El segundo código que tenemos es un script de Python que realiza varias tareas relacionadas con el procesamiento de imágenes médicas y la construcción y evaluación de modelos de aprendizaje automático para la clasificación de imágenes en dos categorías:
- ALL (leucemia linfoblástica aguda).
- HEM (células sanguíneas normales).

Las principales partes del código:

Importación de bibliotecas:

Se importan las bibliotecas necesarias, como NumPy, Pandas, OpenCV (cv2), Matplotlib, Seaborn y las herramientas de aprendizaje automático de TensorFlow y scikit-learn.

Numpy:
Numpy es una librería fundamental de Python para la computación numérica y científica. Proporciona soporte para manejar grandes arreglos y matrices multidimensionales, así como una colección de funciones matemáticas para operar eficientemente en estos arreglos. Numpy es una herramienta crucial en el campo del aprendizaje automático, incluyendo tareas de clasificación de imágenes, y su documentación principal, que se encuentra en https://numpy.org/doc/stable/, describe su importancia y su uso

Pandas:

Pandas es una biblioteca de código abierto en Python que proporciona estructuras de datos y herramientas de análisis de datos para trabajar con datos estructurados. Es ampliamente utilizado en tareas de manipulación, análisis y preparación de datos, lo que lo convierte en una herramienta esencial para científicos de datos y profesionales de aprendizaje automático. Pandas ofrece una forma poderosa y flexible de manejar datos y desempeña un papel crucial en el aprendizaje automático, incluida la clasificación de imágenes.

OpenCV:

OpenCV, que significa Biblioteca de Visión por Computadora de Código Abierto, es una biblioteca de software de visión por computadora y aprendizaje automático de código abierto. Proporciona una amplia gama de herramientas y funciones que permiten a desarrolladores e investigadores trabajar con datos visuales y realizar diversas tareas de visión por computadora, como procesamiento de imágenes y videos, detección de objetos, clasificación de imágenes y más. OpenCV se utiliza ampliamente tanto en entornos académicos como industriales para una variedad de aplicaciones, incluido el análisis de imágenes y videos, la robótica y el aprendizaje automático.

Matplotlib and seaborn:
Matplotlib y Seaborn son dos bibliotecas populares de Python utilizadas para la visualización de datos y la creación de representaciones gráficas de datos. Estas bibliotecas desempeñan un papel fundamental en el aprendizaje automático y en las tareas de

clasificación de imágenes al proporcionar los medios para visualizar datos, explorar patrones y presentar resultados de manera efectiva.

Matplotlib:

Matplotlib es una biblioteca integral de visualización de datos en Python. Se utiliza ampliamente para crear varios tipos de gráficos, incluyendo gráficos de líneas, gráficos de barras, gráficos de dispersión, histogramas y más. El objetivo principal de Matplotlib es ayudar a los usuarios a visualizar sus datos y comprender los patrones y relaciones subyacentes.
Seaborn:
Seaborn:

Seaborn es una biblioteca de visualización de datos en Python construida sobre Matplotlib. Proporciona una interfaz de alto nivel para crear gráficos estadísticos informativos y atractivos. Seaborn simplifica la creación de visualizaciones complejas y se especializa en la visualización de datos estadísticos.

Tensorflow:
TensorFlow es un marco de aprendizaje automático de código abierto desarrollado por Google. Está diseñado para proporcionar una plataforma flexible y eficiente para construir y desplegar modelos de aprendizaje automático. TensorFlow se utiliza ampliamente para una variedad de tareas de aprendizaje automático, incluida la clasificación de imágenes.

Según la documentación oficial de TensorFlow, el uso principal de TensorFlow en el aprendizaje automático se puede resumir de la siguiente manera:

1- Cómputo Flexible: TensorFlow le permite definir y realizar cálculos matemáticos utilizando gráficos de flujo de datos. En estos gráficos, los nodos representan operaciones matemáticas y los bordes representan el flujo de datos entre estas operaciones. Esta flexibilidad es un concepto fundamental en TensorFlow que permite a los usuarios construir y experimentar con una amplia gama de modelos de aprendizaje automático.
2- Escalabilidad: TensorFlow está diseñado para ser escalable. Puede entrenar y desplegar modelos en una variedad de plataformas, desde dispositivos móviles hasta clústeres de potentes GPU. También proporciona soporte para la informática distribuida, lo que le permite aprovechar el poder de varias máquinas para entrenar modelos grandes.
3- Personalización: TensorFlow es altamente personalizable. Puede definir sus propias operaciones, funciones de pérdida y modelos, lo que le permite experimentar e innovar con sus algoritmos de aprendizaje automático. Esto es particularmente útil para adaptar modelos a tareas específicas, como la clasificación de imágenes.
4- APIs de Alto Nivel: TensorFlow ofrece APIs de alto nivel como Keras que simplifican el proceso de construcción, entrenamiento y despliegue de modelos de aprendizaje automático. Estas APIs son amigables para principiantes y permiten un desarrollo más rápido de modelos, incluidos los modelos de clasificación de imágenes.

