**COVID-19 VACCINE ANALYSIS**

**PHASE-5**

**PROJECT DOCUMENTATION AND SUBMISSION**

**PUGALEESHWARAN**

**(TEAM MEMBER)**

* COVID-19 analysis data refers to the collection, organization, and interpretation of information related to the COVID-19 pandemic. This data is critical for understanding the spread and impact of the virus, making informed decisions, and formulating public health strategies.

* COVID-19 analysis data is continually updated and analyzed by epidemiologists, public health officials, researchers, and policymakers to inform strategies for mitigating the spread of the virus, managing healthcare resources, and guiding vaccination efforts during the pandemic. It plays a central role in our collective response to this global health crisis.

**DESIGN THINKING:(STEPS)**

* DATA COLLECTION
* DATA PREPROCESSING
* EXPLORATORY DATA ANALYSIS
* STATISTICAL ANALYSIS
* VISUALIZATION
* INSIGHTS AND RECOMMENDATIONS

ABOUT THE DATASET:

LINK TO GET THE DATASET USED IN THIS PROJECT:

<https://www.kaggle.com/datasets/gpreda/covid-world-vaccination-progress>

A dataset is a structured collection of data. It is a fundamental unit for organizing, storing, and managing information for various purposes, such as analysis, research, and decision-making. Datasets are typically composed of individual data points or records, and each record contains specific information or attributes related to a particular subject or event. Here are some key characteristics of datasets:

Data Points: Datasets consist of individual data points or observations. Each data point represents a specific piece of information. For example, in a dataset of sales transactions, each data point might represent a single sale and include details like the product sold, the date of the sale, and the sale price.

Attributes: Each data point within a dataset is associated with one or more attributes, also known as features or variables. These attributes describe different aspects of the data. In the sales transaction dataset example, attributes could include product ID, date, and price.

Structure: Datasets have a defined structure that specifies how the data is organized. The structure can be tabular, where data is organized in rows and columns, or it can take other forms such as hierarchical, network, or graph structures.

Purpose: Datasets are collected and organized for specific purposes. These purposes can vary widely and may include scientific research, business analytics, machine learning, and more.

Size: Datasets can vary in size from small datasets with just a few data points to very large datasets containing millions or even billions of data points.

Source: Datasets can originate from a wide range of sources, including scientific experiments, surveys, observations, databases, web scraping, and more.

Data Format: Datasets are typically stored in specific data formats, such as spreadsheets (e.g., CSV files), databases, JSON, XML, or other structured formats, depending on the type of data and its intended use.

Accessibility: Datasets can be made available for public use, restricted to specific organizations or researchers, or kept private depending on the data owner's preferences and data-sharing policies.

Datasets are essential for various data-related tasks, including data analysis, machine learning, statistical research, and decision-making. Researchers and analysts use datasets to draw insights, make predictions, and test hypotheses. They are a cornerstone of data-driven approaches in a wide range of fields, from science and healthcare to business and social sciences.

DATACOLLECTION:

Data collection in Python refers to the process of gathering, retrieving, or obtaining data from various sources or input methods for further processing, analysis, or storage. Python provides a wide range of tools, libraries, and techniques for collecting data from various sources

DATA PREPROCESSING:

**Data Cleaning**:

* + 1. Handling Missing Data: Identify and deal with missing values, either by removing rows or columns, imputing values, or using advanced techniques like predictive modeling.
    2. Removing Duplicates: Detect and remove duplicate records from the dataset.

**Data Transformation**:

* + 1. Data Scaling: Standardize or normalize numeric features to ensure that they have similar scales. This is important for algorithms like k-means clustering and support vector machines.

EXPLANATION:

* First the data is collected by importing the data set . Next the data is made into dataframes to look deep into the data
* In Python, importing datasets typically involves loading data from external files or sources into your Python environment so that you can analyze, manipulate, or work with the data. There are several common methods for importing datasets in Python:

**Reading from Text File**

**Using APIs**

**Reading from Databases**

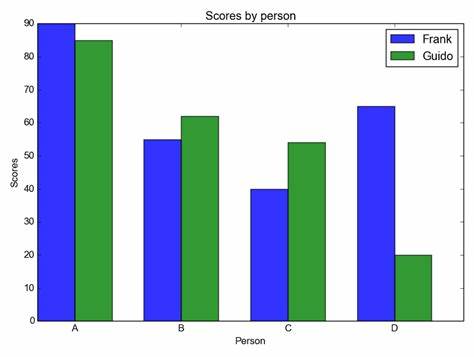
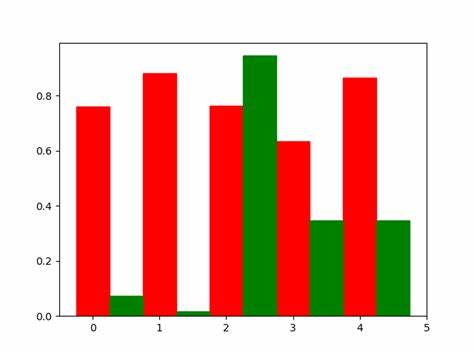
**Web Scraping**

EXPLORATORY DATA ANALYSIS:

Exploratory Data Analysis (EDA) in Python is a critical process that involves visually and statistically summarizing and exploring a dataset to gain insights, discover patterns, identify anomalies, and formulate hypotheses. EDA is typically one of the first steps in any data analysis or data science project. Python provides various libraries, such as **pandas**, **matplotlib**, **seaborn**, and **numpy**, which are commonly used for conducting EDA.

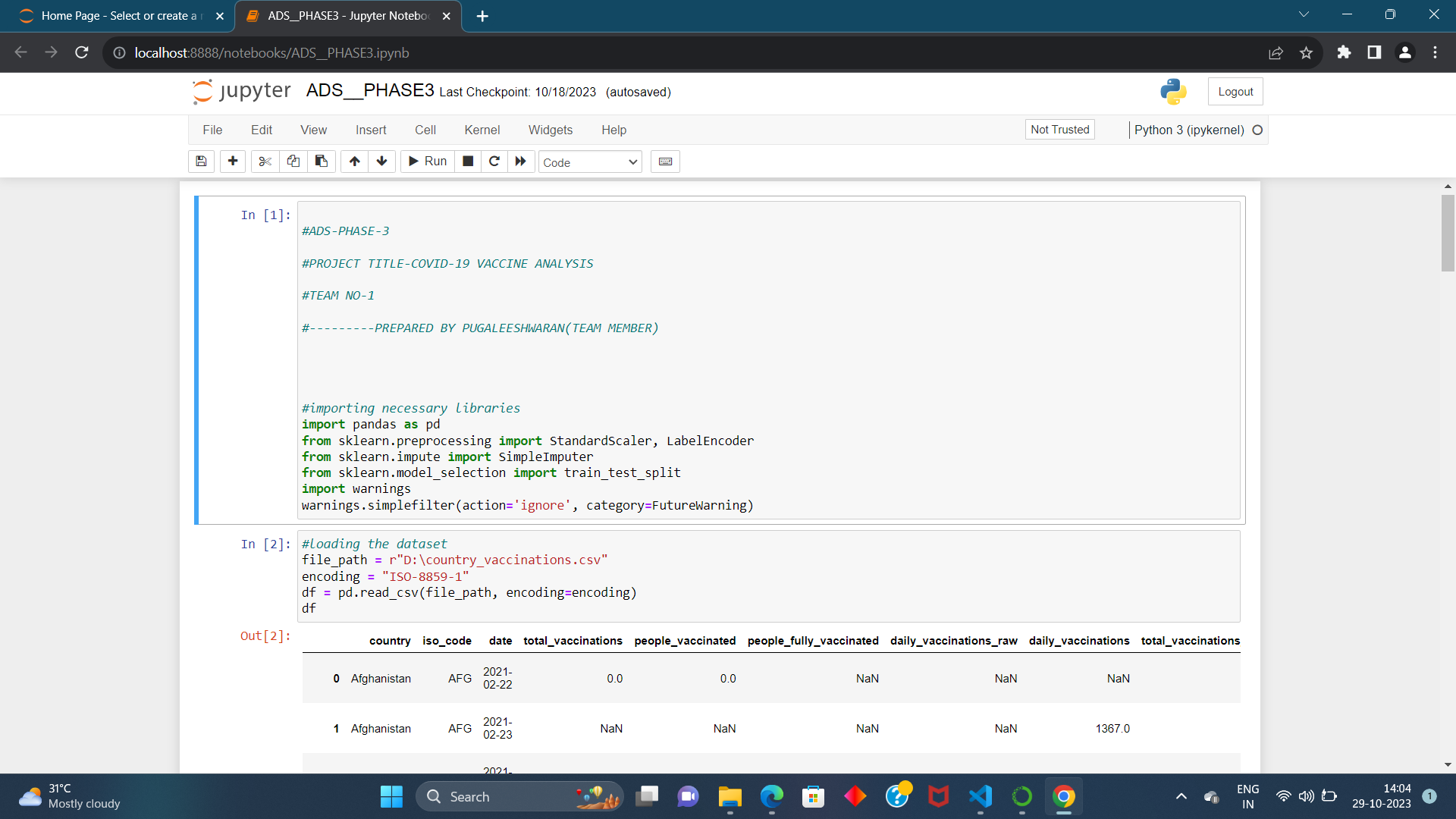
VISUALIZATION:

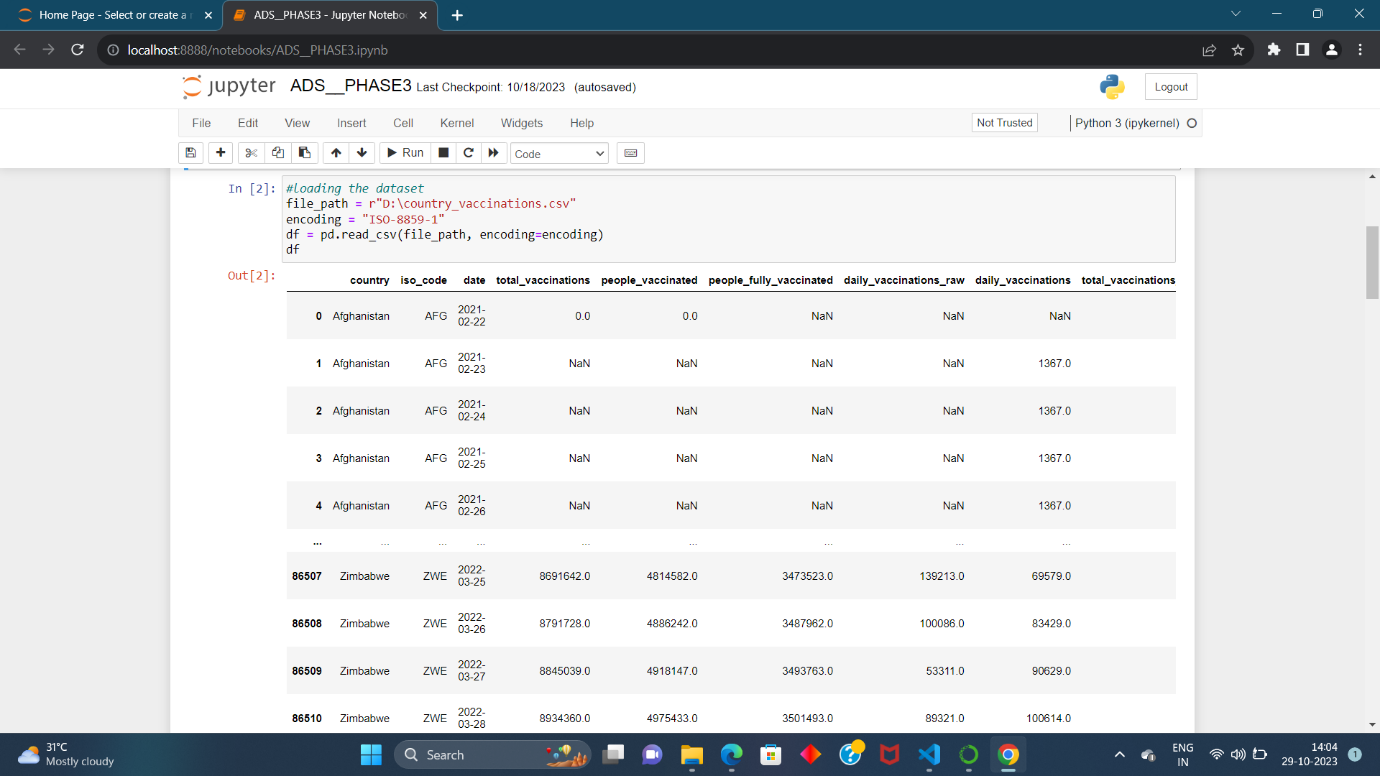
Visualization in Python refers to the creation of graphical representations of data to help understand and communicate information effectively. Data visualization is a crucial aspect of data analysis, storytelling, and reporting, and Python offers several powerful libraries for creating various types of visualizations. The most commonly used libraries for data visualization in Python include **matplotlib**, **seaborn**, **plotly**, and **bokeh**.

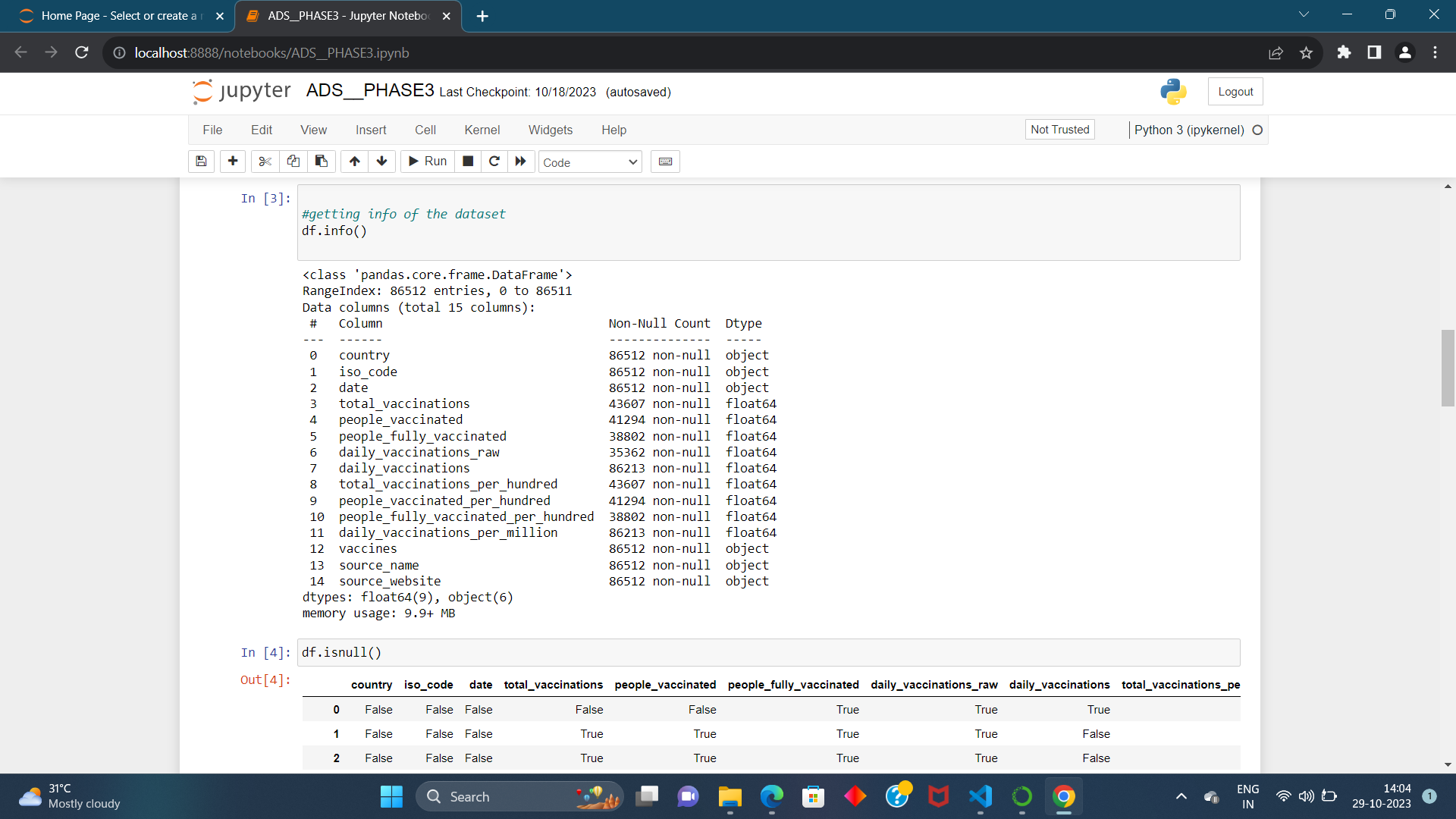
**EXPLANATION FOR DEVELOPMENT PART-1**

**STEP1:** IMPORTING THE REQUIRED LIBRARIES AND LOADING THE DATASET

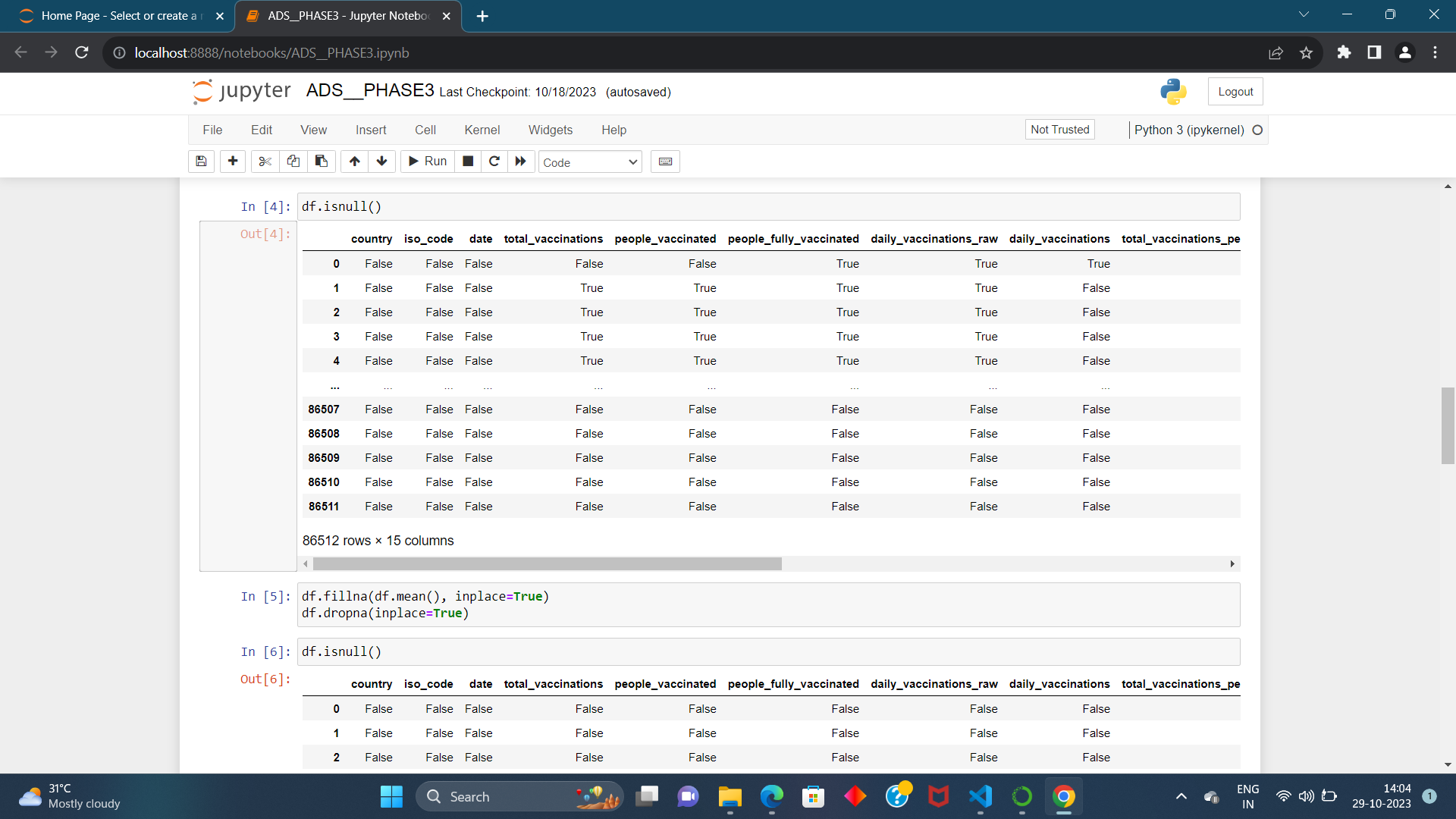
****

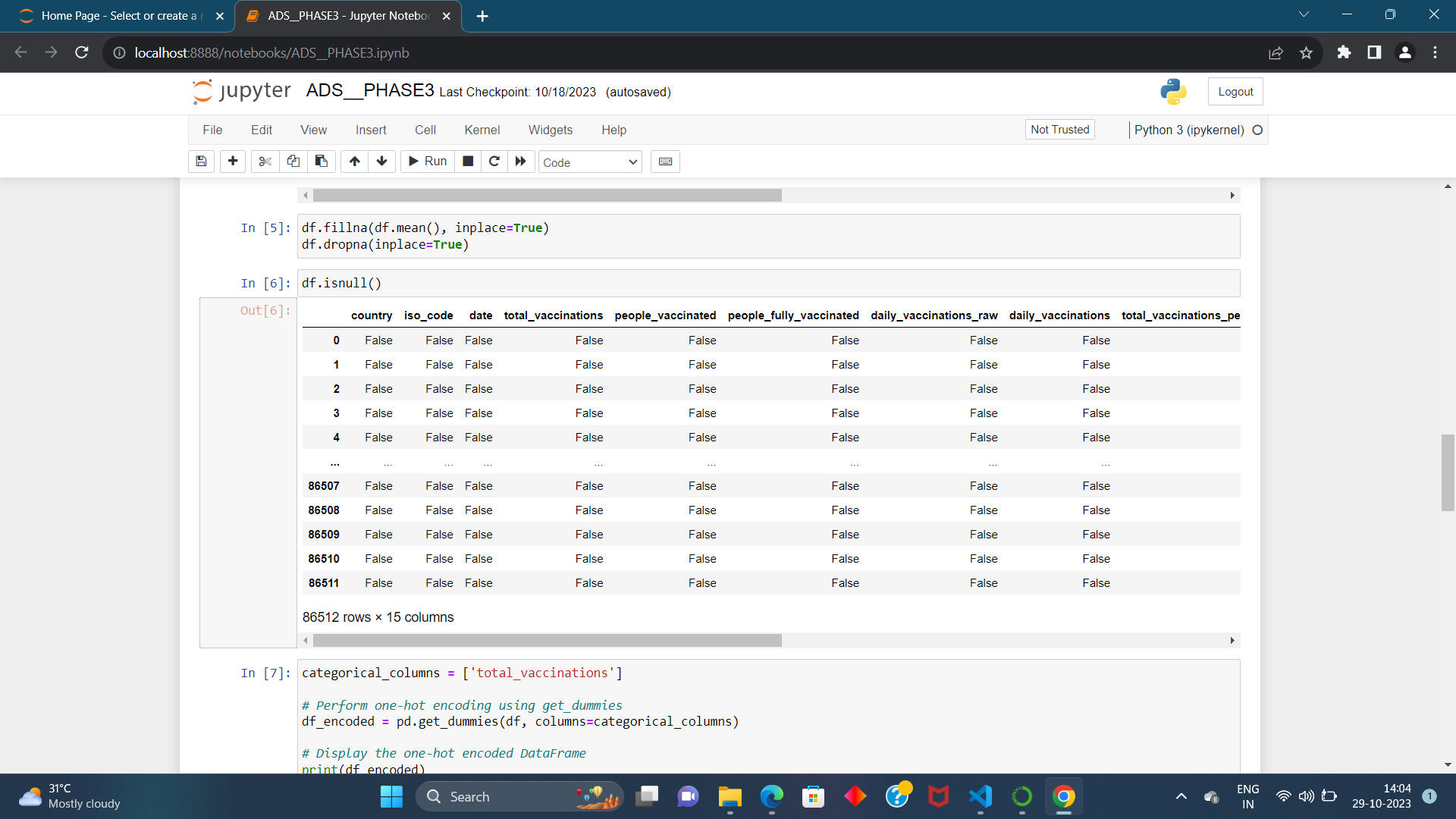
****

**STEP2: GETTING THE INFO**

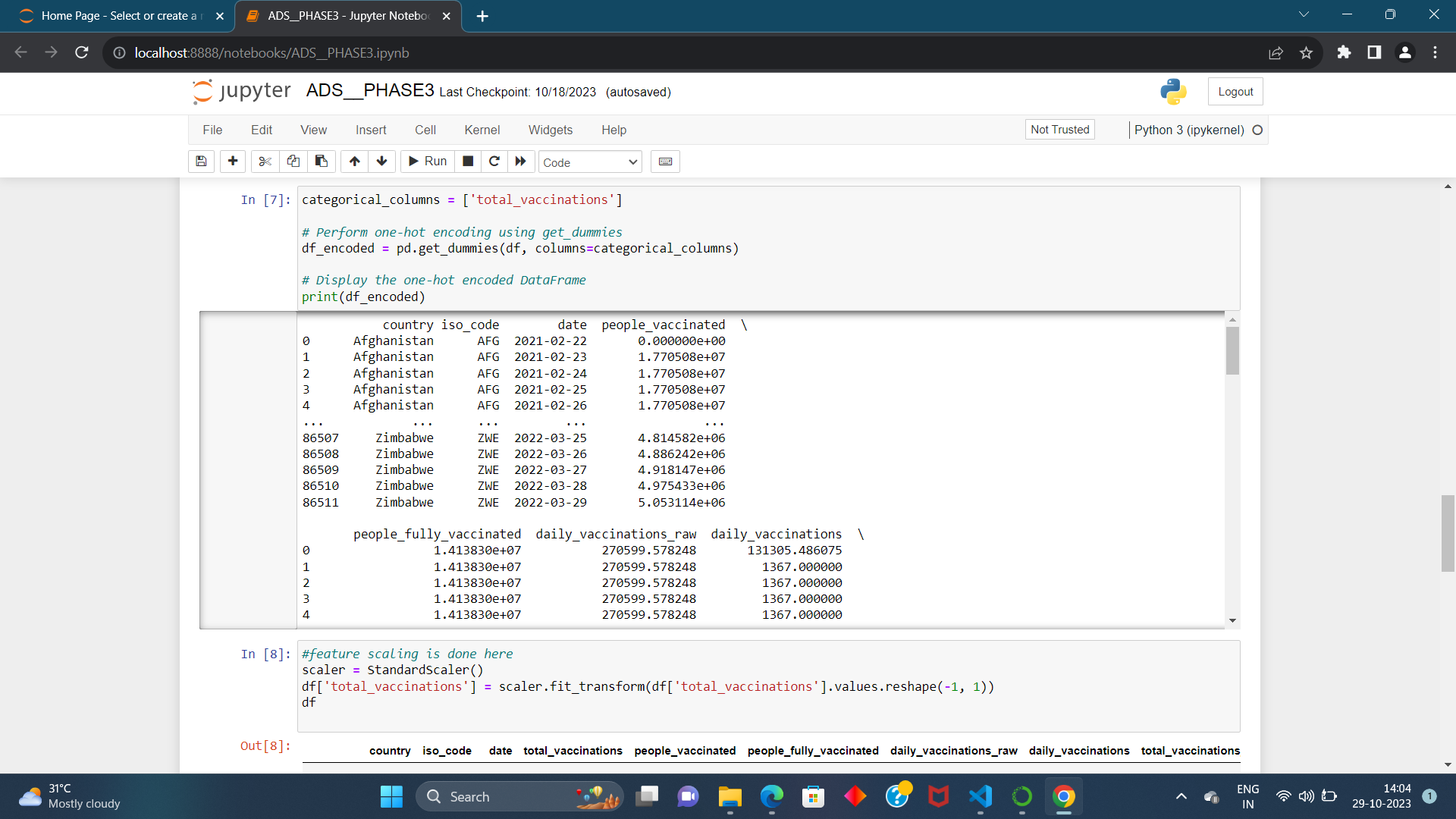
****

**STEP3: HANDLING THE MISSING DATA**

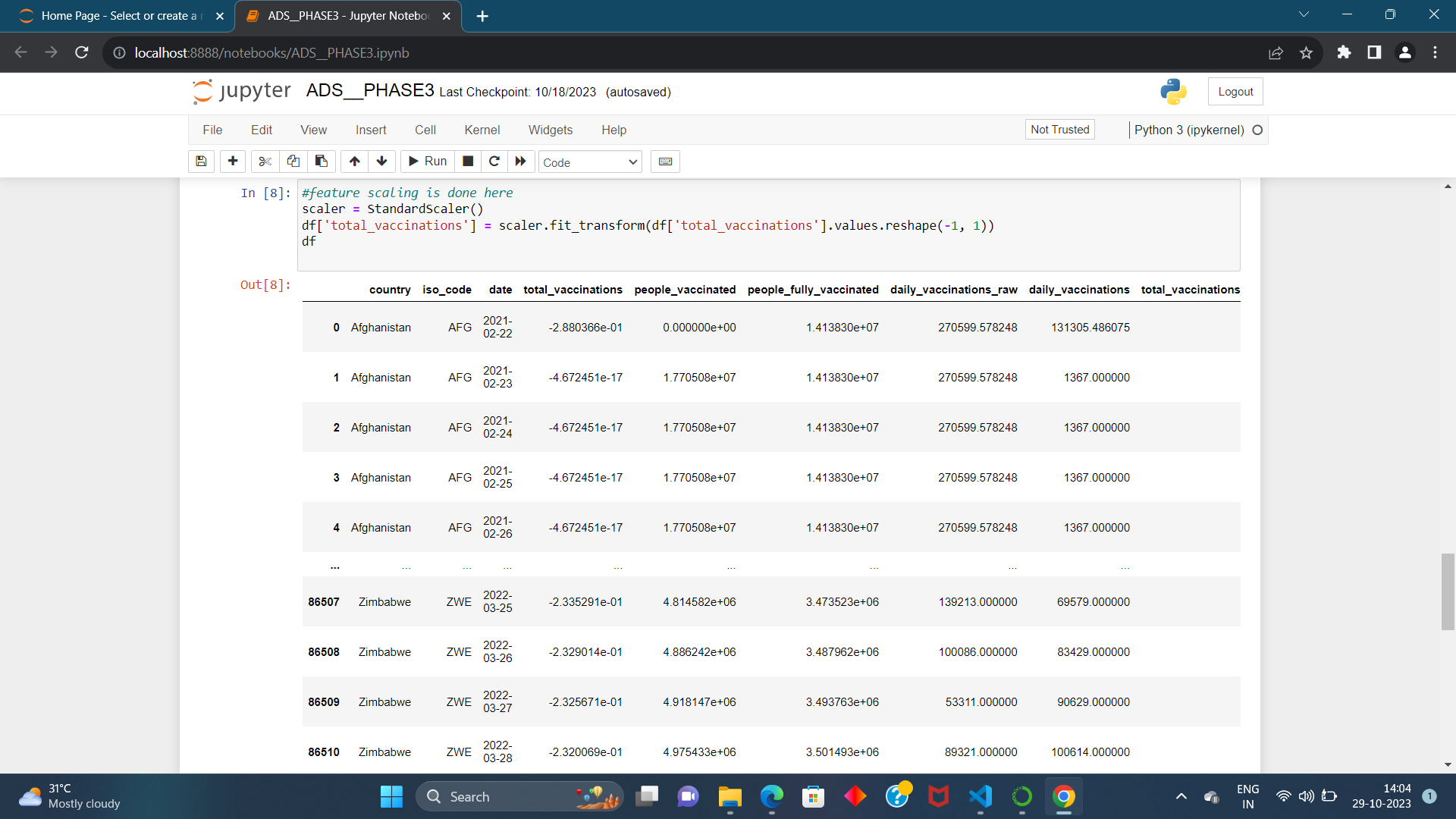
****

****

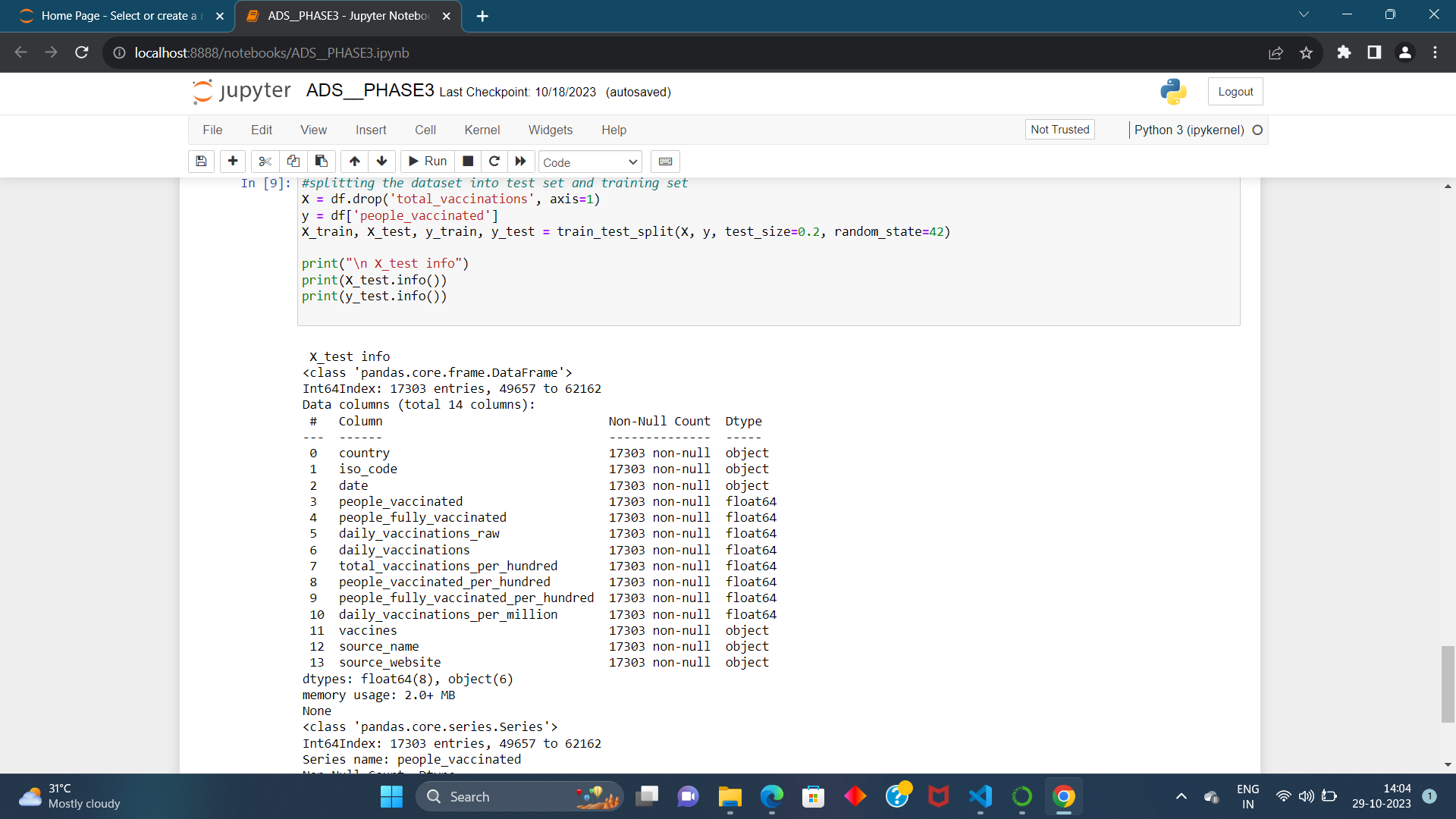
**STEP4: ONE HOT ENCODING**

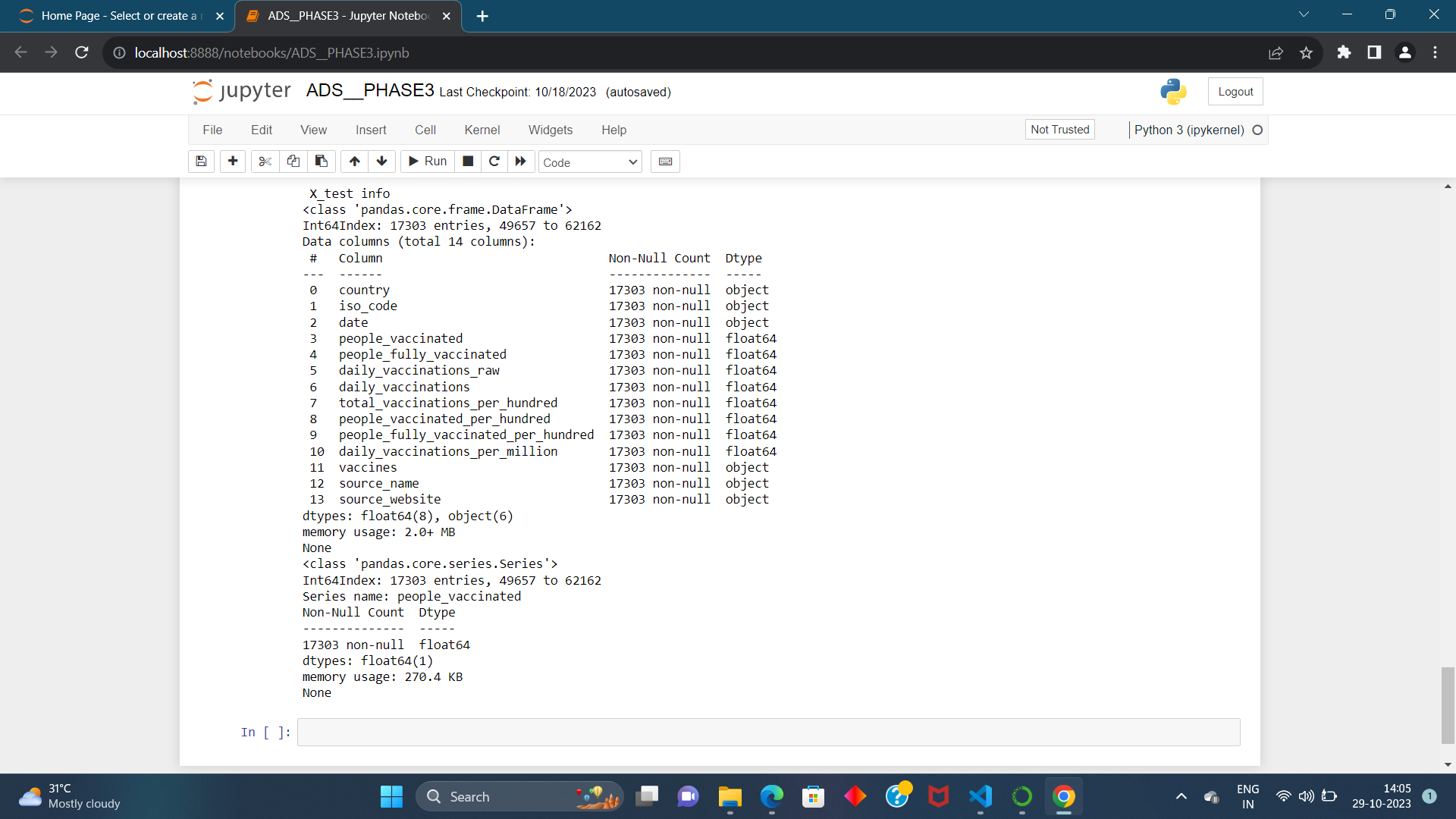
****

**STEP5: FEATURE SCALING USING STANDARD SCALER**

****

**STEP6: SPLITTING THE DATASET INTO TEST SET ANDD TRAINING SET**

****

****

**Feature Scaling, Model Training, and Evaluation Algorithm for Covid-19 Vaccine Analysis:**

This algorithm aims to guide the development of a predictive model for covid-19 vaccine analysis using the provided dataset. It covers essential steps, including feature engineering, model training, and evaluation, to ensure accurate predictions.

Steps: 1. Load and Preprocess the covid-19 vaccine analysis dataset:

• Load the dataset, which includes information on name of the country, total vaccinations, people who vaccinated, date etc…

• Ensure that you understand the dataset's structure and contents.

2. Feature Engineering:

• Review the dataset to identify which features will be used for analysing covid-19 vaccines. In this case, "Country," "Total Vaccinations" and "People Vaccinated" are potential features.

• Handle any missing data. It appears that the dataset does not have any missing values.

• Encode categorical data, such as "Total Vaccinations" using techniques like label encoding or onehot encoding to convert them into a numerical format.

3. Feature Scaling :

• Analys the dataset and determine if feature scaling is required. Some machine learning algorithms benefit from scaled features.

• If needed, apply feature scaling to numerical features. For example, you can use standardization to scale the "Total Vaccinations" feature.

4. Split the Dataset:

• Split the dataset into training and testing sets to assess the model's performance.

• A common split ratio is 80% for training and 20% for testing. Ensure that the split is random to avoid any potential biases.

5. Select a Machine Learning Model:

• -Choose an appropriate machine learning model for regression tasks.

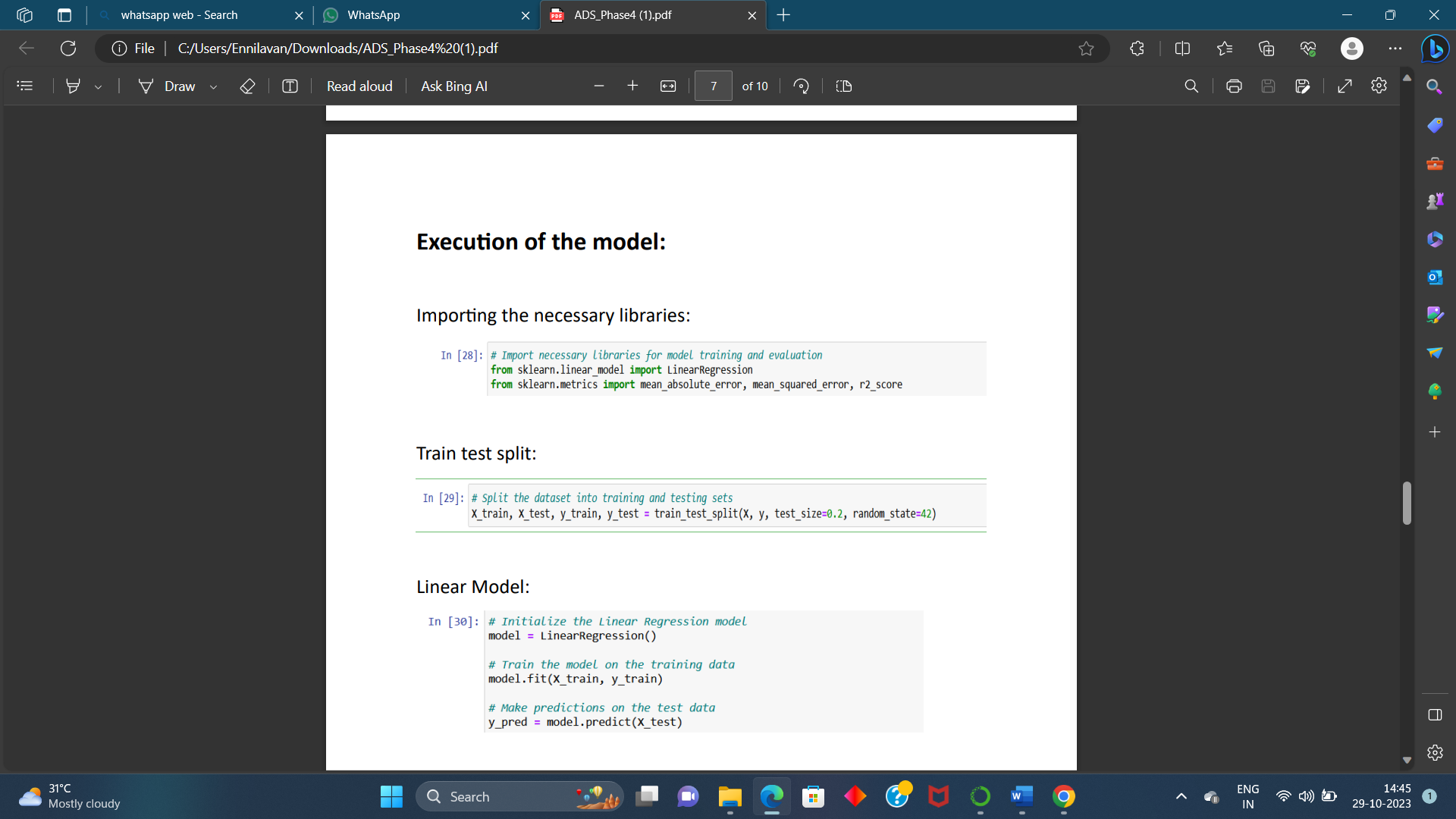
6. Train the Model:

• Initialize the chosen model.

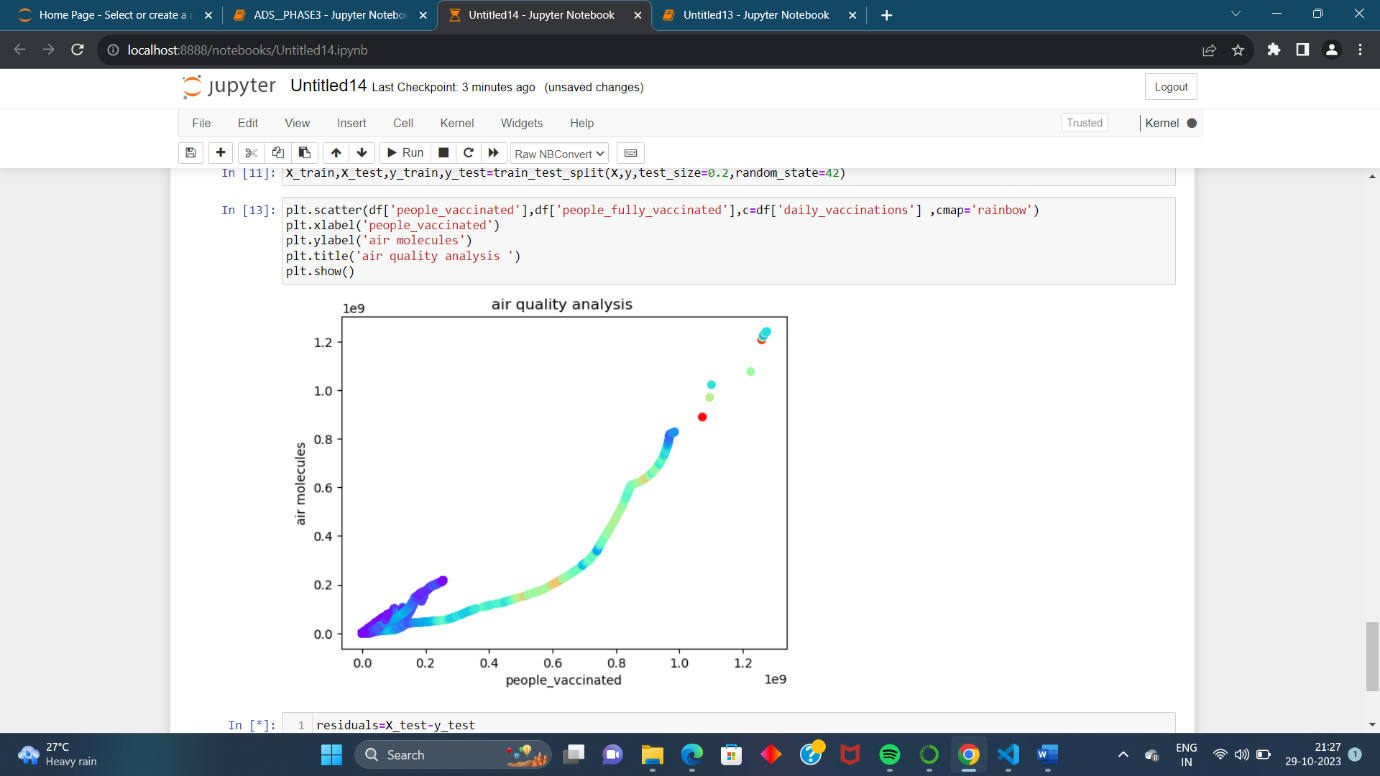
• Fit the model to the training data, using the selected features.

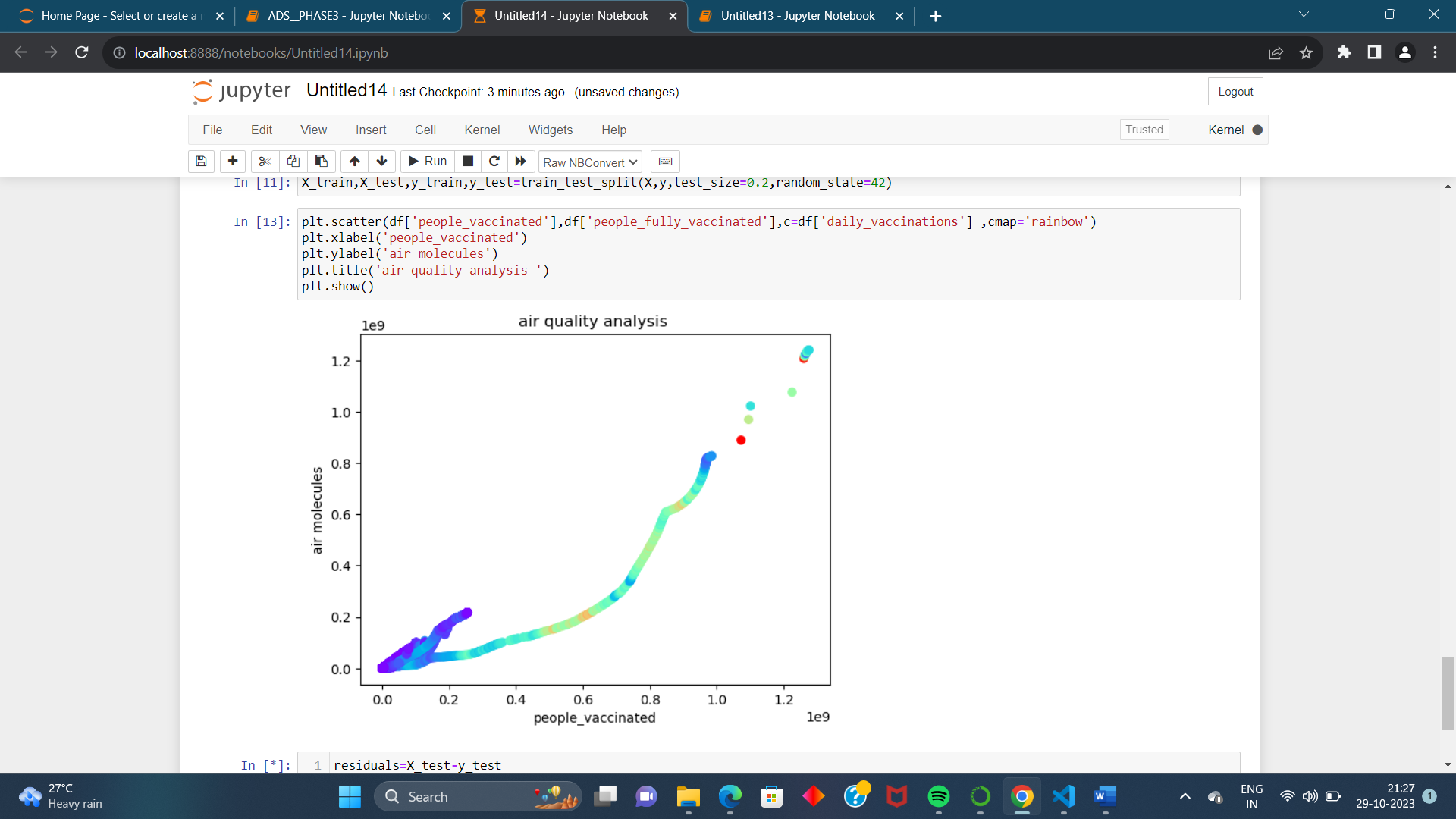
• During training, the model will learn patterns in the data.

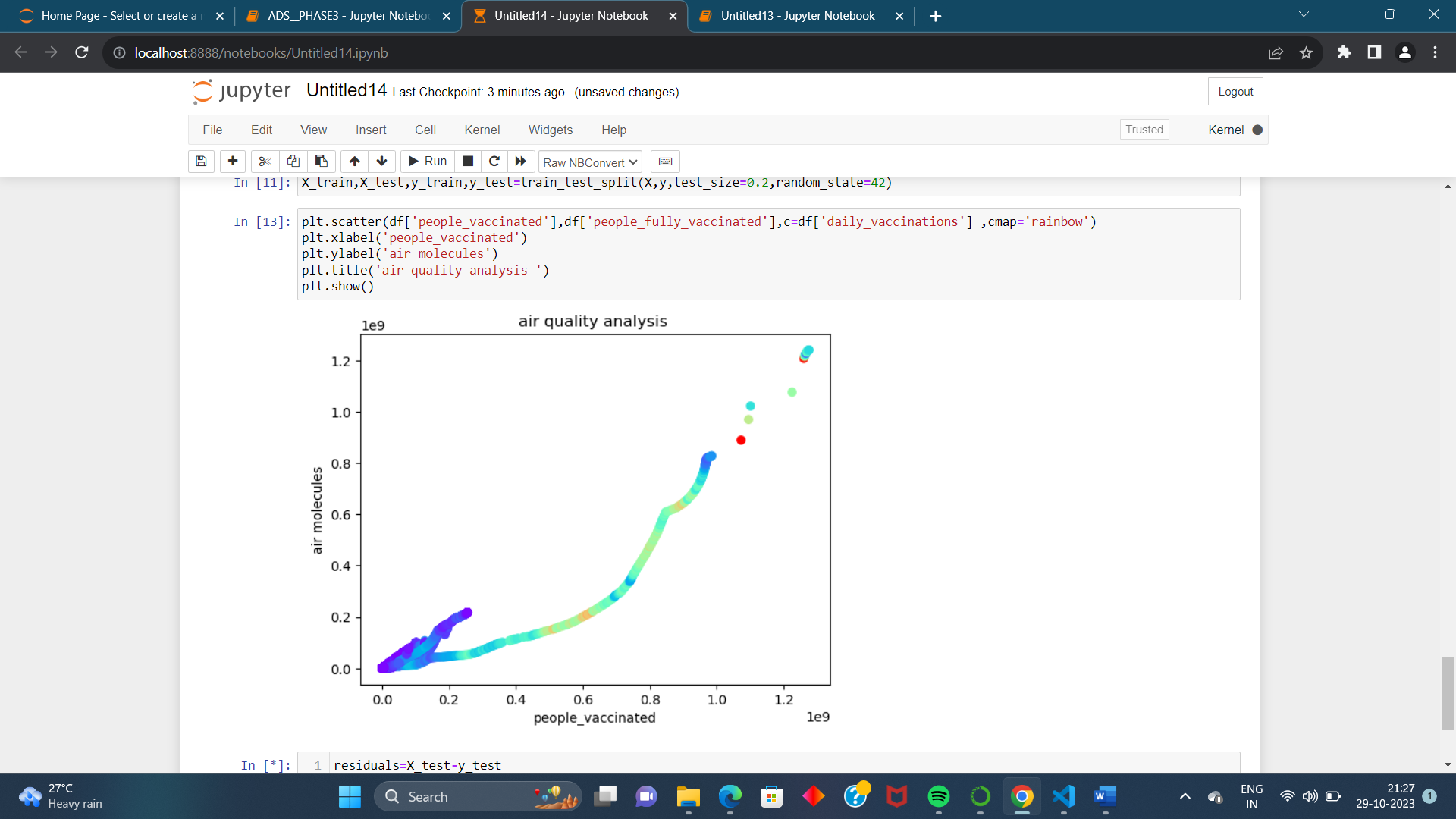
**EXECUTION OF THE MODEL:**

****

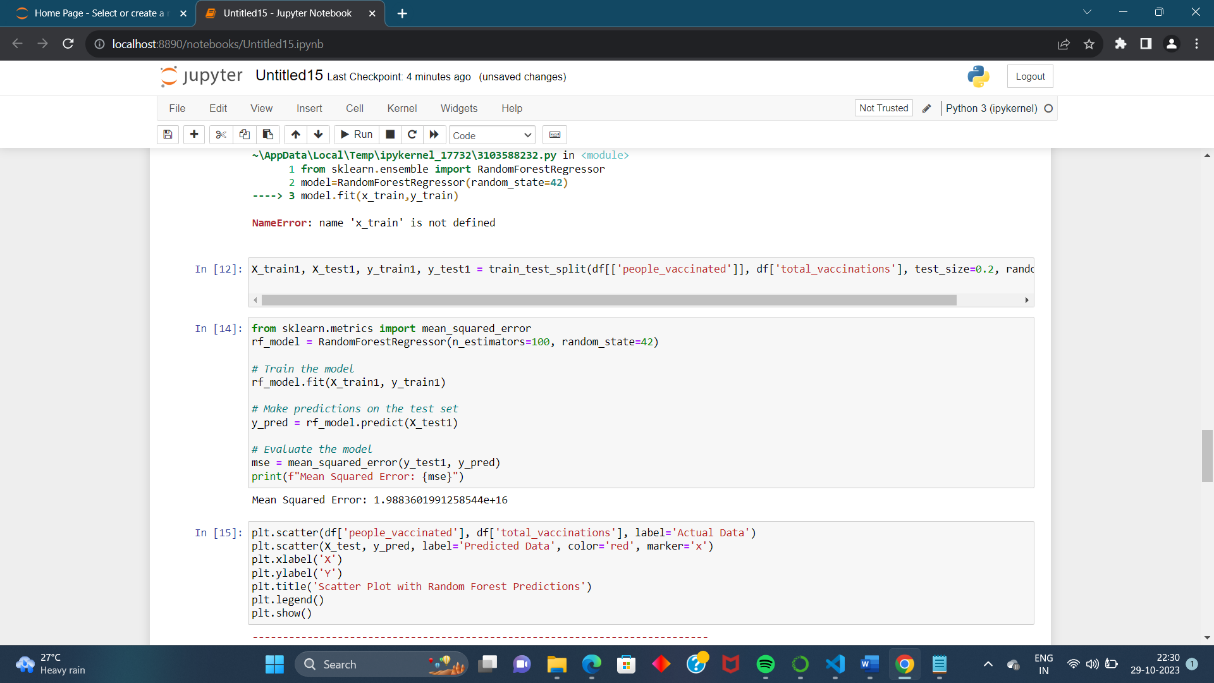
**DOING TESTING FOR SCATTER ANALYSIS:**

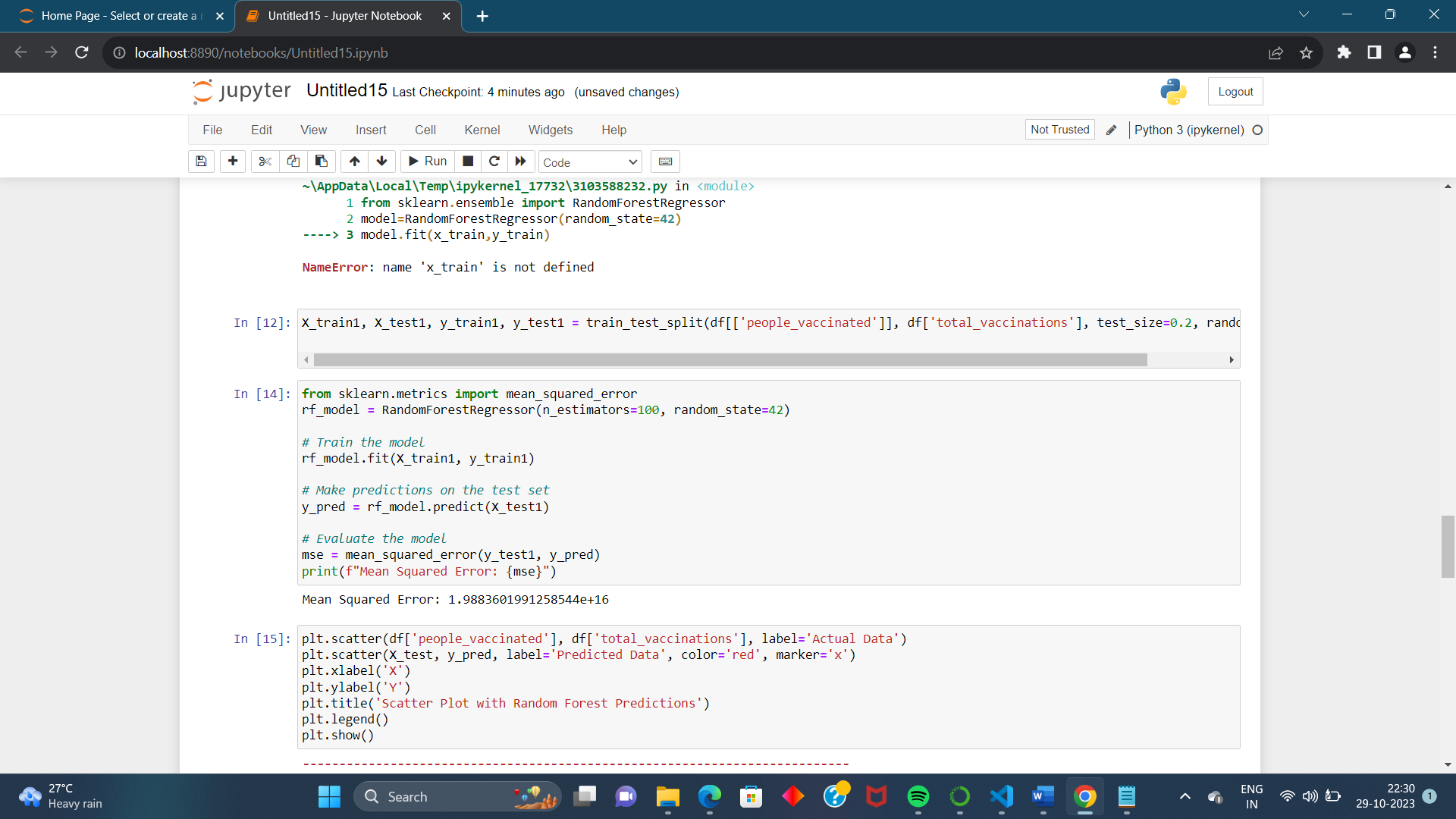
****

****

****

**RANDOM FOREST:**

****

****

**A scatter plot is a type of data visualization used to display individual data points in a two-dimensional space. Each point on the scatter plot represents the values of two variables, making it a useful tool for visualizing the relationship between these variables. Scatter plots are often used to identify patterns, trends, and outliers in the data.**

**Data Preparation:**

**First, you need to prepare your data. Ensure you have a dataset with at least two numerical variables that you want to explore using a scatter plot and predict using a Random Forest model.**

**Data Visualization with Scatter Plots:**

**You can use Python libraries like Matplotlib or Seaborn to create scatter plots. For example, if you have a dataset df and you want to create a scatter plot between two variables X and Y**

**Random Forest Model:**

**To build a Random Forest model in Python, you can use popular machine learning libraries like scikit-learn.**

**To visualize the predictions of your Random Forest model on the scatter plot, you can use the matplotlib library to overlay the model's predictions on the original data points.**

THANKYOU