Practicum CCheck

2025-06-22

```r
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 4.1.3
```

```r
library(data.table)
```

```
## Warning: package 'data.table' was built under R version 4.1.3
```

```r
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.1.3
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

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

```r
library(ggplot2)
library(rpart)
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 4.1.3
```

```r
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.1.3
```

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

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

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(nnet)
```

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

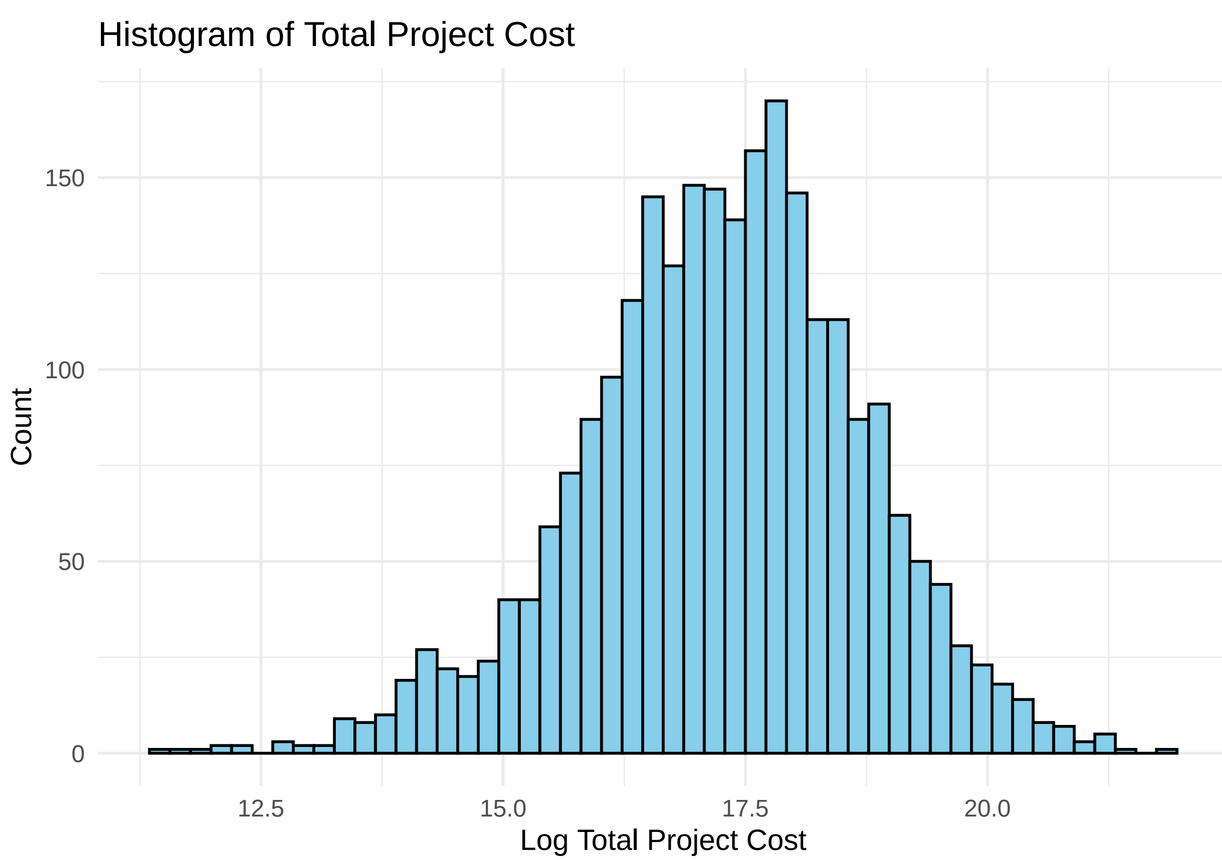
```
ccheckdata <- read.csv('final_data.csv')
```

Remove variables that do not have predictive power and remove NaNs from the area cost factors variable and total project cost variable

```
new_cc_data <- subset(ccheckdata, select = -c(project_id, project_city))
new_cc_data <- new_cc_data[!is.na(new_cc_data$total_project_cost),]
new_cc_data <- new_cc_data[!is.na(new_cc_data$acf_2023),]
```

Remove outliers, normalize response variable, remove original response variable

```
new_cc_data <- new_cc_data[(new_cc_data$total_project_cost != max(new_cc_data$total_project_cost)),]
new_cc_data$log_total_project_cost <- log(new_cc_data$total_project_cost)
ggplot(new_cc_data, aes(x = log_total_project_cost)) +
  geom_histogram(bins = 50, fill = "skyblue", color = "black") +
  labs(title = "Histogram of Total Project Cost",
       x = "Log Total Project Cost",
       y = "Count") +
  theme_minimal()
```



```r
new_cc_data$total_project_cost <- NULL
```

Summarize data and find features that have too few levels. In this case, we're analyzing the project type variable.

```r
table(new_cc_data$type)
```

```
##
##                 Aircraft De-Icing Station
##                                         4
##                  Airport Cargo Facility
##                                        21
##             Airport Electronic Maintenance
##                                        19
##                  Airport Runways/Taxiway
##                                        63
##                  Airport Security Control
##                                         4
##                        Airport Terminals
##                                        19
##                       Bridges & Culverts
##                                         1
##                              Cafeterias
##                                       628
##                               Cathedral
##                                       130
##                                 Churches
##                                         1
##                  Colleges & Universities
##                                         3
##        Commemorative and Funeral Monument
##                                       267
##                   Communication Devices
##                                       190
##                             Courthouses
##                                         2
##                  Critical Care Facility
##                                       271
##                        Custom Residence
##                                       198
##                           Fire Stations
##                                         3
##                                   Hotel
##                                        28
##                                   Motel
##                                         7
## Municipal Water and Wastewater Facilities
##                                        22
##                                 Offices
##                                         1
##                          Oil Refineries
##                                        84
##            Parking Garages (free-standing)
```

```
##                                        6
##                                  Prisons
##                                      364
##                            Rail Stations
##                                       22
##                                    Roads
##                                        1
##                                Site Work
##                                        1
##              Sports and Fitness Facility
##                                      147
##                          Tunnel & Bridge 
##                                        3
##                  Water and Sewage Piping
##                                        5
```

```
level_counts <- table(new_cc_data$type)

# Identify levels with 5 or fewer observations
rare_levels <- names(level_counts[level_counts <= 5])

# Count how many rows belong to these rare levels
num_rare_obs <- sum(new_cc_data$type %in% rare_levels)

cat("Number of observations with rare levels (<5):", num_rare_obs)
```

```
## Number of observations with rare levels (<5): 29
```

```
filtered_data <- new_cc_data[!(new_cc_data$type %in% rare_levels), ]
```

Summarize data and find features that have too few levels. In this case, we're analyzing the project state variable.

```
table(filtered_data$project_state)
```

```
## 
##  AL  AR  AZ  CA  CO  CT  DC  DE  FL  GA  HI  ID  IL  IN  KS  LA  MA  MD  ME  MI 
##  23  45  93 313  51  45   6  33 155 241   3   3  81  30  28  15  50  37   1 109 
##  MN  MO  MS  MT  NC  ND  NE  NH  NJ  NM  NV  NY  OH  OK  OR  PA  RI  SC  SD  TN 
##  68  47  34  18  44  22   9   3  54  20  19  51  91  24  11 160   7  34   3  56 
##  TX  UT  VA  VT  WA  WI  WV  WY 
## 213  21  36   3  26  22  21   7
```

```
level_counts <- table(filtered_data$project_state)

# Identify levels with 5 or fewer observations
rare_levels <- names(level_counts[level_counts <= 3])

# Count how many rows belong to these rare levels
num_rare_obs <- sum(filtered_data$project_state %in% rare_levels)

cat("Number of observations with rare levels (<3):", num_rare_obs)
```

```
## Number of observations with rare levels (<3): 16
```

```r
filtered_data <- filtered_data[!(filtered_data$project_state %in% rare_levels), ]
```

remove predictor that does not vary

```r
filtered_data$cost_div_46 <- NULL
```

Split data into training and testing

```r
set.seed(1)
train_indices <- createDataPartition(filtered_data$log_total_project_cost, p = 0.8, list = FALSE)
train_data <- filtered_data[train_indices, ]
test_data <- filtered_data[-train_indices, ]
```

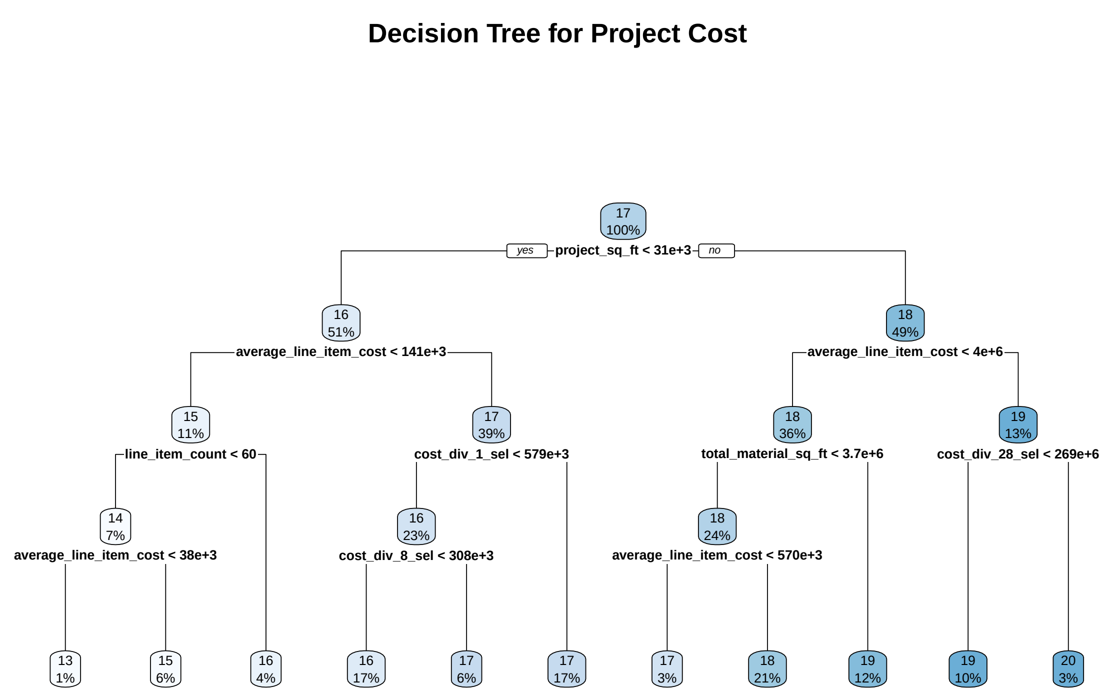
Train decision tree

Note: This model is not used in our final report.

```r
tree_model <- rpart(log_total_project_cost ~ ., data = train_data,
                    method = "anova")
```

Plot the tree

```r
rpart.plot(tree_model, main = "Decision Tree for Project Cost")
```



```r
summary(tree_model)
```

```
## Call:
## rpart(formula = log_total_project_cost ~ ., data = train_data,
##     method = "anova")
##   n= 1978
##
##            CP nsplit rel error    xerror        xstd
## 1  0.49013354      0 1.0000000 1.0002764 0.034998213
## 2  0.11266744      1 0.5098665 0.5198280 0.021431869
## 3  0.08195714      2 0.3971990 0.4133945 0.016508185
## 4  0.03302081      3 0.3152419 0.3446442 0.014610556
## 5  0.03289163      4 0.2822211 0.2923038 0.012725958
## 6  0.02604420      5 0.2493294 0.2758380 0.012183683
## 7  0.01663688      6 0.2232852 0.2540206 0.011718938
## 8  0.01588222      7 0.2066483 0.2366115 0.011178947
## 9  0.01363977      8 0.1907661 0.2144930 0.009764130
## 10 0.01094812      9 0.1771264 0.2041877 0.009281731
## 11 0.01000000     10 0.1661782 0.1854354 0.008658368
##
## Variable importance
##          project_sq_ft average_line_item_cost   total_material_sq_ft
##                     21                     20                     16
##        cost_div_28_sel          cost_div_8_sel          cost_div_1_sel
##                     12                     11                     11
##        line_item_count          cost_div_0_sel         cost_div_12_sel
##                      2                      1                      1
##            cost_div_11          cost_div_9_sel                   type
##                      1                      1                      1
##         cost_div_7_sel
##                      1
##
## Node number 1: 1978 observations,    complexity param=0.4901335
##   mean=17.2775, MSE=2.075171
##   left son=2 (1006 obs) right son=3 (972 obs)
##   Primary splits:
##       project_sq_ft          < 31460     to the left,  improve=0.4901335, (0 missing)
##       total_material_sq_ft   < 1016282   to the left,  improve=0.4645487, (0 missing)
##       average_line_item_cost < 684149    to the left,  improve=0.4100339, (0 missing)
##       cost_div_8_sel         < 743986    to the left,  improve=0.3620665, (0 missing)
##       cost_div_1_sel         < 1003440   to the left,  improve=0.3564567, (0 missing)
##   Surrogate splits:
##       total_material_sq_ft   < 886174.9  to the left,  agree=0.850, adj=0.694, (0 split)
##       average_line_item_cost < 838112.4  to the left,  agree=0.814, adj=0.621, (0 split)
##       cost_div_8_sel         < 589464    to the left,  agree=0.764, adj=0.521, (0 split)
##       cost_div_28_sel        < 28173510  to the left,  agree=0.760, adj=0.512, (0 split)
##       cost_div_1_sel         < 1033310   to the left,  agree=0.760, adj=0.511, (0 split)
##
## Node number 2: 1006 observations,    complexity param=0.1126674
##   mean=16.28617, MSE=1.196617
##   left son=4 (227 obs) right son=5 (779 obs)
##   Primary splits:
##       average_line_item_cost < 141189.4  to the left,  improve=0.3841718, (0 missing)
```

```
##       total_material_sq_ft   < 146565.6  to the left,  improve=0.3287660, (0 missing)
##       project_sq_ft          < 8033.5    to the left,  improve=0.2804845, (0 missing)
##       cost_div_1_sel         < 285916.5  to the left,  improve=0.2626202, (0 missing)
##       cost_div_0_sel         < 133139.5  to the left,  improve=0.2103094, (0 missing)
##   Surrogate splits:
##       project_sq_ft        < 4540      to the left,  agree=0.812, adj=0.167, (0 split)
##       total_material_sq_ft < 66979.7   to the left,  agree=0.803, adj=0.128, (0 split)
##       type                 splits as  LRRLRRRRRLRLRRLRRR, agree=0.796, adj=0.097, (0 split)
##       project_category     splits as  RRRRRRRRLRRLR, agree=0.786, adj=0.053, (0 split)
##       project_state        splits as  RRRRRRRRRRRRRRRRRRLRRRLRRRRRRRRRRRRRRLRRL, agree=0.782, adj=0
##
## Node number 3: 972 observations,    complexity param=0.08195714
##   mean=18.3035, MSE=0.9146568
##   left son=6 (710 obs) right son=7 (262 obs)
##   Primary splits:
##       average_line_item_cost < 3950421   to the left,  improve=0.3783926, (0 missing)
##       cost_div_28_sel        < 106294500 to the left,  improve=0.3782048, (0 missing)
##       cost_div_8_sel         < 2273372   to the left,  improve=0.3626541, (0 missing)
##       cost_div_1_sel         < 3357233   to the left,  improve=0.3524840, (0 missing)
##       cost_div_7_sel         < 2840680   to the left,  improve=0.3103409, (0 missing)
##   Surrogate splits:
##       cost_div_28_sel < 76955650  to the left,  agree=0.866, adj=0.504, (0 split)
##       project_sq_ft   < 183139.5  to the left,  agree=0.820, adj=0.332, (0 split)
##       cost_div_0_sel  < 10032240  to the left,  agree=0.787, adj=0.210, (0 split)
##       cost_div_8_sel  < 2978240   to the left,  agree=0.787, adj=0.210, (0 split)
##       cost_div_9_sel  < 6153552   to the left,  agree=0.784, adj=0.198, (0 split)
##
## Node number 4: 227 observations,    complexity param=0.03289163
##   mean=15.03015, MSE=1.316329
##   left son=8 (146 obs) right son=9 (81 obs)
##   Primary splits:
##       line_item_count        < 59.5      to the left,  improve=0.4518302, (0 missing)
##       total_material_sq_ft   < 146434    to the left,  improve=0.3580747, (0 missing)
##       cost_div_12_sel        < 207459    to the left,  improve=0.3466589, (0 missing)
##       cost_div_11            < 150688.5  to the left,  improve=0.3449982, (0 missing)
##       average_line_item_cost < 78082.79  to the left,  improve=0.3346142, (0 missing)
##   Surrogate splits:
##       total_material_sq_ft < 203866.2  to the left,  agree=0.833, adj=0.531, (0 split)
##       cost_div_12_sel      < 196068    to the left,  agree=0.797, adj=0.432, (0 split)
##       cost_div_11          < 96114     to the left,  agree=0.784, adj=0.395, (0 split)
##       divisions_with_cost  < 12.5      to the left,  agree=0.784, adj=0.395, (0 split)
##       cost_div_0_sel       < 130619    to the left,  agree=0.767, adj=0.346, (0 split)
##
## Node number 5: 779 observations,    complexity param=0.0260442
##   mean=16.65217, MSE=0.5680685
##   left son=10 (446 obs) right son=11 (333 obs)
##   Primary splits:
##       cost_div_1_sel         < 579205.5  to the left,  improve=0.2415757, (0 missing)
##       average_line_item_cost < 559202.3  to the left,  improve=0.2312547, (0 missing)
##       total_material_sq_ft   < 287929.5  to the left,  improve=0.2240399, (0 missing)
##       project_sq_ft          < 18431.5   to the left,  improve=0.2119707, (0 missing)
##       cost_div_3_sel         < 598605    to the left,  improve=0.2089183, (0 missing)
##   Surrogate splits:
##       project_state        splits as  LLLLLLRRRRLLLLLLLLLLLLL-LLRLLRLLRLLLLLRLL-, agree=0.666, adj=0
```

```
##       line_item_count      < 54.5      to the left,  agree=0.660, adj=0.204, (0 split)
##       cost_div_0_sel       < 985507.5  to the left,  agree=0.657, adj=0.198, (0 split)
##       project_sq_ft        < 18386     to the left,  agree=0.655, adj=0.192, (0 split)
##       total_material_sq_ft < 870644.5  to the left,  agree=0.651, adj=0.183, (0 split)
##
## Node number 6: 710 observations,    complexity param=0.03302081
##   mean=17.94613, MSE=0.5832989
##   left son=12 (469 obs) right son=13 (241 obs)
##   Primary splits:
##       total_material_sq_ft < 3725186   to the left,  improve=0.3272792, (0 missing)
##       cost_div_7_sel       < 1548636   to the left,  improve=0.2859645, (0 missing)
##       cost_div_12_sel      < 3797784   to the left,  improve=0.2759009, (0 missing)
##       cost_div_9_sel       < 3064028   to the left,  improve=0.2755372, (0 missing)
##       cost_div_8_sel       < 1283564   to the left,  improve=0.2612079, (0 missing)
##   Surrogate splits:
##       line_item_count < 82        to the left,  agree=0.883, adj=0.656, (0 split)
##       cost_div_12_sel < 3501058   to the left,  agree=0.776, adj=0.340, (0 split)
##       cost_div_7_sel  < 1855237   to the left,  agree=0.766, adj=0.311, (0 split)
##       cost_div_10     < 291399    to the left,  agree=0.754, adj=0.274, (0 split)
##       cost_div_11     < 1002556   to the left,  agree=0.754, adj=0.274, (0 split)
##
## Node number 7: 262 observations,    complexity param=0.01663688
##   mean=19.27196, MSE=0.528609
##   left son=14 (206 obs) right son=15 (56 obs)
##   Primary splits:
##       cost_div_28_sel        < 269119000 to the left,  improve=0.4930789, (0 missing)
##       average_line_item_cost < 15012100  to the left,  improve=0.4858649, (0 missing)
##       cost_div_1_sel         < 10856530  to the left,  improve=0.3937031, (0 missing)
##       total_material_sq_ft   < 2949820   to the left,  improve=0.3774576, (0 missing)
##       cost_div_8_sel         < 4825974   to the left,  improve=0.3633515, (0 missing)
##   Surrogate splits:
##       average_line_item_cost < 16265070  to the left,  agree=0.916, adj=0.607, (0 split)
##       cost_div_8_sel         < 7877166   to the left,  agree=0.863, adj=0.357, (0 split)
##       cost_div_0_sel         < 18053540  to the left,  agree=0.847, adj=0.286, (0 split)
##       cost_div_1_sel         < 10960230  to the left,  agree=0.844, adj=0.268, (0 split)
##       cost_div_42            < 3758001   to the left,  agree=0.840, adj=0.250, (0 split)
##
## Node number 8: 146 observations,    complexity param=0.01094812
##   mean=14.45572, MSE=0.7309211
##   left son=16 (22 obs) right son=17 (124 obs)
##   Primary splits:
##       average_line_item_cost < 37843.45  to the left,  improve=0.4211108, (0 missing)
##       cost_div_0_sel         < 1442      to the left,  improve=0.3140793, (0 missing)
##       cost_div_1_sel         < 42257     to the left,  improve=0.2915135, (0 missing)
##       total_material_sq_ft   < 61289.68  to the left,  improve=0.2901081, (0 missing)
##       line_item_count        < 25.5      to the left,  improve=0.2862524, (0 missing)
##   Surrogate splits:
##       project_state splits as  --RRLR--RRRR-RRRRRR-RRRLR--RR--R-RRR-RRR-R, agree=0.877, adj=0.182, (
##       type          splits as  RRRLRRRRRR-RRR-RRR, agree=0.856, adj=0.045, (0 split)
##
## Node number 9: 81 observations
##   mean=16.06554, MSE=0.7047206
##
## Node number 10: 446 observations,    complexity param=0.01363977
```

```
##   mean=16.33207, MSE=0.4077281
##   left son=20 (333 obs) right son=21 (113 obs)
##   Primary splits:
##       cost_div_8_sel         < 308479    to the left,  improve=0.3078801, (0 missing)
##       total_material_sq_ft   < 202725.3  to the left,  improve=0.3061491, (0 missing)
##       average_line_item_cost < 525237.8  to the left,  improve=0.2843547, (0 missing)
##       cost_div_3_sel         < 329292    to the left,  improve=0.2731279, (0 missing)
##       cost_div_28_sel        < 14293480  to the left,  improve=0.2599348, (0 missing)
##   Surrogate splits:
##       cost_div_5_sel < 461209    to the left,  agree=0.848, adj=0.398, (0 split)
##       cost_div_3_sel < 499947    to the left,  agree=0.832, adj=0.336, (0 split)
##       cost_div_9_sel < 761023.5  to the left,  agree=0.832, adj=0.336, (0 split)
##       cost_div_7_sel < 356558    to the left,  agree=0.821, adj=0.292, (0 split)
##       cost_div_4     < 563233.5  to the left,  agree=0.818, adj=0.283, (0 split)
##
## Node number 11: 333 observations
##   mean=17.08089, MSE=0.4617875
##
## Node number 12: 469 observations,    complexity param=0.01588222
##   mean=17.63293, MSE=0.3811734
##   left son=24 (54 obs) right son=25 (415 obs)
##   Primary splits:
##       average_line_item_cost < 569683.7  to the left,  improve=0.3646667, (0 missing)
##       type                   splits as  -RLLRRRRRRRLRR-R-R, improve=0.2181468, (0 missing)
##       cost_div_28_sel        < 40448940  to the left,  improve=0.2120978, (0 missing)
##       cost_div_9_sel         < 1851057   to the left,  improve=0.1838151, (0 missing)
##       cost_div_6_sel         < 198089.5  to the left,  improve=0.1804262, (0 missing)
##   Surrogate splits:
##       type             splits as  -RLLRRRRRRRLRR-R-R, agree=0.919, adj=0.296, (0 split)
##       cost_div_2       < 223024.2  to the right, agree=0.915, adj=0.259, (0 split)
##       project_category splits as  LRRRRRRRRRRLR, agree=0.908, adj=0.204, (0 split)
##       cost_div_13      < 2825.905  to the right, agree=0.902, adj=0.148, (0 split)
##       cost_div_32      < 38601.9   to the right, agree=0.902, adj=0.148, (0 split)
##
## Node number 13: 241 observations
##   mean=18.55564, MSE=0.4142397
##
## Node number 14: 206 observations
##   mean=19.00577, MSE=0.2737803
##
## Node number 15: 56 observations
##   mean=20.25114, MSE=0.246564
##
## Node number 16: 22 observations
##   mean=13.13858, MSE=0.6749336
##
## Node number 17: 124 observations
##   mean=14.68941, MSE=0.3784461
##
## Node number 20: 333 observations
##   mean=16.12568, MSE=0.2987411
##
## Node number 21: 113 observations
##   mean=16.94029, MSE=0.233442
```

```
##
## Node number 24: 54 observations
##   mean=16.59937, MSE=0.5452008
##
## Node number 25: 415 observations
##   mean=17.76741, MSE=0.2027419
```

Build the prediction

```
predictions <- predict(tree_model, newdata = test_data)
predictions <- exp(predictions)
```

Evaluate Decision Tree Model Performance

```
MAE <- mean(abs(predictions - exp(test_data$log_total_project_cost)))
RMSE <- sqrt(mean((predictions - exp(test_data$log_total_project_cost))^2))
cat("MAE:", MAE, "RMSE:", RMSE)
```

```
## MAE: 35000940 RMSE: 99264470
```

Develop Random Forest

```
rf_model <- randomForest(
  log_total_project_cost ~ .,
  data = train_data,
  ntree = 500,          # Number of trees
  mtry = 3,             # Number of variables randomly sampled at each split (can tune this)
  importance = TRUE     # Enables feature importance output
)
```

Print Random Forest results

```
print(rf_model)
```

```
##
## Call:
##  randomForest(formula = log_total_project_cost ~ ., data = train_data,      ntree = 500, mtry = 3, i
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##           Mean of squared residuals: 0.07339637
##                     % Var explained: 96.46
```

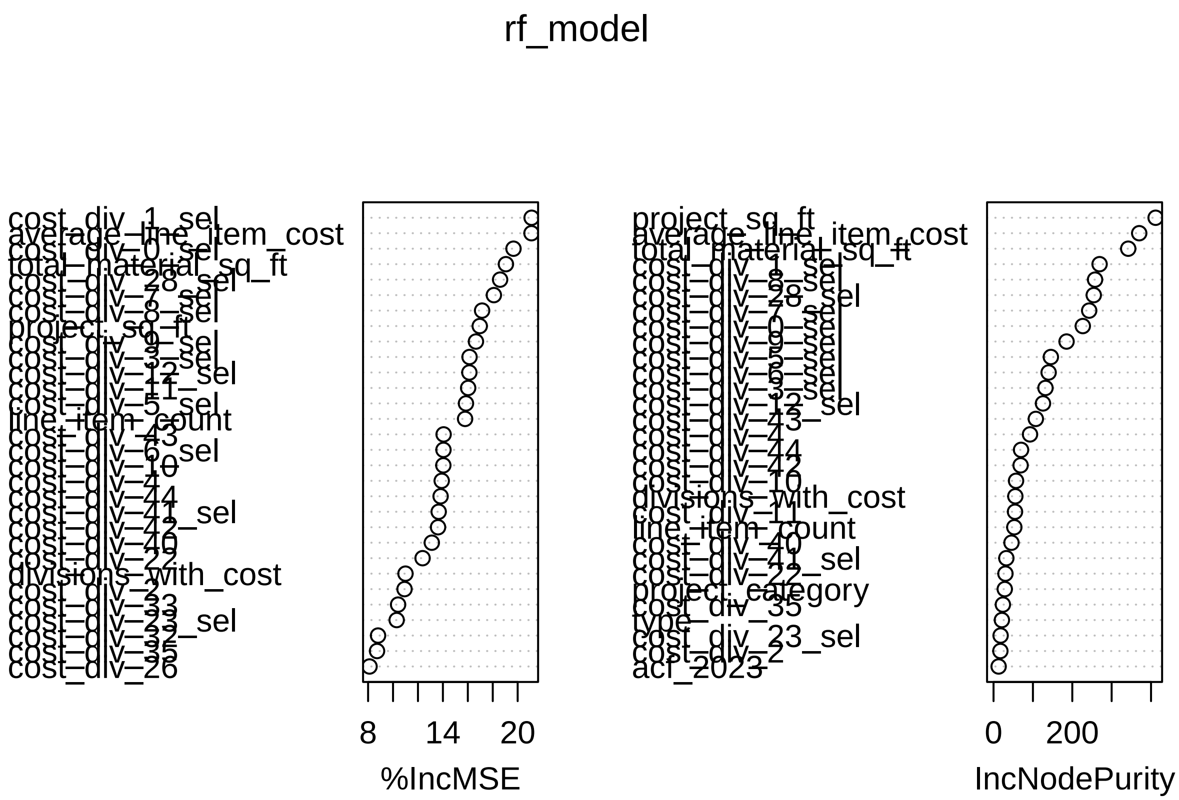
Find the most important variables

```
importance(rf_model)
```

```
##                                                       %IncMSE IncNodePurity
## project_state                                       5.3381119    10.6528069
## project_sq_ft                                      16.9565386   411.6545836
```

```
## type                                          6.0709192    21.1671054
## project_category                              5.7768041    28.5308278
## construction_category                         6.4707350     3.5377223
## acf_2023                                      6.4763579    12.9234849
## new_construction_median_sale_price_per_sqft_norm  5.5982390    12.8385690
## cost_div_0_sel                               19.6631602   226.7050287
## cost_div_1_sel                               21.1174549   269.3339574
## cost_div_2                                   10.9233865    17.3105146
## cost_div_3_sel                               16.1376616   131.9541797
## cost_div_4                                   13.9067846    92.6245118
## cost_div_5_sel                               15.8517741   145.5667149
## cost_div_6_sel                               14.0448463   139.9181850
## cost_div_7_sel                               18.0892505   242.9298566
## cost_div_8_sel                               17.1452175   257.8663326
## cost_div_9_sel                               16.6506882   185.4594215
## cost_div_10                                  14.0356474    57.1757977
## cost_div_11                                  16.0264534    54.5239134
## cost_div_12_sel                              16.1238624   125.5826515
## cost_div_13                                   4.5495583     6.6596379
## cost_div_14                                   4.6922154     2.3919320
## cost_div_21                                   6.5753491     8.2578680
## cost_div_22                                  12.3713175    30.2248836
## cost_div_23_sel                              10.2928492    17.9739847
## cost_div_25                                   0.3084526     0.2559805
## cost_div_26                                   8.1163762     9.9109943
## cost_div_27                                   5.8342438     4.1450005
## cost_div_28_sel                              18.5901340   254.5068910
## cost_div_31                                   7.5497419    10.0250462
## cost_div_32                                   8.7992408     9.9205299
## cost_div_33                                  10.4062068    11.3038690
## cost_div_34                                   6.7170158     5.4531796
## cost_div_35                                   8.7161717    23.7089450
## cost_div_40                                  13.1103496    45.5540151
## cost_div_41_sel                              13.6629038    32.4958938
## cost_div_42                                  13.6122440    68.7250754
## cost_div_43                                  14.0590363   107.3882519
## cost_div_44                                  13.8172084    69.9137670
## cost_div_48                                   0.2497902     0.6231834
## has_div_21_cost                               4.5397020     2.5120055
## has_div_31_cost                               4.1625242     5.0708088
## has_div_22_cost                               5.6582770     3.8936933
## has_div_48_cost                               1.5112340     0.5270865
## line_item_count                              15.7808704    52.7471061
## total_material_sq_ft                         19.0555338   342.0930918
## divisions_with_cost                          10.9999121    55.3609782
## average_line_item_cost                       21.1065982   370.0252884
```

```r
varImpPlot(rf_model)
```



Predict using random forest

```
predictions <- predict(rf_model, newdata = test_data)
predictions <- exp(predictions)
```

Evaluate random forest model

```
true_values <- exp(test_data$log_total_project_cost)
MAE <- mean(abs(predictions - true_values))
RMSE <- sqrt(mean((predictions - true_values)^2))


# R-squared
SS_res <- sum((true_values - predictions)^2)
SS_tot <- sum((true_values - mean(true_values))^2)
R2 <- 1 - (SS_res / SS_tot)

# Print results
cat("MAE:", round(MAE, 2), "\n")
```

```
## MAE: 19119054
```

```
cat("RMSE:", round(RMSE, 2), "\n")
```

```
## RMSE: 80078924
```

```r
cat("R-squared:", round(R2, 4), "\n")
```

```
## R-squared: 0.7436
```

Normalize predictors to use in Neural Network model

```r
train_nn <- train_data
test_nn <- test_data
factor_cols <- c("project_state", "type", "project_category", "construction_category")

train_nn[factor_cols] <- lapply(train_nn[factor_cols], as.factor)
for (col in factor_cols) {
  test_nn[[col]] <- factor(test_nn[[col]], levels = levels(train_nn[[col]]))
}

# Identify numeric predictors (excluding target and factors)
numeric_cols <- setdiff(names(train_nn), c("log_total_project_cost", factor_cols))

# Scale numeric columns using training set statistics
train_nn[numeric_cols] <- scale(train_nn[numeric_cols])

# Scale test set using same stats
center_vals <- attr(scale(train_data[numeric_cols]), "scaled:center")
scale_vals  <- attr(scale(train_data[numeric_cols]), "scaled:scale")
test_nn[numeric_cols] <- scale(test_nn[numeric_cols], center = center_vals, scale = scale_vals)
```

```r
train_nn <- na.omit(train_nn)
test_nn  <- na.omit(test_nn)
```

Train Neural Network

```r
set.seed(42)
nn_model <- nnet(
  log_total_project_cost ~ .,
  data = train_nn,
  size = 1,         # Number of hidden units
  linout = TRUE,    # Regression (not classification)
  maxit = 1000       # Max iterations
)
```

```
## # weights:  118
## initial  value 652659.331534
## iter  10 value 5814.580864
## iter  20 value 5266.102080
## iter  30 value 2870.086964
## iter  40 value 2453.819731
## iter  50 value 1855.702838
## iter  60 value 1717.784677
## iter  70 value 1649.091196
## iter  80 value 1479.711400
## iter  90 value 1288.925117
## iter 100 value 1073.048327
```

```
## iter 110 value 986.360576
## iter 120 value 932.029383
## iter 130 value 883.891787
## iter 140 value 858.464512
## iter 150 value 834.585359
## iter 160 value 822.080937
## iter 170 value 814.778235
## iter 180 value 766.918667
## iter 190 value 756.569499
## iter 200 value 738.581343
## iter 210 value 725.468552
## iter 220 value 713.830199
## iter 230 value 679.768015
## iter 240 value 648.287532
## iter 250 value 606.502078
## iter 260 value 567.856270
## iter 270 value 553.486088
## iter 280 value 550.798374
## iter 290 value 550.435445
## iter 300 value 550.305528
## iter 310 value 548.199666
## iter 320 value 548.098100
## final  value 548.097858
## converged
```

Predict and evaluate neural network

```r
# Predict log(cost)
log_preds <- predict(nn_model, newdata = test_nn)

# Convert back to original scale
preds <- exp(log_preds)
actual <- exp(test_nn$log_total_project_cost)

# Performance metrics
MAE <- mean(abs(preds - actual))
RMSE <- sqrt(mean((preds - actual)^2))

# R-squared
SS_res <- sum((actual - preds)^2)
SS_tot <- sum((actual - mean(actual))^2)
R2 <- 1 - (SS_res / SS_tot)

cat("MAE:", round(MAE, 2), "\n")
```

```
## MAE: 35238431
```

```r
cat("RMSE:", round(RMSE, 2), "\n")
```

```
## RMSE: 125362737
```

```r
cat("R-squared:", round(R2, 4), "\n")
```

```
## R-squared: 0.3715
```

Put results together

```r
results_df <- data.frame(
  Actual = exp(test_data$log_total_project_cost),
  Predicted_rf = predictions,
  Predicted_nn = preds
)
```