***Logistic Regression Doc***

**Logistic Regression Classification Using Pretrained MobileNetV2 Features**

This project combines the feature extraction capabilities of the pretrained MobileNetV2 model with the classification power of Logistic Regression to classify pet images as cats or dogs. Below is a detailed explanation of the methods used:

**1. Image Preprocessing and Data Augmentation**

The **ImageDataGenerator** class from TensorFlow/Keras was used to preprocess the images and apply data augmentation techniques. Preprocessing included:

* **Rescaling**: Normalizing pixel values to a range of [0, 1] using rescale=1./255 to ensure the MobileNetV2 model, which was trained on normalized data, performs accurately.
* **Augmentation**: Random transformations like rotation, width/height shifts, zooming, shearing, and horizontal flipping were applied to artificially increase the diversity of the training dataset.
* **Validation Split**: A subset of 20% of the data was reserved for validation using validation\_split=0.2.

The images were resized to a target size of **224x224 pixels**, the default input size for MobileNetV2, and processed in batches using a flow generator.

**2. Feature Extraction Using MobileNetV2**

The pretrained **MobileNetV2** model, trained on the ImageNet dataset, was used for feature extraction. The top classification layer of the model was excluded using include\_top=False, allowing us to use its convolutional layers to generate meaningful feature maps. Key steps:

* **Freezing the Model**: The MobileNetV2 model weights were frozen (trainable=False) to prevent them from being updated during training.
* **Feature Prediction**: Features for each image batch were extracted using the model's predict() function. This outputs a 3D feature map per image, representing high-level patterns in the image.

The extracted feature maps were **flattened** into 1D arrays to serve as input for the Logistic Regression model.

**3. Logistic Regression for Classification**

The extracted features were fed into a **Logistic Regression** model from scikit-learn. Logistic Regression is a simple yet effective algorithm for classification tasks. Key configurations:

* **Multi-Class Handling**: Using multi\_class='multinomial', the model handled multiple class labels (e.g., cat and dog).
* **Maximum Iterations**: The max\_iter=1000 parameter ensured sufficient iterations for convergence during training. The model was trained on the training features (X\_train) and corresponding labels (y\_train) to map the relationship between features and output classes.

**4. Model Evaluation**

The model's performance was assessed using:

* **Accuracy**: Calculated using accuracy\_score() to measure overall predictive performance.
* **Classification Report**: Provided detailed metrics such as precision, recall, and F1-score for each class.
* **Confusion Matrix**: Displayed the true versus predicted labels in a heatmap, providing a visual understanding of misclassifications. Seaborn's heatmap() function was used for visualization.

**5. Individual Image Prediction**

A single input image was processed to demonstrate real-time predictions. Steps included:

* **Preprocessing**: The image was resized to 224x224 pixels, converted to an array, and normalized to match the MobileNetV2 input format.
* **Feature Extraction**: Features were generated using the pretrained MobileNetV2 model.
* **Prediction**: The flattened features were passed to the trained Logistic Regression model, which returned the predicted class label (cat or dog).
* **Visualization**: The original image was displayed with its predicted class label.