|  |
| --- |
|  |
|  |
| **C**S 779, ADVANCED DATABASE MANAGEMENT  DHARMIT DALVI  DECEMBER 2019 |

NASA WEB SERVER LOGS’ ANALYSIS USING APACHE SPARK

Table of Contents

[**Introduction** 4](#_Toc27270549)

[**Apache Spark** 4](#_Toc27270550)

[The Spark Ecosystem 5](#_Toc27270551)

[Spark Data Structures 5](#_Toc27270552)

[Databricks 6](#_Toc27270553)

[**Web server logs’ analysis** 6](#_Toc27270554)

[Dataset 7](#_Toc27270555)

[**Data Wrangling** 7](#_Toc27270556)

[NCSA common log format 7](#_Toc27270557)

[Loading the data 8](#_Toc27270558)

[Regular expressions 9](#_Toc27270559)

[**Analysis of logs** 13](#_Toc27270560)

[Finding Frequent Hosts 13](#_Toc27270561)

[Finding number of unique hosts 14](#_Toc27270562)

[Finding distinct endpoints and the number of times they were requested 14](#_Toc27270563)

[Finding top ten error endpoints 15](#_Toc27270564)

[Finding all logs for the “not-found” error 15](#_Toc27270565)

[Analyzing and visualizing distinct status code values and the number of times they have occurred 15](#_Toc27270566)

[Content size analysis 16](#_Toc27270567)

[Finding distinct hosts on a particular day 16](#_Toc27270568)

[**Data Visualization** 19](#_Toc27270569)

[Tableau 19](#_Toc27270570)

[**Learnings from the project** 20](#_Toc27270571)

[**Further scope** 20](#_Toc27270572)

[**References** 20](#_Toc27270573)

Table of Figures

[Figure 1: Overview of the project 4](#_Toc27270574)

[Figure 2: Spark Ecosystem (“Apache SparkTM—What is Spark. (n.d.)”) 5](#_Toc27270575)

[Figure 3: Databricks platform home page 6](#_Toc27270576)

[Figure 4: Log file for NASA's web server 7](#_Toc27270577)

[Figure 5: NCSA common log format(stevewhims. (n.d.). NCSA Logging—Win32 apps.) 8](#_Toc27270578)

[Figure 6: Log data loaded as a Spark DataFrame 9](#_Toc27270579)

[Figure 7: 10 sample logs from the entire data 9](#_Toc27270580)

[Figure 8: extracting hosts from the logs using regex pattern 10](#_Toc27270581)

[Figure 9:Extracting timestamps using regex 11](#_Toc27270582)

[Figure 10: Extracting methods from logs 11](#_Toc27270583)

[Figure 11: Extracting status from logs 12](#_Toc27270584)

[Figure 12: Extracting content size from logs 12](#_Toc27270585)

[Figure 13: Extracting all fields from the log data 12](#_Toc27270586)

[Figure 14: Structured dataframe with separate columns as fields 13](#_Toc27270587)

[Figure 15: Converting dataframe to a temporary view 13](#_Toc27270588)

[Figure 16: Finding frequent hosts 14](#_Toc27270589)

[Figure 17: Finding number of unique hosts 14](#_Toc27270590)

[Figure 18: Distinct endpoints and count of how many times they were requested 14](#_Toc27270591)

[Figure 19: Top ten error endpoints 15](#_Toc27270592)

[Figure 20: All logs with status code 404 15](#_Toc27270593)

[Figure 21: Distinct status codes in the logs 16](#_Toc27270594)

[Figure 22: Content size statistics 16](#_Toc27270595)

[Figure 23: Converting timestamp string to a python datetime object 17](#_Toc27270596)

[Figure 24: Converting Spark dataframe to a temporary view 17](#_Toc27270597)

[Figure 25: dataframe with hosts and the day and the month they made requests on 18](#_Toc27270598)

[Figure 26: Creating a dataframe of distinct hosts out of our previous dataframe 18](#_Toc27270599)

[Figure 27: Dataframe of distinct hosts converted to a view 18](#_Toc27270600)

[Figure 28: distinct hosts on 21st July 19](#_Toc27270601)

[Figure 29: Visualizing hosts, their status codes and content sizes in Tableau. 20](#_Toc27270602)

# **Introduction**

Big data analytics is the process of scrutinizing humongous amounts of data, to extract some patterns, derive relations, and gain other insights. It is a field of utmost importance for organizations today; most organizations now understand that they can capture data and analyze it to gain significant value from it. The major benefits of big data analysis are the speed and efficiency it provides.

In this report, we discuss the use case of analyzing web server logs, to find some useful insights from the set of log entries in a log file.

We use Apache Spark for the analysis, which is an extremely fast data analytics engine. Apache Spark is now a leading technology in the field of big data processing, and in this report, we explore various functionalities of Spark, specifically for analyzing all the semi-structured logs’ data. Most of the analysis have been performed using DataFrame operations in PySpark. The report looks into reading two log files, loading the data into Spark, converting the semi-structured log data into a structured dataset and then analyzing and visualizing it using PySpark and SQL.

We further discuss about regular expressions, and how they are used to parse data. We use some regular expressions on our semi-structured logs’ data, and convert them into a structured dataset.

Lastly, we discuss data visualization, and visualize our data using Tableau.

Log files

Analysis using SQL queries and PySpark

Structured dataset

LLLo

Visualization using tableau

Figure : Overview of the project

# **Apache Spark**

Apache Spark is an analytics engine, used for big data processing. Built by UC Berkeley’s AMPLab, it is an open source project, rather the largest open source framework for Big Data Analytics.

Speed: Spark processes data in parallel, hence it can run workloads a 100 times faster.

Ease of use: Spark offers various APIs, making it very easy to write applications quickly and easily in Java, Scala, Python, R and SQL.

Generality: Spark runs on Hadoop, Kubernetes, in the cloud, or even standalone. It can also access from a wide range of data sources. (“Apache SparkTM—Unified Analytics Engine for Big Data. (n.d.)”)

## The Spark Ecosystem

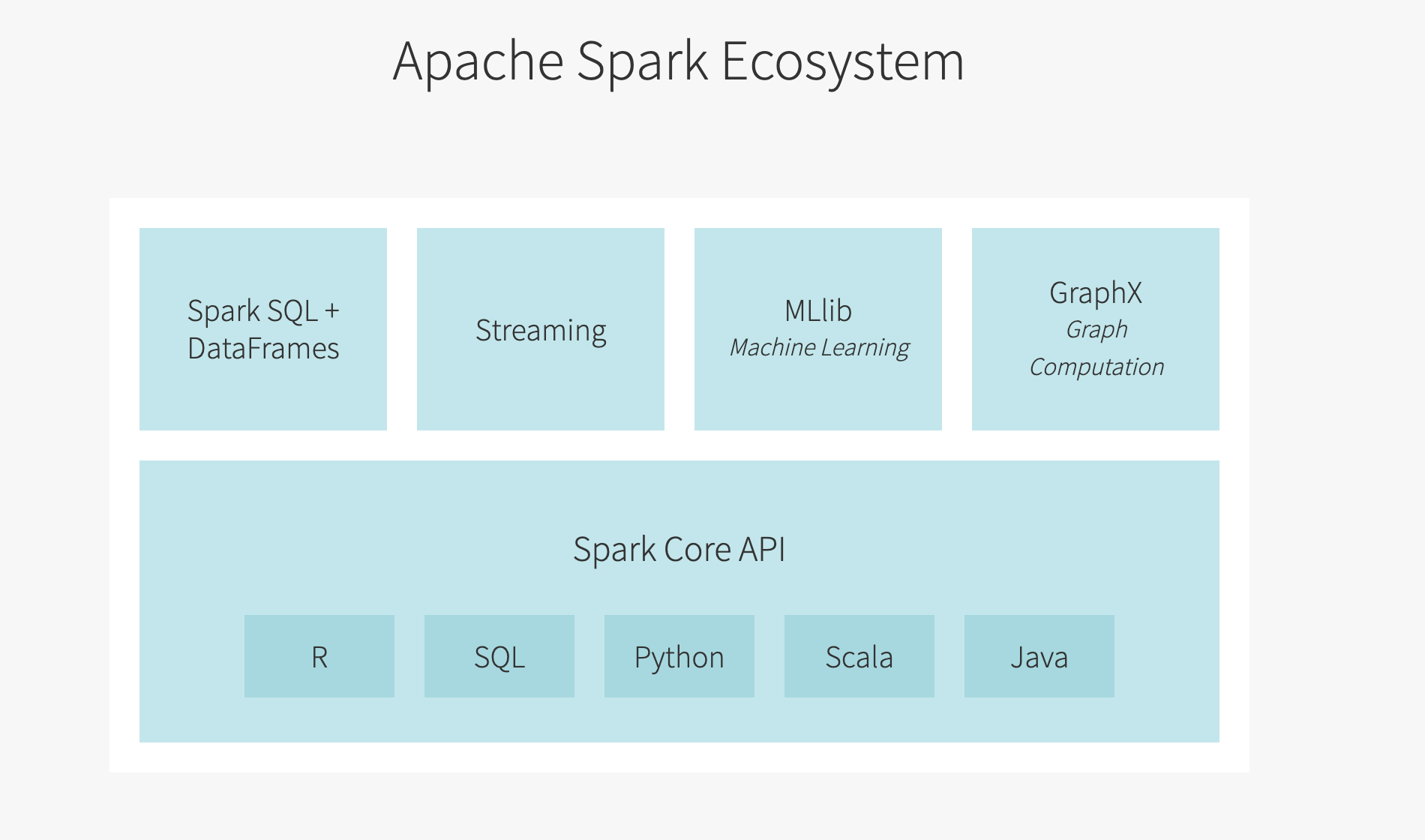
****

Figure : Spark Ecosystem (“Apache SparkTM—What is Spark. (n.d.)”)

The components of Spark ecosystem are as follows:

Spark Core: Spark Core API is the underlying general execution engine for Spark. All other functionality is built on top of this core API. It provides APIs in Python, Java and Scala, which makes development very easy. It further delivers in memory computing capabilities, in order to provide speed.

Spark SQL: Spark SQL is the module for structured data processing. It provides a data structure called dataframes, which we will be exploring in detail as a part of this report.

Spark Streaming: Many applications need the ability to process stream data as well, apart from batch data. The Spark Streaming component provides that functionality.

MLlib for Machine Learning: MLlib is the scalable Machine Learning library in Spark that provides Machine Learning algorithms. It can be used in Python, Scala and Java.

GraphX: This module is used for graph computations. We can work on graph structured data using this module of Spark. (“Apache SparkTM—What is Spark. (n.d.)”)

As a part of this project, we will be utilizing the Spark SQL component of Spark.

## Spark Data Structures

There are three basic data structures in Spark, as follows:

RDDs: RDDs stand for Resilient Distributed Databases. It is a collection of data elements spread across many machines in the cluster.

DataFrames: Dataframes comprise of data organized in columns, just like an RDBMS table.

Datasets: Dataset is an extension of dataframes, that provides the functionality of an Object Oriented Programming interface.

As a part of this report, we are extensively exploring the DataFrames data structure.

## Databricks

Databricks is a platform founded by the original creators of Spark. It is an automated web-based platform, that can be used to perform Spark computations. It provides automated cluster management and ipython notebook style format. The advantage of which being, you can implement your code module by module, and don’t need to implement your entire program at once. One can perform computations in various languages such as Java, Python, SQL, Scala, etc., with just a “%” operator. For instance, to write code in SQL, we use “%sql” and then write the SQL query.

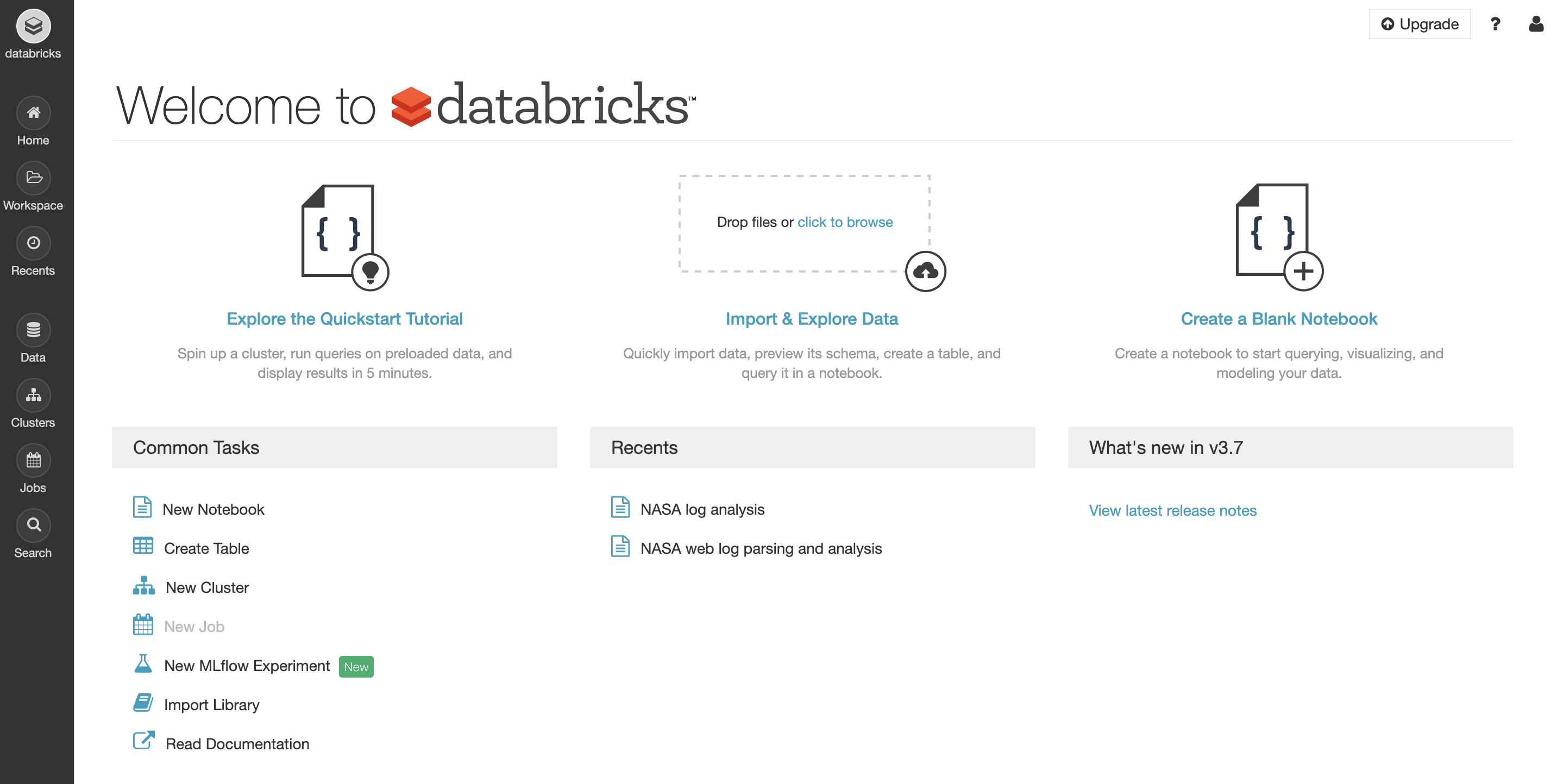


Figure : Databricks platform home page

# **Web server logs’ analysis**

Logs’ analysis is the process of collecting, parsing and analyzing log files generated by web servers. This is done in order to extract information from the logs, and gain some useful insights on what is happening on the system. (Server Log Analysis with the ELK Stack. (2016, December 6).)

Computer systems store a set of logs, that are basically a documentation of activities happening on the system. Log analysis is an evaluation of these logs, in order to get a clearer idea of what’s happening on the system.

Logs are usually created by network devices, applications, operating systems, and programmable or smart devices. They comprise of several messages that are arranged in the order of the requests made to the server. Logs contain information on when requests to the server were made, where it’s from, what specific pages were requested, response codes, etc.

Log analysis is a crucial activity for server administrators, who can use the analysis to monitor activities happening over the system.

Following are certain applications of log analysis:

* To ensure compliance with security policies and regulations
* To find and keep track of any malicious activities or attacks happening on the system
* To troubleshoot systems
* To understand behavior of users trying to request to the server.

(What is Log Analysis? Use Cases, Best Practices, and More | Digital Guardian. (n.d.).)

## Dataset

The dataset under consideration for this study is NASA’s web server logs. We have NASA Kennedy Space Center’s publicly available server logs for two months: July ‘95 and August ’95. The data contains two months-worth of all HTTP requests to the server.

The data is basically two log files containing log entries.

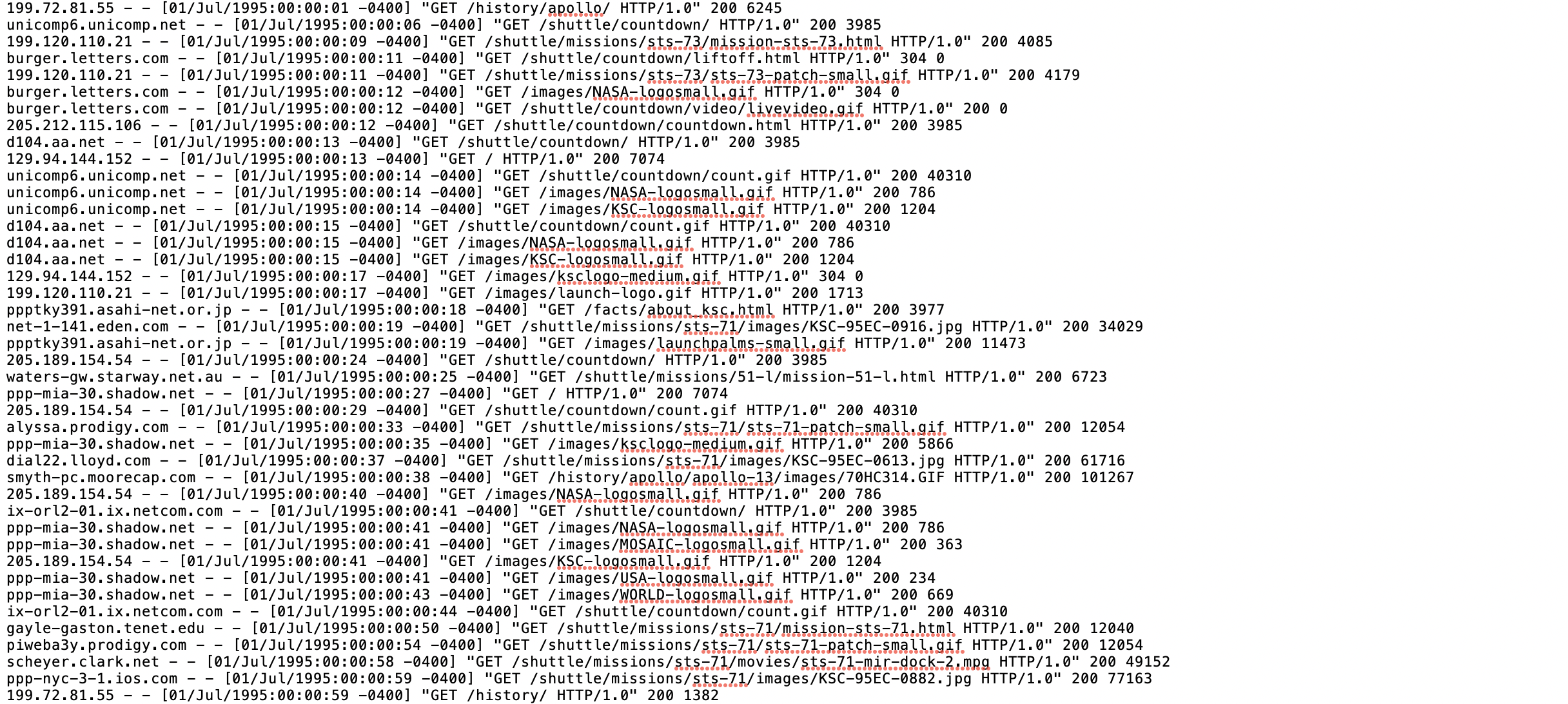


Figure : Log file for NASA's web server

The above figure shows how the data is basically a chunk of logs. Each entry in the above data is a log that indicates what activity happened on the system. This data can be called semi-structured, wherein you can group the whole chunk of logs into a single column.

# **Data Wrangling**

In order to perform any analysis, we would have to convert the semi-structured log data into a structured dataset. Our aim is to convert the log data into a table just like an RDBMS table, where each column in our dataset is a field in the log entry. For that, we need to extract attributes from each log message.

## NCSA common log format

The NCSA Common log file format is a fixed ASCII format. This format is used to provide a structure to the logs, and divide the log message into various fields. Our aim is to structure our logs’ data into various attributes using the same format. (stevewhims. (n.d.). NCSA Logging—Win32 apps.)

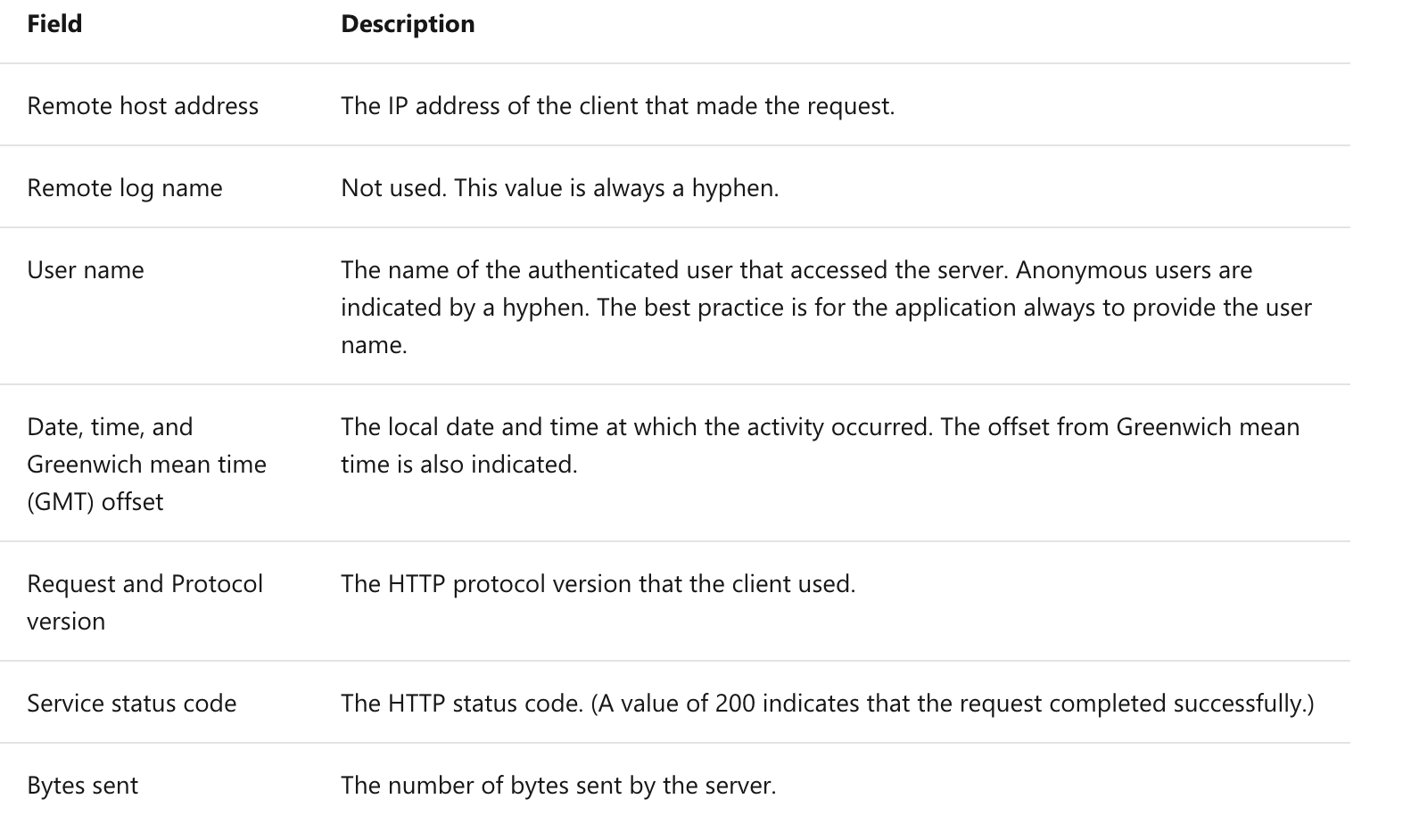


Figure : NCSA common log format(stevewhims. (n.d.). NCSA Logging—Win32 apps.)

Let us consider a single log message from the log file, which is as follows:

199.72.81.55 - - [01/Jul/1995:00:00:01 -0400] "GET /history/apollo/ HTTP/1.0" 200 6245

The first field, “199.72.81.55” is the IP address of the client that has made this request, the host address.

“[01/Jul/1995:00:00:01 -0400]” is the timestamp, which tells us the date and time when this request was made.

"GET /history/apollo/ HTTP/1.0" contains the method requested (GET), the endpoint requested (/history/apollo/) and the protocol the client used (HTTP/1.0)

“200” represents the status code, status code tells us whether the request was successful or no, and if it was not successful, what was the error. Each status code has a message associated with it.

“6245” represents the content side, that is, the number of bytes sent by the server.

Our aim is to parse our log messages according to these fields.

## Loading the data

We first load the two log files in Spark- logs for July and August, and store the data as a Spark DataFrame.



Figure : Log data loaded as a Spark DataFrame

In the figure above, we can see that all of the log messages are put into a single column named “value”, the datatype of which is a string. The data structure created here is a spark dataframe.. Let’s take a sample of 10 logs out of all the log messages, to try out a few regex combinations:



Figure : 10 sample logs from the entire data

## Regular expressions

A regular expression is a series of characters that can be used to define a particular search pattern. It is mainly used for pattern matching.

Regular expressions are a way to match patterns with sequences of characters. They are useful in extracting information from any text. (How to write Regular Expressions? - GeeksforGeeks. (n.d.).)

We use regular expressions to extract required fields from our logs, and then put it all together to make a structured dataset. We have sample logs taken out from the data (Figure 7), we intend to try out a few regular expressions on them to extract required fields.

Let us try building a host pattern on our sample logs to extract only the hosts from our logs:

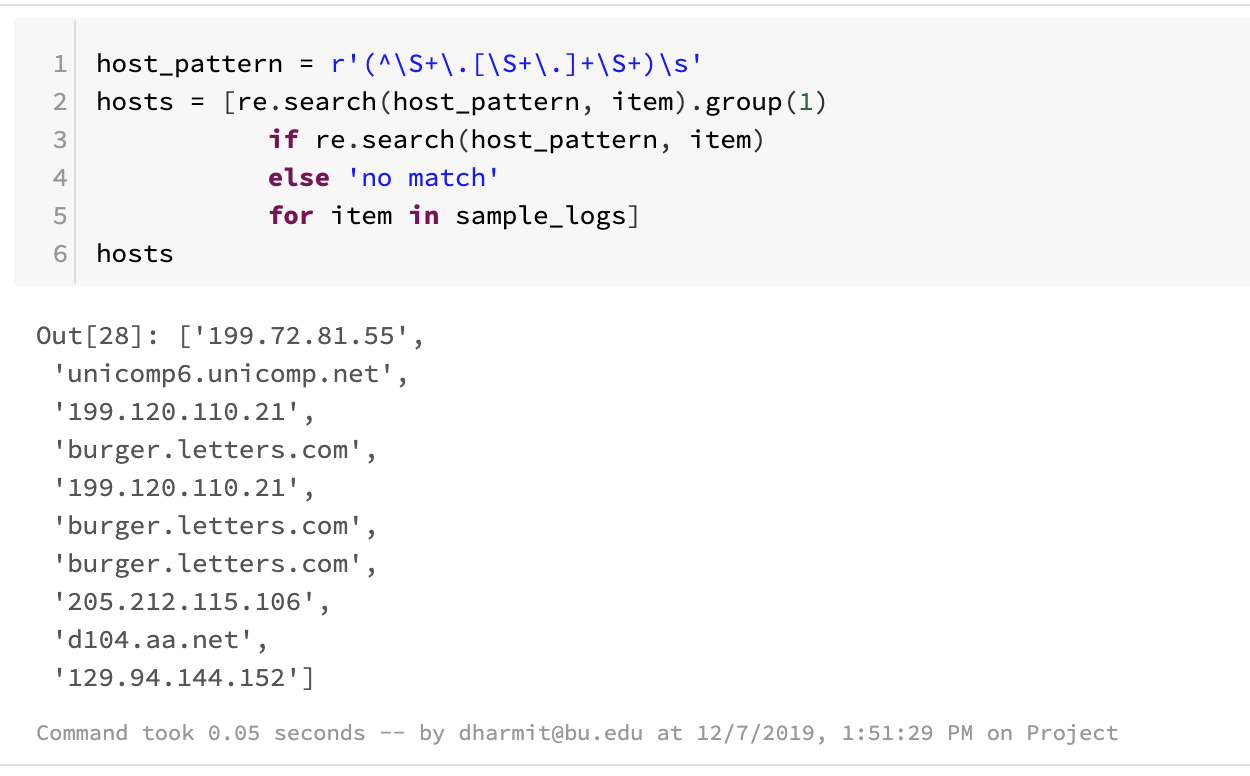


Figure 8: extracting hosts from the logs using regex pattern

The “^” symbol indicates that the match must start at the beginning of the line, “\S” indicates any non-whitespace characters, “+” symbol indicates that the pattern could be repeated one or more times, “.” is for the periods between the IP address fields, and the “\s” at the end selects whitespace that occurs at the end of the pattern. The group() function with “1” as the parameter would drop the whitespace and select only the first pattern, resulting in the extraction of just the IP addresses from our logs, as seen in the output in figure 8.

Similarly, we create a pattern for timestamps, method, status and content size as well, and extract them from the logs:

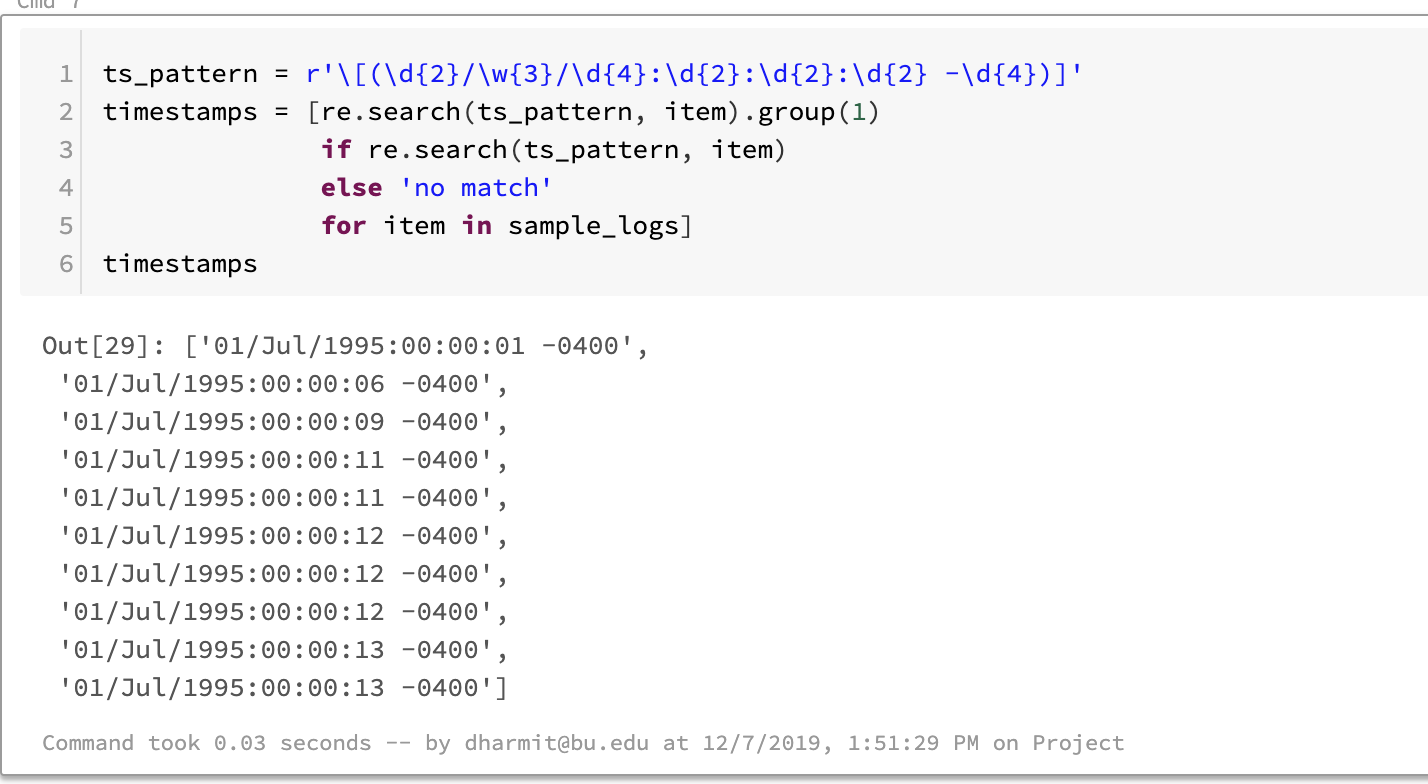


Figure :Extracting timestamps using regex

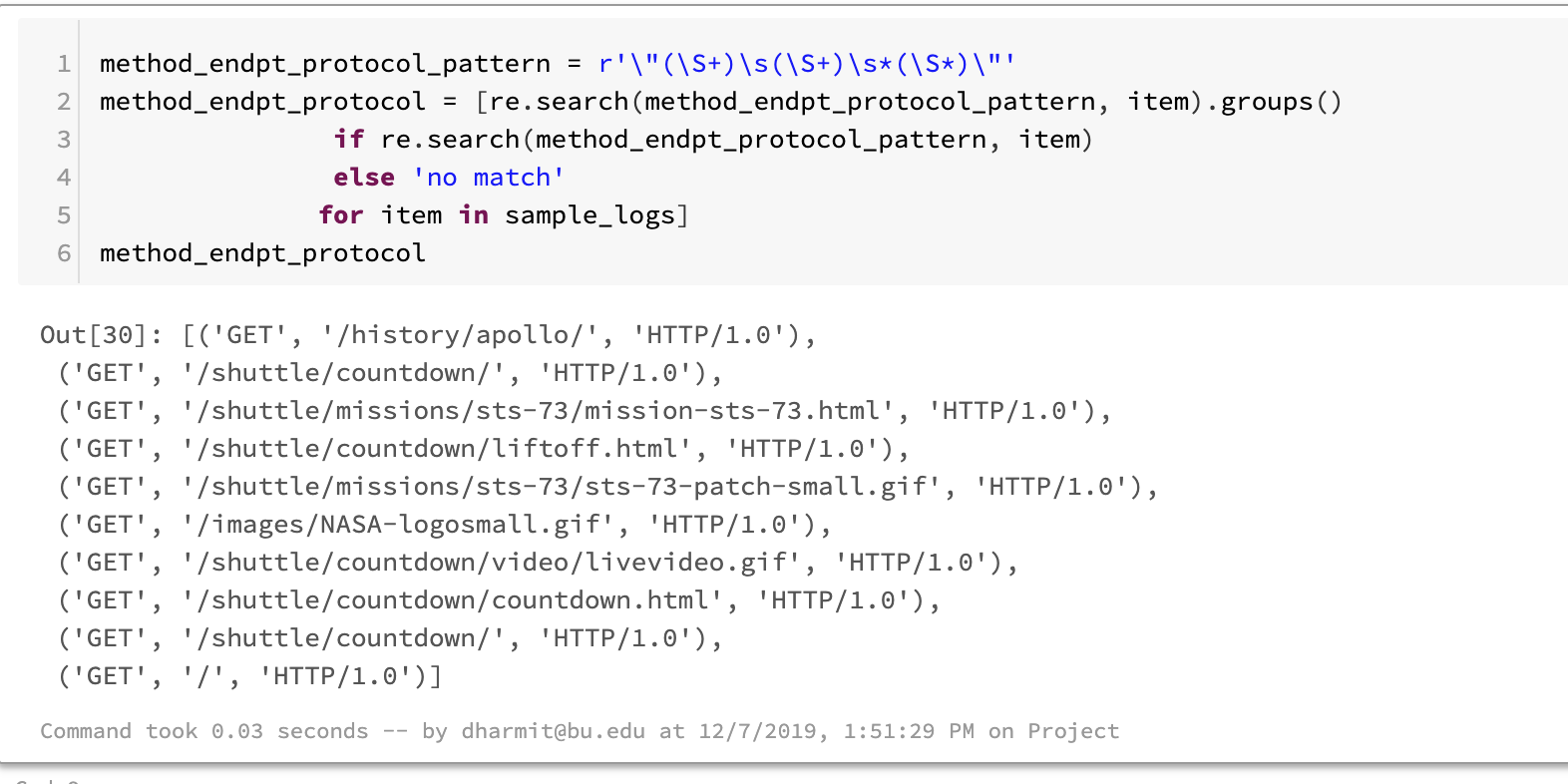


Figure : Extracting methods from logs

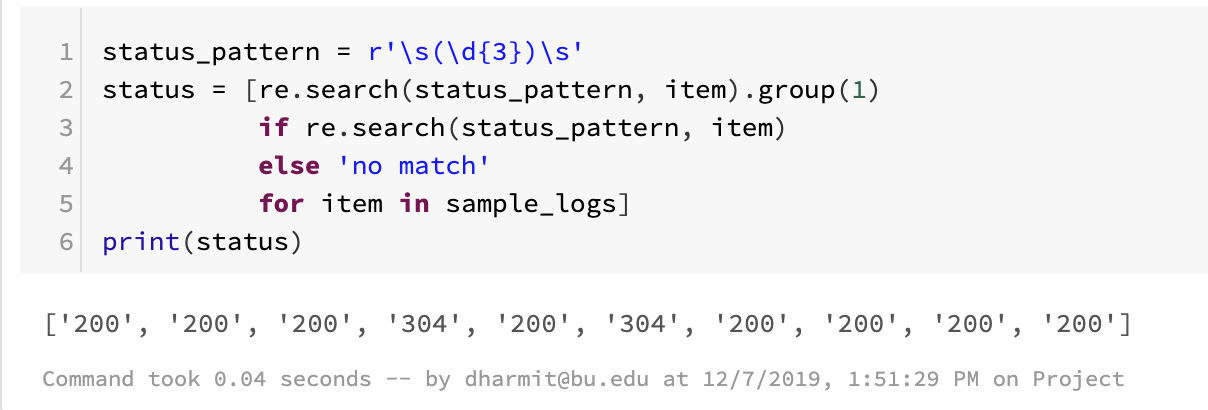


Figure : Extracting status from logs

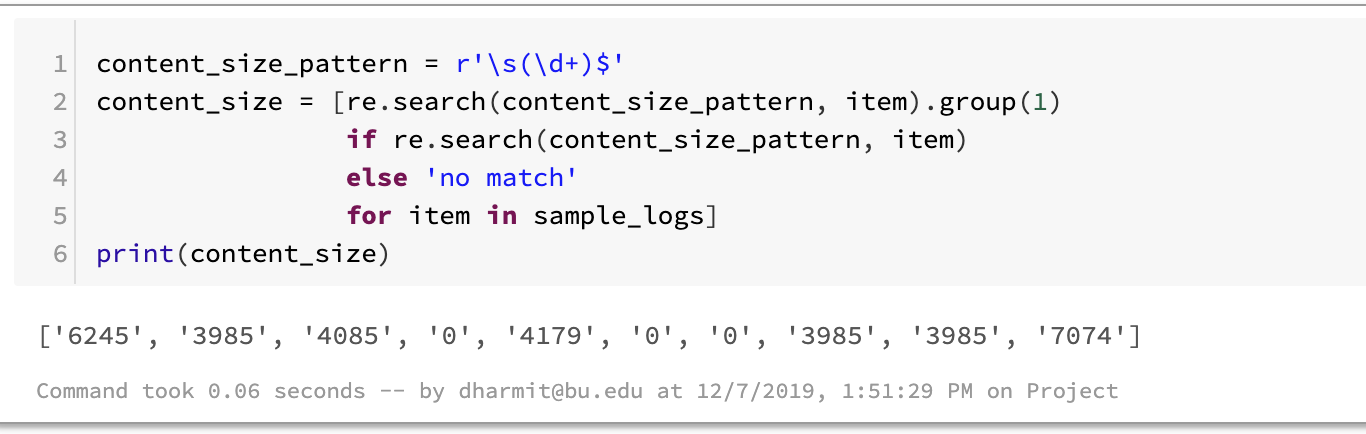


Figure : Extracting content size from logs

We have further put it all together and applied the regular expressions on the original dataset, using the regexp\_extract() function.

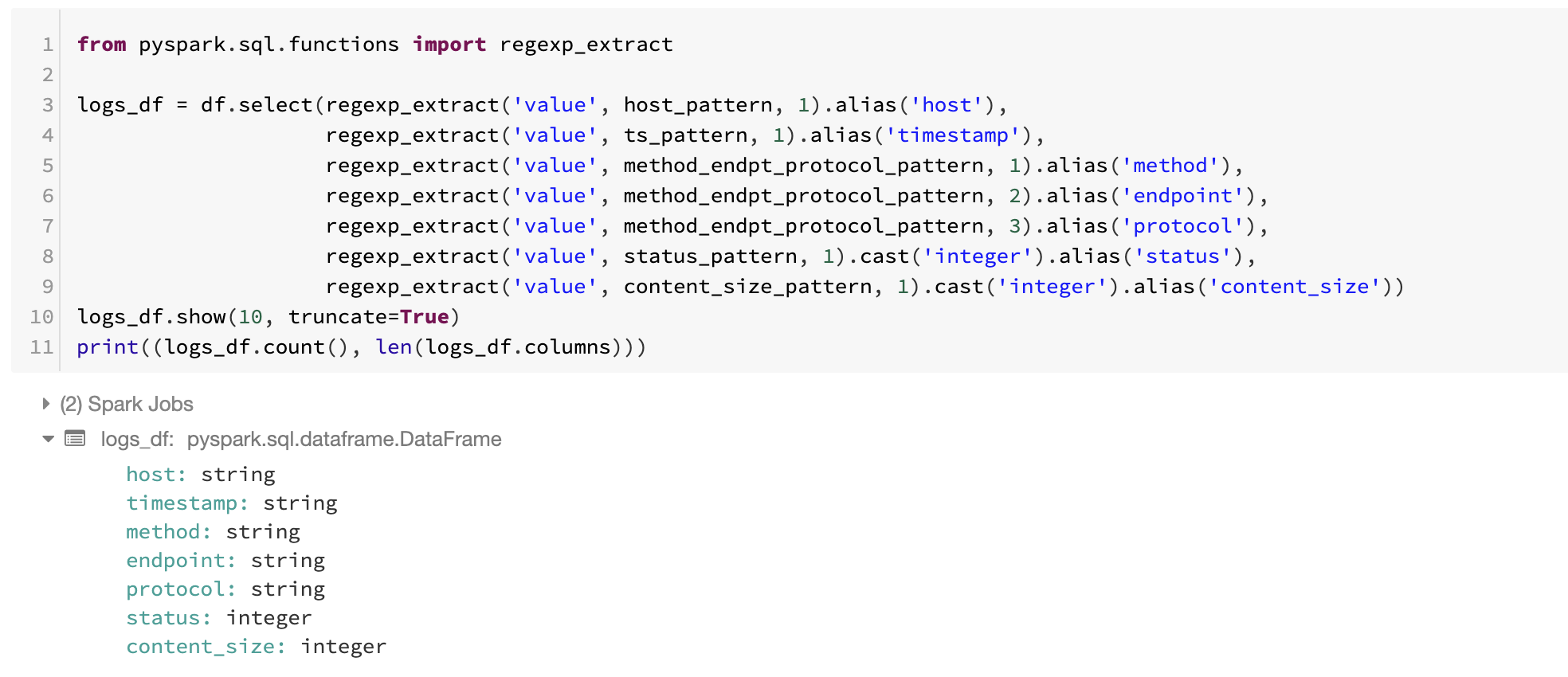


Figure : Extracting all fields from the log data

On performing the above extraction through regular expressions, we get the following dataframe, in a structured data format:

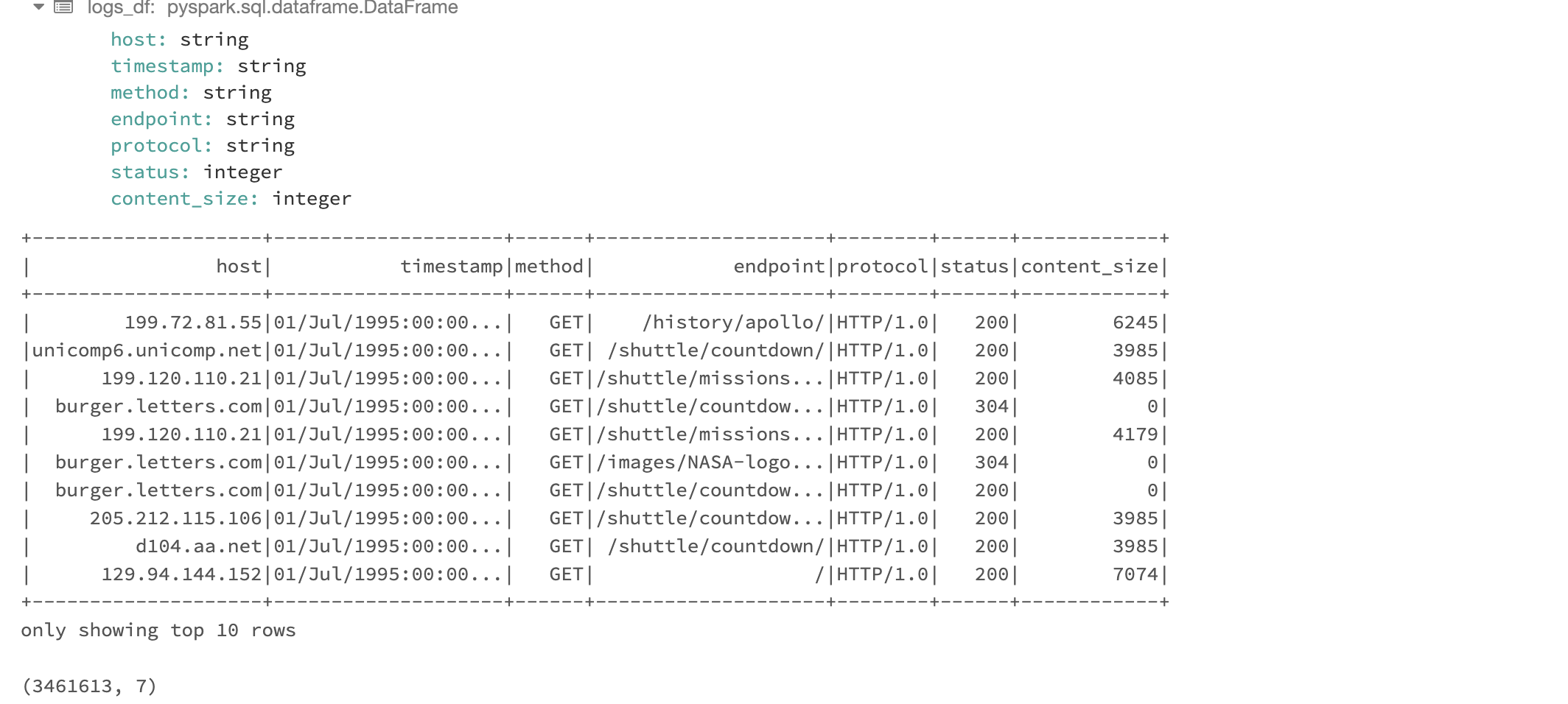


Figure : Structured dataframe with separate columns as fields

As seen in the figure above. We have a dataframe of 3461613 rows and 7 columns.

# **Analysis of logs**

Now that we have a structured dataset, we can perform some analysis on it to derive some insights. We first convert the Spark dataframe into a temporary view.



Figure : Converting dataframe to a temporary view

## Finding Frequent Hosts

We find all the hosts that requested to the server, and the number of times they requested. We sort the rows in descending order, so we get the most frequent hosts at the top.

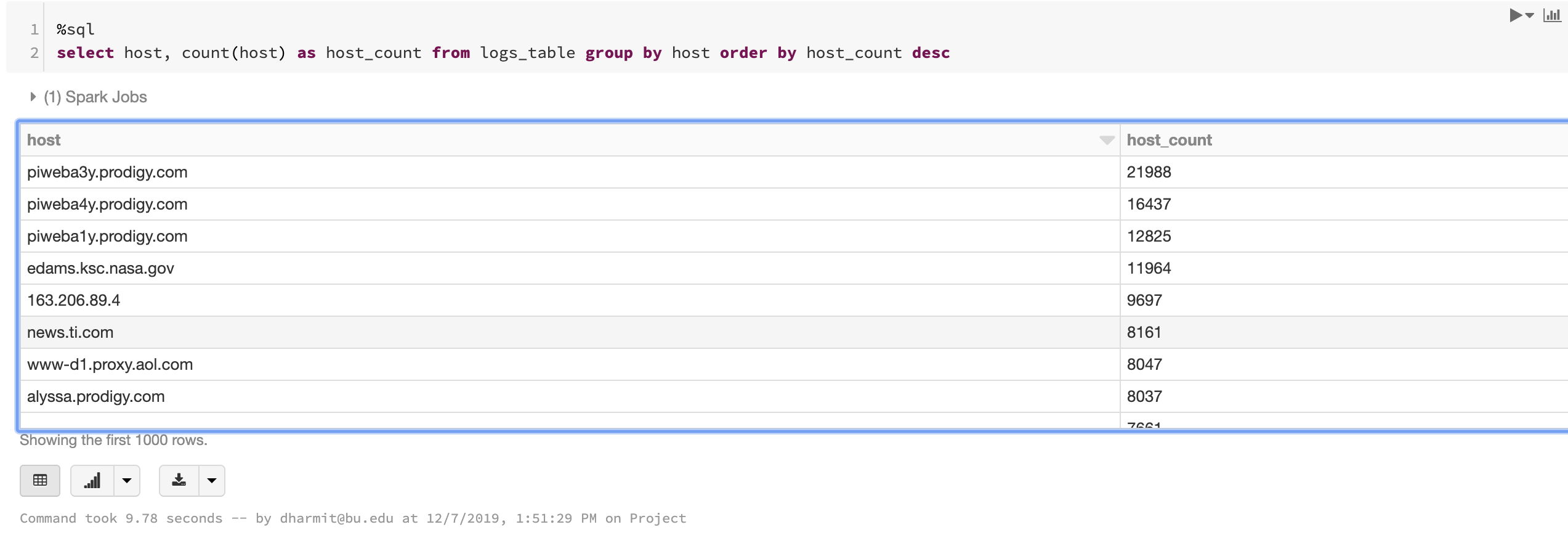


Figure : Finding frequent hosts

In figure 16, we can see that prodigy.com is the most frequent host to the server.

## Finding number of unique hosts

Next, we find the distinct number of hosts that requested to the server.

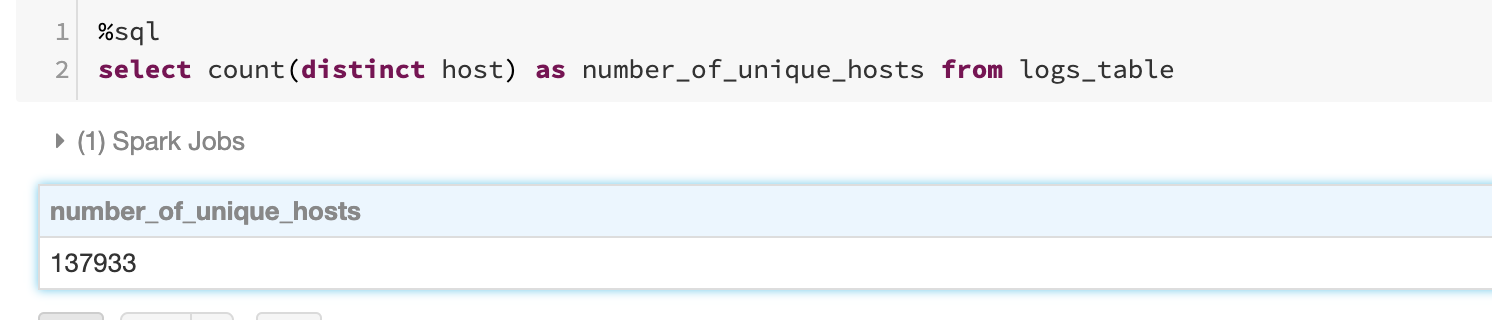


Figure : Finding number of unique hosts

## Finding distinct endpoints and the number of times they were requested

We find the distinct endpoints that were requested for, and the number of times they were requested by the hosts.

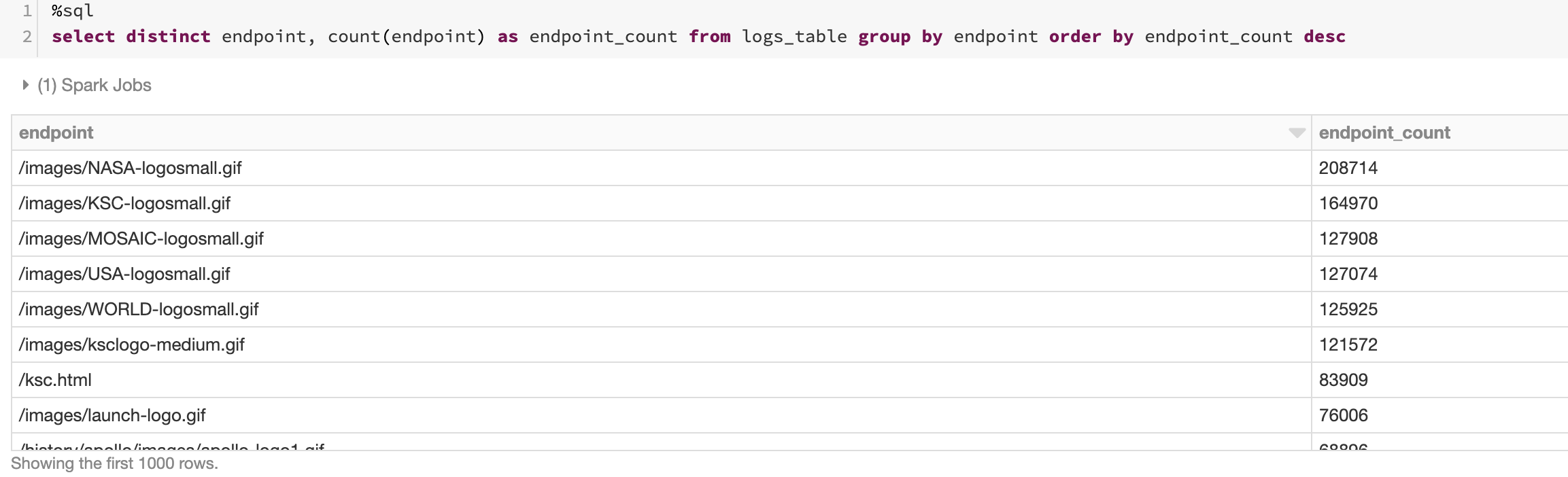


Figure : Distinct endpoints and count of how many times they were requested

## Finding top ten error endpoints

Error endpoints are those that don’t have a status code of 200. All codes except 200 indicate some error.

We display all endpoints and their count, where the status is not 200, order by the count in descending order, and display the top ten rows.



Figure : Top ten error endpoints

## Finding all logs for the “not-found” error

Status “404” indicates a “not-found” error, which we extensively encounter. (For example, “error 404: Page not found). The following query lists all logs with the error, that is with the status code = 404.

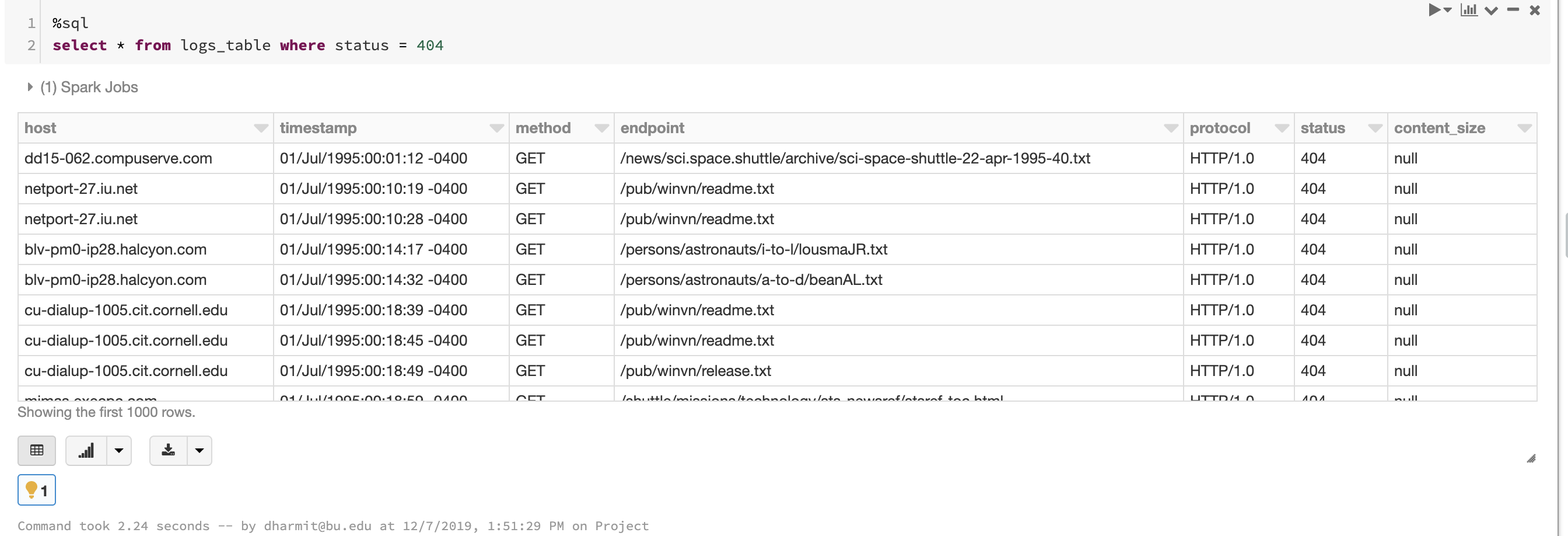


Figure : All logs with status code 404

## Analyzing and visualizing distinct status code values and the number of times they have occurred

We select distinct status code values and sort them in descending order of the number of times they have occurred.



Figure 21: Distinct status codes in the logs

## Content size analysis

We found the minimum, maximum and average content size, which is the number of bytes sent by the server.



Figure 22: Content size statistics

## Finding distinct hosts on a particular day

For doing any analysis for a particular day, we would first need to extract the day from the timestamp. The first step to be able to do that is to convert the timestamp attribute into a datetime object.

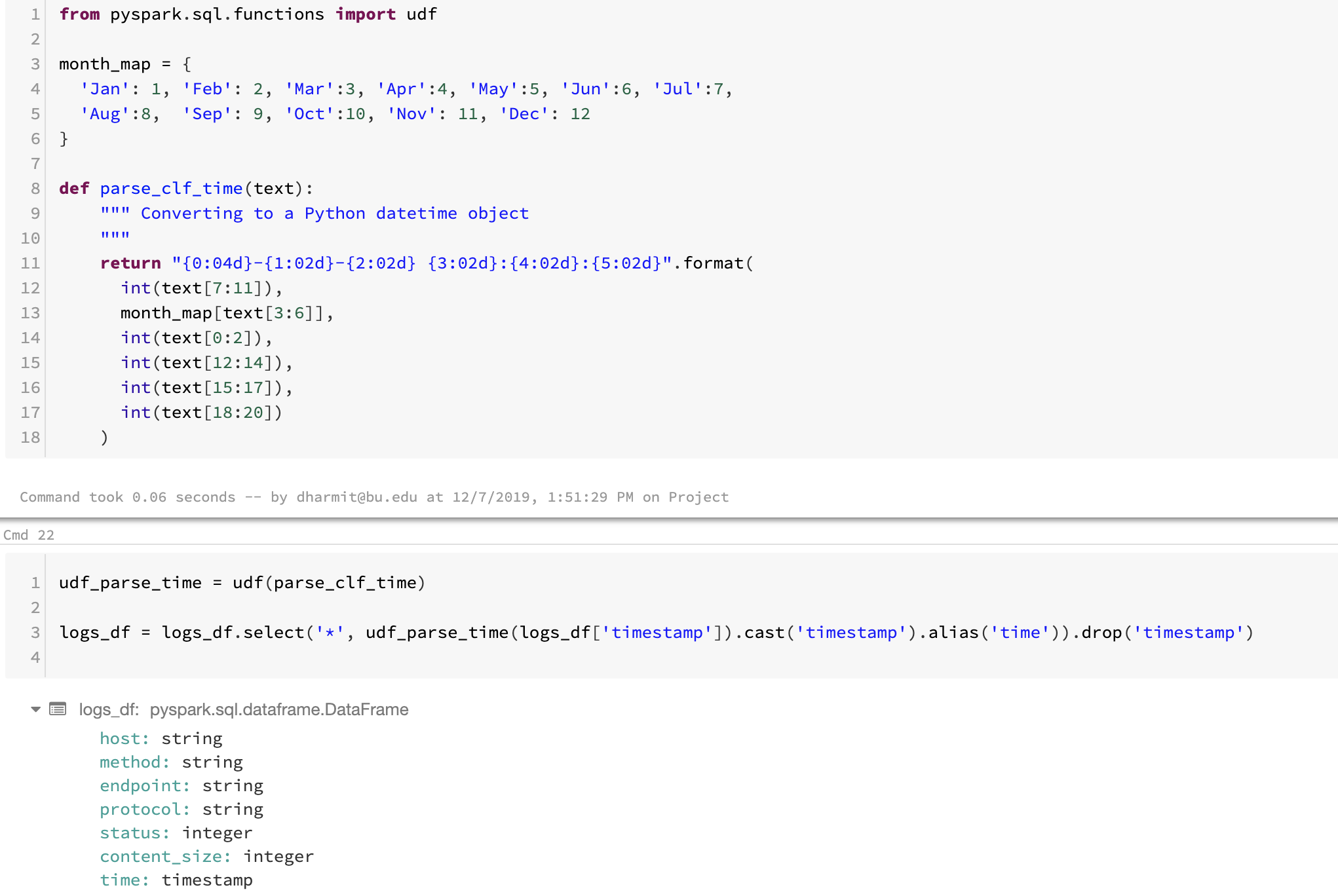


Figure : Converting timestamp string to a python datetime object

In figure 23, we can notice that the time attribute is of type timestamp.

We convert our spark dataframe into a view again:

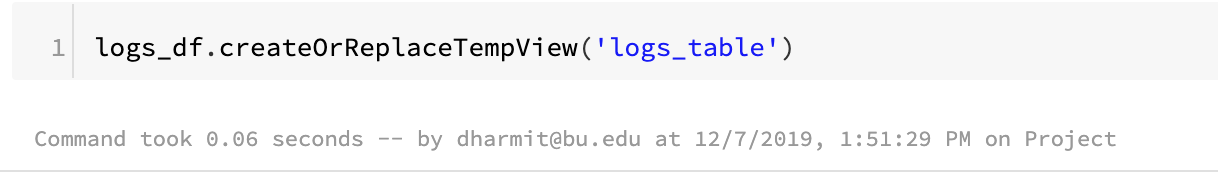


Figure : Converting Spark dataframe to a temporary view

Pyspark.sql has functions called dayofmonth() and month(), with which we can extract the day and month from the date respectively. We make use of these functions to create another dataframe, that contains hosts and the day and month they made requests on.

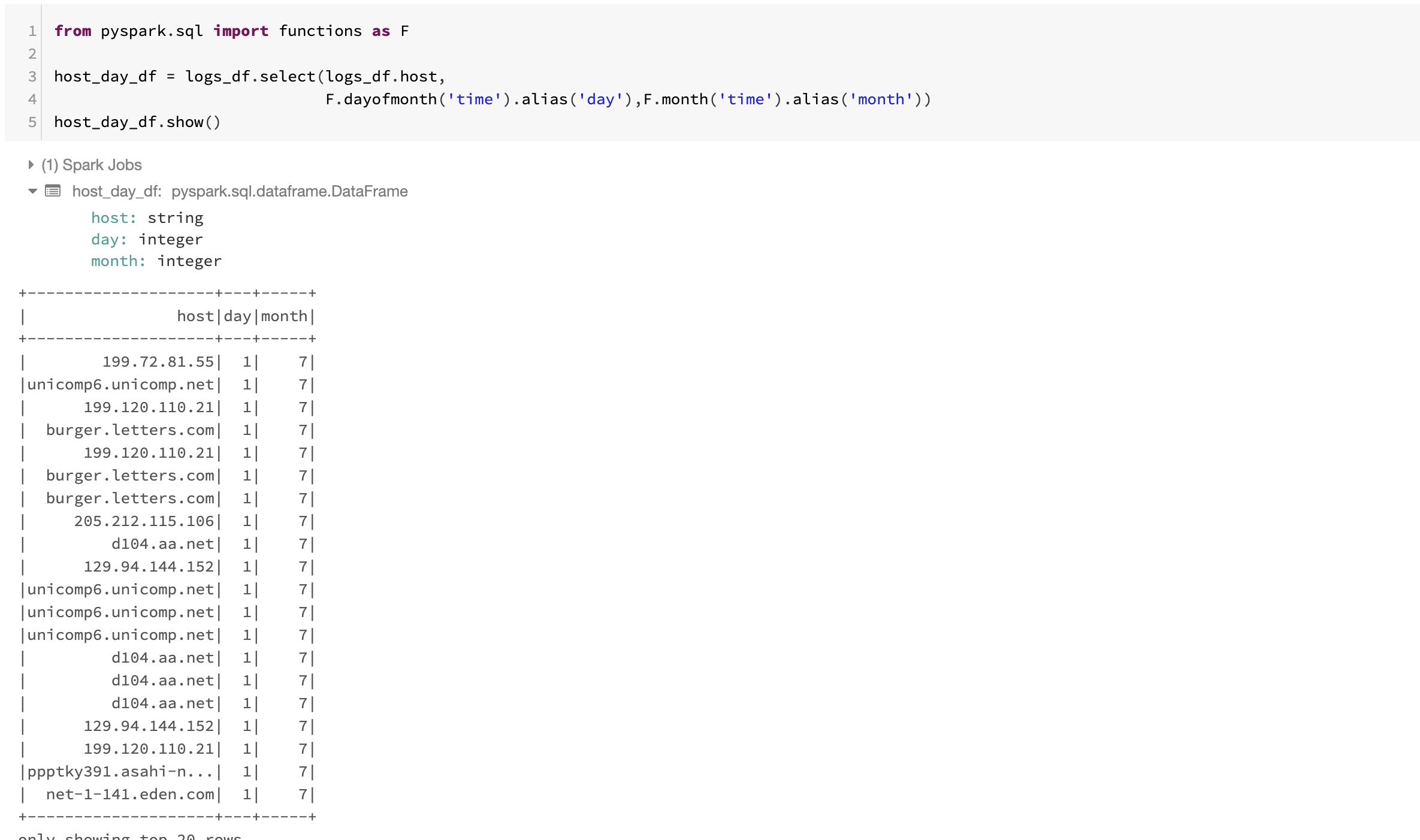


Figure : dataframe with hosts and the day and the month they made requests on

We further used .dropDuplicates() function to create a dataframe of only distinct hosts out of our previous dataframe.

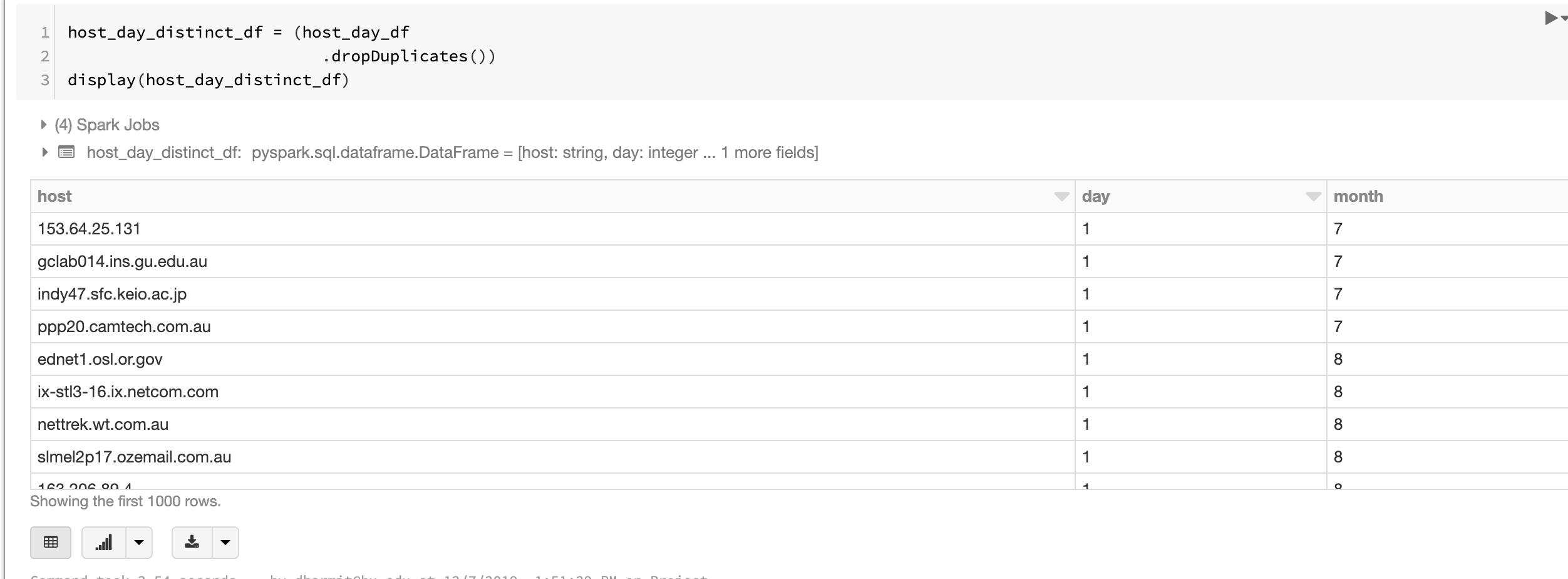


Figure : Creating a dataframe of distinct hosts out of our previous dataframe

We convert the dataframe to a view, so that we can perform SQL queries on it.



Figure : Dataframe of distinct hosts converted to a view

Now, we query the view to get hosts for a particular day and month.

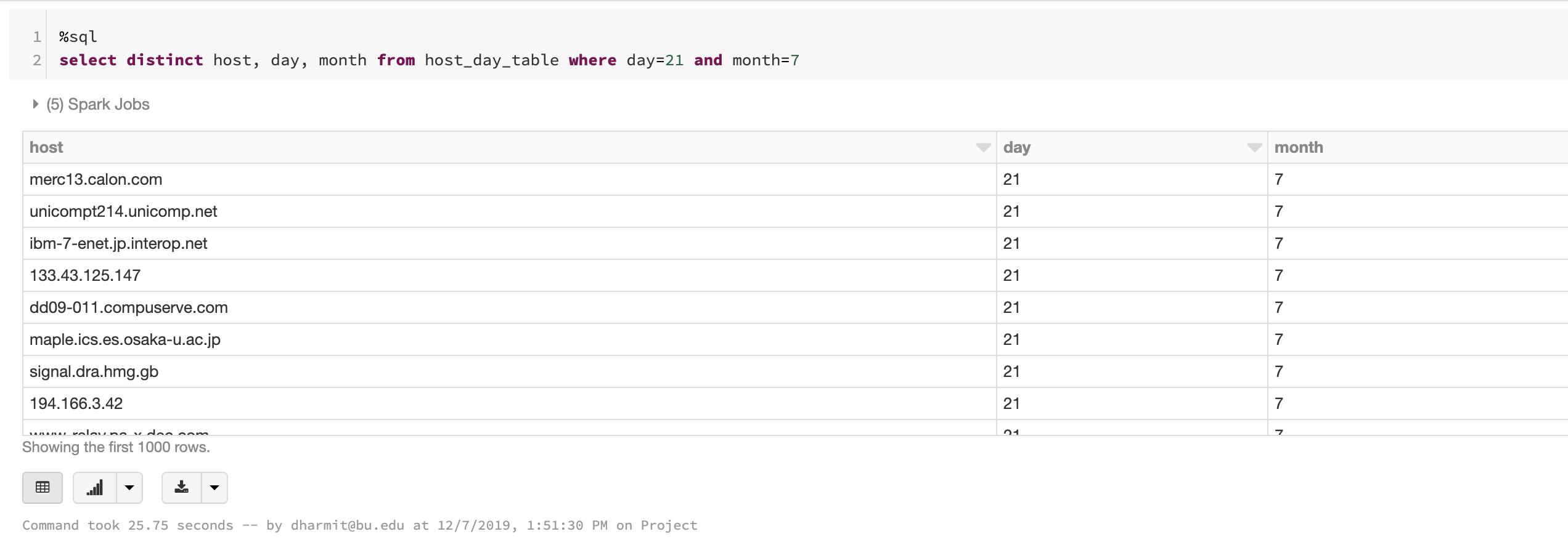


Figure : distinct hosts on 21st July

Thus, we were able to find all unique hosts that requested to the server on 21st July.

# **Data Visualization**

Data visualization involves presenting data in a graphical and pictorial format. Visualizing the data helps gain insights much easily, rather than through dataframes and SQL tables.

In the world of Big Data, data visualization tools and technologies are essential to analyze massive amounts of information and make data-driven decisions. Some of the popular data visualization tools and technologies in use currently are Tableau and Google data studio. A brief introduction to Tableau is provided in this report, and a glance at how it can be used to visualize our logs’ data and derive better insights from the same.

## Tableau

Tableau is a powerful and fastest growing data visualization tool used in the Business Intelligence Industry. It helps in simplifying raw data into the very easily understandable format.

Data analysis is very fast with Tableau and the visualizations created are in the form of dashboards and worksheets. The data that is created using Tableau can be understood by professional at any level in an organization.

Tableau is a powerful and trending data visualization tool, that is widely used in the field of Business Intelligence industry. It works on converting raw data into an easily understandable format. Tableau creates visualizations in the form of dashboards and worksheets. This makes the data clearer and most understandable. (What is Tableau? Uses and Applications. (n.d.))

We use Tableau to visualize our logs’ data. We basically download the temporary view we created in the databricks environment as an excel file. We further loaded the dataset into the Tableau software.

We visualize status code and content size for every host. The hosts have been sorted in descending order of the content sizes.

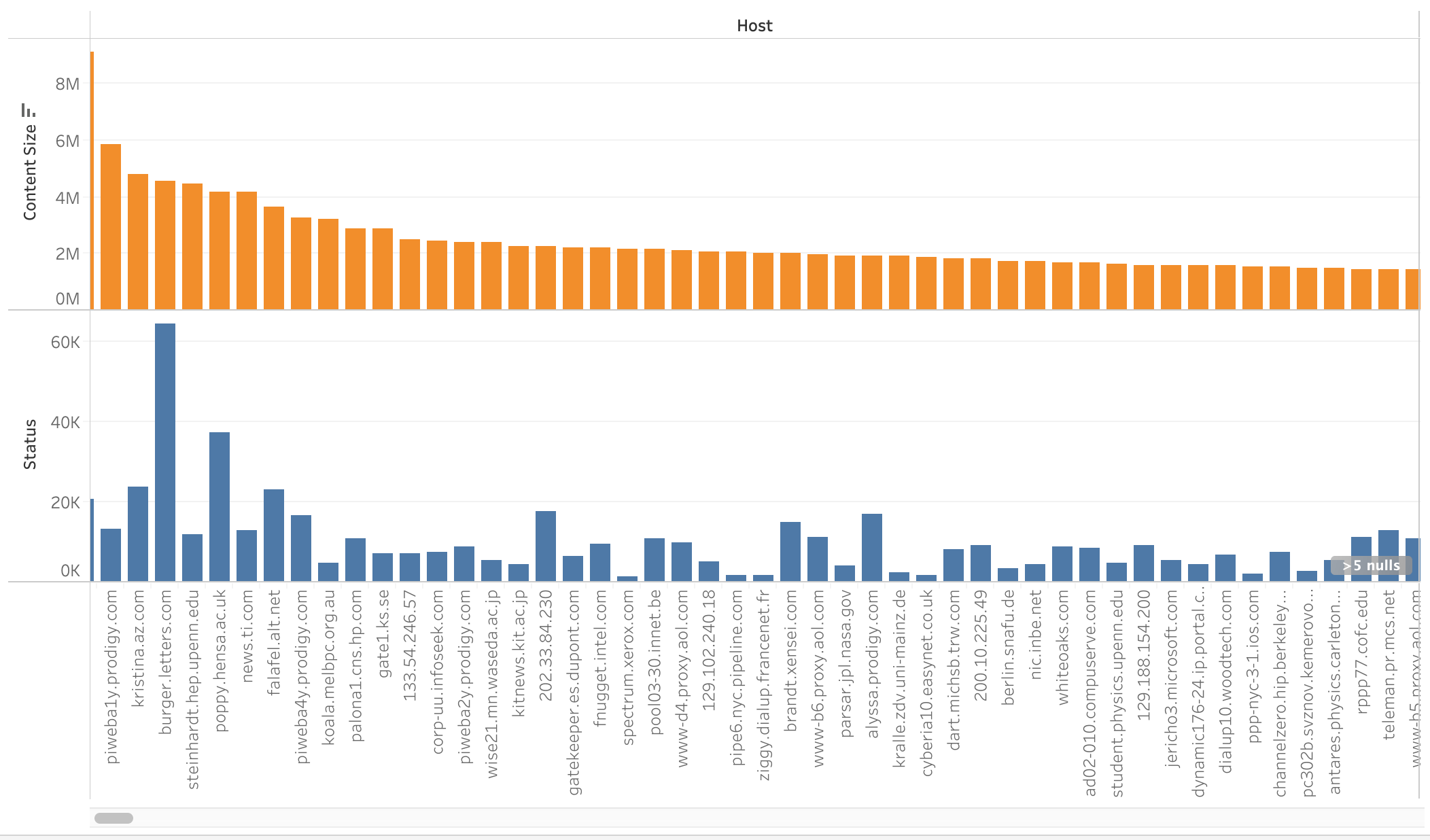


Figure : Visualizing hosts, their status codes and content sizes in Tableau.

# **Learnings from the project**

The goal of this project is to mainly explore functionalities of Spark and the various benefits of using Spark for data analysis. We used NASA’s web server logs as a use case for the same. Following were the lessons learnt from the project:

* Data Wrangling, converting semi-structured data to a structured dataset using regular expressions.
* Using Spark dataframe, python and SQL to perform analysis on our data and derive insights.
* Data visualization using Tableau.

# **Further scope**

Following are the ways this project could be further extended in:

* Deeper analysis into number of hosts, endpoints, etc, per day.
* Extensive visualization to clearly understand all the analysis.

# **References**

Big Data Analytics—What it is and why it matters | SAS. (n.d.). Retrieved December 14, 2019, from <https://www.sas.com/en_us/insights/analytics/big-data-analytics.html>

Apache Spark. (n.d.). Retrieved from <https://en.wikipedia.org/wiki/Apache_Spark>

Apache SparkTM—Unified Analytics Engine for Big Data. (n.d.). Retrieved December 14, 2019, from <https://spark.apache.org/>

Apache SparkTM—What is Spark. (n.d.). Retrieved December 14, 2019, from <https://databricks.com/spark/about>

Spark Data Structure. (n.d.). Retrieved December 14, 2019, from <https://www.roseindia.net/bigdata/pyspark/spark-data-structure.shtml>

Databricks—Unified Data Analytics. (n.d.). Retrieved December 15, 2019, from Databricks website: <https://databricks.com/>

Server Log Analysis with the ELK Stack. (2016, December 6). Retrieved December 14, 2019, from Logz.io website: <https://logz.io/blog/server-log-analysis/>

What is Log Analysis? Use Cases, Best Practices, and More | Digital Guardian. (n.d.). Retrieved December 14, 2019, from <https://digitalguardian.com/blog/what-log-analysis-use-cases-best-practices-and-more>

stevewhims. (n.d.). NCSA Logging—Win32 apps. Retrieved December 14, 2019, from <https://docs.microsoft.com/en-us/windows/win32/http/ncsa-logging>

How to write Regular Expressions? - GeeksforGeeks. (n.d.). Retrieved December 15, 2019, from <https://www.geeksforgeeks.org/write-regular-expressions/>

Sarkar, D. (DJ). (2019, April 12). Scalable Log Analytics with Apache Spark—A comprehensive case-study. Retrieved December 15, 2019, from Medium website: <https://towardsdatascience.com/scalable-log-analytics-with-apache-spark-a-comprehensive-case-study-2be3eb3be977>

Python—PySpark—String matching to create new column. (n.d.). Retrieved December 15, 2019, from Stack Overflow website: <https://stackoverflow.com/questions/46410887/pyspark-string-matching-to-create-new-column>

What is Tableau? Uses and Applications. (n.d.). Retrieved December 15, 2019, from <https://www.guru99.com/what-is-tableau.html>