**EMAIL SPAM CLASSIFICATION**

Natural Language Processing CBCA275

**Submitted by:**

(E23BCAU0082) YASH GARG  
(E23BCAU0085) Aryan Chaudhary

Submitted to

DR. SHAKSHI SHARMA

A close-up of a logo

Description automatically generated

|  |  |  |
| --- | --- | --- |
| Serial No | content | Page No |
| 1 | Abstract | 1 |
| 2 | Introduction | 1 |
| 3 | Related Work | 2 |
| 4 | Methodology | 2 |
| 5 | Hardware/Software Required | 3 |
| 6 | Experimental Results | 3 |
| 7 | Conclusions | 4 |
| 8 | Future scope | 4 |

**1. Abstract**

This project focuses on building a machine learning-based Email Spam Classification system using Natural Language Processing (NLP). The objective is to differentiate between spam and legitimate emails by analyzing their textual content. The project uses a labeled dataset of emails, which is preprocessed, vectorized, and used to train a classification model. The final model achieves high accuracy and can be integrated into email filtering systems to improve user experience.

**2. Introduction**

Email spam is a significant problem in digital communication, where unsolicited and often harmful messages flood users’ inboxes. This project aims to develop a classifier using NLP techniques to identify spam emails. By analyzing textual data and extracting useful features, the model classifies emails as spam or ham (not spam). This can be a valuable component in maintaining secure and efficient communication systems.

**3. Related Work**

Several machine learning models have been applied to spam classification, including Naive Bayes, Support Vector Machines (SVM), Decision Trees, and deep learning models. Past studies have shown that textual features such as word frequency and TF-IDF values can be effective in distinguishing spam. Open-source datasets like the "SMS Spam Collection" and "Enron Email Dataset" have often been used to train and evaluate these models.

**4. Methodology**

The steps involved in this project are:

1. **Data Collection**: Used a public email dataset with labeled spam and ham messages.
2. **Preprocessing**:
   * Lowercasing
   * Removal of stopwords, punctuation, and special characters
   * Tokenization and stemming
3. **Feature Extraction**:
   * CountVectorizer and TF-IDF Vectorizer to convert text into numerical form.
4. **Model Training**:
   * Algorithms used: Multinomial Naive Bayes, Logistic Regression
   * Performance metrics: Accuracy, Precision, Recall, F1-score
5. **Model Evaluation**:
   * Used confusion matrix and classification report for validation.

**5. Hardware/Software Required**

* **Hardware**: Any standard system with 4GB+ RAM
* **Software**:
  + Python 3.x
  + Jupyter Notebook
  + Libraries: pandas, scikit-learn, nltk, matplotlib, seaborn

**6 Experimental Results.**

* **Accuracy Achieved**: -98%
* **Precision and Recall**: High performance with both Naive Bayes and Logistic Regression
* **Observations**:
  + Naive Bayes performed slightly better on smaller datasets
  + TF-IDF representation gave more accurate predictions compared to CountVectorizer

**7. Conclusions**

The Email Spam Classifier effectively distinguishes spam from legitimate emails with high accuracy. The use of NLP for preprocessing and feature extraction significantly boosts the performance of machine learning models. This project proves the practical application of NLP in cybersecurity and communication systems.

**8. Future Scope**

* Integration with real-time email systems (e.g., Gmail API)
* Deployment as a web or mobile application
* Use of deep learning models like LSTM for even better performance
* Adding multilingual spam detection capabilities

**9.GitHub Link**

https://github.com/yashgarg234/email-spam-classification