Telco Customer Churn Analysis

Load required libraries

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(caret)
```

```
## Loading required package: lattice
```

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-8
```

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(corrplot)
```

```
## corrplot 0.95 loaded
```

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
library(randomForest)
```

```
## randomForest 4.7-1.2
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(gbm)
```

```
## Loaded gbm 2.2.2
```

```
## This version of gbm is no longer under development. Consider transitioning to gbm3, https://gi
thub.com/gbm-developers/gbm3
```

```
library(xgboost)
```

```
##
## Attaching package: 'xgboost'
```

```
## The following object is masked from 'package:dplyr':
##
##     slice
```

```
library(cluster)
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

Load and merge data

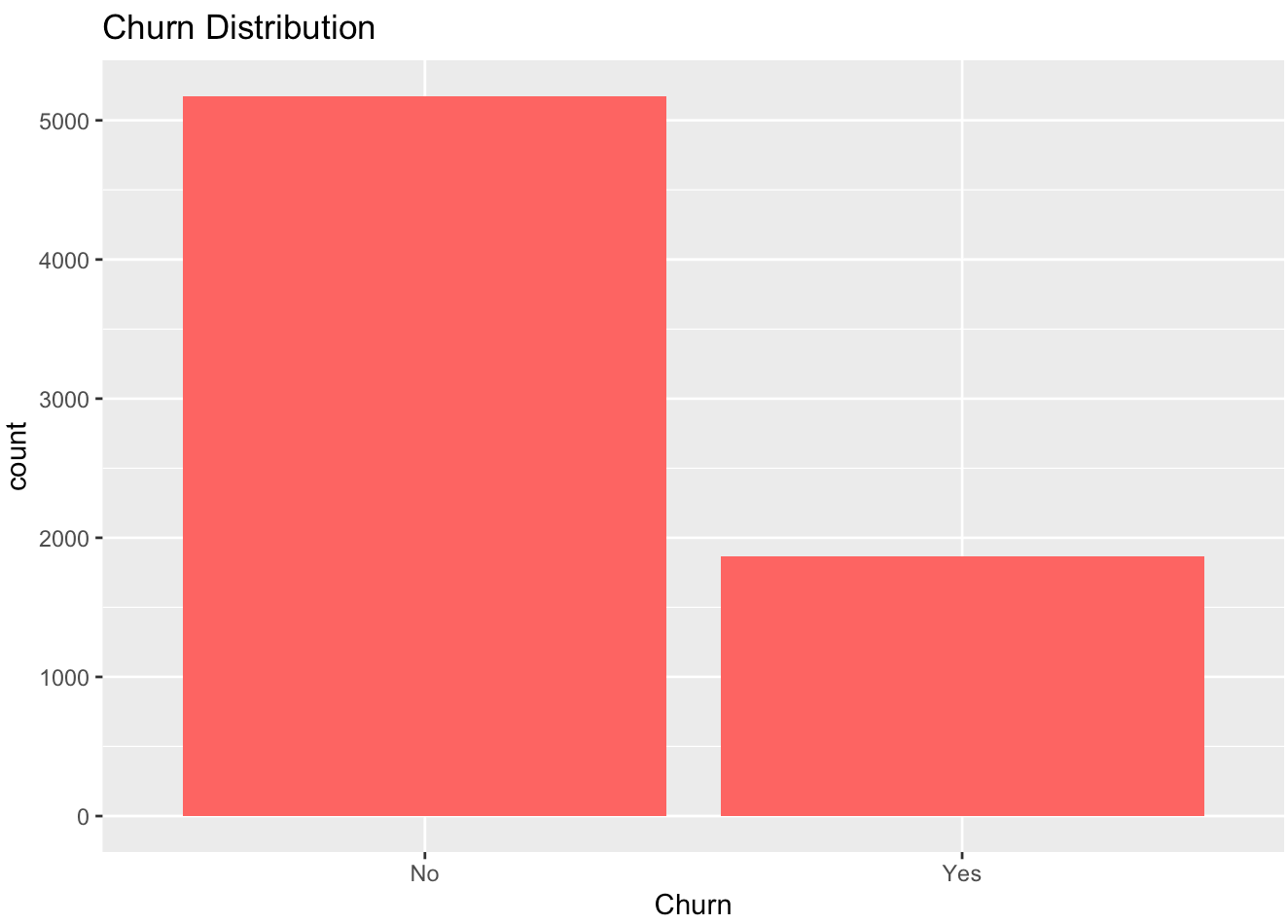
```
status <- read.csv("Telco_customer_churn_status.csv")
demographics <- read.csv("Telco_customer_churn_demographics.csv")
location <- read.csv("Telco_customer_churn_location.csv")
services <- read.csv("Telco_customer_churn_services.csv")

data <- status %>%
  left_join(demographics, by = "Customer.ID") %>%
  left_join(location, by = "Customer.ID") %>%
  left_join(services, by = "Customer.ID")
```

Select and clean variables

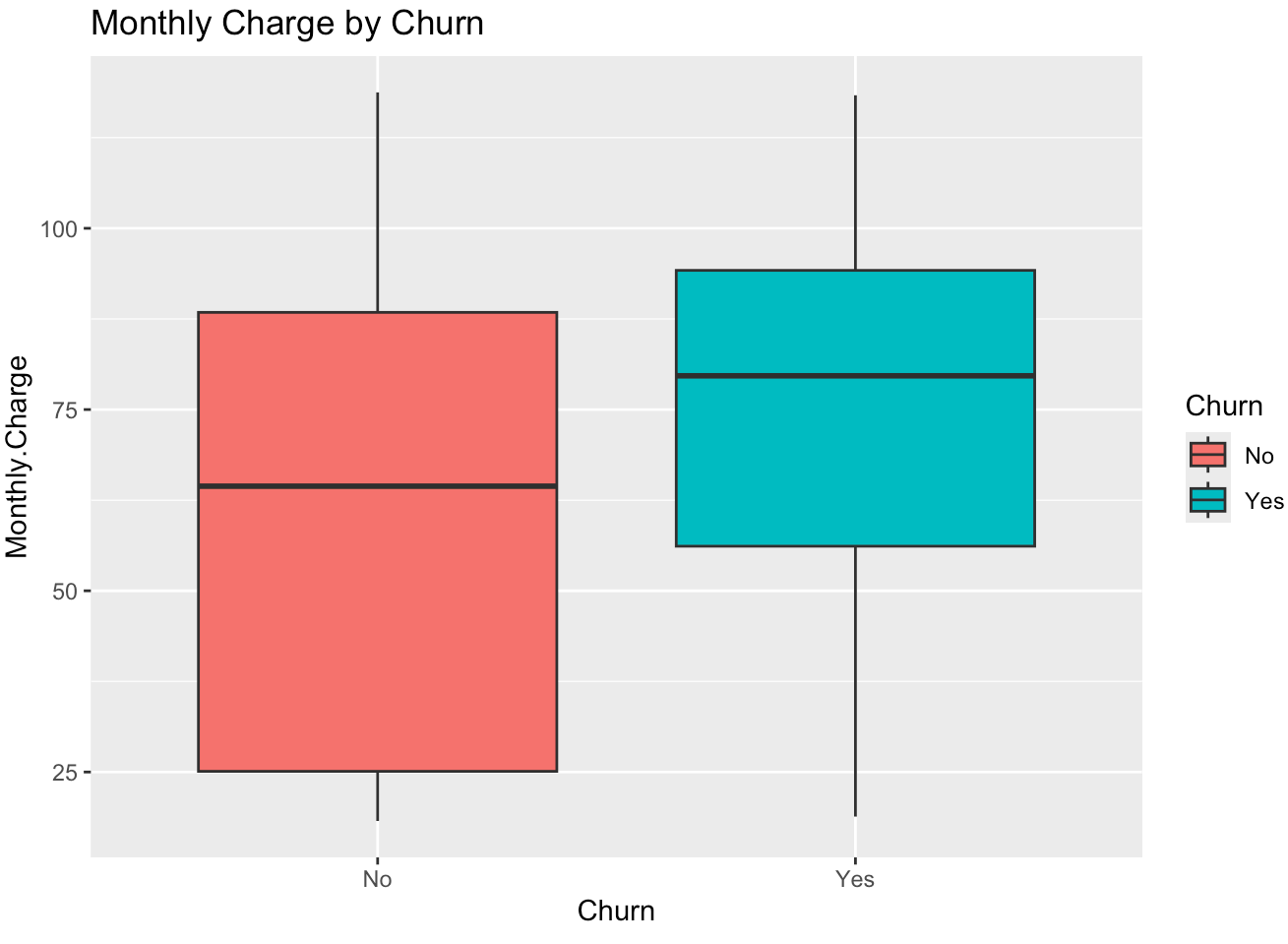
```
selected_vars <- c("Churn.Label", "Gender", "Senior.Citizen", "Married", "Dependents",
                   "Number.of.Dependents", "Tenure.in.Months", "Contract",
                   "Phone.Service", "Multiple.Lines", "Internet.Service",
                   "Online.Security", "Online.Backup", "Device.Protection.Plan",
                   "Premium.Tech.Support", "Streaming.TV", "Streaming.Movies",
                   "Paperless.Billing", "Payment.Method", "Monthly.Charge",
                   "Total.Charges", "CLTV", "Satisfaction.Score")

