INTRODUCTION/NOTES:

For the accompanying R script to work, the right packages has to be available (or installed) and the path to files has to be edited accordingly. This was carried out in R version 3.6.1 (2019-07-05) within RStudio Version 1.2.1335 © 2009-2019 RStudio, Inc. in a MacOS 10.12.6

PHASE 1:

Training and Test data: I generated set of features for each of the users as I did previously in Assignment1, continuing in the same R session I had in Assignment1. I did PCA as in Assignment1 and divided each user data into training and test, in the ratio 60:40. Eating rows were labelled as 1 and the noneating rows as 0. Slight tweak in code to accomplish this which is different from Assignment 1 is shown in the R script named "Project2.R" under the section "PHASE 1: TrainingTestData". Note that though the only code piece for user25 is in Project2.R, the same process was repeated for each of the 30 users. The resulting training and test data for each user is saved as csv files in folder "TrainingTestData". I have confirmed with a TA that this will be accepted.

Code for Project1 from which the last part was modified, as well as the pdf describing the process is attached with this project (enclosed in subfolder "SupportFromProject1"). Since in the first assignment I was allowed to use only one user, I did not previously submit eating and non-eating data (Project1, phase 1) for all the users. I have also added to this "SupportFromProject1" folder, a folder containing all the users' data on eating and non-eating, with the name "AllCSV_emg".

The mean numbers of rows for training is 12618.73 and for testing is 8412.73. Full details of the training and test data for each user are in the table below.

Table 1: Details of Training and Test Data

	Training Data		Test Data	
User	File names	Number of Rows	File names	Number of Rows
9	training09.csv	7435	test09.csv	4957
10	training10.csv	9995	test10.csv	6665
11	training11.csv	12977	test11.csv	8653
12	training12.csv	8663	test12.csv	5777
13	training13.csv	12653	test13.csv	8435
14	training14.csv	12333	test14.csv	8221
16	training16.csv	19711	test16.csv	13141
17	training17.csv	27795	test17.csv	18529
18	training18.csv	16121	test18.csv	10749
19	training19.csv	11943	test19.csv	7961
21	training21.csv	19803	test21.csv	13201
22	training22.csv	11517	test22.csv	7677
23	training23.csv	10861	test23.csv	7241
24	training24.csv	22277	test24.csv	14851
25	training25.csv	5333	test25.csv	3557
26	training26.csv	8219	test26.csv	5481
27	training27.csv	11433	test27.csv	7621
28	training28.csv	11599	test28.csv	7733
29	training29.csv	9029	test29.csv	6019
30	training30.csv	14511	test30.csv	9673
31	training31.csv	13665	test31.csv	9109
32	training32.csv	9767	test32.csv	6511
33	training33.csv	20689	test33.csv	13793
34	training34.csv	9473	test34.csv	6317
36	training36.csv	11255	test36.csv	7503
37	training37.csv	6867	test37.csv	4579
38	training38.csv	12759	test38.csv	8505
39	training39.csv	5477	test39.csv	3653
40	training40.csv	6687	test40.csv	4459
41	training41.csv	17715	test41.csv	11811
Mean		12618.73		8412.73

Selection of 3 Machine Learning Algorithms: I used 3 machine learning algorithms - a) decision trees (using rpart library in R), b) support vector machines (using e1071 library in R), and c) neural networks (using neuralnet library in R).

The correct code for all 3 ML algorithm: The code (along with calculation of precision, recall and F1 score) are under the section "PHASE 1: Decision Tree & SVM" and "PHASE 1: Neural Networks" respectively.

For Decision Tree and SVM, I utilized their respective classification methods. I initially started by doing decision trees, svm and neural networks for each user before the other. But because neural networks was slow, a TA asked me to try finish the trees and svm first before embarking on the neural networks. That is why they are in different sections in the R file.

For training the neural network model, I specifically limited my training to using one hidden layer as most practical problems can be solved with just one hidden layer [1]. For the number of neurons in the hidden layer, I put into consideration the popular "rules of thumb" in Jeff Heaton (2008) - "• The number of hidden neurons should be between the size of the input layer and the size of the output layer. • The number of hidden neurons should be 2/3 the size of the input layer, plus the size of the output layer. • The number of hidden neurons should be less than twice the size of the input layer"[1]. My function for determining number neurons tried to stay within this "rules of thumb", while factoring in the difference in sizes of the training sets.

Reporting of Precision Recall and F1score of all 3 ML algorithm: From the csv files I generated, I subset the relevant columns to get the table of accuracy metrics. The code for this is in section "PHASE 1: Accuracy metrics". The average of accuracies (see table 2) ranged from 0.6165 to 0.6901. The neural networks produced the best average precision and F1 score, while decision trees produced the best average recall. The full details of accuracy metrics of each user is reported in table 2 below.

Table 2: Accuracy Metrics of each User's Data – User Dependent Analysis (DT: Decision tree, SVM: Support Vector Machines and NN: Neural networks)

User	DT Precision	DT Recall	DT F1 Score	SVM Precision	SVM Recall	SVM F1 Score	NN Precision	NN Recall	NN F1 Score
user09	0.8312	0.8826	0.8561	0.9092	0.8688	0.8886	0.9639	0.9149	0.9387
user10	0.8025	0.4574	0.5827	0.9878	0.7521	0.8540	0.9737	0.9238	0.9481
user11	0.9704	0.8569	0.9101	0.9879	0.8491	0.9132	0.9641	0.9424	0.9531
user12	0.4884	0.5623	0.5228	0.4977	0.5918	0.5407	0.5000	0.5568	0.5269
user13	0.5086	0.8041	0.6231	0.5065	0.5798	0.5407	0.5092	0.5257	0.5173
user14	0.5118	0.5075	0.5097	0.5218	0.6341	0.5725	0.5134	0.4672	0.4892
user16	0.9371	0.8452	0.8888	0.9893	0.7855	0.8757	0.9609	0.9169	0.9384
user17	0.9317	0.9325	0.9321	0.9330	0.9407	0.9368	0.9369	0.9555	0.9461
user18	0.5001	0.4831	0.4914	0.5008	0.5374	0.5184	0.4959	0.5644	0.5279
user19	0.5157	0.6367	0.5698	0.5157	0.7410	0.6082	0.5230	0.5314	0.5272
user21	0.5528	0.6009	0.5759	0.5524	0.6609	0.6018	0.5542	0.5950	0.5739
user22	0.4715	0.6115	0.5324	0.4588	0.5995	0.5198	0.4651	0.5347	0.4975
user23	0.5040	0.7318	0.5969	0.4966	0.4442	0.4689	0.5038	0.4906	0.4971
user24	0.5069	0.8389	0.6320	0.5094	0.7995	0.6223	0.5095	0.7581	0.6094
user25	0.5024	0.7115	0.5889	0.4898	0.7795	0.6016	0.4984	0.6305	0.5567
user26	0.4750	0.7179	0.5717	0.4695	0.6058	0.5290	0.4684	0.6051	0.5280
user27	0.4822	0.7963	0.6007	0.4800	0.6667	0.5581	0.4870	0.6570	0.5593
user28	0.5124	0.4058	0.4529	0.5062	0.6674	0.5757	0.4990	0.5882	0.5400
user29	0.4654	0.8471	0.6008	0.4199	0.4513	0.4350	0.4900	0.5211	0.5051
user30	0.5224	0.5935	0.5557	0.5207	0.8387	0.6425	0.5348	0.6297	0.5783
user31	0.5329	0.5990	0.5640	0.5351	0.5740	0.5539	0.5360	0.6012	0.5668
user32	0.4997	0.8916	0.6405	0.5022	0.8329	0.6266	0.5010	0.8092	0.6188
user33	0.4811	0.5899	0.5300	0.4768	0.6356	0.5449	0.4794	0.6557	0.5539
user34	0.9117	0.8467	0.8780	0.9105	0.7863	0.8438	0.9354	0.9218	0.9285
user36	0.9391	0.8270	0.8795	0.8887	0.9152	0.9018	0.9728	0.9336	0.9528
user37	0.4491	0.3735	0.4078	0.4671	0.6081	0.5284	0.4695	0.4740	0.4717
user38	0.4949	0.4718	0.4831	0.4604	0.4508	0.4556	0.4702	0.4487	0.4592
user39	0.8340	0.8335	0.8337	0.9367	0.7371	0.8250	0.9626	0.9173	0.9394
user40	0.8650	0.8394	0.8520	0.9384	0.7725	0.8474	0.9568	0.9448	0.9508
user41	0.4957	0.6083	0.5462	0.5252	0.4325	0.4744	0.4916	0.4708	0.4810
Mean	0.6165	0.6901	0.6403	0.6298	0.6846	0.6468	0.6376	0.6829	0.6560

