EE5907
Pattern Recognition

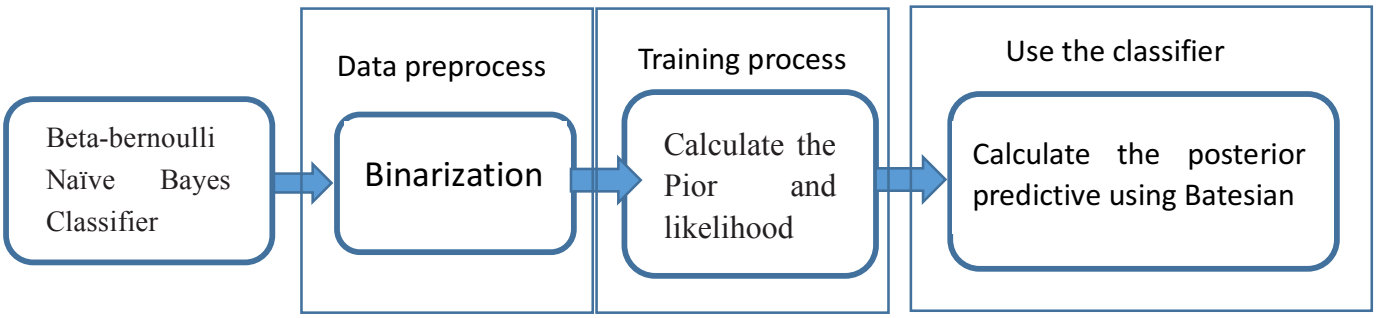
CA1 - Spam Email Filtering

Report

NAEM: Fei Yang
Student ID: A0169096N

Q1. Beta-bernoulli Naive Bayes

1. Process



(1) Data preprocess

I binarize the data features. 0 in Xtest and Xtrain will be set to 0, the rest of features whose values are not 0 will be set to 1.

(2) Calculate prior

The class label prior can be estimated using ML.

$$\pi_C = \frac{N_C}{N}$$

(3) Calculate likelihood

Calculating the posterior predictive distribution of each feature(set a = b):

$$p(x_j = 1|y = c, D) = \bar{\theta}_{jc} = \frac{N_{jc} + a}{N_c + a + b}$$

As this is naive bayas classifier, features are conditionally independent given the class label. We can get the likelihood of each email:

$$p(x|y = c, \theta) = \prod_{j=1}^{D} p(x_j|y = c, \theta_{jc})$$

(4) Calculate the posterior predictive

We use the bayes rules to get the result

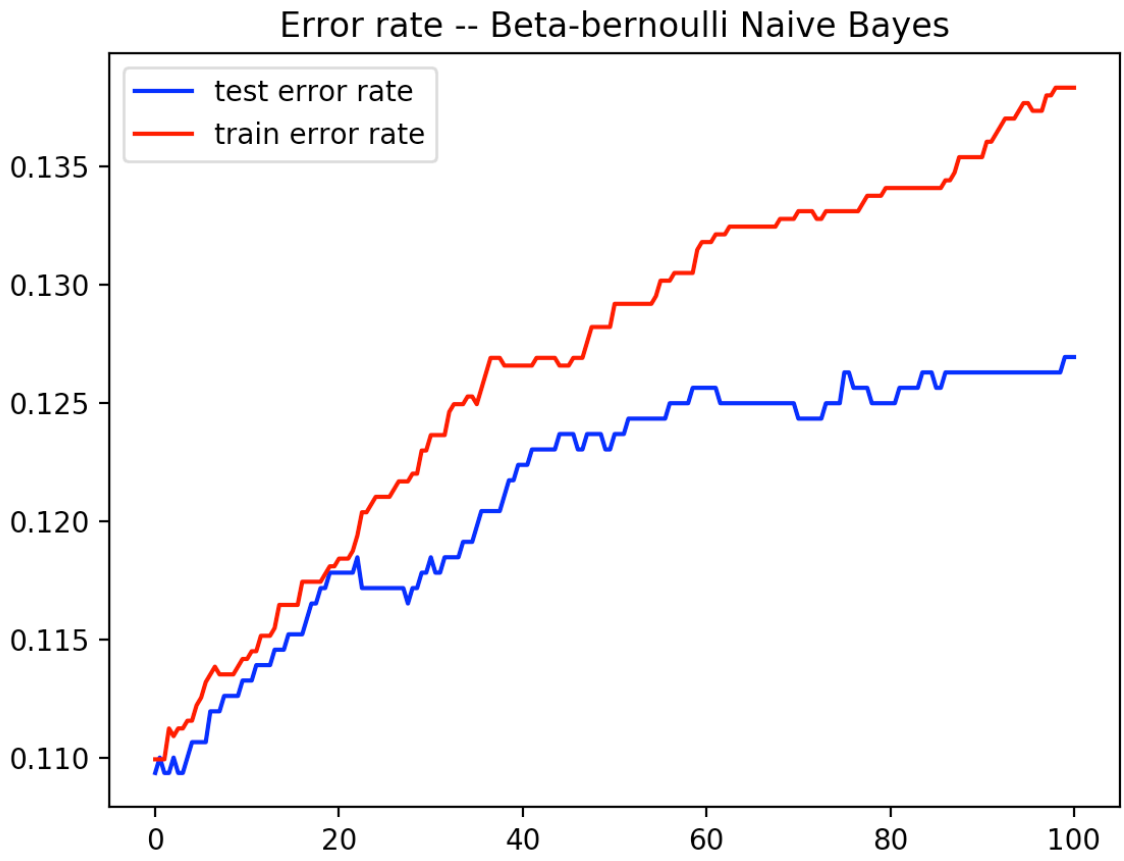
$$p(y = c|x, D) \propto \bar{\pi}_c \prod_{j=1}^{D} \bar{\theta}_{jc}^{\mathbb{I}(x_j=1)} (1 - \bar{\theta}_{jc})^{\mathbb{I}(x_j=0)}$$

