Activity Recognition Project (Part 2)

CSE 572: Data Mining
Spring 2020

Kenji Mah[1]

[1]**kmmah@asu.edu**

ABSTRACT

The purpose of this project is to utilize the PCA feature matrix from the previous project as input for machine learning algorithms. The use of PCA reduces the dimensionality of the extracted features in order to speed up the machine learning algorithm.

Keywords: Principle Component Analysis, Decision Tree, Support Vector Machine, Neural Network

DATA PREPARATION

a: Raw Data analysis

In order to prepare the data for machine learning algorithms we first need to address some problems in the raw data given to us. As in the previous project we must add a comma in the file 1503609551913.txt, from user19/spoon. For this project we needed to aggregate the rows by gestures. Therefore, after labeling the eating and non-eating timeframes we use the five feature extractions (FFT, Average, STD, Max, Min) on every continuous set of frames of eating and then on every continuous set of frames of non-eating. We do this to avoid losing information about this time sensitive data. After doing this for both EMG and IMU data I noticed that the EMG data contained 4 more gesture instances than the IMU data. After doing some analysis on the raw data I narrowed it down to missing gestures from users 25 and 18. After thoroughly going through the raw data, I found that the IMU data was cut off and did not contain the frames needed to cover the entirety of the ground truth file, while the EMG covered it. For example for user 25's spoon data, after converting the number of frames from the ground truth file to seconds elapsed, there needs to be at least 366.9585 seconds of IMU data, but only 227.4495 seconds of IMU and EMG data were given. Because the method in which we synchronize the data is approximate the gestures lost may differ if there is not a sufficient amount of data from the sensors. This only happened in spoon 25 My data and spoon 18 Myo data, So I adjusted the matrices accordingly.

b: Feature Extraction

For every user we will aggregate the pure data by gestures. So we will do each feature extraction for every continuous eating and non eating action. This however leads to problems in which some time spans for for an action are too small to extract the top 5 Fast Fourier Transformation frequencies. In these cases I just nullified the top 5 frequencies. After aggregation of the gestures, the resulting number of samples reduces to 4671 with, 2335 non-eating samples and 2336 eating samples.

c: Principle Component Analysis

For the PCA setup we will use the sklearn library. We will use PCA(.95) which chooses the minimum number of Principle Components to account for 95% of the total explained variance and fit and transform the training data for each phase. Then we will transform the test data using the previous computation of PCA that was on the training data.

PHASE 1: USER DEPENDENT ANALYSIS DATA SPLIT

For this phase we will be analyzing how each model performs when there is training data and test data for each user, specifically 60% training data and 40% test data from each user.

PHASE 2: USER INDEPENDENT ANALYSIS DATA SPLIT

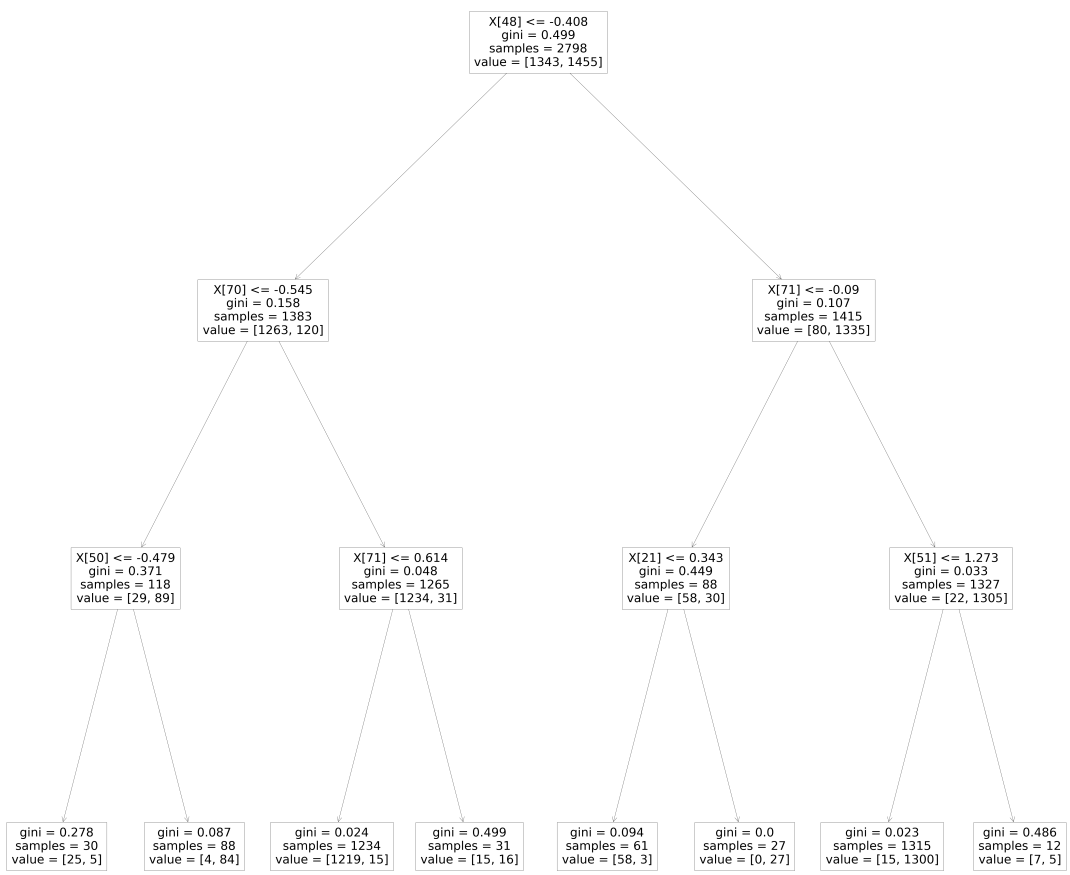
For this phase of the project we will split the training data and test data by users. This means that all of a specific users data will either be in the training set or in the test set. So in this case we will take 60% of users and put them in the training data, and then take 40% of the rest of the users and put them in the test data.

MACHINE LEARNING ALGORITHMS

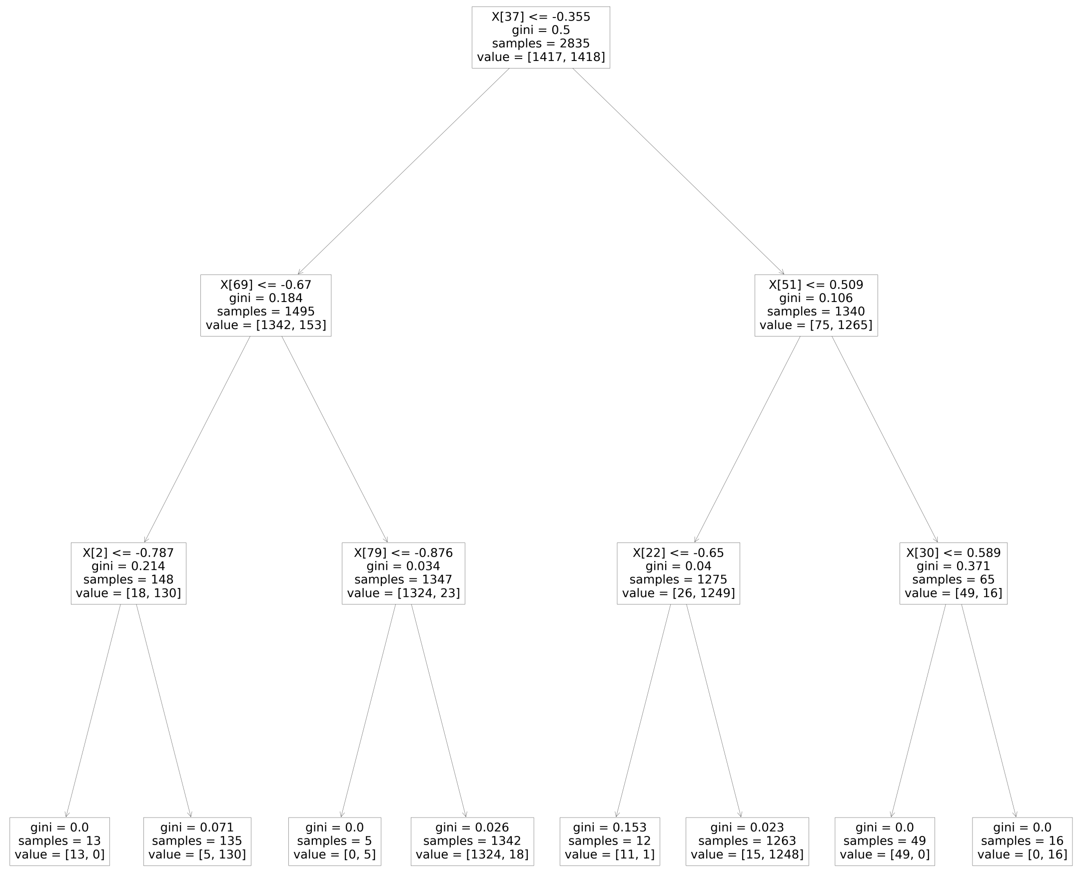
For each phase we will apply the following 3 machine learning algorithms: Decision Trees, Support Vector Machine, Neural Network All of the following algorithms will utalize sklearn's implmentations.

Decision Tree

For this algorithm I found that it was sufficient and somewhat optimal to put pre-prune the tree at a max depth of 3. This is because sklearn does not have an function to post prune. The rest of the algorithm uses default parameters as stated in sklearn's documentation for decision trees. For Phase 1 this algorithm resulted in the following tree:



Phase 2 of the project resulted in the following tree:



Support Vector Machine

For this algorithm we will use a linear SVM with the rest of the parameters being left to their defaults.

Neural Network

For this algorithm we will use the the Limited-memory Broyden–Fletcher–Goldfarb–Shanno algorithm (lbfgs) as our optimizer, an alpha of $1x10^{-5}$, hidden layers size of (15, 5) and 2000 max iterations for ample speed up of the algorithm. The rest of the algorithm uses default parameters as stated in sklearn's documentation for the Multi-layer Perceptron classifier.

RESULTS

For both phases we will be analysing the Accuracy, Precision, Recall, and F1 Score's for each algorithm. For Phase 1 of the project the table shows following results:

phase1MetricsTable - DataFrame

Index	accuracy	precision	recall	f1	method
0	0.945008	0.921909	0.964813	0.942873	Decision Tree
1	0.979712	0.972004	0.985244	0.978579	Support Vector Machine
2	0.983449	0.980769	0.984109	0.982436	Neural Network

For Phase 2 of the project the table shows following results:

phase2MetricsTable - DataFrame

Index	accuracy	precision	recall	f1	method
0	0.938998	0.949777	0.927015	0.938258	Decision Tree
1	0.958606	0.981693	0.934641	0.957589	Support Vector Machine
2	0.967865	0.983127	0.95207	0.967349	Neural Network

It seems that the user independent split performed slightly worst than the user dependent split, but not by much. My assumption is that an individual user's eating pattern differs from other users, therefore making it better to predict the correct class if an individual user's data is in both the test and training set(phase 1).

In conclusion, these set of eating and non eating gestures can be well predicted when aggregated by a continuous set of eating and non eating frames. I don't see this being used in a live prediction example, however by changing the aggregation method we could potentially come up with a solution. One method may be by using a sliding window of a constrained timeframe, similar to a convolution in image processing, and creating new features like that. This will increase computation time by a lot however, but may allow for the algorithm to be able to distinguish eating in real time.