

# iHRDPredict

Navonil De Sarkar, Brian Hanratty

4/21/2021

## LOAD DEPENDENCY PACKAGES; INSTALL THEM IF THEY DON'T EXIST.

```
## Warning: package 'dplyr' was built under R version 4.0.4
```

```
## Warning: package 'MASS' was built under R version 4.0.4
```

```
## Warning: package 'e1071' was built under R version 4.0.5
```

```
## Warning: package 'ROCR' was built under R version 4.0.5
```

```
## Warning: package 'pROC' was built under R version 4.0.5
```

## LOAD DATA AND SET SEED

```
#load("iHRD_predict_prefitted_mCRPC_ws.RData")
data<-read_excel("C:/Users/bhanratt/Documents/GitHub/iHRD-Predict.R/iHRD-datafreeze-July
2020-training_2021Apr22.xlsx", sheet = 1, range=cell_cols("B:K"))
data$HRD_input_category <- as.factor(data$HRD_input_category)
set.seed(1000)
```

## BUILD TRAINING/TEST/VALIDATION DATA

```
train.data <- droplevels(data[data$HRD_input_category%in%c(0, 1), ])
## FEATURES: LOH, Ploidy, Sig 3, Sig 8, Mut_burden, Number_of_segments
## OUTCOME: HRD_input_category/Validation_category
train.data.x <- train.data[, c(3,9, 7, 8, 6, 4)]
train.data.y <- train.data$HRD_input_category
## model test data (input your own test data here)
test.data <- data[data$HRD_input_category%in%c(2) & data$Validation_category!= '.', ]
test.data.x <- test.data[, c(3,9, 7, 8, 6, 4)]
```

## DRY RUN WITH FULL DATA

*## 3i. TUNING SPACE AND LOAD*

```
data<-read_excel("C:/Users/bhanratt/Documents/GitHub/iHRD-Predict.R/iHRD-datafreeze-July
2020-Git-sample-test-file.from1.xlsx", sheet = 1, range=cell_cols("B:K"))
costvec = 10^(-3:1)
gammavec = c(0.00001, 0.0001, 0.001, 0.01, 0.1, 0.15, 0.2, 0.5, 1, 2, 3, 4, 5)
## 3ii. CV TUNING
tune.obj.nl <- tune.svm(HRD_input_category ~ LOH + Mut_burden + Sig3 + Sig8 + Number_of_
segments + Ploidy, data = train.data,
                      kernel = 'radial', cost = costvec, gamma = gammavec, tunecontrol
= tune.control(nrepeat = 20, cross = 10))
tune.obj.nl
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
## gamma cost
## 0.1 10
##
## - best performance: 0.04166667
```

*## 3iii. RUN FULL MODEL*

```
svm.fit.nl <- svm(HRD_input_category ~ LOH + Mut_burden + Sig3 + Sig8 + Number_of_segmen
ts + Ploidy, data = train.data,
                 cost = tune.obj.nl$best.parameters$cost, gamma = tune.obj.nl$best.para
meters$gamma, kernel='radial', probability = T)
## 3iv. TRAINING ERROR
pred.train <- predict(svm.fit.nl, train.data.x, decision.values = T)
pred.train
```

```

## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
## 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1
## 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
## 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0
## 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60
## 1 1 0 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
## 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
## 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100
## 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120
## 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140
## 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160
## 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180
## 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200
## 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220
## 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238
## 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## attr("decision.values")
## 1/0
## 1 3.00346629
## 2 0.99955169
## 3 2.05920597
## 4 2.37571141
## 5 2.93652392
## 6 1.66803133
## 7 0.99970383
## 8 0.99982091
## 9 2.14861987
## 10 0.01193011
## 11 1.62487093
## 12 1.40728993
## 13 2.58746876
## 14 -1.33488393
## 15 2.21511690
## 16 -0.91720549
## 17 1.00005444
## 18 0.99997425
## 19 2.03469479
## 20 2.94843828
## 21 1.00051083
## 22 1.00034728
## 23 1.58453557
## 24 3.23368930
## 25 2.36783512
## 26 1.01815983
## 27 1.22609767

```

```
## 28 1.03873926
## 29 1.53584203
## 30 2.02464632
## 31 2.25872324
## 32 1.41801852
## 33 1.00050145
## 34 2.20247183
## 35 1.51350266
## 36 1.00005125
## 37 1.38132470
## 38 1.00017137
## 39 0.99968029
## 40 -0.15312607
## 41 0.99989784
## 42 0.54500162
## 43 -0.34608893
## 44 0.99335155
## 45 1.10181777
## 46 -1.22421871
## 47 1.82195640
## 48 1.00011327
## 49 -0.99982941
## 50 -1.33936905
## 51 -0.99978074
## 52 -1.34102164
## 53 -0.99998435
## 54 -1.48369060
## 55 -1.32409847
## 56 -1.00000925
## 57 -1.19318562
## 58 -1.00011815
## 59 -1.00031059
## 60 -0.99994504
## 61 -1.31225385
## 62 -1.02438907
## 63 -1.36801539
## 64 -1.13007411
## 65 -1.38371490
## 66 -1.31301937
## 67 -1.00039291
## 68 -1.41047621
## 69 -1.24379396
## 70 -1.14628030
## 71 -1.75057107
## 72 -1.00003862
## 73 -0.81126399
## 74 -1.45998519
## 75 -1.00030556
## 76 -1.64871324
## 77 -0.99990886
## 78 -1.00043954
## 79 -1.16562506
## 80 -0.64399060
## 81 -1.40817067
```

```
## 82 -1.33508286
## 83 -1.45478526
## 84 -1.74365552
## 85 -1.25886156
## 86 -1.28592656
## 87 -0.72046616
## 88 -0.83859543
## 89 -0.60072389
## 90 -1.00001921
## 91 -1.51442051
## 92 -1.58649776
## 93 -1.88997280
## 94 -1.00013391
## 95 -1.34791193
## 96 -0.99996855
## 97 -1.19699873
## 98 -1.45729883
## 99 -1.82592684
## 100 -1.22667376
## 101 -1.70672708
## 102 -1.85201423
## 103 -1.72809941
## 104 -1.00015021
## 105 -1.85722493
## 106 -1.10405970
## 107 -1.65469046
## 108 -1.40175046
## 109 -1.20895595
## 110 -1.69895663
## 111 -1.01564969
## 112 -2.10673790
## 113 -2.08444098
## 114 -2.14238536
## 115 -1.98450616
## 116 -1.73843368
## 117 -1.24869539
## 118 -1.79172220
## 119 -1.54311275
## 120 -1.31472964
## 121 -1.14140122
## 122 -2.05569078
## 123 -1.45033072
## 124 -2.11638724
## 125 -2.01902485
## 126 -1.44948974
## 127 -2.11048715
## 128 -1.95800633
## 129 -1.47407412
## 130 -1.54084901
## 131 -1.77124158
## 132 -2.07719719
## 133 -1.98184404
## 134 -1.23419001
## 135 -2.11932947
```

```
## 136 -1.30974095
## 137 -1.63279782
## 138 -2.01115677
## 139 -2.20176891
## 140 -2.15906220
## 141 -1.02479666
## 142 -1.45103158
## 143 -2.26848880
## 144 -2.03578839
## 145 -2.13441209
## 146 -1.96870407
## 147 -1.00025991
## 148 -1.81343008
## 149 -1.97680288
## 150 -1.57157055
## 151 -1.38325664
## 152 -0.99969850
## 153 -0.99969360
## 154 -1.00016347
## 155 -1.96973966
## 156 -1.75092848
## 157 -1.67240405
## 158 -1.36743701
## 159 -1.33451967
## 160 -0.99976312
## 161 -1.97102221
## 162 -1.52020909
## 163 -1.58389347
## 164 -1.97492597
## 165 -1.94074250
## 166 -1.99425535
## 167 -1.84659527
## 168 -1.88713166
## 169 -1.64967456
## 170 -1.22779121
## 171 -1.30380515
## 172 -2.12805132
## 173 -1.89383768
## 174 -1.86425830
## 175 -1.98125718
## 176 -2.06872705
## 177 -2.03883939
## 178 -1.30419370
## 179 -1.37571565
## 180 -2.01695376
## 181 -1.91758243
## 182 -1.59088620
## 183 -1.99310900
## 184 -1.16846088
## 185 -2.17826538
## 186 -1.90327593
## 187 -2.18295000
## 188 -1.46901001
## 189 -1.37478441
```

```
## 190 -1.90713955
## 191 -2.04489819
## 192 -1.46264849
## 193 -1.84763979
## 194 -1.40403949
## 195 -1.57689607
## 196 -0.99964569
## 197 -1.35963572
## 198 -2.19903802
## 199 -1.57485282
## 200 -1.50189980
## 201 -1.54196852
## 202 -1.78934418
## 203 -1.51312026
## 204 -2.24190943
## 205 -2.03676612
## 206 -1.93549927
## 207 -1.94303788
## 208 -2.19714050
## 209 -1.64387091
## 210 -2.02720747
## 211 -2.23276107
## 212 -2.04275037
## 213 -1.81730250
## 214 -2.19715673
## 215 -2.05827240
## 216 -2.03083660
## 217 -2.14572442
## 218 -1.90269426
## 219 -1.68615268
## 220 -1.49486098
## 221 -1.38826643
## 222 -1.71770017
## 223 -0.99981826
## 224 -2.22591117
## 225 -1.16110271
## 226 -2.15640800
## 227 -2.16429907
## 228 -2.17165595
## 229 -1.93042168
## 230 -1.83308194
## 231 -1.54683295
## 232 -2.20623393
## 233 -1.67612072
## 234 -1.50476327
## 235 -1.51064608
## 236 -2.07391867
## 237 -1.66016426
## 238 -1.29694938
## Levels: 0 1
```

```
train.error <- mean(pred.train != train.data.y)
## 3v. TEST PREDICTION on model test data (input your own data here)
pred.test <- predict(svm.fit.nl, test.data.x, decision.values = T)
pred.test
```



```

## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
## 0 0 1 1 0 1 1 1 1 1 0 1 1 0 1 1 1 1 1 0 1 1 0 0 1 0
## 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52
## 0 1 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78
## 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0
## 79 80 81 82 83 84 85 86 87 88 89 90 91
## 0 0 0 0 0 0 0 0 0 0 0 0 0
## attr("decision.values")
##          1/0
## 1 -1.10838614
## 2 -1.68146537
## 3  1.84267841
## 4  1.46077257
## 5 -0.29195118
## 6  1.89560243
## 7  1.26201548
## 8  1.44267134
## 9  1.22392770
## 10 2.72619604
## 11 -0.22741094
## 12 1.50185808
## 13 2.37648845
## 14 -0.89352589
## 15 2.33504530
## 16 1.49149621
## 17 2.38554900
## 18 1.56626282
## 19 0.13526691
## 20 -0.46373104
## 21 0.42434875
## 22 2.70800503
## 23 -0.25116126
## 24 -1.30385341
## 25 1.88927755
## 26 -0.25184676
## 27 -1.20179754
## 28 2.34400968
## 29 2.62802313
## 30 -0.28307077
## 31 -1.29253201
## 32 -1.83925560
## 33 -1.67612337
## 34 -1.51218959
## 35 -1.63613677
## 36 -1.19382645
## 37 1.00351115
## 38 -0.96879941
## 39 -1.89556792
## 40 -1.82885917
## 41 -2.10033287
## 42 -1.52396100
## 43 -2.13854307

```

```
## 44 -1.95699159
## 45 -1.76617713
## 46 -1.64336201
## 47 -1.17575887
## 48 -1.51617705
## 49 -1.43232796
## 50 -0.97461950
## 51 -1.36819899
## 52 -0.81403606
## 53 -2.04680248
## 54 -1.22466030
## 55 -1.64364752
## 56 -0.91716338
## 57 -1.89494261
## 58 -0.65545829
## 59 -0.41964101
## 60 -1.09573596
## 61 -0.90888521
## 62 -0.67992670
## 63 -0.83708922
## 64 -1.48518092
## 65 0.56238844
## 66 -1.30946419
## 67 -0.67316744
## 68 -1.23240729
## 69 -1.32126667
## 70 -2.22658432
## 71 -1.30218260
## 72 -2.11260185
## 73 -1.72202398
## 74 -1.33367835
## 75 -1.41155882
## 76 0.06918264
## 77 -1.09800352
## 78 -0.93326285
## 79 -0.40785833
## 80 -2.06666394
## 81 -1.15613397
## 82 -0.98288529
## 83 -1.99655007
## 84 -1.98179103
## 85 -1.37884696
## 86 -0.04678794
## 87 -1.27077013
## 88 -1.45966451
## 89 -1.65467547
## 90 -0.98142767
## 91 -1.35004939
## Levels: 0 1
```