**Code:**

import numpy as np

import numpy.random

import matplotlib.pyplot as plt

from sklearn.datasets import make\_classification

import math

def generate\_data\_set\_with\_labels(n):

'''

generates random data set of size n by sampling points uniformly

at random within the unit sphere in the Euclidean plane and labels

the points according to a randomly chosen linear decision boundary

'''

# generate two points which will represent linear decision boundary

data\_set = []

labels = []

r = np.random.uniform(0,1.0, 2)

theta = np.random.uniform(0,2 \* math.pi, 2)

x1 = r[0] \* math.cos(theta[0])

x2 = r[1] \* math.cos(theta[1])

y1 = r[0] \* math.sin(theta[0])

y2 = r[1] \* math.sin(theta[1])

m = (y1-y2)/(x1-x2)

for i in range (n):

r = np.random.uniform(0,1.0)

theta = np.random.uniform(0,2 \* math.pi)

x = r \* math.cos(theta)

y = r \* math.sin(theta)

data\_set.append(np.array([x,y]))

if x - x1 > m\*(y - y1):

labels.append(-1)

else:

labels.append(1)

np\_data\_set = np.array(data\_set)

np\_labels = np.array(labels)

return np\_data\_set, np\_labels

def plot\_data(data, labels):

plt.figure()

t = np.linspace(0,math.pi\*2,100)

plt.plot(np.cos(t), np.sin(t), linewidth=1)

plt.scatter(data[:, 0], data[:, 1], c=labels)

plt.show()

def plot\_mistakes(mistakes):

plt.figure()

lst = []

for i in range(100):

lst.append(i)

plt.scatter(lst, mistakes)

plt.show()

def perceptron\_algorithm(data, labels):

w = np.zeros((100, 2))

t = 0

mistakes = []

mistakes\_counter = 0

for x\_t in data:

p\_t = np.sign(np.dot(w[t],x\_t))

y\_t = labels[t]

if p\_t != y\_t:

mistakes\_counter += 1

mistakes.append(mistakes\_counter)

if t+1<100:

if p\_t\*y\_t <= 0:

w[t+1] = w[t] + y\_t\*x\_t

else:

w[t+1] = w[t]

t += 1

return mistakes

data, labels = generate\_data\_set\_with\_labels(n=100)

print(data.size, labels.size, np.sign(-2))

plot\_data(data, labels)

mistakes = perceptron\_algorithm(data, labels)

plot\_mistakes(mistakes)

Chart, scatter chart

Description automatically generated

**Mistakes:**

Chart, scatter chart

Description automatically generated