```
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
```

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

Elegir archivos df.csv

- **df.csv**(text/csv) - 905077 bytes, last modified: 16/11/2022 - 100% done

```
Saving df.csv to df (2).csv
```

```
data = pd.read_csv('df (2).csv')
data = data.drop(['Group', 'Session', 'Subject', 'Type', 'Sample'], 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	RRA	RA	GRA	Label
0	307.8442	10	2.500001	2.984275	3.015745	HC SWD
1	307.8442	0	2.500001	2.984275	3.015745	HC SWD
2	202.9491	8	1.648148	1.967411	1.988158	HC SWD
3	112.8762	0	0.916667	1.094234	1.105773	HC SWD
4	129.9786	-1	1.055555	1.260027	1.273314	HC SWD
...	...	...	...	...	...	...
9115	299.8630	0	2.231966	3.025579	3.053667	HC SWD
9116	294.1622	20	2.189533	2.968059	2.995612	HC SWD
9117	294.1622	0	2.189533	2.968059	2.995612	HC SWD
9118	196.1081	4	1.459689	1.978706	1.997075	HC SWD
9119	196.1081	0	1.459689	1.978706	1.997075	HC SWD

9120 rows × 6 columns

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

```

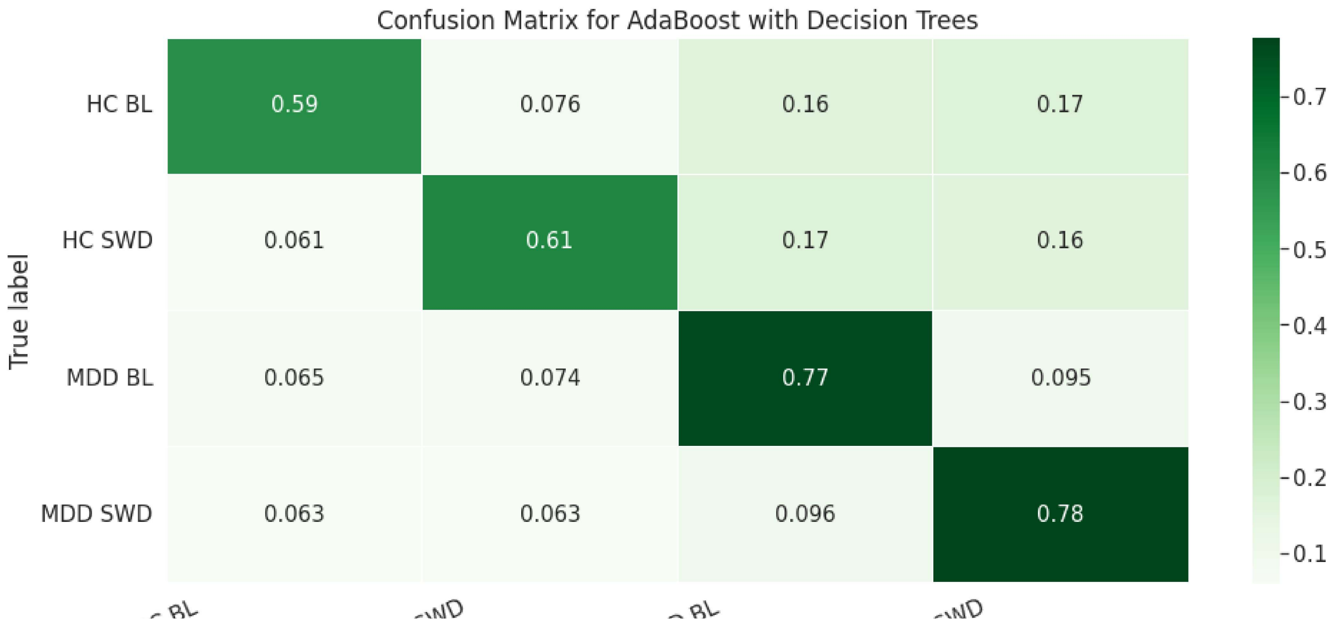
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=42, stratify=y)

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
```

```
0.33114035087719296
```

```
X = data.drop(['Label', 'RA', 'GRA'], 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
```

