Data Science Report

-Pawandeep Singh

(1) What is your one-sentence executive summary?

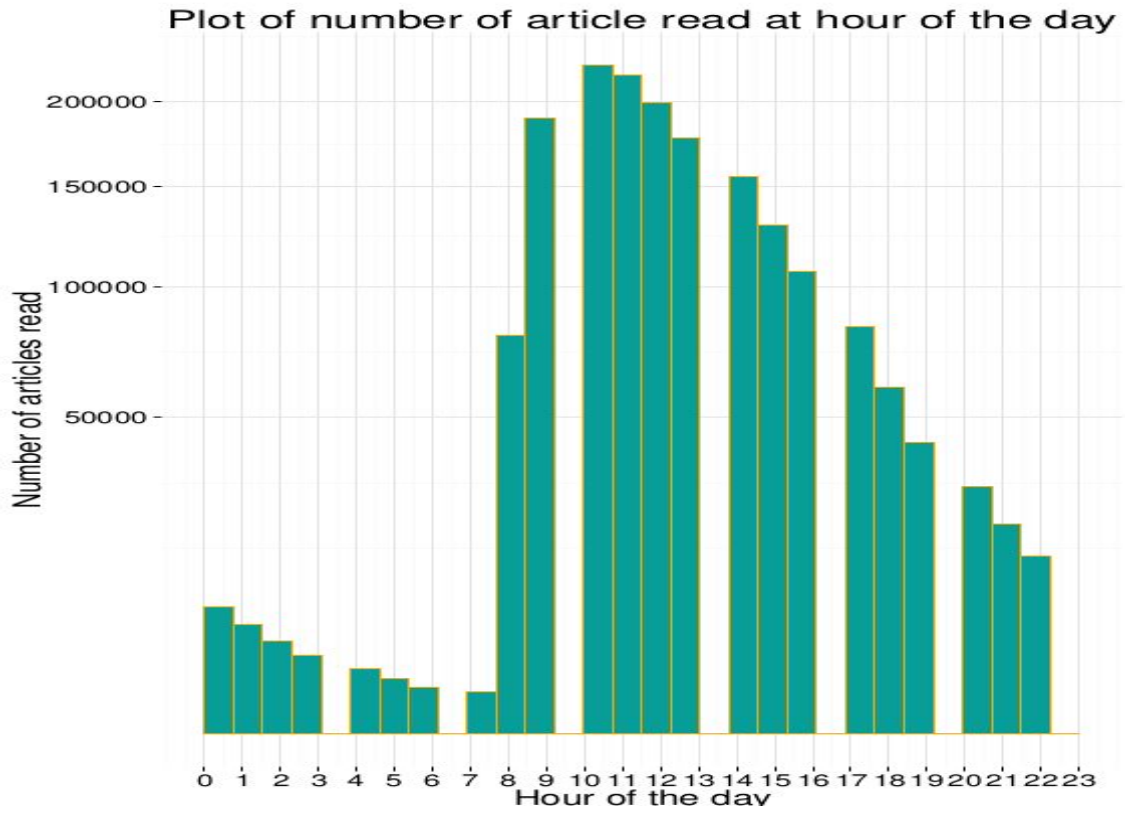
The likelihood of a user viewing an article depends on his history of topics read, type used and the hour of the day.

(2) What is your detailed assessment (for a technical audience)? Please quantify, use technical jargon.

The data are many patterns in the data which can be quantified using exploratory data analysis. The data tells us about the kind of topic that a user prefers or what type or medium he prefers it and at what time of the day is he most likely to read it.

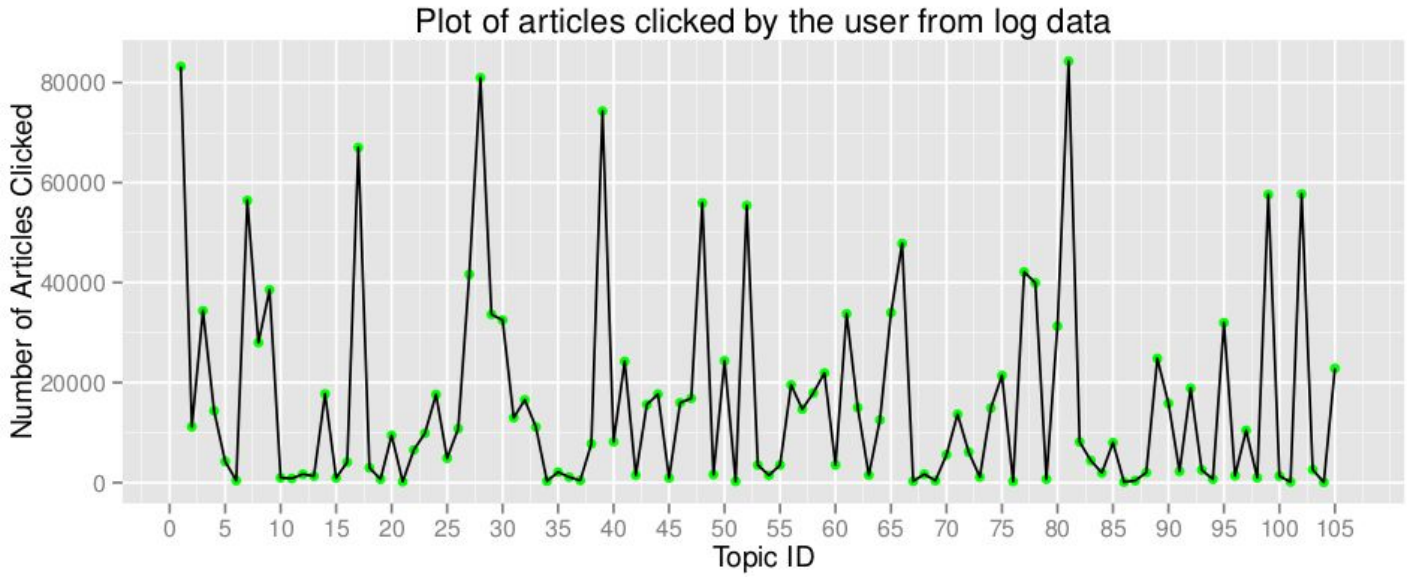
Around 58% of the articles were clicked or read during 10am:1pm time slot and and users were active in morning during 8am-9:am. This pattern shows that most of the people are reading our articles at working or on their way to work. Around 9% of the people read articles after 7:00pm. Therefore, sending articles to a user at a particular time might increase the likelihood of the user reading the article. This tells us and idea about the time user is using our app and we could also speculate where he might be reading it.

8:00am - 9:00am : Early morning commute
10:00am - 1:pm : At work or during small breaks at work
5:00pm- 7:00pm: Evening commute to home.

