SF Crime Prediction

Code ▾

Hide

```
library(dplyr)
```

```
package 'dplyr' was built under R version 3.4.2
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

Hide

```
data <- tbl_df(read.csv("/Users/aarthi/Downloads/codes/SF Crime/train.csv", header=TR
UE))
# number of observations for subset
m = 8842
```

Hide

```
# Discarding some variables
data <- data[-c(3, 6, 7)]
# Discarding outlier at Y = 90
data <- data[data$Y != 90,]
```

Hide

```
library(lubridate)
```

```
package 'lubridate' was built under R version 3.4.2
Attaching package: 'lubridate'

The following object is masked from 'package:base':

    date
```

Hide

```
# convert 'Dates' variable from factor to date type
data$Dates = ymd_hms(data$Dates)
```

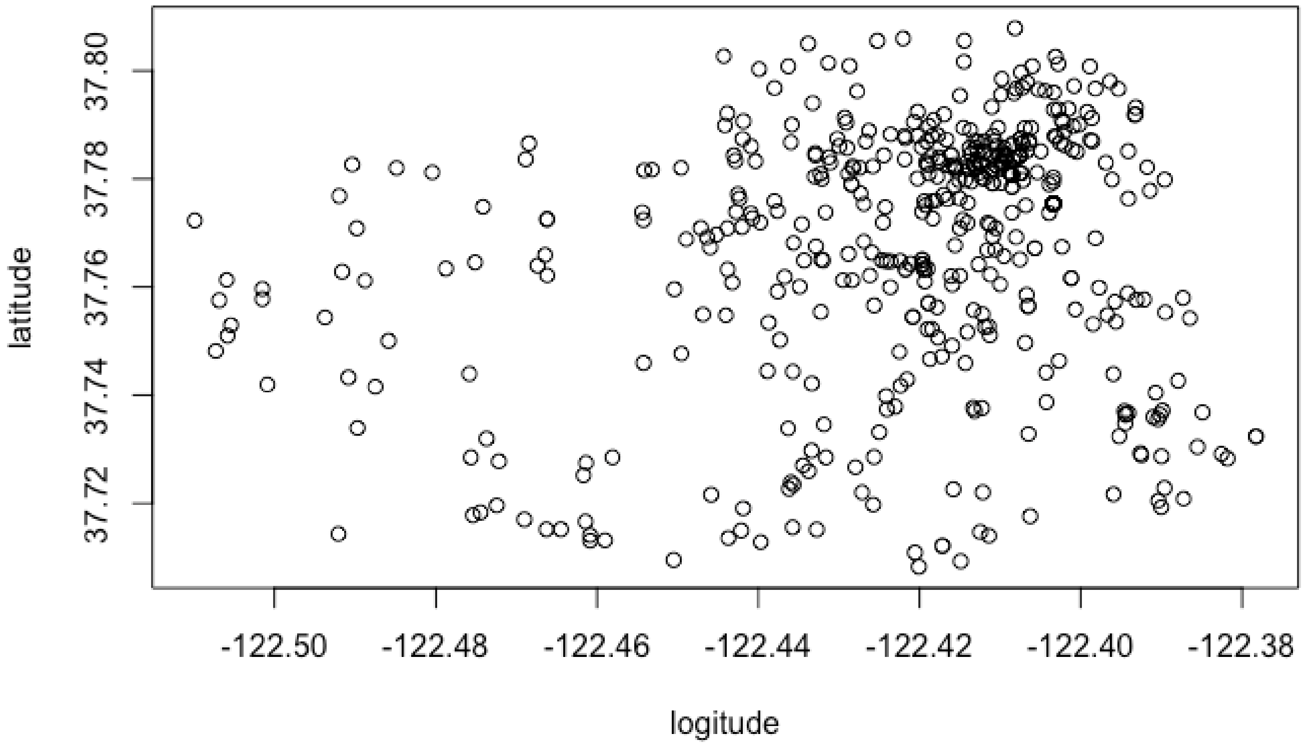
```
unknown timezone 'zone/tz/2018c.1.0/zoneinfo/America/Chicago'
```

Hide

```
# create new variable 'Year'
data$Year = as.factor(year(data$Dates))
# create new variable 'Month'
data$Month = as.factor(month(data$Dates))
# create new variable 'Day'
data$Day = as.factor(day(data$Dates))
# create new variable 'Hour'
data$Hour = as.factor(hour(data$Dates))
# remove the variable: 'Dates' after splitting it
data = data[-1]
```

Hide