```
0.706140350877193
```

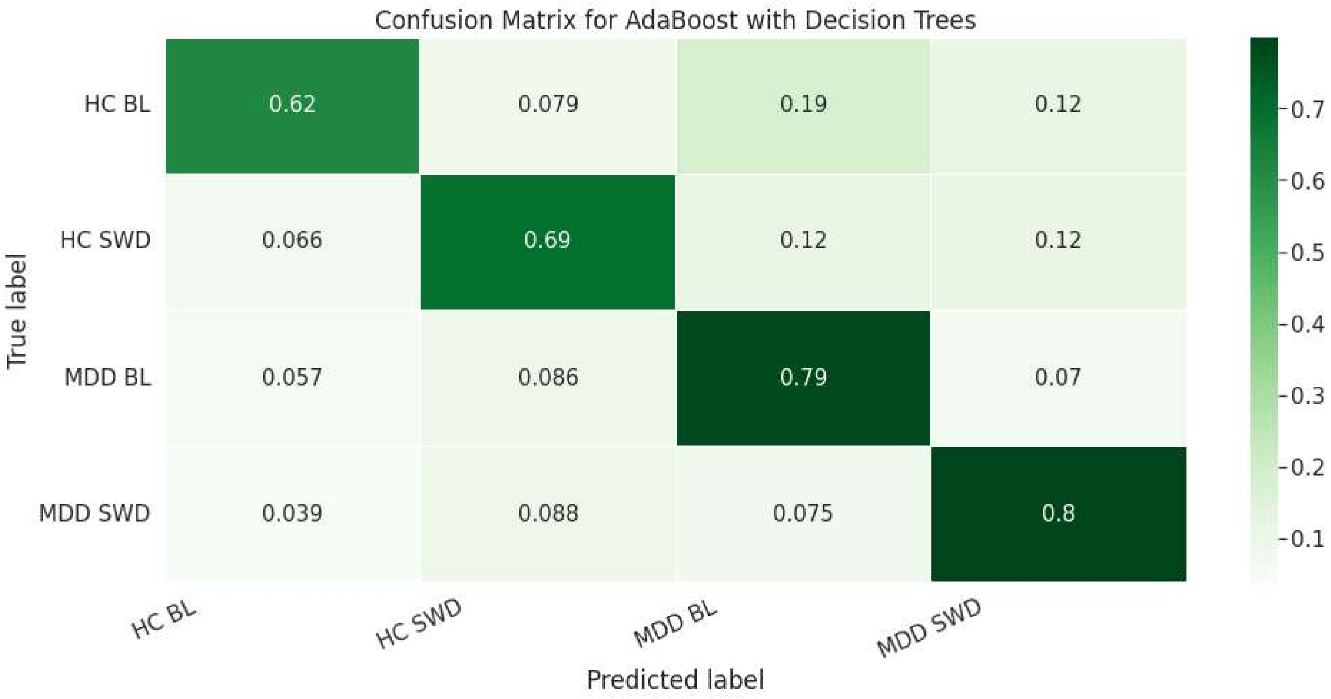
```
matrix = confusion_matrix(y_test, y_pred)
matrix = matrix.astype('float') / matrix.sum(axis=1)[:, np.newaxis]

# 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', 'GRA', '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
```

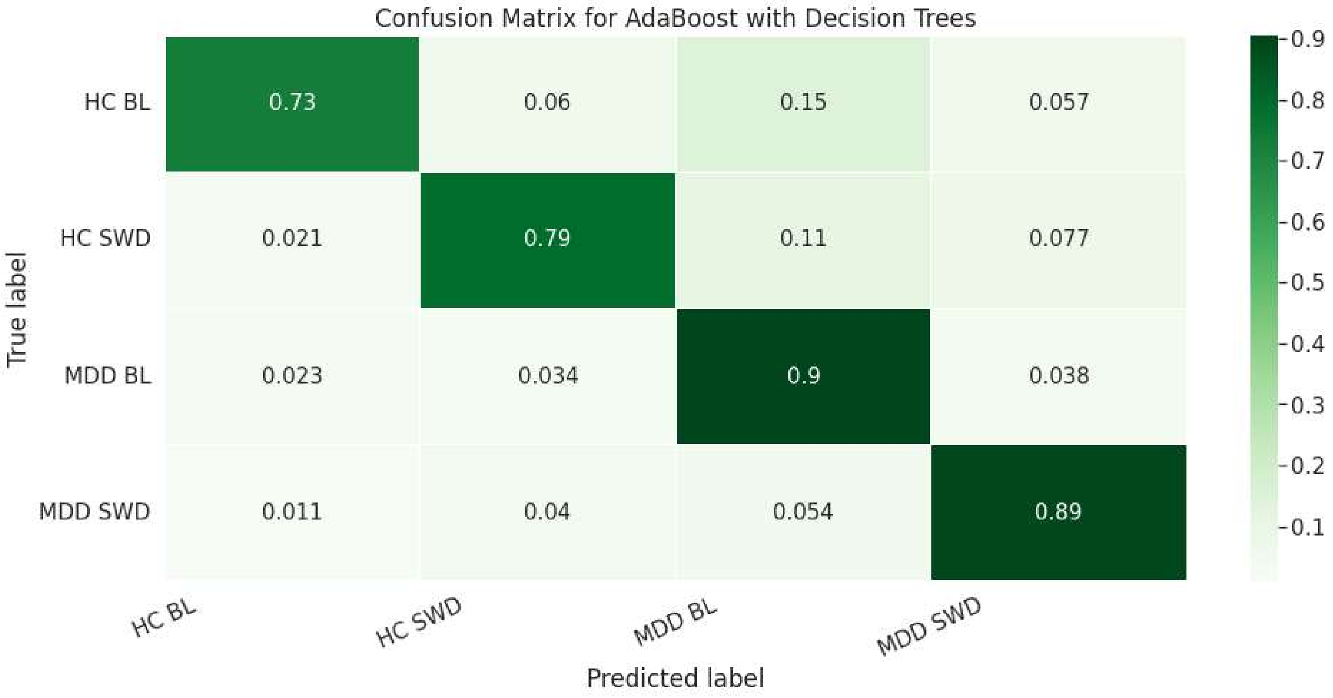
```
matrix = confusion_matrix(y_test, y_pred)
matrix = matrix.astype('float') / matrix.sum(axis=1)[:, np.newaxis]

# 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()
```



```
X = data.drop(['Label', 'GRA'], 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
```

```
0.8481359649122807
```

```
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()
```

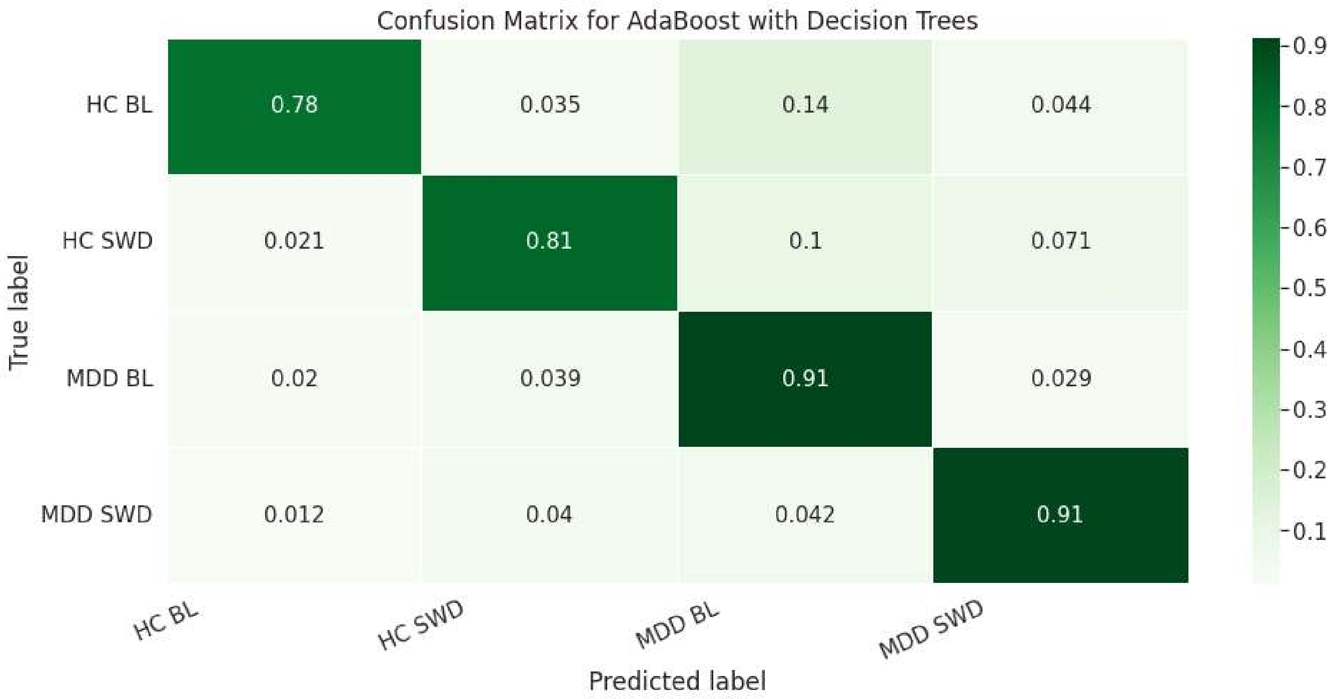


```
X = data.drop(['Label'], axis=1)
y = data['Label']

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
```

```
0.8656798245614035
```

```
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()
```



	HC BL	HC SWD	MDD BL	MDD SWD
HC BL	0.78	0.035	0.14	0.044
HC SWD	0.021	0.81	0.1	0.071
MDD BL	0.02	0.039	0.91	0.029
MDD SWD	0.012	0.04	0.042	0.91

Colab paid products - Cancel contracts here

✓ 1s completed at 3:32 PM ● ✕