Project Report – SMS Spam Detection

Project Name: SMS Spam Detection

Team Name: CodeGuardians
Team Members:
1. Abirami N
2. Sri yaline R
3. Nithila

1. Project Details

This project aims to classify SMS messages into spam or ham (legitimate) using Natural Language Processing (NLP) and machine learning techniques. The dataset contains thousands of real-world SMS messages labeled accordingly. The system processes raw text, extracts relevant features, and applies a trained model to predict whether a message is spam.

2. How We Did It

Step 1 – Data Cleaning
- Loaded the SMSSpamCollection dataset. -
Removed duplicate entries.
- Checked and handled missing values. -
Renamed columns for clarity.

Step 2 – Exploratory Data Analysis (EDA)
- Checked the distribution of spam and ham messages (approx. 86% ham, 14% spam). -
Visualized data using pie charts.
- Added a feature for message length (num_characters).

Step 3 – Text Preprocessing
- Converted all text to lowercase.
- Tokenized sentences into words.
- Removed punctuation, numbers, and stopwords.
- Applied stemming using Porter Stemmer and Lancaster Stemmer. -
Lemmatized words to their base form.

Step 4 – Model Building
- Converted text to numerical features using TF-IDF vectorization. -
Split dataset into training and test sets.
- Trained models like Naïve Bayes, Logistic Regression, and Support Vector Machine.

Step 5 – Evaluation
- Best model: Multinomial Naïve Bayes. -
Accuracy: ~97% - Precision: ~97%

- Recall: ~96%
- F1 Score: ~96%

3. Sources Used

- Dataset: SMS Spam Collection Dataset – UCI Machine Learning Repository.
- Python Libraries: Pandas, NumPy, Scikit-learn, NLTK, Matplotlib, Seaborn.
- References:
https://archive.ics.uci.edu/ml/datasets/SMS+Spam+Collection
Scikit-learn Documentation.
NLTK Documentation.

4. What We Have Learned

- Data preprocessing techniques for NLP.
- How to clean and prepare real-world text data.
- Feature extraction with TF-IDF.
- Applying and comparing multiple ML algorithms.
- Interpreting model performance metrics.
- Deploying ML models in Python environments.

Conclusion

The developed SMS Spam Detection model effectively identifies spam messages with high accuracy, providing a practical tool for filtering unwanted content. With further improvements such as deep learning integration, accuracy could be enhanced even more.