Regression Similarity

Parker Tate

Overview

The data set contains passenger satisfaction scores over multiple areas. These scores are on a scale from 1 to 5, with 0 being NA. These areas of satisfaction included factors such as seat comfort and cleanliness. The data set also contained passenger flight details (class, type of travel) and general flight information (distance, delays). The data set can be found on Kaggle (https://www.kaggle.com/datasets/teejmahal20/airline-passenger-satisfaction (https://www.kaggle.com/datasets/teejmahal20/airline-passenger-satisfaction)).

Although not specified, I am assuming that the satisfaction scores on the scale of 1 to 5 indicate that 1 is the worst and 5 is the best. Units of distance and time were also not given.

The data set was pre-split into train and test data, so we merged them together to clean.

```
set.seed(1234)
data1 <- read.csv("train.csv")
data2 <- read.csv("test.csv")
data <- rbind(data1, data2)
```

Data Cleaning

I remove unnecessary predictors and map qualitative predictors as dummy variables.

```
data <- subset(data, select = -c(X, id))

data$Customer.Type <- ifelse(data$Customer.Type=="Local Customer", 1, 0)
data$Gender <- ifelse(data$Gender=="Female", 1, 0)
data$Type.of.Travel <- ifelse(data$Type.of.Travel=="Business travel", 1, 0)
data$Class[data$Class == "Eco"] <- 0
data$Class[data$Class == "Eco Plus"] <- 1
data$Class[data$Class == "Business"] <- 2
data$Class <- as.numeric(data$Class)
data$satisfaction <- ifelse(data$satisfaction=="satisfied", 1, 0)
```

Check for any NA values, and any columns with a value of 0.

```
sapply(data, function(x) sum(is.na(x) == TRUE))
```

```
##                             Gender                      Customer.Type
##                                  0                                  0
##                                Age                      Type.of.Travel
##                                  0                                  0
##                              Class                     Flight.Distance
##                                  0                                  0
##              Inflight.wifi.service Departure.Arrival.time.convenient
##                                  0                                  0
##             Ease.of.Online.booking                      Gate.location
##                                  0                                  0
##                    Food.and.drink                     Online.boarding
##                                  0                                  0
##                       Seat.comfort             Inflight.entertainment
##                                  0                                  0
##                  On.board.service                   Leg.room.service
##                                  0                                  0
##                  Baggage.handling                    Checkin.service
##                                  0                                  0
##                  Inflight.service                        Cleanliness
##                                  0                                  0
##        Departure.Delay.in.Minutes          Arrival.Delay.in.Minutes
##                                  0                                393
##                      satisfaction
##                                  0
```

```
sapply(data, function(x) sum(length(which(x == 0))))
```

```
##                             Gender                      Customer.Type
##                              63981                             129880
##                                Age                    Type.of.Travel
##                                  0                              40187
##                              Class                   Flight.Distance
##                              58309                                 0
##              Inflight.wifi.service Departure.Arrival.time.convenient
##                               3916                              6681
##            Ease.of.Online.booking                     Gate.location
##                               5682                                 1
##                    Food.and.drink                    Online.boarding
##                                132                              3080
##                       Seat.comfort             Inflight.entertainment
##                                  1                                18
##                  On.board.service                   Leg.room.service
##                                  5                               598
##                  Baggage.handling                   Checkin.service
##                                  0                                 1
##                  Inflight.service                       Cleanliness
##                                  5                                14
##        Departure.Delay.in.Minutes           Arrival.Delay.in.Minutes
##                             73356                             72753
##                      satisfaction
##                             73452
```

I remove all NA observations and any values of 0 that shouldn't be, which are our satisfaction scores.

```
data <- data[!(is.na(data$Arrival.Delay.in.Minutes)),]
data <- data[!(data$Gate.location==0),]
data <- data[!(data$Food.and.drink==0),]
data <- data[!(data$Inflight.wifi.service==0),]
data <- data[!(data$Departure.Arrival.time.convenient==0),]
data <- data[!(data$Ease.of.Online.booking==0),]
data <- data[!(data$Online.boarding==0),]
data <- data[!(data$Seat.comfort==0),]
data <- data[!(data$Inflight.entertainment==0),]
data <- data[!(data$On.board.service==0),]
data <- data[!(data$Leg.room.service==0),]
data <- data[!(data$Checkin.service==0),]
data <- data[!(data$Inflight.service==0),]
data <- data[!(data$Cleanliness==0),]
```

Final overview of our cleaned data.

```
str(data)
```

```
## 'data.frame':    119204 obs. of  23 variables:
##  $ Gender                           : num  0 0 1 1 0 1 0 1 1 0 ...
##  $ Customer.Type                    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Age                              : int  13 25 26 25 61 26 47 52 41 20 ...
##  $ Type.of.Travel                   : num  0 1 1 1 1 0 0 1 1 1 ...
##  $ Class                            : num  1 2 2 2 2 0 0 2 2 0 ...
##  $ Flight.Distance                  : int  460 235 1142 562 214 1180 1276 2035 853 1061
...
##  $ Inflight.wifi.service            : int  3 3 2 2 3 3 2 4 1 3 ...
##  $ Departure.Arrival.time.convenient: int  4 2 2 5 3 4 4 3 2 3 ...
##  $ Ease.of.Online.booking           : int  3 3 2 5 3 2 2 4 2 3 ...
##  $ Gate.location                    : int  1 3 2 5 3 1 3 4 2 4 ...
##  $ Food.and.drink                   : int  5 1 5 2 4 1 2 5 4 2 ...
##  $ Online.boarding                  : int  3 3 5 2 5 2 2 5 3 3 ...
##  $ Seat.comfort                     : int  5 1 5 2 5 1 2 5 3 3 ...
##  $ Inflight.entertainment           : int  5 1 5 2 3 1 2 5 1 2 ...
##  $ On.board.service                 : int  4 1 4 2 3 3 3 5 1 2 ...
##  $ Leg.room.service                 : int  3 5 3 5 4 4 3 5 2 3 ...
##  $ Baggage.handling                 : int  4 3 4 3 4 4 4 5 1 4 ...
##  $ Checkin.service                  : int  4 1 4 1 3 4 3 4 4 4 ...
##  $ Inflight.service                 : int  5 4 4 4 3 4 5 5 1 3 ...
##  $ Cleanliness                      : int  5 1 5 2 3 1 2 4 2 2 ...
##  $ Departure.Delay.in.Minutes       : int  25 1 0 11 0 0 9 4 0 0 ...
##  $ Arrival.Delay.in.Minutes         : num  18 6 0 9 0 0 23 0 0 0 ...
##  $ satisfaction                     : num  0 0 1 0 1 0 0 1 0 0 ...
```

```
head(data)
```

	Gender <dbl>	Customer.Type <dbl>	A… <int>	Type.of.Travel <dbl>	Class <dbl>	Flight.Distance <int>
1	0	0	13	0	1	460
2	0	0	25	1	2	235
3	1	0	26	1	2	1142
4	1	0	25	1	2	562
5	0	0	61	1	2	214
6	1	0	26	0	0	1180

6 rows | 1-7 of 24 columns

Split into train/test, with a 8:2 ratio.

```
i <- sample(1:nrow(data), nrow(data)*0.80, replace=FALSE)
train <- data[i,]
test <- data[-i,]
```

Data Exploration

Range of minimum and maximum flight distances.

```
range(train$Flight.Distance)
```

```
## [1]   31 4983
```

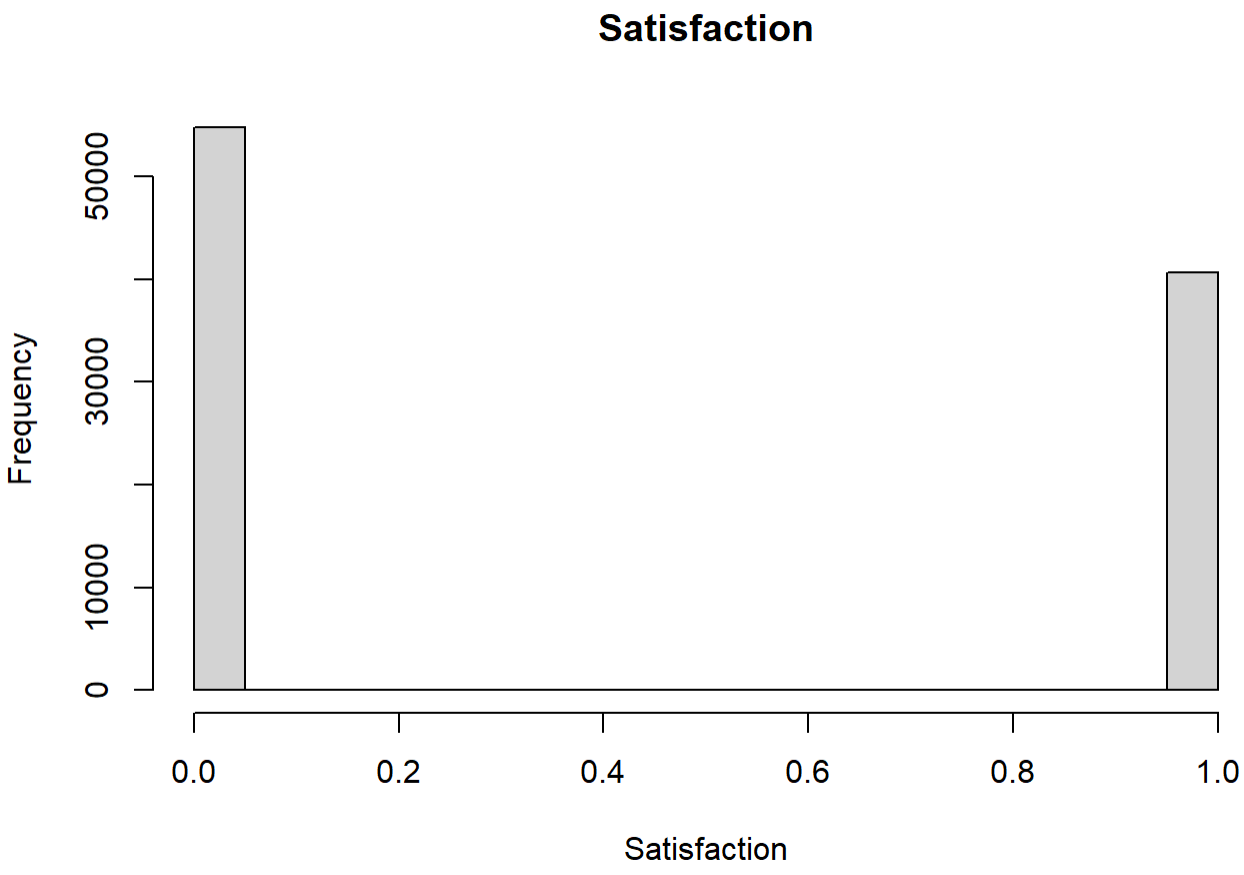
Average age.

```
mean(train$Age)
```

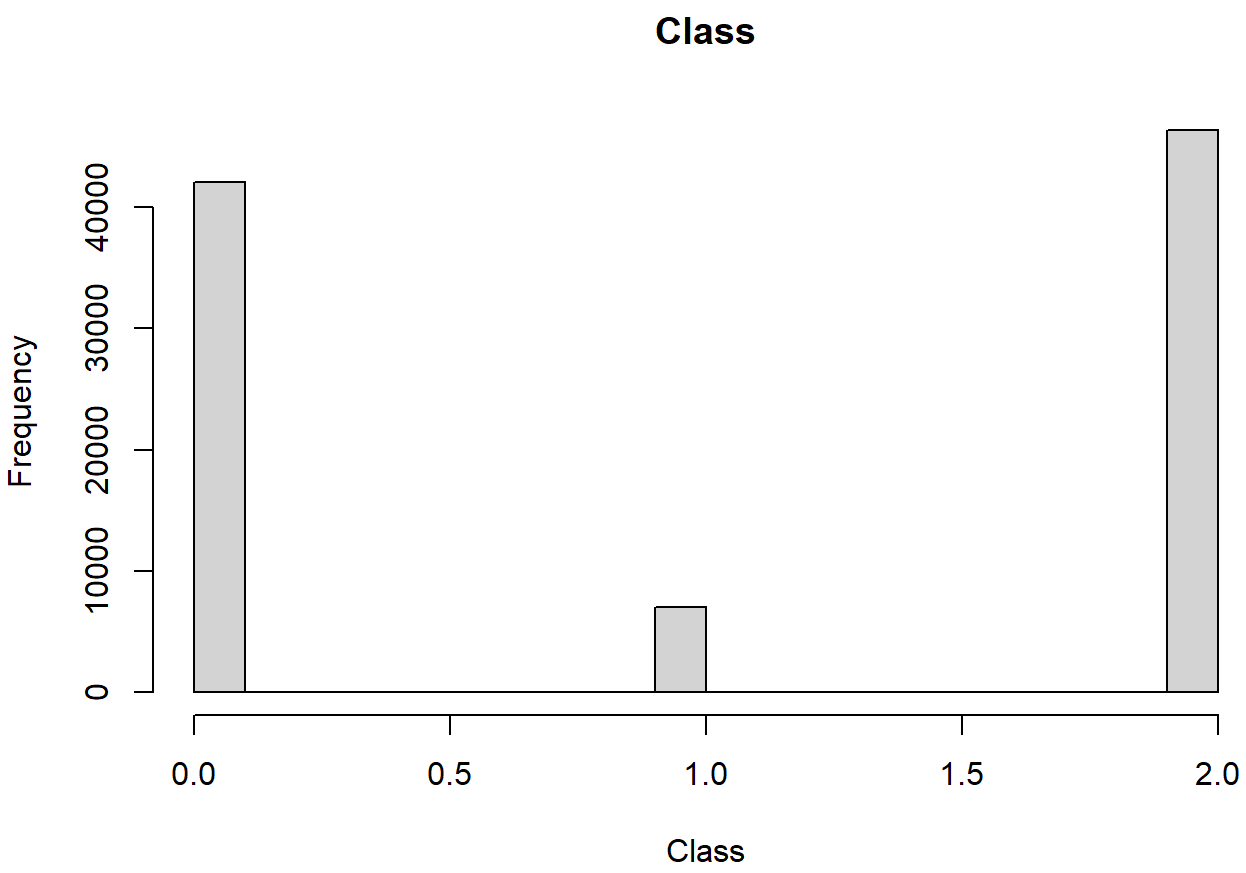
```
## [1] 39.8193
```

Histograms of satisfaction and class, due to them being converted into dummy variables with integers.

```
hist(train$satisfaction, xlab="Satisfaction", main = "Satisfaction")
```




```
hist(train$Class, xlab="Class", main = "Class")
```



Regression Models

Because this notebook focuses on regression we won't be predicting overall satisfaction, as it was given as a factor with only two options: satisfied and neutral/dissatisfied. Instead we are opting to predict a single satisfaction score given the other predictors. In this case we are using the passenger's satisfaction of food and drink (Food.and.drink).

