# **Documentation for Data Processing and Model Training**

## Overview

This document describes the process of handling large datasets, extracting features from images, and training a machine learning model. The workflow is divided into four main tasks: data splitting, feature extraction, data compilation, and model training.

## 1. Data Splitting - File: train\_data\_splitter.ipynb

### Purpose

To manage large datasets by splitting them into smaller, more manageable files of 25,000 rows each.

### Process

* **Load Dataset**: Read the dataset from the specified file path.
* **Calculate Batches**: Determine the number of batches required based on the batch size.
* **Save Batches**: Write each batch to a separate CSV file in the specified output directory.

## 2. Feature Extraction - File: feature\_extraction.ipynb

### Purpose

To extract features from images and prepare the dataset for machine learning models. This involves using a pre-trained ResNet model to extract features and encoding categorical variables.

### Process

* **Load Dataset**: Read the dataset containing image links and categorical data.
* **Encode Variables**: Use LabelEncoder to convert categorical variables into numeric form.
* **Extract Features**: Download images from URLs, preprocess them, and use a ResNet model to extract features.
* **Handle Errors**: Skip rows with problematic images or errors.
* **Save Results**: Write the extracted features and encoded variables to a CSV file.

## 3. Data Compilation - File: data\_compliler.ipynb

### Purpose

To merge individual feature files from training and test datasets into single consolidated files.

### Process

* **Load Feature Files**: Import CSV files with extracted features for both training and test data.
* **Merge Data**: Combine these files into comprehensive datasets for each type (train and test).
* **Save Merged Data**: Export the combined feature datasets to new CSV files.

### 4. Model Training - File: model\_training.ipynb

### Purpose

To train a K-Nearest Neighbors (KNN) model using the processed training data and evaluate its performance on the test data.

### Process

* **Load Data**: Import the consolidated training and test datasets.
* **Train Model**: Fit a KNN classifier to the training data.
* **Make Predictions**: Use the trained model to predict outcomes on the test data.
* **Evaluate Performance**: Assess the model using metrics like accuracy, precision, recall, and F1 score.

## 

## Conclusion

This workflow efficiently handles large datasets through splitting, extracts meaningful features using advanced image processing techniques, and trains a machine learning model for predictive analysis. Each step is designed for clarity and reproducibility, ensuring a robust process from data preparation to model evaluation.