

modelling_results_1_0-orig_data-rem_feat

May 14, 2023

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

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

```
[6]: data
```

```
[6]:
```

	Classifier	Precision	Recall	F1-score	Timestamp
1	Support Vector Mashines	0.76	0.56	0.64	10052023_2303
3	Decision Tree	0.47	0.21	0.29	10052023_2303
5	Random Forest	0.91	0.59	0.71	10052023_2303
7	Neural Network	0.42	0.68	0.52	10052023_2303
1	Support Vector Mashines	0.65	0.38	0.48	10052023_2315
3	Decision Tree	0.42	0.62	0.50	10052023_2315
5	Random Forest	0.89	0.50	0.64	10052023_2315
7	Neural Network	0.43	0.44	0.43	10052023_2315
1	Support Vector Mashines	0.64	0.41	0.50	10052023_2326
3	Decision Tree	0.37	0.44	0.40	10052023_2326
5	Random Forest	0.87	0.59	0.70	10052023_2326
7	Neural Network	0.00	0.00	0.00	10052023_2326

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

```
[19]:
```

	Precision	Recall	F1-score
Classifier			
Decision Tree	0.42	0.42	0.40
Neural Network	0.28	0.37	0.32
Random Forest	0.89	0.56	0.68
Support Vector Mashines	0.68	0.45	0.54

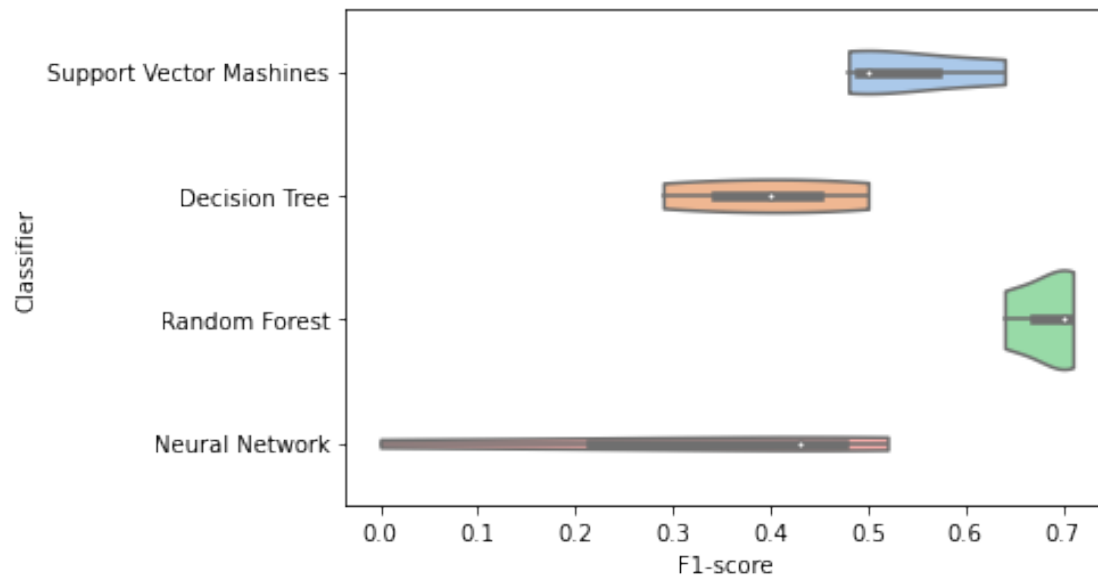
```
[13]: data.describe()
```

```
[13]:
```

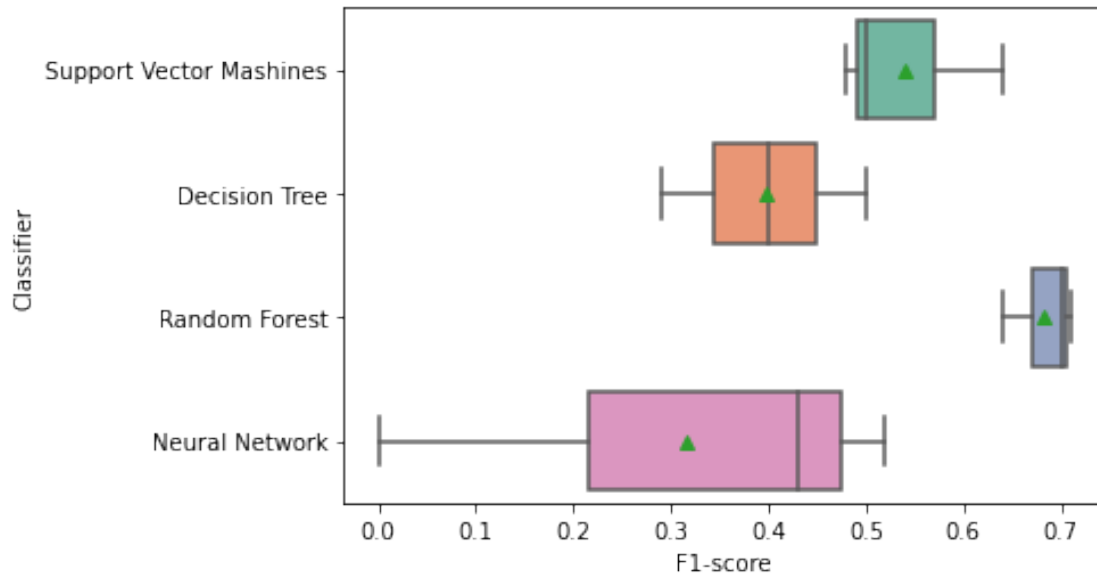
	Precision	Recall	F1-score
count	12.000000	12.000000	12.000000
mean	0.569167	0.451667	0.484167
std	0.268818	0.191256	0.197735
min	0.000000	0.000000	0.000000
25%	0.420000	0.402500	0.422500

