CSE427 – Homework 8

M. Neumann

Due THU 04/07/2016 10am

Getting Started

Update your SVN repository. Find instructions on how to checkout, update, and commit to your SVN repository here: `http://sites.wustl.edu/neumann/resources/cse427s_resources/`

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

Hint: You can **check your submission** to the SVN repository by viewing `https://svn.seas.wustl.edu/repositories/<yourwustlkey>/cse427s_sp16` in a web browser (mind browser caching!).

More Good News

You survived your probation period at *Dualcore*!! In this homework you will help your employer to **plan their next marketing campaign** by **analyzing their sales data** and to **save money and ensure customer happyness** by **improving their product distribution**. If you haven't heard about your new task, find the instructions for Lab 6 on the course webpage and catch up!

Preparation (not for credit - but for your convenience)

If you haven't done so yet, complete **Lab 6 - Part II: Data Ingest with SQOOP** as you will need the data for this homework.

Run this command in the command line, **when you (RE)START your VM**:

```
$ ~/scripts/analyst/toggle_services.sh
```

Indicating Group Work

Use the file `partners.txt` to indicate group work. Follow these instructions exactly, to ensure to get credit!

- `partners.txt` needs to include up to 2 wustlkeys in the first two lines (one line per wustlkey)

- **first line/wustlkey is the repository, where the solution is located**. We will **only** consider the submission in this repository!
- Every student in a group needs to have **the same** `partners.txt` in the hwx folder in their repository (indicating that the partnership are **mutually accepted**)!
- If you do not have a partner, try to find one. If you want to submit on your own, indicate your wusltkey in the first line of `partners.txt` and leave the second line blank.

Problem 1: Analyze Sales Before and After the Advertisement Campaign (10%)

The goal of this task is to observe the effects that the recent advertising campaign has had on *Dualcore's* sales.
The directory for problems 1-3 is:

```
~/training_materials/analyst/exercises/disparate_datasets
```

Calculate the number of orders *Dualcore* received each month for the three months before the ad campaign began (February – April, 2013), as well as for the month during which their campaign ran (May, 2013).

(a) First, find out which dataset(s) is(are) needed for this task. Describe 2 different ways – one using pig and one not using pig – to generate sample datasets. Hint: Make it a habit and generate the sample datasets for each problem and use those to test your programs during development.

(b) Use the stubs provided in `count_orders_by_period.pig` to implement your solution for the task.

(c) Does the (full) data suggest that the advertising campaign we started in May led to a substantial increase in orders? Include the counts in your answer.

Submit your answer to (a) and (c) by editing the **`hw8.txt`** file in your SVN repository. Submit your answers to (b) by adding the `count_orders_by_period.pig` to the `hw8` folder in your SVN repository.
Add the files to your SVN repo before committing:

```
$ svn add count_orders_by_period.pig
$ svn commit -m 'hw8 submission' .
```

Problem 2: Analyze Sales of Advertised Product (20%)

Compare the sales of the tablet *Dualcore* advertised (product ID #1274348) during the same periods as in problem 1 to see whether the sales were actually influenced by the campaign or by other factors.

(a) First, find out which dataset(s) is(are) needed for this task. What is the first thing you do before writing any code?

(b) In the stubs `count_tablet_orders_by_period.pig` provided for this task you will see two FILTER statements directly after the LOAD commands. Discuss when and why this operation is beneficial.

(c) Use the stubs provided in `count_tablet_orders_by_period.pig` to implement your solution for the task.

(d) Does the data show an increase in sales of the advertised product corresponding to the month in which *Dualcore's* campaign was active? Include the counts in your answer.

Submit your answer to (a), (b), and (d) by editing the **`hw8.txt`** file in your SVN repository. Submit your answers to (c) by adding the `count_tablet_orders_by_period.pig` to the `hw8` folder in your SVN repository.

Add the files to your SVN repo before committing:

```
$ svn add count_tablet_orders_by_period.pig
$ svn commit -m 'hw8 submission' .
```

Problem 3: Assign Customers in Loyalty Program (30%)

Dualcore is considering starting a loyalty rewards program. This will provide exclusive benefits to their best customers, which will help to retain them. Another advantage is that it will also allow *Dualcore* to capture even more data about the shopping habits of their customers; for example, *Dualcore* can easily track their customers' in-store purchases when these customers provide their rewards program number at checkout. To be considered for the program, a customer must have made at least five purchases from *Dualcore* during 2012. These customers will be segmented into groups based on the total retail price of all purchases each made during that year:

- Platinum: Purchases totaled at least $10,000
- Gold: Purchases totaled at least $5,000 but less than $10,000
- Silver: Purchases totaled at least $2,500 but less than $5,000

Since we are considering the total sales price of orders in addition to the number of orders a customer has placed, not every customer with at least five orders during 2012 will qualify. In fact, only about one percent of the customers will be eligible for membership in one of these three groups.

(a) First, find out which dataset(s) is(are) needed for this task. Describe the datasets.

(b) Use the stubs provided in `bonus_02/loyalty_program.pig` to implement your solution for the task.

(c) How many customers are in each group? To answer this question, run your script and count the number of entries in the output file for each group. One way of getting the counts is to use the `hadoop fs -getmerge` command to create a local text file for each group from your scripts' output. Use `hadoop fs -help getmerge` to find out how to use this command. Then, you do not need PIG to get the counts from the output files; UNIX commands will do.

Submit your answer to (a) and (c) by editing the **hw8.txt** file in your SVN repository. Submit your answers to (b) by adding the `loyalty_program.pig` to the `hw8` folder in your SVN repository.
Add the files to your SVN repo before committing:

```
$ svn add loyalty_program.pig
$ svn commit -m 'hw8 submission' .
```

Problem 4: Extract Metadata from Call Center Recordings (20%)

Dualcore outsources its call center operations and costs have recently risen due to an increase in the volume of calls handled by these agents. Unfortunately, *Dualcore* does not have access to the call center's database, but they are provided with recordings of the calls stored in MP3 format. By using PIG's STREAM keyword to invoke a provided Python script, you can extract the category and timestamp from the files, and then analyze that data to learn what is causing the recent increase in calls.
The directory for problems 4 and 5 is:

```
~/training_materials/analyst/exercises/extending_pig
```

A Python script (`readtags.py`) is provided for extracting the metadata from the MP3 files. This script takes the path of a file on the command line and returns a record containing five tab-delimited fields: the file path, call category, agent ID, customer ID, and the timestamp of when the agent answered the call.

(a) Create a file named `call_list.txt` containing the paths of the files to analyze with one line for each file. You can use the UNIX `find` command with the `-name` option to do so. Use `man find` to find out how to use this command. The directory for the call data is `$ADIR/data/cscalls/`. Store the file locally and provide the command you used to create the file in the `hw8.txt` file.

(b) Use the stubs provided in `extract_metadata.pig` to implement your solution for the task.

(c) Since the Python library we are using for extracting the tags doesn't support HDFS, we run this script in local mode on the call recordings in `call_list.txt`. What command would you have to add to your script if you wanted to run it in HDFS? What does this command do?

(d) Run the script locally for April 2013 by passing this information as a parameter on the command line. How can you test if the parameter is substituted correctly without actually running a pig job? Provide the command you would use in the `hw8.txt` file. What are the top three call categories in that month?

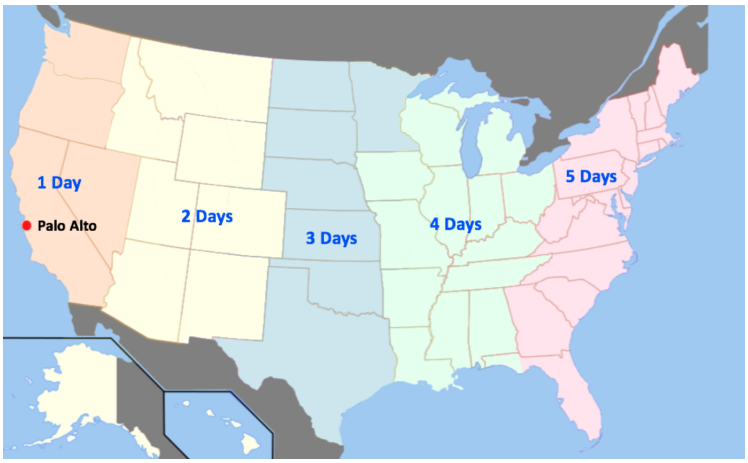
(e) Run the script again for May 2013. By looking at the top three call categories for this month, how does the call volume change? Which call category is showing a significant increase?

Submit your answer to (a), (c), (d), and (e) by editing the **hw8.txt** file in your SVN repository. Submit your answers to (b) by adding the `extract_metadata.pig` to the `hw8` folder in your SVN repository.
Add the files to your SVN repo before committing:

```
$ svn add extract_metadata.pig
$ svn commit -m 'hw8 submission' .
```

Problem 5: Best Location for Distribution Center (20%)

The analysis you just completed uncovered a problem. *Dualcore's* Vice President of Operations launched an investigation based on your findings and has now confirmed the cause: their online advertising campaign is indeed attracting many new customers, but many of them live far from *Dualcore's* only distribution center in Palo Alto, California. All shipments are transported by truck, so an order can take up to five days to deliver depending on the customer's location.



To solve this problem, *Dualcore* will open a new distribution center/warehouse to improve shipping times. The ZIP codes for the three proposed sites are 02118, 63139, and 78237. To decide on the best location you will compute the average distance between each of these locations and the locations of *Dualcore's* customers and select the location with the lowest average distance.

(a) As we cannot compute distances between ZIP codes, we have to transfer them into latitude/longitude pairs. Find a tab-delimited file mapping ZIP codes to latitude/longitude points here: `$ADIR/data/latlon.tsv` . Put this file into HDFS. Observe and run the `create_cust_location_data.pig` script and explain what it does (consider input, output, and data processing steps in your explanation).

(b) Create a data set containing the ZIP code, latitude, and longitude for the possible warehouse locations. You can use this command:
`egrep '^02118|^63139|^78237' $ADIR/data/latlon.tsv > warehouses.tsv`
Put this file into HDFS.

(c) Compute the average distance between each warehouse location and the locations of the *Dualcore* customers based on the latitude/longitude information. Fortunately, you can use a user defined function (UDF) distributed with DataFu, namely `HaversineDistInMiles`, to achieve this. Use the stubs provided in the `calc_average_distances.pig` script to implement your solution for the task.

(d) Which of the three proposed ZIP codes has the lowest average mileage to *Dualcore's* customers?

Submit your answer to (a) and (d) by editing the **`hw8.txt`** file in your SVN repository. Submit your answers to (b) and (c) by adding the `warehouses.tsv` and `calc_average_distances.pig` files to the `hw8` folder in your SVN repository.

Add the files to your SVN repo before committing:

```
$ svn add calc_average_distances.pig
$ svn add warehouses.tsv
$ svn commit -m 'hw8 submission' .
```

Bonus Problem (5% up to a max. of 100%) - no group work!

Write a review for this homework and store it in the file `hw8_review.txt` provided in your SVN repository (and commit your changes). This file should only include the review, **no other information** such as name, wustlkey, etc. Remember that you are not graded for the content of your review, solely it's completion.

You can only earn bonus points if you write **at least 50 words**. Bonus points are given to the **owner of the repository only** (no group work!).