CSE427S

HOMEWORK 8

M. Neumann

Due: THU 31 OCT 2019 4PM

Getting Started

Update your SVN repository.

When needed, you will find additional materials for *homework x* in the folder `hwx`. So, for the current assignment the folder is `hw8`.

SUBMISSION INSTRUCTIONS

WRITTEN:

- all written work needs to be submitted electronically in *pdf format*[1] via GRADESCOPE
- provide the following information on *every* page of your `pdf` file:
  - name
  - student ID
- start every problem on a *new page*
- **FOR GROUPS**: make a **group submission** on GRADESCOPE and provide names and student IDs for **all group members** on every page of your `pdf` file.

CODE:

- No code submission required for this homework.
- If applicable, submit (command line or code) statements in your written submission as instructed below.

Good News

Dualcore Inc. is a leading electronics retailer with more than 1,000 brick-and-mortar stores and a thriving e-commerce web site. Dualcore has hired you to help find value in their customer ratings and feedback data. These data great sources of information for both customers and retailers like Dualcore. However, customer comments are typically free-form text and must be handled accordingly. Fortunately, Hive provides extensive support for text processing.

[1] Please, **type your solutions** or use **clear hand-writing**. If we cannot read your answer, we cannot give you credit nor will we be able to meet any regrade requests concerning your writing.

Problem 1: Create a Hive-Managed Table (15%)

Create a table called `products` as a Hive managed table. Table specifications:

Index	Field	Description	Example
0	**prod_id**	Product ID	1273641
1	brand	Brand name	Foocorp
2	name	Name of product	4-port USB Hub
3	price	Retail sales price, in cents	1999
4	cost	Wholesale cost, in cents	1463
5	shipping_wt	Shipping weight (in pounds)	1

total number of products: 1,114 records

IMPLEMENTATION SPECIFICATIONS:

- Import the `products` folder from `training_materials/analyst/data/pig_multidata` to `/dualcore/products` in HDFS.
- Using the Hive shell create a table named `products` in the `dualcore` database for storing tab-delimited records using the above structure. HINT: you created the `dualcore` database in Lab 7.
- Load the data into this newly created table.

(a) Provide all Hive statements you used to create the table in the dualcore database.

(b) Provide the table description (output of the Hive command describing the schema of the table).

(c) Provide the Hive statement to load the actual data into the products table and verify that the table has the correct amount of records. Provide the Hive statement to get the record count.

Problem 2: Analyze Numeric Product Ratings (10%)

For this problem we will be working with the `ratings` table you created in Lab 7.

(a) We want to find the product that customers liked most, but must guard against being misled by products that have few ratings assigned. Develop a query to find the product with the highest average among products with at least 50 ratings. Provide

- the Hive statement
- the product ID and average rating for this product
- the number of MAPREDUCE jobs your query ran

(b) Modify the Hive query to find the product that customers disliked most. Again, only consider products with at least 50 ratings. Provide the product ID and average rating for this product.

(c) Do you think limiting the minimum number of rating to 50 in your analysis is a good or bad idea. Justify your decision by discussing pros and cons.

Problem 3: Analyze Rating Comments and Product Information (25%)

We observed earlier that customers are very dissatisfied with one of the products that Dualcore sells. Although numeric ratings can help identify which product that is, they don't tell Dualcore why customers don't like the product. We could simply read through all the comments associated with that product to learn this information, but that approach doesn't scale. Next, you will use Hive's text processing support to analyze the comments.

(a) Find all messages for the product with ID 1274673, transfer them into all lowercase and find the five most common bi-grams. HINT: check the HIVE TEXT PROCESSING REFERENCE for an example use of the Hive NGRAMS function: `https://classes.cec.wustl.edu/cse427/slides/13_HiveTextReference.pdf`. Provide

- the Hive statement
- the 5 most common bi-grams

(b) Most of these words are too common to provide much insight, though the word "expensive" does stand out in the list. Modify the previous query to find the five most common tri-grams (three- word combinations). Provide

- the Hive statement
- the 5 most common tri-grams

(c) Among the patterns you should see the phrase "ten times more." This might be related to the complaints that the product is too expensive. Now that you've identified a specific phrase, look at three different customer comments that contain it. Provide

- the Hive statement
- the 3 comments

(d) The comments retrieved in the previous part should reveal that customers complain about a version of a product in a specific color. What is the color? Now, write and execute a query that will find all distinct comments containing a mention of that color that are associated with product ID 1274673. From those comments you should see the product name and brand. Provide

- the color
- the Hive statement to find the name and brand
- the product name and brand of the disliked product

(e) Run a query on the `products` table to identify similar products (products with similar names; possibly different colors) from the same brand along with their costs and prices. Provide

- the Hive statement
- a list of the similar products with their costs and prices

HINT: Explore the hive command `LIKE`.

Congratulations! Based on the cost and price columns, it appears that you just helped Dualcore uncover a pricing error by doing text processing on the product ratings. Congratulations!

Problem 4: SPARK Job Execution (10%)

(a) Describe what *pipelining* means in the context of a SPARK job execution. What is its benefit?

(b) Give an example of two operations that can be pipelined together.

(c) Give an example of two operations that cannot be pipelined together.

More Good News

Another hypothetical company needs your help! Loudacre Mobile is a (*fictional*) fast-growing wireless carrier that provides mobile service to customers throughout western USA. Loudacre just hired you as a data analyst. Congrats and let's get to work! Your first task is to migrate their existing infrastructure to Hadoop and perform some ETL (Extract-Transform-Load) processing. The size and velocity of their data has exceeded their ability to process and analyze their data without a cloud solution. Loudacre has several data sources:

- MySQL database – customer account data (name, address, phone numbers, devices)
- Apache web server logs from Customer Service site, such as HTML files and Knowledge base articles
- XML files such as device activation records
- Real-time device status logs
- Base stations (cell tower locations)

Problem 5: ETL with SPARK (40%)

In this problem you will parse a set of activation records in XML format to extract the account numbers and model names.
SPARK is commonly used for ETL (Extract-Transform-Load) operations. Sometimes data is stored in line-oriented records, like the webserver logs we analyzed previously, but sometimes the data is in a multi-line format that must be processed as a *whole file*. In this exercise you will practice working with **file-based input format** instead of line-based input formats and use **named functions**.

Preparation:

- Understand how to use **file-based** input formats and **named functions** in SPARK
  → the SP1 lecture slides should be helpful!
- There are three **named functions** provided in `ActivationModels.pyspark/scalaspark` in the `hw8` folder in your SVN repository.
  - `getactivations` takes an XML string, parses it, and returns a collection of activation record objects.
  - `getmodel` takes an individual activation record object and finds the device model.
  - `getaccount` takes an individual activation record object and finds the account number.

To use those copy and paste them into the spark shell.

- The data for this problem can be found in your local file system under:

```
~/training_materials/dev1/data/activations
```

  Review the `activations` data. We will be interested in the following information: account-number and model. Each XML file contains data for all the devices activated by customers during a specific month. Create a folder named `loudacre` in the root directory of your HDFS and add the data to it.

- Consult the API documentation for RDD operations on the Spark API page accessible under `https://spark.apache.org/docs/1.6.0/`. From the API Docs menu, select either Scaladoc or Python API, depending on your language preference. You will find the RDD operations under the `SparkContex` class. **Review the list of available methods of the RDD class.**

Go back to these steps whenever you get stuck!

Use the **spark shell** for this problem and include all required commands in your written answers.

(a) Create an RDD from the activations dataset, where the entire content of each XML file is a single RDD element. What information is stored in the first value, what in the second value of the tuples in the resulting RDD?

(b) Which SPARK operation can be used to map the information in the second value to separate RDD elements? Create a new RDD of separate activation records. You can use the named function `getactivations` in `ActivationModels.pyspark/scalaspark`.

(c) Extract the account number and device model for each activation record, and save the list to a text file formatted as `account_number:model`. Store the file in `/loudacre/account-models` in HDFS. You can use the provided `getaccount` and `getmodel` functions described above to find the account number and the device model.

Include **all required pyspark or Scala commands** in your written answer in `hw8.pdf`.

Reflection (Bonus Problem for 5% up to a max. of 100%)

Reflect on your homework experience! Write a paragraph of at least 50 words to express your experiences and feelings when working on this assignment. Answer at least 2 of the following questions:

- What did you like/dislike about the assignment and why?
- What is the most important thing you learned and why do you think so?
- What surprised you, and why?
- Assuming you could start over again (with working on the assignment), what would you do differently and why?

Do not include/copy and past the questions into your reflection!

Submission Instructions

Store your reflection in the `hw8_reflection.txt` file provided in the `hw8`folder in your SVN repository and commit it.

This file should only include the reflection, **no other personal information** such as name, wustlkey, etc. reflections are not graded based on the content, but solely for completion.

To submit your reflection `cd` into the `hw8` folder and run:

```
$ svn commit -m 'hw8 reflection submission' .
```

Take 1 minute to provide an overall **star rating** for this homework.
Submit it via this link: `https://wustl.az1.qualtrics.com/jfe/form/SV_bIRKP1xobpO8yIB`.

Grading - no group work!

You can only earn bonus points if you write a *meaningful* **reflection** of **at least 50 words** answering at least 2 of the prompted questions and provide the corresponding **star rating**. You will **not** be graded on what your reflection says and the number of stars you assign, but rather solely the completion of it.
Bonus points are given to the **owner of the repository only**. No group work!.