Titanic Survival Prediction

Akanksha Garlapati

### 1. Load and Prepare the Data

The first step is to load and preprocess the Titanic data.

# Load the Titanic dataset  
data("titanic\_train")  
titanic <- titanic\_train  
  
# Check the structure of the data  
str(titanic)

## 'data.frame': 891 obs. of 12 variables:  
## $ PassengerId: int 1 2 3 4 5 6 7 8 9 10 ...  
## $ Survived : int 0 1 1 1 0 0 0 0 1 1 ...  
## $ Pclass : int 3 1 3 1 3 3 1 3 3 2 ...  
## $ Name : chr "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)" "Heikkinen, Miss. Laina" "Futrelle, Mrs. Jacques Heath (Lily May Peel)" ...  
## $ Sex : chr "male" "female" "female" "female" ...  
## $ Age : num 22 38 26 35 35 NA 54 2 27 14 ...  
## $ SibSp : int 1 1 0 1 0 0 0 3 0 1 ...  
## $ Parch : int 0 0 0 0 0 0 0 1 2 0 ...  
## $ Ticket : chr "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...  
## $ Fare : num 7.25 71.28 7.92 53.1 8.05 ...  
## $ Cabin : chr "" "C85" "" "C123" ...  
## $ Embarked : chr "S" "C" "S" "S" ...

# Data cleaning: Remove rows with missing values and select relevant columns  
titanic\_clean <- titanic %>%  
 select(Survived, Pclass, Sex, Age, SibSp, Parch, Fare) %>%  
 drop\_na()  
  
# Convert 'Survived' and 'Sex' to factors  
titanic\_clean$Survived <- factor(titanic\_clean$Survived)  
titanic\_clean$Sex <- factor(titanic\_clean$Sex, levels = c("male", "female"))  
  
# Split the data into training and test sets  
set.seed(123)  
trainIndex <- createDataPartition(titanic\_clean$Survived, p = 0.7, list = FALSE)  
trainData <- titanic\_clean[trainIndex, ]  
testData <- titanic\_clean[-trainIndex, ]

### 2. Logistic Regression

We will first use Logistic Regression to predict survival.

# Train logistic regression model  
logistic\_model <- train(Survived ~ ., data = trainData, method = "glm", family = "binomial")  
  
# Make predictions  
logistic\_preds <- predict(logistic\_model, newdata = testData)  
  
# Evaluate the accuracy  
logistic\_accuracy <- mean(logistic\_preds == testData$Survived)  
logistic\_accuracy

## [1] 0.8037383

### 3. Random Forest

Now, let’s apply the Random Forest algorithm.

# Train Random Forest model  
rf\_model <- randomForest(Survived ~ ., data = trainData, ntree = 100)  
  
# Make predictions  
rf\_preds <- predict(rf\_model, newdata = testData)  
  
# Evaluate the accuracy  
rf\_accuracy <- mean(rf\_preds == testData$Survived)  
rf\_accuracy

## [1] 0.8037383

### 4. k-Nearest Neighbors (kNN)

Lastly, we use k-Nearest Neighbors for prediction.

# Normalize the numeric variables for kNN  
normalize <- function(x) {  
 return((x - min(x)) / (max(x) - min(x)))  
}  
  
# Normalize the numeric columns  
trainData\_knn <- trainData %>%  
 mutate\_at(vars(Age, SibSp, Parch, Fare), normalize)  
  
testData\_knn <- testData %>%  
 mutate\_at(vars(Age, SibSp, Parch, Fare), normalize)  
  
# Convert 'Sex' to numeric (0 for male, 1 for female)  
trainData\_knn$Sex <- as.numeric(trainData\_knn$Sex) - 1  
testData\_knn$Sex <- as.numeric(testData\_knn$Sex) - 1  
  
# Check for any remaining NAs  
sum(is.na(trainData\_knn))

## [1] 0

sum(is.na(testData\_knn))

## [1] 0

# Train the kNN model (k = 5)  
knn\_preds <- knn(  
 train = trainData\_knn[, c("Pclass", "Sex", "Age", "SibSp", "Parch", "Fare")],  
 test = testData\_knn[, c("Pclass", "Sex", "Age", "SibSp", "Parch", "Fare")],  
 cl = trainData\_knn$Survived,   
 k = 5  
 )  
  
# Evaluate the accuracy of the kNN model  
knn\_accuracy <- mean(knn\_preds == testData\_knn$Survived)  
knn\_accuracy

## [1] 0.7850467

### 5. Model Comparison

Now, we can compare the accuracies of the three models.

# Compare accuracies  
results <- data.frame(  
 Model = c("Logistic Regression", "Random Forest", "k-Nearest Neighbors"),  
 Accuracy = c(logistic\_accuracy, rf\_accuracy, knn\_accuracy)  
)  
  
results

## Model Accuracy  
## 1 Logistic Regression 0.8037383  
## 2 Random Forest 0.8037383  
## 3 k-Nearest Neighbors 0.7850467

### 6. Interpret the Results

Interpret the results of the three models, including which one performed the best.

# Plot accuracy comparison  
ggplot(results, aes(x = Model, y = Accuracy, fill = Model)) +  
 geom\_bar(stat = "identity") +  
 ylim(0, 1) +  
 ggtitle("Accuracy of Different Classification Models") +  
 theme\_minimal()

