Project Step 3 - Data Cleaning

1. Link of collected data CSV - https://docs.google.com/spreadsheets/d/1m1RNQFYm82FV0qjTu22cmgQthJj1HhwwH0ampHcQ-9o/edit#gid=0

Python script:

1. Clean missing values -

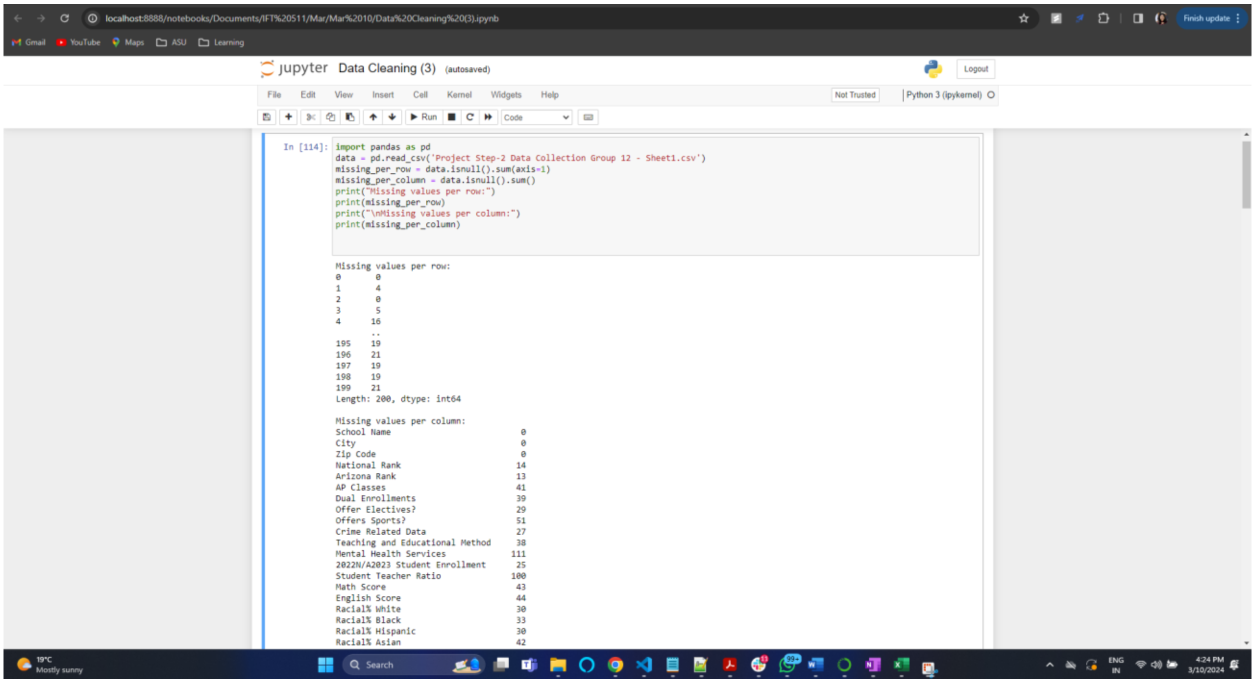
<u>Code:</u>

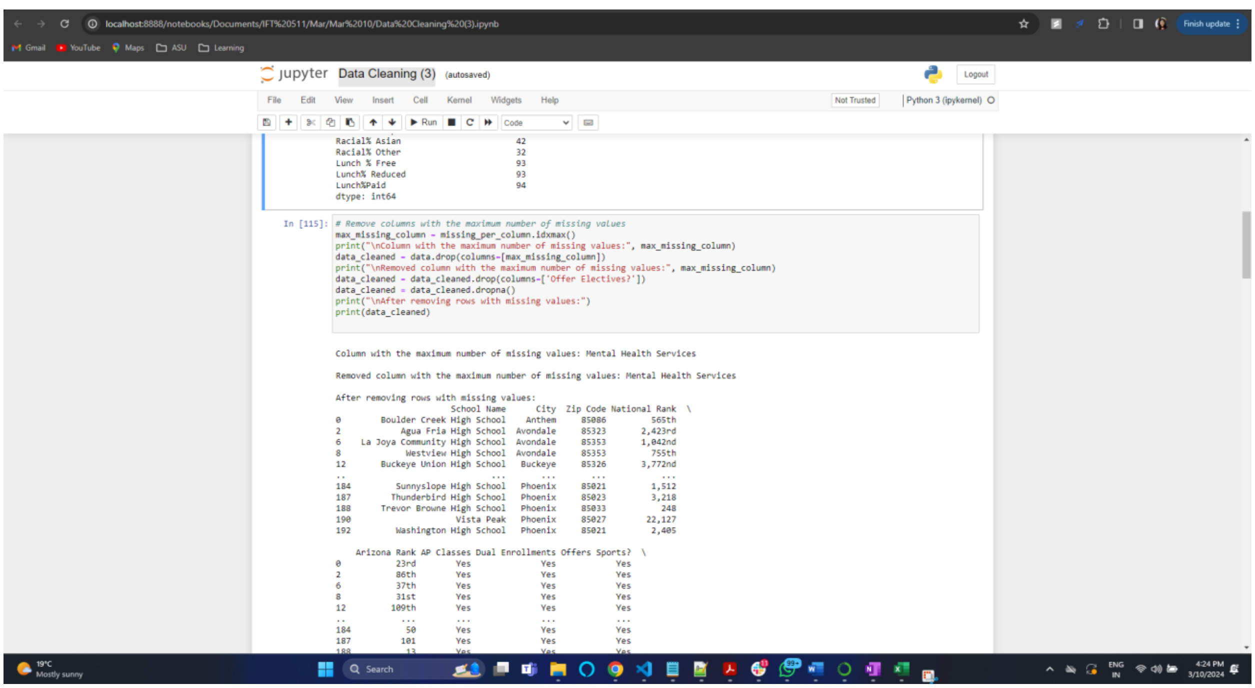
```
import pandas as pd

data = pd.read_csv('Project Step-2 Data Collection Group 12 - Sheet1.csv')

missing_per_row = data.isnull().sum(axis=1)

missing_per_column = data.isnull().sum()

print("Missing values per row:")

print(missing_per_row)

print("\nMissing values per column:")

print(missing_per_column)


# Remove columns with the maximum number of missing values

max_missing_column = missing_per_column.idxmax()

print("\nColumn with the maximum number of missing values:", max_missing_column)

data_cleaned = data.drop(columns=[max_missing_column])

print("\nRemoved column with the maximum number of missing values:",
max_missing_column)

data_cleaned = data_cleaned.drop(columns=['Offer Electives?'])

data_cleaned = data_cleaned.dropna()
```

