EMG Data Classification - Report

Sumanth Doddapaneni
S20160020125

Dataset Description:

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 (~15 actions per experimental session for each subject).

Number of Instances: ~10,000
Number of Attributes: 8

Number of Classes: 20
The dataset consists of 10 normal, and 10 aggressive physical actions.
Normal: Bowing, Clapping, Handshaking, Hugging, Jumping, Running, Seating, Standing, Walking, Waving
Aggressive: Elbowing, Frontkicking, Hammering, Headering, Kneeing, Pulling, Punching, Pushing, Sidekicking, Slapping

Features Computed:

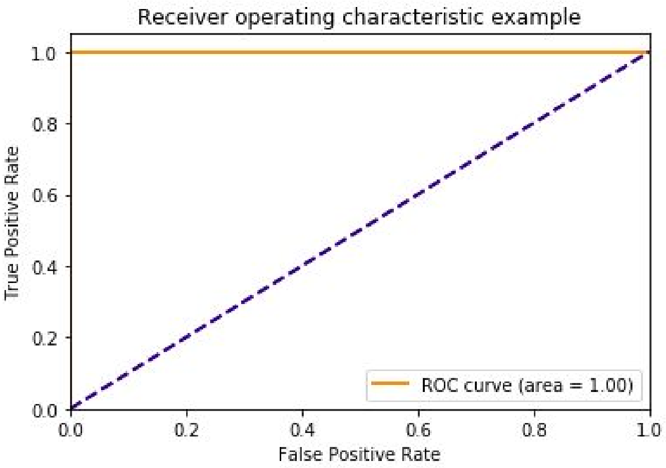
- Mean
- Median
- Variance
- Skewness
- Kurtosis
- Log Moments
- Cross-Correlation

Classifier Results:

1. **Linear SVM**
   Accuracy: 100%

   Confusion Matrix:
   ```
   [[134   0]
    [  0 119]]
   ```

ROC Curve:

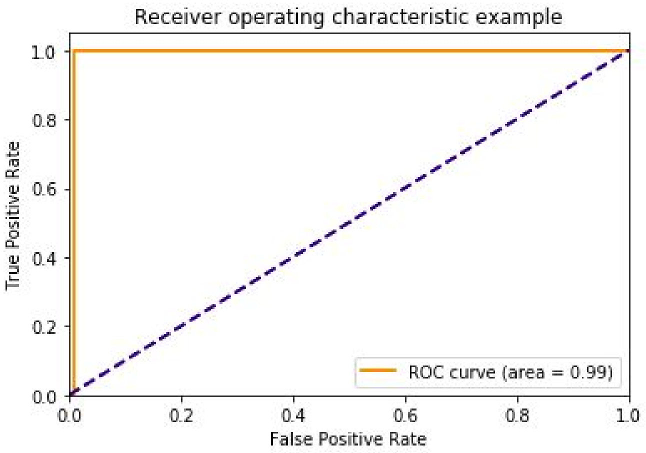


2. **Polynomial SVM (degree = 2):**
   Accuracy: 99.2%

   Confusion Matrix:

```
[[133   1]
 [  1 118]]
```

   ROC Curve:

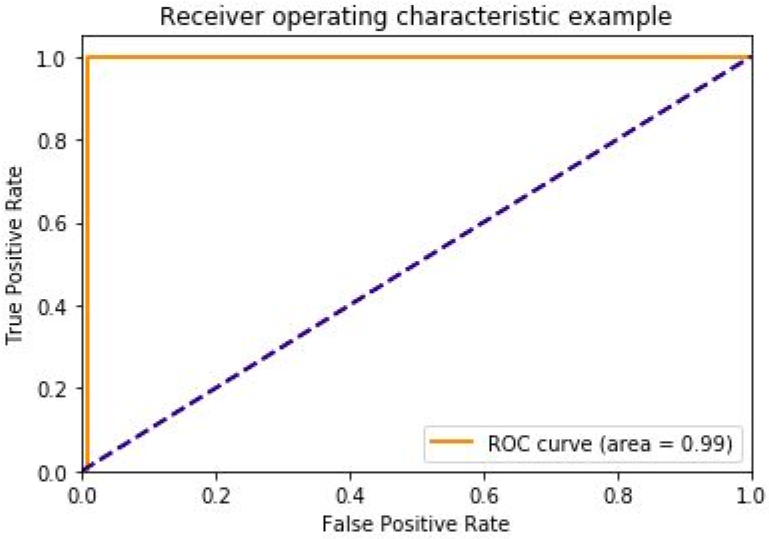


3. **Polynomial SVM (degree = 3):**
   Accuracy: 98.8%

   Confusion Matrix:

```
[[133   1]
 [  2 117]]
```

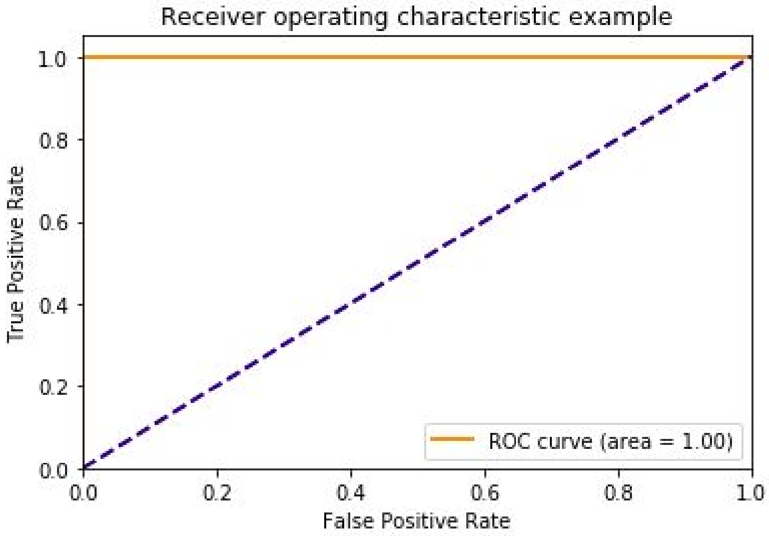
ROC Curve:



4. **Linear Discriminant Analysis:**
   Accuracy: 100%
   Confusion Matrix:

```
[[134   0]
 [  0 119]]
```

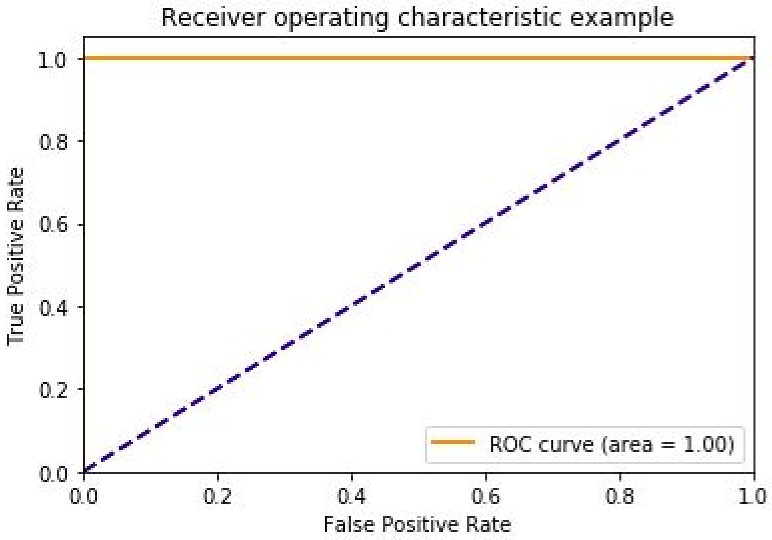
ROC Curve:



5. **Quadratic Discriminant Analysis:**
   Accuracy: 100%
   Confusion Matrix:

```
[[134   0]
 [  0 119]]
```

   ROC Curve:



6. **KNeighborsClassifier**
   Accuracy: 99.6%

   Confusion Matrix:

```
[[133   1]
 [  0 119]]
```

All the classifiers performed well on the data. The data being linearly separable the best accuracies were given by linear classifiers.

Code for Linear SVM

```
from sklearn.svm import SVC

# training and testing part
clf = SVC(gamma='auto', kernel=’linear’)
clf.fit(X_train, Y_train)
y_pred = clf.predict(X_test)
```

```
acc = accuracy_score(Y_test, y_pred)

# ROC curve and finding area under the curve
y_score = clf.decision_function(X_test)
fpr, tpr, _ = roc_curve(Y_test, y_score)
roc_auc = auc(fpr, tpr)

# Plotting the curve
plt.figure()
lw = 2
plt.plot(fpr, tpr, color='darkorange',
         lw=lw, label='ROC curve (area = %0.2f)' % roc_auc)
plt.plot([0, 1], [0, 1], color='navy', lw=lw, 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('Receiver operating characteristic example')
plt.legend(loc="lower right")
plt.show()
plt.savefig('roc', facecolor='w', edgecolor='w',
        orientation='portrait')

# Confusion matrix
print(confusion_matrix(Y_test, y_pred))
```