```
# create sample observation
makeSample <- function(m, seed = 999){
  set.seed(seed)
  sample_entries <- sample( 1 : nrow(data), size = m, replace = FALSE )
  subset <- data[sample_entries,]
  subset
  }
temp.data = makeSample(500)
plot(temp.data$X, temp.data$Y, xlab = "logitude", ylab = "latitude")
```



Hide

```
# Supervised Model
# using step AIC , we can see that the location variables and the intercept are used
 to predict properly
full.model = glm(Category ~ ., data = temp.data, family = binomial)
```

```
glm.fit: algorithm did not convergeglm.fit: fitted probabilities numerically 0 or 1 o
ccurred
```

Hide

```
null.model = glm(Category ~ 1, data = temp.data, family = binomial)
variable.selection = step(null.model, formula(full.model), direction = "forward")
```

```
Start:  AIC=16.43
Category ~ 1

             Df Deviance    AIC
+ Y           1  11.0426 15.043
<none>           14.4272 16.427
+ X           1  14.3878 18.388
+ DayOfWeek   6  10.2381 24.238
+ PdDistrict  9   9.5909 29.591
+ Month      11   9.4990 33.499
+ Year       12   9.8039 35.804
+ Hour       23   8.1359 56.136
+ Day        30   7.7241 69.724

Step:  AIC=15.04
Category ~ Y
```

```
glm.fit: fitted probabilities numerically 0 or 1 occurredglm.fit: fitted probabilitie
s numerically 0 or 1 occurredglm.fit: fitted probabilities numerically 0 or 1 occurre
dglm.fit: algorithm did not convergeglm.fit: fitted probabilities numerically 0 or 1
occurred
```

```
             Df Deviance    AIC
<none>           11.0426 15.043
+ X           1  11.0039 17.004
+ DayOfWeek   6   7.1558 23.156
+ PdDistrict  9   9.4117 31.412
+ Month      11   5.7856 31.786
+ Year       12   7.1309 35.131
+ Hour       23   0.0000 50.000
+ Day        30   4.8575 68.858
```

Hide

```
variable.selection[1]
```

```
$coefficients
(Intercept)           Y
 -3380.7321     89.7418
```

Hide

```
install.packages("data.table", dependencies=TRUE)
```

```
Warning in install.packages :
  dependency 'GenomicRanges' is not available
```

```
trying URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.4/data.table_1.1
0.4-3.tgz'
Content type 'application/x-gzip' length 1430738 bytes (1.4 MB)
==================================================
downloaded 1.4 MB

tar: Failed to set default locale
```

```


The downloaded binary packages are in
    /var/folders/p0/zg1lp0211bj9rkh4_4nwbk780000gn/T//Rtmp2VbSJB/downloaded_packages
```

Hide

```
# Logistic regression
# Split the data in '7:3' ratio for training and testing respectively.
# 'proc.time()' function is used to evalute the efficiency of the model.
library(nnet)
crime = makeSample(m, seed = 33)
# split for training, validation and test set
split.train <- round(0.7 * m)                 # training set ends here
crime.train <- crime[1 : split.train,]        # training set (70% of data)
crime.test <- crime[(split.train + 1) : m,]   # test set (rest 30% of data)
# logistic regression model
log.time <- proc.time()
log.model <- multinom(formula = Category ~ DayOfWeek + Month +
                                X + Y + X ^ 2 + Y ^ 2,
                                maxit = 1000,
                                Hessian = FALSE,
                                data = crime.train)
```

```
groups 'BRIBERY' 'PORNOGRAPHY/OBSCENE MAT' 'TREA' are empty
```

```
# weights:  756 (700 variable)
initial  value 22178.398710
iter  10 value 18284.249625
iter  20 value 18216.374306
iter  30 value 18068.301368
iter  40 value 17322.951329
iter  50 value 16700.841175
iter  60 value 16334.886950
iter  70 value 16248.798947
iter  80 value 16201.378446
iter  90 value 16170.233638
iter 100 value 16154.837670
iter 110 value 16141.907523
iter 120 value 16135.576957
iter 130 value 16131.313146
iter 140 value 16129.252924
iter 150 value 16127.327461
iter 160 value 16125.821260
iter 170 value 16124.883929
iter 180 value 16124.391614
iter 190 value 16123.791822
iter 200 value 16123.146445
iter 210 value 16122.641032
iter 220 value 16122.182712
iter 230 value 16121.799530
iter 240 value 16121.262511
iter 250 value 16120.696055
iter 260 value 16120.183683
iter 270 value 16119.562716
iter 280 value 16118.938656
iter 290 value 16118.276399
iter 300 value 16117.803555
iter 310 value 16117.514092
iter 320 value 16117.292742
iter 330 value 16117.065803
iter 340 value 16116.712552
iter 350 value 16116.292423
iter 360 value 16115.807247
iter 370 value 16114.533993
iter 380 value 16113.279782
iter 390 value 16109.531664
iter 400 value 16106.692588
iter 410 value 16104.262625
iter 420 value 16101.141275
iter 430 value 16099.522240
iter 440 value 16098.686594
iter 450 value 16098.012956
iter 460 value 16096.483548
iter 470 value 16094.085558
iter 480 value 16092.054142
iter 490 value 16091.153051
iter 500 value 16090.911318
iter 510 value 16090.612856
iter 520 value 16090.276909
iter 530 value 16090.179501
iter 540 value 16089.677754
iter 550 value 16088.916423
iter 560 value 16087.780715
iter 570 value 16087.265040
iter 580 value 16085.894477
iter 590 value 16083.896628
iter 600 value 16078.738430
iter 610 value 16075.840205
iter 620 value 16075.446614
iter 630 value 16074.901923
iter 640 value 16074.517526
iter 650 value 16074.349717
iter 660 value 16073.980944
iter 670 value 16071.870748
final  value 16071.869425 
converged
```

Hide

```
log.time <- proc.time() - log.time
log.result <- predict(log.model, crime.test[, -1]) # prediction on test data
log.accuracy <- sum(log.result == t(crime.test[, -1])) # checking for out-of-sample performance
cat("The model took ", log.time[3], " seconds to generate\n",
    "Out of ", dim(crime.test)[1], " test cases, it got", log.accuracy ," right")
```

```
The model took  32.975  seconds to generate
 Out of  2653  test cases, it got 0  right
```

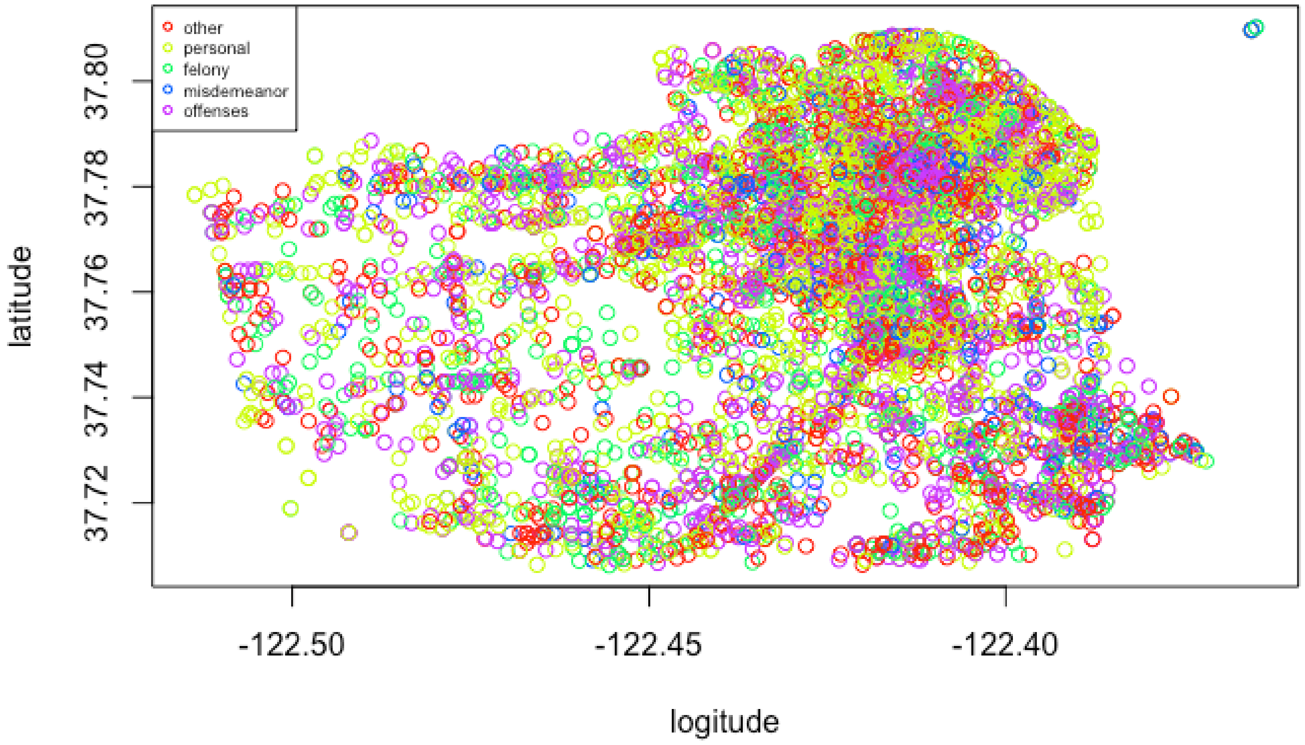
Hide

```
# Unsupervised
# The two approaches used are principal component analysis in combination with k-means clustering.
# For meaningful visualisation, the levels of `Category` are further classified into 5 major groups.
library(data.table) # For data representation and manipulation
```

```
package 'data.table' was built under R version 3.4.2data.table 1.10.4.3
  The fastest way to learn (by data.table authors): https://www.datacamp.com/courses/data-analysis-the-data-table-way
  Documentation: ?data.table, example(data.table) and browseVignettes("data.table")
  Release notes, videos and slides: http://r-datatable.com

Attaching package: 'data.table'

The following objects are masked from 'package:lubridate':

    hour, isoweek, mday, minute, month, quarter, second, wday, week, yday, year

The following objects are masked from 'package:dplyr':

    between, first, last
```

Hide

```
temp = crime
# handpick levels of `Category` and make a new group
felony <- as.factor(unique(data$Category)[c(7, 8, 10, 14, 23, 26, 37)])
personal <- as.factor(unique(data$Category)[c(3, 4, 15, 27, 33, 34)])
misdemeanor <- as.factor(unique(data$Category)[c(5, 11, 13, 18, 19, 20, 21, 24, 30, 3
6)])
offenses <- as.factor(unique(data$Category)[c(1, 9, 12, 16, 22, 28, 32, 37, 39 )])
other <- as.factor(unique(data$Category)[c(2, 6, 17, 25, 29, 31, 35, 38)])
temp = data.table(crime)
# all the values of each group are transformed into the name of the group
temp[Category %in% felony, newCategory := as.factor("felony") ]
temp[Category %in% personal, newCategory := as.factor("personal") ]
temp[Category %in% misdemeanor, newCategory := as.factor("misdemeanor") ]
temp[Category %in% offenses, newCategory := as.factor("offenses") ]
temp[Category %in% other, newCategory := as.factor("other") ]
temp$newCategory = as.factor(temp$newCategory)
few.color = rainbow(5)
plot(temp$X, temp$Y,
     col = few.color[temp$newCategory],
     xlab = "logitude", ylab = "latitude")
legend("topleft", as.character(unique(temp$newCategory)),
       pch = 1, col = few.color, cex = 0.55)
```



Hide

```
summary(temp$newCategory)
```

```
     felony    personal misdemeanor    offenses       other 
       2004        2342        1419         697        2380 
```

Hide

```
# PCA and K-means
# preparing data for principal component analysis
un.data = crime
# feature scaling
un.data$X = scale(un.data$X)
un.data$Y = scale(un.data$Y)
un.data$Year = scale(as.numeric(un.data$Year))
un.data$Month = scale(as.numeric(un.data$Month))
un.data$Day = scale(as.numeric(un.data$Day))
un.data$Hour = scale(as.numeric(un.data$Hour))
# splitting again
un.train <- un.data[1 : split.train,]        # training set (70% of data)
un.test <- un.data[(split.train + 1) : m,]   # test set (rest 30% of data)
```

Hide

```
install.packages('phyclust')
```

```
trying URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.4/phyclust_0.1-2
2.tgz'
Content type 'application/x-gzip' length 1315966 bytes (1.3 MB)
==================================================
downloaded 1.3 MB

tar: Failed to set default locale
```

```


The downloaded binary packages are in
	/var/folders/p0/zg1lp0211bj9rkh4_4nwbk780000gn/T//Rtmp2VbSJB/downloaded_packages
```

Hide

```
# For unsupervised analysis, first, the 5 new categories are mapped on a 2 dimensiona
l space.
# Then clusters are generated and applied on the same 2 dimensional space on a differ
ent plot.
library(phyclust)
```

```
package 'phyclust' was built under R version 3.4.3Loading required package: ape
package 'ape' was built under R version 3.4.2
Attaching package: 'phyclust'

The following object is masked from 'package:lubridate':

    ms
```

Hide

```
# PCA requires numeric values
# As it turns out, preprocessed data from neural network can be easily used
un.data = un.data[, 4:9]
un.data$Category = temp$newCategory
# Unsupervised models
pca.model <- princomp(un.data[,1:6])
clust.model <- kmeans(un.data[,1:6], centers = 5)
# Space for 2 plots
par(mfrow = c(2, 1),
    oma = c(3, 0, 3, 0),
    mar = c(1, 0, 0, 0))
plot(pca.model$scores[, 1:2], type = "n", axes = FALSE)
points(pca.model$scores[, 1:2],
       col = few.color[un.data$Category])
```

Hide

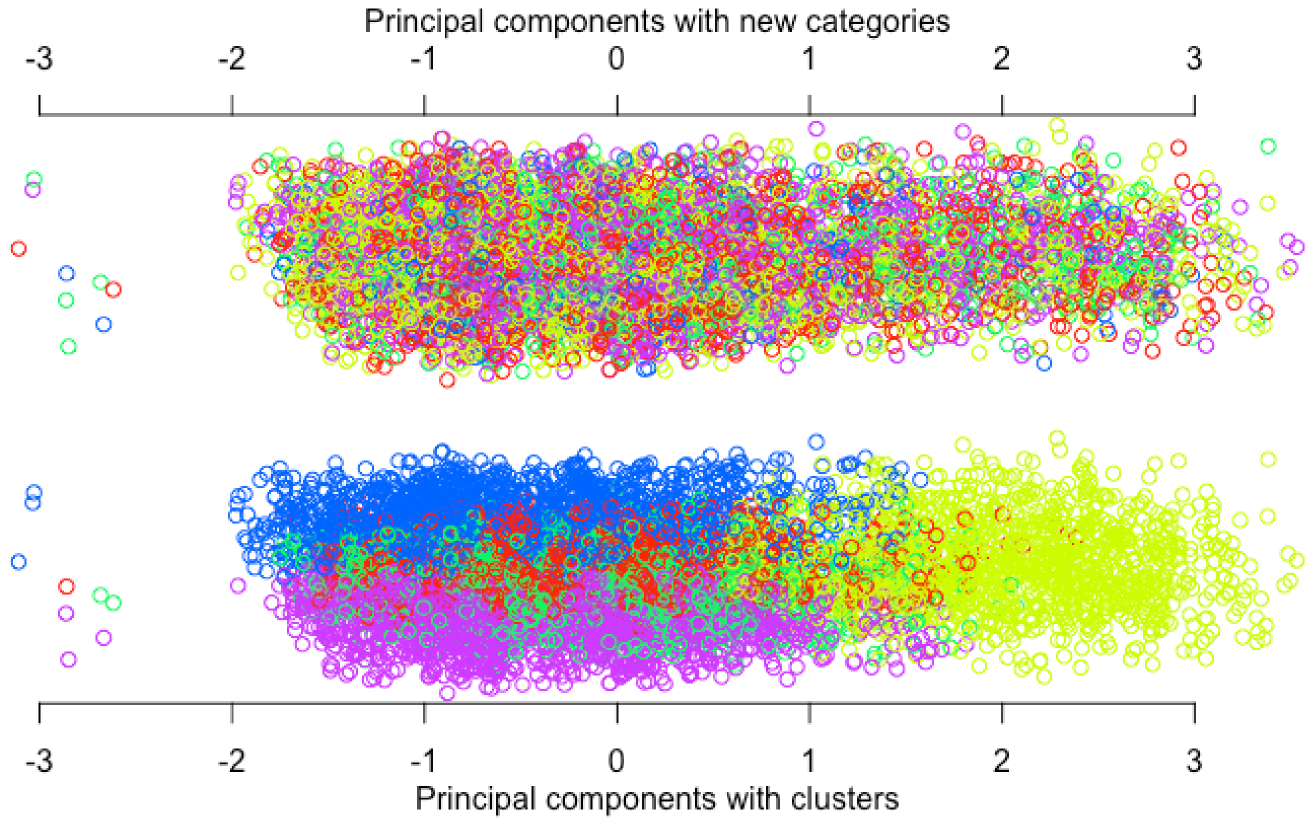
```
axis(3)
mtext("Principal components with new categories",
      side = 3, line = 2)
```

Hide

```
plot(pca.model$scores[, 1:2], type = "n", axes = FALSE)
points(pca.model$scores[, 1:2],
       col = few.color[clust.model$cluster])
```

Hide

```
axis(1)
mtext("Principal components with clusters",
      side = 1, line = 2)
```



Hide

```
RRand(as.numeric(un.data$Category), clust.model$cluster)
```

```
     Rand   adjRand    Eindex 
0.6625403 0.0003113 0.0920216 
```