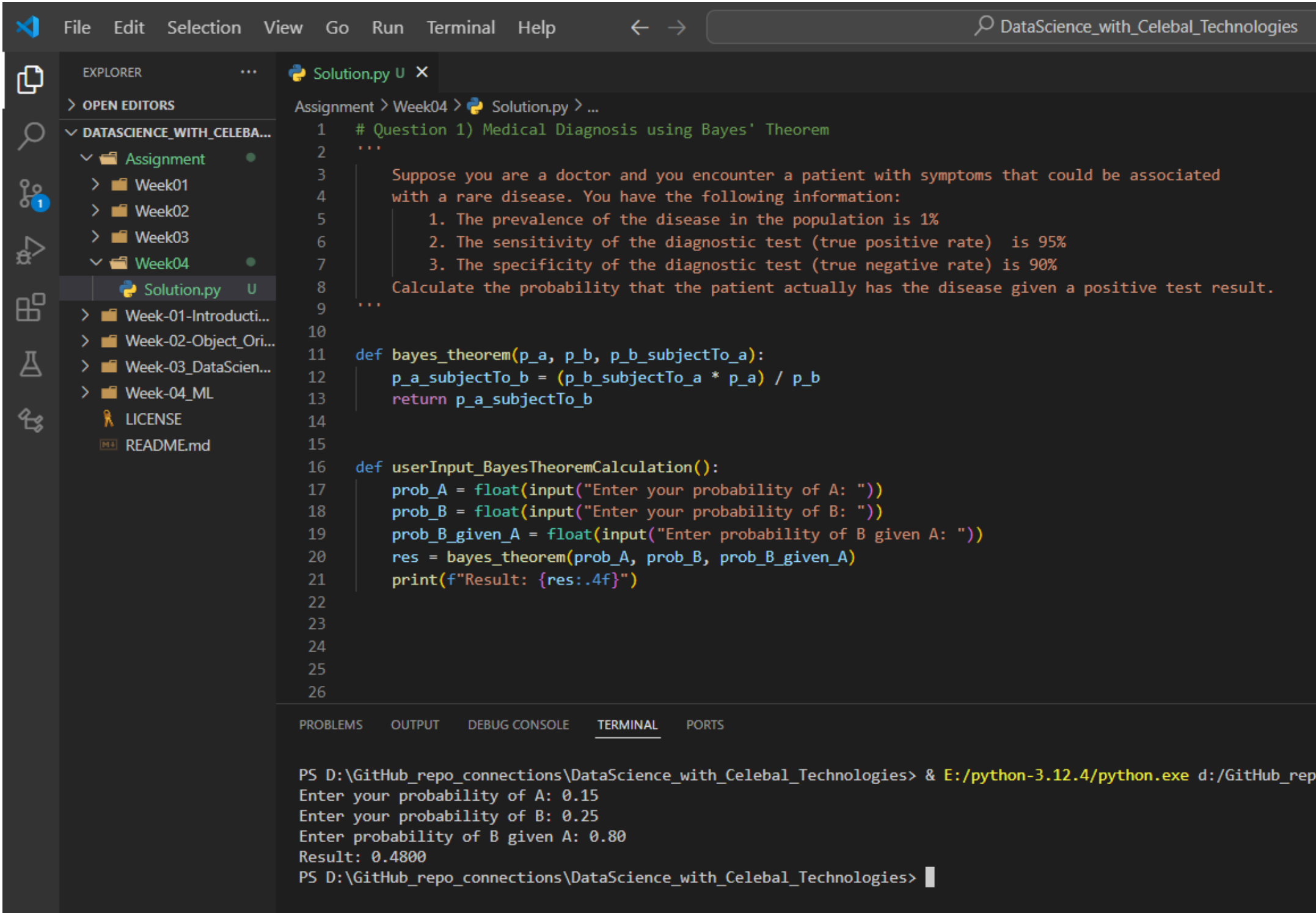
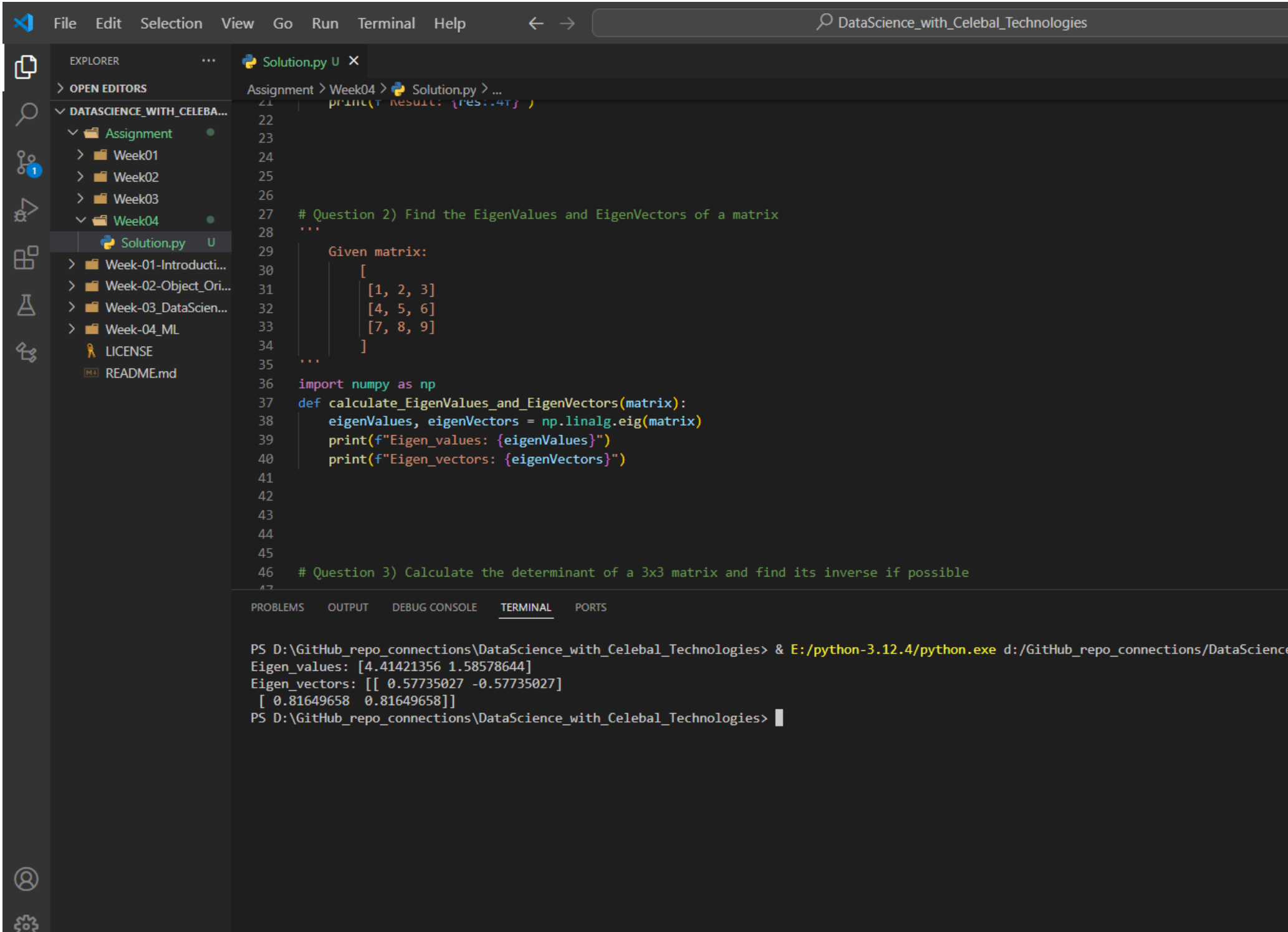
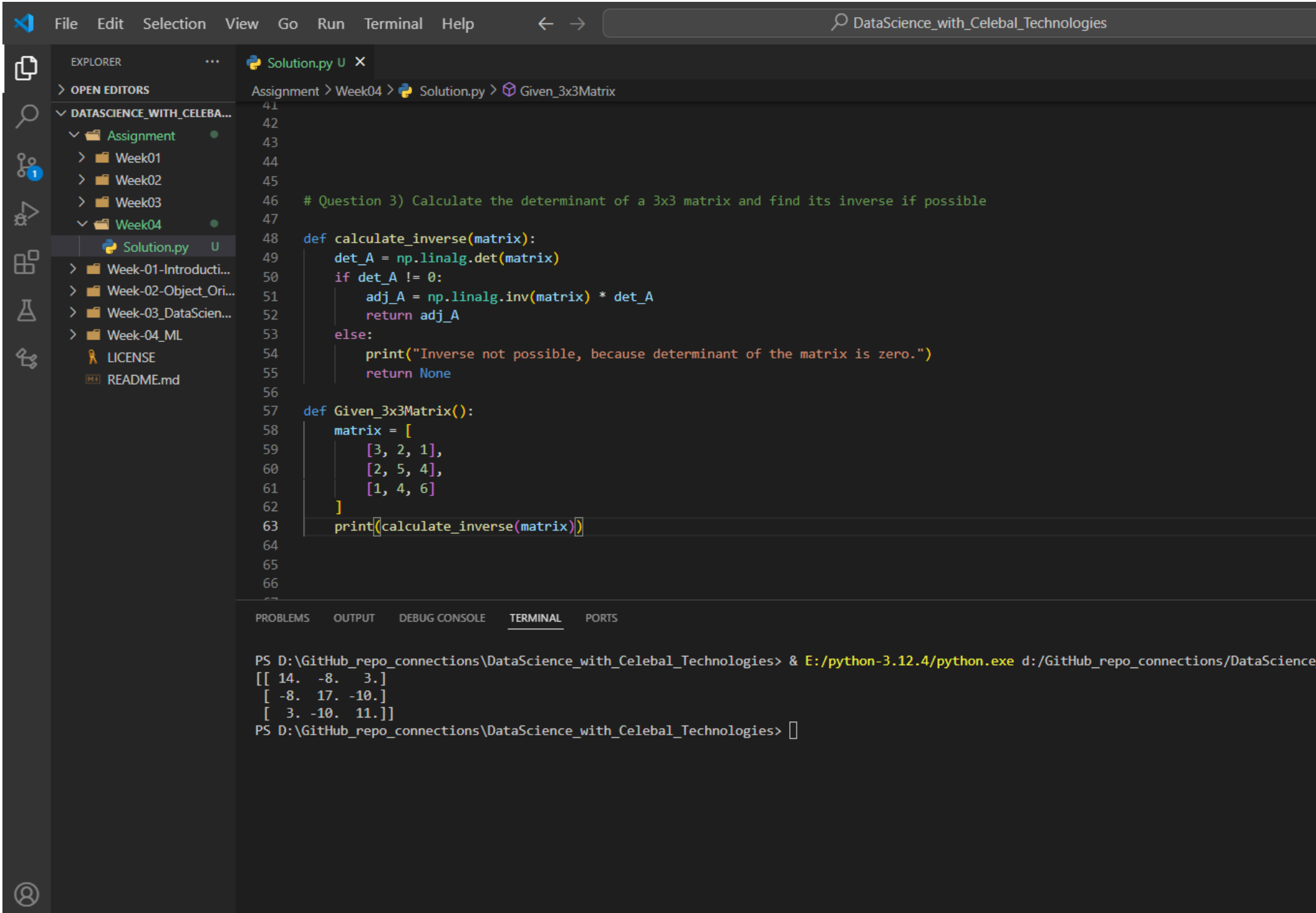
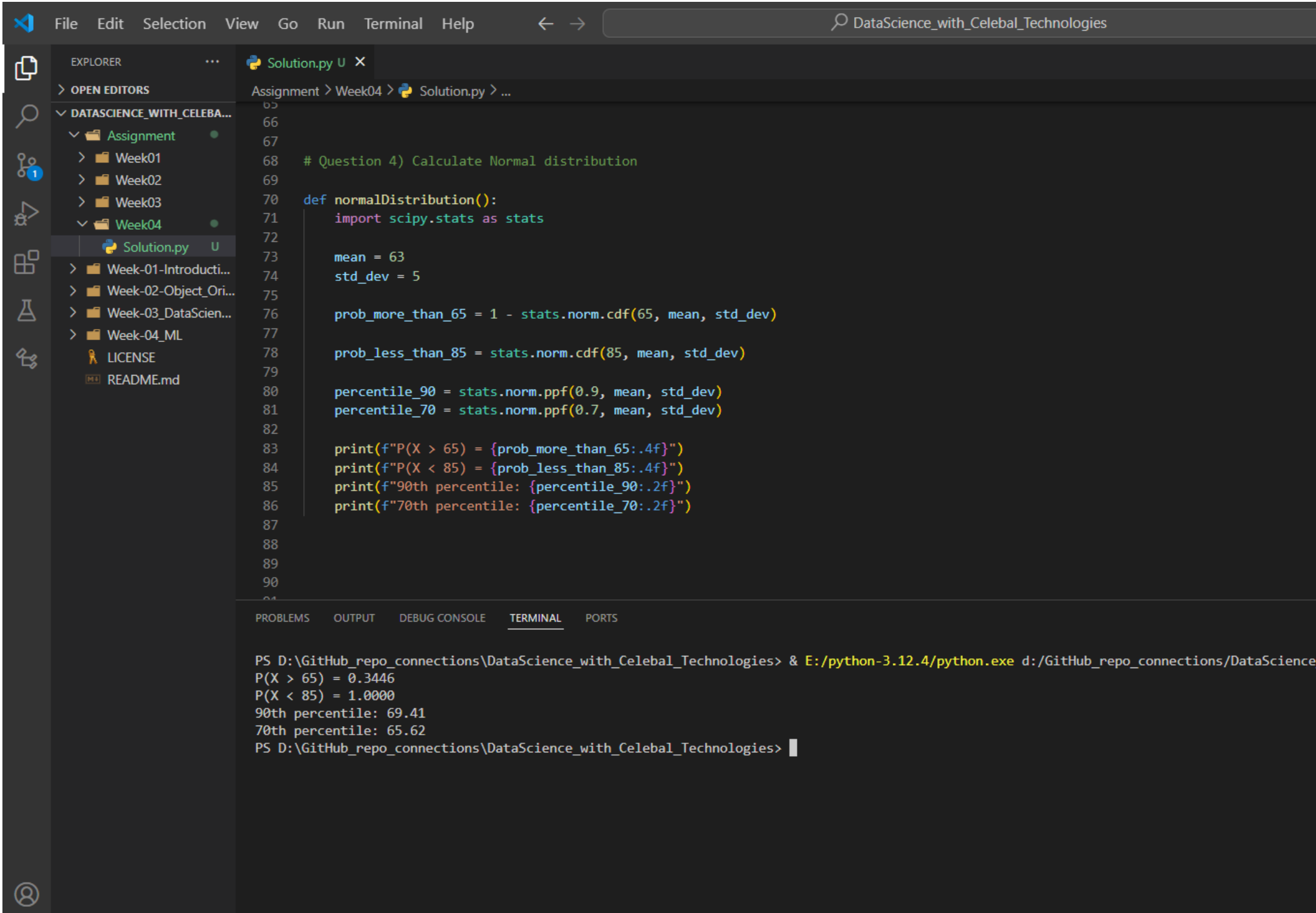

File Edit Selection View Go Run Terminal Help
DataScience_with_Celebal_Technologies
EXPLORER
OPEN EDITORS
DATASCIENCE_WITH_CELEBA...
Assignment
Week01
Week02
Week03
Week04
Solution.py U
Week-01-Introducti...
Week-02-Object_Ori...
Week-03_DataScien...
Week-04_ML
LICENSE
README.md
Solution.py U
