modelling_results_1_0-interp_data-rem_feat-no_border_rem

May 14, 2023

[7]:
```
import pandas as pd
import seaborn as sns
```

[8]:
```
data = pd.read_pickle("df_results_1_0")
```

[9]:
```
data
```

[9]:
```
                Classifier  Precision  Recall  F1-score      Timestamp
1  Support Vector Mashines       0.30    0.38      0.33  10052023_2224
3            Decision Tree       0.27    0.50      0.35  10052023_2224
5            Random Forest       0.51    0.45      0.48  10052023_2224
7           Neural Network       0.20    0.25      0.22  10052023_2224
1  Support Vector Mashines       0.29    0.32      0.31  10052023_2235
3            Decision Tree       0.36    0.25      0.29  10052023_2235
5            Random Forest       0.65    0.38      0.48  10052023_2235
7           Neural Network       0.29    0.45      0.35  10052023_2235
1  Support Vector Mashines       0.48    0.52      0.50  10052023_2247
3            Decision Tree       0.58    0.28      0.37  10052023_2247
5            Random Forest       0.74    0.50      0.60  10052023_2247
7           Neural Network       0.26    0.48      0.34  10052023_2247
```

[18]:
```
data.groupby(by="Classifier").mean().round(2)
```

[18]:
```
                         Precision  Recall  F1-score
Classifier
Decision Tree                 0.40    0.34      0.34
Neural Network                0.25    0.39      0.30
Random Forest                 0.63    0.44      0.52
Support Vector Mashines       0.36    0.41      0.38
```

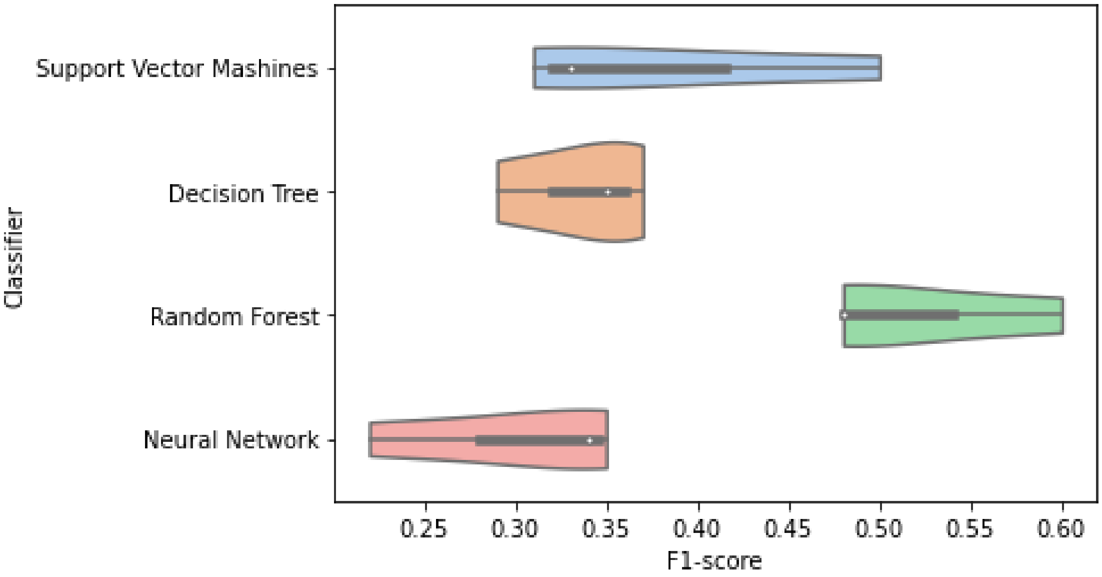
[16]:
```
data.describe()
```

[16]:
```
       Precision     Recall   F1-score
count  12.000000  12.000000  12.000000
mean    0.410833   0.396667   0.385000
std     0.175782   0.101025   0.107492
min     0.200000   0.250000   0.220000
25%     0.285000   0.310000   0.325000
```

```
50%      0.330000    0.415000    0.350000
75%      0.527500    0.485000    0.480000
max      0.740000    0.520000    0.600000
```

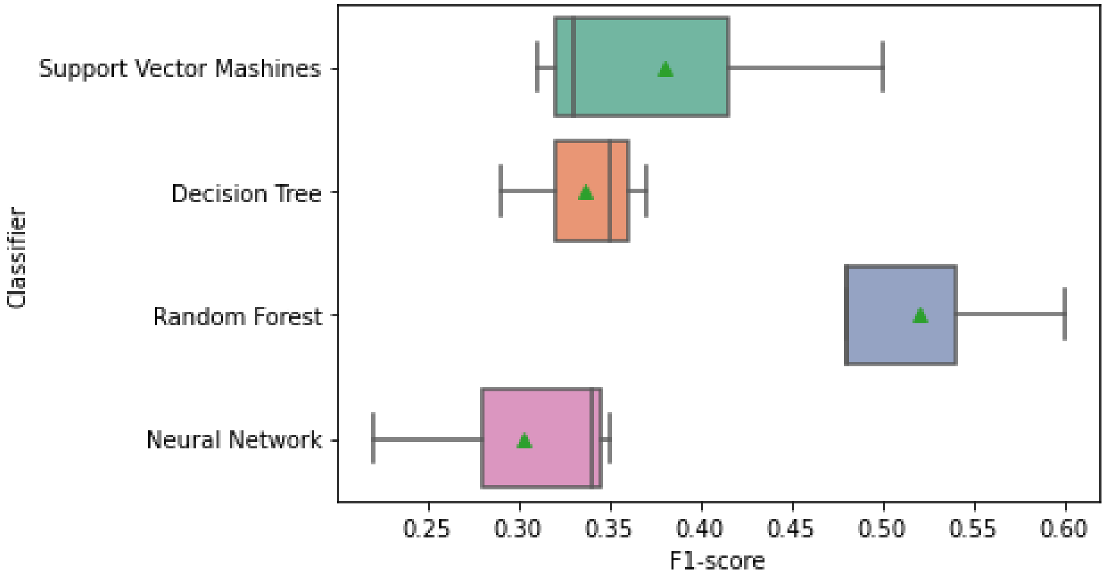
[10]:

```
ax = sns.violinplot(data=data, y="Classifier", x="F1-score", orient="h",␣
 ↪palette="pastel", showmeans=True, cut=0)
```



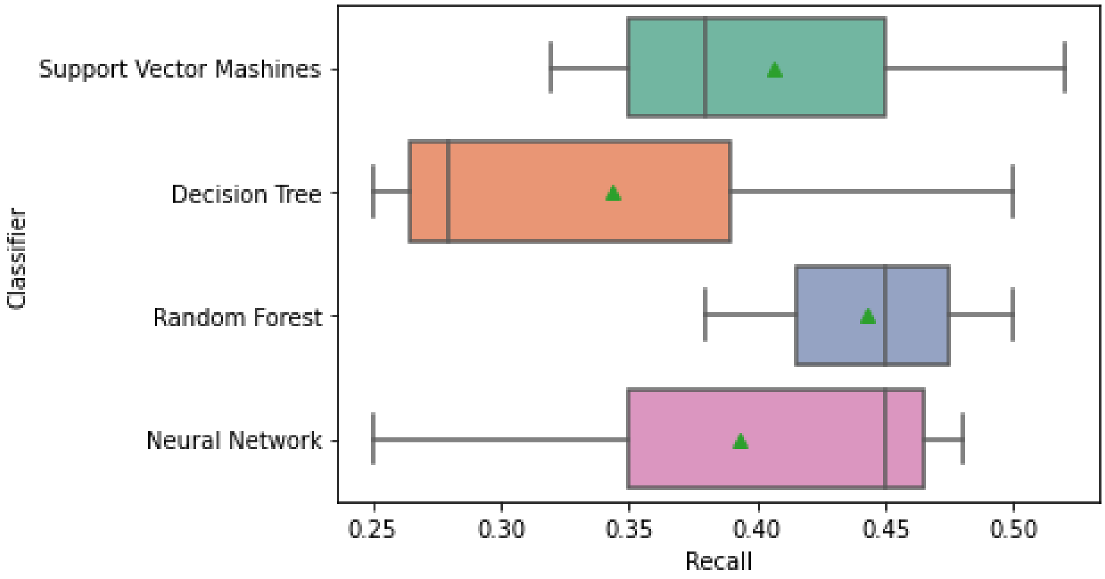
[11]:

```
ax = sns.boxplot(data=data, y="Classifier", x="F1-score", orient="h",␣
 ↪palette="Set2", showmeans=True)
# sns.boxplot(data=data, y="Classifier", x="Recall", orient="h", color="white",␣
 ↪showmeans=True, ax=ax)
```



```
[12]: sns.boxplot(data=data, y="Classifier", x="Recall", orient="h", palette="Set2",␣
      ↪showmeans=True)
```

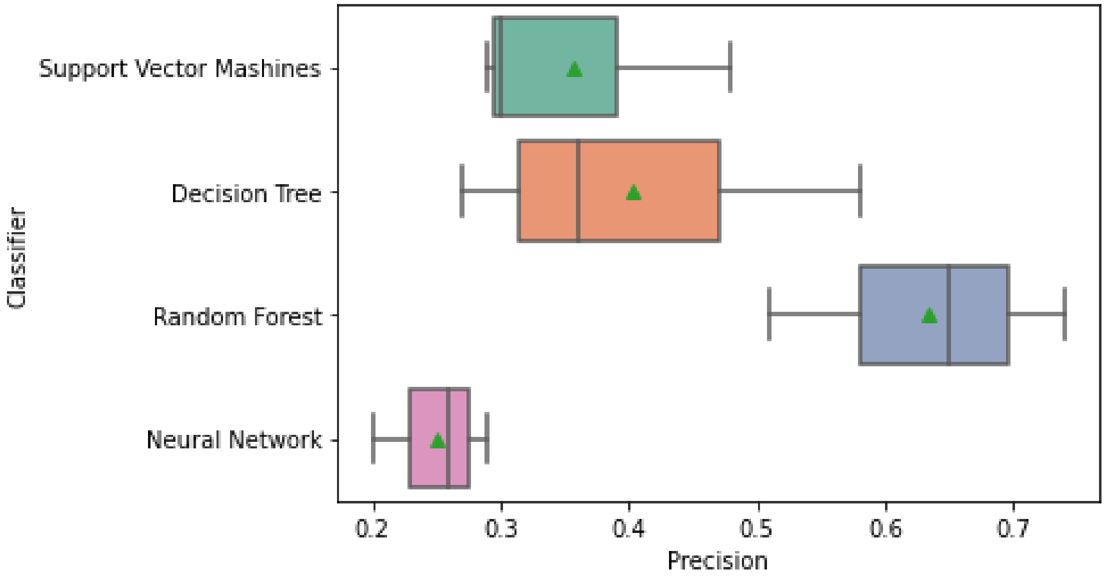
```
[12]: <AxesSubplot:xlabel='Recall', ylabel='Classifier'>
```



```
[13]: sns.boxplot(data=data, y="Classifier", x="Precision", orient="h",␣
      ↪palette="Set2", showmeans=True)
```

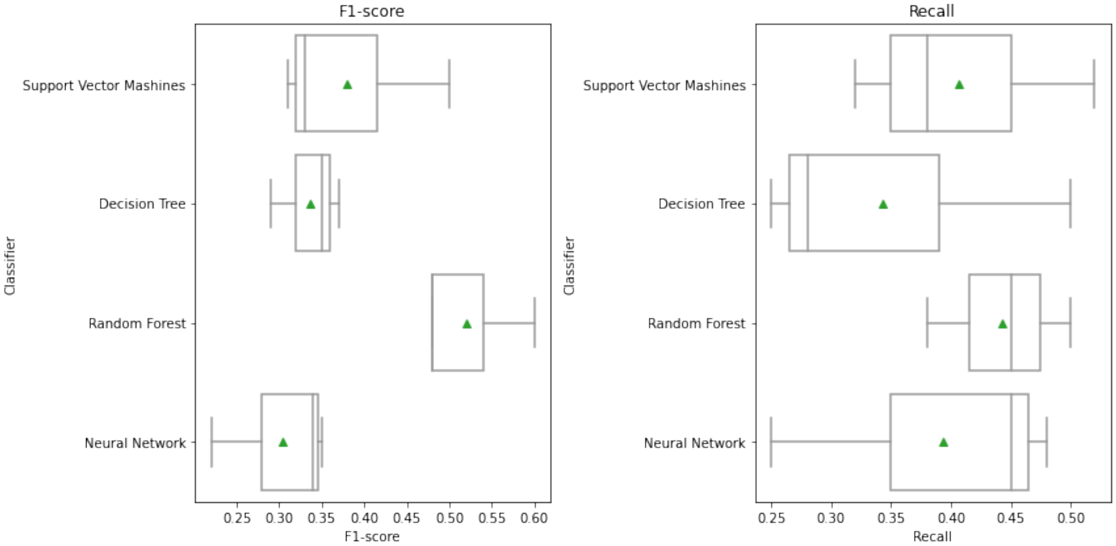
[13]:

```
<AxesSubplot:xlabel='Precision', ylabel='Classifier'>
```



