Customer Support Ticket Analyzer – Python

Project Overview

The objective of this project is to analyze customer support tickets using Python. Customer support teams receive multiple tickets daily, and analyzing these tickets helps understand common issues, service quality, customer sentiment, and priority distribution. The project uses Python data structures, string operations, loops, and functions to clean and analyze ticket data.

Data Description

The dataset consists of customer support tickets stored in a dictionary of lists.
Each ticket contains:

- Ticket Number
- Customer Name
- Issue Description
- Priority (High / Medium / Low)

Initially, 10 tickets were preloaded, and additional tickets were added using user input with auto-incremented ticket numbers and validated priority values.

Data Cleaning Process

Issue descriptions were cleaned to improve consistency and accuracy in analysis.
The following cleaning steps were applied:

- Converted all text to lowercase
- Removed punctuation marks
- Removed extra spaces
- Trimmed leading and trailing spaces
- Replaced shorthand terms such as “ok” with “okay”

This ensured uniform text for keyword analysis.

Keyword-Based Analysis

A function was created to count how many ticket descriptions contain specific keywords.
The analysis showed:

- Multiple tickets contained negative keywords like **“poor”** and **“slow”**
- Positive keywords such as **“good”** and **“excellent”** were also present

This indicates a mix of positive and negative customer experiences.

Priority Analysis

Tickets were categorized based on priority:

- High priority tickets indicate urgent customer issues
- Medium priority tickets indicate normal service requests
- Low priority tickets indicate non-urgent issues

The count of tickets in each category helped identify workload distribution and urgency levels.

Longest Issue Description

The ticket with the longest issue description was identified based on word count.
This helped highlight detailed problem explanations, which usually indicate complex or serious issues.

Unique Word Extraction

All unique words from the cleaned issue descriptions were extracted using a set.
This helped understand the vocabulary commonly used by customers and supported text-based insights.

Conclusion

This project successfully demonstrates how Python can be used to store, clean, and analyze customer support ticket data. By applying core Python concepts such as dictionaries, lists, string operations, loops, and functions, meaningful insights were extracted regarding customer issues, sentiment, and priority levels. This analysis can help support teams improve service quality and response efficiency.