📌 Introduction (as a student)

> “As a student passionate about cybersecurity and SOC operations, I built this project to simulate how phishing emails are detected using Natural Language Processing (NLP) and Machine Learning. Since phishing is the #1 attack vector in modern cyber incidents, this project helped me understand how SOC teams triage and analyze suspicious emails daily.”

🛠️ Step 1: Collect Dataset

- Used **Kaggle Phishing Email Dataset** + **Enron Email Dataset** for legit emails.
- Labeled them as **phishing (1)** and **legitimate (0)**.

🛠️ Step 2: Preprocess Data (NLP)

- Cleaned emails → removed HTML tags, punctuation, stopwords.
- Converted to features using:
  - **Bag of Words (BoW)**
  - **TF-IDF Vectorizer**

🛠️ Step 3: Train Classifier

- Models used:
  ✅ **Naive Bayes** (simple baseline).
  ✅ **Logistic Regression** (stronger classifier).
- Performed **Train/Test split (80/20)**.
- Evaluated with **accuracy, precision, recall, F1-score, confusion matrix**.

🛠️ Step 4: (Optional Simulation – SOC Perspective)

- From **Kali Linux**, crafted a test phishing-style email.
- Sent it to a **Windows test mailbox**.

- Extracted email text → ran through the trained model.
- Model flagged it as **phishing**.

🎯 Learning Outcome (internship focus)

By doing this project, I gained:

- ✅ Hands-on NLP preprocessing skills.
- ✅ Practical exposure to ML in cybersecurity.
- ✅ Understanding of SOC phishing triage workflow.
- ✅ Awareness of how SIEM/EDR tools might integrate AI-based phishing detection.

📌 Why This Matters for a SOC Internship

- Phishing is the **most common SOC ticket type**.
- SOC analysts must analyze email **headers, body, and links**.
- This project shows I can **apply technical + analytical thinking** to real SOC problems.
- It bridges **AI + cybersecurity** → a modern skill highly valued in internships.

---

phishing_lab/
│── collect_dataset.py
│── step_train_model.py
│── README.md
│── requirements.txt

Phishing Email Detection

This project collects a dataset of emails (legitimate + phishing), processes them, and trains a Logistic Regression model to classify phishing attempts.

📌 Steps

1. **Dataset Collection**
   Run `collect_dataset.py` to fetch and preprocess the dataset.
2. **Model Training**
   Run `step_train_model.py` to train the Logistic Regression model.

📊 Sample Results

Confusion Matrix

Classification Report

Class	Precision	Recall	F1-score	Support
Legit	0.75	1.00	0.86	3
Phishing	1.00	0.50	0.67	2
Accuracy			**0.80**	5

🔹 Meaning:

- Out of 3 legit emails → all 3 correctly classified ✅
- Out of 2 phishing emails → 1 detected, 1 missed ❌
- Overall accuracy = **80%**

⚠️ Notes

- Training may take a long time depending on dataset size.
- The dataset is **not included** in this repo (too large 100000+).
- Example results above are from one training run only 5 datasets.