Project 2

Ryan Meagher

I began by splitting the data 2 ways. The first way was user dependent where I put 60% of each user in the training set and 40% of each user in the test set. I did this via sklearn train_test_split with the parameters shuffle=True and stratify=True so that the train and test set samples would be chosen randomly and there would be equivalent ratios of eat:no eat in the train and test sets. In addition to this, since this is time series data and every user has a varying number of rows after computing statistical features for each row of EMG1-EMG8. I decided to aggregate row-wise and the compute the mean of each statistical feature so that every user would have the exact number of samples so that every user was equally represented in the train and test sets. This is applicable for future time series data as once again we could recompute the 5 statistical features for each row and then average this over the time the action takes place.

For the second phase I used user independent splitting where the training set contained 18 users spoon and fork actions and in the test we have 12 different users spoon and fork actions.

For both dependent and independent splitting I calculated the appropriate index of where the necessary split needed to occur so that I could merge the created appropriate train and test datasets so that PCA could be applied to all the data. After this I then used the index to split the datasets back into their appropriate train/test sets.

In order to find the optimal hyperparameters for each model I used the whole dataset and computed sklearn GridsearchCV. This lead me to finding the optimal parameters to be applied for my dependent and independent data splits. After doing this I found my best parameters for svm, decision trees, Random Forest and multilayer neural network to be:

```
Svm: {'C': 10, 'gamma': 1, 'kernel': 'rbf'},
Decision tree:{'criterion': 'gini', 'max_depth': 8, 'splitter': 'best'},
Random Forest:{'criterion': 'gini', 'max_depth': 12, 'n_estimators': 100})
MLP: (activation='logistic', alpha=0.01, hidden_layer_sizes=10,max_iter=15
00,solver= 'lbfgs',random_state=10)
```

The results I got were obtained via sklearn classification_report and can be seen below. The dependent split where every user was contained in both training and testing performed better than the user independent split. I believe my best model to be svm dependent where my training set had a precision, recall and f1score of 85.5%,84.3% and 84.9% respectively and my testing set had a precision, recall and f1score of 82.4%,82.9% and 82.3% respectively. This model is achieving all around good results and you can tell from the training set that it has not overfit the data as it is performing around the same as the testing dataset.

random forest train dependent

	precision	recall	f1-score	support
eat	0.9476	0.9304	0.9389	1767
not eat	0.9314	0.9483	0.9398	1761
accuracy			0.9393	3528
macro avg	0.9395	0.9394	0.9393	3528

```
weighted avg     0.9395    0.9393    0.9393      3528

random forest test dependent
              precision    recall  f1-score   support

         eat     0.8221    0.8235    0.8228      1173
     not eat     0.8241    0.8227    0.8234      1179

    accuracy                         0.8231      2352
   macro avg     0.8231    0.8231    0.8231      2352
weighted avg     0.8231    0.8231    0.8231      2352

random forest train independent
              precision    recall  f1-score   support

         eat     0.9412    0.9257    0.9334      1764
     not eat     0.9269    0.9422    0.9345      1764

    accuracy                         0.9340      3528
   macro avg     0.9341    0.9340    0.9340      3528
weighted avg     0.9341    0.9340    0.9340      3528

random forest test independent
              precision    recall  f1-score   support

         eat     0.7374    0.8741    0.8000      1176
     not eat     0.8455    0.6888    0.7591      1176

    accuracy                         0.7815      2352
   macro avg     0.7915    0.7815    0.7796      2352
weighted avg     0.7915    0.7815    0.7796      2352

svm train dependent
              precision    recall  f1-score   support

         eat     0.8553    0.8427    0.8489      1767
     not eat     0.8444    0.8569    0.8506      1761

    accuracy                         0.8498      3528
   macro avg     0.8498    0.8498    0.8498      3528
weighted avg     0.8499    0.8498    0.8498      3528

svm test dependent
              precision    recall  f1-score   support
```

	precision	recall	f1-score	support
eat	0.8244	0.8286	0.8265	1173
not eat	0.8286	0.8244	0.8265	1179
accuracy			0.8265	2352
macro avg	0.8265	0.8265	0.8265	2352
weighted avg	0.8265	0.8265	0.8265	2352

svm train independent

	precision	recall	f1-score	support
eat	0.8584	0.8594	0.8589	1764
not eat	0.8593	0.8583	0.8588	1764
accuracy			0.8588	3528
macro avg	0.8588	0.8588	0.8588	3528
weighted avg	0.8588	0.8588	0.8588	3528

svm test independent

	precision	recall	f1-score	support
eat	0.7652	0.8588	0.8093	1176
not eat	0.8391	0.7364	0.7844	1176
accuracy			0.7976	2352
macro avg	0.8021	0.7976	0.7969	2352
weighted avg	0.8021	0.7976	0.7969	2352

decision tree train dependent

	precision	recall	f1-score	support
eat	0.8573	0.7685	0.8105	1767
not eat	0.7896	0.8717	0.8286	1761
accuracy			0.8200	3528
macro avg	0.8235	0.8201	0.8196	3528
weighted avg	0.8235	0.8200	0.8195	3528

decision tree test dependent

	precision	recall	f1-score	support
eat	0.7880	0.7067	0.7452	1173
not eat	0.7354	0.8109	0.7713	1179

```
    accuracy                         0.7589      2352
   macro avg     0.7617    0.7588    0.7582      2352
weighted avg     0.7616    0.7589    0.7583      2352

decision tree train independent
              precision    recall  f1-score   support

         eat     0.8485    0.8384    0.8435      1764
     not eat     0.8403    0.8503    0.8453      1764

    accuracy                         0.8444      3528
   macro avg     0.8444    0.8444    0.8444      3528
weighted avg     0.8444    0.8444    0.8444      3528

decision tree test independent
              precision    recall  f1-score   support

         eat     0.7430    0.8333    0.7856      1176
     not eat     0.8103    0.7117    0.7578      1176

    accuracy                         0.7725      2352
   macro avg     0.7766    0.7725    0.7717      2352
weighted avg     0.7766    0.7725    0.7717      2352

neural network train dependent
              precision    recall  f1-score   support

         eat     0.8494    0.8302    0.8397      1767
     not eat     0.8334    0.8524    0.8428      1761

    accuracy                         0.8413      3528
   macro avg     0.8414    0.8413    0.8413      3528
weighted avg     0.8415    0.8413    0.8413      3528

neural network test dependent
              precision    recall  f1-score   support

         eat     0.8300    0.8244    0.8272      1173
     not eat     0.8265    0.8321    0.8292      1179

    accuracy                         0.8282      2352
   macro avg     0.8282    0.8282    0.8282      2352
weighted avg     0.8282    0.8282    0.8282      2352
```

neural network train independent

	precision	recall	f1-score	support
eat	0.8612	0.8333	0.8470	1764
not eat	0.8386	0.8656	0.8519	1764
accuracy			0.8495	3528
macro avg	0.8499	0.8495	0.8495	3528
weighted avg	0.8499	0.8495	0.8495	3528

neural network test independent

	precision	recall	f1-score	support
eat	0.7965	0.8121	0.8042	1176
not eat	0.8083	0.7925	0.8003	1176
accuracy			0.8023	2352
macro avg	0.8024	0.8023	0.8023	2352
weighted avg	0.8024	0.8023	0.8023	2352