**Social Media Sentiment Analysis and Visualization Report**

**Objective**

The aim of this report is to analyze and visualize social media sentiment data, focusing on understanding sentiment trends and their association with different apps.

**Data Loading and Preparation**

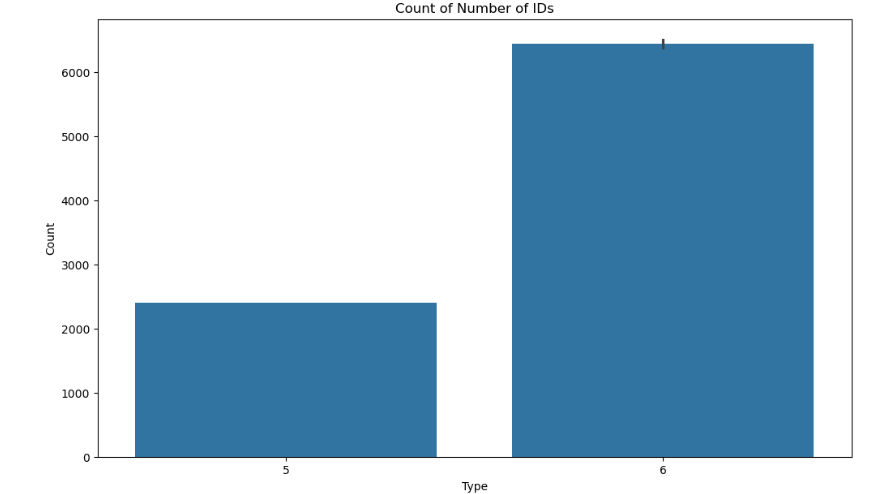
1. **Data Loading**: Imported the training data (twitter\_training.csv) and validation data (twitter\_validation.csv) using *pd.read\_csv.*
2. **Renaming Columns**: Columns were renamed to ['id', 'app', 'sentiment', 'text'] for both datasets to standardize the naming and improve readability.
3. **Initial Data Exploration**:

* **Shape of the Data**: Displayed the number of rows and columns.
* **Column Information**: Used *data.columns* to confirm column names.
* **Summary Statistics**: Called *data.describe(include='all')* to get a summary, including unique values, counts, and descriptive statistics for each column.

**Analysis and Visualization**

1. **ID Distribution**:

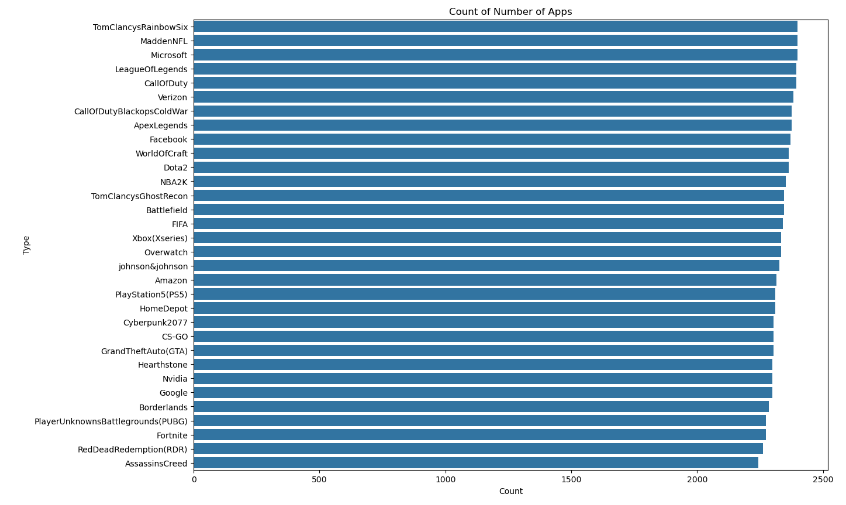
* Checked the distribution of unique IDs with *data['id'].value\_counts()*, followed by a bar plot to show the count for each ID.



* This bar plot highlights the frequency of different IDs, helping to identify any prominent ones in the dataset.

1. **App Distribution**:

* Used *data['app'].value\_counts()* to examine the number of entries per app, visualized with a bar plot to assess the usage of each app.



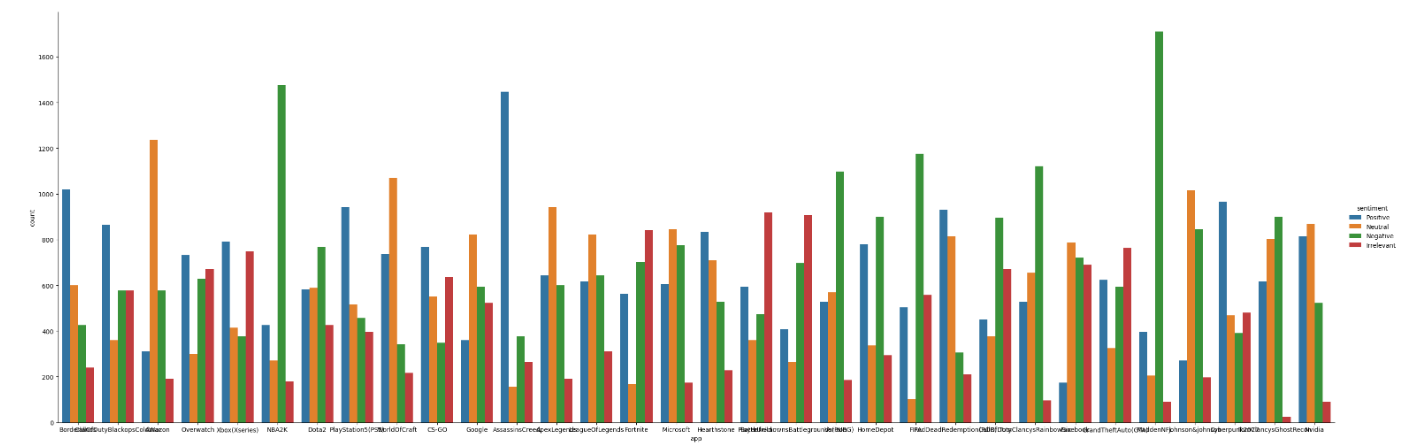
* This plot reveals how frequently each app appears, aiding in understanding app popularity within the dataset.

**Sentiment Distribution by App**

1. **Sentiment Count by App**:

* The following code creates a categorical count plot for each app, showing how sentiment is distributed across different apps.

*sns.catplot(x='app', hue='sentiment', kind='count', height=10, aspect=3, data=data)*



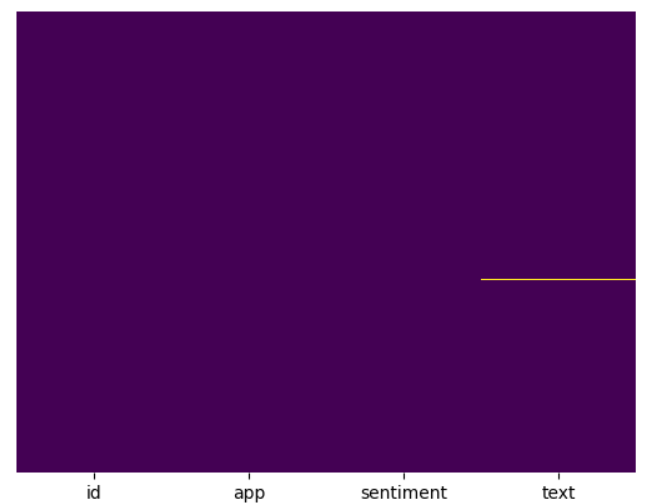
* This helps to see if certain apps have a higher tendency for positive or negative sentiments.

**Missing Data Analysis**

1. **Null Values Check**:

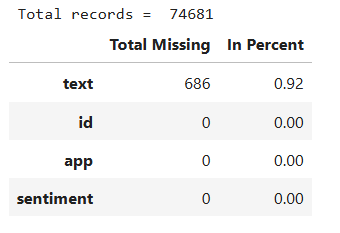
* Used a heatmap to visualize the presence of any missing values in the dataset.

*sns.heatmap(data.isnull(), yticklabels=False, cbar=False, cmap='viridis')*



1. **Quantifying Missing Data**:

* Calculated total and percentage of missing values in each column, displaying the top missing columns.



**Balancing the Dataset**

1. **Sentiment-Based Sampling**:
   * The data was downsampled by taking only a fraction of each sentiment category to balance the dataset.

*train0 = data[data['sentiment'] == "Negative"][:int(train0.shape[0]/12)]*

*train1 = data[data['sentiment'] == "Positive"][:int(train1.shape[0]/12)]*

*train2 = data[data['sentiment'] == "Irrelevant"][:int(train2.shape[0]/12)]*

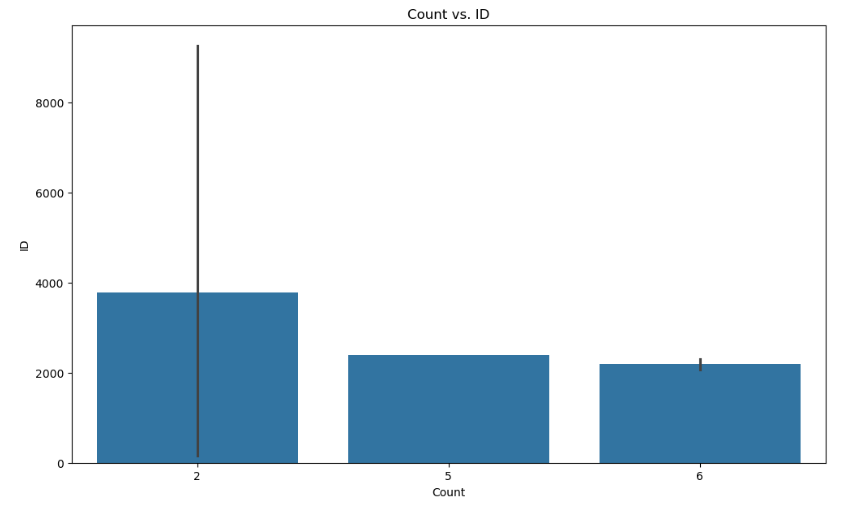
*train3 = data[data['sentiment'] == "Neutral"][:int(train3.shape[0]/12)]*

*data = pd.concat([train0, train1, train2, train3], axis=0)*

**Further Analysis and Visualization**

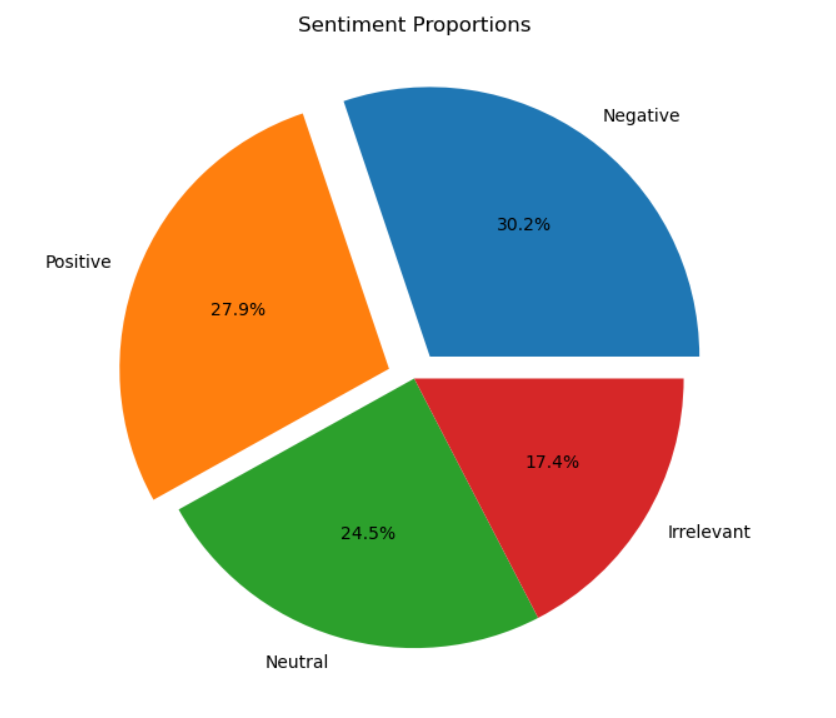
1. **Revised ID and Sentiment Distributions**:

* Plotted ID and app distributions again after downsampling to verify balance.



1. **Sentiment Type Proportion**:

* Generated a pie chart to show the proportions of different sentiment types post-downsampling.



**Encoding and Final Data Preparation**

1. **Label Encoding**:

* Converted categorical values in sentiment and app columns to numeric format to prepare for further analysis or model training.

*label\_encoder = preprocessing.LabelEncoder()*

*data['sentiment'] = label\_encoder.fit\_transform(data['sentiment'])*

*data['app'] = label\_encoder.fit\_transform(data['app'])*

*v\_data['sentiment'] = label\_encoder.fit\_transform(v\_data['sentiment'])*

*v\_data['app'] = label\_encoder.fit\_transform(v\_data['app'])*

1. **Dropping Unnecessary Columns**:

* Dropped the id column as it was not relevant to sentiment analysis.

*data = data.drop(['id'], axis=1)*

**Conclusion**

This analysis revealed key insights into sentiment distributions and app usage in the dataset. Downsampling was critical for balancing sentiment classes, while visualizations helped in understanding patterns in app usage and sentiment types. These steps collectively prepare the dataset for further model training and further analysis.