UCIrvine | The Paul Merage School of Business

BANA275

NATURAL LANGUAGE PROCESSING

SPAM SMS DETECTION

Prepared by:
MSBA Sec.A--Team 13
Yixi(Claire) Cheng
Sanya Wang
Wenqi(Vicky) Wang
Yuh-Shin(Ashley) Yen
Xinru(Violet) Zhao

Date: Mar. 20

Table of Contents

1.Introduction

This project aims to build a machine learning model to detect SMS messages as spam or ham (non-spam). Spam messages are growing fast in recent years, and they can be risky. So, detecting them is important. This project uses Natural Language Processing (NLP) methods. It includes text preprocessing, feature extraction, and machine learning models. We test different classifiers. We also use Synthetic Minority Oversampling Technique (SMOTE) to fix class imbalance.

We evaluate model performance using metrics such as accuracy, precision, recall, F1-score, and AUC. Naïve Bayes with TF-IDF (unigram + bigram + trigram) works best. We also build a live demo to show how the model detects spam in real time.

2.Data Preprocessing

This project includes a dataset of 5,574 SMS messages from kaggle[1].

Variables	Description
Class	If the message is normal, 'Class' is "ham". If the message is spam, 'Class' is "spam".
Message	Message text contents

Figure. Key Variables and Description

Before conducting analysis we first performed essential data preprocessing to ensure data quality and consistency. We initially examined the dataset for missing values and removed duplicate values. After this process, the dataset size was reduced from 5,574 rows to 5,159 clean records.

To standardize text and eliminate case sensitivity issues, all messages were converted to lowercase. Words were then lemmatized, reducing them to their base forms so that variations of the same word were treated as a single entity. Additionally, the text was subsequently tokenized

[1]Dataset
https://www.kaggle.com/datasets/mariumfaheem666/spam-sms-classification-using-nlp

into individual words, and common stop words such as "the," "is," and "and" were further removed to focus on more meaningful terms.

A word frequency study and wordcloud are also conducted to reveal significant patterns. The dataset is highly imbalanced. Ham messages make up 87.6%, while spam messages account for only 12.4%. This imbalance made spam detection harder. To fix this, we used SMOTE(Synthetic Minority Oversampling Technique) to improve model performance.

3.Word Cloud

Figure. Spam and Ham World Cloud

The word clouds were generated before removing stopwords. The spam word cloud prominently features words like "call," "free," "text," "win," "claim," "urgent," "prize," "reply". These words indicate that spam messages often contain promotional content, emphasizing free offers, cash rewards, and urgent responses. In comparison, the ham word cloud contains frequently used words such as "ok," "time," "home," "good," "love," "come," "leave", "going". These words indicate that ham messages focus on casual conversations, daily activities, and relationships. The presence of personal pronouns and common verbs (e.g., "got," "know") suggests more natural, conversational language.

