Big Data Tools and Techniques Course Work

By

Phani Reddy Kukunoori

@00647583

**Introduction/Overview**

The term “big data” has become increasingly popular in recent years, as organizations strive to make better use of the vast amounts of information they are collecting. But what exactly is big data? And what are the best tools and technologies for working with it? In general, big data refers to data sets that are too large or complex to be processed using traditional methods. To be able to work with big data, you need special tools and technologies that are designed for handling large volumes of data. Some of the most popular big data tools and technologies include Hadoop, Spark, and NoSQL databases. Hadoop is a open-source software framework that is designed for distributed storage and processing of big data sets. Spark is a open-source big data processing framework that is designed to be fast and easy to use. NoSQL databases are a type of database that is designed for storing and retrieving large amounts of data. These are just a few of the most popular big data tools and technologies. There are many others to choose from, and the best tool for a particular organization will depend on its specific needs and requirements. Unfortunately, we wont be using any NoSQL fuctionality in our assignment. We will work using Pyspark Dataframe and HiveQL and AWS athena SQL for data processing and achieving desired results.

**Storing Data and Retriveing**

When it comes to data storage, two of the most popular options are DBFS and Amazon AWS S3. Both have their pros and cons, so it’s important to understand the differences before choosing one for your project.

DBFS is a distributed file system that is designed for storing large amounts of data. It is highly scalable and can be used with a variety of database systems. One downside of DBFS is that it can be difficult to manage, so it’s important to have a good understanding of how it works before using it.

Amazon AWS S3 is a cloud storage service that is very popular for storing data. It is easy to use and can be accessed from anywhere. One downside of Amazon AWS S3 is that it can be expensive to use, so it’s important to understand the pricing before using it. In our case we are using aws acedemy learning provided by aws services for storage purposes.

1. To import files to Amazon S3, you will need to first create an Amazon S3 account and then create a "bucket" to store your files.
2. Once you have created your Amazon S3 account and bucket, you can upload your files by logging into your account and selecting the "Upload" button.
3. Name your bucket and choose the region you want to create it in.
4. Choose a region that is closest to your users to minimize latency.
5. Set the bucket to be publicly accessible.
6. On the Set Details page, leave the default values and click Next.
7. On the Set Permissions page, select Grant public read access to this bucket.
8. Click Next again.
9. On the Review page, review your settings and click Create bucket.
10. Now that your bucket is created, you can upload files to it.
11. In the console, select your new bucket, and then click the Upload button.
12. On the Upload page, click Add files.
13. Select the files you want to upload, and then click Open.
14. After your files finish uploading, click Next.
15. On the Set Details page, leave the default values and click Next.
16. On the Set Permissions page, select Grant public read access to this object(s).
17. Click Next again.
18. To import files to DBFS, you will need to first create a DBFS account and then create a "container" to store your files.
19. Once you have created your DBFS account and container, you can upload your files by logging into your account and selecting the "Upload" button.

**Data cleaning and loading**

- Making sure the data is in the correct format - this includes things like ensuring all numerical values are represented as numbers, and all categorical values are represented as strings.

- Paying attention to data types - some big data tools can be very particular about data types, so it's important to make sure everything is formatted correctly.

- Checking for missing values and handle them appropriately - this may mean imputing missing values, or simply dropping rows or columns with missing data.

- Be aware of how the data will be split - if you're working with a large dataset, you'll likely need to split it into smaller chunks in order to process it effectively. Make sure the data is split in a way that makes sense (e.g. by time period or by geographical region) and that each chunk is of a manageable size.

Text

Description automatically generated

Using bdutils commands verifing the dataset in imported successfully

Graphical user interface, text, application

Description automatically generated

handling compressed files with shell

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

Cleaning residual data after successful creation of usable datasets.

Graphical user interface, text, application

Description automatically generated

**Data Analysis**

Using PySpark, Hive and AWS

After successful loading of data we start working on data for our questions and achieve results.

For Question 1 Pyspark analysis mostly counting distinct Id, Question 2 we do the same but finding frequency, Question 3 similar operations but columns is made to array since it has multiple values with “,” as delimiter, Question 4, we use two data set and combine the data using join functions in both Pyspark and Hive, for Question 5 we use both join and relate functions to achieve the desired output, Question 6, we cross check two columns and modify the output as desired.

IN AWS, we do the same using Athena SQL server for all the questions

Necessary measures for analysing data with big data tools

There is no one-size-fits-all answer to this question, as the best approach to data analysis will vary depending on the specific data set and the research goals. However, some tips for effective data analysis include:

- Clearly define the goals of your analysis before starting, so that you can stay focused and avoid getting bogged down in details.

- Use exploratory data analysis to get a feel for the data set and identify potential patterns or relationships.

- Use statistical methods to quantify relationships between variables and test hypotheses.

- Be sure to document your analysis process so that others can understand and replicate your results.

**Visualization**

Pyspark and Hive are already visualizing results by creating table format or text columnar format outputs, for question 6 we additionally use Pandas plot for visualizing the month frequecy in a given year. Using graphical methods to visualise the data and help you to see patterns more clearly sometimes Data visualization is the graphical representation of data. It involves the creation and study of the visual representation of data, in order to see patterns, trends, and correlations that may not be apparent in the raw data.

There are many different ways to visualize data, and the best method depends on the type of data and the question you are trying to answer. Some common visualization techniques include bar charts, line graphs, scatter plots, and heat maps. In our case we do not have much scope for futher analysis, since we are trying to match the desired output and create a replicated results for new dataset.

**Big Data Tools Used:**

* PySpark
* Hive
* Databricks
* Aws Athena SQL Server
* Python

Miscellaneous necessary tools setup for data processing

AWS Athena server configuration

1. Sign into the AWS Management Console and open the Amazon Athena console.

2. In the navigation pane, choose Settings.

3. Under Security Configuration, choose the check box next to Enable Amazon Athena SQL Server Integration.

4. Enter the Amazon S3 bucket name that you want to use for your Athena SQL Server integration in the Bucket Name field.

5. In the Database Name field, enter a name for your Athena SQL Server database.

6. Enter the name of the Amazon S3 folder that you want to use for your Athena SQL Server integration in the Folder Name field.

7. Choose Save. Your Amazon S3 bucket and folder are now ready to be used with Athena SQL Server.

(note: Athena SQL results are included only for dataset clinicaltrail\_2021 for reference)

Task 1

**Question 1:** The number of studies in the dataset. You must ensure that you explicitly check distinct studies.

**Assumption**: Assuming Id column of the dataset is unique and enables us to find count of total number of studies.

**Implementation in PySpark using Dataframes**

Graphical user interface, text, application, email

Description automatically generated

Figure 1

**Implementation in HiveQL**

Graphical user interface, text, application, Teams

Description automatically generated

Figure 2

**Implementation in AWS Athena**

Graphical user interface, text, application, email

Description automatically generated

Aws 1

**Results**

Figure 1, Figure 2 and Aws 1 shows number the total of studies based on the Id collected.

For data set clinicaltrial\_2019, total number of studies are 326348

For data set clinicaltrial\_2020, total number of studies are 356466

For data set clinicaltrial\_2021, total number of studies are 387261

**Task 2**

**Question 2:** You should list all the types (as contained in the Type column) of studies in the dataset along with the frequencies of each type. These should be ordered from most frequent to least frequent.

**Assumption**: listing all the types of studies in the dataset along with the frequencies of each type using “Type” column and be ordered from most frequent to least frequent.

**Implementation in PySpark using Dataframes**

Graphical user interface, text, application, email

Description automatically generated

Figure 3

Graphical user interface, text, application, email

Description automatically generated

Figure 4

**Implementation in Hive**

Graphical user interface, text, table

Description automatically generated

Figure 5

Graphical user interface, text

Description automatically generated

Figure 6

Graphical user interface

Description automatically generated

Figure 7

**Implementation in AWS**

Graphical user interface, text, application

Description automatically generated

**Aws 2**

**Results**

Figure 3, Figure 4, Figure 5, Figure 6, Figure 7 and Aws 2 shows number of the total studies based on the Id collected.

For data set clinicaltrial\_2019, frequency of type of studies are

Type Frequency

Interventional 255945

Observational 64163

Observational [Patient Registry] 6171

Expanded Access 69

For data set clinicaltrial\_2020, frequency of type of studies are

Type Frequency

Interventional 277631

Observational 71434

Observational [Patient Registry] 7332

Expanded Access 69

For data set clinicaltrial\_2021, frequency of type of studies are

Type Frequency

Interventional 301472

Observational 77540

Observational [Patient Registry] 8180

Expanded Access 69

**Task 3**

**Question 3:** The top 5 conditions (from Conditions) with their frequencies.

**Assumptions**: In order to finding top 5 most frequent conditions we use “Conditions” column, where the data is pre-processed to find actual conditions frequency in the dataset.

**Implementation in PySpark using Dataframes**

Graphical user interface, text, application, email

Description automatically generated

Figure 8

Graphical user interface, text, application, email

Description automatically generated

Figure 9

Graphical user interface, text, application, email

Description automatically generated

Figure 10

**Implementation in Hive**

Graphical user interface, text, application, email

Description automatically generated

Figure 11

Graphical user interface, text, application, email

Description automatically generated

Figure 12

**Implementation in AWS**

A screenshot of a computer

Description automatically generated

**Aws 3**

**Results**

Figure 8, Figure 9, Figure 10, Figure 11, Figure 12 and Aws 3 shows number the total of conditions after cleaning of the data.

For data set clinicaltrial\_2019, conditions count are

**new\_conditions count**

Carcinoma 11155

Diabetes Mellitus 9830

Neoplasms 7815

Breast Neoplasms 7486

Syndrome 6842

For data set clinicaltrial\_2020, conditions count are

**new\_conditions count**

Carcinoma 12245

Diabetes Mellitus 10425

Neoplasms 8534

Breast Neoplasms 8009

Syndrome 7419

For data set clinicaltrial\_2021, conditions count are

**new\_conditions count**

Carcinoma 13389

Diabetes Mellitus 11080

Neoplasms 9371

Breast Neoplasms 8640

Syndrome 8032

**Task 4**

**Question 4:** Each condition can be mapped to one or more hierarchy codes. The client wishes to know the 5 most frequent roots (i.e. the sequence of letters and numbers before the first full stop) after this is done.

**Assumption**: By using first three string we find the merge sorted data two datasets, selecting column “term” and “conditions” to find top 5 frequent hierarchical code extraction and frequency.

**Implementation in PySpark using Dataframes**

Graphical user interface, text, application, email

Description automatically generated

Figure 12A

Graphical user interface, text, application, email

Description automatically generated

Figure 13

Graphical user interface, text, application, email

Description automatically generated

Figure 14

**Implementation in Hive**

Graphical user interface, text, application, email, Teams

Description automatically generated

Figure 15

Graphical user interface, text, application

Description automatically generated

Figure 16

Graphical user interface, text, application, email

Description automatically generated

Figure 17

**Implementation in AWS**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Aws 4**

**Results**

Figure 12a, Figure 13, Figure 14, Figure 15, Figure 16 and Aws 4 shows count the top 5 frequency of codes after cleaning of the data.

For data set clinicaltrial\_2019, top 5 code frequency count are

**Root Count**

C04 123221

C23 113997

C14 82043

C10 76665

C01 73477

For data set clinicaltrial\_2020, top 5 code frequency count are

**Root Count**

C04 133091

C23 124589

C01 94293

C14 88065

C10 83894

For data set clinicaltrial\_2021, top 5 code frequency count are

**Root Count**

C04 143994

C23 136079

C01 106674

C14 94523

C10 92310

**Task 5**

**Question 5:** Find the 10 most common sponsors that are not pharmaceutical companies, along with the number of clinical trials they have sponsored.

Assumption: For implementation, we consider the Parent Company column contains all possible pharmaceutical companies, using this we exclude companies and get the sponsor data without pharmaceutical companies.

**Implementation in PySpark using Dataframes**

Graphical user interface

Description automatically generated

Figure 18

Graphical user interface, text, application

Description automatically generated

Figure 19

Graphical user interface, text, application

Description automatically generated

Figure 20

**Implementation in Hive**

Graphical user interface

Description automatically generated with medium confidence

Figure 21

A picture containing graphical user interface

Description automatically generated

Figure 22

Table

Description automatically generated

Figure 23

**Implementation in AWS**

Graphical user interface, text, application, email

Description automatically generated

Aws 5

**Results**

Figure 18, Figure 19, Figure 20, Figure 21, Figure 22 and Aws 5 shows count the top 10 frequency of sponsors after cleaning of the data with no pharmaceutical companies

For data set clinicaltrial\_2019, top 10 sponsor frequency count are

**Sponsor count**

National Cancer Institute (NCI) 3003

M.D. Anderson Cancer Center 2097

Mayo Clinic 1930

Assistance Publique - Hôpitaux de Paris 1764

Massachusetts General Hospital 1715

Hoffmann-La Roche 1694

National Taiwan University Hospital 1608

For data set clinicaltrial\_2021, top 10 sponsor frequency count are

**Sponsor count**

National Cancer Institute (NCI) 3100

M.D. Anderson Cancer Center 2238

Mayo Clinic 2097

Assistance Publique - Hôpitaux de Paris 2043

Massachusetts General Hospital 1823

Assiut University 1806

Hoffmann-La Roche 1761

For data set clinicaltrial\_2022, top 10 sponsor frequency count are

**Sponsor count**

National Cancer Institute (NCI) 3218

M.D. Anderson Cancer Center 2414

Assistance Publique - Hôpitaux de Paris 2369

Mayo Clinic 2300

Assiut University 2154

Massachusetts General Hospital 1971

Cairo University 1928

**Task 6**

**Question 6**: Plot number of completed studies each month in a given year – for the submission dataset, the year is 2021. You need to include your visualization as well as a table of all the values you have plotted for each month.

**Assumption:** Using Submission date and Status columns in dataset we find the total frequency of month in a give year where status is completed.

**Implementation in PySpark using Dataframes**

Graphical user interface, text, application, email

Description automatically generated

Figure 27

Graphical user interface, text, application, email

Description automatically generated

Figure 28

Graphical user interface, text, application, email

Description automatically generated

Figure 29

Graphical user interface, chart, bar chart

Description automatically generated

Figure 30

**Implementation in Hive**

Graphical user interface, text, application

Description automatically generated

Figure 31

Graphical user interface, text, application, email, Teams

Description automatically generated

Figure 32

Graphical user interface, text, application

Description automatically generated

Figure 33

**Implementation in AWS**

Graphical user interface, application, Teams

Description automatically generated

AWS 6

**Results:** Figure 24, Figure 25, Figure 26, Figure 27, Figure 28, Figure 29, Figure 31, Figure 32, Figure 33 and Aws 6 shows count of the frequency of months in a given year where status is completed

For data set clinicaltrial\_2019, frequency count of month in 2019 is

**Completed count**

Dec 2690

Jun 1647

Jul 1547

Mar 1470

Sep 1421

Aug 1406

Apr 1368

For data set clinicaltrial\_2020, frequency count of month in 2020 is

**Comp count**

Dec 2084

Mar 1740

Jan 1544

Jun 1424

Feb 1286

Jul 1237

May 1176

For data set clinicaltrial\_2021, frequency count of month in year 2021 is

**Comp count**

Mar 1227

Jan 1131

Jun 1094

May 984

Apr 967

Feb 934

Jul 819

And Figure 30 visualizes the results using python pandas plot function.

**References:**

* https://docs.aws.amazon.com/
* https://spark.apache.org/documentation.html
* https://cwiki.apache.org/confluence/display/Hive/Home
* https://spark.apache.org/docs/2.4.0/api/python/pySpark.sql.html
* https://databricks.com/blog/2016/02/17/introducing-databricks-community-edition-apache-spark-for-all.html
* https://matplotlib.org/users/intro.html