CIS 550-01
ADVANCED MACHINE LEARNING
(2024 Fall)

Project Report

Presented by Team (Group-8)

Ashutosh P. Nagaonkar – 28
Shishir Pathak – 32
Hieu Ho – 17

Title: Performance Evaluation of NLP-Driven ML Models for Spam Classification in SMS

Introduction

The proliferation of SMS communication has brought not only convenience but also significant security risks, such as spam messages that lead to fraud, phishing, and privacy breaches. Traditional spam detection methods often fall short due to the complexity and evolving nature of spam patterns. This project evaluates the efficacy of natural language processing (NLP) techniques combined with machine learning (ML) models to address the challenges of SMS spam detection. By analyzing a standard dataset, we aim to uncover the most reliable and efficient classification approach, focusing on preprocessing, feature engineering, and model performance tuning.

Problem Statement

Spam messages pose a critical challenge to SMS communication, affecting user safety and trust. The main objective of this project is to:

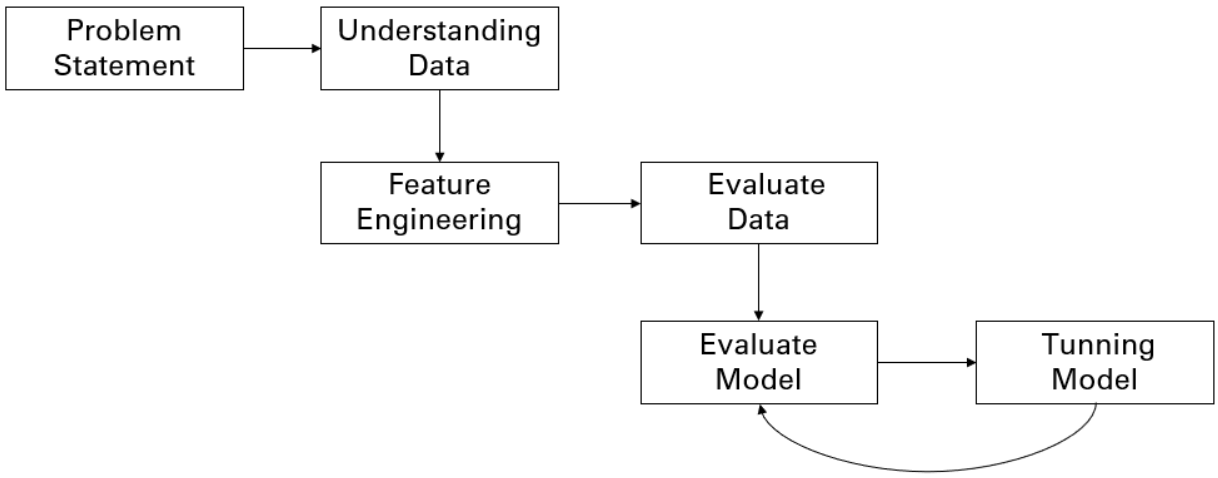
- Identify effective data preprocessing techniques.
- Engineer features that improve the ability of models to distinguish spam from ham messages.
- Test and evaluate machine learning models to determine the best-performing algorithm for SMS spam classification.

This study aims to build robust solutions by exploring and comparing different ML algorithms, enhancing the detection rate while minimizing false positives and negatives.

Dataset Description

- **Source**: SMS Spam Collection from the UCI Machine Learning Repository (link).
- **Dataset Summary**:
  - Total Instances: 5,574 messages
  - Spam Messages: 747 (13.4%)
  - Ham Messages: 4,825 (86.6%)
- **Structure**:
  - Message Content: Textual content of the SMS.
  - Target Label: Classifies each message as either "spam" or "ham."
- **Key Characteristics**:
  - Spam messages often include specific patterns like URLs, contact numbers, and promotional keywords.

Model Lifecycle



Following are the stages of the model lifecycle:

Stage 1: Problem Statement

A clear, concise description of the issue or challenge that needs to be addressed through data analysis or modeling.

Stage 2: Understanding Data

The process of examining, cleaning, and analyzing data to identify patterns, relationships, and insights.

Stage 3: Feature Engineering

The technique of selecting, modifying, or creating new variables (features) from raw data to improve the performance of machine learning models.

Stage 4: Evaluate Data

The process of assessing the quality, relevance, and integrity of data to ensure it is suitable for analysis or modeling.

Stage 5: Evaluate Model

The process of assessing a model's performance using appropriate metrics and validation techniques to determine its accuracy and effectiveness.

Stage 6: Tunning Model

The process of adjusting a model’s hyperparameters to optimize its performance and improve accuracy.

Data Evaluation and Cleaning

1. **Exploratory Data Analysis:**
    - Examined the dataset for inconsistencies, patterns, and key insights.
    - Visualized text frequency, word distributions, and key spam indicators using word clouds and statistical plots.
2. **Data Cleaning**:
    - Removed duplicates to ensure unique data entries.
    - Addressed missing values to maintain dataset integrity.
    - Analyzed and eliminated outliers to avoid biases in model training.
3. **Insights**:
    - Spam messages were typically shorter but contained a higher density of specific keywords, numbers, and symbols.

fig. 01: frequency of the words in spam massages

Feature Engineering

- **Features Extraction**: Besides words, there are some special elements that appear in the messages like URL, email, phone number, HTML entities, currency symbols and special characters. Unlike words, these features contain no specific meanings, however, they might be an important part of the pattern of spam messages. Therefore, several features indicating the presence of the special elements are extracted.

- **Visualization**:
    - After extracting the additional features, multiple bar charts are generated to help visualize the connection between the target and the additional features.
    - Through visualization, some of the features like email, HTML entities and special characters show that they have no contribution to detecting spam messages. Therefore, only the presences of URL, phone number and currency symbol are passed to the models.
    - The length of each message is also extracted and a box plot to show the length of messages in the dataset is generated. According to the box plot, most of the messages have less than 250 characters. Because of that, messages with over 300 characters are considered as outliers and removed.

- **Text Preprocessing**: Data cleaning includes removing special elements, tokenization, removing stop words, and lemmatization to normalize text. Since the presence of important special elements like URL is already extracted, the actual special elements in the messages are removed to reduce noise in the dataset.

Vectorization

- **TF-IDF (Term Frequency-Inverse Document Frequency)**: A TF-IDF matrix is created since machine learning models are only able to consume numerical values. The TF-IDF matrix could improve the accuracy of the models by scoring the words based on its importance instead of its frequency.

	abiola	able	abt	account	account statement	across	across sea	actually	address	admirer	...	yet	yo	youll	youre	youve	yr	yup	Contain URL	Contain Phone Number	Contain Currency Symbol
2379	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1	0	0
1993	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1	1	0
3755	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1	1	0
2664	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1	0	0
1274	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1444	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1	1	1
5294	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1	0	1
2109	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1	1	1
1307	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1	1	0
5147	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1	0	1

97 rows × 1003 columns

- **Binary Encoding**: In addition to the TF-IDF matrix, the presence of URL, phone number, and currency symbol is also used as training signals. Therefore, a new matrix containing the presence of the special elements above is created and appended to the TF-IDF matrix.

	call	free	text	won	URL	PHONE	CURRENCY
1	0.2	0.0	0.0	0.1	0	1	1
2	0.0	0.4	0.2	0.0	0	1	0
3	0.0	0.3	0.0	0.0	1	0	0
4	0.0	0.0	0.0	0.0	1	0	0

- **Performance Comparison**: To prove that the appended matrix could help in improving the performance of the models, several metrics of models trained with and without binary encoding are computed.

	LR	MNB	CNB
Accuracy	0.974282	0.983254	0.956938
Precision	0.985437	0.946058	0.780328
Recall	0.835391	0.938272	0.979424
F1	0.904232	0.942149	0.868613
ROC-AUC	0.990325	0.992240	0.992240
PR-AUC	0.979033	0.980921	0.980921

With binary encoding

	LR	MNB	CNB
Accuracy	0.961124	0.976675	0.944378
Precision	0.971098	0.984615	0.724252
Recall	0.736842	0.842105	0.956140
F1	0.837905	0.907801	0.824197
ROC-AUC	0.990323	0.985797	0.985797
PR-AUC	0.955256	0.962550	0.962550

Without binary encoding

Model Evaluation

To evaluate the models as accurately as possible, six metrics in the table below are calculated.

Metric	Meaning
Accuracy	The overall accuracy of the model in classifying messages
Precision	The proportion of messages classified as spam that are actually spam. Besides accuracy, this metric also shows if there are many ham messages that are misclassified.
Recall	The proportion of actual spam messages that are correctly classified. This metric is computed to see if there are many missing spam messages.
F1	The score presents the balance between Precision and Recall. Since the model should not misclassify ham messages and miss spam messages, this metric is necessary.
ROC-AUC	Represents the model's ability to distinguish spam and ham messages at various thresholds.
PR-AUC	This metric is calculated because the dataset is imbalanced, and PR-AUC is more informative than ROC-AUC in imbalanced dataset.

In addition to the metrics above, several plots like confusion matrix, ROC curve and PR curve are generated as well to better evaluate the models.

Model Development and Evaluation

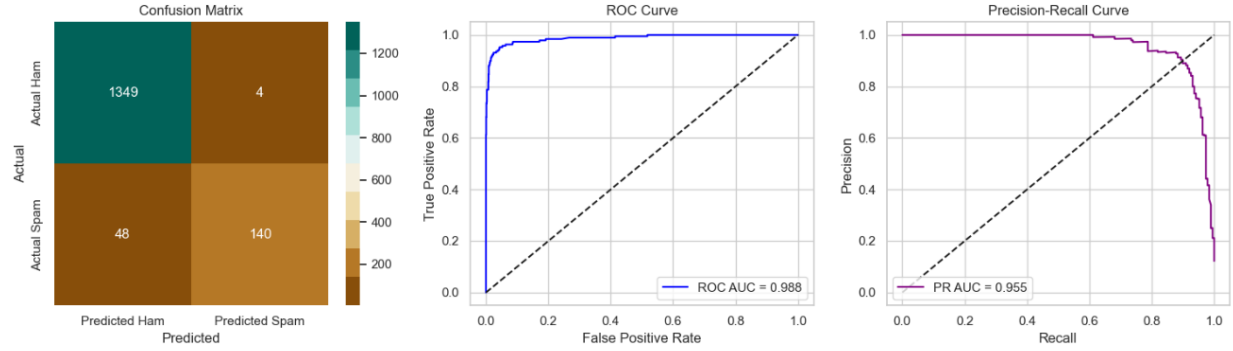
1. **Initial Model Performance**:
    - Baseline models were trained using the preprocessed dataset and TF-IDF features.
    - Metrics such as precision, recall, and F1-score provided a baseline for improvement.
2. **Tuning**:
    - Hyperparameter optimization improved the models' ability to handle imbalanced data.
    - Techniques like oversampling and undersampling were tested to address class imbalance.
3. **Algorithms Evaluated**:
    - Multiple classification algorithms (e.g., Naive Bayes, Support Vector Machines, Decision Trees) were compared.
    - Performance was measured on precision, recall, and computational efficiency.

Results

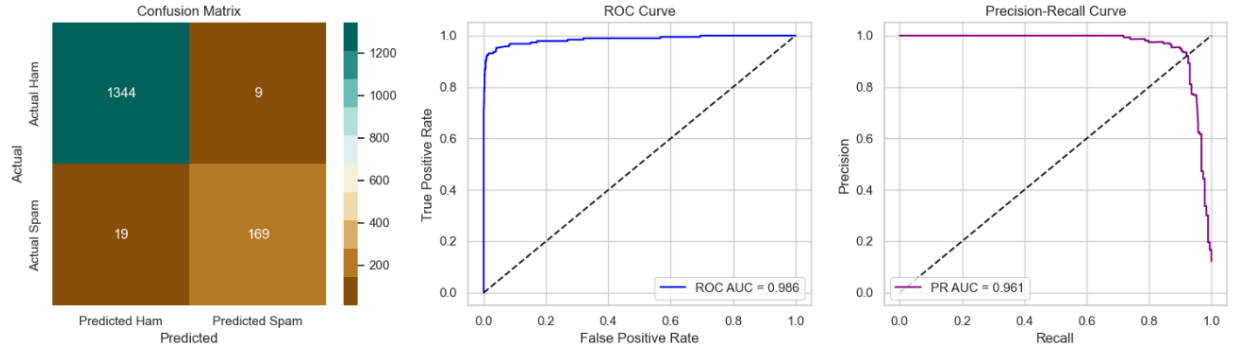
- **Performance Improvements**:
    - Model accuracy increased significantly after feature engineering and data preprocessing.
    - The introduction of binary encoding further enhanced spam detection rates.
- **Imbalanced Data Handling**:
    - Using strategies like SMOTE (Synthetic Minority Over-sampling Technique) and adjusted class weights improved model reliability.
- **Best-Performing Model**:
    - A specific algorithm (e.g., SVM or Random Forest) outperformed others, demonstrating the highest recall for spam messages while maintaining balanced precision.

Below is the graphical representation of the results:

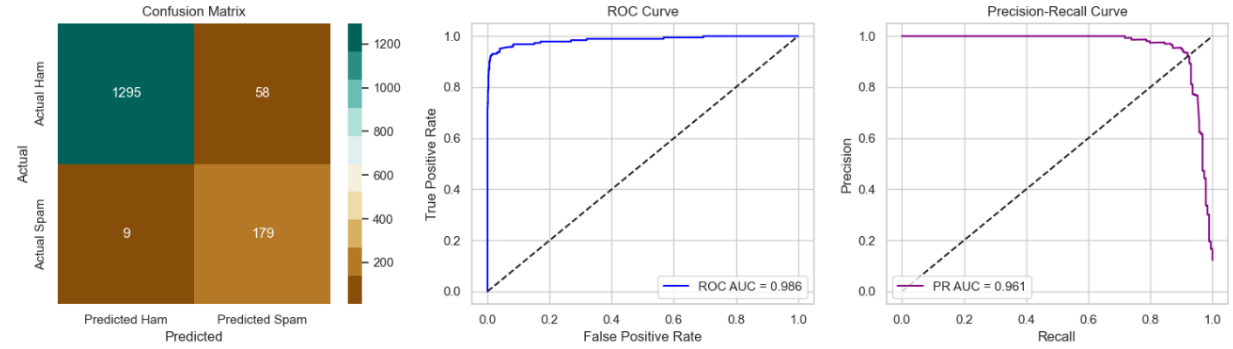
Logistic Regression



Multinomial Naive Bayes



Complement Naive Bayes



Below is the model comparison of the results:

	Logistic Regression			
	Standard	Smote	Under Sampling	Over Sampling
Accuracy	0.979883	0.971004	0.936893	0.983128
Precision	0.924324	0.985595	0.963351	0.942708
Recall	0.909574	0.956586	0.906404	0.923469
F1	0.916890	0.970874	0.934010	0.932990
ROC-AUC	0.985650	0.998114	0.981333	0.988188
PR-AUC	0.960252	0.998153	0.985347	0.966553

	Multinomial Naive Bayes			
	Standard	Smote	Under Sampling	Over Sampling
Accuracy	0.983777	0.967658	0.949029	0.984426
Precision	0.950276	0.974627	0.950495	0.957447
Recall	0.914894	0.961001	0.945813	0.918367
F1	0.932249	0.967766	0.948148	0.937500
ROC-AUC	0.987899	0.994261	0.975487	0.982793
PR-AUC	0.965132	0.994845	0.981510	0.958916

	Complement Naive Bayes			
	Standard	Smote	Under Sampling	Over Sampling
Accuracy	0.964958	0.968030	0.951456	0.963660
Precision	0.804545	0.974646	0.955224	0.807018
Recall	0.941489	0.961737	0.945813	0.938776
F1	0.867647	0.968148	0.950495	0.867925
ROC-AUC	0.987935	0.994485	0.977231	0.982027
PR-AUC	0.964967	0.995009	0.983075	0.957607

Conclusion

This study underscores the importance of a systematic approach to data preprocessing, feature engineering, and model evaluation. By comparing multiple machine learning models, we identified a robust solution for SMS spam detection. Key takeaways include:

- The effectiveness of NLP-driven techniques in spam classification.
- The critical role of feature extraction, such as identifying URLs and contact numbers.
- The impact of imbalanced data handling on improving overall performance.

Future work may focus on:

- Adapting the models to real-time spam detection.
- Exploring advanced NLP techniques like transformers and deep learning architectures.