In [307…

```
from sklearn.datasets import make_classification
import numpy as np
X, y = make_classification(n_samples=100, n_features=2, n_informative=1,n_redun
                           n_classes=2, n_clusters_per_class=1, random_state=41
```

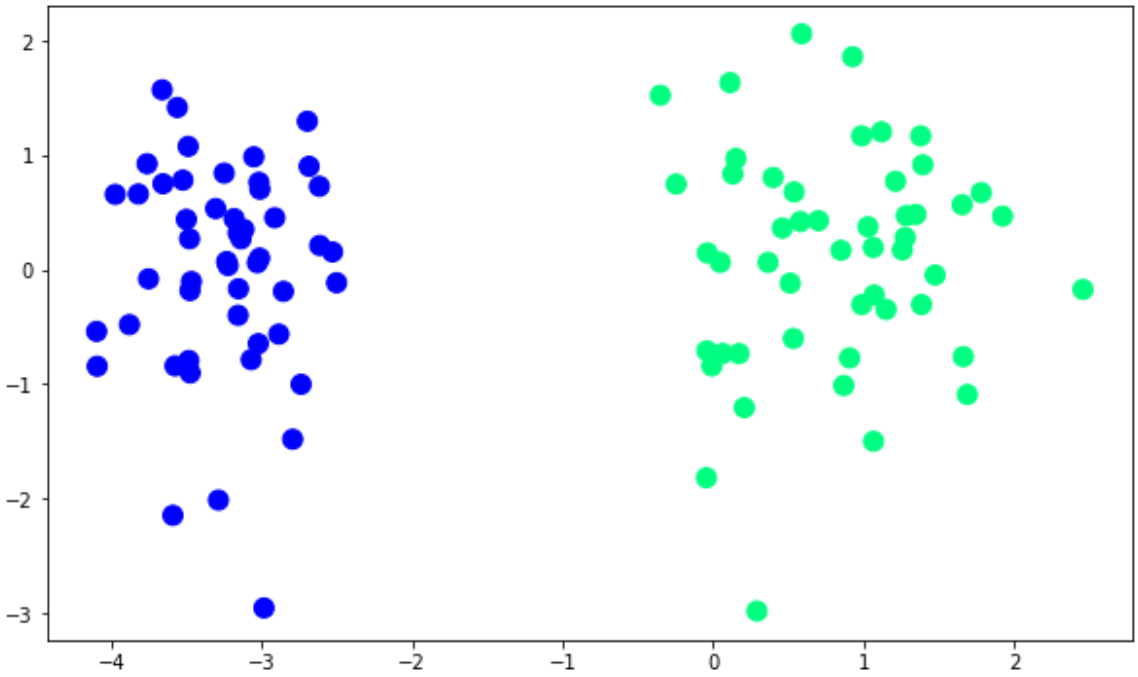
In [308…

```
import matplotlib.pyplot as plt
```

In [309…

```
plt.figure(figsize=(10,6))
plt.scatter(X[:,0],X[:,1],c=y,cmap='winter',s=100)
```

Out[309…



In [310…

```
from sklearn.linear_model import LogisticRegression
lor = LogisticRegression(penalty='none',solver='sag')
lor.fit(X,y)
```

```
C:\Users\91842\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:328: Con
vergenceWarning: The max_iter was reached which means the coef_ did not converge
  warnings.warn("The max_iter was reached which means "
```

Out[310…

```
LogisticRegression(penalty='none', solver='sag')
```

In [311…

```
print(lor.coef_)
print(lor.intercept_)
```

```
[[4.7808362 0.2062583]]
[5.7492783]
```

In [312…

```
m1 = -(lor.coef_[0][0]/lor.coef_[0][1])
b1 = -(lor.intercept_/lor.coef_[0][1])
```

In [313…
```
x_input = np.linspace(-3,3,100)
y_input = m1*x_input + b1
```

In [329…
```
def gd(X,y):

    X = np.insert(X,0,1,axis=1)
    weights = np.ones(X.shape[1])
    lr = 0.5

    for i in range(5000):
        y_hat = sigmoid(np.dot(X,weights))
        weights = weights + lr*(np.dot((y-y_hat),X)/X.shape[0])

    return weights[1:],weights[0]
```

