Simple Logistic

Andres Potapczynski (ap3635)

```
## Rescaling
inputs <- data_sub %>%
  mutate(
    income_st = (log(client_income) - mean(log(client_income))) / sd(log(client_income)),
    appraisal_st = (log(appraisal_value) - mean(log(appraisal_value))) / sd(log(appraisal_value)),
    market_st = (log(asset_market_value) - mean(log(asset_market_value))) / sd(log(asset_market_value))
    mar_2_inc_st = (mar_2_inc - mean(mar_2_inc)) / sd(mar_2_inc),
    app_2_inc_st = (app_2_inc - mean(app_2_inc)) / sd(app_2_inc),
    mar_2_app_st = (mar_2_app - mean(mar_2_app)) / sd(mar_2_app),
    age_st = (age - mean(age)) / sd(age)
         ) %>%
  dplyr::select(
    # income_st,
    mar_2_inc_st,
    appraisal_st,
    # app_2_inc_st,
    # mar_2_app_st,
    market_st,
    # age_st,
    y,
    y2
  )

## Train / Test split
set.seed(seed = 81989843)
pct_train = 0.9
sample_size = round(pct_train * nrow(inputs))
sample <- sample(x = nrow(inputs), size = sample_size, replace = F)

## Allocate train
y = inputs$y

inputs_train = inputs[sample, ]
y_train = y[sample]

## Allocate test
inputs_test = inputs[-sample, ]
y_test = y[-sample]

## Create inputs for STAN
data_stan_train <- list(N=nrow(inputs_train),
                        D=ncol(inputs_train),
                        X=inputs_train,
                        y=y_train)

## Inputs for STAN
X_train = inputs_train %>% dplyr::select(-y, -y2)
X_test = inputs_test %>% dplyr::select(-y, -y2)
N = nrow(X_train)
```

```r
D = ncol(X_train)

data_stan_train = list(N=N, D=D, X=X_train, y=y_train)
```

```
## Inference for Stan model: logistic_v02.
## 4 chains, each with iter=2000; warmup=1000; thin=1;
## post-warmup draws per chain=1000, total post-warmup draws=4000.
##
##          mean se_mean   sd  2.5%   50% 97.5% n_eff Rhat
## alpha   -2.71       0 0.03 -2.76 -2.70 -2.66  4237    1
## beta[1] -0.11       0 0.04 -0.18 -0.11 -0.04  5021    1
## beta[2]  0.13       0 0.06  0.02  0.13  0.24  4257    1
## beta[3] -0.25       0 0.06 -0.37 -0.25 -0.13  4116    1
##
## Samples were drawn using NUTS(diag_e) at Sat Dec  1 09:41:46 2018.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

```r
sims <- rstan::extract(sm.logistic_v01)
beta_median <- apply(X = sims$beta, MARGIN = 2, FUN = median)
alpha_median <- median(sims$alpha)
```

```r
threshold = 0.10

y_hat_baseline <- rep(0, times = length(y_test))

accuracy_base = sum(y_hat_baseline == y_test) / length(y_test)

cat('\nBaseline accuracy: ', accuracy_base * 100)
```

```
##
## Baseline accuracy:  93.54098
```

```r
proba_hat <- invlogit(as.matrix(X_test) %*% beta_median + alpha_median)
proba_hat <- as.numeric(proba_hat)

y_hat = rep(0, times = length(y_test))
y_hat[proba_hat > threshold] = 1

accuracy = sum(y_hat == y_test) / length(y_test)

cat('\nLogistic accuracy: ', accuracy * 100)
```

```
##
## Logistic accuracy:  93.5082
```

```r
cat('\nConfusion table\n')
```

```
##
## Confusion table
```

```r
print(table(y_test, y_hat))
```

```
##       y_hat
## y_test    0    1
##      0 2852    1
```

```
##      1  197    0
```

```r
test_df = data.frame(ID_state = data_sub$ID_state[-sample],
                     state = data_sub$state[-sample],
                     y_test = y_test,
                     y_hat = y_hat)
test_df <- test_df %>%
  group_by(state) %>%
  summarize(y_sum_test = sum(y_test),
            y_sum_hat = sum(y_hat)) %>%
  arrange(desc(y_sum_test)) %>%
  ungroup()

accuracy_baseline = mean(abs(test_df$y_sum_test) ** 2)
accuracy = mean(abs(test_df$y_sum_hat - test_df$y_sum_test) ** 2)

cat('\nBaseline MSE: ', accuracy_baseline * 100)
```

```
##
## Baseline MSE:  8021.875
```

```r
cat('\nLogistic MSE: ', accuracy * 100)
```

```
##
## Logistic MSE:  7956.25
```

```r
test_df = data.frame(ID_city = data_sub$ID_city[-sample],
                     city = data_sub$city[-sample],
                     y_test = y_test,
                     y_hat = y_hat)
test_df <- test_df %>%
  group_by(city) %>%
  summarize(y_sum_test = sum(y_test),
            y_sum_hat = sum(y_hat)) %>%
  arrange(desc(y_sum_test)) %>%
  ungroup()

accuracy_baseline = mean(abs(test_df$y_sum_test) ** 2)
accuracy = mean(abs(test_df$y_sum_hat - test_df$y_sum_test) ** 2)

cat('\nBaseline MSE: ', accuracy_baseline * 100)
```

```
##
## Baseline MSE:  154.7425
```

```r
cat('\nLogistic MSE: ', accuracy * 100)
```

```
##
## Logistic MSE:  154.4715
```

```r
y_sum_train = sum(y_train)
y_sum_rep = apply(X = sims$y_rep, MARGIN = 1, FUN = sum)
cat('\nTotal training defaults: ', y_sum_train)
```

```
##
## Total training defaults:  1756
```

```r
cat('\nTotal replicated defaults: ', mean(y_sum_rep))
```

```
##
## Total replicated defaults:  1756.321
```