

databricks Processing_Log_Files

Reviewing Log Files

```
display(dbutils.fs.ls("/databricks-datasets/learning-spark/data-001/fake_logs"))
```

path
dbfs:/databricks-datasets/learning-spark/data-001/fake_logs/log1.log
dbfs:/databricks-datasets/learning-spark/data-001/fake_logs/log2.log



```
%python
print(dbutils.fs.head("/databricks-datasets/learning-spark/data-001/fake_logs/log1.log"))
```

```
66.249.69.97 - - [24/Sep/2014:22:25:44 +0000] "GET /071300/242153 HTTP/1.1" 404
514 "-" "Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.htm
l)"
71.19.157.174 - - [24/Sep/2014:22:26:12 +0000] "GET /error HTTP/1.1" 404 505 "-"
" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrom
e/37.0.2062.94 Safari/537.36"
71.19.157.174 - - [24/Sep/2014:22:26:12 +0000] "GET /favicon.ico HTTP/1.1" 200
1713 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Geck
o) Chrome/37.0.2062.94 Safari/537.36"
71.19.157.174 - - [24/Sep/2014:22:26:37 +0000] "GET / HTTP/1.1" 200 18785 "-"
"Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrom
e/37.0.2062.94 Safari/537.36"
71.19.157.174 - - [24/Sep/2014:22:26:37 +0000] "GET /jobmineimg.php?q=m HTTP/1.
1" 200 222 "http://www.holdenkarau.com/" "Mozilla/5.0 (X11; Linux x86_64) Apple
WebKit/537.36 (KHTML, like Gecko) Chrome/37.0.2062.94 Safari/537.36"
```

```
print(dbutils.fs.head("/databricks-datasets/learning-spark/data-001/fake_logs/log2.log"))
```

```
71.19.157.174 - - [24/Sep/2014:22:26:12 +0000] "GET /error78978 HTTP/1.1" 404 5
05 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/37.0.2062.94 Safari/537.36"
```

```
log1 = sc.textFile("/databricks-datasets/learning-spark/data-001/fake_logs/log1.log")
log2 = sc.textFile("/databricks-datasets/learning-spark/data-001/fake_logs/log1.log")
```

```
log1.count()
```

```
Out[18]: 5
```

```
log2.count()
```

```
Out[19]: 5
```

Parsing Log Files

```
from pyspark.sql import Row
import datetime
def parse_log(logline):
    line = [x.strip('"[]') for x in logline.split(" ")]
    return Row(remote_ip=line[0], client_id=line[1], user_id=line[2],
log_time=line[3], request_type=line[5],
request_path=line[6]+line[7], status=line[8], bytes_sent=line[9],
http_referer=line[10], http_user_agent=" ".join(line[11:]), raw_log_text=line,
created_time=datetime.datetime.now().strftime("%Y-%m-%dT%H:%M:%S"))
```

```
log1_mapped = log1.map(lambda line: parse_log(line))
log1_mapped.collect()
```

```
Out[21]:
[Row(bytes_sent=u'514', client_id=u'-' , created_time='2017-10-04T00:16:01', ht
tp_referer=u'-' , http_user_agent=u'Mozilla/5.0 (compatible; Googlebot/2.1; +ht
tp://www.google.com/bot.html)', log_time=u'24/Sep/2014:22:25:44', raw_log_text
=[u'66.249.69.97', u'-' , u'-' , u'24/Sep/2014:22:25:44', u'+0000', u'GET', u'/0
71300/242153', u'HTTP/1.1', u'404', u'514', u'-' , u'Mozilla/5.0', u'(compatibl
e;', u'Googlebot/2.1;', u'+http://www.google.com/bot.html)'], remote_ip=u'66.2
49.69.97', request_path=u'/071300/242153HTTP/1.1', request_type=u'GET', status
=u'404', user_id=u'-' ),
Row(bytes_sent=u'505', client_id=u'-' , created_time='2017-10-04T00:16:01', ht
tp_referer=u'-' , http_user_agent=u'Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/37.0.2062.94 Safari/537.36', log_time=u'2
4/Sep/2014:22:26:12', raw_log_text=[u'71.19.157.174', u'-' , u'-' , u'24/Sep/201
4:22:26:12', u'+0000', u'GET', u'/error', u'HTTP/1.1', u'404', u'505', u'-' ,
u'Mozilla/5.0', u'(X11;', u'Linux', u'x86_64)', u'AppleWebKit/537.36', u'(KHT
ML,', u'like', u'Gecko)', u'Chrome/37.0.2062.94', u'Safari/537.36'], remote_ip
```

```
=u'71.19.157.174', request_path=u'/errorHTTP/1.1', request_type=u'GET', status
=u'404', user_id=u'-'),
  Row(bytes_sent=u'1713', client_id=u'-', created_time='2017-10-04T00:16:01', h
ttp_referer=u'-', http_user_agent=u'Mozilla/5.0 (X11; Linux x86_64) AppleWebKi
t/537.36 (KHTML, like Gecko) Chrome/37.0.2062.94 Safari/537.36', log_time=u'2
```

Creating Dataframe and Registering Table

```
log1_df = spark.createDataFrame(log1_mapped)
log1_df.registerTempTable("log1_table")
```

```
%sql select * from log1_table
```

bytes_sent	client_id	created_time	http_referer	http_user_agent
514	-	2017-10-04T00:16:02	-	Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.htm
505	-	2017-10-04T00:16:02	-	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, lik Gecko) Chrome/37.0.2062.94 Safari/537.36
1713	-	2017-10-04T00:16:02	-	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, lik Gecko) Chrome/37.0.2062.94 Safari/537.36



```
log2_mapped = log2.map(lambda line: parse_log(line))
log2_mapped.collect()
```

Out[23]:

```
[Row(bytes_sent=u'514', client_id=u'-', created_time='2017-10-04T00:16:02', ht
tp_referer=u'-', http_user_agent=u'Mozilla/5.0 (compatible; Googlebot/2.1; +ht
tp://www.google.com/bot.html)', log_time=u'24/Sep/2014:22:25:44', raw_log_text
=[u'66.249.69.97', u'-', u'-', u'24/Sep/2014:22:25:44', u'+0000', u'GET', u'/0
71300/242153', u'HTTP/1.1', u'404', u'514', u'-', u'Mozilla/5.0', u'(compatibl
e;', u'Googlebot/2.1;', u'+http://www.google.com/bot.html)'], remote_ip=u'66.2
49.69.97', request_path=u'/071300/242153HTTP/1.1', request_type=u'GET', status
=u'404', user_id=u'-'),
  Row(bytes_sent=u'505', client_id=u'-', created_time='2017-10-04T00:16:02', ht
tp_referer=u'-', http_user_agent=u'Mozilla/5.0 (X11; Linux x86_64) AppleWebKi
t/537.36 (KHTML, like Gecko) Chrome/37.0.2062.94 Safari/537.36', log_time=u'2
```

```
4/Sep/2014:22:26:12', raw_log_text=[u'71.19.157.174', u'-', u'-', u'24/Sep/2014:22:26:12', u'+0000', u'GET', u'/error', u'HTTP/1.1', u'404', u'505', u'-', u'Mozilla/5.0', u'(X11;', u'Linux', u'x86_64)', u'AppleWebKit/537.36', u'(KHTML, u'like', u'Gecko)', u'Chrome/37.0.2062.94', u'Safari/537.36'], remote_ip=u'71.19.157.174', request_path=u'/errorHTTP/1.1', request_type=u'GET', status=u'404', user_id=u'-'),
  Row(bytes_sent=u'1713', client_id=u'-', created_time='2017-10-04T00:16:02', http_referer=u'-', http_user_agent=u'Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/37.0.2062.94 Safari/537.36')])
```

```
log2_df = spark.createDataFrame(log2_mapped)
log2_df.registerTempTable("log2_table")
```

```
%sql select * from log2_table
```

bytes_sent	client_id	created_time	http_referer	http_user_agent
514	-	2017-10-04T00:16:02	-	Mozilla/5.0 (compatible; Googlebot/2.1; +http://www.google.com/bot.htm
505	-	2017-10-04T00:16:02	-	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/37.0.2062.94 Safari/537.36
1713	-	2017-10-04T00:16:02	-	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/37.0.2062.94 Safari/537.36



```
%sql
select count(distinct remote_ip)
from log2_table
```

count(DISTINCT remote_ip)
2



```
import datetime
def format_time(time_text):
    parsed_time = datetime.datetime.strptime(time_text, '%d/%b/%Y:%H:%M:%S')
    return datetime.datetime.strftime(parsed_time, "%d-%m-%Y")

sqlContext.udf.register("format_log_time", format_time)
```

Counting Unique IPs by Date

```
%sql
select format_log_time(log_time), count(distinct remote_ip) as unique_ip
from
(select log_time, remote_ip
from log1_table
union
select log_time, remote_ip
from log2_table)
group by 1
```



```
%sql
select log_time, count(distinct remote_ip) as unique_ip
from
(select log_time, remote_ip
from log1_table
union
select log_time, remote_ip
from log2_table)
group by 1
```

