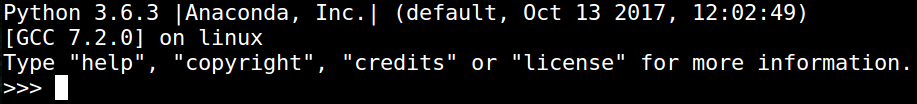
# Intro. to Machine Learning Project3

Comparison b/w KDTree , Decision Tree and Naive Bayes Classifier Report

0416324 An-Fong Hwu

## Build environment (Note, this report is written in md-like format)

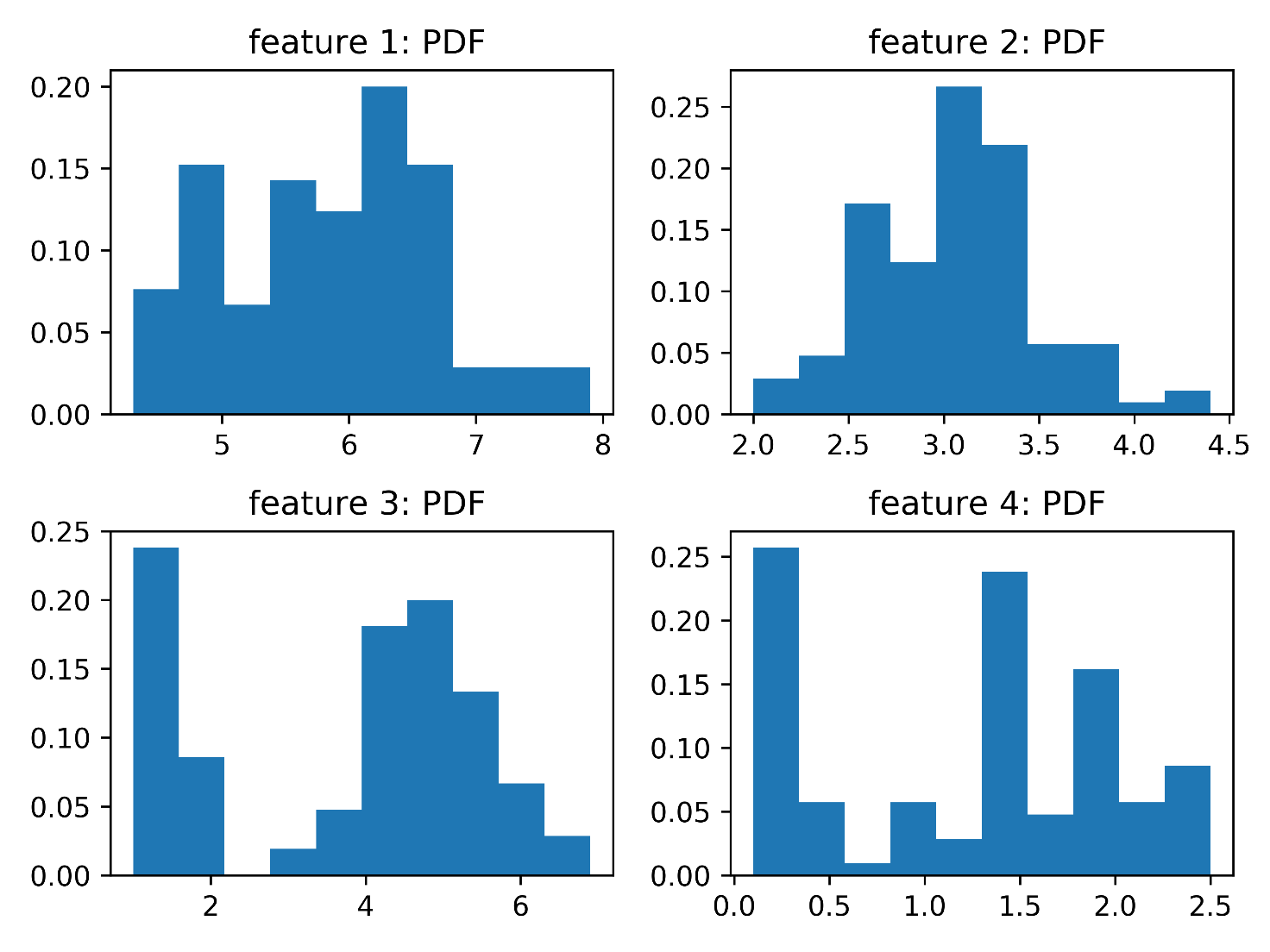
\*



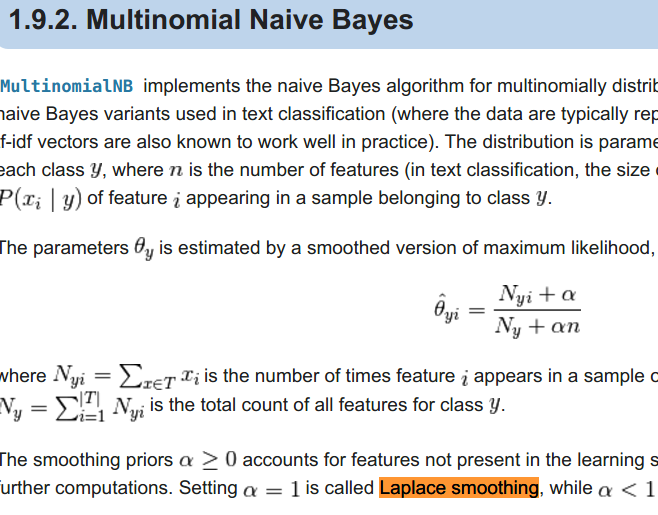
\* Packages including sklearn, numpy, scipy

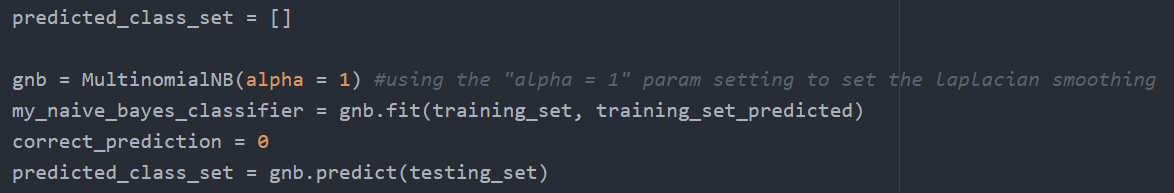
Where sklearn is used for the constructing/training /validating model, numpy and scipy for the numerical and statistical analysis.

