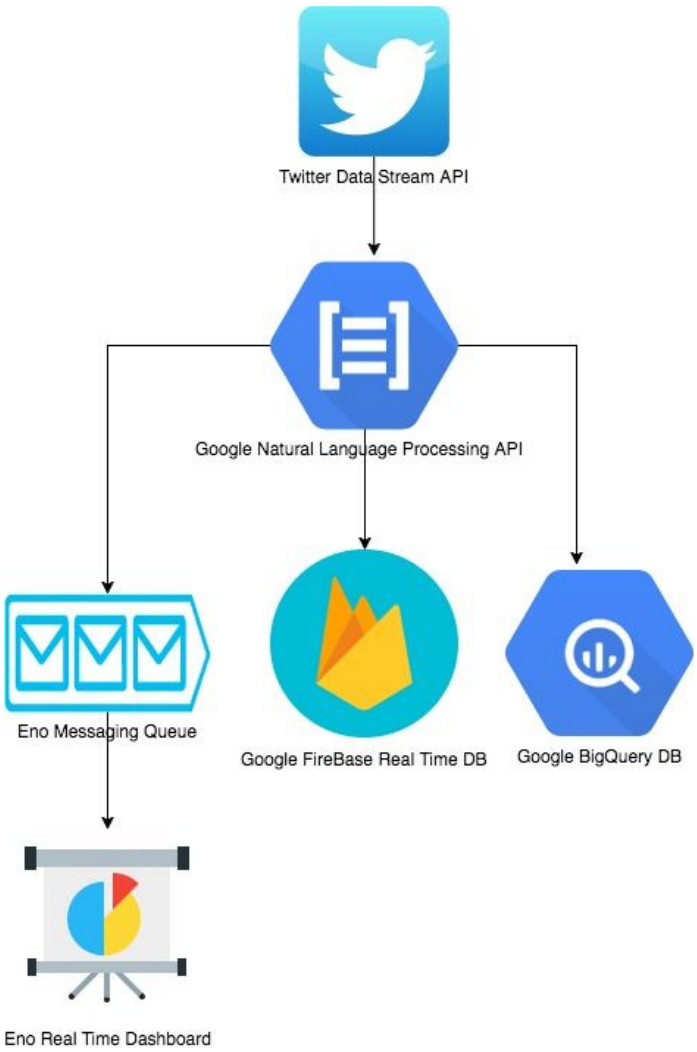
AI Sentiment Analysis for Apple Products

- Lei Pan 07/08/2018

I built this automated real time sentiment analysis system to explore better options of building data pipeline with automated real time data analytics functionality for Apple products. I tried out a lot of different big data technologies for this project such as Hortonworks products, MapR products, Microsoft Azure data products, and Google Cloud data products. For the final project, I chose Google AI and Google cloud products. The comparison of those technologies totally worth a separate discussion for itself. I will leave it to another article. For this one, I will only focus on explaining the automated analytics system I built for Apple products from both technical and business perspective.

Data Pipeline Architecture

There are couple components to form this data pipeline including data input stream, natural language processing engine, real time database storage, long term database storage, real time messaging queue, and real time dashboard.



