ASSIGNMENT 1

PROJECT GROUP - 17
SPARK USE CASE - 12

NOTEBOOK LINK - https://goo.gl/emKk8l

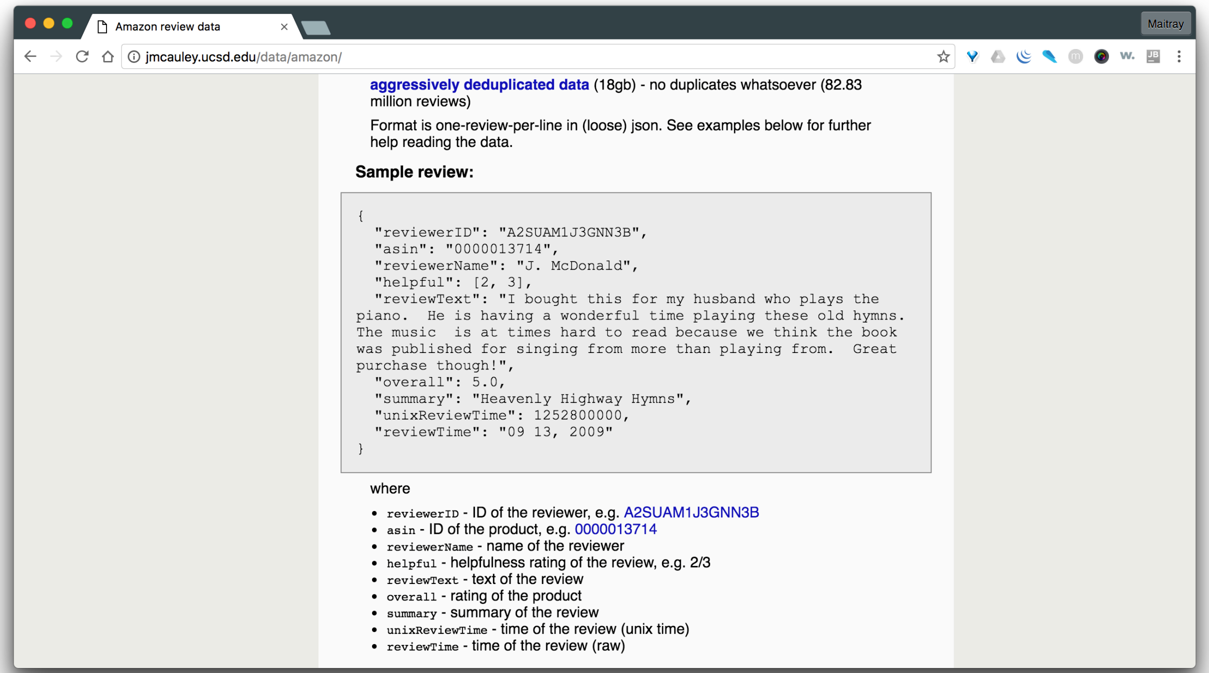
TEAM MEMBERS
KARAN SHAH
MAITRAY SHAH
SALMAAN PEHLARI
KELLY ALVAREZ AGUIRRE

DESCRIPTION

Fraud detection in user reviews
(use this data: https://snap.stanford.edu/data/web-Amazon.html

DATA SOURCE

Sample Data



aggressively deduplicated data (18gb) - no duplicates whatsoever (82.83 million reviews)

Format is one-review-per-line in (loose) json. See examples below for further help reading the data.

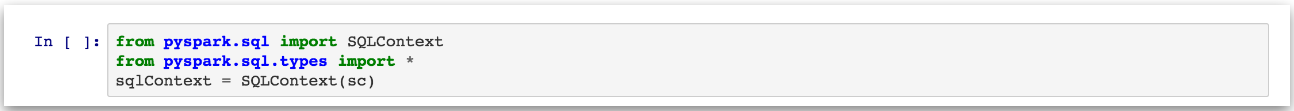
Sample review:

```
{
  "reviewerID": "A2SUAM1J3GNN3B",
  "asin": "0000013714",
  "reviewerName": "J. McDonald",
  "helpful": [2, 3],
  "reviewText": "I bought this for my husband who plays the
piano.  He is having a wonderful time playing these old hymns.
The music  is at times hard to read because we think the book
was published for singing from more than playing from.  Great
purchase though!",
  "overall": 5.0,
  "summary": "Heavenly Highway Hymns",
  "unixReviewTime": 1252800000,
  "reviewTime": "09 13, 2009"
}
```

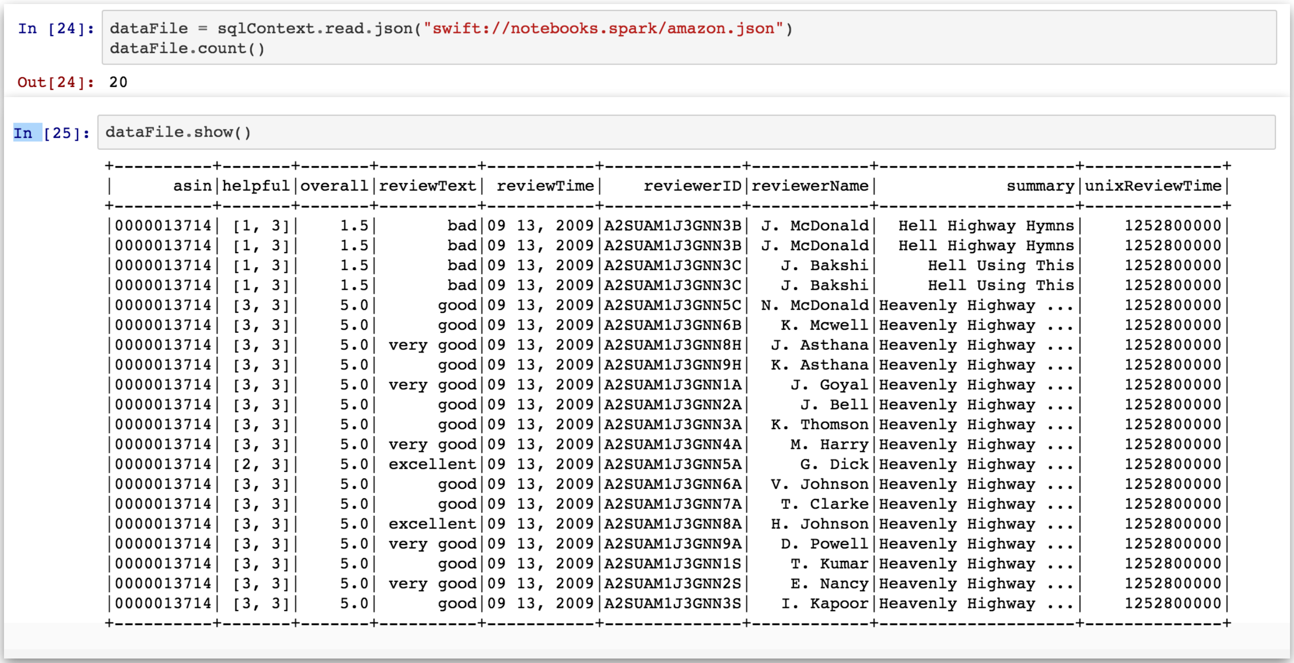
where

- `reviewerID` - ID of the reviewer, e.g. A2SUAM1J3GNN3B
- `asin` - ID of the product, e.g. 0000013714
- `reviewerName` - name of the reviewer
- `helpful` - helpfulness rating of the review, e.g. 2/3
- `reviewText` - text of the review
- `overall` - rating of the product
- `summary` - summary of the review
- `unixReviewTime` - time of the review (unix time)
- `reviewTime` - time of the review (raw)

JUPYTER NOTEBOOK



```
In [ ]: from pyspark.sql import SQLContext
        from pyspark.sql.types import *
        sqlContext = SQLContext(sc)
```



```
In [24]: dataFile = sqlContext.read.json("swift://notebooks.spark/amazon.json")
         dataFile.count()

Out[24]: 20

In [25]: dataFile.show()

+----------+-------+-------+----------+-----------+--------------+------------+--------------------+--------------+
|      asin|helpful|overall|reviewText| reviewTime|    reviewerID|reviewerName|             summary|unixReviewTime|
+----------+-------+-------+----------+-----------+--------------+------------+--------------------+--------------+
|0000013714| [1, 3]|    1.5|       bad|09 13, 2009|A2SUAM1J3GNN3B| J. McDonald|  Hell Highway Hymns|    1252800000|
|0000013714| [1, 3]|    1.5|       bad|09 13, 2009|A2SUAM1J3GNN3B| J. McDonald|  Hell Highway Hymns|    1252800000|
|0000013714| [1, 3]|    1.5|       bad|09 13, 2009|A2SUAM1J3GNN3C|   J. Bakshi|     Hell Using This|    1252800000|
|0000013714| [1, 3]|    1.5|       bad|09 13, 2009|A2SUAM1J3GNN3C|   J. Bakshi|     Hell Using This|    1252800000|
|0000013714| [3, 3]|    5.0|      good|09 13, 2009|A2SUAM1J3GNN5C| N. McDonald|Heavenly Highway ...|    1252800000|
|0000013714| [3, 3]|    5.0|      good|09 13, 2009|A2SUAM1J3GNN6B|   K. Mcwell|Heavenly Highway ...|    1252800000|
|0000013714| [3, 3]|    5.0| very good|09 13, 2009|A2SUAM1J3GNN8H|  J. Asthana|Heavenly Highway ...|    1252800000|
|0000013714| [3, 3]|    5.0|      good|09 13, 2009|A2SUAM1J3GNN9H|  K. Asthana|Heavenly Highway ...|    1252800000|
|0000013714| [3, 3]|    5.0| very good|09 13, 2009|A2SUAM1J3GNN1A|    J. Goyal|Heavenly Highway ...|    1252800000|
|0000013714| [3, 3]|    5.0|      good|09 13, 2009|A2SUAM1J3GNN2A|     J. Bell|Heavenly Highway ...|    1252800000|
|0000013714| [3, 3]|    5.0|      good|09 13, 2009|A2SUAM1J3GNN3A|  K. Thomson|Heavenly Highway ...|    1252800000|
|0000013714| [3, 3]|    5.0| very good|09 13, 2009|A2SUAM1J3GNN4A|    M. Harry|Heavenly Highway ...|    1252800000|
|0000013714| [2, 3]|    5.0| excellent|09 13, 2009|A2SUAM1J3GNN5A|     G. Dick|Heavenly Highway ...|    1252800000|
|0000013714| [3, 3]|    5.0|      good|09 13, 2009|A2SUAM1J3GNN6A|  V. Johnson|Heavenly Highway ...|    1252800000|
|0000013714| [3, 3]|    5.0|      good|09 13, 2009|A2SUAM1J3GNN7A|   T. Clarke|Heavenly Highway ...|    1252800000|
|0000013714| [3, 3]|    5.0| excellent|09 13, 2009|A2SUAM1J3GNN8A|  H. Johnson|Heavenly Highway ...|    1252800000|
|0000013714| [3, 3]|    5.0| very good|09 13, 2009|A2SUAM1J3GNN9A|   D. Powell|Heavenly Highway ...|    1252800000|
|0000013714| [3, 3]|    5.0|      good|09 13, 2009|A2SUAM1J3GNN1S|    T. Kumar|Heavenly Highway ...|    1252800000|
|0000013714| [3, 3]|    5.0| very good|09 13, 2009|A2SUAM1J3GNN2S|    E. Nancy|Heavenly Highway ...|    1252800000|
|0000013714| [3, 3]|    5.0|      good|09 13, 2009|A2SUAM1J3GNN3S|   I. Kapoor|Heavenly Highway ...|    1252800000|
+----------+-------+-------+----------+-----------+--------------+------------+--------------------+--------------+
In [26]: dataFile.printSchema()

root
 |-- asin: string (nullable = true)
 |-- helpful: array (nullable = true)
 |    |-- element: long (containsNull = true)
 |-- overall: double (nullable = true)
 |-- reviewText: string (nullable = true)
 |-- reviewTime: string (nullable = true)
 |-- reviewerID: string (nullable = true)
 |-- reviewerName: string (nullable = true)
 |-- summary: string (nullable = true)
 |-- unixReviewTime: long (nullable = true)
```

```
In [27]: dataFile
         dataFile.registerTempTable("User_Reviews")
```

```
In [28]: df = sqlContext.sql("select reviewerID, reviewerName, reviewText, overall, reviewTime from User_Reviews where overall
          <=1.5")
         df.show()

         +--------------+------------+----------+-------+-----------+
         |    reviewerID|reviewerName|reviewText|overall| reviewTime|
         +--------------+------------+----------+-------+-----------+
         |A2SUAM1J3GNN3B| J. McDonald|       bad|    1.5|09 13, 2009|
         |A2SUAM1J3GNN3B| J. McDonald|       bad|    1.5|09 13, 2009|
         |A2SUAM1J3GNN3C|   J. Bakshi|       bad|    1.5|09 13, 2009|
         |A2SUAM1J3GNN3C|   J. Bakshi|       bad|    1.5|09 13, 2009|
         +--------------+------------+----------+-------+-----------+
```

```
In [29]: df = sqlContext.sql("select reviewerID from User_Reviews where overall <=1.5")
         df.show()

         +--------------+
         |    reviewerID|
         +--------------+
         |A2SUAM1J3GNN3B|
         |A2SUAM1J3GNN3B|
         |A2SUAM1J3GNN3C|
         |A2SUAM1J3GNN3C|
         +--------------+
```

```
In [30]: df = sqlContext.sql("select reviewerID, asin as productid, overall from  User_Reviews where overall <=1.5 group by rev
         iewerID, asin , overall")
         df.show()

         +--------------+----------+-------+
         |    reviewerID| productid|overall|
         +--------------+----------+-------+
         |A2SUAM1J3GNN3C|0000013714|    1.5|
         |A2SUAM1J3GNN3B|0000013714|    1.5|
         +--------------+----------+-------+
```

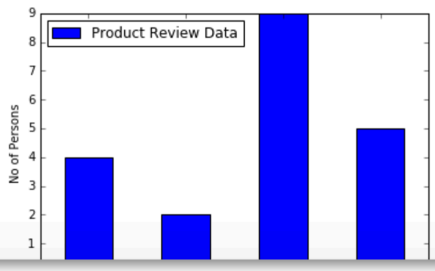
```
In [31]: df = sqlContext.sql("select count(*) as totalpersons , reviewText from User_Reviews group by reviewText")
         df.show()

         +------------+----------+
         |totalpersons|reviewText|
         +------------+----------+
         |           4|       bad|
         |           2| excellent|
         |           9|      good|
         |           5| very good|
         +------------+----------+
In [32]: import pandas as pd
         import numpy as np
         import matplotlib.pyplot as plt
         import urllib2
         pd.__version__
         %matplotlib inline

         pd.pdf_name=df.toPandas()
```

```
In [33]: name_list=pd.pdf_name['reviewText'].head(15).tolist()
         name_amounts=pd.pdf_name['totalpersons'].head(15).tolist()
         data_names = {'Product Review Data':pd.Series(name_amounts, index=name_list)}
         df_name_plot = pd.DataFrame(data_names)
         ax = df_name_plot.plot(kind='bar')
         ax.set_xlabel("Reviews")
         ax.set_ylabel("No of Persons")

Out[33]: <matplotlib.text.Text at 0x7f858816d410>
```


Product Review Data
No of Persons