```
1 import pandas as pd
2 from sklearn.ensemble import AdaBoostClassifier
3 from sklearn.tree import DecisionTreeClassifier
4 from sklearn.model_selection import train_test_split, cross_val_score
5 from sklearn.metrics import accuracy_score, confusion_matrix
6 import seaborn as sns
7 import matplotlib.pyplot as plt
8 import re
9 import numpy as np
```

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

Choose Files double_weights.csv

- **double_weights.csv**(text/csv) - 1129942 bytes, last modified: 11/23/2022 - 100% done

```
Saving double_weights.csv to double_weights.csv
```

```
1 data = pd.read_csv('double_weights.csv')
2 data = data.drop(['Group', 'Session', 'Subject', 'Type', 'Sample', "InvVarWeights"], axis=1
3 #data['Type'] = data['Type'].str.extract('(\d+)').astype(int)
4
5 #data['Type'] = pd.to_numeric(data['Type'])
6 #data['ISI'] = pd.to_numeric(data['ISI'])
7 data
```

	EMGPeakToPeak	ISI	Label	WRA2	RRA	WRA
0	145.9410	-1	HC BL	2.296693	1.718122	1.866674
1	139.1000	-1	HC BL	2.189035	1.637585	1.779174
2	119.7172	-1	HC BL	1.884005	1.409397	1.531256
3	84.3721	-1	HC BL	1.327775	0.993289	1.079171
4	69.5500	-1	HC BL	1.094517	0.818792	0.889587
...	...	...	...	...	...	...
9377	166.4639	20	HC SWD	1.668391	1.239038	1.314828
9378	149.3614	20	HC SWD	1.496980	1.111740	1.179743
9379	103.7549	20	HC SWD	1.039887	0.772277	0.819516
9380	190.4073	20	HC SWD	1.908364	1.417256	1.503947
9381	294.1622	20	HC SWD	2.948252	2.189533	2.323464

9382 rows × 6 columns

```
1 X = data.drop(['Label', 'WRA', "RRA", "WRA2"], axis=1)
2 y = data['Label']
3
4 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=42,
5
6 bag_clf = AdaBoostClassifier(DecisionTreeClassifier(max_depth=8), n_estimators=20, algorith
```

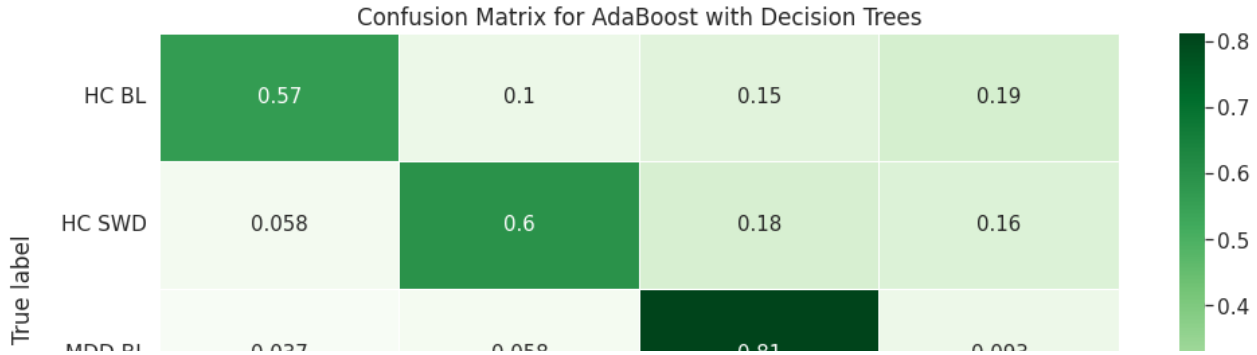
```
bag_clf.fit(X_train, y_train)
y_pred = bag_clf.predict(X_test)
acc = accuracy_score(y_test, y_pred)
acc
```

```
0.33404368673415025
```

```
X = data.drop(['Label', 'WRA', "WRA2"], 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,

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

```
0.7059136920618008
```

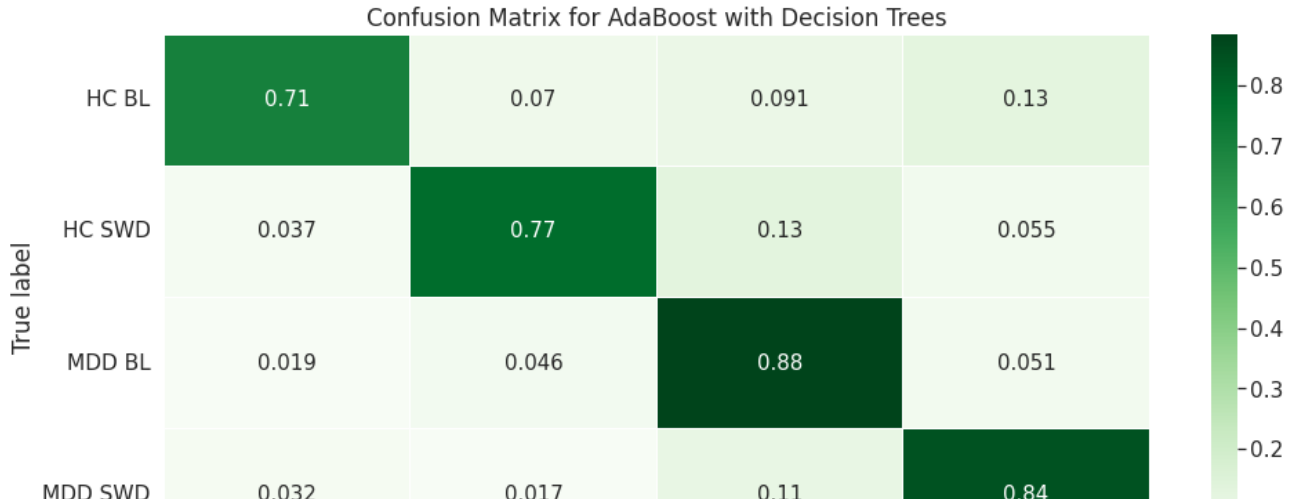
```
matrix = confusion_matrix(y_test, y_pred)
matrix = matrix.astype('float') / matrix.sum(axis=1)[:, np.newaxis]

# 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', "WRA2"], 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,

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

```
0.8172615876398508
```

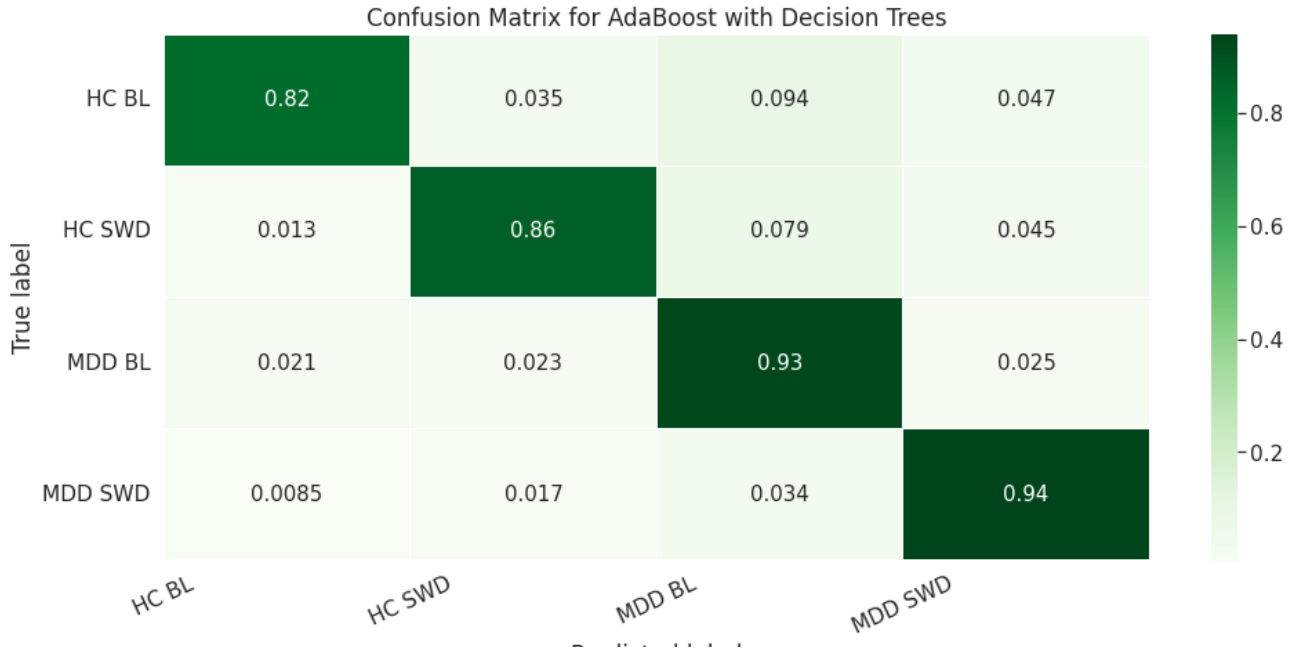
```
matrix = confusion_matrix(y_test, y_pred)
matrix = matrix.astype('float') / matrix.sum(axis=1)[:, np.newaxis]

# 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,

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

```
0.9009057005860416
```

```
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()
```



1

Colab paid products - Cancel contracts here

✓ 0s completed at 10:27 PM