Predict Product Prices with base qwen

Using base qwen (without any Fine-tuning) to predict amazon product prices from my hugging face dataset. This results helps us to evaluate pre and post fine tuning

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 - ignore the error message!

!pip install -q --upgrade torch==2.5.1+cu124 torchvision==0.20.1+cu124 torchaudio==2.5.1+cu124 --index-url https://download.
!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 963.0 kB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.3/7.3 MB 84.8 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.4/3.4 MB 90.7 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 24.6/24.6 MB 51.6 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 883.7/883.7 kB 53.0 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 13.8/13.8 MB 68.5 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 664.8/664.8 MB 2.7 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 363.4/363.4 MB 4.5 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 211.5/211.5 MB 5.9 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 56.3/56.3 MB 13.7 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 127.9/127.9 MB 7.9 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 207.5/207.5 MB 5.9 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 188.7/188.7 MB 5.8 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 99.1/99.1 kB 9.9 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 21.1/21.1 MB 100.5 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 209.6/209.6 MB 5.9 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.2/6.2 MB 112.3 MB/s eta 0:00:00
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 44.4/44.4 kB 2.6 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 64.9/64.9 kB 5.9 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 67.0/67.0 MB 13.0 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 9.7/9.7 MB 123.4 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 336.6/336.6 kB 35.8 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 480.6/480.6 kB 41.8 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 374.8/374.8 kB 39.0 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 313.9/313.9 kB 21.3 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 8.7/8.7 MB 111.9 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 179.3/179.3 kB 18.4 MB/s eta 0:00:00
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.1/3.1 MB 103.1 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

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, BitsAndBytesConfig, TrainingArguments, set_seed
from peft import LoraConfig, PeftModel
from datasets import load_dataset, Dataset, DatasetDict
from datetime import datetime
import matplotlib.pyplot as plt
```

```
# Tokenizers

LLAMA_3_1 = "meta-llama/Meta-Llama-3.1-8B"
QWEN_2_5 = "Qwen/Qwen2.5-7B"
GEMMA_2 = "google/gemma-2-9b"
PHI_3 = "microsoft/Phi-3-medium-4k-instruct"

# Constants
```

```
BASE_MODEL = QWEN_2_5
HF_USER = "harish-anandaramanujam"
DATASET_NAME = f"{HF_USER}/amzn-appliances-price-lite-data"
MAX_SEQUENCE_LENGTH = 182
QUANT_4_BIT = True

# Used for writing to output in color

GREEN = "\033[92m"
YELLOW = "\033[93m"
RED = "\033[91m"
RESET = "\033[0m"
COLOR_MAP = {"red":RED, "orange": YELLOW, "green": GREEN}

%matplotlib inline
```

Log in to HuggingFace

Make Sure HF_TOKEN in colab secrets point to huggingface account

```
# Log in to HuggingFace

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

Comparing different Tokenizer

Just checking out how different tokenizers tokenize the input as visual output, instead of that's happening behind the scenes

```
def investigate_tokenizer(model_name):
  print("Investigating tokenizer for", model_name)
  tokenizer = AutoTokenizer.from_pretrained(model_name, trust_remote_code=True)
  for number in [0, 1, 10, 100, 999, 1000]:
    tokens = tokenizer.encode(str(number), add_special_tokens=False)
    print(f"The tokens for {number}: {tokens}")
```

```
# Now we will try this with each model: LLAMA_3_1, QWEN_2_5, GEMMA_2, PHI_3
# LLMA needs authorization to access through huggingface request

investigate_tokenizer(QWEN_2_5)
```

```
Investigating tokenizer for Qwen/Qwen2.5-7B
tokenizer_config.json:    7.23k/?  [00:00<00:00,  224kB/s]

vocab.json:    2.78M/?  [00:00<00:00,  4.78MB/s]

merges.txt:    1.67M/?  [00:00<00:00,  8.86MB/s]

tokenizer.json:    7.03M/?  [00:00<00:00,  18.9MB/s]
The tokens for 0: [15]
The tokens for 1: [16]
The tokens for 10: [16, 15]
The tokens for 100: [16, 15, 15]
The tokens for 999: [24, 24, 24]
The tokens for 1000: [16, 15, 15, 15]
```

```
investigate_tokenizer(PHI_3)
```

```
Investigating tokenizer for microsoft/Phi-3-medium-4k-instruct
tokenizer_config.json:    3.15k/?  [00:00<00:00,  192kB/s]

tokenizer.model:  100%                        500k/500k  [00:00<00:00,  1.09MB/s]

tokenizer.json:    1.84M/?  [00:00<00:00,  23.3MB/s]

added_tokens.json:  100%                      293/293  [00:00<00:00,  33.2kB/s]

special_tokens_map.json:  100%                568/568  [00:00<00:00,  57.7kB/s]
The tokens for 0: [29871, 29900]
The tokens for 1: [29871, 29896]
The tokens for 10: [29871, 29896, 29900]
The tokens for 100: [29871, 29896, 29900, 29900]
The tokens for 999: [29871, 29929, 29929, 29929]
The tokens for 1000: [29871, 29896, 29900, 29900, 29900]
```

Load data

Uploaded it to Hugging Face, so it's easy to retrieve it now

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

```
README.md: 100%                                    412/412 [00:00<00:00, 44.4kB/s]
data/train-00000-of-00001.parquet: 100%            9.50M/9.50M [00:01<00:00, 7.17MB/s]
data/test-00000-of-00001.parquet: 100%             759k/759k [00:00<00:00, 4.00MB/s]
Generating train split: 100%                       25000/25000 [00:00<00:00, 191045.49 examples/s]
Generating test split: 100%                        2000/2000 [00:00<00:00, 86804.45 examples/s]
```

```
test[0]
```

```
{'text': 'How much does this cost to the nearest dollar?\n\nCONTINENTAL REFRIGERATOR 2-705 Gasket, SNAP in\nSPECIFICATIONS
LENGTH 22 3/8, 568 mm WIDTH 25 3/8, 644 mm WEIGHT 1.225 lb GASKET TYPE 4 SIDED, MAGNETIC, DART PART REFERENCE INFO
CONTINENTAL REFRIGERATOR 2-705 MODEL REFERENCE INFO CONTINENTAL REFRIGERATOR WORK TOP REFRIGERATOR MODELS SW48 CONTINENTAL
REFRIGERATOR REFRIGERATOR DOOR GASKET 2-705 High Quality Gasket Brand Name Continental Refrigerator, Model Info Weight
1.2\n\nPrice is $',
 'price': 36.22}
```

Prepare our Base Llama Model for evaluation

Load our base model with 4 bit quantization and try out 1 example

```
## 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() / 1e9:.1f} GB")
```

```
config.json: 100%                                  686/686 [00:00<00:00, 72.1kB/s]
model.safetensors.index.json:      27.8k/? [00:00<00:00, 682kB/s]
Downloading shards: 100%                           4/4 [02:53<00:00, 40.51s/it]
model-00001-of-00004.safetensors: 100%             3.95G/3.95G [00:53<00:00, 125MB/s]
model-00002-of-00004.safetensors: 100%             3.86G/3.86G [00:41<00:00, 31.0MB/s]
model-00003-of-00004.safetensors: 100%             3.86G/3.86G [00:48<00:00, 201MB/s]
model-00004-of-00004.safetensors: 100%             3.56G/3.56G [00:29<00:00, 174MB/s]
Loading checkpoint shards: 100%                    4/4 [01:21<00:00, 21.32s/it]
generation_config.json: 100%                       138/138 [00:00<00:00, 13.6kB/s]
Memory footprint: 5.4 GB
```

Extract Prices from noise (needed when we get the model output)

```
def extract_price(s):
    if "Price is $" in s:
      contents = s.split("Price is $")[1]
      contents = contents.replace(',','').replace('$','')
      match = re.search(r"[-+]?\d*\.\d+|\d+", contents)
      return float(match.group()) if match else 0
    return 0
```

```
extract_price("Price is $999 blah blah so cheap")
```

```
999.0
```

Model prediction

```
def model_predict(prompt):
    set_seed(42)
    inputs = tokenizer.encode(prompt, return_tensors="pt").to("cuda")
    attention_mask = torch.ones(inputs.shape, device="cuda")
    outputs = base_model.generate(inputs, max_new_tokens=4, attention_mask=attention_mask, num_return_sequences=1)
    response = tokenizer.decode(outputs[0])
    return extract_price(response)
```

```
model_predict(test[0]['text'])
```

```
1.22
```

Evaluation!

Trying out our base qwen 2.5 model against the Test dataset

```
class Tester:

    def __init__(self, predictor, data, title=None, size=250):
        self.predictor = predictor
        self.data = data
        self.title = title or predictor.__name__.replace("_", " ").title()
        self.size = size
        self.guesses = []
        self.truths = []
        self.errors = []
        self.sles = []
        self.colors = []

    def color_for(self, error, truth):
        if error<40 or error/truth < 0.2:
            return "green"
        elif error<80 or error/truth < 0.4:
            return "orange"
        else:
            return "red"

    def run_datapoint(self, i):
        datapoint = self.data[i]
        guess = self.predictor(datapoint["text"])
        truth = datapoint["price"]
        error = abs(guess - truth)
        log_error = math.log(truth+1) - math.log(guess+1)
        sle = log_error ** 2
        color = self.color_for(error, truth)
        title = datapoint["text"].split("\n\n")[1][:20] + "..."
        self.guesses.append(guess)
        self.truths.append(truth)
        self.errors.append(error)
        self.sles.append(sle)
        self.colors.append(color)
        print(f"{COLOR_MAP[color]}{i+1}: Guess: ${guess:,.2f} Truth: ${truth:,.2f} Error: ${error:,.2f} SLE: {sle:,.2f} Ite

    def chart(self, title):
        max_error = max(self.errors)
        plt.figure(figsize=(12, 8))
        max_val = max(max(self.truths), max(self.guesses))
        plt.plot([0, max_val], [0, max_val], color='deepskyblue', lw=2, alpha=0.6)
        plt.scatter(self.truths, self.guesses, s=3, c=self.colors)
        plt.xlabel('Ground Truth')
        plt.ylabel('Model Estimate')
        plt.xlim(0, max_val)
        plt.ylim(0, max_val)
        plt.title(title)
        plt.show()

    def report(self):
        average_error = sum(self.errors) / self.size
        rmsle = math.sqrt(sum(self.sles) / self.size)
        hits = sum(1 for color in self.colors if color=="green")
        title = f"{self.title} Error=${average_error:,.2f} RMSLE={rmsle:,.2f} Hits={hits/self.size*100:.1f}%"
        self.chart(title)

    def run(self):
        self.error = 0
        for i in range(self.size):
            self.run_datapoint(i)
        self.report()

    @classmethod
    def test(cls, function, data):
        cls(function, data).run()
```

```
Tester.test(model_predict, test)
```

```
1: Guess: $1.22 Truth: $36.22 Error: $35.00 SLE: 7.95 Item: CONTINENTAL REFRIGER...
2: Guess: $129.00 Truth: $880.00 Error: $751.00 SLE: 3.66 Item: 12,000 BTU CoVac Duc...
3: Guess: $10.00 Truth: $95.67 Error: $85.67 SLE: 4.72 Item: Whirlpool Oven
Produ...
4: Guess: $10.90 Truth: $21.98 Error: $11.08 SLE: 0.43 Item: Frigidaire Basket
Th...
5: Guess: $10.90 Truth: $95.14 Error: $84.24 SLE: 4.37 Item: GE Genuine OEM Compr...
6: Guess: $12.90 Truth: $8.99 Error: $3.91 SLE: 0.11 Item: 6 Pack Reusable K Cu...
7: Guess: $10.90 Truth: $8.66 Error: $2.24 SLE: 0.04 Item: Silicone Kitchen Sto...
8: Guess: $10.90 Truth: $42.52 Error: $31.62 SLE: 1.68 Item: Whirlpool Refrigerat...
9: Guess: $19.90 Truth: $9.99 Error: $9.91 SLE: 0.41 Item: Oven Temperature Sen...
10: Guess: $10.90 Truth: $42.99 Error: $32.09 SLE: 1.71 Item: Upgraded Full Real M...
11: Guess: $10.00 Truth: $21.99 Error: $11.99 SLE: 0.54 Item: BUNN Basket Filter (...
12: Guess: $129.00 Truth: $53.98 Error: $75.02 SLE: 0.74 Item: Vuenyrun Automatic E...
13: Guess: $10.90 Truth: $20.99 Error: $10.09 SLE: 0.38 Item: Replacement Aluminum...
14: Guess: $10.90 Truth: $14.99 Error: $4.09 SLE: 0.09 Item: FQQWEE Egg Holder fo...
15: Guess: $129.00 Truth: $259.99 Error: $130.99 SLE: 0.49 Item: MAT EXPERT 3.4 Cu.Ft...
16: Guess: $100.00 Truth: $250.33 Error: $150.33 SLE: 0.83 Item: GE Icemaker Assembly...
17: Guess: $10.90 Truth: $34.59 Error: $23.69 SLE: 1.20 Item: Refrigerator Defrost...
18: Guess: $119.00 Truth: $275.99 Error: $156.99 SLE: 0.70 Item: 30 Inch Gas Cooktop ...
19: Guess: $10.90 Truth: $19.96 Error: $9.06 SLE: 0.32 Item: Washer Drain Pump Mo...
20: Guess: $10.90 Truth: $6.99 Error: $3.91 SLE: 0.16 Item: Dryer Timer Control ...
21: Guess: $10.90 Truth: $6.59 Error: $4.31 SLE: 0.20 Item: Coffee Paper Filter ...
22: Guess: $129.00 Truth: $343.99 Error: $214.99 SLE: 0.95 Item: Range Hood 30 inch w...
23: Guess: $12.90 Truth: $21.99 Error: $9.09 SLE: 0.25 Item: GRONGU Cold Brew Cof...
24: Guess: $10.90 Truth: $9.99 Error: $0.91 SLE: 0.01 Item: Dryer Lint Filter Re...
25: Guess: $1.99 Truth: $77.05 Error: $75.06 SLE: 10.64 Item: Berkel Carriage Knob...
26: Guess: $12.90 Truth: $19.95 Error: $7.05 SLE: 0.17 Item: Perfect Pod Eco-Fill...
27: Guess: $12.90 Truth: $21.99 Error: $9.09 SLE: 0.25 Item: Range Kleen Style A ...
28: Guess: $10.90 Truth: $85.02 Error: $74.12 SLE: 3.91 Item: GENUINE Whirlpool Hi...
29: Guess: $12.90 Truth: $20.85 Error: $7.95 SLE: 0.20 Item: Cold Water Inlet Val...
30: Guess: $10.90 Truth: $20.86 Error: $9.96 SLE: 0.37 Item: (4 PACK) - Deflecto ...
31: Guess: $129.00 Truth: $167.33 Error: $38.33 SLE: 0.07 Item: Igloo Automatic Self...
32: Guess: $12.90 Truth: $9.88 Error: $3.02 SLE: 0.06 Item: COT Made for Sunny D...
33: Guess: $10.90 Truth: $52.87 Error: $41.97 SLE: 2.28 Item: Whirlpool Temperatur...
34: Guess: $10.90 Truth: $9.99 Error: $0.91 SLE: 0.01 Item: 2-Pack Single Serve ...
35: Guess: $10.90 Truth: $11.34 Error: $0.44 SLE: 0.00 Item: Compatible With Whir...
36: Guess: $10.90 Truth: $5.99 Error: $4.91 SLE: 0.28 Item: Reusable K Cups, 2 P...
37: Guess: $100.00 Truth: $32.29 Error: $67.71 SLE: 1.23 Item: Supco WV9346 Washer ...
38: Guess: $10.90 Truth: $30.79 Error: $19.89 SLE: 0.97 Item: Sensor OEM Mania New...
39: Guess: $10.90 Truth: $14.99 Error: $4.09 SLE: 0.09 Item: Supplying Demand Ref...
40: Guess: $10.90 Truth: $14.97 Error: $4.07 SLE: 0.09 Item: Fit for Samsung Drye...
41: Guess: $10.90 Truth: $36.99 Error: $26.09 SLE: 1.35 Item: 5 Pack Range Burner ...
42: Guess: $100.00 Truth: $99.95 Error: $0.05 SLE: 0.00 Item: Case Club Pre-Cut Wa...
43: Guess: $10.90 Truth: $18.65 Error: $7.75 SLE: 0.25 Item: Carbon Range Filter ...
44: Guess: $19.90 Truth: $12.99 Error: $6.91 SLE: 0.16 Item: Humidity and Tempera...
45: Guess: $10.90 Truth: $46.12 Error: $35.22 SLE: 1.89 Item: General Electric Ref...
46: Guess: $10.90 Truth: $19.99 Error: $9.09 SLE: 0.32 Item: IOYIJOI Humidifier F...
47: Guess: $12.90 Truth: $9.97 Error: $2.93 SLE: 0.06 Item: Refrigerator Defrost...
48: Guess: $100.00 Truth: $18.95 Error: $81.05 SLE: 2.63 Item: Lifetime Dryer Heati...
49: Guess: $129.00 Truth: $179.99 Error: $50.99 SLE: 0.11 Item: Weceleh Induction Co...
50: Guess: $10.90 Truth: $6.49 Error: $4.41 SLE: 0.21 Item: 25 Universal Gray Gr...
51: Guess: $10.90 Truth: $25.98 Error: $15.08 SLE: 0.67 Item: CAPMESSO Coffee Caps...
52: Guess: $12.90 Truth: $48.49 Error: $35.59 SLE: 1.61 Item: OCTOPUS Compatible w...
53: Guess: $10.90 Truth: $13.99 Error: $3.09 SLE: 0.05 Item: EXCELPURE Replacemen...
54: Guess: $10.90 Truth: $15.63 Error: $4.73 SLE: 0.11 Item: Sealed Unit Parts SU...
55: Guess: $119.00 Truth: $320.41 Error: $201.41 SLE: 0.97 Item: Summit Wide 115V Rad...
56: Guess: $19.90 Truth: $103.77 Error: $83.87 SLE: 2.60 Item: ForeverPRO Timer for...
```

```
Start coding or generate with AI.
59: Guess: $10.90 Truth: $10.99 Error: $0.09 SLE: 0.00 Item: Coffee Filter Basket...
60: Guess: $10.90 Truth: $9.98 Error: $0.92 SLE: 0.01 Item: YQL #2 Cone Coffee F...
61: Guess: $12.90 Truth: $12.39 Error: $0.51 SLE: 0.00 Item: MEISHIDA Egg Holder ...
62: Guess: $10.90 Truth: $41.59 Error: $30.69 SLE: 1.63 Item: Washer Door Lock Lat...
63: Guess: $12.90 Truth: $33.99 Error: $21.09 SLE: 0.85 Item: FrigiLife Replacemen...
64: Guess: $10.00 Truth: $150.18 Error: $140.18 SLE: 6.87 Item: STAR Lower Door
Prod...
65: Guess: $1.99 Truth: $17.99 Error: $16.00 SLE: 3.42 Item: i Cafilas Refillable...
66: Guess: $100.00 Truth: $23.99 Error: $76.01 SLE: 1.95 Item: 279838 Dryer Heating...
67: Guess: $10.90 Truth: $9.90 Error: $1.00 SLE: 0.01 Item: steel filter 10pcs a...
68: Guess: $10.90 Truth: $24.95 Error: $14.05 SLE: 0.61 Item: 6' STAINLESS STEEL B...
69: Guess: $10.90 Truth: $22.83 Error: $11.93 SLE: 0.48 Item: Dundas Jafine EXWTZW...
70: Guess: $10.90 Truth: $12.95 Error: $2.05 SLE: 0.03 Item: GE Part Number FUSE ...
71: Guess: $10.90 Truth: $7.91 Error: $2.99 SLE: 0.08 Item: HQRP Filter compatib...
72: Guess: $12.90 Truth: $9.22 Error: $3.68 SLE: 0.09 Item: Small Size Pour Over...
73: Guess: $10.90 Truth: $36.99 Error: $26.09 SLE: 1.35 Item: HASMX Replacement Hu...
74: Guess: $10.90 Truth: $141.51 Error: $130.61 SLE: 6.16 Item: Kit Diaphragm Lid
52...
75: Guess: $12.90 Truth: $11.99 Error: $0.91 SLE: 0.00 Item: Reusable Grey 3 Part...
76: Guess: $10.90 Truth: $20.24 Error: $9.34 SLE: 0.34 Item: Washer Drive Belt - ...
77: Guess: $10.90 Truth: $8.89 Error: $2.01 SLE: 0.03 Item: (2023 Update) 661566...
78: Guess: $10.90 Truth: $19.97 Error: $9.07 SLE: 0.32 Item: 4-Pack Air Filter Fa...
79: Guess: $10.90 Truth: $6.49 Error: $4.41 SLE: 0.21 Item: Cheerock Collapsible
```