House Price Model

Load Libraries required

```r
#For map Visualisation
library(ggmap)
```

```
## Loading required package: ggplot2

## Google's Terms of Service: https://cloud.google.com/maps-platform/terms/.

## Please cite ggmap if you use it! See citation("ggmap") for details.
```

```r
library(ggplot2)
library(tidyr)
#For Tree
library(rpart)
library(rpart.plot)
#For Linear Regression
library(caTools)
#For Cross Validation
library(caret)
```

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

```r
library(e1071)
```

Read Data Files

```r
boston =read.csv('boston.csv')
str(boston)
```

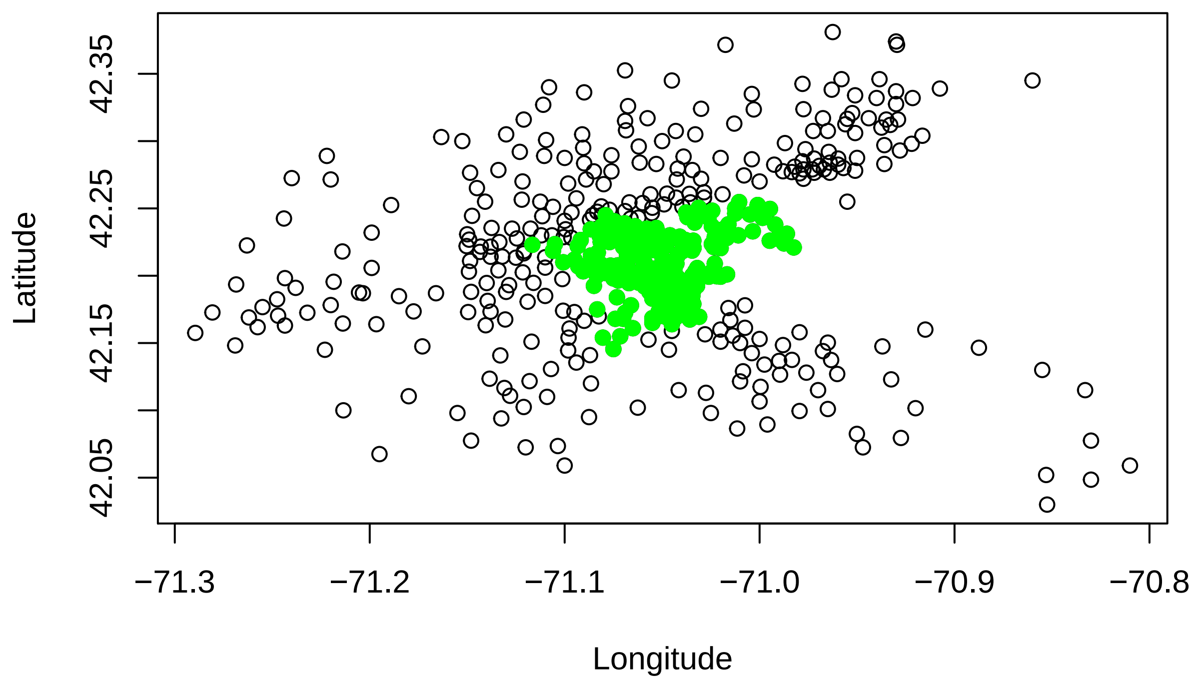
```
## 'data.frame':    506 obs. of  16 variables:
##  $ TOWN   : chr  "Nahant" "Swampscott" "Swampscott" "Marblehead" ...
##  $ TRACT  : int  2011 2021 2022 2031 2032 2033 2041 2042 2043 2044 ...
##  $ LON    : num  -71 -71 -70.9 -70.9 -70.9 ...
##  $ LAT    : num  42.3 42.3 42.3 42.3 42.3 ...
##  $ MEDV   : num  24 21.6 34.7 33.4 36.2 28.7 22.9 22.1 16.5 18.9 ...
##  $ CRIM   : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
##  $ ZN     : num  18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
##  $ INDUS  : num  2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...
##  $ CHAS   : int  0 0 0 0 0 0 0 0 0 0 ...
```

```
##  $ NOX    : num  0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.524 ...
##  $ RM     : num  6.58 6.42 7.18 7 7.15 ...
##  $ AGE    : num  65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...
##  $ DIS    : num  4.09 4.97 4.97 6.06 6.06 ...
##  $ RAD    : int  1 2 2 3 3 3 5 5 5 5 ...
##  $ TAX    : int  296 242 242 222 222 222 311 311 311 311 ...
##  $ PTRATIO: num  15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...
```

```
plot(boston$LON,boston$LAT, xlab = "Longitude", ylab= "Latitude")
summary(boston$NOX)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.3850  0.4490  0.5380  0.5547  0.6240  0.8710
```

```
points(boston$LON[boston$NOX >=.55],
       boston$LAT[boston$NOX >=.55], col = "green",pch =19)
```

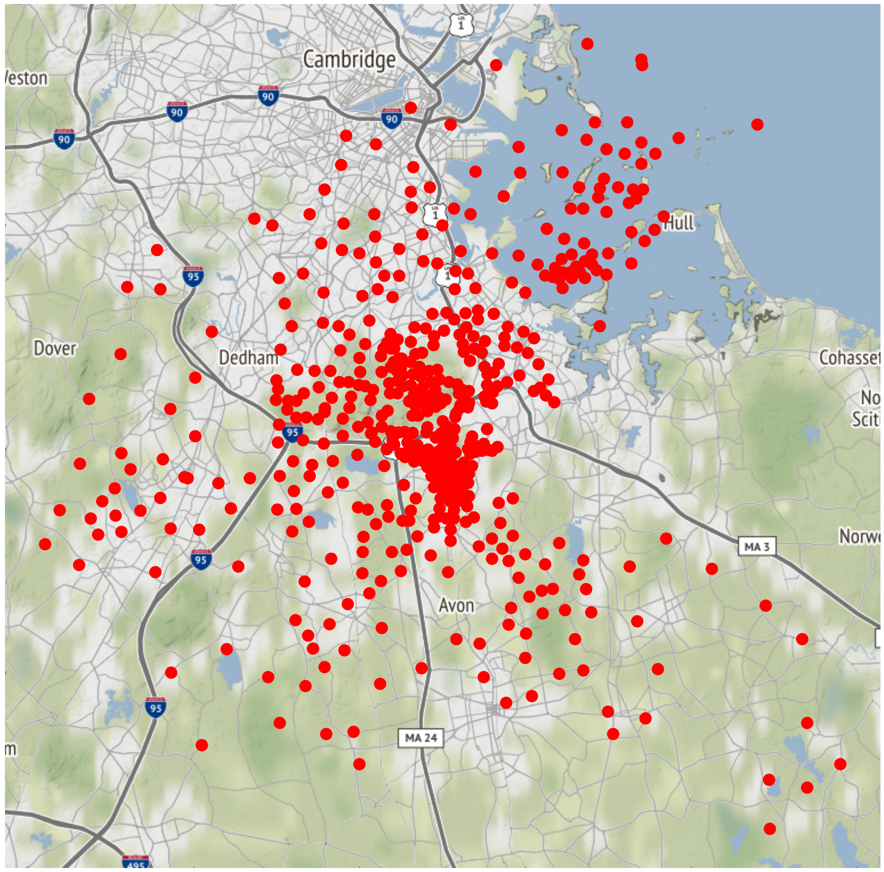



```
#Map Visualisation
qmplot(LON, LAT, data = boston,
       maptype = "toner-lite", color = I("red"))
```

```
## Using zoom = 11...
```

```
## Source : http://tile.stamen.com/terrain/11/618/757.png
```

```
## Source : http://tile.stamen.com/terrain/11/619/757.png
## Source : http://tile.stamen.com/terrain/11/620/757.png
## Source : http://tile.stamen.com/terrain/11/621/757.png
## Source : http://tile.stamen.com/terrain/11/618/758.png
## Source : http://tile.stamen.com/terrain/11/619/758.png
## Source : http://tile.stamen.com/terrain/11/620/758.png
## Source : http://tile.stamen.com/terrain/11/621/758.png
## Source : http://tile.stamen.com/terrain/11/618/759.png
## Source : http://tile.stamen.com/terrain/11/619/759.png
## Source : http://tile.stamen.com/terrain/11/620/759.png
## Source : http://tile.stamen.com/terrain/11/621/759.png
## Source : http://tile.stamen.com/terrain/11/618/760.png
## Source : http://tile.stamen.com/terrain/11/619/760.png
## Source : http://tile.stamen.com/terrain/11/620/760.png
## Source : http://tile.stamen.com/terrain/11/621/760.png
```



Linear Regression Model

```
LinReg = lm(MEDV ~ LAT + LON , data = boston)
summary(LinReg)
```

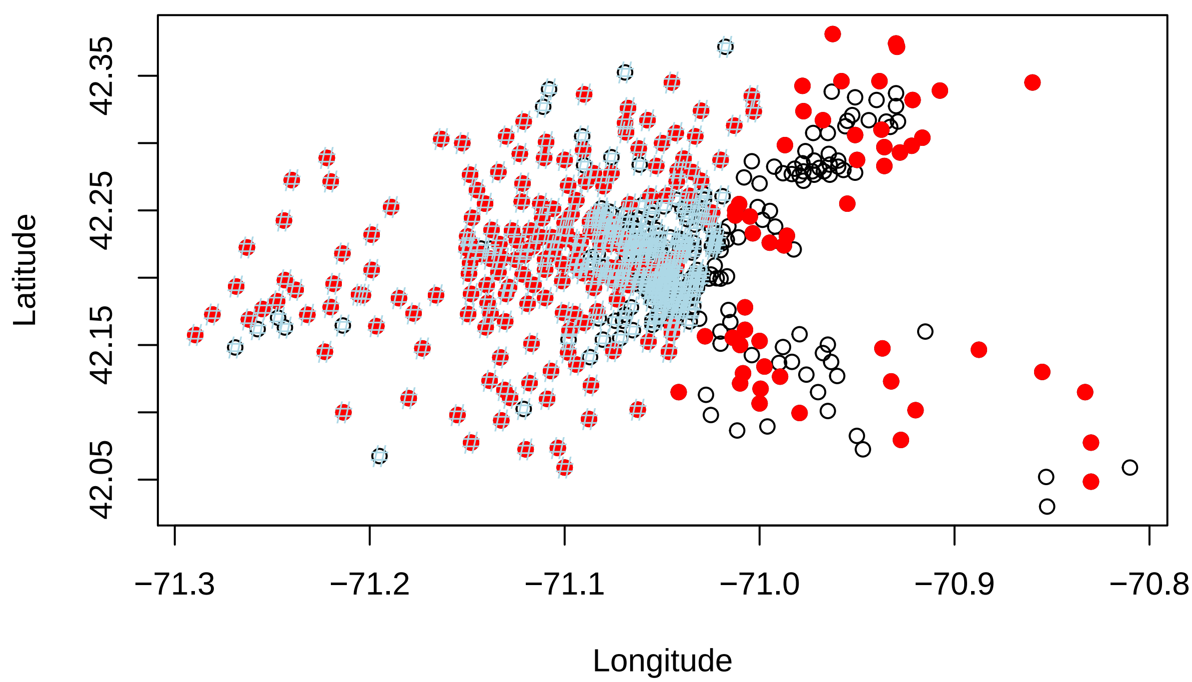
```
##
## Call:
## lm(formula = MEDV ~ LAT + LON, data = boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.460  -5.590  -1.299   3.695  28.129
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3178.472    484.937  -6.554 1.39e-10 ***
## LAT             8.046      6.327   1.272    0.204
## LON           -40.268      5.184  -7.768 4.50e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.693 on 503 degrees of freedom
## Multiple R-squared:  0.1072, Adjusted R-squared:  0.1036
## F-statistic: 30.19 on 2 and 503 DF,  p-value: 4.159e-13
```

Checking Fit

```r
plot(boston$LON,boston$LAT, xlab = "Longitude", ylab= "Latitude")
summary(boston$MEDV)
```

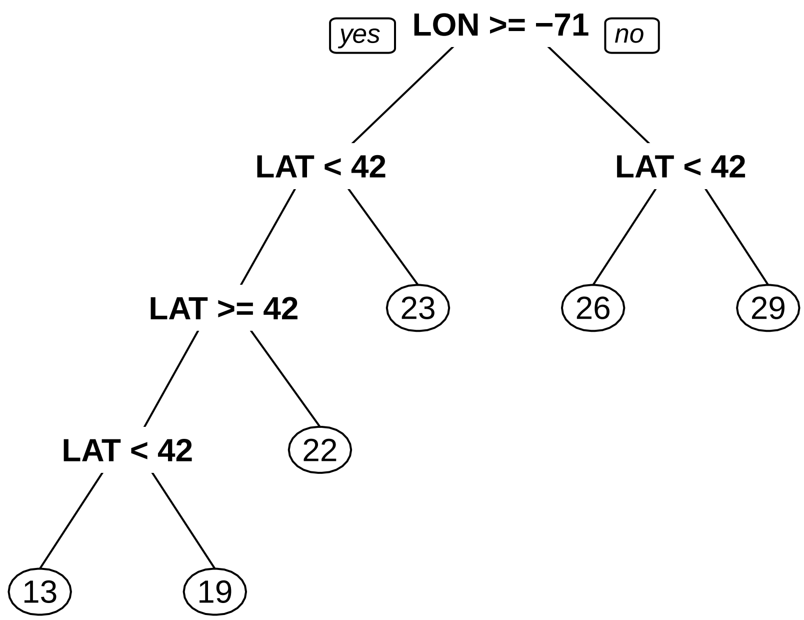
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    5.00   17.02   21.20   22.53   25.00   50.00
```

```r
points(boston$LON[boston$MEDV >=21.2],
       boston$LAT[boston$MEDV >=21.2],
       col = "red",pch =19)
points(boston$LON[LinReg$fitted.values >=21.2],
       boston$LAT[LinReg$fitted.values >=21.2],
       col = "lightblue",pch ="#")
```



Building Tree Model

```r
treeMod = rpart(MEDV ~ LAT + LON, data = boston, minbucket = 50)
prp(treeMod)
pred <- predict(treeMod)
points(boston$LON[pred >=21.2],boston$LAT[pred >=21.2],
       col = "lightblue",pch ="$")
```



##Building Linear Model using all variables

```
split = sample.split(boston$MEDV, SplitRatio = .7)
train = subset(boston, split == TRUE)
test = subset(boston, split == F)

linReg = lm(MEDV ~ LAT + LON + CRIM + ZN +
              INDUS + CHAS + NOX + RM + AGE +
              RAD + TAX + PTRATIO, data = train)
summary(linReg)
```

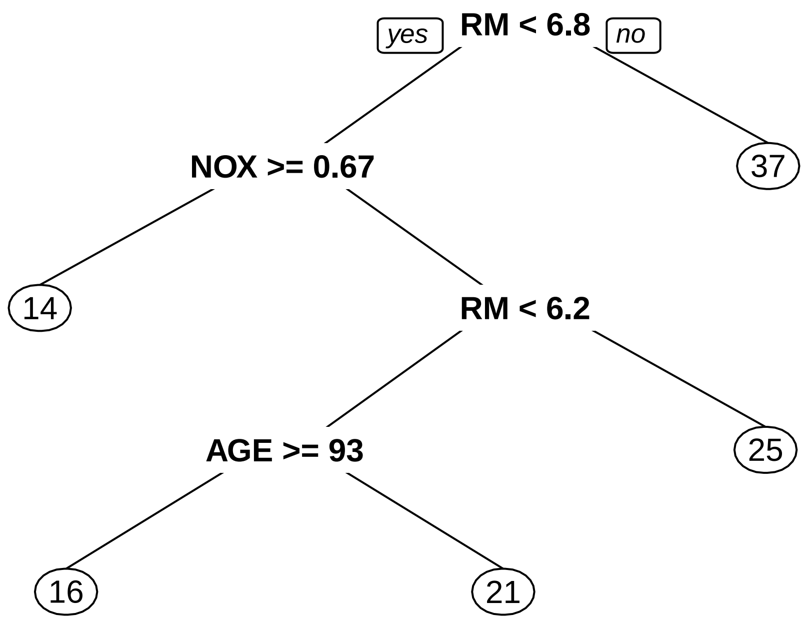
```
##
## Call:
## lm(formula = MEDV ~ LAT + LON + CRIM + ZN + INDUS + CHAS + NOX +
##     RM + AGE + RAD + TAX + PTRATIO, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.559  -2.955  -0.520   1.951  31.850
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.059e+03  4.285e+02  -2.472   0.0139 *
## LAT          8.043e+00  5.198e+00   1.547   0.1227
## LON         -1.023e+01  4.639e+00  -2.206   0.0280 *
## CRIM        -1.055e-01  5.066e-02  -2.082   0.0381 *
```

```
## ZN          -1.518e-02  1.769e-02  -0.858   0.3913
## INDUS        1.745e-02  8.850e-02   0.197   0.8438
## CHAS         2.682e+00  1.144e+00   2.343   0.0197 *
## NOX         -9.723e+00  5.022e+00  -1.936   0.0536 .
## RM           6.819e+00  4.667e-01  14.611  < 2e-16 ***
## AGE         -1.490e-02  1.633e-02  -0.913   0.3621
## RAD          1.798e-01  9.526e-02   1.887   0.0600 .
## TAX         -1.194e-02  5.527e-03  -2.160   0.0315 *
## PTRATIO     -9.654e-01  1.840e-01  -5.247 2.68e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.547 on 351 degrees of freedom
## Multiple R-squared:  0.6698, Adjusted R-squared:  0.6585
## F-statistic: 59.33 on 12 and 351 DF,  p-value: < 2.2e-16
```

```
linreg.pred = predict(linReg, newdata = test)
SSE.linreg = sum((linreg.pred - test$MEDV)**2)
```

Building Tree Model using all Variables

```
tree = rpart(MEDV ~ LAT + LON + CRIM + ZN +
              INDUS + CHAS + NOX + RM +
               AGE + RAD + TAX + PTRATIO, data = train, minbucket = 25)
prp(tree)
```

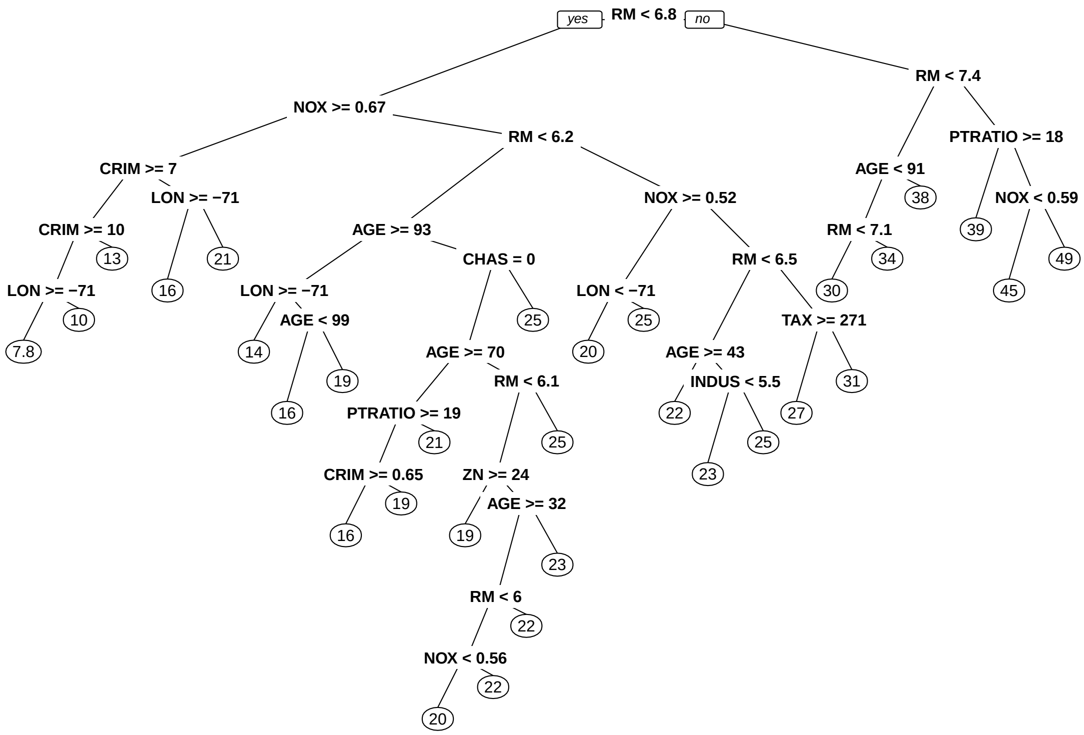



```
tree.pred = predict(tree,newdata=  test)
SSE.tree = sum((tree.pred - test$MEDV)**2)
```

SSE for tree model using 6263 whereas the SSE for linear Regression model is 4521

Cross Validation to obtain the best tree

```
tr.control = trainControl(method ='cv', number = 10)
cp.grid =expand.grid(.cp = (0:10)*0.001)
tr = train(MEDV ~ LAT + LON + CRIM + ZN + INDUS +
           CHAS + NOX + RM +
             AGE + RAD + TAX + PTRATIO,
           data = train,
           method = "rpart",
           trControl = tr.control,
           tuneGrid = cp.grid)
best.tree = tr$finalModel
prp(best.tree)
```




```
best.tree.pred = predict(best.tree,newdata = test)
SSE.best.tree = sum((best.tree.pred - test$MEDV)**2)
```

- SSE for the optimised tree model is 22.69 % lower than the previous tree.
- However, the SSE for linear regression is lower by 6.63 % w.r.t the regression tree.

Citations :

- D. Kahle and H. Wickham. ggmap: Spatial Visualization with ggplot2. The R Journal, 5(1), 144-161. URL http://journal.r-project.org/archive/2013-1/kahle-wickham.pdf