1 Dataset

To train a model, the training dataset plays an important role. In this document, we have made every possible attempt to understand, explore, and identify any missing data from the given dataset ('track-a.csv'). Let's go through the following Python code to first describe the dataset and its structure:

```
#import and data loading remains same throughout the document

import pandas as pd

df = pd.read_csv('track-a.csv')
print(f"Number of samples: {df.shape[0]}\n")

df.info()

#--Output:
#Number of samples: 2768


#RangeIndex: 2768 entries, 0 to 2767
#Data columns (total 7 columns):
#   Column    Non-Null Count  Dtype
#---  ------    --------------  -----
#0   id        2768 non-null   object
#1   text      2768 non-null   object
#2   anger     2768 non-null   int64
#3   fear      2768 non-null   int64
#4   joy       2768 non-null   int64
#5   sadness   2768 non-null   int64
#6   surprise  2768 non-null   int64
#dtypes: int64(5), object(2)
```

Now, let's interpret the output to understand the structure of the given dataset. The dataset consists of 2,768 text snippets, each annotated for five emotions: `anger, fear, joy, sadness, and surprise`. It has 7 columns, described as follows:

'id': A unique identifier for each text row.

'text': The actual text, annotated for emotions.

The remaining five columns ['anger', 'fear', 'joy', 'sadness', 'surprise'] act as binary flags, where a value of '1' indicates the presence of the respective emotion, and '0' its absence. The label `'disgust'` is missing, despite being mentioned in the task. Additionally, `df.info()` confirms that there are no null values across all 2,768 text snippets—meaning all seven columns contain non-null values.

Further analysis are performed without prepossessing of the given dataset.

1.1 Longest and Shortest text snippet:

Now lets consider length distribution within the dataset and the following Python code segment in next phase of analysis of text snippets :

```
longest_text = df.loc[df['word_count'].idxmax()]
print("Longest Text Id:", longest_text['id'], "\nWord Count:", longest_text['word_count'])

shortest_text = df.loc[df['word_count'].idxmin()]
print("\nShortest Text Id:", shortest_text['id'], "\nWord Count:",
shortest_text['word_count'], "\nShortest Text:", shortest_text['text'])
```

```
#--Output:
#Longest Text Id: eng_train_track_a_00612
#Word Count: 89

#Shortest Text Id: eng_train_track_a_01304
#Word Count: 1
#Shortest Text: &lt;/crazy-nutter&gt;
```

This analysis uncovers structural properties of the dataset, with the longest text (89 words), while the shortest (1-word HTML fragment) suggests possible data quality issues requiring prepossessing.

1.2 Text snippets with and without emotions:

In this section we are analyzing the vastness of the given dataset across emotion labels.

1.2.1 Emotionally Rich

The implemented code analyzes a multi-label frequency of text snippet present in the given dataset.

```
df['total_flags'] = df[emotions].sum(axis=1)
max_flags = df['total_flags'].max()
most_flagged_texts = df[df['total_flags'] == max_flags]

print("Count of Text with Rich emotions:", len(most_flagged_texts))

for idx, row in most_flagged_texts.iterrows():
print("Text Id: ",row['id'], "\nEmotions: ",row[emotions].to_dict())


#--Output:
#Count of Text with Rich emotions: 2
#Text Id:  eng_train_track_a_01753
#Emotions:  {'anger': 1, 'fear': 1, 'joy': 1, 'sadness': 1, 'surprise': 1}
#Text Id:  eng_train_track_a_02487
#Emotions:  {'anger': 1, 'fear': 1, 'joy': 1, 'sadness': 1, 'surprise': 1}
```

The above analysis uncovers that among 2,768 text snippets, only 2 such text are Emotionally Rich.

1.2.2 Emotionally Non-Neutral

The implemented code analyzes a minimally labeled instances in the given dataset.

```
labeled_texts = df[df['total_flags'] > 0]
# Find the minimum number of labels among labeled texts
min_flags = labeled_texts['total_flags'].min()
# Get ALL texts with this min count
least_flagged_texts = labeled_texts[labeled_texts['total_flags'] == min_flags]

print("\nCount of Text snippet with
 atleast one Labels (Non-Neutral):", len(least_flagged_texts))

#--Output:
```

```
# Count of Text snippet with atleast one Labels (Non-Neutral): 1141
```

The result of the analysis reveals that, With 1,141/2768 samples of labeled data showing minimal emotion assignments.

1.2.3 Emotionally Neutral

The neutral text detection implements null hypothesis testing for emotional content

```
neutral_texts = df[df['total_flags'] == 0]

print(f"\nCount of Text snippets which are Emotionally Neutral: ", len(neutral_texts))


#--Output:
# Count of Text snippets which are Emotionally Neutral:  239
```

The result of the analysis reveals that, With 239/2768 samples of labeled data showing neutral emotion assignments.

1.3 Plots:

1.3.1 Emotion Label distribution among dataset

We are visualizing the dataset among emotion labels to understand the possible imbalance across the dataset. Following bar graph, plots the data- in x-axis : Emotions and in y-axis: Number of Text Samples.

```
import matplotlib.pyplot as plt
import seaborn as sns

# 'disgust' label is missing in dataset
emotions = ['anger', 'fear', 'joy', 'sadness', 'surprise']

emotion_counts = df[emotions].sum().sort_values()
plt.figure(figsize=(11, 6))
ax = sns.barplot(x=emotion_counts.index, y=emotion_counts.values, palette="viridis")
for p in ax.patches:
ax.annotate(f"{int(p.get_height())}",
(p.get_x() + p.get_width() / 2., p.get_height()),
ha='center', va='top', xytext=(0, 10), textcoords='offset points')
plt.title("Emotion Label Distribution (Excluding 'disgust')")
plt.xlabel("Emotions")
plt.ylabel("Number of Text Samples")

plt.show()
```

As shown in Figure 1 (Section 1.3.1), Distribution of emotion labels across the dataset (excluding `disgust`). The barplot reveals significant class imbalance, with `joy` (674 samples) and `fear` (1611) dominating the distribution, while `anger` (333) and other emotions appear less frequently.

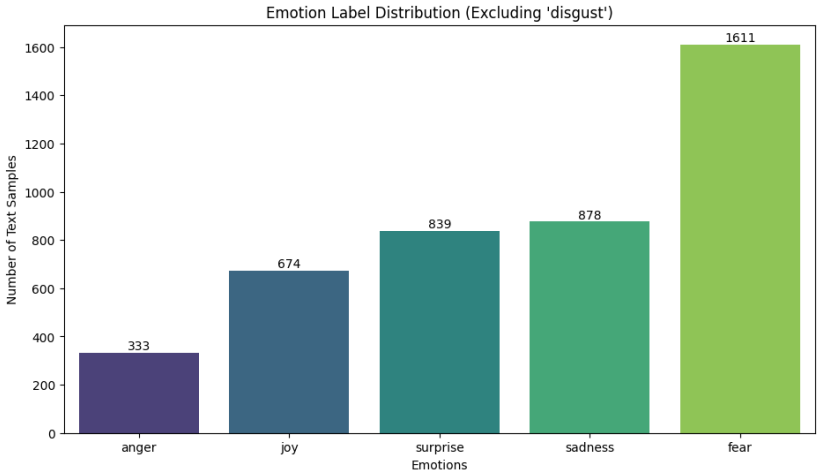

Figure 1: Emotion Label distribution among dataset

1.3.2 Frequency of Texts by Number of Emotions

This plot provides insights into the multi-label distribution of the dataset. A bar graph plot where in x-axis it describe 'Number of Emotions per Text" and y-axis "Count of text snippets".

```
df['num_emotions'] = df[emotions].sum(axis=1)
label_counts = df['num_emotions'].value_counts().sort_index()


plt.figure(figsize=(10, 5))
ax = sns.barplot(x=label_counts.index, y=label_counts.values,  color="teal")

# Add value labels
for p in ax.patches:
 ax.annotate(f"{int(p.get_height())}",
 (p.get_x() + p.get_width() / 2., p.get_height()),
 ha='center', va='center', xytext=(0, 10), textcoords='offset points')

plt.title("Frequency of Texts by Number of Emotions")
plt.xlabel("Number of Emotions per Text")
plt.ylabel("Count of Text snippets")
sns.despine()
plt.show()
```

As shown in Figure 2 (Section 1.3.2), Analysis of the "Frequency of Texts by Number of Emotions" graph uncovers the follows facts of the dataset as follows: There are Dominant Single-Label Texts 1141/2768 texts contain only one emotion, suggesting many instances exhibit unambiguous emotional expressions. Secondly, there are Multi-Label Rarity with 298/2768 texts have 3 emotions, while higher cardinalates are less:57/2768 texts with 4 emotions and 2/2768 texts for five emotions, towards extreme-right bar.

Finally, one may observe that, there are Neutral Texts- 239/2768 texts have zero emotions towards extreme-left bar.

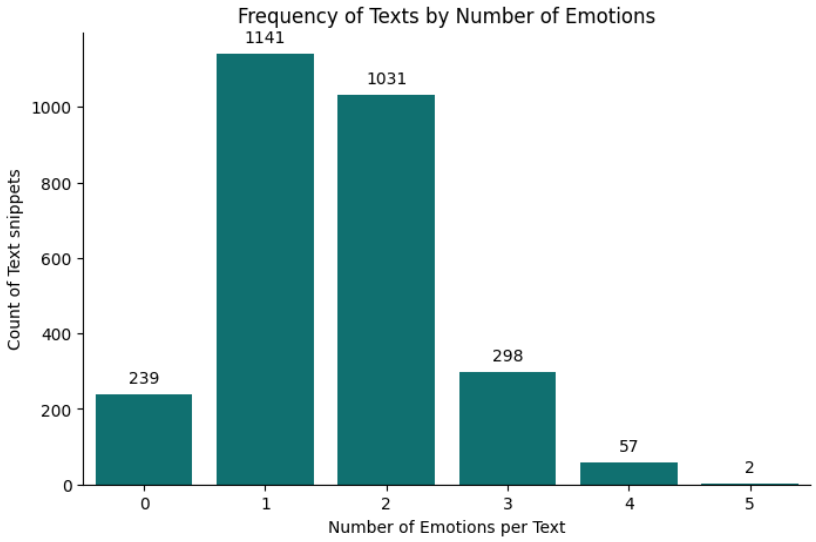

Figure 2: Frequency of Texts by Number of Emotions

1.3.3 Density of text lengths

To Understand the characteristics of dataset, Plotting the density of text lengths (in words and characters) is a foundational step in Natural language processing exploratory data analysis. A benefit of this plot is it, helps decide whether to use word-level, character-level, or sub-word tokenization.

```
import seaborn as sns
plt.figure(figsize=(8, 5))
sns.kdeplot(df['word_count'], fill=True, color='blue', label='Word Count')
sns.kdeplot(df['char_count'], fill=True, color='red', label='Char Count')
plt.title('Density of Text Lengths')
plt.xlabel('Length')
plt.legend()
plt.show()
```

As shown in Figure 3 (Section 1.3.3), Analysis of the Density Plot uncovers the dataset characteristics as follows, One may observe that both distributions are right-skewed. It is also evident that, the dataset contains a small number of longer texts. However, most text samples are short in length.

For both Word count indicated in 'blue' and Character count in 'red', As most texts are short in terms of words, It is peaked sharply at low values.

Additionally, one can observe that, Character count is greater than the that of Word count.

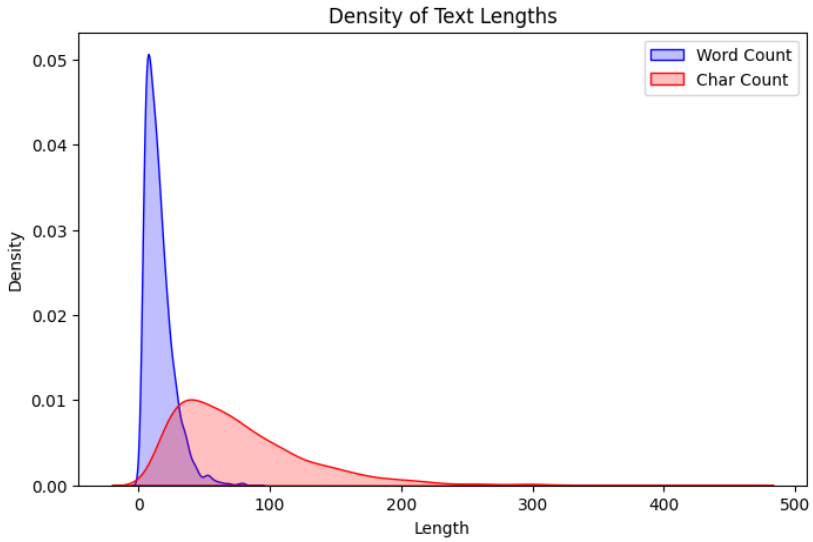

Figure 3: Density of Text Lengths