User Retention Prediction Application

Akshara Narayana, Junjie Zhu, RaviTeja Kallepalli, Prabjyoth Obhi
Department of Software Engineering, San Jose´ State University
San Jose´, CA
akshara.narayana@sjsu.edu, junjie.zhu@sjsu.edu, raviteja.kallepalli@sjsu.edu, prabjyot.obhi@sjsu.edu

Abstract: **Nowadays, online users have become more interested in the quality of service (QoS) that organizations and applications can provide them. Services provided by different applications are not highly distinguished which increases competition between organizations to maintain and increase their QoS. Customer/User Relationship Management systems use machine-learning models to analyze customers' personal and behavioral data to give organization a competitive advantage by increasing user retention rate. Those models can predict users who are expected to churn and reasons of churn. Predictions are used to design targeted marketing plans and service offers. This paper tries to compare and analyze the performance of different BigQueryML models that are used for churn prediction problem based on the user behavior logs and user demographic information of users of Flood-It game application.**

Keywords: Google Cloud Platform, BigQueryML, Google Analytics 4 Data Schema, Logistic Regression Model, XGBoost Model, Predictive Analysis, Feature Engineering.

I. INTRODUCTION

The main objective of Customer Relationship Management as retaining existing customers is at least 5 to 20 times more cost effective than acquiring new ones depending on business domains. User retention includes all actions taken by organization to guarantee customer loyalty and reduce customer churn. User churn refers to customers moving to a competitive organization or service provider. Churn can be for better quality of service, offers and/or benefits. Churn rate is an important indicator that all organizations aim to minimize. For this sake, churn prediction is an integral part of proactive customer retention plan. Churn prediction includes using data mining and predictive analytical models in predicting the customers with high likelihood to churn/defect. These models analyze personal and behavioral customer data for tailored and customer-centric retention marketing campaigns and features for applications.

This application is a tool that helps people in the mobile application space to address the challenges of user engagement and user retention in a highly competitive business environment. The key to the success of mobile applications is a reliable and reusable way to simulate user retention rates. By being able to predict whether a user will be retained or churned, developers can take steps to increase retention through in-app features.

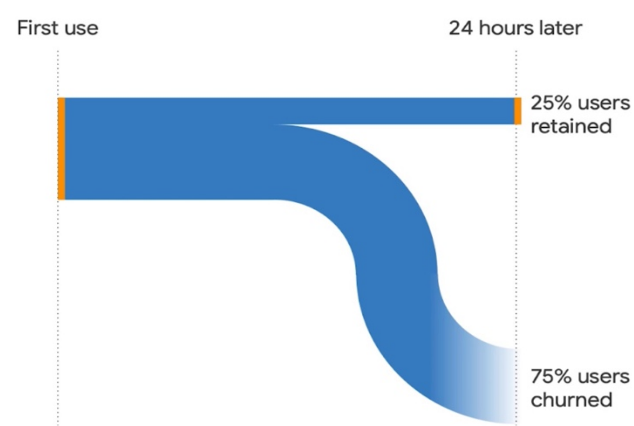

Figure 1: The user retention situation for a typical mobile app in a day (especially gaming app)

II. ARCHITECTURE

This has four major components:

- Data preprocessing in BigQuery.
- Construct and train retention prediction model with BigQueryML.
- Deploy finalized model on AI Platform.
- Generate data visualizations on Data Studio.

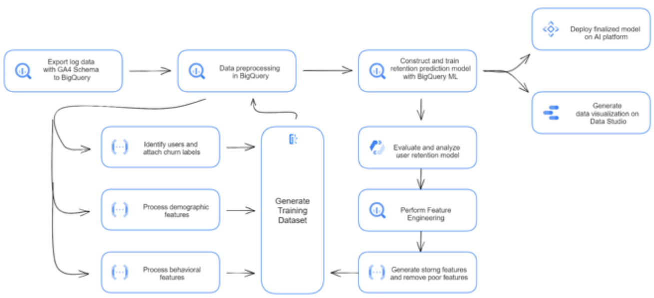

Figure 2: System architecture and workflow

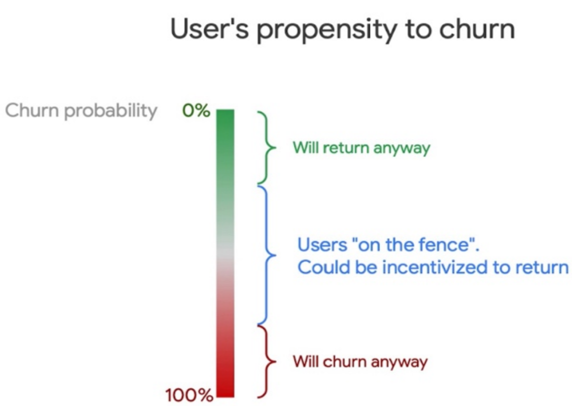

Figure 3: User churn and bounce scale

III. DATA ANALYSIS

As shown in the architecture diagram, the data logs used by the application follow the Google Analytics 4 schema. It is important for mobile application developers to make sure that their data adheres to the GA4 schema. This allows for standard formats in all applications.

Figure 4: Google Analytics 4 schema

The dataset we have considered for this project is an event-based dataset which means that each row is an event that had occurred from the application user's input. A record is created each time an action is taken by the user. This type of data provides insight into what actions a user has taken, the time between those events, and what other event parameters shown in Figure X can change on the developer side, so user retention can be amplified.

A. *Adding Labels to Data*

The data that was attained from the application did not have a prediction label. We had to assign labels based on few observations as stated below:

1. For the churned column, we assign churned=0 if the user performs an action

after 24 hours since their first action, else if their last action was made only within the first 24 hours, then we assign churned=1.

2. For the bounced column, we assign bounced=1 if the user`s last action was within the first ten minutes since their first interaction with the app, otherwise bounced is assigned as 0.

The dataset that shows the user labeling can be seen in Figure 5, and the amount of bounced and churned users along with their individual percentages can be seen in Figure 6.

```
FROM
  bqmlga4.user_labling
LIMIT
  10
```

	user_pseudo_id	user_first_engagement	user_last_engagement	month	julianday	dayofweek	ts_24hr_after_first_engagement	churned	bounced
0	AF2C7C5196C8AC879E4BCDCAAF68D5E4	1529773758309000	1532128993098006	6	174	7	1529860158309000	0	0
1	5217AB1A454DAED6243E1C9818BE6A20	1529861523949001	1535405434229018	6	175	1	1529947923949001	0	0
2	BCEC642620FB741AA4EE43E0FC6C7A05	1530996288345008	1533609643397024	7	188	7	1531082688345008	0	0
3	920DB84FCC0F4421650B9E257E33180B	1528957982564003	1538313563840019	6	165	5	1529044382564003	0	0
4	01B3F8A4E745CED2639D0FBDC2FEFC88	1532142288940007	1532142364913001	7	202	7	1532228688940007	1	1
5	7DBEEB7A2488FE1FFC0808686AC09606	1531332824513008	1535367784934014	7	192	4	1531419224513008	0	0
6	4789C778386485B99F4077B97DFE34E3	1528931183521002	1537040908310004	6	164	4	1529017583521002	0	0
7	9B18A1CBA96D52D05CFA7CB7C9011EDE	1531098901503002	1536093337848003	7	190	2	1531185301503002	0	0
8	1A6F117EC48594AA7A7B3A42A605415A	1532133434700006	1532133547991011	7	202	7	1532219834700006	1	1
9	E512079B53179DF9A608CF4ADE47DE9D	1529351778192007	1538596335194016	6	169	2	1529438178192007	0	0

Figure 5: Dataset with User Labels

```
FROM
  bqmlga4.user_bounced_and_churned
```

	bounced	churned	count_users
0	0	0	6148
1	0	1	2777
2	1	1	4663

Figure 6: Bounced vs Churned Totals

B. *Extracting Demographic Data for each User*

Extracting the demographic information of each user would allow developers and the machine learning models to gain insight into whether users on certain devices or countries are more likely to be churned. The dataset includes a variety of demographic data from the GA4 schema such as app_info, device, ecommerce, event_params, and geo.

Demographics help the model predict whether a user is likely to retain or churn.

We understand that user demographics are not always static, as users may move from one country to another. Therefore, for simplicity, we used the demographic information that GA4 provided to users when the app was first used. Processing demographics in this way allows each user to be represented as one record.

