**Advanced Python – Lab**

Suvash Sharma Subedi

22BCP116

LAB Assignment 1: Product Review from text file

Consider a scenario where you are working as a data scientist for a large e-commerce company.

Your team is responsible for analyzing customer feedback data, which is stored in multiple text

files. Each text file contains customer reviews for different product categories. Your task is to

write a Python script that performs the following operations:

Read the contents of all the text files in a given directory.

For each review, extract the following information:

•Customer ID (a 6-digit alphanumeric code)

•Product ID (a 10-digit alphanumeric code)

•Review date (in the format "YYYY-MM-DD")

•Review rating (an integer between 1 and 5)

•Review text (the actual feedback provided by the customer)

Calculate the average review rating for each product and store it in a dictionary where the

product ID is the key and the average rating is the value.

Determine the top 3 products with the highest average review ratings.

Create a new text file named "summary.txt" and write the following information into it:

•The total number of reviews processed.

•The total number of valid reviews (reviews with all required information extracted

successfully).

•The total number of invalid reviews (reviews with missing or incorrect information).

•The product ID and average rating of the top 3 products with the highest average ratings.

Your Python script should be robust, handling any potential errors or exceptions during the file

handling process.

Additionally, you should implement efficient algorithms to handle large volumes of data without

consuming excessive memory or processing time.

Write the Python script to achieve the above objectives and provide detailed comments

explaining each step of your implementation.

import os

from collections import defaultdict

class Review:

def \_\_init\_\_(self, productId, cId, date, rating, text):

self.productId = productId

self.cId = cId

self.date = date

self.rating = rating

self.text = text

invalid = 0

valid = 0

# Sets and lists to store the required data

product\_ids = set()

customer\_ids = set()

review\_dates = set()

ratings = []

messages = []

def read\_reviews\_from\_file(filename):

global invalid, valid

reviews = []

with open(filename, 'r') as file:

for line in file:

data = line.strip().split(maxsplit=4)

if len(data) == 5:

productId, cId, date, rating, text = data

try:

rating = int(rating)

reviews.append(Review(productId, cId, date, rating, text))

# Collecting data

product\_ids.add(productId)

customer\_ids.add(cId)

review\_dates.add(date)

ratings.append(rating)

messages.append(text)

valid += 1

except ValueError:

invalid += 1

else:

invalid += 1

return reviews

def calculate\_average(reviews):

product\_ratings = defaultdict(list)

for review in reviews:

product\_ratings[review.productId].append(review.rating)

average\_ratings = {}

for productId, ratings in product\_ratings.items():

average\_ratings[productId] = sum(ratings) / len(ratings) if ratings else 0.0

return average\_ratings

folder\_path = '/home/suvash/Desktop/python/reviews'

all\_reviews = []

if os.path.isdir(folder\_path):

for filename in os.listdir(folder\_path):

file\_path = os.path.join(folder\_path, filename)

if os.path.isfile(file\_path) and filename.endswith('.txt'):

all\_reviews.extend(read\_reviews\_from\_file(file\_path))

else:

print(f"Error: {folder\_path} is not a valid directory.")

average\_ratings = calculate\_average(all\_reviews)

sorted\_average\_ratings = sorted(average\_ratings.items(), key=lambda x: x[1], reverse=True)

# Print the average ratings for all products

for productId, avg\_rating in average\_ratings.items():

print(f"Product ID: {productId}, Average Rating: {avg\_rating:.2f}")

# Print the top three products based on average ratings

print("\nTop 3 Products Based on Average Ratings:")

for productId, avg\_rating in sorted\_average\_ratings[:3]:

print(f"Product ID: {productId}, Average Rating: {avg\_rating:.2f}")

# Print the collected data

print(f"\nValid reviews: {valid}, Invalid reviews: {invalid}")

print()

print()

print(f"All product IDs: {product\_ids}")

print()

print(f"All customer IDs: {customer\_ids}")

print()

print(f"All review dates: {review\_dates}")

print()

