

Telco customer churn

by Ketao Li, Kush Halani, Josue Romain, Juan Peña, Priyanka Patil

Abstract Customer churn is a big challenge for large companies, especially in the highly competitive telecom industry. Due to the effect on the revenues of the companies, they are seeking to find ways to predict potential customer to churn. Therefore, identifying the factors that lead to customer churn is very important to take necessary actions to avoid this churn. Our work is to develop a churn prediction model which helps telecom operators to predict customers who are most likely subject to churn.

Background

In an industry as competitive as Telecom, leading companies know that the key to success is not just about acquiring new customers, but rather, retaining existing ones. But how do you know which customers are at risk and why, and which negative experiences and interactions have the biggest impact on churn across touchpoints and channels over time.

Objective

The objective of this research is to find a supervised, binary classification model that would provide accurate forecast of telco customer churn.

Data Analysis

The data set we are going to use for our research contains customer's attributes. There are over 7044 records. It has been sourced from [Kaggle](#).

Data Dictionary

Column Name	Column Description
customerID	Customer ID
gender	Whether the customer is a male or a female
SeniorCitizen	Whether the customer is a senior citizen or not (1, 0)
Partner	Whether the customer has a partner or not (Yes, No)
Dependents	Whether the customer has dependents or not (Yes, No)
tenure	Number of months the customer has stayed with the company
PhoneService	Whether the customer has a phone service or not (Yes, No)
MultipleLines	Whether the customer has multiple lines or not (Yes, No, No phone service)
InternetService	Customer's internet service provider (DSL, Fiber optic, No)
OnlineSecurity	Whether the customer has online security or not (Yes, No, No internet service)
OnlineBackup	Whether the customer has online backup or not (Yes, No, No internet service)
DeviceProtection	Whether the customer has device protection or not (Yes, No, No internet service)
TechSupport	Whether the customer has tech support or not (Yes, No, No internet service)
StreamingTV	Whether the customer has streaming TV or not (Yes, No, No internet service)
StreamingMovies	Whether the customer has streaming movies or not (Yes, No, No internet service)

Column Name	Column Description
Contract	The contract term of the customer (Month-to-month, One year, Two year)
PaperlessBilling	Whether the customer has paperless billing or not (Yes, No)
PaymentMethod	The customer's payment method (Electronic check, Mailed check, Bank transfer (automatic), Credit card (automatic))
MonthlyCharges	The amount charged to the customer monthly
TotalCharges	The total amount charged to the customer
Churn	Whether the customer churned or not (Yes or No)

Data Exploration

Let's take a close look at the data set.

```
customerData = read.csv("../data/WA_Fn-UseC_-Telco-Customer-Churn.csv",
                        header = TRUE, na.strings = c("NA", "", "#NA"), sep = ",")
```

To have the full picture of the data let's print the data summary and sample.

customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines
0002-ORFBO: 1	Female:3488	Min. :0.0000	No :3641	No :4933	Min. : 0.00	No : 682	No :3390
0003-MKNFE: 1	Male :3555	1st Qu.:0.0000	Yes:3402	Yes:2110	1st Qu.: 9.00	Yes:6361	No phone service: 682
0004-TLHLJ: 1		Median :0.0000			Median :29.00		Yes :2971
0011-IGKFF: 1		Mean :0.1621			Mean :32.37		
0013-EXCHZ: 1		3rd Qu.:0.0000			3rd Qu.:55.00		
0013-MHZWF: 1		Max. :1.0000			Max. :72.00		
(Other) :7037							

InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport	StreamingTV	StreamingMusic
DSL :2421	No :3498	No :3088	No :3095	No :3473	No :2810	No :2710
Fiber optic:3096	No internet service:1526	No internet service:1526	No internet service:1526	No internet service:1526	No internet service:1526	No internet service:1526
No :1526	Yes :2019	Yes :2429	Yes :2422	Yes :2044	Yes :2707	Yes :2707

PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges	Churn
No :2872	Bank transfer (automatic):1544	Min. : 18.25	Min. : 18.8	No :5174
Yes:4171	Credit card (automatic) :1522	1st Qu.: 35.50	1st Qu.: 401.4	Yes:1869
	Electronic check :2365	Median : 70.35	Median :1397.5	
	Mailed check :1612	Mean : 64.76	Mean :2283.3	
		3rd Qu.: 89.85	3rd Qu.:3794.7	
		Max. :118.75	Max. :8684.8	
			NA's :11	

Table 2: telco customer churn data Summary

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection
1	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	Yes	No
2	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	No	Yes
3	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	Yes	No
4	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	No	Yes
5	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	No	No
6	9305-CDSKC	Female	0	No	No	8	Yes	Yes	Fiber optic	No	No	Yes
7	1452-KIOVK	Male	0	No	Yes	22	Yes	Yes	Fiber optic	No	Yes	No
8	6713-OKOMC	Female	0	No	No	10	No	No phone service	DSL	Yes	No	No
9	7892-POOKP	Female	0	Yes	No	28	Yes	Yes	Fiber optic	No	No	Yes
10	6388-TABGU	Male	0	No	Yes	62	Yes	No	DSL	Yes	Yes	No

TechSupport	StreamingTV	StreamingMovies	Contract	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges	Churn
No	No	No	Month-to-month	Yes	Electronic check	29.85	29.85	No
No	No	No	One year	No	Mailed check	56.95	1889.50	No
No	No	No	Month-to-month	Yes	Mailed check	53.85	108.15	Yes
Yes	No	No	One year	No	Bank transfer (automatic)	42.30	1840.75	No
No	No	No	Month-to-month	Yes	Electronic check	70.70	151.65	Yes
No	Yes	Yes	Month-to-month	Yes	Electronic check	99.65	820.50	Yes
No	Yes	No	Month-to-month	Yes	Credit card (automatic)	89.10	1949.40	No
No	No	No	Month-to-month	No	Mailed check	29.75	301.90	No
Yes	Yes	Yes	Month-to-month	Yes	Electronic check	104.80	3046.05	Yes
No	No	No	One year	No	Bank transfer (automatic)	56.15	3487.95	No

Table 3: telco customer churn data

```
#To see the names of the rows in the dataset
names(customerData)
```

```
#> [1] "customerID"      "gender"           "SeniorCitizen"    "Partner"
#> [5] "Dependents"      "tenure"           "PhoneService"     "MultipleLines"
#> [9] "InternetService" "OnlineSecurity"   "OnlineBackup"     "DeviceProtection"
#> [13] "TechSupport"     "StreamingTV"      "StreamingMovies"   "Contract"
#> [17] "PaperlessBilling" "PaymentMethod"    "MonthlyCharges"   "TotalCharges"
#> [21] "Churn"
```

```
#Display the dataset structure and summary
str(customerData)
```

```
#> 'data.frame':   7043 obs. of  21 variables:
#> $ customerID      : Factor w/ 7043 levels "0002-ORFBO","0003-MKNFE",...: 5376 3963 2565 5536 6512 6552 1000 ...
#> $ gender          : Factor w/ 2 levels "Female","Male": 1 2 2 2 1 1 2 1 1 2 ...
#> $ SeniorCitizen   : int  0 0 0 0 0 0 0 0 0 0 ...
#> $ Partner         : Factor w/ 2 levels "No","Yes": 2 1 1 1 1 1 1 1 2 1 ...
#> $ Dependents      : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 2 1 1 2 ...
#> $ tenure          : int  1 34 2 45 2 8 22 10 28 62 ...
#> $ PhoneService    : Factor w/ 2 levels "No","Yes": 1 2 2 1 2 2 2 1 2 2 ...
#> $ MultipleLines   : Factor w/ 3 levels "No","No phone service",...: 2 1 1 2 1 3 3 2 3 1 ...
#> $ InternetService : Factor w/ 3 levels "DSL","Fiber optic",...: 1 1 1 1 2 2 2 1 2 1 ...
#> $ OnlineSecurity  : Factor w/ 3 levels "No","No internet service",...: 1 3 3 3 1 1 1 3 1 3 ...
#> $ OnlineBackup    : Factor w/ 3 levels "No","No internet service",...: 3 1 3 1 1 1 3 1 1 3 ...
#> $ DeviceProtection: Factor w/ 3 levels "No","No internet service",...: 1 3 1 3 1 3 1 1 3 1 ...
#> $ TechSupport     : Factor w/ 3 levels "No","No internet service",...: 1 1 1 3 1 1 1 1 3 1 ...
#> $ StreamingTV     : Factor w/ 3 levels "No","No internet service",...: 1 1 1 1 1 3 3 1 3 1 ...
#> $ StreamingMovies : Factor w/ 3 levels "No","No internet service",...: 1 1 1 1 1 3 1 1 3 1 ...
#> $ Contract        : Factor w/ 3 levels "Month-to-month",...: 1 2 1 2 1 1 1 1 2 ...
#> $ PaperlessBilling: Factor w/ 2 levels "No","Yes": 2 1 2 1 2 2 2 1 2 1 ...
#> $ PaymentMethod   : Factor w/ 4 levels "Bank transfer (automatic)",...: 3 4 4 1 3 3 2 4 3 1 ...
#> $ MonthlyCharges  : num  29.9 57 53.9 42.3 70.7 ...
#> $ TotalCharges    : num  29.9 1889.5 108.2 1840.8 151.7 ...
#> $ Churn           : Factor w/ 2 levels "No","Yes": 1 1 2 1 2 2 1 1 2 1 ...
```

```
#Display first rows of the dataset
head(customerData)
```

```
#>   customerID gender SeniorCitizen Partner Dependents tenure PhoneService
#> 1 7590-VHVEG Female           0     Yes         No         1           No
#> 2 5575-GNVDE  Male           0     No          No        34           Yes
#> 3 3668-QPYBK  Male           0     No          No         2           Yes
#> 4 7795-CFOCW  Male           0     No          No        45           No
#> 5 9237-HQITU Female           0     No          No         2           Yes
#> 6 9305-CDSKC Female           0     No          No         8           Yes
```

```
#>      MultipleLines InternetService OnlineSecurity OnlineBackup DeviceProtection
#> 1 No phone service          DSL          No          Yes          No
#> 2          No          DSL          Yes          No          Yes
#> 3          No          DSL          Yes          Yes          No
#> 4 No phone service          DSL          Yes          No          Yes
#> 5          No      Fiber optic          No          No          No
#> 6          Yes      Fiber optic          No          No          Yes
#>      TechSupport StreamingTV StreamingMovies      Contract PaperlessBilling
#> 1          No          No          No Month-to-month          Yes
#> 2          No          No          No      One year          No
#> 3          No          No          No Month-to-month          Yes
#> 4          Yes          No          No      One year          No
#> 5          No          No          No Month-to-month          Yes
#> 6          No          Yes          Yes Month-to-month          Yes
#>      PaymentMethod MonthlyCharges TotalCharges Churn
#> 1      Electronic check          29.85          29.85    No
#> 2          Mailed check          56.95         1889.50    No
#> 3          Mailed check          53.85          108.15   Yes
#> 4 Bank transfer (automatic)          42.30         1840.75    No
#> 5      Electronic check          70.70          151.65   Yes
#> 6      Electronic check          99.65          820.50   Yes
```

#To select just the continuous variables and summarise it

```
library(dplyr)
continues <- select_if(customerData, is.numeric)
#Sumarize the variables to find NA's and outliers
summary(continues)
```

```
#> SeniorCitizen      tenure      MonthlyCharges      TotalCharges
#> Min.   :0.0000   Min.   : 0.00   Min.   : 18.25   Min.   : 18.8
#> 1st Qu.:0.0000   1st Qu.: 9.00   1st Qu.: 35.50   1st Qu.: 401.4
#> Median :0.0000   Median :29.00   Median : 70.35   Median :1397.5
#> Mean   :0.1621   Mean   :32.37   Mean   : 64.76   Mean   :2283.3
#> 3rd Qu.:0.0000   3rd Qu.:55.00   3rd Qu.: 89.85   3rd Qu.:3794.7
#> Max.   :1.0000   Max.   :72.00   Max.   :118.75   Max.   :8684.8
#>                                     NA's   :11
```

