**TECNOLÓGICO NACIONAL DE MÉXICO**

**INSTITUTO TECNOLÓGICO DE TIJUANA**

**DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN**

**ENERO - JUNIO 2020**

**INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES**

**DATOS MASIVOS**

**BDD-1704IF9A**

**“Examen Unidad 2”**

**INTEGRANTES:**  
Enciso Maldonado Aileen Yurely No. Control: 15210329  
García Bautista Ana Laura No. Control: 15210793

**PROFESOR:**

Romero Hernández Jose Christian

**FECHA DE ENTREGA**

**10/Junio/2020**



**TECNOLÓGICO NACIONAL DE MÉXICO**

***INSTITUTO TECNOLÓGICO DE TIJUANA***

**SUBDIRECCIÓN ACADÉMICA**

Departamento de Sistemas y Computación

**EXAMEN**

Carrera: Ingeniería En Sistemas Computacionales/ Tecnologías de la información/ Informatíca Período: **Enero-Junio 2020**

Materia: Datos Masivos Grupo: BDD-1704/IF9A Salón:

Unidad (es) a evaluar: Unidad 3 Tipo de examen:Practico Fecha:

Catedrático: Jose Christian Romero Hernandez Firma del maestro: Calificación:

Alumno:

No. Control:

Instrucciones

Desarrolle las siguientes instrucciones en Spark con el lenguaje de programación Scala.

Objetivo:

El objetivo de este examen practico es tratar de agrupar los clientes de regiones específicas de un distribuidor al mayoreo. Esto en base a las ventas de algunas categorías de productos.

Las fuente de datos se encuentra en el repositorio: <https://github.com/jcromerohdz/BigData/blob/master/Spark_clustering/Whole>[sale](https://github.com/jcromerohdz/BigData/blob/master/Spark_clustering/Wholesalecustomers) [customers](https://github.com/jcromerohdz/BigData/blob/master/Spark_clustering/Wholesalecustomers)data.csv

1. Importar una simple sesión Spark.
2. Utilice las lineas de código para minimizar errores
3. Cree una instancia de la sesión Spark
4. Importar la librería de Kmeans para el algoritmo de agrupamiento.
5. Carga el dataset de Wholesale Customers Data
6. Seleccione las siguientes columnas: Fres, Milk, Grocery, Frozen, Detergents\_Paper, Delicassen y llamar a este conjunto feature\_data
7. Importar Vector Assembler y Vector
8. Crea un nuevo objeto Vector Assembler para las columnas de caracteristicas como un conjunto de entrada, recordando que no hay etiquetas
9. Utilice el objeto assembler para transformar feature\_data
10. Crear un modelo Kmeans con K=3
11. Evalúe los grupos utilizando Within Set Sum of Squared Errors WSSSE e imprima los centroides.

**Instrucciones de evaluación**

* + Tiempo de entrega 4 dias
  + Al terminar poner el codigo y la explicación en la rama (branch) correspondiete de su github asi mismo realizar su explicación de la solución en su google drive.
  + Finalmente defender su desarrollo en un video de 8-10 min el cual servira para dar su calificación, este video debe subirse a youtube para ser compartido por un link.

//Libreria spark

import org.apache.spark.sql.SparkSession

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

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

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

//para que no marque errores

import org.apache.log4j.\_

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

// cargamos el csv

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

val dataset = spark.read.option("header","true").option("inferSchema","true").csv("Wholesale customers data.csv")

//selecionamos las columnas que vamos a entrenar

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

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

//entrenamos la data que colocamos en el vectorassembler

val traning = assembler.transform(feature\_data)

//la data entrenada la pasamos al modelo

val kmeans = new KMeans().setK(2).setSeed(1L)

val model = kmeans.fit(traning)

// Evaluate clustering by calculate Within Set Sum of Squared Errors.

val WSSSE = model.computeCost(traning)

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

//resultado

println("Cluster Centers: ")

model.clusterCenters.foreach(println)