Spam Email Classifier Using Machine Learning

1. Introduction

Project Title: Spam Email Classifier Using Machine Learning

Objective:
To develop a machine learning model that classifies emails as 'Spam' or 'Not Spam' using Natural Language Processing (NLP) techniques.

2. Problem Statement

Spam emails are a major issue in the digital world. They waste time, consume storage, and can contain malicious content. The goal is to build a classifier that automatically detects spam emails.

3. Literature Review

- SpamAssassin: Open-source spam filter using rule-based filtering.
- Naive Bayes Classifier: Common for text classification.
- SVM: Used for high-accuracy binary classification.

4. Methodology

4.1 Data Collection:
Dataset from UCI Machine Learning Repository (SMS Spam Collection).

4.2 Data Preprocessing:
- Remove punctuations
- Lowercase text
- Remove stopwords
- Apply stemming
- Tokenize

4.3 Feature Extraction:
- Bag of Words (BoW)
- TF-IDF
- CountVectorizer from sklearn

4.4 Model Building:
- Naive Bayes, SVM, Logistic Regression
- Train/Test split: 80/20

4.5 Performance Metrics:
- Accuracy, Precision, Recall, F1-Score, Confusion Matrix

5. Tools & Technologies

- Python
- Jupyter Notebook/Google Colab
- Libraries: scikit-learn, pandas, numpy, matplotlib, nltk

6. Sample Implementation (Python)

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score

data = pd.read_csv('spam.csv', encoding='latin-1')[['v1', 'v2']]
data.columns = ['label', 'message']
data['label_num'] = data.label.map({'ham':0, 'spam':1})
X_train, X_test, y_train, y_test = train_test_split(data['message'], data['label_num'], test_size=0.2)
vectorizer = CountVectorizer()
X_train_counts = vectorizer.fit_transform(X_train)
X_test_counts = vectorizer.transform(X_test)
model = MultinomialNB()
model.fit(X_train_counts, y_train)
predictions = model.predict(X_test_counts)
print('Accuracy:', accuracy_score(y_test, predictions))
```

7. Results

Model	Accuracy
Naive Bayes	98.7%
SVM	97.5%
Logistic Regression	96.2%

8. Conclusion

Naive Bayes is effective for spam detection due to its probabilistic approach. The project demonstrates successful spam classification using ML techniques.

9. Future Scope

- Web application deployment
- Train on larger datasets
- Integration with email clients for live filtering

10. References

Spam Email Classifier Using Machine Learning

- UCI Machine Learning Repository
- Scikit-learn Documentation
- NLTK Documentation

Prepared by:
Name: K.Tejaswi
Roll No: 24335A0808
Institution: MVGR