Task: 02

The goal of this task is to implement **Gradient Descent** algorithm in Python. Gradient Descent dictates how the weights get updated from an inital value to ensure we reach a minimal loss value.

In [1]:
```
#import necessary libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Let us make use of a randomly-created sample dataset as follows

In [2]:
```
#sample-dataset
x_data = [1.0, 2.0, 3.0]
y_data = [2.0, 4.0, 6.0]

# Let us initialize our weight value (w) with 1.0
w = 1.0
```

Task: 02 - a

Implement the forward and loss functions

In [3]:
```
#forward function to calculate y_pred for a given x according to the linear model de
def forward(x):
    #implement the forward model to compute y_pred as w*x
    ## YOUR CODE STARTS HERE
    y= w*x
    return y
    ## YOUR CODE ENDS HERE

#loss-function to compute the mean-squared error between y_pred and y_actual
def loss(y_pred, y_actual):
    #calculate the mean-squared-error between y_pred and y_actual
    ## YOUR CODE STARTS HERE
    loss = (y_pred - y_actual)**2
    return loss

    ## YOUR CODE ENDS HERE
```

Gradient Descent

We update the `w` such that loss is minimum. The factor by which `w` is updated each time is called `alpha(learning rate)`.

New `w` is `w` minus `alpha` times derivative of `loss` against `w`, which can be mathematically expressed as follows:

$$w = w - \alpha * \frac{d(loss)}{dw}$$

This equation is dependent on how the loss function has been defined. In the current case below formula will dictate how to update the value of w for each pass.

$w = w - \alpha * (2x) * (y_{pred} - y_{actual})$

Task: 02 - b

Complete the gradient function

In [4]:
```
# Function to calculate the gradient for w to be updated and get min loss.
# Gradient = derivative of the loss for constant x and y with respect to the weight

def gradient(x,y):
    #implement the gradient of loss with respect to the weight (w)
    ## YOUR CODE STARTS HERE
    grad = alpha * 2*x*(y_pred - y)
    return grad

    ## YOUR CODE ENDS HERE
```

Calculate y_{pred} for $x = 4$ without training the model

In [5]:
```
y_pred_without_train = forward(4)
```

Begin Training

In [6]:
```
# In this method, we learn the dataset multiple times (called epochs)
# Each time, the weight (w) gets updates using the graident decent algorithm based o

alpha = 0.01 # Let us set learning rate as 0.01
weight_list = []
loss_list=[]

for epoch in range(100):
    total_loss=0
    count = 0
    for x, y in zip(x_data, y_data):

        #call the forward function to calculate y_pred
        ## YOUR CODE STARTS HERE
        y_pred = forward(x)
        ## YOUR CODE ENDS HERE

        #call the gradient function to obtain the grad for the given data-pair and u
        ## YOUR CODE STARTS HERE
        grad = gradient(x,y)
        w = w - grad
        ## YOUR CODE ENDS HERE

        #call the loss function to obtain the mean_squared_error of the current samp
        ## YOUR CODE STARTS HERE
        current_loss = loss(y_pred , y )
        ## YOUR CODE ENDS HERE

        total_loss+=current_loss

        count += 1

    avg_mse = total_loss / count
```

```
    print('Progress: ', epoch, 'w=', w, 'loss=', avg_mse)
    weight_list.append(w)
    loss_list.append(avg_mse)
```

