```
pip install datasets
```

```
Collecting datasets
  Downloading datasets-3.0.2-py3-none-any.whl.metadata (20 kB)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from datasets) (3.16.1)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-packages (from datasets) (1.26.4)
Requirement already satisfied: pyarrow>=15.0.0 in /usr/local/lib/python3.10/dist-packages (from datasets) (16.1.0)
Collecting dill<0.3.9,>=0.3.0 (from datasets)
  Downloading dill-0.3.8-py3-none-any.whl.metadata (10 kB)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (from datasets) (2.2.2)
Requirement already satisfied: requests>=2.32.2 in /usr/local/lib/python3.10/dist-packages (from datasets) (2.32.3)
Requirement already satisfied: tqdm>=4.66.3 in /usr/local/lib/python3.10/dist-packages (from datasets) (4.66.5)
Collecting xxhash (from datasets)
  Downloading xxhash-3.5.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (12 kB)
Collecting multiprocess<0.70.17 (from datasets)
  Downloading multiprocess-0.70.16-py310-none-any.whl.metadata (7.2 kB)
Requirement already satisfied: fsspec<=2024.9.0,>=2023.1.0 in /usr/local/lib/python3.10/dist-packages (from fsspec[http]<=
Requirement already satisfied: aiohttp in /usr/local/lib/python3.10/dist-packages (from datasets) (3.10.10)
Requirement already satisfied: huggingface-hub>=0.23.0 in /usr/local/lib/python3.10/dist-packages (from datasets) (0.24.7
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from datasets) (24.1)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.10/dist-packages (from datasets) (6.0.2)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.3.
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (24.2.0)
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.4
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (6
Requirement already satisfied: yarl<2.0,>=1.12.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.1
Requirement already satisfied: async-timeout<5.0,>=4.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface-hu
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests>=2.32.2-
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->datasets)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->datas
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->datas
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets) (2024.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas->
Requirement already satisfied: propcache>=0.2.0 in /usr/local/lib/python3.10/dist-packages (from yarl<2.0,>=1.12.0->aioht
Downloading datasets-3.0.2-py3-none-any.whl (472 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 472.7/472.7 kB 14.4 MB/s eta 0:00:00
Downloading dill-0.3.8-py3-none-any.whl (116 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 116.3/116.3 kB 10.2 MB/s eta 0:00:00
Downloading multiprocess-0.70.16-py310-none-any.whl (134 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 134.8/134.8 kB 10.9 MB/s eta 0:00:00
Downloading xxhash-3.5.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (194 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 194.1/194.1 kB 15.0 MB/s eta 0:00:00
Installing collected packages: xxhash, dill, multiprocess, datasets
Successfully installed datasets-3.0.2 dill-0.3.8 multiprocess-0.70.16 xxhash-3.5.0
```

```python
import pandas as pd
import numpy as np
import torch
from transformers import BertTokenizer, BertForSequenceClassification
from transformers import Trainer, TrainingArguments
from datasets import Dataset
```

```python
df = pd.read_csv('/content/normal_tissue.tsv', sep='\t')
df = df.head(2000)
```

```python
df.head()
```

	Gene	Gene name	Tissue	Cell type	Level	Reliability
0	ENSG00000000003	TSPAN6	adipose tissue	adipocytes	Not detected	Approved
1	ENSG00000000003	TSPAN6	adrenal gland	glandular cells	Not detected	Approved
2	ENSG00000000003	TSPAN6	appendix	glandular cells	Medium	Approved
3	ENSG00000000003	TSPAN6	appendix	lymphoid tissue	Not detected	Approved
4	ENSG00000000003	TSPAN6	bone marrow	hematopoietic cells	Not detected	Approved

