Gesture Recognition for Cybernetic Hand Prosthetics

Open Source Prosthetic Control

Austin Brown, Syed Aziz

05.04.2021

INTRODUCTION

Open-Source Software is a fantastic tool for private and public organizations across the globe. Teams with the expensive hardware can receive extensive help with software tasks. One example is Cyber Punk Me. They are attempting to create gesture recognition software for prosthetics that read muscle activity with an EMG. The data read is sent to the persons smartphone and a model decides what gesture is being made. Our goal is to create the model.

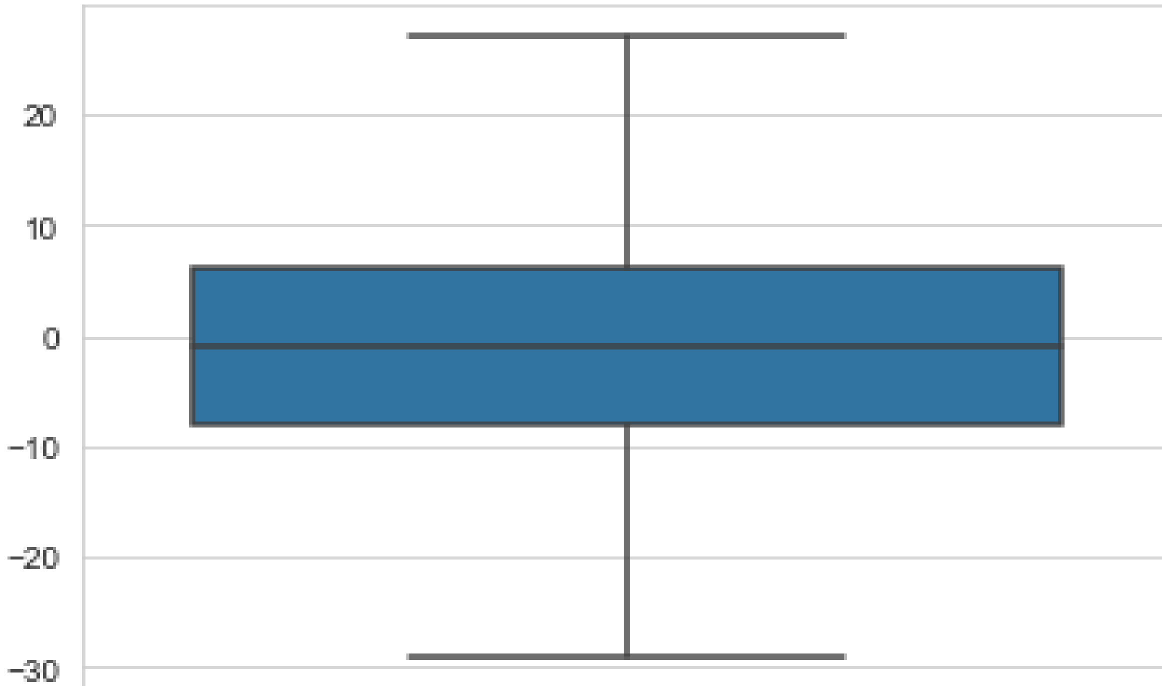
STATEMENT OF GOAL

The only thing needed is an accurate model, which will likely come from modeling the raw EMG data, and we will attempt to provide this. However, as Data Scientists we want to understand how our models work, not just that they do work. To this end we will also be taking another approach to this classification problem in an attempt to recreate what some black box models may be doing. They will be described in their own sections, Procedure 1 and Procedure 2.

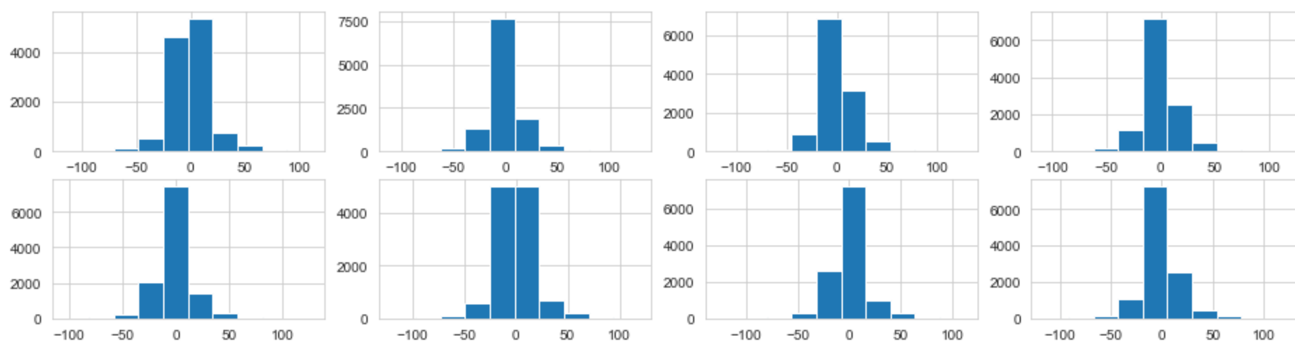
DATA

The input data for this project is EMG readings from 8 sensors around a person's wrist/forearm. The data is recorded while a person is holding a specific gesture. The gestures are rock, paper, scissors, and