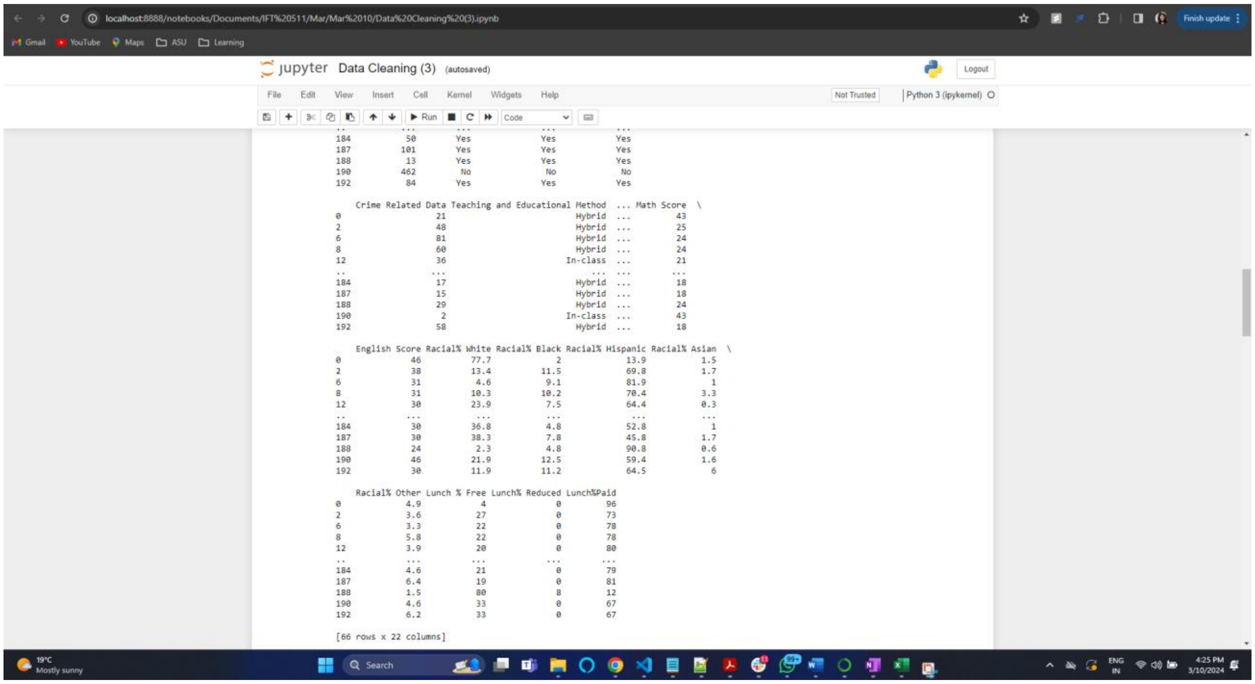
print("\nAfter removing rows with missing values:")

print(data_cleaned)

<u>Screenshots:</u>







Data Cleaning Procedure:

Loading the Dataset:

The first step is to use the pandas library to load the dataset from the CSV file. For this, the pd.read_csv() function is used.

Computing Missing Values:

The isnull() function is used to determine the amount of missing values per row and per column. Using appropriate axis, Sum() is used for providing insights into areas where data is missing within the dataset. missing_per_row determines the total of missing values for each row across columns. The sum of the missing values for each column is determined by missing_per_column.

Finding the Column with the Most Missing Values at Maximum:

The idxmax() function is used in conjunction with the missing_per_column variable to determine which column has the greatest amount of missing values. There are a considerable amount of missing values in this column.

Eliminating Columns with the Highest Amount of Missing Data:

Using the drop() function along the columns axis, the dataset is cleared of the selected column with the greatest number of missing values. The purpose of this action is to remove columns that contain missing or faulty data.

Manually Remove 'Offer Electives?' Column:

Because the 'Offer Electives?' column contains a large percentage of missing values, it is manually eliminated from the dataset. This choice is based on the knowledge that, because of the significant amount of missing data, the column might not offer useful information for analysis.

Eliminating Rows with Missing Values:

Following the removal of missing-value columns, the dropna() method is used to eliminate any remaining rows with missing values from the dataset. This assures that the dataset only includes complete entries, which is critical for many analytical applications.

Displaying the cleaned dataset:

Finally, the code prints the cleaned dataset so that you can inspect the generated data after the cleaning process. This stage provides a visual assessment of the dataset's completeness and quality after cleaning.

2. Clean duplicates:

<u>Code:</u>

```
# Step 2: Clean the data from duplicate values

# duplicate records based on school name and city/zip code

duplicate_records = data_cleaned[data_cleaned.duplicated(subset=['School Name', 'City',
'Zip Code'], keep=False)]

combined_records = duplicate_records.groupby(['School Name']).agg(lambda x:
x.mean() if pd.api.types.is_numeric_dtype(x) else ', '.join(x)).reset_index()

data_cleaned = data_cleaned.drop_duplicates(subset=['School Name', 'City', 'Zip Code'],
keep=False)

data_cleaned = pd.concat([data_cleaned, combined_records], ignore_index=True)
```

print("\nDuplicate records:")

print(duplicate_records)

print("\nCombined duplicate records:")

print(combined_records)

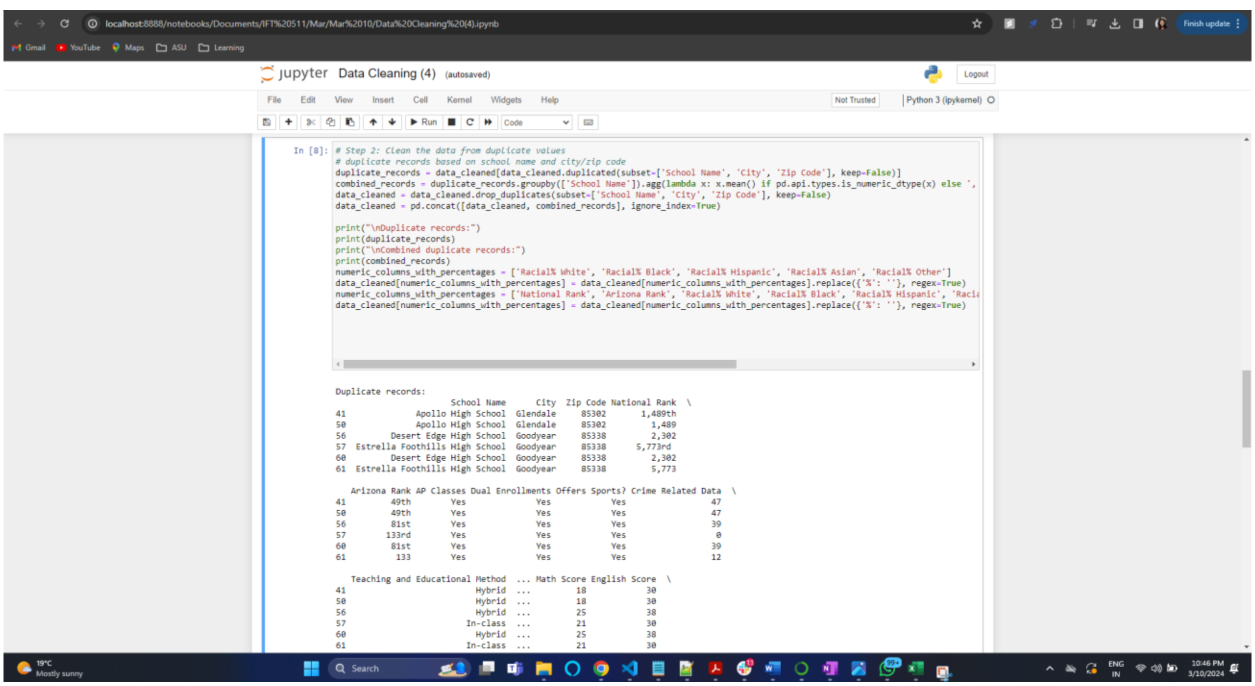
numeric_columns_with_percentages = ['Racial% White', 'Racial% Black', 'Racial% Hispanic', 'Racial% Asian', 'Racial% Other']

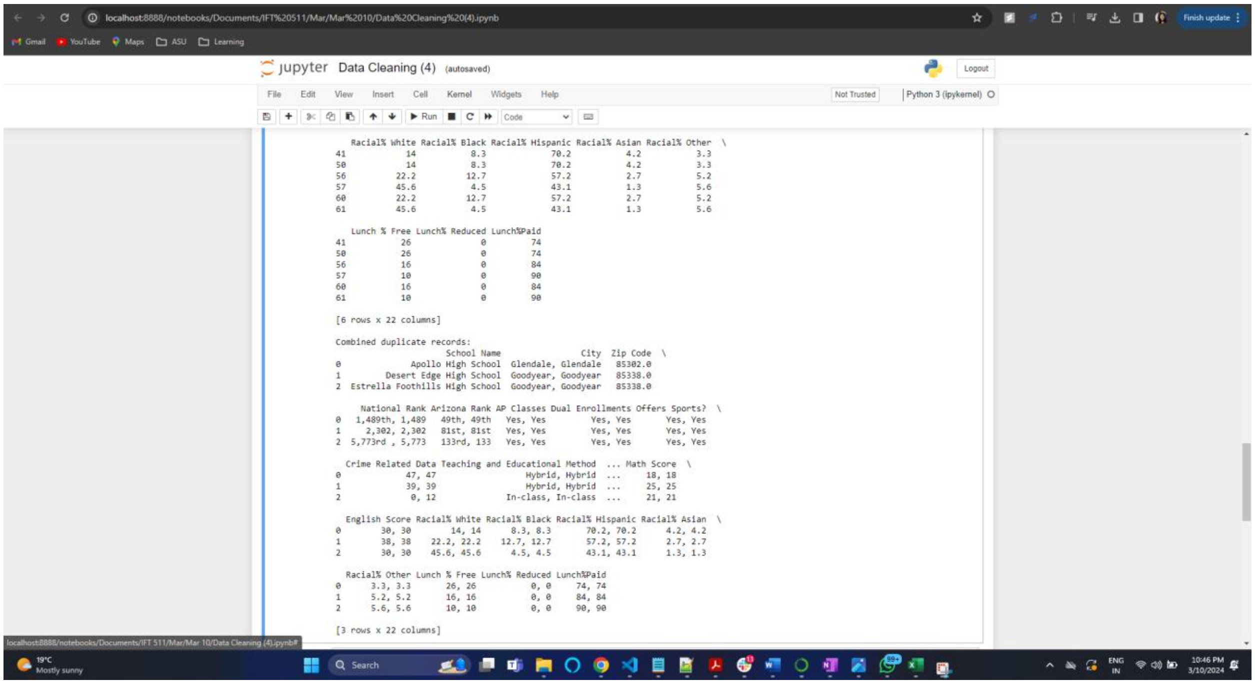
data_cleaned[numeric_columns_with_percentages] = data_cleaned[numeric_columns_with_percentages].replace({'%': ''}, regex=True)

numeric_columns_with_percentages = ['National Rank', 'Arizona Rank', 'Racial% White', 'Racial% Black', 'Racial% Hispanic', 'Racial% Asian', 'Racial% Other', 'Lunch % Free', 'Lunch% Reduced', 'Lunch%Paid']

data_cleaned[numeric_columns_with_percentages] = data_cleaned[numeric_columns_with_percentages].replace({'%': ''}, regex=True)

<u>Screenshots:</u>





<u>Explanation:</u>

Identifying Duplicate Records:

First, the code uses the DataFrame data_cleaned with the subset of columns designated as ['School Name, City, Zip Code'] to find duplicate records using the duplicated() function. If any row in these columns has the same values as another, this function marks it as a duplicate.

Merging Duplicate Records:

The code merges duplicate records for each school into a single record. Using groupby(), it aggregates the values in each column and groups the duplicate data by 'School Name'.

The mean() function is used to determine the mean value for numerical columns. It uses a lambda function with join() to concatenate the values for non-numeric columns into a string separated by commas.

Eliminating Duplicate Records:

The drop_duplicates() function is used to delete the original duplicate records from the dataset after combining the duplicate records. The removal of duplicate records is guaranteed by the argument keep=False.

Combining Records:

pd.concat() is used to concatenate the combined records with the original dataset, guaranteeing that the dataset contains both the unique records and the combined duplicate records. The generated DataFrame is guaranteed to have a new index by the parameter ignore_index=True.

Managing Percentages in Numerical Columns:

The replace() function with a regular expression is used in the code to remove the percentage symbol (%) from numerical columns containing percentages ('Racial% White', 'Racial% Black', 'Racial% Hispanic', 'Racial% Asian', 'Racial% Other', 'National Rank', 'Arizona Rank', 'Lunch % Free', 'Lunch% Reduced', 'Lunch%Paid').

Displaying Combined and Duplicate Records:

To give insight into the cleaning process, the script publishes the combined records (combined_records) and the duplicate records (duplicate_records).

3. Transform the data:

<u>Code:</u>

```
# Step 3: Remove rows based on attribute weights

# Defining weights for each attribute based on their importance for reviewing the best
school

attribute_weights = {

    'National Rank': 5,

    'Arizona Rank': 4,

    'AP Classes': 3,

    'Dual Enrollments': 3,

    'Offers Sports?': 2,

    'Math Score': 5,
```

```
    'English Score': 5,
    'Racial% White': 4,
    'Racial% Black': 4,
    'Racial% Hispanic': 4,
    'Racial% Asian': 4,
    'Racial% Other': 4,
    'Lunch % Free': 3,
    'Lunch% Reduced': 3,
    'Lunch%Paid': 3
}

# Calculate weighted score for each row based on the presence or absence of values in attributes
data_cleaned['Weighted Score'] = data_cleaned.apply(lambda row: sum(attribute_weights[attr] for attr in attribute_weights if pd.notnull(row[attr])), axis=1)
numeric_columns = ['Math Score', 'English Score', 'Racial% White', 'Racial% Black', 'Racial% Hispanic', 'Racial% Asian', 'Racial% Other', 'Lunch % Free', 'Lunch% Reduced', 'Lunch%Paid']
data_cleaned[numeric_columns] = data_cleaned[numeric_columns].replace({'\$': '', '%': '', ',': '', 'th': '','st': '','nd':'','rd':''}, regex=True)

# Remove rows with missing values in attributes with higher weights
threshold_weight = 10  # Adjust the threshold weight as needed
data_cleaned = data_cleaned[data_cleaned['Weighted Score'] >= threshold_weight]
# Drop the 'Weighted Score' column as it is no longer needed
```

data_cleaned = data_cleaned.drop(columns=['Weighted Score'])

data_cleaned = data_cleaned[~data_cleaned.apply(lambda row: (row == 'Permanently Closed').any(), axis=1)]

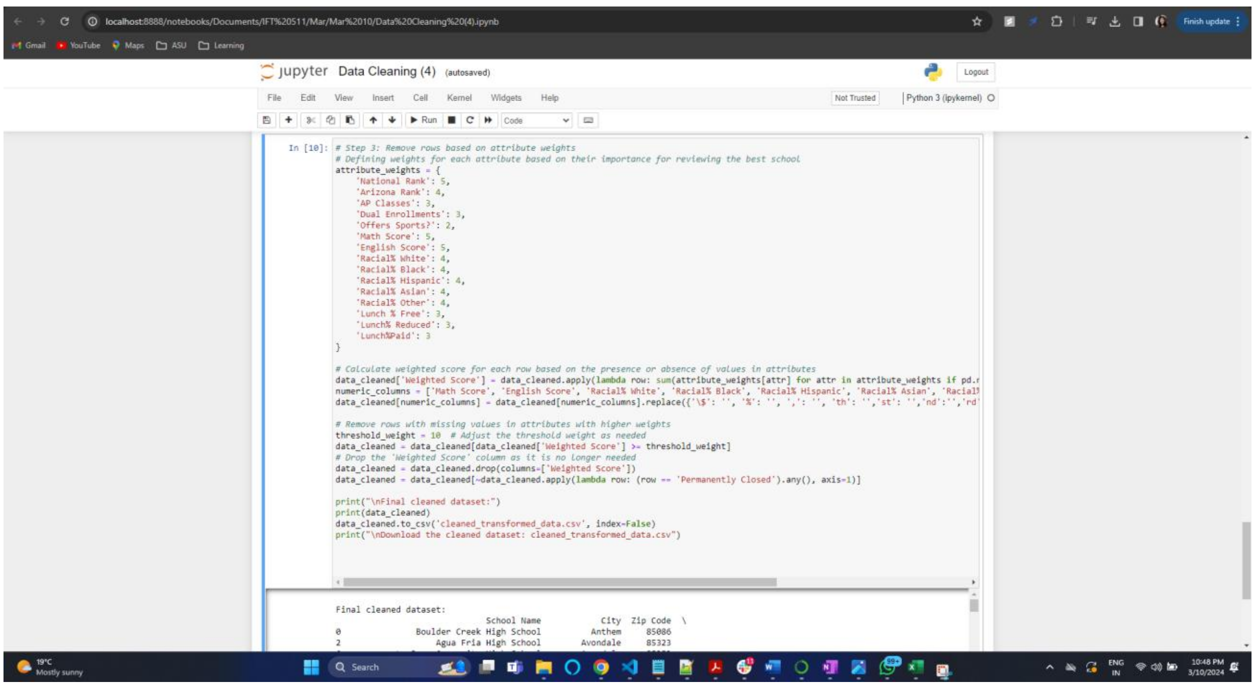
print("\nFinal cleaned dataset:")

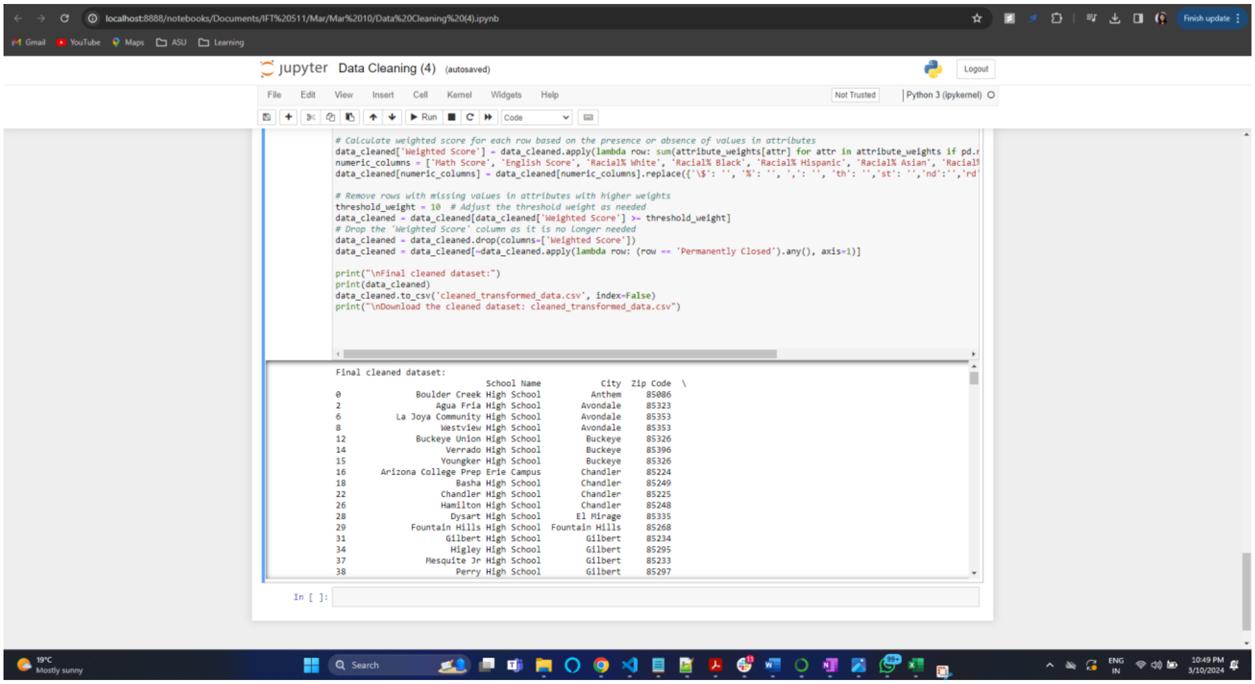
print(data_cleaned)

data_cleaned.to_csv('cleaned_transformed_data.csv', index=False)

print("\nDownload the cleaned dataset: cleaned_transformed_data.csv")

<u>Screenshots:</u>

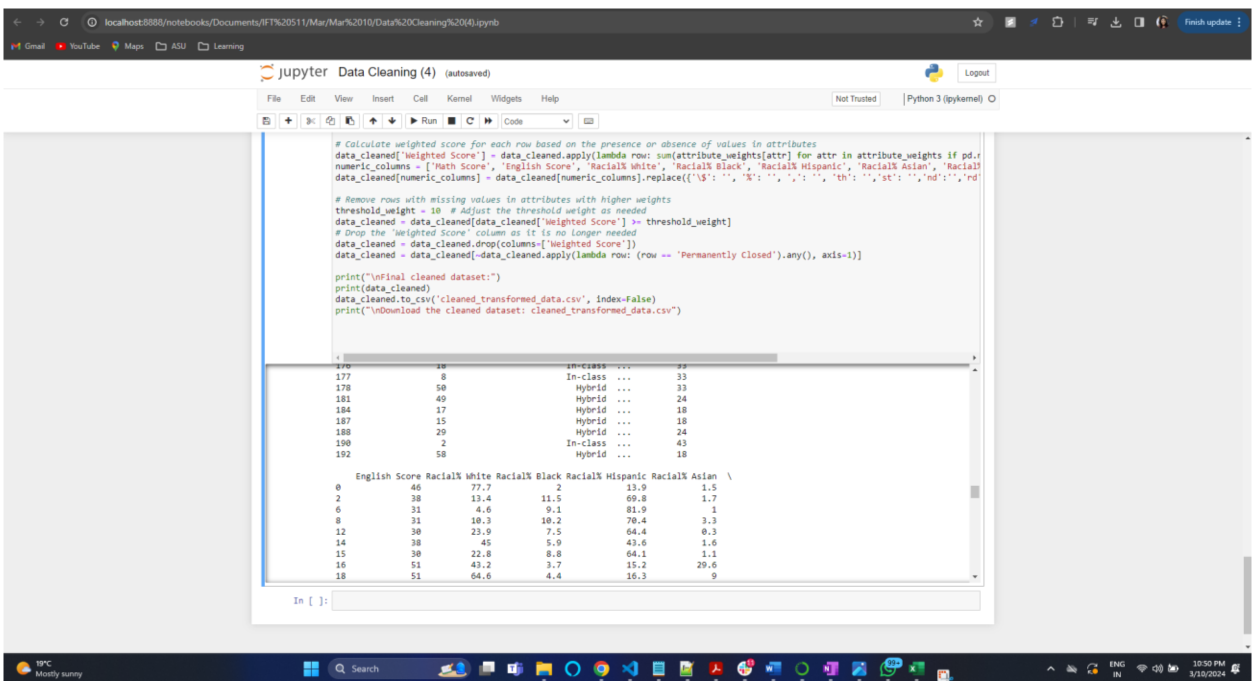




jupyter Data Cleaning (4) (autosaved)

```
# Calculate weighted score for each row based on the presence or absence of values in attributes
data_cleaned['Weighted Score'] = data_cleaned.apply(lambda row: sum(attribute_weights[attr] for attr in attribute_weights if pd.r
numeric_columns = ['Math Score', 'English Score', 'Racial% White', 'Racial% Black', 'Racial% Hispanic', 'Racial% Asian', 'Racial%
data_cleaned[numeric_columns] = data_cleaned[numeric_columns].replace({'\$': '', '%': '', ',': '', 'th': '','st': '','nd':'','rd'

# Remove rows with missing values in attributes with higher weights
threshold_weight = 10  # Adjust the threshold weight as needed
data_cleaned = data_cleaned[data_cleaned['Weighted Score'] >= threshold_weight]
# Drop the 'Weighted Score' column as it is no longer needed
data_cleaned = data_cleaned.drop(columns=['Weighted Score'])
data_cleaned = data_cleaned[~data_cleaned.apply(lambda row: (row == 'Permanently Closed').any(), axis=1)]

print("\nFinal cleaned dataset:")
print(data_cleaned)
data_cleaned.to_csv('cleaned_transformed_data.csv', index=False)
print("\nDownload the cleaned dataset: cleaned_transformed_data.csv")
```

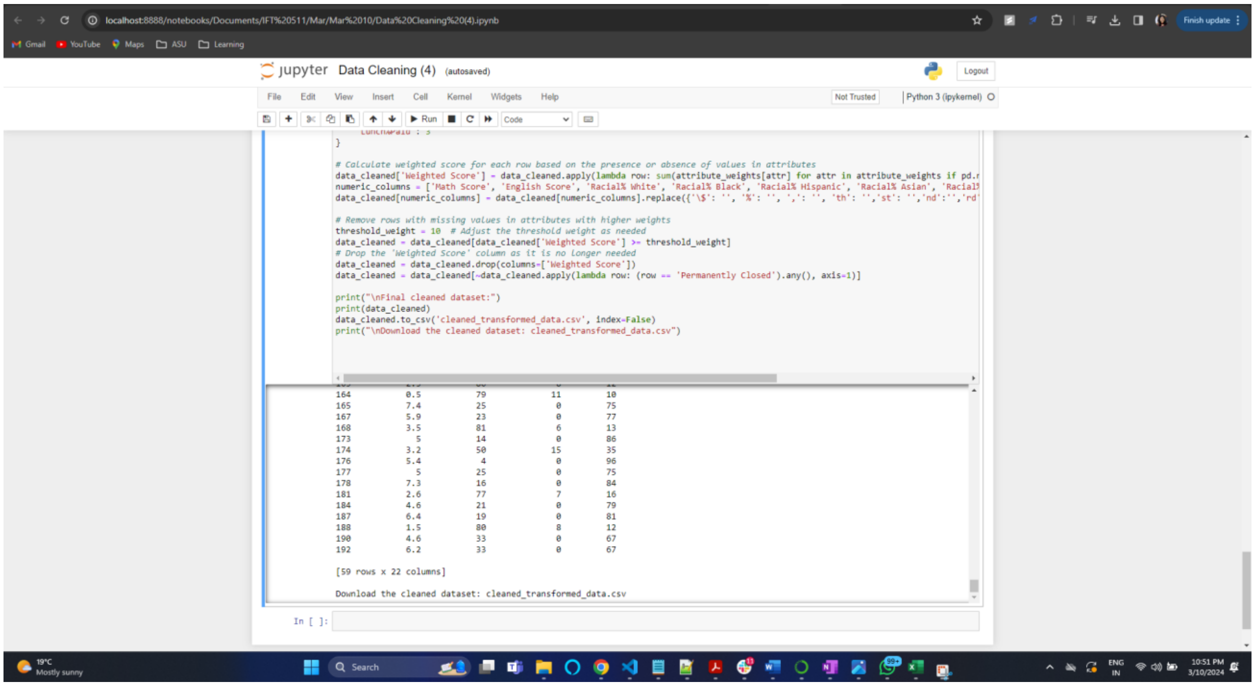
```
Final cleaned dataset:
                         School Name            City  Zip Code  \
0             Boulder Creek High School          Anthem     85086
2                Agua Fria High School        Avondale     85323
6         La Joya Community High School        Avondale     85353
8                Westview High School        Avondale     85353
12            Buckeye Union High School         Buckeye     85326
14                Verrado High School         Buckeye     85396
15               Youngker High School         Buckeye     85326
16      Arizona College Prep Erie Campus        Chandler     85224
18                  Basha High School        Chandler     85249
22               Chandler High School        Chandler     85225
26               Hamilton High School        Chandler     85248
28                 Dysart High School       El Mirage     85335
29         Fountain Hills High School  Fountain Hills     85268
31                Gilbert High School         Gilbert     85234
34                 Higley High School         Gilbert     85295
37            Mesquite Jr High School         Gilbert     85233
38                  Perry High School         Gilbert     85297
```



jupyter Data Cleaning (4) (autosaved)

```
# Calculate weighted score for each row based on the presence or absence of values in attributes
data_cleaned['Weighted Score'] = data_cleaned.apply(lambda row: sum(attribute_weights[attr] for attr in attribute_weights if pd.r
numeric_columns = ['Math Score', 'English Score', 'Racial% White', 'Racial% Black', 'Racial% Hispanic', 'Racial% Asian', 'Racial%
data_cleaned[numeric_columns] = data_cleaned[numeric_columns].replace({'\$': '', '%': '', ',': '', 'th': '','st': '','nd':'','rd'

# Remove rows with missing values in attributes with higher weights
threshold_weight = 10  # Adjust the threshold weight as needed
data_cleaned = data_cleaned[data_cleaned['Weighted Score'] >= threshold_weight]
# Drop the 'Weighted Score' column as it is no longer needed
data_cleaned = data_cleaned.drop(columns=['Weighted Score'])
data_cleaned = data_cleaned[~data_cleaned.apply(lambda row: (row == 'Permanently Closed').any(), axis=1)]

print("\nFinal cleaned dataset:")
print(data_cleaned)
data_cleaned.to_csv('cleaned_transformed_data.csv', index=False)
print("\nDownload the cleaned dataset: cleaned_transformed_data.csv")
```

```
177          8          In-class  ...       33
178         50            Hybrid  ...       33
181         49            Hybrid  ...       24
184         17            Hybrid  ...       18
187         15            Hybrid  ...       18
188         29            Hybrid  ...       24
190          2          In-class  ...       43
192         58            Hybrid  ...       18

    English Score Racial% White Racial% Black Racial% Hispanic Racial% Asian  \
0              46          77.7             2             13.9           1.5
2              38          13.4          11.5             69.8           1.7
6              31           4.6           9.1             81.9             1
8              31          10.3          10.2             70.4           3.3
12             30          23.9           7.5             64.4           0.3
14             38            45           5.9             43.6           1.6
15             30          22.8           8.8             64.1           1.1
16             51          43.2           3.7             15.2          29.6
18             51          64.6           4.4             16.3             9
```



Cleaned Dataset:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	School Name	City	Zip Code	National F	Arizona Ra	AP Classe	Dual Enro	Offers Spc	Crime Rel	Teaching	2022N/A2(	Student Te	Math Scor	English Sc	Racial% W	Racial% B	Racial% H	Racial% A	Racial% C	Lunch % F	Lunch% R	Lunch%Paid
2	Boulder Creek High School	Anthem	85086	565th	23rd	Yes	Yes	Yes	21	Hybrid	2,456	1:01	43	46	77.7	2	13.9	1.5	4.9	4	0	96
3	Agua Fria High School	Avondale	85323	2,423rd	86th	Yes	Yes	Yes	48	Hybrid	1,608	17:01	25	38	13.4	11.5	69.8	1.7	3.6	27	0	73
4	La Joya Community High School	Avondale	85353	1,042nd	37th	Yes	Yes	Yes	81	Hybrid	2,059	0:01	24	31	4.6	9.1	81.9	1	3.3	22	0	78
5	Westview High School	Avondale	85353	755th	31st	Yes	Yes	Yes	60	Hybrid	1,843	21:01	24	31	10.3	10.2	70.4	3.3	5.8	22	0	78
6	Buckeye Union High School	Buckeye	85326	3,772nd	109th	Yes	Yes	Yes	36	In-class	1,718	19:01	21	30	23.9	7.5	64.4	0.3	3.9	20	0	80
7	Verrado High School	Buckeye	85396	1,866th	60th	Yes	Yes	Yes	40	Hybrid	1,715	21:01	25	38	45	5.9	43.6	1.6	3.9	7	0	93
8	Youngker High School	Buckeye	85326	3,378th	106th	Yes	Yes	Yes	46	In-class	1,987	23:01	21	30	22.8	8.8	64.1	1.1	3.3	17	0	83
9	Arizona College Prep Erie Campus	Chandler	85224	9,474th	181st	Yes	Yes	Yes	3	In-class	1,225	22:01	52	51	43.2	3.7	15.2	29.6	8.4	2	0	98
10	Basha High School	Chandler	85249	592nd	26th	Yes	Yes	Yes	24	In-class	2,420	20:01	52	51	64.6	4.4	16.3	9	5.8	3	0	97
11	Chandler High School	Chandler	85225	185th	8th	Yes	Yes	Yes	45	Hybrid	3,549	22:01	52	51	23.9	10.7	54.7	3.8	6.9	19	0	81
12	Hamilton High School	Chandler	85248	64th	1st	Yes	Yes	Yes	37	Hybrid	3,926	21:01	52	51	44.4	7.3	21.5	19	7.8	8	0	92
13	Dysart High School	El Mirage	85335	3,122nd	99th	Yes	No	Yes	134	Hybrid	1,467	21:01	28	37	23.5	8.5	59.6	2.2	6.2	22	0	78
14	Fountain Hills High School	Fountain I	85268	9,922nd	185th	Yes	Yes	Yes	3	Hybrid	481	1:01	27	47	64.7	1.7	14.6	1.9	17.3	9	0	91
15	Gilbert High School	Gilbert	85234	739th	30th	Yes	Yes	Yes	13	Hybrid	2,295	22:01	44	46	53.9	4.1	31.3	3.7	7	9	0	91
16	Higley High School	Gilbert	85295	2,683rd	94th	Yes	Yes	Yes	5	In-class	2,174	21:01	44	46	65.7	3.8	21.1	3.1	6.2	5	0	95
17	Mesquite Jr High School	Gilbert	85233	Unranked	Unknown	No	No	Yes	58	In-class	1,411	23:01	44	46	45.9	4.8	38.1	2.8	8.4	12	0	88
18	Perry High School	Gilbert	85297	131st	5th	Yes	Yes	Yes	21	In-class	3,174	20:01	52	51	67.1	3.4	17.5	8.1	3.9	4	0	96
19	Cactus High School	Glendale	85306	4,643rd	119th	Yes	Yes	Yes	3	Hybrid	1,170	19:01	30	33	52.5	5.1	33.2	2.5	6.8	19	0	81
20	Copper Canyon High School	Glendale	85305	874th	33rd	Yes	Yes	Yes	70	Hybrid	2,195	0:01	24	31	4.6	5.1	86.2	1.6	2.5	23	0	77
21	Deer Valley High School	Glendale	85308	2,141st	76th	Yes	Yes	Yes	38	Hybrid	1,568	10:01	43	46	55.2	5.9	30.5	2.6	5.9	17	0	83
22	Desert Sky Middle School	Glendale	85308	Unranked	Unknown	No	No	Yes	31	In-class	632	22:01	49	52	51.3	3.3	36.1	1.7	7.6	0	0	100
23	Cactus High School	Glendale	85302	4,643	119th	Yes	Yes	Yes	3	Hybrid	1,170	19:01	30	33	52.5	5.1	33.2	2.5	2.3	19	0	81
24	Millennium High School	Goodyear	85395	1,100	38th	Yes	Yes	Yes	40	Hybrid	1,965	21:01	37	38	40.6	9.7	37.4	7.5	4.7	7	0	93
25	Betty Fairfax High School	Laveen	85339	2,112	75th	No	Yes	Yes	23	Hybrid	1,847	21:01	24	24	8.2	20.6	59.6	2.1	9.5	49	9	41
26	Cesar Chavez High School	Laveen	85339	478	18th	Yes	Yes	Yes	26	Hybrid	2,685	20:01	24	24	4.1	14.8	74.7	1.4	4.9	53	9	38
27	Desert Ridge High School	Mesa	85209	400	17th	Yes	No	Yes	35	Hybrid	2,400	0:01	44	46	60.2	3.1	27.9	3.1	5.7	8	0	92
28	Dobson High School	Mesa	85202	501	20th	Yes	Yes	Yes	29	In-class	2,323	20:01	31	30	28.8	7.2	51.8	2.5	9.8	22	0	78
29	Mesa High School	Mesa	85204	113	3rd	Yes	Yes	Yes	50	Hybrid	3,370	20:01	31	30	23.2	4.2	68	0.7	3.9	20	0	80
30	Mountain View High School	Mesa	85213	140	6th	Yes	Yes	Yes	34	Hybrid	3,388	22:01	31	30	59.3	2.8	29.1	1.5	7.2	11	0	89
31	Skyline High School	Mesa	85208	522	22nd	Yes	Yes	Yes	39	Hybrid	2,318	16:01	31	30	42.1	4.1	47.5	1.3	4.3	18	0	82
32	Westwood High School	Mesa	85201	201	10th	Yes	Yes	Yes	51	Hybrid	3,564	20:01	31	30	21.1	6.7	56.6	1.1	14.5	25	0	75
33	Centennial High School	Peoria	85381	1,467	48th	Yes	Yes	Yes	6	Hybrid	2,052	23:01	30	33	47.8	5.7	36.7	3.1	6.7	13	0	87
34	Peoria Flex Academy	Peoria	85381	21,407	439th	No	No	No	0	Hybrid	93	4:01	30	33	31.2	3.2	59.1	0	6.4	30	0	70
35	Sunrise Mountain High School	Peoria	85382	2,765	96th	Yes	Yes	Yes	0	Hybrid	2,025	22:01	30	33	70.9	3	18.2	2.9	4.9	8	0	92
36	Alhambra High School	Phoenix	85019	395	16th	Yes	Yes	Yes	21	In-class	2,395	18:01	24	24	4.2	10.5	78.6	4.7	2.1	78	8	14
37	Dr. Camille Casteel High School	Queen Cre	85142	138	6581	Yes	Yes	Yes	31	In-class	2165	22:01	47	43	71.3	3.2	18.1	2.6	5.8	4	0	4

cleaned_transformed_data (21)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
27	Desert Ridge High School	Mesa	85209	400	17th	Yes	No	Yes	35	Hybrid	2,400	0:01	44	46	60.2	3.1	27.9	3.1	5.7	8	0	92
28	Dobson High School	Mesa	85202	501	20th	Yes	Yes	Yes	29	In-class	2,323	20:01	31	30	28.8	7.2	51.8	2.5	9.8	22	0	78
29	Mesa High School	Mesa	85204	113	3rd	Yes	Yes	Yes	50	Hybrid	3,370	20:01	31	30	23.2	4.2	68	0.7	3.9	20	0	80
30	Mountain View High School	Mesa	85213	140	6th	Yes	Yes	Yes	34	Hybrid	3,388	22:01	31	30	59.3	2.8	29.1	1.5	7.2	11	0	89
31	Skyline High School	Mesa	85208	522	22nd	Yes	Yes	Yes	39	Hybrid	2,318	16:01	31	30	42.1	4.1	47.5	1.3	4.3	18	0	82
32	Westwood High School	Mesa	85201	201	10th	Yes	Yes	Yes	51	Hybrid	3,564	20:01	31	30	21.1	6.7	56.6	1.1	14.5	25	0	75
33	Centennial High School	Peoria	85381	1,467	48th	Yes	Yes	Yes	6	Hybrid	2,052	23:01	30	33	47.8	5.7	36.7	3.1	6.7	13	0	87
34	Peoria Flex Academy	Peoria	85381	21,407	439th	No	No	No	0	Hybrid	93	4:01	30	33	31.2	3.2	59.1	0	6.4	30	0	70
35	Sunrise Mountain High School	Peoria	85382	2,765	96th	Yes	Yes	Yes	0	Hybrid	2,025	22:01	30	33	70.9	3	18.2	2.9	4.9	8	0	92
36	Alhambra High School	Phoenix	85019	395	16th	Yes	Yes	Yes	21	In-class	2,395	18:01	24	24	4.2	10.5	78.6	4.7	2.1	78	8	14
37	Dr. Camille Casteel High School	Queen Cre	85142	138	6581	Yes	Yes	Yes	31	In-class	2165	22:01	47	43	71.3	3.2	18.1	2.6	5.8	4	0	4
38	Queen Creek High School	Queen Cre	85142	61	1891	Yes	No	Yes	28	In-class	2341	21:01	46	47	65.1	3	25.5	1.5	4.9	6	0	6
39	Saguaro High School	Scottsdale	85250	114	4318	No	No	No	19	In-class	1292	21.1:1	38	36	60.1	7.4	22.2	3	10.8	9	0	9
40	Cactus Shadows High School	Scottsdale	85266	90	2512	Yes	Yes	Yes	8	Hybrid	1714	25.9:1	49	45	80.6	0.9	13.4	2.3	2.3	3	0	3
41	Desert Mountain High School	Scottsdale	85259	35	1009	Yes	Yes	Yes	26	Hybrid	2244	23.1:1	49	47	74.8	2.9	9.5	10	2	3	0	3
42	Imagine Prep Surprise	Surprise	85379	265	14874	No	Yes	Yes	0	Hybrid	293	N/A1:1	17	27	53.9	3.9	32.2	2.3	6.9	10	10	0
43	Valley Vista High School	Surprise	85374	27	608	Yes	No	Yes	104	Hybrid	2531	26.8:1	25	32	35.3	9.2	46.7	1.7	7.3	20	20	0
44	Sierra Linda High School	Tolleson	85353	62	1898	Yes	Yes	Yes	52	Hybrid	1,896	24.3:1	10	14	3.3	7.9	85.1	0.9	2.7	25	0	25
45	Maryvale High School	Phoenix	85033	273	14	Yes	Yes	Yes	24	In-class	2,768	21:01	24	24	2.6	4	91	0.5	2.3	80	8	12
46	Metro Tech High School	Phoenix	85015	2,497	89	Yes	Yes	Yes	4	Hybrid	1,802	18:01	24	24	1.2	1.2	96.4	0.6	0.5	79	11	10
47	Moon Valley High School	Phoenix	85029	3,474	107	Yes	Yes	Yes	36	In-class	1,451	20:01	18	30	30.7	8.5	50.7	3.4	7.4	25	0	75
48	North Canyon High School	Phoenix	85024	1,907	64	Yes	Yes	Yes	50	Hybrid	1,919	23:01	33	40	22.7	7.6	61.9	1.9	5.9	23	0	77
49	North High School	Phoenix	85014	573	24	Yes	Yes	Yes	27	In-class	2,194	18:01	24	24	3.8	6.8	85.2	0.7	3.5	81	6	13
50	Paradise Valley High School	Phoenix	85032	2,267	79	Yes	Yes	Yes	21	Hybrid	1,865	18:01	33	40	39.6	3.5	43.6	8.3	5	14	0	86
51	Phoenix Union Bioscience High School	Phoenix	85004	14,142	254	Yes	Yes	No	0	In-class	377	17:01	30	30	12.5	4	71.1	8.2	3.2	50	15	35
52	Pinnacle High School	Phoenix	85050	696	28	Yes	Yes	Yes	18	In-class	2,558	0:01	33	40	74.9	1.8	13.6	4.3	5.4	4	0	96
53	Roadrunner School	Phoenix	85028	17,197	312	No	No	No	8	In-class	16	4:01	33	40	48	5.4	31.2	9.4	5	25	0	75
54	Shadow Mountain High School	Phoenix	85028	3,930	111	Yes	Yes	Yes	50	Hybrid	1,171	15:01	33	40	53	4.8	33.1	1.8	7.3	16	0	84
55	South Mountain High School	Phoenix	85040	2,476	88	No	No	Yes	49	Hybrid	2,143	18:01	24	24	78.3	16.4	2.5	0.2	2.6	77	7	16
56	Sunnyslope High School	Phoenix	85021	1,512	50	Yes	Yes	Yes	17	Hybrid	2,262	1:01	18	30	36.8	4.8	52.8	1	4.6	21	0	79
57	Thunderbird High School	Phoenix	85023	3,218	101	Yes	Yes	Yes	15	Hybrid	1,615	22:01	18	30	38.3	7.8	45.8	1.7	6.4	19	0	81
58	Trevor Browne High School	Phoenix	85033	248	13	Yes	Yes	Yes	29	Hybrid	2,847	19:01	24	24	2.3	4.8	90.8	0.6	1.5	80	8	12
59	Vista Peak	Phoenix	85027	22,127	462	No	No	No	2	In-class	24	4:01	43	46	21.9	12.5	59.4	1.6	4.6	33	0	67
60	Washington High School	Phoenix	85021	2,405	84	Yes	Yes	Yes	58	Hybrid	1,845	23:01	18	30	11.9	11.2	64.5	6	6.2	33	0	67
61																						
62																						
63																						

cleaned_transformed_data (21)

Explanation:

The final cleaned dataset consists of 22 columns and 59 rows.

Data Transformation and Cleaning:

Low attribute weight rows, duplicate entries, and missing values have all been removed from the dataset.

Special characters ('$' and '%') and other undesired characters ('th','st', 'nd', 'rd', and ',') have been eliminated from numerical columns through cleaning.

Complete Dataset:

With characteristics like "School Name," "City," "Zip Code," "National Rank," "Arizona Rank," "AP Classes," "Dual Enrollments," and other scores and percentages, each row represents a school.

Nominal attributes represent categories or labels without any inherent order. In the dataset, attributes like "School Name," "City," "Zip Code," "Offer Electives?," "Offers Sports?," "AP Classes," "Teaching and Educational Method," and "Mental Health Services" are nominal.

Ordinal attributes have a natural order, but the differences between values may not be uniform or meaningful. Examples include "National Rank" and "Arizona Rank," where schools are ranked, but the difference between ranks may not reflect a consistent difference in performance.

Ratio attributes have a true zero point and meaningful ratios between values. In the dataset, "Crime Related Data" (assuming it represents count or rate), "Student Teacher Ratio," "Math Score," "English Score," and "2022N/A2023 Student Enrollment" fall into this category.

Each type of attribute plays a distinct role in understanding and analyzing the dataset, providing valuable insights into various aspects of the educational institutions.