PHASE 2:

Training and Test data: I concatenated the features of the first 18 users to form the training data and the remaining 12 to form the test data. I then did PCA. The training data consist of 401083 rows and is stored in the folder “TrainingTestData” as "trainingAll.csv". The test data is saved in the same folder and consists of 229803 rows. The code for this is under section “PHASE 2:TrainingTestData” of the “Project2.R” file.

Selection of 3 Machine Learning Algorithms: I used 3 machine learning algorithms - a) decision trees (using rpart library in R), b) support vector machines (using e1071 library in R), and c) neural networks (using neuralnet library in R).

The correct code for all 3 ML algorithm: The code (along with calculation of precision, recall and F1 score) are under the section “PHASE 2: Decision Tree & SVM” and “PHASE 2: Neural Networks” respectively. The algorithms were basically the same for both user-dependent and user-independent analysis, except the neural networks.

It should be noted that the neural networks was very slow because of the size of the training set (over 400,000 rows) for all the first 18 users (user09 to user28) and I wasn’t able to train using the whole set; given the assignment timeframe. I asked the TAs if I could use just 2000 rows per user, taking first 1000 eating and first 1000 non-eating rows (which they agreed) and that is what I used. So under the “PHASE 2: Neural Networks” section, I had to first re-read my csv files that consist of EMG data, take the first 1000 rows, extract features, do PCA, split data (60:40) and then did neural networks. The data used for the neural network training model and test are saved as “nntrainingAll.csv” and “nntestAll.csv” in the folder “TrainingTestData”.

For phase 2, I used 2 hidden layers containing 3 and 5 neurons respectively. Maybe the data being much needed “deeper learning”, because I was not successful with just using one layer as in phase 1.

Reporting of Precision Recall and F1score of all 3 ML algorithm: For this phase, the code is in section "PHASE 2: Accuracy metrics". The accuracy metrics of user independent analysis is reported in table 3 below.

Table 3: Accuracy Metrics for All User's Data – User Independent Analysis

ML Algorithms	Precision	Recall	F1 Score
Decision Trees	0.5177	0.4962	0.5067
Support Vector Machines	0.5155	0.5708	0.5417
Neural networks	0.9771	0.9433	0.9599

Conclusion:

It is interesting to see that the average metrics obtained using the user-dependent analysis was better for decision trees and support vector machines than with the user-independent analysis. However the opposite was the case with neural networks. In the first phase, certain users had better metrics than others and this was consistent across the three algorithms. These differences may be due to the quality of the data, as low quality and mislabelled data may results in defective machine-learning systems [2]. My thoughts are:

1. I had to use the information in the "groundTruth" files to determine the which rows were eating and which rows were non-eating. The question is by multiplying by 100 and dividing by 30, how sure is it that the right rows are properly. Even using that, I discovered that two user data (18 and 25) did not have enough emg rows to match the groundTruth.
2. Error may also have been introduced during data capturing.
3. The metrics did not correlate with data size (data not shown), so the metrics was not size dependent.

References:

1. Jeff Heaton. 2008. Introduction to Neural Networks for Java, 2nd Edition (2nd. ed.). Heaton Research, Inc. pages 157-159
2. Udeshi, S., Jiang, X., & Chattopadhyay, S. (2019). Callisto: Entropy based test generation and data quality assessment for Machine Learning Systems.