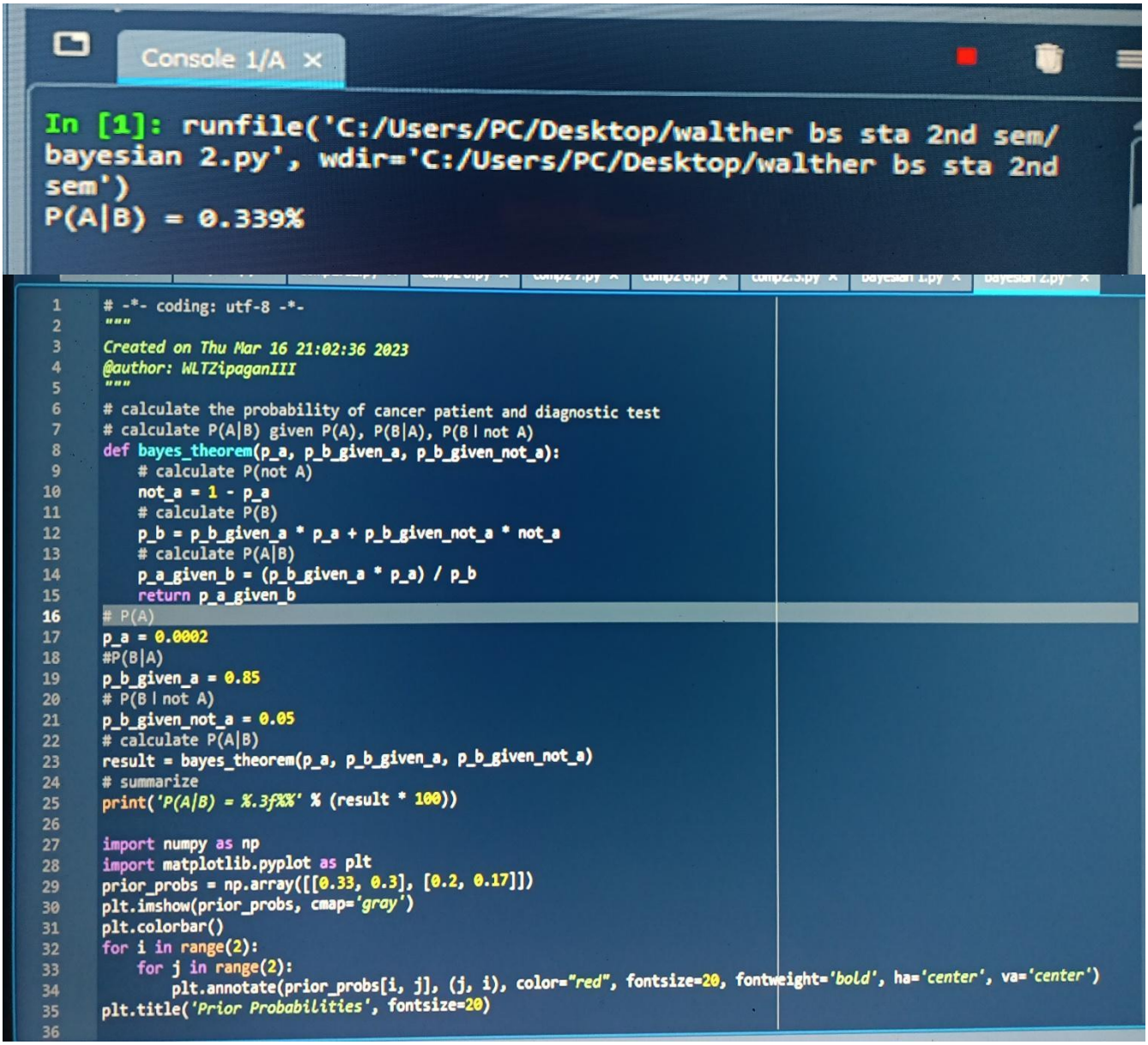
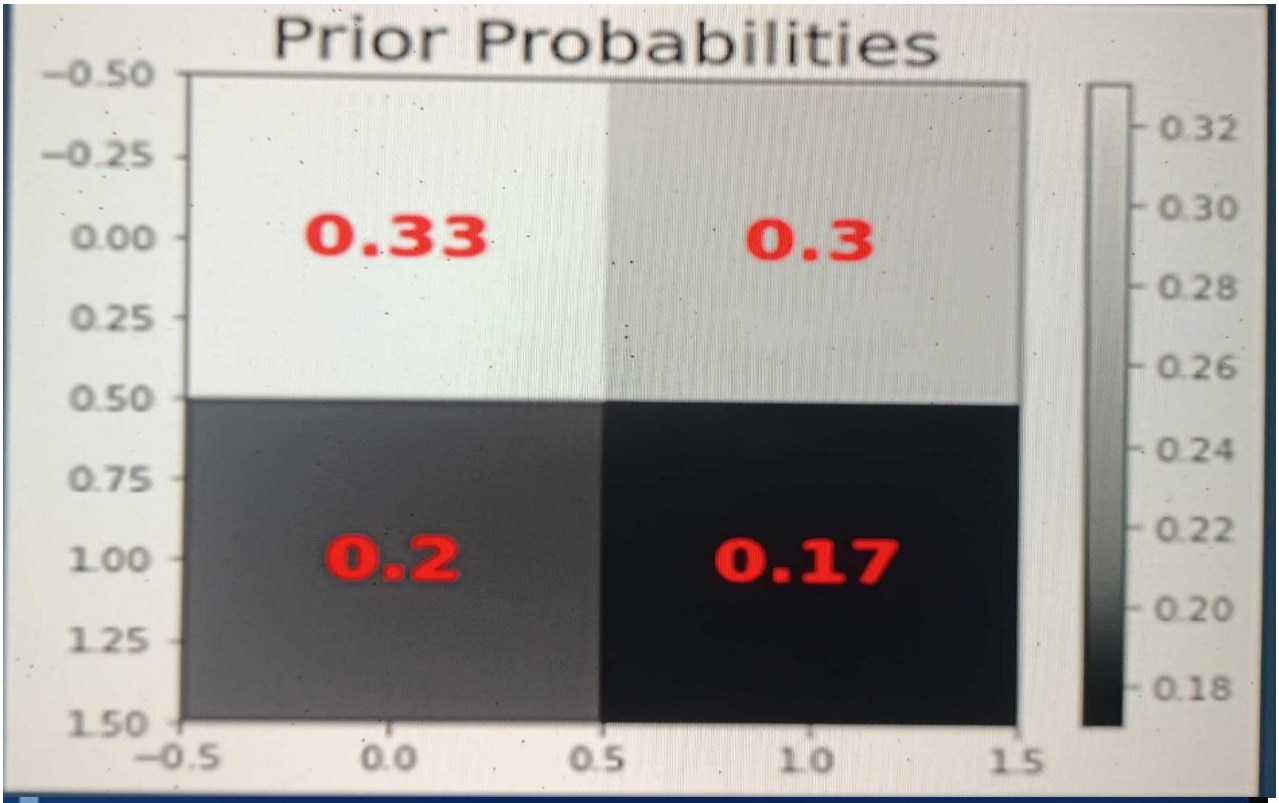
Walther Lewis T. Zipagan III