Assignment > Week04 > Solution.py > ...
Question 1) Medical Diagnosis using Bayes' Theorem
'''
Suppose you are a doctor and you encounter a patient with symptoms that could be associated
with a rare disease. You have the following information:
1. The prevalence of the disease in the population is 1%
2. The sensitivity of the diagnostic test (true positive rate) is 95%
3. The specificity of the diagnostic test (true negative rate) is 90%
Calculate the probability that the patient actually has the disease given a positive test result.
'''
def bayes_theorem(p_a, p_b, p_b_subjectTo_a):
p_a_subjectTo_b = (p_b_subjectTo_a * p_a) / p_b
return p_a_subjectTo_b
def userInput_BayesTheoremCalculation():
prob_A = float(input("Enter your probability of A: "))
prob_B = float(input("Enter your probability of B: "))
prob_B_given_A = float(input("Enter probability of B given A: "))
res = bayes_theorem(prob_A, prob_B, prob_B_given_A)
print(f"Result: {res:.4f}")
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\GitHub_repo_connections\DataScience_with_Celebal_Technologies> & E:/python-3.12.4/python.exe d:/GitHub_rep
Enter your probability of A: 0.15
Enter your probability of B: 0.25
Enter probability of B given A: 0.80
Result: 0.4800
PS D:\GitHub_repo_connections\DataScience_with_Celebal_Technologies>


File Edit Selection View Go Run Terminal Help
DataScience_with_Celebal_Technologies
EXPLORER
OPEN EDITORS
DATASCIENCE_WITH_CELEBA...
