**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

Regresión Lineal Práctica # 1

**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

import org.apache.spark.ml.regression.LinearRegression

import org.apache.log4j.\_

Logger.getLogger("org").setLevel(Level.ERROR)

import org.apache.spark.sql.SparkSession

val spark = SparkSession.builder().getOrCreate()

val data = spark.read.option("header","true").option("inferSchema", "true").format("csv").load("Clean-Ecommerce.csv")

data.printSchema

data.head(1)

val colnames = data.columns

val firstrow = data.head(1)(0)

println("\n")

println("Example data row")

for(ind <- Range(0, colnames.length)){

println(colnames(ind))

println(firstrow(ind))

println("\n")

}

import org.apache.spark.ml.feature.VectorAssembler

import org.apache.spark.ml.linalg.Vectors

val df = data.select(data("Yearly Amount Spent").as("label"),$"Avg Session Length", $"Time on App", $"Time on Website", $"Length of Membership", $"Yearly Amount Spent")

val new\_assembler = new VectorAssembler().setInputCols(Array("Avg Session Length","Time on App","Time on Website","Length of Membership","Yearly Amount Spent")).setOutputCol("features")

val output = new\_assembler.transform(df).select($"label",$"features")

val lr = new LinearRegression()

val lrModel = lr.fit(output)

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

val trainingSummary = lrModel.summary

trainingSummary.residuals.show()

val RMSE = trainingSummary.rootMeanSquaredError

val MSE = scala.math.pow(RMSE, 2.0)

val R2 = trainingSummary.r2

|  |
| --- |
|  |

|  |
| --- |
|  |