INFO370 Problem Set: Categorical variables and logistic regression

February 12, 2023

Instructions

This problem set revolves around logistic regression and categorical variables. It has two main goals:

1. Learn to handle categorical variables
2. Learn to use and interpret logistic regression results

Besides the main goals, Q1 involves some complex data preparation. You need to fill out missing values and combine entries across rows.

You are welcome to discuss and work together, but you still have to submit your own work! Please list your collaborators on the submission!

1 Who will win the elections? (60pt)

This question asks you to do a simple election model. We are looking for the U.S. 2020 presidential elections by counties. Your task is to model the winner (1/0 for democratic/republican candidate winning the presidential elections in this county), and explain the winner using population density, education level, income, and geographic differences (the census region).

The data file is called *us-elections_2000-2020.csv.bz2*. The variables are

FIPS county FIPS code (numeric county code)
year election year
state state name
state2 2-letter state code
region census region (west, midwest, south, northeast)
county county name
office President (we look only at presidential elections)
candidate name of the candidate
party party of the candidate
candidatevotes votes received by this candidate for this particular party
totalvotes total number of votes cast in this county-year
income personal income, USD/per capita (BEA—Bureau of Economic Analysis—data)
population population, census estimate (BEA data)
LND010200D land area (sq.mi) at 2000 (Census data)
EDU600209D Persons 25 years and over, total 2005-2009
EDU695209D Educational attainment - persons 25 years and over - bachelor's degree 2005-2009
POP010210D Resident population (April 1 - complete count) 2010
POP220210D Population of one race - White alone 2010 (complete count)

POP250210D Population of one race - Black or African American alone 2010 (complete count)
POP320210D Population of one race - Asian alone 2010 (complete count)
POP400210D Hispanic or Latino population 2010 (complete count)
PST110209D Resident total population estimate, net change - April 1, 2000 to July 1, 2009
BIRTHS2020 Births in period 4/1/2020 to 6/30/2020
DEATHS2020 Deaths in period 4/1/2020 to 6/30/2020

The obscure variable names are straight from the U.S. Census. Data is based on different sources, a few election files in GH, and the U.S. Census. See my data repo for some more (but incomplete) information and code.

1. (1pt) Load data, and do the basic checks.

You are going to work with 2020 data. However, some important information for 2020 is missing. First, we check how does missing data look like, and thereafter we fill it with the most recent values that are there.

2. (3pt) print the rows of the data frame from index 6264 to 6271 (i.e. these index values of the data frame). For simplicity, you may only include variables *fips*, *county*, *year* and *income*.

   Hint: check out the `.loc[]` attribute, see Python Notes 3.3.4.

3. (6pt) You see that some income values are missing in the example from question 2.

   (a) Which values do you expect to see instead of NA-s in lines 6266, 6267, 6268 and 6269?

   (b) How is it related to the non-missing income, county and fips values?

   (c) What method would you use to fill in the missings (what computer code and variables)?
   Hint: check out the `sort_values` and `DataFrame.fillna` method.

4. (4pt) Fill the missings in all columns you need (not only in income) with the most recent values that exist in the data. Ensure you do not fill missings with values from other counties.

   Hint: when grouping by *fips*, pandas will remove the variable, even if you set *as_index=False* for some reason. A simple way around is to create another variable, e.g. *fips2* = *fips*, and use that for grouping. In this way *fips* will be preserved.

5. (3pt) Print out the same lines you did above in 1.2. Does it look now what you expected?

   Hint: If you did this correctly, then even after filling in NA-s there are a few missing cases left.

Now it is time to select the neccesary data and create the features you need for the analysis below.

6. (10pt) Make a new data frame that only contains 2020 data, and that contains a binary variable: the democrats won in that county in 2020.

   Hint: You have to build that variable using two lines of data in the original data frame *by FIPS* after the data is ordered by year. The original data contains two lines for each county, one for democrats and one for republicans. They contain the party-specific number of votes but are otherwise similar. You may extract the rows for democrats, the rows for republicans,

and then just compare these two rows county-wise to see who won there. Note that it is *not enough* to just check if democrats/republicans got more than 50% of votes.

However, when you extract the vote numbers, it will be a series with an index. You may want to either reset the index (see examples in Combining data into data frames) in Python Notes, or convert the series into a numpy array with the `.values` attribute.

7. (6pt) Create auxiliary variables: population density (population divided by land area); and percentage of college graduates.

   Hint: there are several population measures. You can use any of those, the results will be slighlty different but as population is not changing fast, it should not have much of an impact.

8. (4pt) But are "younger" counties with more births voting differently that "old" counties with a lots of deaths? Compute (estimate) *yearly* birth rate and death rate as number of births per 1000 of population.

9. (2pt) Ensure that the variables you are going to use are in a reasonable range!

   Hint: there are values that do not make sense. Use `min` and `max` to check what is the range in data, and then find offending values and remove those.

10. (8pt) Estimate logistic regression model where you explain democrats' winning with population density, education level, income, and census region.

11. (5pt) Why do we use logistic regression here, instead of linear regression?

12. (8pt) Interpret the results. Which results are statistically significant?

    Note: you may want to change some of the units, e.g. you may want to measure population density in 1000/per sq mi, instead of persons per sq mi.

    Hint: your estimate for North-East should be ~ 0.11 and for South it should be ~ 0.06. The exact value depends on the exact steps you took for cleaning data, and how exactly do you define the variables you engineer.

---

2 Model AirBnB Price (40pt)

Your next task is to analyze the Beijing AirBnB listing price (variable *price*) in Beijing. It is downloaded from Inside Airbnb but we suggest to use the version on canvas (*airbnb-beijing-listings.csv*). You have to work with several sorts of categorical variables, including those that contain way too many too small categories. You are also asked to do log-transforms and interpret the results.

1. (2pt) Load data. Select only relevant variables you need below. Even better, check out the `usecols` argument for `read_csv`. Do basic checks.

2. (5pt) Do the basic data cleaning:

   (a) convert *price* to numeric.

   (b) remove entries with missing or invalid price, bedrooms, and other variables you need below

3. (4pt) Analyze the distribution of *price*. Does it look like normal? Does it look like something else? Does it suggest you should do a log-transformation?

   Hint: consult lecture notes Section 4.1.8 *Interactions and Feature Transformations*.

4. (6pt) Convert the number of bedrooms into another variable with a limited number of categories only, such as 0, 1, 2, 3, 4+, and use these categories in the models below.

   Hint: consult Python Notes Section on cleaning data.

5. (6pt) Run a linear regression where you explain the listing price with number of bedrooms where *bedrooms* uses these categories. Interpret the results, including R^2.

   Hint: if 0-BR is the reference category, the effect for 1BR should be -11.62 (but it depends on how exactly did you clean data).

6. (8pt) Now repeat the process with the model where you analyze log price instead of price. Interpret the results. Which model behaves better in the sense of R^2?

   Hint: if you cleaned the data the same way as me, you should see $R^2 = 0.32$.

   For the following task use either *log(price)* or *price*, depending on your answer here.

7. (9pt) Finally we just add three more variables to the model: *room type*, *accommodates*, and *bathrooms*. While room type only contains three values, the other two contain many different categories. Recode these as

   - accommodates: “1”, “2”, “3”, “4 and more”
   - bathrooms: “0”, “1”, “2”, “3 and more”, where the 0.5 is rounded up to the next integer, e.g. 0.5 becomes 1, and 1.5 becomes 2.

   Run this model. Interpret and comment the more interesting/important results. Do not forget to mention what are the relevant reference categories and R^2.

Finally tell us how many hours did you spend on this PS.