BAN404 Exam

Candidate: 85

2023-05-15

Libraries

```
library(tidyverse)
library(ggplot2)
library(boot)
library(tree)
library(caret)
library(glmnet)
library(randomForest)
library(gbm)
library(MASS)
```

Task 1

a)

```
# Loading data
churn_df <- read.csv("Churn.csv")

# Checking classes
sapply(churn_df, class)
```

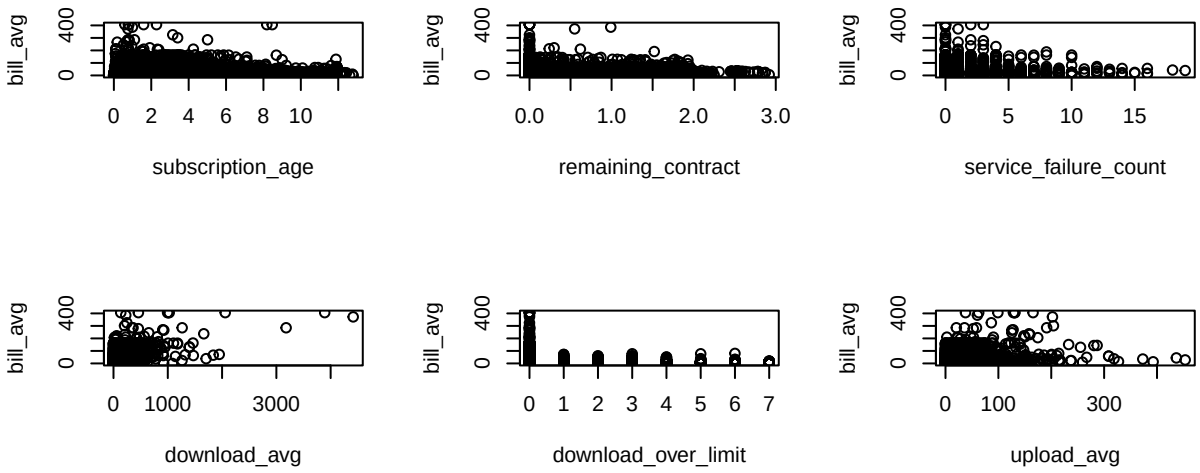
```
##                           id            is_tv_subscriber
##                    "integer"                   "integer"
## is_movie_package_subscriber           subscription_age
##                    "integer"                   "numeric"
##                     bill_avg         remaining_contract
##                    "integer"                   "numeric"
##       service_failure_count               download_avg
##                    "integer"                   "numeric"
##                   upload_avg        download_over_limit
##                    "numeric"                   "integer"
##                        churn
##                    "integer"
```

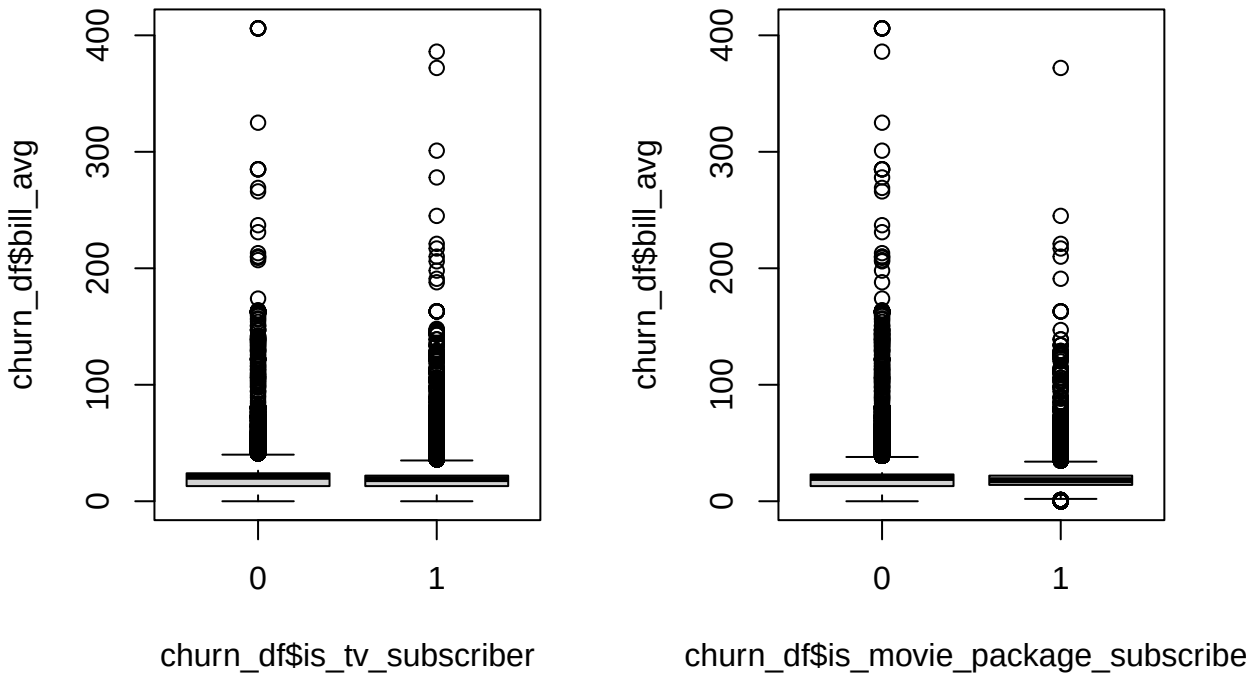
```
# Removing ID
churn_df <- subset(churn_df, select = -id)
```

```r
# Converting to factor
churn_df$churn <- factor(churn_df$churn)
churn_df$is_tv_subscriber <- factor(churn_df$is_tv_subscriber)
churn_df$is_movie_package_subscriber <- factor(churn_df$is_movie_package_subscriber)

n <- nrow(churn_df)
# Splitting into train and test data
set.seed(65923764)
ind <- sample(1:n, size = floor(n/2))
train <- churn_df[ind, ]
test <- churn_df[-ind, ]
```

I decided to remove the ID since this will not contribute to the analysis. Also, i chose to not remove any other variables. I decided to encode the is_tv_subscriber, is_movie_package_subscriber and churn as factors.

b)






```
##
##     0     1     2     3     4     5     6     7
## 67992   766   560   498   456   429   688   504
```

For this task i chose to plot a scatterplots for each variable against the average bill, and boxplots for the variables with 0 and 1. By looking at the scatterplots, it seems like service_failure_count have some trend where customers with higher service_failure_count have a lower average bill. Download_avg and upload_avg also seems like interesting predictors, as higher download and upload seems to yield higher bill averages. It seems sensible that customers who use more data pays more. The download over limit predictor seems interesting in the fact that customers with no downloads over limit have higher bill averages,although there is clearly a lot more observations in this category. When looking at the boxplots it is not too clear which of the predictors that have a clear impact on the bill average.

c)

```r
# Linear regression
linreg <- lm(bill_avg ~ ., data = train)
summary(linreg)
```

```
##
## Call:
## lm(formula = bill_avg ~ ., data = train)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -138.70   -5.29   -0.02    4.32  343.59
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   18.810974   0.237211  79.301  < 2e-16 ***
## is_tv_subscriber1             -3.934575   0.167074 -23.550  < 2e-16 ***
## is_movie_package_subscriber1 -0.578824   0.140322  -4.125 3.72e-05 ***
## subscription_age               0.292657   0.030090   9.726  < 2e-16 ***
## remaining_contract            -2.559986   0.125748 -20.358  < 2e-16 ***
## service_failure_count          0.844834   0.072264  11.691  < 2e-16 ***
## download_avg                   0.079358   0.001107  71.690  < 2e-16 ***
## upload_avg                     0.196993   0.007761  25.383  < 2e-16 ***
## download_over_limit           -2.823693   0.060024 -47.043  < 2e-16 ***
## churn1                         0.442278   0.173201   2.554   0.0107 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.18 on 35936 degrees of freedom
## Multiple R-squared:  0.306,  Adjusted R-squared:  0.3058
## F-statistic:  1760 on 9 and 35936 DF,  p-value: < 2.2e-16
```

```r
lm_pred <- predict(linreg, newdata = test)

mean((test$bill_avg-lm_pred)^2)
```

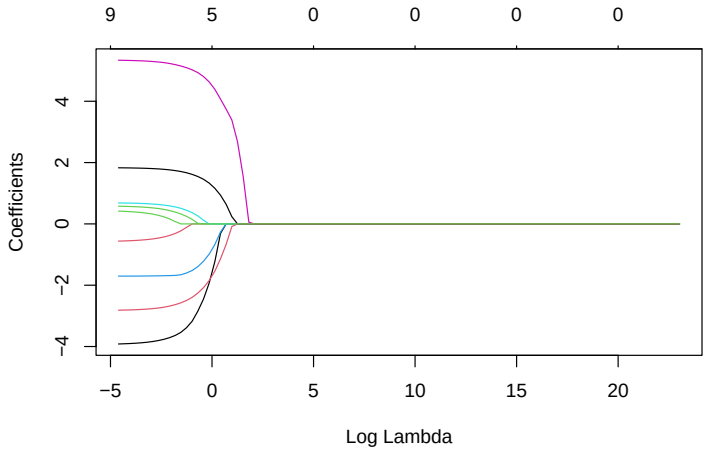
```
## [1] 125.163
```

Most of the predictor are significant, and thus have an impact on the bill_avg. When predicting on the test data i get a test MSE at 125 with all the variables used in the regression. The adjusted r-squared shows that 30% of the variance in the response variable can be explained by the predictor variables

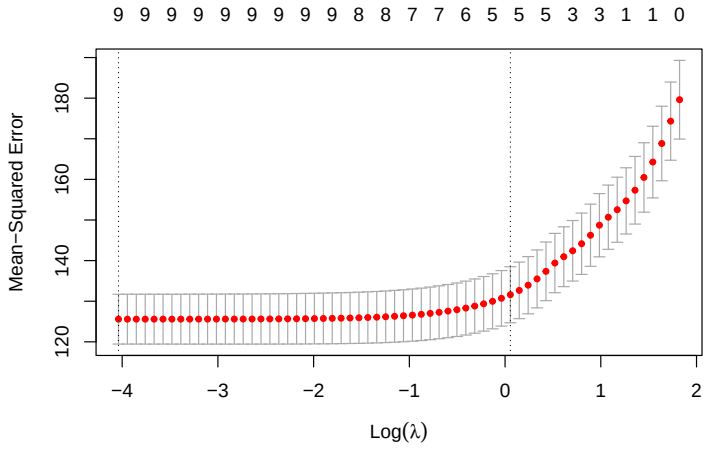
d)

```r
# Lasso regression

# Identify numeric columns
numeric_cols <- sapply(train, is.numeric)

# Scale numeric columns in train
train_scaled <- train
train_scaled[, numeric_cols] <- scale(train_scaled[, numeric_cols])

# Scale numeric columns in test
test_scaled <- test
test_scaled[, numeric_cols] <- scale(test_scaled[, numeric_cols])

# Making a matrix for train and test data
x_train <- model.matrix(bill_avg ~ ., data = train_scaled)
y_train <- train$bill_avg
```

```r
# Matrix for test data <
x_test <- model.matrix(bill_avg ~ ., data = test_scaled)
y_test <- test$bill_avg

grid <- 10^seq(10, -2, length = 100)
# Creating lasso model
lasso_mod <- glmnet(x_train, y_train, alpha = 1, lambda = grid)
plot(lasso_mod, xvar = "lambda")
```




```r
# Cross validation
lasso_cv <- cv.glmnet(x_train, y_train, alpha = 1, maxit = 500000)

plot(lasso_cv)
```




```r
# Best lambda value
bestlam <- lasso_cv$lambda.min
bestlam
```

```
## [1] 0.01762788
```

```
coef(lasso_mod, s=lasso_cv$lambda.min)
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                                       s1
## (Intercept)                   22.1756564
## (Intercept)                    .
## is_tv_subscriber1             -3.9024127
## is_movie_package_subscriber1 -0.5490424
## subscription_age               0.5672305
## remaining_contract            -1.7027285
## service_failure_count          0.6780999
## download_avg                   5.3307077
## upload_avg                     1.8251694
## download_over_limit           -2.8059861
## churn1                         0.3994540
```

By using cross validation and computing a grid of lambda values i get a best lambda value at 0.01762788. The output of the coefficients shows that all the variables in the lasso regression are used. I scaled all the numerical variables with a mean around 0 and standardeviation equal 1, the reason for standardization is because the lasso regression penalizes the size of each variables associated with each variable (Bhalla, n.d). Thus variables with a larger scale will have a bigger impact than smaller scales.

e)

```
# Fitting the model on test data
lasso_pred <- predict(lasso_mod, s = bestlam,
                      newx = x_test)

# Test mse
mean((y_test - lasso_pred)^2)
```

```
## [1] 125.7713
```

The lasso regression is quite similar to the linear regression with all predictors. Both regressions yield a test MSE around 125.

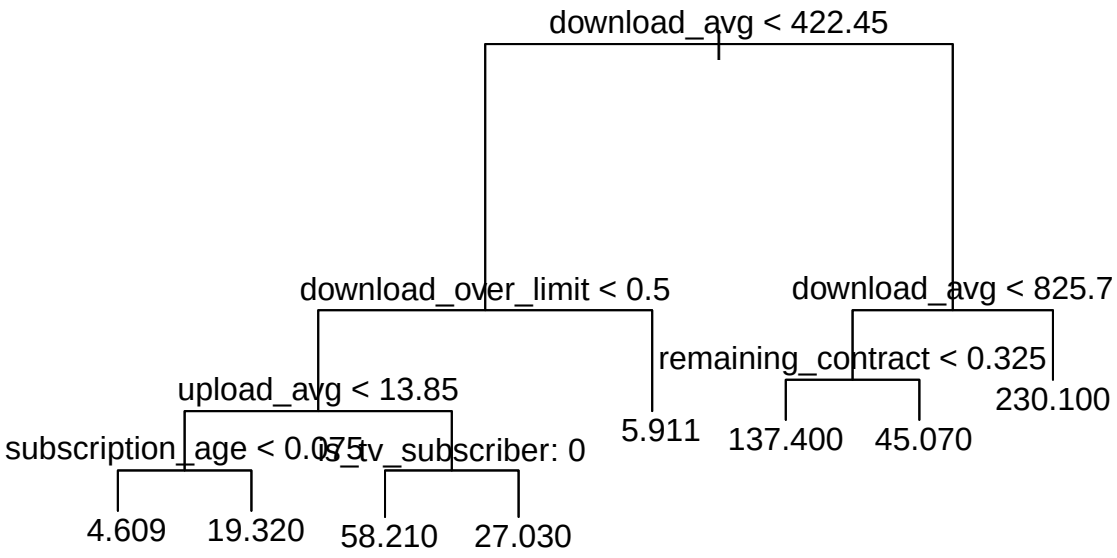
f)

```
# Model
tree_mod <- tree(bill_avg~ ., data = train)
summary(tree_mod)
```

```
##
## Regression tree:
## tree(formula = bill_avg ~ ., data = train)
## Variables actually used in tree construction:
```

```
## [1] "download_avg"        "download_over_limit" "upload_avg"
## [4] "subscription_age"    "is_tv_subscriber"    "remaining_contract"
## Number of terminal nodes:  8
## Residual mean deviance:  120.3 = 4325000 / 35940
## Distribution of residuals:
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -174.1000   -5.9110   -0.3173    0.0000    3.6830  359.0000
```

```
# Plotted tree
plot(tree_mod)
text(tree_mod, pretty = 0)
```



For the regression tree i used all the variables. The plotted tree shows that the most important attribute is the download average for predicting bill_avg. If the download average is less than 422.45, the tree splits into a branch to the left to an internal node where the download over limit is the next split. If the value is less than 0.5, it gets sent to the left, and right if its above. Further down the tree is using upload average, subscription age and is_tv_subscriber for splitting. The leaf nodes at the end shows the output values for bill_avg. Overall the most important variables is the the download average, download over limit, upload_avg, subscription_age, is_tv_subscriber and remaining contract.

g)

```
# Prediction
tree_pred <- predict(tree_mod, newdata = test)
```

```r
# Test MSE
mean((test$bill_avg-tree_pred)^2)
```

