




```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.datasets import make_classification
from sklearn.linear_model import LogisticRegression, SGDClassifier
from mlxtend.plotting import plot_decision_regions
from sklearn.utils import shuffle
```

https://drive.google.com/file/d/1Won6xkyYCCJLJ7eMpVt5VA_4P0tE1nb7/view?usp=sharing

```
!pip install --upgrade --no-cache-dir gdown
!gdown 1Won6xkyYCCJLJ7eMpVt5VA_4P0tE1nb7
```

```
Requirement already satisfied: gdown in /usr/local/lib/python3.10/dist-packages (4.7.3)
Collecting gdown
  Downloading gdown-5.0.0-py3-none-any.whl (16 kB)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.10/dist-packages (from gdown) (4.11.2)
Requirement already satisfied: filelock in /usr/local/lib/python3.10/dist-packages (from gdown) (3.13.1)
Requirement already satisfied: requests[socks] in /usr/local/lib/python3.10/dist-packages (from gdown) (2.31.0)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from gdown) (4.66.1)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.10/dist-packages (from beautifulsoup4->gdown) (2.5)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests[socks]->gdown) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests[socks]->gdown) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests[socks]->gdown) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests[socks]->gdown) (2023.11.17)
Requirement already satisfied: PySocks!=1.5.7,>=1.5.6 in /usr/local/lib/python3.10/dist-packages (from requests[socks]->gdown) (1.7.1)
Installing collected packages: gdown
  Attempting uninstall: gdown
    Found existing installation: gdown 4.7.3
    Uninstalling gdown-4.7.3:
      Successfully uninstalled gdown-4.7.3
  Successfully installed gdown-5.0.0
Downloading...
From: https://drive.google.com/uc?id=1Won6xkyYCCJLJ7eMpVt5VA\_4P0tE1nb7
To: /content/data_banknote_authentication.txt
100% 46.4k/46.4k [00:00<00:00, 72.0MB/s]
```

```
df = pd.read_csv('/content/data_banknote_authentication.txt')
df
```

	x1	x2	x3	x4	y	
0	3.62160	8.66610	-2.8073	-0.44699	0	
1	4.54590	8.16740	-2.4586	-1.46210	0	
2	3.86600	-2.63830	1.9242	0.10645	0	
3	3.45660	9.52280	-4.0112	-3.59440	0	
4	0.32924	-4.45520	4.5718	-0.98880	0	
...	
1367	0.40614	1.34920	-1.4501	-0.55949	1	
1368	-1.38870	-4.87730	6.4774	0.34179	1	
1369	-3.75030	-13.45860	17.5932	-2.77710	1	
1370	-3.56370	-8.38270	12.3930	-1.28230	1	
1371	-2.54190	-0.65804	2.6842	1.19520	1	

1372 rows × 5 columns

```
shuffled_data = shuffle(df)
shuffled_data.to_csv('created_data.csv', index=False)
print(shuffled_data)
```

	x1	x2	x3	x4	y
922	-1.41740	-2.2535	1.51800	0.61981	1
461	3.09480	8.7324	-2.90070	-0.96682	0
35	2.43910	6.4417	-0.80743	-0.69139	0
1009	-3.57130	-12.4922	14.88810	-0.47027	1
263	4.98800	7.2052	-3.28460	-1.16080	0

```

...      ...      ...      ...      ...
1258 -0.62043  0.5587 -0.38587 -0.66423  1
370  2.25260  9.9636 -3.17490 -2.99440  0
1124 -1.75890 -6.4624  8.47730  0.31981  1
606  -1.42750 11.8797  0.41613 -6.99780  0
485   3.65750  7.2797 -2.26920 -1.14400  0

```

[1372 rows x 5 columns]

```
df1 = pd.read_csv('/content/created_data.csv')
df1
```

	x1	x2	x3	x4	y
0	-1.41740	-2.2535	1.51800	0.61981	1
1	3.09480	8.7324	-2.90070	-0.96682	0
2	2.43910	6.4417	-0.80743	-0.69139	0
3	-3.57130	-12.4922	14.88810	-0.47027	1
4	4.98800	7.2052	-3.28460	-1.16080	0
...
1367	-0.62043	0.5587	-0.38587	-0.66423	1
1368	2.25260	9.9636	-3.17490	-2.99440	0
1369	-1.75890	-6.4624	8.47730	0.31981	1
1370	-1.42750	11.8797	0.41613	-6.99780	0
1371	3.65750	7.2797	-2.26920	-1.14400	0

1372 rows x 5 columns

Logistic Regression (from Scratch)

```
def sigmoid(x):
    return 1 / (1 + np.exp(-x))
```

```
def logistic_regression(x, w):
    y_hat = sigmoid(x @ w)
    return y_hat
```

Binary Cross Entropy (BCE)

```
def bce(y, y_hat):
    loss = -(np.mean(y*np.log(y_hat) + (1-y)*np.log(1-y_hat)))
    return loss
```

Gradient

```
def gradient(x, y, y_hat):
    grads = (x.T @ (y_hat - y)) / len(y)
    return grads
```

Gradient Descent

```
def gradient_descent(w, eta, grads):
    w -= eta*grads
    return w
```

Accuracy

```
def accuracy(y, y_hat):
    acc = np.sum(y == np.round(y_hat)) / len(y)
    return acc
```

آموزش داده های validatin

```
X = df1[['x1','x2','x3','x4']].values
y = df1[['y']].values
X, y

(array([[ -1.4174 , -2.2535 ,  1.518 ,  0.61981],
        [ 3.0948 ,  8.7324 , -2.9007 , -0.96682],
        [ 2.4391 ,  6.4417 , -0.80743, -0.69139],
        ...,
        [-1.7589 , -6.4624 ,  8.4773 ,  0.31981],
        [-1.4275 , 11.8797 ,  0.41613, -6.9978 ],
        [ 3.6575 ,  7.2797 , -2.2692 , -1.144 ]]),
 array([[1],
        [0],
        [0],
        ...,
        [1],
        [0],
        [0]]))

تقسیم داده ها به دو دسته آموزش و اعتبارسنجی

x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

x_train.shape, x_test.shape, y_train.shape, y_test.shape

((1097, 4), (275, 4), (1097, 1), (275, 1))

y_hat = logistic_regression(x_test, np.random.randn(4, 1))
print(y_hat.shape)

(275, 1)

x_test = np.hstack((np.ones((len(x_test), 1)), x_test))
x_test.shape

(275, 5)

m = 4
w = np.random.randn(m+1, 1)
print(w.shape)

(5, 1)

eta = 0.01
n_epochs = 10000 #N

error_hist = []

for epoch in range(n_epochs):
    # predictions
    y_hat = logistic_regression(x_test, w)

    # loss
    e = bce(y_test, y_hat)
    error_hist.append(e)

    # gradients
    grads = gradient(x_test, y_test, y_hat)

    # gradient descent
    w = gradient_descent(w, eta, grads)

    if (epoch+1) % 1 == 0:
        print(f'Epoch={epoch}, \t E={e:.4}, \t w={w.T[0]}')
```

