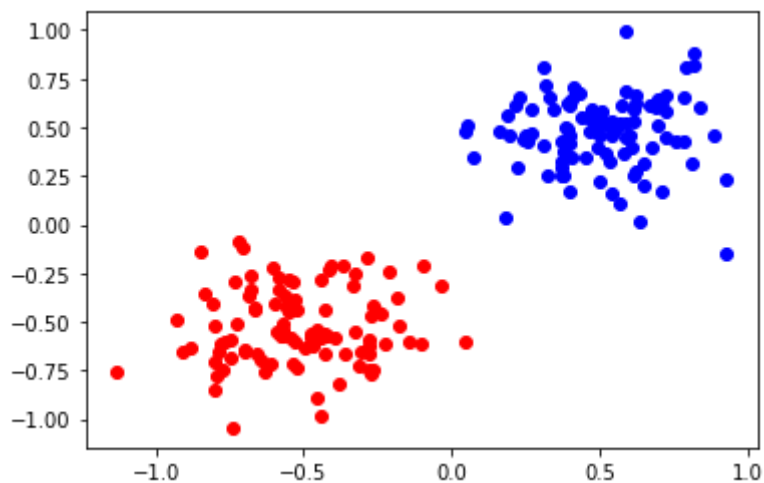


```
In [298...
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Question 5.1

Synthetic Dataset 1: Variance = 0.2

```
In [299...
v1 = np.random.normal(loc=( 0.5, 0.5),scale=0.2,size=(100,2))
v2 = np.random.normal(loc=(-0.5,-0.5),scale=0.2,size=(100,2))
d1 = pd.DataFrame({'x1':[v[0] for v in v1], 'x2':[v[1] for v in v1], 'y':[1]*len(v1)})
d2 = pd.DataFrame({'x1':[v[0] for v in v2], 'x2':[v[1] for v in v2], 'y':[-1]*len(v2)})
all_data = pd.concat([d1,d2]).sample(frac=1)
all_data.head()
plt.scatter([x for x,y in v1],[y for x,y in v1],color='blue')
plt.scatter([x for x,y in v2],[y for x,y in v2],color='red')
plt.show()
```



Question 5.2

```
In [300...
X = np.array([[list(all_data.x1)[i],list(all_data.x2)[i]] for i in range(len(all_data.x1))])
y = np.array(all_data.y)

def perceptron(X, Y, variance):
    w = np.zeros(len(X[0]))
    errors = []
    epochs = 100
    eta = 1

    for t in range(epochs):
        error_count = 0
        for i, x in enumerate(X):
            if (np.dot(X[i], w)*Y[i]) <= 0:
                error_count += 1
                w = w + eta*X[i]*Y[i]
        errors.append(error_count)
        if error_count == 0:
```

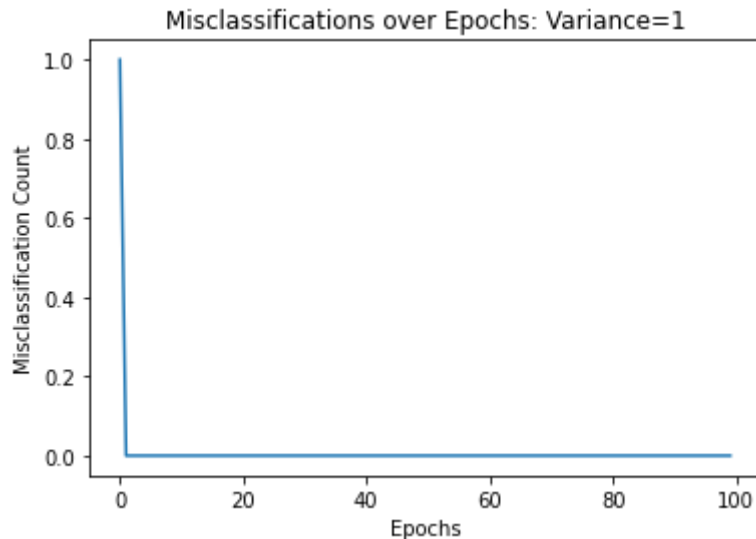
pass

```
plt.plot(range(t+1),errors)
plt.xlabel('Epochs')
plt.ylabel('Misclassification Count')
plt.title('Misclassifications over Epochs: Variance={}'.format(variance))
plt.show()
```

Convergence -- eta = 1

In [301...

```
perceptron(X,y,variance=1)
```



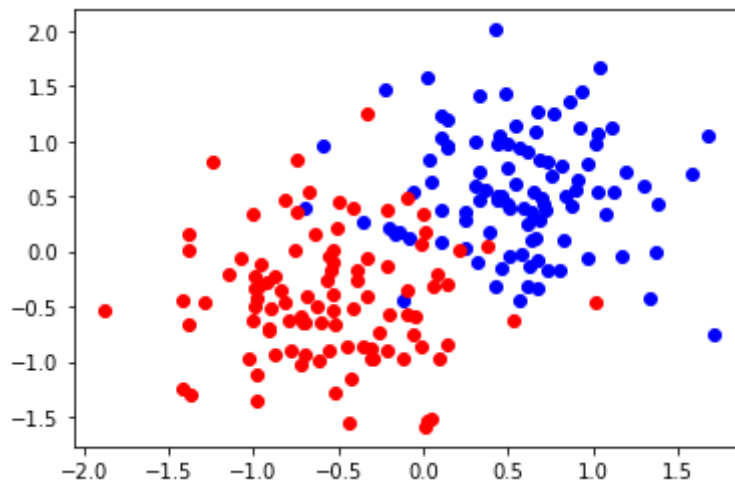
Convergence happens immediately when eta = 1.

Question 5.3

Synthetic Dataset #2 -- Variance = 0.4

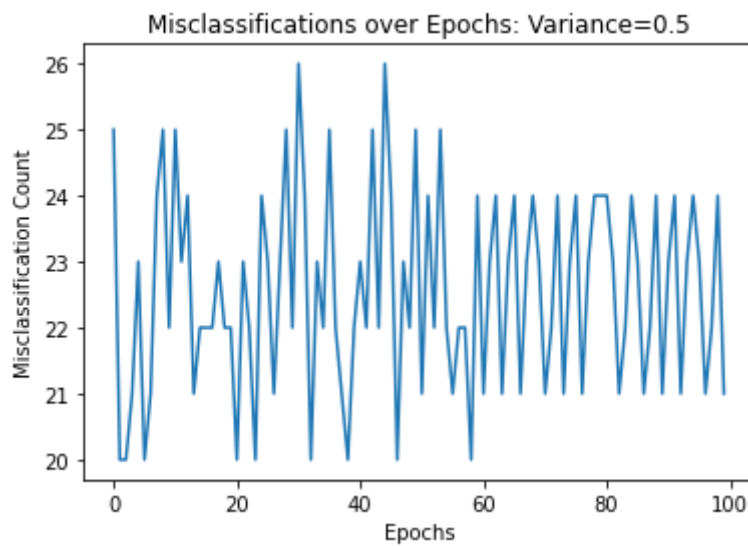
In [302...

```
v1 = np.random.normal(loc=( 0.5, 0.5),scale=0.5,size=(100,2))
v2 = np.random.normal(loc=(-0.5,-0.5),scale=0.5,size=(100,2))
d1 = pd.DataFrame({'x1':[v[0] for v in v1],'x2':[v[1] for v in v1],'y':[1]*len(v1)})
d2 = pd.DataFrame({'x1':[v[0] for v in v2],'x2':[v[1] for v in v2],'y':[-1]*len(v2)})
all_data = pd.concat([d1,d2]).sample(frac=1)
all_data.head()
plt.scatter([x for [x,y] in v1],[y for [x,y] in v1],color='blue')
plt.scatter([x for [x,y] in v2],[y for [x,y] in v2],color='red')
plt.show()
```



In [303...

```
X = np.array([[list(all_data.x1)[i], list(all_data.x2)[i]] for i in range(len(all_data.x1))])
y = np.array(all_data.y)
perceptron(X, y, variance=0.5)
```



Convergence does not occur.