TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE TIJUANA
SUBDIRECCIÓN ACADÉMICA
DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN

SEMESTRE:

Enero - Junio 2022

CARRERA:

Ing. en Sistemas Computacionales

MATERIA:

Datos Masivos

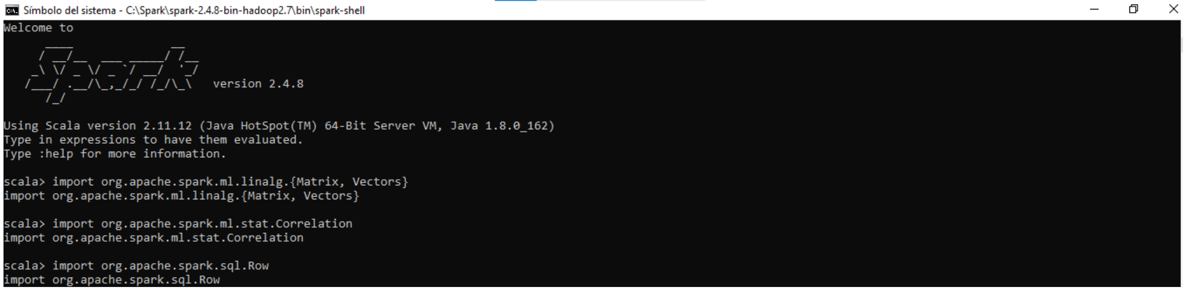
TÍTULO ACTIVIDAD:

Practica 1

NOMBRE Y NÚMERO DE CONTROL DEL ALUMNO:

Hernandez Pablo Anahi Del Carmen - 18210486

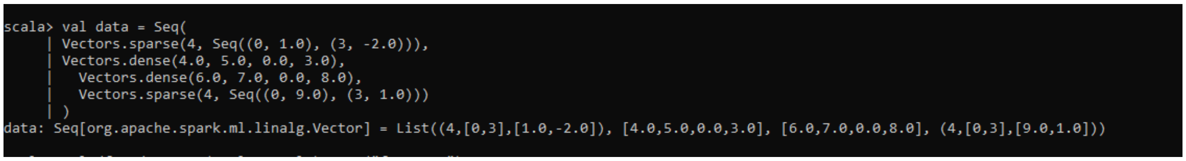
Here we start Spark and add the libraries that we were going to use



Símbolo del sistema - C:\Spark\spark-2.4.8-bin-hadoop2.7\bin\spark-shell

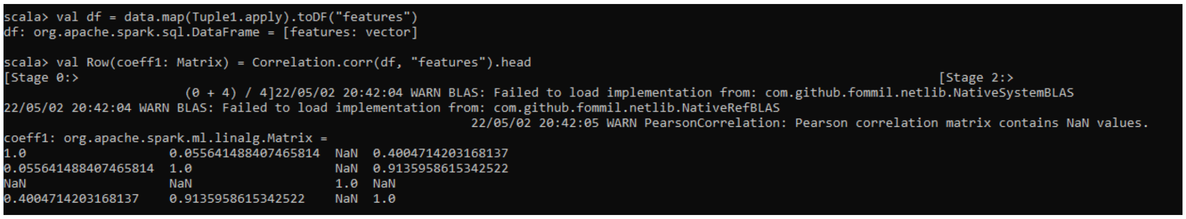
```
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /___/ .__/\_,_/_/ /_/\_\   version 2.4.8
      /_/

Using Scala version 2.11.12 (Java HotSpot(TM) 64-Bit Server VM, Java 1.8.0_162)
Type in expressions to have them evaluated.
Type :help for more information.

scala> import org.apache.spark.ml.linalg.{Matrix, Vectors}
import org.apache.spark.ml.linalg.{Matrix, Vectors}

scala> import org.apache.spark.ml.stat.Correlation
import org.apache.spark.ml.stat.Correlation

scala> import org.apache.spark.sql.Row
import org.apache.spark.sql.Row
```

In this part we create vectors



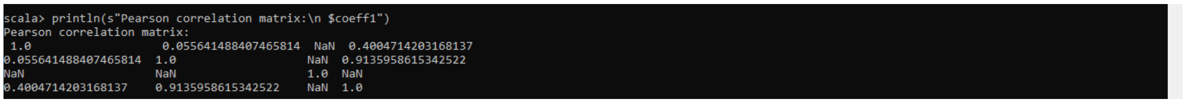
```
scala> val data = Seq(
     | Vectors.sparse(4, Seq((0, 1.0), (3, -2.0))),
     | Vectors.dense(4.0, 5.0, 0.0, 3.0),
     |   Vectors.dense(6.0, 7.0, 0.0, 8.0),
     |   Vectors.sparse(4, Seq((0, 9.0), (3, 1.0)))
     | )
data: Seq[org.apache.spark.ml.linalg.Vector] = List((4,[0,3],[1.0,-2.0]), [4.0,5.0,0.0,3.0], [6.0,7.0,0.0,8.0], (4,[0,3],[9.0,1.0]))
```

Here we print the vectors, and the results are shown in the screenshot.



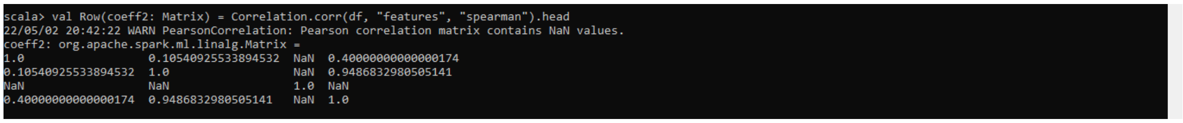
```
scala> val df = data.map(Tuple1.apply).toDF("features")
df: org.apache.spark.sql.DataFrame = [features: vector]

scala> val Row(coeff1: Matrix) = Correlation.corr(df, "features").head
[Stage 0:>                                                                                              [Stage 2:>
                        (0 + 4) / 4]22/05/02 20:42:04 WARN BLAS: Failed to load implementation from: com.github.fommil.netlib.NativeSystemBLAS
22/05/02 20:42:04 WARN BLAS: Failed to load implementation from: com.github.fommil.netlib.NativeRefBLAS
                                                               22/05/02 20:42:05 WARN PearsonCorrelation: Pearson correlation matrix contains NaN values.
coeff1: org.apache.spark.ml.linalg.Matrix =
1.0                   0.055641488407465814  NaN  0.4004714203168137
0.055641488407465814  1.0                   NaN  0.9135958615342522
NaN                   NaN                   1.0  NaN
0.4004714203168137    0.9135958615342522    NaN  1.0
```

Here we print the vectors and the results are shown in the screenshot.



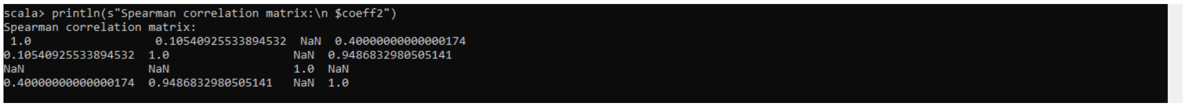
```
scala> println(s"Pearson correlation matrix:\n $coeff1")
Pearson correlation matrix:
 1.0                   0.055641488407465814  NaN  0.4004714203168137
0.055641488407465814  1.0                   NaN  0.9135958615342522
NaN                   NaN                   1.0  NaN
0.4004714203168137    0.9135958615342522    NaN  1.0
```

Here we print the vectors and the results are shown in the screenshot.



```
scala> val Row(coeff2: Matrix) = Correlation.corr(df, "features", "spearman").head
22/05/02 20:42:22 WARN PearsonCorrelation: Pearson correlation matrix contains NaN values.
coeff2: org.apache.spark.ml.linalg.Matrix =
1.0                   0.10540925533894532  NaN  0.40000000000000174
0.10540925533894532   1.0                  NaN  0.9486832980505141
NaN                   NaN                  1.0  NaN
0.40000000000000174   0.9486832980505141   NaN  1.0
```

Here we print the vectors with the correlation in the screenshot.



```
scala> println(s"Spearman correlation matrix:\n $coeff2")
Spearman correlation matrix:
 1.0                   0.10540925533894532  NaN  0.40000000000000174
0.10540925533894532   1.0                  NaN  0.9486832980505141
NaN                   NaN                  1.0  NaN
0.40000000000000174   0.9486832980505141   NaN  1.0
```