## **Sentiment Analysis Model Development Report**

## **Introduction**

This report documents the development of a sentiment analysis model. The objective is to predict emotions in tweets based on the provided dataset. The dataset consists of three main components: data\_identification.csv, emotion.csv, and tweets\_DM.json.

## **Data Integration and Cleaning**

The first step is to integrate the information from different files. The data\_identification and emotion datasets are merged based on the common tweet\_id. This ensures that only tweets with emotion labels are considered for training. Missing tweet IDs are also identified.

**1. Data Exploration and Understanding:**

The initial phase involved loading and exploring the provided datasets:

* data\_identification.csv: This file contains information about each tweet, including a unique identifier (tweet\_id) and whether it belongs to the training or testing set.
* emotion.csv: This file links each tweet to an emotion label. It includes the tweet\_id and the corresponding emotion.
* tweets\_DM.json: This file contains detailed information about each tweet in JSON format

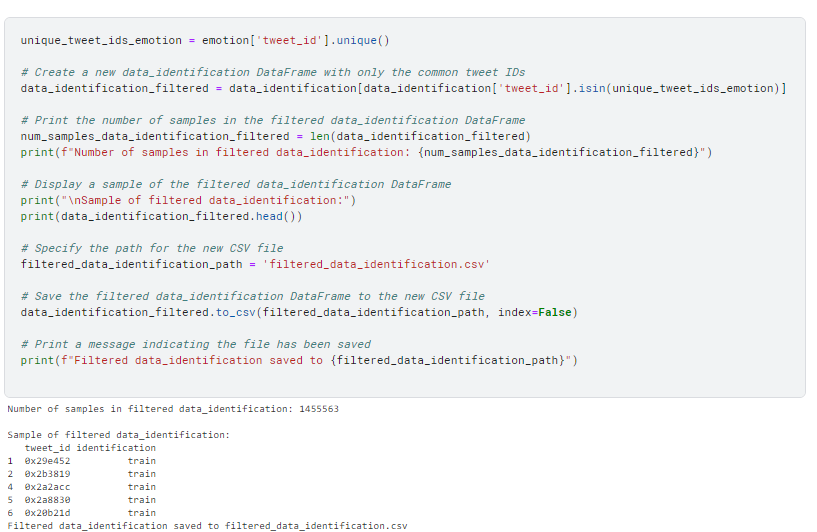
**2. Data Integration and Cleaning:**

As the original data are mixed dataset with test and train identification, data filtration is used to split the data into two .csv file for test and train identification and then merged with similar data\_identification and emotion based on the tweet ID to filter tweets with labeled emotions.

Identified and saved missing tweet IDs.

* The data\_identification and emotion datasets are merged based on the common tweet\_id.

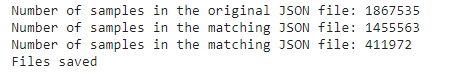
Given below are the snippet for data cleaning.





**3. Text Data Preprocessing:**

After separate csv files are made, the original JSON file (tweets\_DM.json) is then used to extract tweet data for the corresponding csv files. Given below are the number of samples for two separated files.



**4. Feature Engineering:**

For feature engineering, TF-IDF (Term Frequency-Inverse Document Frequency) is used for text vectorization, a crucial step in preparing textual data. Here, the primary feature was derived from the text content using TF-IDF vectorization.

#### 1. **Initialization:**

* The TfidfVectorizer from scikit-learn is initialized with a specified maximum number of features (in this case, 5000). The maximum features represent the top terms with the highest TF-IDF scores.

#### 2. **Training Set Transformation:**

* The training text data (X\_train) is transformed into TF-IDF features using the fit\_transform method of the vectorizer.

#### 3. **Test Set Transformation:**

* The test text data (X\_test) is transformed using the same vectorizer fitted on the training set. This ensures consistent feature representation between training and testing.

#### 4. **Model Training:**

* The TF-IDF vectorized data is then used to train.

**5**.**Model Development:**

### **Model Selection**

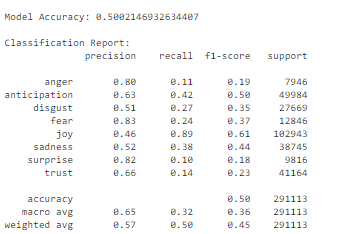
A Multinomial Naive Bayes (NB) classifier is chosen for its simplicity and effectiveness in text classification tasks. Naive Bayes models work well with sparse data, making them suitable for text-based problems.

### **Model Training**

The model is trained using the preprocessed training data, which includes TF-IDF vectorized text features (X\_train\_tfidf) and corresponding emotion labels (y\_train).

**6. Model Evaluation:**

The model is evaluated on a separate test set (X\_test\_tfidf, y\_test). Metrics such as accuracy, precision, recall, and F1-score are computed to assess the model's performance. The model achieve accuracy of 0.50021  
Given below is the result classification.



**7. Model Persistence:**

Saved the trained model using both pickle and joblib libraries for future reuse.

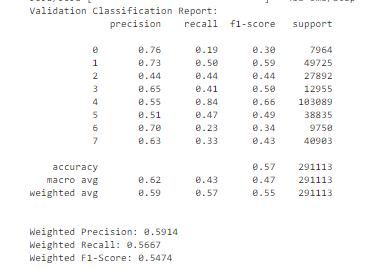
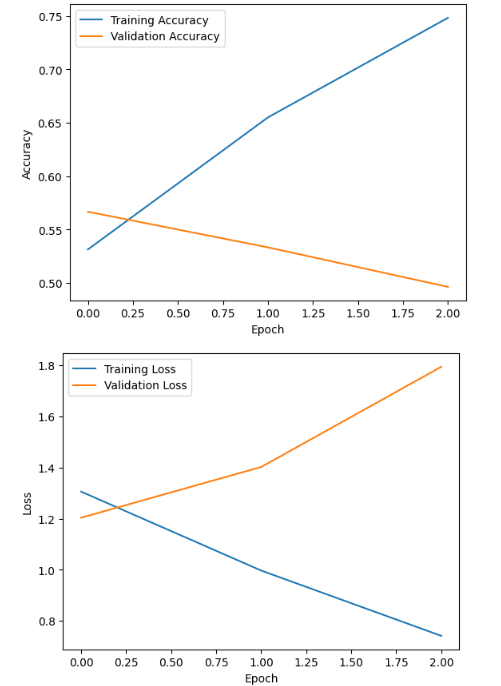
**8. Handling Missing Data**

For tweets without emotion labels, the missing data is loaded, preprocessed, and predictions are made using the trained model. The results, including tweet IDs and predicted emotions, are saved to a CSV file (sentiment\_analysis\_results\_NV.csv).  
  
**9. Conclusion and Future Work**

This report outlines the development of a sentiment analysis model using a Multinomial Naive Bayes classifier. The model demonstrates good performance on the provided dataset. Further improvements could be explored, such as experimenting with different models,exploring advanced feature engineering techniques, and fine-tuning hyperparameters. Additionally, exploring more advanced natural language processing techniques, could be considered for enhanced performance.

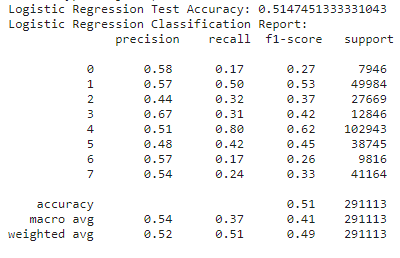
**10. Different method tried and insights you gained:**

* I tried LSTM but the model’s result didn’t give good result. Below is the result from LSTM model

From the above plot it can be seen that LSTM suffer from overfitting. The validation accuracy was more than the training accuracy at the initial stage of training.

* I also tried TF-idf with logistic regression but there was a problem where the classes where given in numeric classes and we need few changes to convert the numeric classes to its corresponding emotion classes and also the testing result was few points lower than the used model. Below is the result of validation evaluation.



In summary, TF-IDF with Naive Bayes emerges as a favorable choice compared to alternative methods. Naive Bayes classifiers, known for their simplicity and computational efficiency, complement the TF-IDF representation well, particularly in scenarios with extensive text datasets and relatively straightforward language patterns. While logistic regression and LSTM offer certain advantages, such as flexibility and the ability to capture sequential dependencies, they come with trade-offs, including increased computational complexity and potential overfitting concerns. TF-IDF with Naive Bayes strikes a balance, demonstrating effectiveness in sentiment analysis tasks by providing accurate predictions, ease of interpretation, and efficient utilization of features derived from term frequency and inverse document frequency.

This report summarizes the entire workflow, from data exploration to model development, and highlights key decisions and steps taken in each phase.