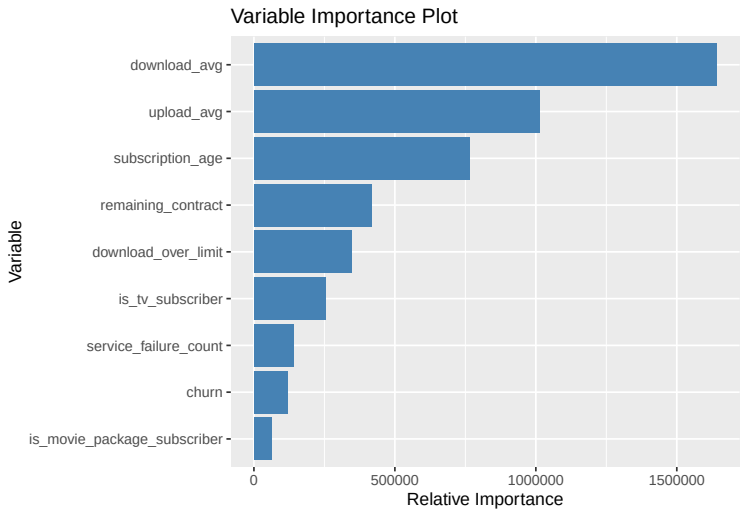
```
## [1] 134.7101
```

The regression tree yields a higher test MSE than the linear regression model, with a testMSE at 134.71.

h)

```r
# Model
rf_mod <- randomForest(bill_avg ~ ., data = train, ntree = 50)
```

Variable importance plot



I decided to reduce the number of trees to 50 because of the computational resources it takes to run more trees. The variable importance plot outlines the download avg as the most important feature, along with the upload avg ang subscription age, which is quite similar to the most important attribute selection as the regression tree. The feauture importance is based on the incNodePurity, which is a measure of the total decrease in the node impurity averaged out of all the trees. For regression trees the measure used is the Residual Sum of Squares. Variables with low nodeIncPurity do not contribute much to decrease the RSS, while variables with a high incNodePurity impacts the model more in terms of reducing the RSS.

i)

```r
# Prediction
rf_pred <- predict(rf_mod, newdata = test)

# Test MSE
mean((test$bill_avg - rf_pred)^2)
```

```
## [1] 104.0858
```

The random forest model is performing better than the previous models with a test MSE at 104. The model could probably do better with a higher number of trees.

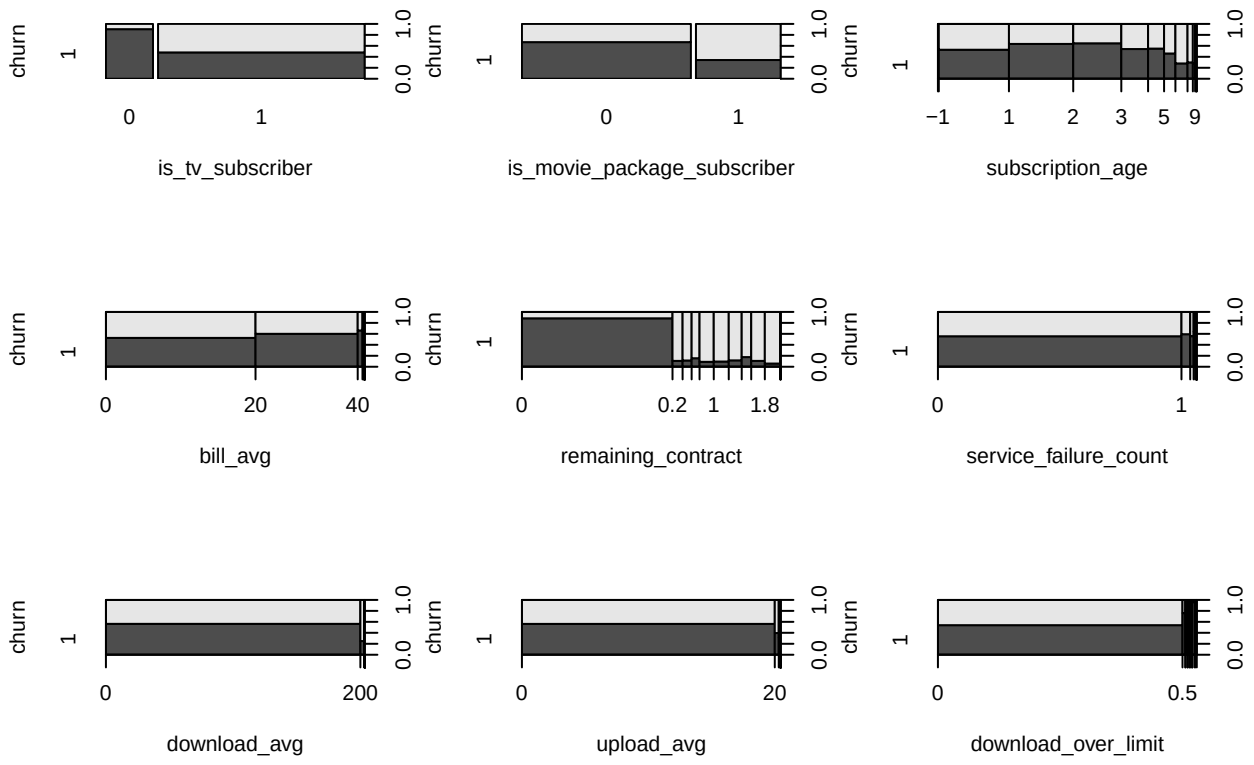
j)

Based on the various models computed, the average downloads in GB for each customer seems to be the best predictor for the customers bill average. If the downloads in gb is higher for a customer, the bill average is likely to be higher. Both the regression tree and the random forest ended up with this feature as the most important. For the linear regression, the coeffiecient for download avg is at 0.079358, meaning that an one unit increase in average downloads for a customer, will increase the bill_avg by 0.079. When using the lasso regression with standardized variables, the download average is also considered the most important with a coefficient at 5.3307077, meaning that an one unit increase in this variable increases the bill_avg with 5.33 units.

Task 2

a)

```
par(mfrow = c(3,3))
plot(churn ~ ., data = train)
```



Customers who are tv_subscriber seems less likely to churn than those who arent, the same with is movie subscriber. Based of the plots it seems that the remaining contract is a strong predictor for churn, where less time left of subscription have a higher churn rate.

b)

```
# 50 first observations
first_obs <- 1:50
churn_sample <- churn_df[first_obs,]


# Converting to numeric for bootstrap
churn_sample$churn <- as.numeric(as.character(churn_sample$churn))
# Bootstrap function with boot package
set.seed(1)
boot_func <- function(data, index) {
  return(mean(data[index]))
}

# Bootstrap with boot function
bootstrap <- boot(churn_sample$churn, boot_func, R = 10)
bootstrap
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = churn_sample$churn, statistic = boot_func, R = 10)
##
##
## Bootstrap Statistics :
##     original  bias    std. error
## t1*      0.8  -0.012  0.06545567
```

```
# 95% confidence interval
conf_interval <- quantile(bootstrap$t, c(0.025, 0.975))
conf_interval
```

```
##   2.5%  97.5%
## 0.6845 0.8755
```

The estimated probability of churn is 0.8, wit a low bias, indicating that the bootstap estimate is close to the original estimate.

c)

```
# Still using the train data
logreg <- glm(churn ~ ., data = train, family = "binomial")
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(logreg)
```

```
##
## Call:
## glm(formula = churn ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.3996  -0.4065   0.1928   0.5371   4.4331
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   4.2082143  0.0766005  54.937  < 2e-16 ***
## is_tv_subscriber1            -1.7579129  0.0644852 -27.261  < 2e-16 ***
## is_movie_package_subscriber1 -0.0753291  0.0360696  -2.088   0.0368 *
## subscription_age             -0.2550886  0.0084155 -30.312  < 2e-16 ***
## bill_avg                      0.0007931  0.0015329   0.517   0.6049
## remaining_contract           -3.1868860  0.0367033 -86.828  < 2e-16 ***
## service_failure_count         0.1555337  0.0218436   7.120 1.08e-12 ***
## download_avg                 -0.0114923  0.0004305 -26.695  < 2e-16 ***
## upload_avg                    0.0022726  0.0021780   1.043   0.2967
## download_over_limit           0.4807050  0.0404724  11.877  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 49338  on 35945  degrees of freedom
## Residual deviance: 24524  on 35936  degrees of freedom
## AIC: 24544
##
## Number of Fisher Scoring iterations: 6
```

The coefficients for is_tv_subscriber is at -1.7579129. Taking the log odds of this coefficient exp(-1.7579129) = 0.1724043, shows that a customer that is tv_subscriber is 0.17 times less likely to churn than the customers who are not. When doing the same calculation for is_movie_package_subscriber with a coefficient at -0.0753291, i get a log odds at 0.9274382. Which shows that customers that are movie_package_subscribers are 0.927 times less likely to churn than customers who is not. Overall, both of these variables have an negative impact on the churn, meaning that customers that either have tv or movie subscriptions are less probable to churn than customers who do not subscribe.

d)

Model with all variables

```
logreg_pred <- predict(logreg, newdata = test, type = "response")

# Storing predictions as factors
predicted <- factor(ifelse(logreg_pred >0.5, 1,0))
true_value <- factor(test$churn)
```

```r
prop.table(table(predicted, test$churn),margin = 1)
```

```
##
## predicted         0         1
##         0 0.8879847 0.1120153
##         1 0.1401238 0.8598762
```

```r
# Confusion matrix with caret package
confusionMatrix(predicted, true_value, positive = "1")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 12985  1638
##          1  2988 18336
##
##                Accuracy : 0.8713
##                  95% CI : (0.8678, 0.8748)
##     No Information Rate : 0.5557
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7372
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9180
##             Specificity : 0.8129
##          Pos Pred Value : 0.8599
##          Neg Pred Value : 0.8880
##              Prevalence : 0.5557
##          Detection Rate : 0.5101
##    Detection Prevalence : 0.5932
##       Balanced Accuracy : 0.8655
##
##        'Positive' Class : 1
##
```

Model with removed variables

```r
# Removing variables bill_avg and upload_avg
logreg2 <- glm(churn ~ .-bill_avg -upload_avg, data = train, family="binomial")
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```r
# Predicting
logreg_pred2 <- predict(logreg2, newdata = test, type = "response")

# Storing the predicted values as factor
predicted2 <- factor(ifelse(logreg_pred2 > 0.5, 1,0))
```

```r
# prop table
prop.table(table(predicted2, test$churn),margin = 1)
```

```
##
## predicted2         0         1
##          0 0.8879623 0.1120377
##          1 0.1399287 0.8600713
```

```r
# Confusion matrix
confusionMatrix(predicted2, true_value, positive = "1")
```

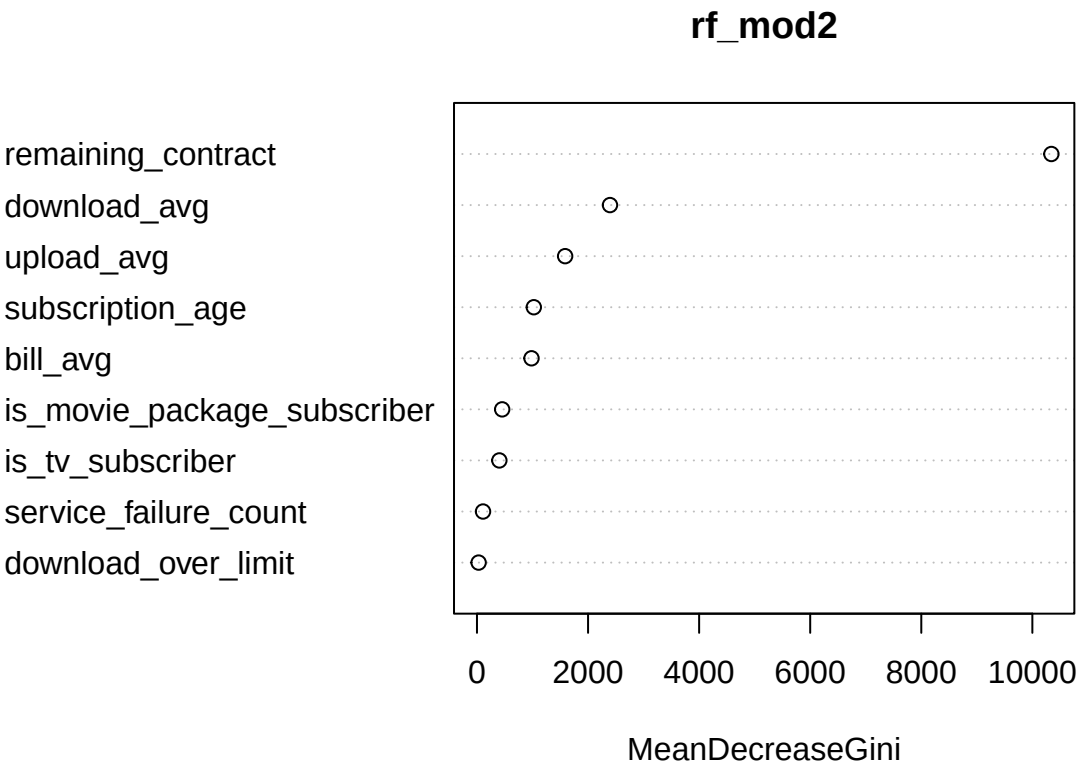
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 12990  1639
##          1  2983 18335
##
##                Accuracy : 0.8714
##                  95% CI : (0.8679, 0.8749)
##     No Information Rate : 0.5557
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.7374
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9179
##             Specificity : 0.8132
##          Pos Pred Value : 0.8601
##          Neg Pred Value : 0.8880
##              Prevalence : 0.5557
##          Detection Rate : 0.5101
##    Detection Prevalence : 0.5930
##       Balanced Accuracy : 0.8656
##
##        'Positive' Class : 1
##
```

The model with removing non-significant variables like bill_avg and upload_avg does not seem to improve the accuracy by a lot. The overall test accuracy for the model with all variables is 87.13%, while the model with removed variables has an test accuracy at 87.14%. Overall the model are good at predicting whether a customer will churn or not. The proportion of the churn in the churn df is 55%, and a dumb model which only predicts 1 would then yield an accuracy at 55%. The sensitivity shows how good the model is at predicting the True Positive Rate. Both of these models have a sensitivity at 0.91, meaning that the model is predicting churn 91% correctly out of all the actual churned customers, the specificity shows the models ability to predict the True Negative Rate, which for both models is 81%. The model is a little bit worse at predicting customers who did not churn out of all the churned customers.

e)

Random Forest model for predicting churn

```
train$churn <- factor(train$churn)
#Fitting the model
rf_mod2 <- randomForest(churn ~ ., data = train, ntree = 50)

# Variable importance
varImpPlot(rf_mod2)
```




```
# Predicting
rf_pred2 <- predict(rf_mod2, newdata = test)

# Storing prediction as a factor for the confusion matrix
predicted3 <- factor(rf_pred2)

# Confusion matrix with caret package
confusionMatrix(predicted3,true_value)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 15038  1230
##          1   935 18744
##
##                Accuracy : 0.9398
##                  95% CI : (0.9373, 0.9422)
```

```
##     No Information Rate : 0.5557
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8783
##
##  Mcnemar's Test P-Value : 2.64e-10
##
##             Sensitivity : 0.9415
##             Specificity : 0.9384
##          Pos Pred Value : 0.9244
##          Neg Pred Value : 0.9525
##              Prevalence : 0.4443
##          Detection Rate : 0.4183
##    Detection Prevalence : 0.4526
##       Balanced Accuracy : 0.9399
##
##        'Positive' Class : 0
##
```

The random forest model with 50 trees performs better than the logistic regression, with a test accuracy at 93.9%. The model has a significant increase in correctly predicted negatives, or customers who did not churn.

f)

Based on the output from the logistic regression, bill_avg and upload_avg have p-values larger than 0.05, thus we cannot conclude that these variables have an significant impact on the churn. The logistic regression model did not perform worse by removing these variables. Interestingly, when computing the variable importance plot for the Random forest mode, the remaining contract variable is the most important feature in the model. This seems rational, since customers who is soon to end their subscription plan may have a bigger probability of churning. Typical features of customers that is churning may be that their subscription soon runs out and also have a high GB usage of data.

References

Bhalla, D (n.d).WHEN AND WHY TO STANDARDIZE A VARIABLE. Retrieved from listendata.com. https://www.listendata.com/2017/04/how-to-standardize-variable-in-regression.html