Exercise 8

Unstructured Spark exercise

Prior Knowledge
Unix Command Line Shell
Simple Python
Spark Python
Simple SQL syntax

Learning Objectives
Pulling together your skills from previous exercises

Software Requirements
(see separate document for installation of these)

- Apache Spark 3.0.0
- Python 3.x
- Jupyter Notebook

Aim
There is a file on your VM that contains some data about health practices (e.g. GP surgeries) in the UK:

```
~/datafiles/practices/ukpractices2015.csv
```

The CSV file has a header line with titles of each column.

The aim is simple:

I'd like you to calculate the number of practices per postcode prefix for the data. The postcode prefix I define as the first few characters of the postcode up to the space.

How many surgeries were there in the UK in 2015?
What is the average number and standard deviation of surgeries per postcode prefix?
Why is that somewhat misleading?

Please tell me the number of surgeries for the postcode areas: **OX1, SW11**.

We are going to do this locally, NOT on EC2.

There are some hints overleaf.

Hints:

1. Create a new Jupyter Notebook as before (hint: pyspark)

2. Use the same CSV reader from previous exercise to load the data in

   You do NOT need to use HDFS or S3 to store the data.
   You can load directly from the local disk since we are just doing this as an exercise. If you wanted to scale this out, you would need a distributed file system.

   e.g.

```
from pyspark.sql import SQLContext
sqlc = SQLContext(sc)

df = sqlc.read.csv(
'file:///home/oxclo/datafiles/practices/ukpractices2015.csv',
   header='true',
   inferSchema='true')
```

3. You should know enough to do this:

   a. either as a set of Map/ReduceByKey operations. You could also look at countByKey

   b. Alternatively, you can do this all in SQL if you like SQL.

   c. You could also try both approaches!

4. There are lots of built in functions in Spark SQL:

   https://spark.apache.org/docs/3.0.0/api/sql/

5. You might want to add Python logic into the SQL world via a "user-defined function".

   https://docs.databricks.com/spark/latest/spark-sql/udf-python.html

6. If you like to mix and match SQL and Map/Reduce you can do that too. I've shown you how to do DataFrame → RDD. The following page shows you how to do RDD → DataFrame:

   https://spark.apache.org/docs/3.0.0/sql-programming-guide.html#interoperating-with-rdds

7. Ask me or the TA if you get stuck.