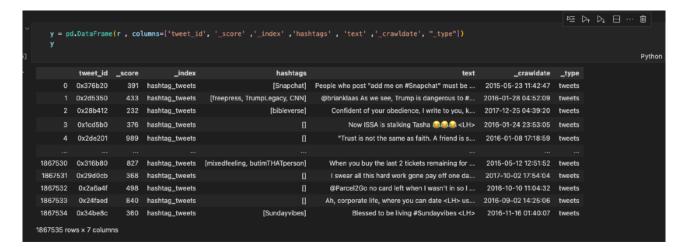
At the part1, I install all the package we need At the part2, I load all the data in pandas data frame



At the part3, I have the data preprocess.

First, I split the data to train and test, and drop some column like hashtags, _type...etc, that I don't know how to utilize in trainning.



I load the pre-train model distilbert to train our model to do the classification.

I use the label encoder to transform the label to a number ,after that, transform the number to one hot encoding.

```
MODEL_NAME = 'distilbert-base-uncased'
from transformers import AutoTokenizer
tokenizer = AutoTokenizer.from_pretrained(MODEL_NAME)

def preprocess(dataslice):

# [ TODO ] use your tokenizor and encoder to get sentence embeddings and encoded labels
tok = tokenizer(dataslice["text"])
tmp = labelencoder.fit_transform(dataslice['emotion'])
label = encoder.fit_transform(tmp.reshape(-1,1)).toarray()
tok['label'] = label
return tok
```

After preprocessing, we have the data below

```
processed_data

Python

DatasetDict({
    train: Dataset({
        features: ['tweet_id', 'emotion', '_score', 'text', '_crawldate', '__index_level_0_', 'input_ids', 'attention_mask', 'label'],
        num_rows: 1455563
    })
})
}
```

Split training data to be training and val

```
train_val_dataset = processed_data['train'].train_test_split(test_size=0.1)
print(train_val_dataset)

DatasetDict({
    train: Dataset({
        features: ['tweet_id', 'emotion', '_score', 'text', '_crawldate', '__index_level_0_', 'input_ids', 'attention_mask', 'label'],
        num_rows: 1310006
})

test: Dataset({
    features: ['tweet_id', 'emotion', '_score', 'text', '_crawldate', '__index_level_0_', 'input_ids', 'attention_mask', 'label'],
        num_rows: 145557
})
})
```

At part4, I set the args for training model.

At first, we set the learn rate to be constant. It will not bring us the better result. So, I change the learn rate to be Adam optimizer that will change when training. After that ,we get a better loss

At the end part5, it did the predict.

Because of how big the dataset are, I do the predict one by one so the kernel won't crash.

Train cost 8 hr and predict also cost 8 hr.