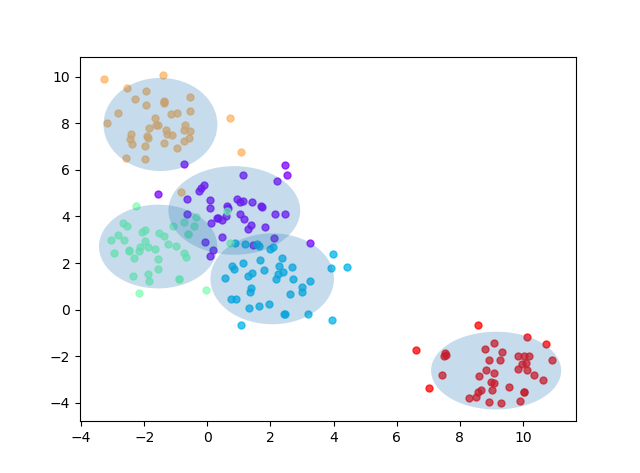
Assignment 1:

To generate some test data of the function we written.

Gaussian distributed data points together with class labels, Compute the ML-estimates for the data and plot the 95%-confidence interval using the function plotGaussians.



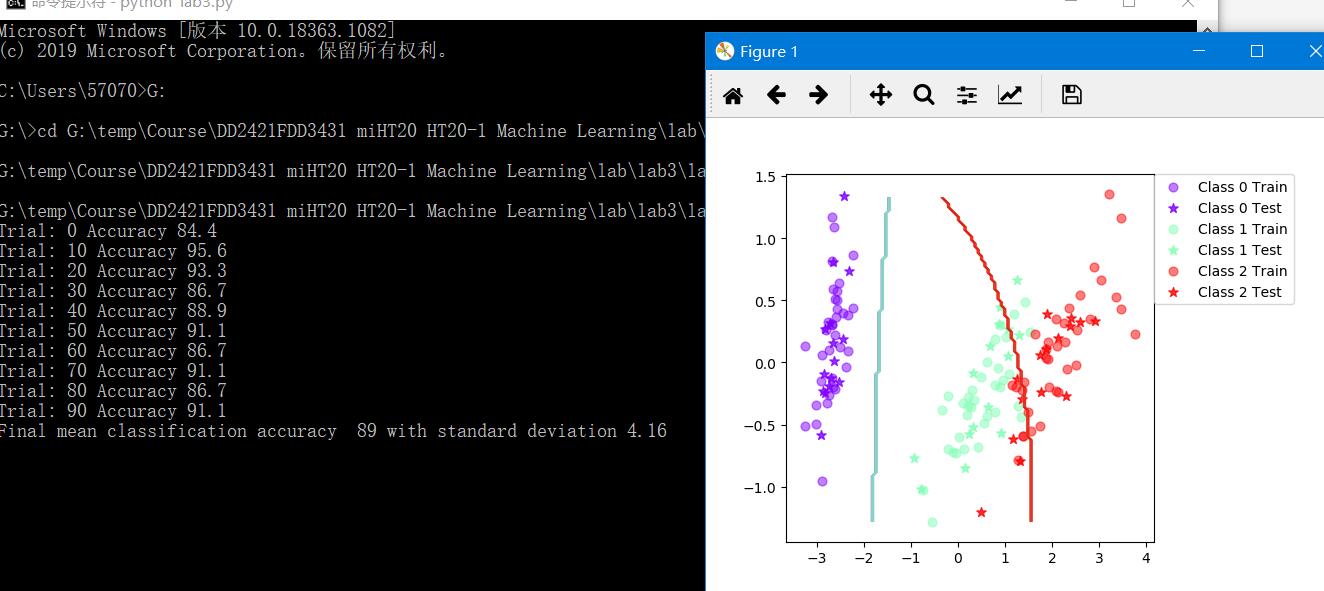
Assignment 3:

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

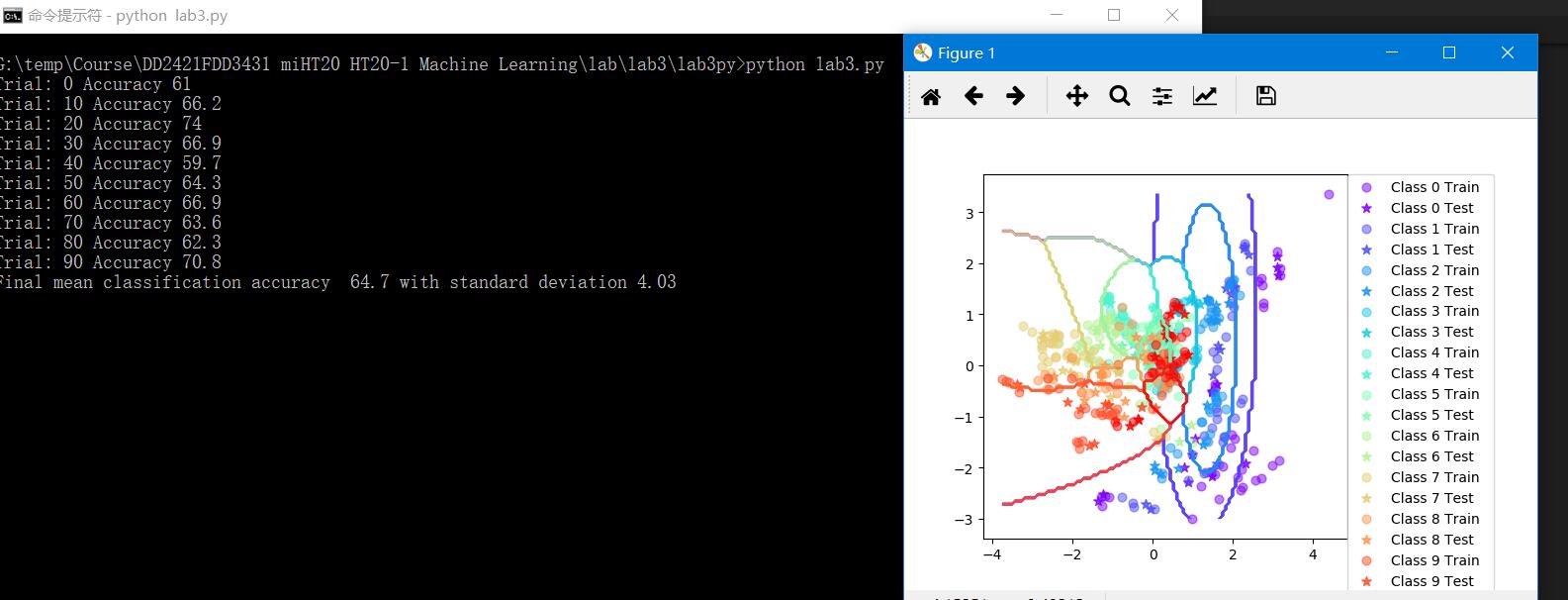
Depend on the features in the dataset, if the features are somehow dependent to each other, then the feature independence assumption isn’t reasonable. We can calculate the correlation coefficient to know their relationship.

(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 is not so fit in the Iris dataset. Maybe we can change the classifier into Decision Tree to get better performance.



**Iris dataset**

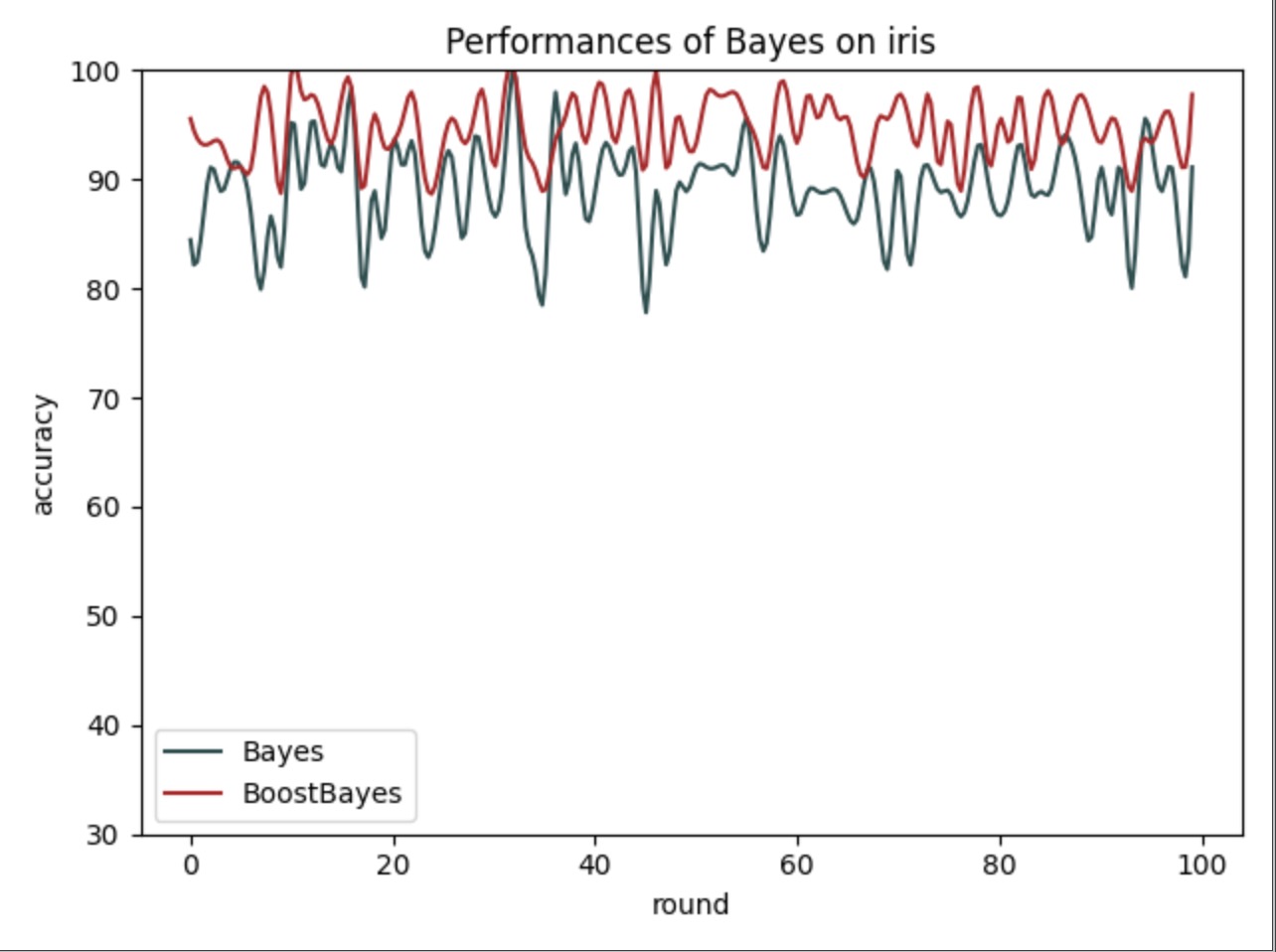


**Vowels Dataset**

Assignment 5:

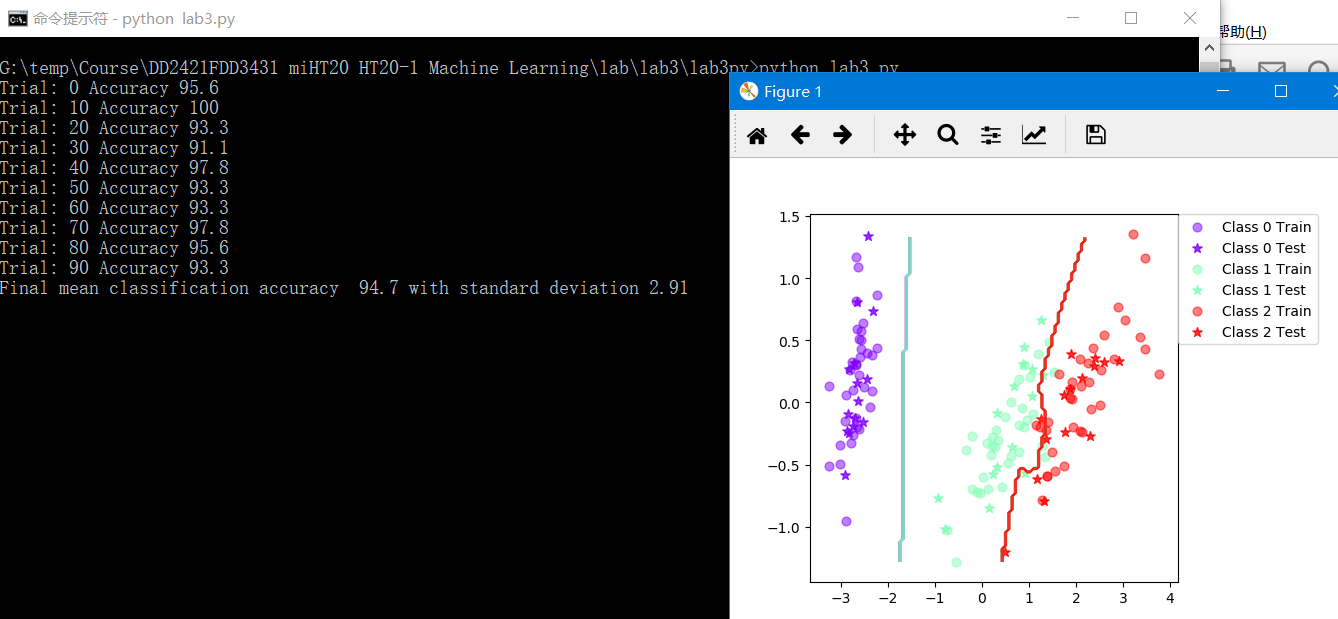
1. Is there any improvement in classification accuracy? Why/why not?

Yes, because we used boosting strategy. A diverse and complementary set of high-bias classifiers, with performance better than chance, combined by voting can produce a classifier with a low-bias.



1. 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 performance of boosted version is better now comparing with the basic classifier. The boundary didn’t get much complex but more fit to the train data now.



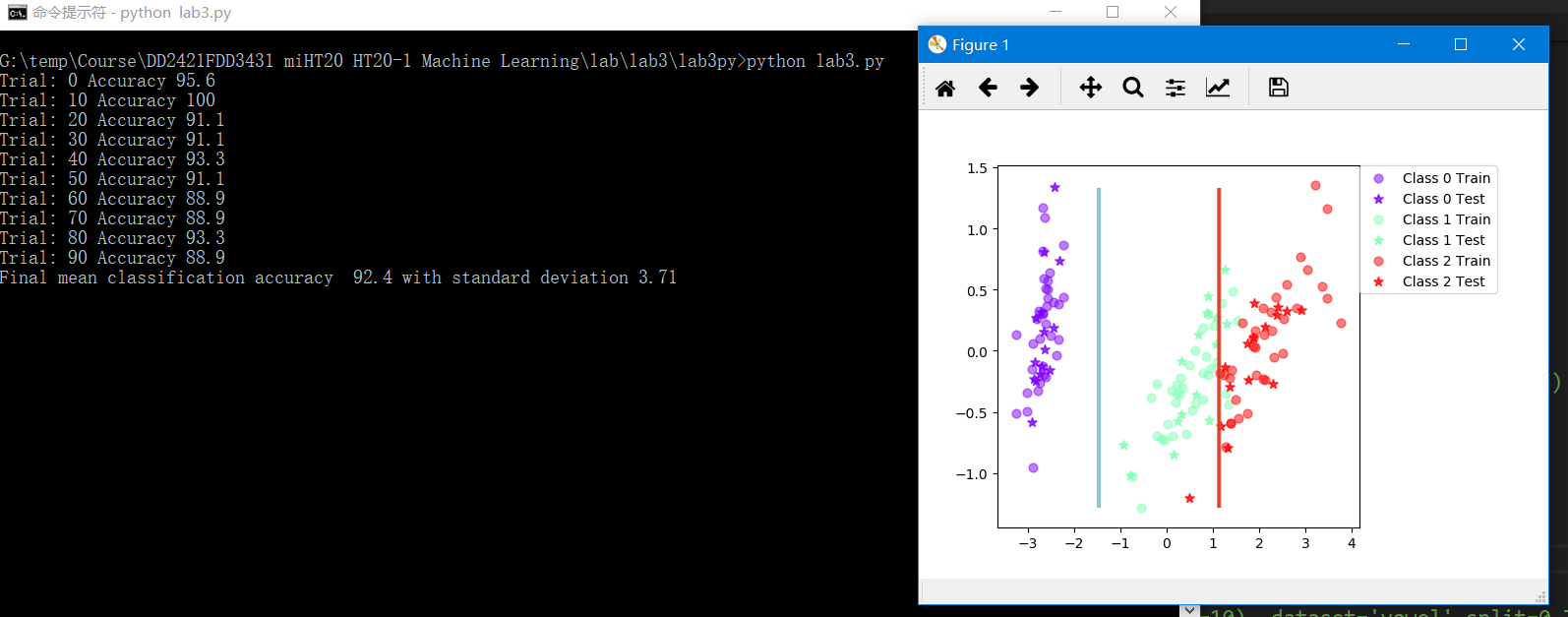
(3) Can we make up for not using a more advanced model in the basic classifier (e.g. independent features) by using boosting?

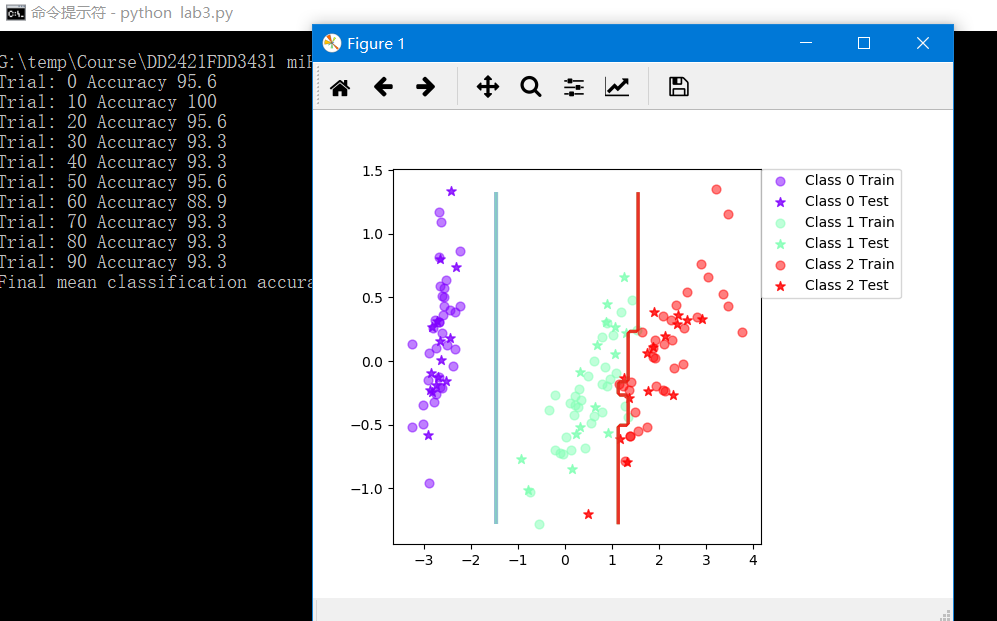
I think if the dataset is small, then a high-bias but low variance model, using boosting to lower the bias will be better. Because the data is not enough, a complex model might easily cause overfitting.

Sometimes when we are facing the more complex data, just repeatedly using basic classifier is not enough. We must use more advanced model to better represent the real situation of complex data.

Assignment 6:

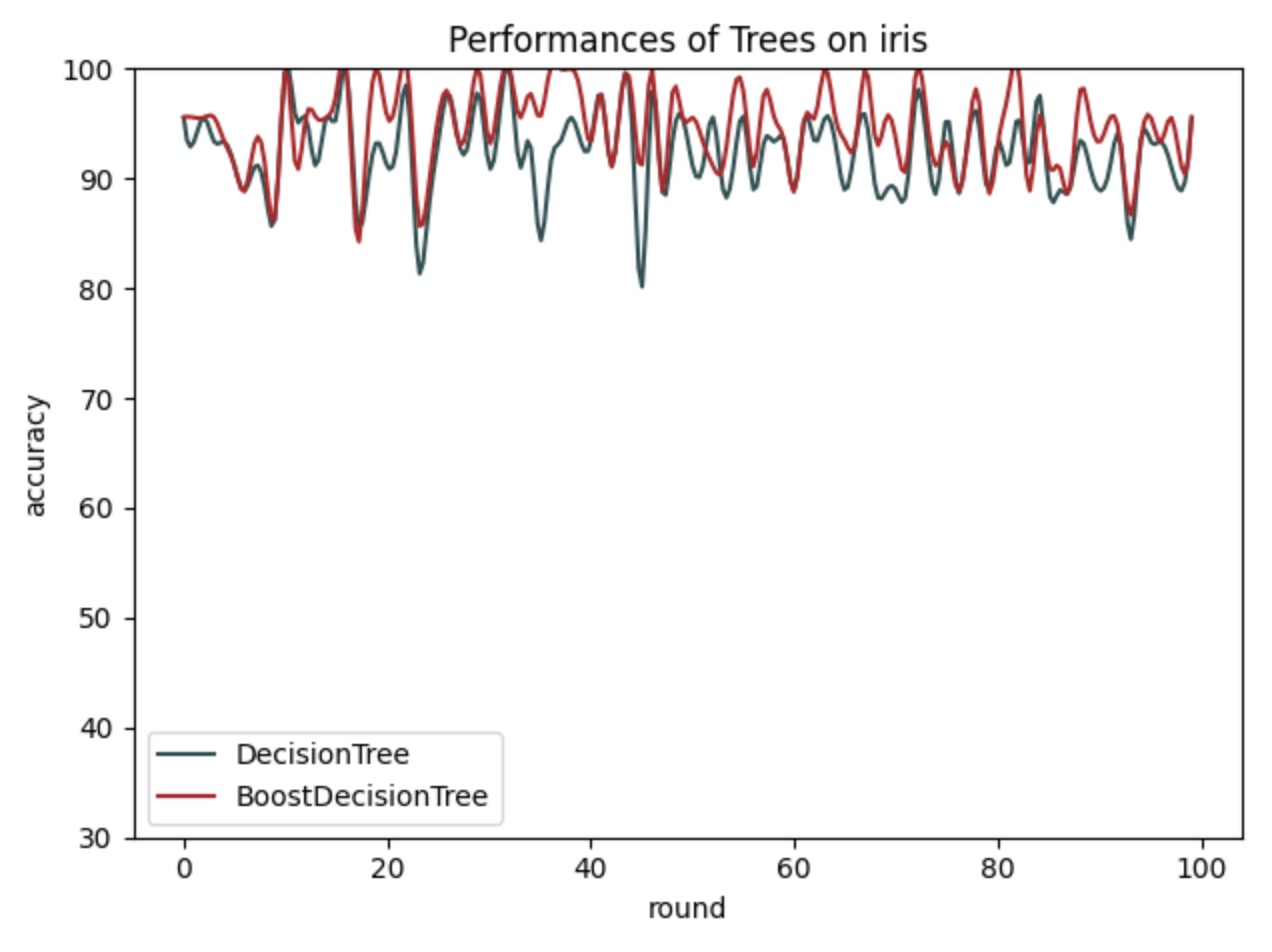
Boosting based on Decision Tree





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

A little bit better now.



1. 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?

More fit to the real data. But the boundary didn’t get more complex.

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

No. Boosting strategy do not have the ability to create a more complex model based on these simple models, it can help these basic models with high vias to get better performance resulting in low bias.

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**

Trees are not sensible to outliers. To some degree properties that don't matter won't be chosen as splits and will get eventually pruned so it's very tolerant of nonsense.

Bayes is. Loss of lots data might affect the calculation of likelihood.

**\_ Irrelevant inputs: part of the feature space is irrelevant**

A crossover feature space has negative influence on Bayes classifier. Bayes assumes that all features are irrelevant.

In this situation, boosted trees, or random forest, usually perform better.

**\_ Predictive power**

Simple decision trees tend to over fit the training data more so that other techniques which means you generally have to do tree pruning and tune the pruning procedures. Other techniques like boosting and random forest decision trees can perform quite well, and some feel these techniques are essential to get the best performance out of decision trees. Again this adds more things to understand and use to tune the tree and hence more things to implement.

I think the predictive power depends on lots of factors, mostly depends on the type of datasets. If only with a small dataset then Bayes is good, and we can use boosting to improve its performance. In a high dimension space trees are likely to perform better.

I have searched some comparison of these two methods in a large dataset, and ensemble-trees (mostly Random Forest) seem always perform better. But I actually didn’t figure it out why.

But in some cases when the important features are few ones, with a low information gain and are more likely to be pruned by decision tree, for example in the face recognition. Bayes is better in this situation.

**\_ Mixed types of data: binary, categorical or continuous features, etc.**

Decision Trees are very flexible, easy to understand, and easy to debug. They will work with classification problems and regression problems. Naive bayes will answer as a continuous classifier. There are techniques to adapt it to categorical prediction however they will answer in terms of probabilities

Decision trees can be used for both classification and regression.

Bayes is only for classification. If the feature space continuous, it has to be processed to be discrete.

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

Naive bayes does quite well when the training data doesn't contain all possibilities so it can be very good with low amounts of data. Decision trees work better with lots of data compared to Naive Bayes.  Naive Bayes have to pick the best features it will use to classify. Picking which features matter is up to you. Decisions trees will pick the best features for you from tabular data.

I think Bayes is a better choice in the small dataset, and low feature dimension. First, a large feature space is hard to remain irrelevant. Second, decision trees can help us to execute feature selection in its pruning process, and Bayes won’t do that.