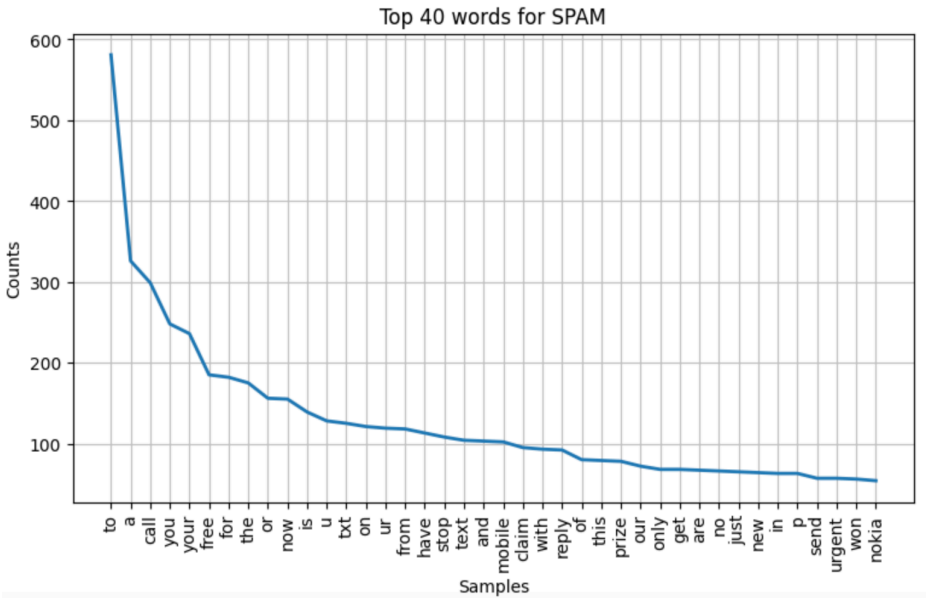


Figure. Top 40 Words for SPAM

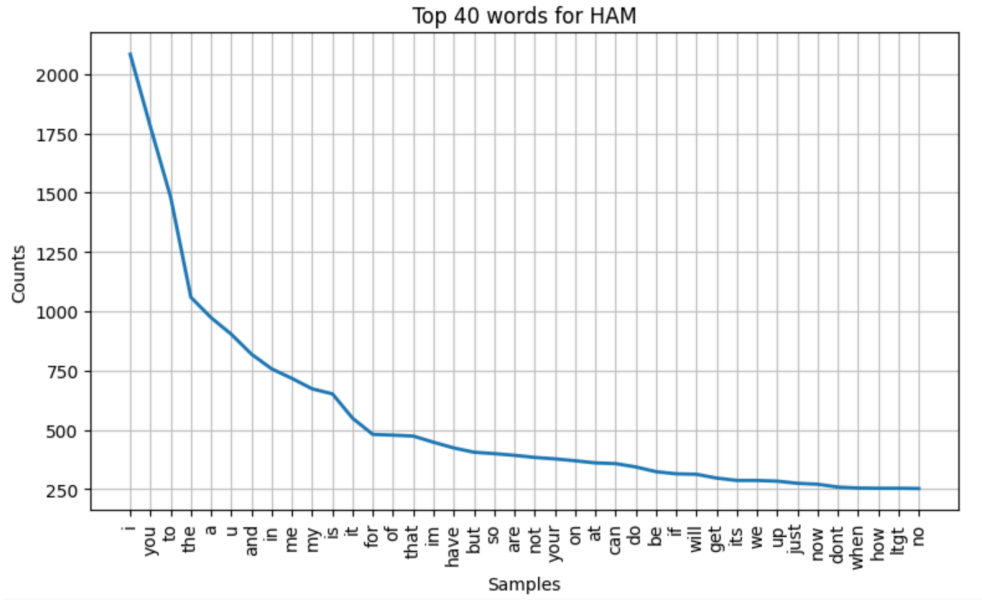


Figure. Top 40 Words for HAM

The frequency analysis of the top 40 words reveals in spam messages, the most frequent words include "now," "txt," "stop," “text,” " claim," “reply,” and “prize.”, which further reinforces the promotional and urgent nature of spam messages, as they often encourage recipients to take immediate action or respond to offers.However, stopwords such as "to," "a," and "your" also appear frequently, highlighting the need for stopword removal in further processing.

The frequency analysis of the top 40 words in ham messages reflects a dominance of pronouns and common function words, such as “get,” and “no.” This suggests that ham messages rely

more on personal interactions and everyday communication rather than promotional content.

Figure. World Cloud for Spam (Stopwords Removed)

The spam word cloud reveals frequent terms like "call," "free," "text," and "win," highlighting the promotional and urgent tone of spam messages. These words are often used to prompt immediate responses from recipients through offers or rewards. Many of these messages encourage users to claim rewards, subscribe to services, or engage in financial transactions, as evidenced by terms like "cash," "prize," "win," and "mobile."

Figure. World Cloud for Ham (Stopwords Removed)

In contrast, the ham word cloud consists of more conversational and relational words, including "ok," "time," "home," "good," "love," "come," "later," and "going." These words suggest that ham messages are primarily focused on personal communication, discussing daily activities, relationships, and informal exchanges.

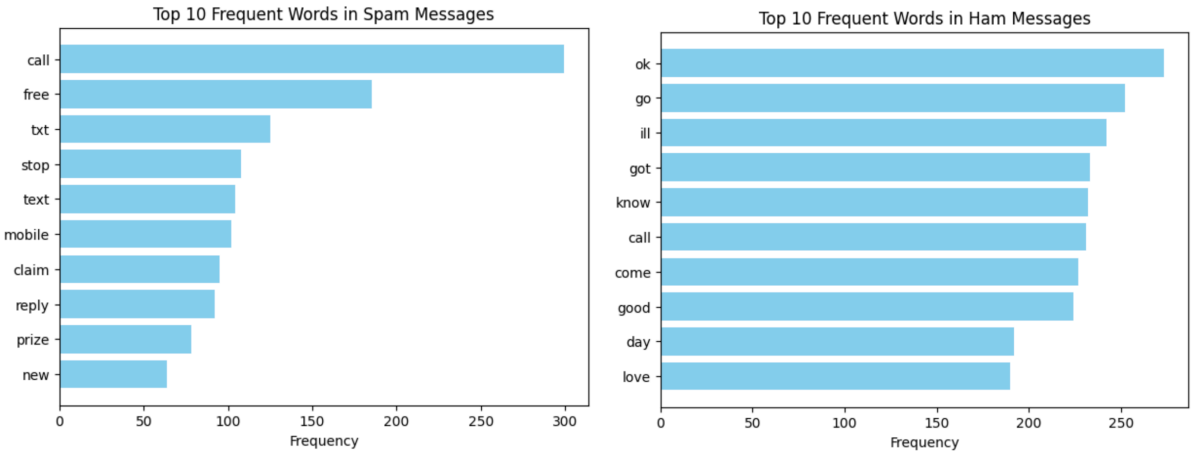


Figure. Top 10 Words in Spam and Ham Messages

The top 10 most frequent words in spam messages bar chart reflect a similar trend as the world of cloud. These words reveal the aggressive marketing and engagement-driven language typical of spam messages, including persuading people to reply to them. The top 10 most frequent words in ham messages, as seen in the bar chart, emphasize the conversational nature of these messages. The words "ok," "go," "good," "day," and "love" suggest that ham messages are centered around social interactions, planning, and emotional connections.

4. Initial Model

In the initial phase of our modeling, we trained three different classifiers on the imbalanced dataset before applying any sampling techniques like SMOTE. The models aimed to classify SMS messages into two categories: ham (non-spam) and spam. We used CountVectorizer and TF-IDF Vectorizer for text feature extraction.

Each classifier was applied six times with different parameters. Except for different methods of feature extraction, we also include different N-gram models with a range of (1,1),(1,2) and (1,3).

Models evaluated showing the best performance:

- Logistic Regression (with CountVectorizer)
- Naive Bayes (with CountVectorizer)
- Random Forest (with TF-IDF Vectorizer)

Model evaluation was based on:

- Accuracy: Overall correctness of the model.
- Precision: How many predicted spam messages were actually spam.
- Recall: How many actual spam messages were correctly identified.
- F1-score: Harmonic mean of precision and recall.
- Confusion Matrix: Breakdown of true/false positives and negatives

Classification Report for Logistic Regression with count_tri13:

	precision	recall	f1-score	support
0	0.98	0.99	0.99	1354
1	0.95	0.84	0.89	194
accuracy			0.97	1548
macro avg	0.96	0.91	0.94	1548
weighted avg	0.97	0.97	0.97	1548

Figure. Classification Report of Logistic Regression with Count13

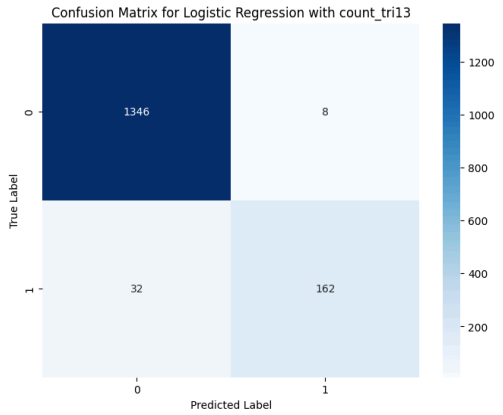


Figure. Confusion Matrix of Logistic Regression with Count13

The best Logistic Regression model was trained using the CountVectorizer with N-gram model (1,3). It achieved an overall accuracy of 97%, with precision of 0.95 and recall of 0.84 for the spam class. The F1-score for spam was 0.89, indicating a reasonable balance between precision and recall. However, the relatively low recall suggests the model missed a notable number of spam messages, misclassifying them as ham. This was confirmed by the confusion matrix, where out of 194 spam messages, 32 were misclassified as ham (false negatives), while 162 were

correctly identified as spam (true positives). Despite its solid accuracy, the Logistic Regression model's performance was hindered by the class imbalance, leading to suboptimal spam detection, which is particularly concerning in real-world applications where missing spam can lead to user dissatisfaction or security risks.

Classification Report for Naive Bayes with count_bi12:

	precision	recall	f1-score	support
0	0.98	0.99	0.99	1354
1	0.95	0.88	0.91	194
accuracy			0.98	1548
macro avg	0.97	0.93	0.95	1548
weighted avg	0.98	0.98	0.98	1548

Figure. Classification Report of Naive Bayes with Count12

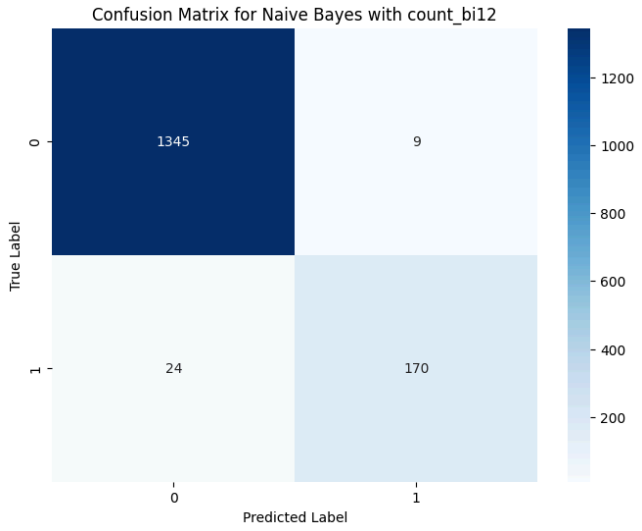


Figure. Confusion Matrix of Naive Bayes with Count12

The best Naive Bayes model, using the CountVectorizer with N-gram model with a range of (1,2), showed superior performance compared to Logistic Regression, achieving an accuracy of 98%. It maintained high precision (0.95) and improved recall (0.88) for spam, with an F1-score of 0.91, reflecting better balance and overall effectiveness in spam classification. The confusion matrix revealed that out of 194 spam messages, only 24 were misclassified as ham (false negatives), while 170 were correctly classified as spam. This model demonstrated a lower false negative rate than Logistic Regression, making it more reliable for detecting spam. The probabilistic nature of Naive Bayes makes it well-suited for text classification tasks, especially

when feature independence assumptions hold reasonably well, as they do in many natural language processing applications.

Classification Report for Random Forest with tfidfvec:

	precision	recall	f1-score	support
0	0.97	0.99	0.98	1354
1	0.96	0.80	0.87	194
accuracy			0.97	1548
macro avg	0.96	0.90	0.93	1548
weighted avg	0.97	0.97	0.97	1548

Figure. Classification Report of Random Forest with Tfidfvec

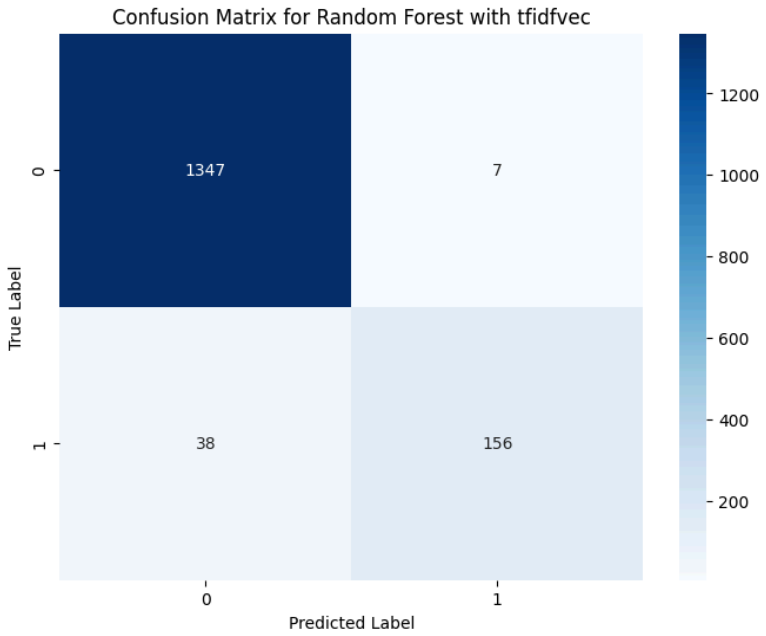


Figure. Confusion Matrix of Random Forest with Tfidfvec

5. Improved Model

5.1 Why SMOTE

In the pie chart, our dataset includes 87.6% of normal(ham) messages and 12.4% of spam messages, indicating that it is an imbalanced dataset, which means the spam class is underrepresented, leading to bias in the process of applying machine learning models—favoring the majority class (ham) while potentially misclassifying spam.

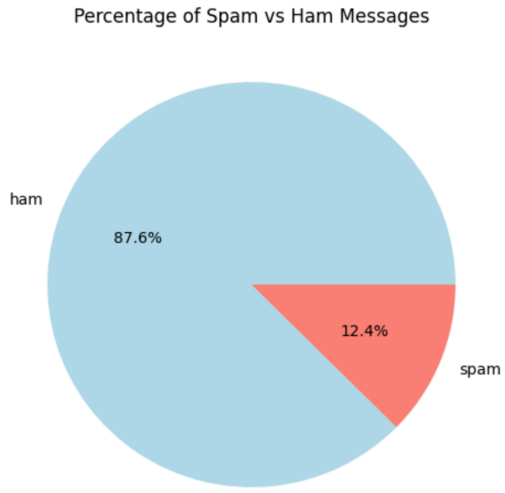


Figure. Pie Chart of "spam" and "ham"

The issues shown as below are caused by imbalanced datasets that happened during the initial analysis.

- High overall accuracy and poor spam detection, which means high precision and low recall for "spam" class.
- The classifier will predict "ham" most of the time and still achieve high accuracy.

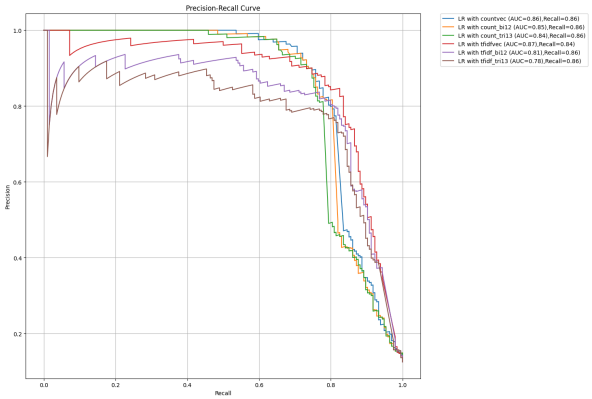
In this case, one effective way to improve this phenomenon is to use resampling techniques. SMOTE(Synthetic Minority Over-sampling Technique) is a widely used technique for handling class imbalance in classification tasks. It is an oversampling method that generates synthetic samples for the minority class(spam), helping machine learning models learn better decision boundaries and improve overall performance.

