1. Necessary Libraries

```
PKG_DIR="/content/drive/MyDrive/offline_packages"

!pip install --no-index --find-links="$PKG_DIR" transformers datasets tokenizers safetensors accelerate evaluate
```

```
Looking in links: /content/drive/MyDrive/offline_packages
Requirement already satisfied: transformers in /usr/local/lib/python3.12/dist-packages (4.57.3)
Requirement already satisfied: datasets in /usr/local/lib/python3.12/dist-packages (4.0.0)
Requirement already satisfied: tokenizers in /usr/local/lib/python3.12/dist-packages (0.22.1)
Requirement already satisfied: safetensors in /usr/local/lib/python3.12/dist-packages (0.7.0)
Requirement already satisfied: accelerate in /usr/local/lib/python3.12/dist-packages (1.12.0)
Processing ./drive/MyDrive/offline_packages/evaluate-0.4.6-py3-none-any.whl
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-packages (from transformers) (3.20.0)
Requirement already satisfied: huggingface-hub<1.0,>=0.34.0 in /usr/local/lib/python3.12/dist-packages (from transformers)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.12/dist-packages (from transformers) (2.0.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from transformers) (25.0)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.12/dist-packages (from transformers) (6.0.3)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.12/dist-packages (from transformers) (2025.11.3
Requirement already satisfied: requests in /usr/local/lib/python3.12/dist-packages (from transformers) (2.32.4)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.12/dist-packages (from transformers) (4.67.1)
Requirement already satisfied: pyarrow>=15.0.0 in /usr/local/lib/python3.12/dist-packages (from datasets) (18.1.0)
Requirement already satisfied: dill<0.3.9,>=0.3.0 in /usr/local/lib/python3.12/dist-packages (from datasets) (0.3.8)
Requirement already satisfied: pandas in /usr/local/lib/python3.12/dist-packages (from datasets) (2.2.2)
Requirement already satisfied: xxhash in /usr/local/lib/python3.12/dist-packages (from datasets) (3.6.0)
Requirement already satisfied: multiprocess<0.70.17 in /usr/local/lib/python3.12/dist-packages (from datasets) (0.70.16)
Requirement already satisfied: fsspec<=2025.3.0,>=2023.1.0 in /usr/local/lib/python3.12/dist-packages (from fsspec[http]<=
Requirement already satisfied: psutil in /usr/local/lib/python3.12/dist-packages (from accelerate) (5.9.5)
Requirement already satisfied: torch>=2.0.0 in /usr/local/lib/python3.12/dist-packages (from accelerate) (2.9.0+cu126)
Requirement already satisfied: aiohttp!=4.0.0a0,!=4.0.0a1 in /usr/local/lib/python3.12/dist-packages (from fsspec[http]<=2
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.12/dist-packages (from huggingface-hub
Requirement already satisfied: hf-xet<2.0.0,>=1.1.3 in /usr/local/lib/python3.12/dist-packages (from huggingface-hub<1.0,>
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests->transfo
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests->transformers) (3.11
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests->transformers)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests->transformers)
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->accelerate) (75.2
Requirement already satisfied: sympy>=1.13.3 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->accelerate) (1
Requirement already satisfied: networkx>=2.5.1 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->accelerate)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->accelerate) (3.1.6)
Requirement already satisfied: nvidia-cuda-nvrtc-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0
Requirement already satisfied: nvidia-cuda-runtime-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch>=2
Requirement already satisfied: nvidia-cuda-cupti-cu12==12.6.80 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0
Requirement already satisfied: nvidia-cudnn-cu12==9.10.2.21 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0-
Requirement already satisfied: nvidia-cublas-cu12==12.6.4.1 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0-
Requirement already satisfied: nvidia-cufft-cu12==11.3.0.4 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->
Requirement already satisfied: nvidia-curand-cu12==10.3.7.77 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0
Requirement already satisfied: nvidia-cusolver-cu12==11.7.1.2 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.
Requirement already satisfied: nvidia-cusparse-cu12==12.5.4.2 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.
Requirement already satisfied: nvidia-cusparselt-cu12==0.7.1 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0
Requirement already satisfied: nvidia-nccl-cu12==2.27.5 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->acc
Requirement already satisfied: nvidia-nvshmem-cu12==3.3.20 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->
Requirement already satisfied: nvidia-nvtx-cu12==12.6.77 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->ac
Requirement already satisfied: nvidia-nvjitlink-cu12==12.6.85 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.
Requirement already satisfied: nvidia-cufile-cu12==1.11.1.6 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0-
Requirement already satisfied: triton==3.5.0 in /usr/local/lib/python3.12/dist-packages (from torch>=2.0.0->accelerate) (3
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas->datasets) (
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas->datasets) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas->datasets) (2025.2)
Requirement already satisfied: aiohappyeyeballs>=2.5.0 in /usr/local/lib/python3.12/dist-packages (from aiohttp!=4.0.0a0,!
Requirement already satisfied: aiosignal>=1.4.0 in /usr/local/lib/python3.12/dist-packages (from aiohttp!=4.0.0a0,!=4.0.0a
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.12/dist-packages (from aiohttp!=4.0.0a0,!=4.0.0a1->
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.12/dist-packages (from aiohttp!=4.0.0a0,!=4.0.0
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from transformers import AutoTokenizer, AutoConfig, AutoModelForSequenceClassification, TrainingArguments, Trainer
from datasets import Dataset, DatasetDict
```

