(29) WhatsApp | Home - Google Drive | Untitled1.ipynb - Colaboratory

colab.research.google.com/drive/1LLu_hx1TtjdH4JTCKOA5KlFiMONC2wSk#scrollTo=CGgGJGokcmaA

YouTube Maps

Untitled1.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Colab AI

```
[1] from sklearn.datasets import load_iris
    import numpy as np
    import matplotlib.pyplot as plt
    %matplotlib inline
```

```
[2] # flowers dataset tes of three different species of the flowers
    iris = load_iris()
    print(iris.feature_names)
```

```
['sepal length (cm)', 'sepal width (cm)', 'petal length (cm)', 'petal width (cm)']
```

```
[3] # three classes of flowers
    print(iris.target_names)
```

```
['setosa' 'versicolor' 'virginica']
```

```
[4] # features
    x = iris.data
    #labels
    y = iris.target
    print(x.shape,y.shape)
    #150 rows with 4 columns
```

```
(150, 4) (150,)
```

```
[5] x[:5]
```

0s completed at 22:57

28°C Sunny

Search

ENG IN 22:59 17-03-2024

(29) WhatsApp | Home - Google Drive | Untitled1.ipynb - Colaboratory

colab.research.google.com/drive/1LLu_hx1TtjdH4JTCKOA5KlFiMONC2wSk#scrollTo=CGgGJGokcmaA

YouTube Maps

Untitled1.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Colab AI

```
[5] x[:5]
```

```
array([[5.1, 3.5, 1.4, 0.2],
       [4.9, 3. , 1.4, 0.2],
       [4.7, 3.2, 1.3, 0.2],
       [4.6, 3.1, 1.5, 0.2],
       [5. , 3.6, 1.4, 0.2]])
```

```
[6] y[:5]
```

```
array([0, 0, 0, 0, 0])
```

```
np.unique(y)
```

```
array([0, 1, 2])
```

```
[9] # create the boolean mask which gives the all flowers
mask = np.where((y==0) | (y==1))
x = x[mask][:,[0,2]]
y = y[mask]
```

```
[10] print(np.unique(y))
```

```
[0 1]
```

```
[11] print(x.shape,y.shape)
```

0s completed at 22:57

28°C Sunny

Search

ENG IN

23:00 17-03-2024

(29) WhatsApp | Home - Google Drive | Untitled1.ipynb - Colaboratory

colab.research.google.com/drive/1LLu_hx1TtjdH4JTCKOA5KlFiMONC2wSk#scrollTo=CGgGJGokcmaA

YouTube Maps

Untitled1.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

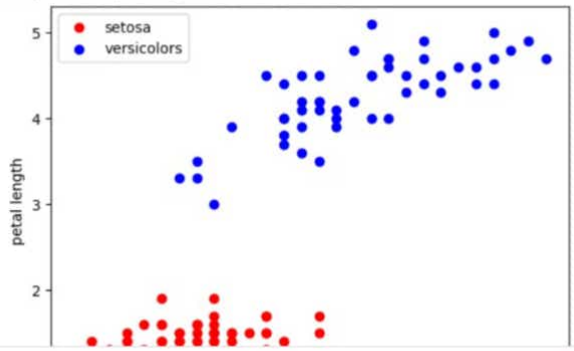
+ Code + Text

RAM Disk Colab AI

```
[12] setosas = x[np.where(y==0)]
     versicolors = x[np.where(y==1)]
```

```
# linearly separable data
plt.scatter(setosas[:,0],setosas[:,1],color ='r',label='setosa')
plt.scatter(versicolors[:,0],versicolors[:,1],color ='b',label='versicolors')
plt.xlabel('sepal length')
plt.ylabel('petal length')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7de6780f2b90>
```



0s completed at 22:57

28°C Sunny

Search

ENG IN 23:00 17-03-2024

(29) WhatsApp | Home - Google Drive | Untitled1.ipynb - Colaboratory

colab.research.google.com/drive/1LLu_hx1TtjdH4JTCKOA5KlFiMONC2wSk#scrollTo=CGgGJGokcmaA

YouTube Maps

Untitled1.ipynb

File Edit View Insert Runtime Tools Help All changes saved

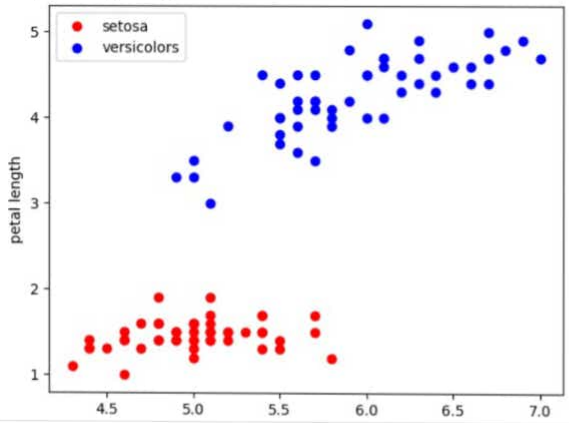
Comment Share

+ Code + Text

RAM Disk Colab AI

```
# linearly separable data
plt.scatter(setosas[:,0],setosas[:,1],color ='r',label='setosa')
plt.scatter(versicolors[:,0],versicolors[:,1],color ='b',label='versicolors')
plt.xlabel('sepal length')
plt.ylabel('petal length')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7de6780f2b90>
```



0s completed at 22:57

28°C Sunny

Search

ENG IN

23:00

(29) WhatsApp | Home - Google Drive | Untitled1.ipynb - Colaboratory

colab.research.google.com/drive/1LLu_hx1TtjdH4JTCKOA5KlFiMONC2wSk#scrollTo=CGgGJGokcmaA

YouTube Maps

Untitled1.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Colab AI

```
[14] print(x.shape,y.shape)
```

```
(100, 2) (100,)
```

```
import sys

class Perceptron(object):

    def __init__(self,input_dim):
       self.weights = np.random.normal(loc=0.0,scale=1.0,size=input_dim)
       self.bias = np.random.normal(loc= 0.0,scale =1.0,size=1)

    def activation(self,pred):
      return np.where(pred>=0,1,0)

    def predict(self,data):
      return self.activation(np.dot(data,self.weights.T))

    def train(self,input_data,targets,epochs,lr=0.01):
      losses = []
      for e in range(1,epochs + 1):
        epoch_loss = 0
        for data, target in zip(input_data,targets):
          pred = self.predict(data)
          error = target - pred
          update = lr * error
          epoch_loss += error ** 2
```

0s completed at 22:57

28°C Sunny

Search

ENG IN

23:00 17-03-2024

(29) WhatsApp | Home - Google Drive | Untitled1.ipynb - Colaboratory

colab.research.google.com/drive/1LLu_hx1TtjdH4JTCKOA5KlFiMONC2wSk#scrollTo=CGgGJGokcmaA

YouTube Maps

Untitled1.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Colab AI

[31]

```
def train(self,input_data,targets,epochs,lr=0.01):
  losses = []
  for e in range(1,epochs + 1):
    epoch_loss = 0
    for data, target in zip(input_data,targets):
      pred = self.predict(data)
      error = target - pred
      update = lr * error
      epoch_loss += error ** 2
      self.weights =+ update * data
      self.bias += update
  losses.append(epoch_loss)


  print(f"\rEpoch {e} / {epochs}, Loss: {epoch_loss}",end="")
  sys.stdout.flush()

  return losses
```

[32]

```
p = Perceptron(input_dim=2)
losses = p.train(x,y ,epochs = 50)
```

```
Epoch 50 / 50, Loss: 25
```

[36]

```
print(p.weights,p.bias)
```

```
[0. 0.] [-13.68894092]
```

0s completed at 22:57

28°C Sunny

Search

ENG IN 23:00 17-03-2024