Data Process

Loading

```
df = read.csv("train_x.csv")
df2 = read.csv("test_x.csv")
tar = read.csv("train_y.csv")
record = nrow(df)
data = rbind(df,df2)
```

Cleaning

```
# Cleaning the data
data$MIL[data$MIL < 0] = 3 # treat missing mil as never in the mil
data$CITSHP[data$CITSHP < 0] = 1 # treat missing citship as us citship

# This variable isn't important
# just make missing type as a different type
data$RACEAS[data$RACEAS < 0] = 9

# This variable isn't important
# just make missing type as a different type
data$RACEPI[data$RACEPI < 0] = 6
# INUSY is cleaned later
```

Processing

(1) Filling in NA

```
# These are important, needs to be filled in
data$WAGP[data$WAGP == -6] = NA
data$SEMP[data$SEMP == -6] = NA
data$SSP[data$SSP == -6] = NA
data$PAP[data$PAP == -6] = NA
data$RETP[data$RETP == -6] = NA
data$INTP[data$INTP == -6] = NA
data$OIP[data$OIP == -6] = NA
data$SSIP[data$SSIP == -6] = NA
```

(2) remove NA by data imputation

The general idea is that for each of the missing money,

if that person's `age` is less than 23, then he/she is likely to have the missing money with 0 with a high probability.

if that person's age is more than 23, then he/she is likely to have an income, then we find a group of people, whose age is within [age-4, age + 4], but at least 23, and at most 86; and those people must have the same birth nation. pick about 4/5 of them to increase the randomness, and take the median of them, to make the data imputation more reasonable.

(Age, and nationality is important based on data visualization at a first place)

```
counter = 0
for (i in 1:nrow(data)) {
  if (any(is.na(data[i,]))) {

    counter = counter + 1
    if (counter %% 100 == 0) {
      print(counter)
    }
    newdata <- na.omit(data)

    age = data$AGE[i]
    # Age range
    lower = max(age - 4, 23)
    upper = min(age + 4, 86)

    if (age >= 22) {

    # add conditions on nationality and Age range
    newdata = newdata[
          (newdata$NATVTY == data$NATVTY[i] &
           newdata$AGE >= lower &
           newdata$AGE <= upper),]
    # Randomly pick sample about 4/5 and then take the median
    newdata = newdata[sample(nrow(newdata), round(4*nrow(newdata)/5)), ]
    if (nrow(newdata)<5) { # If the size is too small, we rather like it to make a general filling
      lower = max(age-12, 23)
      upper = min(age+12, 86)
      newdata <- na.omit(data)
      newdata = newdata[
           (newdata$AGE >= lower &
           newdata$AGE <= upper),]
    }

    if (is.na(data$WAGP[i])) {
    data$WAGP[i] = median(newdata$WAGP)
    }
    if (is.na(data$SEMP[i])) {
    data$SEMP[i] = median(newdata$SEMP)
    }
    if (is.na(data$SSP[i])) {
    data$SSP[i] = median(newdata$SSP)
    }
    if (is.na(data$SSIP[i])) {
    data$SSIP[i] = median(newdata$SSIP)
    }
    if (is.na(data$PAP[i])) {
    data$PAP[i] = median(newdata$PAP)
    }
```

```
    if (is.na(data$RETP[i])) {
    data$RETP[i] = median(newdata$RETP)
    }
    if (is.na(data$INTP[i])) {
    data$INTP[i] = median(newdata$INTP)
    }
    if (is.na(data$OIP[i])) {
    data$OIP[i] = median(newdata$OIP)
    }
    } else { # Teenage have them to be 0 with high probability
      data$WAGP[i] = 0
      data$SEMP[i] = 0
      data$SSP[i] = 0
      data$SSIP[i] = 0
      data$PAP[i] = 0
      data$RETP[i] = 0
      data$INTP[i] = 0
      data$OIP[i] = 0
    }
  }
}
```

(3) Feature Engineering

(a) Introducing features

```
# Net Income
data$NET = data$WAGP + data$SEMP + data$SSP + data$PAP + data$RETP + data$INTP + data$OIP + data$SSIP

# Only Income earned by jobs
data$INCOME = data$WAGP + data$SEMP + data$OIP + data$RETP

# Only what people give them for helping out
data$ASSIST = data$SSP + data$PAP + data$SSIP

# Birthday
# the data is likely to be collected in about 2019
data$BIRTH = 2019 - data$AGE

# At what age they move to the cur address
data$AGE_MOVE = data$MOVE - data$BIRTH + 1

# At what age they come to US, -6 cleaned as birth day
data$INUSYR[data$INUSYR < 0] = data$BIRTH[data$INUSYR < 0]
data$AGE_INUSYR = data$INUSYR - data$BIRTH + 1


# Calculate the median of group of people from many perspective
# like if you have a degree, then probably the amount of money you earn
# is higher than the median of those with age in [age-4, age+4].

data$N_I = 0 # income - same nationality median income
data$S_I = 0 # income - same sex median income
```

```
data$AGE_I = 0 # income - similar age median income
data$N_N = 0 #  net income - same nationality median net income
data$S_N = 0 # net income - same sex median net income
data$AGE_N = 0 # net income - similar Age median net income

data$NS_I = 0 # income - same nationality and sex median income
data$NS_N = 0 # net income - same nationality and sex median net income
```

(b) Process procedure

```
# Make KV pair to save some time in life
# It's for grouping

N_I <- list()
S_I <- list()
AGE_I <- list()
NS_I <- list()
N_N <- list()
S_N <- list()
NS_N <- list()
AGE_N <- list()

for (n in unique(data$NATVTY)) {
  newdata = data[(data$NATVTY == n & data$AGE >= 23),]
  N_I[[ n ]] <- median(newdata$INCOME)
  N_N[[ n ]] <- median(newdata$NET)
}

for (n in unique(data$SEX)) {
  newdata = data[(data$SEX == n & data$AGE >= 23),]
  S_I[[ n ]] <- median(newdata$INCOME)
  S_N[[ n ]] <- median(newdata$NET)
}

for (n in unique(data$NATVTY)) {
  newdata = data[(data$SEX == 1 & data$NATVTY==n & data$AGE >= 23),]
  NS_I[[ n ]] <- median(newdata$INCOME)
  NS_N[[ n ]] <- median(newdata$NET)

  newdata = data[(data$SEX == 2 & data$NATVTY==n & data$AGE >= 23),]
  NS_I[[ n+1000 ]] <- median(newdata$INCOME)
  NS_N[[ n+1000 ]] <- median(newdata$NET)
}

for (n in unique(data$AGE)) {
  if (n >= 23) {
  lower = max(n - 4, 23)
  upper = min(n + 4, 86)
  newdata = data[(data$AGE >= lower & data$AGE <= upper & data$AGE >= 23),]
  AGE_I[[ n ]] <- median(newdata$INCOME)
  AGE_N[[ n ]] <- median(newdata$NET)
  } else {
  AGE_I[[ n ]] <- 0
```

```
  AGE_N[[ n ]] <- 0
  }
}
```

(c) Filling in the group comparison

```
counter = 0
for (i in 1:nrow(data)) {
  if (data$AGE[i] >= 23) {
  counter = counter + 1
  if (counter %% 1000 == 0) {
      print(counter)
  }


  data$N_I[i] = N_I[[data$NATVTY[i]]] # nationality median income
  data$S_I[i] = S_I[[data$SEX[i]]] # sex median income
  data$AGE_I[i] = AGE_I[[data$AGE[i]]] # Age median income
  data$N_N[i] = N_N[[data$NATVTY[i]]] # nationality median net income
  data$S_N[i] = S_N[[data$SEX[i]]] # sex median net income
  data$AGE_N[i] = AGE_N[[data$AGE[i]]] # Age median net income

  if (data$SEX[i] == 1) {
  data$NS_I[i] = NS_I[[data$NATVTY[i]]] # combine nationality and sex
  data$NS_N[i] = NS_N[[data$NATVTY[i]]] #
  } else {
  data$NS_I[i] = NS_I[[data$NATVTY[i] + 1000]] # combine nationality and sex
  data$NS_N[i] = NS_N[[data$NATVTY[i] + 1000]] #
  }
  }
}

data$N_I = data$INCOME - data$N_I # nationality median income
data$S_I = data$INCOME - data$S_I# sex median income
data$AGE_I = data$INCOME - data$AGE_I# Age median income
data$N_N = data$NET - data$N_N # nationality median net income
data$S_N = data$NET - data$S_N# sex median net income
data$AGE_N = data$NET - data$AGE_N# Age median net income

data$NS_I = data$INCOME -data$NS_I # combine nationality and sex
data$NS_N = data$NET -data$NS_N #
```

Pass to the machine learning part

```
data = data[-c(14,27)] # drop birthday, the same as age, Also RACEPI

# Separate data set
x = split(data,cumsum(1:nrow(data)%in%(record+1)))
train = x$`0`
test = x$`1`
```

```r
write.csv(train, './Tree/train_x.csv', row.names = FALSE)
write.csv(test, './Tree/test_x.csv', row.names = FALSE)
```