Exploratory Analysis

Exploring transactional and notification data

The first step in this study is exploratory data analysis. The aim is to analyze how different features in the data set related to the other variable and study the user's behaviors with respect to engagement.

Begin by examining the transaction pattern of total 2,740,075 transactions history with 2,407,968 (8 7.88% of total) transactions status as COMPLETED of 18,766 users.

Table1: Transaction type wise total transactions frequency within transactions state.

Transactions Type	Total transactions	Completed Transactions	Declined Transactions	Pending Transaction	Reverted Transaction	Failed Transaction	Cancelled Transaction
CARD_PAYMENT	54%	85%	9%	1%	4%		
TRANSFER	18%	99%	1%	0.003%	0.4%	0.05%	0.3%
TOPUP	14%	78%		0.001%	11%	11%	
EXCHANGE	6%	100%			0.001%		
ATM	3%	83%	16%	0.4%	1%		
CASHBACK	3%	93%	5%	2%			
FEE	1%	100%					
CARD_REFUND	0.4%	100%	0.02%				
TAX	0.1%	99%		1%	1%		
REFUND	0.1%	100%					

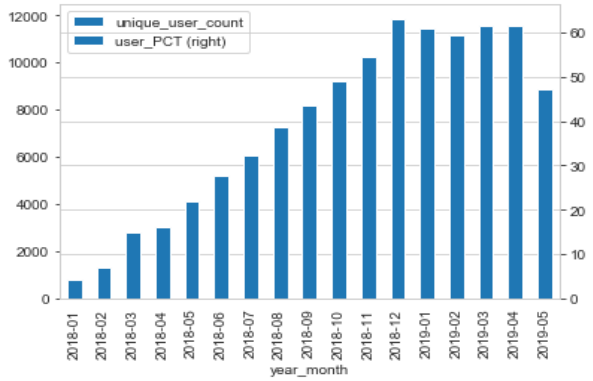
Around 86% of total transactions were with CARD_PAYMENT, TRANSFER and TOPUP methods. 99% of TRANSFER transactions were completed whereas around 11% of TOPUP transactions failed which is the highest percentage of failure as compared to other transaction types.

In addition, CARD_PAYMENT, TRANSFER, TOPUP, EXCHANGE and ATM transaction type indicates involvement of user activity and CASHBACK, FEE, CARD_REFUND, TAX, REFUND need not involve user's activity.

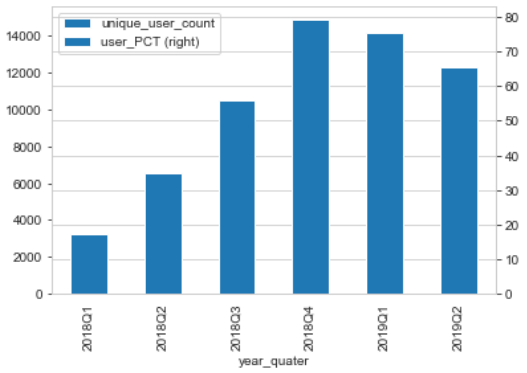
Active users

The best place to begin measuring user engagement is with the total number of active users i.e. any transaction/s made by a user. Let's start with **monthly active users** that will tell us the number of active users in a given month and **quarterly active users** i.e., the number of users active during the given quarter.

Plot 1a: Month wise total active users.



Plot 1b: Quarter wise total active users.



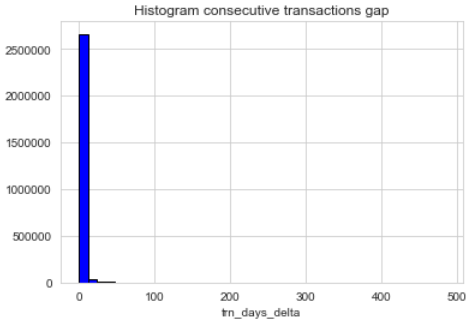
From Plot 1a, we can see that the monthly active users increase month-over-month and during Dec 2018 to Apr 2019, around 60% of total users were monthly active.

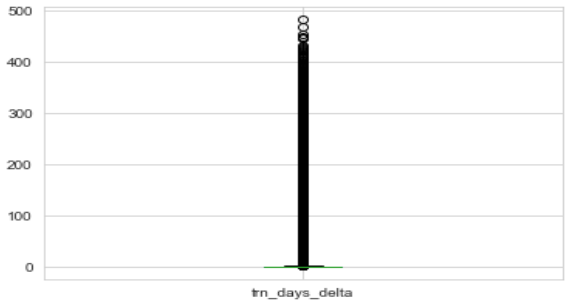
From Plot 1b, quarterly active users increase over quarter as explained in the plot, during the 4th quarter of 2018 and 1st quarter of 2019, around 75% of active users were quarterly active.

From above two plots lets compare quarter-1, 2019(~75% active users), and monthly plot Jan, Feb and March 2019(~60% active users), we can say that out of 75% of acive users 60% came in Jan and similarly in Feb and March. Now, it would be interesting to analyse the behaviour of transaction gap.

Behaviour of the transaction gap

Plot 2a: Transaction day's gap distribution of consecutive transactions. **Plot 2b:** Transaction day's gap boxplot.





From above plots, we can see that days gap between two consecutive transactions are highly skewed. 95% of time transactions have been made within 5 days while some of the transactions have been made after a year gap as well.

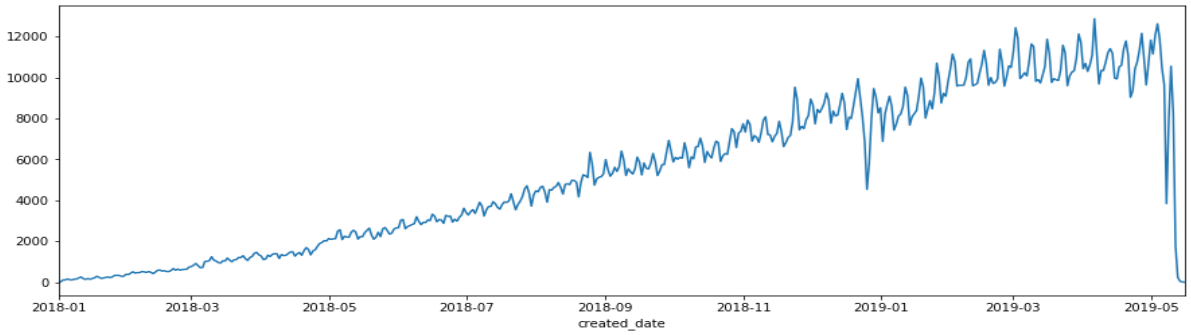
Table2: Percentile of day's gap between two consicutive transactions.

Percentile	10%	20%	30%	40%	50%	60%	70%	80%	90%	95%	99%	99.5%	100%
Days gap between two consecutive translation	0	0	0	0	0	0	1	1	2	5	26	42	483

Daily Transactions

Plot shows the number of transactions by day. We see a periodic nature at the end perhaps due to user's transactional gap frequency. We can also see that the number of transactions has grown over the year which could be due to active user.

Plot 3: Number of transactions per day



Target variable

One way to measure the user engagement is with notifications. Whenever notifications were sent to the users and users made transaction/s after that, it represents notifications were effective to engage those users. Though, the effectiveness of notifications could be measured with A/B testing, in this exercise we assume notifications as the key trigger for the users to engage. We don't have notifications 'seen' data or open/clicked/conversion information that would have been helpful to measure the engagement of users.

Also, from our previous EDA we saw that 95% of the time transactions were made in a gap of 5 days, so combining with the notifications, we can define target variable as whenever a notification was sent to a user and if she made a trascation within 5 days, she should be considered as an engaged user (engaged flag = 1) otherwise unengaged (engaged flag = 0).

Data preparation and Feature engineering

Notifications data were combined with transactional data, user data and device data and created feature with respect to every notification,

a. Transaction history with respect to notification

Feature	Description
trans_1db_not	number of transactions happened 1 day before the notification was sent
trans_3db_not	number of transactions happened 3 day before the notification was sent
trans_7db_not	number of transactions happened 7 day before the notification was sent
trans_15db_not	number of transactions happened 15 day before the notification was sent
trans_30db_not	number of transactions happened 30 day before the notification was sent
trans_45db_not	number of transactions happened 45 day before the notification was sent

b. Transaction type wise history with respect to notification
Example: for transaction type 'CARD_PAYMENT'

Feature	Description
trans_1db_card_pay_not	Total CARD PAYMENT transactions happened 1 day before the notfication was sent
trans_3db_card_pay_not	Total CARD PAYMENT transactions happened 3 day before the notfication was sent
trans_7db_card_pay_not	Total CARD PAYMENT transactions happened 7 day before the notfication was sent
trans_15db_card_pay_not	Total CARD PAYMENT transactions happened 15 day before the notfication was sent
trans_30db_card_pay_not	Total CARD PAYMENT transactions happened 30 day before the notfication was sent
trans_45db_card_pay_not	Total CARD PAYMENT transactions happened 45 day before the notfication was sent

c. Total transaction amount for 5/10 number of days before notification was sent.
d. Transactional state(OUTBOUND/INBOUND) count before 5/10 of notification created date
e. User's age, country
f. Referrals
g. Days since onboard

Model Development

Initial preprocessing

Due to data preparation computational cost, considered all notifications from 2019, Apr to May and all transaction of users during same time window.

One hot encoding has been done for categorical data. Train and test data splits with 80%-20% ratio.

1. **Baseline Logistic regression model**
   For the first attemp at prediction engage/unengage based on historical prepared data, applied logistic regression model.

Baselime model performance:

	Accuracy	Precision	Recall	AUROC
Train Data	85.5%	93.3%	74.6%	94.3%
Test Data	83.5%	92.4%	72.9%	90.6%

2. **Xgboost model**
   Applied Xgboost model on same set of variables used in logistic regression with max_depth =5 and n_estimator

Xgboost model performance:

	Accuracy	Precision	Recall	AUROC
Train Data	92.7%	98.1%	86.3%	97.8%
Test Data	85.4%	93.2%	76.3%	85.4%

Xgboost model result shows heigher deviation between training and test data accuracy that seems due to be overfitting.

Features importance:

1. 'num_contacts' = Number of contacts,
2. 'days_since_onboard' = days from onborad
3. age of user
4. 'tran_45db_not' = 45-days gap in transaction before notification

Above four features are very important for user's engagement prediction. (Please reffer appendix for full plot of feture impotance)

Test experiment

We can utilize standard A/B testing method to test the impact of business in reducing churn.

Step-1: Devides data into groups:

1. Experiment Group(Group A): In this group we performed the business action.
2. Control Group(Group B): This group experiences no change from the current setup.

Step-2:

Assumptions:

1. For Group A and Group B we can start with random split of 50%-50%. i.e., for every user there will be equal chance of fall in Group A or Group B. On 50% users we perform business action while on 50% will go as it is.
2. Consider significance level = 5%. This is also call Type-1 error or false positive.This means 5% of times we will detect the difference between Group A and Group B due to natural variation.

Step-3:

Metric: Consider evaluation metric

$\boldsymbol{Churn_{Rate}} = (\text{number of churn customer in next one week})/(\text{total number of customers})$

$$\boldsymbol{Group\ A} = \boldsymbol{Churn_{Rate_A}}$$

$$\boldsymbol{Group\ B} = \boldsymbol{Churn_{Rate_B}}$$

Step-4: Experiment setup with hypothesis:

H0(null hypothesis) : There is no difference in variants .i.e. $\boldsymbol{Churn_{Rate_A}} == \boldsymbol{Churn_{Rate_B}}$

H1(alternative hypothesis) : $\boldsymbol{Churn_{Rate_A}} > \boldsymbol{Churn_{Rate_B}}$

Step-5: Calculate t-statistics and calculate p-value

Step-6: Compare p-value with significance level

- If p-value < significance level, we can reject the null hypothesis and have evidence for the alternative
- If p-value >= significance level, we can reject the null hypothesis

Appendix:

1. Feature importance