Metrics and ML - Original Imputation Method

Sydney Gu and Isabella Lin

2025-02-02

```
##
## 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

## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

## Loaded glmnet 4.1-8

##
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
purl("cleaning.Rmd", output = "cleaning2.R")
```

```
##
##
## processing file: cleaning.Rmd

##   |                                                                      |

## output file: cleaning2.R

## [1] "cleaning2.R"
```

```r
source("cleaning2.R")
```

```
##
## Attaching package: 'kableExtra'

## The following object is masked from 'package:dplyr':
##
##     group_rows

## Warning: package 'caret' was built under R version 4.3.3

## Loading required package: ggplot2

## Loading required package: lattice

## Thank you for using fastDummies!

## To acknowledge our work, please cite the package:

## Kaplan, J. & Schlegel, B. (2023). fastDummies: Fast Creation of Dummy (Binary) Columns and Rows from
```

Remove observations with NAs:

```r
cex <- na.omit(cex)
```

```r
ndur_model <- lm(ndur ~ income + fsize + sex + age + factor(region) +
                   factor(race), data = cex)
summary(ndur_model)
```

```
##
## Call:
## lm(formula = ndur ~ income + fsize + sex + age + factor(region) +
##     factor(race), data = cex)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -35495  -3787  -1109   2321 123559
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.929e+03  4.425e+02   8.879  < 2e-16 ***
## income            1.676e-01  2.288e-03  73.256  < 2e-16 ***
## fsize             7.261e+02  4.137e+01  17.552  < 2e-16 ***
## sex               1.003e+03  1.875e+02   5.347 9.09e-08 ***
## age               6.293e+01  6.326e+00   9.948  < 2e-16 ***
## factor(region)2 -1.385e+03  1.626e+02  -8.518  < 2e-16 ***
## factor(region)3 -9.162e+02  1.640e+02  -5.586 2.37e-08 ***
## factor(region)4 -2.986e+02  1.720e+02  -1.736   0.0825 .
## factor(race)2   -1.811e+03  2.266e+02  -7.993 1.42e-15 ***
## factor(race)3   -2.093e+03  4.405e+02  -4.750 2.05e-06 ***
```

```
## factor(race)4   -9.168e+02  3.856e+02  -2.377   0.0175 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6723 on 13895 degrees of freedom
## Multiple R-squared:  0.3067, Adjusted R-squared:  0.3062
## F-statistic: 614.7 on 10 and 13895 DF,  p-value: < 2.2e-16
```

```
psid <- psid %>%
  mutate(
    region = as.factor(region),
    race = as.factor(race)
  )


psid$ndur_pred <- predict(ndur_model, newdata = psid)

head(psid$ndur_pred)
```

```
## [1]        NA  8146.593        NA 18414.600 22183.036 65241.918
```

```
#features <- cex %>% select(-c(ndur, ndurplus))
#pca <- prcomp(features, center = TRUE, scale. = TRUE)
#loadings <- abs(pca$rotation)
#top <- 5
#num_pcs <- 5

#top_vars <- apply(loadings[, 1:num_pcs], 2, function(x) {
 # names(sort(x, decreasing = TRUE)[1:top])
#})

#print(top_vars)
```

```
#test_pca_data <- predict(pca, newdata = psid)
```