electionproject

2024-05-18

```r
#loading packages
pacman::p_load(skimr, tidyverse, ggplot2, rsample, tidymodels, dials)
```

```r
#loading data
elect_data = read.csv('election-2016.csv')

#remove unneeded columns
elect_data = subset(elect_data, select = -c(county, state))

#data with pop < 5,000,000
elect_data_small = elect_data |>
  filter(pop < 5000000)
#brief look at the data
#checking data info
nrow(elect_data)
```

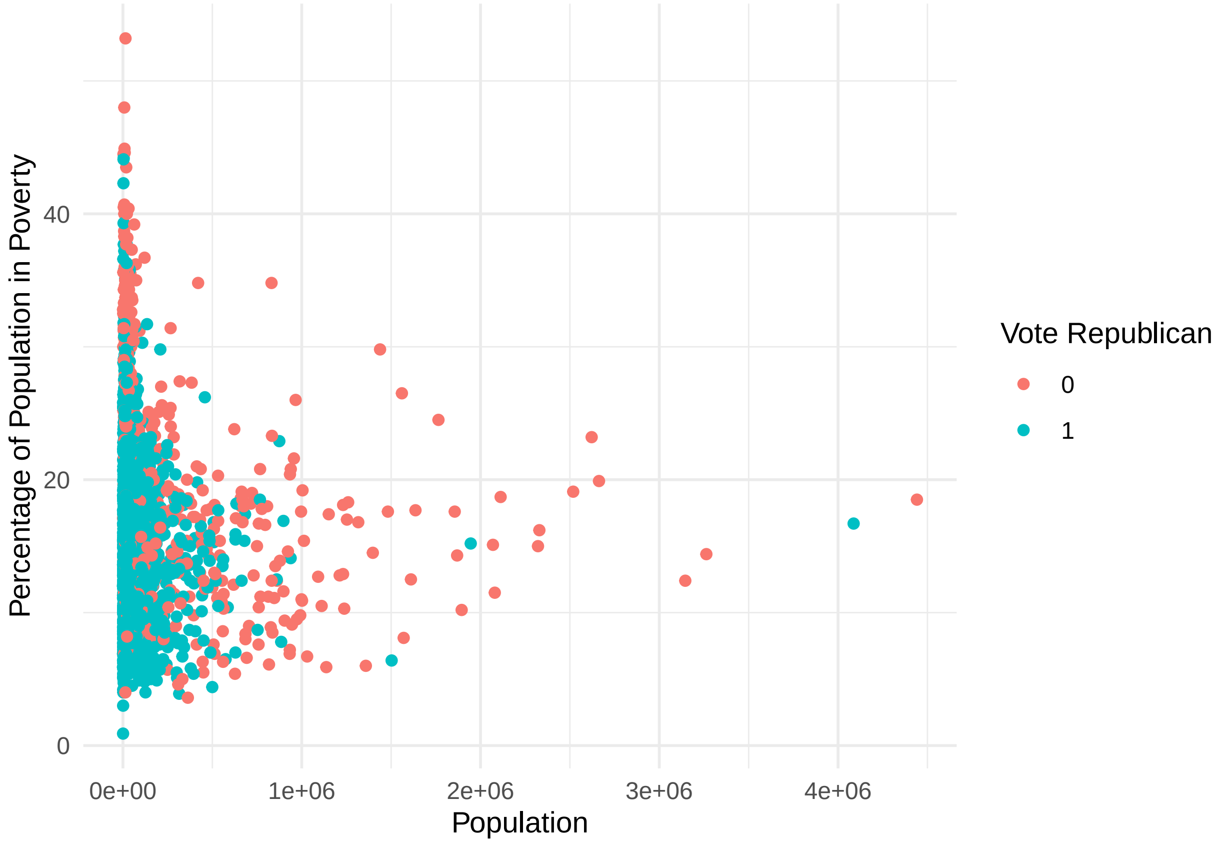
```
## [1] 3116
```

```r
length(elect_data$pop)
```

```
## [1] 3116
```

```r
#simple graph for anaylsis
poverty_graph_small <- ggplot(data = elect_data_small, aes(x = pop, y = pop_pct_poverty, color = factor
  geom_point() +
  labs(x = 'Population', y = "Percentage of Population in Poverty", color = 'Vote Republican') +
  theme_minimal() +
  scale_x_continuous() +
  scale_y_continuous()

poverty_graph_small
```



```
#preparing LASSO model
#5-fold cross validation
#set seed
set.seed(1234)
#5-fold CV on training dataset
elect_cv = elect_data %>% vfold_cv(v = 5)
#view CV
elect_cv %>% tidy()
```

```
## # A tibble: 15,580 x 3
##       Row Data     Fold
##     <int> <chr>    <chr>
##  1      1 Analysis Fold2
##  2      1 Analysis Fold3
##  3      1 Analysis Fold4
##  4      1 Analysis Fold5
##  5      2 Analysis Fold1
##  6      2 Analysis Fold2
##  7      2 Analysis Fold3
##  8      2 Analysis Fold5
##  9      3 Analysis Fold1
## 10      3 Analysis Fold2
## # i 15,570 more rows
```

```r
#first step in the recipe
recipe_all = recipe(i_republican_2016 ~ ., data = elect_data)

#Smashing the whole thing together
elect_recipe = recipe_all %>%
  step_impute_mean(everything() & - fips & - i_republican_2016 & - i_republican_2012) %>% # Impute all
  step_scale(everything() & - fips & - i_republican_2016 & - i_republican_2012) # Standardize all excep
print(elect_recipe)
```

```
##

## -- Recipe ----------------------------------------------------------------------

##

## -- Inputs

## Number of variables by role

## outcome:    1
## predictor: 30

##

## -- Operations

## * Mean imputation for: everything() & -fips & -i_republican_2016 &
##   -i_republican_2012

## * Scaling for: everything() & -fips & -i_republican_2016 & -i_republican_2012
```

```r
#define range of lambdas (glmnet wants decreasing range)
lambdas = 10^seq(from = 5, to = -2, length = 100)

#defining model
lasso_est = linear_reg(penalty = tune(), mixture = 1) %>% set_engine('glmnet')

#defining workflow
lasso_workflow = workflow() |>
  add_model(lasso_est) |>
  add_recipe(elect_recipe)

#CV w/range of lambdas
lasso_cv =
  lasso_workflow %>%
  tune_grid(
    resamples = vfold_cv(elect_data, v = 5),
    grid = data.frame(penalty = lambdas),
    metrics = metric_set(rmse)
  )
#show best models
lasso_cv %>% show_best()
```

```
## Warning in show_best(.): No value of ‘metric‘ was given; "rmse" will be used.
```

```
## # A tibble: 5 x 7
##   penalty .metric .estimator  mean     n std_err .config
##     <dbl> <chr>   <chr>      <dbl> <int>   <dbl> <chr>
## 1  0.01   rmse    standard   0.205     5 0.00282 Preprocessor1_Model001
## 2  0.0118 rmse    standard   0.206     5 0.00281 Preprocessor1_Model002
## 3  0.0138 rmse    standard   0.206     5 0.00279 Preprocessor1_Model003
## 4  0.0163 rmse    standard   0.207     5 0.00277 Preprocessor1_Model004
## 5  0.0192 rmse    standard   0.208     5 0.00276 Preprocessor1_Model005
```

```
#lowest RMSE ~0.205 @ lambda = 0.01
#fitting final model
lasso_final = glmnet(
  x = elect_data %>% dplyr::select(-i_republican_2016, -fips) %>% as.matrix(),
  y = elect_data$i_republican_2016,
  standardize = F,
  alpha = 1,
  lambda = 0.01
)
```

```
#creating elasticnet crossvalidation model
#defining elasticnet model
elas_est = linear_reg(penalty = tune(), mixture = tune()) |> set_engine('glmnet')

#creating elasticnet workflow
elas_workflow = workflow() |>
  add_model(elas_est) |>
  add_recipe(elect_recipe)

#tuning an elasticnet model
#creating tuning range
tuning_grid = grid_regular(penalty(), mixture(), levels = 50)

#running 5Fold CV with tuning range
elas_cv =
  elas_workflow |>
  tune_grid(
    resamples = elect_cv,
    grid = tuning_grid,
    metrics = metric_set(rmse)
  )
elas_cv |> show_best()
```

```
## Warning in show_best(elas_cv): No value of ‘metric‘ was given; "rmse" will be
## used.
```

```
## # A tibble: 5 x 8
##   penalty mixture .metric .estimator  mean     n std_err .config
##     <dbl>   <dbl> <chr>   <chr>      <dbl> <int>   <dbl> <chr>
## 1 0.00222   0.694 rmse    standard   0.204     5 0.00231 Preprocessor1_Model1737
## 2 0.00222   0.673 rmse    standard   0.204     5 0.00230 Preprocessor1_Model1687
## 3 0.00222   0.714 rmse    standard   0.204     5 0.00232 Preprocessor1_Model1787
## 4 0.00222   0.735 rmse    standard   0.204     5 0.00233 Preprocessor1_Model1837
## 5 0.00222   0.653 rmse    standard   0.204     5 0.00229 Preprocessor1_Model1637
```

```
#change data type for log regression
elect_data$i_republican_2016 = as.factor(elect_data$i_republican_2016)

# Model definition (using logistic_reg())
log_est = logistic_reg() %>% set_engine('glm')  # Logistic regression engine

# Workflow creation
log_workflow = workflow() %>%  # Create an empty workflow
  add_model(log_est) %>%  # Add the defined model (log_est)
  add_recipe(elect_recipe)  # Add the pre-defined recipe (elect_recipe)

#creating metrics
metrics = metric_set(yardstick::accuracy, yardstick::precision, yardstick::specificity, yardstick::sens

# Fit model with 5-fold cross-validation and record metrics
log_cv <- log_workflow %>%
  fit_resamples(
    resamples = vfold_cv(elect_data, v = 5),
    metrics = metrics
  )
```

```
## > A | warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## There were issues with some computations   A: x1                                                 > B
## There were issues with some computations   A: x1There were issues with some computations   A: x1   B
```

```
log_cv$.metrics
```

```
## [[1]]
## # A tibble: 5 x 4
##   .metric     .estimator .estimate .config
##   <chr>       <chr>          <dbl> <chr>
## 1 accuracy    binary         0.968 Preprocessor1_Model1
## 2 precision   binary         0.89  Preprocessor1_Model1
## 3 specificity binary         0.979 Preprocessor1_Model1
## 4 sensitivity binary         0.908 Preprocessor1_Model1
## 5 roc_auc     binary         0.992 Preprocessor1_Model1
##
## [[2]]
## # A tibble: 5 x 4
##   .metric     .estimator .estimate .config
##   <chr>       <chr>          <dbl> <chr>
## 1 accuracy    binary         0.979 Preprocessor1_Model1
## 2 precision   binary         0.922 Preprocessor1_Model1
## 3 specificity binary         0.985 Preprocessor1_Model1
## 4 sensitivity binary         0.95  Preprocessor1_Model1
## 5 roc_auc     binary         0.996 Preprocessor1_Model1
##
## [[3]]
## # A tibble: 5 x 4
##   .metric     .estimator .estimate .config
##   <chr>       <chr>          <dbl> <chr>
## 1 accuracy    binary         0.974 Preprocessor1_Model1
## 2 precision   binary         0.923 Preprocessor1_Model1
## 3 specificity binary         0.987 Preprocessor1_Model1
## 4 sensitivity binary         0.903 Preprocessor1_Model1
## 5 roc_auc     binary         0.990 Preprocessor1_Model1
##
## [[4]]
## # A tibble: 5 x 4
##   .metric     .estimator .estimate .config
##   <chr>       <chr>          <dbl> <chr>
## 1 accuracy    binary         0.963 Preprocessor1_Model1
## 2 precision   binary         0.870 Preprocessor1_Model1
## 3 specificity binary         0.973 Preprocessor1_Model1
## 4 sensitivity binary         0.913 Preprocessor1_Model1
## 5 roc_auc     binary         0.979 Preprocessor1_Model1
##
## [[5]]
## # A tibble: 5 x 4
##   .metric     .estimator .estimate .config
##   <chr>       <chr>          <dbl> <chr>
## 1 accuracy    binary         0.974 Preprocessor1_Model1
## 2 precision   binary         0.902 Preprocessor1_Model1
## 3 specificity binary         0.981 Preprocessor1_Model1
## 4 sensitivity binary         0.939 Preprocessor1_Model1
## 5 roc_auc     binary         0.985 Preprocessor1_Model1
```

```
log_cv |> show_best()
```

```
## Warning in show_best(log_cv): No value of ‘metric‘ was given; "accuracy" will
## be used.
```

```
## # A tibble: 1 x 6
##   .metric  .estimator  mean     n std_err .config
##   <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy binary     0.972     5 0.00280 Preprocessor1_Model1
```

```
#creating a logistic lasso regression
log_lasso = log_workflow |>
  fit_resamples(
    resamples = vfold_cv(elect_data, v = 5),
    metrics = metrics,
    grid = data.frame(penalty = lambdas),
  )
```

```
## Warning: The ‘...‘ are not used in this function but one or more objects were
## passed: ’grid’
```

```
## > A | warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
## There were issues with some computations   A: x1                                                 > B
## There were issues with some computations   A: x1There were issues with some computations   A: x2   B
```

```
log_lasso$.metrics
```

```
## [[1]]
## # A tibble: 5 x 4
##   .metric     .estimator .estimate .config
##   <chr>       <chr>          <dbl> <chr>
## 1 accuracy    binary         0.979 Preprocessor1_Model1
## 2 precision   binary         0.935 Preprocessor1_Model1
## 3 specificity binary         0.986 Preprocessor1_Model1
## 4 sensitivity binary         0.944 Preprocessor1_Model1
## 5 roc_auc     binary         0.987 Preprocessor1_Model1
##
## [[2]]
## # A tibble: 5 x 4
##   .metric     .estimator .estimate .config
##   <chr>       <chr>          <dbl> <chr>
## 1 accuracy    binary         0.963 Preprocessor1_Model1
## 2 precision   binary         0.864 Preprocessor1_Model1
## 3 specificity binary         0.971 Preprocessor1_Model1
## 4 sensitivity binary         0.922 Preprocessor1_Model1
## 5 roc_auc     binary         0.981 Preprocessor1_Model1
##
## [[3]]
## # A tibble: 5 x 4
##   .metric     .estimator .estimate .config
##   <chr>       <chr>          <dbl> <chr>
## 1 accuracy    binary         0.966 Preprocessor1_Model1
## 2 precision   binary         0.89  Preprocessor1_Model1
## 3 specificity binary         0.979 Preprocessor1_Model1
## 4 sensitivity binary         0.899 Preprocessor1_Model1
## 5 roc_auc     binary         0.990 Preprocessor1_Model1
##
## [[4]]
## # A tibble: 5 x 4
##   .metric     .estimator .estimate .config
##   <chr>       <chr>          <dbl> <chr>
## 1 accuracy    binary         0.970 Preprocessor1_Model1
## 2 precision   binary         0.863 Preprocessor1_Model1
## 3 specificity binary         0.976 Preprocessor1_Model1
## 4 sensitivity binary         0.932 Preprocessor1_Model1
## 5 roc_auc     binary         0.995 Preprocessor1_Model1
##
## [[5]]
## # A tibble: 5 x 4
##   .metric     .estimator .estimate .config
##   <chr>       <chr>          <dbl> <chr>
## 1 accuracy    binary         0.968 Preprocessor1_Model1
## 2 precision   binary         0.879 Preprocessor1_Model1
## 3 specificity binary         0.977 Preprocessor1_Model1
## 4 sensitivity binary         0.916 Preprocessor1_Model1
## 5 roc_auc     binary         0.992 Preprocessor1_Model1
```

```r
log_lasso |> show_best()
```

```
## Warning in show_best(log_lasso): No value of ‘metric‘ was given; "accuracy"
## will be used.
```

```
## # A tibble: 1 x 6
##   .metric  .estimator  mean     n std_err .config
##   <chr>    <chr>      <dbl> <int>   <dbl> <chr>
## 1 accuracy binary     0.969     5 0.00271 Preprocessor1_Model1
```