LRSUPPORT-36395 Upgrade Timeline Prediction

Paula Lin

8/10/2020

Contents

Model A: Customer Upgrade Likelihood 1
Summary . . . 1
Data and Predictive Variables . . . 1
Exploratory Data Analysis . . . 3
Model A: Logistic Regression . . . 4
Compare Model Performance . . . 6
Test on 6-month subset . . . 7
Use Models to Predict for New Projects . . . 9

Model B: Customer Upgrade Timeline 10
Next steps: . . . 10
Model B: Linear Regression . . . 10
Model B Assessment on test_dataset . . . 12
Model B Assessment on 6 month subset . . . 13
Model B Predictions for new Projects predicted to Upgrade . . . 14

Conclusion: 16
Next Steps . . . 16

Model A: Customer Upgrade Likelihood

Summary

This is an analysis of customer upgrade likelihood, more specifically if Liferay Portal (6.x or below) customers will upgrade to Liferay DXP (7+). Two types of predictive models are used to predict if a customer (Project) will upgrade, a linear regression model and a logistic regression model. Linear regression is useful to show how different predictors contribute to a response. For example, what factors contribute to a customer Upgrading? How much do they contribute to the potential upgrade?Logistic regression is commonly used for modeling binary response data. Logistic regression models the probability of a success (e.g. Upgrade), not the expectation of the response variable, given the predicting variables.

This analysis will be used to help anticipate our customer's upgrade needs so our Customer support teams can reach out with upgrade resources to customers who are probably already considering an upgrade.

Data and Predictive Variables

- A Project is identified as *Upgraded* if they have created Portal (6.x or below) tickets and DXP (7+) tickets.

Note, a customer may have purchased a DXP offering, but not yet created any tickets on DXP yet. For the purposes of this analysis, they will be considered as not-yet upgraded.

One additional factor to take into account is Project Status.

If a Project is Closed and has only Portal tickets, we can conclude that they are a non-Upgrade Project. However, if a Project is Active and has only Portal tickets, they can still potentially *Upgrade*. In fact, our goal is to have every customer *Upgraded,* since Liferay Portal 6.2 EE is now in the Limited Support Phase. Therefore, any Active Project with only Portal tickets is the target in which we seek to predict if they will upgrade or not.

Consequently, our predictive models can only be built using Closed Projects (Upgraded or not-upgrade) and Active Upgraded Projects. Due to this unbalanced dataset, it is assumed that the model is skewed towards predicting *Upgrade* more than in reality, since we are unable to conclusively prove that an Active Project will NEVER upgrade, at this time. Note, future iterations may be able to separate Active Projects into those that can potentially upgrade and those that will NEVER upgrade.

The data is filtered to exclude any Projects closed prior to January 1, 2017. Liferay 7.0 was first released June 15, 2016.

Let's read in the data we will use to build and assess our predictive models.

```
rm(list=ls(all=TRUE))

# Set your working directory to where you have stored the data files
setwd("C:/Users/liferay/SQL practices/LRSUPPORT-36395 Estimate Cust Upgrade Timeline")

## Read the data in R
upgrade = read.csv("dataset.csv", header=TRUE, stringsAsFactors=TRUE)

# Set a seed for reproducibility
set.seed(1)

# Clean data
#
# Set NA to 0
upgrade[is.na(upgrade)] = 0

# Remove the irrelevant columns
clean_data = upgrade[-c(1)] #remove accountEntryId

# Convert the numerical categorical variables to predictors
clean_data$LPP = as.factor(clean_data$LPP)
clean_data$Max_Version = as.factor(clean_data$Max_Version)
clean_data$prev_Upgrade = as.factor(clean_data$prev_Upgrade)
clean_data$Zendesk = as.factor(clean_data$Zendesk)

summary(clean_data)
```

```
##     Feedback                    Industry   LPP     Max_Version
##  Min.   :  0.00   Agriculture       :  4   0:156   5.2:  8
##  1st Qu.:  1.75   Education/Research:111   1:812   6   : 26
##  Median :  6.00   Government        :190           6.1:173
##  Mean   : 14.77   Manufacturing     :105           6.2:761
##  3rd Qu.: 18.00   Null & Other      : 19
##  Max.   :234.00   Services          :539
##
##  prev_Upgrade    Sub_yrs       Support.Region   Time_Max_V
##  0:571        Min.   : 0.500   Hungary:356    Min.   :-3.479
##  1:397        1st Qu.: 3.200   US     :289    1st Qu.: 1.966
##                Median : 5.200   Spain  : 98    Median : 3.053
##                Mean   : 5.406   Brazil : 97    Mean   : 3.074
##                3rd Qu.: 7.000   India  : 47    3rd Qu.: 4.218
##                Max.   :13.500   China  : 36    Max.   : 8.934
##                                 (Other): 45
##   Upgrade_time      Upgraded      Zendesk  crTime_Max_V
##  Min.   :0.000   Min.   :0.0000   0:495   Min.   :  0.00
##  1st Qu.:0.000   1st Qu.:0.0000   1:473   1st Qu.:  9.00
##  Median :0.000   Median :1.0000           Median : 16.36
##  Mean   :1.391   Mean   :0.5021           Mean   : 20.92
##  3rd Qu.:2.893   3rd Qu.:1.0000           3rd Qu.: 26.24
##  Max.   :7.950   Max.   :1.0000           Max.   :202.67
##
##       CSAT          Tix.Max_V           Tix
##  Min.   :0.0000   Min.   :  0.00   Min.   :  1.00
##  1st Qu.:0.0300   1st Qu.:  3.00   1st Qu.: 12.00
##  Median :0.1500   Median : 13.00   Median : 32.00
##  Mean   :0.2042   Mean   : 24.79   Mean   : 56.19
##  3rd Qu.:0.3200   3rd Qu.: 33.25   3rd Qu.: 75.25
##  Max.   :1.0000   Max.   :246.00   Max.   :684.00
##
```

```
dim(clean_data)
```

```
## [1] 968  15
```

