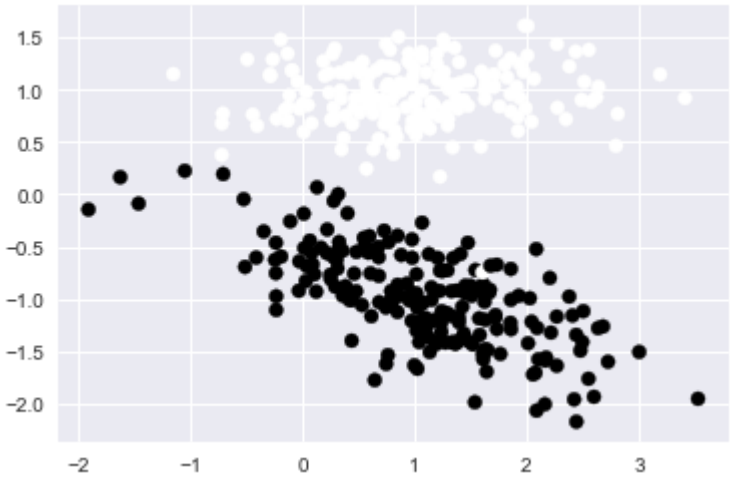
Support Vector Machine

In [1]:
```
from sklearn.datasets import load_digits
from sklearn import svm
from sklearn.model_selection import cross_val_score
from sklearn.datasets import make_classification
import matplotlib.pyplot as plt
import numpy as np
```

In [2]:
```
X,Y = make_classification(n_samples=400,n_classes=2,n_features=2, n_informative=2,
Y[Y==0] = -1
```

In [3]:
```
plt.style.use('seaborn')
plt.scatter(X[:,0],X[:,1],c=Y)
plt.show()
```



In [4]:
```
svc = svm.LinearSVC()
svc.fit(X,Y)
acc = cross_val_score(svc,X,Y,scoring = "accuracy",cv=5).mean()
print(acc)
```

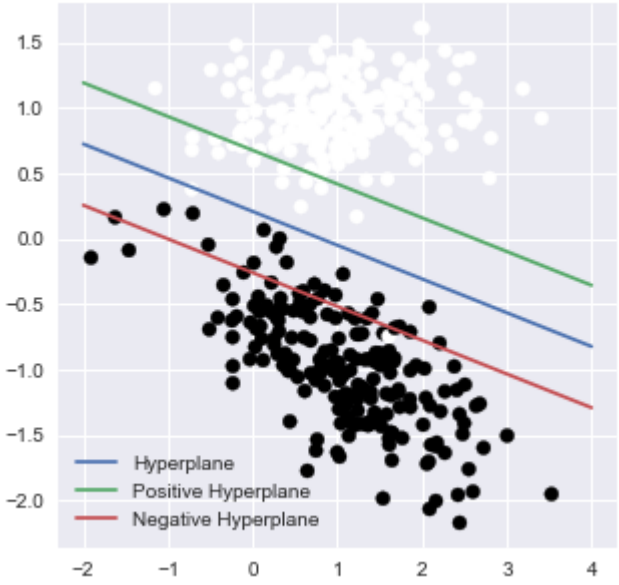
```
0.9949999999999999
```

In [5]:
```
W,B = svc.coef_,svc.intercept_[0]
print(W,B)
```

```
[[-0.55123051 -2.13722989]] 0.4420670537791713
```

In [6]:
```
def plotHyperplane(w1,w2,b):
    x_1 = np.linspace(-2,4,10)
    x_2 = -(w1*x_1 + b)/w2
    x_p = -(w1*x_1 + b+1)/w2
    x_n = -(w1*x_1 + b-1)/w2
    plt.figure(figsize=((5,5)))
    plt.plot(x_1,x_2, label = "Hyperplane")
    plt.plot(x_1,x_p, label = "Positive Hyperplane")
    plt.plot(x_1,x_n, label = "Negative Hyperplane")
```

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



In [ ]: