Analytics Vidhya
Learn everything about analytics

BUSINESS ANALYTICS DATA SCIENCE R

A Practical Introduction to Prescriptive Analytics (with Case Study in R)

PRANOV MISHRA, MAY 10, 2019 LOGIN TO BOOKMARK THIS ARTICLE

This article was submitted as part of Analytics Vidhya's Internship Challenge.

Introduction

"What are the different branches of analytics?" Most of us, when we're starting out on our analytics journey, are taught that there are two types – descriptive analytics and predictive analytics. There's actually a third branch which is often overlooked – prescriptive analytics.

Prescriptive analytics is the most powerful branch among the three. Let me show you how with an example.

Recently, a deadly cyclone hit Odisha, India, but thankfully most people had already been evacuated. The Odisha meteorological department had already predicted the arrival of the monstrous cyclone and made the life-saving decision to evacuate the potentially prone regions.

Contrast that with 1999, when more than 10,000 people died because of a similar cyclone. They were caught unaware since there was no prediction about the coming storm. So what changed?

The government of Odisha was a beneficiary of prescriptive analytics. They were able to utilize the services of the meteorological department's accurate prediction of cyclones – their path, strength, and timing. They used this to make decisions about when and what needs to be done to prevent any loss of life.

So in this article, we will first understand what the term prescriptive analytics means. We will then solidify our learning by taking up a case study and implementing the branches of analytics -descriptive, predictive and prescriptive. Let's go!

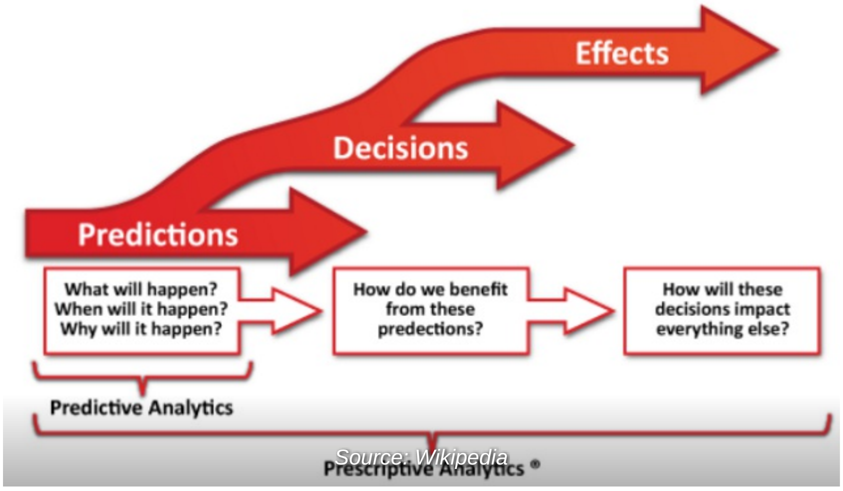
Table of Contents

What is Prescriptive Analytics?

We can broadly classify analytics into three distinct segments – Descriptive, Predictive and Prescriptive Analytics. Let's take a look at each of these:

- **Descriptive Analytics** is the first part of any model building exercise. We analyze the historical data to identify patterns and trends of the dependent and independent variables. This stage also helps in hypothesis generation, variable transformation and any root cause analysis of specific behavioral patterns
- **Predictive Analytics** is the next stage of analytics. Here, we leverage the cleaned and/or transformed data and fit a model on that data to predict the future behavior of the dependent variable. Predictive analytics answers the question of what is likely to happen
- **Prescriptive Analytics** is the last stage where the predictions are used to prescribe (or recommend) the next set of things to be done. That's where our Odisha Government example came from. They leveraged the predictions made by the meteorological department and took a series of measures, like relocating all people from low lying areas, arranging for food, shelter and medical help in advance, etc., to ensure the damage is limited

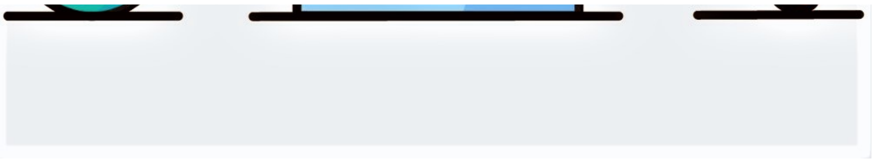
The below image does a nice job of illustrating the components under the prescriptive analytics umbrella:



Source: Wikipedia

Setting up our Problem Statement

I've found the best way of learning a topic is by practicing it. So, let's understand prescriptive analytics by taking up a case study and implementing each analytics segment we discussed above.



The senior management in a telecom provider organization is worried about the rising customer attrition levels. Additionally, a recent independent survey has suggested that the industry as a whole will face increasing churn rates and decreasing ARPU (average revenue per unit).

The effort to retain customers so far has been very reactive. Only when the customer calls to close their account is when we take action. That's not a great strategy, is it? The management team is keen to take more proactive measures on this front.

We as data scientists are tasked with analyzing their data, deriving insights, predicting the potential behavior of customers, and then recommending steps to improve performance.

Getting the Dataset for our Problem

You can download the dataset from **here**. I have also provided the full code on myGithub repository. There are three R files and you should use them in the below order:

- DataPreparation.r
- Visualization.r
- ModelBuilding.r

Hypothesis Generation

Generating a hypothesis is the key to unlocking any data science or analytics project. We should first list down what it is we are trying to achieve through our approach and then proceed from there.

Customer churn is being driven by the below factors (according the the independent industry survey):

- Cost and billing
- Network and service quality
- Data usage connectivity issues

We would like to test the same for our telecom provider. Typically, we encourage the company to come up with an exhaustive set of hypotheses so as not to leave out any variables or major points. However, we'll narrow our focus down to one for the scope of this article:

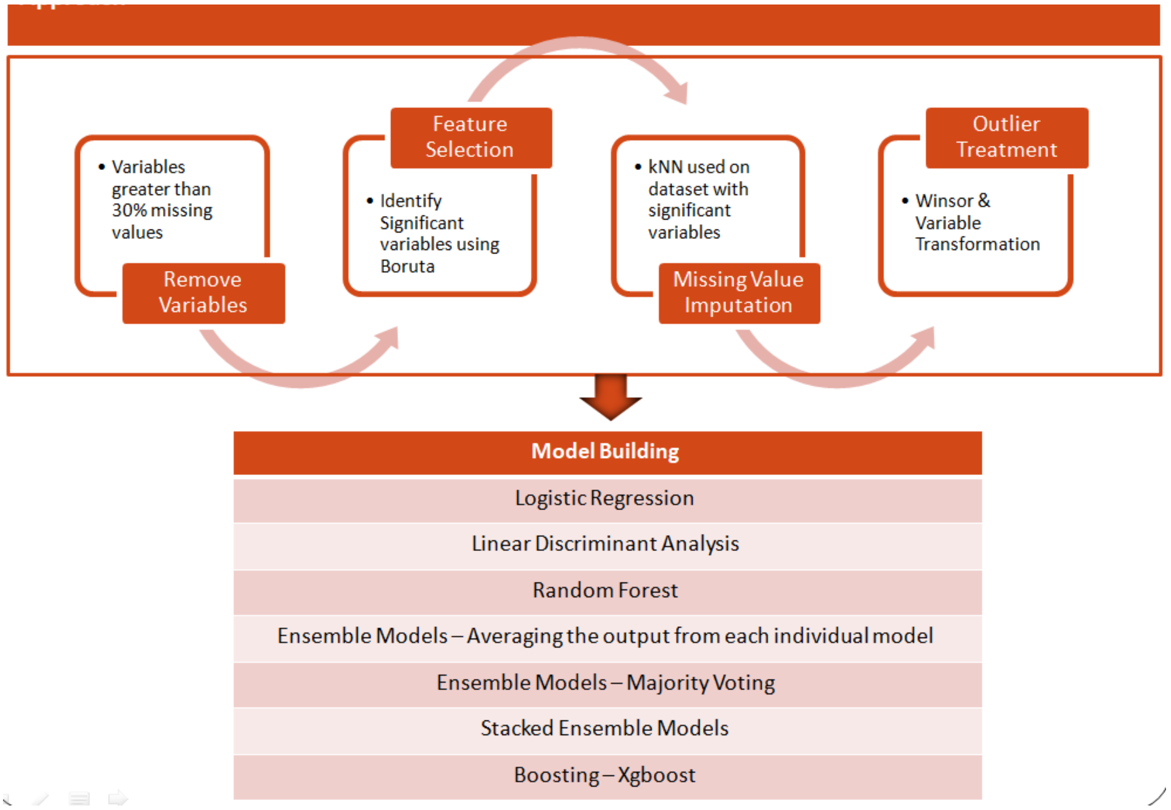
> *Are the variables related to cost, billing, network, and service quality making a significant contribution towards a customer's decision to stay with or leave the service provider?*

Laying Down our Model Building Approach

Now that we have the data set, the problem statement and the hypothesis to test, it's time to get our hands dirty. Let's tear into the data and see what insights can be drawn.

I have summarized my approach in the below illustration. Typically, any model building exercise will go through similar steps. *Note that this is my approach – you can change things up and play around with the data on your end. For instance, we are removing variables with more than 30% missing values but you can take your own call on this.*

Approach



- Variables greater than 30% missing values

Remove Variables

Feature Selection

- Identify Significant variables using Boruta

- kNN used on dataset with significant variables

Missing Value Imputation

Outlier Treatment

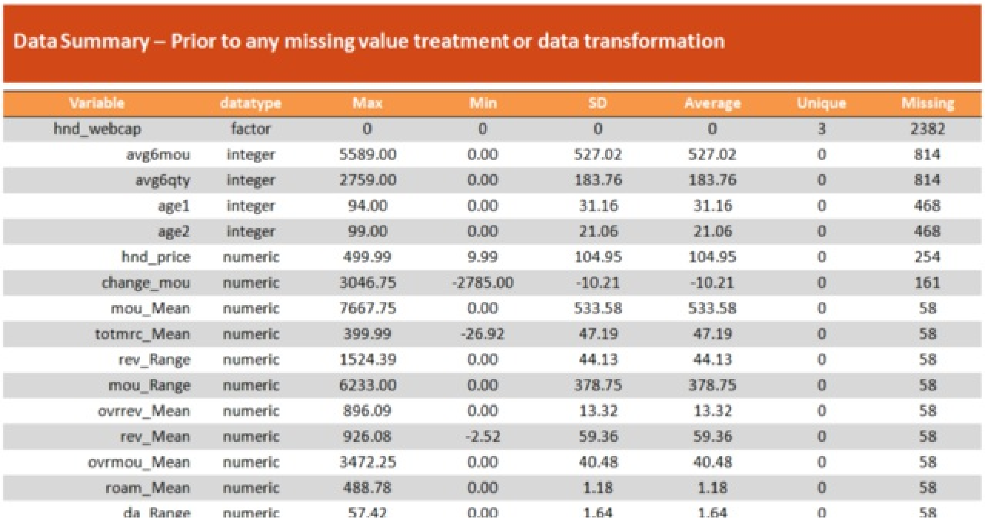
- Winsor & Variable Transformation

Model Building
Logistic Regression
Linear Discriminant Analysis
Random Forest
Ensemble Models – Averaging the output from each individual model
Ensemble Models – Majority Voting
Stacked Ensemble Models
Boosting – Xgboost

Here’s the code to find the variables with more than 30% missing values:

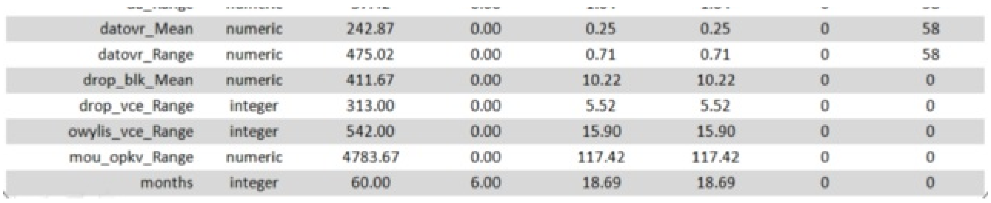
```
mydata=read.csv("Telecom_Sampled.csv")
mydata$churn=as.factor(mydata$churn)
anyNA(mydata)
Percentage_missing=round(colSums(is.na(mydata[,colnames(mydata)[colSums(is.na(mydata))>0]]))/nrow(mydata)*100,2)
data.frame(MissingProportion=sort(Percentage_missing, decreasing = TRUE))

#Finding variable names with more than 30% missing values
Variables_with_High_NAs=colnames(mydata)[colSums(is.na(mydata))/nrow(mydata)>0.3]

#Removing the variables with more than 30% missing values from the original dataset
mydata=mydata[,!names(mydata)%in%Variables_with_High_NAs]

#13 variables removed
```

As you can see in the above illustration, we removed all variables with more than 30% missing values. Here’s the summary of our dataset:

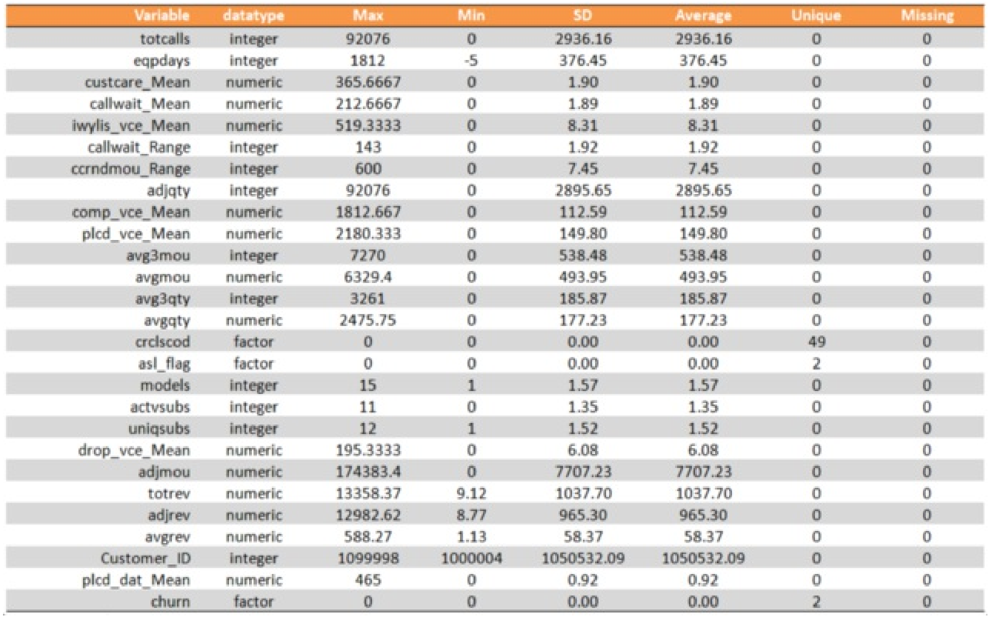


Data Summary – Prior to any missing value treatment or data transformation

Variable	datatype	Max	Min	SD	Average	Unique	Missing
hnd_webcap	factor	0	0	0	0	3	2382
avg6mou	integer	5589.00	0.00	527.02	527.02	0	814
avg6qty	integer	2759.00	0.00	183.76	183.76	0	814
age1	integer	94.00	0.00	31.16	31.16	0	468
age2	integer	99.00	0.00	21.06	21.06	0	468
hnd_price	numeric	499.99	9.99	104.95	104.95	0	254
change_mou	numeric	3046.75	-2785.00	-10.21	-10.21	0	161
mou_Mean	numeric	7667.75	0.00	533.58	533.58	0	58
totmrc_Mean	numeric	399.99	-26.92	47.19	47.19	0	58
rev_Range	numeric	1524.39	0.00	44.13	44.13	0	58
mou_Range	numeric	6233.00	0.00	378.75	378.75	0	58
ovrrev_Mean	numeric	896.09	0.00	13.32	13.32	0	58
rev_Mean	numeric	926.08	-2.52	59.36	59.36	0	58
ovrmou_Mean	numeric	3472.25	0.00	40.48	40.48	0	58
roam_Mean	numeric	488.78	0.00	1.18	1.18	0	58
da_Range	numeric	57.42	0.00	1.64	1.64	0	58



[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
datovr_Mean	numeric	242.87	0.00	0.25	0.25	0	58
datovr_Range	numeric	475.02	0.00	0.71	0.71	0	58
drop_blk_Mean	numeric	411.67	0.00	10.22	10.22	0	0
drop_vce_Range	integer	313.00	0.00	5.52	5.52	0	0
owylis_vce_Range	integer	542.00	0.00	15.90	15.90	0	0
mou_opkv_Range	numeric	4783.67	0.00	117.42	117.42	0	0
months	integer	60.00	6.00	18.69	18.69	0	0



Variable	datatype	Max	Min	SD	Average	Unique	Missing
totcalls	integer	92076	0	2936.16	2936.16	0	0
eqpdays	integer	1812	-5	376.45	376.45	0	0
custcare_Mean	numeric	365.6667	0	1.90	1.90	0	0
callwait_Mean	numeric	212.6667	0	1.89	1.89	0	0
iwylis_vce_Mean	numeric	519.3333	0	8.31	8.31	0	0
callwait_Range	integer	143	0	1.92	1.92	0	0
ccrndmou_Range	integer	600	0	7.45	7.45	0	0
adjqty	integer	92076	0	2895.65	2895.65	0	0
comp_vce_Mean	numeric	1812.667	0	112.59	112.59	0	0
plcd_vce_Mean	numeric	2180.333	0	149.80	149.80	0	0
avg3mou	integer	7270	0	538.48	538.48	0	0
avgmou	numeric	6329.4	0	493.95	493.95	0	0
avg3qty	integer	3261	0	185.87	185.87	0	0
avgqty	numeric	2475.75	0	177.23	177.23	0	0
crclscod	factor	0	0	0.00	0.00	49	0
asl_flag	factor	0	0	0.00	0.00	2	0
models	integer	15	1	1.57	1.57	0	0
actvsubs	integer	11	0	1.35	1.35	0	0
uniqsubs	integer	12	1	1.52	1.52	0	0
drop_vce_Mean	numeric	195.3333	0	6.08	6.08	0	0
adjmou	numeric	174383.4	0	7707.23	7707.23	0	0
totrev	numeric	13358.37	9.12	1037.70	1037.70	0	0
adjrev	numeric	12982.62	8.77	965.30	965.30	0	0
avgrev	numeric	588.27	1.13	58.37	58.37	0	0
Customer_ID	integer	1099998	1000004	1050532.09	1050532.09	0	0
plcd_dat_Mean	numeric	465	0	0.92	0.92	0	0
churn	factor	0	0	0.00	0.00	2	0

We have reduced the number of variables from 82 to 69.

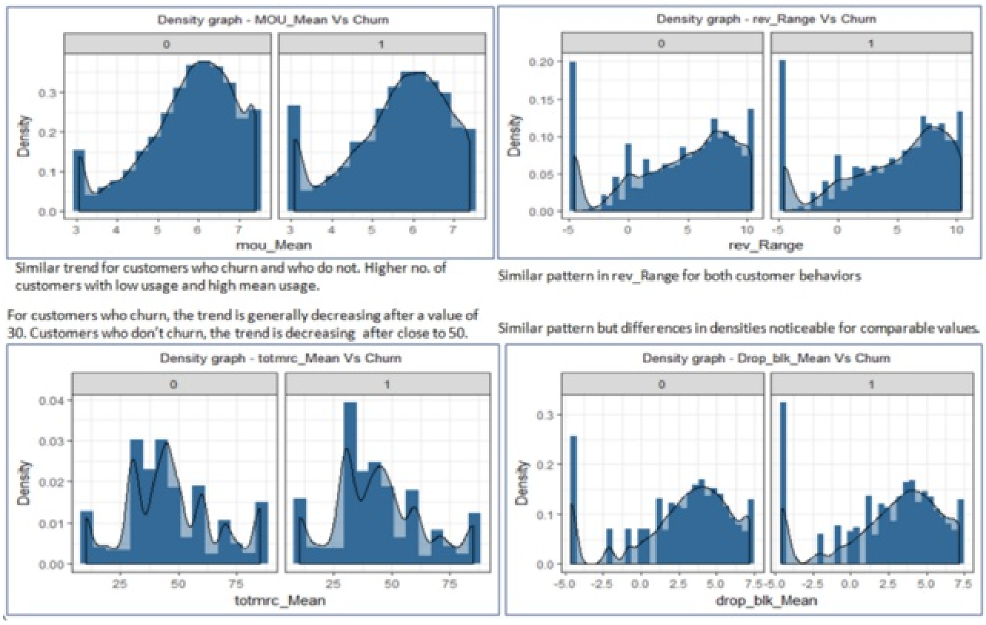
Data Visualization and Data Preparation — Descriptive Analytics

Let’s do a univariate, bivariate and multivariate analysis of various independent variables along with the target variable. This should give us an idea of the effects of churn. I have shared a few visualizations below. You can find the entire exploratory analysis on the GitHub repository.

Let’s start by drawing up three plots (output is below the code block):

```
#Univariate Analysis & Multivariate Analysis
a=ggplot(Telecom_Winsor, aes(x=mou_Mean, y=..density.., fill=1))
a+geom_histogram(stat = "bin", bins = 15)+geom_density(alpha=0.5)+
guides(fill=FALSE)+labs(y="Density", title="Density graph - MOU_Mean Vs Churn")+
theme_bw()+facet_grid(~churn)+theme(plot.title = element_text(size = 10, hjust = 0.5))

a=ggplot(Telecom_Winsor, aes(x=totmrc_Mean, y=..density.., fill=1))
a+geom_histogram(stat = "bin", bins = 15)+geom_density(alpha=0.5)+
guides(fill=FALSE)+labs(y="Density", title="Density graph - totmrc_Mean Vs Churn")+
theme_bw()+facet_grid(~churn)+theme(plot.title = element_text(size = 10, hjust = 0.5))

a=ggplot(Telecom_Winsor,aes(x=F_eqpdays), alpha=0.5)
a+geom_bar(stat = "count", aes(fill=models), position = "dodge")+
facet_grid(~churn)+labs(x="", fill="Models",y="Count", title="F_eqpdays Impact Churn?")+
theme(legend.position = c(0.8,0.8), plot.title = element_text(size = 10, hjust = 0.5),
legend.key.size = unit(0.5,"cm"), legend.title = element_text(size = 8),
legend.text = element_text(size = 8), axis.text.x = element_text( angle=45,size = 8, vjust = 1,
hjust = 1),
legend.direction = "horizontal")
```

First, we will analyze the mean minutes of usage, revenue range, mean total monthly recurring charge and the mean number of dropped or blocked calls against the target variable – churn:



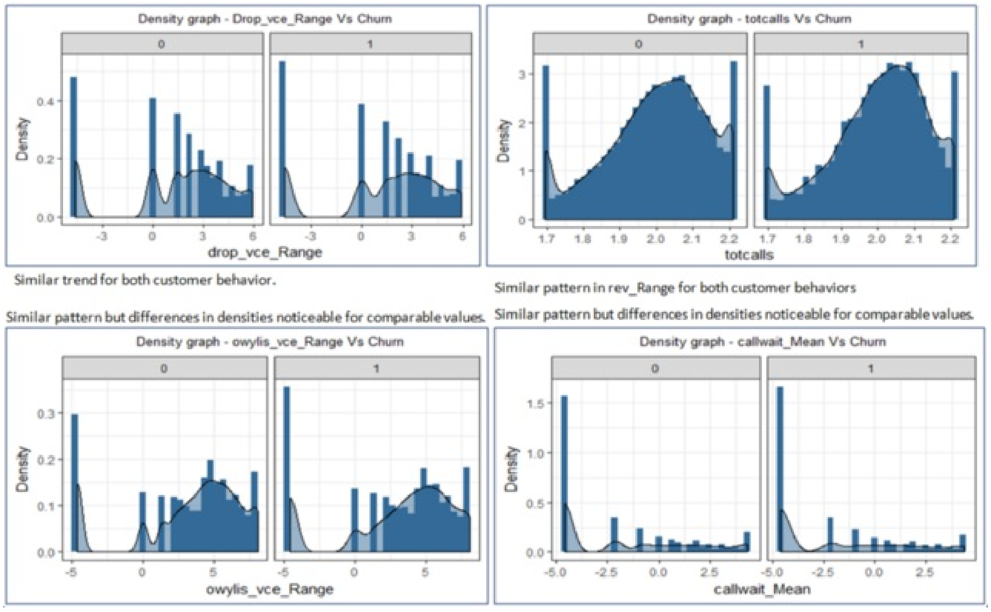
Similar trend for customers who churn and who do not. Higher no. of customers with low usage and high mean usage.

Similar pattern in rev_Range for both customer behaviors

For customers who churn, the trend is generally decreasing after a value of 30. Customers who don't churn, the trend is decreasing after close to 50.

Similar pattern but differences in densities noticeable for comparable values.

Similarly, we shall analyze the mean number of dropped (failed) voice calls, the total number of calls over the life of the customer, the range of the number of outbound wireless to wireless voice calls and the mean number of call waiting against the churn variable:



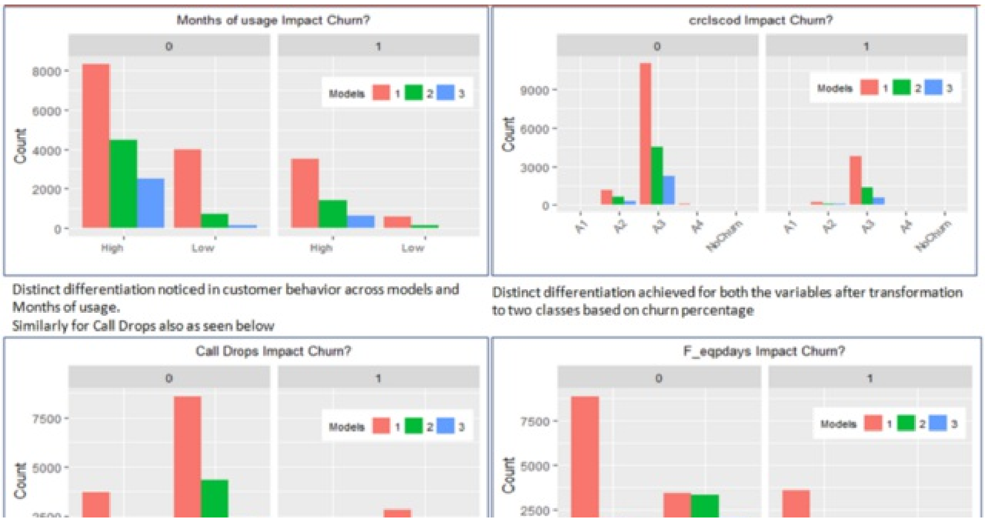
Similar trend for both customer behavior.

Similar pattern in rev_Range for both customer behaviors

Similar pattern but differences in densities noticeable for comparable values.

Similar pattern but differences in densities noticeable for comparable values.

Let's change things up a bit. We'll use the faceting functionality in the awesome ggplot2 package to plot the months of usage, credit class code, call drops and the number of days of current equipment against the churn variable:



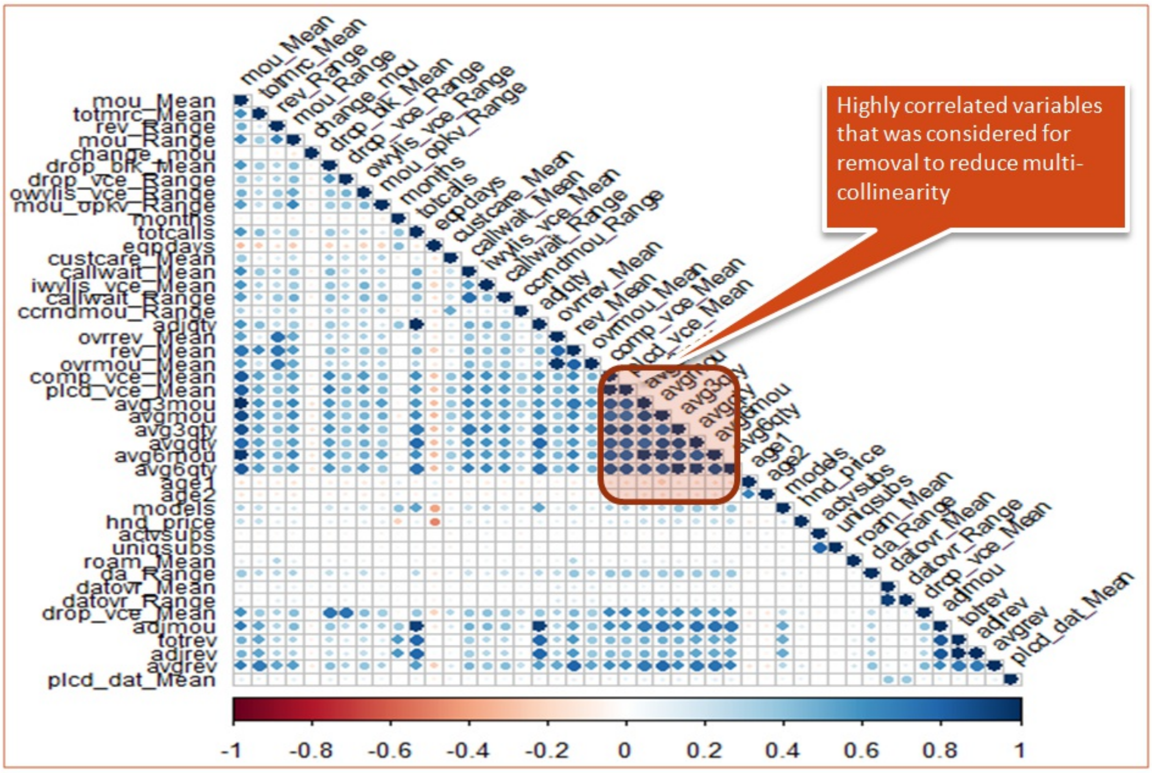
Distinct differentiation noticed in customer behavior across models and Months of usage.
Similarly for Call Drops also as seen below

Distinct differentiation achieved for both the variables after transformation to two classes based on churn percentage

We will analyze the numeric variable separately to see if there are any features that have high degrees of collinearity. This is because the presence of collinear variables always reduces the model's performance since they introduce bias into the model.

We should handle the collinearity problem. Now, there are many ways of dealing with it, such as variable transformation and reduction using principal component analysis (PCA). I have removed the highly correlated variables:

```
###Finding highly correlated variables

Numeric=Filter(is.numeric,mydata_Rev)

library(corrplot)

M=cor(na.omit(Numeric))

corrplot(M, method = "circle", type = "lower",

tl.srt = 45, tl.col = "black", tl.cex = 0.75)
```



Highly correlated variables that was considered for removal to reduce multi-collinearity

Prediction of Customer Behavior—Predictive Analytics

This is the part most of you will be familiar with – building models on the training data. We'll build a number of models so we can compare their performance across the spectrum.

It is generally a good practice to train multiple models starting from simple linear models to complex non-parametric and non-linear ones. The performance of models varies depending on how the dependent and independent variables are related. If the relationship is linear, the simpler models give good results (plus they're easier to interpret).

Alternatively, if the relationship is non-linear, complex models generally give better results. As the complexity of the model increases, the bias introduced by the model reduces and the variance increases. For our problem, we will build around ten models on the training set and validate them on unseen test data.

The models we'll build are:

- Simple Models like **Logistic Regression & Discriminant Analysis** with different thresholds for classification
- **Random Forest** after balancing the dataset using Synthetic Minority Oversampling Technique **(SMOTE)**
- **The ensemble** of five individual models and predicting the output by averaging the individual output probabilities

- **The ensemble** of five individual models and predicting the output by averaging the individual output probabilities
- **XGBoost** algorithm

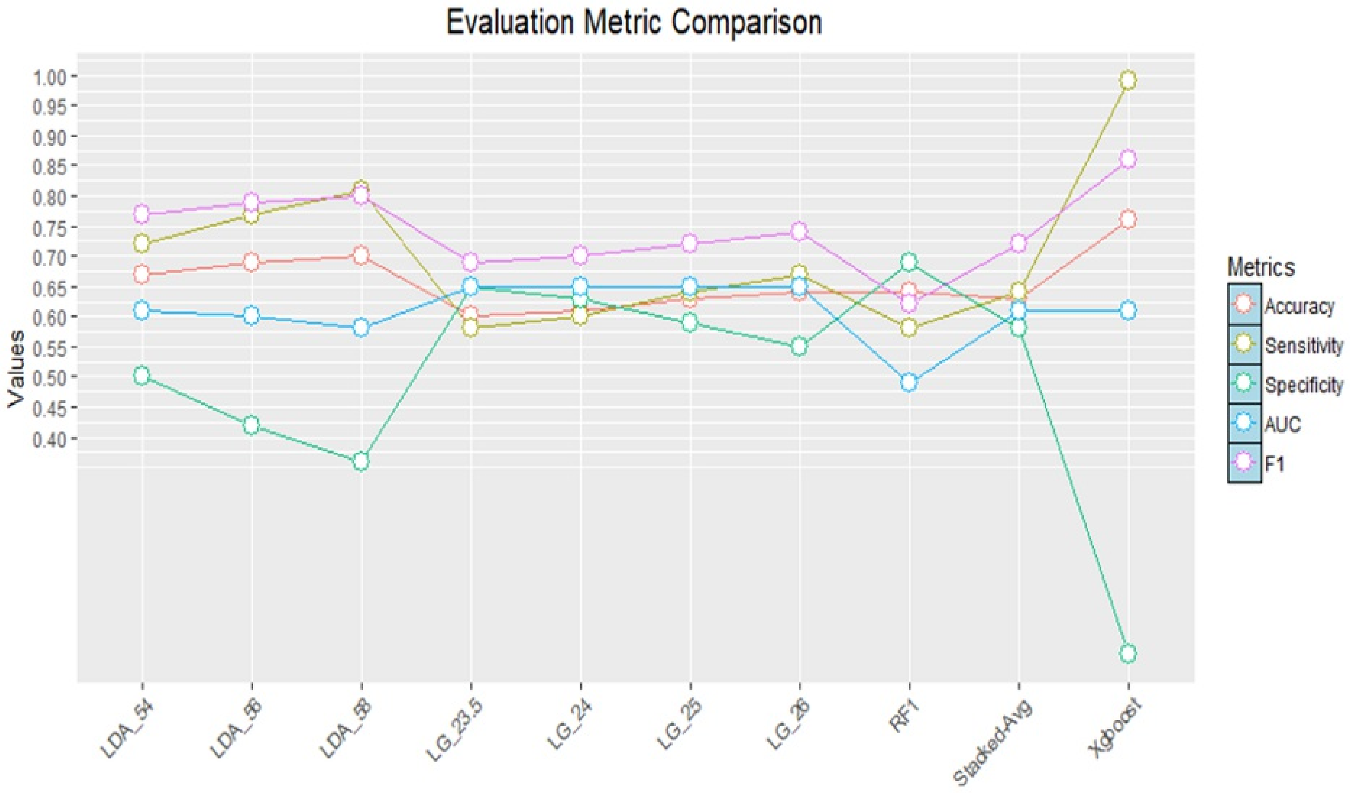
Here’s the code to the logistic regression model (you can try out the rest using the code provided in my GitHub repository):

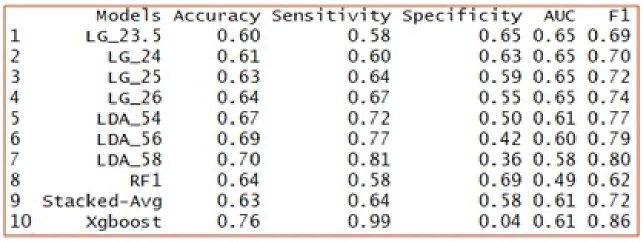
```
LGM1=glm(churn~., data = Telecom_Winsor, family = "binomial")
summary(LGM1)
#Remove hnd_wecap since it did not seem to be a significant and is introducing NAs in the model
names(Telecom_Winsor)
Telecom_Winsor_Lg=subset(Telecom_Winsor,select = -hnd_webcap)
Telecom_Winsor_Lg=droplevels(Telecom_Winsor_Lg)

#Data Splitting
library(caret)
set.seed(1234)

Index=createDataPartition(Telecom_Winsor_Lg$churn,times = 1,p=0.75,list = FALSE)
Train=Telecom_Winsor_Lg[Index,]
Test=Telecom_Winsor_Lg[-Index,]

prop.table(table(Train$churn))

LGM1=glm(churn~., data = Train, family = "binomial")
summary(LGM1)
step(LGM1, direction = "both")

library(car)

LGMF=glm(formula = churn ~ mou_Mean + totmrc_Mean + rev_Range + drop_blk_Mean +
      drop_vce_Range + callwait_Mean + callwait_Range + ccrndmou_Range +
      adjqty + rev_Mean + ovrmou_Mean + avgqty + age1 + age2 +
      hnd_price + actvsubs + uniqsubs + datovr_Range + adjmou +
      adjrev + plcd_dat_Mean + crclscod + asl_flag + mouR_Factor +
      change_mF + F_months + F_eqpdays + F_iwylis_Vmean, family = "binomial",
     data = Train)
car::vif(LGMF)
#adjmou, avgqty and adjqty have very high VIF but with Df considered GVIF is low enough. Hence no requirement for further collinesrity treatment

summary(LGMF)

Pred=predict(LGMF, Test, type = "response")
options(scipen = 9999)
L=data.frame(LogOfOdds=round(exp(coef(LGMF)),3))
L$Variable=row.names(L)
row.names(L)=NULL
L=L[,c(2,1)]

#The variables which if undergo a change of 1 unit there is more than 50% probability of the customer decision changing from terminating service to stayi
ng with the servicer

L%>%arrange(desc(LogOfOdds))%>%filter(LogOfOdds>=1)%>%mutate(Probability=round(LogOfOdds/(1+LogOfOdds),3))

Pred.class=ifelse(Pred>0.24,1,0)
CM=confusionMatrix(as.factor(Pred.class),Test$churn)
```

```
CM=confusionMatrix(as.factor(Pred.class),Test$churn)
CM$table
fourfoldplot(CM$table)
Acc_Log24=CM$overall[[1]]
Sensitivity_Log24=CM$byClass[[1]]
Specificity_Log24=CM$byClass[[2]]
F1sq_Log24=CM$byClass[[7]]
library(ROCR)
Pred.Storage=prediction(Pred,Test$churn)

AUC=performance(Pred.Storage,"auc")
AUC_Log24=AUC@y.values[[1]]
perf=performance(Pred.Storage,"tpr","fpr")

###########################

cut_offs=data.frame(cut=perf@alpha.values[[1]], fpr=perf@x.values[[1]], tpr=perf@y.values[[1]])
cut_offs=cut_offs[order(cut_offs$tpr, decreasing = TRUE),]
library(dplyr)
cut_offs%>%filter(fpr<=0.42,tpr>0.59)

#cutoff of 0.2356 seems to give the highest tpr and relatively low fpr

Pred.class235=ifelse(Pred>0.235,1,0)
CM=confusionMatrix(as.factor(Pred.class235),Test$churn)
fourfoldplot(CM$table)
Acc_Log23.5=CM$overall[[1]]
Sensitivity_Log23.5=CM$byClass[[1]]
Specificity_Log23.5=CM$byClass[[2]]

AUC=performance(Pred.Storage,"auc")
AUC_Log23.5=AUC@y.values[[1]]
F1sq_Log23.5=CM$byClass[[7]]
############
Pred.class=ifelse(Pred>0.25,1,0)
CM=confusionMatrix(as.factor(Pred.class),Test$churn)
fourfoldplot(CM$table)
Acc_Log25=CM$overall[[1]]
Sensitivity_Log25=CM$byClass[[1]]
Specificity_Log25=CM$byClass[[2]]
F1sq_Log25=CM$byClass[[7]]
AUC=performance(Pred.Storage,"auc")
AUC_Log25=AUC@y.values[[1]]
################################
Pred.class=ifelse(Pred>0.26,1,0)
CM=confusionMatrix(as.factor(Pred.class),Test$churn)
fourfoldplot(CM$table)
Acc_Log26=CM$overall[[1]]
Sensitivity_Log26=CM$byClass[[1]]
Specificity_Log26=CM$byClass[[2]]
F1sq_Log26=CM$byClass[[7]]
#Choice of cutoff at 24,25, 26 results in increasing accuracy and sensitivity but decreasing Specificity
AUC=performance(Pred.Storage,"auc")
```

```
AUC_Log26=AUC@y.values[[1]]
```

Below is a comparison of the evaluation of our models:





```
        Models Accuracy Sensitivity Specificity  AUC   F1
1      LG_23.5     0.60        0.58        0.65 0.65 0.69
2        LG_24     0.61        0.60        0.63 0.65 0.70
3        LG_25     0.63        0.64        0.59 0.65 0.72
4        LG_26     0.64        0.67        0.55 0.65 0.74
5       LDA_54     0.67        0.72        0.50 0.61 0.77
6       LDA_56     0.69        0.77        0.42 0.60 0.79
7       LDA_58     0.70        0.81        0.36 0.58 0.80
8          RF1     0.64        0.58        0.69 0.49 0.62
9  Stacked-Avg     0.63        0.64        0.58 0.61 0.72
10     Xgboost     0.76        0.99        0.04 0.61 0.86
```

Logistic regression seems to give the best result when compared with the other models. *LG_26* is a logistic regression model with a threshold of 26%. Let me know if you improved on this score – I would love to hear your thoughts on how you approached this problem.

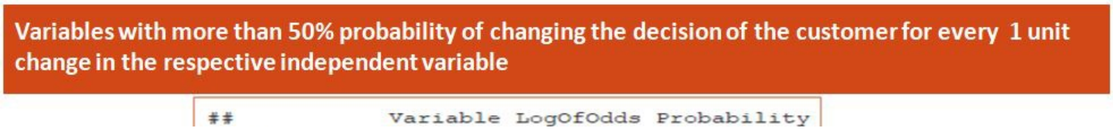
Recommendations to improve performance—Prescriptive Analytics

And now comes the part we've been waiting for – prescriptive analytics! Let's see what recommendations we can come up with to improve the performance of our model.

In the image below, we've listed the variables that have more than 50% probability of changing the decision of the customer for every 1 unit change in the respective independent variable. This insight was generated from the logistic regression model we saw above. That is essentially a relationship between the log of odds of the dependent variable with the independent variables.

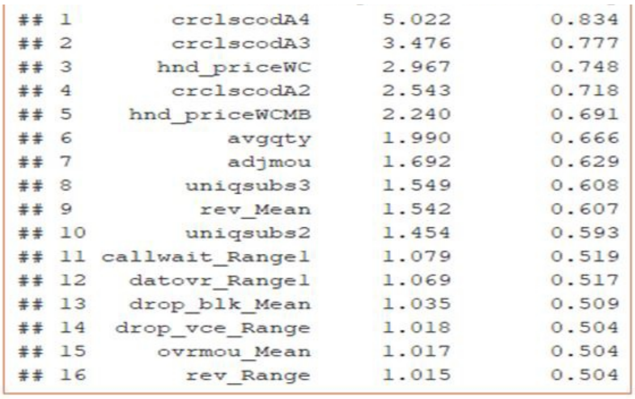
So, if we calculate the exponential of coefficients of the dependent variable, we get the odds and from that, we get the probability*(using formula Probability = Odds/(1+Odds))* of customer behavior changing for one unit change in the independent variable.

The below image will give you a better idea of what I'm talking about:

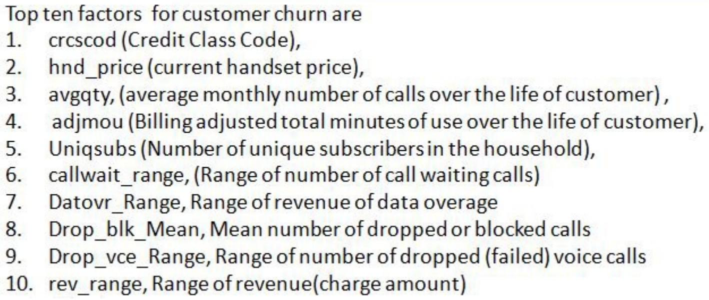


Variables with more than 50% probability of changing the decision of the customer for every 1 unit change in the respective independent variable

```
##                Variable  LogOfOdds  Probability
```



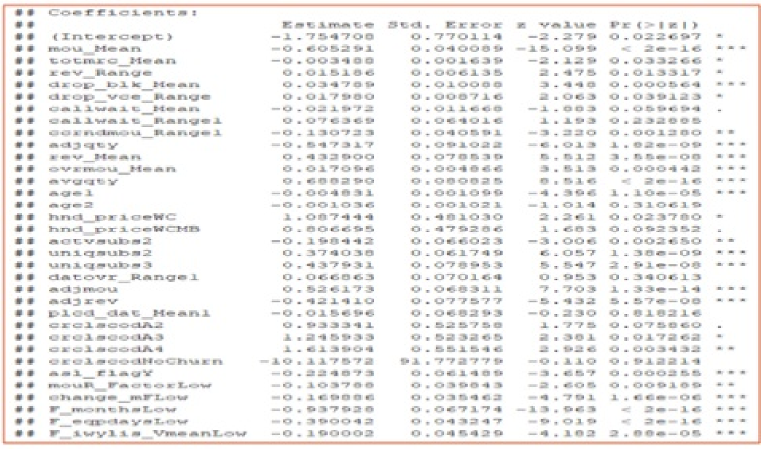
```
## 1         crclscodA4      5.022         0.834
## 2         crclscodA3      3.476         0.777
## 3        hnd_priceWC      2.967         0.748
## 4         crclscodA2      2.543         0.718
## 5      hnd_priceWCMB      2.240         0.691
## 6             avgqty      1.990         0.666
## 7            adjmou      1.692         0.629
## 8         uniqsubs3      1.549         0.608
## 9          rev_Mean      1.542         0.607
## 10        uniqsubs2      1.454         0.593
## 11 callwait_Range1      1.079         0.519
## 12    datovr_Range1      1.069         0.517
## 13    drop_blk_Mean      1.035         0.509
## 14   drop_vce_Range      1.018         0.504
## 15      ovrmou_Mean      1.017         0.504
## 16        rev_Range      1.015         0.504
```



Top ten factors for customer churn are

1. crcscod (Credit Class Code),
2. hnd_price (current handset price),
3. avgqty, (average monthly number of calls over the life of customer) ,
4. adjmou (Billing adjusted total minutes of use over the life of customer),
5. Uniqsubs (Number of unique subscribers in the household),
6. callwait_range, (Range of number of call waiting calls)
7. Datovr_Range, Range of revenue of data overage
8. Drop_blk_Mean, Mean number of dropped or blocked calls
9. Drop_vce_Range, Range of number of dropped (failed) voice calls
10. rev_range, Range of revenue(charge amount)

Remember the hypothesis we generated using the independent survey earlier? This has also come out to be true. The below summary statistics from the logistic model proves that:



```
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
##  (Intercept)          -1.754708   0.770114  -2.279 0.022697 *
## mou_Mean              -0.605291   0.040089 -15.099  < 2e-16 ***
## totmrc_Mean           -0.003488   0.001639  -2.129 0.033266 *
## rev_Range              0.015186   0.006135   2.475 0.013317 *
## drop_blk_Mean          0.034789   0.010088   3.448 0.000564 ***
## drop_vce_Range         0.017980   0.008716   2.063 0.039123 *
## callwait_Mean         -0.021972   0.011668  -1.883 0.059694 .
## callwait_Range1        0.076369   0.064016   1.193 0.232885
## ccrndmou_Range1       -0.130723   0.040591  -3.220 0.001280 **
## adjqty                -0.547317   0.091022  -6.013 1.82e-09 ***
## rev_Mean               0.432900   0.078539   5.512 3.55e-08 ***
## ovrmou_Mean            0.017096   0.004866   3.513 0.000442 ***
## avgqty                 0.688290   0.080825   8.516  < 2e-16 ***
## age1                  -0.004831   0.001099  -4.396 1.10e-05 ***
## age2                  -0.001036   0.001021  -1.014 0.310619
## hnd_priceWC            1.087444   0.481030   2.261 0.023780 *
## hnd_priceWCMB          0.806695   0.479286   1.683 0.092352 .
## actvsubs2             -0.198442   0.066023  -3.006 0.002650 **
## uniqsubs2              0.374038   0.061749   6.057 1.38e-09 ***
## uniqsubs3              0.437931   0.078953   5.547 2.91e-08 ***
## datovr_Range1          0.066863   0.070164   0.953 0.340613
## adjmou                 0.526173   0.068311   7.703 1.33e-14 ***
## adjrev                -0.421410   0.077577  -5.432 5.57e-08 ***
## plcd_dat_Mean1        -0.015696   0.068293  -0.230 0.818216
## crclscodA2             0.933341   0.525758   1.775 0.075860 .
## crclscodA3             1.245933   0.523265   2.381 0.017262 *
## crclscodA4             1.613904   0.551546   2.926 0.003432 **
## crclscodNoChurn      -10.117572  91.772779  -0.110 0.912214
## asl_flagY             -0.224873   0.061489  -3.657 0.000255 ***
## mouR_FactorLow        -0.103788   0.039843  -2.605 0.009189 **
## change_mFLow          -0.169886   0.035462  -4.791 1.66e-06 ***
## F_monthsLow           -0.937928   0.067174 -13.963  < 2e-16 ***
## F_eqpdaysLow          -0.390042   0.043247  -9.019  < 2e-16 ***
## F_iwylis_VmeanLow     -0.190002   0.045429  -4.182 2.88e-05 ***
```

Here’s a quick summary of what we can conclude from our analysis:

- Variables impacting cost and billing are highly significant
- *Adjmou* has one of the top 5 odds ratios
- The mean total monthly recurring charge (*totmrc_Mean*), Revenue (charge amount), Range (*rev_Range*), *adjmou* (Billing adjustments), etc. are found to be highly significant. This suggests that cost and billing impact customer behavior
- Similarly, network and service quality variables like *drop_bkl_Mean* (mean no. of dropped and blocked calls) is highly significant. *Datovr_Range* (Range of revenue of data overage) is not found to be significant but has an odds ratio of more than 1 indicating that 1 unit change in its value has more than a 50% chance of changing the customer behavior from one level to other. Perhaps we need to pay attention to it
- Additionally, the intercept is significant. This constitutes the effects of levels of categorical variables that were removed by the model

Recommendations

Let’s pen down our recommendations based on what we’ve understood.

Recommend rate plan migration as a proactive retention strategy

Mou_Mean (minutes of usage) is one of the most highly significant variables. Hence, it makes sense to **work towards proactively**

Mou_Mean (minutes of usage) is one of the most highly significant variables. Hence, it makes sense to **work towards proactively working with customers to increase their MOU so that they are retained for a longer period.**

Additionally, *mouR_Factor* is highly significant. This, remember, is a derived variable of *mou_Range*.

Changes in MOU are also highly significant. *Change_mF* is a derived variable of *change_mou*.

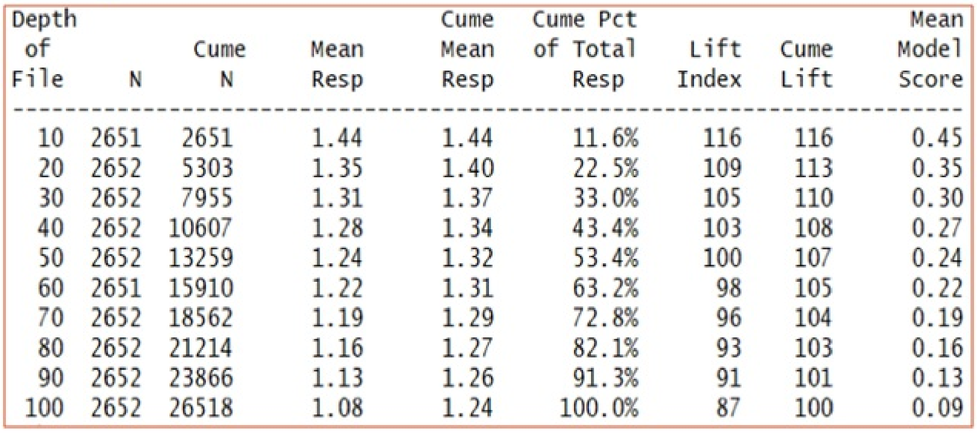
To complement the above, we also see that *ovrmou_Mean* is also a highly significant variable with an odds ratio of more than 1. The variable has a positive estimate of the coefficient indicating an increase in overage churn.

It would help if our company is able to work with the customers. Based on their usage, **we can migrate them to optimal plan rates to avoid overage charges.**

Proactive retention strategy for customers

Identify customers who have the highest probability of churn and develop a proactive retention strategy for them. What if the budget is limited? Then the company can build a lift chart and optimize its retention efforts by reaching out to targeted customers:

Here, with 30% of the total customer pool, the model accurately provides 33% of total potential churn candidates:



```
Depth                              Cume   Cume Pct                     Mean
 of              Cume      Mean    Mean   of Total    Lift    Cume    Model
File      N         N      Resp    Resp       Resp    Index   Lift    Score
---------------------------------------------------------------------------
  10   2651      2651      1.44    1.44      11.6%     116     116     0.45
  20   2652      5303      1.35    1.40      22.5%     109     113     0.35
  30   2652      7955      1.31    1.37      33.0%     105     110     0.30
  40   2652     10607      1.28    1.34      43.4%     103     108     0.27
  50   2652     13259      1.24    1.32      53.4%     100     107     0.24
  60   2651     15910      1.22    1.31      63.2%      98     105     0.22
  70   2652     18562      1.19    1.29      72.8%      96     104     0.19
  80   2652     21214      1.16    1.27      82.1%      93     103     0.16
  90   2652     23866      1.13    1.26      91.3%      91     101     0.13
 100   2652     26518      1.08    1.24     100.0%      87     100     0.09
```

The lift achieved will help us to reach out to churn candidates by targeting much fewer of the total customer pool with the company. Also notice how the first 30 deciles gives us the highest gain. This can give us around 33% of the customers who are likely to terminate the services.

In simple words, the company selects 30% of the entire customer database which covers 33% of the people who are likely to leave. This is much better than randomly calling customers which would have given perhaps a 15% hit rate from all potential churn candidates.

You can use the below code to test the model by identifying 20% of customers who need to be proactively worked with to prevent churn:

```
gains(as.numeric(Telecom_Winsor$churn),predict(LGMF,type="response",newdata=Telecom_Winsor[,-42])
,groups = 10)

Telecom_Winsor$Cust_ID=mydata$Customer_ID

Telecom_Winsor$prob<-predict(LGMF,type="response",newdata=Telecom_Winsor[,-42])

quantile(Telecom_Winsor$prob,prob=c(0.10,0.20,0.30,0.40,0.50,0.60,0.70,0.80,0.90,1))

targeted=Telecom_Winsor%>%filter(prob>0.3224491 & prob<=0.8470540)%>%dplyr::select(Cust_ID)
```

They are the customers whose probability of churn is greater than 32.24% and less than 84.7%. The ModelBuilding.r code will help you with the logical flow of the above code block.

End Notes

Prescriptive analytics is a truly awesome thing if companies are able to utilize it properly. It’s still under the radar as far as the three branches of analytics are concerned.

But as we keep moving up in the hierarchy of analytics, prescriptive analytics is the most favored area as it can help organizations to plan and prepare as they can foresee the future with a fair degree of confidence.

Prescriptive analytics seeks to determine the best solution or outcome among various choices. Just keep in mind that we cannot separate the three branches of analytics. We need to do descriptive and predictive before jumping into prescriptive.

About the Author

Pranov Mishra

Pranov is a Data Science enthusiast with about 11 years of professional experience in the Financial Services industry. Pranov is working as a Vice President in a Multinational Bank and has exposure to Strategic Planning, Intelligent Automation, Data Science, Risk & Controls, Predictive Data Modelling, and People Management. He also mentors analytics (PGPBABI) students enrolled with Great Learning and Great Lakes.

Share this:

TAGS : DATA SCIENCE, PRESCRIPTIVE ANALYTICS, R

NEXT ARTICLE

A Beginner’s Guide to Tidyverse – The Most Powerful Collection of R Packages for Data Science

PREVIOUS ARTICLE

Extracting and Analyzing 1000 Basketball Games using Pandas and Chartify

Pranov Mishra

9 COMMENTS

VINAY GUPTA Reply

May 10, 2019 at 6:11 pm

That’s fantastic!
Loved the simplicity of explanation. Kudos Parnov.

PRANOV SHOBHAN MISHRA Reply
May 11, 2019 at 12:50 am

Thanks, Vinay. Glad that you liked it.

BHARATH Reply
May 11, 2019 at 12:04 pm

Great Post !!!

RITESH Reply
May 11, 2019 at 9:21 pm

Great article. Where can i find the description about the column headers??

RITESH Reply
May 18, 2019 at 9:05 am

where can we find the expanded form of all the variables?

PABITRA Reply
May 12, 2019 at 10:28 pm

Very informative and comprehensive article. Thanks

KARTIK Reply
May 14, 2019 at 9:23 am

Nice one

KRISHNA MOHAN Reply
May 21, 2019 at 1:09 am

Excellent write up Pranov. In addition to making your point about Prescriptive Analytics, you have also outlined how to systematically perform Descriptive and Predictive Analytics as well. A must read for any Analytics student.

PRANOV MISHRA Reply
May 22, 2019 at 10:32 am

Thanks Krishna Mohan. ENcouraging

LEAVE A REPLY

Your email address will not be published.

Comment

Name (required)

Email (required)

Website

SUBMIT COMMENT

☐ Notify me of new posts by email.

JOIN THE NEXTGEN DATA SCIENCE ECOSYSTEM

- Get access to free courses on Analytics Vidhya
- Get free downloadable resource from Analytics Vidhya
- Save your articles
- Participate in hackathons and win prizes

Join Now

HUAWEI WI-FI 6
THE FUTURE-SPEED WI-FI IS HERE
HUAWEI
4 Times the Speed,
Powered By HUAWEI Wireless Technology
Explore more
Huawei Enterprise
Leading Connectivity & Intelligence

POPULAR POSTS

- 24 Ultimate Data Science Projects To Boost Your Knowledge and Skills (& can be accessed freely)
- Essentials of Machine Learning Algorithms (with Python and R Codes)
- 7 Types of Regression Techniques you should know!
- Understanding Support Vector Machine algorithm from examples (along with code)
- A Complete Tutorial to Learn Data Science with Python from Scratch
- Introduction to k-Nearest Neighbors: Simplified (with implementation in Python)

- Introduction to k-Nearest Neighbors: Simplified (with Implementation in Python)
- A Simple Introduction to ANOVA (with applications in Excel)
- Stock Prices Prediction Using Machine Learning and Deep Learning Techniques (with Python codes)

RECENT POSTS

Using the Power of Deep Learning for Cyber Security (Part 2) – Must-Read for All Data Scientists

MAY 23, 2019

Data Science Project: Scraping YouTube Data using Python and Selenium to Classify Videos

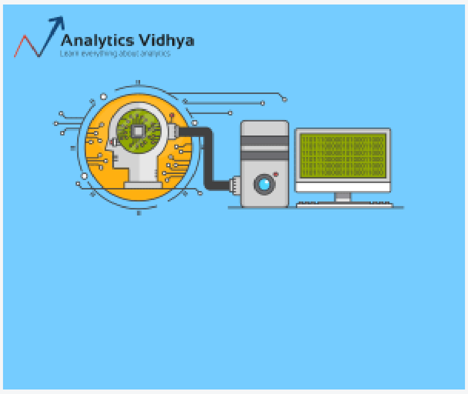
MAY 20, 2019

Statistics for Data Science: Introduction to t-test and its Different Types (with Implementation in R)

MAY 16, 2019

10 Useful Data Analysis Expressions (DAX) Functions for Power BI Beginners

MAY 15, 2019



Analytics Vidhya
Learn everything about analytics

DATA SCIENTISTS

COMPANIES

Don't have an account? Sign up

JOIN OUR COMMUNITY :

f 46336

20707

G+

in 7513