ENG19CS0367 - Yash Raj Ojha

Here is my approach on how I dealt with the dataset and how to use the data for training and testing the model.

Exploring the dataset

We look at the dataset, we have three columns of features and one column of label, although at first look this dataset doesn't look meaningful, some graphs can help us determine how every feature column helps us determine the label.

Steps that I followed to explore the dataset:

- List out the head and tail of the dataset,
- Find out the unique values in the label columns, so we can derive more about them in the graphs used,
- Check for how big is the dataset.

Now that we know the structure of our dataset, we move on to find more info about our dataset and check if there are any null values or not.

After understanding the dataset in numbers, I used count plot to find how many rows are available for every human activity mentioned. Each of the activity is then plotted using the plot function to see the variation in X,Y,Z values. After plotting out X,Y,Z values, we use a box plot to check what range of values these columns lie in.

All the steps above gave me an idea of the dataset, so I decided to move ahead with processing the data which will be used to train the model.

Data Processing

- The first step here was to shuffle and split the dataset into 2 variables, one for features and one for labels.
- The labels data was of string type, and ML models aren't trained on string data, so to convert string data to numbers, I used Label Encoding, which represents categorical variables(example "Jog") as integers.

- After converting string labels data to numbers, we split it to training data and testing data using the `train test split` function from `sklearn.model_selection`.
- One the splitting is over, we have to scale the data, we need to bring values of X,Y,and Z on the same scale so that our ML model doesn't give a higher weight to any one of the columns.
- After scaling, we apply PCA for 2 components on training data and testing data.

Model Selection

Logistic Regression

When we think of classification we think of Logistic Regression. Logistic Regression is used when the dependent variable(target) is categorical. For example, To predict whether an email is spam (1) or (0) Whether the tumor is malignant (1) or not (0).

But in our case the label is not binary, it is not just "Sit" or "Stand", it is 4 unique values in other. I expected Logistic Regression to perform poorly, but the results would give us an idea when we compare the result to other model which I'll apply on the same dataset.

I trained my LR model and got an **accuracy of close to 64%**, which isn't good and was expected, so I had to change the model for better prediction accuracy.

K-Nearest-Neighbours

The next best model I could think of was KNN, because in simpler terms, the KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other.

So I applied the KNN model with 5 neighbours, and distance metric of minkowski. This model gave a **prediction accuracy of close 93%**. It clearly shows how KNN was useful in finding the human activity if three point are given.