DATA621 Final Project: Predicting Customer Churn

Eric Lehmphul

5/14/2022

Abstract

Customer churn is problematic for businesses as they lose out on revenue every time a customer abandons the business. The objective of this paper is to create classification models to predict customer churn. Many classification models have been applied to churn detection in the past. The models used in this paper are binary logistic regression and naive bayes classifier. Most customer churn datasets contain data imbalance in the target variable. The dataset used in this paper is no different. External data balancing techniques were applied to the dataset and tested through the machine learning models allowing for the comparison of the data balancing techniques. The logistic regression model performed best using undersampled data, whereas the naive bayes model performed best using the oversampled data. The ability to predict customer churn using the unbalanced data before undergoing undersampling or oversampling yielded the worst results. The best model was Model 3 - "Logistic Regression with Undersampled Data", which was able to achieve a precision score of 0.782 maintained a fairly high recall and f1 score.

Keywords: customer churn, class imbalance, logistic regression, naive bayes, classification

Introduction

Businesses rely of customers to provide the revenues needed to achieve profitability. Typically, it is more expensive to acquire a new customer than to retain an existing customer. According to an article in the *Harvard Business Review* written by Amy Gallo, it is "anywhere from 5 to 25 times more expensive" attracting new customers than retaining loyal customers. Customer churn is a real concern for almost all businesses, as losing customers is directly related to higher customer acquisition costs and loss in revenue. The rise of big data collection by companies allows the use of demographic, account, service, and activity information to be leveraged to create predictive models that determine the likelihood of a customer churning.

The objective of this project is to predict whether a customer will change telecommunication service providers given information about the customer's engagement, telephone plan, and location. The dataset of interest contains information about customers of an undisclosed telecommunication provider and was obtained from Kaggle.com via the "Customer Churn Prediction 2020" contest. The motivation for conducting a detailed analysis on customer churn prediction is from the ability to genrate important business insights that will aid businesses in improving performance.

Literature Review

Customer churn is well known for having an unbalanced class distribution. Most customers are loyal to the business with only a percentage of customers churning. Data imbalance, huge volumes, and high dimensionality in telecommuncation data makes it arduous to draw meaningful and actionable insights (Eria

& Marikannan, 2018). There are three significant developing techniques that have been applied to balancing imbalanced classes in literature related to customer churn: External, Algorithmic / internal, and Cost-sensitive (Ali et al., 2019). The external approach focuses on rebalancing the data through modifying the dataset rather than adjusting the learning method of the machine learning model (Davarynejad, 2017). Internal approaches adjust or create algorithms to be able to handle imbalanced classification problems, such as modifying the decision threshold or creating a new cost function to add bias toward the minority class (Davarynejad, 2017). The last technique combines both the external and internal approaches (Davarynejad, 2017).

As customer churn is a classification problem, there are many available modeling techniques that one could use. Dahiya and Bhatia implemented decision tree and logistic regression models to predict churn and found decision trees to be more effective (Dahiya & Bhatia, 2015). Almana et al. used a neural networks, statistical models, decision trees, and covering algorithms for predicting churn (Almana et al., 2014). Yi Fei et al. applied Naive Bayes Classifier and K-means to detect customer churn (Yi Fei et al., 2017). These are only a handful of examples of different modeling techniques used in previous literature related to classifying customer churn. This shows that there are many different ways to approach a classification based churn problem.

Methodology

The dataset is composed of a variety of metadata related to the customer including activity levels and geographic location (can be found at https://www.kaggle.com/competitions/customer-churn-prediction-2020/data?select=train.csv). The target variable, `churn`, is a binary indicator variable representing if a customer left or stayed. As mentioned in the Literature Review section, customer churn data is highly unbalanced and should be adjusted to conduct less bias models. External data balancing techniques were used to prepare the data for modeling. This study utilized both undersampling and oversampling data balancing approaches and compared the modeling results between the two methods. Binary logistic regression and naive bayes classifier models were used to predict telecommunication customer churn. The main metrics for determining the best model include precision, recall, f1 score. Precision will be the most important factor as the goal of the project is to predict customer churn which is a rare event. If accuracy is used, a model that only predicts that a customer will stay will yield a high accuracy, but is terrible at detecting customer churn. This is why precision is the preferred metric.

Experimentation and Results

Data Descriptions

Variable	Data Type	Description
state	String	2-letter code of the US state of customer residence
account_length	Numeric	Number of months the customer has been with the current telco provider
area_code	String	3 digit area code of customer
international_plan	String	Indicator variable to identify if customer has an international plan
voice_mail_plan	String	Indicator variable to identify if customer has a voice mail plan
number_vmail_messages	Numeric	Number of voice mail messages recieved by customer
total_day_minutes	Numeric	Total minutes of day calls
total_day_calls	Numeric	Total number of day calls

Variable	Data Type	Description
total_day_charge	Numeric	Total charge of day calls
total_eve_minutes	Numeric	Total minutes of evening calls
total_eve_calls	Numeric	Total number of evening calls
total_eve_charge	Numeric	Total charge of evening calls
total_night_minutes	Numeric	Total minutes of night calls
total_night_calls	Numeric	Total number of night calls
total_night_charge	Numeric	Total charge of night calls
total_intl_minutes	Numeric	Total minutes of international calls
total_intl_calls	Numeric	Total number of international calls
total_intl_charge	Numeric	Total charge of international calls
number_customer_service_calls	Numeric	Number of calls to customer service
churn	String	Binary indicator variable to identify if customer churned

Data Exploration

Data Summary

There are no missing values present in this data source. Most of the numeric data are counts of customer activity, such as the number of phone calls made and the and the number of minutes used. Other variables pertain to metadata related to the customer, like state and area code.

Table 2: Data summary

Name	churn.data
Number of rows	4250
Number of columns	20
Column type frequency:	
factor	5
numeric	15
Group variables	None

Variable type: factor

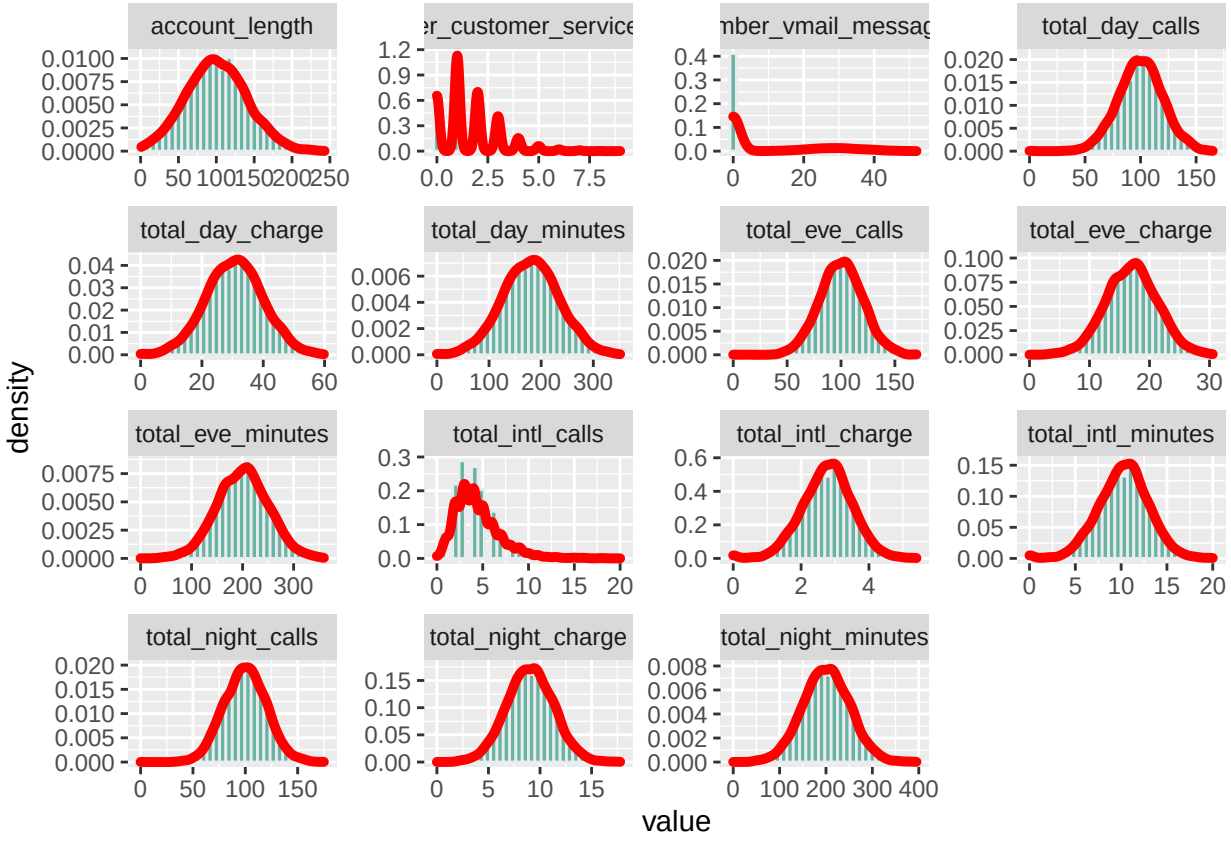
skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
state	0	1	FALSE	51	WV: 139, MN: 108, ID: 106, AL: 101
area_code	0	1	FALSE	3	415: 2108, 408: 1086, 510: 1056
international_plan	0	1	FALSE	2	no: 3854, yes: 396
voice_mail_plan	0	1	FALSE	2	no: 3138, yes: 1112
churn	0	1	FALSE	2	no: 3652, yes: 598