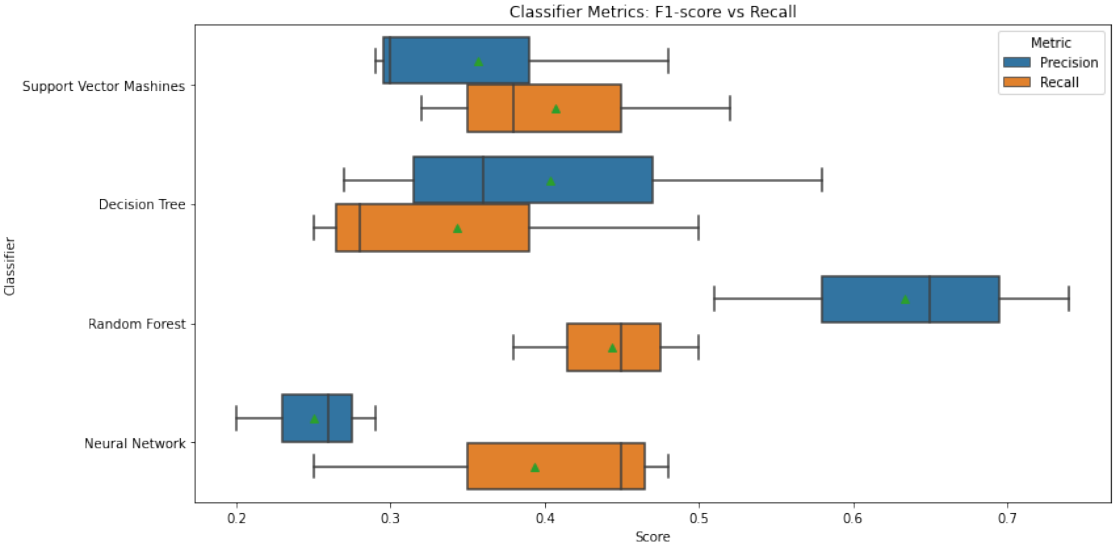
[14]:

```
from matplotlib import pyplot as plt

fig, axs = plt.subplots(ncols=2, figsize=(12, 6))

ax1 = sns.boxplot(data=data, y="Classifier", x="F1-score", orient="h",␣
 ↪color="white", showmeans=True, ax=axs[0])
ax1.set_title("F1-score")

ax2 = sns.boxplot(data=data, y="Classifier", x="Recall", orient="h",␣
 ↪color="white", showmeans=True, ax=axs[1])
ax2.set_title("Recall")

plt.tight_layout()
plt.show()
```



[15]:

```
# Reshape the data
data_melted = pd.melt(data, id_vars='Classifier', value_vars=['Precision', 'Recall'], var_name='Metric', value_name='Score')

# Create the boxplot
plt.figure(figsize=(12, 6))
ax = sns.boxplot(data=data_melted, y="Classifier", x="Score", orient="h", hue="Metric", showmeans=True)
ax.set_title("Classifier Metrics: F1-score vs Recall")
plt.tight_layout()
plt.show()
```



```
[ ]:
```