Classifying Spam Messages on the SMS Spam Collection Dataset

Paulo Wook Kim
Email: paulowk@al.insper.edu.br

I. Dataset

The *SMS Spam Collection Dataset* [1] is a publicly available corpus containing 5,574 SMS messages in English, each labeled as either legitimate (*ham*) or *spam*. Collected Kaggle, the dataset aims to support research in text-based spam detection. The business purpose is to enhance spam filtering systems for mobile communication providers, thereby improving user experience by reducing unsolicited messages and protecting users from potential fraud or malicious content.

II. Classification Pipeline

The classification pipeline consists of several preprocessing steps and feature engineering techniques:

A. Text Preprocessing

- **Text Cleaning**: Removal of non-alphabetic characters and conversion of text to lowercase to standardize the data.
- **Tokenization**: Splitting the cleaned text into individual words (tokens) using NLTK's `word_tokenize` function.
- **Stopword Removal**: Eliminating common English stopwords (e.g., "the", "is") using NLTK's stopword list to focus on meaningful words.
- **Lemmatization**: Reducing words to their base or dictionary form using WordNet Lemmatizer, which helps in treating different forms of a word as a single term.

B. Feature Engineering

After preprocessing, *Term Frequency-Inverse Document Frequency (TF-IDF)* vectorization to convert text data into numerical features.

C. Classification Model

We used the *Multinomial Naive Bayes classifier* [2] for classification. This model is suitable for text data as it considers word frequencies and assumes feature independence. Words like "free", "win", and "prize" are strong indicators of spam, making the presence of particular words highly relevant for this classification problem.

III. Evaluation

We split the dataset into training and test sets using an 80/20 split while ensuring class distribution consistency through stratification. Given the class imbalance (approximately 13% spam), we used the *balanced accuracy score* for evaluation to account for both classes equally.

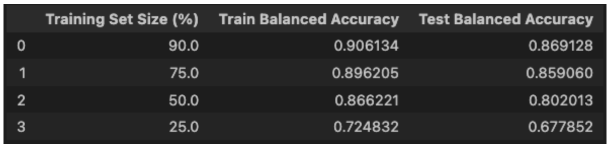
The Multinomial Naive Bayes classifier achieved a **balanced accuracy of 95.2%** on the test set. Key words influencing the classification included:

- **Spam Indicators**: "free", "win", "call", "now", "claim".
- **Ham Indicators**: "ok", "thanks", "tomorrow", "home", "yes".

The classifier effectively distinguishes spam based on these indicative words. However, legitimate messages containing typical spam words (e.g., "Call me now") could be misclassified, highlighting potential limitations due to word overlap.

IV. Dataset Size

We assessed model performance with varying training set sizes (90%, 75%, 50%, 25%) to evaluate the impact of dataset size on accuracy. The results are illustrated in Fig. 1.



	Training Set Size (%)	Train Balanced Accuracy	Test Balanced Accuracy
0	90.0	0.906134	0.869128
1	75.0	0.896205	0.859060
2	50.0	0.866221	0.802013
3	25.0	0.724832	0.677852

Fig. 1. Balanced accuracy versus training set size, showing diminishing returns in accuracy improvement with larger datasets.

A. Analysis

- The test balanced accuracy increased from 67.7% at 25% training size to 86.9% at 90% training size.
- The gains in accuracy diminish beyond a 75% training size, indicating limited benefits from adding more data.
- Given privacy concerns and data collection challenges, increasing the dataset size further may not be feasible for the business case.

V. Topic Analysis

We applied *Latent Dirichlet Allocation (LDA)* [3] to uncover underlying topics within the messages, resulting in three main topics:

1) **Promotions and Offers**: Words like "win", "prize", "free".
2) **Personal Communication**: Words like "hey", "tomorrow", "home".
3) **Financial Transactions**: Words like "account", "credit", "loan".

A. Error Distribution

- **Promotions and Offers**: High classification accuracy due to strong spam indicators.
- **Personal Communication**: Lower accuracy due to overlapping vocabulary with spam messages.
- **Financial Transactions**: Moderate accuracy; some legitimate messages were misclassified as spam.

B. Two-Layer Classifier Implementation

We developed a two-layer classification system:

1) **Topic Classification Layer**: Messages were first classified into topics using LDA.
2) **Topic-Specific Classifier Layer**: A separate Multinomial Naive Bayes classifier was trained for each topic.

C. Results

The two-layer classifier improved the overall balanced accuracy to **79.19%**. This approach allowed for context-specific analysis, reducing misclassifications, particularly in topics with vocabulary overlap. However, it performed worse than the Multinomial Naive Bayes classifier.

References

[1] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami, "Contributions to the study of SMS spam filtering: New collection and results," in *Proceedings of the 11th ACM Symposium on Document Engineering*, 2011, pp. 259–262.

[2] A. McCallum and K. Nigam, "A comparison of event models for naive bayes text classification," *AAAI-98 Workshop on Learning for Text Categorization*, vol. 752, pp. 41–48, 1998.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.