analysis on crime data

December 2, 2020

1 Will analyse the data of Northern Ireland

1.1 Description

The received data will be present in a "data" folder, which will contain only the necessary ranges that should be analysed The collected data is from https://data.police.uk/data/

2 Import

[10]:
```
import org.apache.spark.sql.functions.input_file_name
import java.io.File
import org.apache.spark.sql.DataFrame
import org.apache.spark.sql.functions.{collect_list, map, udf}
```

[10]:
```
import org.apache.spark.sql.functions.input_file_name
import java.io.File
import org.apache.spark.sql.DataFrame
import org.apache.spark.sql.functions.{collect_list, map, udf}
```

3 Ingest

[11]:
```
var outcomesDf = spark
                    .read
                    .option("header", "true")
                    .csv("./data/*/*-outcomes.csv")
                    .withColumn("districtName", input_file_name())
var streetDf = spark
                    .read
                    .option("header", "true")
                    .csv("./data/*/*-street.csv")
                    .withColumn("districtName", input_file_name())
```

[11]:
```
outcomesDf: org.apache.spark.sql.DataFrame = [Crime ID: string, Month: string
… 9 more fields]
streetDf: org.apache.spark.sql.DataFrame = [Crime ID: string, Month: string …
```

```
 11 more fields]
```

4 Analyse data

[12]:
```
streetDf.select($"month").distinct.orderBy($"month").show()

println("**** The street dataframe:")
streetDf.schema.names.foreach(println)

println("\n**** The outcomes dataframe:")
outcomesDf.schema.names.foreach(println)
```

```
+-------+
|  month|
+-------+
|2018-01|
|2018-02|
|2018-03|
|2018-04|
|2018-05|
|2018-06|
|2018-07|
|2018-08|
|2018-09|
|2018-10|
|2018-11|
|2018-12|
|2019-01|
|2019-02|
|2019-03|
+-------+

**** The street dataframe:
Crime ID
Month
Reported by
Falls within
Longitude
Latitude
Location
LSOA code
LSOA name
Crime type
Last outcome category
Context
```

```
districtName

**** The outcomes dataframe:
Crime ID
Month
Reported by
Falls within
Longitude
Latitude
Location
LSOA code
LSOA name
Outcome type
districtName
```

[13]:
```
streetDf.show(3, false)
```

```
+------------------------------------------------------------------+-------+------
--------------------+-------------------------+---------+---------+----------
---------------+---------+------------------+---------------------------+----
-----------------------------------------+-------+-----------------------------
--------------------------------+
|Crime ID                                                          |Month
|Reported by                |Falls within               |Longitude|Latitude
|Location                   |LSOA code|LSOA name          |Crime type
|Last outcome category                          |Context|districtName
|
+------------------------------------------------------------------+-------+------
--------------------+-------------------------+---------+---------+----------
---------------+---------+------------------+---------------------------+----
-----------------------------------------+-------+-----------------------------
--------------------------------+
|e9fe81ec7a6f5d2a80445f04be3d7e92831dbf3090744ebf94c46f359ca94854|2018-07|Metrop
olitan Police Service|Metropolitan Police Service|0.774271 |51.148147|On or near
Bethersden Road|E01024031|Ashford 012B       |Other theft
|Status update unavailable                     |null
|file:/home/jovyan/data/2018-07/2018-07-metropolitan-street.csv|
|076b796ba1e1ba3f69c9144e2aa7a7bc85b61d51bf7a5966fa1a45fecb1c6aca|2018-07|Metrop
olitan Police Service|Metropolitan Police Service|-1.007293|51.893136|On or near
Prison         |E01017674|Aylesbury Vale 010D|Other crime                  |Court
result unavailable                       |null
|file:/home/jovyan/data/2018-07/2018-07-metropolitan-street.csv|
|163e996d58995cf87d14f15711fbd87052681919f02029af4739c2eb88be7f5e|2018-07|Metrop
olitan Police Service|Metropolitan Police Service|0.744706 |52.038219|On or near
Hillside Road  |E01029918|Babergh 007F       |Violence and sexual
offences|Investigation complete; no suspect identified|null
|file:/home/jovyan/data/2018-07/2018-07-metropolitan-street.csv|
+------------------------------------------------------------------+-------+------
```

```
--------------------+-------------------------+--------+---------+----------
---------------+---------+------------------+--------------------------+----
---------------------------------------+-------+-----------------------------
--------------------------------+
only showing top 3 rows
```

