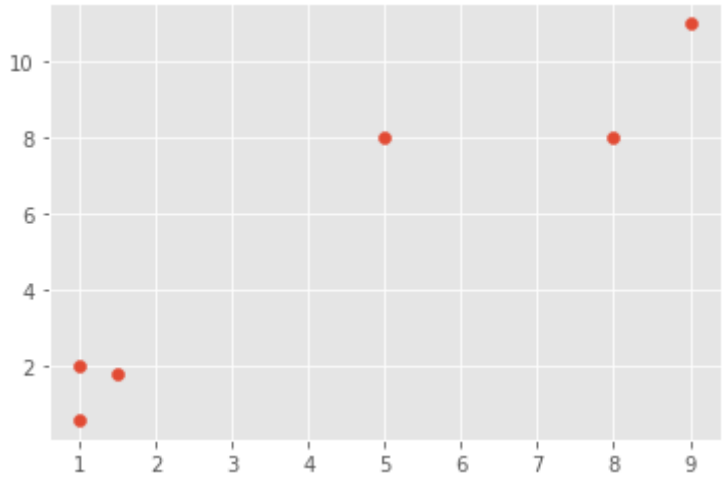
Linear Support Vector Classifier

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import style
style.use("ggplot")
from sklearn import svm
```

In [4]:
```
X = [1,5,1.5,8,1,9]
Y= [2,8,1.8,8,0.6,11]
```

In [5]:
```
plt.scatter(X,Y)
plt.show()
```



In [6]:
```
data = np.array(list(zip(X,Y)))
```

In [7]:
```
data
```

Out[7]:
```
array([[ 1. ,  2. ],
       [ 5. ,  8. ],
       [ 1.5,  1.8],
       [ 8. ,  8. ],
       [ 1. ,  0.6],
       [ 9. , 11. ]])
```

In [44]:
```
target = [0,1,0,1,0,1]
```

In [45]:
```
classifier = svm.SVC(kernel="linear", C=1.0)
classifier.fit(data, target)
```

Out[45]:
```
SVC(kernel='linear')
```

In [47]:
```
p = np.array([0,1]).reshape(1,2)# Tenemos que pasarlo a formato una fila y dos columnas
print(p)
classifier.predict(p)
```

Out[47]:
```
[[0 1]]
array([0])
```

- Modelo orignal: w0 . X + w1 . Y + e = 0
- El hiperplano obtenido en 2D es =
  - Y = a.x + b

In [48]:
```
w = classifier.coef_[0]
w
```

Out[48]:
```
array([0.1380943 , 0.24462418])
```

In [49]:
```
a = -w[0]/w[1]
a # Pendiente de la recta
```

Out[49]:
```
-0.564516129032258
```

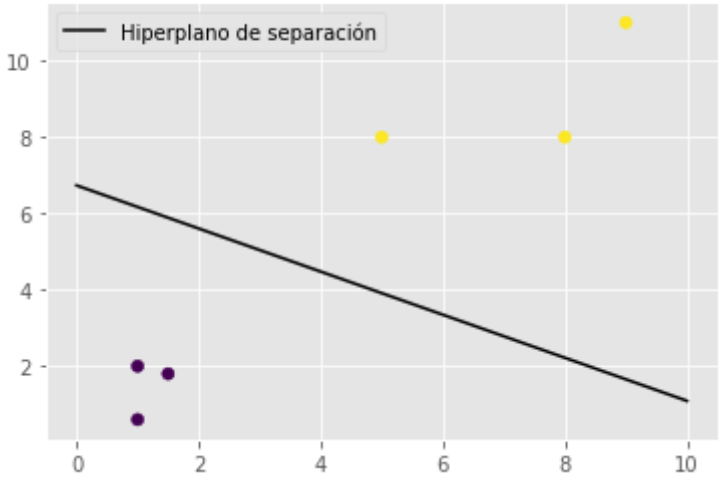
In [50]:
```
b = -classifier.intercept_[0]/w[1]
b #ordenada en el origen
```

Out[50]:
```
6.734677437813051
```

In [51]:
```
xx = np.linspace(0,10)
yy = a * xx + b
```

In [52]:
```
plt.plot(xx,yy, 'k-', label="Hiperplano de separación")
plt.scatter(X,Y, c = target)
plt.legend()
plt.plot()
```

Out[52]:
```
[]
```



In [ ]:
```
```