Final Project Writeup: NLP Social Engineering Classification

# 1. Task and Motivation

This project focused on a multi-class text classification task. The goal was to categorize recruiter-style social engineering conversations based on the likely success of the attack. Each interaction was labeled as one of three categories: failure, partial, or successful. The classification was based on how much sensitive information the target disclosed in response to the recruiter’s message.  
  
This task is particularly relevant in cybersecurity, where language-based manipulation plays a central role in phishing and other social engineering attacks. Being able to predict how a conversation may unfold—based purely on textual signals—could improve early detection systems and human awareness of manipulation tactics.

# 2. Methods and Data

The dataset consisted of 2,000 conversations generated using GPT-4. Each sample simulated a recruiter engaging a target on a professional platform like LinkedIn. Messages included multiple lines of back-and-forth dialogue.  
  
Each conversation was manually labeled as one of the following outcomes:  
- Successful: The target gave away sensitive information or agreed to an external action.  
- Partial: The target responded with mild interest but without revealing critical data.  
- Failure: The target ignored, declined, or disengaged.  
  
Example (Partial):  
Recruiter: 'Hi Jamie, I’m recruiting for a stealth SOC project. Interested?'  
Target: 'Maybe. Send more info.'  
Label: Partial  
  
Preprocessing involved cleaning label text (e.g., 'fail' → 'failure'), removing missing values, and encoding labels using scikit-learn’s LabelEncoder. Tokenization was handled via TF-IDF for the Naive Bayes baseline and with RoBERTaTokenizer for the transformer model. Class weights were computed to account for label imbalance.  
  
Two modeling approaches were used:  
- Naive Bayes: Used TF-IDF vectorization and MultinomialNB classifier.  
- RoBERTa: Used Hugging Face's Transformers library to fine-tune a pretrained RoBERTa-base model for sequence classification.  
  
RoBERTa was trained using Hugging Face's Trainer API with a weighted cross-entropy loss to mitigate label imbalance. Training was conducted over 5 epochs with a learning rate of 2e-5. The model was run on Apple Silicon (MPS) and monitored with live training loss logging.

# 3. Evaluation and Results

Evaluation was performed using Accuracy and Weighted F1 Score.  
  
Initial results with Naive Bayes on a small dataset (500 samples):  
- Accuracy: ~40%  
- F1 Score: ~0.38  
  
Results after scaling to 2,000 samples:  
- Accuracy: ~48%  
- F1 Score: ~0.45  
- With 5-fold Cross-Validation: Accuracy ~52–54%, F1 ~0.50  
  
RoBERTa Transformer Model:  
- Accuracy: 85.5%  
- Weighted F1 Score: 0.85  
- Strongest performance on clear-cut successful/failure samples  
- Borderline cases (partial vs. successful) were the most commonly misclassified  
  
Challenges included:  
- Distinguishing partial vs. successful where intent was ambiguous  
- Managing memory on Apple Silicon during model fine-tuning  
- Secure handling of API keys via environment variables and .gitignore

# 4. Changes Since the Presentation

- Integrated 5-fold cross-validation for Naive Bayes to stabilize results  
- Improved tokenizer settings and class-weight balancing in RoBERTa  
- Re-trained final RoBERTa model and confirmed metrics on new validation fold  
- Updated the slide deck to include training loss visualization and a RoBERTa explanation slide  
- Created and committed a .gitignore file to exclude .env and secure API keys in GitHub repo