

Project Title	AI Customer Support Copilot
Skills Takeaway From This Project	Python, NLP, Deep Learning, HuggingFace Transformers, Transfer Learning, Vector DB, RAG, Synthetic Data Generation, Streamlit, API Integration, Data Preprocessing, Model Fine-Tuning, Intent Classification, Sentiment Analysis
Domain	Telecom Industry, Customer Service AI Automation

Problem Statement:

ZENDS Communications is a virtual telecom company providing mobile, broadband, cloud, and IoT services. Like many telecom companies, support agents face challenges in handling large volumes of customer queries across billing, refunds, technical issues, and product inquiries. Manual response is slow, inconsistent, and resource-intensive.

To address this, we provide an AI Customer Support Copilot that automatically understands customer messages. The system detects customer intent and sentiment, prioritizes queries based on urgency, retrieves relevant policy or product information using RAG, and generates accurate, human-like responses. This enables handling unstructured data, building deep learning models, and deploying an AI copilot system for efficient customer support.

Business Use Cases:

- Reduce customer support response time with AI-suggested replies.
- Help agents efficiently handle large volumes of queries.
- Ensure consistent, policy-compliant communication with customers.
- Improve customer satisfaction and reduce churn.
- Realistic enterprise AI deployment without requiring access to actual company data.
- Generate synthetic datasets based on company documentation.

Approach:

Step 1: Virtual Company & Dataset Design

- Created a virtual telecom company: ZENDS Communications.
- Defined product groups, plans, services, pricing, SLAs, policies.
- Designed realistic customer queries that users may ask about products, billing, refunds, technical issues, and complaints.
- Rationale: In industry, structured datasets are rarely available initially. Synthetic data allows fast model development while maintaining realism.

Step 2: Synthetic Dataset Generation

- 20,000 records covering five intents: Billing, Refund, Technical, Complaint, Product Inquiry.
- Columns: text, intent, sentiment.
- Approach: Intent Templates, Entity Injection, Sentiment Variation, Paraphrasing using HuggingFace Transformers.
- Benefits: Balanced dataset, clean and labeled, suitable for HuggingFace model training.
- Future extension: Students can generate datasets by updating product templates, policies, or regions.

Step 3: Intent & Sentiment Classification

- **Intent Classification Model:** Fine-tune HuggingFace Transformers like **DistilBERT, BERT, or RoBERTa**. Suitable for multi-class intent detection.
- **Sentiment Analysis Model:** Use pre-trained HuggingFace models like **distilbert-base-uncased-finetuned-sst-2-english, BERTweet Sentiment**, or **cardiffnlp/twitter-roberta-base-sentiment** for classifying positive, negative messages. Explore more open source model on Hugging face.
- Evaluate using accuracy, F1-score, and confusion matrix.

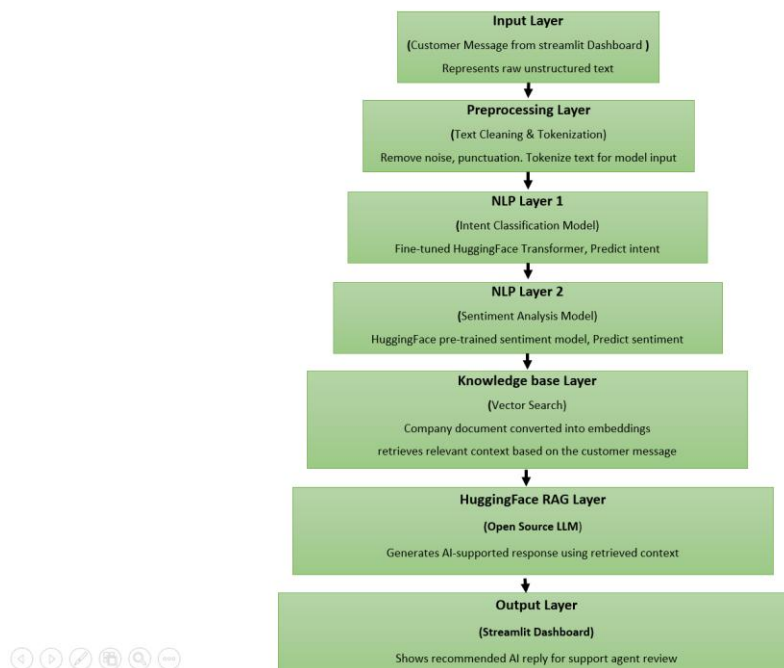
Step 4: Retrieval-Augmented Generation (RAG)

- Convert company policy and product documents into embeddings (Sentence-Transformers).
- Store embeddings in vector database. Options (Fasis, Pinecone, Chroma)
- Retrieve relevant context based on customer query.
- Generate response using open-source HuggingFace LLM (examples: **Mistral-7B, Falcon-7B, LLaMA-2-7B**) for free or lightweight inference. Can Also explore paid LLM API's based on cost of the project.

Step 5: Streamlit Application

- Accepts real-time customer queries.
- Shows predicted intent and sentiment.
- Displays retrieved policy or product information.
- Generates recommended AI response for agent review.
- Provides a fully functional interactive dashboard.

Data Flow and Architecture:



Dataset

Dataset Link: [ZenDS Communications queries.csv](#)

Dataset Description

Features Included:

- Text - Customer message.
- Intent – Purpose of query about (Billing, Refund, Technical, Complaint, Product).
- Sentiment – Sentiment detected from the Customer message (Angry, Neural, Happy)
- Dataset Size: 10,000 records (synthetic, balanced).

Exploratory Data Analysis (EDA)

- Univariate: Distribution of intents, sentiments, message length.
- Bivariate: Sentiment vs intent, common complaints, refund-related queries.
- Multivariate: Interaction of message length, intent, sentiment.
- Outlier detection: unusually long/short messages, rare intents.
- Correlation: sentiment frequency per intent, high-severity complaints.
- No need to show in the dashboard. Only for understanding about synthetic Dataset.

Results:

- Cleaned and labeled synthetic dataset.
- Fine Tuned intent classification and sentiment models.
- Developed RAG-based AI response generator.
- Built interactive Streamlit dashboard.
- Full end-to-end enterprise AI simulation.

Project Evaluation Metrics:

- Accuracy and F1-score for intent classification.
- Accuracy of sentiment analysis.
- Relevance and correctness of AI-generated responses.
- Smoothness and usability of Streamlit dashboard.
- Effective use of RAG for context-aware responses.

Technical Tags:

Python, NLP, Deep Learning, HuggingFace Transformers, Vector DB, RAG, Streamlit, Synthetic Data Generation, API Integration, Data Preprocessing.

Deliverables:

- Python scripts for synthetic dataset generation.
- Fine-tuned intent classification model.
- Sentiment analysis model Integration from Hugging face.
- Vector database of company documents.
- HuggingFace LLM response generator.
- Interactive Streamlit dashboard.
- Complete project documentation.

Timeline:

Check your email for submission deadlines related to this project.

Project Learning Outcomes:

- Simulate real enterprise data from company documents.
- NLP models Integration for intent classification and sentiment analysis.
- Build a RAG-based AI system using vector databases and Open Source LLMs.
- Integrate deep learning models into a functional web app.
- Learn industry best practices in synthetic dataset creation and enterprise AI system design based on domain and company.
- Solve real-world customer support problems without waiting for structured company data.

Problem Solved:

- Explains a virtual telecom company's operations and challenges.
- Automates customer support to address high query volumes and unstructured messages.
- Reduces response time and improves agent efficiency.
- Provides policy-compliant, context-aware replies.
- Demonstrates end-to-end enterprise AI workflow for educational purposes.
- Prepares students for industry-ready AI projects using NLP, deep learning, and RAG.

Project Reference Documents:

All requirement document and sample code file given in the [final project](#) folder.