In [330…
```
def sigmoid(z):
    return 1/(1 + np.exp(-z))
```

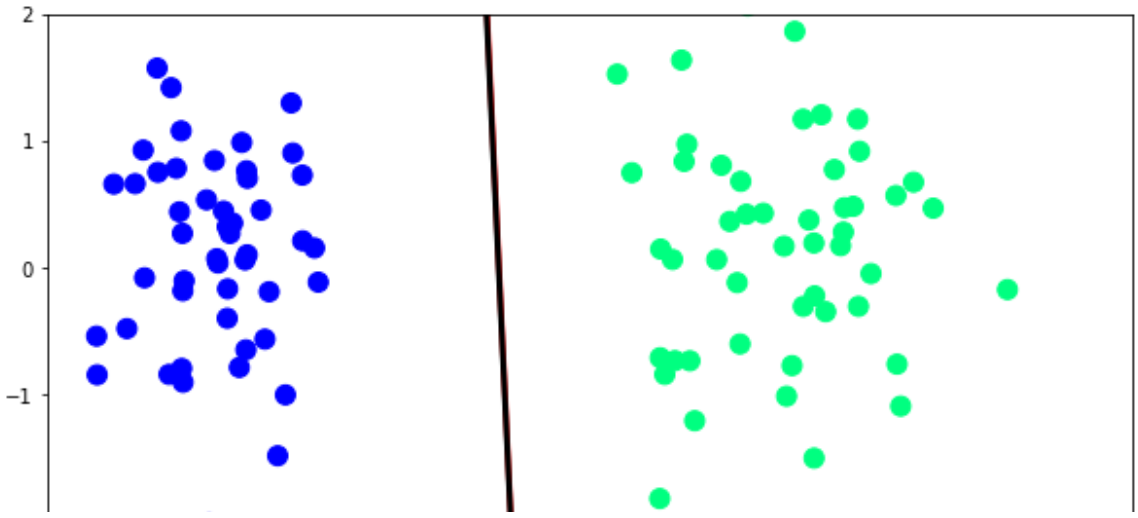
In [331…
```
coef_,intercept_ = gd(X,y)
```

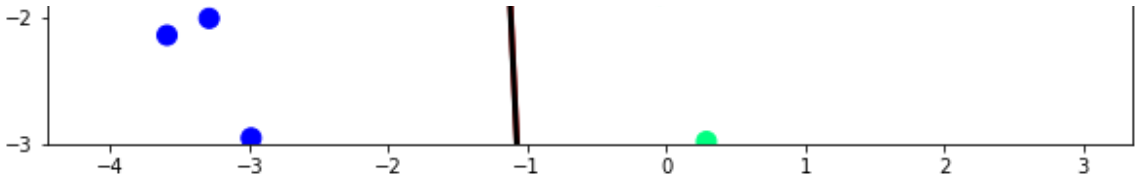
In [332…
```
m = -(coef_[0]/coef_[1])
b = -(intercept_/coef_[1])
```

In [333…
```
x_input1 = np.linspace(-3,3,100)
y_input1 = m*x_input1 + b
```

In [334…
```
plt.figure(figsize=(10,6))
plt.plot(x_input,y_input,color='red',linewidth=3)
plt.plot(x_input1,y_input1,color='black',linewidth=3)
plt.scatter(X[:,0],X[:,1],c=y,cmap='winter',s=100)
plt.ylim(-3,2)
```

Out[334… (-3.0, 2.0)





In [ ]:

In [ ]:

