KNN on Telecom Churn Data

Alexander Vaillant

9/7/2021

Import Necessary Libraries

```
library(caret) # GridSearch
```

```
## Warning: package 'caret' was built under R version 4.1.1
```

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

```
## Loading required package: ggplot2
```

```
library(fastDummies) #DummyColumns
```

```
## Warning: package 'fastDummies' was built under R version 4.1.1
```

```
library(caTools) #colAUC
```

```
## Warning: package 'caTools' was built under R version 4.1.1
```

Data Gathering

Load Dataset into Dataframe using read.csv()

```
## Load dataset into dataframe
url <- "C:/Users/tedda/Desktop/Data Science Portfolio/Machine Learning/Supervised Learning/Classificati
churn_data <- read.csv(url, header = TRUE)
```

Data Preparation

```
# Remove customer demographics by indexing
churn_nodemo <- churn_data[20:50]

# Transform categorical variables into binary dummy variables using fastDummies::dummy_cols()
churn_dummies <- dummy_cols(churn_nodemo, remove_first_dummy = FALSE, remove_selected_columns = TRUE)

# Normalize the dataset using preProcess()
preproc <- preProcess(churn_dummies, method = c("center","scale"))
churn_norm <- predict(preproc,churn_dummies)

dataset_url <- "C:/Users/tedda/Desktop/Data Science Portfolio/Machine Learning/Supervised Learning/Class
write.csv(churn_norm, dataset_url, row.names = FALSE)

# Set seed for random sampling of data
set.seed(123)
```

```r
# Create the index for the random sampling of data
sample_size <- round(0.8*nrow(churn_norm))
train_ind <- sample(1:nrow(churn_norm), size = sample_size)

# Split the data into train and test datasets
churn_train <- churn_norm[train_ind,]
churn_train$Churn_Yes <- factor(churn_train$Churn_Yes, levels = c(max(churn_train$Churn_Yes),min(churn_
write.csv(churn_train, "C:/Users/tedda/Desktop/Data Science Portfolio/Machine Learning/Supervised Learni

churn_test <- churn_norm[-train_ind,]
churn_test$Churn_Yes <- factor(churn_test$Churn_Yes, levels = c(max(churn_test$Churn_Yes),min(churn_tes
write.csv(churn_test, "C:/Users/tedda/Desktop/Data Science Portfolio/Machine Learning/Supervised Learnin
```

Build the Model

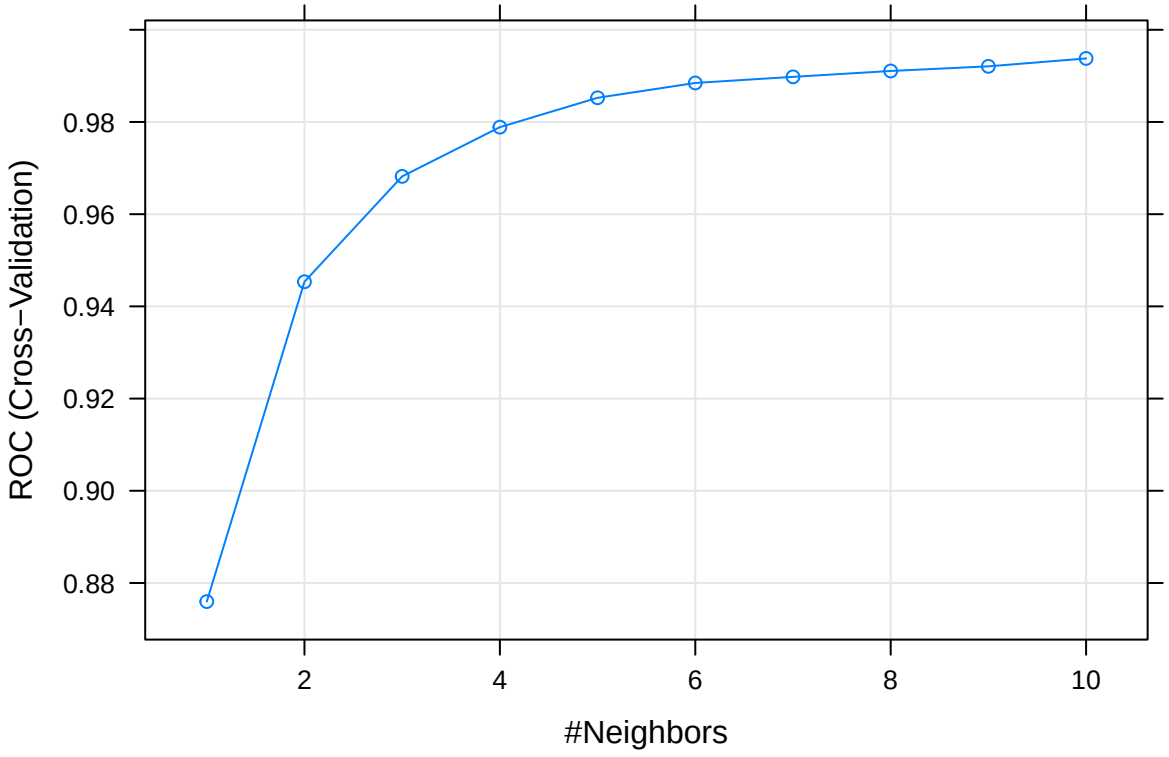
```r
# Build the KNN model using train()
train_ctrl <- trainControl(method = "cv", number = 5, classProbs = TRUE, summaryFunction = twoClassSumma
knn_fit <- train(Churn_Yes ~ ., data = churn_train, method = "knn", metric = "ROC", tuneGrid=expand.grid
```

```
## + Fold1: k= 1
## - Fold1: k= 1
## + Fold1: k= 2
## - Fold1: k= 2
## + Fold1: k= 3
## - Fold1: k= 3
## + Fold1: k= 4
## - Fold1: k= 4
## + Fold1: k= 5
## - Fold1: k= 5
## + Fold1: k= 6
## - Fold1: k= 6
## + Fold1: k= 7
## - Fold1: k= 7
## + Fold1: k= 8
## - Fold1: k= 8
## + Fold1: k= 9
## - Fold1: k= 9
## + Fold1: k=10
## - Fold1: k=10
## + Fold2: k= 1
## - Fold2: k= 1
## + Fold2: k= 2
## - Fold2: k= 2
## + Fold2: k= 3
## - Fold2: k= 3
## + Fold2: k= 4
## - Fold2: k= 4
## + Fold2: k= 5
## - Fold2: k= 5
## + Fold2: k= 6
## - Fold2: k= 6
## + Fold2: k= 7
## - Fold2: k= 7
```

```
## + Fold2: k= 8
## - Fold2: k= 8
## + Fold2: k= 9
## - Fold2: k= 9
## + Fold2: k=10
## - Fold2: k=10
## + Fold3: k= 1
## - Fold3: k= 1
## + Fold3: k= 2
## - Fold3: k= 2
## + Fold3: k= 3
## - Fold3: k= 3
## + Fold3: k= 4
## - Fold3: k= 4
## + Fold3: k= 5
## - Fold3: k= 5
## + Fold3: k= 6
## - Fold3: k= 6
## + Fold3: k= 7
## - Fold3: k= 7
## + Fold3: k= 8
## - Fold3: k= 8
## + Fold3: k= 9
## - Fold3: k= 9
## + Fold3: k=10
## - Fold3: k=10
## + Fold4: k= 1
## - Fold4: k= 1
## + Fold4: k= 2
## - Fold4: k= 2
## + Fold4: k= 3
## - Fold4: k= 3
## + Fold4: k= 4
## - Fold4: k= 4
## + Fold4: k= 5
## - Fold4: k= 5
## + Fold4: k= 6
## - Fold4: k= 6
## + Fold4: k= 7
## - Fold4: k= 7
## + Fold4: k= 8
## - Fold4: k= 8
## + Fold4: k= 9
## - Fold4: k= 9
## + Fold4: k=10
## - Fold4: k=10
## + Fold5: k= 1
## - Fold5: k= 1
## + Fold5: k= 2
## - Fold5: k= 2
## + Fold5: k= 3
## - Fold5: k= 3
## + Fold5: k= 4
## - Fold5: k= 4
```

```
## + Fold5: k= 5
## - Fold5: k= 5
## + Fold5: k= 6
## - Fold5: k= 6
## + Fold5: k= 7
## - Fold5: k= 7
## + Fold5: k= 8
## - Fold5: k= 8
## + Fold5: k= 9
## - Fold5: k= 9
## + Fold5: k=10
## - Fold5: k=10
## Aggregating results
## Selecting tuning parameters
## Fitting k = 10 on full training set
```

```
# Plot the finished model to show ROC at each value of k
plot(knn_fit)
```

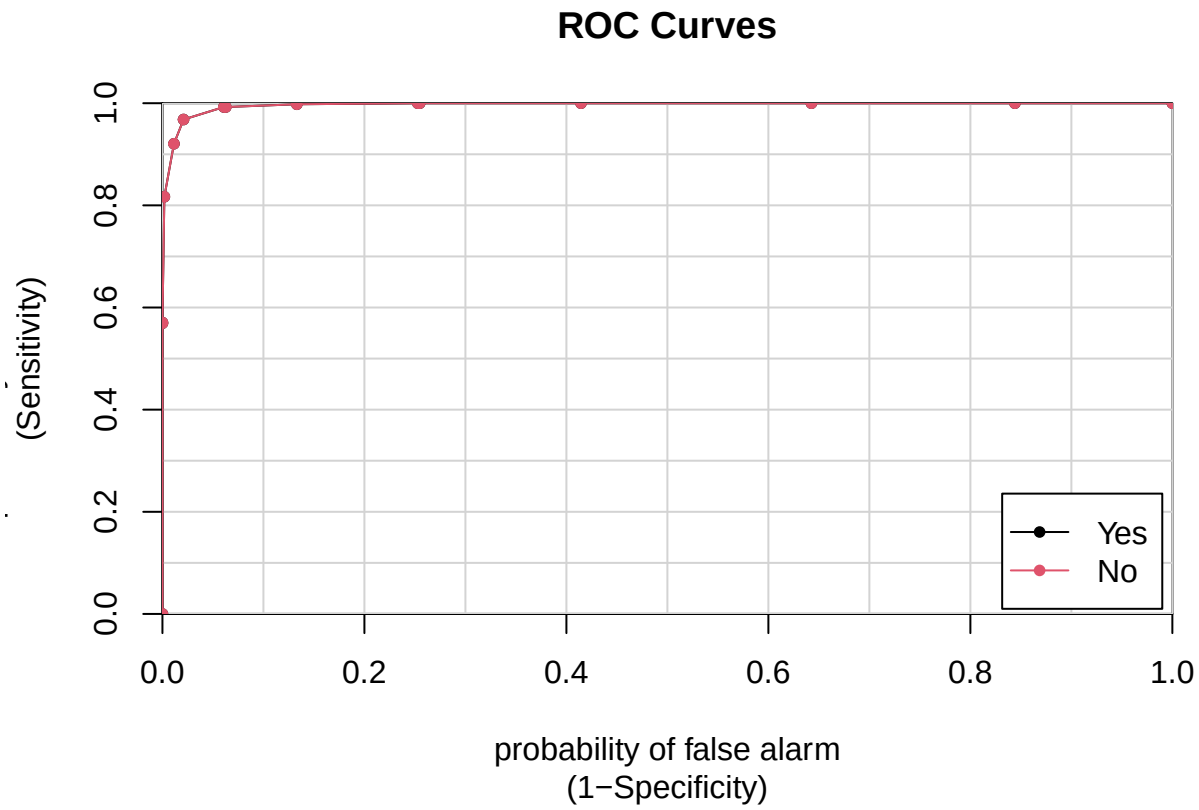


Save and Load the Model

```
# Save and Load KNN model
model_url <- "C:/Users/tedda/Desktop/Data Science Portfolio/Machine Learning/Supervised Learning/Classi
saveRDS(knn_fit, model_url)
KNN_model <- readRDS(model_url)
```

Evaluate the Model

```
# Plot and evaluate the AUC of our final model
pred <- predict(KNN_model, newdata = churn_test, type = "prob")
colAUC(X = pred, y = churn_test$Churn_Yes, plotROC = TRUE)
```




```
##                  Yes        No
## Yes vs. No 0.9963854 0.9963854
```

```
# Create a confusion matrix to show the Accuracy and other metrics of our final model
predconfusion <- predict(KNN_model, newdata = churn_test)
confusionMatrix(predconfusion, churn_test$Churn_Yes)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  Yes   No
##        Yes  474    8
##        No    52 1466
##
##                Accuracy : 0.97
##                  95% CI : (0.9616, 0.977)
##     No Information Rate : 0.737
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9205
##
```

```
##  Mcnemar's Test P-Value : 2.836e-08
##
##             Sensitivity : 0.9011
##             Specificity : 0.9946
##          Pos Pred Value : 0.9834
##          Neg Pred Value : 0.9657
##              Prevalence : 0.2630
##          Detection Rate : 0.2370
##    Detection Prevalence : 0.2410
##       Balanced Accuracy : 0.9479
##
##        'Positive' Class : Yes
##
```