

```

1 # Appendix C17 - test_measure_calculator.py
2
3 from klassify.src.measure_calculator import MeasureCalculator
4 from klassify.src.tables import Topic, Subtopic
5
6 first_set = {
7     "BernoulliNB": {
8         "cross score": 3, "precision": 1, "cross variance": 1
9     },
10    "MultinomialNB": {
11        "cross score": 2, "precision": 2, "cross variance": 2
12    }
13 }
14 second_set = {
15     "BernoulliNB": {"recall": 3, "f1": 1},
16     "MultinomialNB": {"recall": 2, "f1": 2}
17 }
18
19 # Groups two sets of measures by the algorithm type
20 def test_combine_measures():
21     CALC = MeasureCalculator()
22
23     assert CALC.combine_measures(first_set, second_set) == {
24         "BernoulliNB": {
25             "cross score": 3, "precision": 1, "recall": 3, "f1": 1, "cross variance": 1
26         },
27         "MultinomialNB": {
28             "cross score": 2, "precision": 2, "recall": 2, "f1": 2, "cross variance": 2
29         }
30     }
31
32 # Store sets of measures
33 def test_add_measures():
34     CALC = MeasureCalculator()
35
36     CALC.add_measures(first_set, second_set)
37
38     assert CALC.measures == {
39         "BernoulliNB": {
40             "cross score": [3], "precision": [1], "recall": [3], "f1": [1], "cross variance": [1]
41         },
42         "MultinomialNB": {
43             "cross score": [2], "precision": [2], "recall": [2], "f1": [2], "cross variance": [2]
44         }
45     }
46

```