Linear Regression

Fitting a linear regression model on the training data set, as well as a summary of said model.

```
lm1 <- lm(Food.and.drink~., data=train)
summary(lm1)
```

```
##
## Call:
## lm(formula = Food.and.drink ~ ., data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7590 -0.3699  0.0450  0.4206  3.5794
##
## Coefficients: (1 not defined because of singularities)
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       1.477e+00  2.069e-02  71.409  < 2e-16 ***
## Gender                           -2.598e-02  5.923e-03  -4.386 1.15e-05 ***
## Customer.Type                            NA         NA      NA       NA
## Age                              -3.826e-03  2.043e-04 -18.730  < 2e-16 ***
## Type.of.Travel                   -3.211e-02  8.609e-03  -3.730 0.000192 ***
## Class                             1.598e-02  4.305e-03   3.711 0.000206 ***
## Flight.Distance                  -1.780e-05  3.345e-06  -5.321 1.04e-07 ***
## Inflight.wifi.service             3.264e-02  3.644e-03   8.958  < 2e-16 ***
## Departure.Arrival.time.convenient  1.782e-02  2.973e-03   5.994 2.06e-09 ***
## Ease.of.Online.booking            1.008e-02  3.666e-03   2.750 0.005960 **
## Gate.location                    -3.177e-02  3.008e-03 -10.564  < 2e-16 ***
## Online.boarding                  -5.896e-03  3.407e-03  -1.730 0.083545 .
## Seat.comfort                      1.506e-01  3.451e-03  43.649  < 2e-16 ***
## Inflight.entertainment            4.763e-01  4.101e-03 116.143  < 2e-16 ***
## On.board.service                 -9.116e-02  3.089e-03 -29.507  < 2e-16 ***
## Leg.room.service                 -6.360e-02  2.641e-03 -24.083  < 2e-16 ***
## Baggage.handling                 -7.625e-02  3.533e-03 -21.582  < 2e-16 ***
## Checkin.service                   2.633e-02  2.605e-03  10.105  < 2e-16 ***
## Inflight.service                 -1.006e-01  3.675e-03 -27.376  < 2e-16 ***
## Cleanliness                       2.614e-01  3.658e-03  71.457  < 2e-16 ***
## Departure.Delay.in.Minutes        3.331e-04  2.947e-04   1.130 0.258330
## Arrival.Delay.in.Minutes         -5.934e-04  2.907e-04  -2.041 0.041211 *
## satisfaction                     -9.340e-02  9.304e-03 -10.038  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9104 on 95341 degrees of freedom
## Multiple R-squared:  0.5287, Adjusted R-squared:  0.5286
## F-statistic:  5093 on 21 and 95341 DF,  p-value: < 2.2e-16
```

Predict on our model. With a correlation of ~0.734, linear regression fits our data fairly well. The warning indicates that some predictors are perfectly correlated.

```
pred_lm <- predict(lm1, newdata=test)
```

```
## Warning in predict.lm(lm1, newdata = test): prediction from a rank-deficient
## fit may be misleading
```

```
cor_lm <- cor(pred_lm, test$Food.and.drink)
mse_lm <- mean((pred_lm - test$Food.and.drink)^2)
print(paste("cor = ", cor_lm))
```

```
## [1] "cor =  0.725325410519018"
```

```
print(paste("mse = ", mse_lm))
```

```
## [1] "mse =  0.823896396030579"
```

kNN Regression

Fit the kNN algorithm on our training data set. I found a k value of 25 to be best given our data. With a correlation of ~0.479, kNN performs considerably worse than linear regression did. I predict this is because most of our predictors were scores on a scale from 1 to 5, and it created a lot of overlap between neighbors.

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.2.3
```

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

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

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

```
fit <- knnreg(train[,-11], train$Food.and.drink, k=25)
pred_knn <- predict(fit, test[,-11])
cor_knn <- cor(pred_knn, test$Food.and.drink)
mse_knn <- mean((pred_knn - test$Food.and.drink)^2)
print(paste("cor = ", cor_knn))
```

```
## [1] "cor =  0.472034190276989"
```

```
print(paste("mse = ", mse_knn))
```

```
## [1] "mse =  1.39498492657436"
```

Decision Tree Regression

Fit the decision tree model to our training data and print a summary.

```
library(tree)
```

```
## Warning: package 'tree' was built under R version 4.2.3
```

```
tree1 <- tree(Food.and.drink~., data = train)
summary(tree1)
```

```
##
## Regression tree:
## tree(formula = Food.and.drink ~ ., data = train)
## Variables actually used in tree construction:
## [1] "Cleanliness"            "Inflight.entertainment" "Inflight.service"
## [4] "On.board.service"
## Number of terminal nodes:  9
## Residual mean deviance:  0.8078 = 77030 / 95350
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -3.98900 -0.14120 -0.02161  0.00000  0.34070  3.97800
```

Predict on our model with our test data. Our decision tree performed slightly better than linear regression, and much better than kNN regression.

```
pred_tree <- predict(tree1, newdata=test)
cor_tree <- cor(pred_tree, test$Food.and.drink)
mse_tree <- mean((pred_tree - test$Food.and.drink)^2)
print(paste("cor = ", cor_tree))
```

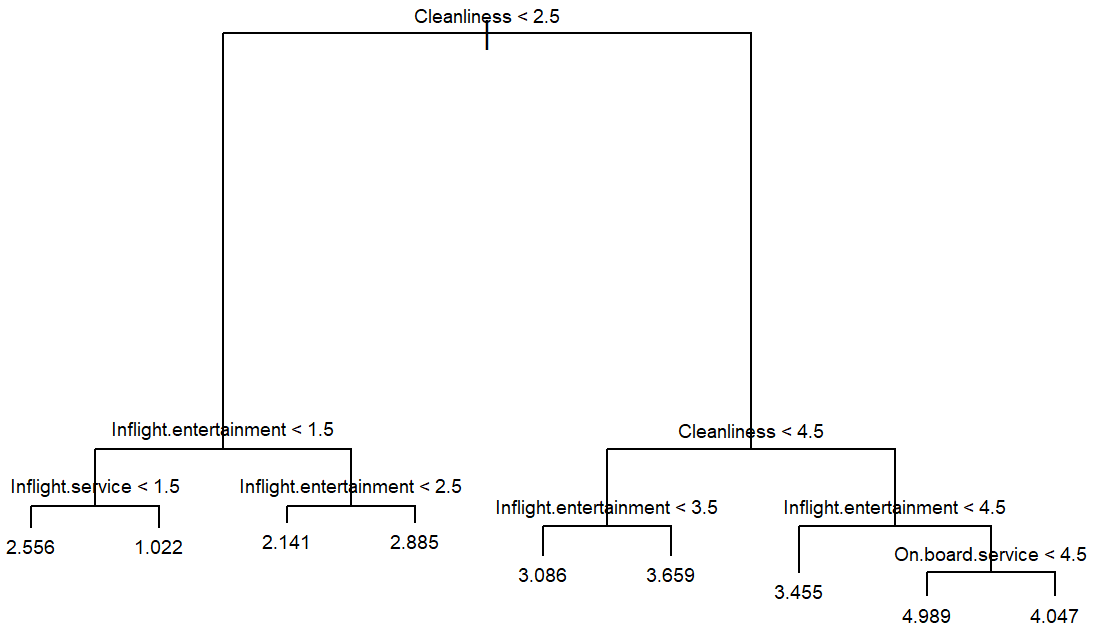
```
## [1] "cor =  0.731274037829636"
```

```
print(paste("mse = ", mse_tree))
```

```
## [1] "mse =  0.808806111337995"
```

Plot the decision tree.

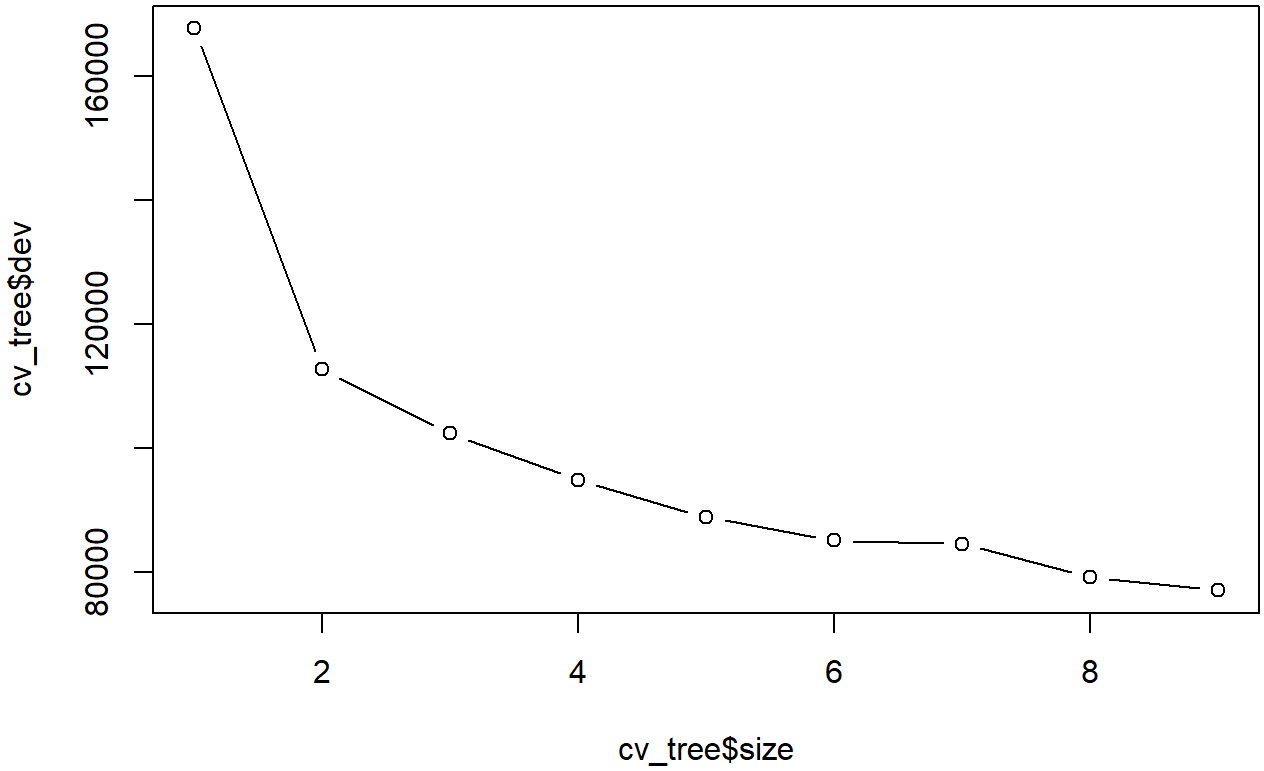
```
plot(tree1)
text(tree1, cex = 0.6, pretty = 0)
```



Cross Validation

This plot shows deviance (y-axis) for different sizes of trees (x-axis). A larger tree size will minimize deviance but may result in over fitting of data, while a small tree will tend to have high deviance.

```
cv_tree <- cv.tree(tree1)
plot(cv_tree$size, cv_tree$dev, type='b')
```



Conclusion

Linear Regression tries to fit a linear trend to our data, so with a correlation of ~0.734 this suggests that the variables possess a fairly linear relationship. This makes sense as most of our predictors are integer scores on a scale of 1 to 5 that can't easily encode a complex relationship due the limited domain. Because our data was in a limited domain, there were less possible outliers which will allow linear regression to perform better.

kNN didn't perform well with a correlation of ~0.479. kNN regression predicts based on the kth nearest neighbors, and because of the limited domain of most of our predictors, they were all clustered together and harder to distinguish between. I predict that scaling the data better would improve this slightly, but still perform poorly.

The decision tree was our best performing model with a correlation of ~0.740. Because our data resulted in a fairly shallow tree, the tree didn't over fit and performed well.