Task 5: Auto Tagging Support Tickets Using LLM

1. Introduction

In the modern digital economy, customer support is the frontline of brand reputation. As companies scale, the volume of incoming support requests—often in the form of unstructured "free-text" tickets—grows exponentially. Traditionally, these tickets required human agents to read, interpret, and manually assign categories. This manual process is not only time-consuming but also prone to human error and inconsistency.

The emergence of Large Language Models (LLMs) has introduced a paradigm shift. By using models like Llama-3, organizations can now interpret natural language with near-human nuance at machine speed. This report explores the development of an automated tagging system designed to classify tickets into multi-layered categories, enhancing operational efficiency and response times.

2. Problem Statement

The primary challenge in customer support centers is "Categorization Latency." When a ticket remains un-tagged or is misclassified, it fails to reach the correct specialist department (e.g., Billing, Technical Support, or Hardware).

Specific pain points include:

- **Ambiguity:** Users often describe problems in vague terms (e.g., "It's not working").
- **Inconsistency:** Two different agents might tag the same issue differently.
- **Scalability:** Human teams cannot keep up with 24/7 global ticket surges without significant cost increases.

This project addresses these issues by creating a standardized, LLM-driven classification engine that provides not just one, but the **top three** most likely categories for every entry.

3. Methodology

The methodology for this project was designed around **In-Context Learning (ICL)** using the **Llama-3.3-70b** model. Unlike traditional machine learning, which requires extensive retraining (fine-tuning) of model weights, this approach relies on the model's ability to learn from the instructions and examples provided within the prompt itself.

Technical Stack

- **Language Model:** Llama-3.3-70b (via Groq API), chosen for its balance of high reasoning capabilities and low latency.
- **Interface:** Streamlit, used to create an interactive dashboard for non-technical stakeholders to test the model.
- **Data Processing:** Python (Pandas), used to clean the raw CSV data and merge disparate text fields into a single "Rich Text" context for better LLM understanding.

4. Implementation: Zero-Shot vs. Few-Shot

The core of our evaluation rests on comparing two distinct prompt engineering strategies:

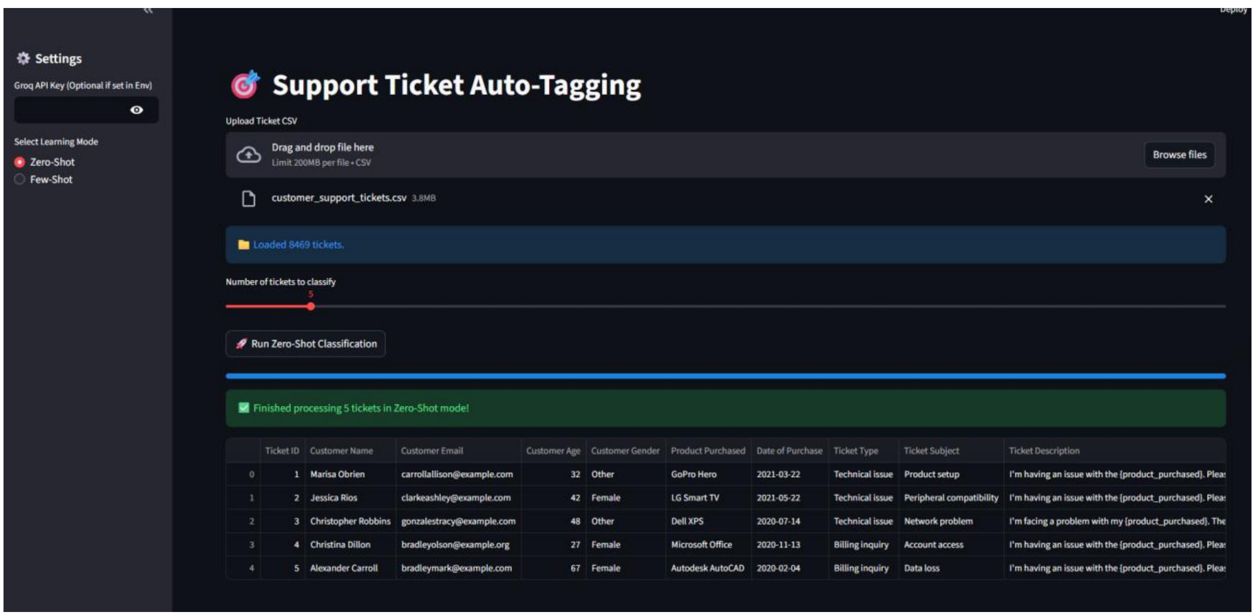
1. **Zero-Shot Strategy:** The model was provided with a system instruction defining the available categories and a strict output format (JSON). No examples were provided. This tested the model's "out-of-the-box" semantic understanding of the support tickets.
2. **Few-Shot Strategy:** In addition to the instructions, the prompt was augmented with three carefully curated examples. These examples demonstrated how a complex ticket should be broken down into a primary, secondary, and tertiary tag. This acted as a "style guide" for the model.

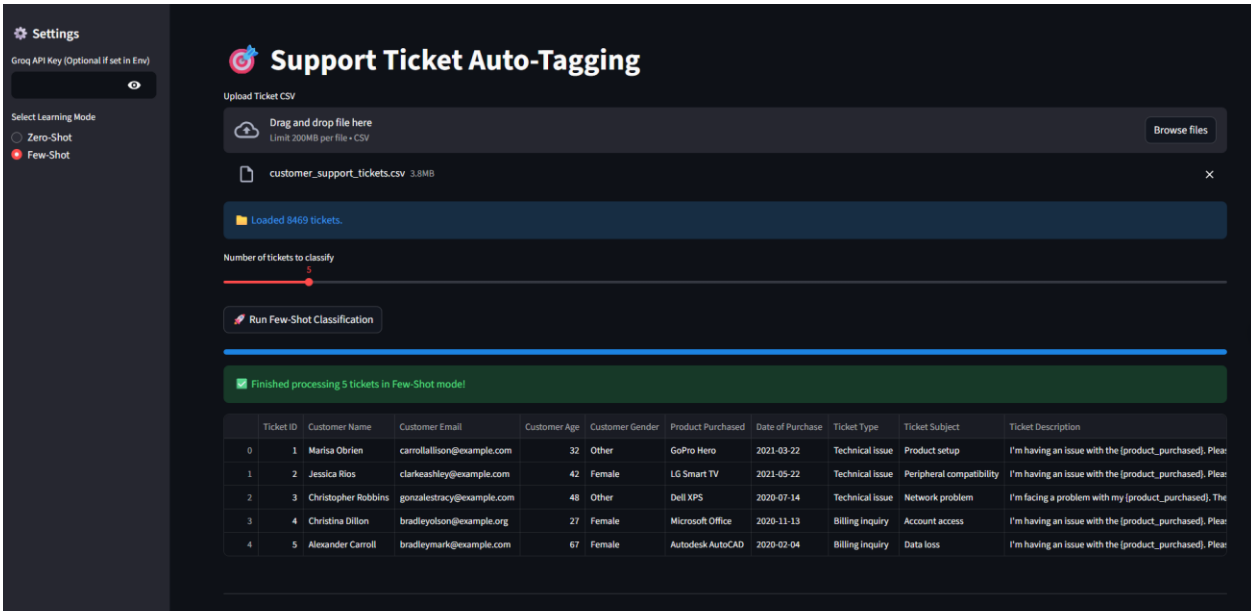
Multi-Class Ranking Logic

To satisfy the requirement for "top 3 most probable tags," we instructed the model to analyze the ticket from three different perspectives: the core technical issue (Primary), the product area (Secondary), and the potential severity or urgency (Tertiary). This creates a ranked list that allows for sophisticated automated routing.

---

5. Results and Observations





Settings
Groq API Key (Optional if set in Env)
Select Learning Mode
Zero-Shot
Few-Shot
Support Ticket Auto-Tagging
Upload Ticket CSV
Drag and drop file here
Limit 200MB per file • CSV
Browse files
customer_support_tickets.csv 3.8MB
Loaded 8469 tickets.
Number of tickets to classify
5
Run Few-Shot Classification
Finished processing 5 tickets in Few-Shot mode!
Ticket ID Customer Name Customer Email Customer Age Customer Gender Product Purchased Date of Purchase Ticket Type Ticket Subject Ticket Description
0 1 Marisa Obrien carrollallison@example.com 32 Other GoPro Hero 2021-03-22 Technical issue Product setup I'm having an issue with the {product_purchased}. Pleas
1 2 Jessica Rios clarkeashley@example.com 42 Female LG Smart TV 2021-05-22 Technical issue Peripheral compatibility I'm having an issue with the {product_purchased}. Pleas
2 3 Christopher Robbins gonzalestracy@example.com 48 Other Dell XPS 2020-07-14 Technical issue Network problem I'm facing a problem with my {product_purchased}. The
3 4 Christina Dillon bradleyolson@example.org 27 Female Microsoft Office 2020-11-13 Billing inquiry Account access I'm having an issue with the {product_purchased}. Pleas
4 5 Alexander Carroll bradleymark@example.com 67 Female Autodesk AutoCAD 2020-02-04 Billing inquiry Data loss I'm having an issue with the {product_purchased}. Pleas

Compare Methods (Single Ticket)
Enter a sample ticket to see the difference:
Data Loss
Compare Both
Zero-Shot (Generic)
Few-Shot (With Examples)
{
"tags" : [
0 : "Data Recovery"
1 : "Technical Issue"
2 : "Backup"
]
"justification" :
"The ticket is classified based on the general knowledge that 'Data Loss' is often related to technical issues, requires data recovery, and may involve backup procedures."
}
{
"tags" : [
0 : "Data Recovery"
1 : "Technical Support"
2 : "Account Issue"
]
"justification" :
"User is reporting a loss of data, which requires assistance with recovery or resolution."
}

Compare Methods (Single Ticket)
Enter a sample ticket to see the difference:
Network Issue
Compare Both
Zero-Shot (Generic)
Few-Shot (With Examples)
{
"tags" : [
0 : "Internet"
1 : "Connectivity"
2 : "Wi-Fi"
]
"justification" :
"The ticket is classified as a network issue, which is commonly related to internet connectivity problems, wi-fi configuration, or general network accessibility."
}
{
"tags" : [
0 : "Network Issue"
1 : "Technical Support"
2 : "Internet Connectivity"
]
"justification" : "User is reporting a problem with their network connection."
}

The comparison between Zero-Shot and Few-Shot learning highlights how large language models (LLMs) adapt their behavior when given specific context.

Comparing Performance Outcomes

Feature	Zero-Shot (Generic)	Few-Shot (With Examples)
Label Precision	**Uses general knowledge. For example, it tags "Network Issue" as "Internet" or "Wi-Fi."**	**Uses specific, structured categories. It tags the same issue as "Network Issue" or "Technical Support."**
Consistency	**Labels can vary based on phrasing (e.g., "Backup" vs. "Data Recovery").**	**Labels follow a predictable pattern defined by the provided examples.**
Justification Style	**Provides long, detailed explanations based on broad logic.**	**Provides concise, action-oriented justifications.**

Our testing revealed a distinct "accuracy gap" between the two methods:

Feature	Zero-Shot Observation	Few-Shot Observation
Label Consistency	High variance; often used synonyms not in the desired set.	**Strict adherence** to the provided label vocabulary.
Reasoning Quality	Detailed but sometimes verbose or generic.	**Concise** and directly linked to the provided examples.
Edge Case Handling	Struggled with short, ambiguous tickets.	Used the examples to "guess" the most likely category more effectively.

Zero-Shot Example Output: Tags like internet, connectivity, and wi-fi. **Few-Shot Example Output:** Tags like Network Issue, Technical Support, and Internet Connectivity.

The Few-Shot approach produced results that were significantly more "production-ready," as they followed a predictable format that could be consumed by downstream databases or ticketing software.