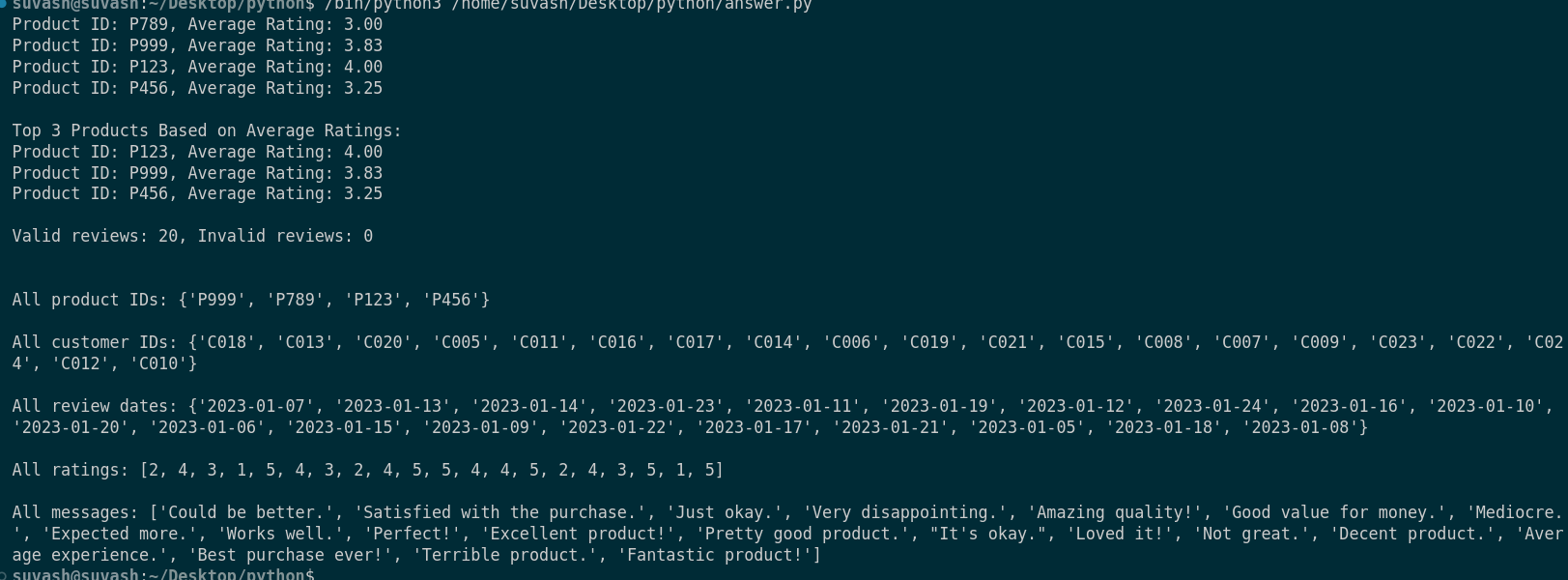
print(f"All ratings: {ratings}")

print()

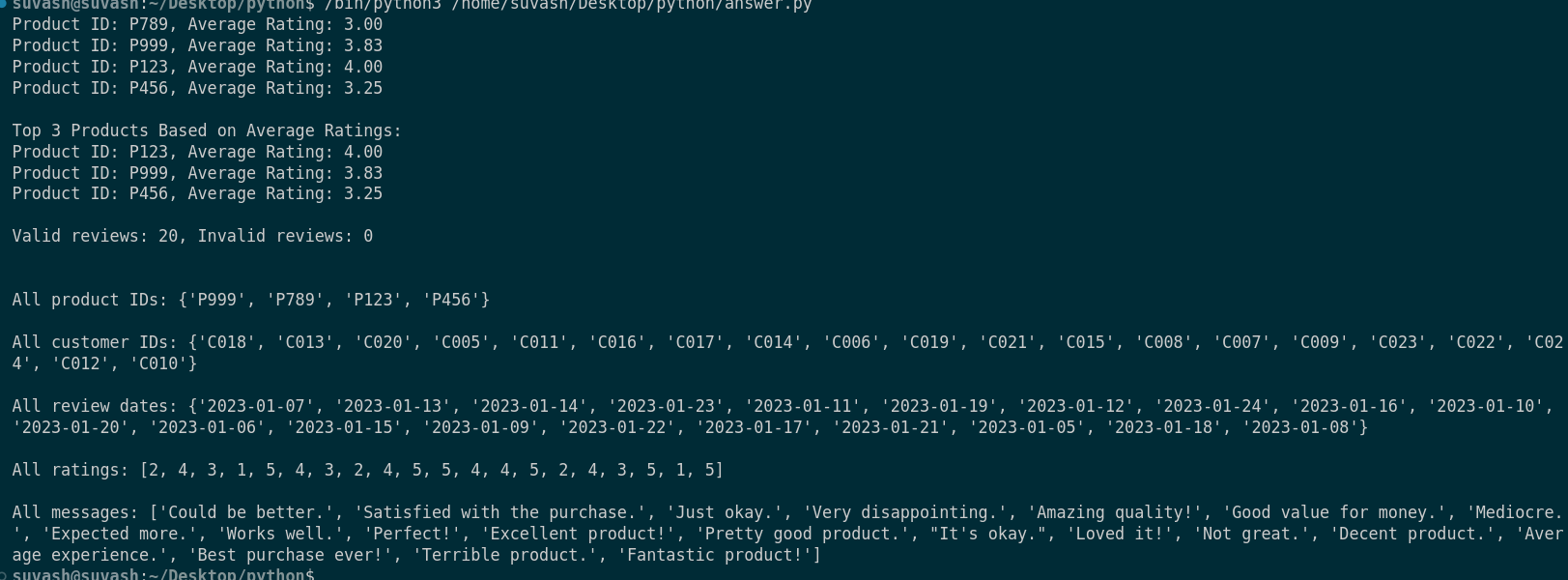
print(f"All messages: {messages}")

**Output:**

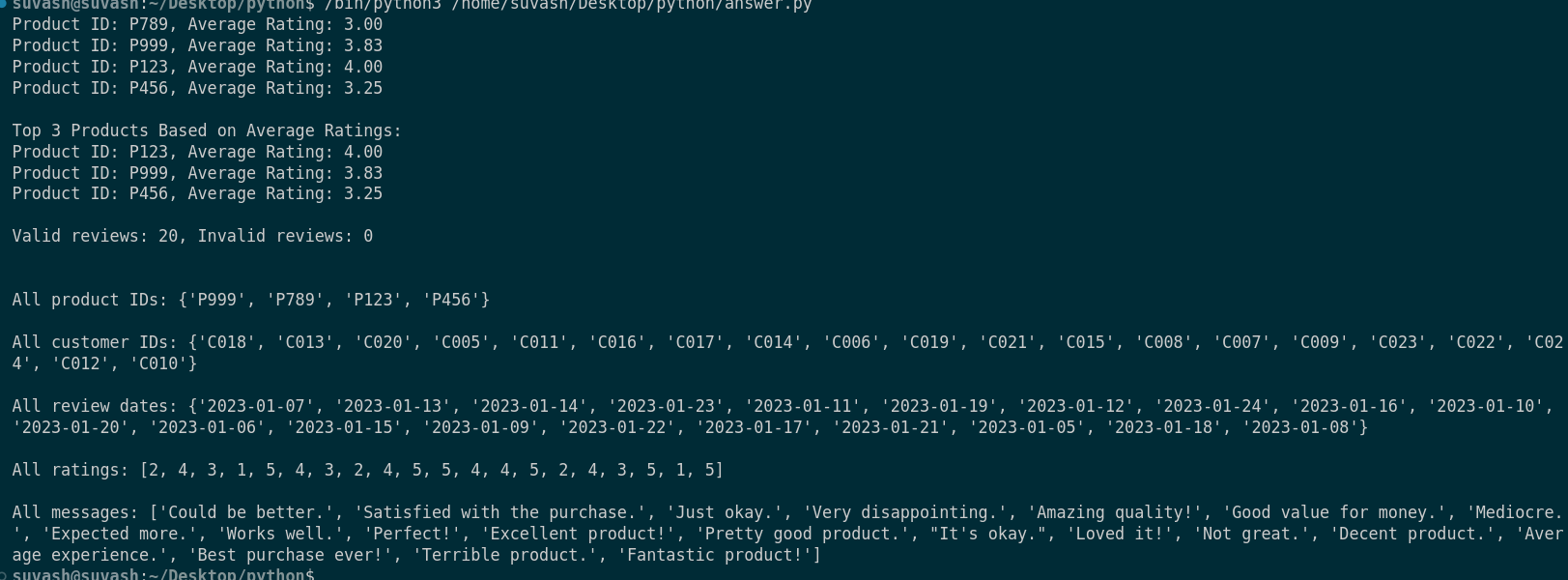
All product Ids along with their ratings:



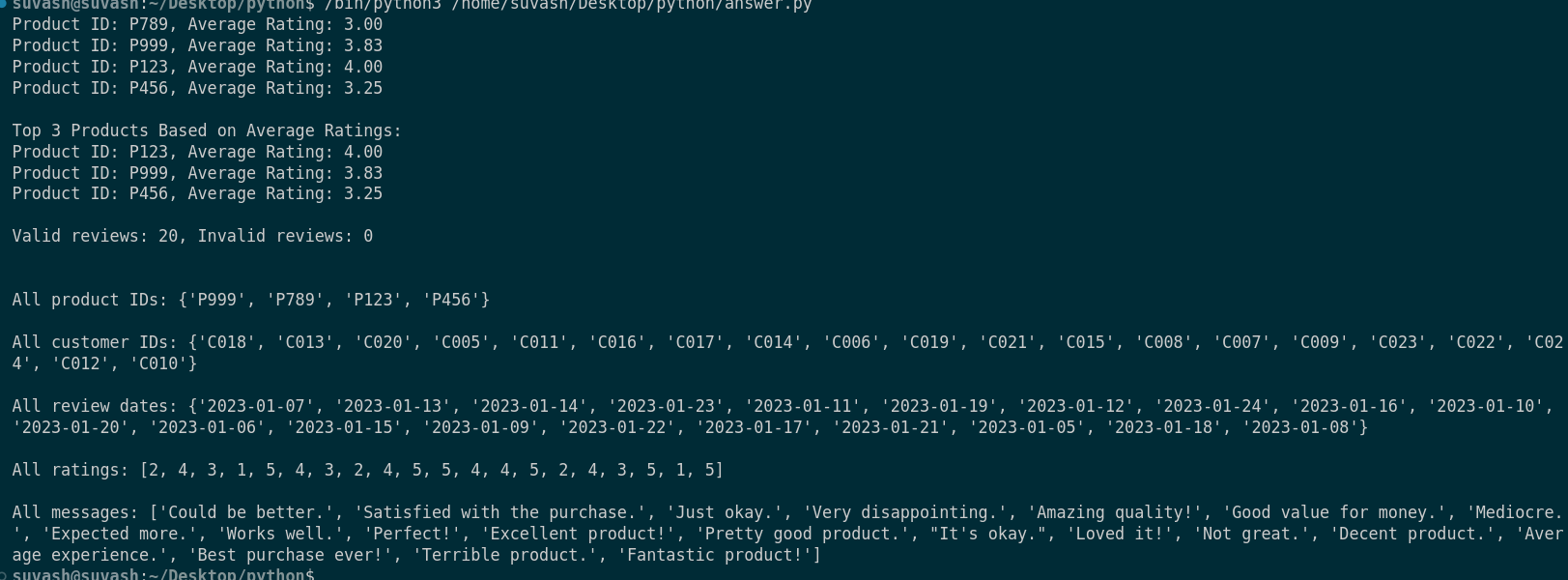
Top 3 products based on their average ratings:



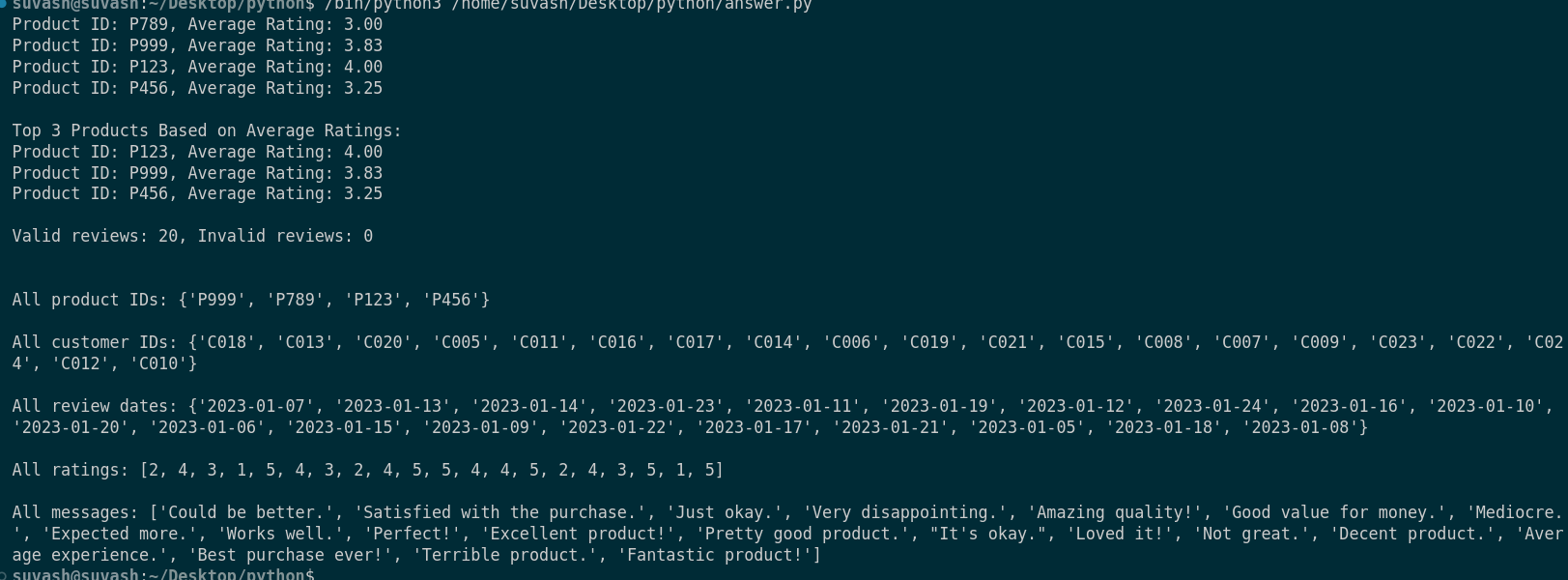
Valid reviews and invalid reviews count:

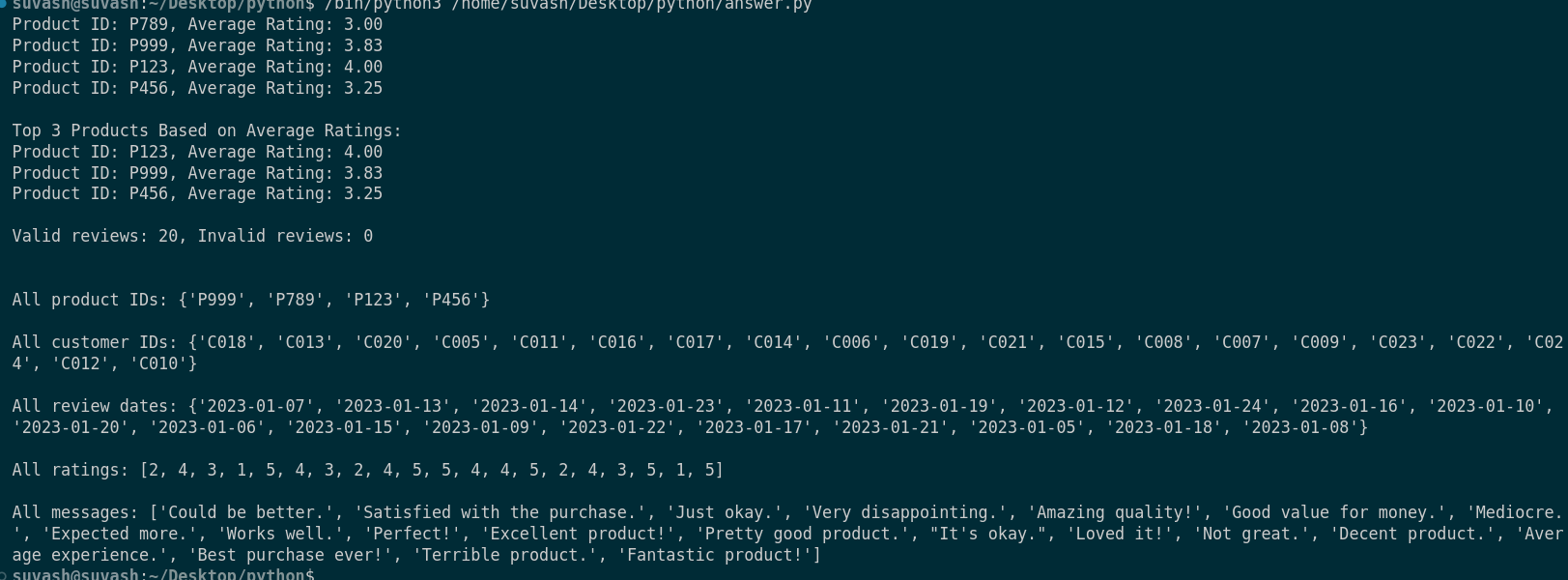


All Product Ids:

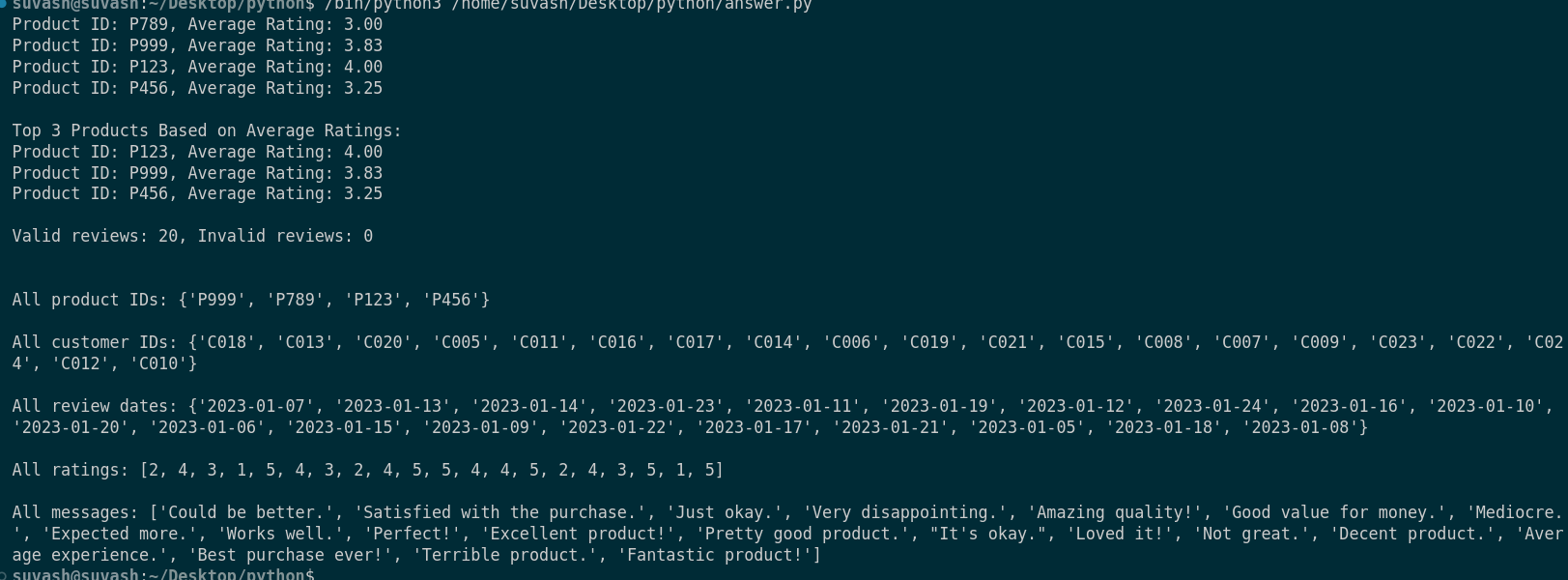


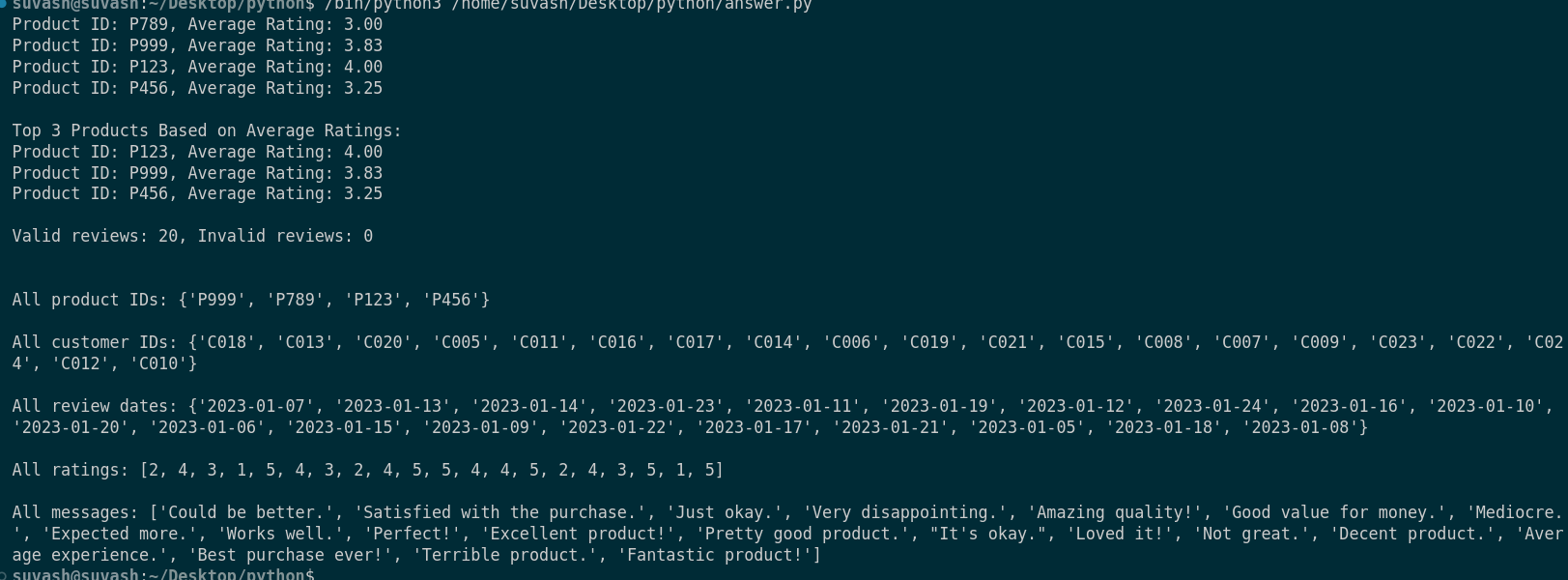
All Customer Ids:



All Reviews dates:

All ratings:



All review messages:

**Observations:**

We can use python’s file handling techniques to read from, write to and modify the contents of files.