50%	0.555000	0.470000	0.500000
75%	0.787500	0.590000	0.640000
max	0.910000	0.680000	0.710000

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

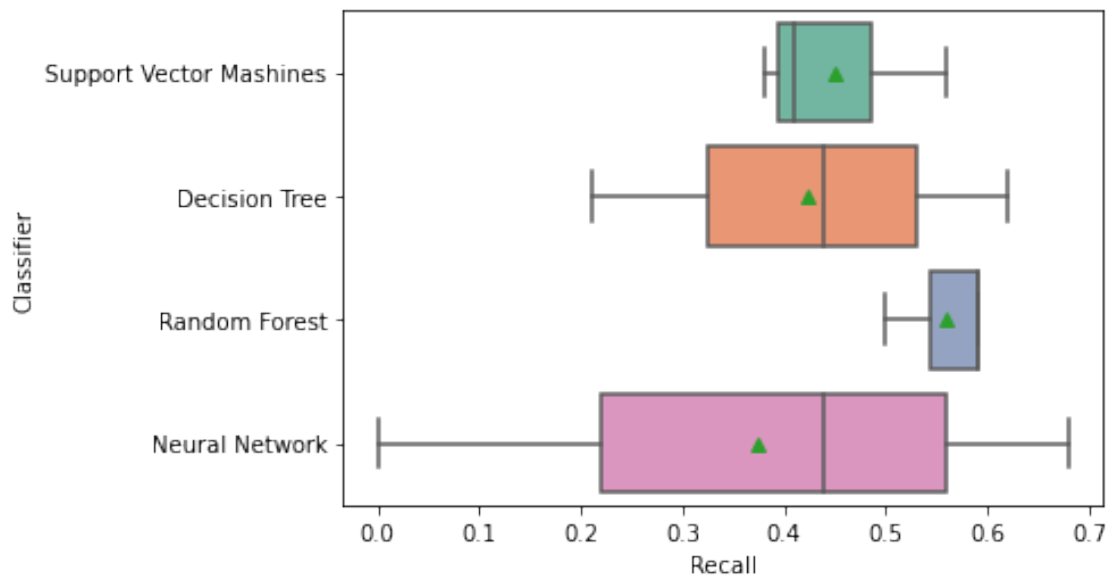


```
[8]: ax = sns.boxplot(data=data, y="Classifier", x="F1-score", orient="h",
    ↪ palette="Set2", showmeans=True)
# sns.boxplot(data=data, y="Classifier", x="Recall", orient="h", color="white",
    ↪ showmeans=True, ax=ax)
```



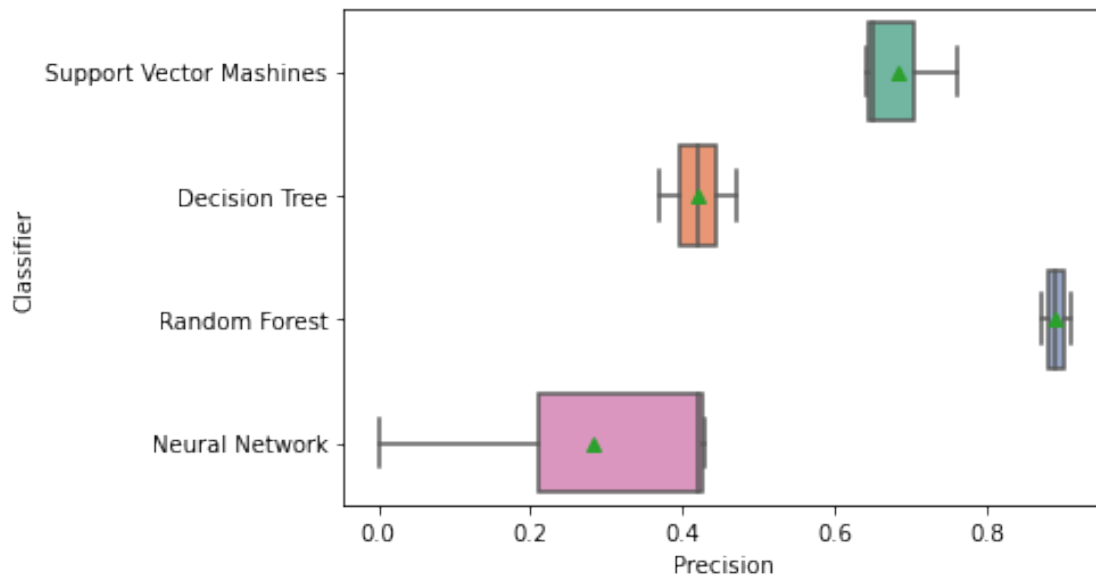
```
[9]: sns.boxplot(data=data, y="Classifier", x="Recall", orient="h", palette="Set2",
    ↪ showmeans=True)
```

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



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

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



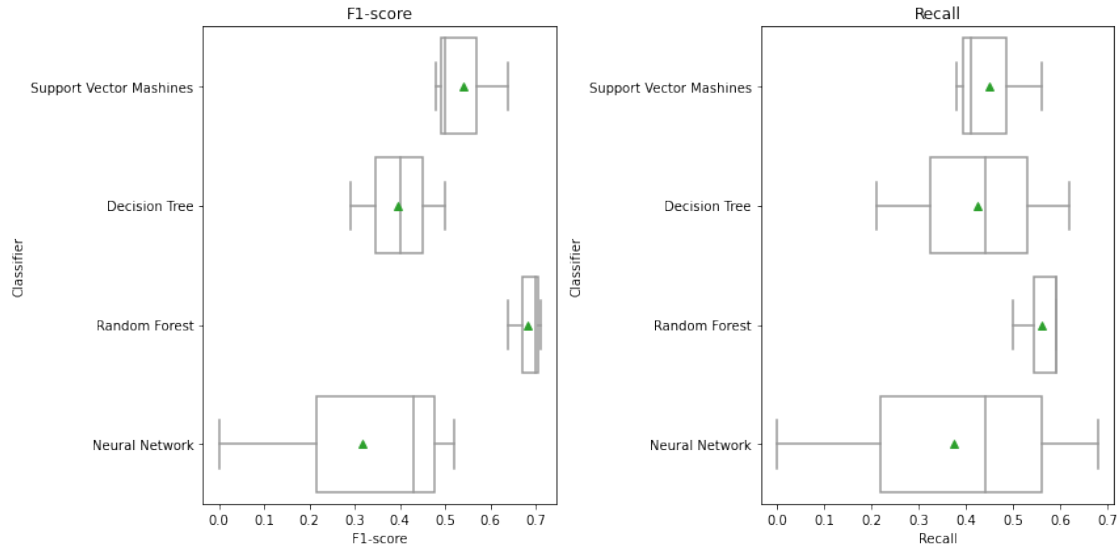
```
[11]: from matplotlib import pyplot as plt

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")

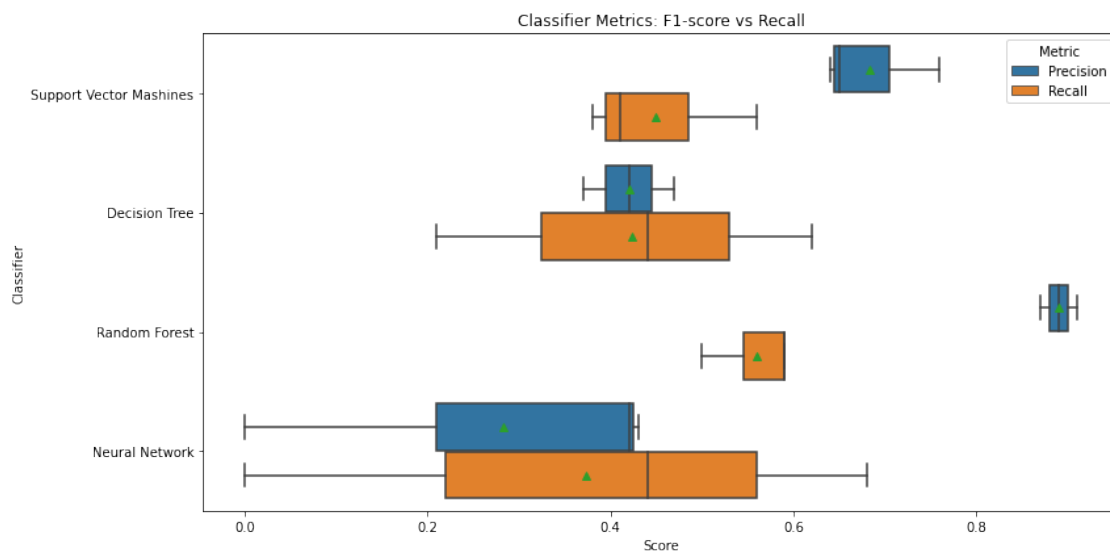
ax2 = sns.boxplot(data=data, y="Classifier", x="Recall", orient="h",
                  color="white", showmeans=True, ax=axs[1])
ax2.set_title("Recall")

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



```
[12]: # 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()
```



[]: