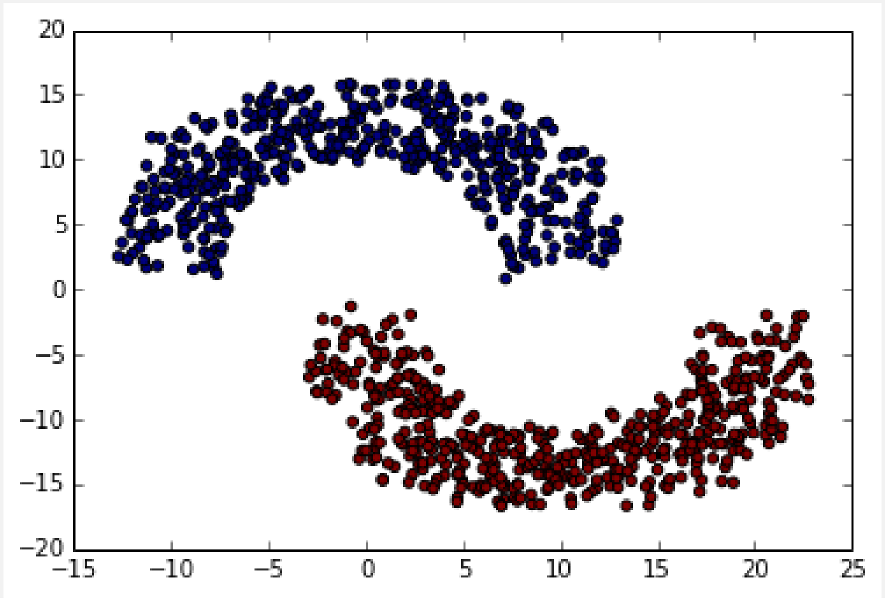
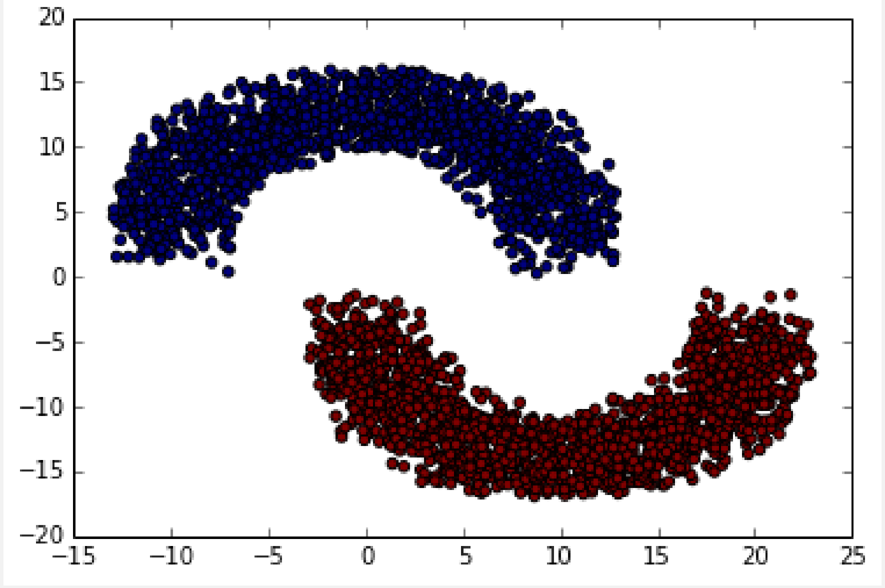
```python
from sklearn.datasets import make_moons
from random import uniform

def moons(samples = 1000, d=0, r=10, w=6, n=None):

    points, moon = make_moons(n_samples=samples, shuffle=False, noise=n, random_state=
        None)

    #scale with r
    for p in points:
        p[0] = p[0] *r
        p[1] = p[1] *r


    #move: with d
    for i in range(len(points)):
        if i < len(points)/2:
            points[i][1] = points[i][1]+0.5*w
        else:
            points[i][1] = points[i][1]-d-w-0.5*w


    #widen with w
    for p in points:
        p[0] = p[0] + uniform(-w/2, w/2)
        p[1] = p[1] + uniform(-w/2, w/2)

    return points, moon
```

```python
from sklearn.svm import SVC


def case(d, n):

    print 'd = ', d
    print 'noise = ', n

    train, m = moons(1000, d, 10, 6, n)
    scatter([p[0] for p in train], [p[1] for p in train], c = m)
    show()

    test, _ = moons(3000, d, 10, 6, n)

    clf = SVC()
    clf.fit(train, m)
    m = clf.predict(test)

    scatter([p[0] for p in test], [p[1] for p in test], c = m)
    show()
```

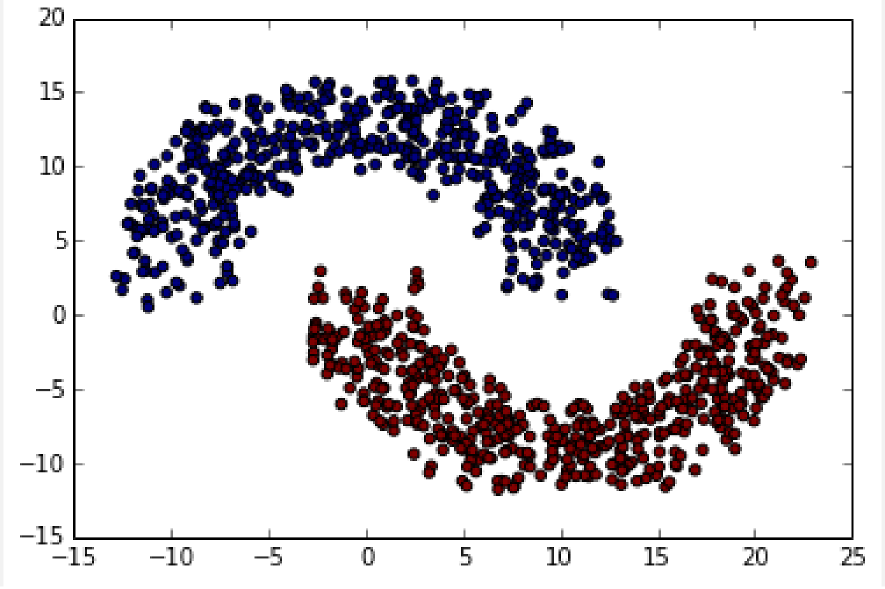
```
case(0, None)
print ''
case(-5, None)
print ''
case(-10, None)
print ''
case(-15, None)
print ''
case(-20, None)
print ''
case(-25, None)
print ''
```