1. Twitter real time stream API grabs data input stream to feed into this pipeline.

2. Real time data stream then gets processed in natural language processing engine.

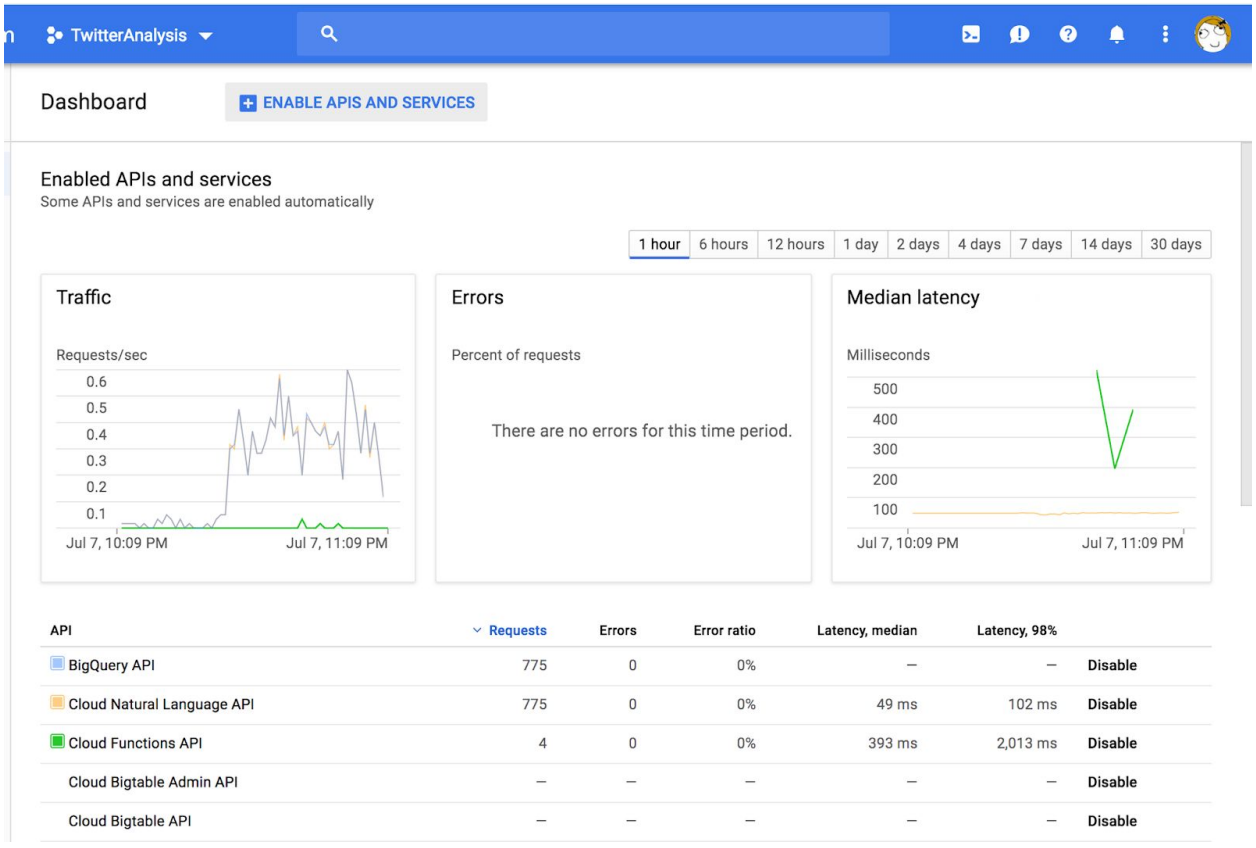
3. Processed data gets injected into Google firebase real time DB and Google BigQuery Table as well as published to Eon Messaging Queue.

4. Eon dashboard project subscribes data from Eon Messaging Queue and displays real time tweets analysis on the dashboard.

The key words that Twitter real time stream API filters on are "Apple watch, Apple music, Iphone 8, Iphone X, IOS, Iphone 7, macbook pro, Apple TV, Apple pay, imac, macOS High Sierra". Those key words are only for minimum viable product (MVP). I plan to use different keywords for a product line such as Apple music for future projects. You can refer to the **Roadmap** section to get more information.

