Evaluate testing data (binary-class) - Lasso

EVE W.

2020-04-24

Contents

Note: The two differences between Lasso and Tree-based methods are:

1. Lasso has its own inherent feature selection process.
2. Lasso's vimp will be based on how many times the feature exist in all runs. Regression coefficients may be presented for binary outcomes

```
## user input
project_home <- "~/EVE/examples"
project_name <- "lasso_binary2"
```

0. Load Data

```
## Error : $ operator is invalid for atomic vectors
```

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

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

```
## 4 runs, each run contains 5 CVs.
```

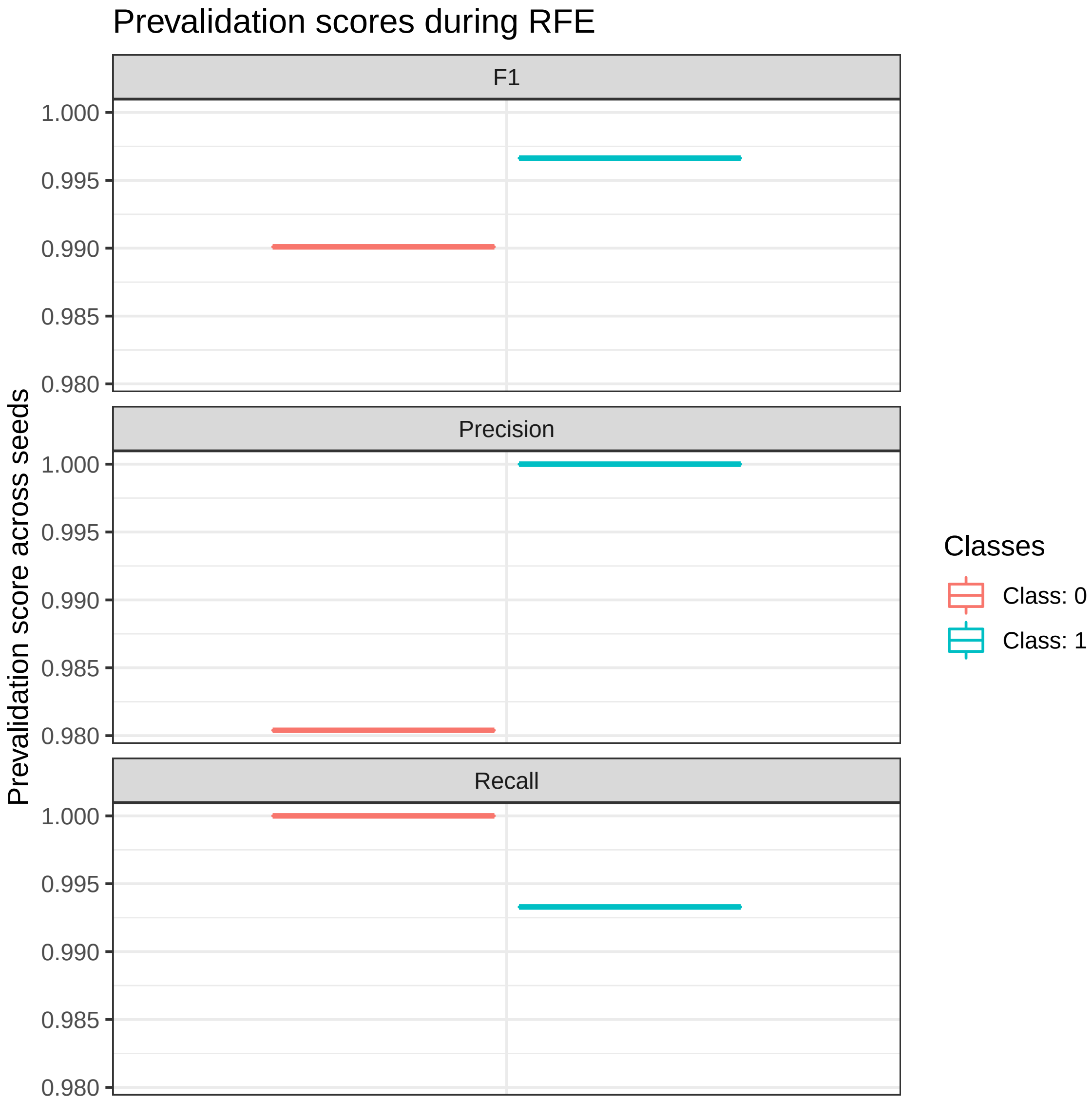
```
## Labels:
```

```
##
##   0   1
##  50 149
```

run with lasso.r with alpha = 0, 0.5, 1.

1. Scores

1.1 Scores per Class

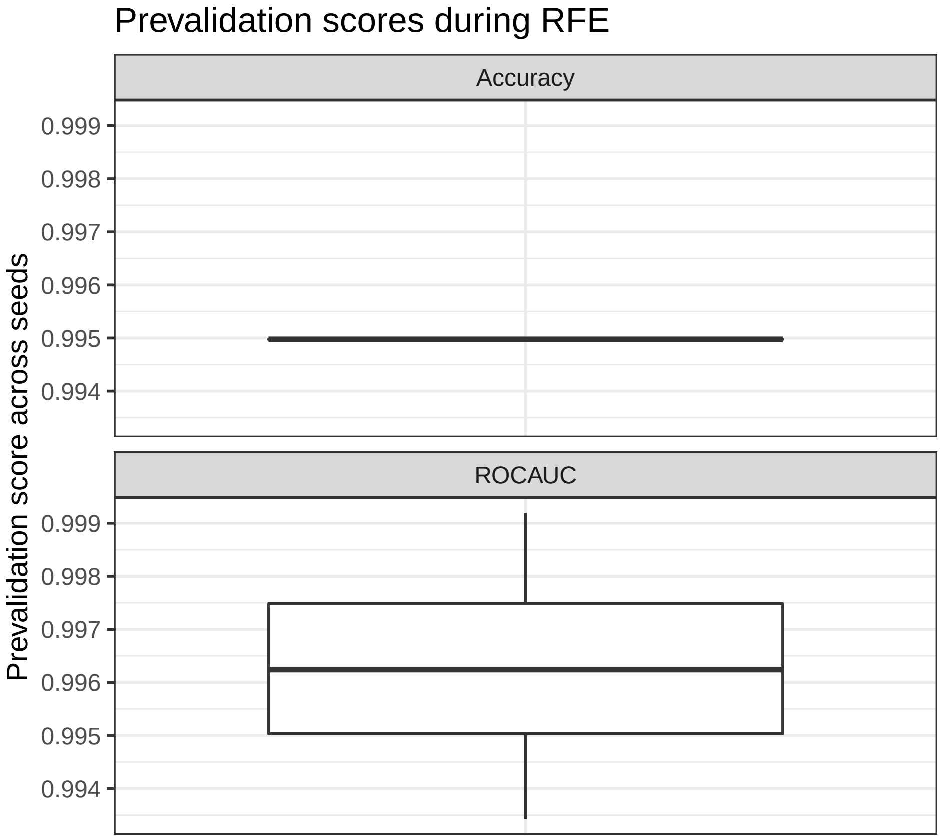


Confusion Matrix

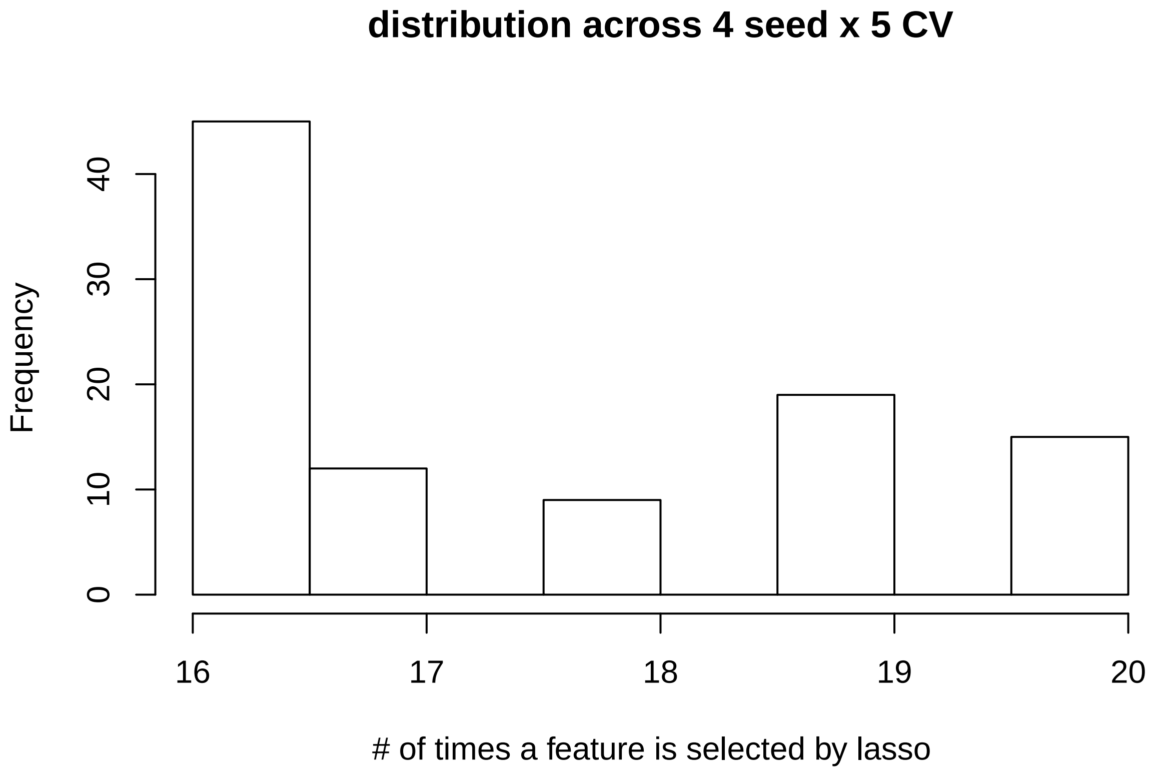
```
## confusion matrix at feature size = 100
## sum across 4 seeds

##           Reference
## Prediction   0   1
##          0 200   4
##          1   0 592
```

1.2 Average score



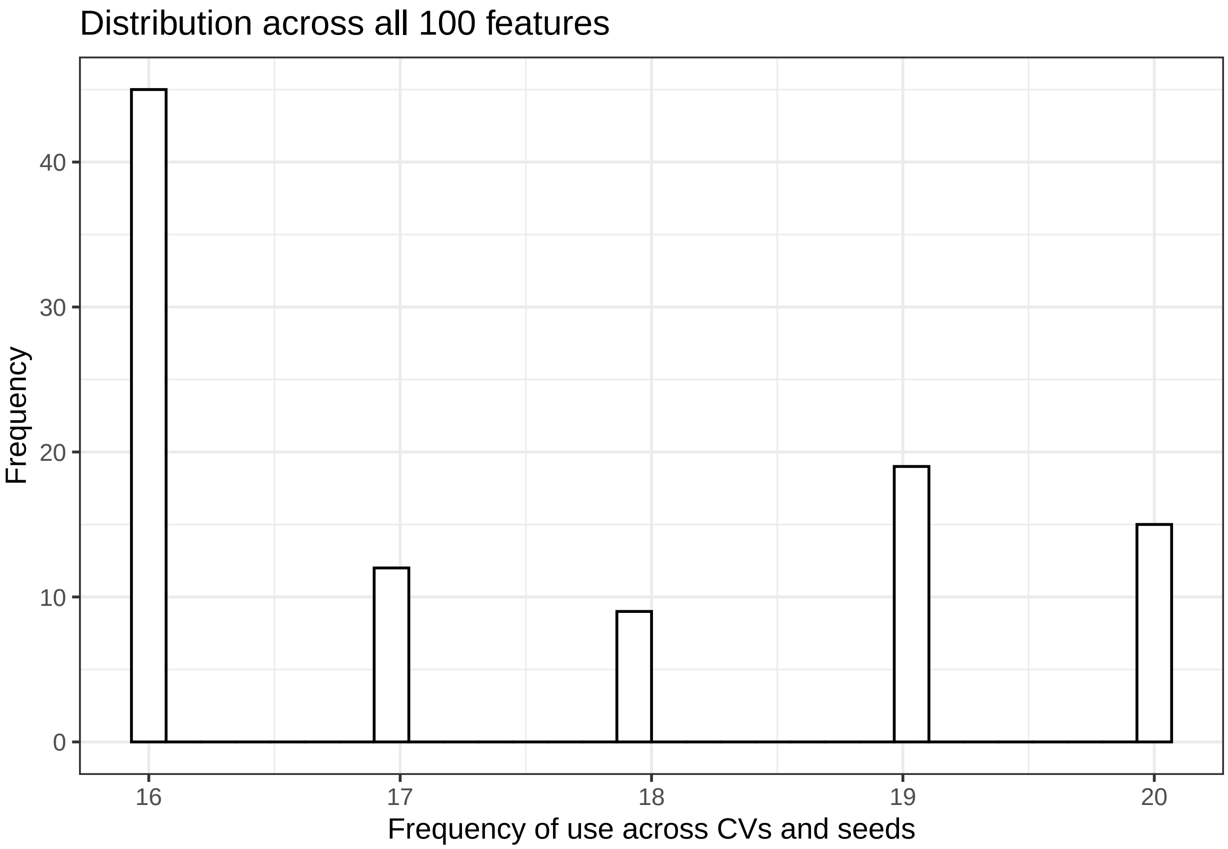
2. Important Features

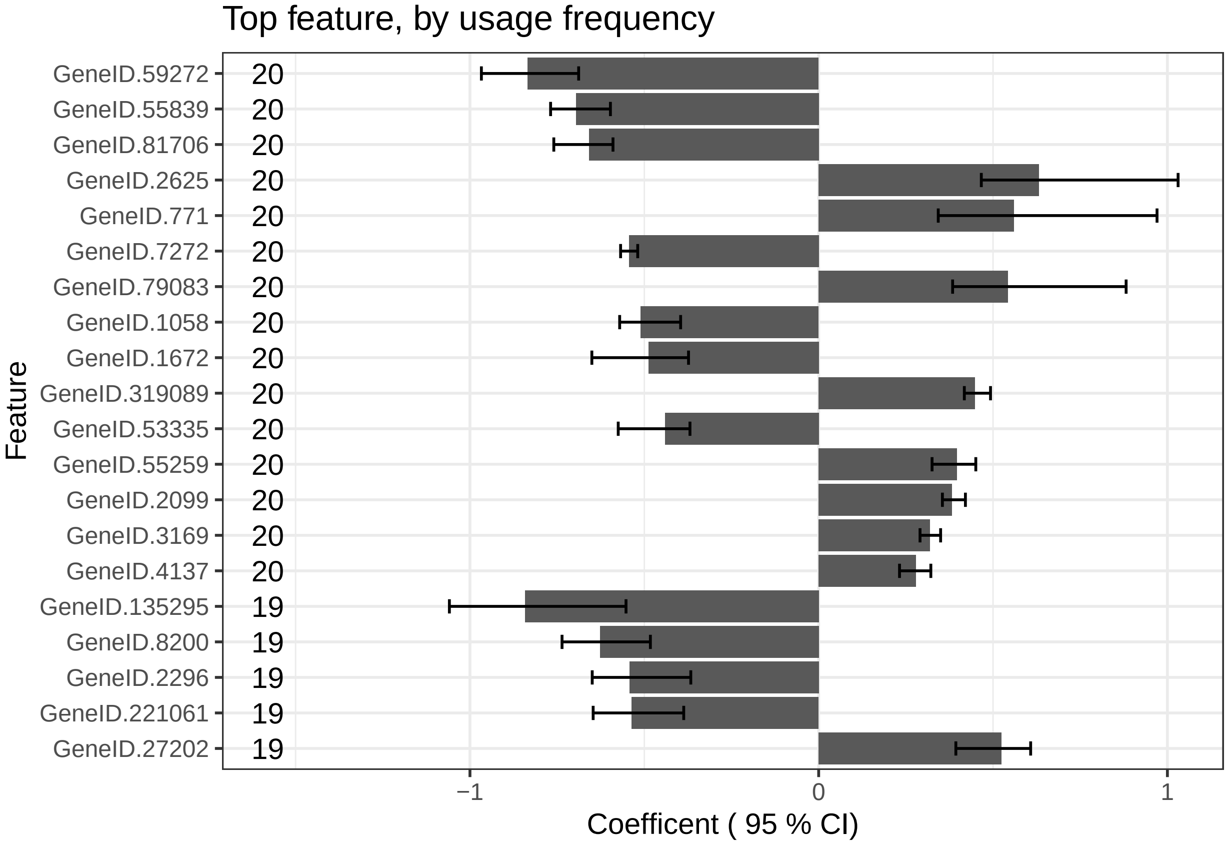


```
## [1] "there are 100 unique features used from the 100 feature set"
## [1] "summary of numer of features used in 4 seeds and 5 CVs"

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   16.00  100.00  100.00   87.35  100.00  100.00

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


Distribution across all 100 features


Top feature, by usage frequency

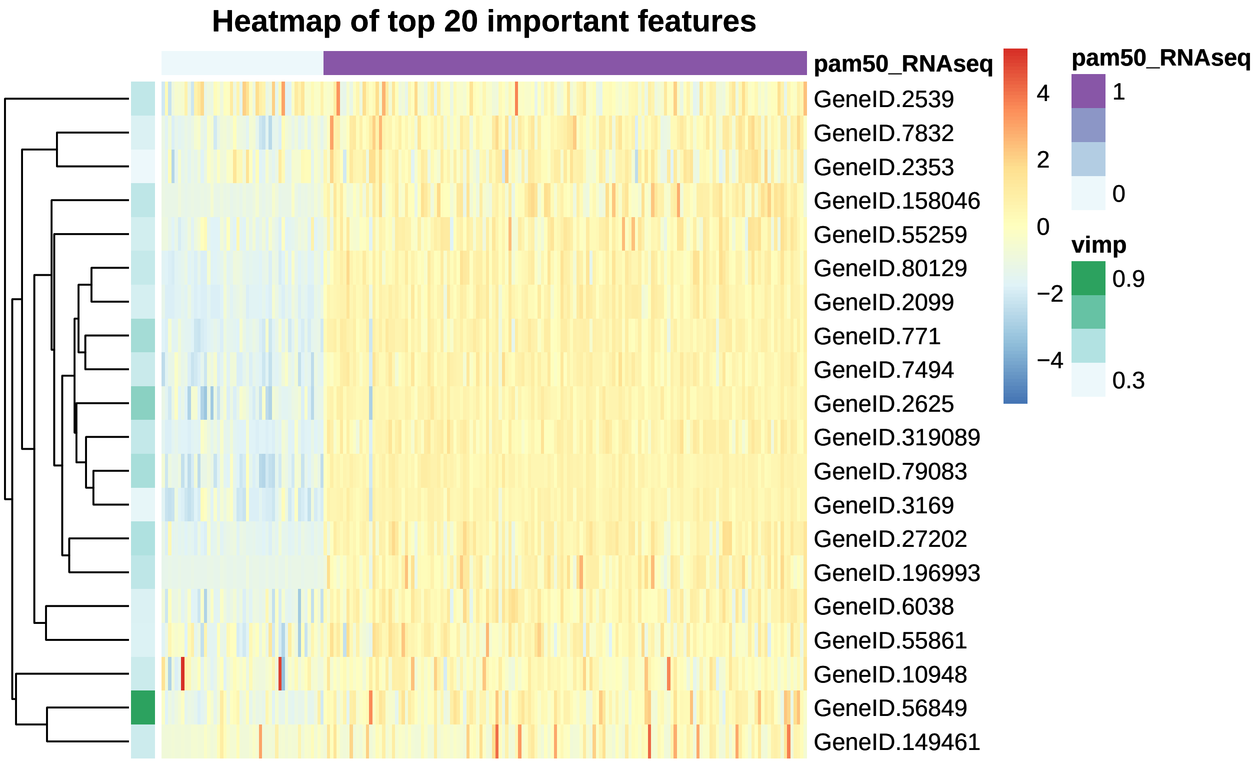


Table 1: parameter selection

seed	alpha	lambda	cv
1001	0.0	0.0390268	1
1001	0.0	0.0393106	2
1001	0.0	0.0387182	3
1001	0.5	0.0005100	4
1001	0.0	0.0388060	5
1002	0.0	0.0388340	1
1002	0.0	0.0387347	2
1002	0.5	0.0005194	3
1002	0.0	0.0388192	4
1002	0.0	0.0389460	5
1003	0.0	0.0392710	1
1003	0.5	0.0005150	2
1003	0.0	0.0387054	3
1003	0.0	0.0389664	4
1003	0.0	0.0389099	5
1004	0.0	0.0387564	1
1004	1.0	0.0003695	2
1004	0.0	0.0391371	3
1004	0.0	0.0389803	4
1004	0.0	0.0391363	5