Homework 3

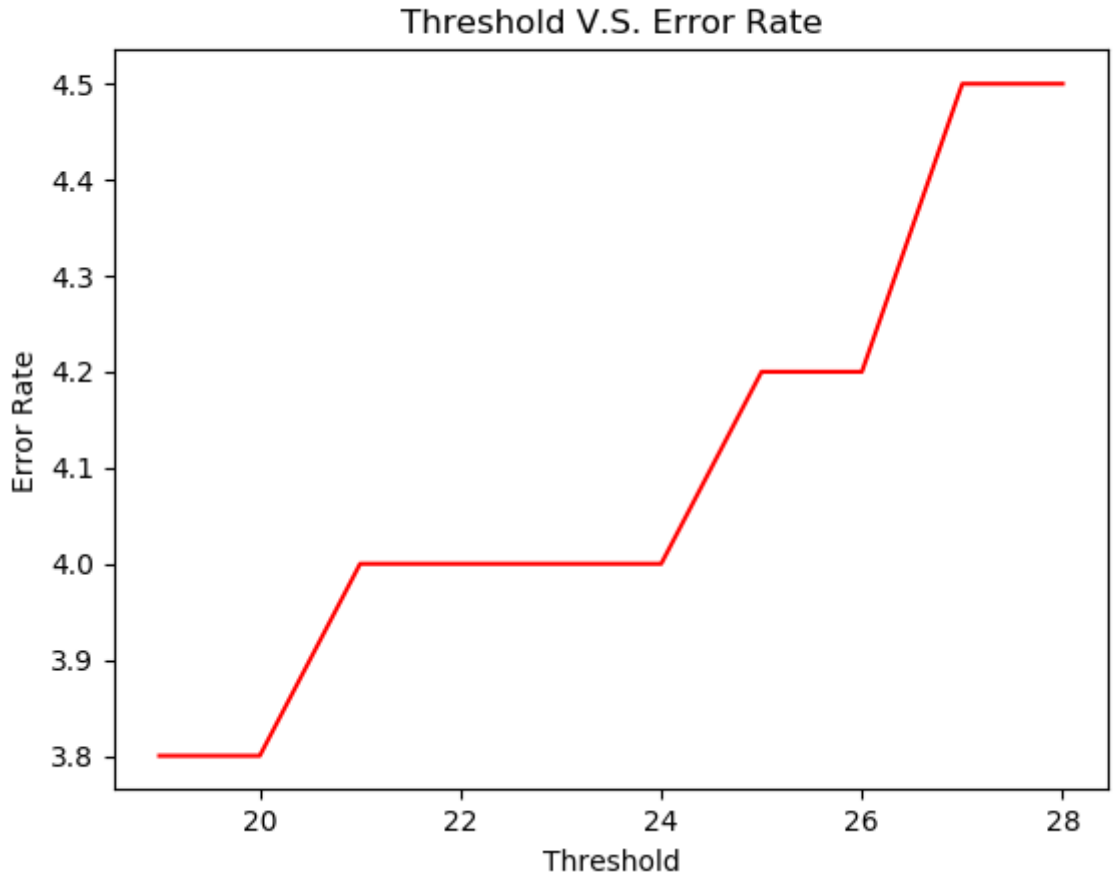
Wenhe Li wl1508

Problem 1

```
python problem-1.py spam_data/spam_train.txt
```

- a The validation error rate is 5.7%
- b

- c



From the graph, we can tell that the best config for threshold is 23.

- d

```
MLE::Validation error,  # =   57,  % =   5.7000%.
MAP::Validation error,  # =   42,  % =   4.2000%.
```

For this model we can see `MAP` performs better. Because we apply a prior to our data distributation, we can gaurantee there is no `0`. Therefore, we can use `log` to avoid `0` problem and underflow to improve accuracy.
Without MAP, we will get a `zero` in our expression which is a much strong bias that effect our accuracy.

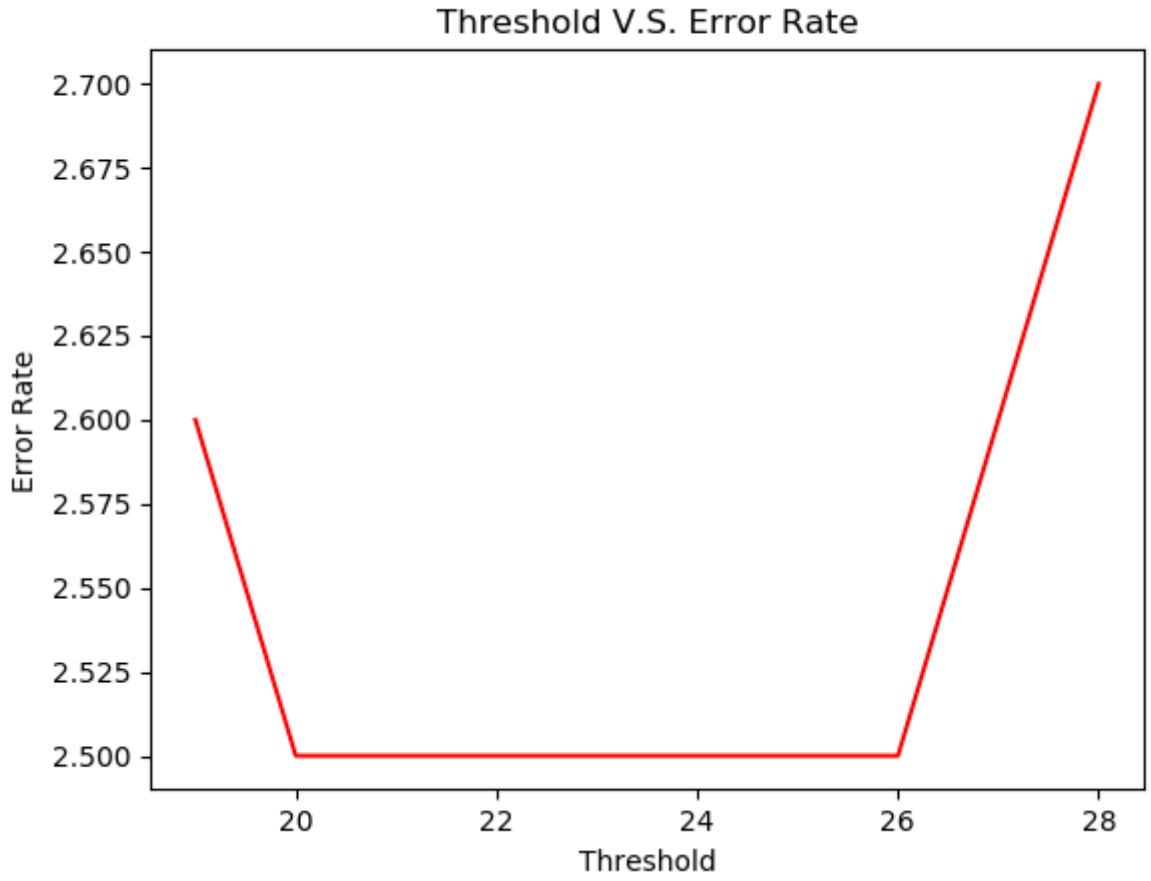
Problem 2

```
python problem-2.py spam_data/spam_train.txt
```

- a The validation error rate with MLE is 21.3%, 2.5% with MAP.
- b


Training Size V.S. Error Rate
train
validation
5.0
4.5
4.0
3.5
3.0
2.5
Error Rate
500
1000
1500
2000
2500
3000
3500
4000
Training Size

- c



From the graph, we can tell from 20 - 26 the hyperparameter works best.

- d

```
MLE::Validation error, # =  213, % =  21.3000%.
MAP::Validation error, # =   25, % =   2.5000%.
```

The `MAP` performs better.

