**PROJECT REPORT**

**DUALCORE’S SUPPLIER ENGAGEMENT AND CUSTOMER CONCERN ANALYSIS**

**IS8034 – Big Data Integration**

**Team 6**

**Tanvi Kini**

**Rashi Sharma**

**Rohit Verma**

**Customer Decision Support**

**Business Question: What can Dualcore possibly do to increase sales in terms of pre-sale and post-sale customer service? Is it possible to make their e-commerce website more customer friendly to reduce the amount of time a customer spends researching the product they want and make decisions? Our goal is to try and identify improvements in the customers decision making process by providing relevant information and finding out supplier engagement on the website**.

1. **Understanding the Importance of Supplier Engagement**

The Q&A section of an e-commerce website is one of the most important sources of information a customer uses to research a product before buying it and even after buying it to resolve concerns and complaints. A supplier’s engagement in this section is therefore very important, not only to provide accurate information about their products but also to pacify disgruntled customers and curb misinformation when other customers try answering questions and unintentionally mislead the asker.

We are attempting to find supplier engagement by simply finding out the percentage of questions answered by suppliers. To do this we are using common phrases used by suppliers, which a customer wouldn’t use if they were to answer questions. We are also analyzing the questions asked by customers to find out what the common concerns about products in a certain category are. Based on our findings, we would then make recommendations about what can possibly be added to the description section of products, so a customer doesn’t have to ask the question and wait for an answer. This time between a customer asking a question and receiving the right answer is crucial because they can change their mind in this time and decide not to buy the product at all or buy it elsewhere.

1. **Data Analysis Tools**

We used an Amazon dataset called qa\_Electronics. We initially had a very large dataset with ~3 million rows and about 7 columns. We used a Python script to clean this dataset to keep only the rows and columns relevant to our business use case and analysis. The cleansed dataset was a .CSV file that had only about ~63000 rows and 3 columns – ‘question’, ‘answer’ and ‘category’. These questions and answers were textual in nature.

To then ingest this data, we used the Hue UI where we could import our file directly and create a table called qna. We also used a Spark enabled virtual machine cluster to run our queries and perform all the text processing required to perform the kind of analysis we wanted. Apache Spark is a powerful unified analytics engine for large-scale distributed data processing and machine learning. To run applications distributed across a cluster, Spark requires a cluster manager. One of the managers supported by Cloudera is YARN short for (Yet another Resource Negotiator). This is the resource manager we used while running our queries. When run on YARN, Spark application processes are managed by the YARN ResourceManager and NodeManager roles. Since there were a large number of records, this kind of processing would have been difficult and time consuming on a single system with less computing power. We used the Hive command line interface as well as the Hue editor to write and run our queries.

1. **Data Sources**

We were provided with a database of an e-commerce company called Dualcore that sells electronics and electronic accessories. The dataset contained information such as the list of products, their sales data and reviews for each of the products written by customers. We combined this data with an Amazon dataset that contained qna information for Electronic products. We categorized each of the rows into categories and matched them with categories of products in Dualcore.

Our dataset didn’t have a relationship between the supplier and the products i.e. there was no direct way to tell which supplier supplied which product. We therefore decided to perform our analysis at a category level to check which had the most engagement. We had to identify the answers given by suppliers in the external ‘qna’ dataset. To do this, we identified commonly used supplier phrases to find answers that would have been given by the supplier. We realized there is a possibility that a supplier has answered questions but has not used any of the phrases we identified. However, our phrases included ones that a supplier should ideally use, firstly to identify their self as a Supplier to the customer and secondly because they are courteous phrases which should be used as best practice while addressing a valued customer. Answers not containing these phrases, we decided, were as good as answers given by anyone else.

1. **Data Integration Process**

Schema Alignment:

This step included identifying the grain, selecting the attributes we need and data source to use. Since the problem we were trying to solve involved finding the suppliers engagement with the customers for each category, the grain of our schema becomes a category which encompasses several products. We came up with the following

Mediated Schema:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Product | Qna | Order\_Count | Supplier\_  Engagement | Popular\_  Brands | Average\_  Statistics | Example | Comments |
| Prod\_Category | name  (grouped) |  |  |  |  |  | ‘TV’ | Joins Product\_Line &  Qna tables. |
| Reply\_Count |  | Qna.Answer  (calculated and filtered) |  |  |  |  | 215 | Count of supplier answers from qna |
| Total\_question |  | Qna.question  (calculated and filtered) |  |  |  |  | 1225 | Count of questions from qna |
| Engagement |  |  |  | Qna.engagement  (calculated) |  |  | 2.25 | (Reply\_count/  Total\_question)  \*100 |
| PopularBrand |  |  |  |  | Product\_  Category  (calculated) |  | “Xroger” | Max(brand) |
| AvgCost |  |  |  |  |  | Product\_Group.  Price  (calculated) | 200.649 | Convert to dollars |
| AvgPrice |  |  |  |  |  | Product\_Group.  Price  (calculated) | 473.199 | Convert to dollars |
| AvgProfit |  |  |  |  |  | [Price-Cost]  Calculated | 14.465 | Convert to dollars |
| Num\_orders |  |  | Count of prod\_id from  Product\_category  calculated |  |  |  | 42250 | Number of orders per category |

Issues encountered

The first issue we encountered was determining the grain. Our initial idea was to perform this analysis on a Supplier level where we could group different products by the ID of the supplier who supplied them and then find the engagement level as a metric for each supplier to gauge how active they were in the Q&A section. Since, the supplier IDs were not linked to the products directly, this wasn’t possible. Also, the product IDs in our external dataset were completely different from those of Dualcore’s since they were from different companies. We there resorted to performing our analysis at a category level.

Record Linkage:

Record linkage is the task of finding records in a data set that refer to the same identity across different data sources. Record linkage is necessary when joining data sets based on entities that may or may not share a common identifier. Since most of our data was free form text, it was highly unlikely that we would have the exact same data in separate rows. Looking at this issue from a broader perspective, we could have possibly identified which questions were repeating for a given category. However, this would dilute our numbers when it came to analyzing supplier engagement as we needed actual figures for the total count of questions asked and answers given by suppliers.

Data Fusion:

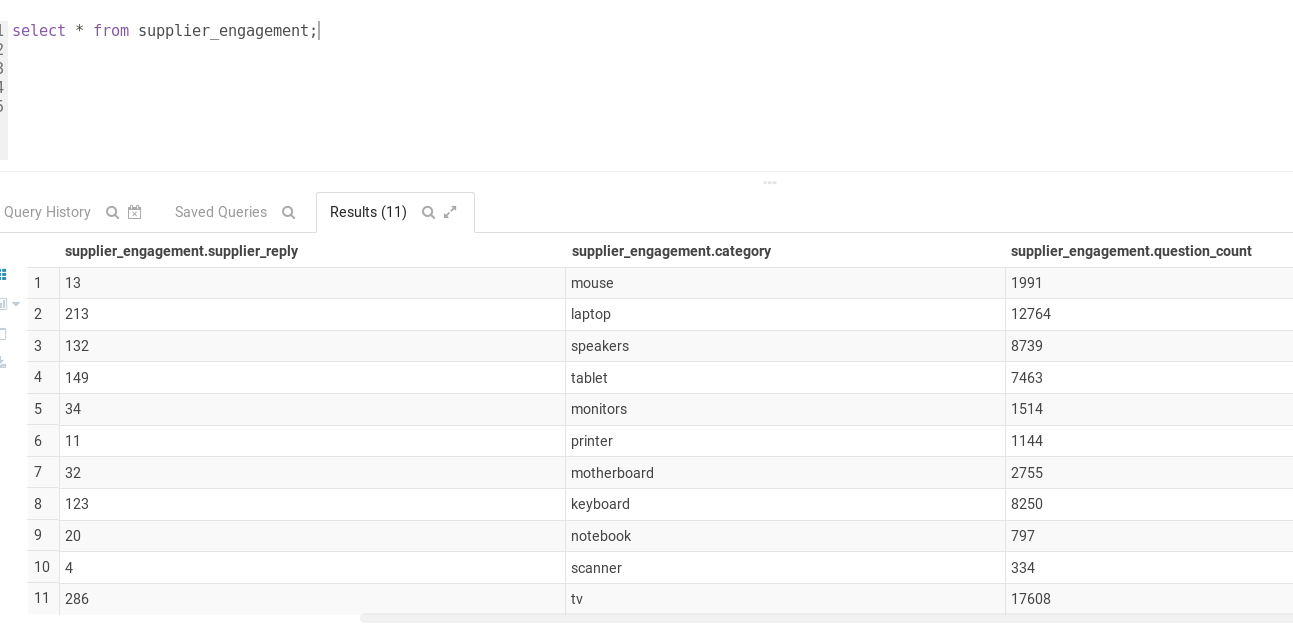
This was the last step of data integration. In our case, his phase involved formatting the source attributes so that they match the fields in the mediated schema. Since we couldn’t match the product IDs from the Dualcore dataset to the external Amazon Q&A dataset, we decided the best way forward was to aggregate this data based on category. We identified categories based on the ‘name’ field in the PRODUCT table. For Example, a product with the name “Cannon Scanner” would be categorized as a ‘scanner’ and placed in a new table called product\_groups. This new table was then joined with the external qna table on category.

Issues Encountered

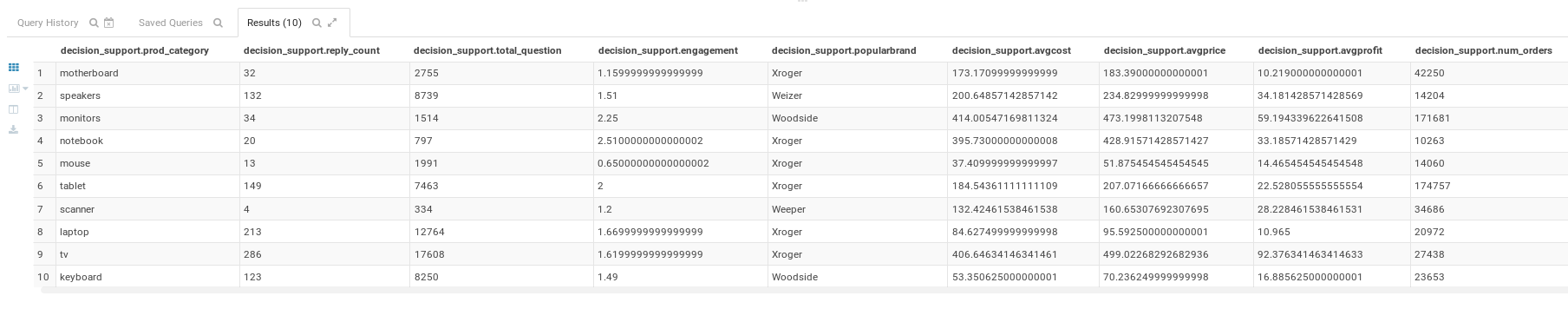
We faced an issue where only some of the rows in the product table of the Dualcore dataset were getting categorized while the others were just showing up under the null category. We overcame by changing our query to be case sensitive. For example, it was now checking to identify the keywords ‘Laptop’ and ‘laptop’ from the product name to categorize it as a laptop.

1. Business Insights

Analyzing engagement



We counted the total number of questions asked in each category and compared it with the number of answers containing the list of supplier best-practice phrases we identified. We then created a table called supplier\_engagement where this information was stored. The screenshot below shows the contents of the supplier\_engagement table.



Now that we had the number we needed to calculate the supplier engagement we used the formula:

**Supplier Engagement = (Answers given by suppliers/ Total question count) \* 100.**

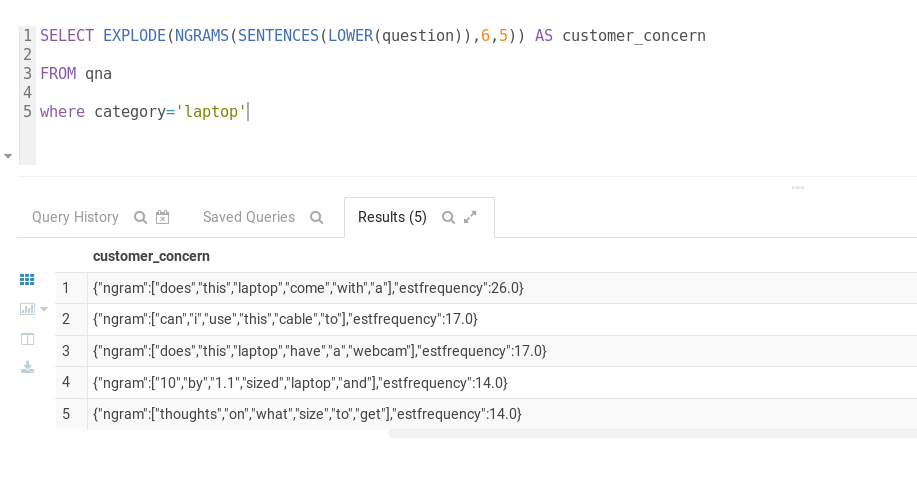
This gave us the percentage of questions answered by suppliers. We created a table called Decision Support which contained all this information. As we can see, the last column indicating engagement shows very low percentage of supplier engagement. This gives us a lot of room for improvement in the way customer support is handled. We recommend that the supplier engagement should ideally be between 40 – 50 % so that customers can get first-hand information about the product directly from the supplier.

* Analyzing customer concerns

We also tried analyzing the kind of questions that were being asked by the customers by using HiveQL’s NGRAMS function that allows us to view words that frequently occur together. We used some trial and error to figure at what value of N we could find some relevant insight.

Here are some of the results we came up with:

1. For Laptops:



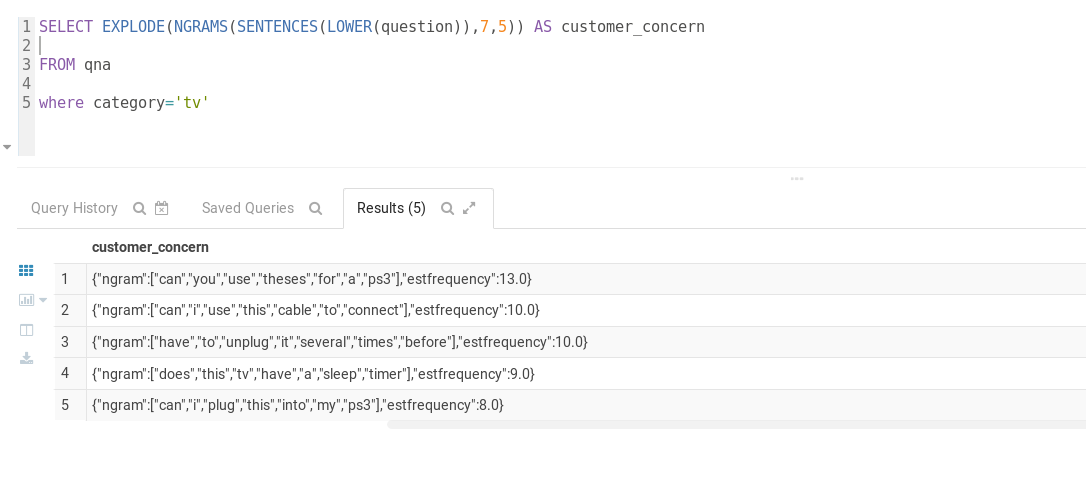
The screenshot above shows the results we got for the category ‘Laptops’. It appears some of the most commonly asked questions include customers asking about whether a certain peripheral accessory is included with the purchase of the laptop itself. Another result indicates that customers wonder if laptops come with an inbuilt webcam. Some of the results suggest that customers need guidance deciding on the screen size of the laptop they want to purchase.

Our suggestions:

Standardize the description of laptops to include the following information:

* The peripherals that come with the product.
* Whether or not the laptop has a webcam.
* Possibly a video (or a link to a video) showing a comparison between different laptop screen sizes and recommended sizes based on type of user.

1. For TVs:



The screenshot above shows the results we got for the category ‘TV’. Some of the most commonly asked questions include whether or not you can use the TV with a PS3 device and if the TV has a sleep timer.

Our suggestions:

Standardize the description of TVs to include the following information:

* List of the different gaming equipment like PS3s and XBOX that are compatible with the TV.
* If the TV has a sleep timer.

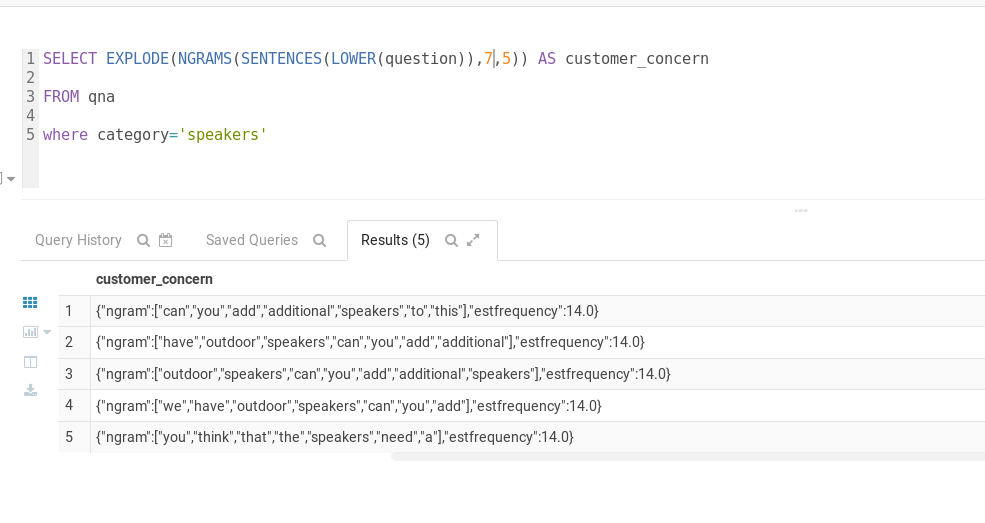
1. For Keyboards:

The screenshot above shows the results we got for the category ‘Keyboard’. The main concern appears to be if the Keyboard has backlighting.

Our suggestion:

Standardize the description of Keyboards to include the presence or absence of Backlighting.

1. For speakers:



The screenshot above shows the results we got for the category ‘Speakers’. Many customers seem to be asking if they can add additional speakers to the product and use it outdoors.

Our suggestion:

* Standardize the description of all speakers to include whether they are capable of being combined with other speakers and type or brand restrictions on the kind of combination that is possible, if any.
* Include a recommendation for if a particular set of speakers should be used indoors or outdoors and the effect on sound quality in each case.

1. **Future Steps**

In the future, such analysis can be done using Machine Learning algorithms which can efficiently identify the tone in which a customer might be asking a question and also make suggestions to the Supplier about best practices to keep in mind while replying to customers. In addition, it is also possible to automate replying to customers using Artificial Intelligence.

Having this analysis on a supplier level could also be done if we could collect more information that would bridge the gap between suppliers and customers. This way we could pin-point individual suppliers and draw parallels between their level of engagement and their rating. An easier way to track supplier answers would be to tag all answers given from Supplier Accounts as a ‘Supplier Answer’ making it much easier to find out their level of engagement.

1. **End Notes**

Sources:

1. Amazon Dataset: <http://jmcauley.ucsd.edu/data/amazon/qa/>
2. Lecture Lab notes provided by Prof. Andrew Harrison while teaching IS-8034 (Big Data Integration) as part of course curriculum helped us write queries for the analysis we wanted to perform.
3. Description for Hive: <http://hadooptutorial.info/hive-cli-commands/>
4. Description for YARN:

<https://www.cloudera.com/documentation/enterprise/5-6-x/topics/spark.html>

1. Description for Spark:

<https://mapr.com/products/apache-spark/>

1. Definition of Record Linkage:

<https://en.wikipedia.org/wiki/Record_linkage>

1. HIVEQL Code:

* **Creating the SUPPLIER\_ENGAGEMENT table**:

CREATE TABLE supplier\_engagement AS

select temp.supplier\_reply , temp. category, count(question) as Question\_count from (SELECT COUNT(\*) as supplier\_reply, category

FROM qna where answer LIKE "%Thank you%" OR answer LIKE "%Thanks%" OR answer LIKE "% our product%" OR answer LIKE "%we are sorry%" OR answer LIKE "%dear customer"

OR answer LIKE "%contact us%" OR answer LIKE "%call us%" OR answer LIKE "%we recommend%" OR answer LIKE "%valued customer%" group by category) temp

inner join  qna

on temp.category=qna.category

group by supplier\_reply, temp. category

* **Creating the PRODUCT\_GROUPS table:**

CREATE TABLE product\_groups AS

SELECT Categorize.prod\_id, Categorize.prod\_category, Categorize.brand, Categorize.price, Categorize.cost

FROM (SELECT prod\_id, "tablet" AS prod\_category, brand, price, cost

FROM products

WHERE name LIKE "%tablet%" OR name LIKE "%Tablet%"

UNION ALL

SELECT prod\_id, "laptop" AS prod\_category,brand, price, cost

FROM products

WHERE name LIKE "%Laptop%" OR name LIKE "%laptop%"

UNION ALL

SELECT prod\_id, "keyboard" AS prod\_category, brand, price, cost

FROM products

WHERE name LIKE "%Keyboard%" OR name LIKE "%keyboard%"

UNION ALL

SELECT prod\_id, "monitors" AS prod\_category, brand, price, cost

FROM products

WHERE name LIKE "%monitor%" OR name LIKE "%Monitor%" OR name LIKE "%display%" OR name LIKE "%Display%"

UNION ALL

SELECT prod\_id, "motherboard" AS prod\_category, brand, price, cost

FROM products

WHERE name LIKE "%motherboard%" OR name LIKE "%Motherboard%"

UNION ALL

SELECT prod\_id, "mouse" AS prod\_category, brand, price, cost

FROM products

WHERE name LIKE "%Mouse%" OR name LIKE "%mouse%"

UNION ALL

SELECT prod\_id, "tv" AS prod\_category, brand, price, cost

FROM products

WHERE name LIKE "%tv%" OR name LIKE "%TV%"

UNION ALL

SELECT prod\_id, "scanner" AS prod\_category, brand, price, cost

FROM products

WHERE name LIKE "%scanner%" OR name LIKE "%Scanner%"

UNION ALL

SELECT prod\_id, "notebook" AS prod\_category, brand, price, cost

FROM products

WHERE name LIKE "%notebook%" OR name LIKE "%Notebook%"

UNION ALL

SELECT prod\_id, "speakers" AS prod\_category, brand, price, cost

FROM products

WHERE name LIKE "%speaker%" OR name LIKE "%Speaker%") Categorize;

* **Creating the ORDER\_COUNT table:**

CREATE TABLE order\_count AS

SELECT prod\_category,

 COALESCE(COUNT(d.prod\_id),0) AS num\_orders

FROM product\_group p

inner JOIN order\_details d

ON p.prod\_id = d.prod\_id

GROUP BY p.prod\_category

* **Creating the POPULAR\_BRANDS table:**

Create table  popular\_brands as

select prod\_category, max(brand) As PopularBrand

from products\_group

group by prod\_category

* **Creating the AVERAGE\_STATISTICS table:**

Create table average\_statistics as

select avg(price) as avgprice ,avg(cost) as avgcost, avg(profit) as avgprofit, temp.prod\_category from (select

ROUND(price/100,2) AS price,

 ROUND(cost/100,2) AS cost,

 ROUND((price-cost)/100,2) AS profit,

 prod\_category

from product\_groups)temp

group by prod\_category

* **Examples of NGRAMS Queries:**

SELECT EXPLODE(NGRAMS(SENTENCES(LOWER(question)),7,5)) AS customer\_concern

FROM qna

where category='laptop'

SELECT EXPLODE(NGRAMS(SENTENCES(LOWER(question)),7,5)) AS customer\_concern

FROM qna

where category='keyboard'