Project Assignment 4 Report

DATA MINING

CSE 572: Spring 2018

Submitted to:

Professor Ayan Banerjee
Ira A. Fulton School of Engineering
Arizona State University

Submitted by(GROUP 23):

Abhinethri Tumkur Umesh (atumkuru@asu.edu**)**
Aishwarya Mohan (amohan34@asu.edu**)**
Jay Patel (jkpatel5@asu.edu**)**
Madhavi Latha Bodeddula (mbodeddu@asu.edu**)**
Vineesha Kasam (vkasam@asu.edu**)**

May 3, 2018

1. General Procedure

In assignment 1, we have done dimensionality reduction using PCA. So, based on that data we have used data of 10 users for training and other remaining user's data for testing. The new feature matrix that was obtained by multiplying the eigenvector with the old feature matrix was further used for classification. So, we have developed training set by using each gestures data of 10 users and testing set by using each gestures data of other remaining users.

We have attempted to resolve the **class imbalance problem** for proper training of data. For example, for about versus non-about classification, the number of non-about instances were overpowering. Hence we decided to pick only 17 random instances from each of the non-about gestures thereby reducing the number of non-about instances for training.

2. Performance metrics

Before discussing the machine learning algorithms, it is required to know the parameters which are used to measure the performance of machine learning algorithms and how to visualize them.

2.1 Confusion Matrix

This matrix which is also called as error matrix is used to represent the values obtained on test data after predictions, from which the performance of a model can be evaluated.

Below is the sample confusion matrix:

N= Number of Observations	**Predicted -> Yes**	**Predicted -> No**	
Actual -> Yes	**True Positive (TP)**	**False Positive (FP)**	TP + FP
Actual -> No	**False Negative (FN)**	**True Negative(TN)**	FN+TP
	TP+FN	FP+TP	

True Positive (TP): It represents the total number of positive classes which have been classified accurately i.e. the class is '+ve' and it has been classified as '+ve' by the model.

True Negative (TN): It represents the total number of negative classes which have been classified accurately i.e. the class is '-ve' and it has been classified as '-ve' by the model.

False Positive (FP): The class is '-ve' but classified as '+ve' by the model.

False Negative (FN): The class is '+ve' but classified as '-ve' by the model.

2.2 Precision, Recall, F1 and Accuracy

For each action, precision, recall, F1 score and accuracy have been calculated for the interpretation of performance measure using True Positive(TP), False Positive(FP), True Negative(TN) and False Negative(FN).

2.2.1 Precision

It is the ratio of correctly predicted '+ve' classes to total number of classes which are classified as '+ve' classes. If the value of this performance metric is high then it shows that the number of classes which are falsely classified as '+ve' classes is less.
Precision = TP / (TP + FP)

2.2.2 Recall

It is the ratio of correctly predicted '+ve' classes to all the actual '+ve' classes. It tells how well the model was able to classify the positive classes.
Recall = TP / (TP + FN)

2.2.3 F1 Score

It is weighted average of Precision and Recall. It is difficult to understand the concept from the definition but it is helpful when class distribution is uneven.
F1= (2 * Precision * Recall) / Precision + Recall

2.2.4 Accuracy

It is the ratio of correctly classified observations to all the observations and it is the simplest performance measure. Accuracy helps to know whether model is good as it is an intuitive parameter.
Accuracy= (TP + TN) / (TP + FP + FN + TN)

Thus, F1 score helps to evaluate the model better when FP and FN have similar cost but if the costs are different then accuracy helps in better evaluation.

3. Implementation of Machine learning algorithms

Implemented the following machine learning algorithms to classify a particular action among 10 different actions (About, And, Can, Cop, Deaf, Decide, Hearing, Father, Find and Go Out):

- Decision Tree
- Support Vector Machine

- Neural Network

3.1 Decision Tree

Decision tree is built top-down from root node to break down dataset into smaller subsets. Each internal node of decision tree represents a "test" on an attribute and each leaf node represents the classification or decision. In the assignment, we have constructed Decision Tree using matlab function **fitctree**.

Input to fitctree for classifying individual actions:

Training and Test Data:
Dimension of Training data: 306 x 4
Dimension of Test Data: 481 x 4

Model Parameters:
Two parameters are passed:
i) PruneCriterion: impurity
ii) SplitCriterion: deviance

Model Details:
Number of branch nodes: 2
Number of leaf nodes: 3
Decision tree created properties:
Decision cut value for first node: 2.5455e+03
Decision cut value for second node: 1.4238E+03

Few snippets from the code:

```
% Training the data
Total_Train_data = table(dataArray{1:end-1}, 'VariableNames', {'VarName1','VarName2','VarName3','VarName4','VarName5'});
Total_Train_data_svm =  Total_Train_data{1:size(Total_Train_data,1),1:4} ;          % without last column
result = zeros(size(Total_Train_data,1),1);
result(1:120,1)=1;
% Call fitcsvm to train SVM model.
decision_model =  fitctree(Total_Train_data_svm,result,'PruneCriterion','impurity','SplitCriterion','deviance');
```

RESULTS FOR DECISION TREE

3.1.1 About

N= 481	Predicted : Yes	Predicted : No

Actual :Yes	TP = 142	FP = 96
Actual : No	FN = 63	TN = 180

Precision = 0.69268	Recall = 0.59664	F1 = 0.64108	Accuracy = 66.944

3.1.2 And

N= 481	Predicted : Yes	Predicted : No
Actual :Yes	TP = 73	FP = 165
Actual : No	FN = 38	TN = 205

Precision = 0.65766	Recall = 0.30672	F1 = 0.41834	Accuracy = 57.796

3.1.3 Can

N= 481	Predicted : Yes	Predicted : No
Actual :Yes	TP = 166	FP = 72
Actual : No	FN = 11	TN = 232

Precision = 0.93785	Recall = 0.69748	F1 = 0.8	Accuracy = 82.744

3.1.4 Cop

N= 481	Predicted : Yes	Predicted : No
Actual :Yes	TP = 114	FP= 124
Actual : No	FN = 50	TN= 193

Precision = 0.69512	Recall = 0.47899	F1 = 0.56716	Accuracy = 63.825

3.1.5 Deaf

N= 481	Predicted : Yes	Predicted : No
Actual :Yes	TP = 199	FP= 39
Actual : No	FN = 38	TN= 205

Precision = 0.83966	Recall = 0.83613	F1 = 0.83789	Accuracy = 83.992

3.1.6 Decide

N= 481	Predicted : Yes	Predicted : No
Actual :Yes	TP = 160	FP= 78
Actual : No	FN = 30	TN= 213

Precision = 0.84211	Recall = 0.67227	F1 = 0.74766	Accuracy = 77.547

3.1.7 Father

N= 481	Predicted : Yes	Predicted : No
Actual :Yes	TP = 138	FP= 100
Actual : No	FN = 24	TN= 219

Precision = 0.85185	Recall = 0.57983	F1 = 0.69	Accuracy = 74.22

3.1.8 Find

N= 481	Predicted : Yes	Predicted : No
Actual :Yes	TP = 112	FP= 126
Actual : No	FN = 49	TN= 194

Precision = 0.69565	**Recall = 0.47059**	**F1 = 0.5614**	**Accuracy = 63.617**

3.1.9 GoOut

N= 481	**Predicted : Yes**	**Predicted : No**
Actual :Yes	**TP = 84**	**FP = 154**
Actual : No	**FN = 25**	**TN = 218**

Precision = 0.77064	**Recall = 0.35294**	**F1 = 0.48415**	**Accuracy = 62.786**

3.1.10 Hearing

N= 481	**Predicted : Yes**	**Predicted : No**
Actual :Yes	**TP = 118**	**FP = 120**
Actual : No	**FN = 5**	**TN = 238**

Precision = 0.95935	**Recall = 0.4958**	**F1 = 0.65374**	**Accuracy = 74.012**

3.2 Support Vector Machine

This is a supervised learning algorithm to construct the model to classify unlabeled data. In the assignment, we have constructed SVM model using matlab function **fitcsvm** to classify an action from rest of the other actions.

Input to fitcsvm for every action:
Training and Test Data:
Dimension of Training data: 306 x 4
Dimension of Test Data: 481 x 4

Model Parameters:
i) Standardize : True
It standardizes the input predictors before the training begins.
ii) 'Kernel Function' : 'Gaussian'

To classify one particular class using Gaussian Kernel.

iii) BoxConstraint : 1

To specify the pair consisting of BoxConstraint and positive scalar.

iv) outlier fraction: [0,1)

To specify the expected outliers present in training data.

We have trained the model with the above model parameters and the test data has been fed into the trained model to label the unknown records. The prediction output has been classified based on the action (For example, the predicted values for all the actual 'About' classes are grouped together and remaining all other actions as another group) and this is used to calculate TP, FP, TN and FN. After calculating the above parameters, four performance metrics i.e. Precision, Recall, Fl score and Accuracy are calculated.

Snippets from the code:

```
% Training the data
Total_Train_data=table(dataArray{1:end-1}, 'VariableNames', {'VarName1', 'VarName2', 'VarName3', 'VarName4', 'VarName5'});
Total_Train_data_svm =  Total_Train_data{1:size(Total_Train_data,1),1:4} ;
% without last column
result = zeros(size(Total_Train_data,1),1);
result(1:120,1)=1;
% Call fitcsvm to train SVM model.
svm_model=
fitcsvm(Total_Train_data_svm,result,'standardize',true,'KernelFunction','gaussian','BoxConstraint',1,'OutlierFraction',0.10);
```

TEST RESULTS FOR SVM:

3.2.1 About

N= 481	Predicted : Yes	Predicted : No
Actual :Yes	TP = 137	FP = 101
Actual : No	FN = 45	TN = 198

Precision = 0.75275	Recall = 0.57563	F1 = 0.65238	Accuracy = 69.647

3.2.2 And

N= 481	Predicted : Yes	Predicted : No
Actual :Yes	TP = 87	FP = 151

Actual : No	**FN = 34**	**TN = 209**

Precision = 0.71901	**Recall = 0.36555**	**F1 = 0.48468**	**Accuracy = 61.538**

3.2.3 Can

N= 481	**Predicted : Yes**	**Predicted : No**
Actual :Yes	**TP = 116**	**FP = 122**
Actual : No	**FN = 6**	**TN = 237**

Precision = 0.95082	**Recall = 0.48739**	**F1 = 0.64444**	**Accuracy = 73.389**

3.2.4 Cop

N= 481	**Predicted : Yes**	**Predicted : No**
Actual :Yes	**TP = 77**	**FP = 161**
Actual : No	**FN = 52**	**TN = 191**

Precision =0.5969	**Recall = 0.69748**	**F1 = 0.41962**	**Accuracy = 55.717**

3.2.5 Deaf

N= 481	**Predicted : Yes**	**Predicted : No**
Actual :Yes	**TP = 166**	**FP = 72**
Actual : No	**FN = 50**	**TN = 193**

Precision = 0.76852	**Recall = 0.59664**	**F1 = 0.73128**	**Accuracy = 74.636**

3.2.6 Decide

N= 481	Predicted : Yes	Predicted : No
Actual :Yes	TP = 72	FP = 166
Actual : No	FN = 34	TN = 209

Precision = 0.67925	Recall = 0.59664	F1 = 0.64108	Accuracy = 66.944

3.2.7 Father

N= 481	Predicted : Yes	Predicted : No
Actual :Yes	TP = 32	FP = 206
Actual : No	FN = 11	TN = 232

Precision = 0.74419	Recall = 0.13445	F1 = 0.13445	Accuracy = 54.886

3.2.8 Find

N= 481	Predicted : Yes	Predicted : No
Actual :Yes	TP = 98	FP = 140
Actual : No	FN = 26	TN = 217

Precision = 0.79032	Recall = 0.41176	F1 = 0.5122	Accuracy = 65.489

3.2.9 GoOut

N= 481	Predicted : Yes	Predicted : No
Actual :Yes	TP = 84	FP = 154
Actual : No	FN = 6	TN = 237

Precision = 0.93333	Recall = 0.35294	F1 = 0.64108	Accuracy = 66.736

3.2.10 Hearing

N= 481	Predicted : Yes	Predicted : No
Actual :Yes	TP = 88	FP = 150
Actual : No	FN = 5	TN = 238

Precision = 0.94624	Recall = 0.36975	F1 = 0.53172	Accuracy = 67.775

3.3 Neural Network

A Neural Network contains an input layer, a set of hidden layers and an output layer with varying number of neurons in each layer (input, output, hidden) depending on the application. In the assignment, we use the **nntool** of MATLAB and the output layer consists of only one neuron, which signifies the classification of each action (Eg: About or Not About).

Input to patternnet:
Patternet (Pattern recognition networks) classify the inputs given to them according to the target classes specified, which are the feed-forward networks. The input to the patternnet is the number of neurons in the hidden layer.

Training and Test Data:
Dimension of Training data: 306 x 4
Dimension of Test Data: 481 x 4

Model Parameters:
Two parameters are passed:
i) TrainingFunction: ‘traingda’
ii) Performance Function: ‘mse'

Model Details:
Number of neurons in input layer - 4
Number of neurons in hidden layer - 15
Number of neurons in the output layer - 1

Number of layers (input, hidden, output) - 1

Below is the snippet from the code:

```
% Training the data
Total_Train_data = table(dataArray{1:end-1}, 'VariableNames', {'VarName1','VarName2','VarName3','VarName4','VarName5'});
Total_Train_data_neural =  Total_Train_data{1:size(Total_Train_data,1),1:4} ;          % without last column
result = zeros(size(Total_Train_data,1),1);
result(1:120,1)=1;
hiddenLayerSize = 15; % Number of neurons in hidden layer
net = patternnet(hiddenLayerSize); % pattern recognition network
net.trainFcn = 'traingda';
net.performFcn = 'mse';
neural_model = train(net,transpose(Total_Train_data_neural),transpose(result));
```

RESULTS FOR NEURAL NETWORK:

3.3.1 About

N= 481	Predicted : Yes	Predicted : No
Actual :Yes	TP = 168	FP = 70
Actual : No	FN = 72	TN = 171

Precision = 0.7	Recall = 0.70588	F1 = 0.70293	Accuracy = 70.478

3.3.2 And

N= 481	Predicted : Yes	Predicted : No
Actual :Yes	TP = 75	FP = 163
Actual : No	FN = 35	TN = 208

Precision = 0.68182	Recall = 0.31513	F1 = 0.43103	Accuracy = 58.836

3.3.3 Can

N= 481	Predicted : Yes	Predicted : No
Actual :Yes	TP = 146	FP = 92
Actual : No	FN = 12	TN = 231

Precision = 0.92405	**Recall = 0.61345**	**F1 = 0.73737**	**Accuracy = 78.378**

3.3.4 Cop

N= 481	**Predicted : Yes**	**Predicted : No**
Actual :Yes	**TP = 90**	**FP = 148**
Actual : No	**FN = 127**	**TN = 116**

Precision = 0.41475	**Recall = 0.37815**	**F1 = 0.3956**	**Accuracy = 42.827**

3.3.5 Deaf

N= 481	**Predicted : Yes**	**Predicted : No**
Actual :Yes	**TP = 222**	**FP = 16**
Actual : No	**FN = 59**	**TN = 184**

Precision = 0.79004	**Recall = 0.93277**	**F1 = 0.85549**	**Accuracy = 84.407**

3.3.6 Decide

N= 481	**Predicted : Yes**	**Predicted : No**
Actual :Yes	**TP = 128**	**FP = 110**
Actual : No	**FN = 17**	**TN = 226**

Precision = 0.88276	**Recall = 0.53782**	**F1 = 0.66841**	**Accuracy = 73.597**

3.3.7 Father

N= 481	**Predicted : Yes**	**Predicted : No**

Actual :Yes	TP = 54	FP = 184
Actual : No	FN = 33	TN = 210

Precision = 0.62069	Recall = 0.22689	F1 = 0.33231	Accuracy = 54.886

3.3.8 Find

N= 481	Predicted : Yes	Predicted : No
Actual :Yes	TP = 109	FP = 129
Actual : No	FN = 49	TN = 194

Precision = 0.68987	Recall = 0.45798	F1 = 0.55051	Accuracy = 62.994

3.3.9 GoOut

N= 481	Predicted : Yes	Predicted : No
Actual :Yes	TP = 108	FP = 130
Actual : No	FN = 19	TN = 224

Precision = 0.85039	Recall = 0.45378	F1 = 0.59178	Accuracy = 69.023

3.3.10 Hearing

N= 481	Predicted : Yes	Predicted : No
Actual :Yes	TP = 94	FP = 144
Actual : No	FN = 7	TN = 236

Precision = 0.93069	Recall = 0.39496	F1 = 0.55457	Accuracy = 68.607