LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CU6051NI Artificial Intelligence

75% Individual Coursework

Submission: Final Submission

Academic Semester: Autumn Semester 2025

Credit: 15 credit semester long module

Student Name: Prashant Rijal

London Met ID: 23048683

College ID: np01230142

Assignment Due Date: 21/01/2026.

Assignment Submission Date: 21/01/2026

Submitted To: Er. Mukesh Regmi

GitHub Link	*https://github.com/Prashant-Rijal-dev/UCI_Spam_Detection*

I confirm that I understand my coursework needs to be submitted online via MST Classroom under the relevant module page before the deadline for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Table of Contents

Table Of Figures

Table Of Tables

1 Introduction

Artificial Intelligence (AI) is the field of computer science which focuses on creating computational system capable of performing, from the simplest of tasks to the most advanced and complex tasks that normally require a human. It involves the study of design and creation of intelligent agents which are equipped with reasoning (Logical or Probabilistic) to perform certain actions in a certain environment with some amount of autonomy.

AI in itself is not a single technology but a broad umbrella term for any systems that is designed to perform tasks that normally require human intelligence. It comprises of a huge range of technologies, methods, and applications that allow a machine to learn from experiences, the core fields that make AI what it is are:

- Machine Learning:
- Deep Learning:
- Computer Vision:
- Robotics:
- Natural Language Processing (NLP):

In this project we will be exploring NLP to address the text-based communication security, mainly Short Messages Service (SMS).

1.1 Problem Domain

The problem domain for this project is text-based communication security, mainly focused on identifying spam messages within the SMS communication system which is a classification problem. The SMS messages are mostly short, informal, and more often than not containing symbols and misleading content making spam detection a challenging task.

1.2 Aim

This project aims to design and develop a machine learning application which is capable of classifying SMS messages as either malicious messages (**Spam**) or real/legit messages (**Ham**), due to the growth of mobile communication SMS has become one of the primary targets for phishing attacks and unprompted marketing. This project aims to filter these types of messages in order to make user experience secure and solicit.

The project uses **Supervised Machine Learning** in which the model learns to map input data(Features) to output labels(Target Variable) based on the labelled dataset. For the project following concepts will be utilized:

- Natural Language Processing (NLP):
    - It is the field of AI focused on helping computers understand human language.
    - The use of NLP in this project is to clean the data (raw text) and convert it to numerical format for the computer to understand.

Figure 1fig. NLP

- Vectorization:
    - It is the process of converting non numerical data into numerical vectors for processing.
    - For the project TF-IDF (Term Frequency-Inverse Document Frequency) will be used to transform text into numerical vectors.
    - TF-IDF‘s main concept is to assign weight based on their importance.

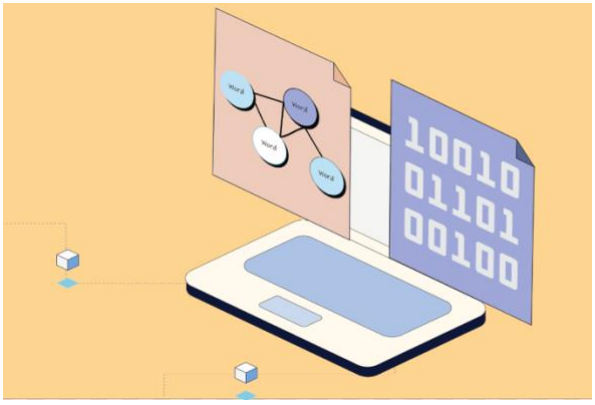

Figure 2 fig. Vectorization

- Classification Algorithm:
  - They are tools in ML which sorts data into categories or classes.
  - The project uses 3 distinct types of learning algorithms for classification model:
    - Multinomial Naïve Bayes (Probabilistic)

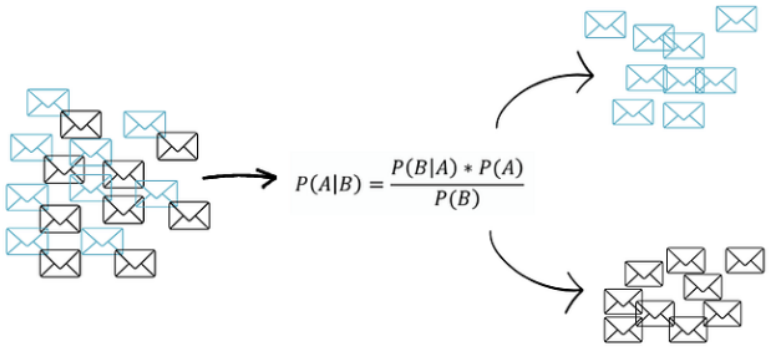

Figure 3 fig. Multinomial Naïve Bayes

- Logistic Regression (statistical)

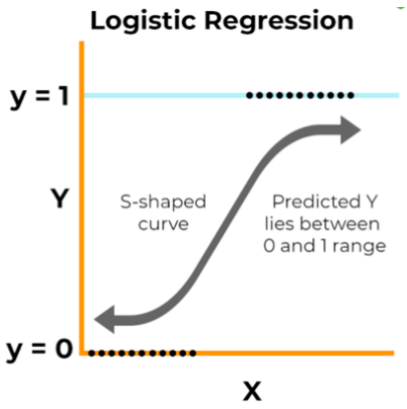

Figure 4 fig. Logistic Regression

- SVM Support Vector Machine (Margin Based)

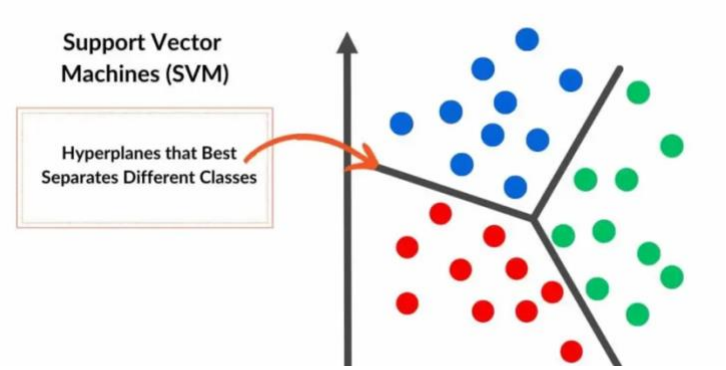

Figure 5 fig. SVM Support Vector Machine

2 Background

The SMS Spam Detection has been studied intensively in the field of NLP; the earliest approaches used the method of finding similar keywords however it was lacklustre and didn't do well as spam patterns kept evolving, now we are at the stage where we use labelled data to identify patterns use linear and probabilistic classifiers for such tasks.

2.1 Related Research

2.1.1 Contributions to the study of SMS spam filtering: new collection and results

Authors: Tiago A. Almeida, José María Gómez Hidalgo, Akebo Yamakami

Year: 2011

Method: Naïve Bayes, Support Vector Machine (SVM)

This is one of the most important papers in this field as it was the paper that first introduced the "SMS Spam Collection" dataset. It compared traditional ML method like Naïve Bayes, Support Vector Machine (SVM). (Almeida, et al., 2011)

2.1.2 A Hybrid CNN-LSTM Model for SMS Spam Detection

Authors: Abdallah Ghourabi, Mahmood A. Mahmood, Qusay M Alzubi

Year: 2020

Method: CNN-LSTM

This paper was released and was a staple paper which showed the new concept of word embedding and semantic meaning. The researchers used CNN and LSTM model to find patter and understand sequence of sentence. (Ghourabi, et al., 2020)

2.1.3 Spam Detection Using BERT

Author: Thaer Sahmoud, M. Mikki

Year: 2022

Method: BERT/Transformer

This paper came after the introduction of transformer concept, instead of using just SMS data to train the model it uses BERT which is trained on the entire internet and finetunes it for SMS spam. (Sahmoud & Mikki, 2022)

2.2 Dataset Description

Dataset used: **UCI SMS Spam Collection**

The project utilizes the UCI SMS Spam Collection which is a public dataset sourced from **University of California, Irvine's (UCI) Machine Learning Repository**.

Here are some facts that we can get from the dataset: (shown in figure 1)

- Type: .txt (the original dataset is in txt file format)
- No of rows of data: 5572
- Data Type: Unstructured English data (object)
- Class Imbalance: Heavily Imbalanced
    - Ham=4824 (86.591276%) ,
    - Spam=747 (13.408724%)

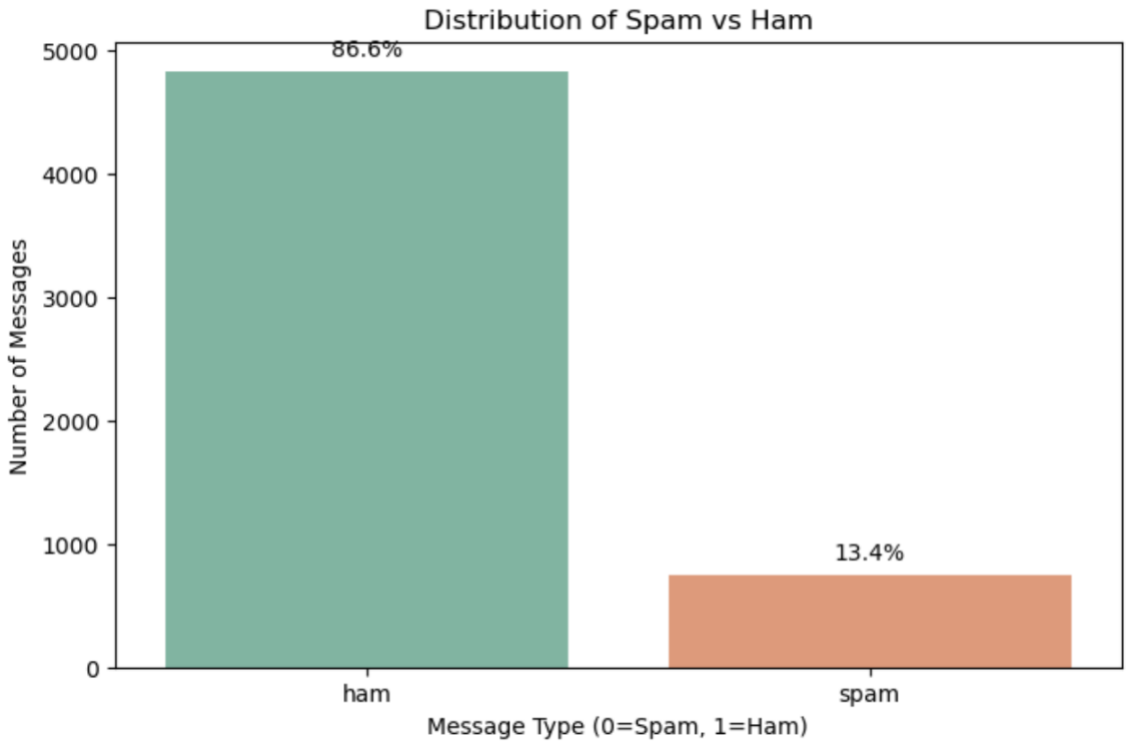

Figure 6 Dataset class distribution

Societal and Business Relevance

- Cybersecurity: SMS phishing is a major threat in current time as it is used to steal banking details or other private information, this classifier is like the first line of defence against these types of threats.
- Business Integrity: Even now almost all top businesses and companies such as X, Meta etc rely on SMS for one time OTP but if due to constant spam email threats users stop trusting or using SMS it compromises security.
- User friendly: the spam filters automatically filter spam SMS saving time for the user and also decluttering mobile storage.

3 Solution

The solution developed for this implements a full machine learning pipeline:

- Data Loading
- Text Preprocessing
- Feature Extraction
- ML Algorithms

3.1 Development Process:

3.1.1 Text Preprocessing:

The raw data (text) is noisy, so we apply the following techniques:

- Normalization: lowercases the text

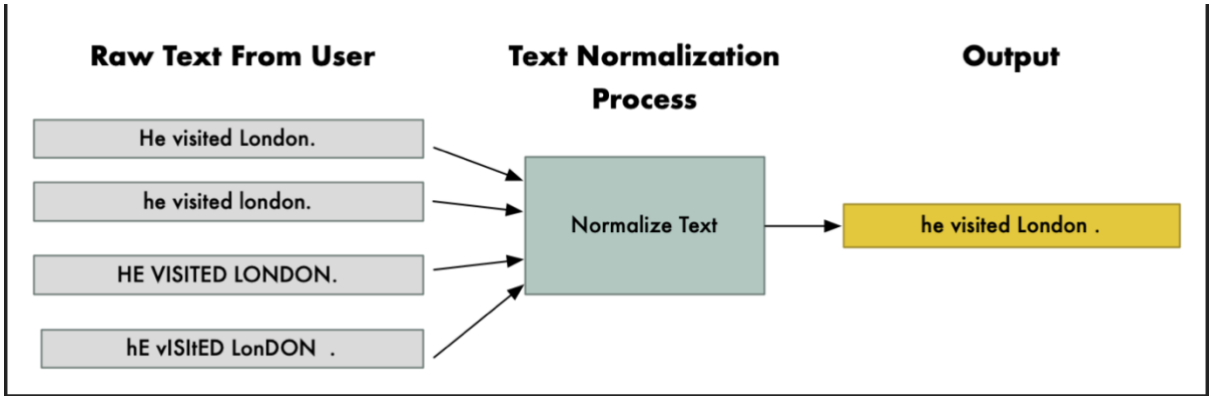

Figure 7 fig. Normalization

- Stop Word Removal: removes words like is, the, that, etc.

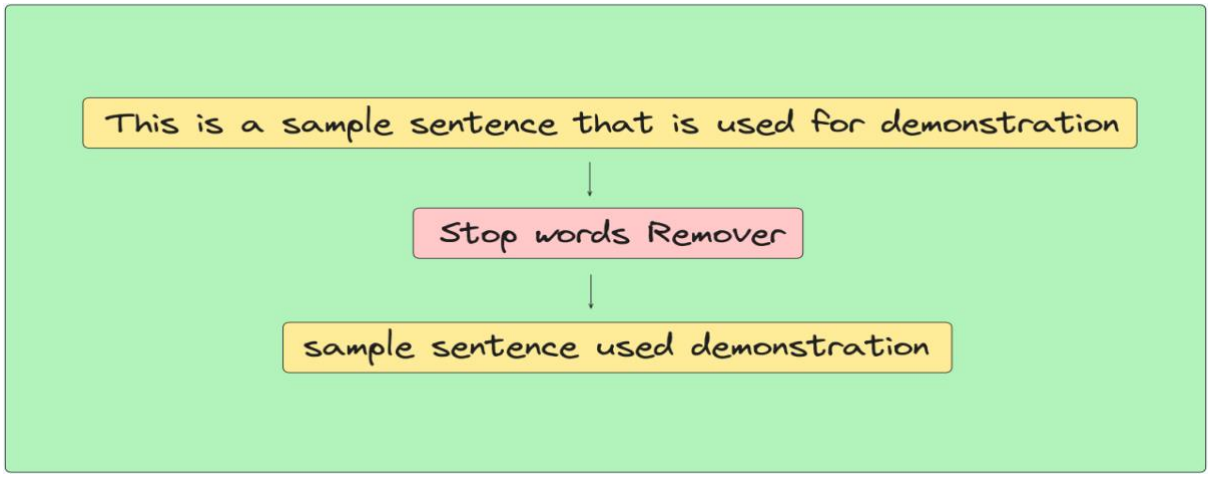

Figure 8 fig. Stopword Removal

- Stemming: convert word to root form

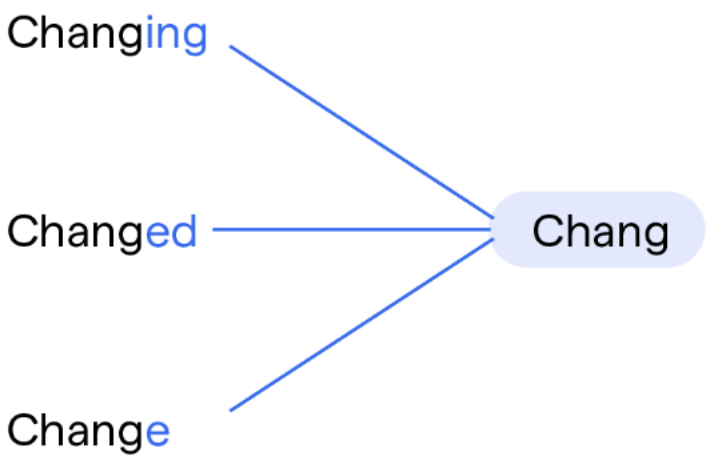

Figure 9 fig. Stemming

- Tokenization: Breaks sentences into words.

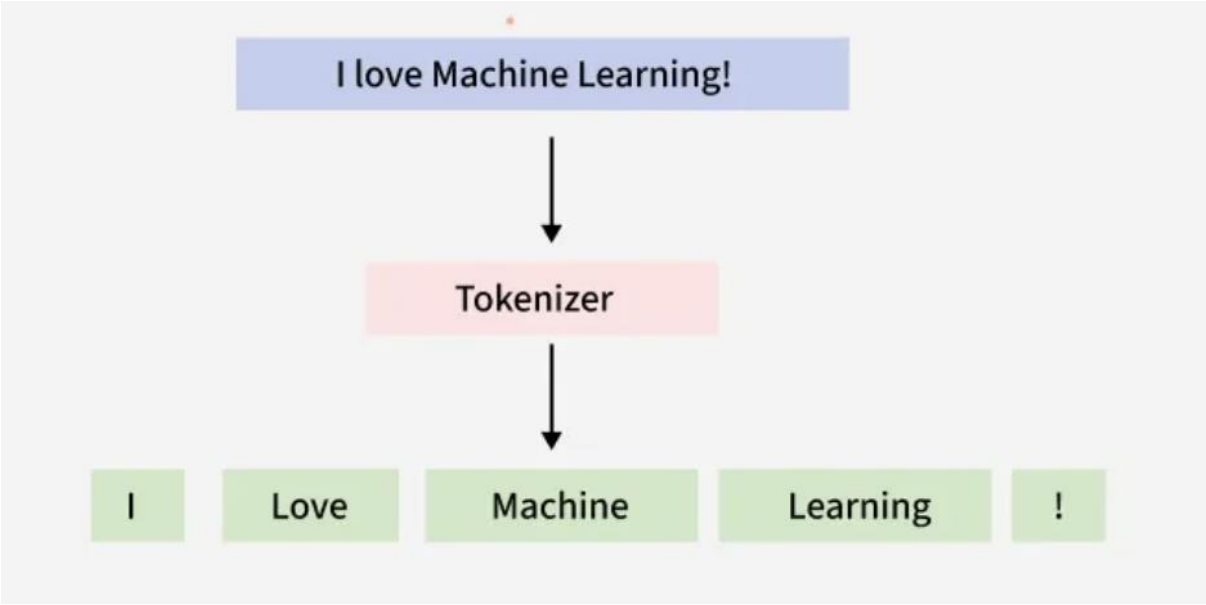

Figure 10 fig. Tokenization

3.1.2 Feature Extraction

Since the data is in textual form, we need to convert it to numerical form and for those following techniques is to be used:

- TF-IDF Vectorization converts text into numerical vectors. (This method was chosen because it normalizes count of words which prevents longer messages from having unfair weightage.)

$$TF(t,d) = \frac{number\ of\ times\ t\ appears\ in\ d}{total\ number\ of\ terms\ in\ d}$$

$$IDF(t) = log\frac{N}{1+df}$$

$$TF - IDF(t,d) = TF(t,d) * IDF(t)$$

Figure 11 fig. TF-IDF

3.1.3 Machine Learning Algorithm:

Following machine learning algorithm are to be used:

- Multinomial Naïve Bayes:
    - Concept: assumes independence between features
    - Advantage: computes extremely fast and performs excellent on discrete data
    - Uses techniques like Laplace smoothing to handle unseen words preventing zero probabilities.
- Logistic Regression:
    - Concept: Learns by creating linear decision boundary between classes.
    - Advantage: it provides clear probability score rather than label making it easy to adjust threshold
    - Uses sigmoid function to provides probabilities for prediction of class.
- SVM(Support Vector Machine):
    - Finds optimal hyperplane and divides classes
    - It handles high dimensional data (vectors) easily and is less likely to overfit.
    - Instance based supervised machine learning algorithm.

3.2 Pseudocode

3.2.1 Pseudocode for Naïve Bayes:

```
PROCESS Naive_Bayes_Training
  INPUT: Labeled_Dataset

  CALCULATE P_Spam AS Count(Spam_Messages) / Count(Total_Messages)
  CALCULATE P_Ham AS Count(Ham_Messages) / Count(Total_Messages)

  INITIALIZE Dictionary_Spam
  INITIALIZE Dictionary_Ham

  FOR EACH Message IN Labeled_Dataset
    IF Label IS "Spam" THEN
      INCREMENT Word_Counts IN Dictionary_Spam
    ELSE
      INCREMENT Word_Counts IN Dictionary_Ham
    END IF
  END FOR

  STORE P_Spam, P_Ham, Dictionary_Spam, Dictionary_Ham
END PROCESS
```

For Output:

```
PROCESS Naive_Bayes_Prediction
  INPUT: New_Message, Model

  SET Spam_Score TO P_Spam
  SET Ham_Score TO P_Ham

  FOR EACH Word IN New_Message
```

```
        COMPUTE Prob_Word_Spam FROM Dictionary_Spam
        COMPUTE Prob_Word_Ham FROM Dictionary_Ham

        MULTIPLY Spam_Score BY Prob_Word_Spam
        MULTIPLY Ham_Score BY Prob_Word_Ham
    END FOR

    IF Spam_Score > Ham_Score THEN
        RETURN "Spam"
    ELSE
        RETURN "Ham"
    END IF
END PROCESS
```

3.2.2 Pseudocode for logistic regression**:**

```
PROCESS Logistic_Regression_Training
  INPUT: Dataset, Learning_Rate, Epochs

  INITIALIZE Weights TO 0
  INITIALIZE Bias TO 0

  FOR i FROM 1 TO Epochs
    FOR EACH Message IN Dataset
      CALCULATE Linear_Score AS (Weights * Message) + Bias
      CALCULATE Prediction AS Sigmoid(Linear_Score)

      CALCULATE Error AS Label - Prediction

      UPDATE Weights AS Weights + (Error * Learning_Rate * Message)
      UPDATE Bias AS Bias + (Error * Learning_Rate)
    END FOR
  END FOR

  STORE Weights, Bias
END PROCESS
```

For Output:

```
PROCESS Logistic_Regression_Prediction
  INPUT: New_Message, Weights, Bias

  CALCULATE Linear_Score AS (Weights * New_Message) + Bias
  CALCULATE Probability AS Sigmoid(Linear_Score)

  IF Probability > 0.5 THEN
```

```
        RETURN "Spam"
    ELSE
        RETURN "Ham"
    END IF
END PROCESS
```

3.2.3 Pseudocode for KNN

```
PROCESS SVM_Training
  INPUT: Dataset, Learning_Rate, Epochs, Lambda

  INITIALIZE Weights AND Bias with random values

  FOR i FROM 1 TO Epochs
    FOR EACH Message IN Dataset
      CALCULATE Position AS (Weights * Message) - Bias

      IF (Label * Position) < 1 THEN
        UPDATE Weights AS Weights - Learning_Rate * (2 * Lambda * Weights - (Label * Message))
        UPDATE Bias AS Bias - Learning_Rate * Label
      ELSE
        UPDATE Weights AS Weights - Learning_Rate * (2 * Lambda * Weights)
      END IF
    END FOR
  END FOR

  STORE Weights, Bias
END PROCESS
```

For Output:

```
PROCESS SVM_Prediction
  INPUT: New_Message, Weights, Bias

  CALCULATE Result AS (Weights * New_Message) - Bias

  IF Result >= 0 THEN
```

```
        RETURN "Spam"
    ELSE
        RETURN "Ham"
    END IF
END PROCESS
```