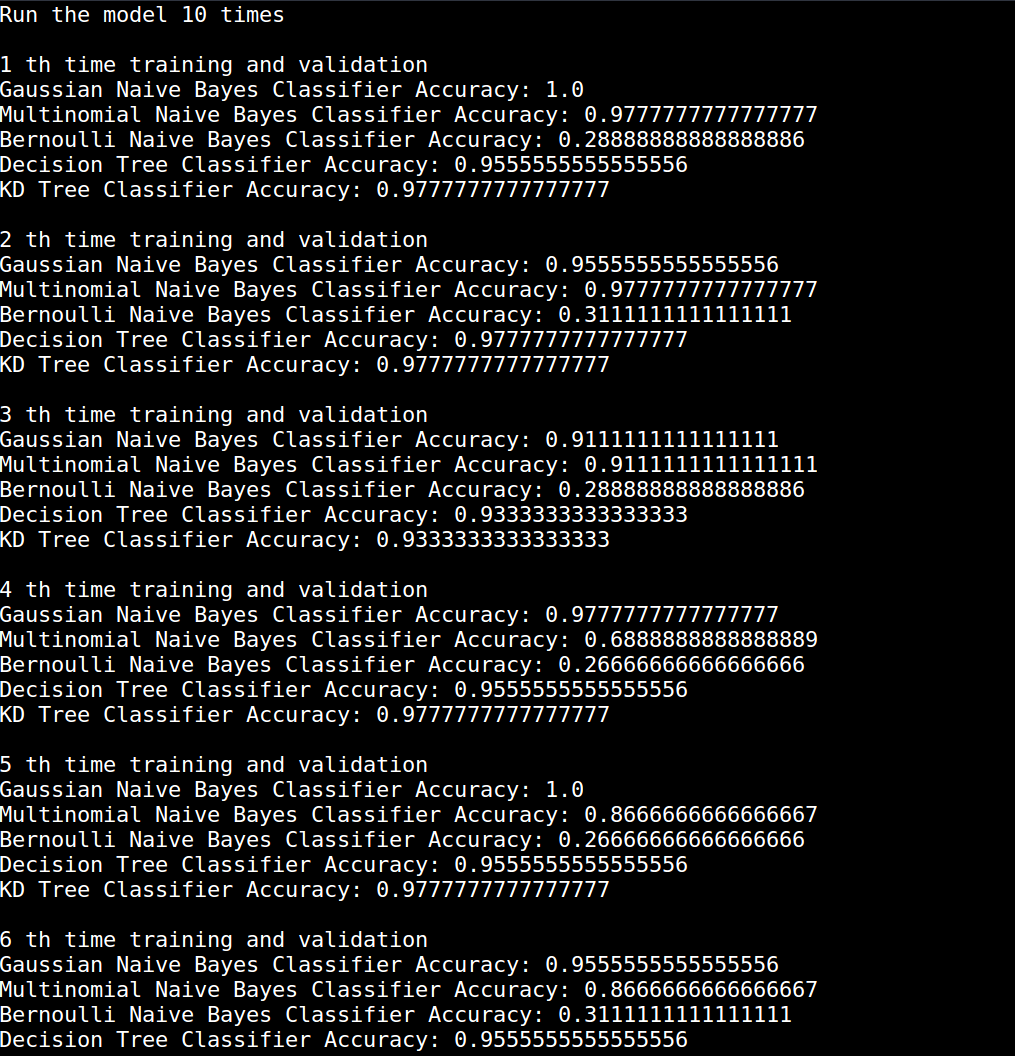
## Iris data set