data_model <- data %>%
  dplyr::select(all_of(selected_vars)) %>%
  na.omit() %>%
  mutate(Churn = as.factor(Churn.Label)) %>%
  dplyr::select(-Churn.Label)
```

Exploratory Data Analysis

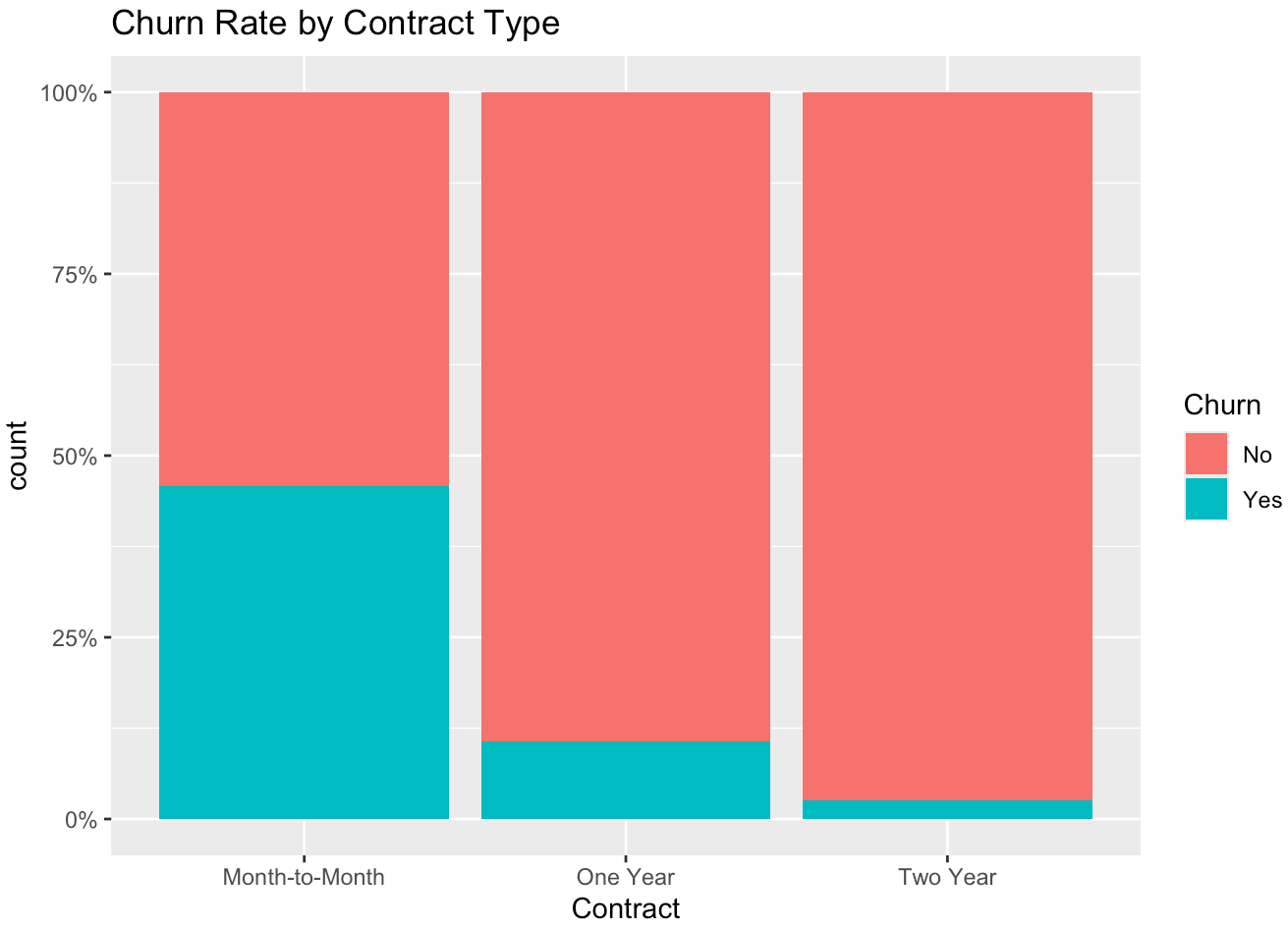
```
ggplot(data_model, aes(x = Churn)) + geom_bar(fill = "#FF6666") + labs(title = "Churn Distributio
n")
```



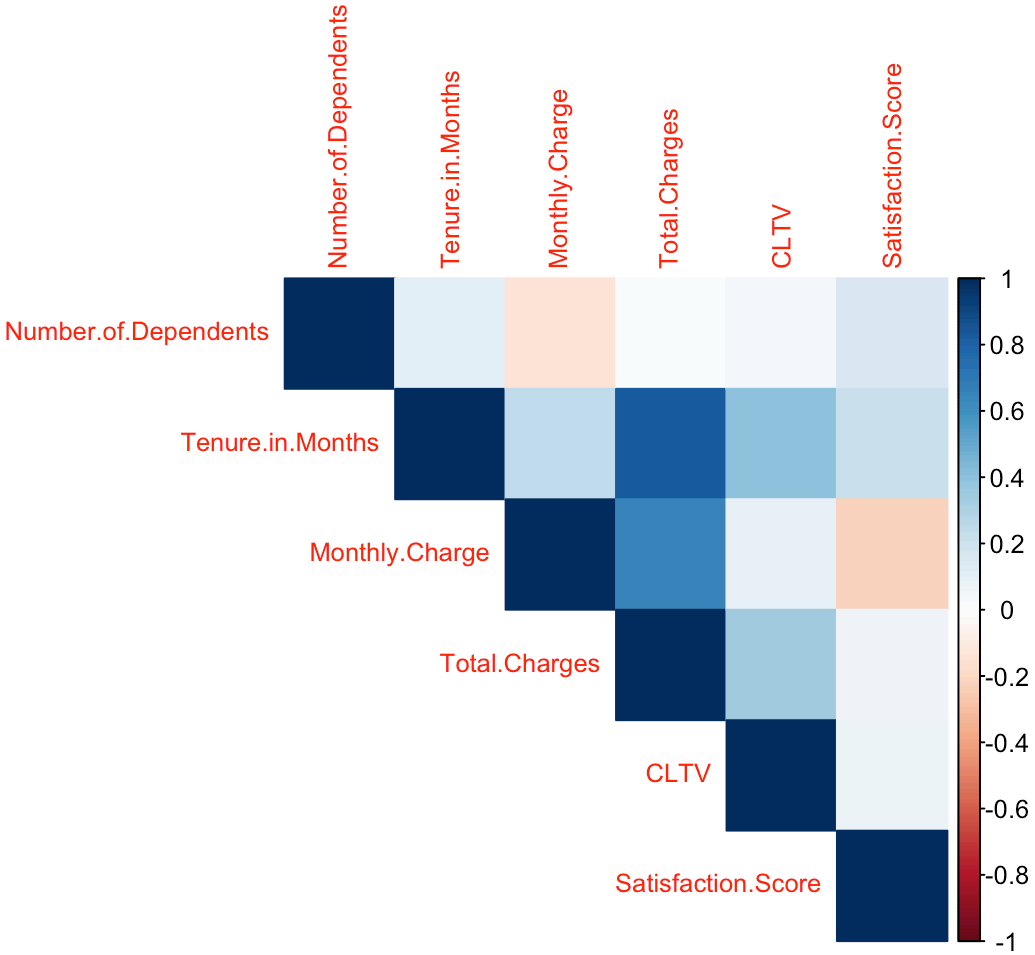

```
ggplot(data_model, aes(x = Churn, y = Monthly.Charge, fill = Churn)) +
  geom_boxplot() + labs(title = "Monthly Charge by Churn")
```



```
ggplot(data_model, aes(x = Contract, fill = Churn)) +
  geom_bar(position = "fill") + scale_y_continuous(labels = scales::percent) +
  labs(title = "Churn Rate by Contract Type")
```




```
num_vars <- data_model[, sapply(data_model, is.numeric)]
corrplot(cor(num_vars), method = "color", type = "upper", tl.cex = 0.8)
```



Train/test split and preprocessing

```
set.seed(42)
split <- createDataPartition(data_model$Churn, p = 0.8, list = FALSE)
train <- data_model[split, ]
test  <- data_model[-split, ]

train <- train %>% mutate(across(where(is.character), as.factor))
test  <- test %>% mutate(across(where(is.character), as.factor))

valid_factors <- sapply(train, function(x) {
  if (is.factor(x)) nlevels(x) > 1 else TRUE
})
train <- train[, valid_factors]
test <- test[, names(train)]

nzv <- nearZeroVar(train, saveMetrics = TRUE)
train <- train[, !nzv$zeroVar]
test <- test[, names(train)]
```

Stepwise Logistic Regression

```
full_model <- glm(Churn ~ ., data = train, family = "binomial")
```

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

```
step_model <- stepAIC(full_model, direction = "both", trace = FALSE)
```

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

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

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

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

```
step_pred_prob <- predict(step_model, newdata = test, type = "response")
step_pred <- ifelse(step_pred_prob > 0.5, "Yes", "No")
confusionMatrix(as.factor(step_pred), test$Churn)
```

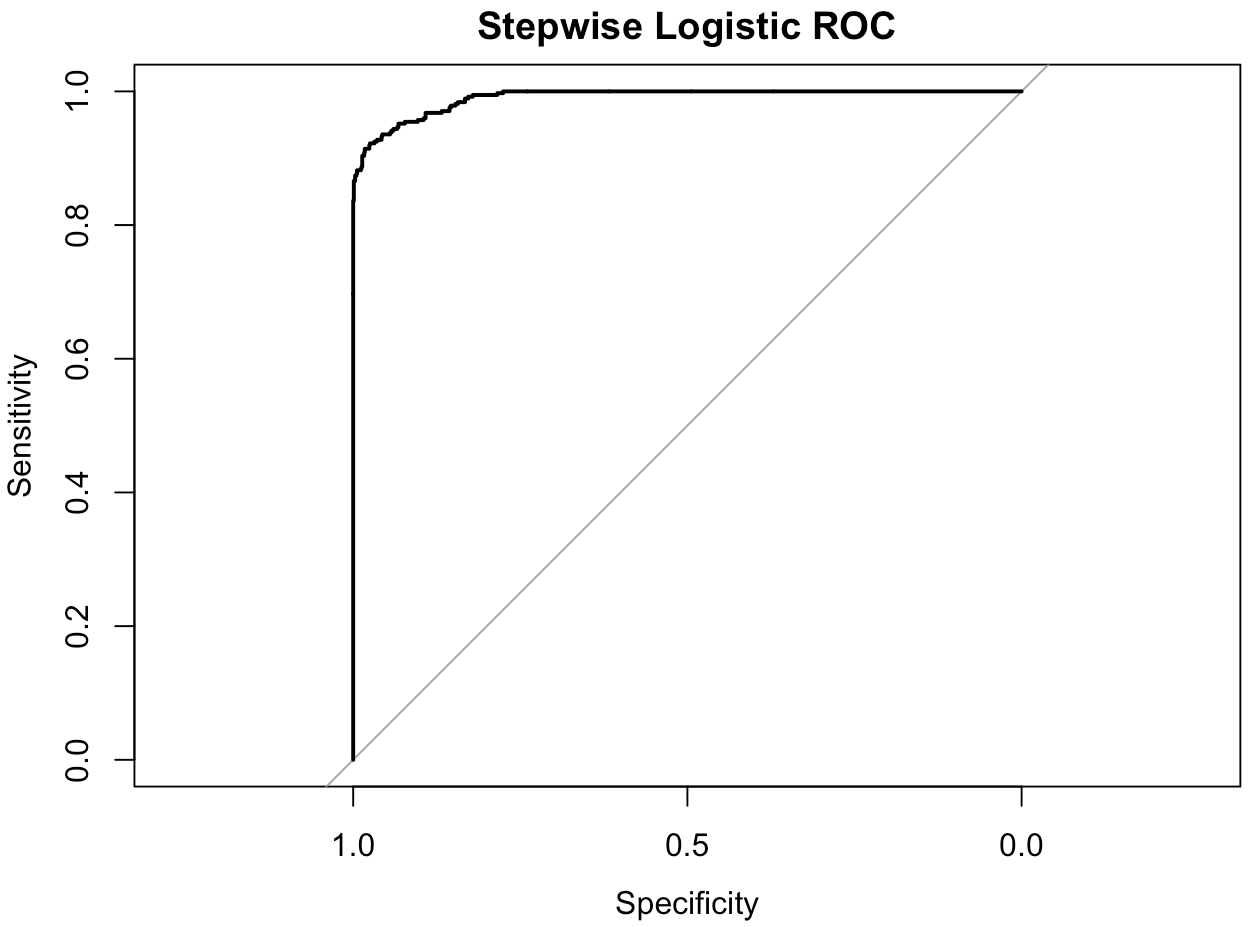
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  1013   32
##        Yes   21  341
##
##                Accuracy : 0.9623
##                  95% CI : (0.951, 0.9717)
##     No Information Rate : 0.7349
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9024
##
##  Mcnemar's Test P-Value : 0.1696
##
##             Sensitivity : 0.9797
##             Specificity : 0.9142
##          Pos Pred Value : 0.9694
##          Neg Pred Value : 0.9420
##              Prevalence : 0.7349
##          Detection Rate : 0.7200
##    Detection Prevalence : 0.7427
##       Balanced Accuracy : 0.9469
##
##        'Positive' Class : No
##
```

```
step_roc <- roc(test$Churn, step_pred_prob)
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
plot(step_roc, main = "Stepwise Logistic ROC")
```




```
auc(step_roc)
```

```
## Area under the curve: 0.9907
```

LASSO Logistic Regression

```
x_train <- model.matrix(Churn ~ ., data = train)[, -1]
y_train <- ifelse(train$Churn == "Yes", 1, 0)
x_test <- model.matrix(Churn ~ ., data = test)[, -1]
y_test <- ifelse(test$Churn == "Yes", 1, 0)

cv_lasso <- cv.glmnet(x_train, y_train, family = "binomial", alpha = 1)
lasso_model <- glmnet(x_train, y_train, family = "binomial", lambda = cv_lasso$lambda.min)
lasso_pred_prob <- predict(lasso_model, newx = x_test, type = "response")
lasso_pred <- ifelse(lasso_pred_prob > 0.5, 1, 0)
confusionMatrix(as.factor(lasso_pred), as.factor(y_test))
```

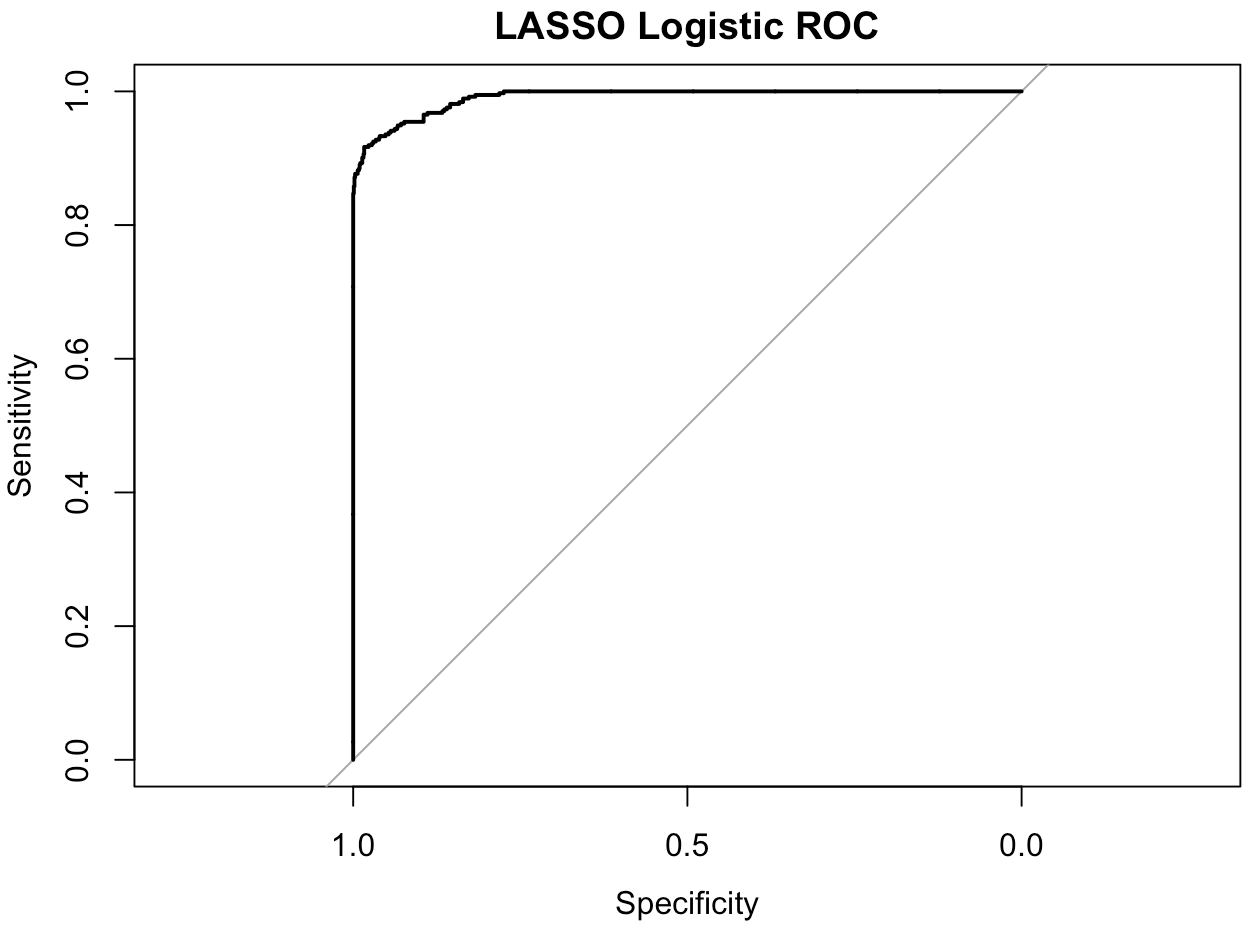
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1017   31
##          1   17  342
##
##                Accuracy : 0.9659
##                  95% CI : (0.955, 0.9747)
##     No Information Rate : 0.7349
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.9114
##
##  Mcnemar's Test P-Value : 0.0606
##
##             Sensitivity : 0.9836
##             Specificity : 0.9169
##          Pos Pred Value : 0.9704
##          Neg Pred Value : 0.9526
##              Prevalence : 0.7349
##          Detection Rate : 0.7228
##    Detection Prevalence : 0.7448
##       Balanced Accuracy : 0.9502
##
##        'Positive' Class : 0
##
```

```
lasso_roc <- roc(y_test, as.vector(lasso_pred_prob))
```

```
## Setting levels: control = 0, case = 1
```

```
## Setting direction: controls < cases
```

```
plot(lasso_roc, main = "LASSO Logistic ROC")
```




```
auc(lasso_roc)
```

```
## Area under the curve: 0.9908
```

Model comparison table

```
results <- data.frame(
  Model = c("Stepwise Logistic", "LASSO Logistic"),
  Accuracy = c(
    sum(step_pred == test$Churn) / length(step_pred),
    sum(lasso_pred == y_test) / length(lasso_pred)
  ),
  AUC = c(auc(step_roc), auc(lasso_roc))
)
print(results)
```

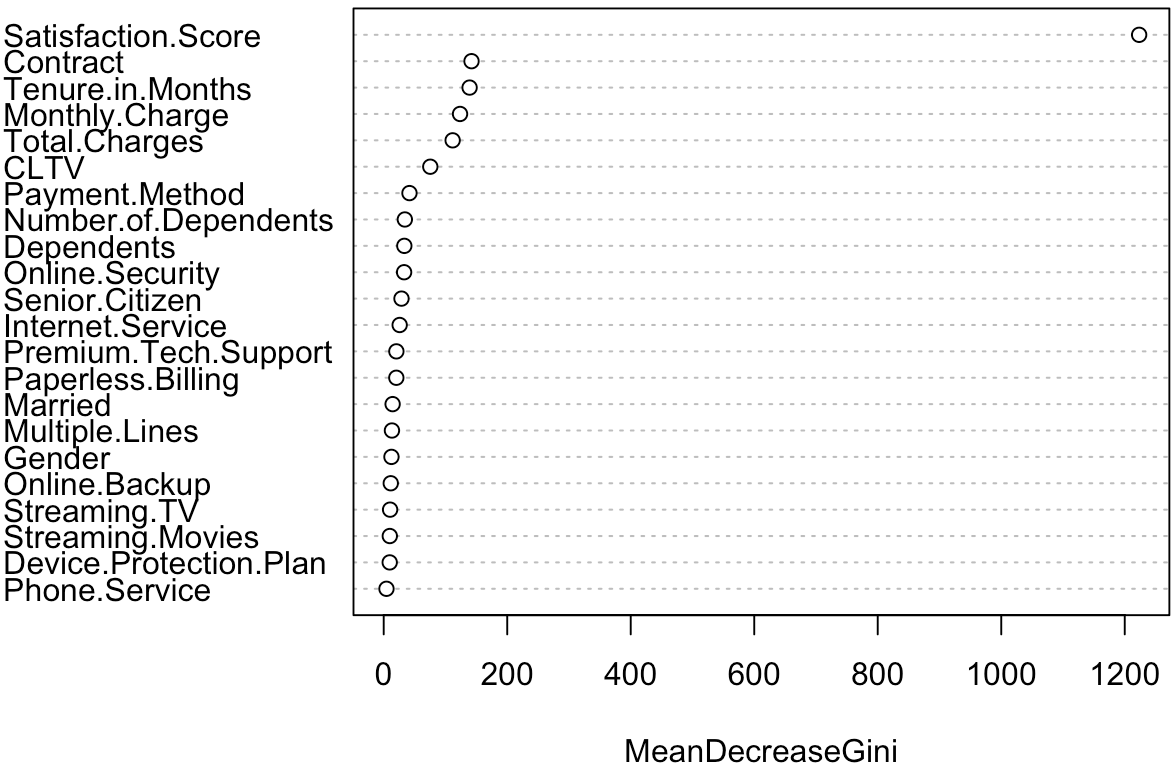
```
##               Model  Accuracy       AUC
## 1 Stepwise Logistic 0.9623312 0.9907281
## 2    LASSO Logistic 0.9658849 0.9908240
```

Extended models: RF, GBM, XGBoost, Clustering

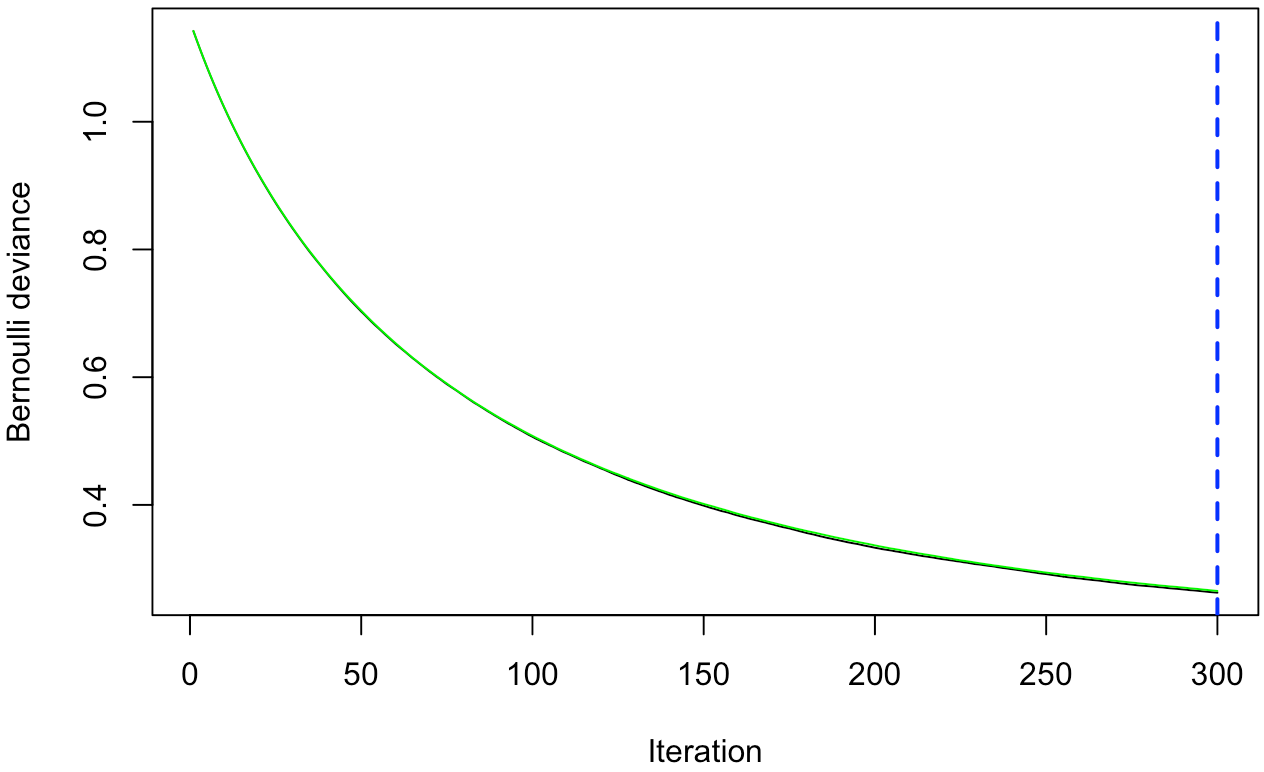
```
rf_model <- randomForest(Churn ~ ., data = train, ntree = 100)
rf_pred <- predict(rf_model, newdata = test)
confusionMatrix(rf_pred, test$Churn)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   No  Yes
##        No  1028   48
##        Yes    6  325
##
##                Accuracy : 0.9616
##                  95% CI : (0.9502, 0.971)
##     No Information Rate : 0.7349
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8978
##
##  Mcnemar's Test P-Value : 2.414e-08
##
##             Sensitivity : 0.9942
##             Specificity : 0.8713
##          Pos Pred Value : 0.9554
##          Neg Pred Value : 0.9819
##              Prevalence : 0.7349
##          Detection Rate : 0.7306
##    Detection Prevalence : 0.7647
##       Balanced Accuracy : 0.9328
##
##        'Positive' Class : No
##
```

```
varImpPlot(rf_model)
```




```
# Copy train and test sets with numeric target for gbm
train_gbm <- train %>%
  mutate(Churn = ifelse(Churn == "Yes", 1, 0))

test_gbm <- test %>%
  mutate(Churn = ifelse(Churn == "Yes", 1, 0))

# Train GBM
gbm_model <- gbm(Churn ~ ., data = train_gbm, distribution = "bernoulli",
                 n.trees = 300, shrinkage = 0.01, interaction.depth = 3, cv.folds = 5)

best_iter <- gbm.perf(gbm_model, method = "cv")
```



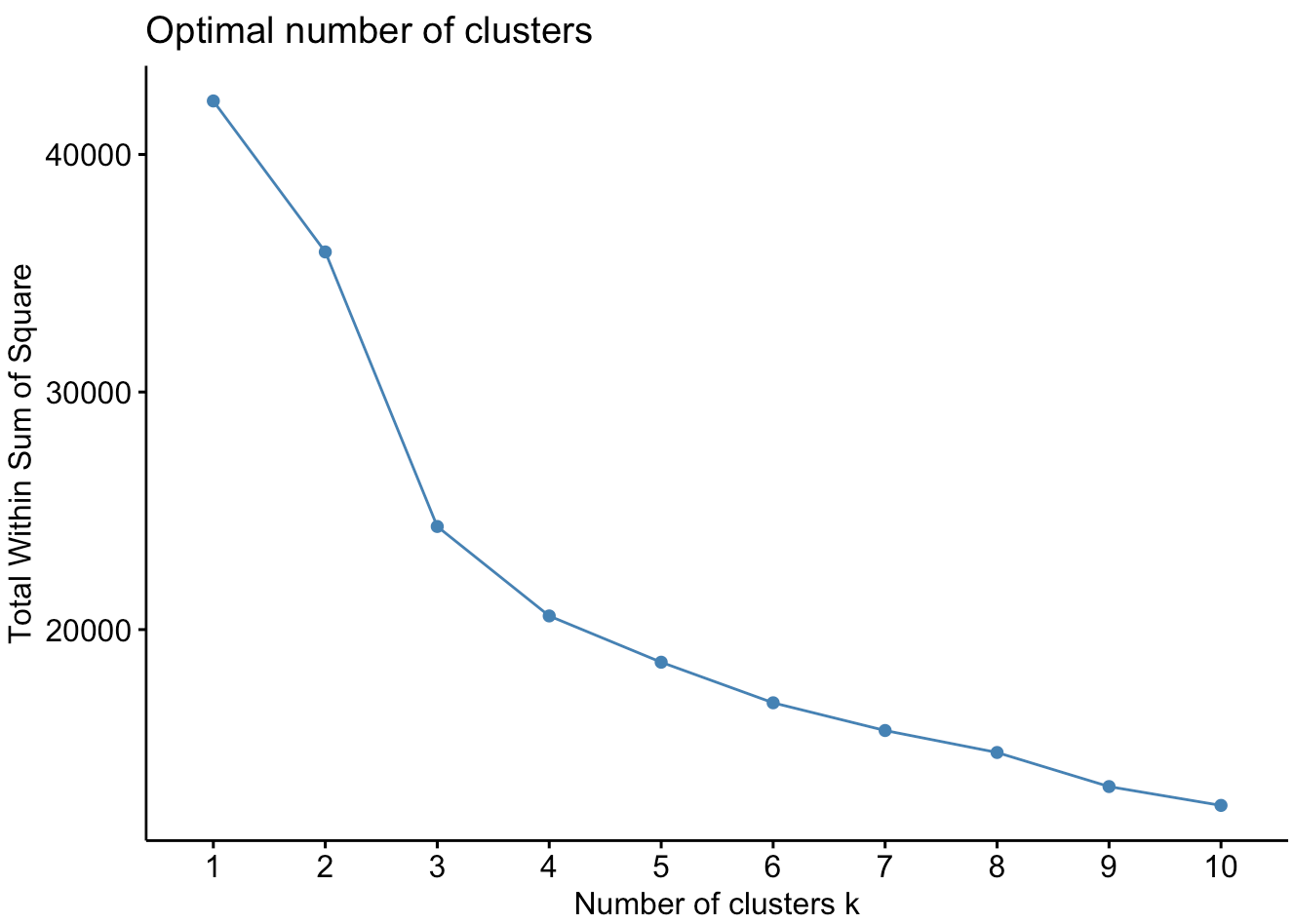

