Toxicity Detection

Guillermo David
James Latanna
Matthias Mangold

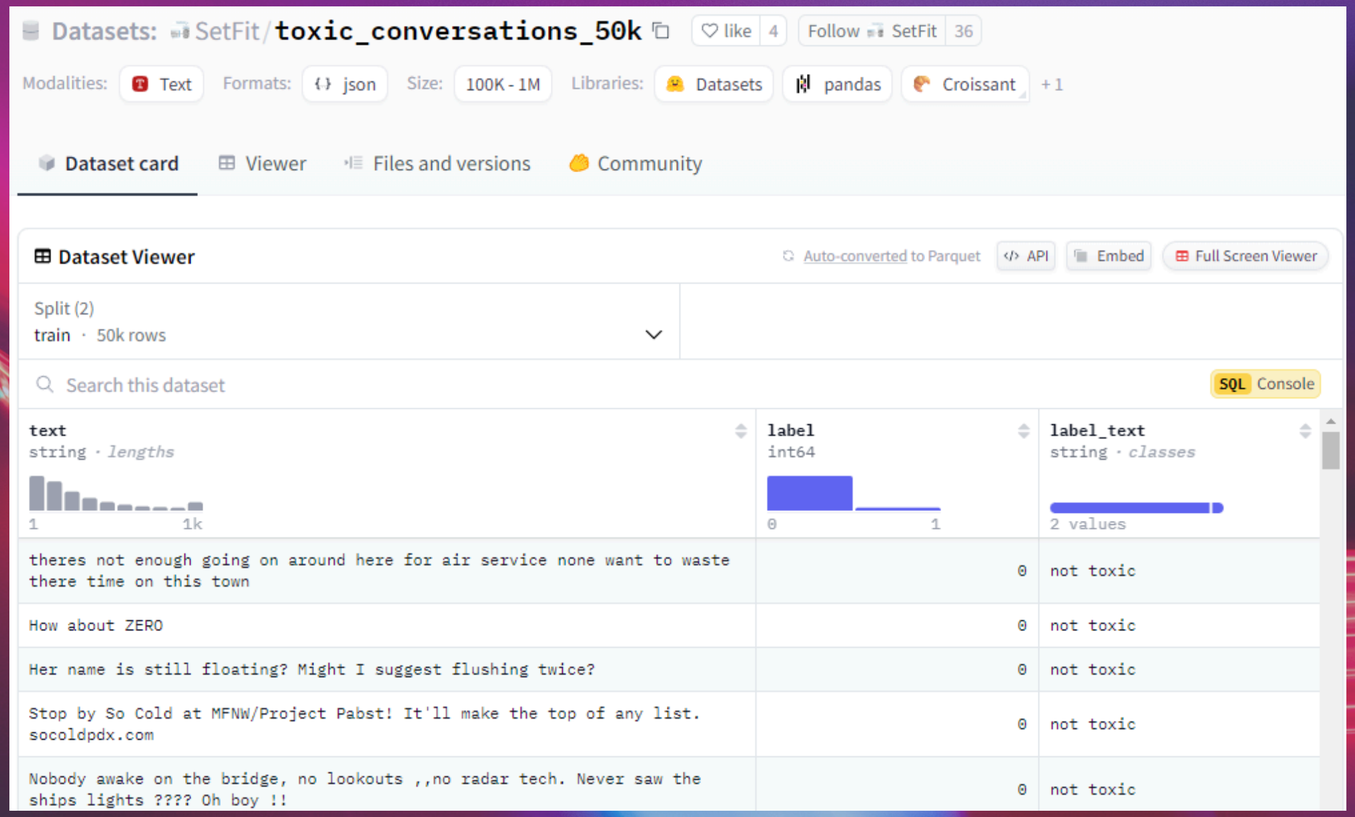
Dataset Selection & Preprocessing

(including EDA - Exploratory Data Analysis)

Dataset Selection:

The dataset is a subset of the Jigsaw Unintended Bias in Toxicity Classification dataset, containing the first 50,000 training examples curated for toxicity classification tasks.

Toxicity is defined when the target value is 0.5 or higher.



Features:

- text: The conversational text
- label: Binary label indicating toxicity
- label_text: Textual representation of the label

Dataset Selection & Preprocessing

Quality of dataset selection:

```python
import pandas as pd
splits = {'train': 'train.jsonl', 'test': 'test.jsonl'}
df = pd.read_json("hf://datasets/SetFit/toxic_conversations_50k/" + splits["train"], lines=True)
```

```python
# General dataset information
print("Dataset Shape:", df.shape)
print("Columns:", df.columns)

# Check for missing values
print("Missing Values:\n", df.isnull().sum())

# Check for duplicates
print("Number of Duplicate Rows:", df.duplicated().sum())
```

```
Dataset Shape: (50000, 3)
Columns: Index(['text', 'label', 'label_text'], dtype='object')
Missing Values:
 text          0
label         0
label_text    0
dtype: int64
Number of Duplicate Rows: 100
```

- 50000 rows, 3 columns
- no missing values
- 100 duplicates

Dataset Selection & Preprocessing

Quality of dataset selection:

Removing duplicates:

```
# Remove duplicate rows
df = df.drop_duplicates().reset_index(drop=True)
print("Dataset Shape After Removing Duplicates:", df.shape)
print("Number of Duplicate Rows:", df.duplicated().sum())
```

```
Dataset Shape After Removing Duplicates: (49900, 3)
Number of Duplicate Rows: 0
```

- Remove duplicates
- Check result

Dataset Selection & Preprocessing

Preprocessing

Preprocessing the Data

```python
import re

# Function to preprocess text
def preprocess_text(text):
    # Convert to lowercase
    text = text.lower()
    # Remove punctuation and special characters
    text = re.sub(r'[^a-z\s]', '', text)
    return text

# Apply preprocessing
df['cleaned_text'] = df['text'].apply(preprocess_text)

# Preview cleaned text
df[['text', 'cleaned_text']].head()
```

	text	cleaned_text
0	theres not enough going on around here for air...	theres not enough going on around here for air...
1	How about ZERO	how about zero
2	Her name is still floating? Might I suggest f...	her name is still floating might i suggest fl...
3	Stop by So Cold at MFNW/Project Pabst! It'll m...	stop by so cold at mfnwproject pabst itll make...
4	Nobody awake on the bridge, no lookouts ,,no r...	nobody awake on the bridge no lookouts no rada...

- Convert to lowercase
- Remove special characters
- Check result

Dataset Selection & Preprocessing

EDA

```python
# Label distribution
label_distribution = df['label'].value_counts(normalize=True) * 100
print("Label Distribution:\n", label_distribution)

# Text length statistics
df['text_length'] = df['text'].apply(len)
print("Text Length Statistics:\n", df['text_length'].describe())
```

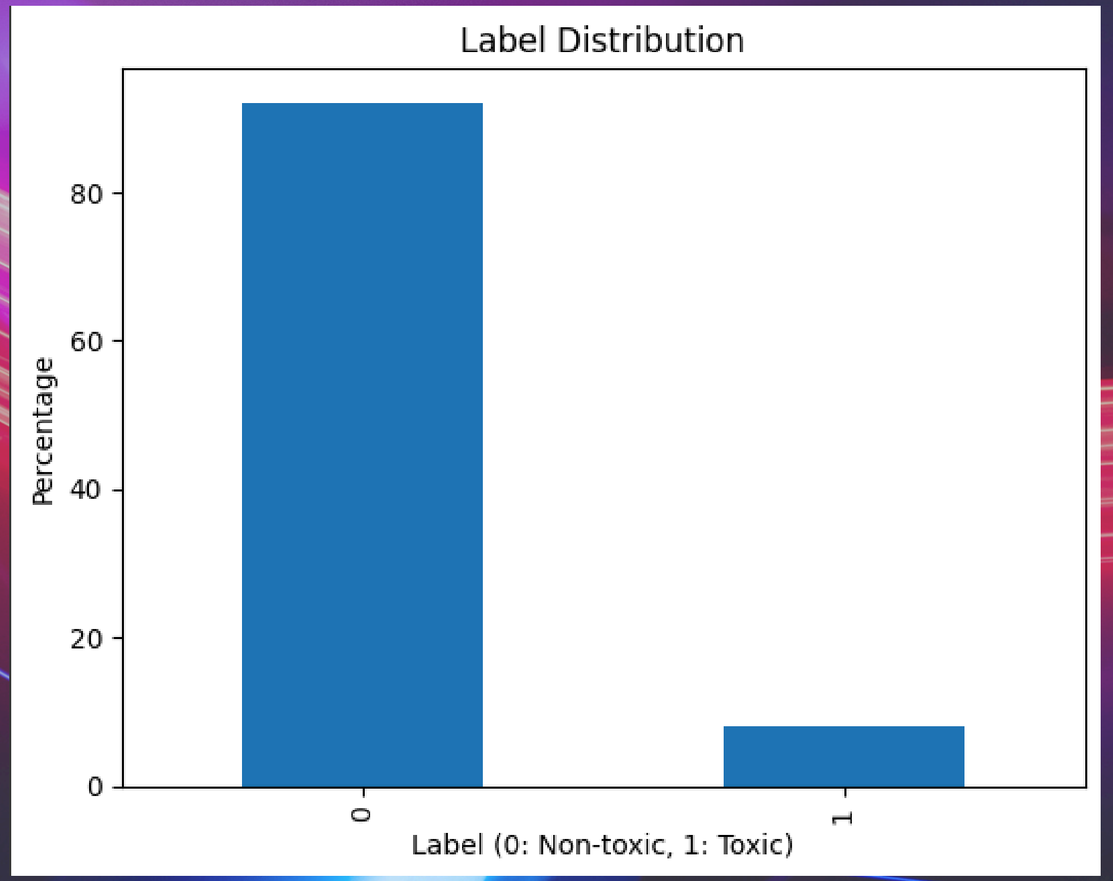
```
Label Distribution:
 label
0    92.056112
1     7.943888
Name: proportion, dtype: float64
Text Length Statistics:
 count    49900.000000
mean       299.286273
std        270.072584
min          1.000000
25%         95.000000
50%        205.000000
75%        415.000000
max       1000.000000
Name: text_length, dtype: float64
```

Around 8% of the dataset is labeled as toxic, making it highly imbalanced.

Label distribution:

Non-toxic: 92.06%

Toxic: 7.94%



Dataset Selection & Preprocessing

EDA

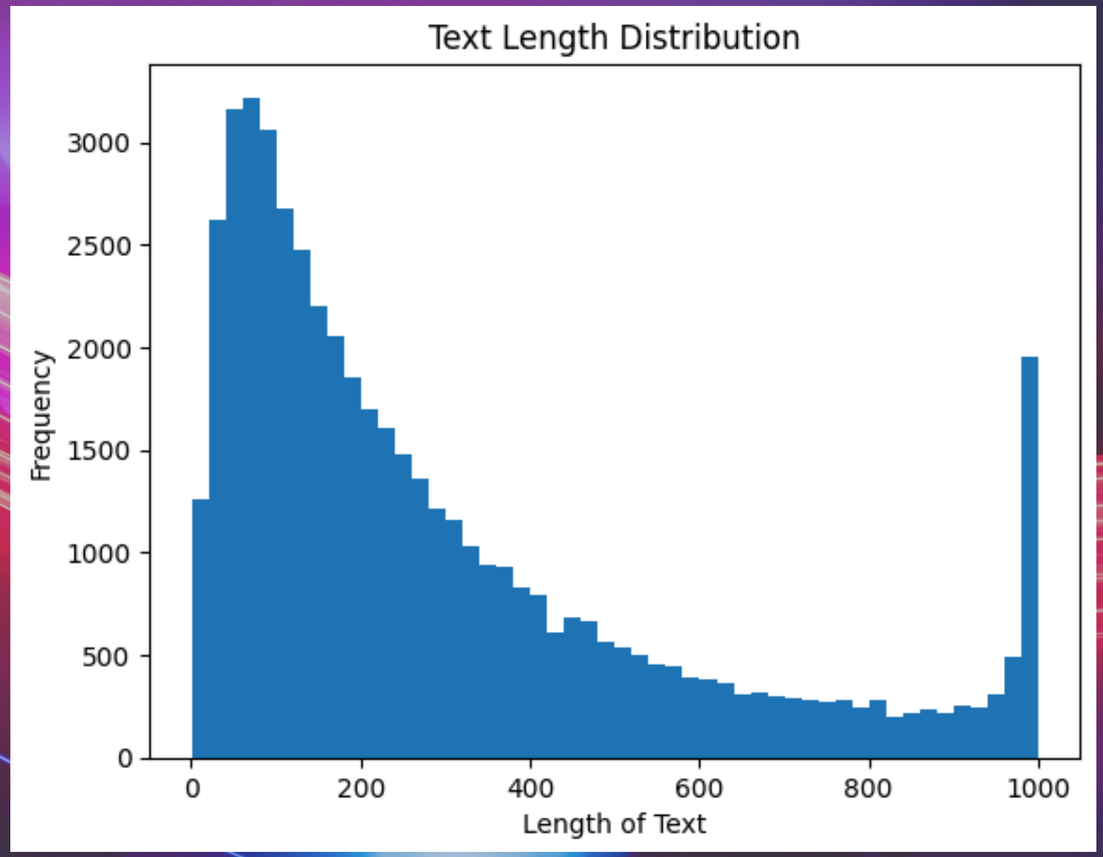
```python
# Label distribution
label_distribution = df['label'].value_counts(normalize=True) * 100
print("Label Distribution:\n", label_distribution)

# Text length statistics
df['text_length'] = df['text'].apply(len)
print("Text Length Statistics:\n", df['text_length'].describe())
```

```
Label Distribution:
 label
0    92.056112
1     7.943888
Name: proportion, dtype: float64
Text Length Statistics:
 count    49900.000000
mean       299.286273
std        270.072584
min          1.000000
25%         95.000000
50%        205.000000
75%        415.000000
max       1000.000000
Name: text_length, dtype: float64
```

The average comment length is 299 characters, with a median of 204 characters. Some comments are as short as 1 character, while others are up to 1,000 characters.

Text length distribution:



Model Description

This model we used is a fine-tuned version of the DistilBERT model to classify toxic comments.

Limitations and Bias

This model is intended to be used for classifying toxic online classifications. However, one limitation of the model is that **it performs poorly for some comments that mention a specific identity subgroup**, like Muslims.

Thank You

www.reallygreatsite.com - hello@reallygreatsite.com