**COVID-19 VACCINE ANALYSIS**

**PHASE-4**

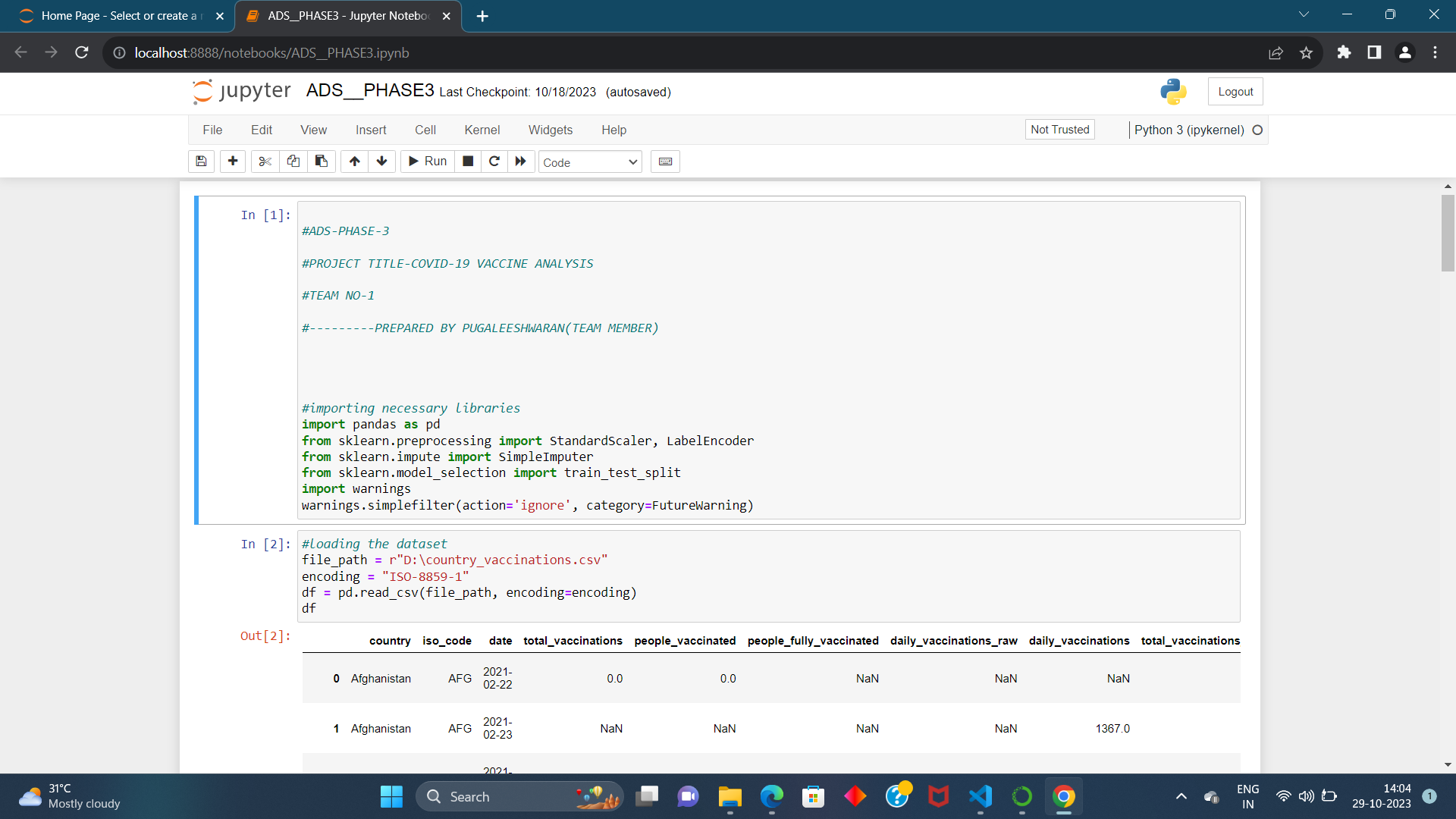
**DEVELOPMENT PART-2**

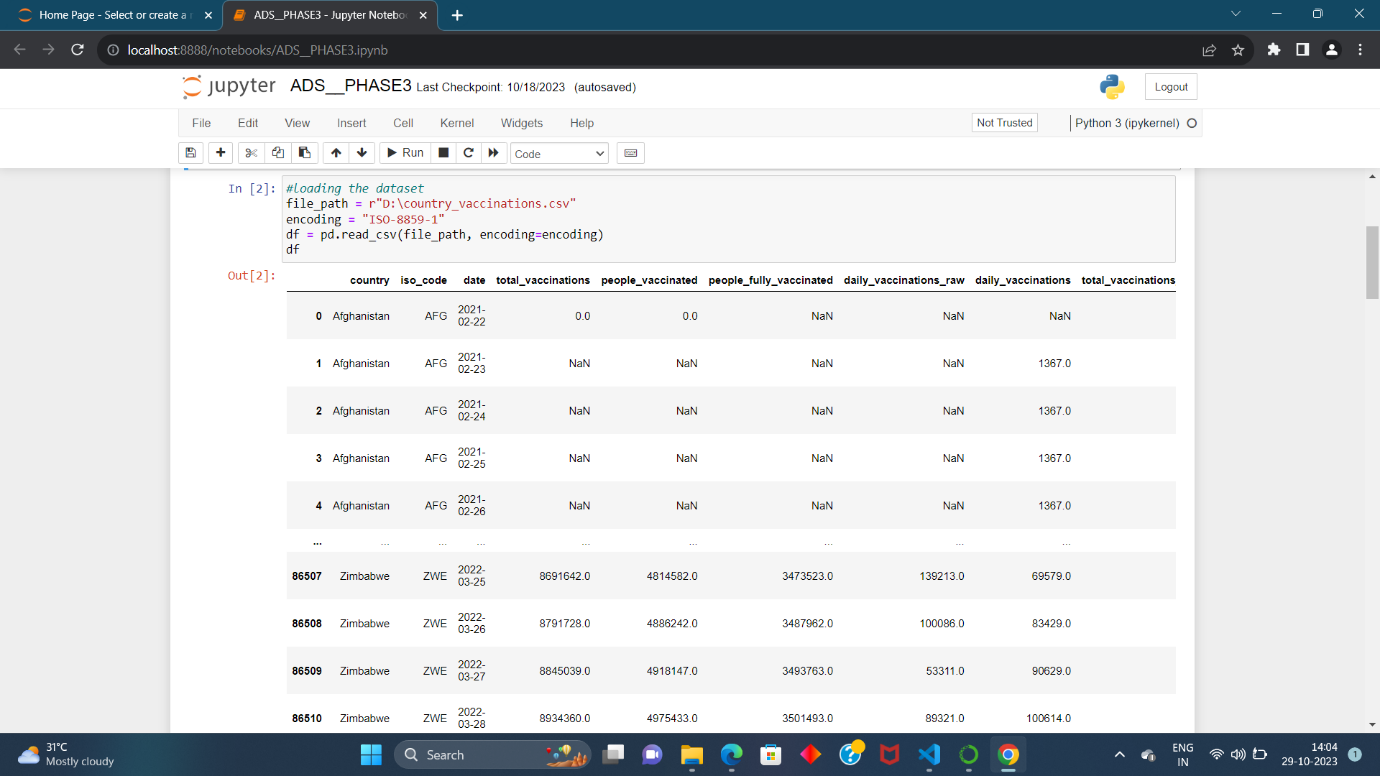
**ENNILAVAN**

**(TEAM MEMBER)**

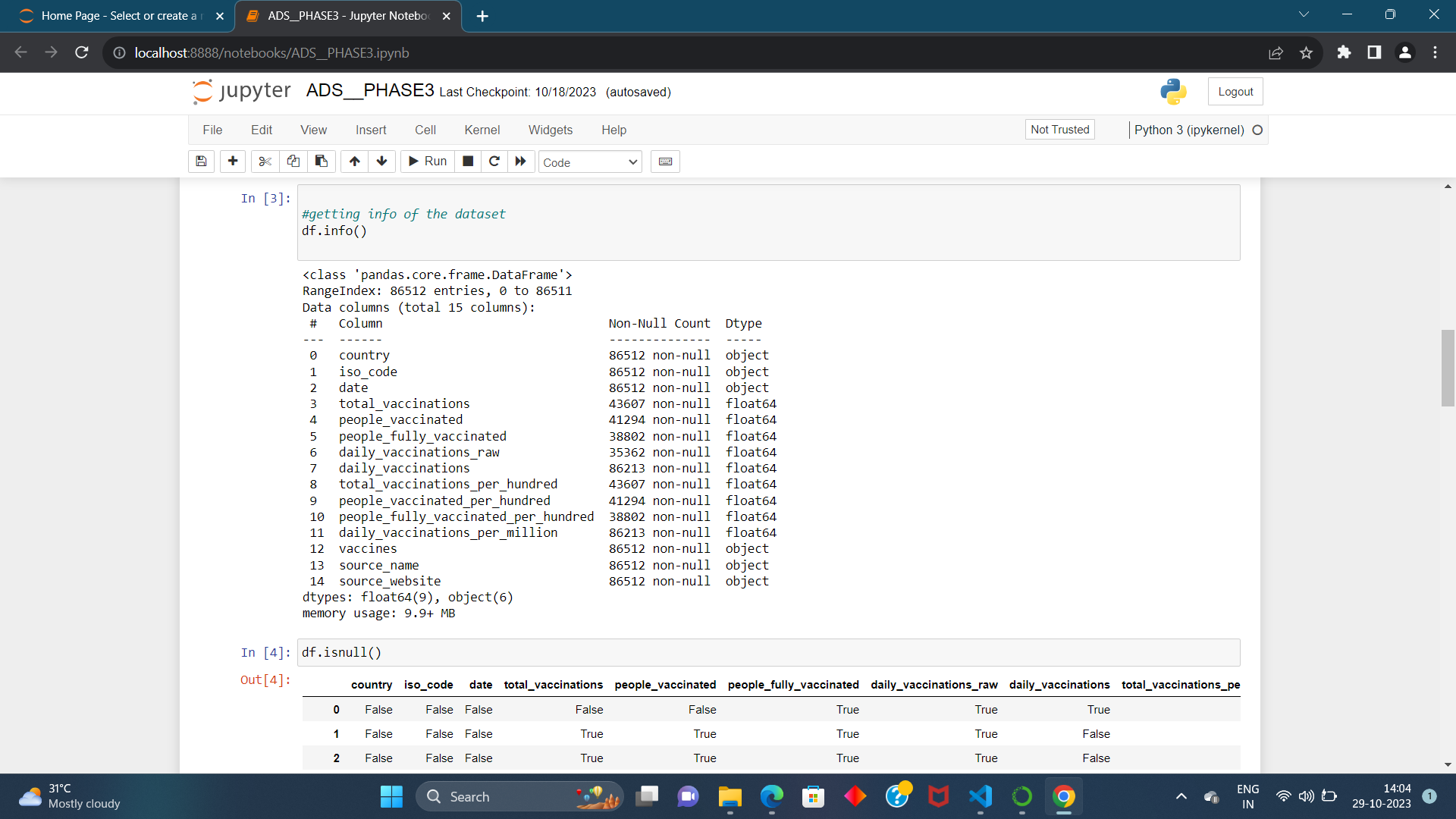
**EXPLANATION FOR DEVELOPMENT PART-1**

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

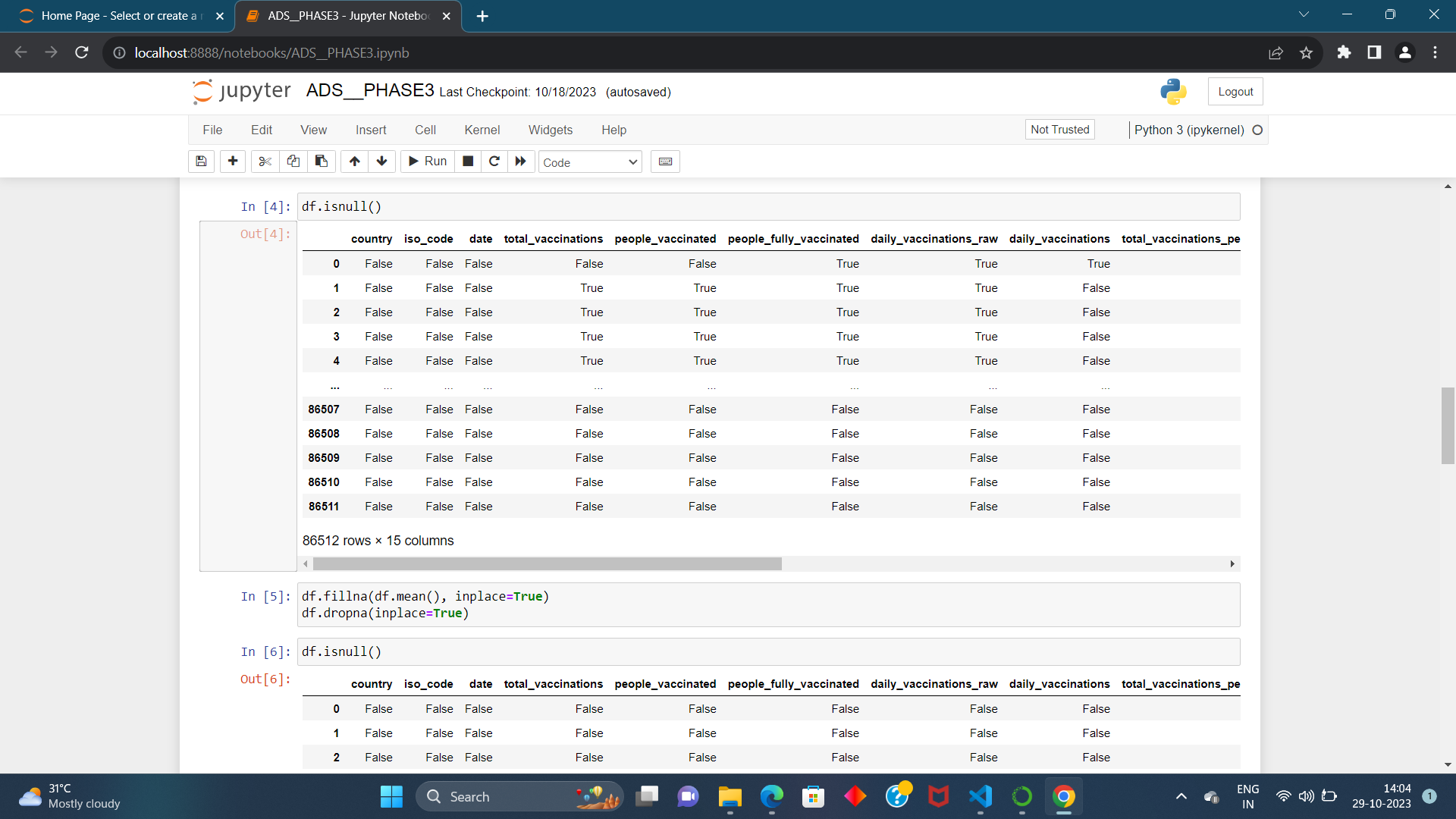
****

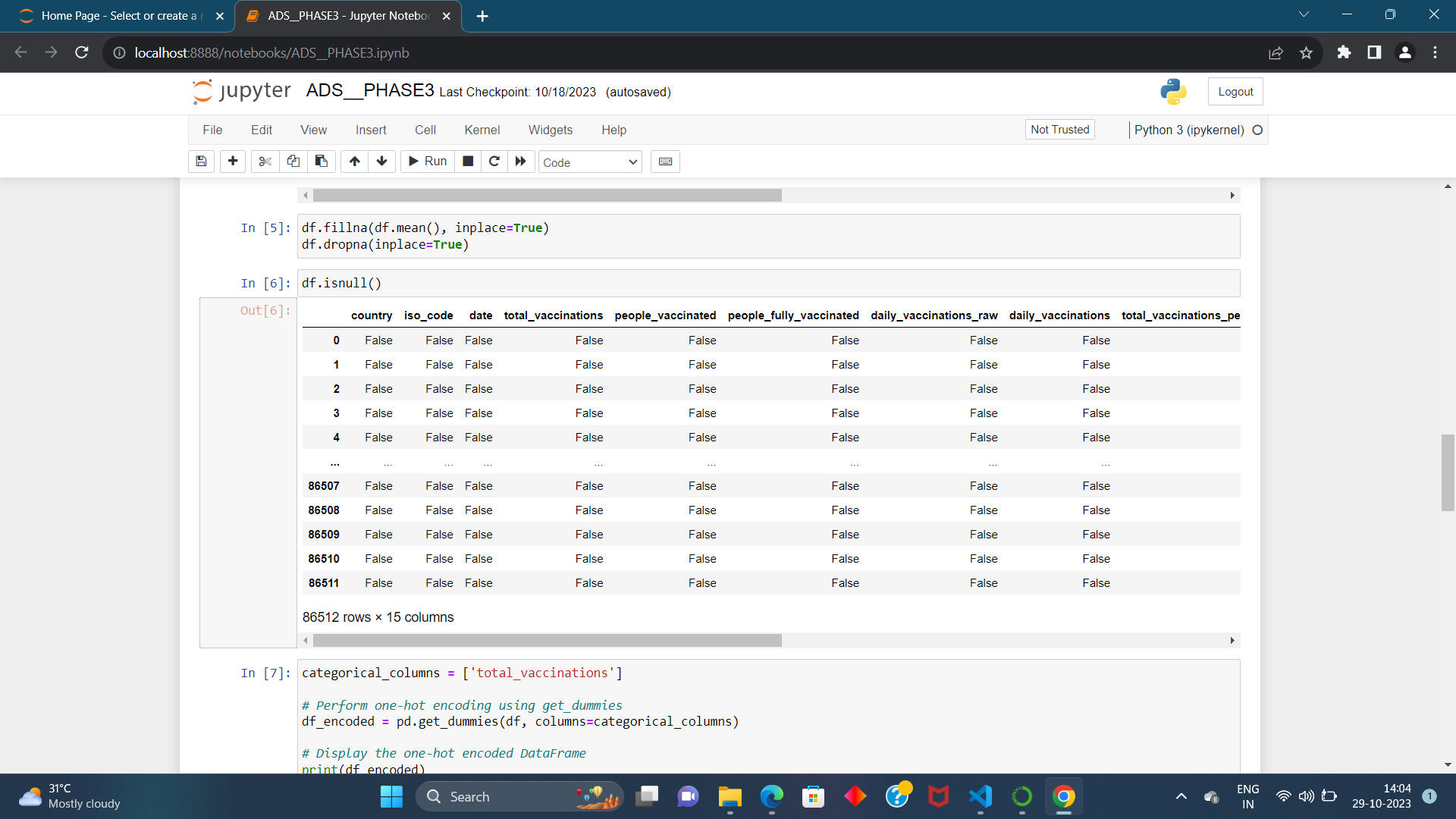
****

**STEP2: GETTING THE INFO**

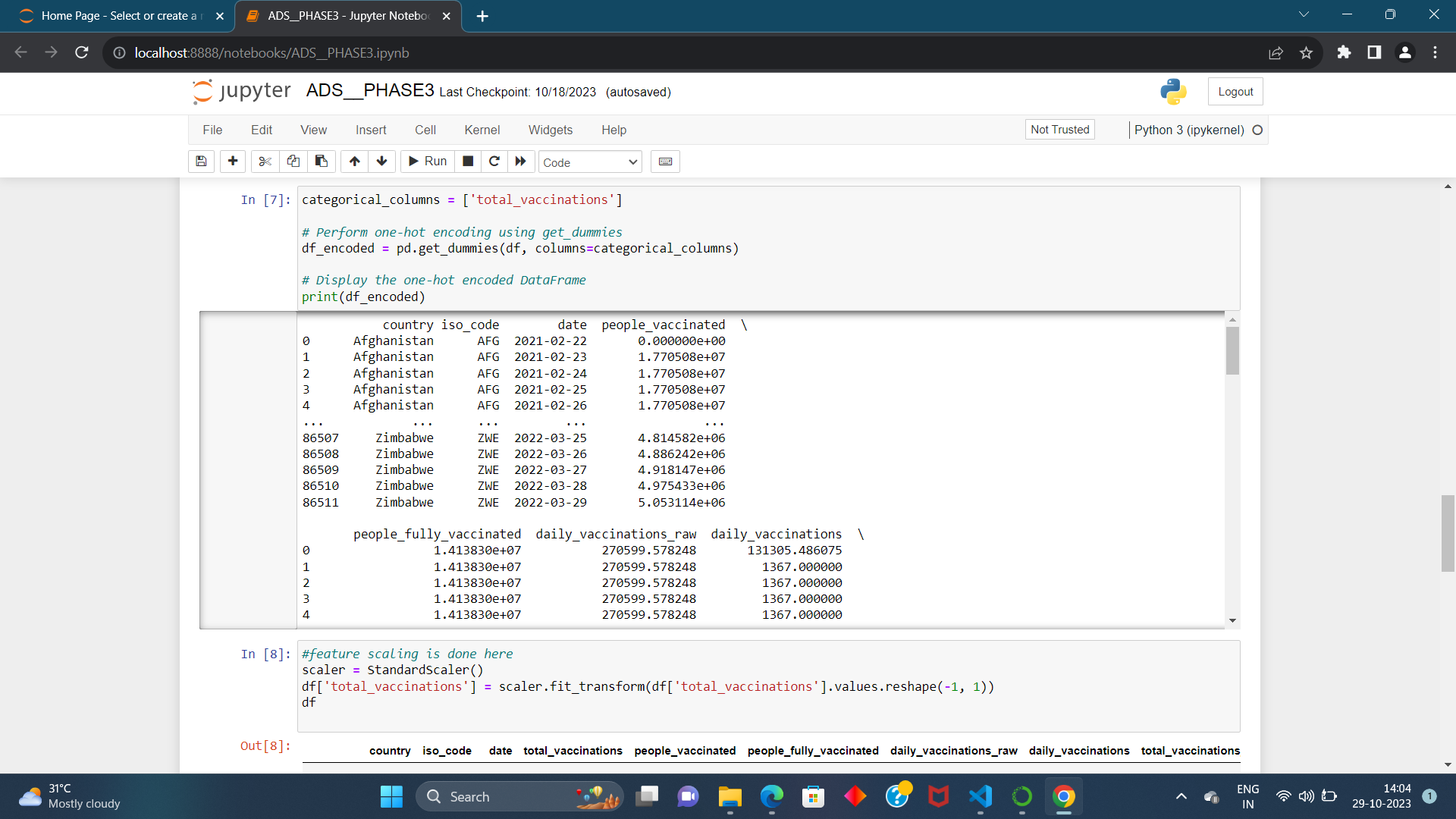
****

**STEP3: HANDLING THE MISSING DATA**

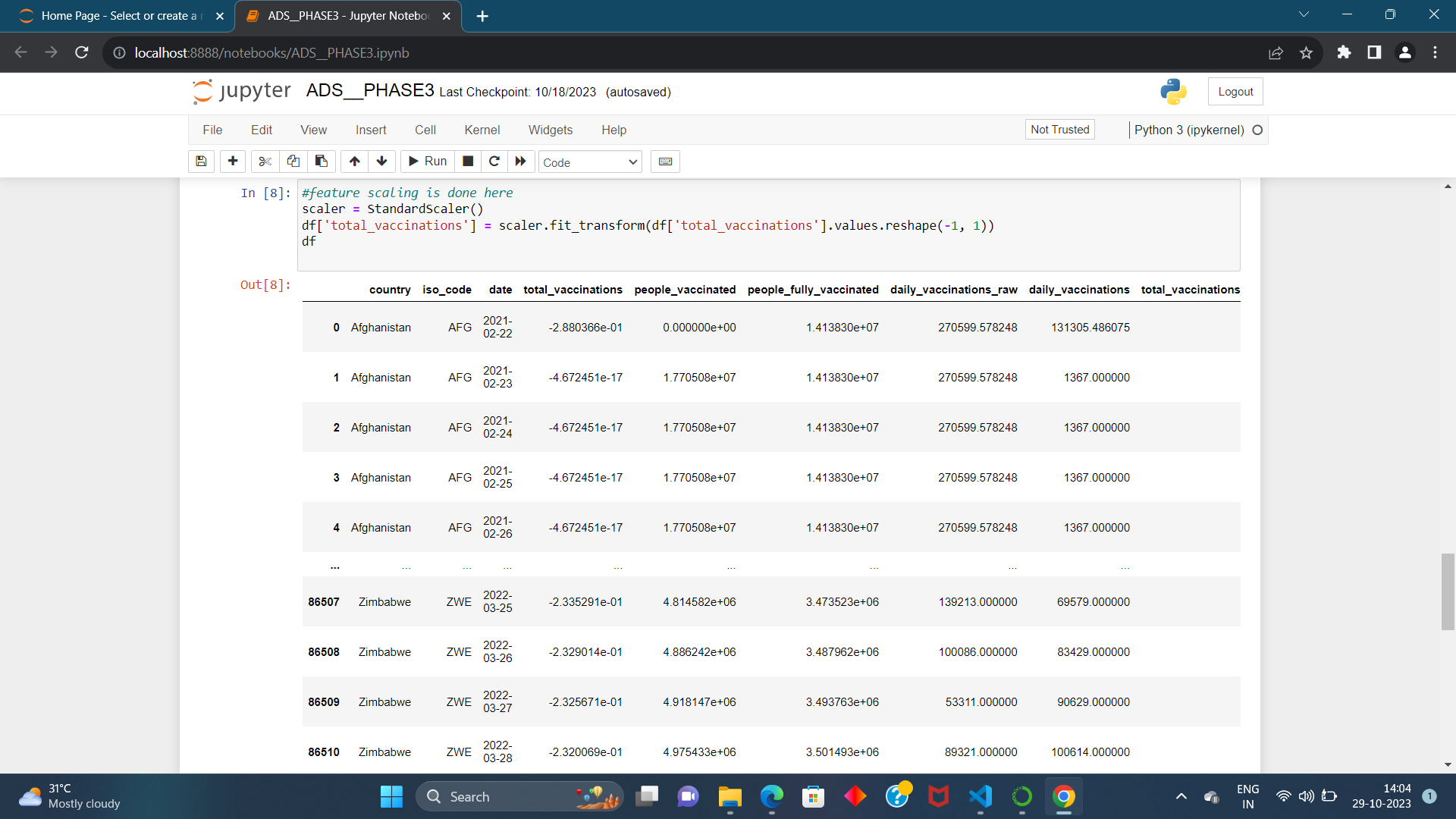
****

****

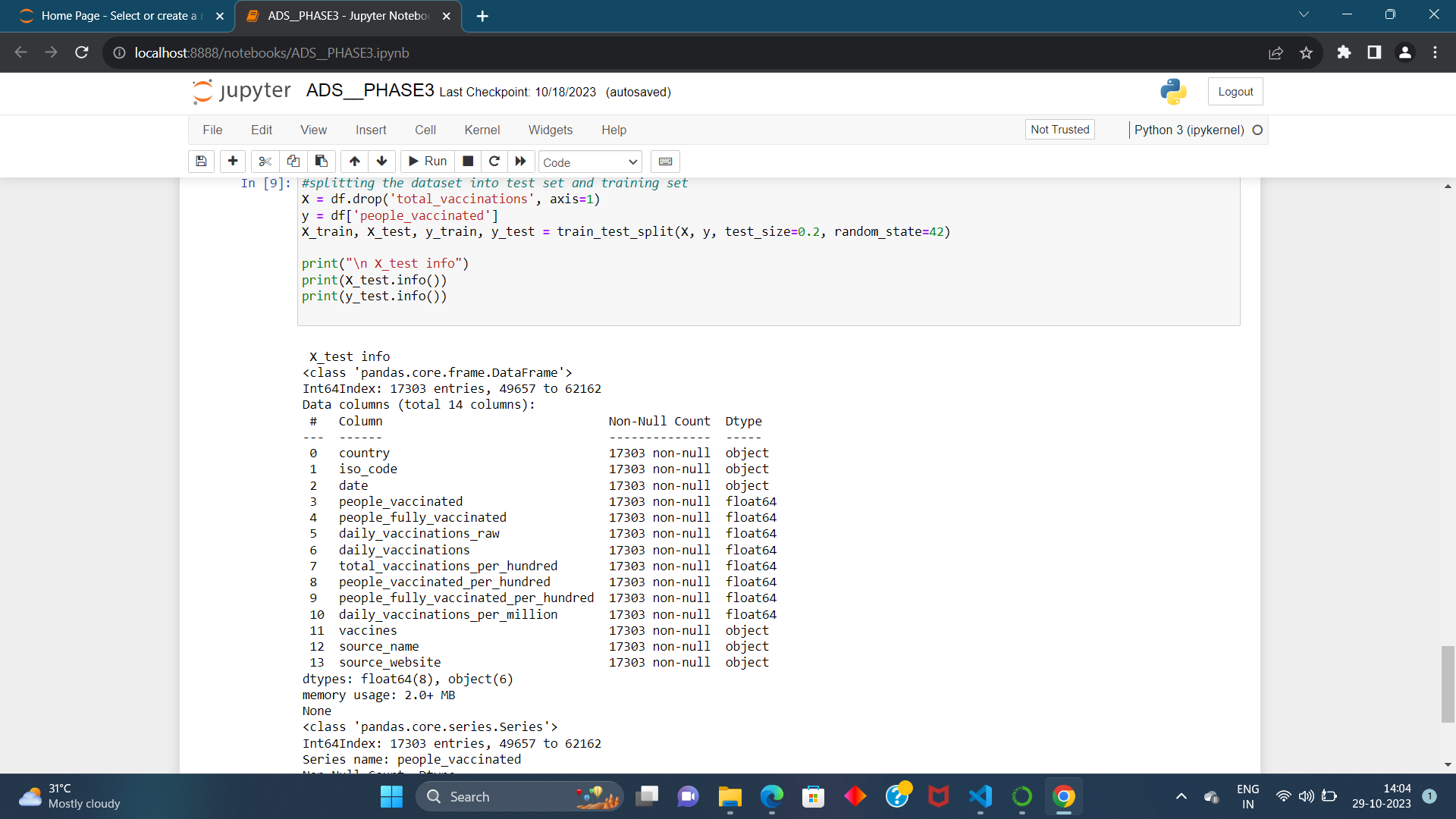
**STEP4: ONE HOT ENCODING**

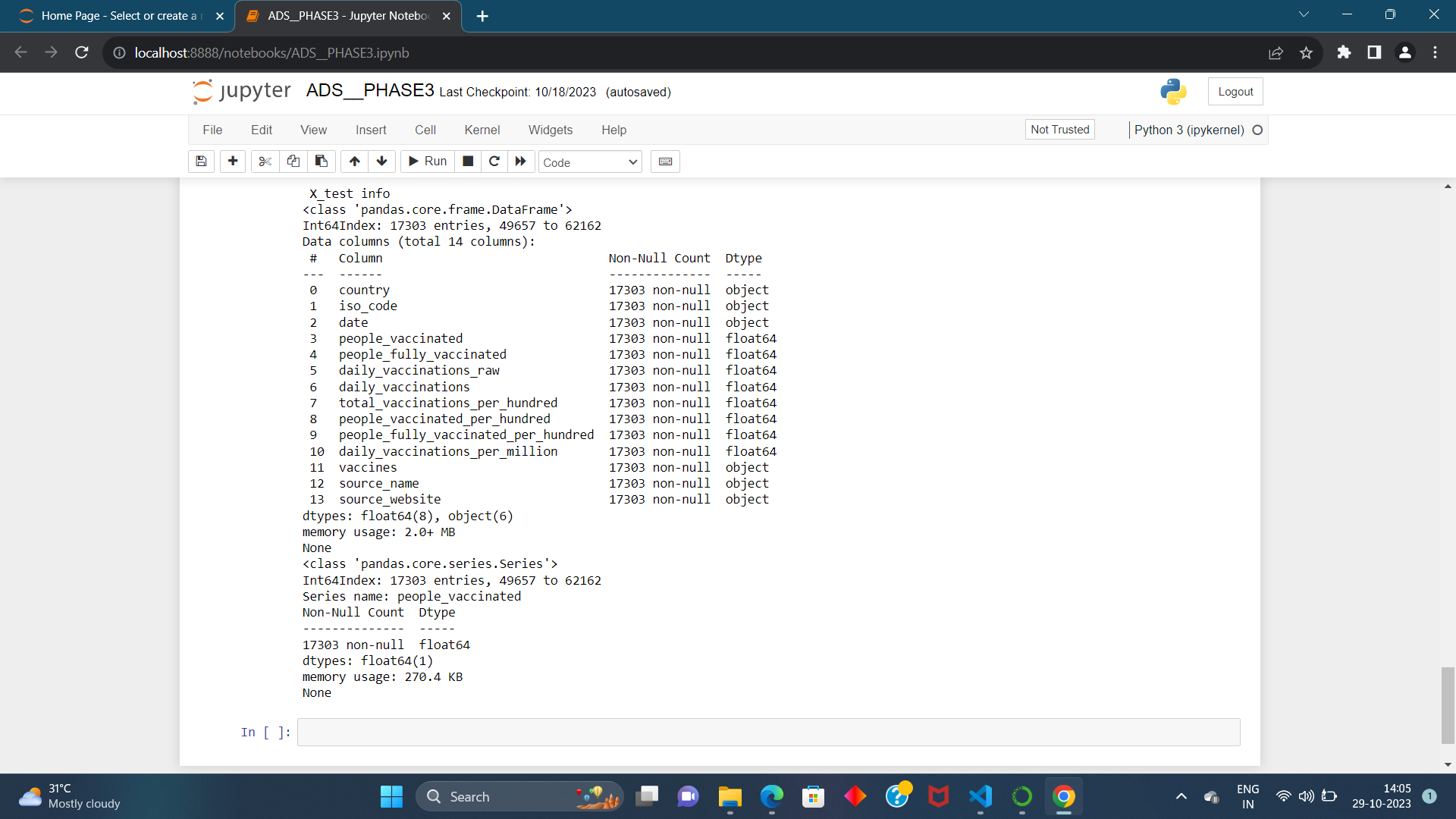
****

**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.

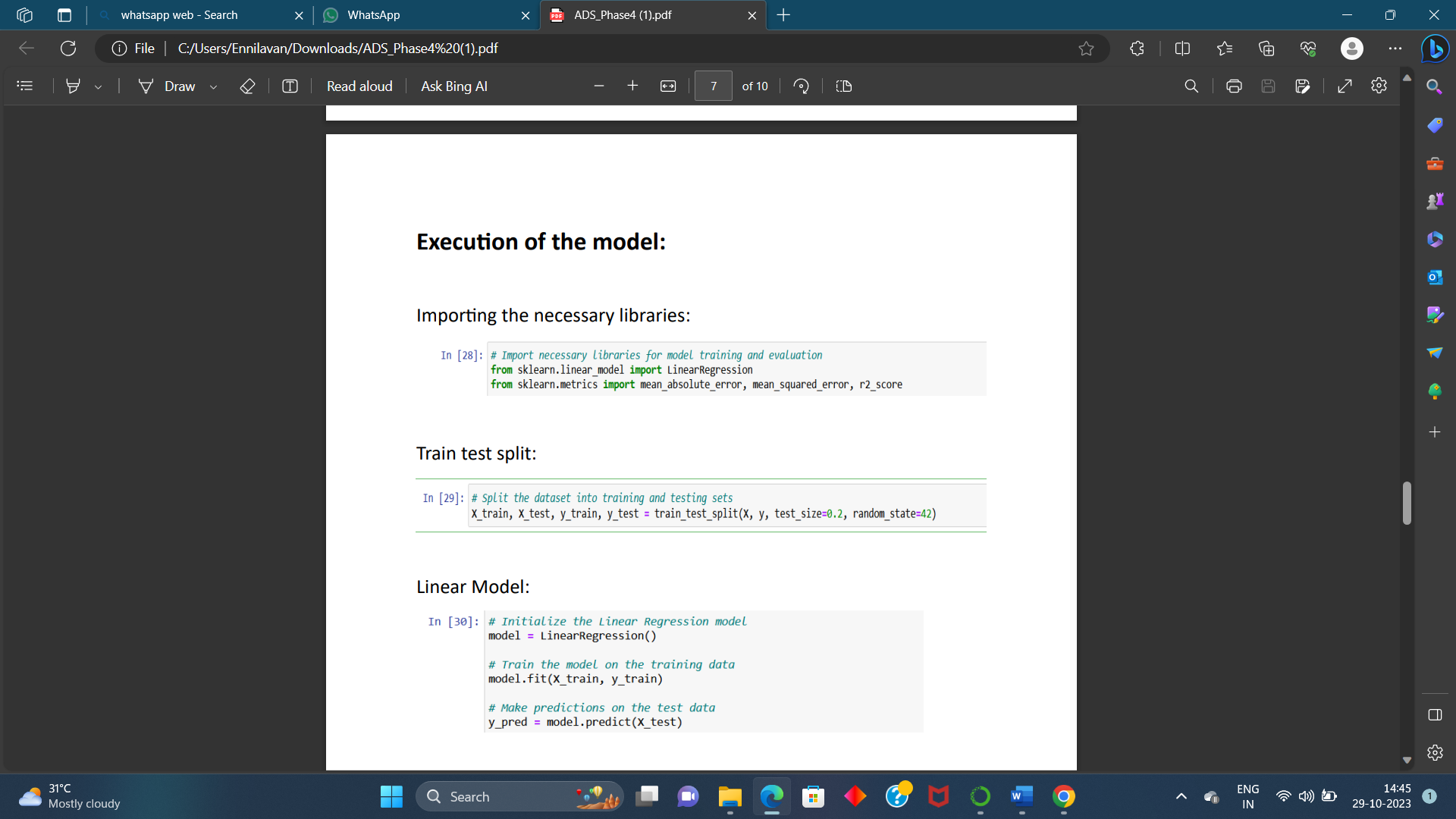
6. Train the Model:

• Initialize the chosen model.

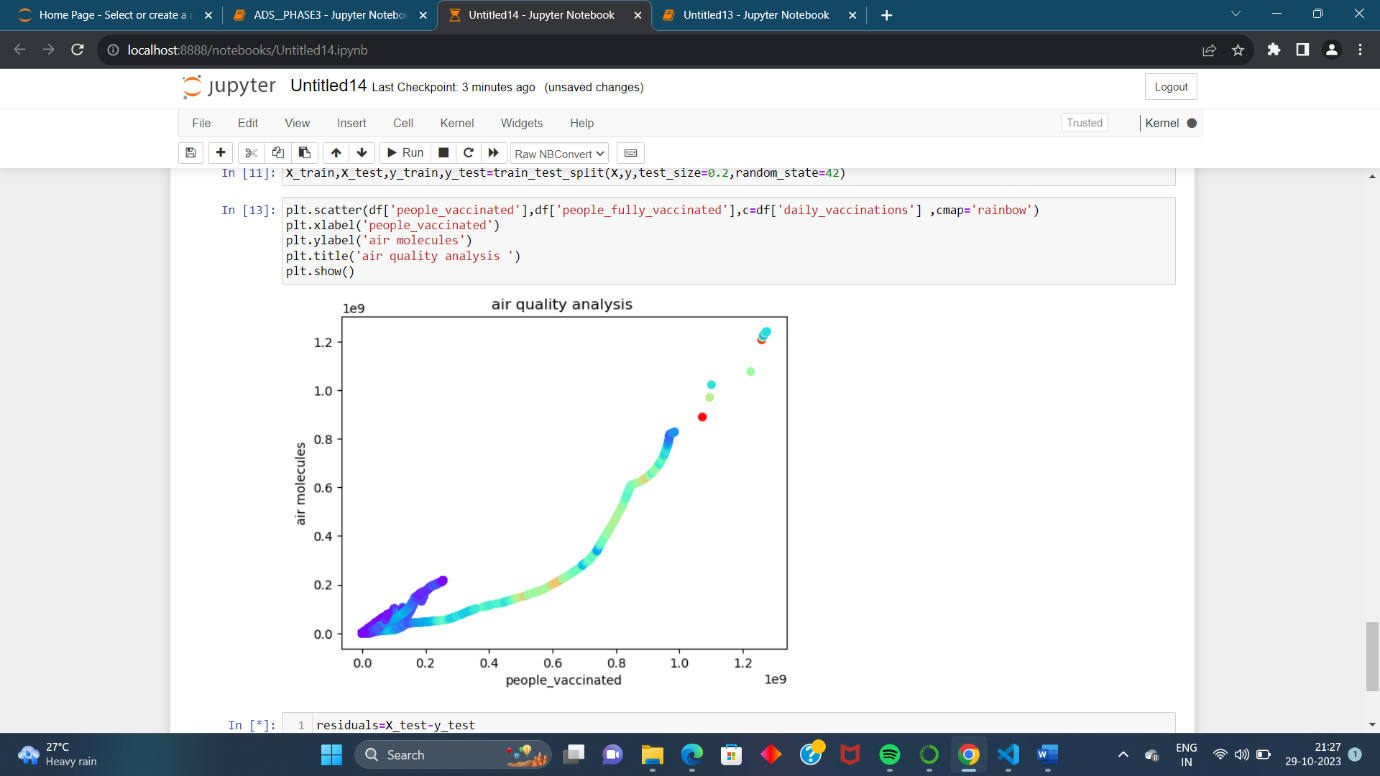
• Fit the model to the training data, using the selected features.

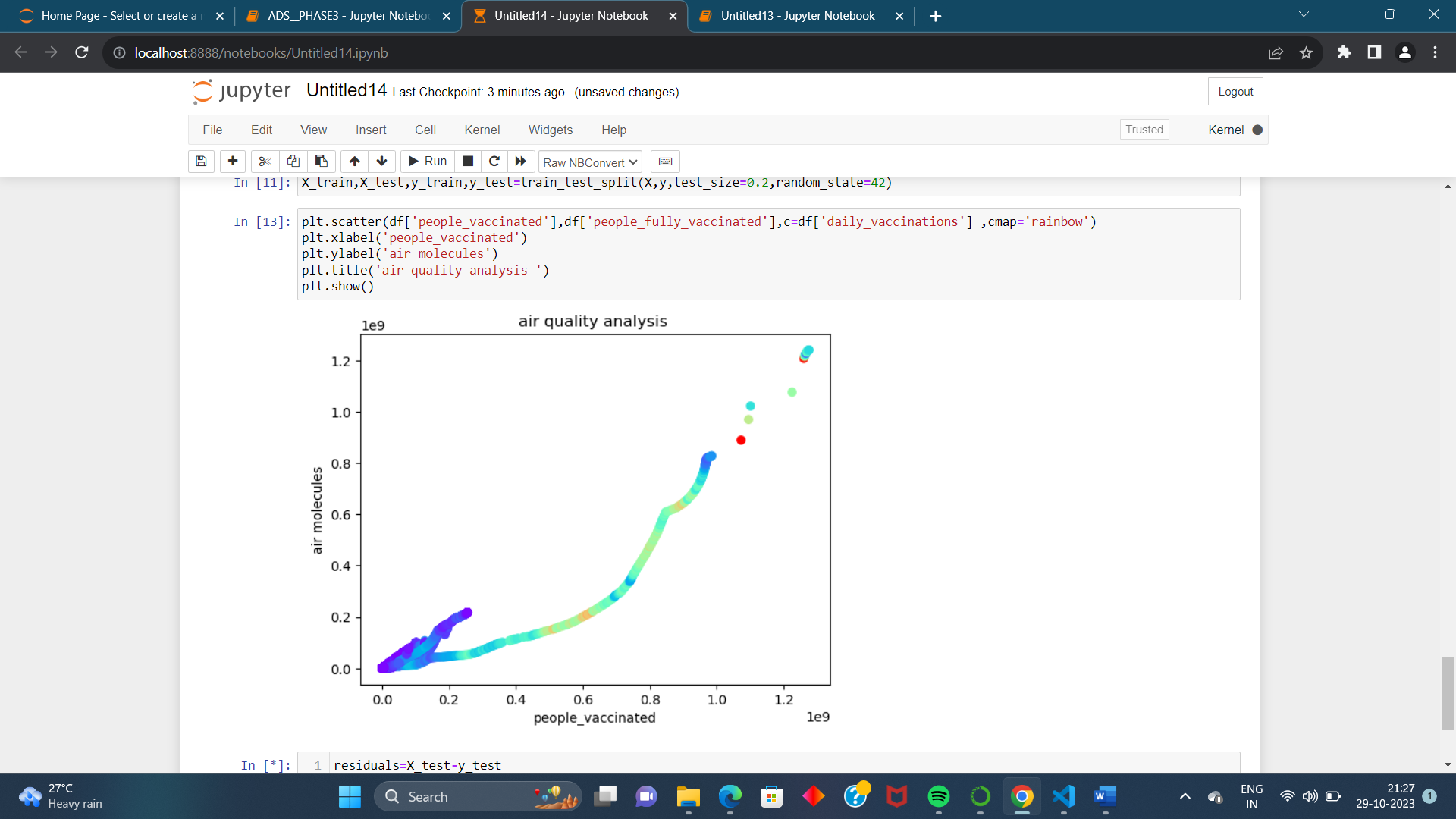
• During training, the model will learn patterns in the data.

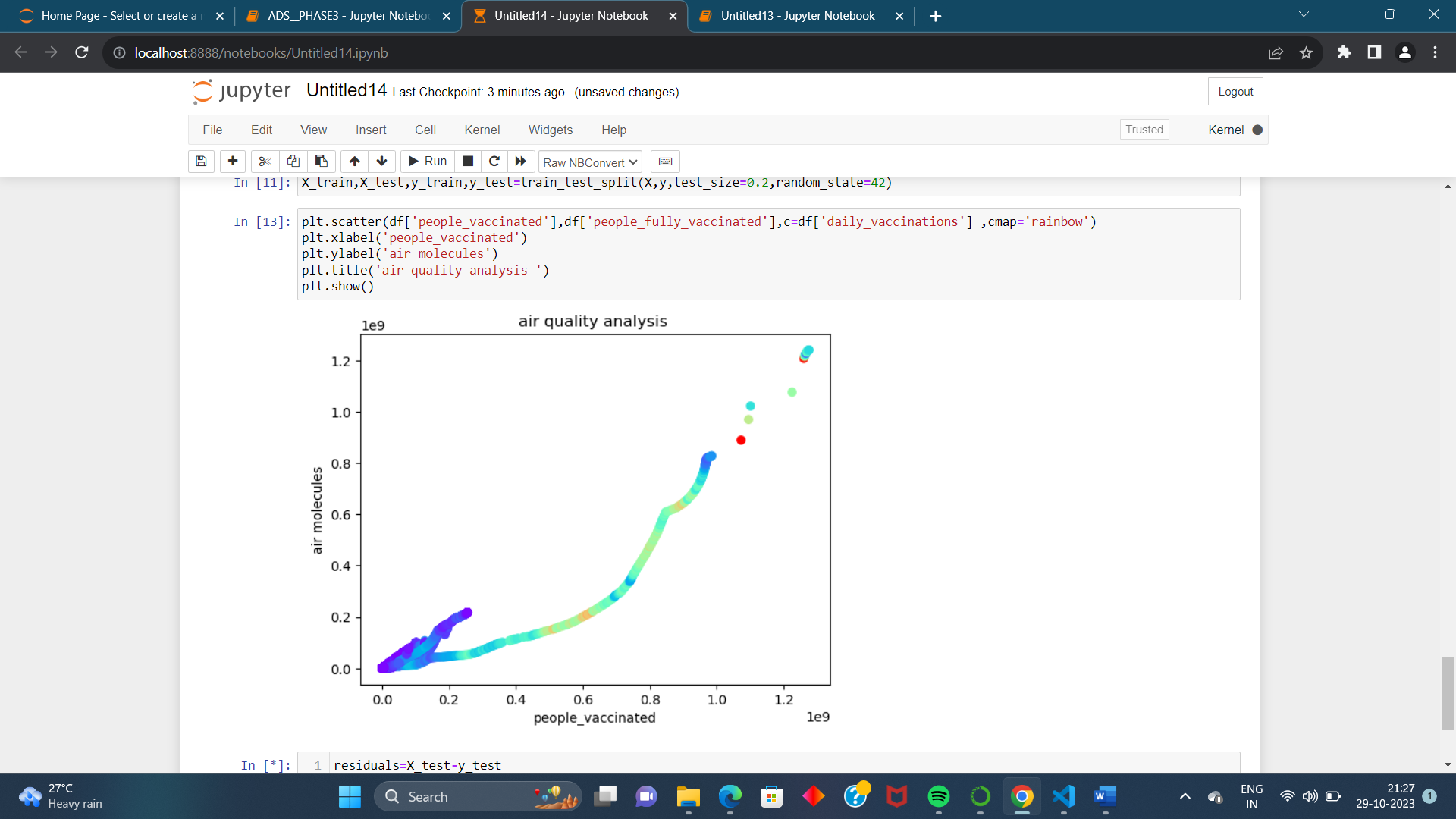
**EXECUTION OF THE MODEL:**

****

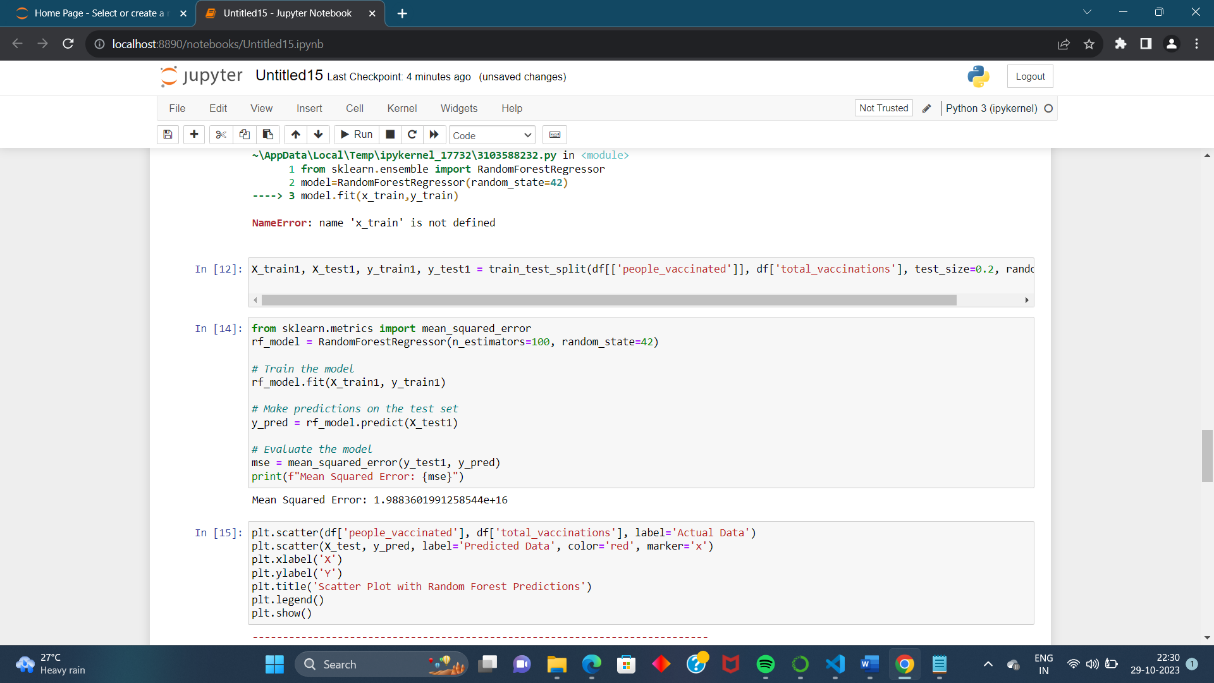
**DOING TESTING FOR SCATTER ANALYSIS**

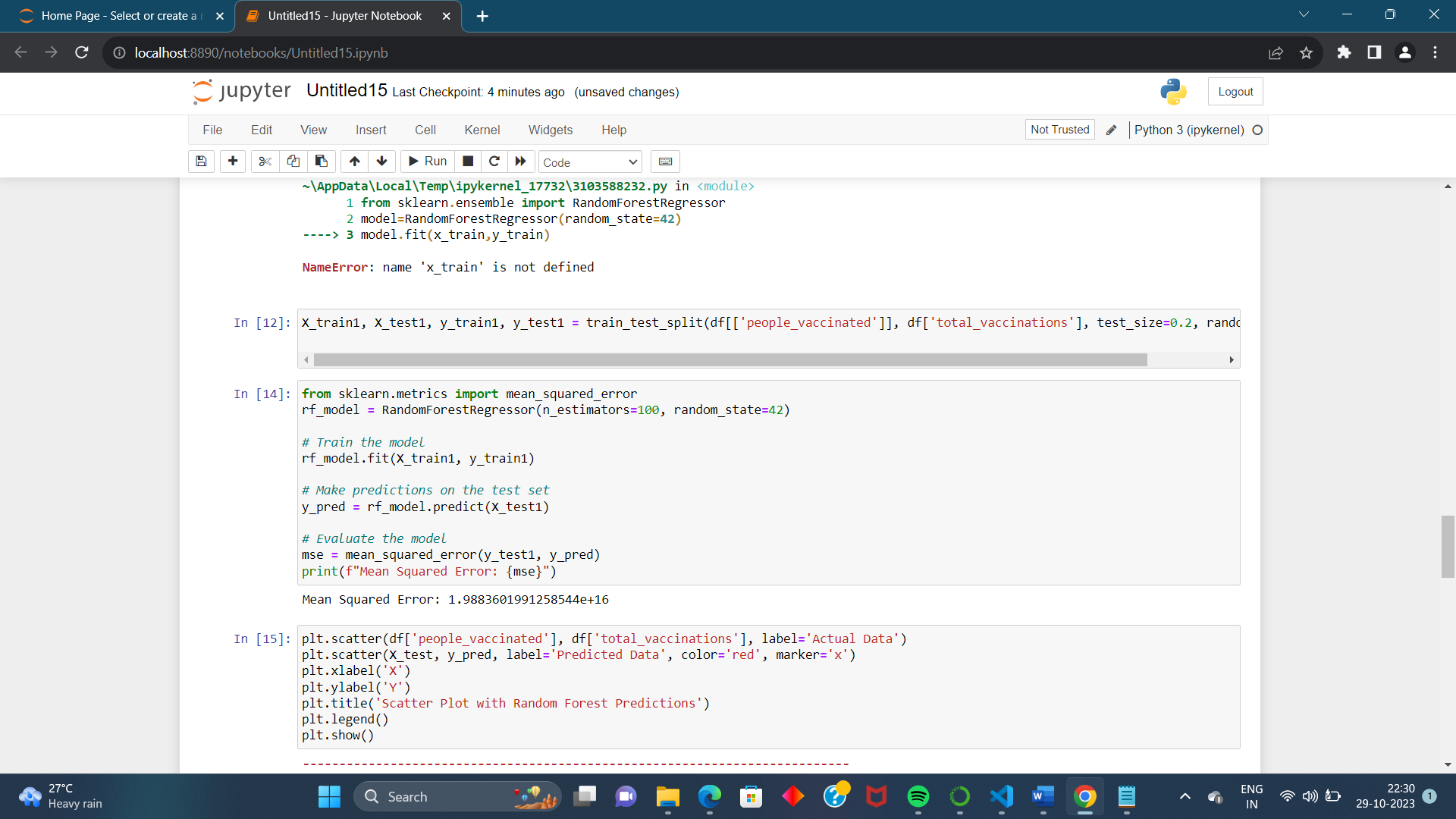
****

****

****

**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.**