Predict Product Prices

IMPORTANT please read me!!

When you run the pip installs below, you may get an error from pip complaining about an incompatible version of fsspec.

You should ignore that error! The version of fsspec is the right version, needed by HuggingFace.

If you ask ChatGPT, it will encourage you to pip install a more recent version of fsspec. But that would be problematic; HuggingFace will fail to load the dataset later with an obscure error about file systems.

So please run the pip installs as they appear below, and look the other way if you get an error!

```
# pip installs

!pip install -q --upgrade torch==2.5.1+cu124 torchvision==0.20.1+cu124 torchaudio==2.5.1+cu124 --index-url https://downloa
!pip install -q --upgrade requests==2.32.3 bitsandbytes==0.46.0 transformers==4.48.3 accelerate==1.3.0 datasets==3.2.0 peft
```

```
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 908.2/908.2 MB 1.6 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.3/7.3 MB 133.9 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.4/3.4 MB 106.0 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 24.6/24.6 MB 116.9 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 883.7/883.7 kB 66.0 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 13.8/13.8 MB 145.9 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 664.8/664.8 MB 1.7 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 363.4/363.4 MB 3.0 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 211.5/211.5 MB 12.9 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 56.3/56.3 MB 46.9 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 127.9/127.9 MB 20.8 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 207.5/207.5 MB 7.8 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 188.7/188.7 MB 3.5 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 99.1/99.1 kB 10.4 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 21.1/21.1 MB 125.1 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 209.6/209.6 MB 7.4 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.2/6.2 MB 142.4 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 44.4/44.4 kB 3.8 MB/s eta 0:00:00
 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 64.9/64.9 kB 886.2 kB/s eta 0:00:00
 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 67.0/67.0 MB 38.5 MB/s eta 0:00:00
 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 9.7/9.7 MB 78.8 MB/s eta 0:00:00
 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 336.6/336.6 kB 35.6 MB/s eta 0:00:00
 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 480.6/480.6 kB 37.5 MB/s eta 0:00:00
 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 374.8/374.8 kB 39.1 MB/s eta 0:00:00
 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 313.9/313.9 kB 31.5 MB/s eta 0:00:00
 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 8.7/8.7 MB 151.1 MB/s eta 0:00:00
 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 20.0/20.0 MB 130.4 MB/s eta 0:00:00
 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 179.3/179.3 kB 21.3 MB/s eta 0:00:00
 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.1/3.1 MB 131.5 MB/s eta 0:00:00
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is
google-colab 1.0.0 requires requests==2.32.4, but you have requests 2.32.3 which is incompatible.
gcsfs 2025.3.0 requires fsspec==2025.3.0, but you have fsspec 2024.9.0 which is incompatible.
google-adk 1.17.0 requires requests<3.0.0,>=2.32.4, but you have requests 2.32.3 which is incompatible.
```

```
# imports
# With much thanks to Islam S. for identifying that there was a missing import!

import os
import re
import math
from tqdm import tqdm
from google.colab import userdata
from huggingface_hub import login
import torch
import transformers
from transformers import AutoModelForCausalLM, AutoTokenizer, TrainingArguments, set_seed, BitsAndBytesConfig
from datasets import load_dataset, Dataset, DatasetDict
import wandb
from peft import LoraConfig
from trl import SFTTrainer, SFTConfig
from datetime import datetime
import matplotlib.pyplot as plt
```

```
# Constants

BASE_MODEL = "Qwen/Qwen2.5-7B"
PROJECT_NAME = "pricer"
HF_USER = "harish-anandaramanujam"

# Data
```

```
DATASET_NAME = f"{HF_USER}/amzn-appliances-price-lite-data"

MAX_SEQUENCE_LENGTH = 182

# Run name for saving the model in the hub

RUN_NAME =  f"{datetime.now():%Y-%m-%d_%H.%M.%S}"
PROJECT_RUN_NAME = f"{PROJECT_NAME}-{RUN_NAME}"
HUB_MODEL_NAME = f"{HF_USER}/{PROJECT_RUN_NAME}"

# Hyperparameters for QLoRA

LORA_R = 32
LORA_ALPHA = 64
TARGET_MODULES = ["q_proj", "v_proj", "k_proj", "o_proj"]
LORA_DROPOUT = 0.1
QUANT_4_BIT = True

# Hyperparameters for Training

EPOCHS = 1 # you can do more epochs if you wish, but only 1 is needed - more is probably overkill
BATCH_SIZE = 4 # on an A100 box this can go up to 16
GRADIENT_ACCUMULATION_STEPS = 1
LEARNING_RATE = 1e-4
LR_SCHEDULER_TYPE = 'cosine'
WARMUP_RATIO = 0.03
OPTIMIZER = "paged_adamw_32bit"

# Admin config - note that SAVE_STEPS is how often it will upload to the hub
# I've changed this from 5000 to 2000 so that you get more frequent saves

STEPS = 50
SAVE_STEPS = 2000
LOG_TO_WANDB = True

%matplotlib inline
```

```
HUB_MODEL_NAME
```

```
'harish-anandaramanujam/pricer-2025-10-29_05.43.51'
```

More on Optimizers

https://huggingface.co/docs/transformers/main/en/perf_train_gpu_one#optimizer-choice

The most common is Adam or AdamW (Adam with Weight Decay).
Adam achieves good convergence by storing the rolling average of the previous gradients; however, it adds an additional memory footprint of the order of the number of model parameters.

Log in to HuggingFace and Weights & Biases

If you don't already have a HuggingFace account, visit https://huggingface.co to sign up and create a token.

Then select the Secrets for this Notebook by clicking on the key icon in the left, and add a new secret called `HF_TOKEN` with the value as your token.

Repeat this for weightsandbiases at https://wandb.ai and add a secret called `WANDB_API_KEY`

```
# Log in to HuggingFace

hf_token = userdata.get('HF_TOKEN')
login(hf_token, add_to_git_credential=True)
```

```
# Log in to Weights & Biases
wandb_api_key = userdata.get('WANDB_API_KEY')
os.environ["WANDB_API_KEY"] = wandb_api_key
wandb.login()

# Configure Weights & Biases to record against our project
os.environ["WANDB_PROJECT"] = PROJECT_NAME
os.environ["WANDB_LOG_MODEL"] = "checkpoint" if LOG_TO_WANDB else "end"
os.environ["WANDB_WATCH"] = "gradients"
```

```
/usr/local/lib/python3.12/dist-packages/notebook/notebookapp.py:191: SyntaxWarning: invalid escape sequence '\/'
  | |_| | '_ \/ _` / _` |  _/ -_)
```

```
wandb: Currently logged in as: harish-anandaramanujam (harish-anandaramanujam-mlops) to https://api.wandb.ai. Use `wandb log
```

```
dataset = load_dataset(DATASET_NAME)
train = dataset['train']
test = dataset['test']
```

```
README.md: 100%                                  412/412 [00:00<00:00, 46.9kB/s]
data/train-00000-of-00001.parquet: 100%          9.50M/9.50M [00:03<00:00, 2.55MB/s]
data/test-00000-of-00001.parquet: 100%           759k/759k [00:00<00:00, 1.43MB/s]
Generating train split: 100%                     25000/25000 [00:00<00:00, 237714.12 examples/s]
Generating test split: 100%                      2000/2000 [00:00<00:00, 128521.65 examples/s]
```

```
# if you wish to reduce the training dataset to 20,000 points instead, then uncomment this line:
# train = train.select(range(20000))
```

```
if LOG_TO_WANDB:
  wandb.init(project=PROJECT_NAME, name=RUN_NAME)
```

```
Changes to your `wandb` environment variables will be ignored because your `wandb` session has already started. For more information on how to modify your settings with `wandb.init()` arguments, please refer to the W&B docs.
Tracking run with wandb version 0.22.3
Run data is saved locally in /content/wandb/run-20251029_054410-pu9r1t9u
Syncing run 2025-10-29_05.43.51 to Weights & Biases (docs)
View project at https://wandb.ai/harish-anandaramanujam-mlops/pricer
View run at https://wandb.ai/harish-anandaramanujam-mlops/pricer/runs/pu9r1t9u
```

Now load the Tokenizer and Model

The model is "quantized" - we are reducing the precision to 4 bits.

```
# pick the right quantization

if QUANT_4_BIT:
  quant_config = BitsAndBytesConfig(
    load_in_4bit=True,
    bnb_4bit_use_double_quant=True,
    bnb_4bit_compute_dtype=torch.bfloat16,
    bnb_4bit_quant_type="nf4"
  )
else:
  quant_config = BitsAndBytesConfig(
    load_in_8bit=True,
    bnb_8bit_compute_dtype=torch.bfloat16
  )
```

```
# Load the Tokenizer and the Model

tokenizer = AutoTokenizer.from_pretrained(BASE_MODEL, trust_remote_code=True)
tokenizer.pad_token = tokenizer.eos_token
tokenizer.padding_side = "right"

base_model = AutoModelForCausalLM.from_pretrained(
    BASE_MODEL,
    quantization_config=quant_config,
    device_map="auto",
)
base_model.generation_config.pad_token_id = tokenizer.pad_token_id

print(f"Memory footprint: {base_model.get_memory_footprint() / 1e6:.1f} MB")
```

```
tokenizer_config.json:   7.23k/?  [00:00<00:00,  925kB/s]
vocab.json:   2.78M/?  [00:00<00:00,  107MB/s]
merges.txt:   1.67M/?  [00:00<00:00,  83.0MB/s]
tokenizer.json:   7.03M/?  [00:00<00:00,  170MB/s]
config.json:  100%                                   686/686  [00:00<00:00,  92.8kB/s]
model.safetensors.index.json:   27.8k/?  [00:00<00:00,  2.88MB/s]
Downloading  shards:  100%                           4/4  [00:40<00:00,  10.01s/it]
model-00001-of-00004.safetensors:  100%              3.95G/3.95G  [00:09<00:00,  793MB/s]
model-00002-of-00004.safetensors:  100%              3.86G/3.86G  [00:09<00:00,  793MB/s]
model-00003-of-00004.safetensors:  100%              3.86G/3.86G  [00:09<00:00,  730MB/s]
model-00004-of-00004.safetensors:  100%              3.56G/3.56G  [00:09<00:00,  605MB/s]
Loading  checkpoint  shards:  100%                   4/4  [00:09<00:00,   2.29s/it]
generation_config.json:  100%                        138/138  [00:00<00:00,  18.8kB/s]
Memory footprint: 5443.3 MB
```

Data Collator

It's important that we ensure during Training that we are not trying to train the model to predict the description of products; only their price.

We need to tell the trainer that everything up to "Price is $" is there to give context to the model to predict the next token, but does not need to be learned.

The trainer needs to teach the model to predict the token(s) after "Price is $".

There is a complicated way to do this by setting Masks, but luckily HuggingFace provides a super simple helper class to take care of this for us.

```
from trl import DataCollatorForCompletionOnlyLM
response_template = "Price is $"
collator = DataCollatorForCompletionOnlyLM(response_template, tokenizer=tokenizer)
```

AND NOW

We set up the configuration for Training

We need to create 2 objects:

A LoraConfig object with our hyperparameters for LoRA

An SFTConfig with our overall Training parameters

```
# First, specify the configuration parameters for LoRA

lora_parameters = LoraConfig(
    lora_alpha=LORA_ALPHA,
    lora_dropout=LORA_DROPOUT,
    r=LORA_R,
    bias="none",
    task_type="CAUSAL_LM",
    target_modules=TARGET_MODULES,
)

# Next, specify the general configuration parameters for training

train_parameters = SFTConfig(
    output_dir=PROJECT_RUN_NAME,
    num_train_epochs=EPOCHS,
    per_device_train_batch_size=BATCH_SIZE,
    per_device_eval_batch_size=1,
    eval_strategy="no",
    gradient_accumulation_steps=GRADIENT_ACCUMULATION_STEPS,
    optim=OPTIMIZER,
    save_steps=SAVE_STEPS,
    save_total_limit=10,
    logging_steps=STEPS,
    learning_rate=LEARNING_RATE,
    weight_decay=0.001,
    fp16=False,
    bf16=True,
    max_grad_norm=0.3,
    max_steps=-1,
    warmup_ratio=WARMUP_RATIO,
    group_by_length=True,
    lr_scheduler_type=LR_SCHEDULER_TYPE,
    report_to="wandb" if LOG_TO_WANDB else None,
    run_name=RUN_NAME,
    max_seq_length=MAX_SEQUENCE_LENGTH,
    dataset_text_field="text",
    save_strategy="steps",
    hub_strategy="every_save",
    push_to_hub=True,
    hub_model_id=HUB_MODEL_NAME,
    hub_private_repo=True
)

# And now, the Supervised Fine Tuning Trainer will carry out the fine-tuning
# Given these 2 sets of configuration parameters
# The latest version of trl is showing a warning about labels - please ignore this warning
# But let me know if you don't see good training results (loss coming down).

fine_tuning = SFTTrainer(
    model=base_model,
    train_dataset=train,
    peft_config=lora_parameters,
    args=train_parameters,
    data_collator=collator
  )
```

```
Map: 100%                                   25000/25000 [00:03<00:00, 7268.41 examples/s]
```

Free Colab may fail

You may need more powerful machine with google colab subscription

```
# Fine-tune!
fine_tuning.train()

# Push our fine-tuned model to Hugging Face
fine_tuning.model.push_to_hub(PROJECT_RUN_NAME, private=False)
print(f"Saved to the hub: {PROJECT_RUN_NAME}")
```