Лабораторна робота №4.Класифікація

студентки групи ЕП-61

Лузанової О.С.

Мета роботи – провести процедуру класифікації різними методами, а саме: Logistic regression, Support Vector Machine (SVM), K-Nearest Neighbors (K-NN), Naive Bayes, Classification Tree, Random Forest.

Після виконання роботи студент повинен:

ЗНАТИ сутність класифікації

УМІТИ організувати та провести зібр даних, кодування, обчислення та інтерпретування статистик, видалити викиди та провести шкалювання даних, обробити відсутні значення – як етап підготовки даних для класифікації та безпосередньо саму класифікацію.

**Класифікація –** встановлення функціональної залежності між вхідними і дискретними вихідними змінними. За допомогою класифікації вирішується завдання приналежності об’єктів до одного зі заздалегідь відомих класів.

Для виконання Лабораторної роботи №4 було обрано тему – збір даних та класифікація рівня злочинності та економічних факторів. Дані зібрані по країнах за 2017 рік.

Щоб виявити економічні фактори, що впливають на кількість вбивст, скоєних у країні, було проведено аналіз наукової літератури. Обрано такі показники, як:

1. Кількість населення (Population).
2. ВВП на душу населення у доларах США (GDP\_p)
3. Рівень безробіття (Unem).
4. Ймовірна тривалість життя (Life\_ex).
5. Рівень смертності (Death).
6. Рівень народжувальності (Bir).
7. % зайнятих віком від 15 років (Emp).
8. % зайнятих у сфері послуг (Emp\_.s)
9. % зайнятих у сфері с/г (Emp\_a).
10. % зайнятих у сфері виробництва (Emp\_i).
11. Державні видатки у сферу освіти (edu\_ex).
12. Державні видатки у сферу збройних сил (Mil\_ex).

lab4\_luzanova\_CLASSIFICATION

#“LOGISTIC REGRESSION” # Download the data

set.seed(123)  
f <- read.csv2('crime\_index.csv', header = TRUE, encoding = 'UNICOD')  
f <- f[,-c(1,2)]  
head(f)

## cr\_in Population gdp\_p unem life\_ex dea bir emp emp\_s emp\_a emp\_i  
## 1 73.85 36296400 556.30 11.18 64.13 6.58 33.21 43.22 37.99 43.99 18.03  
## 2 44.27 2873457 4531.02 13.75 78.33 7.71 11.93 48.22 42.44 38.20 19.35  
## 3 49.63 41389198 4044.28 12.00 76.50 4.72 24.85 36.42 58.85 10.16 30.99  
## 4 65.51 29816748 4095.81 7.12 60.38 8.43 41.28 72.05 41.07 50.56 8.37  
## 5 62.63 44044811 14591.86 8.35 76.37 7.62 17.21 55.53 77.50 0.06 22.44  
## 6 27.08 2944809 3914.50 17.70 74.80 9.86 14.30 45.80 51.95 31.34 16.71  
## edu\_ex mil\_ex  
## 1 530307037 191407113  
## 2 431032412 144382689  
## 3 7361823737 10073364021  
## 4 4091869979 3062872914  
## 5 31276833378 5459643672  
## 6 273378561 443610413

#Розглянемо статистику та переведемо кількісну змінну до бінарних значень, де 1 будуть приймати значення більші за середнє значення, 0 - менші за середнє значення.

library (psych)

## Warning: package 'psych' was built under R version 3.6.3

f$cr\_in<- ifelse(f$cr\_in > mean(f$cr\_in), 1, 0)  
head (f)

## cr\_in Population gdp\_p unem life\_ex dea bir emp emp\_s emp\_a emp\_i  
## 1 1 36296400 556.30 11.18 64.13 6.58 33.21 43.22 37.99 43.99 18.03  
## 2 0 2873457 4531.02 13.75 78.33 7.71 11.93 48.22 42.44 38.20 19.35  
## 3 1 41389198 4044.28 12.00 76.50 4.72 24.85 36.42 58.85 10.16 30.99  
## 4 1 29816748 4095.81 7.12 60.38 8.43 41.28 72.05 41.07 50.56 8.37  
## 5 1 44044811 14591.86 8.35 76.37 7.62 17.21 55.53 77.50 0.06 22.44  
## 6 0 2944809 3914.50 17.70 74.80 9.86 14.30 45.80 51.95 31.34 16.71  
## edu\_ex mil\_ex  
## 1 530307037 191407113  
## 2 431032412 144382689  
## 3 7361823737 10073364021  
## 4 4091869979 3062872914  
## 5 31276833378 5459643672  
## 6 273378561 443610413

describe(f)

## vars n mean sd median trimmed  
## cr\_in 1 123 5.000000e-01 5.000000e-01 0.000000e+00 4.900000e-01  
## Population 2 123 5.579362e+07 1.771883e+08 1.075468e+07 2.292767e+07  
## gdp\_p 3 119 1.893301e+04 2.101956e+04 9.540630e+03 1.534589e+04  
## unem 4 123 7.460000e+00 5.200000e+00 5.760000e+00 6.740000e+00  
## life\_ex 5 123 7.501000e+01 6.360000e+00 7.583000e+01 7.570000e+01  
## dea 6 123 7.730000e+00 2.950000e+00 7.110000e+00 7.570000e+00  
## bir 7 123 1.682000e+01 8.180000e+00 1.430000e+01 1.565000e+01  
## emp 8 123 5.695000e+01 1.119000e+01 5.802000e+01 5.661000e+01  
## emp\_s 9 123 6.019000e+01 1.634000e+01 6.216000e+01 6.145000e+01  
## emp\_a 10 123 1.818000e+01 1.884000e+01 1.070000e+01 1.504000e+01  
## emp\_i 11 123 2.164000e+01 7.440000e+00 2.048000e+01 2.153000e+01  
## edu\_ex 12 116 2.677892e+10 8.743026e+10 4.101771e+09 1.062876e+10  
## mil\_ex 13 112 1.498054e+10 6.182370e+10 2.202939e+09 4.172089e+09  
## mad min max range skew kurtosis  
## cr\_in 0.000000e+00 0.00 1.000000e+00 1.000000e+00 0.02 -2.02  
## Population 1.306531e+07 343400.00 1.386395e+09 1.386052e+09 6.57 45.27  
## gdp\_p 1.050704e+04 556.30 1.076271e+05 1.070709e+05 1.59 2.31  
## unem 3.850000e+00 0.14 2.707000e+01 2.693000e+01 1.31 1.51  
## life\_ex 6.000000e+00 53.95 8.468000e+01 3.073000e+01 -0.95 0.69  
## dea 2.950000e+00 1.17 1.550000e+01 1.433000e+01 0.48 -0.05  
## bir 6.230000e+00 7.00 4.192000e+01 3.492000e+01 1.17 0.73  
## emp 9.270000e+00 33.41 8.671000e+01 5.330000e+01 0.20 -0.18  
## emp\_s 1.661000e+01 13.03 8.794000e+01 7.491000e+01 -0.63 -0.19  
## emp\_a 1.259000e+01 0.06 8.330000e+01 8.324000e+01 1.32 1.13  
## emp\_i 6.260000e+00 3.67 5.451000e+01 5.084000e+01 0.57 2.24  
## edu\_ex 5.279548e+09 191532327.90 8.720000e+11 8.718085e+11 7.97 72.65  
## mil\_ex 2.868804e+09 23823068.18 6.060000e+11 6.059762e+11 8.18 72.85  
## se  
## cr\_in 5.000000e-02  
## Population 1.597653e+07  
## gdp\_p 1.926860e+03  
## unem 4.700000e-01  
## life\_ex 5.700000e-01  
## dea 2.700000e-01  
## bir 7.400000e-01  
## emp 1.010000e+00  
## emp\_s 1.470000e+00  
## emp\_a 1.700000e+00  
## emp\_i 6.700000e-01  
## edu\_ex 8.117696e+09  
## mil\_ex 5.841791e+09

#Висновок: кількість спостережень – 123, кількість змінних – 13, всі кількісні. Є пропущені значеня та викиди

#Заповнимо пропущені значення - середніми значеннями

f\_fill <- f  
f\_fill$gdp\_p <- ifelse(is.na(f$gdp\_p),round(mean(f$gdp\_p,na.rm = TRUE)),f$gdp\_p)  
f\_fill$edu\_ex <- ifelse(is.na(f$edu\_ex),round(mean(f$edu\_ex,na.rm = TRUE)),f$edu\_ex)  
f\_fill$mil\_ex <- ifelse(is.na(f$mil\_ex),round(mean(f$mil\_ex,na.rm = TRUE)),f$mil\_ex)  
f <- f\_fill  
head(f\_fill)

## cr\_in Population gdp\_p unem life\_ex dea bir emp emp\_s emp\_a emp\_i  
## 1 1 36296400 556.30 11.18 64.13 6.58 33.21 43.22 37.99 43.99 18.03  
## 2 0 2873457 4531.02 13.75 78.33 7.71 11.93 48.22 42.44 38.20 19.35  
## 3 1 41389198 4044.28 12.00 76.50 4.72 24.85 36.42 58.85 10.16 30.99  
## 4 1 29816748 4095.81 7.12 60.38 8.43 41.28 72.05 41.07 50.56 8.37  
## 5 1 44044811 14591.86 8.35 76.37 7.62 17.21 55.53 77.50 0.06 22.44  
## 6 0 2944809 3914.50 17.70 74.80 9.86 14.30 45.80 51.95 31.34 16.71  
## edu\_ex mil\_ex  
## 1 530307037 191407113  
## 2 431032412 144382689  
## 3 7361823737 10073364021  
## 4 4091869979 3062872914  
## 5 31276833378 5459643672  
## 6 273378561 443610413

describe(f)

## vars n mean sd median trimmed  
## cr\_in 1 123 5.000000e-01 5.000000e-01 0.000000e+00 4.900000e-01  
## Population 2 123 5.579362e+07 1.771883e+08 1.075468e+07 2.292767e+07  
## gdp\_p 3 123 1.893301e+04 2.067211e+04 1.025423e+04 1.528928e+04  
## unem 4 123 7.460000e+00 5.200000e+00 5.760000e+00 6.740000e+00  
## life\_ex 5 123 7.501000e+01 6.360000e+00 7.583000e+01 7.570000e+01  
## dea 6 123 7.730000e+00 2.950000e+00 7.110000e+00 7.570000e+00  
## bir 7 123 1.682000e+01 8.180000e+00 1.430000e+01 1.565000e+01  
## emp 8 123 5.695000e+01 1.119000e+01 5.802000e+01 5.661000e+01  
## emp\_s 9 123 6.019000e+01 1.634000e+01 6.216000e+01 6.145000e+01  
## emp\_a 10 123 1.818000e+01 1.884000e+01 1.070000e+01 1.504000e+01  
## emp\_i 11 123 2.164000e+01 7.440000e+00 2.048000e+01 2.153000e+01  
## edu\_ex 12 123 2.677892e+10 8.488497e+10 4.501641e+09 1.132612e+10  
## mil\_ex 13 123 1.498054e+10 5.897074e+10 3.062873e+09 5.176146e+09  
## mad min max range skew kurtosis  
## cr\_in 0.000000e+00 0.00 1.000000e+00 1.000000e+00 0.02 -2.02  
## Population 1.306531e+07 343400.00 1.386395e+09 1.386052e+09 6.57 45.27  
## gdp\_p 1.128785e+04 556.30 1.076271e+05 1.070709e+05 1.61 2.50  
## unem 3.850000e+00 0.14 2.707000e+01 2.693000e+01 1.31 1.51  
## life\_ex 6.000000e+00 53.95 8.468000e+01 3.073000e+01 -0.95 0.69  
## dea 2.950000e+00 1.17 1.550000e+01 1.433000e+01 0.48 -0.05  
## bir 6.230000e+00 7.00 4.192000e+01 3.492000e+01 1.17 0.73  
## emp 9.270000e+00 33.41 8.671000e+01 5.330000e+01 0.20 -0.18  
## emp\_s 1.661000e+01 13.03 8.794000e+01 7.491000e+01 -0.63 -0.19  
## emp\_a 1.259000e+01 0.06 8.330000e+01 8.324000e+01 1.32 1.13  
## emp\_i 6.260000e+00 3.67 5.451000e+01 5.084000e+01 0.57 2.24  
## edu\_ex 6.003692e+09 191532327.90 8.720000e+11 8.718085e+11 8.22 77.29  
## mil\_ex 4.030544e+09 23823068.18 6.060000e+11 6.059762e+11 8.58 80.43  
## se  
## cr\_in 5.000000e-02  
## Population 1.597653e+07  
## gdp\_p 1.863940e+03  
## unem 4.700000e-01  
## life\_ex 5.700000e-01  
## dea 2.700000e-01  
## bir 7.400000e-01  
## emp 1.010000e+00  
## emp\_s 1.470000e+00  
## emp\_a 1.700000e+00  
## emp\_i 6.700000e-01  
## edu\_ex 7.653820e+09  
## mil\_ex 5.317213e+09

#Пропущені значення заповнено

#Замінемо значення, що приймаємо за викиди, граничними значеннями максимума та мінімума

qn = quantile(f$Population, c(0.05, 0.95), na.rm = TRUE)  
f = within(f, { Population= ifelse(Population < qn[1], qn[1], Population)  
 Population = ifelse(Population > qn[2], qn[2], Population)})  
qn = quantile(f$gdp\_p, c(0.05, 0.95), na.rm = TRUE)  
f = within(f, { gdp\_p= ifelse(gdp\_p < qn[1], qn[1], gdp\_p)  
 gdp\_p = ifelse(gdp\_p > qn[2], qn[2], gdp\_p)})  
qn = quantile(f$unem, c(0.05, 0.95), na.rm = TRUE)  
f = within(f, { unem = ifelse(unem < qn[1], qn[1], unem)  
 unem = ifelse(unem > qn[2], qn[2], unem)})  
qn = quantile(f$life\_ex, c(0.05, 0.95), na.rm = TRUE)  
f = within(f, { life\_ex = ifelse(life\_ex < qn[1], qn[1], life\_ex)  
 life\_ex = ifelse(life\_ex > qn[2], qn[2], life\_ex)})  
qn = quantile(f$dea, c(0.05, 0.95), na.rm = TRUE)  
f = within(f, { dea= ifelse(dea < qn[1], qn[1], dea)  
 dea = ifelse(dea > qn[2], qn[2], dea)})  
qn = quantile(f$bir, c(0.05, 0.95), na.rm = TRUE)  
f = within(f, { bir= ifelse(bir < qn[1], qn[1], bir)  
 bir = ifelse(bir > qn[2], qn[2], bir)})  
qn = quantile(f$emp, c(0.05, 0.95), na.rm = TRUE)  
f = within(f, { emp= ifelse(emp < qn[1], qn[1], emp)  
 emp = ifelse(emp> qn[2], qn[2], emp)})  
qn = quantile(f$emp\_s, c(0.05, 0.95), na.rm = TRUE)  
f = within(f, { emp\_s= ifelse(emp\_s < qn[1], qn[1], emp\_s)  
 emp\_s = ifelse(emp\_s > qn[2], qn[2], emp\_s)})  
qn = quantile(f$emp\_a, c(0.05, 0.95), na.rm = TRUE)  
f = within(f, { emp\_a= ifelse(emp\_a < qn[1], qn[1], emp\_a)  
 emp\_a = ifelse(emp\_a > qn[2], qn[2], emp\_a)})  
qn = quantile(f$emp\_i, c(0.05, 0.95), na.rm = TRUE)  
f = within(f, { emp\_i= ifelse(emp\_i < qn[1], qn[1], emp\_i)  
 emp\_i = ifelse(emp\_i > qn[2], qn[2], emp\_i)})  
qn = quantile(f$edu\_ex, c(0.05, 0.95), na.rm = TRUE)  
f = within(f, { edu\_ex= ifelse(edu\_ex < qn[1], qn[1], edu\_ex)  
 edu\_ex = ifelse(edu\_ex > qn[2], qn[2], edu\_ex)})  
qn = quantile(f$mil\_ex, c(0.05, 0.95), na.rm = TRUE)  
f = within(f, { mil\_ex= ifelse(mil\_ex < qn[1], qn[1], mil\_ex)  
 mil\_ex = ifelse(mil\_ex > qn[2], qn[2], mil\_ex)})  
  
describe(f)

## vars n mean sd median trimmed  
## cr\_in 1 123 5.000000e-01 5.000000e-01 0.000000e+00 4.900000e-01  
## Population 2 123 3.460812e+07 4.958875e+07 1.075468e+07 2.292767e+07  
## gdp\_p 3 123 1.813714e+04 1.821617e+04 1.025423e+04 1.528928e+04  
## unem 4 123 7.300000e+00 4.580000e+00 5.760000e+00 6.740000e+00  
## life\_ex 5 123 7.518000e+01 5.750000e+00 7.583000e+01 7.570000e+01  
## dea 6 123 7.730000e+00 2.690000e+00 7.110000e+00 7.570000e+00  
## bir 7 123 1.658000e+01 7.440000e+00 1.430000e+01 1.565000e+01  
## emp 8 123 5.689000e+01 1.047000e+01 5.802000e+01 5.661000e+01  
## emp\_s 9 123 6.039000e+01 1.530000e+01 6.216000e+01 6.145000e+01  
## emp\_a 10 123 1.767000e+01 1.738000e+01 1.070000e+01 1.504000e+01  
## emp\_i 11 123 2.147000e+01 6.420000e+00 2.048000e+01 2.153000e+01  
## edu\_ex 12 123 1.901020e+10 3.067557e+10 4.501641e+09 1.132612e+10  
## mil\_ex 13 123 8.330987e+09 1.273161e+10 3.062873e+09 5.176146e+09  
## mad min max range skew kurtosis  
## cr\_in 0.000000e+00 0.00 1.000000e+00 1.000000e+00 0.02 -2.02  
## Population 1.306531e+07 647881.60 1.877530e+08 1.871052e+08 1.99 3.09  
## gdp\_p 1.128785e+04 1393.23 6.081815e+04 5.942491e+04 1.12 -0.02  
## unem 3.850000e+00 1.82 1.768000e+01 1.586000e+01 0.92 -0.18  
## life\_ex 6.000000e+00 63.06 8.266000e+01 1.959000e+01 -0.63 -0.44  
## dea 2.950000e+00 3.49 1.327000e+01 9.780000e+00 0.50 -0.65  
## bir 6.230000e+00 8.30 3.317000e+01 2.487000e+01 0.89 -0.30  
## emp 9.270000e+00 39.52 7.781000e+01 3.829000e+01 0.10 -0.61  
## emp\_s 1.661000e+01 28.61 8.161000e+01 5.299000e+01 -0.47 -0.78  
## emp\_a 1.259000e+01 1.12 5.969000e+01 5.857000e+01 1.04 0.01  
## emp\_i 6.260000e+00 8.52 3.305000e+01 2.454000e+01 -0.02 -0.54  
## edu\_ex 6.003692e+09 386855546.29 1.196822e+11 1.192953e+11 2.22 4.19  
## mil\_ex 4.030544e+09 71955687.86 4.632868e+10 4.625672e+10 2.03 3.19  
## se  
## cr\_in 5.000000e-02  
## Population 4.471267e+06  
## gdp\_p 1.642500e+03  
## unem 4.100000e-01  
## life\_ex 5.200000e-01  
## dea 2.400000e-01  
## bir 6.700000e-01  
## emp 9.400000e-01  
## emp\_s 1.380000e+00  
## emp\_a 1.570000e+00  
## emp\_i 5.800000e-01  
## edu\_ex 2.765923e+09  
## mil\_ex 1.147971e+09

write.csv2(f, file = "crime\_index\_pre.csv")

#Позбавились викидів та пропущених значень.

# Features Scaling

mPopulation <- mean(f$Population)  
sPopulation <- sd(f$Population)  
f$Population <- (f$Population-mPopulation)/sPopulation  
  
mgdp\_p <- mean(f$gdp\_p)  
sgdp\_p <- sd(f$gdp\_p)  
f$gdp\_p<- (f$gdp\_p-mgdp\_p)/sgdp\_p  
  
munem <- mean(f$unem)  
sunem <- sd(f$unem)  
f$unem <- (f$unem-munem)/sunem  
  
mlife\_ex <- mean(f$life\_ex)  
slife\_ex <- sd(f$life\_ex)  
f$life\_ex <- (f$life\_ex-mlife\_ex)/slife\_ex  
  
mdea <- mean(f$dea)  
sdea <- sd(f$dea)  
f$dea<- (f$dea-mdea)/sdea  
  
mbir <- mean(f$bir)  
sbir <- sd(f$bir)  
f$bir <- (f$bir-mbir)/sbir  
  
memp <- mean(f$emp)  
semp <- sd(f$emp)  
f$emp <- (f$emp-memp)/semp  
  
memp\_s <- mean(f$emp\_s)  
semp\_s <- sd(f$emp\_s)  
f$emp\_s<- (f$emp\_s-memp\_s)/semp\_s  
  
memp\_a <- mean(f$emp\_a)  
semp\_a <- sd(f$emp\_a)  
f$emp\_a <- (f$emp\_a-memp\_a)/semp\_a  
  
memp\_i <- mean(f$emp\_i)  
semp\_i <- sd(f$emp\_i)  
f$emp\_i <- (f$emp\_i-memp\_i)/semp\_i  
  
medu\_ex <- mean(f$edu\_ex)  
sedu\_ex <- sd(f$edu\_ex)  
f$edu\_ex<- (f$edu\_ex-medu\_ex)/sedu\_ex  
  
mmil\_ex <- mean(f$mil\_ex)  
smil\_ex <- sd(f$mil\_ex)  
f$mil\_ex <- (f$mil\_ex-mmil\_ex)/smil\_ex  
  
head (f)

## cr\_in Population gdp\_p unem life\_ex dea bir  
## 1 1 0.03404554 -0.9191786 0.8472187 -1.92163256 -0.428923107 2.23006043  
## 2 0 -0.63995695 -0.7469257 1.4084049 0.54801921 -0.008656897 -0.62521158  
## 3 1 0.13674621 -0.7736459 1.0262742 0.22974718 -1.120688727 1.11185648  
## 4 1 -0.09662224 -0.7708171 -0.0393244 -2.10703037 0.259123343 2.23006043  
## 5 1 0.19029893 -0.1946230 0.2292589 0.20713769 -0.042129427 0.08467382  
## 6 0 -0.63851808 -0.7807704 2.2665612 -0.06591535 0.790964652 -0.30656984  
## emp emp\_s emp\_a emp\_i edu\_ex mil\_ex  
## 1 -1.3054994 -1.4645832 1.5143179 -0.5351602 -0.6024304 -0.6393203  
## 2 -0.8281562 -1.1736896 1.1812142 -0.3296526 -0.6056667 -0.6430138  
## 3 -1.6591152 -0.1009786 -0.4319510 1.4825507 -0.3797282 0.1368544  
## 4 1.4468612 -1.2632456 1.8922956 -2.0162161 -0.4863262 -0.4137821  
## 5 -0.1302805 1.1181597 -0.9523177 0.1514220 0.3998826 -0.2255286  
## 6 -1.0591903 -0.5520271 0.7865525 -0.7406678 -0.6071068 -0.6195111

#Висновок: моделі класифікації вимагають попереднього шкалювання кількісних змінних. Шкалювання виконано.

# Splitting the scaled dataset into the TRAIN set and TEST set

set.seed(123)  
library(caTools)

## Warning: package 'caTools' was built under R version 3.6.3

split = sample.split(f$cr\_in, SplitRatio = 0.7)  
f\_train\_c = subset(f, split == TRUE)  
f\_test\_c = subset(f, split == FALSE)

#Висновок: підготований датасет розділено на навчальну та тестову вибірки. До тренувальної вибірки обрано 86 значень (70% від усіх спостережень), до тестової - 37 значення (30% від усіх спостережень)

# Fitting (Benchmark model)

class\_lr <- glm(cr\_in ~ ., f\_train\_c, family = binomial)  
summary(class\_lr)

##   
## Call:  
## glm(formula = cr\_in ~ ., family = binomial, data = f\_train\_c)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.4860 -0.5117 -0.0028 0.3855 2.2671   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.1984 0.4232 0.469 0.63926   
## Population 1.8712 0.8299 2.255 0.02415 \*   
## gdp\_p -0.9677 0.7339 -1.319 0.18731   
## unem -0.7932 0.6785 -1.169 0.24239   
## life\_ex 0.4746 1.0555 0.450 0.65292   
## dea 0.9173 0.6522 1.406 0.15959   
## bir 3.6270 1.3661 2.655 0.00793 \*\*  
## emp 0.1782 0.5699 0.313 0.75453   
## emp\_s 3.3086 5.0076 0.661 0.50879   
## emp\_a 1.3677 5.5564 0.246 0.80557   
## emp\_i -0.3131 2.0057 -0.156 0.87596   
## edu\_ex -1.4889 1.0051 -1.481 0.13853   
## mil\_ex 0.2372 0.7067 0.336 0.73716   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 119.221 on 85 degrees of freedom  
## Residual deviance: 53.648 on 73 degrees of freedom  
## AIC: 79.648  
##   
## Number of Fisher Scoring iterations: 7

## Optimized model

class\_opt <- glm(cr\_in ~ edu\_ex + Population, f\_train\_c, family = binomial)  
summary(class\_opt)

##   
## Call:  
## glm(formula = cr\_in ~ edu\_ex + Population, family = binomial,   
## data = f\_train\_c)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -2.56058 -1.06829 -0.06459 1.04608 1.81096   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.07671 0.25251 0.304 0.76130   
## edu\_ex -1.59527 0.50400 -3.165 0.00155 \*\*  
## Population 1.77339 0.54412 3.259 0.00112 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 119.221 on 85 degrees of freedom  
## Residual deviance: 95.968 on 83 degrees of freedom  
## AIC: 101.97  
##   
## Number of Fisher Scoring iterations: 5

#обрано саме ці змінні на підставі аналізу двофакторної моделі. Саме ці дві змінні дають найбільш гарний результат

# Predicting

p <- predict(class\_opt, f\_test\_c[, c("edu\_ex","Population" )], type = 'response')  
y <- ifelse(p > 0.7, 1, 0)

#Висновок: розраховані ймовірності віднесення об’єктів до кожного з двох класів (вектор р), визначені класи об’єктів (вектор у).

## Confusion Matrix

cm = table(f\_test\_c[, 'cr\_in'], y > 0.7)  
print(cm)

##   
## FALSE TRUE  
## 0 17 2  
## 1 14 4

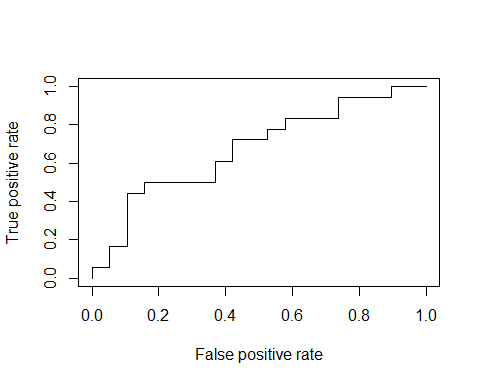
#Висновок: точність моделі - (17 + 4) / 37 = 56.8 %, частка невірно класифікованих випадків – (14 + 2) / 37 = 43,2 %. Чутливість моделі – 4 / (4 + 14) = 22.2 %, специфічність – 17 / (17 + 2) = 89.5 %, тобто модель більш чутлива до виявлення позитивних випадків. У цьому разі – “країн”, з високим рівнем злочинності.

## ROC

library(ROCR)

## Warning: package 'ROCR' was built under R version 3.6.3

pref <- prediction(p, f\_test\_c$cr\_in)  
perf <- performance(pref, "tpr", "fpr")  
plot(perf)



#Висновок: співвідношення істинно-позитивних і хибно-позитивних випадків свідчить про відносно погану якість моделі.

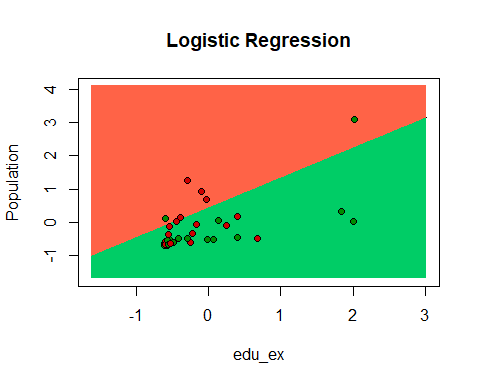
# Visualising the Test set results

library(ggplot2)

##   
## Attaching package: 'ggplot2'

## The following objects are masked from 'package:psych':  
##   
## %+%, alpha

set = f\_test\_c[,c('edu\_ex','Population','cr\_in')]  
X1 = seq(min(set['edu\_ex']) - 1, max(set['edu\_ex']) + 1, by = 0.01)  
X2 = seq(min(set['Population']) - 1, max(set['Population']) + 1, by = 0.01)  
grid\_set = expand.grid(X1, X2)  
colnames(grid\_set) = c('edu\_ex', 'Population')  
prob\_set = predict(class\_opt, grid\_set, type = 'response')  
y\_grid = ifelse(prob\_set > 0.7, 1, 0)  
plot(set[, -3],  
 main = 'Logistic Regression',  
 xlab = 'edu\_ex', ylab = 'Population',  
 xlim = range(X1), ylim = range(X2))  
contour(X1, X2, matrix(as.numeric(y\_grid), length(X1), length(X2)), add = TRUE)  
points(grid\_set, pch = '.', col = ifelse(y\_grid == 1, 'tomato', 'springgreen3'))  
points(set, pch = 21, bg = ifelse(set[, 3] == 1, 'red3', 'green4'))

 #Висновок: на графіку червоним позначені “країни” з високим рівнем злочинності, зеленим – з низьким рівнем злочинності.Червоним виділена зона високої ймовірності високого значення рівня злочинності. Модель описує лінійний варіант розділяючої кривої.

#“K-Nearest Neighbors (K-NN)”

# Fitting & predicting

library(class)  
y = knn(train = f\_train\_c[,c('edu\_ex','Population')],  
 test = f\_test\_c[,c('edu\_ex','Population')],  
 cl = f\_train\_c[, 'cr\_in'],  
 k = 5,  
 prob = TRUE)

#Висновок: і навчання, і прогнозування за моделлю k найближчих сусідів здійснюється однією функцією. У результаті отримуємо вектор класів об’єктів.

## Confusion Matrix

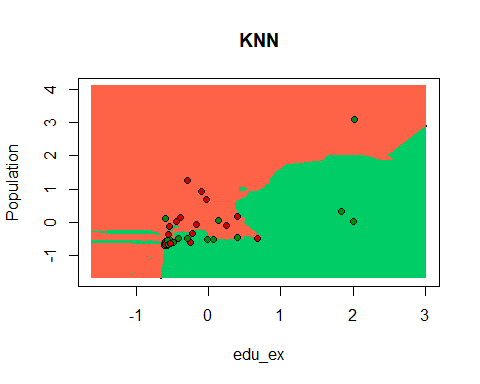
cm = table(f\_test\_c[, 'cr\_in'], y == '1')  
print(cm)

##   
## FALSE TRUE  
## 0 11 8  
## 1 5 13

#Висновок: точність моделі – (11+13) / 37 = 64.9 %, частка невірно класифікованих випадків – (5+8) / 37 = 35.1 %. Чутливість – 13 / (13+5) = 72.2 %, специфічність – 11 / (11+8) = 57,9 %, тобто модель більш чутлива до виявлення позитивних випадків. У цьому разі – “країн”, з високим рівнем злочинності

## Visualising the Test set results

library(ggplot2)  
set = f\_test\_c[,c('edu\_ex','Population','cr\_in')]  
X1 = seq(min(set['edu\_ex']) - 1, max(set['edu\_ex']) + 1, by = 0.01)  
X2 = seq(min(set['Population']) - 1, max(set['Population']) + 1, by = 0.01)  
grid\_set = expand.grid(X1, X2)  
colnames(grid\_set) = c('edu\_ex', 'Population')  
y\_grid = knn(train = f\_train\_c[,c('edu\_ex','Population')], test = grid\_set, cl = f\_train\_c[, 'cr\_in'], k = 5)  
plot(set[, -3],  
 main = 'KNN',  
 xlab = 'edu\_ex', ylab = 'Population',  
 xlim = range(X1), ylim = range(X2))  
contour(X1, X2, matrix(as.numeric(y\_grid), length(X1), length(X2)), add = TRUE)  
points(grid\_set, pch = '.', col = ifelse(y\_grid == 1, 'tomato', 'springgreen3'))  
points(set, pch = 21, bg = ifelse(set[, 3] == 1, 'red3', 'green4'))

 #Висновок: на графіку червоним позначені “країни” з високим рівнем злочинності, зеленим – з низьким рівнем злочинності.Червоним виділена зона високої ймовірності високого значення рівня злочинності. Модель описує нелінійний варіант розділяючої кривої.

#“Support Vector Machine (SVM)”

# Fitting SVM model

library(e1071)

## Warning: package 'e1071' was built under R version 3.6.3

class\_svm\_l = svm(cr\_in ~ edu\_ex + Population, data = f\_train\_c, kernel = 'linear')  
summary(class\_svm\_l)

##   
## Call:  
## svm(formula = cr\_in ~ edu\_ex + Population, data = f\_train\_c, kernel = "linear")  
##   
##   
## Parameters:  
## SVM-Type: eps-regression   
## SVM-Kernel: linear   
## cost: 1   
## gamma: 0.5   
## epsilon: 0.1   
##   
##   
## Number of Support Vectors: 82

# Predicting

p <- predict(class\_svm\_l, f\_test\_c[, c('edu\_ex','Population')])  
y <- ifelse(p > 0.5, 1, 0)

#Висновок: визначено класи об’єктів (вектор у).

## Confusion Matrix

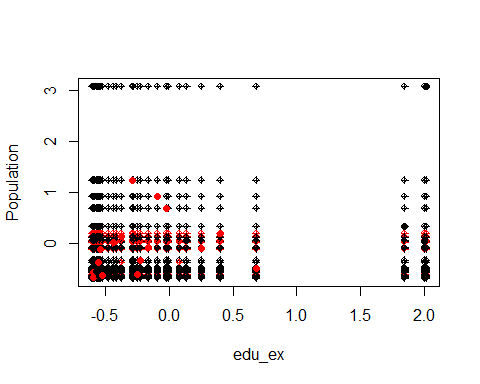
cm = table(f\_test\_c[, 'cr\_in'], y)  
print(cm)

## y  
## 0 1  
## 0 18 1  
## 1 16 2

#Висновок: точність моделі - (18 + 2) / 37 = 54.1 %, частка невірно класифікованих випадків – (16 + 1) / 37 = 45,9 %. Чутливість моделі – 2 / (16 + 2) = 11.1 %, специфічність – 18 / (18 + 1) = 94,7 %, тобто модель більш чутлива до виявлення негативних випадків. У цьому разі – “країн”, з низьким рівнем злочинності

# Visualising the Test set results

xgrid = expand.grid(edu\_ex = f\_test\_c$edu\_ex, Population = f\_test\_c$Population)  
ygrid = predict(class\_svm\_l, xgrid)  
#Finally, you plot the points and color them according to the decision boundary. You can see that the decision boundary is linear. You can put the data points in the plot as well to see where they lie.  
plot(xgrid, col = as.numeric(y\_grid), pch = 10, cex = .9)   
points(f\_test\_c[, c('edu\_ex','Population')], col = as.factor(f\_test\_c$cr\_in), pch = 19)



# Fitting RBF-kernel model

library(e1071)  
class\_svm\_r = svm(cr\_in ~ edu\_ex + Population, data = f\_train\_c, kernel = 'radial')  
summary(class\_svm\_r)

##   
## Call:  
## svm(formula = cr\_in ~ edu\_ex + Population, data = f\_train\_c, kernel = "radial")  
##   
##   
## Parameters:  
## SVM-Type: eps-regression   
## SVM-Kernel: radial   
## cost: 1   
## gamma: 0.5   
## epsilon: 0.1   
##   
##   
## Number of Support Vectors: 75

# Predicting

p <- predict(class\_svm\_r, f\_test\_c[, c('edu\_ex','Population')])  
y <- ifelse(p > 0.5, 1, 0)

#Висновок: визначені класи об’єктів (вектор у).

## Confusion Matrix

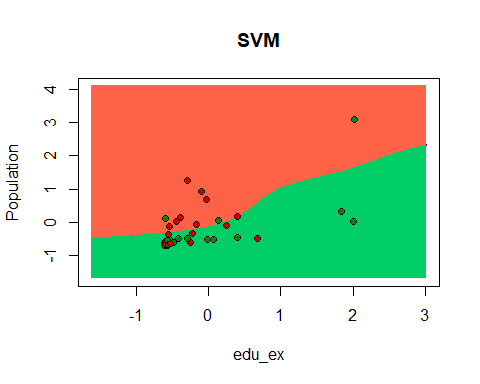
cm = table(f\_test\_c[, 'cr\_in'], y)  
print(cm)

## y  
## 0 1  
## 0 16 3  
## 1 10 8

#Висновок: точність моделі - (16 + 8) / 37 = 64.9 %, частка невірно класифікованих випадків – (10 + 3) / 37 = 35,1 %. Чутливість моделі – 8 / (8 + 10) = 44.4 %, специфічність – 16 / (16 + 8) = 66.7 %, тобто модель більш чутлива до виявлення негативних випадків. У цьому разі – “країн”, з низьким рівнем злочинності

# Visualising the Test set results

library(ggplot2)  
set = f\_test\_c[,c('edu\_ex','Population','cr\_in')]  
X1 = seq(min(set['edu\_ex']) - 1, max(set['edu\_ex']) + 1, by = 0.01)  
X2 = seq(min(set['Population']) - 1, max(set['Population']) + 1, by = 0.01)  
grid\_set = expand.grid(X1, X2)  
colnames(grid\_set) = c('edu\_ex', 'Population')  
p\_grid = predict(class\_svm\_r, grid\_set)  
y\_grid <- ifelse(p\_grid > 0.5, 1, 0)  
plot(set[, -3],  
 main = 'SVM',  
 xlab = 'edu\_ex', ylab = 'Population',  
 xlim = range(X1), ylim = range(X2))  
contour(X1, X2, matrix(as.numeric(y\_grid), length(X1), length(X2)), add = TRUE)  
points(grid\_set, pch = '.', col = ifelse(y\_grid == 1, 'tomato', 'springgreen3'))  
points(set, pch = 21, bg = ifelse(set[, 3] == 1, 'red3', 'green4'))

 #Висновок: на графіку червоним позначені “країни” з високим рівнем злочинності, зеленим – з низьким рівнем злочинності.Червоним виділена зона високої ймовірності високого значення рівня злочинності. Модель описує нелінійний варіант розділяючої кривої.

#“Naive Bayes” # Fitting

library(e1071)  
f\_train\_c$cr\_in <- as.factor(f\_train\_c$cr\_in)  
f\_test\_c$cr\_in <- as.factor(f\_test\_c$cr\_in)  
class\_nb = naiveBayes(cr\_in ~ edu\_ex + Population, data = f\_train\_c)

#Висновок: для навчання моделі використано функцію naiveBayes.

# Predicting

y <- predict(class\_nb, f\_test\_c[, c('edu\_ex','Population')])

#Висновок: визначено класи об’єктів (вектор у).

## Confusion Matrix

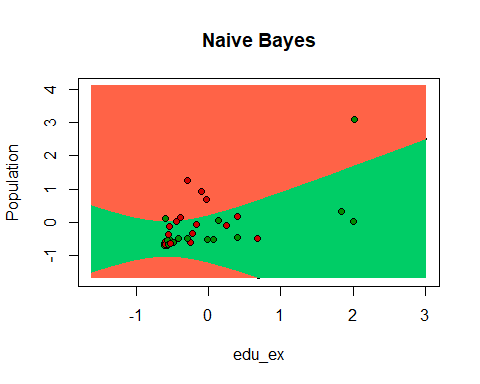
cm = table(f\_test\_c[, 'cr\_in'], y)   
print(cm)

## y  
## 0 1  
## 0 17 2  
## 1 14 4

#Висновок: точність моделі - (17 + 4) / 37 = 56.8 %, частка невірно класифікованих випадків – (14 + 2) / 37 = 43,2 %. Чутливість моделі – 4 / (4 + 14) = 22.2 %, специфічність – 17 / (17 + 2) = 89.5 %, тобто модель більш чутлива до виявлення пнегативних випадків. У цьому разі – “країн”, з низьким рівнем злочинності

# Visualising the Test set results

library(ggplot2)  
set = f\_test\_c[,c('edu\_ex','Population','cr\_in')]  
X1 = seq(min(set['edu\_ex']) - 1, max(set['edu\_ex']) + 1, by = 0.01)  
X2 = seq(min(set['Population']) - 1, max(set['Population']) + 1, by = 0.01)  
grid\_set = expand.grid(X1, X2)  
colnames(grid\_set) = c('edu\_ex', 'Population')  
y\_grid = predict(class\_nb, grid\_set)  
plot(set[, -3],  
 main = 'Naive Bayes',  
 xlab = 'edu\_ex', ylab = 'Population',  
 xlim = range(X1), ylim = range(X2))  
contour(X1, X2, matrix(as.numeric(y\_grid), length(X1), length(X2)), add = TRUE)  
points(grid\_set, pch = '.', col = ifelse(y\_grid == 1, 'tomato', 'springgreen3'))  
points(set, pch = 21, bg = ifelse(set[, 3] == 1, 'red3', 'green4'))

 #Висновок: на графіку червоним позначені “країни” з високим рівнем злочинності, зеленим – з низьким рівнем злочинності.Червоним виділена зона високої ймовірності високого значення рівня злочинності. Модель описує нелінійний варіант розділяючої кривої.

#“Classification Tree”

# Fitting

library(rpart)

## Warning: package 'rpart' was built under R version 3.6.3

f\_train\_c$cr\_in <- as.factor(f\_train\_c$cr\_in)  
f\_test\_c$cr\_in <- as.factor(f\_test\_c$cr\_in)  
class\_dt = rpart(cr\_in ~ ., data = f\_train\_c)

## Predicting

y <- predict(class\_dt, f\_test\_c[-14], type = 'class')

#Висновок: визначені класи об’єктів (вектор у).

## Confusion Matrix

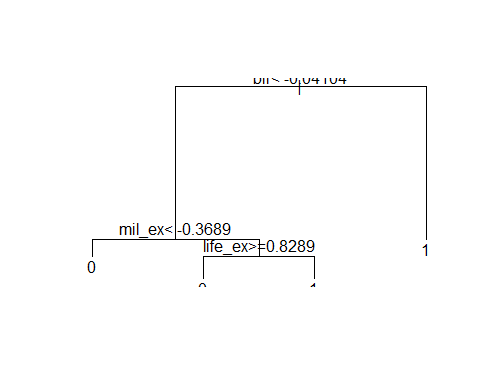
cm = table(f\_test\_c[, 'cr\_in'], y)  
print(cm)

## y  
## 0 1  
## 0 14 5  
## 1 4 14

#Висновок: точність моделі – (14+14) / 37 = 75.7 %, частка невірно класифікованих випадків – (4+5) / 37 = 24.3 %. Чутливість – 14 / (14+4) = 77.8 %, специфічність – 14 / (14+5) = 73.7 %, тобто модель більш чутлива до виявлення позитивних випадків. У цьому разі – “країн”, з високим рівнем злочинносі

# Plotting the tree

plot(class\_dt)  
text(class\_dt)

 #для аналізу необхідно перейти від шкальованих даних до реальних

# Fitting 2 factors

class\_ct = rpart(cr\_in ~ edu\_ex + Population, data = f\_train\_c)

## Predicting

y <- predict(class\_ct, f\_test\_c[, c('edu\_ex', 'Population')], type = 'class')

#Висновок: визначено класи об’єктів (вектор у). Для цього використано параметр type = ‘class’.

## Confusion Matrix

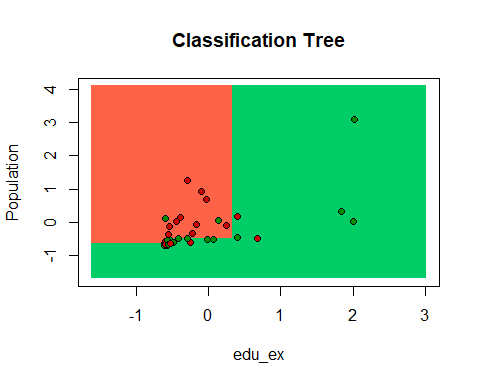
cm = table(f\_test\_c[, 'cr\_in'], y)  
print(cm)

## y  
## 0 1  
## 0 13 6  
## 1 7 11

#Висновок: точність моделі – (13+11) / 37 = 64.9 %, частка невірно класифікованих випадків – (7+6) / 37 = 35.1 %. Чутливість – 11 / (11+7) = 61.1 %, специфічність – 13/ (13+6) = 68.4 %, тобто модель більш чутлива до виявлення негативних випадків. У цьому разі – “країн”, з низьким рівнем злочинності

# Visualising the Test set results

library(ggplot2)  
set = f\_test\_c[,c('edu\_ex','Population','cr\_in')]  
X1 = seq(min(set['edu\_ex']) - 1, max(set['edu\_ex']) + 1, by = 0.01)  
X2 = seq(min(set['Population']) - 1, max(set['Population']) + 1, by = 0.01)  
grid\_set = expand.grid(X1, X2)  
colnames(grid\_set) = c('edu\_ex', 'Population')  
y\_grid = predict(class\_ct, grid\_set, type = 'class')  
plot(set[, -3],  
 main = 'Classification Tree',  
 xlab = 'edu\_ex', ylab = 'Population',  
 xlim = range(X1), ylim = range(X2))  
contour(X1, X2, matrix(as.numeric(y\_grid), length(X1), length(X2)), add = TRUE)  
points(grid\_set, pch = '.', col = ifelse(y\_grid == 1, 'tomato', 'springgreen3'))  
points(set, pch = 21, bg = ifelse(set[, 3] == 1, 'red3', 'green4'))

 #Висновок: на графіку червоним позначені “країни” з високим рівнем злочинності, зеленим – з низьким рівнем злочинності.Червоним виділена зона високої ймовірності високого значення рівня злочинності. Модель описує нелінійний варіант розділяючої кривої.

# Fitting Random Forest Classification to the Training set

library(randomForest)

## Warning: package 'randomForest' was built under R version 3.6.3

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:ggplot2':  
##   
## margin

## The following object is masked from 'package:psych':  
##   
## outlier

set.seed(123)  
class\_rf = randomForest(cr\_in ~ edu\_ex + Population, data = f\_train\_c, ntree = 10)

## Predicting

y <- predict(class\_rf, f\_test\_c[, c('edu\_ex','Population')])

#Висновок: визначені класи об’єктів (вектор у).

## Confusion Matrix

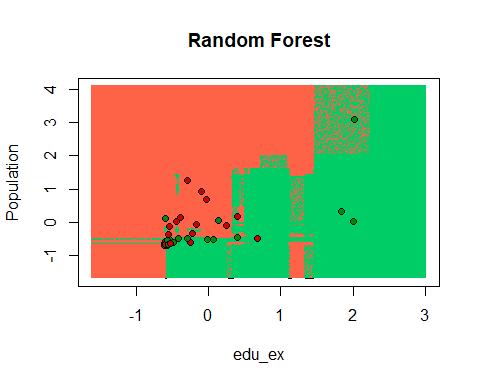
cm = table(f\_test\_c[, 'cr\_in'], y)  
print(cm)

## y  
## 0 1  
## 0 13 6  
## 1 5 13

#Висновок: точність моделі – (13+13) / 37 = 70.3 %, частка невірно класифікованих випадків – (5+6) / 37 = 29.7 %. Чутливість – 13 / (13+5) = 72.2 %, специфічність – 13 / (13+6) = 69,4 %, тобто модель більш чутлива до виявлення позитивних випадків. У цьому разі – “країн”, з високим рівнем злочинності

# Visualising the Test set results

set = f\_test\_c[,c('edu\_ex','Population','cr\_in')]  
X1 = seq(min(set['edu\_ex']) - 1, max(set['edu\_ex']) + 1, by = 0.01)  
X2 = seq(min(set['Population']) - 1, max(set['Population']) + 1, by = 0.01)  
grid\_set = expand.grid(X1, X2)  
colnames(grid\_set) = c('edu\_ex', 'Population')  
y\_grid = predict(class\_rf, grid\_set)  
plot(set[, -3],  
 main = 'Random Forest',  
 xlab = 'edu\_ex', ylab = 'Population',  
 xlim = range(X1), ylim = range(X2))  
contour(X1, X2, matrix(as.numeric(y\_grid), length(X1), length(X2)), add = TRUE)  
points(grid\_set, pch = '.', col = ifelse(y\_grid == 1, 'tomato', 'springgreen3'))  
points(set, pch = 21, bg = ifelse(set[, 3] == 1, 'red3', 'green4'))

 #Висновок: на графіку червоним позначені “країни” з високим рівнем злочинності, зеленим – з низьким рівнем злочинності.Червоним виділена зона високої ймовірності високого значення рівня злочинності. Модель описує нелінійний варіант розділяючої кривої.

#“NEUORAL NETWORKS FOR CLASSIFICATION”

library(neuralnet)

## Warning: package 'neuralnet' was built under R version 3.6.3

##   
## Attaching package: 'neuralnet'

## The following object is masked from 'package:ROCR':  
##   
## prediction

# fit neural network  
nn = neuralnet(cr\_in ~ edu\_ex + Population, f\_train\_c, hidden = 3, linear.output = T)  
# plot neural network  
plot(nn)

#Висновок: на основі змінних edu\_ex,Population побудовано двошарову нейронну мережу для прогнозування классу.

# Fitting the NN

library(nnet)

## Warning: package 'nnet' was built under R version 3.6.3

set.seed(11)  
ff\_cl <- nnet(data = f\_train\_c, cr\_in ~ edu\_ex + Population, size = 3, maxit = 1000)

## # weights: 13  
## initial value 62.488002   
## iter 10 value 46.984984  
## iter 20 value 45.800771  
## iter 30 value 42.812030  
## iter 40 value 41.600810  
## iter 50 value 40.700508  
## iter 60 value 40.347143  
## iter 70 value 40.229492  
## iter 80 value 40.220782  
## iter 90 value 40.218863  
## iter 100 value 40.204982  
## iter 110 value 40.195220  
## iter 120 value 40.190299  
## iter 130 value 40.184970  
## iter 140 value 40.183648  
## iter 150 value 40.177266  
## iter 160 value 40.173339  
## iter 170 value 40.171194  
## iter 180 value 40.168755  
## iter 190 value 40.166189  
## iter 190 value 40.166189  
## final value 40.166189   
## converged

library(graphics)  
source(file = 'plot.nnet.R')  
plot.nnet(ff\_cl)

## Loading required package: scales

##   
## Attaching package: 'scales'

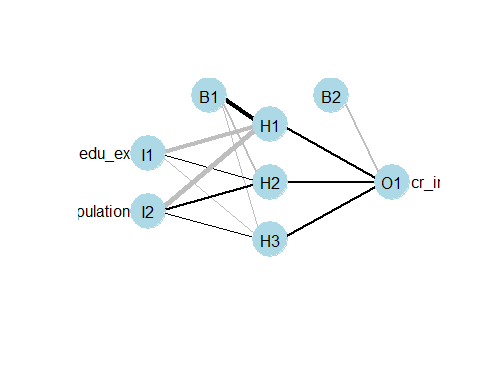
## The following objects are masked from 'package:psych':  
##   
## alpha, rescale

## Loading required package: reshape

## Warning: package 'reshape' was built under R version 3.6.3

##   
## Attaching package: 'reshape'

## The following object is masked from 'package:class':  
##   
## condense



## Predicting

#p\_nn\_cl <- predict(nn, f\_test\_c)  
#p <- (p > 0.5)  
#p <- as.factor(p\_ff\_cl)  
p\_ff\_cl <- predict(ff\_cl, f\_test\_c, type = "class")   
p <- as.factor(p\_ff\_cl)

#### Висновок: визначені класи об’єктів (вектор p)

## Confusion Matrix

cm = table(f\_test\_c[, 'cr\_in'], p)  
print(cm)

## p  
## 0 1  
## 0 16 3  
## 1 10 8

#Висновок: точність моделі – (16+8) / 37 = 64.9 %, частка невірно класифікованих випадків – (10+3) / 37 = 35.1 %. Чутливість – 8 / (10+8) = 44.4 %, специфічність – 16 / (16+3) = 84.2 %, тобто модель більш чутлива до виявлення негативних випадків. У цьому разі – “країн”, з низьким рівнем злочинності

#Отже, проведено класифікацію різними методами, серед яких: Logistic Regression, KNN, SVM, Naive Bayes, Classification Tree, Random Forest. За методом Classification Tree найменша кількість невірно віднесених значень - 9, при вибірці у 37 спостережень. Висновки по методу Classification Tree: тточність моделі – (14+14) / 37 = 75.7 %, частка невірно класифікованих випадків – (4+5) / 37 = 24.3 %. Чутливість – 14 / (14+4) = 77.8 %, специфічність – 14 / (14+5) = 73.7 %, тобто модель більш чутлива до виявлення позитивних випадків. У цьому разі – “країн”, з високим рівнем злочинносі