Real-Time Digital Marketing Analytics

Project 4

# Group 10

# LE, TRUNG

# [DESWAL, SUPRITA](https://bruinlearn.ucla.edu/courses/150644/users/75371)

# [RAVISHANKAR, NIDHI](https://bruinlearn.ucla.edu/courses/150644/users/74058)

[VERIAH, MICHELLE](https://bruinlearn.ucla.edu/courses/150644/users/74123)

[SRIVASTAV, PRAKHAR](https://bruinlearn.ucla.edu/courses/150647/users/76287)

[PIDAPARTY, SARVARI](https://bruinlearn.ucla.edu/courses/150647/users/73919)

[XIN, JIA RUI](https://bruinlearn.ucla.edu/courses/150647/users/74552)

[WANG, XINHUI](https://bruinlearn.ucla.edu/courses/150647/users/40359)

Table of Contents

[**Executive Summary**](#_580r92uvyt0z) **3**

[**Project Statement**](#_6nizejb3kw41) **3**

[**Data Literacy**](#_tlon6vgvg39j) **4**

[**Data Dictionary**](#_9fmk0rqhbo54) **8**

[**Data Processing and Architecture**](#_ippnmfbpmy52) **10**

[**KPIs & Insights**](#_6l50qkmve883) **13**

[**Project Challenges**](#_f18v87ip05j0) **26**

[**Appendix**](#_kvy2jdv1qobp) **27**

# **Executive Summary**

Our project aimed to enhance our services for clients by analyzing their performance through the identification of 8 key performance indicators (KPIs). We categorized our clients into three categories based on performance, analyzed their costs with us, conversion rates, and offer strategies. Using this information, we were able to optimize offer-creation strategies and improve our services for better client outcomes.

However, we encountered challenges that require additional support from the company, such as the need for more comprehensive data, including multi-city data, to scale up our findings and present better business solutions.

Based on our analysis, we recommend that the company does the following:

* To enhance our services for better client outcomes, we recommend sharpening our service's segment-targeting ability.
* We recommend optimizing offer-creation strategies with our clients by considering factors such as the frequency at which users want ads to be shown and the discount rates offered. We also want to regulate bid values with clients.

# **Project Statement**

The Real-Time Digital Marketing application powered by SingleStore utilizes user behavior and real-time location data to serve targeted ads. Using millions of subscribers' location, purchase, and request history data, companies with maximum offer bid prices can send timely advertisements to subscribers who are physically within a specific geographic area to promote products, services, or events. Companies can also send offers to subscribers based on segment, which is defined by simple filters such as recent coffee purchases or grocery store visits. Subscribers are matched to segments and delivered push notifications as ads.

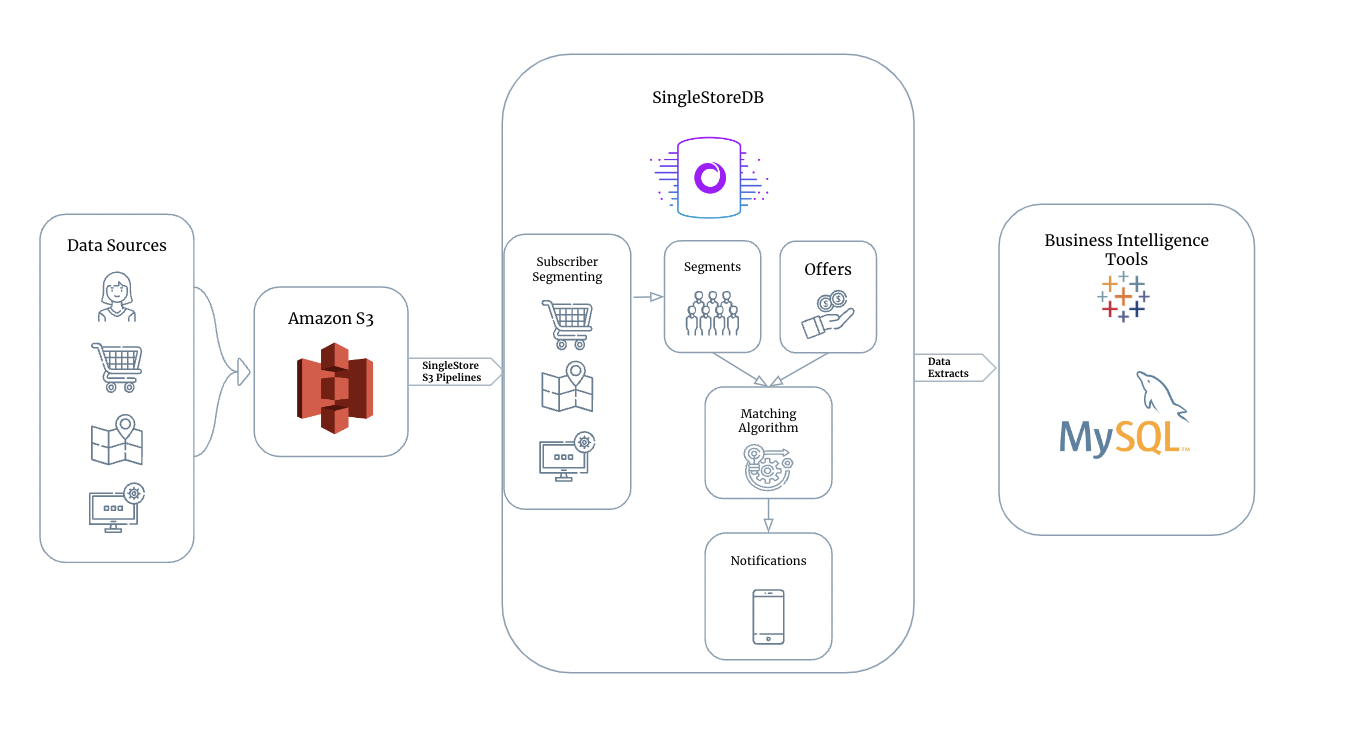
This approach can help businesses build stronger relationships with their customers by providing them with offers and promotions that are tailored to their specific interests and needs.

The primary purpose of this project is to analyze the effectiveness of SingleStore's Real-Time Digital Marketing application in driving sales and conversions for businesses. Our methodologies include ingesting data from SingleStore, understanding the data attributes and relationships, performing analysis using SQL, establishing KPIs for performance, creating visualizations to evaluate and showcase the application's impact, deriving insights, and, lastly, identifying limitations and challenges.

We expect to gain insights into the application's performance and identify improvement opportunities. By doing so, we hope to help businesses make more informed decisions about leveraging this technology to drive sales and improve customer engagement.

# **Data Literacy**

The RealTime Digital Marketing application simulates serving ads to users based on historical behavioral data such as purchase history and requests along with the user’s real-time location.



The sources are stored in Amazon S3 buckets and SingleStore S3 Pipelines are used to ingest the data into SingleStore in real-time. The simulation sets up a one time creation and insertion of schemas to be used with the live streamed data.

1. The schemas, procedures, and functions are created and populated in SingleStore.
2. Pipelines are enabled to stream data using SingleStore S3 Pipelines from S3 to SingleStore.
3. Procedures and functions are called to:
   1. Process locations
   2. Process requests
   3. Process purchases
   4. Update subscriber segments
   5. Prune subscriber segments
   6. Match subscribers to offers, populating the notification to simulate a push offer

## **Ingestion**

Three pipelines are dedicated for streaming locations, requests, and purchases data from S3.

| Data Stream | Fields | Sample Record | S3 Bucket Path | Avg Rows/Sec |
| --- | --- | --- | --- | --- |
| locations | city\_id,  subscriber\_id  ts,  lonlat,  olc\_8 | 120658,  354,  2023-01-30 16:08:45.812682, POINT(-73.98643021 40.72619964), 87G8P2G7 | singlestore-realtime-digital-marketing/v2/100k-2p/locations.\* | 8159.18 |
| requests | city\_id,  subscriber\_id ts,  domain | 120658,  58064, 2023-01-30 16:10:24.716805, youspan.biz | singlestore-realtime-digital-marketing/v2/100k-2p/requests.\* | 9630.012 |
| purchases | city\_id, subscriber\_id ts,  vendor | 120658,  55740, 2023-01-30 16:07:43.226184, Divape | singlestore-realtime-digital-marketing/v2/100k-2p/purchases.\* | 2142.87 |

## **Table Statistics**

In the Singlestore ecosystem, data is stored as either columnstores or rowstores. Live streamed data (locations, purchases, requests, notifications) is stored as columnstores which is optimized for OLTP.