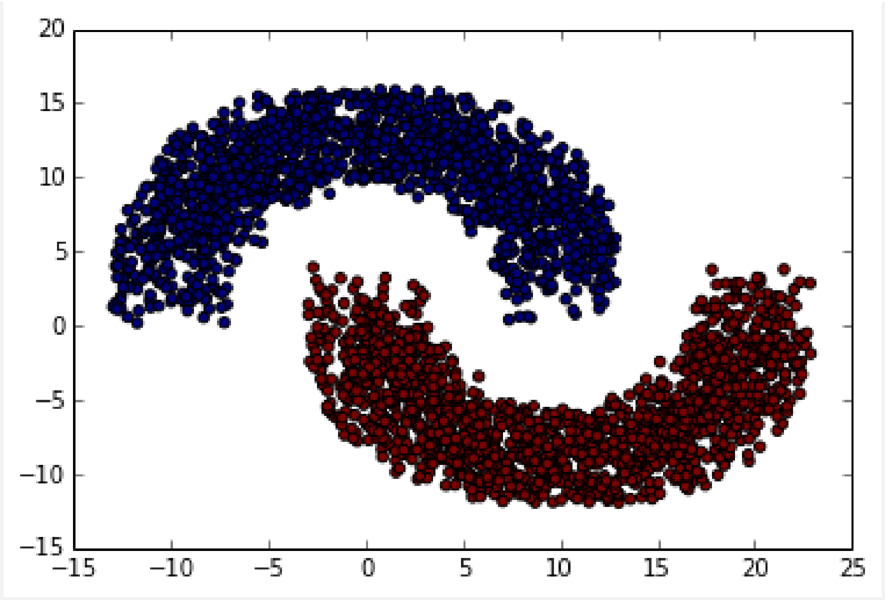
```
d =  0
noise =  None
```



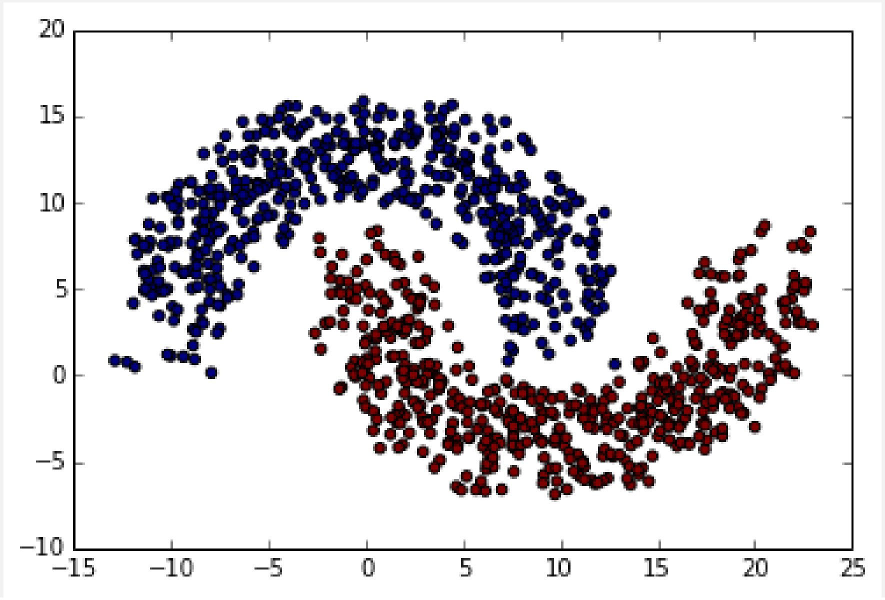


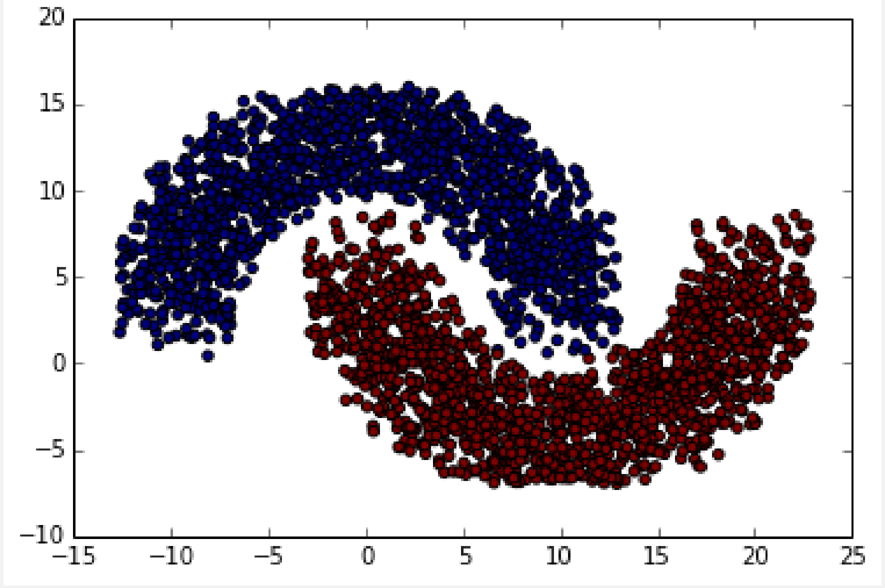
```
d =  -5
noise =  None
```



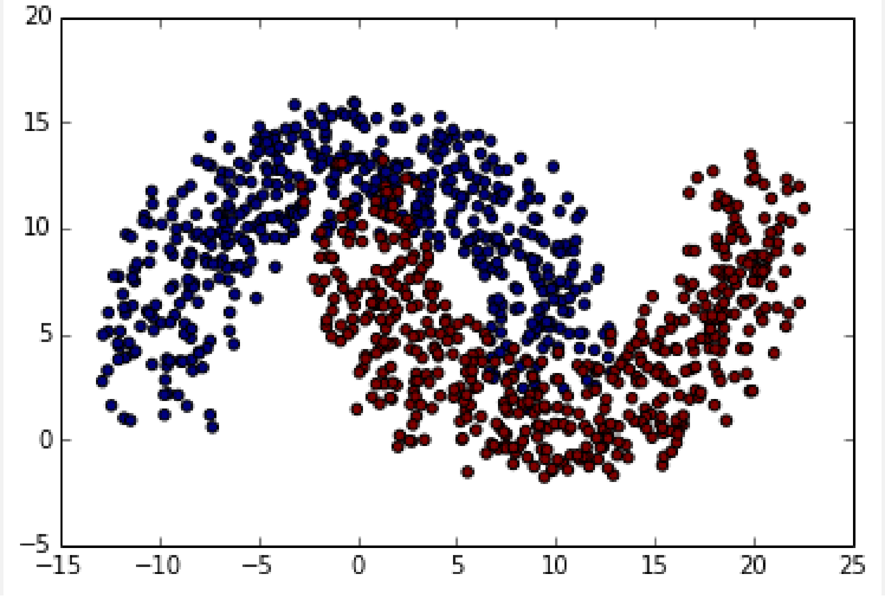


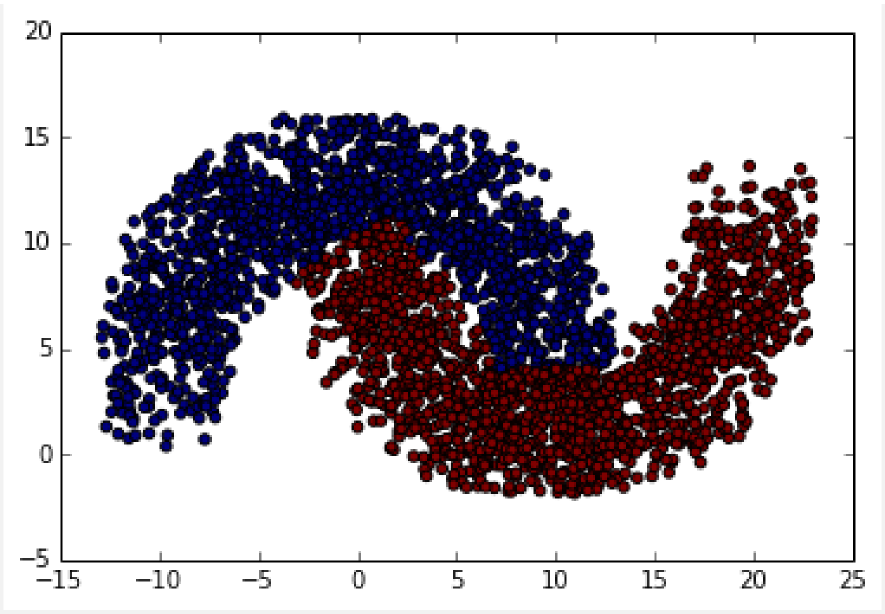
```
d =  -10
noise =  None
```



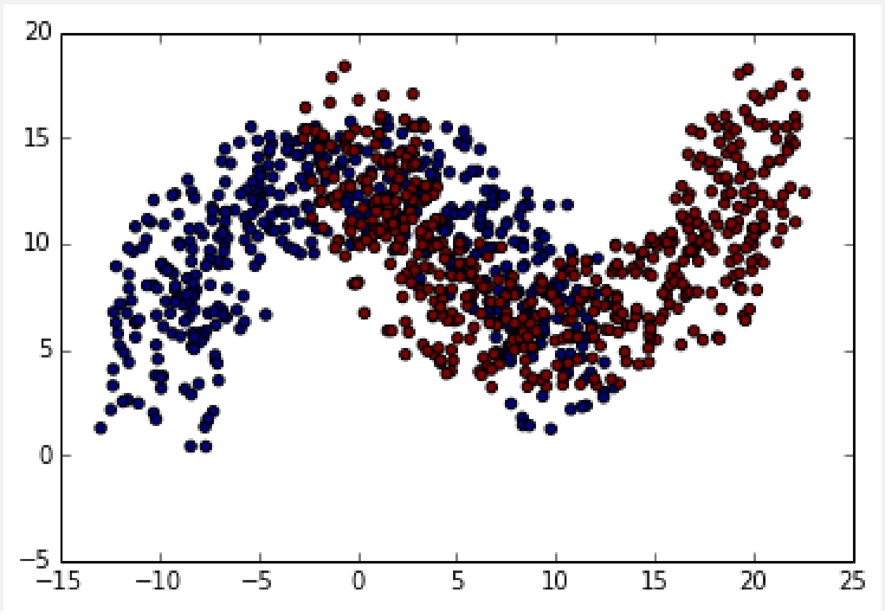


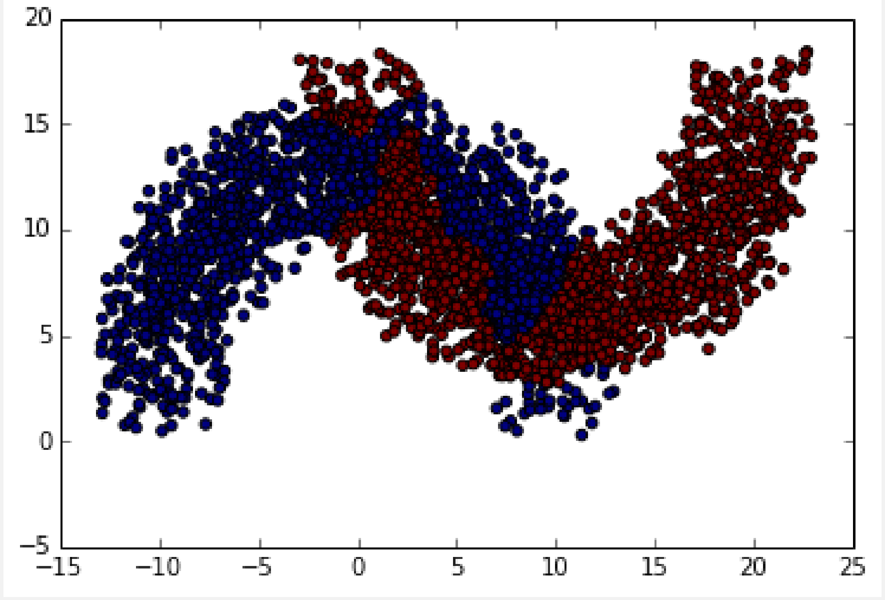
```
d =  -15
noise =  None
```



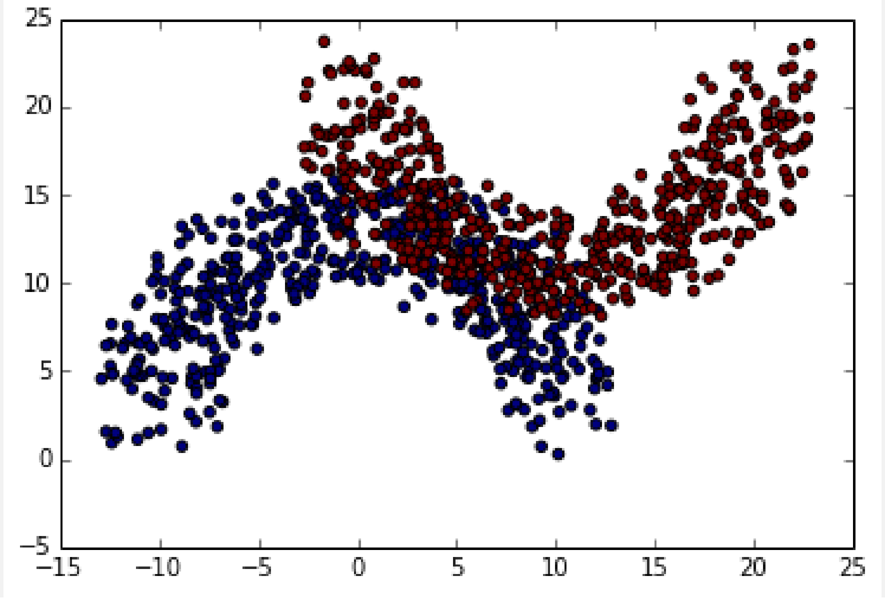


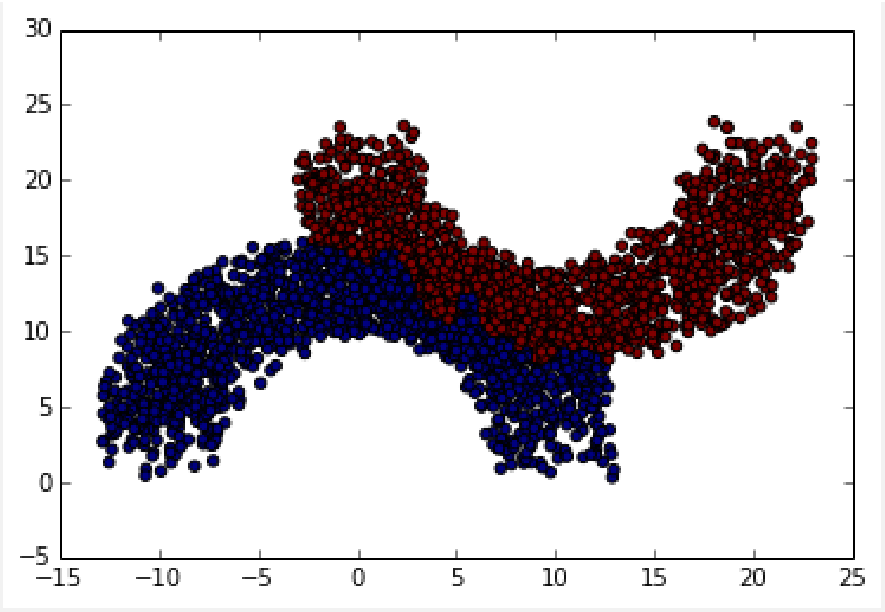
```
d =  -20
noise =  None
```





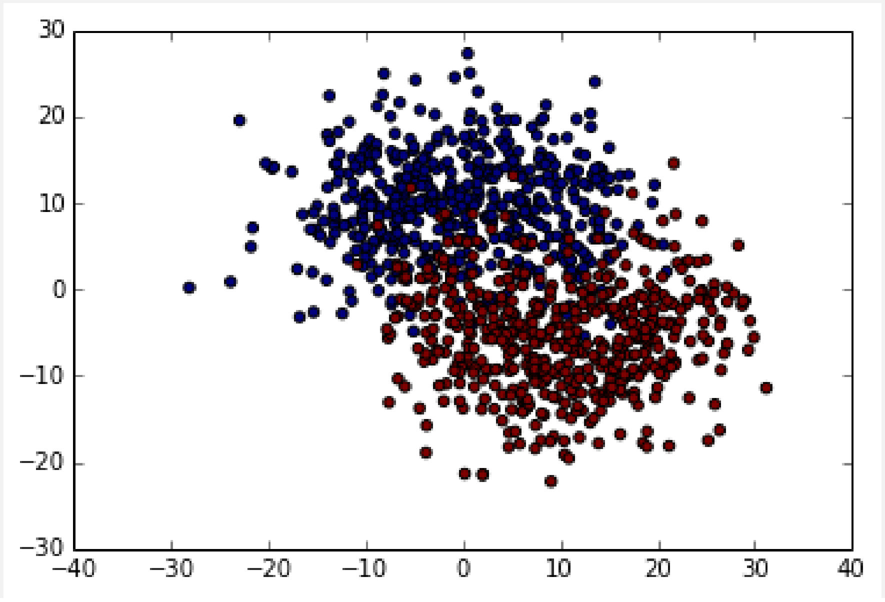
```
d =  -25
noise =  None
```

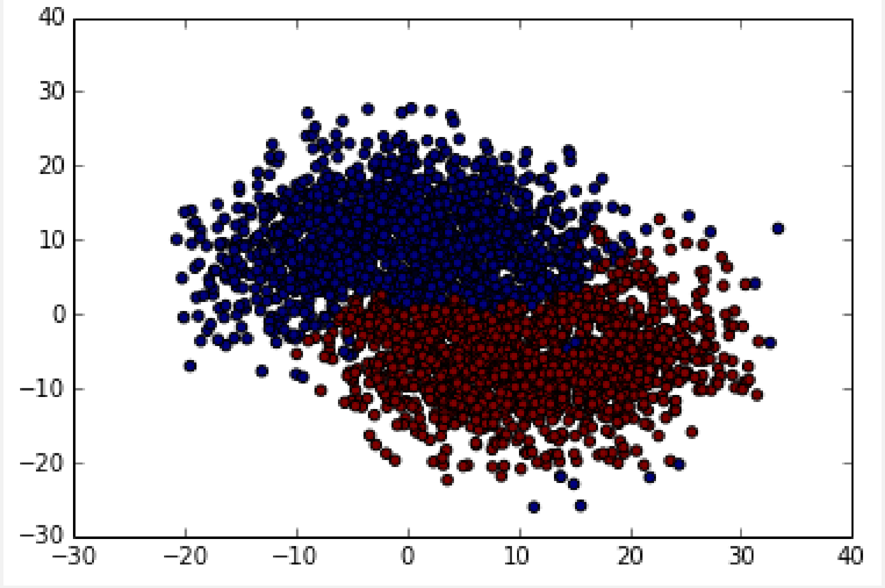




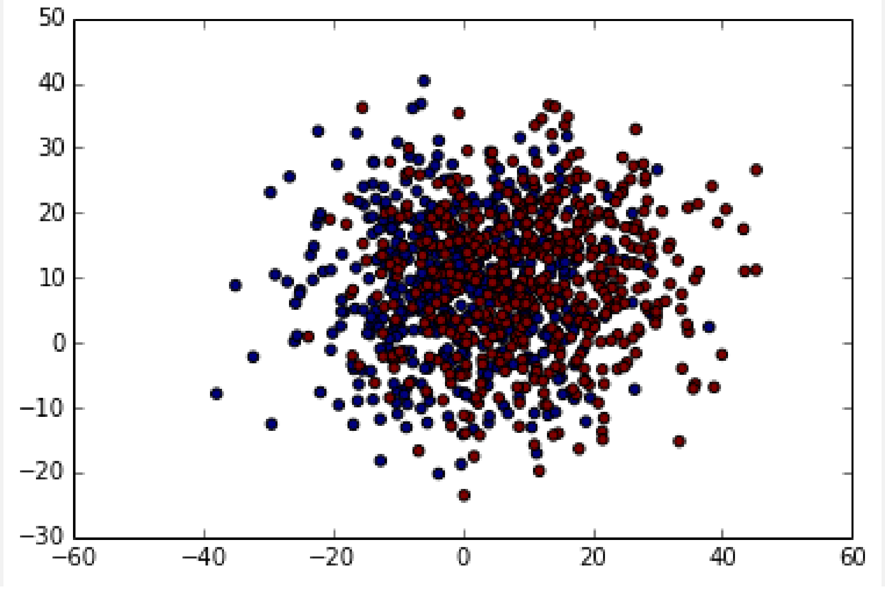
```
case(-5, 0.5)
print ''

case(-20, 1)
print''
```

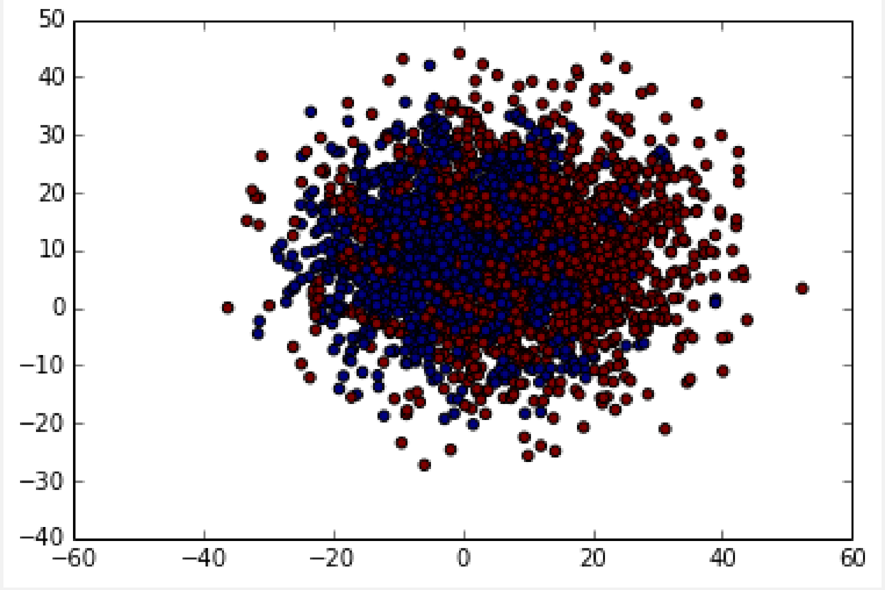
```
d =  -5
noise =  0.5
```





```
d =  -20
noise =  1
```




50
40
30
20
10
0
−10
−20
−30
−40
−60
−40
−20
0
20
40
60