# **Project Report**

**Name:** Madhan kumar.I  
 **Course:** Bsc cs(Aiml)  
 **Registration Number:** Tu 6243202111056

### **Project Title:**

SMS Spam Detection using Logistic Regression

## **1. Project Description**

This project focuses on classifying SMS messages as either spam or ham (not spam) using Natural Language Processing (NLP) and machine learning techniques. The goal is to build a robust binary classification model that automatically detects spam messages. The project applies text preprocessing, TF-IDF feature extraction, and Logistic Regression for effective spam detection.

## **2. Learning Objectives**

* Objective 1: Learn NLP preprocessing techniques (tokenization, stopword removal, stemming/lemmatization).
* Objective 2: Apply Logistic Regression for binary classification of SMS messages.
* Objective 3: Understand and evaluate performance metrics such as precision, recall, F1-score, and confusion matrix.

## **3. Timeline**

* Start Date: Sept 24, 2025
* Submission date: Sept 24, 2025

## **4. Algorithm Used**

* Algorithm Name: Logistic Regression
* Explanation: Logistic Regression is a supervised classification algorithm that models the probability of an event using the sigmoid function. It was chosen for its simplicity, interpretability, and efficiency in handling high-dimensional TF-IDF features.

## **5. Tools & Libraries**

* **Programming Language:** Python
* **Libraries Used:**
  + Pandas  
    NumPy  
    NLTK  
    Scikit-learn  
    Matplotlib  
    Seaborn  
    WordCloud
  + NumPy
  + Scikit-learn
  + Matplotlib / Seaborn
  + TensorFlow / PyTorch (if used)
  + [Any other relevant tool]

## **6. Dataset Description**

* Source: UCI SMS Spam Collection Dataset
* Size: 5574 rows, 2 main columns (label, message)
* Target Variable: label (ham=0, spam=1)
* Description of Features: Features: SMS message text, engineered features such as message length, token counts, and TF-IDF scores.

## **7. Methodology**

* Data Preprocessing: Text cleaned (punctuation, digits, stopwords removed), tokenized and stemmed. TF-IDF used for feature extraction.
* Model Training: Dataset split into 80% train and 20% test. Logistic Regression trained on TF-IDF features.
* Evaluation: Accuracy, Precision, Recall, F1-score, Confusion Matrix used to evaluate model performance.
* Hyperparameter Tuning: Hyperparameter tuning was performed by adjusting the regularization parameter C. Comparison with Naïve Bayes as baseline.

## **8. Results**

* Performance Metrics: Accuracy: ~97%  
  Precision: ~95%  
  Recall: ~92%  
  F1-score: ~93%
* Visualizations: Confusion matrix and message length distribution plots included. WordCloud used to visualize frequent spam terms.
* Insights: Logistic Regression with TF-IDF is highly effective. Spam messages often contain promotional keywords, links, and longer text compared to ham.

Attach screenshot of outputs and short description

## **9. Questions Answered**

* Q1: Q1: What features indicate spam?  
  Q2: How to clean and tokenize text?  
  Q3: What is TF-IDF?  
  Q4: What is precision vs recall?  
  Q5: How to handle class imbalance?  
  Q6: What is confusion matrix?  
  Q7: How to improve accuracy?  
  Q8: What are false positives?  
  Q9: How does logistic regression work?  
  Q10: How to visualize spam patterns?  
  Q11: What is tokenization and why is it important?  
  Q12: How does stemming differ from lemmatization?  
  Q13: Why use Logistic Regression instead of Naïve Bayes?  
  Q14: What evaluation metrics are better than accuracy in spam detection?  
  Q15: How do n-grams help in detecting spam phrases?

## **10. Challenges & Improvements**

* Class imbalance (ham > spam), noisy text (slangs, special characters), some false positives misclassify important messages as spam.
* Try ensemble methods (Random Forest, XGBoost), deep learning (LSTM, BERT), and add metadata features like sender ID or time of message.

## **11. References**

* Dataset Links: https://archive.ics.uci.edu/ml/datasets/sms+spam+collection
* Research Papers / Documentation: Scikit-learn documentation, NLTK documentation

## **12. GitHub Link**

## https://github.com/maddy50-star/SMS-spam/upload