**Analyzing Phishing Email Language Patterns Through Word Frequencies and Visualization**

**🎯 Objective**

This project explores how phishing emails differ from legitimate emails based on word usage. By cleaning and analyzing text from a labeled dataset, we identify frequent terms used in phishing attempts and visualize their prevalence using word clouds. This linguistic insight can help inform the design of phishing detection systems and raise user awareness of common red flags in malicious communications.

**📁 Dataset**

* **Name:** Phising\_email.csv
* **Source:** [Kaggle phishing email dataset](https://www.kaggle.com/datasets/educatorsrlearners/phising-email)
* **Key Features:**
  + **Email Text:** The content of each email.
  + **Email Type:** Label indicating whether it's a 'Phishing Email' or 'Safe Email'.

**⚙️ Tools Used**

* **Platform:** Google Colab
* **Libraries:**  
  pandas, scikit-learn, nltk, wordcloud, matplotlib

**🧹 Step 1: Data Cleaning and Preprocessing**

* Converted text to lowercase
* Removed punctuation
* Removed stop words (e.g., "the", "and")
* Tokenized the content into individual words

A screen shot of a computer program

AI-generated content may be incorrect.  
  
**Why this matters:**  
Cleaning the data removes noise and helps the model focus on meaningful features. Words like “the,” “and,” and punctuation don’t help the classifier, so they’re removed.

**🧠 Step 2: Word Frequency Analysis**

We extracted and ranked the most frequently used words in phishing vs. safe emails:

A computer screen shot of a program

AI-generated content may be incorrect.  
  
🎯 **Why this matters:**  
Phishing emails often contain words like **"free," "email," "get,"** and **"information"**. Legitimate ones tend to have organizational names or context-rich terms like **"university," "language,"** and **"enron"**.

**🌥️ Word Cloud Visualization**

**Word Cloud of Phishing Emails**

A screenshot of a computer program

AI-generated content may be incorrect.  
  
A close up of words

AI-generated content may be incorrect.  
  
Word Cloud of Safe Emails  
  
A screenshot of a computer program

AI-generated content may be incorrect.  
  
A close up of words

AI-generated content may be incorrect.  
  
  
**Step 3: Machine Learning Classification**

We used a basic **Naive Bayes** classifier with TF-IDF vectorization:

A screenshot of a computer program

AI-generated content may be incorrect.  
  
**Why This Matters**

Phishing emails are crafted to manipulate users into taking harmful actions, often through deceptive language. By identifying and analyzing the most common words used in phishing attempts, we gain critical insights that can strengthen defenses in several ways:

* **Keyword-Based Detection**: Understanding which terms frequently appear in phishing emails (e.g., “verify,” “account,” “click”) allows cybersecurity teams to develop keyword-based filters or heuristics to catch suspicious messages before they reach the user.
* **Training & Awareness**: Users often fall victim to phishing because they don’t know what to look for. Highlighting common phishing language can be integrated into security training, making employees more alert when reading emails containing suspicious terms.
* **Enhancing AI Models**: Natural language processing (NLP) models can benefit from known phishing word patterns. These words can be prioritized as features, improving the performance of email classification models.
* **Real-Time Monitoring**: In enterprise environments, monitoring communication channels for high-frequency phishing terms can serve as an early warning system for ongoing attacks or breaches.

In short, phishing word frequency analysis provides both tactical and strategic value. It empowers automated systems and end-users alike with better tools to detect and respond to social engineering threats embedded in email content.

**Next Steps**

1. **Remove Non-Informative High-Frequency Tokens**  
   Words like “â” or single-digit numbers (e.g., '1', '2', '3') appeared frequently but don’t contribute to phishing detection. A next step would be refining the preprocessing pipeline to better filter out such noise and improve word frequency relevance.
2. **N-gram Analysis (Phrases)**  
   Move beyond individual words and analyze bigrams or trigrams (e.g., “verify account,” “click here now”). Phishing language often relies on specific phrases rather than just single words.
3. **Topic Modeling (LDA)**  
   Apply Latent Dirichlet Allocation to discover hidden thematic structures in phishing and legitimate emails. This can help identify types of scams (e.g., fake shipping notices vs. fake login pages).
4. **Compare Across Datasets**  
   Validate your findings by applying the same analysis to different phishing email datasets. Consistent word patterns across sources would strengthen confidence in your results.
5. **Time-Based Trend Analysis**  
   If timestamps are available, examine how phishing language evolves over time. This could reveal trends like emerging scams (e.g., COVID-related phishing in 2020).
6. **Incorporate External Word Lists**  
   Compare your findings with external resources like the Anti-Phishing Working Group’s keyword lists or cybersecurity blacklists to find overlaps or gaps.
7. **Build a Live Word-Scanner Tool**  
   Use the frequency data to create a basic tool or browser extension that highlights suspicious words in incoming emails, helping users recognize red flags in real time.
8. **Feed Keywords into a Classification Pipeline**  
   Use the top phishing words as features in a lightweight model to test if a keyword-only approach can still achieve decent classification performance.
9. **Visualize Word Frequency Overlap**  
   Create a Venn diagram or bar chart to show overlapping words in both phishing and legitimate emails, and highlight uniquely phishing-specific ones.
10. **Collaborate with Security Teams**  
    Share findings with security analysts or IT departments to see how word frequency analysis can be integrated into their current spam filtering or threat detection systems.