**Project Proposal: SkyComm-AIDE – An NLP Agent for Air Traffic Communication for Instrument Control and Flight Management**

**Description:**  
This project builds an AI agent that listens to pilot transmissions and generates appropriate air traffic control (ATC) responses. To achieve this, we first classify each utterance as either “pilot” or “controller.” We then focus on pilot inputs—detecting their intent and critical flight parameters—and automatically propose safe, standardized ATC replies.

The project will be implemented as a minimum viable Python pipeline within a Jupyter Notebook. We will preprocess communication transcripts, extract relevant flight management information, and classify utterances according to their function within instrument control procedures, aiming to support automated monitoring or decision assistance tools for air traffic management.

**Problem Statement**

Clear, timely responses from ATC are vital for safe flights—especially under instrument flight rules (IFR). Pilots transmit requests (e.g., altitude change, heading queries) and expect precise instructions. Human delays or misinterpretation can cause inefficiencies or safety risks.  
We aim to automate the loop:

1. **Detect** pilot speech in mixed transcripts.
2. **Extract** intent and flight parameters.
3. **Generate** compliant ATC responses.  
   Automating this process will support controller decision-making and reduce workload peaks.

**Background**

Air traffic control communications have unique linguistic characteristics, including use of standardized phraseology, abbreviations, and procedural commands. Previous research has explored speech recognition for ATC communications, intent detection, and anomaly detection in air traffic exchanges.

* Classified utterances into instructions and acknowledgments.
* Detected intent and extracted flight parameters from text
* Shown small-scale chatbots proof-of-concept.

We extend this by creating a closed-loop text pipeline: pilot-to-ATC generation.

**Methodology**

1. **Transcript Segmentation & Speaker Classification**
   * Input: mixed transcripts.
   * Model: fine-tuned transformer (e.g. BERT) to tag each line “pilot” vs. “controller.”
2. **Pilot Intent & Parameter Extraction**
   * Within “pilot” segments, classify intents:
     + Altitude change, heading change, clearance request, position report, etc.
   * Use sequence-labeling (BiLSTM-CRF or fine-tuned BERT-NER) to pull out parameters (flight levels, headings, waypoints).

3. **Context Tracking**

* Maintain a simple state dictionary: current altitude, heading, clearance status.
* Update state after each pilot and generated ATC turn.

1. **ATC Response Generation**

* Fine-tune a seq-to-seq model (e.g. T5 or GPT-2) on paired pilot-controller text.
* Input: pilot utterance + current state.
* Output: ATC-compliant sentence (“Climb and maintain FL200,” “Turn right heading 090.”).

1. **Evaluation**

* **Automatic**: BLEU/ROUGE, intent-parameter accuracy.
* **Domain**: Phraseology compliance checks.
* **Human**: Expert ratings on clarity and safety.

All code and analysis will be implemented and documented in Jupyter notebooks and on GitHub Repo, ensuring transparency and reproducibility.

**Datasets**

We plan to use the following datasets:

* **ATCOSIM Dataset:** Publicly available simulated ATC communications with transcripts and labels for various instruction types.
* **FAA Communications Corpus:** A collection of real-world ATC-pilot voice transcripts annotated for procedural commands and interactions.
* **OpenSky Network Dataset (optional):** Flight trajectory data that can complement the textual data for validation and correlation.

These datasets provide both realistic and controlled environments to train and test NLP models on domain-specific communications.

**Expected Outcomes**

We expect to deliver:

* End-to-end NLP pipeline:

1. Speaker separation
2. Intent & parameter extraction
3. ATC reply generation

* Comparative study of classical vs. transformer-based approaches.
* Jupyter notebooks documenting the workflow.
* Insights for real-time, text-based decision support in ATC.

**Team Roles**

| **Team Member** | **Responsibilities** |
| --- | --- |
| **Member 1** | Speaker classification & dataset prep |
| **Member 2** | Pilot intent categorization & slot extraction |
| **Member 3** | Context management & response-generation model training |
| **Member 4** | Evaluation framework, expert coordination, visualization, reporting |

All team members will collaborate on data analysis, project documentation, and presentation preparation.