Predicting Customer Churn on Telecomdataset

LakshmiManaswitha

Customer churn occurs when customers or subscribers stop doing business with a company or service, also known as customer attrition. It is also referred as loss of clients or customers. One industry in which churn rates are particularly useful is the telecommunications industry, because most customers have multiple options from which to choose within a geographic location.

DataLoading

```
churn <- read.csv('/Users/manaswithachimakurthi/Desktop/pro/WA_Fn-UseC_-Telco-Customer-C
hurn.csv')
str(churn)
```

```
## 'data.frame':    7043 obs. of  21 variables:
##  $ customerID      : Factor w/ 7043 levels "0002-ORFBO","0003-MKNFE",..: 5376 3963 25
65 5536 6512 6552 1003 4771 5605 4535 ...
##  $ 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 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 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 ...
```

Checking for NA Values int he Predictors

```
sapply(churn, function(x) sum(is.na(x)))
```

```
##       customerID           gender    SeniorCitizen          Partner
##                0                0                0                0
##       Dependents           tenure     PhoneService    MultipleLines
##                0                0                0                0
##  InternetService   OnlineSecurity     OnlineBackup DeviceProtection
##                0                0                0                0
##      TechSupport      StreamingTV  StreamingMovies         Contract
##                0                0                0                0
## PaperlessBilling    PaymentMethod   MonthlyCharges     TotalCharges
##                0                0                0               11
##            Churn
##                0
```

```
churn <- churn[complete.cases(churn), ]
```

A Glimpse of the data set

```
head(churn)
```

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

data wrangling changing the No internet to No

We will change “No internet service” to “No” for six columns, they are: “OnlineSecurity”, “OnlineBackup”, “DeviceProtection”, “TechSupport”, “streamingTV”, “streamingMovies”.

```
cols_recode1 <- c(10:15)
for(i in 1:ncol(churn[,cols_recode1])) {
        churn[,cols_recode1][,i] <- as.factor(mapvalues
                                              (churn[,cols_recode1][,i], from =c("No int
ernet service"),to=c("No")))
}
```

Changing the values to no

```
churn$MultipleLines <- as.factor(mapvalues(churn$MultipleLines,
                                           from=c("No phone service"),
                                           to=c("No")))
```

Categorizing the tenure into periods

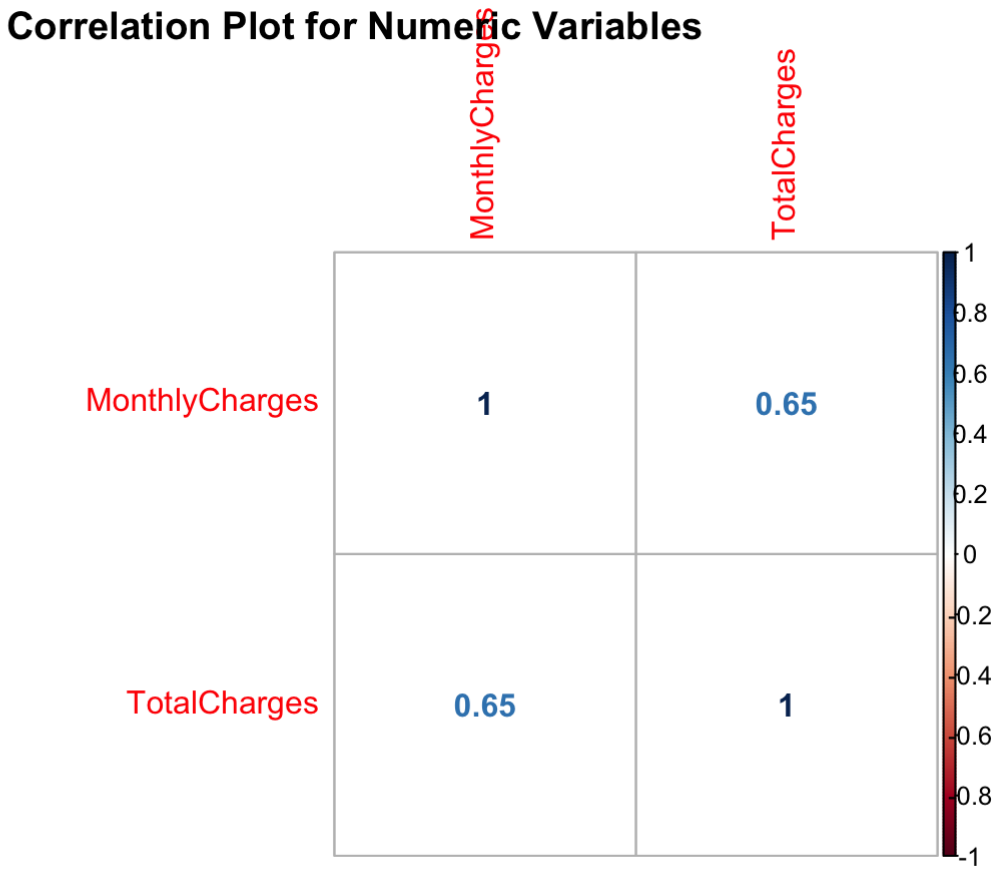
```
group_tenure <- function(tenure){
    if (tenure >= 0 & tenure <= 12){
        return('0-12 Month')
    }else if(tenure > 12 & tenure <= 24){
        return('12-24 Month')
    }else if (tenure > 24 & tenure <= 48){
        return('24-48 Month')
    }else if (tenure > 48 & tenure <=60){
        return('48-60 Month')
    }else if (tenure > 60){
        return('> 60 Month')
    }
}
churn$tenure_group <- sapply(churn$tenure,group_tenure)
churn$tenure_group <- as.factor(churn$tenure_group)
```

Change the values in column “SeniorCitizen” from 0 or 1 to “No” or “Yes”.

```
churn$SeniorCitizen <- as.factor(mapvalues(churn$SeniorCitizen,
                                           from=c("0","1"),
                                           to=c("No", "Yes")))
#Remove the columns we do not need for the analysis:

churn$customerID <- NULL
churn$tenure <- NULL
```

Exploratory data analysis and feature selection

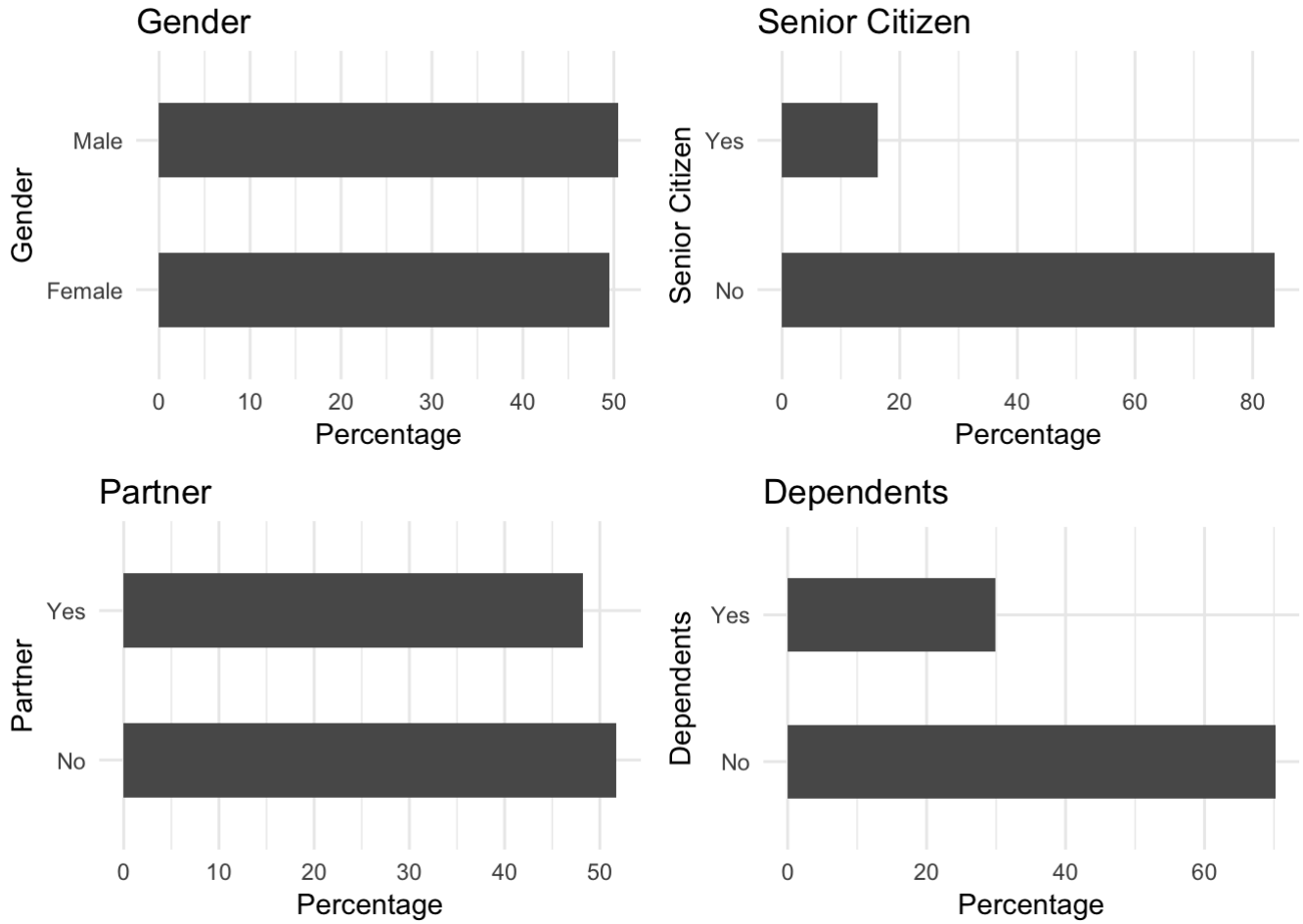
```
numeric.var <- sapply(churn, is.numeric) ## Find numerical variables
corr.matrix <- cor(churn[,numeric.var])  ## Calculate the correlation matrix
corrplot(corr.matrix, main="\n\nCorrelation Plot for Numeric Variables", method="number"
)
```



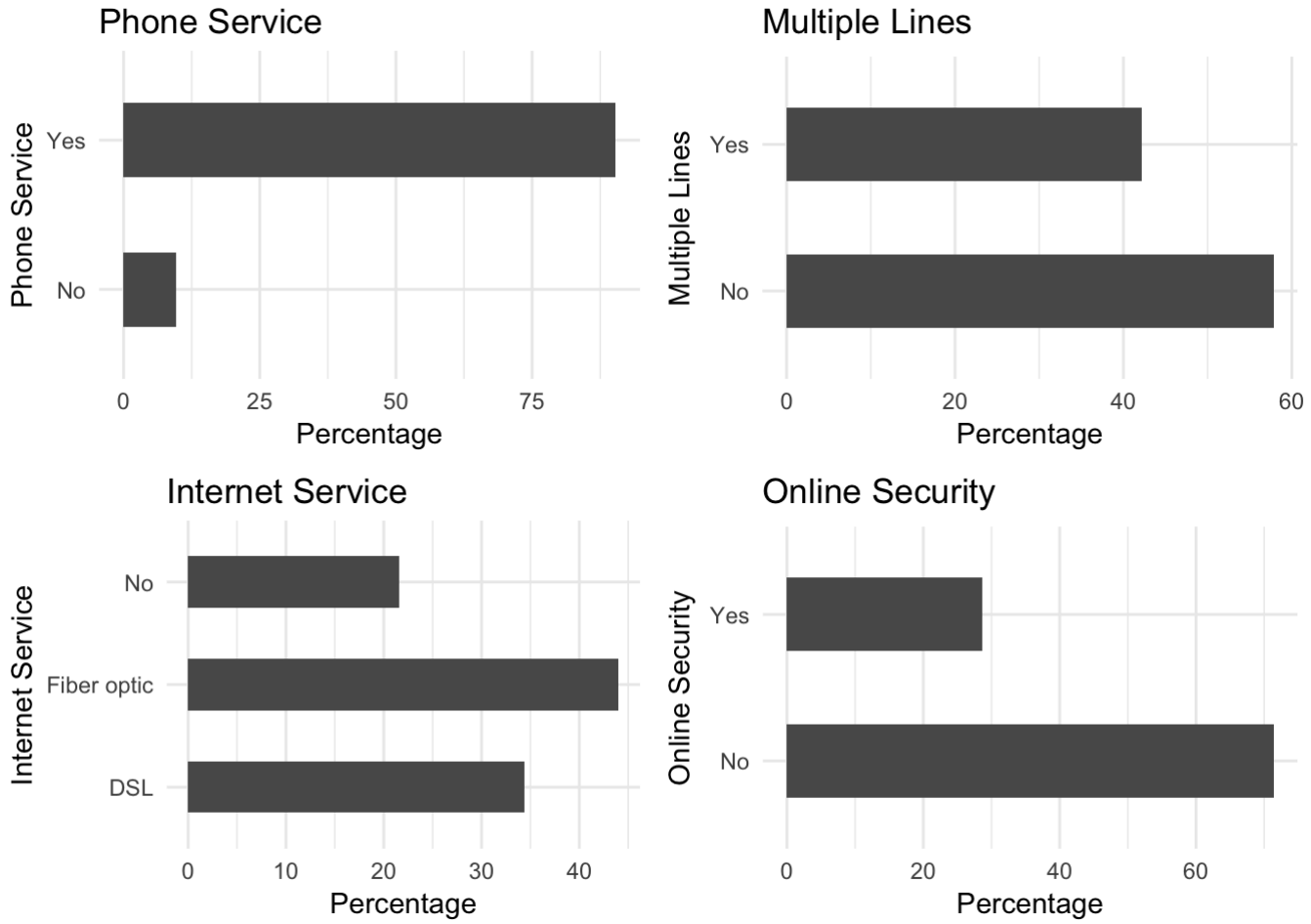

```
#The Monthly Charges and Total Charges are correlated. So one of them will be removed fr
om the model. We remove Total Charges.

churn$TotalCharges <- NULL
```

Distribution of categorical Variables

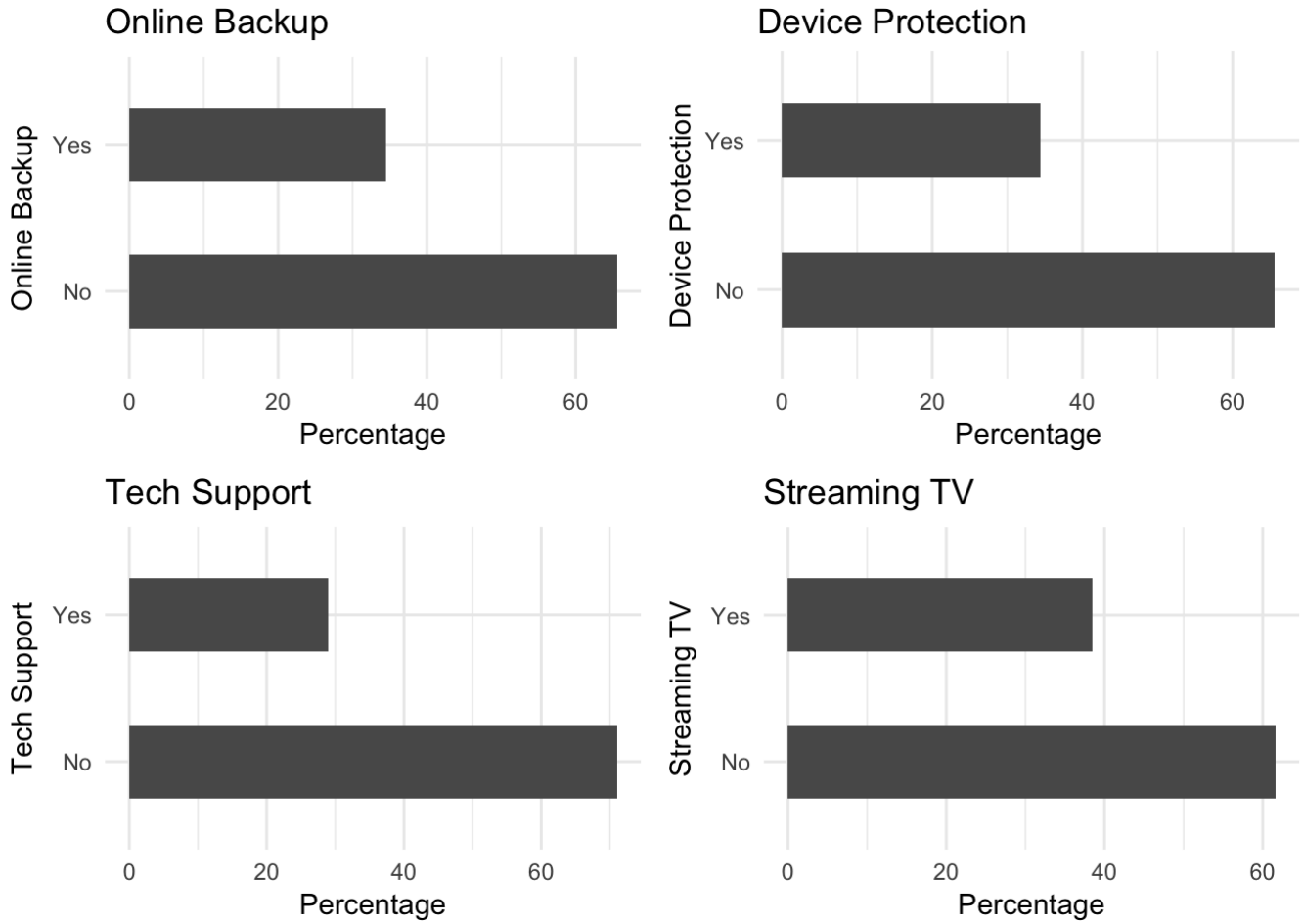
```
p1 <- ggplot(churn, aes(x=gender)) + ggtitle("Gender") + xlab("Gender") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") +
coord_flip() + theme_minimal()
p2 <- ggplot(churn, aes(x=SeniorCitizen)) + ggtitle("Senior Citizen") + xlab("Senior Cit
izen") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") +
coord_flip() + theme_minimal()
p3 <- ggplot(churn, aes(x=Partner)) + ggtitle("Partner") + xlab("Partner") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") +
coord_flip() + theme_minimal()
p4 <- ggplot(churn, aes(x=Dependents)) + ggtitle("Dependents") + xlab("Dependents") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") +
coord_flip() + theme_minimal()
grid.arrange(p1, p2, p3, p4, ncol=2)
```




```
p5 <- ggplot(churn, aes(x=PhoneService)) + ggtitle("Phone Service") + xlab("Phone Servic
e") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") +
coord_flip() + theme_minimal()
p6 <- ggplot(churn, aes(x=MultipleLines)) + ggtitle("Multiple Lines") + xlab("Multiple L
ines") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") +
coord_flip() + theme_minimal()
p7 <- ggplot(churn, aes(x=InternetService)) + ggtitle("Internet Service") + xlab("Intern
et Service") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") +
coord_flip() + theme_minimal()
p8 <- ggplot(churn, aes(x=OnlineSecurity)) + ggtitle("Online Security") + xlab("Online S
ecurity") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") +
coord_flip() + theme_minimal()
grid.arrange(p5, p6, p7, p8, ncol=2)
```




```
p9 <- ggplot(churn, aes(x=OnlineBackup)) + ggtitle("Online Backup") + xlab("Online Backu
p") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") +
coord_flip() + theme_minimal()
p10 <- ggplot(churn, aes(x=DeviceProtection)) + ggtitle("Device Protection") + xlab("Dev
ice Protection") +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") +
coord_flip() + theme_minimal()
p11 <- ggplot(churn, aes(x=TechSupport)) + ggtitle("Tech Support") + xlab("Tech Support"
) +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") +
coord_flip() + theme_minimal()
p12 <- ggplot(churn, aes(x=StreamingTV)) + ggtitle("Streaming TV") + xlab("Streaming TV"
) +
  geom_bar(aes(y = 100*(..count..)/sum(..count..)), width = 0.5) + ylab("Percentage") +
coord_flip() + theme_minimal()
grid.arrange(p9, p10, p11, p12, ncol=2)
```



Splitting the data into train and test data sets

```
set.seed(2017)
ratio = sample(1:nrow(churn), size = 0.7*nrow(churn))
training<-churn[ratio,]
testing<-churn[-ratio,]
```

Dimensions of the train and test datasets

```
dim(training)
```

```
## [1] 4922   19
```

```
dim(testing)
```

```
## [1] 2110   19
```

logistic regression model

```
LogModel <- glm(Churn ~ .,family=binomial(link="logit"),data=training)
print(summary(LogModel))
```

```
##
## Call:
## glm(formula = Churn ~ ., family = binomial(link = "logit"), data = training)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9477  -0.6671  -0.2740   0.6591   3.1163
##
## Coefficients:
##                                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)                       -1.820583   0.985536  -1.847  0.06470
## genderMale                        -0.075589   0.077976  -0.969  0.33235
## SeniorCitizenYes                   0.308723   0.102245   3.019  0.00253
## PartnerYes                        -0.087392   0.093582  -0.934  0.35038
## DependentsYes                     -0.051469   0.108202  -0.476  0.63431
## PhoneServiceYes                   -0.319875   0.776548  -0.412  0.68040
## MultipleLinesYes                   0.374221   0.209893   1.783  0.07460
## InternetServiceFiber optic         1.033753   0.953857   1.084  0.27847
## InternetServiceNo                 -1.159318   0.965159  -1.201  0.22969
## OnlineSecurityYes                 -0.385072   0.213824  -1.801  0.07172
## OnlineBackupYes                   -0.195948   0.209792  -0.934  0.35030
## DeviceProtectionYes                0.060394   0.212830   0.284  0.77659
## TechSupportYes                    -0.201154   0.215156  -0.935  0.34983
## StreamingTVYes                     0.300211   0.389960   0.770  0.44139
## StreamingMoviesYes                 0.343099   0.393922   0.871  0.38376
## ContractOne year                  -0.798861   0.130920  -6.102 1.05e-09
## ContractTwo year                  -1.636556   0.217776  -7.515 5.70e-14
## PaperlessBillingYes                0.389359   0.089415   4.355 1.33e-05
## PaymentMethodCredit card (automatic) -0.108663   0.135607  -0.801  0.42295
## PaymentMethodElectronic check      0.358409   0.113588   3.155  0.00160
## PaymentMethodMailed check         -0.002276   0.137111  -0.017  0.98675
## MonthlyCharges                    -0.008560   0.037940  -0.226  0.82150
## tenure_group0-12 Month             1.904132   0.212168   8.975  < 2e-16
## tenure_group12-24 Month            0.990116   0.208288   4.754 2.00e-06
## tenure_group24-48 Month            0.611728   0.190747   3.207  0.00134
## tenure_group48-60 Month            0.418348   0.203527   2.055  0.03983
##
## (Intercept)                       .
## genderMale
## SeniorCitizenYes                  **
## PartnerYes
## DependentsYes
## PhoneServiceYes
## MultipleLinesYes                  .
## InternetServiceFiber optic
## InternetServiceNo
## OnlineSecurityYes                 .
## OnlineBackupYes
## DeviceProtectionYes
## TechSupportYes
## StreamingTVYes
## StreamingMoviesYes
## ContractOne year                  ***
```

```
## ContractTwo year                     ***
## PaperlessBillingYes                  ***
## PaymentMethodCredit card (automatic)
## PaymentMethodElectronic check        **
## PaymentMethodMailed check
## MonthlyCharges
## tenure_group0-12 Month               ***
## tenure_group12-24 Month              ***
## tenure_group24-48 Month              **
## tenure_group48-60 Month              *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5755.6  on 4921  degrees of freedom
## Residual deviance: 4061.7  on 4896  degrees of freedom
## AIC: 4113.7
##
## Number of Fisher Scoring iterations: 6
```

```
anova(LogModel, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Churn
##
## Terms added sequentially (first to last)
##
##
##                  Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                                4921     5755.6
## gender            1     1.70      4920     5753.9  0.192791
## SeniorCitizen     1   116.66      4919     5637.2 < 2.2e-16 ***
## Partner           1   132.75      4918     5504.5 < 2.2e-16 ***
## Dependents        1    23.83      4917     5480.7 1.051e-06 ***
## PhoneService      1     0.10      4916     5480.6  0.751556
## MultipleLines     1     7.50      4915     5473.1  0.006153 **
## InternetService   2   465.06      4913     5008.0 < 2.2e-16 ***
## OnlineSecurity    1   203.17      4912     4804.8 < 2.2e-16 ***
## OnlineBackup      1    98.49      4911     4706.3 < 2.2e-16 ***
## DeviceProtection  1    40.37      4910     4666.0 2.100e-10 ***
## TechSupport       1    72.61      4909     4593.3 < 2.2e-16 ***
## StreamingTV       1     0.49      4908     4592.9  0.484013
## StreamingMovies   1     0.54      4907     4592.3  0.462853
## Contract          2   307.83      4905     4284.5 < 2.2e-16 ***
## PaperlessBilling  1    17.70      4904     4266.8 2.585e-05 ***
## PaymentMethod     3    41.79      4901     4225.0 4.453e-09 ***
## MonthlyCharges    1     0.11      4900     4224.9  0.737634
## tenure_group      4   163.17      4896     4061.7 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Analyzing the deviance table we can see the drop in deviance when adding each variable one at a time. Adding InternetService, Contract and tenure_group significantly reduces the residual deviance. The other variables such as PaymentMethod and Dependents seem to improve the model less even though they all have low p-values.

logistic regression Accuracy

```
testing$Churn <- as.character(testing$Churn)
testing$Churn[testing$Churn=="No"] <- "0"
testing$Churn[testing$Churn=="Yes"] <- "1"
fitted.results <- predict(LogModel,newdata=testing,type='response')
fitted.results <- ifelse(fitted.results > 0.5,1,0)
misClasificError <- mean(fitted.results != testing$Churn)
print(paste('Logistic Regression Accuracy',1-misClasificError))
```

```
## [1] "Logistic Regression Accuracy 0.800947867298578"
```

Confusion Matrix for Logistic Regression Model

```
print("Confusion Matrix for Logistic Regression")
```

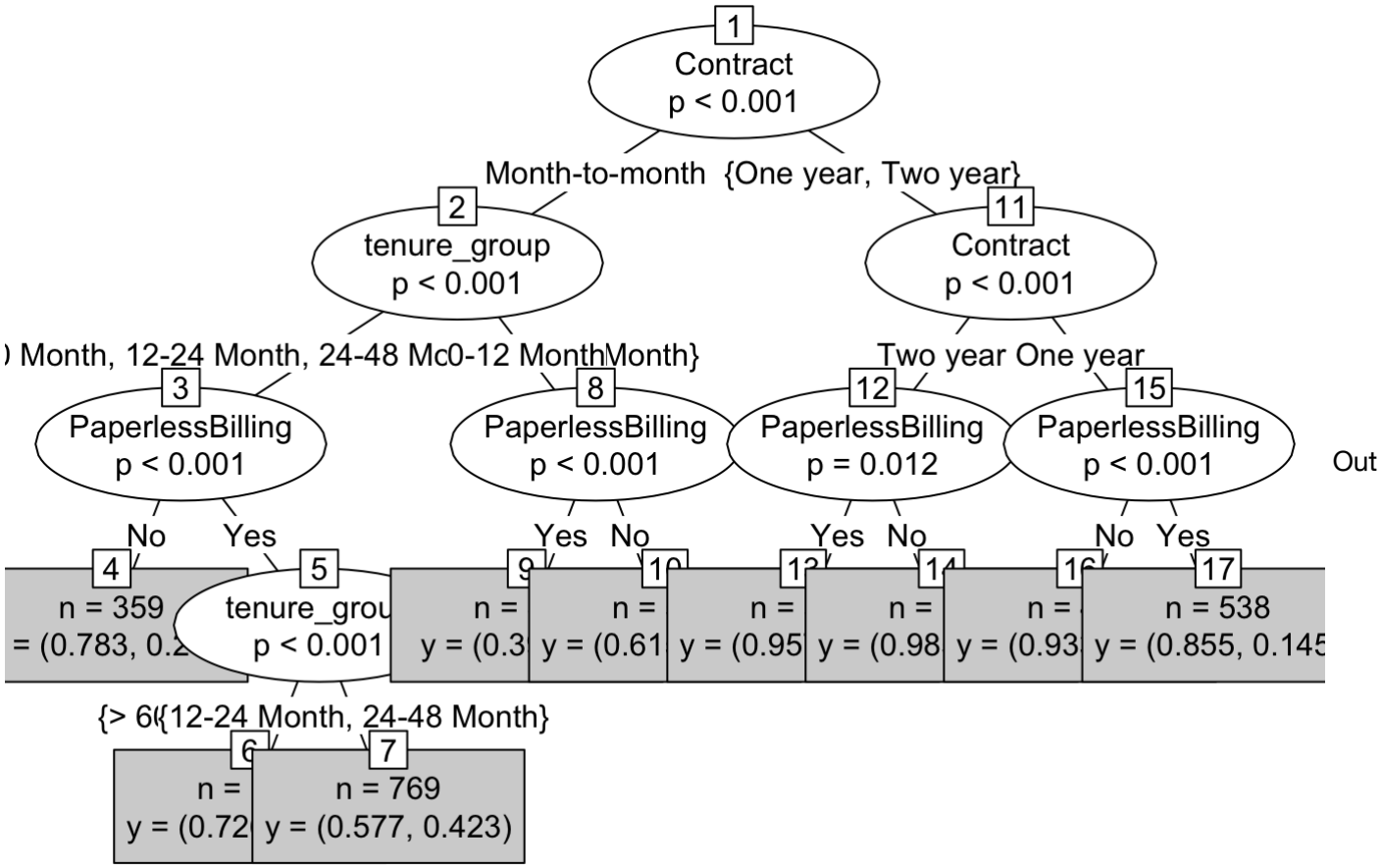
```
## [1] "Confusion Matrix for Logistic Regression"
```

```
table(testing$Churn, fitted.results > 0.5)
```

```
##
##     FALSE TRUE
##   0  1408  169
##   1   251  282
```

Decision Tree

```
tree <- ctree(Churn~Contract+tenure_group+PaperlessBilling, training)
plot(tree, type='simple')
```



of three variables we use, Contract is the most important variable to predict customer churn or not churn. If a customer in a one-year or two-year contract, no matter he (she) has PapelessBilling or not, he (she) is less likely to churn. On the other hand, if a customer is in a month-to-month contract, and in the tenure group of 0–12 month, and using PaperlessBilling, then this customer is more likely to churn.

Decision Tree Confusion Matrix

We are using all the variables to product confusion matrix table and make predictions.

```
pred_tree <- predict(tree, testing)
print("Confusion Matrix for Decision Tree"); table(Predicted = pred_tree, Actual = testi
ng$Churn)
```

```
## [1] "Confusion Matrix for Decision Tree"
```

```
##          Actual
## Predicted    0    1
##       No  1415  328
##       Yes  162  205
```

Decision tree Accuracy

```
p1 <- predict(tree, training)
tab1 <- table(Predicted = p1, Actual = training$Churn)
tab2 <- table(Predicted = pred_tree, Actual = testing$Churn)
print(paste('Decision Tree Accuracy',sum(diag(tab2))/sum(tab2)))
```

```
## [1] "Decision Tree Accuracy 0.767772511848341"
```

The accuracy for Decision Tree has hardly improved. Let’s see if we can do better using Random Forest.

Random Forest

Random Forest Initial Model

```
rfModel <- randomForest(Churn ~., data = training)
print(rfModel)
```

```
##
## Call:
##  randomForest(formula = Churn ~ ., data = training)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 21.27%
## Confusion matrix:
##       No Yes class.error
## No  3181 405   0.1129392
## Yes  642 694   0.4805389
```

The error rate is relatively low when predicting “No”, and the error rate is much higher when predicting “Yes”.

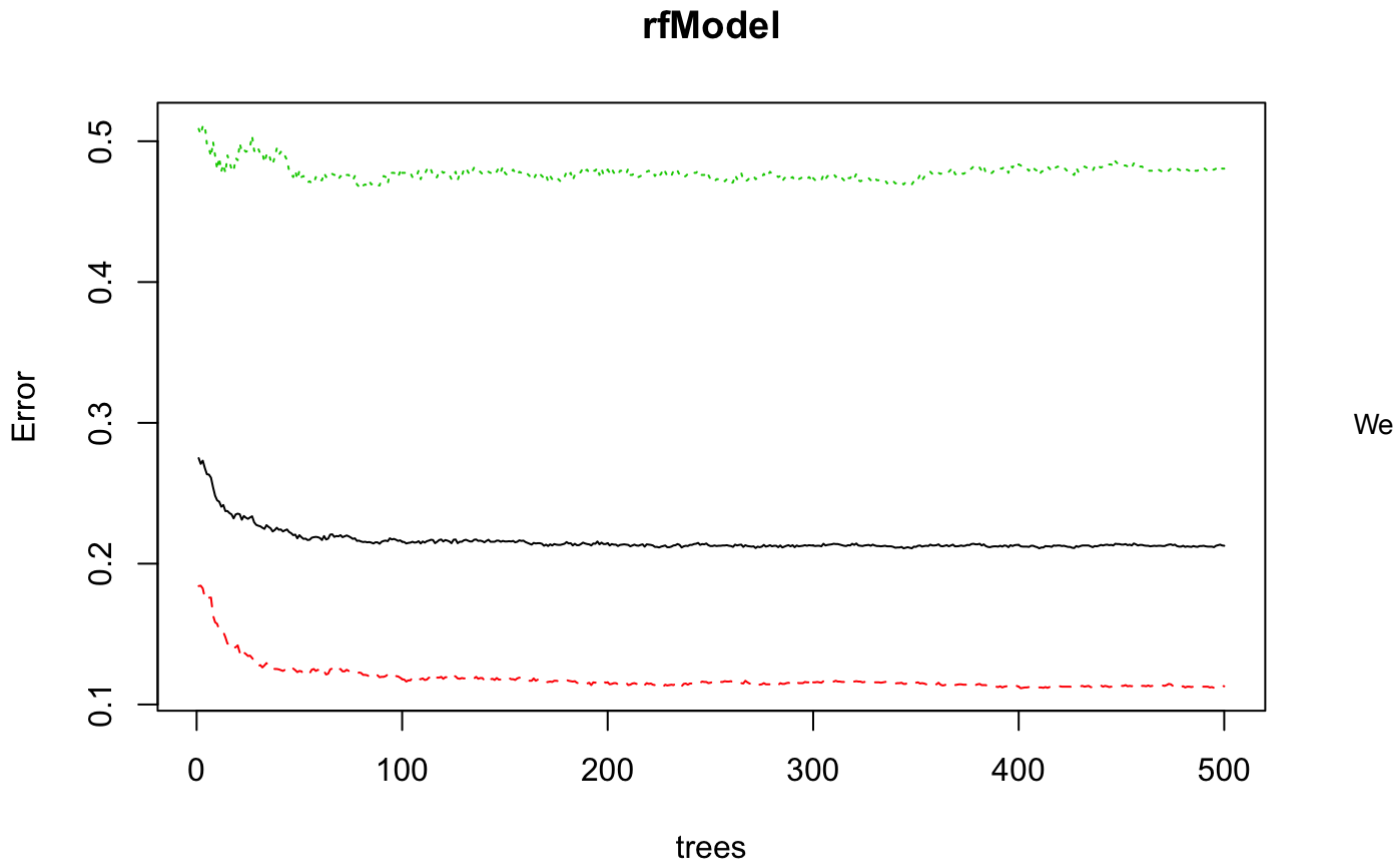
Random Forest Prediction and Confusion Matrix

```
pred_rf <- predict(rfModel, testing)
table(pred_rf, testing$Churn)
```

```
##
## pred_rf    0    1
##      No  1388  254
##      Yes  189  279
```

Random Forest Error Rate

```
plot(rfModel)
```

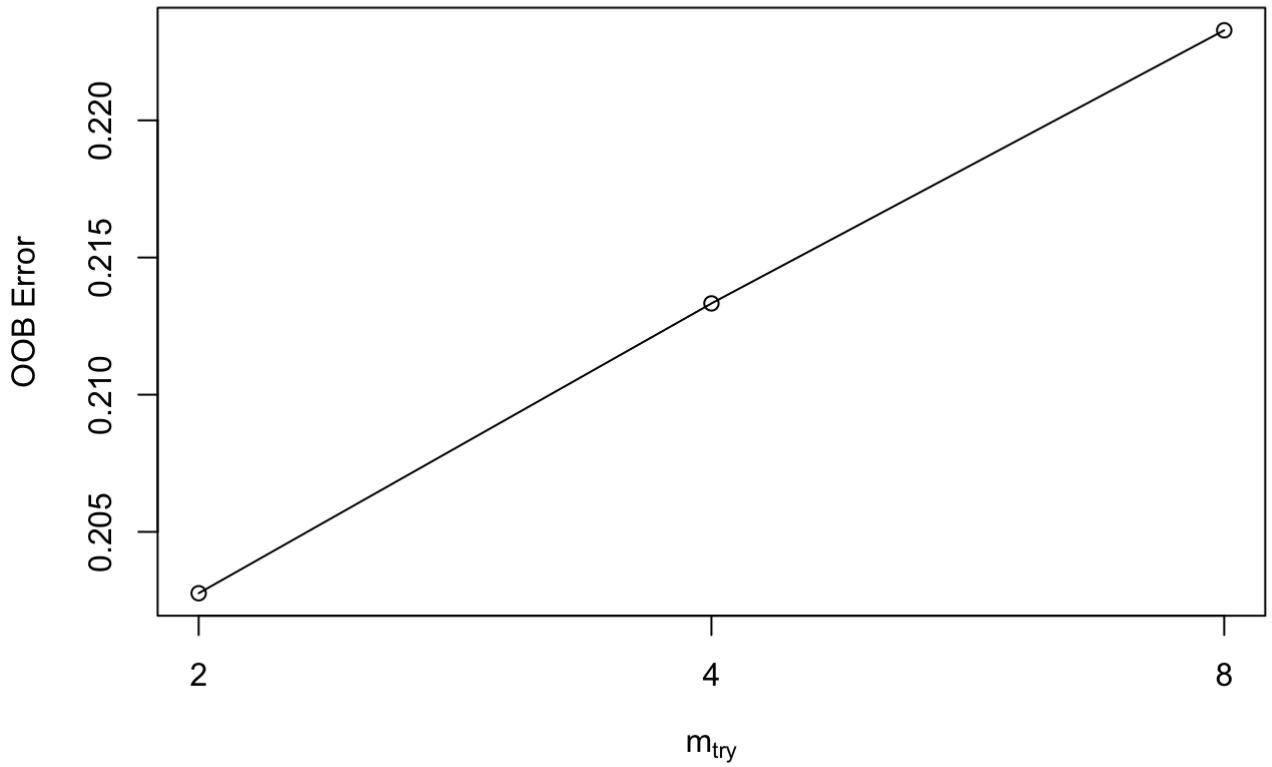


We use this plot to help us determine the number of trees. As the number of trees increases, the OOB error rate decreases, and then becomes almost constant. We are not able to decrease the OOB error rate after about 100 to 200 trees.

Tune Random Forest Model

```
t <- tuneRF(training[, -18], training[, 18], stepFactor = 0.5, plot = TRUE, ntreeTry = 2
00, trace = TRUE, improve = 0.05)
```

```
## mtry = 4  OOB error = 21.33%
## Searching left ...
## mtry = 8     OOB error = 22.33%
## -0.04666667 0.05
## Searching right ...
## mtry = 2     OOB error = 20.28%
## 0.04952381 0.05
```



We use this plot to give us some ideas on the number of mtry to choose. OOB error rate is at the lowest when mtry is 2. Therefore, we choose mtry=2.

Fit the Random Forest Model After Tuning

```
rfModel_new <- randomForest(Churn ~., data = training, ntree = 200, mtry = 2, importance
 = TRUE, proximity = TRUE)
print(rfModel_new)
```

```
##
## Call:
##  randomForest(formula = Churn ~ ., data = training, ntree = 200,      mtry = 2, impor
tance = TRUE, proximity = TRUE)
##                Type of random forest: classification
##                      Number of trees: 200
## No. of variables tried at each split: 2
##
##         OOB estimate of  error rate: 20.09%
## Confusion matrix:
##       No Yes class.error
## No  3265 321  0.08951478
## Yes  668 668  0.50000000
```

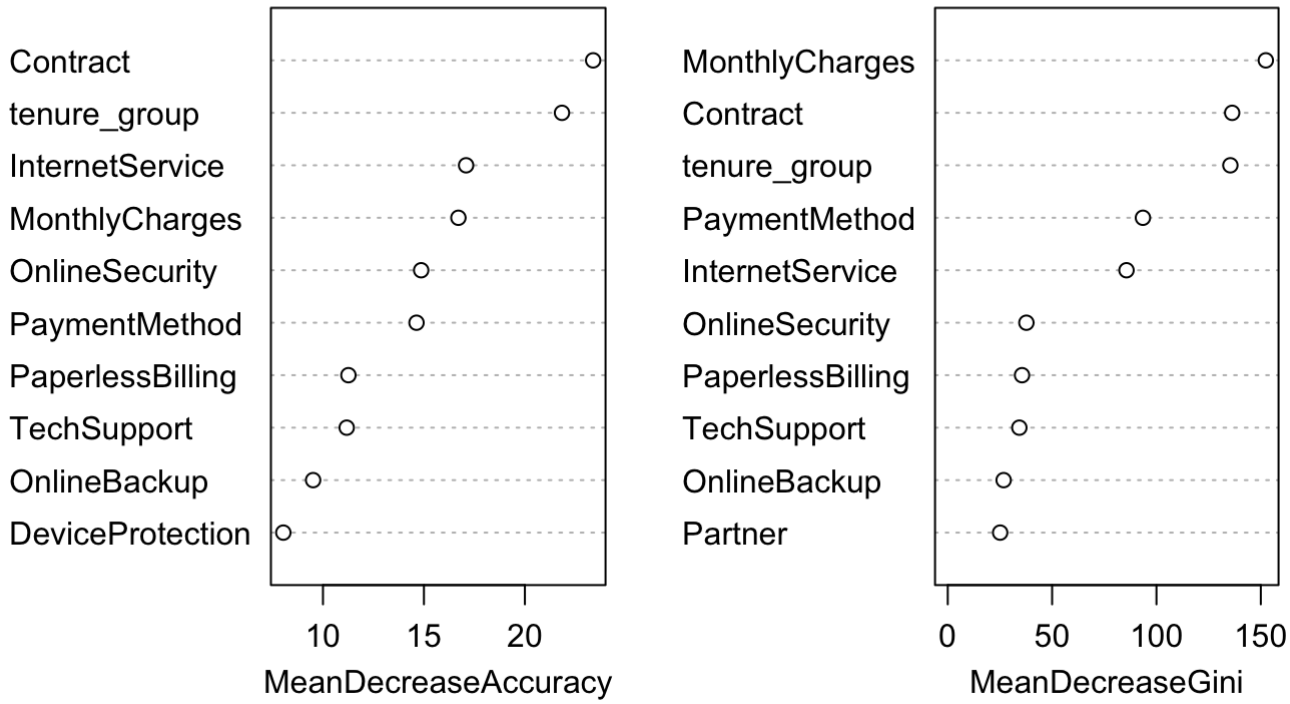
Random Forest Predictions and Confusion Matrix After Tuning

```
pred_rf_new <- predict(rfModel_new, testing)
table(pred_rf_new, testing$Churn)
```

```
##
## pred_rf_new    0    1
##         No  1421  284
##         Yes  156  249
```

Random Forest Feature Importance

```
varImpPlot(rfModel_new, sort=T, n.var = 10, main = 'Top 10 Feature Importance')
```



Conclusion

From the above analysis, we can see that Logistic Regression, Decision Tree and Random Forest can be used for customer churn analysis for this particular dataset equally fine.

Features such as tenure_group, Contract, PaperlessBilling, MonthlyCharges and InternetService appear to play a role in customer churn. There does not seem to be a relationship between gender and churn. Customers in a month-to-month contract, with PaperlessBilling and are within 12 months tenure, are more likely to churn; On the other hand, customers with one or two year contract, with longer than 12 months tenure, that are not using PaperlessBilling, are less likely to churn.