**AI-DRIVEN EXPLORATION AND PREDICTION OF COMPANY REGISTRATION TRENDS WITH REGISTRAR OF COMPANIES (RoC)**

**INSERTING LIBRARIES:**import numpy as np

import pandas as pd

import seaborn as sns

import os

import matplotlib as plt

import matplotlib.pyplot as plt

**numpy (np):**

For handling large, multi-dimensional arrays and matrices, and performing mathematical functions on them.

**pandas (pd):**

Provides high-level data structures and tools for data analysis, including DataFrames for in-memory 2D tables.

**seaborn (sns):**

A visualization library for statistical plotting, building on matplotlib and integrating with pandas.

os:

Offers OS-dependent functionality, such as interfacing with the file system.

**matplotlib (plt):**

Note: The initial import is redundant. The library is used for creating a wide range of static and interactive visualizations.

**matplotlib.pyplot (plt):**

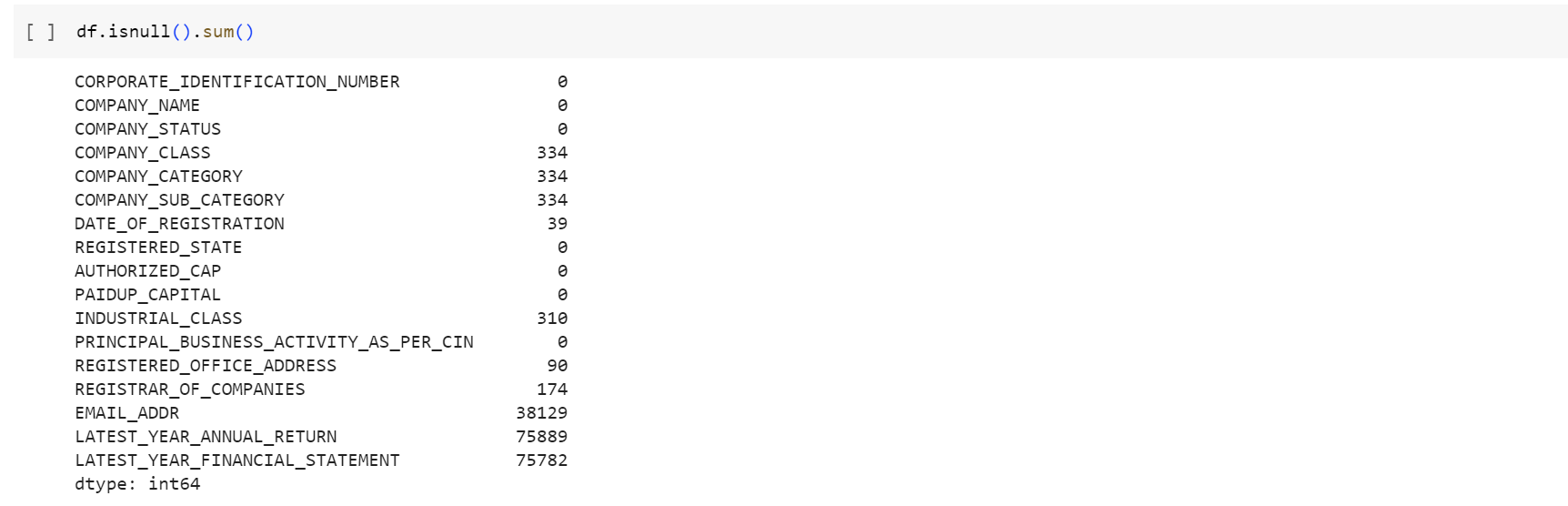
A submodule in matplotlib that offers a state-based interface for creating visualizations.

These libraries collectively support various stages of data manipulation, analysis, and visualization in Python.

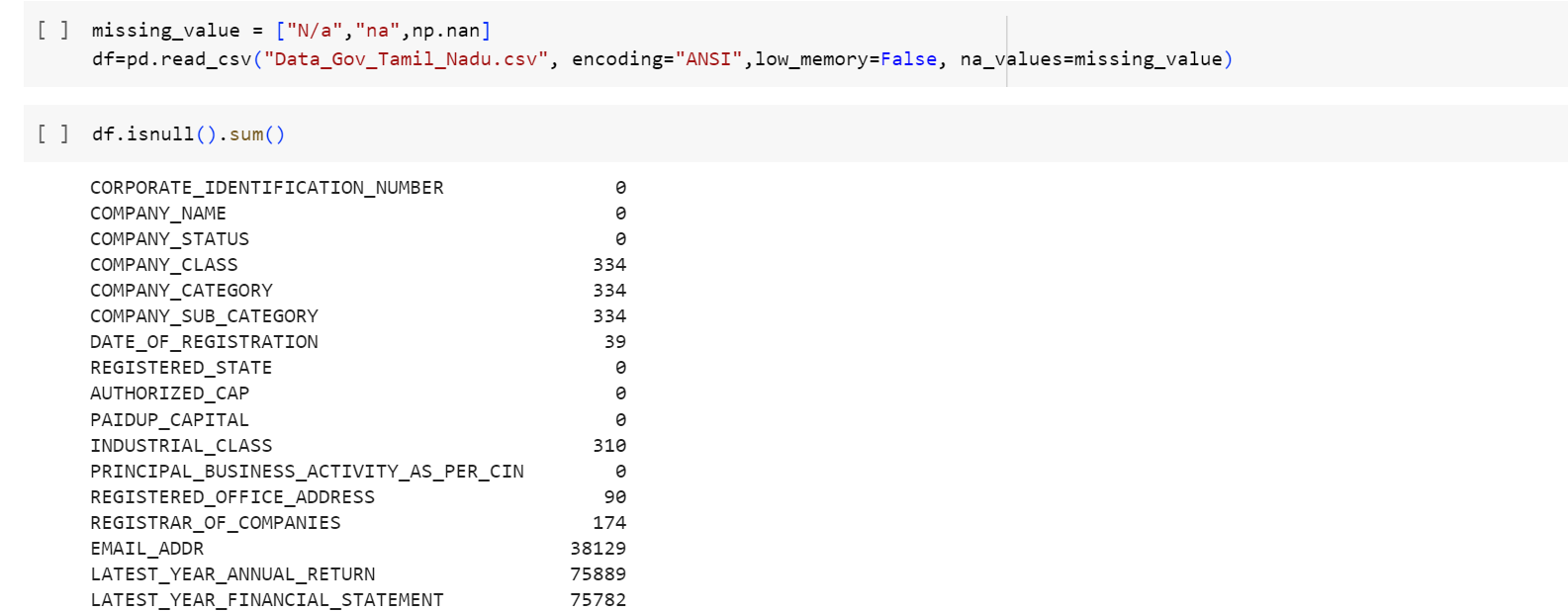
**LOADING CSV DATASET**

****

**DISPLAYING NULL VALUES**

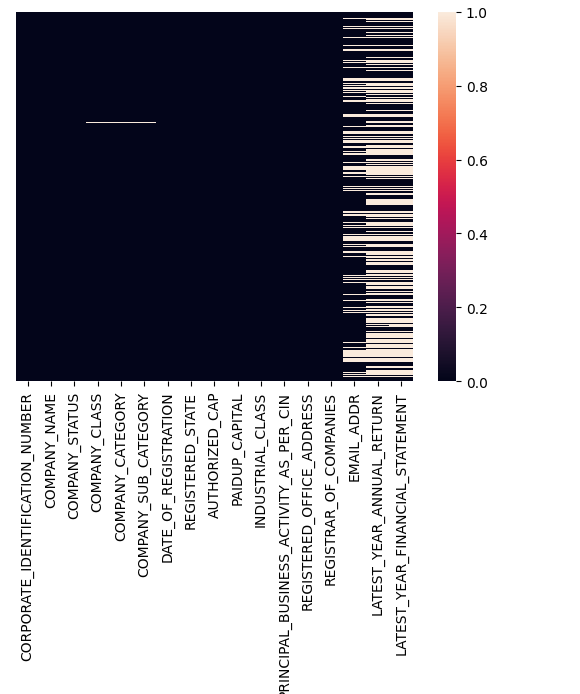


**HANDLING NULL VALUES**



na\_values=missing\_value. This means that while reading the CSV, if any of the values in the list missing\_value are encountered, they will be treated as NaN (Not a Number) values, which is a standard way of representing missing data in pandas.

**VISUALISATION OF NULL VALUES USING A HEAT MAP**





This line drops rows from the DataFrame df where all of its elements are NaN (missing values). The resulting DataFrame after dropping such rows is stored in df\_dropped.

The dropna() method is used to drop missing values.