#Display the factor columns and summarise it

```
factorColumns <- select_if(customerData, is.factor)
summary(factorColumns)
```

```
#>      customerID      gender      Partner      Dependents PhoneService
#> 0002-ORFBO: 1 Female:3488 No :3641 No :4933 No : 682
#> 0003-MKNFE: 1 Male :3555 Yes:3402 Yes:2110 Yes:6361
#> 0004-TLHLJ: 1
#> 0011-IGKFF: 1
#> 0013-EXCHZ: 1
#> 0013-MHZWF: 1
#> (Other) :7037
#>      MultipleLines      InternetService      OnlineSecurity
#> No :3390 DSL :2421 No :3498
#> No phone service: 682 Fiber optic:3096 No internet service:1526
#> Yes :2971 No :1526 Yes :2019
#>
#>
#>
#>      OnlineBackup      DeviceProtection
#> No :3088 No :3095
#> No internet service:1526 No internet service:1526
#> Yes :2429 Yes :2422
#>
#>
#>
#>
```

```

#>           TechSupport           StreamingTV
#> No           :3473 No           :2810
#> No internet service:1526 No internet service:1526
#> Yes           :2044 Yes           :2707
#>
#>
#>
#>           StreamingMovies           Contract           PaperlessBilling
#> No           :2785 Month-to-month:3875 No :2872
#> No internet service:1526 One year :1473 Yes:4171
#> Yes           :2732 Two year :1695
#>
#>
#>
#>           PaymentMethod Churn
#> Bank transfer (automatic):1544 No :5174
#> Credit card (automatic) :1522 Yes:1869
#> Electronic check :2365
#> Mailed check :1612
#>
#>
#>

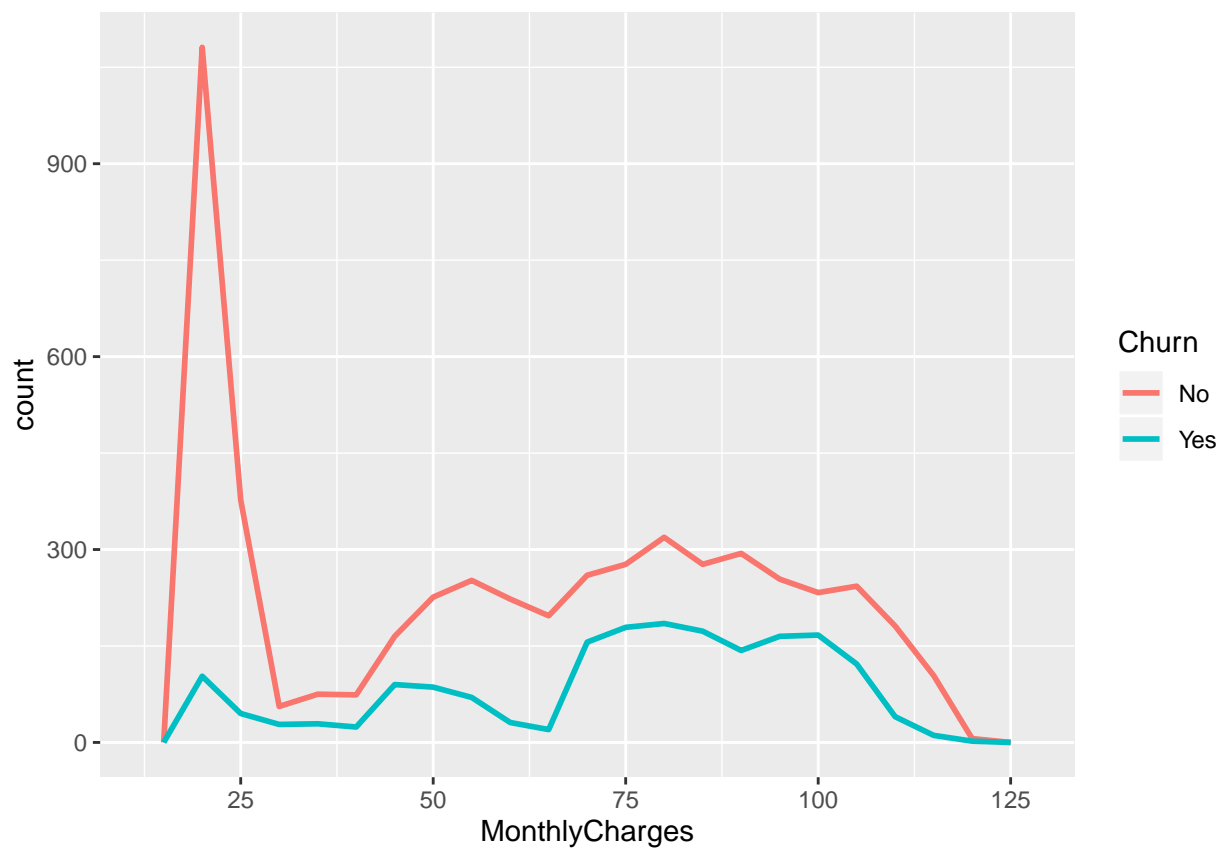
```

Missing Data

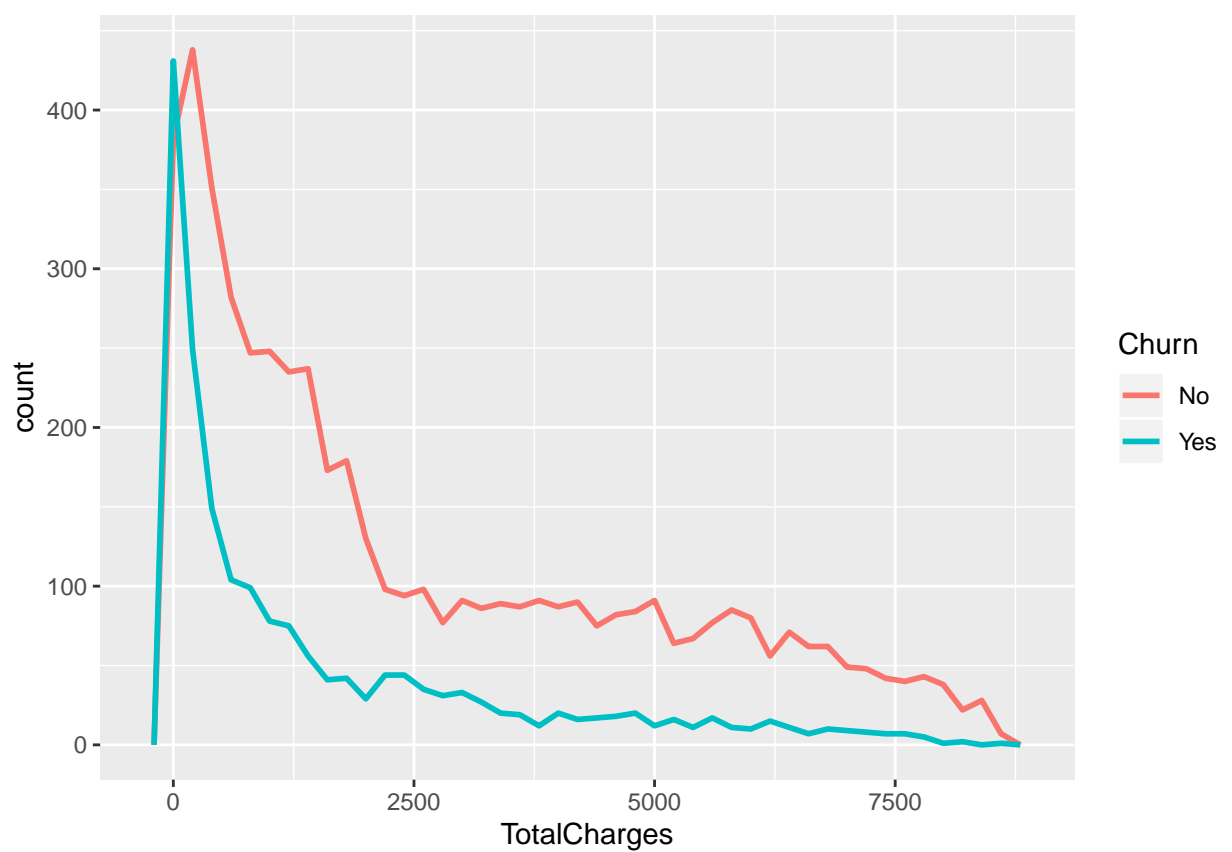
From the above summary, we can observe that there are only 11 TotalCharges NA. Considering the number of missing data is quite small, we can remove them directly.

Continuous Variables

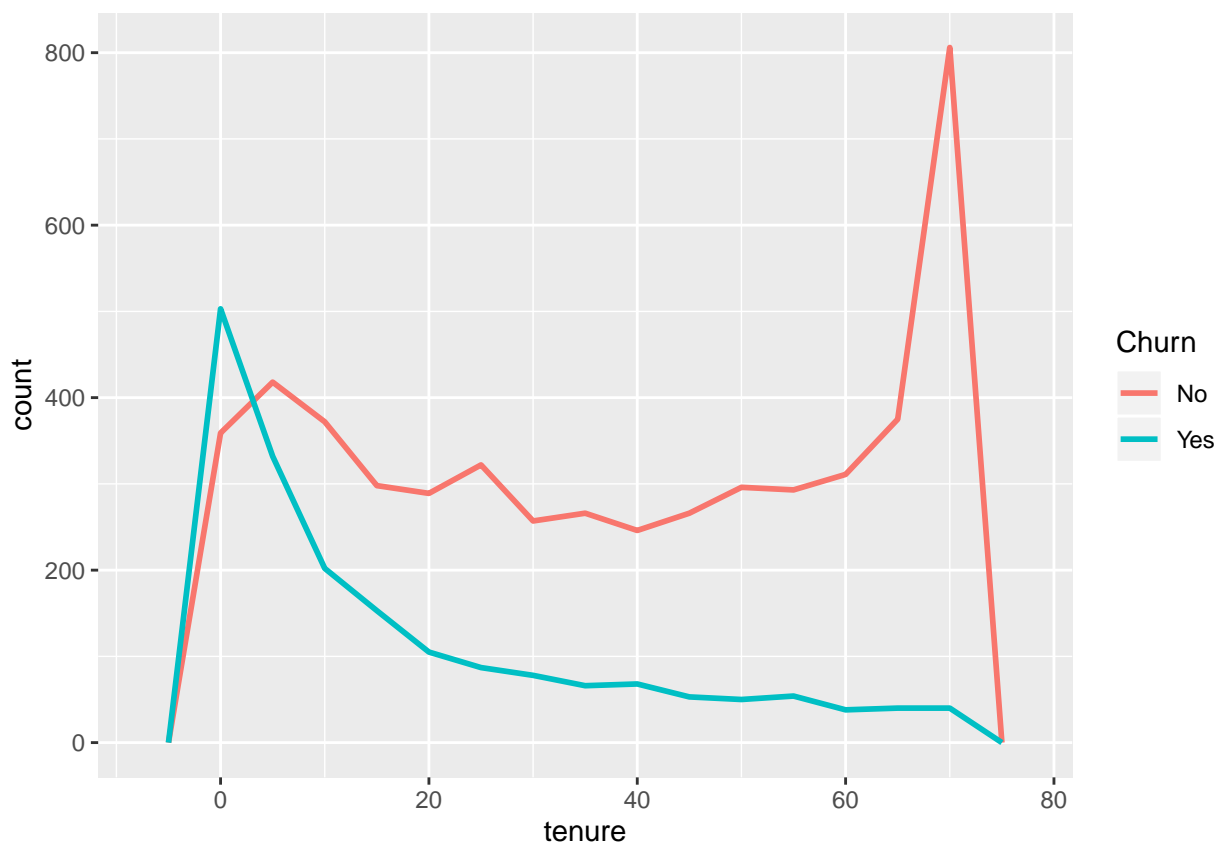
For continuous variables, let's check for their distributions.



The number of current customers with MonthlyCharges below \$25 is extremely high. For the customers with Monthlycharges greater than \$30, the distributions are similar between who churned and who did not churn.



The distribution of TotalCharges is highly positive skew for all customers no matter whether they churned or not.



The distributions for tenure are very different between customers who churned and who didn't churn. For customers who churned, the distribution is positive skew, which means customers who churned are more likely to cancel the service in the first couple of months. For current customers who didn't churn, there are two spikes. The second spike is much more drastic than the first one, which means a large group of current customers have been using the service more than 5 years. There is no obvious outliers for 3 numeric variables.

Data correlation and other observations

We plot the relationship between predictive variable and Churn to identify the possible key variable.

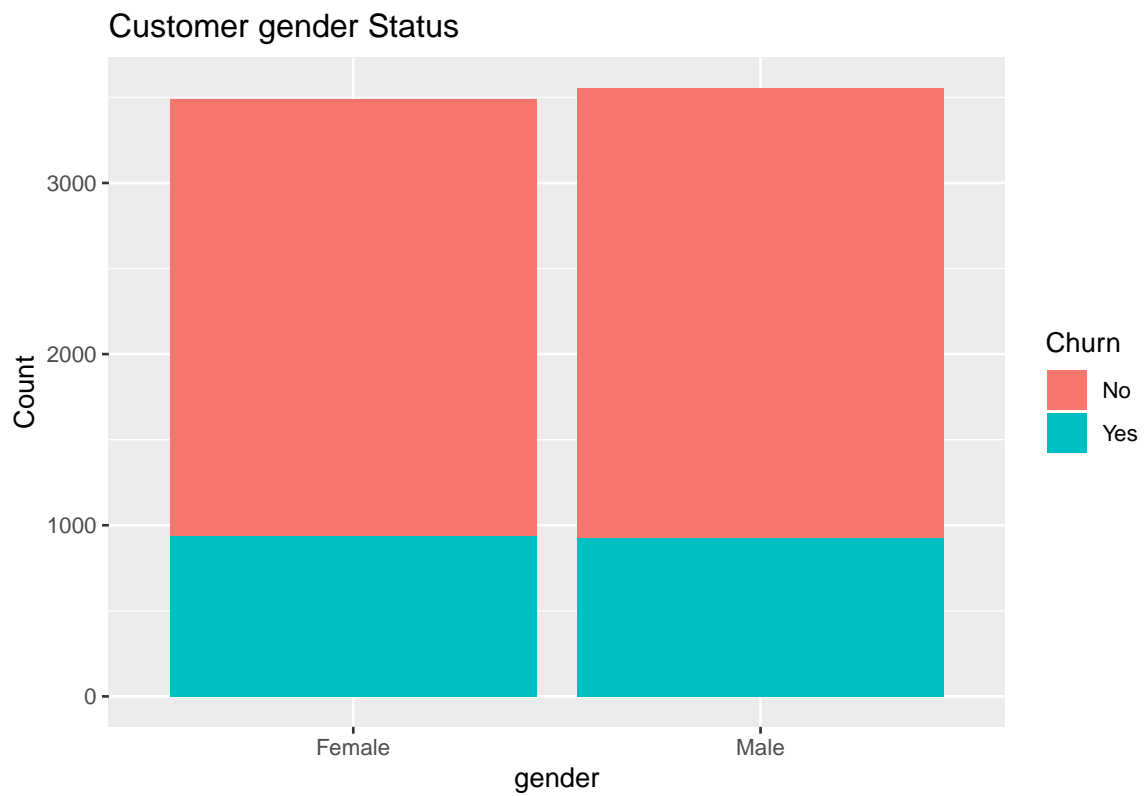


Figure 1

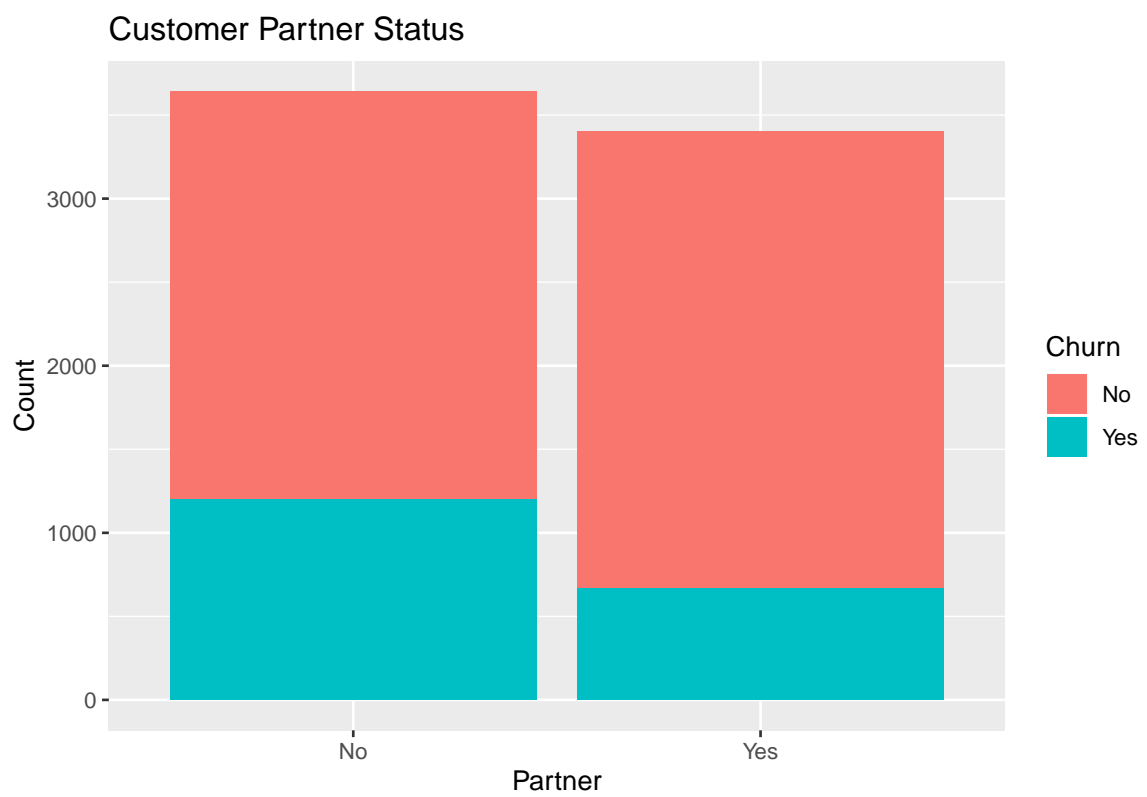


Figure 2

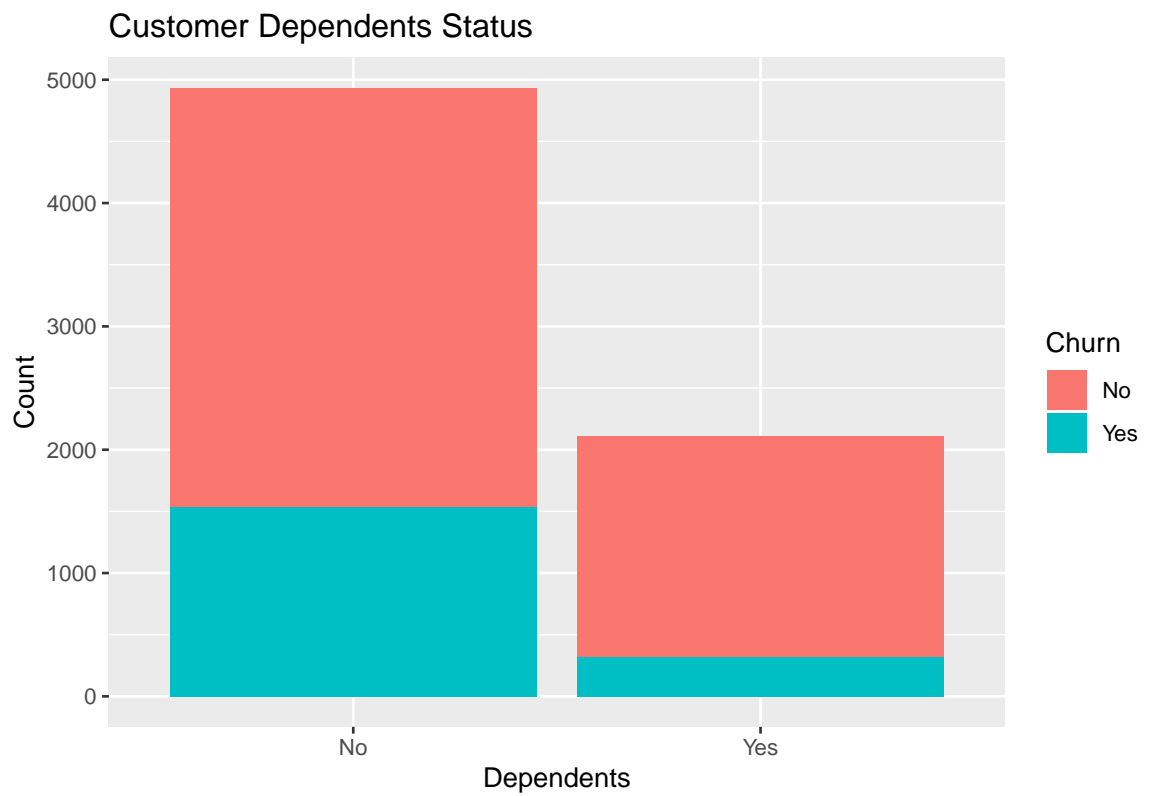


Figure 3

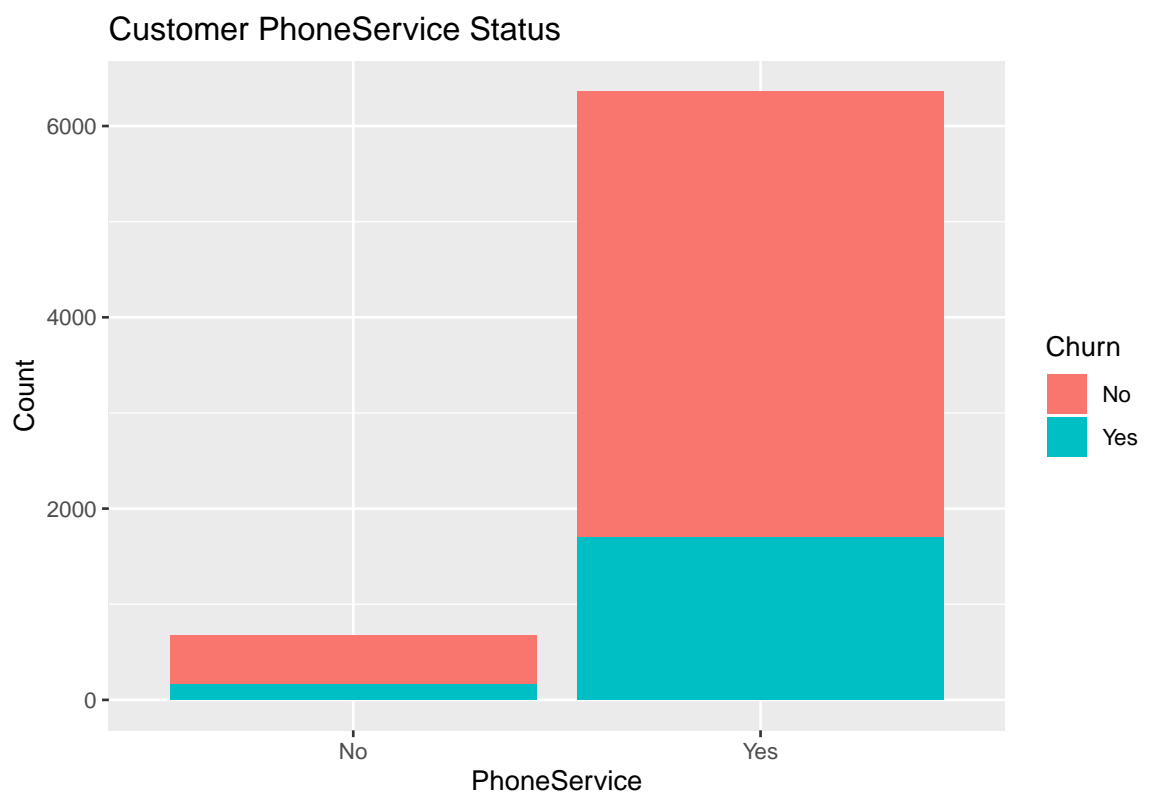


Figure 4

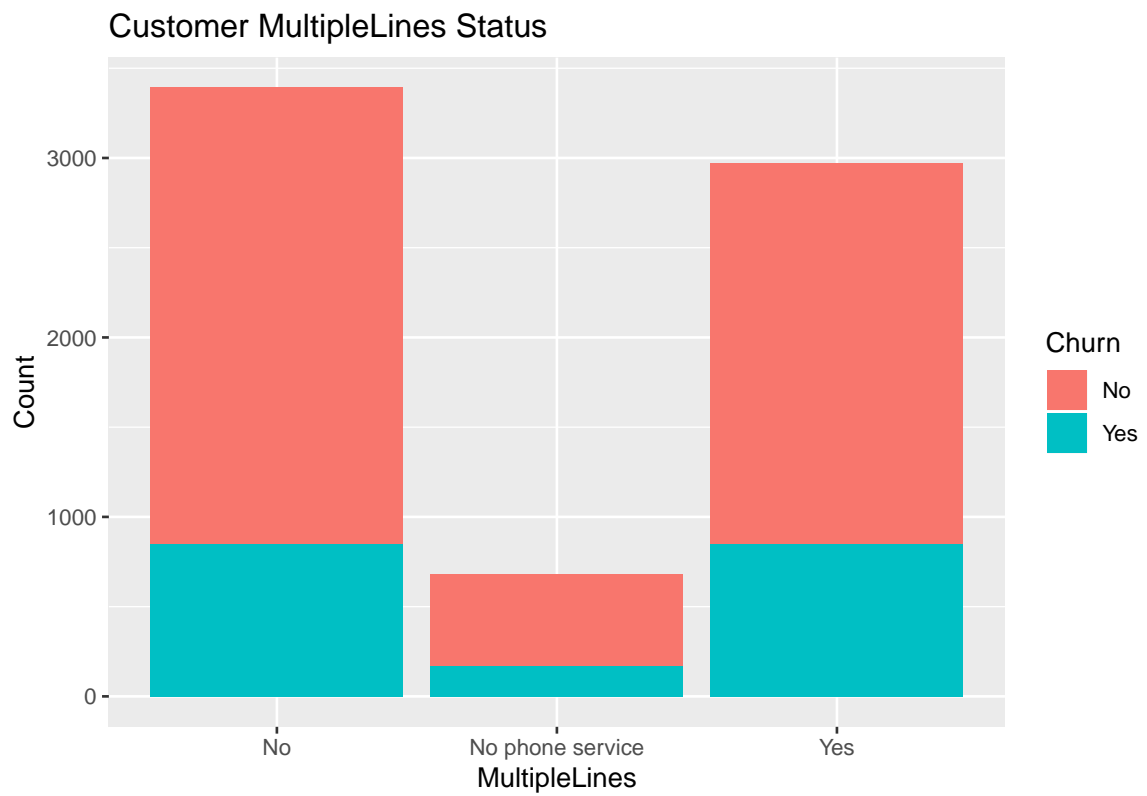


Figure 5

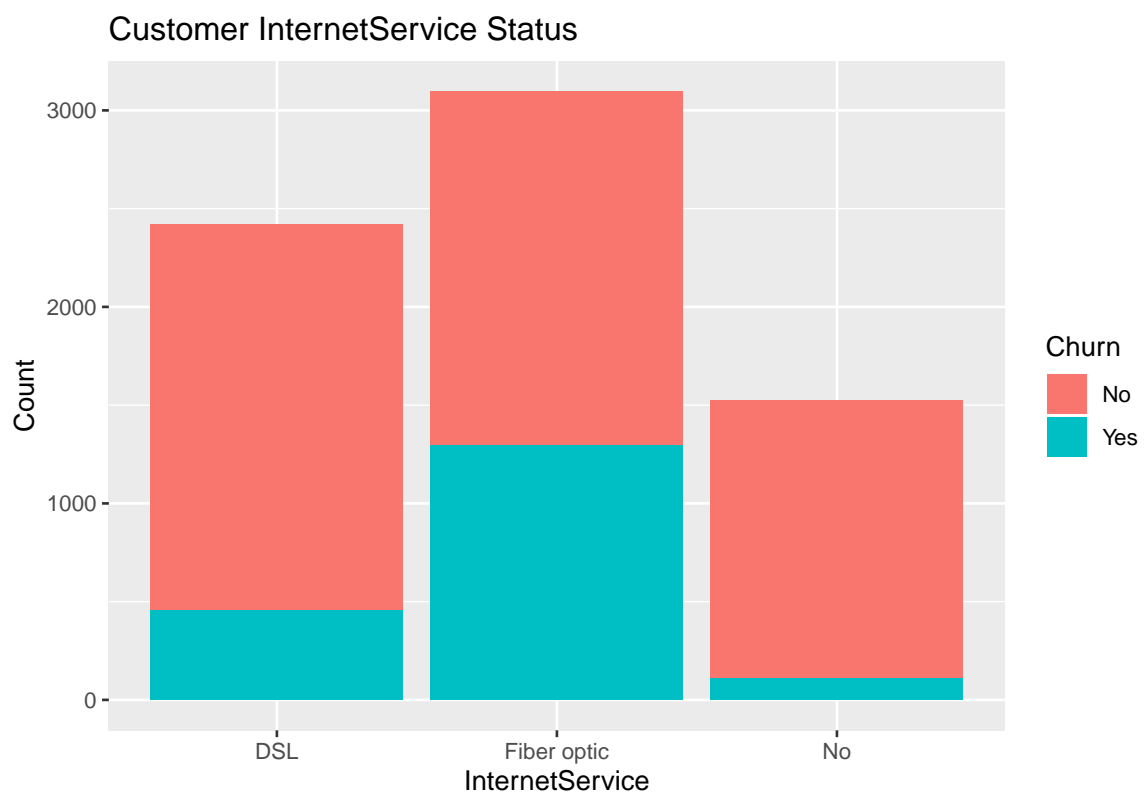


Figure 6



Figure 7

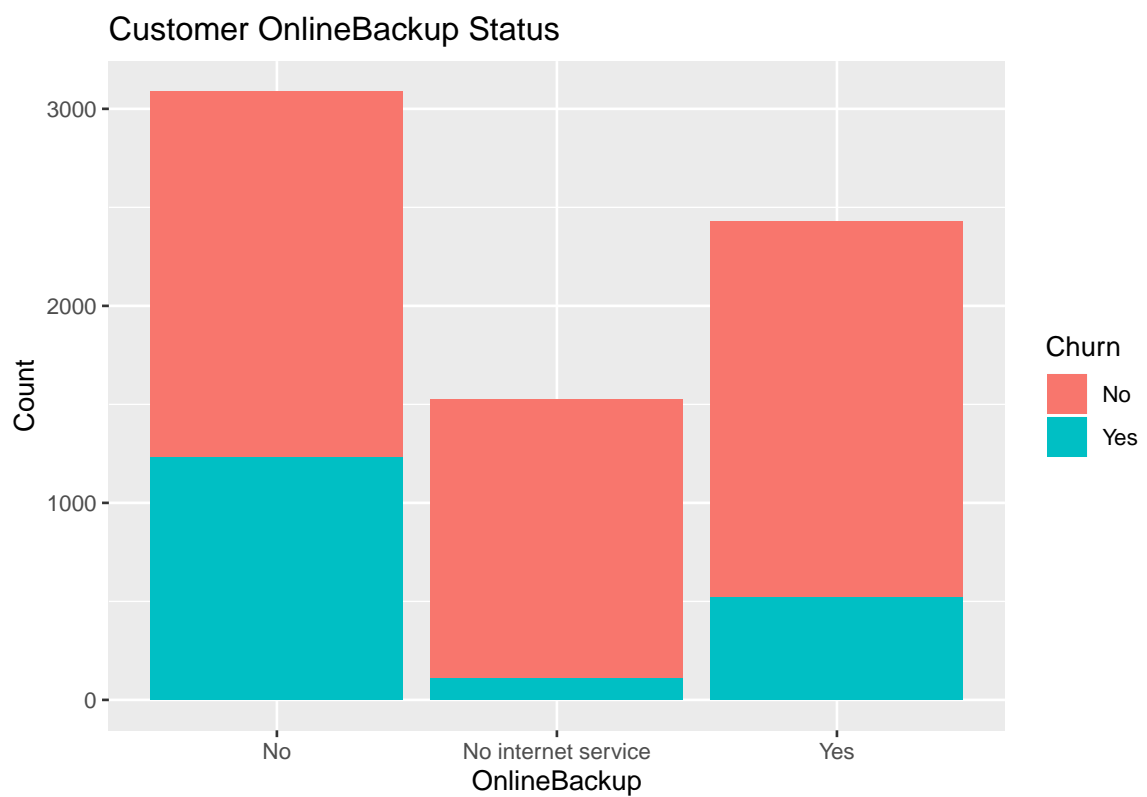


Figure 8

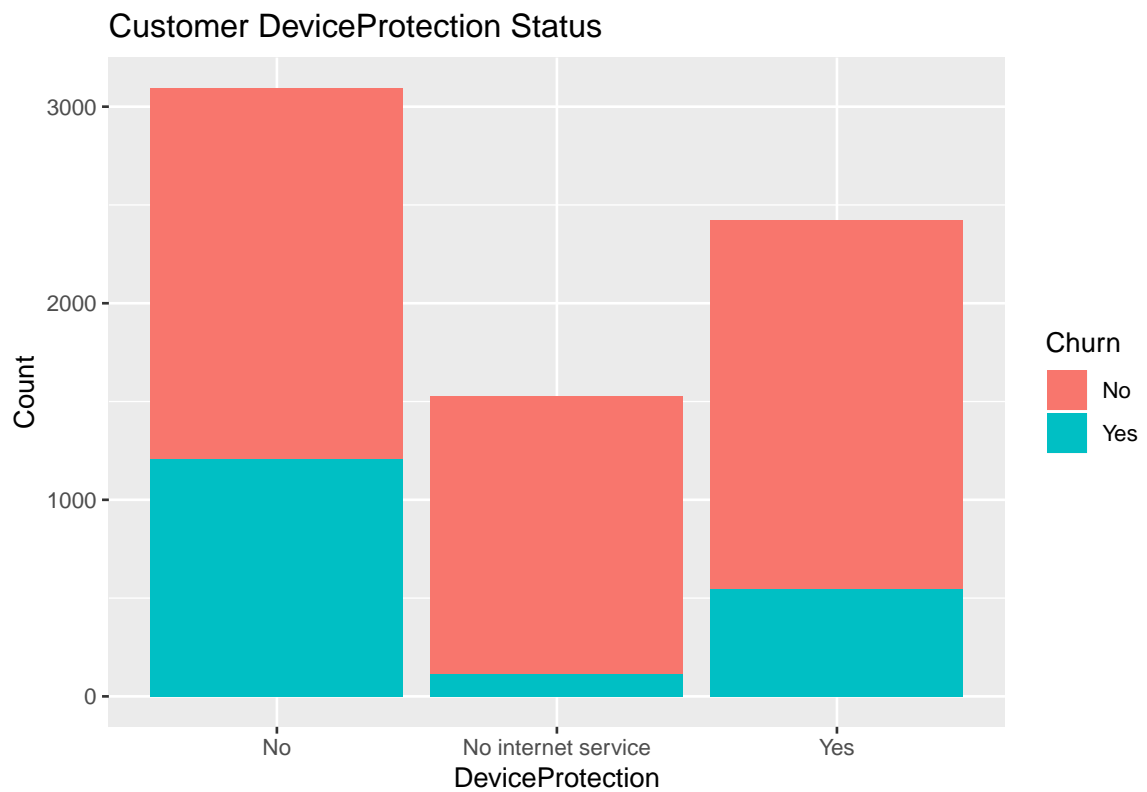


Figure 9

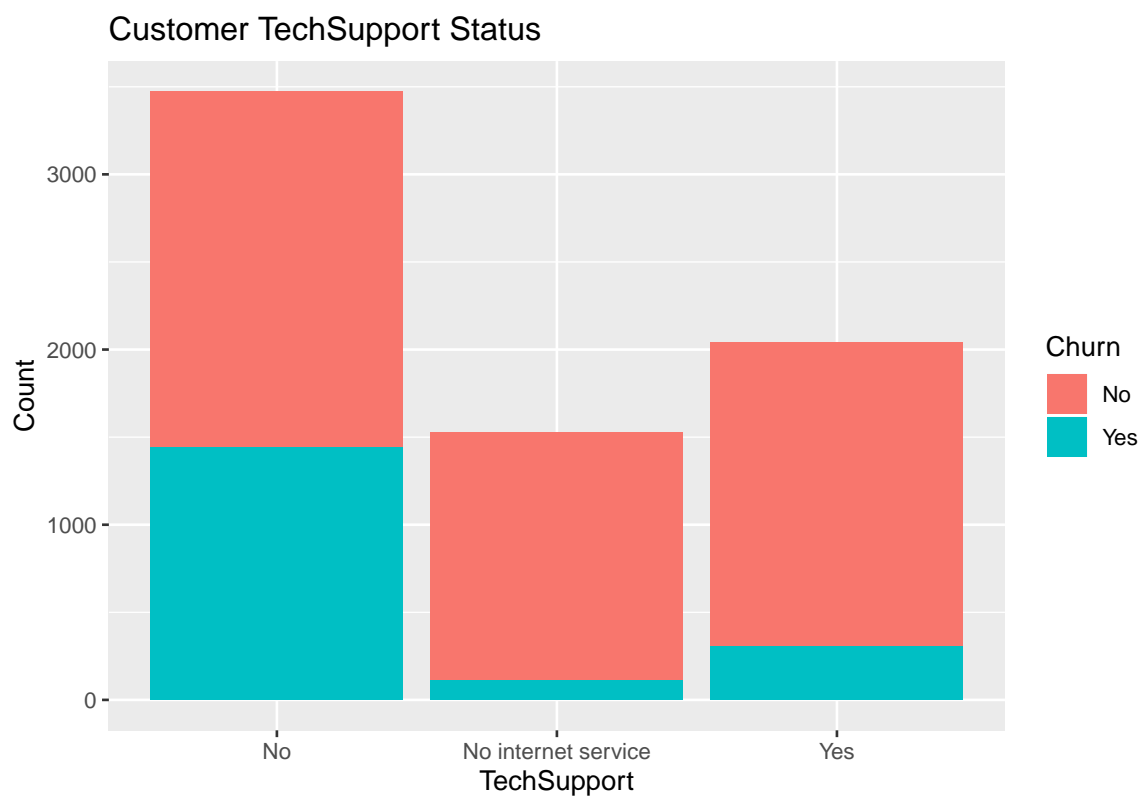


Figure 10

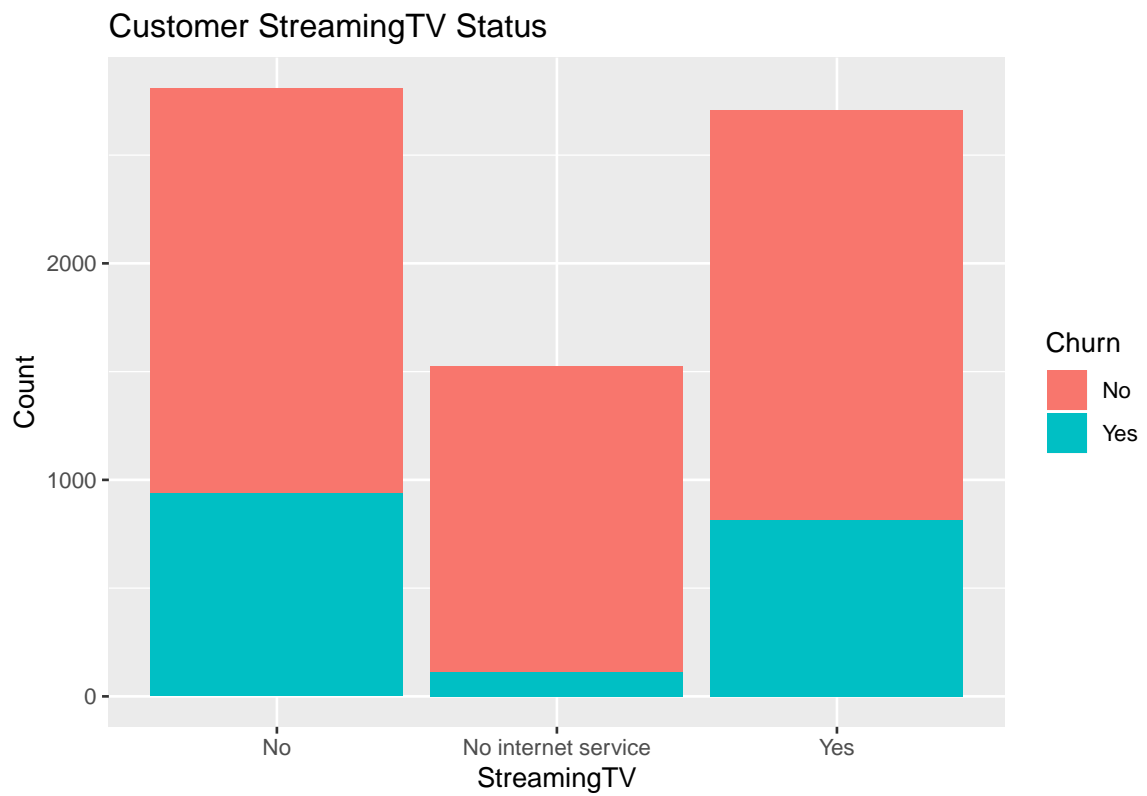


Figure 11

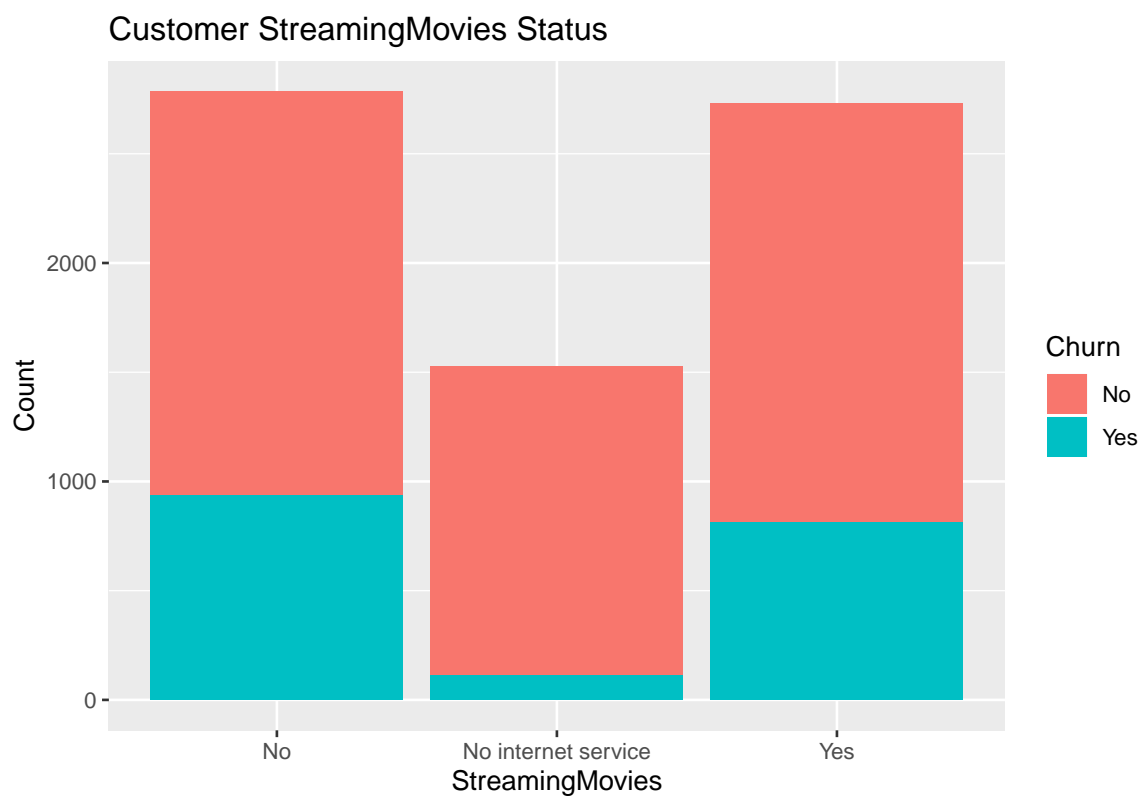


Figure 12

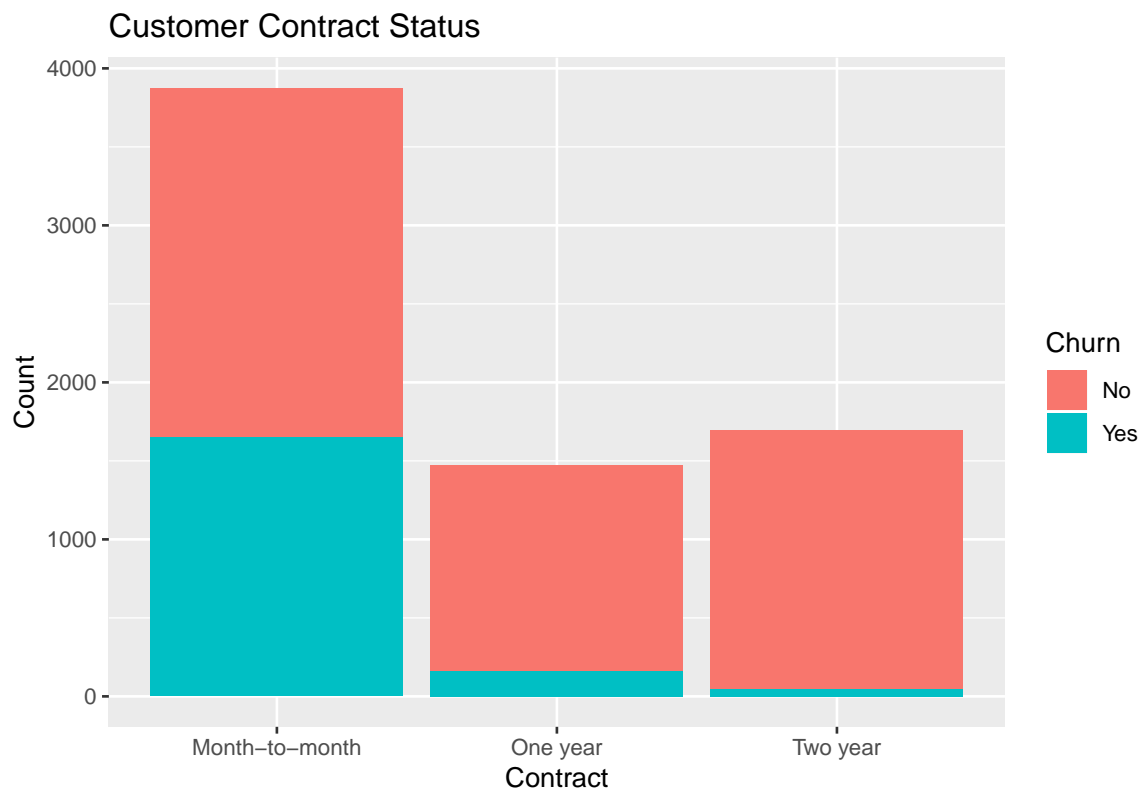


Figure 13

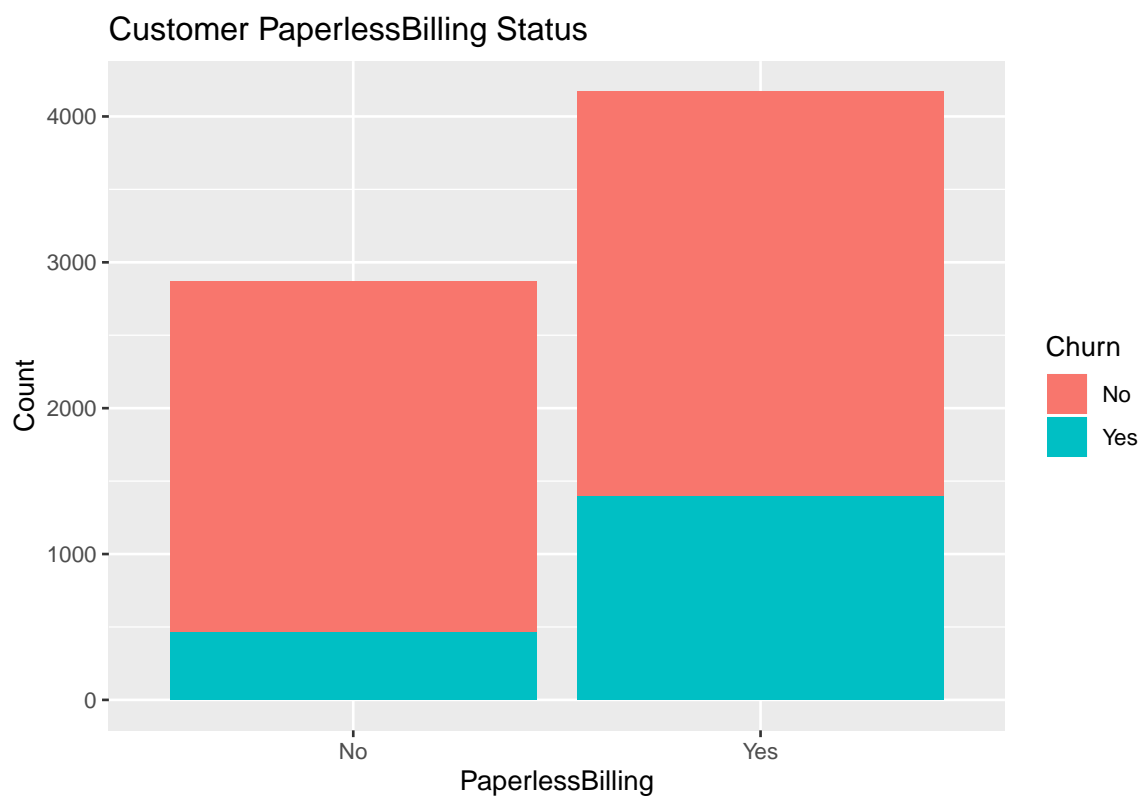


Figure 14

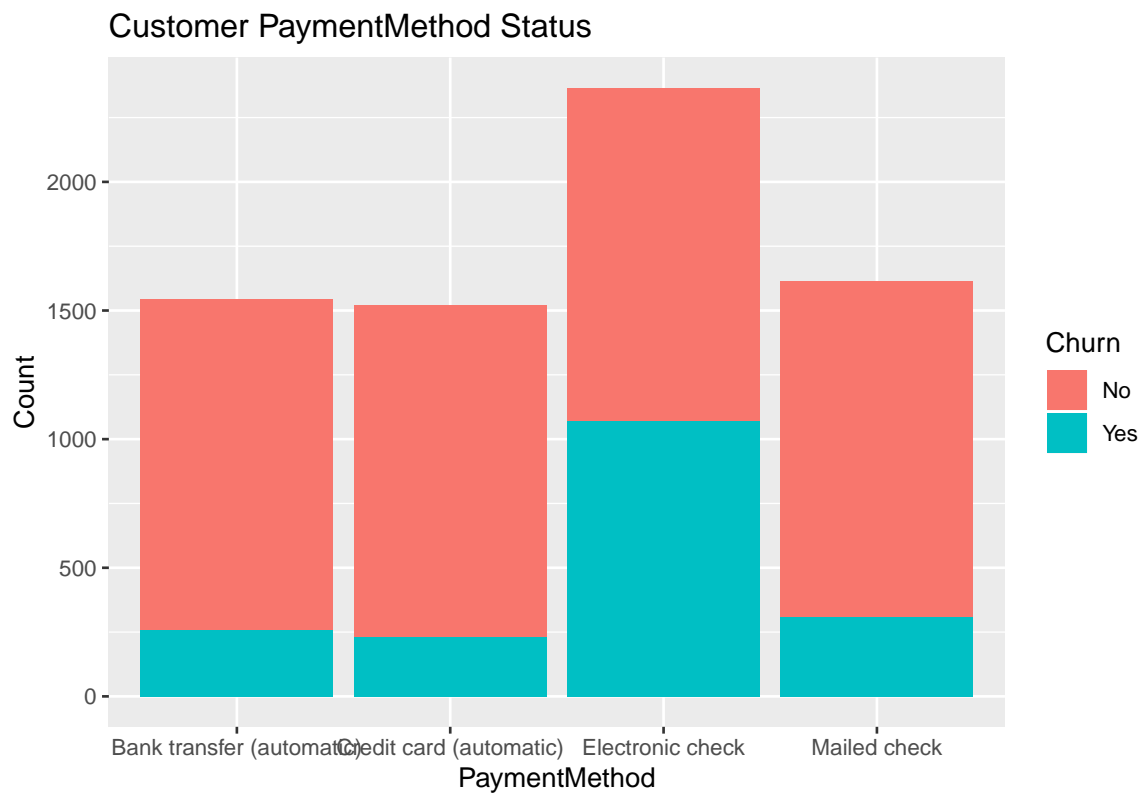


Figure 15: Test1

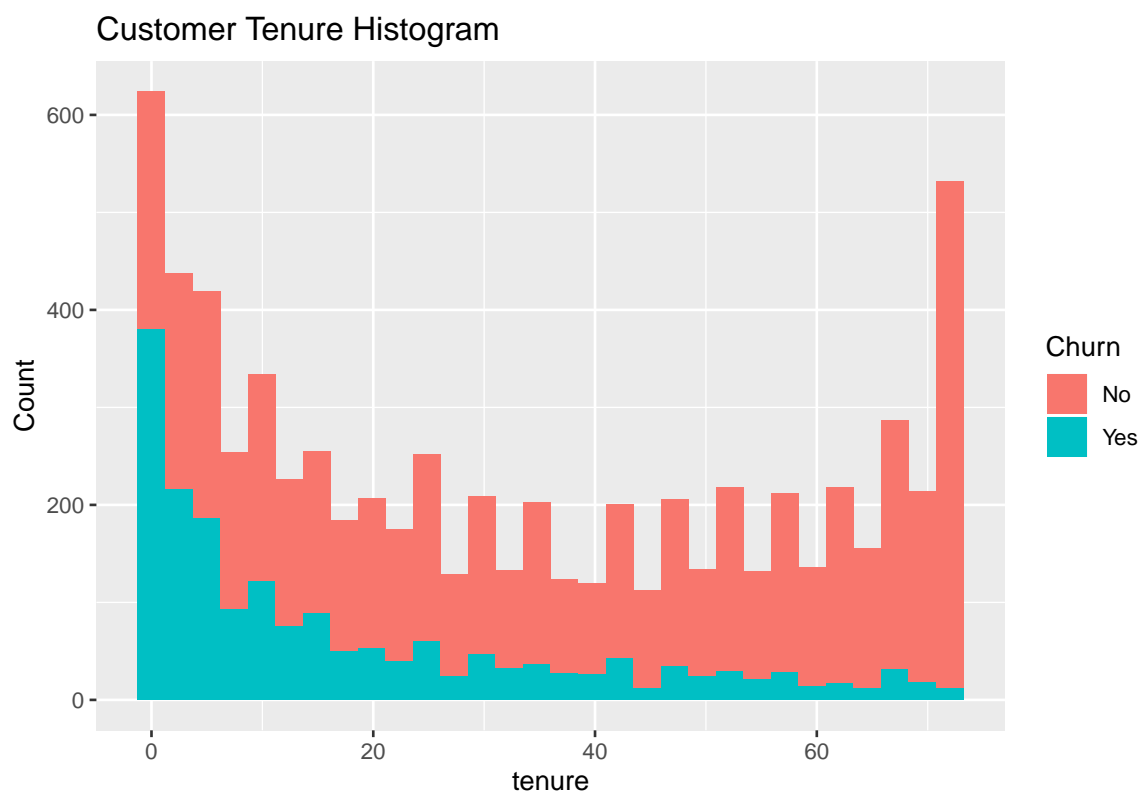


Figure 16

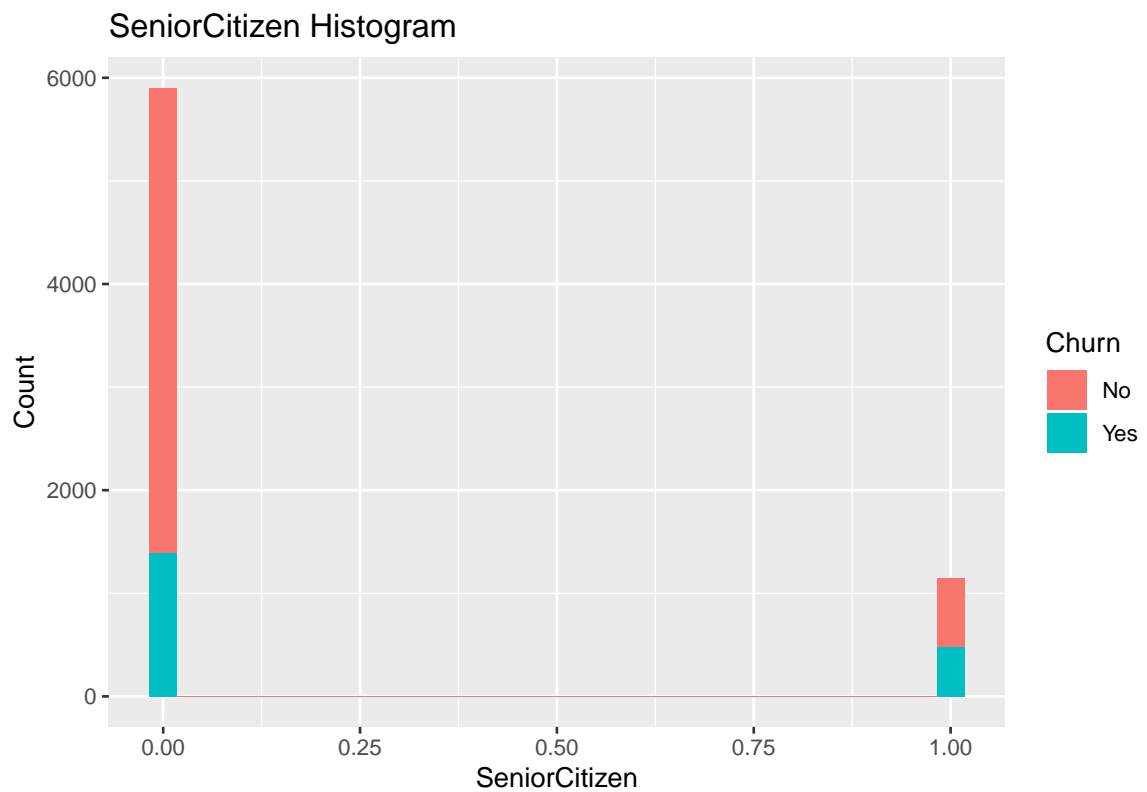


Figure 17

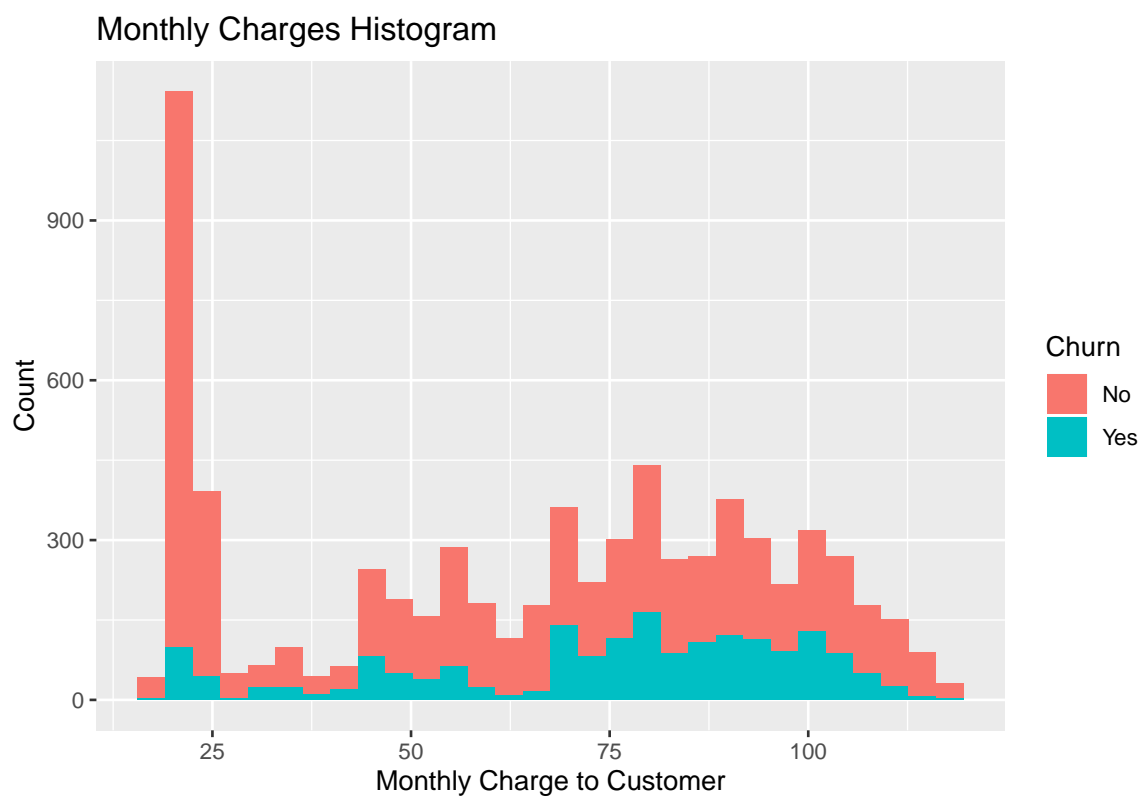
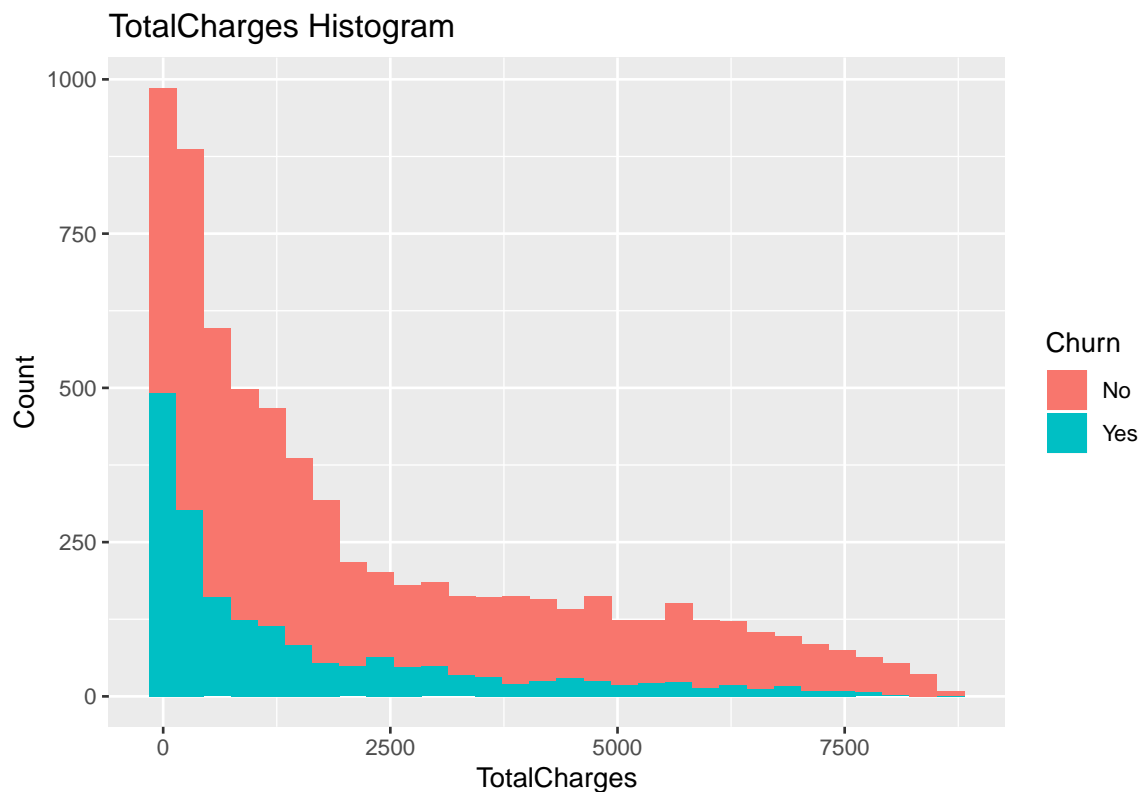


Figure 18

**Figure 19**

General Insights From Initial Data Exploration These are simple observations made by looking at the above charts.

- Customers with a partner or dependents are much less likely to cancel their service
- The longer a customer's tenure, the less likely they are to cancel
- Customers with fiber optic internet service are especially likely to cancel
- Customers on month-to-month contracts are especially likely to cancel
- Customers who use paperless billing are more likely to cancel
- Customers who pay by electronic check are more likely to cancel

We guess that Having a partner is highly correlated with having dependents. Plot the correlation between them.

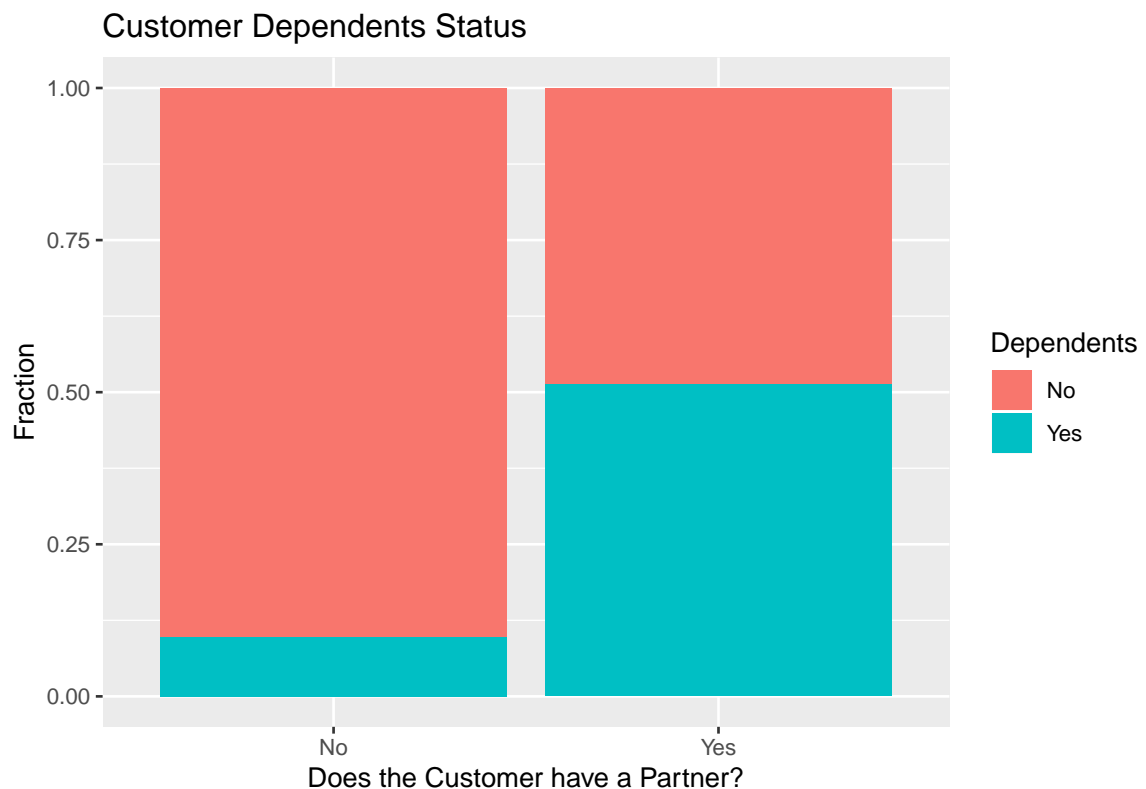


Figure 20

From the above plot, our guess is verified.

Next, we use the other method to visualize the feature-target relationships.

```
customer_churn_tbl = read.csv("../data/WA_Fn-UseC_-Telco-Customer-Churn.csv" )
data("customer_churn_tbl")

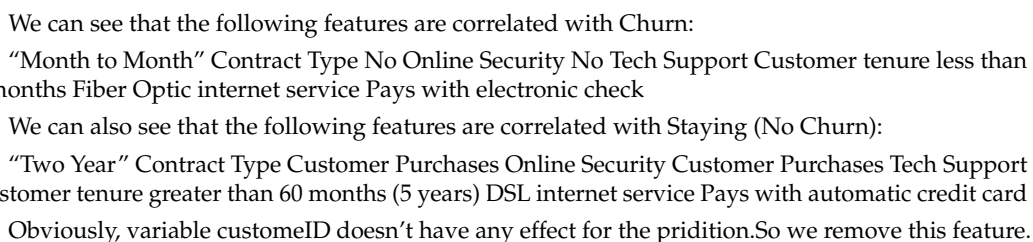
customer_churn_binarized_tbl <- customer_churn_tbl %>%
  dplyr::select(-customerID) %>%
  mutate(TotalCharges = ifelse(is.na(TotalCharges), MonthlyCharges, TotalCharges)) %>%
  binarize(n_bins = 5, thresh_infreq = 0.01, name_infreq = "OTHER", one_hot = TRUE)

customer_churn_corr_tbl <- customer_churn_binarized_tbl %>%
  correlate(Churn__Yes)

customer_churn_corr_tbl

#> # A tibble: 60 x 3
#>   feature      bin      correlation
#>   <fct>      <chr>      <dbl>
#> 1 Churn      No          -1
#> 2 Churn      Yes           1
#> 3 Contract   Month-to-month  0.405
#> 4 OnlineSecurity No          0.343
#> 5 TechSupport No          0.337
#> 6 tenure     -Inf_6        0.309
#> 7 InternetService Fiber_optic  0.308
#> 8 Contract   Two_year     -0.302
#> 9 PaymentMethod Electronic_check 0.302
#> 10 OnlineBackup No          0.268
#> # ... with 50 more rows

customer_churn_corr_tbl %>%
  plot_correlation_funnel()
```



Modeling and Evaluation

Feature Selection

```
customerData = mutate(customerData,
  gender = as.factor(unclass(gender)),
  Partner = as.factor(unclass(Partner)),
  Dependents = as.factor(unclass(Dependents)),
  PhoneService = as.factor(unclass(PhoneService)),
```

```

MultipleLines = as.factor(unclass(MultipleLines)),
InternetService = as.factor(unclass(InternetService)),
OnlineSecurity = as.factor(unclass(OnlineSecurity)),
OnlineBackup = as.factor(unclass(OnlineBackup)),
DeviceProtection = as.factor(unclass(DeviceProtection)),
TechSupport = as.factor(unclass(TechSupport)),
StreamingTV = as.factor(unclass(StreamingTV)),
StreamingMovies = as.factor(unclass(StreamingMovies)),
Contract = as.factor(unclass(Contract)),
PaperlessBilling = as.factor(unclass(PaperlessBilling)),
PaymentMethod = as.factor(unclass(PaymentMethod)),
Churn = as.factor(unclass(Churn))

```

It is time to run feature selection algorithm.

```

predictors = subset(customerData,select = -Churn)
label = customerData[,20]

# run the RFE algorithm
rfePrediction = rfe(predictors, label, sizes=c(1:19),
                    rfeControl = rfeControl(functions=rffuncs, method="cv", number=3))
print(rfePrediction)

#>
#> Recursive feature selection
#>
#> Outer resampling method: Cross-Validated (3 fold)
#>
#> Resampling performance over subset size:
#>
#> Variables Accuracy Kappa AccuracySD KappaSD Selected
#>      1  0.7527 0.2560  0.0054356 0.022580
#>      2  0.7685 0.3564  0.0076594 0.020001
#>      3  0.7726 0.3491  0.0015382 0.052940
#>      4  0.7699 0.3799  0.0014982 0.006421
#>      5  0.7874 0.4019  0.0038711 0.020980
#>      6  0.7915 0.4186  0.0046993 0.010468
#>      7  0.7968 0.4340  0.0030267 0.006644
#>      8  0.7986 0.4448  0.0044541 0.007512      *
#>      9  0.7912 0.4340  0.0043785 0.005856
#>     10  0.7935 0.4404  0.0074016 0.016981
#>     11  0.7958 0.4471  0.0049446 0.012006
#>     12  0.7917 0.4316  0.0006517 0.010464
#>     13  0.7941 0.4350  0.0008881 0.014438
#>     14  0.7958 0.4383  0.0027094 0.010557
#>     15  0.7955 0.4370  0.0021045 0.017705
#>     16  0.7932 0.4276  0.0059778 0.022605
#>     17  0.7918 0.4222  0.0055461 0.021613
#>     18  0.7944 0.4299  0.0063134 0.023861
#>     19  0.7954 0.4316  0.0065307 0.021233
#>
#> The top 5 variables (out of 8):
#>      tenure, TotalCharges, MonthlyCharges, Contract, TechSupport

```

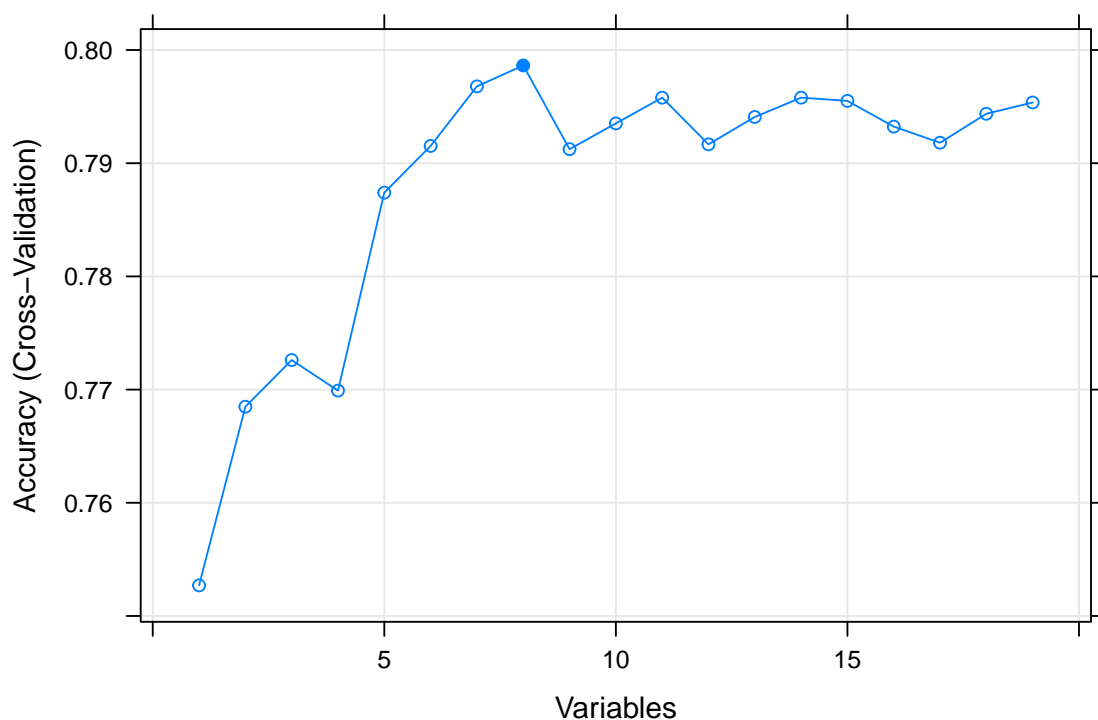


Figure 21: Number of Predictors vs Accuracy

Figure 21 illustrates that the accuracy practically flattens out when a number of predictors reaches 8. The accuracy improves a bit more when a number of features reaches 8 but the gain is negligible. Here is the list of features ordered by importance. We take first nine for model training.

```
#> [1] "tenure"          "TotalCharges"    "MonthlyCharges"  "Contract"
#> [5] "TechSupport"    "OnlineSecurity"  "InternetService" "OnlineBackup"
```

Data Upsampling

There is one more step to make before we get to the model training. As shown in Figure ?? our data set is unbalanced. This could cause model over-fitting. So let's split the data into the training and testing sets and up-sample the training set.

```
set.seed(1608)

# keep only the selected features
finalSample = customerData %>% dplyr::select(c(selectedPredictors,"Churn"));

splitIdx = createDataPartition(finalSample$Churn, p=0.7, list = F) # 70% training data
trainData = finalSample[splitIdx, ]
testData = finalSample[-splitIdx, ]

set.seed(590045)
columns = colnames(trainData)
trainData = upSample(x = trainData[, columns[columns != "Churn"] ],
  y = trainData$Churn, list = F, yname = "Churn")

rm(splitIdx, columns, finalSample)
print(table(trainData$Churn))

#>
#>    1    2
#> 3615 3615
```

As we can see now the training set is balanced.

Thus we have prepared our training and test data sets. We have identified the most important features. We are ready to work on the prediction models.

Decision Tree Model

Decision Tree algorithm is simple to understand, interpret and visualize. Effort required for data preparation is small

```
#> Setting levels: control = no, case = yes

#> Setting direction: controls < cases

#> Conditional Inference Tree
#>
#> 7230 samples
#> 8 predictor
#> 2 classes: 'no', 'yes'
#>
#> No pre-processing
#> Resampling: Cross-Validated (5 fold)
#> Summary of sample sizes: 5784, 5784, 5784, 5784, 5784
#> Resampling results across tuning parameters:
#>
#>   mincriterion   ROC         Sens       Spec
#> 0.01           0.8552514  0.7125864  0.8420470
#> 0.50           0.8506961  0.7195021  0.8320885
#> 0.99           0.8392769  0.6829876  0.8406639
#>
#> ROC was used to select the optimal model using the largest value.
#> The final value used for the model was mincriterion = 0.01.

confusionMatrix(data = pred.decisionTreeModel.raw, testDataCopy$Churn)

#> Confusion Matrix and Statistics
#>
#>           Reference
#> Prediction  no  yes
#>      no  1050  107
#>      yes   498  453
#>
#>               Accuracy : 0.713
#>               95% CI : (0.6932, 0.7322)
#>      No Information Rate : 0.7343
#>      P-Value [Acc > NIR] : 0.9871
#>
#>               Kappa : 0.3984
#>
#>  Mcnemar's Test P-Value : <2e-16
#>
#>               Sensitivity : 0.6783
#>               Specificity : 0.8089
#>               Pos Pred Value : 0.9075
#>               Neg Pred Value : 0.4763
#>               Prevalence : 0.7343
#>               Detection Rate : 0.4981
#>      Detection Prevalence : 0.5489
#>      Balanced Accuracy : 0.7436
#>
#>      'Positive' Class : no
#>
```

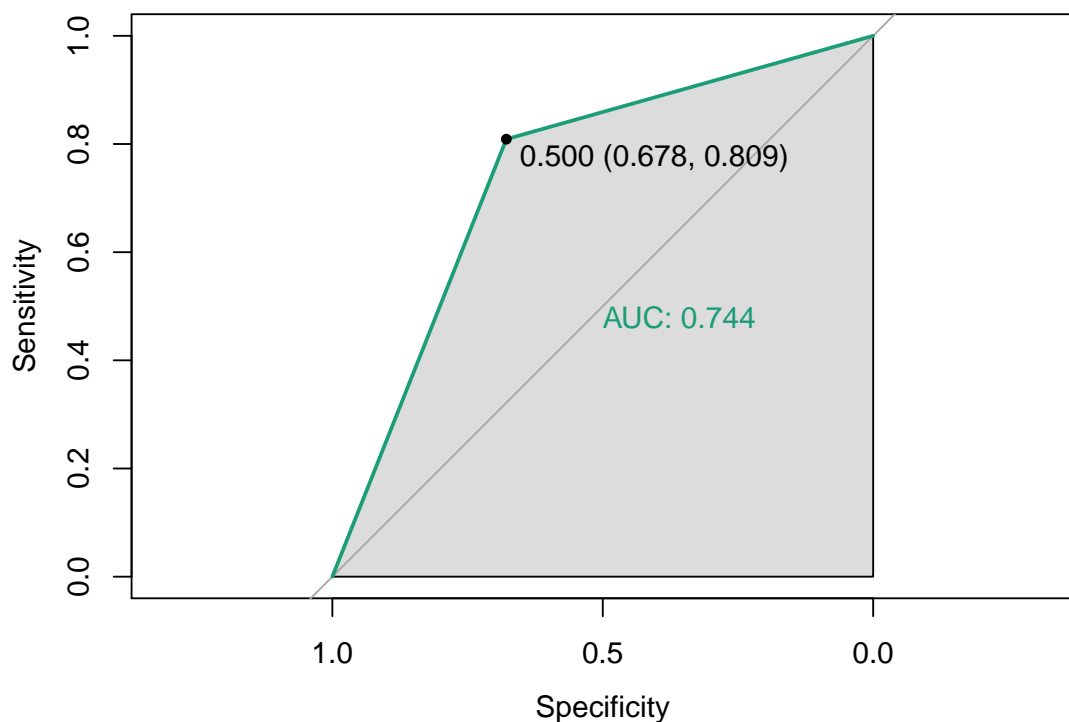


Figure 22: Classification Tree Model AUC and ROC Curve

Random Forest Model

Random Forest is also considered as a very handy and easy to use algorithm, because its default hyperparameters often produce a good prediction result. Random Forest adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model.

```
#> user system elapsed
#> 27.56 0.57 28.16

#> Random Forest
#>
#> 7230 samples
#> 8 predictor
#> 2 classes: 'no', 'yes'
#>
#> No pre-processing
#> Resampling: Cross-Validated (3 fold)
#> Summary of sample sizes: 4820, 4820, 4820
#> Resampling results across tuning parameters:
#>
#> mtry ROC Sens Spec
#> 2 0.8552015 0.6946058 0.8489627
#> 7 0.9335415 0.7875519 0.9302905
#> 13 0.9311955 0.7850622 0.9247580
#>
#> ROC was used to select the optimal model using the largest value.
#> The final value used for the model was mtry = 7.

confusionMatrix(data = pred.randomForestModel.raw, testDataCopy$Churn)

#> Confusion Matrix and Statistics
#>
```

```

#>           Reference
#> Prediction  no  yes
#>         no 1258 213
#>         yes 290 347
#>
#>           Accuracy : 0.7614
#>           95% CI : (0.7426, 0.7794)
#>       No Information Rate : 0.7343
#>       P-Value [Acc > NIR] : 0.0024481
#>
#>           Kappa : 0.4141
#>
#>   McNemar's Test P-Value : 0.0007023
#>
#>           Sensitivity : 0.8127
#>           Specificity : 0.6196
#>       Pos Pred Value : 0.8552
#>       Neg Pred Value : 0.5447
#>           Prevalence : 0.7343
#>       Detection Rate : 0.5968
#>   Detection Prevalence : 0.6978
#>       Balanced Accuracy : 0.7162
#>
#>       'Positive' Class : no
#>

