EE5907 CA1 Programming Assignment

Introduction: In this spam email filtering project, four models (Beta-binomial Naive Bayes, Gaussian Naive Bayes, Logistic Regression and K-Nearest Neighbours) are implemented to demonstrate classification functions in pattern recognition fields. The spam email dataset consists of 4601 email messages with 57 features. And the dataset is split into training set (3605 emails) and test set (1536 emails) with classified labels (1 = spam, 0 = not spam). The dataset is in MATLAB format. The four models are implemented in Python 3.7.4.

1. Beta-binomial Naive Bayes

Fit a Beta-binomial Naive Bayes model on the binarized data. The training and testing data are binarized. When data value is greater than 1, it will be set to 1, otherwise, it will be set to 0. The class label prior λ can be estimated using Maximum Likelihood $\lambda_{ML} = N_c/N$. Assume a prior Beta(α,α) on the feature distribution. The likelihood for each feature can be express as $p(x_j = 1|y=c,D) = (N_{jc} + \alpha)/(N+2\alpha)$.

1.1 Plots of training and test error rates versus α

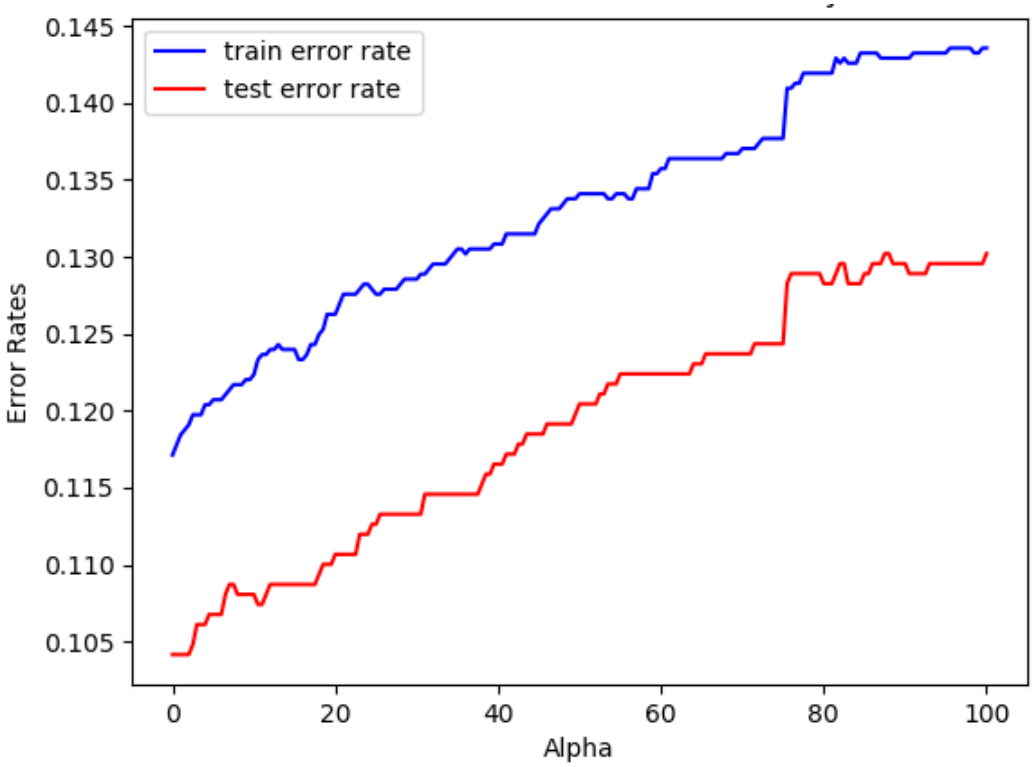

Figure 1-1 Plots of training and test error rates versus α for Beta-binomial Naive Bayes

As shown on the Figure 1-1, both training error rate and test error rate grow up as α increases. And test error rate is lower than train error rate.

1.2 Training and testing error rates for α = 1, 10 and 100

α	Training error rate	Testing error rate
1	0.118434	0.104167
10	0.122349	0.108073
100	0.134095	0.120442

Table 4-1 Training and testing error rates for α = 1, 10 and 100

1.3 Observation on the training and test errors as α change

From the training and testing Error Rates curve, the testing error rate is lower than the training error rate. The testing error rate increases from 10.4% to 12% as α goes from 0 to 100. The testing error rate is lowest When α is 0. That is to say the prior Beta(α, α) on the feature distribution may not be a proper assumption. In other words, we can use Bernoulli to estimate feature distribution.

