SVM Classification

Code

Importing Data

Data is already split into train and test Columns consist of a category of land, which is the target, and various normalized difference vegetation indexes over time, which are the predictors. These values were determined via satellite imagery, and the vegetation indexes were taken from different times.

Hide

```
library(performanceEstimation)
library(e1071)

train <- read.csv("C:/Users/jah200003/Downloads/training.csv", header=TRUE, stringsAsFactors=TRUE)
test <- read.csv("C:/Users/jah200003/Downloads/testing.csv", header=TRUE, stringsAsFactors=TRUE)
```

Data Cleaning

Data is already prepared fairly well, but the classes are very imbalanced Applied synthetic oversampling and undersampling to improve balance before SVM

Hide

```
table(train$class)
```

```
     farm    forest     grass impervious    orchard     water 
     1441      7431       446       969        53       205 
```

Hide

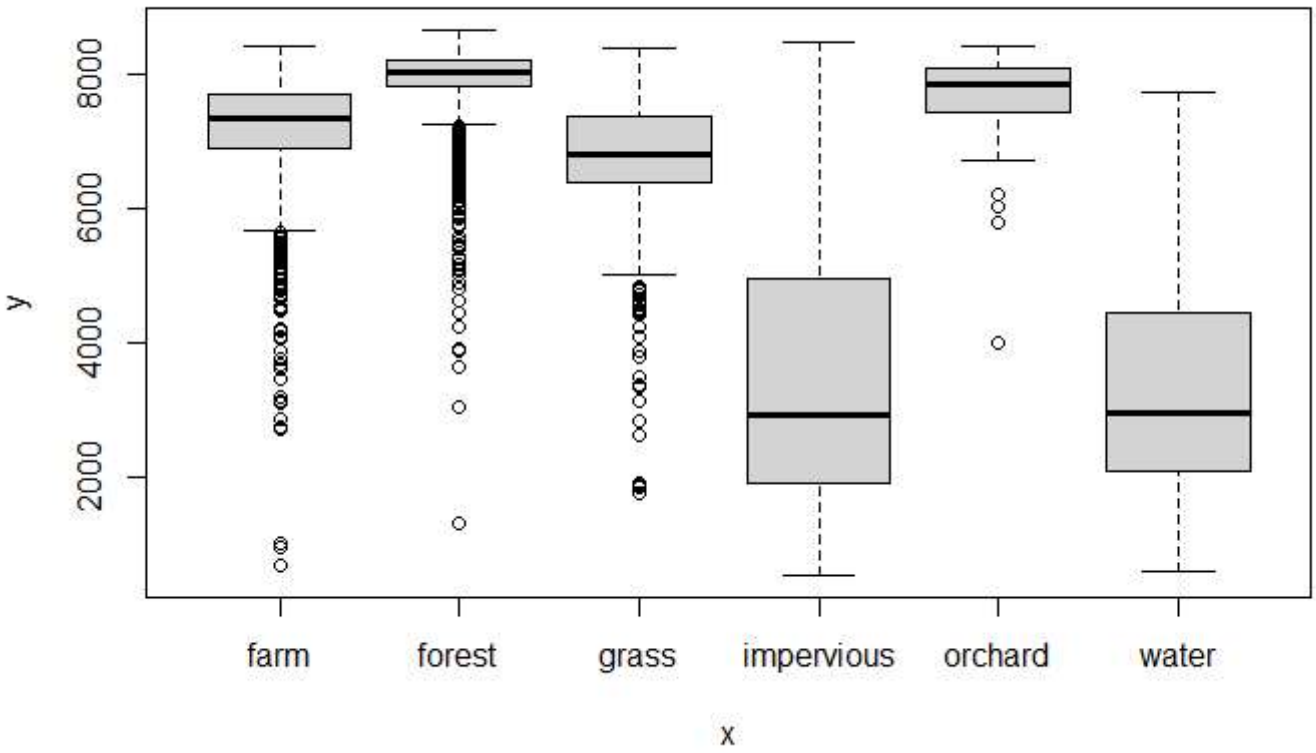
```
over <- smote(class~ ., train, perc.over = 50, perc.under=5)
table(over$class)
```

```
     farm    forest     grass impervious    orchard     water 
     1823      9385       566      1234      2703       242 
```

The balance could still be better, but this is certainly an improvement over the original training data.

Hide

```
plot(train$class, train$max_ndvi)
```



Hide

```
head(train)
```

class <fctr>	max_n... <dbl>	X20150720_N <dbl>	X20150602_N <dbl>	X20150517_N <dbl>	X20150501_N <dbl>	X20150415_N <dbl>	X20150330_N <dbl>
1 water	997.904	637.5950	658.668	-1882.030	-1924.36	997.904	-1739.990
2 water	914.198	634.2400	593.705	-1625.790	-1672.32	914.198	-692.386
3 water	3800.810	1671.3400	1206.880	449.735	1071.21	546.371	1077.840
4 water	952.178	58.0174	-1599.160	210.714	-1052.63	578.807	-1564.630
5 water	1232.120	72.5180	-1220.880	380.436	-1256.93	515.805	-1413.180
6 forest	7091.960	5102.9000	6996.710	201.956	6130.95	6439.300	6818.670

6 rows | 1-9 of 29 columns

Hide

```
summary(train)
```

```
       class         max_ndvi       X20150720_N      X20150602_N     X20150517_N     X20150501_N
 farm      :1441   Min.   : 563.4   Min.   :-433.7   Min.   :-1782   Min.   :-2940   Min.   :-3537
 forest    :7431   1st Qu.:7285.3   1st Qu.:4027.6   1st Qu.: 2061   1st Qu.: 1447   1st Qu.: 2984
 grass     : 446   Median :7886.3   Median :6737.7   Median : 5270   Median : 4394   Median : 5584
 impervious: 969   Mean   :7282.7   Mean   :5713.8   Mean   : 4777   Mean   : 4353   Mean   : 5077
 orchard   :  53   3rd Qu.:8121.8   3rd Qu.:7589.0   3rd Qu.: 7484   3rd Qu.: 7318   3rd Qu.: 7440
 water     : 205   Max.   :8650.5   Max.   :8377.7   Max.   : 8566   Max.   : 8650   Max.   : 8516
  X20150415_N       X20150330_N     X20150314_N     X20150226_N     X20150210_N     X20150125_N
 Min.   :-1815.6   Min.   :-5992   Min.   :-1678   Min.   :-2625   Min.   :-3403   Min.   :-3024
 1st Qu.:  526.9   1st Qu.: 2456   1st Qu.: 1018   1st Qu.: 2322   1st Qu.: 1379   1st Qu.: 2392
 Median : 1585.0   Median : 5638   Median : 2873   Median : 5673   Median : 4279   Median : 6262
 Mean   : 2871.4   Mean   : 4898   Mean   : 3338   Mean   : 4903   Mean   : 4249   Mean   : 5095
 3rd Qu.: 5460.1   3rd Qu.: 7245   3rd Qu.: 5517   3rd Qu.: 7396   3rd Qu.: 7144   3rd Qu.: 7546
 Max.   : 8267.1   Max.   : 8499   Max.   : 8002   Max.   : 8452   Max.   : 8422   Max.   : 8401
  X20150109_N       X20141117_N     X20141101_N       X20141016_N       X20140930_N      X20140813_N
 Min.   :-4505.7   Min.   :-1571   Min.   :-3305.1   Min.   :-1634.0   Min.   :-483.0   Min.   :-1137.2
 1st Qu.:  559.9   1st Qu.: 1069   1st Qu.:  616.8   1st Qu.:  947.8   1st Qu.: 513.2   1st Qu.:  718.1
 Median : 1157.2   Median : 2278   Median : 1770.3   Median : 1601.0   Median :1210.2   Median : 1260.3
 Mean   : 2141.9   Mean   : 3255   Mean   : 2628.1   Mean   : 2780.8   Mean   :2397.2   Mean   : 1548.2
 3rd Qu.: 3007.0   3rd Qu.: 5291   3rd Qu.: 4514.0   3rd Qu.: 4066.9   3rd Qu.:3963.6   3rd Qu.: 1994.9
 Max.   : 8477.6   Max.   : 8625   Max.   : 7932.7   Max.   : 8630.4   Max.   :8210.2   Max.   : 5915.7
  X20140626_N      X20140610_N     X20140525_N     X20140509_N     X20140423_N     X20140407_N
 Min.   : 372.1   Min.   :-3766   Min.   :-1043   Min.   :-4869   Min.   :-1506   Min.   :-1445.4
 1st Qu.:1582.5   1st Qu.: 2004   1st Qu.: 1392   1st Qu.: 1405   1st Qu.: 1010   1st Qu.:  429.9
 Median :2779.6   Median : 5267   Median : 3597   Median : 2671   Median : 2619   Median : 1245.9
 Mean   :3015.6   Mean   : 4787   Mean   : 3640   Mean   : 3027   Mean   : 3022   Mean   : 2041.6
 3rd Qu.:4255.6   3rd Qu.: 7549   3rd Qu.: 5818   3rd Qu.: 4174   3rd Qu.: 4838   3rd Qu.: 3016.5
 Max.   :7492.2   Max.   : 8490   Max.   : 7982   Max.   : 8445   Max.   : 7919   Max.   : 8206.8
  X20140322_N       X20140218_N      X20140202_N     X20140117_N       X20140101_N
 Min.   :-4354.6   Min.   :-232.3   Min.   :-6808   Min.   :-2139.9   Min.   :-4145.2
 1st Qu.:  766.5   1st Qu.: 494.9   1st Qu.: 5647   1st Qu.:  689.9   1st Qu.:  685.7
 Median : 1511.2   Median : 931.7   Median : 6862   Median : 1506.6   Median : 1458.9
 Mean   : 2691.6   Mean   :2058.3   Mean   : 6109   Mean   : 2563.5   Mean   : 2558.9
 3rd Qu.: 4508.5   3rd Qu.:2950.9   3rd Qu.: 7378   3rd Qu.: 4208.7   3rd Qu.: 4112.6
 Max.   : 8235.4   Max.   :8247.6   Max.   : 8410   Max.   : 8418.2   Max.   : 8502.0
```

It is difficult to visualize many of the predictors, since they are all so similar. The most important predictor should be the maximum vegetation index over all dates included, and there seem to be significant differences between some of the classes. The data is fairly noisy, and there are plenty of outliers, especially for the overrepresented forest class.

Linear SVM

Hide

```
svml <- svm(class~ ., data = over, kernel = "linear", cost = 50, scale = TRUE)
```

```
WARNING: reaching max number of iterations
```

Hide

```
summary(svml)
```

```
Call:
svm(formula = class ~ ., data = over, kernel = "linear", cost = 50, scale = TRUE)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  linear
       cost:  50

Number of Support Vectors:  4284

 ( 1529 1194 386 409 118 648 )


Number of Classes:  6

Levels:
 farm forest grass impervious orchard water
```

Hide

```
predl <- predict(svml, newdata = test)
table(predl, test$class)
```

```
predl        farm forest grass impervious orchard water
  farm         35      5     6          1       3     1
  forest       14     55    13          1      19     4
  grass         1     13    14          3       0     1
  impervious    0      1     1         34       0    10
  orchard       3      4     1          1      25     0
  water         0      0     1          0       0    30
```

Hide

```
mean(predl == test$class)
```

```
[1] 0.6433333
```

Hide

```
svml <- svm(class~ ., data = over, kernel = "linear", cost = 100, scale = TRUE)
```

```
WARNING: reaching max number of iterations

WARNING: reaching max number of iterations

WARNING: reaching max number of iterations
```

Hide

```
summary(svml)
```

```
Call:
svm(formula = class ~ ., data = over, kernel = "linear", cost = 100, scale = TRUE)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  linear
       cost:  100

Number of Support Vectors:  4288

 ( 1533 1195 383 409 119 649 )


Number of Classes:  6

Levels:
 farm forest grass impervious orchard water
```

Hide

```
predl <- predict(svml, newdata = test)
table(predl, test$class)
```

```
predl        farm forest grass impervious orchard water
  farm         35      5     6          1       3     1
  forest       15     55    13          1      19     4
  grass         1     13    14          3       0     1
  impervious    0      1     1         34       0    10
  orchard       2      4     1          1      25     0
  water         0      0     1          0       0    30
```

Hide

```
mean(predl == test$class)
```

```
[1] 0.6433333
```

Incredibly, these two SVMs came out with the same rounded accuracy despite different hyperparameters. The second iteration used a higher cost to allow more error within the large data set. Ideally, the tune() function would be used instead of a haphazard assignment of hyperparameters, but that was avoided due to incredibly high runtimes and memory usage.

Polynomial SVM

Hide

```
svmp <- svm(class~ ., data = over, kernel = "polynomial", cost = 100, scale = TRUE, gamma = 1)
summary(svmp)
```

```
Call:
svm(formula = class ~ ., data = over, kernel = "polynomial", cost = 100, gamma = 1, scale = TRUE)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  polynomial
       cost:  100
     degree:  3
     coef.0:  0

Number of Support Vectors:  3089

 ( 1526 734 213 424 84 108 )


Number of Classes:  6

Levels:
 farm forest grass impervious orchard water
```

Hide

```
predp <- predict(svmp, newdata = test)
table(predp, test$class)
```

```
predp        farm forest grass impervious orchard water
  farm         45     12     8          3      14     0
  forest        7     35     8          1      12     5
  grass         0     25    15          4       3     3
  impervious    0      5     3         31       0     3
  orchard       1      1     1          1      18     0
  water         0      0     1          0       0    35
```

Hide

```
mean(predp == test$class)
```

```
[1] 0.5966667
```

Hide

```
svmp <- svm(class~ ., data = over, kernel = "polynomial", cost = 50, scale = TRUE, gamma = 0.001)
summary(svmp)
```

```
Call:
svm(formula = class ~ ., data = over, kernel = "polynomial", cost = 50, gamma = 0.001, scale = TRUE)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  polynomial
       cost:  50
     degree:  3
     coef.0:  0

Number of Support Vectors:  9934

 ( 3605 1823 566 995 242 2703 )


Number of Classes:  6

Levels:
 farm forest grass impervious orchard water
```

Hide

```
predp <- predict(svmp, newdata = test)
table(predp, test$class)
```

```
predp        farm forest grass impervious orchard water
  farm          0      0     0          0       0     0
  forest       53     78    36         23      47     9
  grass         0      0     0          0       0     0
  impervious    0      0     0         17       0    35
  orchard       0      0     0          0       0     0
  water         0      0     0          0       0     2
```

Hide

```
mean(predp == test$class)
```

```
[1] 0.3233333
```

Hide

```
svmp <- svm(class~ ., data = over, kernel = "polynomial", cost = 100, scale = TRUE)
summary(svmp)
```

```
Call:
svm(formula = class ~ ., data = over, kernel = "polynomial", cost = 100, scale = TRUE)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  polynomial
       cost:  100
     degree:  3
     coef.0:  0

Number of Support Vectors:  3089

 ( 1529 731 223 409 84 113 )


Number of Classes:  6

Levels:
 farm forest grass impervious orchard water
```

Hide

```
predp <- predict(svmp, newdata = test)
table(predp, test$class)
```

```
predp        farm forest grass impervious orchard water
  farm         45     12     8          2      14     0
  forest        7     35     8          1      12     5
  grass         0     26    15          1       3     3
  impervious    0      4     3         35       0     3
  orchard       1      1     1          1      18     0
  water         0      0     1          0       0    35
```

Hide

```
mean(predp == test$class)
```

```
[1] 0.61
```

I generally stuck with a cost of 100 because that was the maximum value in the suggested range for that hyperparameter, and the number of entries made me want to keep cost high to reduce overfitting, especially for polynomial SVMs. These SVMs also have a gamma hyperparameter that works in tandem with cost to adjust bias and variance. The best accuracy here was found with the default gamma value of 1/dim, which in this case would be about 0.035.

Radial SVM

Hide

```
svmr <- svm(class~ ., data = over, kernel = "radial", cost = 50, scale = TRUE, gamma = 1)
summary(svmr)
```

```
Call:
svm(formula = class ~ ., data = over, kernel = "radial", cost = 50, gamma = 1, scale = TRUE)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  50

Number of Support Vectors:  7922

 ( 5294 1057 337 612 150 472 )


Number of Classes:  6

Levels:
 farm forest grass impervious orchard water
```

Hide

```
predr <- predict(svmr, newdata = test)
table(predr, test$class)
```

```
predr        farm forest grass impervious orchard water
  farm          0      0     0          0       0     0
  forest       53     78    35         24      47    46
  grass         0      0     1          0       0     0
  impervious    0      0     0         16       0     0
  orchard       0      0     0          0       0     0
  water         0      0     0          0       0     0
```

Hide

```
mean(predr == test$class)
```

```
[1] 0.3166667
```

Hide

```
svmr <- svm(class~ ., data = over, kernel = "radial", cost = 10, scale = TRUE, gamma = 0.0001)
summary(svmr)
```

```
Call:
svm(formula = class ~ ., data = over, kernel = "radial", cost = 10, gamma = 1e-04, scale = TRUE)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  10

Number of Support Vectors:  7301

 ( 2753 1580 566 630 242 1530 )


Number of Classes:  6

Levels:
 farm forest grass impervious orchard water
```

Hide

```
predr <- predict(svmr, newdata = test)
table(predr, test$class)
```

```
predr        farm forest grass impervious orchard water
  farm         29      3     4          2       2     0
  forest       19     70    28          2      12     6
  grass         0      0     0          0       0     0
  impervious    0      1     1         36       0    38
  orchard       5      4     2          0      33     0
  water         0      0     1          0       0     2
```

Hide

```
mean(predr == test$class)
```

```
[1] 0.5666667
```

Hide

```
svmr <- svm(class~ ., data = over, kernel = "radial", cost = 25, scale = TRUE, gamma = 0.001)
summary(svmr)
```

```
Call:
svm(formula = class ~ ., data = over, kernel = "radial", cost = 25, gamma = 0.001, scale = TRUE)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  25

Number of Support Vectors:  4085

 ( 1359 1059 432 456 136 643 )


Number of Classes:  6

Levels:
 farm forest grass impervious orchard water
```

Hide

```
predr <- predict(svmr, newdata = test)
table(predr, test$class)
```

```
predr        farm forest grass impervious orchard water
  farm         35      5     9          0       1     2
  forest       15     58    10          1      15     3
  grass         0     10    13          2       1     1
  impervious    0      1     2         37       0     6
  orchard       3      4     1          0      30     0
  water         0      0     1          0       0    34
```

Hide

```
mean(predr == test$class)
```

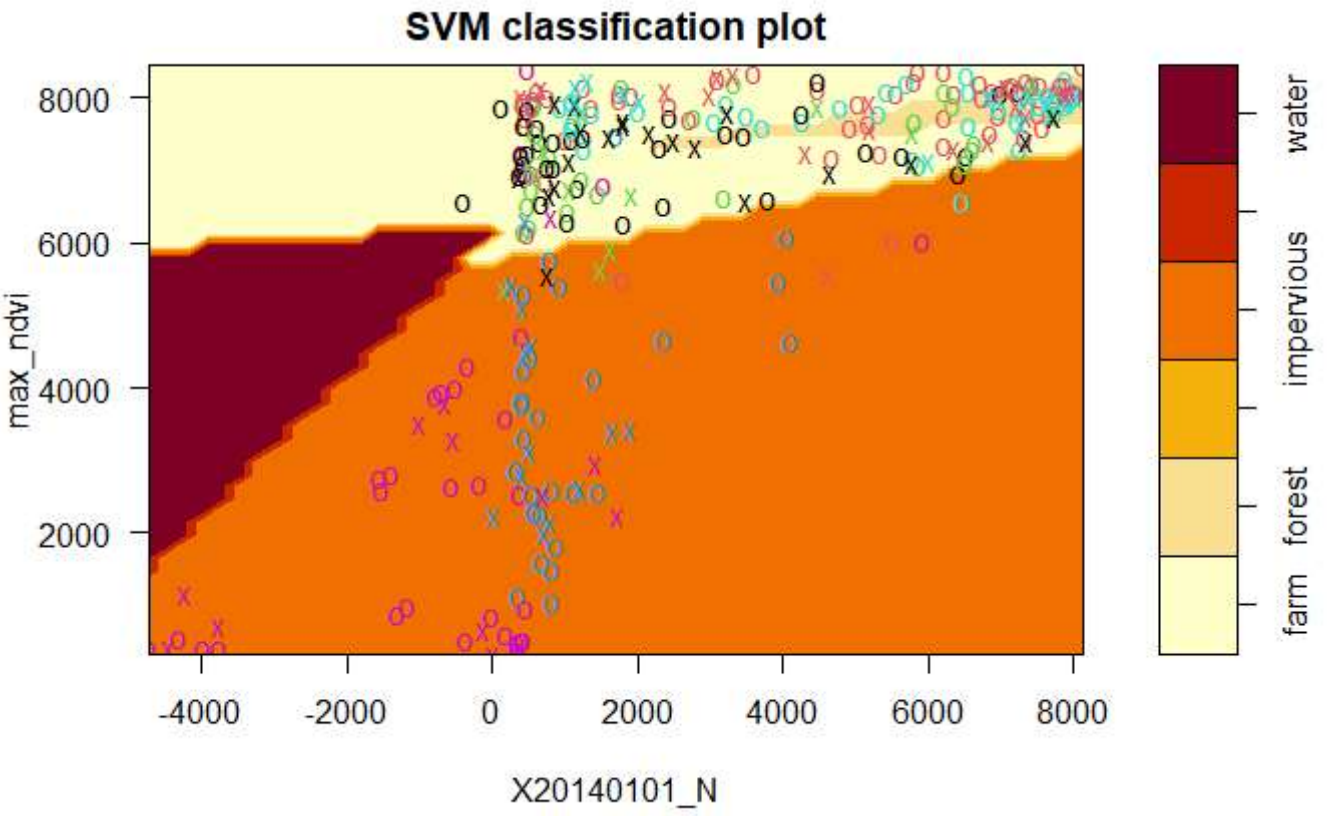
```
[1] 0.69
```

I found the most success with the radial kernel, which may just be the nature of the data. I actually decreased the cost here relative to my use of the other kernels, and it seemed to work well without overfitting. The low gamma hyperparameter may have helped with that, increasing bias but decreasing variance.

Visualization

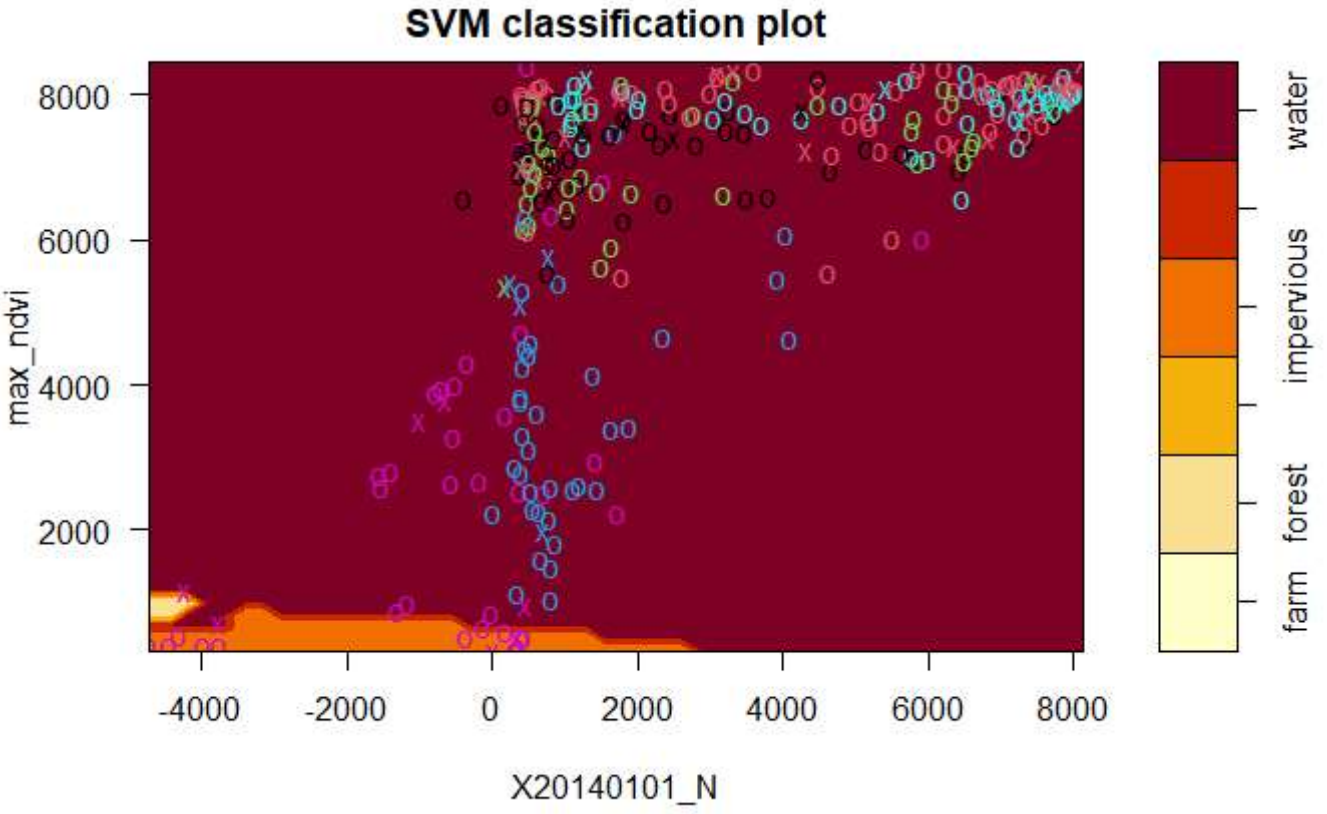
Hide

```
plot(svml, test, max_ndvi ~ X20140101_N)
```



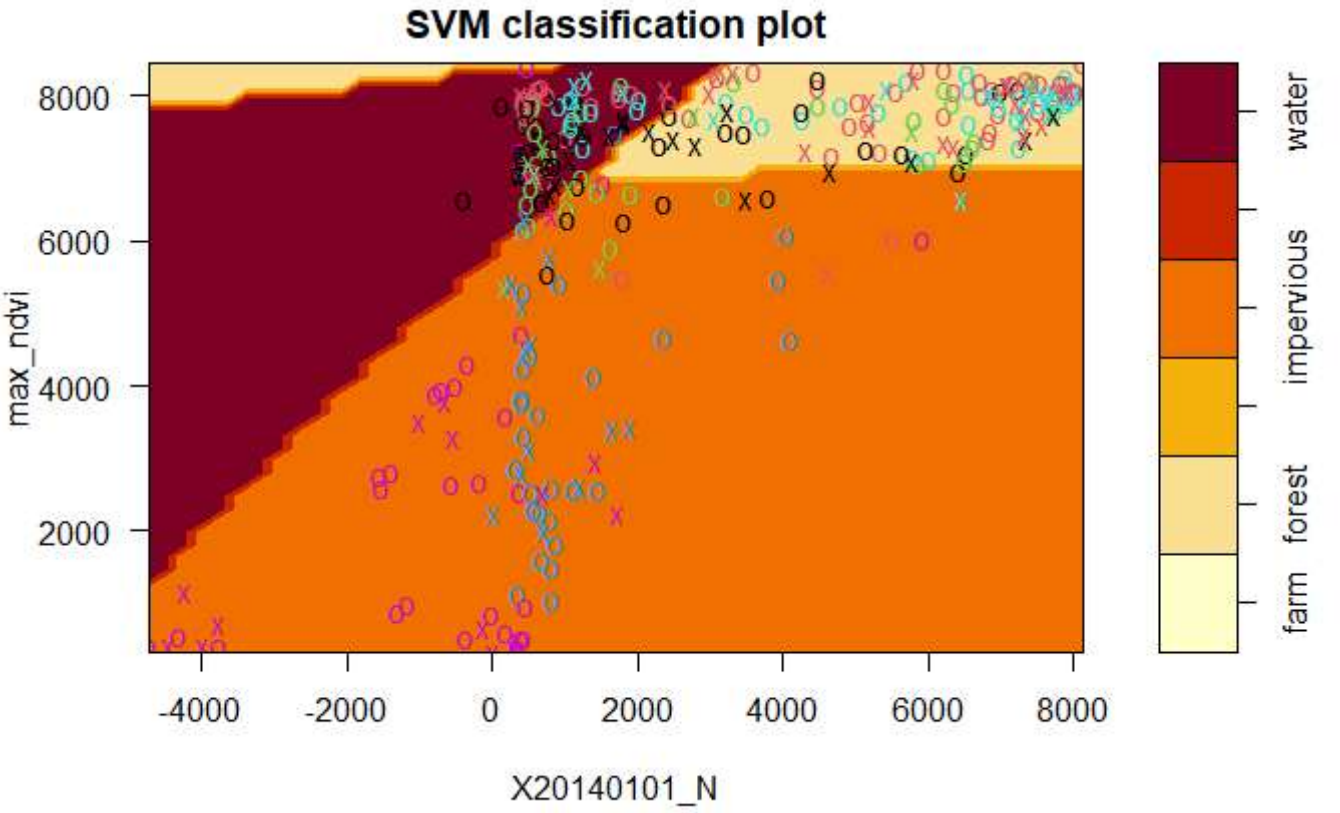
Hide

```
plot(svmp, test, max_ndvi ~ X20140101_N)
```



Hide

```
plot(svmr, test, max_ndvi ~ X20140101_N)
```



These plots are fun to look at but they are totally meaningless; the support vectors incorporate every predictor, whereas these plots only take the maximum and earliest vegetation indexes. Thus, these slices show only a small fraction of the relevant dimensions.