American University - DATA 312

Instructions for Homework 4-C = Project 4 = Final Project

Unit 4 – Machine Learning

Pre-requisite work:
Please complete Homework 4-B before starting this project. You have all the libraries that you will need already installed for this project! You will likely need

- tidyverse
- modelr
- e1071
- usmap

Objectives:

The purpose of this assignment is to bring together the different tools you've learned in this course into a more comprehensive analysis.

Data source:

The data were collected and merged by Prof. Dietz from the following sources:

https://github.com/tonmcg/US_County_Level_Election_Results_08-20 (Voting)

https://usafacts.org/visualizations/coronavirus-covid-19-spread-map/ (Covid)

https://www.ers.usda.gov/data-products/county-level-data-sets/download-data/ (Education, etc)

https://www.census.gov/data/tables/time-series/demo/popest/2010s-counties-detail.html (Racial/Pop.)

Data are broken up by County. Some counties might not be present if they failed to appear in any of the original data sets. This mostly applies to Alaska.

Data by county includes some columns which are obvious and some which require explanation. Any columns not explained here may be researched at the original data source's website.

Unemployment_rate_2019 is out of the total possible number of employed persons, reported in the column Civilian_labor_force_2019.

Med_HH_Income_Percent_of_State_Total means that the value of MHHI (Median Household Income) has been divided by the median of the entire state and turned into a percent. Thus, if this value is 100, it is right at that state's median income level.Rural_urban_Continuum_Code (two are included) indicates which areas contain more people. Smaller values indicate larger urban areas, while larger values are for more rural areas.

PCTPOVALL gives overall percent poverty, while PCTPOV017 is for children and

PCTPOV517 is for school-aged children.
TOT_POP is the total raw population.
WA_MALE (FEMALE) gives “White alone” males (females).
BA_MALE (FEMALE) gives “Black alone” males (females).
H_MALE (FEMALE) gives “Hispanic” males (females).

The “Hispanic” indication is separate from racial classification, thus it can only be used to get a rough idea of what a person identifies as. Note: NHWA means “Non-Hispanic White Alone”.

IA_MALE(FEMALE) gives the count of males (females) who are Native American.
AA_MALE (FEMALE) is for Asian descent.

X2021.03.02 (or similar) means this is the Covid death count as of 3/2/2021 in that county.

votes_gop, votes_dem, total_votes gives Republican votes cast for President in 2020, Democrat votes, and finally total votes.

Work process:

DATA PREPARATION AND CLEANING

1. Download the EduUnempPovPopCovidVoting_StatCrunchV2.csv file from Canvas.

2. Read the file into R. Each row is a county in the US, with many columns of interesting facts about them

```
megadata <- read_csv(' EduUnempPovPopCovidVoting_StatCrunchV2.csv')
```

(Call the table whatever you like, of course!)

3. Just as a refresher from earlier in the semester, it's easy to use the usmap library to plot. Now that you know about ggplot(), it's worth noting that plot_usmap() understands ggplot() options, but it's not quite tidyverse-compatible. In particular, it doesn't like the fact that in our dataset, the county FIPS code is not called `fips`. It's easy to temporarily fix that if you want to plot a map quickly:

```
temporary <- megadata %>% mutate(fips=countyFIPS)
plot_usmap(regions='counties',
  data=temporary,
  values='TOT_POP',
  include=c('CT','MA','RI'))
```

Don't get too distracted by the plots, but please include one or two relevant maps in your report.

4. Look over the columns of `megadata` to get an idea of what's in there. Pay particular attention to the types of the data in the columns: most of the columns are quantitative, but a few are categorical. Most of the columns are pretty self-explanatory, though a few are less obvious. Explanations are on the first page of this assignment.

5. A caution is in order. When R reads the CSV file, it makes some assumptions about the types of values in each column. Sometimes it is correct, but sometimes it is not. For instance, counts are usually typed correctly:

```
is.numeric(megadata$X2020.06.01) # Is the number of COVID deaths a quantitative variable?
```

You should get TRUE, which means all is well. Also, you should get FALSE for the Postal Code

```
is.numeric(megadata$PostalCode)
```

However, R thinks that FIPS is a number… it's actually categorical

```
is.numeric(megadata$countyFIPS) # Returns TRUE but it really shouldn't be!
```

You'll occasionally get weird results because of this. You can easily fix it using

```
megadata <- megadata %>% mutate(countyFIPS=as.factor(countyFIPS))
```

This also is true of other columns. You can get R's opinion of the types of all columns at once

```
map(megadata,is_numeric)
```

and then comb through and fix any that are wrong by hand. If you don't, you may get a nasty surprise later…

6. Let's get to business and make our sampling frame. This presents no surprises:

```
raw_data_samplingframe <- megadata %>%
  mutate(snum=sample.int(n(),n())/n())

training <- raw_data_samplingframe %>%
  filter(snum<0.6) %>%
  select(-snum)

query<- raw_data_samplingframe %>%
  filter(snum>=0.6,snum<0.8) %>%
  select(-snum)

test<- raw_data_samplingframe %>%
  filter(snum>=0.8) %>%
  select(-snum)
```

DATA EXPLORATION

You are expected to do quite a bit of exploration with the data. It's interesting and intricate, and there are tantalizing stories that can be told using the data. The professors involved with this course have already spent quite a few hours playing with the data, and you should expect to do the same! That said, you **may not use the examples presented here** in your submission.

7. Beware that most of the rows correspond to sparsely populated counties, as you can see from a simple histogram:

```
training %>%
  ggplot(aes(TOT_POP)) +
  geom_histogram()
```

This can mess up your analysis because "most of the rows" does not equate to "most of the population"!

You might want to add a

```
+ scale_x_log10()
```

to the histogram above to spread the data a little more evenly, for instance.

Be aware that there are some strong correlations in the data that can swamp your analysis if you're not careful!

8. TOT_POP is highly correlated with many of the counts (for instance, with the number people of different ethnicities, and also with the vote counts). You are permitted (and encouraged) to add new variables temporarily to your data that normalize by TOT_POP.

For instance, you might be interested in the number of votes cast based on the "Rural Urban Continuum Code", which tries to classify how urban (values like 1 and 2) or rural (values like 8 or 9) a given county is. Caution: this code is not quantitative even though R thinks it is... (Go back to Step 5 if you missed it!) You might look at

```
training %>%
  ggplot(aes(group=Rural_urban_continuum_code_2013,total_votes)) +
  geom_boxplot()
```

But this is a bit misleading because the TOT_POP is correlated with both the code and the votes. So instead try:

```
training %>%
  mutate(voting_rate = total_votes/TOT_POP) %>%
  ggplot(aes(group=Rural_urban_continuum_code_2013,voting_rate)) +
  geom_boxplot()
```

Well... that's much more interesting!

9. You are also welcome and encouraged to explore other relationships. You can look at lots of crazy stats, for instance:

```
training %>%
  ggplot(aes(Percent.of.adults.completing.some.college.or.associate.s.degree..2015.19,
          X2021.04.01, # total COVID deaths up to this date
          color=Rural_urban_continuum_code_2013)) +
  geom_point()
```

Maybe this plot doesn't show much of a relationship, but maybe there's a different way to slice the data. Find out!

10. In your assignment, you'll have to make some models that use quantitative variables as explanatory variables. Since we've done Step 5 as a preliminary cleaning step to fix all the types, we can just grab all the quantitative variables (ie. the ones where is_numeric() returns TRUE) and make a new table from them. This is useful if you want a PCA for a quick look at the data

```
quantitatives <- training %>%
  select(where(is.numeric))
```

11. It's easy to get a quick PCA to see if there are any clusters worthy of consideration. Look back at previous assignments if you forget how this works!

```
training_pca <- quantitatives %>% prcomp()
training_pca$x %>%
  as_tibble() %>%
  mutate(PostalCode=training$PostalCode) %>% # Adding Postal Code back for plotting purposes
  ggplot(aes(PC1,PC2,color=PostalCode)) +
  geom_point()
```

You might also find it useful to focus your attention on a single state or some other feature. This is easy, just change how you compute the training_pca. But you can't filter `quantitatives` by postal code because we removed that… no problem:

```
training_pca <- training %>%
  filter(PostalCode=='MI') %>%
  select(where(is.numeric)) %>%
  prcomp()
```

If you do run this example, you'll probably see that there's an outlier. Figure out which county it is! (Hint: you'll have to do a left_join() or bind_cols() **after** you do the PCA -- since PCA doesn't like non-quantitative variables -- and then you can use geom_text() to add a label…)

12. Finally, let's look at county population versus ethnic makeup

```
training %>% filter(PostalCode == 'MI'|PostalCode == 'MS') %>%
  ggplot(aes(TOT_POP,white,color=Rural_urban_continuum_code_2013)) +
  geom_point() +
  facet_wrap(~PostalCode,scale='free_x')
```

There are different trends involved; they might be worthy of exploration!

TRAINING: PREDICTING QUANTITATIVE VARIABLES

For this assignment, you'll need to make a total of four models. Two of them must predict quantitative variables and two must predict categorical variables. You can add new composite variables that you'd like to predict, provided the explanatory variables you use to predict them aren't the ones in your formula. Otherwise, that would just be silly. Ultimately, I am not expecting that you will get really amazing predictions, but you should try for some reasonably good predictions. If you can't, then please try to explain what you have found.

13. As an example, let's try to predict voting rate from the county's racial breakdown. Voting rate is the number of votes divided by the county population. There is also a variable `white` that is the fraction of the population that identifies as white. Start with a plot that should show what we're after:

```
training %>%
  mutate(voting_rate=total_votes/TOT_POP) %>%
```

```
 ggplot(aes(white,voting_rate)) +
 geom_point() +
 facet_wrap(~Rural_urban_continuum_code_2013)
```

14. We can cook up a model using good-old linear regression

```
lfit <- lm(total_votes/TOT_POP ~ white, data=training)
```

and then we can fill in the residuals and see how well this model did. Let's plot against Postal code. This is effectively choosing Postal code as a hyperparameter in our model

```
training %>%
 add_residuals(lfit) %>% # Hint: this knows that we're testing prediction of
                         # total_votes/TOT_POP, not some "actual" column in our data
 mutate(PostalCode=reorder(PostalCode,desc(resid))) %>%
 ggplot(aes(resid,PostalCode)) +
 geom_boxplot()
```

You can also try plotting against others as well.

QUERY & TEST: PREDICTING QUANTITATIVE VARIABLES

15. You can select, say, the most appropriate rural urban continuum code for your model

```
query %>%
 add_residuals(lfit) %>%
 ggplot(aes(resid,Rural_urban_continuum_code_2013)) +
 geom_boxplot()
```

You'll get strange results if you haven't fixed the type of the `Rural_urban_continuum_code_2013` in Step 5. You can hack around this by wrapping it in as.factor() if you forgot it earlier.

16. Don't forget to test your final model! (See HW4A if you forgot how.)

TRAINING: PREDICTING CATEGORICAL VARIABLES

There are not many categorical variables in the dataset, so your choice will probably be pretty limited. That's OK, because you have many options for combinations of explanatory variables. The support vector machines (SVMs) we explored in HW4B can take any number of quantitative variables as input, even if you can't scatter plot them all! Don't be afraid to experiment!

17. After a bunch of poking, I found that I could predict the `PoliticsGroup` variable from a handful of variables. Of course, you can't use my example, but here's how it goes...

```
pg_svm_linear <- training %>%
 select(TOT_POP,  # These are the four variables I'm using as explanatory...
       Med_HH_Income_Percent_of_State_Total_2019,
       Percent.of.adults.with.less.than.a.high.school.diploma..2015.19,
       PCTPOVALL_2019) %>%
 svm(y=as.factor(training$PoliticsGroup), # Note: we need to tell R that this is a categorical!
    kernel='linear')
```

18. When I'm exploring models, I like to have everything in one big plot. First of all, run the SVM on the data:

```
pg_pred <- training %>% mutate(predicted_PG=predict(pg_svm_linear, training %>%
                    select(TOT_POP,
                        Med_HH_Income_Percent_of_State_Total_2019,
                        Percent.of.adults.with.less.than.a.high.school.diploma..2015.19,
                        PCTPOVALL_2019)))
```

Caution: when you go to query and test this model, don't forget that `training` shows up in **two** places, and both need to be changed…

19. Now, let's score the model's performance

```
pg_scored <- pg_pred %>%  mutate(status=(predicted_PG==PoliticsGroup)) %>%
  count(PostalCode,status)
```

20. Plot your models' performance. The mutate() is optional, but sorts the postal codes based on their size, which is visually helpful

```
pg_scored %>% mutate(PostalCode=reorder(PostalCode,desc(n))) %>%
  ggplot(aes(n,PostalCode,fill=status)) +
  geom_col()
```

QUERY & TEST: PREDICTING CATEGORICAL VARIABLES

21. Well, just repeat steps 18-20 with the query data! You can try other categorical variables instead of postal code, but don't tweak the model!

22. Once you find the variable and its value that you like the most, move on to the test! Report your performance as a percentage of correct predictions!

Assignment:
For submission, include your .Rmd file as well as your .pdf file. The main point of this project is to produce a carefully written report concerning part of the data. Therefore, make sure you explore the training data thoroughly!

- You must produce at least one map
- You must run a statistical analysis of some sort (scatter plot, T-test, chi squared, ANOVA, or linear regression) as part of your exploration of the training set.
- You must train two "problems" using predictive models from the data, with two models trained in each, for a total of four models.

Problem 1. As in HW4A, use one or more quantitative variables to predict another quantitative variable. If you don't find a quantitative variable to your liking, you may create one using mutate(). Don't use the variables in your formula as explanatory variables!

Problem 2. As in HW4B, use one or more quantitative variables to predict a categorical variable. If you don't find a categorical variable to your liking, you may create one using mutate() and some appropriate thresholds. Don't use the variables in your formula as explanatory variables!

For each of the above problems, write an explanation of your experimental hypothesis, based on the training data. That is, please explain what you expect the model to find. Also explain which models you think might do best at solving the problems and why.

- Use the query set to narrow down to one model from each of the two problems, as we've done in previous assignments. You may also use the query set to narrow down via another categorical variable, like was done in Steps 15 and 21.
- Run your top two models (one for each problem) on the test data. Explain your results.
- Finally, interpret your results in terms of the test data. For this project, what that means is that you should do a little reading about the states and counties you've found and explain your data findings in the context of those readings. You may find that your results don't agree; that's fine. But be thorough, and make sure to cite your sources.