Generate Dataset

In [ ]:

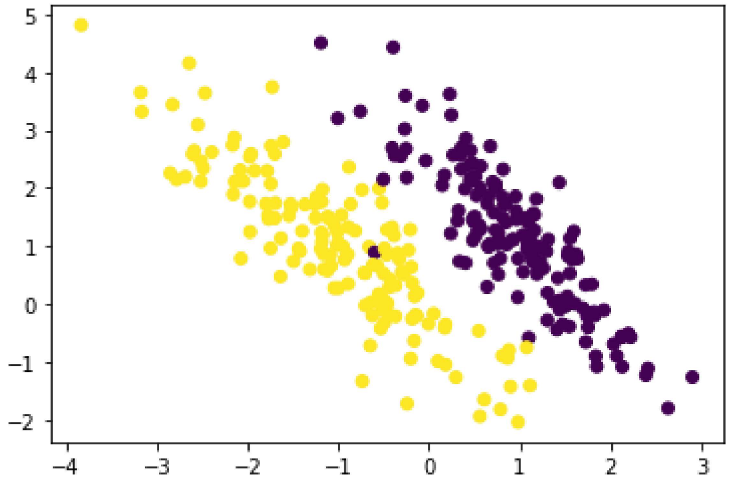
```
from sklearn.datasets import  make_classification
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import numpy as np
```

In [ ]:

```
X,Y = make_classification(n_classes=2,n_samples=300,n_clusters_per_class=1,random_
```

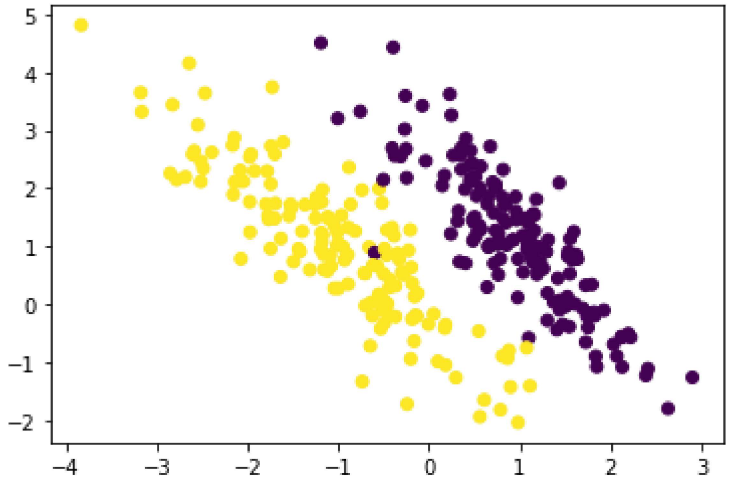
In [ ]:

```
plt.scatter(X[:,0],X[:,1],c=Y)
plt.show()
```



In [ ]:

```
plt.scatter(X[:,0],X[:,1],c=Y)
plt.show()
```



In [ ]:

```
from sklearn import svm
```

In [ ]:

```
svc = svm.SVC(kernel='linear')
svc.fit(X,Y)
print(svc.score(X,Y))
```

```
0.9966666666666667
```

In [ ]:

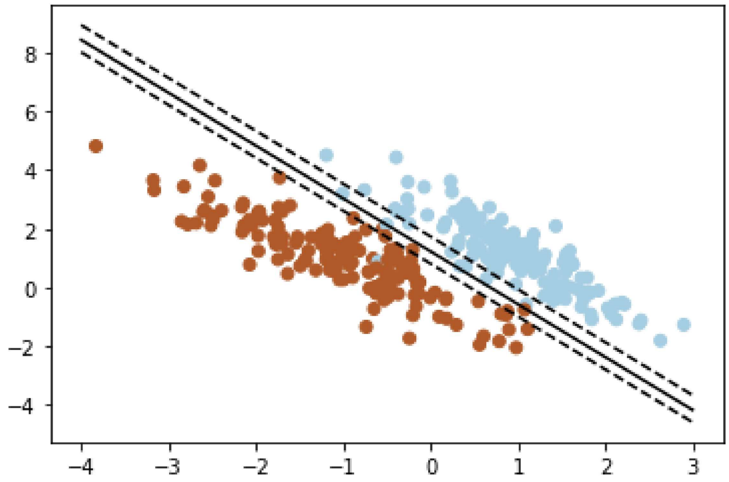
```
w = svc.coef_[0]
a = -w[0] / w[1]
xx = np.linspace(-4, 3)
yy = a * xx - (svc.intercept_[0]) / w[1]
```

```
# plot the parallels to the separating hyperplane that pass through the
# support vectors
b = svc.support_vectors_[0]
yy_down = a * xx + (b[1] - a * b[0])
b = svc.support_vectors_[-1]
yy_up = a * xx + (b[1] - a * b[0])

# plot the line, the points, and the nearest vectors to the plane
plt.plot(xx, yy, 'k-')
plt.plot(xx, yy_down, 'k--')
plt.plot(xx, yy_up, 'k--')

plt.scatter(svc.support_vectors_[:, 0], svc.support_vectors_[:, 1],
            s=80, facecolors='none')
plt.scatter(X[:, 0], X[:, 1], c=Y, cmap=plt.cm.Paired)

plt.axis('tight')
plt.show()
```



In [ ]: