modelling_results_2_0-10x-interp_data-rem_feat

May 14, 2023

[53]:
```
import pandas as pd
import seaborn as sns
```

[76]:
```
data = pd.read_pickle("df_results_1_0")
```

[77]:
```
data.reset_index(inplace=True)
```

[78]:
```
data.drop([16,17, 18, 19], axis=0, inplace=True)
data.drop("index", axis=1, inplace=True)
```

[79]:
```
data
```

[79]:

	Classifier	Precision	Recall	F1-score	Timestamp
0	Support Vector Mashines	0.71	0.50	0.59	11052023_0018
1	Decision Tree	0.50	0.42	0.46	11052023_0018
2	Random Forest	0.83	0.62	0.71	11052023_0018
3	Neural Network	0.33	0.60	0.43	11052023_0018
4	Support Vector Mashines	0.62	0.32	0.43	11052023_0034
5	Decision Tree	0.37	0.55	0.44	11052023_0034
6	Random Forest	0.57	0.62	0.60	11052023_0034
7	Neural Network	0.41	0.60	0.48	11052023_0034
8	Support Vector Mashines	0.51	0.57	0.54	11052023_0047
9	Decision Tree	0.48	0.40	0.44	11052023_0047
10	Random Forest	0.81	0.65	0.72	11052023_0047
11	Neural Network	0.43	0.50	0.46	11052023_0047
12	Support Vector Mashines	0.70	0.35	0.47	11052023_0059
13	Decision Tree	0.58	0.28	0.37	11052023_0059
14	Random Forest	0.84	0.68	0.75	11052023_0059
15	Neural Network	0.30	0.52	0.38	11052023_0059
20	Support Vector Mashines	0.68	0.42	0.52	11052023_0111
21	Decision Tree	0.47	0.62	0.54	11052023_0111
22	Random Forest	0.86	0.60	0.71	11052023_0111
23	Neural Network	0.29	0.52	0.38	11052023_0111
24	Support Vector Mashines	0.65	0.50	0.56	11052023_0124
25	Decision Tree	0.33	0.48	0.39	11052023_0124
26	Random Forest	0.84	0.52	0.65	11052023_0124
27	Neural Network	0.35	0.55	0.43	11052023_0124
28	Support Vector Mashines	0.46	0.48	0.47	11052023_0135

```
29            Decision Tree       0.65    0.50      0.56  11052023_0135
30            Random Forest       0.79    0.75      0.77  11052023_0135
31           Neural Network       0.40    0.70      0.51  11052023_0135
32  Support Vector Mashines       0.70    0.48      0.57  11052023_0146
33            Decision Tree       0.32    0.62      0.43  11052023_0146
34            Random Forest       0.97    0.70      0.81  11052023_0146
35           Neural Network       0.46    0.45      0.46  11052023_0146
36  Support Vector Mashines       0.61    0.48      0.54  11052023_0157
37            Decision Tree       0.37    0.45      0.40  11052023_0157
38            Random Forest       0.89    0.62      0.74  11052023_0157
39           Neural Network       0.50    0.50      0.50  11052023_0157
40  Support Vector Mashines       0.55    0.45      0.49  11052023_0214
41            Decision Tree       0.38    0.28      0.32  11052023_0214
42            Random Forest       0.81    0.62      0.70  11052023_0214
43           Neural Network       0.51    0.48      0.49  11052023_0214
```

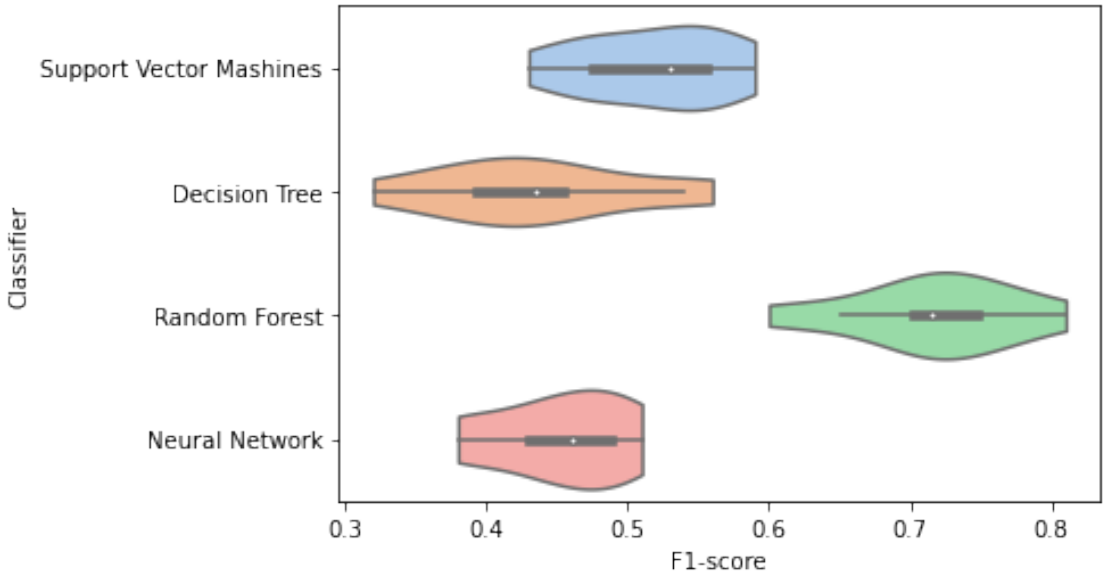
[80]:
```
data.groupby(by="Classifier").mean().round(2)
```

[80]:
```
                        Precision  Recall  F1-score
Classifier
Decision Tree                0.44    0.46      0.43
Neural Network               0.40    0.54      0.45
Random Forest                0.82    0.64      0.72
Support Vector Mashines      0.62    0.45      0.52
```

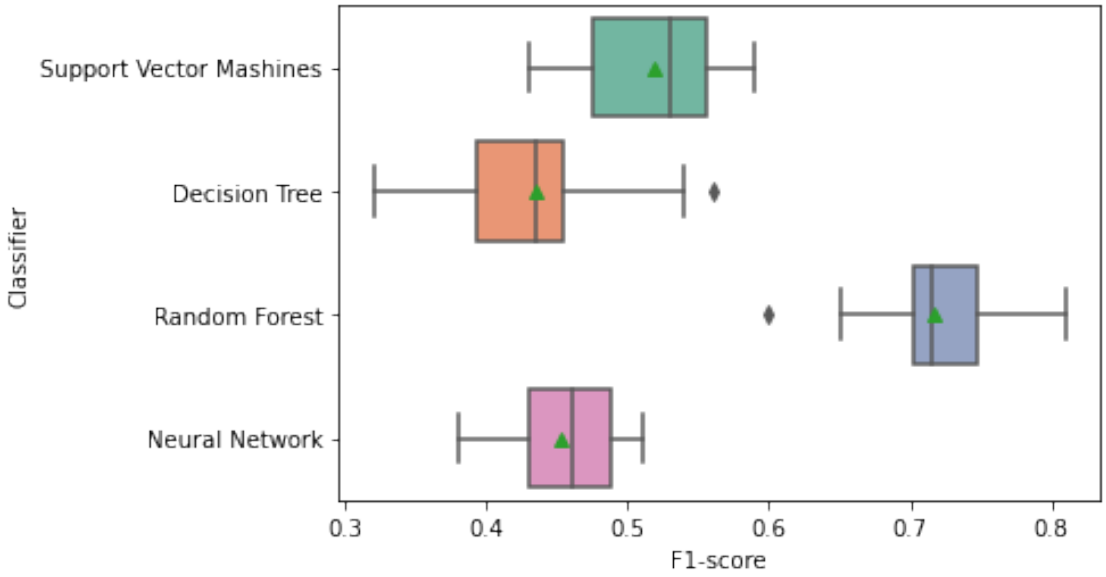
[81]:
```
data.describe()
```

[81]:
```
       Precision    Recall   F1-score
count  40.000000  40.00000  40.000000
mean    0.570750   0.52375   0.530250
std     0.191705   0.11167   0.126217
min     0.290000   0.28000   0.320000
25%     0.407500   0.47250   0.437500
50%     0.530000   0.51000   0.495000
75%     0.702500   0.62000   0.592500
max     0.970000   0.75000   0.810000
```

[82]:
```
ax = sns.violinplot(data=data, y="Classifier", x="F1-score", orient="h",␣
↪palette="pastel", showmeans=True, cut=0)
```