Next steps: Generate code with df View recommended plots New interactive sheet

```
df.isnull().sum()
```

```
                  0
       Gene       0
  Gene name       0
     Tissue       1
  Cell type       1
      Level       1
Reliability       0

dtype: int64
```

```
for column in df.columns:
    # Calculate the mode for the column
    mode_value = df[column].mode()
    # Fill NaN values with the mode (taking the first mode if multiple)
    if not mode_value.empty:
        df[column].fillna(mode_value[0], inplace=True)
```

```
<ipython-input-8-47e37fa7aa73>:6: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through cl
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we a

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[co


  df[column].fillna(mode_value[0], inplace=True)
```

```
df.isnull().sum()
```

```
                  0
       Gene       0
  Gene name       0
     Tissue       0
  Cell type       0
      Level       0
Reliability       0

dtype: int64
```

```
df['Reliability'].replace({'Approved': 0, 'Enhanced': 1, 'Uncertain': 2, 'Supported': 3}, inplace=True)
```

```
<ipython-input-10-be5be02d7c9e>:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through 
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we a

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[co


  df['Reliability'].replace({'Approved': 0, 'Enhanced': 1, 'Uncertain': 2, 'Supported': 3}, inplace=True)
<ipython-input-10-be5be02d7c9e>:1: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in 
  df['Reliability'].replace({'Approved': 0, 'Enhanced': 1, 'Uncertain': 2, 'Supported': 3}, inplace=True)
```

```
X = df.drop('Reliability', axis=1)
y = df['Reliability']
```

```
from sklearn.model_selection import train_test_split
# Split the data into training and testing sets
X_train, X_test, Y_train, Y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
print(X_train.shape)
print(X_test.shape)
print(Y_train.shape)
print(Y_test.shape)
```

```
(1600, 5)
(400, 5)
(1600,)
(400,)
```

```
tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')
model = BertForSequenceClassification.from_pretrained('bert-base-uncased', num_labels=4)
```

```
tokenizer_config.json: 100%                                    48.0/48.0 [00:00<00:00, 2.93kB/s]

vocab.txt: 100%                                    232k/232k [00:00<00:00, 1.71MB/s]

tokenizer.json: 100%                                    466k/466k [00:00<00:00, 7.10MB/s]

config.json: 100%                                    570/570 [00:00<00:00, 40.7kB/s]
/usr/local/lib/python3.10/dist-packages/transformers/tokenization_utils_base.py:1601: FutureWarning: `clean_up_tokenizati
  warnings.warn(
model.safetensors: 100%                                    440M/440M [00:01<00:00, 281MB/s]
Some weights of BertForSequenceClassification were not initialized from the model checkpoint at bert-base-uncased and are
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
```

```
X_train['text'] = X_train['Tissue'] + ' ' + X_train['Cell type'] + ' ' + X_train['Gene'] + ' ' + X_train['Level']
X_test['text'] = X_test['Tissue'] + ' ' + X_test['Cell type'] + ' ' + X_test['Gene'] + ' ' + X_test['Level']
```

```
print(X_train['text'].head())
print('\n')
print(X_test['text'].head())
```

```
968    cerebral cortex neuropil ENSG00000001561 Not d...
240     adipose tissue adipocytes ENSG00000000460 Medium
819            skin 1 keratinocytes ENSG00000001461 High
692    cerebral cortex endothelial cells ENSG00000001...
420    cerebral cortex neuronal cells ENSG00000000971...
Name: text, dtype: object


1860       hippocampus glial cells ENSG00000002587 Medium
353     kidney cells in glomeruli ENSG00000000938 Not ...
1333    nasopharynx respiratory epithelial cells ENSG0...
905     lymph node germinal center cells ENSG000000014...
1289    vagina squamous epithelial cells ENSG000000016...
Name: text, dtype: object
```

```
train_encodings = tokenizer(X_train['text'].tolist(), truncation=True, padding=True, max_length=128)
test_encodings = tokenizer(X_test['text'].tolist(), truncation=True, padding=True, max_length=128)
```

```
training_args = TrainingArguments(
    output_dir='./results',
    evaluation_strategy="epoch",
    learning_rate=2e-5,
    per_device_train_batch_size=16,
    per_device_eval_batch_size=16,
    num_train_epochs=3,
    weight_decay=0.01,
)
```

```
/usr/local/lib/python3.10/dist-packages/transformers/training_args.py:1525: FutureWarning: `evaluation_strategy` is depre
  warnings.warn(
```

```
Y_train.head()
```

	Reliability
968	0
240	2
819	0
692	1
420	3

dtype: int64

```
Y_test.head()
```

	Reliability
1860	0
353	1
1333	0
905	3
1289	2

dtype: int64

```
train_dataset = Dataset.from_dict({'input_ids': train_encodings['input_ids'],
                                    'attention_mask': train_encodings['attention_mask'],
                                    'labels': Y_train})

test_dataset = Dataset.from_dict({'input_ids': test_encodings['input_ids'],
                                   'attention_mask': test_encodings['attention_mask'],
                                   'labels': Y_test})


trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=train_dataset,
    eval_dataset=test_dataset,
)



trainer.train()

# Evaluate the model
trainer.evaluate()
```

```
wandb: WARNING The `run_name` is currently set to the same value as `TrainingArguments.output_dir`. If this was not inten
wandb: Using wandb-core as the SDK backend. Please refer to https://wandb.me/wandb-core for more information.
wandb: Appending key for api.wandb.ai to your netrc file: /root/.netrc
```

Tracking run with wandb version 0.18.5
Run data is saved locally in `/content/wandb/run-20241022_185049-7ptojmnh`
Syncing run **./results** to Weights & Biases (docs)
View project at https://wandb.ai/jani-miya/huggingface
View run at https://wandb.ai/jani-miya/huggingface/runs/7ptojmnh

[300/300 00:41, Epoch 3/3]

Epoch	Training Loss	Validation Loss
1	No log	0.882034
2	No log	0.050350
3	No log	0.016277

[25/25 00:00]

```
{'eval_loss': 0.01627664640545845,
 'eval_runtime': 0.6632,
 'eval_samples_per_second': 603.154,
 'eval_steps_per_second': 37.697,
 'epoch': 3.0}
```

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

Now Freezing the layers in the model

```
for param in model.bert.embeddings.parameters():
    param.requires_grad = False
for i in range(6):  # Freezing the first 6 layers (you can change this number)
    for param in model.bert.encoder.layer[i].parameters():
        param.requires_grad = False
```

```
training_args = TrainingArguments(
    output_dir='./results',          # output directory
    evaluation_strategy="epoch",     # evaluate each epoch
    learning_rate=2e-5,
    per_device_train_batch_size=16,
    per_device_eval_batch_size=16,
    num_train_epochs=3,
    weight_decay=0.01,
    logging_dir='./logs',            # directory for logs
    logging_steps=10,
    save_strategy="epoch",
)
```

```
/usr/local/lib/python3.10/dist-packages/transformers/training_args.py:1525: FutureWarning: `evaluation_strategy` is depre
  warnings.warn(
```

```
trainer2 = Trainer(
    model=model,
    args=training_args,
    train_dataset=train_dataset,
    eval_dataset=test_dataset
)
```

```
trainer2.train()

# Evaluate the model
results = trainer2.evaluate()
print("Evaluation results:", results)
```

[300/300 00:50, Epoch 3/3]

Epoch	Training Loss	Validation Loss
1	0.047600	0.001570
2	0.007800	0.000856
3	0.001000	0.000707

[25/25 00:00]

```
Evaluation results: {'eval_loss': 0.0007073960732668638, 'eval_runtime': 0.7961, 'eval_samples_per_second': 502.47, 'eval_
```

```
pip install peft
```

```
Collecting peft
  Downloading peft-0.13.2-py3-none-any.whl.metadata (13 kB)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.10/dist-packages (from peft) (1.26.4)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from peft) (24.1)
Requirement already satisfied: psutil in /usr/local/lib/python3.10/dist-packages (from peft) (5.9.5)
Requirement already satisfied: pyyaml in /usr/local/lib/python3.10/dist-packages (from peft) (6.0.2)
Requirement already satisfied: torch>=1.13.0 in /usr/local/lib/python3.10/dist-packages (from peft) (2.4.1+cu121)
Requirement already satisfied: transformers in /usr/local/lib/python3.10/dist-packages (from peft) (4.44.2)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from peft) (4.66.5)
Requirement already satisfied: accelerate>=0.21.0 in /usr/local/lib/python3.10/dist-packages (from peft) (0.34.2)
Requirement already satisfied: safetensors in /usr/local/lib/python3.10/dist-packages (from peft) (0.4.5)
Requirement already satisfied: huggingface-hub>=0.17.0 in /usr/local/lib/python3.10/dist-packages (from peft) (0.24.7)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.17.0->peft) (
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.17.0-
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.17.0->peft) (
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface-hu
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from torch>=1.13.0->peft) (1.13.3)
Requirement already satisfied: networkx in /usr/local/lib/python3.10/dist-packages (from torch>=1.13.0->peft) (3.4.1)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.10/dist-packages (from torch>=1.13.0->peft) (3.1.4)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.10/dist-packages (from transformers->peft) (20
Requirement already satisfied: tokenizers<0.20,>=0.19 in /usr/local/lib/python3.10/dist-packages (from transformers->peft
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.10/dist-packages (from jinja2->torch>=1.13.0->pe
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->huggin
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface-hub>=0
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface-
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests->huggingface-
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from sympy->torch>=1.13.0->
Downloading peft-0.13.2-py3-none-any.whl (320 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 320.7/320.7 kB 7.4 MB/s eta 0:00:00
Installing collected packages: peft
Successfully installed peft-0.13.2
```

```
from peft import LoraConfig, get_peft_model
```

Using LoRa Technique

```
lora_config = LoraConfig(
    r=8,
    lora_alpha=16,
    target_modules=['query', 'key', 'value'],
    lora_dropout=0.1,
    bias="none"
)
```

```
model3 = get_peft_model(model, lora_config)
```

```
from transformers import AdamW
```

```
optimizer = AdamW(model.parameters(), lr=2e-5)
```

```
/usr/local/lib/python3.10/dist-packages/transformers/optimization.py:591: FutureWarning: This implementation of AdamW is 
  warnings.warn(
```

```
trainer3 = Trainer(
    model=model,
    args=training_args,
    train_dataset=train_dataset,
    eval_dataset=test_dataset,
    optimizers=(optimizer, None),        # Custom optimizer
)
```

```
trainer3.train()

# Evaluate the model
results = trainer3.evaluate()
print("Evaluation results:", results)
```

[300/300 00:57, Epoch 3/3]

Epoch	Training Loss	Validation Loss
1	0.041900	0.000705
2	0.001100	0.000707
3	0.001000	0.000707

[25/25 00:00]

```
Evaluation results: {'eval_loss': 0.0007074868772178888, 'eval_runtime': 0.837, 'eval_samples_per_second': 477.902, 'eval_
```

```
predictions = trainer.predict(test_dataset)
y_predicted = np.argmax(predictions.predictions, axis=1)
```

```
print("Predicted classes:", y_predicted)
```

```
Predicted classes: [0 1 0 3 2 2 3 3 0 0 0 0 1 0 1 2 1 0 1 0 0 1 0 0 0 0 1 0 0 0 0 3 0 3 0 3 1
 0 0 2 0 0 2 0 1 0 3 3 0 1 0 2 3 0 3 3 2 3 1 1 3 0 1 0 0 0 0 0 0 0 3 1 3 0
 1 1 1 0 1 3 0 3 0 0 3 3 3 3 2 1 0 3 0 2 0 0 0 2 0 0 3 0 3 0 0 3 0 2 0 0 0
 3 0 3 1 1 0 0 1 1 3 0 0 3 1 0 3 3 1 1 0 0 1 0 0 2 0 0 3 0 0 2 1 3 1 0 0 2
 0 0 3 3 0 1 2 0 0 3 0 1 0 0 0 1 0 3 3 2 0 3 1 1 0 0 3 0 0 0 3 1 2 0 3 1 0
 1 2 1 1 2 0 2 3 0 0 3 0 3 1 0 0 0 0 1 3 0 3 0 0 0 3 0 2 0 1 3 1 1 3 2 3 0
 0 1 3 0 3 0 1 1 0 0 3 0 3 1 0 3 0 0 1 0 0 3 3 0 0 0 0 0 0 2 0 3 3 0 0 0 0
 2 0 0 0 0 0 1 1 0 0 0 0 0 3 1 0 0 0 3 0 0 0 3 0 3 0 0 2 3 0 0 2 1 3 1 0 2
 0 3 0 3 0 0 0 3 0 0 1 0 3 0 0 1 2 0 0 3 0 1 3 1 0 3 0 2 0 0 0 0 1 1 3 0 0
 0 1 3 1 0 1 3 3 2 0 0 0 1 1 0 0 0 0 3 3 1 0 0 3 3 3 1 3 1 0 0 0 3 3 1 1 2
 0 3 0 3 1 1 3 3 0 3 3 3 3 3 3 0 0 3 0 0 3 1 3 1 0 0 2 3 2 0]
```

```
predictions2 = trainer2.predict(test_dataset)
y_predicted2 = np.argmax(predictions2.predictions, axis=1)
```

```
print("Predicted classes:", y_predicted2)
```

```
Predicted classes: [0 1 0 3 2 2 3 3 0 0 0 0 1 0 1 2 1 0 1 0 0 1 0 0 0 0 1 0 0 0 0 3 0 3 0 3 1
 0 0 2 0 0 2 0 1 0 3 3 0 1 0 2 3 0 3 3 2 3 1 1 3 0 1 0 0 0 0 0 0 0 3 1 3 0
 1 1 1 0 1 3 0 3 0 0 3 3 3 3 2 1 0 3 0 2 0 0 0 2 0 0 3 0 3 0 0 3 0 2 0 0 0
 3 0 3 1 1 0 0 1 1 3 0 0 3 1 0 3 3 1 1 0 0 1 0 0 2 0 0 3 0 0 2 1 3 1 0 0 2
 0 0 3 3 0 1 2 0 0 3 0 1 0 0 0 1 0 3 3 2 0 3 1 1 0 0 3 0 0 0 3 1 2 0 3 1 0
 1 2 1 1 2 0 2 3 0 0 3 0 3 1 0 0 0 0 1 3 0 3 0 0 0 3 0 2 0 1 3 1 1 3 2 3 0
 0 1 3 0 3 0 1 1 0 0 3 0 3 1 0 3 0 0 1 0 0 3 3 0 0 0 0 0 0 2 0 3 3 0 0 0 0
 2 0 0 0 0 0 1 1 0 0 0 0 0 3 1 0 0 0 3 0 0 0 3 0 3 0 0 2 3 0 0 2 1 3 1 0 2
 0 3 0 3 0 0 0 3 0 0 1 0 3 0 0 1 2 0 0 3 0 1 3 1 0 3 0 2 0 0 0 0 1 1 3 0 0
 0 1 3 1 0 1 3 3 2 0 0 0 1 1 0 0 0 0 3 3 1 0 0 3 3 3 1 3 1 0 0 0 3 3 1 1 2
 0 3 0 3 1 1 3 3 0 3 3 3 3 3 3 0 0 3 0 0 3 1 3 1 0 0 2 3 2 0]
```

```
predictions3 = trainer3.predict(test_dataset)
y_predicted3 = np.argmax(predictions3.predictions, axis=1)
```

```
print("Predicted classes:", y_predicted3)
```

```
Predicted classes: [0 1 0 3 2 2 3 3 0 0 0 0 1 0 1 2 1 0 1 0 0 1 0 0 0 0 1 0 0 0 0 3 0 3 0 3 1
 0 0 2 0 0 2 0 1 0 3 3 0 1 0 2 3 0 3 3 2 3 1 1 3 0 1 0 0 0 0 0 0 0 3 1 3 0
 1 1 1 0 1 3 0 3 0 0 3 3 3 3 2 1 0 3 0 2 0 0 0 2 0 0 3 0 3 0 0 3 0 2 0 0 0
 3 0 3 1 1 0 0 1 1 3 0 0 3 1 0 3 3 1 1 0 0 1 0 0 2 0 0 3 0 0 2 1 3 1 0 0 2
 0 0 3 3 0 1 2 0 0 3 0 1 0 0 0 1 0 3 3 2 0 3 1 1 0 0 3 0 0 0 3 1 2 0 3 1 0
 1 2 1 1 2 0 2 3 0 0 3 0 3 1 0 0 0 0 1 3 0 3 0 0 0 3 0 2 0 1 3 1 1 3 2 3 0
 0 1 3 0 3 0 1 1 0 0 3 0 3 1 0 3 0 0 1 0 0 3 3 0 0 0 0 0 0 2 0 3 3 0 0 0 0
 2 0 0 0 0 0 1 1 0 0 0 0 0 3 1 0 0 0 3 0 0 0 3 0 3 0 0 2 3 0 0 2 1 3 1 0 2
 0 3 0 3 0 0 0 3 0 0 1 0 3 0 0 1 2 0 0 3 0 1 3 1 0 3 0 2 0 0 0 0 1 1 3 0 0
 0 1 3 1 0 1 3 3 2 0 0 0 1 1 0 0 0 0 3 3 1 0 0 3 3 3 1 3 1 0 0 0 3 3 1 1 2
 0 3 0 3 1 1 3 3 0 3 3 3 3 3 3 0 0 3 0 0 3 1 3 1 0 0 2 3 2 0]
```

```
input_text = ['kidney cells in glomeruli ENSG00000005175 High']
inputs = tokenizer(input_text, padding=True, truncation=True, return_tensors="pt")
dataset = Dataset.from_dict(inputs)
predictions = trainer.predict(dataset)
```

```
y_predicted_value = np.argmax(predictions.predictions, axis=1)
```

```
reverse_mapping = {0: 'Approved', 1: 'Enhanced', 2: 'Uncertain', 3: 'Supported'}

predicted_label = reverse_mapping[y_predicted_value[0]]  # Access the first element if it's an array
print(predicted_label)
```

```
Approved
```

```
input_text = ['kidney cells in glomeruli ENSG00000005175 High']
inputs = tokenizer(input_text, padding=True, truncation=True, return_tensors="pt")
dataset = Dataset.from_dict(inputs)
predictions2 = trainer2.predict(dataset)
```

```
y_predicted_value2 = np.argmax(predictions2.predictions, axis=1)
```

```
reverse_mapping = {0: 'Approved', 1: 'Enhanced', 2: 'Uncertain', 3: 'Supported'}

predicted_label2 = reverse_mapping[y_predicted_value2[0]]  # Access the first element if it's an array
print(predicted_label2)
```

```
Approved
```

```
input_text = ['kidney cells in glomeruli ENSG00000005175 High']
inputs = tokenizer(input_text, padding=True, truncation=True, return_tensors="pt")
dataset = Dataset.from_dict(inputs)
predictions3 = trainer3.predict(dataset)
```

```
y_predicted_value3 = np.argmax(predictions3.predictions, axis=1)
```

```
reverse_mapping = {0: 'Approved', 1: 'Enhanced', 2: 'Uncertain', 3: 'Supported'}

predicted_label3 = reverse_mapping[y_predicted_value3[0]]  # Access the first element if it's an array
print(predicted_label3)
```

```
Approved
```

Start coding or generate with AI.