📊 Spam Classification on Enron Database

This project builds a **spam classifier** on the Enron Email dataset.
It demonstrates a complete pipeline from **text preprocessing** to **custom ML models** built from scratch, and finally **ensemble learning**.

📂 Dataset

- **Source:** Enron Spam dataset
- **Total samples:** 33,716
- **Classes:** `Ham (0)` and `Spam (1)`
- **Train/Test split:** 80/20
  - Training samples: 26,972
  - Testing samples: 6,744

🛠️ Preprocessing

1. **Text Cleaning**
   - Removed non-alphanumeric characters
   - Converted to lowercase
   - Tokenization, stopword removal, stemming (PorterStemmer)
2. **Feature Extraction**
   - TF-IDF with max 5000 features
   - Transformed `processed_text` into numerical feature vectors

👨‍💻 Baseline Classifiers (From Scratch)

1. Naive Bayes

- Implements:
  - `fit()` → learns class priors and likelihoods with Laplace smoothing
  - `predict()` → class labels
  - `predict_proba()` → class probabilities (for ensembles)
  - `score()` → accuracy

2. Logistic Regression

- Implements:
  - Gradient descent optimization
  - Sigmoid activation
  - `predict()` (threshold 0.5), `predict_proba()`, `score()`

🤝 Ensemble Models

1. Voting Ensemble

- Combines Naive Bayes and Logistic Regression
- Supports:
  - **Hard Voting:** majority vote on predicted labels
  - **Soft Voting:** averages predicted probabilities

2. Stacking Ensemble

- Base Models: Naive Bayes + Logistic Regression
- Meta-Model: Logistic Regression (from scratch)
- Uses base classifiers' **probabilities** as input features

📈 Evaluation Metrics

Models evaluated using:
- **Accuracy**
- **Precision**
- **Recall**
- **F1 Score**
- **Confusion Matrix**

✅ Results (Example)

Model	Accuracy	Precision	Recall	F1 Score
Naive Bayes	0.9816	0.9651	1.00	0.9822
Logistic Regression	0.9837	0.9689	1.00	0.9842
Voting Ensemble (Soft)	0.9844	0.9703	1.00	0.9849
Stacking Ensemble	0.9838	0.9692	1.00	0.9843

(Values here are placeholders — replace with your actual run outputs.)

🔍 Insights

- Logistic Regression outperformed Naive Bayes as a single model.
- Soft Voting improved performance slightly by leveraging both models.
- Stacking Ensemble gave the best results by learning when to trust each base classifier.

🚀 Future Work

- Hyperparameter tuning (learning rate, epochs, TF-IDF features)
- Add deep learning baselines (RNN, LSTM, Transformer)
- Use cross-validation for more robust evaluation
- Deploy model as an API with **FastAPI**
