Random forest method analysis

Ajinkya Ghodekar

Importing the dataset

```
dataset = read.csv('transaction1_data.csv')

View(dataset)
```

Taking Care of Missing Data

```
dataset$QUANTITY = ifelse(is.na(dataset$QUANTITY),
                          ave(dataset$QUANTITY, FUN = function(x) mean(x, na.
rm = TRUE)),
                          dataset$QUANTITY)

dataset$QUANTITY = ifelse(0,1,dataset$QUANTITY)



dataset$SALES_VALUE = ifelse(is.na(dataset$SALES_VALUE),
                             ave(dataset$SALES_VALUE, FUN = function(x) mean(
x, na.rm = TRUE)),
                             dataset$SALES_VALUE)
```

Formula

```
dataset$Actual_price = (dataset$SALES_VALUE - (dataset$RETAIL_DISC + dataset$
COUPON_MATCH_DISC)/dataset$QUANTITY)
```

Required Libraries and drop the unnecessary column from the dataset

```
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
mydata <- dataset
```

```
library(sos)
```

```
## Loading required package: brew
```

```
##
## Attaching package: 'sos'
```

```
## The following object is masked from 'package:dplyr':
##
##     matches
```

```
## The following object is masked from 'package:utils':
##
##     ?
```

```
findFn("select")
```

```
## Warning in parseHTML(href): Too many documents hit. Ignored
```

```
## found Inf matches
```

```
## x has zero rows;   nothing to display.
```

```
T = select (mydata, -c(X,household_key,BASKET_ID, DAY, PRODUCT_ID, STORE_ID,
TRANS_TIME, QUANTITY, COUPON_DISC, SALES_VALUE, COUPON_MATCH_DISC, RETAIL_DIS
C ))
```

```
View(T)
```

```
summary(T)
```

```
##     WEEK_NO        Actual_price
##  Min.   : 1.00   Min.   : -0.010
##  1st Qu.:32.00   1st Qu.:  1.580
##  Median :54.00   Median :  2.590
##  Mean   :53.71   Mean   :  3.643
##  3rd Qu.:76.00   3rd Qu.:  3.990
##  Max.   :97.00   Max.   :840.000
```

Agreegate the data

```
u = aggregate( Actual_price ~ WEEK_NO, T, sum)
```

Required library Splitting the dataset into Training set and Test Set

```
# install.packages("caTools")

library(caTools)

set.seed(123)

split = sample.split(u$WEEK_NO, SplitRatio = 0.8)

training_set = subset(u, split == TRUE)

test_set = subset(u, split == FALSE)
```

Random forest Regression

```
# install.packages("randomForest")

library(randomForest)

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine

set.seed(1234)

regressor = randomForest(x = u[1],
                         y = u$Actual_price,
                         ntree = 500)
```

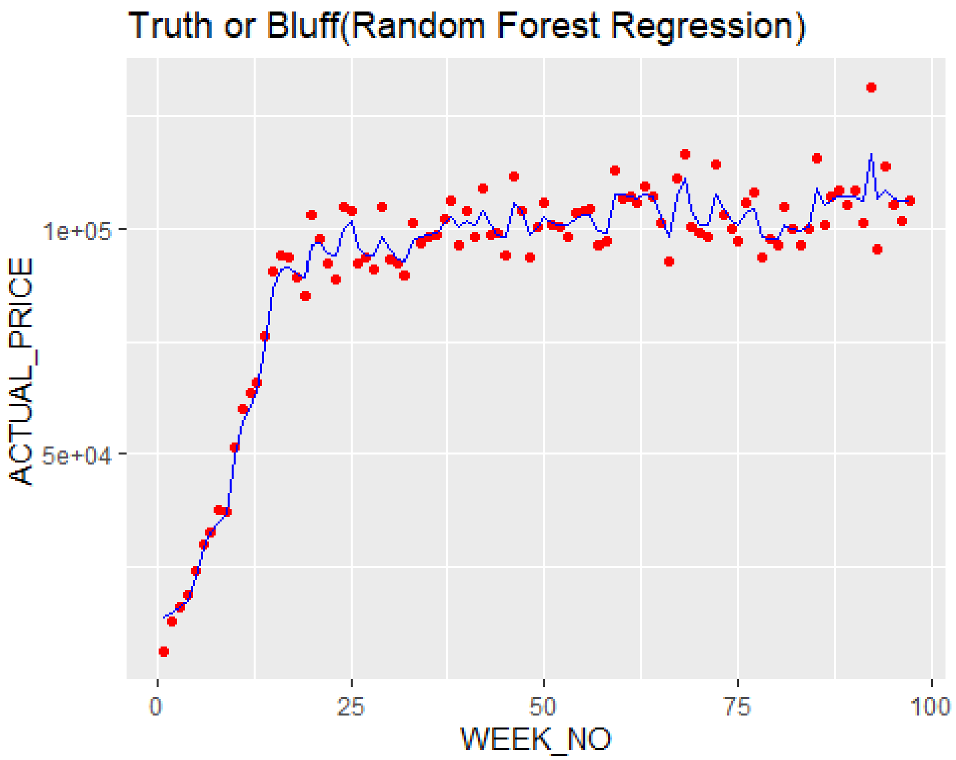
Visualizing Random Forest regression for aggregated data

```
library(ggplot2)

##
## Attaching package: 'ggplot2'

## The following object is masked from 'package:randomForest':
##
##     margin

ggplot() +
  geom_point(aes(x = u$WEEK_NO, y = u$Actual_price),
             colour = 'red') +
  geom_line(aes(x = u$WEEK_NO, y = predict(regressor, newdata = u )),
            colour = 'blue') +
  ggtitle('Truth or Bluff(Random Forest Regression)') +
  xlab('WEEK_NO')+
  ylab('ACTUAL_PRICE')
```

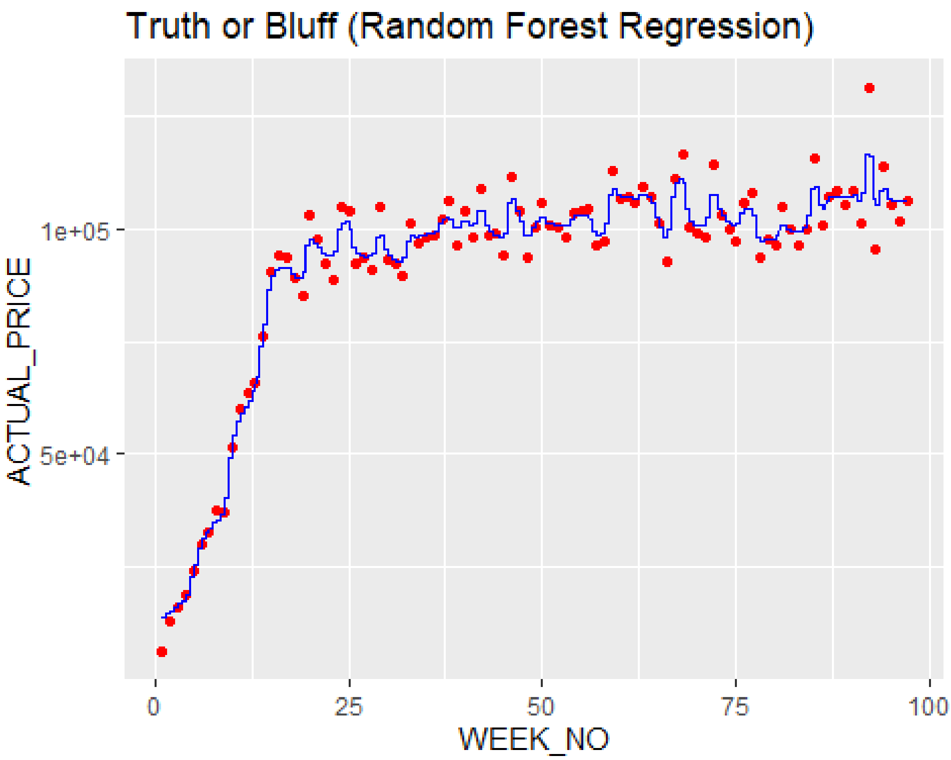


Visualizing Random Forest regression for higher resolution for aggregated data

```
# install.packages('ggplot2')

library(ggplot2)

x_grid = seq(min(u$WEEK_NO), max(u$WEEK_NO), 0.01)
ggplot() +
  geom_point(aes(x = u$WEEK_NO, y = u$Actual_price),
             colour = 'red') +
  geom_line(aes(x = x_grid, y = predict(regressor, newdata = data.frame(WEEK_
NO = x_grid))),
            colour = 'blue') +
  ggtitle('Truth or Bluff (Random Forest Regression)') +
  xlab('WEEK_NO') +
  ylab('ACTUAL_PRICE')
```



Predicting a result for WEEK_NO 98

```
y_pred = predict(regressor, data.frame(WEEK_NO = 98))
y_pred

##        1
## 106389.9
```