TITLE: AIR QUALITY ANALYSIS IN TAMIL NADU

PHASE 3: DEVELOPMENT PART ONE

Importing Libraries

import numpy as np

import pandas as pd

import seaborn as sns

from sklearn.preprocessing import Imputer

import matplotlib.pyplot as plt

%matplotlib inline

plt.rcParams['figure.figsize'] = (10, 7)

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_log\_error

from sklearn.metrics import mean\_squared\_error

from sklearn.metrics import r2\_score, mean\_squared\_error

from sklearn.feature\_selection import RFE

from sklearn.linear\_model import Ridge

from sklearn.linear\_model import Lasso

import statsmodels.formula.api as sm

from sklearn.model\_selection import KFold

from sklearn.model\_selection import cross\_val\_score

from statsmodels.regression.linear\_model import OLS

from statsmodels.tools import add\_constant

from sklearn.preprocessing import StandardScaler

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import classification\_report

from sklearn import metrics

from statsmodels.stats.outliers\_influence import variance\_inflation\_factor

import warnings; warnings.simplefilter('ignore')

\*NumPy can be used to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.

\* Pandas is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data. The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008

\*Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures. Seaborn helps you explore and understand your data.

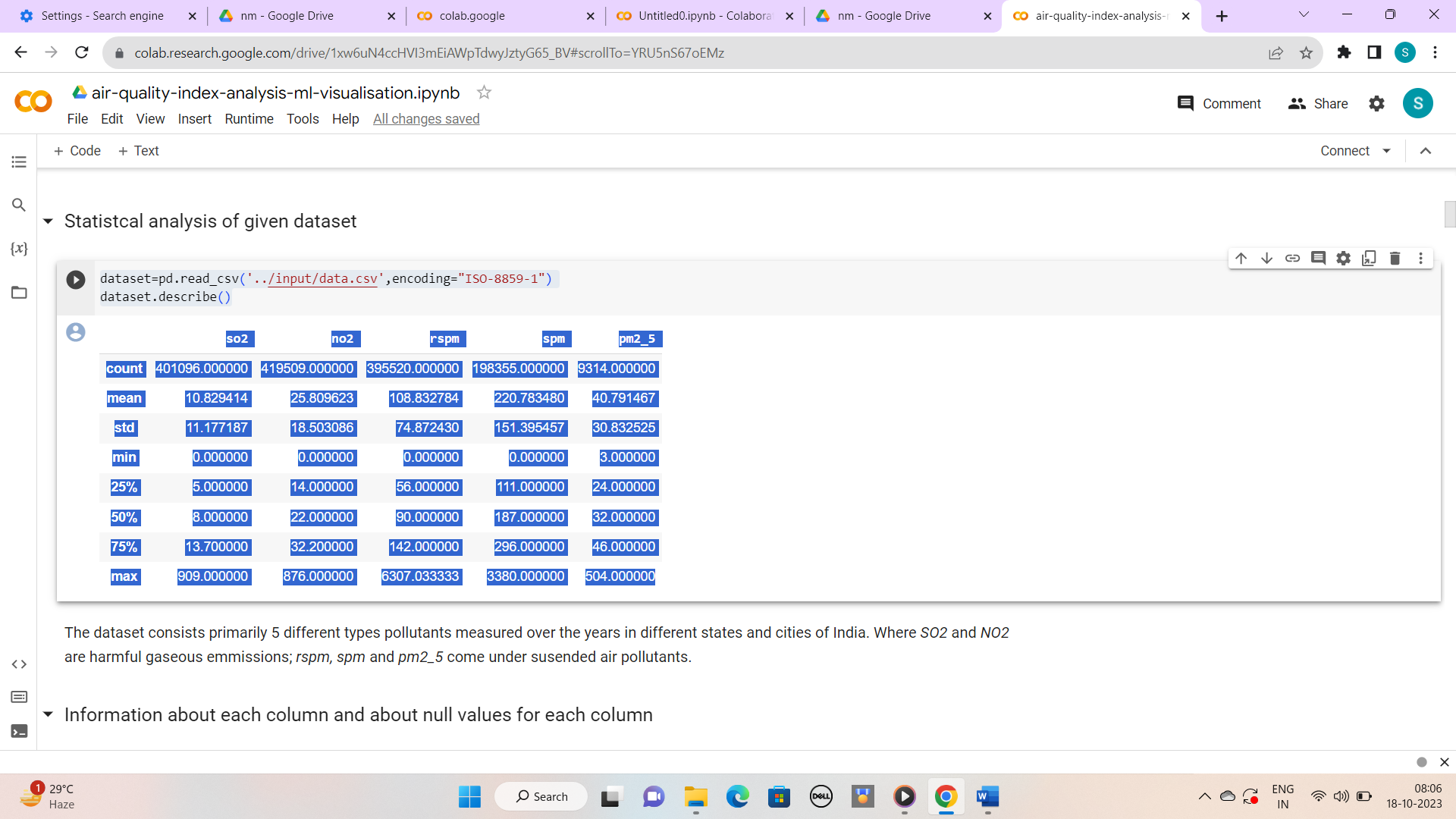
\*Scikit-Learn, also known as sklearn is a python library to implement machine learning models and statistical modelling. Through scikit-learn, we can implement various machine learning models for regression, classification, clustering, and statistical tools for analyzing these models.

# Dataset Summary

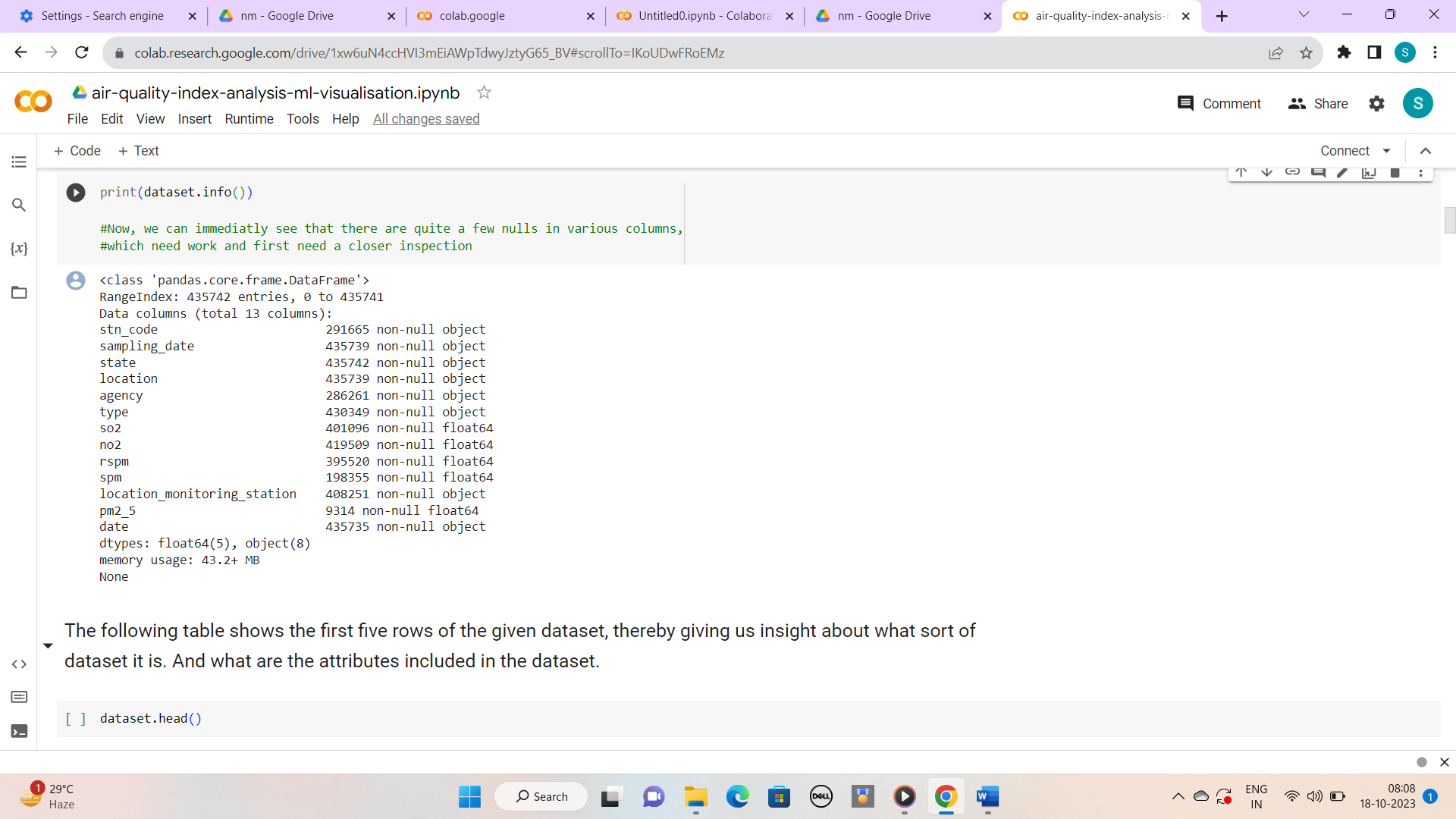
### **Statistcal analysis of given dataset**

dataset=pd.read\_csv('../input/data.csv',encoding="ISO-8859-1")

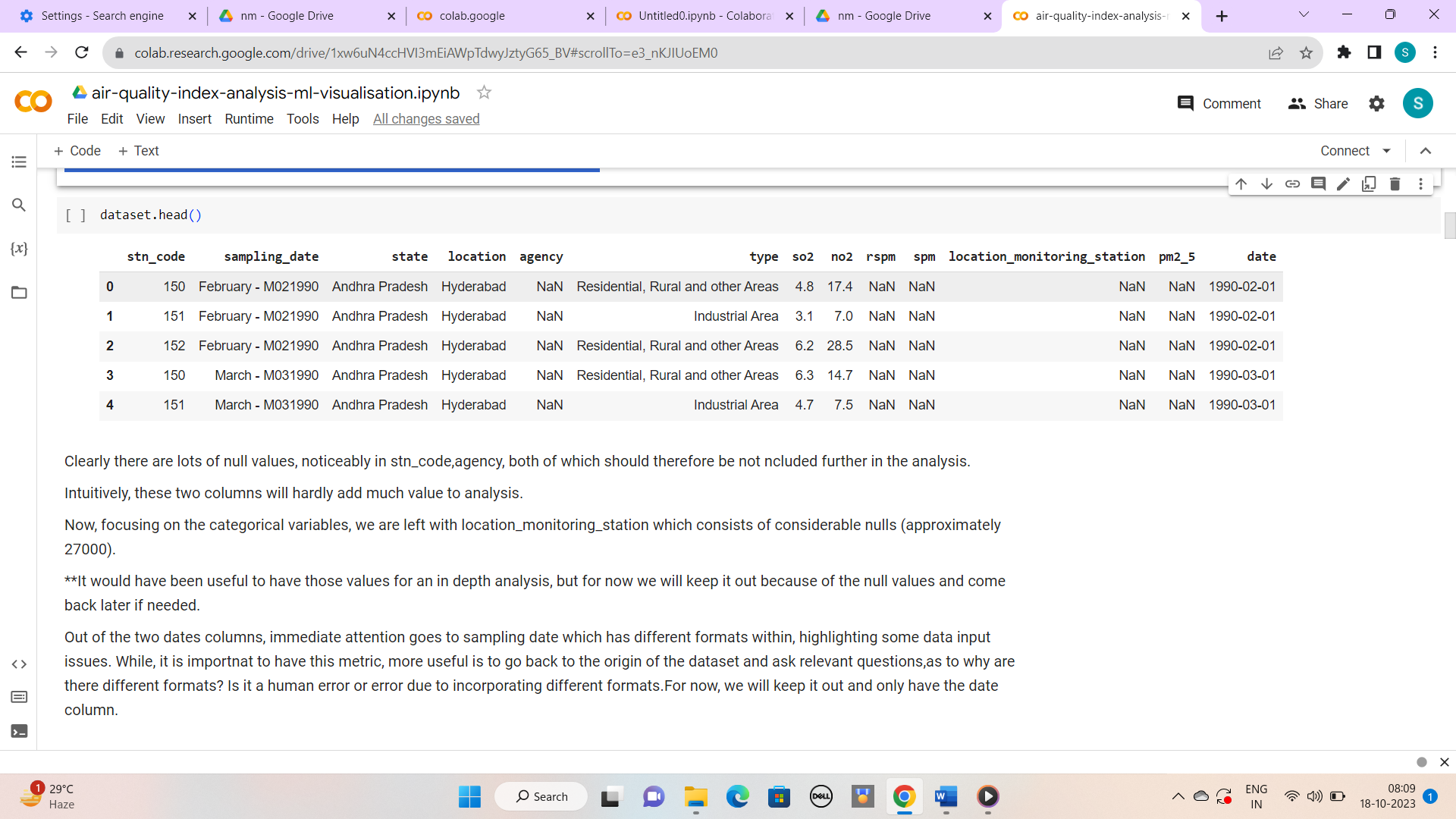
dataset.describe()



### **Information about each column and about null values for each column**

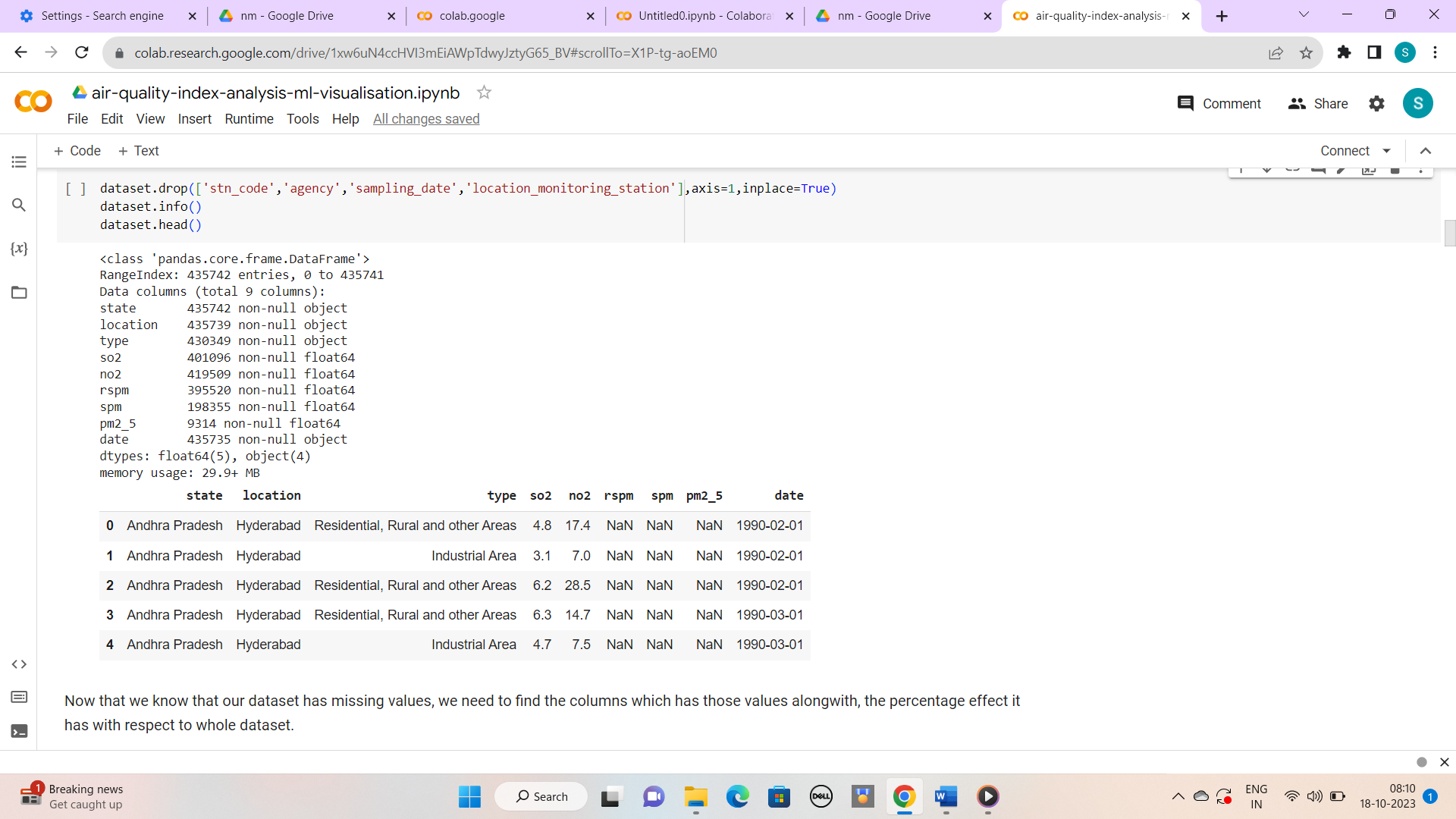


### **The following table shows the first five rows of the given dataset, thereby giving us insight about what sort of dataset it is. And what are the attributes included in the dataset.**



# Dataset Cleaning

### **Following tables gives information about new dataset after dropping of unneccessary columns**



Often a DataFrame will contain columns that are not useful to your analysis. Such columns should be dropped from the DataFrame to make it easier for you to focus on the remaining columns. The columns can be removed by specifying label names and corresponding axis, or by specifying index or column names directly.

\*code:

dataset.drop(['stn\_code','agency','sampling\_date','location\_monitoring\_station'],axis=1,inplace=True)

dataset.info()

dataset.head()

FINDING THE MISSING VALUES IN THE DATASET:

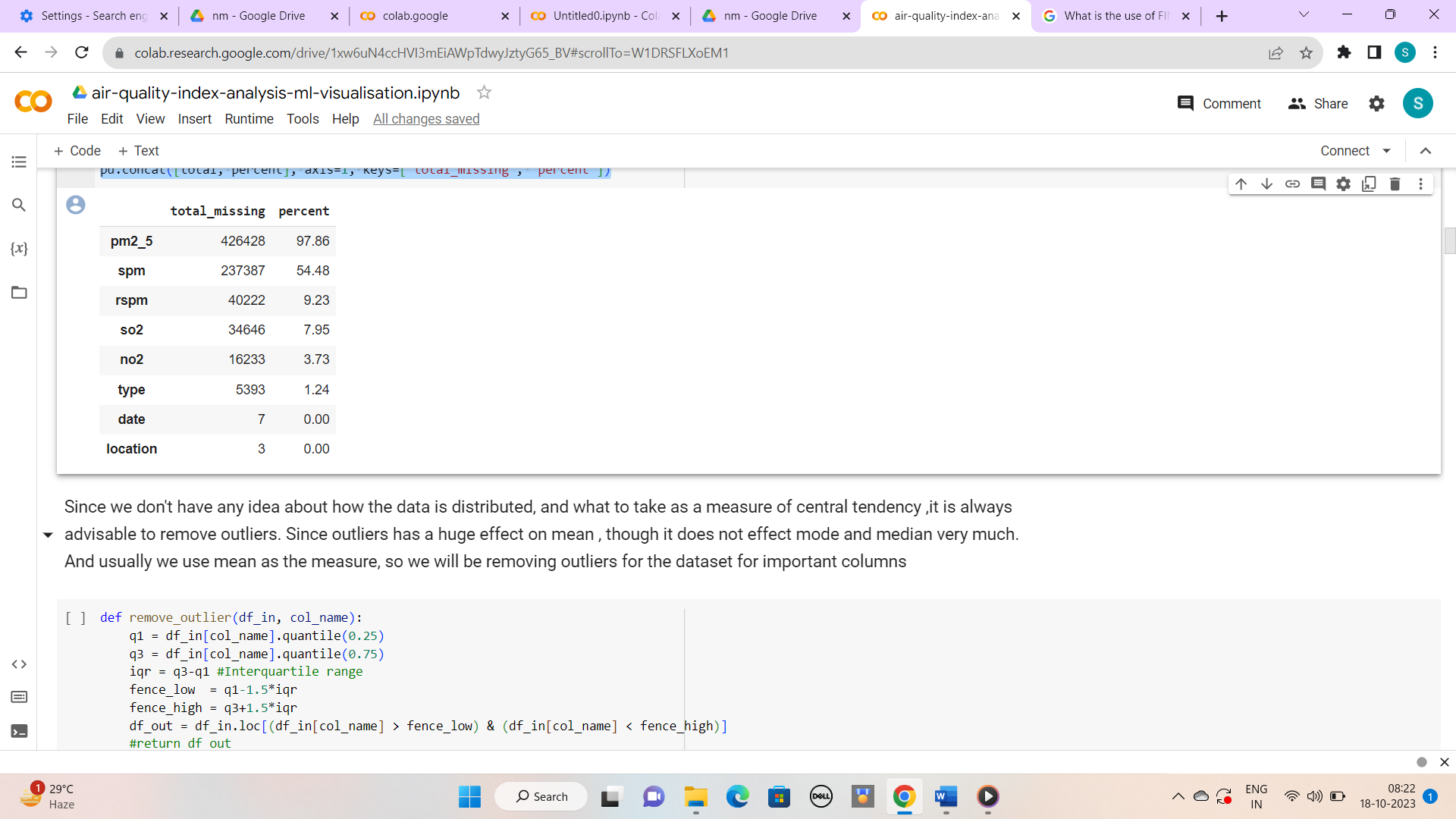
 If you are aiming for a job as a data scientist, you must know how to handle the problem of missing values, which is quite common in many real-life datasets. Incomplete data can bias the results of the machine learning models and/or reduce the accuracy of the model.

\*CODE:

total = dataset.isnull().sum()[dataset.isnull().sum() != 0].sort\_values(ascending = False)

percent = pd.Series(round(total/len(dataset)\*100,2))

pd.concat([total, percent], axis=1, keys=['total\_missing', 'percent'])



REMOVING OUTLIERS:

Outliers can distort statistical analyses and skew results as they are extreme values that differ from the rest of the data. Removing outliers makes the results more robust and accurate by eliminating their influence.

CODE:

def remove\_outlier(df\_in, col\_name):

    q1 = df\_in[col\_name].quantile(0.25)

    q3 = df\_in[col\_name].quantile(0.75)

    iqr = q3-q1 #Interquartile range

    fence\_low  = q1-1.5\*iqr

    fence\_high = q3+1.5\*iqr

    df\_out = df\_in.loc[(df\_in[col\_name] > fence\_low) & (df\_in[col\_name] < fence\_high)]

    #return df\_out

remove\_outlier(dataset,'so2')

remove\_outlier(dataset,'no2')

remove\_outlier(dataset,'rspm')

remove\_outlier(dataset,'spm')