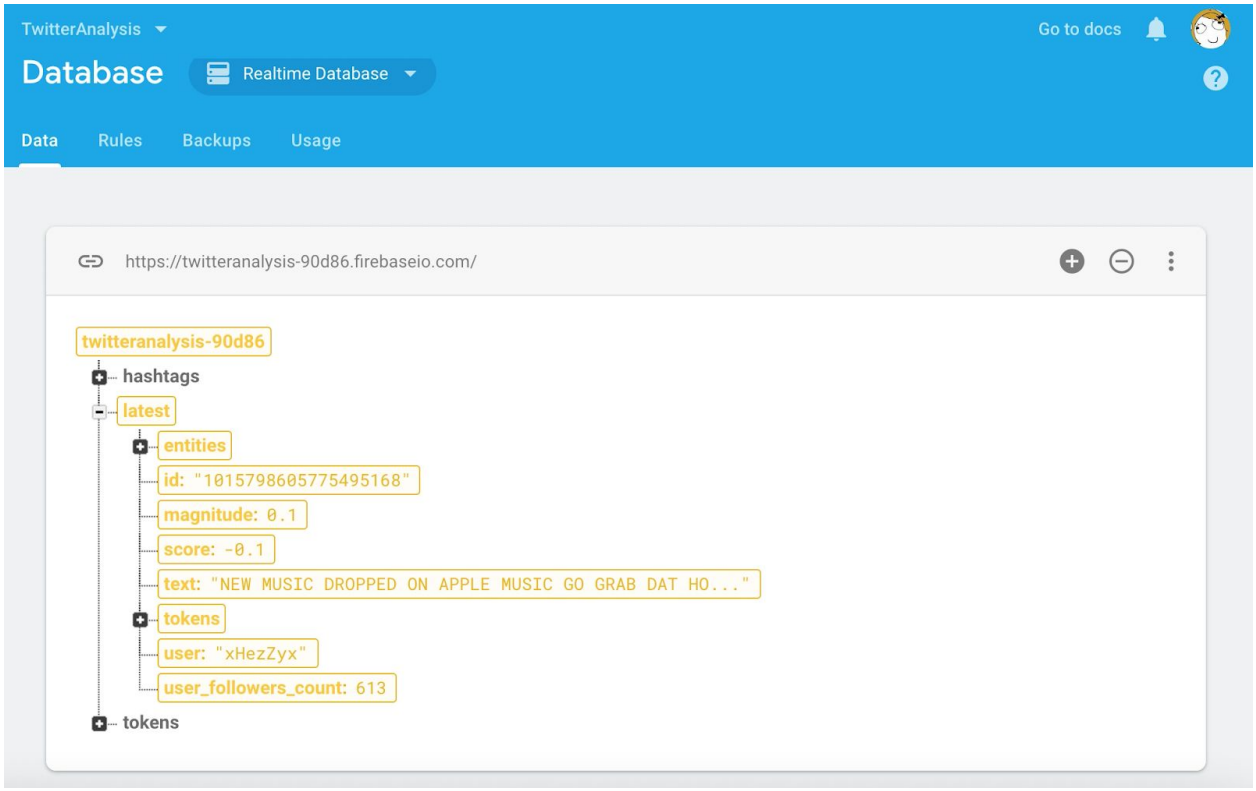
After twitter API pushes real time Apple products related tweets stream to Google Natural language processing API through the application I built, Google Natural language processing API starts real time analysis and parse them to tokens as well as give every single tweet a score and magnitude. The application then grabs the analyzed results from Natural language processing API and sends the data to FireBase real time database and Google BigQuery DB.

This is what Google Dashboard looked like when Google Natural language API and Google BigQuery API were processing the real time tweets data of Apple products.



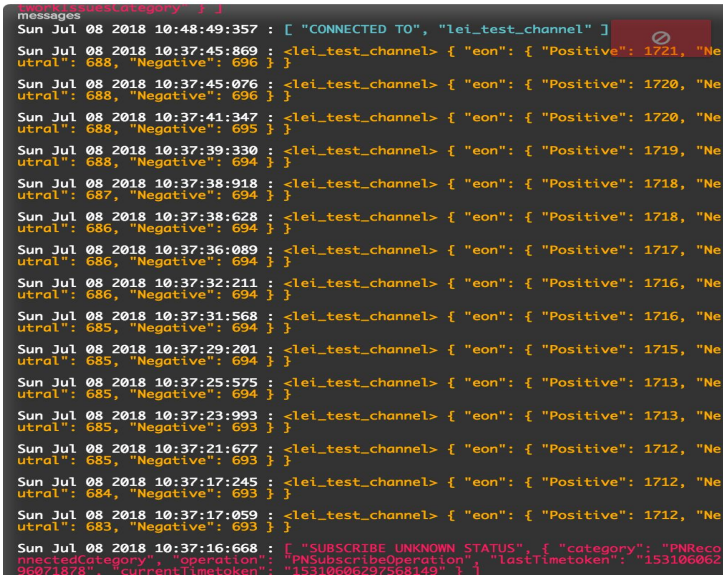
Google Natural Lan API + BigQuery API Processing the Data

