# Evaluate testing data (multi-class) - Lasso

EVE W.

2020-04-19

## Contents



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

1. Lasso has its own inherent feature selection process.
2. Lasso vimp will be based on how many times the feature exist in all runs.

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

### 0. Load Data

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

```
## 300 of samples were used
```

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

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

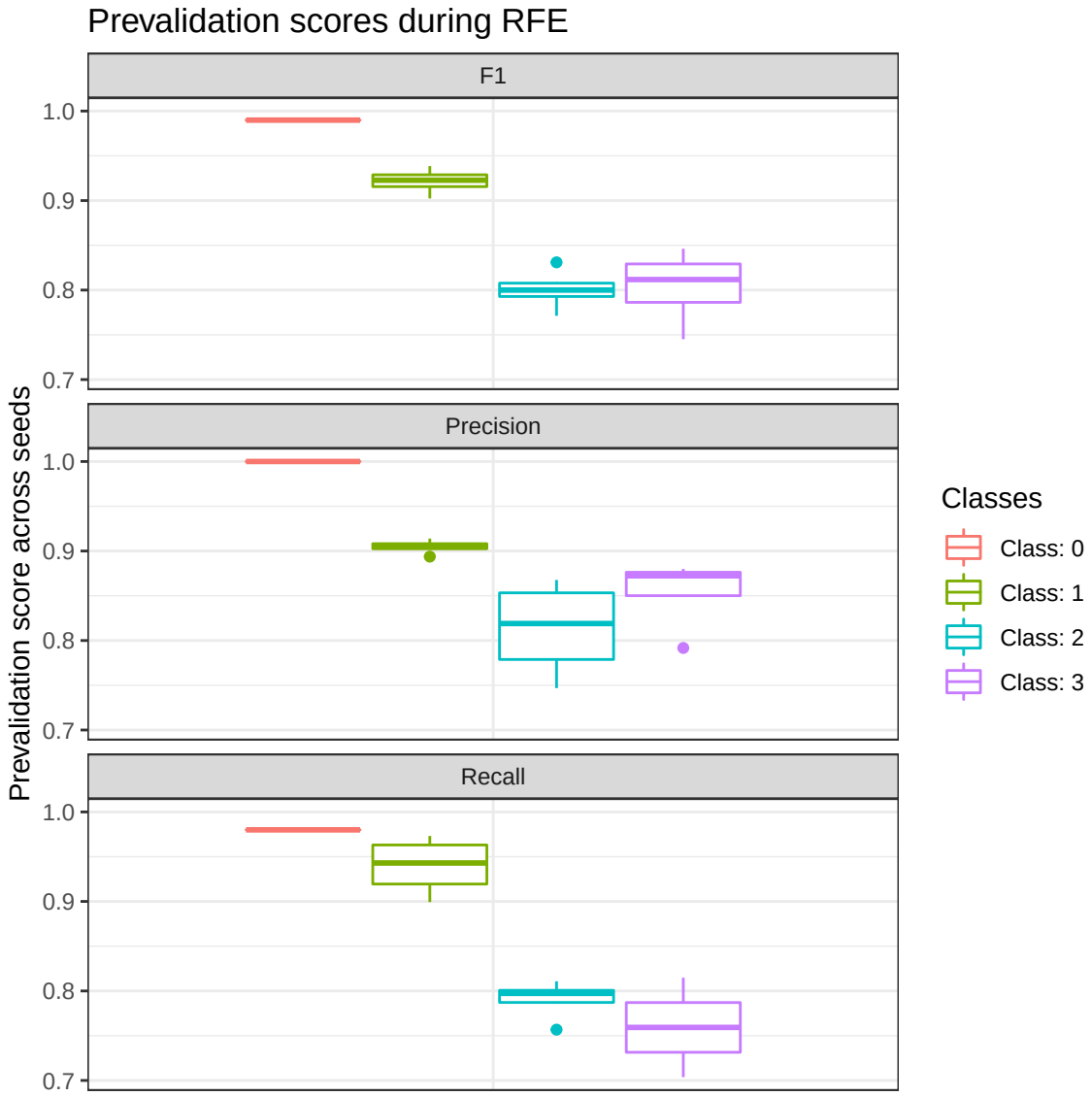
```
## Labels:
```

```
##
##   0   1   2   3
##  50 149  74  27
```

run with lasso.r with alpha = 0.5.

