BayesianReal

March 3, 2025

[1]:
```
from src.utils.TestHelper import TestHelper
from src.utils.Dataset import Dataset
```

[2]:
```
tester = TestHelper()
test_sizes = [0.1, 0.2, 0.3, 0.4]
```

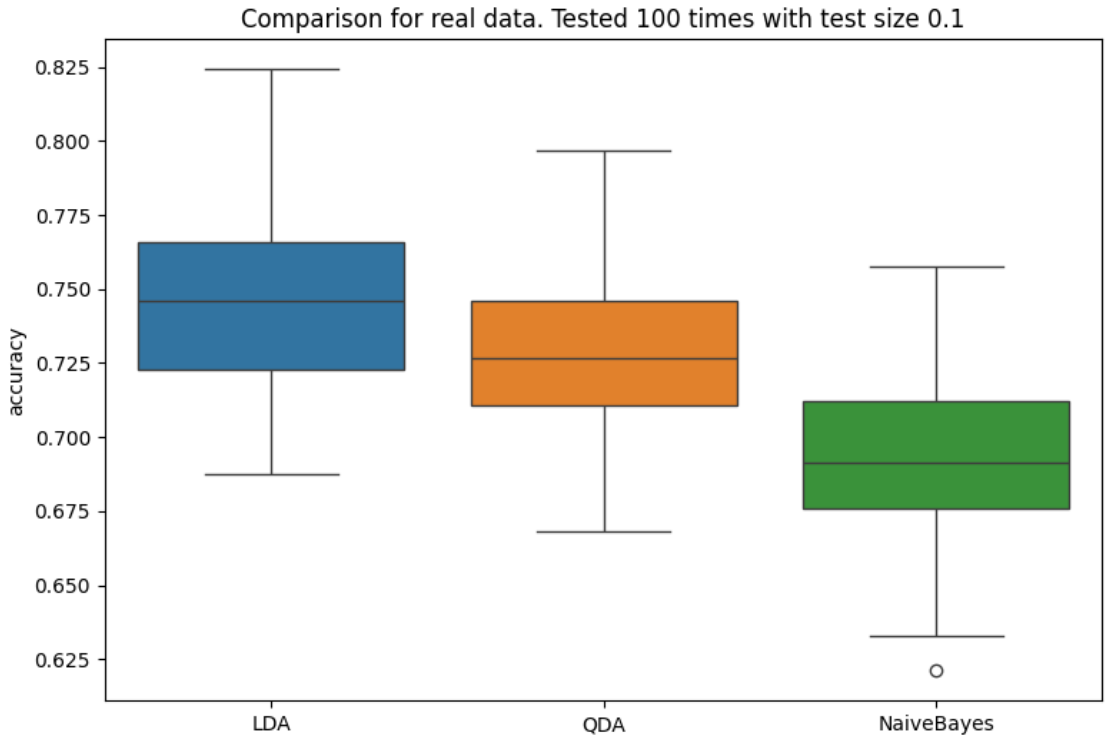
1 wine dataset

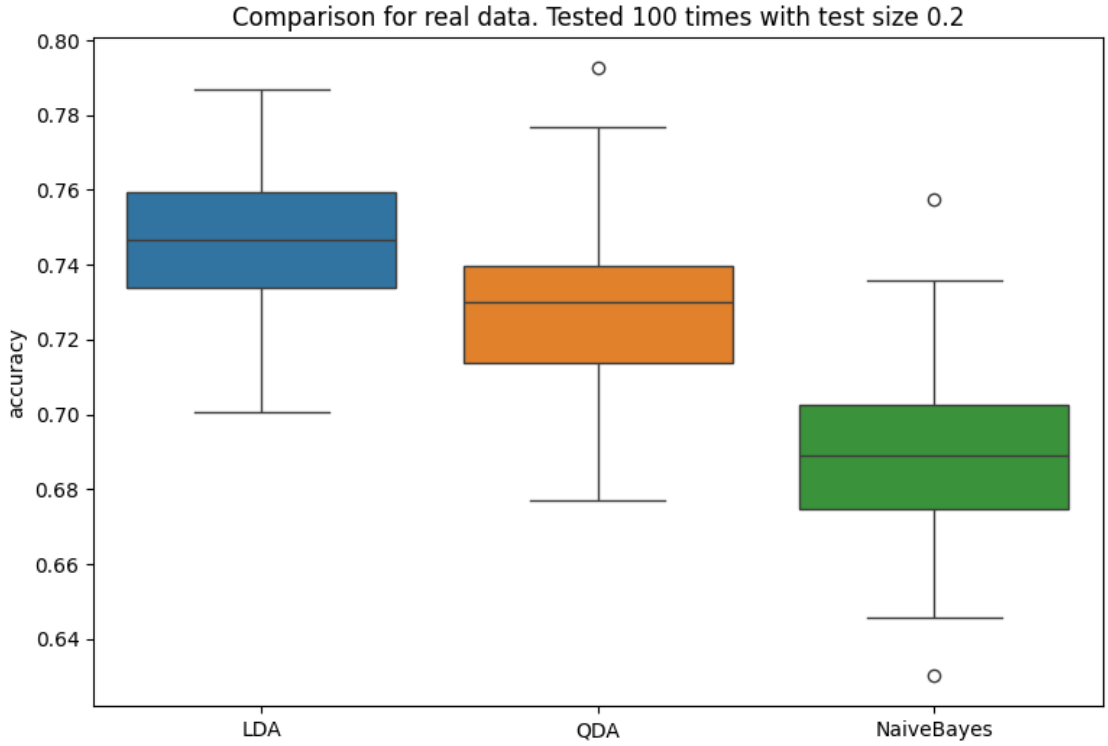
[14]:
```
X, y = Dataset.load_data(44091)
```

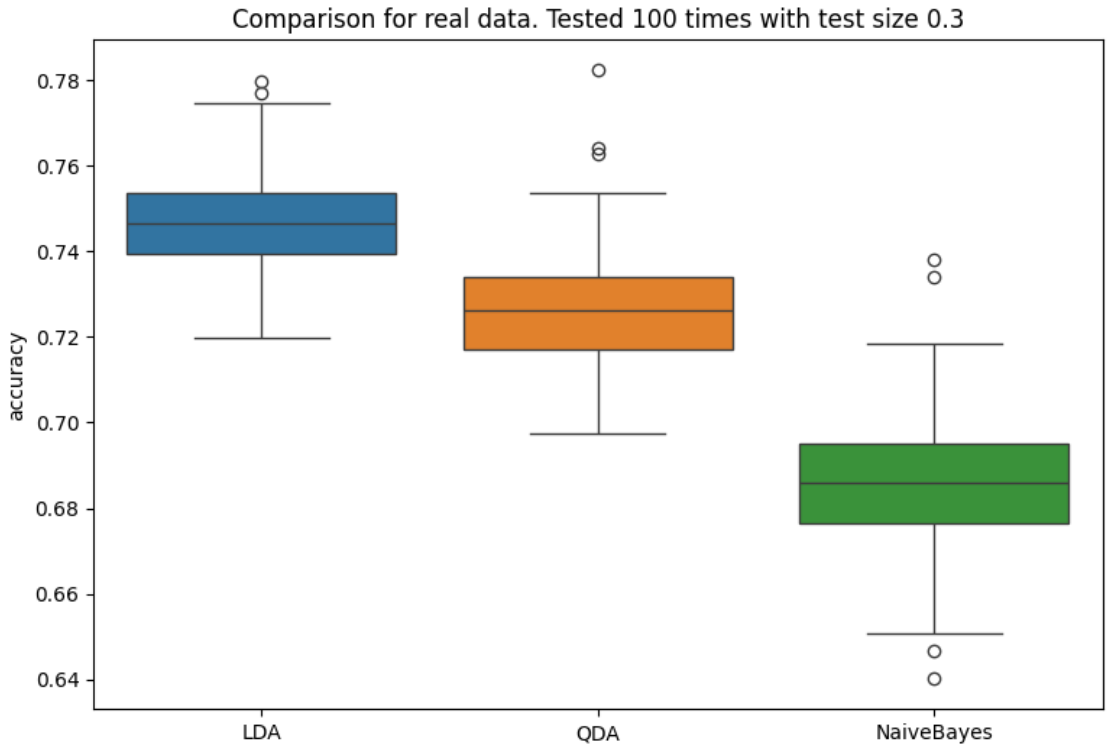
[15]:
```
print(X.shape)
print(y.shape)
```

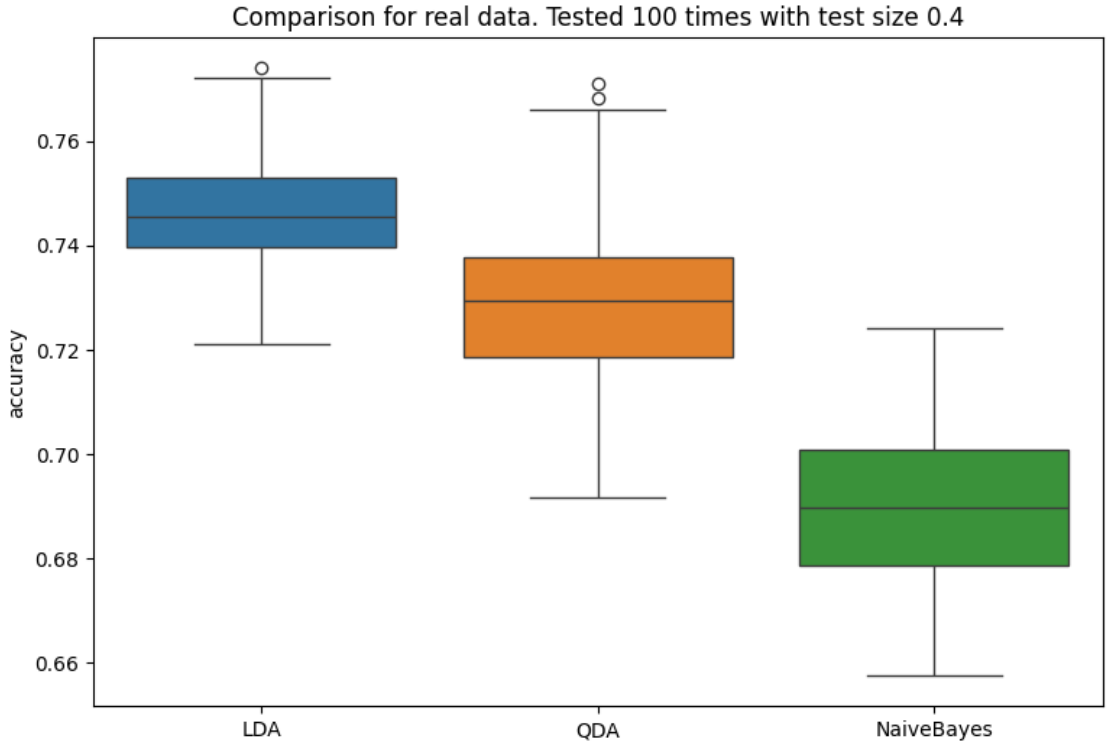
```
(2554, 11)
(2554,)
```

[16]:
```
for test_size in test_sizes:
    tester.test_real_data(X, y, 100, test_size)
```


Comparison for real data. Tested 100 times with test size 0.1
accuracy
0.825
0.800
0.775
0.750
0.725
0.700
0.675
0.650
0.625
LDA
QDA
NaiveBayes


Comparison for real data. Tested 100 times with test size 0.2
accuracy
0.80
0.78
0.76
0.74
0.72
0.70
0.68
0.66
0.64
LDA
QDA
NaiveBayes


Comparison for real data. Tested 100 times with test size 0.3
accuracy
0.78
0.76
0.74
0.72
0.70
0.68
0.66
0.64
LDA
QDA
NaiveBayes


Comparison for real data. Tested 100 times with test size 0.4
accuracy
0.76
0.74
0.72
0.70
0.68
0.66
LDA
QDA
NaiveBayes

1.1 Key observations

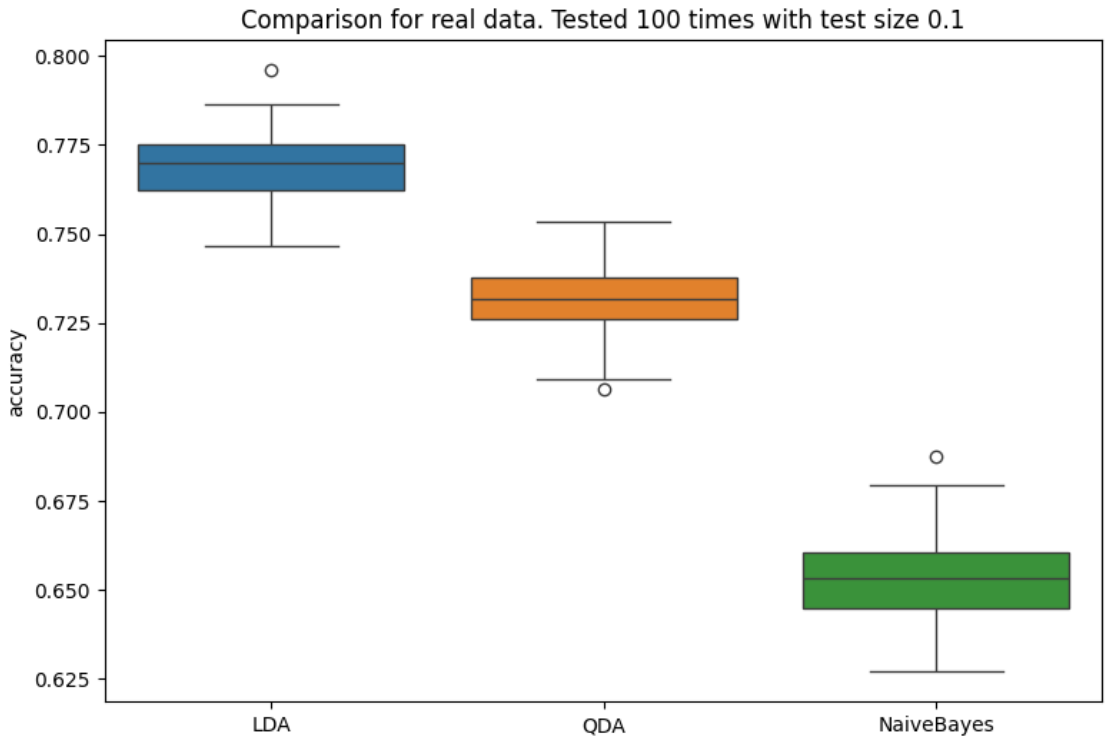
2 MagicTelescope dataset

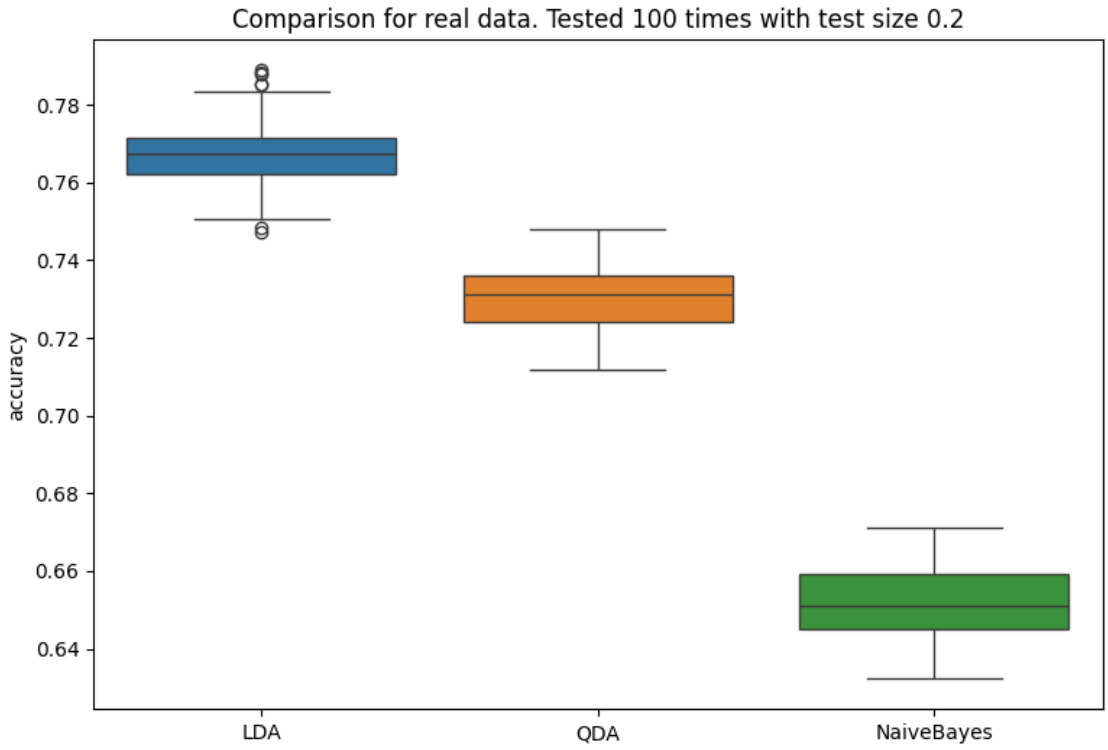
[14]:
```
X, y = Dataset.load_data(44125)
```

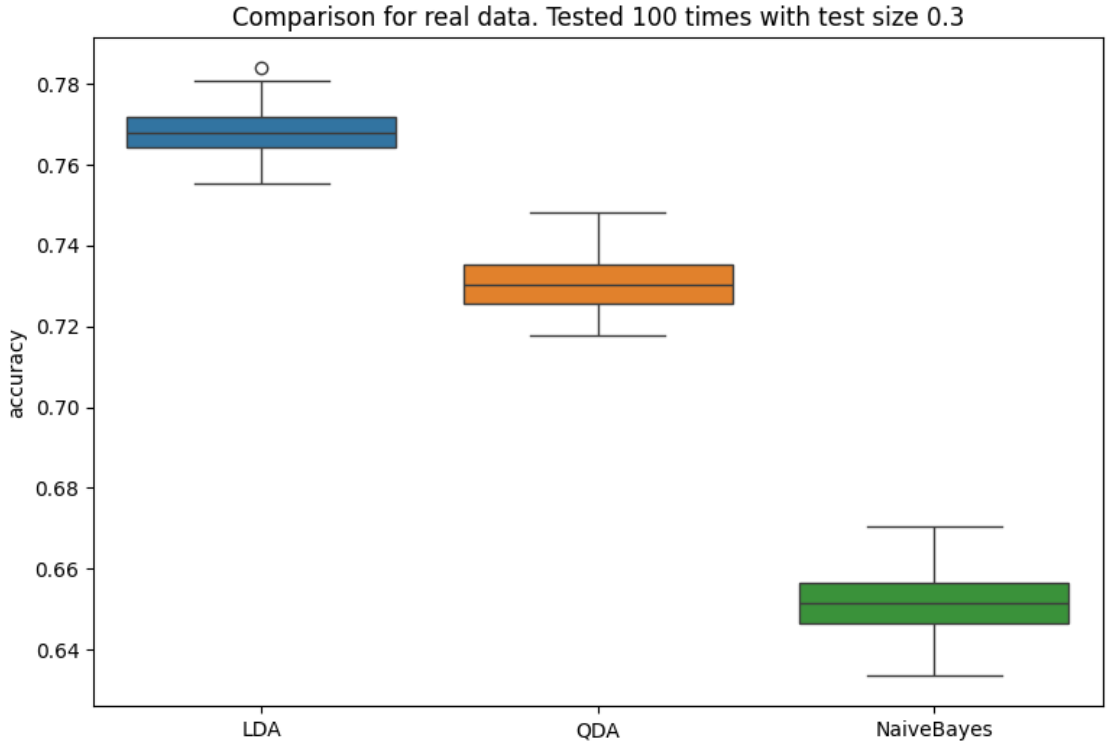
[15]:
```
print(X.shape)
print(y.shape)
```

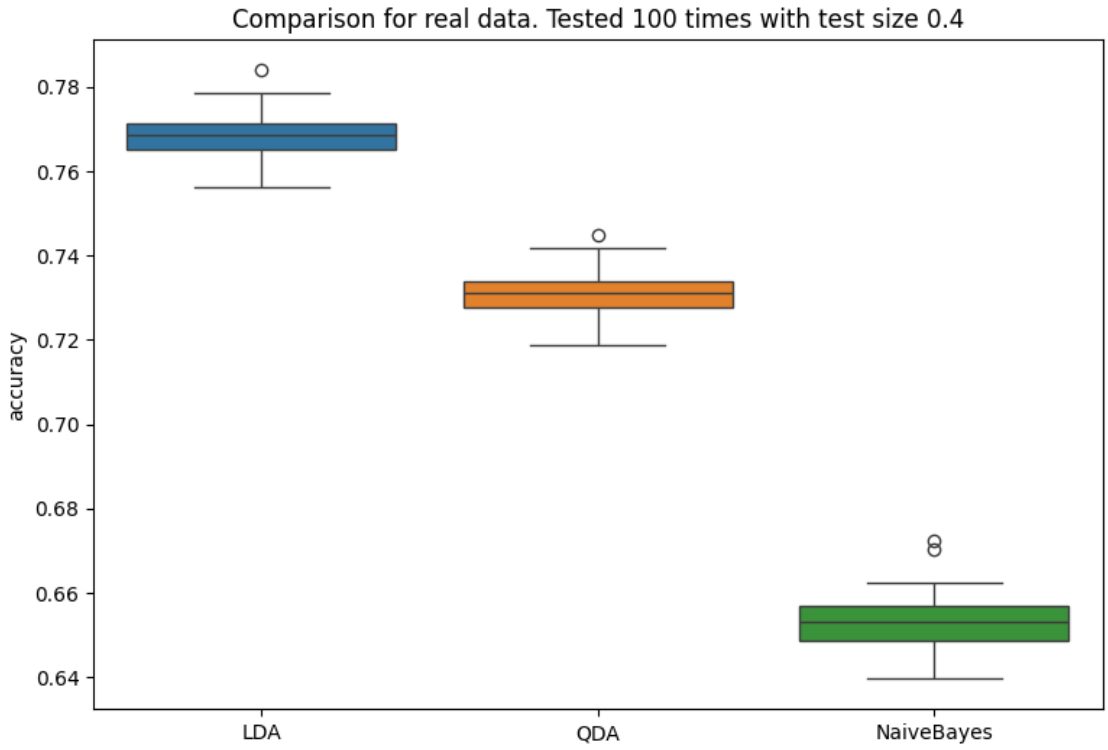
```
(13376, 10)
(13376,)
```

[17]:
```
for test_size in test_sizes:
    tester.test_real_data(X, y, 100, test_size)
```




Comparison for real data. Tested 100 times with test size 0.2
accuracy
0.78
0.76
0.74
0.72
0.70
0.68
0.66
0.64
LDA
QDA
NaiveBayes


Comparison for real data. Tested 100 times with test size 0.3
accuracy
0.78
0.76
0.74
0.72
0.70
0.68
0.66
0.64
LDA
QDA
NaiveBayes



2.1 Key observations

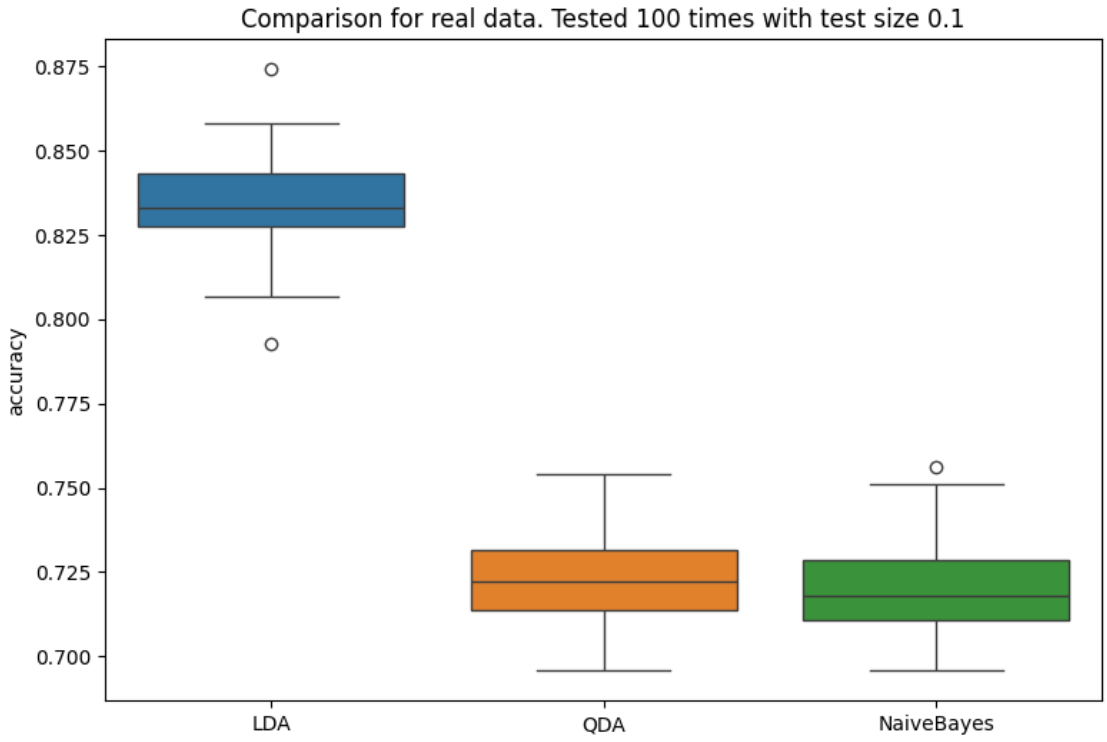
3 pol dataset

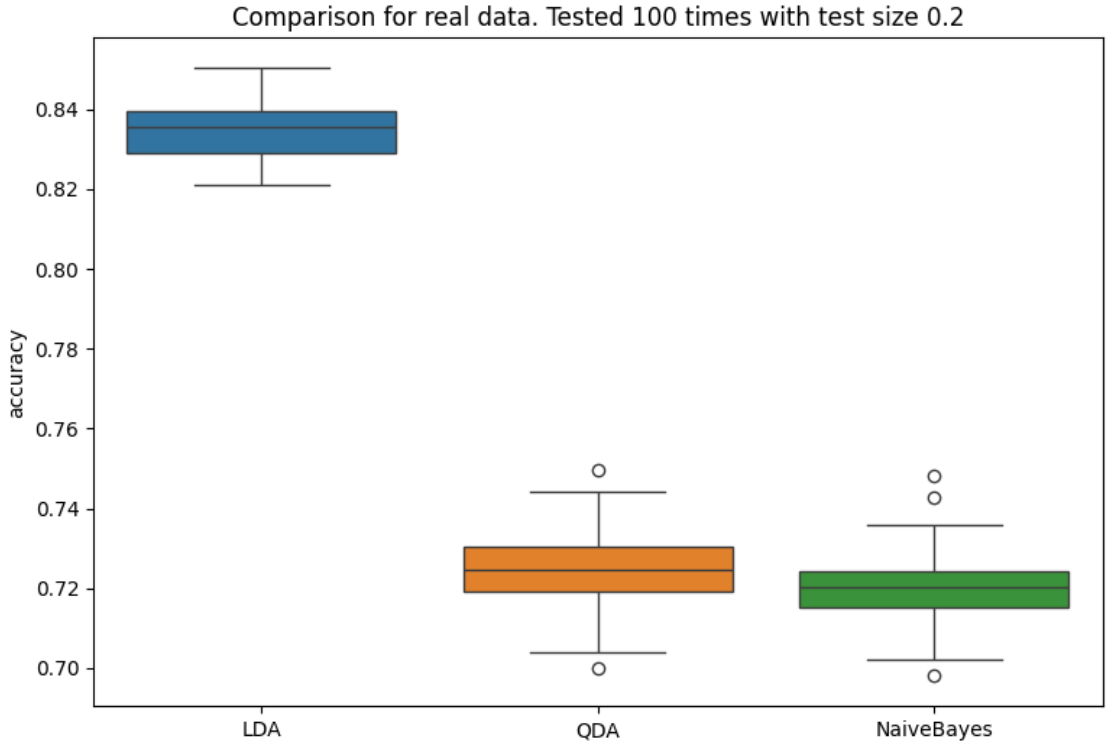
[11]:
```
X, y = Dataset.load_data(44122)
```

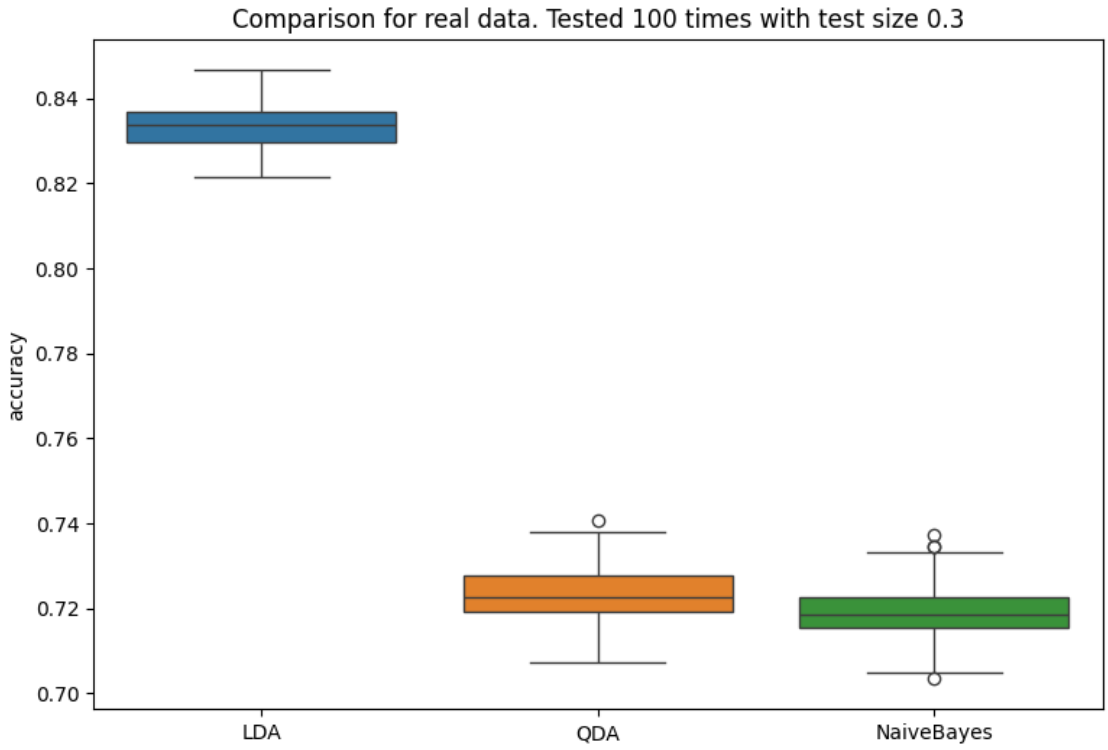
[12]:
```
print(X.shape)
print(y.shape)
```