This is how FireBase real time dashboard looked like when processing real time tweets data of Apple products which has already been analyzed by Google Natural language API. You can see magnitude, score, text, tokens associated with every tweet here.



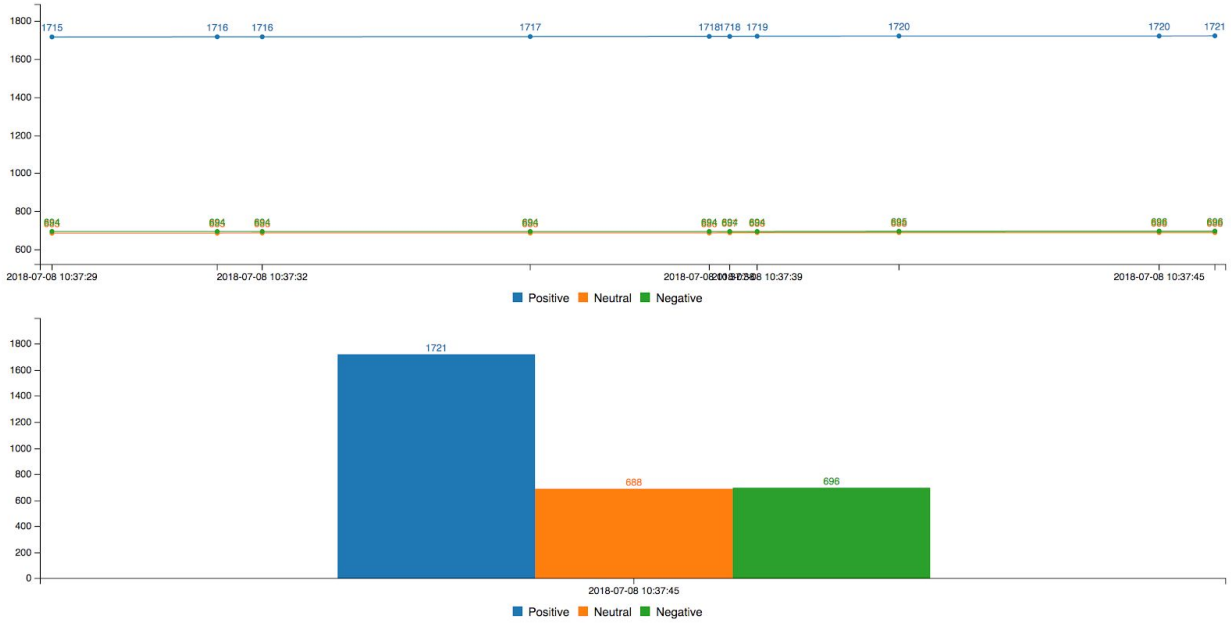
Before the application sends the processed data to Eon real time messaging Queue, I put some calculation there to calculate 3 values - counts of positive, neutral, and negative tweets. First, I multiplied score and magnitude. A positive score means the customer demonstrates a positive attitude in this tweet. A negative score means the customer demonstrates a negative attitude in this tweet. Zero means neutral attitude. Magnitude means to what degree that customer demonstrates that attitude. By multiplying these two values together, I get more accurate sentiments from customers. Second, I used the results of multiplication to compare with Zero to identify positive, neutral, and negative attitude. Third, I use 3 counters to count total number of positive, neutral, and negative tweets. Only those 3 values will be pushed into Eon Messaging Queue for every tweet.

You can check all the messages you send to Eon messaging queue at their console.

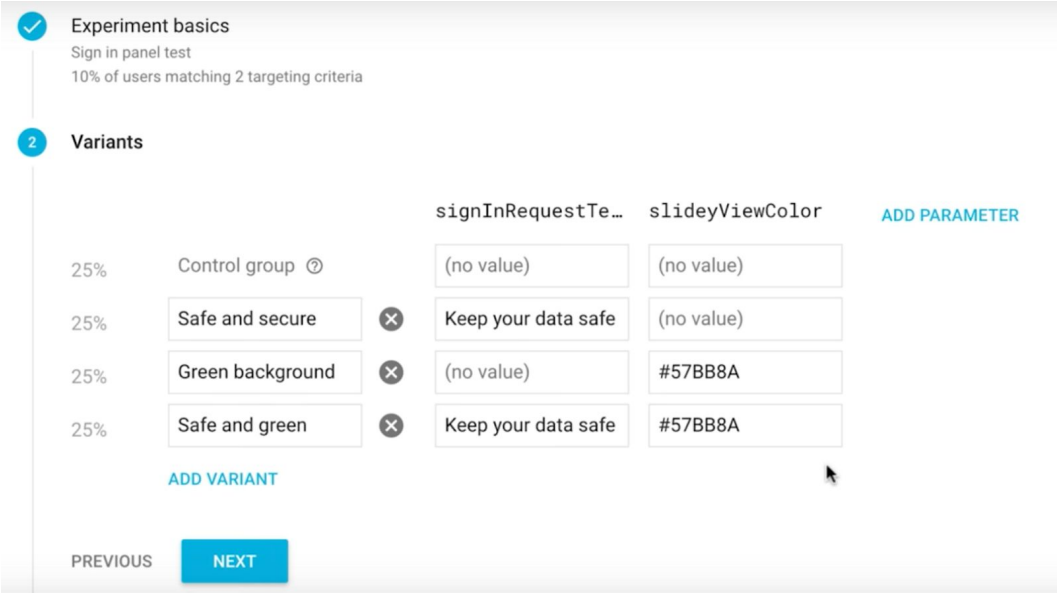


I used Eon open source project to build a simple dashboard which subscribes to the queue I created in Eon Messaging system. The dashboard shows real time changes of total counts of positive, neutral, and negative sentiments for Apple products. Here is the final Dashboard!! You can check this video to see real time dashboard changes. You can check out the project from my GitHub repo here

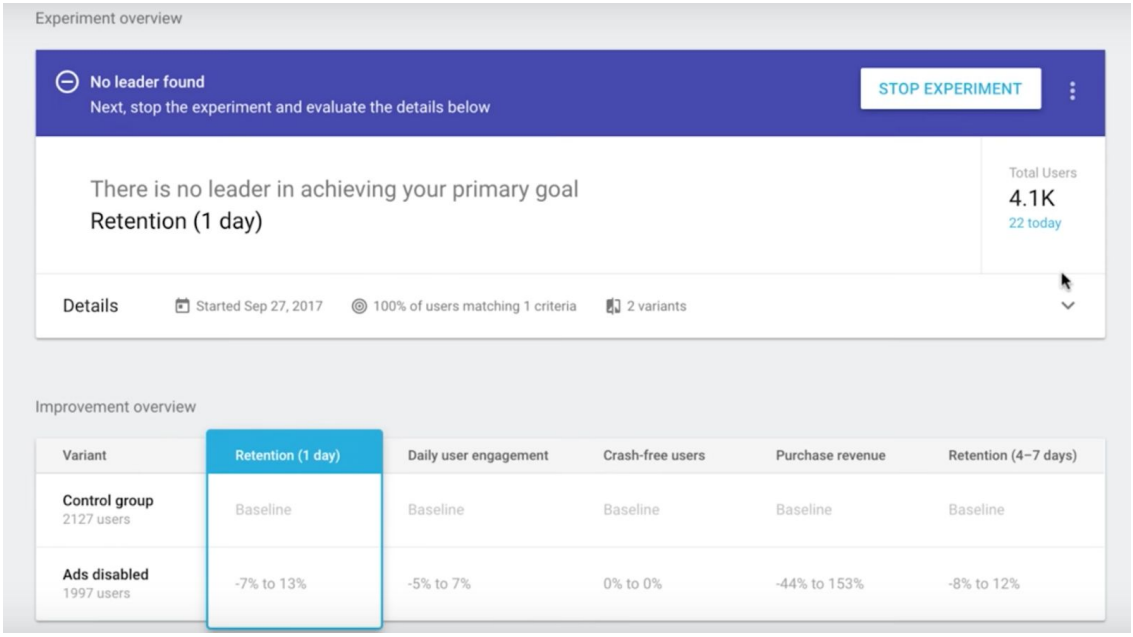
Dashboard - AI Sentiment Analysis for Apple Products