# 1. Scores

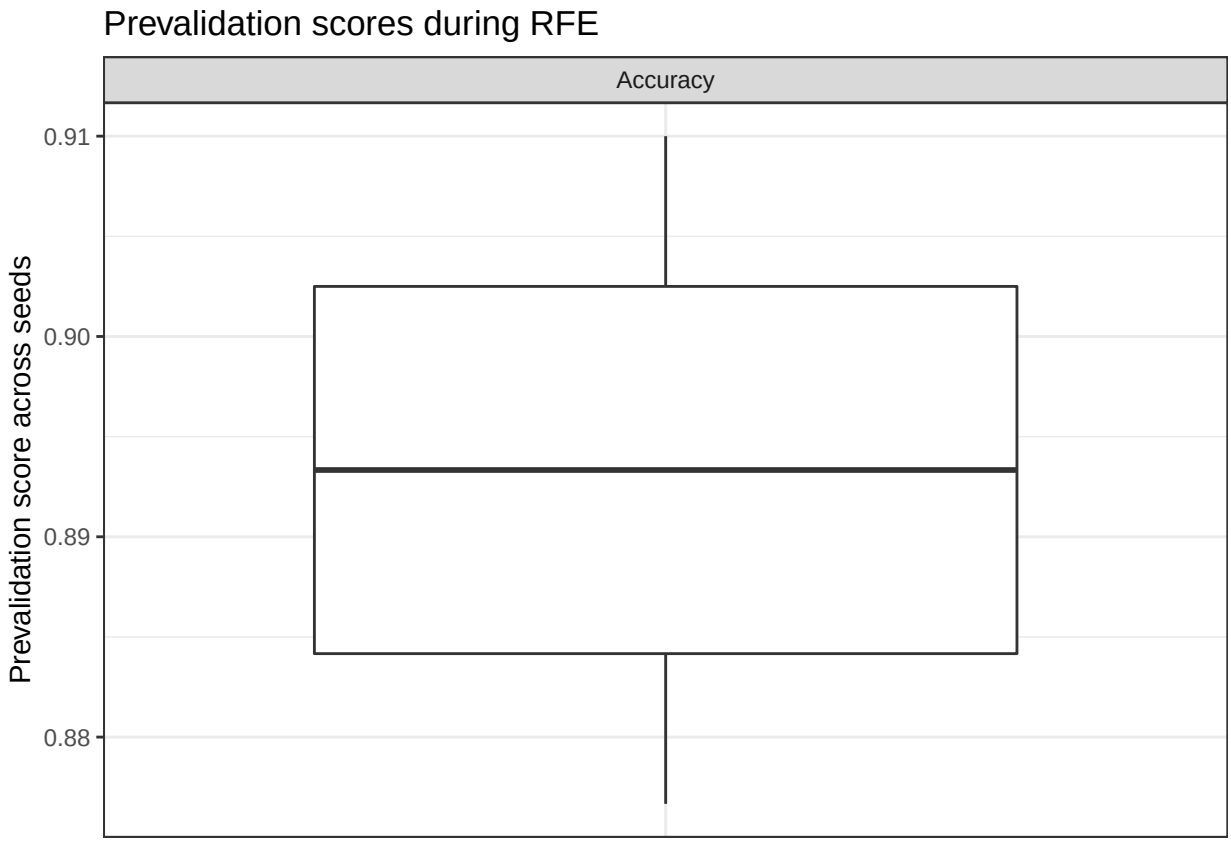
## 1.1 Scores per Class




Confusion Matrix

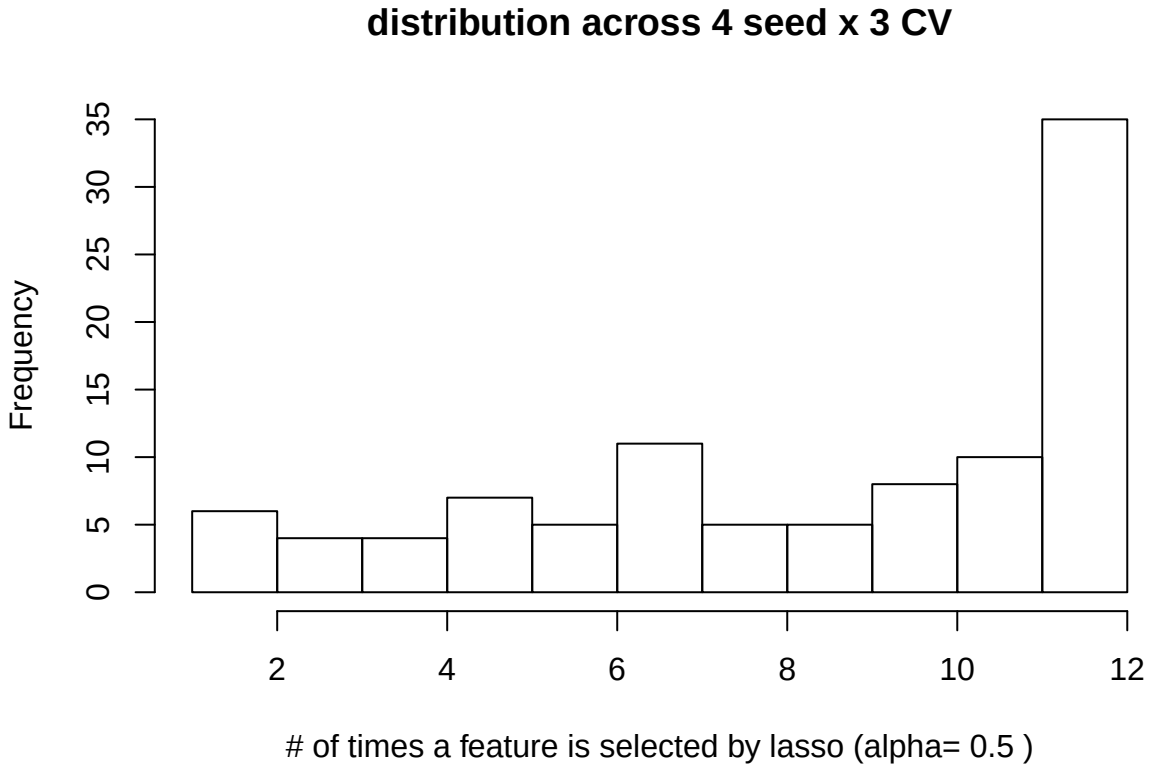
```
## confusion matrix at feature size = 100
## sum across 4 seeds

##            Reference
## Prediction   0   1   2   3
##          0 196   0   0   0
##          1   0 560  52   7
##          2   0  36 234  19
##          3   4   0  10  82
```

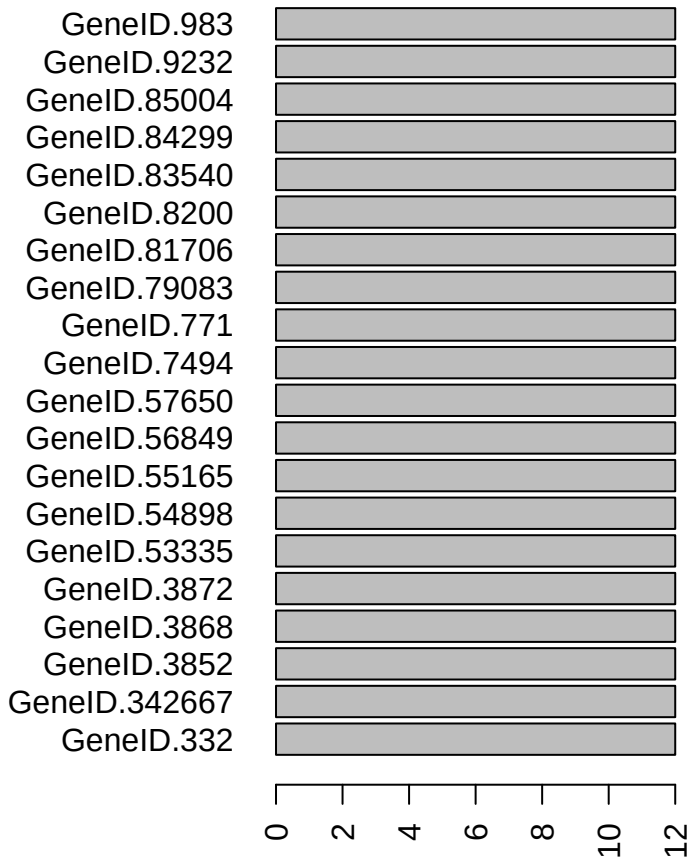
### 1.2 Average score




## 2. Important Features

For Lasso, we calculate how many times a given features is being used in all the runs.


distribution across 4 seed x 3 CV
Frequency
0
5
10
15
20
25
30
35
2
4
6
8
10
12
# of times a feature is selected by lasso (alpha= 0.5 )

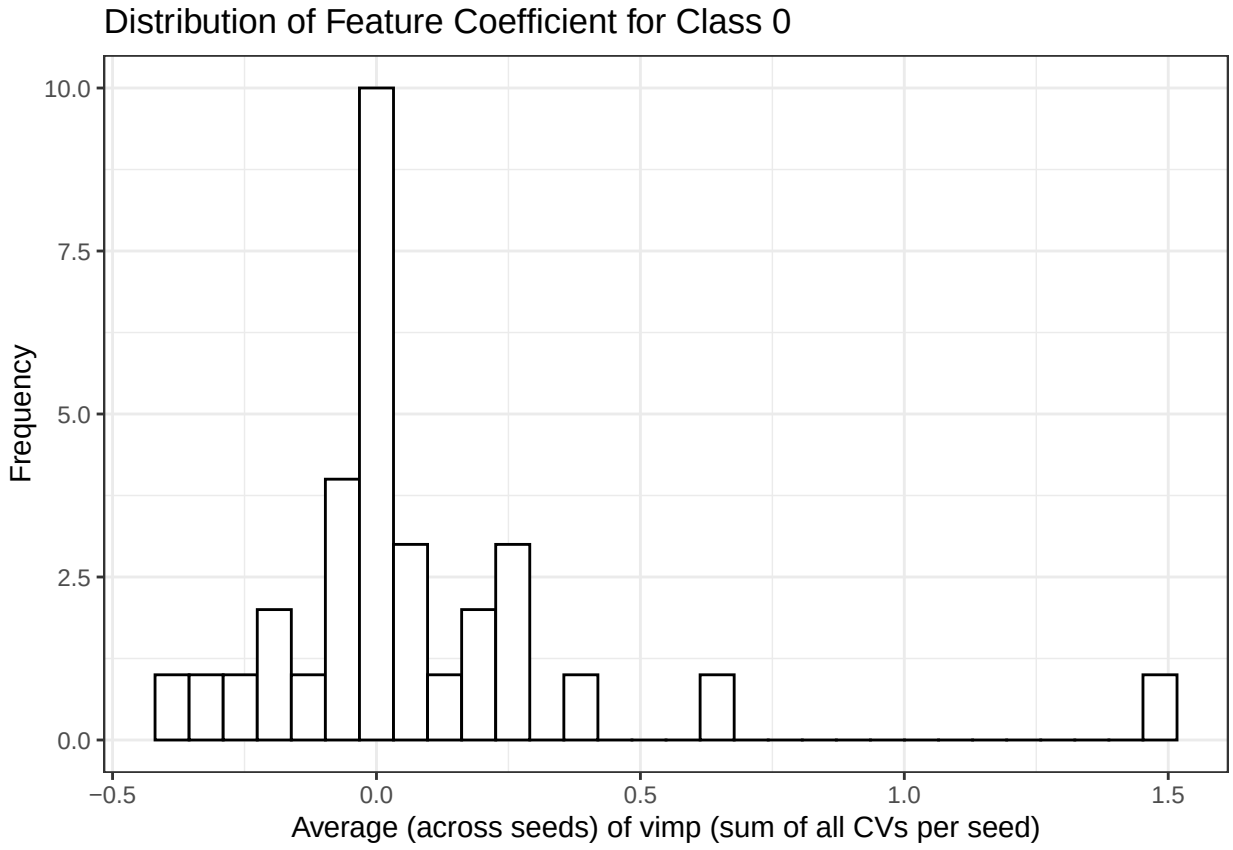


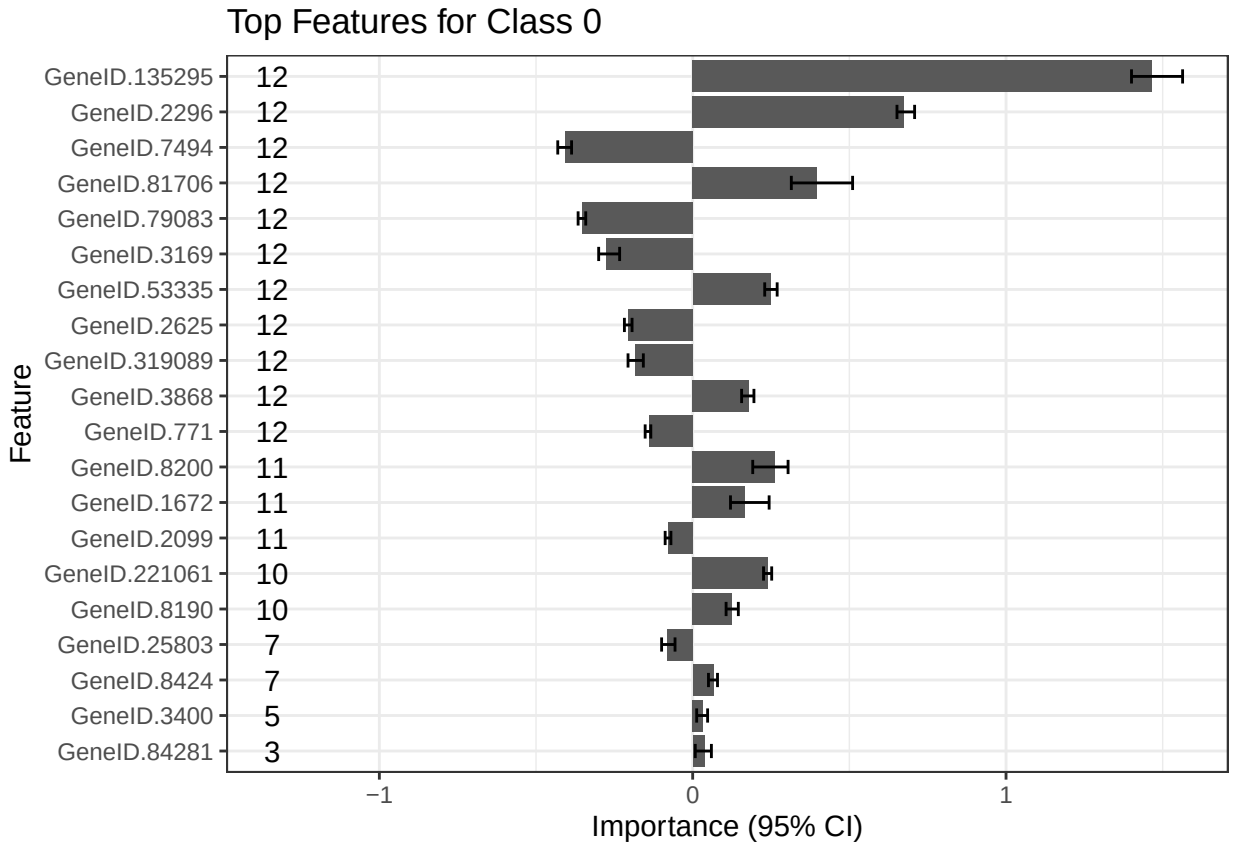

```
## (currently only Lasso has this graph)[1] "there are 100 unique features used from the 100 feature se
## [1] "summary of number of features used in each run under 4 seeds and 3 CVs"
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   68.00   70.75   72.50   73.00   76.00   77.00
```

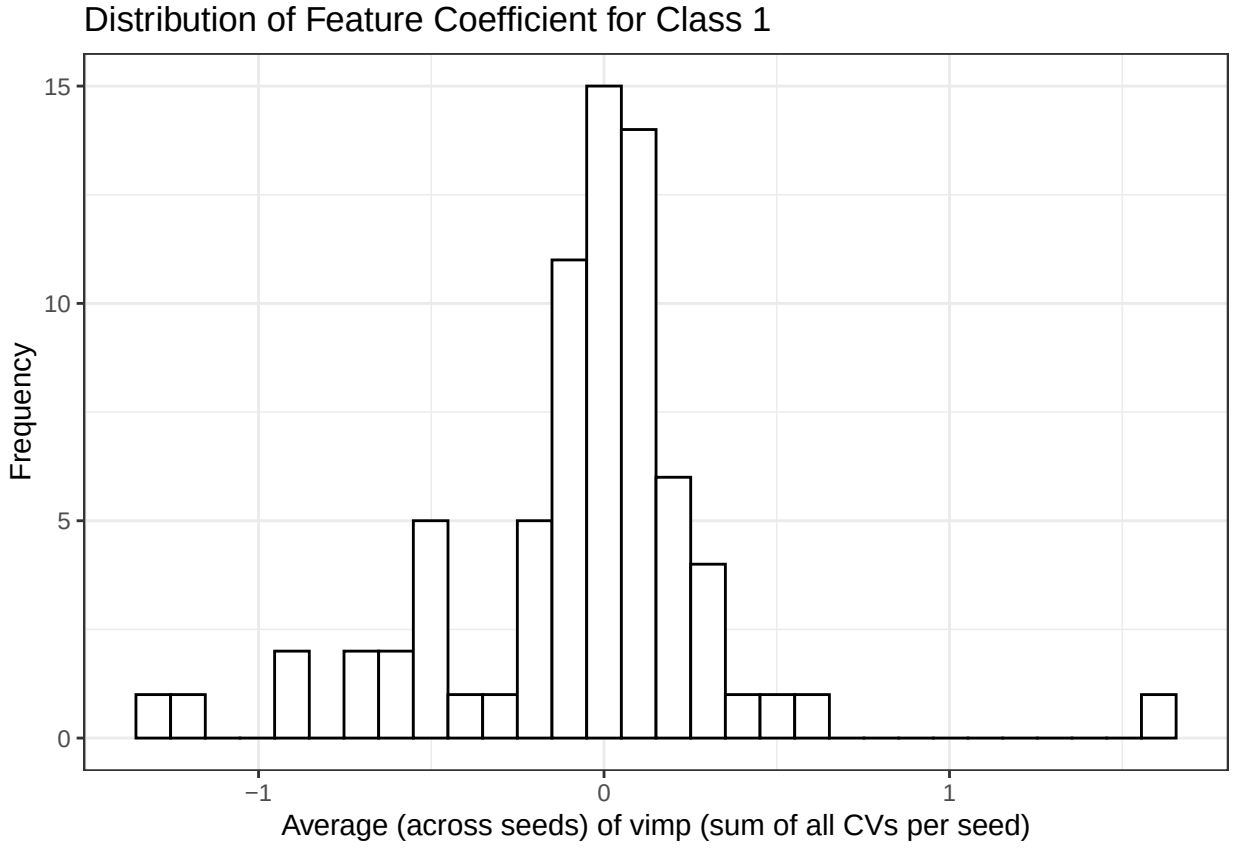
For regularized multinomial regression, glmnet does not use a reference level from the outcome variable and provides coefficients for each level of multinomial distribution. Please check out section **Regularized multinomial regression** section from here. The following barplots show the coefficients for each level.

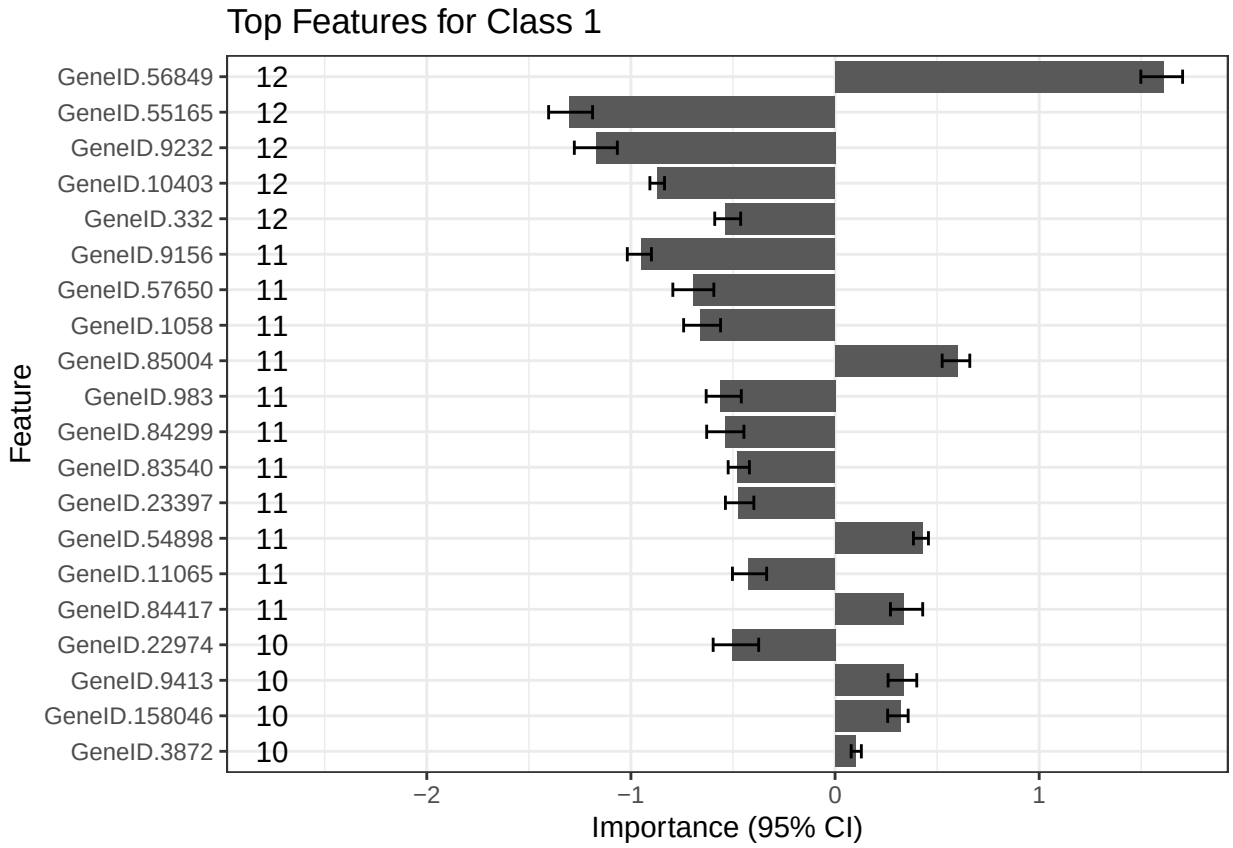
```
## [1] "removing 644 records as their vimp is less than 1e-06"
```


Distribution of Feature Coefficient for Class 0
Frequency
10.0
7.5
5.0
2.5
0.0
−0.5
0.0
0.5
1.0
1.5
Average (across seeds) of vimp (sum of all CVs per seed)

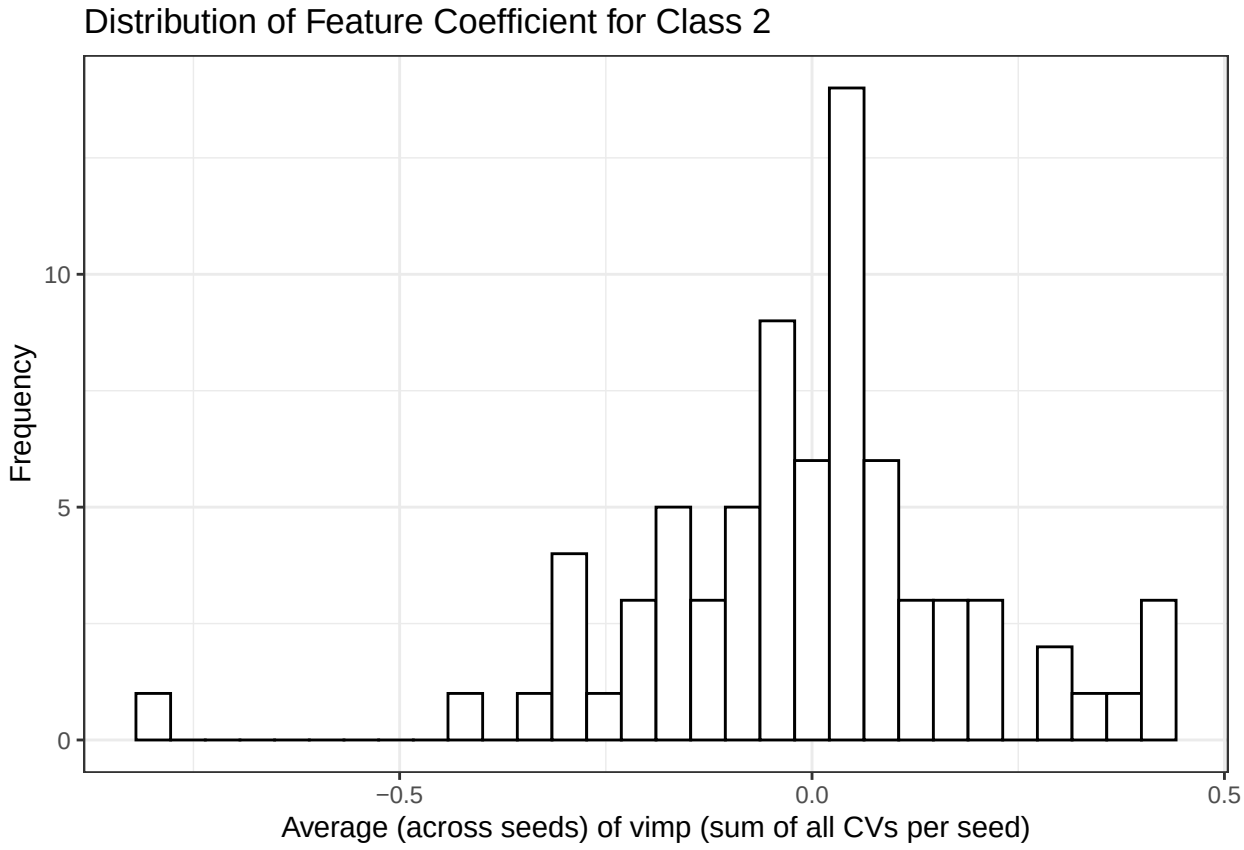

Top Features for Class 0
Feature
GeneID.135295 12
GeneID.2296 12
GeneID.7494 12
GeneID.81706 12
GeneID.79083 12
GeneID.3169 12
GeneID.53335 12
GeneID.2625 12
GeneID.319089 12
GeneID.3868 12
GeneID.771 12
GeneID.8200 11
GeneID.1672 11
GeneID.2099 11
GeneID.221061 10
GeneID.8190 10
GeneID.25803 7
GeneID.8424 7
GeneID.3400 5
GeneID.84281 3
−1
0
1
Importance (95% CI)

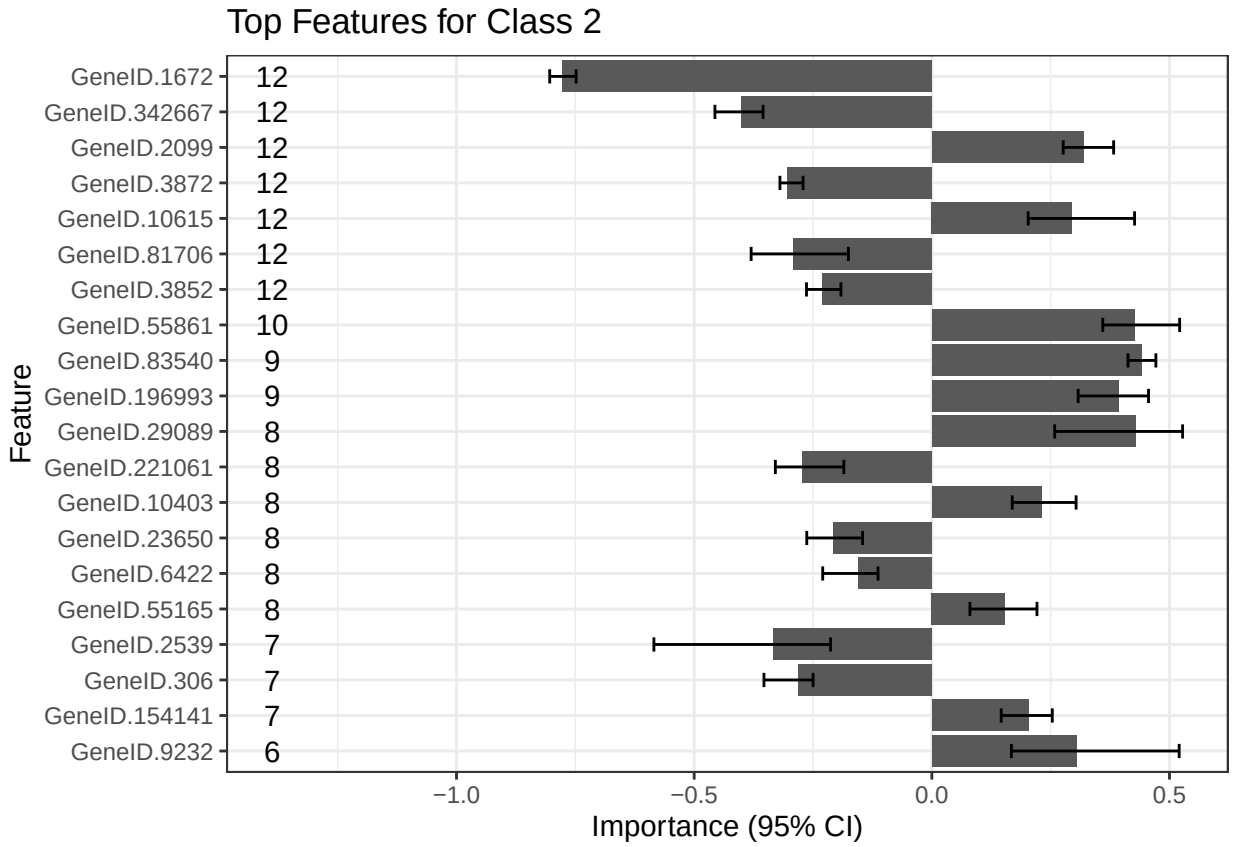
```
## [1] "removing 449 records as their vimp is less than 1e-06"
```



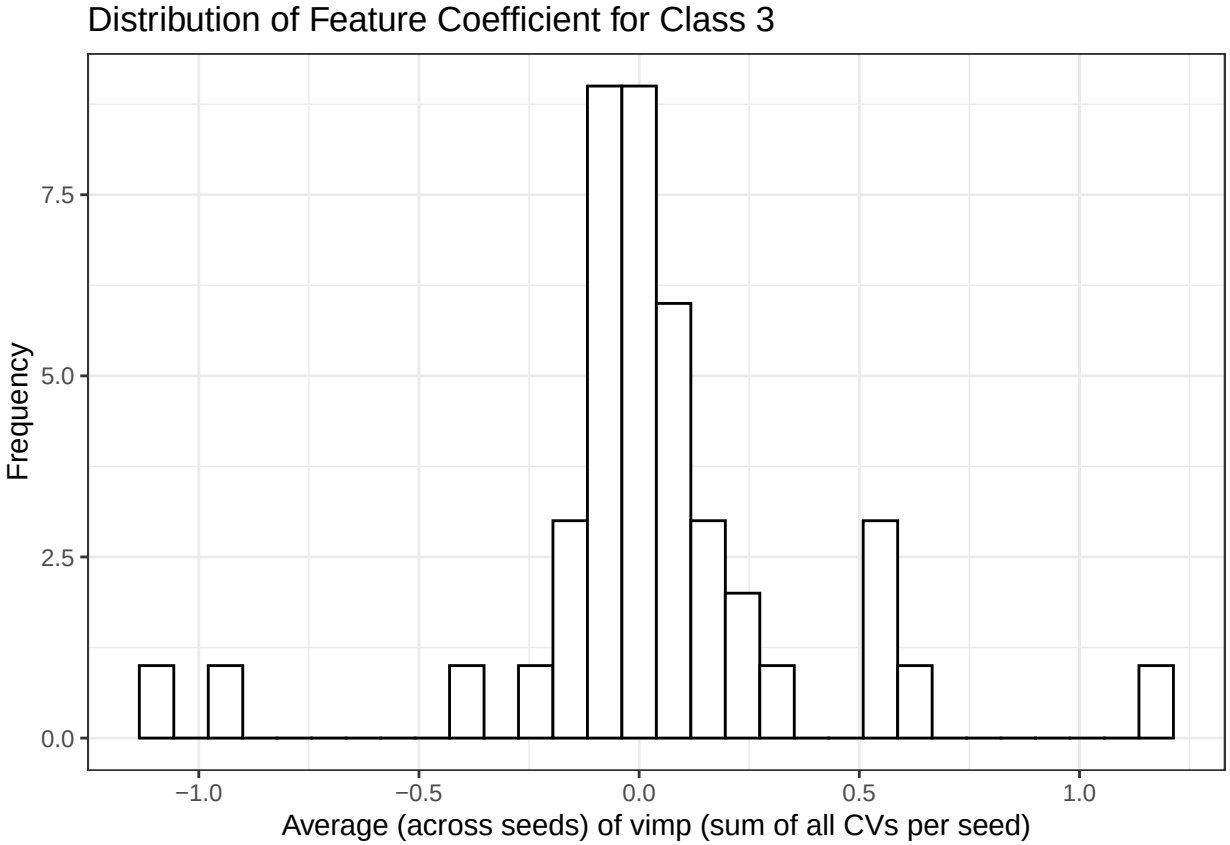


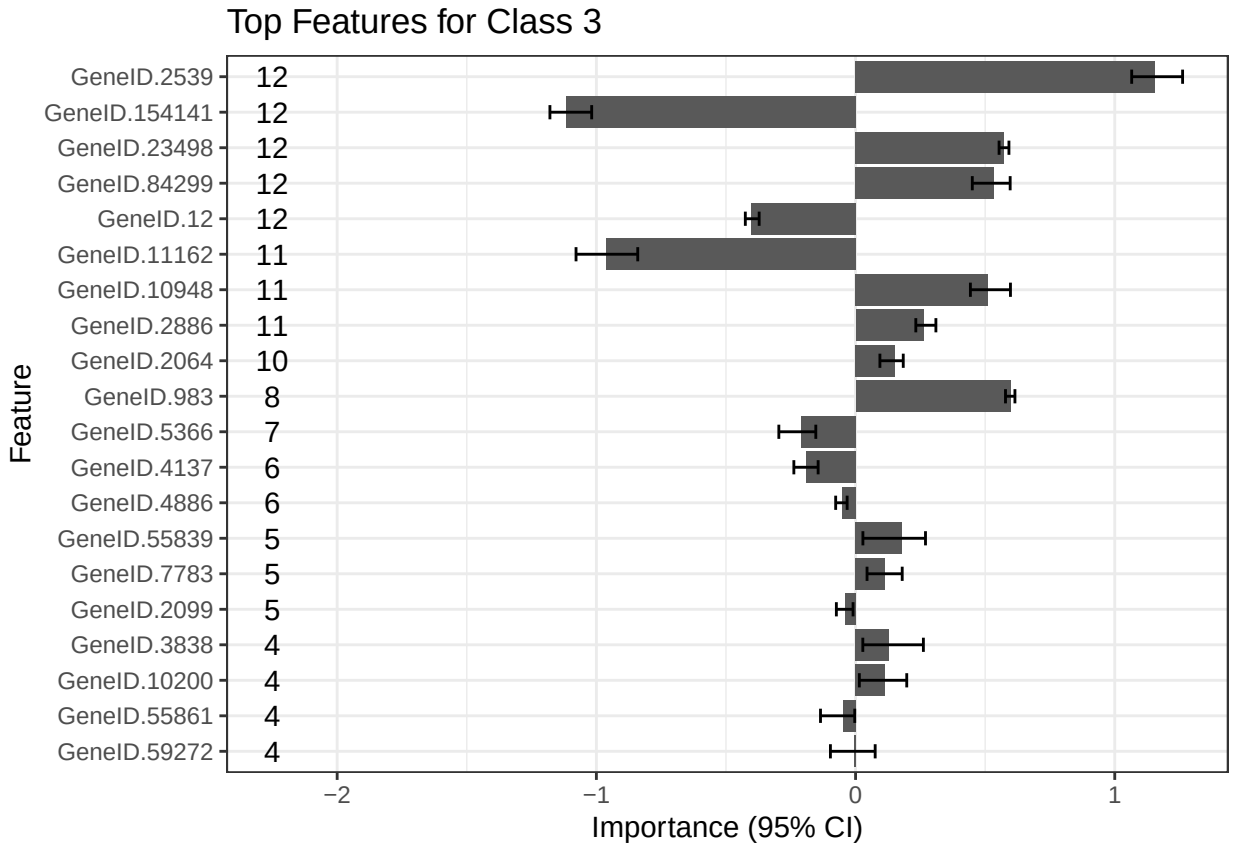

```
## [1] "removing 539 records as their vimp is less than 1e-06"
```


Distribution of Feature Coefficient for Class 2
Frequency
10
5
0
−0.5
0.0
0.5
Average (across seeds) of vimp (sum of all CVs per seed)


Top Features for Class 2
Feature
GeneID.1672 12
GeneID.342667 12
GeneID.2099 12
GeneID.3872 12
GeneID.10615 12
GeneID.81706 12
GeneID.3852 12
GeneID.55861 10
GeneID.83540 9
GeneID.196993 9
GeneID.29089 8
GeneID.221061 8
GeneID.10403 8
GeneID.23650 8
GeneID.6422 8
GeneID.55165 8
GeneID.2539 7
GeneID.306 7
GeneID.154141 7
GeneID.9232 6
−1.0
−0.5
0.0
0.5
Importance (95% CI)

```
## [1] "removing 684 records as their vimp is less than 1e-06"
```




Top Features for Class 3
Feature
GeneID.2539
GeneID.154141
GeneID.23498
GeneID.84299
GeneID.12
GeneID.11162
GeneID.10948
GeneID.2886
GeneID.2064
GeneID.983
GeneID.5366
GeneID.4137
GeneID.4886
GeneID.55839
GeneID.7783
GeneID.2099
GeneID.3838
GeneID.10200
GeneID.55861
GeneID.59272
12
12
12
12
12
11
11
11
10
8
7
6
6
5
5
5
4
4
4
4
−2
−1
0
1
Importance (95% CI)