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	median	min	max
account_length	0	1	100.24	39.70	100.00	1	243.00
number_vmail_messages	0	1	7.63	13.44	0.00	0	52.00
total_day_minutes	0	1	180.26	54.01	180.45	0	351.50
total_day_calls	0	1	99.91	19.85	100.00	0	165.00
total_day_charge	0	1	30.64	9.18	30.68	0	59.76
total_eve_minutes	0	1	200.17	50.25	200.70	0	359.30
total_eve_calls	0	1	100.18	19.91	100.00	0	170.00
total_eve_charge	0	1	17.02	4.27	17.06	0	30.54
total_night_minutes	0	1	200.53	50.35	200.45	0	395.00
total_night_calls	0	1	99.84	20.09	100.00	0	175.00
total_night_charge	0	1	9.02	2.27	9.02	0	17.77
total_intl_minutes	0	1	10.26	2.76	10.30	0	20.00
total_intl_calls	0	1	4.43	2.46	4.00	0	20.00
total_intl_charge	0	1	2.77	0.75	2.78	0	5.40
number_customer_service_calls	0	1	1.56	1.31	1.00	0	9.00

Distribution for Numeric Variables

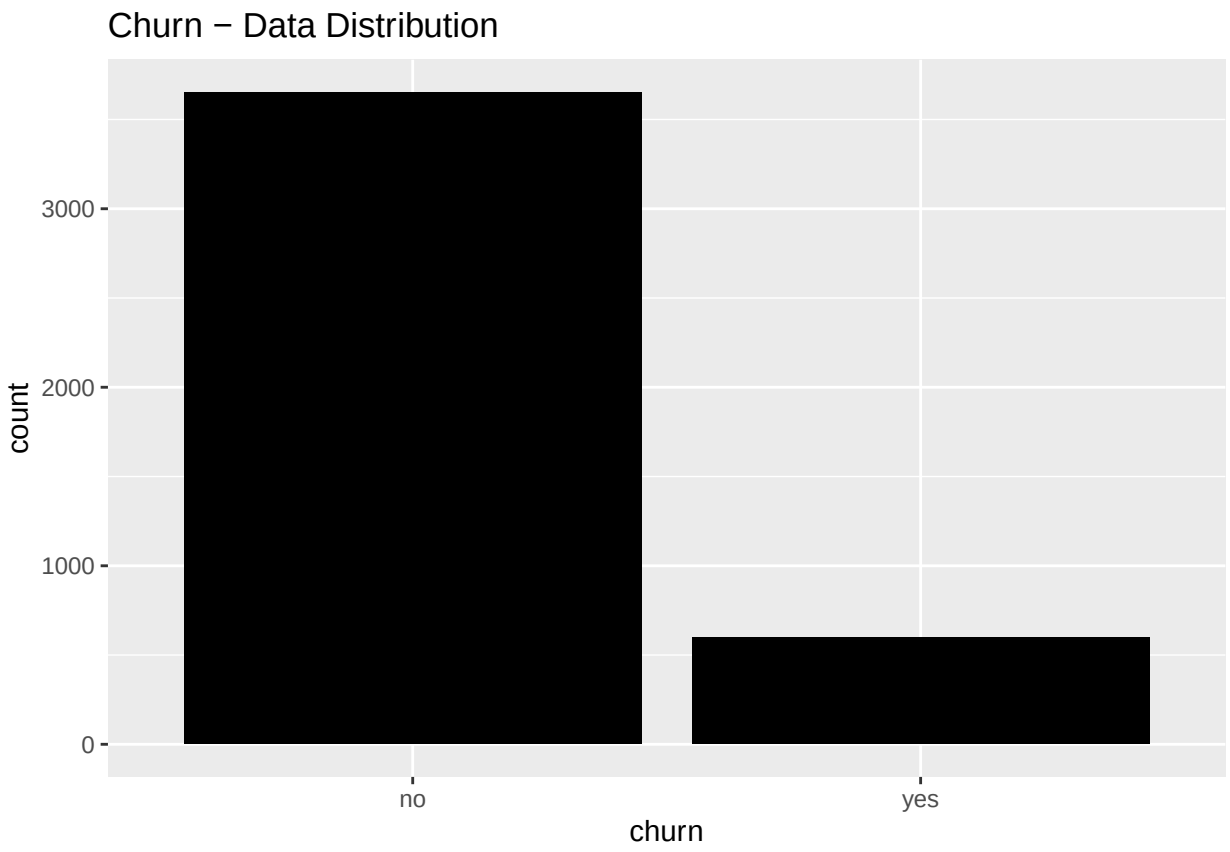
Most of the variables appear to follow a near normal data distribution. The variables `number_customer_service_calls` and `number_vmail_messages` are count variables with right skewed distributions.



Check for Data Imbalance in Target Variable - churn

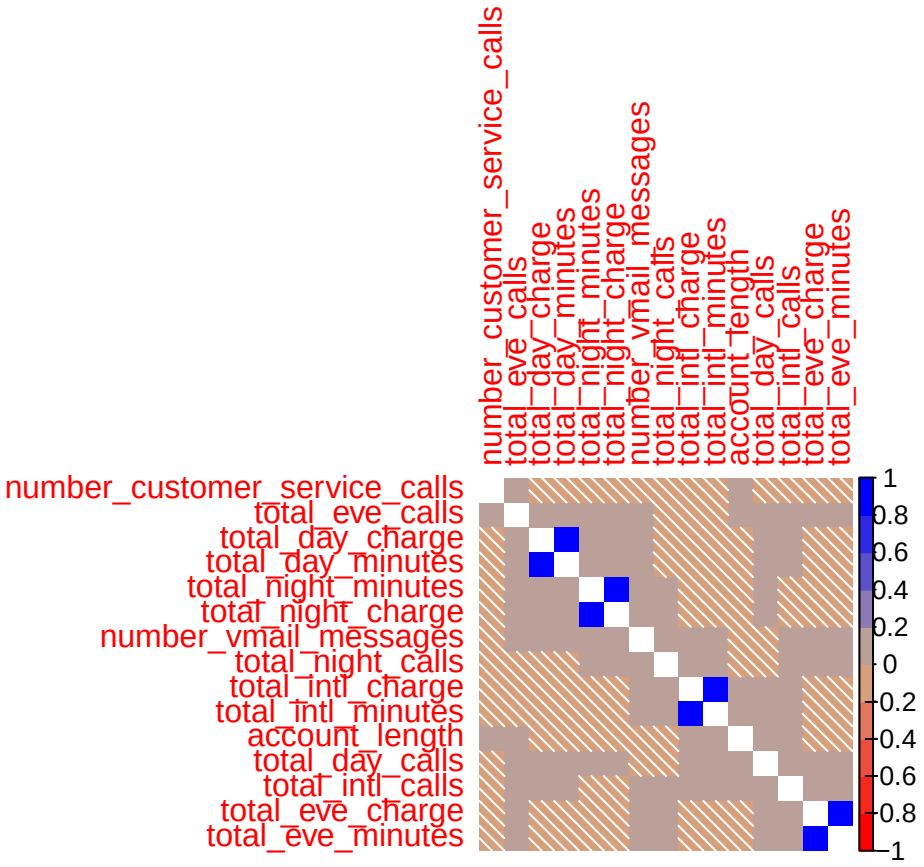
The target variable is dominated by the "no" class, indicating that customer churn is a rare event to occur.

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```



Correlation Matrix

Most of the variables are not highly correlated with eachother. As expected, the variables that are highly correlated are the number of phone calls and the total amount charged to the customer.



Data Preprocessing

Adjusted Data Types

Five of the variables were stored in the incorrect data type. The most common error was that the categorical variables were stored as a string data type instead of a factor data type. Below is a list of the adjusted variables.

- String to Factor Data Type
  - `state`
  - `area_code`
  - `international_plan`
  - `voice_mail_plan`
  - `churn`

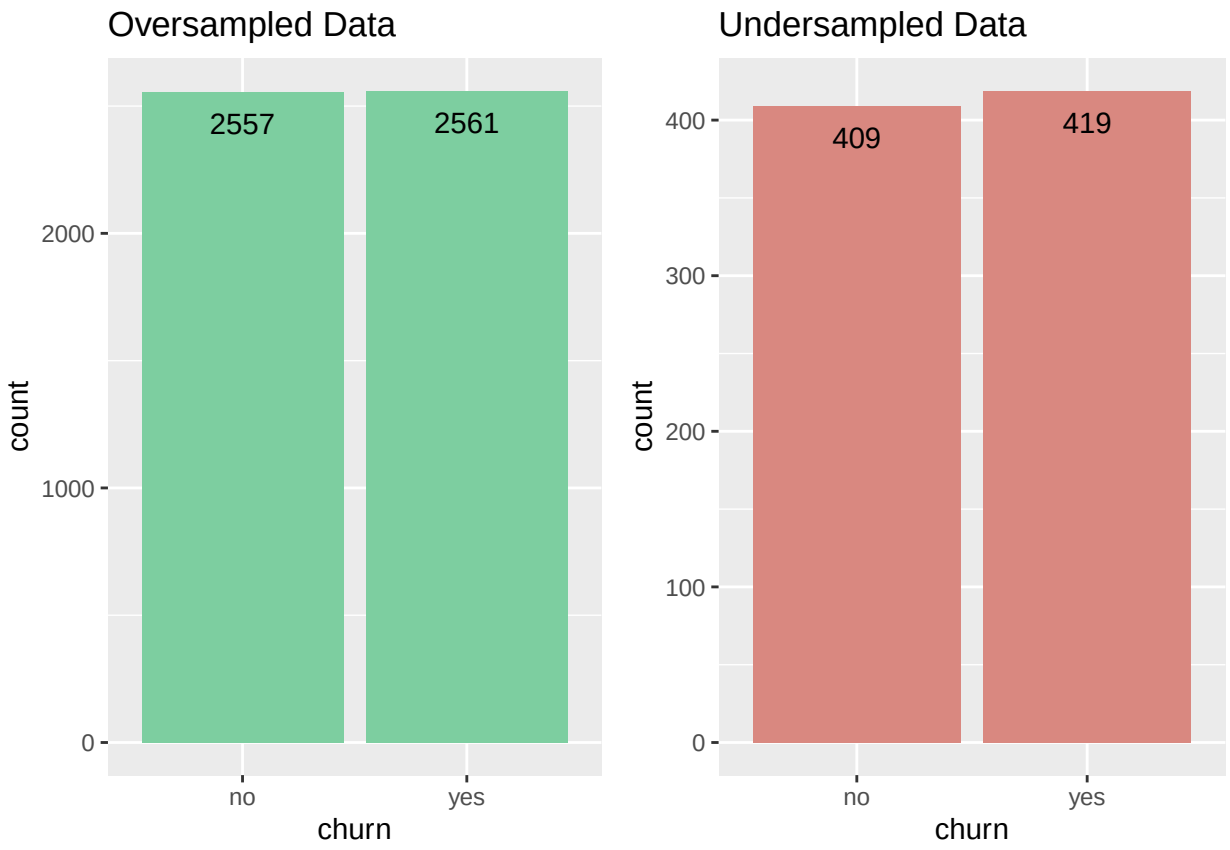
Modified Area Code Variable

The variable, `area_code`, was stored as 'area_code_###' where ### is a 3 digit area code. I modified this variable to only include the 3 digit number by removing 'area_code_' in all rows.

Create Balanced Dataset for modeling

As shown in the Data Exploration, `churn` has a large data imbalance. This tends to be problematic for building effective classification models if left unaccounted for. I employed both undersampling and oversam-

pling techniques to be used on the training dataset to build classification models. With imbalanced data, accuracy is not the best metric to use. I will be using F1 score as the main method of model evaluation.



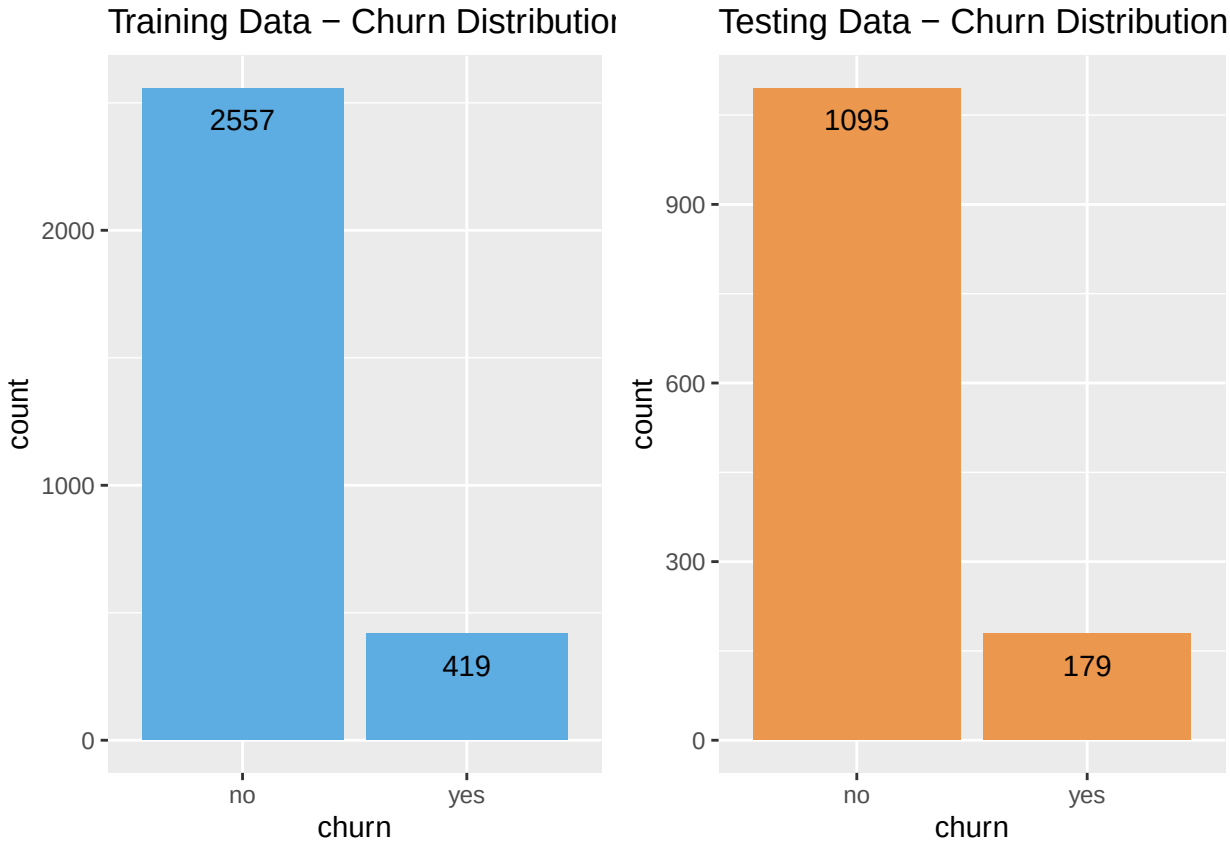
Modeling

Six models were built to produce an effective classifier of customer churn in the telecommunication sector. The first three models are created using a logistic regression algorithm and the last three models use a naive bayes classifier to predict customer churn. To keep the models consistent, both algorithms were trained with the same three training datasets (Original training data, Oversampled data, and undersampled data). The Logistic regression and naive bayes classifier use the same dependent variables for the corresponding data source. For example, Model 1 and Model 4 are both using the original training data, so the model formula will be identical. The models are using different datasets and algorithms from one another, hindering the use of AIC, BIC, etc. to assess model performance, Instead, I will use metrics obtained by the confusion matrices and ROC curves to compare model performance. The following models were explored in this experimentation:

- Model 1: Unbalanced Data Logistic Regression
- Model 2: Oversampled Data Logistic Regression
- Model 3: Undersampled Data Logistic Regression
- Model 4: Unbalanced Data Naive Bayes
- Model 5: Oversampled Data Naive Bayes
- Model 6: Undersampled Data Naive Bayes

Train Test Split

The customer churn dataset was divided into a train and test set to be able to assess model performance. The training dataset represents 70% of the customer churn data and the test contains the remaining 30%. Stratified random sampling was used to obtain the train and test datasets as it is important to guarantee that the target variable `churn` is represented equally in both datasets.



Logistic Regression Models

Model 1 - Unbalanced Data Logistic Regression

```
##
## Call:
## glm(formula = churn ~ international_plan + voice_mail_plan +
##     total_day_minutes + total_eve_minutes + total_night_minutes +
##     total_intl_minutes + total_intl_calls + number_customer_service_calls,
##     family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9206  -0.4992  -0.3293  -0.1802   3.2181
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   -8.478156   0.560295 -15.132  < 2e-16 ***
```

```
## international_planyes          2.046960   0.158415  12.921  < 2e-16 ***
## voice_mail_planyes            -1.315067   0.175455  -7.495 6.62e-14 ***
## total_day_minutes              0.014007   0.001205  11.620  < 2e-16 ***
## total_eve_minutes              0.007045   0.001239   5.684 1.31e-08 ***
## total_night_minutes            0.003605   0.001185   3.042  0.00235 **
## total_intl_minutes             0.094413   0.022258   4.242 2.22e-05 ***
## total_intl_calls              -0.055247   0.025773  -2.144  0.03207 *
## number_customer_service_calls  0.540210   0.042287  12.775  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2418.9  on 2975  degrees of freedom
## Residual deviance: 1863.7  on 2967  degrees of freedom
## AIC: 1881.7
##
## Number of Fisher Scoring iterations: 6
```

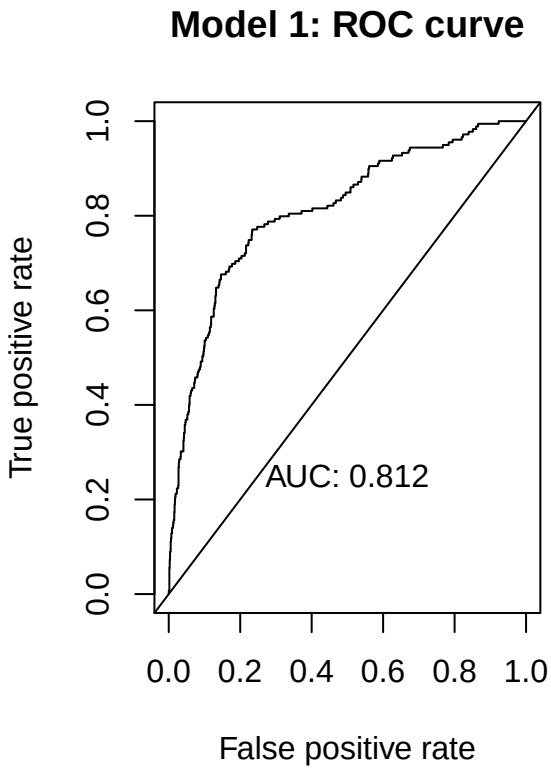
Variance Inflation Factor

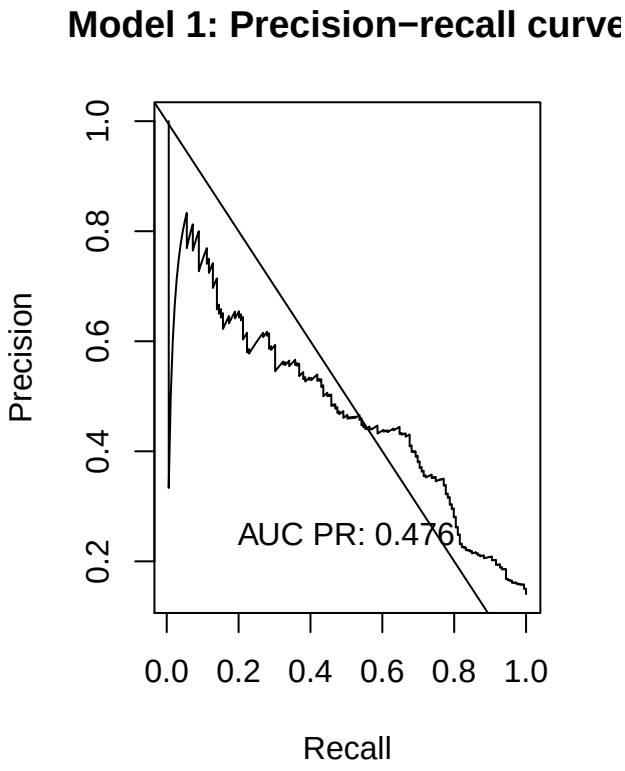
```
##            international_plan                voice_mail_plan
##                      1.065549                       1.026758
##             total_day_minutes             total_eve_minutes
##                      1.063396                       1.021508
##           total_night_minutes            total_intl_minutes
##                      1.013535                       1.013279
##              total_intl_calls number_customer_service_calls
##                      1.004294                       1.092314
```

Confusion Matrix and ROC Curves

```
## Confusion Matrix and Statistics
##
##      pred.1
##         no  yes
##   no  1065   30
##   yes  138   41
##
##                Accuracy : 0.8681
##                  95% CI : (0.8483, 0.8862)
##     No Information Rate : 0.9443
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.2697
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.57746
##             Specificity : 0.88529
##          Pos Pred Value : 0.22905
##          Neg Pred Value : 0.97260
##              Prevalence : 0.05573
```

```
##          Detection Rate : 0.03218
##    Detection Prevalence : 0.14050
##       Balanced Accuracy : 0.73138
##
##        'Positive' Class : yes
##
```





Model 2 - Oversampled Data

```
##
## Call:
## glm(formula = churn ~ state + account_length + international_plan +
##     voice_mail_plan + number_vmail_messages + total_day_charge +
##     total_eve_minutes + total_eve_calls + total_night_calls +
##     total_night_charge + total_intl_minutes + total_intl_calls +
##     number_customer_service_calls, family = "binomial", data = over)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.81635  -0.73504   0.03395   0.79398   2.65400
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -8.097497   0.551889 -14.672  < 2e-16 ***
## stateAL                      -0.145123   0.416386  -0.349  0.72744
```

```
## stateAR                    0.126271   0.454290   0.278  0.78105
## stateAZ                   -0.500539   0.453763  -1.103  0.26999
## stateCA                    2.393635   0.475560   5.033 4.82e-07 ***
## stateCO                   -0.369571   0.458096  -0.807  0.41981
## stateCT                    0.592898   0.420491   1.410  0.15854
## stateDC                    0.092983   0.487304   0.191  0.84867
## stateDE                    0.467016   0.412624   1.132  0.25771
## stateFL                    0.192932   0.425407   0.454  0.65017
## stateGA                   -0.106217   0.473677  -0.224  0.82257
## stateHI                   -1.389926   0.542436  -2.562  0.01040 *
## stateIA                   -0.191338   0.452432  -0.423  0.67236
## stateID                    0.447659   0.400722   1.117  0.26394
## stateIL                   -0.360046   0.462473  -0.779  0.43626
## stateIN                    0.652253   0.423377   1.541  0.12341
## stateKS                    0.874065   0.403804   2.165  0.03042 *
## stateKY                    0.868234   0.405340   2.142  0.03219 *
## stateLA                    0.320863   0.437839   0.733  0.46366
## stateMA                    0.372500   0.420892   0.885  0.37614
## stateMD                    0.679372   0.413917   1.641  0.10073
## stateME                    0.882115   0.413184   2.135  0.03277 *
## stateMI                    0.359691   0.416798   0.863  0.38814
## stateMN                    0.454689   0.405019   1.123  0.26159
## stateMO                    0.017297   0.423759   0.041  0.96744
## stateMS                    0.538815   0.429011   1.256  0.20914
## stateMT                    1.253263   0.409766   3.058  0.00222 **
## stateNC                   -0.772439   0.506408  -1.525  0.12718
## stateND                    0.452552   0.437058   1.035  0.30046
## stateNE                   -0.270141   0.475607  -0.568  0.57004
## stateNH                    0.199451   0.421829   0.473  0.63634
## stateNJ                    1.205941   0.402508   2.996  0.00273 **
## stateNM                    0.048972   0.430295   0.114  0.90939
## stateNV                    0.459558   0.409436   1.122  0.26168
## stateNY                    0.472861   0.413689   1.143  0.25302
## stateOH                    0.308141   0.419145   0.735  0.46224
## stateOK                    0.449112   0.417152   1.077  0.28165
## stateOR                   -0.026835   0.415974  -0.065  0.94856
## statePA                    0.091093   0.438469   0.208  0.83542
## stateRI                   -0.525512   0.463047  -1.135  0.25642
## stateSC                    1.393526   0.428610   3.251  0.00115 **
## stateSD                    0.082118   0.474218   0.173  0.86252
## stateTN                    0.434488   0.429450   1.012  0.31167
## stateTX                    1.099812   0.393665   2.794  0.00521 **
## stateUT                    0.554035   0.404595   1.369  0.17089
## stateVA                   -1.171006   0.479489  -2.442  0.01460 *
## stateVT                   -0.967843   0.469809  -2.060  0.03939 *
## stateWA                    1.352128   0.415538   3.254  0.00114 **
## stateWI                   -0.019913   0.422596  -0.047  0.96242
## stateWV                    0.520364   0.389935   1.334  0.18204
## stateWY                    0.014464   0.432550   0.033  0.97332
## account_length             0.001887   0.000921   2.049  0.04043 *
## international_planyes      2.473339   0.115160  21.477  < 2e-16 ***
## voice_mail_planyes        -2.284944   0.373294  -6.121 9.30e-10 ***
## number_vmail_messages      0.030377   0.011829   2.568  0.01023 *
## total_day_charge           0.086509   0.004018  21.532  < 2e-16 ***
```

```
## total_eve_minutes              0.008497   0.000756  11.240  < 2e-16 ***
## total_eve_calls                0.002962   0.001827   1.621  0.10509
## total_night_calls             -0.003849   0.001852  -2.079  0.03766 *
## total_night_charge             0.086525   0.016740   5.169 2.36e-07 ***
## total_intl_minutes             0.100336   0.013631   7.361 1.83e-13 ***
## total_intl_calls              -0.042586   0.014660  -2.905  0.00367 **
## number_customer_service_calls  0.674702   0.027394  24.630  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 7095.1  on 5117  degrees of freedom
## Residual deviance: 4835.2  on 5055  degrees of freedom
## AIC: 4961.2
##
## Number of Fisher Scoring iterations: 5
```

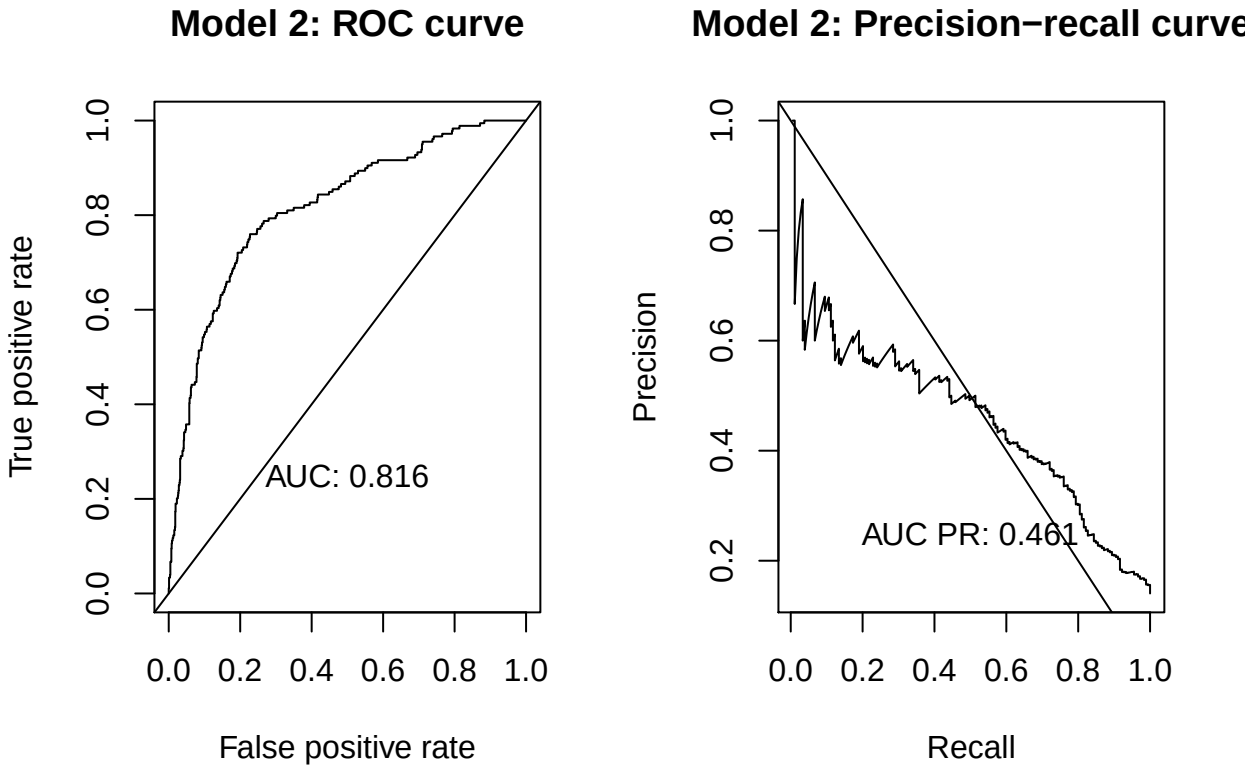
Variance Inflation Factor

```
##                                  GVIF Df GVIF^(1/(2*Df))
## state                        1.906792 50        1.006475
## account_length               1.062446  1        1.030750
## international_plan           1.184290  1        1.088251
## voice_mail_plan             15.205799  1        3.899461
## number_vmail_messages       15.147540  1        3.891984
## total_day_charge             1.314151  1        1.146364
## total_eve_minutes            1.125660  1        1.060971
## total_eve_calls              1.056905  1        1.028059
## total_night_calls            1.067281  1        1.033093
## total_night_charge           1.060378  1        1.029747
## total_intl_minutes           1.097191  1        1.047469
## total_intl_calls             1.102806  1        1.050146
## number_customer_service_calls  1.415084  1        1.189573
```

Confusion Matrix and ROC Curves

```
## Confusion Matrix and Statistics
##
##      pred.2
##        no yes
##   no  855 240
##   yes  47 132
##
##                Accuracy : 0.7747
##                  95% CI : (0.7508, 0.7974)
##     No Information Rate : 0.708
##     P-Value [Acc > NIR] : 4.762e-08
##
##                   Kappa : 0.3572
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
```

```
##             Sensitivity : 0.3548
##             Specificity : 0.9479
##          Pos Pred Value : 0.7374
##          Neg Pred Value : 0.7808
##              Prevalence : 0.2920
##          Detection Rate : 0.1036
##    Detection Prevalence : 0.1405
##       Balanced Accuracy : 0.6514
##
##        'Positive' Class : yes
##
```



Model 3 - Undersampled Data

```
##
## Call:
## glm(formula = churn ~ international_plan + voice_mail_plan +
##     total_day_charge + total_eve_charge + total_night_calls +
##     total_intl_minutes + number_customer_service_calls, family = "binomial",
##     data = under)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6265  -0.7759   0.1115   0.8377   2.6604
##
```

```
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  -6.936162   0.815145  -8.509  < 2e-16 ***
## international_planyes         2.239218   0.262514   8.530  < 2e-16 ***
## voice_mail_planyes           -0.931563   0.234052  -3.980 6.89e-05 ***
## total_day_charge              0.096926   0.009854   9.836  < 2e-16 ***
## total_eve_charge              0.099814   0.020992   4.755 1.99e-06 ***
## total_night_calls            -0.007242   0.004379  -1.654 0.098139 .
## total_intl_minutes            0.123965   0.032472   3.818 0.000135 ***
## number_customer_service_calls  0.741656   0.068628  10.807  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1147.73  on 827  degrees of freedom
## Residual deviance:  810.96  on 820  degrees of freedom
## AIC: 826.96
##
## Number of Fisher Scoring iterations: 5
```

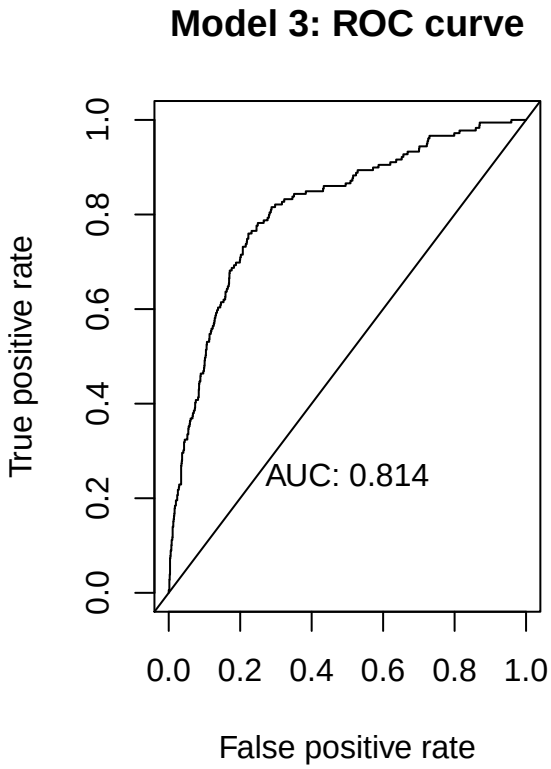
Variance Inflation Factor

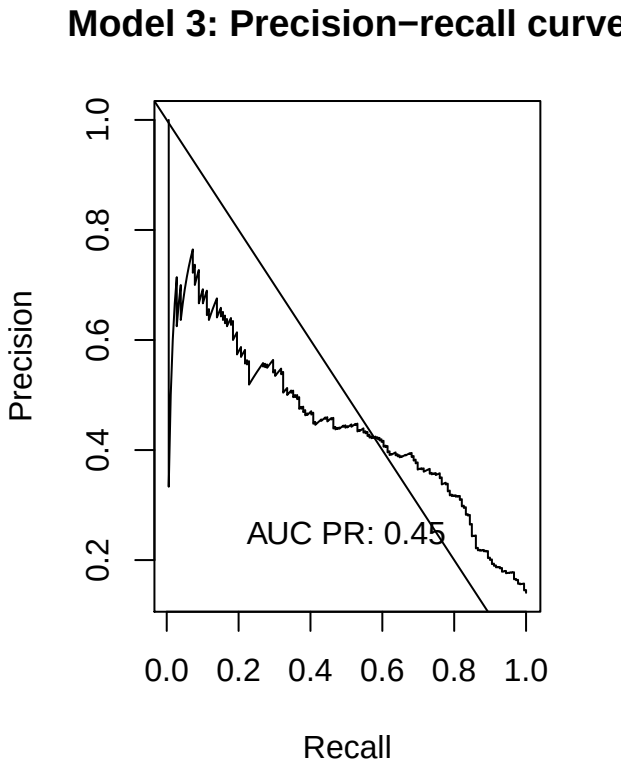
```
##            international_plan               voice_mail_plan
##                      1.119927                      1.024505
##              total_day_charge              total_eve_charge
##                      1.321296                      1.064832
##             total_night_calls           total_intl_minutes
##                      1.014916                      1.030463
## number_customer_service_calls
##                      1.418679
```

Confusion Matrix and ROC Curves

```
## Confusion Matrix and Statistics
##
##      pred.3
##        no yes
##   no  817 278
##   yes  39 140
##
##                Accuracy : 0.7512
##                  95% CI : (0.7265, 0.7747)
##     No Information Rate : 0.6719
##     P-Value [Acc > NIR] : 4.152e-10
##
##                   Kappa : 0.3389
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.3349
##             Specificity : 0.9544
##          Pos Pred Value : 0.7821
```

```
##          Neg Pred Value : 0.7461
##              Prevalence : 0.3281
##          Detection Rate : 0.1099
##    Detection Prevalence : 0.1405
##       Balanced Accuracy : 0.6447
##
##        'Positive' Class : yes
##
```





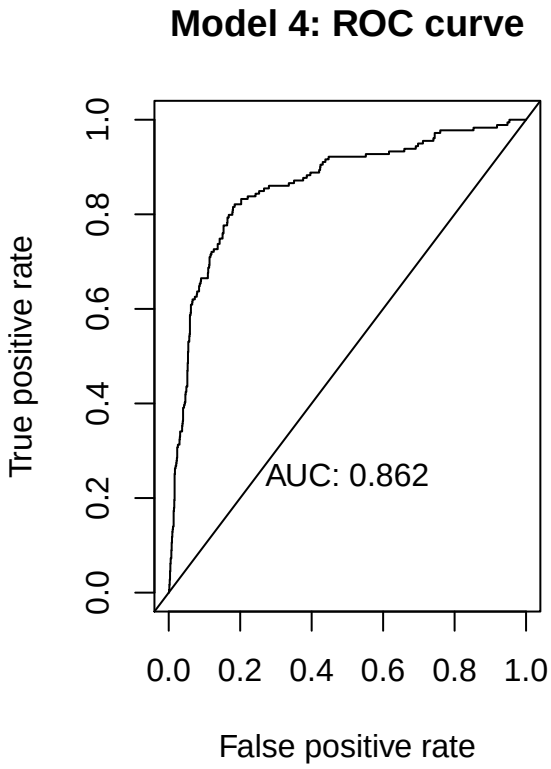
Naive Bayes Classifier

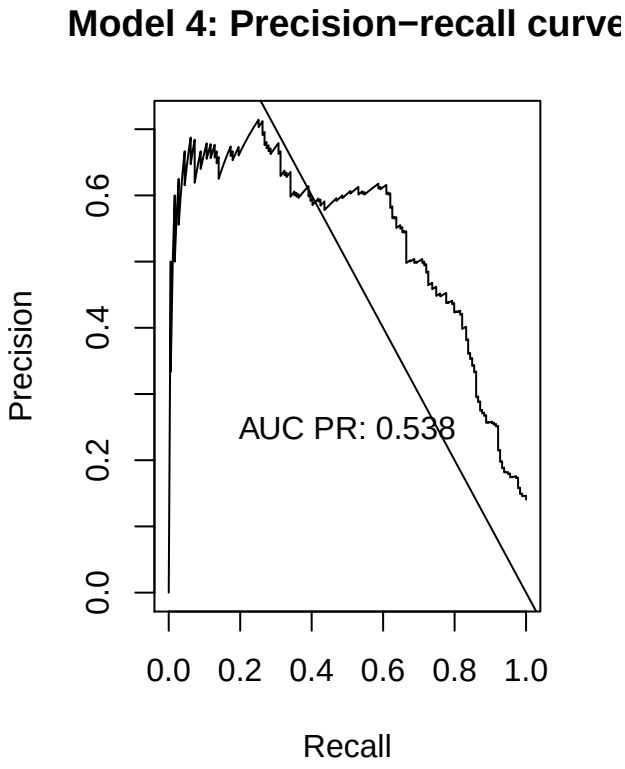
Model 4 - Unbalanced Data

Confusion Matrix and ROC Curves

```
## Confusion Matrix and Statistics
##
##      y_pred
##         no  yes
##   no  1074   21
##   yes  131   48
##
##                Accuracy : 0.8807
##                  95% CI : (0.8616, 0.898)
```

```
##     No Information Rate : 0.9458
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.3351
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.69565
##             Specificity : 0.89129
##          Pos Pred Value : 0.26816
##          Neg Pred Value : 0.98082
##              Prevalence : 0.05416
##          Detection Rate : 0.03768
##    Detection Prevalence : 0.14050
##       Balanced Accuracy : 0.79347
##
##        'Positive' Class : yes
##
```



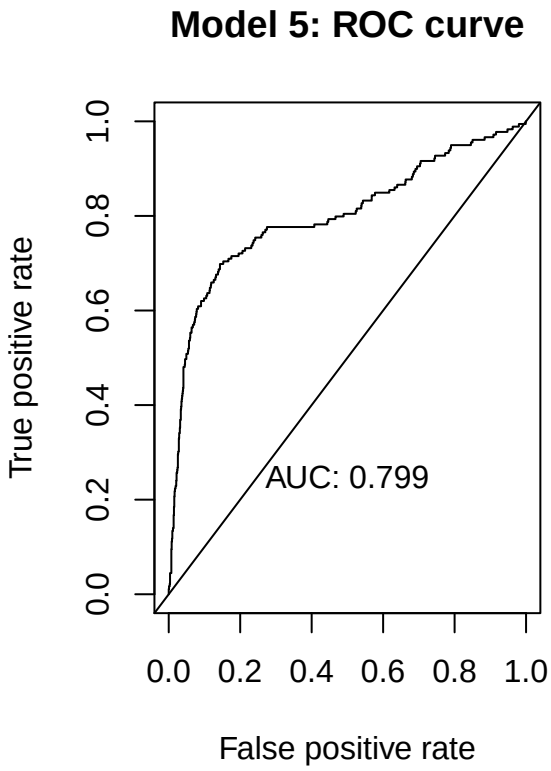


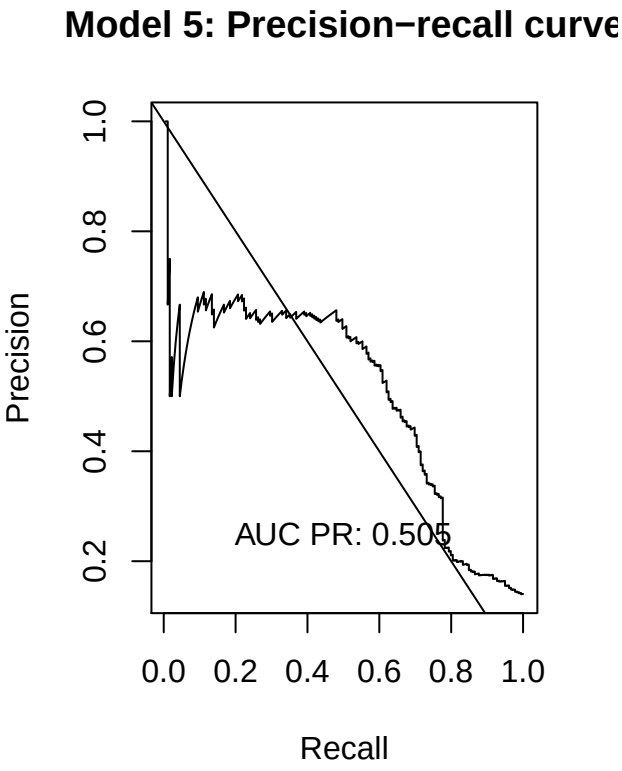

```
## NULL
```

Model 5 - Oversampled Data

Confusion Matrix and ROC Curves

```
## Confusion Matrix and Statistics
##
##      y_pred
##        no yes
##   no  828 267
##   yes  44 135
##
##                Accuracy : 0.7559
##                  95% CI : (0.7313, 0.7793)
##     No Information Rate : 0.6845
##     P-Value [Acc > NIR] : 1.201e-08
##
##                   Kappa : 0.3355
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.3358
##             Specificity : 0.9495
##          Pos Pred Value : 0.7542
##          Neg Pred Value : 0.7562
##              Prevalence : 0.3155
##          Detection Rate : 0.1060
##    Detection Prevalence : 0.1405
##       Balanced Accuracy : 0.6427
##
##        'Positive' Class : yes
##
```





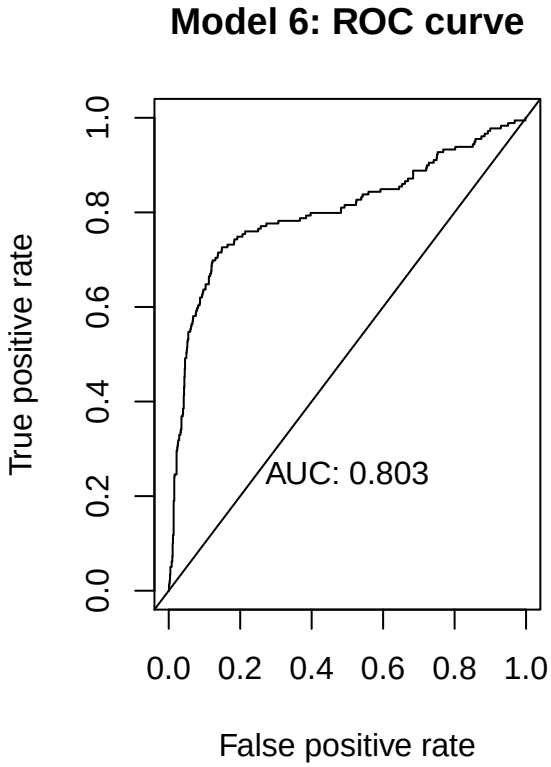

```
## NULL
```

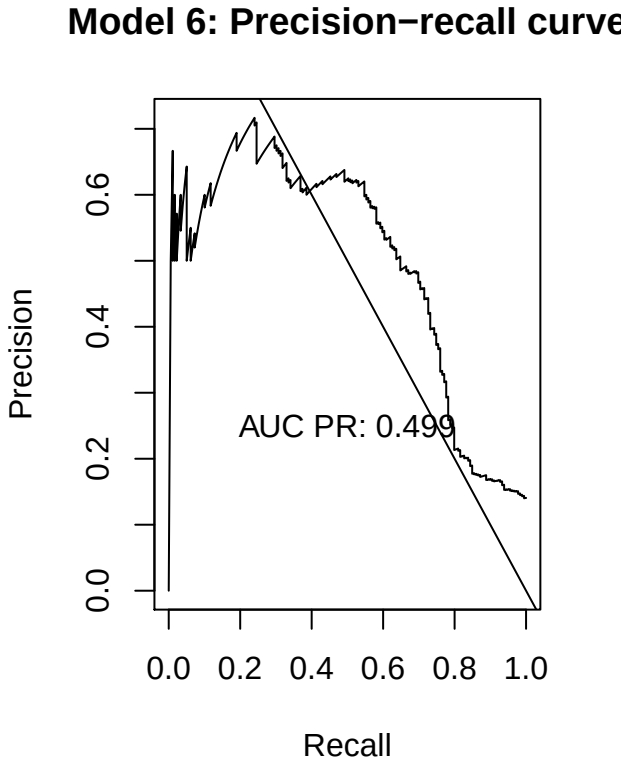
Model 6 - Undersampled Data

Confusion Matrix and ROC Curves

```
## Confusion Matrix and Statistics
##
##      y_pred
##        no yes
##   no  805 290
##   yes  41 138
##
##                Accuracy : 0.7402
##                  95% CI : (0.7152, 0.7641)
##     No Information Rate : 0.6641
##     P-Value [Acc > NIR] : 2.519e-09
##
##                   Kappa : 0.32
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.3224
##             Specificity : 0.9515
```

```
##          Pos Pred Value : 0.7709
##          Neg Pred Value : 0.7352
##              Prevalence : 0.3359
##          Detection Rate : 0.1083
##    Detection Prevalence : 0.1405
##       Balanced Accuracy : 0.6370
##
##        'Positive' Class : yes
##
```






```
## NULL
```

Evaluate Models

	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
Accuracy	0.868	0.775	0.751	0.881	0.756	0.740
Sensitivity	0.577	0.355	0.335	0.696	0.336	0.322
Specificity	0.885	0.948	0.954	0.891	0.950	0.952
Pos.Pred.Value	0.229	0.737	0.782	0.268	0.754	0.771
Neg.Pred.Value	0.973	0.781	0.746	0.981	0.756	0.735
Precision	0.229	0.737	0.782	0.268	0.754	0.771
Recall	0.577	0.355	0.335	0.696	0.336	0.322
F1	0.328	0.479	0.469	0.387	0.465	0.455
Prevalence	0.056	0.292	0.328	0.054	0.316	0.336
Detection.Rate	0.032	0.104	0.110	0.038	0.106	0.108
Detection.Prevalence	0.141	0.141	0.141	0.141	0.141	0.141
Balanced.Accuracy	0.731	0.651	0.645	0.793	0.643	0.637
AUC	0.812	0.816	0.814	0.862	0.799	0.803
AUC_PR	0.476	0.461	0.450	0.538	0.505	0.499

Results

The logistic regression and naive bayes models performed similarly across the 6 models. Logistic Regression outperformed the naive bayes classifier when using undersampled data (Model 3 vs Model 6) in terms of precision, recall, f1 score, and detection rate. The Naive Bayes classifier was a better model when using oversampled data (Model 2 vs Model 5) because it had a higher precision, recall, f1 score, and detection rate. The best model is Model 3 - Logistic Regression with undersampled data because it yeilds the highest precision in classifying that a customer will churn.

Discussion and Conclusion

The resulting models created in this study allow for customer churn to be detected with high precision. Given the same input features, a telecommunication company can detect potential customers of interest that are likely to leave. The telecommunication company can choose to either accept that the consumer will churn or they can take steps to retain the customer for continued service.

This study compared the techniques, undersampling and oversampling, for handling imbalanced data. Logistic regression was the best modeling algorithm when using undersampling techniques and naive bayes classifier was the best modeling algorithm when using oversampled data. The best results across all models was Model 3 - "Logistic Regression with Undersampled Data", which was able to achieve a precision score of 0.782. It should also be noted that the training dataset without any target class balancing techniques yielded the worst results for detecting customer churn.

References

Ali, H., Nahib Mohd Salleh, M., Saedudin, R., Hussain, K., & Mushtaq, M. F. (2019). Imbalance class problems in data mining: a review. Indonesian Journal of Electrical Engineering and Computer Science, 14(3), 1560–1571. https://doi.org/10.11591/ijeecs.v14.i3.pp1560-1571

Almana, A. M., Aksoy, M. S., & Alzahran, R. (2014). A Survey On Data Mining Techniques In Customer Churn Analysis For Telecom Industry. Int. Journal of Engineering Research and Applications, 4(5), 165–171. Retrieved from https://www.ijera.com/papers/Vol4_issue5/Version%206/AF4506165171.pdf.

Dahiya, K., & Bhatia, S. (2015). Customer churn analysis in telecom industry. IEEE. Retrieved from https://ieeexplore.ieee.org/abstract/document/7359318/authors#authors.

Davarynejad, M. (2017, July 29). Classification with imbalanced data: Internal and external approaches and selection performance metrics. Dr. Ir. Mohsen Davarynejad. Retrieved May 21, 2022, from https://behsys.com/mohsen/Classification-with-Imbalanced-Data-Internal-External-Approaches-Selection-Performance-Metrics.html#internal-approaches

Eria, K., & Marikannan, B. P. (2018). Systematic Review of Customer Churn Prediction in the Telecom. Journal of Applied Technology and Innovation, 2(1), 7–14. Retrieved from https://jati.sites.apiit.edu.my/files/2018/07/2018_Issue1_Paper2.pdf.

Gallo, A. (2014, November 5). The value of keeping the right customers. Harvard Business Review. Retrieved May 21, 2022, from https://hbr.org/2014/10/the-value-of-keeping-the-right-customers

Yi Fei, T., Hai Shuan, L., Jie Yan, L., Xiaoning, G., & Wooi King, S. (2017). Prediction on Customer Churn in the Telecommunications Sector Using Discretization and Naïve Bayes Classifier. Int. Journal Advanced Software Computer. Applications, 9(3), 24–35. Retrieved from https://www.i-csrs.org/Volumes/ijasca/2_Page-23_35_Predictive-Analysis-for-Telecommunications-Customer-Churn-on-Big-Data-Platform.pdf.

Appendices

```
library(tidyverse)
library(skimr)
library(corrplot)
library(caret)
library(ggpubr)
library(ROSE)
library(e1071)
library(modEvA)
library(ROCR)
library(PRROC)
library(pROC)
library(car)

churn.data <-  read.csv("https://raw.githubusercontent.com/SaneSky109/DATA621/main/Final_Project/Data/t

churn.data$area_code <- str_remove_all(churn.data$area_code, "area_code_")

churn.data$state <- as.factor(churn.data$state)
churn.data$area_code <- as.factor(churn.data$area_code)
churn.data$international_plan <- as.factor(churn.data$international_plan)
churn.data$voice_mail_plan <- as.factor(churn.data$voice_mail_plan)
churn.data$churn <- as.factor(churn.data$churn)

summary_table <- skim_with(numeric = sfl(median = ~ median(., na.rm = TRUE),
                                         min = ~ min(., na.rm = TRUE),
                                         max = ~ max(., na.rm = TRUE),
                                         hist = NULL, p0 = NULL, p25 = NULL,
                                         p50 = NULL, p75 = NULL, p100 = NULL))

summary_table(churn.data)
```

```
churn.data %>%
  gather(-c(state, area_code, international_plan, voice_mail_plan, churn), key = variable, value = valu
  ggplot(., aes(x = value)) +
  geom_histogram(aes(x=value, y = ..density..), bins = 30, fill="#69b3a2", color="#e9ecef") +
  geom_density(aes(x=value), color='red', lwd = 1.75) +
  facet_wrap(~variable, scales ="free", ncol = 4)

churn.data %>%
  ggplot(aes(x=churn)) + geom_histogram(stat="count",fill="black") +
  ggtitle("Churn - Data Distribution")

num.data <- churn.data[,-c(1,3,4,5,20)]
corrplot(cor(num.data), method = 'shade', order = 'AOE',col= colorRampPalette(c("red","tan", "blue"))(1

set.seed(15)

trainIndex <- createDataPartition(churn.data$churn, p = .7,
                                  list = FALSE,
                                  times = 1)

train <- churn.data[ trainIndex,]
test <- churn.data[-trainIndex,]

over <- ovun.sample(churn~., data = train, method = "over")$data

under <- ovun.sample(churn~., data=train, method = "under")$data

p1 <- over %>%
  ggplot(aes(x=churn)) + geom_bar(fill = "#7DCEA0") +
  geom_text(stat='count', aes(label=..count..), vjust=2) +
  ggtitle("Oversampled Data")

p2 <- under %>%
  ggplot(aes(x=churn)) + geom_bar(fill = "#D98880") +
  geom_text(stat='count', aes(label=..count..), vjust=2) +
  ggtitle("Undersampled Data")




ggarrange(p1,p2,
          ncol = 2, nrow = 1)

p1 <- train %>%
  ggplot(aes(x=churn)) + geom_bar(fill = "#5DADE2") +
  geom_text(stat='count', aes(label=..count..), vjust=2) +
  ggtitle("Training Data - Churn Distribution")

p2 <- test %>%
  ggplot(aes(x=churn)) + geom_bar(fill = "#EB984E") +
  geom_text(stat='count', aes(label=..count..), vjust=2) +
  ggtitle("Testing Data - Churn Distribution")
```

```
ggarrange(p1,p2,
          ncol = 2, nrow = 1)

all.variables <- glm(churn ~ .-total_day_charge, family = "binomial", data = train)

m1 <- step(all.variables, direction = "backward", trace = 0)

summary(m1)

vif(m1)

pred.1.raw <- predict(m1, type = "response", newdata = test)
pred.1 <- as.factor(ifelse(pred.1.raw < .5, "no", "yes"))

cm <- table(test$churn, pred.1)

cm1 <- confusionMatrix(cm, positive="yes")

cm1

plot.rocs1 <- function(model, model_num){
  par(mfrow=c(1,2))
  prednb <- predict(model, test, type='response')

pred<-prediction(prednb, test$churn, label.ordering = c("no", "yes"))
perf<-performance(pred,"tpr","fpr")
p1 <- plot(perf, main=paste0("Model ", model_num, ": ROC curve"))
abline(0,1)
auc <- performance(pred,"auc")
auc.val <- auc@y.values[[1]]
text(x = 0.5, y = 0.25, labels = paste0("AUC: ", round(auc.val,3)),
    cex = 1)

perf1 <- performance(pred, "prec", "rec")
aucpr<- performance(pred,"aucpr")
aucpr.val<- aucpr@y.values[[1]]
p2 <- plot(perf1, main=paste0("Model ", model_num, ": Precision-recall curve"))
abline(1,-1)
text(x = 0.5, y = 0.25, labels = paste0("AUC PR: ", round(aucpr.val,3)),
    cex = 1)
}

plot.rocs1(m1,1)

all.variables <- glm(churn ~ .-total_day_minutes-total_eve_charge, family = "binomial", data = over)

m2 <- step(all.variables, direction = "backward", trace = 0)

summary(m2)

vif(m2)

pred.2.raw <- predict(m2, type = "response", newdata = test)
```

```r
pred.2 <- as.factor(ifelse(pred.2.raw < .5, "no", "yes"))

cm <- table(test$churn, pred.2)

cm2 <- confusionMatrix(cm, positive="yes")

cm2

plot.rocs1(m2,2)

all.variables <- glm(churn ~ .-total_day_minutes, family = "binomial", data = under)

m3 <- step(all.variables, direction = "backward", trace = 0)

summary(m3)

vif(m3)

pred.3.raw <- predict(m3, type = "response", newdata = test)
pred.3 <- as.factor(ifelse(pred.3.raw < .5, "no", "yes"))

cm <- table(test$churn, pred.3)

cm3 <- confusionMatrix(cm, positive="yes")

cm3

plot.rocs1(m3,3)

m4 <- naiveBayes(churn ~ international_plan + voice_mail_plan +
    total_day_minutes + total_eve_minutes + total_night_minutes +
    total_intl_minutes + total_intl_calls + number_customer_service_calls, data = train)

y_pred <- predict(m4, type = "class", newdata = test)
cm <- table(test$churn, y_pred)

cm4 <- confusionMatrix(cm, positive="yes")

cm4

plot.rocs <- function(model, model_num){
  par(mfrow=c(1,2))
  prednb <- predict(model, test, type='raw')

pred<-prediction(prednb[,2], test$churn, label.ordering = c("no", "yes"))
perf<-performance(pred,"tpr","fpr")
p1 <- plot(perf, main=paste0("Model ", model_num, ": ROC curve"))
abline(0,1)
auc <- performance(pred,"auc")
auc.val <- auc@y.values[[1]]
text(x = 0.5, y = 0.25, labels = paste0("AUC: ", round(auc.val,3)),
    cex = 1)
```

```
perf1 <- performance(pred, "prec", "rec")
aucpr<- performance(pred,"aucpr")
aucpr.val<- aucpr@y.values[[1]]
p2 <- plot(perf1, main=paste0("Model ", model_num, ": Precision-recall curve"))
abline(1,-1)
text(x = 0.5, y = 0.25, labels = paste0("AUC PR: ", round(aucpr.val,3)),
    cex = 1)

p1
p2
}

plot.rocs(m4,4)

m5 <- naiveBayes(churn ~ state + international_plan + voice_mail_plan +
    number_vmail_messages + total_day_calls + total_day_charge +
    total_eve_minutes + total_eve_calls + total_night_minutes +
    total_night_calls + total_intl_minutes + total_intl_calls +
    number_customer_service_calls, data = over)

y_pred <- predict(m5, newdata = test)
cm <- table(test$churn, y_pred)

cm5 <- confusionMatrix(cm, positive="yes")

cm5

plot.rocs(m5,5)

m6 <- naiveBayes(churn ~ international_plan + voice_mail_plan +
    number_vmail_messages + total_day_charge + total_eve_charge +
    total_intl_charge + number_customer_service_calls, data = under)

y_pred <- predict(m6, newdata = test)
cm <- table(test$churn, y_pred)

cm6 <- confusionMatrix(cm, positive="yes")

cm6

plot.rocs(m6,6)

get_metrics <- function(confusion.matrix){
by.class <- round(confusion.matrix$byClass, 3)
metrics <- data.frame(t(by.class))

Accuracy <- round(confusion.matrix$overall[1], 3)

classification.metrics <- cbind(Accuracy, metrics)

return(classification.metrics)
}
```

```r
metric1 <- get_metrics(cm1)
metric2 <- get_metrics(cm2)
metric3 <- get_metrics(cm3)
metric4 <- get_metrics(cm4)
metric5 <- get_metrics(cm5)
metric6 <- get_metrics(cm6)

model.metrics <- rbind(metric1, metric2, metric3, metric4, metric5, metric6)
rownames(model.metrics) <- c("Model 1", "Model 2", "Model 3", "Model 4", "Model 5", "Model 6")

AUC <- c(0.812, 0.816, 0.814, 0.862, 0.799, 0.803)
AUC_PR <- c(0.476, 0.461, 0.45, 0.538, 0.505, 0.499)

model.metrics <- t(model.metrics)

model.metrics <- rbind(model.metrics, AUC, AUC_PR)


kableExtra::kable(model.metrics)
```