# R Notebook

Code

Hide

```
df <- read.csv("picks.csv")
str(df)
```

```
'data.frame':	16035 obs. of  17 variables:
 $ date          : chr  "2020-03-18" "2020-03-18" "2020-03-18" "2020-03-17" ...
 $ team_1        : chr  "TeamOne" "Rugratz" "New England Whalers" "Complexity" ...
 $ team_2        : chr  "Recon 5" "Bad News Bears" "Station7" "forZe" ...
 $ inverted_teams: int  1 0 0 1 0 1 0 0 1 1 ...
 $ match_id      : int  2340454 2340453 2340461 2340279 2340456 2340397 2340130 2340131 2340443
2340444 ...
 $ event_id      : int  5151 5151 5243 5226 5247 5226 5243 5243 5236 5236 ...
 $ best_of       : chr  "3" "3" "1" "3" ...
 $ system        : int  123412 123412 121212 123412 123412 123412 121212 121212 123412 123412
...
 $ t1_removed_1  : chr  "Vertigo" "Dust2" "Mirage" "Inferno" ...
 $ t1_removed_2  : chr  "Train" "Nuke" "Dust2" "Nuke" ...
 $ t1_removed_3  : chr  "0.0" "0.0" "Vertigo" "0.0" ...
 $ t2_removed_1  : chr  "Nuke" "Mirage" "Nuke" "Overpass" ...
 $ t2_removed_2  : chr  "Overpass" "Train" "Train" "Vertigo" ...
 $ t2_removed_3  : chr  "0.0" "0.0" "Overpass" "0.0" ...
 $ t1_picked_1   : chr  "Dust2" "Vertigo" "0.0" "Dust2" ...
 $ t2_picked_1   : chr  "Inferno" "Inferno" "0.0" "Train" ...
 $ left_over     : chr  "Mirage" "Overpass" "Inferno" "Mirage" ...
```

Hide

```
df <- df[,c(7,9,10,12,13)] #grab best_of, t1_removed_1, t1_removed_2, t2_removed_1, t2_removed_
2, left_over


df$best_of <- factor(df$best_of)
df$t1_removed_1 <- factor(df$t1_removed_1)
df$t1_removed_2 <- factor(df$t1_removed_2)
df$t2_removed_1 <- factor(df$t2_removed_1)
df$t2_removed_2 <- factor(df$t2_removed_2)

levels(df$best_of)[levels(df$best_of)=="1(Online)"] <- "1"
levels(df$best_of)[levels(df$best_of)=="2(Online)"] <- "2"
levels(df$best_of)[levels(df$best_of)=="3(LAN)"] <- "3"
levels(df$best_of)[levels(df$best_of)=="3(Online)"] <- "3"
levels(df$best_of)[levels(df$best_of)=="3."] <- "3"
levels(df$best_of)[levels(df$best_of)=="of"] <- "3"

sapply(df, function(x) sum(is.na(x)==TRUE))
```

```
    best_of t1_removed_1 t1_removed_2 t2_removed_1 t2_removed_2
          0            0            0            0            0
```

Hide

```
set.seed(1234)
i <- sample(1:nrow(df), .8*nrow(df), replace=FALSE)
train <- df[i,]
test <- df[-i,]
```

# Data Exploration

The Data Exploration of this dataset was already done in pfc4: Searching for Similarity, so we'll just do the same thing here - a cursory look at the columns and the pairs they make

Hide

```
summary(train)
```

```
 best_of   t1_removed_1    t1_removed_2    t2_removed_1    t2_removed_2 
 1:4539   Nuke    :2609   Overpass:2030   Nuke    :2569   Overpass:1952  
 2: 171   Overpass:1984   Train   :1926   Overpass:1962   Train   :1868  
 3:8118   Train   :1699   Nuke    :1745   Train   :1625   Mirage  :1693  
          Cache   :1424   Mirage  :1575   Cache   :1564   Nuke    :1670  
          Mirage  :1252   Inferno :1523   Mirage  :1263   Inferno :1611  
          Inferno :1203   Cache   :1428   Inferno :1173   Cache   :1414  
          (Other) :2657   (Other) :2601   (Other) :2672   (Other) :2620  
```

Hide

```
dim(train)
```

```
[1] 12828     5
```

Hide

```
summary(train$best_of)
```

```
   1    2    3 
4539  171 8118 
```

Hide

```
summary(train$t1_removed_1)
```

```
     Cache Cobblestone       Dust2     Inferno      Mirage        Nuke    Overpass       Train 
Vertigo 
      1424         868         953        1203        1252        2609        1984        1699 
836 
```

Hide

```
summary(train$t1_removed_2)
```

```
       0.0       Cache Cobblestone       Dust2     Inferno      Mirage        Nuke    Overpass 
Train 
         9        1428         831        1229        1523        1575        1745        2030 
1926 
    Vertigo 
       532 
```

Hide

```
summary(train$t2_removed_1)
```

```
       0.0       Cache Cobblestone       Dust2     Inferno      Mirage        Nuke    Overpass 
Train 
         1        1564         899         976        1173        1263        2569        1962 
1625 
    Vertigo 
       796 
```

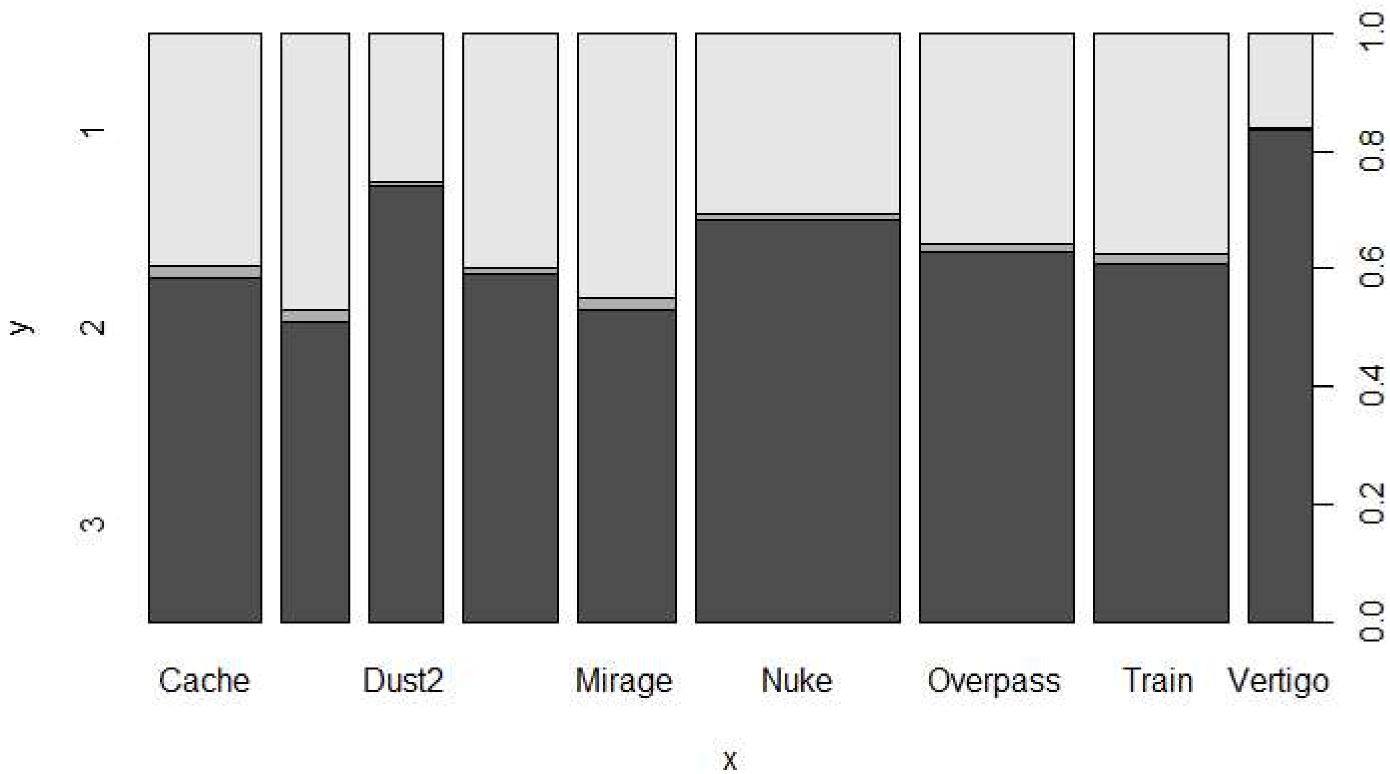
Hide

```
summary(train$t2_removed_2)
```

```
       0.0       Cache Cobblestone       Dust2     Inferno      Mirage        Nuke    Overpass 
Train 
         6        1414         862        1266        1611        1693        1670        1952 
1868 
    Vertigo 
       486 
```

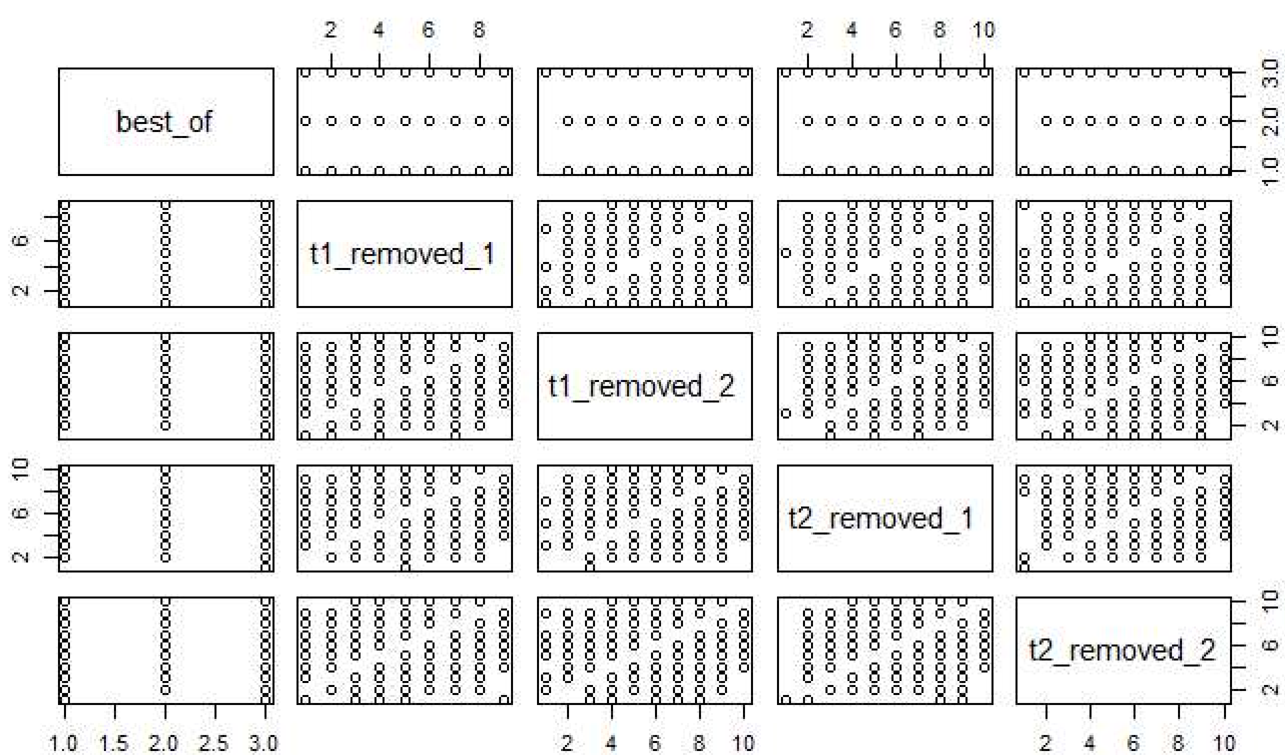
Hide

```
plot(x=train$t1_removed_1, y=train$best_of)
```




Hide

```
pairs(train)
```



# linear kernel

Hide

```
svm1 <- svm(best_of~., data=train, kernel="linear",cost=10,scale=TRUE)
summary(svm1)
```

```


Call:
svm(formula = best_of ~ ., data = train, kernel = "linear", cost = 10, scale = TRUE)


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

Number of Support Vectors:  9677

 ( 4967 4539 171 )


Number of Classes:  3

Levels:
 1 2 3
```

Hide

```
pred <- predict(svm1, newdata=test)
table(pred, test$best_of)
```

```


pred    1    2    3
   1   89    8  108
   2    0    0    0
   3 1030   46 1926
```

Hide

```
acc1 <- mean(pred==test$best_of)
```

# polynomial

Hide

```
svm2 <- svm(best_of~., data=train, kernel="polynomial",cost=10,scale=TRUE)
summary(svm2)
```

```
Call:
svm(formula = best_of ~ ., data = train, kernel = "polynomial", cost = 10, scale = TRUE)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  polynomial
       cost:  10
     degree:  3
     coef.0:  0

Number of Support Vectors:  9685

 ( 4975 4539 171 )


Number of Classes:  3

Levels:
 1 2 3
```

Hide

```
pred <- predict(svm2, newdata=test)
table(pred, test$best_of)
```

```
pred    1    2    3
   1    0    0    0
   2    0    0    0
   3 1119   54 2034
```

Hide

```
acc2 <- mean(pred==test$best_of)
```

# radial

Hide

```
svm3 <- svm(best_of~., data=train, kernel="radial", cost=10, gamma=1, scale=TRUE)
summary(svm3)
```

```
Call:
svm(formula = best_of ~ ., data = train, kernel = "radial", cost = 10, gamma = 1, scale = TRUE)


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

Number of Support Vectors:  9062

 ( 4680 4211 171 )


Number of Classes:  3

Levels:
 1 2 3
```

Hide

```
pred <- predict(svm3, newdata=test)
table(pred, test$best_of)
```

```

pred    1    2    3
   1  309   16  363
   2    2    1    1
   3  808   37 1670
```

Hide

```
acc3 <- mean(pred==test$best_of)
```

# accuracies

Hide

```
print(paste("acc1=",acc1))
```

```
[1] "acc1= 0.628313065169941"
```

Hide

```
print(paste("acc2=",acc2))
```

```
[1] "acc2= 0.634237605238541"
```

Hide

```
print(paste("acc3=",acc3))
```

```
[1] "acc3= 0.617399438727783"
```

# Analysis

The accuracies on these ones are much closer, likely due to the data being more uniform with less variance. As such, the addition of a new hyperparameter in radial SVM doesn’t change too much in the way of the overall accuracy. With a smaller variance than the linear regression on the Twitch data, the polynomial SVM regression is more consistent.