Customer Support Ticket Analyser – Summary & Findings

Project Overview:

The Customer Support Ticket Analyser project uses Python to analyse customer support tickets and extract insights related to service quality, issue patterns, and priority levels. The system stores ticket data, accepts new tickets with validation, cleans unstructured text, and performs keyword-based and statistical analysis to support data-driven decision-making.

Data Cleaning & Preparation:

The dataset initially contained inconsistencies such as punctuation, extra spaces, mixed casing, and informal language. These were addressed by:

- Removing punctuation and extra spaces
- Converting all text to lowercase
- Trimming leading/trailing spaces
- Replacing slang (e.g., "ok" → "okay")

This cleaning ensured accurate keyword searches and improved the reliability of text-based insights.

Key Findings from Analysis:

Keyword Analysis:

- Words such as **"poor"** and **"slow"** appeared frequently, indicating dissatisfaction related to service quality and response time.
- Positive terms like **"good"** and **"excellent"** were also present, showing that effective support experiences exist alongside negative ones.

Priority Distribution:

- A higher number of **High-priority tickets** reflected urgent technical issues requiring immediate attention.
- **Medium and Low priority tickets** mainly involved general assistance and non-critical complaints.

Issue Complexity:

- Tickets with longer descriptions typically contained multiple concerns or detailed explanations, suggesting the need for escalated or personalized handling.

Vocabulary Insights:

- Extracting unique words revealed commonly used issue-related terms and reduced redundancy caused by inconsistent formatting.

Conclusion:

The analysis highlights that **service quality, response speed, and technical reliability** are the most critical areas impacting customer satisfaction. Text cleaning significantly improved analytical accuracy, and priority-based insights can help support teams allocate resources more effectively. This project demonstrates how Python can be used to transform raw support data into meaningful operational insights.