2. Gaussian Naive Bayes

Fit a Gaussian Naive Bayes model on the log-transformed data. The class label prior λ can be estimated using ML, $\lambda = N_c/N$. Use maximum likelihood to estimate the class conditional mean and variance of each feature and use ML estimates as a plug-in estimator for testing. I use $\log(x_i+0.1)$ to pre-process training and testing data.

2.1 Results

Training error rate	Testing error rate
0.165742	0.171224

Table 2-1 Training and testing error rates for the log-transformed data

2.2 Analysis

The testing error rate is very close to the training error rate. In other words, the Gaussian Naïve Bayes model is effective on the data set. However, compare the testing error rate obtained in Gaussian Naïve Bayes model with testing error rate obtained from Beta-binomial Naive Bayes, we can observe that testing error rate in Gaussian Naïve Bayes model is obviously higher. Therefore, the trained Beta-binomial Naive Bayes model is better for filtering spam email than Gaussian Naïve Bayes model.

3. Logistic Regression

Fit a logistic regression model with L_2 regularization on the log-transformed data. I use $\log(x_i+0.1)$ to pre-process training and testing data. I also add one more column of 1 in both training and testing data set as bias term.

3.1 Plots of training and test error rates versus λ

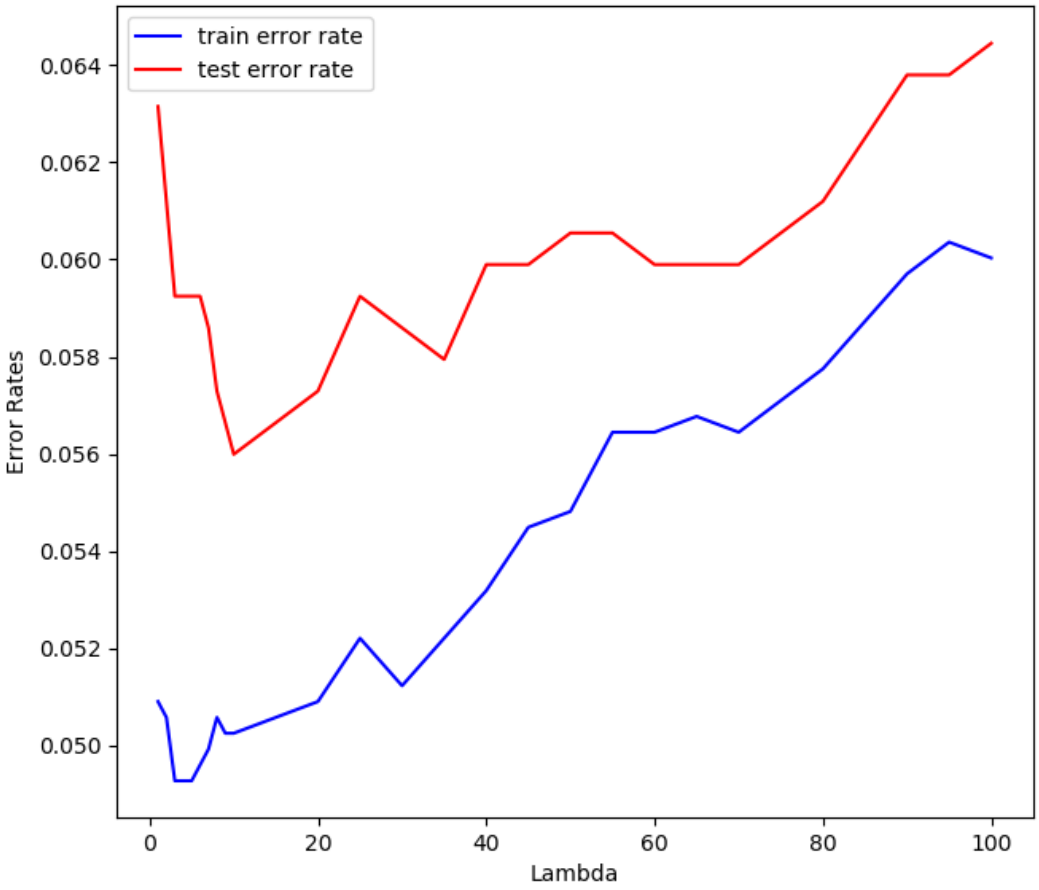

Figure 3-1 Plots of training and test error rates versus λ (1 - 100) for Logistic regression

As shown on the Figure 3-1, both training error rate and test error rate grow up as λ increases. And test error rate is higher than train error rate.