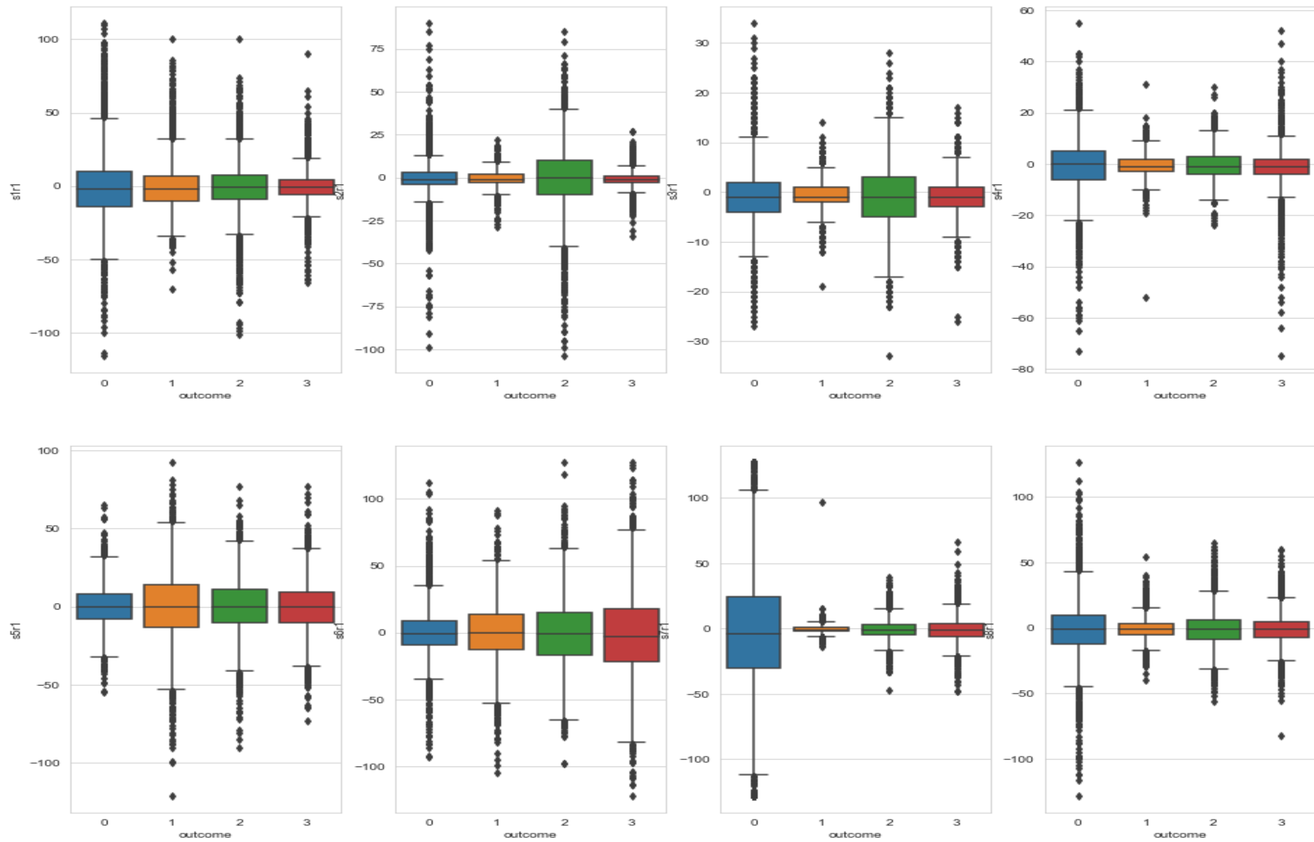
OK (👌). Each sensor took 8 readings for a total of 64 numbers per data point. Each reading results in a decimal number between about –120 and 120.



But, as you can see in this boxplot, which excludes outliers, most EMG readings are between 30 and -30. A prudent question is what does this data look like and how will that effect our models?



These histograms each graph a specific reading from sensor 1. So, the top left graph is the first readings from each data point. These graphs correspond well with the above boxplot, and each sensor looks similar to these histograms. This means that there is no obvious feature that majorly influences prediction and that data reduction techniques may not prove effective for this data set (Spoiler Alert: They do not). But this observation does not mean that convoluted models are the only way we will be able to classify these hand gestures. It may be the best for this data set, but this data set is not a comprehensive representation of the environment. We only have 8 sensors in this data. The placement of each sensor will determine how relevant its readings are to classifying gestures.



The above boxplots show each sensor first reading, divided by gesture. There are a few patterns here that may be useful for classification, but I want to focus on Sensors 7 and 8, the bottom 2 on the right. We can guess that these sensors are placed around muscles that only flex for gesture 0. These are well placed sensors if they can easily detect specific gestures. If we had more sensors placed like 7 and 8, but for other muscles, then we could vastly increase the reliability of our models. Smartly placed sensors would allow for the same accuracy with less sensors which would be a less invasive experience for the user. Recommendations on sensor placement is outside the scope of this report.

DATA REDUCTION

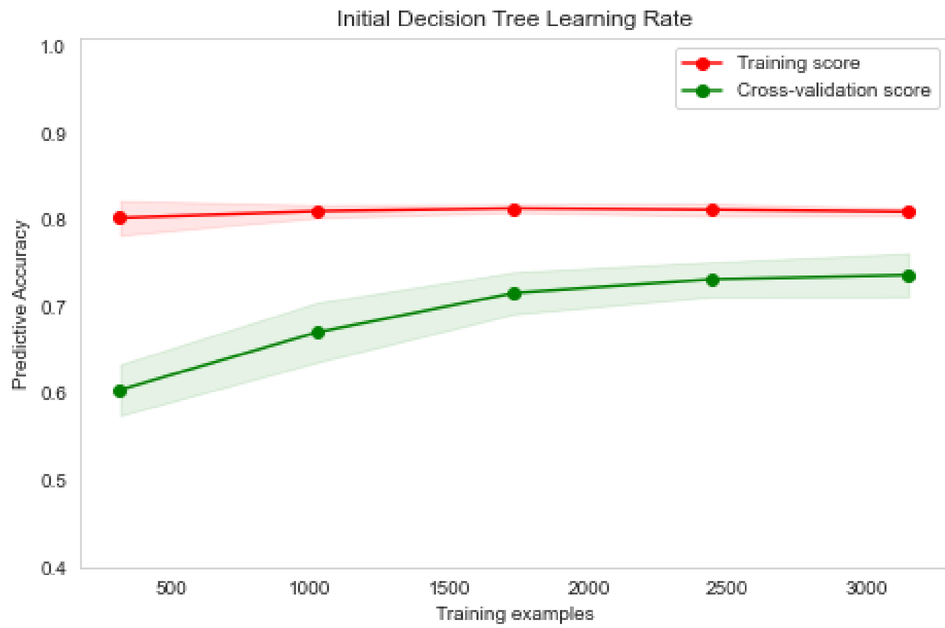
As stated in the previous section, it is likely that data reduction and scaling may not improve the performance of our models. To test this, we attempted to train two models on data that we scaled, and then performed LDA and PCA. Our models, Random Forest and KNN, had an accuracy of 99.1% and 66.3% respectively, on the raw data. After scaling the accuracy was reduced by at most 1% each. After LDA the accuracy was reduced to 40% and 37%, and after PCA accuracy was 78% and 64%. These results support our assumption that this data will not benefit from data reduction. All of our models will be trained and tested on unmodified data.

