Company-X Assessment - Monica Iyer

Monica Iyer

02/01/2021

Aim

To understand behaviours that are most predictive of a new user starting and staying active on Product-X. The Growth team at Company-X is interested in month one retention, defined as whether a user remains active after signup. We will use the data sets to understand what factors are the best predictors of retention, and offer suggestions to operationalize these insights and help Company-X!

Analyze the data

Company-X has an interactive design platform, where designers can make prototypes for online products that can be reviewed and accessed by different account holders. We are given two datasets - users data and their corresponding events. Our aim is to predict the number of users that will be retained in the first month of activity. For now, lets go ahead and convert the data into dataframes.

	USER_ID <dbl>	SIGNUP_DATE <chr>	SIGNUP_CHANNEL <chr>	EMAIL_TYPE <chr>	JOB_TITLE <chr>	COUNTRY <chr>	M1_RETAINED <int>
1	7.81805e+17	2030-11-28 0:00	direct	Business	designer	Ukraine	1
2	7.75628e+17	2030-11-11 0:00	direct	Personal	other	China	1
3	7.86972e+17	2030-12-12 0:00	search	Business	other	Brazil	1
4	7.77574e+17	2030-11-16 0:00	direct	Personal	developer	Romania	0
5	7.91957e+17	2030-12-26 0:00	invites	Business	designer	Malaysia	1
6	7.74589e+17	2030-11-08 0:00	search	Personal	designer	Russia	1

6 rows

	RECEIVED_AT <chr>	USER_ID <dbl>	FILE_KEY <chr>	EVENT_NAME <chr>
1	2030-11-17 14:51	7.76508e+17	4gyJ8HUHkwM0vfkrgBOo3G	file_edited
2	2030-12-13 19:52	7.86575e+17	XAUWxpYSQ1mFMwJzY43zKq	file_edited
3	2030-12-16 19:21	7.88480e+17	u0FfIGyN4GerN0EHxuh7u8	file_edited
4	2030-11-19 13:53	7.70913e+17	ljp2t05FDJr5eJhuRjCq6l	file_edited
5	2030-12-16 3:37	7.78333e+17	35DJnOUjZp0p2kOzZEtYt9	file_edited
6	2030-12-05 15:25	7.79314e+17	KlH9hWCz9bGg3TRw9ux9Pl	file_edited

6 rows

Clean the data

The data doesn't seem to require extensive cleaning. I haven't found null values in key columns and the only redundant value would be `FILE_KEY` in `events_data` which will not be considered in this analysis, or prediction modelling for that matter.

```
#lets explore the data!

#Result 1
sqldf('SELECT JOB_TITLE, COUNT(JOB_TITLE) NUMBER_OF_USERS_RETAINED
       FROM users_df
       WHERE M1_RETAINED=1
       GROUP BY JOB_TITLE
       ORDER BY 2 DESC')
```

JOB_TITLE <chr>	NUMBER_OF_USERS_RETAINED <int>
developer	2011
designer	1822
other	825
project-manager	498
marketer	353

5 rows

```
#Result 2
sqldf('SELECT EVENT_NAME, COUNT(EVENT_NAME) EVENTS_COUNT
       FROM events_df
       GROUP BY EVENT_NAME
       ORDER BY 2 DESC')
```

EVENT_NAME <chr>	EVENTS_COUNT <int>
file_opened	759631
file_edited	154825
prototype_viewed	53783
file_created	29772
comment_created	18037

5 rows

```
#Result 3
sqldf('SELECT U.JOB_TITLE,E.EVENT_NAME HIGHEST_USED_EVENT
       FROM users_df U INNER JOIN
       (SELECT USER_ID, EVENT_NAME, COUNT(EVENT_NAME) EVENTS_COUNT
        FROM events_df
        GROUP BY EVENT_NAME) E
        ON U.USER_ID=E.USER_ID
        GROUP BY U.JOB_TITLE
        ORDER BY E.EVENTS_COUNT')
```

JOB_TITLE <chr>	HIGHEST_USED_EVENT <chr>
developer	comment_created
project-manager	file_created
designer	file_edited
other	file_opened

4 rows

```
#Result 4
sqldf('SELECT U.JOB_TITLE, SUM(E.EVENTS_COUNT) NUMBER_OF_EVENTS_BY_JOB
       FROM users_df U INNER JOIN
       (SELECT USER_ID, COUNT(EVENT_NAME) EVENTS_COUNT
       FROM events_df
       GROUP BY USER_ID) E
       ON U.USER_ID=E.USER_ID
       GROUP BY JOB_TITLE
       ORDER BY NUMBER_OF_EVENTS_BY_JOB DESC')
```

JOB_TITLE <chr>	NUMBER_OF_EVENTS_BY_JOB <int>
designer	512608
developer	368173
other	182258
project-manager	87780
marketer	64008

5 rows

```
temp_df <- data.frame(sqldf('SELECT U.USER_ID,E.NUMBER_OF_EVENTS
      FROM users_df U LEFT JOIN
      (SELECT USER_ID,COUNT(EVENT_NAME) NUMBER_OF_EVENTS
      FROM events_df
      GROUP BY USER_ID) E
      ON U.USER_ID=E.USER_ID'))
```

```
  sqldf('SELECT U.USER_ID,  E1.FIRST_DATE - E2.SECOND_DATE DIFF
 FROM users_df U
 INNER JOIN (SELECT USER_ID, MIN(RECEIVED_AT) FIRST_DATE FROM events_df WHERE EVENT_NAME="file_opened"
       GROUP BY USER_ID ORDER BY RECEIVED_AT) E1
       ON U.USER_ID=E1.USER_ID
 INNER JOIN (SELECT USER_ID, MIN(RECEIVED_AT) SECOND_DATE FROM events_df WHERE RECEIVED_AT NOT IN (SELECT MIN(REC
EIVED_AT) FROM events_df  WHERE EVENT_NAME="file_opened")  GROUP BY user_id ORDER BY RECEIVED_AT ) E2
       ON U.USER_ID=E2.USER_ID')
```

USER_ID <dbl>	DIFF <int>
6.61881e+17	0
6.61919e+17	0
6.61922e+17	0
6.61937e+17	0
6.61976e+17	0
6.61746e+17	0
6.61851e+17	0
6.62234e+17	0
6.62239e+17	0
6.62224e+17	0

1-10 of 10,000 rows Previous 1 2 3 4 5 6 ... 1000 Next

Explore the data

1. Explore the categories of the month-1 retained users Notice in **Result #1** that the number of users retained is highest in developers, and the lowest with marketers.The number of customers retained after the first month are about 30% of those who initially signed up, which is an imbalance in the data that will be addressed later. We see that the `other` category is the highest after developer and designer, however since there isn't much to be known about this category, we can analyze their events to understand their activity on Company-X, mainly to see if they differentiate themselves in any way from project-managers and marketers.
2. Explore events There's five types of actions and theire usability, in **Result #2**. It is key to the growth team to analyze the events most used by each category of users. **Result #3** shows that Designers edited files the most, while project managers created them, developers made comments and it seems like others mainly viewed the files (they could play the role of team members who view the design work and don't interact with it as much) - which is expected intuitively and its even better to see that the data reflects that. Lastly, we assess interactibility with the platform based on `JOB_TITLE` with **Result #4**, and the results are as expected. We see that designers use the platform the most followed by developers and other categories.
3. Add features to the data We can add the number of events per user, that indicates their extent of use of Company-X. Intuitively, the more a customer interacts with the platform, greater the chances of them coming back to it.
4. Balance of the data We notice that the proportion of retention is 30% and this represents an imbalance in the data. Just focussing on accuracy will not be the best choice in this case since its a poor measure of imbalanced data, so it will be best to focus on both recall and accuracy.

Preprocess the data

Create a final dataframe, that can be split into train and test sets. Replace NAs (if any) with zero, change the `SIGNUP_DATE` to just the month since all the dates fall in 2030 and the day of the month is irrelevant, and remove `USER_ID`.

```
#Create a final dataframe, that we can divide into training and test sets
#This dataframe contains the number of events corresponding to each user in users_df
final_user_df <- sqldf('SELECT DISTINCT * FROM users_df U inner join temp_df T using (USER_ID) where U.USER_ID=T.
USER_ID')
final_user_df[is.na(final_user_df)] <- 0 #replace NAs in user activity with 0
final_user_df$SIGNUP_DATE <- format(as.Date(final_user_df$SIGNUP_DATE), "%m") #retain only month in date
#remove USER_ID - does not add any useful information to the model
final_user_df <- final_user_df[, !(names(final_user_df) %in% c("USER_ID"))] #remove USER_ID from the data since i
ts irrelevant
```

Visualization

We can Visualize the first month retention over the year 2030.

```
final_user_df %>%
  filter(M1_RETAINED == 1) %>%
  group_by(SIGNUP_DATE) %>%
  summarise(n=n()) %>%
  plot_ly(x= ~SIGNUP_DATE, y = ~n, type='scatter', mode='lines')
```

Separate the categorical and continuous variables from variables that don't require any further pre-processing. Convert categorical variables to numeric factors and join all the variable back into the `final_user_df`. Once that's done, split into train and test sets and extract the test month 1 retain values `m1_retained_test` from the test set.We're ready to build our model!

```
final_user_df_cat <- final_user_df[,c('SIGNUP_CHANNEL', 'EMAIL_TYPE','JOB_TITLE')] #categorical variables
final_user_df_exclude <- final_user_df[, c('SIGNUP_DATE', 'NUMBER_OF_EVENTS', 'M1_RETAINED')] #variables to exclu
de from any conversion
final_user_df_cont <- final_user_df$COUNTRY # continuous variables

numeric_cat = apply(final_user_df_cat, 2, function(x) as.numeric(as.factor(x))) #convert categorical variables to
numerical factors

numeric_cat_df<- as.data.frame(numeric_cat)
cont_df <- as.data.frame(final_user_df_cont)
exclude_df <- as.data.frame(final_user_df_exclude)

#bind all dataframes after conversions to the final dataframe
final_user_df <- cbind(numeric_cat_df,exclude_df,cont_df)
colnames(final_user_df)[7] <- "COUNTRY"
final_user_df$M1_RETAINED <- as.factor(final_user_df$M1_RETAINED)

#Split final data into train and test data - 75%, 25%
trainIndex <- sample(1:nrow(final_user_df), size = round(0.8*nrow(final_user_df)), replace=FALSE)
train_df <- final_user_df[trainIndex,]
test_df <- final_user_df[-trainIndex,]

#remove M1_RETAINED from the test set
m1_retained_test <- test_df$M1_RETAINED
test_df <- test_df[, !(names(test_df) %in% c("M1_RETAINED"))]
summary(final_user_df)
```

```
##  SIGNUP_CHANNEL    EMAIL_TYPE      JOB_TITLE    SIGNUP_DATE
##  Min.   :1.000   Min.   :1.000   Min.   :1.00   Length:18310
##  1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.00   Class :character
##  Median :5.000   Median :3.000   Median :2.00   Mode  :character
##  Mean   :3.638   Mean   :2.356   Mean   :2.32
##  3rd Qu.:5.000   3rd Qu.:3.000   3rd Qu.:4.00
##  Max.   :8.000   Max.   :3.000   Max.   :5.00
##  NUMBER_OF_EVENTS  M1_RETAINED   COUNTRY
##  Min.   :   0.00   0:12803     Length:18310
##  1st Qu.:   5.00   1: 5507     Class :character
##  Median :  17.00               Mode  :character
##  Mean   :  66.25
##  3rd Qu.:  63.00
##  Max.   :6063.00
```

Predictive Model

In measuring retention, binary classifiers are very common. However, survival analysis is also applied to measure retention, which I would have opted to use with more time. For the purpose of this assessment, I have chosen to use the following models - Logistic, Decision Trees and Random Forest.

Feature Selection

Use `StepAIC` to assess which features are most valuable in building the model. Also make sure to include variables that are relevant to the knowledge base of customer retention, which may not appear in the `ANOVA` analysis.

```
#full model
full <- glm(M1_RETAINED ~ ., family='binomial', data=train_df)
#summary(full)

#stepwise, forward and backward AIC and anova checks
step <- stepAIC(full, trace=FALSE)
step$anova
```

Step <chr>	Df <dbl>	Deviance <dbl>	Resid. Df <dbl>	Resid. Dev <dbl>	AIC <dbl>
	NA	NA	14476	15285.11	15629.11
- JOB_TITLE	1	0.06304798	14477	15285.18	15627.18
- COUNTRY	156	310.96258374	14633	15596.14	15626.14

3 rows

```
backward <- stepAIC(full, direction="backward", trace=FALSE)
backward$anova
```

Step <chr>	Df <dbl>	Deviance <dbl>	Resid. Df <dbl>	Resid. Dev <dbl>	AIC <dbl>
	NA	NA	14476	15285.11	15629.11
- JOB_TITLE	1	0.06304798	14477	15285.18	15627.18
- COUNTRY	156	310.96258374	14633	15596.14	15626.14

3 rows

Logistic Regression

```
#GLM Model
 model_glm <- glm(M1_RETAINED ~ SIGNUP_CHANNEL + EMAIL_TYPE + NUMBER_OF_EVENTS,
                 data = train_df,
                 family = binomial(link='logit'))

 pred_glm <- predict(model_glm, newdata=test_df, type="response")
 prob_glm <- as.factor(ifelse(pred_glm>0.5,1,0))

#Evaluation Metrics
 result_glm <- confusionMatrix(data=prob_glm, m1_retained_test)
 precision_glm <- result_glm$byClass['Pos Pred Value']
 recall_glm <- result_glm$byClass['Sensitivity']
 f1_glm <- result_glm$byClass['F1']
```

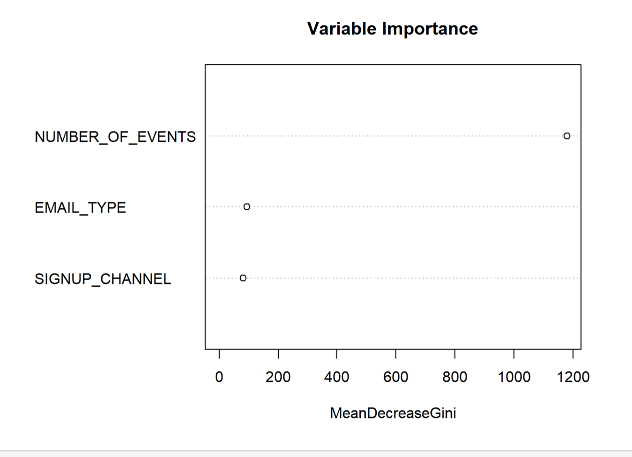
Decision Trees

```
#Decision Tree
 model_tree <- rpart(M1_RETAINED ~ SIGNUP_CHANNEL + EMAIL_TYPE + NUMBER_OF_EVENTS,
                    data=train_df,
                    method="class",
                    control= rpart.control(xval=10))

 rpart.plot(model_tree)

 #Evaluation Metrics
 pred_tree <- predict(model_tree, newdata=test_df, type="class")
 result_tree <- confusionMatrix(data=pred_tree, m1_retained_test)
 precision_tree <- result_tree$byClass['Pos Pred Value']
 recall_tree <- result_tree$byClass['Sensitivity']
 f1_tree <- result_tree$byClass['F1']

#Decision tree - variant with job_title

 model_treevar <- rpart(M1_RETAINED ~ SIGNUP_CHANNEL + EMAIL_TYPE + NUMBER_OF_EVENTS + JOB_TITLE,
                    data=train_df,
                    method="class",
                    control= rpart.control(xval=10))

 rpart.plot(model_treevar)
```

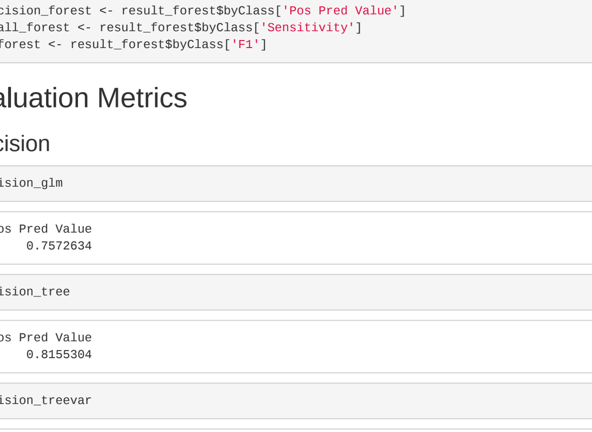

```
 #Evaluation Metrics
 pred_treevar<- predict(model_treevar, newdata=test_df, type="class")
 result_treevar <- confusionMatrix(data=pred_treevar, m1_retained_test)
 precision_treevar <- result_treevar$byClass['Pos Pred Value']
 recall_treevar <- result_treevar$byClass['Sensitivity']
 f1_treevar <- result_treevar$byClass['F1']
```

Random Forest

```
#Random Forest
model_forest <- randomForest(M1_RETAINED ~ SIGNUP_CHANNEL + EMAIL_TYPE + NUMBER_OF_EVENTS,
                             data = train_df,
                             ntree=200,
                             type="classification")
 plot(model_forest)
```




```
#Evaluation Metrics
 varImpPlot(model_forest, sort=T, main="Variable Importance")
```




```
 pred_forest <- predict(model_forest, newdata=test_df, type='class')
 result_forest <- confusionMatrix(data=pred_forest, m1_retained_test)
 precision_forest <- result_forest$byClass['Pos Pred Value']
 recall_forest <- result_forest$byClass['Sensitivity']
 f1_forest <- result_forest$byClass['F1']
```

Evaluation Metrics

Precision

```
precision_glm
```

```
## Pos Pred Value
##      0.7572634
```

```
precision_tree
```

```
## Pos Pred Value
##      0.8155304
```

```
precision_treevar
```

```
## Pos Pred Value
##      0.8155304
```

```
precision_forest
```

```
## Pos Pred Value
##       0.816886
```

Recall

```
recall_glm
```

```
## Sensitivity
##   0.9517079
```

```
recall_tree
```

```
## Sensitivity
##   0.8782882
```

```
recall_treevar
```

```
## Sensitivity
##   0.8782882
```

```
recall_forest
```

```
## Sensitivity
##   0.8775029
```

F1 Statistic

```
f1_glm
```

```
##        F1
## 0.8434238
```

```
f1_tree
```

```
##        F1
## 0.8457467
```

```
f1_treevar
```

```
##        F1
## 0.8457467
```

```
f1_forest
```

```
##        F1
## 0.8461102
```

Conclusions

Assessing from the Precision and Recall metrics, its best to choose the Decision Tree Model although the Random Forest model is a close second. With more time, I would tweak the models with different hyperparameters to see which one in fact performs better. For now, I will choose the Decision Trees Model.

Additionally, I made a model **model_treevar** that used `JOB_TITLE` as one of the variables to consider even though it wasn't a feature variable through stepwise model selection. Intuiitively, it made more sense that a designer is more likely to continue to on Company-X after the first month vs a marketer or project-designer. This model has slightly better recall than **model_tree** and similar precision.

Insights

1. It's interesting to note that the `SIGNUP_CHANNEL` plays an important role in determing whether the customer continues on Company-X after the first month. We can continue to build on the model to find which channels are likely to create customer 'stickiness'. The growth team can determine which marketing channels are valuable investments on financial and creative resources, and invest in analytics through CTRs and monitoring behaviour using Google Analytics for the chosen streams.
2. `EMAIL_TYPE` plays a key role in understanding client accounts. Intuitively, Businesses and Personal users are more likely to continue on the platform. Investing in marketing streams that target these specific users and combining that with a focus on building features that these streams are more likely to use is advantageous in retaining them and diversifying the product line!
3. By enhancing the functionality for key and subsidary users of Company-X accounts, we can better retain them. For instance, when including `JOB_TITLE` as a variate in `model_treevar`, we were able to deduce that it is significant to know what role the user plays in their corporation. By creating functionality for project-managers and developers alongside the key user (a designer), Company-X as a platform is enables more collaborative work and retains better in the long run.
4. Using `NUMBER_OF_EVENTS` turned out to be very useful. Intuitively, a customer who uses Company-X more is more likely to be retained. As mentioned previously, understanding the most used event by `JOB_TITLE` helps with adding features beneficial to specific use cases. Hence, by improving events/activities that keep customers coming back and fixing those that have not been used as much could help.

Errors and Assumptions

When adding `NUMBER_OF_EVENTS` to `final_user_df`, I encountered an issue with duplication that was caused during the use of both an inner join and left join. I solved this issue with by using a different SQL query when creating the dataframe, however instead of **18323** users, I was left with **18310** users. I believe this error occurred since R detected duplicate entries of `USER_ID` in one of the tables that maps to all duplicate values in the other table through `inner join` or `left join`. However, the `users_data` has no duplicate entries on inspecting the CSV. Since this accounts for less than 5% of the total users, I continued with building the predictive model using `final_user_df`.