X1 X2 Y

2.7810836 2.550537003 0

1.465489372 2.362125076 0

3.396561688 4.400293529 0

1.38807019 1.850220317 0

3.06407232 3.005305973 0

7.627531214 2.759262235 1

5.332441248 2.088626775 1

6.922596716 1.77106367 1

8.675418651 -0.2420686549 1

7.673756466 3.508563011 1

import numpy as np

import matplotlib.pyplot as plt

X = np.array([

[2.7810836,2.550537003],

[1.465489372,2.362125076],

[3.396561688,4.400293529],

[1.38807019,1.850220317],

[3.06407232,3.005305973],

[7.627531214,2.759262235],

[5.332441248,2.088626775],

[6.922596716,1.77106367],

[8.675418651,-0.2420686549],

[7.673756466,3.508563011]

])

# 2 GRUPOS BINARIO

y = np.array([0, 0, 0, 0, 0, 1, 1, 1, 1, 1])

import pandas as pd

x1 = pd.DataFrame(X)

y1 = pd.DataFrame(y)

x1['target'] = y1

x1 = x1.set\_axis(['x', 'y', 'Target'], axis=1, inplace=False)

df1 = x1[x1['Target'] == 0]

df2 = x1[x1['Target'] == 1]

del df1['Target']

del df2['Target']

import matplotlib.pyplot as plt

fig = plt.figure()

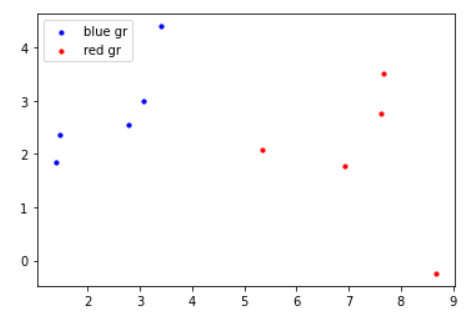
ax1 = fig.add\_subplot(111)

ax1.scatter(df1['x'], df1['y'], s = 10, c = 'b', label = 'blue gr')

ax1.scatter(df2['x'], df2['y'], s = 10, c = 'r', label = 'red gr')

plt.legend(loc='upper left');

plt.show()



Lo que nos muestra acá es que los rojos están de un lado y los azules de otro

Si se busca predecir un nuevo punto, y esta en medio, se utiliza KNN o SVM

from sklearn import linear\_model

model1 = linear\_model.LogisticRegression(solver = 'liblinear', C = 75)

model1.fit(X, y)

test = pd.DataFrame([[3, 2]])

result = model1.predict(test)

result[0]

valor 3,2 nos da 0

valor 6,3 nos da 1

x\_min, x\_max = X[:, 0].min() - 1, X[:, 0].max() + 1

y\_min, y\_max = X[:, 1].min() - 1, X[:, 1].max() + 1

h = (x\_max / x\_min)/100

xx, yy = np.meshgrid(np.arange(x\_min, x\_max, h),

np.arange(y\_min, y\_max, h))

X\_plot = np.c\_[xx.ravel(), yy.ravel()]

C1 = 1.0

from sklearn import svm

svc\_classifier = svm.SVC(kernel='linear', C=C1, decision\_function\_shape = 'ovr').fit(X, y)

Z = svc\_classifier.predict(X\_plot)

Z = Z.reshape(xx.shape)

plt.figure(figsize = (15, 5))

plt.subplot(121)

plt.contourf(xx, yy, Z, cmap = plt.cm.tab10, alpha = 0.3)

plt.scatter(X[:, 0], X[:, 1], c = y, cmap = plt.cm.Set1)

plt.xlabel('Sepal length')

plt.ylabel('Sepal width')

plt.xlim(xx.min(), xx.max())

plt.title('SVC with linear kernel')