[14]:
```
println("total number of records: " + streetDf.count)
println("total number of distinct crime IDs: " + streetDf.select($"Crime ID").
 ↪distinct.count)
println("total number of outcomes: " + outcomesDf.count)
println("total number of outcomes by distinct crime Ids: " + outcomesDf.
 ↪select($"Crime ID").distinct.count)
```

```
total number of records: 8261884
total number of distinct crime IDs: 6273525
total number of outcomes: 5708337
total number of outcomes by distinct crime Ids: 5063409
```

4.0.1 Verify the unicity of the Crime ID

From this analysis we can see that there are multiple crimes with the same ID

[15]:
```
// STREET
println(streetDf.groupBy($"Crime ID")
        .agg(count($"Crime ID").as("count"))
        .where($"count" > 1)
        .count
)

// Trying to identify the posibility of a multi-value primary key
println(streetDf.groupBy($"Crime ID", $"Month")
        .agg(count($"Crime ID").as("count"))
        .where($"count" > 1)
        .count
)
println(streetDf.groupBy($"Crime ID", $"Longitude")
        .agg(count($"Crime ID").as("count"))
        .where($"count" > 1)
        .count
)
println(streetDf.groupBy($"Crime ID", $"LSOA code")
        .agg(count($"Crime ID").as("count"))
        .where($"count" > 1)
        .count
)
println(streetDf.groupBy($"Crime ID",
                         $"Month",
```

```
                            $"Longitude",
                            $"Latitude",
                            $"Reported by",
                            $"LSOA code",
                            $"Falls within")
        .agg(count($"Crime ID").as("count"))
        .where($"count" > 1)
        .count
)
```

```
60736
51716
46830
56233
46526
```

From the analysis we gather that there is a high chance that these values are just ERRORS, thus a filter should be but in place to rule them out

[16]:
```
var streetMultiIdDf = streetDf.groupBy($"Crime ID")
                              .agg(count($"Crime ID").as("count"))
                              .where($"count" > 1)
                              .select($"Crime ID")
                              .withColumnRenamed("Crime ID", "crimeIdMulti")

streetDf.join(streetMultiIdDf, $"Crime ID" === $"crimeIdMulti", "inner")
        .groupBy($"Crime ID")
        .agg(
             collect_list($"Longitude").as("longitude_list"),
             collect_set($"Longitude").as("longitude_set"),
             countDistinct($"Longitude").as("count_dictinct")
            )
        .where($"count_dictinct" > 1)
        .show()
```

```
+--------------------+--------------------+--------------------+--------------+
|            Crime ID|      longitude_list|       longitude_set|count_dictinct|
+--------------------+--------------------+--------------------+--------------+
|01c4971330ba99d0d…|[-5.761179, -5.84…|[-6.317724, -6.30…|            14|
|05835ad9b99479fc3…|[-5.642241, -5.71…|[-6.665021, -5.94…|            14|
|06ac8d52630a7d582…|[-2.557898, -2.55…|[-2.557898, -2.55…|             2|
|0f16eef9874953881…|[-5.905221, -5.92…|[-5.959611, -7.65…|             9|
|14abaa4c688812a6b…|[-2.350023, -2.35…|[-2.351300, -2.35…|             2|
|180ec98648a24f6b8…|[-5.700116, -5.85…|[-6.688120, -7.62…|            13|
|19b0da5dc82df7534…|[-5.700995, -5.82…|[-5.860830, -7.30…|            15|
|1adffe4a3d31f5a5a…|[-5.535522, -5.90…|[-6.303628, -5.93…|            14|
|2081173e60040dcd2…|[-5.712780, -5.85…|[-5.931754, -6.11…|            15|
|247ff19de8d7a9544…|[-1.949217, -1.95…|[-1.949217, -1.95…|             2|
```

```
|26c4115526615e931…|[-2.382062, -2.40…|[-2.408787, -2.38…|              2|
|28ea5d5462b690099…|[-5.806113, -5.91…|[-7.592636, -6.77…|             14|
|2921f8806488c0601…|[-0.080322, -0.08…|[-0.080322, -0.08…|              2|
|33010a0bf9727ffad…|[-5.666286, -5.69…|[-5.931754, -6.61…|             15|
|387f4c28bae6d30be…|[-0.947233, -0.95…|[-0.953699, -0.94…|              2|
|39954ba5b64a4e599…|[-5.670828, -5.70…|[-6.168122, -6.67…|             12|
|40fff217660e49ce2…|[-5.702351, -5.86…|[-6.975427, -6.94…|             15|
|4280fbb7be6b5faf3…|[-1.120947, -1.36…|[-1.120947, -1.36…|              2|
|4dddced897e539971…|[-5.839357, -5.84…|[-5.936493, -6.01…|             15|
|4f67297d191c1d8d1…|[-5.676660, -5.70…|[-7.309140, -6.05…|             15|
+------------------+------------------+------------------+---------------+
only showing top 20 rows
```

[16]:
```
streetMultiIdDf: org.apache.spark.sql.DataFrame = [crimeIdMulti: string]
```

[17]:
```
// OUTCOMES
outcomesDf.groupBy($"Crime ID")
        .agg(count($"Crime ID").as("count"))
        .where($"count" > 1)
        .count
```

[17]:
```
res8: Long = 492850
```

[18]:
```
var streetMultiIdDf = streetDf.groupBy($"Crime ID")
                              .agg(count($"Crime ID").as("count"))
                              .where($"count" > 1)
                              .select($"Crime ID")
                              .withColumnRenamed("Crime ID", "crimeIdMulti")

streetDf.join(streetMultiIdDf, $"Crime ID" === $"crimeIdMulti", "inner")
        .groupBy($"Crime ID")
        .agg(
             collect_list($"Longitude").as("longitude_list"),
             collect_set($"Longitude").as("longitude_set"),
             countDistinct($"Longitude").as("count_dictinct")
            )
        .where($"count_dictinct" > 1)
        .show()
```

```
+------------------+------------------+------------------+---------------+
|          Crime ID|     longitude_list|      longitude_set|count_dictinct|
+------------------+------------------+------------------+---------------+
|01c4971330ba99d0d…|[-5.761179, -5.84…|[-6.317724, -6.30…|             14|
|05835ad9b99479fc3…|[-5.642241, -5.71…|[-6.665021, -5.94…|             14|
|06ac8d52630a7d582…|[-2.557898, -2.55…|[-2.557898, -2.55…|              2|
```

```
|0f16eef9874953881…|[-5.905221, -5.92…|[-5.959611, -7.65…|             9|
|14abaa4c688812a6b…|[-2.350023, -2.35…|[-2.351300, -2.35…|             2|
|180ec98648a24f6b8…|[-5.700116, -5.85…|[-6.688120, -7.62…|            13|
|19b0da5dc82df7534…|[-5.700995, -5.82…|[-5.860830, -7.30…|            15|
|1adffe4a3d31f5a5a…|[-5.535522, -5.90…|[-6.303628, -5.93…|            14|
|2081173e60040dcd2…|[-5.712780, -5.85…|[-5.931754, -6.11…|            15|
|247ff19de8d7a9544…|[-1.949217, -1.95…|[-1.949217, -1.95…|             2|
|26c4115526615e931…|[-2.382062, -2.40…|[-2.408787, -2.38…|             2|
|28ea5d5462b690099…|[-5.806113, -5.91…|[-7.592636, -6.77…|            14|
|2921f8806488c0601…|[-0.080322, -0.08…|[-0.080322, -0.08…|             2|
|33010a0bf9727ffad…|[-5.666286, -5.69…|[-5.931754, -6.61…|            15|
|387f4c28bae6d30be…|[-0.947233, -0.95…|[-0.953699, -0.94…|             2|
|39954ba5b64a4e599…|[-5.670828, -5.70…|[-6.168122, -6.67…|            12|
|40fff217660e49ce2…|[-5.702351, -5.86…|[-6.975427, -6.94…|            15|
|4280fbb7be6b5faf3…|[-1.120947, -1.36…|[-1.120947, -1.36…|             2|
|4dddced897e539971…|[-5.839357, -5.84…|[-5.936493, -6.01…|            15|
|4f67297d191c1d8d1…|[-5.676660, -5.70…|[-7.309140, -6.05…|            15|
+------------------+------------------+------------------+--------------+
only showing top 20 rows
```

[18]:
```
streetMultiIdDf: org.apache.spark.sql.DataFrame = [crimeIdMulti: string]
```

5 Transformations

6 Ingest

[19]:
```
outcomesDf = spark
                    .read
                    .option("header", "true")
                    .csv("./data/*/*-outcomes.csv")
                    .withColumn("districtName", input_file_name())
streetDf = spark
                    .read
                    .option("header", "true")
                    .csv("./data/*/*-street.csv")
                    .withColumn("districtName", input_file_name())
```

[19]:
```
outcomesDf: org.apache.spark.sql.DataFrame = [Crime ID: string, Month: string
… 9 more fields]
streetDf: org.apache.spark.sql.DataFrame = [Crime ID: string, Month: string …
11 more fields]
```

6.1 Select the minimum required fields

[20]:
```
// SELECT
streetDf = streetDf.select($"Crime ID".as("crimeID"),
                           $"districtName",
                           $"Latitude".as("latitude"),
                           $"Longitude".as("longitude"),
                           $"Crime type".as("crimeType"),
                           $"Last outcome category".as("lastOutcomeCategory")
                          )
outcomesDf = outcomesDf.select($"Crime ID".as("crimeId"),
                               $"Outcome type".as("outcomeType")
                              )
```

[20]:
```
streetDf: org.apache.spark.sql.DataFrame = [crimeID: string, districtName:
string … 4 more fields]
outcomesDf: org.apache.spark.sql.DataFrame = [crimeId: string, outcomeType:
string]
```

6.2 Removing duplicate IDs

[21]:
```
// get DF with duplicate IDs
streetMultiIdDf = streetDf.groupBy($"crimeID")
                                .agg(count($"crimeID").as("count"))
                                .where($"count" > 1)
                                .select($"crimeID")
                                .withColumnRenamed("crimeID", "crimeIdMulti")

var outcomesMultiIdDf = outcomesDf.groupBy($"crimeID")
                                .agg(count($"crimeID").as("count"))
                                .where($"count" > 1)
                                .select($"crimeID")
                                .withColumnRenamed("crimeID", "crimeIdMulti")
```

[21]:
```
streetMultiIdDf: org.apache.spark.sql.DataFrame = [crimeIdMulti: string]
outcomesMultiIdDf: org.apache.spark.sql.DataFrame = [crimeIdMulti: string]
```

[22]:
```
// Filter duplicate IDs
streetDf = streetDf.join(streetMultiIdDf, $"crimeID" === $"crimeIdMulti",␣
↪"leftanti")
outcomesDf = outcomesDf.join(outcomesMultiIdDf, $"crimeID" === $"crimeIdMulti",␣
↪"leftanti")
```

[22]:
```
streetDf: org.apache.spark.sql.DataFrame = [crimeID: string, districtName:
string … 4 more fields]
```

```
outcomesDf: org.apache.spark.sql.DataFrame = [crimeId: string, outcomeType:
string]
```

[23]:
```
// Verify the remaining values
println(streetDf.count)
println(outcomesDf.count)
```

```
8029355
4570559
```

6.3 District name

Adding the name of the district

[24]:
```
// extract function
val extractName = udf((path: String) => new File(path).getName().split("-").
 ↪drop(2).dropRight(1).mkString(" "))

// DF with extract
streetDf = streetDf.withColumn("districtName", extractName($"districtName"))

// print
streetDf.distinct.show(5, false)
```

```
+----------------------------------------------------------------+-------------
-+---------+---------+--------------------------+--------------------------
---------------+
|crimeID                                                         |districtName
|latitude |longitude|crimeType                 |lastOutcomeCategory
|
+----------------------------------------------------------------+-------------
-+---------+---------+--------------------------+--------------------------
---------------+
|0006c2f699d8672953293d94af50131ad6b9aee0d5a0c129a8cabc194b7aa912|merseyside
|53.425884|-2.952538|Shoplifting               |Investigation complete; no
suspect identified|
|0007023f792cd822fdb06eeb55e8ba44c96404434af0d957654e3b9c9b73ac1b|west midlands
|52.476815|-1.895378|Shoplifting               |Investigation complete; no
suspect identified|
|00070d3e37d069af5b16fe9f30cd33108f635dfc0738ba3d78af5d428cbb1710|bedfordshire
|52.134816|-0.483361|Public order             |Formal action is not in the
public interest  |
|000bdac925f679da7991f767d63817701e9545d1054793d27a3ea2c4f67775de|north
yorkshire|54.229990|-1.347311|Violence and sexual offences|Formal action is not
in the public interest  |
|001127959c7a7590dbff0bffe61e9480e0583fe5dbaa7376d1b5b1d8595afa5b|gwent
|51.780318|-3.208747|Shoplifting               |Investigation complete; no
```

```
suspect identified|
+--------------------------------------------------------------------+-------------
-+---------+---------+---------------------------+---------------------------
----------------+
only showing top 5 rows
```

[24]:
```
extractName: org.apache.spark.sql.expressions.UserDefinedFunction = SparkUserDef
inedFunction($Lambda$4751/0x0000000840f43040@6aba2420,StringType,List(Some(class
[value[0]: string])),None,true,true)
streetDf: org.apache.spark.sql.DataFrame = [crimeID: string, districtName:
string … 4 more fields]
```

6.4 Join to get final outcome

[25]:
```
val finalDf = streetDf.join(outcomesDf, Seq("crimeID"), "inner")
                      .withColumn("lastOutcome", when($"outcomeType".isNotNull,␣
 ↪$"outcomeType")
                                                                  .
 ↪otherwise($"lastOutcomeCategory")
                                  )
                      .select($"crimeID",
                              $"districtName",
                              $"latitude",
                              $"longitude",
                              $"crimeType",
                              $"lastOutcome"
                             )

finalDf.show
```

```
+--------------------+-----------------+---------+---------+-------------------
-+--------------------+
|              crimeID|     districtName| latitude|longitude|
crimeType|         lastOutcome|
+--------------------+-----------------+---------+---------+-------------------
-+--------------------+
|0006c2f699d867295…|        merseyside|53.425884|-2.952538|
Shoplifting|Investigation com…|
|0007023f792cd822f…|     west midlands|52.476815|-1.895378|
Shoplifting|Investigation com…|
|001cec56bc43aa35a…|            sussex|51.064738|-0.332847|       Bicycle
theft|Investigation com…|
|002163e215119dce0…|devon and cornwall|50.351150|-3.600262|Violence and
sexu…|Unable to prosecu…|
|0024c795b8692eace…|           norfolk|52.485340| 0.519512|Violence and
```

```
sexu…|Action to be take…|
|002c5bd49c4a165f9…|     thames valley|51.458122|-1.476708|Violence and
sexu…|Investigation com…|
|002e2c23238dbbcd6…|            sussex|50.812559|-0.374688|
Burglary|Investigation com…|
|002e5fc85f2d6bf3a…|            durham|54.774222|-1.423850|
Burglary|Unable to prosecu…|
|002fd8cc42b661bfb…|       west mercia|52.388024|-2.251112|Theft from the
pe…|Investigation com…|
|002fe31c42c47d60e…|       northumbria|55.129214|-1.515453|Criminal damage
a…|Investigation com…|
|0031e8ff296a6bdf6…|    cambridgeshire|52.329957|-0.184222|Violence and
sexu…|Unable to prosecu…|
|0032fd3c0c29f917d…|              kent|     null|     null|Violence and
sexu…|Unable to prosecu…|
|00338194bfc04dbea…|     staffordshire|52.626633|-2.151944|Violence and
sexu…|Action to be take…|
|0034028cad2404127…|         cleveland|54.606096|-1.073173|Violence and
sexu…|Unable to prosecu…|
|0037d66b60c933f72…|           cumbria|54.320521|-2.746572|Violence and
sexu…|Investigation com…|
|003ab13aeec5f2eda…|greater manchester|53.514377|-2.351346|Criminal damage
a…|Investigation com…|
|003c19e4547e0f0a1…|      metropolitan|51.547097|-0.009573|         Other
crime|Investigation com…|
|003ed2f68c0f61166…|      metropolitan|51.556772|-0.285145|         Other
theft|Investigation com…|
|0049caaf747c3043d…|devon and cornwall|50.442355|-4.561556|Violence and
sexu…|Unable to prosecu…|
|004b0994414c051c9…|       dyfed powys|51.717747|-5.033104|Criminal damage
a…|Unable to prosecu…|
+--------------------+------------------+---------+---------+------------------
-+--------------------+
only showing top 20 rows
```

[25]:
```
finalDf: org.apache.spark.sql.DataFrame = [crimeID: string, districtName: string
… 4 more fields]
```

7 KPI

[26]:
```
finalDf.groupBy("crimeType")
       .agg(count($"crimeID").as("count"))
       .orderBy($"count".desc)
       .show(false)
```

```
+---------------------------+-------+
|crimeType                  |count  |
+---------------------------+-------+
|Violence and sexual offences|1491597|
|Criminal damage and arson  |533891 |
|Other theft                |468270 |
|Vehicle crime              |400012 |
|Burglary                   |371944 |
|Public order               |361977 |
|Shoplifting                |302199 |
|Drugs                      |105926 |
|Other crime                |79373  |
|Bicycle theft              |76815  |
|Theft from the person      |63594  |
|Robbery                    |56783  |
|Possession of weapons      |29818  |
+---------------------------+-------+
```

[ ]: