In [28]:
```
import numpy as np
import pandas as pd
df = pd.read_csv("https://raw.githubusercontent.com/simransinghgulati/chemical-segregation/master/Logistic_X_Train.cs
df
```

Out[28]:

	f1	f2	f3
0	-1.239375	0.749101	-0.528515
1	-1.036070	0.801436	-1.283712
2	-0.615579	1.579521	-1.391927
3	1.335978	1.348651	1.433564
4	0.658925	1.300019	0.571603
...	...	...	...
2995	-0.455628	1.302303	-1.338027
2996	-0.434551	1.597813	-1.748643
2997	0.088277	1.638789	-2.193641
2998	1.525155	0.859234	1.505308
2999	-0.979817	0.563954	-1.539394

3000 rows × 3 columns

In [41]:
```
ones = np.ones((df.shape[0],1))
x = df
x = np.hstack((ones,x))
x = np.asarray(x)
x
```

Out[41]:
```
array([[ 1.        , -1.23937466,  0.74910149, -0.52851491],
       [ 1.        , -1.03607028,  0.80143631, -1.28371152],
       [ 1.        , -0.61557914,  1.57952053, -1.39192706],
       ...,
       [ 1.        ,  0.08827657,  1.63878904, -2.19364056],
```

```
       [ 1.        ,  1.52515501,  0.85923429,  1.50530791],
       [ 1.        , -0.97981744,  0.56395376, -1.53939416]])
```

In [42]:
```
x.shape
```

Out[42]:
```
(3000, 4)
```

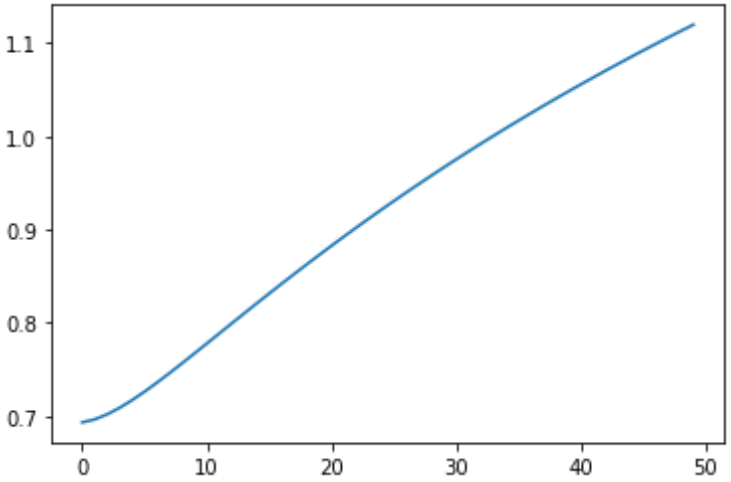
In [46]:
```
y = pd.read_csv("https://raw.githubusercontent.com/simransinghgulati/chemical-segregation/master/Logistic_Y_Train.csv
y = np.array(y)
y = y.reshape((3000,))
```

In [52]:
```
def hypothesis(x,theta):
    z = np.dot(x,theta)
    y_ = sigmoid(z)
    return y_
def sigmoid(z):
    return 1/(1+np.exp(-z))
def error(x,theta,y):
    m = len(x)
    y_ = hypothesis(x,theta)
    err = np.dot(-y,np.log(y_)) - np.dot((1-y),np.log(y_))
    return err/m
def gradient(x,theta,y):
    y_ = hypothesis(x,theta)
    m= len(x)
    grad = np.dot(x.T,y_-y)
    return grad/m
def gradient_des(x,y,learning_rate=0.1,epochs=50):
    m,n = x.shape
    theta = np.zeros((n,))
    err_list = []
    for i in range(epochs):
        err = error(x,theta,y)
        err_list.append(err)
        grad = gradient(x,theta,y)
        theta = theta - learning_rate * grad
    return theta, err_list
```

In [55]:
```
theta, err_list = gradient_des(x,y)
```

In [56]:
```
import matplotlib.pyplot as plt
plt.plot(err_list)
```

Out[56]:
```
[<matplotlib.lines.Line2D at 0x1bb4a8afd90>]
```



In [58]:
```
theta
```

Out[58]:
```
array([-0.10242894, -0.9628054 ,  0.23282612, -1.00459897])
```

In [59]:
```
y_ = hypothesis(x,theta)
y_
```

Out[59]:
```
array([0.85768617, 0.91461367, 0.90519568, ..., 0.91666396, 0.0529986 ,
       0.92544389])
```

In [60]:
```
y_[y_>0.5]=1
y_[y_<0.5] = 0
y_
```

Out[60]:
```
array([1., 1., 1., ..., 1., 0., 1.])
```

In [61]:
```
np.mean(y_==y)*100
```

```
96.73333333333333
```

Out[61]:

In [70]:
```
y_pred = hypothesis(x[8],theta)
y_pred
```

Out[70]: 0.8916807809091882

In [ ]:
```

```