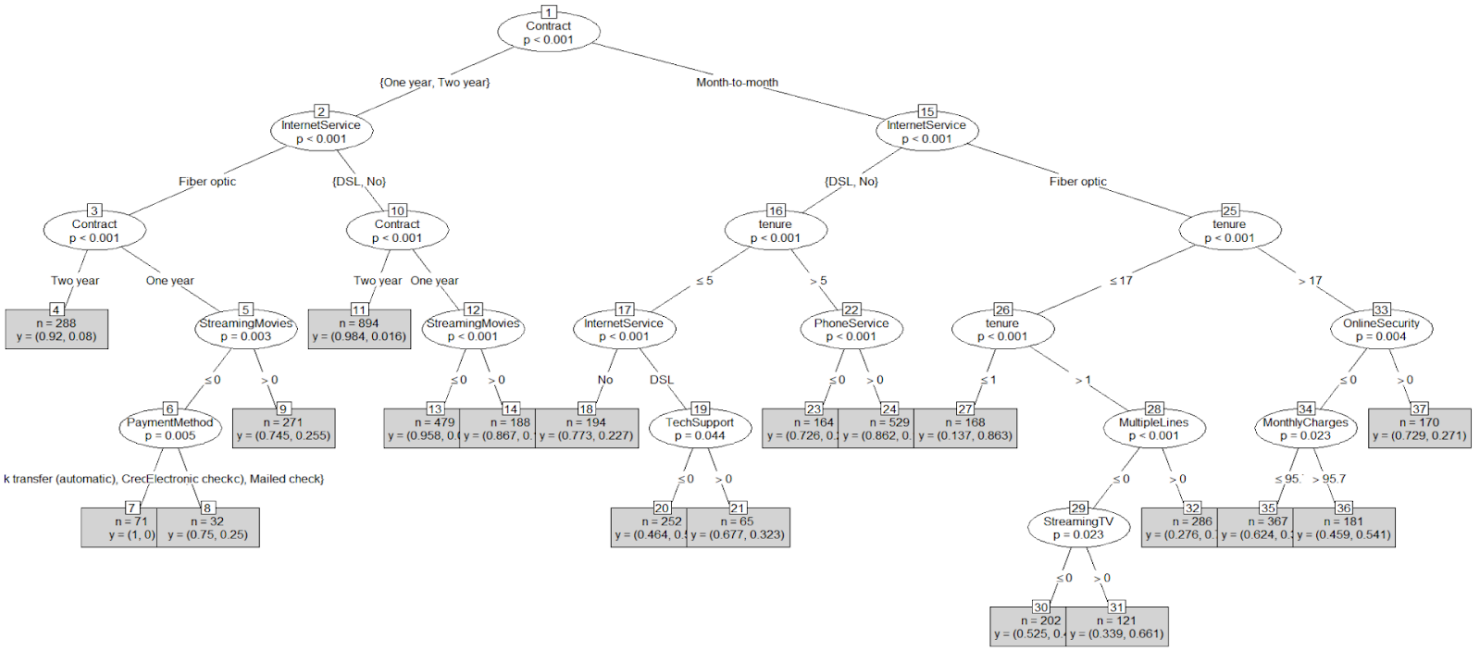
Open in app

Following 598K Followers



Random Forest Image, Image by Author

Telco Customer ChurnRate Analysis

The basic but detailed churn rate analysis

Shuheng.Ma 9 hours ago · 16 min read

In this blog, we will describe how we built basic but useful models to explain the churn rate based on the Kaggle Telco Customer dataset. The specific process includes (1) Background and Problem, (2) Data Summary and Exploratory Analysis, (3) Data Analyses, (4) Strategy Recommendations, Limitations, and Future Research.

This project was built by Shuheng. Ma, Li. Zhou. To see the full code used, find our GitHub.

Open in app

With the enormous increase in the number of customers using telephone services, the marketing division for a telco company wants to attract more new customers and avoid contract termination from existing customers (churn rate). For the telco company to expand its clientele, its growth rate (number of new customers) must exceed its churn rate (number of customers existing). Some of the factors that caused existing customers to leave their telco companies are better price offers, faster internet services, and a more secure online experience from other companies.

A high churn rate will adversely affect a company's profits and impede growth. Our churn prediction would be able to provide clarity to the telco company on how well it is retaining its existing customers and understand what are the underlying reasons that are causing existing customers to terminate their contract (high churn rate).

The telco company can use our analysis to measure if it is providing a useful product compared with the product provided by its competitors. Since the cost of acquiring new customers is much higher than retaining its existing customers, the company can use the churn rate analysis to provide discounts, special offers, and superior products to keep current customers.

1.2 Data Source

The telco company's data set is available **on Kaggle**, which stems from the IBM sample set collection. The company provides home and internet services to 7043 customers in California. Our challenge is to help the company predict behavior to retain customers and analyze all relevant customer data to develop focused customer retention programs.

The provided dataset consists of the information below:

1. Demographic information about customers including gender, age, marital status
2. Customer account information including the number of months staying with the company, paperless billing, payment method, monthly charges, and total charges
3. Customer usage behavior, such as streaming TV, streaming movie

Open in app

5. Customer churn where the customer left within the last month

1.3 Research Objectives

1. Which is the most important factor that contributes to the high retention rate?
2. Which analytics model can accurately predict a customer's churn rate?
3. What are the advantages and disadvantages of using different analytical models?
4. How could the telco company use our analysis to develop focused retention programs?

1.4 Justification of the Research

Our churn analysis is important for the telco company to understand why the customer has stopped using its product or service. Unless the company understands what is the total loss of revenue caused by customer's cancellations, which customers are canceling, and why they are canceling, it is hard for the telco company to improve its product and service.

Since churn rate analysis is a typical classification problem within the domain of supervised learning, we will be using Simple Linear Regression, Binomial Logit Regression, Binomial Probit Regression, and Random Forest Regression to analyze customer's churn behavior.

Our research will help the company provide insight on how to reduce customer churn by targeting specific customer's demographic information, account information, usage behavior, and signed-up services.

Section 2 — Data Summary and Exploratory Analysis

The data that we used to analyze is the secondary data that is available on Kaggle, an open-source data aggregation platform. A portion of the data is attached in figure 1.

```
<class 'pandas.core.frame.DataFrame'>
```

Open in app

```
---   ------            --------------   -----
 0    customerID        7043 non-null    int64
 1    gender            7043 non-null    object
 2    SeniorCitizen     7043 non-null    int64
 3    Partner           7043 non-null    object
 4    Dependents        7043 non-null    object
 5    tenure            7043 non-null    int64
 6    PhoneService      7043 non-null    object
 7    MultipleLines     7043 non-null    object
 8    InternetService   7043 non-null    object
 9    OnlineSecurity    7043 non-null    object
 10   OnlineBackup      7043 non-null    object
 11   DeviceProtection  7043 non-null    object
 12   TechSupport       7043 non-null    object
 13   StreamingTV       7043 non-null    object
 14   StreamingMovies   7043 non-null    object
 15   Contract          7043 non-null    object
 16   PaperlessBilling  7043 non-null    object
 17   PaymentMethod     7043 non-null    object
 18   MonthlyCharges    7043 non-null    float64
 19   TotalCharges      7043 non-null    object
 20   Churn             7043 non-null    object
dtypes: float64(1), int64(3), object(17)
memory usage: 1.1+ MB
```

Figure 1: Data Source Table, Image by Author

2.1 Data Introduction:

After reading the data using Pandas in Python, we found that there was no missing data from the raw data set and most of the features such as gender, phone service, all the way up to payment method were all categorical data.

Monthly Charges and TotalCharges are both numerical data. The summary statistics for MonthlyCharges is as follows:

Open in app

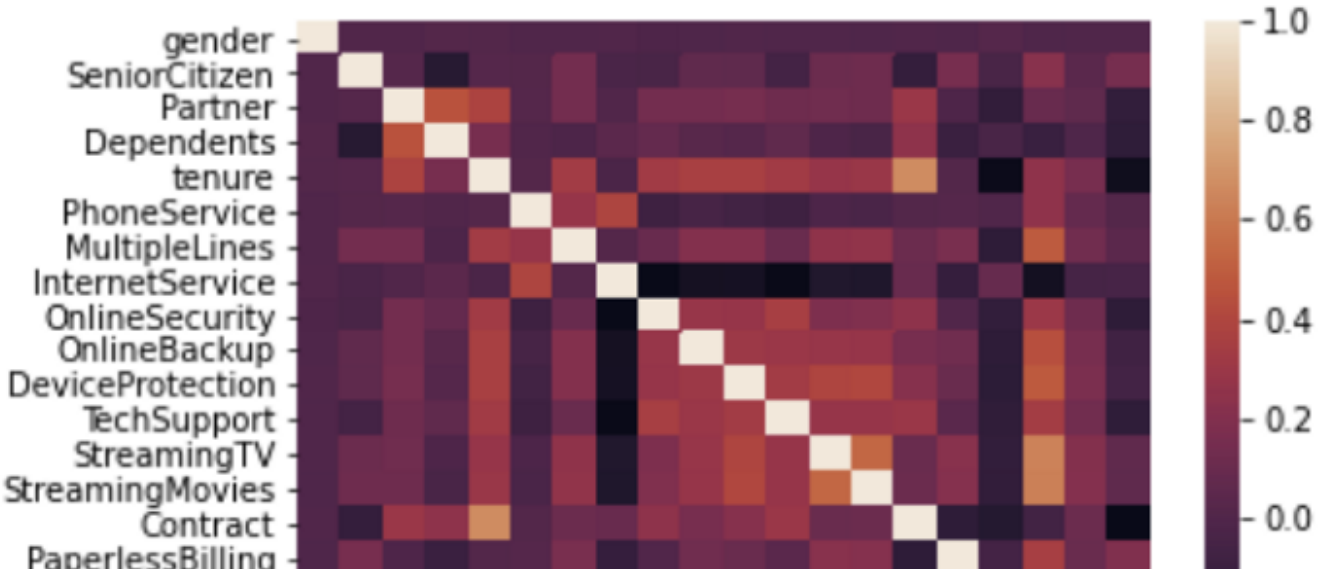
	MonthlyCharges
count	7043.000000
mean	64.761692
std	30.090047
min	18.250000
25%	35.500000
50%	70.350000
75%	89.850000
max	118.750000

Figure 2: Monthly Charges, Image by Author

On average, people who pay for the services $64.76 and the most expensive charge monthly is $118.75. The cheapest monthly charge is $18.25.

2.2 Correlation:

After converting all of the categorical data using Label Encoding and encoder, we ran a pair-wise correlation for all of the features:



Open in app

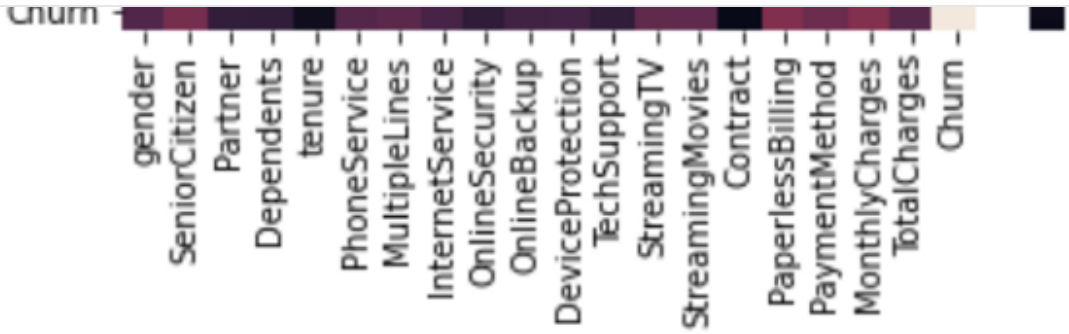


Figure 3: Heatmap: Correlation of Features, Image by Author

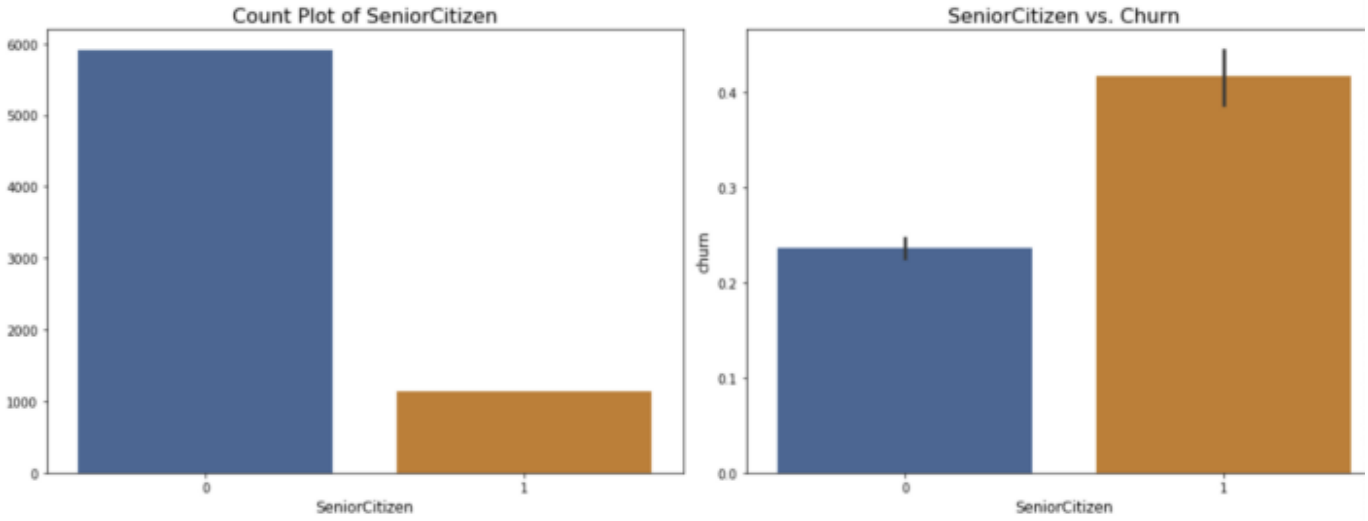
From the heatmap, we could see that the features 'Contract' and 'Tenure' have a high correlation. It makes sense because these features measure the loyalty of the customer.

"StreamingTV", "StreamingMovie", "Multiple Lines" and "Monthly Charges" have a high correlation with one another. We think that this is because customers who stream movies are more likely to stream TV as well. Their monthly charges tend to go up due to the large amount of data they use while watching movies or TV shows. For customers who have multiples on their account, they would be more likely to pay more than a customer who has only a single line.

2.3 Exploratory data analysis

2.31 Categorical Data Analysis:

- As in Figure 4: Senior Citizen, the customer who is a senior citizen will be more likely to churn for the Telco service.



Open in app

- As in Figure 5: Partner, the customer who doesn't have a partner will be more likely to churn for the Telco service.

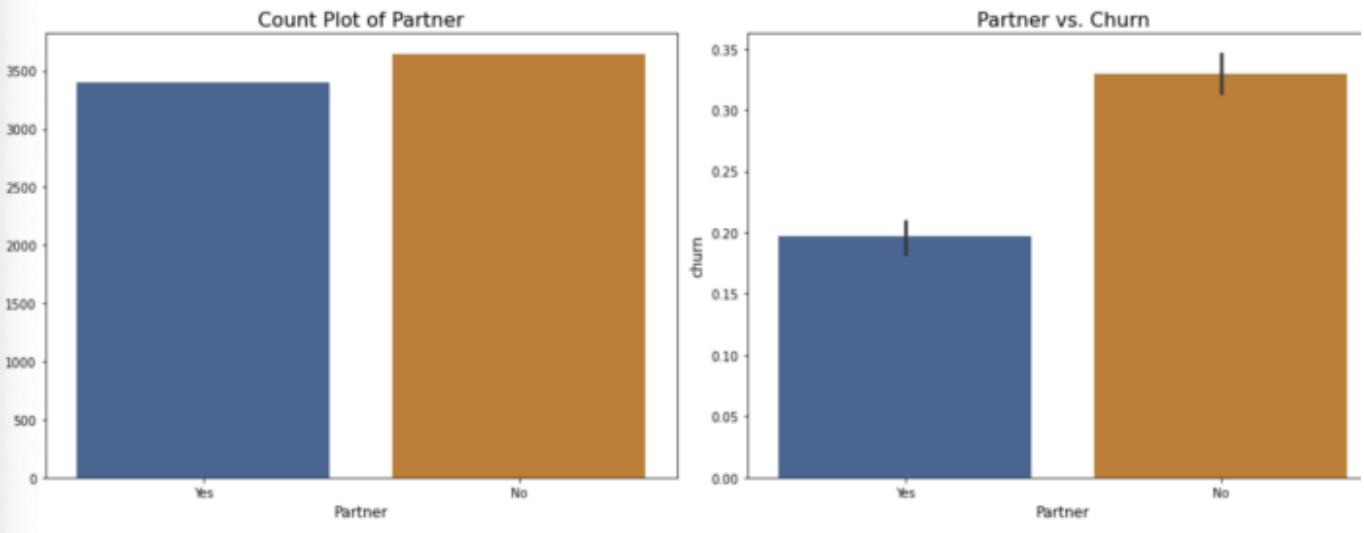


Figure 5: Partner, Image by Author

- As in Figure 6: Dependent, the customer who doesn't have Dependents will be more likely to churn for the Telco service.

