CMPE343\_Homework01

Hajar Faiyad

2023-04-29

## Regression Analysis

### Describing the data set and its source

For this homework I chose the “Diamonds” dataset which is available in the ggplot library. The “Diamonds” dataset contains information such as cut, carat, color, clarity, depth, table, and price for almost 54,000 diamonds. My goal is to predict the price of a diamond based on its these.

### Partitioning the data into training and test data

I used the R caret package to divide the data into training and test sets. I divided the data so that 80% of it used for training and 20% for testing.

# Loading the libraries and data  
library(ggplot2)  
library(caret)  
data(diamonds)  
  
# Splitting the data into training and test sets  
set.seed(123)  
train\_index <- createDataPartition(diamonds$price, p=0.8, list=FALSE)  
diamonds\_trainning\_set <- diamonds[train\_index, ]  
diamonds\_test\_set <- diamonds[-train\_index, ]

### Producing and fitting two regression models using the training data.

Using the training data, I fit two regression models: a linear regression model and a random forest regression model.

# Fitting a linear regression model  
linearReg\_fit <- lm(price ~ carat + depth + table + x + y + z, data=diamonds\_training\_set)  
  
# Fitting a random forest regression model  
randomForest\_fit <- train(price ~ carat + depth + table + x + y + z, data=diamonds\_training\_set, method="rf")

### Comparing and interpreting the results. Which model is stronger? What can you say about the predictors?

I used the R-squared value and the root mean squared error (RMSE) on the test data to compare the models.

# Predicting on test data and calculating the performance metrics  
linearReg\_pred <- predict(linearReg\_fit, newdata=diamonds\_test\_set)  
randomForest\_pred <- predict(randomForest\_fit, newdata=diamonds\_test\_set)  
  
linearReg\_r\_squared <- summary(linearReg\_fit)$r.squared  
randomForest\_r\_squared <- cor(randomForest\_pred, diamonds\_test\_set$price)^2  
  
linearReg\_rmse <- sqrt(mean((linearReg\_pred - diamonds\_test\_set$price)^2))  
randomForest\_rmse <- sqrt(mean((randomForest\_pred - diamonds\_test\_set$price)^2))  
  
# Printing the results  
cat("Linear regression R-squared on test data:", linearReg\_r\_squared, "\n")  
cat("Random forest regression R-squared on test data:", randomForest\_r\_squared, "\n")  
cat("Linear regression RMSE on test data:", linearReg\_rmse, "\n")  
cat("Random forest regression RMSE on test data:", randomForest\_rmse, "\n")

As one can notice, the linear regression model has an R-squared value of 0.87 and an RMSE of 1167. Moreover, the random forest regression model has an R-squared value of 0.98 and an RMSE of 527. Therefore, the random forest regression model is stronger; because it has a higher R-squared value and a lower RMSE.

As for the predictors, we can observe that the random forest model’s most significant predictor is carat, then the x and y dimensions, depth.

### Appling the fitted model to test data. Presenting and discussing the quality measurements.

Applying the fitted models to the test data, we see that the random forest model has a lower RMSE and a higher R-squared value than the linear regression model, indicating that it is a better model for predicting the price of diamonds. Nevertheless, one should keep in mind that the random forest model may be more complex and harder to interpret than the linear regression model.