Name: 莊智宇

Student ID: 113065540

Final Result: (Use BERTweet model: https://github.com/VinAIResearch/BERTweet)

7	—	yuuch3		0.55842	20	3d

---

1. **In the beginning, TfidfVectorizer was used for embedding, and MultinomialNB was used for prediction.**

   - Preprocessing:

```
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
import re

def clean_text(text):

    # Convert the text to lowercase
    text = text.lower()

    # Remove the tag
    text = re.sub(r'@\w+', '', text)
    text = re.sub(r'<lh>', '', text)

    # Token
    tokenized_text = word_tokenize(text)

    # Remove the stopwords
    stop_words = set(stopwords.words('english'))
    tokenized_text = [" ".join(word for word in tokenized_text if word not in stop_words)]

    # Do the lemmatization
    lemmatizer = WordNetLemmatizer()
    tokenized_text = " ".join(lemmatizer.lemmatize(word) for word in tokenized_text)

    return tokenized_text
```

   - Use TF-IDF to do embedding:

```
# Use TF-IDF
from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer_tfidf = TfidfVectorizer()
train_emotion_data_tfidf = vectorizer_tfidf.fit_transform(train_emotion_data.cleaned_text)
train_emotion_list = train_emotion_data['emotion'].tolist()
```

   - MultinomialNB to predict:

```
from sklearn.naive_bayes import MultinomialNB

nb_classifier_multi = MultinomialNB()
nb_classifier_multi.fit(train_emotion_data_tfidf, train_emotion_list)

test_emo_pred = nb_classifier_multi.predict(test_emotion_data_tfidf)
```

   - Result: (Left one is Private Score, Right one is Public)

output.csv Complete · 18d ago		0.28459	0.29736

2. **Next, Decision Tree was used for prediction.**
   - The same preprocessing steps were applied, using TF-IDF for embedding, and finally using Decision Tree for prediction:

```
from sklearn.tree import DecisionTreeClassifier

## build DecisionTree model
DT_model = DecisionTreeClassifier(random_state=1)

## training!
DT_model = DT_model.fit(train_emotion_data_tfidf, train_emotion_list)

## predict!
test_emo_pred = DT_model.predict(test_emotion_data_tfidf)

## so we get the pred result
test_emo_pred[:10]
```

   - Result: (Left one is Private Score, Right one is Public)

3. **Changed to use BERT.**
   - Preprocessing:

```
import re

def clean_text(text):

    # Remove the tag
    text = re.sub(r'@\w+', '', text)
    text = re.sub(r'<LH>', '', text)

    return text
```

   - Use google-bert/bert-base-uncased and BertTokenizer to do embedding.
   - learning rate = 3e-5, epochs = 1, batch_size = 64
   - Use google-bert/bert-base-uncased and BertForSequenceClassification to predict.
   - Result: (Left one is Private Score, Right one is Public)

4. **Final Version: Use BERTweet**
   - https://github.com/VinAIResearch/BERTweet
   - A pre-trained language model for English Tweets.
   - My code: https://github.com/yuuch3/DM2024-Lab2-Homework/blob/main/DM2024-Lab2-Homework.ipynb (Third Part)
   - Final Result: (Left one is Private Score, Right one is Public)