https://www.tutorialspoint.com/artificial\_intelligence\_with\_python/index.htm

Support Vector Machines (SVM)

* Data de iris de flor
* Tomara 2 features
* Presentara un diagrama donde clasificara los 2 features
* No se necesita target, solo se desea conocer cuantos grupos para 2 features

SVM is a supervised machine learning algorithm that can be used for both regression and classification. The main concept of SVM is to plot each data item as a point in n-dimensional space with the value of each feature being the value of a particular coordinate. Here n would be the features we would have. Following is a simple graphical representation to understand the concept of SVM –

|  |  |
| --- | --- |
| We have two features. The 2 variables are plot in two dimensional space where each point has two co-ordinates, called support vectors.  The line splits the data into two different classified groups. This line would be the classifier. | Support Vector Machines 2 |

Here, we are going to build an SVM classifier by using scikit-learn and iris dataset. Scikitlearn library has the **sklearn.svm** module and provides sklearn.svm.svc for classification. The SVM classifier to predict the class of the iris plant based on 4 features are shown below.

Dataset

We will use the iris dataset which contains 3 classes of 50 instances each, where each class refers to a type of iris plant. Each instance has the four features namely sepal length, sepal width, petal length and petal width. The SVM classifier to predict the class of the iris plant based on 4 features is shown below.

Kernel

It is a technique used by SVM. Basically these are the functions which take low-dimensional input space and transform it to a higher dimensional space.

The kernel function can be any one among linear, polynomial, rbf and sigmoid. In this example, we will use the linear kernel.

import pandas as pd

import numpy as np

from sklearn import svm, datasets

import matplotlib.pyplot as plt

iris = datasets.load\_iris()

We are taking first two features −

X = iris.data[:, :2]

#y = iris.target # por ahora no usamos target

We will plot the support vector machine boundaries with original data.

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()]

We need to give the value of regularization parameter, and create a SVM classifier object

C1 = 1.0

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)

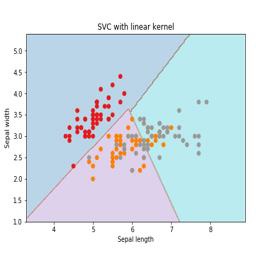
plt.scatter(X[:, 0], X[:, 1], c = y, cmap = plt.cm.Set1)

plt.xlabel('Sepal length')

plt.ylabel('Sepal width')

plt.xlim(xx.min(), xx.max())

plt.title('SVC with linear kernel')



import sklearn

from sklearn import svm, datasets

iris = datasets.load\_iris()

#from 4 features take las 2

features = iris.data[:, :2]

target\_data = iris.target  # por ahora no usamos target

from sklearn.model\_selection import train\_test\_split

train, test, train\_labels, test\_labels = train\_test\_split(features,target\_data,test\_size = 0.40, random\_state = 42)

from sklearn.naive\_bayes import GaussianNB

#initialize model

Model\_gnb = GaussianNB()

#train the model by fitting it to the data with Model\_gnb.fit()

model = Model\_gnb.fit(train, train\_labels)

NewData = Model\_gnb.predict(test)

from sklearn.metrics import accuracy\_score

print(accuracy\_score(test\_labels, NewData))

#PREDECIR

#PREDECIR para 1 fila

import numpy as np

valores = test[2]

# o dando los valoers

valores = np.array([7.7 , 2.6])

valores = valores.reshape(1, -1)

ArrPreddiccion = Model\_gnb.predict(valores)

print(ArrPreddiccion[0])

#PREDECIR PARA UNA SET

ArrPreddiccion = Model\_gnb.predict(test)

print(ArrPreddiccion)