E commerce Text Classification using Transfer Learning

Dataset

This is the classification based E-commerce text dataset for 4 classes - "Electronics", "Household", "Books" and "Clothing & Accessories", which almost cover 80% of any E-commerce website.

Pretrained model used: **DistilBERT**

Import libraries

```
!pip install WordCloud
```

```
Requirement already satisfied: WordCloud in /usr/local/lib/python3.10/dist-packages (1.9.3)
Requirement already satisfied: numpy>=1.6.1 in /usr/local/lib/python3.10/dist-packages (from WordCloud) (1.26.4)
Requirement already satisfied: pillow in /usr/local/lib/python3.10/dist-packages (from WordCloud) (10.4.0)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (from WordCloud) (3.8.0)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->WordCloud) (1.3.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib->WordCloud) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->WordCloud) (4.54.1)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->WordCloud) (1.4.7)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib->WordCloud) (24.1)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib->WordCloud) (3.2.0)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib->WordCloud) (2.8.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib->WordClou
```

```
!pip install datasets
```

```
Collecting datasets
  Downloading datasets-3.1.0-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) (17.0.0)
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) (4.66.6)
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)
Collecting fsspec<=2024.9.0,>=2023.1.0 (from fsspec[http]<=2024.9.0,>=2023.1.0->datasets)
  Downloading fsspec-2024.9.0-py3-none-any.whl.metadata (11 kB)
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) (2.4.3)
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.3.1)
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) (1.5.0)
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (6.1.0)
Requirement already satisfied: yarl<2.0,>=1.12.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (1.17.0)
Requirement already satisfied: async-timeout<5.0,>=4.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->datasets) (4.0.3)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.10/dist-packages (from huggingface-hub>=0.23.0-
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->datasets
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->datasets) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->datasets) (2.2
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.32.2->datasets) (202
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas->datasets) (2.8.2)
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->datasets)
Requirement already satisfied: propcache>=0.2.0 in /usr/local/lib/python3.10/dist-packages (from yarl<2.0,>=1.12.0->aiohttp->datase
Downloading datasets-3.1.0-py3-none-any.whl (480 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 480.6/480.6 kB 4.7 MB/s eta 0:00:00
Downloading dill-0.3.8-py3-none-any.whl (116 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 116.3/116.3 kB 4.2 MB/s eta 0:00:00
Downloading fsspec-2024.9.0-py3-none-any.whl (179 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 179.3/179.3 kB 4.3 MB/s eta 0:00:00
Downloading multiprocess-0.70.16-py310-none-any.whl (134 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 134.8/134.8 kB 6.6 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 8.9 MB/s eta 0:00:00
Installing collected packages: xxhash, fsspec, dill, multiprocess, datasets
  Attempting uninstall: fsspec
    Found existing installation: fsspec 2024.10.0
```

```
        Uninstalling fsspec-2024.10.0:
          Successfully uninstalled fsspec-2024.10.0
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the sou
gcsfs 2024.10.0 requires fsspec==2024.10.0, but you have fsspec 2024.9.0 which is incompatible.
Successfully installed datasets-3.1.0 dill-0.3.8 fsspec-2024.9.0 multiprocess-0.70.16 xxhash-3.5.0
```

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud
import transformers
from transformers import AutoTokenizer, AutoModelForSequenceClassification
from datasets import Dataset, DatasetDict
from datasets.features import Value, ClassLabel
from datasets import Features
from sklearn.metrics import classification_report
from sklearn.model_selection import train_test_split
from transformers import TrainingArguments
from transformers import Trainer
from transformers import DataCollatorWithPadding
```

Import Data

```python
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

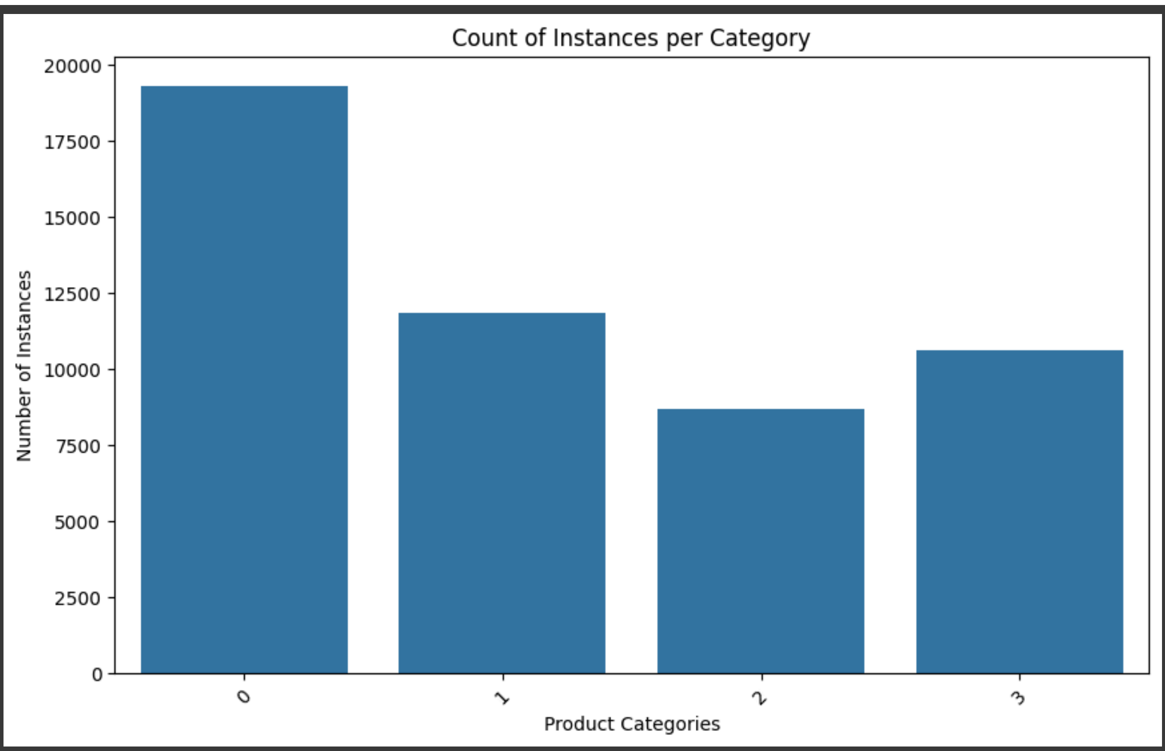
```python
data_path = "/content/drive/MyDrive/NLP/FA2/data/ecommerceDataset.csv"
df = pd.read_csv(data_path, header=None)
df.columns = ['category', 'text']
```

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

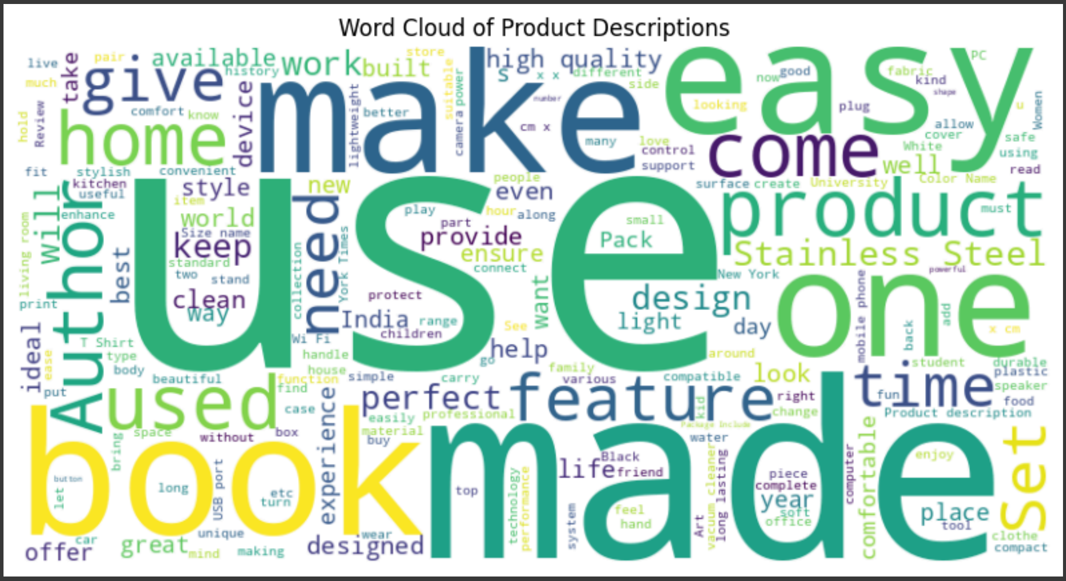
	category	text
0	Household	Paper Plane Design Framed Wall Hanging Motivat...
1	Household	SAF 'Floral' Framed Painting (Wood, 30 inch x ...
2	Household	SAF 'UV Textured Modern Art Print Framed' Pain...
3	Household	SAF Flower Print Framed Painting (Synthetic, 1...
4	Household	Incredible Gifts India Wooden Happy Birthday U...

Visualization

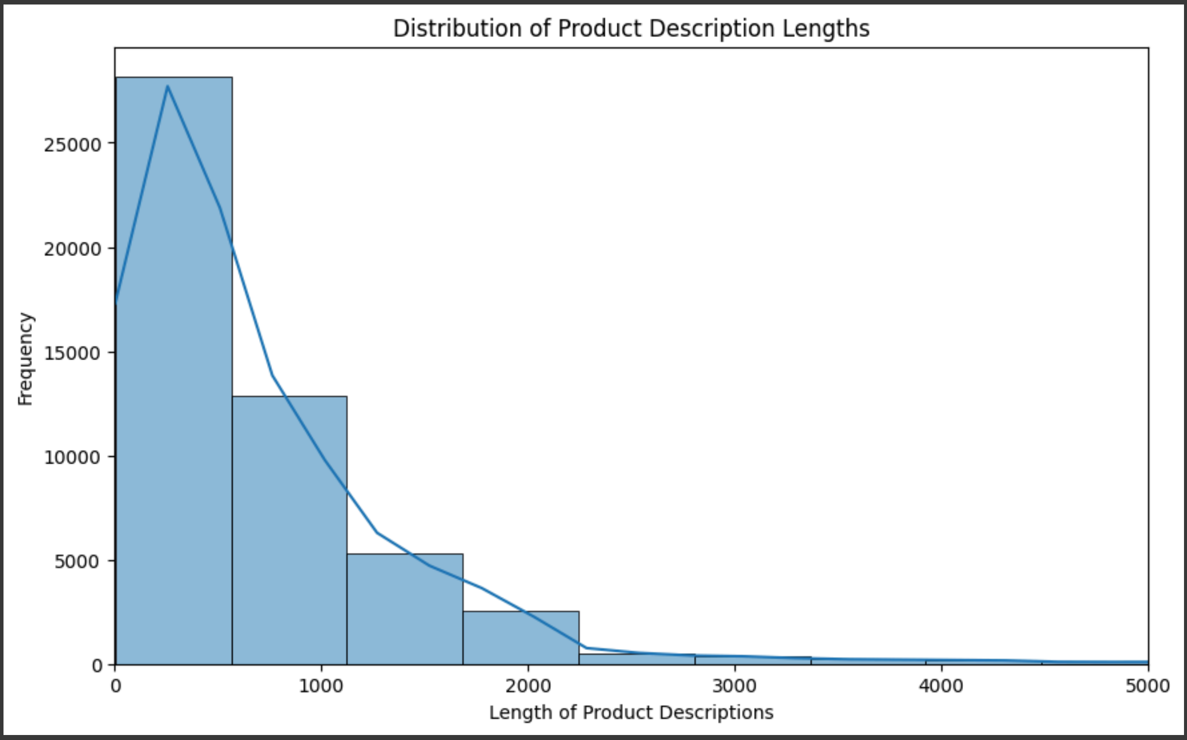
```python
category_counts = df['label'].value_counts()

plt.figure(figsize=(10, 6))
sns.barplot(x=category_counts.index, y=category_counts.values)
plt.xlabel('Product Categories')
plt.ylabel('Number of Instances')
plt.title('Count of Instances per Category')
plt.xticks(rotation=45)
plt.show()
```



```python
valid_descriptions = [desc for desc in df['text'] if isinstance(desc, str)]

text = ' '.join(valid_descriptions)

wordcloud = WordCloud(width=800, height=400, background_color='white').generate(text)

plt.figure(figsize=(10, 6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis('off')
plt.title('Word Cloud of Product Descriptions')
plt.show()
```



```
df['description_length'] = df['text'].str.len()

plt.figure(figsize=(10, 6))
sns.histplot(df['description_length'], bins=90, kde=True)
plt.xlabel('Length of Product Descriptions')
plt.ylabel('Frequency')
plt.title('Distribution of Product Description Lengths')
plt.xlim(0, 5000)
plt.show()
```



Label To Index and Index to Label Maps.

```
# label map
label2idx = {label:i for i, label in enumerate(df.category.unique().tolist())}
label2idx
```

```
{'Household': 0, 'Books': 1, 'Clothing & Accessories': 2, 'Electronics': 3}
```

```
# reverse for correct structure
idx2label = {v:k for k,v in label2idx.items()}
idx2label
```

```
{0: 'Household', 1: 'Books', 2: 'Clothing & Accessories', 3: 'Electronics'}
```

Encoding

```
df['label'] = df.category.map(label2idx)
```

```
df
```

	category	text	label
0	Household	Paper Plane Design Framed Wall Hanging Motivat...	0
1	Household	SAF 'Floral' Framed Painting (Wood, 30 inch x ...	0
2	Household	SAF 'UV Textured Modern Art Print Framed' Pain...	0
3	Household	SAF Flower Print Framed Painting (Synthetic, 1...	0
4	Household	Incredible Gifts India Wooden Happy Birthday U...	0
...	...	...	...
50420	Electronics	Strontium MicroSD Class 10 8GB Memory Card (Bl...	3
50421	Electronics	CrossBeats Wave Waterproof Bluetooth Wireless ...	3
50422	Electronics	Karbonn Titanium Wind W4 (White) Karbonn Titan...	3
50423	Electronics	Samsung Guru FM Plus (SM-B110E/D, Black) Colou...	3
50424	Electronics	Micromax Canvas Win W121 (White)	3

50425 rows × 3 columns

Cleaning

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50425 entries, 0 to 50424
Data columns (total 3 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   category  50425 non-null  object
 1   text      50424 non-null  object
 2   label     50425 non-null  int64 
dtypes: int64(1), object(2)
memory usage: 1.2+ MB
```

We've got one **Null** value in the **text** column.

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

	0
category	0
text	1
label	0

```
df[df.text.isna()]
```

	category	text	label
39330	Clothing & Accessories	NaN	2

since only 1, we can drop it

```
df.dropna(inplace=True)
df.reset_index(drop=True, inplace=True)
df.drop(columns=['category'], inplace=True)
```

Train Test Validation Split

```
train_df, test_df = train_test_split(df, test_size=0.2, shuffle=True, stratify=df['label'], ran
train_df, eval_df = train_test_split(train_df, test_size=0.2, shuffle=True, stratify=train_df['
```

```
train_df.shape, test_df.shape, eval_df.shape
```

```
((32271, 2), (10085, 2), (8068, 2))
```

Verify stratification

```
train_df.label.value_counts(normalize=True)
```

	proportion
label	
0	0.383006
1	0.234421
3	0.210623
2	0.171950

```
test_df.label.value_counts(normalize=True)
```

	proportion
label	
0	0.383044
1	0.234408
3	0.210610
2	0.171939

```
eval_df.label.value_counts(normalize=True)
```

	proportion
label	
0	0.382995
1	0.234383
3	0.210709
2	0.171914

Convert dataframes to transformers datasets

```
features=Features({"text": Value(dtype='string', id=None),
                "label": ClassLabel(num_classes=4,
                                    names=['Household', 'Books', 'Clothing & Accessories', 'Ele
```

Dataframes to datasets.

```
train_dataset = Dataset.from_pandas(train_df, features=features)
test_dataset = Dataset.from_pandas(test_df, features=features)
eval_dataset = Dataset.from_pandas(eval_df, features=features)
```

```
train_dataset
```

```
Dataset({
    features: ['text', 'label'],
    num_rows: 32271
})
```

```
train_dataset.features
```

```
{'text': Value(dtype='string', id=None),
 'label': ClassLabel(num_classes=4, names=['Household', 'Books', 'Clothing & Accessories', 'Electronics'], id=None)}
```

```python
test_dataset
```

```
Dataset({
    features: ['text', 'label'],
    num_rows: 10085
})
```

```python
eval_dataset
```

```
Dataset({
    features: ['text', 'label'],
    num_rows: 8068
})
```

Create a Dataset Dict. (Optional)

Dataset Dict can hold data subsets. For example, our train, test and validation subsets.

```python
dataset = DatasetDict({"train": train_dataset, "test": test_dataset, "validation": eval_dataset
```

```python
dataset
```

```
DatasetDict({
    train: Dataset({
        features: ['text', 'label'],
        num_rows: 32271
    })
    test: Dataset({
        features: ['text', 'label'],
        num_rows: 10085
    })
    validation: Dataset({
        features: ['text', 'label'],
        num_rows: 8068
    })
})
```

Tokenization

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

```
/usr/local/lib/python3.10/dist-packages/huggingface_hub/utils/_token.py:89: 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 it as
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(
tokenizer_config.json: 100%  48.0/48.0 [00:00<00:00, 2.29kB/s]
config.json: 100%  483/483 [00:00<00:00, 20.0kB/s]
vocab.txt: 100%  232k/232k [00:00<00:00, 3.95MB/s]
tokenizer.json: 100%  466k/466k [00:00<00:00, 19.8MB/s]
/usr/local/lib/python3.10/dist-packages/transformers/tokenization_utils_base.py:1601: FutureWarning: `clean_up_tokenization_spaces`
  warnings.warn(
```

```python
def tokenize_function(example):
    return tokenizer(example['text'], truncation=True)
```

```python
tokenized_datasets = dataset.map(tokenize_function, batched=True)
```

Padding

```python
data_collator = DataCollatorWithPadding(tokenizer=tokenizer)
```

Load DistilBERT Model.

```
model = AutoModelForSequenceClassification.from_pretrained(model_checkpoint, num_labels=4)
```

```
model.safetensors: 100%  268M/268M [00:01<00:00, 176MB/s]
Some weights of DistilBertForSequenceClassification were not initialized from the model checkpoint at distilbert-base-uncased and ar
You should probably TRAIN this model on a down-stream task to be able to use it for predictions and inference.
```

Training

```
training_args = TrainingArguments(per_device_train_batch_size=32,
                                  per_device_eval_batch_size=16,
                                  learning_rate=5e-5,
                                  num_train_epochs=5,
                                  evaluation_strategy='epoch',
                                  load_best_model_at_end=True
                                  )
```

```
trainer = Trainer(model=model,
                  args=training_args,
                  train_dataset=tokenized_datasets['train'],
                  eval_dataset=tokenized_datasets['validation'],
                  data_collator=data_collator,
                  tokenizer=tokenizer)
```

```
%%time
trainer.train()
```

```
You're using a DistilBertTokenizerFast tokenizer. Please note that with a fast tokenizer, using the `__call__` method is faster thar
[5045/5045 1:11:57, Epoch 5/5]
```

Epoch	Training Loss	Validation Loss
1	0.139200	0.139756
2	0.076800	0.109455
3	0.042400	0.103477
4	0.018700	0.097567
5	0.012300	0.104120

```
CPU times: user 1h 11min 54s, sys: 7.99 s, total: 1h 12min 2s
Wall time: 1h 11min 59s
TrainOutput(global_step=5045, training_loss=0.06804902793690046, metrics={'train_runtime': 4319.3742, 'train_samples_per_second':
```

Inference

```
predictions = trainer.predict(tokenized_datasets["test"])
predictions.predictions.shape
```

```
preds = np.argmax(predictions.predictions, axis=-1)
test_df['preds'] = preds
```

```
print(classification_report(test_df.label, test_df.preds, target_names=list(idx2label.values()))
```

```
                        precision    recall  f1-score   support

             Household       0.98      0.98      0.98      3863
                 Books       0.98      0.98      0.98      2364
Clothing & Accessories       0.99      0.99      0.99      1734
           Electronics       0.98      0.97      0.97      2124
```

```
    accuracy                           0.98     10085
   macro avg       0.98      0.98      0.98     10085
weighted avg       0.98      0.98      0.98     10085
```