Other data is stored as rowstores which is optimized for OLAP.

| Table Name | Table Storage | Space in memory (mb) | Space in disk, uncompressed (mb) | Space in disk, compressed (mb) | Volume (in rows) |
| --- | --- | --- | --- | --- | --- |
| subscriber\_segments | Rowstore | 1,128 | 0 | 0 | 4077170 |
| cities | Rowstore | 1 | 0 | 0 | 1 |
| offers | Rowstore | 1 | 0 | 0 | 200 |
| offers\_fact | Rowstore | 1 | 0 | 0 | 289 |
| worldcities | Rowstore | 35 | 0 | 0 | 128769 |
| subscribers\_last\_notification | Rowstore | 20 | 0 | 0 | 49421 |
| notification\_target\_dim | Rowstore | 1 | 0 | 0 | 166 |
| notification\_content\_dim | Rowstore | 1 | 0 | 0 | 197 |
| subscribers | Rowstore | 51 | 0 | 0 | 100000 |
| segment\_dim | Rowstore | 1 | 0 | 0 | 277 |
| sessions | Rowstore | 1 | 0 | 0 | 1 |
| customer\_dim | Rowstore | 1 | 0 | 0 | 137 |
| notifications | Columnstore | 0 | 69 | 12 | 696790 |
| locations | Columnstore | 0 | 2,002 | 344 | 21800000 |
| purchases | Columnstore | 0 | 915 | 78 | 12310115 |
| requests | Columnstore | 0 | 1842 | 145 | 22353773 |
| **SUM** |  | **1.24 GB** | **4.82 GB** | **579 MB** | **61,517,306 rows** |

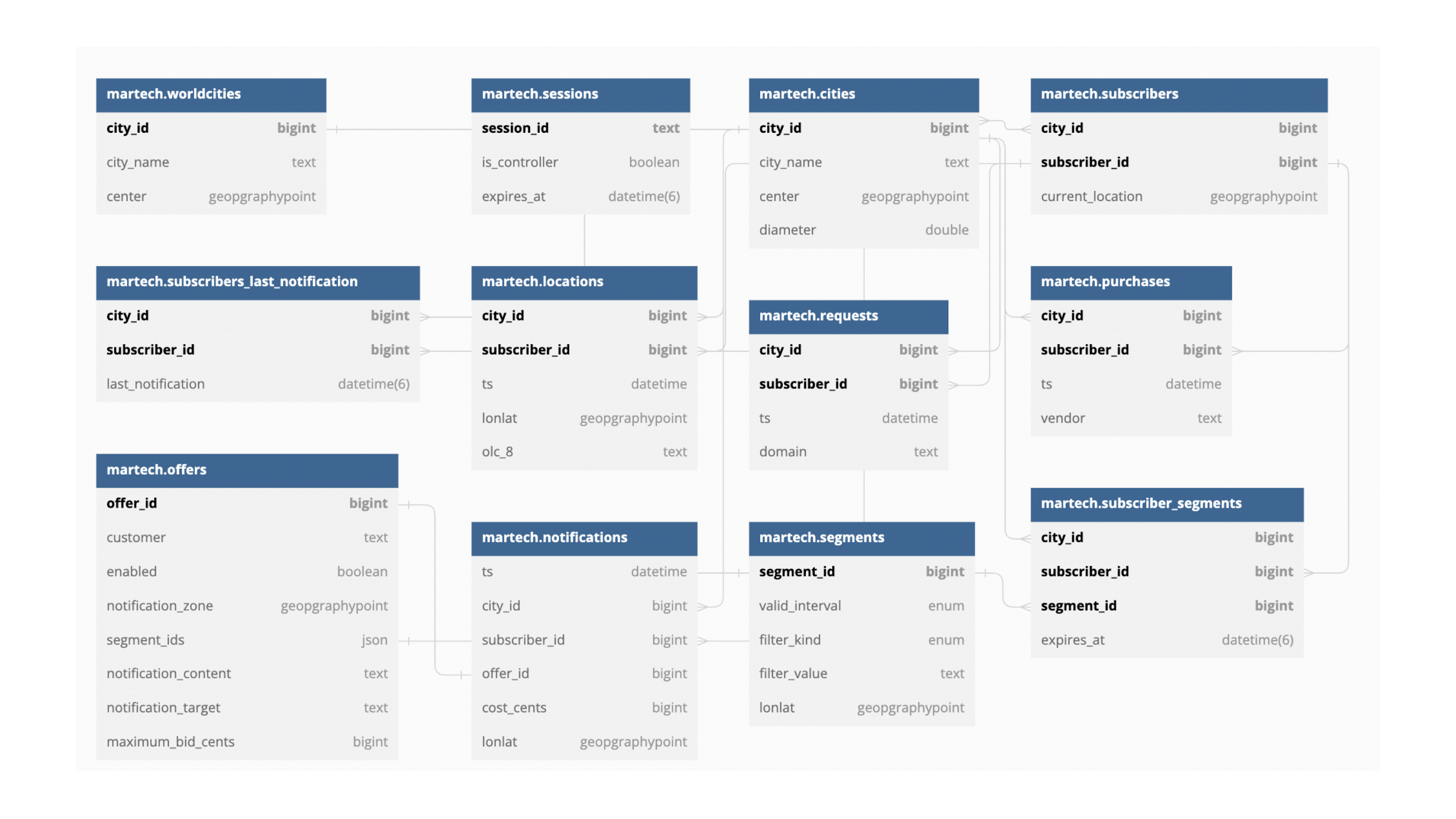
## **Procedures & Functions**

In addition to the data, the simulation also creates procedures and functions to process the streamed data to simulate segmenting, matching, and notifications.

| **Name** | **Type** | **Purpose** | **Data Affected** |
| --- | --- | --- | --- |
| process\_locations | Procedure | Process locations of subscribers. | INSERT locations  INSERT subscribers |
| process\_requests | Procedure | Process requests of subscribers. | INSERT requests |
| process\_purchases | Procedure | Process purchases of subscribers. | INSERT purchases |
| prune\_segments | Procedure | Prunes subscriber segments when expired. | DELETE subscriber\_segments |
| run\_matching\_process | Procedure | Matches subscribers to offers and updates notifications. | INSERT subscribers\_last\_notification  INSERT notifications |
| update\_segments | Procedure | Creates new subscriber segments based on requests, purchases, locations. | INSERT subscriber\_segments |
| dynamic\_subscriber\_segments\_purchases | Table Valued Function | Creates new subscriber segments based on purchases. |  |
| dynamic\_subscriber\_segments\_locations | Table Valued Function | Creates new segments for subscribers based on locations. |  |
| dynamic\_subscriber\_segments\_requests | Table Valued Function | Creates new segments for subscribers based on requests |  |
| dynamic\_subscriber\_segments | Table Valued Function | Uses the previous three functions to create subscriber segments. |  |
| match\_offers\_to\_subscribers | Table Valued Function | Matches offers to subscribers based on segments. |  |

# **Data Dictionary**

## **ER Diagram**



| **Field** | **Type** | **Length (characters)** | **Table(s)** | **Description** |
| --- | --- | --- | --- | --- |
| city\_id | bigint | 20 | worldcities, cities, subscribers\_last\_notification, subscribers, locations, requests, purchases, notifications, subscriber\_segments | Unique identifier for city. |
| subscriber\_id | bigint | 20 | cities, subscribers, subscribers\_last\_notification, locations, requests, purchases, notifications, subscriber\_segments | Unique identifier for a subscriber. |
| city\_name | Text | 21845 | worldcities, cities | City name. |
| center | geographypoint |  | worldcities, cities | Longitude/Latitude of the city center. |
| diameter | double | 20 | cities | Used to calculate the diameter of city limits. |
| current\_location | geographypoint |  | subscribers | Last recorded location of subscriber. |
| last\_notification | datetime |  | subscribers\_last\_notification | Timestamp of most recent notification sent to subscriber. |
| ts | datetime |  | Locations, requests, purchases, notifications | Timestamp of event. |
| lonlat | geographypoint |  | Locations, notifications | Longitude/Latitude of subscriber/notification. |
| domain | text | 21845 | requests | Domain name. |
| vendor | text | 21845 | purchases | Customer name. |
| customer | text | 21845 | offers, offers\_fact | Customer name. |
| offer\_id | text | 21845 | offers | Offer unique id. |
| enabled | boolean | 1 | offers | Indicates active offer. |
| notification\_zone | geographypoint |  | offers, offers\_fact | Activation zone for notification. |
| segment\_ids | JSON |  | offers | List of segment\_ids to show notification to. |
| notification\_content | text | 21845 | offers | Content of ad. |
| notification\_target | text | 21845 | offers | Domain target for notification. |
| maximum\_bid\_cents | bigint | 20 | offers, offers\_fact | Maximum bid for this offer. |
| cost\_cents | bigint | 20 | notifications | Cost to show notification. |
| segment\_id | bigint | 20 | segments, subscriber\_segments, segment\_dim | Unique identifier for segment group. |
| expires\_at | datetime |  | subscriber\_segments | Expiration time. |
| filter\_kind | enum (olc\_8, request, purchase) |  | segments, segment\_dim | Segment filter enum. |
| valid\_interval | enum (minute, hour, day, week, month) |  | segments, segment\_dim | Valid interval. |
| segment\_key | bigint | 20 | segment\_dim, offers\_fact | Segment Key. |
| notification\_content\_key | bigint | 20 | notification\_content\_dim, offers\_fact | Notification content key. |
| notification\_target\_key | bigint | 20 | notification\_target\_key, offers\_fact | Notification target key. |
| customer\_key | bigint | 20 | customer\_dim, offers\_fact | Customer key. |
| offer\_key | bigint | 20 | offers\_fact | Offer key. |

# **Data Processing and Architecture**

Exact details for how the data extracts were constructed are in the transformations.sql addendum.

