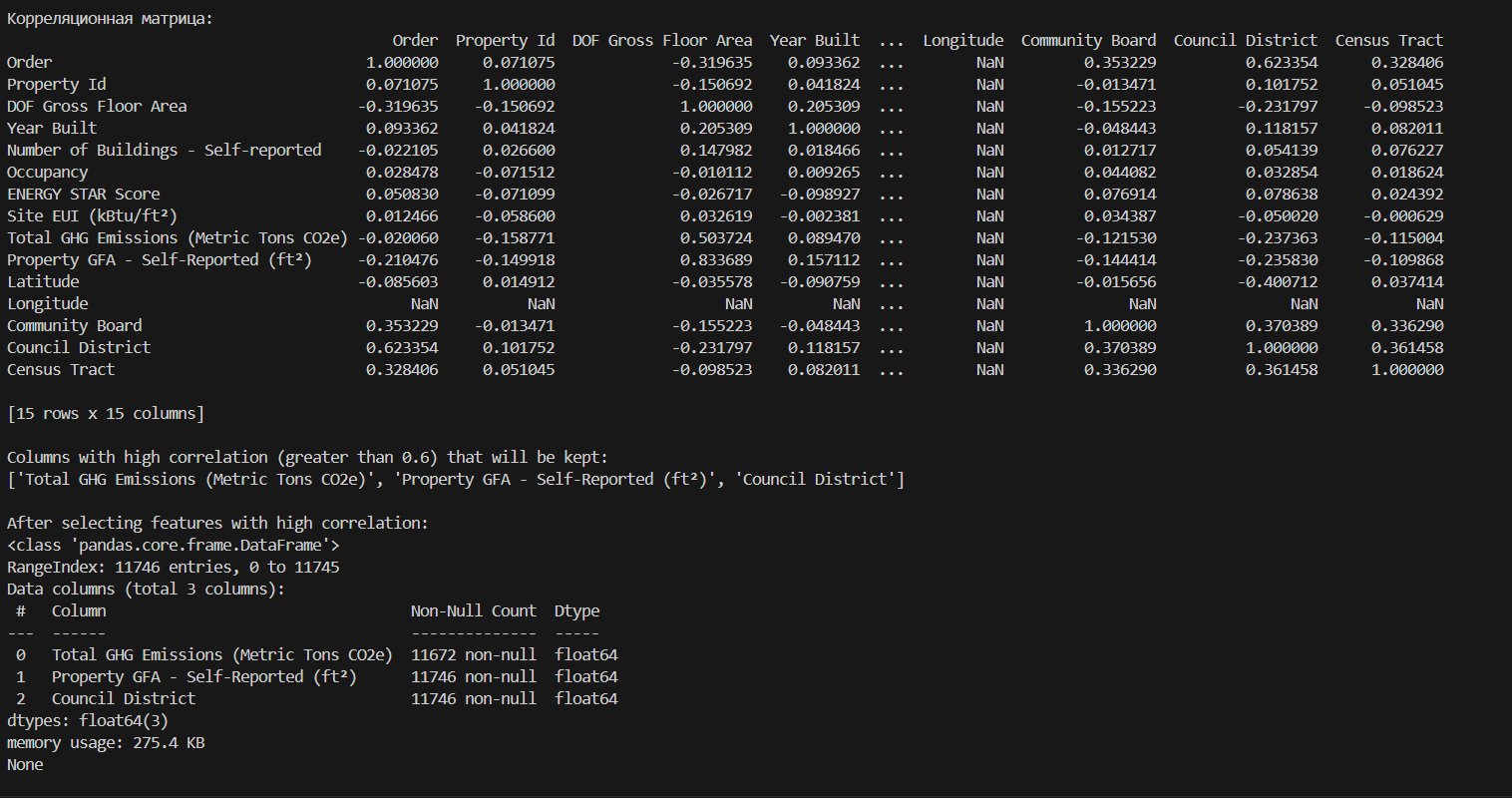
|  |  |  |
| --- | --- | --- |
| Лабораторная работа №4  Знакомство с python | Ф.И.О. | Алхабил Бахаа А М |
| Группа | Прин 368 |
| Преподаватель | Сибирный |
| Дата сдачи |  |
| Оценка |  |



**1. Correlation Matrix:**

The **correlation matrix** shows the pairwise correlation coefficients between different numerical columns in your dataset. These coefficients measure how closely related two variables are:

* **Values closer to 1** indicate a strong positive correlation (both variables increase together).
* **Values closer to -1** indicate a strong negative correlation (one variable increases while the other decreases).
* **Values closer to 0** suggest weak or no correlation between the variables.

**Key observations from your correlation matrix:**

* Columns such as Council District and Latitude have strong correlations with other variables.
* Longitude has a **NaN (Not a Number)** value across the entire column, meaning it's completely missing in the dataset.
* Total GHG Emissions (Metric Tons CO2e) and Property GFA - Self-Reported (ft²) have a moderately high positive correlation, meaning these two variables are related.

**2. Columns with High Correlation (Greater than 0.6):**

* After calculating the correlation matrix, the script identifies columns that have a high correlation (greater than 0.6).
* In your case, the columns kept for further analysis are:
  + **Total GHG Emissions (Metric Tons CO2e)**
  + **Property GFA - Self-Reported (ft²)**
  + **Council District**

These are the columns that will remain in the dataset for further analysis, as they show a strong relationship with other variables.

**3. Dataframe after Selecting High Correlation Features:**

After filtering for columns with high correlation, the resulting dataset (df\_reduced) contains only the three columns mentioned above:

* **Total GHG Emissions (Metric Tons CO2e)**
* **Property GFA - Self-Reported (ft²)**
* **Council District**

**Details:**

* The dataset still has **11746 entries** (rows), but only **3 columns**.
* The dtype for all three columns is float64, which means the data is stored as floating-point numbers.
* The column Total GHG Emissions (Metric Tons CO2e) has **74 missing values**, as shown by the "Missing Values" count.

**4. Missing Values Table:**

This table reports how many missing values exist in each column of the reduced dataset:

* **Total GHG Emissions (Metric Tons CO2e)** has **74 missing values**, which is **0.6%** of the total entries in that column.

**5. After Removing Columns with Missing Values:**

* The script removes any columns with missing values. Since the column Total GHG Emissions (Metric Tons CO2e) has 74 missing values, it is removed from the dataset.
* The remaining columns are:
  + **Property GFA - Self-Reported (ft²)**
  + **Council District**

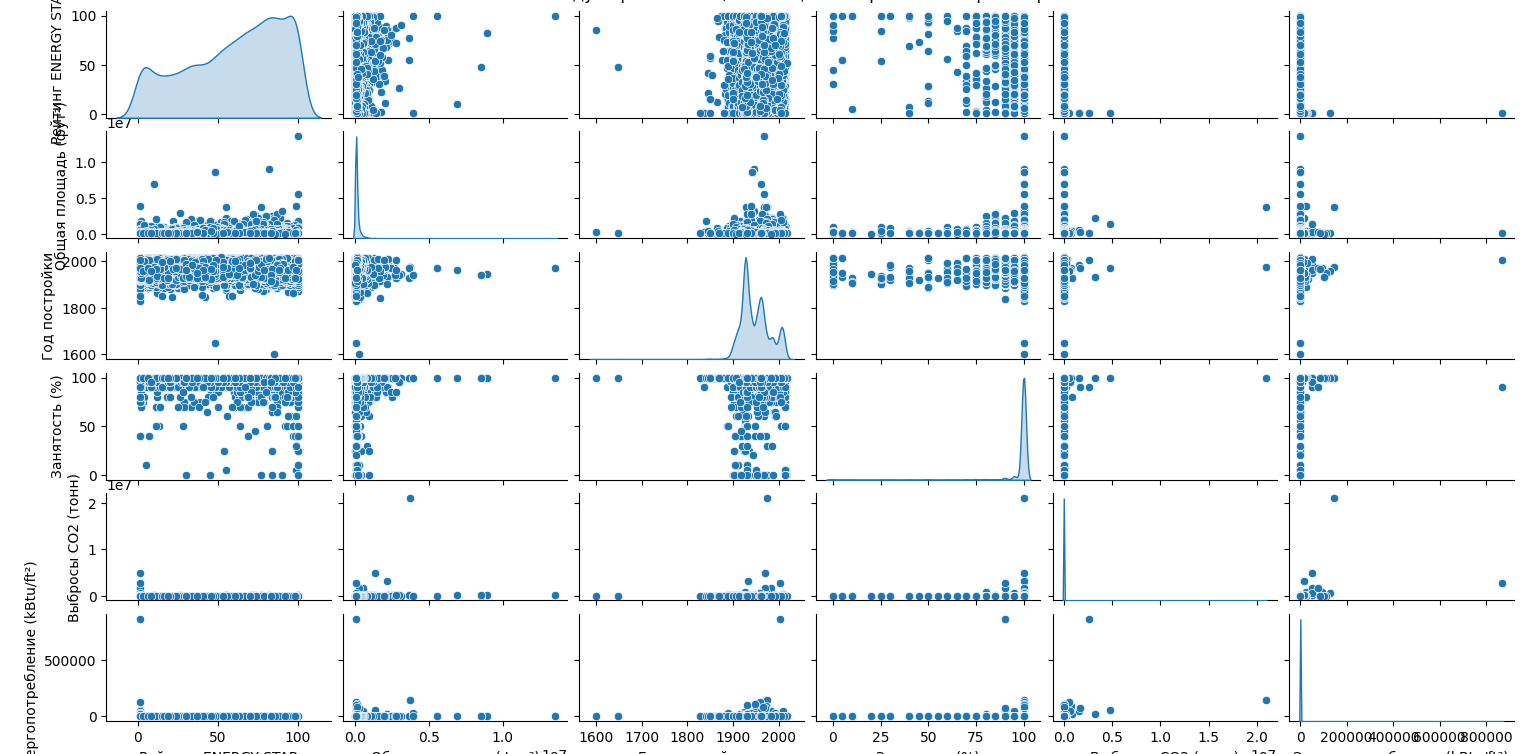
These columns no longer have any missing values, and the dataset now has only **2 columns**.

The resulting DataFrame is printed with the following details:

* **11746 entries** (rows)
* **2 columns** remaining: Property GFA - Self-Reported (ft²) and Council District.
* The **memory usage** of the new DataFrame is **183.7 KB**.

**6. Final DataFrame:**

After removing columns with missing values, the final DataFrame has:

* No missing values in any of the columns.
* Only two columns, making it a more concise and clean dataset for analysis.

**Summary:**

The output shows the correlation between various columns in your dataset, identifies columns with high correlations, and performs cleaning by removing columns with missing values. The dataset is reduced from 15 columns to 2 (Property GFA - Self-Reported (ft²) and Council District), making it more manageable for further analysis.