

```

1 #####
2 # FILE : main.py
3 # WRITER : Raphael Haehnel
4 # DESCRIPTION: Introduction to Machine Learning (BIU), ex1
5 #####
6
7 import numpy as np
8
9
10 def generate_x(n: int, m: int):
11     """
12     Generate a random matrix of dimension n x m
13     :param n: The number of samples
14     :param m: The number of features
15     :return: The matrix X
16     """
17     return np.random.uniform(low=0, high=1, size=(n, m))
18
19
20 def generate_beta(m: int):
21     """
22     Generate the weight matrix
23     :param m: The number of features
24     :return: The matrix beta
25     """
26     return np.array([np.arange(1, m+1, 1)]).T
27
28
29 def generate_epsilon(n: int, sigma: float = 1.0):
30     """
31     Generate a random vector
32     :param n: Size of the vector
33     :param sigma: The standard deviation of the random generated elements
34     :return: A vector of random elements
35     """
36     return np.random.normal(0, sigma, size=(n, 1))
37
38
39 def compute_y(x: np.ndarray, beta: np.ndarray, epsilon: np.ndarray):
40     """
41     Compute the y values of the training example
42     :param x: The x values of the training example
43     :param beta: The vector of parameters
44     :param epsilon: The vector of random values
45     :return: A vector of values
46     """
47     return np.dot(x, beta) + epsilon
48
49
50 def solve_beta(x: np.ndarray, y: np.ndarray):
51     """
52     Find the beta parameters of the linear regression
53     :param x: Training example
54     :param y: Training example
55     :return: The parameter vector beta of the regression model

```

```
56     """
57     a1 = np.linalg.inv(np.dot(x.T, x))
58     a2 = np.dot(x.T, y)
59     return np.dot(a1, a2)
60
61
62 if __name__ == '__main__':
63     n = 40
64     m = 2
65     sigma = 1
66
67     X = generate_x(n, m)
68     beta = generate_beta(m)
69     epsilon = generate_epsilon(n, sigma)
70     Y = compute_y(X, beta, epsilon)
71     new_beta = solve_beta(X, Y)
72
73
```