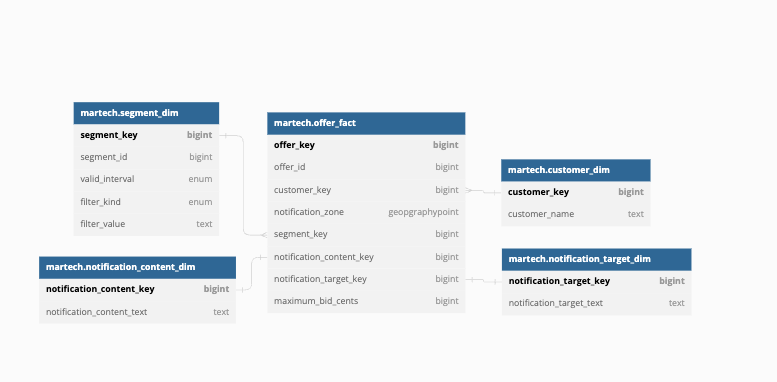
## **Data Transformation**

The live streamed data is kept as rowstores to optimize random reads and writes. The rowstore tables are kept in the same format.

To aid with analytical processing and ingestion with BI tools, the `offers\_fact` table leverages dimensional modeling to capture the data stored in the out-of-the-box `offers` table.

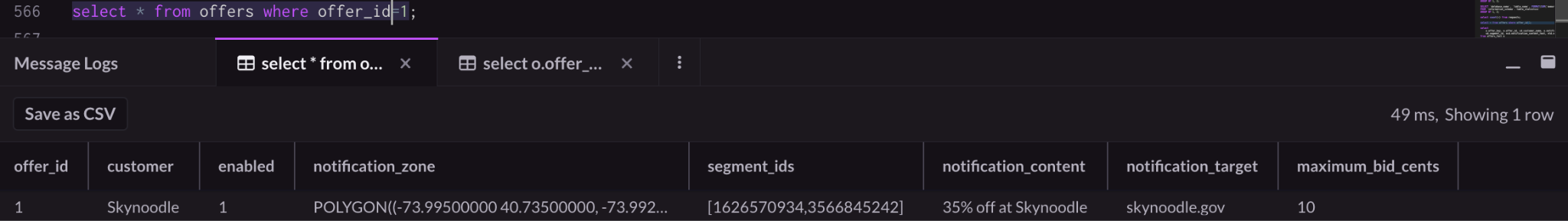
The out-of-the-box `offers` table used a JSON array to store segment\_ids which proved to be challenging to parse and inefficient resource-wise. Instead, the `offers\_fact` table is granularized to provide the finest detail for an offer. If an offer targets two segment\_ids in the `offers` table then there will be two records in the`offers\_fact` table that correspond to the original offer.

In accordance with dimensional modeling principles, a star schema is used to model `offers\_fact`.

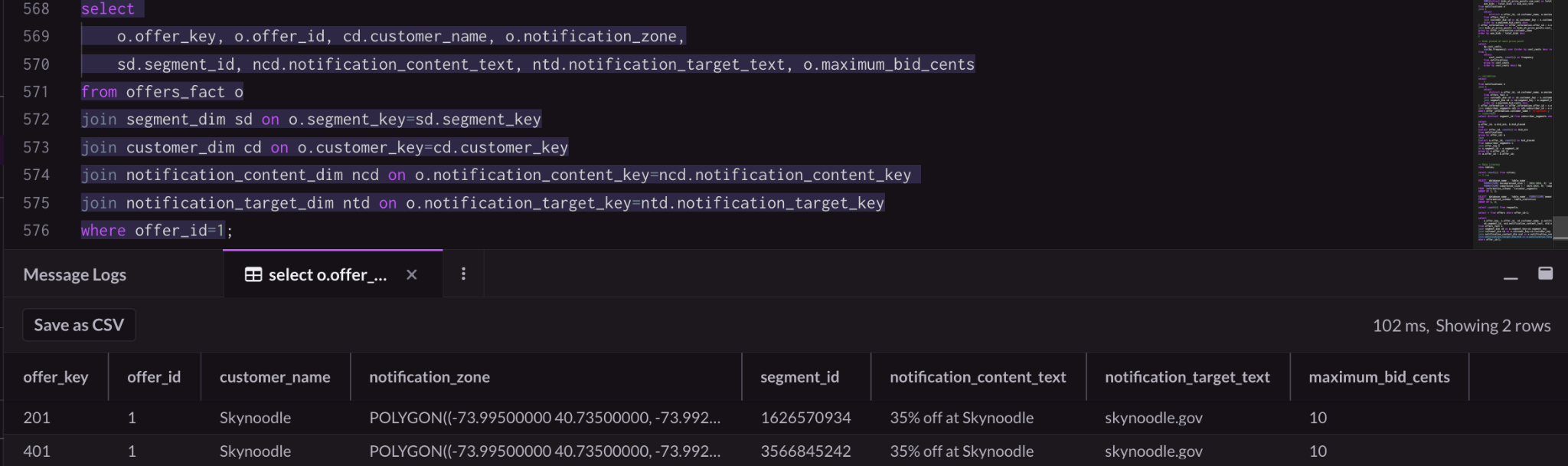


The `offer\_fact` table contains the finest granular detail of an offer. The dimensional tables, `segment\_dim`, `notification\_content\_dim`, `customer\_dim`, `notification\_target\_dim` provide further detail.

`offers`



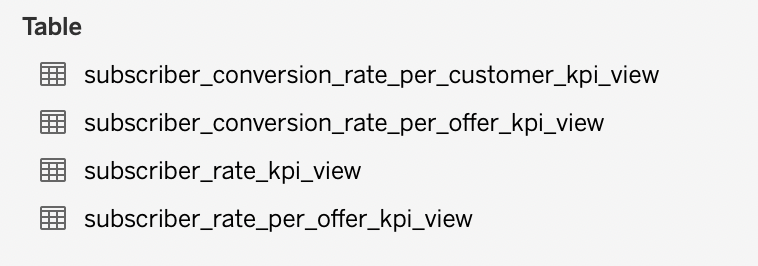
`offers\_fact`



## **Views**

Following the principles of least-privilege for security and data integrity purposes, data analysts were granted read-only access to Views of the data and not the actual data itself in SingleStore.

