PROJECT - 3

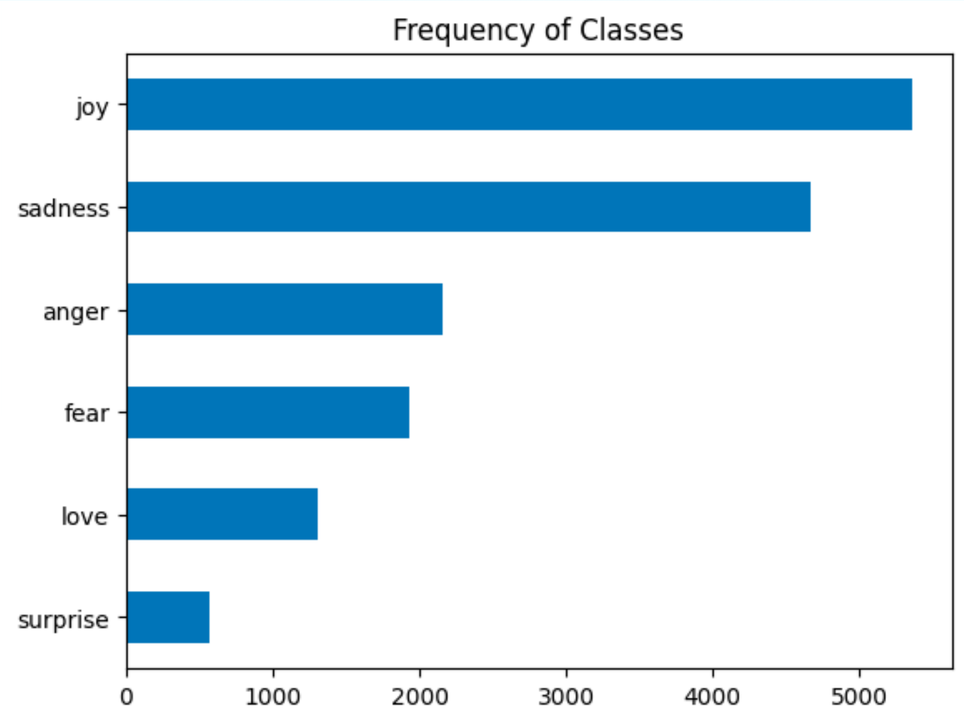
Affective Computing

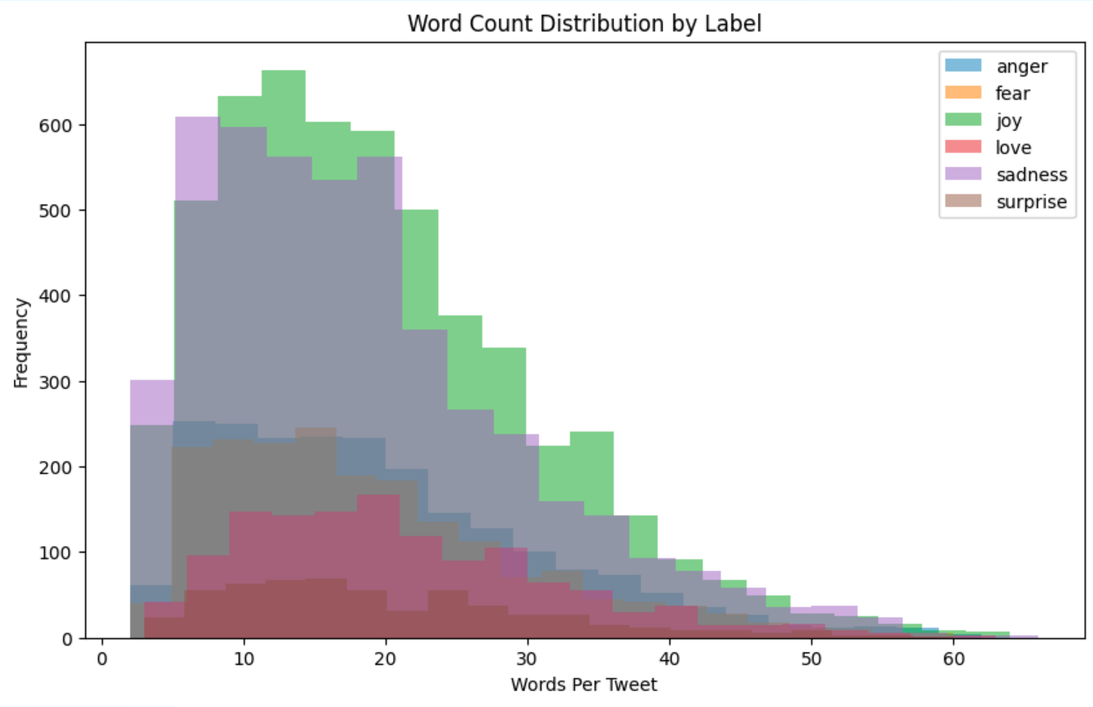
Group -YR

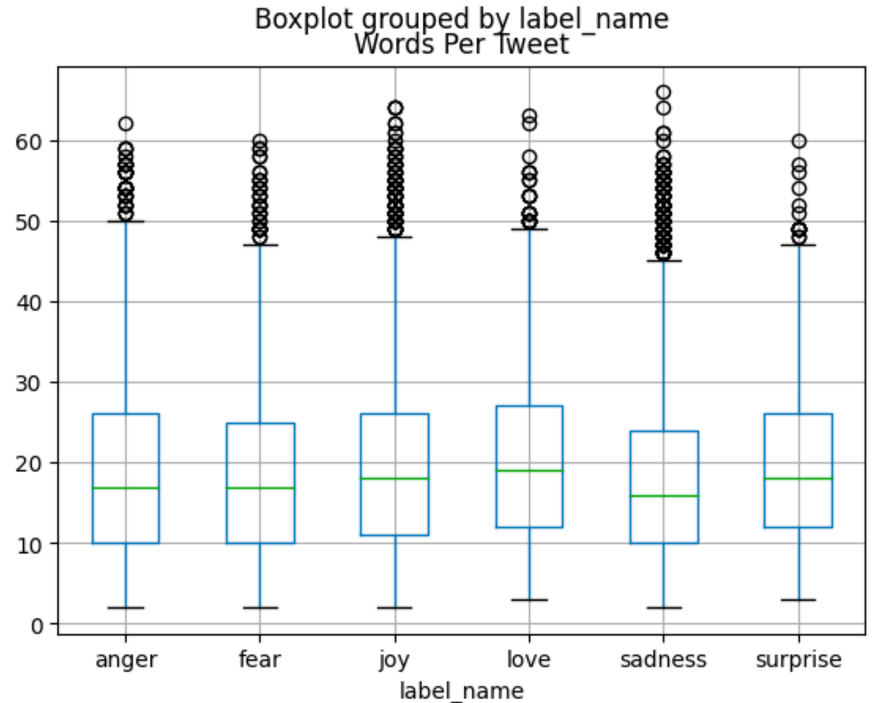
EMOTIONAL ANALYSIS

- In this analysis we tend to segregate the sentences based on different emotions like joy, anger, etc.
- We also found at what accuracy or how correctly our model segregate the sentences based on emotions.
- We access the dataset given using ‘dair-ai/emotion’

Data Preparation







Text Tokenization and Model

Tokenize text using DistilBERT tokenizer.

We use ‘map’ method of the ‘emotion’ dataset.

We use AutoModelForSequenceClassification for sequence classification.

```python
from sklearn.metrics import accuracy_score, f1_score

def compute_metrics(pred):
    labels = pred.label_ids
    preds = pred.predictions.argmax(-1)
    f1 = f1_score(labels, preds, average='weighted')
    acc = accuracy_score(labels, preds)
    return {"accuracy": acc, "f1": f1}
```

RESULT

```
['sadness', 'joy', 'love', 'anger', 'fear', 'surprise']
              precision    recall  f1-score   support

           0       0.95      0.96      0.95       581
           1       0.93      0.94      0.94       695
           2       0.77      0.79      0.78       159
           3       0.93      0.92      0.93       275
           4       0.89      0.87      0.88       224
           5       0.74      0.70      0.72        66

    accuracy                           0.91      2000
   macro avg       0.87      0.86      0.86      2000
weighted avg       0.91      0.91      0.91      2000
```

```
{'test_loss': 0.2207542359828949,
 'test_accuracy': 0.9135,
 'test_f1': 0.9133188734875534,
 'test_runtime': 3.547,
 'test_samples_per_second': 563.861,
 'test_steps_per_second': 9.022}
```

```
surprise     572
love        1304
fear        1937
anger       2159
sadness     4666
joy         5362
Name: label_name, dtype: int64
```

Let's test by our input

```
text = 'i will kiss u'
input_encoded = tokenizer(text, return_tensors='pt').to(device)
with torch.no_grad():
  outputs = model(**input_encoded)

logits = outputs.logits
pred = torch.argmax(logits, dim=1).item()
pred, classes[pred]
```

```
(2, 'love')
```

```
[ ] outputs
```

```
SequenceClassifierOutput(loss=None, logits=tensor([[-0.4150,  0.7472,  1.8437, -0.5844, -0.9316, -1.2663]],
       device='cuda:0'), hidden_states=None, attentions=None)
```