Spam Detection using NLP — Summary & Insights

🚀 Project Objective

Build a machine learning model to accurately classify SMS messages as **spam** or **ham (not spam)** using Natural Language Processing (NLP) techniques, with careful consideration of **class imbalance**.

🔪 Phase 1: Prototyping & Experimentation (File1)

Initial experiments were conducted on a **smaller subset** of the dataset to efficiently evaluate various pipelines.

⚖️ Key Challenge:

- The dataset is highly **imbalanced** (~87% ham vs ~13% spam)
- Using **accuracy** as a metric would be misleading, since predicting only "ham" yields high accuracy without real predictive power

✅ Metric Selection:

The **F1-Score** was selected, especially for the minority spam class.
It balances **precision** (avoiding false positives) and **recall** (catching true spam), making it ideal for imbalanced classification.

🔮 Pipelines Compared:

Several text processing and model combinations were explored:

- Variations included **CountVectorizer**, **TF-IDF**, and different classifiers

✅ Best Performer Identified:

- **TF-IDF Vectorizer + Logistic Regression**

📈 Phase 2: Final Model on Full Dataset (File2)

After identifying the optimal pipeline, it was retrained on the **entire dataset** for full-scale evaluation and deployment.

🧱 Final Pipeline:

```
Pipeline([
  ('tfidf', TfidfVectorizer()),
  ('clf', LogisticRegression())
```

])

🔍 **Evaluation Approach:**

- **GridSearchCV** was used for hyperparameter tuning
- Evaluation was performed using classification_report with **Precision**, **Recall**, and **F1-score**

Typical results:

Class	Precision	Recall	F1-score
Ham	~0.97	~0.99	~0.98
Spam	~0.95	~0.89	~0.92

🧠 **Interpretation:**

- High **precision for spam**: Few false alarms
- Strong **recall**: Most spam messages are detected
- The model performs reliably and balances both classes well

📁 **Final Step: Model Persistence**

The trained pipeline was saved using joblib, making it ready for production or inference with minimal overhead.

📌 **Final Summary:**

A practical, two-phase workflow was followed — beginning with prototype evaluation on a sample dataset and finalizing with full-scale model training and tuning. The resulting pipeline (TF-IDF + Logistic Regression) demonstrated strong performance, especially on the imbalanced spam class, and was saved for future use. This approach reflects best practices in model selection, evaluation, and deployment readiness.