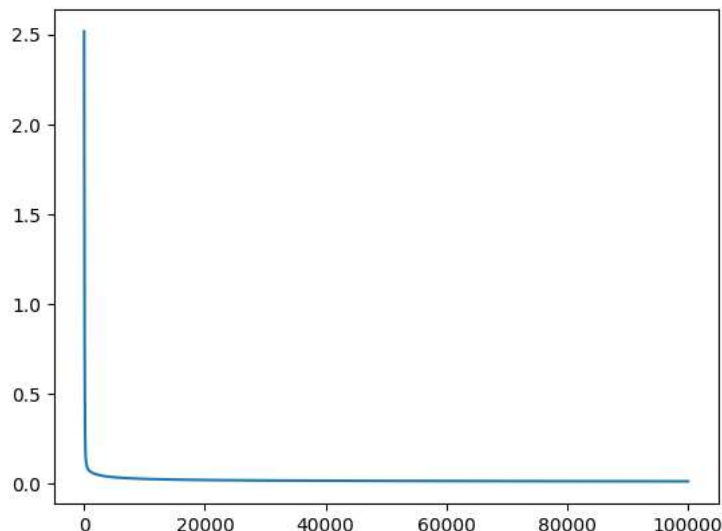
```

epoch=99948, E=0.01314, W=[ 4.192223443 -4.01696789 -2.37549781 -2.80133189 -0.2142698 ]
Epoch=99949, E=0.01314, W=[ 4.19224392 -4.01697888 -2.37550354 -2.8013389 -0.2142706 ]
Epoch=99950, E=0.01314, W=[ 4.19225341 -4.01698986 -2.37550922 -2.80134591 -0.2142714 ]
Epoch=99951, E=0.01314, W=[ 4.1922629 -4.01700085 -2.37551489 -2.80135292 -0.2142722 ]
Epoch=99952, E=0.01314, W=[ 4.19227239 -4.01701184 -2.37552056 -2.80135992 -0.21427301 ]
Epoch=99953, E=0.01314, W=[ 4.19228188 -4.01702283 -2.37552623 -2.80136693 -0.21427381 ]
Epoch=99954, E=0.01314, W=[ 4.19229137 -4.01703381 -2.37553191 -2.80137394 -0.21427461 ]
Epoch=99955, E=0.01314, W=[ 4.19230086 -4.0170448 -2.37553758 -2.80138095 -0.21427541 ]
Epoch=99956, E=0.01314, W=[ 4.19231035 -4.01705579 -2.37554325 -2.80138796 -0.21427621 ]
Epoch=99957, E=0.01314, W=[ 4.19231984 -4.01706677 -2.37554892 -2.80139496 -0.21427701 ]
Epoch=99958, E=0.01314, W=[ 4.19232933 -4.01707776 -2.3755546 -2.80140197 -0.21427781 ]
Epoch=99959, E=0.01314, W=[ 4.19233882 -4.01708875 -2.37556027 -2.80140898 -0.21427861 ]
Epoch=99960, E=0.01314, W=[ 4.19234831 -4.01709973 -2.37556594 -2.80141599 -0.21427941 ]
Epoch=99961, E=0.01314, W=[ 4.19235779 -4.01711072 -2.37557161 -2.80142299 -0.21428021 ]
Epoch=99962, E=0.01314, W=[ 4.19236728 -4.01712171 -2.37557729 -2.80143 -0.21428101 ]
Epoch=99963, E=0.01314, W=[ 4.19237677 -4.01713269 -2.37558296 -2.80143701 -0.21428181 ]
Epoch=99964, E=0.01314, W=[ 4.19238626 -4.01714368 -2.37558863 -2.80144401 -0.21428262 ]
Epoch=99965, E=0.01314, W=[ 4.19239575 -4.01715467 -2.3755943 -2.80145102 -0.21428342 ]
Epoch=99966, E=0.01314, W=[ 4.19240524 -4.01716565 -2.37559997 -2.80145803 -0.21428422 ]
Epoch=99967, E=0.01314, W=[ 4.19241473 -4.01717664 -2.37560565 -2.80146504 -0.21428502 ]
Epoch=99968, E=0.01314, W=[ 4.19242422 -4.01718762 -2.37561132 -2.80147204 -0.21428582 ]
Epoch=99969, E=0.01314, W=[ 4.19243371 -4.01719861 -2.37561699 -2.80147905 -0.21428662 ]
Epoch=99970, E=0.01314, W=[ 4.19244319 -4.0172096 -2.37562266 -2.80148606 -0.21428742 ]
Epoch=99971, E=0.01314, W=[ 4.19245268 -4.01722058 -2.37562833 -2.80149306 -0.21428822 ]
Epoch=99972, E=0.01314, W=[ 4.19246217 -4.01723157 -2.37563401 -2.80150007 -0.21428902 ]
Epoch=99973, E=0.01314, W=[ 4.19247166 -4.01724255 -2.37563968 -2.80150708 -0.21428982 ]
Epoch=99974, E=0.01314, W=[ 4.19248115 -4.01725354 -2.37564535 -2.80151408 -0.21429062 ]
Epoch=99975, E=0.01314, W=[ 4.19249064 -4.01726452 -2.37565102 -2.80152109 -0.21429142 ]
Epoch=99976, E=0.01314, W=[ 4.19250013 -4.01727551 -2.37565669 -2.8015281 -0.21429222 ]
Epoch=99977, E=0.01314, W=[ 4.19250961 -4.01728649 -2.37566236 -2.8015351 -0.21429303 ]
Epoch=99978, E=0.01314, W=[ 4.1925191 -4.01729748 -2.37566804 -2.80154211 -0.21429383 ]
Epoch=99979, E=0.01314, W=[ 4.19252859 -4.01730847 -2.37567371 -2.80154912 -0.21429463 ]
Epoch=99980, E=0.01314, W=[ 4.19253808 -4.01731945 -2.37567938 -2.80155612 -0.21429543 ]
Epoch=99981, E=0.01314, W=[ 4.19254757 -4.01733044 -2.37568505 -2.80156313 -0.21429623 ]
Epoch=99982, E=0.01314, W=[ 4.19255705 -4.01734142 -2.37569072 -2.80157013 -0.21429703 ]
Epoch=99983, E=0.01314, W=[ 4.19256654 -4.01735241 -2.37569639 -2.80157714 -0.21429783 ]
Epoch=99984, E=0.01314, W=[ 4.19257603 -4.01736339 -2.37570206 -2.80158415 -0.21429863 ]
Epoch=99985, E=0.01314, W=[ 4.19258552 -4.01737438 -2.37570773 -2.80159115 -0.21429943 ]
Epoch=99986, E=0.01314, W=[ 4.192595 -4.01738536 -2.37571341 -2.80159816 -0.21430023 ]
Epoch=99987, E=0.01314, W=[ 4.19260449 -4.01739634 -2.37571908 -2.80160516 -0.21430103 ]
Epoch=99988, E=0.01314, W=[ 4.19261398 -4.01740733 -2.37572475 -2.80161217 -0.21430183 ]
Epoch=99989, E=0.01314, W=[ 4.19262347 -4.01741831 -2.37573042 -2.80161918 -0.21430263 ]
Epoch=99990, E=0.01314, W=[ 4.19263295 -4.0174293 -2.37573609 -2.80162618 -0.21430343 ]
Epoch=99991, E=0.01314, W=[ 4.19264244 -4.01744028 -2.37574176 -2.80163319 -0.21430423 ]
Epoch=99992, E=0.01314, W=[ 4.19265193 -4.01745127 -2.37574743 -2.80164019 -0.21430503 ]
Epoch=99993, E=0.01314, W=[ 4.19266141 -4.01746225 -2.3757531 -2.8016472 -0.21430584 ]
Epoch=99994, E=0.01314, W=[ 4.1926709 -4.01747324 -2.37575877 -2.8016542 -0.21430664 ]
Epoch=99995, E=0.01314, W=[ 4.19268039 -4.01748422 -2.37576444 -2.80166121 -0.21430744 ]
Epoch=99996, E=0.01314, W=[ 4.19268988 -4.0174952 -2.37577011 -2.80166822 -0.21430824 ]
Epoch=99997, E=0.01314, W=[ 4.19269936 -4.01750619 -2.37577578 -2.80167522 -0.21430904 ]
Epoch=99998, E=0.01314, W=[ 4.19270885 -4.01751717 -2.37578146 -2.80168223 -0.21430984 ]
Epoch=99999, E=0.01314, W=[ 4.19271834 -4.01752816 -2.37578713 -2.80168923 -0.21431064 ]

```

```
plt.plot(error_hist)
```

```
[<matplotlib.lines.Line2D at 0x79df15a48370>]
```



```

y_hat = logistic_regression(x_test, w)
accuracy(y_test, y_hat)

```

0.9927272727272727

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.