There are 968 Projects that are in our starting dataset; 482 non-upgraded and 486 Upgraded. We have 13 predictive variables that we will use to build our predictive models:

- *Feedback*: # feedback responses the Project has provided
- *Industry*: Project's Account Industry, pulled from Salesforce, grouped into 6 main groups (Manufacturing, Agriculture, Services, Education/Research, Government, and Null & Other)
- *LPP*: Binary indicator if the Project has any LPPs (1 = has LPP, 0 = no LPP)
- *Max_Version*: Highest version Project was on prior to Project's upgrade, if no Upgrade, the Project's current version (5.2, 6, 6.1, 6.2)
- *prev_Upgrade*: Binary indicator if the Project has upgraded before, including minor version upgrades (1 = has prior upgrade, 0 = no prior upgrade)
- *Sub_yrs*: Project's total years as a Subscription Service Subscriber, from Salesforce
- *Support.Region*: Project's Support REgion (Hungary, US, Spain, Brazil, India, China, Other)
- *Time_Max_V*: Number of years Project was on Max Version prior to Upgrade, if no Upgrade, how long on current version
- *Zendesk*: Binary indicator if the Project has Zendesk tickets (1 = has Zendesk tickets, 0 = no Zendesk tickets)
- *Tix_Max_V*: Number of tickets Project has created, on Max Version prior to Upgrade, if no Upgrade, tickets on current version
- *Tix*: Total Number of tickets Project has created
- *crTime_Max_V*: Average crTime (days from Ticket Create Date to Ticket Closed/Solved Date) for tickets on Max Version prior to Upgrade, if no Upgrade, tickets on current version
- *CSAT*: Average Customer Satisfaction response provided by Project (1 if satisfied, 0 if not satisfied)

Note: $Upgrade_time$ will be used in Model B, when predicting the time it takes to Upgrade.

Exploratory Data Analysis

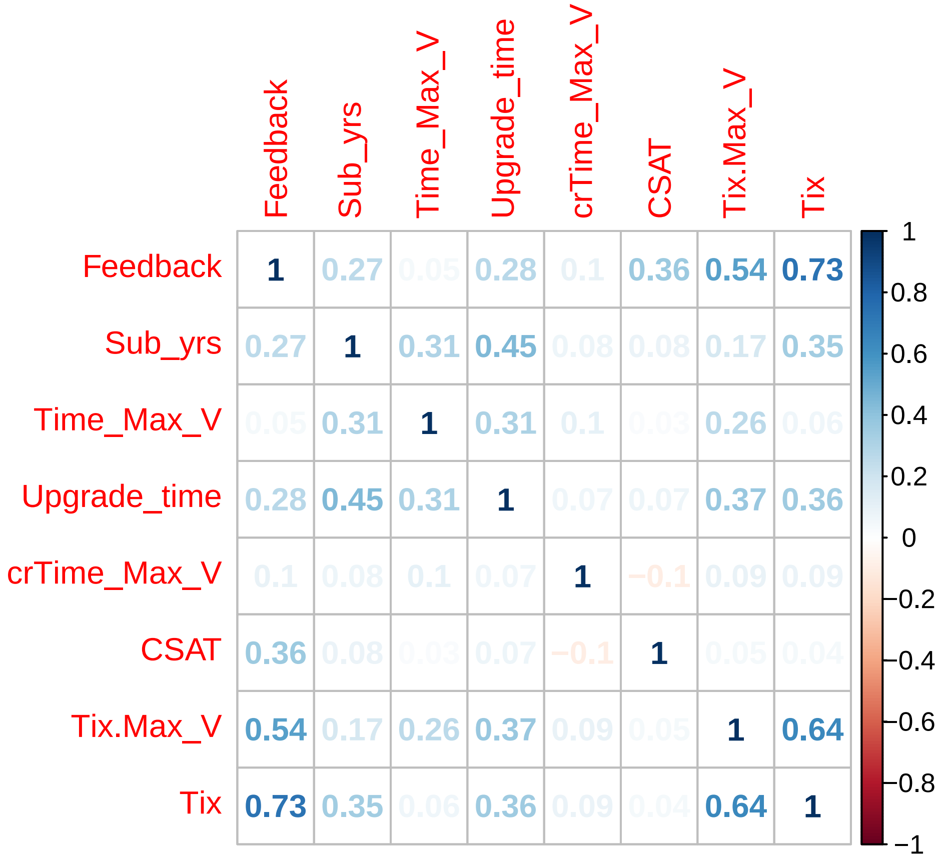
Let's assess the correlation between the quantitative predictors.

```r
# Assess correlation between quant predictors
Q = cor(clean_data[,-c(2,3,4,5,7,10,11)])
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 3.6.3
```

```
## corrplot 0.84 loaded
```

```r
library(RColorBrewer)
corrplot(Q, method="number")
```



	Feedback	Sub_yrs	Time_Max_V	Upgrade_time	crTime_Max_V	CSAT	Tix.Max_V	Tix
Feedback	1	0.27	0.05	0.28	0.1	0.36	0.54	0.73
Sub_yrs	0.27	1	0.31	0.45	0.08	0.08	0.17	0.35
Time_Max_V	0.05	0.31	1	0.31	0.1	0.03	0.26	0.06
Upgrade_time	0.28	0.45	0.31	1	0.07	0.07	0.37	0.36
crTime_Max_V	0.1	0.08	0.1	0.07	1	−0.1	0.09	0.09
CSAT	0.36	0.08	0.03	0.07	−0.1	1	0.05	0.04
Tix.Max_V	0.54	0.17	0.26	0.37	0.09	0.05	1	0.64
Tix	0.73	0.35	0.06	0.36	0.09	0.04	0.64	1

Feedback and Total Tickets (Tix) are strongly correlated (p = 0.73), which is reasonable. The more tickets a Project creates, the more opportunity for feedback. Tickets on Max Version and Total Tickets semi-strongly correlated (p = 0.64), which is reasonable.

Split data into train dataset (for training the model) and test dataset (to assess the model performance).

```r
# 80% Train 20% Test split
sample_size = floor(0.8*nrow(clean_data))
picked = sample(seq_len(nrow(clean_data)), size=sample_size)
train_up = clean_data[picked,]
test_up = clean_data[-picked,]
```

Model A: Logistic Regression

```r
# Build Model A to predict Upgrade using all predictors except Upgrade_time
modelA = glm(Upgraded ~ .-Upgrade_time,  family=binomial, data=train_up)
summary(modelA)
```

```
##
## Call:
## glm(formula = Upgraded ~ . - Upgrade_time, family = binomial,
##     data = train_up)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.3342  -0.3327  -0.0003   0.2674   3.4693
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -1.416e+01  8.500e+02  -0.017   0.9867
## Feedback                  -2.479e-04  1.256e-02  -0.020   0.9843
## IndustryEducation/Research -2.616e-01  2.375e+00  -0.110   0.9123
## IndustryGovernment        -7.632e-01  2.364e+00  -0.323   0.7468
## IndustryManufacturing     -5.928e-01  2.385e+00  -0.249   0.8037
## IndustryNull & Other      -4.836e-01  2.561e+00  -0.189   0.8502
## IndustryServices          -5.916e-01  2.345e+00  -0.252   0.8009
## LPP1                       5.480e-01  4.463e-01   1.228   0.2195
## Max_Version6               1.160e+01  8.500e+02   0.014   0.9891
## Max_Version6.1             1.167e+01  8.500e+02   0.014   0.9890
## Max_Version6.2             1.177e+01  8.500e+02   0.014   0.9890
## prev_Upgrade1             -6.857e-01  3.861e-01  -1.776   0.0758 .
## Sub_yrs                    2.968e-01  7.560e-02   3.926 8.63e-05 ***
## Support.RegionBrazil       2.287e-01  9.399e-01   0.243   0.8077
## Support.RegionChina       -1.325e+00  1.144e+00  -1.158   0.2469
## Support.RegionHungary      5.656e-01  8.697e-01   0.650   0.5155
## Support.RegionIndia        1.117e+00  1.066e+00   1.048   0.2947
## Support.RegionJapan        1.226e+00  1.361e+00   0.900   0.3679
## Support.RegionSpain        8.800e-01  9.577e-01   0.919   0.3582
## Support.RegionUS           1.200e+00  8.805e-01   1.362   0.1731
## Time_Max_V                -7.799e-01  1.243e-01  -6.274 3.53e-10 ***
## Zendesk1                   4.827e+00  3.859e-01  12.509  < 2e-16 ***
## crTime_Max_V               2.191e-03  7.852e-03   0.279   0.7802
## CSAT                       1.445e+00  7.775e-01   1.859   0.0631 .
## Tix.Max_V                  2.585e-03  7.961e-03   0.325   0.7454
## Tix                        5.635e-03  5.574e-03   1.011   0.3120
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1072.91  on 773  degrees of freedom
## Residual deviance:  347.24  on 748  degrees of freedom
## AIC: 399.24
##
## Number of Fisher Scoring iterations: 15
```

```
## Save Predictions to compare with observed data
test.predA = predict(modelA, test_up, type='response')
```

- *Equation of Model 2*:

$$Upgraded = e^{\sum_{n=1}^{n} estimated\_coeff*predictor}/(1+e^{\sum_{n=1}^{n} estimated\_coeff*predictor})$$

where is the number of predictors.

Some predictors of note:

- *prev_Upgrade*1 (-6.857e-01) is stat sig at alpha = 0.1 level. Therefore, if a customer has a prior upgrade, the log odds of Upgrade decreases by -0.851217. Or the odds of upgrade decreases by 57.31%, since (e^-6.857e-01)=0.5037 (which means Projects that have a previous upgrade are 49.62% less likely to upgrade).
- *Sub_yrs* (2.968e-01) is stat sig at alpha = 0.001 level. Therefore, for a one unit increase in total years as subscriber, the log odds of Upgrade increases by 2.968e-01. Or the odds of upgrade increases by 34.55%, since (e^2.968e-01)=1.3455 (which is 0.3455 more).
- *Time_Max_V* (-7.798e-01) is stat sig at alpha = 0.001 level. Therefore, for a one unit increase in tickets on max version, the log odds of Upgrade decreases by -7.798e-01. Or the odds of upgrade decreases by 54,15%, since (e^-7.798e-01)=0.4585 (which is 0.5415 less).
- *Zendesk*1 (4.826) is stat sig at alpha = 0.001 level. Therefore, if a customer has zendesk tickets, the log odds of Upgrade increases by 4.826. Or the odds of upgrade is 123 times as likely, since (e^4.826)=124.71 (which is 123.71). Note this may not be as meaningful because currently active Projects can only create tickets on Zendesk.
- *CSAT* (1.445) is stat sig at alpha = 0.01 level. Therefore, for a one unit increase in Customer Satisfaction, the log odds of Upgrade increases by 1.445. Or the odds of upgrade is 3.24 times as likely, since (e^1.445)=4.2419 (which is 3.2419 more).

