Covid vaccines analysis

**Introduction:**

The global battle against COVID 19 pandemic can be won only if a large part of the world gets vaccinated against the SARS-CoV-2 virus. In this blog, we study the COVID 19 vaccination trends across the world using python, and we aim to derive key insights from the data which can help policymakers modify their policies.Give recommendation manufacturers for their vacccinations At country.

On the project we by manufactures. These two datasets define vaccination among the country and vaccination by manufacture on day by day.have two different Datasets that is country vaccinations and country vaccination

**Problem outline:**

The problem is to conduct in-depth analysis of covid 19 vaccine data, focusing on the vaccine efficacy, distribution and adverse effects. The main is to provide hidden insights that aid policymakers and health organiztions in optimizing vaccine deployment strategies.It could useful to their vaccine distribution among the countries.

**Design Thinking:**

**1)Data collection:**

* + - Datasets were collected on the Kaggle website On the name of covid-19 world vaccination.
    - It contains two different datasets named country vaccination by manufactures, country vaccinations

**2)Load the data and required libraries:**

In this we need import the required libraries to notebook and

Load the downloaded dataset using a variable

**3)Data preprocessing:**

* + - Datasets may contain missing values and many outliers, so we need perform anomaly detection And cleansing on the datasets.

**4)Exploratory Data Analysis:**

* + - Perform the EDA process on the datasets to get some patterns among the data. Exploratory Data Analysis (EDA) is an analysis approach that identifies general patterns in the data. In this data
    - I got some hidden patterns among countries and their total vaccinations and also we can understand the relationship visually by use of libraries matplotlib and seaborn

**5)Statical Analysis:**

* + - Statistical analysis is the process of collecting and analysing large volumes of data in order to identify trends and develop valuable insights. In this data
    - We can make some statistical analysis to identify the descriptive among the data on both datasets. First of all we convert the date and time into proper format. We can find a good valuable insight to Optimize the vaccination distribution and adverse affects

**6)Machine learning Technique:**

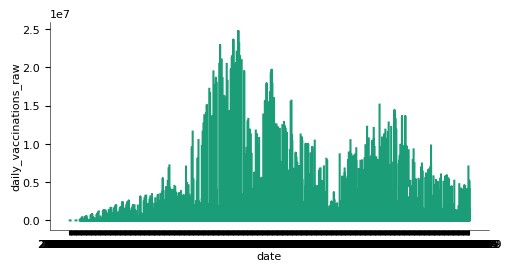
* + - Using the machine learning techniques to we can make a model to predict values attribute.Machine Learning Algorithms use full to make a good model And feed the data as per the preprocessed source. On my hope we can make a Good Regression model By this Dataset to we can predict the value.

Ex:-

* + - We can use people vaccinated column to predict the people fully vaccinated column for any country in the world.

**7)Time series Forecasting:**

* In this Data we seen the date wise vaccinations Country on all over world. So we make the TimeSeries Analysis on the datasets. To make Time Series Forecasting strategy to we plot future Time intervels to find the growth of the Vaccination rate among the Countries



**7)Visualization:**

Some visualization is good to use to understand the data and their relationship of the attributes .visualize the data easily understand

Technique. Some visualization I got online it provide



**8)Provide Insights and Recommendation:**

* At last we find some good hidden insights .It could use full for the vaccine distribution on the country so manufacture can focus on the Vaccine effect and country that are vaccinated In least count And also focus on the vaccines that might be used in multiple countries.
* We give the reccomedations on the vaccine manufacture growth and Vaccination on the countries that are people vaccinated in a least count.so th world organization took care on that countries

**Data collection:**

It is the process of gathering and analyzing accurate data from various sources to find answers to research problems, trends and probabilities.on our project data is collected at kaggle website.

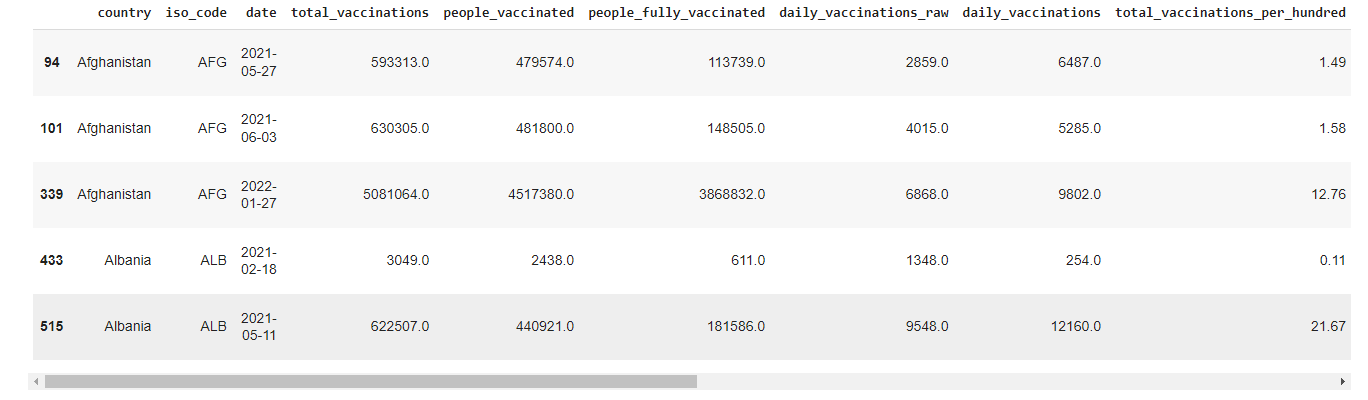
**About the data:-**

On project we have two different Datasets that is country vaccinations and country vaccination by manufactures. These two datasets define vaccination among the country and vaccination by manufacture on day by day.

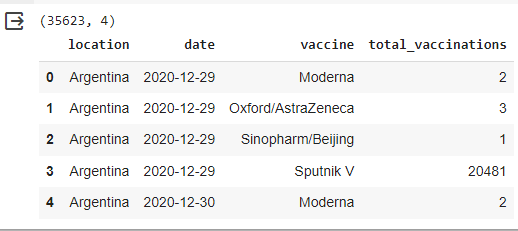
That data was displayed below first five entry with their size

**Country vaccinations**:

(86512, 15)



**Country vaccinations by manufactures:**

****

**Load the data and required libraries:**

Load the required libraries for the data analysis On the dataset to find the hidden details and patterns in the data.major language that is used in the analysis is python.libraries that are used for major analysis is Pandas,Numby,Matplotlib,Seaborn, And for the machine learning process we using the pycaret for model building.

**As per code**

**import libraries**

import numpy as np

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

import warnings

warnings.filterwarnings("ignore")

load the datasets

data=pd.read\_csv("/content/drive/MyDrive/country\_vaccinations.csv")

data\_manu=pd.read\_csv("country\_vaccinations\_by\_manufacturer.csv")

print(data.shape)

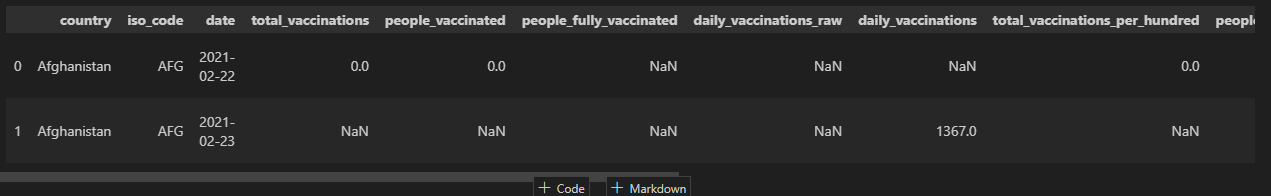
print(data.head(2))

print(datamanu.shape)

print(data.head(4))

output:

(86512, 15)



(35623, 4)

|  | **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 | 2048 |

**Data preprocessing:**

**Data preprocessing** can refer to manipulation or dropping of data before it is used in order to ensure or enhance performance,[[1]](https://en.wikipedia.org/wiki/Data_Preprocessing#cite_note-1) and is an important step in the data analysis process.

**Handling missing values:**

Find how many values are missed in datasets

**Using codes**

#using info we can undestsand the datatypes of column

print(data.info())

print(data\_manu.info())

print(data.isnull().sum())

print(data\_manu.isnull().sum())

output:

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 86512 entries, 0 to 86511

Data columns (total 15 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 country 86512 non-null object

1 iso\_code 86512 non-null object

2 date 86512 non-null object

3 total\_vaccinations 43607 non-null float64

4 people\_vaccinated 41294 non-null float64

5 people\_fully\_vaccinated 38802 non-null float64

6 daily\_vaccinations\_raw 35362 non-null float64

7 daily\_vaccinations 86213 non-null float64

8 total\_vaccinations\_per\_hundred 43607 non-null float64

9 people\_vaccinated\_per\_hundred 41294 non-null float64

10 people\_fully\_vaccinated\_per\_hundred 38802 non-null float64

11 daily\_vaccinations\_per\_million 86213 non-null float64

12 vaccines 86512 non-null object

13 source\_name 86512 non-null object

14 source\_website 86512 non-null object

dtypes: float64(9), object(6)

memory usage: 9.9+ M

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 35623 entries, 0 to 35622

Data columns (total 4 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 location 35623 non-null object

1 date 35623 non-null object

2 vaccine 35623 non-null object

3 total\_vaccinations 35623 non-null int64

dtypes: int64(1), object(3)

memory usage: 1.1+ MB

country 0

country 0

iso\_code 0

date 0

total\_vaccinations 42905

people\_vaccinated 45218

people\_fully\_vaccinated 47710

daily\_vaccinations\_raw 51150

daily\_vaccinations 299

total\_vaccinations\_per\_hundred 42905

people\_vaccinated\_per\_hundred 45218

people\_fully\_vaccinated\_per\_hundred 47710

daily\_vaccinations\_per\_million 299

vaccines 0

source\_name 0

source\_website 0

location 0

date 0

vaccine 0

total\_vaccinations 0

on the above data there is no missing value in data\_manu but data contain many missing values so we need handle the missing values in the data.half of the data contain null values so the best option just drop it.After droping the missing values data become null free.after checking the size it confirm it(because it reduced).

data.dropna(axis=0,inplace=True)

data.shape

output:

(30847, 15)

**Outlier handling:**

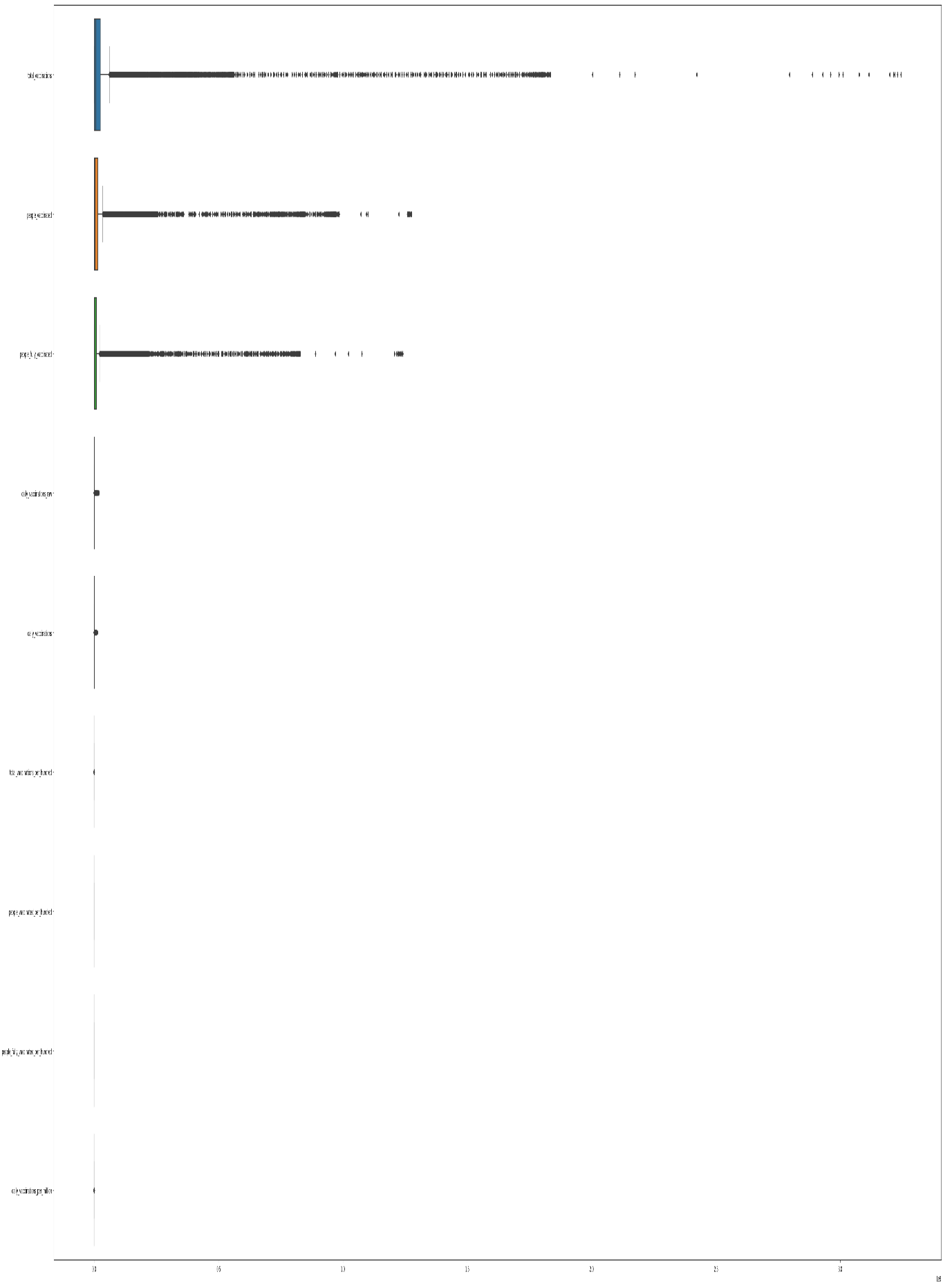
Outlier is a point that differs significantly from the Observation.The**interquartile range method**is one of the most commonly used methods to identify outlier.

