Big Data Class Project - Part 2

Class: BUAN 6346.502 - MIS 6346.502

Semester: Spring 2020

Total points: 125

The class project report is individual work. If you choose to consult anyone, please list names of every person you have discussed project with. The Due date for the project will be published in the eLearning.

===

Important - to minimize costs associated with AWS services:

1. Always terminate EMR cluster when not in use for extended period of time
2. Always provision EMR boxes using **Spot** pricing. **Don't** use On-demand pricing
3. Save Spark notebooks by downloading to local computer
4. Save intermediate work by persisting Spark DFs to S3

===

Starter Materials

Two starter jupyter notebooks are provided:

- Load sample files and transform date from integer to date format
- LDA modeling notebook

The notebooks are written to run on VM. When running on EMR you will need to remove lines of code dealing with creating Spark session. The notebooks assume that you placed sample files in the `/home/jupyter/data/amazon/amazon_review_sample/` folder on the VM.

To unzip files on the VM run `unzip file_name` command.

Project Description

The project is an essential part of this class. It will allow you to practice and demonstrate your Big Data (BD) and Analytics skills. It can also be a valuable addition to your projects portfolio that you can demonstrate to prospective employers.

The project is divided to two parts. First part is described in this document.

In part 2 you will be able to practice concepts you have learned in the first and second part of the class:

1. HDFS
2. AWS Big data technologies: EMR
3. Spark

The emphasize of the project is on Spark. All analysis should be performed by utilizing Spark DataFrame APIs. It is expected that you will utilize Python Spark API for the project. If you decide to use other API, please contact professor to get permission.

If you decide to use SQL instead of DataFrame API, there will be reduction of **25 points.**

Project Requirements

Part 2 - Amazon Reviews Analysis

For the project, you will perform analysis of the provided dataset using Big Data techniques.

Project Environment: AWS EMR - AWS Educate Class account.

Project tools:

1. AWS S3
2. AWS EMR - HDFS, Spark, Hive (optional)
3. AWS Athena - exploratory analysis only, don't submit findings obtained via Athena.

Data Set

Amazon reviews dataset: https://registry.opendata.aws/amazon-reviews/

Documentation: https://s3.amazonaws.com/amazon-reviews-pds/readme.html

We will using parquet version of the dataset in order to reduce storage requirements on AWS EMR and to speed up processing and analytics.

The dataset is partitioned by product category, here is the subset of the partitions. Note the shell command you can execute from AWS EMR master machine shell:

```
aws s3 ls s3://amazon-reviews-pds/parquet/
PRE product_category=Apparel/
PRE product_category=Automotive/
PRE product_category=Baby/
PRE product_category=Beauty/
PRE product_category=Books/
PRE product_category=Camera/
PRE product_category=Digital_Ebook_Purchase/
.............................................
.............................................
PRE product_category=Video_Games/
PRE product_category=Watches/
PRE product_category=Wireless/
```

Each category folder folder contains parquet files with the reviews for the category. Example:

```
aws s3 ls s3://amazon-reviews-pds/parquet/product_category=Automotive/
2018-04-09 06:35:42   84703245 part-00000-495c48e6-96d6-4650-aa65-
3c36a3516ddd.c000.snappy.parquet
..................................................................
2018-04-09 06:35:46   85203786 part-00009-495c48e6-96d6-4650-aa65-
3c36a3516ddd.c000.snappy.parquet
```

Write shell script so that you can run every time new EMR cluster is being provisioned. The script will create folders in HDFS and copy files from S3 to EMR HDFS.

Requirements

Use the following product categories:

```
- Digital_Ebook_Purchase
- Books
- Wireless
- PC
- Mobile_Apps
- Video_DVD
- Digital_Video_Download
```

Start your analysis from year 2005. Exclude multiple reviews by the same users for the same product. In the case the same user has reviewed particular product more than once, exclude all reviews following the first review. First review should remain as part of the analysis.

Required. To improve performance of your Spark transformations follow below steps to create DataFrame with no duplicates. Spark is sensitive to memory allocation, and if you don't follow those steps, most likely your application will fail on memory problem.

Reminder: as per project requirement, we need to keep one review per product_category, customer_id and product_id - best way to do it is via Spark Window function: `row_number()`.

Here are high level steps:

1. Load DF from Files – **Don’t** persist/cache
2. Create new DF with relevant columns only and for relevant dates - **Don’t** persist/cache
3. Create new DF and assign `row_number()` – make sure to use **product_category**. **Don’t** persist/cache
4. Create new DF using one in #3 by filtering only records with `rownum=1`. **Persist**

To “activate” `persist()` run `df.count()`.

It less that 5 minutes to create DataFrame with no duplicates, while running on three m4.xlarge Core machines.

You may save the resulting DataFrame (from step #4) into Hive. You would need to copy the table/files to S3 to use for the next session or rerun the steps.

Using Spark DF APIs, answer the following questions:

1. Explore the dataset and provide analysis by product-category and year:
    1. Number of reviews
    2. Number of users
    3. Average and Median review stars
    4. Percentiles of length of the review. Use the following percentiles: [0.1, 0.25, 0.5, 0.75, 0.9, 0.95]
    5. Percentiles for number of reviews per product. For example, 10% of books got 5 or less reviews. Use the following percentiles: [0.1, 0.25, 0.5, 0.75, 0.9, 0.95]

    6. Identify week number (each year has 52 weeks) for each year and product category with most positive reviews (4 and 5 star).

2. Provide detailed analysis of "Digital eBook Purchase" versus Books.
    1. Using Spark Pivot functionality, produce DataFrame with following columns:
        1. Year
        2. Month
        3. Total number of reviews for "Digital eBook Purchase" category
        4. Total number of reviews for "Books" category
        5. Average stars for reviews for "Digital eBook Purchase" category
        6. Average stars for reviews for "Books" category
    2. Produce two graphs to demonstrate aggregations from #1:
        1. Number of reviews
        2. Average stars
    3. Identify similar products (books) in both categories. Use "product_title" to match products. To account for potential differences in naming of products, compare titles after stripping spaces and converting to lower case.
        1. Is there a difference in average rating for the similar books in digital and printed form?
        2. To answer #1, you may calculate number of items with high stars in digital form versus printed form, and vise versa. Alternatively, you can make the conclusion by using appropriate pairwise statistic.
    4. Using provided LDA starter notebook, perform LDA topic modeling for the reviews in Digital_Ebook_Purchase and Books categories. **Consider reviews for the January of 2015 only.**
        1. Perform LDA separately for reviews with 1/2 stars and reviews with 4/5 stars.
        2. Add stop words to the standard list as needed. In the example notebook, you can see some words like `34, br, p` appear in the topics.
        3. Identify 5 top topics for each case (1/2 versus 4/5)
        4. Does topic modeling provides good approximation to number of stars given in the review?

Technical details on working with the Dataset

AWS Athena

You can use AWS Athena to perform some of the analysis. This is to simplify access and avoid charges associated with EMR. However, final report should be produced by utilizing Spark running on EMR and pulling data from HDFS.

Use instructions from Part-1 to get you started with Athena.

AWS EMR

The final project deliverables should be produced using AWS EMR Spark. Here are the steps to get you going:

1. Provision EMR cluster. You might want to provision 2-3 Slave servers to speed up processing. Make sure to select **Spot** pricing when provision EMR.
2. Copy Amazon reviews to EMRs HDFS, you should run similar command for the relevant product categories:

```
hdfs dfs -mkdir -p /hive/amazon-reviews-pds/parquet/product_category=Apparel/

s3-dist-cp --src=s3://amazon-reviews-pds/parquet/product_category=Apparel/ \
--dest=hdfs:///hive/amazon-reviews-pds/parquet/product_category=Apparel/
```

3. Create Spark DataFrame by loading files from `hdfs:///hive/amazon-reviews-pds/parquet/`
4. When you just getting started, it is suggested to limit number of records (for example 10k per category/year) in the Spark DataFrame. Easiest way to do it is by using `limit()` function to create separate DF per product category and year. Concatenate the smaller DFs into a single DF for analysis. This is not needed when loading full data-set.
5. To speed up development and minimize EMR time, sample dataset is provided and available on eLearning. Starter Jupyter notebook is provided as well.

Project report

Deliverables:

1. You will create a well formatted project report that would include the following sections. You can use any editor, and your report will be delivered in PDF file format.
2. Jupyter notebook in two formats: *.ipynb and *.html. The notebook will include all data manipulations and Spark commands. Include descriptions in the markdown cells to document the flow.

Introduction and problem description

Formulate your own problem description and describe it in detail.

Formulate a list of tasks that you are about to perform to solve this problem.

Describe in details your approach and techniques to solve the tasks.

Visualization

You can enhance your findings by utilizing graphs and plots. You can use any tool and libraries to do so.

Conclusion

Provide conclusions to any metric and or finding. You should provide your own assessment and not just rely on the report user to make his own conclusions.

References

Provide relevant references. These may include a description of data, previous analysis of the data available on the internet, code references etc.

References are very important, any code that you reuse should be recognized and mentioned.