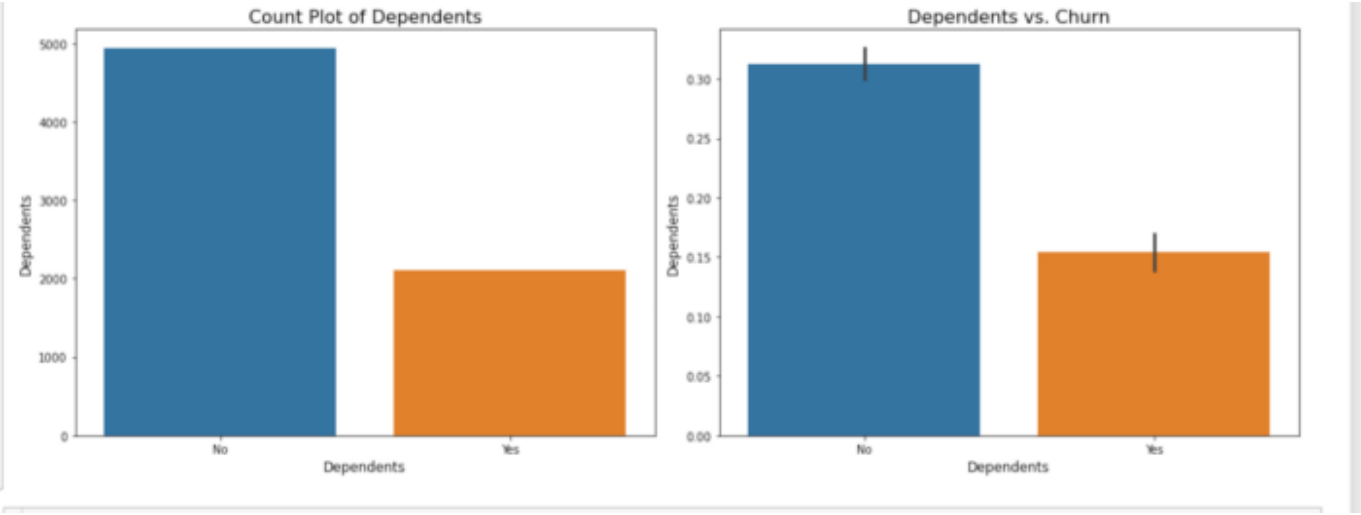
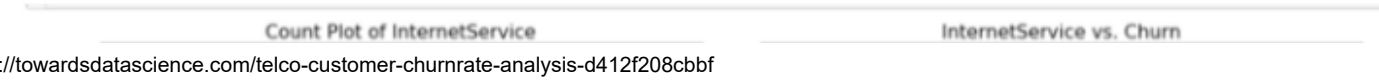


Figure 6: Dependents, Image by Author

- As in Figure 7: Internet Service, it looks like most people are using Fiber internet and the customers who subscribe to Fiber internet are more likely to Churn.



Open in app

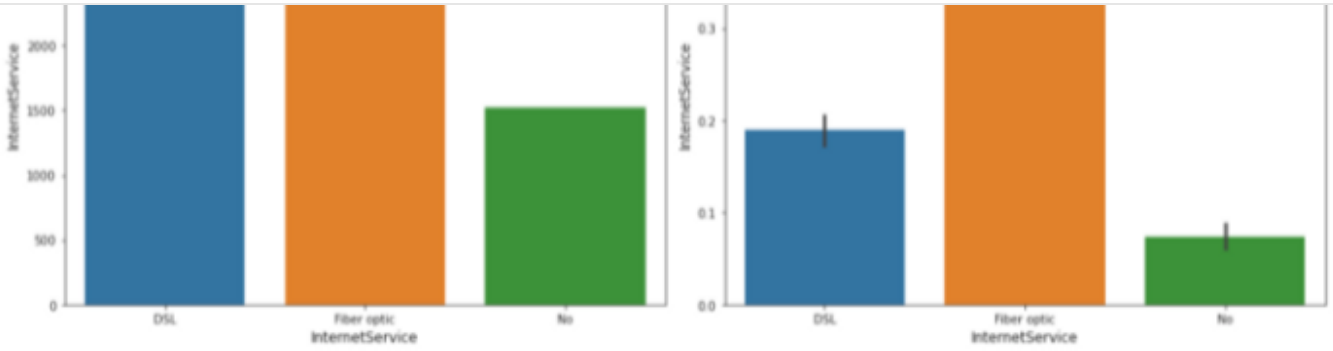


Figure 7: Internet Service, Image by Author

- As in Figure 7: Stream TV and Figure 8: Stream Movies, it is interesting to see that people who stream TV and stream Movies are more likely to churn. This could mean that the customers who stream TV and movies are not satisfied with the telco company's streaming service.

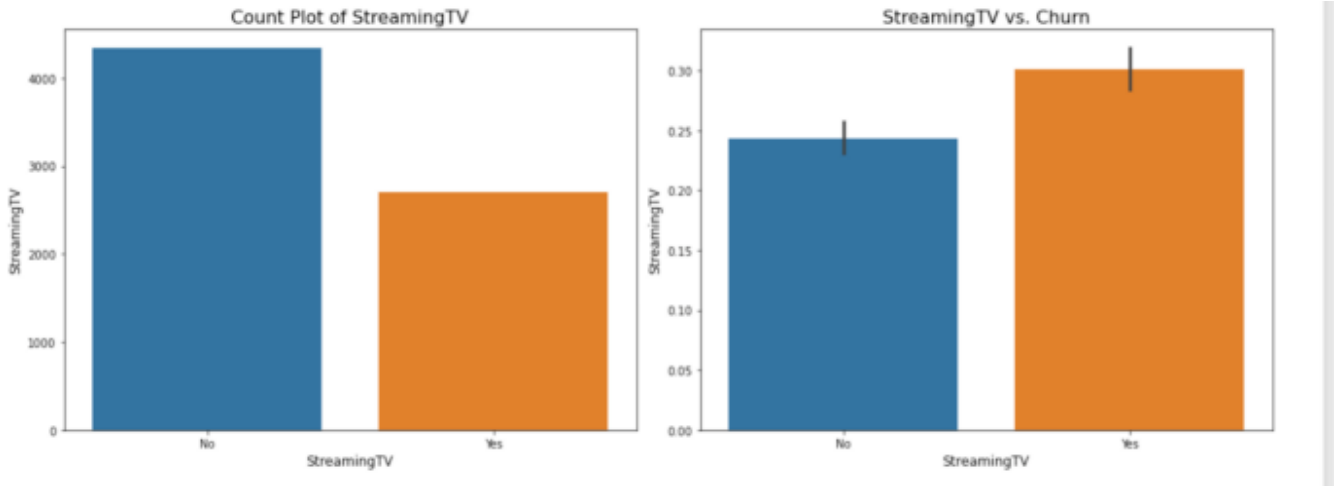
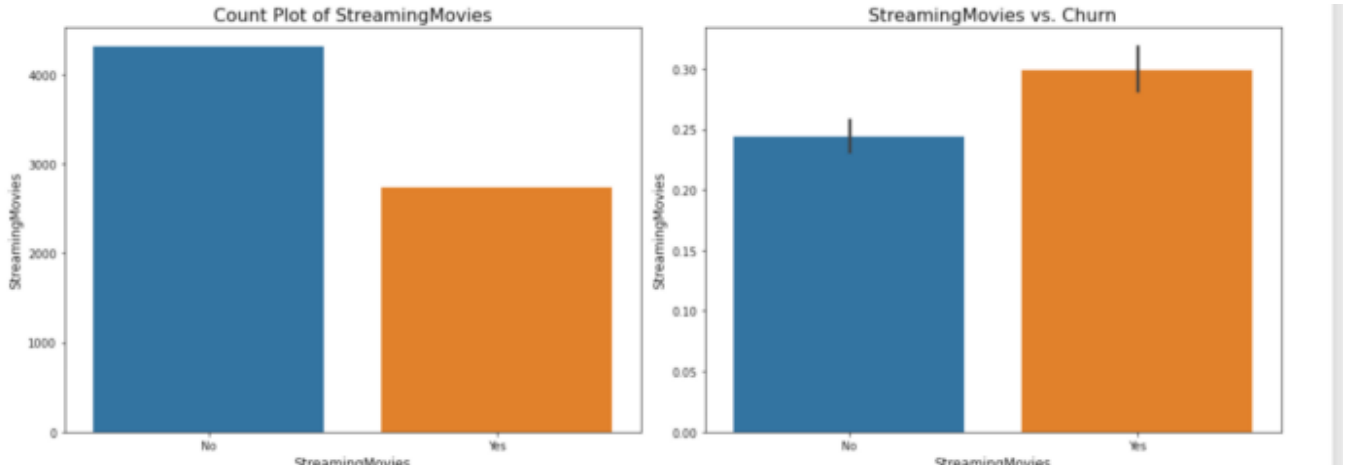


Figure 7: StreamTV, Image by Author



Open in app

2.32 Numerical Data Analysis:

- As in Figure 9: Monthly Charges, after binning the data into 6 bins for better visualization, we have an equal amount of customers in each bin. Customers who pay the monthly charges between $70.35 and $118.75 are more likely to churn compared with the people who pay less.

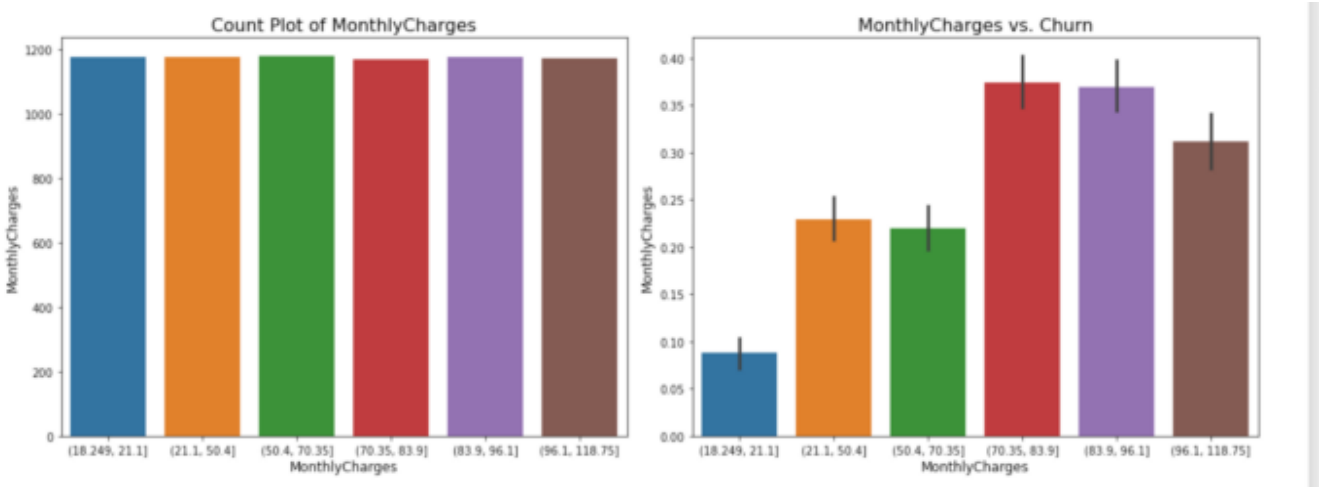
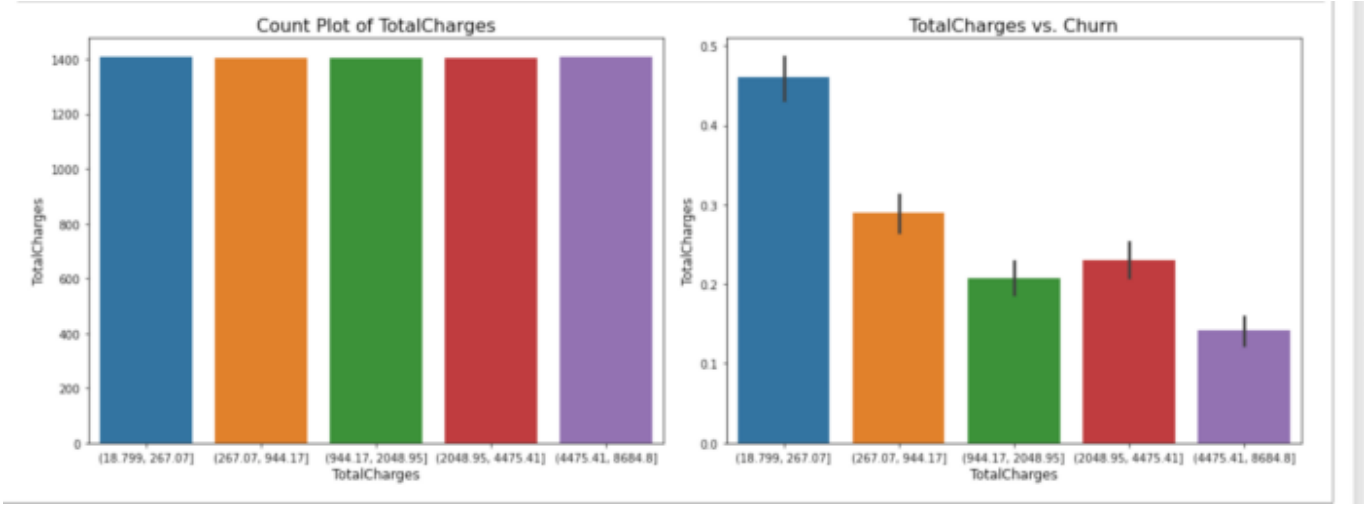


Figure 9: Monthly Charges, Image by Author

As in Figure 10: Monthly Charges, after binning the data into 5 bins for better visualization, we have an equal amount of customers in each bin. Customers who pay the monthly charges between $18.799 and $267.07 are most likely to churn compared with the people who pay more than that all the way up to $8684.8.



Open in app

Section 3 — Data Analyses, key findings, and conclusions

The four methods we have chosen for our data are (1) Simple Linear Regression; (2) Binomial Logit Regression; (3) Binomial Probit Regression; (4) Random Forest Regression.

3.1 Model Introduction

Let's start explaining our **first choice** of model: simple linear regression. A linear regression model predicts the target as a weighted sum of the feature inputs. The advantages and disadvantages of linear regression are mainly due to its simplicity and ease of use as our benchmark accuracy and reference.

```
Call:
lm(formula = Churn ~ gender + SeniorCitizen + Partner + Dependents +
    tenure + PhoneService + MultipleLines + InternetService +
    OnlineSecurity + OnlineBackup + DeviceProtection + TechSupport +
    StreamingTV + StreamingMovies + Contract + PaperlessBilling +
    PaymentMethod + MonthlyCharges + TotalCharges, data = ChurnData)
```

Figure 11: Simple Linear Model, Image by Author

Our Second choice of model: binomial logit regression is different from model 1. A binomial logistic regression is a sigmoid function where output is probability and input can be from -infinity to +infinity.

```
Call:
glm(formula = Churn ~ gender + SeniorCitizen + Partner + Dependents +
    tenure + PhoneService + MultipleLines + InternetService +
    OnlineSecurity + OnlineBackup + DeviceProtection + TechSupport +
    StreamingTV + StreamingMovies + Contract + PaperlessBilling +
    PaymentMethod + MonthlyCharges + TotalCharges, family = binomial(link = "logit"),
    data = ChurnData)
```

Figure 12: Binomial Logit Regression, Image by Author

For our specific dataset, the **pro side** includes (1) It is extremely useful for the binomial output dataset. (2) It is still fairly easy to interpret compared to advanced machine learning models. (3) It can be easily extended to multiple classes in the future. (4) It can interpret model coefficients as indicators of feature importance.

Open in app

Our **third choice** of model: binomial probit regression has plenty of similarities with our second probit model. However, they do have some differences. For the similarity side, they both perform quite well for the binomial dataset. And for the difference, in the probit model, it represents the cumulative normal pdf. Since the logistic model has slightly flatter tails, the probit curve approaches the axes more quickly than the logistic curve. Although we have to admit that the logistic model has easier interpretation than the probit model due to the fact that logistic regression can be interpreted as modeling log-odds, it is still not a bad idea to try both of them and then inspect afterward.

```
Call:
glm(formula = Churn ~ gender + SeniorCitizen + Partner + Dependents +
    tenure + PhoneService + MultipleLines + InternetService +
    OnlineSecurity + OnlineBackup + DeviceProtection + TechSupport +
    StreamingTV + StreamingMovies + Contract + PaperlessBilling +
    PaymentMethod + MonthlyCharges + TotalCharges, family = binomial(link = "probit"),
    data = ChurnData)
```

Figure 13: Binomial Probit Regression, Image by Author

Our fourth and last model is one of the fairly commonly used machine learning models: random forest. The random forest model consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction, and the class with the most votes becomes our model's prediction.

In our case, the pro side includes (1) It generally provides high accuracy and balances the bias-variance trade-off well. (2) It can be used as feature importance visualization. (3) It is not influenced by outliers to a fair degree. (4) It can handle both linear and nonlinear relationships. And the cons are (1) It is much harder to interpret compared to previous models. (2) It will take a much longer time if the dataset is huge.

3.2 Details of data analysis

Before we apply our dataset to the selected model, the very first step is always to transform and clean our data. In our case, there are several steps we need to make in order to fully reveal the power of our data.

Open in app

```
# delete NA
ChurnData = ChurnData[complete.cases(ChurnData), ]
```

DeleteNA.rmd hosted with ♡ by GitHub view raw

Code for Delete NA, Image by Author

Our data is from a realistic situation, which is highly likely to contain empty data cells. And it indeed has. Before the deletion, it had a total of 7043 observations.

1-10 of 7,043 rows | 1-9 of 21 columns

Figure 14: Before Deletion, Image by Author

And after the deletion, it has 7032 observations.

1-10 of 7,032 rows | 1-9 of 21 columns

Figure 15: After Deletion, Image by Author

Therefore, it only deleted 9 empty observations, which should only be an ignorable impact on our dataset.

3.22: Convert “No” to 0 and “Yes” to 1

A lot of our data columns’ default values are binary: they are either “ Yes” or “No”.

```
# Convert No to 0 and Yes to 1
ChurnData$Partner = as.numeric(as.factor(ChurnData$Partner)) - 1
ChurnData$Dependents = as.numeric(as.factor(ChurnData$Dependents)) - 1
ChurnData$PhoneService = as.numeric(as.factor(ChurnData$PhoneService)) - 1
ChurnData$MultipleLines = as.numeric(as.factor(ChurnData$MultipleLines)) - 1
ChurnData$OnlineSecurity = as.numeric(as.factor(ChurnData$OnlineSecurity)) - 1
ChurnData$OnlineBackup = as.numeric(as.factor(ChurnData$OnlineBackup)) - 1
ChurnData$DeviceProtection = as.numeric(as.factor(ChurnData$DeviceProtection)) - 1
ChurnData$TechSupport = as.numeric(as.factor(ChurnData$TechSupport)) - 1
ChurnData$StreamingTV = as.numeric(as.factor(ChurnData$StreamingTV)) - 1
ChurnData$StreamingMovies = as.numeric(as.factor(ChurnData$StreamingMovies)) - 1
ChurnData$PaperlessBilling = as.numeric(as.factor(ChurnData$PaperlessBilling)) - 1
```

Open in app

Code for Convert Binary, Image by Author

customerID <int>	gender <fctr>	SeniorCitizen <int>	Partner <fctr>	Dependents <fctr>	tenure <int>	PhoneService <fctr>	MultipleLines <fctr>	InternetService <fctr>
1	Female	0	Yes	No	1	No	No	DSL
2	Male	0	No	No	34	Yes	No	DSL
3	Male	0	No	No	2	Yes	No	DSL
4	Male	0	No	No	45	No	No	DSL
5	Female	0	No	No	2	Yes	No	Fiber optic
6	Female	0	No	No	8	Yes	Yes	Fiber optic
7	Male	0	No	Yes	22	Yes	Yes	Fiber optic
8	Female	0	No	No	10	No	No	DSL
9	Female	0	Yes	No	28	Yes	Yes	Fiber optic
10	Male	0	No	Yes	62	Yes	No	DSL

1-10 of 7,043 rows | 1-9 of 21 columns Previous 1 2 3 4 5 6 ... 100 Next

Figure 16: Default Values, Image by Author

For better data manipulation, it is best for us to convert them to 0 and 1.

3.23: Factor Conversion

```
# InternetService/Contract/paymentmethod  into factor
ChurnData$InternetService = as.factor(ChurnData$InternetService)
ChurnData$Contract = as.factor(ChurnData$Contract)
ChurnData$PaymentMethod = as.factor(ChurnData$PaymentMethod)
```

FactorConversion.rmd hosted with ♡ by GitHub view raw

Code for Factor Conversion, Image by Author

	customerID <int>	gender <dbl>	SeniorCitizen <int>	Partner <dbl>	Dependents <dbl>	tenure <int>	PhoneService <dbl>	MultipleLines <dbl>
1	1	0	0	1	0	1	0	0
2	2	1	0	0	0	34	1	0
3	3	1	0	0	0	2	1	0
4	4	1	0	0	0	45	0	0
5	5	0	0	0	0	2	1	0
6	6	0	0	0	0	8	1	1
7	7	1	0	0	1	22	1	1
8	8	0	0	0	0	10	0	0
9	9	0	0	1	0	28	1	1
10	10	1	0	0	1	62	1	0

1-10 of 7,032 rows | 1-9 of 21 columns Previous 1 2 3 4 5 6 ... 100 Next

Figure 17: After Conversion, Image by Author

There are three columns of features that should be coded into factors:

Open in app

3. Payment Method which includes Bank transfer, Credit Card, Electronic Check, and Mailed check.

After these three steps, we can now fit our data into our selected models.

3.3 Key results, Findings, and Interpretations

```
ChurnData.lm = lm(Churn~gender+SeniorCitizen+Partner+Dependents+tenure+PhoneService+MultipleLines+

ChurnData.glm1 = glm(Churn~gender+SeniorCitizen+Partner+Dependents+tenure+PhoneService+MultipleLi

ChurnData.glm2 = glm(Churn~gender+SeniorCitizen+Partner+Dependents+tenure+PhoneService+MultipleLi

summary(ChurnData.lm)
summary(ChurnData.glm1)
summary(ChurnData.glm2)
```

CreateModel.rmd hosted with ♡ by GitHub view raw

Code for Creating Models, Image by Author

3.31 Simple Regression Model

Let's run all the four models selected above one by one. The first one is a simple linear regression model.

```
Call:
lm(formula = Churn ~ gender + SeniorCitizen + Partner + Dependents +
    tenure + PhoneService + MultipleLines + InternetService +
    OnlineSecurity + OnlineBackup + DeviceProtection + TechSupport +
    StreamingTV + StreamingMovies + Contract + PaperlessBilling +
    PaymentMethod + MonthlyCharges + TotalCharges, data = ChurnData)

Residuals:
     Min       1Q   Median       3Q      Max
-0.81383 -0.25921 -0.05863  0.27684  1.12726

Coefficients:
                                        Estimate Std. Error t value Pr(>|t|)
(Intercept)                            4.133e-01  1.120e-01   3.690 0.000226 ***
gender                                -3.356e-03  8.940e-03  -0.375 0.707401
SeniorCitizen                          4.445e-02  1.300e-02   3.419 0.000631 ***
```

Open in app

```
MultipleLines                          5.866e-02  2.441e-02   2.403 0.016279 *
InternetServiceFiber optic             2.104e-01  1.096e-01   1.920 0.054902 .
InternetServiceNo                     -1.795e-01  1.107e-01  -1.621 0.105047
OnlineSecurity                        -4.251e-02  2.486e-02  -1.710 0.087357 .
OnlineBackup                          -1.130e-02  2.449e-02  -0.462 0.644434
DeviceProtection                       4.587e-03  2.474e-02   0.185 0.852888
TechSupport                           -4.392e-02  2.504e-02  -1.754 0.079442 .
StreamingTV                            6.378e-02  4.504e-02   1.416 0.156798
StreamingMovies                        6.576e-02  4.503e-02   1.460 0.144227
ContractOne year                      -1.056e-01  1.399e-02  -7.549 4.94e-14 ***
ContractTwo year                      -7.001e-02  1.704e-02  -4.110 4.01e-05 ***
PaperlessBilling                       4.491e-02  9.990e-03   4.495 7.06e-06 ***
PaymentMethodCredit card (automatic)  -6.070e-03  1.355e-02  -0.448 0.654191
PaymentMethodElectronic check          6.756e-02  1.328e-02   5.086 3.76e-07 ***
PaymentMethodMailed check             -6.746e-03  1.450e-02  -0.465 0.641804
MonthlyCharges                        -1.322e-03  4.368e-03  -0.303 0.762188
TotalCharges                          -4.438e-05  6.477e-06  -6.852 7.92e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3744 on 7008 degrees of freedom
Multiple R-squared:  0.2841,    Adjusted R-squared:  0.2817
F-statistic: 120.9 on 23 and 7008 DF,  p-value: < 2.2e-16
```

Figure 18: Simple Linear Regression Result, Image by Author

From the result in Figure 18: Simple Linear Regression, after carefully check the significance of each variable and only keep those that have a p-value smaller than 0.05, our resulting equation would be:

$$Churn = 0.413 + 0.0445 * Senior\ Citizen - 0.002 * Tenure + 0.059 * Multiple\ Line$$
$$- 0.10 * Contract\ one\ year - 0.07 * Contract\ two\ year + 0.045 * Paperless\ Billing$$
$$+ 0.068 * Payment\ Method\ Electronic\ Check - 4.43 * 10^{-5} * Total\ Charges$$

Figure 19: Simple Linear Regression with Coefficients, Image by Author

This equation shows us that if the customer is a senior citizen, uses multiple lines, uses paperless billing, uses electronic check as the payment method, he or she is more likely to churn.

Whereas if the customer stays with the company longer (tenure), chooses one year or two-year contract instead of month to month option, and the company offers a cheaper price, he or she is less likely to churn.

Open in app

Our second model is a binomial logistic regression model:

```
Call:
glm(formula = Churn ~ gender + SeniorCitizen + Partner + Dependents +
    tenure + PhoneService + MultipleLines + InternetService +
    OnlineSecurity + OnlineBackup + DeviceProtection + TechSupport +
    StreamingTV + StreamingMovies + Contract + PaperlessBilling +
    PaymentMethod + MonthlyCharges + TotalCharges, family = binomial(link = "logit"),
    data = ChurnData)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9180  -0.6791  -0.2855   0.7282   3.4300

Coefficients:
                                         Estimate Std. Error z value Pr(>|z|)
(Intercept)                             1.165e+00  8.151e-01   1.430  0.15284
gender                                 -2.183e-02  6.480e-02  -0.337  0.73619
SeniorCitizen                           2.168e-01  8.453e-02   2.564  0.01033 *
Partner                                -3.840e-04  7.783e-02  -0.005  0.99606
Dependents                             -1.485e-01  8.973e-02  -1.655  0.09796 .
tenure                                 -6.059e-02  6.236e-03  -9.716  < 2e-16 ***
PhoneService                            1.715e-01  6.487e-01   0.264  0.79153
MultipleLines                           4.484e-01  1.773e-01   2.530  0.01142 *
InternetServiceFiber optic              1.747e+00  7.981e-01   2.190  0.02855 *
InternetServiceNo                      -1.786e+00  8.073e-01  -2.213  0.02691 *
OnlineSecurity                         -2.054e-01  1.787e-01  -1.150  0.25031
OnlineBackup                            2.604e-02  1.754e-01   0.148  0.88197
DeviceProtection                        1.474e-01  1.764e-01   0.836  0.40339
TechSupport                            -1.805e-01  1.806e-01  -0.999  0.31759
StreamingTV                             5.905e-01  3.263e-01   1.810  0.07035 .
StreamingMovies                         5.993e-01  3.267e-01   1.834  0.06658 .
ContractOne year                       -6.608e-01  1.076e-01  -6.142 8.15e-10 ***
ContractTwo year                       -1.357e+00  1.764e-01  -7.691 1.46e-14 ***
PaperlessBilling                        3.424e-01  7.450e-02   4.596 4.31e-06 ***
PaymentMethodCredit card (automatic)   -8.779e-02  1.141e-01  -0.770  0.44156
PaymentMethodElectronic check           3.045e-01  9.450e-02   3.222  0.00127 **
PaymentMethodMailed check              -5.759e-02  1.149e-01  -0.501  0.61627
MonthlyCharges                         -4.034e-02  3.176e-02  -1.270  0.20392
TotalCharges                            3.289e-04  7.063e-05   4.657 3.20e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 8143.4  on 7031  degrees of freedom
Residual deviance: 5826.3  on 7008  degrees of freedom
AIC: 5874.3

Number of Fisher Scoring iterations: 6
```

Figure 20: Binomial Logistic Regression Result, Image by Author

From the result in Figure 20: Logistic Regression Model, after carefully check the significance of each variable and only keep these that has p-value smaller than 0.05, our resulting equation would be:

Open in app

$$- 0.661 * Contract\ one\ year - 1.357 * Contract\ two\ year + 0.342 * Paperless\ Billing$$
$$+ 0.305 * Payment\ Method\ Electronic\ Check - 3.289 * 10^{-4} * Total\ Charges$$

Figure 21: Binomial Logistic Regression with Coefficients, Image by Author

This equation shows us that if the customer is a senior citizen, uses multiple lines, uses fiber optic internet service, uses paperless billing, uses electronic check as the payment method, he or she is more likely to churn.

Whereas if the customer stays with the company longer (tenure), chooses no internet service at all, chooses one year or two-year contract instead of month to month option, and the company offers a cheaper price, he or she is less likely to churn.

3.33 Probit Regression Model

Our third model is a binomial probit regression model.

```
Call:
glm(formula = Churn ~ gender + SeniorCitizen + Partner + Dependents +
    tenure + PhoneService + MultipleLines + InternetService +
    OnlineSecurity + OnlineBackup + DeviceProtection + TechSupport +
    StreamingTV + StreamingMovies + Contract + PaperlessBilling +
    PaymentMethod + MonthlyCharges + TotalCharges, family = binomial(link = "probit"),
    data = ChurnData)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9169  -0.6992  -0.2903   0.7403   3.6279

Coefficients:
                                 Estimate Std. Error z value Pr(>|z|)
(Intercept)                     5.849e-01  4.699e-01   1.245 0.213202
gender                         -9.608e-03  3.751e-02  -0.256 0.797846
SeniorCitizen                   1.330e-01  4.979e-02   2.671 0.007556 **
Partner                        -6.190e-03  4.497e-02  -0.138 0.890529
Dependents                     -8.769e-02  5.106e-02  -1.717 0.085934 .
tenure                         -2.779e-02  3.192e-03  -8.707  < 2e-16 ***
PhoneService                    1.557e-01  3.738e-01   0.416 0.677147
MultipleLines                   2.709e-01  1.024e-01   2.647 0.008126 **
InternetServiceFiber optic      1.050e+00  4.599e-01   2.283 0.022418 *
InternetServiceNo              -1.054e+00  4.648e-01  -2.269 0.023283 *
OnlineSecurity                 -1.049e-01  1.030e-01  -1.019 0.308429
OnlineBackup                    2.748e-02  1.014e-01   0.271 0.786301
DeviceProtection                8.972e-02  1.020e-01   0.879 0.379165
TechSupport                    -8.952e-02  1.040e-01  -0.861 0.389438
StreamingTV                     3.581e-01  1.882e-01   1.903 0.057024 .
StreamingMovies                 3.712e-01  1.883e-01   1.971 0.048719 *
ContractOne year               -3.708e-01  5.936e-02  -6.247 4.19e-10 ***
ContractTwo year               -6.486e-01  8.610e-02  -7.533 4.97e-14 ***
```

Open in app

```
MonthlyCharges                           -2.370e-02   1.650e-02  -1.435 0.151314
TotalCharges                              1.059e-04   3.686e-05   2.872 0.004081 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 8143.4  on 7031  degrees of freedom
Residual deviance: 5851.1  on 7008  degrees of freedom
AIC: 5899.1

Number of Fisher Scoring iterations: 6
```

Figure 22: Binomial Probit Regression Result, Image by Author

From the result in Figure 22: Probit Regression Model, after carefully check the significance of each variable and only keep those that have a p-value smaller than 0.05, our resulting equation would be:

$$\begin{aligned} Churn &= 0.113 * Senior\ Citizen - 0.028 * Tenure + 0.271 * Multiple\ Line \\ &+ 1.050 * Internet\ Service\ Fiber\ Optic - 1.054 * Internet\ Service\ No + 0.371 * Steam\ Movies \\ &- 0.371 * Contract\ one\ year - 0.649 * Contract\ two\ year + 0.188 * Paperless\ Billing \\ &+ 0.191 * Payment\ Method\ Electronic\ Check - 1.059 * 10^{-4} * Total\ Charges \end{aligned}$$

Figure 23: Binomial Probit Regression with Coefficients, Image by Author

This equation shows us that if the customer is a senior citizen, uses multiple lines, uses fiber optic internet service, uses streaming movies, uses paperless billing, and uses electronic check as the payment method, he or she is more likely to churn.

Whereas if the customer stays with the company longer (tenure), chooses no internet service at all, chooses one year or two-year contract instead of month to month option, and the company offers a cheaper price, he or she is less likely to churn.

3.34 Random Forest Model

Our fourth model is random forest. The specific code is listed below for Random Forest.

```
library(randomForest)
rf_train= train
rf_test = test
rf_train$Churn = as.factor(rf_train$Churn)
```

Open in app

```
rf.prediction = predict(rf, rf_test)

# for rf
rf.d_binomial <- tibble("target" = truth,"prediction" = rf.prediction)
rf.basic_table <- table(rf.d_binomial)


rf.accuracy = (rf.basic_table[1,1] + rf.basic_table[2,2])/sum(rf.basic_table)
rf.accuracy

varImpPlot(rf)

importance(rf)

# error
plot(rf)


# plot a sample tree
library("party")
x <- ctree(Churn ~ ., data=rf_train)
plot(x, type="simple")
```

rf.rmd hosted with ♡ by GitHub view raw

Code for Random Forest, Image by Author

One sample tree among the forest has the shape below:

Open in app

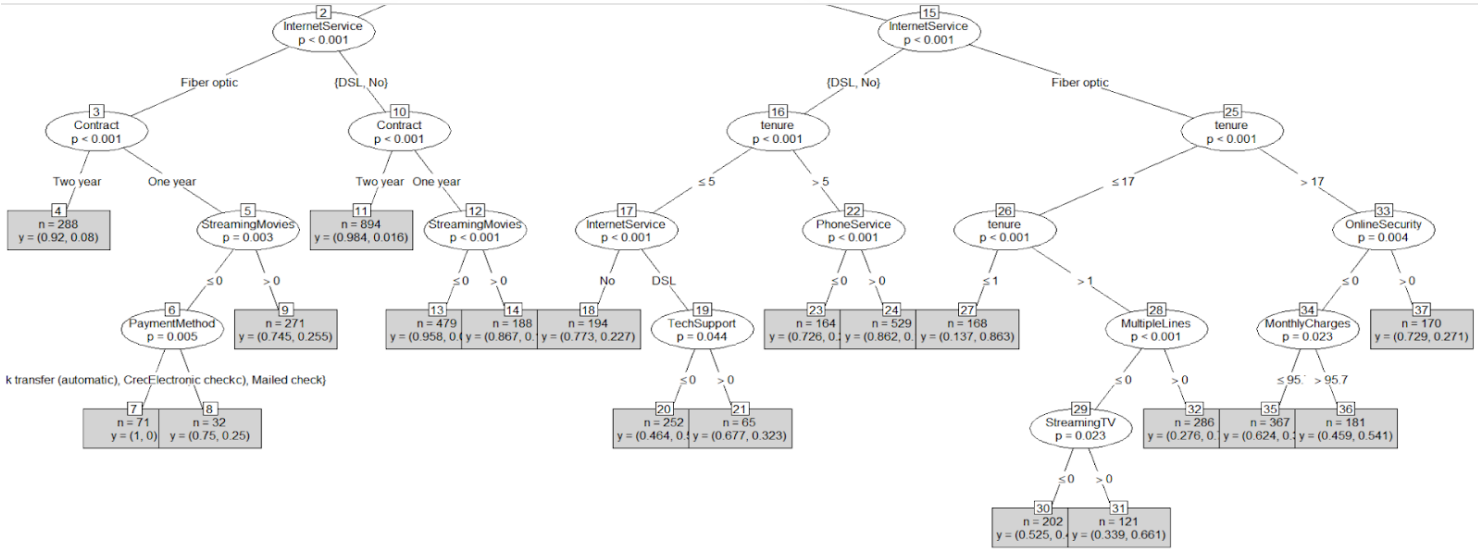
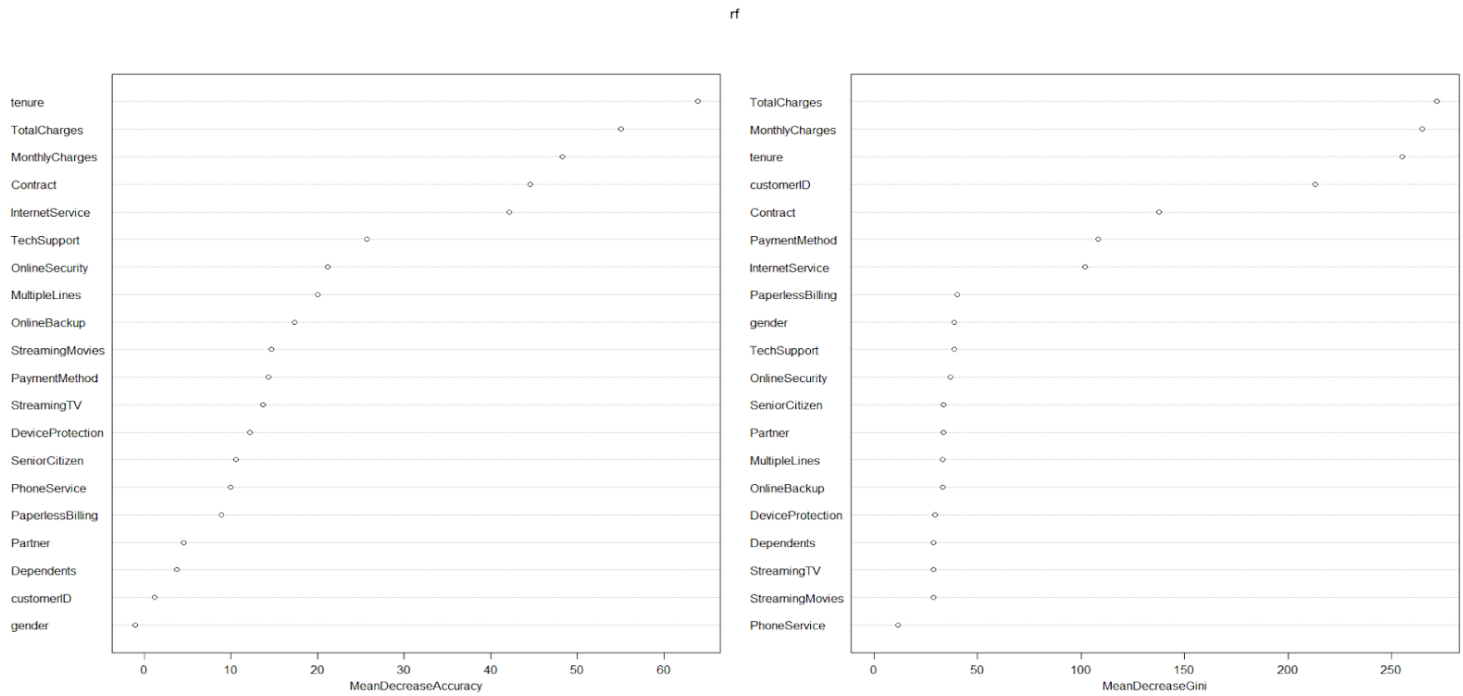


Figure 25: Sample Tree, Image by Author

It's indeed a very crowded tree due to the fact that we have plenty of features for the split. Every node represents a decision or criteria made by the tree and each branch represents the relationship between a parent node and child node. Eventually, all the results will be assigned to the last level of the node.

And from the special feature of random forest, we also have the option to inspect the importance of the features:



Open in app

From Figure 23: Random Forest Feature Importance in the appendix, the importance is calculated by the mean decrease accuracy: how much the model accuracy decreases if we drop that variable. On the right side, the table uses the mean decrease Gini to measure: the measure of variable importance based on the Gini Impurity index used for the calculation of the splits in the tree.

For both tables, the top ones are Tenure, Total Charges, Monthly Charges, Contract, Internet Service. The results agree with our previous' models' significant results.

```
glm1.prediction = predict(ChurnData.glm1, test)
glm1.prediction = ifelse(glm1.prediction > 0,1,0)
glm2.prediction = predict(ChurnData.glm2, test)
glm2.prediction = ifelse(glm2.prediction > 0,1,0)

glm1.prediction = matrix(glm1.prediction)
glm2.prediction = matrix(glm2.prediction)
truth = matrix(test$Churn)

# for glm1
glm1.d_binomial <- tibble("target" = truth,"prediction" = glm1.prediction)
glm1.basic_table <- table(glm1.d_binomial)

# for glm2
glm2.d_binomial <- tibble("target" = truth,"prediction" = glm2.prediction)
glm2.basic_table <- table(glm2.d_binomial)


glm1.accuracy = (glm1.basic_table[1,1] + glm1.basic_table[2,2])/sum(glm1.basic_table)
glm2.accuracy = (glm2.basic_table[1,1] + glm2.basic_table[2,2])/sum(glm2.basic_table)


cat(glm1.accuracy,glm2.accuracy)
```

Accuracy.rmd hosted with ♡ by GitHub view raw

Code for Accuracy Comparision, Image by Author

3.4 Accuracy Comparison and Exploration

Open in app

result in the AIC/BIC table below. Please note that AIC and BIC are not for the random forest, we will compare them later with a selected high accuracy model.

3.41 AIC/BIC Table

Model	AIC	BIC
Linear Regression	6165.736	6337.192
Binomial Logit	5874.272	6038.87
Binomial Probit	5899.066	6063.664

Figure 26: AIC/BIC Table, Image by Author

From the AIC/BIC table above, since the lower the AIC/BIC, the better the model, our best model would be the binomial logistic regression model. Let's continue to compare it with the random forest model by calculating the predicting accuracy.

3.42 Accuracy Table

Model	Accuracy
Binomial Logit	0.7957346
Random Forest	0.7947867

Figure 27: Accuracy Table, Image by Author

From the accuracy table above, we could see that although their accuracy is similar, the binomial logistic regression model has slightly better performance.

Since the binomial logistic regression model is the best one, let's explore the details of its accuracy. We can use the ROC curve and confusion matrix to inspect more about its performance.

Open in app

```r
# Accuracy
library(ROCR)
library(Metrics)

# AUC curve
pr <- prediction(glm1.prediction,test$Churn)
perf <- performance(pr,measure = "tpr",x.measure = "fpr")
plot(perf) > auc(test$Churn,glm1.prediction)
```

ConfusionMatrix.rmd hosted with ♡ by GitHub view raw

Code for ROC Curve, Image by Author

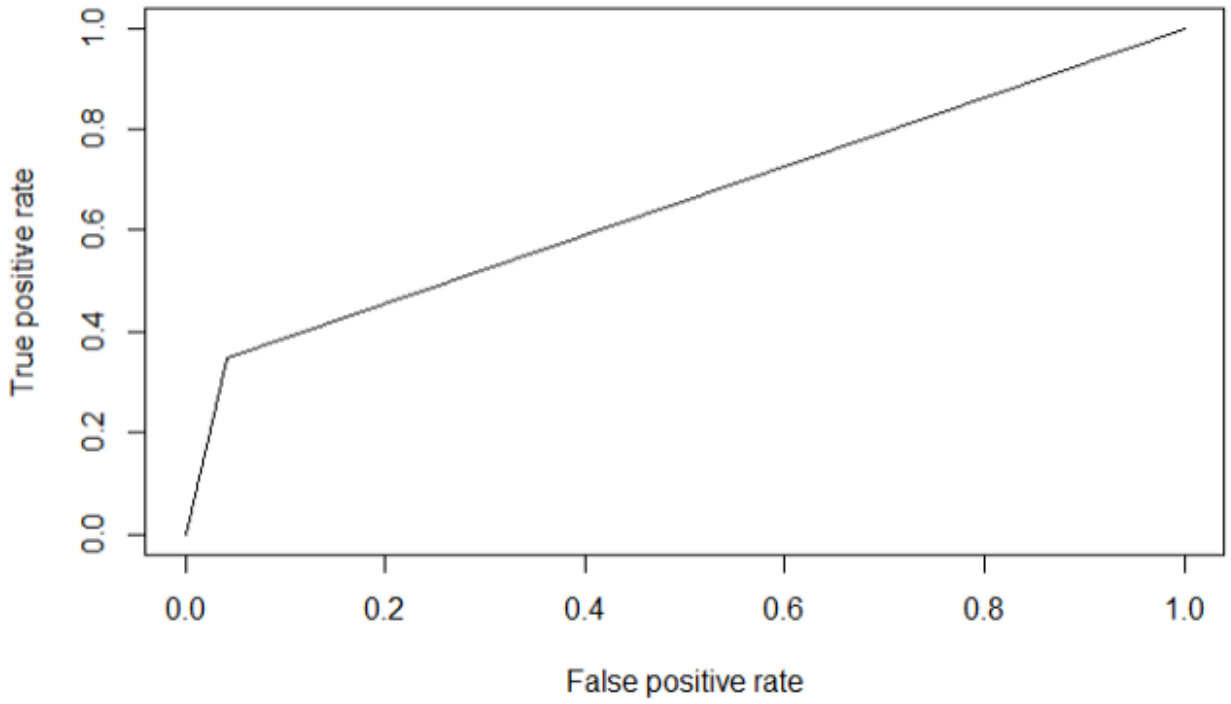


Figure 28: ROC Plot, Image by Author

In Figure 28: ROC Plot, ROC determines the accuracy of a classification model at a user-defined threshold value. It determines the model's accuracy using Area Under Curve (AUC). The higher the area, the better the model. ROC is plotted between True Positive Rate (Y-axis) and False Positive Rate (X-Axis).

Open in app

sections.

3.44 Confusion Matrix

```
# Confusion Matrix with plot


d_binomial <- tibble("target" = truth,"prediction" = glm1.prediction)
basic_table <- table(d_binomial)
cfm <- tidy(basic_table)

plot_confusion_matrix(cfm,
                      target_col = "target",
                      prediction_col = "prediction",
                      counts_col = "n")
```

gistfile1.txt hosted with ♡ by GitHub view raw

Code for Confusion Matrix, Image by Author

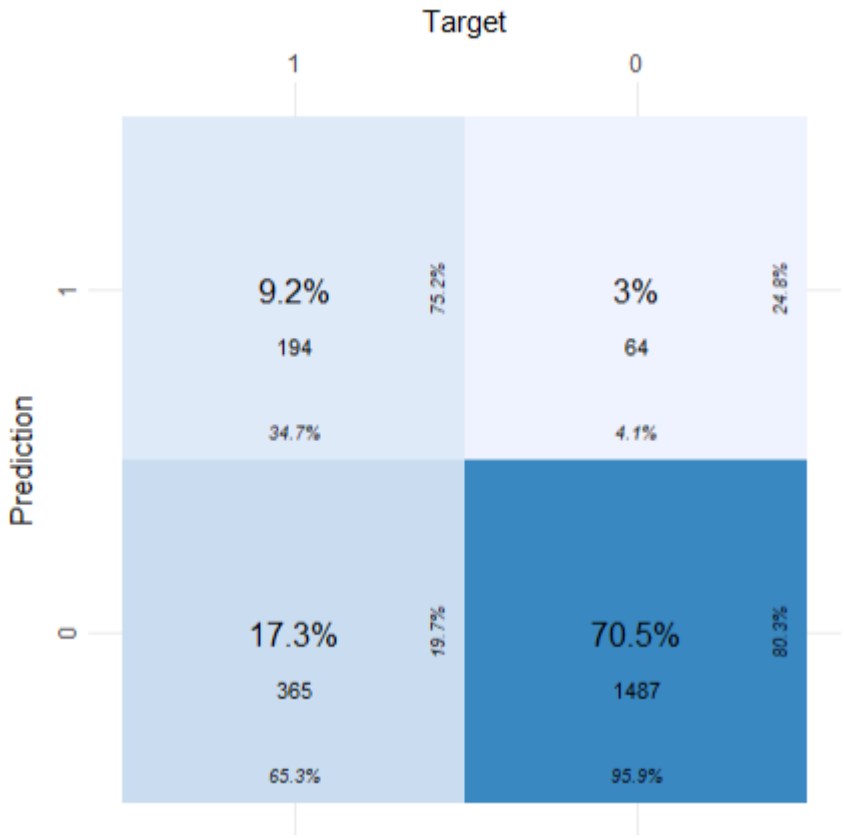


Figure 29: Confusion Matrix, Image by Author

Open in app

1. In the middle of each tile, we have the overall percentage of the count. And the actual count number is beneath it.
2. At the bottom, we have the column percentage. Of all the observations where Target is 1, 34.7% of them were predicted to be 1 and 65.3% to be 0.
3. At the right side of each tile, we have the row percentage. Of all the observations where Prediction is 1, 4.1% of them were actually 1, while 95.9% were 0.
4. The color intensity is based on the counts. The more counts, the deeper the color.

3.45 Important Calculations From Confusion Matrix

From the aggregated confusion matrix, we can calculate plenty of matrices.

- Accuracy — It determines the overall predicted accuracy of the model.

> *Accuracy = (True Positives + True Negatives)/(True Positives + True Negatives + False Positives + False Negatives)*

$$Accuracy = (194 + 1487)/2110 = 0.7957$$

- True Positive Rate (TPR) — It indicates how many positive values, out of all the positive values, have been correctly predicted. It also represents sensitivity.

> *TPR = (TP/TP + FN)*

$$TPR = (194)/(194 + 64) = 0.7519$$

- False Positive Rate (FPR) — It indicates how many negative values, out of all the negative values, have been incorrectly predicted.

> *FNR = (FP/FP + TN)*

$$FPR = (365)/(365 + 1487) = 0.1971$$

Open in app

> *TNR = (TN/TN + FP)*

$$TNR = (1487)/(1487 + 365) = 0.8029$$

- Precision: It indicates how many values, out of all the predicted positive values, are actually positive.

> *Precision = (TP / TP + FP)*

$$Precision = (194)/(194 + 365) = 0.3470$$

Section 4 — Strategy recommendations, limitations, and future research

4.1 Recommendations

From all the visualizations and analysis we have present in Section 3, we can thus make the following recommendations for the decision-maker:

1. It is best to choose binomial logistic regression for a similar question since the logit model has the highest accuracy and indicates the feature importance. It is also easy to interpret and apply.
2. For our current case, the best model logit model indicates that, if the firm wants to keep the customers, it can do the following measures:

- Target more on young and middle-aged customers since they are more likely to adopt modern technology and have the budget to enjoy.
- Offer more discount for the customers who decide to choose the one year or two-year contract so that more customers will be bound with the contract.
- Consider an overall discount since the price is always one of the major factors for customers to choose among existing incumbents.

Open in app

For the limitations of the research, we should mention the following limitations for both our model and our dataset.

1. The number of observations is decent, but if we could have more columns of features like the customers' geographic location, competitor's information, and other important factors, we could draw more insights from the result.
2. Since we have chosen our model not only depends on the complexity and predicting power but more importantly on the ease of interpretation, there are more powerful models outside of our range. For example, neural networks or extreme gradient boosting may perform much better and result in increased accuracy.
3. The nature of our dataset is a cross-sectional dataset. This means that there are no time series factors inside it. Since our goal is to predict churn rate, we have the option of contracts from monthly, one year to two years. It is best that we can find a time series dataset containing all the customer's information for up to two years to obtain better results for predicting and making decisions for the future market.

4.3 Next Step

1. We can potentially consider fit neural networks or extreme gradient boosting to obtain better accuracy. However, we do need to take the explanation difficulty into account since both of the methods are more like a black box rather than traditional clear regression.
2. Try to find a time series dataset containing all the customer's information for up to two years so that we can apply not only the models described above but also time series domain models like ARIMA models.

Sign up for The Variable

By Towards Data Science

Every Thursday, the Variable delivers the very best of Towards Data Science: from hands-on tutorials and cutting-edge research to original features you don't want to miss. Take a look.

Emails will be sent to mshch96@utexas.edu.

Open in app

Data Analysis Model

About Help Legal

Get the Medium app

Download on the App Store GET IT ON Google Play