

```
!git clone https://github.com/ArjunYadav18/dl_project.git
%cd dl_project
```

```
fatal: destination path 'dl_project' already exists and is not an empty directory.
/content/dl_project
```

```
%cd dl_project
```

```
[Errno 2] No such file or directory: 'dl_project'
/content/dl_project
```

```
!mkdir -p notebooks model saved_models evaluation data
```

```
!ls
```

```
data evaluation model notebooks saved_models
```

```
!git remote set-url origin https://ArjunYadav18:ghp_Rarws23buwqoUb0XRPLKDpFzZTmluk2nsAzD@github.com/ArjunYadav18/dl_project.git
```

```
fatal: not a git repository (or any of the parent directories): .git
```

```
!pip install datasets transformers
```

```
Requirement already satisfied: datasets in /usr/local/lib/python3.11/dist-packages (2.14.4)
Requirement already satisfied: transformers in /usr/local/lib/python3.11/dist-packages (4.53.1)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.11/dist-packages (from datasets) (2.0.2)
Requirement already satisfied: pyarrow>=8.0.0 in /usr/local/lib/python3.11/dist-packages (from datasets) (18.1.0)
Requirement already satisfied: dill<0.3.8,>=0.3.0 in /usr/local/lib/python3.11/dist-packages (from datasets) (0.3.7)
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (from datasets) (2.2.2)
Requirement already satisfied: requests>=2.19.0 in /usr/local/lib/python3.11/dist-packages (from datasets) (2.32.3)
Requirement already satisfied: tqdm>=4.62.1 in /usr/local/lib/python3.11/dist-packages (from datasets) (4.67.1)
Requirement already satisfied: xxhash in /usr/local/lib/python3.11/dist-packages (from datasets) (3.5.0)
Requirement already satisfied: multiprocessing in /usr/local/lib/python3.11/dist-packages (from datasets) (0.70.15)
Requirement already satisfied: fsspec>=2021.11.1 in /usr/local/lib/python3.11/dist-packages (from datasets) (2021.11.1)
Requirement already satisfied: aiohttp in /usr/local/lib/python3.11/dist-packages (from datasets) (3.11.15)
Requirement already satisfied: huggingface-hub<1.0.0,>=0.14.0 in /usr/local/lib/python3.11/dist-packages (from datasets) (0.24.2)
Requirement already satisfied: packaging in /usr/local/lib/python3.11/dist-packages (from datasets) (24.2)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.11/dist-packages (from datasets) (6.0.2)
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (from transformers) (3.18.0)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.11/dist-packages (from transformers) (2024.11.6)
Requirement already satisfied: tokenizers<0.22,>=0.21 in /usr/local/lib/python3.11/dist-packages (from transformers) (0.21.1)
Requirement already satisfied: safetensors>=0.4.3 in /usr/local/lib/python3.11/dist-packages (from transformers) (0.5.3)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (2.4.4)
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (1.4.0)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (25.3.0)
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (1.7.0)
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (6.6.0)
Requirement already satisfied: propcache>=0.2.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (0.3.2)
Requirement already satisfied: yarl<2.0,>=1.17.0 in /usr/local/lib/python3.11/dist-packages (from aiohttp->datasets) (1.20.1)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub<1.0.0,>=0.14.0) (4.12.2)
Requirement already satisfied: hf-xet<2.0.0,>=1.1.2 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub<1.0.0,>=0.14.0) (1.1.2)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests>=2.19.0) (3.4.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests>=2.19.0) (3.10.1)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests>=2.19.0) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests>=2.19.0) (2025.11.11)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas->datasets) (2.9.0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas->datasets) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas->datasets) (2025.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas->datasets) (1.17.0)
```

```
import pandas as pd
```

```
df1 = pd.read_csv("goemotions_1.csv", header=None)
df2 = pd.read_csv("goemotions_2.csv", header=None)
df3 = pd.read_csv("goemotions_3.csv", header=None)
```

```
# Combine them
df = pd.concat([df1, df2, df3], ignore_index=True)
print(df.shape)
df.head()
```

```

/tmp/ipython-input-5-1038994141.py:3: DtypeWarning: Columns (6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28)
df1 = pd.read_csv("goemotions_1.csv", header=None)
/tmp/ipython-input-5-1038994141.py:4: DtypeWarning: Columns (6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28)
df2 = pd.read_csv("goemotions_2.csv", header=None)
(211228, 37)
/tmp/ipython-input-5-1038994141.py:5: DtypeWarning: Columns (6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28)
df3 = pd.read_csv("goemotions_3.csv", header=None)

```

	0	1	2	3	4	5	6	7	8	
0	text	id	author	subreddit	link_id	parent_id	created_utc	rater_id	example_very_unclear	admirati
1	That game hurt.	eew5j0j	Brdd9	nrl	t3_ajis4z	t1_eew18eq	1548381039.0	1	False	
2	>sexuality shouldn't be a grouping category I...	eemcysk	TheGreen888	unpopularopinion	t3_ai4q37	t3_ai4q37	1548084169.0	37	True	
3	You do right, if you don't care then fuck 'em!	ed2mah1	Labalool	confessions	t3_abru74	t1_ed2m7g7	1546427744.0	37	False	
4	Man I love reddit.	eeibobj	MrsRobertshaw	facepalm	t3_ahulml	t3_ahulml	1547965054.0	18	False	

5 rows × 37 columns

```

# Define emotion labels (GoEmotions official 28 emotions)
emotion_labels = [
    'admiration', 'amusement', 'anger', 'annoyance', 'approval', 'caring',
    'confusion', 'curiosity', 'desire', 'disappointment', 'disapproval',
    'disgust', 'embarrassment', 'excitement', 'fear', 'gratitude', 'grief',
    'joy', 'love', 'nervousness', 'optimism', 'pride', 'realization', 'relief',
    'remorse', 'sadness', 'surprise', 'neutral'
]

```

```

# Extract only 'text' and label columns from the raw dataframe
text_column = 1 # text column index
label_columns = list(range(9, 37)) # label columns (from screenshot)

```

```

df_cleaned = df.iloc[:, [text_column] + label_columns]
df_cleaned.columns = ['text'] + emotion_labels

```

```

# Drop any rows with missing text or labels
df_cleaned = df_cleaned.dropna()

```

```

# Drop the first row as it contains column names
df_cleaned = df_cleaned.iloc[1:]

```

```

print("Cleaned shape:", df_cleaned.shape)
df_cleaned.head()

```

Cleaned shape: (211227, 29)

	text	admiration	amusement	anger	annoyance	approval	caring	confusion	curiosity	desire	...	love	nervousness
1	eew5j0j	0	0	0	0	0	0	0	0	0	...	0	0
2	eemcysk	0	0	0	0	0	0	0	0	0	...	0	0
3	ed2mah1	0	0	0	0	0	0	0	0	0	...	0	0
4	eeibobj	0	0	0	0	0	0	0	0	0	...	1	0
5	eda6yn6	0	0	0	0	0	0	0	0	0	...	0	0

5 rows × 29 columns

```
import torch
```


```

# Remove any accidental rows where label columns have string values
def is_label_header_row(row):
    return any([label in str(v).lower() for label in emotion_labels for v in row])

```


```
df_cleaned = df_cleaned[~df_cleaned[emotion_labels].apply(is_label_header_row, axis=1)]
```

```
df_cleaned['label_vector'] = df_cleaned[emotion_labels].apply(encode_labels, axis=1)
```

 /tmp/ipython-input-11-1568242959.py:9: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view
df_cleaned['label_vector'] = df_cleaned[emotion_labels].apply(encode_labels, axis=1)

```
df_cleaned['label_vector'] = df_cleaned[emotion_labels].apply(encode_labels, axis=1)
```

 /tmp/ipython-input-12-3587747778.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view
df_cleaned['label_vector'] = df_cleaned[emotion_labels].apply(encode_labels, axis=1)

```
from sklearn.model_selection import train_test_split
```

```
train_texts, val_texts, train_labels, val_labels = train_test_split(
    df_cleaned['text'], df_cleaned['label_vector'], test_size=0.1, random_state=42
)
```

```
!pip install transformers
```

```
from transformers import AutoTokenizer
```

```
tokenizer = AutoTokenizer.from_pretrained("distilbert-base-uncased")
```

```
def tokenize_texts(texts):
    return tokenizer(
        list(texts),
        padding='max_length',
        truncation=True,
        max_length=128,
        return_tensors='pt'
    )
```

```
train_encodings = tokenize_texts(train_texts)
```

```
val_encodings = tokenize_texts(val_texts)
```

 Requirement already satisfied: transformers in /usr/local/lib/python3.11/dist-packages (4.53.1)
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (from transformers) (3.18.0)
Requirement already satisfied: huggingface-hub<1.0,>=0.30.0 in /usr/local/lib/python3.11/dist-packages (from transformers) (0.21.1)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.11/dist-packages (from transformers) (2.0.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (from transformers) (24.2)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.11/dist-packages (from transformers) (6.0.2)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.11/dist-packages (from transformers) (2024.11.6)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from transformers) (2.32.3)
Requirement already satisfied: tokenizers<0.22,>=0.21 in /usr/local/lib/python3.11/dist-packages (from transformers) (0.21.1)
Requirement already satisfied: safetensors>=0.4.3 in /usr/local/lib/python3.11/dist-packages (from transformers) (0.5.3)
Requirement already satisfied: tqdm>=4.27 in /usr/local/lib/python3.11/dist-packages (from transformers) (4.67.1)
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub<1.0,>=0.30) (2024.10.1)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub<1.0,>=0.30) (4.12.2)
Requirement already satisfied: hf-xet<2.0.0,>=1.1.2 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub<1.0,>=0.30) (1.1.2)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests->transformers) (3.4.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests->transformers) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests->transformers) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests->transformers) (2025.1.1)

tokenizer_config.json: 100% 48.0/48.0 [00:00<00:00, 2.16kB/s]

config.json: 100% 483/483 [00:00<00:00, 52.5kB/s]

vocab.txt: 100% 232k/232k [00:00<00:00, 2.59MB/s]

tokenizer.json: 100% 466k/466k [00:00<00:00, 2.83MB/s]

```
from torch.utils.data import Dataset
```

```

class GoEmotionsDataset(Dataset):
    def __init__(self, encodings, labels):
        self.encodings = encodings
        self.labels = labels

    def __getitem__(self, idx):
        return {
            'input_ids': self.encodings['input_ids'][idx],
            'attention_mask': self.encodings['attention_mask'][idx],
            'labels': self.labels[idx]
        }

    def __len__(self):
        return len(self.labels)

# Wrap datasets
train_dataset = GoEmotionsDataset(train_encodings, list(train_labels))
val_dataset = GoEmotionsDataset(val_encodings, list(val_labels))

import torch
from transformers import DistilBertForSequenceClassification

NUM_LABELS = 28

model = DistilBertForSequenceClassification.from_pretrained(
    "distilbert-base-uncased",
    num_labels=NUM_LABELS,
    problem_type="multi_label_classification"
)

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model.to(device)

```



model.safetensors: 100%

268M/268M [00:08<00:00, 31.1MB/s]

Some weights of DistilBertForSequenceClassification were not initialized from the model checkpoint at distilbert-base-uncased. You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.

```

DistilBertForSequenceClassification(
  (distilbert): DistilBertModel(
    (embeddings): Embeddings(
      (word_embeddings): Embedding(30522, 768, padding_idx=0)
      (position_embeddings): Embedding(512, 768)
      (LayerNorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
      (dropout): Dropout(p=0.1, inplace=False)
    )
    (transformer): Transformer(
      (layer): ModuleList(
        (0-5): 6 x TransformerBlock(
          (attention): DistilBertSdpaAttention(
            (dropout): Dropout(p=0.1, inplace=False)
            (q_lin): Linear(in_features=768, out_features=768, bias=True)
            (k_lin): Linear(in_features=768, out_features=768, bias=True)
            (v_lin): Linear(in_features=768, out_features=768, bias=True)
            (out_lin): Linear(in_features=768, out_features=768, bias=True)
          )
          (sa_layer_norm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
          (ffn): FFN(
            (dropout): Dropout(p=0.1, inplace=False)
            (lin1): Linear(in_features=768, out_features=3072, bias=True)
            (lin2): Linear(in_features=3072, out_features=768, bias=True)
            (activation): GELUActivation()
          )
          (output_layer_norm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
        )
      )
    )
  )
  (pre_classifier): Linear(in_features=768, out_features=768, bias=True)
  (classifier): Linear(in_features=768, out_features=28, bias=True)
  (dropout): Dropout(p=0.2, inplace=False)
)

```

```

from tqdm import tqdm # make sure this is imported
from torch.utils.data import DataLoader

```

```

# Define loss function and optimizer
loss_fn = torch.nn.BCEWithLogitsLoss() # Appropriate for multi-label classification

```

```
optimizer = torch.optim.AdamW(model.parameters(), lr=5e-5)

# Create DataLoaders
train_loader = DataLoader(train_dataset, batch_size=16, shuffle=True)
val_loader = DataLoader(val_dataset, batch_size=16)

epochs = 1

for epoch in range(epochs):
    model.train()
    total_loss = 0

    for batch in tqdm(train_loader):
        input_ids = batch['input_ids'].to(device)
        attention_mask = batch['attention_mask'].to(device)
        labels = batch['labels'].to(device) # Removed torch.stack

        outputs = model(input_ids, attention_mask=attention_mask)
        logits = outputs.logits

        loss = loss_fn(logits, labels)
        loss.backward()
        optimizer.step()
        optimizer.zero_grad()

        total_loss += loss.item()

    avg_loss = total_loss / len(train_loader)
    print(f"Epoch {epoch+1} | Training Loss: {avg_loss:.4f}")

➡ 100%|██████████| 11882/11882 [36:16<00:00, 5.46it/s]Epoch 1 | Training Loss: 0.1578
```

```
from sklearn.metrics import classification_report, accuracy_score, f1_score, precision_score, recall_score
import numpy as np
```

```
model.eval()
all_preds = []
all_labels = []

with torch.no_grad():
    for batch in val_loader:
        input_ids = batch['input_ids'].to(device)
        attention_mask = batch['attention_mask'].to(device)
        labels = batch['labels'].cpu().numpy() # Removed torch.stack

        outputs = model(input_ids, attention_mask=attention_mask)
        logits = outputs.logits
        preds = torch.sigmoid(logits).cpu().numpy()
        preds = (preds > 0.5).astype(int)

        all_preds.extend(preds)
        all_labels.extend(labels)
```

```
all_preds = np.array(all_preds)
all_labels = np.array(all_labels)
```

```
# ---- METRICS ----
print("\n 🌈 Multi-label Evaluation Metrics")
```

```
# Subset Accuracy (strict match of all labels)
subset_accuracy = accuracy_score(all_labels, all_preds)
print(f"Subset Accuracy: {subset_accuracy:.4f}")
```

```
# Micro-averaged
micro_precision = precision_score(all_labels, all_preds, average='micro', zero_division=0)
micro_recall = recall_score(all_labels, all_preds, average='micro', zero_division=0)
micro_f1 = f1_score(all_labels, all_preds, average='micro', zero_division=0)
print(f"Micro Precision: {micro_precision:.4f}")
print(f"Micro Recall: {micro_recall:.4f}")
print(f"Micro F1-score: {micro_f1:.4f}")
```

```
# Macro-averaged
macro_precision = precision_score(all_labels, all_preds, average='macro', zero_division=0)
macro_recall = recall_score(all_labels, all_preds, average='macro', zero_division=0)
macro_f1 = f1_score(all_labels, all_preds, average='macro', zero_division=0)
```

```

print(f"Macro Precision: {macro_precision:.4f}")
print(f"Macro Recall: {macro_recall:.4f}")
print(f"Macro F1-score: {macro_f1:.4f}")

# Classification Report (per class)
print("\nDetailed Classification Report:")
print(classification_report(all_labels, all_preds, target_names=emotion_labels, zero_division=0))

```



Multi-label Evaluation Metrics

```

Subset Accuracy: 0.0153
Micro Precision: 0.0000
Micro Recall: 0.0000
Micro F1-score: 0.0000
Macro Precision: 0.0000
Macro Recall: 0.0000
Macro F1-score: 0.0000

```

Detailed Classification Report:

	precision	recall	f1-score	support
admiration	0.00	0.00	0.00	1728
amusement	0.00	0.00	0.00	942
anger	0.00	0.00	0.00	838
annoyance	0.00	0.00	0.00	1384
approval	0.00	0.00	0.00	1709
caring	0.00	0.00	0.00	566
confusion	0.00	0.00	0.00	734
curiosity	0.00	0.00	0.00	999
desire	0.00	0.00	0.00	400
disappointment	0.00	0.00	0.00	842
disapproval	0.00	0.00	0.00	1156
disgust	0.00	0.00	0.00	539
embarrassment	0.00	0.00	0.00	262
excitement	0.00	0.00	0.00	550
fear	0.00	0.00	0.00	310
gratitude	0.00	0.00	0.00	1156
grief	0.00	0.00	0.00	50
joy	0.00	0.00	0.00	789
love	0.00	0.00	0.00	815
nervousness	0.00	0.00	0.00	201
optimism	0.00	0.00	0.00	873
pride	0.00	0.00	0.00	133
realization	0.00	0.00	0.00	886
relief	0.00	0.00	0.00	131
remorse	0.00	0.00	0.00	242
sadness	0.00	0.00	0.00	619
surprise	0.00	0.00	0.00	539
neutral	0.00	0.00	0.00	5581
micro avg	0.00	0.00	0.00	24974
macro avg	0.00	0.00	0.00	24974
weighted avg	0.00	0.00	0.00	24974
samples avg	0.00	0.00	0.00	24974

```

model.save_pretrained("goemotions-distilbert")
tokenizer.save_pretrained("goemotions-distilbert")

```



```

('goemotions-distilbert/tokenizer_config.json',
'goemotions-distilbert/special_tokens_map.json',
'goemotions-distilbert/vocab.txt',
'goemotions-distilbert/added_tokens.json',
'goemotions-distilbert/tokenizer.json')

```

```
import shutil
```

```
# Save model + tokenizer
```

```
model_path = "/content/dl_project/model"
```

```
tokenizer_path = "/content/dl_project/model/tokenizer"
```

```
model.save_pretrained(model_path)
```

```
tokenizer.save_pretrained(tokenizer_path)
```

```
# Save notebook
```

```
shutil.copy("/content/train.ipynb", "/content/dl_project/train.ipynb")
```



```

('/content/dl_project/model/tokenizer/tokenizer_config.json',
'/content/dl_project/model/tokenizer/special_tokens_map.json',

```

```
    '/content/dl_project/model/tokenizer/vocab.txt',  
    '/content/dl_project/model/tokenizer/added_tokens.json',  
    '/content/dl_project/model/tokenizer/tokenizer.json')  
  
with open("requirements.txt", "w") as f:  
    f.write("transformers\nscikit-learn\ntorch\npandas\n")  
  
from google.colab import files  
files.download("train.ipynb")  
  
!zip -r model.zip goemotions-distilbert  
from google.colab import files  
files.download("model.zip")
```

