Capstone Project 1 - Data Wrangling

Highest performing schools depending on average ACT scores

In this project, I worked on two different datasets:

- ACT scores for high school students in California from 2014 to 2018
- Non-educational data about high schools in California such as: census, geographic location, parent ratings etc.

The ACT scores data came from the California Department of Education website. These datasets included average ACT scores (math, literature and writing) of students for every high school in California. I downloaded the census data from greatschools.org website. I accomplished this in two steps. First I requested the school ID numbers from greatschools.org using API. Then, I used these ID numbers in a separate API request to acquire the census data. Along with the census data, I also retrieved longitude, latitude, and parent ratings of schools. I have two jupyter notebooks showing the data wrangling steps for these two different datasets.

Data Wrangling Part 1:

For the ACT scores, I had five different datasets belonging to different years ranging from 2014 to 2018. Before taking the cleaning steps, I added year columns to some of the datasets with no year column information.

1. **Cleaning steps:**
    a. To have consistent column names, I cleaned the extra white spaces in the column names and capitalized them to concatenate my datasets from all years.
    b. I constructed new columns by trimming the redundant ones.
    c. I formatted the year column and set it as index.
    d. I corrected the incompatible value types in columns.
    e. To eliminate empty spaces in the text of some columns, I used col.astype(str).str.strip() method.

2. **Missing Data:** I handled the missing data in two steps.
    a. In the data, the missing values were shown with ‘ * ’ sign. I changed them using na_values=['*'].
    b. Then, I dropped the rows with too many Nan values using .dropna(thresh=7) method.

3. **Outliers:** Applying the .describe() method on the data, I inspected the outcomes. The max and min values of the columns were compatible with the other numbers. I concluded that the data was free of outliers.

Data Wrangling Part 2:

I had three csv files in this section: The ACT file, gsid file (school ID information used to retrieve other data from greatschools website) and the census data that I acquired using api request. The main process in this notebook was merging the datasets with as much common information as possible. The act data had school and county names, the gsid data had school and city names, and the census data had only the school names column that I could use in the merging process. Thus, I added county names to the gsid data with gsid['City'].map(city_cnty) method using the excel file which includes city names and their corresponding counties. Then, using the school ID numbers, I merged the census and gsid datasets and created census_id data with the command shown below.
census_id = pd.merge(census, gsid, left_on='Id', right_on='gsId', how ='inner', suffixes = ('','_gsid')).
Some school names were spelled differently in the ACT file and census_id data. To merge these files, I used the fuzzywuzzy library string matching. Another issue was that there were multiple schools with the same name in California. To fix this issue, I matched the schools names only if they were in the same county. I merged the ACT data with census_id data based on two columns (school names and county names) using the final_df = pd.merge(act, census_id, left_on=['Sname2', 'Cname'], right_on=['Sname', 'County'], how ='inner',suffixes = ('','3')) command.

1. **Cleaning steps:**
    a. I corrected the incompatible value types in columns (strings, integers, etc.).
    b. Then I removed the school names that caused duplicates in the final data.
    c. There were a few school names which were mismatched. Manually, I created a dictionary to correct the mismatched school names. Then, I dropped the redundant columns.

2. **Missing Data:** In the final data there wasn't much missing data because of the major cleaning steps that I've performed in the first data wrangling part. In the final data, I replaced the missing values with Nan and left them as is.

3. **Outliers:** I applied the .describe() method on the final data to be able to inspect the outliers. In the 'Enrollment' column, the maximum value seemed very high. So, setting up a threshold value for the number of students, I obtained the school names using final_df[final_df['Enrollment']>4500] code. Those schools were the largest high schools in California according to the information on the website, and I kept them as is. The other columns didn't have any outliers.