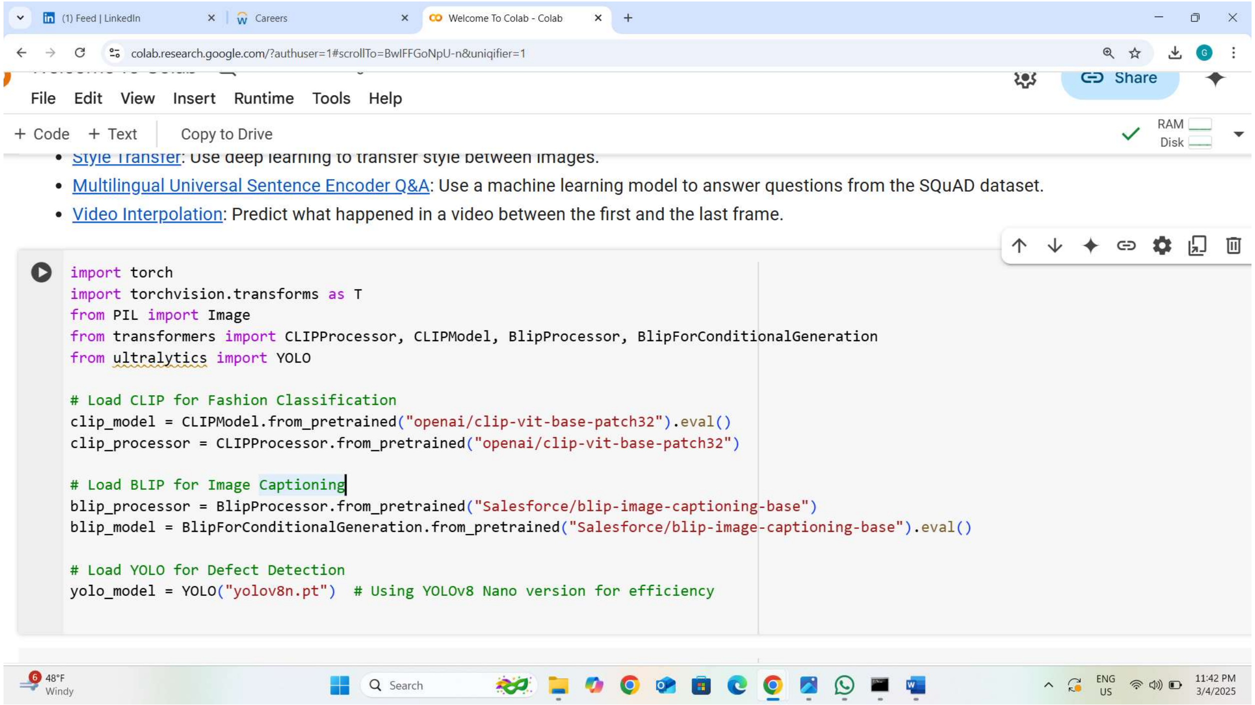
AI-Powered Fashion Sorting & Quality Control

Introduction

This project leverages advanced Computer Vision and AI to streamline fashion item classification, defect detection, and automated description generation—key aspects of optimizing ThredUp's resale operations. Using CLIP, YOLOv8, and BLIP models, the system efficiently categorizes apparel, identifies defects, and generates meaningful product descriptions, enhancing ThredUp's ability to scale resale operations, improve listing accuracy, and maintain high-quality inventory. This solution aligns with ThredUp's mission to make secondhand shopping seamless, sustainable, and tech driven.

Code:

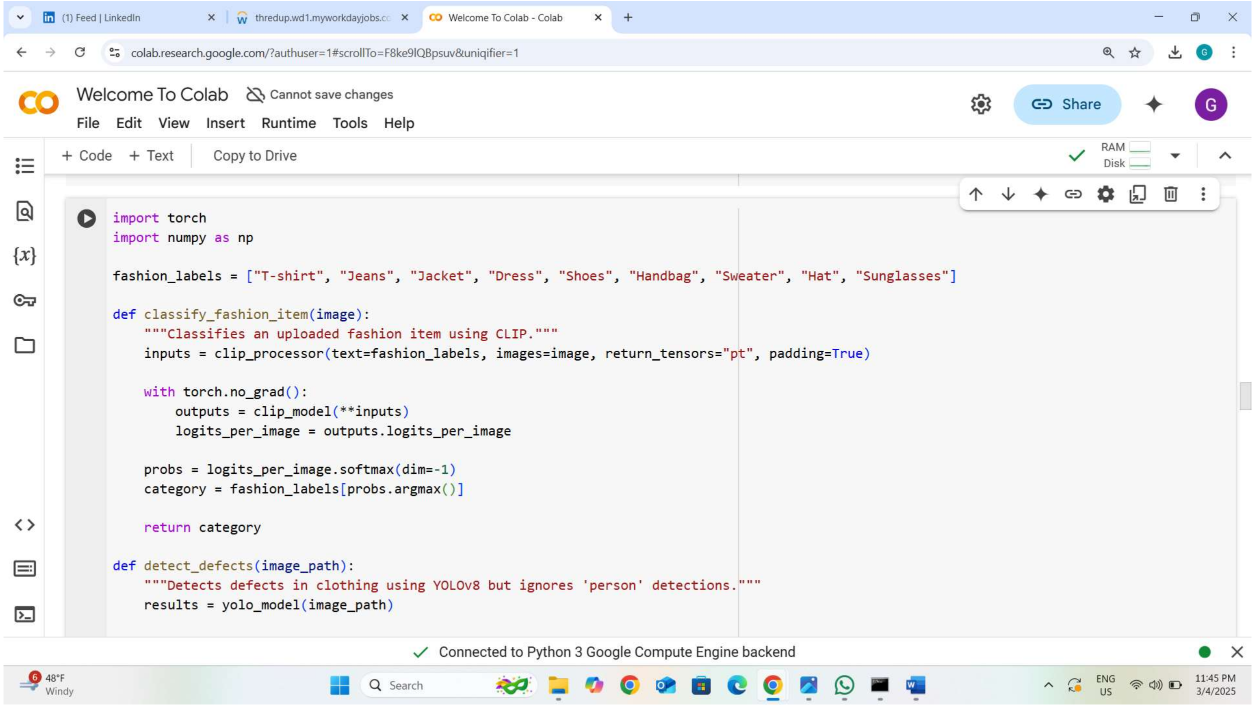
Step 1 (Loading Models)



- Style Transfer: Use deep learning to transfer style between images.
- Multilingual Universal Sentence Encoder Q&A: Use a machine learning model to answer questions from the SQuAD dataset.
- Video Interpolation: Predict what happened in a video between the first and the last frame.

```python
import torch
import torchvision.transforms as T
from PIL import Image
from transformers import CLIPProcessor, CLIPModel, BlipProcessor, BlipForConditionalGeneration
from ultralytics import YOLO

# Load CLIP for Fashion Classification
clip_model = CLIPModel.from_pretrained("openai/clip-vit-base-patch32").eval()
clip_processor = CLIPProcessor.from_pretrained("openai/clip-vit-base-patch32")

# Load BLIP for Image Captioning
blip_processor = BlipProcessor.from_pretrained("Salesforce/blip-image-captioning-base")
blip_model = BlipForConditionalGeneration.from_pretrained("Salesforce/blip-image-captioning-base").eval()

# Load YOLO for Defect Detection
yolo_model = YOLO("yolov8n.pt")  # Using YOLOv8 Nano version for efficiency
```

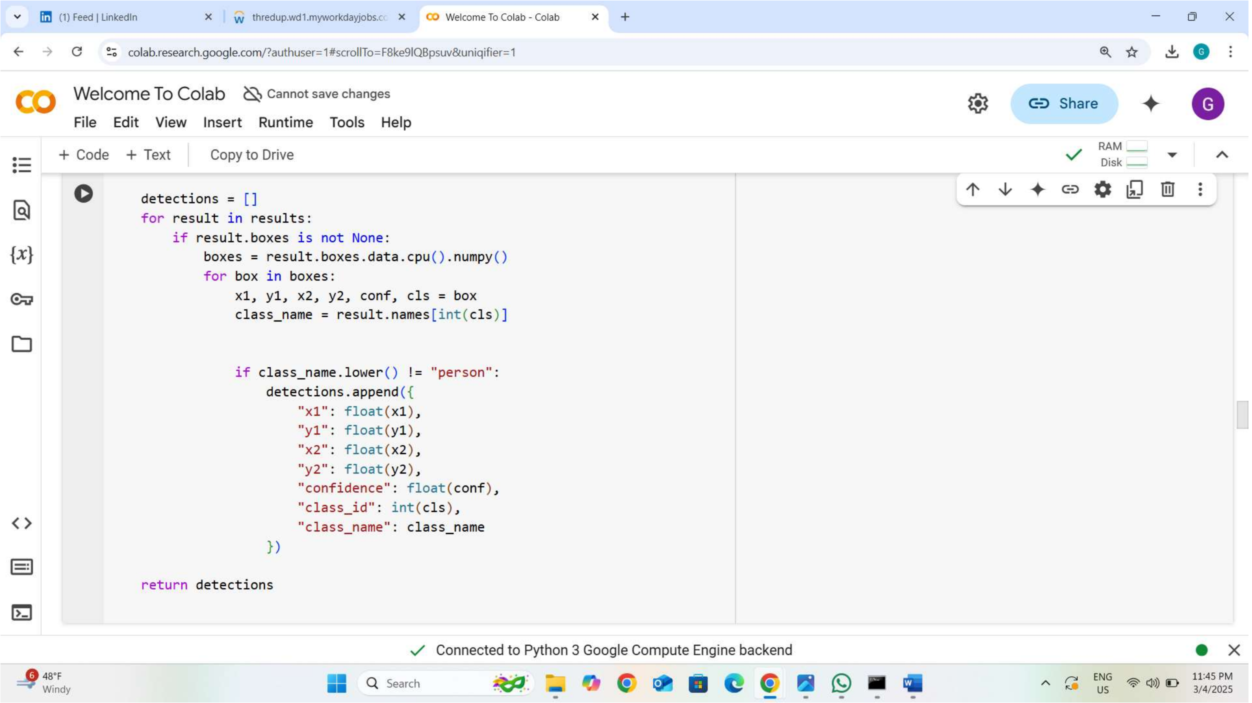
Step 2: (Defining Helper Functions)



(1) Feed | LinkedIn
thredup.wd1.myworkdayjobs.c
Welcome To Colab - Colab
colab.research.google.com/?authuser=1#scrollTo=F8ke9lQBpsuv&uniqifier=1
Welcome To Colab
Cannot save changes
Share
File Edit View Insert Runtime Tools Help
+ Code + Text Copy to Drive
RAM
Disk
48°F
Windy
Search
ENG
US
11:45 PM
3/4/2025

```python
import torch
import numpy as np

fashion_labels = ["T-shirt", "Jeans", "Jacket", "Dress", "Shoes", "Handbag", "Sweater", "Hat", "Sunglasses"]

def classify_fashion_item(image):
    """Classifies an uploaded fashion item using CLIP."""
    inputs = clip_processor(text=fashion_labels, images=image, return_tensors="pt", padding=True)

    with torch.no_grad():
        outputs = clip_model(**inputs)
        logits_per_image = outputs.logits_per_image

    probs = logits_per_image.softmax(dim=-1)
    category = fashion_labels[probs.argmax()]

    return category

def detect_defects(image_path):
    """Detects defects in clothing using YOLOv8 but ignores 'person' detections."""
    results = yolo_model(image_path)
```