In addition, FireBase has some cool A/B Testing functions that you can use to design A/B testing experiment. We can customize this pipeline to grab user behavior data directly from Apple products such as Apple music app. Then we can push A/B testing easily by reusing this pipeline with firebase A/B testing tools. You can set up goals and multivalent test cells.



You can also measure A/B testing result with statistical analysis.



Roadmap

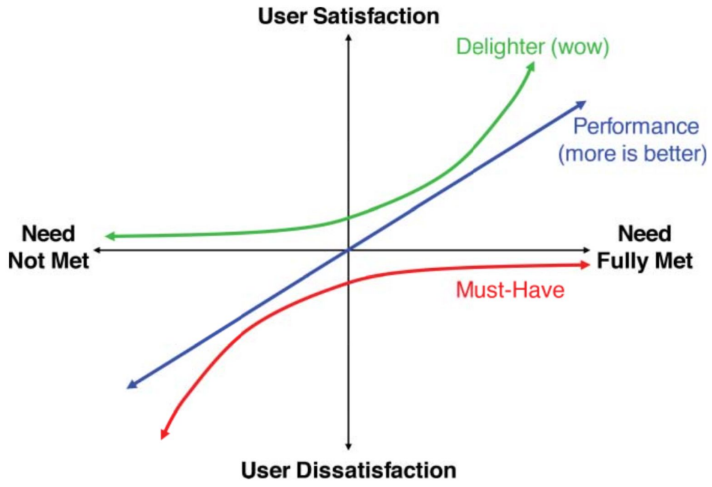
- This is the first phase of sentiment analysis project for Apple products. It filters on all Apple products to analyze people's attitude towards Apple products in general.
- In the second phase of the project, we can filter by a specific product line or specific Event such Apple Music or WWDC to identify customers' attitude towards a specific product line or event.
- In the third phase, we can use machine learning algorithms such as clustering algorithms to improve analytics to cluster issues that customers complain about and rank those issues for further examination. By adding this functionality, we can be more proactive and understand top product issues in real time.
- In the fourth phase, this system can be expanded to all social media platforms. We can grab information from all social media to get a larger population of random data.
- In the fifth phase, we can run real time analysis on competitors' products to identify strengths and weaknesses of their products in order to identify opportunities that we can tackle on as well as threats we need to be prepared for.
- In the sixth phase, we can expand this automated real time data analytics system to call center to analyze the reasons why people call and to predict call volumes and ways to reduce calls.
- In the seventh phase, we can use production log information as input data to find out production issues in our applications and predict issues in production.
- In the eighth phase, we can use this pipeline to grab user behavior data from Apple devices and apps to design A/B testing and recommendation system.

