CSE 572 Data Mining Assignment – 4

Team bit_miner's

Ankit Anand 1213250712
Rohith Reddy Vajrala 1212908292
Lakshmisagar Kusnoor 1211009498
Divya Prakash Sivakumar 1213204601

We used the PCA and features set from previous assignment as new feature set by multiplying both and divided this new feature set into two parts for each user:
Part 1: Training and
Part 2: Test.
We used 60% of the data for each user as training and the rest of 40% as test data.

We used three types of machines in this assignment:
a) Decision trees,
b) Support vector machines and
c) Neural networks.

We trained each machine with the training data and then used the test data to report accuracy.
Using the accuracy metrics of Precision, Recall, F1 score and ROC reported each metric for every group.

Following are the Formulae's Used:

1. **Accuracy**: is the ratio of correct predictions to total predictions made.
   classification accuracy = correct predictions / total predictions * 100

2. **Accuracy matrix/ Confusion Matrix:** A confusion matrix is a summary of prediction results on a classification problem. It shows the ways in which our classification model is confused when it makes predictions.

Process for calculating an Accuracy Matrix

- We used our test data with expected outcome values.
- Made predictions for each row in our test dataset.
- From the expected outcomes and predictions count:
- The number of correct predictions for each class.
- The number of incorrect predictions for each class, organized by the class that was predicted.
- These numbers are then organized into a matrix with each row of the matrix corresponds to a predicted class and each column of the matrix corresponds to an actual class.
- The counts of correct and incorrect classification are then filled into the matrix.

	Predicted class		
Actual Class		Class = Yes	Class = No
	Class = Yes	True Positive	False Negative
	Class = No	False Positive	True Negative

3. **Precision:** It tells what percent of positive predictions were correct. It is the ratio of correctly predicted positive observations to the total predicted positive observations

Precision = True Positive / True Positive + False Positive

4. **Recall:** Tells what percent of the positive cases did we catch. It is the ratio of correctly predicted positive observations to the all observations in actual class

Recall = True Positive / True Positive + False negative

5. **F1 Score:** F1 Score is the weighted average of Precision and Recall.

F1 Score = 2*(Recall * Precision) / (Recall + Precision)

6. **ROC Curve**: A Receiver Operating Characteristic (ROC) Curve is a way to compare diagnostic tests. It is a plot of the true positive rate against the false positive rate.

7. **AUC:** Area under the curve(AUC) is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one (assuming 'positive' ranks higher than 'negative').

1. **Decision tree (Classification):** A decision tree is a decision support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

   Procedure:

   - We read the Eating and Non- Eating csv files
   - Then we separate the above data as training and testing
   - We pass the train data as parameter to **fitctree** method of MATLAB which returns a classifier. This classifier is further used to predict the scores against the test data.
   - Now that we have eating and non-eating predictions. We calculate True Positive matrix, False Negative matrix, True Negative matrix, False Positive matrix.
   - Using the above matrices, we calculate Precision, Recall, F1 Score and AUC and write the result into a file.

   Classification decision tree gives responses in the leaf node in form of true or false.

   Dimensions of feature matrix:
   Training Data: 90 x 1440 (60% of this)
   Testing Data: 90 x 1440 (40% of this)

   Class-Name (Labels): 1 (Eating actions) and 0 (Non-Eating action)

2. **Support vector machine:** A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples.

   Procedure:

   - We read the Eating and Non- Eating csv files
   - Then we separate the above data as training and testing
   - We pass the train data as parameter to **fitcsvm** method of MATLAB which returns a support vector machine classifier. This classifier is further used to predict the scores against the test data.
   - Now that we have eating and non-eating predictions. We calculate True Positive matrix, False Negative matrix, True Negative matrix, False Positive matrix.
   - Using the above matrices, we calculate Precision, Recall, F1 Score and AUC and write the a result file.

   Dimensions of feature matrix:
   Training Data: 90 x 1440 (60% of this)
   Testing Data: 90 x 1440 (40% of this)

   Class-Name (Labels): 1 (Eating actions) and 0 (Non-Eating action)

3. **Neural Network:** is a learning algorithm that is inspired by the structure and functional aspects of biological neural networks. They are usually used to model complex relationships between inputs and outputs, to find patterns in data, or to capture the statistical structure in an unknown joint probability distribution between observed variables.

