# Emotion Classification on Twitter Using Fine-Tuned DistilBERT

## 1. Introduction

Social media platforms like Twitter offer a wealth of insights into public sentiment and emotions. By automatically classifying these emotions, we can gain a better understanding of public opinion, identify emerging trends, and address societal issues in real time. This project aimed to create and fine-tune a model that accurately predicts emotions in tweets using advanced natural language processing (NLP) techniques. **Large Language Models (LLMs)** are a type of deep learning model specifically designed to comprehend, generate, and manipulate natural language text. These models, trained on extensive text datasets, can perform a variety of language-related tasks, including answering questions, generating content, summarizing information, translating languages, and more. (3)

## 2. Methodology

**2.1) Dataset:** Many datasets are structured as binary classification tasks in sentiment analysis. However, this dataset involves six distinct sentiments, so we'll approach it as a Multi-Class classification problem. To tackle this, we'll use three main libraries from the Hugging Face ecosystem: Datasets, Tokenizers, and Transformers. (2)

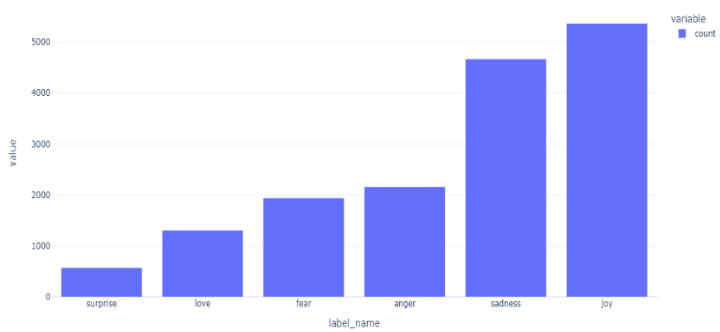


**Figure 1 (2): Class Distribution**

**2.2) Background on Transformer Models:**

Transformer models, such as BERT (Bidirectional Encoder Representations from Transformers), have transformed natural language processing (NLP) by providing a deep, contextual comprehension of text. These models utilize a self-attention mechanism that enables them to evaluate the significance of each word in a sentence in relation to others, capturing intricate dependencies and relationships within the text. BERT, specifically, is trained using a method known as "masked language modeling," where certain words in a sentence are hidden, and the model learns to predict them, thereby developing a profound understanding of language context. (4)

**DistilBERT** is a streamlined version of BERT, designed to maintain much of BERT's advanced language understanding abilities while being more efficient in terms of speed and resource use. This model is 40% smaller, operates 60% faster, and retains approximately 97% of BERT's performance, making it an ideal choice for applications that need a balance between high performance and computational efficiency.

**How DistilBERT Works:** DistilBERT is trained through a process called knowledge distillation, where a smaller model (the student) learns to emulate the behavior of a larger, pre-trained model (the teacher, in this case, BERT). In this process, the student model is trained to replicate the outputs of the teacher model, effectively acquiring the teacher's language understanding abilities in a more compact and efficient form.

### 2.3) Model Design and Implementation

**Model Architecture:** The backbone of the model is the DistilBERT transformer, a smaller and faster version of BERT (Bidirectional Encoder Representations from Transformers). DistilBERT retains most of BERT's language understanding capabilities while being more computationally efficient. On top of DistilBERT, a classifier layer was added to output emotion categories based on the processed tweet embeddings.

**Tokenization and Data Preparation:** Before feeding the tweets into the model, they were tokenized using AutoTokenizer from the Hugging Face library. Tokenization involved converting words into subword tokens that the model could interpret, ensuring that even rare words or slang typical in tweets are understood correctly. The data was further processed to create attention masks, which help the model focus on the meaningful parts of each tweet.

**Training Process:** The model was fine-tuned on a dataset of labeled tweets, with the training process involving backpropagation to minimize a cross-entropy loss function. The training loop was efficiently implemented with the use of PyTorch, which handled the gradient descent and parameter updates.

### 2.4) Fine-Tuning

In the fine-tuning approach, we train the hidden states of a model from a specific starting point, requiring a differentiable classification head. We'll load the DistilBERT model with `AutoModelForSequenceClassification`, which includes a classification head, and specify the number of labels to predict. The model will undergo training for 3 epochs with a learning rate set at 2e-5 and a batch size of 64, utilizing `TrainingArguments`. (1)

**2.5) Evaluation:** The model's performance was evaluated on a validation set, where it predicted the emotional category of unseen tweets. Metrics like accuracy and loss were used to gauge how well the model generalized to new data.

**Table 1: Performance Matrix of the Model**

| Epoch | Training Loss | Validation Loss | Accuracy | F1 |
|---|---|---|---|---|
| 1 | 0.801200 | 0.267229 | 0.920000 | 0.919263 |
| 2 | 0.205400 | 0.179894 | 0.925000 | 0.924990 |
| 3 | 0.143400 | 0.164295 | 0.933000 | 0.933102 |

This table summarizes the performance metrics of a model over three training epochs. As the training progresses through the epochs, both the training and validation losses decrease, while the accuracy and F1 score improve. This suggests that the model is effectively learning from the training data and is becoming more accurate and balanced in its predictions on the validation data.

## 3. Results:

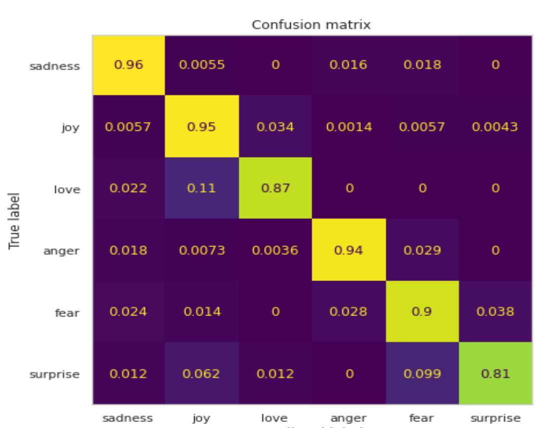

**Figure 4: Confusion matrix of DistilBERT model**

The confusion matrix shows that the fine-tuning approach with DistilBERT significantly outperforms simply extracting embeddings and training a separate machine-learning model. While love is still sometimes mistaken for joy (0.08), this happens far less frequently than with the initial method. Similarly, surprise is occasionally confused with joy (0.09) or fear (0.10), but these rates are also much lower compared to the first approach.

The model is trained using `AutoModelForSequenceClassification`, which added a classification head to the base DistilBERT model. For making predictions on new, unseen data, we can use the pipeline method.

- **Example:**

Tweet: 'I watched a movie last night, it was quite brilliant'.

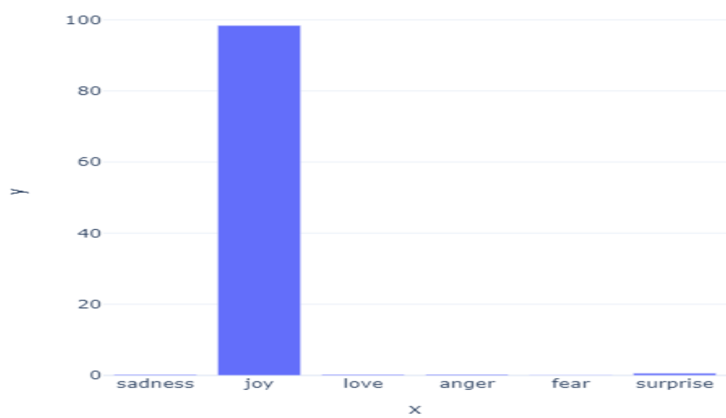


**Figure 5 (2): Prediction Plot**

## 4. Conclusion

This project effectively showcases the use of an advanced NLP model for classifying emotions in tweets. Although the model performs well, particularly with the implementation of transfer learning through DistilBERT, there are opportunities for further improvement (4). Future efforts could concentrate on expanding the dataset, fine-tuning the model, and conducting more thorough evaluations to strengthen the model's robustness and reliability across various real-world scenarios.

## 5. References


1) https://huggingface.co/learn/nlp-course/chapter7/3?fw=pt
2) https://www.kaggle.com/code/shtrausslearning/twitter-emotion-classification/notebook
3) https://www.techopedia.com/definition/34948/large-language-model-llm
4) https://towardsdatascience.com/distilbert-11c8810d29fc