SMS Spam Detection Using Machine Learning

1. Title Page

Project Title: SMS Spam Detection Using Machine Learning
Intern Name: Keerthana D
Institution / Department: Pallavan college of Engineering/ B.E. CSE
Date: 21/10/2025
Reg No: 511723104006

Github link: Keerthanadhanasekaran/SMS-spam-detection-ml: Oracle Internship Project — SMS Spam Detection using Machine Learning

2. Table of Contents

3. Project Overview

Spam detection in SMS messages is a critical task in modern communication. The project aims to build a machine learning model
that automatically classifies messages as 'spam' or 'ham' (not spam). This improves user experience and prevents fraudulent activities.

Scope: The project focuses only on SMS text classification into two categories: spam and ham.
The solution includes preprocessing, feature extraction, model training, evaluation, and prediction.

Dataset: SMS Spam Collection Dataset (UCI Repository)

4. Objectives & Problem Statement

Problem Statement

The main goal is to classify a given SMS message as spam or ham. The challenge lies in dealing with short, informal, and diverse
text patterns, including slang, typos, and mixed-language expressions.

Objectives

- Build an accurate predictive model for SMS spam detection.
- Compare performance across multiple machine learning algorithms.
- Implement text preprocessing and feature extraction using TF-IDF.
- Evaluate model performance with standard metrics (Accuracy, Precision, Recall, F1-score).
- Visualize and analyze results to identify the best model.

5. Proposed Solution

Approach: Supervised machine learning using text classification.
Pipeline: Raw SMS → Preprocessing → TF-IDF Vectorization → Model Training → Evaluation → Prediction.
Algorithms Used: Naive Bayes, Logistic Regression, Random Forest, and Support Vector Machine (SVM).
Justification: These algorithms are effective for text classification tasks and balance accuracy with computational efficiency.

6. Features

Functional Features

- Input SMS text and output classification result ('spam' or 'ham').
- Display model performance metrics and confusion matrix.
- Support batch message classification.

Non-Functional Features

- High accuracy and low latency.
- User-friendly implementation in Google Colab.

- Scalable and maintainable codebase with reusable models.

7. Technologies & Tools

- Language: Python
- Libraries: pandas, numpy, scikit-learn, nltk, seaborn, matplotlib, joblib
- Environment: Google Colab, GitHub for version control
- Dataset Source: UCI Machine Learning Repository

8. System Architecture

Architecture Flow:
Input SMS → Text Preprocessing → TF-IDF Vectorization → Model Training → Prediction & Evaluation

Components:
1. Input Layer: Accepts SMS messages.
2. Preprocessing Layer: Cleans and normalizes text.
3. Feature Extraction: Converts text to TF-IDF features.
4. Model Layer: Trains multiple ML algorithms.
5. Evaluation: Measures model accuracy, precision, recall, and F1-score.
6. Output: Displays prediction results and performance visuals.

9. Implementation Steps

1. Load dataset from online source.
2. Perform exploratory data analysis (EDA) to understand message distribution.
3. Preprocess text data (cleaning, tokenization, stopword removal, stemming).
4. Convert text to numerical features using TF-IDF Vectorizer.
5. Split dataset into training and testing sets.
6. Train models: Naive Bayes, Logistic Regression, Random Forest, SVM.
7. Evaluate models and visualize results.
8. Identify the best-performing model.
9. Save model and vectorizer using joblib.

10. Test the system on new SMS samples.

10. Output / Screenshots

The system successfully classifies SMS messages as spam or ham.

- Dataset preview

✅ Dataset Loaded Successfully!

	label	message
0	ham	Go until jurong point, crazy.. Available only ...
1	ham	Ok lar... Joking wif u oni...
2	spam	Free entry in 2 a wkly comp to win FA Cup fina...
3	ham	U dun say so early hor... U c already then say...
4	ham	Nah I don't think he goes to usf, he lives aro...

Figure 1: Sample of SMS Spam Dataset

- Class distribution chart

```
Dataset info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5572 entries, 0 to 5571
Data columns (total 2 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   label    5572 non-null   object
 1   message  5572 non-null   object
dtypes: object(2)
memory usage: 87.2+ KB
None

Class Distribution:
label
ham     4825
spam     747
Name: count, dtype: int64
/tmp/ipython-input-429945280.py:9: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.countplot(x='label', data=df, palette='viridis')
```

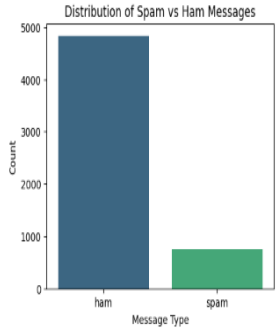


Figure 2: Distribution of Spam vs Ham Messages

- Model performance comparison bar graph

Model Comparison:

	Accuracy	Precision	Recall	F1
Naive Bayes	0.981166	1.000000	0.859060	0.924188
Logistic Regression	0.962332	0.990826	0.724832	0.837209
Random Forest	0.981166	1.000000	0.859060	0.924188
SVM	0.985650	0.985401	0.906040	0.944056

```
/tmp/ipython-input-2983375317.py:7: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=results_df.index, y=results_df['Accuracy'], palette='mako')
```

