

# **SML - Kaggle Competition**

## Problem :

We are given a dataset of images and we have to perform classification task. Main code is with name : “2017233\_sml.py”. There are total 25 categories. We are suppose to predict the correct category of images residing in SML\_Test folder.

## Data :

For training, we have 16000 labeled images and for testing, we have 1500 images.

## Methodology:

Firstly, I have extracted the training image and have converted them to a matrix. Then, I flatten the matrix making it a vector. Now, I have a matrix of 16000 images, where each element is a vector (image). Length of vector is 4096 (since image's dimension are 64x64).

### Reducing the size :

Since not all features of image are important, I have applied **PCA (Principle Component Analysis)** to reduce the size of 4096 feature vector. The resultant size of vector after PCA is 1371.

### Dealing with outliers :

I computed mean and variance of images in a particular category and then, I just deleted the image vector if the image vector was more that  $1.25 \times \text{variance}$  away from the mean. For example, let's say I am dealing with images of category 0, I'll compute mean of vector residing in category 0 ( = mean) and variance ( = std). Now, I iterate over the image vector ( = vector) of category 0 and if the  $(\text{vector} - \text{mean}) \geq 1.25 \times \text{std}$ , I reject that vector and if  $(\text{vector} - \text{mean}) < 1.25 \times \text{std}$ , I accept that image vector for training purpose.

### Multilayer Perceptron (MLP) :

I have used MLP from sklearn library, which will be having 5 hidden layers with each layer having 100 nodes. I feed in the training data to the model and then use that model to predict the testing data. I have saved the trained model in “model.joblib” file. You can load the model by following statement :

```
from joblib import dump, load
model = load("model.joblib")
```

After that, you can use the model to predict the value by inputting PCA vectors (of dimension 1371) matrix.