Chat
Feature

EXPLORING BUSINESS ENHANCEMENT THROUGH GENAI

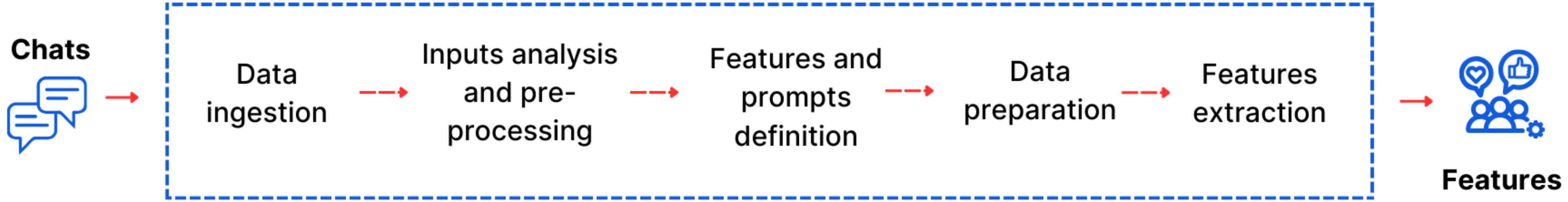
USE CASE OVERVIEW

This is an overview of an advanced analytics framework that harnesses **GPT-4** to transform **customer service chat** data into **structured insights**, driving business process improvements.

The project presented a unique challenge: how to effectively deploy cutting-edge AI models within a **business environment**. Working within a corporate context required careful consideration of several key factors. We needed to maintain strict cost controls, seamlessly integrate with existing systems, and ensure real-time analysis capabilities. This solution balances these competing demands, optimizing for cost efficiency, performance, speed, and reproducibility to deliver an analytics system that meets real-world business needs.

This project was conducted while working at a consultancy firm and developed for a client company related to the public gambling sector in Italy.

USE CASE WORKFLOW



Chats

Data ingestion

Inputs analysis and pre-processing

Features and prompts definition

Data preparation

Features extraction

Features

DATA

Chats from an online chat support service for customers accessible from the client website

ARCHITECTURE

Advanced analytics architecture based on **Microsoft Azure** cloud services

MODEL

OpenAI **GPT-4-Turbo** integrated through Azure OpenAI

DATASET

The data employed for this project are provided by the client company and includes a compilation of **chat conversations written in Italian**. These conversations are stored in a **CSV** file where each row contains a message related to a conversation, reconstructable in its entirety through a unique identifier for each chat. The data is characterized by various **key attributes**, including temporal information, user details and identifiers, crucial for reconstructing the interaction or providing customer information.

After the pre-processing stage, the data was analyzed using various statistical methods (shown on the left).

5.435.698 ROWS

14 FIELDS

265.592 CHAT

20.47 avg n of mess per chat

4.16 avg minutes per chat

1650 avg chat per day

141,25 average tokens per chat

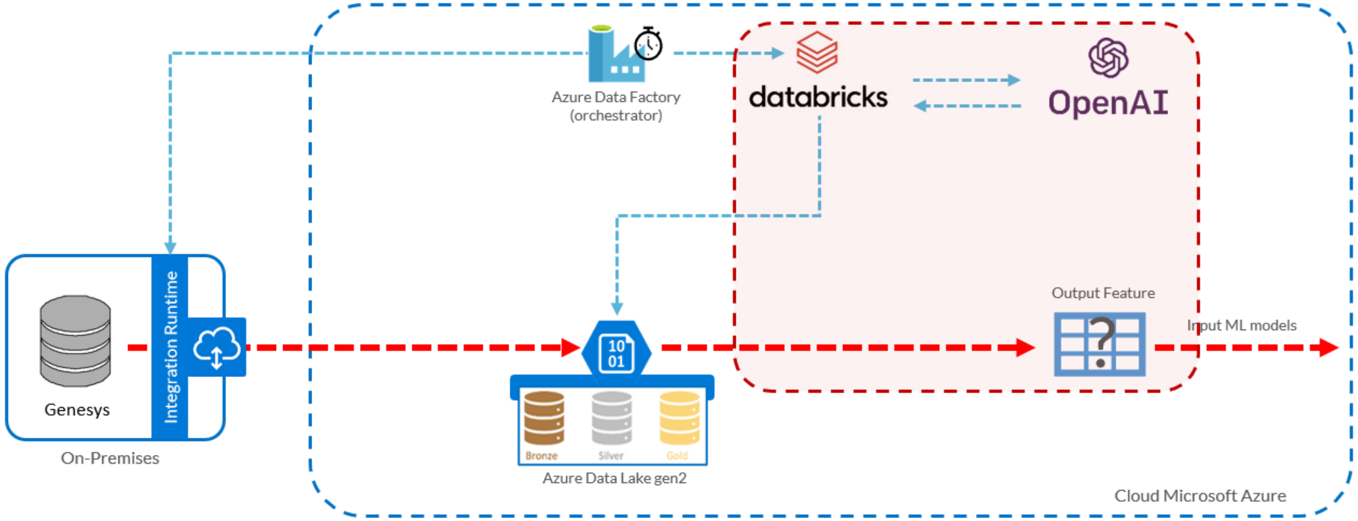
3 USERTYPES

Agent, Client, External

102.395 distinct customers

45% returning customers

PROJECT ARCHITECTURE



MICROSOFT AZURE CLOUD SERVICES

Azure Data Factory: data orchestration phase. *Azure Data Lake Gen2* serves as the storage system

Azure Databricks: analysis service for processing and managing data (direct connection with the data lake storage).

Azure Open AI: interaction with OpenAI models. Integrated into Databrick notebooks.

GOALS

- integrate this architecture into company Advanced analytics framework;
- use the extracted features as inputs for other ML and AI models

LLM SELECTION

The heart of this experimental framework lies in the utilization of a Large Language Model for feature extraction. Given the pivotal role of the LLM, it was crucial to define the best way possible to include it in the advanced analytics framework. The potential candidate models for the experiment included **OpenAI**'s GPT-3.5-Turbo-1106, GPT-4, and GPT-4-Turbo and all of them can be easily integrated into the project's architecture via Azure Open AI

SELECTION BASED ON

- **Cost** (*tokens*) – estimated through Tiktoken library
- **Performance**
- Additional features (**output format**, context tokens)

Considering these parameters, **GPT-4-Turbo** emerges as the selected model for the experiment.

FEATURES DEFINITION

A crucial aspect of the solution design involved identifying specific features that the client aimed to derive for each conversation. The primary objective of the use case was to generate **reusable features capable of serving as inputs for other analyses**. To achieve this goal, a comprehensive series of meetings was arranged with the client company. To ensure a robust and well defined framework that aligns both with business objectives and technical requirements, great importance was given to fostering **collaboration between technical experts and professionals possessing domain knowledge** specific to the legal gaming sector. After several meetings, **nine features** have been defined. The first seven features can be primarily associated with a classification task involving predefined classes, while the last two are freely generated by the model without any constraints.

FEATURES

Pre-defined classes generation

Request Resolution

Resolved, Unresolved, Not Resolvable

Customer Satisfaction

Very Dissatisfied, Dissatisfied, Neutral, Satisfied, Very Satisfied

Customer Emotions

Joy, Surprise, Anger, Sadness, Fear, Neutral

Customer Conversation Tone

Formal, Informal, Emotional, Sarcastic, Ironic

Customer Linguistic Level

Low, Medium, High

Operator Emotions *(same as customer)*

Operator Conversation Tone *(same as customer)*

Free generation

Chat Summary

Text of max 180 characters

Conversation Topic

Keyword extracted from the summary

For each feature, the value NA is designated for cases where they are not applicable or cannot be ascertained.

PROMPT ENGINEERING

Since feature extraction is implemented using a generative LLM, formulating the in struction as a **prompt** has proven to be the most crucial and delicate phase of the experimentation.

The design of the prompt posed several **challenges**, which can be summarized into three primary focuses:

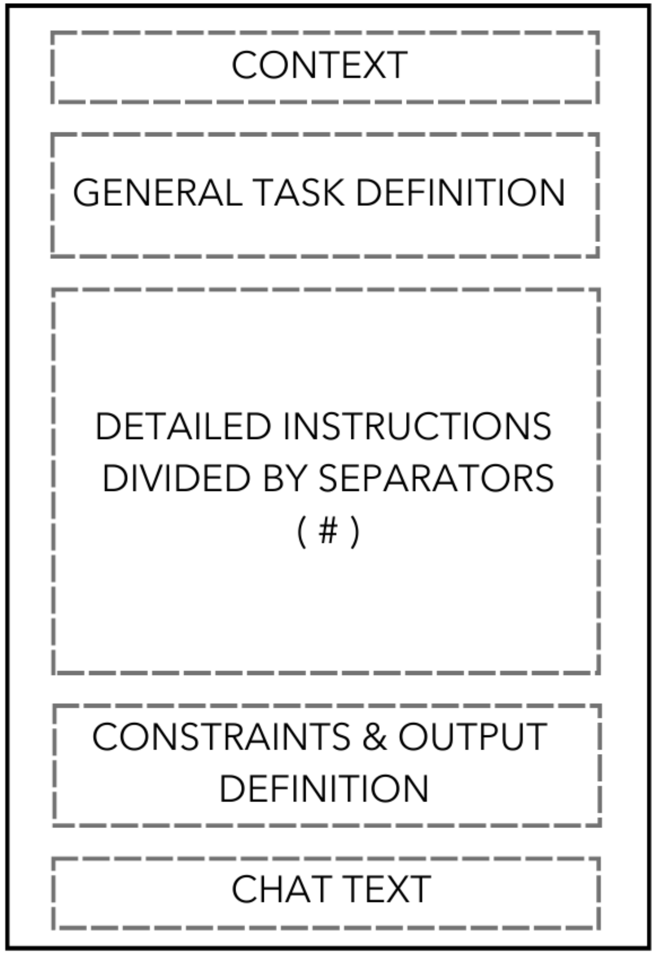
1. Determining the **optimal number of instructions** to write. Given the complexity of extracting nine features, a series of experiments were required to ascertain the most effective approach in crafting a single prompt for all features.
2. Refining the **language** and **style** used in instructions. Extensive testing has been conducted to identify the most clear and effective way to communicate with the model.
3. Selecting **techniques** for prompt crafting. A careful selection was made from state of-the-art methods and best practices to incorporate into the process

PROMPT

Final prompt is a structured text in **Italian** with instruction for **all the nine features.**

STRUCTURE

1. **contextual sentence** to define the task's domain.
2. **pre-definition** of the **nine features** with possible classes enclosed in parentheses.
3. **instruction for the features** separated using **delimiters (#)**
    - For the features generated by the model, Chat *Summary* and *Conversation Topic*, a **step-by-step** approach was adopted
4. **Constraints & output format**
5. **text** of the **chat**

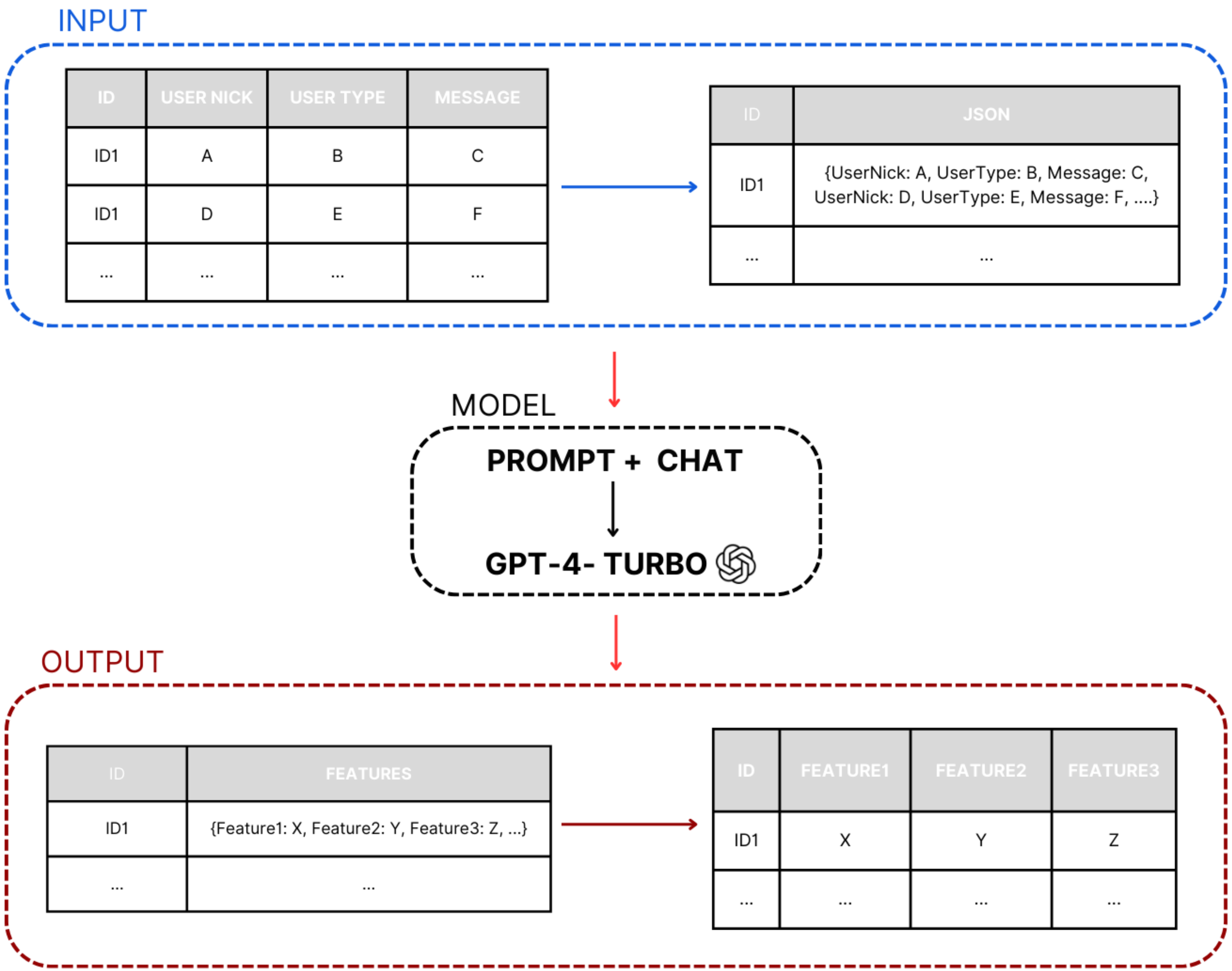

CONTEXT
GENERAL TASK DEFINITION
DETAILED INSTRUCTIONS DIVIDED BY SEPARATORS (#)
CONSTRAINTS & OUTPUT DEFINITION
CHAT TEXT

PERFORMANCE (over a 40 chats sample)

Quantitative analysis on 7 features: **90% avg accuracy**

Qualitative analysis on 2 features carried by **domain experts**

FEATURES EXTRACTION WORKFLOW



INPUT

ID	USER NICK	USER TYPE	MESSAGE
ID1	A	B	C
ID1	D	E	F
...	...	...	...

ID	JSON
ID1	{UserNick: A, UserType: B, Message: C, UserNick: D, UserType: E, Message: F, ...}
...	...

MODEL

PROMPT + CHAT

GPT-4- TURBO

OUTPUT

ID	FEATURES
ID1	{Feature1: X, Feature2: Y, Feature3: Z, ...}
...	...

ID	FEATURE1	FEATURE2	FEATURE3
ID1	X	Y	Z
...	...	...	...

OUTPUT

```
{"Risoluzione_richiesta": "Soddisfatto”,
    "Soddisfazione_cliente": "Soddisfatto",
    "Emozioni_cliente": ["Gioia", "Sorpresa"],
    "Emozioni_operatore": ["Neutralità"],
    "Tono_cliente": "Emotivo",
    "Tono_operatore": "Neutrale",
    "Livello_linguistico_cliente": "Medio",
    "Riassunto_conversazione": "Cliente segnala
un  problema  relativo al suo conto gioco.
L’operatore risolve il problema indicandogli la
soluzione.",
    "Argomento_conversazione": ["Conto gioco"] }
```

GPT output is returned in **JSON** format, consisting of a dictionary with nine pairs of key values. Data is post-processed using the decoding Json.loads from JSON API, encapsulated within a function to handle potential values not in JSON format since the function managing GPT output is designed to track possible errors returned by the API chat completions, which may not be encoded in JSON. The final output is a dataframe containing the **interaction ID**, the **success flag** of the decoding operation, **and nine columns** containing the extracted features.

RESULTS & INSIGHTS

The feature extraction workflow has proven **generally effective**, supported by the ability to define a **semi-structured output format** like JSON as a parameter of the model. Additionally, including a *Success* flag column in the final dataset makes it easy to manage exceptions during post-processing, filtering only the correct results. In any case, instances of failure were never found to be related to difficulties in processing the generated output unless it indicated an error generated by the Azure Open AI content filter, confirming the **validity of the data manipulation process**.

RESULTS IN A NUTSHELL

- General adherence to **established classes** with a broad scope.
    - Exception: *Customer emotions*
- Effective **text summarization** and **keyword extraction** capabilities
- Coherent utilization of ***NA***

The case study was designed as an **experiment** with purpose is to **integrate** this designed **analysis architecture** into **company** internal advanced analytics system

FUTURE APPLICATIONS & IMPROVEMENTS

POTENTIAL IMPROVEMENTS

- **Prompting** experiments (*few-shot prompting*)
- **Technical** improvements (*fine-tuning*)

APPLICATIONS

- **Analytics** related to customer care (*dashboards, reports*)
    - tailored business operations
- Data to train other **Machine Learning** models

SIDE APPLICATIONS

- Legal public gaming & ethical usage of AI