PSTAT 131/231 Final Project (2021)

Due December 9, 2021, 7:00 PM PT

Instructions and Expectations

- You are allowed and encouraged to work with one partner on this project. Include your names, perm numbers, and whether you are taking the class for 131 or 231 credit.
- You are welcome and encouraged to write up your report as a research paper (e.g. abstract, introduction, methods, results, conclusion) as long as you address each of the questions below. Alternatively, you can format the assignment like a long homework by addressing each question in parts.
- All of your results should be formatted in a professional and visually appealing manner. That means, either as a polished visualization or for tabular data, a nicely formatted table.
- All R code should be available from your Rmarkdown file, but does not need to be shown in the body of the report! Use the chunk option `echo=FALSE` to exclude code from appearing in your write-up when necessary. In addition to your Rmarkdown, you should turn in the write-up as either a pdf document or an html file (both are acceptable).
- All files should be submitted electronically via GauchoSpace. See course syllabus for submission instructions.

In this project, we will study and analyze the Untied States county-level census data and county-level education data. In particular, our target would be to build and evaluate statistical machine learning models to understand some of the potential causes of poverty.

Data

Census data

We essentially start with the 2017 United States county-level census data, which is available here. This dataset contains many demographic variables for each county in the U.S.

We load in and clean the `census` dataset by transforming the full state names to abbreviations (to match the `education` dataset in later steps). Specifically, `R` contains default global variables `state.name` and `state.abb` that store the full names and the associated abbreviations of the **50** states. However, it does not contain `District of Columbia` (and the associated `DC`). We added it back manually since `census` contains information in DC. We further remove data from Purto Rico to ease the visualization in later steps.

```
state.name <- c(state.name, "District of Columbia")
state.abb <- c(state.abb, "DC")
## read in census data
census <- read_csv("./acs2017_county_data.csv") %>% select(-CountyId, -ChildPoverty, -Income, -IncomeErr, -Income
PerCap, -IncomePerCapErr) %>%
  mutate(State = state.abb[match(`State`, state.name)]) %>%
  filter(State != "PR")
```

Followings are the first few rows of the `census` data. The column names are all very self-explanatory:

State	County	TotalPop	Men	Women	Hispanic	White	Black	Native	Asian	Pacific	VotingAgeCitizen	Poverty	Professional
AL	Autauga County	55036	26899	28137	2.7	75.4	18.9	0.3	0.9	0	41016	13.7	35.3
AL	Baldwin County	203360	99527	103833	4.4	83.1	9.5	0.8	0.7	0	155376	11.8	35.7
AL	Barbour County	26201	13976	12225	4.2	45.7	47.8	0.2	0.6	0	20269	27.2	25.0
AL	Bibb County	22580	12251	10329	2.4	74.6	22.0	0.4	0.0	0	17662	15.2	24.4
AL	Blount County	57667	28490	29177	9.0	87.4	1.5	0.3	0.1	0	42513	15.6	28.5
AL	Bullock County	10478	5616	4862	0.3	21.6	75.6	1.0	0.7	0	8212	28.5	19.7

Education data

We also include the `education` dataset, available at Economic Research Service at USDA. The dataset contains county-level educational attainment for adults age 25 and older in 1970-2019. We specifically use educational attainment information for the time period of 2015-2019.

To clean the data, we remove uninformative columns (as in `FIPS Code`, `2003 Rural-urban Continuum Code`, `2003 Urban Influence Code`, `2013 Rural-urban Continuum Code`, and `2013 Urban Influence Code`). To be consistent with `census` data, we exclude data from Purto Rico and we rename `Area name` to `County` in order to match that in the `census` dataset.

```
## read in education data
education <- read_csv("./education.csv") %>%
  filter(!is.na(`2003 Rural-urban Continuum Code`)) %>%
  filter(State != "PR") %>%
  select(-`FIPS Code`,
         -`2003 Rural-urban Continuum Code`,
         -`2003 Urban Influence Code`,
         -`2013 Rural-urban Continuum Code`,
         -`2013 Urban Influence Code`) %>%
  rename(County = `Area name`)
```

Preliminary data analysis

1. (1 pts) Report the dimension of `census`. (1 pts) Are there missing values in the data set? (1 pts) Compute the total number of distinct values in `State` in `census` to verify that the data contains all states and a federal district.

2. (1 pts) Report the dimension of `education`. (1 pts) How many distinct counties contain missing values in the data set? (1 pts) Compute the total number of distinct values in `County` in `education`. (1 pts) Compare the values of total number of distinct county in `education` with that in `census`. (1 pts) Comment on your findings.

Data wrangling

3. (2 pts) Remove all NA values in `education`, if there is any.

4. (2 pts) In `education`, in addition to `State` and `County`, we will start only on the following 4 features: `Less than a high school diploma, 2015-19`, `High school diploma only, 2015-19`, `Some college or associate's degree, 2015-19`, and `Bachelor's degree or higher, 2015-19`. Mutate the `education` dataset by selecting these 6 features only, and create a new feature which is the total population of that county.

5. (3 pts) Construct aggregated data sets from `education` data: i.e., create a state-level summary into a dataset named `education.state`.

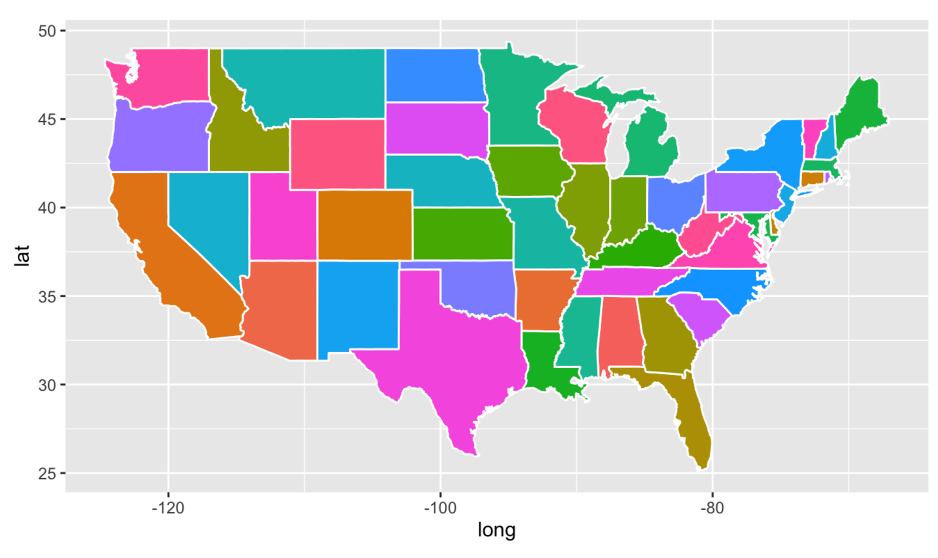
6. (4 pts) Create a data set named `state.level` on the basis of `education.state`, where you create a new feature which is the name of the education degree level with the largest population in that state.

Visualization

Visualization is crucial for gaining insight and intuition during data mining. We will map our data onto maps.

The R package `ggplot2` can be used to draw maps. Consider the following code.

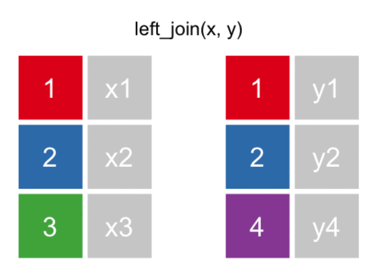
```
states <- map_data("state")

ggplot(data = states) +
  geom_polygon(aes(x = long, y = lat, fill = region, group = group),
               color = "white") +
  coord_fixed(1.3) +
  guides(fill=FALSE)  # color legend is unnecessary for this example and takes too long
```



The variable `states` contain information to draw white polygons, and fill-colors are determined by `region`.

7. (6 pts) Now color the map (on the state level) by the education level with highest population for each state. Show the plot legend.

First, combine `states` variable and `state.level` we created earlier using `left_join()`. Note that `left_join()` needs to match up values of states to join the tables. A call to `left_join()` takes all the values from the first table and looks for matches in the second table. If it finds a match, it adds the data from the second table; if not, it adds missing values:



Here, we'll be combing the two data sets based on state name. However, the state names in `states` and `state.level` can be in different formats: check them! Before using `left_join()`, use certain transform to make sure the state names in the two data sets: `states` (for map drawing) and `state.level` (for coloring) are in the same formats. Then `left_join()`.

8. (6 pts) (Open-ended) Create a visualization of your choice using `census` data. Use this R graph gallery for ideas and inspiration.

9. The `census` data contains county-level census information. In this problem, we clean and aggregate the information as follows. (4 pts) Start with `census`, filter out any rows with missing values, convert {`Men`, `Employed`, `VotingAgeCitizen`} attributes to percentages, compute `Minority` attribute by combining {Hispanic, Black, Native, Asian, Pacific}, remove these variables after creating `Minority`, remove {`Walk`, `PublicWork`, `Construction`, `Unemployment`}.
(Note that many columns are perfectly collineared, in which case one column should be deleted.)

10. (1 pts) Print the first 5 rows of `census.clean`

Dimensionality reduction

11. Run PCA for the cleaned county level census data (with `State` and `County` excluded). (2 pts) Save the first two principle components PC1 and PC2 into a two-column data frame, call it `pc.county`. (2 pts) Discuss whether you chose to center and scale the features before running PCA and the reasons for your choice. (2 pts) What are the three features with the largest absolute values of the first principal component? (2 pts) Which features have opposite signs and what does that mean about the correlation between these features?

12. (2 pts) Determine the number of minimum number of PCs needed to capture 90% of the variance for the analysis. (2 pts) Plot proportion of variance explained (PVE) and cumulative PVE.

Clustering

13. (2 pts) With `census.clean` (with `State` and `County` excluded), perform hierarchical clustering with complete linkage. (2 pts) Cut the tree to partition the observations into 10 clusters. (2 pts) Re-run the hierarchical clustering algorithm using the first 2 principal components from `pc.county` as inputs instead of the original features. (2 pts) Compare the results and comment on your observations. For both approaches investigate the cluster that contains *Santa Barbara County*. (2 pts) Which approach seemed to put Santa Barbara County in a more appropriate clusters? Comment on what you observe and discuss possible explanations for these observations.

Modeling

We start considering supervised learning tasks now. The most interesting/important question to ask is: *can we use census information as well as the education information in a county to predict the level of poverty in that county?*

For simplicity, we are interested in a binary classification problem. Specifically, we will transform `Poverty` into a binary categorical variable: high and low, and conduct its classification.

In order to build classification models, we first need to combine `education` and `census.clean` data (and removing all NAs), which can be achieved using the following code.

```
# we join the two datasets
all <- census.clean %>%
  left_join(education, by = c("State"="State", "County"="County")) %>%
  na.omit
```

14. (4 pts) Transform the variable `Poverty` into a binary categorical variable with two levels: 1 if Poverty is greater than 20, and 0 if `Poverty` is smaller than or equal to 20. Remove features that you think are uninformative in classfication tasks.

Partition the dataset into 80% training and 20% test data. Make sure to `set.seed` before the partition.

```
set.seed(123)
n <- nrow(all)
idx.tr <- sample.int(n, 0.8*n)
all.tr <- all[idx.tr, ]
all.te <- all[-idx.tr, ]
```

Use the following code to define 10 cross-validation folds:

```
set.seed(123)
nfold <- 10
folds <- sample(cut(1:nrow(all.tr), breaks=nfold, labels=FALSE))
```

Using the following error rate function. And the object `records` is used to record the classification performance of each method in the subsequent problems.

```
calc_error_rate = function(predicted.value, true.value){
  return(mean(true.value!=predicted.value))
}
records = matrix(NA, nrow=3, ncol=2)
colnames(records) = c("train.error","test.error")
rownames(records) = c("tree","logistic","lasso")
```

Classification

16. Decision tree: (2 pts) train a decision tree by `cv.tree()`. (2 pts) Prune tree to minimize misclassification error. Be sure to use the `folds` from above for cross-validation. (2 pts) Visualize the trees before and after pruning. (1 pts) Save training and test errors to `records` object. (2 pts) Interpret and discuss the results of the decision tree analysis. (2 pts) Use this plot to tell a story about Poverty.

16. (2 pts) Run a logistic regression to predict `Poverty` in each county. (1 pts) Save training and test errors to `records` variable. (1 pts) What are the significant variables? (1 pts) Are they consistent with what you saw in decision tree analysis? (2 pts) Interpret the meaning of a couple of the significant coefficients in terms of a unit change in the variables.

17. You may notice that you get a warning `glm.fit: fitted probabilities numerically 0 or 1 occurred`. As we discussed in class, this is an indication that we have perfect separation (some linear combination of variables *perfectly* predicts the winner).
This is usually a sign that we are overfitting. One way to control overfitting in logistic regression is through regularization.

(3 pts) Use the `cv.glmnet` function from the `glmnet` library to run a 10-fold cross validation and select the best regularization parameter for the logistic regression with LASSO penalty. Set `lambda = seq(1, 20) * 1e-5` in `cv.glmnet()` function to set pre-defined candidate values for the tuning parameter λ.

(1 pts) What is the optimal value of $\lambda\lambda$ in cross validation? (1 pts) What are the non-zero coefficients in the LASSO regression for the optimal value of $\lambda\lambda$? (1 pts) How do they compare to the unpenalized logistic regression? (1 pts) Comment on the comparison. (1 pts) Save training and test errors to the `records` variable.

18. (6 pts) Compute ROC curves for the decision tree, logistic regression and LASSO logistic regression using predictions on the test data. Display them on the same plot. (2 pts) Based on your classification results, discuss the pros and cons of the various methods. (2 pts) Are the different classifiers more appropriate for answering different kinds of questions about Poverty?

Taking it further

This part will be worth up to a 20% of your final project grade!

19. (9 pts) Explore additional classification methods. Consider applying additional *two* classification methods from KNN, LDA, QDA, SVM, random forest, boosting, neural networks etc. (You may research and use methods beyond those covered in this course). How do these compare to the tree method, logistic regression, and the lasso logistic regression?

20. (9 pts) Tackle *at least* one more interesting question. Creative and thoughtful analysis will be rewarded! Some possibilities for further exploration are:

- Bootstrap: Perform boostrap to generate plots similar to ISLR Figure 4.10/4.11. Discuss the results.
- Consider a regression problem! Use regression models to predict the actual value of `Poverty` (before we transformed `Poverty` to a binary variable) by county. Compare and contrast these results with the classification models. Which do you prefer and why? How might they complement one another?
- Instead of using the native attributes (the original features), we can use principal components to create new (and lower dimensional) set of features with which to train a classification model. This sometimes improves classification performance. Compare classifiers trained on the original features with those trained on PCA features.

21. (9 pts) (Open ended) Interpret and discuss any overall insights gained in this analysis and possible explanations. Use any tools at your disposal to make your case: visualize errors on the map, discuss what does/doesn't seems reasonable based on your understanding of these methods, propose possible directions (collecting additional data, domain knowledge, etc).