In [315…
```
np.insert(X,0,1,axis=1)
```

Out[315…
```
array([[ 1.        ,  0.51123145, -0.11697552],
       [ 1.        ,  0.06316371, -0.73115232],
       [ 1.        , -0.0425064 , -0.7081059 ],
       [ 1.        , -3.2891569 , -2.01199214],
       [ 1.        ,  0.1111445 ,  1.63493163],
       [ 1.        , -2.53070306,  0.15599044],
       [ 1.        , -3.49036198,  1.07782053],
       [ 1.        ,  0.3976447 ,  0.80626713],
       [ 1.        , -0.24666899,  0.74859527],
       [ 1.        , -3.65803446,  0.75152794],
       [ 1.        , -3.47658131, -0.90114581],
       [ 1.        , -3.47815037, -0.1815243 ],
       [ 1.        ,  0.29004249, -2.98092432],
       [ 1.        ,  1.11761831,  1.20500136],
       [ 1.        , -3.52530398,  0.78302407],
       [ 1.        ,  0.69929128,  0.42968688],
       [ 1.        ,  0.17089733, -0.73229726],
       [ 1.        , -3.57785124, -0.83930476],
       [ 1.        ,  0.12965489,  0.83727062],
       [ 1.        , -3.46888717, -0.10255323],
       [ 1.        , -3.97487212,  0.65867001],
       [ 1.        , -3.76348686,  0.92649819],
       [ 1.        , -3.01519735,  0.10216193],
       [ 1.        ,  1.92241659,  0.46886454],
       [ 1.        , -2.91479578,  0.45432938],
       [ 1.        ,  0.9259563 ,  1.8613386 ],
       [ 1.        , -3.4859014 , -0.79255991],
       [ 1.        , -2.73978345, -1.0004391 ],
       [ 1.        , -4.09896768, -0.53814137],
       [ 1.        , -3.50212636,  0.44027716],
       [ 1.        , -3.13904797,  0.27047889],
       [ 1.        ,  1.66188378, -0.75869267],
       [ 1.        ,  0.53718567,  0.6802322 ],
       [ 1.        ,  0.84886927,  0.17018845],
       [ 1.        ,  0.86681859, -1.01121977],
       [ 1.        , -0.00979785, -0.8394709 ],
       [ 1.        , -2.68650432,  0.90327412],
       [ 1.        ,  0.98593205,  1.16981747],
       [ 1.        , -3.30742775,  0.53461406],
       [ 1.        ,  1.06925594, -0.22100631],
       [ 1.        , -3.0242056 , -0.64584571],
       [ 1.        , -2.61612476,  0.21243302],
       [ 1.        , -3.05407331,  0.98654083],
       [ 1.        ,  1.78068059,  0.67382928],
```

```
[ 1.        , -2.50369897, -0.11323563],
[ 1.        ,  1.39258216,  0.91694693],
[ 1.        ,  0.45865727,  0.3645213 ],
[ 1.        , -3.07153732, -0.78470874],
[ 1.        , -2.79571273, -1.48004137],
[ 1.        , -2.6184666 ,  0.72931763],
[ 1.        ,  0.53218913, -0.60000139],
[ 1.        , -0.0395452 ,  0.14720034],
[ 1.        , -3.11964851,  0.35215601],
[ 1.        ,  1.2561371 ,  0.17388705],
[ 1.        ,  2.45686591, -0.17162755],
[ 1.        , -3.2252872 ,  0.03763193],
[ 1.        ,  1.3467772 ,  0.48360467],
[ 1.        , -3.75380319, -0.07909256],
[ 1.        , -3.03070579,  0.06158897],
[ 1.        ,  0.20583358, -1.20413846],
[ 1.        ,  1.21108994,  0.77324391],
[ 1.        , -3.01981877,  0.7640092 ],
[ 1.        , -3.66318854,  1.57200032],
[ 1.        ,  1.28298186,  0.47296516],
[ 1.        ,  0.3640611 ,  0.0642074 ],
[ 1.        , -0.35195584,  1.52469135],
[ 1.        , -3.59205904, -2.1450828 ],
[ 1.        ,  0.9061286 , -0.7701428 ],
[ 1.        , -3.25090707,  0.84455766],
[ 1.        , -2.69817214,  1.29753064],
[ 1.        , -4.09493435, -0.84276652],
[ 1.        ,  0.58735838,  2.0615874 ],
[ 1.        ,  1.3843216 , -0.3036533 ],
[ 1.        , -3.15603064, -0.1656085 ],
[ 1.        ,  1.37865857,  1.16969713],
[ 1.        ,  0.98696744, -0.3038555 ],
[ 1.        , -3.88083743, -0.4788978 ],
[ 1.        ,  1.27593515,  0.28128262],
[ 1.        , -3.01387754,  0.70412341],
[ 1.        ,  1.0278443 ,  0.3757601 ],
[ 1.        ,  0.15011746,  0.97247545],
[ 1.        ,  1.653853  ,  0.56868381],
[ 1.        , -3.15852226, -0.39767934],
[ 1.        ,  1.68843893, -1.08933729],
[ 1.        , -0.04572327, -1.81663939],
[ 1.        , -3.56234524,  1.41775924],
[ 1.        , -3.48093864,  0.27098296],
[ 1.        , -2.88757001, -0.56186639],
[ 1.        ,  0.57853245,  0.42168159],
[ 1.        , -2.98638242, -2.95373116],
[ 1.        ,  1.06361882,  0.19492133],
[ 1.        ,  1.47452652, -0.04578681],
[ 1.        , -2.8571924 , -0.18860967],
[ 1.        , -3.18522871,  0.44635682],
[ 1.        ,  1.06491218, -1.49814775],
[ 1.        , -3.82161901,  0.66198774],
[ 1.        , -3.23429001,  0.07153265],
[ 1.        , -3.15863898,  0.32323638],
[ 1.        ,  1.14824982, -0.34574201],
[ 1.        ,  0.04627774,  0.06499922]])
```

In [316…

```
X1 = np.insert(X,0,1,axis=1)
np.ones(X1.shape[1])
```

Out[316… array([1., 1., 1.])

In [ ]: