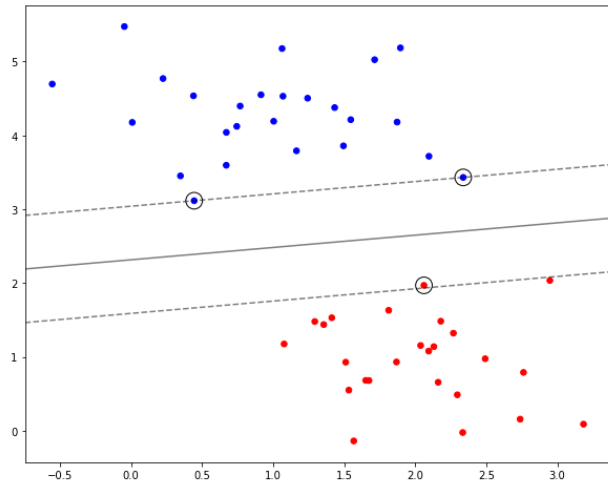
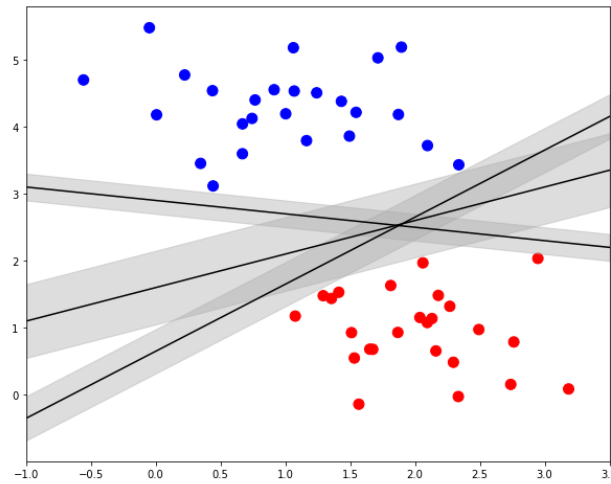
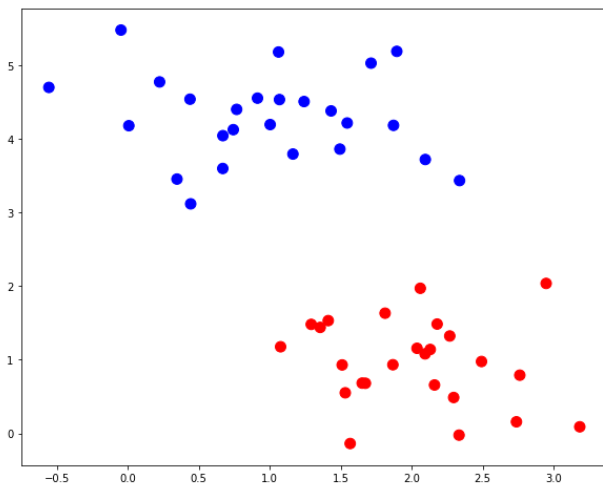


**Nombre:** Vergara Kenia.






**C. I. N°:** V- 16.443.850.

## Máquinas de Vectores de Soporte (SVM)

Es un algoritmo de aprendizaje supervisado que se puede usar tanto para problemas de clasificación como para problemas de regresión. SVM tiene como objetivo encontrar el hiperplano óptimo (maximizar la distancia entre los puntos más cercanos de cada clase) que no sólo clasifique el conjunto de datos existente, sino que también ayude a predecir la clase de los datos no vistos. El hiperplano óptimo es el que tiene el mayor margen.



## Código

```
Abrir  svm1.py ~/Escritorio Guardar      
import numpy as np                #Importando biblioteca para funciones matematicas  
import matplotlib.pyplot as plt    #Importando biblioteca para generacion de graficas  
from matplotlib import style  
style.use("ggplot")  
from sklearn.svm import SVC  
  
x = [2, 6, 1.5, 9, 3, 13]         #Consideracion de dos caracteristicas x,y  
y = [2, 9, 1.8, 8, 0.6, 11]  
  
plt.scatter(x,y)                  #Graficando los datos  
plt.show()  
  
X = np.array([[2,2],[6,9],[1.5,1.8],[9,8],[3,0.6],[13,11]])    #Convirtiendo las caracteristicas x,y en una matriz  
  
y = [0,1,0,1,0,1]                #Etiquetas 0 y 1, donde: "0" Pares de coordenadas que son numeros bajos  
                                # "1" Pares de coordenadas que son numeros altos  
  
clf = SVC(kernel='linear', probability=True, tol=1e-3)          #Definiendo el clasificador  
  
clf.fit(X,y)                    #Metodo de ajuste, aprende de los datos  
  
j=clf.predict(np.matrix([4.58,7.76]))    #Predice "1" por ser un par de coordenadas de numeros altos  
print(j)  
  
j1=clf.predict(np.matrix([0.58,0.7]))    #Predice "0" por ser un par de coordenadas de numeros bajos  
print(j1)  
  
w = clf.coef_[0]                #Calculando para graficar el hiperplano  
a = -w[0] / w[1]  
xx = np.linspace(0,12)  
yy = a * xx - clf.intercept_[0] / w[1]  
h0 = plt.plot(xx, yy, 'k-', label="Hiperplano")  
plt.scatter(X[:, 0], X[:, 1], c = y)  
plt.legend()  
plt.show()
```

## Resultados por el Terminal

Figure 1

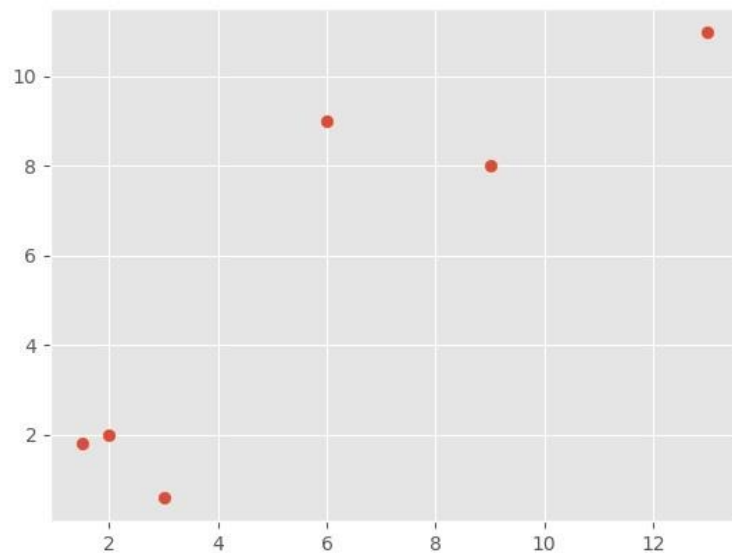
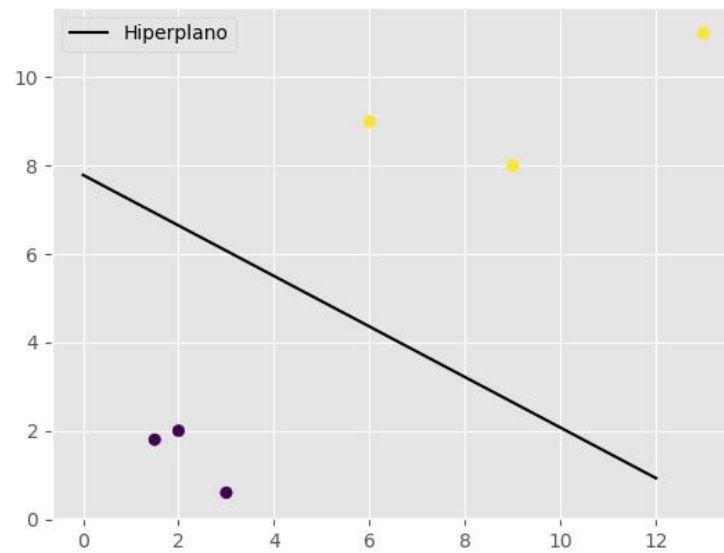


Figure 1



## Resultados por el Terminal

```
kenia@kenia-Dell-System-Inspiron-N4110: ~/Escritorio
Archivo Editar Ver Buscar Terminal Ayuda
kenia@kenia-Dell-System-Inspiron-N4110:~$ cd Escritorio/
kenia@kenia-Dell-System-Inspiron-N4110:~/Escritorio$ ls
codigo ProyectoGradoLatex svm1.py SVM2 ULA
kenia@kenia-Dell-System-Inspiron-N4110:~/Escritorio$ python svm1.py
[1]
[0]
kenia@kenia-Dell-System-Inspiron-N4110:~/Escritorio$
```