```
# Predict and evaluate
gbm_pred_prob <- predict(gbm_model, newdata = test_gbm, n.trees = best_iter, type = "response")
gbm_pred <- ifelse(gbm_pred_prob > 0.5, 1, 0)
confusionMatrix(as.factor(gbm_pred), as.factor(test_gbm$Churn))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1032   68
##          1    2  305
##
##                Accuracy : 0.9502
##                  95% CI : (0.9376, 0.961)
##     No Information Rate : 0.7349
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8647
##
##  Mcnemar's Test P-Value : 7.912e-15
##
##             Sensitivity : 0.9981
##             Specificity : 0.8177
##          Pos Pred Value : 0.9382
##          Neg Pred Value : 0.9935
##              Prevalence : 0.7349
##          Detection Rate : 0.7335
##    Detection Prevalence : 0.7818
##       Balanced Accuracy : 0.9079
##
##        'Positive' Class : 0
##
```

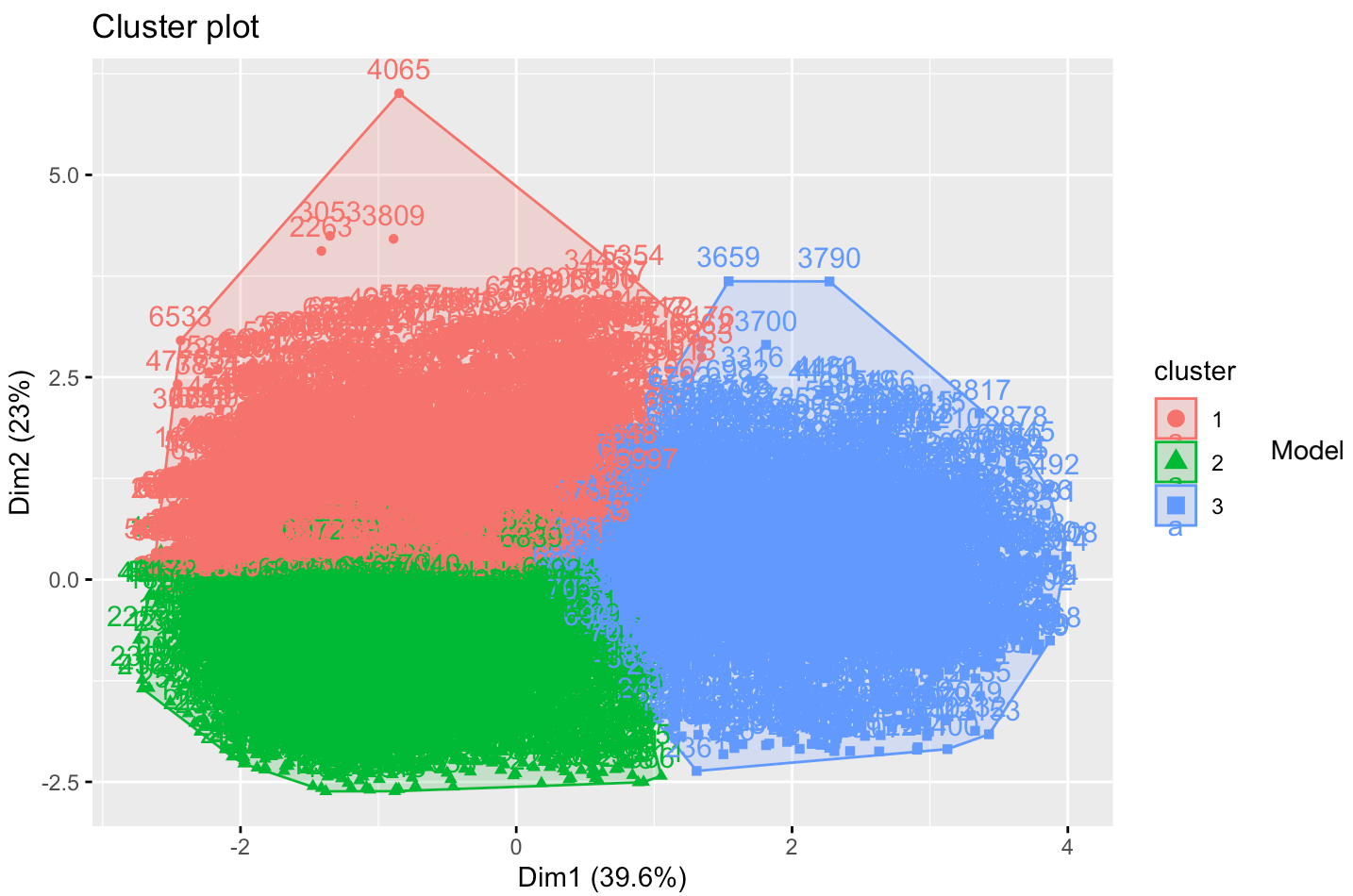
```
xgb_train <- xgb.DMatrix(data = x_train, label = y_train)
xgb_test <- xgb.DMatrix(data = x_test, label = y_test)
xgb_model <- xgboost(data = xgb_train, objective = "binary:logistic", nrounds = 100, verbose = 0)
xgb_pred_prob <- predict(xgb_model, newdata = xgb_test)
xgb_pred <- ifelse(xgb_pred_prob > 0.5, 1, 0)
confusionMatrix(as.factor(xgb_pred), as.factor(y_test))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 1017   40
##          1   17  333
##
##                Accuracy : 0.9595
##                  95% CI : (0.9478, 0.9692)
##     No Information Rate : 0.7349
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8939
##
##  Mcnemar's Test P-Value : 0.003569
##
##             Sensitivity : 0.9836
##             Specificity : 0.8928
##          Pos Pred Value : 0.9622
##          Neg Pred Value : 0.9514
##              Prevalence : 0.7349
##          Detection Rate : 0.7228
##    Detection Prevalence : 0.7512
##       Balanced Accuracy : 0.9382
##
##        'Positive' Class : 0
##
```

```
numeric_scaled <- scale(data_model[, sapply(data_model, is.numeric)])
fviz_nbclust(numeric_scaled, kmeans, method = "wss")
```




```
km <- kmeans(numeric_scaled, centers = 3)
fviz_cluster(km, data = numeric_scaled)
```



Comparison

```
# Model comparison for Stepwise, LASSO, RF, GBM, XGBoost

# Convert any needed factors for fairness
rf_auc <- roc(test$Churn, as.numeric(rf_pred == "Yes"))
```

```
## Setting levels: control = No, case = Yes
```

```
## Setting direction: controls < cases
```

```
gbm_auc <- roc(test_gbm$Churn, gbm_pred_prob)
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
xgb_auc <- roc(y_test, xgb_pred_prob)
```

```
## Setting levels: control = 0, case = 1
## Setting direction: controls < cases
```

```
results_all <- data.frame(
  Model = c("Stepwise Logistic", "LASSO Logistic", "Random Forest", "GBM", "XGBoost"),
  Accuracy = c(
    mean(step_pred == test$Churn),
    mean(lasso_pred == y_test),
    mean(rf_pred == test$Churn),
    mean(gbm_pred == test_gbm$Churn),
    mean(xgb_pred == y_test)
  ),
  AUC = c(
    auc(step_roc),
    auc(lasso_roc),
    auc(rf_auc),
    auc(gbm_auc),
    auc(xgb_auc)
  )
)

print(results_all)
```

```
##               Model  Accuracy       AUC
## 1 Stepwise Logistic 0.9623312 0.9907281
## 2    LASSO Logistic 0.9658849 0.9908240
## 3     Random Forest 0.9616205 0.9327555
## 4               GBM 0.9502488 0.9880342
## 5           XGBoost 0.9594883 0.9894810
```