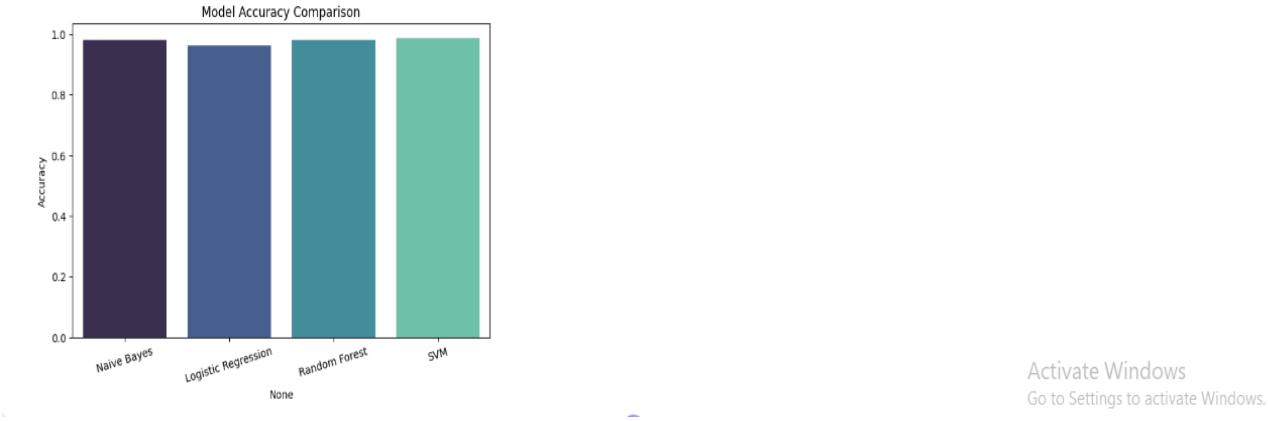
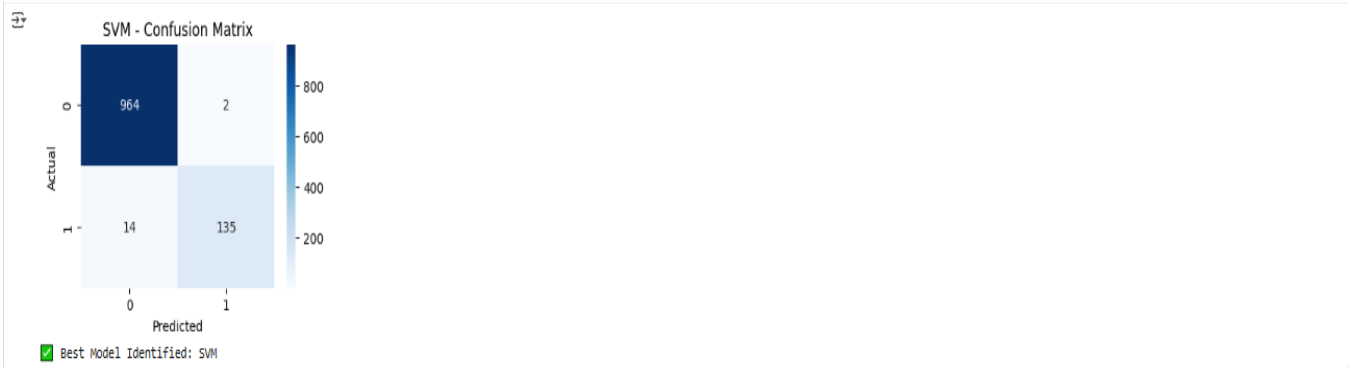


Figure 3: Model Accuracy Comparison

- Confusion matrix heatmap



-

Figure 4: Confusion Matrix of Best Model (SVM)

-Sample message predictions

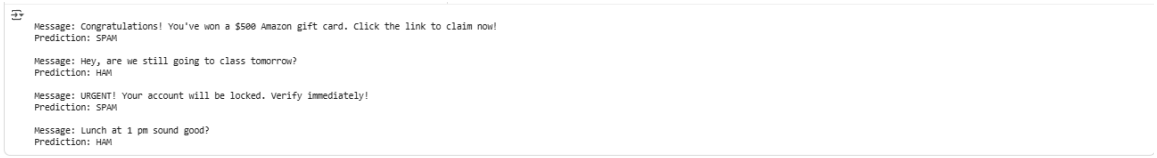


Figure 5: Example Predictions for New SMS Messages

11. Advantages

- Automates SMS spam detection efficiently.
- Achieves high precision and recall.
- Easy to extend with new data or algorithms.
- Visualization improves interpretability.
- Works seamlessly in cloud environments like Google Colab.

12. Future Enhancements

- Use deep learning models like LSTM or BERT for improved accuracy.
- Support multilingual spam detection.
- Develop a web or mobile interface using Flask or Streamlit.
- Implement real-time detection and alert system.
- Deploy as a REST API service on cloud platforms.

13. Conclusion

The project successfully developed and evaluated multiple machine learning models for SMS spam detection.
Among the tested algorithms, the best-performing model achieved high accuracy and robust generalization.
This project demonstrates the practical use of NLP and ML in communication security and can be expanded into
real-time systems for commercial deployment.