Analysis

Here is a summary of the trainset. In this dataset, we have two kinds of features/reponses: 1. common features (1-24) 2. product features (25-48)

```
options(warn=-1)
require(data.table, quietly = TRUE)
require(dplyr, quietly = TRUE)

#the reason that five features are read as character is that they are actually binary variables easy to
colclasses <- list(
  integer = c("ncodpers", "ind_nuevo", "antiguedad", "age",  "ind_actividad_cliente"),
  numeric = c("renta"),
  factor = c("ind_empleado", "pais_residencia", "indrel_1mes", "tiprel_1mes",  "canal_entrada", "cod_pr
  character = c("sexo", "indrel", "indresi", "indext", "indfall", "conyuemp"),
  date = c("fecha_dato", "ult_fec_cli_1t", "fecha_alta"))
naStrings <- c("", " ", "NA")
trainSet <- fread("../data/train.csv", verbose = FALSE, na.strings = naStrings)
```

```
##
Read 0.0% of 13647309 rows
Read 2.8% of 13647309 rows
Read 3.1% of 13647309 rows
Read 6.6% of 13647309 rows
Read 10.0% of 13647309 rows
Read 13.6% of 13647309 rows
Read 16.9% of 13647309 rows
Read 20.2% of 13647309 rows
Read 23.4% of 13647309 rows
Read 26.9% of 13647309 rows
Read 27.8% of 13647309 rows
Read 31.4% of 13647309 rows
Read 35.0% of 13647309 rows
Read 38.6% of 13647309 rows
Read 42.1% of 13647309 rows
Read 45.7% of 13647309 rows
Read 49.4% of 13647309 rows
Read 53.0% of 13647309 rows
Read 56.6% of 13647309 rows
Read 60.2% of 13647309 rows
Read 63.8% of 13647309 rows
Read 67.5% of 13647309 rows
Read 71.1% of 13647309 rows
Read 74.8% of 13647309 rows
Read 78.3% of 13647309 rows
Read 82.1% of 13647309 rows
Read 85.7% of 13647309 rows
Read 89.4% of 13647309 rows
Read 93.1% of 13647309 rows
Read 96.7% of 13647309 rows
Read 13647309 rows and 48 (of 48) columns from 1.963 GB file in 00:00:38
```

```r
nTrain <- nrow(trainSet)
featureDescription <- fread("../data/description.csv", verbose = FALSE)
summary(trainSet)
```

```
##   fecha_dato           ncodpers       ind_empleado
##  Length:13647309    Min.   :  15889   Length:13647309
##  Class :character   1st Qu.: 452813   Class :character
##  Mode  :character   Median : 931893   Mode  :character
##                     Mean   : 834904
##                     3rd Qu.:1199286
##                     Max.   :1553689
##
##  pais_residencia        sexo                age          fecha_alta
##  Length:13647309    Length:13647309    Min.   :  2.00   Length:13647309
##  Class :character   Class :character   1st Qu.: 24.00   Class :character
##  Mode  :character   Mode  :character   Median : 39.00   Mode  :character
##                                        Mean   : 40.18
##                                        3rd Qu.: 50.00
##                                        Max.   :164.00
##                                        NA's   :27734
##    ind_nuevo       antiguedad           indrel       ult_fec_cli_1t
##  Min.   :0.00    Min.   :-999999.0   Min.   : 1.000   Length:13647309
##  1st Qu.:0.00    1st Qu.:     23.0   1st Qu.: 1.000   Class :character
##  Median :0.00    Median :     50.0   Median : 1.000   Mode  :character
##  Mean   :0.06    Mean   :     76.6   Mean   : 1.178
##  3rd Qu.:0.00    3rd Qu.:    135.0   3rd Qu.: 1.000
##  Max.   :1.00    Max.   :    256.0   Max.   :99.000
##  NA's   :27734   NA's   :27734       NA's   :27734
##  indrel_1mes        tiprel_1mes          indresi
##  Length:13647309    Length:13647309    Length:13647309
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    indext             conyuemp         canal_entrada
##  Length:13647309    Length:13647309    Length:13647309
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    indfall              tipodom          cod_prov       nomprov
##  Length:13647309    Min.   :1       Min.   : 1.00   Length:13647309
##  Class :character   1st Qu.:1       1st Qu.:15.00   Class :character
##  Mode  :character   Median :1       Median :28.00   Mode  :character
##                     Mean   :1       Mean   :26.57
##                     3rd Qu.:1       3rd Qu.:35.00
##                     Max.   :1       Max.   :52.00
##                     NA's   :27735   NA's   :93591
##  ind_actividad_cliente     renta            segmento
##  Min.   :0.000         Min.   :    1203   Length:13647309
```

```
##  1st Qu.:0.000         1st Qu.:   68711   Class :character
##  Median :0.000         Median :  101850   Mode  :character
##  Mean   :0.458         Mean   :  134254
##  3rd Qu.:1.000         3rd Qu.:  155956
##  Max.   :1.000         Max.   :28894396
##  NA's   :27734         NA's   :2794375
##  ind_ahor_fin_ult1   ind_aval_fin_ult1  ind_cco_fin_ult1
##  Min.   :0.0000000   Min.   :0.00e+00   Min.   :0.0000
##  1st Qu.:0.0000000   1st Qu.:0.00e+00   1st Qu.:0.0000
##  Median :0.0000000   Median :0.00e+00   Median :1.0000
##  Mean   :0.0001023   Mean   :2.32e-05   Mean   :0.6555
##  3rd Qu.:0.0000000   3rd Qu.:0.00e+00   3rd Qu.:1.0000
##  Max.   :1.0000000   Max.   :1.00e+00   Max.   :1.0000
##
##  ind_cder_fin_ult1   ind_cno_fin_ult1  ind_ctju_fin_ult1
##  Min.   :0.0000000   Min.   :0.00000   Min.   :0.000000
##  1st Qu.:0.0000000   1st Qu.:0.00000   1st Qu.:0.000000
##  Median :0.0000000   Median :0.00000   Median :0.000000
##  Mean   :0.0003939   Mean   :0.08087   Mean   :0.009474
##  3rd Qu.:0.0000000   3rd Qu.:0.00000   3rd Qu.:0.000000
##  Max.   :1.0000000   Max.   :1.00000   Max.   :1.000000
##
##  ind_ctma_fin_ult1  ind_ctop_fin_ult1 ind_ctpp_fin_ult1 ind_deco_fin_ult1
##  Min.   :0.000000   Min.   :0.000     Min.   :0.00000   Min.   :0.000000
##  1st Qu.:0.000000   1st Qu.:0.000     1st Qu.:0.00000   1st Qu.:0.000000
##  Median :0.000000   Median :0.000     Median :0.00000   Median :0.000000
##  Mean   :0.009727   Mean   :0.129     Mean   :0.04331   Mean   :0.001779
##  3rd Qu.:0.000000   3rd Qu.:0.000     3rd Qu.:0.00000   3rd Qu.:0.000000
##  Max.   :1.000000   Max.   :1.000     Max.   :1.00000   Max.   :1.000000
##
##  ind_deme_fin_ult1  ind_dela_fin_ult1 ind_ecue_fin_ult1 ind_fond_fin_ult1
##  Min.   :0.000000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.000000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000
##  Median :0.000000   Median :0.00000   Median :0.00000   Median :0.00000
##  Mean   :0.001661   Mean   :0.04297   Mean   :0.08274   Mean   :0.01849
##  3rd Qu.:0.000000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000
##  Max.   :1.000000   Max.   :1.00000   Max.   :1.00000   Max.   :1.00000
##
##  ind_hip_fin_ult1   ind_plan_fin_ult1  ind_pres_fin_ult1
##  Min.   :0.000000   Min.   :0.000000   Min.   :0.000000
##  1st Qu.:0.000000   1st Qu.:0.000000   1st Qu.:0.000000
##  Median :0.000000   Median :0.000000   Median :0.000000
##  Mean   :0.005887   Mean   :0.009171   Mean   :0.002627
##  3rd Qu.:0.000000   3rd Qu.:0.000000   3rd Qu.:0.000000
##  Max.   :1.000000   Max.   :1.000000   Max.   :1.000000
##
##  ind_reca_fin_ult1 ind_tjcr_fin_ult1 ind_valo_fin_ult1 ind_viv_fin_ult1
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.000000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.000000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median :0.000000
##  Mean   :0.05254   Mean   :0.04439   Mean   :0.02561   Mean   :0.003848
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.000000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :1.000000
##
```

```
##  ind_nomina_ult1 ind_nom_pens_ult1 ind_recibo_ult1
##  Min.   :0.000   Min.   :0.000     Min.   :0.0000
##  1st Qu.:0.000   1st Qu.:0.000     1st Qu.:0.0000
##  Median :0.000   Median :0.000     Median :0.0000
##  Mean   :0.055   Mean   :0.059     Mean   :0.1279
##  3rd Qu.:0.000   3rd Qu.:0.000     3rd Qu.:0.0000
##  Max.   :1.000   Max.   :1.000     Max.   :1.0000
##  NA's   :16063   NA's   :16063
```

Note that indresi and indext may need special attention here, which is not taken care of right now.

```
testSet <- fread("../data/test.csv", header = TRUE, verbose = FALSE, na.strings = naStrings)
nTest <- nrow(testSet)
commonFeatureNames <- colnames(trainSet)[1:24]
productFeatureNames <- colnames(trainSet)[25:48]
testSet[,productFeatureNames] <- NA
wholeSet <- rbindlist(list(trainSet, testSet))
```

Let us remove the non-product features with too many NA values. The features to remove are

```
naRatios <- colMeans(is.na(wholeSet[, 1:24, with = FALSE]))
withTooManyNas <- naRatios > 0.9
for (feature in names(which(withTooManyNas))){
  cat(sprintf("feature %s is removed, with NA ratio being %.3f.\n", feature, naRatios[feature]))
}
```

```
## feature ult_fec_cli_1t is removed, with NA ratio being 0.998.
## feature conyuemp is removed, with NA ratio being 1.000.
```

```
wholeSet <- wholeSet[, which(withTooManyNas):=NULL]
commonFeatureNames <- commonFeatureNames[!withTooManyNas]
```

Then we remove the numeric features with very little variance:

```
withLittleVariance <- sapply(commonFeatureNames, function(featureName) is.numeric(wholeSet[[featureName]

for (feature in commonFeatureNames[which(withLittleVariance)]){
  cat(sprintf("feature %s is removed.\n", feature))
}
```

```
## feature tipodom is removed.
```

```
wholeSet <- wholeSet[, which(withLittleVariance):=NULL]
commonFeatureNames <- commonFeatureNames[which(!withLittleVariance[1:length(commonFeatureNames)])]
```

Now let us look at the features with NAs

```
nCommonFeatures <- length(commonFeatureNames)
nProductFeatures <- 24
CheckFeaturesWithNas <- function(nCommonFeatures){
  nTotal <- wholeSet %>% nrow
  naRatioCommonFeatures <- sapply(wholeSet[, 1:nCommonFeatures, with = FALSE], function(feature) mean(is
  naRatioProductFeatures <- sapply(wholeSet[1:(nTotal-nTest), 1:nProductFeatures + nCommonFeatures, with
  naRatio <- c(naRatioCommonFeatures, naRatioProductFeatures)

  hasNa <- naRatio != 0
  colnamesWithNa <- colnames(wholeSet)[hasNa]
  naRatio <- naRatio[hasNa]
  if (length(naRatio) == 0){
```

```
    cat("No NAs!")
    return()
  }

  for(i in 1:length(naRatio)){
    colname <- colnamesWithNa[i]
    cat(sprintf("Col %s has %d NAs with ratio being %.3f %%: %s\n", colname, ceiling(naRatio[i] * nrow(
  }
}
CheckFeaturesWithNas(nCommonFeatures)
```

```
## Col ind_empleado has 27734 NAs with ratio being 0.190 %: Employee index: A active, B ex employed, F
## Col pais_residencia has 27734 NAs with ratio being 0.190 %: Customer's Country residence
## Col sexo has 27809 NAs with ratio being 0.191 %: Customer's sex
## Col age has 27734 NAs with ratio being 0.190 %: Age
## Col fecha_alta has 27734 NAs with ratio being 0.190 %: The date in which the customer became as the
## Col ind_nuevo has 27734 NAs with ratio being 0.190 %: New customer Index. 1 if the customer registere
## Col antiguedad has 27734 NAs with ratio being 0.190 %: Customer seniority (in months)
## Col indrel has 27734 NAs with ratio being 0.190 %: 1 (First/Primary), 99 (Primary customer during the
## Col indrel_1mes has 153933 NAs with ratio being 1.056 %: Customer type at the beginning of the month
## Col tiprel_1mes has 153933 NAs with ratio being 1.056 %: Customer relation type at the beginning of
## Col indresi has 27734 NAs with ratio being 0.190 %: Residence index (S (Yes) or N (No) if the reside
## Col indext has 27734 NAs with ratio being 0.190 %: Foreigner index (S (Yes) or N (No) if the custome
## Col canal_entrada has 192901 NAs with ratio being 1.323 %: channel used by the customer to join
## Col indfall has 27734 NAs with ratio being 0.190 %: Deceased index. N/S
## Col cod_prov has 97588 NAs with ratio being 0.669 %: Province code (customer's address)
## Col nomprov has 97588 NAs with ratio being 0.669 %: Province name
## Col ind_actividad_cliente has 27734 NAs with ratio being 0.190 %: Activity index (1, active customer
## Col renta has 3027129 NAs with ratio being 20.760 %: Gross income of the household
## Col segmento has 196413 NAs with ratio being 1.347 %: segmentation: 01 - VIP, 02 - Individuals 03 - 
## Col ind_nomina_ult1 has 17163 NAs with ratio being 0.118 %: Payroll
## Col ind_nom_pens_ult1 has 17163 NAs with ratio being 0.118 %: Pensions
```

It looks very suspiciout that some features have the same number of NAs. It turns out that it is the same set of samples that have those missing features: `ind_empleado`, `pais_residencia`, `age`, `fecha_alta`, `ind_nuevo`, `antiguedad`, `indrel`, `indresi`, `indext`, `indfall`.

```
areNas <- is.na(wholeSet$ind_empleado)
all(areNas == is.na(wholeSet$pais_residencia))
```

```
## [1] TRUE
```

```
all(areNas == is.na(wholeSet$age))
```

```
## [1] TRUE
```

```
all(areNas == is.na(wholeSet$fecha_alta))
```

```
## [1] TRUE
```

```
all(areNas == is.na(wholeSet$ind_nuevo))
```

```
## [1] TRUE
```

```
all(areNas == is.na(wholeSet$antiguedad))
```

```
## [1] TRUE
```

```
all(areNas == is.na(wholeSet$indrel))
```

```
## [1] TRUE
```

```
all(areNas == is.na(wholeSet$indresi))
```

```
## [1] TRUE
```

```
all(areNas == is.na(wholeSet$indext))
```

```
## [1] TRUE
```

```
all(areNas == is.na(wholeSet$indfall))
```

```
## [1] TRUE
```

Now let us check these samples:

```
naDataTable <- wholeSet[areNas, ]
tail(naDataTable[,1:24, with = FALSE], 5)
```

```
##    fecha_dato ncodpers ind_empleado pais_residencia sexo age fecha_alta
## 1: 2015-06-28   550081           NA              NA   NA  NA         NA
## 2: 2015-06-28   550693           NA              NA   NA  NA         NA
## 3: 2015-06-28   549518           NA              NA   NA  NA         NA
## 4: 2015-06-28   549207           NA              NA   NA  NA         NA
## 5: 2015-06-28   549255           NA              NA   NA  NA         NA
##    ind_nuevo antiguedad indrel indrel_1mes tiprel_1mes indresi indext
## 1:        NA         NA     NA          NA          NA      NA     NA
## 2:        NA         NA     NA          NA          NA      NA     NA
## 3:        NA         NA     NA          NA          NA      NA     NA
## 4:        NA         NA     NA          NA          NA      NA     NA
## 5:        NA         NA     NA          NA          NA      NA     NA
##    canal_entrada indfall cod_prov nomprov ind_actividad_cliente renta
## 1:            NA      NA       NA      NA                    NA    NA
## 2:            NA      NA       NA      NA                    NA    NA
## 3:            NA      NA       NA      NA                    NA    NA
## 4:            NA      NA       NA      NA                    NA    NA
## 5:            NA      NA       NA      NA                    NA    NA
##    segmento ind_ahor_fin_ult1 ind_aval_fin_ult1 ind_cco_fin_ult1
## 1:       NA                 0                0               0
## 2:       NA                 0                0               0
## 3:       NA                 0                0               0
## 4:       NA                 0                0               0
## 5:       NA                 0                0               0
```

It looks like these samples misses all common features. Let us verify the hypo:

```
allValuesMissing <- sapply(commonFeatureNames[3:nCommonFeatures], function(featureName){
  all(is.na(naDataTable[[featureName]]))
})

all(allValuesMissing)
```

```
## [1] TRUE
```

It seems that maybe all data associated with these ncodpers in this set are corrupted. Let us verify:

```
ncodpers <- naDataTable$ncodpers %>% unique
cat(sprintf("We have %d unique customers", length(ncodpers)))
```

```
## We have 7340 unique customers
```

```
withIds <- wholeSet$ncodpers %in% ncodpers
ntotalTransactions <- withIds %>% sum

cat(sprintf("We have %d instances of this customers in the whole dataset, while %d of those instances mi
```

```
## We have 29719 instances of this customers in the whole dataset, while 27734 of those instances miss a
```

So this hypo is not entirely true, but almost true. To see if it is safe to remove all those instances, we have to make sure there are no instances in test set that can benefit from those instances. The instances to predict that can benefit from those bad instances are also the ones with common features missing.

Then let us check if there are samples in test set like this:

```
is.na(testSet[,commonFeatureNames, with=FALSE]) %>% colSums
```

```
##            fecha_dato              ncodpers          ind_empleado
##                     0                     0                     0
##       pais_residencia                  sexo                   age
##                     0                     5                     0
##            fecha_alta             ind_nuevo            antiguedad
##                     0                     0                     0
##                indrel           indrel_1mes           tiprel_1mes
##                     0                  4152                  4152
##               indresi                indext         canal_entrada
##                     0                     0                  6775
##               indfall              cod_prov               nomprov
##                     0                  3997                  3997
## ind_actividad_cliente                 renta              segmento
##                     0                232754                  7045
```

As the results shows, we do not have instances with all common features missing to predict. So we can safely remove those broken data from whole/train dataset.

```
wholeSet <- wholeSet[!areNas, ]
dim(wholeSet)
```

```
## [1] 14554020       45
```

Now we check again which features still have missing values

```
CheckFeaturesWithNas(length(commonFeatureNames))
```

```
## Col sexo has 75 NAs with ratio being 0.001 %: Customer's sex
## Col indrel_1mes has 126199 NAs with ratio being 0.867 %: Customer type at the beginning of the month
## Col tiprel_1mes has 126199 NAs with ratio being 0.867 %: Customer relation type at the beginning of t
## Col canal_entrada has 165167 NAs with ratio being 1.135 %: channel used by the customer to join
## Col cod_prov has 69854 NAs with ratio being 0.480 %: Province code (customer's address)
## Col nomprov has 69854 NAs with ratio being 0.480 %: Province name
## Col renta has 2999395 NAs with ratio being 20.609 %: Gross income of the household
## Col segmento has 168679 NAs with ratio being 1.159 %: segmentation: 01 - VIP, 02 - Individuals 03 - G
## Col ind_nomina_ult1 has 232 NAs with ratio being 0.002 %: Payroll
## Col ind_nom_pens_ult1 has 232 NAs with ratio being 0.002 %: Pensions
```

Now we preprocess the dataset, together with the test set.

As we can see, we have 5 customers with gender info REALLY missing, we impute them with the mode number.

```r
# sexo
uids <- wholeSet$ncodpers[is.na(wholeSet$sexo)] %>% unique()
print(uids)
```

```
## [1] 394860 415005 476023 278257 216507
```

```r
wholeSet$sexo[is.na(wholeSet$sexo)] <- "V"
table(wholeSet$sexo)
```

```
##
##       H       V
## 6622378 7931642
```

```r
# indrel_1mes: Customer type at the beginning of the month ,1 (First/Primary customer), 2 (co-owner ),P
tb <- table(wholeSet$indrel_1mes)
wholeSet$indrel_1mes[wholeSet$indrel_1mes == "1.0"] <- "1"
wholeSet$indrel_1mes[wholeSet$indrel_1mes == "2.0"] <- "2"
wholeSet$indrel_1mes[wholeSet$indrel_1mes == "3.0"] <- "3"
wholeSet$indrel_1mes[wholeSet$indrel_1mes == "4.0"] <- "4"
tb <- table(wholeSet$indrel_1mes)
wholeSet$indrel_1mes[is.na(wholeSet$indrel_1mes)] <- names(which.max(tb))[1]
table(wholeSet$indrel_1mes)
```

```
##
##        1        2        3        4        P
## 14546630     1421     4707      326      936
```

```r
# tiprel_1mes: Customer relation type at the beginning of the month, A (active), I (inactive), P (forme
tb <- table(wholeSet$tiprel_1mes)
wholeSet$tiprel_1mes[is.na(wholeSet$tiprel_1mes)] <- names(which.max(tb))[1]
table(wholeSet$tiprel_1mes)
```

```
##
##       A       I       N       P       R
## 6581012 7967039       6    5033     930
```

```r
# canal_entrada: channel used by the customer to join
tb <- table(wholeSet$canal_entrada)
wholeSet$canal_entrada[is.na(wholeSet$canal_entrada)] <- names(which.max(tb))[1]
table(wholeSet$canal_entrada)
```

```
##
##     004     007     013     025     K00     KAA     KAB     KAC     KAD
##     226   31262   28798      12     162   70952   66395    8181   11342
##     KAE     KAF     KAG     KAH     KAI     KAJ     KAK     KAL     KAM
##   54057   32472   78960   26463   40127   25950     888    8089   12027
##     KAN     KAO     KAP     KAQ     KAR     KAS     KAT     KAU     KAV
##    1456    7127   15947   19218   34823   91807 3474073     264     113
##     KAW     KAY     KAZ     KBB     KBD     KBE     KBF     KBG     KBH
##   36481   71791   34325    1386     258     156    4022    1828    7679
##     KBJ     KBL     KBM     KBN     KBO     KBP     KBQ     KBR     KBS
##     552     613     590      66    7878      90    4446    1993     763
##     KBU     KBV     KBW     KBX     KBY     KBZ     KCA     KCB     KCC
##    3340     957    1213     108     688   49455    1632    5523   52428
##     KCD     KCE     KCF     KCG     KCH     KCI     KCJ     KCK     KCL
##    3499     330     450    5736   25741   28323     228    1022    4466
##     KCM     KCN     KCO     KCP     KCQ     KCR     KCS     KCT     KCU
```

```
##    3468    1194     192     168     210     206     252     114    1158
##     KCV     KCX     KDA     KDB     KDC     KDD     KDE     KDF     KDG
##     246      72     406      18    1636     576     889     776     504
##     KDH     KDI     KDL     KDM     KDN     KDO     KDP     KDQ     KDR
##     204      18      12    2641     210    1890    1073    1183    8587
##     KDS     KDT     KDU     KDV     KDW     KDX     KDY     KDZ     KEA
##    2067    1382    2771     456     822    1856    2103     570     912
##     KEB     KEC     KED     KEE     KEF     KEG     KEH     KEI     KEJ
##     396     246    3205     186     354    2015    1643    1002    9885
##     KEK     KEL     KEM     KEN     KEO     KEQ     KES     KEU     KEV
##     576    2639      72    5245     906     156    6275     282     896
##     KEW     KEY     KEZ     KFA     KFB     KFC     KFD     KFE     KFF
##    6046   37419    2596  435323     114 3299082   47412     267    5903
##     KFG     KFH     KFI     KFJ     KFK     KFL     KFM     KFN     KFP
##    7247    2766     942    7080    4184    4060     396    4827   10114
##     KFR     KFS     KFT     KFU     KFV     KGC     KGN     KGU     KGV
##     396    7170    8592    5256      72      30      18      30    9534
##     KGW     KGX     KGY     KHA     KHC     KHD     KHE     KHF     KHK
##    1062   10106    4413      54   17548  124281 4472102   22051  258003
##     KHL     KHM     KHN     KHO     KHP     KHQ     KHR     KHS     RED
##   48076  217356  134756   10520     776  666008       2      27   81935
```

```
# nomprov
wholeSet$nomprov[wholeSet$nomprov %>% is.na] <- "Unknown"
wholeSet <- wholeSet[, nomprov:= as.factor(nomprov)]

# cod_prov is redundant with nomprov
wholeSet <- wholeSet[,cod_prov:=NULL]
table(wholeSet$nomprov)
```

```
##
##                 ALAVA              ALBACETE              ALICANTE
##                 40317                121933                335729
##               ALMERIA               ASTURIAS                 AVILA
##                 64648                284157                 41421
##               BADAJOZ        BALEARS, ILLES             BARCELONA
##                205967                134118               1364230
##               BIZKAIA                BURGOS               CACERES
##                198454                103852                138540
##                 CADIZ             CANTABRIA             CASTELLON
##                314588                166588                111488
##                 CEUTA           CIUDAD REAL               CORDOBA
##                  7733                127507                154585
##            CORUÑA, A                CUENCA              GIPUZKOA
##                458134                 61108                 76460
##                GIRONA               GRANADA            GUADALAJARA
##                 96865                191057                 69329
##               HUELVA                HUESCA                  JAEN
##                130343                 42976                 68908
##                  LEON                LERIDA                  LUGO
##                 88760                 84505                 90630
##                MADRID               MALAGA               MELILLA
##               4709361                391767                 10108
##                MURCIA               NAVARRA               OURENSE
##                424652                 94653                 89632
```

```
##               PALENCIA            PALMAS, LAS              PONTEVEDRA
##                  52608                 252183                  299048
##             RIOJA, LA            SALAMANCA SANTA CRUZ DE TENERIFE
##                  91027                 175346                   76131
##               SEGOVIA                SEVILLA                   SORIA
##                  45232                 645875                   18927
##             TARRAGONA                 TERUEL                  TOLEDO
##                 107789                  24067                  195792
##               Unknown               VALENCIA              VALLADOLID
##                  69854                 730543                  254342
##                ZAMORA               ZARAGOZA
##                  54371                 365782
```

```
commonFeatureNames <- commonFeatureNames[commonFeatureNames != "cod_prov"]
```

```
summary(wholeSet$renta)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's
##     1203    68700   101800   134200   155900 28890000  2999395
```

```
# renta 20.476 % missing is a very high number
wholeSet$renta[wholeSet$renta %>% is.na] <- median(wholeSet$renta, na.rm = TRUE)
```

```
tb <- table(wholeSet$segmento)
wholeSet$segmento[is.na(wholeSet$segmento)] <- names(which.max(tb))[1]
table(wholeSet$segmento)
```

```
##
##           01 - TOP  02 - PARTICULARES 03 - UNIVERSITARIO
##             598103            8674310            5281607
```

```
# ind_nomina_ult1: product feature
testSetIndexes <- tail(1:nrow(wholeSet), nTest)
wholeSet$ind_nomina_ult1[wholeSet$ind_nomina_ult1 %>% is.na] <- median(wholeSet$ind_nomina_ult1, na.rm =
wholeSet$ind_nomina_ult1[testSetIndexes] <- NA
# ind_nom_pens_ult1: product feature
wholeSet$ind_nom_pens_ult1[wholeSet$ind_nom_pens_ult1 %>% is.na] <- median(wholeSet$ind_nom_pens_ult1, n
wholeSet$ind_nom_pens_ult1[testSetIndexes] <- NA
```

Now we check if there still any missing values existing:

```
CheckFeaturesWithNas(length(commonFeatureNames))
```

```
## No NAs!
```

```
## NULL
```

Now let us write the preprocessed datasets in files

```
#write.csv(wholeSet[1:(nrow(wholeSet) - nrow(testSet)), ], "../data/preprocessed_train.csv", row.names =
#write.csv(wholeSet[testSetIndexes,], "../data/preprocessed_test.csv", row.names = FALSE)
```