Report on Naive Bayes Classifiers

Introduction:

Before I get into the different types of models I want to describe what a naive bayes model is, **Naive Bayes** is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of features, where the class labels are drawn from some finite set. There is not a single algos for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a particular feature is independent of the value of any other feature, given the class variable

$$p(C_k \mid \mathbf{x}) = \frac{p(C_k)\, p(\mathbf{x} \mid C_k)}{p(\mathbf{x})}$$

Data Pre-Processing:

We can't use the data as it is, as it's not really computable at this point. So I made some changes to the representation of the data. Taking this small example of **3 messages**.

	Label	SMS
0	spam	SECRET PRIZE! CLAIM SECRET PRIZE NOW!!
1	ham	Coming to my secret party?
2	spam	Winner! Claim secret prize now!

We need to **remove the punctuations** and then **split it by the spaces** then finally we can have our **word base**(Dictionary). And finally the data looks like the following.

	Label	secret	prize	claim	now	coming	to	my	party	winner
0	spam	2	2	1	1	0	0	0	0	0
1	ham	1	0	0	0	1	1	1	1	0
2	spam	1	1	1	1	0	0	0	0	1

Now for the real dataset we will have a data set that looks like the following.

10]:

	label	Text	simulate	upd8	matthew	clearer	arsenal	age23	scotsman	fatty	...	fa	help08714742804	ammae	scared	swimsuit	expensive	ma
0	0	[yep, by, the, pretty, sculpture]	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
1	0	[yes, princess, are, you, going, to, make, me,...	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
2	0	[welp, apparently, he, retired]	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
3	0	[havent]	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0
4	0	[i, forgot, 2, ask, ü, all, smth, there, s, a,...	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0

5 rows × 7785 columns

As you can see the number of **features is 7785.** These are all the unique words encountered in the training set. And the numbers below them are the frequency of each word in that particular message.

We will need to convert the test set to a similar form to make the prediction based on these classifiers.

Different Types of Naive Bayes:

1) **Multinomial :** It uses term frequency i.e. the number of times a given term appears in a document. Term frequency is often normalized by dividing the raw term frequency by the document length. After normalization, term frequency can be used to compute maximum likelihood estimates based on the training data to estimate the conditional probability.

$$p(\mathbf{x} \mid C_k) = \frac{(\sum_i x_i)!}{\prod_i x_i!} \prod_i {p_{ki}}^{x_i}$$

So for our dataset we just calculate the constants like p_spam, p_ham, etc. Then to classify the new messages we just multiply the probability for that word with all the words in the sentence, if a word is absent from the word base then we just ignore the word

2) **Bernoulli :** In this event model, features are independent Booleans (binary variables) describing inputs. Like the multinomial model, this model is popular for document classification tasks, where binary term occurrence features are used rather than term frequencies.

$$p(\mathbf{x} \mid C_k) = \prod_{i=1}^{n} p_{ki}^{x_i} (1 - p_{ki})^{(1-x_i)}$$

Similar to the multinomial NB the dataset will contain the same form with a minor but crucial change, all the values need to be either 0 or 1. So the numbers will become 1 if the word exists in the message or 0 if it's absent. The further steps are similar to the above classifier.

3) **Gaussian :** When working with continuous data, an assumption often taken is that the continuous values associated with each class are distributed according to a normal (or Gaussian) distribution. The likelihood of the features is assumed to be-

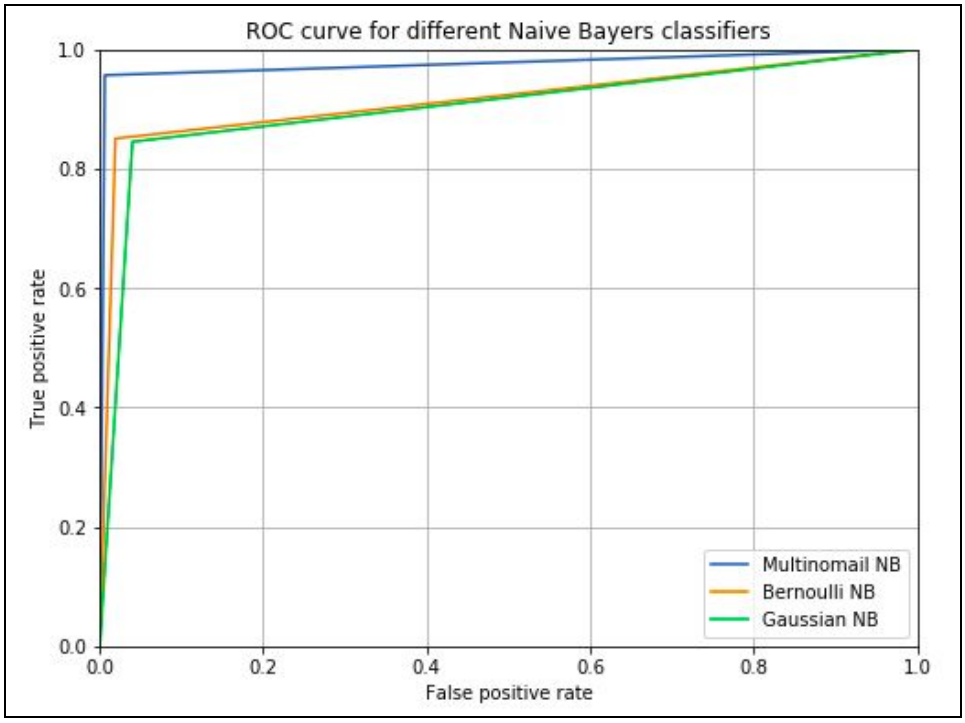
$$p(x = v \mid C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{-\frac{(v-\mu_k)^2}{2\sigma_k^2}}$$

Results:

The **Accuracy** of the models are as follows :

Multinomial Naive Bayers	98.115 %
Bernoulli Naive Bayers	91.472 %
Gaussian Naive Bayers	90.365 %

The **ROC Curve** of the Classifiers is like so :



Conclusion:

From the results it's clear that the **Multinomial Naive Bayes Classifier** works best for this dataset. This is understandable as the dataset is not very uniform and definitely not distributed normally so even attempting Gaussian NB is not wise. Now Bernoulli NB had a good chance as it does the same thing as MNB but it loses learning accuracy due to not testing for word frequency.

In short the more information you provide your model the better it will perform, and if you can't analyse the pattern or flow in the data then its better to apply all the methods and see what works best.