INSTITUTO TECNOLÓGICO CAMPUS TIJUANA

ING. EN SISTEMAS COMPUTACIONALES

Subdirección Académica

Departamento de Sistemas y Computación

BDD-1704 TI9A - 6:00pm-7:00pm

ASIGNATURA:

Datos Masivos

SEMESTRE:

Septiembre- Enero 2020

Examen:

Práctica Evaluatoria

MAESTRO:

JOSE CHRISTIAN ROMERO HERNANDEZ

Equipo:

Marco Antonio Rodriguez Medrano

Aide Ceballos Bobadilla

Introducción

En el siguiente trabajo se muestra el desarrollo de la práctica evaluatoria de la unidad 3, la cual consiste en agrupar los clientes de regiones específicas de un distribuidor al mayoreo. Esto en base a las ventas de algunas categorías de productos.

Desarrollo

//1.-Importamos la sesion Spark

import org.apache.spark.sql.SparkSession

//4.-Importar la librería de Kmeans para el algoritmo de agrupamiento.

import org.apache.spark.ml.clustering.KMeans

//7. Importar Vector Assembler y Vector

import org.apache.spark.ml.feature.{VectorAssembler,StringIndexer,VectorIndexer,OneHotEncoder}

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

import org.apache.log4j.\_

//2.-Minimizar Errores

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

//3.-Crear una instancia de la sesion Spark

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

//5.-Carga el dataset de Wholesale Customers Data

val dataset = spark.read.option("header","true").option("inferSchema","true").csv("C:/Users/aide0/OneDrive/Escritorio/Practica-Evaluatoria-Unidad3/Wholesale customers data.csv")

//6.-Seleccione las siguientes columnas: Fresh, Milk, Grocery, Frozen, Detergents\_Paper,Delicassen

//y llamar a este conjunto feature\_data

val feature\_data = dataset.select($"Fresh", $"Milk", $"Grocery", $"Frozen", $"Detergents\_Paper", $"Delicassen")

/\*8. Crea un nuevo objeto Vector Assembler para las columnas de caracteristicas

como un conjunto de entrada, recordando que no hay etiquetas\*/

val assembler = new VectorAssembler().setInputCols(Array("Fresh", "Milk", "Grocery", "Frozen", "Detergents\_Paper", "Delicassen")).setOutputCol("feature")

//9. Utilice el objeto assembler para transformar feature\_data

val traning = assembler.transform(feature\_data)

//10.Crear un modelo Kmeans con K=3

val kmeans = new KMeans().setK(3).setSeed(1L)

val model = kmeans.fit(traning)

//11.Evalúe los grupos utilizando Within Set Sum of Squared Errors WSSSE e imprima los centroides.

val WSSSE = model.computeCost(traning)



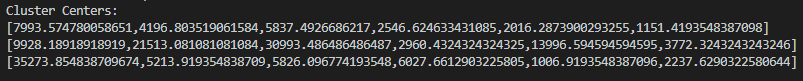
println(s"Within Set Sum of Squared Errors = $WSSSE")



//ressultado

println("Cluster Centers: ")

model.clusterCenters.foreach(println)



Conclusión

-Ceballos Bobadilla Aide 15211282

Como conclusión, de mi parte considero que al igual que las herramientas anteriormente utilizadas nos permite representar los datos y darles forma, específicamente con estas herramientas nos es posible agruparlas según sus características.

-Rodriguez Medrano Marco Antonio 17210635

Como conclusión, de mi parte, esta librería llamada Kmeans nos ayuda al momento de querer clasificar una cantidad considerable de datos de una forma eficiente y rápida algo que todo analista de datos desea, y con este programa emulamos ese principio.