**Project Title: COVID Vaccines Analysis**

**PHASE 1 : PROJECT DEFINITION** This

Project mainly aims to find out the trend of the vaccinations around the world for the prevention of the Covid 19.

pandemic and how much has been achieved so far.

**Design Thinking:**

1. **Data Collection**

The COVID 19 pandemic caused due to the Corona virus devastated the world by causing several fatalities around the world. This virus originated in Wuhan, China in 2019 and was later spread throughout the world due to human contact in one way or the other. The disease showed symptoms as basic as mild fever and cold but also caused life threatening symptoms like breathing problems caused by damage to the lungs. As this virus was new to the world and there was no vaccine or cure to it at the initial period there were several deaths around the world. The countries around the world were forced to shut themselves to others in order to avoid the further spread of the virus and people were stuck inside their houses and faced many issues with their finances, mental health etc., and felt like animals in a cage. An effort was made to find a cure or vaccine by several health organizations to bring a stop to this pandemic.

In later stages of 2020 several experimental vaccines were developed and was administered to humans. The efforts were successful as the vaccines were helpful in reducing the affects the virus and even if people were infected, they were not in any life threating situation and escaped the illness having only minor symptoms.

Many countries later developed their own vaccines and also helped other countries without the resources by providing them with vaccines developed.

1. **Visualization**

**Insights –**

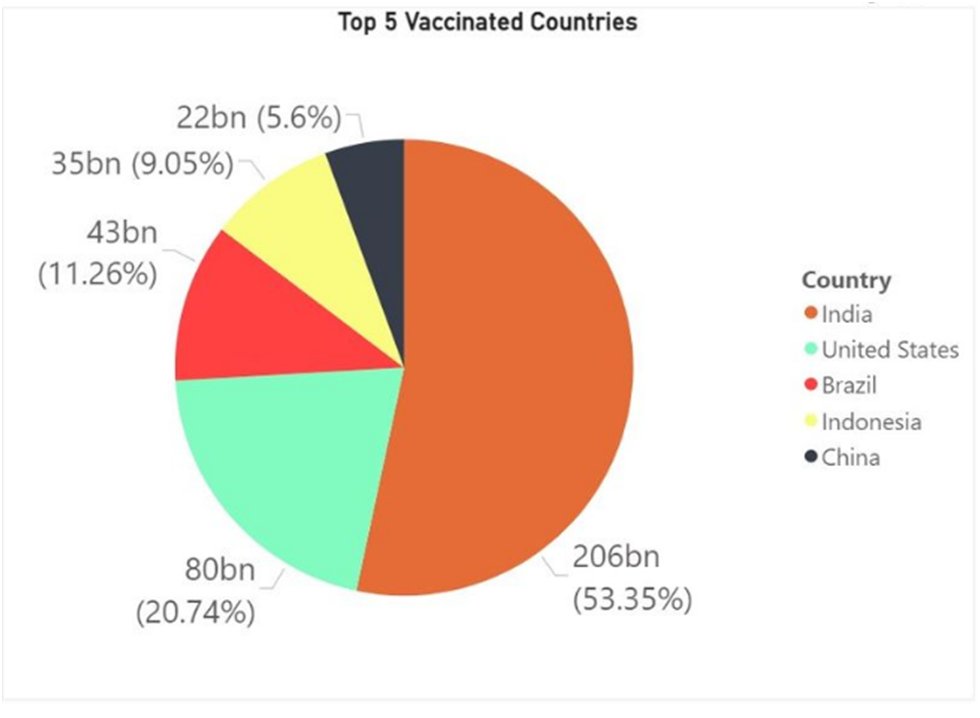
* + Here we analyzed the top 10 fully vaccinated countries in which India tops the list which indicates that people in the country where showing lots of interests to get vaccinated.
  + And also analyzed top 5 vaccinated countries here also India tops the list.
  + And then analyzed top 5 daily vaccinating countries and here China tops the list.
  + And also we analyse the sum of daily vaccinating details, fully vaccinating and vaccinating people details.
  + And our year wise analyse shows that 2021 was the peak year for every vaccination details.

**Recommendations –**

* + We should collect day to day reports and we should update our records daily to get more accurate details.
  + So that we can move forward with more vaccination to the right country which needs the most.

**Conclusions –**

In this dataset we came to know that the vaccination process in every country was going in good pace that indicates we can have control of this disease very soon all over the world.



**COVID VACCINE ANALYSIS**

# Phase 2 :

**Innovation**

Vaccines underpin our global health security by preventing and

controlling over 30 infectious diseases, reducing unnecessary hospitalizations and controlling infectious disease outbreaks**.**

**SYSTEM DESIGN**

✔ They consist of biological preparations that are capable of stimulating the immune system to confer protective immunity against a particular harmful pathogen/agent. Vaccine design and development have evolved through the years.

✔ Research and Discovery. In this early stage of vaccine development, researchers explore their idea for a potential vaccine. ...

✔ Proof of Concept. Before a vaccine can be tested in people, researchers study its ability to cause an immune response with small animals, like mice. ✔ Testing the Vaccine.

## MODULES:

### 1.Identification

COVID-19, invade our bodies, they attack and multiply. This invasion, called an infection, is what causes illness. Our immune system uses several tools to fight infection. Blood contains red cells, which carry oxygen to tissues and organs, and white or immune cells, which fight infection. Different types of white blood cells fight infection in different ways:

**Macrophages** are white blood cells that swallow up and digest germs and dead or dying cells. The macrophages leave behind parts of the invading germs called antigens. The body identifies antigens as dangerous and stimulates antibodies to attack them.

**B-lymphocytes** are defensive white blood cells. They produce antibodies that attack the pieces of the virus left behind by the macrophages.

**T-lymphocytes** are another type of defensive white blood cell. They attack cells in the body that have already been infected.

# SCREENING

**COVID-19** vaccine analysis and screening are essential processes in public health to ensure the safety, efficacy, and distribution of vaccines during a pandemic. These processes involve several key steps:

1. **Vaccine Development**: This phase involves extensive research, preclinical testing, and clinical trials to develop a vaccine candidate. Scientists and pharmaceutical companies work to identify antigens that can trigger an immune response without causing the disease. Multiple vaccine candidates are typically developed and tested.
2. **Clinical Trials**: COVID-19 vaccine candidates go through three phases of clinical trials to evaluate their safety and efficacy. These trials involve human participants and assess the vaccine's ability to protect against the virus while monitoring side effects and adverse reactions.
3. **Regulatory Approval**: Regulatory authorities, such as the U.S. Food and Drug Administration (FDA) and the European Medicines Agency (EMA), review the trial data and grant emergency use authorization or full approval for the vaccine. Stringent safety and efficacy criteria must be met.
4. **Vaccine Distribution and Storage**: Approved vaccines are manufactured and distributed to healthcare facilities and vaccination centers. Special attention is given to maintaining the cold chain, ensuring vaccines are stored at the appropriate temperatures to remain effective.
5. **Post-Market Surveillance**: Once vaccines are administered to the public, ongoing surveillance and monitoring are essential to identify and address any rare or unexpected side effects. This includes tracking and analyzing reported adverse events.
6. **Vaccine Effectiveness and Variants**: Continuous research and analysis assess the effectiveness of vaccines against new variants of the virus. This may lead to the development of booster shots or modified vaccines to combat emerging strains.
7. **Public Health Policy and Communication**: Governments and health organizations analyze vaccination data to make informed decisions about policies and recommendations, such as mask mandates, social distancing, and vaccination strategies. Communication strategies are crucial for addressing vaccine hesitancy and ensuring accurate information reaches the public.
8. **Equitable Distribution**: Ensuring that vaccines are distributed equitably, both within countries and globally, is a critical aspect of analysis and screening. Efforts are made to avoid vaccine inequality and to provide access to underserved populations.
9. **Global Collaboration**: International organizations, governments, and vaccine manufacturers collaborate to share data, research findings, and resources to combat the pandemic on a global scale.
10. **Vaccine Passports and Records**: Some countries have introduced digital or paper-based vaccine passports to verify vaccination status for travel or entry into certain venues. This involves the analysis of vaccination records and their security.

**COVID-19 vaccine analysis and screening** are ongoing processes that require the expertise of scientists, healthcare professionals, regulators, and public health officials. These efforts are essential for controlling the spread of the virus, preventing severe illness and death, and ultimately ending the pandemic.

# INCLUDED

The analysis of COVID-19 vaccines is a comprehensive process that involves rigorous evaluation, testing, and monitoring at various stages to ensure the safety and efficacy of the vaccines. Here are the key aspects of COVID-19 vaccine analysis:

1. **Preclinical Testing:**

* Before advancing to human trials, potential vaccine candidates undergo extensive preclinical testing in laboratories and on animals.
* This phase helps researchers identify suitable antigens, assess the vaccine's immunogenicity, and screen for potential side effects.

### Clinical Trials:

* COVID-19 vaccines undergo three phases of clinical trials:
* Phase 1: A small group of healthy volunteers is vaccinated to assess safety and determine dosage.
* Phase 2: A larger group receives the vaccine to further evaluate safety, dosage, and the immune response.

- Phase 3:Thousands of volunteers are enrolled to assess vaccine efficacy, monitor adverse reactions, and compare the vaccine's effectiveness to a placebo.

* These trials are randomized, double-blind, and placebo-controlled to minimize bias.

### Regulatory Approval

1. **Post-Market Surveillance**

### Efficacy Against Variants

1. **Public Health Data Analysis**

### Vaccine Equity Analysis

1. **Research and Data Sharing**

### Immunity Durability Analysis.

1. **Vaccine Hesitancy Studies**

### Vaccine Passport and Records

The analysis of COVID-19 vaccines is an ongoing and multidisciplinary effort that involves collaboration among scientists, healthcare professionals, regulators, and public health experts. It is crucial for the safety and effectiveness of vaccination programs and for controlling the spread of the virus.

**COVID Vaccines Analysis**

**DAC\_PHASE 3 Submission document**

Coronavirus disease 2019 (COVID-19), also known as the coronavirus, or COVID, is a contagious disease caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). The first known case was identified in Wuhan, China, in December 2019. The disease has since spread worldwide,

leading to an ongoing pandemic. And this data is all about the COVID 19 Vaccination Coverage across Health Unit district wise in Tamil Nadu.

1. Health Unit District
2. Achievement towards vaccination of 1st Dosage Covishield to HCW 3 Achievement towards vaccination of 2nd Dosage Covishield to HCW 4 Achievement towards vaccination of 1st Dosage Covishield to FLW 5 Achievement towards vaccination of 2nd Dosage Covishield to FLW
3. Achievement towards vaccination of 1st Dosage Covishield to beneficiaries of 18 years and less than 44 years age group
4. Achievement towards vaccination of 2nd Dosage Covishield to beneficiaries of 18 years and less than 44 years age group
5. Achievement towards vaccination of 1st Dosage Covishield to beneficiaries of 45 years and less than 60 years age group with Comorbidities
6. Achievement towards vaccination of 2nd Dosage Covishield to beneficiaries of 45 years and less than 60 years age group with Comorbidities
7. Achievement towards vaccination of 1st Dosage Covishield to 60+ years beneficiaries with Comorbidities
8. Achievement towards vaccination of 2nd Dosage Covishield to 60+ years beneficiaries with Comorbidities
9. Total Achievement of vaccination to beneficiaries under 1st Dose of Covishield 13 Total Achievement of vaccination to beneficiaries under 2nd Dose of Covishield 14 Achievement towards vaccination of 1st Dosage Covaxin to HCW

15 Achievement towards vaccination of 2nd Dosage Covaxin to HCW 16 Achievement towards vaccination of 1st Dosage Covaxin to FLW 17 Achievement towards vaccination of 2nd Dosage Covaxin to FLW

1. Achievement towards vaccination of 1st Dosage Covaxin to beneficiaries of 18 years and less than 44 years age group
2. Achievement towards vaccination of 2nd Dosage Covaxin to beneficiaries of 18 years and less than 44 years age group
3. Achievement towards vaccination of 1st Dosage Covaxin to beneficiaries of 45 years and less than 60 years age group with Comorbidities
4. Achievement towards vaccination of 2nd Dosage Covaxin to beneficiaries of 45 years and less than 60 years age group with Comorbidities
5. Achievement towards vaccination of 1st Dosage Covaxin to 60+ years beneficiaries with Comorbidities
6. Achievement towards vaccination of 2nd Dosage Covaxin to 60+ years beneficiaries with Comorbidities
7. Total Achievement of vaccination to beneficiaries under 1st Dose of Covaxin 25 Total Achievement of vaccination to beneficiaries under 2nd Dose of Covaxin

26 Total Achievement towards vaccination of 1st Dosage Covishield and Covaxin to HCW 27 Total Achievement towards vaccination of 2nd Dosage Covishield and Covaxin to HCW 28 Total Achievement towards vaccination of 1st Dosage Covishield and Covaxin to FLW 29 Total Achievement towards vaccination of 2nd Dosage Covishield and Covaxin to FLW

30 Total Achievement towards vaccination of 1st Dosage Covishield and Covaxin to beneficiaries of 18 years and less than 44 years age group 31 Total Achievement towards vaccination of 2nd Dosage Covishield and Covaxin to beneficiaries of 18 years and less than 44 years age group

1. Total Achievement towards vaccination of 1st Dosage Covishield and Covaxin to beneficiaries of 45 years and less than 60 years age group with Comorbidities
2. Total Achievement towards vaccination of 2nd Dosage Covishield and Covaxin to beneficiaries of 45 years and less than 60 years age group with Comorbidities
3. Total Achievement towards vaccination of 1st Dosage Covishield and Covaxin to 60+ years beneficiaries with Comorbidities
4. Total Achievement towards vaccination of 2nd Dosage Covishield and Covaxin to 60+ years beneficiaries with Comorbidities
5. Total Achievement towards vaccination to beneficiaries under 1st Dose of Covishield and Covaxin
6. Total Achievement towards vaccination to beneficiaries under 2nd Dose of Covishield and Covaxin
7. Total Achievement towards vaccination of Covishield and Covaxin (1st and 2nd Dose)

libraries are needed:

import pandas as pd # to import the dataset import numpy as np # to handle matrices import matplotlib.pyplot as plt # to plot

dataset = pd.read\_csv("country\_vaccinations.csv")

dataset.head(10) # we check the first 10 rows of our dataset

dataset.columns # we read the column namesIndex(['country', 'iso\_code', 'date', 'total\_vaccinations',

'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',

'vaccines', 'source\_name', 'source\_website'], dtype='object')

As i’m planning to predict the **total vaccinations (y)** based on the other **features of the dataset (x)**, where x is my independent variable (in this case, more than one) and y is the dependent variable.

x\_df = dataset[['country', 'date','daily\_vaccinations\_raw',

'daily\_vaccinations','total\_vaccinations\_per\_hundred', 'people\_vaccinated\_per\_hundred',

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

'daily\_vaccinations\_per\_million','vaccines']] # independent variables y\_df = dataset[['total\_vaccinations']] # dependent variable

x = x\_df.values # as array

y = y\_df.values # as arrayx[0:10,:] # we check the first 10 rows of our array xarray([['Albania', '2021-01-10', nan, nan, 0.0, 0.0, nan, nan,