(5) Test and calculate the error rate

We calculate the probability of each email being classified to class 1 and class 0 and compare them. If p(y=1|x,D)>p(y=0|x,D), we assume it is a spam email, otherwise non spam. After testing all the test data(Xtest and Xtrain), we compare them with ytest and ytrain and count the different results between them to calculate the error rate.

2. Result

(1) Plots of training and test error rates versus α



(2) What do you observe about the training and test errors as α change?

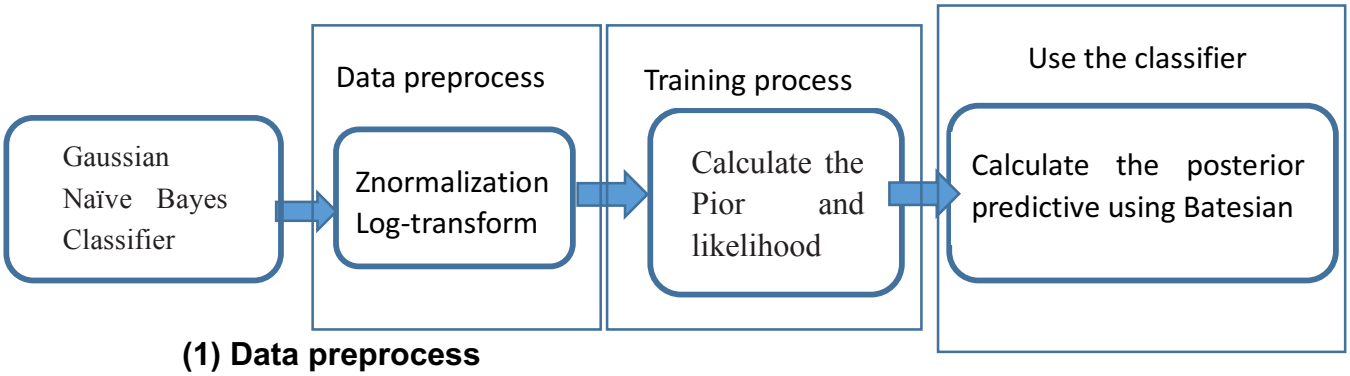
In general, both of the training and test error rate curves rise when α rises, with similar growing trend, though go down at some points. And the training error rate is always higher than the testing error rate.

(3) Training and testing error rates for α = 1, 10 and 100

a	1	10	100
Testing error rate	0.109375	0.11328125	0.126953125
Training error rate	0.109951060359	0.114192495922	0.138336052202

Q2. Gaussian Naive Bayes

1. Process



(1) Data preprocess

I use the Z-normalization and Log-transform to initialize all the data. In Z-normalization, I directly use the 'zscore' function to initialize the Xtrain data, and I use the definitional formula to initialize the Xtest based on the Xtrain. Then I just use log(x ij + 0.1) to initialize Xtrain and Xtest.

