ML Project

Project Report

In this project, we are using google colab for the implementation of two classifier. First of all, we connect the project with drive as we are loading data from drive. We are using np.load() function to load data file by file from drive into variables. Now, all variables have a multi-dimensional numpy array in them. We append these numpy arrays into a train\_data variable so that operations can be performed easily on all the files and we don’t have to work on each file separately. Same steps are happening with test\_data to load.

We have implemented a function for the feature calculation. This function is taking a list of three vectors as argument and calculating each feature from a vector and append it in features list one by one. We are calculation one feature for all three vectors at a time so that we don’t have to change the feature matrix again and to keep it simple for our understanding. This function calculates all features and returns a list of all 54 features.

After feature calculation function, we are using a nested loop to get features of all 9 sensors. Outer loop takes each sensor data and inner loop call feature\_calculation() function on each example of this data by slicing data into three vectors. Each time function returns a list, we append this list into a matrix variable. Once all the examples of one sensor are done, We append the data of this sensor to the train\_feature\_matrix variable. After calling function on all the sensors, We have a feature matrix with the features of all sensors.

As data is in three dimensions, we cannot apply our classifier on it. We need to decrease a dimension and change the data to our desire form. For this purpose, we are using np.hstack() function. While calculating the skew and kurtosis feature, some of the values become NaN because of the data redundancy. In this state, we cannot apply classifiers to this data. It gives error. So, we have places 0 in the place of NaN value to overcome this error.

We are calculating the features of testing data in the same way we are calculating the training data features. And then stacking the data to change it to desire dimensions as we have done before.

After all these operations on data, we are using SVC classifier from sklearn library to classify. We are using pipeline from sklearn library. We don’t have to much knowledge of pipeline. One of our senior asked us to use it as it is a good practice. In this pipeline, we are applying preprocessing (StandardScaler) from sklearn library before using classifier. Classification report gives the matrices of all the 55 classes and we get total accuracy and other matrices to check our classifier.

After SVC, we are also using Random Forest Classifier. Working of this classifier is same as above. This classifier is also giving accuracy almost same as the SVC is giving.

SVC:

Accuracy : 0.6774475524475524

Recall : 0.6777564273826368

F1-score : 0.6744904068098813

Random Forest:

Accuracy : 0.6722027972027972

Recall : 0.6724781940859956

F1-score : 0.665925634748932

Note:

We used Chat-Gpt to get the formulas of all the features calculated in feature\_calculation() function.

We used sklearn web site to see how to use sklearn library.