HW 1 Start

1.

a. Despite there being input data, this is not learning because they have already derived a statistical model to classify coins thus the machine is simply applying their model not learning and deriving its own.
b. This is supervised learning because the input coins are already labeled thus the machine already knows the correct output as all the coins will go with their own group.
c. This is Reinforcement Learning because the machine has an input data of possible moves however it must decide which move to make based on the penalty that comes with making that move. Since the machine has to weight the grade output of each move by playing out possible scenarios, this makes it reinforcement learning.
d. [a] Wrong because it does not match the above analysis.
e. [b] Wrong because it does not match the above analysis.
f. [c] Wrong because it does not match the above analysis.
g. [d] Correct because it matches the above analysis.
h. [e] Wrong because it does not match the above analysis.
i. [e] Wrong because it does not match the above analysis.

2.

a. This problem is not suited for machine learning because we already have an easy mathematical formula to identify prime and non-primes thus it would be a waste when we could just write a program that applies that formula.
b. This problem is suited for machine learning because we have data on factors related to fraud such as spending history, and we cannot pin down this issue mathematically, and there are patterns that exist between fraud and external factors like large or unusual purchases. As this problem meets the criteria for the essence of machine learning we can apply it here so that the machine can learn these patterns and generate a hypothesis.
c. This problem is not suited for machine learning because we already have a mathematical formula to calculate this thus we do not need to apply machine learning as we can just apply the formula.
d. This problem is suited for machine learning because the data on traffic cycles exists, and there is a pattern between cycles and amount of traffic, however an equation describing this relationship at such an intersection is unknown. As it meets the criteria for machine learning, this problem is suited for machine learning.
e. [a] Correct because it matches the above analysis.
f. [b] Wrong because it does not match the above analysis.
g. [c] Wrong because it does not match the above analysis.
h. [d] Wrong because it does not match the above analysis.
i. [e] Wrong because it does not match the above analysis.

3.

a. We originally have 3 Black balls out of 4 balls total. After the first pick, we picked a Black ball meaning that we have 3 balls left over with two of those balls being Black. Now the probability of getting a Black ball from the remaining 3 balls is ⅔ because only 2 out of the three remaining balls are Black hence ⅔ because getting a Black ball does not tell us the location of the white ball thus the choice ,are still random.
b. [a] This wrong because it does not match the above analysis.
c. [b] This is wrong because it does not match the above analysis.
d. [c] This is wrong because it does not match the above analysis.
e. [d] This is correct because it matches the above analysis
f. [e] This is wrong because it does not match the above analysis.

Q4: $P(\text{red}) = .55$ thus

$P(\text{!red}) = 1 - .55 = .45$

let $c = P(U=0 \text{ for a sample of size } n) = .45^n$

As $n = 10$ & there is one sample

thus $c = (45)^{10} = 3.41 \times 10^{-4}$.

Now closest answer: C - option -> closest to zero

[a] $(.45)^{10} - 7.331 \times 10^{-6} \approx 3.41 \times 10^{-4}$

[b] $(.45)^{10} - 3.405 \times 10^{-4} \approx 6.29 \times 10^{-4}$

[c] $(.45)^{10} - 0.289 \approx 0.289$

[d] $(.45)^{10} - 0.1150 \approx 0.450$

[e] $(.45)^{10} - 0.550 \approx 0.550$

Thus answer is b because it is closest to zero;

Q5: $P(\text{atleast one red}) = 1 - .45$

$P(\text{atleast one red marble in a sample size of } n) = 1 - .45^n$

$P(\text{atleast one red in a sample size } n \text{ with } K \text{ samples}) = 1 - \left([1-.45]^n\right)^K$

Thus $c = 1 - (1 - .45^{10})^{1000}$

Now find closest to C - option

[a] $C - 7.331 \times 10^{-6} \approx 284$

[b] $C - 3.405 \times 10^{-4} \approx 288$

[c] $C - 0.289 \approx 3.67 \times 10^{-4}$

[d] $C - 0.1150 \approx 0.161$

[e] $C - 0.550 \approx 0.261$

Thus answer is c bcz it's closest to 0.

6)

| input | outputs | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 101 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 110 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 111 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |

Calculate scores:

(a) g returns 1 for all 3 points:

score $= (3 \times 1) + (2 \times 3) + (1 \times 3) + (0 \times 1) = 12$

(b) g returns 0 for all 3 points:

score $= (3 \times 1) + (2 \times 3) + (1 \times 3) + (0 \times 1) = 12$

(c) g is XOR function:

score $= (3 \times 1) + (2 \times 3) + (1 \times 3) + (0 \times 1) = 12$

(d) g is XOR func:

score $= (3 \times 1) + (2 \times 3) + (1 \times 3) + (0 \times 1) = 12$

Hence the answer is E bcz they are all equivalent.

