

## modelling\_results\_1\_0-interp\_data

May 14, 2023

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

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

```
[26]: data
```

```
[26]:
```

	Classifier	Precision	Recall	F1-score	Timestamp
1	Support Vector Mashines	0.29	0.08	0.12	10052023_2042
3	Decision Tree	0.33	0.23	0.27	10052023_2042
5	Random Forest	0.53	0.35	0.42	10052023_2042
7	Neural Network	0.67	0.15	0.25	10052023_2042
1	Support Vector Mashines	0.30	0.62	0.40	10052023_2054
3	Decision Tree	0.36	0.46	0.41	10052023_2054
5	Random Forest	0.63	0.46	0.53	10052023_2054
7	Neural Network	0.32	0.65	0.43	10052023_2054
1	Support Vector Mashines	0.33	0.27	0.30	10052023_2120
3	Decision Tree	0.24	0.31	0.27	10052023_2120
5	Random Forest	0.48	0.42	0.45	10052023_2120
7	Neural Network	0.35	0.42	0.39	10052023_2120

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

```
[35]:
```

	Precision	Recall	F1-score
Classifier			
Decision Tree	0.31	0.33	0.32
Neural Network	0.45	0.41	0.36
Random Forest	0.55	0.41	0.47
Support Vector Mashines	0.31	0.32	0.27

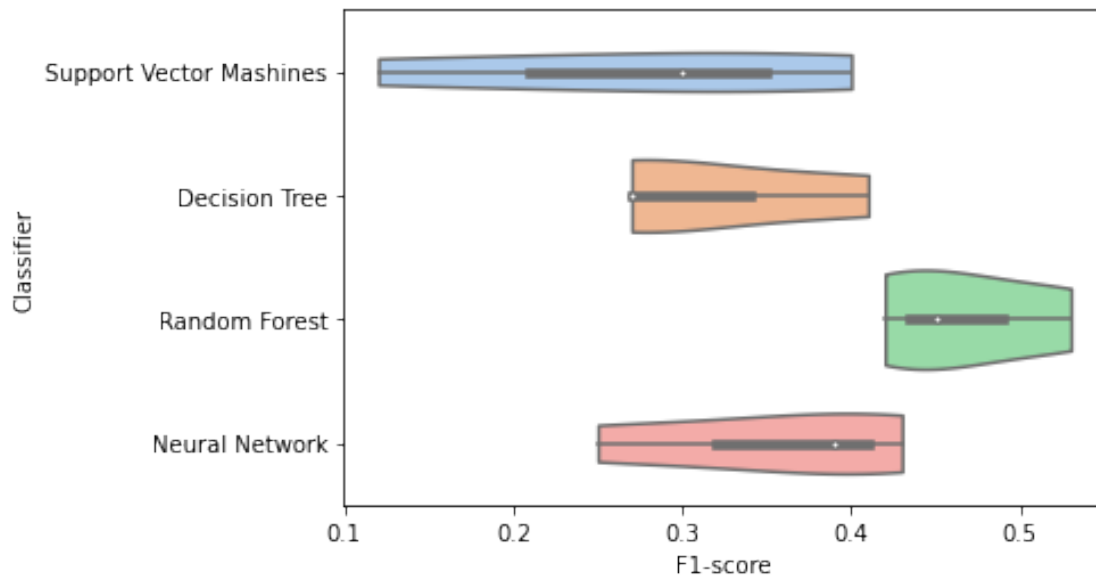
```
[33]: data.describe()
```

```
[33]:
```

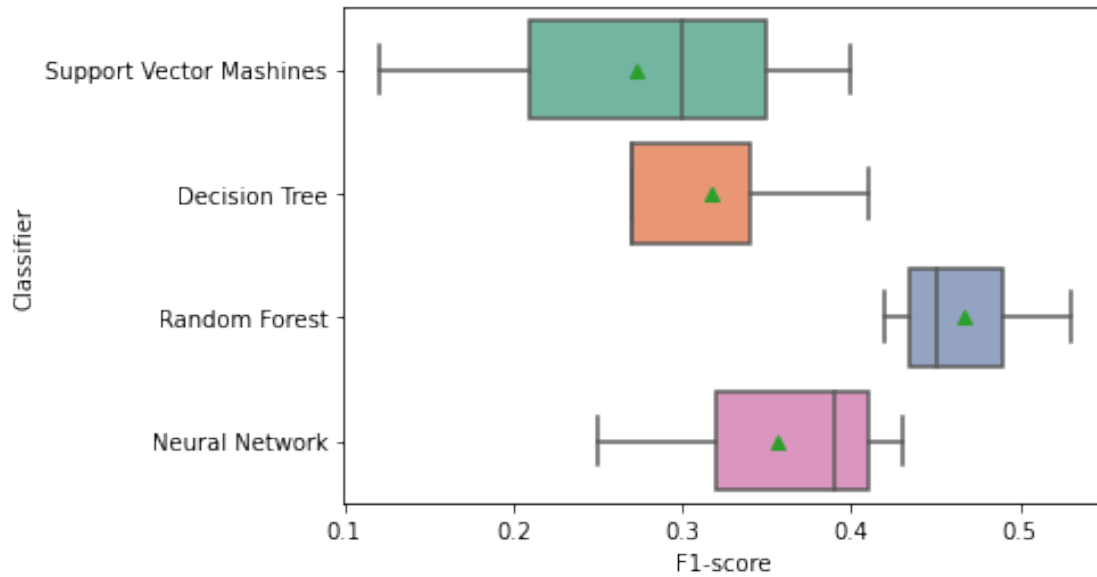
	Precision	Recall	F1-score
count	12.000000	12.000000	12.000000
mean	0.402500	0.368333	0.353333
std	0.140462	0.172723	0.112600
min	0.240000	0.080000	0.120000
25%	0.315000	0.260000	0.270000

