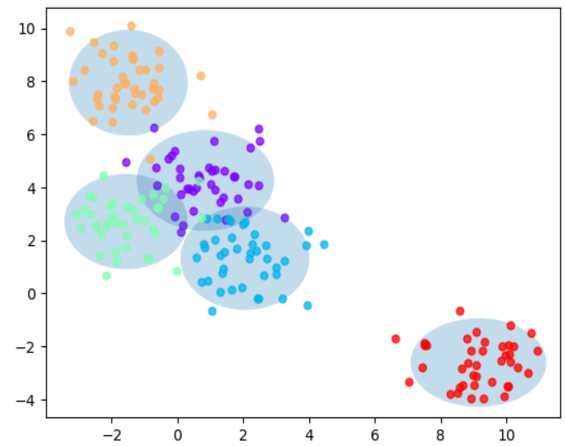
Assignment1:

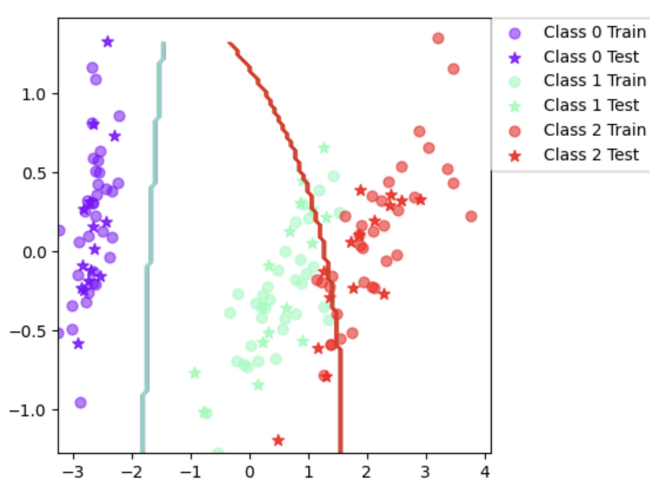


Assignment3:

Iris dataset-

```
testClassifier(BayesClassifier(), dataset='iris', split=0.7)
```

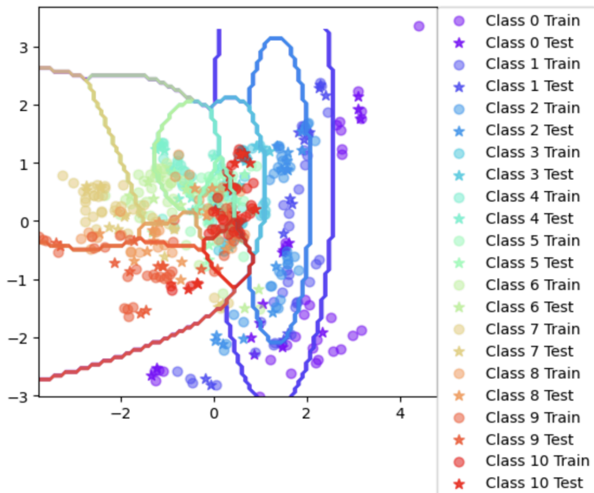
```
Trial: 0 Accuracy 84.4
Trial: 10 Accuracy 95.6
Trial: 20 Accuracy 93.3
Trial: 30 Accuracy 86.7
Trial: 40 Accuracy 88.9
Trial: 50 Accuracy 91.1
Trial: 60 Accuracy 86.7
Trial: 70 Accuracy 91.1
Trial: 80 Accuracy 86.7
Trial: 90 Accuracy 91.1
Final mean classification accuracy  89 with standard deviation 4.16
```



Vowel dataset-

```
testClassifier(BayesClassifier(), dataset='vowel', split=0.7)
```

```
Trial: 0 Accuracy 61
Trial: 10 Accuracy 66.2
Trial: 20 Accuracy 74
Trial: 30 Accuracy 66.9
Trial: 40 Accuracy 59.7
Trial: 50 Accuracy 64.3
Trial: 60 Accuracy 66.9
Trial: 70 Accuracy 63.6
Trial: 80 Accuracy 62.3
Trial: 90 Accuracy 70.8
Final mean classification accuracy  64.7 with standard deviation 4.03
```



(1) When can a feature independence assumption be reasonable and when not?

Reasonable:

- Feature independence is reasonable when we are working with a dataset where we have reason to believe there is a high likelihood that features are independent of each other.
    - This might be appropriate when each feature contributes independently to the outcome.
- Assuming feature independence can also be reasonable in cases where we want to simplify our models and make computation less expensive, because even when it's not an entirely correct assumption, we can still sometimes get pretty decent results

Unreasonable:

- It might not be reasonable when features have a known relationship or dependence on one another. For example: In a dataset of images , adjacent pixels are often correlated , so the independence assumption would not hold. The same would probably apply for a dataset consisting of words and text, since the meaning of words is heavily correlated to their context.

(2) How does the decision boundary look for the Iris dataset? How could one improve the classification results for this scenario by changing classifier or, alternatively, manipulating the data?

The decision boundary of the iris dataset is pretty clearly defined between class 0 and the other classes, but it is not linear with clear separations for all the classes.

There is an overlap between class 1 and class 2 which might imply that the classifier has difficulty in distinguishing them but not for the class 0 and class 1.

The classification could be improved by changing the classifier into one that can give classify more complex datasets, for example one of the SVM classifiers used in the last lab (such as SVM with the RBF kernel).

Another way of improving the classification could be through manipulating the data's features so that they become more separable, for example by combining or removing certain features.

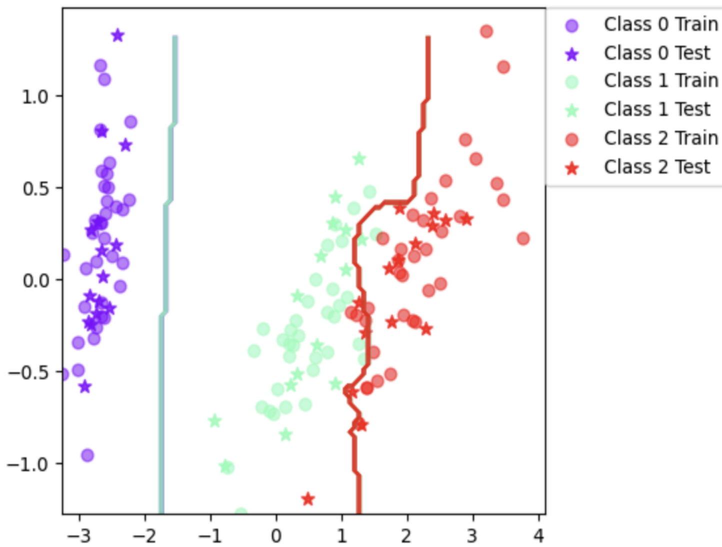
If we're having problems classifying data due to the Curse of Dimensionality (so we're in a high-dimensional space), we could perform dimensionality reduction to transform the dataset into a lower-dimensional space and remove some features while retaining some important properties of the data.

Assignment5 :

Iris dataset:

```
testClassifier(BoostClassifier(BayesClassifier(), T=10), dataset='iris',split=0.7)
```

```
Trial: 0 Accuracy 97.8
Trial: 10 Accuracy 100
Trial: 20 Accuracy 95.6
Trial: 30 Accuracy 91.1
Trial: 40 Accuracy 97.8
Trial: 50 Accuracy 97.8
Trial: 60 Accuracy 91.1
Trial: 70 Accuracy 93.3
Trial: 80 Accuracy 95.6
Trial: 90 Accuracy 93.3
Final mean classification accuracy  94.5 with standard deviation 2.73
```

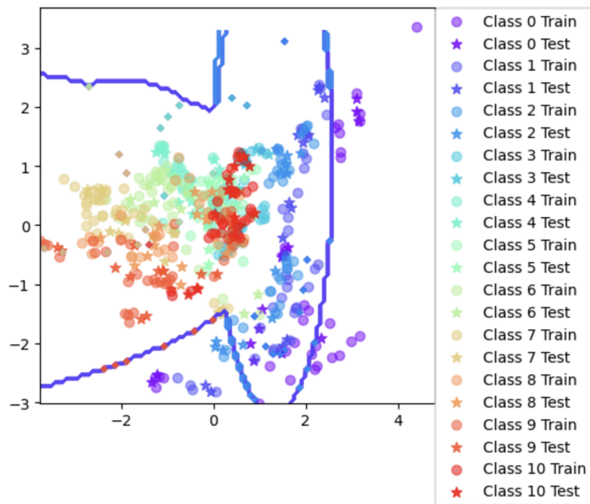


Vowel dataset

:
```
testClassifier(BoostClassifier(BayesClassifier(), T=10), dataset='vowel',split=0.7)
```

```
Trial: 0 Accuracy 77.9
Trial: 10 Accuracy 87.7
Trial: 20 Accuracy 85.1
Trial: 30 Accuracy 76.6
Trial: 40 Accuracy 73.4
Trial: 50 Accuracy 75.3
Trial: 60 Accuracy 81.2
Trial: 70 Accuracy 85.7
Trial: 80 Accuracy 77.9
Trial: 90 Accuracy 83.8
Final mean classification accuracy  80.2 with standard deviation 3.7
```

:



(1) Is there any improvement in classification accuracy? Why/why not?

Yes, the accuracy improvement is rather noticeable. Before, they were mostly in the 80-90 accuracy range for Iris, and now, they're all in the 90-range (went from mean 89 to 94.5). For the Vowel dataset, it went from mean 64.7 to 80.2

It's improving because we're using the Adaboost algorithm, which takes takes the misclassified data points from previous iterations and gives them higher weights to train the later iteration of classifiers better.

(2) Plot the decision boundary of the boosted classifier on iris and compare it with that of the basic. What differences do you notice? Is the boundary of the boosted version more complex?

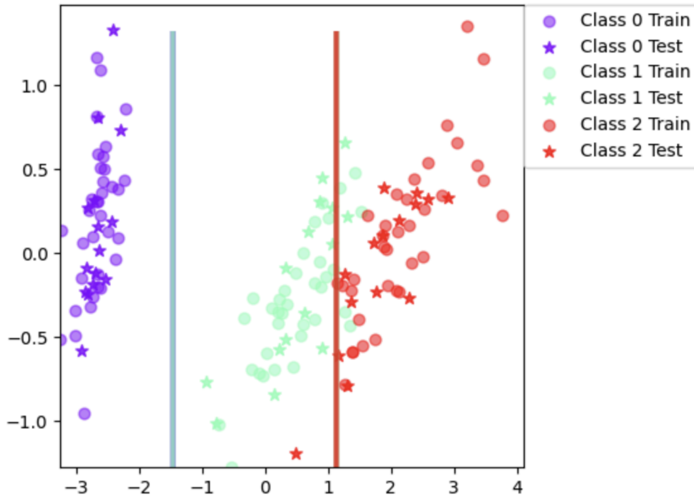
The decision boundary is more complex and more accurate than before.

(3) Can we make up for not using a more advanced model in the basic classifier (e.g. independent features) by using boosting?

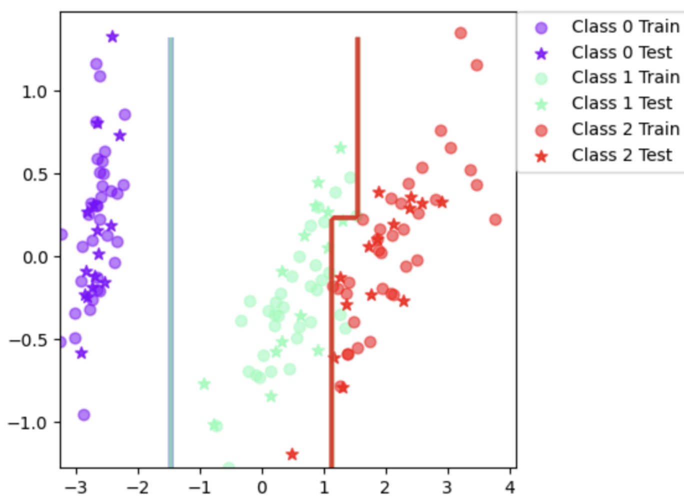
Yes, boosting is one alternative to using a more advanced model.

Assignment6:

Iris dataset:
Decision tree mean: 92.4



Boosted decision tree mean: 94.5



Vowel dataset:
Decision tree mean: 64.1
Boosted decision tree mean: 86.7

(1) Is there any improvement in classification accuracy? Why/why not?

There is still improvement with boosting, though it isn't as significant for the Iris data set. It's bigger for the Vowel data set, however, where there is a lot of overlap between data points.

(2) Plot the decision boundary of the boosted classifier on iris and compare it with that of the basic. What differences do you notice? Is the boundary of the boosted version more complex?

The decision boundary is just a straight line in the basic version, while the boosted line is *slightly* more complex.

(3) Can we make up for not using a more advanced model in the basic classifier (e.g. independent features) by using boosting?

Yes, boosting is one alternative to using a more advanced model.

Assignment 7:

If you had to pick a classifier, naive Bayes or a decision tree or the boosted versions of these, which one would you pick? Motivate from the following criteria:

• Outliers
We would probably want Naive Bayes since Decision Trees more easily overfit.

We might not want to boost, since boosting will make the classifier focus more on the misclassified data points which might cause it to overfit as well.

• Irrelevant inputs: part of the feature space is irrelevant

Unboosted Naive Bayes. Naive Bayes ignores irrelevant features, so it would work better for irrelevant inputs, and since boosting makes it focus more on misclassified samples (such as the irrelevant inputs), we might not want to boost.

Decision trees, meanwhile, tend to overfit on such inputs.

• Predictive power

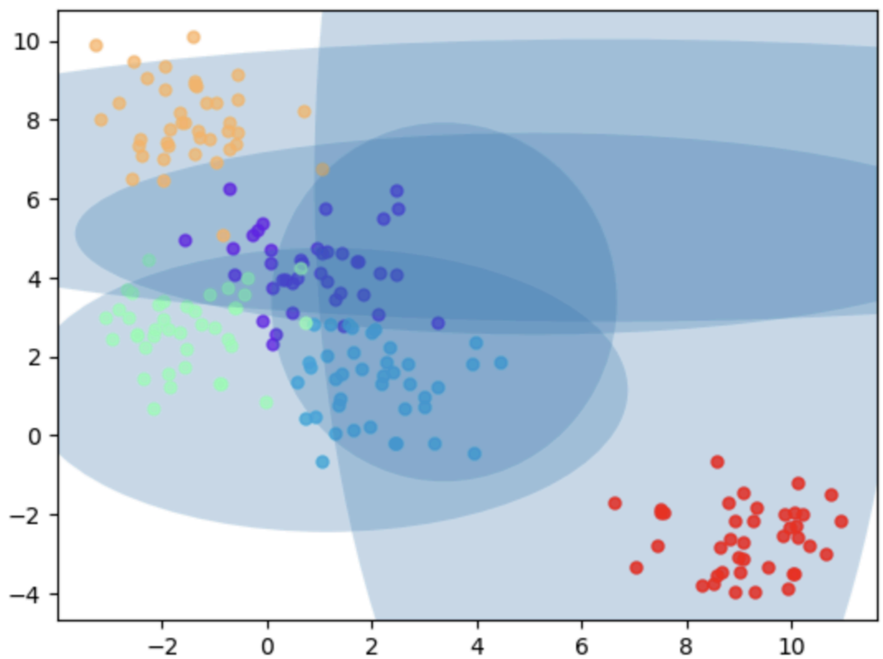
We might want a decision tree with boosting, since as shown on these data sets it had slightly better accuracy.

• Mixed types of data: binary, categorical or continuous features, etc.

Boosted decision tree. Decision works immediately, while Naive Bayes might require more work to work with different data types.

• Scalability: the dimension of the data, D, is large or the number of instances, N, is large, or both.

Unboosted Naive Bayes, since it's pretty scalable and the independence assumption simplifies things greatly. Boosting doesn't necessarily scale as well computationally, which is why we might want to avoid it.



10
8
6
4
2
0
−2
−4
−2
0
2
4
6
8
10