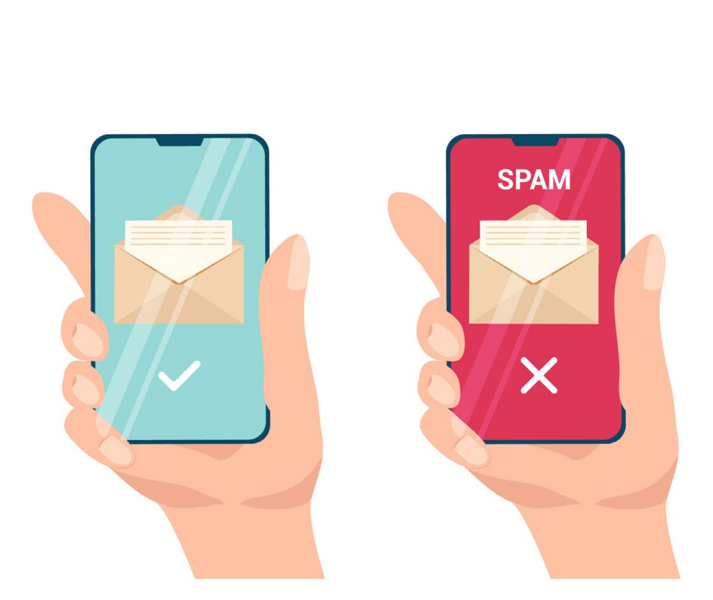
SMS Spam Detection Using Machine Learning



Presented By

Monalisa

Executive Summary

This report presents a machine learning-based approach to detecting spam emails by analyzing their content and structure. It outlines the use of text preprocessing, feature extraction (like TF-IDF), and classification algorithms such as Naive Bayes , Logistic Regressor and Random Forest to distinguish between spam and ham messages. The system improves email security by filtering harmful or irrelevant content, helping users maintain a clean and trustworthy inbox.

Problem Statement

The Challenge

Detecting spam emails is difficult because spammers keep changing their strategies to bypass filters. The report points out challenges like imbalanced data and deceptive content, which make it tough for models to stay accurate without regular updates and fine-tuning.

Project Objective

The objective of this project is to develop a system that can automatically detect spam emails using machine learning techniques. By analyzing patterns in email content, the report aims to improve inbox safety and reduce exposure to scams and unwanted messages.

Data Overview

The dataset used in this project is the **SMS Spam Collection Dataset** which is publicly available on **Kaggle.**

Relevant Columns used for this project:

- label(ham or spam)
- message

	label	message
685	ham	Have you finished work yet? :)
4728	ham	I've reached already.
1385	ham	That's ok. I popped in to ask bout something a...
3166	ham	When people see my msgs, They think Iam addict...
653	ham	Fine i miss you very much.

Feature Engineering

To enhance the predictive power of the spam classification model, several preprocessing and feature engineering applied to the raw dataset.

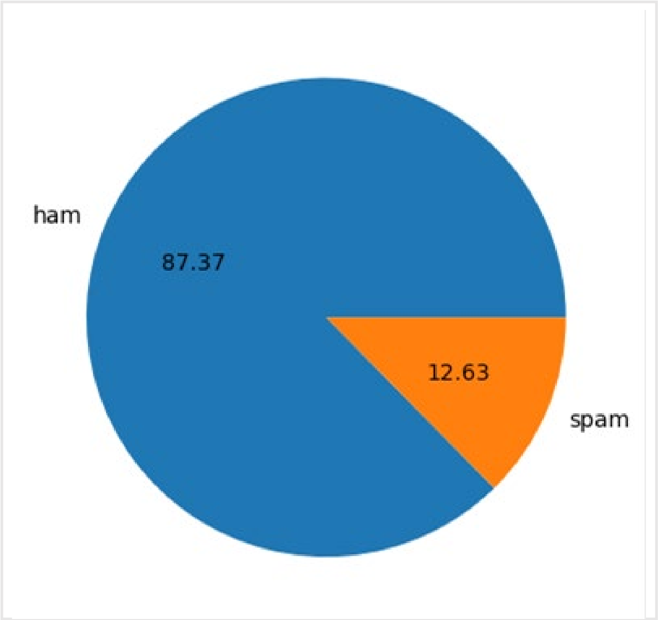
- The categorical label encoded numerically assigning 0 to ham and 1 to spam

- Additional features:
  - num_characters
  - num_words
  - num_sentences
  - transformed_message

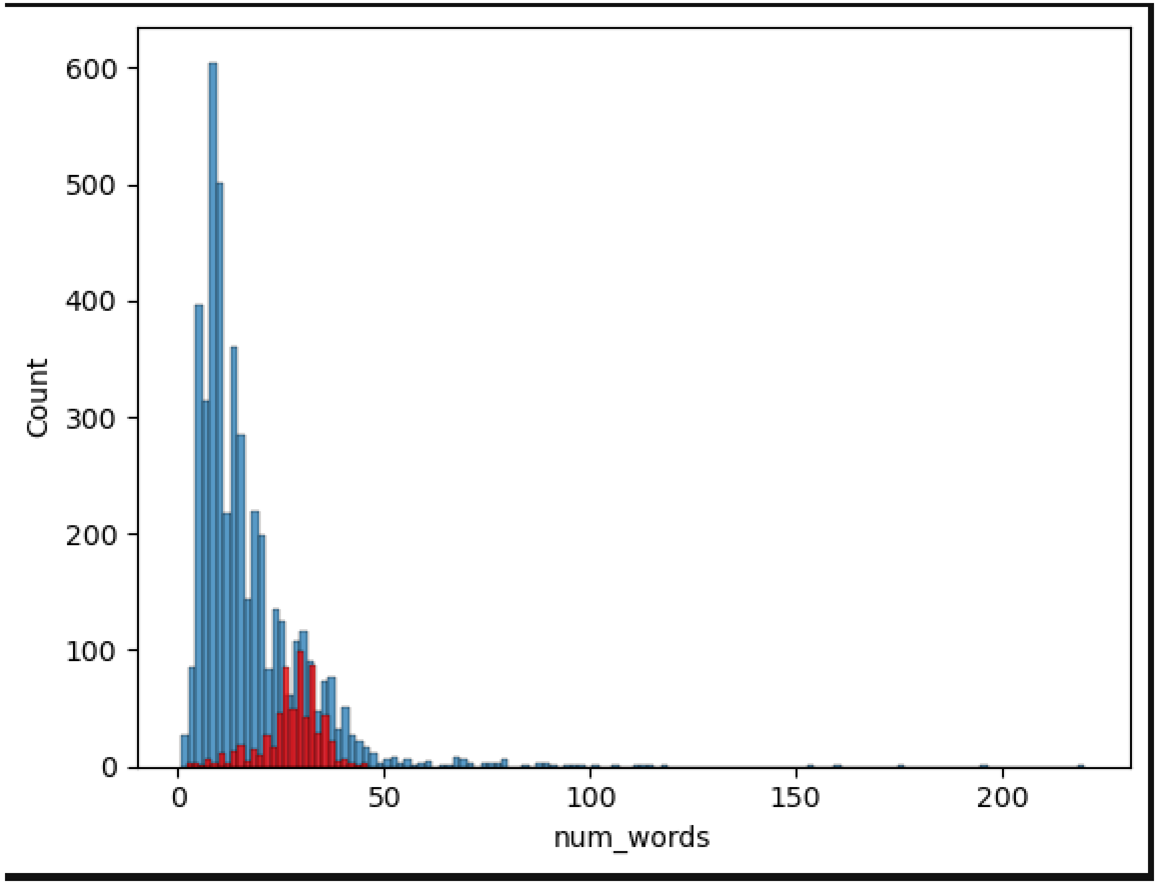
	label	message	num_characters	num_words	num_sentences	transformed_message
0	0	Go until jurong point, crazy.. Available only ...	111	24	2	go jurong point crazi avail bugi n great world...
1	0	Ok lar... Joking wif u oni...	29	8	2	ok lar joke wif u oni
2	1	Free entry in 2 a wkly comp to win FA Cup fina...	155	37	2	free entri 2 wkli comp win fa cup final tkt 21...
3	0	U dun say so early hor... U c already then say...	49	13	1	u dun say earli hor u c alreadi say
4	0	Nah I don't think he goes to usf, he lives aro...	61	15	1	nah think goe usf live around though

