APPLICATION PROJECT

COS20028 – Big Data Architecture and Application

SYED OMAIR MAQDOOM MOHIUDDIN

102863768

Executive Summary

The aim of this report is to analyze the company's overall performance. Based on the outputs produced using Apache Pig alongside of descriptive analysis and visualization of the customer satisfaction, sales, average feedback for top selling products and frequent buyers. Additionally, recommendations were made for the company to maximize sales and boost customer base.

The following are the pertinent insights of the analysis:

- The customer satisfaction was constant between 2012 to 2013
- 1274348, 1274673 and 1274021 were the most sold products in 2 years period
- Product 1274673 has the least customer satisfaction with an average feedback score of 1.11 within the top 20 selling products
- Customer 1043182 was the most loyal customer, and their average feedback score was 2.09
- 65% of the customers haven't returned to the company after purchasing the company's goods.
- 9.2% of the customers have placed 3 or more orders.

The company's overall performance level tends to improve between 2012-2013. However, by using the data and critical insights presented for decision-making, as well as implementing the given recommendations, the company has a significant prospect of boosting sales and customer volume.

Table of Contents

Executive Summary 1
Introduction 3
Basic Goal 3
Solving Logic 5
Solution 5
First Additional Goal 10
Definition and Solving Logic 11
Solution 11
Second Additional Goal 14
Definition and Solving Logic 14
Solution 15
Conclusion 18
References 18

Introduction

The company would like to know their customer satisfaction trend over the 2 years period, key insights which aid in decision-making and maximize their sales and boost the customer base.

We analyzed and visualized the data provided to the company for them to do so. This report contains data insights based on analysis at several levels. We will employ Apache Pig to analyze the data in this scenario. As it provides an alternative to writing low-level MapReduce Code and is widely used for data processing (ETL). Apache Pig consists of several features which enables advanced data analysis and processing. Pig Latin code will be used for coding since Pig understands Pig Latin code to construct MapReduce tasks, which are subsequently sent to the Hadoop cluster. Furthermore, we can run Pig Latin Scripts from the terminal, and the code is reusable.

Proposed methods and suggestions are given to present the company with the information they need for decision making.

Basic Goal

As the basic goal provided by the company is to know the trend in the customer satisfaction from 2012 to 2013. To achieve this basic goal, we need to get the average feedback score for each month in the individual months and analyze the results to determine the trend. Before we start let us know more about the given data, what does it say and how can we use individual information to gain key insights. Understanding the data is one of the critical parts for data analysis. In the below table we will discuss about the key feature of the given data.

Column Name	Description	Key Feature
Date and Time	It is the date and time stamp for the transaction.	We may gain significant insights from this data, such as which (time/day/month) has the highest and lowest transaction (sale) levels, and trends across various time periods.
User ID	This is a unique id which uniquely identifies each customer.	This information can be used to evaluate client loyalty, customers with highest and lowest order count and many more.
Product ID	This is a unique id which uniquely identifies each product.	Using this information, we can examine the products sale, which is the highest and lowest sold product and much more.
Feedback Score	This is the feedback score given by the customers (/5) for the products.	The company might utilize this feedback data to determine which products have the highest and lowest satisfaction rates, the pace at which feedback is received over time (per month), the customer satisfaction pattern, and far more.
Comments	These are comments given by customers on each product.	The company may leverage positive customers feedback (comments) to advertise items (marketing), evaluate negative customer feedback to improve products based on their experiences, and so on.

Solving Logic

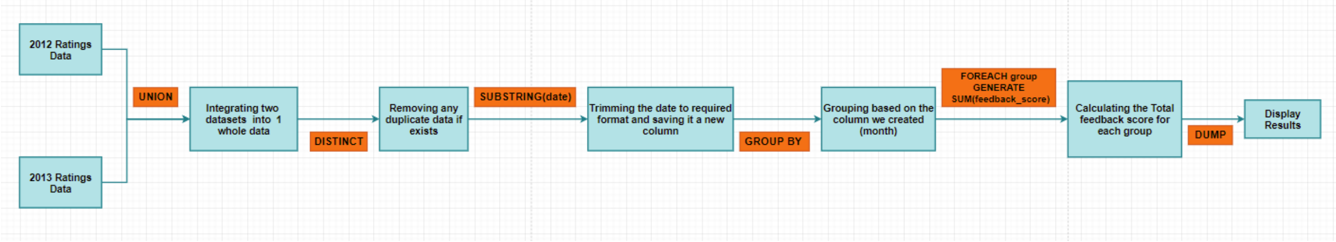
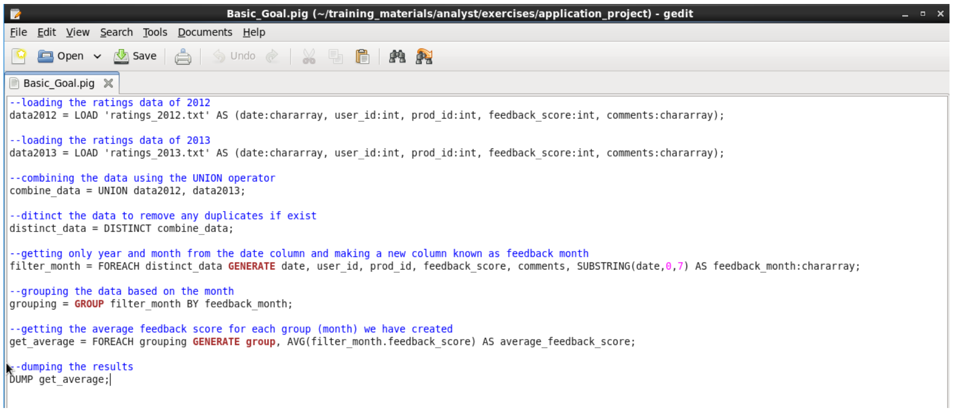

2012 Ratings Data
2013 Ratings Data
UNION
Integrating two datasets into 1 whole data
DISTINCT
Removing any duplicate data if exists
SUBSTRING(date)
Trimming the date to required format and saving it a new column
GROUP BY
Grouping based on the column we created (month)
FOREACH group GENERATE SUM(feedback_score)
Calculating the Total feedback score for each group
DUMP
Display Results

Figure 1: Sequence Diagram Displaying the Solving Logic

The series of steps we'll take to get the desired result is illustrated in Figure 1. To begin, we'll load the data for 2012 and 2013 from the provided files and combine all the data into a single table. We don't know whether there are any duplicate records in the data because we haven't performed data cleaning, therefore we'll distinct the data to eliminate duplicates if any exist. Followed by, we want to retrieve the data on monthly basis, trim the data and time to get it in 'YYYY-MM' format and save data into a new field. Group the data by recently created field to obtain the data for each month. Calculate the average feedback score for each of the groups we've generated. Finally display the results.

Solution

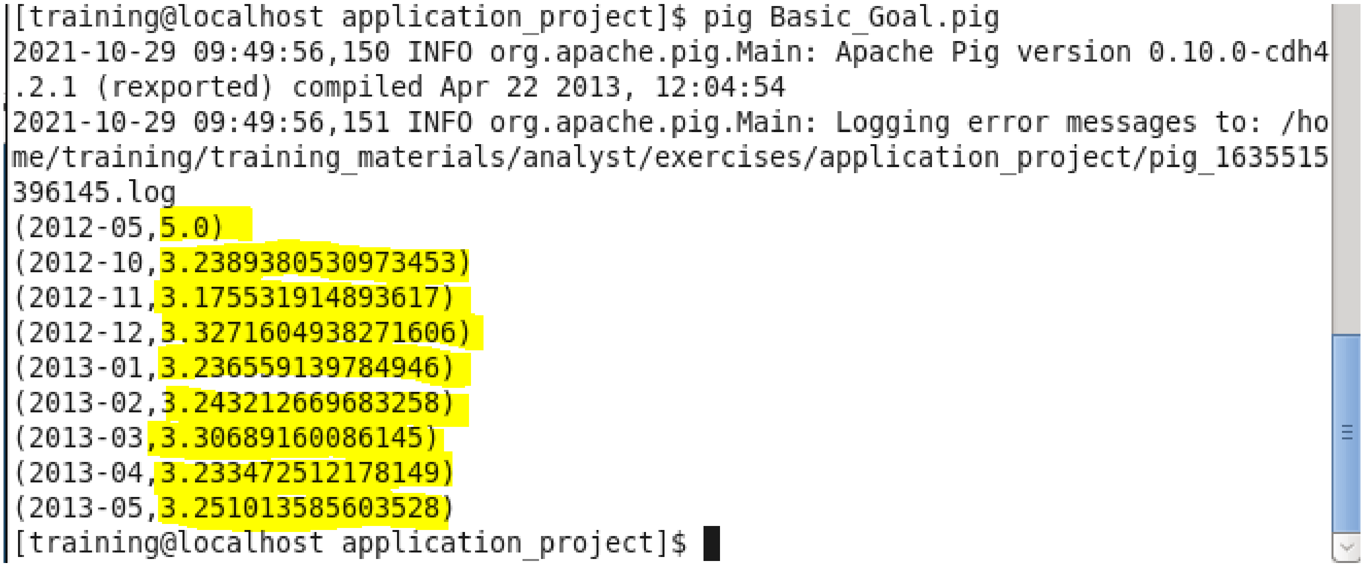
The Pig Latin Script is displayed in the figure below. In the script (Basic_Goal.pig), we load the data from both the data files using 'LOAD' Function. Integrated the data into a single table using 'UNION' operator. 'DISTINCT' the data to remove duplicates. Using the 'FOREACH' statement, trim the date and time data to get Year and month using 'SUBSTRING' function, save the trimmed data into new field and get the required datasets in required format. Group the data using 'GROUP' function based on recently created year and month field. Using 'AVG' function to get the average of feedback score for each month by using 'FOREACH' expression. Lastly, dump the results using 'DUMP' function.


Basic_Goal.pig (~/training_materials/analyst/exercises/application_project) - gedit
File Edit View Search Tools Documents Help
Open
Save
Undo
Basic_Goal.pig

```
--loading the ratings data of 2012
data2012 = LOAD 'ratings_2012.txt' AS (date:chararray, user_id:int, prod_id:int, feedback_score:int, comments:chararray);

--loading the ratings data of 2013
data2013 = LOAD 'ratings_2013.txt' AS (date:chararray, user_id:int, prod_id:int, feedback_score:int, comments:chararray);

--combining the data using the UNION operator
combine_data = UNION data2012, data2013;

--ditinct the data to remove any duplicates if exist
distinct_data = DISTINCT combine_data;

--getting only year and month from the date column and making a new column known as feedback month
filter_month = FOREACH distinct_data GENERATE date, user_id, prod_id, feedback_score, comments, SUBSTRING(date,0,7) AS feedback_month:chararray;

--grouping the data based on the month
grouping = GROUP filter_month BY feedback_month;

--getting the average feedback score for each group (month) we have created
get_average = FOREACH grouping GENERATE group, AVG(filter_month.feedback_score) AS average_feedback_score;

--dumping the results
DUMP get_average;
```

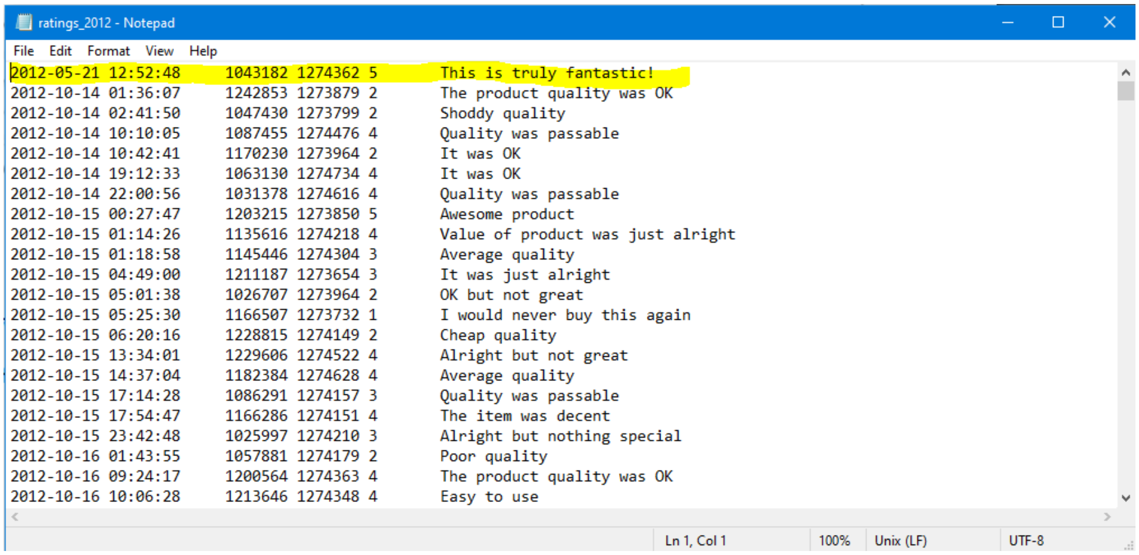
Figure 2: Coding for Average Feedback Score for Individual Month [2012-2013]

The below figure is the output we received for the pig script we used above. We got the results for average feedback score per month [2012-2013]. As it can be seen from the figure, the average score looks fluctuating between 3.1 and 3.3 (rounded to one decimal) for maximum months except for May 2012. Figure 4 shows that there was just one order placed in the month of May 2012. We can consider it as an outlier and remove from the analysis because it may bias the data analysis towards it (graph).



```
[training@localhost application_project]$ pig Basic_Goal.pig
2021-10-29 09:49:56,150 INFO org.apache.pig.Main: Apache Pig version 0.10.0-cdh4
.2.1 (rexported) compiled Apr 22 2013, 12:04:54
2021-10-29 09:49:56,151 INFO org.apache.pig.Main: Logging error messages to: /ho
me/training/training_materials/analyst/exercises/application_project/pig_1635515
396145.log
(2012-05,5.0)
(2012-10,3.2389380530973453)
(2012-11,3.175531914893617)
(2012-12,3.3271604938271606)
(2013-01,3.236559139784946)
(2013-02,3.243212669683258)
(2013-03,3.30689160086145)
(2013-04,3.233472512178149)
(2013-05,3.251013585603528)
[training@localhost application_project]$ 
```

Figure 3: output for Average Feedback Score for Individual Month [2012-2013]



```
2012-05-21 12:52:48     1043182 1274362 5       This is truly fantastic!
2012-10-14 01:36:07     1242853 1273879 2       The product quality was OK
2012-10-14 02:41:50     1047430 1273799 2       Shoddy quality
2012-10-14 10:10:05     1087455 1274476 4       Quality was passable
2012-10-14 10:42:41     1170230 1273964 2       It was OK
2012-10-14 19:12:33     1063130 1274734 4       It was OK
2012-10-14 22:00:56     1031378 1274616 4       Quality was passable
2012-10-15 00:27:47     1203215 1273850 5       Awesome product
2012-10-15 01:14:26     1135616 1274218 4       Value of product was just alright
2012-10-15 01:18:58     1145446 1274304 3       Average quality
2012-10-15 04:49:00     1211187 1273654 3       It was just alright
2012-10-15 05:01:38     1026707 1273964 2       OK but not great
2012-10-15 05:25:30     1166507 1273732 1       I would never buy this again
2012-10-15 06:20:16     1228815 1274149 2       Cheap quality
2012-10-15 13:34:01     1229606 1274522 4       Alright but not great
2012-10-15 14:37:04     1182384 1274628 4       Average quality
2012-10-15 17:14:28     1086291 1274157 3       Quality was passable
2012-10-15 17:54:47     1166286 1274151 4       The item was decent
2012-10-15 23:42:48     1025997 1274210 3       Alright but nothing special
2012-10-16 01:43:55     1057881 1274179 2       Poor quality
2012-10-16 09:24:17     1200564 1274363 4       The product quality was OK
2012-10-16 10:06:28     1213646 1274348 4       Easy to use
```

Figure 4: Outliers Identified in the 2012 Data

The below table shows the average feedback score (rounded to two decimal places) for each month (after removing outliers).

Month	Average feedback Score
October 2012	3.23
November 2012	3.17
December 2012	3.33
January 2013	3.23
February 2013	3.24
March 2013	3.31
April 2013	3.23
May 2013	3.25

We plotted bar (Figure 5) and line (Figure 6) graph to aid visualize the outcome and make accurate analysis. As the bar graph illustrates, the average score decreased from October 2012 to November 2012, and a sharp increase in December 2012. Average Score got down from 3.32 to 3.23 from December 2012 to January 2013, gradually increase up to March 2013, followed by a downfall in April 2013 and slightly increase in May 2013. Overall, December 2012 (3.33) has recorded the highest average feedback score and least with November 2012 (3.17) in the duration of 8 month.

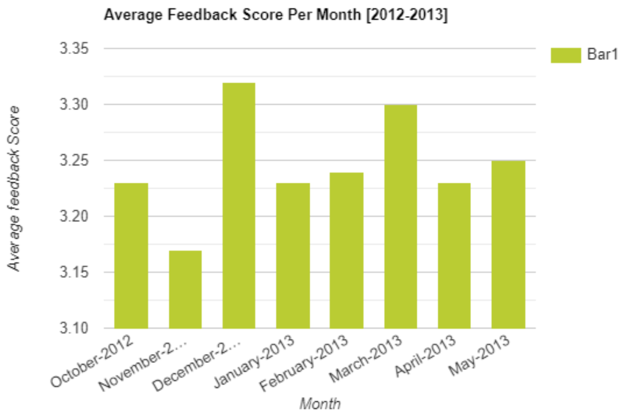


Figure 5: Bar Graph for Average Feedback Score Per Month

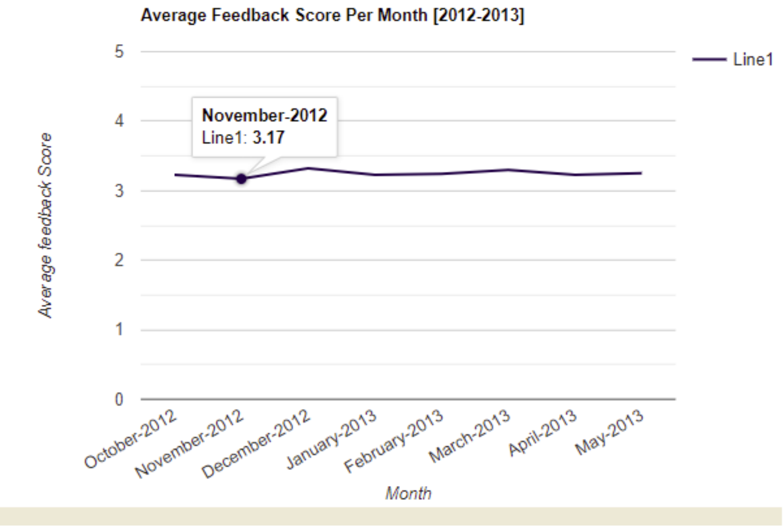
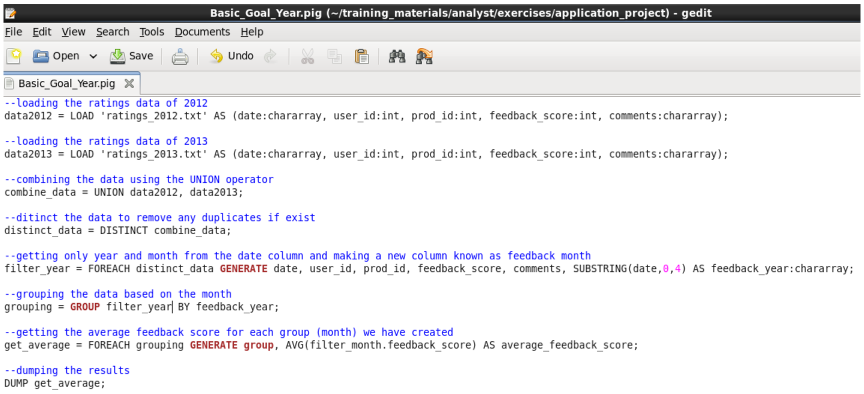
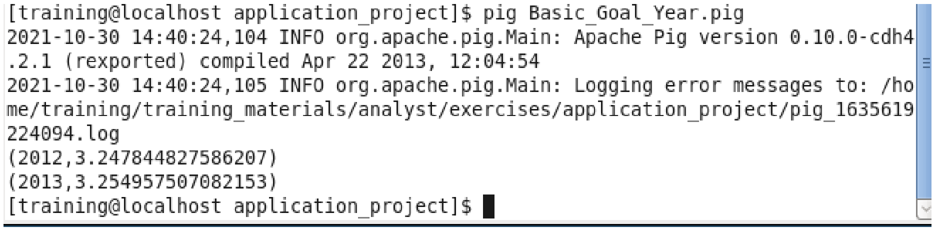


Figure 6: Line Graph for Average Feedback Score Per Month


Basic_Goal_Year.pig (~/training_materials/analyst/exercises/application_project) - gedit
File Edit View Search Tools Documents Help
Open
Save
Undo
Basic_Goal_Year.pig

```
--loading the ratings data of 2012
data2012 = LOAD 'ratings_2012.txt' AS (date:chararray, user_id:int, prod_id:int, feedback_score:int, comments:chararray);

--loading the ratings data of 2013
data2013 = LOAD 'ratings_2013.txt' AS (date:chararray, user_id:int, prod_id:int, feedback_score:int, comments:chararray);

--combining the data using the UNION operator
combine_data = UNION data2012, data2013;

--ditinct the data to remove any duplicates if exist
distinct_data = DISTINCT combine_data;

--getting only year and month from the date column and making a new column known as feedback month
filter_year = FOREACH distinct_data GENERATE date, user_id, prod_id, feedback_score, comments, SUBSTRING(date,0,4) AS feedback_year:chararray;

--grouping the data based on the month
grouping = GROUP filter_year BY feedback_year;

--getting the average feedback score for each group (month) we have created
get_average = FOREACH grouping GENERATE group, AVG(filter_month.feedback_score) AS average_feedback_score;

--dumping the results
DUMP get_average;
```

Figure 7: Coding for Average Feedback Score Per Year

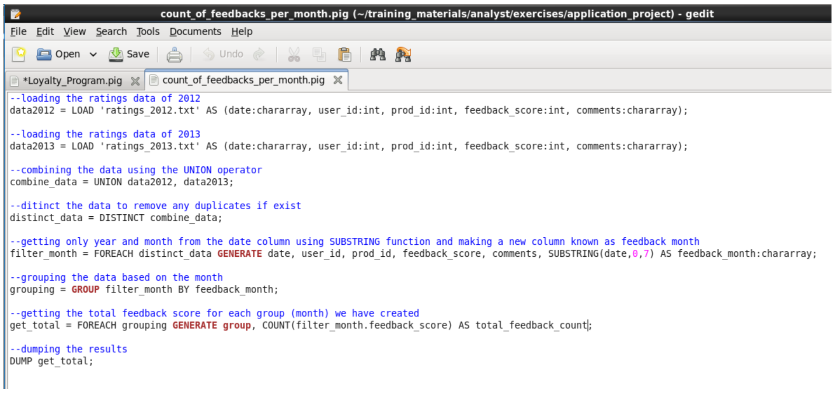


```
[training@localhost application_project]$ pig Basic_Goal_Year.pig
2021-10-30 14:40:24,104 INFO org.apache.pig.Main: Apache Pig version 0.10.0-cdh4
.2.1 (rexported) compiled Apr 22 2013, 12:04:54
2021-10-30 14:40:24,105 INFO org.apache.pig.Main: Logging error messages to: /ho
me/training/training_materials/analyst/exercises/application_project/pig_1635619
224094.log
(2012,3.247844827586207)
(2013,3.254957507082153)
[training@localhost application_project]$
```

Figure 8: Output for Average Feedback Score Per Year

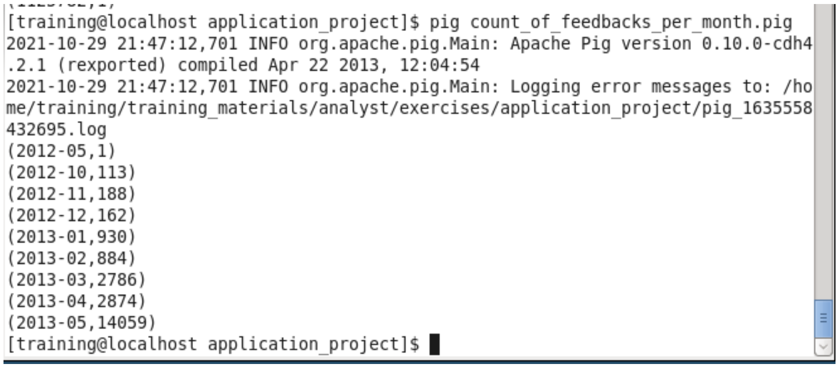
Although the feedback score fluctuated for individual month in the given timeline, the line graph in Figure 6 displays that the average feedback score was consistent over the time and oscillated between 3.17 and 3.33. Figure 8 illustrates the average feedback score for 2012 and 2013. The average feedback score for 2013 was 3.25 (rounded to 2 decimal places) which was equal to the average feedback score of 2012 3.25 (rounded to 2 decimal places). To conclude, the customer satisfaction (Feedback Score) remained constant for both the years.

Below figure shows the Pig Latin Script for retrieving the Total feedback count received per month. In the script (count_of_feedback_per_month.pig), we load the data from both the data files using 'LOAD' Function. Integrated the data into a single table using 'UNION' operator. 'DISTINCT' the data to remove duplicates. Using the 'FOREACH' statement, trim the date and time data to get Year and month using 'SUBSTRING' function, save the trimmed data into new field and get the required datasets in required format. Group the data using 'GROUP' function based on recently created year and month field. Using 'COUNT function to get the count of feedback count for each month by using 'FOREACH' expression. At the end, dump the results. Moreover, we have slightly updated the previous code to get the required output.


count_of_feedbacks_per_month.pig (~/training_materials/analyst/exercises/application_project) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
*Loyalty_Program.pig
count_of_feedbacks_per_month.pig

```
--loading the ratings data of 2012
data2012 = LOAD 'ratings_2012.txt' AS (date:chararray, user_id:int, prod_id:int, feedback_score:int, comments:chararray);

--loading the ratings data of 2013
data2013 = LOAD 'ratings_2013.txt' AS (date:chararray, user_id:int, prod_id:int, feedback_score:int, comments:chararray);

--combining the data using the UNION operator
combine_data = UNION data2012, data2013;

--ditinct the data to remove any duplicates if exist
distinct_data = DISTINCT combine_data;

--getting only year and month from the date column using SUBSTRING function and making a new column known as feedback month
filter_month = FOREACH distinct_data GENERATE date, user_id, prod_id, feedback_score, comments, SUBSTRING(date,0,7) AS feedback_month:chararray;

--grouping the data based on the month
grouping = GROUP filter_month BY feedback_month;

--getting the total feedback score for each group (month) we have created
get_total = FOREACH grouping GENERATE group, COUNT(filter_month.feedback_score) AS total_feedback_count;

--dumping the results
DUMP get_total;
```

Figure 9: Coding for Count of feedback Received per Month [2012-2013]



```
[training@localhost application_project]$ pig count_of_feedbacks_per_month.pig
2021-10-29 21:47:12,701 INFO org.apache.pig.Main: Apache Pig version 0.10.0-cdh4
.2.1 (rexported) compiled Apr 22 2013, 12:04:54
2021-10-29 21:47:12,701 INFO org.apache.pig.Main: Logging error messages to: /ho
me/training/training_materials/analyst/exercises/application_project/pig_1635558
432695.log
(2012-05,1)
(2012-10,113)
(2012-11,188)
(2012-12,162)
(2013-01,930)
(2013-02,884)
(2013-03,2786)
(2013-04,2874)
(2013-05,14059)
[training@localhost application_project]$
```

Figure 10: Output for Count of feedback Received per Month [2012-2013]

As it can be seen from Figure 10, feedback count is steadily increasing from October 2012 to May 2013, meaning that the customers are actively participating in survey and providing their valuable feedback.

First Additional Goal

The first additional goal is to identify the top 20 hot selling products with best average feedback score or worst average feedback score and top 20 customers with highest orders and their overall satisfaction. To achieve this aim, for product analysis, we need to calculate the average feedback score and count of orders for each individual product.

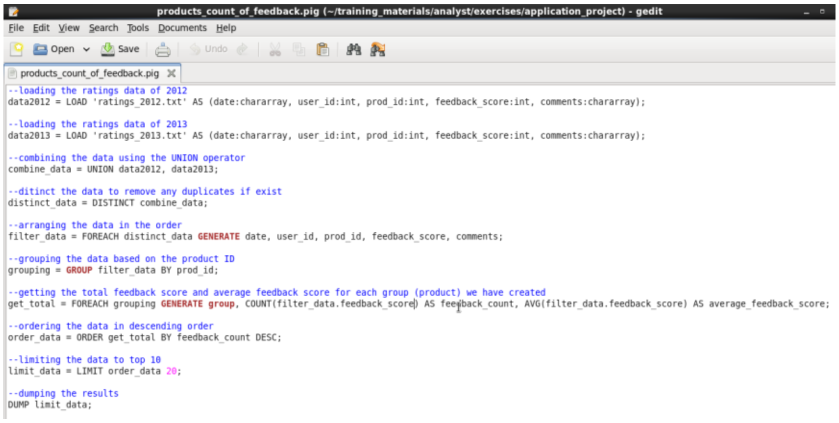
Definition and Solving Logic

To get the top 20 hot selling products and their average feedback we'll load the data for 2012 and 2013 from the provided data files and combine all the data into a single table. We don't know whether there are any duplicate records in the data because we haven't performed data cleaning, therefore we'll distinct the data to eliminate duplicates if any exist. Followed by, we will arrange the data into the required format and group the data by product ID. Calculate the count of orders and average feedback score for individual group (product) we've generated, sort the results in descending order and limit the data to 20. Finally display the results.

For identifying the satisfaction rate of the top 20 customers who have placed the most orders We'll load the data for 2012 and 2013 from the provided data files and combine all the data into a single table. We don't know whether there are any duplicate records in the data because we haven't performed data cleaning, therefore we'll distinct the data to eliminate duplicates if any exist. Followed by, we will arrange the data into the required format and group the data by User ID. Calculate the count of orders and average feedback score for individual group (customer) we've generated, sort the results in descending order and limit the data to 20. Finally display the results.

Solution

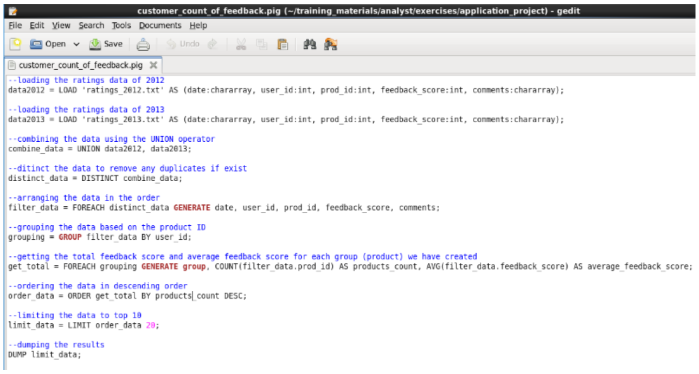
Below figure shows the Pig Latin Script for retrieving top 20 hot selling products with highest average feedback score or worst average feedback score. In the script (products_count_of_feedback.pig), we load the data from both the data files using 'LOAD' Function. Integrated the data into a single table using 'UNION' operator. 'DISTINCT' the data to remove duplicates. Using the 'FOREACH' statement to arrange the data in required format. Group the data using 'GROUP' function based on product ID. Using 'COUNT' function to get the count of feedback and 'AVG' function to get average feedback score for each individual product using 'FOREACH' expression. Sort the results in descending order using 'DESC' key word and limit to 20 records using 'LIMIT' function. At the end, dump the results using 'DUMP' function.


products_count_of_feedback.pig (~/training_materials/analyst/exercises/application_project) - gedit
File Edit View Search Tools Documents Help
Open
Save
Undo
products_count_of_feedback.pig

```
--loading the ratings data of 2012
data2012 = LOAD 'ratings_2012.txt' AS (date:chararray, user_id:int, prod_id:int, feedback_score:int, comments:chararray);

--loading the ratings data of 2013
data2013 = LOAD 'ratings_2013.txt' AS (date:chararray, user_id:int, prod_id:int, feedback_score:int, comments:chararray);

--combining the data using the UNION operator
combine_data = UNION data2012, data2013;

--ditinct the data to remove any duplicates if exist
distinct_data = DISTINCT combine_data;

--arranging the data in the order
filter_data = FOREACH distinct_data GENERATE date, user_id, prod_id, feedback_score, comments;

--grouping the data based on the product ID
grouping = GROUP filter_data BY prod_id;

--getting the total feedback score and average feedback score for each group (product) we have created
get_total = FOREACH grouping GENERATE group, COUNT(filter_data.feedback_score) AS feedback_count, AVG(filter_data.feedback_score) AS average_feedback_score;

--ordering the data in descending order
order_data = ORDER get_total BY feedback_count DESC;

--limiting the data to top 10
limit_data = LIMIT order_data 20;

--dumping the results
DUMP limit_data;
```

Figure 11: Code for Top 20 Products with highest feedback Count and their Individual Average Feedback Score

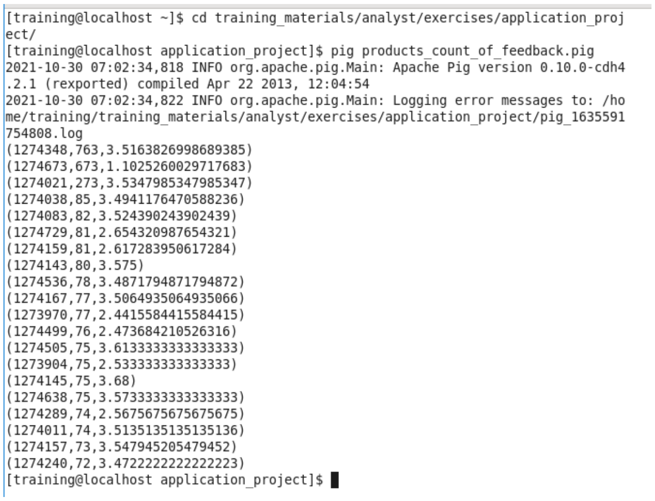
Below figure shows the Pig Latin Script for retrieving the satisfaction rate of the top 20 customers who have placed the most orders. In the script (customer_count_of_feedback.pig), we load the data from both the data files using 'LOAD' Function. Integrated the data into a single table using 'UNION' operator. 'DISTINCT' the data to remove duplicates. Using the 'FOREACH' statement to arrange the data in required format. Group the data using 'GROUP' function based on User ID. Using 'COUNT' function to get the count of feedback count and 'AVG' function to get average feedback score for each individual product using 'FOREACH' expression. Sort the results in descending order using 'DESC' key word and limit to 20 records using 'LIMIT' function. At the end, dump the results using 'DUMP' function.


customer_count_of_feedback.pig (~/training_materials/analyst/exercises/application_project) - gedit
File Edit View Search Tools Documents Help
Open
Save
Undo
customer_count_of_feedback.pig

```
--loading the ratings data of 2012
data2012 = LOAD 'ratings_2012.txt' AS (date:chararray, user_id:int, prod_id:int, feedback_score:int, comments:chararray);

--loading the ratings data of 2013
data2013 = LOAD 'ratings_2013.txt' AS (date:chararray, user_id:int, prod_id:int, feedback_score:int, comments:chararray);

--combining the data using the UNION operator
combine_data = UNION data2012, data2013;

--ditinct the data to remove any duplicates if exist
distinct_data = DISTINCT combine_data;

--arranging the data in the order
filter_data = FOREACH distinct_data GENERATE date, user_id, prod_id, feedback_score, comments;

--grouping the data based on the product ID
grouping = GROUP filter_data BY user_id;

--getting the total feedback score and average feedback score for each group (product) we have created
get_total = FOREACH grouping GENERATE group, COUNT(filter_data.prod_id) AS products_count, AVG(filter_data.feedback_score) AS average_feedback_score;

--ordering the data in descending order
order_data = ORDER get_total BY products_count DESC;

--limiting the data to top 10
limit_data = LIMIT order_data 20;

--dumping the results
DUMP limit_data;
```

Figure 12: Code for Top 20 Customers who have highest feedback Count and their Average Feedback Score

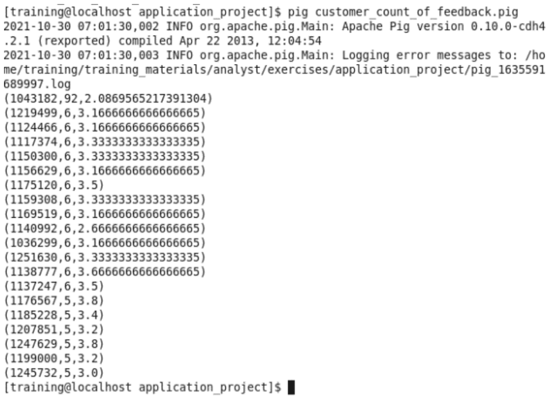
The below figure is the output we received for the pig script we used for retrieving the top 20 products with highest feedback count and their individual average feedback score. We got the results for average feedback score and total feedback count for individual products. As it can be seen from the figure, product 1274348 has the highest feedback count with 763 feedbacks followed by 1274673 with 673, 1274021 with 273 and other. In other words, these products can be declared as most selling products for the company. Products 1274348 and 1274021 have an average feedback score of 3.52 and 3.54 (rounded to two decimals) respectively, which describes that these products satisfied the customers. Although, product with second highest feedback count has an average feedback score of 1.11 (rounded to two decimals), which tells that this product didn't satisfy the customer. 1274729, 1274159, 1273970, 1274499, 1273904, 1274289 are some more products which require company's attention. Moreover, more than 50% of the products has an average feedback score between 3.4 and 3.7, which mean that customers are happy after consuming the product. Company should pay attention towards the products with low average feedback score and take measure to make the products more satisfying and popular and advertise the products with high average feedback score and make sure that these products have constant satisfaction and popularity to improve their sales.



```
[training@localhost ~]$ cd training_materials/analyst/exercises/application_proj
ect/
[training@localhost application_project]$ pig products_count_of_feedback.pig
2021-10-30 07:02:34,818 INFO org.apache.pig.Main: Apache Pig version 0.10.0-cdh4
.2.1 (rexported) compiled Apr 22 2013, 12:04:54
2021-10-30 07:02:34,822 INFO org.apache.pig.Main: Logging error messages to: /ho
me/training/training_materials/analyst/exercises/application_project/pig_1635591
754808.log
(1274348,763,3.5163826998689385)
(1274673,673,1.1025260029717683)
(1274021,273,3.5347985347985347)
(1274038,85,3.4941176470588236)
(1274083,82,3.524390243902439)
(1274729,81,2.654320987654321)
(1274159,81,2.617283950617284)
(1274143,80,3.575)
(1274536,78,3.4871794871794872)
(1274167,77,3.5064935064935066)
(1273970,77,2.4415584415584415)
(1274499,76,2.473684210526316)
(1274505,75,3.6133333333333333)
(1273904,75,2.533333333333333)
(1274145,75,3.68)
(1274638,75,3.5733333333333333)
(1274289,74,2.5675675675675675)
(1274011,74,3.5135135135135136)
(1274157,73,3.547945205479452)
(1274240,72,3.4722222222222223)
[training@localhost application_project]$ █
```

Figure 13: Output for Top 20 Products with highest feedback Count and their Individual Average Feedback Score

The below figure is the output we received for the pig script we used for retrieving the top 20 customers with highest feedback count and their individual average feedback score. We got the results for average feedback score and total feedback count for individual customers. As it can be seen from the figure, customer 1043182 has the highest feedback count with 92 feedbacks, i.e., this is the customer with highest overall orders [2012-2013]. Majority of the customers have a similar feedback count i.e., 6 feedbacks. Moreover, more than 80% of the customers have an average feedback score between 3.0 and 3.8, which mean that these customers are satisfied with the products. Although, the customer with highest feedback count (92) has the lowest average feedback score (2.09, rounded to two decimals), the company should analyze the comments provided by the customer and contact the customer to resolve the issue of the customer. In addition to, customer 1140992 had an average feedback score of 2.7 which can be said as average or meets standard feedback (>60%), this customer's issue should also be resolved.



```
[training@localhost application_project]$ pig customer_count_of_feedback.pig
2021-10-30 07:01:30,002 INFO org.apache.pig.Main: Apache Pig version 0.10.0-cdh4
.2.1 (rexported) compiled Apr 22 2013, 12:04:54
2021-10-30 07:01:30,003 INFO org.apache.pig.Main: Logging error messages to: /ho
me/training/training_materials/analyst/exercises/application_project/pig_1635591
689997.log
(1043182,92,2.0869565217391304)
(1219499,6,3.1666666666666665)
(1124466,6,3.1666666666666665)
(1117374,6,3.3333333333333335)
(1150300,6,3.3333333333333335)
(1156629,6,3.1666666666666665)
(1175120,6,3.5)
(1159308,6,3.3333333333333335)
(1169519,6,3.1666666666666665)
(1140992,6,2.6666666666666665)
(1036299,6,3.1666666666666665)
(1251630,6,3.3333333333333335)
(1138777,6,3.6666666666666665)
(1137247,6,3.5)
(1176567,5,3.8)
(1185228,5,3.4)
(1207851,5,3.2)
(1247629,5,3.8)
(1199000,5,3.2)
(1245732,5,3.0)
[training@localhost application_project]$ █
```

Figure 14: Output for Top 20 Customers who have highest feedback Count and their Average Feedback Score

Second Additional Goal

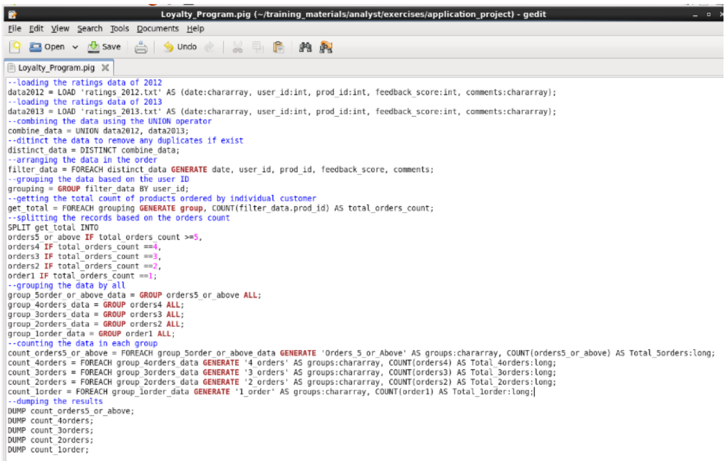
The second addition goal is to identify how many high frequent buyers the company has in the dataset provided. Using the analyzed the data the company's stakeholder can determine the affect of marketing strategies, how effective the company is in sustaining its customers and may organize a thanksgiving program/loyalty program to thank their loyal customers.

Definition and Solving Logic

To determine the frequent buyers of the company we'll load the data for 2012 and 2013 from the provided data files and integrate all the data into a single table. We don't know whether there are any duplicate records in the data because we haven't performed data cleaning, therefore we'll distinct the data to eliminate duplicates if any exist. Followed by, we will arrange the data into the required format and group the data by Customer ID. Calculate the count of products (orders) for individual group (customer) we've generated, split the result into 5 categories based on the order count of 1, 2, 3, 4, and 5 or above. Before we jump onto the next step, we need to save the customer data based on their order count, so we will save the data. Now again we will group all the data in each category and find the count of customers in individual group. At the end, we will display the results.

Solution

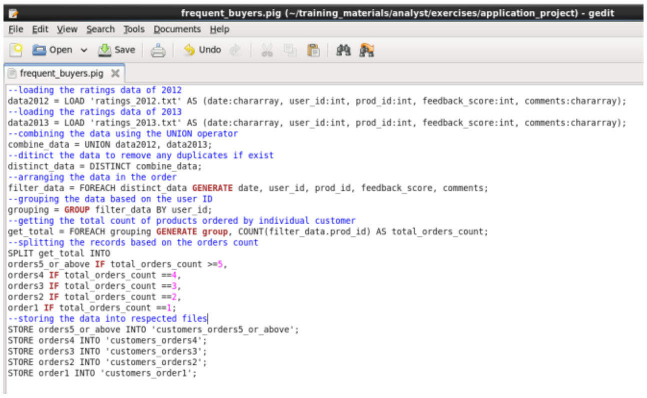
Below figure shows the Pig Latin Script for retrieving the frequent buyer's data. In the script (Loyalty_program.pig), we load the data from both the data files using 'LOAD' Function. Integrated the data into a single dataset using 'UNION' operator. 'DISTINCT' the data to remove duplicates. Using the 'FOREACH' statement to arrange the data in required format. Group the data using 'GROUP' function based on User ID. Using 'COUNT' function to get the count of products (orders) for each individual customer using 'FOREACH' expression. Split the data using 'SPLIT' function based on the order count, again group the data based on the categorized data using 'GROUP ALL' function. Again using 'COUNT' function to get the count of customers for each individual group using 'FOREACH' expression and repeat it for each individual group and display the results individually using 'DUMP' function.


Loyalty_Program.pig (~/training_materials/analyst/exercises/application_project) - gedit
File Edit View Search Tools Documents Help
Open
Save
Undo
Loyalty_Program.pig

```
--loading the ratings data of 2012
data2012 = LOAD 'ratings_2012.txt' AS (date:chararray, user_id:int, prod_id:int, feedback_score:int, comments:chararray);
--loading the ratings data of 2013
data2013 = LOAD 'ratings_2013.txt' AS (date:chararray, user_id:int, prod_id:int, feedback_score:int, comments:chararray);
--combining the data using the UNION operator
combine_data = UNION data2012, data2013;
--ditinct the data to remove any duplicates if exist
distinct_data = DISTINCT combine_data;
--arranging the data in the order
filter_data = FOREACH distinct_data GENERATE date, user_id, prod_id, feedback_score, comments;
--grouping the data based on the user ID
grouping = GROUP filter_data BY user_id;
--getting the total count of products ordered by individual customer
get_total = FOREACH grouping GENERATE group, COUNT(filter_data.prod_id) AS total_orders_count;
--splitting the records based on the orders count
SPLIT get_total INTO
orders5_or_above IF total_orders_count >=5,
orders4 IF total_orders_count ==4,
orders3 IF total_orders_count ==3,
orders2 IF total_orders_count ==2,
order1 IF total_orders_count ==1;
--grouping the data by all
group_5order_or_above_data = GROUP orders5_or_above ALL;
group_4orders_data = GROUP orders4 ALL;
group_3orders_data = GROUP orders3 ALL;
group_2orders_data = GROUP orders2 ALL;
group_1order_data = GROUP order1 ALL;
--counting the data in each group
count_orders5_or_above = FOREACH group_5order_or_above_data GENERATE 'Orders_5_or_Above' AS groups:chararray, COUNT(orders5_or_above) AS Total_5orders:long;
count_4orders = FOREACH group_4orders_data GENERATE '4_orders' AS groups:chararray, COUNT(orders4) AS Total_4orders:long;
count_3orders = FOREACH group_3orders_data GENERATE '3_orders' AS groups:chararray, COUNT(orders3) AS Total_3orders:long;
count_2orders = FOREACH group_2orders_data GENERATE '2_orders' AS groups:chararray, COUNT(orders2) AS Total_2orders:long;
count_1order = FOREACH group_1order_data GENERATE '1_order' AS groups:chararray, COUNT(order1) AS Total_1order:long;
--dumping the results
DUMP count_orders5_or_above;
DUMP count_4orders;
DUMP count_3orders;
DUMP count_2orders;
DUMP count_1order;
```

Figure 15: Code for Finding the Frequent Buyers of the Company

Below figure displays the Pig Latin Script for saving the frequent buyer's data based on the orders count. We have slightly updated the recent Pig Script (Loyalty_program.pig) and named it (frequent_buyers.pig), we deleted all the code below the splitting part and stored the data based on the order count using 'STORE' function. These customer files aid the company in identifying the customers customer who are loyal with 3 or more orders and customers who haven't returned i.e., with 1 order.

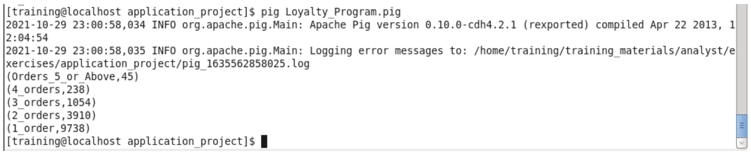

frequent_buyers.pig (~/training_materials/analyst/exercises/application_project) - gedit
File Edit View Search Tools Documents Help
Open
Save
Undo
frequent_buyers.pig

```
--loading the ratings data of 2012
data2012 = LOAD 'ratings_2012.txt' AS (date:chararray, user_id:int, prod_id:int, feedback_score:int, comments:chararray);
--loading the ratings data of 2013
data2013 = LOAD 'ratings_2013.txt' AS (date:chararray, user_id:int, prod_id:int, feedback_score:int, comments:chararray);
--combining the data using the UNION operator
combine_data = UNION data2012, data2013;
--ditinct the data to remove any duplicates if exist
distinct_data = DISTINCT combine_data;
--arranging the data in the order
filter_data = FOREACH distinct_data GENERATE date, user_id, prod_id, feedback_score, comments;
--grouping the data based on the user ID
grouping = GROUP filter_data BY user_id;
--getting the total count of products ordered by individual customer
get_total = FOREACH grouping GENERATE group, COUNT(filter_data.prod_id) AS total_orders_count;
--splitting the records based on the orders count
SPLIT get_total INTO
orders5_or_above IF total_orders_count >=5,
orders4 IF total_orders_count ==4,
orders3 IF total_orders_count ==3,
orders2 IF total_orders_count ==2,
order1 IF total_orders_count ==1;
--storing the data into respected files
STORE orders5_or_above INTO 'customers_orders5_or_above';
STORE orders4 INTO 'customers_orders4';
STORE orders3 INTO 'customers_orders3';
STORE orders2 INTO 'customers_orders2';
STORE order1 INTO 'customers_order1';
```

Figure 16: Code for Storing the Data of Frequent Buyers

The below figure is the output we received for the pig script we used for retrieving the frequent buyers. We got the customers count based on the orders they have placed with the company. As the figure displays, there are 45 customers who have 5 or more orders, followed by 238 customers with 4 orders, 1054 customers with 3 orders, 3910 customers with 2 orders and 9738 customers with 1 order.

When we plot a pie graph with the following results, 65% of the customers have placed just 1 order, in other words they haven't return to the company after they bought the product. Very few percentages (0.6% and 1.6%) of customers have purchased the company's goods four times and five or more times.



```
[training@localhost application_project]$ pig Loyalty_Program.pig
2021-10-29 23:00:58,034 INFO org.apache.pig.Main: Apache Pig version 0.10.0-cdh4.2.1 (rexported) compiled Apr 22 2013, 1
2:04:54
2021-10-29 23:00:58,035 INFO org.apache.pig.Main: Logging error messages to: /home/training/training_materials/analyst/e
xercises/application_project/pig_1635562858025.log
(Orders_5_or_Above,45)
(4_orders,238)
(3_orders,1054)
(2_orders,3910)
(1_order,9738)
[training@localhost application_project]$ █
```

Figure 17: Output for Frequent Buyers of the Company

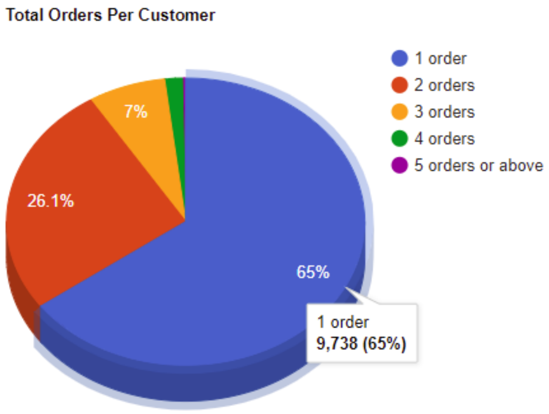
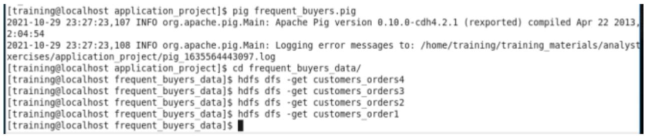


Figure 18: Pie Chart Showing Customers Based on Order Count



```
[training@localhost application_project]$ pig frequent_buyers.pig
2021-10-29 23:27:23,107 INFO org.apache.pig.Main: Apache Pig version 0.10.0-cdh4.2.1 (rexported) compiled Apr 22 2013,
2:04:54
2021-10-29 23:27:23,108 INFO org.apache.pig.Main: Logging error messages to: /home/training/training_materials/analyst
xercises/application_project/pig_1635564443097.log
[training@localhost application_project]$ cd frequent_buyers_data/
[training@localhost frequent_buyers_data]$ hdfs dfs -get customers_orders4
[training@localhost frequent_buyers_data]$ hdfs dfs -get customers_orders3
[training@localhost frequent_buyers_data]$ hdfs dfs -get customers_orders2
[training@localhost frequent_buyers_data]$ hdfs dfs -get customers_order1
[training@localhost frequent_buyers_data]$ █
```

Figure 19: Output for Storing the Data of Frequent Buyers

As the below figure illustrates, we have stored all the customer data by dividing them based on their order count. The company could use this data to identify the customers with least order count and start promoting the products via emails. Moreover, company can introduce some offers for customers based on their order count or start some loyalty program to attract customers.

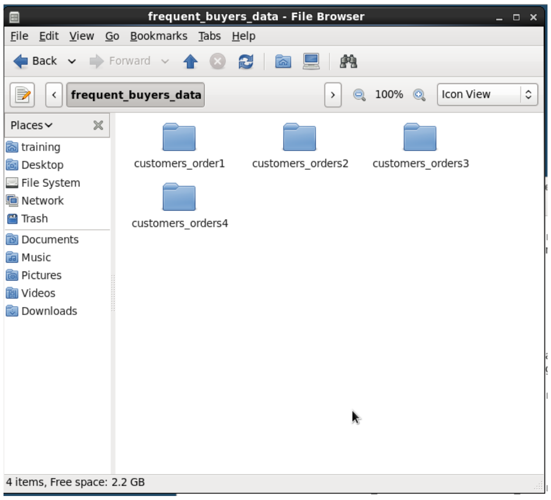


Figure 20: Directory Containing all the Data Related to Frequent Buyers

Conclusion

The company's overall performance level tends to improve between 2012-2013. The customer satisfaction remains constant for the period of 2 years. It was found that product 1274348, 1274673 and 1274021 generated highest sales for the company and customers were satisfied with product 1274348 and 127402. Despite being one of the most selling products, 1274673 has the lowest average feedback score (1.11) among the top 20 selling products. Moreover, 65% of customers haven't returned to the company after buying their first product. Most of the frequent buyers of the company were satisfied with the goods with an average feedback score between 3.0 and 3.8 excluding customer 1043182 with highest orders with an average feedback score of 2.09. further, the graphs provided aid to the company with comprehending their data further, allowing access to analysis with ease.

The choice to use a data-driven strategy might result in improved decision-making and operating excellence. However, we have suggested certain strategies to help company analyze their sales, customer satisfaction and frequent buyers to promote industry growth and increase their sales and customers.

References

1) Gates, A 2011, *Programming Pig,* O'Reilly Media, O'Reilly.

2) Sammer, E 2012, *Hadoop Operations,* O'Reilly Media, O'Reilly.

3) Tsai, P 2021, 'Introduction to Apache Pig', Introduction to Data Science, Learning material via canvas, Swinburne University of Technology, 1 July. [9 October 2021]

4) White, T 2015, *Hadoop: The Definitive Guide, 4th Edition*, O'Reilly Media, O'Reilly.