The demographic data per user can be observed on the image below:

```
FROM
  bqmlga4.user_demographics
```

	user_pseudo_id	country	operating_system	language
0	06DE385FB8FE1F9B3C866E5645866023	Norway	None	nb-no
1	096F51F57A27CD13F0FEB78BF778B50F	Uruguay	ANDROID	es-uy
2	1AD872561C6D6A28BA60CCA8F6376CD5	Guatemala	IOS	en-us
3	1CA5216B55152DE31E8C236E95CE7D50	South Korea	None	en-us
4	361F3570C32F1CE252B63BA587E7BDDC	Venezuela	ANDROID	es-es
...	...	...	...	...
13583	849938168F737FB0F6311F088956C871	United Kingdom	IOS	en-gb
13584	8DD47B40A0938BA8BD9FF0B34D83B28E	United Kingdom	IOS	en-gb
13585	CE141433511ABA7ED38F6C34445B91C7	United Kingdom	ANDROID	nl-nl
13586	E8205FDB6730D96FBA5804ADE5E6A1FA	United Kingdom	ANDROID	en-gb
13587	EC822F5ACEA06DC432D92DF2EE6CD787	United Kingdom	IOS	en-gb

Figure 7: Demographic data per user

C. *Extracting Behavioral Data for each User*

While examining the raw behavioral data of FloodIt, we found that the data circulated through multiple events for each user. This meant that the event data spanned multiple rows, and the behavioral data had to be aggregated and extracted for each user. Processing the data in this way allows one line of action for each unique user.

```
FROM
  bqmlga4.user_aggregate_behavior_new
LIMIT
  10
```

	user_pseudo_id	cnt_screen_view	cnt_user_engagement	cnt_level_start_quickplay	cnt_level_end_quickplay	cnt_post_score	cnt_level_complete_quickplay	cnt_
0	6A2F4E7AB1816D2F6EC741BC92141766	236	168	48	31	16	13	
1	7BCFAF1D468E309D866E9FEA7795FF85	20	13	1	0	3	0	
2	171871CA6AF9A891FB7B3E19CD144B2E	118	75	1	0	10	0	
3	43B8EEA09D7D62EC90F40D0E3716E960	28	20	0	0	2	0	
4	7C7C8524146AA5006BE260A041101DF3	4	4	0	0	0	0	
5	C59286E780962666D2A9C13A478542C8	4	4	0	0	0	0	
6	7CB9FDA17938289CE3019602888EEC58	2	2	0	0	0	0	
7	67E34D40DA8EB486DC847509A97762DA	13	9	5	1	0	0	
8	6A99D4E5FE96E6BAA99D7A52932F7E15	379	258	3	2	94	2	
9	E156684FAFCEDF58457F297C0E8F4BE5	30	19	1	0	3	0	

Figure 8: Behavioral Data

Data Correlation

Once the data was aggregated, we generated a heatmap as shown in Figure 9, to see how each attribute correlates with each other. Correlation plots are used to understand which variables are related to each other and the strength of this relationship. A correlation plot typically contains a number of numerical variables, with each variable represented by a column. The rows represent the relationship between each pair of variables. The values in the cells indicate the strength of the relationship, with positive values indicating a positive relationship and negative values indicating a negative relationship. The color-coding of the cells makes it easy to identify relationships between variables at a glance. Correlation heatmaps can be used to find both linear and nonlinear relationships between variables. We also combined multiple intermediate datasets to create a training dataset for the model.

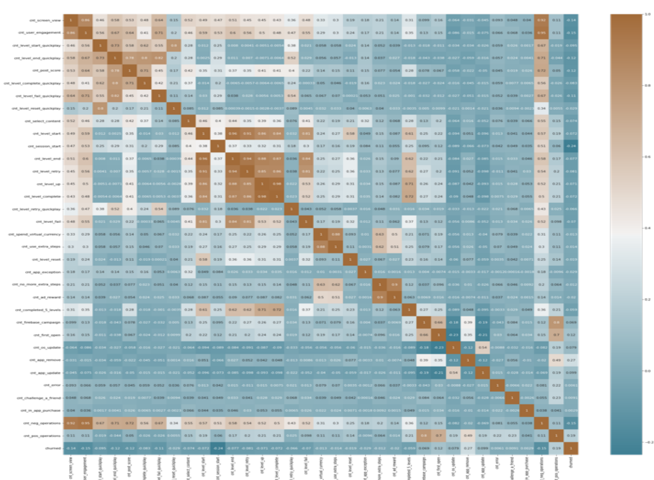

Figure 9: Behavioral Data

IV. TRAINING THE PROPENSITY MODEL WITH BIGQUERY ML

BigQuery ML democratizes machine learning by letting SQL practitioners build models using existing SQL tools and skills. BigQuery ML increases development speed by eliminating the need to move data.

Google BigQuery ML is fast to create, evaluate and execute various Machine Learning models easily using standard SQL queries. It offers many benefits over other Cloud Data Warehouses. Some of the benefits are stated below:

- It eliminates the need-to-know Python or any other language for managing Machine Learning models. Data Analysts with expertise in SQL only can now train models and make predictions.
- The data export involves many steps, and it’s a time-consuming process. Google BigQuery ML saves time and resources by letting users use Machine Learning models in Google BigQuery.
- It allows users to run Machine Learning models on large datasets within minutes as it uses computation resources of Google BigQuery Data Warehouse.

It features some automated Machine Learning models that reduce the workload to manipulate data manually. It saves time and allows users to quickly train and test models on the dataset.

With the training data we generated in previous steps, we now train machine learning models in SQL using BigQuery ML. We will train four different machine learning models to compare all possible predictions and refined the final prediction results.

Each of these models output a probability score between 0 and 1 of how likely the model prediction is based on the training data which we created. The model predicts whether the user will churn (1) or return (0) after 24 hours of the user's first engagement with the application.

A. LOGISTIC REGRESSION

Utilizing the Model explainability API in BigQueryML, we can further analyze the correlation between a feature and the actual prediction result as its prediction contribution. According to Table 1, we can see that many features are positively contributing to the model prediction, and the most significant two are user_last_engagement and cnt_in_app purchase. It can also be interrupted in real-world logic such that if a user continues to engage with the application, they are more likely to continue to use it due to habit/need. Also, if users are financially invested in the application, such as making in-app purchases, they are more likely to be retained due to the sunk cost effects.

Feature Contribution to User Retention

	feature	contribution ▼
1.	user_last_engagement	1.22
2.	cnt_in_app_purchase	0.76
3.	user_first_engagement	0.52
4.	julianday	0.52
5.	month	0.44
6.	cnt_app_remove	0.34
7.	cnt_ad_reward	0.25
8.	cnt_session_start	0.25
9.	country	0.15
10.	operating_system	0.13
11.	cnt_challenge_a_friend	0.11
12.	cnt_pos_operations	0.11
13.	language	0.1
14.	cnt_firebase_campaign	0.09

1 - 42 / 42

Table 1: Feature Importance for Predictions

After the logistic model was created and used, its accuracy needed to be evaluated. The evaluation metrics of the logistic regression model can be observed in Table 2. Figure 10 shows AUC-ROC Curve.

Table 2: Evaluation Metrics for Logistic Regression Model

precision	recall	accuracy	f1_score	log_loss
0.795	0.928	0.830	0.856	0.404

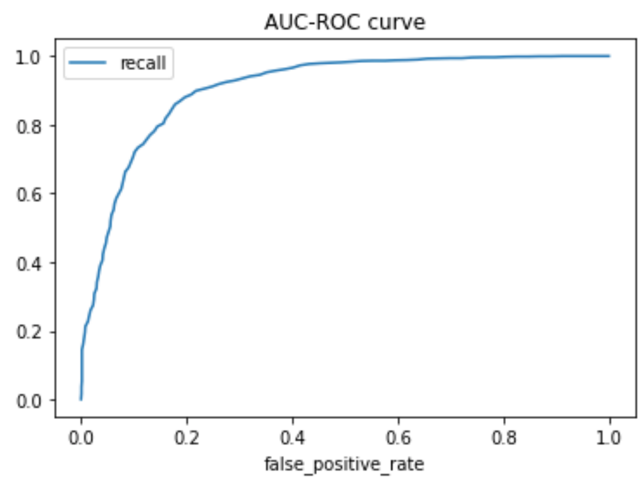

Figure 10: AUC-ROC Curve for Logistic Regression

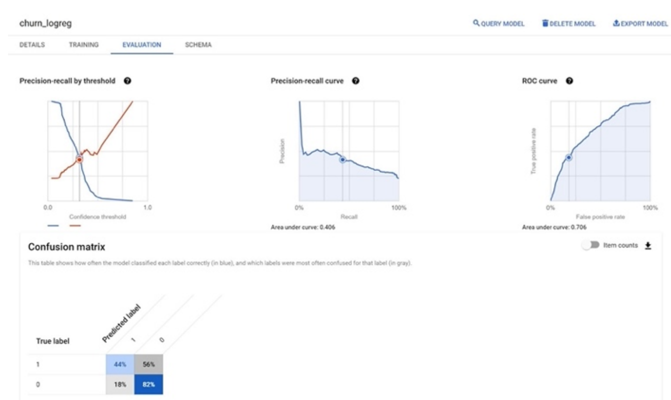

Figure 11: Evaluation metrics of logistic regression model with non-optimized training dataset

The evaluation metrics of the logistic regression model with non-optimized training dataset can be observed in figure 11.

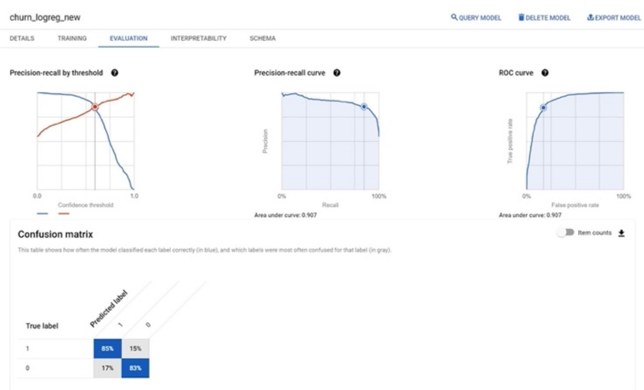

Figure 12: evaluation metrics of the logistic regression model with optimized training dataset

The evaluation metrics of the logistic regression model with optimized training dataset (feature engineering) can be observed in figure 12.

B. XGBOOST MODEL

After the initial logistic regression model was completed, we decided that it was possible to perform feature engineering on the dataset by getting rid of various attributes that were not pertinent in the prediction process.

Once the xgboost model was created and used, its accuracy needed to be evaluated and the following table 3 shows the evaluation metrics of the xgboot model.

Table 3: Evaluation Metrics for XGBoost Model

precision	recall	accuracy	f1_score	log_loss	roc_auc
0.932	0.993	0.956	0.962	0.149	0.991

Figure 13 shows the XGBoost model's AUC-ROC Curve.

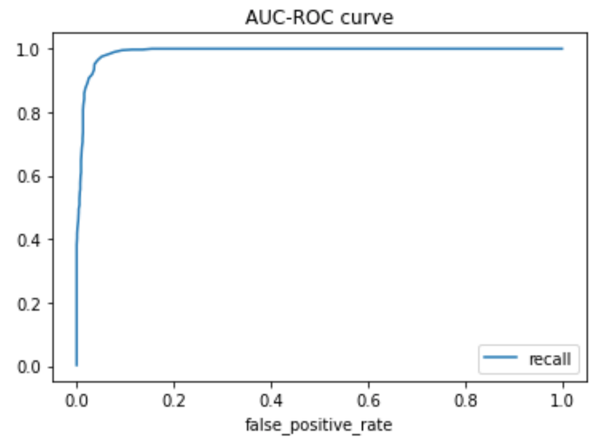

Figure 13 XGBoost model's AUC-ROC Curve

The evaluation metrics of the xgboost model with new training dataset can be observed in figure 14.

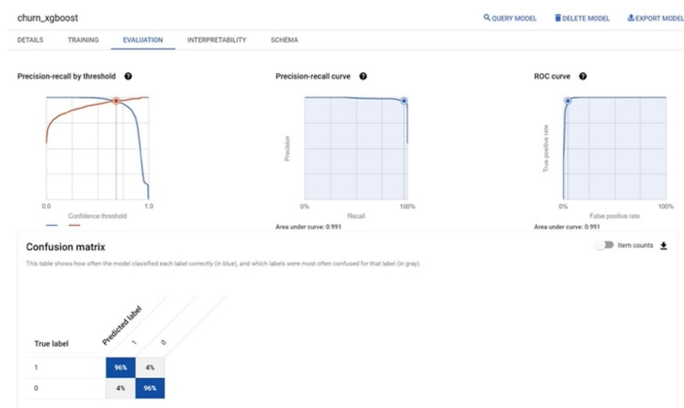

Figure 14 evaluation metrics of the xgboost model with new training dataset

V. RESULTS AND DATA VISUALIZATION

Model Performance overview

By meticulous evaluation and analysis, we can see that the performance of the first model with the non-optimized training dataset is very low (44% accuracy) and the model is good at predicting the retained user but not the churned users, so we continue to optimize our training dataset using feature engineering.

