```python
import pandas as pd
from sklearn.ensemble import AdaBoostClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.metrics import accuracy_score, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt
import re
import numpy as np
```

```python
from google.colab import files
data = files.upload()
```

Elegir archivos jdf.csv

- **jdf.csv**(text/csv) - 1418182 bytes, last modified: 16/11/2022 - 100% done

```
Saving jdf.csv to jdf.csv
```

```python
data = pd.read_csv('jdf.csv')
data = data.drop(['Group', 'Session', 'Subject', 'Type', 'Sample', "RA", "RRA", "GRA"], axis=1)
#data['Type'] = data['Type'].str.extract('(\d+)').astype(int)

#data['Type'] = pd.to_numeric(data['Type'])
#data['ISI'] = pd.to_numeric(data['ISI'])
data
```

```
      EMGPeakToPeak  ISI   Label      _RRA       QRA   GaussRA
   0       129.9786   -1  HC SWD  1.055555  1.246736  1.243814
   1       123.1377   -1  HC SWD  1.000000  1.181119  1.178351
   2        79.8115   -1  HC SWD  0.648149  0.765541  0.763746
   3       115.1565   -1  HC SWD  0.935185  1.104565  1.101976
   4       189.2671   -1  HC SWD  1.537037  1.815423  1.811168
 ...            ...  ...     ...       ...       ...       ...
9115       166.4639   20  HC SWD  1.239038  1.667441  1.660530
9116       149.3614   20  HC SWD  1.111740  1.496128  1.489927
9117       103.7549   20  HC SWD  0.772277  1.039296  1.034988
9118       190.4073   20  HC SWD  1.417256  1.907278  1.899373
9119       294.1622   20  HC SWD  2.189533  2.946574  2.934360

9120 rows × 6 columns
```

```python
X = data.drop(['Label', '_RRA', 'QRA', 'GaussRA'], axis=1)
y = data['Label']

```

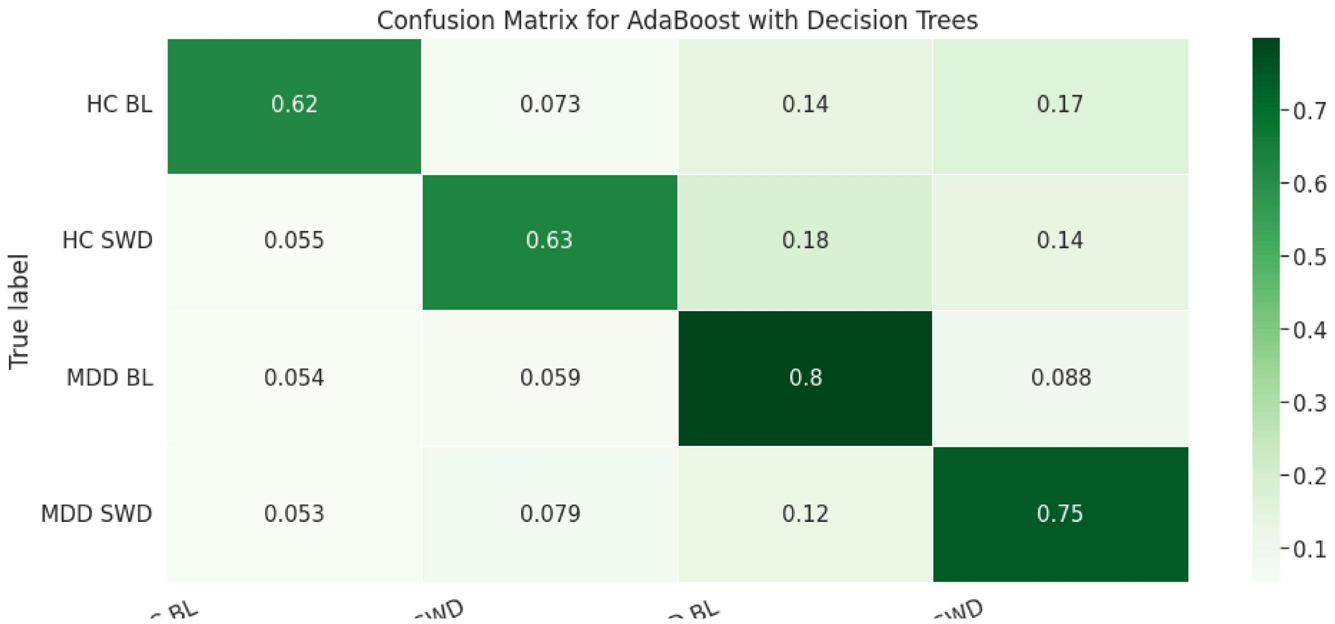
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=42, stratify=y)

bag_clf = AdaBoostClassifier(DecisionTreeClassifier(max_depth=8), n_estimators=20, algorithm='SAMME.R',
bag_clf.fit(X_train, y_train)
y_pred = bag_clf.predict(X_test)
acc = accuracy_score(y_test, y_pred)
acc
```

```
0.3393640350877193
```

```
X = data.drop(['Label', 'QRA', 'GaussRA'], axis=1)
y = data['Label']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=42, stratify=y)

bag_clf = AdaBoostClassifier(DecisionTreeClassifier(max_depth=8), n_estimators=500, algorithm='SAMME.R'
bag_clf.fit(X_train, y_train)
y_pred = bag_clf.predict(X_test)
acc = accuracy_score(y_test, y_pred)
acc
```

```
0.7160087719298246
```

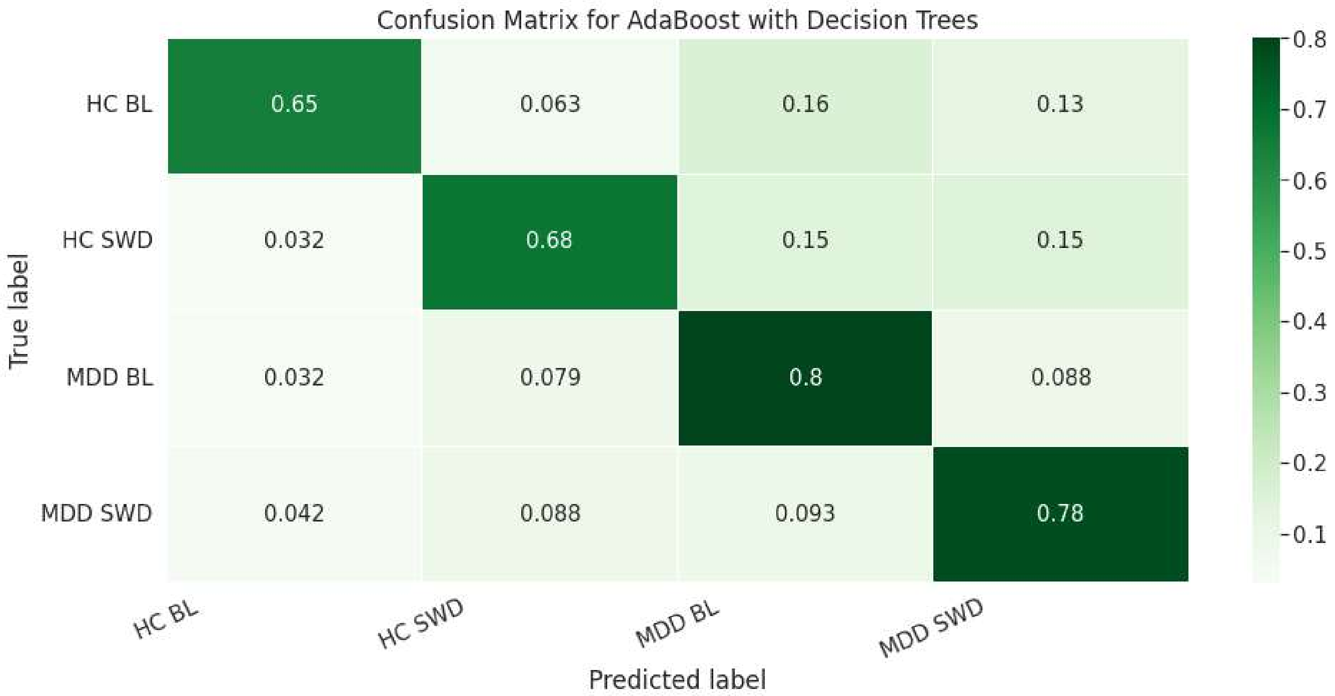
```
matrix = confusion_matrix(y_test, y_pred)
matrix = matrix.astype('float') / matrix.sum(axis=1)[:, np.newaxis]

# Build the plot
plt.figure(figsize=(16,7))
sns.set(font_scale=1.4)
sns.heatmap(matrix, annot=True, annot_kws={'size':15},
            cmap=plt.cm.Greens, linewidths=0.2)

# Add labels to the plot
class_names = ['HC BL', 'HC SWD', 'MDD BL',
               'MDD SWD']
tick_marks = np.arange(len(class_names))
tick_marks2 = tick_marks + 0.5
plt.xticks(tick_marks, class_names, rotation=25)
plt.yticks(tick_marks2, class_names, rotation=0)
plt.xlabel('Predicted label')
plt.ylabel('True label')
plt.title('Confusion Matrix for AdaBoost with Decision Trees')
plt.show()
```



```
X = data.drop(['Label', 'GaussRA', '_RRA' ], axis=1)
y = data['Label']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=42, stratify=y)

bag_clf = AdaBoostClassifier(DecisionTreeClassifier(max_depth=8), n_estimators=500, algorithm='SAMME.R'
bag_clf.fit(X_train, y_train)
y_pred = bag_clf.predict(X_test)
acc = accuracy_score(y_test, y_pred)
acc
```

```
0.7412280701754386
```

```
matrix = confusion_matrix(y_test, y_pred)
matrix = matrix.astype('float') / matrix.sum(axis=1)[:, np.newaxis]

# Build the plot
plt.figure(figsize=(16,7))
sns.set(font_scale=1.4)
sns.heatmap(matrix, annot=True, annot_kws={'size':15},
            cmap=plt.cm.Greens, linewidths=0.2)

# Add labels to the plot
class_names = ['HC BL', 'HC SWD', 'MDD BL',
               'MDD SWD']
tick_marks = np.arange(len(class_names))
tick_marks2 = tick_marks + 0.5
plt.xticks(tick_marks, class_names, rotation=25)
plt.yticks(tick_marks2, class_names, rotation=0)
plt.xlabel('Predicted label')
plt.ylabel('True label')
plt.title('Confusion Matrix for AdaBoost with Decision Trees')
plt.show()
```



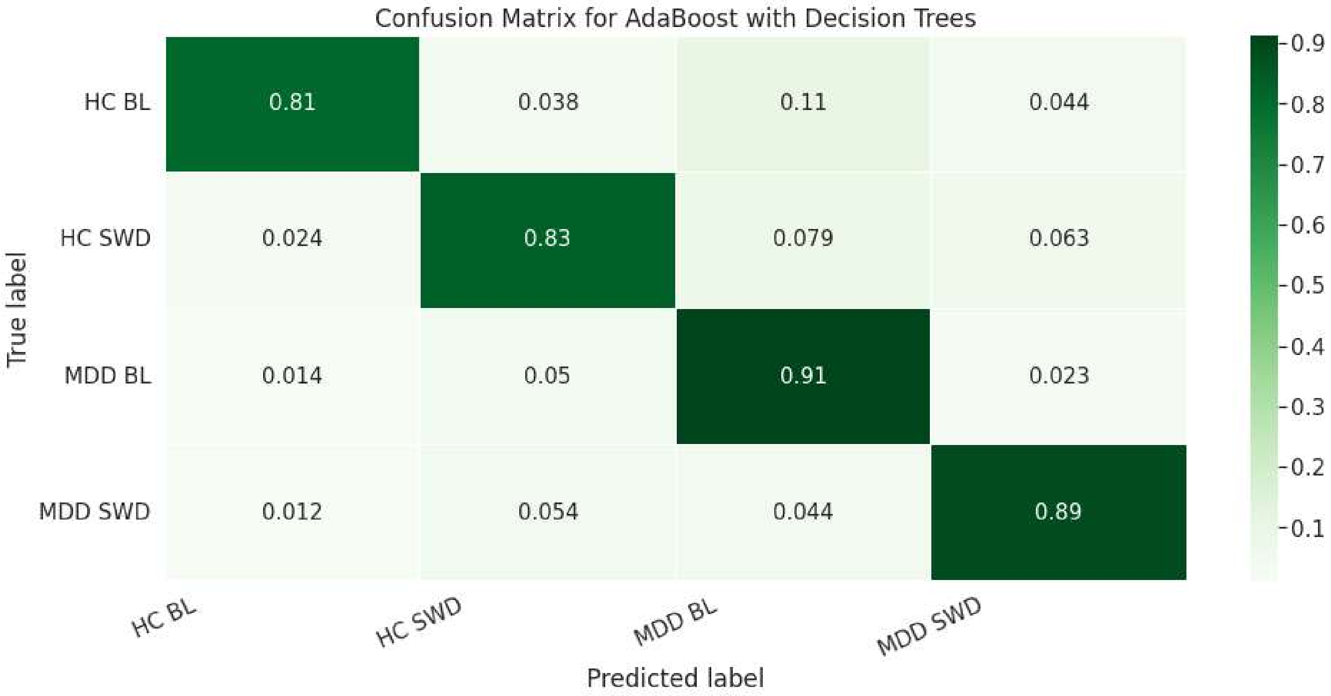
Confusion Matrix for AdaBoost with Decision Trees

True label \ Predicted label	HC BL	HC SWD	MDD BL	MDD SWD
HC BL	0.65	0.063	0.16	0.13
HC SWD	0.032	0.68	0.15	0.15
MDD BL	0.032	0.079	0.8	0.088
MDD SWD	0.042	0.088	0.093	0.78

```
X = data.drop(['Label', 'GaussRA'], axis=1)
y = data['Label']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=42, stratify=y)

bag_clf = AdaBoostClassifier(DecisionTreeClassifier(max_depth=8), n_estimators=500, algorithm='SAMME.R'
bag_clf.fit(X_train, y_train)
y_pred = bag_clf.predict(X_test)
acc = accuracy_score(y_test, y_pred)
acc
```

```
0.8711622807017544
```

```
matrix = confusion_matrix(y_test, y_pred)
matrix = matrix.astype('float') / matrix.sum(axis=1)[:, np.newaxis]

# Build the plot
plt.figure(figsize=(16,7))
sns.set(font_scale=1.4)
sns.heatmap(matrix, annot=True, annot_kws={'size':15},
            cmap=plt.cm.Greens, linewidths=0.2)

# Add labels to the plot
class_names = ['HC BL', 'HC SWD', 'MDD BL',
               'MDD SWD']
tick_marks = np.arange(len(class_names))
tick_marks2 = tick_marks + 0.5
plt.xticks(tick_marks, class_names, rotation=25)
plt.yticks(tick_marks2, class_names, rotation=0)
plt.xlabel('Predicted label')
plt.ylabel('True label')
plt.title('Confusion Matrix for AdaBoost with Decision Trees')
plt.show()
```

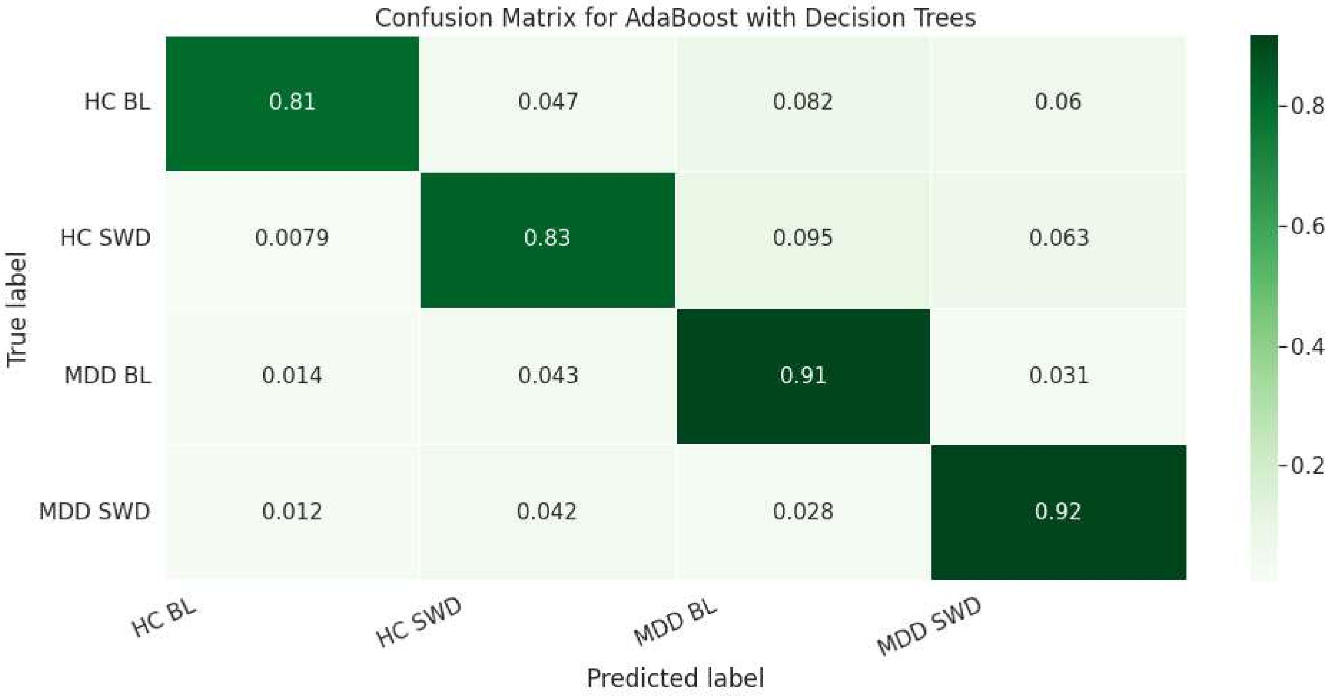


```
X = data.drop(['Label'], axis=1)
y = data['Label']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=42, stratify=y)

bag_clf = AdaBoostClassifier(DecisionTreeClassifier(max_depth=8), n_estimators=500, algorithm='SAMME.R'
bag_clf.fit(X_train, y_train)
y_pred = bag_clf.predict(X_test)
acc = accuracy_score(y_test, y_pred)
acc
```

```
0.8799342105263158
```

```
matrix = confusion_matrix(y_test, y_pred)
matrix = matrix.astype('float') / matrix.sum(axis=1)[:, np.newaxis]

# Build the plot
plt.figure(figsize=(16,7))
sns.set(font_scale=1.4)
sns.heatmap(matrix, annot=True, annot_kws={'size':15},
            cmap=plt.cm.Greens, linewidths=0.2)
```

```

# Add labels to the plot
class_names = ['HC BL', 'HC SWD', 'MDD BL',
               'MDD SWD']
tick_marks = np.arange(len(class_names))
tick_marks2 = tick_marks + 0.5
plt.xticks(tick_marks, class_names, rotation=25)
plt.yticks(tick_marks2, class_names, rotation=0)
plt.xlabel('Predicted label')
plt.ylabel('True label')
plt.title('Confusion Matrix for AdaBoost with Decision Trees')
plt.show()
```



✓ 0s completed at 9:29 PM ● ✕