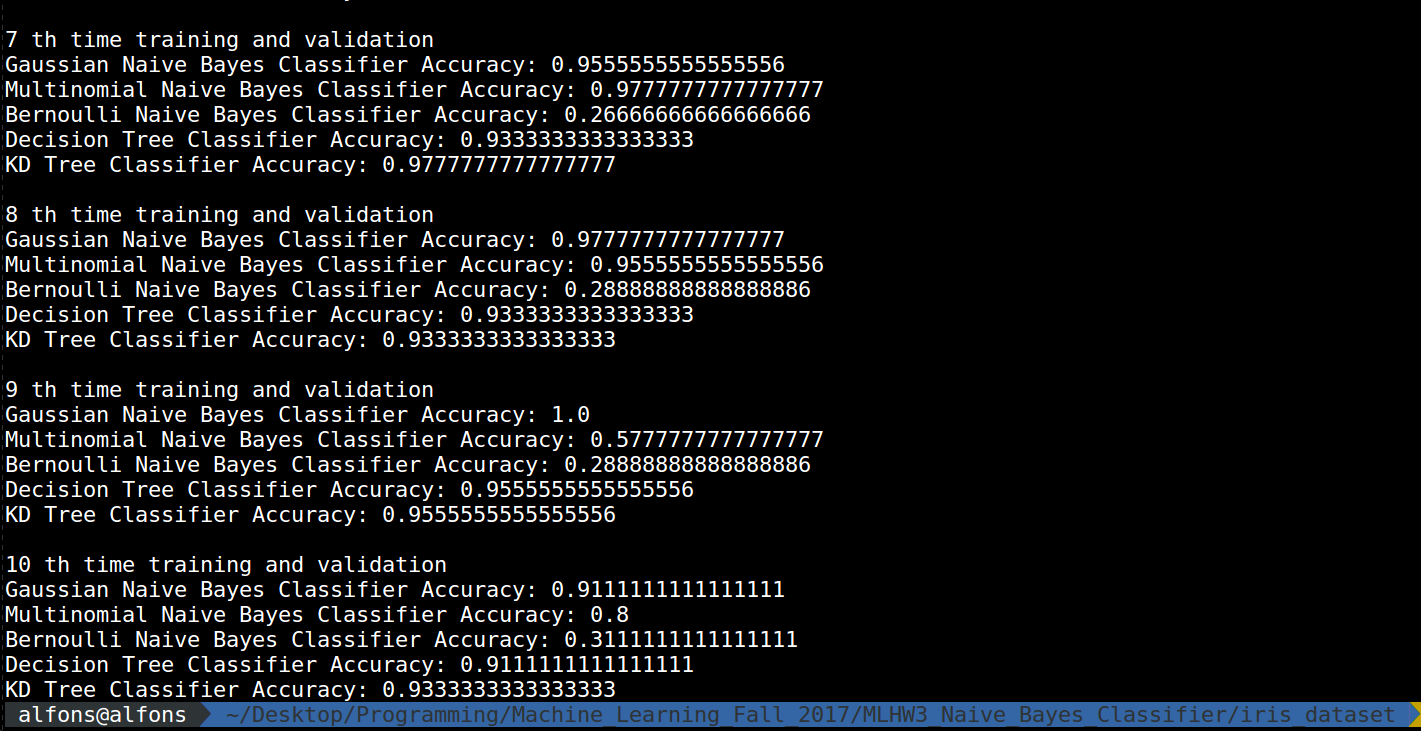
\* Probability Distribution Plot of the iris\_dataset

\* The required Laplacian Smoothing can be found in the Multinomial Naive Bayes Classifier:



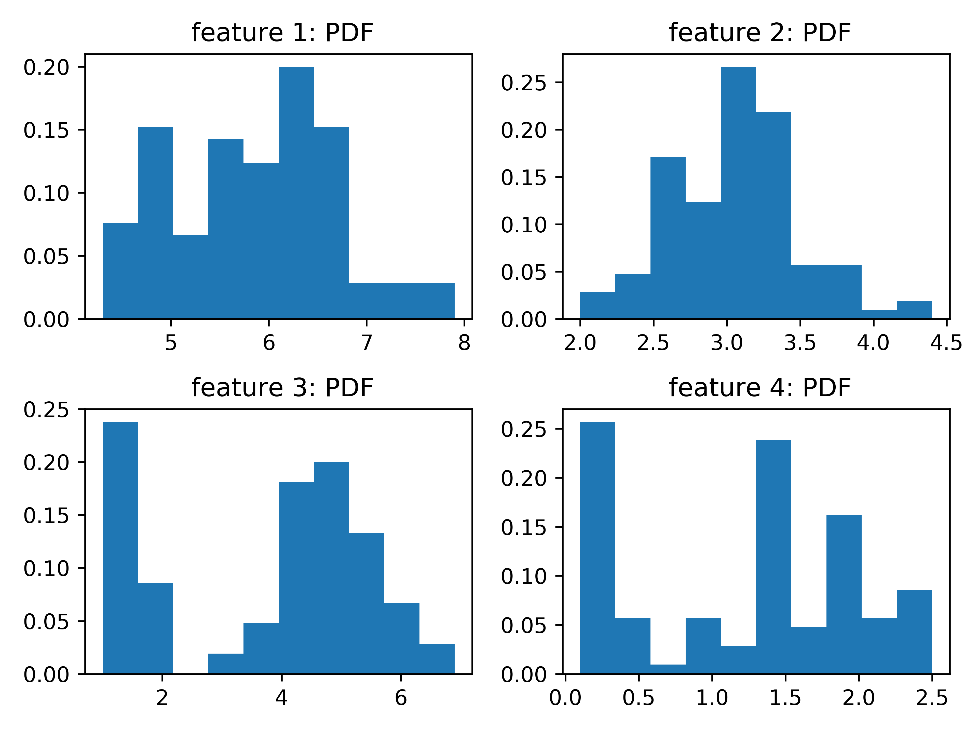






\* Observation and my inference from the prediction result of the iris dataset

As we can see, the GaussianNB , KD Tree and Decision Tree Classifier have an outstanding result compared to the Multinominal and Bernoulli NB which they both have a quite inaccurate one. Hence, what is the reason?

●First, consider the probability distribution function of the iris data set

All of the features are rather bearing the resemblance to that of the Normal Distribution, or say the Gaussian Distribution.

Namely, the GaussianNB will be quite suitable for the prediction.

But how come the Multinominal and Bernoulli NB produce such an dissatisfying result?

The Multinomial Naïve Bayes model counts how often a certain event occurs in the dataset (for example how often a certain word occurs in a document).

The Bernoulli Naïve Bayes model is similar to the Multinomial Naive Bayes model, but instead of counting how often an event occurred, it only describes whether or not an event occurred (for example whether or not a certain word occurs in a document, where it doesn't matter if it occurs once or 100000 times)

In short, the GNB group the similar data together according to the Gaussian Distribution like mean mean+-std mean+-2std and mean +-3std.

In the other two Naïve Based models, they count each distinct value, even though this is the continuous one, 0.1 0.2 0.3 0.4 will be counted to different type respectively, where they originally should produce the same result. Therefore, it undoubtedly produces a result which is quite inaccurate.

●Then we consider the KD Tree model and Decision Tree model for the iris\_dataset

The KD Tree KNN algorithm produces a slightly better accuracy then the Decision Tree while they both produces the results which are quite satisfying.

Def supervised learning (from Wikipedia): Supervised learning is the machine learning task of inferring a function from labeled training data.[1] The training data consist of a set of training examples. In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal). A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples.

And they are both supervised learning since in Decision Tree, it has the classification as input while in KNN, it self-search the NN to find the classification.

●Difference b/w regression and classcification?

Regression involves estimating or predicting a response.

Classification is identifying group membership.

Given the following

f:x→y

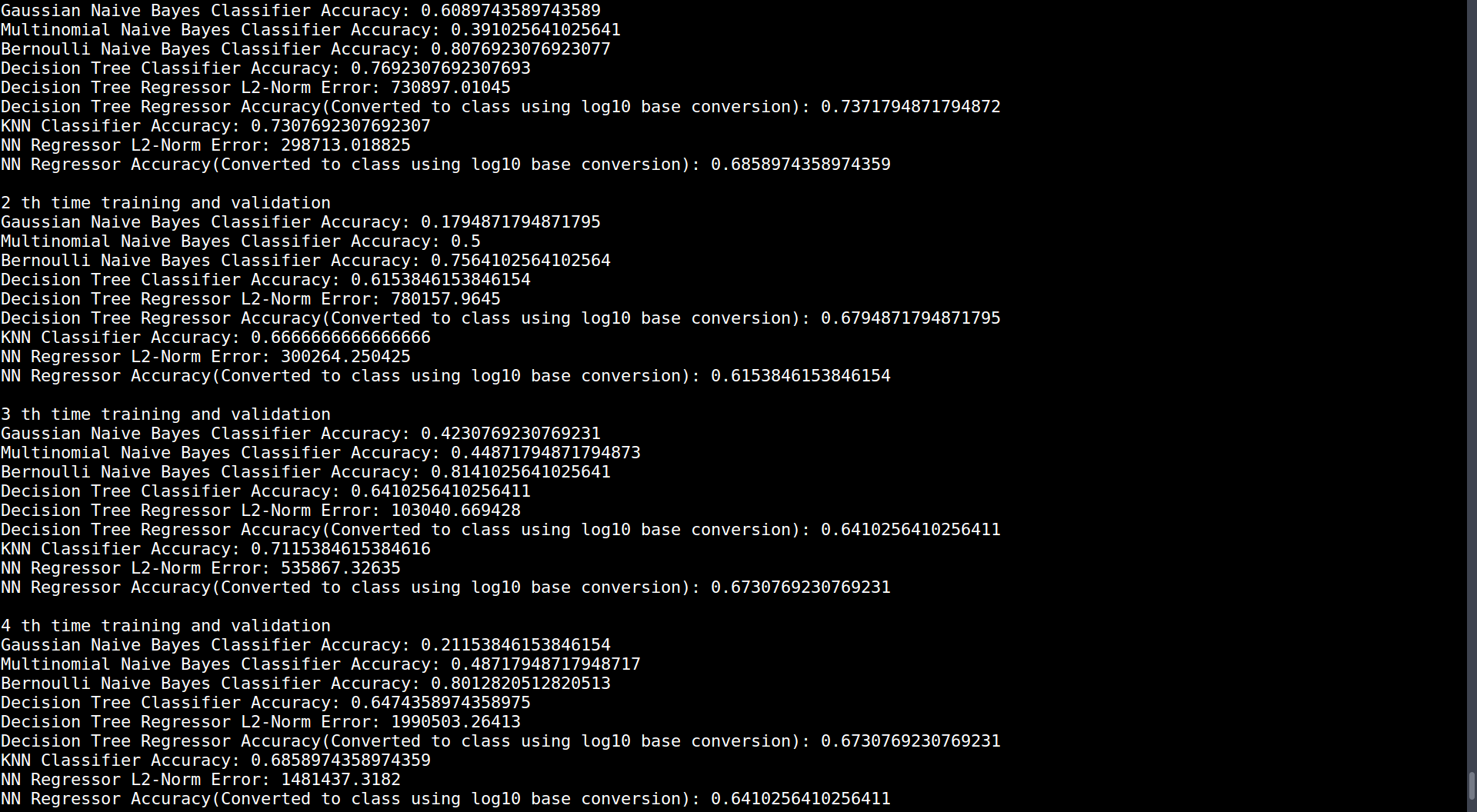
f:x→y

If y is discrete/categorical variable, then this is classification problem.

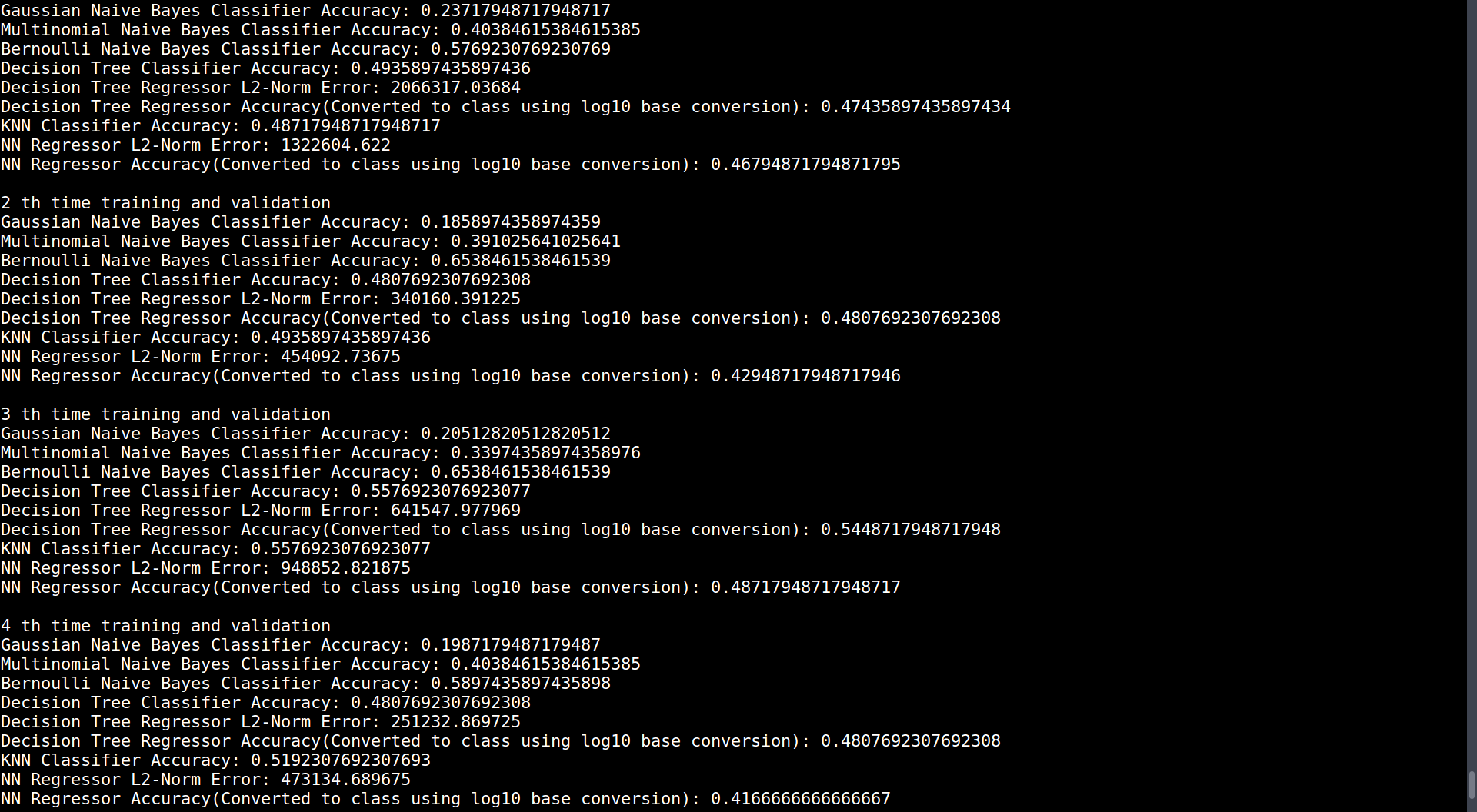
If y is real number/continuous, then this is a regression problem.

## Iris data set

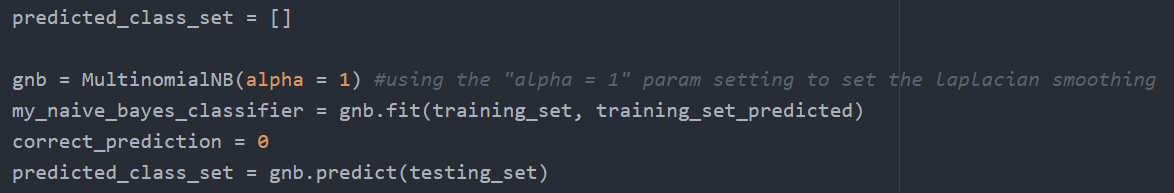
\* Probability Distribution Plot of the forestfire\_dataset



Up without round (only int) after log Down with round after log, round to the closest integer



\* The required Laplacian Smoothing



\* Observation and my inference from the prediction result of the iris dataset

● First we can see the result with only converting from float to int generates the better result than the training set with round to the nearest int.

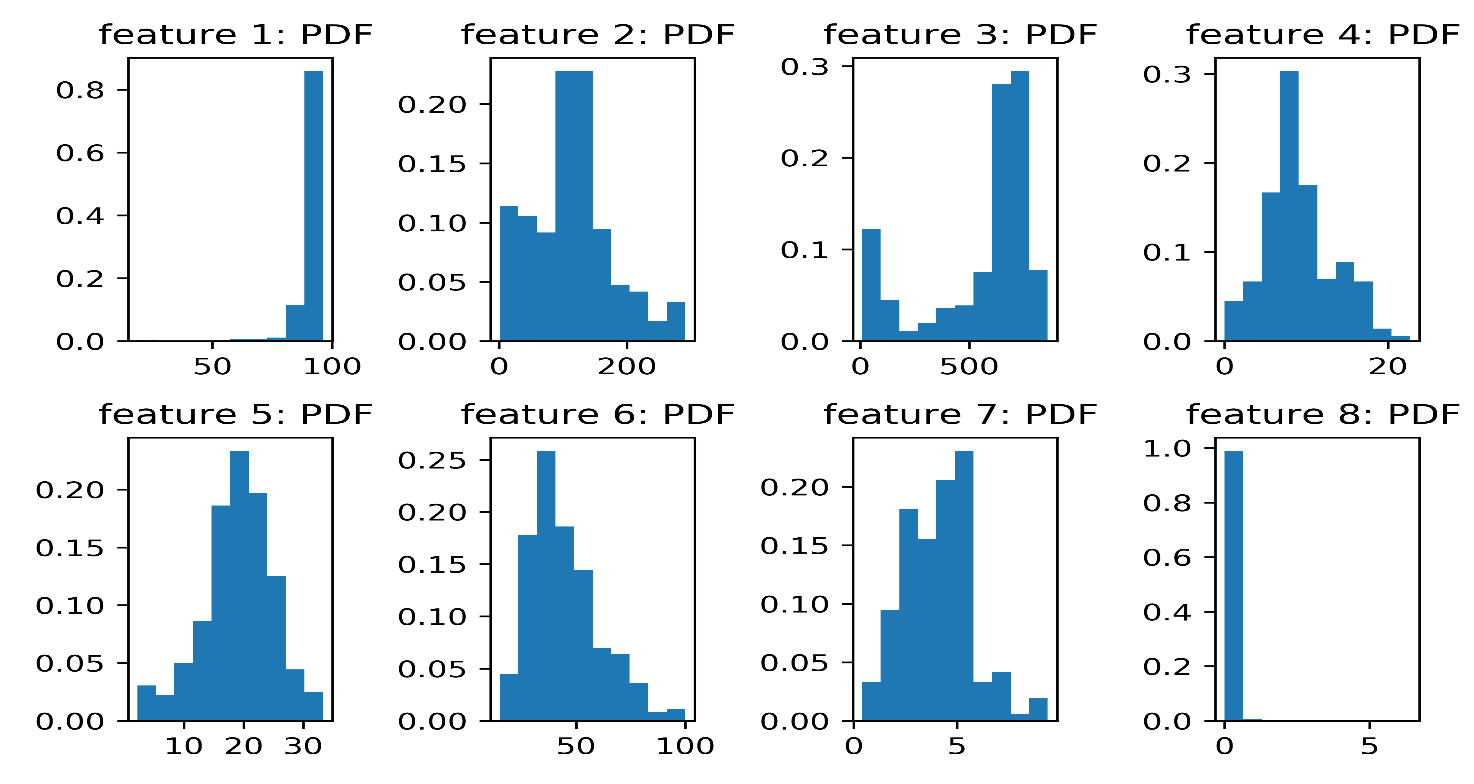
Never the less, they still produce the similar AE, the absolute in the L2 norm space of distance.

Reason is that the log + int will cut out all the float part after the integer while log + round will make such logarithmic interval to become more strictly-classified.

e.g. log 90 = 1.95, w/o wound it becomes 1, but w round it will be converted to 2, the classification process will consequently drop.

However, the absolute error does not count in the discrete level, it counts as the continuous level, thus both of the model will produce the similar AE.

●



As we can see, the PDF of forestfire dataset is quite unevenly distributed.

Hence for the GaissianNB which takes the presumption that the dataset is in Gaussian distribution (only feature 1 4 5 6 are close to Gaussian distribution) failed to be accurate.

Due to the unevenly distributed dataset, the KDTree, Decision Tree Classifier ,NN Regressor and

DT Regressor all generate the inaccurate result.

BernoulliNB surprisingly result in an outstanding accuracy, the reason I guess is that dataset in forestfire although unevenly distributed, but Bernoulli only check (not count) the occurrence or not(namely happened or not, binary-like processing ) in the dataset.

MultinomialNB counts the times of occurrence in the multinomial distribution, compared to the BernoulliNB, it maybe classify the dataset too strictly (sometimes the strict classification can prevent the overfitting problem while this time we did not see such result.)

Overall, the difference b/w BernoulliNB and MultinomialNB mainly lies in “how they count the dataset.” The former only counts whether the event happens or not while the latter counts the OCCURANCE of such event.

## Conclusion

In short, suppose we only care whether an event happened or not, we use BerboulliNB, taking a step further we may use the times of occurrence of an event using for analyzing, classification we use MultinomialNB, last but not least, for the dataset which resembles normal distribution, we use GaussianNB.

For the discrete prediction, using the classifier, and using regressor for continuous prediction. (Continuous dataset may use some mapping method to transform into discrete one such as log and round)