Regression

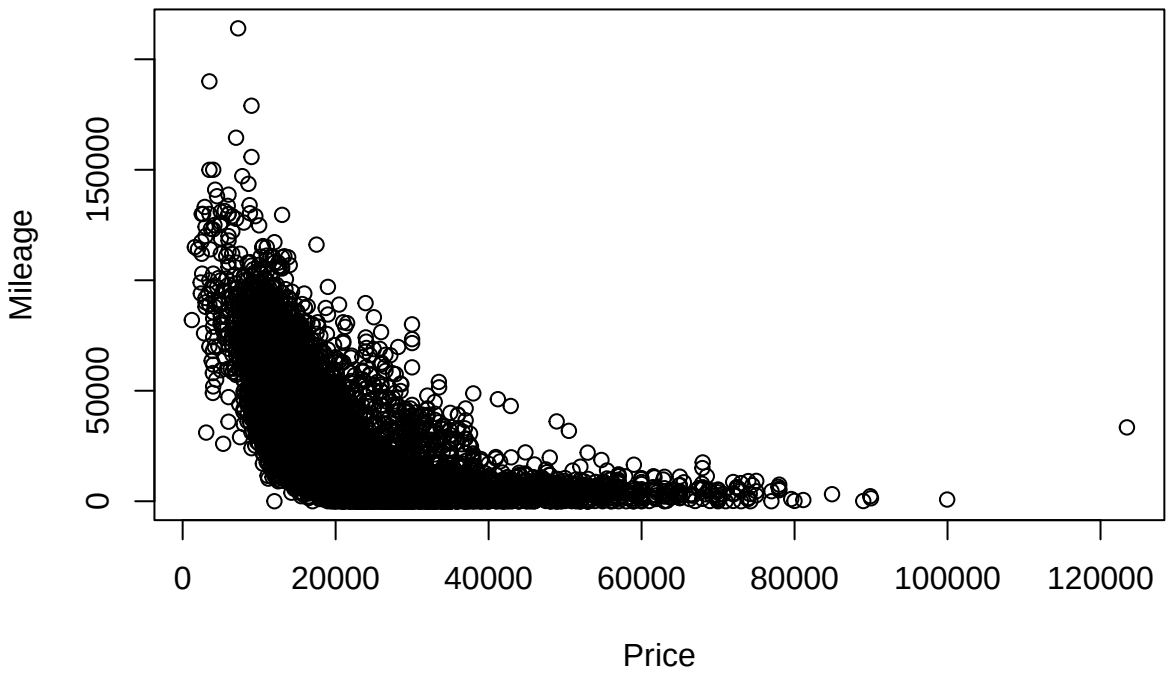
The dataset used here is information regarding used car sales in the UK, specifically BMW sales. Used Car dataset.

Here is where we read in the data and divide it into training and testing data.

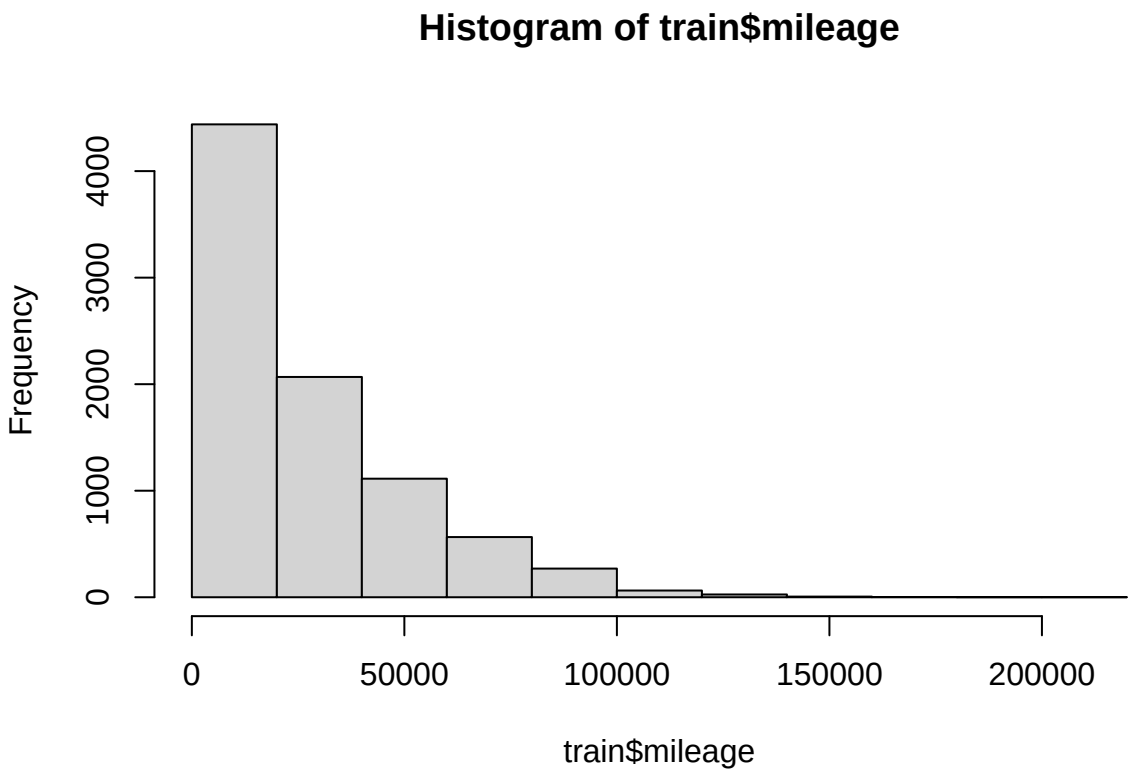
```
library(readr)

df <- read.csv("bmw.csv")

set.seed(1)

sample <- sample(c(TRUE,FALSE), nrow(df), replace = TRUE, prob=c(.8,.2))

train <- df[sample,]
test <-  df[!sample,]
```

Next lets take a look at the data in a few different ways.

```
plot(train$price, train$mileage, xlab="Price", ylab="Mileage")
```




```
hist(train$mileage)
```




```
range(train$mileage)
```

```
## [1]      1 214000
```

```
range(train$price)
```

```
## [1]   1200 123456
```

Here is the linear regression and data, predicting price on year, mileage, tax, mpg, and enginesize.

```
lm1 <- lm(price~year+mileage+tax+mpg+engineSize, data = train)
pred1 <- predict(lm1, newdata = test)
cor1 <- cor(pred1, test$price)
mse1 <- mean((pred1-test$price)^2)
rmse1 <- sqrt(mse1)

print(paste('correlation:', cor1))
```

```
## [1] "correlation: 0.802922640586992"
```

```
print(paste('mse:', mse1))
```

```
## [1] "mse: 44839997.8836365"
```

```
print(paste('rmse:', rmse1))
```

```
## [1] "rmse: 6696.26745908767"
```

Here we can see a fairly high correlation which means we can accuratly predict price based on the other data

points. The large mse and rmse can probably be explained with the extremely large outliers seen in the data exploration stage from before as well as overfitting of the data by the linear regression.

Here is the kNN regression and data, using the same values as before. We use a k value of 3 because I found that 3 gave the best correlation and the lowest mse.

```
library(caret)
```

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

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

```
fit <- knnreg(price~year+mileage+tax+mpg+engineSize, data = train, k = 3)
pred2 <- predict(fit, newdata = test)
cor2 <- cor(pred2, test$price)
mse2 <- mean((pred2-test$price)^2)
rmse2 <- sqrt(mse2)

print(paste('correlation:', cor2))
```

```
## [1] "correlation: 0.727718664929371"
```

```
print(paste('mse:', mse2))
```

```
## [1] "mse: 60322194.8679245"
```

```
print(paste('rmse:', rmse2))
```

```
## [1] "rmse: 7766.73643610522"
```

Here we can see a lower but still somewhat high correlation which suggests that used car price is similar across different cars with other similar traits but follows a more parametric line than similarity to its neighbors. The mse and rmse are also higher which supports this idea.

Lets take a look at Decision Tree regression using all the same values.

```
library(tree)
tree1 <- tree(price~year+mileage+tax+mpg+engineSize, data = train)

pred3 <- predict(tree1, newdata = test)
cor3 <- cor(pred3, test$price)
mse3 <- mean((pred3-test$price)^2)
rmse3 <- sqrt(mse3)

print(paste('correlation:', cor3))
```

```
## [1] "correlation: 0.876123322843842"
```

```
print(paste('mse:', mse3))
```

```
## [1] "mse: 29343792.1409581"
```

```
print(paste('rmse:', rmse3))
```

```
## [1] "rmse: 5416.99105970816"
```

The decision tree approach produced a greater than 7 percent increase in correlation from the previous highest being linear regression. The mse and rmse are also the lowest yet. This indicates that these used cars better

fall into groups than a linear regression, which contradicts the results from kNN. I think that kNN had a worse result because of outliers where as decision trees are not affected by them as much.

#Overall Results

Overall decision trees worked best, which means that the used cars here can be divided into groups well. Since decision trees break the data into smaller groups until each is roughly the same but kNN simply looks at the closest K neighbors, this could explain why kNN had the lowest correlation of the three. Linear regression and DT aren't affected as much and so had higher correlations. Despite this all three methods had high correlations, so used car prices could somewhat accurately be predicted with any one of them.