Compare Model Performance

Let's round the prediction values to get binary predictions from which we can compute accuracy (classification rate).

```r
#install.packages("pROC")
library(pROC)
```

```
## Warning: package 'pROC' was built under R version 3.6.1
```

```r
roc_objA = roc(test_up$Upgraded, test.predA)

# Assess optimal threshold
threshA = coords(roc_objA, "best", "threshold",  transpose = TRUE)[1]
threshA
```

```
## threshold
## 0.5209771
```

```r
yhat_threshA = as.integer(test.predA > threshA, transpose = TRUE)

conf_matrixA = as.matrix(table(yhat_threshA, test_up$Upgraded))
conf_matrixA
```

```
##
## yhat_threshA  0  1
##            0 85  9
##            1  6 94
```

```r
accuracyA = sum(yhat_threshA== test_up$Upgraded)/nrow(test_up)
accuracyA
```

```
## [1] 0.9226804
```

```r
# Mean Squared Prediction Error (MSPE)
mspeA = mean((test.predA-test_up$Upgraded)^2)
mspeA
```

```
## [1] 0.0675165
```

```
mspeA_r = mean((yhat_threshA-test_up$Upgraded)^2)
mspeA_r
```

```
## [1] 0.07731959
```

```
# Precision Measure (PM)
pmA = sum((test.predA-test_up$Upgraded)^2)/sum((test_up$Upgraded-mean(test_up$Upgraded))^2)
pmA
```

```
## [1] 0.2711033
```

```
# R-squared
TSS = sum((test_up$Upgraded-mean(test_up$Upgraded))^2)
RSS_A = sum((test_up$Upgraded-test.predA)^2)
R_squared_A = 1 - (RSS_A/TSS)
R_squared_A
```

```
## [1] 0.7288967
```

```
RSS_A_r = sum((test_up$Upgraded-yhat_threshA)^2)
R_squared_A_r = 1 - (RSS_A_r/TSS)
R_squared_A_r
```

```
## [1] 0.6895338
```

For the optimal threshold = 0.5209771, ModelA has accuracy 92.27%. The confusion matrix shows 85 Projects are correctly predicted as non-upgraded, 94 Projects are correctly predicted as upgrade. The mean squared prediction error for rounded estimates = 0.07731959 (the lower the error, the better). The precision measure = 0.2711099. The R-squared value for rounded predictions is 68.95%.

Test on 6-month subset

We will assess the Model A be comparing it's predictions for a subset of projects that either Upgraded or Closed (non-upgrade) in the last 6 months (115 Projects). This subset was withheld from the original dataset that was used to train and test Model A.

```
# use the model to forecast the result for approved/non-upgraded projects
six_mnth = read.csv("dataset (6 mnth).csv", header=TRUE, stringsAsFactors=TRUE)

# Clean data
#
# Set NA to 0
six_mnth[is.na(six_mnth)] = 0

# Remove the irrelevant columns
six_mnth_data = six_mnth[-c(1)] #remove accountEntryId and Upgrade_time
# Convert the numerical categorical variables to predictors
six_mnth_data$LPP = as.factor(six_mnth_data$LPP)
six_mnth_data$Max_Version = as.factor(six_mnth_data$Max_Version)
six_mnth_data$prev_Upgrade = as.factor(six_mnth_data$prev_Upgrade)
six_mnth_data$Zendesk = as.factor(six_mnth_data$Zendesk)

summary(six_mnth_data)
```

```
##     Feedback                    Industry  LPP    Max_Version prev_Upgrade
##  Min.   :  0.00   Education/Research:14   0:18   6  :  3     0:72
##  1st Qu.:  1.00   Government        :21   1:97   6.1: 12     1:43
##  Median :  4.00   Manufacturing     : 4          6.2:100
##  Mean   : 11.86   Services          :76
##  3rd Qu.: 10.00
##  Max.   :121.00
##
##     Sub_yrs          Support.Region   Time_Max_V       Upgrade_time
##  Min.   : 1.000   Hungary  :45     Min.   :-0.1288   Min.   :0.000
##  1st Qu.: 4.500   US       :29     1st Qu.: 3.6288   1st Qu.:0.000
##  Median : 6.200   Spain    :15     Median : 4.5726   Median :0.000
##  Mean   : 6.178   Brazil   :13     Mean   : 4.5208   Mean   :1.877
##  3rd Qu.: 8.000   China    : 6     3rd Qu.: 5.5507   3rd Qu.:4.115
##  Max.   :11.000   Australia: 4     Max.   : 8.7479   Max.   :6.230
##                   (Other)  : 3
##     Upgraded      Zendesk  crTime_Max_V          CSAT
##  Min.   :0.0000   0:40    Min.   :  0.000   Min.   :0.0000
##  1st Qu.:0.0000   1:75    1st Qu.:  9.417   1st Qu.:0.0400
##  Median :0.0000           Median : 17.154   Median :0.1400
##  Mean   :0.4522           Mean   : 23.262   Mean   :0.2039
##  3rd Qu.:1.0000           3rd Qu.: 25.167   3rd Qu.:0.3300
##  Max.   :1.0000           Max.   :307.000   Max.   :1.0000
##
##    Tix.Max_V           Tix
##  Min.   :  1.00   Min.   :  1.0
##  1st Qu.:  4.00   1st Qu.:  8.0
##  Median : 12.00   Median : 23.0
##  Mean   : 27.03   Mean   : 45.9
##  3rd Qu.: 28.50   3rd Qu.: 50.5
##  Max.   :265.00   Max.   :358.0
##
```

```r
dim(six_mnth_data)
```

```
## [1] 115  15
```

```r
## Use Model A to predict for 6 month subset
six_mnth.predA = predict(modelA, six_mnth_data, type='response')

# round based on optimal threshold identified using test_data
six_mnth_yhat_threshA = as.integer(six_mnth.predA > threshA)

six_mnth_predictions = cbind.data.frame(six_mnth[1], six_mnth.predA, six_mnth_yhat_threshA, six_mnth$Up

# Prediction if upgrade output into a csv file
write.csv(six_mnth_predictions,'six_month_predictions.csv')

# Assess Model A performance
sum(six_mnth_yhat_threshA != six_mnth$Upgraded)
```

```
## [1] 24
```

```r
#confusion matrix of 6-month actuals vs. model A predictions
as.matrix(table(six_mnth_yhat_threshA, six_mnth$Upgraded))
```

```
##
```

```
## six_mnth_yhat_threshA  0  1
##                      0 42  3
##                      1 21 49
```

```
# accuracy= (42+49)/(42+3+21+49)=91/115=79.13% accuracy
```

Model A has 79.13% accuracy for the 6 month subset. It correctly predicts 42 Projects that did not Upgrade and correctly predicts 49 Projects that did upgrade. It incorrectly predicts 24 Projects. There are 3 False Negatives (Projects that are predicted to not upgrade that actually did upgrade) and 21 False Positives (Projects predicted to Upgrade, that did not upgrade).

As expected, there are more False Positives as the model overpredicts upgrades.

Use Models to Predict for New Projects

Use models to predict if not-yet Upgraded Projects (Active Projects with only Portal Tickets) will upgrade or not. There are 371 Active Projects that will be fed into the model.

```
# use the model to forecast the result for approved/non-upgraded projects
new = read.csv("new.csv", header=TRUE, stringsAsFactors=TRUE)

# Clean data
#
# Set NA to 0
new[is.na(new)] = 0

# Remove the irrelevant columns
new_data = new[-c(1)] #remove accountEntryId
# Convert the numerical categorical variables to predictors
new_data$LPP = as.factor(new_data$LPP)
new_data$Max_Version = as.factor(new_data$Max_Version)
new_data$prev_Upgrade = as.factor(new_data$prev_Upgrade)
new_data$Zendesk = as.factor(new_data$Zendesk)

summary(new_data)
```

```
##     Feedback                     Industry   LPP     Max_Version
##  Min.   :  0.000   Agriculture       :  3   0: 57   5.2:  3
##  1st Qu.:  1.000   Education/Research: 31   1:314   6  :  6
##  Median :  4.000   Government        : 57           6.1: 41
##  Mean   :  9.334   Manufacturing     : 48           6.2:321
##  3rd Qu.: 11.000   Null & Other      :  1
##  Max.   :250.000   Services          :231
##
##  prev_Upgrade    Sub_yrs         Support.Region   Time_Max_V
##  0:225        Min.   : 0.400   Hungary  :168    Min.   :0.09589
##  1:146        1st Qu.: 5.000   US       : 72    1st Qu.:4.00822
##               Median : 6.100   Spain    : 58    Median :4.97534
##               Mean   : 6.481   Brazil   : 30    Mean   :4.90882
##               3rd Qu.: 8.300   Australia: 13    3rd Qu.:5.90000
##               Max.   :12.000   India    : 13    Max.   :9.67945
##                                (Other)  : 17
##   Upgrade_time    Upgraded  Zendesk  crTime_Max_V          CSAT
##  Min.   :0     Min.   :0     0:147    Min.   :  0.000   Min.   :0.0000
##  1st Qu.:0     1st Qu.:0     1:224    1st Qu.:  9.781   1st Qu.:0.0100
##  Median :0     Median :0              Median : 15.833   Median :0.1200
##  Mean   :0     Mean   :0           Mean   : 21.250   Mean   :0.2036
##  3rd Qu.:0     3rd Qu.:0           3rd Qu.: 24.978   3rd Qu.:0.3100
##  Max.   :0     Max.   :0           Max.   :427.000   Max.   :1.0000
##
##    Tix.Max_V           Tix
##  Min.   :  1.00   Min.   :  1.0
##  1st Qu.:  5.00   1st Qu.:  9.0
##  Median : 15.00   Median : 22.0
##  Mean   : 25.77   Mean   : 40.4
##  3rd Qu.: 31.00   3rd Qu.: 51.5
##  Max.   :234.00   Max.   :361.0
##
```

```
dim(new_data)
```

```
## [1] 371  15
```

```
## Use Model A to predict for new data
new.predA = predict(modelA, new_data, type='response')

# round based on optimal threshold identified using test_data
new_yhat_threshA = as.integer(new.predA > threshA)

new_predictionsA = cbind.data.frame(new[1], new.predA, new_yhat_threshA)
# Prediction if upgrade output into a csv file
write.csv(new_predictionsA,'new_predictions_A.csv')
```

Model B: Customer Upgrade Timeline

Next steps:

Next, we will build a second model to predict WHEN the Upgrade will occur. Instead of building the model with response "Upgraded," we will build the model using "Upgrade_time."

Upgrade_time is the difference between 1st Portal ticket (start on 6 date) create date and 1st DXP ticket (start on 7 date) create date, note Portal ticket is Max Liferay Version prior to upgrade.

Model B will be built using only Upgraded Projects, since Projects that never Upgraded don't have a start on 7 date.

Model B: Linear Regression

```
# Filter for upgraded projects
clean_dataB = clean_data[which(clean_data$Upgraded==1),]
summary(clean_dataB)
```

```
##     Feedback                   Industry    LPP     Max_Version
##  Min.   :  0.00   Agriculture       :  3   0: 16   5.2:  0
##  1st Qu.:  5.00   Education/Research: 60   1:470   6  :  3
##  Median : 12.00   Government        : 95           6.1: 47
##  Mean   : 21.82   Manufacturing     : 59           6.2:436
##  3rd Qu.: 27.00   Null & Other      :  4
##  Max.   :234.00   Services          :265
##
##  prev_Upgrade    Sub_yrs         Support.Region   Time_Max_V
##  0:266        Min.   : 0.500   US       :182    Min.   :-3.479
##  1:220        1st Qu.: 4.600   Hungary  :162    1st Qu.: 1.593
##               Median : 6.100   Spain    : 53    Median : 2.888
##               Mean   : 6.345   Brazil   : 38    Mean   : 2.710
##               3rd Qu.: 8.000   India    : 23    3rd Qu.: 3.917
##               Max.   :13.500   Australia: 15    Max.   : 7.948
##                                (Other)  : 13
##   Upgrade_time      Upgraded Zendesk  crTime_Max_V          CSAT
##  Min.   :0.000   Min.   :1   0: 47   Min.   :  0.00   Min.   :0.000
##  1st Qu.:1.593   1st Qu.:1   1:439   1st Qu.: 10.47   1st Qu.:0.080
##  Median :2.885   Median :1           Median : 17.93   Median :0.180
##  Mean   :2.770   Mean   :1           Mean   : 20.99   Mean   :0.221
##  3rd Qu.:3.917   3rd Qu.:1           3rd Qu.: 27.69   3rd Qu.:0.330
##  Max.   :7.950   Max.   :1           Max.   :108.03   Max.   :1.000
##
##    Tix.Max_V           Tix
##  Min.   :  0.00   Min.   :  1.00
##  1st Qu.:  6.00   1st Qu.: 29.00
##  Median : 20.50   Median : 59.00
##  Mean   : 32.86   Mean   : 83.18
##  3rd Qu.: 46.75   3rd Qu.:110.00
##  Max.   :234.00   Max.   :684.00
##
```

```r
# 80% Train 20% Test split
sample_sizeB = floor(0.8*nrow(clean_dataB))
pickedB = sample(seq_len(nrow(clean_dataB)), size=sample_sizeB)
train_upB = clean_dataB[pickedB,]
test_upB = clean_dataB[-pickedB,]

# Build Model A to predict Upgrade using all predictors except Upgrade_time
modelB = lm(Upgrade_time ~ .-Upgraded,  data=train_upB)
summary(modelB)
```

```
##
## Call:
## lm(formula = Upgrade_time ~ . - Upgraded, data = train_upB)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.44285 -0.14213 -0.02952  0.08729  2.55452
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                0.4792061  0.3315153   1.446   0.1492
## Feedback                  -0.0008908  0.0010003  -0.891   0.3738
## IndustryEducation/Research -0.0440557  0.2226619  -0.198   0.8433
## IndustryGovernment        -0.0409905  0.2211586  -0.185   0.8531
## IndustryManufacturing     -0.0998358  0.2225308  -0.449   0.6540
## IndustryNull & Other      -0.2389321  0.2833047  -0.843   0.3996
## IndustryServices          -0.0902597  0.2192360  -0.412   0.6808
## LPP1                       0.1046738  0.0932616   1.122   0.2624
## Max_Version6.1            -0.1394087  0.2256606  -0.618   0.5371
## Max_Version6.2            -0.2842630  0.2247159  -1.265   0.2067
## prev_Upgrade1              0.0746234  0.0377913   1.975   0.0491 *
## Sub_yrs                    0.0124862  0.0074395   1.678   0.0941 .
## Support.RegionBrazil        0.0336745  0.1087948   0.310   0.7571
## Support.RegionChina        -0.1151995  0.1389090  -0.829   0.4075
## Support.RegionHungary      -0.0433912  0.0906048  -0.479   0.6323
## Support.RegionIndia        -0.0488443  0.1126850  -0.433   0.6649
## Support.RegionJapan        -0.0989853  0.1766514  -0.560   0.5756
## Support.RegionSpain         0.0504106  0.0984154   0.512   0.6088
## Support.RegionUS           -0.0869829  0.0901053  -0.965   0.3350
## Time_Max_V                  0.8889231  0.0105428  84.316   <2e-16 ***
## Zendesk1                    0.0960696  0.0599685   1.602   0.1100
## crTime_Max_V               -0.0020557  0.0010665  -1.927   0.0547 .
## CSAT                        0.1275524  0.1055569   1.208   0.2277
## Tix.Max_V                   0.0009807  0.0005783   1.696   0.0908 .
## Tix                        -0.0002407  0.0003341  -0.721   0.4716
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3006 on 363 degrees of freedom
## Multiple R-squared:  0.9715, Adjusted R-squared:  0.9696
## F-statistic: 515.1 on 24 and 363 DF,  p-value: < 2.2e-16
```

```
## Save Predictions to compare with observed data
test.predB = predict(modelB, test_upB)
```

Model B seems to fit the dataset of Upgraded Projects well with Multiple R-squared: 0.7794, Adjusted R-squared: 0.7648.

Some significant variables in Model B are:

- $Feedback$: -0.012014, for each additional feedback provided, the upgrade time decreases by -0.012014 years, holding all other predictors in the model constant.
- $prev\_Upgrade1$: 1.782731, if a customer has a previous upgrade, the upgrade time increases by 1.782731 years, holding all other predictors in the model constant.
- Sub_yrs: 0.193481, for each additional year as a subscriber, the upgrade time increases by 0.193481 years, holding all other predictors in the model constant.
- $Time\_Max\_V$: 0.707220, for each additional year on the Project's Max Version before Upgrade, the upgrade time increases by 0.707220 years, holding all other predictors in the model constant.
- $Tix.Max\_V$: -0.008496, for each additional ticket on Max Version, the upgrade time decreases by -0.008496 years, holding all other predictors in the model constant.
- Tix: 0.007902, for each additional year as a subscriber, the upgrade time increases by 0.007902, holding all other predictors in the model constant.

Model B Assessment on test_dataset

```
# Mean Squared Prediction Error (MSPE)
mspeB = mean((test.predB-test_upB$Upgrade_time)^2)
mspeB
```

```
## [1] 0.03274517
```

```
# Precision Measure (PM)
pmB = sum((test.predB-test_upB$Upgrade_time)^2)/sum((test_upB$Upgrade_time-mean(test_upB$Upgrade_time))
pmB
```

```
## [1] 0.01294531
```

```
# R-squared
TSS_B = sum((test_upB$Upgrade_time-mean(test_upB$Upgrade_time))^2)
```

```
RSS_B = sum((test_upB$Upgrade_time-test.predB)^2)
R_squared_B = 1 - (RSS_B/TSS_B)
R_squared_B
```

```
## [1] 0.9870547
```

Model B performs pretty well on the test subset. It has Mean Squared Prediction Error = 1.094938, Precision Measure = 0.2110325, and R-squared = 0.7889675.

Model B Assessment on 6 month subset

Let's see how Model B performs on the 6 month subset of Projects that Upgraded in the last 6 months (since Model B upgrade time only applies to Upgraded Projects).

```
# Filter for upgraded projects
six_mnth_dataB = six_mnth_data[which(six_mnth_data$Upgraded==1),]
summary(six_mnth_dataB)
```

```
##     Feedback                     Industry  LPP    Max_Version prev_Upgrade
##  Min.   :  0.00   Education/Research: 3   0: 4   6  : 0      0:32
##  1st Qu.:  2.75   Government        :16   1:48   6.1: 3      1:20
##  Median :  5.50   Manufacturing     : 0          6.2:49
##  Mean   : 12.94   Services          :33
##  3rd Qu.: 10.50
##  Max.   :121.00
##
##     Sub_yrs        Support.Region   Time_Max_V       Upgrade_time
##  Min.   : 1.00   Hungary  :20     Min.   :-0.1288   Min.   :0.000
##  1st Qu.: 4.95   US       :11     1st Qu.: 3.4726   1st Qu.:3.473
##  Median : 7.00   Spain    :10     Median : 4.2151   Median :4.215
##  Mean   : 6.60   Brazil   : 6     Mean   : 4.1497   Mean   :4.152
##  3rd Qu.: 8.70   Australia: 3     3rd Qu.: 5.2781   3rd Qu.:5.275
##  Max.   :10.00   China    : 2     Max.   : 6.2301   Max.   :6.230
##                  (Other)  : 0
##     Upgraded Zendesk  crTime_Max_V         CSAT           Tix.Max_V
##  Min.   :1   0: 0    Min.   :  2.00   Min.   :0.0000   Min.   :  1.00
##  1st Qu.:1   1:52    1st Qu.: 10.65   1st Qu.:0.0575   1st Qu.:  5.75
##  Median :1           Median : 15.15   Median :0.1600   Median : 13.50
##  Mean   :1           Mean   : 21.95   Mean   :0.2085   Mean   : 31.75
##  3rd Qu.:1           3rd Qu.: 25.30   3rd Qu.:0.3300   3rd Qu.: 29.75
##  Max.   :1           Max.   :121.14   Max.   :0.5900   Max.   :265.00
##
##       Tix
##  Min.   :  2.00
##  1st Qu.: 11.50
##  Median : 27.50
##  Mean   : 52.04
##  3rd Qu.: 57.50
##  Max.   :358.00
##
```

```
## Use Model B to predict for 6 month subset
six_mnth.predB = predict(modelB, six_mnth_dataB)

six_mnth_predictionsB = cbind.data.frame(six_mnth_dataB[1], six_mnth.predB, six_mnth_dataB$Upgrade_time
head(six_mnth_predictionsB)
```

```
##    Feedback six_mnth.predB six_mnth_dataB$Upgrade_time
## 1        14       4.316064                        4.24
## 5         2       1.816757                        1.51
## 6         9       5.739224                        6.02
## 10       16       4.641991                        4.52
## 12       10       5.921712                        6.12
## 17        6       4.803094                        5.08
```

```
# Prediction if upgrade output into a csv file
write.csv(six_mnth_predictionsB,'six_month_predictionsB.csv')

# Assess Model B performance for 6 month subset

# Mean Squared Prediction Error (MSPE)
mspeB = mean((six_mnth.predB-six_mnth_dataB$Upgrade_time)^2)
mspeB
```

```
## [1] 0.02957758
```

```
# Precision Measure (PM)
pmB = sum((six_mnth.predB-six_mnth_dataB$Upgrade_time)^2)/sum((six_mnth_dataB$Upgrade_time-mean(six_mnt
pmB
```

```
## [1] 0.01395463
```

```
# R-squared
TSS_B = sum((six_mnth_dataB$Upgrade_time-mean(six_mnth_dataB$Upgrade_time))^2)
RSS_B = sum((six_mnth_dataB$Upgrade_time-six_mnth.predB)^2)
R_squared_B = 1 - (RSS_B/TSS_B)
R_squared_B
```

```
## [1] 0.9860454
```

Model B predictions for the six month subset has Mean Squared Prediction Error = 1.729391, precision measure = 0.8159225, but the R-squared is very low = 0.1840775. This is a sign that Model B (built without most recent projects) is less accurate for more recent Projects.