PROCEDURE 1

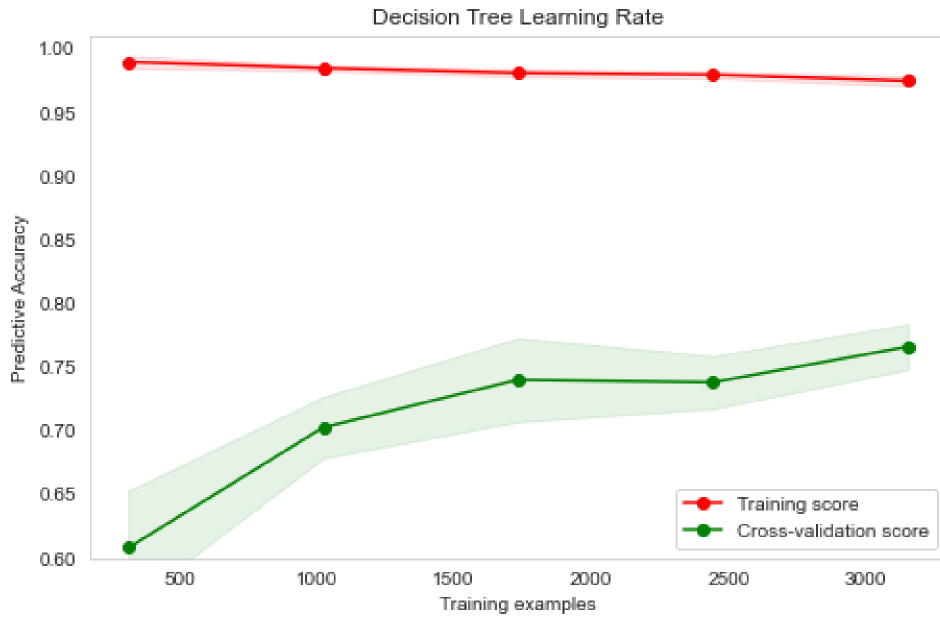
This section will describe the process of creating an accurate model to classify the gesture based off the raw EMG data. We are going to try 4 different models, Decision Tree, SVM, Random Forest, and LSTM Neural Net.

Decision Tree

A Decision Tree with some default parameters had a test accuracy of 73% and the following learning curve.



After a grid search to find the optimal parameters it achieved a test accuracy of 75% and the following

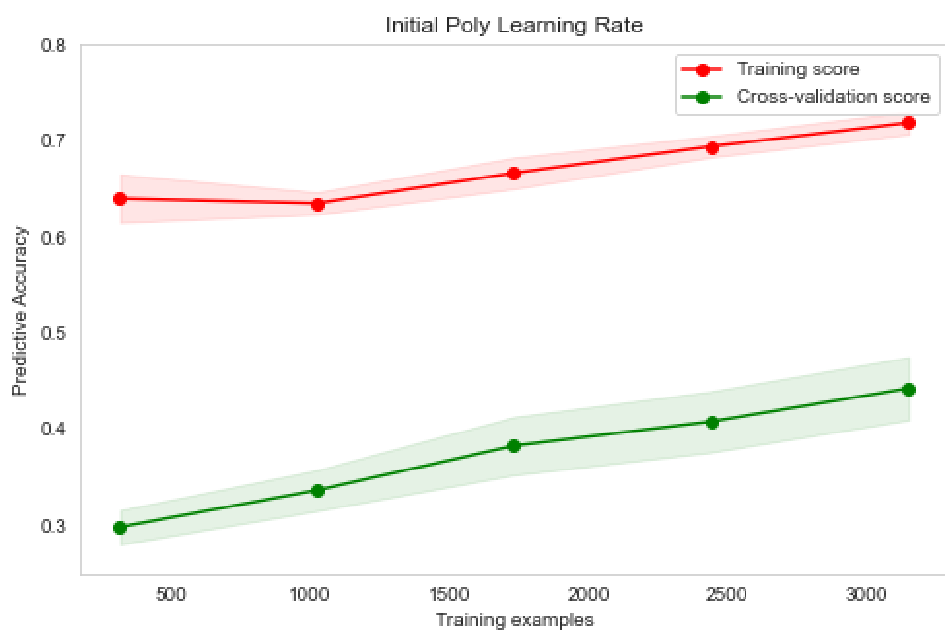


learning curve.

This model had an F1 score of .758, which helps us conclude that this model is not bad but it is defiantly not good enough.

SVM

For the SVM model we decided to use two different kernels, the Polynomial kernel and the Radial basis function (RBF) kernel. The polynomial kernel starts with poor performance with a test accuracy of 43%.



After using a grid search to find the optimal parameters, the test accuracy increased to 87% and has a r^2 score of 0.804. This model also had an F1 score of 0.878. The results after tuning the model were good.

For the RBF kernel, we see a test accuracy of 82%, a r^2 score of 0.718, and a F1 score of 0.82. Overall, we see the polynomial kernel was much better for this dataset.

RANDOM FOREST

With this model we dove straight into a grid search and found a promising solution with an accuracy of 91% and an F1 score of .91. Here we decided to look at the confusion matrix to see what area was hardest to classify for this model. The 4th gesture, the 👌, is the hardest to classify for this model with it both having false negatives and false positives. Analysis of the confusion matrix may be the key to finding insight into the placement of the sensors. By this I mean you could iteratively add/move sensors and observe the effect on the confusion matrix and see the positive or negative effects on classification of individual gestures. This is of course an anecdotal observation and testing of this hypothesis is outside the scope of this report.

LSTM NEURAL NET

Saving the best for last, a neural net has been shown by other groups to be the most accurate model for classifying this data.

Neural nets are versatile and scalable, and they can handle high dimensionality tasks with ease. The number of epochs is a hyperparameter that defines the number times that the learning algorithm will work through the entire training dataset. One epoch means that each sample in the training dataset has had an opportunity to update the internal model parameters. The number of epochs is traditionally large, often hundreds or thousands, allowing the learning algorithm to run until the error from the model has been effectively minimized, thus making it more effective compared to other models.

For our attempt we used a 4-layer LSTM model from KERAS with dropout layers in between. After training for 250 epochs, we created a model with a 99.9% accuracy on the training set and a 95.5% accuracy on the testing set.

PROCEDURE 2

This section will describe the 2nd approach to this problem where the goal is to see if a more explainable model can be used. The approaches to this classifying problem we have seen so far revolve mostly around using the data as 64 dimensional points. This is fine and works well for some models, however it can become ambiguous as to why these models come to their conclusions. This is the problem of interoperability is important when we need to trust a model is making its decisions for the right reasons. Biases in the training of a model will produce that bias in the output. So, it is a good idea to attempt methods of classification that are more explainable as we might observe these biases and fix them. This can also help us create better models for similar tasks as we have a better idea as to what can transfer from task to task. As an example, one of the models that seem to work best for this data is a neural network. Attempting to know why this model makes its decisions is difficult. So, if we can create other, similarly accurate models we may get insight as to how the neural network makes its decisions.

WHAT DID WE DO

We explored the ability of the sensors to be individually classified. We wanted to know if we could classify a sensor into groups that we can then use to classify the gesture being held. A possibility is that we can have two classes, muscles at rest, and muscles that are stressed. This would break up our gesture classification into 2 steps. First, we would need to cluster each of the 8 sensors, then use those classifications to classify the gesture. Because we have no data about the individual sensors, like their position or what the muscle was doing, we had to use unsupervised techniques to determine whether the sensors could be classified into meaningful groups. We used a few clustering algorithms that attempt to apply classes to clusters of data, such as k-means, 2 hierarchical models, and DBSCAN. Then, once the model labeled all our data we had to determine if it was good fit. To do this we calculated the silhouette score for each model. This is a ratio of how similar a point is to its cluster over how different it is to other clusters. The score ranges from –1 to 1 with a –1 being the worst model and 1 being highly dense, well separated clusters. Our intuition about the data says that a score very close to one is unlikely and a score around .6 would mean this approach may be viable.

RESULTS

First, we ran k-means, a quick and general-purpose cluster algorithm, on all 8 sensors with numbers of clusters from 2 to 10. In total we trained k-means 72 times, 8 sensors by 9 different amounts of clusters. This resulted in silhouette scores of about .2 and a max of .3. This means that there were some good clusters, but it was mostly random clusters. We are looking for a higher score, a score closer to 1. Next, we tried the affinity propagation model. This model performed the second worst with a score of .08. However, we only tested this on the first sensor. But with such a low score we can expect poor performance for the other sensors. Next is the worst performing model, DBSCAN. This model consistently resulted in scores below 0. This is the opposite of our expectations. We may continue to test and tune this model to see if better scores can be achieved. Lastly, we have two hierarchical clustering models.

These two are ward linked and single linked. Ward linked was also only tested on one sensor. It produced a score of .36 when it used 2 clusters had decreasing scores as the number of clusters increased. The single linked model produced the best results. We tested it on all 8 sensors and had significantly better scores. The scores were between .5 and .6 with the best scores being for 2 cluster models.

This result tells us that the sensors can be classified with resendable meaning. This is exactly what we wanted to see. However, upon closer examination we find this score was misleading. In fact, the clustering was useless, as the highest scoring clustering resulted in 2 clusters, one with a single point, and one with all other data points. This is not useful clustering and is a precautionary tale as to why multi-faceted testing is needed. But we did not stop here as there was still a second step to this procedure.

For classification we used Decision Trees, KNN, and SVM. We also used the clustering labels from 3 clustering algorithms, Kmeans, Ward, and single linkage. After all the model training and testing the highest accuracy was 59%, which was from the SVM using the ward linkage clustering labels.

In retrospect it seems obvious why a neural net model would outperform this process significantly, that is because it has feedback. Our clustering step is unsupervised and had no feedback from the classification step. This is likely a massive hindrance to the success of this approach.

CONCLUSION

We have succeeded in creating an accurate model to classify the gesture based on the EMG data. A Neural Net achieved an accuracy of 95.5%. Along the way we tried and failed to achieve an accurate model using a much less "black box" method. However, that is not to say it was fruitless as we devised ways to improve the process for future attempts.