EDA steps:

1) **DICOM Image Loading: Loaded** DICOM images from a specified folder path (folder_path) by iterating through all files with the. dcm extension.
2) **DICOM Image Processing:**
   - **Pixel Array Extraction**: Extracted the pixel array from each DICOM file using dicom_data.pixel_array.
   - **Resizing**: Resized each DICOM image to a fixed size of (128, 128) pixels for consistency.
   - **Normalization**: Normalized pixel values by dividing each pixel value by the maximum pixel intensity to scale values between 0 and 1.
3) **Label Extraction and Encoding:**
   - **Filename Parsing for Labels**: Extracted a numeric identifier from each filename (e.g., 001 from 1-001.dcm).
   - **Synthetic Label Creation**: Assigned labels based on whether the extracted number was even or odd ("normal" for even and "abnormal" for odd).
   - **One-Hot Encoding of Labels**: Converted the labels into a one-hot encoded format using pd.get_dummies.
4) **Data Conversion and Structuring:**
   - **Image Array Conversion**: Converted the list of processed images into a NumPy array with a shape that includes a channel dimension (samples, 128, 128, 1).
   - **One-Hot Encoded Labels to Numpy**: Converted the one-hot encoded labels to a NumPy array with float32 data type.
5) **Data Splitting:**
   - **Train-Test Split**: Split the dataset into training and validation sets with an 80-20 ratio using train_test_split.