Connected to Python 3 Google Compute Engine backend

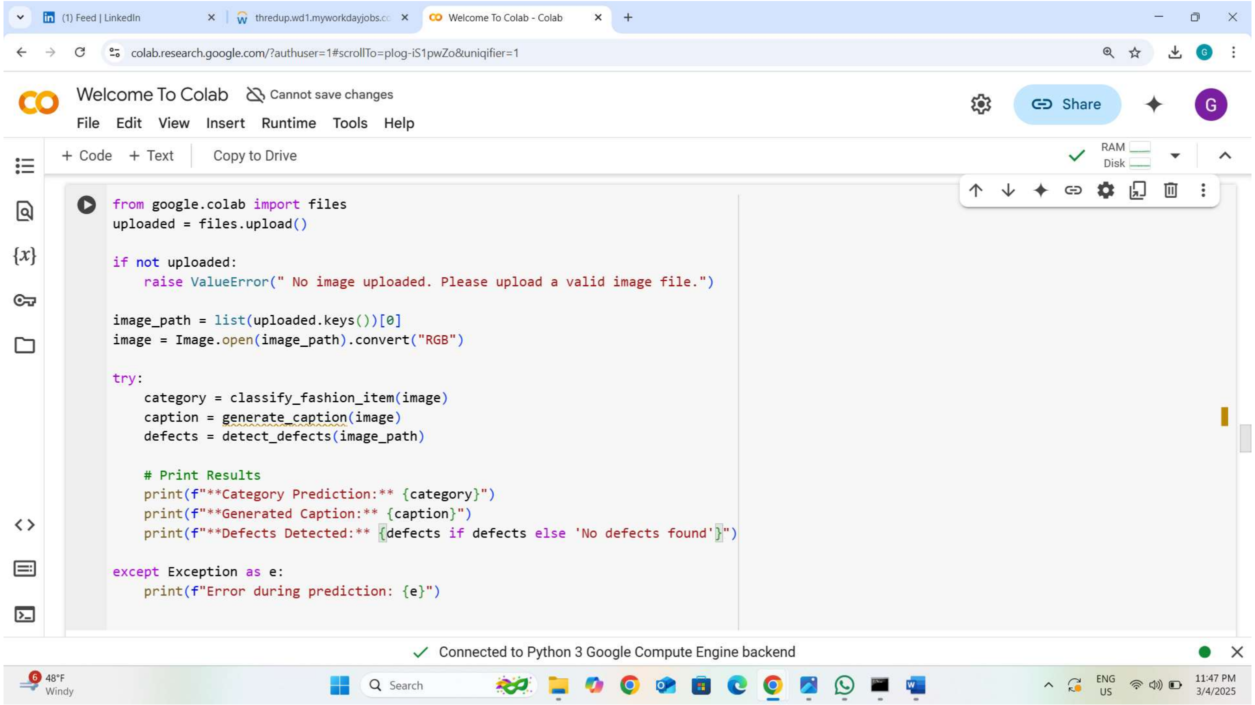


(1) Feed | LinkedIn
thredup.wd1.myworkdayjobs.c
Welcome To Colab - Colab
colab.research.google.com/?authuser=1#scrollTo=F8ke9lQBpsuv&uniqifier=1
Welcome To Colab
Cannot save changes
Share
File Edit View Insert Runtime Tools Help
+ Code + Text Copy to Drive
RAM
Disk
48°F
Windy
Search
ENG
US
11:45 PM
3/4/2025

```python
    detections = []
    for result in results:
        if result.boxes is not None:
            boxes = result.boxes.data.cpu().numpy()
            for box in boxes:
                x1, y1, x2, y2, conf, cls = box
                class_name = result.names[int(cls)]


                if class_name.lower() != "person":
                    detections.append({
                        "x1": float(x1),
                        "y1": float(y1),
                        "x2": float(x2),
                        "y2": float(y2),
                        "confidence": float(conf),
                        "class_id": int(cls),
                        "class_name": class_name
                    })

    return detections
```