Assignment
Week01
Week02
Week03
Week04
Solution.py U
Week-01-Introducti...
Week-02-Object_Ori...
Week-03_DataScien...
Week-04_ML
LICENSE
README.md
Solution.py U
Assignment > Week04 > Solution.py > ...
print(f"Result: {res:.4f}")
# Question 2) Find the EigenValues and EigenVectors of a matrix
'''
Given matrix:
[
[1, 2, 3]
[4, 5, 6]
[7, 8, 9]
]
'''
import numpy as np
def calculate_EigenValues_and_EigenVectors(matrix):
eigenValues, eigenVectors = np.linalg.eig(matrix)
print(f"Eigen_values: {eigenValues}")
print(f"Eigen_vectors: {eigenVectors}")
# Question 3) Calculate the determinant of a 3x3 matrix and find its inverse if possible
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\GitHub_repo_connections\DataScience_with_Celebal_Technologies> & E:/python-3.12.4/python.exe d:/GitHub_repo_connections/DataScience_with_C
Eigen_values: [4.41421356 1.58578644]
Eigen_vectors: [[0.57735027 -0.57735027]
[0.81649658 0.81649658]]
PS D:\GitHub_repo_connections\DataScience_with_Celebal_Technologies>


File Edit Selection View Go Run Terminal Help
DataScience_with_Celebal_Technologies
EXPLORER
OPEN EDITORS
DATASCIENCE_WITH_CELEBA...
Assignment
Week01
Week02
Week03
Week04
Solution.py U
Week-01-Introducti...
Week-02-Object_Ori...
Week-03_DataScien...
Week-04_ML
LICENSE
README.md
Solution.py U
Assignment > Week04 > Solution.py > Given_3x3Matrix
# Question 3) Calculate the determinant of a 3x3 matrix and find its inverse if possible
def calculate_inverse(matrix):
det_A = np.linalg.det(matrix)
if det_A != 0:
adj_A = np.linalg.inv(matrix) * det_A
return adj_A
else:
print("Inverse not possible, because determinant of the matrix is zero.")
return None
def Given_3x3Matrix():
matrix = [
[3, 2, 1],
[2, 5, 4],
[1, 4, 6]
]
print(calculate_inverse(matrix))
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\GitHub_repo_connections\DataScience_with_Celebal_Technologies> & E:/python-3.12.4/python.exe d:/GitHub_repo_connections/DataScience
[[14. -8. 3.]
[-8. 17. -10.]
[3. -10. 11.]]
PS D:\GitHub_repo_connections\DataScience_with_Celebal_Technologies>


File Edit Selection View Go Run Terminal Help
DataScience_with_Celebal_Technologies
EXPLORER
OPEN EDITORS
DATASCIENCE_WITH_CELEBA...
Assignment
Week01
Week02
Week03
Week04
Solution.py U
Week-01-Introducti...
Week-02-Object_Ori...
Week-03_DataScien...
Week-04_ML
LICENSE
README.md
Solution.py U
Assignment > Week04 > Solution.py > ...
# Question 4) Calculate Normal distribution
def normalDistribution():
import scipy.stats as stats
mean = 63
std_dev = 5
prob_more_than_65 = 1 - stats.norm.cdf(65, mean, std_dev)
prob_less_than_85 = stats.norm.cdf(85, mean, std_dev)
percentile_90 = stats.norm.ppf(0.9, mean, std_dev)
percentile_70 = stats.norm.ppf(0.7, mean, std_dev)
print(f"P(X > 65) = {prob_more_than_65:.4f}")
print(f"P(X < 85) = {prob_less_than_85:.4f}")
print(f"90th percentile: {percentile_90:.2f}")
print(f"70th percentile: {percentile_70:.2f}")
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\GitHub_repo_connections\DataScience_with_Celebal_Technologies> & E:/python-3.12.4/python.exe d:/GitHub_repo_connections/DataScience
P(X > 65) = 0.3446
P(X < 85) = 1.0000
90th percentile: 69.41
70th percentile: 65.62
PS D:\GitHub_repo_connections\DataScience_with_Celebal_Technologies>