**GROUP C — WEEK 1 REPORT**

**Dataset Exploration & Preprocessing**

**Project:** Customer Support Ticket Classifier

**1. Introduction**

Week 1 focused on exploring, cleaning, and preparing the **Customer Support Ticket Dataset** for model training.  
Our main objective was to ensure the dataset is clean, balanced, and ready for fine-tuning **DistilBERT** to classify tickets into *Billing, Technical, Account,* or *Other*.

We began by inspecting the raw data, identifying missing or inconsistent entries, and applying structured preprocessing to make the dataset model-ready.

**2. Data Inspection Findings**

From the initial inspection, we observed:

* The dataset contains **520 rows and 4 columns.**
* Missing values were found in **‘text’** and **‘label’** columns (25 each).
* Several **typos and inconsistent casing** appeared in the ‘label’ column (e.g., *“Accnt”*, *“Tech-support”*, *“Othr”*).
* **No duplicate rows** were detected.

These findings confirmed the need for data cleaning to handle missing entries and normalize label names before training.

**3. Data Cleaning Process**

Our cleaning process followed a consistent workflow:

**Step 1 – Handling Missing Values**

* Removed rows with missing entries in either the ‘text’ or ‘label’ columns.

**Step 2 – Label Normalization**

* Converted all labels to lowercase.
* Trimmed spaces and mapped incorrect variations to four core categories:  
  *billing, technical, account, other.*

**Step 3 – Text Cleaning**  
We standardized text by:

* Removing **HTML tags** and decoding **HTML entities**.
* Removing **mentions**, **hashtags**, **emojis**, and **URLs** (via re).
* Stripping **extra spaces** and normalizing spacing.

After cleaning, all text entries were consistent, readable, and aligned with standardized label names.

**4. Exploratory Findings**

**4.1 Label Distribution**

After normalization, class counts were:

| **Label** | **Count** |
| --- | --- |
| Other | 125 |
| Technical | 119 |
| Account | 115 |
| Billing | 113 |

‘Other’ and ‘Technical’ tickets were slightly more frequent, indicating moderate class imbalance that will be addressed during model training.

**4.2 Text Length Analysis**

We analyzed the word count of ticket texts.

| **Metric** | **Value** |
| --- | --- |
| Count | 472 |
| Mean | 7.38 words |
| Std. Dev. | 1.08 |
| Min | 5 |
| 25% | 7 |
| 50% | 7 |
| 75% | 8 |
| Max | 10 |

Most ticket descriptions are concise, averaging about 7 words — typical for quick customer messages.

**5. Visual Summary**

To better understand the dataset, we visualized key features:

**5.1 Label Distribution Plot**

A **bar chart** highlights the class counts.

* The x-axis shows the four ticket categories.
* The y-axis represents the number of records.
* ‘Other’ and ‘Technical’ appear slightly higher, confirming minor imbalance.
* import matplotlib.pyplot as plt
* import seaborn as sns
* print(df['label'].value\_counts())
* # Plotting bar chart to know the distirbution of each class in the label class
* df['label'].value\_counts().plot(kind = 'bar', )
* plt.title('Label Distribution')
* plt.xlabel('Label')
* plt.ylabel('Count')
* plt.show()

label\_counts = {'billing':113, 'technical':119, 'account':115, 'other':125}

**5.2 Text Length Histogram**

A histogram of text length (in words) shows that most tickets contain an average of 7 words , 5 minimum words and a max of 19 words. This is good considering that our model would have a threshold for best performance.

The visuals confirmed that the dataset is well-structured, compact, and balanced enough for stratified splitting.

**6. Key Insights**

* The dataset is now **clean, consistent, and ready** for modeling.
* **Moderate class imbalance** exists but can be handled with stratified sampling or weighted loss.
* Ticket texts are **short**, enabling efficient contextual modeling with DistilBERT.

**8. Conclusion**

The Week 1 exploration and preprocessing phase built a solid foundation for the project.  
By cleaning, standardizing, and analyzing the dataset, we’ve ensured that the next stages — Baseline model development, model fine-tuning etc — will rest on reliable, high-quality data.