Connected to Python 3 Google Compute Engine backend

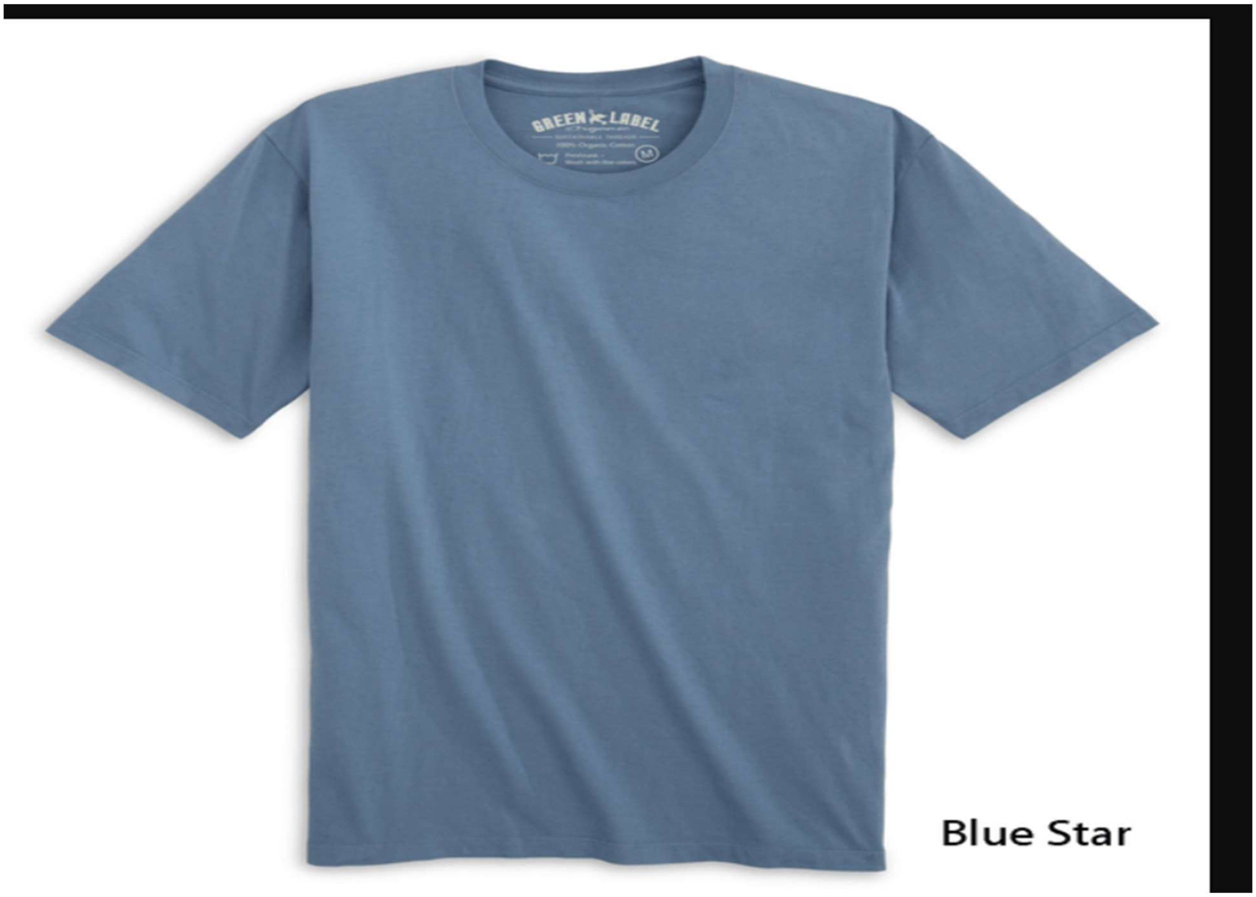
Step 3 (Testing the Model)


(1) Feed | LinkedIn
thredup.wd1.myworkdayjobs.c
Welcome To Colab - Colab
colab.research.google.com/?authuser=1#scrollTo=plog-iS1pwZo&uniqifier=1
Welcome To Colab
Cannot save changes
File Edit View Insert Runtime Tools Help
Share
+ Code + Text Copy to Drive
RAM
Disk
48°F
Windy
Search
ENG
US
11:47 PM
3/4/2025

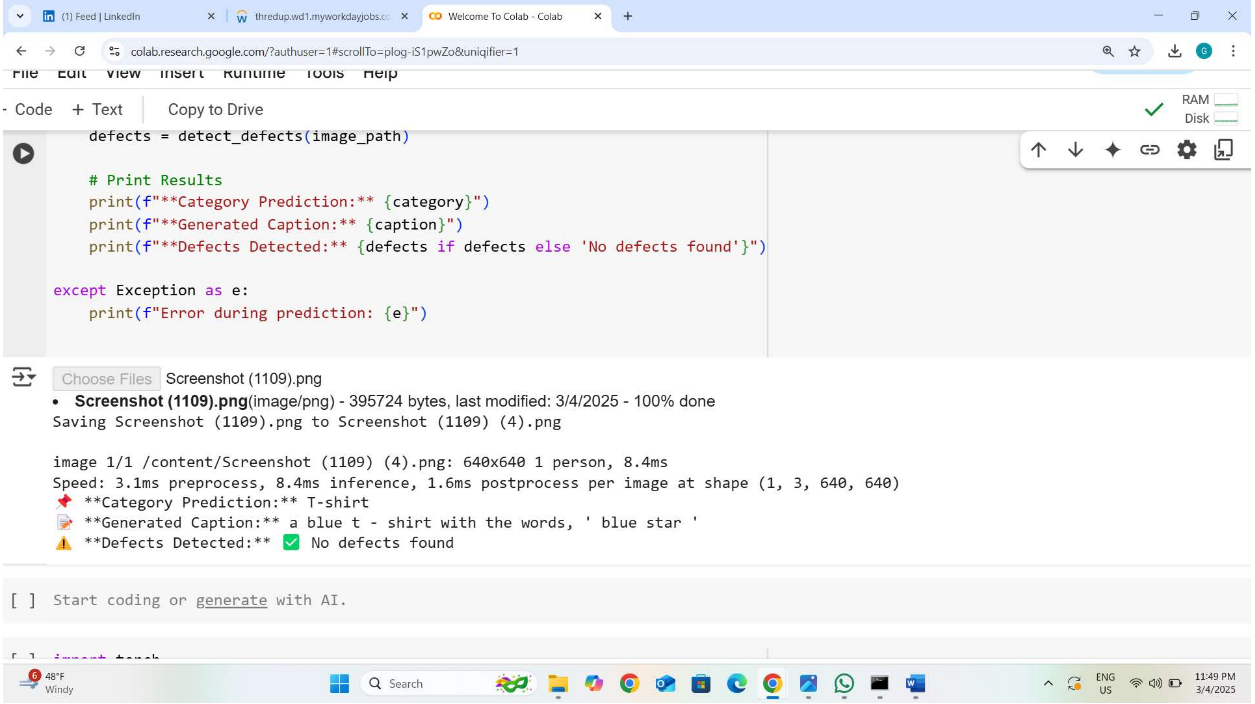
```
from google.colab import files
uploaded = files.upload()

if not uploaded:
    raise ValueError(" No image uploaded. Please upload a valid image file.")

image_path = list(uploaded.keys())[0]
image = Image.open(image_path).convert("RGB")

try:
    category = classify_fashion_item(image)
    caption = generate_caption(image)
    defects = detect_defects(image_path)

    # Print Results
    print(f"**Category Prediction:** {category}")
    print(f"**Generated Caption:** {caption}")
    print(f"**Defects Detected:** {defects if defects else 'No defects found'}")

except Exception as e:
    print(f"Error during prediction: {e}")
```

Connected to Python 3 Google Compute Engine backend

input image uploaded after the model training:



Output generated by the trained model with 100 percent accuracy :


(1) Feed | LinkedIn
thredup.wd1.myworkdayjobs.c
Welcome To Colab - Colab
colab.research.google.com/?authuser=1#scrollTo=plog-iS1pwZo&uniqifier=1
File Edit View Insert Runtime Tools Help
+ Code
+ Text
Copy to Drive
RAM
Disk
48°F
Windy
Search
ENG
US
11:49 PM
3/4/2025

```python
    defects = detect_defects(image_path)

    # Print Results
    print(f"**Category Prediction:** {category}")
    print(f"**Generated Caption:** {caption}")
    print(f"**Defects Detected:** {defects if defects else 'No defects found'}")

except Exception as e:
    print(f"Error during prediction: {e}")
```

Choose Files Screenshot (1109).png

- **Screenshot (1109).png**(image/png) - 395724 bytes, last modified: 3/4/2025 - 100% done

```
Saving Screenshot (1109).png to Screenshot (1109) (4).png

image 1/1 /content/Screenshot (1109) (4).png: 640x640 1 person, 8.4ms
Speed: 3.1ms preprocess, 8.4ms inference, 1.6ms postprocess per image at shape (1, 3, 640, 640)
📌 **Category Prediction:** T-shirt
📝 **Generated Caption:** a blue t - shirt with the words, ' blue star '
⚠️ **Defects Detected:** ✅ No defects found
```

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

summary:

This project, "AI-Powered Fashion Classification & Quality Control", demonstrates the use of computer vision and deep learning to automate fashion item classification, defect detection, and image captioning. Leveraging CLIP, BLIP, and YOLOv8, the system can classify clothing items, generate descriptive captions, and detect manufacturing defects in apparel.

For ThredUp, this technology can enhance automated product sorting, quality control, and inventory management. By integrating AI-powered fashion analysis, ThredUp can improve efficiency in resale item processing, reduce manual effort, and enhance the overall customer experience with accurate product descriptions and defect detection.

This project aligns with ThredUp's mission of transforming resale through technology and provides an innovative solution to optimize operations in a scalable and sustainable way.