The code is used to classify the image as a night or day.

Firstly, we load the datasets and visualize the images. We then preprocess the input data images, the function takes in a list of image-label pairs and outputs a standardized list of resized images and numerical labels. Then we create a feature that represents the brightness in an image. We'll be extracting the average brightness using HSV color space. Specifically, we'll use the V channel (a measure of brightness), add up the pixel values in the V channel, then divide that sum by the area of the image to get the average value of the image. Then we build a classifier using the average brightness feature that takes in a standardized image and returns a predicted_label for that image. This estimate_label function should return a value: of 0 or 1 (night or day, respectively). Then we test the classification algorithm using our test set of data that we set aside at the beginning of the notebook! Below, we load the test dataset, standardize it using the standardize function you defined above, and then shuffle it; this ensures that order will not play a role in testing accuracy. We received an accuracy of 93.75% by using only one feature extraction, i.e., the average brightness of the image.

1) Explain when we should use Machine Learning and when not.

   Machine Learning is used in many ways these days, in detecting unusual transactions, sorting spam emails, chatbots, voice recognition, text classification, image classification, etc. We can use ML when we need to make a future decision based on past data, especially when the input data is huge, and many factors are influencing the decision. Sometimes, it becomes hard for humans to perform tedious tasks that could be simply handled by ML, like finding fraudulent transactions, sorting spam mail, etc.

   Machine Learning is not always the solution for all problems, there could be many other problems that can provide a better solution than ML. We should not be using ML especially when we are solving fewer complex problems that can be solved with traditional methods with equal efficiency as that of ML. Also, when there is no clearly labeled data, it is not advisable to use ML. There is a simple rule: Don’t build a machine-learning model where a simpler approach might succeed just as well.

2) Compare the accuracy of both ways. Also, explain why you are getting the difference in accuracy.

The accuracy is high while using software development for image classification, it is around 93.5% but the accuracy is around 97% when we use the ML model, I have used a Neural network with 50 epochs to train the model. If the number of epochs is reduced the accuracy decreases. There is a difference in the accuracies because in the traditional software development for image classification, we do not train the model repeatedly to efficiently classify the images, just based on the brightness of the image it is classifying the image as day or night, this might not work effectively for any other dataset. But for ML models considers various features in the training images are learned to classify images even if we change the dataset, the model learns from the images and will predict the outputs effectively, just by tuning the parameters. The main reason for the difference in accuracy is that the traditional software just classifies images based on brightness but the ML model considers many features to classify the image.

3) Why do we use Machine Learning rather than just software development?

We use ML because it handles huge data with multiple trends to make a decision, which traditional software might not succeed at. We might use software development to solve a specific problem that might not work for any other dataset. But ML has a scope of learning on its own and producing an effective output. In the traditional software development for image classification, we do not train the model repeatedly to efficiently classify the images, just based on the brightness of the image it is classifying the image as day or night, this might not work effectively for any other dataset. But ML models consider various features in the training images that are learned to classify images even if we change the dataset, the model learns from the images and will predict the outputs effectively, just by tuning the parameters.

4) How do you improve the performance of a model and why do we need to improve the model performance?

We can improve the performance of the model by keeping in mind a few important factors before developing a model.

1. It is important to choose the right amount of data. It is important to have large training data for efficiently training the model.
2. The quality of data is also important, we must include diverse data in the training dataset for accurate results. If the dataset is not good the model will give inaccurate outputs.
3. It is also important to choose the right algorithm. The algorithm takes input data and predicts outcomes. Hence it is important to have an efficient algorithm for the specified problem to get accurate results.
4. It also depends on the method we use for training i.e either supervised, unsupervised, or reinforcement training.
5. It is also important to validate and test the model with the right data, we need to make sure that we are not using training data in model validation and testing.

It is important to improve the model performance because the better the performance of the model, the more accurate are the predictions. Especially in medical and scientific fields, it is important to have accurate predictions because a slight mistake can cost millions or even a life.