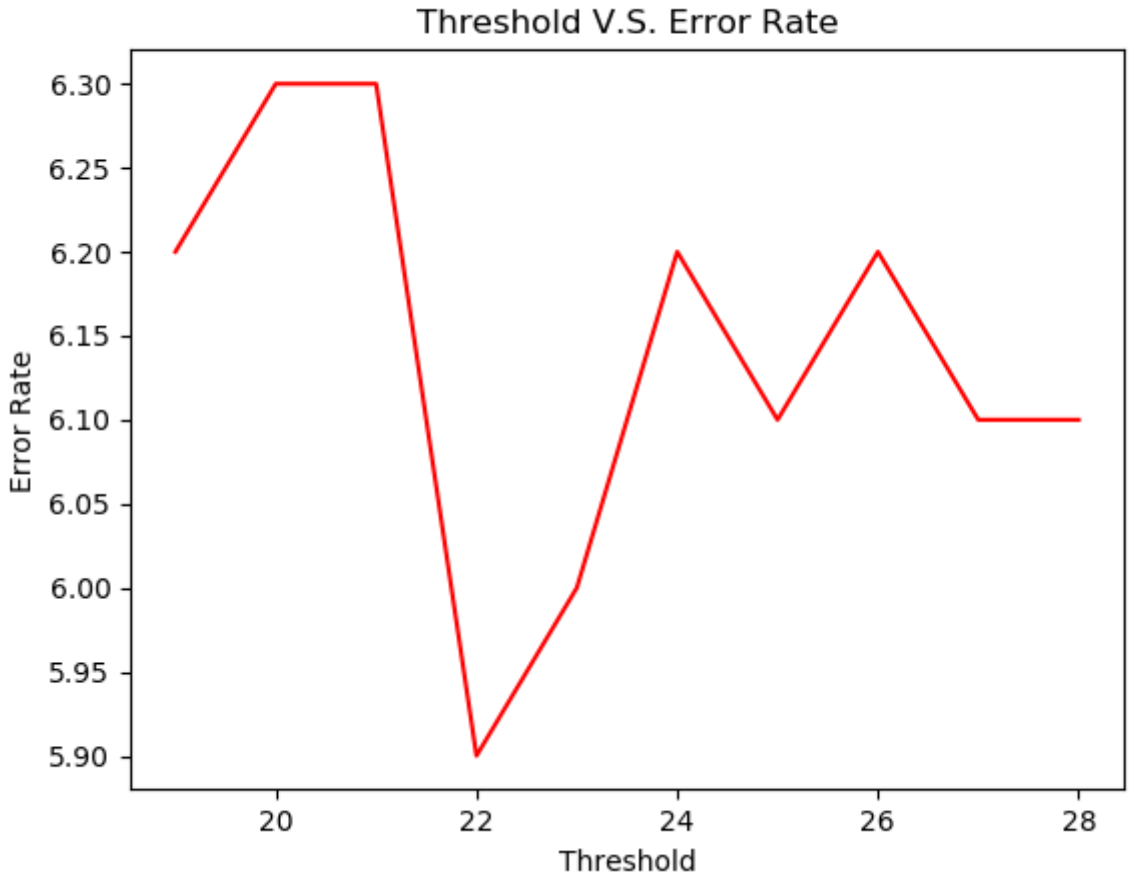
Problem 3

```
python problem-3.py spam_data/spam_train.txt
```

- a With MAP it can achieve 6.2% error rate.
- b


Training Size V.S. Error Rate
train
validation
Error Rate
Training Size
16
14
12
10
8
6
4
2
500
1000
1500
2000
2500
3000
3500
4000

- c



- d

  `MAP` is 6.2%

  `MLE` is 26.9%

  Beacuse by applying `MAP` , we can use `log` to prevent underflow from happening.

  Besides, by using `MAP` we can assure the mean and variance can not be zero to gaurantee our model is not poisioned by dirty data point.

Probelm 4

The `perceptron` algorithm can only deal with a relatively low dimension of data distributation.

However, `Naive Bayes` applies a strong persumption that all features are independent. Besides, `Naive Bayes` is a generative model and `perceptron` is a discriminative model.

When it comes to `Gaussian Naive Bayes` , it makes an assumption that all features follow a Normal Distributation, which is a really strong assumption. Normally, some tasks that are required a continous features should use `GNB` .