

Classification EDA

2024-07-30

Set-Up

To run code when knitting set `eval = TRUE`, to not run code for while knitting the PDF set `eval = FALSE`.

```
train <- read_csv("Data/train_class.csv")

## Rows: 2331 Columns: 126
## -- Column specification -----
## Delimiter: ","
## chr (2): winner, name
## dbl (124): id, total_votes, x0001e, x0002e, x0003e, x0005e, x0006e, x0007e, ...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
test <- read_csv("Data/test_class.csv")

## Rows: 780 Columns: 124
## -- Column specification -----
## Delimiter: ","
## dbl (124): id, total_votes, x0001e, x0002e, x0003e, x0005e, x0006e, x0007e, ...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

CSV Analysis

Summary

The most important takeaways are:

- We need to deal with incomplete rows, as data on income/GDP is not available for every county
- Like with the regression data, we need to deal with interaction effects within the data
- May be a good idea to use log transformation on some data (e.g. `total_votes`)
- As we can see in the graphs, some of the data is skewed
 - E.g. Biden has a disproportional amount of votes in most urban counties compared to the bars

Feature Groups

After looking at the CSVs, I noticed that like with the regression project data, a lot of the features can be grouped together in as demographics. Each column represents the total population of each sub-demographic. For instance, there are several features that represent the total population of certain age ranges, income levels, etc. Furthermore, like with the regression project, we may need to check for interaction effects of these columns. Here are all of the columns that can be grouped together:

Columns	Demographic Category (in total pop.)
x0002E:x0003E	Represents total population of men and women respectively
x0005E:x0017E	Represents age range with each column being 5 year intervals
x0019E:x0024E*	Represents population above certain ages (e.g. > 18, > 21, etc.)
x0025E:x0031E**	Represents population above certain ages per gender (e.g. > 18 men, > 18 women, etc.)
x0034E:x0057E	Racial and ethnicity demographics
x0058E:x0069E	Multiracial combination demographics
x0071E:x0075E	Demographics of Hispanic or Latino
x0076E:x0085E	Demographics of Non-Hispanic or Latino Races
x0087E:0089E	Citizens
C01_001E:C01_027E	Education levels of certain age groups (e.g. Bachelor's 18-24, High school graduates 25-34, etc.)
income_per_cap_2016:income_per_cap_2020	income per capita for county in year
gdp_2016:gdp_2020	GDP for the county in year

If this data is anything like the data in the regression project, then we'll need to account for interaction effects. Also I realized all of these columns are actually lower case in the dataset but was capitalized in the `col_descriptions.csv` file, so in the actual code just un-capitalize the column names.

Note that some columns seem similar if not identical to other columns. for instance, `x0021E` and `x0024E` are both populations of 18 and over, however `x0021E` should be grouped within its category while `x0024E` appears to be the total of `x0026E` and `x0027E`. Perhaps these duplicates can be deleted to avoid linear dependency. Moreover, some categories may be unnecessary and all those columns can be dropped altogether. For instance, I don't see how `x0019E:x0024E` would be useful especially when we already have age data. It's possible it may be a powerful feature, but it's something worth considering in my opinion.

The education-related columns are weird. We have age ranges of education as well as general age floors (e.g. we have the data of 25-34 year olds with bachelor's as well as how total population of those > 25 with a bachelor's). Moreover, information regarding associate's and graduate/professional degrees is inconsistent. We only have that specific level of information for the population of those 25 and older, we don't know the information regarding other groups (e.g. we don't know how many 35-44 year olds specifically have an associate's, only high school or bachelor's).

Data regarding income and GDP is missing for some counties, so this **must** be accounted for. Some ways to deal with this include but are not limited to: - Somehow extrapolate data for incomplete rows - Drop those rows completely Also, it's possible these rows could have high interaction effects as well, so there's a chance we may only need one of these categories.

Not every column groups are necessarily consecutively numbered. For instance, there does not exist `x0004E` or `x0032E`. Also, some columns are standalone features that can't be grouped into categories. Here are these columns:

Columns	Column Description (in total pop.)
id	not a predictor
winner	Biden or Trump
total_votes	total votes
name of county	not a predictor or in test set
x0001E	Total population
x0018E	Median age (years)
x0033E	Duplicate of x0001E
x0086E	Total housing units

Columns	Column Dsecription (in total pop.)
x2013_code	urban/rural code used by CDC with 1 being most urban 6 being most rural

Correlation Matrix

TODO: cells regarding income/GDP are NA because there is missing data, need to remake correlation matrix to account for this

First create correlation matrix to see which variables are highly correlated with each other
Because there's so many features and so many cells (127^2 to be exact), I decided to save the matrix

```
cor_mx <-
  train %>%
    select_if(is.numeric) %>%
    cor()

write.csv(cor_mx, 'cor_mx.csv')
```

Graphs



