

Tarea M52 – Bryan Alberto Coronado García

- Configuración de sesión de Spark en terminal

Configuración de plataforma Spark

```
1 # Importación de librerías para la práctica de Spark
2 import sys
3 from operator import add
4 from pyspark.sql import SparkSession
5 import warnings
6 warnings.filterwarnings("ignore")
7
8 # Configuración de Spark
9 spark = SparkSession.builder.appName("MS2").getOrCreate()
1]
Python
WARNING: Using incubator modules: jdk.incubator.vector
Using Spark's default log4j profile: org/apache/spark/log4j2-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
25/07/31 17:00:07 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
```

- Importación de datos Housing en Spark y verificación del tipo de dataframe

Importación de datos de Housing a una estructura Spark

- Modificación del tipo de datos

```
1 # Transformación de los tipos de datos para asegurar que son correctos
2
3 from pyspark.sql.types import IntegerType, FloatType
4
5 df = df.withColumn("price", df.price.cast(IntegerType()))
6 df = df.withColumn("area", df.area.cast(IntegerType()))
7 df = df.withColumn("bedrooms", df.bedrooms.cast(IntegerType()))
8 df = df.withColumn("bathrooms", df.bathrooms.cast(IntegerType()))
9 df = df.withColumn("stories", df.stories.cast(IntegerType()))
10 df = df.withColumn("parking", df.parking.cast(IntegerType()))
11
12 df.printSchema()

root
|-- price: integer (nullable = true)
|-- area: integer (nullable = true)
|-- bedrooms: integer (nullable = true)
|-- bathrooms: integer (nullable = true)
|-- stories: integer (nullable = true)
|-- mainroad: string (nullable = true)
|-- guestroom: string (nullable = true)
|-- basement: string (nullable = true)
|-- hotwaterheating: string (nullable = true)
|-- airconditioning: string (nullable = true)
|-- parking: integer (nullable = true)
|-- prefarea: string (nullable = true)
|-- furnishingstatus: string (nullable = true)
```

- Listado completo ordenado por ZipCode (Area)

1) Listado completo de columnas ordenado por zipcode

- Listado con mayor número de casas, precio promedio, y ordenado por ZipCode (Area)

```

1 df.groupBy("area").agg( \
2   F1.count("*").alias("num_casas"),
3   F1.avg("price").alias("precio_promedio")) \
4   .orderBy(F1.desc("num_casas")).show()
5
6 # NOTA: No es posible realizar una columna del tamaño de la casa, ya que no existe la data correspondiente.
7 # El punto actualizado es "Listado de zipcodes ordenados por el número de casas, y el promedio de precio"

+-----+-----+
|area|num_casas|  precio_promedio|
+-----+-----+
|6000|    24| 7051479.166666667|
|3000|    14| 3309000.0|
|4500|    13| 4031192.3076923075|
|4000|    11| 4040272.727272727|
|5500|     9| 5762555.555555556|
|6600|     9| 6443111.111111111|
|3600|     8| 3360437.5|
|3180|     7| 3530000.0|
|4040|     7| 4139000.0|
|6360|     7| 5604000.0|
|3640|     7| 3542000.0|
|3630|     7| 3515000.0|
|3500|     6| 4275833.333333333|
|5400|     6| 4631666.666666667|
|2145|     6| 3606166.666666665|
|3450|     5| 3680600.0|
|3850|     5| 3136000.0|
|3480|     5| 3227000.0|
|4800|     5| 5742800.0|
|3520|     5| 4107600.0|
+-----+
only showing top 20 rows

```

- Listado con mayor número de habitaciones, de baños, precio promedio, y ordenado por ZipCode (Area)

```

1 df.groupBy("area").agg( \
2   F1.count("bedrooms").alias("num_habitaciones"),
3   F1.count("bathrooms").alias("num_baños"),
4   F1.avg("price").alias("precio_promedio")) \
5   .orderBy(F1.desc("num_habitaciones"), F1.desc("num_baños")).show(20)

+-----+-----+-----+
|area|num_habitaciones|num_baños|  precio_promedio|
+-----+-----+-----+
|6000|        24|      24| 7051479.166666667|
|3000|        14|      14| 3309000.0|
|4500|        13|      13| 4031192.3076923075|
|4000|        11|      11| 4040272.727272727|
|5500|         9|       9| 5762555.555555556|
|6600|         9|       9| 6443111.111111111|
|3600|         8|       8| 3360437.5|
|3180|         7|       7| 3530000.0|
|4040|         7|       7| 4139000.0|
|6360|         7|       7| 5604000.0|
|3640|         7|       7| 3542000.0|
|3630|         7|       7| 3515000.0|
|3500|         6|       6| 4275833.333333333|
|5400|         6|       6| 4631666.666666667|
|2145|         6|       6| 3606166.666666665|
|3450|         5|       5| 3680600.0|
|3850|         5|       5| 3136000.0|
|3480|         5|       5| 3227000.0|
|4800|         5|       5| 5742800.0|
|3520|         5|       5| 4107600.0|
+-----+
only showing top 20 rows

```