Problem: Predicting Delay in Flights for Airline Client

Solution: Supervised Regression with Subset Selection and testing with least MSE.

library(caret)

library(tidyverse)

install.packages("dplyr")

#step0 Perform EDA

flight\_data = read.csv(file=file.choose(),header=TRUE)

library(skimr)

summaryStats <- skim(flight\_data)

summaryStats

boxplot(flight\_data$Arr\_Delay~flight\_data$Carrier, ylab="Arrival Delay")

### Step 1: Partition our Data

#We first change categorical variables into dummy variables using model.matrix (we will also use one-hot encoding in the future for non-linear models). The function model.matrix() can automatically convert categorical predictor variables to dummy variables. It also creates a column of 1s, which we don’t need at this time. That column of 1 is used for estimating intercept if you write algorithm by yourself, but most available functions automatically creates that column during estimation.

flight\_predictors\_dummy <- model.matrix(Arr\_Delay~ ., data = flight\_data)#create dummy variables expect for the response

flight\_predictors\_dummy<- data.frame(flight\_predictors\_dummy[,-1]) #get rid of intercept and make data frame

flight\_data <- cbind(Arr\_Delay=flight\_data$Arr\_Delay, flight\_predictors\_dummy)

#We randomly split the data into training (80%) and testing (20%) datasets:

set.seed(99) #set random seed

index <- createDataPartition(flight\_data$Arr\_Delay, p = .8,list = FALSE)

flight\_train <- flight\_data[index,]

flight\_test <- flight\_data[-index,]

###Step 2: Train or Fit Model

#We train the model using the train function. In this train function

#In the first argument, provide a formula with the response variable ~ predictor variables or use response variable ~ . to include all predictor variables

#In the data argument, provide the training data set

#In the method argument, provide the method or machine learning model to use We must also load the relevant libraries for this machine learning model before running the train function. For the linear regression we use method = “lm”

#In the trControl argument, this “none” means fit one model to the entire training set. We will discuss more advance validation approaches later.

library(MASS)

subset\_model <- train(Arr\_Delay ~ .,

data = flight\_train,

method = "glmStepAIC",

direction = "backward",

trControl =trainControl(method = "none"))

#Get results

coef(subset\_model$finalModel)

#Step 3: Get Predictions using Testing Set Data

#For regression problems, we want to get the predicted median house price for all observations in the test set.

subset\_pred<-predict(subset\_model, flight\_test)

#Step 4: Evaluate Model Performance

#For regression problems, we evaluate performance of models using the mean squared error of the test set.

#Calculate the mean squared error for this model.

MSE<-mean((subset\_pred- flight\_test$Arr\_Delay)^2)

MSE