After feature engineering, it achieves an acceptable performance (84% accuracy) for making the prediction. Yet we were not satisfied with it and try to seek a breakthrough with model optimization and by exploring various other models.

After researching a list of state-of-art models which could be the best match to the nature of the training dataset and the prediction needs. We finalized our model with xgboost classifier since it gives us the best performance (96% accuracy) among all models we trained.

As seen in the image 15, the xgboost model's precision score is 0.9320, prediction accuracy is 0.9564, Recall and F1 score are 0.9934 and 0.9617 respectively. For any model, we expect the log loss to be as low as possible and our model has the least log loss score of 0.1493.

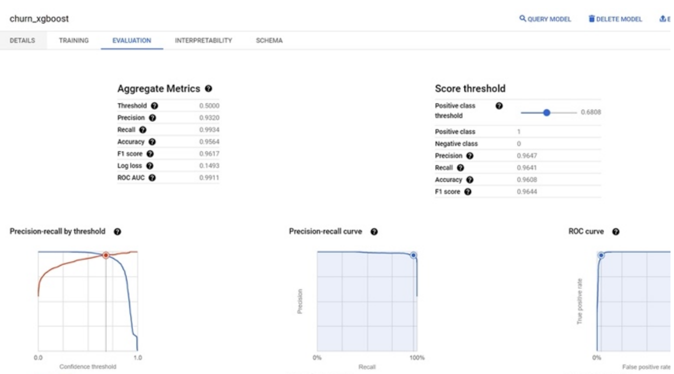

Figure 15: XGBoost Model evaluation metrics

Table for total numbers

total_actual_churn	**total_actual_retain**	**total_predict_churn**	**total_predict_retain**
7440	6148	7887	5701

Table for %s

total_accurate_predict	**total_inaccurate_predict**	**prediction_accuracy**
13061	527	**0.961**

Model visualization overview

For visualization of our data, evaluation metrics for the models and performance analysis, we used Google Data Studio. There are four sections as stated below:

1. Dashboard: This section consists of graphs and charts that represents our data by various categories such as, active users by day, active users by device model, active users by device category, active users by language, active users by location.

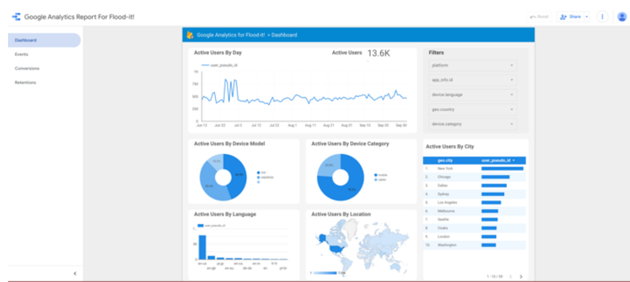

Figure 16: Google Analytics for Flood-I Dashboard

2. Events: This section shows how many events had been created under various categories such as, events by day, events by city, events by country.

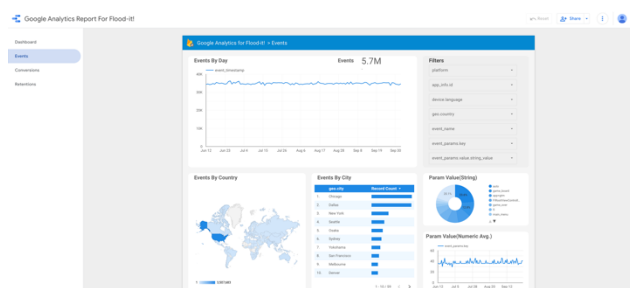

Figure 17: Google Analytics for Flood-It Events

3. Conversions: This section highlights on the conversions such as, total conversions, conversions by event name, conversion attribution.

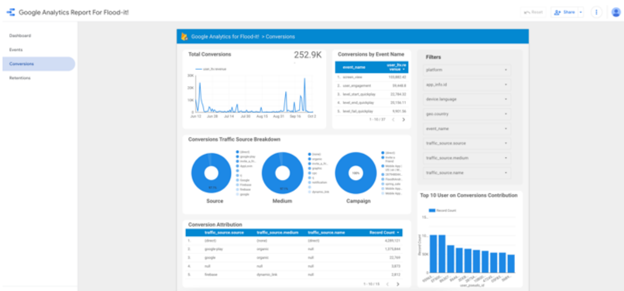

Figure 18: Google Analytics for Flood-It Conversions

4. Retentions: This section highpoints retention metrics such as, retained users by ML predictions, feature contribution to user retention, retained users by country, user categorization based on retention prediction.

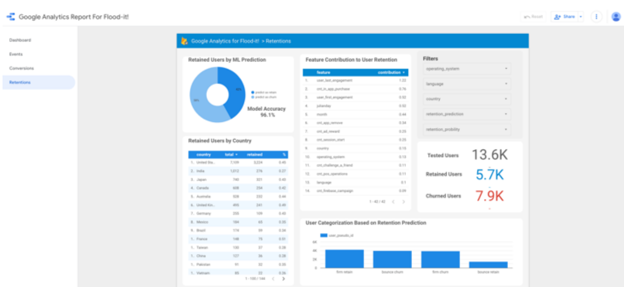

Figure 19: Google Analytics for Flood-It Retentions

VI. PROJECT IMPACT

Our project has many uses to the teams of data analysts, business analysts and product managers and those are briefly stated below:

- *To the data analysis team:*

1. Provided a cloud-based ML model for them to perform user retention prediction in real-time (via terminal or REST API).
2. Enable them to further analyze and summarize the characteristics of users in different retention categories, which could bring
a positive analytic impact to the company in the future.

- *To product managers:*

1. Provided an insightful data visualization report for them to better understand users' retention situation, and incentive features
that critical to user retention.
2. Enable them to optimize existing gaming features, and design better incentivized and customized retention tasks for users in
different retention categories.

- *To company:*

1. User viscosity will be directly boosted by transforming users from lower to higher retention category, which will bring
the significantly positive financial impact on the company in the long run.

VII. CLOUD HOSTED APPLICATION

We used Google Data Studio to generate reports and visualize our results.

https://datastudio.google.com/u/3/reporting/c38fc4ea-009f-4514-8602-010c4cc3fa98/page/Gg3

VIII. ACKNOWLEDGEMENT

We would like to express our gratitude to Professor Rakesh Ranjan for providing us with the opportunity to work on this project and for guiding us through it. We are confident that through working on the project, we have gained invaluable information and research expertise.

IX. REFERENCES

1. https://cloud.google.com/architecture/reference-patterns/overview
2. Predicting customer retention and profitability by using random forests and regression forests techniques by Dirk Van Den Poel.

3. https://www.sciencedirect.com/science/article/pii/S0957417405000965
4. A Review and Analysis of Churn Prediction Methods for Customer Retention by Ammar ahmed & Maheshwari linen.
5. https://cloud.google.com/bigquery-ml/docs/introduction#supported_models_in
6. https://cloud.google.com/bigquery-ml/docs/reference/standard-sql/bigqueryml-syntax-create-glm
7. https://cloud.google.com/bigquery-ml/docs/reference/standard-sql/bigqueryml-syntax-create-dnn-models
8. https://cloud.google.com/bigquery-ml/docs/reference/standard-sql/bigqueryml-syntax-create-automl
9. https://console.cloud.google.com/bigquery?p=firebase-public-project&d=analytics_153293282&t=events_20181003&page=table&authuser=3&project=user-retention-prediction-appl&supportedpurview=project
10. https://conf.slac.stanford.edu/xldb2019/sites/xldb2019.conf.slac.stanford.edu/files/Wed_10.55_Seyed_Umar_BigQueryML-XLDB2019.pdf
11. https://datastudio.google.com/u/3/reporting/c38fc4ea-009f-4514-8602-010c4cc3fa98/page/Y4oY
12. https://support.google.com/datastudio/answer/6283323?hl=en
13. https://www.youtube.com/watch?v=4lPmu8fsOHc
14. https://datastudio.google.com/gallery?category=marketing
15. https://medium.com/analytics-vidhya/confusion-matrix-accuracy-precision-recall-f1-score-ade299cf63cd
16. https://cloud.google.com/bigquery/docs/bigquery-storage-python-pandas#download_table_data_using_the_client_library