Machine Learning: Logistic Regression vs. Naive Bayes

1st Yan Liu 10443939

2st ZhenRong Zhang 10544922

The course work of
Foundations of Machine Learning

Computer Scinece
University of Manchester
2019.10.31

Logistic Regression vs. Naive Bayes

* Foundations of Machine Learning course work

1st ZhenRong Zhang
10544922
university of manchester
Computer Scinece
zhenrong.zhang@postgrad.manchester.ac.uk

2nd Yan Liu
10443939
university of manchester
Computer Scinece
yan.liu-29@postgrad.manchester.ac.uk

***Abstract*—Logistic Regression and Naive Bayes are two popular models used to solve many problems of machine learning, in many ways the two algorithms are similar, but at the same time very different. This report will perform some experimental comparison of Logistic Regression versus Naive Bayes, and will indicate their generalization performance differs when supplied with different size of training dataset.**

***Index Terms*—Machine Learning, Logistic Regression, Naive Bayes**

I. Introduction

The task of supervised learning is training a model and predicting the output with the given input, therefore how to determine a suitable model is significant. Generally, there are two types of models called generative models ($Y = f(X)$) and discriminative models ($P(Y|X)$) as typified by Naive Bayes and Logistic Regression.

Generative Model is a model which training data to learn joint probability distribution $P(X, Y)$ and calculate the conditional probability distribution $P(Y|X)$ and then use the result as the model of prediction: $P(Y|X) = \frac{P(X,Y)}{P(X)}$. It is called the Generative Model because the model represents the generative relation between input X and output Y. One of the typical generative models is the probability model such as Naive Bayes.

While Discriminative Model is a model that uses the decision function $f(X)$ or the conditional probability distribution $P(Y|X)$ to do the predict. This kind of model only concerns which prediction of Y it should give with the given input X. Being different from the generative modeling, there is no need for discriminative modeling to learn the joint probability distribution $P(X, Y)$ in advance. The typical discriminative model is Logistic Regression.

In fact, during the process of model building, people prefer a discriminative model rather than a generative model. Vapnik [1] pointed out that one should solve the classification problem directly and never solve a more general problem as an intermediate step. However, compared with the discriminative model such as logistic regression, whether generative models such as Naive Bayes do worse in classification assignments is remaining uncertain.

In this paper, we use both continuous data and discrete data to find a comparison between the generative model (Naive Bayes) and the discriminative model (Logistic Regression). Besides, during the investigation, we will increase the number of examples to compare the performance of the two models.

We find that in a continues dataset, the logistic regression may have higher accuracy (or lower asymptotic error), but the speed for Naive Bayes to converge is faster[2]. While in the discrete dataset with independent features, the accuracy of the logistic regression model may not change a lot, but Naive Bayes may run faster (which means that the speed of Naive Bayes to approach the asymptotic is much faster even it may have higher asymptotic error).

As Andrew Y.Ng and Michael I. Jordan[3] conclude that the generative model does indeed have a higher asymptotic model, but the generative model may also approach its asymptotic error much faster than the discriminative model—possibly with many parameters. This means that when the training size reaches infinity the discriminative model (logistic regression) performs better than the generative model (Naive Bayes). However the generative model reaches its asymptotic faster ($O(logn)$) than the discriminative model($O(n)$), i.e the generative model (Naive Bayes) reaches the asymptotic solution for fewer training sets than the discriminative model (Logistic Regression).

II. Mathematic

In this section, we do some formula derivation for Naive Bayes and Gaussian Bayes to see the expression and find some relationship between Logistic Regression and Naive Bayes.

A. discrete data

If the dataset is discrete, for the Naive Bayes which based Bayesian theory, the formula is $P(Y|X) = \frac{P(X,Y)}{P(X)}$. For $x = (x_1, \cdots, x_n)$ (the field of x belong R^D which is the D-dimensional input space). To predict y for the testing dataset x, it can be calculated as $Y_k = argmax(P(Y_k|X))$ and the field of Y_k is Y. which equals to calculate $(P(Y_k|X))$. The Naïve Bayes suppose that all the attribute are which means that the joint probability of all the attribute $P(X_1, \cdots, X_n) = P(X_1)P(X_2)\cdots P(X_n)$, so we can calculate the $P(X|Y_k)$ as following formula (1):

$$P(X|Y_k) = \sum_{d=1}^{D} P(X_d|Y_k) \tag{1}$$

To predict the label, we can see the number of the $P(X|Y_k)$ and compare each of them to see which is larger and to judge the label. However, if there are too many attributes, it may be very difficult to compare each of them. From that, we can do a sample calculation like formula(2) to get the label:

$$\log \frac{\left(\prod_{i=1}^{d} P(X|Y_i)P(Y_i)\right)}{\left(\prod_{j=1}^{d} P(X|Y_j)P(Y_j)\right)} \tag{2}$$

For every i and j in the D-dimension, we do that calculation and if the formula bigger than 0, we can judge that the label is i and we use a loop to do all the judge to compare the i and j in the D-dimension to judge the label Y.

B. Continuous data

If the attribute is continuous, there are two ways to deal with discrete. The first way is to make every continues attribute become discrete then use the discrete interval to take place of continues attributes, however, this way cannot control the size of the dividing of the discrete interval. If the interval is too small, it will not make a reliable estimate to the $P(X|Y)$ because the number of the training recording is too small and if there are too many records, those intervals will have different records from different classes and it will lose the correct decision boundary[4].

The second way is that we can suppose that continues attributes obeys a certain probability distributionand use the training set to estimate those parameters of those distributions. We suppose that the attribute following the Gaussian distribution that the mean is mu and the std is sigma the result can be calculated as the formula(3):

$$P\left(X_d\,|Y_k\right) = G\left(X_d, \mu_{y_k}, \sigma_{y_k}\right) = \left(\frac{1}{\sqrt{2\pi}\sigma}\right) e^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{3}$$

To calculate the expression of Guassian Naive Bayes, suppose Y is a binary labellet's take $Y = 1$ as example, then $P(Y = 1|X)$=

$$\frac{P(Y=1)P(X|Y=1)}{P(Y=1)P(X|Y=1) + P(Y=0)P(X|Y=0)} \tag{4}$$

Then using the law of total probability to transform $P(X)$ into the form of factorization then make the element and denominator in the right side in the formula above dividing the element and use the exp and ln matching that we can get:

$$P(Y=1|X) = \frac{1}{1 + exp(\ln \frac{P(Y=0)P(X|Y=0)}{P(Y=1)P(X|Y=1)})} \tag{5}$$

then we can transform the formula(5) because all of the features meet the requirement of independence:

$$P(Y=1|X) = \frac{1}{1 + exp(\ln \frac{1-\pi}{\pi} + \sum_{i=1}^{n} \ln \frac{P(X_i|Y=0)}{P(X_i|Y=1)})} \tag{6}$$

Formula(6) may look like the Logistic Regression, however there also some difference that the summarize in the denominator is the form of probability instead of the form of a linear combination of X_i So next we use the normal distribution to transform it into the linear combination:

$$\sum_{i=1}^{n} \ln \frac{P(X_i|Y=0)}{P(X_i|Y=1)} = \sum_{i=1}^{n} (\frac{\mu_{i0} - \mu_{i1}}{\alpha_i^2} + \frac{\mu_{i1}^2 - \mu_{i0}^2}{2\alpha_i^2}) \tag{7}$$

Then we put formula(7) into the $P(Y = 1|X)$ to get the result shown as formula(8)(9):

$$P(Y=1|X) = \frac{1}{1 + exp(\omega_0 + \sum_{i=1}^{n} \omega_i X_i)} \tag{8}$$

$$P(Y=0|X) = 1 - \frac{1}{1 + exp(\omega_0 + \sum_{i=1}^{n} \omega_i X_i)} \tag{9}$$

and for ω_0 and ω_1 we have:

$$\omega_0 = \ln \frac{1-\pi}{\pi} + \sum_{i=1}^{n} \frac{\mu_{i1}^2 - \mu_{i0}^2}{2\alpha_i^2} \tag{10}$$

$$\omega_1 = \frac{\mu_{i0} - \mu_{i1}}{\alpha_i^2} \tag{11}$$

we can see that this is the same with the Logistic Regression. However, the ω in different models is different. In Logistic Regression, it tries to minimize the cost function to get the best ω, however in Guassian Naive Bayes, the form of ω is given. It is decided by the mean and variance from normal distribution and what training set do is to estimate those mean and std instead of calculation w which result in the difference between Discriminative Modeling and Generative Modeling.

III. Experiment

As we know, For those data sets which are not linearly separable, none of those two algorithms can do well, we may prefer to use the SVM but here we will not discuss it. We will choose the continues data or the discrete data which are linear separable so that we can compare the performance between Naive Bayes and the Logistic Regression. In experiments, different test sets were chosen.

A. Comparison of Logistic Regression and Gaussian Nave Bayes classifiers

First of all, we choose three datasets from scikit learn called make_moons, make_circles and make_classification [5]. Then do a comparison of Logistic Regression and Gaussian Nave Bayes classifiers on these synthetic datasets.

We split the dataset into a training subset (train_size=0.6) and a testing subset (test_size=0.4)by using train_test_split method. Then plot the input data and the decision boundary by using a scatter diagram. The plots show training points in solid colors and testing points semi-transparent. The lower right shows the classification accuracy on the test set. From the fig-1, we can see that the Gaussian Naive Bayes seems to have a better classification accuracy in the second dataset, then look the input data figure, we find that the sample points are not so separable (red points and blue points are mixed).

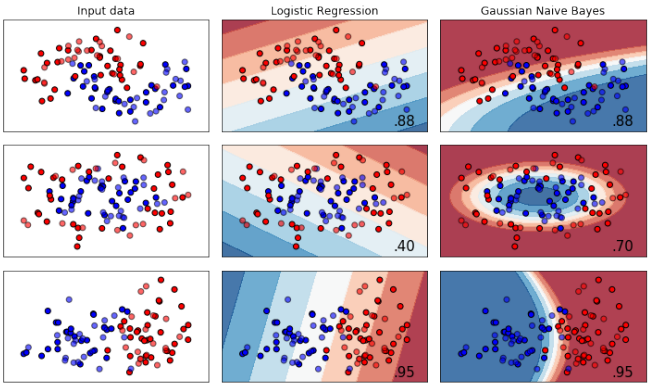

Fig. 1. Classifier comparison

B. Detailed Classifier and Performance comparison

To demonstrate this finding, we conducted another similar but more detailed experiment. We choose HandwrittenDigits-DataSet from the UCI machine learning repository. This time, we split the data sets evenly. We print the Classification report for classifier and have a look at the first 20 examples in the testing subset to analyze the advantages and disadvantages of the two models (which shown below).

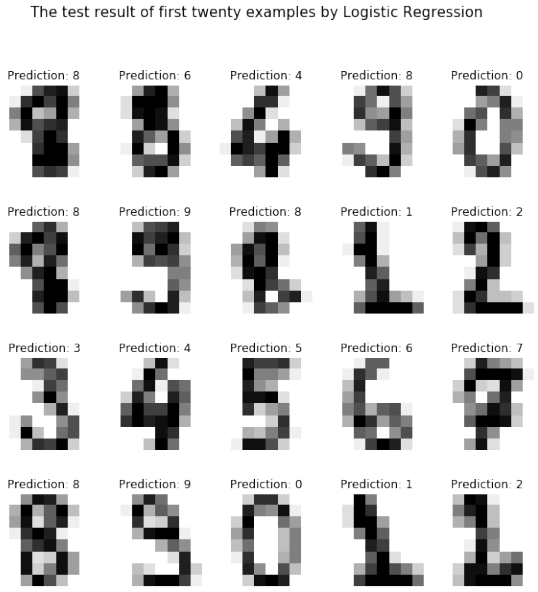

Fig. 2. Predict result of Logistic Regression on digits dataset

From the fig-2 and fig-3, we can see that the Logistic Regression predict digits more accurate (although there are still errors). To check the accuracy of every label(0,1,...,9), we can look at fig-4 and fig-5, which show the precision and score of every lablel. It also illustrates that Logistic Regression does a better job because the accuracy of Logistic Regression is 0.92 while the accuracy of the Gaussian Naive Bayes is 0.81.

Then, we do another experiment to check whether the size of the dataset will affects the generalization performance.

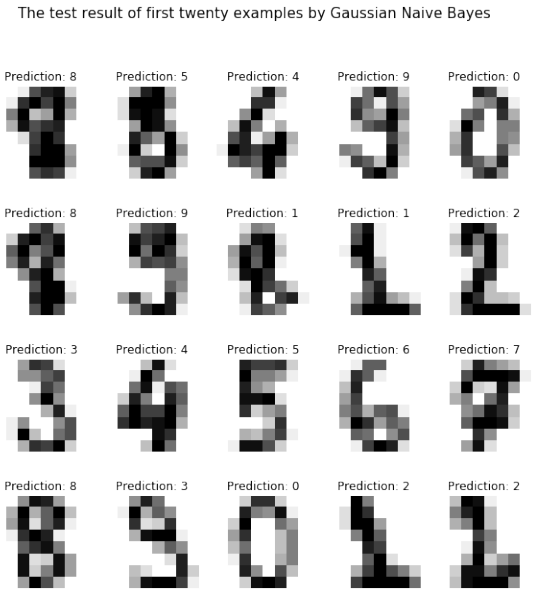

Fig. 3. Predict result of Gaussian Nave Bayes on digits dataset

Classification report for classifier LogisticRegression(C=1.0, intercept_scaling=1, l1_ratio=None, max_ite multi_class='warn', n_jobs=None, penalty='l random_state=None, solver='warn', tol=0.000 warm_start=False):

	precision	recall	f1-score	support
0	0.94	0.97	0.96	88
1	0.86	0.89	0.88	91
2	0.98	0.98	0.98	86
3	0.99	0.82	0.90	91
4	0.99	0.93	0.96	92
5	0.83	0.90	0.86	91
6	0.94	0.99	0.96	91
7	0.98	0.89	0.93	89
8	0.89	0.88	0.88	88
9	0.83	0.92	0.87	92
accuracy			0.92	899
macro avg	0.92	0.92	0.92	899
weighted avg	0.92	0.92	0.92	899

Fig. 4. Logistic Regression Classification report

Classification report for classifier GaussianNB(priors=None,

	precision	recall	f1-score	support
0	0.98	0.95	0.97	88
1	0.81	0.74	0.77	91
2	0.87	0.84	0.85	86
3	0.88	0.79	0.83	91
4	1.00	0.73	0.84	92
5	0.70	0.81	0.76	91
6	0.96	0.99	0.97	91
7	0.65	0.81	0.72	89
8	0.61	0.76	0.68	88
9	0.77	0.66	0.71	92
accuracy			0.81	899
macro avg	0.82	0.81	0.81	899
weighted avg	0.82	0.81	0.81	899

Fig. 5. Gaussian Nave Bayes Classification report

C. Comparison of Logistic Regression and Nave Bayes on different size of dataset

We realized a Logistic Regression by ourselves. There are several functions defined (in code file) by us, including: sigmoid, Cost_function, derivate_Of_cost_function, Gradient_descent. We set the epsilon to judge whether the gradient descent convergence.

But when we compared the accuracy between sklearn Logistic Regression model and our Logistic Regression model. We find that when the dataset is small, they may have the same accuracy, however, with the increasing of the size of the data set, the accuracy for the Logistic Regression in sklearn is higher than the Logistic Regression model that we realized. Because we did not set the punishment for the w which may cause the overfitting.

Therefore, we abandoned our model for the sake of accuracy. we choose the Logistic Regression model in sklearn library to do the following experiments.

We choose the iris data set and set the best c for the Logistic Regression model to see the different time costs and accuracy between the Logistic Regression model and the Naive Bayes model. The test result shown below. We can see from fig-6

```
the fit time is:  0:00:00.011628
the accuracy for the logistic regression: 0.9866666666666667
the fit time is: 0:00:00.000798
the accuracy for naive bayes:  0.96
```

Fig. 6. time cost and accuracy of LR and Naive Bayes for iris data

that the time for the Logistic Regression is about 0.01 and the accuracy is 0.98 while for the Naive Bayes the time is 0.00053 and the accuracy is 0.96. Therefore, it is clear that Logistic Regression costs more time than Naive Bayes, however the accuracy of the Logistic Regression is better.

Next, in order to compare Naive Bayes and Logistic Regression between discrete data with few examples, we create a dataset which shows what kind of man will be welcomed by women, in which x is the set for several attributes and Y is the binary data contains labels 0 or 1. The training subset is shown in table1 and the testing subset is shown in table2.

TABLE I
CREATED TRAINNING DATASET

Handsome	Character	Height	Hard-Working	WetherWelcome
YES	NO	NO	NO	NO
NO	YES	NO	YES	NO
YES	YES	NO	YES	YES
NO	YES	YES	YES	YES
YES	NO	NO	YES	NO
NO	NO	NO	NO	NO
YES	YES	YES	NO	YES
NO	YES	YES	YES	YES
YES	YES	YES	YES	YES
NO	NO	YES	YES	YES
YES	YES	NO	NO	NO
YES	YES	NO	NO	NO

From the result below (fig-7 and fig-8), we can see that the accuracy is almost the same (the data set is only about

TABLE II
CREATED TESTING DATASET

Handsome	Character	Height	Hard-Working	WetherWelcome
YES	NO	NO	NO	NO
NO	YES	NO	YES	NO
YES	NO	NO	YES	NO
YES	YES	YES	YES	YES
NO	YES	YES	YES	YES
NO	NO	NO	NO	YES
NO	YES	YES	YES	YES

10 examples) but the speed for the logistic regression is slower than the Gaussian Naive Bayes. And for those data set with few examples, Naive Bayes can do better than Logistic Regression:

```
the time to fit gaussiannb for discerete: 0:00:00.000584
the score for gaussiannb:
0.8571428571428571
```

Fig. 7. time cost and accuracy of Gaussian Naive Bayes for created dataset

```
the tiem to fit logistic for discerete: 0:00:01.437402
the score for logistic regression:
0.8571428571428571
```

Fig. 8. time cost and accuracy of Logistic Regression for created dataset

For the final test we choose the dataset named California housing which contains about more than 20 thousand examples, and we split the dataset from 10% to 90% as the training examples, For Logistic Regression we trained 10 model and for Naive Bayes we also trained 10 examples, and plot the relationship between the number of training examples and the accuracy and we can compare them between Logistic Regression and Naive Bayes. The plot is shown below:

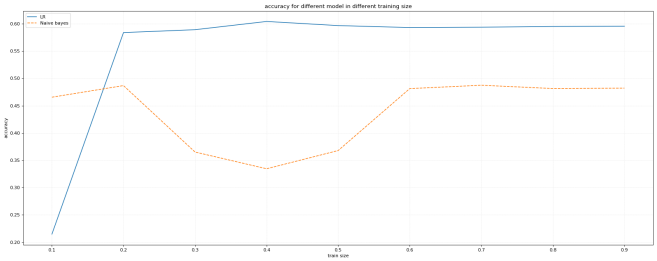

Fig. 9. time cost and accuracy of Logistic Regression

We can see that, in the beginning, the Naive Bayes perform better when the training set is small because the time for Naive Bayes to approach the asymptotic error is small so that Naive Bayes can do better, however with the increase of the training set, Logistic Regression does better as we supposed before.

Above all, we can see that Logistic Regression does better than Naïve Bayes when the dataset is big enough but the time cost is far more than Naive Bayes and when the training set is small, we may prefer the Naive Bayes.

IV. Analysis

A. Naive Bayes algorithm analysis

For Naive Bayes algorithm, we assume a sample set:

$$\begin{aligned} D = \{ & (X_1^{(1)}, X_2^{(1)}, \cdots, X_n^{(1)}, y_1), \\ & (X_1^{(2)}, X_2^{(2)}, \cdots, X_n^{(2)}, y_2), \\ & \cdots (X_1^{(m)}, X_2^{(m)}, \cdots, X_n^{(m)}, y_n) \ \} \end{aligned} \tag{12}$$

These $y_i (i = 1, 2, \cdots, m)$ means the category of the example and the value is $\{C1, C2, \cdots\}$.

- Calculate the prior probability: Set the number of example equals K. For every sample, we set $Y = C_k$, and Calculate $P(Y = C_k)$. It is the sequence of C_k in the total examples
- Calculate the conditional probability: Divide example set into K subset and do the calculation for the subset which belongs to C_k to calculate the $possibilityOfFeature = \frac{a_{jl}}{P(X_j = a_{jl}|Y = C_k)}$. It is the rate between the number of examples whose feature is ajl and the number of the subset.
- Calculate the posterior probability for the C_k with the test example X^test: let $G = P(Y = c_k|X = x^{test})$

$$G = P(Y = C_k) \prod_{j=1}^{n} P(X_j = x_j^{test}|Y = Ck) \tag{13}$$

And it will be classified as the class which have the highest possibility we can see that the time complexity for Naïve Bayes is about O(logn)

B. Logistic Regression algorithm analysis

For Logistic Regression algorithm, we also set the sample set:

$$\begin{aligned} D = \{ & (X_1^{(1)}, X_2^{(1)}, \cdots, X_n^{(1)}, y_1), \\ & (X_1^{(2)}, X_2^{(2)}, \cdots, X_n^{(2)}, y_2), \\ & \cdots (X_1^{(m)}, X_2^{(m)}, \cdots, X_n^{(m)}, y_n) \ \} \end{aligned} \tag{14}$$

It is a good idea to take binary to classify as an example, in which, every sample belongs to 0 or 1. As it is mentioned before, differing from Naive Bayes, LR makes models to calculate $P(Y|X)$ directly instead of calculating $P(X|Y)$ and $P(Y)$ then use Bayes theory. When trying to calculate $P(Y|X)$, Logistic Regression set the Function:

$$P(Y = 1|X) = \frac{1}{1 + e^{-(e^t X)}} \tag{15}$$

$$P(Y = 0|X) = 1 - \frac{1}{1 + e^{-(e^t X)}} \tag{16}$$

What Logistic Regression does is try to train dataset to find the optimal theta to make $P(Y = 1|X)$ as big as possible when the real label of sample is 1 and make $P(Y = 0|X)$ as small as possible when the label of sample is 0, which equals that what do logistic regression is trying to minimize the cost function J which can be expressed like:

$$J(\theta) = \frac{-1}{m} \left[\sum_{i=1}^{m} \log h_\theta(x^i) + (1 - y^i) \log(1 - h_\theta(x^i)) \right] \tag{17}$$

where $h(X) = g(w^t X)$ and g is sigmoid function,then do the gradient descent:

repeat

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \tag{18}$$

while the calculation of derivative is :

$$\frac{\partial}{\partial \theta_j} J(\theta) = \sum_{i=1}^{m} (h\theta(x^i) - y^i) x_j^i \tag{19}$$

and during the process all the parameters should be updated at the same time the time complexity for the gradient descent is $O(n * C * I)$, n represent the number of the example, C represents the calculation for the single sample and I is the time of iteration. To conclude, the time complexity can be shown as O(n). From that we can see, $n < \log n$ and suppose T1 and T2 be the time that Naïve Bayes and LR that convergence and T1 will indeed lower than T2.

When both two models become convergence, the error rate of LR is lower than the error rate of Naive Bayes. Which means, with the change of numbers of training examples, he different between Logistic Regression and Naive Bayes will be explored, When the training set is very small, the accuracy of Naive Bayes will be higher than Logistic Regression because the speed of Naive Bayes to approach the asymptotic error is faster, however with the increasing of the training set, the error rate of Logistic Regression is lower so it does better than the Naive Bayes.

When the feature meets the requirements of independence, with the increasing of training example, when both Gaussian Naive Bayes and Logistic Regression approach to the convergence, the parameters that the get will be the same, however, that suppose based on the basis that all the features are independence. In the real world, it is very hard for every feature to become independent so that the accuracy of Naive Bayes will be lower. While for the Logistic Regression, the maximum conditional likelihood solution of Logistic Regression can adjust the optimal parameters according to the non-independence in the data.

V. Conclusion

Overall, for Logistic Regression, it gets the answer by modeling $P(Y|X)$ directly and this model supposes $P(Y = 1|X)$ meet the logistic function $h(z) = \frac{1}{1+exp(w'X)}$, and sometimes there will be a punishment to prevent overfitting. While for Naive Bayes, it modeling the $P(X, Y)$ and use Naive theorem to solve $P(Y|X)$ and it supposes that all of the features meet the independence. Especially, Gaussian Naive Bayes supposes all the features meet the normal distribution.

When all of the features meet the requirement of independence, with the increasing of training examples, in the

limited situation, Naive Bayes can perform as well as Logistic Regression in terms of accuracy, however, it is very hard to do that because it is very hard for all of the features to become independent in the real world. When there are few training examples, Naive Bayes can perform well in the most situation because it approaches its asymptotic error faster. With the increasing of training examples, Logistic Regression will defeat Naive Bayes because the asymptotic error is lower than Naive Bayes.

Therefore, although Naive Bayes and Logistic regression are both linear classifiers, compared to logistic regression, Naive Bayes has a higher bias but lower variance. If the data set follows the bias then Naive Bayes will be a better classifier.

REFERENCES

[1] V. N. Vapnik, Statistical Learning Theory. John Wiley & Sons, 1998.
[2] Sanghamitra Deb, Naive Bayes vs Logistic Regression, 2016.
[3] On Discriminative vs. Generative classifiers: A comparison of logistic regression and naive Bayes. Andrew Y. Ng & Michael I. Jordan.
[4] Yi Zhang, Logistic regression, Gaussian naive Bayes, and their connections, Machine Learning, Spring, 2011.
[5] Gael Varoquaux & Andreas Müller,Classifier comparison, scikit-learn.