<u>El uso de TensorFlow en la clasificación de imágenes, según la documentación oficial:</u>

En la clasificación de imágenes, TensorFlow proporciona una plataforma potente para crear y entrenar modelos de aprendizaje profundo. Así es cómo TensorFlow se usa comúnmente para tareas de clasificación de imágenes:

1- Preparación de Datos: TensorFlow le permite cargar y preprocesar eficientemente datos de imágenes. Puede utilizar herramientas como TensorFlow Datasets y TensorFlow Data API para gestionar y aumentar sus conjuntos de datos de imágenes.
2- Construcción de Modelos: TensorFlow proporciona varios modelos predefinidos, como redes neuronales convolucionales (CNN), que son adecuados para tareas de clasificación de imágenes. Además, puede personalizar y construir sus propias arquitecturas de redes neuronales utilizando la API flexible de TensorFlow.
3- Entrenamiento de Modelos: La diferenciación automática y las bibliotecas de optimización de TensorFlow facilitan el entrenamiento de sus modelos de clasificación de imágenes. Puede definir funciones de pérdida y utilizar optimizadores para mejorar iterativamente la precisión del modelo en los datos de entrenamiento.
4- Evaluación de Modelos: TensorFlow ofrece herramientas para evaluar el rendimiento de su modelo de clasificación de imágenes, incluyendo métricas como precisión, precisión y exhaustividad.
5- Despliegue de Modelos: Una vez que su modelo de clasificación de imágenes esté entrenado y evaluado, TensorFlow ofrece opciones para desplegarlo en sistemas de producción, dispositivos móviles o la nube, lo que le permite hacer predicciones sobre nuevas imágenes no vistas.

Definición de rutas de directorio:

Se definen rutas de directorio para las imágenes de entrenamiento de las dos clases (ALL y HEM) en tres carpetas diferentes, correspondientes a tres conjuntos de datos plegados (fold_0, fold_1, fold_2).

Función get_path_image:

Se define una función que recopila las rutas de todas las imágenes en una carpeta dada y las almacena en una lista llamada img_data.

Creación del DataFrame data:

Se crea un DataFrame de Pandas llamado data con dos columnas: "img_data" para las rutas de las imágenes y "labels" inicializadas con valores NaN (serán etiquetas de clase).

Etiquetado de las imágenes:

Se asignan etiquetas a las imágenes en función de la carpeta en la que se encuentran. Las imágenes en las carpetas "all_" se etiquetan como 1 (ALL) y las imágenes en las carpetas "hem_" se etiquetan como 0 (HEM).

Preprocesamiento de imágenes:

Se recorren todas las imágenes y se aplican una serie de operaciones de procesamiento de imágenes, como convertirlas a escala de grises, aplicar umbralización, eliminar fondos y recortarlas para obtener regiones de interés.

Extracción de características:

Se utiliza un modelo de aprendizaje profundo (ResNet50, VGG19 o ResNet101) preentrenado para extraer características de las imágenes preprocesadas. Las características se almacenan en un DataFrame llamado features_df.

Escalado de características:

Las características extraídas se escalan utilizando Min-Max scaling.

Selección de características:

Se aplican diferentes métodos de selección de características, como ANOVA, RFE y SelectFromModel (basado en Random Forest), para reducir la dimensionalidad de las características.

División de datos:

Se dividen los datos en conjuntos de entrenamiento y prueba.

Entrenamiento y evaluación de modelos de aprendizaje automático:

Se entrenan varios modelos de clasificación, incluyendo K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Random Forest (RF) y Gaussian Naive Bayes (NB).

Se realizan evaluaciones de precisión, recall, F1-score y se genera una matriz de confusión para cada modelo.

Búsqueda de hiperparámetros (GridSearchCV):

Se realiza una búsqueda de hiperparámetros para SVM y Random Forest utilizando GridSearchCV.

Guardado de modelos:

Los modelos entrenados se guardan en archivos utilizando la biblioteca joblib.

Este código realiza un flujo completo de procesamiento de imágenes médicas y construcción de modelos de aprendizaje automático para la clasificación de leucemia linfoblástica aguda (ALL) y células sanguíneas normales (HEM). Se aplican diferentes técnicas de preprocesamiento de imágenes y selección de características, y se evalúan varios modelos de clasificación para encontrar el que tenga el mejor rendimiento en el conjunto de datos.