|  |  |
| --- | --- |
| Project Title - [EMAIL SPAM CLASSIFICATION USING NLP, DATA AUGMENTATION & MACHINE LEARNING] | Document subtitle  Internship Project Report  Course Title  GLOBAL NEXT CONSULTING INDIA PVT LTD – Internship Program  Abstract  This project develops a machine learning-based spam email classifier using NLP and data augmentation to improve accuracy. The approach enhances recall and F1-score, making spam detection more robust for real-world applications.  AUTHOR NAME  Anmol Dhiman |

Email Spam Classification using NLP, Data Augmentation & Machine Learning

# Acknowledgement

I would like to express my sincere gratitude to InternGeek and the mentors who guided me during this six-week internship. Their support, encouragement, and feedback were invaluable throughout this project. I also thank my peers and colleagues for providing constructive discussions which helped refine my approach.

# Declaration

I hereby declare that this project report titled **“Email Spam Classification using NLP, Data Augmentation & Machine Learning”** is my original work carried out during my six- week internship under the guidance of InternGeek. The results, methodologies, and insights documented here are authentic to the best of my knowledge.

# Executive Summary

Email communication is an integral part of modern organizations. However, the exponential growth of unsolicited spam emails leads to productivity loss, phishing attacks, and potential financial damage. This project aims to develop a spam classification model using machine learning techniques applied on textual data.

We followed a systematic workflow: dataset preprocessing, exploratory data analysis (EDA), class imbalance handling through data augmentation, model training, evaluation, and visualization.

Key highlights:

* Dataset cleaning & preprocessing improved consistency.
* EDA revealed that spam emails are shorter and keyword-rich.
* Data augmentation helped balance the dataset, improving recall.
* Models trained: RandomForest (baseline) and LightGBM (augmented).
* Evaluation metrics showed improved F1-score and recall with augmentation.

This report also discusses business applications, limitations, and future improvements such as using Generative AI for synthetic data generation.

# Introduction

Emails are the backbone of business communication. According to global statistics, billions of spam emails are sent daily, accounting for more than 50% of all email traffic. Traditional filtering methods, such as rule-based systems and blacklists, fail to keep up with evolving

spam strategies.

This motivates the need for machine learning (ML) based spam detection systems. ML algorithms can automatically learn the distinguishing patterns between spam and non- spam (ham) emails.

## Objectives of the Project

* 1. Preprocess email dataset for ML readiness.
  2. Perform exploratory data analysis to identify patterns.
  3. Apply augmentation to balance spam/ham classes.
  4. Train ML models (RandomForest / LightGBM).
  5. Compare baseline vs augmented performance.
  6. Visualize evaluation results.

The aim is to not just build a spam filter, but also highlight the importance of balanced datasets in classification tasks.

# Literature Review / Background

## Traditional Methods of Spam Detection

* Keyword-based filters (e.g., detecting words like “win”, “lottery”).
* Blacklist-based detection (blocking emails from suspicious domains).
* Rule-based heuristics.

These methods are limited, as spammers constantly evolve tactics.

## Machine Learning Approaches

* **Naïve Bayes Classifier**: Early models used word probabilities.
* **Support Vector Machines (SVM)**: Effective in high-dimensional feature spaces.
* **Decision Trees & RandomForest**: Ensemble methods improving robustness.
* **Deep Learning (LSTMs, CNNs)**: More recent approaches using neural nets.

## Why NLP is Crucial

Spam detection is fundamentally a Natural Language Processing (NLP) problem. The ability to extract features such as word frequency, n-grams, and semantic meaning is crucial for classification accuracy.

# Dataset Overview

The dataset was provided in a compressed ZIP archive. It contains email texts and their associated labels (spam/ham).

* **Text column**: Contains the email content.
* **Label column**: Binary values (0 = Ham, 1 = Spam).
* **Size**: ~5000 records after preprocessing.

## Imbalance Problem

* Ham emails (legitimate) are significantly more frequent than spam.
* This imbalance biases models towards predicting ham more often.

