**TECNOLÓGICO​ ​NACIONAL​ ​DE​ ​MÉXICO**

**INSTITUTO TECNOLÓGICO DE TIJUANA**

**SUBDIRECCIÓN ACADÉMICA**

**DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN**

**SEMESTRE:**Enero - Junio 2020

**CARRERA:**Ing. Tecnologías de la Información y Comunicaciones

**MATERIA:**Datos Masivos

**UNIDAD​ ​A​ ​EVALUAR:​**​Unidad​ ​2

**NOMBRE Y NÚMERO DE CONTROL DEL ALUMNO:**Diaz Martinez Ruben Emilio # 15210791

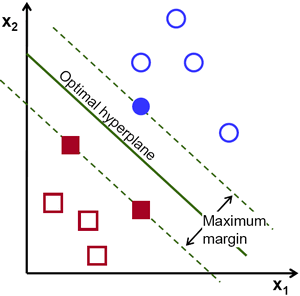
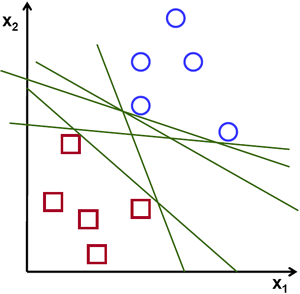
**NOMBRE DEL MAESTRO (A):**Dr. Christian Romero Hernandez

Introducción

Supongo que ya te habrías acostumbrado a los algoritmos de regresión lineal y regresión logística. Si no, le sugiero que los vea antes de pasar a admitir la máquina de vectores. La máquina de vectores de soporte es otro algoritmo simple que todo experto en aprendizaje automático debería tener en su arsenal. La máquina de vectores de soporte es muy preferida por muchos, ya que produce una precisión significativa con menos potencia de cálculo. Support Vector Machine, abreviado como SVM, se puede usar tanto para tareas de regresión como de clasificación. Pero, es ampliamente utilizado en los objetivos de clasificación.

¿Qué es la máquina de vectores de soporte?

El objetivo del algoritmo de máquina de vectores de soporte es encontrar un hiperplano en un espacio N-dimensional (N - el número de características) que clasifica claramente los puntos de datos.

1. Import libraries and package

package org.apache.spark.examples.ml

import org.apache.spark.ml.classification.LinearSVC

2. Import a Spark Session.

import org.apache.spark.sql.SparkSession

3.Load the data from the file and add it to a variable to train it.

val spark = SparkSession.builder.appName("LinearSVCExample").getOrCreate()

4. Load the data stored in LIBSVM format as a DataFrame.

val training = spark.read.format("libsvm").load("/usr/local/spark-2.3.4-bin-hadoop2.6/data/mllib/sample\_libsvm\_data.txt")

5. Create an object of type LinearSVC.

Set the number of iterations to 10 with the setMaxIter method and Set the regularization parameter

val lsvc = new LinearSVC().setMaxIter(10).setRegParam(0.1)

6. Fit the model

val lsvcModel = lsvc.fit(training)

7. Print result

println(s"Coefficients: ${lsvcModel.coefficients} Intercept: ${lsvcModel.intercept}")