**1. Environment and Tools**

* **Platform**: Google Colab (T4 GPU)
* **Programming Language**: Python
* **Libraries Used**:
  + pandas for data handling
  + re for regex-based text processing
  + transformers (Hugging Face) for NLP pipelines
* **Models Used**:
  + google/flan-t5-small for summarization
  + typeform/distilbert-base-uncased-mnli for zero-shot classification

**2. Objective**

To analyze customer-agent conversation transcripts and extract structured insights such as:

* Agent identity
* Complaint and dissatisfaction indicators
* Regulatory issue detection
* Core issue summarization
* Issue categorization using zero-shot learning

**3. Data Preprocessing**

* **Source**: conversations.csv
* **Text Cleaning**:
  + Removed common punctuations (" , ! ’ “ ”)
  + Lowercased and stripped whitespace
* **Channel Split**:
  + Filtered utterances by channel (agent or customer)
  + Aggregated all utterances by callid

**4. Agent Name Extraction**

* Aggregated all agent utterances per call ID
* Used regex patterns to extract names from phrases like:
  + "my name is [name]"
  + "this is [name]"
  + "you're speaking with [name]"
* Extracted names were capitalized and added as a new column agent\_name

**5. Customer Utterance Aggregation**

* Aggregated customer utterances by callid
* Retained only the first two meaningful sentences (split by ., ?, or /)
* Saved into all\_customer\_utterances

**6. Sentiment and Complaint Detection**

* **Dissatisfaction Detection**:
  + Regex-based search for keywords such as not happy, unhappy, angry, bad, etc.
  + Result saved in column EOD
* **Complaint Detection**:
  + Looked for presence of complaint, complain
  + Boolean flag stored in is\_complaint\_call
* **Regulatory Detection**:
  + Detected money-related keywords like refund, billing, charged, account statement
  + Boolean flag mentions\_money
  + Regulatory calls are those where both complaint and money terms are present
  + Final flag is\_regulatory

**7. Issue Summarization (Customer-Focused)**

* Used google/flan-t5-small summarization model
* Created a custom prompt asking the model to extract only the customer's main issue
* Prompt included instructions to:
  + Focus only on problem or request
  + Ignore personal details
* Summary output saved to customer\_issue\_summary

**8. Issue Categorization using Zero-Shot Classification**

* Used zero-shot classification pipeline with typeform/distilbert-base-uncased-mnli
* Predefined list of candidate labels (e.g., billing inquiry, refund request, technical support, etc.)
* Each utterance categorized into the most likely issue type
* Result saved in column issue\_category\_zeroshot

**9. Output DataFrame Summary**

Final structured DataFrame customer\_utterances\_df contains the following columns:

* callid
* all\_customer\_utterances
* EOD (dissatisfaction keywords found)
* is\_complaint\_call
* mentions\_money
* is\_regulatory
* customer\_issue\_summary
* issue\_category\_zeroshot

**10. Conclusion**

This project demonstrates a practical NLP pipeline that converts raw conversation data into structured, actionable insights. It supports:

* Detection of complaint and regulatory calls
* Customer issue summarization using a lightweight generative model
* Automated classification of issue types without any labeled training data
* Potential downstream use cases include dashboarding, alerting, and smart call routing