**IFT 511 Analyzing Big Data**

**Project Step 3: Data Cleaning**

**Instructor: Asmaa Elbadrawy**

**Team 56**

Advitha Bawgi

Lakshman Sai Yaganti

Sai Sushank Manthri Pramod

Sathwika Kannareddy

A thorough data cleaning procedure for a dataset loaded from a CSV file using the pandas library is described in the Python script that is being shown. The script evaluates the data in the first steps by giving an overview of each row’s and column's missing values, which provides information about how much data is missing. Then, it removes columns with missing value count greater than 20 first, then all the rows containing any missing values. This is how it methodically handles missing values. Because of the repetitive nature of the cleaning process, additional modifications can be made in response to the unique features of the dataset.

Subsequently, the script concentrates on eliminating duplicate records by detecting and eliminating rows that have the same school name and city/zip code. By taking this step, redundancy in the dataset is ensured. After cleaning, the script moves on to a data transformation step. Here, special characters are removed from a numeric column using the replace method with a regular expression. This illustrates how flexible the script is to different requirements for data manipulation.

The script ends by printing a summary of the data cleaning process that highlights the important actions done to improve the quality of the data. It places special emphasis on transforming particular data items and eliminating duplicate records and missing values. A clear and recorded trail of the data cleaning procedure is provided by saving the transformed and cleaned data into a new CSV file for analysis at a later time. This meticulous process guarantees data integrity and reliability by ensuring that the final dataset is ready for further analysis.

The script's primary transformation operation entails iterating through each dataset column and using SchoolListData[col] to determine whether the column includes text.dtype is equivalent to "object." A transformation is used to change numeric values with special characters in columns that contain text. The pandas library's p.to\_numeric function is used to perform this transformation. To exclude non-numeric characters ([^\d.]), a regular expression is utilized. To deal with non-convertible values, the errors='coerce' argument is used to replace them with NaN. This particular transformation is intended to handle situations in which special characters, like percentages or currency symbols, may appear in numeric columns.

Following the transformation process, the script prints the dataset's final dimensions, illuminating how the cleaning and transformation processes affected the data's general structure. This data is essential for evaluating how well the procedures were applied and making sure the dataset is still appropriate for further analysis.

The script ends by using the SchoolListData.to\_csv function to save the cleaned and transformed dataset to a new CSV file called "cleaned\_transformed\_dataset.csv." This stage makes sure that the data has been processed and is readily loadable for additional analysis or modeling.

Regarding data attribute kinds and transformation techniques, the script focuses mostly on text and numeric attributes. The elimination of non-numeric characters specifically addresses numerical attributes with special characters. Nevertheless, other attribute types, including nominal or datetime attributes, are not specifically handled by the script. Additional stages could be required to fully address different attribute types during the data cleaning and transformation process, depending on the features of the dataset.

Code:

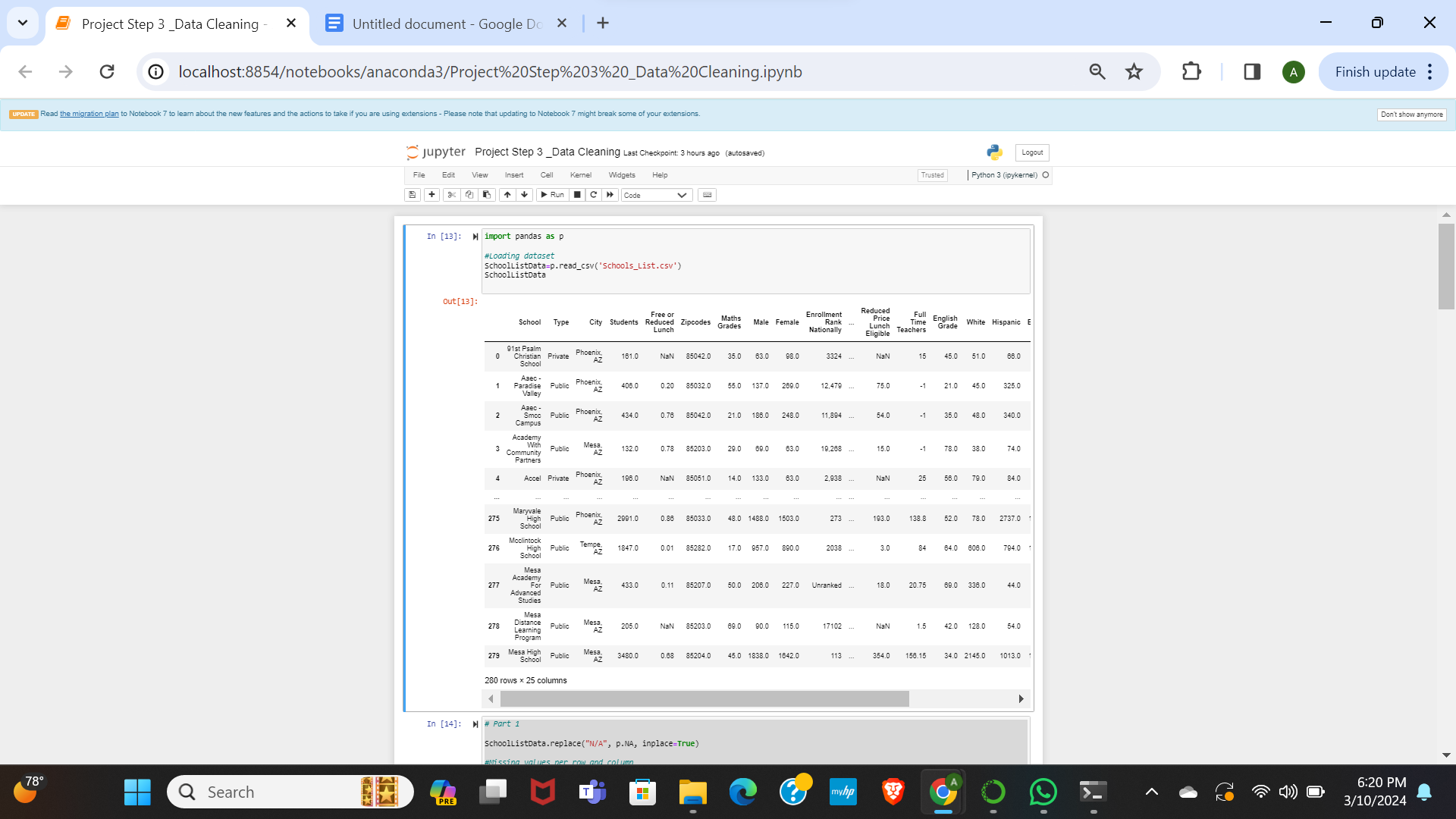
import pandas as p

#Loading dataset

SchoolListData=p.read\_csv('Schools\_List.csv')

SchoolListData

Output:



Code:

# Part 1

SchoolListData.replace("N/A", p.NA, inplace=True)

#Missing values per row and column

missing\_val\_per\_row = SchoolListData.isnull().sum(axis=1)

missing\_val\_per\_column = SchoolListData.isnull().sum()

# Summary of missing values per row and column

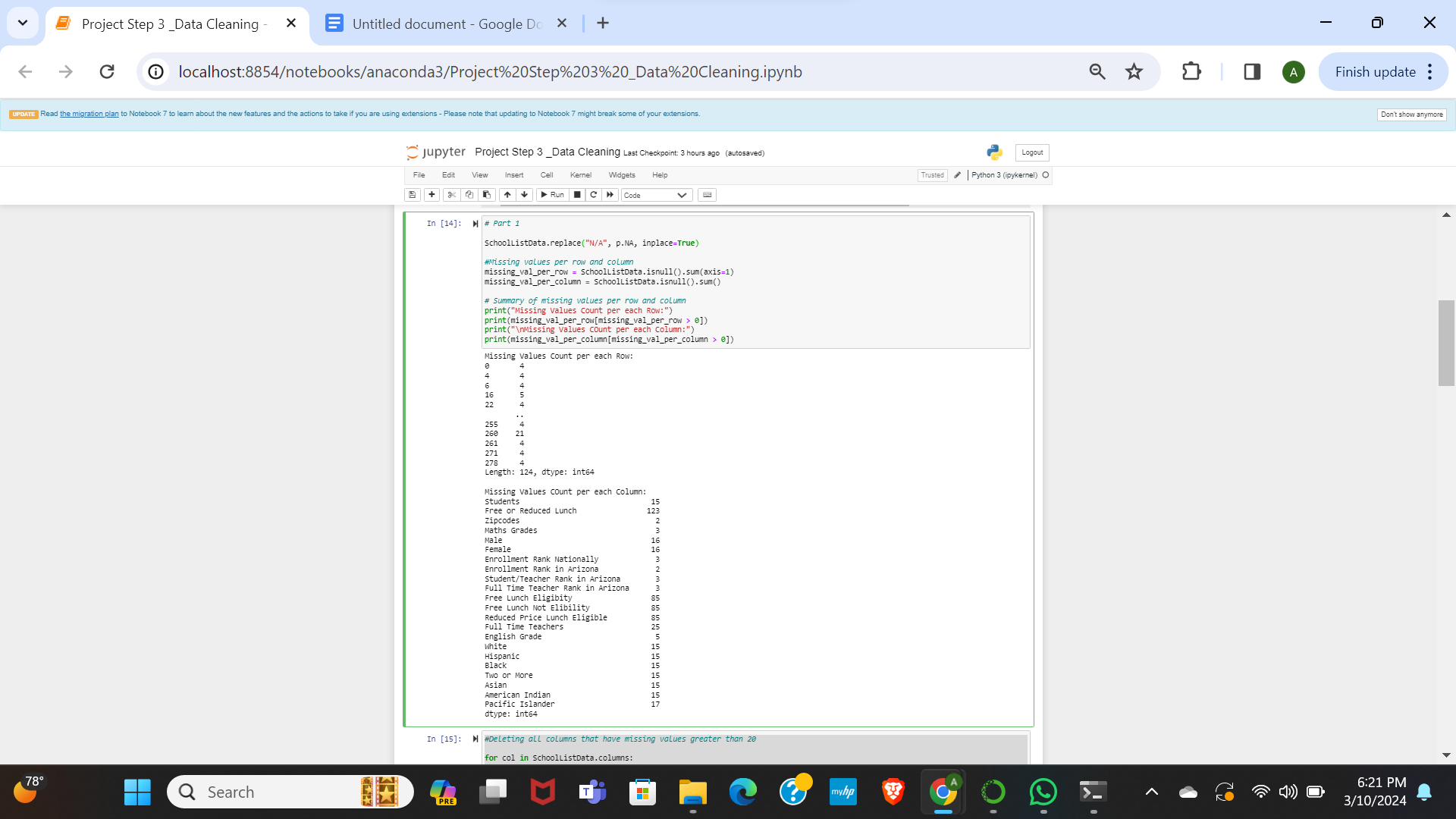
print("Missing Values Count per each Row:")

print(missing\_val\_per\_row[missing\_val\_per\_row > 0])

print("\nMissing Values COunt per each Column:")

print(missing\_val\_per\_column[missing\_val\_per\_column > 0])

Output:



Code:

#Deleting all columns that have missing values greater than 20

for col in SchoolListData.columns:

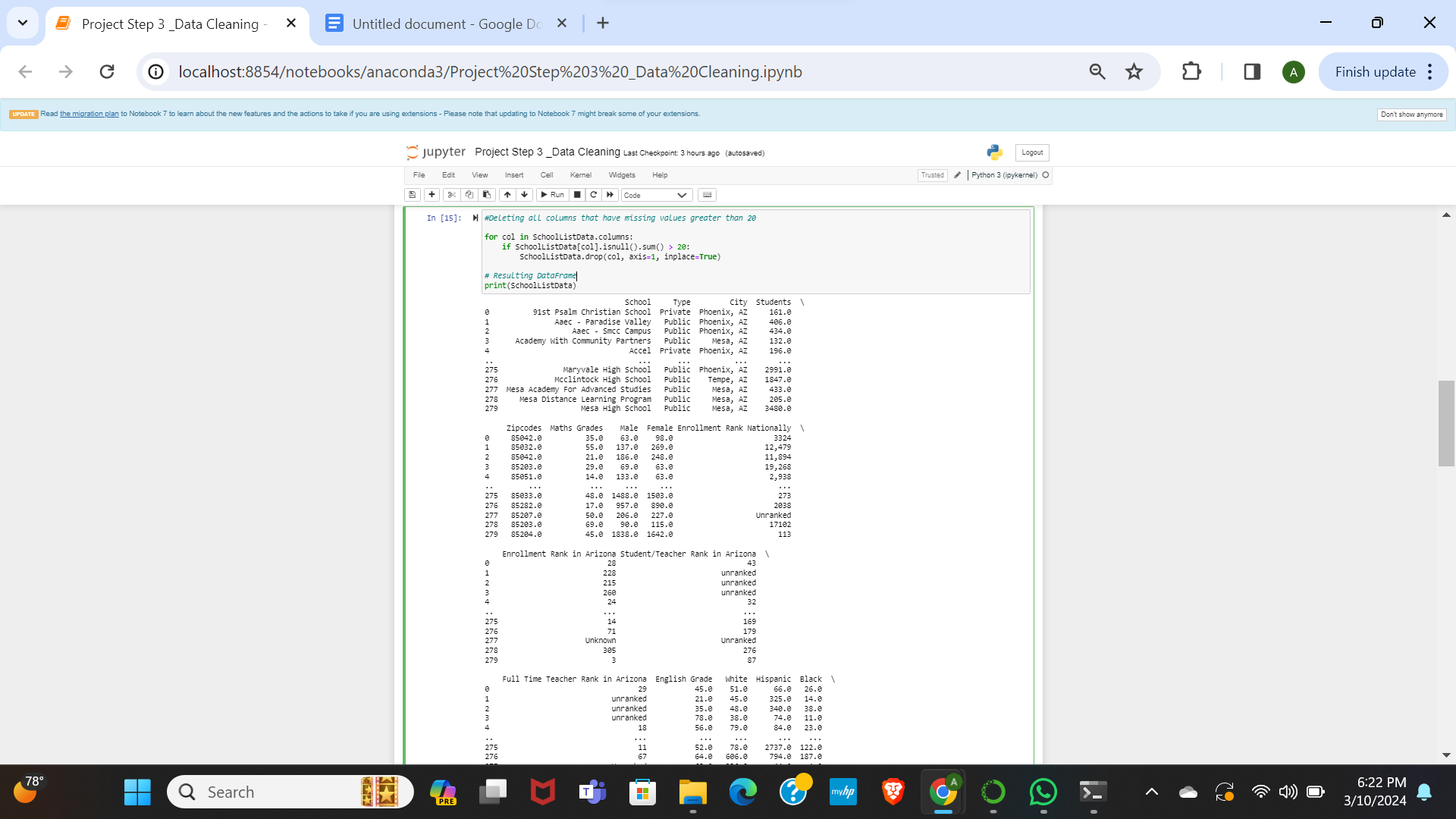
if SchoolListData[col].isnull().sum() > 20:

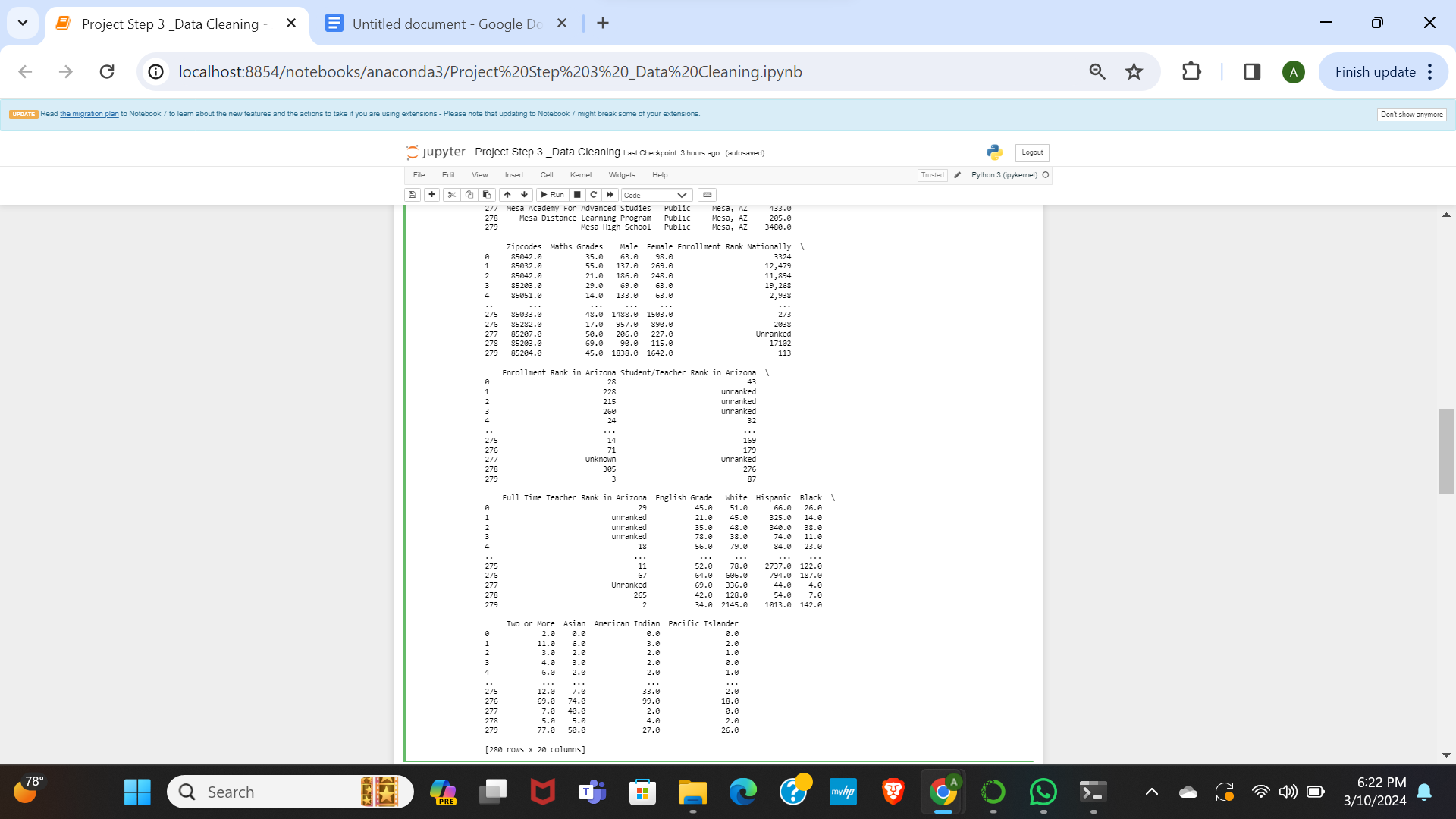
SchoolListData.drop(col, axis=1, inplace=True)

# Resulting DataFrame

print(SchoolListData)

Output:





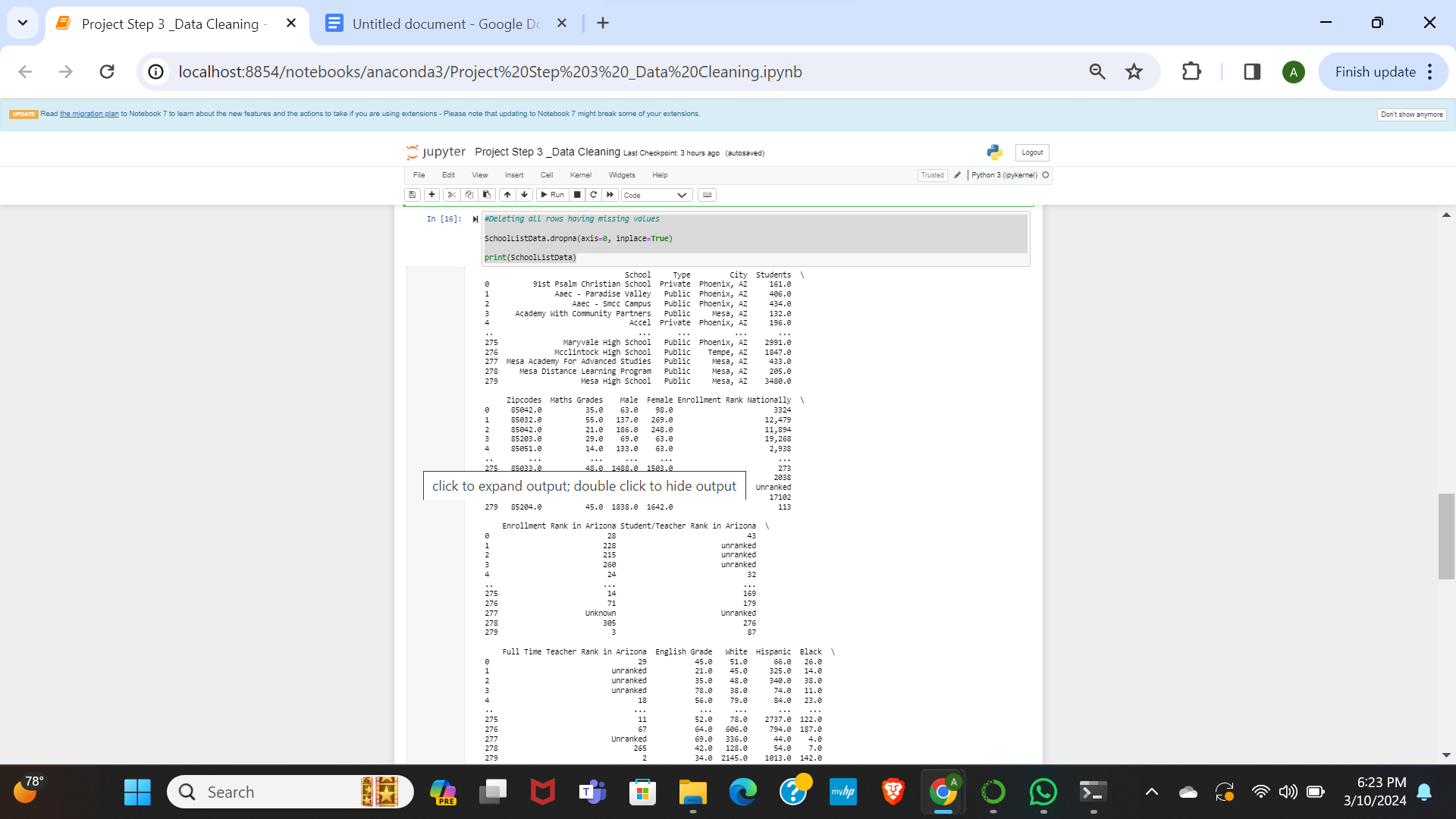
Code:

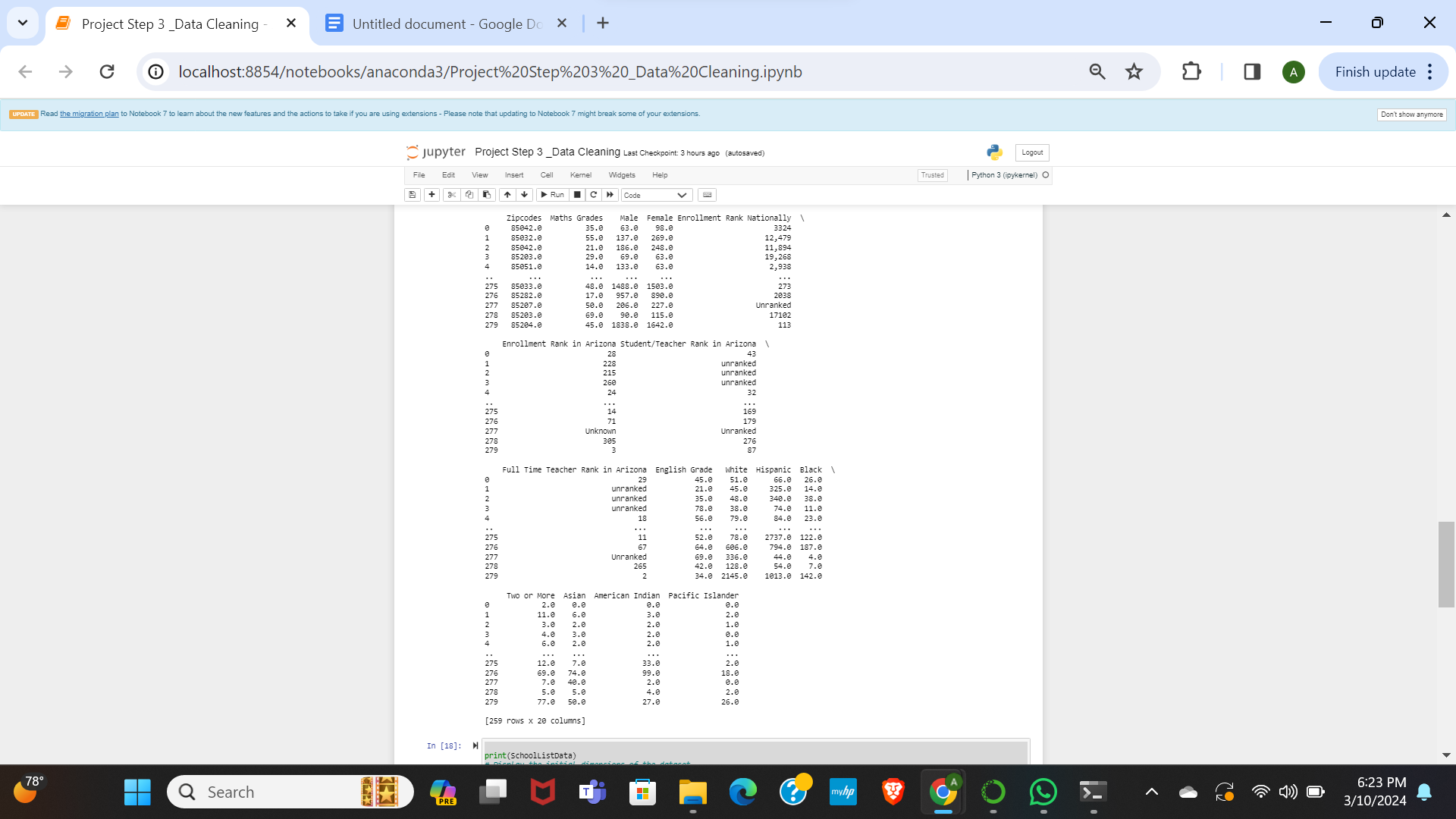
#Deleting all rows having missing values

SchoolListData.dropna(axis=0, inplace=True)

print(SchoolListData)

Output:





Code:

# Display the initial dimensions of the dataset

print("Initial Dataset Dimensions:", SchoolListData.shape)

# Iterate through each column and transform numeric values with special characters

for col in SchoolListData.columns:

if SchoolListData[col].dtype == 'object': # Check if the column contains text

SchoolListData[col] = p.to\_numeric(SchoolListData[col].str.replace('[^\d.]', '', regex=True), errors='coerce')

# Display the final dimensions of the dataset after transformation

print("Final Dataset Dimensions:", SchoolListData.shape)

# Save the cleaned and transformed dataset to a new CSV file

SchoolListData.to\_csv('cleaned\_transformed\_dataset.csv', index=False)

Output:

