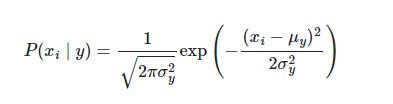
1. **Gaussian Naïve Bayes**

[**GaussianNB**](https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html#sklearn.naive_bayes.GaussianNB) implements the Gaussian Naive Bayes algorithm for classification. The likelihood of the features is assumed to be Gaussian:

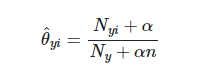
The parameters σy and μy are estimated using maximum likelihood.



1. **Multinomial Naïve Bayes**

[**MultinomialNB**](https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html#sklearn.naive_bayes.MultinomialNB) implements the naive Bayes algorithm for multinomially distributed data, and is one of the two classic naive Bayes variants used in text classification (where the data are typically represented as word vector counts, although tf-idf vectors are also known to work well in practice). The distribution is parametrized by vectors θy=(θy1,…,θyn) for each class y, where n is the number of features (in text classification, the size of the vocabulary) and θyi is the probability P(xi∣y) of feature i appearing in a sample belonging to class y.

The parameters θy is estimated by a smoothed version of maximum likelihood, i.e. relative frequency counting.



where Nyi=∑x∈Txi is the number of times feature i appears in a sample of class y in the training set T, and Ny=∑i=1nNyi is the total count of all features for class y.

The smoothing priors α≥0 accounts for features not present in the learning samples and prevents zero probabilities in further computations. Setting α=1 is called Laplace smoothing, while α<1 is called Lidstone smoothing.

1. **Bernoulli Naïve Bayes**

[**BernoulliNB**](https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.BernoulliNB.html#sklearn.naive_bayes.BernoulliNB) implements the naive Bayes for data that is distributed according to multivariate Bernoulli distributions; i.e., there may be multiple features but each one is assumed to be a binary-valued (Bernoulli, boolean) variable. Therefore, this class requires samples to be represented as binary-valued feature vectors; if handed any other kind of data, a BernoulliNB instance may binarize its input (depending on the binarize parameter).

The decision rule for Bernoulli naive Bayes is based on



which differs from multinomial NB’s rule in that it explicitly penalizes the non-occurrence of a feature i that is an indicator for class y, where the multinomial variant would simply ignore a non-occurring feature.

In the case of text classification, word occurrence vectors (rather than word count vectors) may be used to train and use this classifier. BernoulliNB might perform better on some datasets, especially those with shorter documents. It is advisable to evaluate both models, if time permits.

**Comment on 1a**

For the same Iris data set, both Gaussian and Multinomial Naïve Bayes gives same output ie; [Virginicia], but for Bernoulli Naïve Bayes the output is [sentosa].

Bernoulli NB uses multivariate Bernoulli distributions; i.e., there may be multiple features but each one is assumed to be a binary-valued (Bernoulli, boolean) variable. Therefore, this class requires samples to be represented as binary-valued feature vectors. Here Bernoulli NB binarize its input (depending on the binarize parameter).