Predicting beneficieries of health assistance program in Indonesia villages

Alfian Bin Aman & Hamidah Alatas

December 23, 2021

1 Introduction

This project aims to predict the share of households in a village with national health insurance. This health insurance program's provides health coverage to low-income families in Indonesia. This analysis is done in village-level using village census provided by Statistics Indonesia. We use and compare several supervised learning techniques that will predict better our target variable. For some techniques, we use tuning to estimate parameters through bootstrapping. The goal is to find a model that minimizes root mean-squared error (rmse) in the testing data.

2 Data and Variables

The data that we use is Indonesia's Village Potential Statistics (PODES) from 2006. This dataset is a census of all villages (*desa*) in Indonesia, with number of observations of 69,957. The dataset contains useful information on village characteristics, such as main sources of income, number of households in the village, land characteristics, crime that happened, pandemic, etc. We obtained this dataset from Columbia's Research Data Services. In total, podes data has 488 variables and all are clean data with no major missing data issues. However, in this analysis we will only use in total 39 variables.

Before using the data we do some data cleaning process that includes renaming and dropping variables that will not be used for the analysis. We also create some new variables that were derivations of existing variables.

```
selected_var = c("r605", "r401c", "r401a","r401b", "r510ak2","r510bk2",
                 "r510ck2", "r510dk2", "r601ak2", "r601ak3", "r601bk2",
                 "r601bk3", "r601ck2", "r601ck3", "r601dk2", "r601dk3",
                 "r601ek2", "r601ek3", "r601fk2", "r601fk3", "r601gk2",
                 "r601gk3", "r603ak2", "r603bk2", "r603ck2", "r603dk2",
                 "r603ek2", "r603fk2", "r603gk2", "r603hk2", "r603ik2",
                 "r603jk2", "r604a1", "r604a2",  "r606", "r607ak2", "r607bk2",
                 "r607ck2", "r607dk2",  "r607ek2", "r704a1k2", "r704a2k2",
                 "r704a3k2", "r704a4k2",  "r704a5k2", "r704a6k2", "r901b2",
                 "r1107", "r1106", "r1204a1k3", "r1204a2k3", "r1204a3k3",
                 "r1204a4k3", "r1204a5k3", "r1204a6k3", "r1204a7k3",
                 "r1204a8k3", "r1204a9k3", "r1204b", "r1204a10k3",
```

```
                 "r705ak2", "r705bk2", "r705ck2", "r705dk2", "r705ek2",
                 "r1124a", "r1124b", "r1124c", "r1124d", "r1123")

# load the data
podes_raw <- read_dta("podes06_merged.dta")

# transform some as numeric
podes_raw[,selected_var] <- lapply(podes_raw[,selected_var], as.numeric)

# generate predictor variable and filter data
podes_coded <- podes_raw %>%
  mutate(province = prop,
         urbanicity = r105a,
         has_govt_body = r302,
         near_sea = r304a,
         near_forest = r305,
         source_income = r402,
         hh_electricity = r501a,
         street_lighting = r502a,
         waste_disposal = r504,
         sanitation = r505,
         has_river_bank = r506a,
         has_luxury_res = r509a,
         has_slums = r509b,
         perc_hh_sa = r605/r401c*100,
         num_pop = r401a+r401b,
         prone_disaster = r512,
         any_pollution = ifelse((r510ak2 == 1 |
                                  r510bk2 == 1 |
                                  r510ck2 == 1 |
                                  r510dk2 == 1), 1, 0),
         has_mining = r511,
         num_k12_edu = r601ak2 + r601ak3 + r601bk2 + r601bk3 +
           r601ck2 + r601ck3 + r601dk2 + r601dk3 + r601ek2 + r601ek3,
         num_high_edu = r601fk2 + r601fk3,
         num_health_facil = r603ak2 + r603bk2 + r603ck2 + r603dk2 +
           r603ek2 + r603fk2 + r603gk2 + r603hk2 + r603ik2,
         num_doctors = r604a1 + r604a2,
         any_poor_letter = ifelse(r606 > 0 , 1, 0),
         any_pandemic = ifelse((r607ak2 == 1 | r607bk2 == 1 |
                                 r607ck2 == 1 | r607dk2 == 1 |
                                 r607ek2 == 1), 1, 0),
         water_source = r608a,
         any_social_institution = ifelse((r704a1k2 == 1 | r704a2k2 == 1 |
                                 r704a3k2 == 1 | r704a4k2 == 1 |
                                 r704a5k2 == 1 | r704a6k2 == 1), 1, 0),
         type_road = ifelse(is.na(r901b1), 0, r901b1),
```

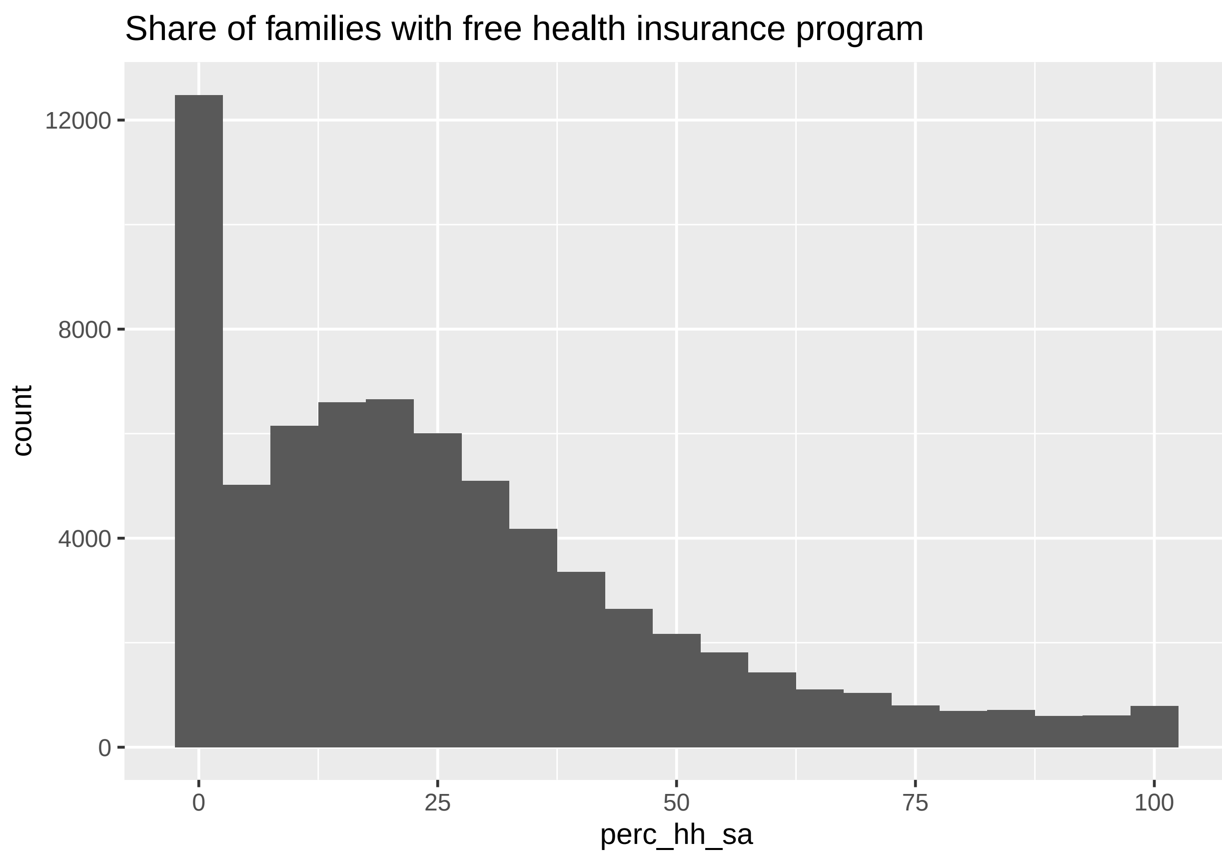
```r
         trafficability = ifelse(is.na(r901b2), 2, r901b2),
         distance_city = r902ak21,
         num_hh_cable = r904,
         signal_strength = r911,
         land_area = r10011,
         industrial_area = r1103,
         num_large_med_industry = r1106 + r1107,
         num_commercial_bank = r1119,
         num_village_bank = r1120a,
         most_type_crime = r1204b,
         any_increasing_crime = ifelse((r1204a1k3 == 3 |
                                          r1204a2k3 == 3 |
                                          r1204a3k3 == 3 |
                                          r1204a4k3 == 3 |
                                          r1204a5k3 == 3 |
                                          r1204a6k3 == 3 |
                                          r1204a7k3 == 3 |
                                          r1204a8k3 == 3 |
                                          r1204a9k3 == 3 |
                                          r1204a10k3 == 3), 1, 0),
         any_police_office = r1207bk2,
         disability = ifelse((r705ak2 == 1 |
                                r705bk2 == 1 |
                                r705ck2 == 1 |
                                r705dk2 == 1 |
                                r705ek2 == 1), 1, 0),
         credit_facilities = ifelse((r1124a == 1 |
                                       r1124b == 3 |
                                       r1124c == 5 |
                                       r1124d == 7), 1, 0),
         microfinance = r1123) %>%
  select(perc_hh_sa, urbanicity, has_govt_body, near_sea, near_forest,
         source_income,hh_electricity, street_lighting, waste_disposal, sanitation,
         has_river_bank, has_luxury_res, has_slums,
         num_pop, prone_disaster, any_pollution, has_mining,
         num_k12_edu, num_high_edu, num_health_facil, num_doctors,
         any_poor_letter, any_pandemic, water_source, any_social_institution,
         type_road, trafficability, distance_city, num_hh_cable,
         signal_strength, land_area, industrial_area, num_large_med_industry,
         province, num_commercial_bank, num_village_bank,
         most_type_crime, any_increasing_crime, any_police_office,
         credit_facilities, disability, microfinance) %>%
  mutate(any_increasing_crime = ifelse(is.na(any_increasing_crime), 0, 1),
         most_type_crime = ifelse(is.na(most_type_crime), 0, most_type_crime))

# numeric variable
numeric_var = c("perc_hh_sa", "num_pop", "num_k12_edu", "num_high_edu",
```

```
                "num_health_facil", "num_doctors", "distance_city",
                "num_hh_cable", "land_area", "num_large_med_industry",
                "num_commercial_bank", "num_village_bank")

# factor variable
factor_var = c("province", "urbanicity", "has_govt_body", "near_sea", "near_forest",
               "source_income", "hh_electricity", "street_lighting",
               "waste_disposal", "sanitation", "has_river_bank", "has_luxury_res",
               "has_slums", "prone_disaster", "any_pollution", "has_mining",
               "any_poor_letter", "any_pandemic", "water_source",
               "any_social_institution", "type_road",
               "signal_strength", "industrial_area", "most_type_crime",
               "any_increasing_crime", "any_police_office", "trafficability",
               "credit_facilities", "disability", "microfinance")


podes_coded[,numeric_var] <- lapply(podes_coded[,numeric_var], as.numeric)
podes_coded[,factor_var] <- lapply(podes_coded[,factor_var], factor)

saveRDS(podes_coded, "podes_coded.RDS")
```

2.1. Outcome Variables

In this analysis, we use `perc_hh_sa` as our outcome variable which indicates the share of households in the village who receive government's free health insurance program. This variable is derived from two variable "Number of families that receive health insurance for poor families program in the last year" and "Number of families in the village". This variable takes into value 0-100 with 100 means that all families in the village are the beneficiaries of this program, which is possible to happen in a small and poor villages.

```
ggplot() + geom_histogram(data = podes_coded, aes(perc_hh_sa), binwidth = 5) +
  ggtitle("Share of families with free health insurance program")
```



2.2. Predictors

In this analysis we decided to use 38 variables as predictors that includes continous and categorical variables. When choosing the variables we were thinking of variables that we believe can be used to predict poverty in a village.

No.	Variable	Type
1	Province code	Categorical
2	Urbanicity	Binary
3	Village Has Representative Body	Binary
4	Near from the sea	Binary
5	Within/Outside of a forested area	Categorical
6	No. of villagers	Continuous
7	Source of income (e.g. agriculture, business, service)	Categorical
8	Has houses with electricity	Binary
9	Has street lighting	Binary
10	Waste disposal method	Categorical
11	Most used sanitation method	Categorical
12	Near/far from river bank	Binary
13	Has luxury residences	Binary
14	Has slums	Binary
15	Environmental pollution in the past year	Binary

No.	Variable	Type
16	Mining activities	Binary
17	Prone to disaster	Binary
18	No. of K-12 institutions	Continuous
19	No. of higher education institutions	Continuous
20	No. of hospitals and other healthcare facilities	Continuous
21	No. of doctors	Continuous
22	Any letter to indicate poverty issued withing the past year	Binary
23	Presence of pandemic in the past year	Binary
24	Water source	Categorical
25	Any social institutions	Binary
26	Type of road surface	Categorical
27	Vehicle trafficability	Binary
28	Distance to capital district	Continuous
29	Handphone signal strength	Categorical
30	Number of households with cable subscriptions	Continuous
31	Land area	Continuous
32	Any industrial Area	Binary
33	No. of large/medium industries	Continuous
34	No. of commercial banks	Continuous
35	No. of village bank	Continuous
36	Most common type of crime	Categorical
37	Any crime with increasing trend compared to last year	Binary
38	Any police office in the village	Binary

3 Data Mining

In this analysis, we use `tidymodels` package to

```
set.seed(2112)
podes <- readRDS("podes_coded.RDS")
podes_split0 <- initial_split(podes, prob = 0.50, strata = province)
podes_sample <- training(podes_split0)

podes_split1 <- initial_split(podes_sample, prob = 0.70, strata = province)
podes_train <- training(podes_split1)
podes_test  <- testing(podes_split1)
```

```
podes_recipe <-
  recipe(perc_hh_sa ~ ., data = podes_train) %>%
  step_dummy(all_nominal_predictors()) %>%
  step_center(all_numeric_predictors()) %>%
  step_scale(all_numeric_predictors()) %>%
  prep()
```

3.1. Linear Regression

```r
lm_model <-
  linear_reg() %>%
  set_engine("lm")

lm_workflow <-
  workflow() %>%
  add_model(lm_model) %>%
  add_recipe(podes_recipe)

lm_fit <- fit(lm_workflow, data = podes_train)

bind_cols(podes_test,
          predict(lm_fit, new_data = podes_test)) %>%
  rmse(truth = podes_test$perc_hh_sa, estimate = .pred)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard        21.8
```

3.2. Regression with polynomial

```r
lm_model <-
  linear_reg() %>%
  set_engine("lm")

poly_recipe <-
  podes_recipe %>%
  step_poly(num_pop, degree = tune())

poly_bs <- bootstraps(podes_train, times = 10)
poly_grid <- tibble(degree = 1:5)

poly_wf <-
  workflow() %>%
  add_model(lm_model) %>%
  add_recipe(poly_recipe)

results <- tune_grid(poly_wf, resamples = poly_bs, grid = poly_grid)
```

```r
(best <- select_best(results, "rmse"))
```

```
## # A tibble: 1 x 2
##   degree .config
##    <int> <chr>
## 1      5 Preprocessor5_Model1
```

```
poly_recipe <-
  podes_recipe %>%
  step_poly(num_pop, degree = 5)

linear_wf <-
  workflow() %>%
  add_model(lm_model) %>%
  add_recipe(poly_recipe)

linear_fit <- fit(linear_wf, podes_train)

bind_cols(podes_test,
          predict(linear_fit, new_data = podes_test)) %>%
  rmse(truth = perc_hh_sa, estimate = .pred)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard        21.6
```

3.3. Elastic Net

```
bake_train <- bake(podes_recipe, new_data = NULL)
bake_test <- bake(podes_recipe, new_data = podes_test)

podes_rs <- bootstraps(bake_train, times = 10)

glmnet_model <-
  linear_reg(penalty = tune(), mixture = tune()) %>%
  set_engine("glmnet")

glmnet_wf <-
  workflow() %>%
  add_model(glmnet_model) %>%
  add_recipe(recipe(perc_hh_sa ~ ., data = bake_train))

glmnet_grid <- grid_regular(parameters(glmnet_model), levels = 10)

glmnet_results <- tune_grid(glmnet_wf, resamples = podes_rs, grid = glmnet_grid)
```

```
best <-
  glmnet_results %>%
  select_best("rmse")

final_wf <- finalize_workflow(glmnet_wf, best)

tuned_fit <- fit(final_wf, data = bake_train)

bind_cols(bake_test,
          predict(tuned_fit, new_data = bake_test)) %>%
  rmse(truth = bake_test$perc_hh_sa, estimate = .pred)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard        21.8
```

3.4. Random Forest

```
rf_model <- rand_forest() %>%
  set_engine("randomForest",
             num.threads = parallel::detectCores(),
             importance = TRUE,
             verbose = TRUE) %>%
  set_mode("regression") %>%
  set_args(trees = 100)

rf_wf <- workflow() %>%
  add_model(rf_model) %>%
  add_recipe(podes_recipe)

rf_fit <- fit(rf_wf, podes_train)
```

```
predict(rf_fit, new_data = podes_test) %>%
  bind_cols(podes_test) %>%
  rmse(truth = perc_hh_sa, estimate = .pred)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard        20.0
```

3.5. Nearest Neighbor

```
knn_model <-
  nearest_neighbor(neighbors = tune()) %>%
  set_engine("kknn") %>%
  set_mode("regression")

knn_grid <- tibble(neighbors = 1:5)

knn_wf <-
  workflow() %>%
  add_model(knn_model) %>%
  add_recipe(podes_recipe)

knn_rs <- bootstraps(podes_train, times = 5)

knn_results <- tune_grid(knn_wf, resamples = knn_rs, grid = knn_grid)
```

```
(best <- knn_results %>% select_best("rmse"))
```

```
## # A tibble: 1 x 2
##   neighbors .config
##       <int> <chr>
## 1         5 Preprocessor1_Model5
```

```
knn_model <-
  nearest_neighbor(neighbors = 5) %>%
  set_engine("kknn") %>%
  set_mode("regression")

knn_wf <-
  workflow() %>%
  add_model(knn_model) %>%
  add_recipe(podes_recipe)

knn_fit <- fit(knn_wf, data = podes_train)
```

```
bind_cols(podes_test,
          predict(knn_fit, new_data = podes_test)) %>%
  rmse(truth = perc_hh_sa, estimate = .pred)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard        23.5
```

3.6. GAM

```
library(additive)

GAM <-
  additive() %>%
  set_engine("mgcv") %>%
  set_mode("regression")

GAM_wf <-
  workflow() %>%
  add_model(GAM, formula = perc_hh_sa ~ s(num_pop, land_area)) %>%
  add_recipe(podes_recipe)

GAM_fit <- fit(GAM_wf, data = podes_train)
```

```
bind_cols(podes_test,
          predict(GAM_fit, new_data = podes_test)) %>%
  rmse(truth = perc_hh_sa, estimate = .pred)
```

```
## # A tibble: 1 x 3
##   .metric .estimator .estimate
##   <chr>   <chr>          <dbl>
## 1 rmse    standard        23.5
```