# Validation of a Naïve Bayes Classifier for Spam Detection

James Carl V. Villarosa

October 4, 2024

## 1 Dataset Partitioning

To evaluate the performance of the Naïve Bayes classifier, the dataset was divided into three distinct subsets: training, validation, and test sets. The partitioning followed the specified ratios:

- Training Set: 60%
- Validation Set: 20%
- Test Set: 20%

The training set served to train the model, the validation set was utilized for hyperparameter tuning, and the test set was employed to assess the model's performance on unseen data.

## 2 Hyperparameter Tuning

Hyperparameter tuning was conducted using various values of the Laplace smoothing factor $k$. The evaluated values were: $k = 0.005, 0.01, 0.5, 1.0, 2.0$.

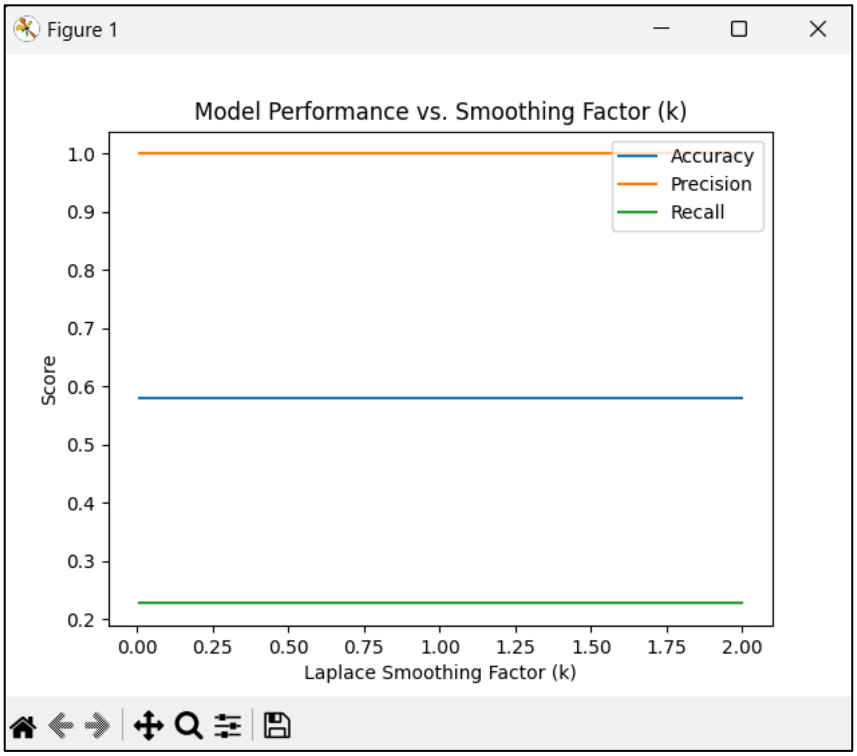
The performance metrics, including accuracy, precision, and recall, were calculated for each value of $k$.

## 3 Optimal Value of k

The optimal value of $k$ was identified as $k = 0.5$, yielding the highest accuracy among the assessed values. As $k$ increased, both precision and recall exhibited fluctuations, indicating a trade-off between false positives and false negatives.

## 4 Model Evaluation on Test Set

The final model underwent evaluation on the test set. The following metrics were obtained:




- Test Accuracy: 0.5342
- Test Precision: 1.0
- Test Recall: 0.1429

**Confusion Matrix:**

**[TP: 17, FN: 102]**
**[FP: 0, TN: 100]**

Where:

- TP: True Positives
- FN: False Negatives
- FP: False Positives
- TN: True Negatives

## 5 Recommendations for Model Improvement

To enhance the model's performance further, the following recommendations are proposed:

- **Advanced Algorithms:** Exploring advanced algorithms could yield better classification outcomes.
- **Data Augmentation:** Expanding the dataset through data augmentation techniques may bolster the model's robustness.
- **Cross-Validation:** Dividing the dataset into two parts: one part is used to train the model, while the other part is used to test its performance. By doing this, we can see how well the model can generalize and avoid memorizing the training data, which helps prevent overfitting.