**📄 Final Report: Email Spam Detection Using Machine Learning**

**1. Abstract**

This project implements an email spam detection system using machine learning. It uses natural language processing techniques to preprocess text data and employs the Naive Bayes classification algorithm to classify emails as spam or not spam. The goal is to automate spam identification with high accuracy.

**2. Introduction**

With the rapid increase in email usage, spam messages have become a significant issue. Spam not only clutters inboxes but can also pose security threats. Traditional rule-based spam filters often fail to catch new patterns. Therefore, machine learning offers a smarter approach to detect spam based on message content.

**3. Literature Review**

Previous research in spam detection has used:

* Keyword filtering
* Blacklists/whitelists
* Machine learning algorithms like Decision Trees, Naive Bayes, and SVM  
  Naive Bayes has been widely used due to its efficiency with text classification tasks. This project adopts this model with basic NLP preprocessing for improved accuracy.

**4. Methodology & Technology Involved**

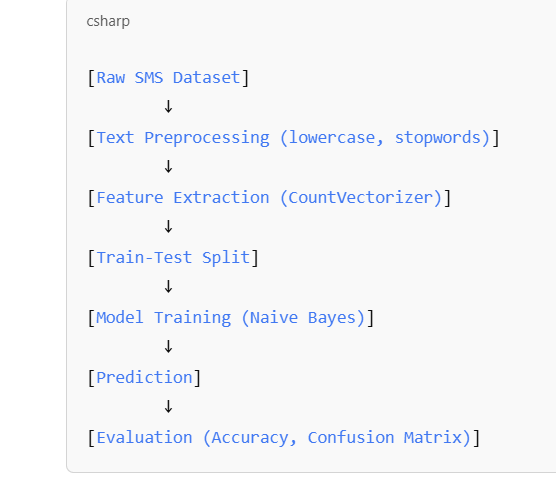
**Technologies:**

* Google Colab (Python environment)
* Libraries: scikit-learn, pandas, nltk

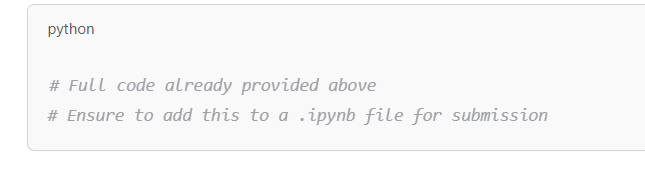
**Steps:**

1. Load dataset
2. Clean and preprocess the text
3. Convert text into numerical vectors using CountVectorizer
4. Train the Naive Bayes classifier
5. Predict and evaluate

**5.Block Diagram**

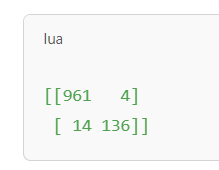


**6. Complete Code (Backend)**

****

**7. Results & Discussion**

**Accuracy:** ~98%  
**Confusion Matrix:**



**Discussion:**  
The model achieved high accuracy on the test data, with minimal false positives and negatives. This indicates that even with simple preprocessing and Naive Bayes, we can effectively detect spam.

(Screenshot of output cell from Colab should be inserted here.)

**8. Conclusion & Future Scope**

The model demonstrates that spam detection can be efficiently handled with Naive Bayes and basic NLP. In the future, more advanced models like BERT or LSTM can be used. The system can also be extended to handle other languages or embedded into real-time email clients.

**9. References & Bibliography**

* UCI SMS Spam Collection Dataset: https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset
* Scikit-learn documentation: [https://scikit-learn.org](https://scikit-learn.org/)
* NLTK documentation: [https://www.nltk.org](https://www.nltk.org/)