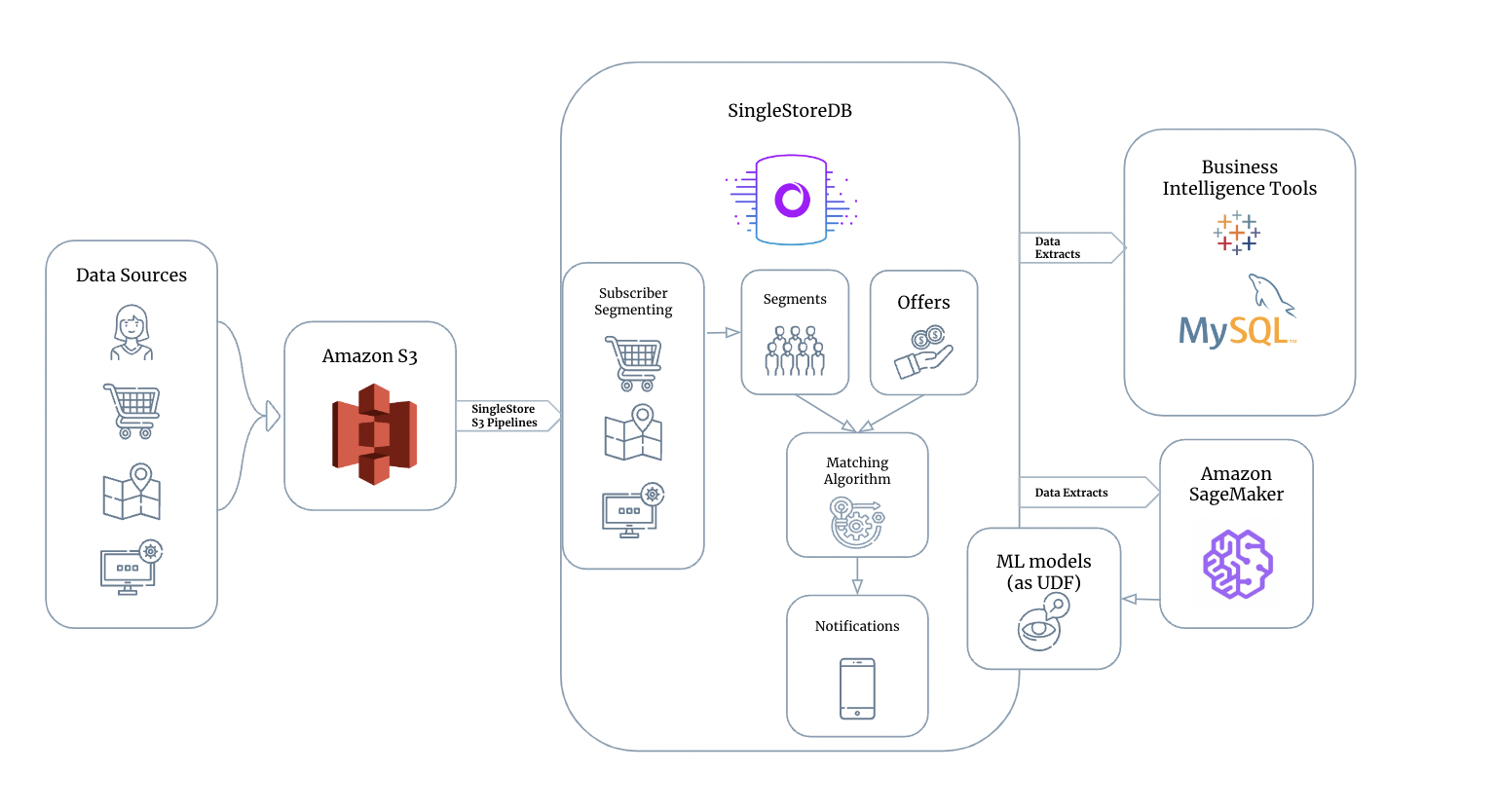
The Views are defined by the SQL queries used to construct the KPIs.



## **Future Considerations**

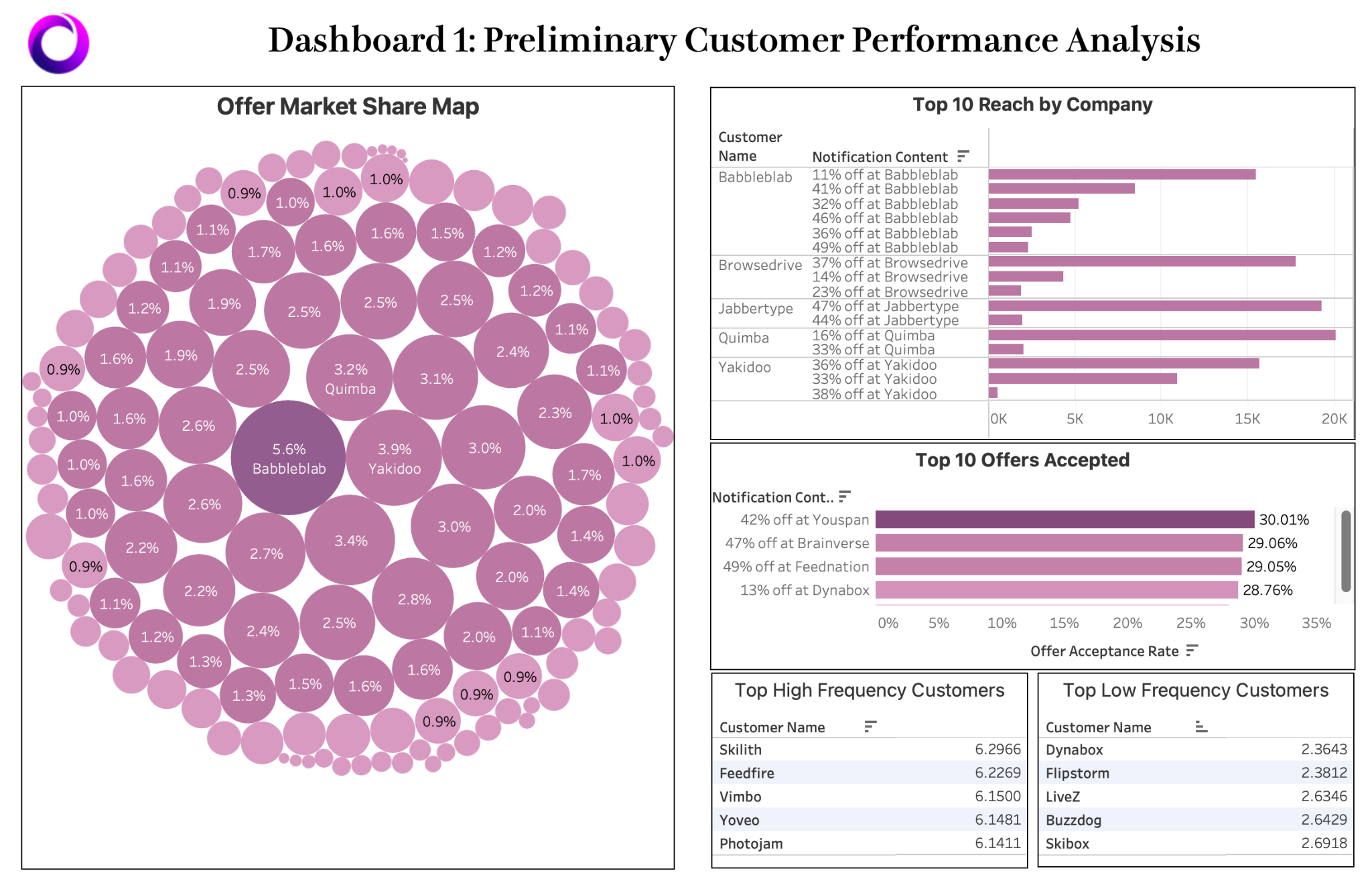
The current architecture and schemas are well suited for descriptive and diagnostic analytics. Both OLTP and OLAP workloads are optimized.