```

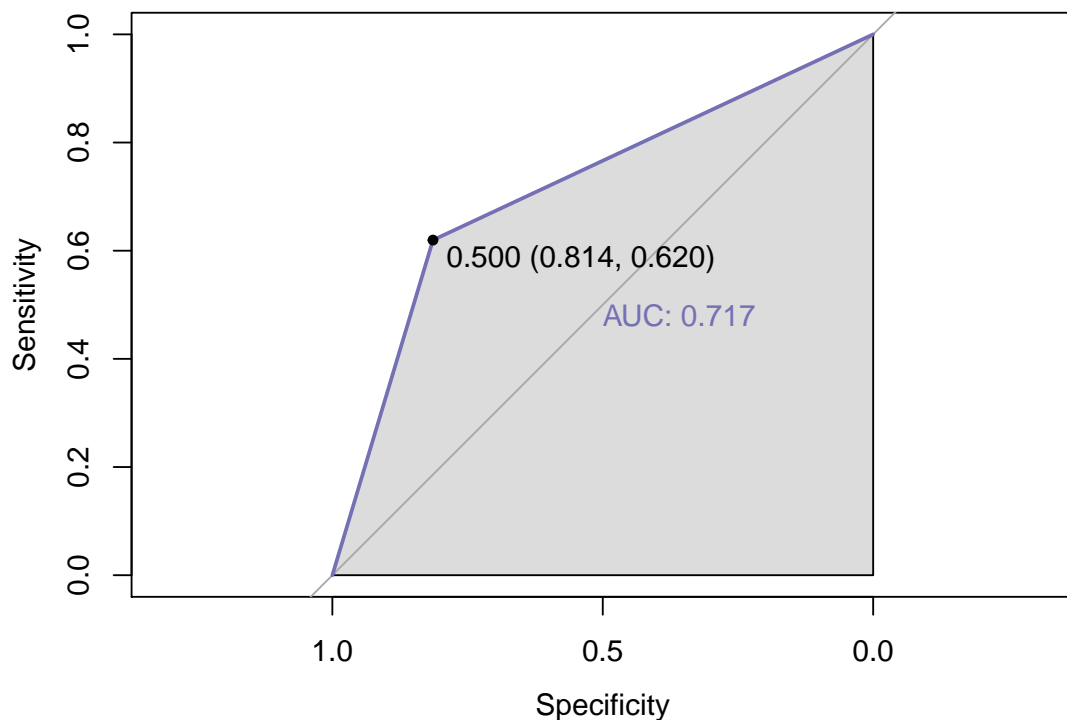


Figure 23: Random Forest Model AUC and ROC Curve

Logistic Regression Model

Logistic regression is an efficient, interpretable and accurate method, which fits quickly with minimal tuning. Logistic regression prediction accuracy will benefit if the data is close to Gaussian distribution. Thus we apply addition transformation to the training data set.

```
confusionMatrix(data = pred.logRegModel.raw, testDataCopy$Churn)
```



```

#> Confusion Matrix and Statistics
#>
#>           Reference
#> Prediction    1    2
#>           1 1129  125
#>           2  419  435
#>
#>           Accuracy : 0.7419
#>           95% CI : (0.7227, 0.7605)
#>           No Information Rate : 0.7343
#>           P-Value [Acc > NIR] : 0.2228
#>
#>           Kappa : 0.4335
#>
#>           Mcnemar's Test P-Value : <2e-16
#>
#>           Sensitivity : 0.7293
#>           Specificity : 0.7768
#>           Pos Pred Value : 0.9003
#>           Neg Pred Value : 0.5094
#>           Prevalence : 0.7343
#>           Detection Rate : 0.5356
#>           Detection Prevalence : 0.5949
#>           Balanced Accuracy : 0.7531
#>
#>           'Positive' Class : 1
#>

```

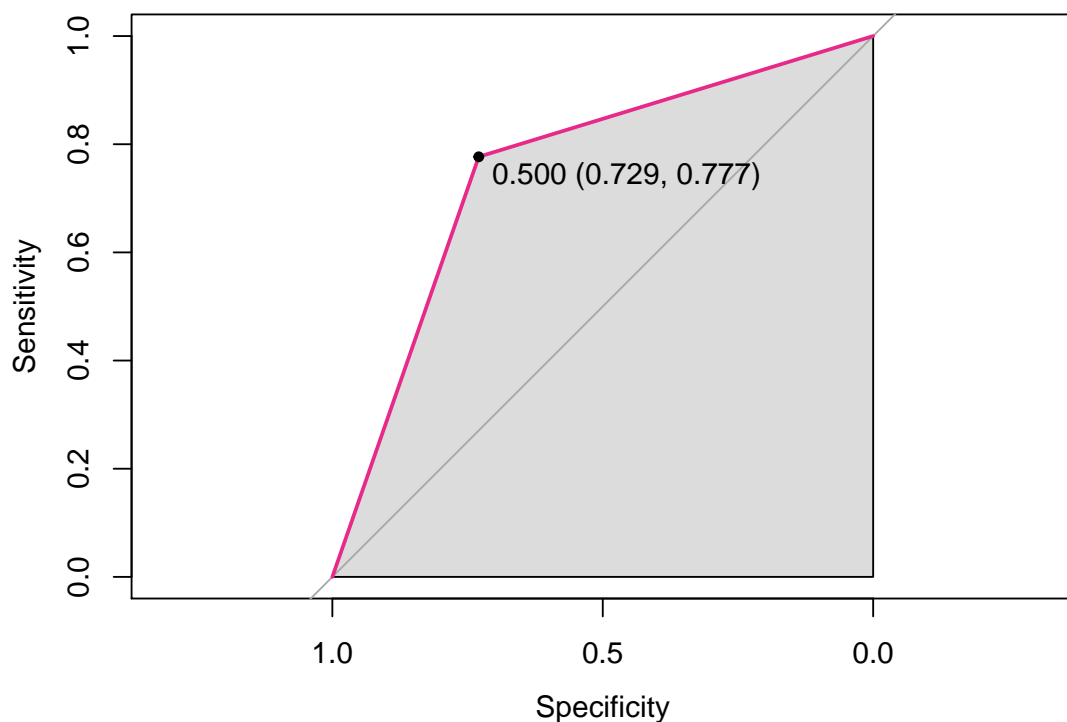


Figure 24: Logistic Regression Model AUC and ROC Curve

Model Comparison

Now it is time to compare the models side by side and pick a winner.

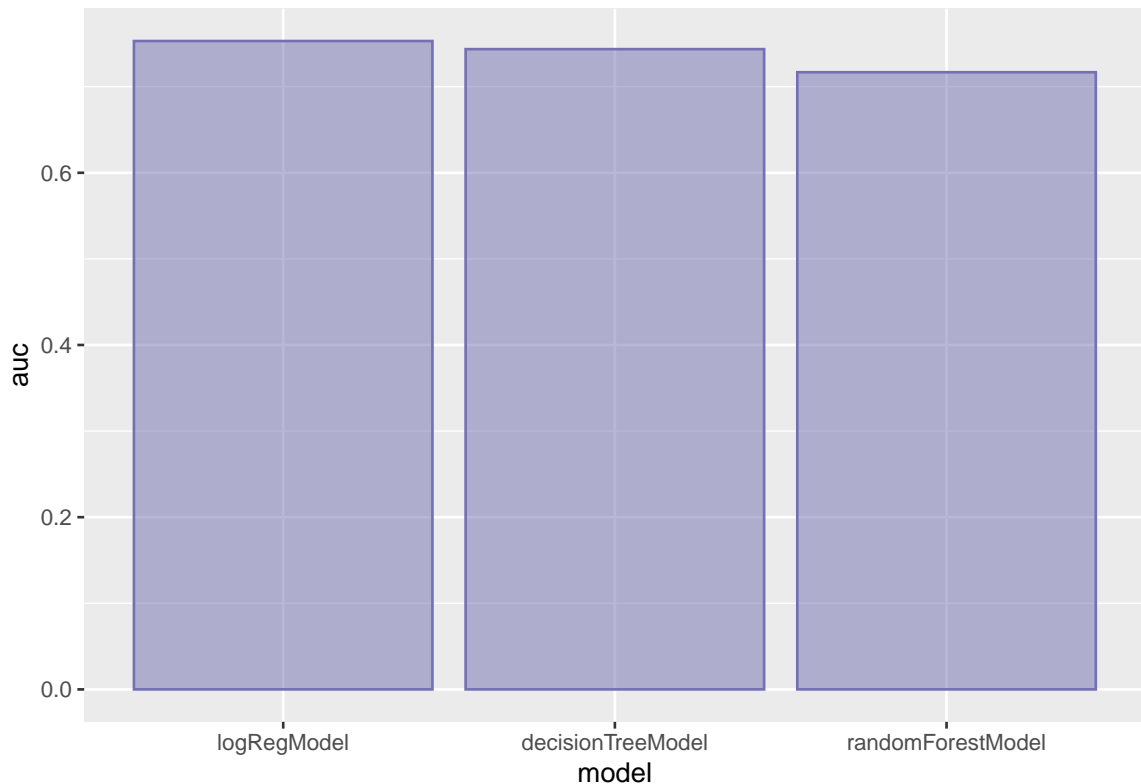


Figure 25: Model AUC Comparison

```
#>           model      auc
#> 1    logRegModel 0.7530569
#> 2 decisionTreeModel 0.7436116
#> 3 randomForestModel 0.7167982
```

AUC - ROC performance AUC stands for Area under the ROC Curve and ROC for Receiver operating characteristic curve. This is one of the most important KPIs of the classification algorithms. These two metrics measure how well the models distinguishing between the classes. The higher AUC the better model predicts positive and negative outcome.

Figures 22, 23, 24 and accompanying data show that on the test data set all the models demonstrated very close results. Random Forest has the highest overall accuracy (76%) but the balanced accuracy is lower (about 72%).

Logistic regression model scores the best having the highest AUC and all other metrics. It's balanced accuracy is 75%.

Model interpretability Logistic Regression, Decision Tree are all highly interpretable models. It is easy to explain to the business what impact each input parameter has. The decision tree could be visualized (provided if it is not too large).

Random Forest on the other hand is a black-box model, complex algorithm which is difficult to explain in simple terms.

Model Deployment

The model can demonstrate how various customer elements affect the probability of the churn. This model is helpful for the telco companies to identify the customers who can churn. For these customers, some retain actions should be taken accordingly. Currently we use the features provided by the dataset. However, in the future there could be new features show up, then we must update our model accordingly.

We also develop one shiny app. It is simple to understand and deploy.

Conclusion

Through exploring customer churn dataset, we selected and tuned a model to predict whether one customer could churn.

We commenced our research analyzing and understanding available data. Then we identified the missing data, its distribution. We continued our research selecting the most impactful data attributes to use as an input for our future model. We apply the feature identification algorithm to do the job.

When the data preparation phase was finished we picked and analysed four different classification models: Decision Tree, Random Forest and Logistic Regression. We conducted comparative analysis of the models, reviewed their strength and weaknesses. We fitted each model using K-fold cross-validation technique. Subsequently we evaluated performance of each model applying them to the test data set and comparing AUC - ROC and balanced accuracy metrics.

Finally we moved to identifying a winning model. In order to do so we reviewed each model from different angles including performance, interpretability and data quality sensitivity and data preparation effort.

The winning model scored the highest in the majority of the categories. It was Logistic Regression, which we employed to build a Shiny App Web application.

We consider the project to be a success.

Note from the Authors

This file was generated using *The R Journal* style article [template](#), additional information on how to prepare articles for submission is here - [Instructions for Authors](#). The article itself is an executable R Markdown file that could be [downloaded from Github](#) with all the necessary artifacts.

Ketao Li
York University

liketao@yahoo.com

Kush Halani
York University

kush.halani@ontariotechu.net

Josue Romain
York University

josue.rolland.romain@gmail.com

Juan Peña
York University

jppena62@my.yorku.ca

Priyanka Patil
York University

priyanka181994@gmail.com