Evaluate testing data (binary-class) - XGBoost

EVE W.

2019-11-16

Contents

```r
## user input
project_home <- "~/EVE/examples"
project_name <- "xgboostR_binary_2"
```

0. Load Data

```
## Warning: `cols` is now required.
## Please use `cols = c(df)`

## Warning: `cols` is now required.
## Please use `cols = c(df)`
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   Patient_ID = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

```
## 199 of samples were used
```

```
## 100 of full features
```

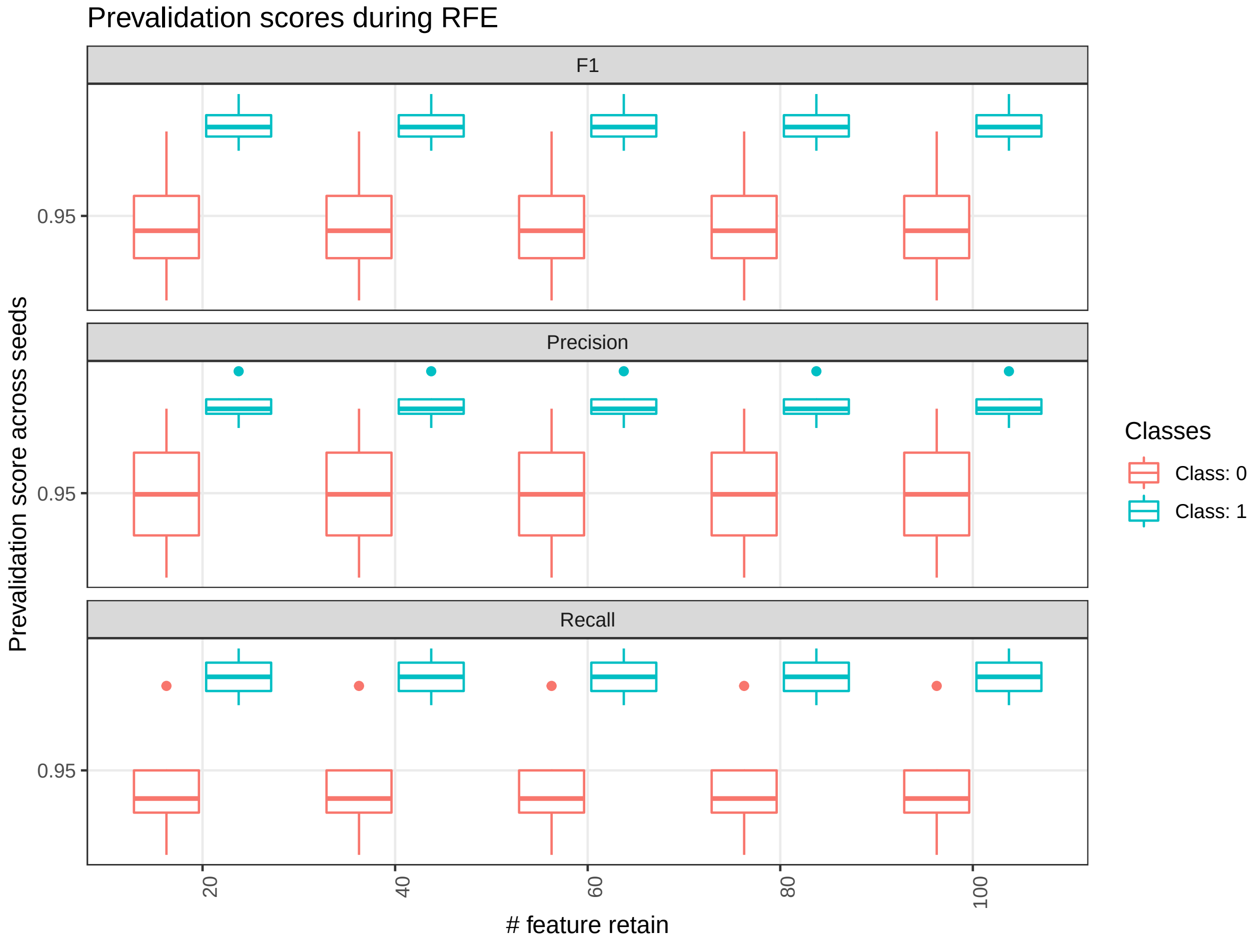
```
## 4 runs, each run contains 3 CVs.
```

```
## Labels:
```

run with XGBoost.r evaluation metric: f1_harmonic2.

1. Scores

1.1 Scores per Class



Confusion Matrix

```
## confusion matrix at feature size = 100
## sum across 4 seeds
```

```
##            Reference
## Prediction   0   1
##          0 189  10
##          1  11 586
```

1.2 Average score

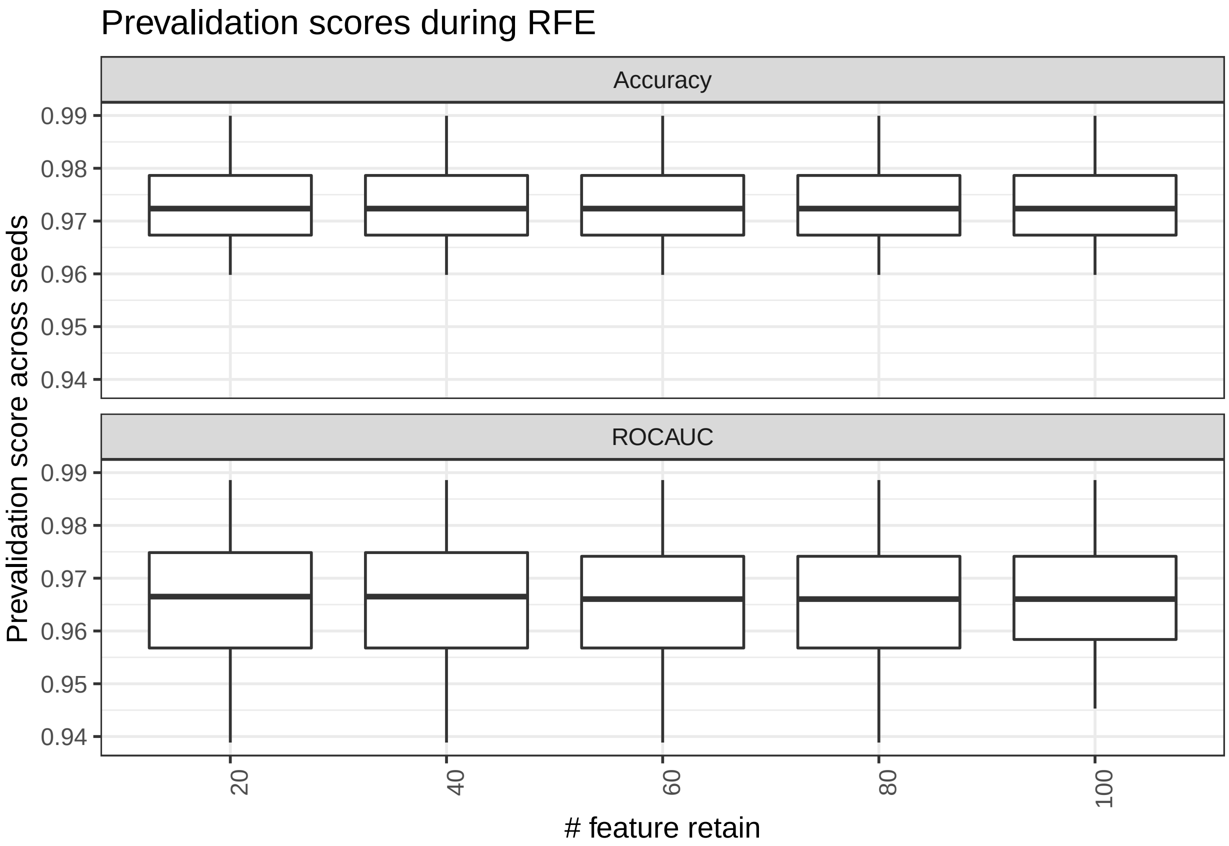
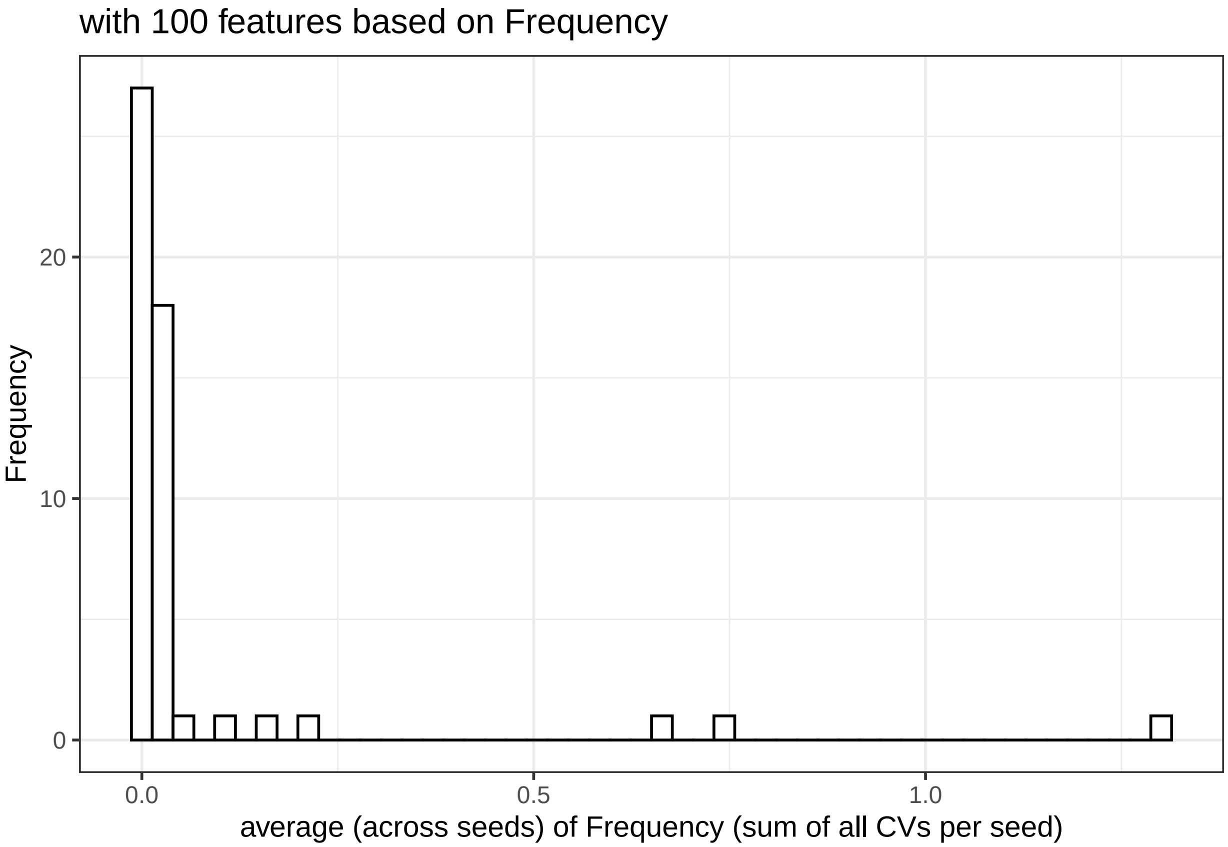
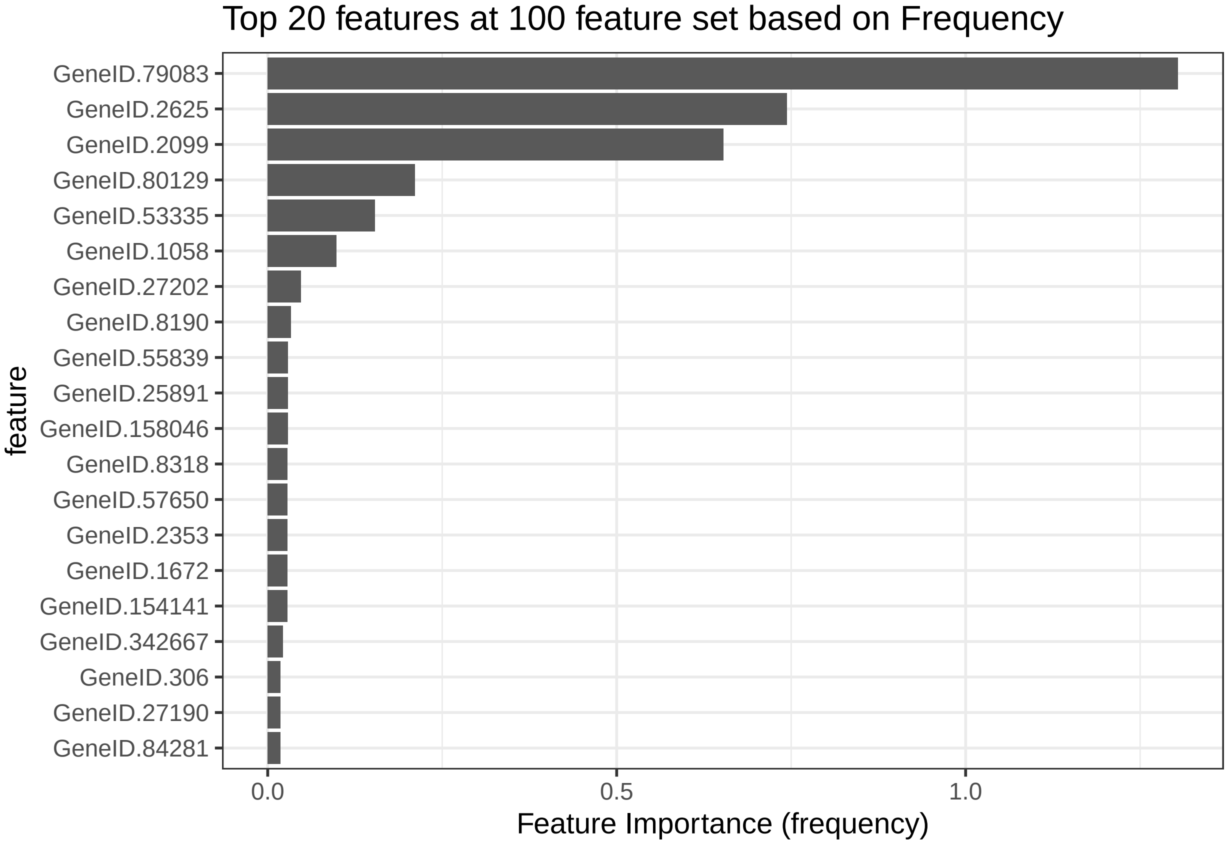


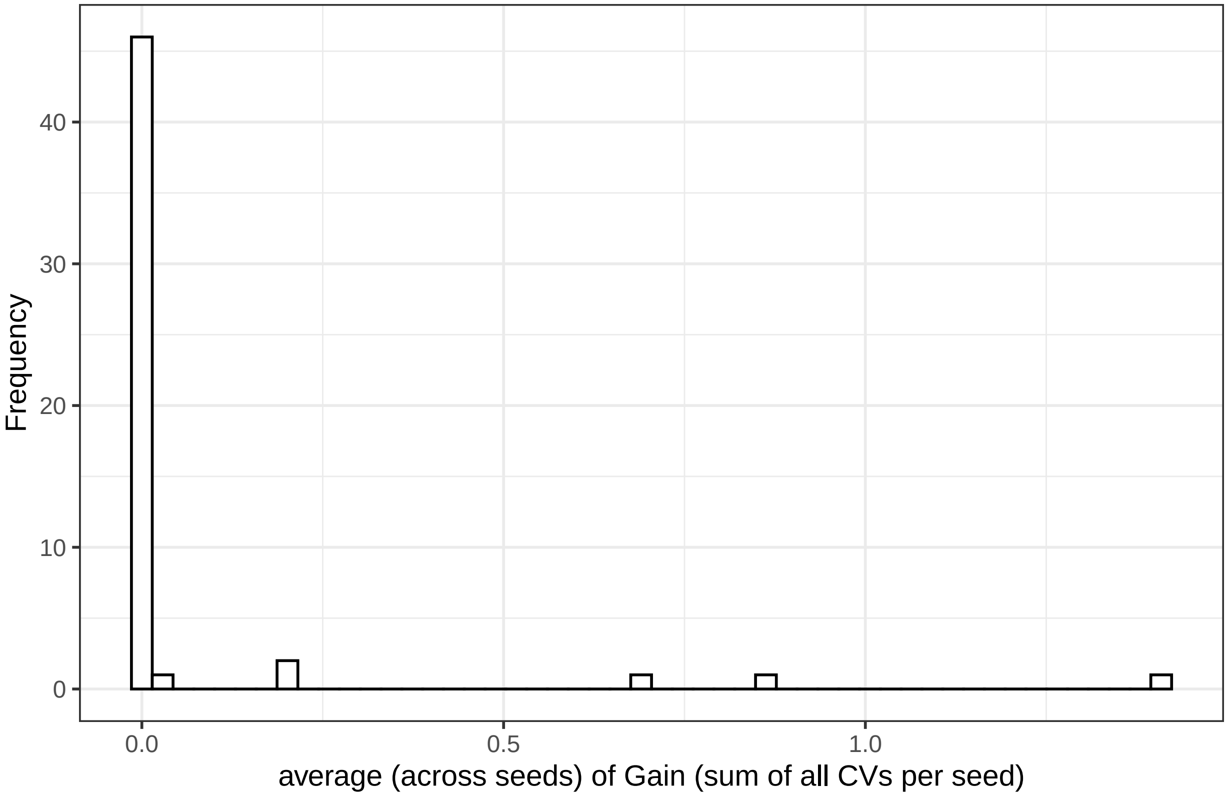
Table 1: best scores

metrics	size.max	median.max	size.min	median.min
Accuracy	20	0.972	20	0.972
F1	20	0.963	20	0.963
Precision	20	0.965	20	0.965
Recall	20	0.962	20	0.962
ROCAUC	20	0.967	60	0.966

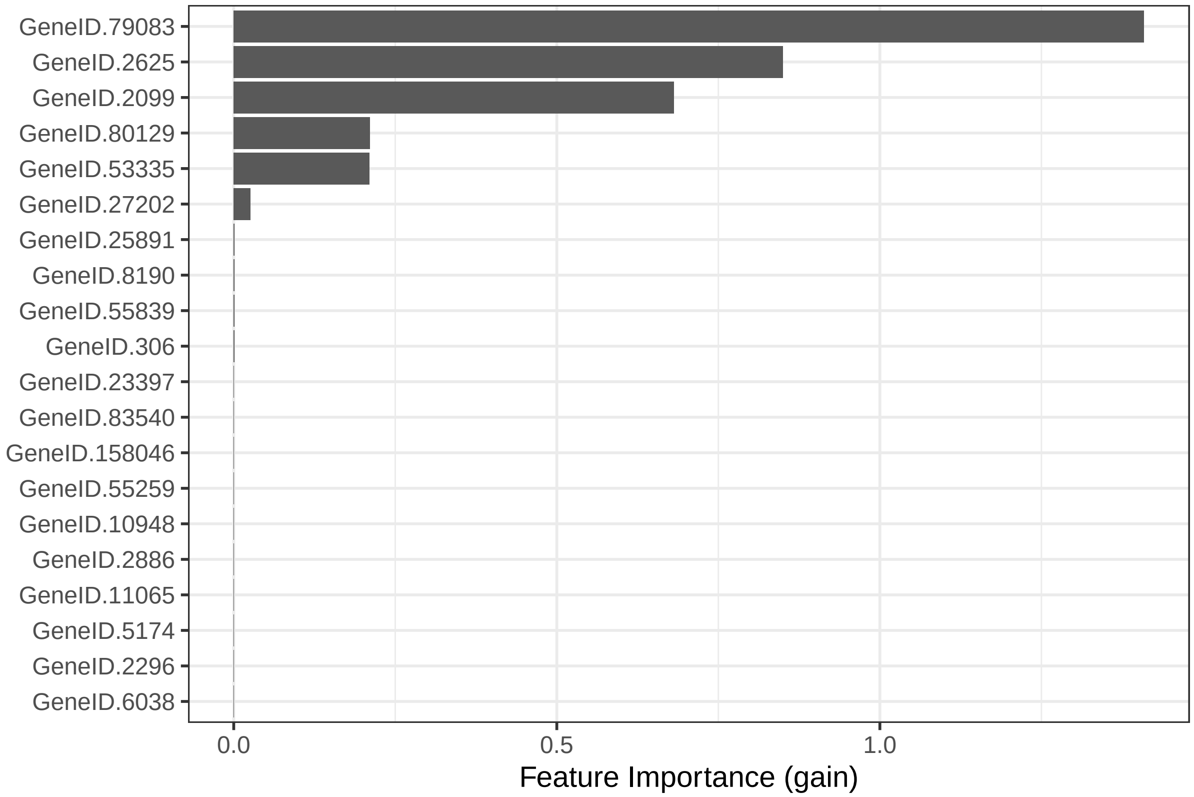
2. Important Features



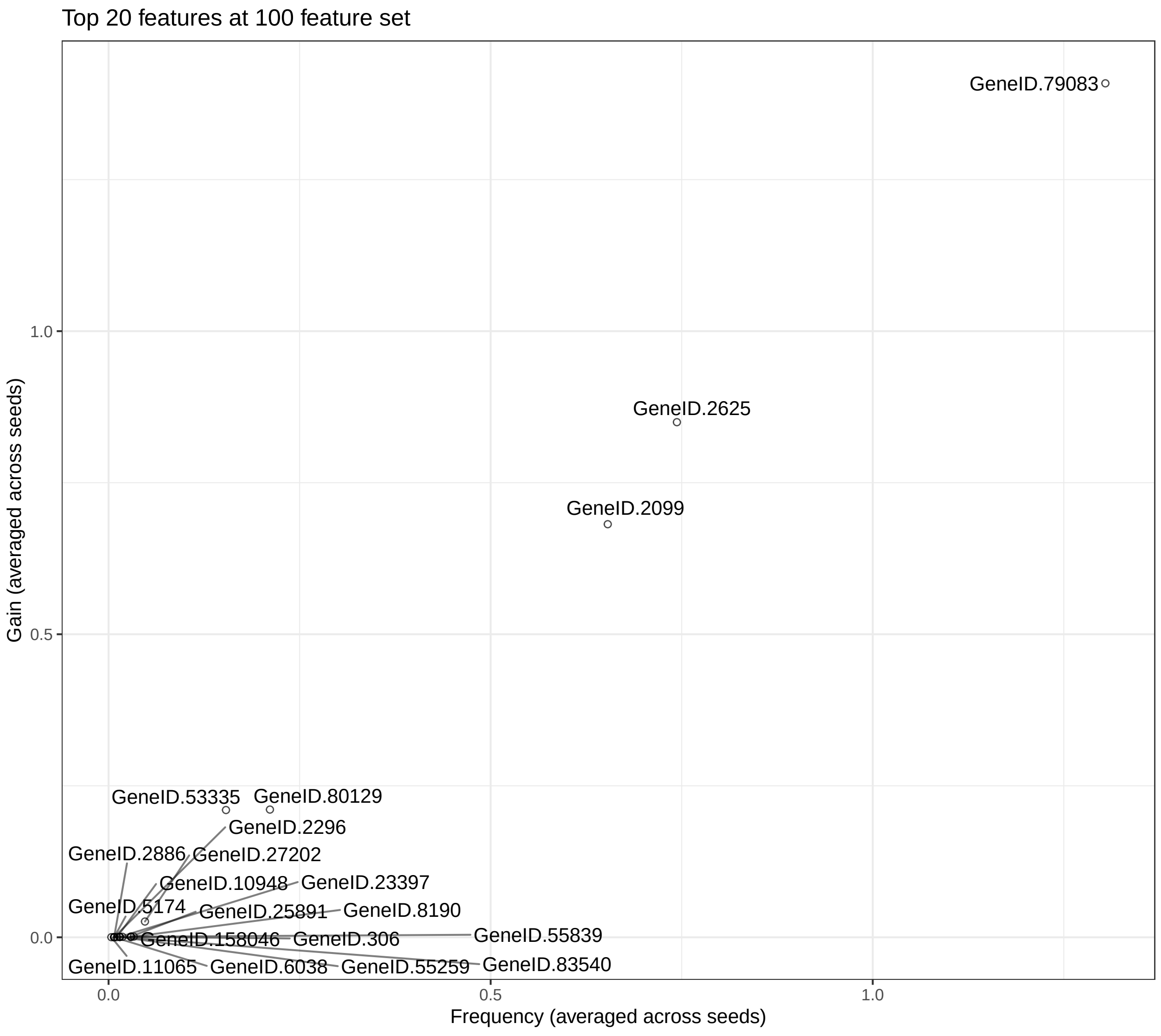


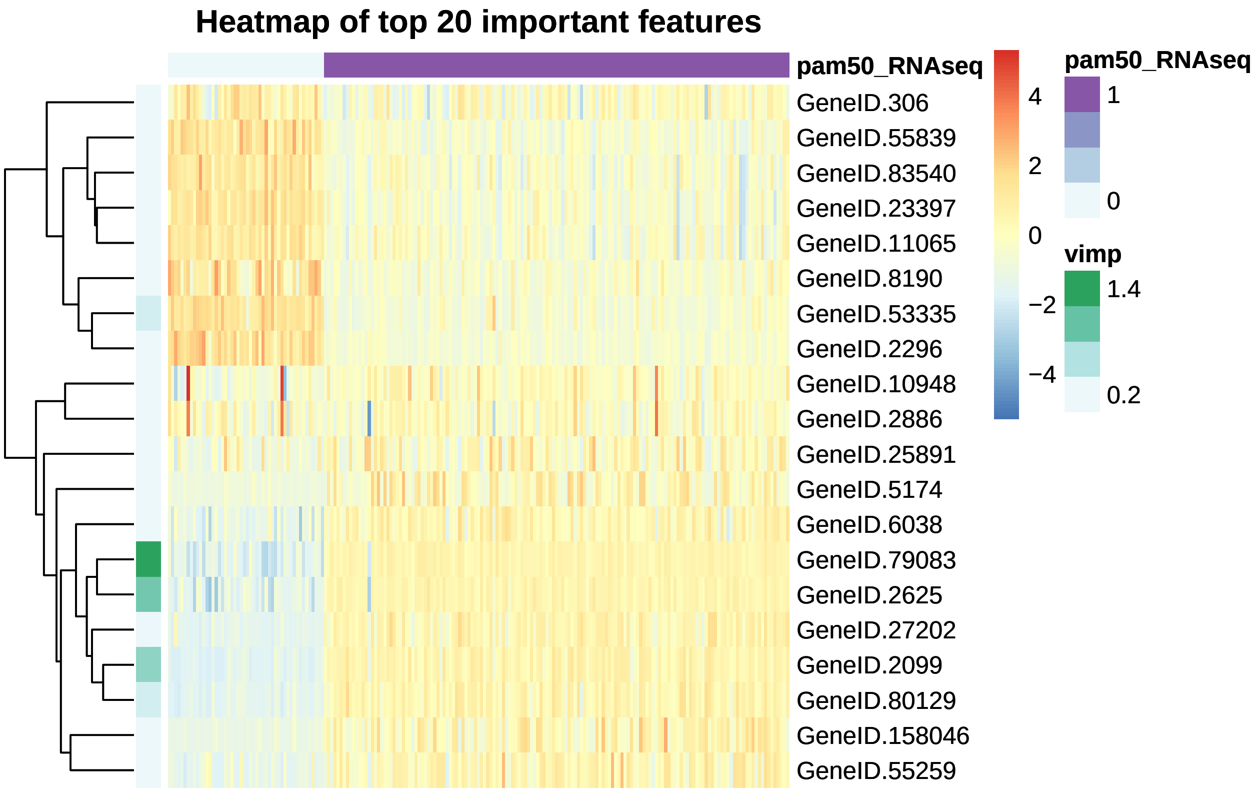


with 100 features based on Gain



Top 20 features at 100 feature set based on Gain


Top 20 features at 100 feature set
Gain (averaged across seeds)
Frequency (averaged across seeds)
0.0
0.5
1.0
0.0
0.5
1.0
GeneID.79083
GeneID.2625
GeneID.2099
GeneID.53335
GeneID.80129
GeneID.2296
GeneID.2886
GeneID.27202
GeneID.10948
GeneID.23397
GeneID.5174
GeneID.25891
GeneID.8190
GeneID.158046
GeneID.306
GeneID.55839
GeneID.11065
GeneID.6038
GeneID.55259
GeneID.83540

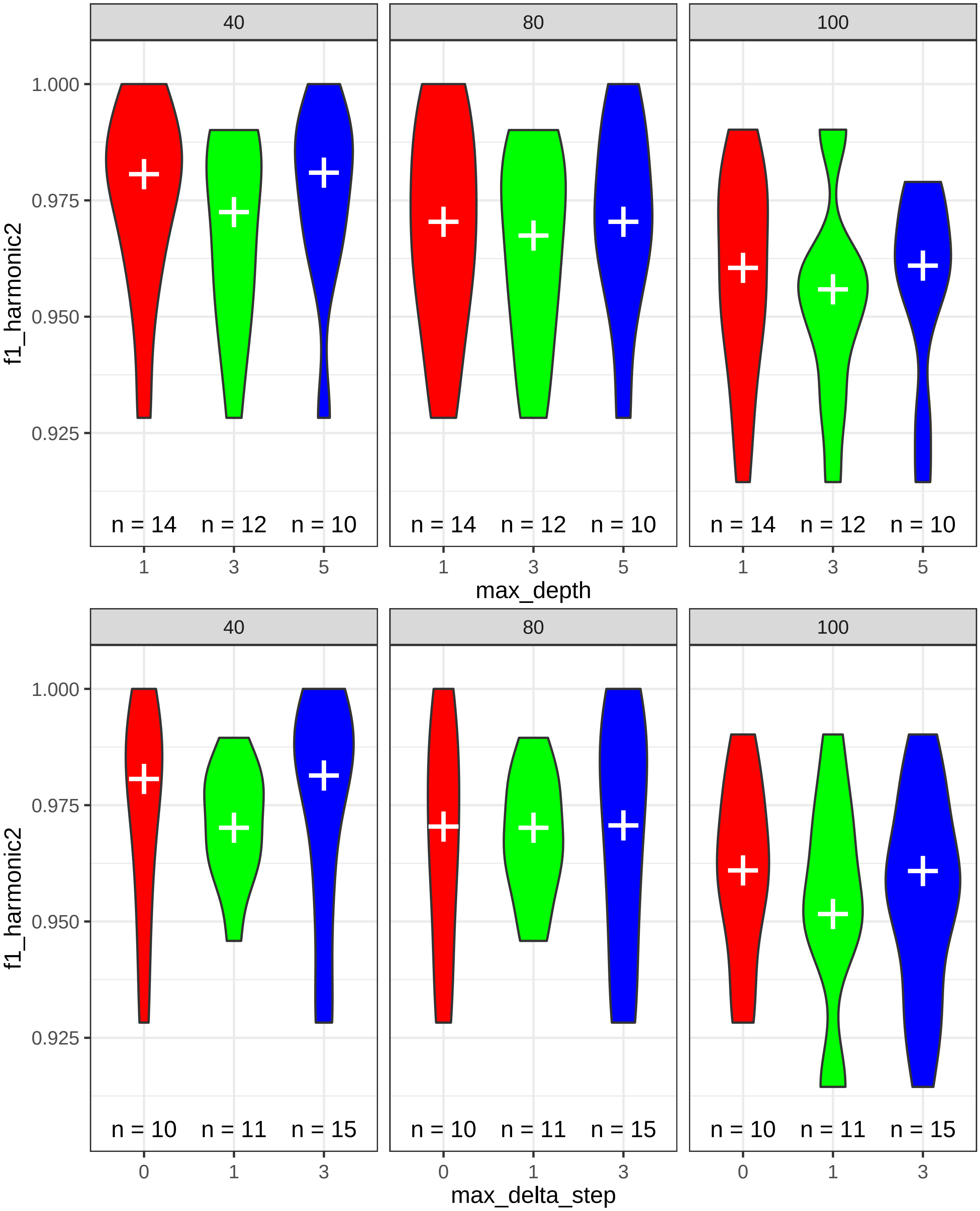


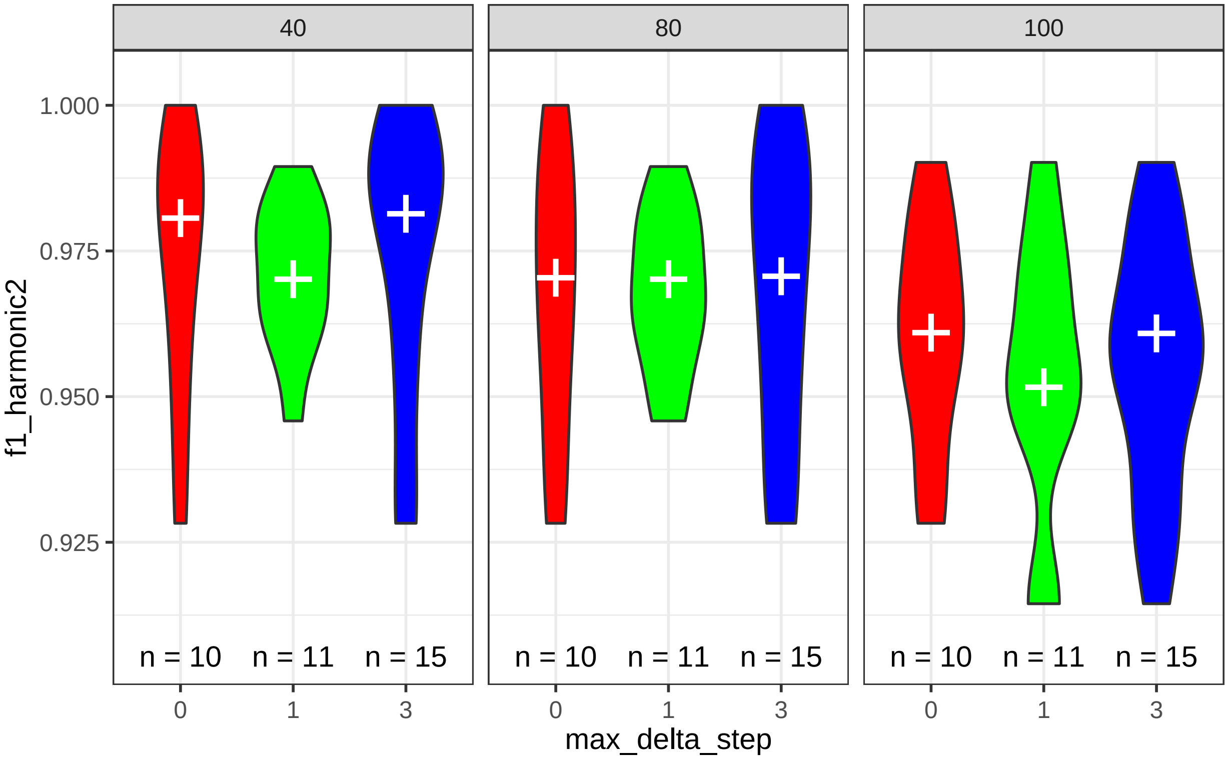
3. Hyper-parameters

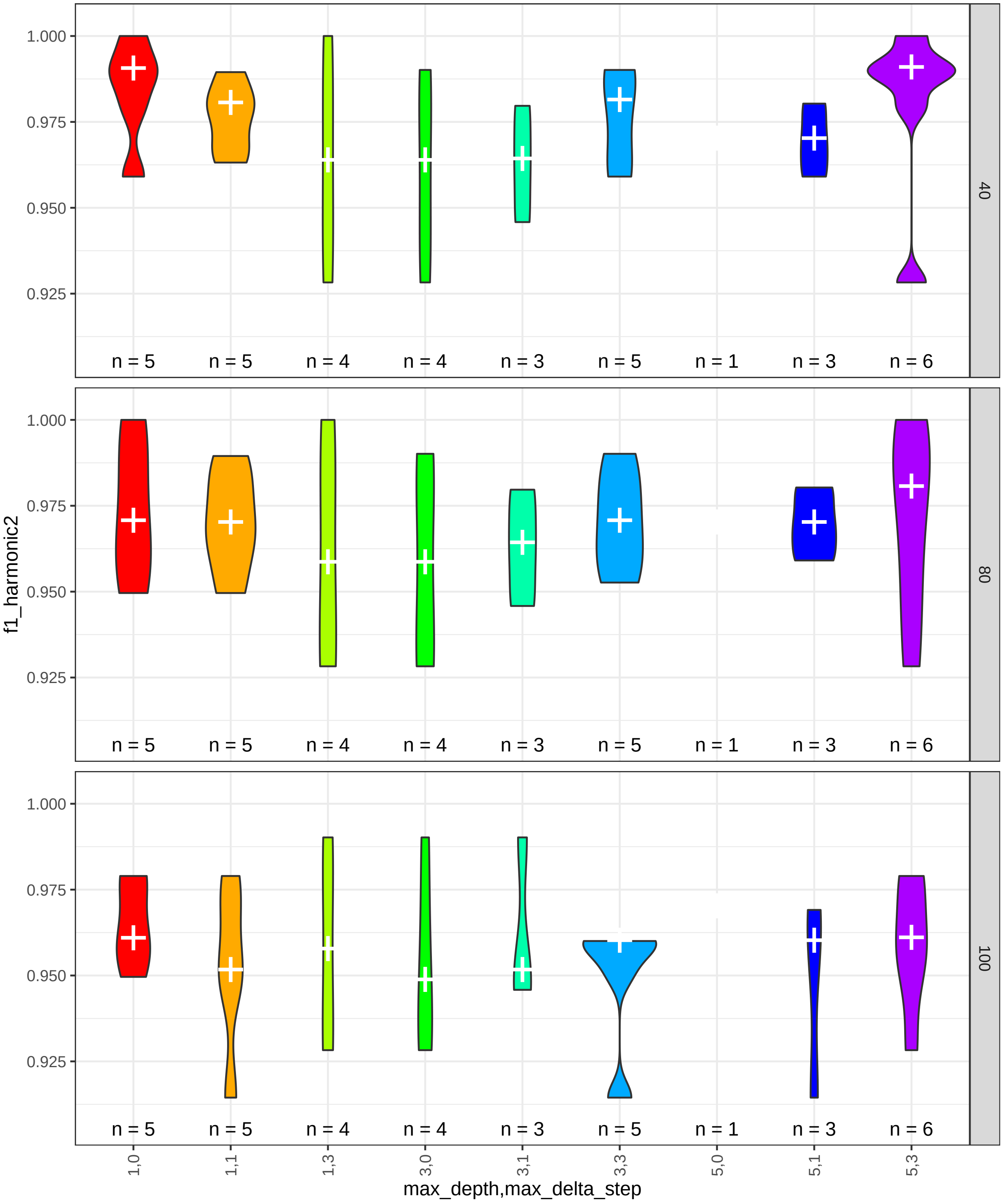
```
## Warning: `cols` is now required.
## Please use `cols = c(df)`
```

parameter optimization file (108 records) includes 4 seeds. Each seed generates 3 cv splits. Within each cv split, there is a 3 step RFE (at 40, 80, 100). So 108 / 4 / 3 / 3 = 3 parameter combinations tried in each cv split.

all grid search results

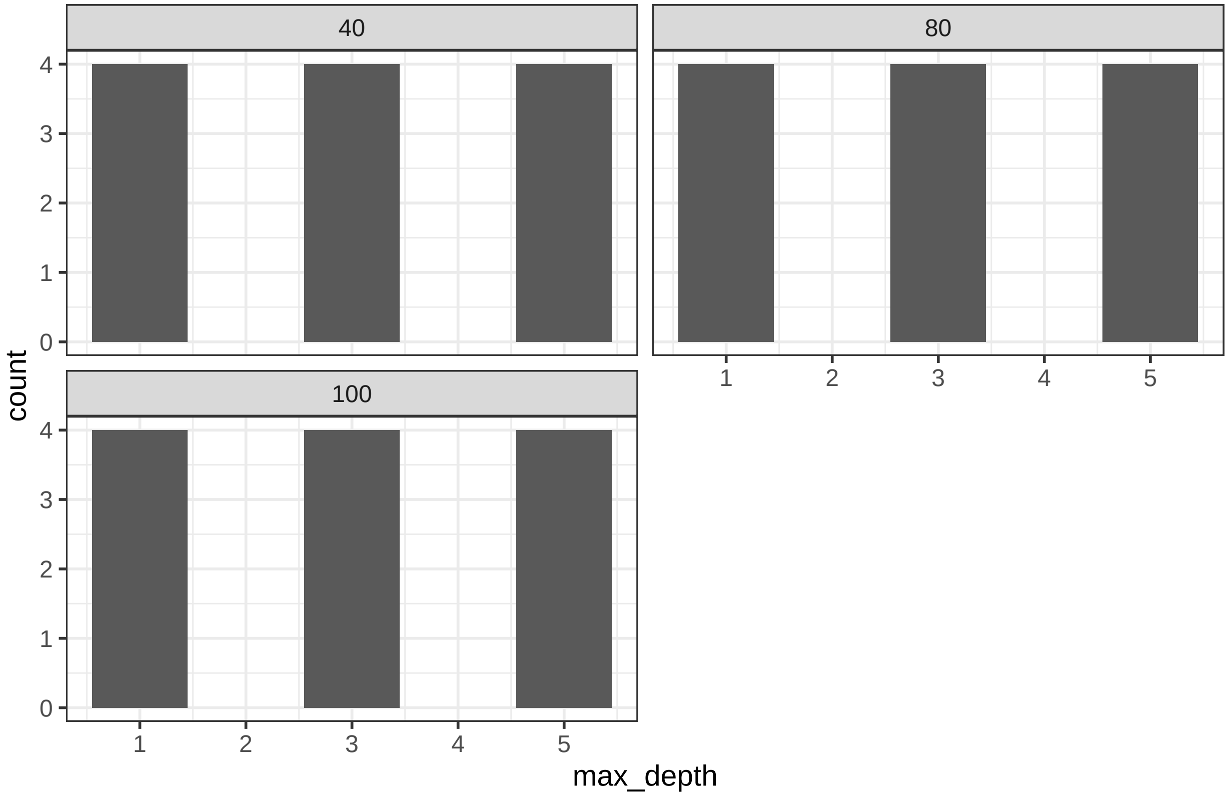



40
80
100
1.000
0.975
0.950
0.925
f1_harmonic2
n = 10
n = 11
n = 15
n = 10
n = 11
n = 15
n = 10
n = 11
n = 15
0
1
3
0
1
3
0
1
3
max_delta_step

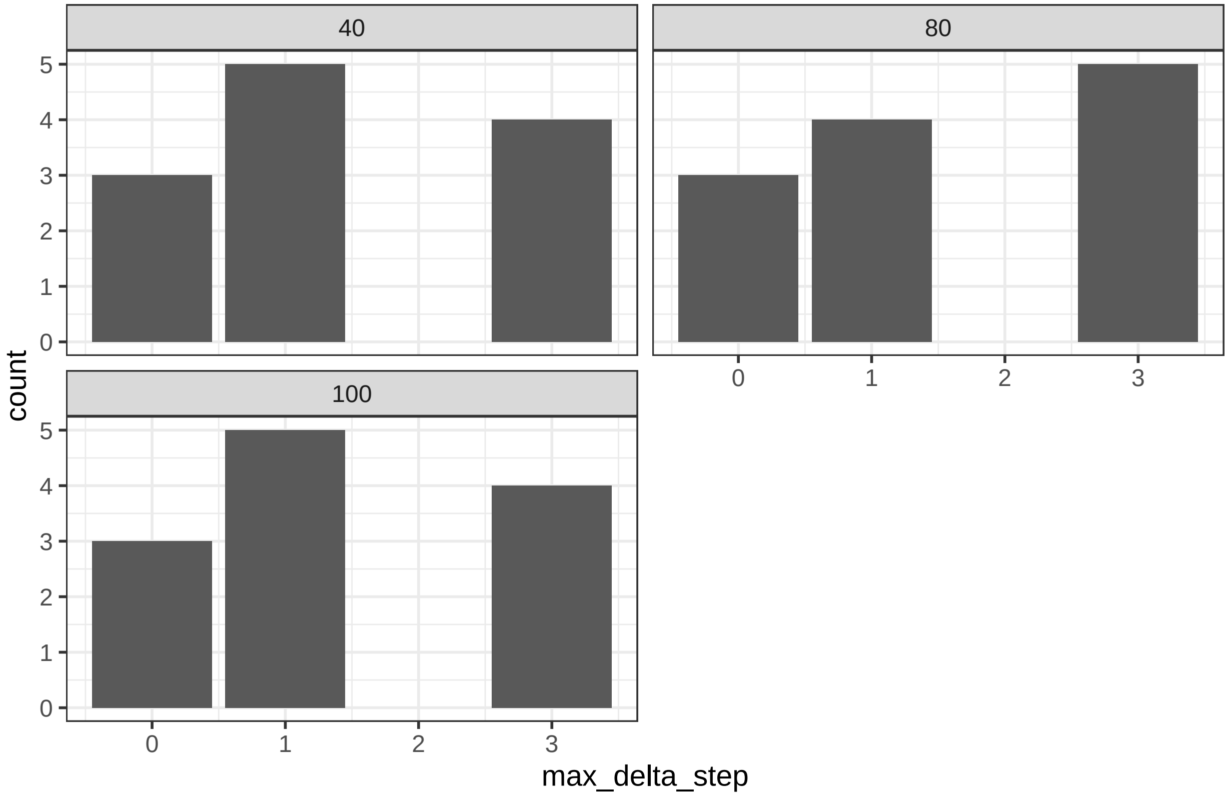


over best parameter combo per cv

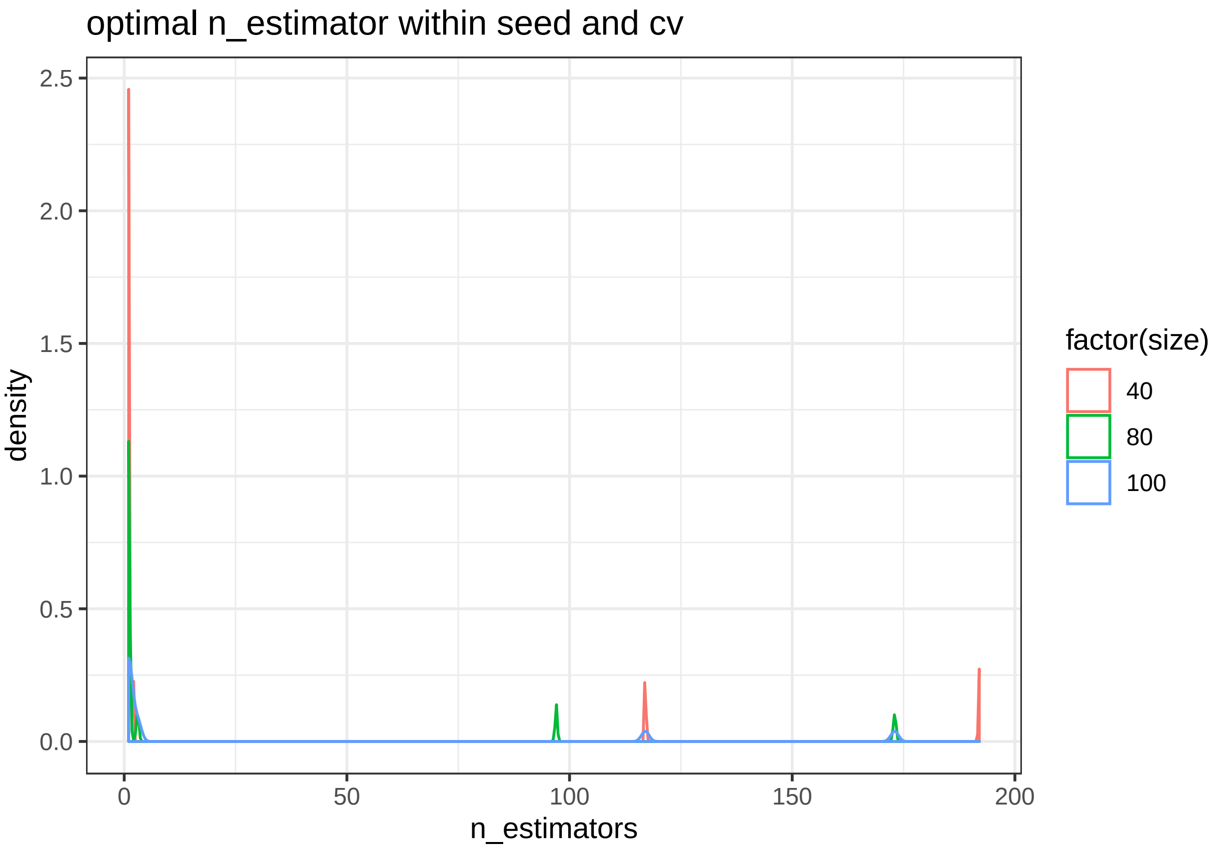
Note the 2nd /3rd best parameter combinations might not be too bad either.



optimal max_depth across seed and cv



optimal max_delta_step across seed and cv



more about the best parameter combination selection

```
select_ft_step <- 100

df1 <- subset(grid_best, size==select_ft_step & max_depth==1 &  max_delta_step == 0 )
print( paste('summary of n estimator at',select_ft_step, 'feature step'))
```

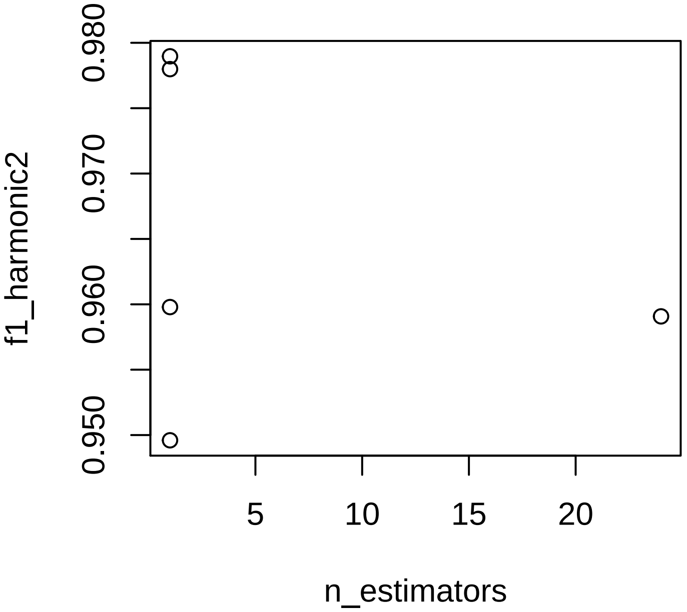
```
## [1] "summary of n estimator at 100 feature step"
```

```
print(summary(df1$n_estimators))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1       1       1       1       1       1
```

```
df2 <- subset(df.grid, size==select_ft_step & max_depth==1 &  max_delta_step == 0 )

with(df2, plot(x = n_estimators, y=score, ylab=score_label))
```


f1_harmonic2
0.950
0.960
0.970
0.980
5
10
15
20
n_estimators