2. EDA

```
df = pd.read_csv("https://raw.githubusercontent.com/laxmimerit/All-CSV-ML-Data-Files-Download/master/twitter_multi_class_se
df.head()
```

	text	label	label_name
0	i didnt feel humiliated	0	sadness
1	i can go from feeling so hopeless to so damned...	0	sadness
2	im grabbing a minute to post i feel greedy wrong	3	anger
3	i am ever feeling nostalgic about the fireplac...	2	love
4	i am feeling grouchy	3	anger

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 16000 entries, 0 to 15999
Data columns (total 3 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   text        16000 non-null  object
 1   label       16000 non-null  int64 
 2   label_name  16000 non-null  object
dtypes: int64(1), object(2)
memory usage: 375.1+ KB
```

```
df.isna().sum()
```

	0
text	0
label	0
label_name	0

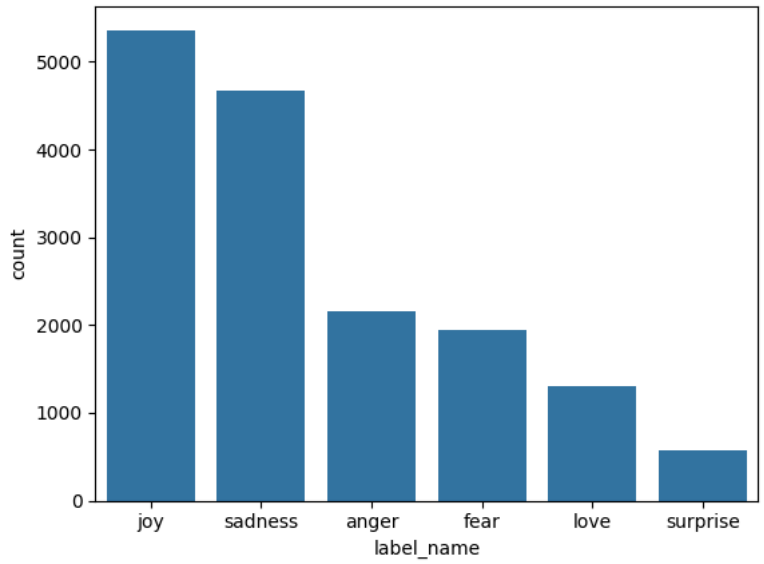
dtype: int64

```
df['label_name'].value_counts()
```

label_name	count
joy	5362
sadness	4666
anger	2159
fear	1937
love	1304
surprise	572

dtype: int64

```
sns.barplot(x=df['label_name'].value_counts().index,y=df['label_name'].value_counts())
plt.show()
```



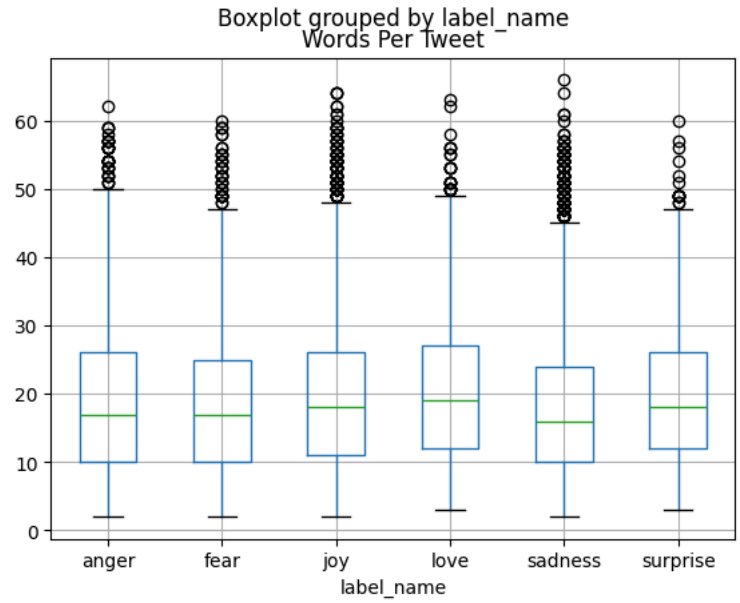
```
df['Words Per Tweet'] = df['text'].apply(lambda x:len(x.split()))
df.head()
```

	text	label	label_name	Words Per Tweet
0	i didnt feel humiliated	0	sadness	4
1	i can go from feeling so hopeless to so damned...	0	sadness	21
2	im grabbing a minute to post i feel greedy wrong	3	anger	10
3	i am ever feeling nostalgic about the fireplac...	2	love	18
4	i am feeling grouchy	3	anger	4

```
df.groupby('label_name')['Words Per Tweet'].describe()
```

label_name	count	mean	std	min	25%	50%	75%	max
anger	2159.0	19.229736	11.426926	2.0	10.0	17.0	26.0	62.0
fear	1937.0	18.844605	10.890188	2.0	10.0	17.0	25.0	60.0
joy	5362.0	19.498135	10.772118	2.0	11.0	18.0	26.0	64.0
love	1304.0	20.700153	10.827956	3.0	12.0	19.0	27.0	63.0
sadness	4666.0	18.361980	11.037223	2.0	10.0	16.0	24.0	66.0
surprise	572.0	19.970280	11.031756	3.0	12.0	18.0	26.0	60.0

```
df.boxplot('Words Per Tweet',by='label_name')
plt.show()
```



3.Stratified Train-test-validation Split

```
from sklearn.model_selection import train_test_split

train , test = train_test_split(df,test_size=0.3,random_state=42,stratify=df['label_name'])
test , validation = train_test_split(test,test_size=1/3,random_state=42,stratify=test['label_name'])

print(train.shape)
print(test.shape)
print(validation.shape)
```

```
(11200, 4)
(3200, 4)
(1600, 4)
```

4.Conversion to HF Dataset

```
dataset = DatasetDict({
    'train':Dataset.from_pandas(train,preserve_index=False),
    'test':Dataset.from_pandas(test,preserve_index=False),
    'validation':Dataset.from_pandas(validation,preserve_index=False)
})
```

```
dataset
```

```
DatasetDict({
    train: Dataset({
        features: ['text', 'label', 'label_name', 'Words Per Tweet'],
        num_rows: 11200
    })
    test: Dataset({
        features: ['text', 'label', 'label_name', 'Words Per Tweet'],
        num_rows: 3200
    })
    validation: Dataset({
        features: ['text', 'label', 'label_name', 'Words Per Tweet'],
        num_rows: 1600
    })
})
```

5.label2id , id2label

```
label2id = {x['label_name']:x['label'] for x in dataset['train']}
id2label = {v:k for k,v in label2id.items()}

label2id, id2label
print(id2label)
print(label2id)
```

```
{0: 'sadness', 1: 'joy', 4: 'fear', 5: 'surprise', 3: 'anger', 2: 'love'}
{'sadness': 0, 'joy': 1, 'fear': 4, 'surprise': 5, 'anger': 3, 'love': 2}
```

6. Model Selection

```
labels = df['label_name'].unique().tolist()
labels
```

```
['sadness', 'anger', 'love', 'surprise', 'fear', 'joy']
```

```
model_dir = '/content/drive/MyDrive/offline_models/bert-base-uncased'

config = AutoConfig.from_pretrained(model_dir,local_files_only=True,id2label=id2label,label2id=label2id,num_classes=len(lat
tokenizer = AutoTokenizer.from_pretrained(model_dir,local_files_only=True)
model = AutoModelForSequenceClassification.from_pretrained(model_dir,local_files_only=True,config=config)
```

```
Some weights of BertForSequenceClassification were not initialized from the model checkpoint at /content/drive/MyDrive/offli
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
```

```
model
```

```
BertForSequenceClassification(
  (bert): BertModel(
    (embeddings): BertEmbeddings(
      (word_embeddings): Embedding(30522, 768, padding_idx=0)
      (position_embeddings): Embedding(512, 768)
      (token_type_embeddings): Embedding(2, 768)
      (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
      (dropout): Dropout(p=0.1, inplace=False)
    )
    (encoder): BertEncoder(
      (layer): ModuleList(
        (0-11): 12 x BertLayer(
          (attention): BertAttention(
            (self): BertSdpaSelfAttention(
              (query): Linear(in_features=768, out_features=768, bias=True)
              (key): Linear(in_features=768, out_features=768, bias=True)
              (value): Linear(in_features=768, out_features=768, bias=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
            (output): BertSelfOutput(
              (dense): Linear(in_features=768, out_features=768, bias=True)
              (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
          )
          (intermediate): BertIntermediate(
```

```
            (dense): Linear(in_features=768, out_features=3072, bias=True)
            (intermediate_act_fn): GELUActivation()
          )
          (output): BertOutput(
            (dense): Linear(in_features=3072, out_features=768, bias=True)
            (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
            (dropout): Dropout(p=0.1, inplace=False)
          )
        )
      )
    )
    (pooler): BertPooler(
      (dense): Linear(in_features=768, out_features=768, bias=True)
      (activation): Tanh()
    )
  )
  (dropout): Dropout(p=0.1, inplace=False)
  (classifier): Linear(in_features=768, out_features=6, bias=True)
)
```

```
model.config
```

```
BertConfig {
  "architectures": [
    "BertModel"
  ],
  "attention_probs_dropout_prob": 0.1,
  "classifier_dropout": null,
  "dtype": "float32",
  "gradient_checkpointing": false,
  "hidden_act": "gelu",
  "hidden_dropout_prob": 0.1,
  "hidden_size": 768,
  "id2label": {
    "0": "sadness",
    "1": "joy",
    "2": "love",
    "3": "anger",
    "4": "fear",
    "5": "surprise"
  },
  "initializer_range": 0.02,
  "intermediate_size": 3072,
  "label2id": {
    "anger": 3,
    "fear": 4,
    "joy": 1,
    "love": 2,
    "sadness": 0,
    "surprise": 5
  },
  "layer_norm_eps": 1e-12,
  "max_position_embeddings": 512,
  "model_type": "bert",
  "num_attention_heads": 12,
  "num_hidden_layers": 12,
  "pad_token_id": 0,
  "position_embedding_type": "absolute",
  "transformers_version": "4.57.3",
  "type_vocab_size": 2,
  "use_cache": true,
  "vocab_size": 30522
}
```

7. Tokenization

```
def tokenize(batch):
  return tokenizer(batch['text'],truncation=True,padding=True,max_length=120)

print(tokenize(dataset['train'][0]))
print(dataset['train'][0])
```

```
{'input_ids': [101, 1045, 2514, 12511, 2009, 2003, 2053, 2393, 2005, 2033, 2008, 2060, 5381, 2360, 2008, 1045, 2572, 3407, 2
{'text': 'i feel unhappy it is no help for me that other persons say that i am happy how much truth there may be in it', 'la
```

```
tokenized_dataset = dataset.map(tokenize,batched=True,batch_size=None)
tokenized_dataset
```

```
Map: 100%                                                    11200/11200 [00:02<00:00, 4638.87 examples/s]

Map: 100%                                                    3200/3200 [00:01<00:00, 2814.04 examples/s]

Map: 100%                                                    1600/1600 [00:00<00:00, 2090.31 examples/s]
```

```
DatasetDict({
    train: Dataset({
        features: ['text', 'label', 'label_name', 'Words Per Tweet', 'input_ids', 'token_type_ids', 'attention_mask'],
        num_rows: 11200
    })
    test: Dataset({
        features: ['text', 'label', 'label_name', 'Words Per Tweet', 'input_ids', 'token_type_ids', 'attention_mask'],
        num_rows: 3200
    })
    validation: Dataset({
        features: ['text', 'label', 'label_name', 'Words Per Tweet', 'input_ids', 'token_type_ids', 'attention_mask'],
        num_rows: 1600
    })
})
```

Removing Other columns to save space

```
columns_to_remove = ['text', 'Words Per Tweet']

final_dataset = tokenized_dataset.remove_columns(columns_to_remove)
final_dataset
```

```
DatasetDict({
    train: Dataset({
        features: ['label', 'label_name', 'input_ids', 'token_type_ids', 'attention_mask'],
        num_rows: 11200
    })
    test: Dataset({
        features: ['label', 'label_name', 'input_ids', 'token_type_ids', 'attention_mask'],
        num_rows: 3200
    })
    validation: Dataset({
        features: ['label', 'label_name', 'input_ids', 'token_type_ids', 'attention_mask'],
        num_rows: 1600
    })
})
```

8.Training Arguements

```
from transformers import TrainingArguments

batch_size = 64

training_args = TrainingArguments(
    output_dir="./results",
    eval_strategy="epoch",
    per_device_train_batch_size=batch_size,
    per_device_eval_batch_size=batch_size,
    num_train_epochs=3,
    learning_rate=2e-5,
    weight_decay=0.01,
    disable_tqdm=False,
)
```

9. Compute Metrics

```
import evaluate

accuracy = evaluate.load('accuracy')
f1 = evaluate.load('f1')
precision = evaluate.load('precision')

def compute_metrics_evaluate(eval_pred):
  logits,labels = eval_pred
  preds = np.argmax(logits,axis=-1)
  results ={}
  results['accuracy'] = accuracy.compute(predictions=preds,references=labels)
  results['f1'] = f1.compute(predictions=preds,references=labels,average='weighted')
  results['precision'] = precision.compute(predictions=preds,references=labels,average='weighted')
  return results
```

```
/usr/local/lib/python3.12/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
Downloading builder script:       4.20k/? [00:00<00:00, 109kB/s]

Downloading builder script:       6.79k/? [00:00<00:00, 107kB/s]

Downloading builder script:       7.56k/? [00:00<00:00, 143kB/s]
```

10. Training

```
trainer = Trainer(
    model = model,
    args = training_args,
    compute_metrics = compute_metrics_evaluate,
    tokenizer = tokenizer,
    train_dataset = final_dataset['train'],
    eval_dataset = final_dataset['validation']
)
```

```
/tmp/ipython-input-2343835500.py:1: FutureWarning: `tokenizer` is deprecated and will be removed in version 5.0.0 for `Trair
  trainer = Trainer(
```

```
trainer.train()
```

```
/usr/local/lib/python3.12/dist-packages/notebook/notebookapp.py:191: SyntaxWarning: invalid escape sequence '\/'
  | |_| | '_ \/ _` / _` |  _/ -_)
wandb: (1) Create a W&B account
wandb: (2) Use an existing W&B account
wandb: (3) Don't visualize my results
wandb: Enter your choice: 1
wandb: You chose 'Create a W&B account'
wandb: Create an account here: https://wandb.ai/authorize?signup=true&ref=models
wandb: Paste an API key from your profile and hit enter: ··········
wandb: No netrc file found, creating one.
wandb: Appending key for api.wandb.ai to your netrc file: /root/.netrc
wandb: Currently logged in as: pavan220405 (pavan220405-iit-ropar-tif) to https://api.wandb.ai. Use `wandb login --relogin`
Tracking run with wandb version 0.23.1
Run data is saved locally in /content/wandb/run-20251210_092022-hcvwgepq
Syncing run golden-rain-5 to Weights & Biases (docs)
View project at https://wandb.ai/pavan220405-iit-ropar-tif/huggingface
View run at https://wandb.ai/pavan220405-iit-ropar-tif/huggingface/runs/hcvwgepq
```

[525/525 08:25, Epoch 3/3]

Epoch	Training Loss	Validation Loss	Accuracy	F1	Precision
1	No log	0.334006	{'accuracy': 0.901875}	{'f1': 0.8970588701834807}	{'precision': 0.9020262374070493}
2	No log	0.182766	{'accuracy': 0.93625}	{'f1': 0.9367571042440944}	{'precision': 0.9378804654104752}
3	0.484200	0.169274	{'accuracy': 0.9375}	{'f1': 0.9385452127879522}	{'precision': 0.9420090835759831}

```
Trainer is attempting to log a value of "{'accuracy': 0.901875}" of type <class 'dict'> for key "eval/accuracy" as a scalar.
Trainer is attempting to log a value of "{'f1': 0.8970588701834807}" of type <class 'dict'> for key "eval/f1" as a scalar. T
Trainer is attempting to log a value of "{'precision': 0.9020262374070493}" of type <class 'dict'> for key "eval/precision"
Trainer is attempting to log a value of "{'accuracy': 0.93625}" of type <class 'dict'> for key "eval/accuracy" as a scalar.
Trainer is attempting to log a value of "{'f1': 0.9367571042440944}" of type <class 'dict'> for key "eval/f1" as a scalar. T
Trainer is attempting to log a value of "{'precision': 0.9378804654104752}" of type <class 'dict'> for key "eval/precision"
Trainer is attempting to log a value of "{'accuracy': 0.9375}" of type <class 'dict'> for key "eval/accuracy" as a scalar. T
Trainer is attempting to log a value of "{'f1': 0.9385452127879522}" of type <class 'dict'> for key "eval/f1" as a scalar. T
Trainer is attempting to log a value of "{'precision': 0.9420090835759831}" of type <class 'dict'> for key "eval/precision"
TrainOutput(global_step=525, training_loss=0.4682055818466913, metrics={'train_runtime': 532.6382,
'train_samples_per_second': 63.082, 'train_steps_per_second': 0.986, 'total_flos': 1502253632275200.0, 'train_loss':
0.4682055818466913, 'epoch': 3.0})
```

11. Prediction and Evaluation

```
output_preds = trainer.predict(final_dataset['test'])
output_preds.metrics
```

```
{'test_loss': 0.19707100093364716,
 'test_accuracy': {'accuracy': 0.925},
 'test_f1': {'f1': 0.9256846783506574},
 'test_precision': {'precision': 0.9271564049129979},
 'test_runtime': 12.9292,
 'test_samples_per_second': 247.502,
 'test_steps_per_second': 3.867}
```

```
print(final_dataset['test']['label'][:])
```

```
[3, 0, 4, 1, 3, 3, 1, 1, 1, 0, 5, 1, 0, 1, 3, 0, 0, 1, 0, 0, 0, 4, 5, 2, 4, 3, 4, 0, 1, 3, 0, 4, 1, 1, 1, 1, 0, 0, 1, 1, 0,
```

```
y_pred = np.argmax(output_preds.predictions,axis=-1)
y_true = final_dataset['test']['label'][:]
```

```
from sklearn.metrics import classification_report,confusion_matrix

print(classification_report(y_true,y_pred))
```

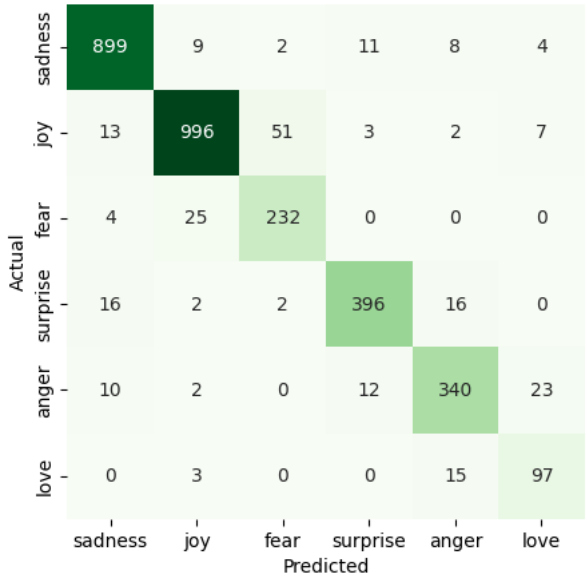
```
              precision    recall  f1-score   support

           0       0.95      0.96      0.96       933
           1       0.96      0.93      0.94      1072
           2       0.81      0.89      0.85       261
           3       0.94      0.92      0.93       432
           4       0.89      0.88      0.89       387
           5       0.74      0.84      0.79       115

    accuracy                           0.93      3200
   macro avg       0.88      0.90      0.89      3200
weighted avg       0.93      0.93      0.93      3200
```

```
cm = confusion_matrix(y_true, y_pred)

plt.figure(figsize=(5,5))
sns.heatmap(cm, annot=True, xticklabels=label2id.keys(), yticklabels=label2id.keys(), fmt='d', cbar=False, cmap='Greens')
plt.ylabel("Actual")
plt.xlabel("Predicted")
plt.show()
```



Save the model

```
trainer.save_model('/content/drive/MyDrive/offline_models/bert-sentiment-model')
```

```
from transformers import pipeline

tokenizer = AutoTokenizer.from_pretrained('/content/drive/MyDrive/offline_models/bert-sentiment-model',local_files_only=Tru
classifier = pipeline('sentiment-analysis',model='/content/drive/MyDrive/offline_models/bert-sentiment-model')
```

```
Device set to use cuda:0
```

```
classifier(['hello, how are you?', "love you", "i am feeling low","I am so happy today! Best day ever.",
            "I love this phone — camera is amazing.","I feel empty and alone.","I'm terrified of the exam tomorrow.",
            "This is outrageous! I will never buy from them again.","Well done 👏👏 I'm proud of you!",
            "Honestly, I love and hate this at the same time.","Can't sleep, heart racing, so anxious.",
            "She is my soulmate, forever <3"])
```

```
[{'label': 'joy', 'score': 0.7938095927238464},
 {'label': 'love', 'score': 0.7806172966957092},
 {'label': 'sadness', 'score': 0.9862167835235596},
 {'label': 'joy', 'score': 0.9843908548355103},
 {'label': 'surprise', 'score': 0.48596784472465515},
 {'label': 'sadness', 'score': 0.987054169178009},
 {'label': 'fear', 'score': 0.9601675868034363},
 {'label': 'anger', 'score': 0.9373120069503784},
 {'label': 'joy', 'score': 0.9833775758743286},
 {'label': 'anger', 'score': 0.4796634614467621},
 {'label': 'fear', 'score': 0.9555342197418213},
 {'label': 'joy', 'score': 0.7872892618179321}]
```

```
Start coding or generate with AI.
```