Report on Analyzing Disparate Data Sets with Pig

Introduction:

In this project, we combined, joined and analyzed the product sales data so we can observe the effects that the recent advertising campaign by DualCore has had on sales.

(i) **Show Per-Month Sales Before and After Campaign:**
This step involves calculating the number of orders Dualcore received each month for the three months before their ad campaign from Feb to May, 2013.

- **The format data in original input file:**
The schema for input file is unknown.
- **Data Load method in Pig:**
Once the LOAD statement is defined in script, we may run the script for data to check if the fields are in the expected order and the values appear similar to how we defined the fields while using LOAD statement.
- **Data Processing Procedure:**
The data is first filtered by defining a FILTER statement for the months from Feb to May, 2013. Following that we created a new relation with just one field, i.e., the order's year and month and lastly displaying the count by month after counting the number of orders in each of the months.
- **Data Output & Result:**
The output upon this pig script delivers order counts by months and thus it helps analyzing the increase in sales of advertised product during the campaign.

(ii) **Count Advertised Product Sales by Month:**
The previous step suggests that sales increased dramatically the same month Dualcore began advertising. We'll compare the sales of the specific product Dualcore advertised (product ID #1274348) during the same period to see whether the increase in sales was actually related to their campaign. We will be joining two data sets during this portion of the project.

- **The format data in original input file:**
Schema of input file is unknown.
- **Data Load method in Pig:**
Once the LOAD statement is defined in script, we may run the script for data to check if the fields are in the expected order and the values appear similar to how we defined the fields while using LOAD statements.
- **Data Processing Procedure:**
We join the two relations on the order_id field both the input file have have in common and create a new relation from the joined data that contains a single field i.e., order year and month. Further, we group the records by month and then count the records in each group. We can now analyze an increase in sales of the advertised product corresponding to the month in which Dualcore's campaign was active.

- **Data Output & Result:**
  The output upon running the script delivers the result that shows an increase in sales of the advertised product corresponding to the month in which Dualcore's campaign was active.

(iii) Calculation of Average Order Size

This step demands the calculation of average number of items for all orders that contain the advertised tablet during the campaign period.

- **The format data in original input file:**
  Schema of input file is unknown.
- **Data Load method in Pig:**
  Once the LOAD statement is defined in script, we may run the script for data to check if the fields are in the expected order and the values appear similar to how we defined the fields while using LOAD statements.
- **Data Processing Procedure:**
  We filter the orders by date to include only those placed during the campaign period and exclude any orders which do not contain the advertised product with id #1274348. Further a new relation containing the order_id and product_id fields for these orders is created. Up next, we calculate the total number of products per order and then average number of products for all orders.
- **Data Output & Result:**
  The output upon running the script delivers the result that shows that the average order contained additional items to the tablet Dualcore advertised.

(iv) Segment Customers for Loyalty Program

This step involves writing a script needed to filter the list of orders based on date, grouping them by customer ID, counting the number of orders per customer and then filtering this to exclude any customer who did not have at least five orders. Then we will join this information with the order details and products data sets in order to calculate the total sales of those orders for each customer, splitting them into the groups based on the criteria described above, and then writing the data for each group (customer ID and total sales) into a separate directory in HDFS.

- **The format data in original input file:**
  Schema of input file is unknown.
- **Data Load method in Pig:**
  Once the LOAD statement is defined in script, we may run the script for data to check if the fields are in the expected order and the values appear similar to how we defined the fields while using LOAD statements.
- **Data Processing Procedure:**
  Upon loading the data sets, we filter the orders only from 2012 and customers with at least 5 orders. Further, we will then join this information with the order details and products data sets in order to calculate the total sales of those orders for each customer, splitting them into the groups, and then write the data for each group (customer ID and total sales) into a separate directory in HDFS.

- **Data Output & Result:**
  The output upon running the script results in customers segmented according to the loyalty program based on the criterion described already.

Report on Extending Pig with Streaming and UDFs

Introduction:

In this project we will use the STREAM keyword in Pig to analyze metadata from Dualcore's customer service call recordings to identify the cause of a sudden increase in complaints. You will then use this data in conjunction with a user-defined function to propose a solution for resolving the problem.

(i) Extract Call Metadata:

A Python script (readtags.py) has been 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.

- **The format data in original input file:**
  The input file is a text file.
- **Data Load method in Pig:**
  Once the LOAD statement is defined in script, we may run the script for data to check if the fields are in the expected order and the values appear similar to how we defined the fields while using LOAD statements.
- **Data Processing Procedure:**
  We replace the hardcoded parameter in the SUBSTRING function used to filter by month with a parameter named MONTH whose value you can assign on the command line. This will make it easy to check the leading call categories for different months without having to edit the script. We further add the code necessary to count calls by category and display the top three categories to the screen.
- **Data Output & Result:**
  The output upon running the script delivers the result that confirms that not only is call volume is substantially higher in May. The SHIPPING_DELAY category has more than twice the amount of calls as the other two.

(ii) Choose Best Location for Distribution Center:

Dualcore is supposed to open a new distribution center to improve shipping times. The ZIP codes for the three proposed sites are 02118, 63139, and 78237. We will look up the latitude and longitude of these ZIP codes, as well as the ZIP codes of customers who have recently ordered, using a supplied data set. Once we have the coordinates, we will invoke the use the HaversineDistInMiles UDF distributed with DataFu to determine how far each customer is from the three data centers. We will then calculate the average distance for all customers to each of these data centers in order to propose the one that will benefit the most customers.

- **The format data in original input file:**
  The input file is a tab-delimited file.
- **Data Load method in Pig:**

Once the LOAD statement is defined in script, we may run the script for data to check if the fields are in the expected order and the values appear similar to how we defined the fields while using LOAD statements.

- **Data Processing Procedure:**
  We will use the HaversineDistInMiles function to calculate the distance from each customer to each of the three proposed warehouse locations. This function requires us to supply the latitude and longitude of both the customer and the warehouse. While the script we executed created the latitude and longitude for each customer, we must create a data set containing the ZIP code latitude, and longitude for these warehouses.

- **Data Output & Result:**
  The output upon running the script delivers the result with proposed ZIP codes having the lowest average mileage to Dualcore's customers.