   Procedure:

   - We read the Eating and Non- Eating csv files
   - Then we separate the above data with training and testing
   - We pass the train data as parameter to **fitnet** method which has parameter of 10 neurons and trainlm algorithm (trainlm is a network training function that updates weight and bias values according to Levenberg-Marquardt optimization.). Then we train the machine and predict the scores against the test data.
   - Now that we have eating and non-eating predictions. We calculate True Positive matrix, False Negative matrix, True Negative matrix, False Positive matrix.
   - Using the above matrices, we calculate Precision, Recall, F1 Score and AUC and write a result file.

   Dimensions of feature matrix:
   Training Data: 90 x 1440 (60% of this)
   Testing Data: 90 x 1440 (40% of this)

   Class-Name (Labels): 1 (Eating actions) and 0 (Non-Eating action)

==========

Phase 1: User dependent analysis result

We have combined each group's spoon and fork data for both the categories and have randomly selected 60% (random sampling) of the data as train data and the rest was used as test data to report the accuracy metrics such as Precision, Recall, F1 score and ROC for each of the three models, which are Decision Trees (fitctree) and SVM (fitcsvm) and Neural Network (Neural Network Toolbox).

Group	Decision Tree				Neural Network				SVM			
	Precision	Recall	F1	AUC	Precision	Recall	F1	AUC	Precision	Recall	F1	AUC
1	1	0.965517	0.982456	0.985714	0.925926	0.862069	0.892857	0.889286	1	0.967742	0.983607	0.984848
2	0.941176	0.969697	0.955224	0.928922	0.878788	0.878788	0.878788	0.796537	1	1	1	1
3	0.931034	0.9	0.915254	0.9214	0.777778	0.7	0.736842	0.763889	1	1	1	1
4	0.962963	0.866667	0.912281	0.925926	0.83871	0.866667	0.852459	0.841734	1	0.972222	0.985915	0.982143
5	0.857143	1	0.923077	0.928571	0.727273	0.8	0.761905	0.763636	1	1	1	1
6	1	0.969697	0.984615	0.983871	0.892857	0.757576	0.819672	0.832143	1	1	1	1
7	0.884615	0.958333	0.92	0.928794	0.676471	0.958333	0.793103	0.810714	0.962963	1	0.981132	0.981481
8	0.96875	0.96875	0.96875	0.968246	0.96875	0.96875	0.96875	0.968246	1	1	1	1
9	0.777778	1	0.875	0.888889	0.777778	1	0.875	0.888889	1	1	1	1
10	0.933333	0.777778	0.848485	0.84902	0.666667	0.533333	0.592593	0.658333	1	1	1	1
11	1	0.903226	0.949153	0.957143	0.774194	0.774194	0.774194	0.777722	1	1	1	1
12	0.742857	0.866667	0.8	0.784472	0.675	0.870968	0.760563	0.732237	0.928571	0.896552	0.912281	0.914286
13	1	1	1	1	0.333333	0.266667	0.296296	0.391667	1	1	1	1
14	0.807692	0.7	0.75	0.782225	0.75	0.617647	0.677419	0.656187	1	0.914286	0.955224	0.951613
15	0.820513	1	0.901408	0.910256	0.764706	0.8125	0.787879	0.778905	0.961538	0.925926	0.943396	0.953742
16	0.722222	0.928571	0.8125	0.824074	0.714286	0.535714	0.612245	0.702381	0.928571	0.83871	0.881356	0.892857
17	0.925926	0.78125	0.847458	0.865741	0.848485	0.933333	0.888889	0.923935	1	0.965517	0.982456	0.985714
18	0.909091	1	0.952381	0.954545	0.807692	0.7	0.75	0.782225	0.941176	1	0.969697	0.970588
19	0.866667	0.962963	0.912281	0.918182	0.807692	0.777778	0.792453	0.822765	0.966667	1	0.983051	0.983333
20	0.933333	0.933333	0.933333	0.904167	0.742857	0.866667	0.8	0.68961	0.967742	1	0.983607	0.983871
21	0.96875	1	0.984127	0.984375	0.88	0.6875	0.771923	0.808421	1	1	1	1
22	0.787234	0.948718	0.860465	0.831117	0.961538	0.641026	0.769231	0.773774	0.967742	1	0.983607	0.983871