50%	0.340000	0.385000	0.395000
75%	0.492500	0.460000	0.422500
max	0.670000	0.650000	0.530000

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

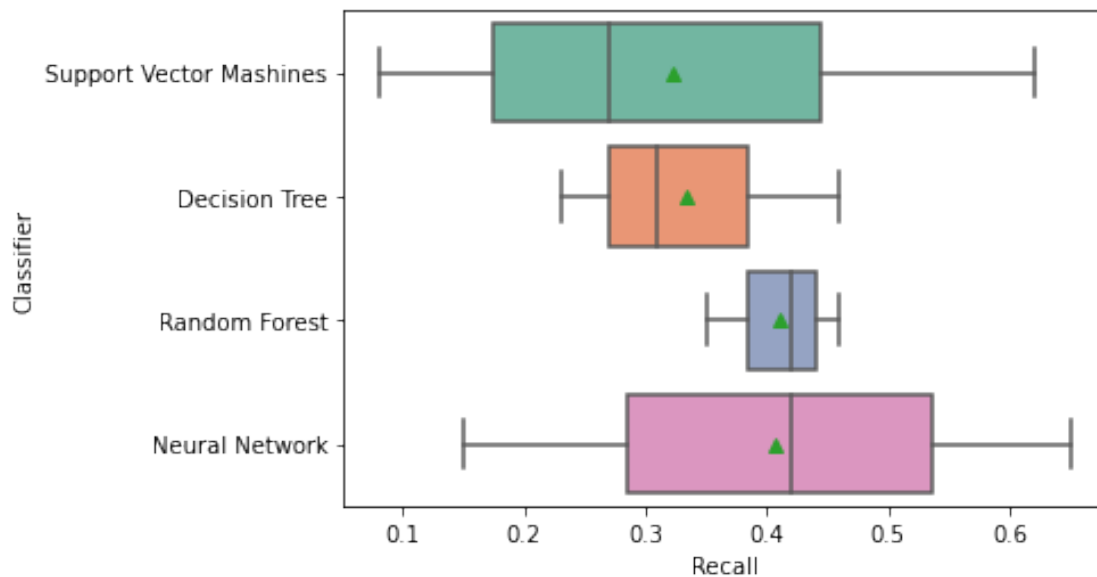


```
[28]: ax = sns.boxplot(data=data, y="Classifier", x="F1-score", orient="h",
    ↪ palette="Set2", showmeans=True)
# sns.boxplot(data=data, y="Classifier", x="Recall", orient="h", color="white",
    ↪ showmeans=True, ax=ax)
```



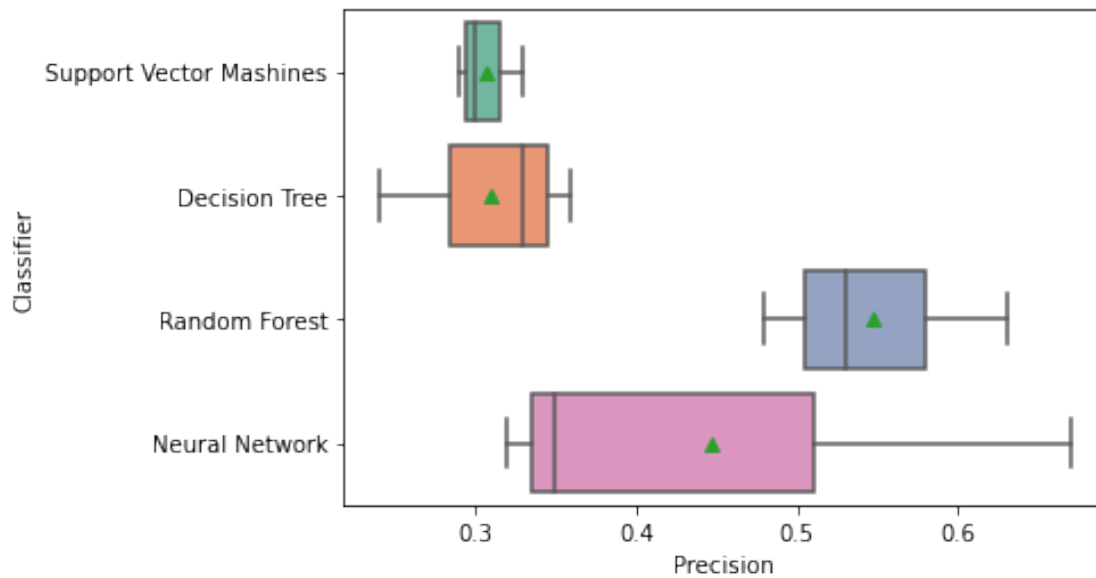
```
[29]: sns.boxplot(data=data, y="Classifier", x="Recall", orient="h", palette="Set2",
→showmeans=True)
```

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



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

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



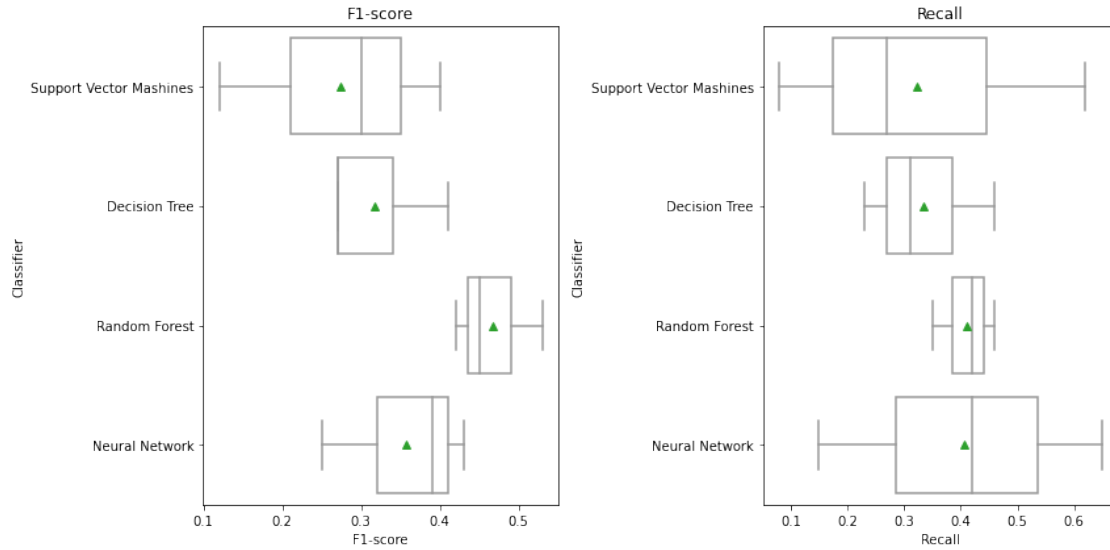
```
[31]: from matplotlib import pyplot as plt

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")

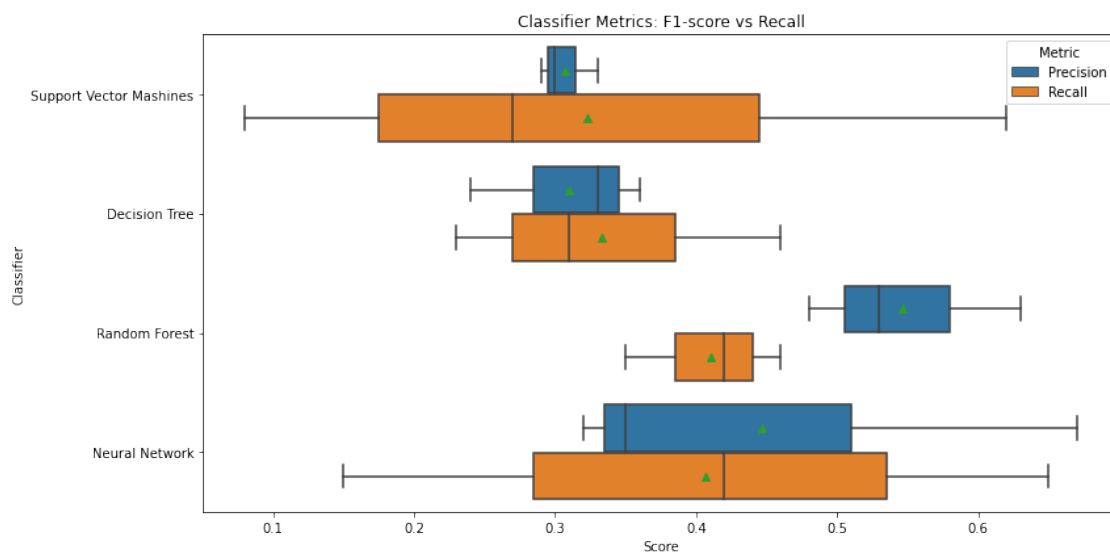
ax2 = sns.boxplot(data=data, y="Classifier", x="Recall", orient="h",
                  color="white", showmeans=True, ax=axs[1])
ax2.set_title("Recall")

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



```
[32]: # 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()
```



[ ]: