```
Call:
 randomForest(formula = price ~ ., data = df)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 4

          Mean of squared residuals: 4134536
                    % Var explained: 93.65
> plot(m1)
> which.min(m1$mse)
[1] 97
> sqrt(m1$mse[which.min(m1$mse)])
[1] 2013.531
```
we can find which number of tress providing the lowest error rate and the result is 97 trees providing an average car sales price error of $2013.53

```
model_randomforest <- train(price~., data = df, method = "rf",
trControl = trainControl("cv", number = 13), importance = TRUE)
model_randomforest$bestTune
mtry
2    7
model_randomforest$finalModel
Call:
 randomForest(x = x, y = y, mtry = param$mtry, importance = TRUE)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 7

          Mean of squared residuals: 4177836
                    % Var explained: 93.58
```
#in this method, we used the caret workflow, which invoked the randomforest() function to automatically select the optimal number(mtry) [which is 7]of predictor variables randomly sampled as candidates at each split

```
> varImp(m1)
                   Overall
highwaympg       1109093447
citympg          1722186229
peakrpm           122418726
horsepower       1462668319
compressionratio  109739950
stroke             65811137
bore              167225932
enginesize       3285178939
curbweight       2191885663
height             79562142
width             884431084
length            488106057
wheelbase         389420029
Warning messages:
```