Prioritization

We can prioritize roadmap according to couple frameworks, experiences and intuition. The first framework that I love to use is **the importance versus satisfaction framework.**

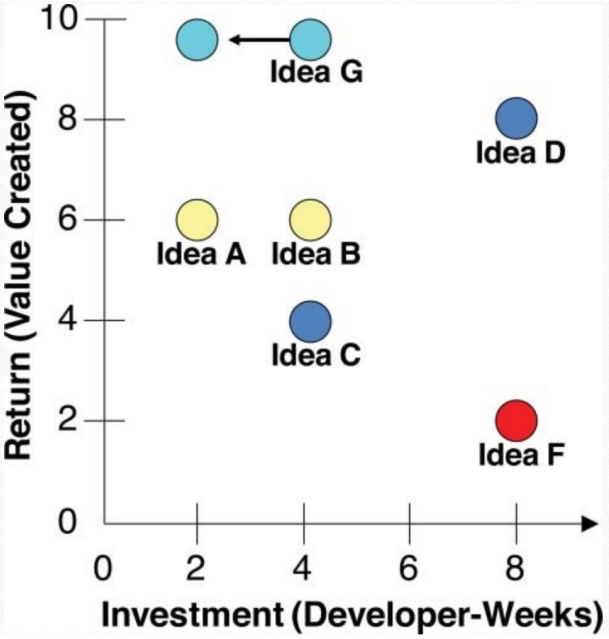
For example, we need to decide the priority between phase 5 and phase 6. Should we expand the functionalities to analyze competitors' product first or analyze call center issue first? Since this is an internal analytical system, our customer is Apple including Apple employees and stakeholders. We can first collect information regarding current solutions and situations on competitor analysis and call center. Does Apple have a good data analytical solution towards one of them or both of them? Which issue is more important to Apple? Then we can pick up more important issues which Apple is not currently satisfied with as the first priority.

The second framework that I like is **Kano model**



We can prioritize features based on if it's must-have, nice to have, or Delightful to have. We can make a comparison matrix with competitive products according to Kano model to see what strategies and what features worth investing to get competitive advantages.

Another idea to prioritize is **Return on Investment**



In this Return on Investment chart, we can see Idea A and G clearly have more ROI than any other ideas on the chart. Those two can be prioritized based on this chart.

Prioritization can't be solely relied on those frameworks. Experience and intuition are important as well. Business world is not perfect, we can't always get all the information that we need to make an absolutely correct decisions. We need to be able to make decisions faster with fewer

errors even with little or no information. Those decision making ability or intuition can be obtained through a combination of multiple ways such as work experience, building side projects, running side business, reading books and etc.

Measure the Success

How do we know if the data pipeline project works for Apple? We can measure the result from couple metrics. Some of my favorite KPIs are **AARRR (Acquisition, Activation, Retention, Referral, Revenue).**

Maybe at first glance, you don't think those metrics are applicable for an internal project. I challenge you to think about running an internal project as a real business. For example, we kick off the project by building it for one particular team or just a POC and then we socialize inside the company. We will gather acquisition and activation data as we sell it to more product lines and departments. Same thing is for retention and referral rates. We can use revenue metrics as well. For example, for the revenue data, we can measure the cost saving plus revenue of new business opportunities identified by this real time analytics system. We definitely can simplify the process of measurement and data collection and lower the cost. We don't want to consume too much resources while serving internal customers.

Conclusion

This pipeline has a lot of potentials to help Apple save cost and identify new product opportunities by analyzing customers' feelings towards Apple products. For MVP, we only need one engineer for a week to make it production ready. Let's work together to put MVP into production first and then make roadmap a reality!

Appendix

- You can check out the project from my GitHub repo here. If you want to know more, you can shoot me an email at: legatopan@gmail.com.
- You can check this video to see real time dashboard changes.
- You can check out my personal website here.
- Images in prioritization and measure the success sections are from this book - *The Lean Product Playbook: How to Innovate with Minimum Viable Products and Rapid Customer Feedback*
- I learned some of data pipeline ideas from *Sarah's Google I/O talk*.