Churn Prediction for Subscription-Based Services using Business Analytics techniques

Debasmita Ray
The University of Texas at Dallas
Master's in Business Analytics and Artificial Intelligence

INTRODUCTION

In today's competitive landscape, subscription-based businesses face a serious challenge: rising customer churn. With some companies experiencing monthly churn rates above 5%, this translates to a potential loss of over half their customer base annually. Studies show that reducing churn by just 5% can increase profits by up to 95%, making effective churn prediction essential. This project aims to develop a predictive model in R, analyzing historical customer data to identify those at risk of leaving. The resulting insights will support targeted retention strategies, helping businesses reduce churn and improve profitability.

OBJECTIVE

Our primary goal is to enhance customer retention for a subscription-based company by developing an effective churn prediction model. Our primary objectives are twofold: to accurately identify customers at risk of churning and to uncover the key factors driving this churn. The end goal is not just to reduce churn rates, but to significantly improve customer loyalty and lifetime value. In the competitive landscape of subscription services, this project represents a critical step towards sustainable growth and increased market share.

METHODOLOGY

We are using Decision Tree, Random Forest, and Logistic Regression for our project to predict churning of customers for subscription-based products. Decision Tree, Logistic Regression, and Random Forest models are commonly used for churn prediction due to their strengths in classification tasks. Decision Trees provide interpretable decision paths, helping to identify key churn indicators. Logistic Regression estimates the probability of churn, offering insights into feature influence, while Random Forest captures complex data relationships, enhances accuracy, and provides feature importance rankings. Together, these models offer robust, interpretable, and effective methods for predicting churn and informing customer retention strategies. By combining the interpretability of Decision Trees, the predictive power of Random Forests, and the statistical rigor of Logistic Regression, this methodology ensures a comprehensive approach to churn prediction. It provides actionable insights to reduce churn and enhance customer loyalty.

1. DATA COLLECTION:

We have taken Customer Subscription Data from Kaggle Dataset repository. This data is about a subscription-based digital product offering financial advisory that includes newsletters, webinars, and investment recommendations. The offering has a couple of varieties, annual subscription, and digital subscription. The product also provides daytime support for customers to reach out to a care team that can help them with any product-related questions and sign-up/cancellation-related queries.

The data set contains the following information:

- ***Customer Sign-up and Cancellation Dates:*** Detailed records of customer start and end dates at the product level, providing timing of cancellations.

- ***Subscription Details:*** Information about the subscription plan, payment frequency, and contract renewal status.
- ***Customer Demographics:*** Demographic data such as age, location, and potentially other customer attributes, offering context on the types of customers who are more likely to cancel.
- ***Product Pricing Information:*** Details on product pricing, which may correlate with churn likelihood

2. DATA PREPROCESSING:

Effective data preprocessing is essential for building a reliable and accurate churn prediction model. The preprocessing steps applied to this dataset include handling missing values, encoding categorical variables, and scaling numerical features. By transforming the data into a clean and structured format, we lay a solid foundation for accurate analysis and model training, for predictions in the churn prediction model.

Loading the data

The first step in data preprocessing is to load the dataset into the working environment. The chosen modeling platform for this project is RStudio.

R

```
library(readxl)
customer.df <- read_excel("Customer Data Project.xlsx")
```

Understanding the Data

Variables	Type	Description
Case_id	Nominal	A unique identifier for each case
Date_time	DateTime	Timestamp for the event
Customer_id	Nominal	Unique identifier for each customer
Channel	Nominal	Channel through which the interaction happened
Reason	Nominal	Reason for interaction
Age	Numeric	Age of the customer
Gender	Nominal	Gender of the customer

Product_id	Nominal	Product associated with the subscription
Signup_date_time	DateTime	Timestamps for subscription signup
Cancel_date_time	DateTime	Timestamps for subscription cancellation
Subscription	Nominal	Type of subscription
Price	Numeric	Subscription price
Billing_cycle	Numeric	Subscription billing cycle in months

Missing Data

Identifying and filling or removing missing data points using methods like mean/median imputation or deletion, depending on the nature and proportion of missing values is the next step towards a well-structured dataset.

R

```
# Check for missing values in the datasets
sum(is.na(customer.df))
```

Output:

```
[1] 0
```

There are no missing values in the dataset.

Encoding Categorical Variable

Categorical variables need to be converted into numerical representations. We are using **Binary encoding**, which is a method of transforming categorical data into numerical format, specifically into 0 and 1, as the categorical variable has only two distinct categories. For variables with more than two categories, we consider one-hot encoding instead, but the categories in our variables can be simplified meaningfully into 0 and 1. By converting each category into binary variables, it helps algorithms process data effectively while maintaining logical distinctions. Machine learning models, like Logistic Regression, Decision Trees, and Random Forests, require numerical input, hence this approach allows the model to capture each category's unique impact on churn behavior independently, thereby enhancing both the interpretability and predictive accuracy of the analysis.

- **Scaling Feature**

```
R
# Check for missing values in the datasets
customer.df$Channel <- ifelse(customer.df$Channel == "phone", 0, 1)
customer.df$Reason <- ifelse(customer.df$Reason == "signup", 0, 1)
customer.df$Gender <- ifelse(customer.df$Gender == "male", 0, 1)
customer.df$Subscription <- ifelse(customer.df$Subscription ==    "monthly_subscription", 0, 1)
```

Normalization is typically required when using distance-based models (KNN, SVM) or gradient-based optimization (Neural Networks) to ensure all features contribute equally regardless of scale. However, for this dataset, normalization was not necessary because the chosen models—Decision Tree, Random Forest, and Logistic Regression—are not sensitive to feature magnitudes. Additionally, binary encoding was applied to categorical variables, which inherently standardized them to a 0–1 range. Since the numerical variables did not exhibit extreme variations, normalization would add unnecessary complexity without improving model performance.

- **Feature Engineering**

1. The first three columns in the dataset (Case_id, Date_time, and Customer_id) were dropped because they serve as identifiers rather than features relevant for predicting churn. These columns do not provide meaningful information about customer behavior or subscription patterns.

2. Additionally, we removed redundant columns like Subscription since their information overlaps with the target variable Target_churn. Keeping only Target_churn ensures the model focuses on the primary outcome of interest while avoiding over-representation of similar information, which could distort predictions. This streamlining improves model interpretability and efficiency.

3. Created new variables that better capture customer behavior. For example: **Customer Tenure**: Calculate the number of years a customer has been subscribed. Use relevant features like age, gender, Subscription, price, billing_cycle, and possibly time-based features like tenure (time from signup_date_time to cancel_date_time or to the present if active).

- **Target Variable**

Define a target variable, churn, based on cancel_date_time. For instance, if cancel_date_time is not null, we can label the customer as “churned” (1); otherwise, label as “not churned” (0).

3. EXPLORATORY DATA ANALYSIS:

CHURN RATE ANALYSIS

Determine the proportion of customers who have churned. This provides a baseline understanding of the churn rate and helps in evaluating the model’s performance. From the sample data, we can define churn based on the cancel_date_time column: if this column has a value, it indicates that the customer has churned (canceled their subscription); if it’s NA (not a timestamp), the customer is still active. For a subscription-based business, churn rate could help us understand the issues potentially prompting a deeper analysis to identify the causes and develop retention strategies.

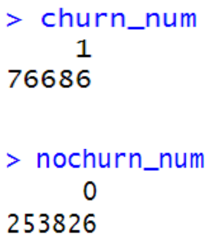
R

```
#Calculating Churn Numbers
target_num <- table(customer.df$Target_churn)

#Number of individuals who churned
churn_num <- target_num["1"]
Churn_num

#Number of individuals who did not churn
nochurn_num <- target_num["0"]
nochurn_num
```

Output:



```
> churn_num
    1
76686

> nochurn_num
     0
253826
```

The frequency analysis reveals that approximately **23.2%** of customers in the dataset have churned, while the remaining **76.8%** have not churned. This indicates a significant imbalance in the dataset, with the majority of customers retaining their subscriptions. Understanding this distribution is essential for evaluating the model's performance, as the imbalance may influence metrics like accuracy and sensitivity.

- **Visualizing Data**

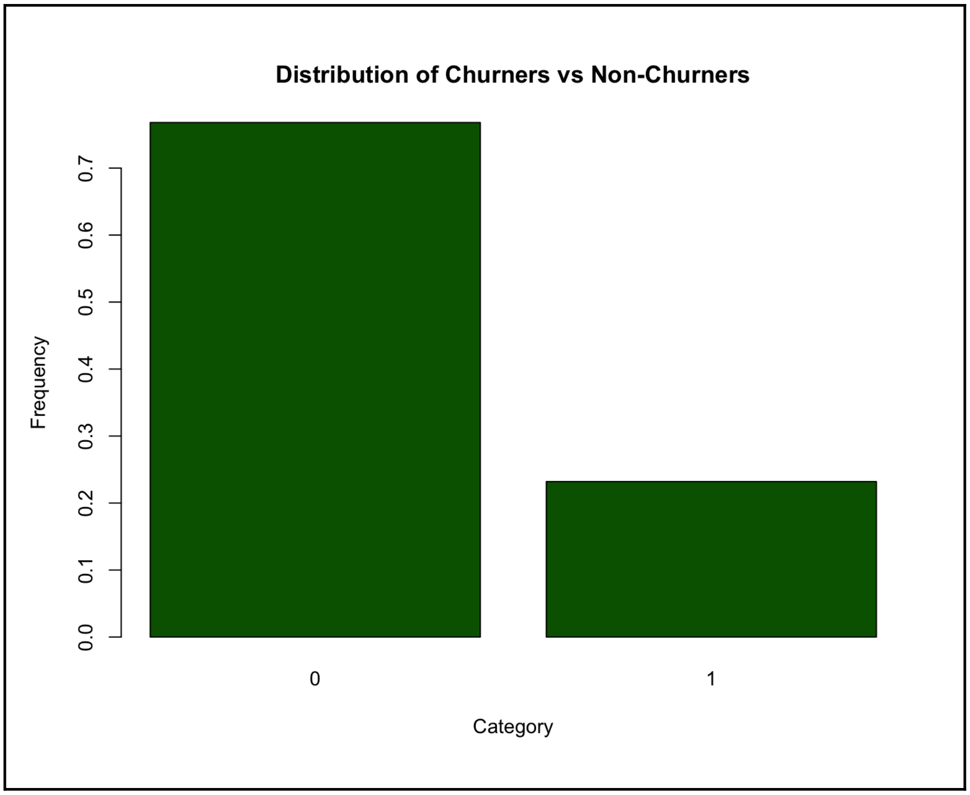
Exploratory Data Analysis (EDA) visualizations are crucial for gaining insights into the data before modeling.

Distribution of Churners VS Non Churners(Bar chart)

R

```
#Plotting frequency of observations of each category
barplot(target_freq, main = "Distribution of Churners vs Non-Churners", xlab = "Category", ylab =
"Frequency", col = "darkgreen", border = "black”)
```

Output:



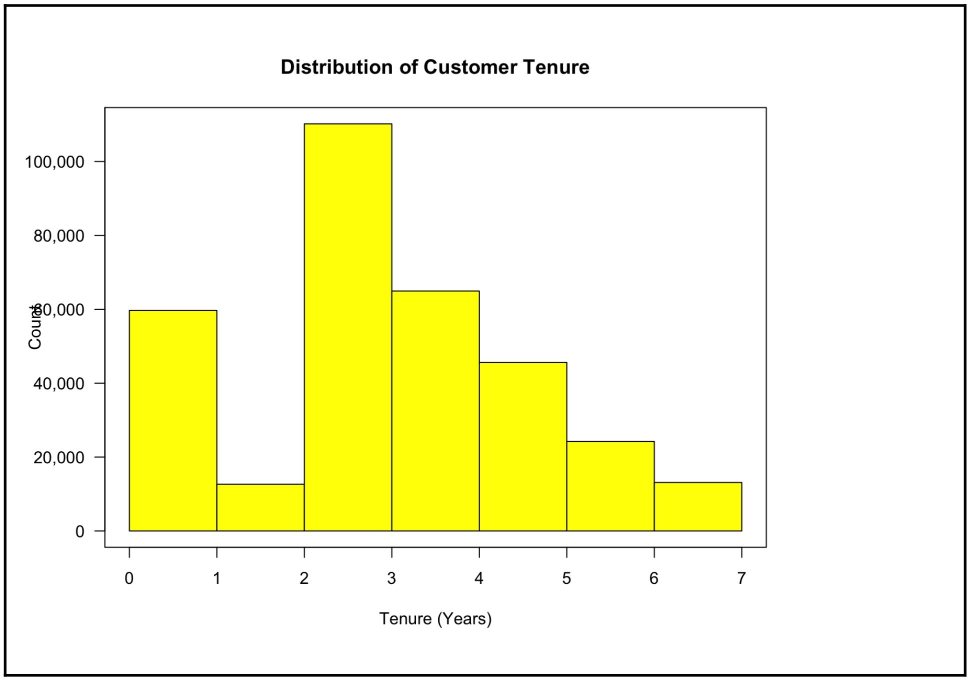
The bar chart shows the distribution of two categories: Churners (1) and Non-Churners (0). The height of the bars represents the relative frequency of each category. Non-Churners (0) have a significantly higher frequency compared to Churners (1), indicating that most customers in this dataset did not churn.

Customer Tenure (Histogram)

R

```
# Generating histogram to capture the distribution of customer tenure
hist_data <- hist(customer.df$Tenure,
                  breaks = 7,
                  col = "Yellow",
                  main = "Distribution of Customer Tenure",
                  xlab = "Tenure (Years)",
                  ylab = "Count",
                  axes = FALSE) # Suppress default axes
# Add x-axis
axis(1)
# Add y-axis with formatted labels
y_ticks <- pretty(range(hist_data$counts))
axis(2, at = y_ticks, labels = format(y_ticks, big.mark = ","), las = 1)
# Add box around the plot
box()
```

Output:



The histogram indicates that customer churn significantly decreases after the 3rd year of tenure. The number of customers reduces sharply beyond this point, suggesting that customers who remain subscribed for more than 3 years are less likely to churn, possibly indicating increased loyalty or satisfaction with the service.

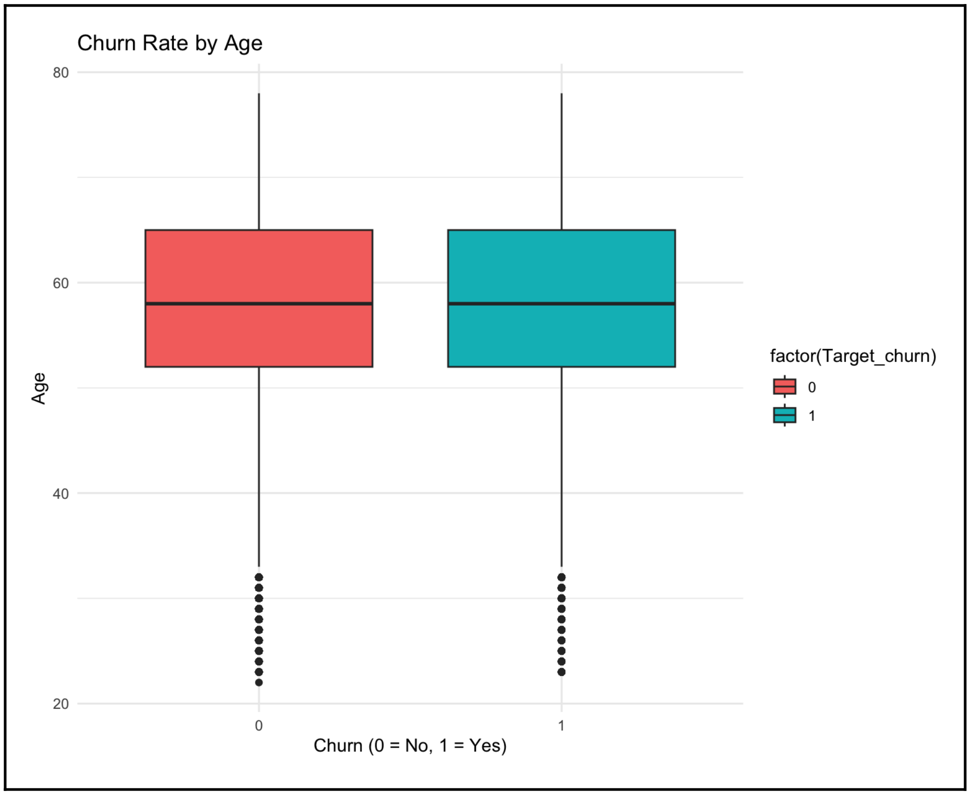
Churn Rate by Demographic Feature (Box plot)

By Age

R

```
# Boxplot by demographic feature (age)
library(ggplot2)
ggplot(customer.df, aes(x=factor(Target_churn), y=Age, fill=factor(Target_churn))) +
  geom_boxplot() +
  labs(title="Churn Rate by Age", x="Churn (0 = No, 1 = Yes)", y="Age") +
  theme_minimal()
```

Output:



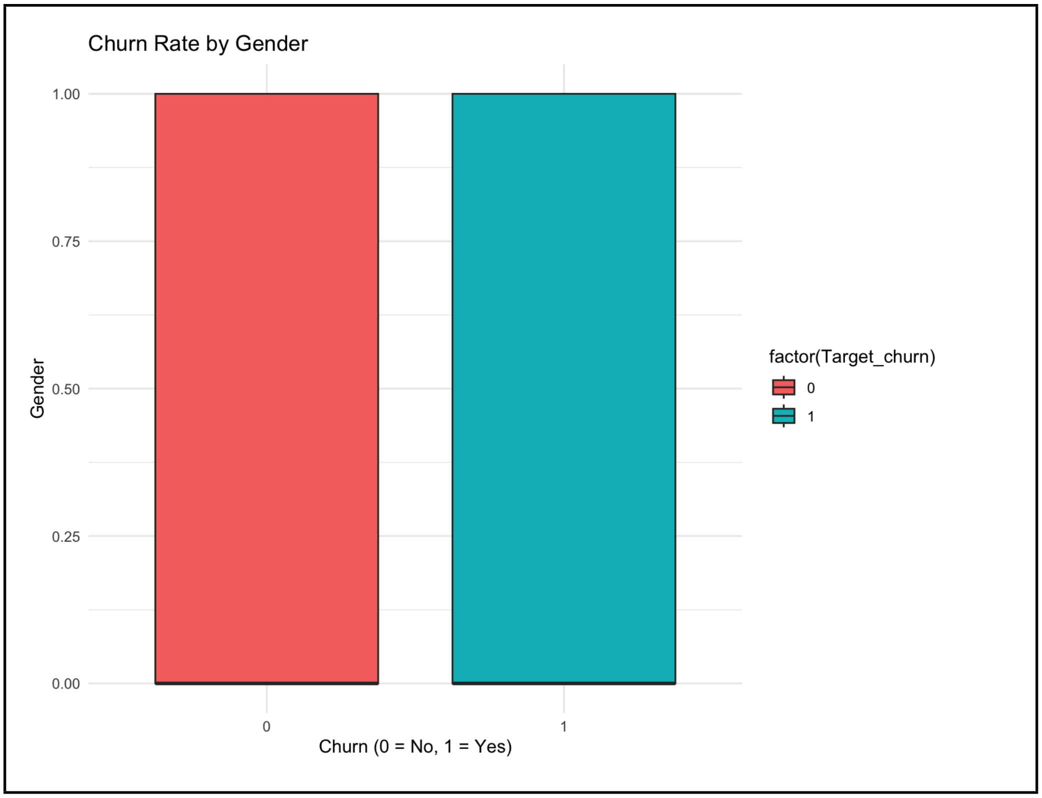
The box plot illustrates the distribution of customer ages for those who churned (1) and those who did not churn (0). The median age and the overall range of ages are nearly identical for both groups, with significant overlap in the interquartile ranges. This indicates that age does not appear to have a meaningful impact on churn behavior and the model.

By gender

R

```
# For gender
ggplot(customer.df, aes(x=factor(Target_churn), y=Gender, fill=factor(Target_churn))) +
  geom_boxplot() +
  labs(title="Churn Rate by Gender", x="Churn (0 = No, 1 = Yes)", y="Gender") +
  theme_minimal()
```

Output:



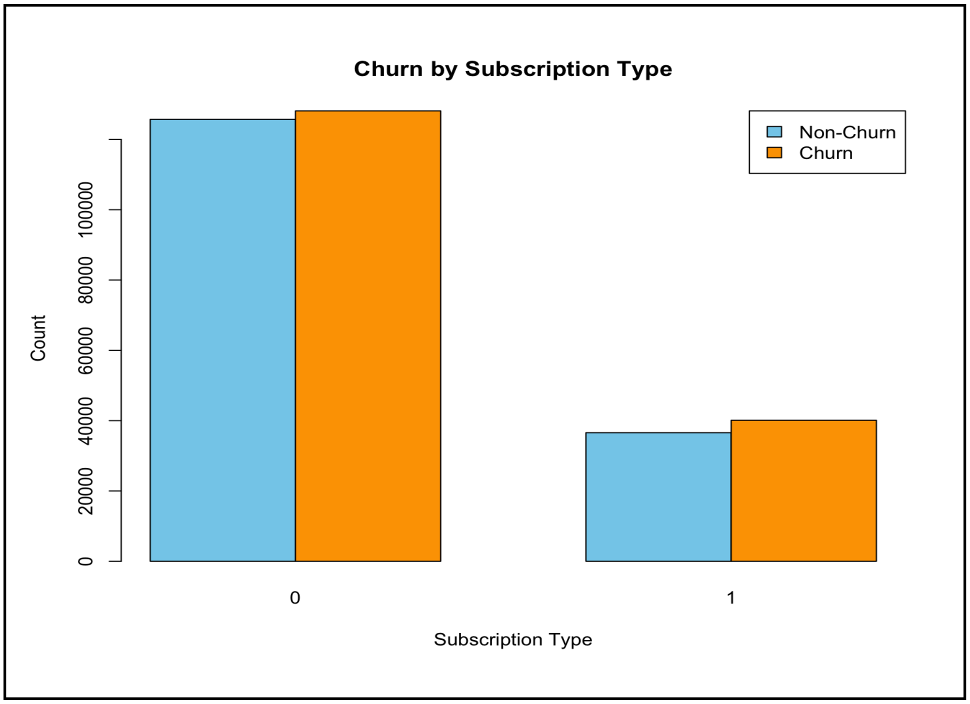
The bar chart displays the churn rate by gender, showing the proportion of churners (1) and non-churners (0) for each gender. The proportions are nearly identical across genders, with no significant difference in churn behavior observed. This suggests that gender does not have a meaningful impact on churn and to the model.

Subscription Type(Bar chart)

R

```r
# Create a contingency table for Subscription type and Churn
subscription_table <- table(customer.df$Subscription, customer.df$Target_churn)
# Create the barplot
barplot(subscription_table,
        beside = TRUE, # To create grouped bars
        col = c("skyblue", "orange"), # Colors for each group
        legend.text = c("Non-Churn", "Churn"), # Legend
        args.legend = list(x = "topright"), # Position of legend
        main = "Churn by Subscription Type",
        xlab = "Subscription Type",
        ylab = "Count")
```

Output:



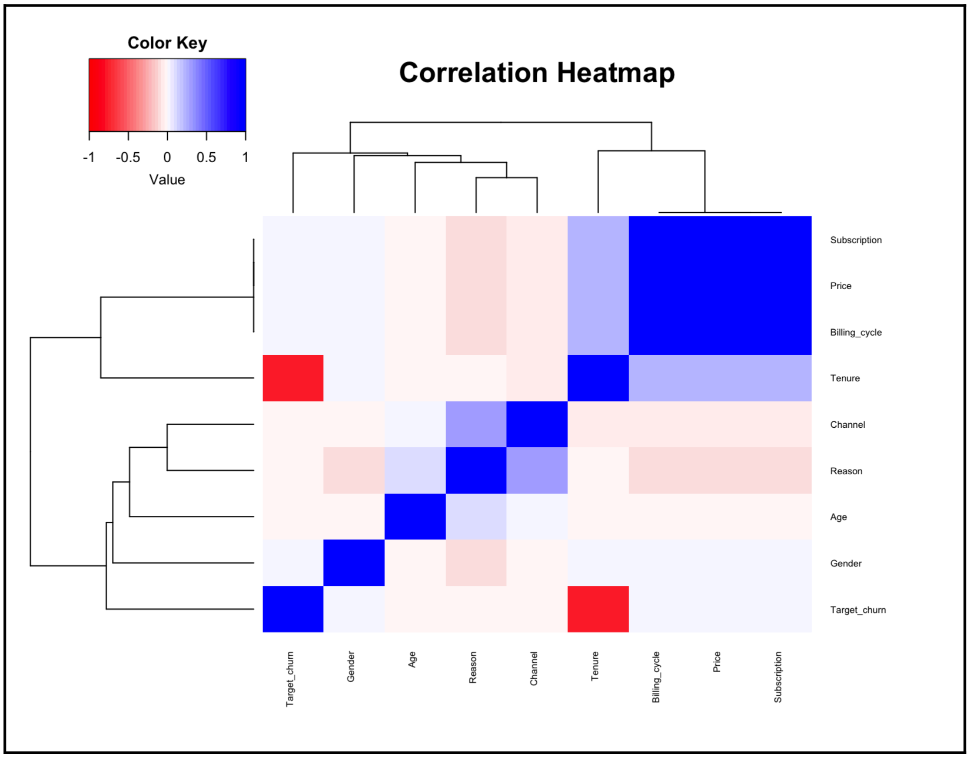
The bar chart compares churn rates for monthly (0) and yearly (1) subscription types. For monthly subscriptions, the number of churned customers (orange) slightly exceeds the non-churned (blue), indicating a higher churn rate. Conversely, for yearly subscriptions, the churn count is slightly lower than the non-churn count, suggesting better retention. While there is a small difference, it may not be significant enough to heavily impact the model's predictions.

Feature Correlation(Heatmap)

R

```r
#Feature Correlation (heatmap)
library(corrplot)
# Compute correlation matrix (numeric variables only)
numeric_data <- customer.df[, sapply(customer.df, is.numeric)]
# Compute the correlation matrix
cor_matrix <- cor(numeric_data, use = "complete.obs")
# Generate a heatmap
library(gplots)
heatmap.2(cor_matrix,
          main = "Correlation Heatmap",
          col = colorRampPalette(c("red", "white", "blue"))(50),
          trace = "none",
          density.info = "none",
          cexRow = 0.6,  # Reduce row label size
          cexCol = 0.6)
```

Output:



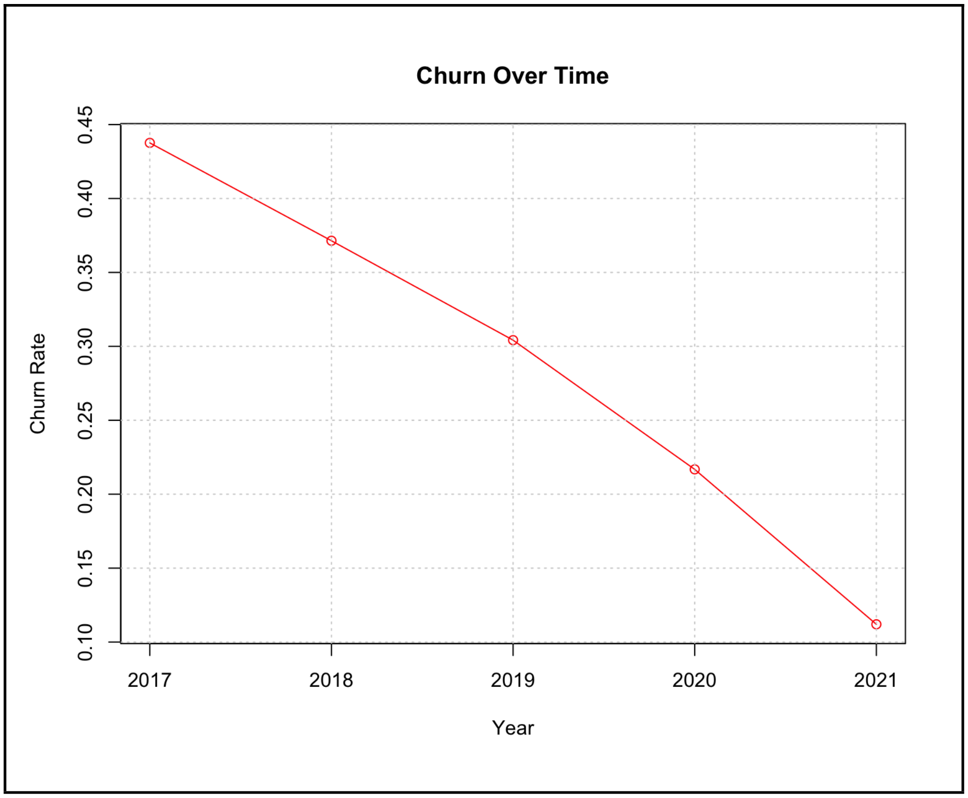
The correlation heatmap highlights the relationships between variables in the dataset. Notably, Target Churn shows a weak negative correlation with Tenure, suggesting that customers with longer tenure are less likely to churn. Billing Cycle and Price are highly correlated, as longer billing cycles (e.g., yearly subscriptions) often correspond to higher prices, indicating potential redundancy. Subscription type has a weak correlation with churn, while variables like Age and Gender show little to no correlation, suggesting limited predictive value. Overall, the heatmap helps identify key relationships and redundant features, guiding effective feature selection for the model.

Churn over time (Line chart)

R

```r
#Churn Over Time (Line Chart)
# Convert signup_date_time to Date format
customer.df$Signup_date_time <- as.Date(customer.df$Signup_date_time, format="%m/%d/%y")
# Group by year and calculate churn rate
library(dplyr)
churn_trend <- customer.df %>%
  mutate(year = format(Signup_date_time, "%Y")) %>%
  group_by(year) %>%
  summarise(churn_rate = mean(as.numeric(as.character(Target_churn))))
# Convert to data frame
churn_trend <- as.data.frame(churn_trend)
# Line chart using base R
plot(
  churn_trend$year, churn_trend$churn_rate,
  type = "o",                    # "o" for line and points
  col = "red",                   # Line color
  xlab = "Year",                   # X-axis label
  ylab = "Churn Rate",               # Y-axis label
  main = "Churn Over Time"            # Chart title
)
# Add grid lines for better readability
grid()
```

Output:



The line chart shows a steady decline in churn rates from 2017 to 2021. In 2017, the churn rate was around 45%, but it gradually decreased each year, reaching approximately 10% by 2021. This trend indicates improved customer retention over time, potentially reflecting effective strategies or changes in offerings that enhanced customer satisfaction and loyalty.

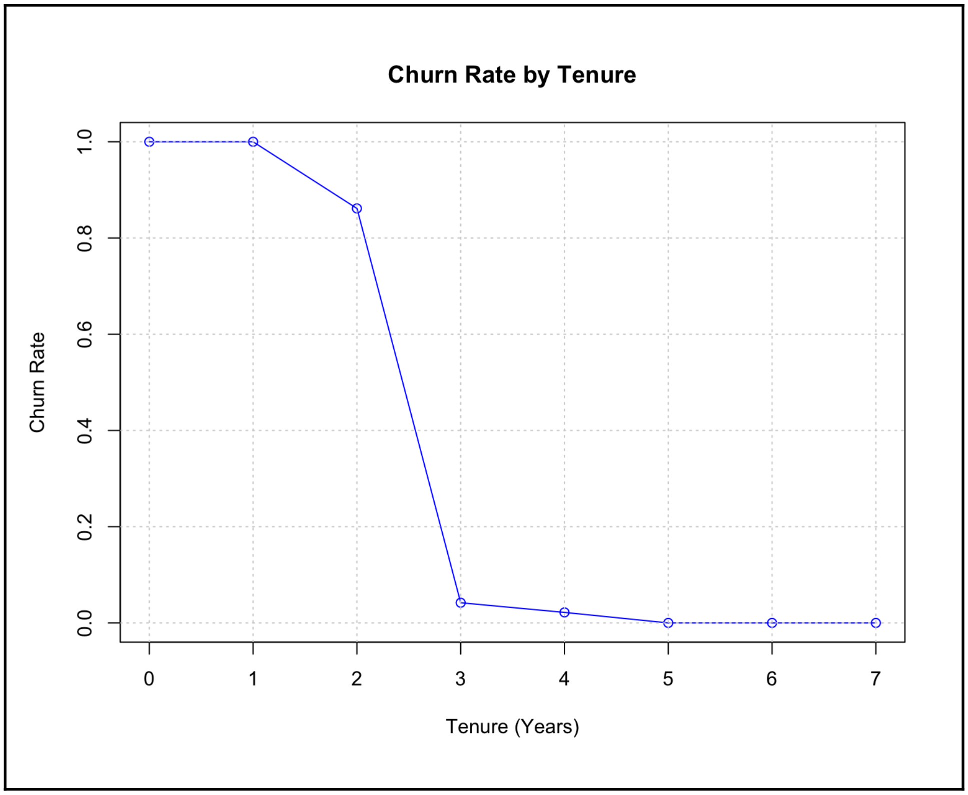
Churn over Tenure

R

```
#Churn Over Tenure (Line Chart)
# Ensure Tenure column is numeric
customer.df$Tenure <- as.numeric(customer.df$Tenure)
# Calculate churn rate for each Tenure value
churn_by_tenure <- aggregate(as.numeric(as.character(Target_churn)) ~ Tenure, data = customer.df, FUN = mean)
# Rename columns for clarity
colnames(churn_by_tenure) <- c("Tenure", "Churn_Rate")
# View the result
print(churn_by_tenure)
# Plot churn rate by Tenure
```

```
plot(
  churn_by_tenure$Tenure, churn_by_tenure$Churn_Rate,
  type = "o",                           # "o" for line and points
  col = "blue",                         # Line color
  xlab = "Tenure (Years)",              # X-axis label
  ylab = "Churn Rate",                  # Y-axis label
  main = "Churn Rate by Tenure",        # Chart title
  ylim = c(0, 1)                        # Set Y-axis range to 0-1
)
# Add grid for better readability
grid()
```

Output:



The line chart depicts the churn rate by customer tenure. It shows that churn is highest during the first two years, with a sharp decline after the third year. By the fourth year and beyond, the churn rate approaches zero, indicating that customers who remain subscribed for over three years are significantly less likely to leave. This highlights the importance of retaining customers during their early tenure to reduce overall churn.

CORRELATION ANALYSIS

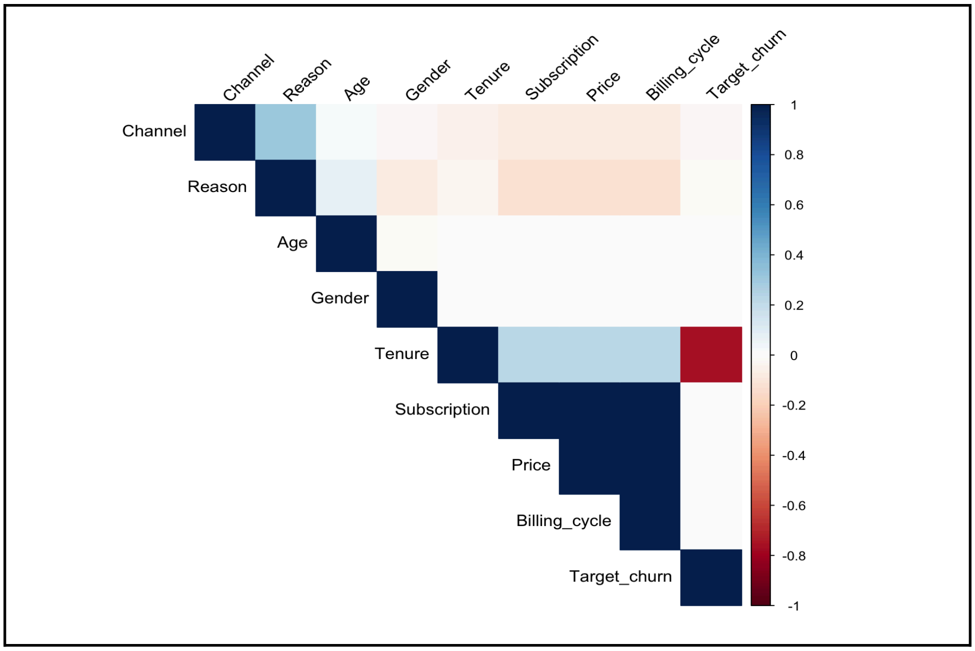
Measures the strength and direction of the relationship between features and the target. Features with high correlation to the target are typically retained, while low-correlation features are removed. Correlations can help you understand relationships that might influence churn:

- ***Identify Key Factors:*** If price has a high positive correlation with churn, it could mean higher prices are associated with higher churn rates, suggesting customers may find the product too expensive.
- ***Feature Selection for Modeling:*** Strongly correlated features with churn (either positively or negatively) are likely more predictive and can be prioritized in models.
- ***Detect Multicollinearity:*** If two features (like price and billing_cycle) are highly correlated with each other, they might introduce redundancy in models like logistic regression, where multicollinearity can affect model stability.

R

```
# Correlation Analysis
print(cor_matrix)
corrplot(cor_matrix, method="color", type="upper", tl.col="black", tl.srt=45)
```

Output:



4. FEATURE SELECTION

Correlation Analysis helps us to assess the relationship between each feature and the target variable and understand the importance of each variable in the dataset.
We need to divide the variables into the ones that we want to retain and the ones we are going to remove.

1. Dropped Variables: Weakly or Non-Correlated Predictors
In feature selection, dropped variables are those that show weak or no correlation with churn. These variables likely have minimal or no impact on the target variable, so they're removed to simplify the model without sacrificing its predictive power.

2. Remaining Variables: Strongly Correlated Predictors
After correlation analysis, remaining variables are those with the strongest correlations to churn. These variables show a significant relationship with the likelihood of a customer churning and are therefore considered the most important predictors. Retaining these variables ensures that the model or analysis focuses on factors that truly impact churn.

R

```
customer.df<-customer.df[,-c(1,6,7,8,11,12)]
```

The first three columns in the dataset (Case_id, Date_time, and Customer_id) were excluded as they serve solely as unique identifiers and do not contribute meaningful information for predicting churn. These columns lack relevance to customer behavior or subscription patterns.

Additionally, redundant columns such as Subscription were removed because their information overlaps with the target variable Target_churn. Retaining only Target_churn ensures the model remains focused on the primary prediction task while avoiding redundancy or multicollinearity, which could skew results. This process enhances the model's interpretability, efficiency, and overall predictive accuracy.

5. MODEL BUILDING:

Splitting the data into train and test for the models.

R

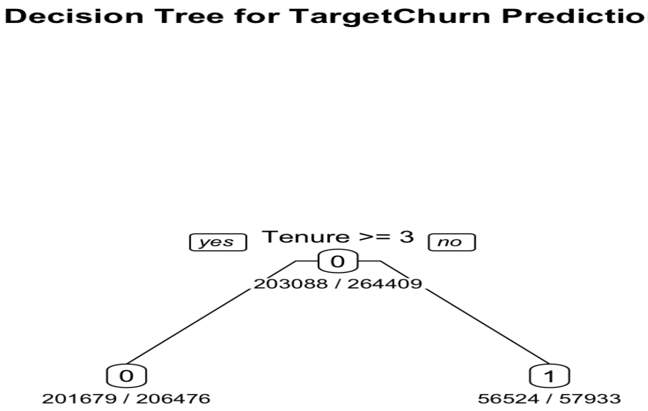
```
#Splitting data into training and validation sets
set.seed(1)
train.index <- sample(c(1:dim(customer.df)[1]), dim(customer.df)[1]*0.8)
train.df <- customer.df[train.index, ]
valid.df <- customer.df[-train.index, ]
```

DECISION TREE

R

```
library(rpart)
library(rpart.plot)
library(caret)
#Decision Tree
target_churn.ct <- rpart(Target_churn ~ ., data = train.df, method = "class")
# plot tree
prp(target_churn.ct, type = 1, extra = 2, under = TRUE, split.font = 1, varlen = 10, main="Decision Tree
for TargetChurn Prediction")
```

Output:



Overfitting :

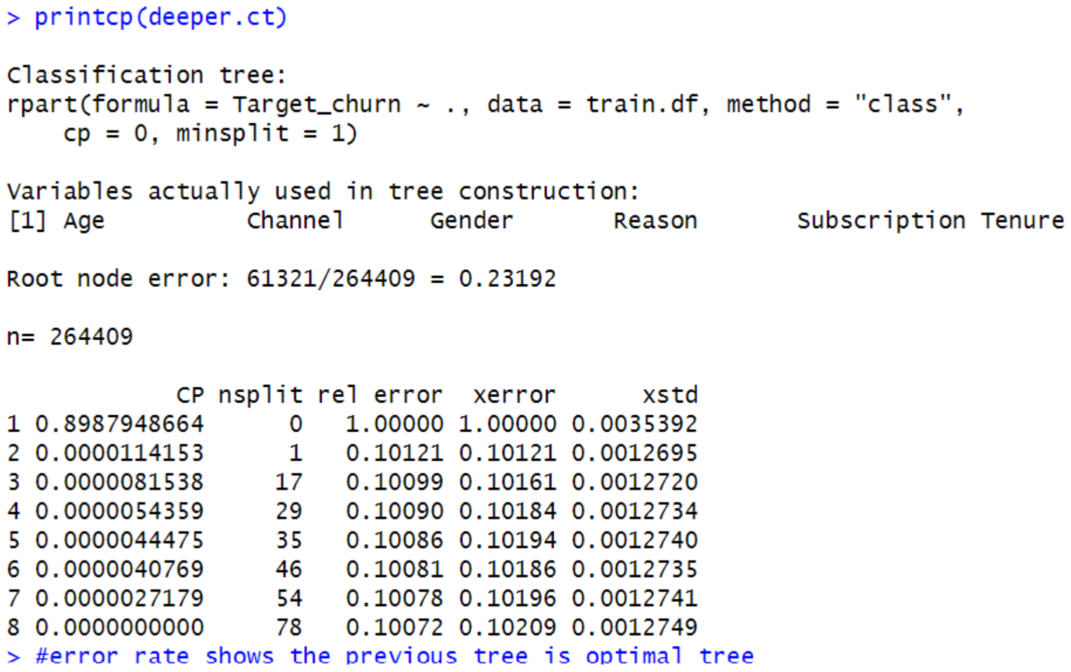
Overfitting occurs when a model learns the training data too well, including its noise and irrelevant patterns, leading to poor generalization of new data. While the model achieves high accuracy on the training set, its performance on validation or test sets deteriorates. Overfitting often results from overly complex models, insufficient data, or lack of regularization. Techniques like cross-validation, simplifying the model, pruning (for trees), or using regularization methods can mitigate overfitting and improve generalization.

DEEPER DECISION TREE (Post Pruning)

R

```
deeper.ct <- rpart(Target_churn ~ ., data = train.df, method = "class", cp = 0 , minsplit = 1)
length(deeper.ct$frame$var[deeper.ct$frame$var == "<leaf>"])
prp(deeper.ct, type = 1, extra = 1, under = TRUE, split.font = 1, varlen = -10,
   box.col=ifelse(deeper.ct$frame$var == "<leaf>", 'gray', 'white'))
printcp(deeper.ct)
#error rate shows the previous tree is optimal tree
```

Output:



```
> printcp(deeper.ct)

Classification tree:
rpart(formula = Target_churn ~ ., data = train.df, method = "class",
    cp = 0, minsplit = 1)

Variables actually used in tree construction:
[1] Age          Channel      Gender       Reason       Subscription Tenure

Root node error: 61321/264409 = 0.23192

n= 264409

            CP nsplit rel error  xerror      xstd
1 0.8987948664      0   1.00000 1.00000 0.0035392
2 0.0000114153      1   0.10121 0.10121 0.0012695
3 0.0000081538     17   0.10099 0.10161 0.0012720
4 0.0000054359     29   0.10090 0.10184 0.0012734
5 0.0000044475     35   0.10086 0.10194 0.0012740
6 0.0000040769     46   0.10081 0.10186 0.0012735
7 0.0000027179     54   0.10078 0.10196 0.0012741
8 0.0000000000     78   0.10072 0.10209 0.0012749
> #error rate shows the previous tree is optimal tree
```

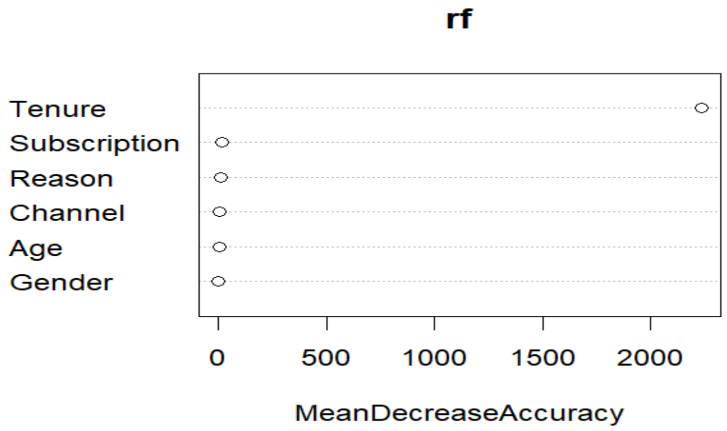
In a deeper decision tree, to address the issue of overfitting, pruning is performed by selecting the Complexity Parameter (CP) value associated with the minimum cross-validation error (xerror). Upon analyzing the CP table, it is observed that the second entry has the lowest xerror value, which aligns with the tree previously built. Therefore, the earlier tree already represents the optimal model for prediction. As a result, no further pruning is required, and the existing tree can be utilized for subsequent predictions effectively.

RANDOM FOREST

R

```
library(randomForest)
rf <- randomForest(as.factor(Target_churn) ~ ., data = train.df, ntree = 100,
                mtry = 4, nodesize = 5, importance = TRUE)
# variable importance plot
varImpPlot(rf, type = 1)
```

Output:



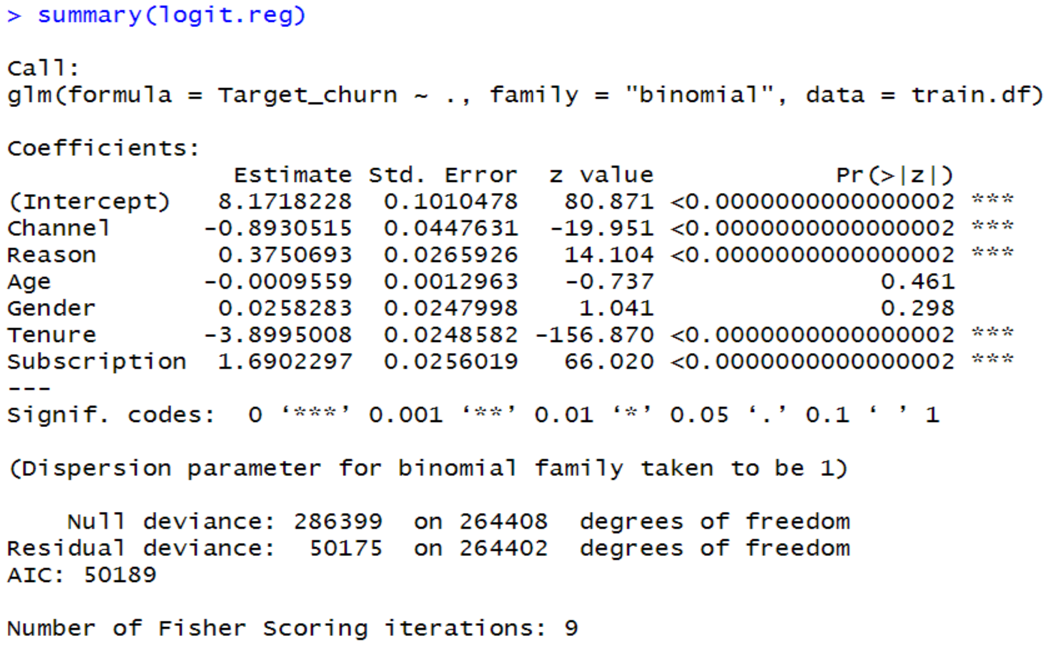
This plot shows the feature importance from the Random Forest (rf) model based on the MeanDecreaseAccuracy metric, which reflects how much each feature contributes to the model's predictive accuracy. Tenure is the most important feature, having the highest impact on model accuracy. Other features like Subscription, Reason, Channel, Age, and Gender have lower importance. This highlights that customer tenure plays a critical role in predicting churn, while demographic factors like Age and Gender contribute minimally to the model's performance.

LOGISTIC REGRESSION

R

```
logit.reg <- glm(Target_churn ~ ., data = train.df, family = "binomial")
options(scipen=999)
summary(logit.reg)
```

Output:



```
> summary(logit.reg)

Call:
glm(formula = Target_churn ~ ., family = "binomial", data = train.df)

Coefficients:
               Estimate Std. Error  z value            Pr(>|z|)    
(Intercept)   8.1718228  0.1010478   80.871 <0.0000000000000002 ***
Channel      -0.8930515  0.0447631  -19.951 <0.0000000000000002 ***
Reason        0.3750693  0.0265926   14.104 <0.0000000000000002 ***
Age          -0.0009559  0.0012963   -0.737               0.461    
Gender        0.0258283  0.0247998    1.041               0.298    
Tenure       -3.8995008  0.0248582 -156.870 <0.0000000000000002 ***
Subscription  1.6902297  0.0256019   66.020 <0.0000000000000002 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 286399  on 264408  degrees of freedom
Residual deviance:  50175  on 264402  degrees of freedom
AIC: 50189

Number of Fisher Scoring iterations: 9
```

The logistic regression model identifies key predictors of customer churn. Significant variables include Channel, Reason, Tenure, and Subscription, indicating their strong influence on churn behavior. For instance, longer Tenure significantly reduces the likelihood of churn, while certain Subscription types increase it. Variables like Age and Gender are not statistically significant, suggesting they have minimal impact on churn.

ANALYSIS OF MODELS

Measures	**Decision Tree**		**Random Forest**		**Logistic Regression**	
	Training data	**Validation data**	**Training data**	**Validation data**	**Training data**	**Validation data**
Accuracy	97.6%	97.5%	97.6%	97.5%	97.5%	97.5%
Error Rate	2.3%	2.4%	2.3%	2.4%	2.4%	2.4%
Sensitivity	92.1%	91.8%	92.1%	91.8%	91.8%	91.4%
Specificity	99.3%	99.3%	99.3%	99.3%	99.3%	99.3%

False Positive Rate	0.69%	.68%	.69%	.67%	.66%	.66%
False Negative Rate	7.8%	8.1%	7.8%	8.1%	8.1%	8.5%
Precision	97.5%	97.6%	97.5%	97.6%	97.6%	97.6%
F1 score	94.7%	94.6%	94.7%	94.6%	94.6%	94.4%

This table summarizes the performance evaluation of three machine learning models— Decision Tree, Random Forest, and Logistic Regression—used for predicting customer churn. The metrics are evaluated on both training and validation datasets to assess model performance and potential overfitting.

EXPLANATION OF METRICS
Accuracy: All models achieve high accuracy (~97.5%-97.6%) on both training and validation datasets, indicating that they are effective at predicting churn and non-churn classes
Error Rate: The error rates (~2.3%-2.4%) are consistent across all models, suggesting minimal misclassification overall.
Sensitivity (Recall): Sensitivity measures the ability to correctly identify churners (positive cases). The values (~91.4%-92.1%) are slightly lower than the overall accuracy, indicating that while the models are accurate, there is some room for improvement in detecting churners.
Specificity: Specificity measures the ability to correctly identify non-churners (negative cases). All models achieve extremely high specificity (~99.3%), meaning they are highly effective at identifying non-churn customers.

False Positive Rate (FPR): The FPR values (~0.66%-0.69%) are low, indicating that very few non-churners are mistakenly classified as churners.
False Negative Rate (FNR): The FNR (~7.8%-8.5%) is relatively higher than the FPR, showing that some churners are being missed, which could be critical for retention strategies.
Precision: Precision values (~97.5%-97.6%) indicate that when the models predict a customer will churn, they are correct in most cases, minimizing false positives.
F1 Score: The F1 scores (~94.4%-94.7%) provide a balanced measure of precision and recall, confirming that the models perform well overall in predicting churn.

MODEL COMPARISON

Decision Tree and Random Forest: These models perform almost identically, with high accuracy, precision, and F1 scores. Random Forest has a slightly lower FPR and FNR, making it marginally better at avoiding false classifications.

Logistic Regression: Logistic Regression also performs comparably, with similar accuracy and precision, but it's slightly higher FNR (8.5%) suggests it is less effective at identifying churners than the other models.

All three models perform well for churn prediction, with high accuracy and precision. However, Random Forest slightly outperforms the others due to its lower error rates, FPR, and FNR. For business purposes, focusing on improving sensitivity (recall) would help reduce the number of missed churners, which is critical for customer retention strategies.

6. MODEL EVALUATION

LIFT CHART

A lift chart evaluates a model's effectiveness by comparing its results to those of random selection. The lift is calculated as the ratio of true positives identified by the model to the expected true positives under random guessing.

CUMULATIVE LIFT

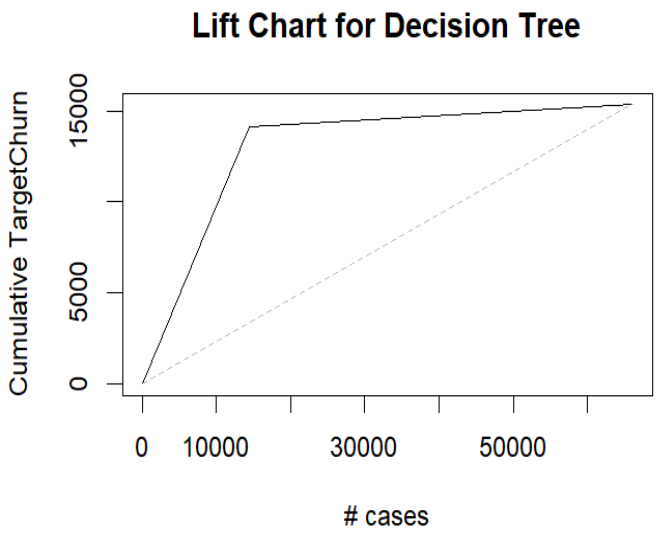
Cumulative lift reflects the model's performance of the target outcome. It shows how well the model captures the desired outcomes (e.g., churners) cumulatively as a percentage of the population is included. Higher cumulative lift indicates better performance in prioritizing high-probability cases.

1. Lift Chart for Decision Tree

R

```
tree_pred_valid_prob <- predict(target_churn.ct, valid.df, type = "prob")[, 2]
gain.dt <- gains(valid.df$Target_churn[!is.na(tree_pred_valid_prob)],
tree_pred_valid_prob[!is.na(tree_pred_valid_prob)])
# cumulative lift chart
plot(c(0,gain.dt$cume.pct.of.total*sum(TargetChurn))~c(0,gain.dt$cume.obs),
    xlab="# cases", ylab="Cumulative TargetChurn", main="Lift Chart for Decision Tree", type="l")
# baseline
lines(c(0,sum(TargetChurn))~c(0,dim(valid.df)[1]), col="gray", lty=2)
```

Output:

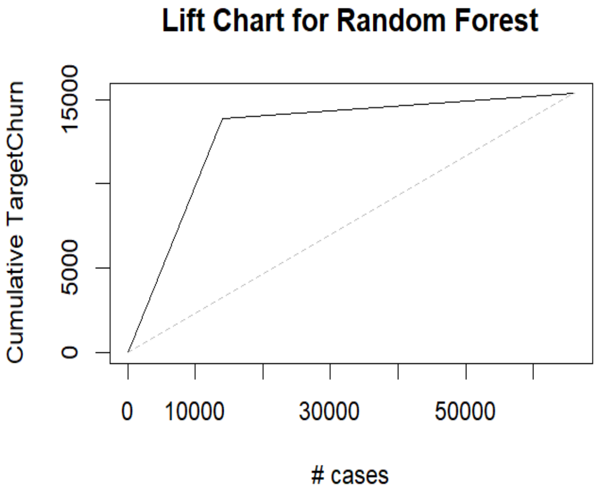


2. Lift Chart for Random Forest

R

```r
rf_pred_valid_prob <- predict(rf, valid.df, type = "prob")[, 2]
gain.rf <- gains(valid.df$Target_churn[!is.na(rf_pred_valid_prob)],
rf_pred_valid_prob[!is.na(rf_pred_valid_prob)])
# cumulative lift chart
plot(c(0,gain.rf$cume.pct.of.total*sum(TargetChurn))~c(0,gain.rf$cume.obs),
   xlab="# cases", ylab="Cumulative TargetChurn", main="Lift Chart for Random Forest", type="l")
# baseline
lines(c(0,sum(TargetChurn))~c(0,dim(valid.df)[1]), col="gray", lty=2)
```

Output:

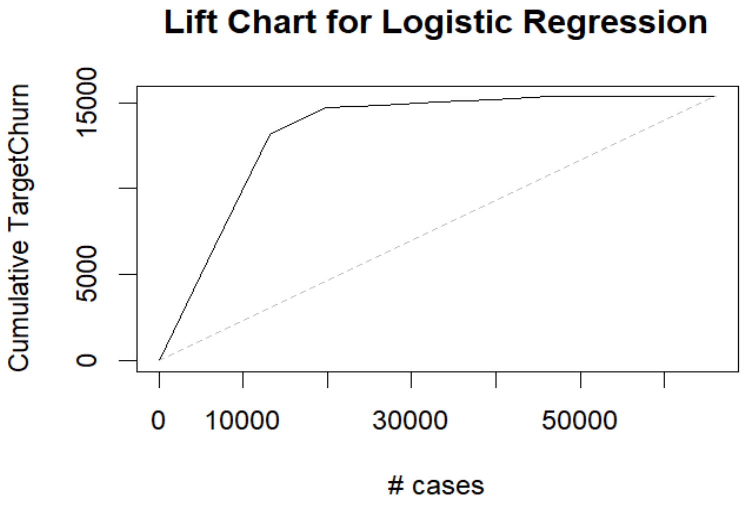


3. Lift Chart for Logistic Regression

R

```r
gain.lr <- gains(valid.df$Target_churn[!is.na(valid_pred_prob)],
valid_pred_prob[!is.na(valid_pred_prob)])
# cumulative lift chart
plot(c(0,gain.lr$cume.pct.of.total*sum(TargetChurn))~c(0,gain.lr$cume.obs),
   xlab="# cases", ylab="Cumulative TargetChurn", main="Lift Chart for Logistic Regression", type="l")
# baseline
lines(c(0,sum(TargetChurn))~c(0,dim(valid.df)[1]), col="gray", lty=2)
```

Output:



The lift charts for the Decision Tree, Random Forest, and Logistic Regression models compare their ability to predict customer churn against a random baseline. The Decision Tree demonstrates the highest lift, with a steep curve that rises quickly and effectively identifies churners, especially among the top-ranked cases. While Random Forest performs well, its curve is slightly less steep, indicating it may not prioritize high-risk churners as effectively as the Decision Tree. Logistic Regression has the least steep curve, showing relatively lower effectiveness in identifying churners compared to the Decision Tree and Random Forest. Overall, the Decision Tree emerges as the best-performing model, making it the most effective choice for predicting churn and informing customer retention strategies.

AUC-ROC (Area Under the Curve - Receiver Operating Characteristic)

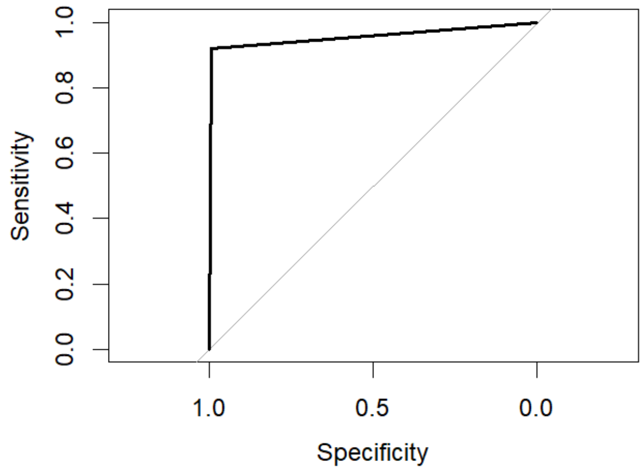
The **AUC-ROC** is to evaluate the performance of a binary classification model. It provides a comprehensive measure of a model's ability to distinguish between the positive and negative classes, considering all possible classification thresholds.

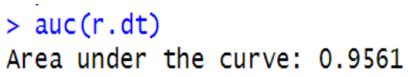
1. ROC for Decision Tree

R

```
r.dt <- roc(valid.df$Target_churn, tree_pred_valid_prob, levels = c(0, 1), direction = "<")
plot.roc(r.dt)
#Computing AUC for Decision Tree
auc(r.dt)
```

Output:





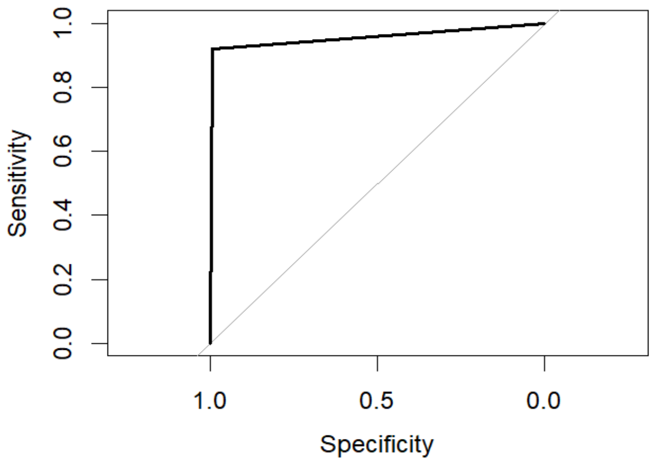
```
> auc(r.dt)
Area under the curve: 0.9561
```

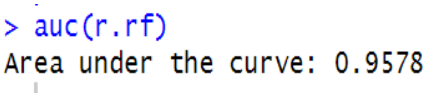
2. ROC for Random Forest :

R

```
r.rf <- roc(valid.df$Target_churn, rf_pred_valid_prob, levels = c(0, 1), direction = "<")
plot.roc(r.rf)
#Computing AUC for Random Forest
auc(r.rf)
```

Output





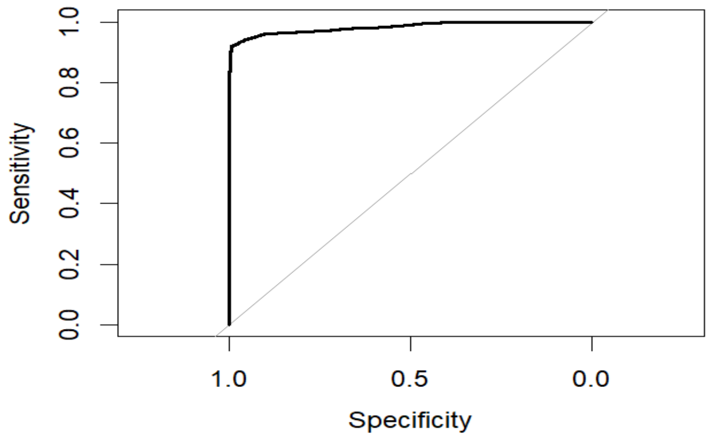
```
> auc(r.rf)
Area under the curve: 0.9578
```

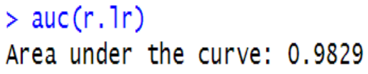
3. ROC for Logistic Regression

R

```
r.lr <- roc(valid.df$Target_churn, valid_pred_prob, levels = c(0, 1), direction = "<")
plot.roc(r.lr)
#Computing AUC for Logistic Regression
auc(r.lr)
```

Output





```
> auc(r.lr)
Area under the curve: 0.9829
```

Based on the Lift Chart and AUC values, Logistic Regression is the better model. The lift chart for Logistic Regression shows strong performance in identifying high-probability churners early on, and its results remain consistent. Additionally, the AUC value of 0.9829 for Logistic Regression is higher than both the Decision Tree (0.9561) and Random Forest (0.9578), showing it does a better job of telling churners and non-churners apart. These results make Logistic Regression the best choice for this analysis.

DEPLOYMENT AND REPORT

The AUC-ROC, accuracy, and F1 scores summarize the overall effectiveness of the models in predicting churn. AUC-ROC evaluates the model's ability to distinguish between churners and non-churners across all thresholds. Accuracy reflects the proportion of correctly classified cases, while F1 balances precision and recall.

If the focus is on identifying churners accurately (reducing false negatives), recall should be prioritized. High recall ensures that most actual churners are correctly identified, even if some non-churners are mistakenly flagged as churners. This approach is suitable when missing a churner (false negative) has significant consequences, such as losing valuable customers.

Alternatively, if minimizing false positives is more critical (predicting churn when it's not), the focus should shift to precision. High precision ensures that when the model predicts churn, it is likely correct, reducing unnecessary interventions or costs.

Depending on the company's requirements, whether reducing false negatives or false positives is more critical, the model with either high recall or high precision can be selected. For example:

- We can use the model with high recall when the cost or impact of missing churners is high
- We can use the model with high precision when the cost of acting on false churn predictions is high, such as offering incentives or discounts to customers who were not at risk of leaving.

KEY DRIVERS OF CHURN

Feature importance analysis across models indicates that Tenure consistently emerges as the most significant predictor of churn. Features like Subscription, Reason, and Channel also contribute to predictions but to a lesser degree. Demographic variables like Age and Gender show limited predictive power, aligning with the earlier analysis.

SENSITIVITY ANALYSIS

A sensitivity analysis reveals that subscription type and pricing play a crucial role in influencing customer tenure and churn rates. Higher prices are associated with shorter customer tenure and increased churn, particularly for customers on monthly subscriptions. Conversely, longer billing cycles, such as yearly subscriptions, tend to extend customer tenure and reduce churn rates. This suggests that offering pricing incentives or discounts for annual subscriptions could encourage longer commitments, enhancing customer retention and minimizing churn.

ACTIONABLE INSIGHTS

1. Pricing Recommendations: Consider introducing tiered pricing with discounts for longer billing cycles to reduce churn.

2. Retention Strategies:

- Focus on retaining customers in their first three years, as churn rates are highest during this period.
- Offer targeted retention campaigns based on tenure and subscription type.

3. Customer Segmentation:

- Prioritize interventions for monthly subscribers and customers nearing the end of their first or second year of tenure.\

- Use predictive models to identify high-risk churn segments for proactive outreach.

These insights provide a roadmap for reducing churn while optimizing customer retention strategies and pricing models.

CONCLUSION:

This project focused on predicting customer churn for subscription-based services using Decision Tree, Random Forest, and Logistic Regression models. Through thorough data preprocessing, exploratory data analysis, and model evaluation, we identified key factors contributing to churn and developed predictive models.

Among the models, Logistic Regression proved to be the most optimal based on both the lift chart and the AUC value (0.9829), demonstrating its ability to accurately distinguish between churners and non-churners.

By using these insights from this analysis, subscription-based businesses can prioritize high-risk customers, implement targeted retention efforts, and reduce churn. This project highlights the importance of predictive analytics in improving customer loyalty and maximizing profitability.