**Twitter Sentiments Documentation**

1. **Data Exploration:**

The given dataset has 1,600,000 from twitter API. The dataset has 6 columns:

1. Sentiments: This column gives information regarding polarity of the tweet. It has only two values where “0” indicates negative tweets and ”4” indicates positive tweets.
2. ID: It gives detail about the user who tweeted that tweet. It has numeric datatype.
3. Date: It gives exact time and date of the tweet. It is of the format “Day Month HH:MM:SS TimeZone YYYY”.
4. Query: The query (lyx). If there is no query, then this value is NO\_QUERY.
5. User\_id: It contains username of the person who tweeted the tweet.
6. Text: It contains content of the tweet. It can have alphabets, numbers, special characters.
7. **Data Cleaning:**

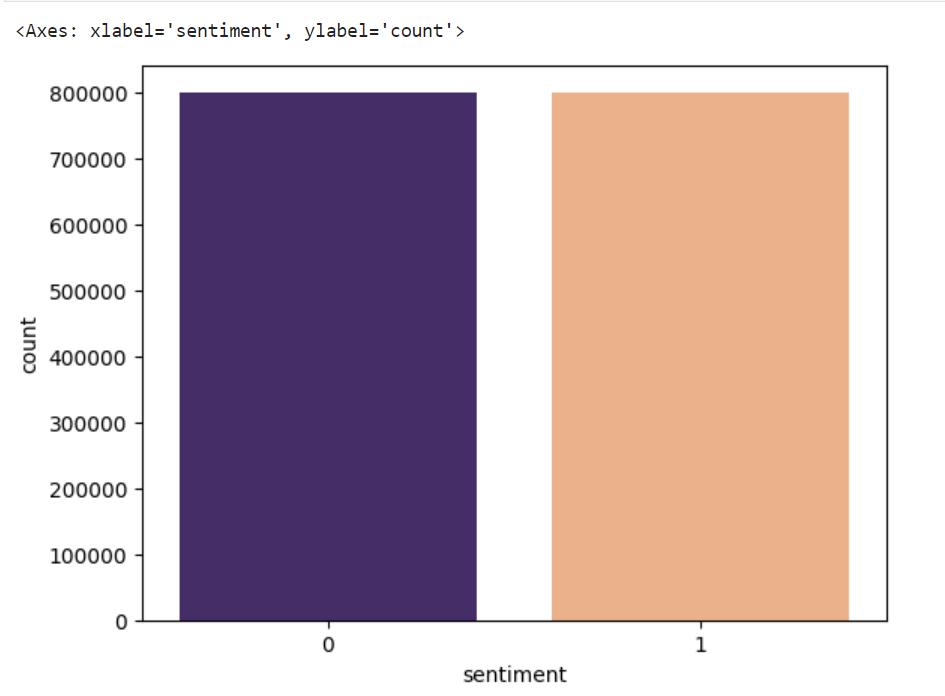
It is not a great idea to have “4” as entry for positive tweets, so replace 4 with 1 in Sentiment column. The columns do not have any missing or None values and duplicate entries, so data cleaning is not important for missing values. Also, query, user\_id and id have hardly any impact on sentiment of tweet, so drop these columns.

1. **Exploratory Data Analysis (EDA):**

The given data has almost equal number of positive and negative sentiments data, so the data is perfectly balanced in order to apply any machine learning algorithm. Also, the dataset has tweets from 6th-April 2009 to 16th-June 2009 and the time zone is PDT. The length of tweet is varying from 6 characters to 374 characters with most of the tweets having 50-150 characters. World cloud can be used to estimate words that have been used most in these tweets and same can be done for tweets having positive sentiments and negative sentiments separately. It can be seen that words such as love, thank, hope, etc have been used in most of the positive sentiments tweets while the words such as work, now, miss, etc have been used majorly in tweets having negative sentiments.

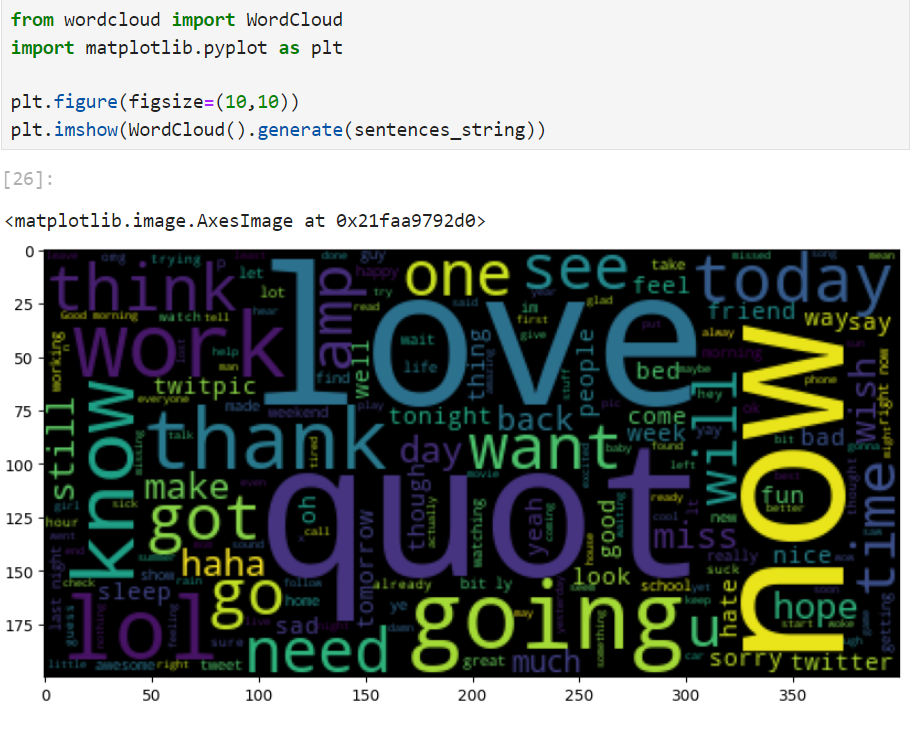
1. **Sentiment Distribution:**

The number of positive and negative sentiments tweets in the dataset is almost equal, so the data is perfectly balanced, so it will make training a model easier because it helps prevent the model from becoming biased towards one class.



1. **Word Frequency Analysis:**

World cloud can be used to estimate words that have been used most in these tweets and same can be done for tweets having positive sentiments and negative sentiments separately. It can be seen that words such as love, thank, hope, etc. have been used in most of the positive sentiments tweets while the words such as work, now, miss, etc. have been used majorly in tweets having negative sentiments. This can be seen in world cloud in Jupyter notebook.







1. **Temporal Analysis:**

The date column of dataset contains some useless information such as year, time zone in context of temporal variation of tweets. Make another column ‘timestamp’ which contains only time of the tweet in the format “HH:MM: SS” , which can be then grouped in terms of the hour of the tweet. It can be seen from chart that most of the tweets are tweeted around 07:00 hour while the variation of positive and negative tweets does not change significantly around time, so time has almost negligible impact on sentiment of tweet.

A graph with lines and numbers

Description automatically generated

1. **Text Preprocessing:**

It is important to remove stop words, special characters, and URLs from the data to get more specific correlation of content of the tweets to its sentiments. So, use stop words of English language from nltk for cleaning for content of tweet.

1. **Sentiment Prediction:**

**Logistic Regression(LR):**

Precision:

Class '0': 0.80 Class '1': 0.79

Recall:

Class '0': 0.79 Class '1': 0.81

F1-Score:

Class '0': 0.79 Class '1': 0.80

Accuracy: 0.80

**Naive Bayes (BernoulliNB):**

Precision:

Class '0': 0.77 Class '1': 0.80

Recall:

Class '0': 0.81 Class '1': 0.76

F1-Score:

Class '0': 0.79 Class '1': 0.78

Accuracy: 0.78

**Naive Bayes (MultinomialNB):**

Precision:

Class '0': 0.76 Class '1': 0.80

Recall:

Class '0': 0.82 Class '1': 0.74

F1-Score:

Class '0': 0.79 Class '1': 0.77

Accuracy: 0.78

Logistic Regression and Naive Bayes (BernoulliNB) have balanced precision and recall for both classes. Naive Bayes (MultinomialNB) has slightly lower precision for class '0' and lower recall for class '1’. F1-scores are similar among the models, indicating a good balance between precision and recall.

1. **Insights:**

Consistent Performance:

Logistic Regression, Naive Bayes (BernoulliNB), and Naive Bayes (MultinomialNB) show similar overall performance with accuracies ranging from 78% to 80%.

Balanced Predictions:

All models exhibit balanced precision and recall for both positive and negative sentiments.

Feature Importance:

Logistic Regression provides insights into feature importance, valuable for understanding key contributors to sentiment predictions.

1. **Recommendations:**
   * + Analyze features from Logistic Regression to identify critical words or phrases influencing sentiment.
     + If dealing with textual features, consider using Naive Bayes (MultinomialNB) designed for such data.
     + Explore hyperparameter tuning to potentially enhance model performance.
     + Integrate user feedback for continuous model improvement.
     + Utilize visualizations like trend charts for a clearer representation of sentiment patterns.
     + Investigate sentiment variations across different topics or categories.
     + Tailor communication strategies based on sentiment insights for more effective messaging.
     + Periodically evaluate and update models to ensure ongoing accuracy.