases minoritarias, el modelo también tiene un rendimiento aceptable con una precisión global del 72%, destacando especialmente en la clasificación de esófago (f1-score de 1.00), aunque el rendimiento es más variable para otros tipos de tumores minoritarios.

Reporte de clasificación para clases mayoritarias:

	precision	recall	f1-score	support
Colorectum	0.76	0.81	0.79	43
Breast	0.85	0.91	0.88	80
Lung	0.59	0.40	0.48	25
Pancreas	0.80	0.73	0.76	11
accuracy			0.79	
macro avg	0.75	0.71	0.73	159
weighted avg	0.78	0.79	0.78	159

Reporte de clasificación para clases minoritarias:

	precision	recall	f1-score	support
Stomach	0.57	0.50	0.53	8
Ovary	0.58	0.78	0.67	9
Esophagus	1.00	1.00	1.00	11
Liver	0.69	0.60	0.64	15
accuracy			0.72	
macro avg	0.71	0.72	0.71	43
weighted avg	0.73	0.72	0.72	43

Fig. 24

Notas

Este enfoque de modelado predictivo, que incluye la clasificación de tumores en mayoritarios y minoritarios, así como el entrenamiento de modelos específicos para cada categoría, permite una mejor comprensión y predicción de los diferentes tipos de tumores. Este método mejora significativamente la precisión del diagnóstico, facilitando así una toma de decisiones más informada y eficaz en el manejo y tratamiento de los pacientes con cáncer.

6.2.2. PASO 2

1. **Carga y Preprocesamiento de Datos**

Inicialmente, se cargan los datos del archivo *transformed_combined_dataframe.xlsx*, el cual contiene información detallada sobre diversos biomarcadores y características clínicas de los pacientes. Se mapean los valores enteros a sus correspondientes tipos de tumor y se analiza la frecuencia de cada tipo de tumor para entender la distribución en el conjunto de datos. A partir de estas frecuencias, se clasifica cada tumor como mayoritario o minoritario utilizando un umbral definido.

2. **División en Características y Objetivo**

El siguiente paso consiste en dividir el conjunto de datos en dos partes: las características independientes (X) y la variable objetivo (y). La variable objetivo, "*Majority_Minority*", categoriza los tumores en mayoritarios y minoritarios según su frecuencia en el conjunto de datos.

3. **Entrenamiento y Evaluación del Modelo Inicial**

Se procede a dividir los datos en conjuntos de entrenamiento y prueba. Posteriormente, se entrena un modelo de *Random Forest* para predecir si un tumor pertenece a la categoría mayoritaria o minoritaria. La evaluación de este modelo se realiza utilizando un reporte de clasificación, que proporciona métricas detalladas de precisión, recall y f1-score para ambas categorías.

4. **Modelado Específico para Clases Mayoritarias y Minoritarias**

Para mejorar la precisión del análisis, se implementan dos modelos de *Random Forest* separados para los tumores mayoritarios y minoritarios. Los datos se dividen en subconjuntos específicos para cada categoría: uno para los tumores mayoritarios y otro para los minoritarios. Cada subconjunto se utiliza para entrenar y evaluar un modelo específico, permitiendo una mayor precisión en la predicción de cada tipo de tumor dentro de sus respectivas categorías.

5. **Evaluación de Modelos y Reportes de Clasificación**

Los modelos específicos para tumores mayoritarios y minoritarios se evalúan de manera independiente, generando reportes de clasificación que detallan el rendimiento del modelo en términos de precisión, recall y f1-score para cada tipo de tumor dentro de las categorías mayoritarias y minoritarias. Estos reportes proporcionan una visión clara de la efectividad de los modelos en la clasificación precisa de los distintos tipos de tumores, como se refleja en las métricas obtenidas y representadas en las figuras correspondientes.

6. **Análisis de Resultados del Modelo de Clasificación de Tumores**

Reporte de clasificación para mayoritarias vs minoritarias:

	precision	recall	f1-score	support
Minoritaria	0.68	0.46	0.55	41
Mayoritaria	0.87	0.94	0.91	160
accuracy		0.85		
macro avg	0.78	0.70	0.73	201
weighted avg	0.83	0.85	0.83	201

Distribución de clases en majority_data antes y después de la subdivisión:

Tumor type	count
Antes de la subdivisión	
1	381
0	209
4	102
6	93
2	10
7	5
3	4
5	3
Después de la subdivisión	
1	590
0	195
-1	22

Reporte de clasificación para mayoritarias subdivididas:

	precision	recall	f1-score	support
Minoritaria	0.90	0.68	0.77	40
Mayoritaria	0.90	0.97	0.93	117
accuracy		0.90		
macro avg	0.90	0.82	0.85	157
weighted avg	0.90	0.90	0.89	157

Distribución de clases en minority_data después de la subdivisión:

Majority_Minority_Sub	count
1	114
0	75
-1	9

Reporte de clasificación para minoritarias subdivididas:

	precision	recall	f1-score	support
Minoritaria	0.89	0.44	0.59	18
Mayoritaria	0.66	0.95	0.78	20
accuracy		0.71		
macro avg	0.77	0.70	0.68	38
weighted avg	0.77	0.71	0.69	38

Fig. 25

Los resultados del análisis, que podemos observar en la tabla *Fig. 25*, muestran un rendimiento diferenciado entre las categorías mayoritarias y minoritarias de tumores, evaluado a través de métricas clave como precision, recall y f1-score.

- **Resultados de Clasificación para Mayoritarias vs Minoritarias (Modelo Inicial)** : El modelo inicial de clasificación de tumores en categorías mayoritarias y minoritarias muestra un rendimiento sólido con una precisión promedio del 83% y un f1-score promedio del 83%.
  - **Categoría Minoritaria**: El modelo alcanza un precision del 68% y un recall del 46%, indicando que identifica correctamente el 68% de las predicciones positivas, pero solo el 46% de los casos reales de tumores minoritarios.
  - **Categoría Mayoritaria**: El modelo muestra un precision del 87% y un recall del 94%, demostrando una alta precisión y capacidad para identificar la mayoría de los casos reales de tumores mayoritarios.
- **Distribución y Subdivisión de Clases Mayoritarias** : Antes de la subdivisión basada en la predicción del modelo, la distribución de tumores en la categoría mayoritaria revela un desequilibrio significativo con varios tipos predominantes como 1, 0, 4 y 6. Después de la subdivisión y corrección, se observa una mejora en la distribución de clases con una clara asignación de tumores a categorías mayoritarias y minoritarias dentro de los tumores mayoritarios.
- **Resultados para Mayoritarias Subdivididas** : Tras la subdivisión y entrenamiento de modelos específicos para cada categoría de tumores mayoritarios:
  - **Categoría Minoritaria**: La precisión mejora a 90% y el recall a 68%, indicando una notable mejora en la capacidad del modelo para identificar correctamente los tumores de esta clase.
  - **Categoría Mayoritaria**: Se mantiene una alta precisión del 90% y un recall del 97%, demostrando consistencia en la identificación adecuada de tumores mayoritarios.
- **Resultados para Minoritarias Subdivididas** : Para los tumores minoritarios, tras la subdivisión y entrenamiento de modelos específicos:
  - La categoría **Mayoritaria** muestra una precisión del 66% y un recall del 95%, indicando una buena capacidad para identificar correctamente los tumores minoritarios más frecuentes.
  - Sin embargo, la categoría **Minoritaria** dentro de los tumores minoritarios exhibe una precisión del 89% pero un recall del 44%, sugiriendo oportunidades de mejora en la identificación de todos los casos reales de tumores minoritarios.

Notas

El modelo exhibe un rendimiento sólido en la clasificación de tumores mayoritarios y minoritarios, con resultados positivos en la mayoría de las categorías evaluadas. Sin embargo, la variabilidad en precisión y recall entre estas categorías sugiere áreas potenciales para mejoras futuras, especialmente en la identificación precisa de tumores minoritarios.

Para abordar esta variabilidad, hemos implementado la subdivisión y entrenamiento de modelos específicos para cada subconjunto de datos (mayoritarios y minoritarios). Este enfoque está diseñado para mejorar la precisión y el rendimiento del modelo general. Es particularmente beneficioso en escenarios donde existe un desequilibrio significativo entre las clases, asegurando que el modelo no esté sesgado hacia las clases mayoritarias y pueda manejar con precisión los casos minoritarios.

6.3. Evaluación

6.3.1. Descripción del Flujo de Trabajo para la Clasificación de Tumores

Para este flujo de trabajo se han utilizado técnicas de aprendizaje automático para clasificar tumores en categorías mayoritarias y minoritarias, optimizando la precisión mediante la subdivisión y entrenamiento de modelos específicos para cada subconjunto de datos.

1. **Carga y Preprocesamiento de Datos**
   - Se carga el conjunto de datos del DataFrame *transformed_combined_dataframe* que contiene información sobre tipos de tumores.
   - Se realiza un conteo inicial de la cantidad de cada tipo de tumor.
2. **Definición de Clasificación Mayoritaria/Minoritaria**
   - Se añade una columna que etiqueta cada tumor como mayoritario o minoritario, basándose en un umbral predefinido.
3. **Entrenamiento del Modelo Inicial**
   - Se divide el conjunto de datos en características (X) y objetivo (y).
   - Se aplica una división de entrenamiento y prueba para evaluar el desempeño del modelo de Bosques Aleatorios en la predicción de tumores mayoritarios vs minoritarios.
4. **Evaluación del Modelo Inicial**
   - Se genera un reporte detallado de clasificación que incluye precisiones, recalls y f1-scores para cada clase (Mayoritaria y Minoritaria).
5. **Subdivisión y Entrenamiento de Modelos Específicos**
   - Se subdividen los datos según las predicciones del modelo inicial.
   - Se corrige la lógica de subdivisión para optimizar la precisión de los modelos:
     - Para las clases mayoritarias subdivididas, se entrena un modelo específico para predecir subcategorías dentro de las mayoritarias.
     - Para las clases minoritarias subdivididas, se entrena otro modelo específico para predecir subcategorías dentro de las minoritarias.
6. **Evaluación de Modelos Subdivididos**
   - Se generan reportes separados para evaluar la precisión de los modelos en la predicción de subcategorías dentro de las clases mayoritarias y minoritarias.

7. **Predicciones Finales y Evaluación del Modelo Completo**
    - Se realiza una evaluación final combinando las predicciones de los modelos de clasificación mayoritaria y minoritaria.
    - Se ajusta el reporte de clasificación final para incluir todas las clases presentes en las predicciones.
8. **Resultados y Análisis**

	precision	recall	f1-score	support
Minoritaria	0.68	0.46	0.55	41
Mayoritaria	0.87	0.94	0.91	160
accuracy	0.85			
macro avg	0.78	0.70	0.73	201
weighted avg	0.83	0.85	0.83	201

Distribución de clases en majority_data antes de la subdivisión

Clase	Count
1	381
0	209
4	102
6	93
2	10
7	5
3	4
5	3

Distribución de clases en majority_data después de la subdivisión

Clase	Count
1	590
0	195

	precision	recall	f1-score	support
Minoritaria	0.90	0.68	0.77	40
Mayoritaria	0.90	0.97	0.93	117
accuracy	0.90			
macro avg	0.90	0.82	0.85	157
weighted avg	0.90	0.90	0.89	157

Distribución de clases en minority_data después de la subdivisión

Clase	Count
1	114
0	75

	precision	recall	f1-score	support
Minoritaria	0.89	0.44	0.59	18
Mayoritaria	0.66	0.95	0.78	20
accuracy	0.71			
macro avg	0.77	0.70	0.68	38
weighted avg	0.77	0.71	0.69	38

	precision	recall	f1-score	support
Breast	0.03	0.02	0.03	41
Colorectum	0.85	0.74	0.79	160
Esophagus	0.00	0.00	0.00	0
Stomach	0.00	0.00	0.00	0
accuracy	0.59			

macro avg	0.22	0.19	0.20	201
weighted avg	0.68	0.59	0.63	201

Fig. 26

En la figura Fig. 26, podemos ver los resultados del modelo que acabamos de explicar. Por ello, mediante la visualización de la tabla podemos determinar que el modelo muestra un buen rendimiento en la clasificación de tumores mayoritarios y minoritarios, con precisión y recall variados entre ambas categorías. Se destaca la necesidad de mejorar la precisión en la identificación de tumores minoritarios.

- **Mayoritarias**
  - **Antes de la Subdivisión**: Se observa un desequilibrio en la distribución de clases.
  - **Después de la Subdivisión**: Mejora significativa en la distribución y precisión del modelo, especialmente en la subcategorización.
- **Minoritarias**
  - **Subdivididas**: El modelo muestra precisión y recall mejorados, aunque se identifica una necesidad de optimización en la precisión de la clasificación.
- **Conclusión**

Los resultados indican un rendimiento sólido en la clasificación de tumores, especialmente tras la subdivisión y entrenamiento de modelos específicos. Sin embargo, la precisión en la identificación de tumores minoritarios puede mejorarse mediante técnicas avanzadas, ya que somos plenos conocedores de las posibilidades de mejora del proyecto.

6.3.2. Generación y Análisis del Modelo de Clasificación de Tumores con Random Forest

Este código utiliza el algoritmo *Random Forest*, implementado desde la biblioteca *scikit-learn*, para clasificar tipos de tumores en categorías mayoritarias y minoritarias. Aquí se describe el flujo de trabajo:

1. **Carga y Preprocesamiento de Datos:**
   - Se carga el conjunto de datos desde el archivo `transformed_combined_dataframe.xlsx` utilizando pandas.
   - Se crea un mapeo para convertir valores enteros de `Tumor type` a tipos de tumor específicos como Colorectum, Breast, Lung, etc.
   - Se cuenta la frecuencia de cada tipo de tumor en el conjunto de datos.
2. **Preparación de Datos:**
   - Se añade una columna `Majority_Minority` que clasifica cada registro como mayoritario (1) o minoritario (0) basado en un umbral de frecuencia.
3. **División de Datos:**
   - Se separan las características (`X`) y el objetivo (`y`) del dataset para entrenamiento y prueba usando `train_test_split`. El 20% de los datos se reservan para prueba.
4. **Entrenamiento del Modelo:**
   - Se entrena un modelo de Random Forest (`best_rf_model`) con parámetros ajustados manualmente (`n_estimators=200, max_depth=20, min_samples_split=5, min_samples_leaf=2, bootstrap=True, random_state=42`).
   - El modelo se ajusta usando los datos de entrenamiento (`X_train` y `y_train`).
5. **Predicción y Evaluación:**
   - Se realizan predicciones sobre el conjunto de prueba (`X_test`) utilizando el modelo entrenado.
   - Se genera un reporte detallado de clasificación utilizando `classification_report`, evaluando precisión, recall y f1-score para las clases 'Minoritaria' y 'Mayoritaria'.
6. **Subdivisión y Almacenamiento de Resultados:**
   - Se predice la clasificación (`Predicted_MM`) para todo el conjunto de datos y se guarda en un archivo CSV llamado `subdivided_data.csv` para su uso posterior.

Mediante la ejecución de este script, no solo se entrena un modelo de Random Forest para clasificación binaria de tumores, sino que también se proporciona métricas detalladas y guarda los resultados subdivididos para análisis adicionales o implementaciones posteriores.

	precision	recall	f1-score	support
Minoritaria	0.69	0.44	0.54	41
Mayoritaria	0.87	0.95	0.91	160
accuracy	0.85			
macro avg	0.78	0.69	0.72	201
weighted avg	0.83	0.85	0.83	201

Fig. 27

Este reporte de clasificación muestra el desempeño de un modelo aplicado a datos de tumores. Para la categoría minoritaria, el modelo alcanza una precisión del 69%, lo cual es aceptable, aunque el recall del 44% indica que hay una cantidad significativa de falsos negativos, lo cual podría ser problemático en contextos donde identificar correctamente todos los casos es crucial. El f1-score de 0.54 también refleja un balance moderado entre precisión y recall para esta clase.

En contraste, el modelo muestra un rendimiento sólido para la categoría mayoritaria, con una alta precisión del 87% y un recall del 95%. Estas métricas sugieren que el modelo es efectivo para identificar correctamente los casos de esta categoría. El f1-score elevado de 0.91 confirma esta observación, indicando un buen equilibrio entre precisión y recall.

La exactitud global del modelo es del 85%, evaluada sobre un total de 201 muestras. Esta métrica global es moderadamente buena, pero es importante considerar que está influenciada por el desbalance de clases en el conjunto de datos.

En resumen, mientras que el modelo tiene un desempeño fuerte en la clasificación de la categoría mayoritaria, podría beneficiarse de mejoras para mejorar la precisión y el recall en la categoría minoritaria.

6.3.3. Proceso de Entrenamiento y Evaluación de Modelos

En este análisis de datos sobre tumores, se emplearon diversas técnicas de aprendizaje automático para clasificar entre categorías mayoritarias y minoritarias. A continuación se detalla el proceso seguido:

1. **Carga de Datos y Filtrado:** Se cargaron los datos del DataFrame *subdivided_data* desde un archivo CSV. Estos datos fueron filtrados en dos conjuntos distintos: mayoritarios y minoritarios, según las predicciones anteriores.
2. **Entrenamiento de Modelos:**
   - **Modelo para Tumores Mayoritarios:** Se aplicó un modelo de Random Forest ajustado manualmente para predecir entre tumores mayoritarios, obteniendo métricas de precisión y recall significativas.
   - **Modelo para Tumores Minoritarios:** Se empleó otro modelo de Random Forest para clasificar tumores minoritarios, optimizando sus parámetros para mejorar el rendimiento.
3. **Evaluación de Modelos:**
   - **Subconjuntos Mayoritarios y Minoritarios:** Cada modelo fue evaluado por separado utilizando las métricas de precision, recall y f1-score para las categorías "Minoritaria" y "Mayoritaria".
   - **Modelo Final Integrado:** Se implementó una lógica combinada para evaluar el modelo final, que clasifica los tumores según la categoría final determinada por los modelos anteriores.
4. **Reporte Final:** Se ha producido un informe detallado de clasificación que resume el rendimiento global del modelo final en la clasificación de tumores, abarcando todas las categorías y métricas pertinentes. Sin embargo, los resultados presentan cierto sesgo, lo que indica la necesidad de continuar trabajando en ellos para alcanzar resultados óptimos.

6.3.4. Proceso de Ajuste de Hiperparámetros y Generación de Datos Sintéticos

En este análisis de datos sobre tumores, logramos mejorar los resultados previos mediante la implementación de *CTGAN*, una técnica avanzada de aprendizaje automático que permite la generación de datos sintéticos. A continuación, se detalla el proceso seguido:

1. **Ajuste de Hiperparámetros con Grid Search:** Se utilizó un modelo de *Random Forest* para clasificar tumores en categorías mayoritarias y minoritarias. Se realizaron pruebas exhaustivas de hiperparámetros utilizando Grid Search para optimizar el rendimiento del modelo, evaluando métricas como *precision, recall y f1-score*.
2. **Mejora del Modelo con CTGAN:** Se identificaron clases con menor rendimiento del modelo inicial y se empleó *CTGAN*, una técnica de generación de datos sintéticos, para aumentar el conjunto de datos. Esto permitió mejorar la representación de las clases minoritarias y optimizar el modelo de clasificación.
3. **Entrenamiento y Evaluación del Modelo Mejorado:**
   - **Entrenamiento con Datos Aumentados:** El modelo de *Random Forest* fue reentrenado con los datos originales combinados con datos sintéticos generados por *CTGAN*.
   - **Evaluación del Modelo Mejorado:** Se evaluó el modelo final utilizando las métricas de clasificación para las categorías "Minoritaria" y "Mayoritaria", mostrando el impacto positivo del aumento de datos en el rendimiento general.
4. **Análisis de Resultados del Proceso de Ajuste de Hiperparámetros y Generación de Datos Sintéticos**

	precision	recall	f1-score	support
Minoritaria	0.90	0.95	0.92	270
Mayoritaria	0.88	0.79	0.83	131
accuracy	0.90			
macro avg	0.89	0.87	0.88	401
weighted avg	0.89	0.90	0.89	401

Fig. 28

El reporte de clasificación de la tabla *Fig. 28* revela el desempeño del modelo final después de aplicar técnicas avanzadas de generación de datos. Observamos que el modelo logra una precisión global del 90%, con una precisión del 90% para la clase minoritaria (tumores menos frecuentes) y del 88% para la clase mayoritaria (tumores más frecuentes). Esto indica una capacidad notable para identificar correctamente tanto los tumores minoritarios como los mayoritarios en el conjunto de datos evaluado.

El recall muestra que el modelo es especialmente eficaz en la identificación de la clase minoritaria, alcanzando un valor del 95%, lo cual es crucial en aplicaciones médicas donde la detección precisa de tumores menos comunes es fundamental. Sin embargo, el recall para la clase mayoritaria es del 79%, lo que indica que el modelo podría mejorar en la capacidad de identificar correctamente todos los casos de tumores más frecuentes.

El f1-score, que combina precisión y recall, muestra un desempeño equilibrado con valores del 92% para la clase minoritaria y del 83% para la clase mayoritaria. Este equilibrio sugiere que el modelo está logrando un buen balance entre la precisión y la capacidad de recuperación de ambas clases de tumores.

Notas

Este enfoque integrado permite una clasificación más precisa y robusta de tumores, mejorando la capacidad de detección y diagnóstico en el ámbito médico.

6.3.4.1. Mayoritaria: Lung, Pancreas VS. Minoritaria: Stomach, Ovary, Esophagus, Liver

Este código utiliza el algoritmo *Random Forest* de la biblioteca *scikit-learn* para clasificar tipos de tumores en categorías mayoritarias y minoritarias. Aquí se describe el flujo de trabajo:

1. **Filtrado de Datos Minoritarios:**
   - Se seleccionan los datos minoritarios del conjunto aumentado (`augmented_data`) donde `Majority_Minority` es igual a 0.
2. **Clasificación de Tumores:**
   - Se aplica una función (`classify_minority_majority`) para clasificar los tumores en dos categorías: Lung y Pancreas como mayoritarios (1) y Stomach, Ovary, Esophagus, Liver como minoritarios (0).
3. **Preparación de Datos:**
   - Se separan las características (`X_minority`) y el objetivo (`y_minority`) del dataset filtrado para entrenamiento y prueba utilizando `train_test_split`. El 20% de los datos se reservan para prueba.

4. **Entrenamiento del Modelo:**
   - Se inicializa un modelo de Random Forest (`rf_model_m`) con parámetros predeterminados.
   - Se realiza una búsqueda de hiperparámetros usando `GridSearchCV` para encontrar el mejor modelo (`best_rf_model_m`) basado en la validación cruzada con 3 folds.
5. **Predicción y Evaluación:**
   - Se realizan predicciones sobre el conjunto de prueba (`X_minority_test`) utilizando el mejor modelo encontrado.
   - Se genera un reporte detallado de clasificación (`classification_report`) evaluando precisión, recall y f1-score para las clases 'Minoritaria' y 'Mayoritaria'.
6. **Generación de Resultados:**
   - Se calcula la probabilidad de predicción para el conjunto de prueba usando `proba_predict_results` con el mejor modelo de Random Forest, enfocado en clasificar entre Lung, Pancreas vs Stomach, Ovary, Esophagus, Liver.

Notas

Este proceso no solo proporciona una clasificación precisa de tumores en categorías mayoritarias y minoritarias, sino que también optimiza el modelo de Random Forest mediante ajuste de hiperparámetros, garantizando resultados robustos y aplicables en análisis de datos y ciencia de datos.

7. **Resultados**

	Predicted_Proba_Class_0	Predicted_Proba_Class_1	Actual	Model	Set	Predicted
0	0.267698	0.732302	NaN	Random Forest - Colorectum vs Breast	Test Set	Colorectum
1	0.195118	0.804882	NaN	Random Forest - Colorectum vs Breast	Test Set	Colorectum
2	0.052917	0.947083	NaN	Random Forest - Colorectum vs Breast	Test Set	Colorectum
3	0.921773	0.078227	Colorectum	Random Forest - Colorectum vs Breast	Test Set	Breast
4	0.892563	0.107437	NaN	Random Forest - Colorectum vs Breast	Test Set	Breast
...	...	...	...	...	...	...
397	0.722024	0.277976	NaN	Random Forest - Lung, Pancreas vs Stomach, Ova...	Test Set	Breast
398	0.585488	0.414512	Colorectum	Random Forest - Lung, Pancreas vs Stomach, Ova...	Test Set	Breast
399	0.327746	0.672254	NaN	Random Forest - Lung, Pancreas vs Stomach, Ova...	Test Set	Colorectum
400	0.420960	0.579040	NaN	Random Forest - Lung, Pancreas vs Stomach, Ova...	Test Set	Colorectum
401	0.658440	0.341560	NaN	Random Forest - Lung, Pancreas vs Stomach, Ova...	Test Set	Breast

402 rows × 6 columns

Fig. 29

Los resultados de la tabla proporcionada revelan cómo el modelo de Random Forest clasificó diferentes tipos de tumores en dos categorías principales: Colorectum vs Breast y Lung, Pancreas vs Stomach, Ovary, Esophagus, Liver. Aquí están los puntos clave:

1. **Modelo y Conjunto de Datos:**
   - Se utilizaron dos modelos diferentes de Random Forest para clasificar los tumores en conjuntos específicos: Colorectum vs Breast y Lung, Pancreas vs Stomach, Ovary, Esophagus, Liver.
   - Los datos fueron evaluados en un conjunto de prueba.
2. **Predicciones y Probabilidades:**
   - Para cada modelo y conjunto de datos, se calcularon las probabilidades predichas (`Predicted_Proba_Class_0` y `Predicted_Proba_Class_1`) para cada clase de tumor.
   - Se observa una variabilidad considerable en las probabilidades predichas entre las clases, reflejando la incertidumbre o la confianza del modelo en sus predicciones.
3. **Comparación con Resultados Actuales:**
   - Se compararon las predicciones del modelo (`Predicted`) con los resultados actuales (`Actual`), revelando cómo el modelo interpretó y clasificó correctamente los tipos de tumores en la mayoría de los casos.
   - Se identificaron casos donde las predicciones del modelo difieren de los resultados reales, lo cual podría indicar áreas donde el modelo necesita mejorar o donde hay desafíos en la clasificación.
4. **Evaluación del Rendimiento:**
   - A través de las métricas de evaluación como precisión, recall y f1-score, se puede evaluar el rendimiento del modelo en términos de su capacidad para distinguir entre las clases de tumores.
   - Es crucial analizar las métricas de rendimiento para cada clase y considerar cualquier desequilibrio en los resultados que pueda afectar la utilidad clínica o investigativa del modelo.

6.3.4.2. Búsqueda de Hiperparámetros para Lung vs Pancreas

Este código utiliza el algoritmo *Random Forest* de la biblioteca *scikit-learn* para clasificar tipos de tumores entre Lung y Pancreas. Aquí se describe el flujo de trabajo:

1. **Filtrado de Datos Minoritarios Mayoritarios:**
   - Se seleccionan los datos del conjunto aumentado (`augmented_data`) donde `Majority_Minority` es igual a 0 y el tipo de tumor es Lung (4) o Pancreas (6).
2. **Clasificación de Tumores:**
   - Se aplica una función (`classify_lung_pancreas`) para clasificar los tumores entre Lung y Pancreas, asignando 1 a Lung y 0 a Pancreas.

3. **Preparación de Datos:**
    - Se separan las características (`X_lung_pancreas`) y el objetivo (`y_lung_pancreas`) del dataset filtrado para entrenamiento y prueba utilizando `train_test_split`. El 20% de los datos se reservan para prueba.
4. **Entrenamiento del Modelo:**
    - Se inicializa un modelo de Random Forest (`rf_model_lp`) con parámetros predeterminados.
    - Se realiza una búsqueda de hiperparámetros usando `GridSearchCV` para encontrar el mejor modelo (`best_rf_model_lp`) basado en la validación cruzada con 3 folds.
5. **Predicción y Evaluación:**
    - Se realizan predicciones sobre el conjunto de prueba (`X_lp_test`) utilizando el mejor modelo encontrado.
    - Se genera un reporte detallado de clasificación (`classification_report`) evaluando precisión, recall y f1-score para las clases 'Pancreas' y 'Lung'.
6. **Generación de Resultados:**
    - Se calcula la probabilidad de predicción para el conjunto de prueba usando `proba_predict_results` con el mejor modelo de Random Forest, enfocado en clasificar entre Lung y Pancreas.
7. **Análisis de Resultados del Proceso de Búsqueda de Hiperparámetros para Lung vs Pancreas**

Fig. 30

7. Referencias:

1. **Cuantificación de ADN libre en plasma sanguíneo de voluntarios sanos en una población bogotana - Salazar-Jordan, Revista Nova, 2009**
2. **Onco-proteogenomics: cancer proteomics joins forces with genomics - Alfaro et all, 2014**
3. **On protein synthesis - Crick, 1958**
4. **Community Assessment of the Predictability of Cancer Protein and Phosphoprotein Levels from Genomics and Transcriptomics - Mi Yang et all, 2020**
5. **Crowdsourcing biomedical research: leveraging communities as innovation engines - Saez-Rodriguez et al., 2016**
6. **Fine-mapping of 150 breast cancer risk regions identifies 191 likely target genes - Wu et all**