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

```python
y = pd.DataFrame(r , columns=['tweet_id', '_score' ,'_index' ,'hashtags' , 'text' ,'_crawldate', "_type"])
y
```

Python

	tweet_id	_score	_index	hashtags	text	_crawldate	_type
0	0x376b20	391	hashtag_tweets	[Snapchat]	People who post "add me on #Snapchat" must be ...	2015-05-23 11:42:47	tweets
1	0x2d5350	433	hashtag_tweets	[freepress, TrumpLegacy, CNN]	@brianklaas As we see, Trump is dangerous to #...	2016-01-28 04:52:09	tweets
2	0x28b412	232	hashtag_tweets	[bibleverse]	Confident of your obedience, I write to you, k...	2017-12-25 04:39:20	tweets
3	0x1cd5b0	376	hashtag_tweets	[]	Now ISSA is stalking Tasha 😂😂😂 <LH>	2016-01-24 23:53:05	tweets
4	0x2de201	989	hashtag_tweets	[]	"Trust is not the same as faith. A friend is s...	2016-01-08 17:18:59	tweets
...	...	...	...	...	...	...	...
1867530	0x316b80	827	hashtag_tweets	[mixedfeeling, butimTHATperson]	When you buy the last 2 tickets remaining for ...	2015-05-12 12:51:52	tweets
1867531	0x29d0cb	368	hashtag_tweets	[]	I swear all this hard work gone pay off one da...	2017-10-02 17:54:04	tweets
1867532	0x2a6a4f	498	hashtag_tweets	[]	@Parcel2Go no card left when I wasn't in so I ...	2016-10-10 11:04:32	tweets
1867533	0x24faed	840	hashtag_tweets	[]	Ah, corporate life, where you can date <LH> us...	2016-09-02 14:25:06	tweets
1867534	0x34be8c	360	hashtag_tweets	[Sundayvibes]	Blessed to be living #Sundayvibes <LH>	2016-11-16 01:40:07	tweets

1867535 rows × 7 columns

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.

	tweet_id	emotion	_score	text	_crawldate
0	0x29e452	joy	809	Huge Respect @JohnnyVegasReal talking about I...	2015-01-17 03:07:03
1	0x2b3819	joy	808	Yoooo we hit all our monthly goals with the ne...	2016-07-02 09:34:06
2	0x2a2acc	trust	16	@KIDSNTS @PICU_BCH @uhbcomms @BWCHBoss Well do...	2016-08-15 18:18:39
3	0x2a8830	joy	768	Come join @ambushman27 on #PUBG while he striv...	2017-02-11 08:49:46
4	0x20b21d	anticipation	70	@fanshixieen2014 Blessings!My #strength little...	2016-11-23 05:37:10
...	...	...	...	...	...
1455558	0x227e25	disgust	361	@BBCBreaking Such an inspirational talented pe...	2016-09-09 14:28:19
1455559	0x293813	sadness	15	And still #libtards won't get off the guy's ba...	2017-02-04 14:15:32
1455560	0x1e1a7e	joy	174	When you sow #seeds of service or hospitality ...	2015-12-03 16:53:39
1455561	0x2156a5	trust	515	@lorettalrose Will you be displaying some <LH>...	2016-10-27 03:23:51
1455562	0x2bb9d2	trust	850	Lord, I <LH> in you.	2016-08-26 08:41:46

1455563 rows × 5 columns

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.

```python
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

```python
processed_data
```

```
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

```python
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

```python
OUTPUT_DIR = ".//model/"
optimizer = Adam(model.parameters(),
                 betas = (0.9, 0.98),
                 eps = 1.0e-9)
LEARNING_RATE = optimizer
# LEARNING_RATE = 2e-5

BATCH_SIZE = 30
EPOCH = 6
SAVE_LIMIT = 5
training_args = TrainingArguments(
    output_dir = OUTPUT_DIR,
#     learning_rate = LEARNING_RATE
    per_device_train_batch_size = BATCH_SIZE,
    per_device_eval_batch_size = BATCH_SIZE,
    num_train_epochs = EPOCH,
    save_total_limit  = SAVE_LIMIT
    # you can set more parameters here if you want
)

# now give all the information to a trainer
trainer = Trainer(

    model,
    training_args,
    train_dataset=train_val_dataset ["train"],
    eval_dataset=train_val_dataset ["test"],
    # 可以省略，默认的data_collator就是DataCollatorWithPadding
    data_collator=data_collator,
    tokenizer=tokenizer,
```

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.

```python
test_data = pd.merge(test,y)

tmp = [test_data['text'][i] for i in range(len(test_data['text']))]
input = tokenizer( tmp  , truncation=True, padding=True , return_tensors="pt")
# Get the output
logits = mymodel(**input).logits
logits

# get predictions

predicts = nn.functional.softmax(logits, dim = -1)
predicts
```

Train cost 8 hr and predict also cost 8 hr.