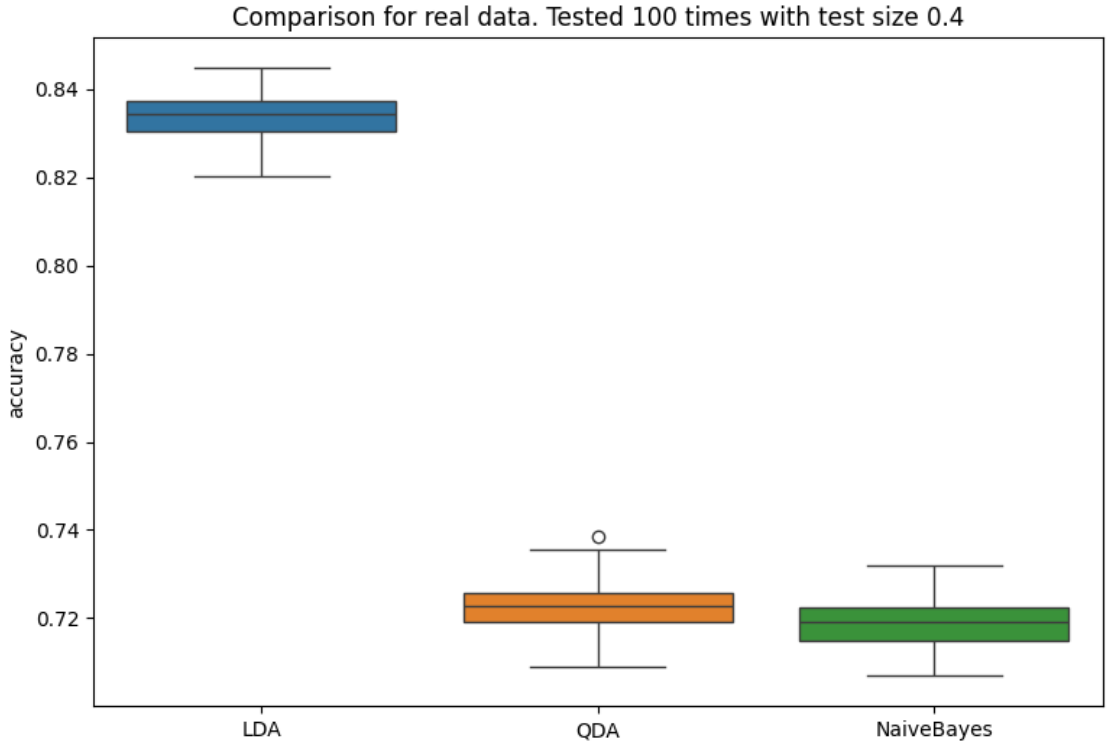
```
(10082, 26)
(10082,)
```

[13]:
```
for test_size in test_sizes:
    tester.test_real_data(X, y, 100, test_size)
```


Comparison for real data. Tested 100 times with test size 0.1
accuracy
0.875
0.850
0.825
0.800
0.775
0.750
0.725
0.700
LDA
QDA
NaiveBayes


Comparison for real data. Tested 100 times with test size 0.2
accuracy
0.84
0.82
0.80
0.78
0.76
0.74
0.72
0.70
LDA
QDA
NaiveBayes


Comparison for real data. Tested 100 times with test size 0.3
accuracy
0.84
0.82
0.80
0.78
0.76
0.74
0.72
0.70
LDA
QDA
NaiveBayes


Comparison for real data. Tested 100 times with test size 0.4
accuracy
0.84
0.82
0.80
0.78
0.76
0.74
0.72
LDA
QDA
NaiveBayes

3.1 Key observations

4 chscase_geyser1 dataset

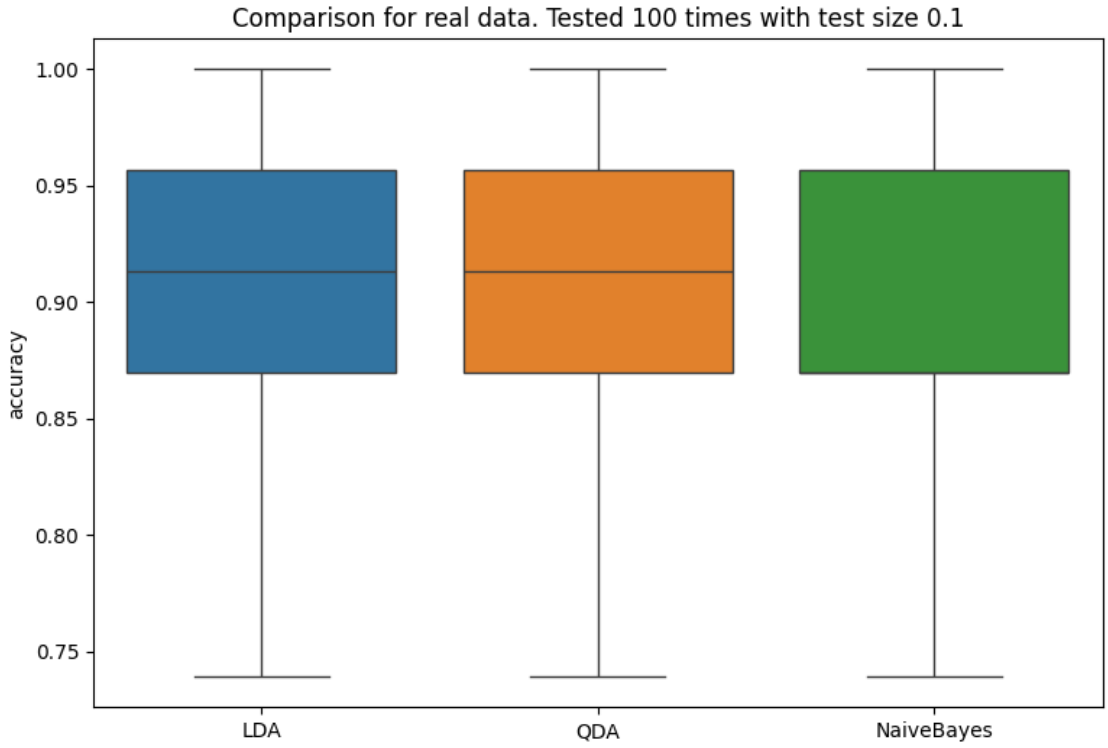
Extra case with 2 features and decision boundary plot

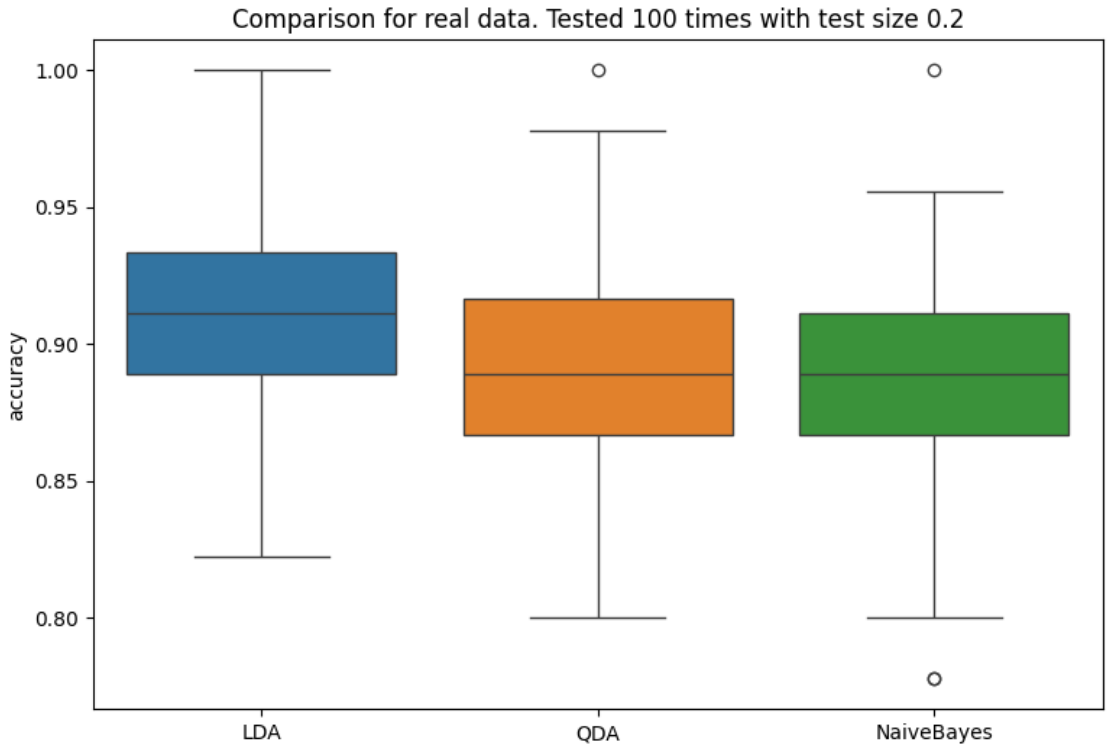
[35]:
```
X, y = Dataset.load_data(895)
```

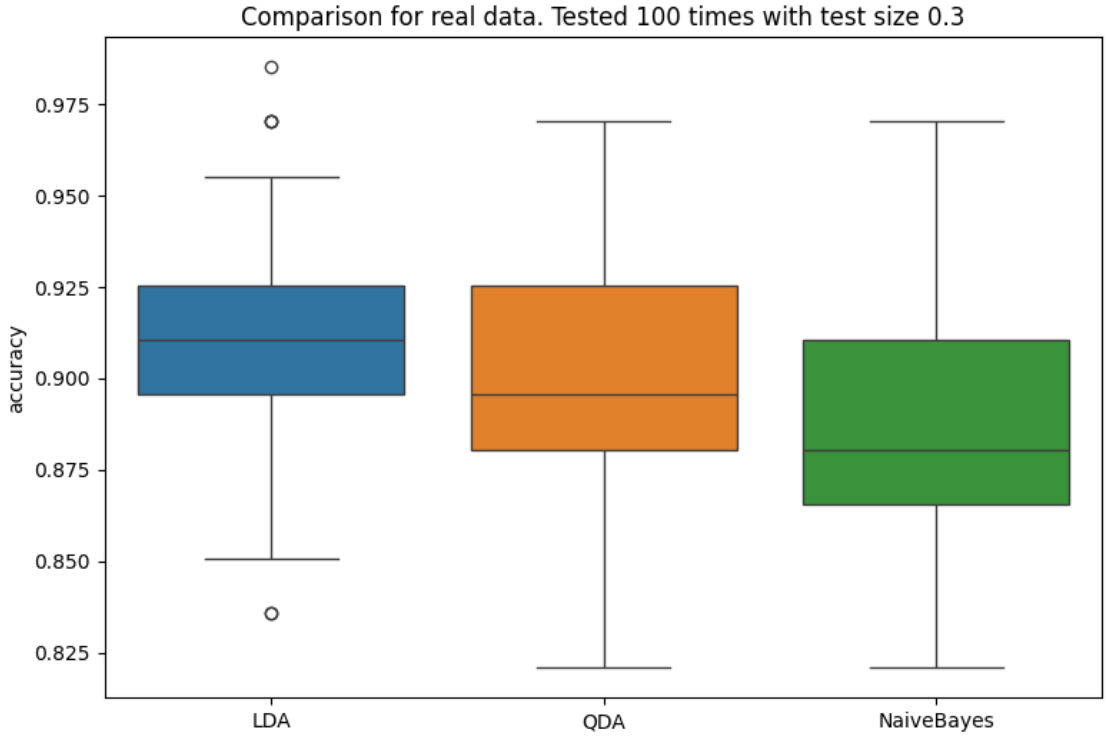
[36]:
```
print(X.shape)
print(y.shape)
```

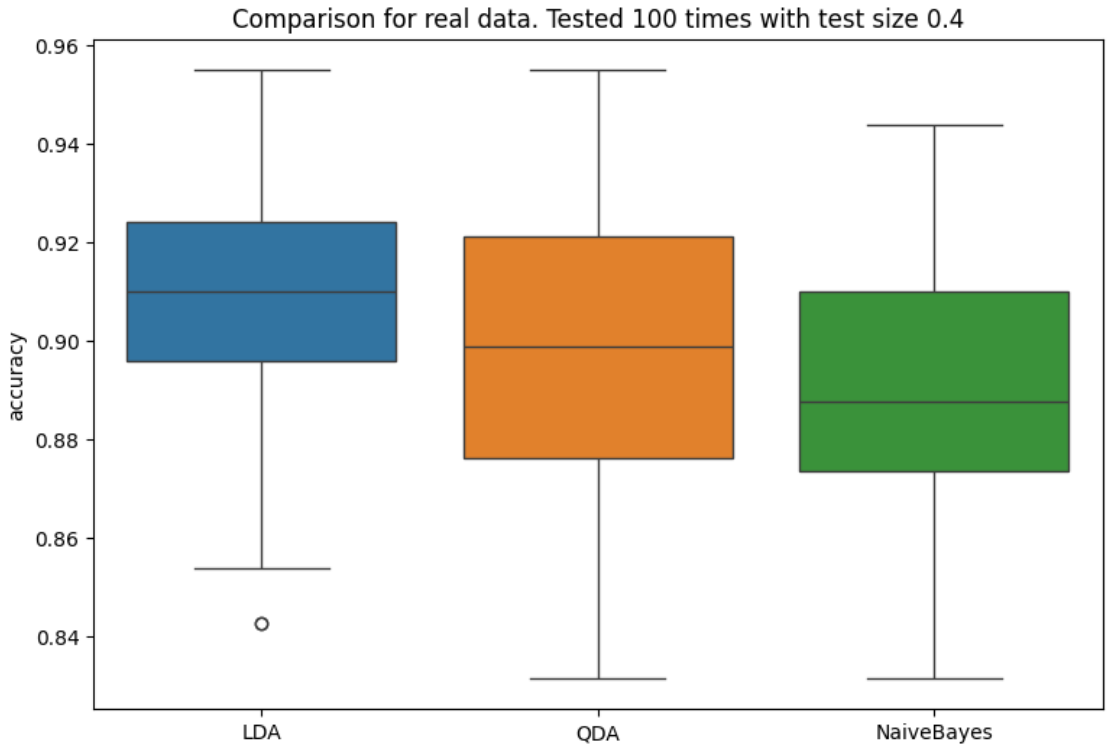
```
(222, 2)
(222,)
```

[37]:
```
for test_size in test_sizes:
    tester.test_real_data(X, y, 100, test_size)
```



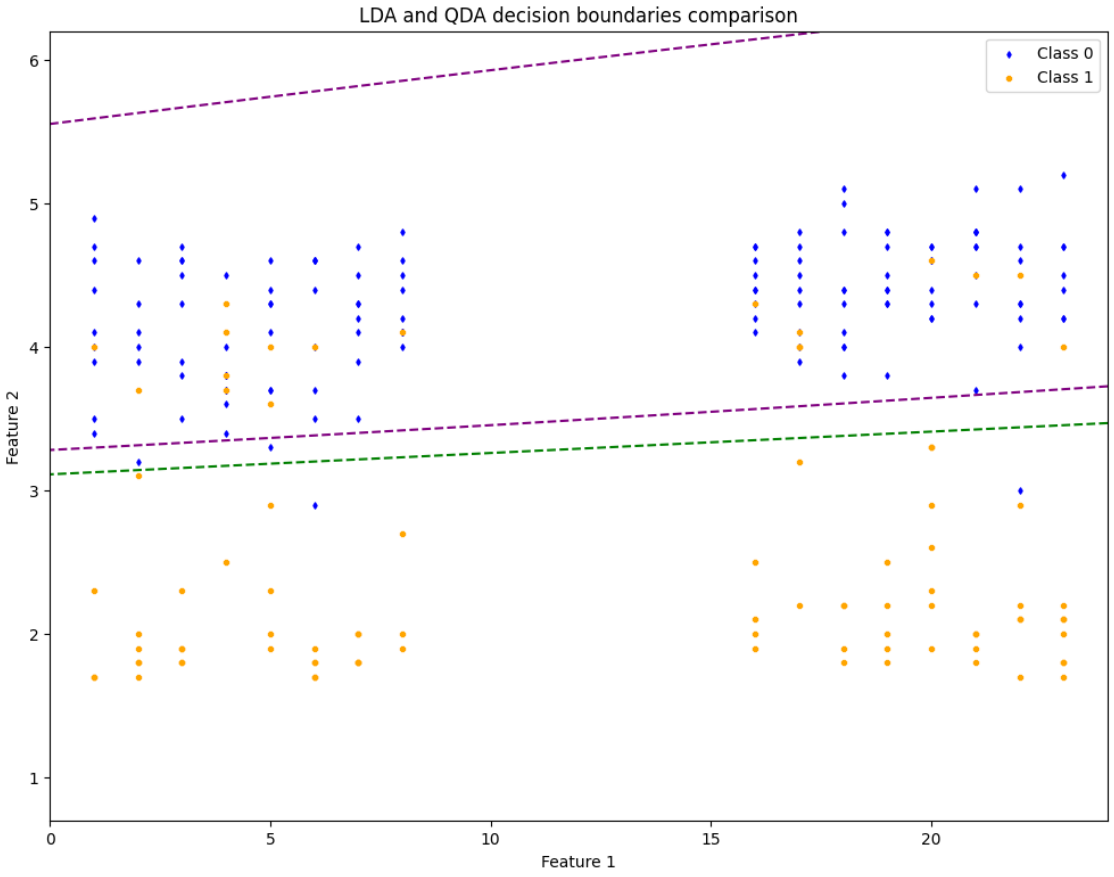

Comparison for real data. Tested 100 times with test size 0.2
accuracy
1.00
0.95
0.90
0.85
0.80
LDA
QDA
NaiveBayes


Comparison for real data. Tested 100 times with test size 0.3
accuracy
0.975
0.950
0.925
0.900
0.875
0.850
0.825
LDA
QDA
NaiveBayes



4.1 Key observations

[38]:
```
tester.plot_boundaries(X=X, y=y)
```



4.2 Key observations