COVID 19 VACCINE ANALYSIS

PHASE 4

OVER VIEW:

This phase of our project is arriving at

- Feature Engineering
- Model Evaluation

Where the preprocessed data on which earlier stage preliminary fundamental analytical tests have been performed is taken and further put to deeper understanding of the data information, clearer information processing and crisp on-point analysis that are advanced and extensively efficient.

FEATURE ENGINEERING:

•Feature engineering is the process of selecting, transforming, or creating data attributes (features) to enhance the performance of machine learning models.

•It involves tasks such as feature selection, handling missing data, scaling, encoding categorical variables, and creating new informative features.

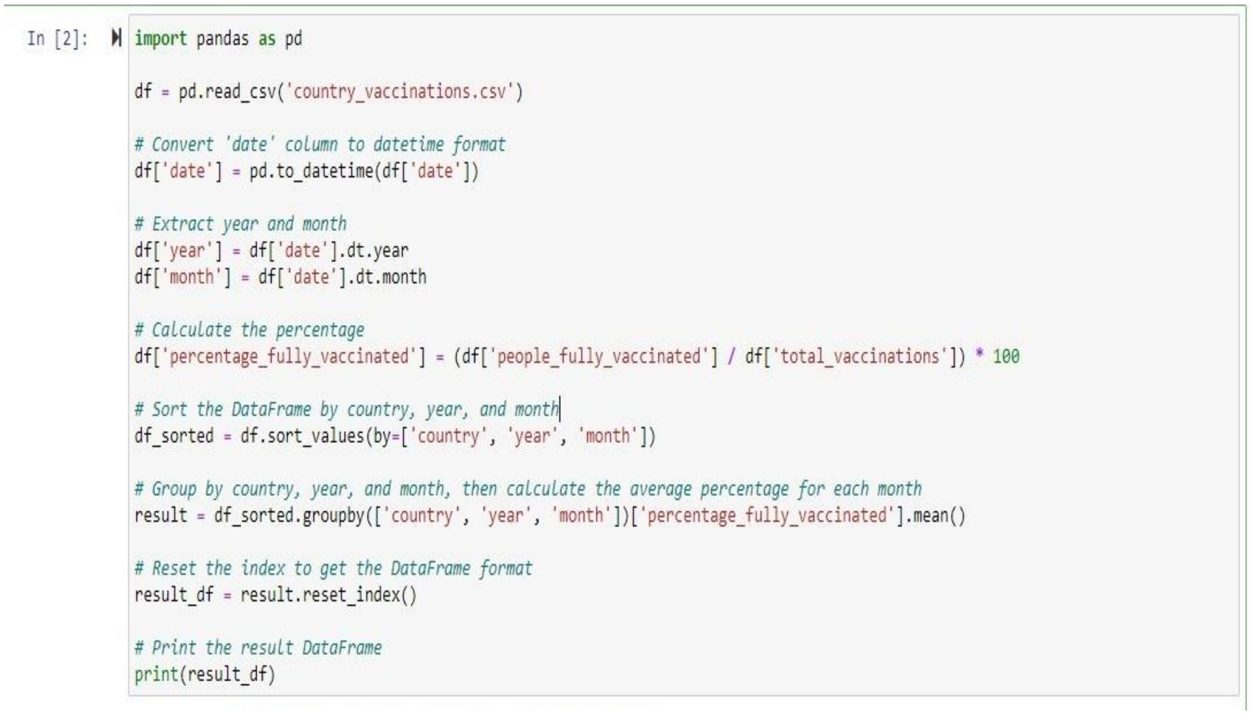
•Effective feature engineering can significantly impact a model's accuracy and predictive power. It requires domain knowledge and a balance between simplification and information retention.

TIME SERIES ANALYSIS:

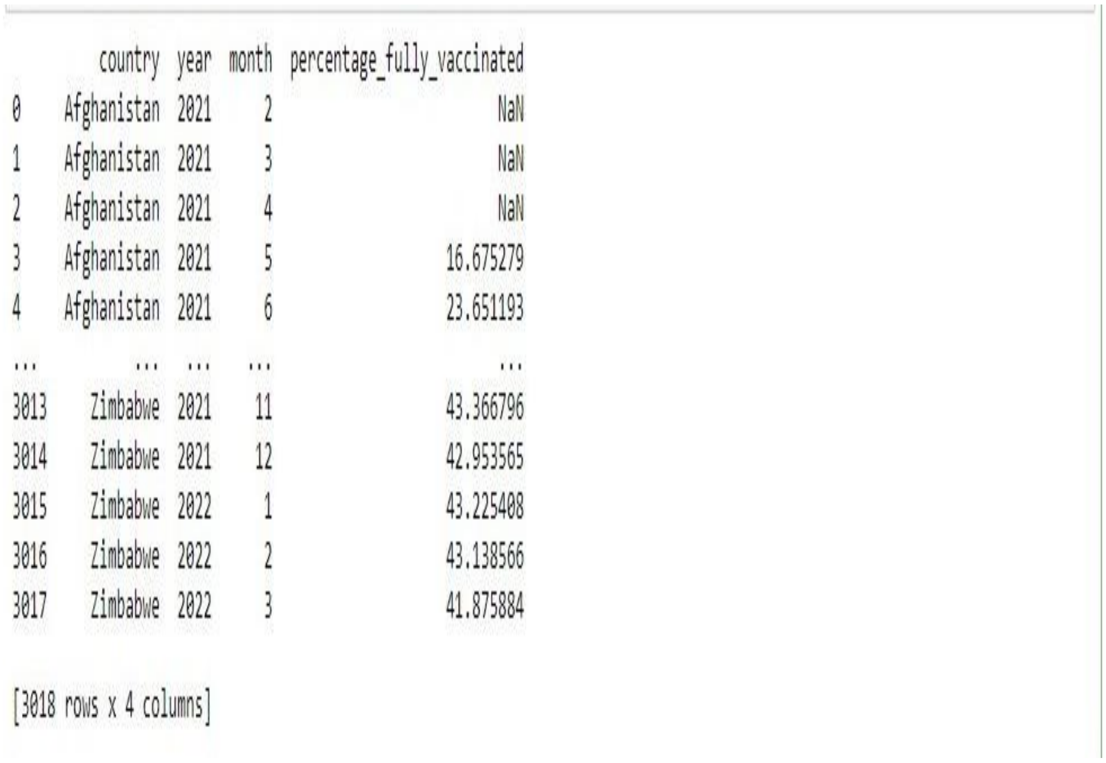
Time series analysis is a statistical method for studying data collected over time. It involves identifying patterns, trends, and seasonality within the data, making predictions, and detecting anomalies.

Common techniques include decomposition, forecasting models like ARIMA, and visualizing data for better insights into temporal relationships.

We performed this analysis in our code below to find percentage of a country fully vaccinated month wise,



```
In [2]:  import pandas as pd

         df = pd.read_csv('country_vaccinations.csv')

         # Convert 'date' column to datetime format
         df['date'] = pd.to_datetime(df['date'])

         # Extract year and month
         df['year'] = df['date'].dt.year
         df['month'] = df['date'].dt.month

         # Calculate the percentage
         df['percentage_fully_vaccinated'] = (df['people_fully_vaccinated'] / df['total_vaccinations']) * 100

         # Sort the DataFrame by country, year, and month
         df_sorted = df.sort_values(by=['country', 'year', 'month'])

         # Group by country, year, and month, then calculate the average percentage for each month
         result = df_sorted.groupby(['country', 'year', 'month'])['percentage_fully_vaccinated'].mean()

         # Reset the index to get the DataFrame format
         result_df = result.reset_index()

         # Print the result DataFrame
         print(result_df)
```



```
          country  year  month  percentage_fully_vaccinated
0     Afghanistan  2021      2                          NaN
1     Afghanistan  2021      3                          NaN
2     Afghanistan  2021      4                          NaN
3     Afghanistan  2021      5                    16.675279
4     Afghanistan  2021      6                    23.651193
...           ...   ...    ...                          ...
3013     Zimbabwe  2021     11                    43.366796
3014     Zimbabwe  2021     12                    42.953565
3015     Zimbabwe  2022      1                    43.225408
3016     Zimbabwe  2022      2                    43.138566
3017     Zimbabwe  2022      3                    41.875884

[3018 rows x 4 columns]
```

•Initially, we load the dataset and ensure the dates are in the right format for accurate analysis. Then, we extract the year and month, which will be crucial for understanding trends over time.

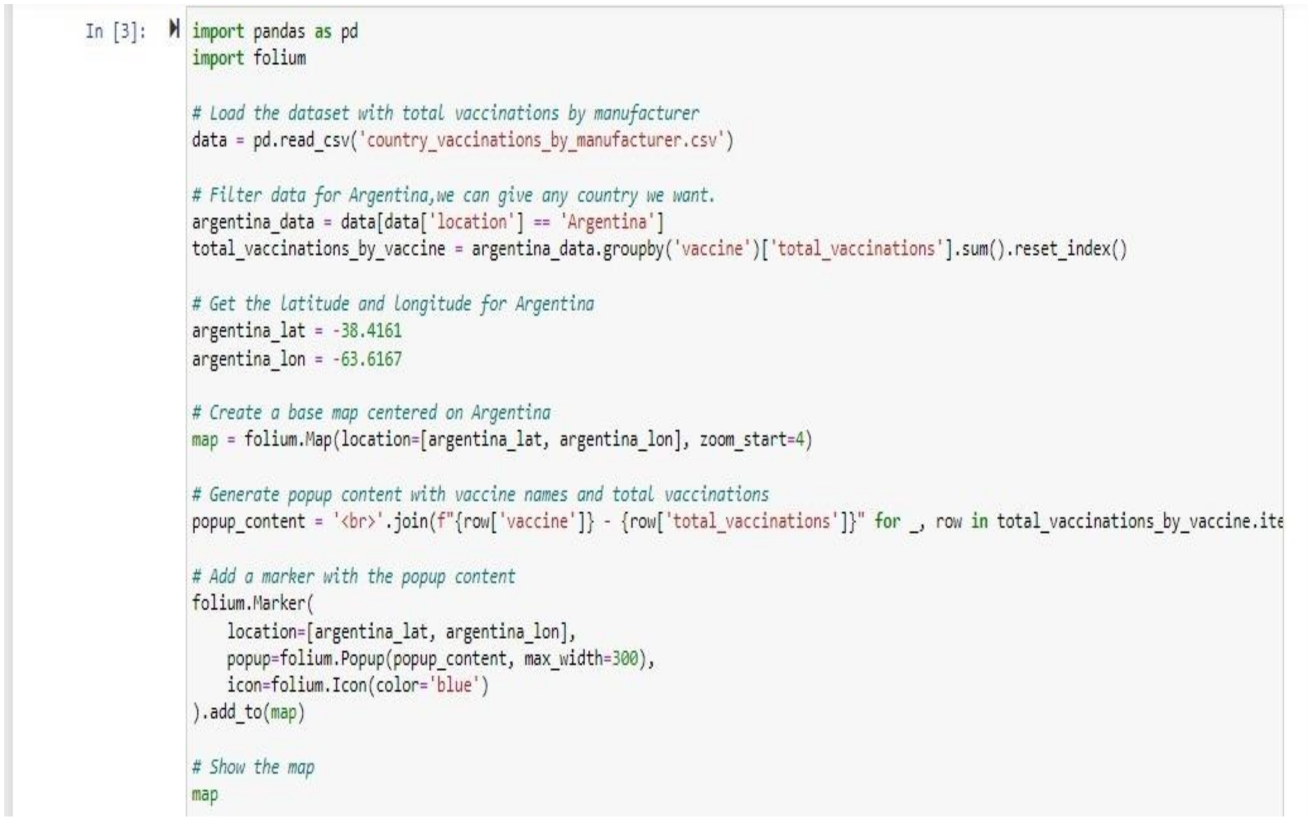
•Moving forward, we calculate the percentage of individuals who are fully vaccinated. To keep things organized, we sort the data by country, year, and month.

Following that, we group the data and compute the average percentage of fully vaccinated individuals for each month

<u>GEOSPATIAL ANALYSIS</u>:

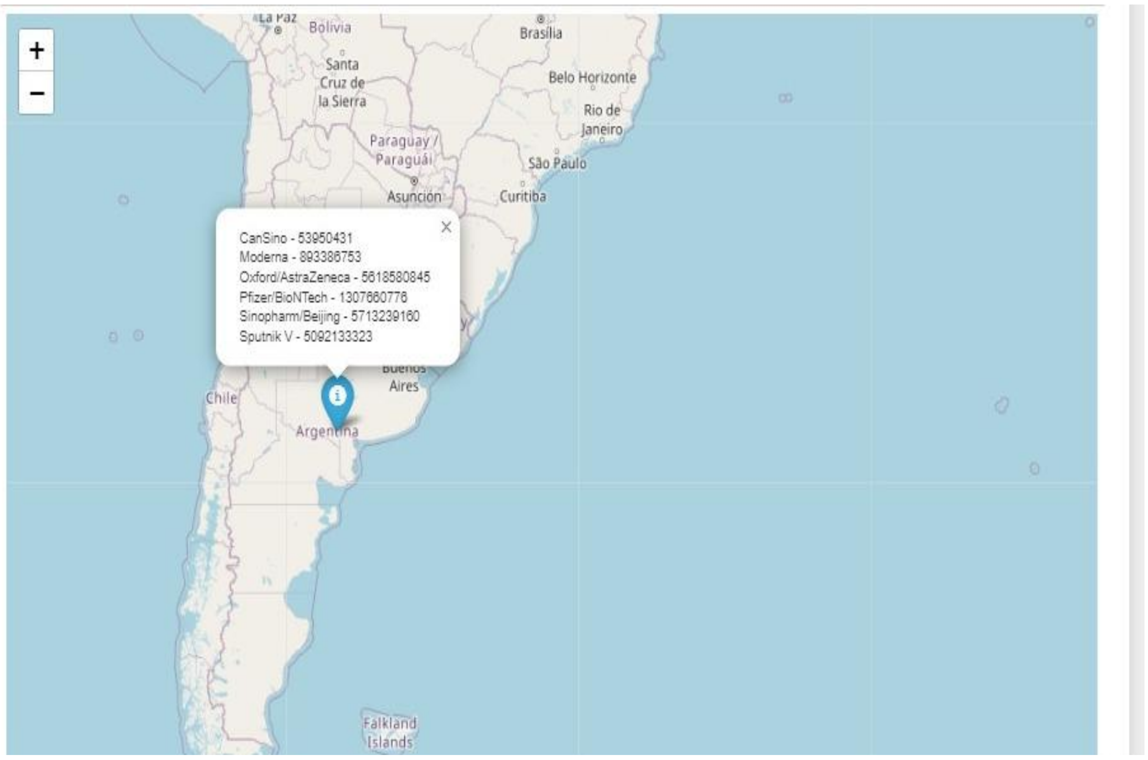
Geospatial analysis is the process of examining and interpreting data that is linked to geographic locations. It involves using geographic information systems (GIS) and other tools to analyze spatial patterns, relationships, and trends.

Geospatial analysis is vital in fields like urban planning, environmental management, logistics, and disaster response for informed decision-making and problem-solving based on location-based data.We performed this analysis in our code below to examine a visualize about total vaccination



```
In [3]:  import pandas as pd
         import folium

         # Load the dataset with total vaccinations by manufacturer
         data = pd.read_csv('country_vaccinations_by_manufacturer.csv')

         # Filter data for Argentina,we can give any country we want.
         argentina_data = data[data['location'] == 'Argentina']
         total_vaccinations_by_vaccine = argentina_data.groupby('vaccine')['total_vaccinations'].sum().reset_index()

         # Get the latitude and longitude for Argentina
         argentina_lat = -38.4161
         argentina_lon = -63.6167

         # Create a base map centered on Argentina
         map = folium.Map(location=[argentina_lat, argentina_lon], zoom_start=4)

         # Generate popup content with vaccine names and total vaccinations
         popup_content = '<br>'.join(f"{row['vaccine']} - {row['total_vaccinations']}" for _, row in total_vaccinations_by_vaccine.ite

         # Add a marker with the popup content
         folium.Marker(
             location=[argentina_lat, argentina_lon],
             popup=folium.Popup(popup_content, max_width=300),
             icon=folium.Icon(color='blue')
         ).add_to(map)

         # Show the map
         map
```

Out[3]:



```
CanSino - 53950431
Moderna - 893386753
Oxford/AstraZeneca - 5618580845
Pfizer/BioNTech - 1307660776
Sinopharm/Beijing - 5713239160
Sputnik V - 5092133323
```

•Firstly, we load a dataset containing information about total vaccinations by manufacturer. We then filter this data specifically for Argentina to focus our analysis.

•After that, we aggregate the total vaccinations for each vaccine type.Next, we gather the latitude and longitude coordinates for Argentina. Using these coordinates, we create a map centered on the country.

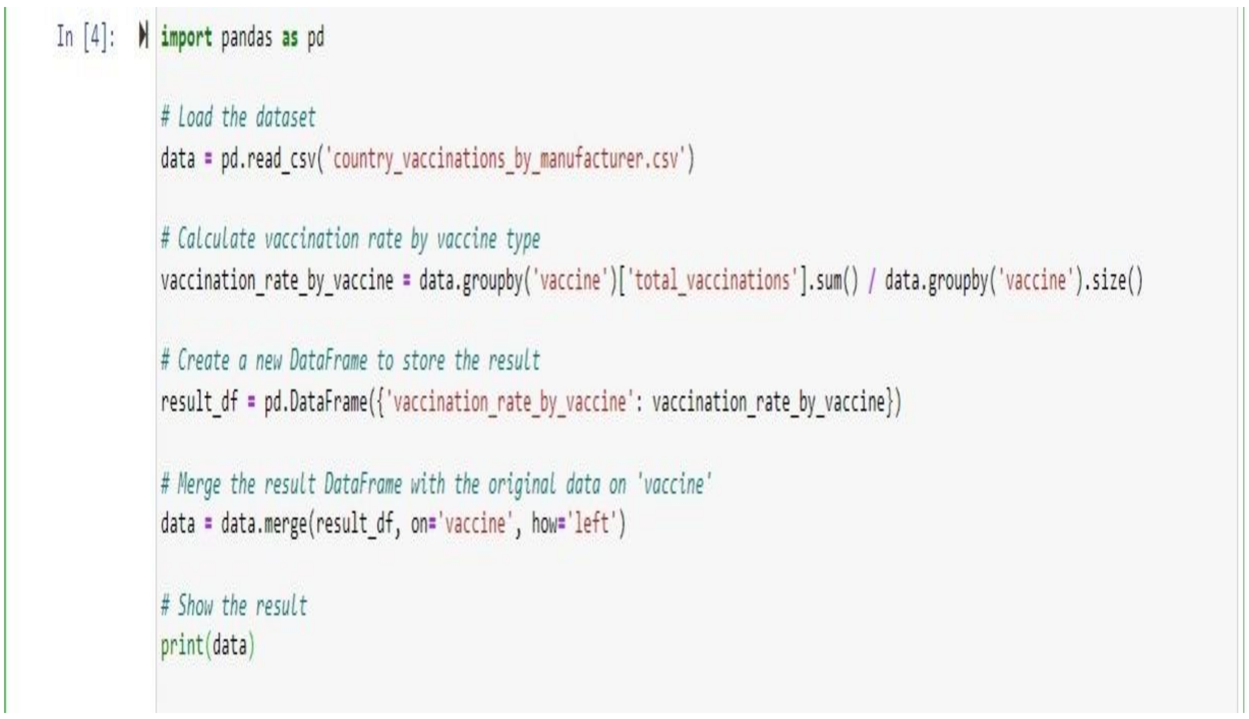
•Now, for each vaccine, we generate a popup displaying its name along with the total vaccinations administered. To visualize this on the map, we add a marker precisely at the center of Argentina.

•Clicking on this marker will provide detailed information about the vaccines and their respective totals. Finally, we display the map to see the distribution of vaccinations across different vaccine types.

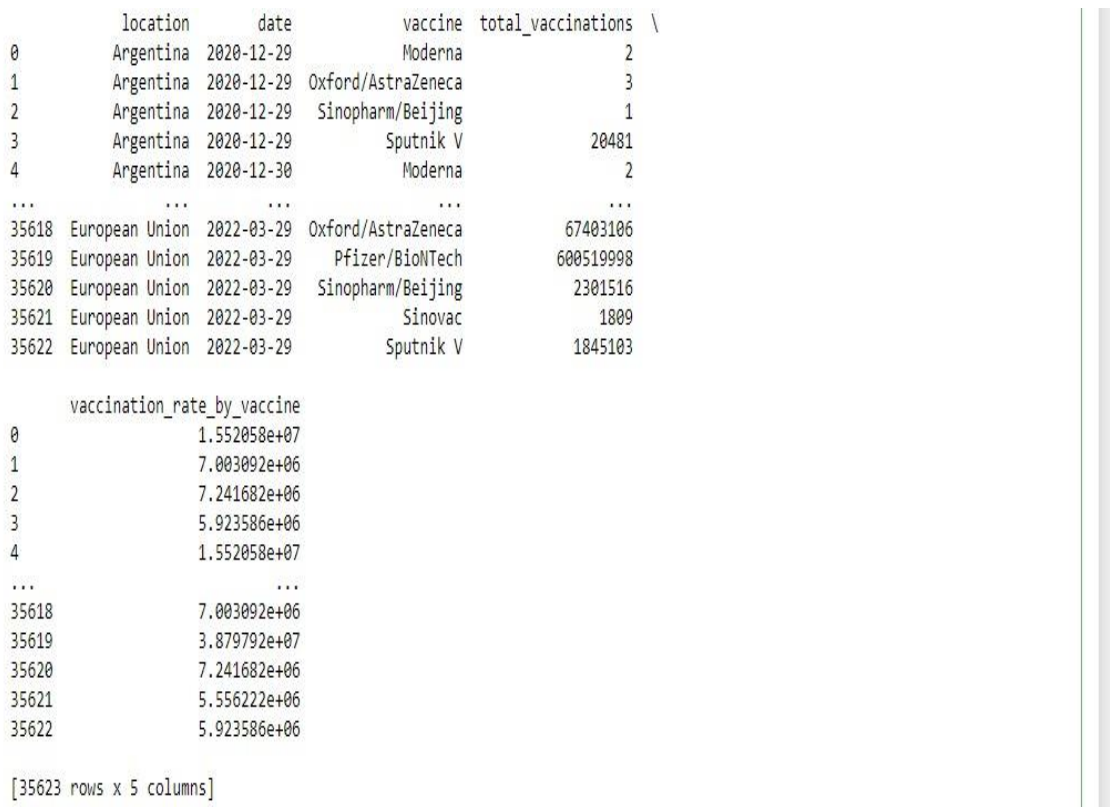
<u>DERIVED METRICS:</u>

Derived metrics are calculated or derived from existing data to provide additional insights. These metrics help to understand underlying patterns, relationships, and trends.

They often involve mathematical operations or transformations on the original data, such as calculating percentages, averages, or ratios. Derived metrics are valuable in analytics and reporting for making data-driven decisions and gaining a deeper understanding of complex datasets



```
In [4]:  import pandas as pd

         # Load the dataset
         data = pd.read_csv('country_vaccinations_by_manufacturer.csv')

         # Calculate vaccination rate by vaccine type
         vaccination_rate_by_vaccine = data.groupby('vaccine')['total_vaccinations'].sum() / data.groupby('vaccine').size()

         # Create a new DataFrame to store the result
         result_df = pd.DataFrame({'vaccination_rate_by_vaccine': vaccination_rate_by_vaccine})

         # Merge the result DataFrame with the original data on 'vaccine'
         data = data.merge(result_df, on='vaccine', how='left')

         # Show the result
         print(data)
```



```
             location        date            vaccine  total_vaccinations  \
0           Argentina  2020-12-29            Moderna                   2
1           Argentina  2020-12-29  Oxford/AstraZeneca                  3
2           Argentina  2020-12-29   Sinopharm/Beijing                  1
3           Argentina  2020-12-29          Sputnik V               20481
4           Argentina  2020-12-30            Moderna                   2
...               ...         ...                ...                 ...
35618  European Union  2022-03-29  Oxford/AstraZeneca            67403106
35619  European Union  2022-03-29    Pfizer/BioNTech           600519998
35620  European Union  2022-03-29   Sinopharm/Beijing             2301516
35621  European Union  2022-03-29            Sinovac                1809
35622  European Union  2022-03-29          Sputnik V             1845103

       vaccination_rate_by_vaccine
0                     1.552058e+07
1                     7.003092e+06
2                     7.241682e+06
3                     5.923586e+06
4                     1.552058e+07
...                            ...
35618                 7.003092e+06
35619                 3.879792e+07
35620                 7.241682e+06
35621                 5.556222e+06
35622                 5.923586e+06

[35623 rows x 5 columns]
```

MODEL TRAINING:

Model training is the phase in the data science development lifecycle where practitioners try to fit the best combination of weights and bias to a machine learning algorithm to minimize a loss function over the prediction range.

A linear regression model can be trained using the optimization algorithm gradient descent by iteratively modifying the model's parameters to reduce the mean squared error (MSE) of the model on a training dataset. To update θ_1 and θ_2 values in order to reduce the Cost function (minimizing RMSE value) and achieve the best-fit line the model uses Gradient Descent.

The idea is to start with random θ_1 and θ_2 values and then iteratively update the values, reaching minimum cost. A gradient is nothing but a derivative that defines the effects on outputs of the function with a little bit of variation in inputs

Let's differentiate the cost function(J) with respect to θ_1

$$\begin{aligned}
J'_{\theta_1} &= \frac{\partial J(\theta_1, \theta_2)}{\partial \theta_1} \\
&= \frac{\partial}{\partial \theta_1}\left[\frac{1}{n}\left(\sum_{i=1}^{n}(\hat{y}_i - y_i)^2\right)\right] \\
&= \frac{1}{n}\left[\sum_{i=1}^{n} 2(\hat{y}_i - y_i)\left(\frac{\partial}{\partial \theta_1}(\hat{y}_i - y_i)\right)\right] \\
&= \frac{1}{n}\left[\sum_{i=1}^{n} 2(\hat{y}_i - y_i)\left(\frac{\partial}{\partial \theta_1}(\theta_1 + \theta_2 x_i - y_i)\right)\right] \\
&= \frac{1}{n}\left[\sum_{i=1}^{n} 2(\hat{y}_i - y_i)\,(1 + 0 - 0)\right] \\
&= \frac{1}{n}\left[\sum_{i=1}^{n}(\hat{y}_i - y_i)\,(2)\right] \\
&= \frac{2}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)
\end{aligned}$$

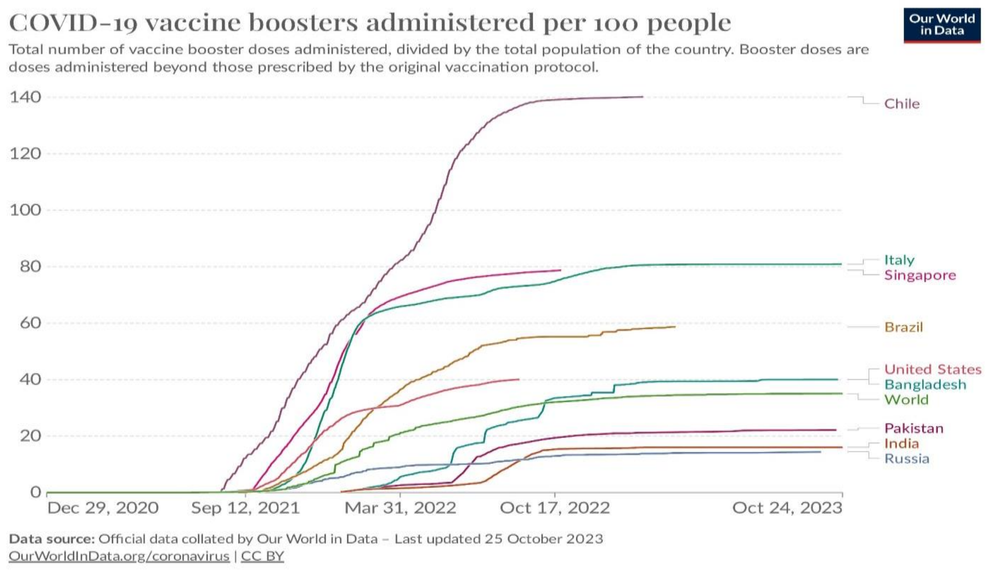
Let's differentiate the cost function(J) with respect to θ_2

$$
\begin{aligned}
J'_{\theta_2} &= \frac{\partial J(\theta_1,\theta_2)}{\partial \theta_2} \\
&= \frac{\partial}{\partial \theta_2}\left[\frac{1}{n}\left(\sum_{i=1}^{n}(\hat{y}_i - y_i)^2\right)\right] \\
&= \frac{1}{n}\left[\sum_{i=1}^{n} 2(\hat{y}_i - y_i)\left(\frac{\partial}{\partial \theta_2}(\hat{y}_i - y_i)\right)\right] \\
&= \frac{1}{n}\left[\sum_{i=1}^{n} 2(\hat{y}_i - y_i)\left(\frac{\partial}{\partial \theta_2}(\theta_1 + \theta_2 x_i - y_i)\right)\right] \\
&= \frac{1}{n}\left[\sum_{i=1}^{n} 2(\hat{y}_i - y_i)\,(0 + x_i - 0)\right] \\
&= \frac{1}{n}\left[\sum_{i=1}^{n} (\hat{y}_i - y_i)\,(2x_i)\right] \\
&= \frac{2}{n}\sum_{i=1}^{n} (\hat{y}_i - y_i)\cdot x_i
\end{aligned}
$$

Finding the coefficients of a linear equation that best fits the training data is the objective of linear regression. By moving in the direction of the Mean Squared Error negative gradient with respect to the coefficients, the coefficients can be changed. And the respective intercept and coefficient of X will be if α is the learning rate.

Parameter Tuning

Selecting the correct parameter that will be modified to influence the ML model is key to attaining accurate correlation. The set of parameters that are selected based on their influence on the model architecture are called hyperparameters. The process of identifying the hyperparameters by tuning the model is called parameter tuning. The parameters for correlation should be clearly defined in a manner in which the point of diminishing returns for validation is as close to 100% accuracy as possible.

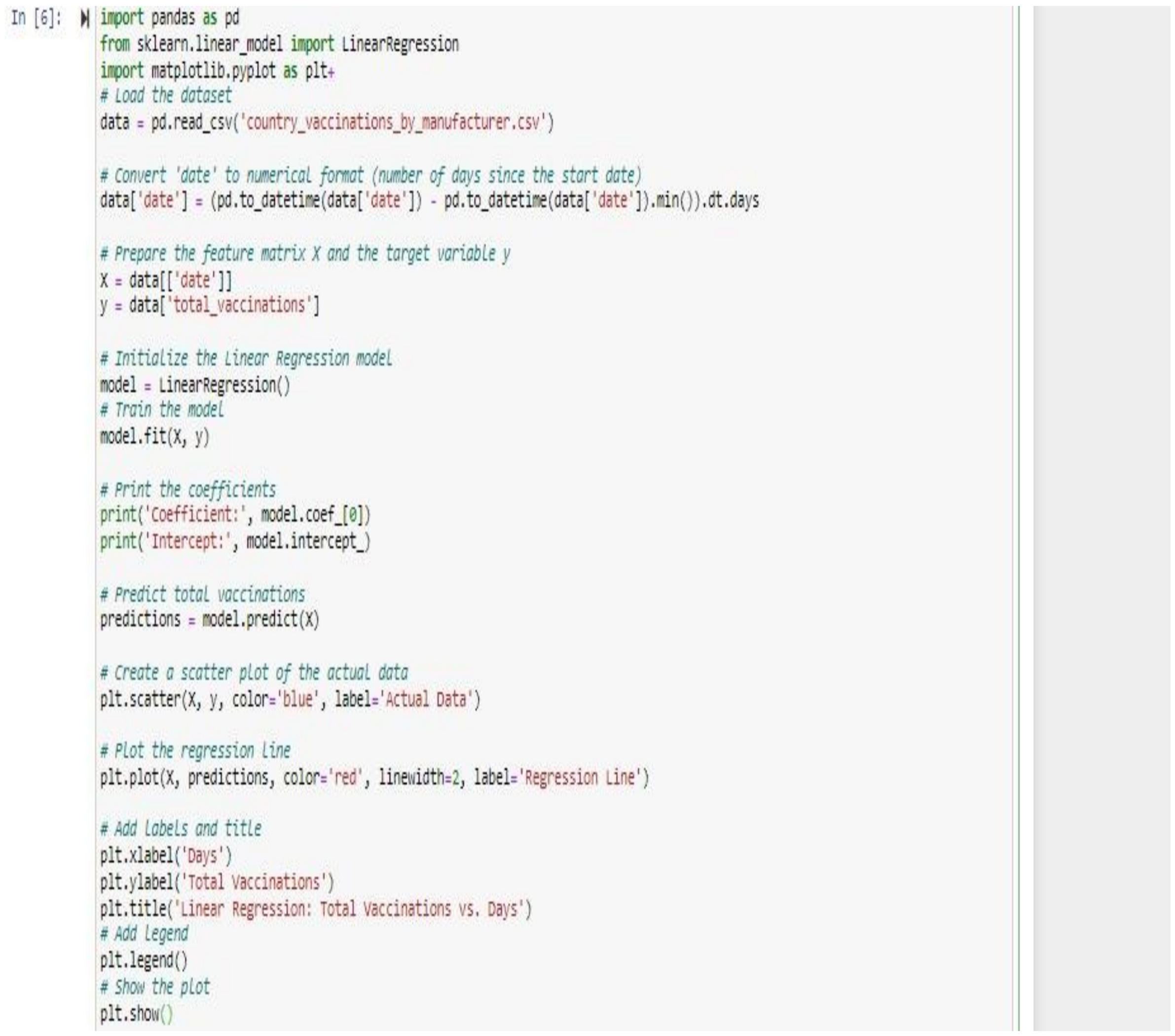


LINEAR REGRESSION:

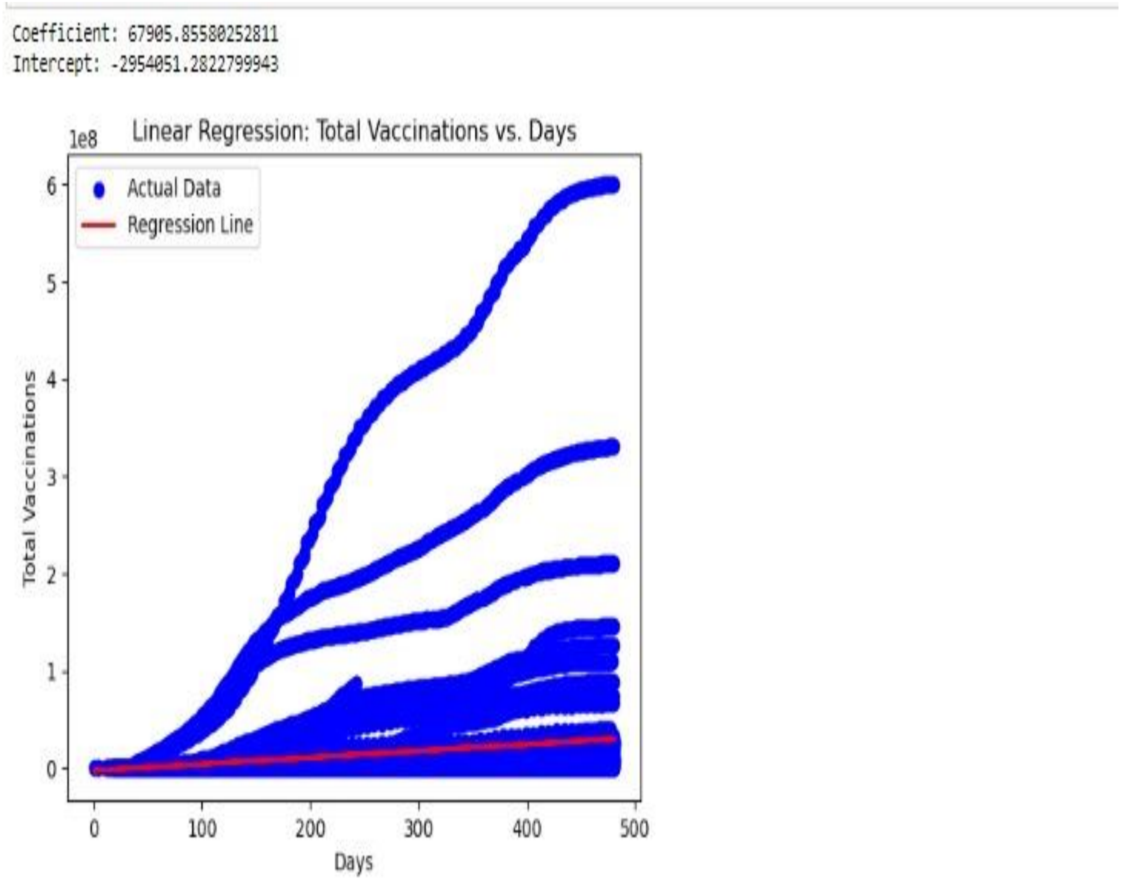
Linear regression is a statistical method used to model the relationship between a dependent variable and one or more independent variables by fitting a linear equation. It aims to predict the dependent variable based on the values of the independent variables.

The model assumes a linear relationship, with coefficients representing the impact of each independent variable on the dependent variable. Linear regression is widely used for predictive modeling and understanding the strength and direction of relationships between variables.

In the code below.we, are training the model using the linear regression technique to predict the vaccinations rate by the number of vaccinations obtained in certain days .if thr coefficient is positive, it means that as the number of days increases, the total vaccinations are expected to increase as well.



```
In [6]:  import pandas as pd
         from sklearn.linear_model import LinearRegression
         import matplotlib.pyplot as plt+
         # Load the dataset
         data = pd.read_csv('country_vaccinations_by_manufacturer.csv')

         # Convert 'date' to numerical format (number of days since the start date)
         data['date'] = (pd.to_datetime(data['date']) - pd.to_datetime(data['date']).min()).dt.days

         # Prepare the feature matrix X and the target variable y
         X = data[['date']]
         y = data['total_vaccinations']

         # Initialize the Linear Regression model
         model = LinearRegression()
         # Train the model
         model.fit(X, y)

         # Print the coefficients
         print('Coefficient:', model.coef_[0])
         print('Intercept:', model.intercept_)

         # Predict total vaccinations
         predictions = model.predict(X)

         # Create a scatter plot of the actual data
         plt.scatter(X, y, color='blue', label='Actual Data')

         # Plot the regression line
         plt.plot(X, predictions, color='red', linewidth=2, label='Regression Line')

         # Add labels and title
         plt.xlabel('Days')
         plt.ylabel('Total Vaccinations')
         plt.title('Linear Regression: Total Vaccinations vs. Days')
         # Add legend
         plt.legend()
         # Show the plot
         plt.show()
```



```
Coefficient: 67905.85580252811
Intercept: -2954051.2822799943
```

- We're applying Linear Regression to analyze COVID-19 vaccination data. Firstly, we load a dataset .Then, to make the 'date' column usable for the regression model, we convert it into a numerical format representing the number of days since the start date. Next, we prepare the feature matrix 'X', containing the numerical representation of dates, and the target variable 'y', representing the total vaccinations
- . We then initialize a Linear Regression model. Moving on, we train the model on the given data. After training, we print out the coefficients, providing insights into how the model has learned from the data.

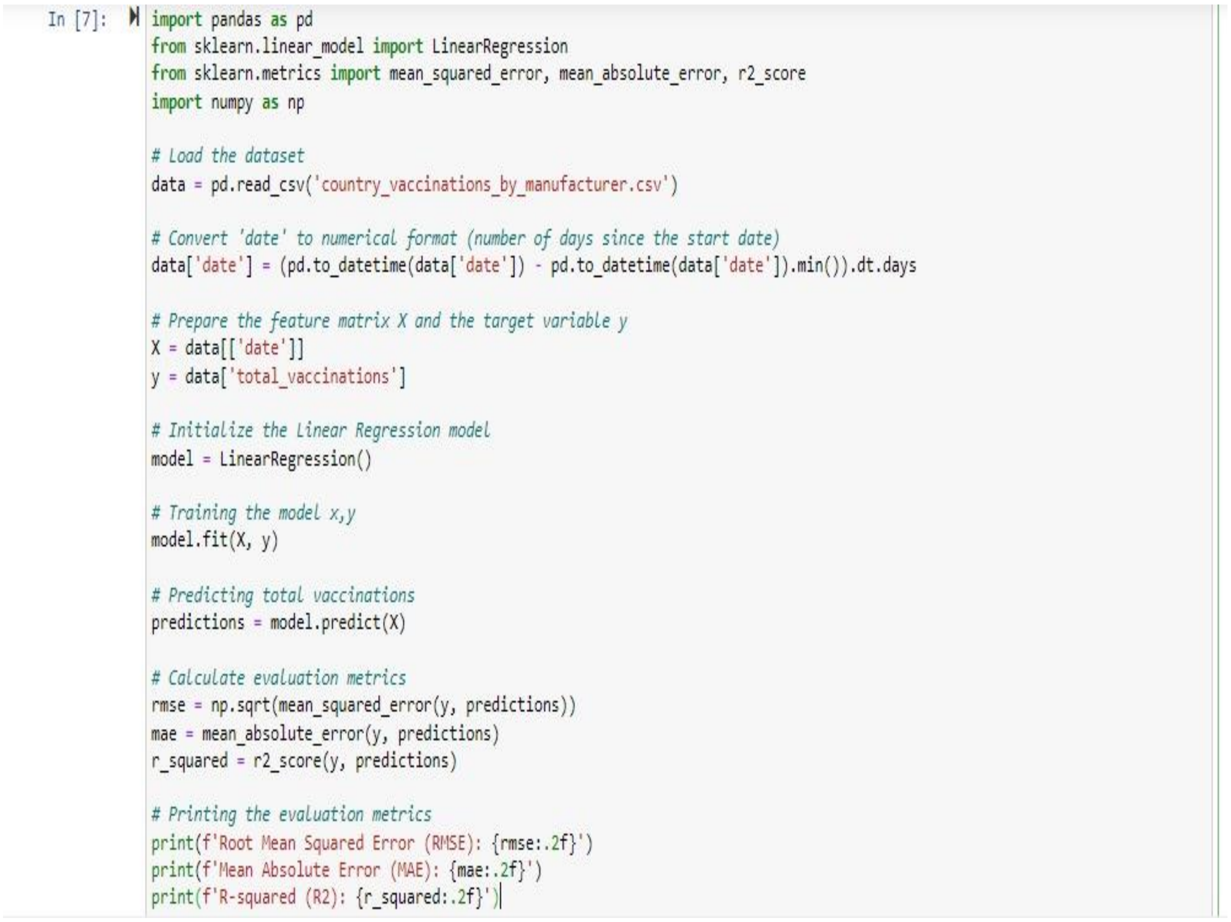
We then use the trained model to predict total vaccinations. For visualization, we create a scatter plot of the actual data points in blue. Additionally, we plot the regression line in red, which represents the model's predictions. This line showcases the relationship between days and total vaccinations as predicted by the model.

Coefficient: 67905.85580252811

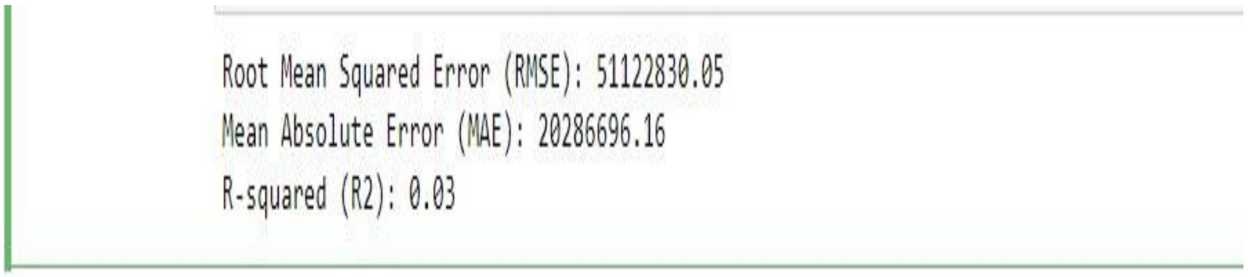
This means that for every additional day, the total number of vaccinations is expected to increase by approximately 67906. This is the slope of the regression line.

MODEL EVALUATION:

Model evaluation is the process of assessing the performance and effectiveness of a machine learning model. It involves techniques like splitting data into training and testing sets, cross-validation, and various metrics (e.g., accuracy, precision, recall, F1-score) to measure a model's predictive quality. The goal is to ensure the model generalizes well to new, unseen data and to make informed decisions about its suitability for a specific task.



```
In [7]:  import pandas as pd
         from sklearn.linear_model import LinearRegression
         from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
         import numpy as np

         # Load the dataset
         data = pd.read_csv('country_vaccinations_by_manufacturer.csv')

         # Convert 'date' to numerical format (number of days since the start date)
         data['date'] = (pd.to_datetime(data['date']) - pd.to_datetime(data['date']).min()).dt.days

         # Prepare the feature matrix X and the target variable y
         X = data[['date']]
         y = data['total_vaccinations']

         # Initialize the Linear Regression model
         model = LinearRegression()

         # Training the model x,y
         model.fit(X, y)

         # Predicting total vaccinations
         predictions = model.predict(X)

         # Calculate evaluation metrics
         rmse = np.sqrt(mean_squared_error(y, predictions))
         mae = mean_absolute_error(y, predictions)
         r_squared = r2_score(y, predictions)

         # Printing the evaluation metrics
         print(f'Root Mean Squared Error (RMSE): {rmse:.2f}')
         print(f'Mean Absolute Error (MAE): {mae:.2f}')
         print(f'R-squared (R2): {r_squared:.2f}')
```



```
Root Mean Squared Error (RMSE): 51122830.05
Mean Absolute Error (MAE): 20286696.16
R-squared (R2): 0.03
```

Moving on to model evaluation, we calculate three important metrics. These metrics help to assess how well the model is performing in terms of predicting the total vaccinations based on the number of days since the start date.:

- Root Mean Squared Error (RMSE): This measures the average error between predicted and actual values, with higher values indicating greater deviation.
- Mean Absolute Error (MAE): It computes the average absolute difference between predicted and actual values, providing another perspective on the model's performance.
- R-squared (R2) Score: This metric assesses how well the model's predictions align with the actual data. An R2 score of 1 indicates a perfect fit.

Finally, we print out these evaluation metrics, offering a comprehensive view of the model's performance. These metrics are crucial in assessing the accuracy and effectiveness of the Linear Regression model.

CONCLUSION:

In this phase our datasets have been transformed into a Data Model for Analysis by employing the following techniques.

Feature Engineering

- Time series analysis
- Geospatial Analysis
- Derived Metrics

Model Training and Machine Learning

- Linear Regression Training
- History Based Learning

The model evaluation of our data science model’s have been carried out using a new-to-tech methodology involving Machine Learning which evaluates the model’s based on training memory to compare and improve the models error free efficiency and overall analysis success.