

FightML_HitsLandedByArse

Janis Corona

10/28/2019

This machine learning program examines the Arse May 2019 UFC fight

https://www.espn.com/mma/fightcenter/_/id/401107798/league/ufc has access to the 3 rounds of 5 minutes fight 5/18/2019 The csv file is found at:

<https://github.com/JanJanJan2018/FightML/blob/master/ArseErosaFightAudit.csv>

Predict the number of hits Arse lands based on the seconds into the round, elapsed since last action, cumulative number of hits received/landed/missed and number of hits received total for that round This data is split into a training set of 70% and a testing set of 30% based only on those instances action other than circling/standing/stepping away/toward/to the side occurs There are 224 instances of 156 training set samples of each second of action, and the remaining 68 are testing samples of each second of action

```
ArseErosa <- read.csv('ArseErosaFightAudit.csv',
                      sep=',', header=TRUE,
                      na.strings=c('', 'NA')) #224X18

ArseErosa <- ArseErosa[,1:8]

ArseErosa

##      Round SecondsIntoRound SecondsLastRoundAction cmTotHitsR.A
## 1         1             6              6              0
## 2         1            19             13              0
## 3         1            21              2              0
## 4         1            24              3              1
## 5         1            36             12              1
## 6         1            39              3              2
## 7         1            46              7              2
## 8         1            49              3              2
## 9         1            51              2              2
## 10        1            57              6              2
## 11        1            61              4              2
## 12        1            76             15              2
## 13        1            80              4              2
```

## 14	1	88	8	2
## 15	1	89	1	2
## 16	1	103	14	2
## 17	1	104	1	2
## 18	1	106	2	3
## 19	1	107	1	4
## 20	1	121	14	4
## 21	1	127	6	4
## 22	1	132	5	4
## 23	1	145	13	4
## 24	1	152	7	4
## 25	1	153	1	4
## 26	1	161	8	5
## 27	1	177	16	5
## 28	1	179	2	6
## 29	1	181	2	7
## 30	1	184	3	7
## 31	1	185	1	7
## 32	1	189	4	7
## 33	1	190	1	7
## 34	1	195	5	7
## 35	1	205	10	7
## 36	1	206	1	7
## 37	1	212	6	8
## 38	1	214	2	8
## 39	1	218	4	9
## 40	1	226	8	9
## 41	1	227	1	10
## 42	1	234	7	10
## 43	1	239	5	10
## 44	1	241	2	10
## 45	1	242	1	10
## 46	1	243	1	10
## 47	1	244	1	10
## 48	1	254	10	10
## 49	1	255	1	10
## 50	1	256	1	11

## 51	1	259	3	11
## 52	1	262	3	12
## 53	1	263	1	12
## 54	1	264	1	12
## 55	1	265	1	12
## 56	1	266	1	12
## 57	1	267	1	12
## 58	1	273	6	12
## 59	1	274	1	12
## 60	1	275	1	12
## 61	1	276	1	12
## 62	1	284	8	12
## 63	1	287	3	12
## 64	1	288	1	12
## 65	1	294	6	13
## 66	1	295	1	13
## 67	1	297	2	14
## 68	1	302	5	14
## 69	2	1	0	0
## 70	2	2	1	0
## 71	2	3	1	0
## 72	2	4	1	0
## 73	2	6	2	0
## 74	2	8	2	1
## 75	2	9	1	2
## 76	2	16	7	2
## 77	2	18	2	2
## 78	2	23	5	2
## 79	2	24	1	2
## 80	2	25	1	2
## 81	2	43	18	2
## 82	2	47	4	2
## 83	2	49	2	2
## 84	2	52	3	2
## 85	2	66	14	2
## 86	2	67	1	2
## 87	2	68	1	2

## 88	2	73	5	2
## 89	2	74	1	2
## 90	2	75	1	2
## 91	2	76	1	2
## 92	2	96	20	2
## 93	2	102	6	2
## 94	2	103	1	2
## 95	2	105	2	2
## 96	2	106	1	2
## 97	2	108	2	2
## 98	2	115	7	2
## 99	2	119	4	2
## 100	2	121	2	2
## 101	2	127	6	2
## 102	2	129	2	3
## 103	2	131	2	3
## 104	2	132	1	3
## 105	2	133	1	3
## 106	2	135	2	3
## 107	2	138	3	3
## 108	2	139	1	4
## 109	2	140	1	4
## 110	2	148	8	4
## 111	2	150	2	4
## 112	2	153	3	4
## 113	2	155	2	4
## 114	2	166	11	4
## 115	2	167	1	4
## 116	2	169	2	4
## 117	2	170	1	4
## 118	2	171	1	4
## 119	2	173	2	4
## 120	2	174	1	4
## 121	2	175	1	4
## 122	2	178	3	4
## 123	2	179	1	4
## 124	2	186	7	5

## 125	2	187	1	5
## 126	2	189	2	5
## 127	2	190	1	5
## 128	2	197	7	5
## 129	2	200	3	5
## 130	2	202	2	5
## 131	2	207	5	5
## 132	2	211	4	5
## 133	2	213	2	5
## 134	2	214	1	5
## 135	2	219	5	5
## 136	2	221	2	5
## 137	2	223	2	5
## 138	2	224	1	5
## 139	2	225	1	5
## 140	2	227	2	5
## 141	2	229	2	5
## 142	2	231	2	5
## 143	2	232	1	5
## 144	2	233	1	5
## 145	2	235	2	5
## 146	2	236	1	5
## 147	2	239	3	5
## 148	2	241	2	5
## 149	2	245	4	5
## 150	2	247	2	5
## 151	2	255	8	5
## 152	2	257	2	5
## 153	2	258	1	5
## 154	2	259	1	5
## 155	2	260	1	5
## 156	2	264	4	5
## 157	2	267	3	5
## 158	2	272	5	6
## 159	2	275	3	6
## 160	2	276	1	7
## 161	2	277	1	7

## 162	2	280	3	7
## 163	2	297	17	7
## 164	2	302	5	7
## 165	3	1	0	0
## 166	3	2	1	0
## 167	3	3	1	0
## 168	3	5	2	0
## 169	3	6	1	0
## 170	3	7	1	0
## 171	3	10	3	0
## 172	3	11	1	0
## 173	3	13	2	0
## 174	3	14	1	0
## 175	3	15	1	0
## 176	3	16	1	0
## 177	3	17	1	0
## 178	3	19	2	0
## 179	3	21	2	0
## 180	3	24	3	0
## 181	3	27	3	0
## 182	3	29	2	0
## 183	3	32	3	0
## 184	3	35	3	0
## 185	3	36	1	0
## 186	3	38	2	0
## 187	3	41	3	0
## 188	3	43	2	0
## 189	3	50	7	0
## 190	3	51	1	1
## 191	3	52	1	1
## 192	3	56	4	2
## 193	3	57	1	2
## 194	3	58	1	2
## 195	3	59	1	2
## 196	3	60	1	2
## 197	3	65	5	3
## 198	3	66	1	3

## 199	3	67	1	4
## 200	3	68	1	4
## 201	3	69	1	4
## 202	3	71	2	4
## 203	3	73	2	4
## 204	3	74	1	4
## 205	3	76	2	4
## 206	3	77	1	4
## 207	3	78	1	4
## 208	3	79	1	4
## 209	3	81	2	4
## 210	3	82	1	4
## 211	3	83	1	4
## 212	3	87	4	4
## 213	3	88	1	4
## 214	3	89	1	4
## 215	3	90	1	5
## 216	3	91	1	5
## 217	3	93	2	5
## 218	3	94	1	5
## 219	3	103	9	5
## 220	3	104	1	6
## 221	3	105	1	6
## 222	3	106	1	6
## 223	3	107	1	6
## 224	3	111	4	6
##	cmTotHitsL.A	cmTotHitsM.A	Hits.Recvd.A	Hits.Lnd.A
## 1	0	0	0	0
## 2	0	0	0	0
## 3	0	0	0	0
## 4	0	0	1	0
## 5	0	1	0	0
## 6	0	1	1	0
## 7	0	2	0	0
## 8	0	2	0	0
## 9	0	2	0	0
## 10	0	3	0	0

## 11	0	3	0	0
## 12	0	3	0	0
## 13	0	3	0	0
## 14	0	3	0	0
## 15	0	3	0	0
## 16	0	3	0	0
## 17	0	3	0	0
## 18	0	4	1	0
## 19	0	4	1	0
## 20	0	4	0	0
## 21	0	5	0	0
## 22	0	6	0	0
## 23	0	8	0	0
## 24	0	9	0	0
## 25	0	10	0	0
## 26	0	10	1	0
## 27	2	10	0	2
## 28	2	10	1	0
## 29	2	10	1	0
## 30	3	10	0	1
## 31	3	11	0	0
## 32	3	11	0	0
## 33	4	12	0	1
## 34	5	12	0	1
## 35	5	12	0	0
## 36	5	14	0	0
## 37	5	14	1	0
## 38	6	14	0	1
## 39	6	14	1	0
## 40	6	15	0	0
## 41	6	15	1	0
## 42	7	15	0	1
## 43	8	15	0	1
## 44	8	15	0	0
## 45	9	15	0	1
## 46	9	15	0	0
## 47	9	16	0	0

## 48	9	17	0	0
## 49	10	17	0	1
## 50	10	17	1	0
## 51	10	17	0	0
## 52	10	17	1	0
## 53	10	17	0	0
## 54	10	17	0	0
## 55	11	17	0	1
## 56	12	18	0	1
## 57	12	19	0	0
## 58	13	19	0	1
## 59	13	20	0	0
## 60	13	21	0	0
## 61	13	21	0	0
## 62	13	21	0	0
## 63	13	21	0	0
## 64	14	21	0	1
## 65	14	21	1	0
## 66	14	21	0	0
## 67	14	21	1	0
## 68	14	21	0	0
## 69	0	0	0	0
## 70	0	1	0	0
## 71	1	1	0	1
## 72	1	1	0	0
## 73	2	1	0	1
## 74	2	1	1	0
## 75	2	2	1	0
## 76	2	2	0	0
## 77	2	3	0	0
## 78	2	4	0	0
## 79	2	5	0	0
## 80	3	5	0	1
## 81	3	5	0	0
## 82	3	5	0	0
## 83	4	5	0	1
## 84	4	6	0	0

## 85	4	6	0	0
## 86	4	6	0	0
## 87	4	7	0	0
## 88	4	8	0	0
## 89	4	8	0	0
## 90	4	8	0	0
## 91	4	9	0	0
## 92	4	10	0	0
## 93	4	10	0	0
## 94	4	10	0	0
## 95	4	11	0	0
## 96	4	11	0	0
## 97	5	11	0	1
## 98	6	11	0	1
## 99	6	11	0	0
## 100	6	11	0	0
## 101	6	11	0	0
## 102	6	11	1	0
## 103	6	11	0	0
## 104	6	11	0	0
## 105	7	11	0	1
## 106	7	12	0	0
## 107	7	12	0	0
## 108	7	12	1	0
## 109	7	12	0	0
## 110	7	14	0	0
## 111	8	14	0	1
## 112	9	14	0	1
## 113	9	14	0	0
## 114	9	14	0	0
## 115	9	14	0	0
## 116	9	15	0	0
## 117	10	15	0	1
## 118	11	15	0	1
## 119	11	15	0	0
## 120	11	15	0	0
## 121	11	17	0	0

## 122	11	17	0	0
## 123	11	18	0	0
## 124	11	19	1	0
## 125	11	19	0	0
## 126	11	19	0	0
## 127	12	19	0	1
## 128	12	20	0	0
## 129	12	20	0	0
## 130	12	21	0	0
## 131	12	21	0	0
## 132	12	21	0	0
## 133	12	22	0	0
## 134	12	23	0	0
## 135	12	24	0	0
## 136	12	25	0	0
## 137	12	26	0	0
## 138	12	27	0	0
## 139	12	27	0	0
## 140	12	27	0	0
## 141	12	28	0	0
## 142	12	28	0	0
## 143	12	28	0	0
## 144	13	28	0	1
## 145	13	28	0	0
## 146	13	29	0	0
## 147	13	30	0	0
## 148	14	30	0	1
## 149	15	30	0	1
## 150	15	31	0	0
## 151	16	31	0	1
## 152	16	32	0	0
## 153	16	32	0	0
## 154	16	33	0	0
## 155	16	33	0	0
## 156	16	33	0	0
## 157	16	33	0	0
## 158	16	33	1	0

## 159	16	34	0	0
## 160	16	35	1	0
## 161	16	35	0	0
## 162	16	35	0	0
## 163	16	35	0	0
## 164	16	35	0	0
## 165	0	0	0	0
## 166	1	1	0	1
## 167	1	1	0	0
## 168	1	2	0	0
## 169	1	3	0	0
## 170	2	3	0	1
## 171	2	4	0	0
## 172	3	4	0	1
## 173	3	5	0	0
## 174	3	5	0	0
## 175	4	5	0	1
## 176	5	5	0	1
## 177	5	5	0	0
## 178	5	6	0	0
## 179	5	7	0	0
## 180	5	9	0	0
## 181	5	10	0	0
## 182	5	11	0	0
## 183	5	12	0	0
## 184	6	12	0	1
## 185	6	12	0	0
## 186	6	12	0	0
## 187	6	12	0	0
## 188	6	13	0	0
## 189	7	13	0	1
## 190	7	13	1	0
## 191	8	13	0	1
## 192	8	13	1	0
## 193	8	14	0	0
## 194	9	14	0	1
## 195	10	14	0	1

```
## 196      10      14      0      0
## 197      11      14      1      1
## 198      11      14      0      0
## 199      12      14      1      1
## 200      13      14      0      1
## 201      13      15      0      0
## 202      13      15      0      0
## 203      13      15      0      0
## 204      13      16      0      0
## 205      14      16      0      1
## 206      14      17      0      0
## 207      14      18      0      0
## 208      14      19      0      0
## 209      14      19      0      0
## 210      14      20      0      0
## 211      14      21      0      0
## 212      14      21      0      0
## 213      14      22      0      0
## 214      14      22      0      0
## 215      14      22      1      0
## 216      14      23      0      0
## 217      14      23      0      0
## 218      15      23      0      1
## 219      15      24      0      0
## 220      15      25      1      0
## 221      16      26      0      1
## 222      17      26      0      1
## 223      17      26      0      0
## 224      17      26      0      0
```

```
head(ArseErosa)
```

```
##      Round SecondsIntoRound SecondsLastRoundAction cmTotHitsR.A cmTotHitsL.A
## 1      1           6           6           0           0
## 2      1          19          13           0           0
## 3      1          21           2           0           0
## 4      1          24           3           1           0
## 5      1          36          12           1           0
## 6      1          39           3           2           0
```

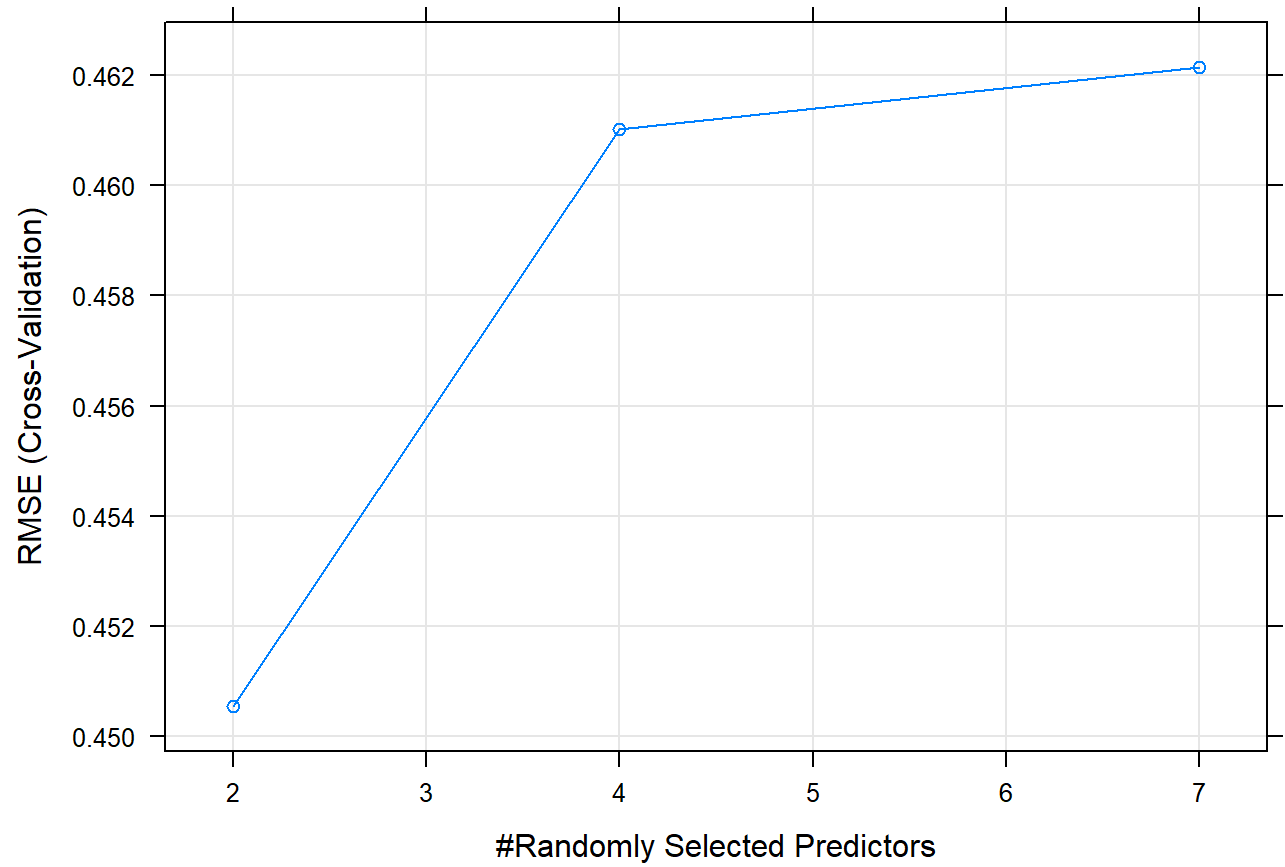
```
##      cmTotHitsM.A Hits.Recvd.A Hits.Lnd.A
## 1          0          0          0
## 2          0          0          0
## 3          0          0          0
## 4          0          1          0
## 5          1          0          0
## 6          1          1          0

library(caret)
library(randomForest)
library(MASS)
library(gbm)
library(dplyr)
set.seed(189678345)
inTrain <- createDataPartition(y=ArseErosa$Hits.Lnd.A, p=0.7, list=FALSE)

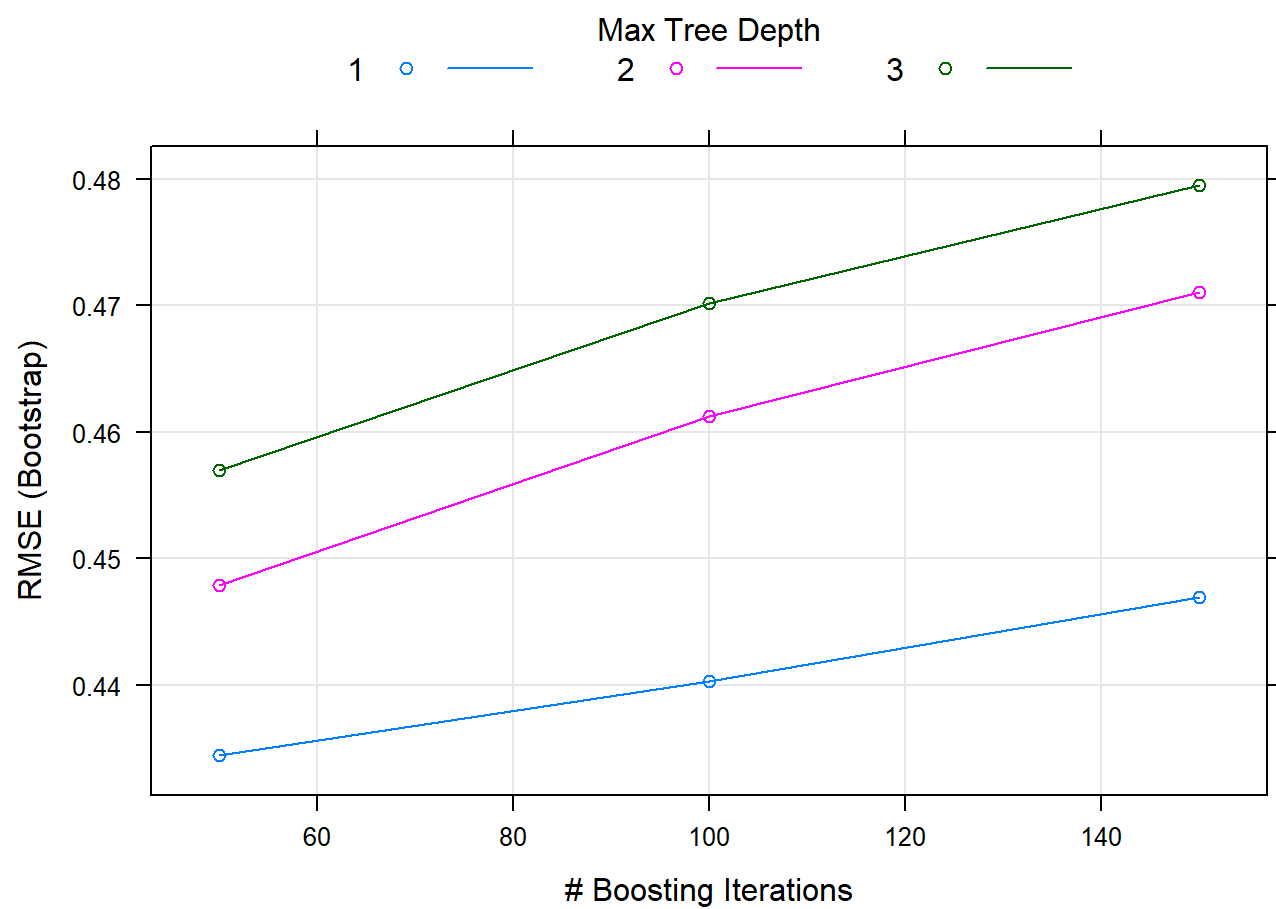
trainingSet <- ArseErosa[inTrain,]
testingSet <- ArseErosa[-inTrain,]
system.time(rfMod <- train(Hits.Lnd.A~., method='rf', data=(trainingSet),
                           trControl=trainControl(method='cv'), number=5))

##      user  system elapsed
##      7.11    0.20    7.81

plot(rfMod)
```



```
system.time(gbmMod <- train(Hits.Lnd.A~., method='gbm', data=trainingSet, verbose=FALSE))  
##      user  system elapsed  
##    3.01    0.14    3.92  
plot(gbmMod)
```



```
predRF <- round(predict(rfMod, testingSet))
predGbm <- round(predict(gbmMod, testingSet))

predDF <- data.frame(predRF, predGbm, type=testingSet$Hits.Lnd.A)
predDF
```

##	predRF	predGbm	type
## 1	0	0	0
## 2	0	0	0
## 5	0	0	0
## 6	0	0	0
## 9	0	0	0
## 12	0	0	0
## 13	0	0	0
## 16	0	0	0
## 19	0	0	0

## 22	0	0	0
## 23	0	0	0
## 24	0	0	0
## 26	0	0	0
## 27	0	0	2
## 28	0	0	0
## 34	0	0	1
## 38	0	0	1
## 40	0	0	0
## 41	0	0	0
## 47	0	0	0
## 50	0	0	0
## 55	0	0	1
## 59	0	0	0
## 60	0	0	0
## 66	0	0	0
## 71	0	0	1
## 72	0	0	0
## 73	0	0	1
## 83	0	0	1
## 93	0	0	0
## 96	0	0	0
## 99	0	0	0
## 102	0	0	0
## 105	0	0	1
## 108	0	0	0
## 118	0	0	1
## 119	0	0	0
## 125	0	0	0
## 126	0	0	0
## 128	0	0	0
## 129	0	0	0
## 131	0	0	0
## 132	0	0	0
## 133	0	0	0
## 134	0	0	0
## 142	0	0	0

```

## 146      0      0      0
## 147      1      0      0
## 153      0      0      0
## 157      0      0      0
## 161      0      0      0
## 162      0      0      0
## 164      0      0      0
## 166      0      0      1
## 169      0      0      0
## 171      0      0      0
## 172      1      0      1
## 174      1      0      0
## 182      0      0      0
## 185      0      0      0
## 187      0      0      0
## 190      1      0      0
## 195      0      0      1
## 206      0      0      0
## 211      0      0      0
## 219      0      0      0
## 222      0      0      1

CombinedModels <- train(type~., method='gam', data=predDF)
CombinedPredictions <- round(predict(CombinedModels, predDF))
CombinedPredictions

##   1   2   5   6   9  12  13  16  19  22  23  24  26  27  28  34  38  40
##   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##  41  47  50  55  59  60  66  71  72  73  83  93  96  99 102 105 108 118
##   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 119 125 126 128 129 131 132 133 134 142 146 147 153 157 161 162 164 166
##   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
## 169 171 172 174 182 185 187 190 195 206 211 219 222
##   0   0   0   0   0   0   0   0   0   0   0   0   0   0

sum <- sum(CombinedPredictions==testingSet$Hits.Lnd.A)
length <- length(testingSet$Hits.Lnd.A)
accuracy_CP1 <- sum/length #97.01

sum <- sum(predRF==testingSet$Hits.Lnd.A)

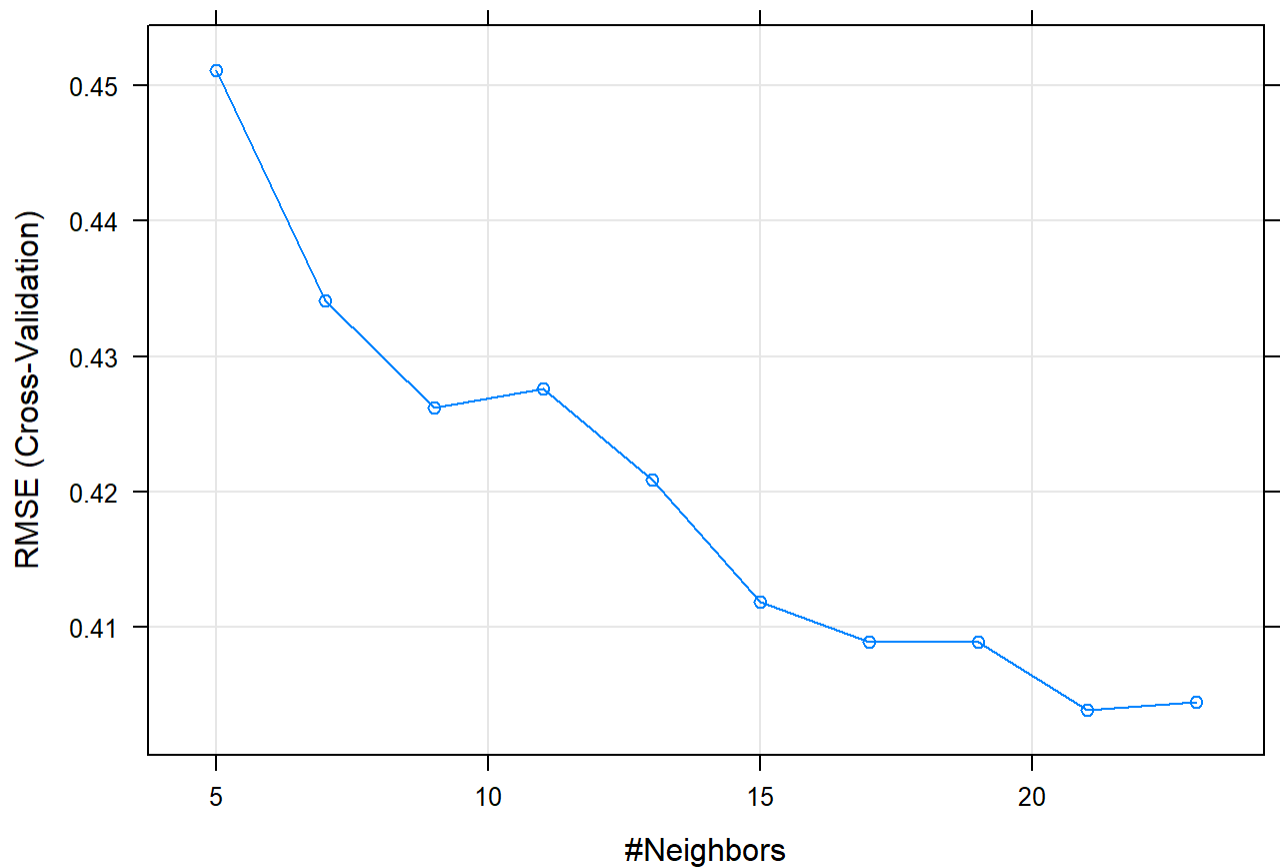
```

```

length <- length(testingSet$Hits.Lnd.A)
accuracy_rfMod <- (sum/length) #97.01

sum <- sum(predGbm==testingSet$Hits.Lnd.A)
accuracy_Gbm <- (sum/length) #97.01
system.time(knnMod <- train(Hits.Lnd.A ~ .,
                           method='knn', preProcess=c('center','scale'),
                           tuneLength=10, trControl=trainControl(method='cv'), data=trainingSet))
##      user  system elapsed
##      1.51    0.03    1.78
plot(knnMod)

```

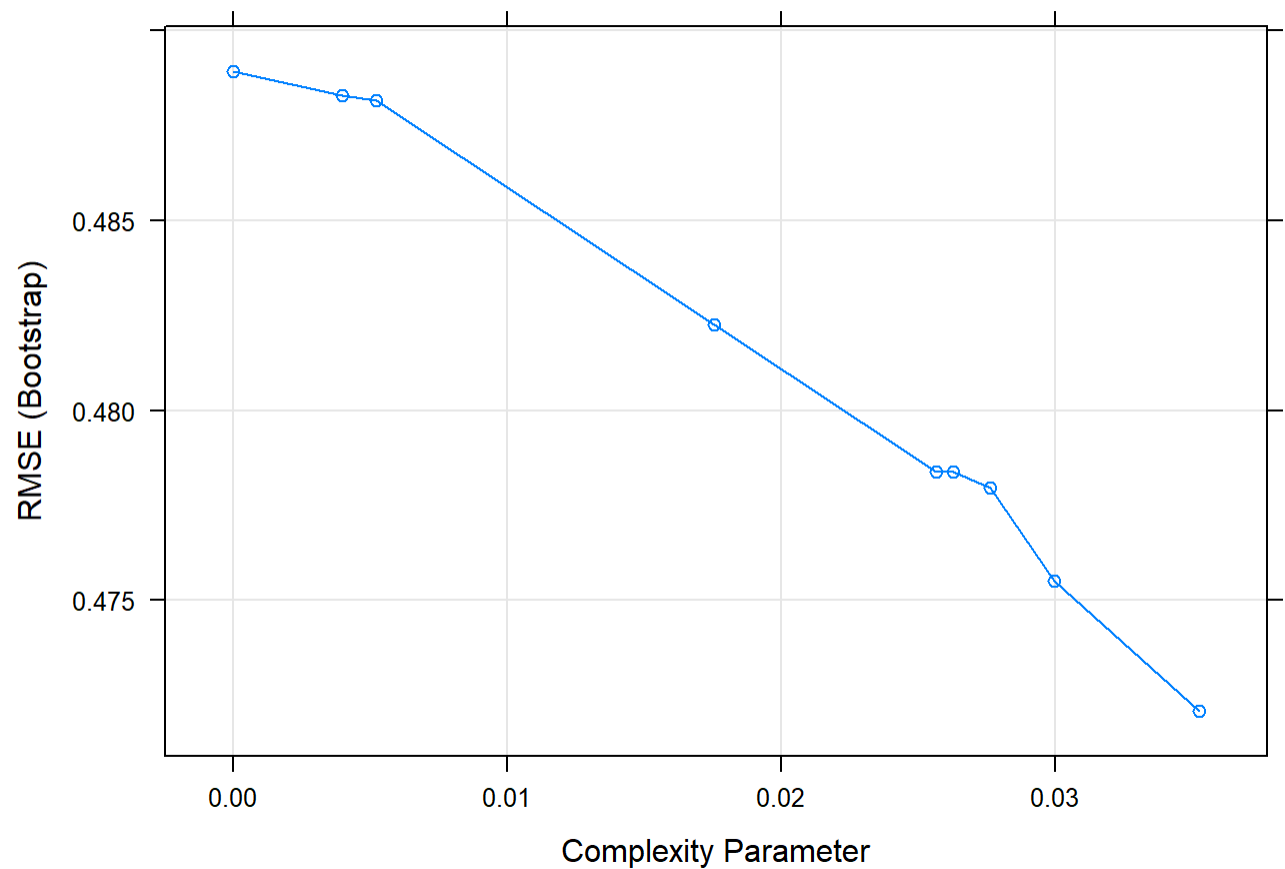


```

system.time(rpartMod <- train(Hits.Lnd.A ~ ., method='rpart', tuneLength=9, data=trainingSet))
##      user  system elapsed
##      2.44    0.00    2.77

```

```
plot(rpartMod)
```



```
system.time(glmMod <- train(Hits.Lnd.A ~ .,
                           method='glm', data=trainingSet))

##      user system elapsed
##    1.08    0.01    1.25

predKNN <- round(predict(knnMod, testingSet))
predRPART <- round(predict(rpartMod, testingSet))
predGLM <- round(predict(glmMod, testingSet))

df3 <- cbind(predKNN, predRPART, predGLM, testingSet$Hits.Lnd.A)
colnames(df3)[4] <- 'TrueValue'
length=length(testingSet$Hits.Lnd.A)

sumKNN <- sum(predKNN==testingSet$Hits.Lnd.A)
sumRPart <- sum(predRPART==testingSet$Hits.Lnd.A)
```

```

sumGLM <- sum(predGLM==testingSet$Hits.Lnd.A)

accuracy_KNN <- sumKNN/length
accuracy_RPART <- sumRPart/length
accuracy_GLM <- sumGLM/length

predDF3 <- data.frame(predRF,predGbm,df3)
system.time(CombinedModels <- train(TrueValue ~ ., method='gam', data=predDF3))
##      user  system elapsed
##    1.55    0.01    1.86
CombinedPredictions2 <- round(predict(CombinedModels, predDF3))
accuracy_CP2 <- sum(CombinedPredictions2==testingSet$Hits.Lnd.A)/length

predDF4 <- data.frame(predDF3, CombinedPredictions2)
colnames(predDF4)
## [1] "predRF"          "predGbm"          "predKNN"
## [4] "predRPART"       "predGLM"          "TrueValue"
## [7] "CombinedPredictions2"
predDF4 <- predDF4[,c(1:5,7,6)]
colnames(predDF4)
## [1] "predRF"          "predGbm"          "predKNN"
## [4] "predRPART"       "predGLM"          "CombinedPredictions2"
## [7] "TrueValue"
results <- c(round(accuracy_rfMod,2),
              round(accuracy_Gbm,2),
              round(accuracy_KNN,2), round(accuracy_RPART,2),
              round(accuracy_GLM,2),
              round(accuracy_CP2,2), round(100,2))

results <- as.factor(results)
results <- t(data.frame(results))#1X7
colnames(results) <- colnames(predDF4)
Results <- rbind(predDF4, results) #68X7
Results
##      predRF predGbm predKNN predRPART predGLM CombinedPredictions2
## 1          0      0      0          0      0                      0
## 2          0      0      0          0      0                      0

```

## 5	0	0	0	0	0	0
## 6	0	0	0	0	0	0
## 9	0	0	0	0	0	0
## 12	0	0	0	0	0	0
## 13	0	0	0	0	0	0
## 16	0	0	0	0	0	0
## 19	0	0	0	0	0	0
## 22	0	0	0	0	0	0
## 23	0	0	0	0	0	0
## 24	0	0	0	0	0	0
## 26	0	0	0	0	0	0
## 27	0	0	0	0	0	0
## 28	0	0	0	0	0	0
## 34	0	0	0	0	0	0
## 38	0	0	0	0	0	0
## 40	0	0	0	0	0	0
## 41	0	0	0	0	0	0
## 47	0	0	0	0	0	0
## 50	0	0	0	0	0	0
## 55	0	0	0	0	0	0
## 59	0	0	0	0	0	0
## 60	0	0	0	0	0	0
## 66	0	0	0	0	0	0
## 71	0	0	0	0	0	0
## 72	0	0	0	0	0	0
## 73	0	0	0	0	0	0
## 83	0	0	0	0	0	0
## 93	0	0	0	0	0	0
## 96	0	0	0	0	0	0
## 99	0	0	0	0	0	0
## 102	0	0	0	0	0	0
## 105	0	0	0	0	0	0
## 108	0	0	0	0	0	0
## 118	0	0	0	0	0	0
## 119	0	0	0	0	0	0
## 125	0	0	0	0	0	0
## 126	0	0	0	0	0	0

## 128	0	0	0	0	0	0
## 129	0	0	0	0	0	0
## 131	0	0	0	0	0	0
## 132	0	0	0	0	0	0
## 133	0	0	0	0	0	0
## 134	0	0	0	0	0	0
## 142	0	0	0	0	0	0
## 146	0	0	0	0	0	0
## 147	1	0	0	0	0	0
## 153	0	0	0	0	0	0
## 157	0	0	0	0	0	0
## 161	0	0	0	0	0	0
## 162	0	0	0	0	0	0
## 164	0	0	0	0	0	0
## 166	0	0	0	0	0	0
## 169	0	0	0	0	0	0
## 171	0	0	0	0	0	0
## 172	1	0	0	0	0	0
## 174	1	0	0	0	0	0
## 182	0	0	0	0	0	0
## 185	0	0	0	0	0	0
## 187	0	0	0	0	0	0
## 190	1	0	0	0	0	0
## 195	0	0	0	0	0	0
## 206	0	0	0	0	0	0
## 211	0	0	0	0	0	0
## 219	0	0	0	0	0	0
## 222	0	0	0	0	0	0
## results	0.78	0.81	0.81	0.81	0.81	0.81
##	TrueValue					
## 1	0					
## 2	0					
## 5	0					
## 6	0					
## 9	0					
## 12	0					
## 13	0					

## 16	0
## 19	0
## 22	0
## 23	0
## 24	0
## 26	0
## 27	2
## 28	0
## 34	1
## 38	1
## 40	0
## 41	0
## 47	0
## 50	0
## 55	1
## 59	0
## 60	0
## 66	0
## 71	1
## 72	0
## 73	1
## 83	1
## 93	0
## 96	0
## 99	0
## 102	0
## 105	1
## 108	0
## 118	1
## 119	0
## 125	0
## 126	0
## 128	0
## 129	0
## 131	0
## 132	0
## 133	0


```
## 134      0
## 142      0
## 146      0
## 147      0
## 153      0
## 157      0
## 161      0
## 162      0
## 164      0
## 166      1
## 169      0
## 171      0
## 172      1
## 174      0
## 182      0
## 185      0
## 187      0
## 190      0
## 195      1
## 206      0
## 211      0
## 219      0
## 222      1
## results    100

#write.csv(Results,'TrueHitsLanded_ML_predictionResults.csv', row.names=FALSE)
```