**ML ASSIGNMENT 1**

**NAME : Nishit Reddy Lingala**

**Id:** 700747128

**Git hub:**  <https://github.com/Nishitreddy/machine_learning.git>

**Video link:** [Assignment1.mp4](https://drive.google.com/file/d/1J9IrEXZaLR2mQxTgEys6ZCefAVLwyAZn/view?usp=share_link)

# Question 1

ages = [19, 22, 19, 24, 20, 25, 26, 24, 25, 24]

ages.sort() # Sort the list

# Find the min and max age

minimum\_age = min(ages)

maximum\_age = max(ages)

# Add the min age and the max age again to the list

ages.append(minimum\_age)

ages.append(maximum\_age)

# Find the median age

if len(ages) % 2 == 0: #check if there are 2 medians

med = (ages[len(ages)//2] + ages[len(ages)//2 - 1])/2

else: #else 1 median

med = ages[len(ages)//2]

# Find the average age

average = sum(ages) / len(ages)

# Find the range of ages

range = maximum\_age - minimum\_age

# Print the results

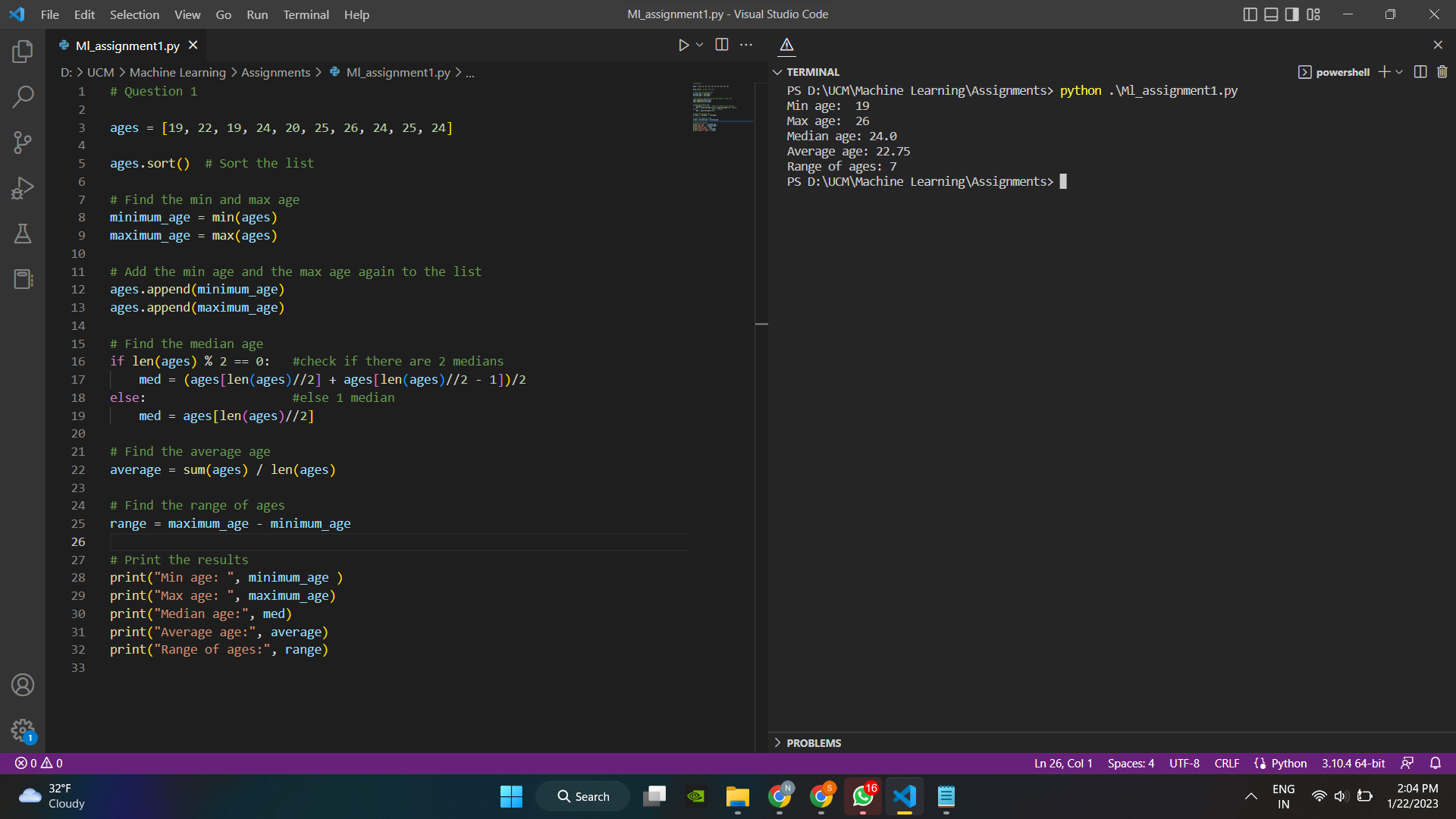
print("Min age: ", minimum\_age )

print("Max age: ", maximum\_age)

print("Median age:", med)

print("Average age:", average)

print("Range of ages:", range)



Explanation:

As the supplied query sorted the list of ages first, it then used min() and max() to find the least and maximum ages and imported those values into the ages list. We then found the median, average and the age. We have also checked if the number of elements are even or odd accordingly and found the median.

# Question 2

# Create an empty dictionary called dog

dog = {}

# Add name, color, breed, legs, age to the dog dictionary

dog['name'] = 'lucky'

dog['color'] = 'white'

dog['breed'] = 'Retriever'

dog['legs'] = 4

dog['age'] = 1

# Create a student dictionary

# add first\_name, last\_name, gender, age, marital status,

# skills, country, city and address as keys for the dictionary

student = {

'first\_name': 'Jesse',

'last\_name': 'Pinkman',

'gender': 'male',

'age': 28,

'marital\_status': 'single',

'skills': ['Python', 'JavaScript'],

'country': 'USA',

'city': 'Albequrqeu',

'address': '55th St'

}

# Get the length of the student dictionary

length = len(student)

# Get the value of skills and check the data type

skills = student['skills']

print(type(skills)) #type returns the datatype

# Change the skills values by adding one or two skills

student['skills'].append('CSS')

student['skills'].append('Node')

# Get the dictionary keys as a list

student\_keys = list(student.keys())

# Get the dictionary values as a list

student\_values = list(student.values())

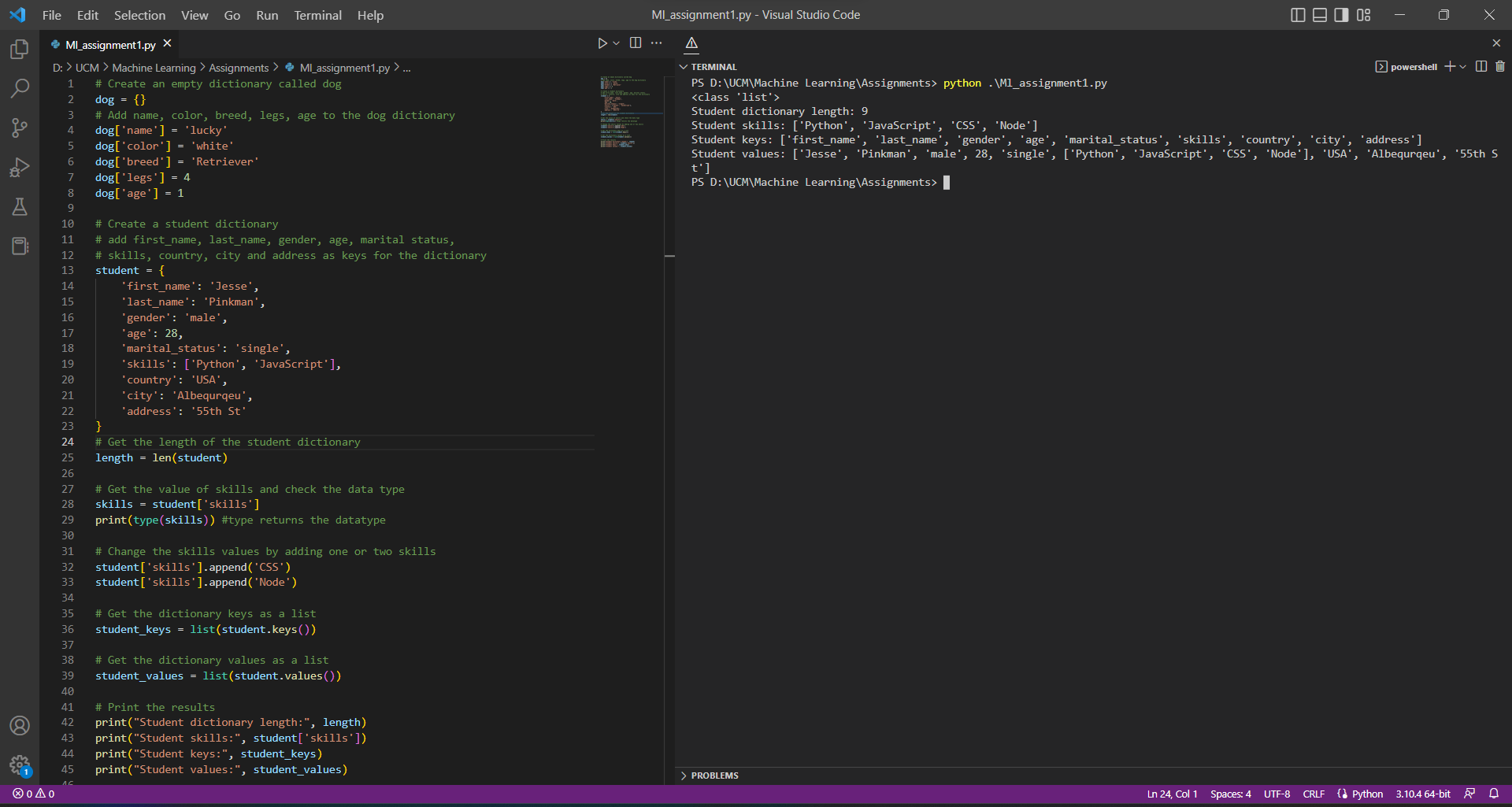
# Print the results

print("Student dictionary length:", length)

print("Student skills:", student['skills'])

print("Student keys:", student\_keys)

print("Student values:", student\_values)



Explanation:

Construct a dictionary dog and then add the legs, age, breed, color, and name. Create another dictionary named student next, add first name, last name, gender, age, marital status, skills, nation, city, and address to the dictionary you created. Calculate the length of the dictionary using len() after adding all characteristics.

# Question 3

# Create a tuple containing names of your sisters and your brothers

sisters = ("siri", "chandana", "sai")

brothers = ("mike", "rahool", "shhivaa")

# Join brothers and sisters assign it to siblings

siblings = sisters + brothers

# Number of siblings you have?

num = len(siblings)

#siblings tuple and add the name of your father and mother and assign it to familyMembers

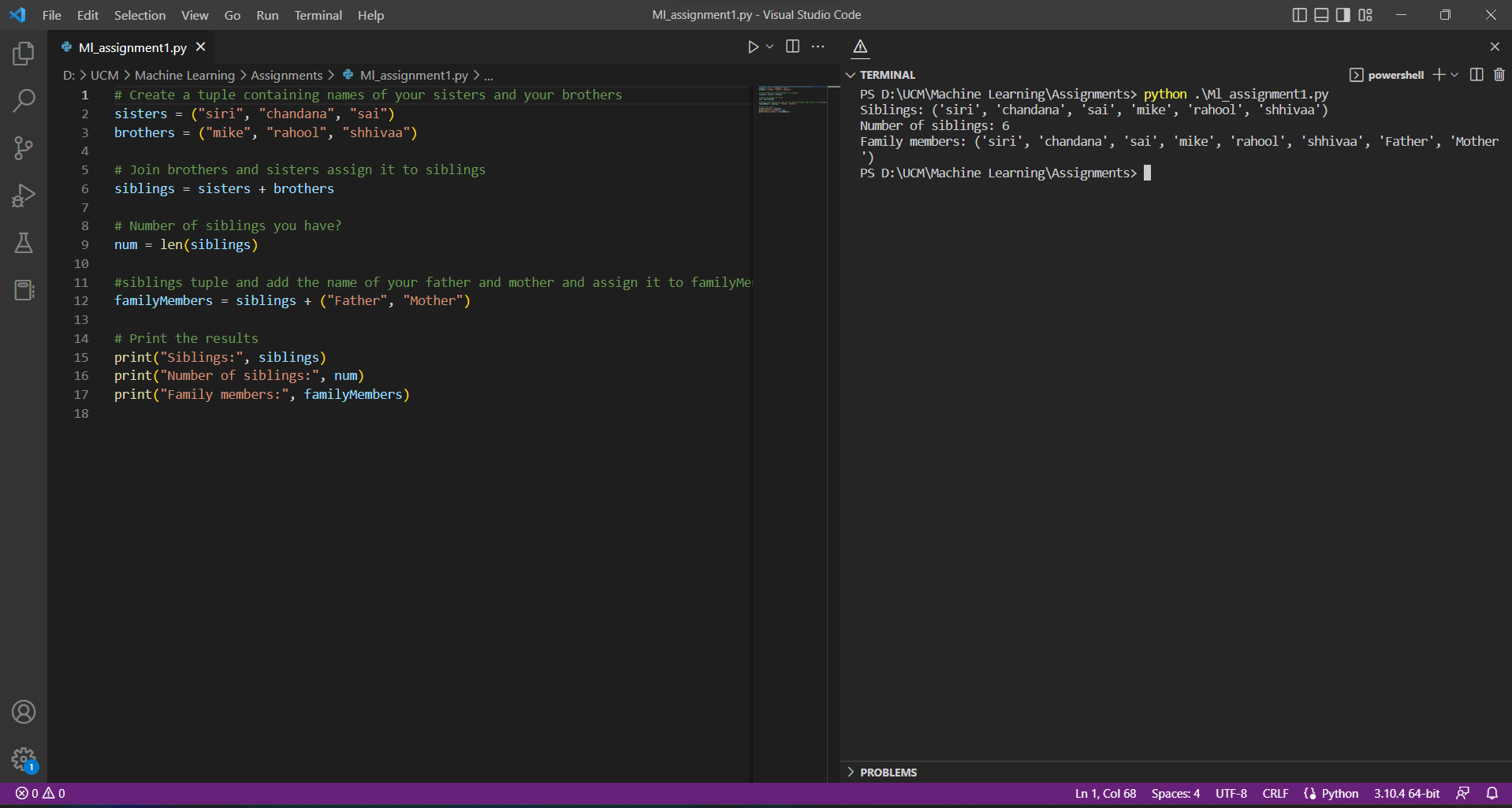
familyMembers = siblings + ("Father", "Mother")

