Final Report: Spam Classification using TF-IDF and Logistic Regression

Baseline vs Hyperparameter Optimization across three datasets

Abstract. We evaluate a classical NLP pipeline (TF-IDF + Logistic Regression) for spam detection. We compare a reproducible baseline model to a hyperparameter-optimized model using the same train/test split strategy and metrics. Experiments are run on three datasets with different text styles and scales. We focus on spam recall and interpretability using confusion-matrix based error analysis.

Key contribution. A consistent baseline vs tuned comparison shows that optimization yields clear gains on small and medium datasets and smaller-but-still-real gains on a large dataset, with improved spam recall and fewer false negatives.

Deliverables. This report summarizes the final results produced by the accompanying Jupyter notebooks for Dataset 1 (SMS), Dataset 2 (Ling-Spam), and Dataset 3 (Large Email).

1. Problem and Motivation

Spam has evolved from simple advertisements to phishing and social-engineering attacks. Rule-based keyword filters fail against paraphrasing and evolving tactics. A learning-based classifier can capture statistical patterns in text. Because missed spam can be costly, we report spam-class recall and inspect false negatives.

2. Datasets

Dataset 1 (SMS): short informal messages; compact vocabulary; harder to detect when spam is short.
Dataset 2 (Ling-Spam): email-style text; longer context; more structured patterns.
Dataset 3 (Large Email): large modern dataset; used to evaluate scalability and diminishing returns of tuning.

3. Methodology

Pipeline: TF-IDF vectorization followed by Logistic Regression. The baseline uses default settings. Hyperparameter optimization uses RandomizedSearchCV (3-fold) over TF-IDF settings (n-gram range, min_df, max_df) and Logistic Regression regularization (C) and class weighting when applicable. Metrics: accuracy, precision, recall, F1, and confusion matrix.

Optimized settings (best found):
- **Dataset 1 (SMS):** ngram_range=(1,2), min_df=5, max_df=0.8, C=10
- **Dataset 2 (Ling-Spam):** ngram_range=(1,2), min_df=5, max_df=0.8, class_weight=balanced, C=10
- **Dataset 3 (Large Email):** ngram_range=(1,2), min_df=5, max_df=0.8, C=10

4. Results Summary

Table 1 compares baseline vs tuned performance. Spam precision/recall/F1 correspond to the spam class (label 1) as reported by the notebooks. Confusion matrices are shown as TN/FP/FN/TP.

Dataset	Model	Accuracy	Spam Precision	Spam Recall	Spam F1	TN	FP	FN	TP
Dataset 1 (SMS)	Baseline	0.968	1.000	0.760	0.860	966	0	36	113
Dataset 1 (SMS)	Tuned	0.982	0.990	0.870	0.930	965	1	19	130
Dataset 2 (Ling-Spam)	Baseline	0.967	1.000	0.800	0.890	483	0	19	77
Dataset 2 (Ling-Spam)	Tuned	0.993	0.990	0.970	0.980	482	1	3	93
Dataset 3 (Large Email)	Baseline	0.970	0.960	0.970	0.970	3254	114	79	2928
Dataset 3 (Large Email)	Tuned	0.990	0.990	0.990	0.990	3330	38	28	2979

Main observation: Hyperparameter optimization improves spam recall and reduces false negatives across all datasets. Gains are largest on Dataset 2; Dataset 3 shows smaller relative improvement due to already-strong baseline performance at scale.

5. Dataset-Level Error Analysis

Confusion matrices highlight error types. In spam filtering, false negatives (FN) are the most critical because spam that passes the filter can cause harm.

Dataset 1 (SMS)

Baseline confusion matrix

	Pred Ham (0)	Pred Spam (1)
True Ham (0)	966	0
True Spam (1)	36	113

Tuned confusion matrix

	Pred Ham (0)	Pred Spam (1)
True Ham (0)	965	1
True Spam (1)	19	130

Dataset 2 (Ling-Spam)

Baseline confusion matrix

	Pred Ham (0)	Pred Spam (1)
True Ham (0)	483	0
True Spam (1)	19	77

Tuned confusion matrix

	Pred Ham (0)	Pred Spam (1)
True Ham (0)	482	1
True Spam (1)	3	93

5. Dataset-Level Error Analysis (continued)

Dataset 3 is large-scale, so even small percentage improvements correspond to many messages. Tuning reduces both FP and FN compared to baseline.

Dataset 3 (Large Email)

Baseline confusion matrix

	Pred Ham (0)	Pred Spam (1)
True Ham (0)	3254	114
True Spam (1)	79	2928

Tuned confusion matrix

	Pred Ham (0)	Pred Spam (1)
True Ham (0)	3330	38
True Spam (1)	28	2979

Typical failure patterns (qualitative):
- Short spam without obvious spam keywords can look like normal messages, especially in SMS.
- Promotional language in legitimate emails can create false positives.
- TF-IDF is surface-form based; it can miss semantic intent when wording is subtle or heavily obfuscated.

6. Conclusion and Future Work

TF-IDF + Logistic Regression provides strong baseline performance for spam detection. Hyperparameter optimization consistently improves spam recall and reduces false negatives. Improvements are largest on small/medium datasets where defaults are not ideal; on large-scale data, the baseline is already strong and tuning yields smaller relative gains but still meaningful absolute reductions in errors.

Future work: Evaluate contextual embeddings (e.g., transformer-based features) or hybrid approaches for semantic understanding, and perform deeper error analysis by grouping failures (URL-heavy spam, short spam, phishing-like language).

Reproducibility: Full code, notebooks, and instructions are provided in the GitHub repository accompanying this submission.

References (selected): Androutsopoulos et al. (Ling-Spam); standard TF-IDF and Logistic Regression references; course materials in nlp-undergrad repository.