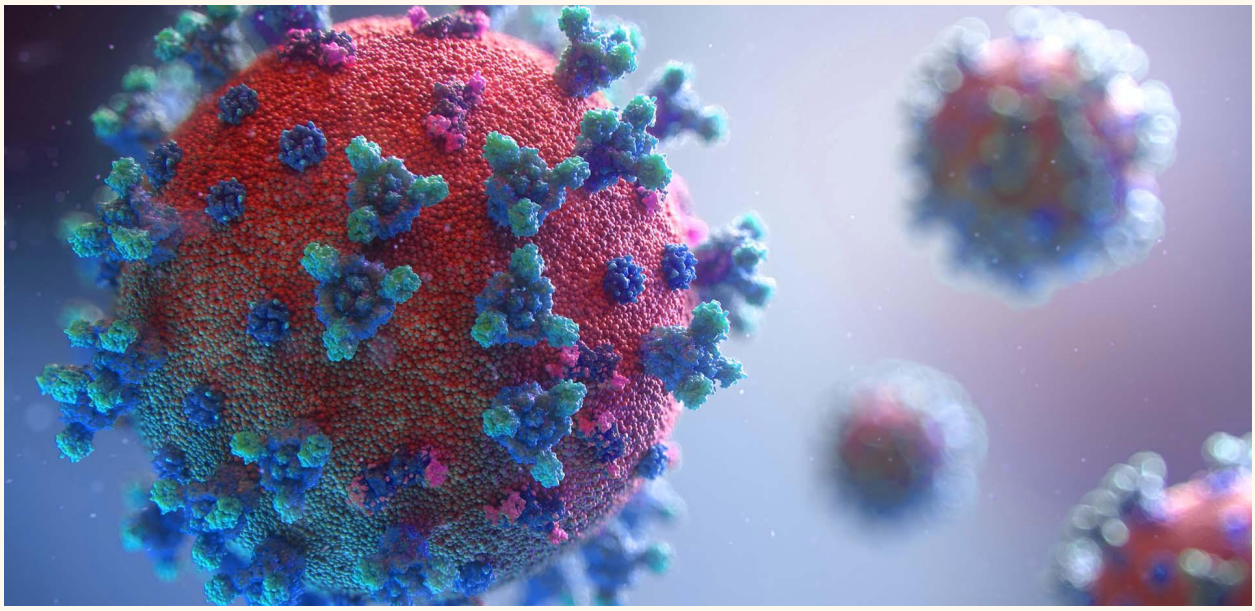
Phase 5
COVID VACCINES ANALYSIS

By,

Soundhara Srihari. S



INTRODUCTION

The COVID-19 outbreak, officially identified as the coronavirus disease outbreak, would be a continuing major worldwide public health problem of coronavirus disease 2019 (COVID-19) impacted by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). The epidemic happened in Wuhan, China, in December 2019. The World Health Organization classified COVID-19 as a Public Health Emergency of International Concern on January 30, 2020, and a pandemic on March 11, 2020. More than 167 million cases had occurred as of May 24, 2021, with more than 3.46 million authenticated fatalities attributed to COVID-19, making it one of the worst pandemics in history. We created this document based on the design thinking approach. This document is all about the insights and recommendations to be provided for the betterment of lives.

Objective

The aim is to conduct a Covid_vaccines analysis on the dataset given on Kaggle and to perform a time series analysis on the given dataset. We have done all required analysis and investigation to complete the process. The code was written in “Python” programming language over the platforms called Google Collab and IBM Cognos. The program is also attached to this document which is done by our team for Naan Mudhalvan project submission. We uploaded the “ipynb” file which contains the code and it was also uploaded to the GitHub account for the Evaluators. We were provided the dataset with the code file to the GitHub account.

Pre-processing

Data collected from the covid_vaccines 19 datasets have been pre-processed using various mathematical formulas, such as active cases, percentage of recovery rate, percentage of mortality, and week of days to generate features. There is a significant likelihood that the number of active topics has increased since some of the confirmed patients are now dead, and fewer new cases are being found. To calculate it, use Eq. (1). The recovery rate is the proportion of recovered instances, while the mortality rate is the percentage of death cases. Equations (2), and (3) display the formulas. The last parameter, the week of days, is calculated by importing the library named WEEKOFYEAR.