Model B Predictions for new Projects predicted to Upgrade

Next, let's run Model B predictions for our new subset of Project that have been predicted to upgrade.

```
# New Projects predicted to Upgrade
new_B = new_predictionsA[which(new_predictionsA$new_yhat_threshA==1),]

new_data_B = merge(x = new, y = new_B, by = "ï..account.Entry.Id")#, all.y = TRUE)

# Remove the irrelevant columns
new_dataB = new_data_B[-c(1)] #remove accountEntryId
# Convert the numerical categorical variables to predictors
new_dataB$LPP = as.factor(new_dataB$LPP)
new_dataB$Max_Version = as.factor(new_dataB$Max_Version)
new_dataB$prev_Upgrade = as.factor(new_dataB$prev_Upgrade)
new_dataB$Zendesk = as.factor(new_dataB$Zendesk)

summary(new_dataB)
```

```
##     Feedback                     Industry   LPP     Max_Version
##  Min.   :  0.00   Agriculture        :  1   0: 18   6.1:  6    
##  1st Qu.:  2.00   Education/Research: 13   1:181   6.2:193    
##  Median :  5.00   Government         : 36                      
##  Mean   : 11.69   Manufacturing      : 21                      
##  3rd Qu.: 13.50   Null & Other       :  0                      
##  Max.   :250.00   Services           :128                      
##                                                                
##  prev_Upgrade    Sub_yrs       Support.Region   Time_Max_V     
##  0:118        Min.   : 1.000   Hungary:74     Min.   :0.09589  
##  1: 81        1st Qu.: 5.000   Spain  :41     1st Qu.:3.60959  
##               Median : 6.000   US     :38     Median :4.51507  
##               Mean   : 6.318   Brazil :20     Mean   :4.33185  
##               3rd Qu.: 8.050   India  :11     3rd Qu.:5.38356  
##               Max.   :11.000   Japan  : 9     Max.   :7.91507  
##                                (Other): 6                      
##   Upgrade_time    Upgraded Zendesk  crTime_Max_V         CSAT       
##  Min.   :0     Min.   :0    1:199   Min.   :  0.00   Min.   :0.0000  
##  1st Qu.:0     1st Qu.:0            1st Qu.: 10.11   1st Qu.:0.0500  
##  Median :0     Median :0            Median : 14.71   Median :0.1600  
##  Mean   :0     Mean   :0            Mean   : 18.37   Mean   :0.2174  
##  3rd Qu.:0     3rd Qu.:0            3rd Qu.: 24.02   3rd Qu.:0.3250  
##  Max.   :0     Max.   :0            Max.   :117.25   Max.   :1.0000  
##                                                                      
##    Tix.Max_V           Tix          new.predA      new_yhat_threshA
##  Min.   :  1.00   Min.   :  1.0   Min.   :0.5263   Min.   :1       
##  1st Qu.:  6.00   1st Qu.: 11.0   1st Qu.:0.7788   1st Qu.:1       
##  Median : 20.00   Median : 29.0   Median :0.8682   Median :1       
##  Mean   : 32.03   Mean   : 51.9   Mean   :0.8390   Mean   :1       
##  3rd Qu.: 41.50   3rd Qu.: 74.0   3rd Qu.:0.9279   3rd Qu.:1       
##  Max.   :234.00   Max.   :361.0   Max.   :0.9972   Max.   :1       
## 
```

```
dim(new_dataB)
```

```
## [1] 199  17
```

```
## Use Model B to predict for new data
new.predB = predict(modelB, new_dataB)


new_predictionsB = cbind.data.frame(new_data_B[1], new.predB)

# Prediction if upgrade output into a csv file
write.csv(new_predictionsB,'new_predictions_B.csv')
```

After loading Model B into Tableau and adding the forecasted Upgrade Time (new.predB) to each Project's Start on 6 Date (Create Date of 1st Portal Ticket on Max Version), I observed some predicted Upgrade Projects have a forecasted Upgrade date that has already passed

- 150 Projects have a predicted Upgrade date in the past
- 49 has a predicted Upgrade date in the future).

Note: The earliest passed predicted Upgrade is still in 2020, February 17, 2020, which could still be helpful information.

Since, we earlier noticed that Model B built without the most recent 6-months worth of data has a low

R-squared value, it's possible that Model B performs worse for Active Projects as times goes on. This is because as an Active Project continues to not Upgrade and not Close, it is not represented in the dataset used to build the model (since it is still active). The majority of Upgraded Projects in the model building dataset will have shorter Upgrade Time, so Model B will tend to under-estimate the Upgrade Time.

Testing the Model B predictions after adding the 6 month subset does not improve predictions for new Projects. Because the 6 month subset has no Upgrades that have taken over 6.5 years, the predictions actually decrease.

Testing alternative Start Dates, such as Last Portal Offering Date, gave a worse model without strong improvements. Additionally, Offering Dates do not always reflect when customers actually start on a version/start generating tickets.

Conclusion:

In conclusion, Model A predictions for active Projects with only Portal tickets are output into a csv file. The equation is hard-coded into the Tableau Report.

currently, the model likely predicts more upgrades than in reality because no Active non-upgraded Projects are included in the training/testing dataset, ideally we should have examples of active accounts that did not upgrade/will not upgrade.

Model B predictions for when a Project will upgrade are also output into a csv file and hard-coded into the Tableau Report. Model B is built using only Upgrade projects. Unfortunately, the majority of the predicted Upgrade dates are in the past. This is due to the fact that the model building dataset fails to accurately capture the new dataset of Projects. The model building dataset fails to include many Projects that have Upgraded after a long time, since they have either Closed without Upgrading or are still Active in the new dataset.

Considering how to deal with Projects with predicted Upgrade Dates in the past:

- We can consider the customers with Upgrade timeline in the past as Non-upgrade. Since the model predicts they should've upgraded, but they did not perhaps it is sufficient to conclude they are non-upgrade? This could help generate more examples of Active Portal customers who are Non-upgrade customers. However, this is non-ideal because our goal is to have every customer upgrade. If we settle for having these customers are non-upgrade customers and they eventually do upgrade, we will be unprepared.
- We can try to add more time to their predicted Upgrade date through alternative methods. This method is harder because the problem is limited data on which to forecast.
- Or we can treat all Projects with past Upgrade Date as high priority/assume they are Upgrading now.

Next Steps

Moving forward, Patricia Draut will be reaching out to CAS and RSM team members to get their suggestions about valuable predictors.

Additionally, as time moves on, more long-standing Projects will either Upgrade or Close, which will be helpful for improving our Models. However, we will need to generate the new Model at a time when there are still sufficient non-Upgraded Projects, so that the model will be useful.