Milestone 4 Regression Tree

Group 13

2023-11-10

Regression Question: How indicative are the prices of everyday commodities of a city's average income?

```
library(pacman)
p_load(MASS, tree, randomForest, gbm, dplyr, tidyverse)

train <- readRDS("C:/Users/abz20/OneDrive/Desktop/UVA Courses/Statistical Machine Learning (STAT 4630)/S
test <- readRDS("C:/Users/abz20/OneDrive/Desktop/UVA Courses/Statistical Machine Learning (STAT 4630)/S
```

Parts A and B: Data Cleaning/Processing, Subsetting to Plausible Predictors

```
# exclude predictors related to each other or unrelated to question
train = train %>% dplyr::select(-city,
                                -country,
                                -beer.rest.domestic,
                                -beer.rest.imported,
                                -coffee,
                                -soda,
                                -water.rest,
                                -taxi.km,
                                -taxi.hr,
                                -rent1.center,
                                -rent1.outer,
                                -rent3.center,
                                -rent3.outer,
                                -sqm.center,
                                -sqm.outer,
                                -quality)

test = test %>% dplyr::select(-city,
                                -country,
                                -beer.rest.domestic,
                                -beer.rest.imported,
                                -coffee,
                                -soda,
                                -water.rest,
                                -taxi.km,
                                -taxi.hr,
                                -rent1.center,
                                -rent1.outer,
                                -rent3.center,
                                -rent3.outer,
```

```r
                                   -sqm.center,
                                   -sqm.outer,
                                   -quality)
```

Excluded variables: - city and country because they're identifiers, neither predictor nor response - beer.rest.domestic, beer.rest.imported, coffee, soda, water.rest, because they're related to meal1, meal2, and mcmeal - taxi.km and taxi.hr because they're related to taxi.start - rent1.center because this was converted to a categorical variable, expensive, so it would be redundant to include this - rent1.outer, rent3.center, rent3.outer, sqm.center, and sqm.outer because they're closely related to "expensive" - quality because this isn't related to cost of living

Part C: Recursive Binary Splitting

```r
# fit tree model using training data with binary recursive splitting
tree.result = tree::tree(salary ~ ., data = train) # x54 is monthly salary
```

i.

```r
# see output
summary(tree.result)
```

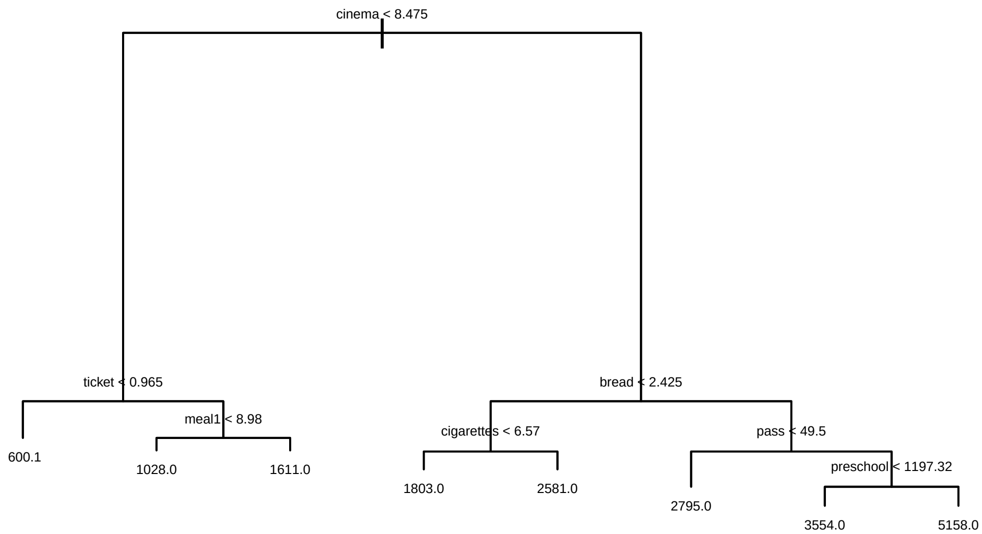
```
##
## Regression tree:
## tree::tree(formula = salary ~ ., data = train)
## Variables actually used in tree construction:
## [1] "cinema"     "ticket"     "meal1"      "bread"      "cigarettes"
## [6] "pass"       "preschool"
## Number of terminal nodes:  8
## Residual mean deviance:  963400 = 3.334e+09 / 3461
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -4158.0  -427.4  -166.9     0.0   306.1  6412.0
```

ii. 8 terminal nodes

iii. Variables actually used in tree construction: "cinema" "ticket" "meal1" "bread" "cigarettes" "pass" "preschool"

iv.

```r
# decision tree built on training data with recursive binary splitting
plot(tree.result)
text(tree.result, cex=0.4)
```



v. This answers our question of interest by selecting which predictors out of all our plausible predictors are most important in predicting the average monthly salary in a city.

vi.

```
# find predictions for test data
tree.pred.test = predict(tree.result, newdata=test)

# find test MSE
recursive_binary_test_mse = mean((test$salary - tree.pred.test)^2)

cat("Recursive Binary Test MSE:", recursive_binary_test_mse)
```

```
## Recursive Binary Test MSE: 844078.3
```

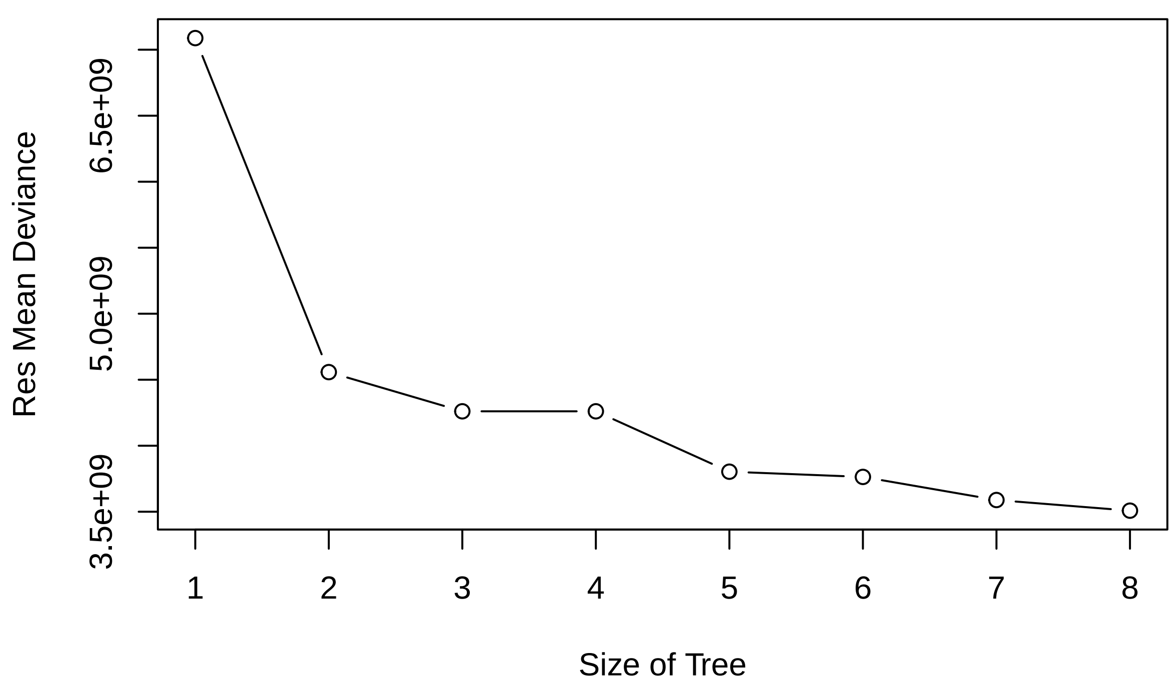
Part D: Pruned Tree

Note: The pruned tree is the same as the tree from recursive binary splitting. I still answered the questions from Part C just in case we decide to redo this.

```
# use 10-fold CV to prune tree
cv.Dataset = tree::cv.tree(tree.result, K=10)
cv.Dataset
```

```
## $size
## [1] 8 7 6 5 4 3 2 1
##
## $dev
## [1] 3508114240 3588591761 3763517153 3803168613 4260552508 4260552508 4557538162
## [8] 7087947986
##
## $k
## [1]       -Inf   86395274  123437701  131622376  244469163  254533571  349046227
## [8] 2555984694
##
## $method
## [1] "deviance"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

```r
# plot of residual mean deviance vs size of tree with pruning
plot(cv.Dataset$size, cv.Dataset$dev, type="b", xlab="Size of Tree", ylab="Res Mean Deviance")
```

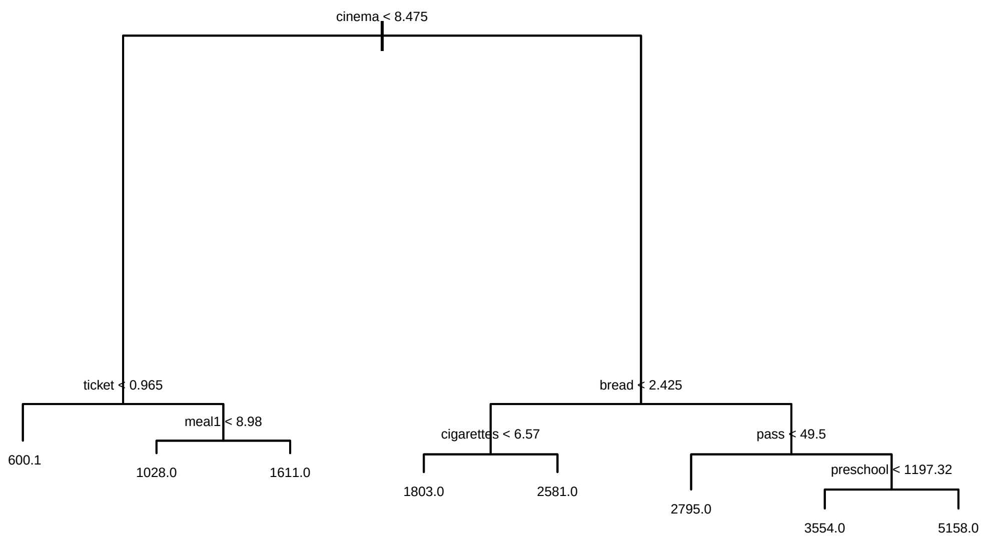



```r
# see size of tree which gives best tree based on pruning and 10-fold CV
trees.num = cv.Dataset$size[which.min(cv.Dataset$dev)]
trees.num
```

```
## [1] 8
```

```r
# refit with training data
tree.train = tree::tree(salary ~ ., data = train)
prune.train = tree::prune.tree(tree.train, best=trees.num)

# decision tree with pruning, with training data
plot(prune.train)
text(prune.train, cex=0.4)
```




```r
# numerical summary of pruned tree
prune.train
```

```
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 3469 7.080e+09 1709.0
##    2) cinema < 8.475 1984 1.129e+09  966.3
##      4) ticket < 0.965 970 1.877e+08  600.1 *
##      5) ticket > 0.965 1014 6.869e+08 1317.0
##       10) meal1 < 8.98 512 1.265e+08 1028.0 *
##       11) meal1 > 8.98 502 4.740e+08 1611.0 *
##    3) cinema > 8.475 1485 3.395e+09 2701.0
##      6) bread < 2.425 907 1.213e+09 2314.0
##       12) cigarettes < 6.57 311 2.701e+08 1803.0 *
##       13) cigarettes > 6.57 596 8.196e+08 2581.0 *
##      7) bread > 2.425 578 1.832e+09 3308.0
```

```
##       14) pass < 49.5 356 8.675e+08 2795.0 *
##       15) pass > 49.5 222 7.205e+08 4132.0
##         30) preschool < 1197.32 142 3.696e+08 3554.0 *
##         31) preschool > 1197.32 80 2.192e+08 5158.0 *
```

i.

```r
# see output
summary(prune.train)
```

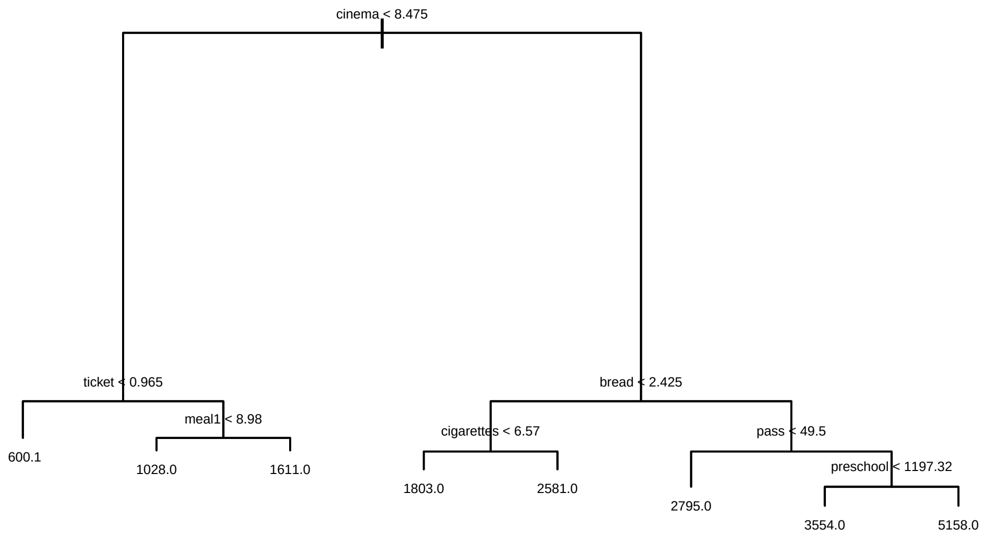
```
##
## Regression tree:
## tree::tree(formula = salary ~ ., data = train)
## Variables actually used in tree construction:
## [1] "cinema"     "ticket"     "meal1"      "bread"      "cigarettes"
## [6] "pass"       "preschool"
## Number of terminal nodes:  8
## Residual mean deviance:  963400 = 3.334e+09 / 3461
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -4158.0  -427.4  -166.9     0.0   306.1  6412.0
```

ii. 8 terminal nodes

iii. Variables actually used in tree construction: “cinema” “ticket” “meal1” “bread” “cigarettes” “pass” “preschool”

iv.

```r
# decision tree built on training data with recursive binary splitting
plot(prune.train)
text(prune.train, cex=0.4)
```



v. This answers our question of interest by selecting which predictors out of all our plausible predictors are most important in predicting the average monthly salary in a city, but with less overfitting.

vi.

```
# find predictions for test data
tree.pred.test = predict(prune.train, newdata=test)

# find test MSE
pruned_tree_test_mse = mean((test$salary - tree.pred.test)^2)

cat("Pruned Tree Test MSE:", pruned_tree_test_mse)
```

```
## Pruned Tree Test MSE: 844078.3
```

Part E: Random Forests

```
rf.class = randomForest::randomForest(salary ~ ., data=train, mtry=2,importance=TRUE) # mtry = p/3 for
rf.class
```

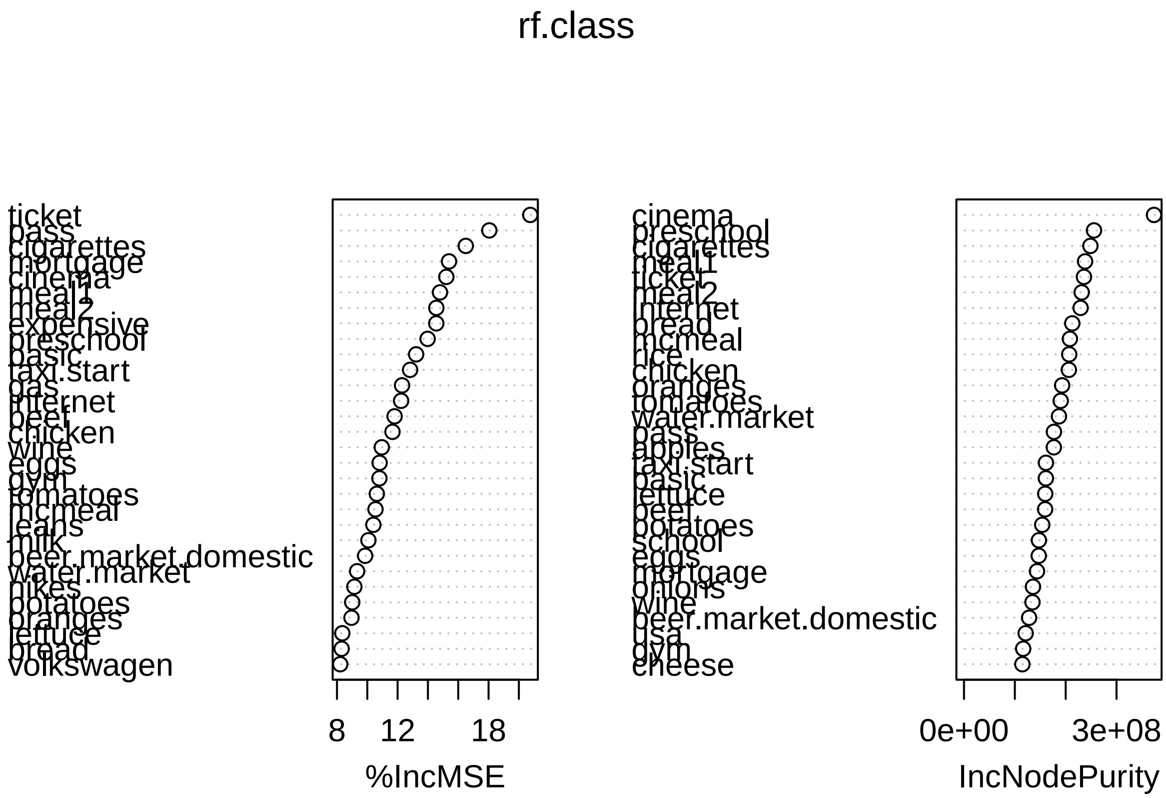
```
##
## Call:
##  randomForest(formula = salary ~ ., data = train, mtry = 2, importance = TRUE)
##                Type of random forest: regression
```

```
##                      Number of trees: 500
## No. of variables tried at each split: 2
##
##           Mean of squared residuals: 860748.3
##                     % Var explained: 57.82
```

```r
importance(rf.class)
```

```
##                        %IncMSE IncNodePurity
## meal1                14.796393     238157657
## meal2                14.555734     231513905
## mcmeal               10.542906     208272745
## milk                 10.073646      76271505
## bread                 8.307182     212960682
## rice                  7.304565     206929331
## eggs                 10.813562     147083314
## cheese                6.790259     114821109
## chicken              11.662435     206232536
## beef                 11.800918     159569640
## apples                7.603433     176696830
## bananas               7.800040      89183997
## oranges               8.958266     193016431
## tomatoes             10.629167     190210465
## potatoes              9.006724     153877002
## onions                4.359201     135731988
## lettuce               8.351248     159743999
## water.market          9.327800     186825412
## wine                 10.955881     134702886
## beer.market.domestic  9.855623     128107771
## beer.market.imported  7.591608      82697518
## cigarettes           16.498616     248628321
## ticket               20.752658     235921720
## pass                 18.050147     176864904
## taxi.start           12.824204     161153177
## gas                  12.293727      90890057
## volkswagen            8.215332      67562264
## toyota                7.168431      68684274
## basic                13.222801     161034169
## mobile                5.762668      68554905
## internet             12.232467     229309545
## gym                  10.798512     116429371
## tennis                7.342126      67981890
## cinema               15.214146     373675312
## preschool            13.981132     255669862
## school                6.635243     147370148
## jeans                10.400378      94035021
## dresses               5.354704      72567600
## nikes                 9.145872      76575927
## shoes                 6.946215      98066776
## mortgage             15.391919     143578180
## expensive            14.552249     110945941
## usa                   7.668275     121307359
```

```r
varImpPlot(rf.class)
```



i. ticket and cinema are most important predictors

```r
# test accuracy with Random Forest
pred.rf<-predict(rf.class, newdata=test)

RF_test_mse = mean((test$salary - pred.rf)^2)

cat("Random Forest Test MSE:", RF_test_mse)
```

```
## Random Forest Test MSE: 673278.9
```

Part F: Conclusion

i.

```r
cat("Recursive Binary Test MSE:", recursive_binary_test_mse, "\n",
    "Pruned Test MSE:", pruned_tree_test_mse, "\n",
    "Random Forest Test MSE:", RF_test_mse)
```

```
## Recursive Binary Test MSE: 844078.3
##  Pruned Test MSE: 844078.3
##  Random Forest Test MSE: 673278.9
```