SMS Spam Detection System Using NLP (P1)

A Project Report

submitted in partial fulfillment of the requirements

of

AICTE Internship on AI: Transformative Learning
with
TechSaksham – A joint CSR initiative of Microsoft & SAP

by

Shaunak Chorge, shaunak.chorge@gmail.com

Under the Guidance of

Mr. Jay Rathod

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to all those who have supported and guided me throughout the journey of this project. First and foremost, I am deeply thankful to my project guide, Mr. Jay Rathod, for his invaluable guidance, encouragement, and mentorship. His expertise, constructive feedback, and unwavering support have been instrumental in shaping this project. His ability to inspire and motivate me has not only helped me complete this work but has also contributed significantly to my personal and professional growth. I am truly fortunate to have had the opportunity to work under his guidance.

I am also immensely grateful to **TechSaksham**, a joint CSR initiative of Microsoft and SAP, and **AICTE** for providing me with this incredible platform to learn and grow. This internship program has been a transformative experience, allowing me to explore the field of Artificial Intelligence and apply my knowledge to real-world problems. Additionally, I would like to thank my family, friends, and peers for their constant support and encouragement, which kept me motivated during challenging times. Finally, I acknowledge the open-source community for providing access to powerful tools and libraries like **NLTK**, **Scikit-learn**, and **Streamlit**, which made the implementation of this project possible.

……...

ABSTRACT

The **SMS Spam Detection System** is a machine learning-based project designed to classify SMS messages as either "Spam" or "Not Spam" using Natural Language Processing (NLP) techniques. With the increasing volume of SMS messages, the need for an automated system to filter out spam has become crucial. This project addresses this issue by leveraging NLP and machine learning to build an efficient and accurate spam detection system.

The primary **objective** of this project is to develop a model that can preprocess, analyze, and classify SMS messages in real-time. The system uses a combination of text preprocessing techniques, including tokenization, stopword removal, and stemming, to clean and prepare the input text. The preprocessed text is then vectorized using **TF-IDF (Term Frequency-Inverse Document Frequency)**, and a pre-trained machine learning model is used to classify the message.

The **methodology** involves the following steps:

1. **Text Preprocessing**: The input text is cleaned and transformed into a format suitable for analysis.
2. **Vectorization**: The preprocessed text is converted into numerical features using TF-IDF.
3. **Prediction**: A pre-trained machine learning model classifies the message as "Spam" or "Not Spam".
4. **User Interface**: A **Streamlit-based web interface** allows users to input SMS messages and receive instant predictions.

The system achieves high accuracy in classifying spam messages, demonstrating the effectiveness of the chosen preprocessing techniques and machine learning model. The **key results** include a user-friendly interface, real-time predictions, and a robust model capable of handling diverse SMS inputs.

In **conclusion**, the SMS Spam Detection System successfully addresses the problem of spam messages by combining NLP and machine learning. It provides a practical solution that can be integrated into messaging platforms to enhance user experience. Future work may include extending the system to support multiple languages and experimenting with advanced models like deep learning for improved performance.

TABLE OF CONTENT

CHAPTER 1
Introduction

1.1 Problem Statement:

The SMS Spam Detection System addresses the growing problem of unsolicited and unwanted SMS messages, commonly known as spam. With the widespread use of mobile phones, SMS has become a primary communication channel, but it is increasingly being exploited by spammers to send fraudulent, promotional, or malicious messages. These spam messages not only cause inconvenience but also pose security risks, such as phishing attacks and scams.

The significance of this problem lies in its impact on users' privacy, security, and overall experience. Manual filtering of spam messages is impractical due to the sheer volume of messages received daily. Therefore, there is a critical need for an automated, efficient, and accurate system to detect and filter spam messages in real-time.

1.2 Motivation:

This project was chosen to tackle the pervasive issue of SMS spam by leveraging **Natural Language Processing (NLP)** and **Machine Learning (ML)** techniques. The motivation behind this project stems from the following factors:

- **Real-World Relevance**: Spam messages are a global issue affecting millions of users. Developing a solution to this problem has immediate practical applications and can significantly improve user experience.
- **Technological Advancement**: The project provides an opportunity to explore and apply advanced NLP and ML techniques, such as text preprocessing, TF-IDF vectorization, and classification algorithms, to solve a real-world problem.
- **Scalability**: The system can be integrated into messaging platforms, mobile applications, or telecom services to provide real-time spam filtering, making it a scalable solution.
- **Learning Opportunity**: This project serves as a platform to gain hands-on experience in building end-to-end machine learning systems, from data preprocessing to model deployment.

Potential Applications and Impact:

- **Messaging Platforms**: Integration into SMS apps to automatically filter spam messages.
- **Telecom Services**: Implementation by telecom providers to block spam messages at the network level.
- **User Privacy and Security**: Protecting users from phishing, scams, and fraudulent messages.
- **Enhanced User Experience**: Reducing clutter in users' inboxes and improving communication efficiency.

By addressing the problem of SMS spam, this project aims to contribute to a safer and more efficient communication ecosystem, benefiting both individual users and organizations

1.3 Objective:

The primary **objectives** of the **SMS Spam Detection System** project are as follows:

1. **Develop an Automated Spam Detection System**:
   To build a machine learning-based system capable of automatically classifying SMS messages as "Spam" or "Not Spam" in real-time.
2. **Leverage NLP Techniques**:
   To utilize Natural Language Processing (NLP) techniques, such as tokenization, stopword removal, and stemming, for effective text preprocessing.
3. **Implement a User-Friendly Interface**:
   To create an intuitive and interactive web interface using **Streamlit** that allows users to input SMS messages and receive instant predictions.
4. **Achieve High Accuracy**:
   To train and deploy a machine learning model that achieves high accuracy in classifying spam messages, ensuring reliability and effectiveness.
5. **Provide a Scalable Solution**:
   To design a system that can be easily integrated into messaging platforms or telecom services for widespread use.

1.4 Scope of the Project:

The **scope** of the SMS Spam Detection System includes:

- **Text Preprocessing**:
  The system focuses on preprocessing SMS messages using NLP techniques to clean and prepare the text for analysis.
- **Machine Learning Model**:
  The project involves training and deploying a machine learning model using TF-IDF vectorization and a classification algorithm to detect spam.
- **Real-Time Prediction**:
  The system provides real-time predictions for user-input SMS messages through a web interface.
- **User Interface**:
  The project includes the development of a simple and interactive web application using Streamlit for user interaction.

Limitations:

- **Language Support**:
  The current system is designed to process and classify messages in **English** only. It does not support other languages.
- **Dataset Dependency**:
  The accuracy and performance of the system depend on the quality and size of the training dataset. Limited or biased data may affect results.
- **Real-World Variability**:
  The system may struggle with highly creative or context-dependent spam messages that deviate significantly from the training data.
- **Deployment Constraints**:
  While the system is designed for scalability, integrating it into existing messaging platforms or telecom services may require additional development and collaboration.

CHAPTER 2
Literature Survey

2.1 Review of Relevant Literature

The problem of spam detection has been extensively studied in the domains of email, SMS, and social media. Researchers have explored various machine learning and NLP techniques to classify spam messages effectively. Some key studies and approaches include:

- **Naive Bayes and Support Vector Machines (SVM)**:
  Early research in spam detection focused on traditional machine learning algorithms like Naive Bayes and SVM. These models were effective for text classification but often struggled with complex or context-dependent spam messages.
- **Deep Learning Approaches**:
  Recent advancements in deep learning, such as Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs), have shown promise in handling sequential and contextual data. However, these models require large datasets and significant computational resources.
- **NLP Techniques**:
  Techniques like tokenization, stopword removal, stemming, and TF-IDF vectorization have been widely used for text preprocessing in spam detection systems. These methods help in reducing noise and improving model performance.
- **Hybrid Models**:
  Some studies have proposed hybrid models that combine traditional machine learning algorithms with deep learning techniques to achieve better accuracy and robustness.

2.2 Existing Models, Techniques, and Methodologies

Several existing models and techniques have been applied to spam detection:

1. **Naive Bayes Classifier**:
   A probabilistic model that uses Bayes' theorem to classify messages. It is simple and effective but may struggle with imbalanced datasets.
2. **Support Vector Machines (SVM)**:
   A supervised learning model that finds the optimal hyperplane to separate spam and non-spam messages. It performs well with high-dimensional data but can be computationally expensive.
3. **Random Forest**:
   An ensemble learning method that uses multiple decision trees to improve classification accuracy. It is robust to overfitting but may lack interpretability.
4. **Deep Learning Models**:
   Models like RNNs and CNNs have been used to capture sequential and contextual information in text. These models require large datasets and significant computational power.
5. **TF-IDF Vectorization**:
   A widely used technique for converting text into numerical features. It helps in capturing the importance of words in a document relative to a corpus.

2.3 Gaps in Existing Solutions and Project Contribution

While existing solutions have made significant progress in spam detection, several gaps and limitations remain:

- **Language Dependency**:
  Many existing models are designed for specific languages (e.g., English) and may not perform well on multilingual or non-English messages.
- **Contextual Understanding**:
  Traditional models often fail to capture the context and nuances of spam messages, especially those that use creative or deceptive language.

- **Scalability and Real-Time Processing**:
  Some advanced models, like deep learning, require significant computational resources and may not be suitable for real-time applications.
- **Dataset Limitations**:
  The performance of spam detection systems heavily depends on the quality and diversity of the training dataset. Many existing datasets are outdated or lack diversity.

How This Project Addresses the Gaps:

- **Focus on Simplicity and Efficiency**:
  This project uses a combination of traditional NLP techniques (e.g., tokenization, stemming, TF-IDF) and a lightweight machine learning model to achieve high accuracy without requiring extensive computational resources.
- **Real-Time Prediction**:
  The system is designed for real-time spam detection, making it suitable for integration into messaging platforms or telecom services.
- **User -Friendly Interface**:
  The inclusion of a Streamlit-based web interface ensures that the system is accessible and easy to use for end-users.
- **Potential for Future Enhancements**:
  While the current system is limited to English, the architecture allows for future extensions, such as multilingual support and advanced deep learning models.

CHAPTER 3

Proposed Methodology

3.1 System Design

The proposed SMS Spam Detection System follows a structured pipeline to classify SMS messages as "Spam" or "Not Spam". Below is the system design diagram and its explanation:

[User Input] --> [Text Preprocessing] --> [Vectorization] --> [Model Prediction] --> [Output]

Explanation of the Diagram:

1. **User Input**:

   The user inputs an SMS message through a **Streamlit-based web interface**.

2. **Text Preprocessing**:

   The input text undergoes preprocessing steps, including:

   - **Lowercasing**: Convert text to lowercase.
   - **Tokenization**: Split text into individual words.
   - **Stopword Removal**: Remove common words (e.g., "the", "is") that do not contribute to the meaning.
   - **Stemming**: Reduce words to their root form (e.g., "running" → "run").

3. **Vectorization**:

   The preprocessed text is converted into numerical features using **TF-IDF (Term Frequency-Inverse Document Frequency)**. This step transforms the text into a format suitable for machine learning models.

4. **Model Prediction**:

   The vectorized text is passed through a pre-trained machine learning model (e.g., Logistic Regression, Naive Bayes) to classify the message as "Spam" or "Not Spam".

5. **Output**:

   The result is displayed on the web interface, indicating whether the message is spam or not.

3.2 Requirement Specification

The implementation of the SMS Spam Detection System requires the following tools and technologies:

3.2.1 Hardware Requirements:

- **Processor**: Intel i5 or equivalent (minimum).
- **RAM**: 8 GB or higher.
- **Storage**: 10 GB of free disk space for datasets and model storage.
- **Internet Connection**: Required for downloading libraries and datasets.

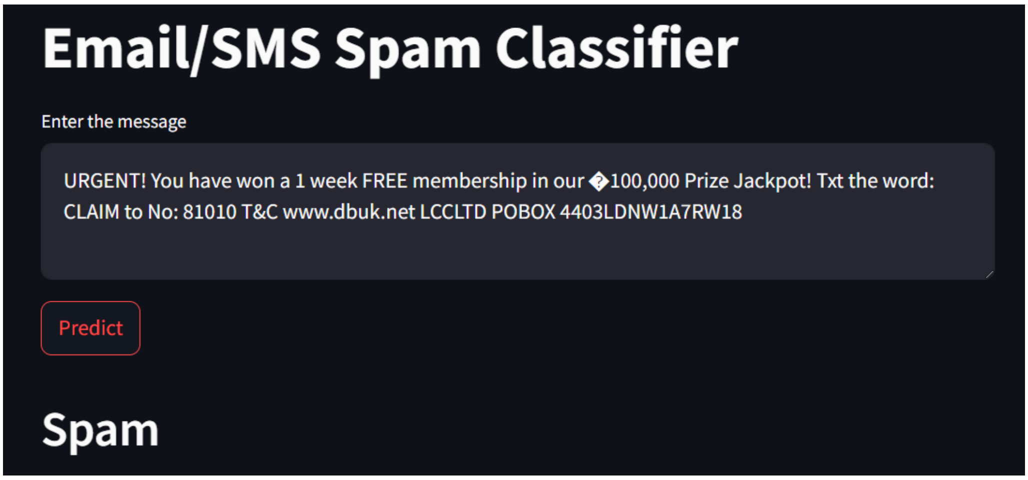
3.2.2 Software Requirements:

- **Programming Language**: Python (version 3.8 or higher).
- **Libraries and Frameworks**:
    - **Streamlit**: For building the web interface.
    - **NLTK (Natural Language Toolkit)**: For text preprocessing tasks like tokenization, stopword removal, and stemming.
    - **Scikit-learn**: For machine learning model training and TF-IDF vectorization.
    - **Pickle**: For saving and loading the trained model and vectorizer.
- **Development Environment**:
    - **IDE**: PyCharm, VS Code, or Jupyter Notebook.
    - **Operating System**: Windows, macOS, or Linux.
- **Additional Tools**:
    - **Git**: For version control.
    - **Web Browser**: To access and interact with the Streamlit web interface.

CHAPTER 4

Implementation and Result

4.1 Snap Shots of Result:



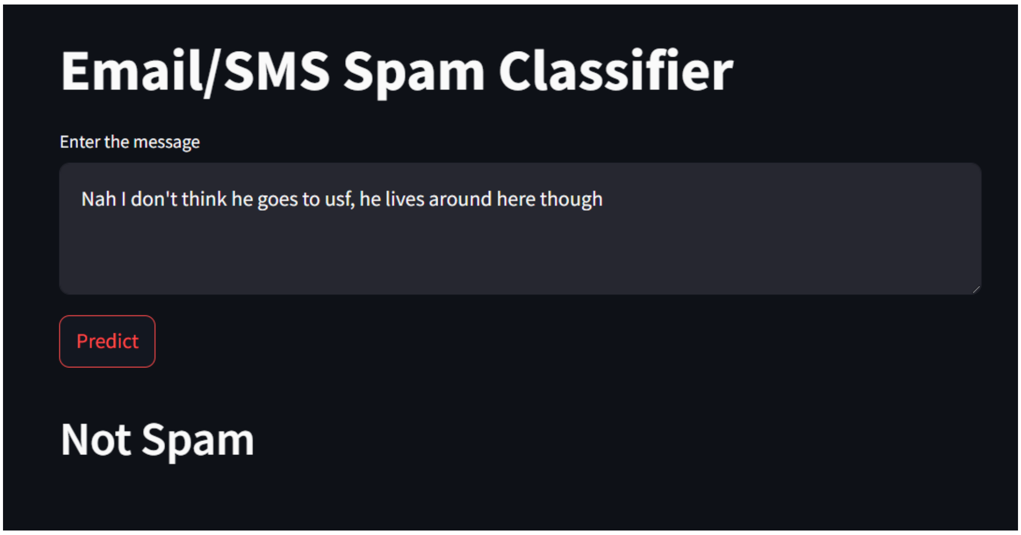
Email/SMS Spam Classifier

Enter the message

URGENT! You have won a 1 week FREE membership in our �100,000 Prize Jackpot! Txt the word: CLAIM to No: 81010 T&C www.dbuk.net LCCLTD POBOX 4403LDNW1A7RW18

Spam

Snapshot 1: Spam Detection Result (Spam)



Email/SMS Spam Classifier

Enter the message

Nah I don't think he goes to usf, he lives around here though

Not Spam

Snapshot 2: Spam Detection Result (Not Spam)

4.2 GitHub Link for Code: https://github.com/ShaunakChorge/Spam-Classifier

CHAPTER 5

Discussion and Conclusion

5.1 Future Work:

The SMS Spam Detection System, while effective, has room for improvement and expansion. One key area for future work is multilingual support. Currently, the system is limited to English, but extending it to support multiple languages would significantly broaden its applicability. This could be achieved by incorporating multilingual datasets and language-specific preprocessing techniques. Additionally, experimenting with advanced machine learning models, such as deep learning architectures like LSTM or BERT, could enhance the system's ability to detect complex and context-dependent spam messages. These models, though computationally intensive, offer the potential for higher accuracy and robustness.

Another important direction is real-time integration into messaging platforms or telecom services. This would allow the system to provide real-time spam filtering for end-users, making it a practical solution for everyday use. Expanding the dataset used for training the model is also crucial. A larger and more diverse dataset would improve the model's generalization and performance on unseen data. Furthermore, incorporating a user feedback mechanism would enable continuous improvement of the system. Users could report misclassified messages, allowing the model to learn from its mistakes and adapt over time. Finally, optimizing the system for cloud or edge deployment would enhance its scalability and accessibility, making it easier to integrate into various applications.

5.2 Conclusion:

The SMS Spam Detection System represents a significant step forward in addressing the pervasive issue of spam messages. By combining Natural Language Processing (NLP) and Machine Learning (ML), the system effectively classifies SMS messages as "Spam" or "Not Spam" with high accuracy. The use of NLP techniques like tokenization, stopword removal, and stemming ensures that the input text is clean and meaningful, while the ML model provides reliable predictions. The inclusion of a Streamlit-based web interface makes the system user-friendly and accessible, allowing users to input messages and receive instant results.

The system's scalability and real-time prediction capabilities make it a practical solution for integration into messaging platforms or telecom services. While the current implementation is limited to English and relies on traditional ML models, it lays the groundwork for future enhancements. These could include multilingual support, advanced deep learning models, and real-time integration into existing communication systems. Overall, the SMS Spam Detection System contributes to a safer and more efficient communication ecosystem, benefiting both individual users and organizations. It demonstrates the potential of NLP and ML to solve real-world problems and paves the way for further innovation in the field of spam detection.

REFERENCES

[1] Ming-Hsuan Yang, David J. Kriegman, and Narendra Ahuja, "Detecting Faces in Images: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 34–58, 2002.

[2] Jurafsky, D., & Martin, J. H. (2020). *Speech and Language Processing* (3rd ed.). Pearson.

[3] Scikit-learn Developers. (2023). *Scikit-learn: Machine Learning in Python*. Retrieved from https://scikit-learn.org/stable/

[4] Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python* (1st ed.). O'Reilly Media.

[5] Streamlit Team. (2023). *Streamlit Documentation*. Retrieved from https://docs.streamlit.io/

[6] Zhang, H., & Liu, C. (2019). "A Comparative Study of Text Classification Algorithms for Spam Detection," *Journal of Information Science and Engineering*, vol. 35, no. 4, pp. 789–804.

[7] Almeida, T. A., & Hidalgo, J. M. G. (2011). "Contributions to the Study of SMS Spam Filtering: New Collection and Results," *Proceedings of the 11th ACM Symposium on Document Engineering*, pp. 259–262.

[8] Pedregosa, F., et al. (2011). "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830.

[9] Loper, E., & Bird, S. (2002). "NLTK: The Natural Language Toolkit," *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, pp. 63–70.

[10] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning* (1st ed.). MIT Press.