

# Smart Lead Gen: A Generative AI for Hyper-Personalized Professional Outreach

- Final Project report.
- Team: Group 7
- Course: DS and AI Lab

Content:

1. Problem Definition
2. Dataset Preparation
3. Model architecture
4. Model Training
5. Model Evaluation
6. Final Models

## 1. Problem Definition

In professional sales, volume is a poor substitute for value. A staggering 98% of outreach emails are ignored, with generic cold outreach campaigns yielding a response rate of less than 2%. This highlights a critical disconnect: the methods used to establish professional connections are fundamentally broken.

The core problem is the manual, time-consuming, and non-scalable process of crafting hyper-personalized outreach messages. This operational bottleneck forces B2B sales teams into a difficult trade-off:

- **Mass Outreach:** Deploying automated, low-effort campaigns that suffer from extremely low engagement, damage brand perception, and yield a poor signal-to-noise ratio.
- **Manual Personalization:** Investing significant manual hours into researching and writing bespoke messages, a high-conversion strategy that severely limits reach, productivity, and scalability.

This inability to scale personalized outreach results in a substantial opportunity cost, leaving countless qualified leads and significant revenue potential untapped.

### Current Landscape & Identified Gaps

A review of the market reveals that while tools exist for parts of the outreach process, no solution effectively addresses the core personalization challenge.

- **Existing Platforms:** Sales automation tools (HubSpot, Outreach.io) automate sequences using static, token-based templates. Lead identification platforms (LinkedIn Talent Solutions, Apollo) help find prospects but do not assist in crafting the message.
- **Generalist LLMs:** Foundational models like GPT-4 can generate text but produce generic, low-impact outputs without significant prompt engineering and structured data integration.

Research in NLP confirms that blending structured data (job history, recent posts, company updates) with generative models leads to more relevant and context-aware outputs, boosting reply rates by 3-5x. Despite this, critical gaps remain.

### **Identified Gaps & Competitive Weaknesses**

*Lack of an Integrated Data-to-Draft Pipeline:* Current solutions are fragmented. While emerging competitors like **Jeeva.ai** and **trycoolie.ai** attempt to bridge this gap, their pipelines are not truly end-to-end, their message personalization is inefficient, and they lack a significant market presence in India.

*Poor Signal Extraction:* Automated systems fail to leverage rich, unstructured data signals from professional profiles, such as a prospect's recent posts, interviews, or company announcements.

*Absence of Domain-Specific LLMs:* No prominent generative model has been specifically fine-tuned for the domain of professional outreach with clear commercial intents like "book a demo" or "initiate a discovery call."

### **Opportunity for Innovation**

These gaps present a clear opportunity. By developing a proprietary, fine-tuned generative AI model that takes structured professional data + outreach intent as input, we can build an end-to-end system that produces hyper-personalized, conversion-optimized outreach drafts at scale. This will empower B2B sales organizations to move beyond ineffective "spray and pray" tactics and unlock the full potential of meaningful, authentic engagement.

### **Objective:**

To develop and deploy a **fine-tuned generative AI model** that autonomously creates hyper-personalized, context-aware B2B sales messages optimized for high engagement and conversion.

## **Beneficiaries & Impact**

This solution is designed to deliver a direct and measurable impact on professionals who rely on outreach to achieve their goals.

## **Target Users:**

- Sales Development Representatives (SDRs) & Account Executives: The primary beneficiaries who will use this tool to connect with potential customers.
- Recruiters & Talent Acquisition Specialists: To engage with high-value candidates.
- Entrepreneurs & Founders: For networking, fundraising, and partnership development.
- Job Seekers & Freelancers: To connect with hiring managers and potential clients.

## **Impact & Value Proposition:**

- Increased Efficiency: Drastically reduce the time spent on manual research and message crafting, allowing users to focus on building relationships.
- Improved Effectiveness: Significantly increase engagement, reply rates, and conversion rates by delivering messages that resonate with the recipient.
- Scalability: Enable high-quality, personalized outreach at a scale that is impossible to achieve manually.
- Enhanced Brand Perception: Foster more meaningful and professional first impressions, strengthening the user's personal or company brand.

## 2. Dataset Preparation

### Datasets Used

1. (MBTI) Myers-Briggs Personality Type Dataset
2. Pandora - Big 5 Dataset
3. Enron Email Dataset

#### (MBTI) Myers-Briggs Personality Type Dataset

[Link](#)

This dataset offers a valuable resource for the development and fine-tuning of personality classification models. It comprises over 8,600 distinct samples, each containing two primary components: a labeled four-letter MBTI personality type and a corresponding corpus of the user's 50 most recent posts from the PersonalityCafe forum. This structure provides a direct mapping between an individual's written communication style and their classified personality profile, making it an ideal feature set for training a model to predict personality based on text analysis.

#### Pandora - Big 5 Dataset

[Link](#)

The Pandora dataset is a comprehensive collection of Reddit comments ideal for fine-tuning personality classification models. It contains data from over 10,000 users, with 1,600 of those users labeled with the well-established Big Five personality model. What makes this dataset particularly powerful is that it not only includes the raw text of user comments but also demographic information such as age, gender, and location.

For the purposes of our project, the Big Five labels within this dataset will be reparametrized to correspond with the MBTI personality types. This strategic conversion will allow us to leverage

PANDORA's extensive text and demographic data for the direct training and refinement of our MBTI-specific classification model, helping to create a more nuanced and accurate predictive tool.

## **Enron Email Dataset**

[Link](#)

The sales message generation model will be fine-tuned using the Enron email dataset, a collection of approximately 500,000 emails exchanged within the Enron corporation. This dataset is an invaluable asset as it comprises real-world business communications, primarily from senior management. The inherently professional tone and structure of these emails provide a rich foundation for training our model. By learning from this corpus, the model will develop a nuanced understanding of corporate communication styles, appropriate language, and persuasive techniques used in a professional context. This will enable it to generate highly relevant and effective sales messages that resonate with a business audience.

## **Dataset Preprocessing**

## **Email Message Generation**

1. Keyword Filtering: Started by filtering the original, massive email dataset (containing over 500,000 entries) to create a smaller, more relevant subset. This was done by searching the subject and body of each email for a list of specific outreach and business-related keywords like "meeting," "proposal," and "opportunity."
2. Robust Text Cleaning: Developed a multi-stage cleaning function that was applied to the body of each filtered email to:
  - Removed large, noisy blocks of text (like --- Forwarded by --- and ----- Original Message -----).

- Stripped out individual header lines (To:, From:, Subject:, etc.) that were mixed in with the body content.
  - Isolated the clean, natural language of the core message, which was then saved into a new cleaned\_body column.
3. Final CSV Creation: The final step was to prepare the output file. Selected the most important columns (subject, from, to, date, and our new cleaned\_body) from the processed data and saved them into a final, ready-to-use CSV file named filtered\_cleaned\_enron.csv.

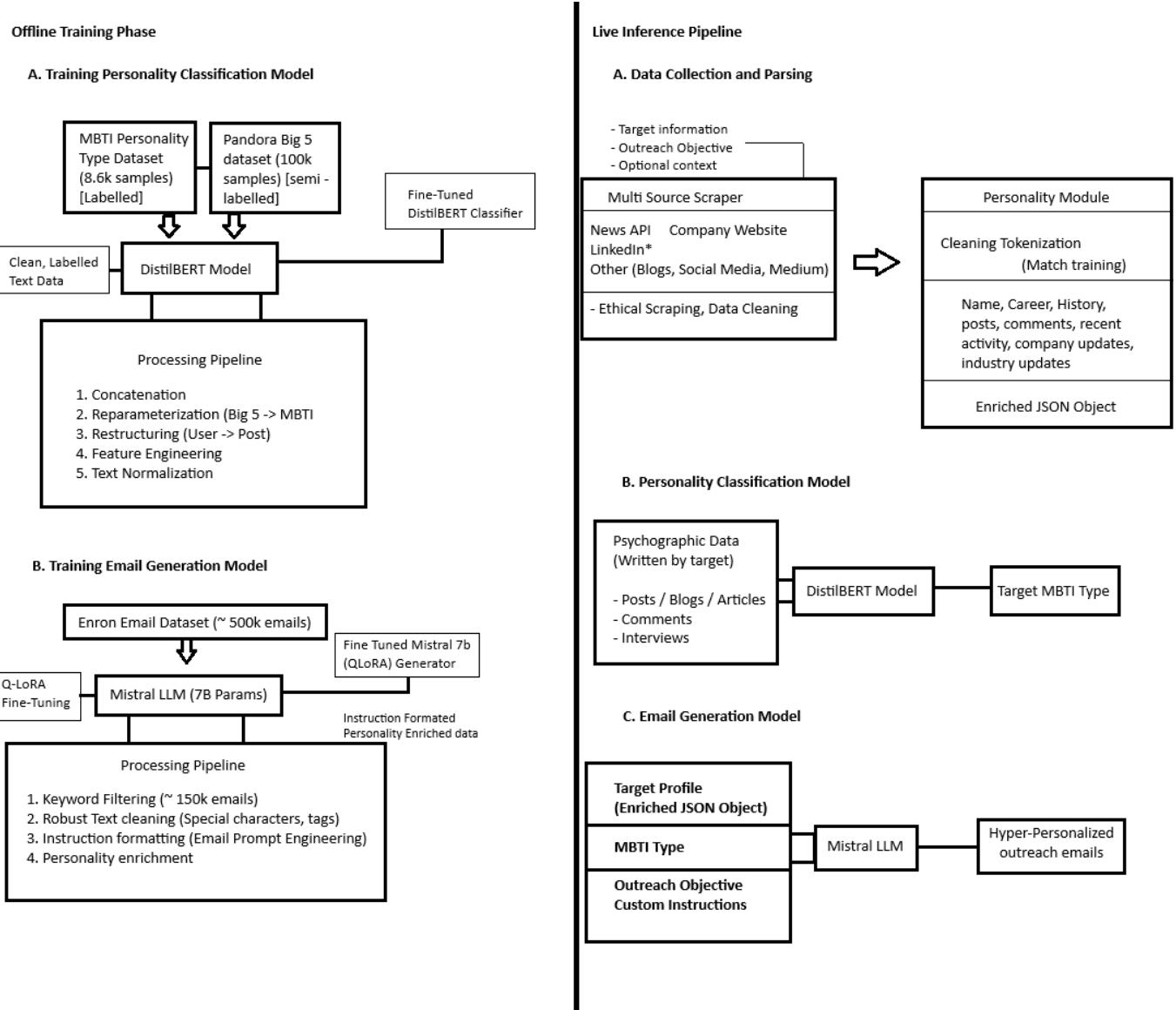
## Personality Classification

The distinct personality classification datasets, the MBTI personality type dataset and Pandora big 5 dataset have been concatenated into a unified corpus for analysis. The Pandora Big 5 classifications were processed and transformed into MBTI classifications based on each entry's MBTI dimensions: Introversion vs. Extroversion, Intuition vs. Sensing, Intuition vs. Sensing, Judging vs. Perceiving. The combined dataset was subjected to a multi-stage preprocessing pipeline:

1. Data Restructuring: User posts, originally concatenated by '||' delimiters, were parsed and expanded into individual entries. This transformed the data from a user-centric to a post-centric format.
2. Feature Engineering: The 16-point 'type' label was deconstructed into four binary dimensional features: 'I-E', 'N-S', 'T-F', and 'J-P'.
3. Text Normalization: A cleaning function was systematically applied to each text entry to remove URLs, punctuation, and other non-alphanumeric artifacts, ensuring data uniformity.

### 3. Model Architecture

The project employs a sophisticated, multi-stage model architecture that transforms a simple user request into a hyper-personalized, context-aware, and strategically tailored message. Stage 1 handles data collection and structuring, Stage 2 enriches this data with MBTI insights, and Stage 3 synthesizes available information to generate the final output. This modular design ensures that each step of the personalization process is handled by a specialized, efficient, and robust component, culminating in a system that enables high-quality outreach at scale.



## 4. Model Training

### A. MBTI Personality classification model

The MBTI Classification Model is a specialized text analysis engine designed to infer a lead's personality type (one of 16 Myers-Briggs types) based on their recent professional writing (e.g., LinkedIn posts, articles). This inference serves as the upstream input, determining the tone and strategy of the final generated email.

#### Model Architecture

- **Base Model:** distilbert-base-uncased. This lighter, distilled version of BERT was chosen for low-latency inference while maintaining strong semantic understanding capabilities.
- **Optimization Technique:** Low-Rank Adaptation (LoRA). Instead of fine-tuning the entire model, adapters were attached to specific attention layers (q\_lin, k\_lin, v\_lin), significantly reducing the number of trainable parameters to approximately 824,000 (1.2% of the total model).

#### Dataset Preparation & Filtering

- **Source Data:** A dataset of 8,675 rows (mbti\_train\_data.csv) containing user posts and associated personality labels.
- **Professional Context Filtering:** To ensure the model is relevant for B2B contexts, a custom filtering algorithm was applied.
  - *Heuristic:* A "Professional Score" was calculated based on the presence of business keywords (e.g., "revenue", "roadmap", "quarter", "compliance", "stakeholders") and text length.
  - *Result:* The dataset was reduced to **7,974 high-quality "professional" samples**, filtering out irrelevant casual social media noise.

- **Preprocessing:** Text was normalized by removing URLs, special characters, and converting to lowercase.
- **Tokenization:** Inputs were padded and truncated to a maximum sequence length of **256 tokens**.

## Hyperparameter Configuration

The model was trained using the Hugging Face Trainer with the following specific configurations:

- **LoRA Config:**
  - Rank: 8
  - Alpha: 16
  - Dropout: 0.1
- **Training Arguments:**
  - **Epochs:** 45 (Extensive training duration to maximize convergence on complex personality traits).
  - **Learning Rate:**  $2 \times 10^{-4}$
  - **Batch Size:** 16.
  - **Precision:** Mixed precision (fp16) enabled for GPU acceleration.

## Performance Results

- **Training Convergence:** The model achieved near-perfect fitting on the training set, with an accuracy of **99.89%** and an F1-Macro score of **0.999**.
- **Generalization (Test Set):** On a held-out test set of 51 distinct samples, the model achieved an accuracy of **49.02%**.
  - **Analysis:** While significantly higher than random chance (6.25% for 16 classes), the gap between training and test performance indicates the high complexity of personality inference and suggests the model relies heavily on the specific linguistic markers present in the training distribution.

## B. Email Generation Model

The core generation engine is a fine-tuned iteration of Mistral-7B-Instruct-v0.2. The model was optimized specifically to generate B2B cold emails that strictly adhere to a JSON schema, ensuring seamless programmatic integration with the backend application.

### Training Architecture

- **Base Model:** Mistral-7B-Instruct-v0.2.
- **Optimization Technique:** QLoRA (Quantized Low-Rank Adaptation). This allowed for parameter-efficient fine-tuning on consumer-grade hardware while retaining model performance.
- **Quantization:** The base model was loaded in 4-bit precision (nf4 type) with double quantization enabled to minimize memory footprint.

### Dataset Preparation

- **Source:** The Enron Email Corpus (filtered\_cleaned\_enron.csv).
- **Sampling:** A subset of 5,000 records was utilized to balance training time with diversity.
- **Instruction Formatting:** The data was restructured into a strict instruction-response pair using the Mistral chat template <s>[INST] {instruction} [/INST] {response} </s>.
- **System Prompt:** A custom system prompt was injected into every training example to enforce constraints:
  - *Length:* 60–120 words.
  - *Format:* Strict JSON output {"subject": "...", "body": "..."}.
  - *Constraint:* No invention of facts; exclusive reliance on the provided lead\_profile.

## Hyperparameter Configuration

The model was trained using the SFTTrainer (Supervised Fine-tuning Trainer) from the trl library with the following configurations:

- **LoRA Config:**
  - Rank: 16
  - Alpha: 32
  - Target Modules: q\_proj, k\_proj, v\_proj, o\_proj, gate\_proj, up\_proj, down\_proj (targeting all linear layers for maximum adaptation).
- **Training Arguments:**
  - **Epochs:** 2
  - **Learning Rate:**  $2 \times 10^{-5}$
  - **Optimizer:** paged\_adamw\_8bit
  - **Batch Size:** 8 (per device) with 4 gradient accumulation steps.
  - **Precision:** bf16 (BFloat16) enabled for training stability.

## Training Results

The training process demonstrated significant convergence over 2 epochs:

- **Initial Loss:** ~1.18
- **Final Loss:** ~0.63
- **Outcome:** The model successfully learned to ignore its default conversational nature and act as a strict "JSON-speaking" copywriter, significantly reducing post-processing errors (parsing failures) compared to the base model.

## 5. Model Evaluation

### A. MBTI Classification Model

#### Performance Analysis and Findings

During the training process, the model demonstrated a gradual decrease in both training and validation losses, alongside a steady improvement in accuracy. Blackbox optimization was used to tune the LoRA hyperparameters, which successfully improved the model's convergence, stability, and generalization.

A key part of this evaluation was comparing the LoRA method against a traditional full fine-tuning approach:

- **Efficiency:** The LoRA method significantly reduced both training time and GPU usage. The full fine-tuning approach required **three times more training time** to achieve a performance level similar to the LoRA-based model.
- **Weight Updates:** While more time-consuming, the loss reduction per epoch was steeper in the full fine-tuning approach, indicating more substantial weight updates during training.
- **Conclusion:** Given the current limitations in GPU resources and time, LoRA is the preferred and more practical option.

#### Limitations and Future Work

Further training is required to improve the model's overall accuracy. This need stems from the inherent complexity of the MBTI dataset and the aforementioned resource constraints. Future work will focus on improving model accuracy once sufficient GPU resources become available.

## B. Email Generation Model

The generated emails must adhere to several strict design rules:

- **Length:** Keep the email concise (target 60-120 words).
- **Call-to-Action (CTA):** Include **exactly one** clear CTA.
- **Links:** Contain **no links** in the first email.
- **Grounding:** Use *only* information from the provided lead and style profiles.
- **Structure:** Use a friendly greeting with the recipient's name and a proper closing with the sender's name.

### Evaluation Setup

The model was tested on **7 unseen test leads** from diverse B2B contexts (e.g., fintech, SaaS, e-commerce, healthtech). Each lead profile was detailed, including data such as role, company history, recent activity, skills, and inferred MBTI.

Since there is no single "correct" email, performance was measured using **rule-based checks** rather than traditional accuracy. These checks included:

- Basic structure (subject/body present).
- Word count against the target band.
- Presence of correct greeting and closing names.
- Absence of URL-like text.
- A count of CTA verbs (e.g., "schedule," "chat") to verify the "exactly one" rule.
- Checks for personalization (lead's name/company) and objective grounding.
- A final **1-10 quality score** was assigned to quantify how well the email matched all rules.

### Key Findings

The model's performance was mixed, with clear successes and specific areas needing improvement. Overall quality scores for the 7 emails fell in the middle of the 1-10 scale.

#### Successes:

- **Reliable Structure:** The model consistently produced a subject and body for all 7 test cases.
- **No Links:** The model successfully adhered to the "no links" rule.
- **Objective Grounding:** The model performed well at incorporating the lead's specific objective, using relevant terms like "activation," "churn," or "onboarding".

### **Areas for Improvement:**

- **Length:** Emails were frequently longer than the 60-120 word target. They often read more like full examples or templates rather than short, concise cold emails.
- **Placeholders:** The model often used placeholders (e.g., [First Name]) instead of inserting the actual name from the profile. This caused the strict greeting/closing checks to fail, even when the email's tone was polite.
- **CTA Consistency:** While a CTA was generally present, the *number* of CTAs was inconsistent and did not reliably meet the "exactly one" rule.
- **Personalization:** The use of the lead's name and company was inconsistent, appearing in some emails but not all.

### **Overall Assessment**

The current prompt-based model generates emails that are readable, polite, and relevant to the lead's business objective. However, it does not yet fully and consistently adhere to the specific design rules, particularly regarding length, placeholder usage, and CTA count.

## **6. Final Models**

This part covers development roadmaps for the MBTI classifier and email generation models.

### **A. MBTI Personality classifier**

#### **1. Problem Definition**

The task is to predict a person's MBTI personality type based on their written sentences

This is a 16-class text classification problem, where each MBTI type is formed from four traits: I/E, N/S, T/F, J/P

#### **2. The initial dataset - 2,332,229 samples**

We first trained DistilBert on the dataset. However the dataset was noisy, highly imbalanced, and mostly short and low-information sequences were available.

Because of this, the fine-tuned model produced only 7 % training accuracy, confirming the dataset had insufficient signal for personality prediction.

#### **3. Mini Dataset - around 9,000 Samples (Clean Version)**

We later identified a cleaner mini version of the dataset containing 9,000 samples. After preprocessing to match our target evaluation domain (professional / formal sequences), we retained around 7,000 cleaned samples and balanced distribution across all 16 MBTI classes.

This dataset contained longer, meaningful sentences, giving the model better personality cues.

#### **4. Final Model - DistilBert and LoRA (PEFT)**

We fine-tuned DistilBERT using LoRA (Parameter-Efficient Fine-Tuning) on the curated 78k dataset. Because of the clean, balanced, and semantically richer data

##### **Training Performance**

Training accuracy reached ~90%, showing that the model could successfully learn patterns from the curated dataset.

##### **Test Performance**

Test accuracy was ~40% percent.

This is significantly higher than the original noisy dataset experiments (7–14%) and demonstrates effective generalization, given the difficulty of 16-class MBTI prediction.

#### **5. Deployment**

For model deployment, we selected HuggingFace Spaces due to its simplicity, free hosting. This allowed the DistilBERT + LoRA fine-tuned MBTI classifier to be deployed as an interactive web interface accessible to anyone.

The Model is deployed at:

**<https://huggingface.co/spaces/PavanB99Indian/mbti-classifier>**

Users can enter text, and the interface returns:

- The predicted MBTI type (16 classes)
- Trait-level confidence scores for I/E, N/S, T/F, J/P

The HuggingFace Space contains the following files:

```
├── app.py  
├── requirements.txt  
├── README.md  
└── model/  
    ├── config.json  
    ├── tokenizer.json  
    ├── adapter_model.safetensors  
    ├── adapter_config.json  
    └── model.safetensors
```

## B. Email Generation Model

### 1. The Hardware Wall

Objective: Build an intelligent agent capable of generating personalized B2B cold emails based on lead profiles.

The initial plan was ambitious: leverage the power of Mistral 7B Instruct to handle complex reasoning for email personalization. However, during the early development stages, we hit a significant infrastructure bottleneck.

- The Block: The development environment (Google Colab T4 GPU) struggled to load the full Mistral 7B model efficiently for testing.

- The Evidence: The code reveals commented-out sections where mistralai/Mistral-7B-Instruct-v0.2 was initially attempted but abandoned in favor of a lighter architecture.
- 

## 2. The Lightweight Prototype

Strategy: To validate the logic and prompt engineering without waiting for hardware upgrades, we pivoted to a "Small Language Model" approach.

- The Switch: We swapped the heavy 7B model for TinyLlama/TinyLlama-1.1B-Chat-v1.0. This allowed for rapid iteration on the logic.
- The Logic Development: During this phase, we perfected the Python logic that maps personality types to writing styles. We defined specific dictionaries for MBTI types (e.g., ENTJ maps to high assertiveness and decisive CTAs).
- Prompt Engineering: We established the core "System Prompt" rules that would later define the final model:
  - 60–120 words max.
  - No links in the first email.
  - Strict JSON output format.

Key Lesson: This phase proved that the *logic* (mapping user profiles to email styles) was sound, but the 1.1B model lacked the nuance required for high-stakes B2B communication.

---

## 3. Fine-Tuning

Context: With access to better technical specifications (local environment execution), we returned to the original vision: Mistral 7B.

Instead of just *using* Mistral, we decided to fine-tune it to strictly adhere to our JSON output requirements and the specific "Enron corpus" style.

## 1. Data Preparation

We utilized the Enron email dataset but applied a strict subset strategy to ensure high-quality training without overfitting or excessive training times.

- Source: `filtered_cleaned_enron.csv`.
- Subset: We sampled exactly 5,000 records for the fine-tuning process.
- Formatting: Data was formatted into a strict instruction-response pair: `<s>[INST] {instruction} [/INST] {response} </s>`.

## 2. QLoRA Implementation

To make fine-tuning feasible and efficient, we utilized QLoRA (Quantized Low-Rank Adaptation).

- Quantization: Loaded the model in 4-bit precision using `BitsAndBytesConfig` (`nf4` type) to minimize memory usage.
- LoRA Config: We applied adapters to every major linear layer to ensure deep learning of the style:
  - Rank: 16
  - Alpha: 32
  - Target Modules: `q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, `down_proj`.

3. Training Results: The training run spanned 2 epochs with a learning rate of `2e-5`. Starting Loss:  $\sim 1.18$ , Final Loss:  $\sim 0.63$ . The significant drop in loss indicated the model successfully learned to map the lead profiles to the specific JSON email format we required.

## 4. The Final Deployment

Following the successful training run, the adapter weights and tokenizer were saved locally to `mistral-7b-enron-email-finetune`.

The final model, capable of taking structured lead data and outputting a perfectly formatted JSON email object, has been deployed for public use.

Live Model: [huggingface.co/anishark/LEAD\\_EMAIL\\_GENERATION\\_MODEL](https://huggingface.co/anishark/LEAD_EMAIL_GENERATION_MODEL)

Feature	Prototype Phase	Final Production Phase
Base Model	TinyLlama 1.1B Chat	Mistral 7B Instruct v0.2
Technique	Zero-shot Prompting	QLoRA Fine-Tuning
Data Source	N/A (Prompt Only)	Enron Dataset (5k Subset)
Infrastructure	Colab T4 (Cloud)	Local Execution (High Spec)
Context Length	220 Tokens	768 Tokens (Truncated)