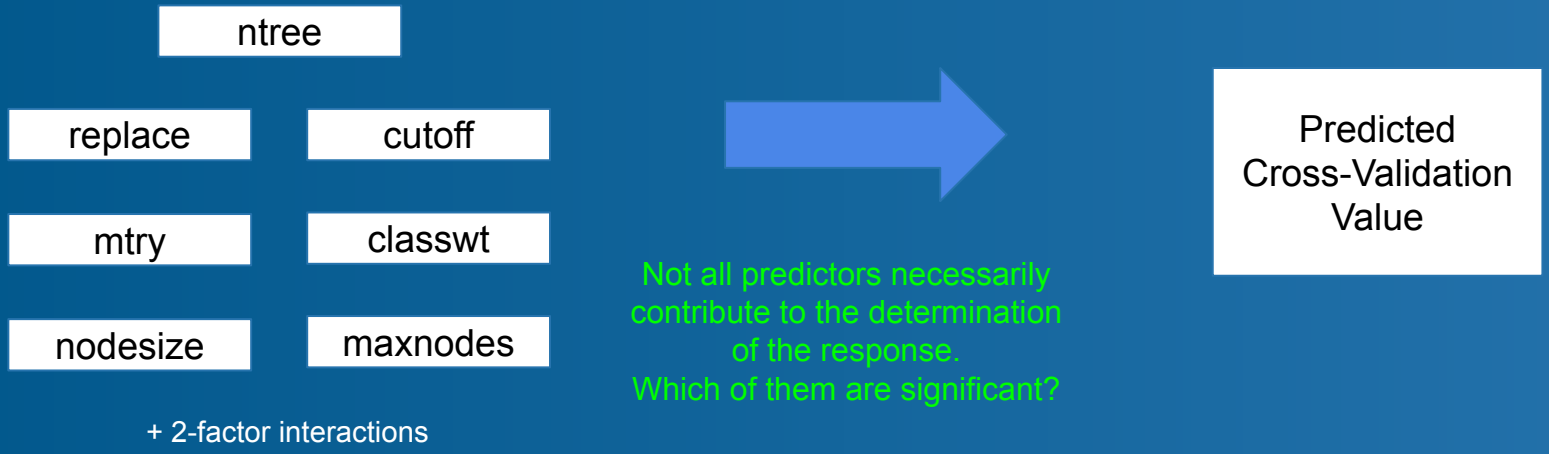
UCLA

Parameter Tuning of Random Forest

UCLA

Introduction

- Goal: identify the subset of tuning parameters of a random forest tree algorithm that affects cross-validation accuracy



UCLA

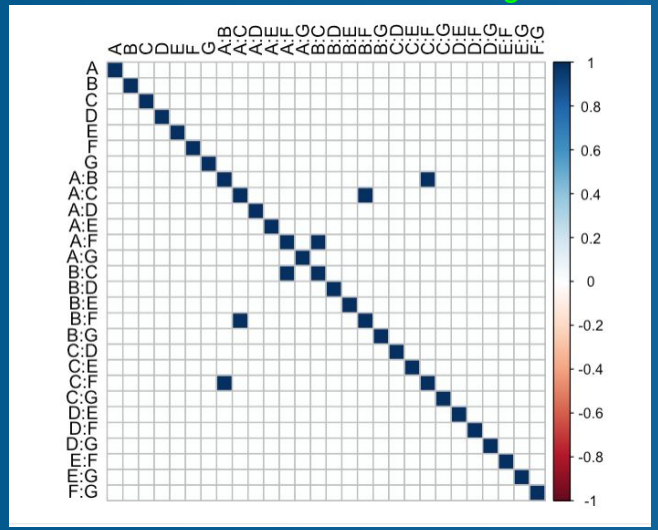
Methodology

1. Determine an experimental design
    a. Limit resources used i.e. 35 runs
    b. Reduce aliasing
2. Collect data with experimental design using cv.rf function and diabetes data
    a. Use diabetes data to build random forest, and calculate cross-validation value according to combination of random forest tuning parameters in experimental design
3. Determine Final Model
    a. Perform analysis to reach final model with important predictors that maximize cross-validation accuracy
    b. Ensure model is valid
4. Confirmation Experiments

UCLA

1. Determining Experimental Design

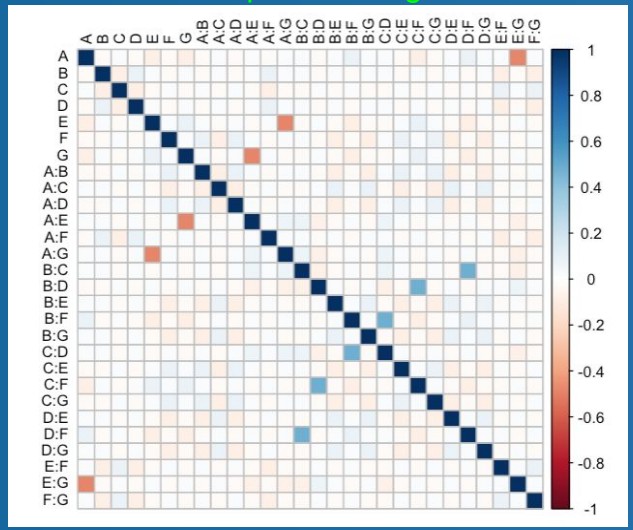
2^{7-2} Fractional Factorial design



- 32 runs
- Three pairs of factors are fully aliased; A:C, A:F, and C:F and are inestimable
- Remaining factors not aliased

Our recommended design: 2^{7-2} fractional-factorial

D-optimal Design



- 35 runs
- Six pairs of factors are partially aliased, and most factors are at least a little correlated with other factors

UCLA

2. Collect Data With Our Design

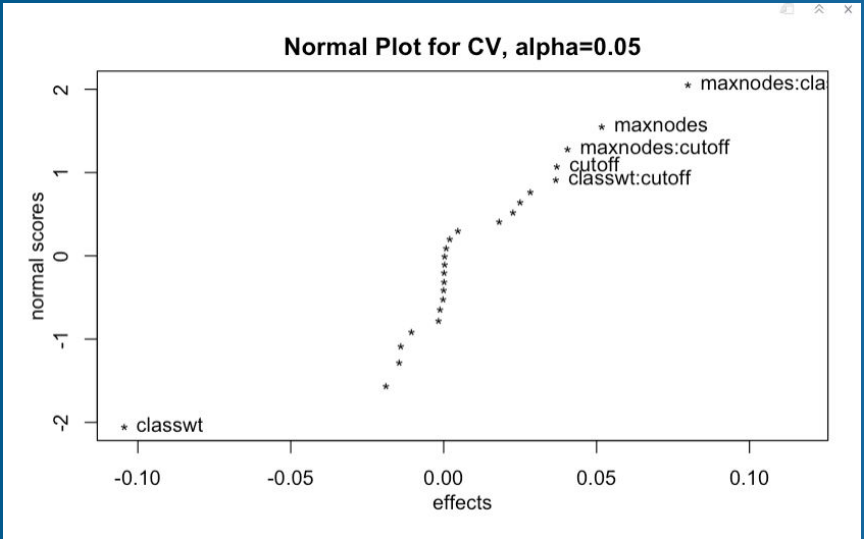
- First 10 rows of dataset generated using fractional-factorial design:

	ntree	replace	mtry	nodesize	maxnodes	classwt	cutoff	CV
1	100	0	2	1	10	0.5	0.8	0.6635027
2	1000	0	2	1	10	0.9	0.2	0.5000001
3	100	1	2	1	10	0.9	0.2	0.5000000
4	1000	1	2	1	10	0.5	0.8	0.6650844
5	100	0	6	1	10	0.9	0.8	0.5093115
6	1000	0	6	1	10	0.5	0.2	0.7096307
7	100	1	6	1	10	0.5	0.2	0.7074754
8	1000	1	6	1	10	0.9	0.8	0.5141738
9	100	0	2	11	10	0.5	0.2	0.6804575
10	1000	0	2	11	10	0.9	0.8	0.5003956

UCLA

3. Determine Final Model

- First, fit full model using fractional-factorial design
  - Includes all main factors + two-factor interactions (excluding A:C, B:C and C:F)
- Next, determine which factors are actually significant



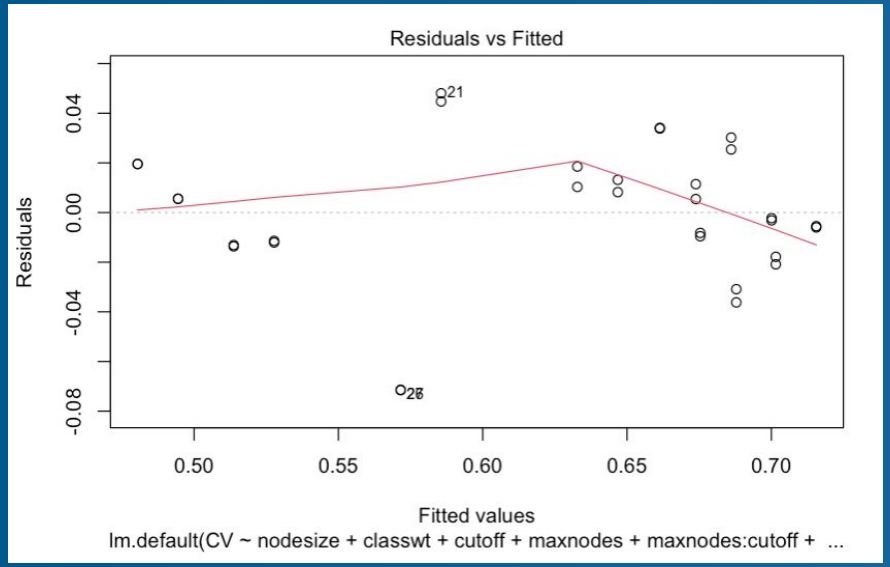
Revised Model's predictors:

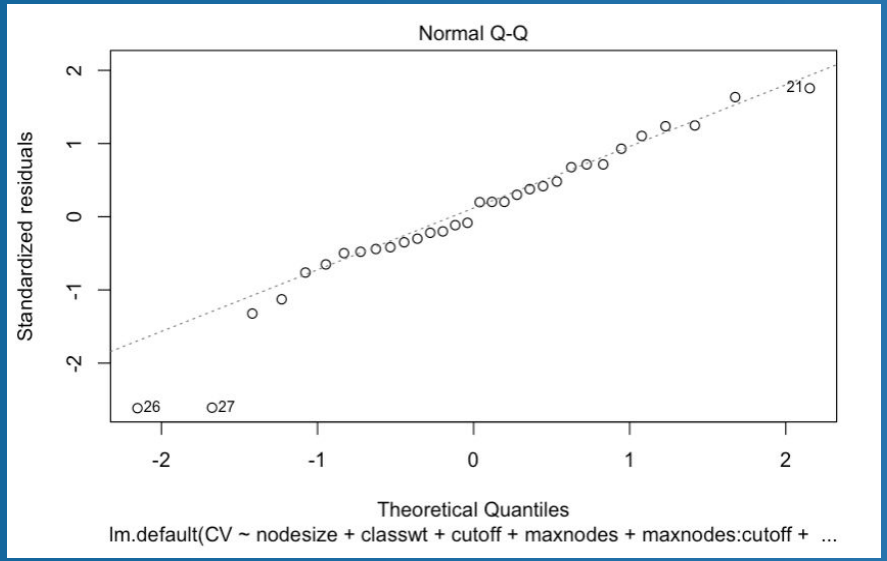
- nodesize
- classwt
- cutoff
- maxnodes:cutoff
- classwt:cutoff
- maxnodes:classwt
- maxnodes

UCLA

- Revised Model: 7 predictors (versus initial 25 predictors)

- Now, we must check that the model's assumptions are valid:

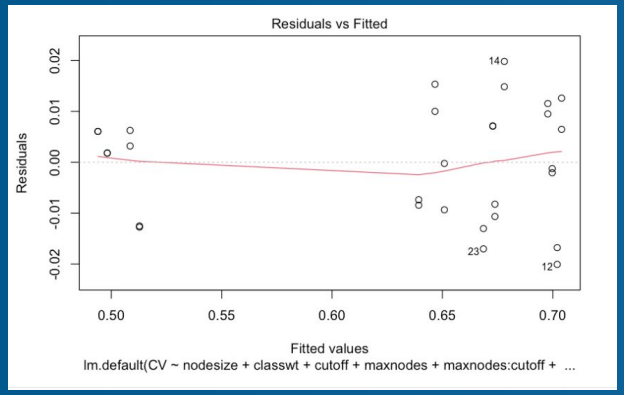


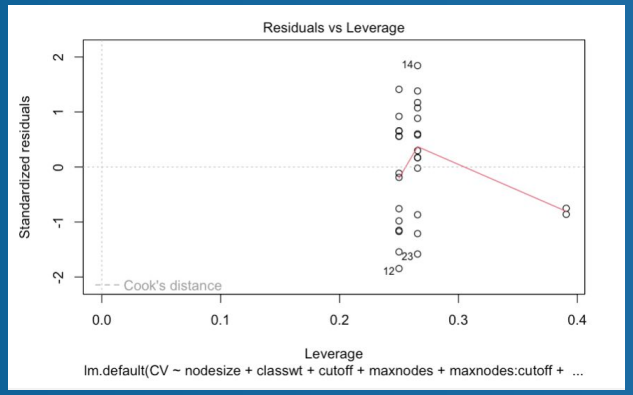


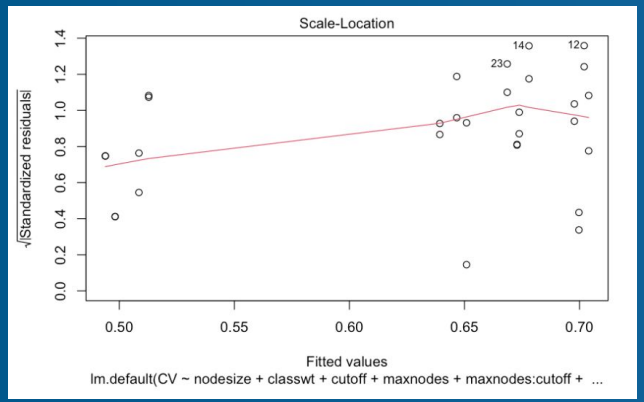
Slight trend in residuals according to residuals vs fitted plot, therefore predictions may be inaccurate. However, remaining assumptions are satisfied. Two outliers (obs. 26 and 27, so **remove points and refit model.**

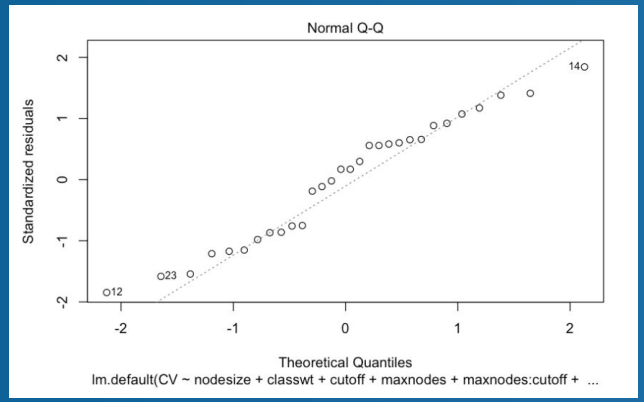
UCLA

- Regression Diagnostics with Outliers Removed









- Outliers with high standardized residuals removed
- Constant variance assumption satisfied
- No influential points outside Cook's Distance
- Normality assumption satisfied

UCLA

4. Confirmation Experiment

- All predictions are considered “good” or “accurate”
- High R^2 : 0.98

1	2	3	4	5	6
0.0106494449	0.0060715984	0.0060716300	0.0082405694	0.0031993263	0.0115415106
7	8	9	10	11	12
0.0095057046	0.0062664334	0.0167701641	0.0127396004	0.0125129810	0.0200638728
13	14	15	16	17	18
0.0018217898	0.0198204395	0.0148563964	0.0018218036	0.0153422280	0.0020505919
19	20	21	22	23	24
0.0012414321	0.0100053010	0.0073556092	0.0130219509	0.0170129215	0.0084312125
25	28	29	30	31	32
0.0070931004	0.0071549502	0.0064727917	0.0002270095	0.0093336979	0.0126060540

```
Call:
lm.default(formula = CV ~ nodesize + classwt + cutoff + maxnodes +
    maxnodes:cutoff + classwt:cutoff + maxnodes:classwt, data = rev.cv.fr.design)

Residuals:
      Min        1Q    Median        3Q       Max
-0.020064 -0.009108  0.001822  0.007140  0.019820

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)       9.793e-01  1.913e-02  51.192  < 2e-16 ***
nodesize          4.250e-04  4.707e-04   0.903 0.376360
classwt          -5.466e-01  2.519e-02 -21.696 2.42e-16 ***
cutoff           -1.210e-01  2.944e-02  -4.110 0.000461 ***
maxnodes         -3.151e-04  1.799e-05 -17.512 2.10e-14 ***
cutoff:maxnodes   7.726e-05  1.585e-05   4.875 7.14e-05 ***
classwt:cutoff    1.606e-01  3.922e-02   4.094 0.000480 ***
classwt:maxnodes  4.961e-04  2.377e-05  20.867 5.48e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.01255 on 22 degrees of freedom
Multiple R-squared:  0.9817,    Adjusted R-squared:  0.9758
F-statistic: 168.4 on 7 and 22 DF,  p-value: < 2.2e-16
```

UCLA

5. Maximizing Cross Validation Accuracy

- Find final fitted equation using summary function

```
$par
[1]  1 -1 -1 -1
```

- Using optim function in R to find levels that maximize CV

- Higher level (11) for the nodesize, and the lower level for classwt (0.5), cutoff (0.2), and maxnodes (10) will maximize the CV

```
obj_func <- function(x){
    pred.y <- 9.772e-01 + 3.543e-04*x[1] -5.444e-01*x[2] - 1.223e-01*x[3] -
-3.096e-04*x[4] +  7.235e-05*x[3]*x[4] + 1.663e-01*x[2]*x[3] + 4.912e-04*x[2]*x[4]
    return(-1*pred.y) # Because the 'optim' function minimizes.
}

optim(par = c(0, 0, 0, 0), fn = obj_func, lower = -1, upper = 1, method = "L-BFGS-B")
```

UCLA

Conclusions

- Given the budget of 35 runs, we wanted to minimize the computational cost of our design, while reducing multicollinearity and aliasing; we believe we accomplished this task
- Strengths of Model:
  - Predictions appear accurate
  - No partial aliasing/multicollinearity between factors that are present
  - Involves three less runs than our budget
- Weaknesses of Model:
  - Did not initially take into account three interactions, so may be missing out on potential significant interaction
- Future Recommendations:
  - See how we can make use of the three leftover runs that we did not initially use
  - See if we can minimize the number of factors we must take out without succumbing to partial aliasing of other factors