```
#calculating the norm of a matrix
#function to generate random matrix A
import numpy as np

matrix_val=np.random.random((23,11)) #this can be adjusted to fit the 23*11 rule
#writing a function to calculate the norm of the above matrix

#this function calculaetes the l_infinity_norm of the matrix
matrix__val_norm = np.linalg.norm(matrix_val)

print(f'The calculated l∞ norm of the matrix is ={matrix__val_norm} ')
```

```
The calculated l∞ norm of the matrix is =9.399744096164651
```

```
import numpy as np #neccesary libraries
import matplotlib.pyplot as plt

#defining the gradient descent mean value function
def mse(init_val, final_val):

    Xk_vals = np.sum((init_val-final_val)**2) / len(init_val)
    return Xk_vals

#defining the gradient descent function with the given stopping criterion
def gradient_descent_function(A,b, iterations = 100, rate = 0.001,stopping_criterion = 1e-
    #defining and initializing the parameters
    start_iter_val = 0.1
    b_val = 0.01
    iterations = iterations
    rate = rate
    n = float(len(A))
    Xk = []
    iter_var = []
    Xk_prev = None
    for i in range(iterations):

        #updating the Xk values
        b_up = (start_iter_val * A) + b_val
        Xk_new = mse(b, b_up)

        # defining the stopping criterion
        if Xk_prev and abs(Xk_prev-Xk_new)<=stopping_criterion:
            break

        Xk_prev = Xk_new
        Xk.append(Xk_new)
        iter_var.append(start_iter_val)

        #finding the values of gradients
        iter_val_der = -(2/n) * sum(A * (b-b_up))
        b_val_der = -(2/n) * sum(b-b_up)
```

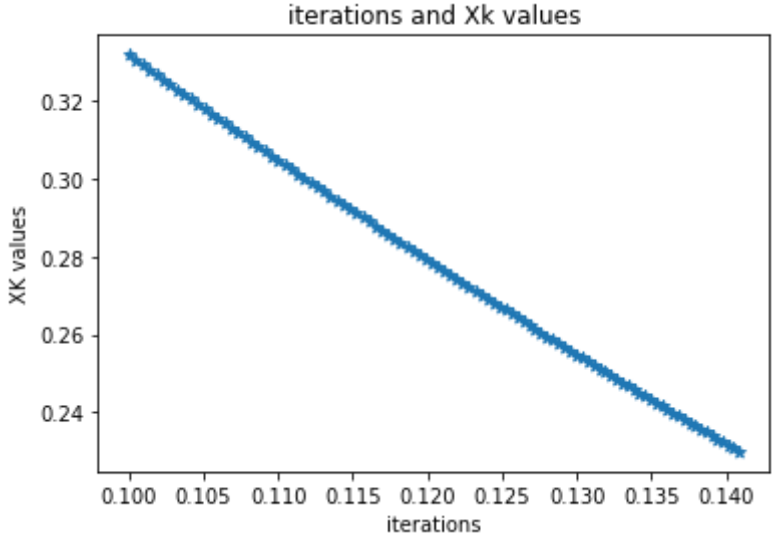
```
        start_iter_val = start_iter_val - (rate * iter_val_der)
        b_val = b_val - (rate * b_val_der)

    plt.plot(iter_var, Xk)
    plt.scatter(iter_var, Xk,marker='*')
    plt.title("iterations and Xk values")
    plt.ylabel("XK values")
    plt.xlabel("iterations")
    return Xk_new

#initializing the random marix
A=np.random.random((12))
b=np.random.random((12))

minimum_val = gradient_descent_function(A, b, iterations=100)
print(f"The minimum occurs at  {minimum_val}")
```

```
The minimum occurs at  0.22988247923006125
```



✓ 0s completed at 1:17 AM