3.2 Results

Lambda(λ)	Training error rates	Testing error rates
1	0.050897	0.063151
10	0.050245	0.055990
100	0.060033	0.064453

Table 3-1 training and testing error rates for λ = 1, 10 and 100 for Logistic regression

3.3 Analysis

From the training and testing Error Rates curve, the test error rate decreases as λ grows from 0 to 10. And the test error rate increases again when λ grows up from 10 to 100. The L_2 regularization works well in penalizing high value weights when λ is around 10. However, if lambda is very large then it will add too much weight and it will lead to under-fitting. In other words, it's important how lambda is chosen.

4. K-Nearest Neighbours

Fit a K-Nearest Neighbours (KNN) model on the log-transformed data. I use $\log(x_i+0.1)$ to pre-process training and testing data.

4.1 Plots of training and test error rates versus K

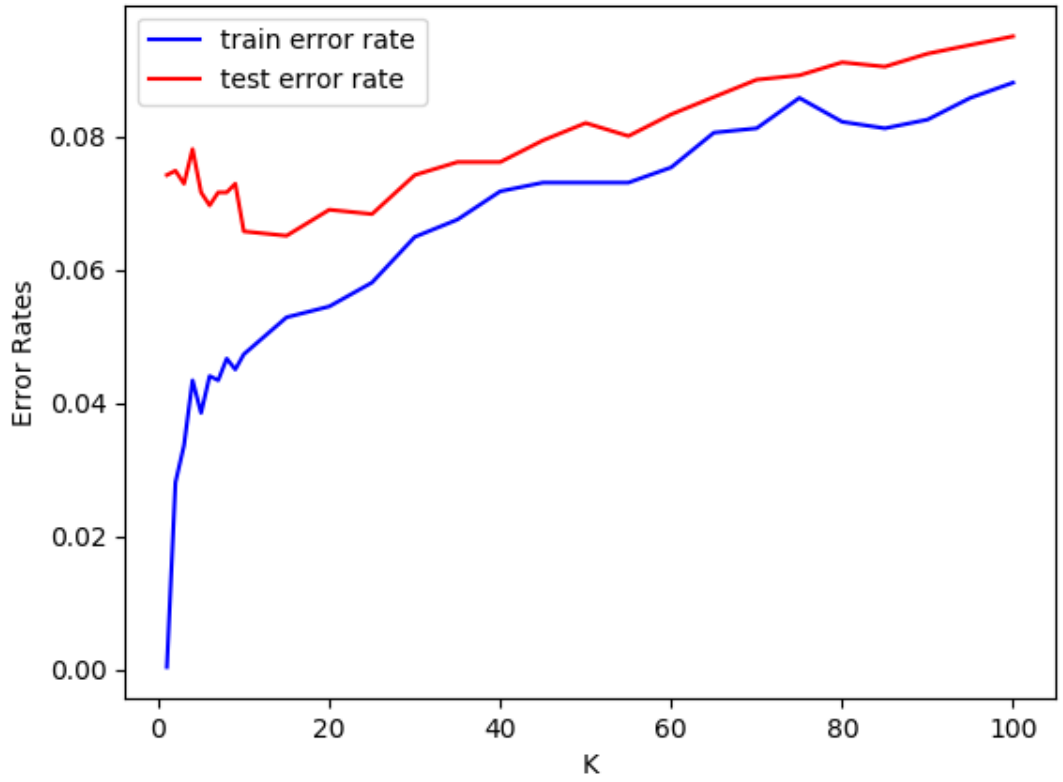

Figure 4-2 Plots of training and testing error rates versus K for KNN (log-transform)

As shown on the Figure 4-1, both training error rate and test error rate grow up as λ increases. And test error rate is higher than train error rate. Test set have the best performance when K = 15.

4.2 Results

K	Training error rate	Testing error rate
1	0.000326	0.074219
10	0.044731	0.065755
100	0.088091	0.095052

Table 4-1 Training and testing error rates for K = 1, 10 and 100

4.3 Analysis

The training data set is almost perfectly predicted when K = 1. The bias will be 0 when K = 1. However, when it comes to test data set, it has higher chance to be an error, which causes high variance. When K increases, the training error rate will increase, but the test error rate may decrease at the same time (K = 0~15). Both training error rate and testing error rate fluctuate near K = 10. A small value of K means that noise will have a higher influence on the result. As K keep increasing from 15, both training error rate and test error rate will increase, but larger values of K will have smoother decision boundaries which mean lower variance but increased bias.

5. Summary

Compare the classification performance of four models, Logistic Regression and KNN models can achieve better classification accuracy in this spam email filtering assignment. For KNN model, it is simple to implement, but computation cost is quite high because we need to compute the distance of each data instance to all training samples when training sample is huge.

6. Survey

I spent 30 hours on this assignment. This assignment helps me understand these models and algorithms well from theories to practice.