**Boxplot is very useful visualize the ouliers on covid vaccinations**

plt.figure(figsize=(60,30))

sns.boxplot(data,orient='h')

plt.show()



Taking the count of an outliers

Outlier on total\_vaccinations is 4407

Outlier on people\_vaccinated is 4384

Outlier on people\_fully\_vaccinated is 4826

Outlier on daily\_vaccinations\_raw is 4091

Outlier on daily\_vaccinations is 4004

Outlier on total\_vaccinations\_per\_hundred is 26

Outlier on daily\_vaccinations\_per\_million is 769

After using interquatile method again to clear the outliers in data ,data become

list1=['total\_vaccinations','people\_vaccinated','people\_fully\_vaccinated','daily\_vaccinations\_raw','daily\_vaccinations','total\_vaccinations\_per\_hundred','daily\_vaccinations\_per\_million']

for c in list1:

  col=data[c]

  q1=col.quantile(0.25)

  q3=col.quantile(0.75)

  iqr=q3-q1

  lower=q1 - 1.5 \*iqr

  upper=q3 + 1.5\*iqr

  col[col<lower]=lower

  col[col>upper]=upper

  print(" outlier handling completed")

output:

outlier handling completed

outlier handling completed

outlier handling completed

outlier handling completed

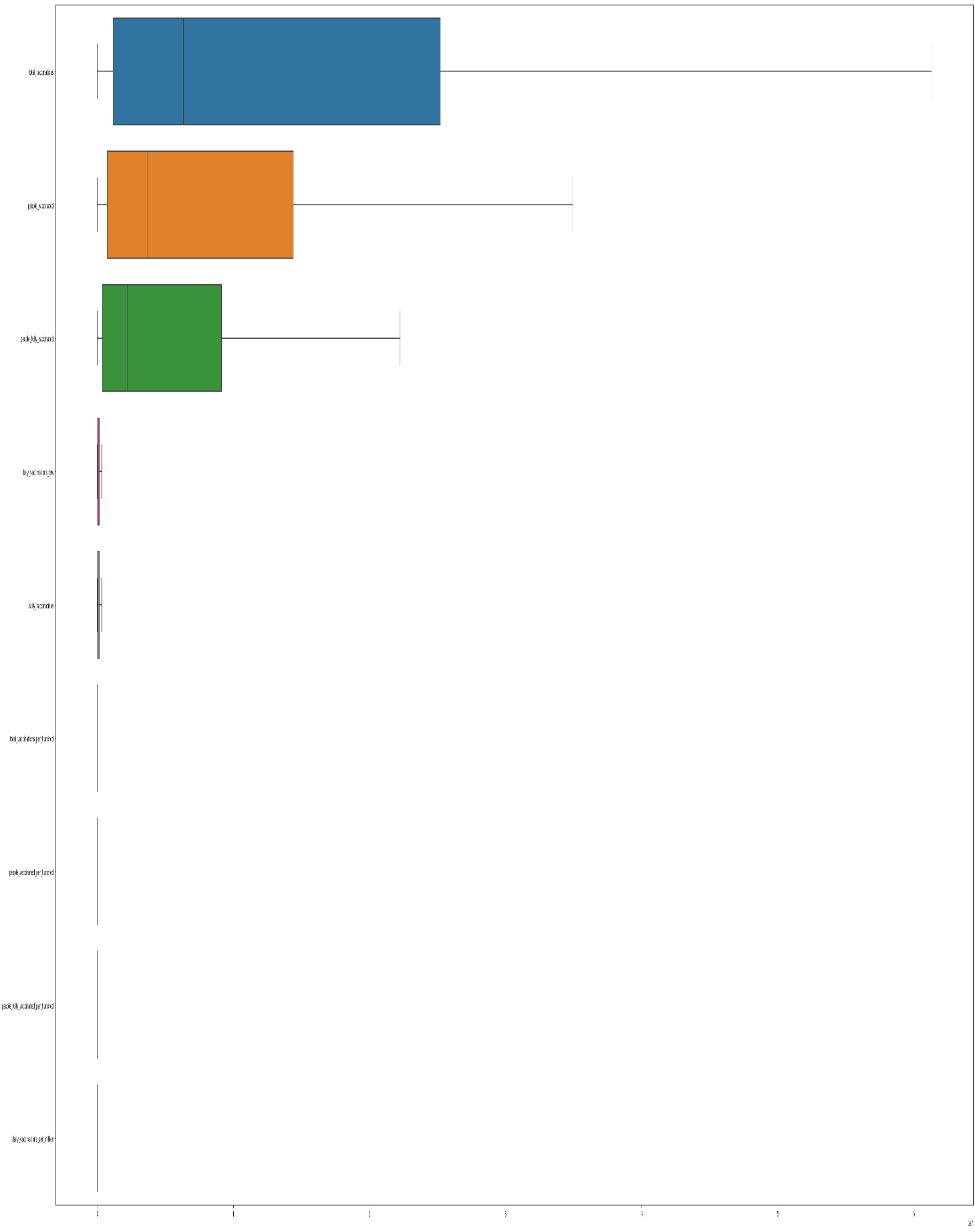
outlier handling completed

outlier handling completed

outlier handling completed

**After outlier handling**

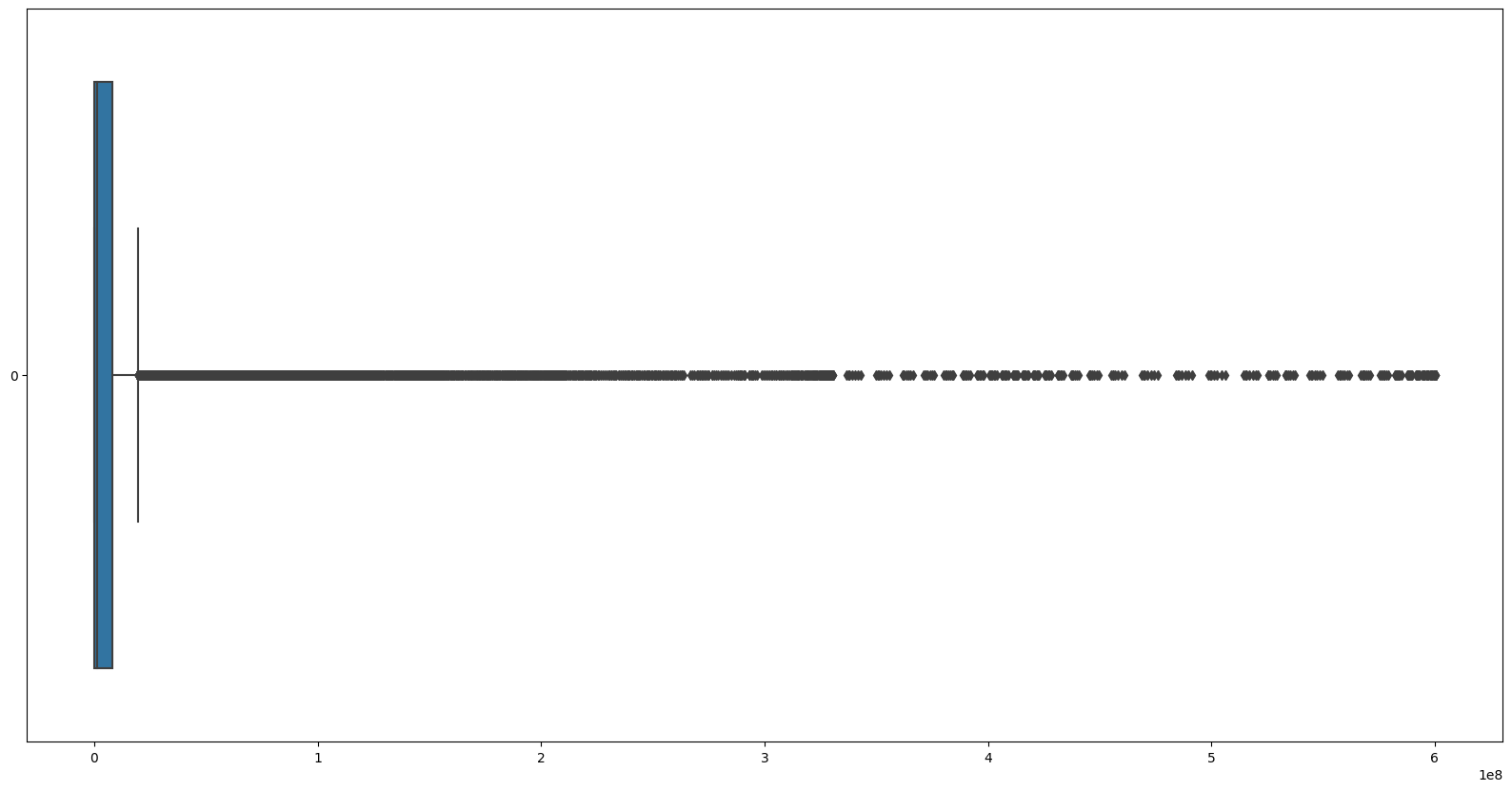
**Box polt become**



Making same method on the covid vaccinations by manufactures:

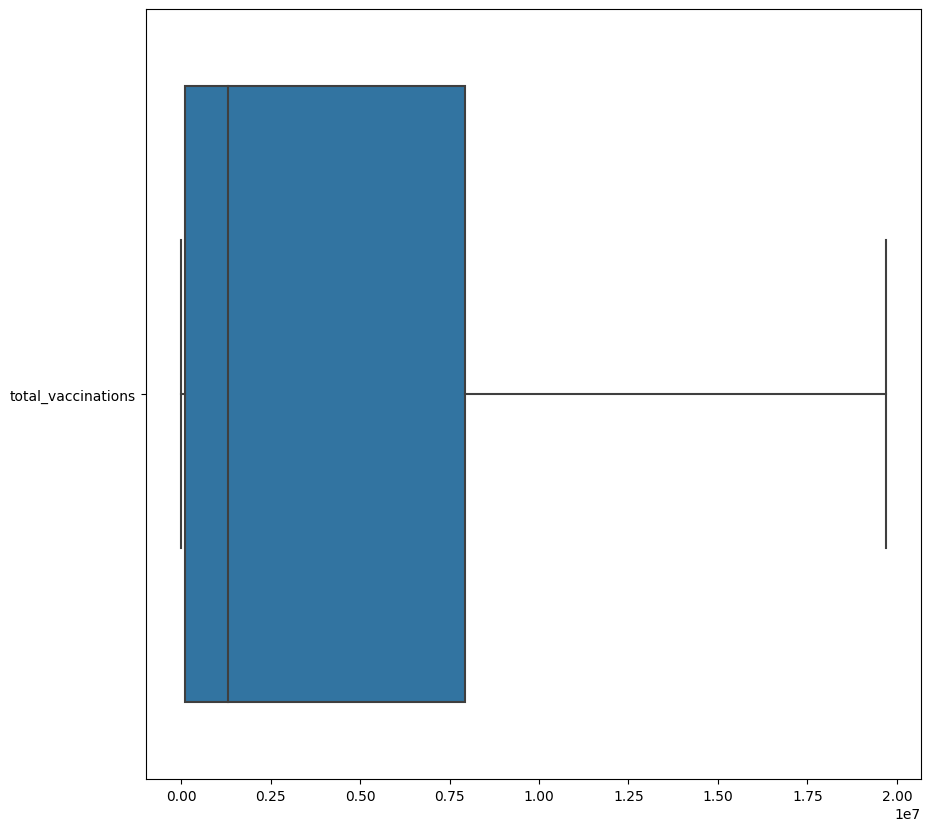
But in this dataset have only one numerical column.

**Boxplot for the total vaccination:**

****

Outlier on total\_vaccinations is 4544

After using same method to Handle the outlier and Box polt become

****

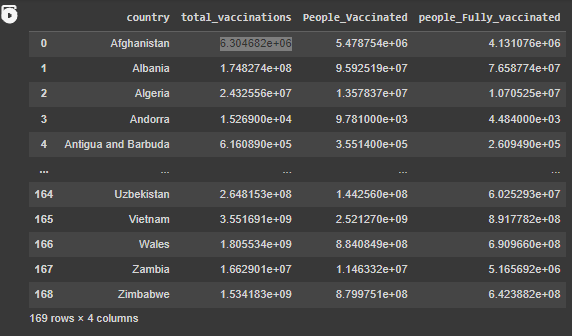
Outlier on total\_vaccinations is 0

Exploratory Data Analysis:

On covid vaccinations dataset

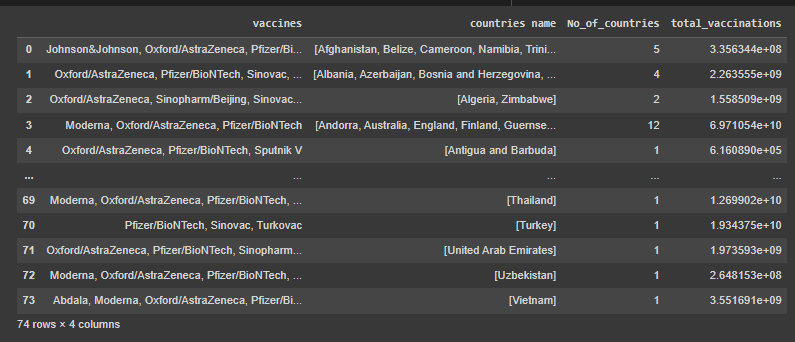
Number of countries that present in covid vaccinations dataset**:169**

Without date all countries total vaccinations people fully vaccinated,people vaccinated dataframe



There are 74 group vaccines used all over countries.

That vaccine groups used countries and its total vaccination

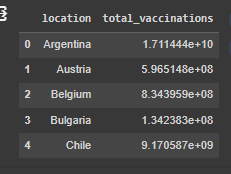


EDA on thehe manufacture vacciantions

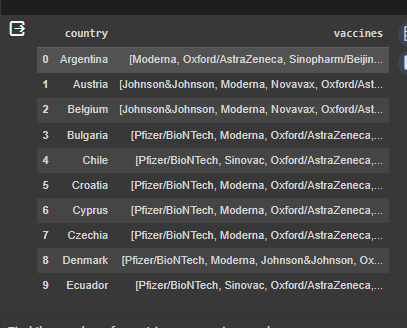
Number of countries in Data:43

Number of specified in dataset:10

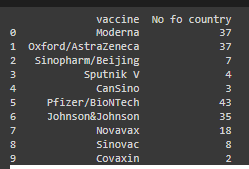
Countries total vacciantions first five entries,



Countries and its used vaccinations(first 10 entry):

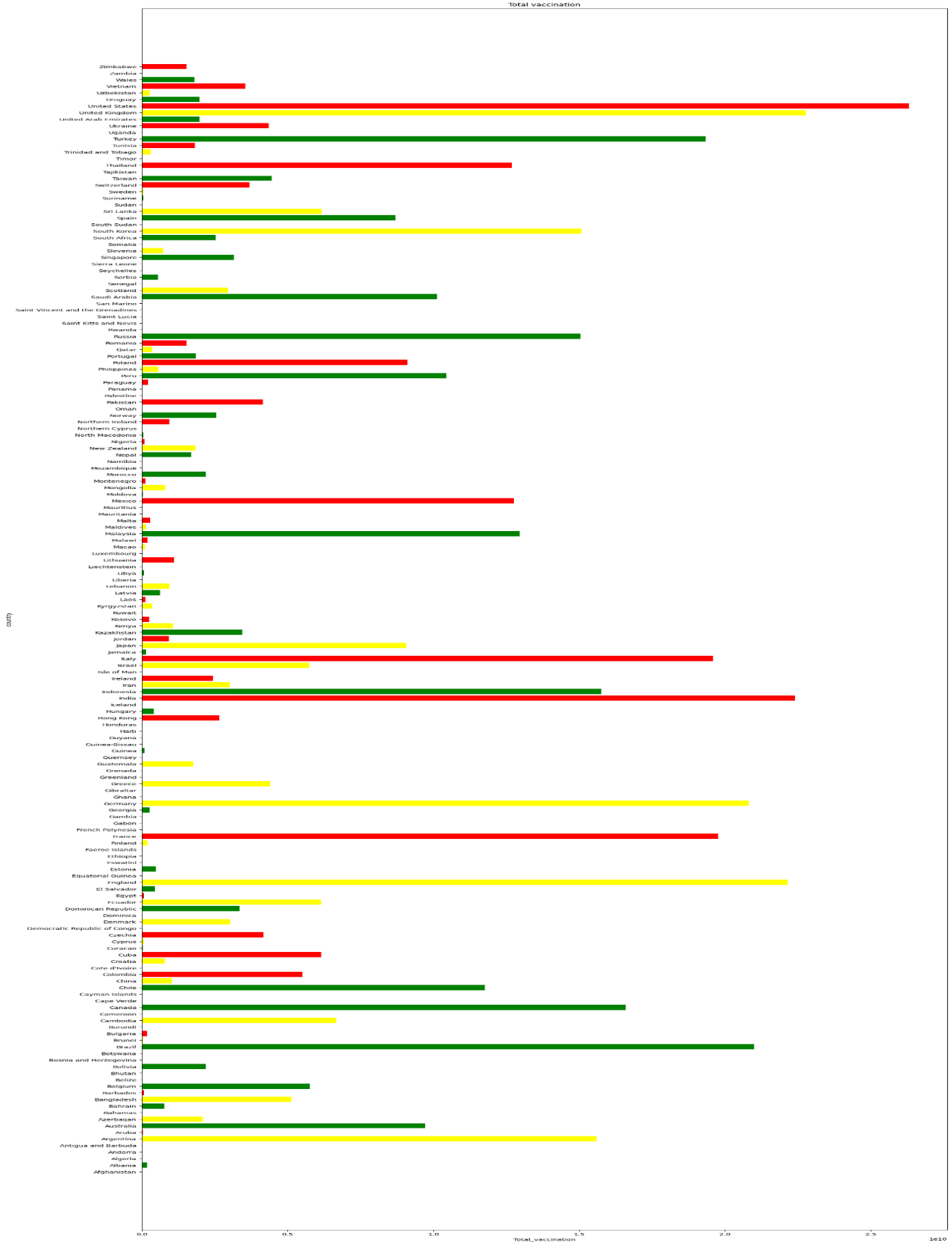


Vaccines and its number of countries used count:

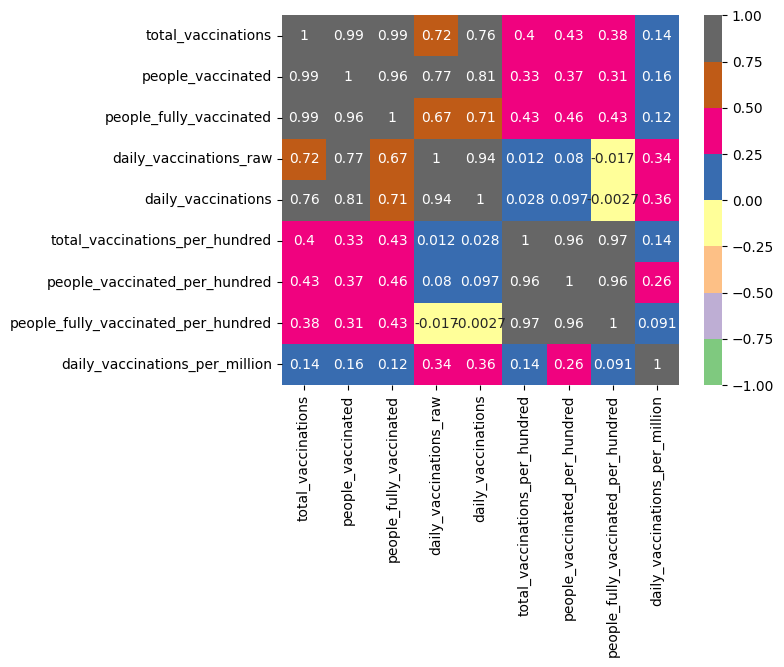


Visualization:

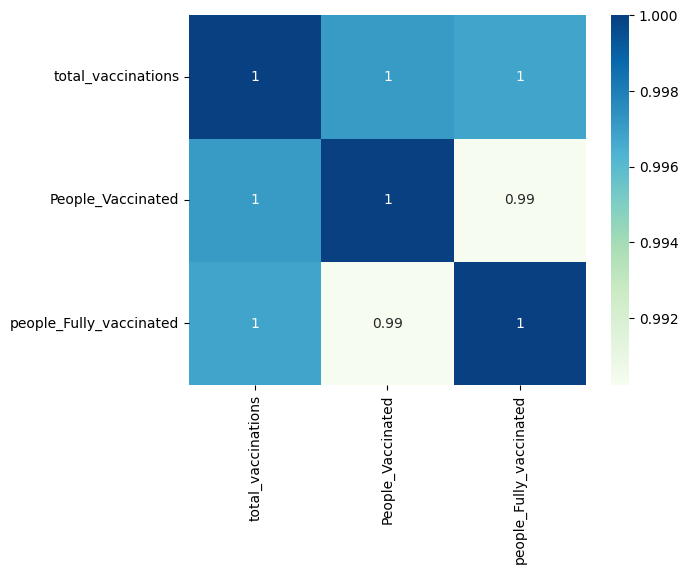
Country wise total vaccinations:



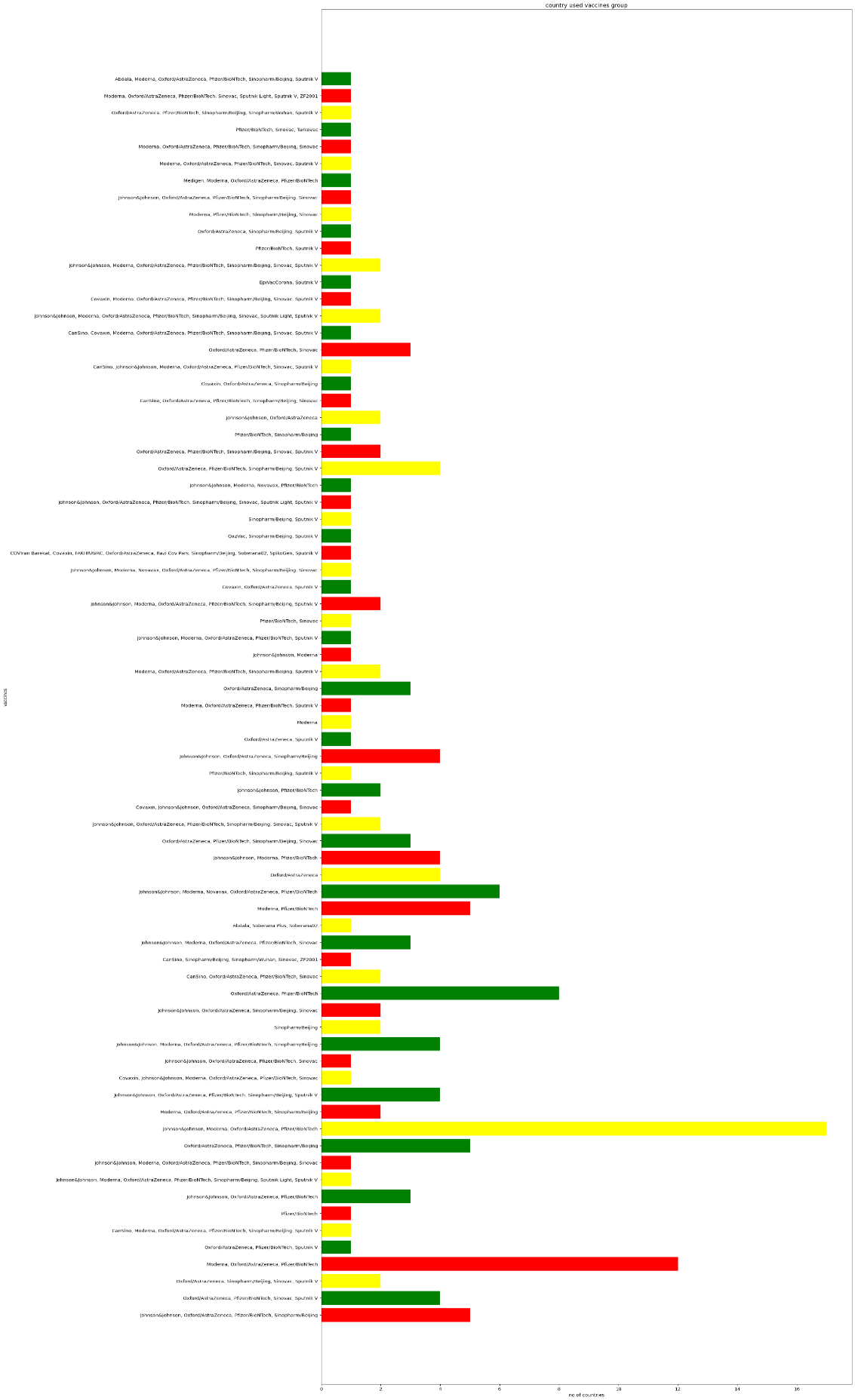
Country vacciantions Heatmap:



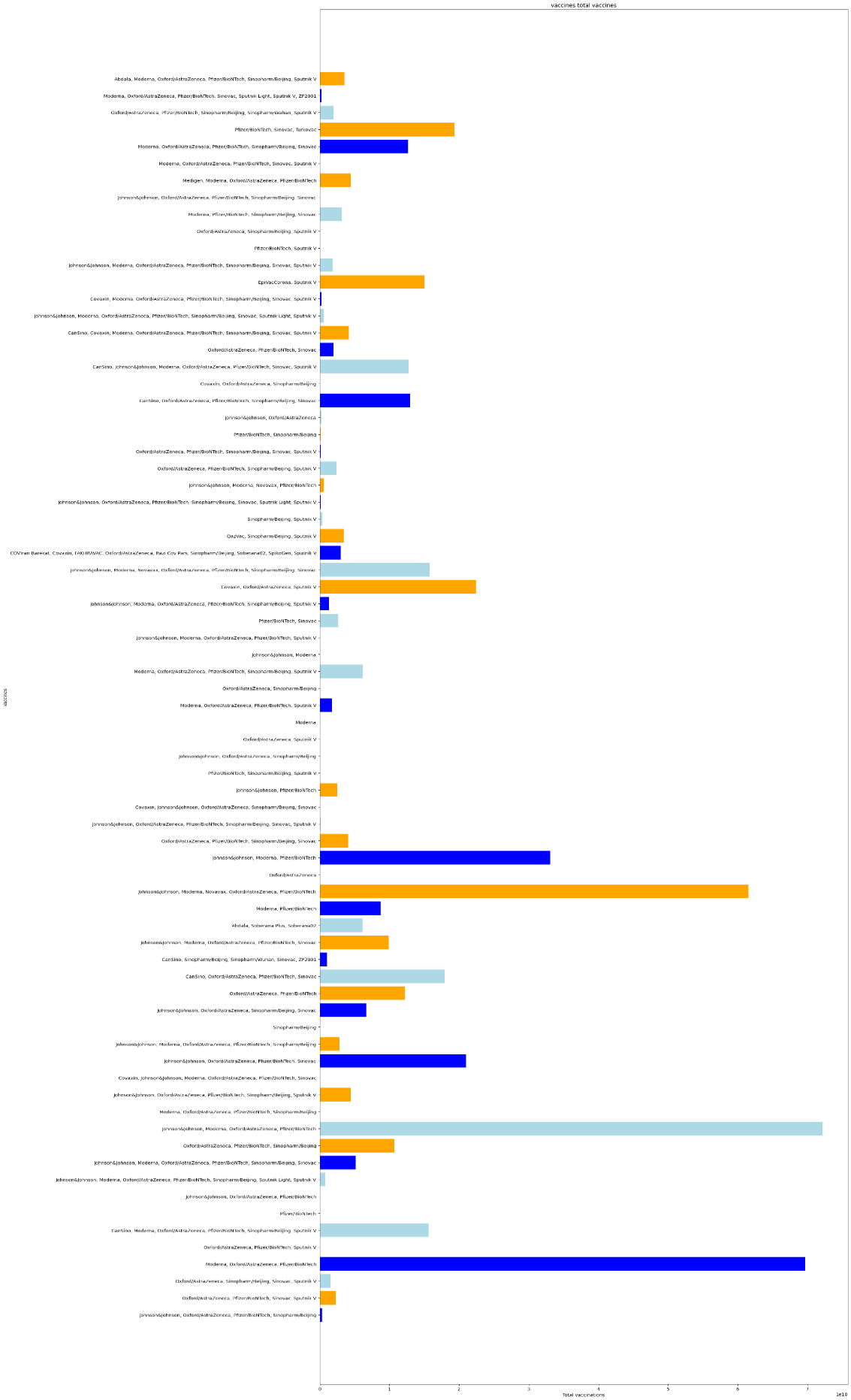
Heatmap for without date finded new dataframe

vaccine

**Vaccine group total vaccinations:**



**Vaccine and number countries used:**



**Machine Learning Technique:**

| **Description** | **Value** |
| --- | --- |
| **0** | Session id | 1265 |
| **1** | Target | total\_vaccinations |
| **2** | Target type | Regression |
| **3** | Original data shape | (30847, 15) |
| **4** | Transformed data shape | (30847, 15) |
| **5** | Transformed train set shape | (21592, 15) |
| **6** | Transformed test set shape | (9255, 15) |
| **7** | Numeric features | 8 |
| **8** | Categorical features | 6 |
| **9** | Preprocess | True |
| **10** | Imputation type | simple |
| **11** | Numeric imputation | mean |
| **12** | Categorical imputation | mode |
| **13** | Maximum one-hot encoding | 25 |
| **14** | Encoding method | None |
| **15** | Fold Generator | KFold |
| **16** | Fold Number | 10 |
| **17** | CPU Jobs | -1 |
| **18** | Use GPU | False |
| **19** | Log Experiment | False |
| **20** | Experiment Name | reg-default-name |
| **21** | USI | 19e4 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Model** | **MAE** | **MSE** | **RMSE** | **R2** | **RMSLE** | **MAPE** | **TT (Sec)** |
| **et** | Extra Trees Regressor | 34493.3769 | 24069774864.4843 | 149981.5199 | 1.0000 | 0.0181 | 0.0045 | 6.4600 |
| **rf** | Random Forest Regressor | 71519.2035 | 70872907863.2902 | 262394.3854 | 0.9999 | 0.0247 | 0.0083 | 19.0150 |
| **lightgbm** | Light Gradient Boosting Machine | 157914.9927 | 117269374154.8219 | 340029.8134 | 0.9998 | 0.2602 | 0.1678 | 1.6390 |
| **xgboost** | Extreme Gradient Boosting | 129275.2648 | 90804408729.6000 | 298858.0938 | 0.9998 | 0.1953 | 0.0758 | 0.6610 |
| **dt** | Decision Tree Regressor | 107110.1125 | 148027334504.1575 | 376612.9728 | 0.9997 | 0.0349 | 0.0139 | 0.4380 |
| **gbr** | Gradient Boosting Regressor | 356324.2147 | 500508546030.8246 | 706325.6443 | 0.9990 | 0.3767 | 0.4121 | 7.5050 |
| **knn** | K Neighbors Regressor | 357565.5188 | 611235836723.2000 | 775262.3375 | 0.9987 | 0.3740 | 0.7709 | 0.4700 |
| **br** | Bayesian Ridge | 741982.3081 | 1773742198117.8313 | 1330690.6318 | 0.9963 | 0.7277 | 1.2503 | 0.2340 |
| **en** | Elastic Net | 741609.0001 | 1773813552992.6714 | 1330715.3628 | 0.9963 | 0.7333 | 1.2617 | 0.5180 |
| **ridge** | Ridge Regression | 742051.8977 | 1773742366365.3560 | 1330691.0420 | 0.9963 | 0.7257 | 1.2484 | 0.2290 |
| **lasso** | Lasso Regression | 742051.9349 | 1773742368075.9712 | 1330691.0428 | 0.9963 | 0.7257 | 1.2484 | 0.8500 |
| **lr** | Linear Regression | 742131.6837 | 1773858148285.1355 | 1330736.3468 | 0.9963 | 0.7272 | 1.2484 | 0.7370 |
| **huber** | Huber Regressor | 772739.8265 | 2948895942392.7075 | 1715874.3694 | 0.9939 | 0.4038 | 0.1858 | 0.6010 |
| **par** | Passive Aggressive Regressor | 1250606.7672 | 4579474547023.5488 | 2076977.2645 | 0.9905 | 0.8340 | 2.7361 | 0.2520 |
| **ada** | AdaBoost Regressor | 1832934.0717 | 4750308947411.0488 | 2177918.6896 | 0.9901 | 1.3988 | 12.2781 | 2.5750 |
| **llar** | Lasso Least Angle Regression | 3754807.9698 | 48051847055468.2344 | 5819314.5749 | 0.9000 | 1.3153 | 14.9825 | 0.4540 |
| **lar** | Least Angle Regression | 7926486.0272 | 143453591747571.4688 | 11440649.6386 | 0.7023 | 1.8775 | 36.7547 | 0.3270 |
| **omp** | Orthogonal Matching Pursuit | 8221054.3718 | 164771343943652.4688 | 12828726.6462 | 0.6572 | 1.5177 | 24.3647 | 0.2190 |
| **dummy** | Dummy Regressor | 18021419.6000 | 481136631349248.0000 | 21932706.8000 | -0.0006 | 2.5470 | 86.8031 | 0.2430 |

|  | **MAE** | **MSE** | **RMSE** | **R2** | **RMSLE** | **MAPE** |
| --- | --- | --- | --- | --- | --- | --- |
| **Fold** |  |  |  |  |  |  |
| **0** | 29156.9509 | 13191665333.4341 | 114854.9752 | 1.0000 | 0.0160 | 0.0041 |
| **1** | 35629.6076 | 19401552298.0424 | 139289.4551 | 1.0000 | 0.0132 | 0.0042 |
| **2** | 42664.8630 | 65554078398.4964 | 256035.3069 | 0.9999 | 0.0182 | 0.0042 |
| **3** | 30525.3944 | 13048857808.6591 | 114231.5972 | 1.0000 | 0.0120 | 0.0041 |
| **4** | 35627.9274 | 21052843240.0043 | 145095.9794 | 1.0000 | 0.0155 | 0.0042 |
| **5** | 33808.5197 | 21612112593.8392 | 147010.5867 | 1.0000 | 0.0286 | 0.0056 |
| **6** | 33927.8836 | 23633581306.4297 | 153732.1739 | 1.0000 | 0.0357 | 0.0059 |
| **7** | 37107.5519 | 31043860933.5644 | 176192.6813 | 0.9999 | 0.0093 | 0.0037 |
| **8** | 32244.3700 | 17471985552.7957 | 132181.6385 | 1.0000 | 0.0198 | 0.0045 |
| **9** | 34240.7007 | 14687211179.5777 | 121190.8048 | 1.0000 | 0.0126 | 0.0044 |
| **Mean** | 34493.3769 | 24069774864.4843 | 149981.5199 | 1.0000 | 0.0181 | 0.0045 |
| **Std** | 3554.1291 | 14758275875.0401 | 39690.2828 | 0.0000 | 0.0078 | 0.0007 |

Final saved model

Transformation Pipeline and Model Successfully Saved

(Pipeline(memory=Memory(location=None),

steps=[('numerical\_imputer',

TransformerWrapper(include=['people\_vaccinated',

'people\_fully\_vaccinated',

'daily\_vaccinations\_raw',

'daily\_vaccinations',

'total\_vaccinations\_per\_hundred',

'people\_vaccinated\_per\_hundred',

'people\_fully\_vaccinated\_per\_hundred',

'daily\_vaccinations\_per\_million'],

transformer=SimpleImputer()...

transformer=SimpleImputer(strategy='most\_frequent'))),

('rest\_encoding',

TransformerWrapper(include=['country', 'iso\_code', 'date',

'vaccines', 'source\_name',

'source\_website'],

transformer=TargetEncoder(cols=['country',

'iso\_code',

'date',

'vaccines',

'source\_name',

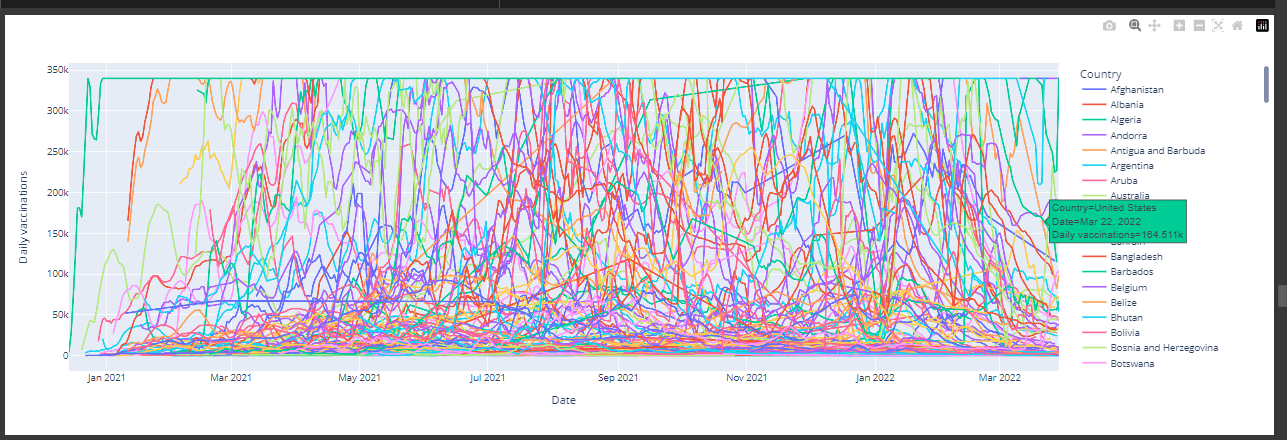
'source\_website'],

handle\_missing='return\_nan'))),

('trained\_model',

ExtraTreesRegressor(n\_jobs=-1, random\_state=8192))]),

Time Series Forecasting:



Providing Hidden Insights:

Some hidden insight I got,

* There among 225 country India has maximum People is fully vaccinated.
* Pitcarin has lowest vaccinated people

So maily focus on that country

In the start of covid Johnson &Johnson,

Moderna ,Pfizer has maximum vaccination all over the world

But Covaxine plays major role in the

covid-90 Vaccination in field.

That was produce by our Indian government.

It could be Assist for the pharmaceutical scientists

In the medicine filed.

**Conclusion:**

Thi**s** project could help vaccine distributers and Countries on their vaccination.it should be mor versatile when we create model for prediction and analysis of the data. At last

I find hidden insights should valuable for workers on the field and manufactures

OUR TEAM MEMBERS

S.SIVAKARTHICK

952421104051

https://github.com/siva-karthick-DA/DAC\_Phase1.git

V.VIMAL

95242104051

https://github.com/vimalv3/vimalv3.git

S.SHERJIN

952421104050