3.4.1: R - Model Validation

Stat 5100: Dr. Bean

Project 2 is focused on using information regarding Tinder profiles to predict the genuineness of the user. Information regarding the total set of variables are included in the project 2 description. For purposes of illustration, only a subset of variables are considered here.

```
# Set a seed so that results are reproducible
set.seed(222)
library(stat5100)

# If you run this code, you will have to import the data differently. This
# particular dataset is not included in the stat5100 package for copyright
# reasons, hence why I load it locally here.
tinder <- read.csv("../../data_csv/tinder.csv")

# All the column names in the tinder dataset
names(tinder)
```

```
##  [1] "ID"            "Genuine"       "SocPrivConc"   "InstPrivConc"
##  [5] "Narcissism"    "SelfEsteem"    "Loneliness"    "Hookup"
##  [9] "Friends"       "Partner"       "Travel"        "SelfValidation"
## [13] "Entertainment" "Orientation"   "Gender"        "Education"
## [17] "Income"        "Employment"    "Age"           "ImpFitness"
## [21] "ImpEnergy"     "ImpAttractive"
```

```
# These columns by default are set as factors, which are categorical variables.
# To make this work out for a continuous response variable, we need to convert
# some of the columns to numeric values.
numeric_columns <- c("Genuine", "SocPrivConc", "InstPrivConc", "Narcissism",
                     "SelfEsteem", "Loneliness", "Hookup", "Friends", "Partner",
                     "Travel", "SelfValidation", "Entertainment", "Age",
                     "ImpFitness", "ImpEnergy", "ImpAttractive")
for (nc in numeric_columns) {
  tinder[[nc]] <- as.numeric(tinder[[nc]])
}
```

```
## Warning:  NAs introduced by coercion
## Warning:  NAs introduced by coercion
## Warning:  NAs introduced by coercion
## Warning:  NAs introduced by coercion
## Warning:  NAs introduced by coercion
## Warning:  NAs introduced by coercion
## Warning:  NAs introduced by coercion
## Warning:  NAs introduced by coercion
## Warning:  NAs introduced by coercion
## Warning:  NAs introduced by coercion
## Warning:  NAs introduced by coercion
## Warning:  NAs introduced by coercion
```

```r
# We see a few warnings above because there were some missing values coded as
# "." inside of the csv file.

# Some of the variables we want to be factors (for example, different gender
# classes, sexual orientations, etc.)
factor_columns <- c("Orientation", "Gender", "Education", "Income", "Employment")
for (fc in factor_columns) {
  tinder[[fc]] <- as.factor(tinder[[fc]])
}

# Separate the data into training and test sets. Here we will withhold 20%
# for validation.
n <- nrow(tinder)
training_index <- sample(1:n, size = 0.20 * n)

tinder_train <- tinder[training_index, ]
tinder_test <- tinder[-training_index, ]

# Fit one model with 4 variables
tinder_lm1 <- lm(Genuine ~ SocPrivConc + InstPrivConc + Narcissism + SelfEsteem,
                 data = tinder_train)

# Fit another model with more variables
tinder_lm2 <- lm(Genuine ~ SocPrivConc + InstPrivConc + Narcissism +
                   SelfEsteem + Loneliness + Hookup + Friends + Partner +
                   Travel + SelfValidation + Entertainment, data = tinder_train)

# To fit a third model with no predictors, we simply use the average of the
# response variable (Genuine). Having this third "model" can help us decide
# if there is any significant improvement using the predictors over simply
# guessing the average.
tinder_baseline_pred <- mean(tinder_train$Genuine)
```

Calculate MSPR for each model

```r
# To do this, we make predictions with the testing dataset and then compare
# it to the known value of the response variable Genuine in the testing dataset.
tinder_test_pred1 <- predict(tinder_lm1, newdata = tinder_test)
tinder_test_pred2 <- predict(tinder_lm2, newdata = tinder_test)
tinder_test_pred3 <- rep(tinder_baseline_pred, times = nrow(tinder_test))

mspr1 <- mean((tinder_test_pred1 - tinder_test$Genuine)^2, na.rm = T)
mspr2 <- mean((tinder_test_pred2 - tinder_test$Genuine)^2, na.rm = T)
mspr3 <- mean((tinder_test_pred3 - tinder_test$Genuine)^2, na.rm = T)

# Show results
data.frame(mspr1, mspr2, mspr3)

##      mspr1    mspr2    mspr3
## 1 2.701384 2.329362 2.747672

# Based upon the MSPR, it looks like models 1 and 3 perform roughly the same
# on the testing dataset, but model 2 does better.
```