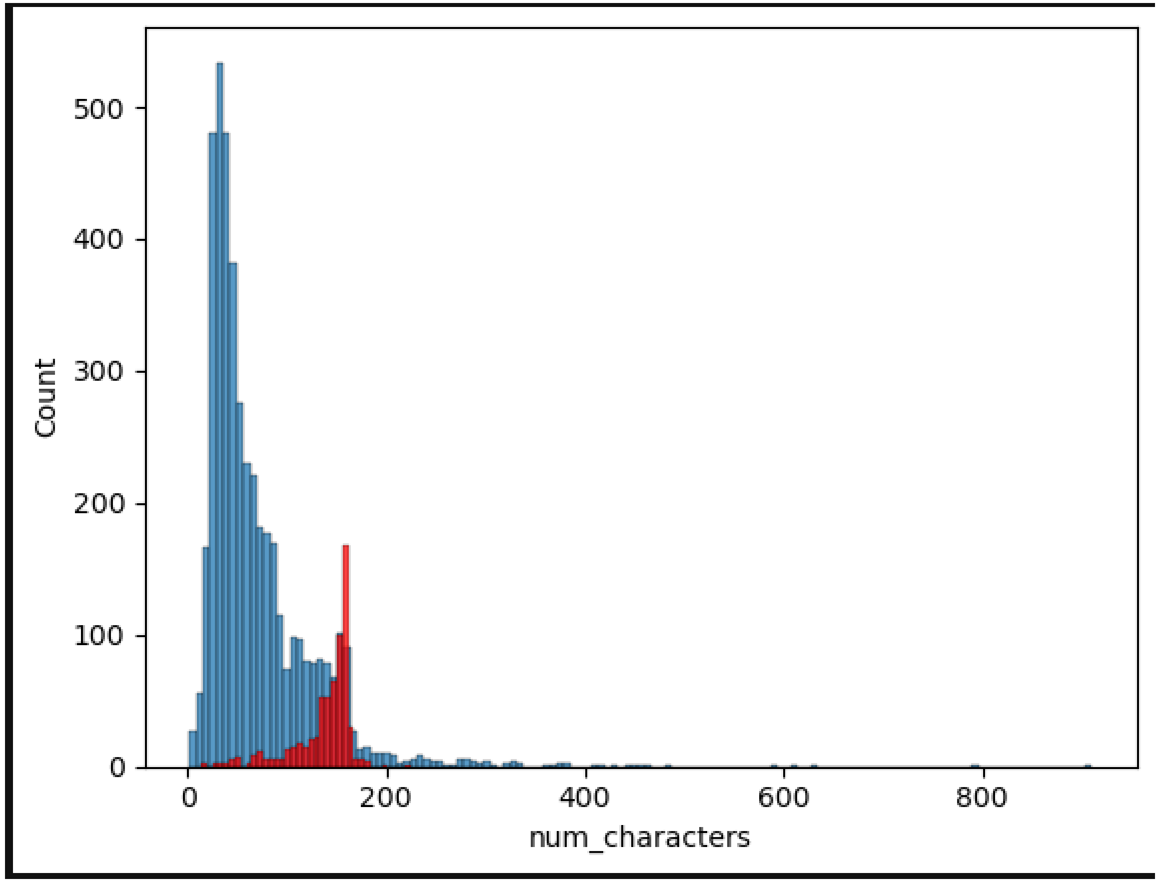
Spam vs Ham Pattern Insights

- In this dataset distribution of ham and spam messages where ham accounts for 87.37% and spam makes up 12.63%.
- The dataset is imbalanced with a much larger proportion of ham messages.

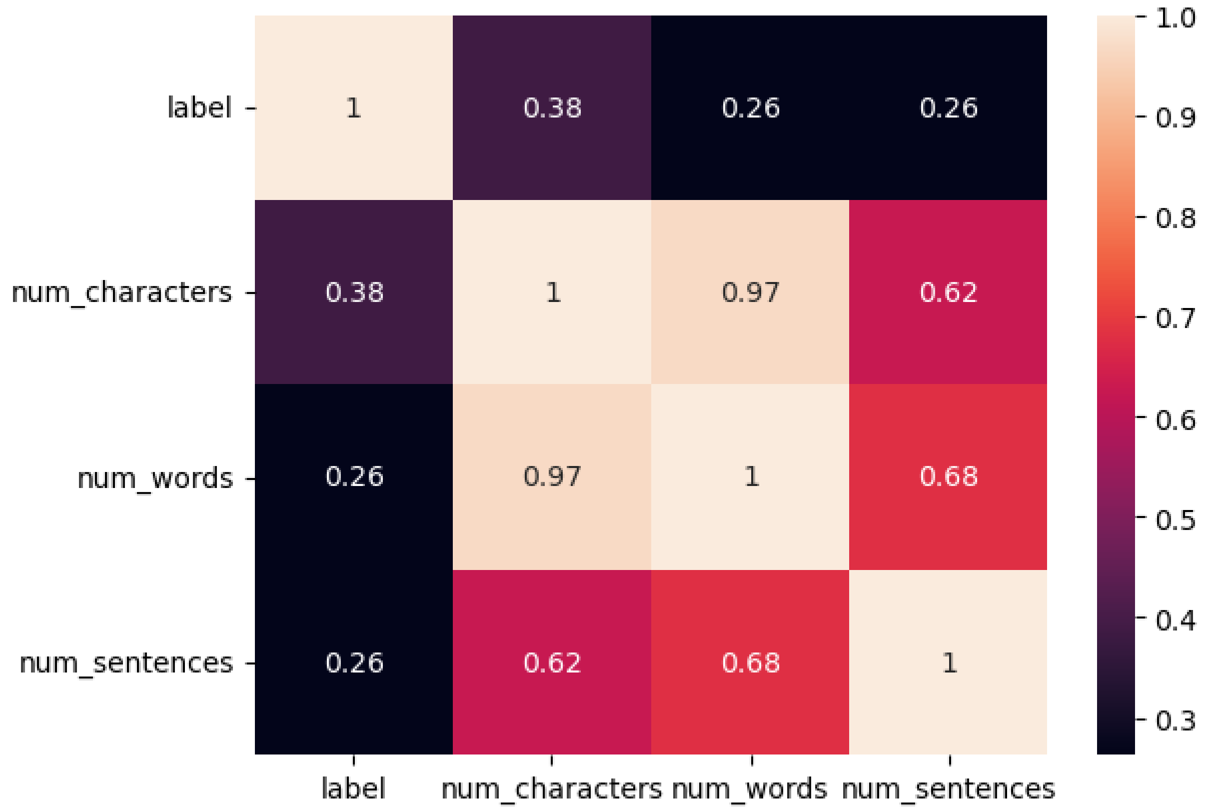


- Majority of ham messages number of characters and words are less than as compare to spam messages .

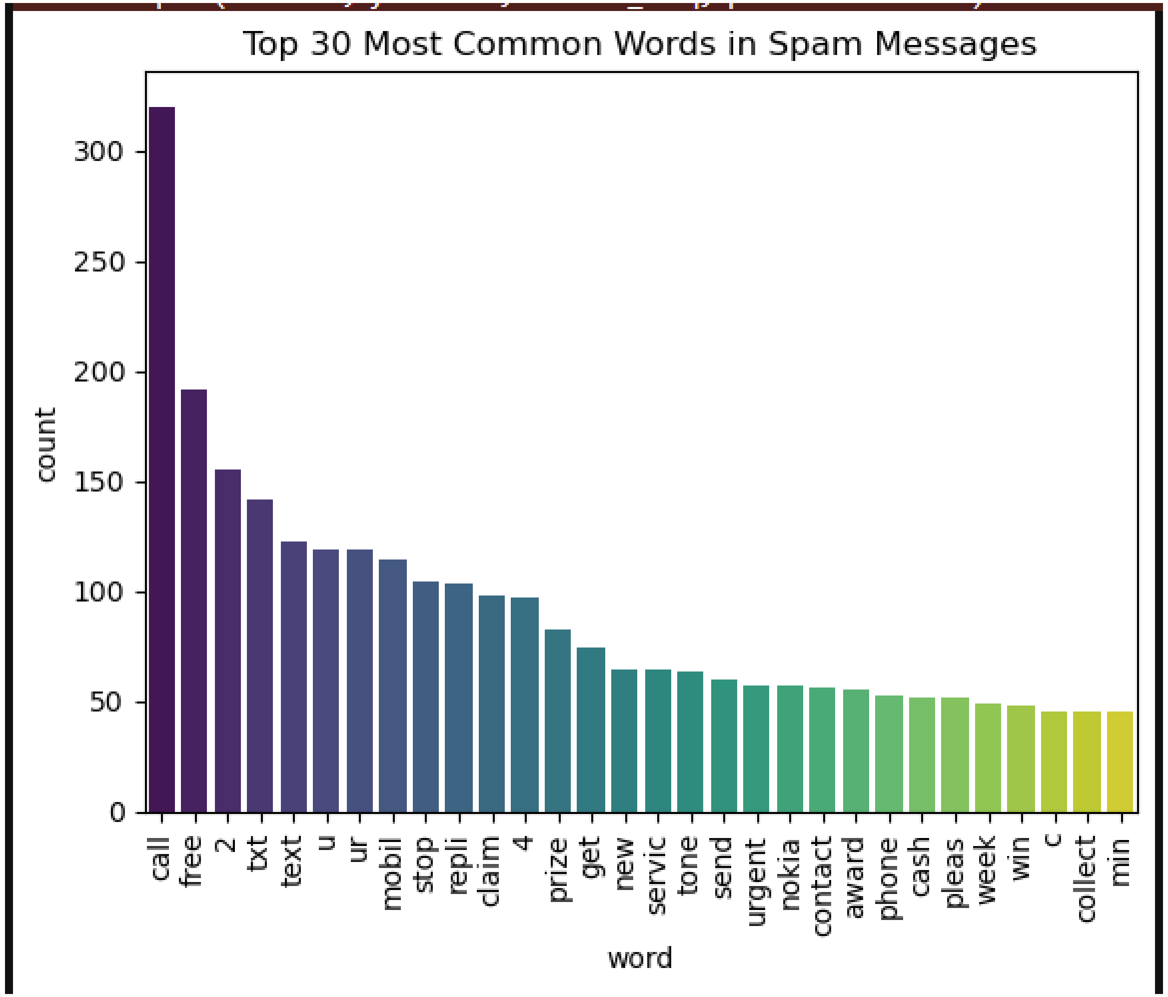


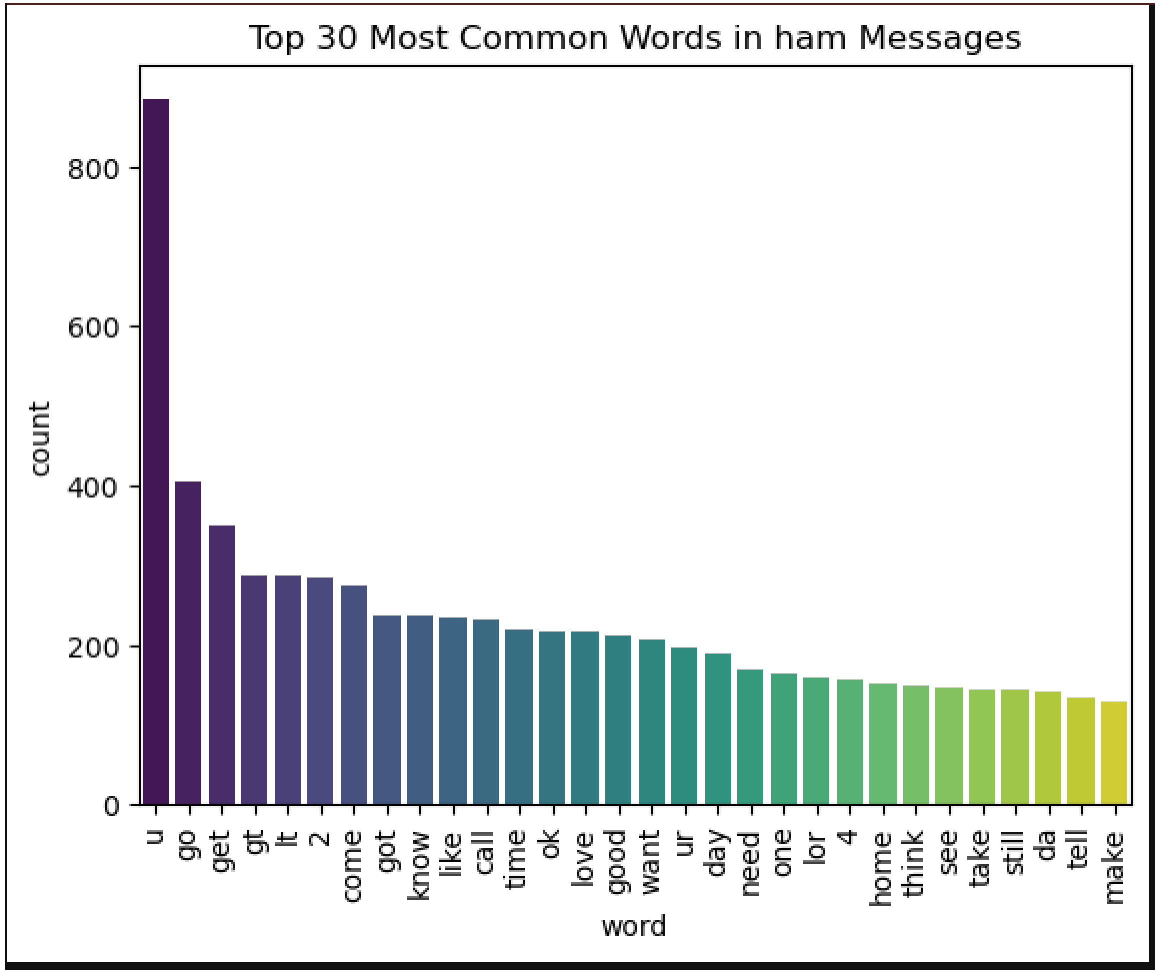


- Correlation of num_character with label is higher than num_words and num_sentences
- Higher correlation among num_character , num_words and num_sentence



- In spam messages most used words like call , free, text mobile so on
- In ham messages most used words like u, go, get so on





Feature Extraction & Model Training

- Feature extraction
  - Bag of Words (CountVectorizer)
  - TF-IDF (TfidfVectorizer)

- For model training and testing taken 70% for training and 30% for testing

- Model used Naive Bayes , Logistic Regressor , Random Forest

- In this projects Naïve Bayes classifier are used GaussianNB, MultinomialNB, BernoulliNB

Model Matrix Performance in different feature extraction

Bag of Words (CountVectorizer)

```
GaussianNB
accuracy_score: 0.8794326241134752
confusion_matrix:
[[1205  155]
 [  32  159]]
precision_score: 0.5063694267515924
```

```
MultinomialNB
accuracy_score: 0.9677627337201805
confusion_matrix:
[[1330   30]
 [  20  171]]
precision_score: 0.8507462686567164
```

```
BernoulliNB
accuracy_score: 0.9696969696969697
confusion_matrix:
[[1356    4]
 [  43  148]]
precision_score: 0.9736842105263158
```

TF-IDF (TfidfVectorizer)

```
GaussianNB
accuracy_score: 0.8723404255319149
confusion_matrix:
[[1202  158]
 [  40  151]]
precision_score: 0.4886731391585761
```

```
MultinomialNB
accuracy_score: 0.9716312056737588
confusion_matrix:
[[1360    0]
 [  44  147]]
precision_score: 1.0
```

```
BernoulliNB
accuracy_score: 0.9819471308833011
confusion_matrix:
[[1358    2]
 [  26  165]]
precision_score: 0.9880239520958084
```

Performance Analysis of ML models

	Algorithm	Accuracy	Precision	Accuracy_max_ft_3000	Precision_max_ft_3000
0	NB	0.955513	1.000000	0.971631	1.000000
1	RF	0.972276	0.993333	0.973565	0.974684
2	LR	0.949065	0.924242	0.952289	0.933333

- **Naive Bayes (NB)** achieved perfect precision, meaning it didn't misclassify any ham as spam but its overall accuracy was slightly lower than RF.
- **Random Forest (RF)** delivered the highest accuracy and strong precision, making it the most balanced and robust performer.
- **Logistic Regression (LR)** had the lowest scores across both metrics, indicating it may be less effective for this dataset without further tuning.

Final Result Summary

After checking all the results, Naive Bayes got a perfect precision score, which means it caught every spam email without wrongly marking any normal emails. That makes it the best choice if we want to avoid blocking real messages by mistake.

Thank You!