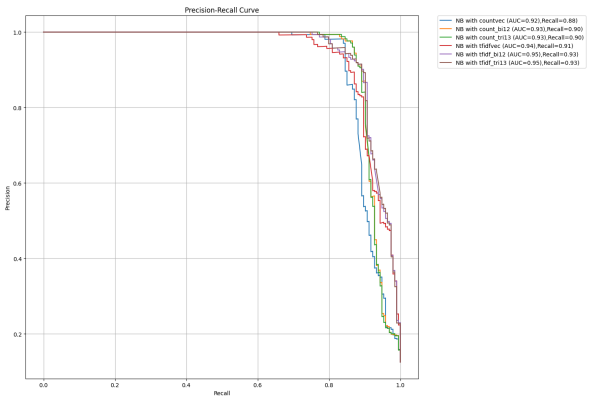
5.2 Analysis with SMOTE

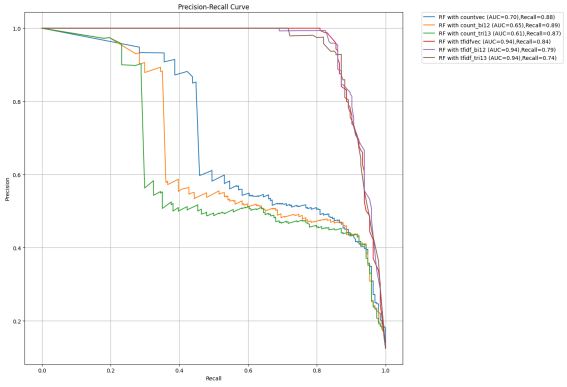
During six different trials using Count-vectorizer and Tfidf-vectorizer along with N-gram Model with the range of (1,1),(1,2) and (1,3) for each classifier, and below we will simplify the name as Countvec, Count12,Count13, Tfidfvec, Tfidf12,Tfidf13. Each model has a different best model by choosing higher recall and AUC.

Firstly, for logistic regression, the best model is the one using Countvec with spam recall 0.86 and overall f1 score 0.86. From the PR curve, though the model using Tfidfvec has the highest AUC, the recall is lower than the model using Countvec. Secondly, for naive bayes, the model

with best performance is the one using Tfidf13 with spam recall 0.93, overall f1 score 0.93 and AUC 0.95. Last, for the random forest, the model applying Tfidfvec performs best with spam recall 0.84, overall f1 score 0.98 and AUC 0.94.







Classification Report for Logistic Regression with countvec:

	precision	recall	f1-score	support
0	0.98	0.84	0.90	1354
1	0.43	0.86	0.57	194
accuracy			0.84	1548
macro avg	0.70	0.85	0.74	1548
weighted avg	0.91	0.84	0.86	1548

Classification Report for Naive Bayes with tfidf_tri13:

	precision	recall	f1-score	support
0	0.99	0.92	0.95	1354
1	0.63	0.93	0.75	194
accuracy			0.92	1548
macro avg	0.81	0.93	0.85	1548
weighted avg	0.94	0.92	0.93	1548

Classification Report for Random Forest with tfidfvec:

	precision	recall	f1-score	support
0	0.98	1.00	0.99	1354
1	0.98	0.84	0.90	194
accuracy			0.98	1548
macro avg	0.98	0.92	0.95	1548
weighted avg	0.98	0.98	0.98	1548

Figure. PR Curve and Classification Report for each model

Finally, by comparing all of the best models in each classification method, the results show that the final model that performed best is **Naive Bayes model using Tfidf13**. Within 194 spam messages, we only misclassify 13 messages as ham messages, showing high recall in this model. It is more evident from the PR curve graph shown below. The red highlighted line indicates that this model has the best AUC and highest recall, indicating that Naive Bayes using the N-gram model with a range of (1,3) performs best.

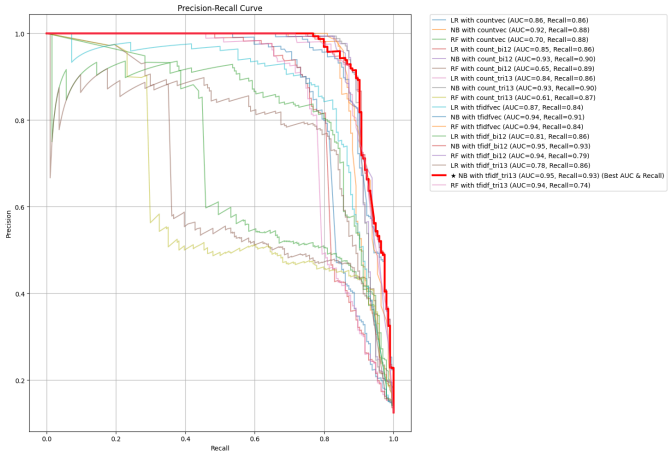


Figure. Overall PR Curve

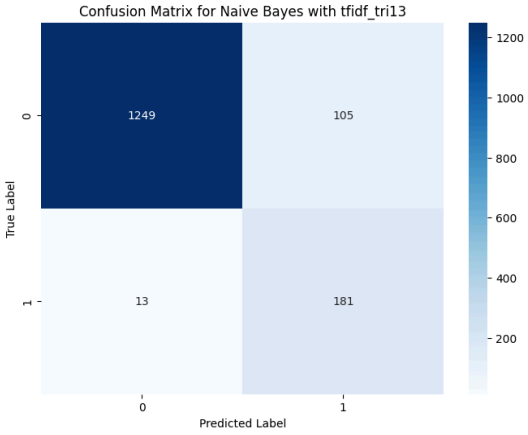


Figure. Confusion Matrix for Naive Bayes(Tfidf13)

For further enhancement and improvement, since combining SMOTE with TF-IDF may cause noise problems, we will refine this model by exploring more machine learning methods.

6. Live Demo

```
===== Spam Message Classifier =====
Type 'quit' or 'exit' to end the program.

Enter a message to classify: I'm on my way
Message: I'm on my way
Classification: not spam
Confidence: 92.39%

Enter a message to classify: Congratulations! 🎉 You have won a FREE iPhone 15! Click the link below to claim your prize now. Hurry, this offer expires in 24
Message: Congratulations! 🎉 You have won a FREE iPhone 15! Click the link below to claim your prize now. Hurry, this offer expires in 24 hours!
Classification: spam
Confidence: 98.33%

Enter a message to classify: RGENT! 🚨 Your account has been compromised. Verify your identity immediately to prevent suspension. Click here:
Message: RGENT! 🚨 Your account has been compromised. Verify your identity immediately to prevent suspension. Click here:
Classification: spam
Confidence: 89.31%

Enter a message to classify:
```

Figure. Live Demo of Message Classifier

This is the live demo which could help classify spam messages. After we enter the message in the chat bot, it will tell us if this message is spam or not spam. Besides, there is a confidence level below which reflects how certain the model is about its classification decision. We tested the spam classifier by inputting several examples from the images. The model successfully identified a casual message as not spam with high confidence. It also correctly classified a promotional message about winning an iPhone and a phishing alert about a compromised account as spam, demonstrating its ability to detect spam messages.

7.Conclusion

In this study, we successfully developed a spam SMS classification model using Natural Language Processing (NLP) and machine learning techniques. Our approach focused on text preprocessing, feature extraction, and classification model evaluation to effectively distinguish spam messages from legitimate ones.

Among the models tested, Naïve Bayes with TF-IDF (unigram + bigram + trigram) demonstrated the best performance, achieving the highest AUC and recall. The recall metric was particularly important in this context, as missing a spam message (false negative) poses a greater risk than misclassifying a legitimate message as spam (false positive). To address the inherent class imbalance in the dataset, we applied SMOTE (Synthetic Minority Oversampling Technique), which significantly improved the model's ability to detect spam messages while maintaining robust classification performance.

Additionally, feature engineering played a crucial role in enhancing model accuracy. The use of n-grams enabled the identification of spam-related phrases beyond individual words, further improving classification effectiveness. Our evaluation metrics, including the confusion matrix and precision-recall (PR) curve, confirmed the reliability of our selected model for real-world application.

While the results demonstrate the effectiveness of machine learning-based spam detection, further refinements could enhance the model's robustness. Expanding the dataset to include more diverse and recent spam messages would improve generalization and adaptability to new spam patterns. Additionally, optimizing hyperparameters and feature selection could further enhance classification accuracy.

Overall, this project highlights the potential of machine learning-driven spam detection in ensuring a safer communication environment. By continuously refining the model and adapting to emerging spam trends, this approach could be effectively deployed in real-world spam filtering systems.