```
Progress:  0 w= 1.260688 loss= 4.0525143466666655
Progress:  1 w= 1.453417766656 loss= 2.2150323422596667
Progress:  2 w= 1.5959051959019805 loss= 1.2106973245614812
Progress:  3 w= 1.701247862192685 loss= 0.6617456475624202
Progress:  4 w= 1.7791289594933983 loss= 0.36169841395033914
Progress:  5 w= 1.836707389300983 loss= 0.19769792689395907
Progress:  6 w= 1.8792758133988885 loss= 0.1080581744091792
Progress:  7 w= 1.910747160155559 loss= 0.05906267829964466
Progress:  8 w= 1.9340143044689266 loss= 0.03228261061229798
Progress:  9 w= 1.9512159834655312 loss= 0.01764510140664452
Progress:  10 w= 1.9639333911678687 loss= 0.00964449893442513
Progress:  11 w= 1.9733355232910992 loss= 0.005271511767061955
Progress:  12 w= 1.9802866323953892 loss= 0.0028813146747399157
Progress:  13 w= 1.9854256707695 loss= 0.0015748754098861908
Progress:  14 w= 1.9892250235079405 loss= 0.0008607989187741338
Progress:  15 w= 1.9920339305797026 loss= 0.000470497395483675
Progress:  16 w= 1.994110589284741 loss= 0.0002571655172060062
Progress:  17 w= 1.9956458879852805 loss= 0.0001405621027335221
Progress:  18 w= 1.9967809527381737 loss= 7.68287480356221e-05
Progress:  19 w= 1.9976201197307648 loss= 4.199322868633153e-05
Progress:  20 w= 1.998240525958391 loss= 2.295275272069943e-05
Progress:  21 w= 1.99869919972735 loss= 1.2545566843471576e-05
Progress:  22 w= 1.9990383027488265 loss= 6.857183943871714e-06
Progress:  23 w= 1.9992890056818404 loss= 3.7480149144916997e-06
Progress:  24 w= 1.999474353368653 loss= 2.0485983625691286e-06
Progress:  25 w= 1.9996113831376856 loss= 1.1197274682384291e-06
Progress:  26 w= 1.9997126908902887 loss= 6.120231403258154e-07
Progress:  27 w= 1.9997875889274812 loss= 3.345209748970213e-07
Progress:  28 w= 1.9998429619451539 loss= 1.8284322155969052e-07
Progress:  29 w= 1.9998838998815958 loss= 9.993885639181628e-08
Progress:  30 w= 1.9999141657892625 loss= 5.462480332433112e-08
Progress:  31 w= 1.9999365417379913 loss= 2.9856946996895115e-08
Progress:  32 w= 1.9999530845453979 loss= 1.6319276770280646e-08
Progress:  33 w= 1.9999653148414271 loss= 8.919826743712008e-09
Progress:  34 w= 1.999974356846045 loss= 4.875418822697411e-09
Progress:  35 w= 1.9999810417085633 loss= 2.664817308553861e-09
Progress:  36 w= 1.9999859839076413 loss= 1.4565417959819284e-09
Progress:  37 w= 1.9999896377347262 loss= 7.961198678232563e-10
Progress:  38 w= 1.999992339052936 loss= 4.3514497534261613e-10
Progress:  39 w= 1.9999943361699042 loss= 2.3784251245645866e-10
Progress:  40 w= 1.9999958126624442 loss= 1.3000049163724156e-10
Progress:  41 w= 1.999996904251097 loss= 7.105595905649731e-11
Progress:  42 w= 1.999997711275687 loss= 3.8837924790431617e-11
Progress:  43 w= 1.9999983079186507 loss= 2.1228119672861727e-11
Progress:  44 w= 1.9999987490239537 loss= 1.1602913056989068e-11
Progress:  45 w= 1.9999990751383971 loss= 6.3419461313939024e-12
Progress:  46 w= 1.9999993162387186 loss= 3.4663950822524747e-12
Progress:  47 w= 1.9999994944870796 loss= 1.8946699647269e-12
Progress:  48 w= 1.9999996262682318 loss= 1.0355929406354807e-12
Progress:  49 w= 1.999999723695619 loss= 5.660367024448634e-13
Progress:  50 w= 1.9999997957248556 loss= 3.0938560464107196e-13
Progress:  51 w= 1.9999998489769344 loss= 1.691046745512068e-13
Progress:  52 w= 1.9999998883468353 loss= 9.242961093241007e-14
Progress:  53 w= 1.9999999174534755 loss= 5.052038309321895e-14
Progress:  54 w= 1.999999938972364 loss= 2.7613543938978922e-14
Progress:  55 w= 1.9999999548815364 loss= 1.509307240699509e-14
Progress:  56 w= 1.9999999666433785 loss= 8.24960519393613e-15
Progress:  57 w= 1.9999999753390494 loss= 4.509087636292239e-15
Progress:  58 w= 1.9999999817678633 loss= 2.4645871936597784e-15
Progress:  59 w= 1.9999999865207625 loss= 1.3470995683872634e-15
```

```
Progress:  60 w= 1.999999990034638 loss= 7.363006925426945e-16
Progress:  61 w= 1.9999999926324883 loss= 4.0244888963833234e-16
Progress:  62 w= 1.99999999455311 loss= 2.1997141494645185e-16
Progress:  63 w= 1.9999999959730488 loss= 1.2023247480972207e-16
Progress:  64 w= 1.9999999970228268 loss= 6.57169288724515e-17
Progress:  65 w= 1.9999999977989402 loss= 3.5919704877484956e-17
Progress:  66 w= 1.9999999983727301 loss= 1.9633069604482358e-17
Progress:  67 w= 1.9999999987969397 loss= 1.0731089361601375e-17
Progress:  68 w= 1.9999999999110563 loss= 5.865424223592648e-18
Progress:  69 w= 1.9999999993424284 loss= 3.20593709572895e-18
Progress:  70 w= 1.9999999995138495 loss= 1.7523084712433846e-18
Progress:  71 w= 1.9999999996405833 loss= 9.577798640455507e-19
Progress:  72 w= 1.999999999734279 loss= 5.235054256286194e-19
Progress:  73 w= 1.9999999998035491 loss= 2.8613885224338016e-19
Progress:  74 w= 1.9999999998547615 loss= 1.563982536494851e-19
Progress:  75 w= 1.9999999998926234 loss= 8.548468207419028e-20
Progress:  76 w= 1.9999999999206153 loss= 4.672426679712873e-20
Progress:  77 w= 1.9999999999413098 loss= 2.5538821055088412e-20
Progress:  78 w= 1.9999999999566096 loss= 1.395911981720485e-20
Progress:  79 w= 1.9999999999679208 loss= 7.629835489260853e-21
Progress:  80 w= 1.9999999999762834 loss= 4.1703587348634304e-21
Progress:  81 w= 1.999999999982466 loss= 2.2794492685101314e-21
Progress:  82 w= 1.9999999999870368 loss= 1.2459146530984002e-21
Progress:  83 w= 1.999999999990416 loss= 6.809938252034201e-22
Progress:  84 w= 1.9999999999929146 loss= 3.7222634171833075e-22
Progress:  85 w= 1.9999999999947617 loss= 2.0345645351347754e-22
Progress:  86 w= 1.9999999999961273 loss= 1.111956767993656e-22
Progress:  87 w= 1.999999999997137 loss= 6.078024234250996e-23
Progress:  88 w= 1.9999999999978835 loss= 3.321867956024464e-23
Progress:  89 w= 1.9999999999984353 loss= 1.8155153909015986e-23
Progress:  90 w= 1.9999999999988431 loss= 9.924324587996338e-24
Progress:  91 w= 1.9999999999991447 loss= 5.4254937562672315e-24
Progress:  92 w= 1.9999999999993676 loss= 2.965078402510872e-24
Progress:  93 w= 1.9999999999995324 loss= 1.6208041997509026e-24
Progress:  94 w= 1.9999999999996543 loss= 8.861543418439488e-25
Progress:  95 w= 1.9999999999997444 loss= 4.844903798592123e-25
Progress:  96 w= 1.999999999999811 loss= 2.645787049730281e-25
Progress:  97 w= 1.9999999999998603 loss= 1.4456574558768205e-25
Progress:  98 w= 1.9999999999998967 loss= 7.894303641869682e-26
Progress:  99 w= 1.9999999999999236 loss= 4.3314528064842435e-26
```

Calculate y_{pred} for $x = 4$ after training the model

In [7]:

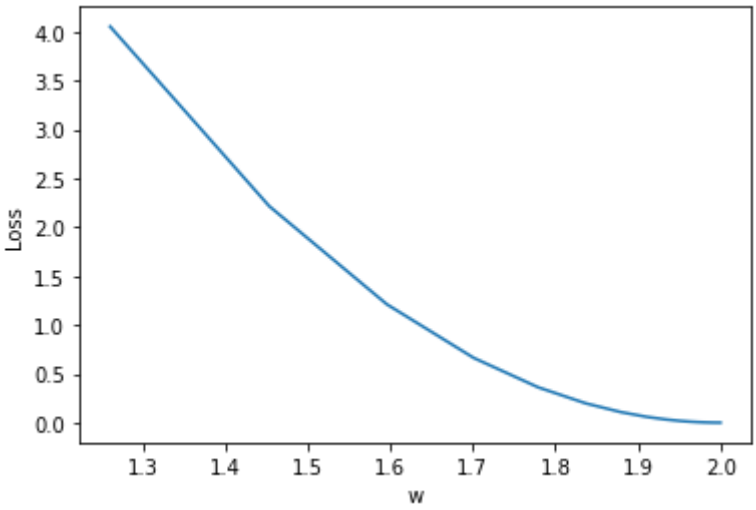
```
y_pred_with_train = forward(4)

print("Actual Y Value for x=4 : 8")
print("Predicted Y Value before training : " , y_pred_without_train)
print("Predicted Y Value after training : " , y_pred_with_train)
```

```
Actual Y Value for x=4 : 8
Predicted Y Value before training :  4.0
Predicted Y Value after training :  7.9999999999996945
```

Visualize Loss as a function of weight

In [8]:

```
plt.plot(weight_list, loss_list)
plt.ylabel('Loss')
plt.xlabel('w')
plt.show()
```



In [ ]: