IE6600 Team 1 Project Models: SVM & Linear Regression

```
library(e1071)
library(ISLR)
library(Metrics)
dataset <- read.csv("www/data/crimedata.csv",fileEncoding="latin1")
#keep only columns needed
dataset1 <- dataset[,c(1,2,5,6,7,8,9,10,11,12,13,14,15,17,18,25,27,28,29,30,31,
                       32,34,35,36,37,38,39,48,66,67,69,73,74,75,78,97,98,130,131,
                       132,133,134,135,136,137,138,139,140,141,142,143,144,145,146,147)]
#change to all numeric
for (i in c(41:56)){
  dataset1[,i] = data.frame(apply(dataset1[i], 2, as.numeric))}
```

```
## Warning in apply(dataset1[i], 2, as.numeric): NAs introduced by coercion

## Warning in apply(dataset1[i], 2, as.numeric): NAs introduced by coercion

## Warning in apply(dataset1[i], 2, as.numeric): NAs introduced by coercion

## Warning in apply(dataset1[i], 2, as.numeric): NAs introduced by coercion

## Warning in apply(dataset1[i], 2, as.numeric): NAs introduced by coercion

## Warning in apply(dataset1[i], 2, as.numeric): NAs introduced by coercion

## Warning in apply(dataset1[i], 2, as.numeric): NAs introduced by coercion

## Warning in apply(dataset1[i], 2, as.numeric): NAs introduced by coercion

## Warning in apply(dataset1[i], 2, as.numeric): NAs introduced by coercion

## Warning in apply(dataset1[i], 2, as.numeric): NAs introduced by coercion

## Warning in apply(dataset1[i], 2, as.numeric): NAs introduced by coercion

## Warning in apply(dataset1[i], 2, as.numeric): NAs introduced by coercion

## Warning in apply(dataset1[i], 2, as.numeric): NAs introduced by coercion

## Warning in apply(dataset1[i], 2, as.numeric): NAs introduced by coercion

## Warning in apply(dataset1[i], 2, as.numeric): NAs introduced by coercion

## Warning in apply(dataset1[i], 2, as.numeric): NAs introduced by coercion
```

```r
dataset1[,"OtherPerCap"] <- as.numeric(dataset1[,"OtherPerCap"])
#change column name
names(dataset1)[1] <- "community"
drops <- c("community","state","fold")
#missing value all to 0
dataset1[is.na(dataset1)] <- 0
dataset1[dataset1 == "?"] <- 0
head(dataset1)
```

```
##                community state fold population householdsize racepctblack
## 1 BerkeleyHeightstownship    NJ    1      11980          3.10         1.37
## 2          Marpletownship    PA    1      23123          2.82         0.80
## 3              Tigardcity    OR    1      29344          2.43         0.74
## 4        Gloversvillecity    NY    1      16656          2.40         1.70
## 5             Bemidjicity    MN    1      11245          2.76         0.53
## 6         Springfieldcity    MO    1     140494          2.45         2.51
##   racePctWhite racePctAsian racePctHisp agePct12t21 agePct12t29 agePct16t24
## 1        91.78         6.50        1.88       12.47       21.44       10.93
## 2        95.57         3.44        0.85       11.01       21.30       10.48
## 3        94.33         3.43        2.35       11.36       25.88       11.01
## 4        97.35         0.50        0.70       12.55       25.20       12.19
## 5        89.16         1.17        0.52       24.46       40.53       28.69
## 6        95.65         0.90        0.95       18.09       32.89       20.04
##   agePct65up pctUrban medIncome medFamInc whitePerCap blackPerCap indianPerCap
## 1      11.33      100     75122     79584       30233       13600         5725
## 2      17.18      100     47917     55323       20191       18137            0
## 3      10.28      100     35669     42112       17103       16644        21606
## 4      17.57        0     20580     26501       10909        9984         4941
## 5      12.65        0     17390     24018        9009         887         4425
## 6      13.26      100     21577     27705       12029        7382        10264
##   AsianPerCap OtherPerCap HispPerCap PctPopUnderPov PctLess9thGrade
## 1       27101        1022      22838           1.96            5.81
## 2       20074        1049      12222           3.98            5.61
## 3       15528        1174       8405           4.75            2.80
## 4        3541         717       4391          17.23           11.05
## 5        3352         784       1328          29.99           12.15
## 6       10753        1418       8104          17.78            8.76
##   PctNotHSGrad PctBSorMore PctUnemployed PctEmploy PersPerFam PctSpeakEnglOnly
## 1         9.90       48.18          2.70     64.55       3.22            85.68
## 2        13.72       29.89          2.43     61.96       3.11            87.79
## 3         9.09       30.13          4.01     69.80       2.95            93.11
## 4        33.68       10.81          9.86     54.74       2.98            94.98
## 5        23.06       25.28          9.08     52.44       2.98            94.64
## 6        23.03       20.66          5.72     59.02       2.89            96.87
##   PctNotSpeakEnglWell PctLargHouseOccup PctPersOwnOccup PctPersDenseHous
## 1                1.37              4.17           91.46             0.39
## 2                1.81              3.34           89.03             1.01
## 3                1.14              2.05           64.18             2.03
## 4                0.56              2.56           58.18             1.21
## 5                0.39              3.11           58.13             2.94
## 6                0.60              1.92           57.81             2.11
##   PctHousLess3BR PctHousOccup NumInShelters NumStreet murders murdPerPop rapes
## 1          11.06        98.37            11         0       0       0.00     0
```

```
## 2          23.60        97.15             0         0       0       0.00     1
## 3          47.46        95.68            16         0       3       8.30     6
## 4          45.66        91.19             0         0       0       0.00    10
## 5          55.64        92.45             2         0       0       0.00     0
## 6          53.19        91.81           327         4       7       4.63    77
##   rapesPerPop robberies robbbPerPop assaults assaultPerPop burglaries
## 1        0.00         1        8.20        4         32.81         14
## 2        4.25         5       21.26       24        102.05         57
## 3       16.60        56      154.95       14         38.74        274
## 4       57.86        10       57.86       33        190.93        225
## 5        0.00         4       32.04       14        112.14         91
## 6       50.98       136       90.05      449        297.29       2094
##   burglPerPop larcenies larcPerPop autoTheft autoTheftPerPop arsons
## 1      114.85       138    1132.08        16          131.26      2
## 2      242.37       376    1598.78        26          110.55      1
## 3      758.14      1797    4972.19       136          376.30     22
## 4     1301.78       716    4142.56        47          271.93      0
## 5      728.93      1060    8490.87        91          728.93      5
## 6     1386.46      7690    5091.64       454          300.60    134
##   arsonsPerPop ViolentCrimesPerPop nonViolPerPop
## 1        16.41               41.02       1394.59
## 2         4.25              127.56       1955.95
## 3        60.87              218.59       6167.51
## 4         0.00              306.64          0.00
## 5        40.05                0.00       9988.79
## 6        88.72              442.95       6867.42
```

```
df.working <- dataset1[, !(names(dataset1) %in% drops)]
head(df.working)
```

```
##   population householdsize racepctblack racePctWhite racePctAsian racePctHisp
## 1      11980          3.10         1.37        91.78         6.50        1.88
## 2      23123          2.82         0.80        95.57         3.44        0.85
## 3      29344          2.43         0.74        94.33         3.43        2.35
## 4      16656          2.40         1.70        97.35         0.50        0.70
## 5      11245          2.76         0.53        89.16         1.17        0.52
## 6     140494          2.45         2.51        95.65         0.90        0.95
##   agePct12t21 agePct12t29 agePct16t24 agePct65up pctUrban medIncome medFamInc
## 1       12.47       21.44       10.93      11.33      100     75122     79584
## 2       11.01       21.30       10.48      17.18      100     47917     55323
## 3       11.36       25.88       11.01      10.28      100     35669     42112
## 4       12.55       25.20       12.19      17.57        0     20580     26501
## 5       24.46       40.53       28.69      12.65        0     17390     24018
## 6       18.09       32.89       20.04      13.26      100     21577     27705
##   whitePerCap blackPerCap indianPerCap AsianPerCap OtherPerCap HispPerCap
## 1       30233       13600         5725       27101        1022      22838
## 2       20191       18137            0       20074        1049      12222
## 3       17103       16644        21606       15528        1174       8405
## 4       10909        9984         4941        3541         717       4391
## 5        9009         887         4425        3352         784       1328
## 6       12029        7382        10264       10753        1418       8104
##   PctPopUnderPov PctLess9thGrade PctNotHSGrad PctBSorMore PctUnemployed
## 1           1.96            5.81         9.90       48.18          2.70
## 2           3.98            5.61        13.72       29.89          2.43
```

```
## 3           4.75            2.80         9.09       30.13          4.01
## 4          17.23           11.05        33.68       10.81          9.86
## 5          29.99           12.15        23.06       25.28          9.08
## 6          17.78            8.76        23.03       20.66          5.72
##   PctEmploy PersPerFam PctSpeakEnglOnly PctNotSpeakEnglWell PctLargHouseOccup
## 1     64.55       3.22            85.68                1.37              4.17
## 2     61.96       3.11            87.79                1.81              3.34
## 3     69.80       2.95            93.11                1.14              2.05
## 4     54.74       2.98            94.98                0.56              2.56
## 5     52.44       2.98            94.64                0.39              3.11
## 6     59.02       2.89            96.87                0.60              1.92
##   PctPersOwnOccup PctPersDenseHous PctHousLess3BR PctHousOccup NumInShelters
## 1           91.46             0.39          11.06        98.37            11
## 2           89.03             1.01          23.60        97.15             0
## 3           64.18             2.03          47.46        95.68            16
## 4           58.18             1.21          45.66        91.19             0
## 5           58.13             2.94          55.64        92.45             2
## 6           57.81             2.11          53.19        91.81           327
##   NumStreet murders murdPerPop rapes rapesPerPop robberies robbbPerPop assaults
## 1         0       0       0.00     0        0.00         1        8.20        4
## 2         0       0       0.00     1        4.25         5       21.26       24
## 3         0       3       8.30     6       16.60        56      154.95       14
## 4         0       0       0.00    10       57.86        10       57.86       33
## 5         0       0       0.00     0        0.00         4       32.04       14
## 6         4       7       4.63    77       50.98       136       90.05      449
##   assaultPerPop burglaries burglPerPop larcenies larcPerPop autoTheft
## 1         32.81         14      114.85       138    1132.08        16
## 2        102.05         57      242.37       376    1598.78        26
## 3         38.74        274      758.14      1797    4972.19       136
## 4        190.93        225     1301.78       716    4142.56        47
## 5        112.14         91      728.93      1060    8490.87        91
## 6        297.29       2094     1386.46      7690    5091.64       454
##   autoTheftPerPop arsons arsonsPerPop ViolentCrimesPerPop nonViolPerPop
## 1          131.26      2        16.41               41.02       1394.59
## 2          110.55      1         4.25              127.56       1955.95
## 3          376.30     22        60.87              218.59       6167.51
## 4          271.93      0         0.00              306.64          0.00
## 5          728.93      5        40.05                0.00       9988.79
## 6          300.60    134        88.72              442.95       6867.42
```

```
################set seed and train ind######################
set.seed(1)
train_ind = sample(1:nrow(df.working), 0.7 * nrow(df.working))
normalize <- function(x) {
    return((x - min(x))/(max(x) - min(x)))
}
df.working_dt <- df.working
notneededFeatures <- c("PctSpeakEnglOnlyCat", "PctNotSpeakEnglWellCat",
    "PctHousOccupCat", "RentQrange")
possible_predictors = colnames(df.working)[!(colnames(df.working) %in%
    notneededFeatures)]
df.working = df.working[, names(df.working) %in% possible_predictors]
df.norm <- as.data.frame(lapply(df.working, normalize))
##############################
```

```
####################SVM##################
model_svmradial.cv <- tune.svm(ViolentCrimesPerPop ~ ., data = df.norm[train_ind,], kernel = "radial", g

summary(model_svmradial.cv)
```

```
##
## Parameter tuning of 'svm':
##
## - sampling method: 10-fold cross validation
##
## - best parameters:
##  gamma cost
##  0.002 2.75
##
## - best performance: 0.002354691
##
## - Detailed performance results:
##    gamma cost       error  dispersion
## 1  5e-04 1.75 0.002775946 0.001951957
## 2  1e-03 1.75 0.002677407 0.001970940
## 3  2e-03 1.75 0.002423565 0.001500749
## 4  5e-04 2.00 0.002769160 0.002048916
## 5  1e-03 2.00 0.002679394 0.001997981
## 6  2e-03 2.00 0.002406493 0.001515218
## 7  5e-04 2.25 0.002779464 0.002131277
## 8  1e-03 2.25 0.002664455 0.002000118
## 9  2e-03 2.25 0.002392207 0.001524992
## 10 5e-04 2.50 0.002804139 0.002217998
## 11 1e-03 2.50 0.002654209 0.002014214
## 12 2e-03 2.50 0.002371437 0.001521263
## 13 5e-04 2.75 0.002811058 0.002259564
## 14 1e-03 2.75 0.002657798 0.002025840
## 15 2e-03 2.75 0.002354691 0.001517923
```

```
model_svmradial.tuned <- svm(ViolentCrimesPerPop ~ ., data = df.norm[train_ind,
    ], kernel = "radial", gamma = model_svmradial.cv$best.parameters$gamma,
    cost = model_svmradial.cv$best.parameters$cost)

summary(model_svmradial.tuned)
```

```
##
## Call:
## svm(formula = ViolentCrimesPerPop ~ ., data = df.norm[train_ind,
##     ], kernel = "radial", gamma = model_svmradial.cv$best.parameters$gamma,
##     cost = model_svmradial.cv$best.parameters$cost)
##
##
## Parameters:
##    SVM-Type:  eps-regression
##  SVM-Kernel:  radial
```

```
##        cost:  2.75
##       gamma:  0.002
##     epsilon:  0.1
##
##
## Number of Support Vectors:  353
```

```
y_hat = predict(model_svmradial.tuned, df.norm[-train_ind, -52])
MSE_SVM = mse(df.norm[-train_ind, 52], y_hat)

residul <- df.norm[-train_ind, 52]-y_hat
new_predict <- (y_hat)*(max(df.working$ViolentCrimesPerPop)-min(df.working$ViolentCrimesPerPop))+min(df

yy_hat <- data.frame("Predicted"=new_predict,"Actual"=df.working[-train_ind,52],
                     "Residuals"=residul)
MSE_SVM
```

```
## [1] 0.002421323
```

```
head(yy_hat)
```

```
##     Predicted  Actual    Residuals
## 2    84.90235  127.56  0.008746591
## 3   149.60393  218.59  0.014145012
## 4   348.78902  306.64 -0.008642301
## 6   406.20348  442.95  0.007534563
## 10 1637.65275 1544.24 -0.019153496
## 12 2622.47510 2605.96 -0.003386282
```

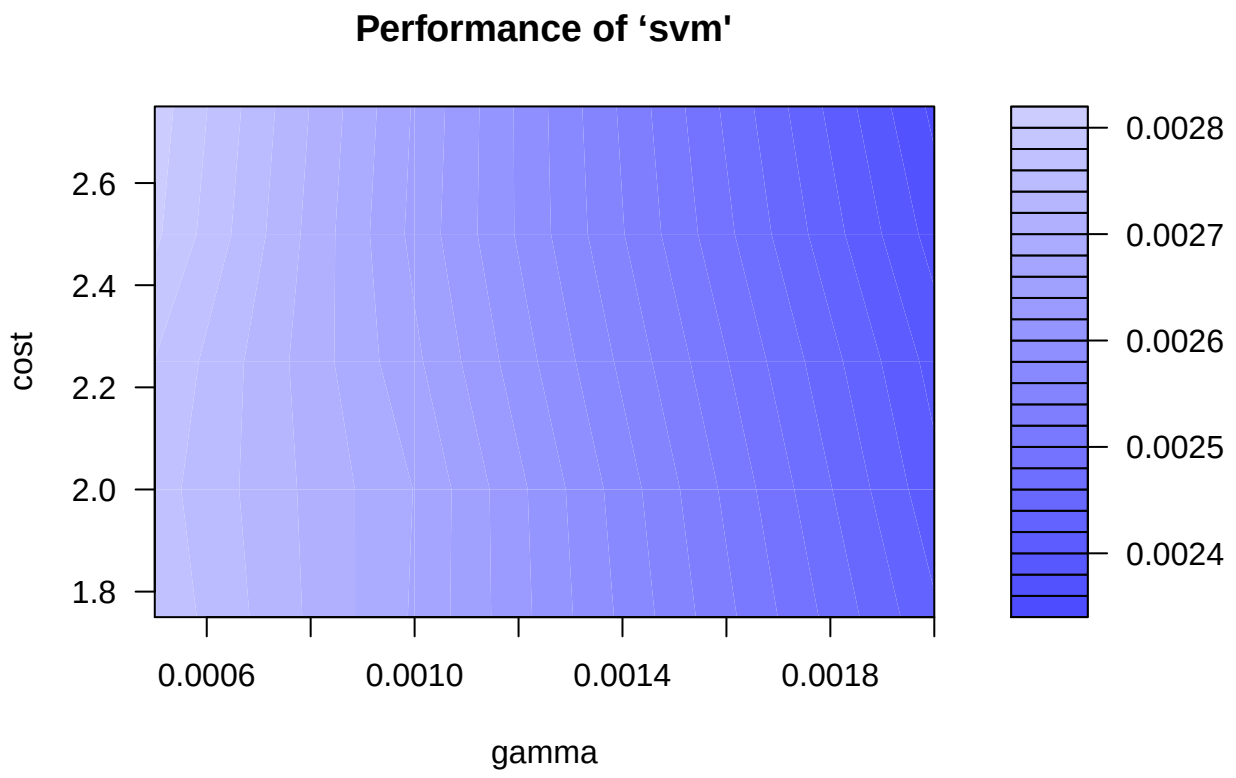
```
accuracy(df.norm[-train_ind, 52], y_hat)
```

```
## [1] 0
```

```
typeof(df.norm[-train_ind, 52])
```

```
## [1] "double"
```

```
plot(model_svmradial.cv, cex = 0.6)
```




```
#Multi-Linear Regression
#clean data
dataset2 <- dataset[,-c(1:5)]
dataset2[dataset2 == "?"] <- NA

#find columns with NA
#names(which(sapply(dataset2, function(x) any(is.na(x)))))

dataset4 <- dataset2[,c("NumUnderPov","PctLess9thGrade","PctUnemployed","NumInShelters",
                        "PctBornSameState","rapesPerPop","robbbPerPop","assaultPerPop",
                        "ViolentCrimesPerPop")]

for (i in 1:length(dataset4)){
  dataset4[,i] = data.frame(apply(dataset4[i], 2, as.numeric))}

library(zoo)
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
# na.aggregate(dataset4)
# replace NA with mean
NA2mean <- function(x) replace(x, is.na(x), mean(x, na.rm = TRUE))
head(replace(dataset4, TRUE, lapply(dataset4, NA2mean)))
```

```
##   NumUnderPov PctLess9thGrade PctUnemployed NumInShelters PctBornSameState
## 1         227            5.81          2.70            11            53.72
## 2         885            5.61          2.43             0            77.17
## 3        1389            2.80          4.01            16            44.77
## 4        2831           11.05          9.86             0            88.71
## 5        2855           12.15          9.08             2            73.75
## 6       23223            8.76          5.72           327            64.35
##   rapesPerPop robbbPerPop assaultPerPop ViolentCrimesPerPop
## 1     0.00000        8.20         32.81             41.0200
## 2     4.25000       21.26        102.05            127.5600
## 3    16.60000      154.95         38.74            218.5900
## 4    57.86000       57.86        190.93            306.6400
## 5    36.25848       32.04        112.14            589.0789
## 6    50.98000       90.05        297.29            442.9500
```

```
dataset4[] <- lapply(dataset4, NA2mean)

#length(dataset4)

set.seed(11)
train_index <- sample(1:nrow(dataset4), 0.8*nrow(dataset4))
normalize <- function(x){
  return((x-min(x))/(max(x) - min(x)))
 }
d4_norm <- as.data.frame(lapply(dataset4, normalize))
train_lm <- d4_norm[train_index,]
test_lm <- d4_norm[-train_index,]

# train_lm <- dataset2[train_index,]
# test_lm <- dataset2[-train_index,]

lm_fit <- lm(ViolentCrimesPerPop ~ ., data = train_lm)
sm <- summary(lm_fit)
sse <- sum(sm$residuals^2)
mse <- mean(sm$residuals^2)

y_hat1 <- predict(lm_fit, test_lm[,-9])
residuls1 <- test_lm$ViolentCrimesPerPop-y_hat1

y_y_hat <- data.frame("Predicted" = y_hat1,
                      "Actual" =test_lm$ViolentCrimesPerPop,
                      "Residul" = residuls1)


new_predict <- (y_hat1)*(max(dataset4$ViolentCrimesPerPop)-min(dataset4$ViolentCrimesPerPop))+min(datas

library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

##
## Attaching package: 'forecast'
```

```
## The following object is masked from 'package:Metrics':
##
##     accuracy
```

```
accuracy(y_hat1, test_lm$ViolentCrimesPerPop)
```

```
##                     ME       RMSE       MAE       MPE     MAPE
## Test set 0.0009447821 0.03987273 0.0235999 -20.10201 45.43916
```

```
sm
```

```
##
## Call:
## lm(formula = ViolentCrimesPerPop ~ ., data = train_lm)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.79702 -0.01533 -0.00406  0.01301  0.18363
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.010965   0.003703   2.961  0.00311 **
## NumUnderPov      -0.607707   0.096969  -6.267 4.61e-10 ***
## PctLess9thGrade   0.103091   0.009997  10.312  < 2e-16 ***
## PctUnemployed    -0.197946   0.015619 -12.674  < 2e-16 ***
## NumInShelters     0.595385   0.108304   5.497 4.42e-08 ***
## PctBornSameState  0.011273   0.005433   2.075  0.03815 *
## rapesPerPop       0.261730   0.014111  18.548  < 2e-16 ***
## robbbPerPop       0.484621   0.013222  36.654  < 2e-16 ***
## assaultPerPop     0.760104   0.015212  49.968  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04215 on 1763 degrees of freedom
## Multiple R-squared:  0.8775, Adjusted R-squared:  0.877
## F-statistic:  1579 on 8 and 1763 DF,  p-value: < 2.2e-16
```

```
mse
```

```
## [1] 0.001767696
```

```
head(new_predict)
```

```
##         3         4         7        10        11        21
##  259.9992  321.8005  237.6893 1525.3505  805.6323  243.6456
```