

**LINK TO GITHUB REPO: [HERE](#)**

### Results and Discussion

1. What is the effect of removing stop words in terms of precision, recall, and accuracy? Show a plot or a table of these results.

NaiveBaye's Classifier		NaiveBaye's Classifier without Stopwords	
Evaluation Metrics	Score	Evaluation Metrics	Score
Accuracy	95.64%	Accuracy	94.74%
Recall	96.60%	Recall	95.44%
Precision	96.93%	Precision	96.71%

- The Classifier becomes less reliable in terms of the three evaluation metrics if the stop words were removed.

2. Experiment on the number of words used for training. Filter the dictionary to include only words occurring more than k times (1000 words, then  $k > 100$ , and  $k = 50$  times). For example, the word "offer" appears 150 times, that means that it will be included in the dictionary.

```
Summary of Results:
Threshold k=50:
  Accuracy: 92.34%
  Recall: 92.21%
  Precision: 96.28%
Threshold k=100:
  Accuracy: 91.37%
  Recall: 90.97%
  Precision: 96.02%
Threshold k=1000:
  Accuracy: 85.99%
  Recall: 86.85%
  Precision: 91.94%
```

- The results were shocking i thought that as you filter the dictionary the more you'll get accurate as you can spot the word's pattern more specifically because of low pool of words however the results tells otherwise.

3. Discuss the results of the different parameters used for Lambda smoothing. Test it on 5 varying values of the  $\lambda$  (e.g.  $\lambda = 2.0, 1.0, 0.5, 0.1, 0.005$ ), Evaluate performance metrics for each.

Results for Lambda = 2.0:

Accuracy: 95.44%  
Recall: 96.38%  
Precision: 96.84%  
True Positive (TP): 10820  
True Negative (TN): 5063  
False Positive (FP): 353  
False Negative (FN): 406

Results for Lambda = 1.0:

Accuracy: 95.64%  
Recall: 96.60%  
Precision: 96.93%  
True Positive (TP): 10844  
True Negative (TN): 5072  
False Positive (FP): 344  
False Negative (FN): 382

Results for Lambda = 0.1:

Accuracy: 95.55%  
Recall: 96.33%  
Precision: 97.06%  
True Positive (TP): 10814  
True Negative (TN): 5088  
False Positive (FP): 328  
False Negative (FN): 412

Results for Lambda = 0.5:

Accuracy: 95.69%  
Recall: 96.62%  
Precision: 96.98%  
True Positive (TP): 10847  
True Negative (TN): 5078  
False Positive (FP): 338  
False Negative (FN): 379

Results for Lambda = 0.005:

Accuracy: 95.27%  
Recall: 95.82%  
Precision: 97.13%  
True Positive (TP): 10757  
True Negative (TN): 5098  
False Positive (FP): 318  
False Negative (FN): 469

- There was not much difference or a notable trend upon testing on various lambda values. Based on what I've researched about Laplace smoothing, it generally solves issues regarding absent features or 0 value features however our features in this naivebayes classifier is from the email set itself? So meaning that all features (words) are present in the dataframe therefore not much difference in the metrics of the classifier.