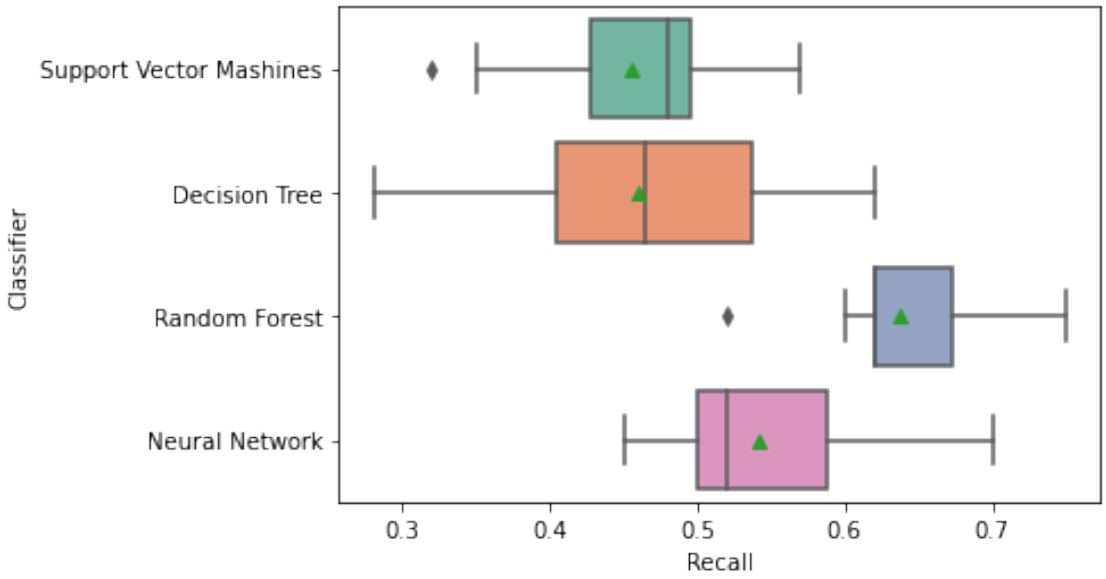


[83]:
```
ax = sns.boxplot(data=data, y="Classifier", x="F1-score", orient="h",␣
↪palette="Set2", showmeans=True)
# sns.boxplot(data=data, y="Classifier", x="Recall", orient="h", color="white",␣
↪showmeans=True, ax=ax)
```



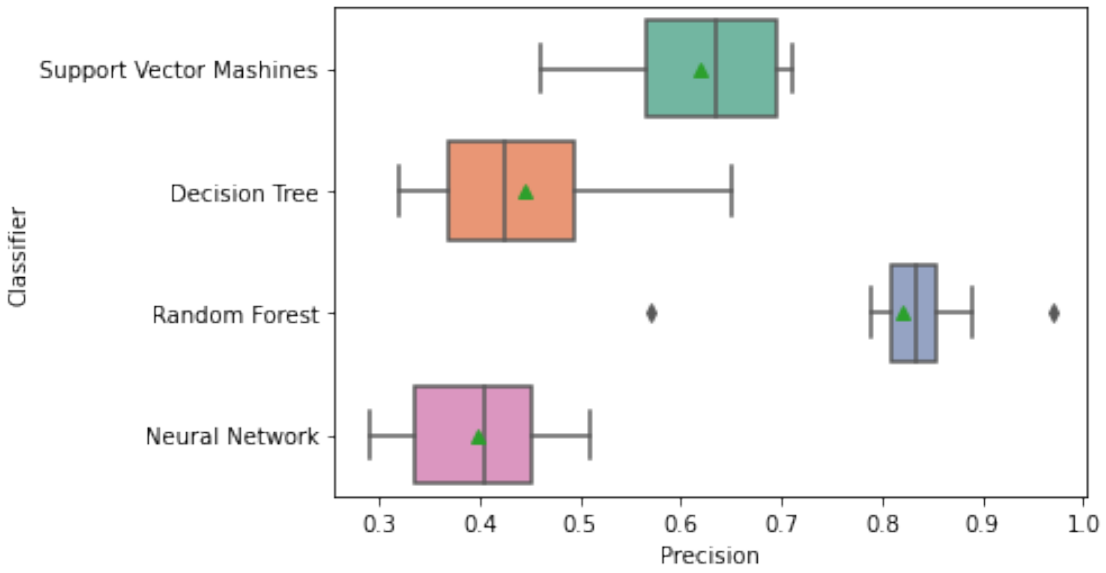
[84]:
```
sns.boxplot(data=data, y="Classifier", x="Recall", orient="h", palette="Set2",␣
↪showmeans=True)
```

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

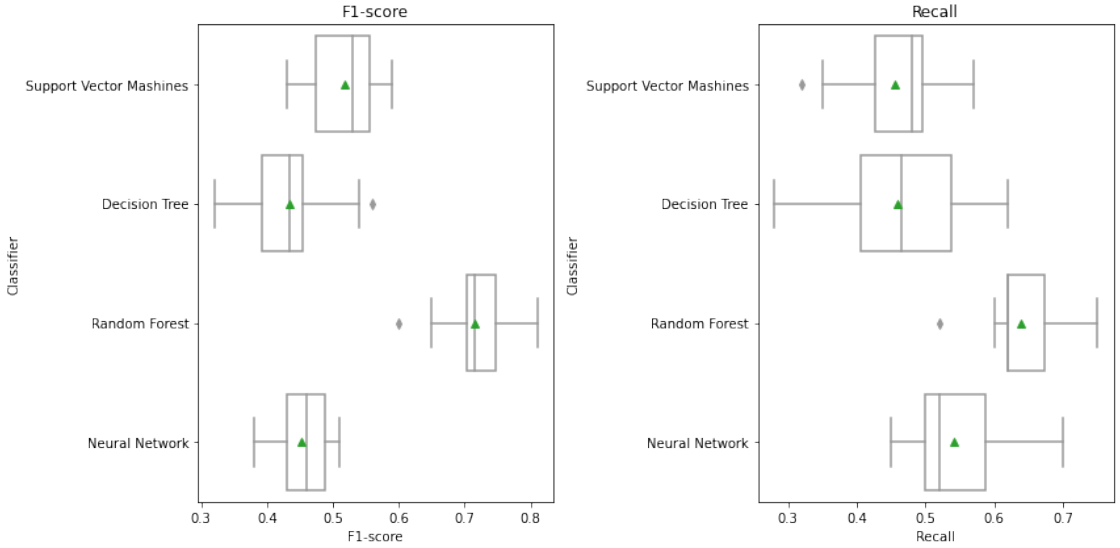


[85]:
```
sns.boxplot(data=data, y="Classifier", x="Precision", orient="h",␣
↪palette="Set2", showmeans=True)
```

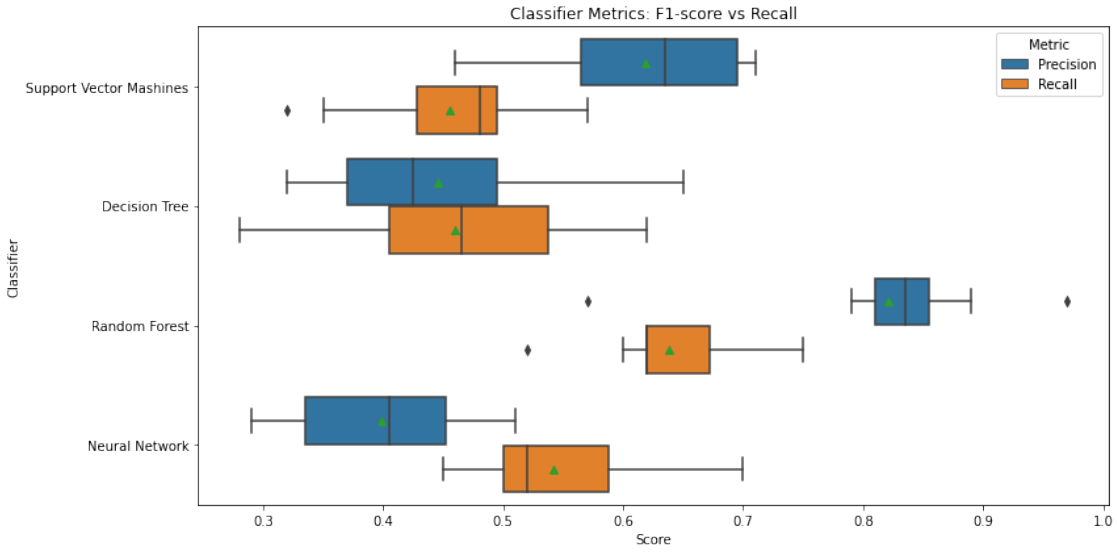
[85]: <AxesSubplot:xlabel='Precision', ylabel='Classifier'>



[86]:
```
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()
```



[87]:
```
# 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()
```


Classifier Metrics: F1-score vs Recall
Metric
Precision
Recall
Support Vector Mashines
Decision Tree
Random Forest
Neural Network
Classifier
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
Score

[ ]: