EMG Data Classification Report

B S N V Chaitanya
S20160020115

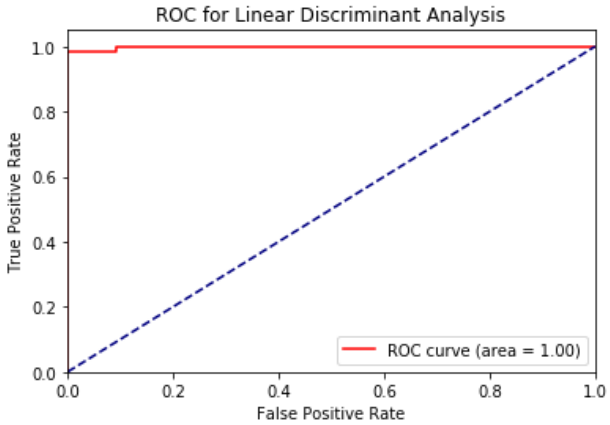
1. The objective of the assignment is to categorize the given data into 'Normal' and 'Aggressive'
2. Dataset :
   The data set consists of EMG signals collected on S = 4 subjects performing C = 20 physical activities, 10 being normal and the 10 being aggressive. Each subject repeats the physical actions R = 15 times. The overall number of electrodes is 8, which corresponds to 8 input time series one for a muscle channel (ch1-8). Each time series contains ~10000 samples.
3. Feature Extraction :
   Each text file is segmented into 15 segments, summing up to 1200 segment(i.e., examples in total). From each text file(which are 40 altogether), features are extracted and are concatenated to a npArray of dimension (1200,180) and target classes are concatenated to another array of size (1200,). 1200 are the total examples and 180 are the total features.
4. Feature Extracted are :
   a. Time domain statistics (TDS) : mean, variance, skewness, kurtosis
   b. Inter-channel statistics (ICS) : maximum of the cross-correlation values between few pairs of the channels.
   c. Log moments of Fourier spectra (LMF)

TDS -> 4* 8 = 32 features
ICS -> 12 features (12 pairs of channels considered)
LMF -> 17 * 8 = 136 features
Total = 180 features per a segment

5. Classifiers used : Linear Discriminant Analysis, Quadratic Discriminant Analysis, SVM with polynomial kernel, SVM with Gaussian kernel, K-Nearest Neighbours

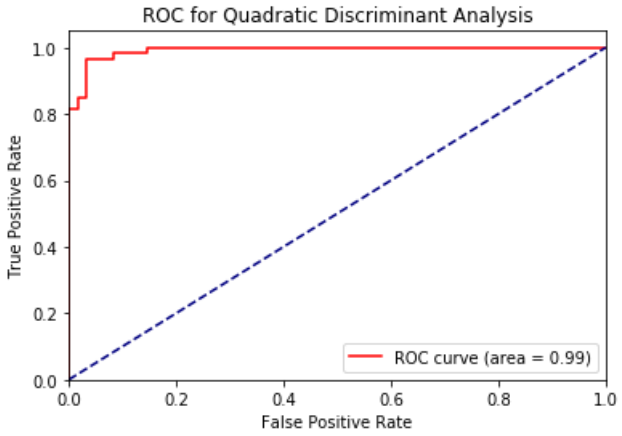
   a. Linear Discriminant Analysis:

```
Accuracy: 0.9666666666666667
Confusion matrix :[[61  3]
                  [ 1 55]]
F1 score: 0.9649122807017544
```



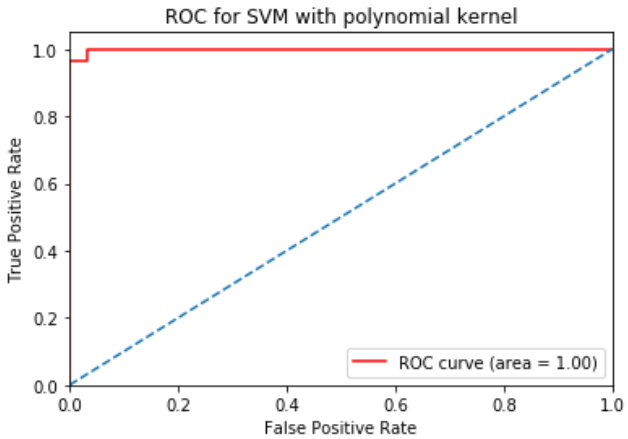
b. Quadratic Discriminant Analysis

```
Accuracy: 0.9416666666666667
Confusion Matrix :[[56  5]
                  [ 2 57]]
F1 score: 0.9421487603305785
```



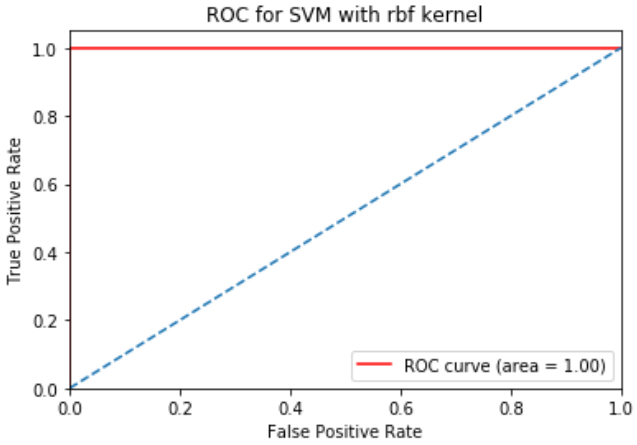
c. SVM with polynomial kernel

```
Accuracy: 0.975
Confusion matrix : [[58  2]
                   [ 1 59]]
F1 score: 0.9752066115702478
```



d. SVM with gaussian kernel

```
Accuracy: 1.0
Confusion matrix : [[52  0]
                    [ 0 68]]
F1 score: 1.0
```



e. K-Nearest Neighbours

```
Accuracy: 0.9916666666666667
Confusion matrix : [[65  0]
                    [ 1 54]]
F1 score: 0.9908256880733944
```

6. Comparison of classifiers : Since the given task is binary classification, all the tried classifiers gave almost near accuracies. SVM with Gaussian kernel gave best accuracy of 100% while QDA was the least with 94%.
7. Code for SVM for Gaussian kernel:

```
############################## SVM with rbf kernel
#########################
kf = KFold(n_splits=10)
temp = 0
for train, test in kf.split(features_sub):
    scaler = StandardScaler()
    features_sub_norm = scaler.fit_transform(features_sub[train])
    g_svm = svm.SVC(C=10, gamma='scale')
    g_svm.fit(features_sub_norm, Y[train])
    features_sub_norm_test = scaler.transform(features_sub[test])
    acc = g_svm.score(features_sub_norm_test, Y[test])
    if(acc > temp):
        temp = acc
        G_SVM = g_svm
        test_data = [features_sub_norm_test, Y[test]]

g_svm_prd = G_SVM.predict(test_data[0])
```

```
print("Accuracy: %s" % (G_SVM.score(test_data[0], test_data[1])))
print(confusion_matrix(test_data[1], g_svm_prd))
print("F1 score: %s" % f1_score(test_data[1], g_svm_prd))


# ROC curve and finding area under the curve
y_score = G_SVM.decision_function(test_data[0])
fpr, tpr, thr = roc_curve(test_data[1], y_score)
roc_auc = auc(fpr, tpr)
# Plotting the curve
plt.figure()
plt.plot(fpr, tpr, color='red',label='ROC curve (area = %0.2f)' % roc_auc)
plt.plot([0, 1], [0, 1], linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC for SVM with rbf kernel')
plt.legend(loc="lower right")
plt.show()
```