Regression

Alekhya Pinnamaneni and Aloksai Choudari

October 6, 2022

Select a data set

Data set: Metro Interstate Traffic Volume Data

Source: https://archive.ics.uci.edu/ml/datasets/Metro+Interstate+Traffic+Volume

Target column: 'traffic_volume'

No. of rows: 48,205 rows

Load the data

```
traffic <- read.csv("traffic.csv")
```

Clean the data

```
traffic$holiday <- as.numeric(as.integer(factor(traffic$holiday)))
traffic$temp <- as.numeric(as.integer(factor(traffic$temp)))
traffic$rain_1h <- as.numeric(as.integer(factor(traffic$rain_1h)))
traffic$snow_1h <- as.numeric(as.integer(factor(traffic$snow_1h)))
traffic$clouds_all <- as.numeric(as.integer(factor(traffic$clouds_all)))
traffic$weather_main <- as.numeric(as.integer(factor(traffic$weather_main)))
traffic$weather_description <- as.numeric(as.integer(factor(traffic$weather_description)))
traffic$date_time <- as.numeric(as.integer(factor(traffic$date_time)))
traffic$traffic_volume <- as.numeric(as.integer(factor(traffic$traffic_volume)))
```

Split the data into train and test data

```
set.seed(1234)
sample <- sample(1:nrow(traffic), nrow(traffic)*0.8, replace=FALSE)
train <- traffic[sample,]
test <- traffic[-sample,]
```

Statistical and graphical data exploration

```r
attach(train)

# Prints the most common weather condition and the most common weather description
names(which.max(table(weather_main)))
```

```
## [1] "2"
```

```r
names(which.max(table(weather_description)))
```

```
## [1] "26"
```

```r
# Prints all the holidays
unique(holiday)
```

```
##  [1]  8  7  3  1 12  6  9 10  4  5  2 11
```

```r
# Prints the smallest and largest amount of hourly rain and hourly snow in mm
range(rain_1h)
```

```
## [1]   1 372
```

```r
range(snow_1h)
```

```
## [1]  1 12
```

```r
# Prints statistics for temperature (in kelvins), cloud coverage, and hourly traffic volume
summary(temp)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1    2179    3342    3235    4422    5840
```

```r
summary(clouds_all)
```

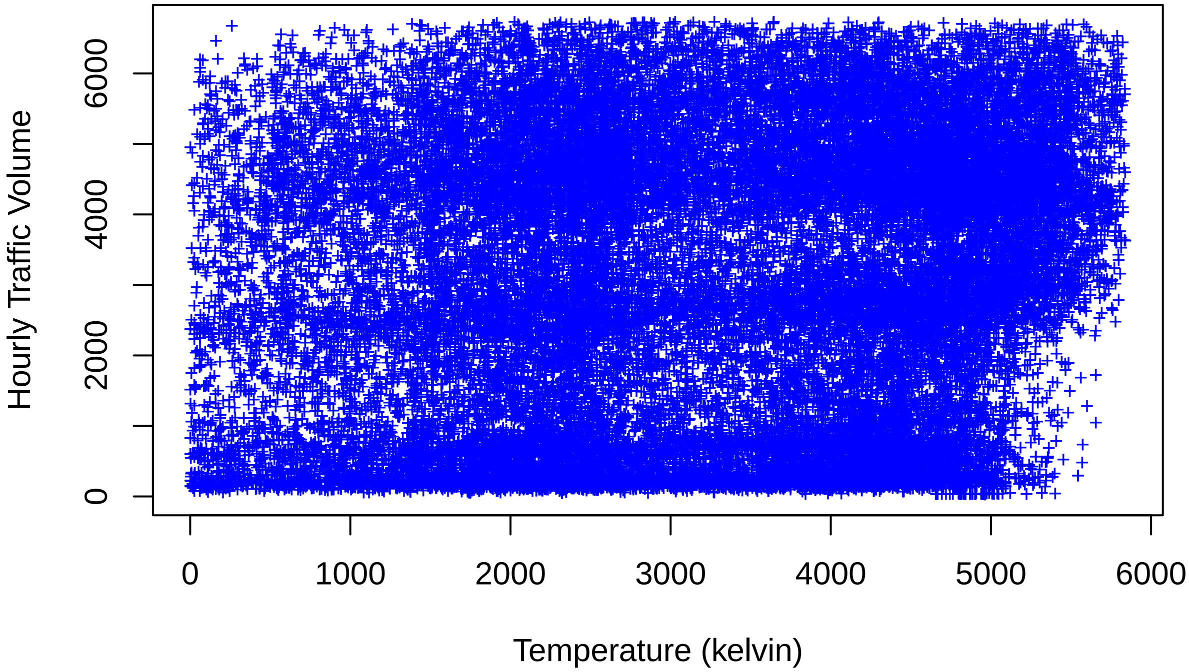
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    2.00   34.00   28.24   52.00   60.00
```

```r
summary(traffic_volume)
```

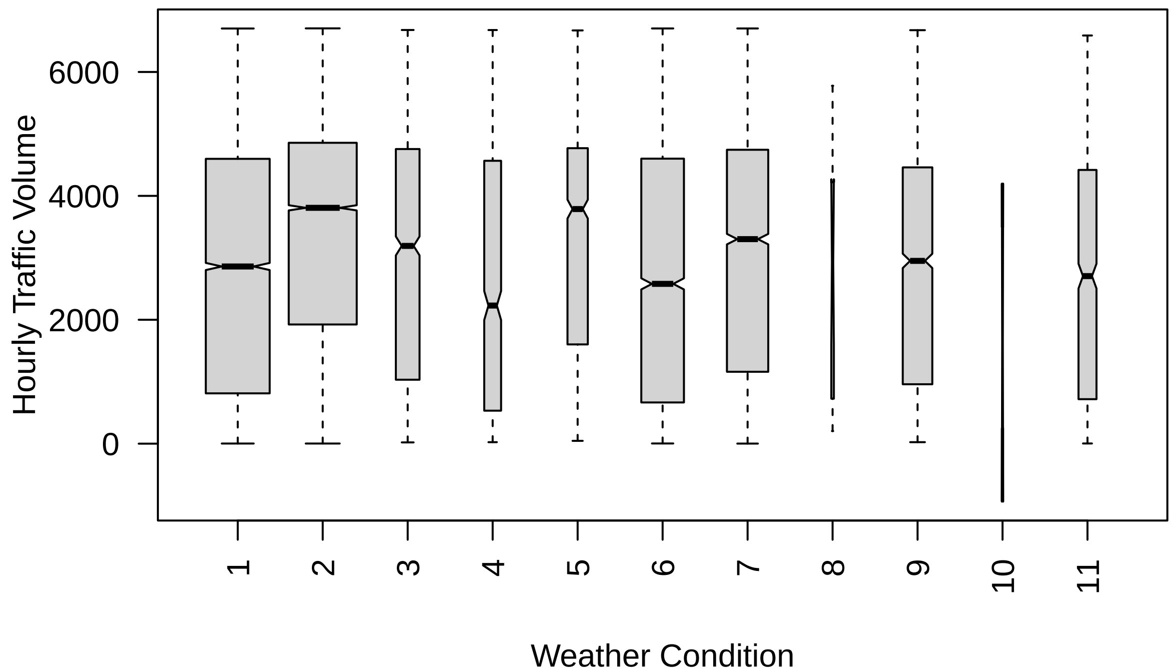
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1    1032    3166    3056    4711    6704
```

```r
# Removes rows with missing temperature data
train <- train[train$temp != 0, ]

# Creates a scatter plot of temperature vs. traffic volume
plot(train$temp, train$traffic_volume, pch='+', cex=0.75, col="blue", xlab="Temperature (kelvin)", ylab=
```




```
# Creates a box plot of the traffic volume based on the weather condition
boxplot(train$traffic_volume~train$weather_main, varwidth=TRUE, notch=TRUE, xlab="Weather Condition", yl
```



Here is a linear model using all of the numeric predictors from the data file.

Linear regression

```
lm1 <- lm(traffic_volume~.-(date_time+weather_description+weather_main+holiday), data=train)

# Output the summary of the model
summary(lm1)
```

```
##
## Call:
## lm(formula = traffic_volume ~ . - (date_time + weather_description +
##     weather_main + holiday), data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3585.8 -1908.6    94.4  1628.5  4371.7
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.229e+03  5.337e+01  41.768   <2e-16 ***
## temp         2.041e-01  7.228e-03  28.234   <2e-16 ***
## rain_1h     -2.507e+00  2.942e-01  -8.522   <2e-16 ***
## snow_1h     -1.798e+01  4.383e+01  -0.410    0.682
```

```
## clouds_all   7.175e+00  4.648e-01  15.437   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1937 on 38558 degrees of freedom
## Multiple R-squared:  0.02374,    Adjusted R-squared:  0.02364
## F-statistic: 234.4 on 4 and 38558 DF,  p-value: < 2.2e-16
```

The results from lm1 indicated that temp and clouds_all are the significant predictors out of all of the numeric predictors. Using these two predictors, here is another linear model.

```r
lm2 <- lm(traffic_volume~temp+clouds_all, data=train)
pred <- predict(lm2, newdata=test)
cor_lm <- cor(pred, test$traffic_volume)
mse_lm <- mean((pred - test$traffic_volume)^2)
print(paste("cor=", cor_lm))
```

```
## [1] "cor= 0.17680646395887"
```

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

```
## [1] "mse= 3719969.57248902"
```

kNN regression

```r
library(caret)
```

Before scaling data

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

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

```r
# fit the model
fit <- knnreg(train[,2:8],train[,1],k=3)

# evaluate
pred2 <- predict(fit, test[,2:8])
cor_knn1 <- cor(pred2, test$traffic_volume)
mse_knn1 <- mean((pred2 - test$traffic_volume)^2)
print(paste("cor=", cor_knn1))
```

```
## [1] "cor= 0.029190148116397"
```

```r
print(paste("mse=", mse_knn1))
```

```
## [1] "mse= 13086930.3283431"
```

These are the results of the correlation and mean squared error for kNN regression before scaling the data.

```r
train_scaled <- train[, 2:8]  # omit name and don't scale mpg
means <- sapply(train_scaled, mean)
stdvs <- sapply(train_scaled, sd)
train_scaled <- scale(train_scaled, center=means, scale=stdvs)
test_scaled <- scale(test[, 2:8], center=means, scale=stdvs)
```

Scale the data

```r
fit <- knnreg(train_scaled, train$traffic_volume, k=3)
pred3 <- predict(fit, test_scaled)
cor_knn2 <- cor(pred3, test$traffic_volume)
mse_knn2 <- mean((pred3 - test$traffic_volume)^2)
print(paste("cor=", cor_knn2))
```

After scaling data

```
## [1] "cor= 0.373828615195925"
```

```r
print(paste("mse=", mse_knn2))
```

```
## [1] "mse= 3717689.07704552"
```

These are the results of the correlation and mean squared error for kNN regression before scaling the data. As seen above, the correlation was much higher and the mean squared error was much lower, using kNN, after scaling the data.

Decision tree regression

```r
library(tree)
tree1 <- tree(temp~., data=train)
summary(tree1)
```

Using unpruned tree

```
##
## Regression tree:
## tree(formula = temp ~ ., data = train)
## Variables actually used in tree construction:
## [1] "date_time"
## Number of terminal nodes:  16
## Residual mean deviance:  471100 = 1.816e+10 / 38550
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -2182.00  -443.10    45.29     0.00   471.00  2654.00
```

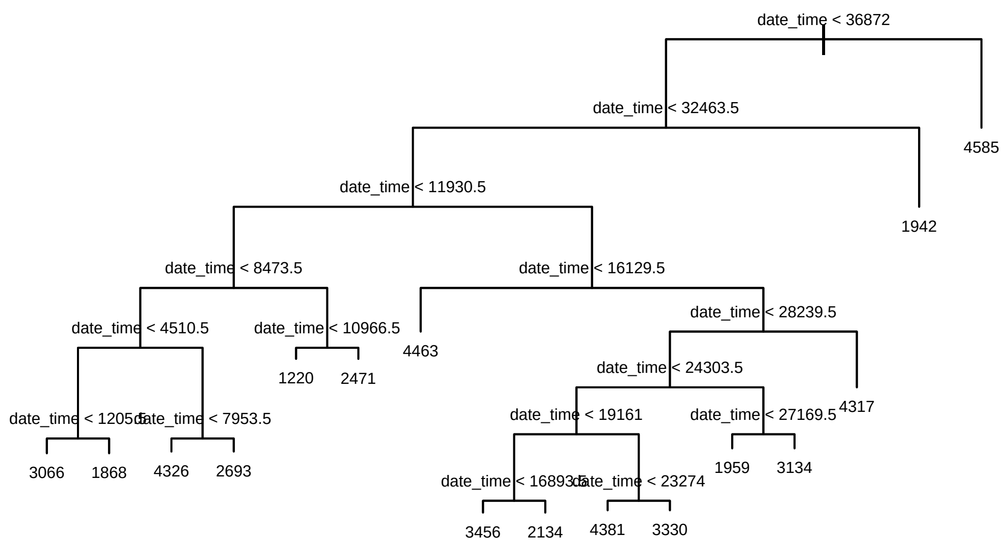
```r
pred <- predict(tree1, newdata=test)
print(paste("cor=", cor(pred, test$temp)))
```

```
## [1] "cor= 0.868742314117584"
```

```r
rmse_tree <- sqrt(mean((pred-test$temp)^2))
print(paste("rmse=", rmse_tree))
```

```
## [1] "rmse= 686.947821562118"
```

```r
plot(tree1)
text(tree1, cex=0.5, pretty=0)
```



This is the decision tree, correlation, and root mean squared error for the unpruned tree of the data. As seen, the results are much better than the kNN regression and linear regression, shown previously.

```r
library(tree)
tree_pruned <- prune.tree(tree1, best=5)
summary(tree_pruned)
```

Testing pruned tree

```
##
## Regression tree:
## snip.tree(tree = tree1, nodes = c(17L, 33L, 32L, 9L))
## Variables actually used in tree construction:
## [1] "date_time"
## Number of terminal nodes:  6
## Residual mean deviance:  1083000 = 4.174e+10 / 38560
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -3604.0  -682.4   115.1     0.0   754.8  3189.0
```

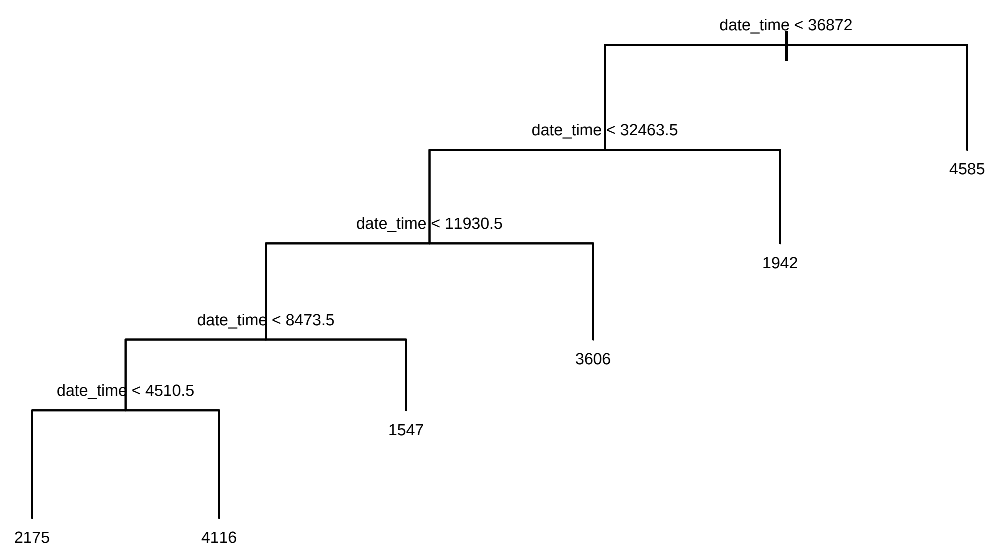
```
pred <- predict(tree_pruned, newdata=test)
print(paste("cor=", cor(pred, test$temp)))
```

```
## [1] "cor= 0.663533183243995"
```

```
rmse_tree <- sqrt(mean((pred-test$temp)^2))
print(paste("rmse=", rmse_tree))
```

```
## [1] "rmse= 1037.71483985172"
```

```
plot(tree_pruned)
text(tree_pruned, cex=0.5, pretty=0)
```



After the tree is pruned to 5 terminal nodes, this is the plot, correlation, and root mean squared error of the pruned tree. Although the results are not as good as the unpruned tree, they are still much higher than the kNN regression and linear regression results.

Analysis

The results for Decision tree regression were much better than kNN regression and linear regression shown in above. This is because decision trees support non-linear solutions, while linear regression is best performed on solely linear data sets. In this case, the decision tree has better accuracy than the linear regression results. Similarly, kNN regression is slower and more inaccurate than the decision tree method because decision tree regression better supports multi-variable regression.