3.3 Flowcharts

3.3.1 Multinomial Naïve Bayes

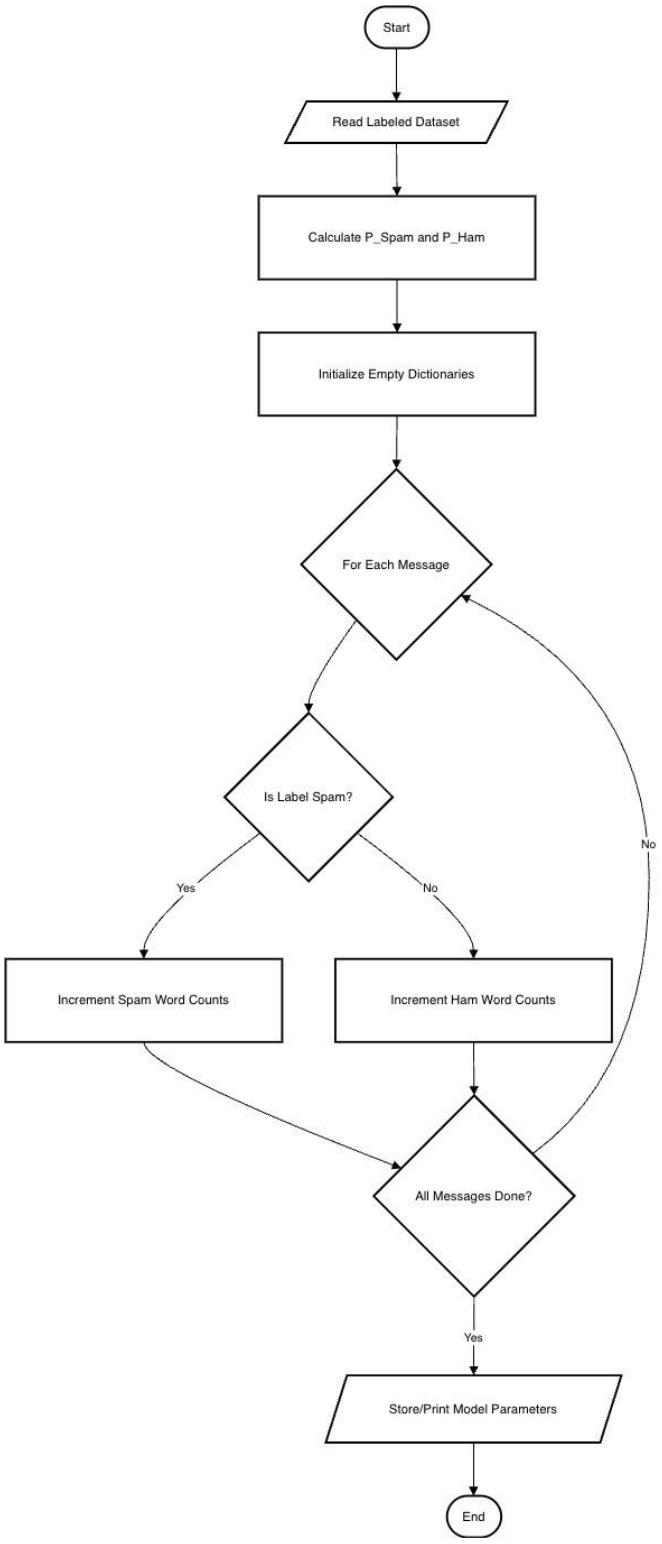

Figure 12 fig. Multinomial Naïve Bayes Flowchart

3.3.2 Logistic Regression

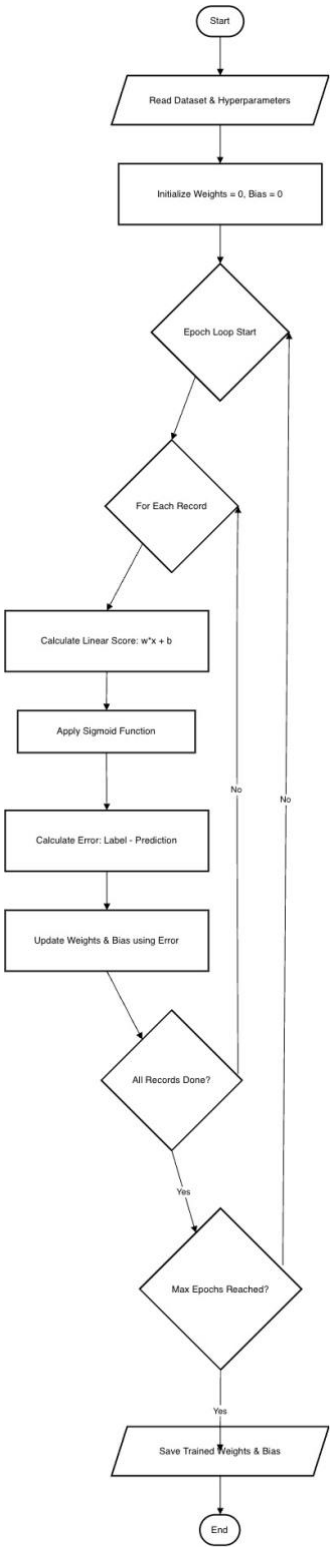

Figure 13 fig. Logistic Regression Flowchart

3.3.3 Support Vector Machine (SVM)

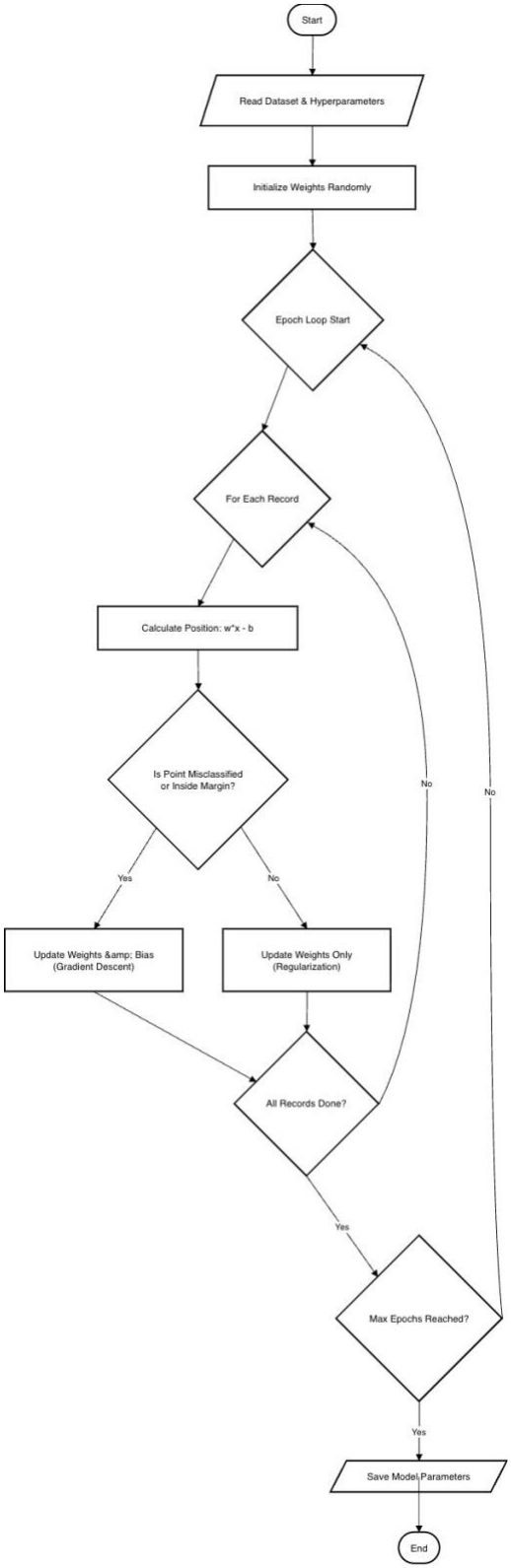

Figure 14 fig. Support Vector Machine (SVM) Flowchart

3.4 Tools and Technologies Used

For the completion of the project following tools and technologies were used:

- **Programming Language and Environment**
  - **Python 3.0:**

    This is primary language chosen for the project due to its easiness and vast libraries.

Figure 15 Python

  - **Jupyter Notebook:**

    This is the IDE (Integrated Development Environment) chosen for the project which is due to its cell-based structure and other auxiliary functions.

Figure 16 Jupyter notebook

- **Data Manipulation and Analysis:**

- Pandas: used to load and transform the data

Figure 17 Pandas

- **Machine Learning and NLP**
  - **Scikit-learn:** The most critical library for the project this library holds all the core logic of the models that we ran e.g.
    - TFidfVectorizer
    - Train Test Split
    - Multinomial Naive Bayes, Logistic Regression, SVM
    - Evaluation Matrices

Figure 18 Scikit Learn

- **Nltk:** The primary library mostly used for text preprocessing tasks such as:
  - Stopword removal
  - Stemming

Figure 19 NLTK

- **re (Regular Expression):** The library used to handle removal of punctuations, numbers and special characters.

Figure 20 Regular Expression (re)

In summary:

Table 1 fig. Tools and Technologies

Category	Technology
Language	Python
Environment	Jupyter Notebook
Data Cleaning	Pandas, Re (Regex)
NLP	NLTK (Stemming, Stopwords)
Vectorization	TF-IDF (Scikit-learn)
ML Algorithms	Multinomial Naive Bayes, Logistic Regression, SVM

4 Result

After doing all the processes for the creation of the project and all its processes here are the final outcomes for each of the ML models and how they performed in the given dataset and also in real world (unseen) messages.

4.1 Multinomial Naïve Bayes

	precision	recall	f1-score	support
0	0.99	0.87	0.92	149
1	0.98	1.00	0.99	966
accuracy			0.98	1115
macro avg	0.99	0.93	0.96	1115
weighted avg	0.98	0.98	0.98	1115

Figure 21 Performance Index of Multinomial Naive Bayes

This model's performance is the best on the task of SMS spam detection. It has achieved high accuracy and recall in identifying spam messages but a bit low on the ham message detection (which is due to the imbalanced nature of our dataset).

The confusion matrix for the model demonstrates that that model has correctly classified major ham messages while also maintaining a low number of false negative's for spam which shows that the model is effective at detecting SMS spam messages.

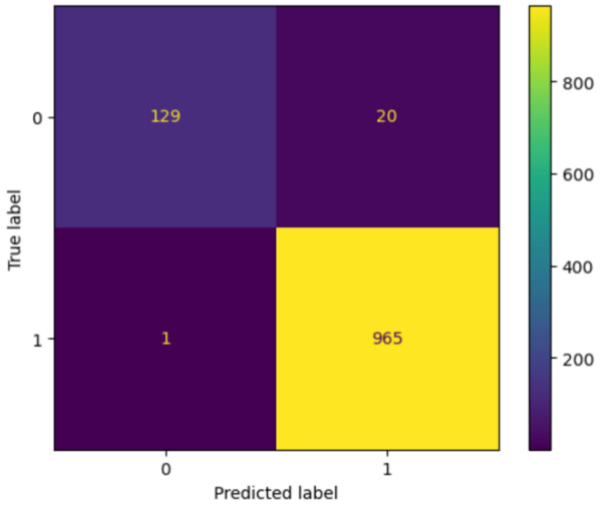

Figure 22 Confusion Matrix for multinomial Naive Bayes

4.2 Logistic Regression

The Logistic Regression model achieved good performance, but it lacks a bit in comparison to Multinomial Naïve Bayes model. The model also shows high accuracy but the recall for ham detection is a bit lacking.

	precision	recall	f1-score	support
0	1.00	0.79	0.88	149
1	0.97	1.00	0.98	966
accuracy			0.97	1115
macro avg	0.98	0.89	0.93	1115
weighted avg	0.97	0.97	0.97	1115

Figure 23 Performance Index of Logistic Regression

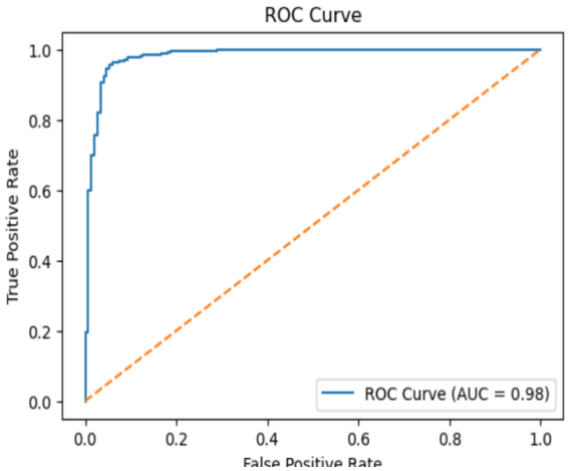

Figure 24 ROC curve for Logistic Regression

The Receiver Operating Characteristics (ROC) curve dose not show much but the high area under the curve indicates high separability between the spam and ham classes.

