Homework 1 - Question #2

Part 1

Draw the datapoints and regression curve.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
points = {'x_values': [0, 2, 2, 5],
          'y_values': [2, 10, 12, 20],
          'estimation': [3, 11, 11, 23]}
points_dataset = pd.DataFrame(data=points)
```

```
points_dataset
```

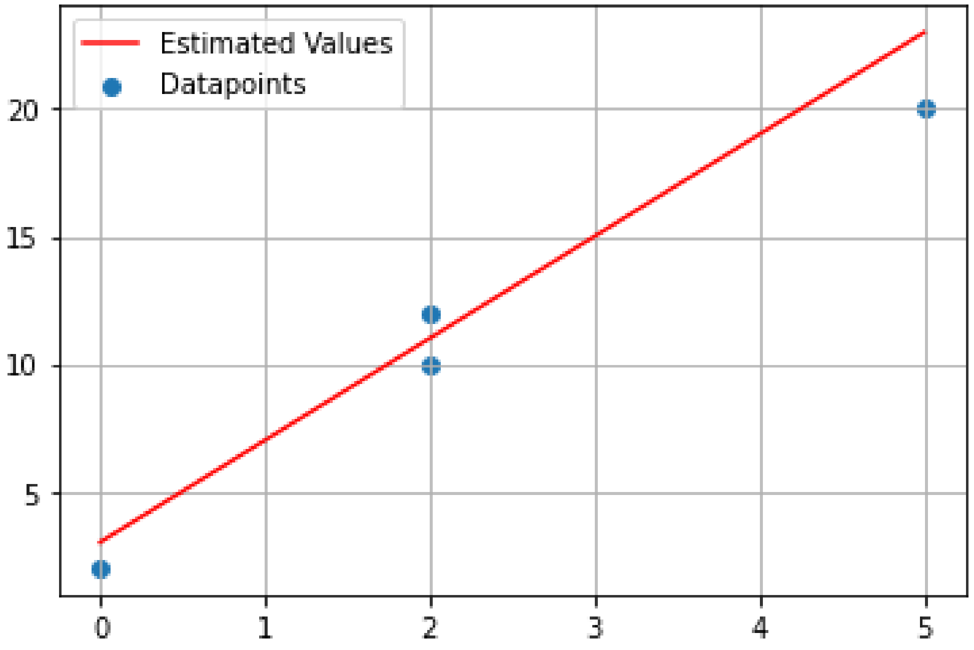
```
   x_values  y_values  estimation
0         0         2           3
1         2        10          11
2         2        12          11
3         5        20          23
```

```
reg_points = {'reg_x': [0, 2, 5],
              'reg_y': [3, 11, 23]}
reg_dataset = pd.DataFrame(data=reg_points)
```

```
reg_dataset
```

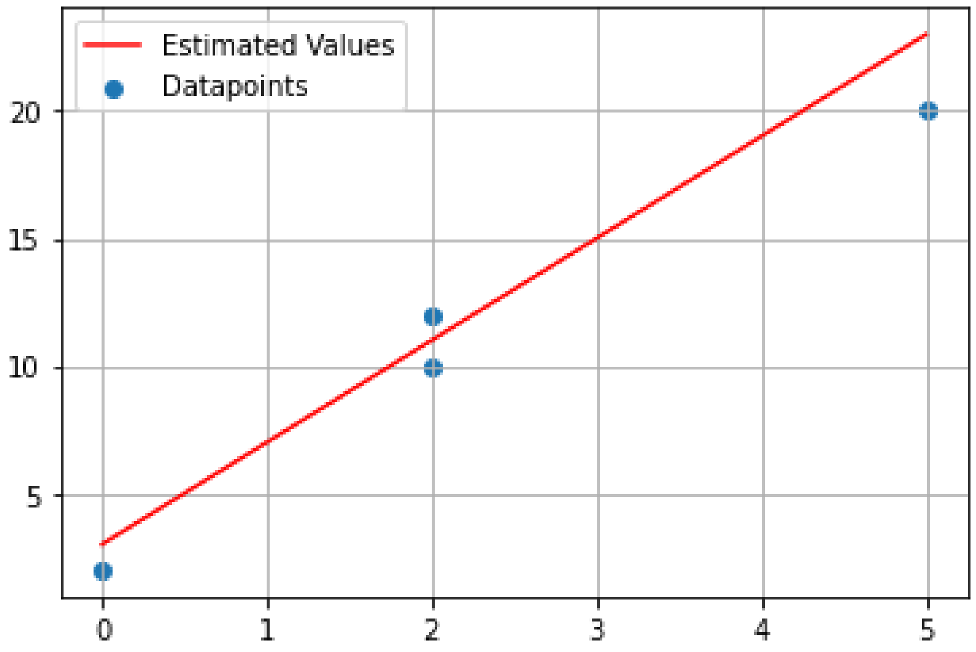
```
   reg_x  reg_y
0      0      3
1      2     11
2      5     23
```

```
#SOLUTION APPROACH 1: Regression Line by Enterin Points
x = points_dataset.x_values
y = points_dataset.y_values

regg_x = reg_dataset.reg_x
regg_y = reg_dataset.reg_y

plt.scatter(x,y)
plt.plot(regg_x, regg_y, color = 'red')

plt.grid()
plt.legend(["Estimated Values", "Datapoints"])
plt.show()
```



```
#SOLUTION APPROACH 2: Regression Line by Formula
points = np.array([(0,3), (2,11), (5,23)])

# get x and y vectors
x = points[:,0]
y = points[:,1]

# calculate polynomial
z = np.polyfit(x, y, 1)

print(z)

a = z[0]
b = z[1]

y_pred = a*x + b
plt.plot(x, y_pred, color = 'red')

x = points_dataset.x_values
y = points_dataset.y_values

plt.scatter(x,y)
plt.grid()
plt.legend(["Estimated Values", "Datapoints"])
plt.show()
```

```
[4. 3.]
```



Part 2

What is the MSE? Calculate error for each instance.

- Instance 1 ---> $(2-3)^2=1$
- Instance 2 ---> $(10-11)^2=1$
- Instance 3 ---> $(12-11)^2=1$
- Instance 4 ---> $(20-23)^2=9$

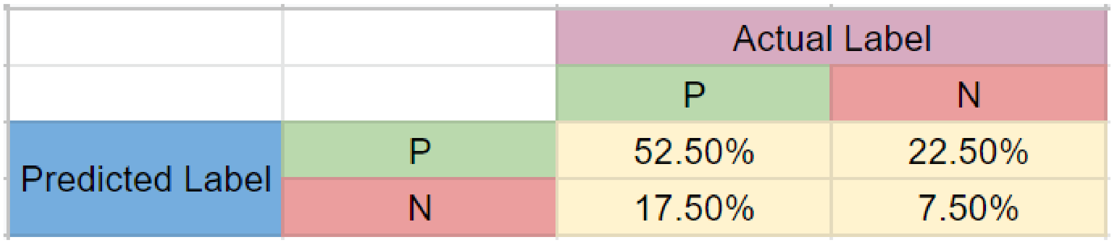
MSE (Mean Squared Error) = $(1+1+1+9)/4=3$

Homework 1 - Question #3

You are given a biased classifier that produce random results for any given query. Probability of getting positive label estimated to be 0.75

Part 1

What will be the accuracy of such model on a dataset with a class imbalance 70% positive instances?



		Actual Label	
		P	N
Predicted Label	P	52.50%	22.50%
	N	17.50%	7.50%

```
prob_positive_label = 0.75
positive_instances = 70

prob_negative_label = 0.25
negative_instances = 30

TruePositives = (prob_positive_label * positive_instances)
TrueNegatives = (prob_negative_label * negative_instances)

FalsePositives = 75 - TruePositives
FalseNegatives = 25 - TrueNegatives

Accuracy = TruePositives + TrueNegatives

print("True Positive:", TruePositives)
print("False Positive:", FalsePositives)
print("False Negatives:", FalseNegatives)
print("True Negatives:", TrueNegatives)

print('\n')

print("Accuracy (TP + TN):", Accuracy)
```

```
True Positive: 52.5
False Positive: 22.5
False Negatives: 17.5
True Negatives: 7.5


Accuracy (TP + TN): 60.0
```

Part 2

What is the entropy of the random model predictions?

```
# calculate the entropy for a dataset
from math import log2

# proportion of examples in each class
prob_y_positive = 0.75
prob_y_negative = 0.25

# calculate entropy
entropy = -(prob_y_positive * log2(prob_y_positive) + prob_y_negative
* log2(prob_y_negative))

# print the result
print("Entropy: ", entropy, " =  %.4f" % entropy)
```

```
Entropy:  0.8112781244591328  =  0.8113
```

CS 412 - Homework 1
Berna Yildiran
byildiran
26431

Question #1
https://drive.google.com/file/d/142s1p3sqvl3MzogIW-oRGLvTIxnKFOxl/view?usp=sharing

Question #2
https://colab.research.google.com/drive/1iCZ7KhIWX_FiES-8N3C3LDGGxVj-bhhs?usp=sharing

Question #3
https://colab.research.google.com/drive/1f-e4P1nBWsOFSWbANW0U45e0OSvm0j_7?usp=sharing