# Print the results

print("Siblings:", siblings)

print("Number of siblings:", num)

print("Family members:", familyMembers)



Explanation:

Create two tuples at first, one for each sibling, then unite them both and assign the items to the siblings. After determining the number of siblings, add the names of the parents to the siblings, and then output the tuple of siblings.

# Question 4

it\_companies = {'Facebook', 'Google', 'Microsoft', 'Apple', 'IBM', 'Oracle', 'Amazon'}

A = {19, 22, 24, 20, 25, 26}

B = {19, 22, 20, 25, 26, 24, 28, 27}

age = [22, 19, 24, 25, 26, 24, 25, 24]

# Find the length of the set it\_companies

length\_it\_companies = len(it\_companies)

print("Length of IT companies set:", length\_it\_companies)

# Add 'Twitter' to it\_companies

it\_companies.add('Twitter')

print("After adding Twitter:", it\_companies)

# Insert multiple IT companies at once to the set it\_companies

it\_companies.update(['wipro','TCS','Infosys'])

print("After adding multiple companies:", it\_companies)

# Remove one of the companies from the set it\_companies

it\_companies.remove('wipro')

print("IT companies set after removing Uber:", it\_companies)

it\_companies.discard('TCS')

print("IT companies set after discarding Tesla:", it\_companies)

# Diff between remove and discard

# 'remove' raises an error if the item is not found in the set, 'discard' does not

# Join A and B

AunionB = A.union(B)

print("A union B:", AunionB)

# Find A intersection B

AintersectionB = A.intersection(B)

print("A intersection B:", AintersectionB)

# Is A subset of B

isAsubsetofB = A.issubset(B)

print("Is A subset of B:", isAsubsetofB)

# Are A and B disjoint sets

isABdisjoint = A.isdisjoint(B)

print("Are A and B disjoint sets:", isABdisjoint)

# Join A with B and B with A

ABjoin = A.symmetric\_difference(B)

print("Symmetric difference between A and B:", ABjoin)

# Delete the sets completely

del A

del B

# Convert the ages to a set and compare the length of the list and the set

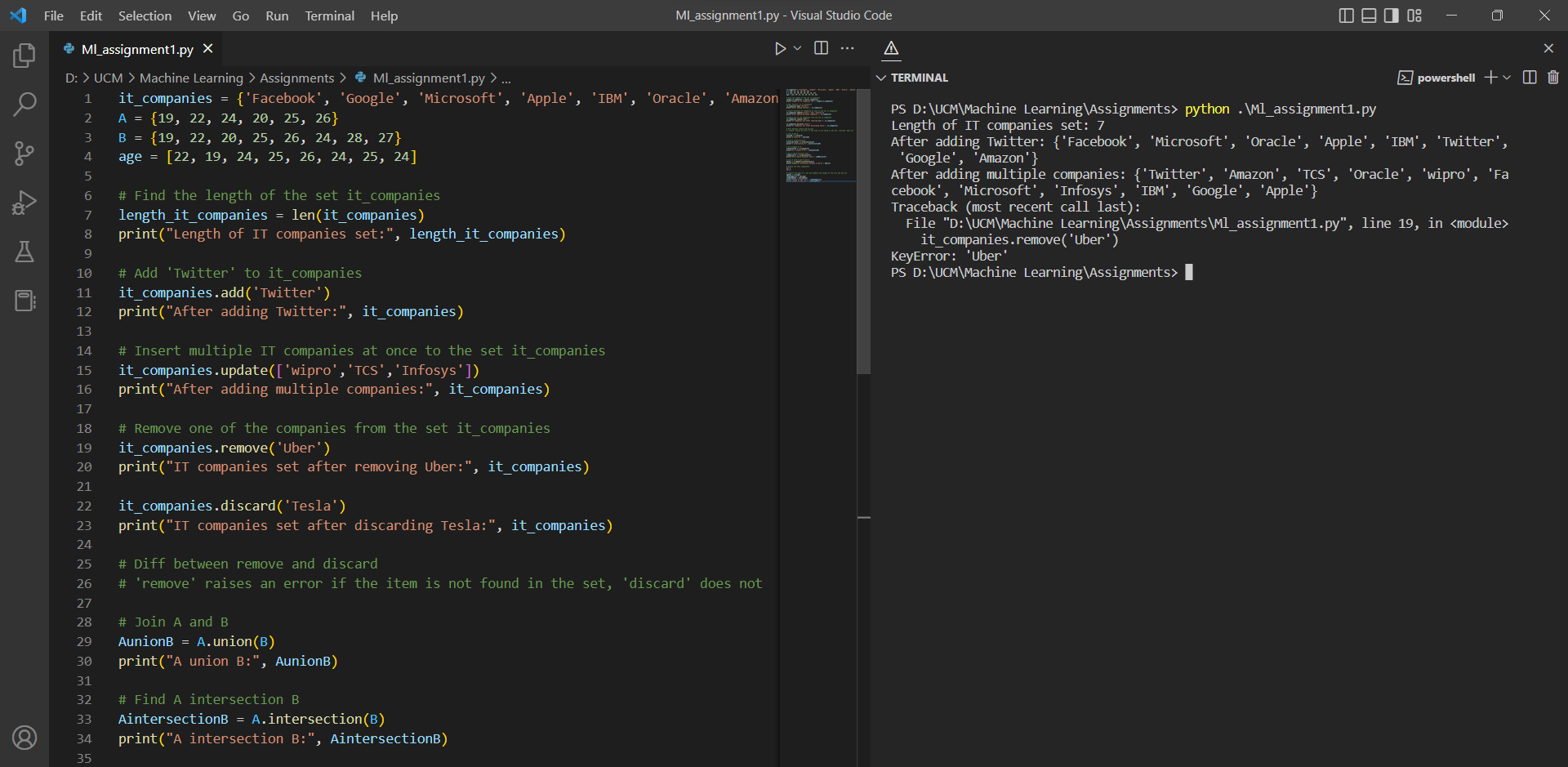
ageSet = set(age)

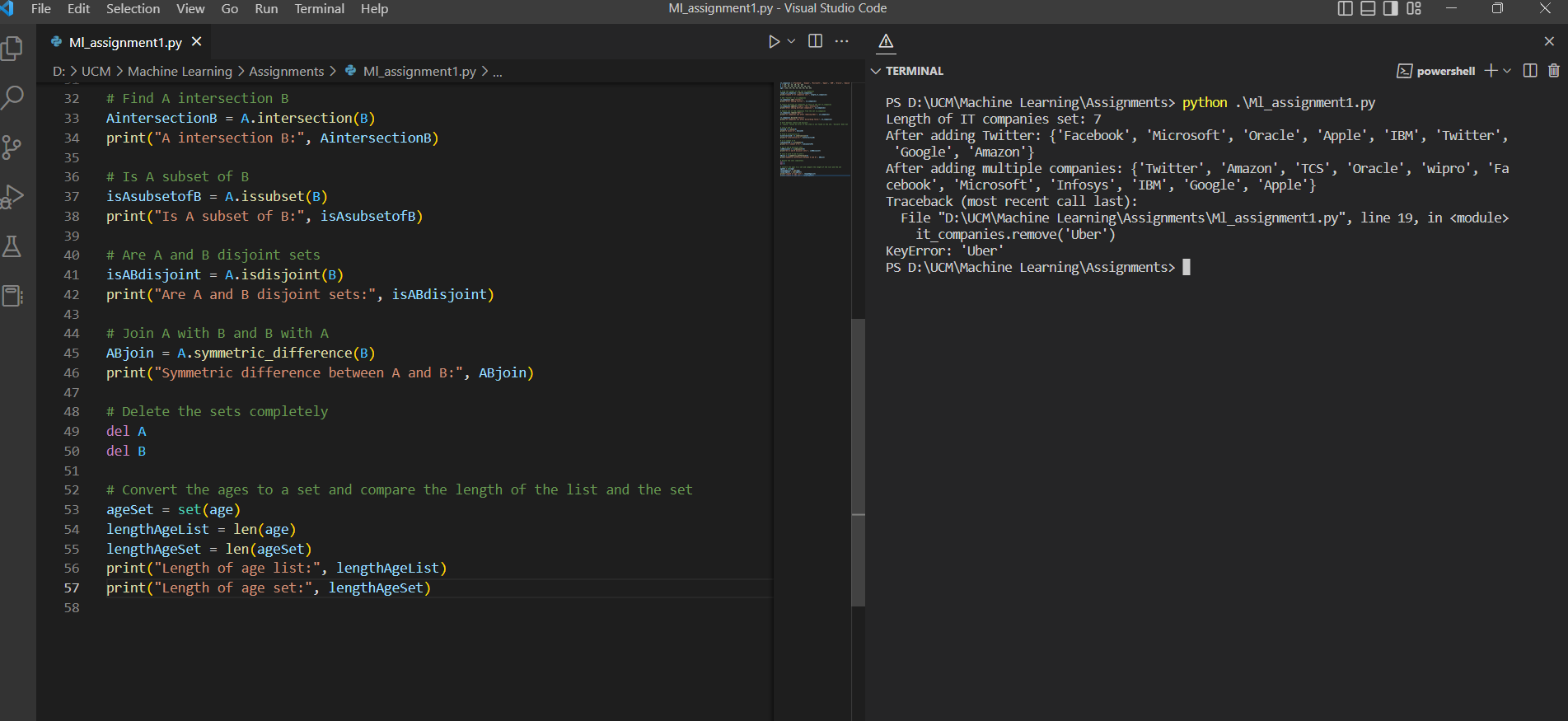
lengthAgeList = len(age)

lengthAgeSet = len(ageSet)

print("Length of age list:", lengthAgeList)

print("Length of age set:", lengthAgeSet)





# Question 5

import math

# The radius of a circle is 30 meters.

radius = 30

# Calculate the area of a circle

areaOfCircle = math.pi \* (radius \*\* 2)

# Calculate the circumference of a circle

circumOfCircle = 2 \* math.pi \* radius

# Print the results

print("Area of circle:", areaOfCircle)

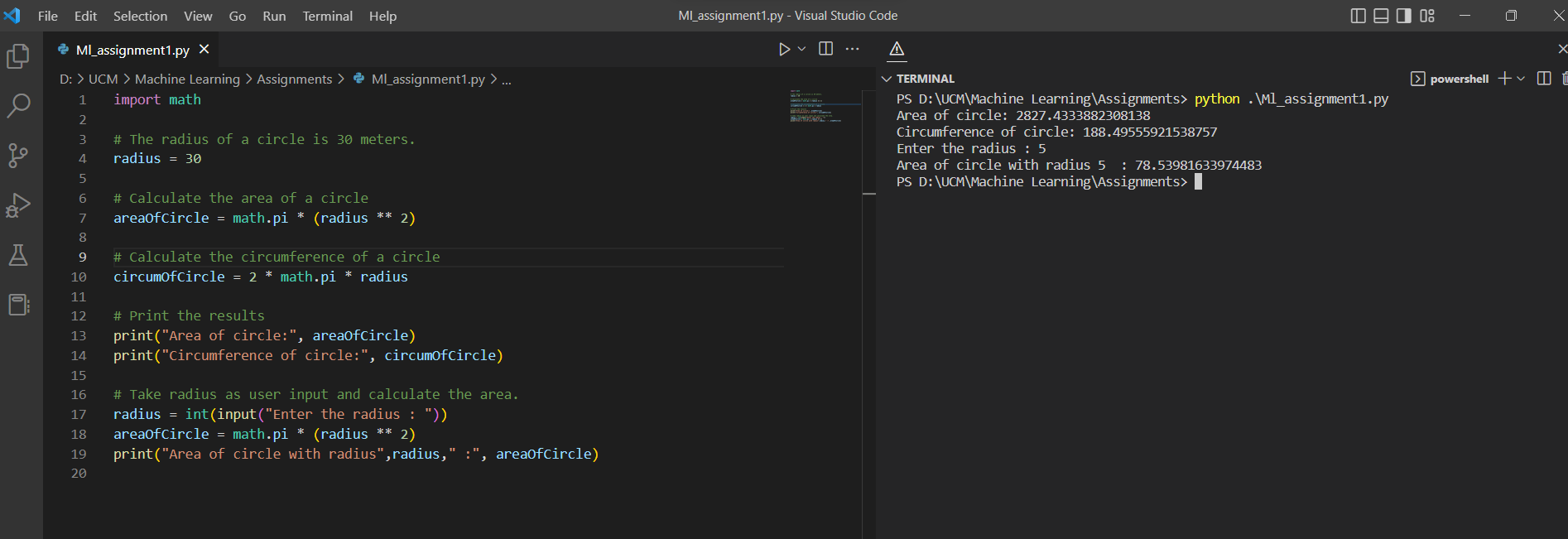
print("Circumference of circle:", circumOfCircle)

# Take radius as user input and calculate the area.

radius = int(input("Enter the radius : "))

areaOfCircle = math.pi \* (radius \*\* 2)

print("Area of circle with radius",radius," :", areaOfCircle)



Explanation:

Determine the circle's area and assign a value to areaOfCircle. Next, determine the circle's circumference and assign a value to circumOfCircle. Now compute the area using the radius as an input.

We here import math class to retrieve the value of pi using math.pi method.

