6414 Project Customer Churn

Group 4

Tyler Burns

Rishidhar Duvvuru

Victor Lai

Rayan Maleck

Spiros Valouxis

Under the Guidance of **Pr. Gamze Tokol-Goldsman**

Summary

Georgia Tech

Introduction

Problem Description

This project addresses the challenge of high **customer churn rates** within an Iranian telecom company by leveraging various logistic regression techniques to analyze and model customer data.

Significance

- Business Impact: Retain revenue and market share
- Operational Efficiency: Allocate resources effectively
- Data Insights: Understand the business & customers on a deeper level through data

Georgia Tech

Objectives

Conduct in-depth data analysis on customer profiles to understand churn patterns

Build and evaluate various logistic regression models to identify the most effective model & predictors for churn prediction

Provide actionable insights for strategic decision-making based on model output

Georgia Tech

Data Description & Analysis

Georgia Tech

Original Dataset

Our dataset (3150 rows x 14 columns) is randomly collected from an Iranian telecom company's database over a period of 12 months.

8 Quantitative Variables:
Call Failures: number of call failures
Subscription Length: total months of subscription
Distinct Called Numbers: number of distinct phone calls
Seconds of Use: total seconds of calls
Frequency of use: total number of calls
Frequency of SMS: total number of text messages
Customer Value: The calculated value of customer
Age: age of customer

Dependent Variable:
Churn: binary (1: churn, 0: non-churn)

Original Dataset

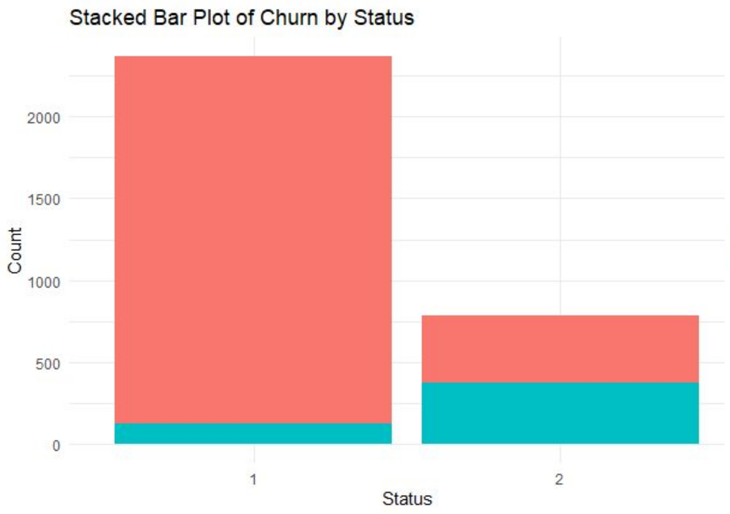
Our dataset (3150 rows x 14 columns) is randomly collected from an Iranian telecom company's database over a period of 12 months.

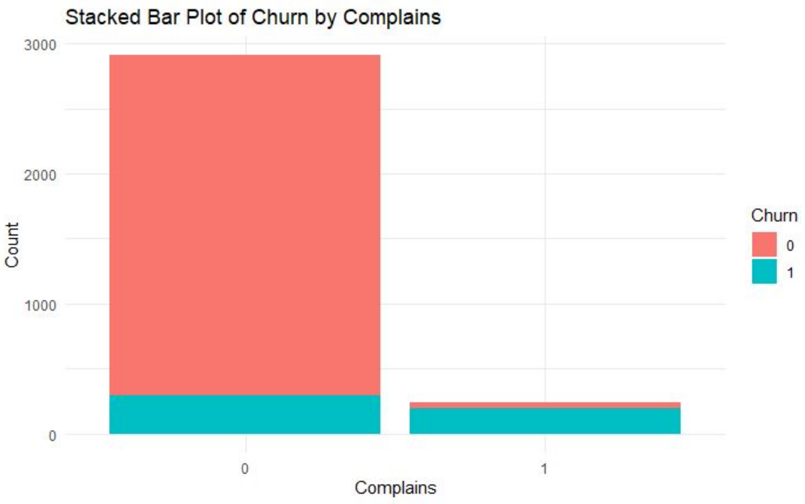
5 Qualitative Variables:
Complains: binary (0: No complaint, 1: complaint)
Age Group: ordinal (1: younger, 5: older)
Tariff Plan: binary (1: Pay as you go, 2: contractual)
Status: binary (1: active, 2: non-active)
Charge Amount: Ordinal (0: lowest, 9: highest)

Dependent Variable:
Churn: binary (1: churn, 0: non-churn)

Churn vs Status & Complains

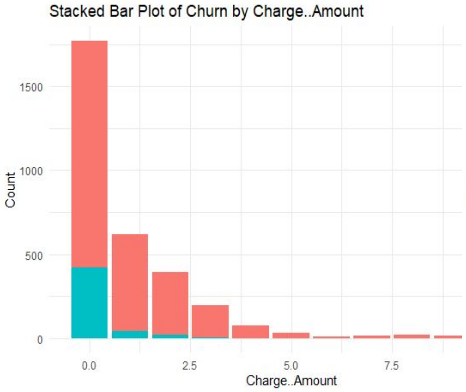
- Customers who are inactive(status =2) has very high churn rate compared to active users.
- While a small % of users had complaints, Majority of those users who had complaints has churned.

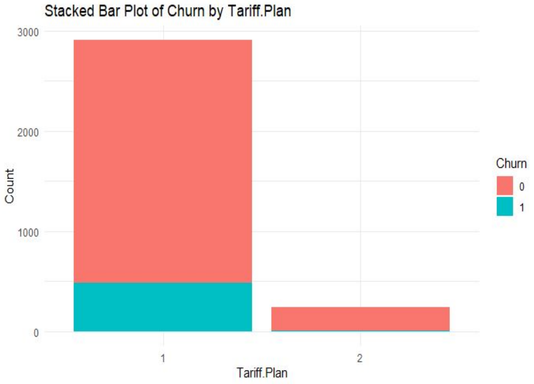


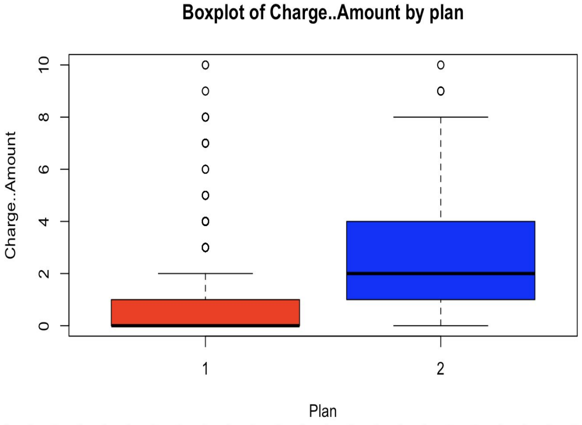


Churn vs Charge Amount & Tariff plan

- The %customers and #churns is decreasing with increase in charge amount(from 0 to 9).
- Majority of the customers are in “Pay as you go”(=1) plan which is witnessing most of the churns.
- “Pay as you go” has lower charge amounts vs “contractual” plans.

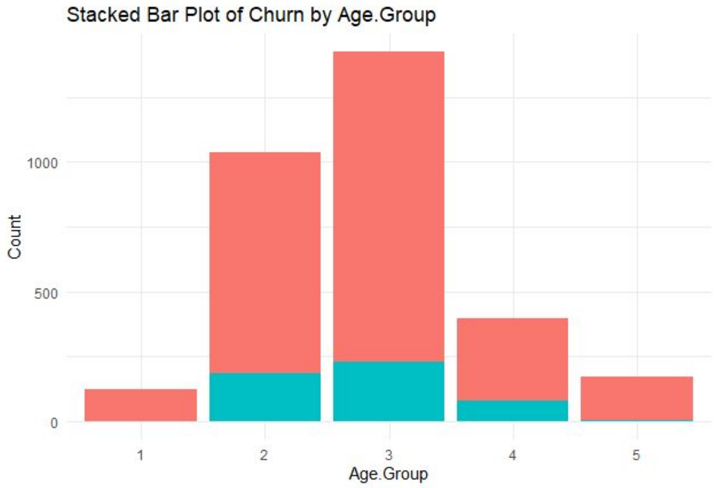


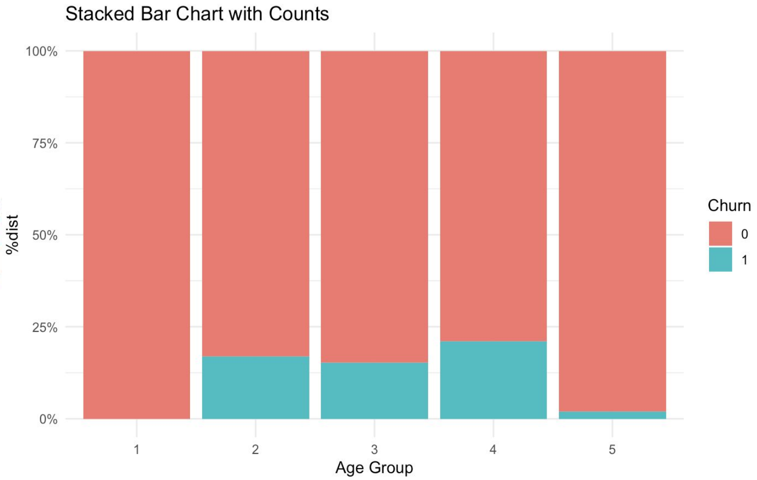




Churn vs Age Group & Tariff Plan

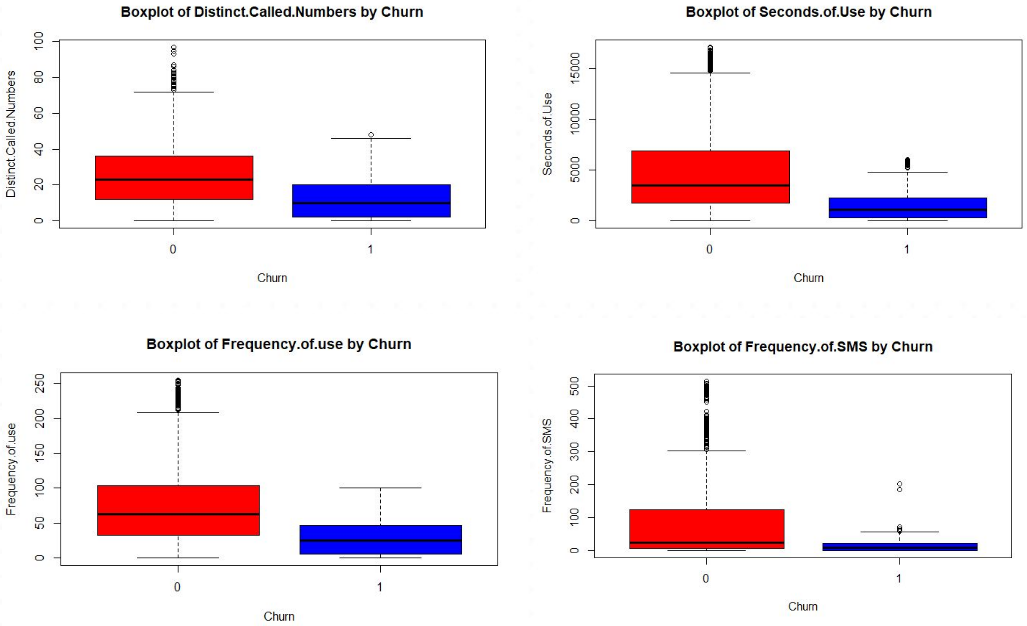
- The distribution of customers and number of customers churned initially increases and then decreases as Age group increases, following a bell shaped trend.





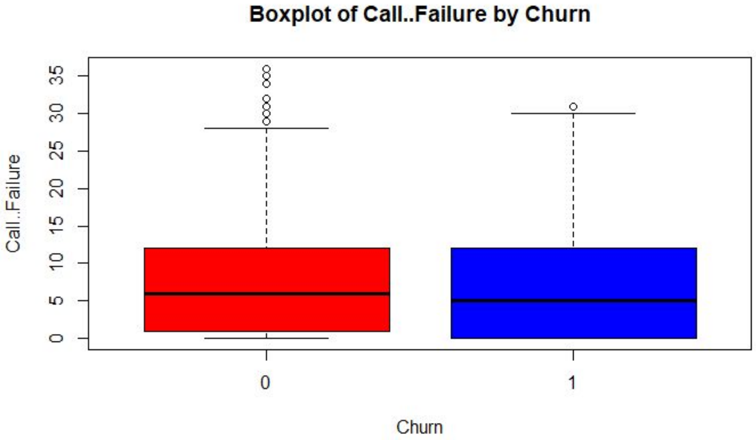
Box plot of Churn vs No Churn - 1/2

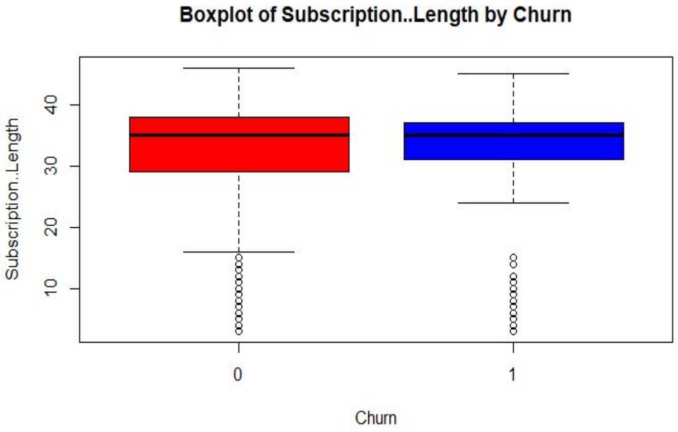
The box plots show that the customer behavior variables could be good indicators to predict churn probability.
Churned customers display lower activity compared to normal customers.



Box plot of Churn vs No Churn - 2/2

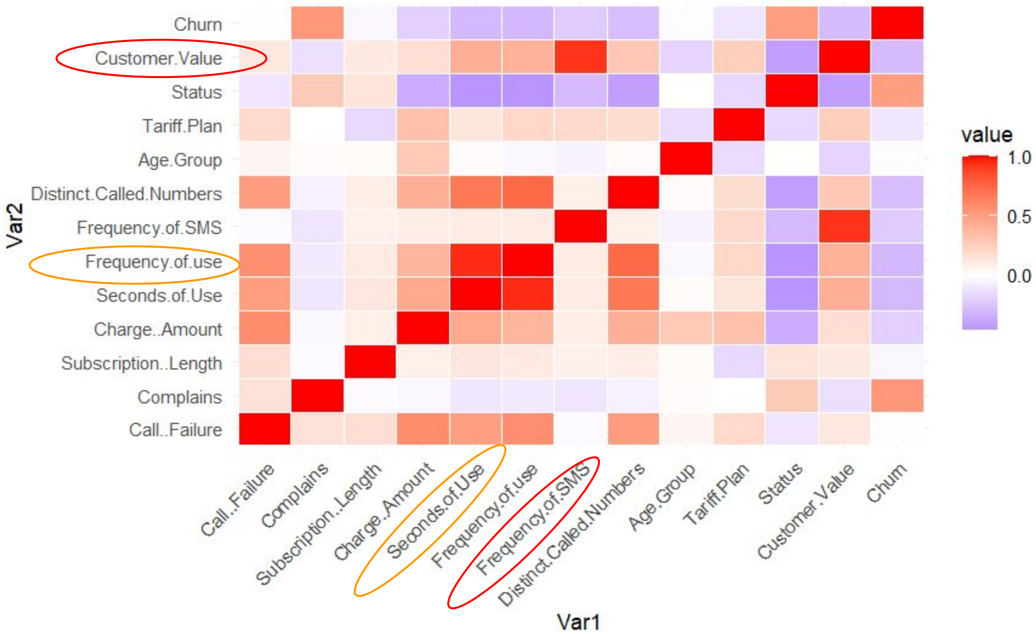
- The boxplot of Subscription Length by Churn and the boxplot of Call..Failure by Churn both have similar distributions for the customer leaving and staying.
- The primary difference between churn vs no churn is simply the no churn data has more variance for subscription length.





Correlation Matrix Heatmap

- “Frequency of use” is strongly correlated with “Seconds of use”. While “Customer Value” is strongly related to “Frequency of SMS”.
- “Churn” shows a positive correlation among customers that had “complaints” or were “inactive”, while showing mild negative correlation with most other predictors.



From the heat map we could identify the correlation between the variables. There could be an issue with multicollinearity

Model Fitting

Logistic regression - Full model

- We randomly split the data into Train & Test data(70/30) and fit the logistic model on train dataset.

```
glm(formula = Churn ~ ., family = "binomial", data = train_data)

Coefficients:
                        Estimate Std. Error z value Pr(>|z|)
(Intercept)           -1.664e+01  5.014e+02  -0.033  0.97353
Call..Failure          1.653e-01  2.820e-02   5.862 4.58e-09 ***
Complains1             3.887e+00  4.043e-01   9.616  < 2e-16 ***
Subscription..Length  -1.983e-02  1.628e-02  -1.218  0.22314
Charge..Amount        -4.462e-01  1.963e-01  -2.273  0.02300 *
Seconds.of.Use        -7.728e-05  2.598e-04  -0.297  0.76614
Frequency.of.use      -6.425e-02  1.372e-02  -4.683 2.82e-06 ***
Frequency.of.SMS      -7.204e-02  2.201e-02  -3.274  0.00106 **
Distinct.Called.Numbers  2.205e-03  1.387e-02   0.159  0.87366
Age.Group2             1.539e+01  5.014e+02   0.031  0.97552
Age.Group3             1.561e+01  5.014e+02   0.031  0.97517
Age.Group4             1.642e+01  5.014e+02   0.033  0.97387
Age.Group5             1.485e+01  5.014e+02   0.030  0.97637
Tariff.Plan2           7.747e-01  1.001e+00   0.774  0.43897
Status2                1.275e+00  3.204e-01   3.978 6.94e-05 ***
Customer.Value         1.351e-02  5.011e-03   2.697  0.00700 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1197.4  on 1399  degrees of freedom
Residual deviance:  568.2  on 1384  degrees of freedom
AIC: 600.2
```

Significant: Call Failure, Complains, Charge Amount, Frequency of Use, Frequency of SMS, Status, Customer Value.

Not Significant: Intercept, Subscription length, Seconds of Use, Distinct called Numbers, Age group, Tariff plan.

Need to check for multicollinearity

We considered a significance level of 0.1 for our models

Georgia Tech

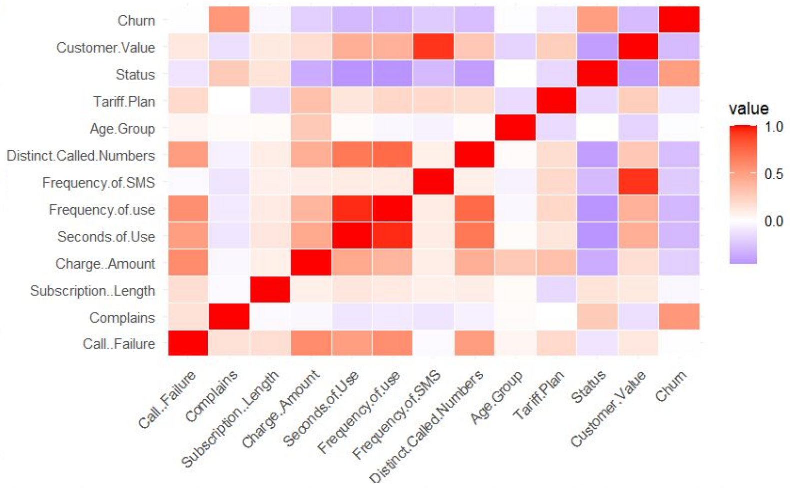
Logistic regression - Full model(VIF Test)

	GVIF	Df	GVIF^(1/(2*Df))
Call..Failure	2.778638	1	1.666925
Complains	1.194657	1	1.093004
Subscription..Length	1.418752	1	1.191114
Charge..Amount	2.434240	1	1.560205
Seconds.of.Use	26.761961	1	5.173196
Frequency.of.use	15.804565	1	3.975496
Frequency.of.SMS	32.394665	1	5.691631
Distinct.Called.Numbers	2.699617	1	1.643051
Age.Group	2.719626	4	1.133218
Tariff.Plan	1.567810	1	1.252122
Status	2.002214	1	1.414996
Customer.Value	62.065115	1	7.878142

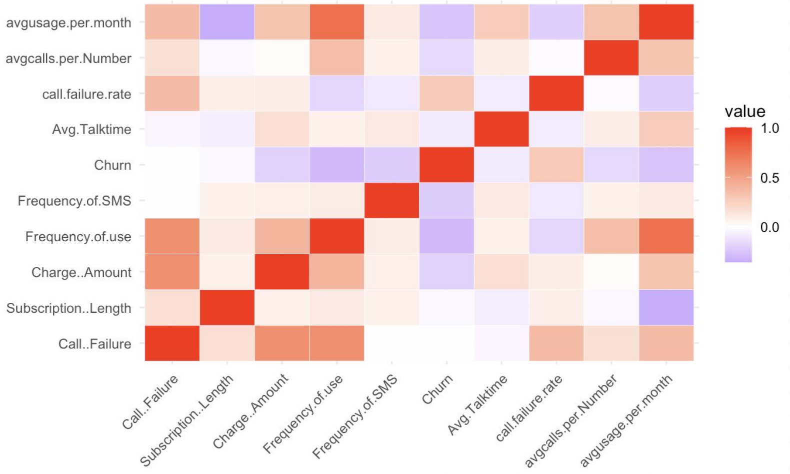
GVIF > 10: Seconds of Use, Frequency of Use, Frequency of SMS, Customer Value. Some of these variables needs to be removed.

Eliminating Multicollinearity & creating new variables

- We would want to remove the variables causing multicollinearity without losing additional information provided by these variables.
- So, we created interaction variables like “avg talk time”, “Call Failure rate”, “Avg calls per number dialed” etc before removing the correlated variables.



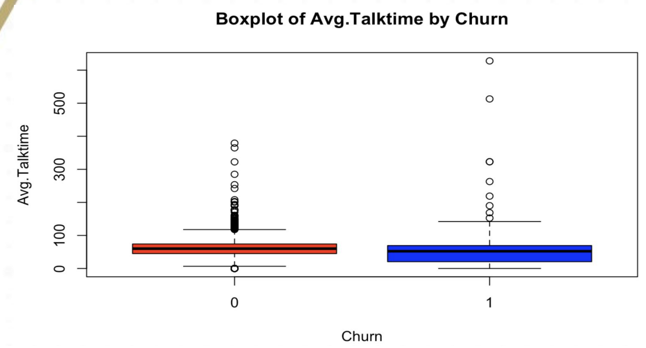
Original Variables

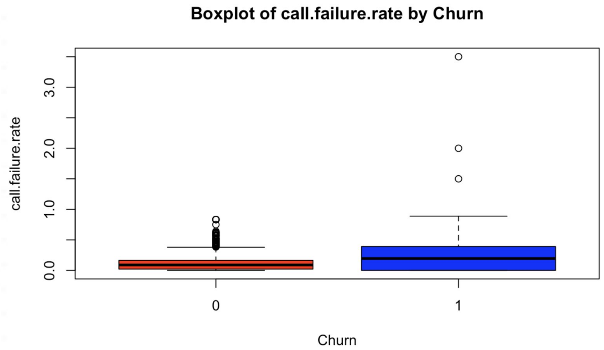


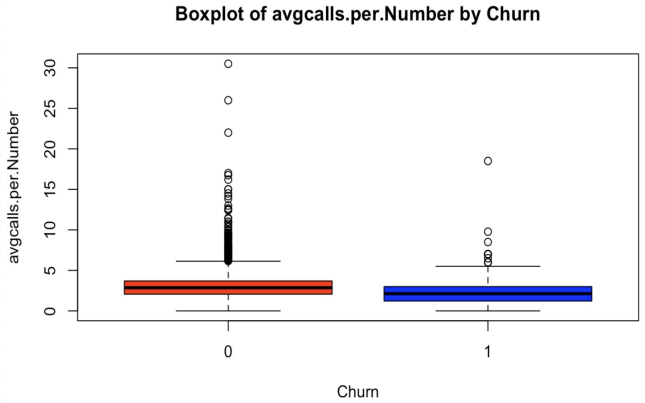
New set of variables

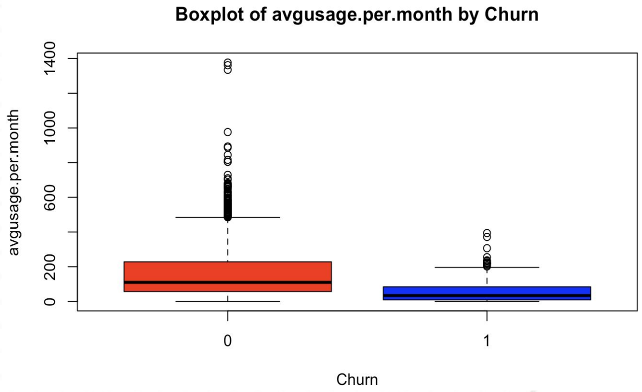
Quick EDA on the new variables

The box plots shows that the distribution of new variables between churned vs customer who stayed.









Logistic regression - New model

- We removed the variables with high multicollinearity and retrained the model.

```
glm(formula = Churn ~ ., family = "binomial", data = train_data2)

Coefficients:
                       Estimate Std. Error z value Pr(>|z|)
(Intercept)          -1.633e+01  5.156e+02  -0.032 0.974731
Call..Failure         1.243e-01  3.155e-02   3.940 8.14e-05 ***
Complains1            3.870e+00  4.119e-01   9.397  < 2e-16 ***
Subscription..Length -2.230e-03  1.793e-02  -0.124 0.901027
Charge..Amount       -5.355e-01  1.793e-01  -2.987 0.002814 **
Frequency.of.use     -3.576e-02  6.912e-03  -5.174 2.29e-07 ***
Frequency.of.SMS     -2.191e-02  5.903e-03  -3.713 0.000205 ***
Age.Group2            1.500e+01  5.156e+02   0.029 0.976797
Age.Group3            1.505e+01  5.156e+02   0.029 0.976720
Age.Group4            1.534e+01  5.156e+02   0.030 0.976261
Age.Group5            1.338e+01  5.156e+02   0.026 0.979302
Tariff.Plan2          1.494e+00  1.034e+00   1.445 0.148366
Status2               9.526e-01  3.160e-01   3.015 0.002574 **
Avg.Talktime          5.732e-05  2.639e-03   0.022 0.982671
call.failure.rate     9.194e-01  6.184e-01   1.487 0.137105
avgcalls.per.Number  -1.288e-01  6.306e-02  -2.042 0.041107 *
avgusage.per.month    4.704e-03  1.464e-03   3.213 0.001316 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1197.37  on 1399  degrees of freedom
Residual deviance:  567.59  on 1383  degrees of freedom
AIC: 601.59
```

Significant: Call Failure, Complains, Charge Amount, Frequency of Use, Frequency of SMS, Status, Customer Value, Avg calls per number, Avg usage per month

Not Significant: Intercept, Subscription length, Age group, Tariff plan, Avg Talk Time, Call failure rate.

check for multicollinearity

We considered a significance level of 0.1 for our models

Logistic regression - New model(VIF Test)

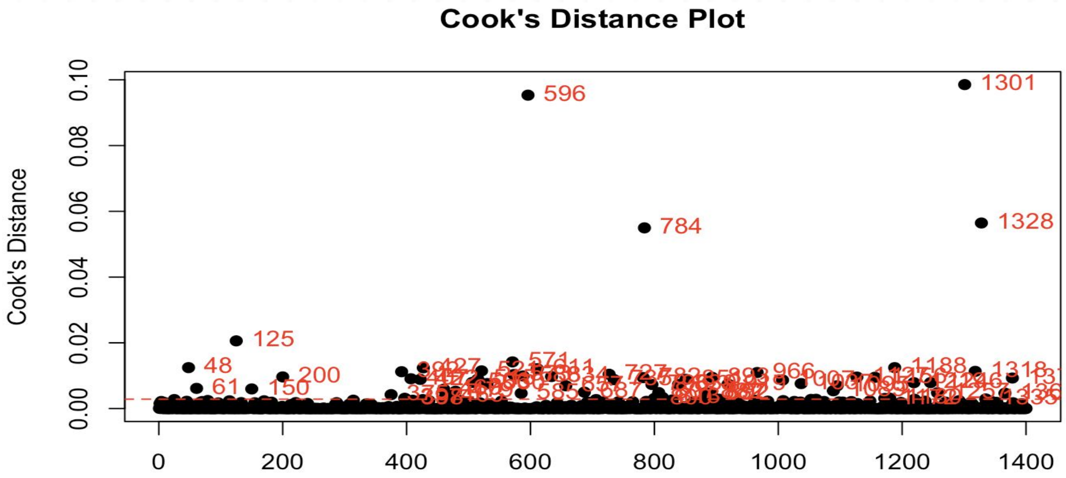
All the predictors have GVIF < 10 so multicollinearity is not an issue anymore

	GVIF	Df	GVIF^(1/(2*Df))
Call..Failure	3.353955	1	1.831381
Complains	1.171691	1	1.082447
Subscription..Length	1.941790	1	1.393481
Charge..Amount	2.428447	1	1.558347
Frequency.of.use	3.454169	1	1.858539
Frequency.of.SMS	2.049576	1	1.431634
Age.Group	1.593310	4	1.059955
Tariff.Plan	1.859141	1	1.363503
Status	1.959336	1	1.399763
Avg.Talktime	1.258479	1	1.121819
call.failure.rate	1.848863	1	1.359729
avgcalls.per.Number	1.281503	1	1.132035
avgusage.per.month	3.113549	1	1.764525

Logistic regression - New model(Outlier test)

Using cook's distance to identify potential influential points for the model.

```r
#check for outliers, leverage points using cook's distance
cooks_distance <- cooks.distance(full.model2)
cook_threshold <- 4 / nrow(train_data2)
outliers <- which(cooks_distance > cook_threshold)
plot(cooks_distance, pch = 19, main = "Cook's Distance Plot", xlab = "Observation", ylab = "Cook's Distance")
abline(h = cook_threshold, col = "red", lty = 2)
text(outliers, cooks_distance[outliers], labels = outliers, col = "red", pos = 4)
```



From the plot we could see 4 potential influential points.

Logistic regression (Removing influential points)

- Based on the cook's distance plot we retrained the model after removing the influential points.

```
train_no_outliers <- train_data2[-c(596,784,1301,1328),]
full.model.no_outliers <- glm(Churn ~ ., data = train_no_outliers, family = "binomial")
summary(full.model.no_outliers)
```

```
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
Call:
glm(formula = Churn ~ ., family = "binomial", data = train_no_outliers)

Coefficients:
                        Estimate Std. Error z value Pr(>|z|)
(Intercept)           -1.807e+01  1.311e+03  -0.014 0.989004
Call..Failure          1.516e-01  3.288e-02   4.611 4.02e-06 ***
Complains1             3.648e+00  4.074e-01   8.953  < 2e-16 ***
Subscription..Length   2.403e-03  1.861e-02   0.129 0.897252
Charge..Amount        -4.485e-01  1.622e-01  -2.766 0.005679 **
Frequency.of.use      -3.728e-02  6.785e-03  -5.495 3.90e-08 ***
Frequency.of.SMS      -4.074e-02  8.716e-03  -4.674 2.95e-06 ***
Age.Group2             1.652e+01  1.311e+03   0.013 0.989949
Age.Group3             1.674e+01  1.311e+03   0.013 0.989814
Age.Group4             1.717e+01  1.311e+03   0.013 0.989551
Age.Group5             1.438e+01  1.311e+03   0.011 0.991250
Tariff.Plan2          -1.393e+01  7.172e+02  -0.019 0.984498
Status2                1.195e+00  3.239e-01   3.690 0.000224 ***
Avg.Talktime          -1.983e-04  2.634e-03  -0.075 0.939991
call.failure.rate      7.754e-01  6.028e-01   1.286 0.198299
avgcalls.per.Number   -2.328e-01  7.891e-02  -2.950 0.003179 **
avgusage.per.month     7.890e-03  1.798e-03   4.387 1.15e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1182.28  on 1395  degrees of freedom
Residual deviance:  539.67  on 1379  degrees of freedom
AIC: 573.67
```

Removing the outliers has slightly improved the significance of some of the variables slightly and reduced the AIC value. But still the intercept is not statistically significant.

The warning "glm.fit: fitted probabilities numerically 0 or 1 occurred" in logistic regression often indicates that the model is having difficulty estimating probabilities for extreme values of the predictors. So removing influential points doesn't seem to be the right approach.

We considered a significance level of 0.1 for our models

Georgia Tech

Logistic regression - Stepwise model

- To decrease model complexity, we applied stepwise regression to eliminate potential insignificant variables and overfitting from the model.

```
#step wise
min.model <- glm(Churn~1,family="binomial",data =train_data2)
step.model <- step(min.model, scope = list(lower = min.model, upper = full.model2),
direction = "both", trace = FALSE)
summary(step.model)
```

```
Call:
glm(formula = Churn ~ Status + Complains + Frequency.of.use +
    Call..Failure + Frequency.of.SMS + Charge..Amount + avgusage.per.month +
    avgcalls.per.Number + call.failure.rate, family = "binomial",
    data = train_data2)

Coefficients:
                     Estimate Std. Error z value Pr(>|z|)
(Intercept)         -1.580629   0.295678  -5.346  9.0e-08 ***
Status2              1.149371   0.278587   4.126  3.7e-05 ***
Complains1           3.914692   0.412679   9.486  < 2e-16 ***
Frequency.of.use    -0.035566   0.006269  -5.674  1.4e-08 ***
Call..Failure        0.119163   0.029891   3.987  6.7e-05 ***
Frequency.of.SMS    -0.017636   0.005367  -3.286 0.001017 **
Charge..Amount      -0.503015   0.139732  -3.600 0.000318 ***
avgusage.per.month   0.004351   0.001210   3.594 0.000325 ***
avgcalls.per.Number -0.104702   0.060499  -1.731 0.083515 .
call.failure.rate    0.828733   0.579616   1.430 0.152775
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1197.37  on 1399  degrees of freedom
Residual deviance:  580.44  on 1390  degrees of freedom
AIC: 600.44
```

The intercept is now statistically significant.

Predictors like Status 2 (inactive), Complains 1(Yes), Call Failures **increase the odds of customer churn** per unit increase in the respective predictors while others are kept constant.

Predictors like Frequency of use, SMS, Charge Amount, Avg calls per number etc **decreases the odds of customer churn** per unit increase in the respective predictors while others are constant.

We considered a significance level of 0.1 for our models

Logistic regression - Stepwise (Model significance)

- Testing the model significance using chi-squared test.
- Null Hypothesis : $\beta 1 = \beta 2 = ... = \beta k = 0$
- Alternate Hypothesis : βi !=0 for at least one of i in {1:k}

```
dof3 = 1399-1390 #df of null deviance - df of residual deviance
test_stat =(step.model$null.deviance - step.model$deviance)
critical_deviance <- qchisq(1 - 0.05, dof3)
p_val=1-pchisq(test_stat,dof3)
print(c(test_stat,critical_deviance,p_val))
```

```
[1] 616.93075  16.91898   0.00000
```

From the above chi-squared test, we can see that p-value ~0. So, we reject the null hypothesis and conclude that **the model is statistically significant**.

We considered a significance level of 0.1 for our models

Georgia Tech

Logistic regression - Stepwise(GOF)

- Testing the model for Goodness of Fit using Deviance & Pearson's tests
- Null Hypothesis : Model is a good fit
- Alternate Hypothesis : Model is not a good fit

```
#GOF
deviance_value <- deviance(step.model)
df2 <- df.residual(step.model)
critical_dev <- qchisq(1 - 0.05, df2)
p_val2 <- 1 - pchisq(deviance_value, df2)
print(c(deviance_value,critical_dev,p_val2))
```

```
pearson_resid <- residuals(step.model, type = "pearson")
pearson_chi_square <- sum(pearson_resid^2)
pearson_df <- df.residual(step.model)
pearson_p_value <- 1 - pchisq(pearson_chi_square, df = pearson_df)
print(c(pearson_chi_square,pearson_p_value))
```

```
[1]  580.4423 1477.8481    1.0000
```

```
[1] 967.3599   1.0000
```

For both deviance & pearson's test the p-value ~1.0 which is >> alpha = 0.1. So we fail to reject the null hypothesis. Thus, we conclude that **the model is a good fit**.

We considered a significance level of 0.1 for our models

Logistic Regression - Lasso

```
library(glmnet)

X <- model.matrix(Churn~., data = train_data[, -train_data$Churn])
y <- train_data$Churn

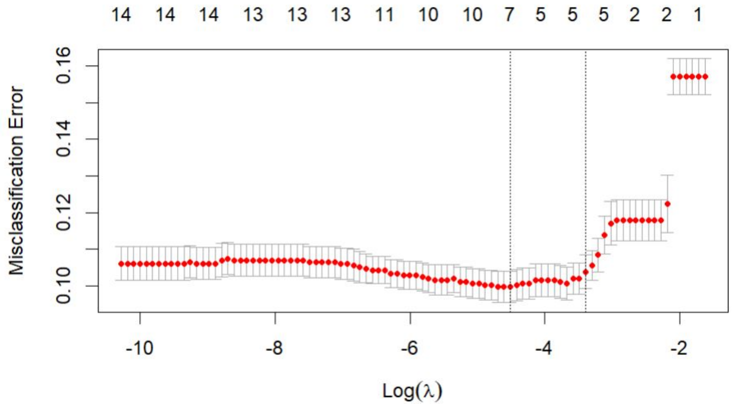
lasso_logistic_model <- cv.glmnet(X, y, family = "binomial", alpha = 1,
type.measure = "class")
```

Selected Variables: Subscription Length, Frequency of Use, Distinct Called Numbers, Age Group 5, Status, Customer Value

```
best_lambda <- lasso_logistic_model$lambda.min
lasso_logistic_coefficients <- coef(lasso_logistic_model, s = best_lambda)
print(lasso_logistic_coefficients)
```

```
16 x 1 sparse Matrix of class "dgCMatrix"
                                 s1
(Intercept)            -1.669576450
(Intercept)             .
Complains1              3.433064682
Subscription..Length   -0.005773390
Charge..Amount          .
Seconds.of.Use          .
Frequency.of.use       -0.005617937
Frequency.of.SMS        .
Distinct.Called.Numbers -0.013847857
Age.Group2              .
Age.Group3              .
Age.Group4              .
Age.Group5             -0.452452147
Tariff.Plan2            .
Status2                 1.476588345
Customer.Value         -0.001104077
```

```
plot(lasso_logistic_model)
```



Analysis of Lasso Model

```
```{r}
best_lasso_model <- glmnet(X, y, alpha = 1, lambda = best_lambda)
train_prob <- predict(best_lasso_model,newx=X,s=best_lambda,type="response")
```
```

```
```{r}
x_test <- model.matrix(Churn~.,test_data[, -test_data$Churn])
#predict class, type="response"
lasso_prob <- predict(best_lasso_model,newx=x_test,s=best_lambda,type="response"
)
#translate probabilities to predictions
predictions5 <- rep(0,nrow(test_data))
predictions5[lasso_prob>.27] <- 1
```

```
```{r}
#The model does not fit the data well based on the Hosmer Lemeshow test
library(ResourceSelection)
hoslem.test(y, train_prob)
```
```

```
	Hosmer and Lemeshow goodness of fit (GOF) test

data:  y, train_prob
X-squared = 28.961, df = 8, p-value = 0.0003221
```

- We decided to use the Hosmer-Lemeshow goodness of fit test.
- With a p-value of 0.0003, we can conclude that this model does not fit the data well.
```

Random Forest Model

```
library(randomForest)
library(randomForestSRC)
set.seed(123)
rf_model <- rfsrc(y ~ ., data = data.frame(X, y), ntree = 1000, nodesize = 5)
print(rf_model)
```

```
                         Sample size: 2204
                     Number of trees: 1000
           Forest terminal node size: 5
       Average no. of terminal nodes: 76.737
No. of variables tried at each split: 5
              Total no. of variables: 15
       Resampling used to grow trees: swor
    Resample size used to grow trees: 1393
                            Analysis: RF-R
                              Family: regr
                      Splitting rule: mse *random*
       Number of random split points: 10
                     (OOB) R squared: 0.72008087
   (OOB) Requested performance error: 0.03706195
```

- R-square is 0.72, which means the model does well to explain the variability in y.
- The error rate is 3.7%.

Random Forest Model

```r
importance = vimp(rf_model)$importance
kable(importance, caption = "Variable Importance")
```

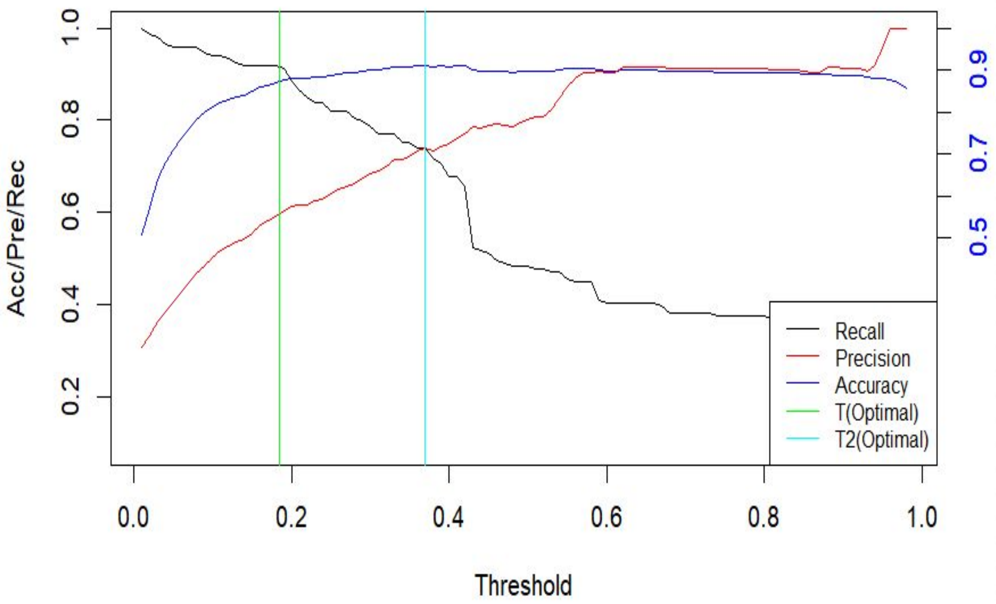
Variable Importance

	x
X.Intercept.	0.0000000
Complains1	0.3639560
Subscription..Length	0.0800021
Charge..Amount	0.0096145
Seconds.of.Use	0.0688263
Frequency.of.use	0.0546812
Frequency.of.SMS	0.0196713
Distinct.Called.Numbers	0.0460106
Age.Group2	0.0217788
Age.Group3	0.0078184
Age.Group4	0.0179016
Age.Group5	0.0033748
Tariff.Plan2	0.0017470
Status2	0.1745406
Customer.Value	0.0251976

The 4 most important variables in the random forest model are Complains, Subscription length, Seconds of Use, and Status.

Validation

Choosing the Threshold

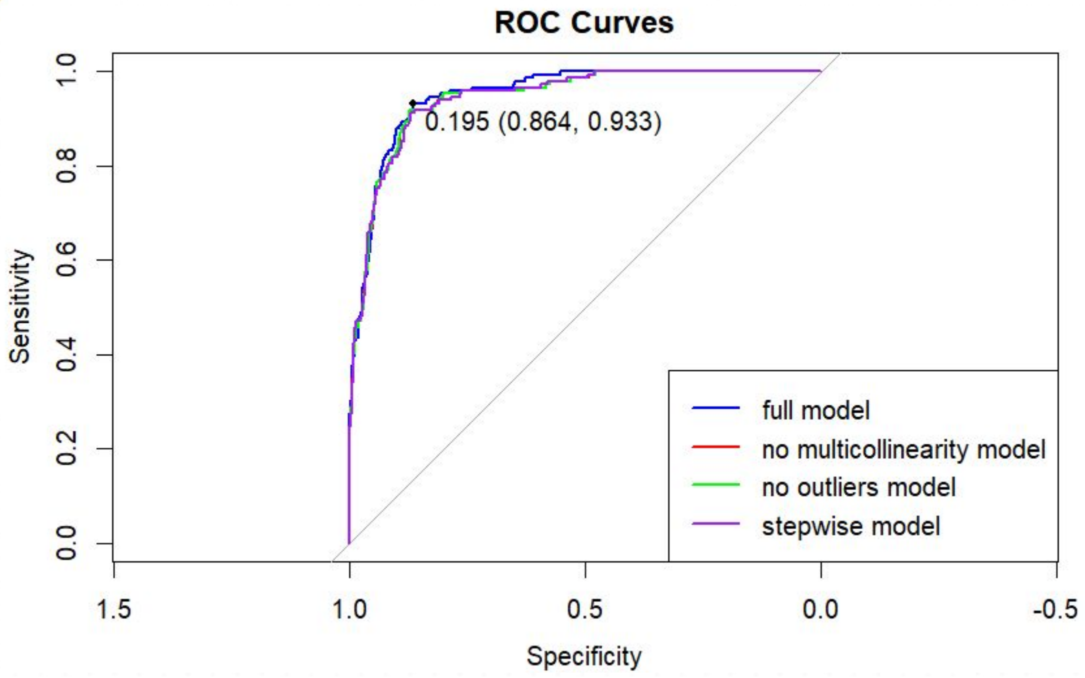


Assuming the cost of acquiring a customer is much greater than keeping a customer we focus on recall and accuracy while selecting the optimal threshold.

For our **stepwise model** (chart) we choose 0.19 as the best threshold.

If the cost of acquiring customers wasn't so significant, a better threshold value would be 0.37.

ROC Curves



The ROC Curves show the balance between Sensitivity and Specificity.

We observe minor differences in the performance of the models.

Model Comparison based on Test data

	Threshold	Accuracy	Precision	Recall	F1-Score	Importance	GOF
Full Model	0.20	0.878	0.571	0.919	0.704	Yes	Yes
Full Model (No Multicollinearity)	0.19	0.879	0.573	0.919	0.706	Yes	Yes
Full Model(No Outliers)	0.19	0.879	0.573	0.919	0.706	Yes	Yes
Stepwise Model	0.18	0.876	0.567	0.913	0.70	Yes	Yes
Random Forest Model (extra model)	0.32	0.959	0.831	0.926	0.88	-	-

Based on the above, we see that the models are all pretty close to each other (except for the Random Forest model). Thus, we decide to choose the **stepwise model** as our final model, because it's simpler and more robust (more interpretable, more likely to generalize well on new data, reduced risk of overfitting, computational efficiency).

Conclusion

Preferred model: Logistic regression - Stepwise

```
Call:
glm(formula = Churn ~ Status + Complains + Frequency.of.use +
    Call..Failure + Frequency.of.SMS + Charge..Amount + avgusage.per.month +
    avgcalls.per.Number + call.failure.rate, family = "binomial",
    data = train_data2)

Coefficients:
                     Estimate Std. Error z value Pr(>|z|)
(Intercept)         -1.580629   0.295678  -5.346  9.0e-08 ***
Status2              1.149371   0.278587   4.126  3.7e-05 ***
Complains1           3.914692   0.412679   9.486  < 2e-16 ***
Frequency.of.use    -0.035566   0.006269  -5.674  1.4e-08 ***
Call..Failure        0.119163   0.029891   3.987  6.7e-05 ***
Frequency.of.SMS    -0.017636   0.005367  -3.286 0.001017 **
Charge..Amount      -0.503015   0.139732  -3.600 0.000318 ***
avgusage.per.month   0.004351   0.001210   3.594 0.000325 ***
avgcalls.per.Number -0.104702   0.060499  -1.731 0.083515 .
call.failure.rate    0.828733   0.579616   1.430 0.152775
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1197.37  on 1399  degrees of freedom
Residual deviance:  580.44  on 1390  degrees of freedom
AIC: 600.44
```

Complaints : The most important predictor by far, the odds are 50 times higher when there has been a complaint

Status: The odds are 3.15 times higher when the customer is inactive

Charge amount: For a one unit increase in the charge amount (1 category up) the customer is 40% less likely to leave at the end of the year

Call failure: The odds are 1.13 times higher for each failure

How to prevent customer churn ?

- React immediately when there has been a complaint: commercial gestures, incentives to encourage to stay,...
- Higher-priced plans are associated with lower churn rates → Customer perceive greater value in these plans? It is worth exploring if the quality and the attractiveness of the offerings scale proportionately with the price
- Investigate repeated call failures: one may not be significant but multiples can cumulatively become a major issue

Future directions

- Consider new features: region, competitor information, plan details, device information & other usage patterns.
- Explore other models: Decision trees, SVM, KNN, deep learning models,...
- Time Series Analysis: Identify seasonal pattern or other temporal factors
- More data points: Help the model learn more complex patterns, make more accurate predictions

Any questions ?