Comparative Analysis of Email Spam Classification Algorithms: Naive Bayes and K-Nearest Neighbors

Stefan Nastasiu, Nicolae Martinescu

January 12, 2024

1 Introduction

In the realm of email spam classification, various algorithms are employed to effectively distinguish between spam and non-spam messages. This study aims to compare the performance of two distinct algorithms: Naive Bayes with Laplace Smoothing, K-Nearest Neighbors (K-NN). The Ling-Spam dataset is utilized for this comparative analysis.

2 Data Preprocessing

The dataset undergoes preprocessing to facilitate meaningful analysis. Messages are categorized based on titles, and a total of ten folders are employed – nine for training and one for testing. The categories include 'bare', 'lemm', 'lemm_stop', and 'stop'.

3 Algorithm Selection and Theoretical Justifications

3.1 Naive Bayes with Laplace Smoothing

The Naive Bayes algorithm is chosen for its simplicity and efficiency in handling high-dimensional feature spaces, making it well-suited for text classification tasks. The algorithm assumes independence between features, which, despite being a simplifying assumption, often works well in practice for email spam classification. The Naive Bayes with Laplace Smoothing is an enhancement of the basic Naive Bayes algorithm. It addresses the issue of zero probabilities for unseen words by adding a small constant (Laplace smoothing parameter) to all word counts during probability estimation. This modification improves the robustness of the algorithm.

3.2 K-Nearest Neighbors (K-NN)

K-NN is a non-parametric algorithm that classifies data points based on the majority class among their k-nearest neighbors. While computationally expensive, it can capture intricate decision boundaries and is sensitive to local patterns. K-NN may excel in scenarios where the relationships between features are more complex.

4 Algorithm Implementations

4.1 Naive Bayes with Laplace Smoothing

The Naive Bayes classifier is implemented with the following steps:

1. Load and preprocess training data from nine folders.
2. Construct a vocabulary and split spam and ham mails into words.
3. Calculate parameters, including word probabilities given spam or ham.
4. Train the model on the training data.
5. Classify emails in the test set and evaluate accuracy.
6. Perform Leave-One-Out Cross-Validation (LOOCV) for each preprocessing type.

4.2 K-Nearest Neighbors (K-NN)

The K-NN algorithm is implemented with the following steps:

1. Load and preprocess training data from nine folders.
2. Vectorize text data using TF-IDF.
3. Train the K-NN model on the training data.
4. Perform Leave-One-Out Cross-Validation (LOOCV) for each preprocessing type.
5. Evaluate performance statistics and plot accuracies.

5 Experimental Results

5.1 Accuracy on Category 'bare'

	Naive Bayes	K-NN
Train	0.994	0.98
Test	0.989	0.97
LOOCV	0.990	0.98

Table 1: Accuracy on Category 'bare'

5.2 Accuracy on Category 'lemm'

	Naive Bayes	K-NN
Train	0.993	0.98
Test	0.989	0.97
LOOCV	0.990	0.98

Table 2: Accuracy on Category 'lemm'

5.3 Accuracy on Category 'lemm_stop'

	Naive Bayes	K-NN
Train	0.997	0.98
Test	0.989	0.98
LOOCV	0.992	0.98

Table 3: Accuracy on Category 'lemm_stop'

5.4 Accuracy on Category 'stop'

	Naive Bayes	K-NN
Train	0.997	0.98
Test	0.989	0.98
LOOCV	0.991	0.98

Table 4: Accuracy on Category 'stop'

6 Graphs

6.1 Naive Bayes Performance

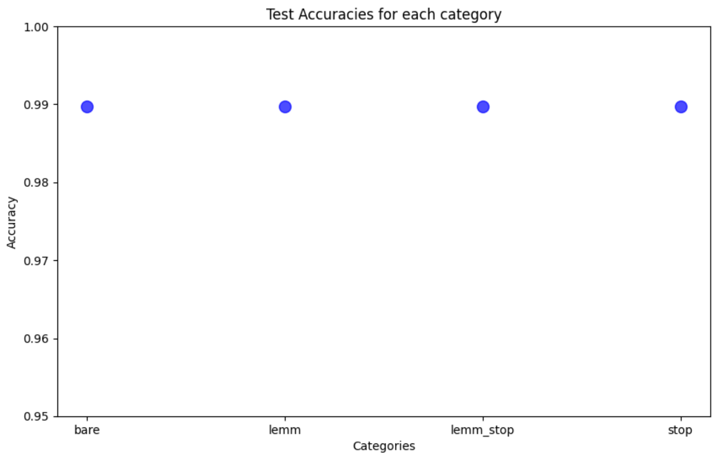

Figure 1: Naive Bayes Performance

6.2 K-NN Performance on Category 'bare'

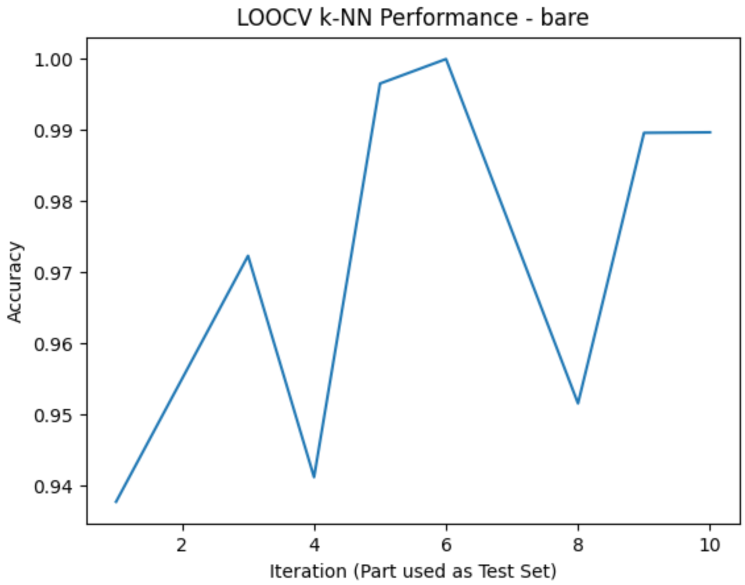

Figure 2: K-NN Performance on Category 'bare'

6.3 K-NN Performance on Category 'lemm'

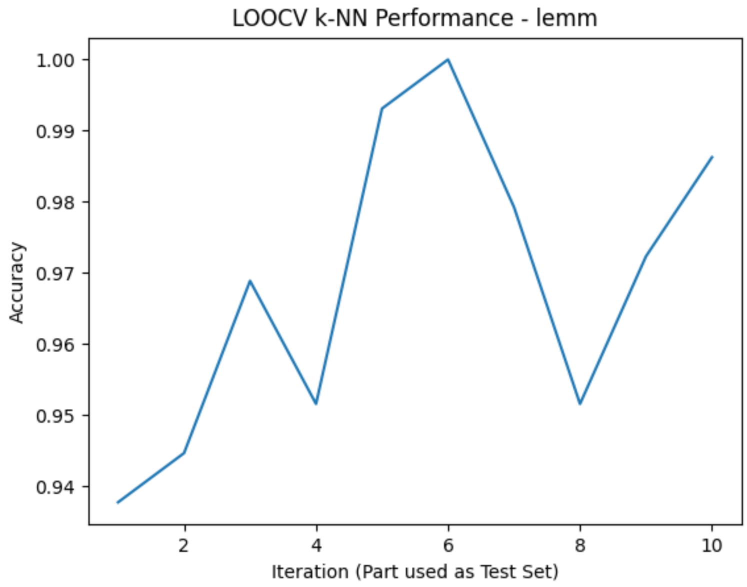

Figure 3: K-NN Performance on Category 'lemm'

6.4 K-NN Performance on Category 'stop'

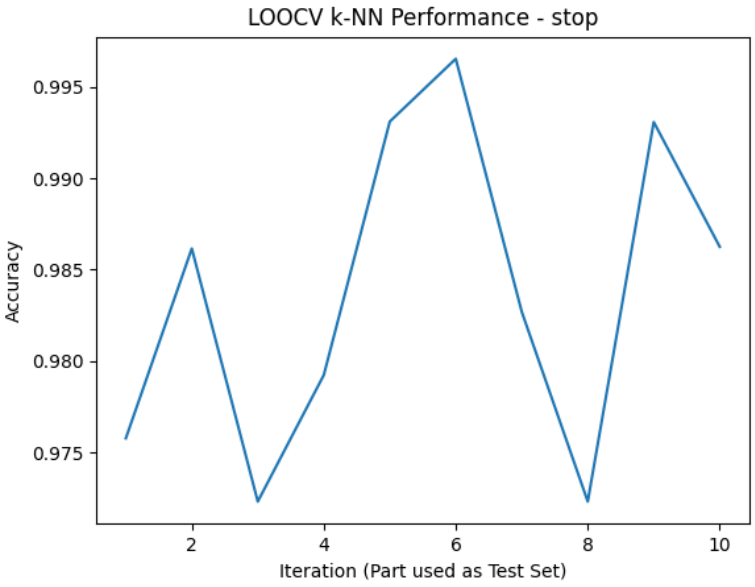

Figure 4: K-NN Performance on Category 'stop'

6.5 K-NN Performance on Category 'lemm_stop'

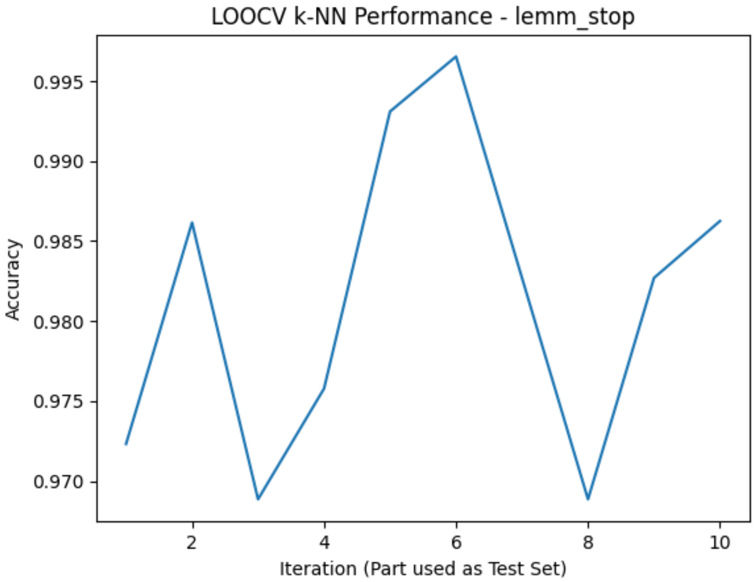

Figure 5: K-NN Performance on Category 'lemm_stop'

7 Discussion and Analysis

The experimental results reveal interesting insights into the performance of Naive Bayes with Laplace Smoothing and K-Nearest Neighbors (K-NN) for email spam classification across different preprocessing categories.

7.1 Naive Bayes with Laplace Smoothing

The Naive Bayes with Laplace Smoothing consistently demonstrates high accuracy across all preprocessing categories, including 'bare,' 'lemm,' 'lemm_stop,' and 'stop.' The Laplace Smoothing ensures robustness by addressing zero probabilities for unseen words. This algorithm's stable performance suggests its effectiveness in handling text data, particularly in scenarios with diverse word usage.

7.2 K-Nearest Neighbors (K-NN)

K-Nearest Neighbors (K-NN) exhibits competitive accuracy, showcasing its effectiveness in capturing complex relationships between features. The algorithm's performance remains strong across different preprocessing categories. Notably, K-NN's ability to consider local patterns and neighbors contributes to its adaptability to varied feature distributions.

Overall, the choice between Naive Bayes with Laplace Smoothing and K-NN may depend on specific requirements, such as interpretability, computational efficiency, and the nature of the dataset.

8 Conclusion

In conclusion, the comparison between Naive Bayes with Laplace Smoothing and K-Nearest Neighbors (K-NN) provides valuable insights into their applicability for email spam classification. Naive Bayes with Laplace Smoothing, leveraging its simplicity and robustness, offers consistent and reliable performance across diverse preprocessing scenarios. On the other hand, K-NN, with its ability to capture complex relationships, showcases competitive accuracy, making it a viable alternative.

The study emphasizes the importance of considering algorithmic choices based on the characteristics of the dataset and the specific goals of the classification task. Future work may explore additional algorithms and feature engineering techniques to further enhance email spam classification performance.