## Sample Data

Text: "Free money now! Claim your prize", Label: Spam

Text: "Please find attached the meeting agenda", Label: Ham

# Data Preprocessing

The dataset was cleaned and transformed before modeling. Steps include:

* 1. **Removing duplicates** to avoid bias.
  2. **Handling missing values** by dropping nulls.
  3. **Label encoding** (Ham=0, Spam=1).
  4. **Feature engineering**: Added a text\_length feature.
  5. **Standardization**: Converting all text to lowercase.
  6. **Tokenization**: Splitting text into words.
  7. **Stopword removal** to eliminate common irrelevant words.

This ensured consistent, machine-readable input for NLP models.

# Exploratory Data Analysis (EDA)

EDA provided critical insights into the dataset:

* **Class Distribution**: Countplot showed spam ≈ 20%, ham ≈ 80%.
* **Text Length**: Spam emails generally shorter.
* **Word Clouds**: Spam had words like “win, offer, money”, while ham showed conversational terms.

## Visualizations

* Figure 1: Bar chart of class distribution.
* Figure 2: Histogram of email text length.
* Figure 3: WordCloud for spam messages.
* Figure 4: WordCloud for ham messages.

These visuals confirmed the imbalance and linguistic differences.

# Data Augmentation

Given the class imbalance, we applied simple augmentation techniques on spam emails:

* **Random word shuffle**: Rearranging words randomly.
* **Word removal**: Dropping random words to create variations.

This generated synthetic spam samples, increasing the minority class count.

**Impact**: Training data became more balanced, reducing model bias towards ham.

# Modeling Approach

Two pipelines were implemented:

* 1. **Baseline Pipeline**
     + TF-IDF Vectorizer (n-grams up to 2, max features=5000).
     + RandomForest / LightGBM classifier.
  2. **Augmented Pipeline**
     + TF-IDF Vectorizer.
     + Classifier trained on augmented dataset.

The use of pipelines ensures reproducibility and modularity.

# Model Evaluation

Models were evaluated on unseen test data (20% split).

## Metrics Used

* **Precision**: Correctly identified spam out of predicted spam.
* **Recall**: Ability to catch all actual spam.
* **F1-score**: Balance of precision and recall.
* **ROC-AUC**: Discrimination power between classes.
* **PR-AUC**: Useful for imbalanced datasets.

## Results

* **Baseline Model**: High precision but lower recall (missed some spam).
* **Augmented Model**: Recall improved significantly while maintaining precision.

## Visualizations

* Figure 5: ROC curves for baseline vs augmented.
* Figure 6: PR curves comparison.
* Figure 7: Confusion matrices.

# Business Impact & Applications

Effective spam detection ensures:

* **Security**: Reduces phishing and malware risks.
* **Productivity**: Saves employee time.
* **Cost savings**: Avoids financial fraud via spam.
* **User trust**: Stronger confidence in email communication.

Industries like banking, healthcare, and IT heavily benefit from robust spam filters.

# Limitations & Future Scope

## Limitations

* Augmentation used simple methods only.
* Dataset limited in size and diversity.
* Some spam emails may mimic ham closely.

## Future Scope

* Use of **Generative AI** (GPT, T5) for realistic spam generation.
* **Explainability (SHAP, LIME)** for model transparency.
* Deploying as **real-time API** for enterprises.

# Conclusion

The project successfully demonstrated how NLP and machine learning can be applied to spam detection. Augmentation significantly improved recall and F1-score, proving its value in imbalanced datasets.

This framework can be expanded with advanced augmentation and deployed in enterprise systems.

# Appendix

## Libraries Used

* pandas, numpy, matplotlib, seaborn
* sklearn (TF-IDF, RandomForest)
* lightgbm (optional)
* wordcloud (visualization)

## Sample Code Snippet

pipeline = Pipeline([

('tfidf', TfidfVectorizer(max\_features=5000, stop\_words='english', ngram\_range=(1,2))),

('clf', RandomForestClassifier(n\_estimators=100, random\_state=42))

])

pipeline.fit(X\_train, y\_train)

## Outputs (Screenshots)

* Countplot of classes.
* WordClouds.
* ROC & PR Curves.
* Confusion matrices.

# References

* 1. Research papers on spam detection using ML.
  2. Kaggle datasets and discussions.
  3. Scikit-learn documentation.
  4. LightGBM official docs.
  5. WordCloud Python library.