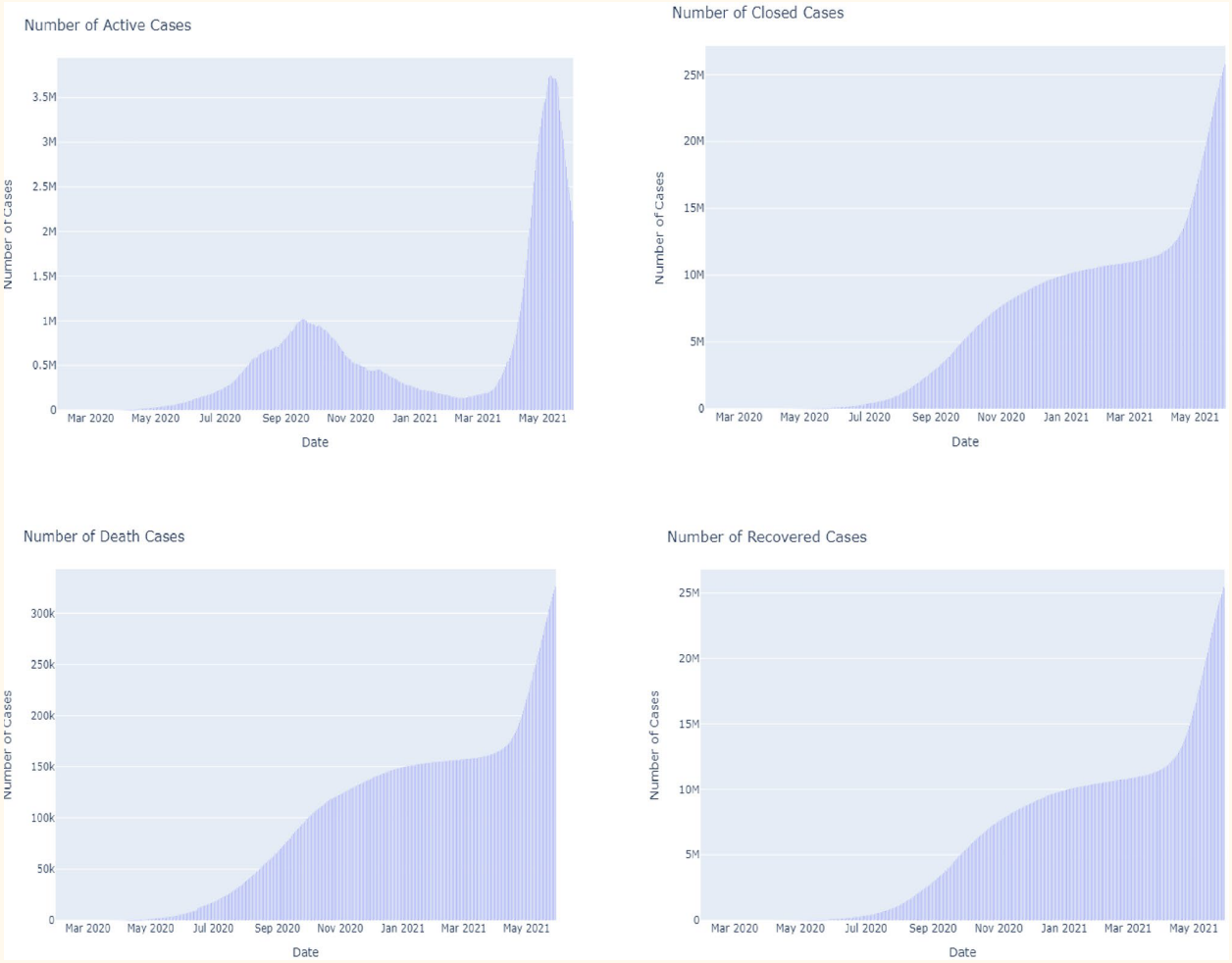
Formulae

Active cases = Total number of confirmed cases − (total number of recovered cases + total number of death cases, (1) Recovery rate = (2)Number of recovered cases/number of confirmed cases * 100, Mortality rate = (3)Number of death cases/Number of confirmed cases * 100.

Exploratory Data Analysis

Exploratory Data Analysis is a vital process that entails performing preliminary analyses of data to uncover patterns, identify anomalies, test hypotheses, and verify assumptions using summary statistics and graphical representations. Some of the critical steps in exploratory data analysis are importing the data set in which we will get two data frames; one consisting of the data to be trained and the other for predicting the target value, identifying the number of

features and columns, identifying the qualities or cues, identifying the data types of components, identifying the number of observations, checking if the dataset has empty cells or samples, identifying the number of empty cells by features or columns, and exploring categorical features. This work employed an exploratory analysis of ten different countries after pre-processing to assess their features via statistical graphs. The figures shown below depict the graphical analysis of active cases, death cases, closed points, and recovered cases that have been recorded from Jan 2020 to May 2021.



Feature Scaling

Normalizing the range of independent variables or features of data using feature scaling is a feature scaling approach. Min–Max scaling technique has been used to perform normalization on the parts obtained during data pre-processing. The Min–Max Normalization or Min–Max Scaling technique creates a scale that goes from 0 to 1 or from 1 to − 1. Deciding on a range of data to aim for relies on the type of data you are working with. Min–Max for the range[0,1] can be computed using Eq. (4)

Formulae

x =x − min(x)/max (x) − min(x) —---------(4)

After normalization, the data were split into two subsets: the training set, which would be used to assess machine learning methods, and the testing set, which would be used to evaluate deep learning techniques. It applies to issues involving classification or regression, as well as to any supervised learning technique. Following data partitioning, the first subset is utilized to fit the model; this is the training dataset. The second subset is used as an input element in the dataset supplied to the model, and predictions and comparisons to predicted values are performed. The test dataset is the second dataset. In a nutshell, the train data set is used to fit the machine learning model, while the test data set is used to verify the fit. The goal is to assess the performance of time series, machine learning, and deep learning models on new data. The most often used split percentages are as follows:

80% training, 20% testing.

67% training, 33% testing.

50% training, 50% testing.

Model selection

We have chosen the Time series model named the prophet model to perform analytics and the result visualization.

K Nearest Regressor

Non-parametric regression involves averaging nearby observations to determine if one or more independent variables are associated with a continuous result. For an analysis to be effective, the size of the neighborhood should be selected by the analyst. However, in some cases, it can be randomized to reduce the mean squared error. An algorithm that considers the K-nearest neighbor numerical objective is utilized to determine the average of the K target values. KNN regression and KNN classification both utilize the same distance functions. KNN regression uses the same distance functions as KNN classification. The formulae to compute K-nearest regressor are shown in Eqs. (8)–(10)

$$\text{Euclidean formula}: \sqrt{\sum_{i=1}^{k}\left(x_i - y_i\right)^2}, \tag{8}$$

$$\text{Manhattan formula}: \sum_{i=1}^{k}\left|x_i - y_i\right|, \tag{9}$$

$$\text{Minkowski formula}: \left[\sum_{i=1}^{k}\left(\left|x_i - y_i\right|\right)^q\right]^{\frac{1}{q}}. \tag{10}$$

Kernel Ridge Regressor

Bayesian Regressor

Bayesian Regressor is a regression approach that uses Bayesian inference to do statistical analysis. This method enables a natural process to persist in the presence of limited or poorly dispersed data. It generates predictions based on the posterior probability of all feasible regression weights. With Bayesian Linear Regression, the aim is not to choose the "best" model parameter but to estimate the distribution of model parameters [39]. It is demonstrated by Eq. (25)

P(y, X) = P(y, X) × P(X) / P(yX) ——————————————————————————(25)

Here, P(β|y, X) is the posterior probability distribution of the model parameters given the inputs and outputs. This is equal to the likelihood of the data, P(β|y, X), multiplied by the prior probability of the parameters and divided by a normalization constant.

Time Series Models: Facebook Prophet

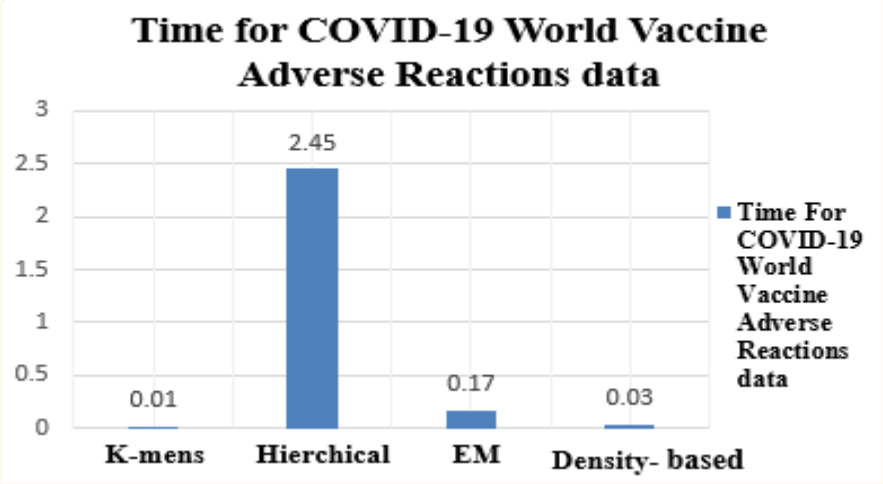
A forecasting approach based on an additive model known as a prophet is used to correlate nonlinear trends with seasonal and holiday impacts as well as yearly, weekly, and daily patterns. Time series with strong seasonal influences and extensive historical data spanning many seasons do well with this approach. The Prophet works well with outliers, which makes it resistant to data and trend shifts. The time series model is built on a prophet, and it is fast, fully automated, and very exact. The trend, seasonality, and holidays from our time series model, which we break down into three key components: trend, seasonality, and holidays. They are merged in Eq. (5) as follows:

$y(t) = g(t) + s(t) + h(t) + \in t$, —----------(5)

g(t): For modeling non-periodic changes in time series, a piecewise linear or logistic growth curve is used. s(t): changes on a regular basis (e.g., weekly/yearly seasonality).

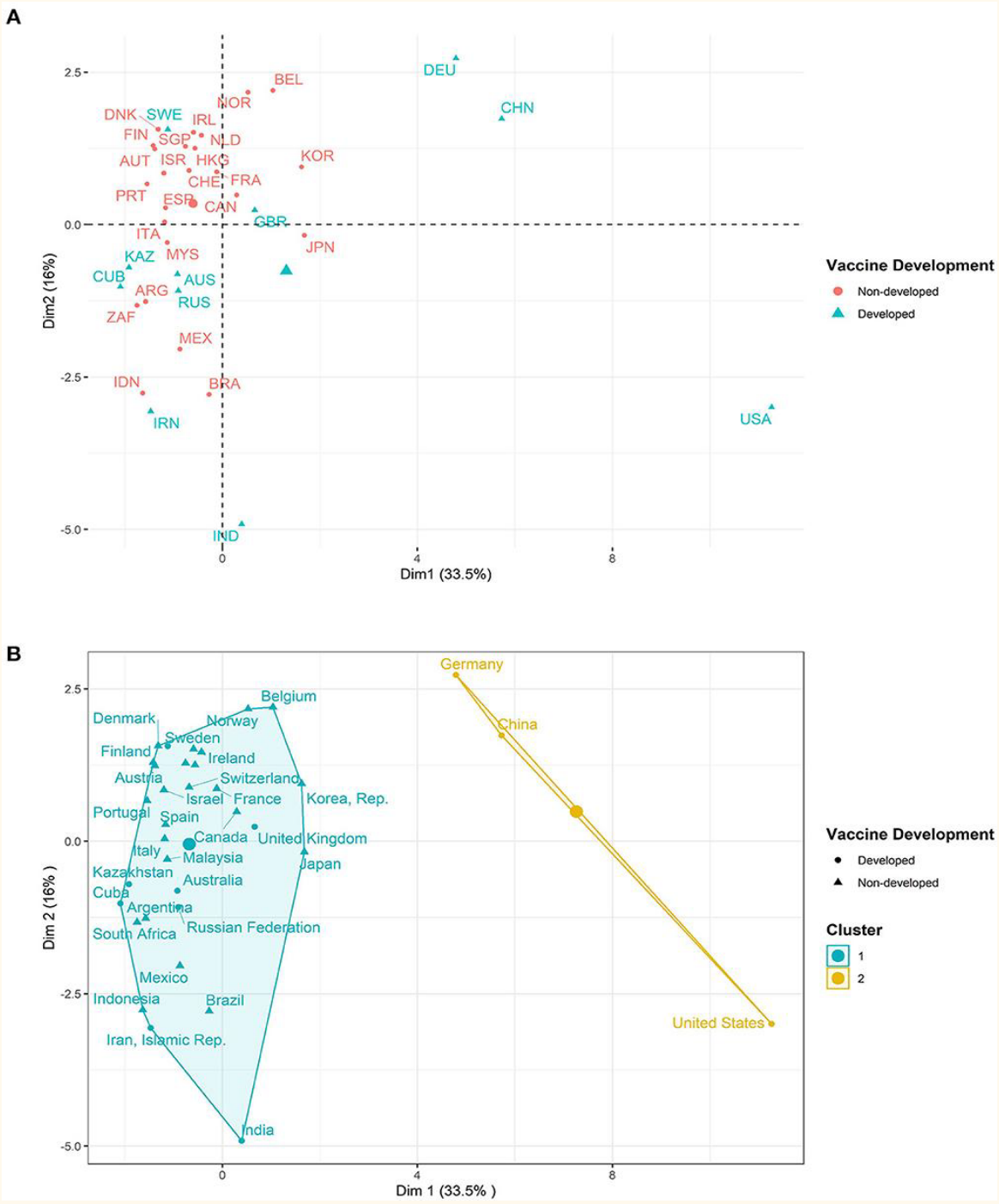
h(t): The impact of vacations (supplied by the user) on individuals with irregular schedules.

εt: The error term is used to account for any unforeseen changes that the model does not account for.



Statistical analysis

Statistical analysis is essential in verifying assumptions and demonstrating them to create a concrete conclusion about a study. This study focuses on the efficacy of vaccination over COVID-19 cases and COVID-19 deaths. The linear regression analysis will investigate the relevance of vaccinations, followed by polynomial and OLS regression models and SVM models. This will provide information about the effectiveness of 6 being immunized.


A
BEL
DEU
NOR
CHN
DNK
SWE
IRL
FIN
SGP
NLD
KOR
AUT
ISR
HKG
CHE
FRA
PRT
ESP
CAN
GBR
ITA
JPN
KAZ
MYS
CUB
AUS
ARG
RUS
ZAF
MEX
IDN
BRA
IRN
USA
IND
Dim2 (16%)
Dim1 (33.5%)
Vaccine Development
Non-developed
Developed
B
Germany
Belgium
China
Denmark
Norway
Sweden
Finland
Ireland
Austria
Switzerland
Israel
France
Korea, Rep.
Portugal
Spain
Canada
United Kingdom
Italy
Malaysia
Japan
Kazakhstan
Australia
Cuba
Argentina
Russian Federation
South Africa
Mexico
Indonesia
Brazil
United States
Iran, Islamic Rep.
India
Dim 2 (16%)
Dim 1 (33.5%)
Vaccine Development
Developed
Non-developed
Cluster
1
2

Code:

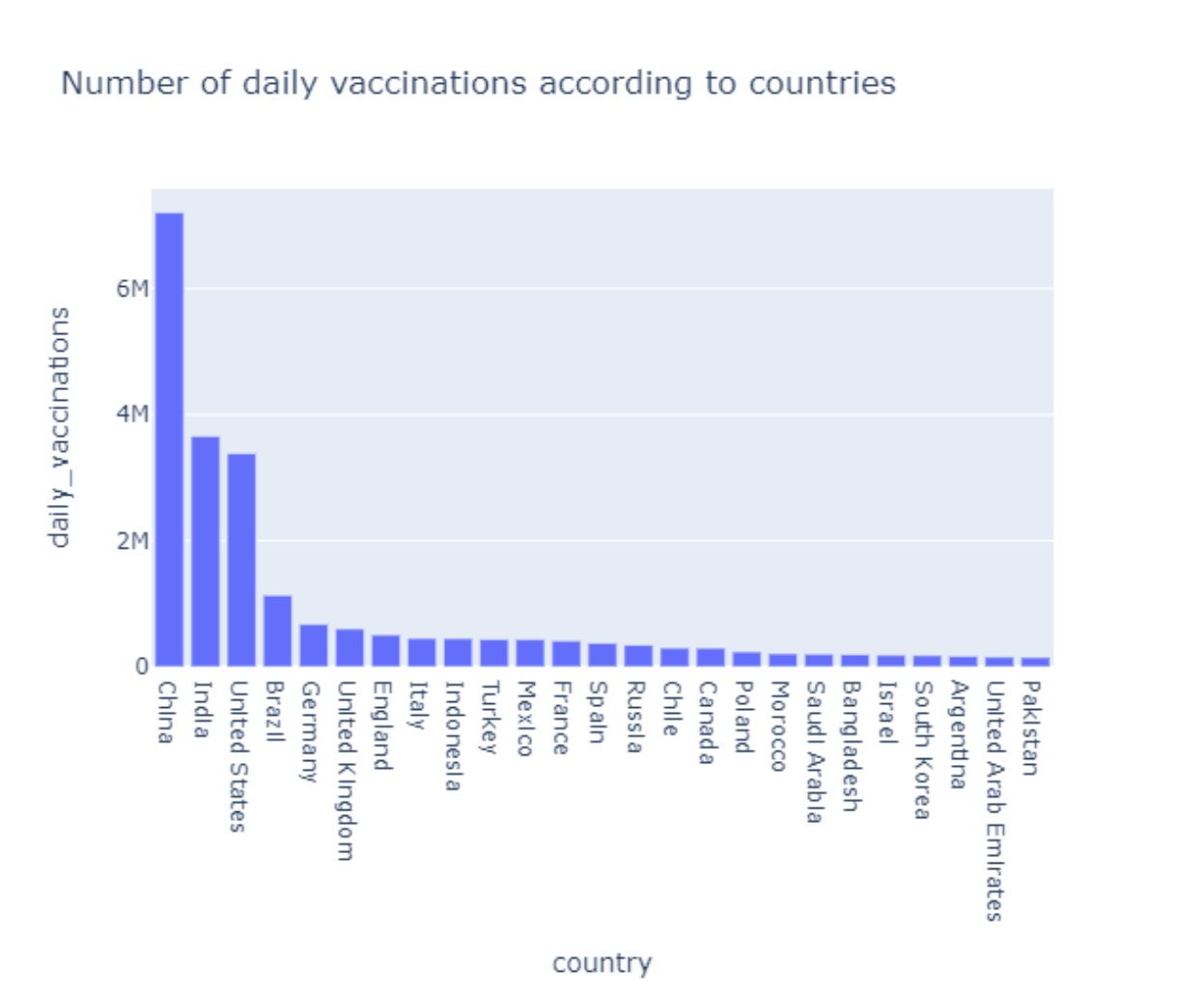
Importing Libraries:

```
import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

import matplotlib.pyplot as plt

import seaborn as sns

import plotly.express as px

from plotly.offline import download_plotlyjs,init_notebook_mode,plot,iplot

import plotly.graph_objects as go

import plotly.figure_factory as ff from plotly.colors

import n_colors from wordcloud import WordCloud,ImageColorGenerator init_notebook_mode(connected=True)

from plotly.subplots import make_subplots

from pywaffle import Waffle

import warnings

warnings.filterwarnings("ignore")

top10 = new_df['vaccines'].value_counts().nlargest(10) top10

data = dict(new_df['vaccines'].value_counts(normalize = True).nlargest(10)*100)

#dict(new_df['vaccines'].value_counts(normalize = True) * 100)

vaccine = ['Oxford/AstraZeneca', 'Moderna, Oxford/AstraZeneca, Pfizer/BioNTech', 'Oxford/AstraZeneca, Pfizer/BioNTech', 'Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech', 'Pfizer/BioNTech', 'Sputnik V', 'Oxford/AstraZeneca, Sinopharm/Beijing', 'Sinopharm/Beijing', 'Moderna, Pfizer/BioNTech', 'Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac']

fig = plt.figure(
```

```
rows=7,
columns=12,
FigureClass = Waffle,
values = data,
title={'label': 'Proportion of Vaccines', 'loc': 'center',
'fontsize':15},
colors=("#FF7F0E", "#00B5F7",
"#AB63FA","#00CC96","#E9967A","#F08080","#40E0D0","#DFFF00","#DE
3163","#6AFF00"),
labels=[f"{k} ({v:.2f}%)" for k, v in data.items()],
legend={'loc': 'lower left', 'bbox_to_anchor': (0, -0.4),
'ncol': 2, 'framealpha': 0},
figsize=(12, 9))
fig.show()
```

Observation:

- In a range of percentage of vaccines 28.44% used Oxford/AstraZeneca
- Oxford/AstraZeneca is the most used Vaccine
- Later Pfizer/BioNTech was the most used Vaccine and now it’s in 5th place also Oxford/AstraZeneca was not in the top 3 & now it’s in 1st place. Looks like Oxford/AstraZeneca works best among the vaccines




```
data = new_df[['country','total_vaccinations']].nlargest(25,'total_vaccinations')
fig = px.bar(data, x = 'country',y =
'total_vaccinations',title="Number of total vaccinations
according to countries",)
fig.show()
data =new_df[['country','daily_vaccinations']].nlargest(25,'daily_vaccinations')
fig = px.bar(data, x = 'country',y =
'daily_vaccinations',title="Number of daily vaccinations according to countries",)
fig.show()
```

```
Vaccine: Sputnik V
Used countries: ['Argentina', 'Russia']
----------------------------------------------------------------------
Vaccine: Pfizer/BioNTech
Used countries: ['Austria', 'Belgium', 'Chile', 'Costa Rica', 'Croatia', 'Cyprus', 'Estonia', 'Finland', 'France', 'Gibraltar',
'Greece', 'Hungary', 'Ireland', 'Israel', 'Italy', 'Kuwait', 'Latvia', 'Luxembourg', 'Malta', 'Mexico', 'Netherlands', 'Norwa
y', 'Oman', 'Poland', 'Portugal', 'Romania', 'Saudi Arabia', 'Serbia', 'Singapore', 'Slovakia', 'Slovenia', 'Sweden', 'Switzerl
and']
----------------------------------------------------------------------
Vaccine: Pfizer/BioNTech, Sinopharm
Used countries: ['Bahrain', 'United Arab Emirates']
----------------------------------------------------------------------
Vaccine: Sinovac
Used countries: ['Brazil', 'Turkey']
----------------------------------------------------------------------
Vaccine: Moderna, Pfizer/BioNTech
Used countries: ['Bulgaria', 'Canada', 'Czechia', 'Denmark', 'Germany', 'Iceland', 'Lithuania', 'Spain', 'United States']
----------------------------------------------------------------------
Vaccine: CNBG, Sinovac
Used countries: ['China']
----------------------------------------------------------------------
Vaccine: Covaxin, Covishield
Used countries: ['India']
----------------------------------------------------------------------
Vaccine: Sinopharm
Used countries: ['Seychelles']
----------------------------------------------------------------------
Vaccine: Oxford/AstraZeneca, Pfizer/BioNTech
Used countries: ['United Kingdom']
----------------------------------------------------------------------
```

Which vaccine is used by which Country?