# PLA Code :

psc1.py

```python
from calendar import c
import random
import numpy

def make_random_point():
    return(1, random.uniform(-1, 1), random.uniform(-1, 1))
def get_slope(x1, y1, x2, y2):
    slope = ((y2 -y1)/(x2-x1))
    intercept = y1 - (slope * x1)
    return(slope, intercept)
def algo(n):
    points = []
    point1 = make_random_point()
    point2 = make_random_point()
    slope = get_slope(point1[1], point1[2], point2[1], point2[2])
    for i in range(n):
        rand_point = make_random_point()
        if ((slope[0]*rand_point[1]) + slope[1]) > rand_point[2]:
            points.append((rand_point, -1))
        else:
            points.append((rand_point, 1))

    w = [0,0,0]
    iterations = 0
    while True:
        to_fix = []
        for point, score in points:
            curr = numpy.sign((w[0]*point[0]) + (w[1] * point[1]) + (w[2] * point[2]))
            curr = int(curr)
            if curr != score:
                to_fix.append((point, score))
        if len(to_fix) == 0:
            return(w, iterations,slope[0], slope[1])
        else:
            iterations+=1
            point, score = random.choice(to_fix)
            w[0] = w[0] + score * point[0]
            w[1] = w[1] + score * point[1]
            w[2] = w[2] + score * point[2]

N = 100
runs = 1000
points = 500
to_fix = []
number_of_iterations = []

for i in range(runs):
    incorrect = 0
    pla = algo(N)
    (w, iterations,slope, intercept) = (pla[0], pla[1], pla[2], pla[3])
    number_of_iterations.append(pla[1])
    for num in range(points):
        curr = [1, random.uniform(-1,1), random.uniform(-1,1)]
        func = pla[2]*curr[1] + pla[3]
        foutcome = -1 if (func > curr[2]) else 1
        woutcome = numpy.sign((pla[0][0]*curr[0]) + (pla[0][1] * curr[1]) + (pla[0][2] * curr[2]))
        woutcome = int(woutcome)
        if foutcome != woutcome:
            incorrect += 1
    to_fix.append(incorrect/points)
frq1 = sum(number_of_iterations) / len(number_of_iterations)
frq2 = sum(to_fix) / len(to_fix)
print(frq1, frq2)
```

```
zsh: permission denied: /var/folders/3q/xzc9d2cn7nn5w3slm1qxt32w0000gn/T/TemporaryItems/NSIRD_screencaptureui_bCJcGY/Screen Shot 2022-10-03 at 2.41.31 PM.pn
g
thierno@Thiernos-MacBook-Pro ~ % /usr/bin/python3 /Users/thierno/Desktop/psc1.py
9.917 0.10602199999999994   N = 10
thierno@Thiernos-MacBook-Pro ~ % /usr/bin/python3 /Users/thierno/Desktop/psc1.py
102.083 0.013259999999999928   N = 100
thierno@Thiernos-MacBook-Pro ~ %
```

(7) we got 9.917 thus;

[a] 1 − 9.917 = −8.917

✓ [b] 15 − 9.917 = 5.083

[c] 300 − 9.917 = 290.083

[d] 5000 − 9.917 = 4990.083

[e] 10000 − 9.917 = 9990.083

Thus answer is b bcz it's closet to zero.

(8)

[a] 0.001 − 0.106 = −0.105

[b] 0.01 − 0.106 = −0.096

✓ [c] 0.1 − 0.106 = −0.006

[d] 0.5 − 0.106 = 0.394

[e] 0.8 − 0.106 = 0.694

Thus answer is c bcz it's closes to zero.

(9)

[a] 50 − 102 = −52

✓ [b] 100 − 102 = −2

[c] 500 − 102 = 398

[d] 1000 − 102 = 898

[e] 5000 − 102 = 4898

Thus answer is b bcz it's closes to zero.

(10)

[a] 0.001 − 0.0133 = −0.0123

✓ [b] 0.01 − 0.0133 = −0.0033

[c] 0.1 − 0.0133 = 0.0867

[d] 0.5 − 0.0133 = 0.4867

[e] 0.8 − 0.0133 = 0.7867

Thus answer is b bcz it's closes to zero.

# HW1 Ends