DT_Regression

Austin Girouard

2023-03-21

Decision Tree Regression

Load the Data

```
df <- read.csv ("job_profitability.csv", header = TRUE) # specifies where to load data from
df <- df[,c(3, 4, 5, 10)] # specifies which columns we want
str(df) # print data frame structure
```

```
## 'data.frame':    14479 obs. of  4 variables:
##  $ Jobs_Gross_Margin: num  -4.01 254.13 151.83 -32.15 222.7 ...
##  $ Labor_Pay        : num  0 91 0 0 0 ...
##  $ Labor_Burden     : num  22.2 14.9 133.2 81.2 66.3 ...
##  $ Jobs_Total       : num  79.5 360 289 49 289 ...
```

Sample Training and Testing Data

```
set.seed(1234)
i <- sample(1:nrow(df), nrow(df) * 0.8, replace = FALSE) # split data into 80/20 train/test
train <- df[i, ] # training data
test <- df[-i, ] # testing data
```

Using tree

```
library(tree)
```

```
## Warning: package 'tree' was built under R version 4.2.3
```

```
tree1 <- tree(Jobs_Gross_Margin~., data=train)
summary(tree1)
```

```
##
## Regression tree:
## tree(formula = Jobs_Gross_Margin ~ ., data = train)
## Number of terminal nodes:  9
## Residual mean deviance:  98960 = 1.145e+09 / 11570
## Distribution of residuals:
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -7260.000   -87.730    -8.877     0.000   111.000  8209.000
```

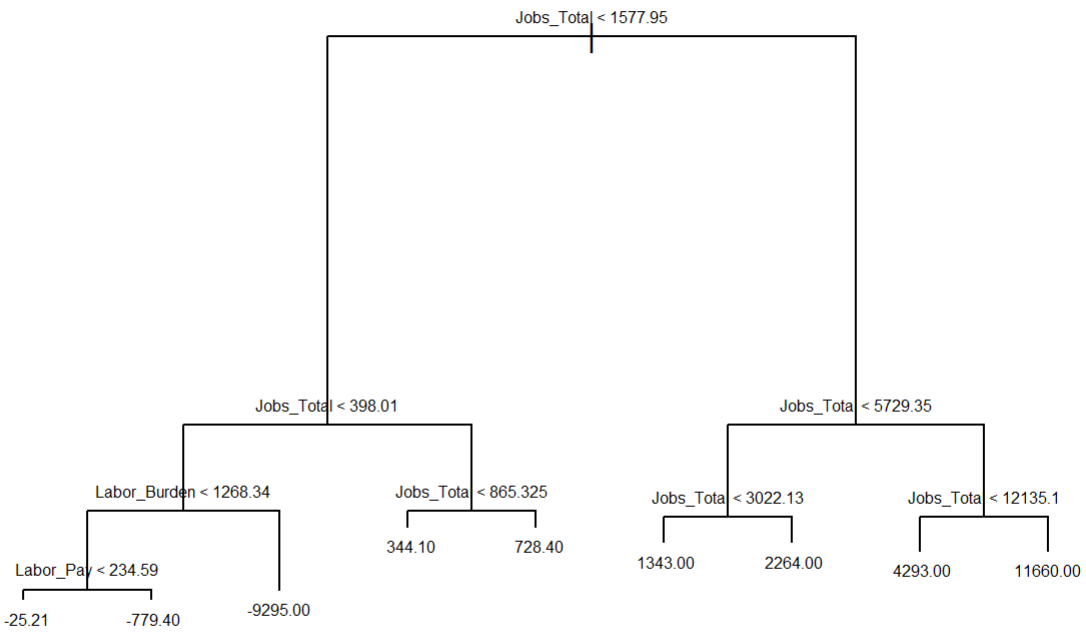
```
pred <- predict(tree1, newdata=test)
print(paste('correlation:', cor(pred, test$Jobs_Gross_Margin)))
```

```
## [1] "correlation: 0.914753236805936"
```

```
rmse_tree <- sqrt(mean((pred-test$Jobs_Gross_Margin)^2))
print(paste('rmse:', rmse_tree))
```

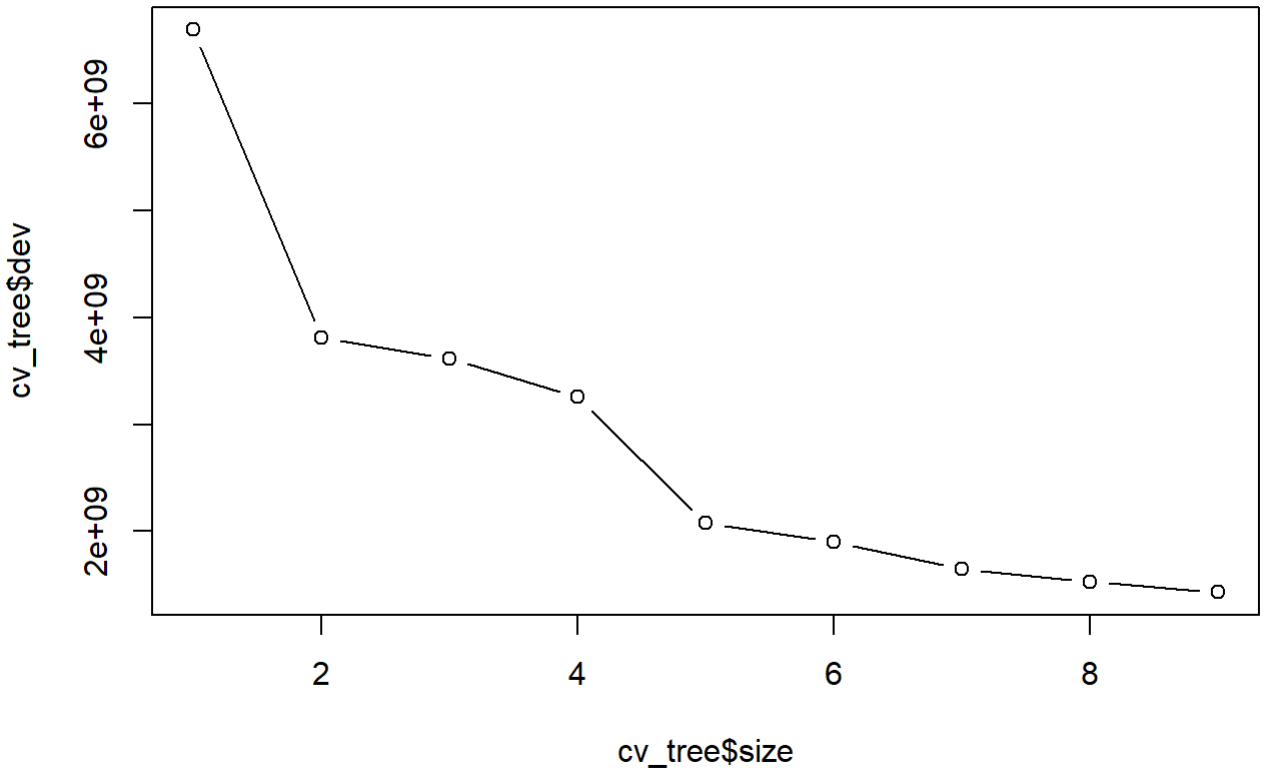
```
## [1] "rmse: 270.155476202041"
```

```
plot(tree1)
text(tree1, cex=0.5, pretty=0)
```



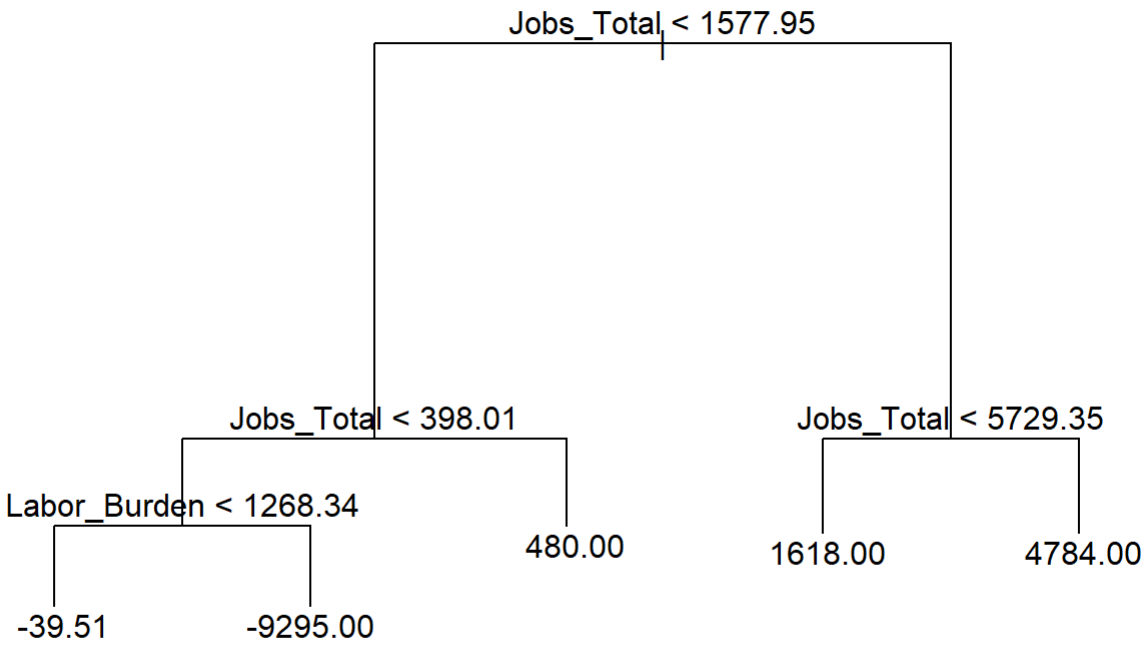
cross validation

```
cv_tree <- cv.tree(tree1)
plot(cv_tree$size, cv_tree$dev, type='b')
```



prune the tree

```
tree_pruned <- prune.tree(tree1, best=5)
plot(tree_pruned)
text(tree_pruned, pretty=0)
```



Find correlation of pruned tree

```
pred_pruned <- predict(tree_pruned, newdata=test)
cor_pruned <- cor(pred_pruned, test$Jobs_Gross_Margin)
mse_pruned <- mean((pred_pruned-test$Jobs_Gross_Margin)^2)
print(paste('cor: ', cor_pruned))
```

```
## [1] "cor:  0.859246727417666"
```

```
print(paste('mse: ', mse_pruned))
```

```
## [1] "mse:  116650.137250333"
```

```
print(paste('rmse: ', sqrt(mse_pruned)))
```

```
## [1] "rmse:  341.54082808697"
```

Random Forest

The importance=TRUE argument tells the algorithm to consider the importance of predictors.

```
#install.packages('randomForest')
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.2.3
```

```
## randomForest 4.7-1.1
```

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

```
set.seed(1234)
rf <- randomForest(Jobs_Gross_Margin~., data=train, importance=TRUE)
rf
```

```
##
## Call:
##  randomForest(formula = Jobs_Gross_Margin ~ ., data = train, importance = TRUE)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 1
##
##           Mean of squared residuals: 86323.38
##                     % Var explained: 85.05
```

predict on the random forest

Now the correlation is much higher than even linear regression and the rmse is almost half.

```
pred_rf <- predict(rf, newdata=test)
cor_rf <- cor(pred_rf, test$Jobs_Gross_Margin)
print(paste('corr:', cor_rf))
```

```
## [1] "corr: 0.956120183805892"
```

```
rmse_rf <- sqrt(mean((pred_rf-test$Jobs_Gross_Margin)^2))
print(paste('rmse:', rmse_rf))
```

```
## [1] "rmse: 196.839738174002"
```

bagging

Setting mtry to the number of predictors, p, will result in bagging

```
bag <- randomForest(Jobs_Gross_Margin~., data=train, mtry=3)
bag
```

```
##
## Call:
##  randomForest(formula = Jobs_Gross_Margin ~ ., data = train, mtry = 3)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##           Mean of squared residuals: 79532.88
##                     % Var explained: 86.23
```

predict

Our results for bagging were slightly lower than for the random forest.

```
pred_bag <- predict(bag, newdata=test)
cor_bag <- cor(pred_bag, test$Jobs_Gross_Margin)
mse_bag <- mean((pred_bag-test$Jobs_Gross_Margin)^2)

print(paste('cor: ', cor_bag))
```

```
## [1] "cor:  0.957201775647217"
```

```
print(paste('mse: ', mse_bag))
```

```
## [1] "mse:  37384.8144143277"
```

```
print(paste('rmse: ', sqrt(mse_bag)))
```

```
## [1] "rmse:  193.351530674902"
```