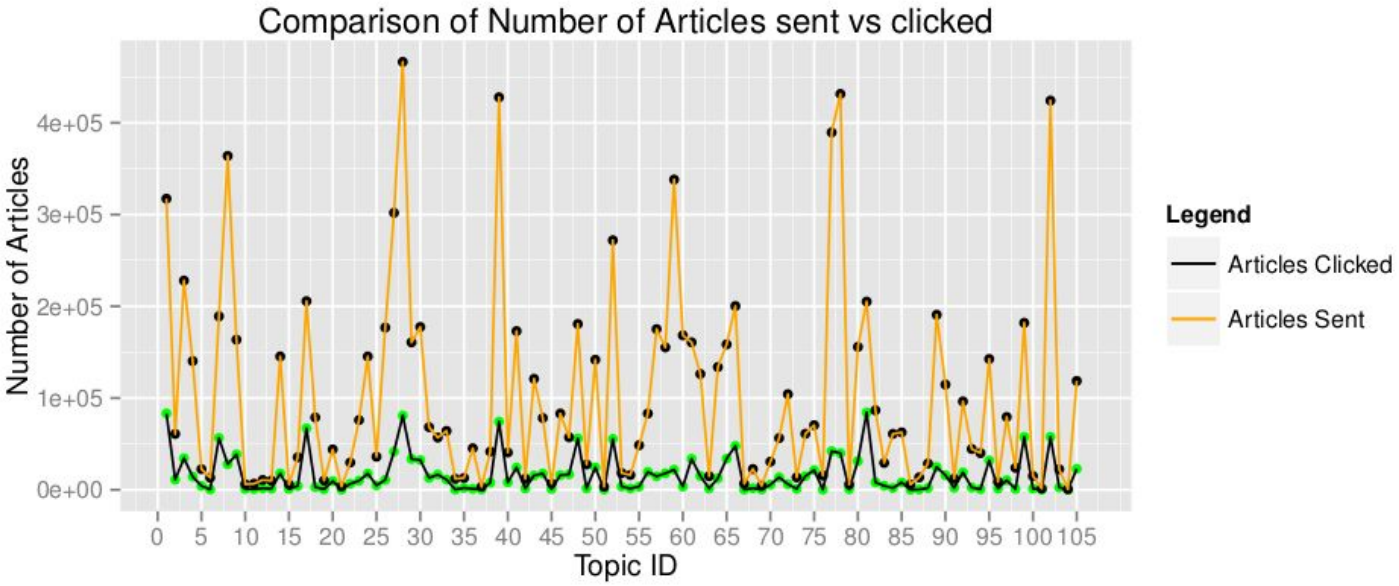


The plot for the time of the day when article was read is unimodal

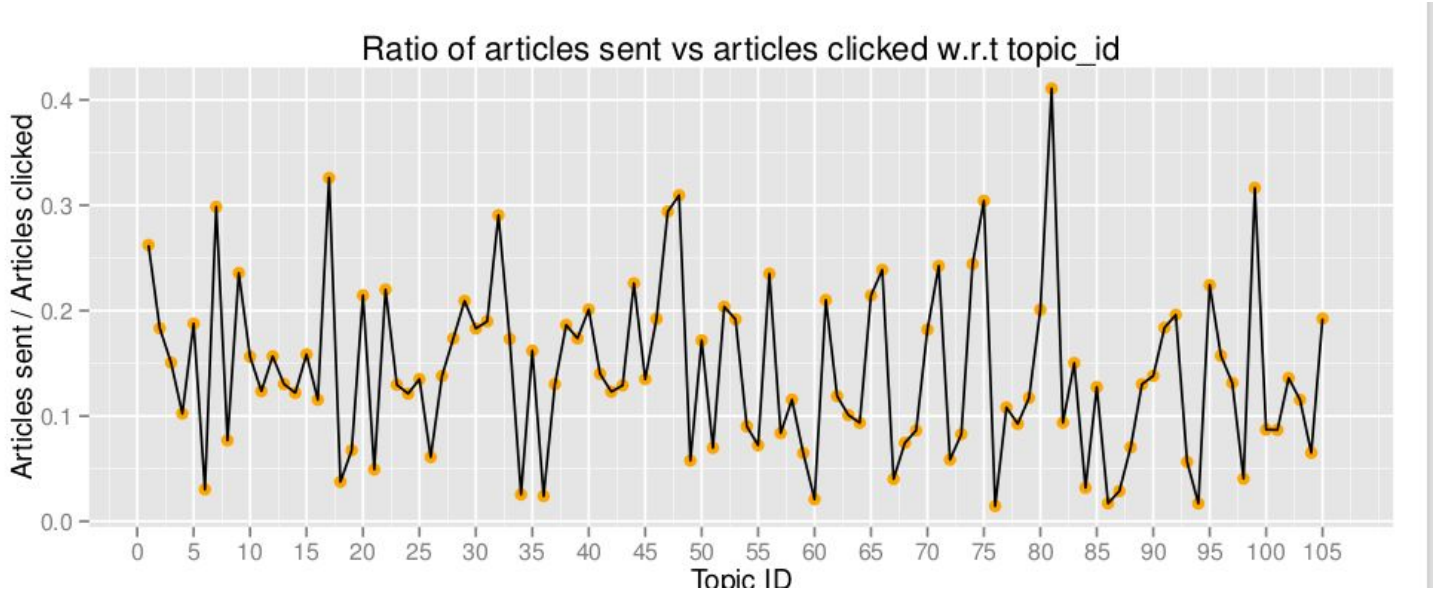
The other insight we get is in what kind of topics are our user interested in when we plot this data against the number of articles read by user we get a multimodal graph with various spikes for certain topic ids.
If we look at the granular level. we will see that 5 most viewed topics comprise the around 24% of the articles read. Among the most read are Advertising(1), Consumer Behavior(17), Entrepreneurship(28), Growth Hacking(39), Public Finance(81). This gives us the vital information into the kind of topics that people read.



But we should also take into context the number of articles we sent to the user. Doing this the pattern becomes clear that for example: the greater number of articles we sent to user for a particular topic the likelihood of user viewing that topic increases. But looking at this data without any context might bring us to a wrong conclusion.

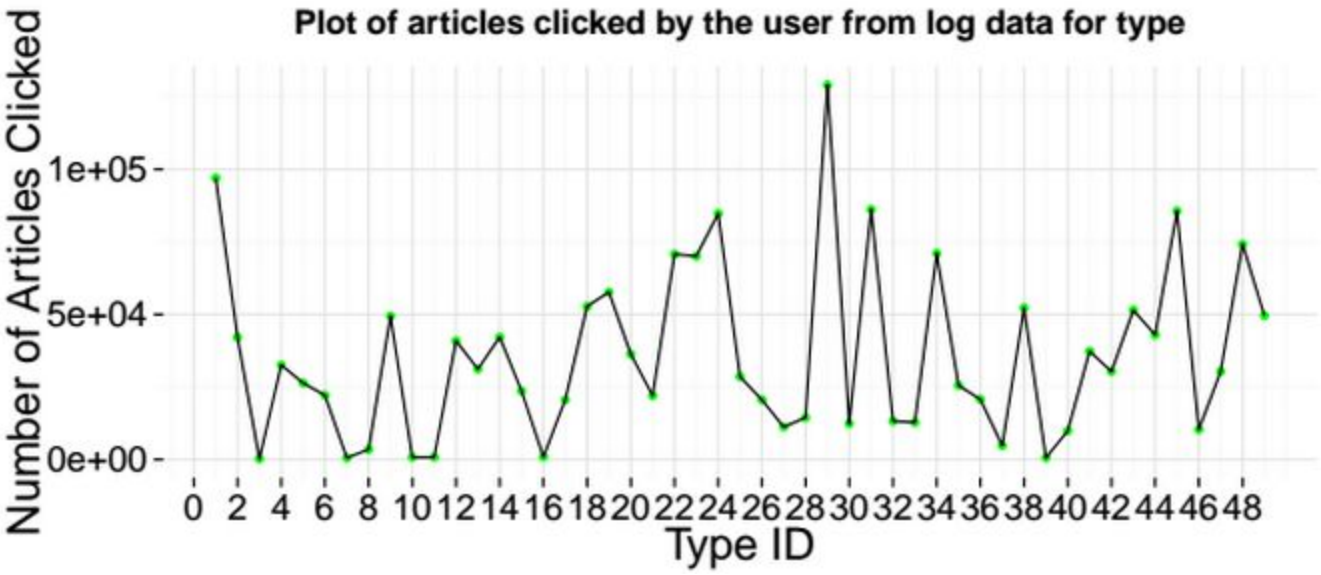


Instead, of doing we need to plot a ratio of articles read vs articles sent. Doing this we clearly see that some topics have 40% probability viewing even if we send them in smaller numbers. The other important insight we get from this plot is that we see that people are interested in these topics more that we are sending them in smaller number and we can increase their viewing likelihood by sending them similar or articles of the same topic.

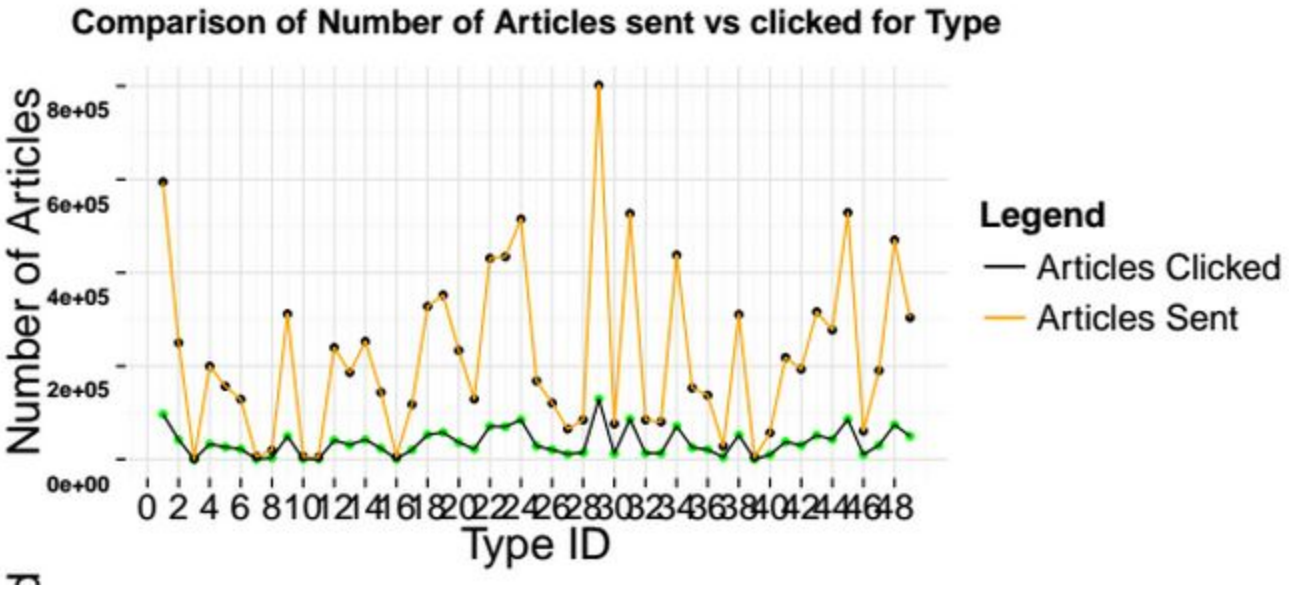


Further looking into the ratio of articles read/articles sent gives us the fraction of topics which were most likely to be read irrespective of the number of articles sent. A clear pattern emerges and this pattern will help us send the articles in greater number which users are more likely to read but are sent in smaller numbers. The ratios tells us the reading rate for a particular topic. 33% of the Advertising articles sent were read. 42% of public finance articles were read . 26% of advertising articles were read. 32% of Training articles were read.

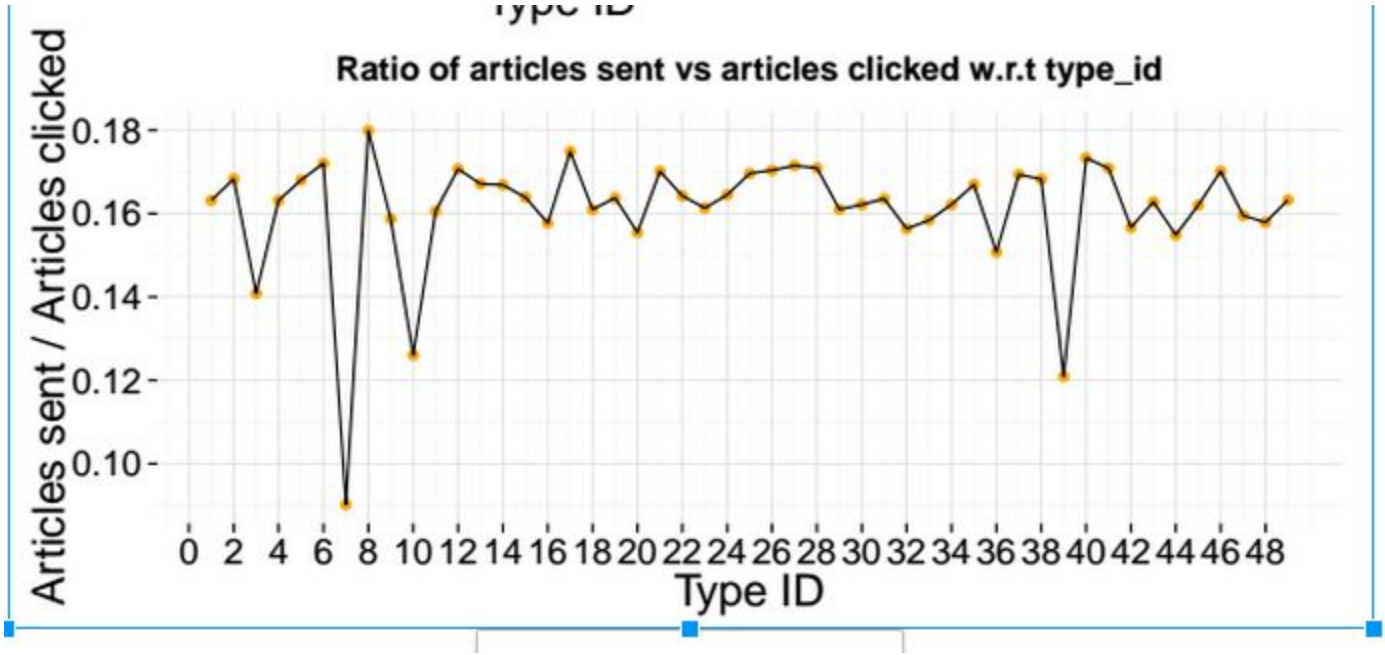
The other important insight we can get is from the type or medium in which the content is delivered. The top 8 ways user like to consume the content was in the following format comprising around 40% of the total types viewed.
For example: User prefered or like to consume their content in Blog_Post(1), Image(19), List(24), News(29), Opinion(31),
Podcast(34), Summary(45), White Paper(48).



Looking at context the number of articles sent for a particular type gives us a better understanding of the medium that people like to receive their content.



Further plotting the ratios of typeID for clicked to sent to received gives us the proportions and tell us on a relative scale which medium has better acceptance.

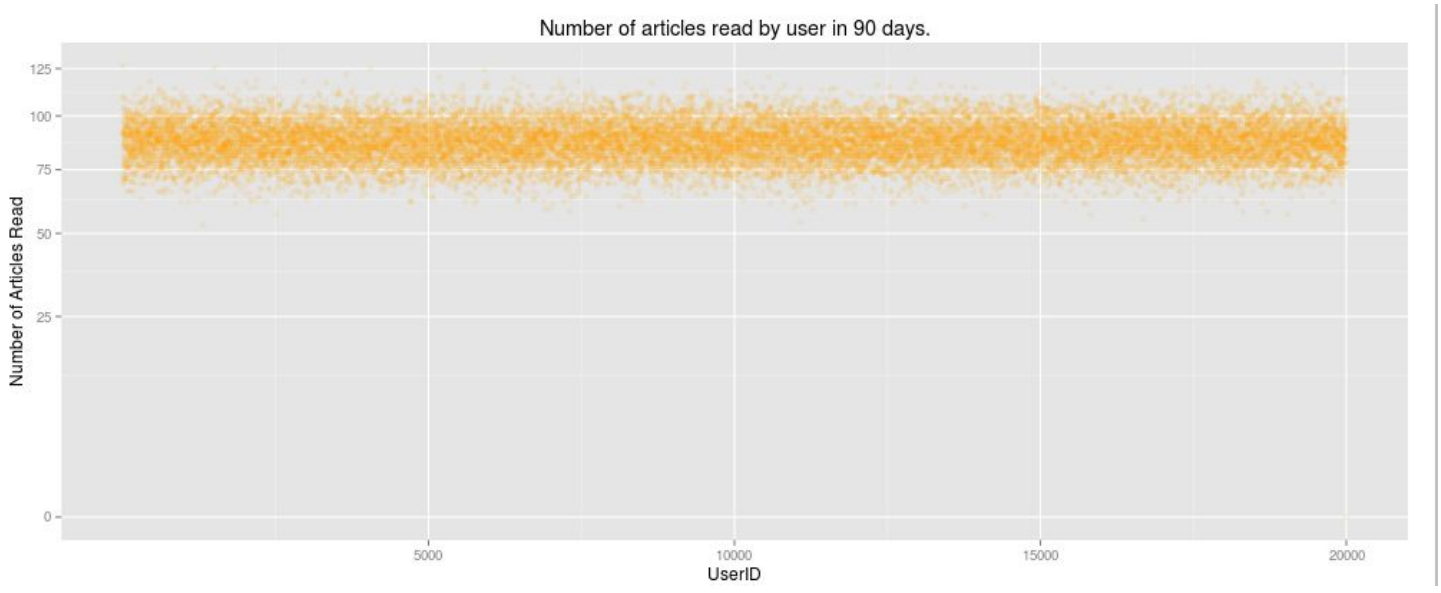


Hence knowing the time give us an idea about the surrounding our user is using the app. Further knowing what kind of topics are liked by our user and in what medium(type) will help us serve our user better and further improve their experience by understanding the behaviour of our users.

To further improve the user experience we can user recommendation system to suggest new topics to the users.

Note: The model we created is normalized and also takes into account users' non preference i.e the content we sent to the user but he choose not to view it and adjusts this feature in the model.

User Engagement : we have another metric to measure user engagement by plotting the number of article read by the user in 3 month period.



```
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.00   81.00   88.00   87.66   94.00  127.00
```

We can see that more than 25 % of the people read more than 94 articles per day and user seems pretty engaged reading almost 1 article per day. Therefore we can use this data to nudge users with low reading rate to read more articles.

The likelihood over a three months period is calculated by providing appropriate file according to the month split. and then run a test file for the 3 different combinations of access_log file and email_log file. Plotting the graphs will give us an idea of how the likelihood is changing. I ran the test on the values for the month of april and it gave decent likelihood. and show that likelihood in most cases was increasing over months and in rare case was decreasing or remaining stagnant. It is not convenient to plot values for all the users. Individual users likelihood can be easily tracked over a period of time.

(3) What tools did you use?

For Exploratory Data Analysis , I used R
Data Cleaning: Python Scripts
Data formatting: Python Scripts
SQL: sqlite3 for data joins and formatting back and fro to csv.
Modelling: Python.

I converted the files in the json format to be loaded into the mongoDB format. But since the database was not that large ,sqlite3 seemed fine. To truly scale a database as the number of user grows and getting high level summary and aggregation results , mongoDB seems a perfect candidate. Further it has good integration with python which will help us automate the data pipeline. The fields from the access log file were parsed as a dictionary using the python script and then loaded the data into the database so that it could be joined with the articles table to get articles which have a entry in the database and get the topic and type information using the join.

(4) What techniques did you try?

The first approach I took is that I converted it into a classification problem, and then to calculate the probability of the user seeing the link give a particular article id. I took the naive bayes approach to calculate the likelihood of a user viewing the link.

$$P(Seen|TopicID) = \frac{P(Seen) * P(TopicID \mid Seen)}{P(TopicID)}$$

$$P(Seen) = \frac{\text{Total number of links viewed by the user}}{\text{Total number of links emailed to the user}}$$

$$P(TopicID) = \frac{\text{Total number of links for this Topicid viewed}}{\text{Total number of links emailed for a TopicID}}$$

$$P(TopicID \mid Seen) = \frac{\text{Number of articles read by user for this TopicID}}{}$$

Total number of articles read by the user.

Since we are making use of likelihood we will not calculate the actual probability. Since the denominator is constant for the other class.

Then the final likelihood is calculated as follow.

P(Seen | TopicID, TypeID) = P(Seen|TopicID) * P(Seen | TypeID)

The above formula will give us the likelihood of seeing the article with a particular Topic ID and a TypeID

In this model we cannot capture the other latent feature I call Topic Throughput and Type Throughput which further gives us vital information about the preference of the user. Certain user For example I found will read all the articles for a given type. and some read a significant person of the topics sent to them.

The features are not used to calculate the likelihood but can be incorporated in the model as of now to give additional information but we can modify our likelihood algorithm to make a custom model capturing this feature.

I think this is the one of the most important hidden feature in the dataset.

The second approach I used was to create a feature, I call Topic throughput and Type Throughput.

Topic Throughput = Total number of links viewed by the user for a particular topic
Total number of links emailed to the user for a particular topic

Type Throughput = Total number of links viewed by the user for this particular type
Total number of links emailed to the user of this particular type

The above feature takes into account non-preference of a viewers for a particular topic id.

Other techniques such as content based filtering and recommending users similar topics could be easily implemented since we already have the data structure in place which keeps the top 10 topics similar to the user's preference in place.
and when an unseen article with a different topic id is presented it will take weighted score according to the ranked list of user recommended topics to generate likelihood.

There are other approaches use a hybrid of collaborative and content based filtering to refine the model further to generate recommendations.

Note: Both the content and collaborative filtering techniques have their shortcomings. Therefore using Topic modelling on the actual text and then adjusting this model based on signals from the reader. Further, modeling the reader preference and generating recommendations by similarity between preference and content to generate a rich UX.

Experiment: To model how user preference has changed over a period of time. I divided the data in the following way:

1. get all the data for the month of jan for each user.

2. get all the data for the month of jan and feb
3. model using all of the data.

There are two school of thoughts on how to split. Some people would split on each month and then see the behaviour. But I want to split this so as to capture the bias or user preference of the user from its first month of use. Therefore for any user you can see how the likelihood changed over a period of time. by running the test case against the split files.

Logistic Regression: I chose naive bayes since it performs better when you have an appropriate bias. In this case the bias was user clicking history which was important in predicting the likelihood of user viewing an article.

Although we can also easily calculate the likelihood of user reading a topic at a certain time. But this statistic is more important in telling us when is the best time to send them an email to the user.

Further we are trying to find the negative log likelihood, since likelihood for unseen is extremely small. We see whether the NLL is increasing or decreasing over the period of 3 months. Lesser the value of NLL the higher the likelihood of user clicking the link. Further using the NLL gives us a better perspective of the values.

(5) What three plots did you make to best explain the data?

The Three plots I made are:

1. Plots to determine when users are most likely to read the article by plotting the histogram to denote the number of articles read by the user an hour of a day.
2. line and frequency plot of the total number of articles viewed by users for a particular topicID. Then overlaying it with the number of articles sent to the user and a third overlay to denote the ratio of the articles viewed /articles sent.
3. line and frequency plot of the total number of articles viewed by users for a particular typeID. Then overlaying it with the number of articles sent to the user for each typeID and a third overlay to denote the ratio of the articles viewed /articles sent for each typeID.
4. Further I plotted the av. number of article read by user during the 90 day period.

(6) What is your commercial recommendation for business unit heads who are non-technical?

Another important feature according to me to the ability to save the article that a user likes in a read it later. This feature can further helps us understand the user and at the same time give user an important feature as what article the user is really interested in and we can make this a weighted feature.

Further, the user should be able to easily share his favorite articles with his/her friends. Their sharing history further gives us input of their preference and could be modeled.

From the data we know that what time our users are most active and hence we can send them some notifications during that time and not sending them notifications at odd hours which might cause them discomfort.

To really understand the user behaviour and deliver them a rich UX, we need to develop a personalized recommendations for each of our user. As google news and NYtimes do that.

Further sensor data from the mobile can be collected to analyze the environment of the user.

(7) What other data would you like to see about the platform? What questions would this additional data help you answer?

The other data that would be great will be actual content of the articles so that we can model topic similarity based on the content rather than based on self -generated tags and assign them tags in real time. Using topic modeling ,LDA can also help us in generating or tagging an article a mixture of two articles. Ex: an article having an intersection in news and art or politics and entertainment can be tagged which will help us widen the number of articles we can send to user based on his interest.

Further, it would be great to know how well the user is engaged with the app. For ex: whether he is sharing the link, or commenting on the link and so on. This will further provide information regarding how the user is using the app.

Given that it will be a social app, we can further do use his sharing history, saving history , comments and time spent on app or clicking a link. Among other features which are relatively easy to capture would dramatically enhance our model and help us understand our user behavior better.