SMS Spam Classifier

Abstract

In this project, we develop an automated spam detection system to classify SMS messages as **Spam** or **Ham** using Natural Language Processing (NLP) and Machine Learning techniques. The dataset consists of labeled SMS text messages which are preprocessed through standard NLP steps including lowercasing, punctuation removal, tokenization, stopword removal, and lemmatization.

We then convert the cleaned text into numerical features using TF-IDF vectorization. Three machine learning models — **Multinomial Naive Bayes, Logistic Regression, and Support Vector Machine (SVM)** — are trained and evaluated. To address best model we tune the model parameters using **GridSearchCV** to the training data.

Introduction

We are given with SMS and its classifier whether it is spam or ham for 5572 SMS. (5572*2)

Feature Engineering

NLP: Natural Language Processing is a branch of AI that focuses on enabling computers to understand, interpret and generate human languages.

Text Cleaning and Lowercasing: Removes unnecessary elements like punctuation, special characters, and numbers. Lowercasing ensures uniformity so that "Spam" and "spam" are treated the same.

Tokenize: Break a sentence into individual units like words or phrases.

Stopword Removal: Removes common but uninformative words like "the", "is", "and", etc.

Stemming/Lemmatization: Reduces words to their root/base form (e.g., "running" → "run"). This helps group similar words and reduces feature space.

Vectorization: Convert a text into numerical format using TF-IDF that is to specify whether a word is in the text (1) or not (0).

LabelEncoder: Transforms categorical text labels (e.g., "spam", "ham") into numerical values (e.g., 1, 0). Essential for training models on classification tasks.

Data Splitting

Split the data into 80% training and 20% testing part.

Data Tuning

GridSeachCV: GridSearchCV is a technique in machine learning used to find the best combination of hyperparameters for a model. It performs an exhaustive search over a specified grid of parameter values using cross-validation.

We define grid of possible values for each parameter. It tries every combination of parameters from grid and uses cross-validation to evaluate model performance. Then it returns best combination of parameters based on chosen metics.

Modelling

Naïve Bayes: Naive Bayes is a powerful probabilistic classification algorithm based on Bayes' Theorem. It assumes that the features (e.g., words in a message) are independent of each other, which is why it's called "naive".

In a spam classifier, Naive Bayes calculates the probability that a given message is spam or ham, based on the presence of specific words. It uses word frequencies learned from training data to make predictions. We typically use the **Multinomial Naive Bayes** variant for text classification, as it is well-suited for discrete features like word counts or TF-IDF scores. Due to its fast training, low resource usage, and strong performance on high-dimensional text data, Naive Bayes is often a preferred baseline for spam detection tasks.

Logistic Regression: Logistic Regression is a statistical model used for binary classification, such as determining whether a message is spam (1) or ham (0). It models the probability that a given input belongs to a particular class using the logistic (sigmoid) function.

In spam detection, logistic regression assigns weights to each word (feature), and then calculates a probability score. If the score exceeds a certain threshold (usually 0.5), the message is classified as spam.

Support Vector Machine(SVM): Support Vector Machine (SVM) is a powerful supervised learning algorithm used for classification tasks. It aims to find the optimal hyperplane that best separates the classes in the feature space — in this case, spam and ham messages.SVM is especially effective in high-dimensional spaces (like text data after vectorization), and is known for its robustness in handling imbalanced or sparse datasets.

For text classification tasks like spam detection, we commonly use **Linear SVM**, which is computationally efficient and performs well when the data can be linearly separated.

Result

Model	Accuracy	Precision	Recall	F1_Score	AUC
Naive Bayes	0.9766	0.9428	0.8800	0.9103	0.9358
Logistic Regression	0.9730	0.9687	0.8266	0.8920	0.9112
SVM	0.9748	0.9692	0.8400	0.9000	0.9179

Recall detects as much spam as possible and F1 score tells us how well the model handles both kind of errors. Here, both the Recall and F1-Score of Naïve Bayes is much higher than that of other models. Also, the other metrics perform better.

Conclusion

In this project, Naive Bayes outperformed Logistic Regression and SVM, likely because it naturally fits the text classification problem structure, assumes feature independence, handles high-dimensional sparse data efficiently, and performs well even with simplistic feature assumptions. Also, Naive Bayes is simple and doesn't overfit easily, especially on small/medium datasets like SMS spam. Its probabilistic nature makes it more confident in identifying spam-like messages based on word distributions.

For future work this model can be further improved by more diverse spam dataset, fine tuning hyperparameters and developing more effective algorithms.

Shreya Saha