Harvard|Choose Your Own Project| Swedish Data Crime

Edy Susanto

Table of Contents

Part 1 Introduction

Crime analysis and prevention is a systematic approach for identifying and analyzing patterns and trends in crime. Our system can predict regions which have high probability for crime occurrence and can visualize crime prone areas.

Part 2 Project Goal

This project will mainly focus on creating a Classification Machine Learning System using Swedish Data Crime. This data set contains statistics on reported crimes in Sweden (by 100.000) from 1950 to 2015. It contains the following columns:

crimes.total: total number of reported crimes crimes.penal.code: total number of reported crimes against the criminal code crimes.person: total number of reported crimes against a person murder: total number of reported murder sexual.offences: total number of reported sexual offences rape: total number of reported rapes assault: total number of reported aggravated assaults stealing.general: total number of reported crimes involving stealing or robbery robbery: total number of reported armed robberies burglary: total number of reported armed burglaries vehicle.theft: total number of reported vehicle thefts house.theft: total number of reported theft inside a house shop.theft: total number of reported theft inside a shop out.of.vehicle.theft: total number of reported theft from a vehicle criminal.damage: total number of reported criminal damages other.penal.crimes: number of other penal crime offenses fraud: total number of reported frauds narcotics: total number of reported narcotics abuses drunk.driving: total number of reported drunk driving incidents Year: the year population: the total estimated population of Sweden at the time

Link for the datasets is https://www.kaggle.com/mguzmann/swedishcrime

Part 3.Load Requirement Packages

Load all packages dan all libraries into RStudio

Part 2 Load Dataset

```
##   Year crimes.total crimes.penal.code crimes.person murder assault
## 1 1950         2784              2306           120      1     105
## 2 1951         3284              2754           125      1     109
## 3 1952         3160              2608           119      1     104
## 4 1953         2909              2689           119      1     105
## 5 1954         3028              2791           126      1     107
## 6 1955         3357              3101           135      1     118
##   sexual.offenses rape stealing.general burglary house.theft vehicle.
theft
## 1              40    5             1578      295          NA
    NA
## 2              45    6             1899      342          NA
    NA
## 3              39    4             1846      372          NA
    NA
## 4              45    5             1929      361          NA
    NA
## 5              41    5             1981      393          NA
    NA
## 6              44    5             2254      459          NA
    NA
##   out.of.vehicle.theft shop.theft robbery fraud criminal.damage
## 1                   NA         NA       3   209              72
## 2                   NA         NA       3   310              73
## 3                   NA         NA       3   217              82
## 4                   NA         NA       4   209              88
## 5                   NA         NA       4   236             101
## 6                   NA         NA       4   236             111
##   other.penal.crimes narcotics drunk.driving population
## 1                477         0            49    7014000
## 2                530         0            66    7073000
## 3                553         0            78    7125000
## 4                220         0            91    7171000
## 5                237         0           103    7213000
## 6                255         0           125    7262000
```

Part 4.Data exploration and visualization

The dataset is a data table made of 21 (columns) and a total of observations (67 rows).

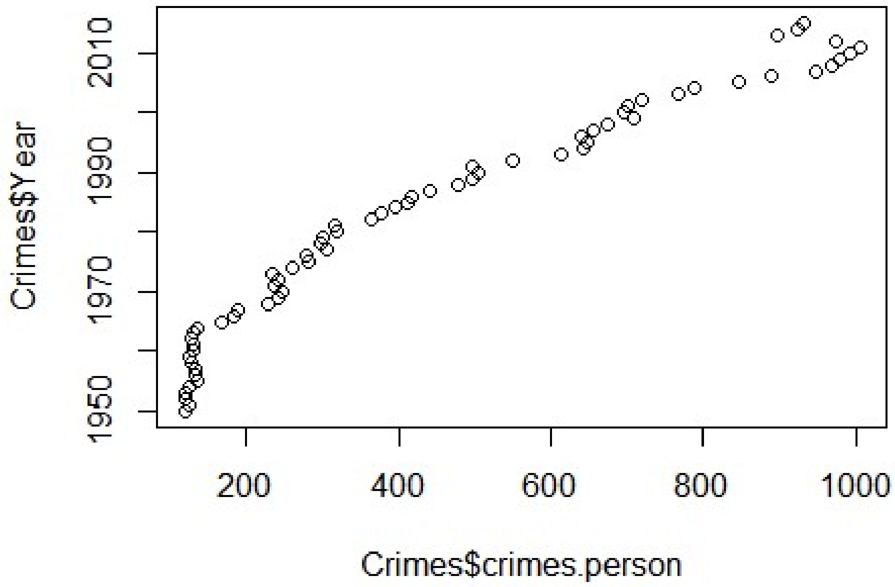
```
## # A tibble: 66 x 21
##     Year crimes.total crimes.penal.code crimes.person murder assault
##    <int>        <int>             <int>         <int>  <int>   <int>
##  1  1950         2784              2306           120      1     105
##  2  1951         3284              2754           125      1     109
##  3  1952         3160              2608           119      1     104
##  4  1953         2909              2689           119      1     105
##  5  1954         3028              2791           126      1     107
##  6  1955         3357              3101           135      1     118
##  7  1956         3488              3215           133      1     116
##  8  1957         3774              3520           133      1     116
##  9  1958         4064              3791           127      1     113
## 10  1959         4033              3733           125      1     110
## # ... with 56 more rows, and 15 more variables: sexual.offenses <in
t>,
## #   rape <int>, stealing.general <int>, burglary <int>, house.theft
<int>,
## #   vehicle.theft <int>, out.of.vehicle.theft <int>, shop.theft <in
t>,
## #   robbery <int>, fraud <int>, criminal.damage <int>,
## #   other.penal.crimes <int>, narcotics <int>, drunk.driving <int>,
## #   population <int>
```

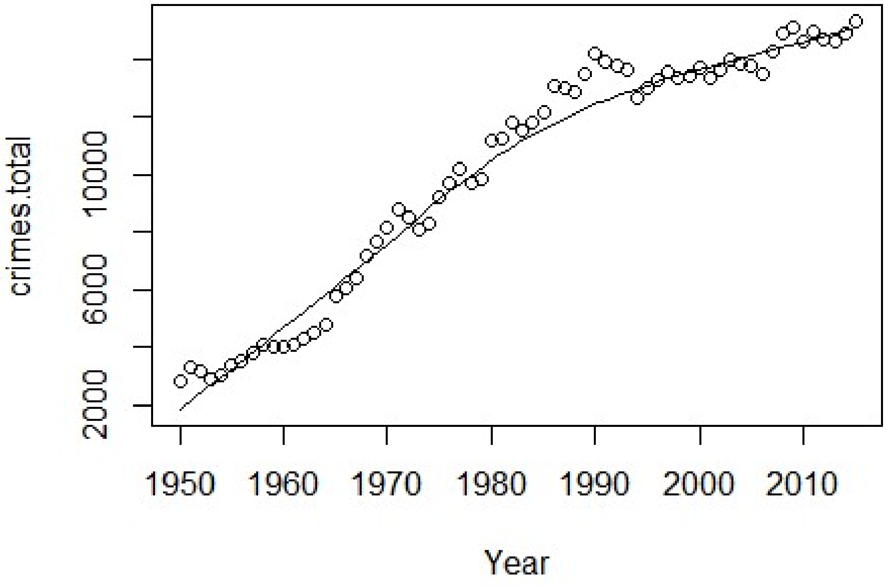
Number of NA into the dataset:

```
##                 Year         crimes.total    crimes.penal.code
##                    0                    0                    0
##        crimes.person               murder              assault
##                    0                    0                    0
##      sexual.offenses                 rape     stealing.general
##                    0                    0                    0
##             burglary          house.theft        vehicle.theft
##                    0                   15                    7
## out.of.vehicle.theft           shop.theft              robbery
##                   15                   15                    0
##                fraud      criminal.damage   other.penal.crimes
##                    0                    0                    0
##            narcotics        drunk.driving           population
##                    4                    0                    0
```

There No Missing Value on Data Set

Show Proportion Crimes Data On Plot





We can see that there are total crimes in Swedish is increase by Year

Part 5 Pre Data Processing

Principal Component Analysis(PCA)

We can get variable importance without using a predictive model using information theory, ordered from highest to lowest:

```
variable_importance = var_rank_info(Crimes, "Year")
```

```
## Warning: `funs()` was deprecated in dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warnin
g was generated.
```

```
## Warning in KL.plugin(freqs2d, freqs.null, unit = unit): Vanishing va
lue(s) in
## argument freqs2!
```

```
## Warning in KL.plugin(freqs2d, freqs.null, unit = unit): Vanishing va
lue(s) in
## argument freqs2!
```

```
## Warning in KL.plugin(freqs2d, freqs.null, unit = unit): Vanishing va
lue(s) in
## argument freqs2!
```

```
## Warning in KL.plugin(freqs2d, freqs.null, unit = unit): Vanishing va
lue(s) in
## argument freqs2!
```

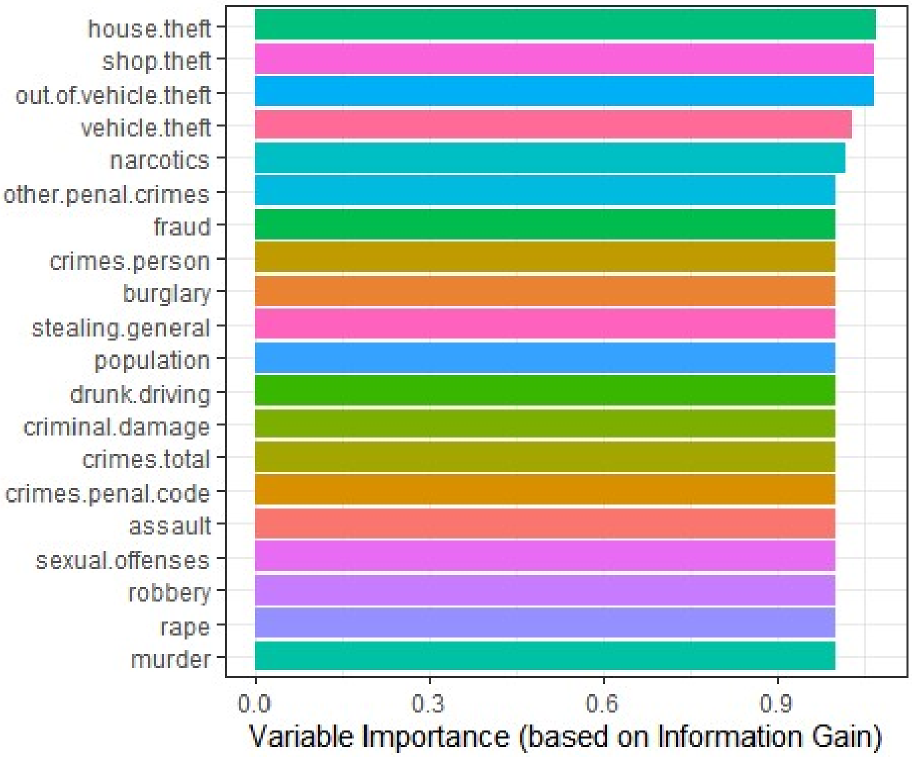
```
## Warning in KL.plugin(freqs2d, freqs.null, unit = unit): Vanishing va
lue(s) in
## argument freqs2!
```

```
variable_importance
```

```
##                        var    en    mi       ig       gr
## en9           house.theft 5.672 5.383 5.755083 1.069099
## en12           shop.theft 5.672 5.516 5.887531 1.067440
## en11 out.of.vehicle.theft 5.672 5.672 6.044394 1.065575
## en10        vehicle.theft 5.883 5.679 5.841004 1.028481
## en17            narcotics 5.954 5.438 5.528265 1.016586
## en2         crimes.person 6.044 5.893 5.892879 1.000000
## en8              burglary 6.044 5.953 5.953485 1.000000
## en14                fraud 6.044 5.923 5.923182 1.000000
## en16   other.penal.crimes 6.044 5.953 5.953485 1.000000
## en           crimes.total 6.044 6.044 6.044394 1.000000
## en1     crimes.penal.code 6.044 6.014 6.014091 1.000000
## en4               assault 6.044 5.832 5.832273 1.000000
## en7      stealing.general 6.044 6.044 6.044394 1.000000
## en15      criminal.damage 6.044 6.014 6.014091 1.000000
## en18        drunk.driving 6.044 5.863 5.862576 1.000000
## en19           population 6.044 6.044 6.044394 1.000000
## en5       sexual.offenses 6.044 5.368 5.368194 1.000000
## en6                  rape 6.044 4.517 4.517212 1.000000
## en13              robbery 6.044 5.266 5.265847 1.000000
## en3                murder 6.044 1.502 1.502098 1.000000
```

```
ggplot(variable_importance, aes(x = reorder(var, gr), y = gr, fill = va
r)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  theme_bw() +
  xlab("") +
  ylab("Variable Importance (based on Information Gain)") +
  guides(fill = FALSE)
```

```
## Warning: `guides(<scale> = FALSE)` is deprecated. Please use `guides
(<scale> =
## "none")` instead.
```



As we can see in the graphic, the displacement variable is the most important for our predictive model. We can see that the most crimes happen in Swedish is House Theft

```
set.seed(1)
pca <- prcomp(Crimes %>% select(Year), scale = TRUE, center = TRUE)

str(pca)

## List of 5
##  $ sdev    : num 1
##  $ rotation: num [1, 1] 1
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr "Year"
##   .. ..$ : chr "PC1"
##  $ center  : Named num 1982
##   ..- attr(*, "names")= chr "Year"
##  $ scale   : Named num 19.2
##   ..- attr(*, "names")= chr "Year"
##  $ x       : num [1:66, 1] -1.69 -1.64 -1.59 -1.54 -1.48 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : NULL
##   .. ..$ : chr "PC1"
##  - attr(*, "class")= chr "prcomp"

summary(pca)
```

```
## Importance of components:
##                         PC1
## Standard deviation       1
## Proportion of Variance   1
## Cumulative Proportion    1
```

```
set.seed(1)
# set.seed(1, sample.kind="Rounding") if using R 3.5.3 or later

test_index <- createDataPartition(y = Crimes$Year,
                                  times = 1, p = 0.2, list = FALSE)
edx <- Crimes[-test_index,]
validation <- Crimes[test_index,]

#We will split edx data into train_set and test_set.

set.seed(1)
test_index <- createDataPartition(y = edx$Year,
                                  times = 1, p = 0.2,
                                  list = FALSE)  # test_set 20%
train_set <- edx[-test_index,]
test_set <- edx[test_index,]
```

Part 6 Building Model

```
###
models <- c("glm", "lda", "naive_bayes", "svmLinear",
            "gamLoess", "qda", "knn", "kknn",
            "gam", "rf", "ranger", "wsrf", "mlp")

control <- trainControl(method = "cv",   # cross validation
                        number = 10,      # 10 k-folds or number
                                  # of resampling iterations
                        repeats = 5)

## Warning: `repeats` has no meaning for this resampling method.

data_train <- train_set       # first value for data parameter
data_test <-  test_set        # first we´ll use train and test dataset
true_value <- test_set$Year # true outcome from test_set
```

Part 7 Prediction

```
#####
model <- train(Year ~ crimes.total,
               data = Crimes,
               method = "lm")

model

## Linear Regression
##
```

```
## 66 samples
##  1 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 66, 66, 66, 66, 66, 66, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   5.389604  0.9234798  4.508358
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

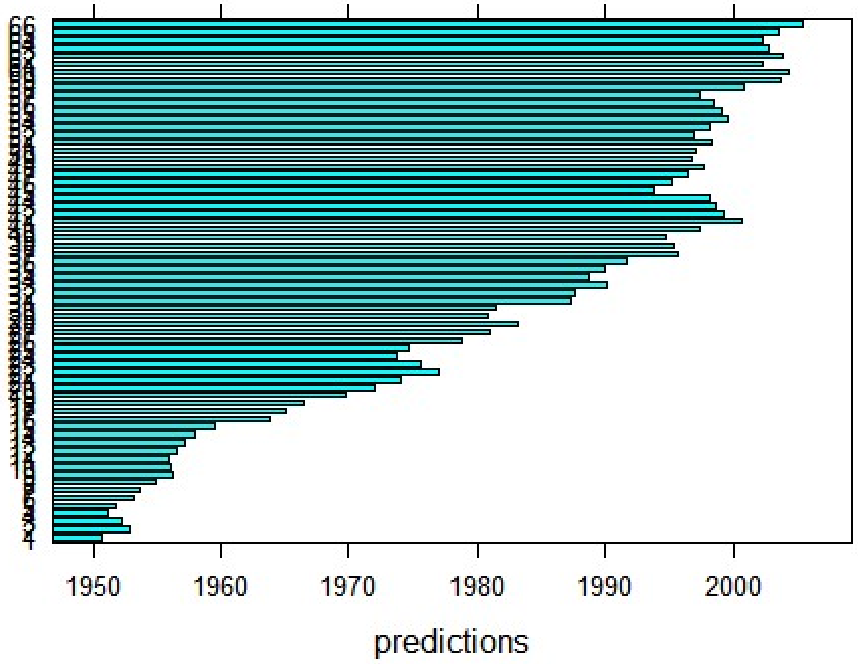
```
fitControl <- trainControl(method = "repeatedcv",
                           number = 10,     # number of folds
                           repeats = 5)    # repeated five times

model.cv <- train(Year ~ crimes.total,
               data = Crimes,
               method = "lm",  # now we're using the lm method
               trControl = fitControl)

model.cv
```

```
## Linear Regression
##
## 66 samples
##  1 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 5 times)
## Summary of sample sizes: 59, 58, 60, 61, 60, 60, ...
## Resampling results:
##
##   RMSE      Rsquared   MAE
##   5.356584  0.9253796  4.548874
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

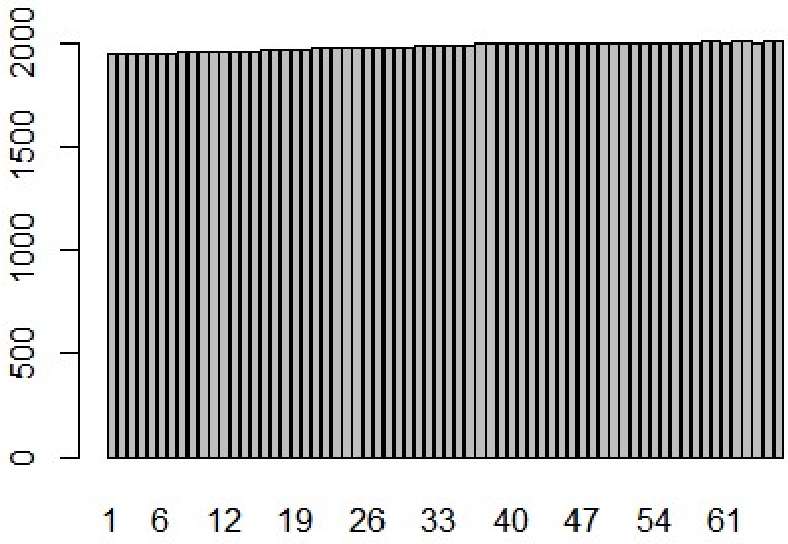
```
predictions <- predict(model.cv, Crimes)

predictions
```

```
##        1        2        3        4        5        6        7
  8
## 1950.668 1952.848 1952.307 1951.213 1951.732 1953.166 1953.737 1954.
984
##        9       10       11       12       13       14       15
 16
## 1956.249 1956.114 1955.891 1956.441 1957.117 1958.002 1959.454 1963.
```

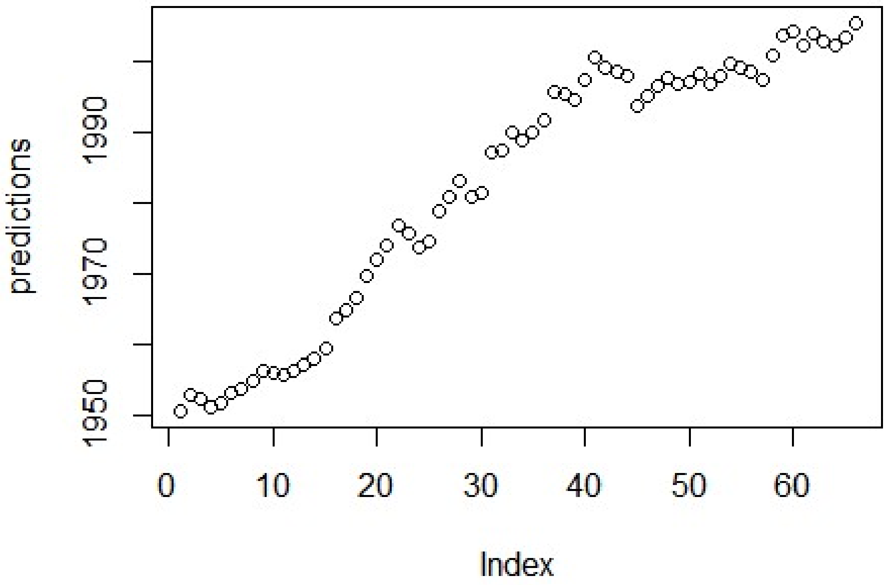
```
823
##       17       18       19       20       21       22       23
 24
## 1964.966 1966.527 1969.775 1971.977 1974.096 1977.005 1975.631 1973.
652
##       25       26       27       28       29       30       31
 32
## 1974.611 1978.736 1980.912 1983.149 1980.851 1981.435 1987.234 1987.
570
##       33       34       35       36       37       38       39
 40
## 1990.056 1988.739 1989.951 1991.704 1995.593 1995.301 1994.669 1997.
303
##       41       42       43       44       45       46       47
 48
## 2000.621 1999.204 1998.650 1998.109 1993.775 1995.135 1996.496 1997.
630
##       49       50       51       52       53       54       55
 56
## 1996.714 1997.037 1998.240 1996.827 1998.105 1999.561 1999.073 1998.
497
##       57       58       59       60       61       62       63
 64
## 1997.351 2000.795 2003.664 2004.375 2002.212 2003.882 2002.775 2002.
204
##       65       66
## 2003.455 2005.426

results <- sort(predictions)
barchart(predictions)
```



```
barplot.default(predictions)
```



```
plot.default(predictions)
```



## Part 8

Conclusion

Our Model Has succesfully made with RMSE 5.38960 , which is valid for prediction in SWedish Data Crime.