# Question 6.

sentence = "I am a teacher and I love to inspire and teach people"

# Split the sentence into words

words = sentence.split()

# Use set to get the unique words

unique\_words = set(words)

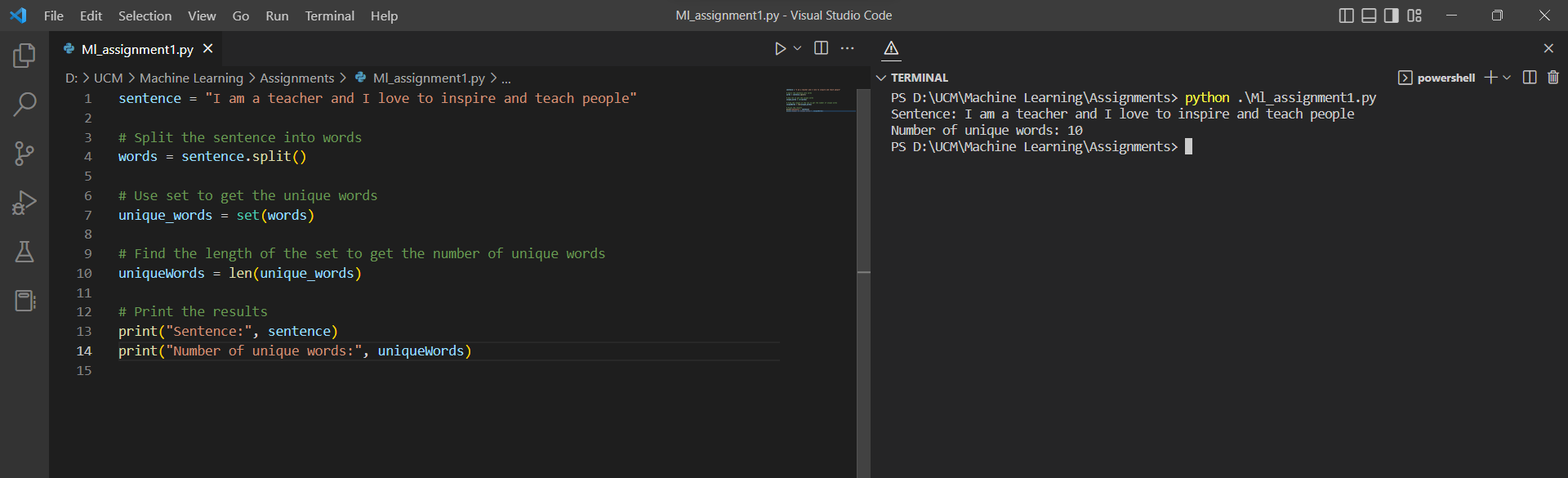
# Find the length of the set to get the number of unique words

uniqueWords = len(unique\_words)

# Print the results

print("Sentence:", sentence)

print("Number of unique words:", uniqueWords)



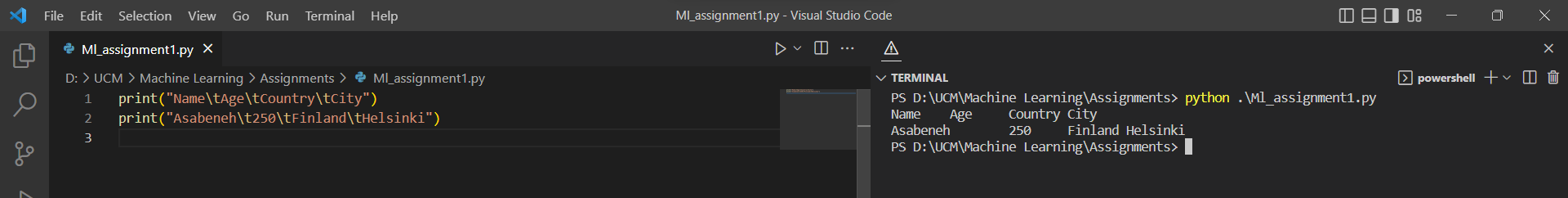
Explanation:

In the given sentence, we must first use split() to determine the number of distinct words, and then len() to determine the length of distinct words.

# Question 7.

print("Name\tAge\tCountry\tCity")

print("Asabeneh\t250\tFinland\tHelsinki")



Explanation:

We must use "t" to create a tab space between the provided words.

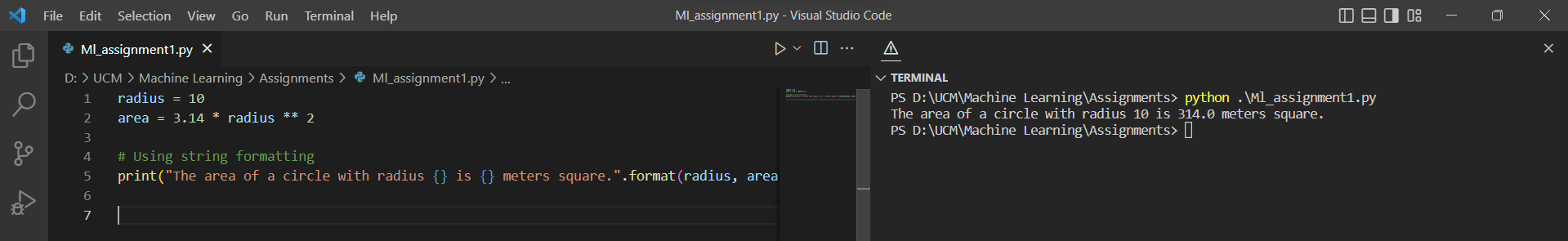
# Question8

radius = 10

area = 3.14 \* radius \*\* 2

# Using string formatting

print("The area of a circle with radius {} is {} meters square.".format(radius, area))



Explanation:

Applying the string formatting method, calculate the area of a circle and consider radius() and area ().

# Question 9

# Read the number of students

num\_students = int(input("Enter the number of students: "))

# Initialize an empty list to store the weights in pounds

weightsLbs = []

# Read the weights of the students in pounds

for i in range(num\_students):

weight = float(input(f"Enter the weight of student {i+1} in pounds: "))

weightsLbs.append(weight)

# Initialize an empty list to store the weights in kilograms

weightsKgs = []

# Convert the weights from pounds to kilograms

for weight in weightsLbs:

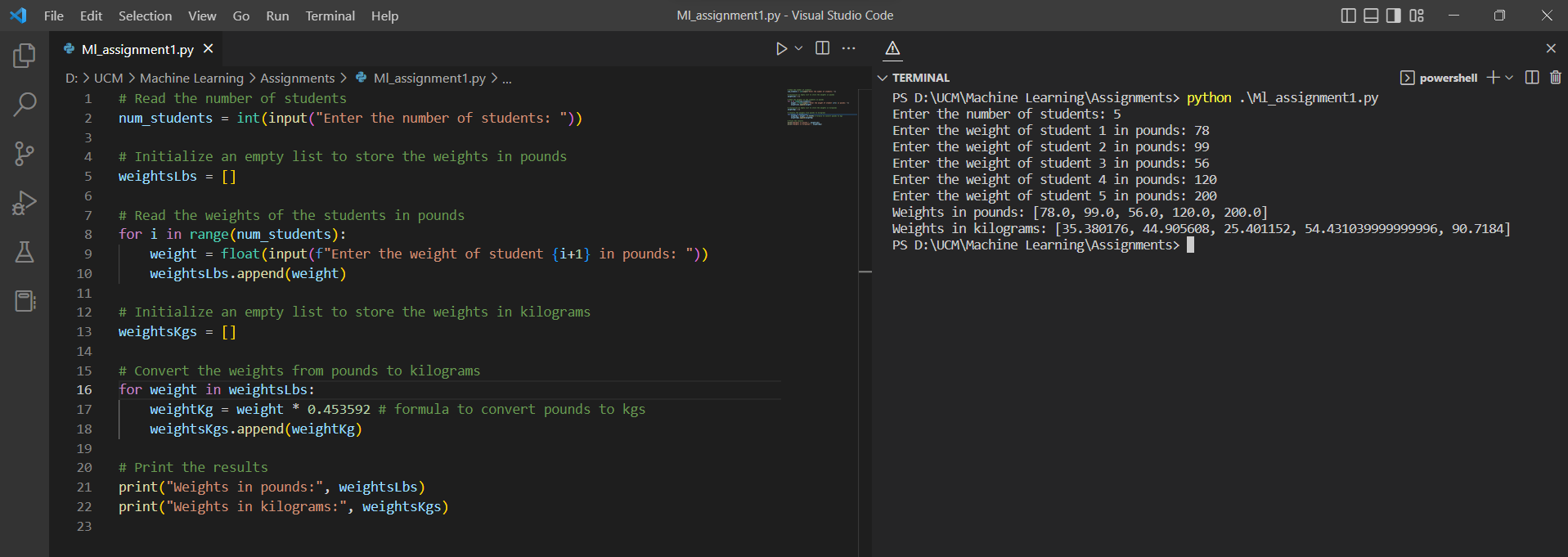
weightKg = weight \* 0.453592 # formula to convert pounds to kgs

weightsKgs.append(weightKg)

# Print the results

print("Weights in pounds:", weightsLbs)

print("Weights in kilograms:", weightsKgs)



Explanation:

We take input from users asking for the number of students and weight of each corresponding student in lbs and we store it in a list. We declare the empty weightKgs list and then in the range of the previous list we use the formula (weight \* 0.453592) to convert lbs into kgs and store the values in the weightKgs list.

#Question 10