23	0.933333	0.903226	0.918033	0.921212	1	0.806452	0.892857	0.921053	0.933333	1	0.965517	0.966667
24	1	0.9375	0.967742	0.969697	0.774194	0.75	0.761905	0.762097	1	0.971429	0.985507	0.982759
25	1	0.970588	0.985075	0.982759	0.837838	0.885714	0.861111	0.841996	0.909091	1	0.952381	0.954545
26	0.891892	0.970588	0.929577	0.926715	0.740741	0.588235	0.655738	0.675926	0.941176	0.914286	0.927536	0.918864
27	0.83871	0.962963	0.896552	0.896628	0.95	0.678571	0.791667	0.842647	0.95	0.95	0.95	0.959848
28	0.954545	0.75	0.84	0.891907	0.857143	0.75	0.8	0.814286	1	0.9375	0.967742	0.969697
29	1	1	1	1	1	0.911765	0.953846	0.953125	1	1	1	1
30	0.969697	1	0.984615	0.984848	0.863636	0.575758	0.690909	0.613636	0.928571	0.962963	0.945455	0.930952
31	0.64	0.761905	0.695652	0.542222	0.722222	0.541667	0.619048	0.517361	0.95	0.791667	0.863636	0.796429
32	1	1	1	1	0.553191	1	0.712329	0.776596	1	1	1	1
33	0.84375	1	0.915254	0.921875	0.785714	0.611111	0.6875	0.673428	0.972973	1	0.986301	0.986486
Average	**0.903424**	**0.929028**	**0.91275**	**0.911016**	**0.796226**	**0.754812**	**0.76606**	**0.771072**	**0.976064**	**0.969964**	**0.972406**	**0.970745**

Analysis of Phase 1 Result: - We think that in phase 1 since we are training every group with their own new feature set and as the training data is 60% of the feature set, thus the machine is getting overfit and giving very high accuracy and all the other class metrics are also high.

Since the machine is trained well with good percentage of training data from every group's dataset we are getting high classification accuracy results.

For this phase Decision tree and Support Vector machine performed better than Neural Network.

==========

Phase 2: User independent analysis result

We combined the first 10 groups data as the train data both the categories of spoon and fork. The data of the other 23 groups were selected without combining spoon and fork data and used to test the models and report the accuracy metrics

Group	Decision Tree				Neural Network				SVM			
	Precision	Recall	F1	AUC	Precision	Recall	F1	AUC	Precision	Recall	F1	AUC
11	0.65	0.46325	0.4698	0.596186	0.534884	0.575	0.554217	0.531331	0.50236	0.6123	0.5936	0.5311
12	0.666667	0.32973	0.426923	0.639744	0.45945	0.22973	0.306306	0.470639	0.44898	0.455696	0.486486	0.470588
13	0.686275	0.875	0.769231	0.75693	0.575758	0.974359	0.72381	0.706659	0.42663	0.3945 6	0.4102 5	0.4126 3