CAS-05-601P

```
# -*- coding: utf-8 -*-
"""
Created on Thu Mar 16 21:02:36 2023

@author: WLTZipaganIII
"""
# calculate the probability of cancer patient and diagnostic test

# calculate P(A|B) given P(A), P(B|A), P(B | not A)
def bayes_theorem(p_a, p_b_given_a, p_b_given_not_a):
    # calculate P(not A)
    not_a = 1 - p_a
    # calculate P(B)
    p_b = p_b_given_a * p_a + p_b_given_not_a * not_a
    # calculate P(A|B)
    p_a_given_b = (p_b_given_a * p_a) / p_b
    return p_a_given_b

# P(A)
p_a = 0.0002
#P(B|A)
p_b_given_a = 0.85
# P(B | not A)
p_b_given_not_a = 0.05
# calculate P(A|B)
result = bayes_theorem(p_a, p_b_given_a, p_b_given_not_a)
```

```
# summarize
print('P(A|B) = %.3f%%' % (result * 100))

import numpy as np
import matplotlib.pyplot as plt

prior_probs = np.array([[0.33, 0.3], [0.2, 0.17]])

plt.imshow(prior_probs, cmap='gray')
plt.colorbar()

for i in range(2):
    for j in range(2):
        plt.annotate(prior_probs[i, j], (j, i), color="red", fontsize=20, fontweight='bold', ha='center', va='center')

plt.title('Prior Probabilities', fontsize=20)
```


Console 1/A
In [1]: runfile('C:/Users/PC/Desktop/walther bs sta 2nd sem/bayesian 2.py', wdir='C:/Users/PC/Desktop/walther bs sta 2nd sem')
P(A|B) = 0.339%
-*- coding: utf-8 -*-
"""
Created on Thu Mar 16 21:02:36 2023
@author: WLTZipaganIII
"""
# calculate the probability of cancer patient and diagnostic test
# calculate P(A|B) given P(A), P(B|A), P(B | not A)
def bayes_theorem(p_a, p_b_given_a, p_b_given_not_a):
# calculate P(not A)
not_a = 1 - p_a
# calculate P(B)
p_b = p_b_given_a * p_a + p_b_given_not_a * not_a
# calculate P(A|B)
p_a_given_b = (p_b_given_a * p_a) / p_b
return p_a_given_b
# P(A)
p_a = 0.0002
#P(B|A)
p_b_given_a = 0.85
# P(B | not A)
p_b_given_not_a = 0.05
# calculate P(A|B)
result = bayes_theorem(p_a, p_b_given_a, p_b_given_not_a)
# summarize
print('P(A|B) = %.3f%%' % (result * 100))
import numpy as np
import matplotlib.pyplot as plt
prior_probs = np.array([[0.33, 0.3], [0.2, 0.17]])
plt.imshow(prior_probs, cmap='gray')
plt.colorbar()
for i in range(2):
for j in range(2):
plt.annotate(prior_probs[i, j], (j, i), color="red", fontsize=20, fontweight='bold', ha='center', va='center')
plt.title('Prior Probabilities', fontsize=20)



Name	Type	Size	Value
i	int	1	1
j	int	1	1
p_a	float	1	0.0002
p_b_given_a	float	1	0.85
p_b_given_not_a	float	1	0.05
prior_probs	Array of float64	(2, 2)	[[0.33 0.3] [0.2 0.17]]
result	float	1	0.0033891547049441782

Help | Variable Explorer | Plots | Files | Code Analysis