'Pfizer/BioNTech'],

['Albania', '2021-01-11', nan, 64.0, nan, nan, nan, 22.0,

'Pfizer/BioNTech'],

['Albania', '2021-01-12', nan, 64.0, 0.0, 0.0, nan, 22.0,

'Pfizer/BioNTech'],

['Albania', '2021-01-13', 60.0, 63.0, 0.01, 0.01, nan, 22.0,

'Pfizer/BioNTech'],

['Albania', '2021-01-14', 78.0, 66.0, 0.01, 0.01, nan, 23.0,

'Pfizer/BioNTech'],

['Albania', '2021-01-15', 42.0, 62.0, 0.01, 0.01, nan, 22.0,

'Pfizer/BioNTech'],

['Albania', '2021-01-16', 61.0, 62.0, 0.01, 0.01, nan, 22.0,

'Pfizer/BioNTech'],

['Albania', '2021-01-17', 36.0, 58.0, 0.01, 0.01, nan, 20.0,

'Pfizer/BioNTech'],

['Albania', '2021-01-18', 42.0, 55.0, 0.02, 0.02, nan, 19.0,

'Pfizer/BioNTech'],

['Albania', '2021-01-19', 36.0, 51.0, 0.02, 0.02, nan, 18.0,

'Pfizer/BioNTech']], dtype=object)y[0:10,:] # we check the first 10 rows of our array yarray([[ 0.],

[ nan],

[128.],

[188.],

[266.],

[308.],

[369.],

[405.],

[447.],

[483.]])

**Taking care of missing data**

We notice that the dataset has some ‘nan’ values, which we can not use to train and test our model, so **we need to normalize** them, although deleting them is also an option, we will replace them using the SimpleImputer class.

fromsklearn.impute import SimpleImputer # importing the SimpleImputer class that let's us replace the missing values

# with the average

of the column

imputer

strategy="mean")

=

SimpleImputer(missing\_values=

np.nan,

Applied to x:

imputer.fit(X = x[:, 2:8]) # expects only the numerical values columns, not the category ones

x[:,2:8] = imputer.transform(x[:,2:8]) # we replace the missing values on the original array

x[0:10,:]array([['Albania', '2021-01-10', 74045.48985801217,

57184.43567251462,

0.0, 0.0, 1.7913693270735522, 2284.0280701754386,

'Pfizer/BioNTech'],

['Albania', '2021-01-11', 74045.48985801217, 64.0,

6.074087996582658, 5.177969072164948,

1.7913693270735522, 22.0,

'Pfizer/BioNTech'],

['Albania', '2021-01-12', 74045.48985801217, 64.0, 0.0,

0.0,

1.7913693270735522, 22.0, 'Pfizer/BioNTech'],

['Albania', '2021-01-13', 60.0, 63.0, 0.01, 0.01,

1.7913693270735522, 22.0, 'Pfizer/BioNTech'],

['Albania', '2021-01-14', 78.0, 66.0, 0.01, 0.01,

1.7913693270735522, 23.0, 'Pfizer/BioNTech'],

['Albania', '2021-01-15', 42.0, 62.0, 0.01, 0.01,

1.7913693270735522, 22.0, 'Pfizer/BioNTech'],

['Albania', '2021-01-16', 61.0, 62.0, 0.01, 0.01,

1.7913693270735522, 22.0, 'Pfizer/BioNTech'],

['Albania', '2021-01-17', 36.0, 58.0, 0.01, 0.01,

1.7913693270735522, 20.0, 'Pfizer/BioNTech'],

['Albania', '2021-01-18', 42.0, 55.0, 0.02, 0.02,

1.7913693270735522, 19.0, 'Pfizer/BioNTech'],

['Albania', '2021-01-19', 36.0, 51.0, 0.02, 0.02,

1.7913693270735522, 18.0, 'Pfizer/BioNTech']],

dtype=object)

Applied to y:

imputer.fit(X = y) # expects only the numerical values columns, not the category ones

y = imputer.transform(y) # we replace the missing values on the original array

y[0:10,:]array([[0.00000000e+00], [1.48631876e+06], [1.28000000e+02], [1.88000000e+02], [2.66000000e+02], [3.08000000e+02], [3.69000000e+02], [4.05000000e+02], [4.47000000e+02], [4.83000000e+02]])

#### Encoding categorical data

In order to train our models with categorical data, we need to **encode** the categories into “dummy variables”, this needs to be done because we can not train regression models directly with strings.

x\_df

=

pd.DataFrame(x,

columns

=

['country',

'date','daily\_vaccinations\_raw', 'daily\_vaccinations','total\_vaccinations\_per\_hundred', 'people\_vaccinated\_per\_hundred','people\_fully\_vaccinated\_per\_h undred', 'daily\_vaccinations\_per\_million','vaccines'])

x\_df

x\_df\_nodummies = x\_df

x\_df = pd.get\_dummies(x\_df, columns=["date","country",

"vaccines"],

x\_df has the

prefix=["date","country",

"vaccines"]) # now,

dummy variablesx\_df # we verify th structure with

the dummy variables

x = x\_df.values # convertind the data frame into an array xarray([[74045.48985801217, 57184.43567251462, 0.0, ..., 0,

0, 0],

[74045.48985801217, 64.0, 6.074087996582658, ..., 0, 0,

0],

[74045.48985801217, 64.0, 0.0, ..., 0, 0, 0],

...,

[17161.0, 23033.0, 26.02, ..., 0, 0, 0],

[21636.0, 22012.0, 26.7, ..., 0, 0, 0],

[22523.0, 20649.0, 27.42, ..., 0,

0,

0]],

dtype=object)yarray([[0.00000000e+00], [1.48631876e+06], [1.28000000e+02],

..., [8.20339000e+05], [8.41975000e+05],

[8.64498000e+05]])

#### Splitting the dataset into the Training set and Test set

Let’s create our train and test arrays with 20% of the data corresponding to the test array.

fromsklearn.model\_selection import train\_test\_split

x\_train, x\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size = 0.2, random\_state = 1)len(x\_train) # verifying its size2844len(x\_test) # verifying its size711

**Regression Model Application**

1. Multiple Linear Regression

**Training the Multiple Linear Regression model on the Training set**

fromsklearn.linear\_model import LinearRegression

regressor = LinearRegression()

regressor.fit(x\_train, y\_train)LinearRegression()

#### Predicting the Test set results

We test the trained model with the test data and concatenate it in an array for easy comparison

y\_pred = regressor.predict(x\_test) np.set\_printoptions(precision = 0)

print(np.concatenate((y\_pred.reshape(len(y\_pred),1), y\_test.reshape(len(y\_test),1)),1))[[ 1.e+06 3.e+04]

[-8.e+05 0.e+00] [ 5.e+06 5.e+06]

...

[-1.e+05 9.e+04] [ 2.e+07 2.e+07] [-2.e+05 2.e+04]]

#### Evaluating the Model Performance

We obtain an r squared of 0.82, meaning a fit of 82%

fromsklearn.metrics import r2\_score r2\_score(y\_test, y\_pred)0.820361971864213

#### Polynomial Regression

**Training the Polynomial Regression model on the Training set**

from sklearn.preprocessing import PolynomialFeatures from sklearn.linear\_model import LinearRegression

poly\_reg = PolynomialFeatures(degree = 2) x\_poly = poly\_reg.fit\_transform(x\_train) regressor = LinearRegression() regressor.fit(x\_poly, y\_train)LinearRegression()

#### Predicting the Test set results

y\_pred = regressor.predict(poly\_reg.transform(x\_test)) np.set\_printoptions(precision=2)

print(np.concatenate((y\_pred.reshape(len(y\_pred),1), y\_test.reshape(len(y\_test),1)),1))[[-5.50e+03 3.20e+04]

[-2.63e+08 0.00e+00] [ 7.68e+06 5.30e+06]

...

[ 5.99e+04 9.13e+04] [ 2.42e+07 2.27e+07] [ 2.23e+04 1.93e+04]]

#### Evaluating the Model Performance

fromsklearn.metrics import r2\_score r2\_score(y\_test, y\_pred)-275.73153181051356

This result means the performance of our model was really really bad.

#### Decision Tree Regression

**Training the model**

fromsklearn.tree import DecisionTreeRegressor regressor = DecisionTreeRegressor(random\_state = 0)

regressor.fit(x\_train, y\_train)DecisionTreeRegressor(random\_state=0)

#### Predicting the Test set results

y\_pred = regressor.predict(x\_test) np.set\_printoptions(precision = 2)

print(np.concatenate((y\_pred.reshape(len(y\_pred),1), y\_test.reshape(len(y\_test),1)),1))[[6.68e+02 3.20e+04]

[0.00e+00 0.00e+00] [5.19e+06 5.30e+06]

...