14	0.657895	0.3125	0.423729	0.599781	0.571429	0.55	0.56051	0.563492	0.506329	0.511111	0.575	0.541176
15	0.61	0.5569	0.433333	0.595763	0.574713	0.625	0.598802	0.576089	0.575949	0.582278	0.575	0.578616
16	0.648649	0.523	0.410256	0.592919	0.48	0.6	0.533333	0.464138	0.462025	0.471264	0.5125	0.491018
17	0.54902	0.469	0.427481	0.531519	0.591549	0.525	0.556291	0.577384	0.54256	0.51364	0.50106	0.49568
18	0.56488	0.556	0.514	0.49898	0.634146	0.325	0.429752	0.586304	0.56962	0.565217	0.65	0.604651
19	0.5625	0.4623	0.492365	0.53125	0.54386	0.3875	0.452555	0.529356	0.443038	0.453488	0.4875	0.46988
20	0.783784	0.367089	0.651246	0.579392	0.833333	0.506329	0.629921	0.634058	0.470085	0.634921	0.506329	0.56338
21	0.621	0.39564	0.46385	0.498	0.512195	0.531646	0.521739	0.509431	0.452229	0.457831	0.481013	0.469136
22	0.794872	0.382716	0.516667	0.689103	0.639535	0.679012	0.658683	0.641685	0.628931	0.634146	0.641975	0.638037
23	0.75	0.675948	0.61235	0.663136	0.607143	0.85	0.708333	0.673137	0.506329	0.511628	0.55	0.53012
24	0.678571	0.59243	0.45236	0.60467	0.55	0.55	0.55	0.544231	0.4956	0.4321	0.4625	0.4836
25	0.653061	0.5236	0.496124	0.604308	0.666667	0.55	0.60274	0.635531	0.490446	0.5	0.475	0.487179
26	0.752641	0.65489	0.59461	0.6574	0.45679	0.4625	0.459627	0.449174	0.64525	0.61235	0.5823	0.52891
27	0.6965	0.6112	0.6523	0.6425	0.416667	0.367647	0.390625	0.417793	0.432836	0.442857	0.455882	0.449275
28	0.67	0.712	0.6945	0.6813	0.551724	0.8	0.653061	0.585386	0.386076	0.408602	0.475	0.439306
29	0.818182	0.675124	0.352941	0.68115	0.549296	0.975	0.702703	0.712148	0.71256	0.61235	0.6547	0.6248
30	0.75	0.657	0.448598	0.543831	0.691358	0.746667	0.717949	0.506393	0.495413	0.66129	0.546667	0.59854
31	0.85	0.7103	0.441558	0.617308	0.823529	0.736842	0.777778	0.691176	0.635294	0.724138	0.736842	0.730435
32	0.632653	0.58465	0.55648	0.591556	0.666667	0.625	0.645161	0.65261	0.43038	0.439024	0.45	0.444444
33	0.6423	0.7512	0.6812	0.756	0.666667	0.55	0.60274	0.637681	0.59631	0.52364	0.54523	0.5136

Average	**0.68215**	**0.558325**	**0.520952**	**0.615336**	**0.59119**	**0.596619**	**0.579854**	**0.578079**	**0.515445**	**0.528454**	**0.537167**	**0.525918**

Analysis of Phase 1 Result: - We think that in phase 2 since we are training the machine with random 10 groups new feature set and testing on each group individually, the accuracy is low, and all the other class metrics are also low. This is because the machine will never be able to define a generalized classification function since the dataset for each group is different even though the feature vectors used are same, this is because each group eating and non-eating "actions/habits" are different and since the training has been done on different groups, it is tough to find a generalized machine.

Since the machine is trained with random 10 groups training data and being tested on each of the different groups individually, we are getting low classification accuracy results.

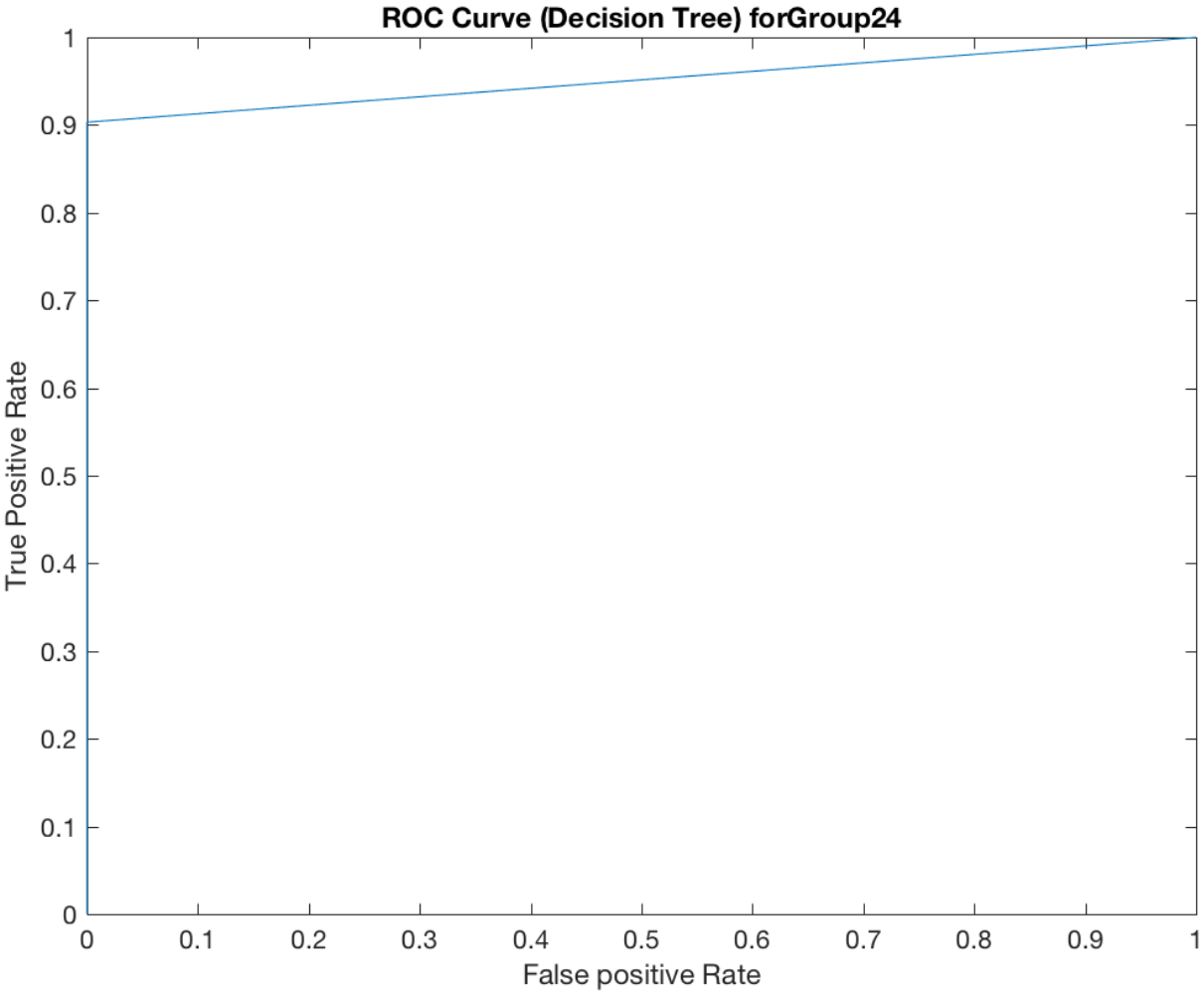
For this phase Decision tree performed better than Neural Network and Support Vector. Overall, we had much better classification accuracy results for Phase 1 then Phase 2.

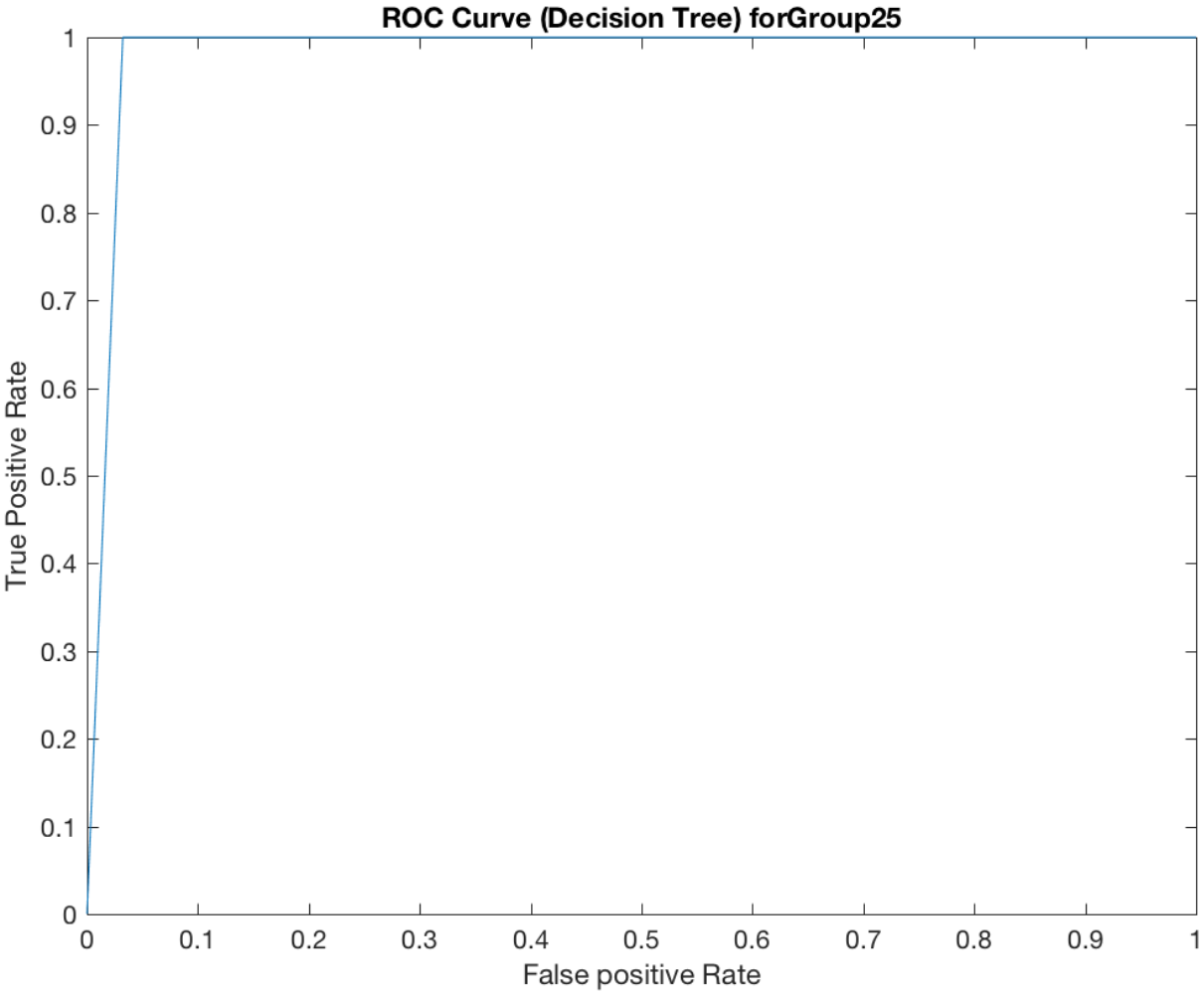
==========

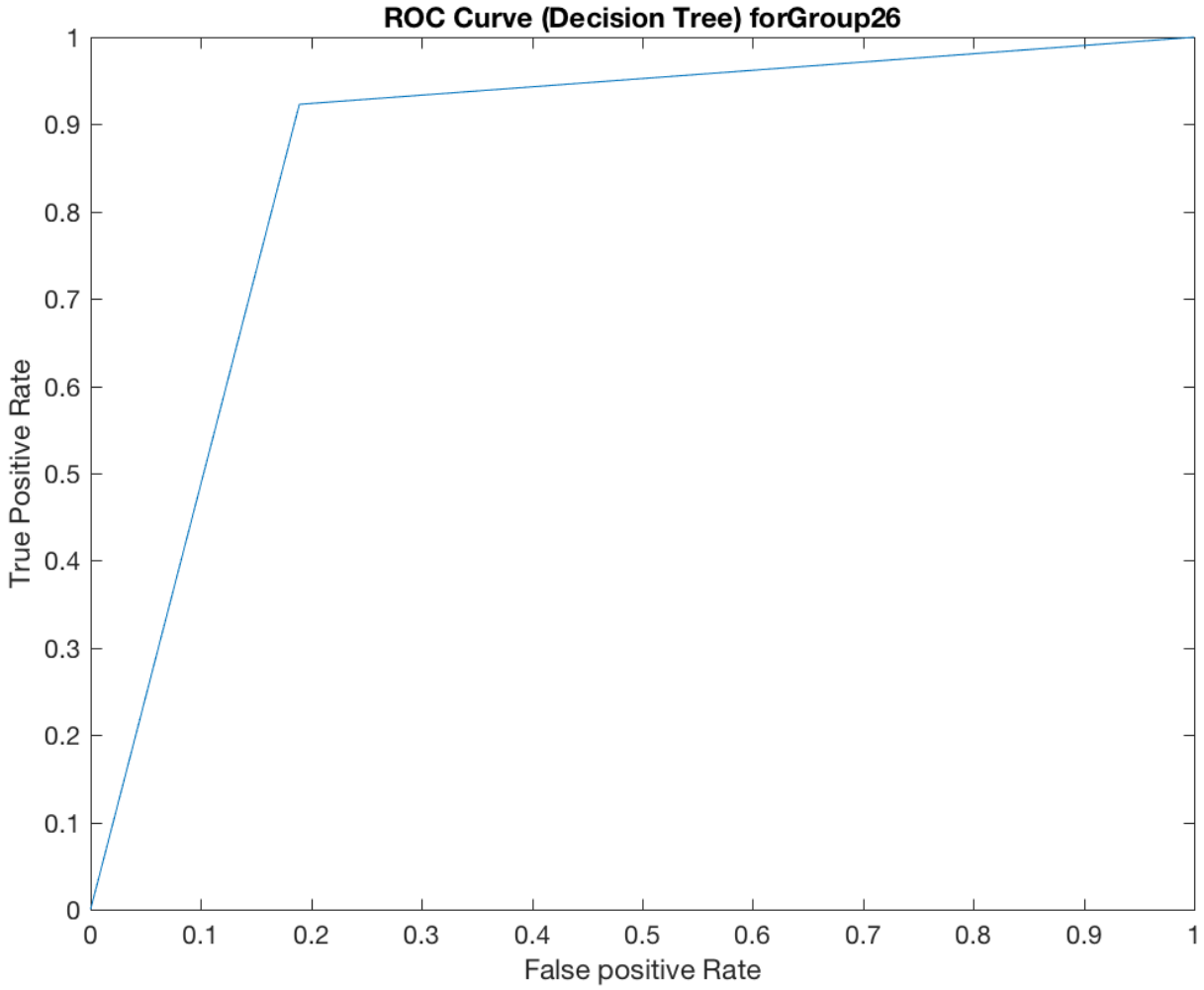
Sample ROC curves:

We have submitted ROC curves for 33 groups using each machine i.e. total 99 ROC curves in the submission zip file inside "ROC_Curves_Phase 1" folder. Their names follow the following pattern: - "RocCurve_"MachineType"_GroupNo."

We have submitted ROC curves for 23 groups using each machine i.e. total 69 ROC curves in the submission zip file inside "ROC_Curves_Phase 2" folder. Their names follow the following pattern: - "RocCurve_"MachineType_task2"_GroupNo."


ROC Curve (Decision Tree) forGroup24
True Positive Rate
False positive Rate
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


ROC Curve (Decision Tree) forGroup25
True Positive Rate
False positive Rate
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1



Execution Steps(Readme):

1. Download the submitted Zip file and extract.
2. Find the final_task1.m file and run the file in MATLAB.
3. After execution completion, you will find the result.mat file which has 33 rows * 12 columns, which has Precision, Recall, F1 and AUC for three model decision tree , SVM and neural network ordered on the basis of group numbers.
4.Find the final_task2.m file for executing task 2 and run this file in MATLAB.
5.After execution you get resultmat_task2.mat file which has 23 rows* 12 columns of result.
6.Roc curves will be saved in the same folder path.

References:

- *Wiki: https://en.wikipedia.org/wiki/Decision_tree 2.*
- *https://en.wikipedia.org/wiki/Artificial_neural_network*
- *https://www.mathworks.com/help/stats/fitctree.html*
- *https://www.mathworks.com/help/stats/fitcsvm.html*
- *https://en.wikipedia.org/wiki/Support_vector_machine*