4.3 Support Vector Machine (SVM)

	precision	recall	f1-score	support
0	0.99	0.89	0.94	149
1	0.98	1.00	0.99	966
accuracy			0.98	1115
macro avg	0.99	0.94	0.96	1115
weighted avg	0.98	0.98	0.98	1115

Figure 25 Performance Index for SVM

For this model LinearSVC was used as it boosts efficiency as well as handles text classifications very well. Its performance in this project is excellent whith high scores all over the board

As the confusion matrix shows the model has very low error rates with a little error rate while predicting ham, messages

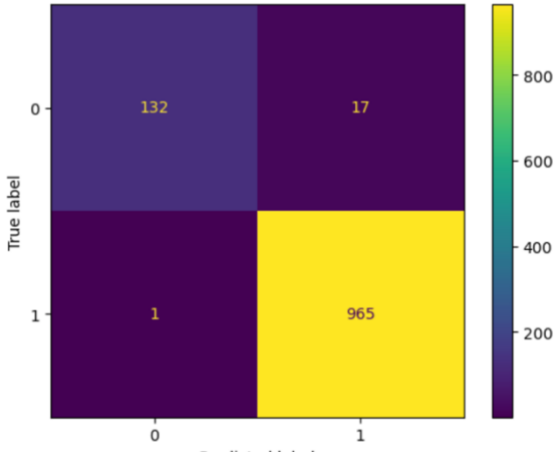

Figure 26 Confusion Matrix for SVM

4.4 Ensemble Learning

The Ensemble Learning approach was implemented to further enhance the predictive power and robustness of the SMS spam detection system. This approach combines multiple models to give a final prediction that is more reliable than one single individual system.

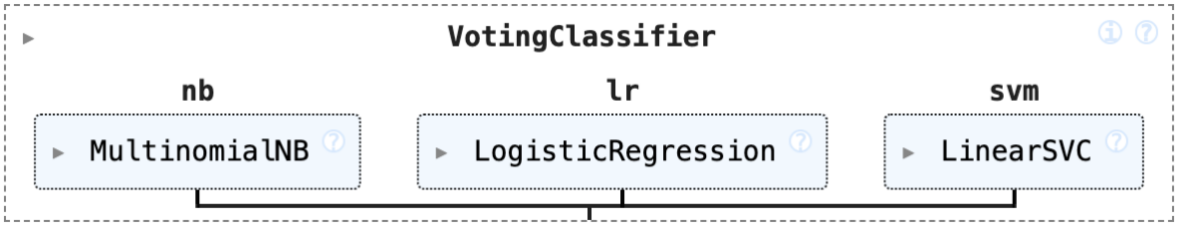

Figure 27 Ensemble Learning

4.4.1 Model Selection and Mechanism

For this project a majority voting classifier was utilized, this model masses the prediction of the three previously trained algorithms:

- Multinomial Naïve Bayes
- Logistic Regression
- Support Vector Machine (SVM)

This ensemble model used the hard voting logic, where each model casts a vote for class 0 or 1 and the class that receives the most amount of votes becomes the ensembles final prediction

```
Ensemble Accuracy: 0.979372197309417

Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.85      0.92       149
           1       0.98      1.00      0.99       966

    accuracy                           0.98      1115
   macro avg       0.99      0.92      0.95      1115
weighted avg       0.98      0.98      0.98      1115

Naive Bayes Accuracy: 0.9811659192825112
Logistic Regression Accuracy: 0.9713004484304932
SVM Accuracy: 0.9838565022421525
Ensemble Accuracy: 0.979372197309417
```

Figure 28 Performance Index for Ensemble Learning

4.5 Testing

After all this process of cleaning, processing and training the models in the labelled SMS spam dataset following were the results

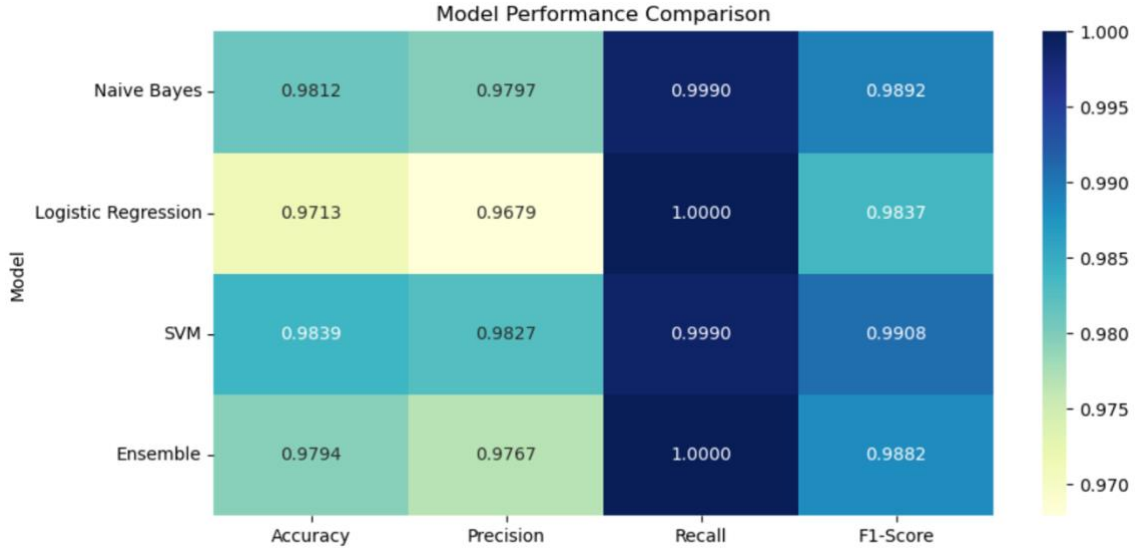

Figure 29 Comparison between Models performance

Among all the models according to this experimental project Multinomial Naïve Bayes is overall the best performer with high accuracy, precision, recall and f1 score among all other models tested in this category.

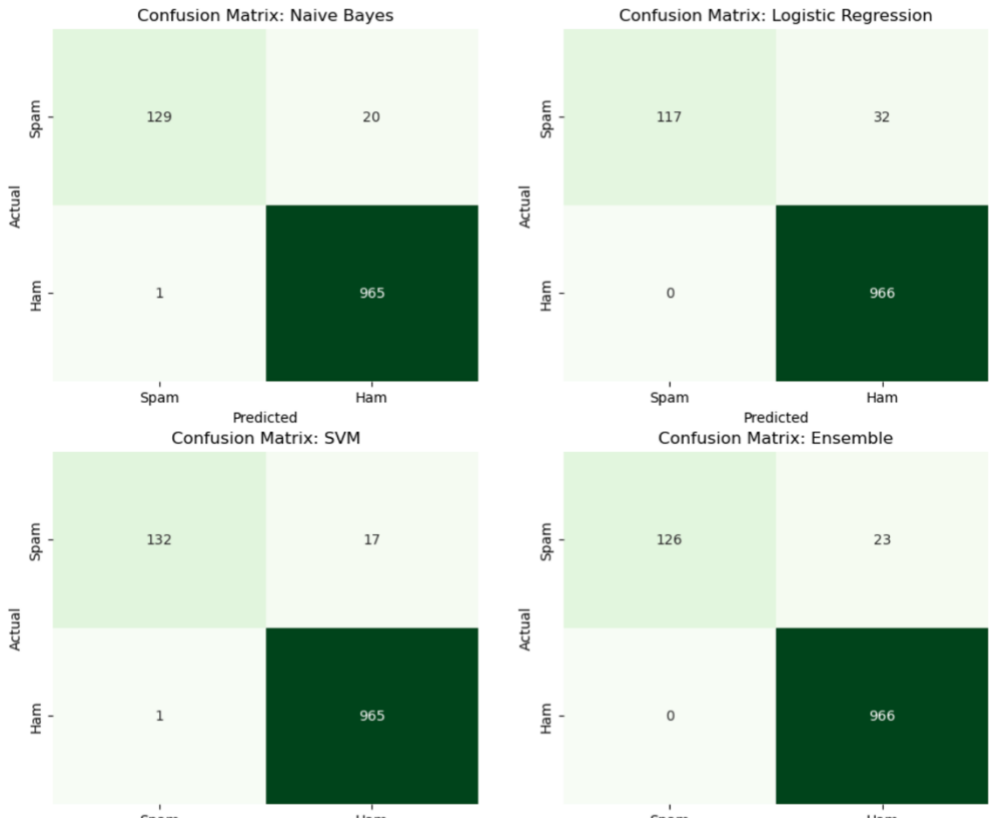

Figure 30 Side By side comparison of confusion matrix

4.5.1 Testing in Unseen Data

After all the results were gathered and all the models were working and functional then unseen data was fit into the model to see how it performs and surprisingly its performs well on some particular types of messages which are similar to the dataset which is old school data.

```
[158]: test_new_message("WINNER!! You have won a 1000 dollar prize. Call 09061701461 to claim your gift now!")

Original Message: WINNER!! You have won a 1000 dollar prize. Call 09061701461 to claim your gift now!
------------------------------
Naive Bayes          : SPAM
Logistic Regression  : SPAM
SVM                  : SPAM
Ensemble             : SPAM
```

Figure 31 Test 1: unseen data similar to the dataset

However, when we test it on messages that are used by spammers, fishers we get a completely different output.

```
[168]: test_new_message("URGENT: Your bank account has a security alert. Please verify your details at bit.ly/fraud-check")

Original Message: URGENT: Your bank account has a security alert. Please verify your details at bit.ly/fraud-check
------------------------------
Naive Bayes          : HAM
Logistic Regression  : HAM
SVM                  : HAM
Ensemble             : HAM
```

Figure 32 Test 2: unseen data not similar to the dataset

Not a single model is capable to identify the spam message as a spam which is due to it having no sample of it in the training dataset. This is a huge problem as it cannot be used in real world scenarios if it does not have adequate classifying skillset and in order to make this, we need to constantly change the dataset adding new messages that spammers and fishers use or use deep learning and sentiment based analysis models to counter this which is not viable for devices with no top of the line specs and computing power.

5 Conclusion

The project addresses the growing cybersecurity threat of SMS phishing, scamming and tries to solve this problem by using Supervised Machine Learning and NLP. The project aims to build a classifier capable of somewhat accurately distinguish between Spam and Ham messages.

The proposed solution implements a complete ML pipeline starting from data collection to data preprocessing and finally model training and output. The comparative analysis of the three algorithms Naïve Bayes, Logistic Regression and Support Vector Machine (SVM) will ensure that the most efficient and accurate model is selected for final application.

This project is not just a academic exercise but a practical application of all the knowledge and skills accumulated. The successful implementation of this project will demonstrate the effectiveness of NLP and ML to solve the problem of SMS Scam Detection.

Bibliography

Almeida, T. A. d., Hidalgo, J. M. G. & Yamakami, A., 2011. *ACM Digital Library.* [Online]
Available at: https://dl.acm.org/doi/10.1145/2034691.2034742
[Accessed 14 December 2025].

Ghourabi, A., Mahmood, M. A. & Al-Zubi, Q. M., 2020. *MDPI.* [Online]
Available at: https://www.mdpi.com/1999-5903/12/9/156
[Accessed 14 December 2025].

Sahmoud, T. & Mikki, M., 2022. *Cornell University AirXiv.* [Online]
Available at: https://arxiv.org/abs/2206.02443
[Accessed 14 December 2025].