[8.80e+04 9.13e+04] [2.35e+07 2.27e+07] [2.20e+04 1.93e+04]]

#### Evaluating Model Performance

fromsklearn.metrics import r2\_score r2\_score(y\_test, y\_pred)0.9627743674208652

We notice that our Decision Tree Regression model has a fit of 96%, which is actually pretty good!

## Random Forest Regression

#### Training the Random Forest Regression model

fromsklearn.ensemble import RandomForestRegressor

regressor = RandomForestRegressor(n\_estimators = 10, random\_state = 0) regressor.fit(x\_train, y\_train)

#### Predicting the Test set results

y\_pred = regressor.predict(x\_test) np.set\_printoptions(precision=2)

print(np.concatenate((y\_pred.reshape(len(y\_pred),1), y\_test.reshape(len(y\_test),1)),1))[[1.53e+05 3.20e+04]

[0.00e+00 0.00e+00] [5.29e+06 5.30e+06]

...

[1.28e+05 9.13e+04] [2.15e+07 2.27e+07] [2.11e+04 1.93e+04]]

#### Evaluating the Model Performance

fromsklearn.metrics import r2\_score

r2\_score(y\_test, y\_pred)0.9659467525523365plt.plot\_date(x\_df\_nodummies.date, x\_df\_nodummies.vaccines)

plt.xticks(rotation ='vertical') plt.show()

Finally, we observe that the winner of our previously tested regression models is the **Random Forest Regression model**

**with a fit of 96.59%** which is really great! Now, you can be sure that this is the model that will perform better in case you want to make a prediction of the total vaccinations based on the features selected as predictors in x. **(Stay tuned for the part 2, where we will work a little bit more on the predictions and visualizations of the data)**. By the way, don’t forget the performance of each model changes for each dataset, so you actually do need to compare them to select the best one for each case (hint: you can also change their performance by optimizing the default hyperparameters).

**Covid vaccine analysis**

**Phase 4 :** Coding part

**Project Title :** Covid vaccine analysis with python programming

# Items in the dataset:

* Countries
* Dates
* Vaccines
* Total Vaccinations

# Desireddata to find:

* + Most commonly used vaccinesin countries
    - Average daily vaccination count in countries -Number of countries where vaccines are used
  + Choropleth map of the most used vaccine

INPUT:

data=pd.DataFrame(columns=['Count ry', 'Vaccine', 'Total\_vaccine'])

for country in df["location"].unique(): for vaccine in df["vaccine"].unique():

filtered\_data = df[(df['location'] == country) & (df['vaccine'] == vaccine)]

total\_count = filtered\_data['total\_vaccinations'].max()

data = pd.concat([data, pd.DataFrame({'Country': [country], 'Vaccine': [vaccine], 'Total\_vaccine': [total\_count]})], ignore\_index=True)

# SUB-INPUT:

data.head(10)

# Most commonlyused vaccines INPUT:

## data\_2=pd.DataFrame(columns=['Country',

**'Vaccine']) data["Total\_vaccine"] = pd.to\_numeric(data["Total\_vaccine"], errors="coerce")**

## for country in data["Country"].unique():

**new\_data = data[data["Country"] == country]**

## max\_vaccine = new\_data.loc[new\_data["Total\_vaccine"].idxm ax(), "Vaccine"]

**data\_2 = pd.concat([data\_2, pd.DataFrame({'Country': [countr y], 'Vaccine': [max\_vaccine]})], ignore\_index=True)**

## SUB-INPUT:

data\_2.head()

# OUTPUT:

|  |  |  |
| --- | --- | --- |
|  | Country | Vaccine |
| 0 | Argentina | Sinopharm/Beijing |
| 1 | Austria | Pfizer/BioNTech |
| 2 | Belgium | Pfizer/BioNTech |
| 3 | Bulgaria | Pfizer/BioNTech |
| **4** | **Chile** | **Sinovac** |

**INPUT:**

**data\_2["Vaccine"].value\_counts().plot(kind="bar", color=["Red","Gray","Gray","Gray"])**

**OUTPUT:**

**<Axes: >**

