# Final Report

## Implementation of the Naive Bayes Spam Detector

In our project, we used the Naive Bayes technique to construct a spam detector. This technique is a well-liked option for text categorization jobs since it is straightforward, effective, and efficient [1].

Our implementation approach included multiple steps, such as feature extraction from the data, model training, and evaluation. We transformed the text input into numerical features using a bag-of-words model. The dataset's individual words were each handled as a feature. We assessed the model's performance on the test data after training it on the training set of data.

## Visualization and Interpretation of Results

On the test dataset, the Naive Bayes classifier had an accuracy of roughly 86.44%. Even though this is a good outcome, it signals that more work has to be done.

```
Correct: 963
Incorrect: 151
Accuracy: 0.8644524236983842
```

| | Label | SMS | predicted |
|---|---|---|---|
| 0 | ham | Later i guess. I needa do mcat study too. | ham |
| 1 | ham | But i haf enuff space got like 4 mb... | ham |
| 2 | spam | Had your mobile 10 mths? Update to latest Oran... | ham |
| 3 | ham | All sounds good. Fingers . Makes it difficult ... | ham |
| 4 | ham | All done, all handed in. Don't know if mega sh... | ham |

## Potential Problems and Suggestions for Improvement

Although our concept worked well, it has a number of potential drawbacks:

1. **Handling Unseen Words:** Our current model may not be particularly effective at handling unseen words, which are words that appear in test data but not in training data.
2. **Assumption of Independence:** Naive Bayes makes the assumption that each feature is independent, which isn't necessarily true in natural language.
3. **Imbalanced Data:** Dataset is severely unbalanced, containing far more ham messages than spam. This might make the model more likely to predict ham.

Our model could be enhanced by taking into account the following:

1. **Handling Unseen Words**: Implement a method for dealing with unnoticed words, such example by employing smoothing techniques.
2. **Feature engineering**: We could utilize more intricate features, like n-grams [2], as opposed to treating each word as a separate feature.
3. **Techniques for Resampling:** To solve the issue of unbalanced data, we could employ sampling strategies like oversampling minority groups or undersampling majority groups [3].

# Stretch Goals

In our upcoming efforts, we want to:

1. **Test Our Model on more Text Classification Datasets:** We intend to test our model on more text classification datasets in order to further assess its generalizability.
2. **Cross-validation:** We intend to use a cross-validation strategy in our model evaluation in order to get a more reliable estimate of the model's performance.

# References

[1] Rish, Irina. "An empirical study of the naive Bayes classifier." IJCAI 2001 workshop on empirical methods in artificial intelligence. Vol. 3. 2001.

[2] Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." Advances in neural information processing systems 26 (2013).

[3] Chawla, Nitesh V., et al. "SMOTE: synthetic minority over-sampling technique." Journal of artificial intelligence research 16 (2002): 321-357.