**DROPPING DUPLICATE VALUES**

df\_dupfree =df.drop\_duplicates(["CORPORATE\_IDENTIFICATION\_NUMBER","COMPANY\_NAME","COMPANY\_STATUS","COMPANY\_CLASS","COMPANY\_CATEGORY","COMPANY\_SUB\_CATEGORY","DATE\_OF\_REGISTRATION","REGISTERED\_STATE","AUTHORIZED\_CAP","PAIDUP\_CAPITAL","INDUSTRIAL\_CLASS","PRINCIPAL\_BUSINESS\_ACTIVITY\_AS\_PER\_CIN","REGISTERED\_OFFICE\_ADDRESS","REGISTRAR\_OF\_COMPANIES","EMAIL\_ADDR","LATEST\_YEAR\_ANNUAL\_RETURN","LATEST\_YEAR\_FINANCIAL\_STATEMENT"])

This line removes duplicate rows from the df DataFrame based on a specified set of columns. The drop\_duplicates() method is used to drop duplicate rows.

The columns passed inside the method specify the criteria for determining duplicates. In this case, a row is considered a duplicate if all the mentioned columns have the same values as another row.

# HANDLING MISSING DATA

# For Categorical Columns:

## Handling missing values in Date\_Of\_Registration

# 

## Handling missing values in Industrial\_Class



# .fillna() is a pandas method to fill NA/NaN values using the specified method. In this case, the method chosen is to fill missing values with the mode of the column.

# Here, mode() calculates the mode of the 'COMPANY\_CLASS' column. The mode is the value that appears most frequently in a column. Since the mode() method returns a Series (because a column can technically have more than one mode), [0] is used to select the first mode.

# inplace=True ensures that the changes are made directly to the df\_dupfree DataFrame without the need to assign the result back to df\_dupfree.

# Fill missing values in the REGISTERED\_OFFICE\_ADDRESS and REGISTRAR\_OF\_COMPANIES column with "UNKNOWN"

# 

This line replaces missing values in the columns REGISTERED\_OFFICE\_ADDRESS, REGISTRAR\_OF\_COMPANIES, of the df\_dupfree DataFrame with the string "UNKNOWN".

## Count of Null values after data handling

## 

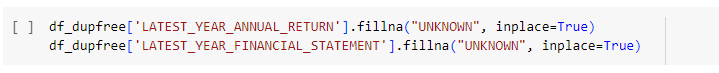
## 

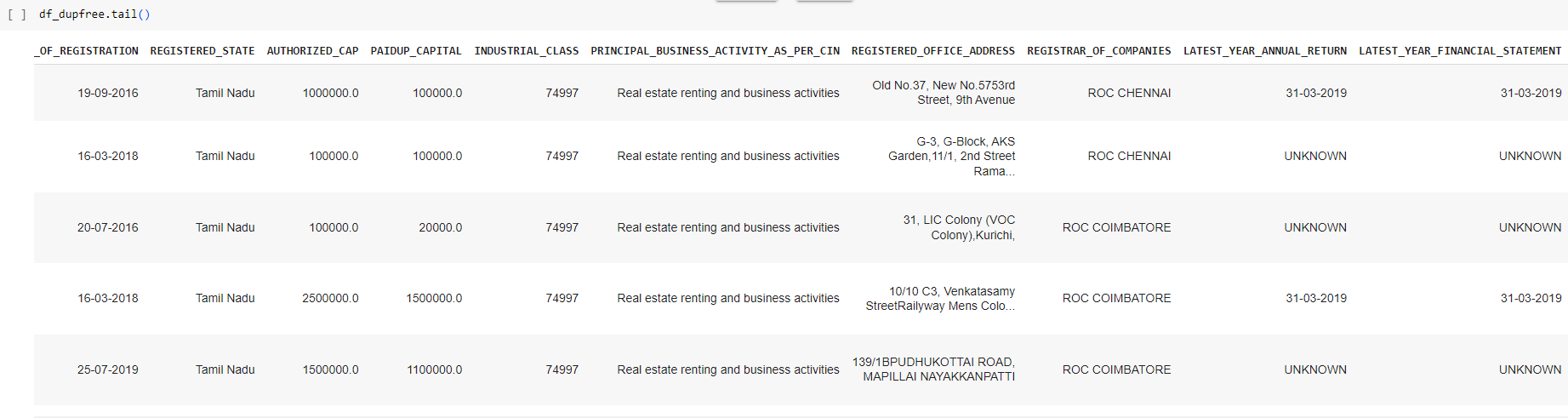
## We should handle the columns EMAIL\_ADDR, LATEST\_YEAR\_ANNUAL\_RETURN and LATEST\_YEAR\_FINANCIAL\_STATEMENT. Every time, after handling a column, check changes to the dataset which will help in deciding what to do next.

## DROPPING IRRELEVANT DATA

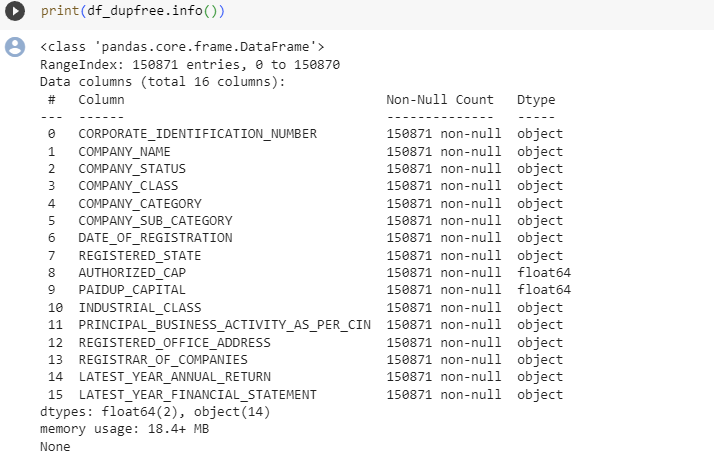
This line  drops the EMAIL\_ADDR column, as it is not relevant to the analysis.

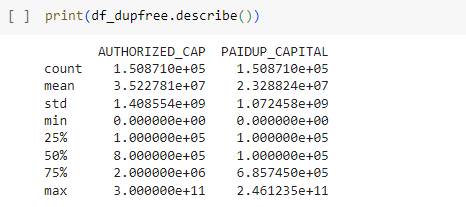
## Fill missing values in the LATEST\_YEAR\_ANNUAL\_RETURN and LATEST\_YEAR\_FINANCIAL\_STATEMENT column with "UNKNOWN"

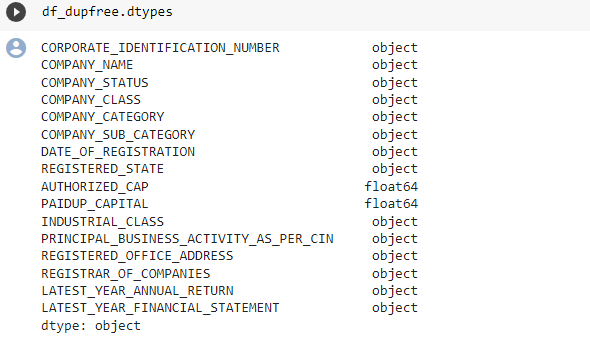




**GET AN OVERVIEW OF DATA**

****

**SUMMARY STATISTICS  
**

****

**Correlation matrix**

****

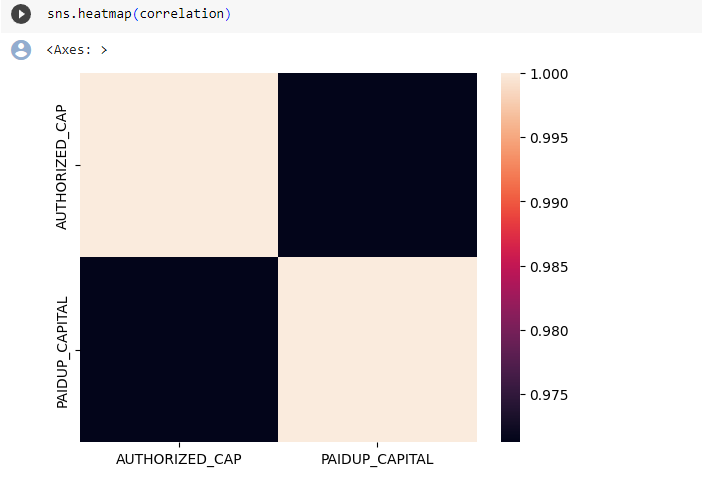
This line calculates the Pearson correlation coefficient between the columns "AUTHORIZED\_CAP" and "PAIDUP\_CAPITAL". The result is a matrix (in the form of a DataFrame) that contains correlation coefficients between the provided columns.

The .corr() method calculates the Pearson correlation, which provides a value between -1 and 1 indicating the extent to which two variables change in relation to each other.

* A value closer to 1 implies a strong positive correlation: as one variable increases, the other also tends to increase.
* A value closer to -1 implies a strong negative correlation: as one variable increases, the other tends to decrease.
* A value closer to 0 implies little to no linear relationship between the variables.

The correlation between these two columns might give insights into how closely companies tend to operate to their authorized limits.

These fields have been chosen specifically because understanding the relationship between authorized and paid-up capital is crucial for financial analysis or understanding the financial health and strategies of companies.

****

This line uses the seaborn library to visualize the correlation matrix as a heatmap. In a heatmap, data values are represented as colors. Darker or lighter colors in the heatmap will represent the strength and direction (positive or negative) of the correlation.