# STATS 101C - Statistical Models and Data Mining - Homework 5

*Darren Tsang, Discussion 4B*

Produced on Saturday, Nov. 21 2020 @ 02:59:13 AM

## Question 1 (Exercise 2 from Section 8.4)

If $d = 1$, then you are only making one split (which is based on only one predictor). Then, as seen in Algorithm 8.2, the final model will just be the combination of all these stumps. This leads to a model in the form of given in the question, as desired.

# Question 2 (Exercise 5 from Section 8.4)

### Using majority vote approach

Taking the average:

$$\frac{.1 + .15 + .2 + .2 + .55 + .6 + .6 + .65 + .7 + .75}{10} = .45$$

Since $.45 < .5$, the the final classification is **green**.

### Using average probability approach

There are 4 (.1, .15, .2, .2) that classify as green, and there are 6 (.55, .6, .6, .65, .7, .75) that classify as red. Since there are more that classify as red, the final classification is **red**.

## Question 3 (Exercise 8 from Section 8.4)

```
library(ISLR)
library(caret)
library(MASS)
library(tree)
library(rattle)
library(randomForest)
```

### Part A

```
data(Carseats)
Carseats$ShelveLoc <- as.numeric(Carseats$ShelveLoc)
set.seed(999)
indices <- createDataPartition(Carseats$Sales, p = .8, list = FALSE)

carseats.train <- Carseats[indices, ]
carseats.test <- Carseats[-indices, ]

dim(carseats.train)
```

```
## [1] 321  11
```

```
dim(carseats.test)
```

```
## [1] 79 11
```

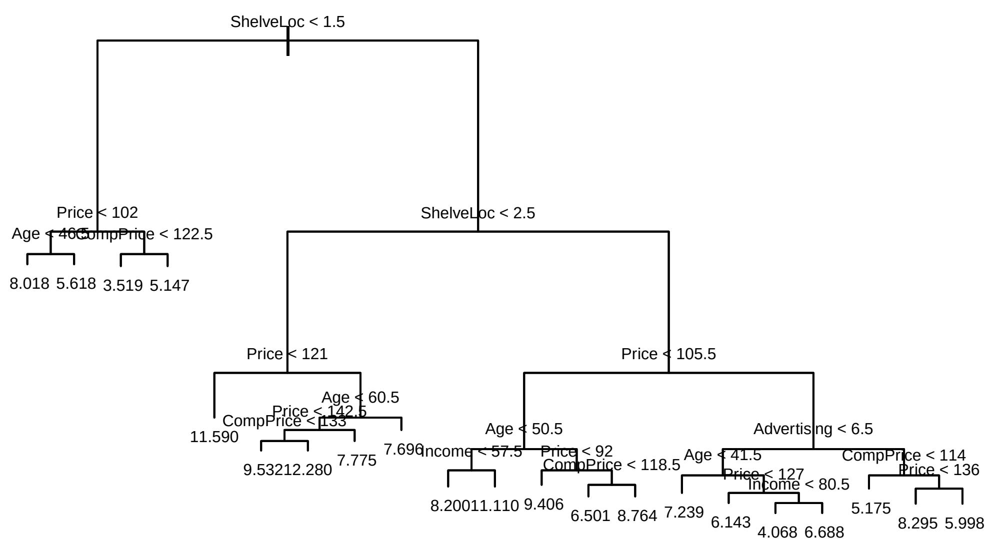
## Part B

```r
base.tree <- tree(Sales ~ ., data = carseats.train)

summary(base.tree)
```

```
##
## Regression tree:
## tree(formula = Sales ~ ., data = carseats.train)
## Variables actually used in tree construction:
## [1] "ShelveLoc"   "Price"       "Age"         "CompPrice"   "Income"
## [6] "Advertising"
## Number of terminal nodes:  21
## Residual mean deviance:  2.241 = 672.2 / 300
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -4.70600 -0.96590 -0.03261  0.00000  0.95280  4.68400
```

```r
plot(base.tree)
text(base.tree, cex = .5)
```




```r
base.tree.predictions <- predict(base.tree, carseats.test)
base.tree.mse <- mean((base.tree.predictions - carseats.test$Sales)^2)
base.tree.mse
```

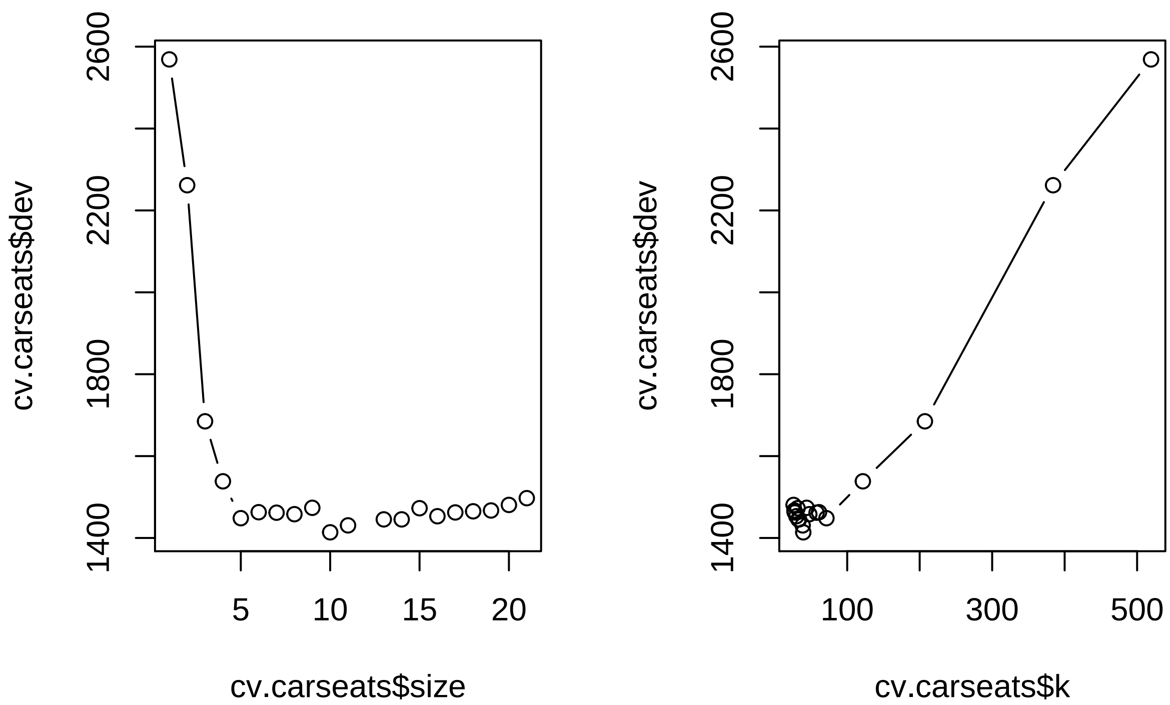
```
## [1] 6.083221
```

The plot of the tree is shown above, which seems pretty messy. It appears that we only used the variables ‘ShelveLoc’, ‘Price’, ‘Age’, ‘CompPrice’, ‘Income’, and ‘Advertising’. The test MSE is 6.083221.

### Part C

```
set.seed(999)
cv.carseats <- cv.tree(base.tree, K = 5)

par(mfrow=c(1, 2))
plot(cv.carseats$size, cv.carseats$dev, type="b")
plot(cv.carseats$k, cv.carseats$dev, type="b")
```

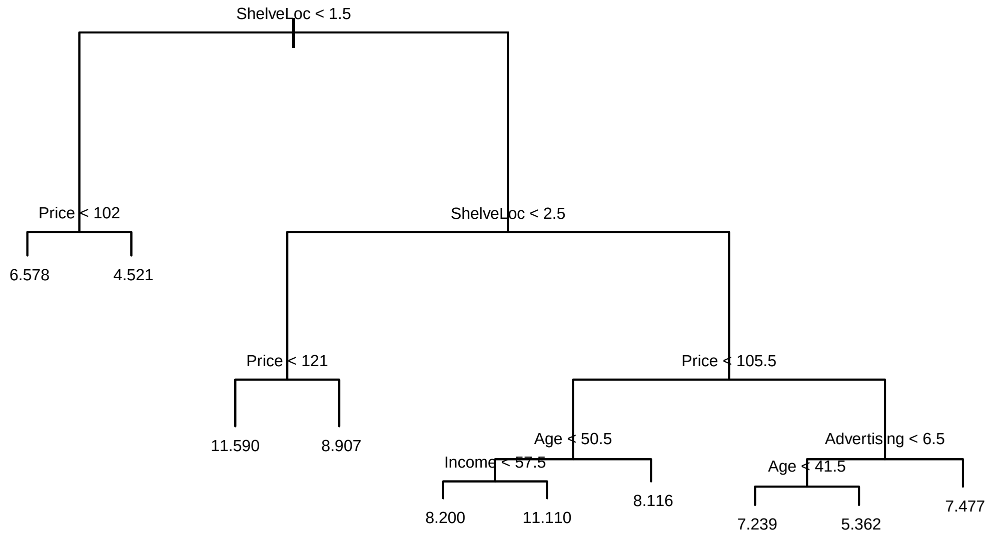



```
best.id <- which.min(cv.carseats$dev)
best.size <- cv.carseats$size[best.id]
best.size
```

```
## [1] 10
```

```
pruned.tree <- prune.tree(base.tree, best = best.size)

plot(pruned.tree)
text(pruned.tree, cex = .5)
```




```r
pruned.tree.predictions <- predict(pruned.tree, carseats.test)
pruned.tree.mse <- mean((pruned.tree.predictions - carseats.test$Sales)^2)
pruned.tree.mse
```

```
## [1] 5.557804
```

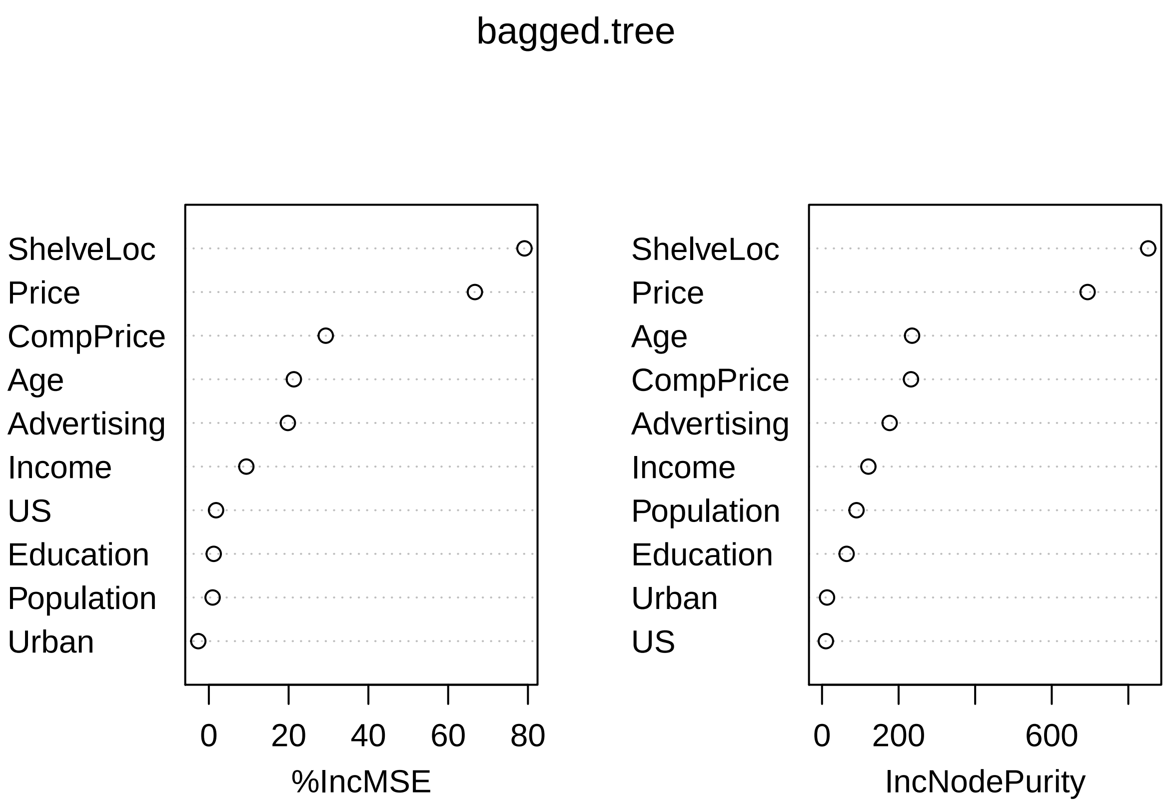
The optimal level is 10. Furthermore, after pruning, the test MSE is 5.557804, which is a decrease from before.

## Part D

```
set.seed(999)
bagged.tree <- randomForest(Sales ~ .,
                            data = carseats.train,
                            mtry = dim(carseats.train)[2] - 1,
                            ntree = 500,
                            importance = T)
print(bagged.tree)
```

```
##
## Call:
##  randomForest(formula = Sales ~ ., data = carseats.train, mtry = dim(carseats.train)[2] -       1, nt
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 10
##
##           Mean of squared residuals: 2.476393
##                     % Var explained: 68.69
```

```
varImpPlot(bagged.tree)
```



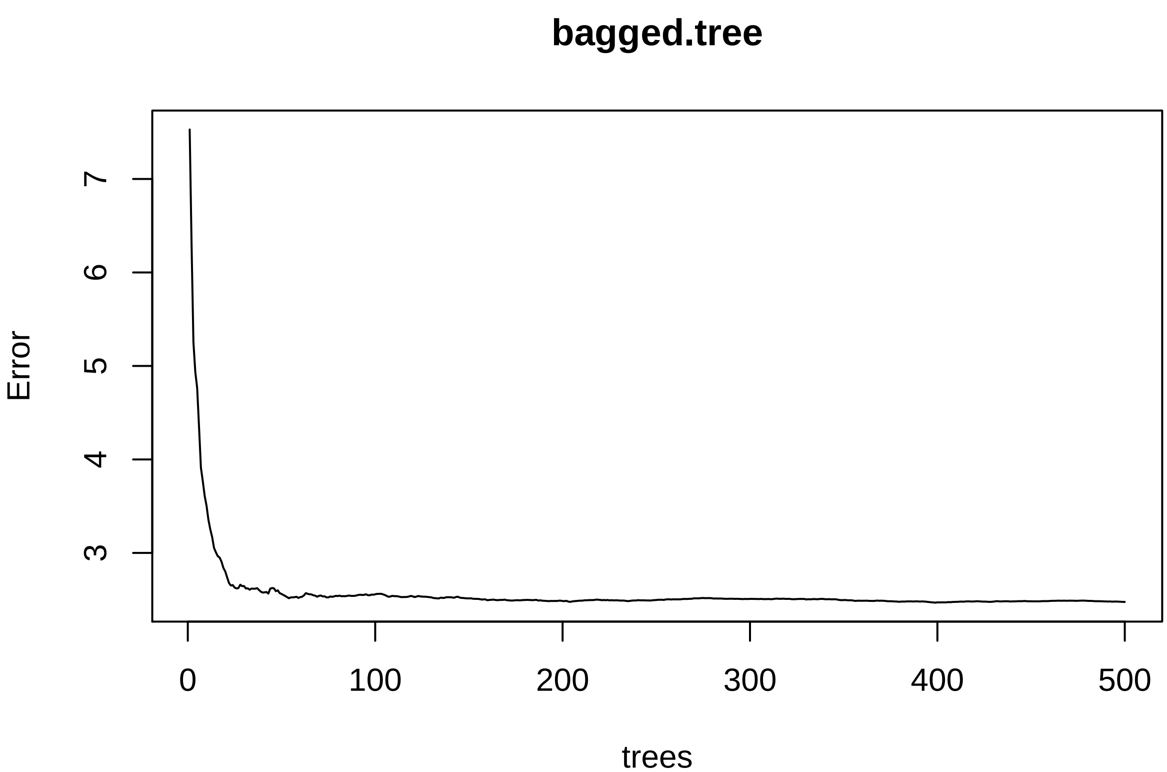

```
bagged.tree.predictions <- predict(bagged.tree, carseats.test)
bagged.tree.mse <- mean((bagged.tree.predictions - carseats.test$Sales)^2)
bagged.tree.mse
```

```
## [1] 3.20789
```

```
importance(bagged.tree)
```

```
##                %IncMSE IncNodePurity
## CompPrice   29.3081775     232.36071
## Income       9.3978773     121.08465
## Advertising 19.8055284     176.61330
## Population   0.9504858      90.05110
## Price       66.6994895     693.64825
## ShelveLoc   79.1357091     851.97497
## Age         21.3087387     235.20261
## Education    1.2190839      64.21132
## Urban       -2.6489751      13.02823
## US           1.8060725      10.02173
```

```
plot(bagged.tree)
```



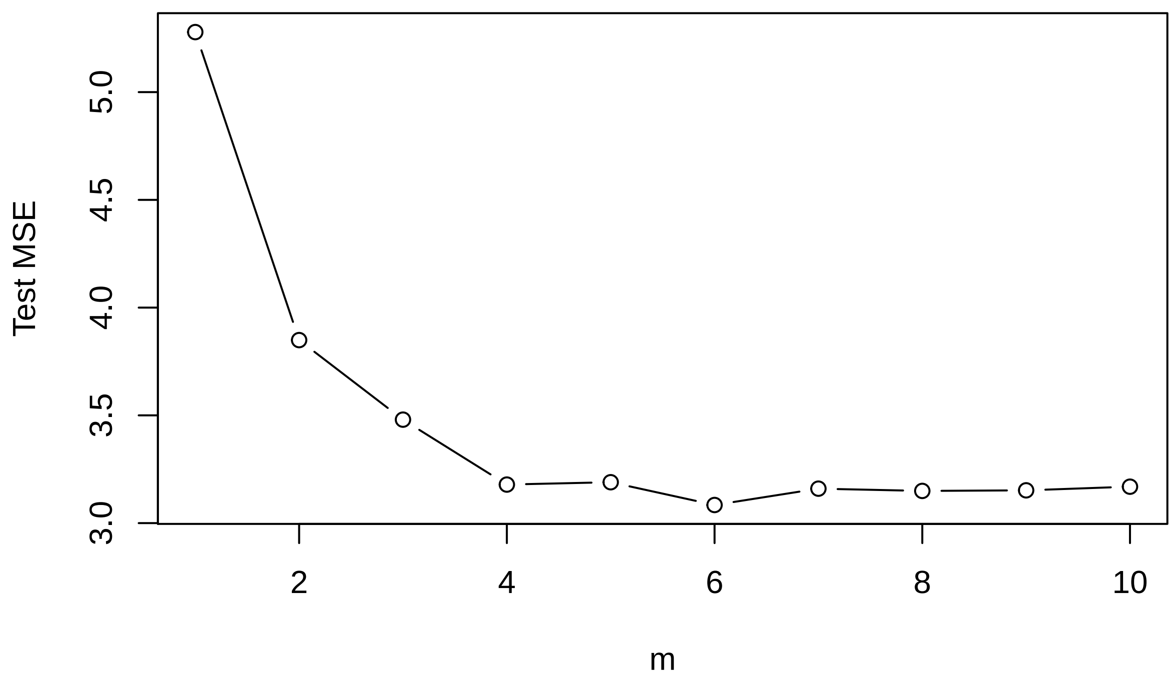

After bagging, the test MSE drops to 3.2078899. The most important variables are 'ShelveLoc', 'Price', 'CompPrice', 'Age', and 'Advertising'.

### Part E

```r
set.seed(999)
test.mse.rf <- rep(0, (dim(carseats.train)[2] - 1))
for (i in 1:(dim(carseats.train)[2] - 1)){
  temp.m <- randomForest(Sales ~ .,
                    data = carseats.train,
                    mtry = i,
                    ntree = 500,
                    importance = TRUE)


  temp.p <- predict(temp.m, carseats.test)
  temp.mse <- mean((temp.p - carseats.test$Sales)^2)
  test.mse.rf[i] <- temp.mse
}

plot(test.mse.rf, type = "b",
     xlab = "m",
     ylab = "Test MSE")
```




```r
mtry.best.rf <- which.min(test.mse.rf)
mtry.best.rf
```

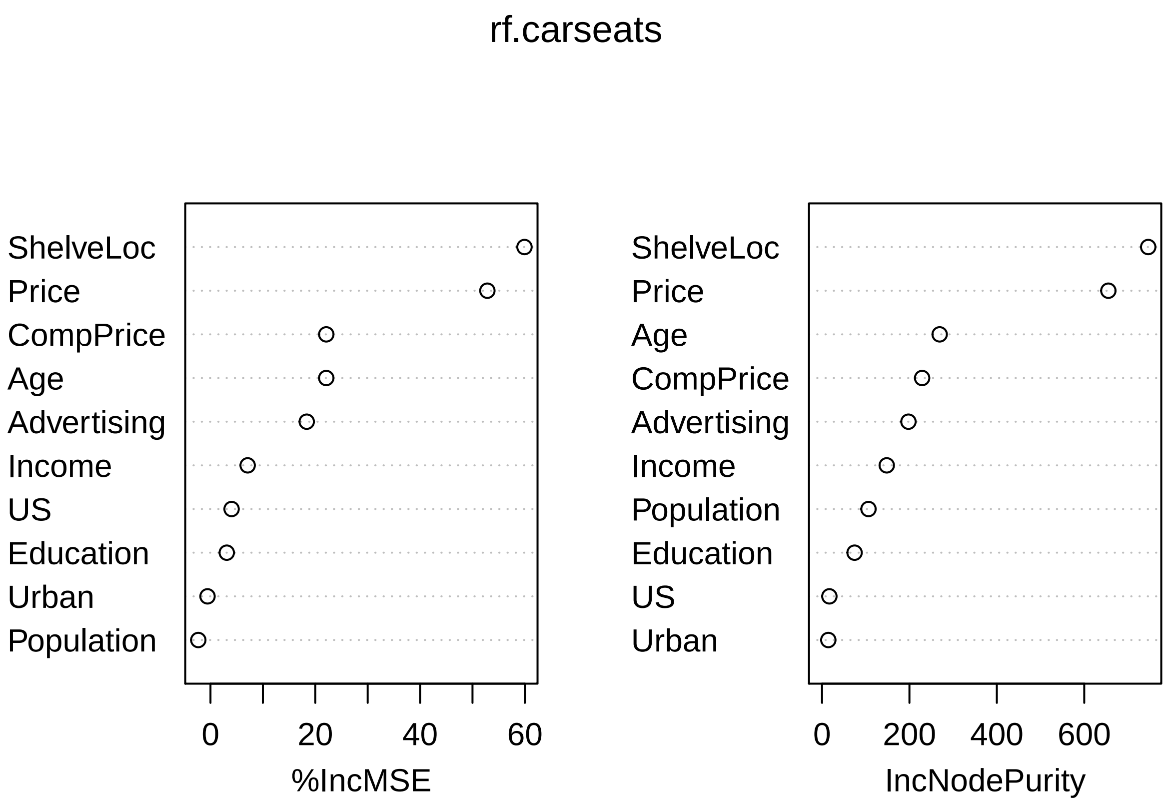
```
## [1] 6
```

```r
rf.carseats <- randomForest(Sales ~ .,
                            data = carseats.train,
                            mtry = mtry.best.rf,
                            ntree = 500,
```

```
                             importance = TRUE)
print(rf.carseats)
```

```
##
## Call:
##  randomForest(formula = Sales ~ ., data = carseats.train, mtry = mtry.best.rf,      ntree = 500, impo
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 6
##
##           Mean of squared residuals: 2.461985
##                     % Var explained: 68.87
```

```
varImpPlot(rf.carseats)
```




```
rf.predictions <- predict(rf.carseats, carseats.test)
rf.carseats.mse <- mean((rf.predictions - carseats.test$Sales)^2)
rf.carseats.mse
```

```
## [1] 3.120469
```

```
importance(rf.carseats)
```

```
##                %IncMSE IncNodePurity
## CompPrice    22.092289     229.07404
```

```
## Income        7.091961     148.09233
## Advertising 18.364808     197.78040
## Population  -2.327269     106.30375
## Price       52.837313     655.25299
## ShelveLoc   59.920760     746.44398
## Age         22.089119     269.18124
## Education    3.107283      74.50061
## Urban       -0.565052      14.42528
## US           4.024871      16.83711
```

Using random forest, the best MSE is 3.1204691, which is obtained when *mtry* is 6. The most important variables are 'ShelveLoc', 'Price', 'CompPrice', 'Age', and 'Advertising'. Furthermore, the test MSE for $m$ values from 4 to 10 are all about the same. There is a sharp decrease in test MSE when going from $m = 1$ to $m = 2$.