CIS 4930 Introduction to Hadoop and Big Data

Assignment 6

Akshat Shah

U10348398

* 1. Analyzing Disparate Data with Pig
* Introduction

In this project, we practice various ways to combine, join and analyze product sales data from dualcore so we can observe the effects of their recent advertising campaign on sales.

1. Count\_orders\_by\_period.pig

* In this script we analyze the amount of orders dualcore received before and after their ad campaign began to if the campaign makes any difference.
* The format of the data in the original data input file orders.

1. Field: order\_id, data type:int
2. Field: cust\_id, data type: int
3. Field: order\_dtm, data type:chararray

* Data load method in pig

To load input data in this lab, instead of using the Grunt Shell we write a Pig Script to load and manipulate data. The data load function uses the LOAD function to load the input file and the AS function to mention the data types for the contents of the input the file. The function we used to load the input in this script was

data = LOAD '/dualcore/orders' AS (order\_id:int,

cust\_id:int,

order\_dtm:chararray);

* Data processing procedure.

First of all we the LOAD function to load the data and use the AS function to describe the data type for each field.

Then use the FILTER functionality to filter out the data by the order\_dtm field and use the SUBSTRING functionality in FOREACH….GENERATE Statement to create a new relation with just the orders year and month.

Then we count the number of orders in each month by using group and COUNT statements.

Then we use the DUMP functionality to dump all the data to the terminal.

* Data output and results.

(2013-02,76170)

(2013-03,84549)

(2013-04,87853)

(2013-05,115038)

Thus, the data supports the increase in sales during the advertising campaign. As there is an increasing the number of order in May.

1. Count\_tablet\_orders\_by\_period.pig

* In this script we analyze and count the number of orders for a particular product (a tablet) to see if the increase in sales was actually related to the advertising campaign.
* The format of the data in the original data input file orders.

1. Field: order\_id , data type:int
2. Field: cust\_id , data type: int
3. Field: order\_dtm , data type:chararray

* The format of the data in the original data input file order\_details.

1. Field: order\_id , data type:int
2. Field: prod\_id , data type: int

* Data load method in pig.

To load input data in this lab, instead of using the Grunt Shell we write a Pig Script to load and manipulate data. The data load function uses the LOAD function to load the file from its path and the AS function to mention the data types for the contents of the input file. The code used in this file is:

orders = LOAD '/dualcore/orders' AS (order\_id:int,

cust\_id:int,

order\_dtm:chararray);

details = LOAD 'dualcore/order\_details' AS (

order\_id:int,

prod\_id:int);

* Data processing procedure.
* First of all we the LOAD function to load the data and use the AS function to describe the data type for each field. In this script we load two files orders and order\_details.
* Then use the FILTER functionality to filter out the data by the order\_dtm field and only include the months we need to analyze.
* Then we use the FILTER functionality again to only include the product id we want to analyze.
* Then we use JOIN to join the two filtered data sets via order id so that only the entries we need are used.
* use the FOREACH….GENERATE Statement to create a new relation with just the month by using the SUBSTRING functionality.
* Then we count the number of orders for the product in each month by using GROUP to group by month and COUNT the number of tablet orders in each month.
* Then we use the DUMP functionality to dump all the data to the terminal.
* Data output and results.

(2013-02,3598)

(2013-03,3904)

(2013-04,4134)

(2013-05,49514)

Thus, the data supports the increase in sales during the advertising campaign.

.

3) average\_order\_size.pig

* In this script we analyze and count the average number of orders to see if the customers who buy the tablet also buy other stuff and the sales was actually related to the campaign.
* The format of the data in the original data input file orders.

1. Field: order\_id , data type:int
2. Field: cust\_id , data type: int
3. Field: order\_dtm , data type:chararray

* The format of the data in the original data input file order\_details.

1. Field: order\_id , data type:int
2. Field: prod\_id , data type: int

* Data load method in pig.

To load input data in this lab, instead of using the Grunt Shell we write a Pig Script to load and manipulate data. The data load function uses the LOAD function to mention the file and the AS function to mention the data types.

orders = LOAD '/dualcore/orders' AS (order\_id:int,

cust\_id:int,

order\_dtm:chararray);

details = LOAD 'dualcore/order\_details' AS (

order\_id:int,

prod\_id:int);

* Data processing procedure.
* First of all we the LOAD function to load the data and use the AS function to manipulate the data type for each field. In this script we load two files orders and order\_details
* Then use the FILTER functionality to filter out the data by the order\_dtm field and only include the months we need to analyze which were during the ad campaign.
* Then we use the FILTER functionality again to only include the product id we want to analyze.
* Then we use JOIN to join the order\_id from the two filered data sets so that only the orders we require are included.
* Then we use GROUP Statement to generate group the joined results from the two data sets by the order id.
* Then we count the number of orders by using FOREACH….GENERATE statement to generate a group and COUNT the number of tablet orders in the month.
* Then we GROUP the number of items.
* Then we use the FOREACH….GENERATE functionality to calculate the average for the number of items for each tablet order.
* Then we use the DUMP functionality to dump all the data to the terminal.
* Data output and results.

(2.118472995933963)

Thus, the data supports the idea that the average customer that ordered the tablet ordered more than one item

1. Loyalty\_program.pig

* In this script we help dualcore create a customer loyalty program with three tiers: Silver, Gold, and Platinum. This would allow dualcore to capture more data about the shopping habits of their customers.
* The format of the data in the original data input file orders.

1. Field: order\_id , data type:int
2. Field: cust\_id , data type: int
3. Field: order\_dtm , data type:chararray

* The format of the data in the original data input file order\_details.

1. Field: order\_id , data type:int
2. Field: prod\_id , data type: int

* The format of the data in the original data input file products.

1. Field: prod\_id , data type: int
2. Field: brand, data type:chararray
3. Field: name , data type:chararray
4. Field: price , data type:int
5. Field: cost , data type:int
6. Field: shipping\_wt , data type:int

* Data load method in pig.

To load input data in this lab, instead of using the Grunt Shell we write a Pig Script to load and manipulate data. The data load function uses the LOAD function to mention the file and the AS function to mention the data types.

orders = LOAD '/dualcore/orders' AS (order\_id:int,

cust\_id:int,

order\_dtm:chararray);

details = LOAD 'dualcore/order\_details' AS (

order\_id:int,

prod\_id:int);

products = LOAD 'dualcore/products' AS (prod\_id:int,

brand:chararray,

name:chararray,

price:int,

cost:int,

shipping\_wt:int);

* Data processing procedure.
* First of all we the LOAD function to load the data and use the AS function to describe the data type for each field. In this script we load three files orders, products and order\_details.
* Then use the FILTER functionality to filter out the data by the order\_dtm field and only include the year we need to analyze.
* Then we use the GROUP functionality again to group the orders by cust\_id and count the number of orders for each customer by using COUNT in FOREACH….GENERATE.
* Then we use FILTER again to filter customers with more than 5 orders.
* Then we use JOIN to join the two data sets we filtered.[join 1]
* Then we use JOIN again twice to join the the dataset we previously joined(join 1) and the dataset we filtered to only have the year we need to analyze by their order\_id. [join 2]
* Then we JOIN again to join the datasets we previously joined [join 2] or the details of the customers orders and the products dataset by their product id.
* Then we use the FOREACH….GENERATE statement to generate a list of customer id and products for each customer who had more than 5 orders.
* Then we GROUP customer prices by customer id and use SUM in FOREACH…GENERATE to find the total for each customer.
* Then we use the SPLIT functionality to split the customer totals into three for each tier of the loyalty program.
* Then we use the STORE functionality to store all the data to the three files for three tiers of loyalty programs.
* Data output and results.

The output was as expected each text file outputted for the three tiers contains the customer id and their total. The total number of customers in each program were:

111655 platinum

5241 silver

10976 gold

1. Extending Pig with Streaming and UDFs
   1. Project introduction

In this part of the lab, we use pigs STREAM keyword to invoke a python script and extract the timestamp and category data from the scripts stored in MP3 format and find out the reason for recent increase in complaints. The format of the data in the original data input file.

* + For extract\_metadata.pig:

The format should be MP3 which gets processed by a python script readtags.py

* 1. Data load method in pig.
  + For extract\_metadata.pig:

We extract the metadata using a python script readtags.py and use the output of the find command to create data in the required format to LOAD into the script.

* 1. Data processing procedure.
  + For extract\_metadata.pig
    - We define ‘readtags.py’ as the tagreader
    - Then we use LOAD to load call\_list.txt as a chararray
    - The we use the STREM functionality to extract the metadata by using the tagreader.
    - We get data as:
      * Path : chararray
      * Category: chararray
      * Agent\_id: chararray
      * Customer\_id : chararray
      * Timestamp: chararray
    - Then we use FILTER functionality to filter out the data given by the user
    - Then we group the calls by their category and count the total number of calls for each category
    - Then we sort the number of calls for each category in descending order,
    - Then we use DUMP functionality to print the top 3 into the terminal
  1. Data output and results.
  + For extract\_metadata.pig
    - For the month 2013-04

(BILLING,16)

(TECH\_SUPPORT,14)

(STORE\_LOCATIONS,6)

* + - For the month 2013-05

(SHIPPING\_DELAY,116)

(TECH\_SUPPORT,24)

(STORE\_LOCATIONS,24)

We can see that the majority of increase in complains was due to shipping delays.

1. Project introduction

In the second part we use two files to create customer location data and calculate the average distance in order to find the best location for a distribution center and reduce the amount of complains

* + We add a tab-delimited file mapping zip codes to latitude/longitude points to HDFS
  + For create\_cust\_location\_data.pig
    - For The format of the data in the original data input file orders.
    1. Field: order\_id , data type:int
    2. Field: cust\_id , data type: int
    3. Field: order\_dtm , data type:chararray
    - The format of the data in the original data input file order\_details.
      * 1. Field: fname , data type:chararray
        2. Field: lname , data type:chararray
        3. Field: address, data type:chararray
        4. Field: city , data type:chararray
        5. Field: state , data type:chararray
        6. Field: zipcode , data type:chararray
    - The format of the data in the original data input file latlon
    1. Field: zip , data type:chararray
    2. Field:lat , data type:double
    3. Field:lon , data type:double
  + For calc\_average\_distance.pig
    - For The format of the data in the original data input file cust\_locations.
      * Field: zip , data type:chararray
      * Field:lat , data type:double
      * Field:lon , data type:double
    - The format of the data in the original data input file warehouses.
      * Field: zip , data type:chararray
      * Field:lat , data type:double
      * Field:lon , data type:double

1. Data load method in pig.
   * For create\_cust\_location\_data.pig and For calc\_average\_distance.pig :

To load input data in this lab, instead of using the Grunt Shell we write a Pig Script to load and manipulate data. The data load function uses the LOAD function to load the input file and the AS function to mention the data types for the contents of the input the file

1. Data processing procedure.
   * For create\_cust\_location\_data.pig:
     + We use the LOAD…AS function to load the three datasets
     + Then we use FILTER to filter records during the ad campaign
     + Then we use the FILTER functionality to exclude the customers in two day delivery area.
     + Then we use JOIN functionality to join recent orders and distant customers by customer id.
     + Then we generate a unique zip for each order
     + Then we JOIN the zip with latitude and longitude data
     + Then we store the zip, latitude and longitude for further analysis.
   * For calc\_average\_distances.pig:
     + We use the LOAD…AS function to load the two datasets
     + Then we use CROSS functionality to create a combination for every warehouse location and customer location.
     + Then we use the DIST functionality to calculate the distance of the customer from every warehouse.
     + Then we use JOIN functionality to join recent orders and distant customers by customer id.
     + Then we calculate the average distance for all customers to each warehouse.
     + Then we DUMP the average distance for each warehouse
2. Data output and results.

The results for average distances are

* + (02118,1059.3379591467183)
  + (63139,423.01945102918006)
  + (78237,955.8300496917677)

Thus, 78237 is the most appropriate zip for dualcores new warehouse.