(2) Calculate prior:

The class label prior can be estimated using ML.

$$\pi_C = \frac{N_C}{N}$$

(3) Divide matrix:

I divide the Xtrain into two separate matrix according to class 1 or class 0, which contributes to get the mean and variance of different class.

(4) Calculate likelihood using the Gaussian distribution:

When given the mean and variance of class 1 or class 0, I can calculate the probability of each feature:

$$Pr(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-0.5(x-\mu)^2/\sigma^2\right]$$

(4) Calculate the posterior predictive

We use the bayes rules to get the result

$$\mathrm{p}(\mathrm{y}=\mathrm{c}|\mathrm{x},\theta) \propto \pi_c \prod_{j=1}^{D} \frac{1}{\sqrt{2\pi\sigma_{cj}^2}} \exp\left[\frac{-0.5\left(x_{ij}-\mu_{cj}^2\right)}{\sigma_{cj}^2}\right]$$

(5) Test and calculate the error rate

I calculate the probability of each email being classified to class 1 and class 0 and compare them. If p(y=1|x,D)>p(y=0|x,D), we assume it is a spam email, otherwise non spam. After testing all the test data(Xtest and Xtrain), we compare them with ytest and ytrain and count the different results between them to calculate the error rate.

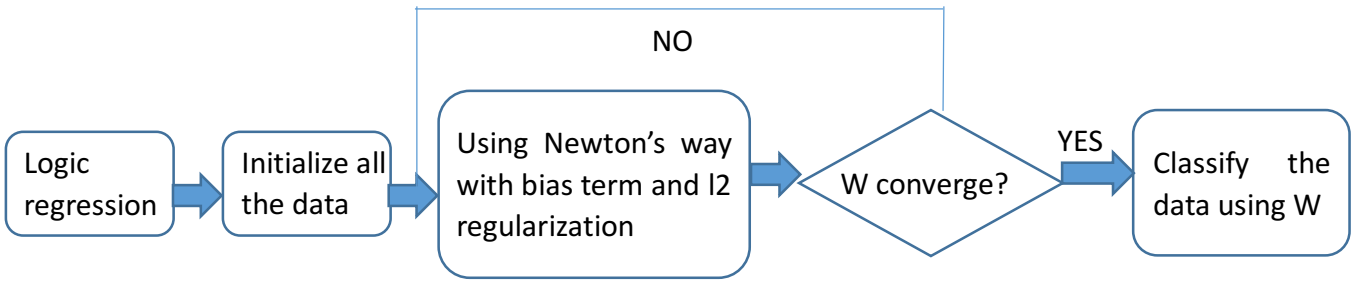
2. Result

Training and testing error rates for both z-normalized and log-transformed data

Data processing	Training error rate	Testing error rate
Z-normalization	0.17650897	0.201822916
Log-transform	0.16117455	0.18424479

Q3. Logistic regression

1. Process



(1) Data preprocess

I preprocess the data using three different ways, Z-normalization, log-transform and binarization, which is the same as that has been mentioned above. In addition, I plug in a column of 1 in the front of Xtrain and Xtest to match the bias term w0. Besides, I set the learning rate to 1.

(2) Training process

In order to prevent w from exploding to cause overfitting, I add a l2 regularization term into the original Newton's formulas. Therefore, our purpose is to calculate these formulas shown below:

$$NLL_{reg}(w) = NLL(w) + \frac{1}{2}\lambda w^T w$$
$$g_{reg}(w) = g(w) + \lambda w$$
$$H_{reg}(w) = H(w) + \lambda I$$

then, calculate w until converge

$$w_{k+1} = w_k - H_k^{-1} g_k$$

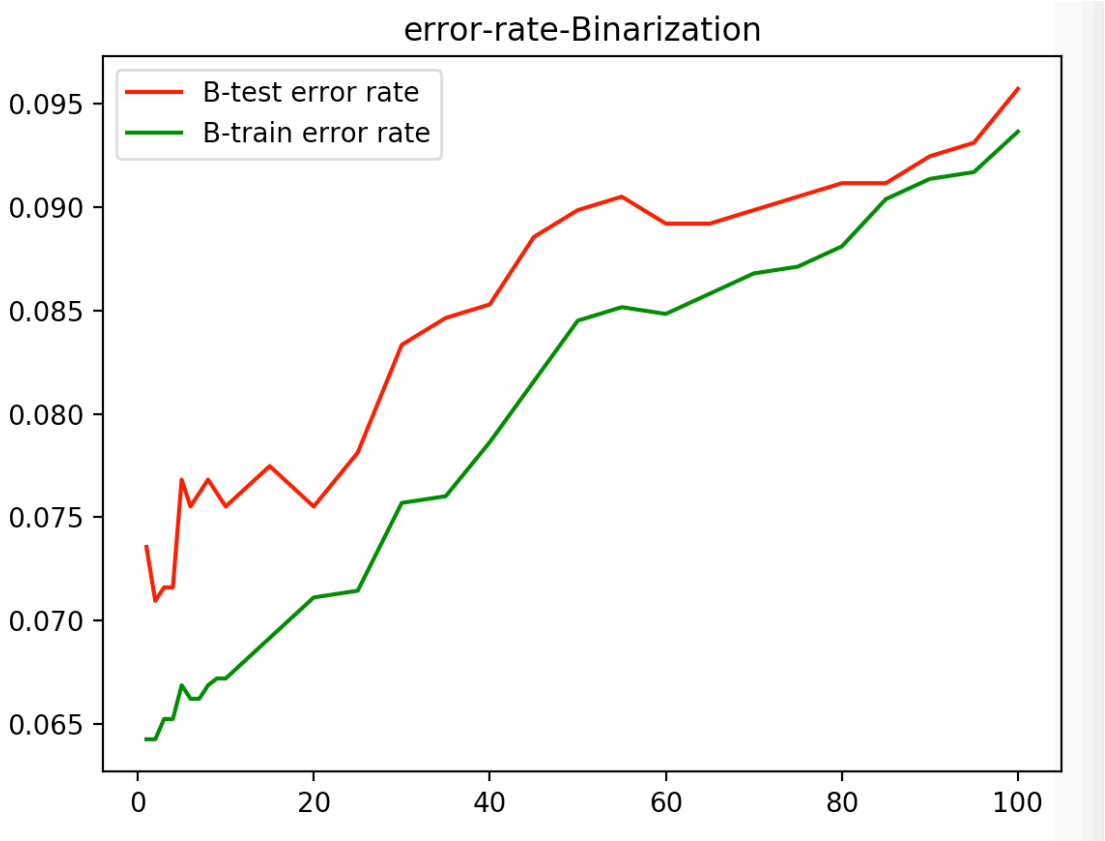
(3) Test and calculate the error rate

Use the new w to classify the data to class 1 or class 0 by calculating the log odds ($\omega^T x$), if log odds is bigger than 0, it is spam email. Then, calculate the error rate by comparing result with ytrain or ytest.

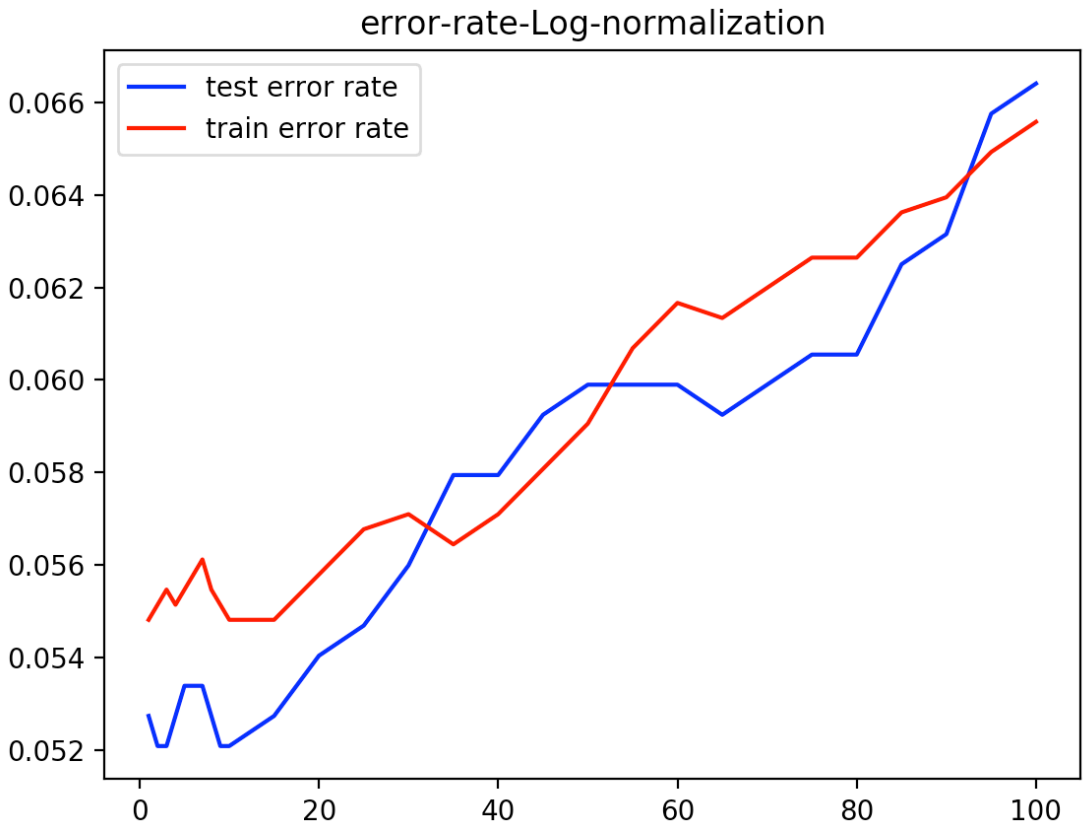
2. Result

(1) Plots of training and test error rates versus λ

Binarization:



Log-transform:



Z-normalization:

(2) What do you observe about the training and test errors as λ change?

In general, both of the training and test error rate curves rise when α rises, though go down at some points. When λ is between 1 and 10, the curves fluctuate more frequently. Under Log-transform, two curves cross with each other sometimes.

(3) What do you observe about the error rates of the diff erent preprocessing strategies?

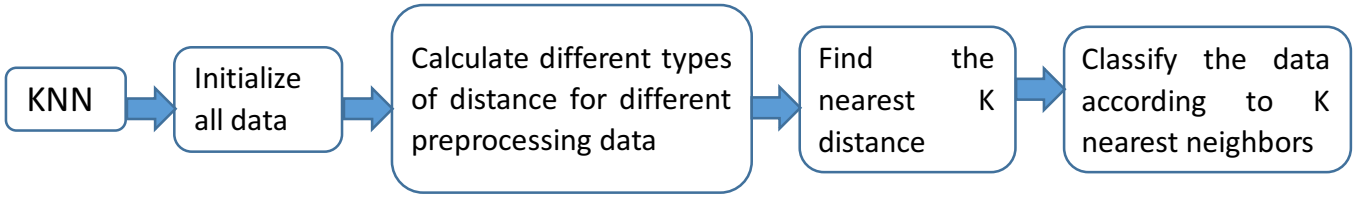
Log-transform preprocessing way will give the lowest error rate among these three ways, which is the most accurate one. Binarization performs better than Z-normalization when λ is small in some degree. But the curve of binarization grows more dramatically than the raise of Z-normalization.

(4) Training and testing error rates for λ = 1, 10 and 100.

λ	1	10	100
Bin_test_error_rate	0.0735677083333	0.0755208333333	0.0963541666667
Bin_train_error_rate	0.0642740619902	0.0672104404568	0.0936378466558
Z_test_error_rate	0.068359375	0.0716145833333	0.0852864583333
Z_train_error_rate	0.0763458401305	0.0822185970636	0.0965742251223
L_test_error_rate	0.052734375	0.0520833333333	0.06640625
L_train_error_rate	0.0548123980424	0.0548123980424	0.0655791190865

Q4. K-Nearest Neighbors

1. Process



(1) Data preprocess

I preprocess the data using three different ways, Z-normalization, log-transform and binarization, which is the same as that has been mentioned above.

(2) Training process and calculate the distance

For the continuous data (z-normalization, log-transform) use the Euclidean distance to measure distance between neighbors. For the binarized data, use the Hamming distance.

(3) Testing process and calculate the error rate

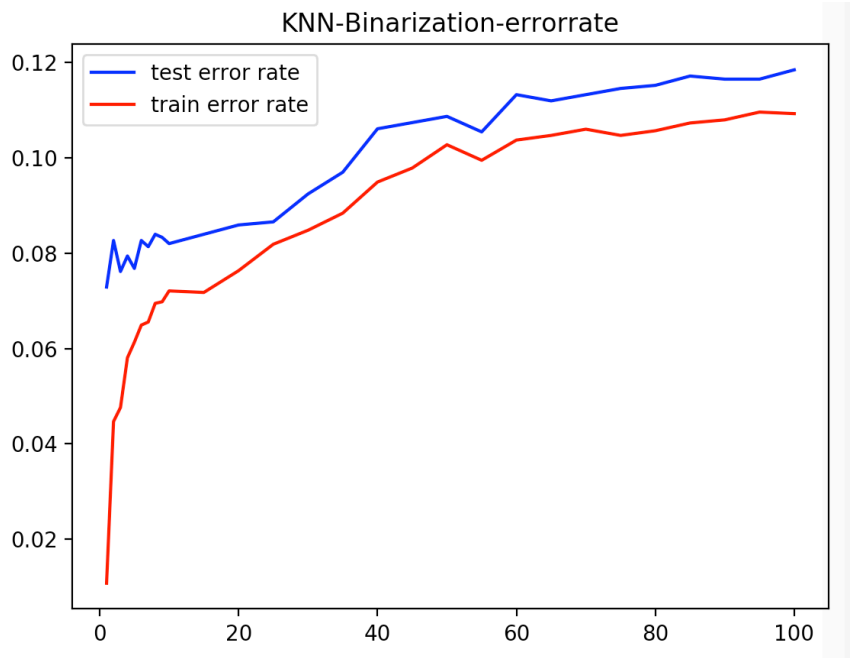
Through sorting these different distances I get before to find the K nearest neighborhoods, and count the class 1 and class 0 of these K nearest neighborhoods. Then, classify the testing data according to the higher

probability of class 1 or class 0 in these K nearest neighborhoods. Finally, calculate the error rate and sketch graphs.

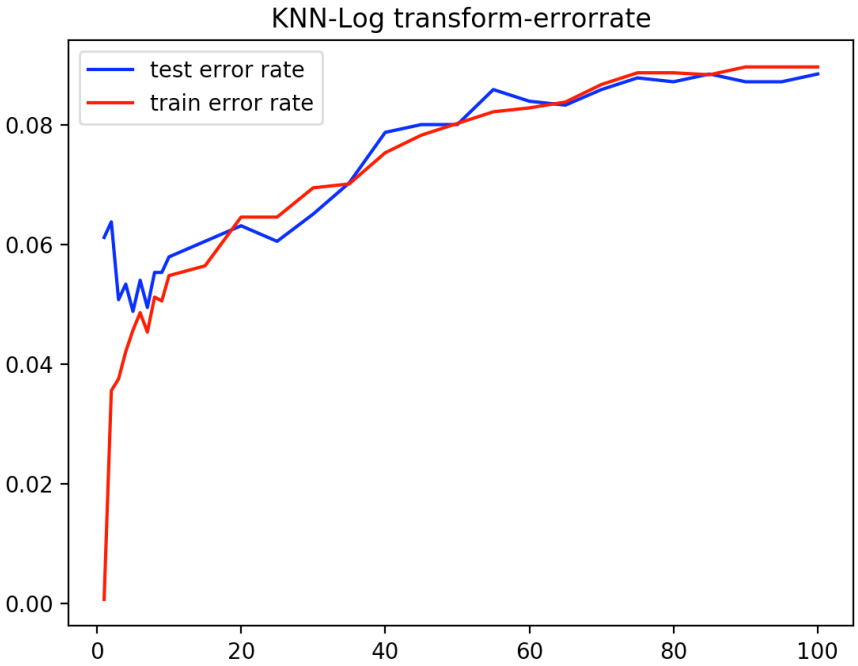
2. Result

(1) Plots of training and test error rates versus K

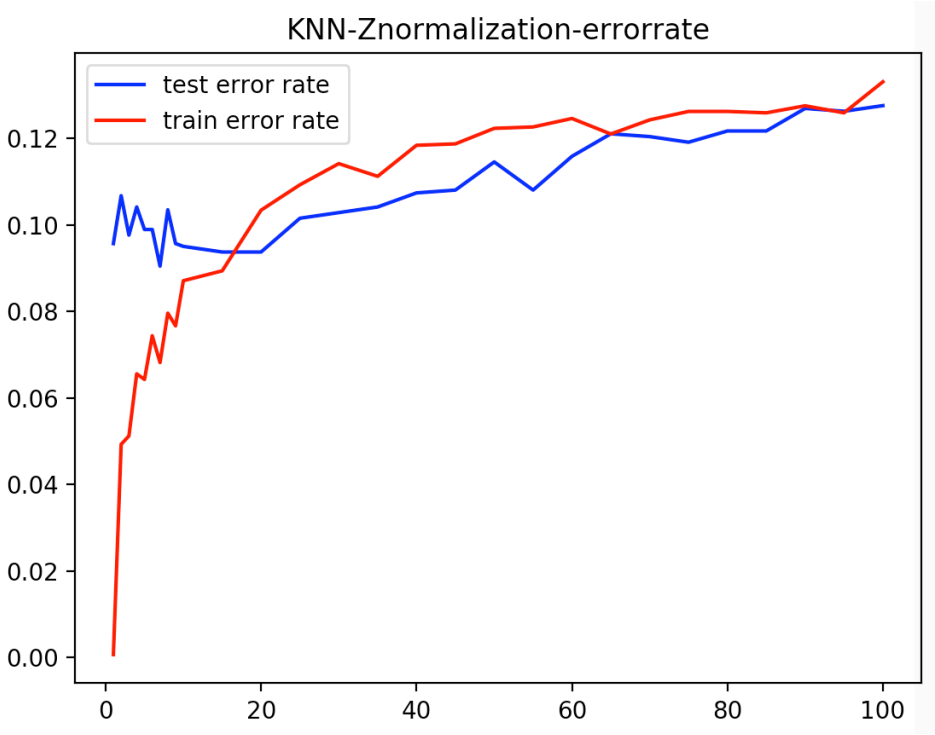
Binarization:



Log-transform:



Z-normalization:



(2) What do you observe about the training and test errors as K change? Why is training error not 0 when K = 1?

When K equals to 1, the training error rate is lowest, almost to zero. When K is between 0 and 10, the training error rate is lower than the testing one and the testing error rate curve fluctuates. After the point K equals to 20, two curves prove a similar trend under three different situations as K rises. Overall, the testing one prove a more stable rising trend.

When K=1, only one nearest point will be found, which will label the new data under testing just by one point. But there may be some other points also having the same distance as the chosen one. In other words, choosing only one point to predict a new data is not accurate because there will be more uncertainty.

Specifically, for Binarization, we set all the values except 0 to 1. Then we calculate the Hamming distance, which produces not only one nearest distance with a relatively high probability. Therefore, when K=1, training error rate under Binarization is the highest among three preprocessing ways. For Z-normalization and Log-transform, I calculate the Euclidean distance, which produces same nearest distance with low probability. Therefore, training error rates are really low in these tow preprocessing ways.

(3) What do you observe about the error rates of the different preprocessing strategies?

Log transform performs better than the other two preprocessing ways with the lowest error rate. And the Binarization is a little better than Z-normalization.

(4) Training and testing error rates for K = 1, 10 and 100.

K	1	10	100
Bin_test_error_rate	0.0729166666667	0.0833333333333	0.0930989583333
Bin_train_error_rate	0.010766721044	0.0721044045677	0.109298531811
Z_test_error_rate	0.095703125	0.0950520833333	0.127604166667
Z_train_error_rate	0.000652528548124	0.0871125611746	0.133115823817
L_test_error_rate	0.0611979166667	0.0579427083333	0.0885416666667
L_train_error_rate	0.000652528548124	0.0548123980424	0.089722675367

Q5. Survey

I have spent about 5 days on this assignment.
About 1 day per question (5 hours), 1 day to write and modify this report.
In Q3, I meet the overflow and not full rank matrix problem. But finally solved.

Feedback:
There are some issues on transferring the theoretical knowledge to practical operation, which needs most time in this whole process. Matrix operation is faster than algebra. Library function is more optimization.

NAME: FEI YANG A0169096N
DATE: 20 Feb 2018