EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

NATIONAL TECHNOLOGICAL INSTITUTE OF MEXICO
TECHNOLGICAL INSTITUTE OF TIJUANA

ACADEMIC SUBDIRECTION
Systems and Computing Department

SEMESTER
August - December 2021

ACADEMIC CAREER
Eng. Information and Communication Technologies

SUBJECT AND KEY:
Big Data BDD-1704TI9A

STUDENT’S NAME AND REGISTRATION:
Velázquez Farrera César Alejandro 17212937

NAME OF THE WORK:
Evaluative Practice - Unit 3

UNIT:
Unit III

NAME OF THE TEACHER:
M.C. José Christian Romero Hernández

Tijuana, Baja California **Due Date:** Diciembre 7, 2021

Introduction

Instructions: Develop the following instructions in Spark with the Scala programming language.

Objective: In this evaluation practice, grouping the customers of specific regions of a wholesale distributor. This is based on the sales of some product categories.

Data source:
The data source is in the Github repository belonging to jcromerohdz with the name of "Wholesale customers data.csv" in the following link: https://github.com/jcromerohdz/BigData/blob/master/Spark_clustering/Wholesale%20customers%20data.csv

1. Import a simple session into Spark.

```
//Import a simple Spark session
import org.apache.spark.sql.SparkSession
```

2. Use lines of code to minimize errors
3. Create an instance of the Spark session

```
//Create a simple Spark Session
val session = SparkSession.builder().getOrCreate()
```

4. Import the Kmeans library for the clustering algorithm.

```
//Import the Kmeans library for the clustering algorithm
import org.apache.spark.ml.clustering.KMeans
```

5. Loads the Wholesale Customers Data dataset.

```
//Load the Wholesale Customers Data dataset
val data = session.read.option("header", "true").option("inferSchema",
"true").format("csv").load("CSV/Wholesale customers data.csv")
```

6. Select the following columns: Fresh, Milk, Grocery, Frozen, Detergents_Paper, Delicassen and call this set "feature_data."

```
// Select the following columns: Fresh, Milk, Grocery, Frozen,
Detergents_Paper, Delicassen and call this set feature_data
```

```
var feature_data = data.select($"Fresh", $"Milk", $"Grocery",
$"Frozen", $"Detergents_Paper", $"Delicassen")
```

7. Import the VectorAssembler and Vector libraries.

```
//import VectorAssembler and Vector
import org.apache.spark.ml.feature.VectorAssembler
import org.apache.spark.ml.linalg.Vectors
```

8. Create a Vector Assembler object for the feature columns as an input set, remembering that there are no labels.

```
//Create a new Vector Assembler object for the feature columns as an
input set, remembering that there are no labels
val vector_assembler = (new
VectorAssembler().setInputCols(Array("Fresh", "Milk", "Grocery",
"Frozen", "Detergents_Paper", "Delicassen")).setOutputCol("features"))
```

9. Use the assembler object to transform feature_data.

```
//Use the assembler object to transform feature_data
val data_kmeans = vector_assembler.transform(feature_data)
```

10. Create a Kmeans model with K = 3

```
// Create a Kmeans model with K = 3
val model = new KMeans().setK(3).setSeed(1L)
```

11. Evaluate the clusters using Within Set Sum of Squared Errors WSSS and print the generated centroids.

```
//Evaluate the groups using Within Set Sum of Squared Errors WSSSE and
print the  centroids.
val result = model.fit(data_kmeans)
import org.apache.spark.ml.evaluation.ClusteringEvaluator
val WSSSE = result.computeCost(data_kmeans)
println(s"Within set sum of squared errors = $WSSSE")
```

```
//Print the centroids
println("Cluster Centers:")
result.clusterCenters.foreach(println)
```