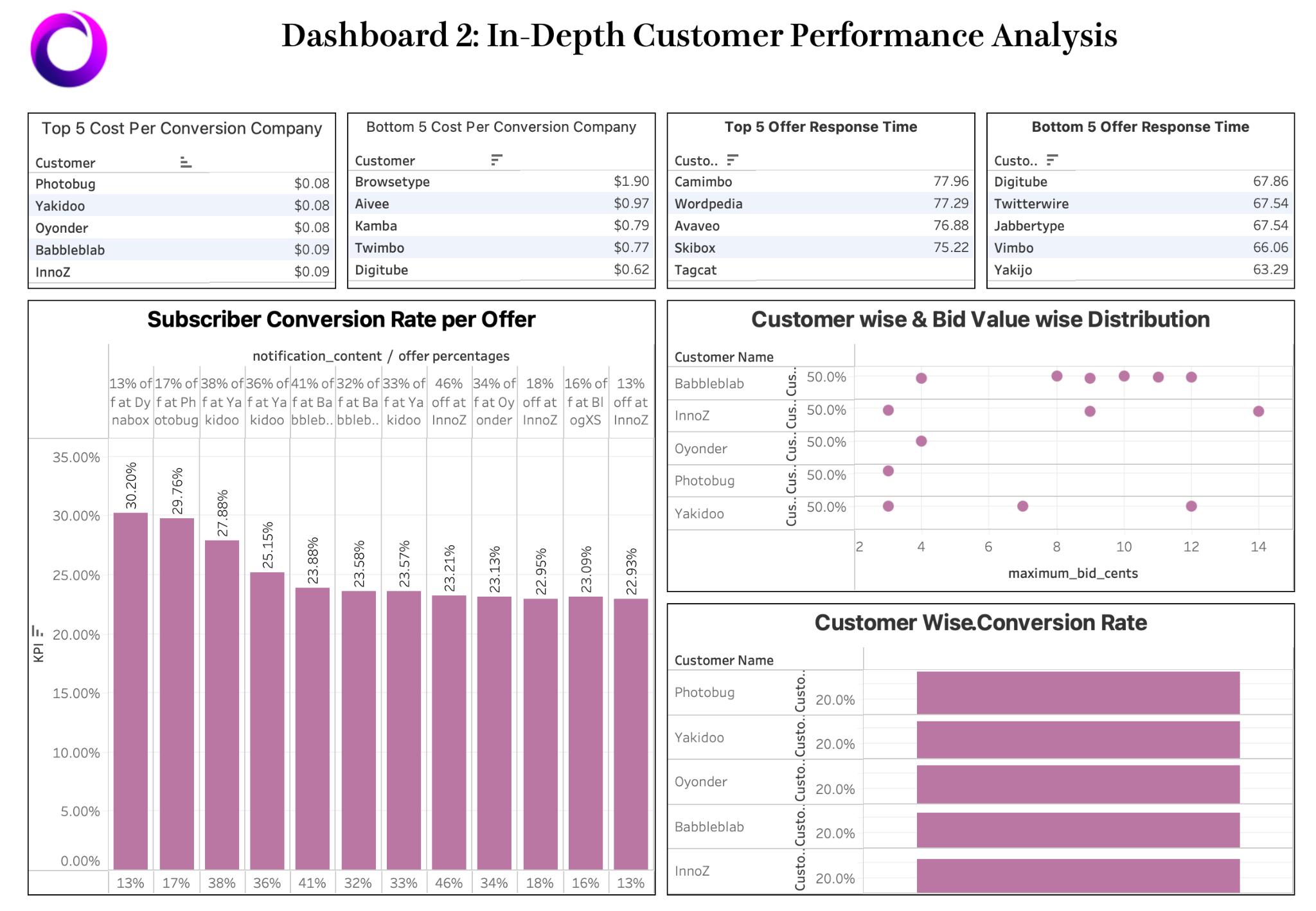
In the future, should predictive or prescriptive analytics be desired, integrations with additional services such as AWS SageMaker or AzureML can be leveraged to deploy regression models that can execute against real time data to suggest information like suggested bid price, suggested segment to target.



# **KPIs & Insights**

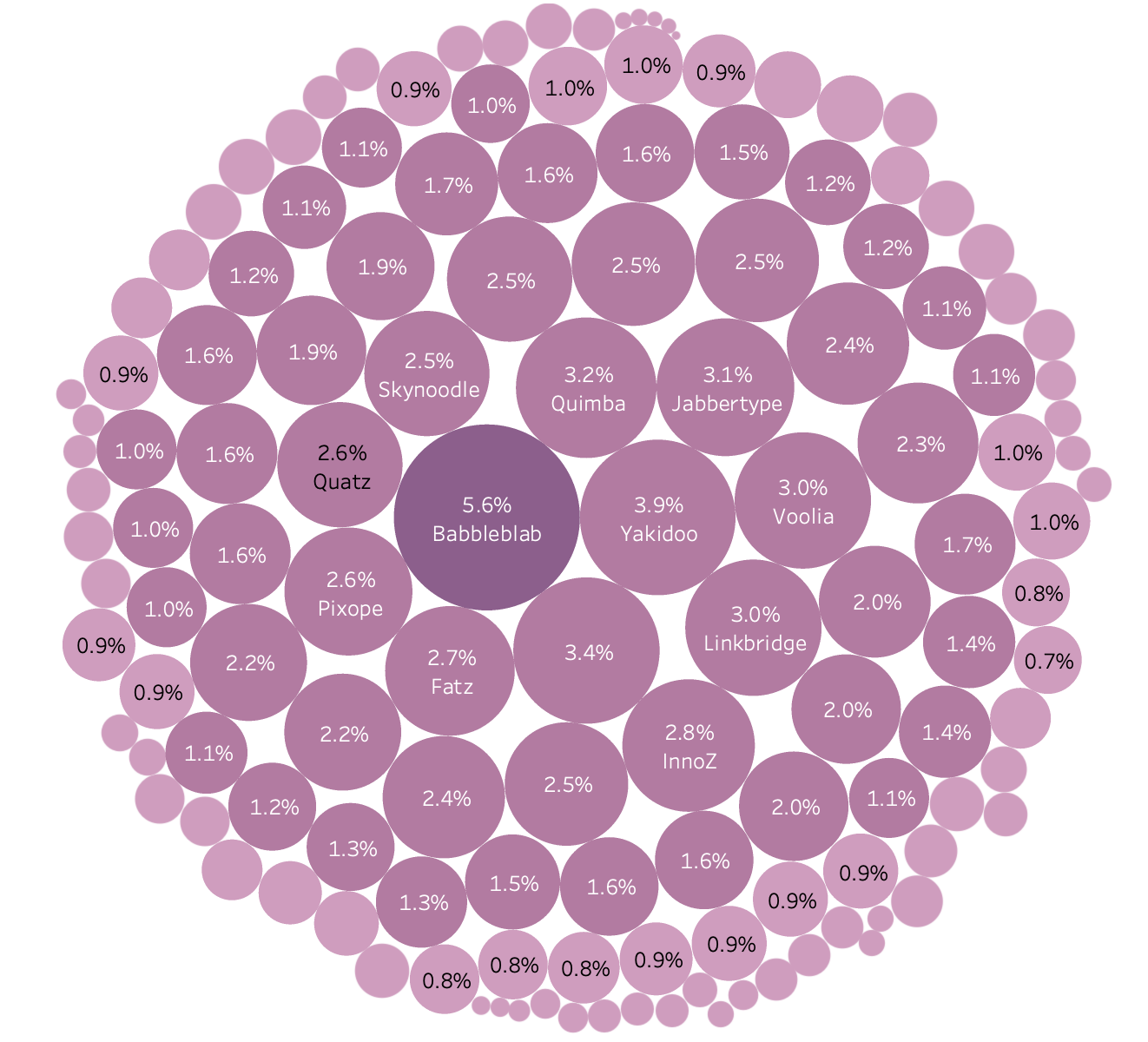
**Main Dashboards**





**KPI: Offer Market Share**

The offer market share KPI is a measure of a customer's share of the total offers won out of all offers made. It is calculated by dividing the total number of offers won by a customer by the total number of notifications received by all customers. This KPI provides insights into the customer's quantity of winning offers compared to the overall market.



### Key Takeaways

The key takeaway from this KPI is that it helps in identifying the customers who are winning a larger share of the offers compared to others, thus identifying the most important customers of ours.

### Assumptions

It provides insights into the customer's performance in their bidding strategy.

### Methodology

*Calculation: # of notifications by each customer / Total # of notifications by all customers*

The methodology of calculating this KPI involves using data from the notifications table and joining it with the offers\_fact and customer\_dim tables to get information about each customer’s winning offers. The query groups the result by customer name and calculates the total number of offers won by each customer and their bid market share. The bid market share is calculated by dividing the number of offers won by a customer by the total number of notifications received by all customers.

### Insights

As shown from the table, Bubbleblab has the largest Offer Market Share (5.6%), while Brainbox has almost none (0.0%). The market share ranges from 0.0% to 5.6%, indicating variation in the performance of different customers in the market. Some customers are winning a trivial share of offers, while others are winning a larger share. However, the disparity is not significant since the range is within 5.6%. This is advantageous for SingleStore because it has a diverse customer base and is not overly reliant on a few customers. It reduces the company's risk of losing a significant portion of its revenue if it loses one or a few key customers.

Based on the data, we can categorize the customers into three groups. Customers with a market share between 0.001% to 1% are considered to have a low market share; Customers with a market share between 1% to 5% are considered to have a moderate market share; Those with a market share above 5% are considered to have a high market share. SingleStore should pay special attention to this group since it is winning a substantial share of the offers compared to its competitors, indicating that it is bidding for more offers and has a competitive advantage in their bidding strategy.

### Challenges

One challenge with this KPI is that it only provides insights into the number of offers won by a customer, but it does not consider the quality of the offers or their impact on the business's revenue. A customer may have a high market share in terms of the number of offers won, but these offers may not generate significant revenue for the business. Therefore, it is essential to combine this KPI with other metrics, such as the revenue generated from the offers to get a more comprehensive picture of a customer's performance in the market. Some customers are winning a trivial share of offers, while others are winning a larger share. However, the disparity is not significant since the range is within 5.6%.

Based on the data, we can categorize the customers into three groups. Customers with a market share between 0.001% to 1% are considered to have a low market share; Customers with a market share between 1% to 5% are considered to have a moderate market share; Those with a market share above 5% are considered to have a high market share. This group is winning a substantial share of the offers compared to their competitors, indicating that they are bidding for more offers and have a competitive advantage in their bidding strategy.

**KPI: Cost per Conversion (CPCo)**

****

## 

### Key Take-aways

* We did well in keeping our service cost low and keeping our effectiveness consistent for each of our clients.
* We need to make sure we communicate with clients about their offer strategies in order to maximize their results on our platform.

### Methodology

The methodology of calculating this KPI involves joining data from offers, notifications, and purchases tables to get information about each customer’s total cost and its conversion number. The Cost per Conversion is calculated by dividing the total cost for each customer by the total number of notifications received by all customers.

Assumptions

There are no significant changes in market conditions or customer behavior that may impact the KPI results.

Insights

We analyzed the cost per conversion rate for an app notification distribution platform and found that the average cost per conversion is 0.2018 dollars, with a maximum of 1.897 dollars and a minimum of 0.076 dollars.

Several companies, including Photobug, Yakidoo, Oyonder, Babbleblab, and InnoZ, perform exceptionally well on the cost-per-conversion KPI. They may have achieved this by offering high discounts, targeting the right customer segments, or having attractive soft attributes like good design. We may need to conduct a more detailed investigation into their offer strategies in order to learn about what they did well that have also boosted our service quality. We can zoom in a particular company – Babbleblab. Most companies have a cost per notification of 12 cents, which is considered the market standard. However, Babbleblab stands out by offering a cost per conversion of only 10 cents, which is likely due to its 41% discount offer. Babbleblab performed very well on the cost-per-conversion KPI, possibly due to its low per-offer cost.

However, some companies have very high conversion costs, with Browsetype, Aivee, Kamba, Twimbo, and Digitube costing up to 2 dollars per conversion. This may be due to low conversion rates. There are reasons behind this which may require individual clients to change their pricing strategy. For example for Twimbo, they only offer an 18% discount rate, while other companies normally offer more than a 40% discount.

On a positive note, we noticed that the cost per conversion for each offer within a company is roughly the same, indicating that we offer consistent services and have good metrics for customer segmentation and offer promotion. This suggests that our platform is reliable.

## 

### Challenges

Comparability may be an issue when comparing the cost per conversion rates of different companies. It is important to ensure that the comparisons are fair and meaningful but companies may have different pricing strategies, customer segments, offer types, and other factors that can affect their cost per conversion rates. It is important to take these factors into account when comparing different companies to avoid drawing incorrect or unfair conclusions. It is also why we should be discussing these factors with the consumers in our service and maybe provide guidelines for them.

# **KPI: Offer Response Time**

### Key Take-aways

Offer response time is used to evaluate the time taken for a subscriber to react to a notification. This is a measure of both the “effectiveness” of a customer's brand and/or offer, as well as a measure of subscriber engagement.

Based on the data we have from real-time simulation, there are no key take-aways for offer response time. We believe this metric still holds important information, provided we have more robust data.

### Methodology

Offer Response Time = Request\_ts - Notification\_ts

Notification\_ts is the timestamp of a notification pushed out by a customer. Request\_ts is the timestamp of the first request a subscriber makes asking for information about the client. The difference between the two timestamps is the offer response time.

Besides analyzing offer response time by itself, we also analyzed this metric across multiple segments, including customer, time (hour of day), and content.

### 

### Assumptions

We are assuming that the lack of insights is caused by the robustness of data compared to the usefulness of the KPI itself. We will talk more about data challenges in the section below. We assume that this KPI would be able to provide insightful analysis if the data we had on hand was more robust. For example, we are interested in whether specific customer segments are more engaged with ads, or if there is a correlation between ad content (the offer being given) and quickness in response time. We believe that with better data, these insights could potentially be extracted.

### Insights

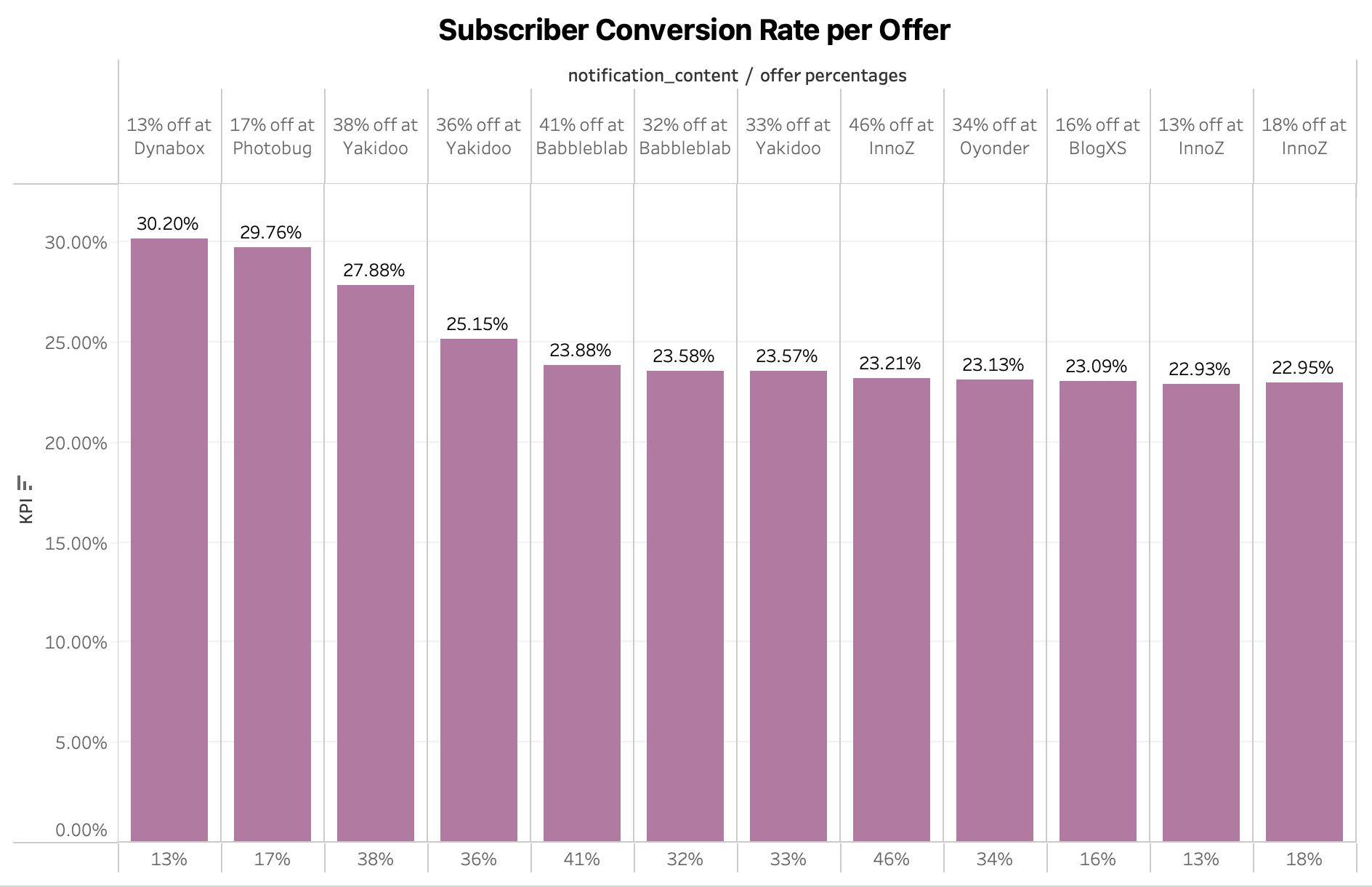
Due to data challenges faced during analysis, there was no useful insight from this analysis.

### 

### Challenges

The simulated data proved to be a significant challenge for us when analyzing offer response time. While looking into the data, we noticed that notification\_ts and request\_ts did not overlap, meaning that all notifications came through in the span of an hour, before all requests started coming through in the span of another hour after the notifications. In real life, there should be an overlap. As such, moving forward, we will require more robust data, such as real world data instead of simulated data, in order to provide a better analysis of offer response time.

**KPI: Subscriber Conversion Rate per Offer**

****

### Key Take-aways

Subscriber conversion rate is a very important metric for both our own organization to evaluate how well our customers’ offers are working and the value they bring to us, as well as for our customers to understand their own offer performances . As an organization, we need to only put out the best offers so that our subscribers see value in our platform.

### Methodology

This KPI is measured as:

COUNT([purchase\_ts])/COUNT([notification\_ts])

Notification\_ts is the timestamp of a notification pushed out by a customer to a subscriber. Purchase\_ts is the earliest purchase made by a subscriber based on the offer. This way, we are able to estimate the exact effect of the offer. For each offer, we are calculating the number of purchases made by subscribers and dividing that by the number of times the offer was pushed out.

### Assumptions

Here, we are assuming that only the first purchase made based on an offer is the true effect of the offer.

### Insights

The highest subscriber conversion rate is 0.3027, for the offer ‘13% from Dynabox’. This is surprising because we would expect the highest conversion rates to be from much higher discount rates. Therefore, the customer (here, Dynabox) is likely a more important factor than discount rates. We should prioritize ‘higher quality’ customers.

Within customers, as expected, conversion rates are higher for offers with higher discount rates. If we look at the company ‘Yakidoo’, the conversion rate for their 38% offer is 0.27751, their 36% is 0.252, and their 33% is 0.234. Similarly, for the company ‘Babbleblab’, their conversion rates for their 41% discount offer and their conversion rate for their 33% offer are 0.24 and 0.235 respectively. However, the difference in the conversion rates for the different offer rates does not seem to justify the extra cost borne by the company for higher discount rates. A more thorough analysis needs to be done to weigh the pros and cons.

It would be useful for each of our companies to do this cost benefit analysis so they can spend less on the offers and make higher profits by offering lower discounts. More profit for our companies means that they will continue to advertise with us, leading to higher profits for our own organization in the long run.

### Challenges

The data being simulated may not give us a completely realistic view of the behaviour of the conversion rates of different offers. Right now, it seems like offering higher discounts may not be the best strategy, but more data and more analysis on real data needs to be done to verify this trend.

# **KPI: Customer Conversion Rate**

Customer Conversion Rate measures the percentage of customers (companies) whose ad (offer notification) is clicked on by a subscriber and a purchase has been made.

It is calculated by total no. Of purchases due to notifications made by total no. Of notifications showed. It helps in assessing the success of notifications.

**Top 5 performing companies:**

## 

### Key Take-aways

Companies can be categorized into three categories - underachieving, average and overachieving depending on the customer conversion rate and the conversion rate is mainly driven by segment targeting and supplemented by bid value. Therefore, special attention needs to be paid to underachieving companies with high bid values.

### Methodology

#distinct purchases made within an hour of showing the notification / #total notifications showed

Data from tables notifications and offers was utilized to identify customers whose ads were shown to subscribers. These notifications were then mapped with purchases using subscriber\_is and timestamp to identify the corresponding purchase(s) made.

### Assumptions

Purchase is driven by the notification if the purchase has been made within an hour after showing the notification.

## 

### Insights

Average customer conversion rate is 25.3% and goes as high as 55.2% (for Photobug)

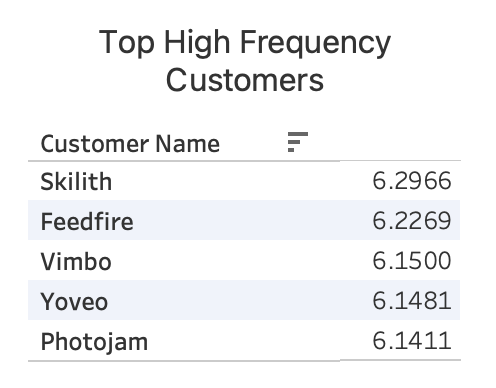
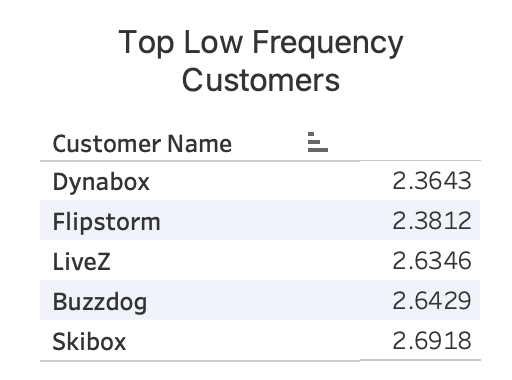
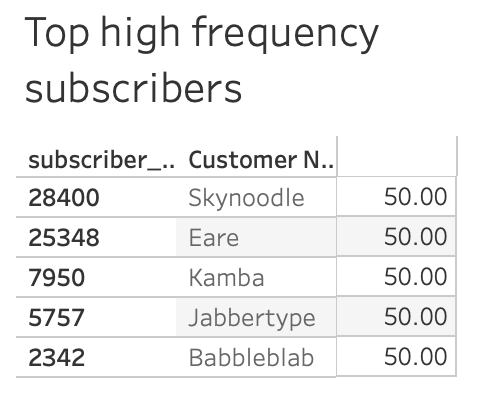
Top 5 companies with conversion rate are - Photobug, Yakidoo, Oyonder Babbleblab and Innoz whereas Bottom 5 companies with conversion rate are - Yata, Browsetype, Kamba, Twimbo & Yakijo with lowest conversion rate at 1.5%. As can be seen from the graph, companies like Yata, Yakijo, & Browsetype have high bid value but still the lowest conversion rate. Possible reason could be inefficient segment targeting thus these companies have the most scope of improvement.

### Challenges

To map purchases with corresponding notifications while accounting for duplicate calculations. To overcome this, a timeframe restriction was applied to limit the purchases that could be mapped to notifications. The timeframe was taken to be an hour by calculating the average from the data. Additionally, a distinct count of purchases was taken to solve the problem of multiple counting.

Panel Data wasn’t available so in-depth analysis on the notification performance over time could not be conducted.

**KPI: Frequency**



### 

### 

### 

### 

### 

### 

### Key Take-aways

There is a huge disparity between frequency of ads amongst different customers. Some customers/companies are showing the same ad up to 50 times per subscriber on average. A high frequency may indicate that users are being shown the same ad too many times, which can lead to ad fatigue. Our organization needs to make sure that our subscribers are being shown the right ads that lead to purchases, rather than the same ad over and over again.

### Methodology

This KPI is measured in two ways, one based on subscriber\_id (average number of times the:

1. The average number of repeated offers every subscriber is receiving (average frequency of ads per subscriber)
2. The average number of repeated offers every customer is pushing out.

### Insights

Ad frequency needs to be optimized to lead to the largest possible conversion. If an ad is shown multiple times to subscribers in order to make them aware of the product and slowly increase their likelihood of purchasing the product, it may be worth the repeated ads. However, generally, a high frequency is a poor strategy. Therefore, the companies that have the highest average frequency (eg: Skilith, Feedfire, Vimbo etc.) need to be looked at and changes need to be made in their strategy. It is important to ensure that these companies do not continue to create a poor experience for our customers.

Additionally, from the subscriber point of view, the top highest frequency customers need to be prioritized and our organization needs to take additional steps to ensure that they have a better experience on our platform and do not leave.

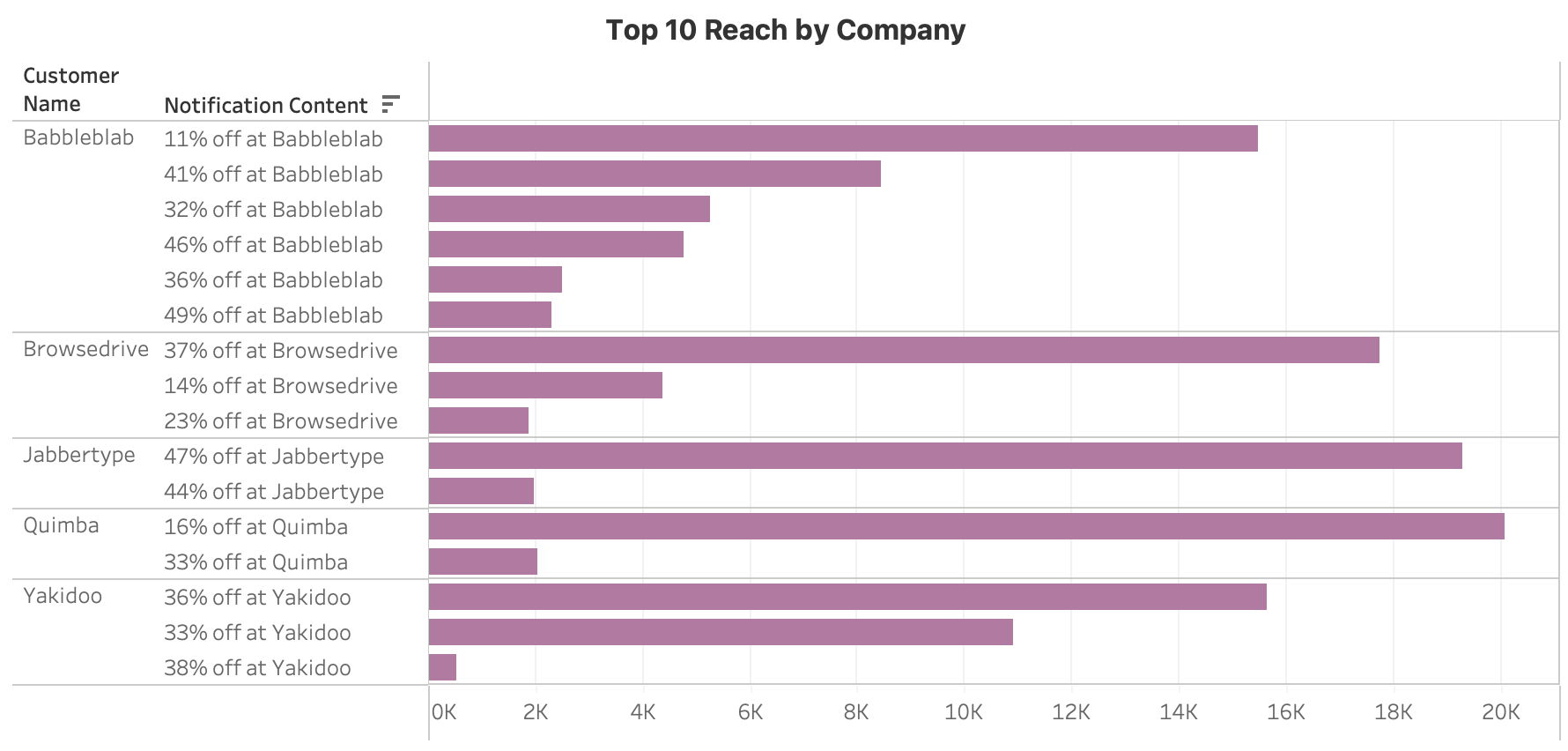
### Challenges

Due to simulated data, it is possible that this ad behaviour is not a reflection of true offer behaviour. It is very unlikely that the same ad/same company ad is shown to the same subscriber 50 times (see top high frequency subscribers). However, if this ad behaviour is realistic, then serious measures need to be taken to ensure that this does not happen.

# **KPI: Reach**

This measures the number of unique subscribers who have seen an ad (offer notification). A high reach indicates that the ad is reaching a large audience. We further dive into offer and notification content level to get a better understanding of the reach.

Top 5 companies based on Reach



### Key Take-aways

Reach is an important metric that sets a baseline for the customers to measure the impact of their marketing efforts. More the reach for an organization, more revenue and business it will generate.

### Methodology

Calculating reach is not a complicated process. We can iterate through all the notifications that are sent and join this table with offers\_fact table which contains all the details about the offer present in the notification as well as notification\_content\_dim for the content of the notification.

### Assumptions

We assume that the offers are effective enough to further drive sales.

## 

### Insights

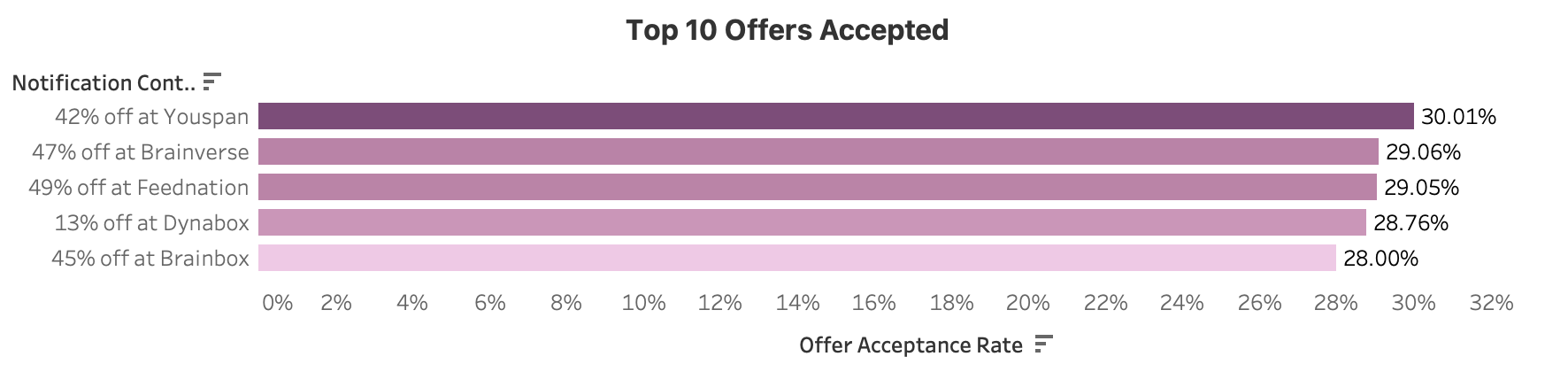
We see that Babbleblab, Browserdrive, Jabbertype, Quimba and Yakidoo are the top 5 customers wrt Reach. Offers like 16% off at Quimba have been sent to over 20,000 end users. Companies like Babbleblab and Yakidoo with very high reach are performing exceptionally in different KPIs as well.

## 

### Challenges

Data quality and consistency play an important role for this KPI. Broken pipelines and missed records are some of the primary challenges that we might have to face while calculating this metric. It may be a bit of a challenge to calculate it with a large number of records.

**KPI: Offer Acceptance Rate**

****Key Take-aways

The top 3 notifications that had the greatest offer acceptance rates all had discounts above 40%. We think offering a higher discount rate makes the offers more successful in general.

### Methodology

#requests / #notifications = count(request\_ts) / count(notification\_ts)

We calculate the number of requests made from the requests table and divide it by the total number of notifications, which we get from the offers and notifications tables, grouped by each offer\_id.

Assumptions

We assume that seeing the offer led the subscribers to make the purchase and use the offer. We ignore the fact that subscribers who saw the offer wanted to make purchases on the domain already, and they happened to see the offer thus used it to make the purchase.

### Insights

An offer once received by the subscriber may be accepted or not accepted. The offer acceptance rate measures the ratio of offers accepted against the notifications that went out to all subscribers for the specific offer. This is an important metric to quantify how effective the campaign is since it measures how many offers that were seen by someone led them to use the offer to make a purchase. It is a direct measure of how successful the discount offer displayed to the users was to drive them to make a purchase, while using the discount offer.

Challenges

Since the data is simulated and not real, we do not learn much about actual subscriber behavior. For our case, it so happens that the top 3 offer acceptance rates are all above 40% off, but we see offers with lower discount rates at the top too. Due to our assumption that seeing the offers led the subscribers to make the purchase, we may be overestimating the effect of showing the offer to the subscriber.

# **Project Challenges**

A common challenge across the project was that of simulated data which might not give us a complete realistic picture of various metrics. For ‘Reach’, we have to ensure data quality and consistency and fix any broken pipelines or production issues immediately to get the most accurate number. Segment IDs also proved challenging due to its many-to-many nature. Our initial assumption was that offers only bid for notifications for subscribers that belong to its list of segments. This was not an accurate reflection of the data. Subscribers were being notified about offers that were unrelated to them. One interpretation could be that the matching algorithm is still immature, another interpretation is that this was an oversight of the simulation.

As the consumer’s business grows and data increases drastically, the amount of compute resources has to be increased. For ‘Cost per conversion’, comparability comes out to be a challenge as companies have different pricing strategies and customer segments that can affect their cost per conversion rate. For ‘Offer Response Time’ and ‘Customer conversion rate’, we are facing counter-intuitive results possibly due to simulated data. More data and analysis will further confirm this issue.

Working with the notification zone and geographic polygon constructed from longitude & latitude points also proved challenging. Tableau was not able to parse the list of longitudes and latitudes. To make this feasible, additional transformations can be done to the notification zone field.

# **Appendix**

| Finals\_2023\_group10\_KPIs.sql | SQL queries for all KPIs. Includes both aggregated and non-aggregated granular data. |
| --- | --- |
| Finals\_2023\_group10\_transformations.sql | SQL queries for creating fact & dim tables and table statistics. |
| Finals\_2023\_group10.twbx | Tableau Dashboard Extract. |