vacc = new_df["vaccines"].unique()

for i in vacc:

c = list(new_df[new_df["vaccines"] == i]['country'])

print(f"Vaccine: {i}nUsed countries: {c}")

print('-'*70)

fig = px.choropleth(new_df,locations = 'country',locationmode =

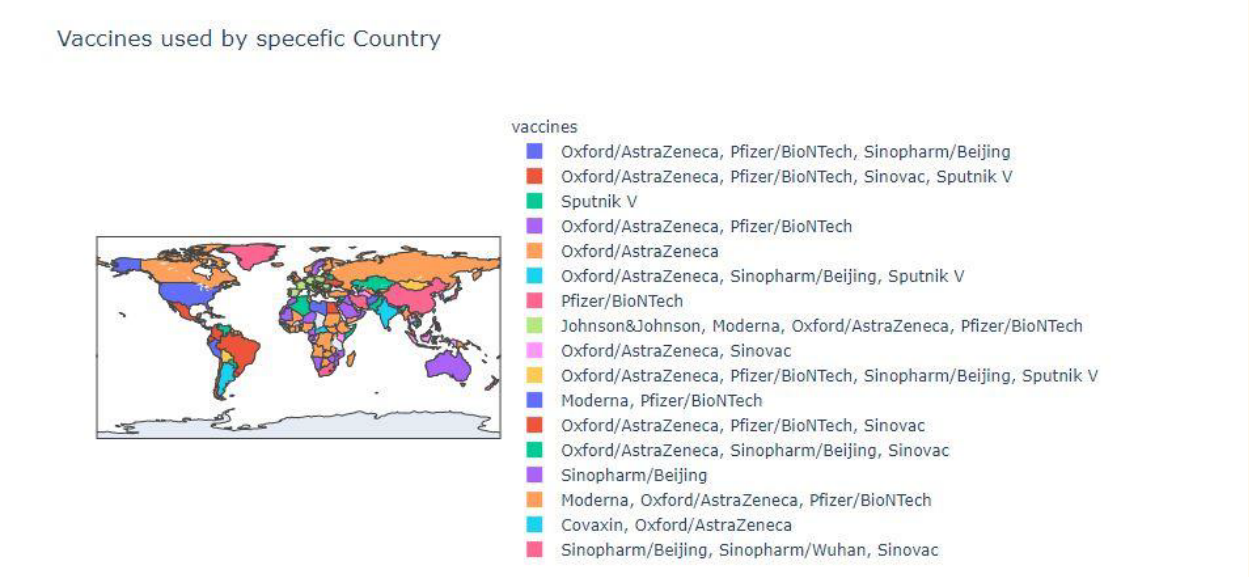
'country names',color = 'vaccines',

title = 'Vaccines used by specefic

Country',hover_data= ['total_vaccinations'])

fig.show()

Which Vaccine is Used the most?



vaccine = new_df["vaccines"].value_counts().reset_index()

vaccine.columns = ['Vaccines','Number of Country']

vaccine.nlargest(5,"Number of Country")

Oxford/AstraZeneca is being used by 60 Countries.

	Vaccines	Number of Country
0	Oxford/AstraZeneca	60
1	Moderna, Oxford/AstraZeneca, Pfizer/BioNTech	19
2	Johnson&Johnson, Moderna, Oxford/AstraZeneca, ...	14
3	Oxford/AstraZeneca, Pfizer/BioNTech	13
4	Pfizer/BioNTech	11

Total Vaccinations per country grouped by Vaccines:

fig = px.treemap(new_df,names = 'country',values =

'total_vaccinations',

path = ['vaccines','country'],

title="Total Vaccinations per country grouped

```
by Vaccines",

color_discrete_sequence

=px.colors.qualitative.Set1)

fig.show()


fig = go.Choropleth(locations = new_df["country"],locationmode =

'country names',

z = new_df['total_vaccinations'],

text= new_df['country'],colorbar =

dict(title= "Total Vaccinations"))

data = [fig] layout = go.Layout(title = 'Total Vaccinations per Country')

fig = dict(data = data,layout = layout) iplot(fig)
```

Daily Vaccinations per Countries:

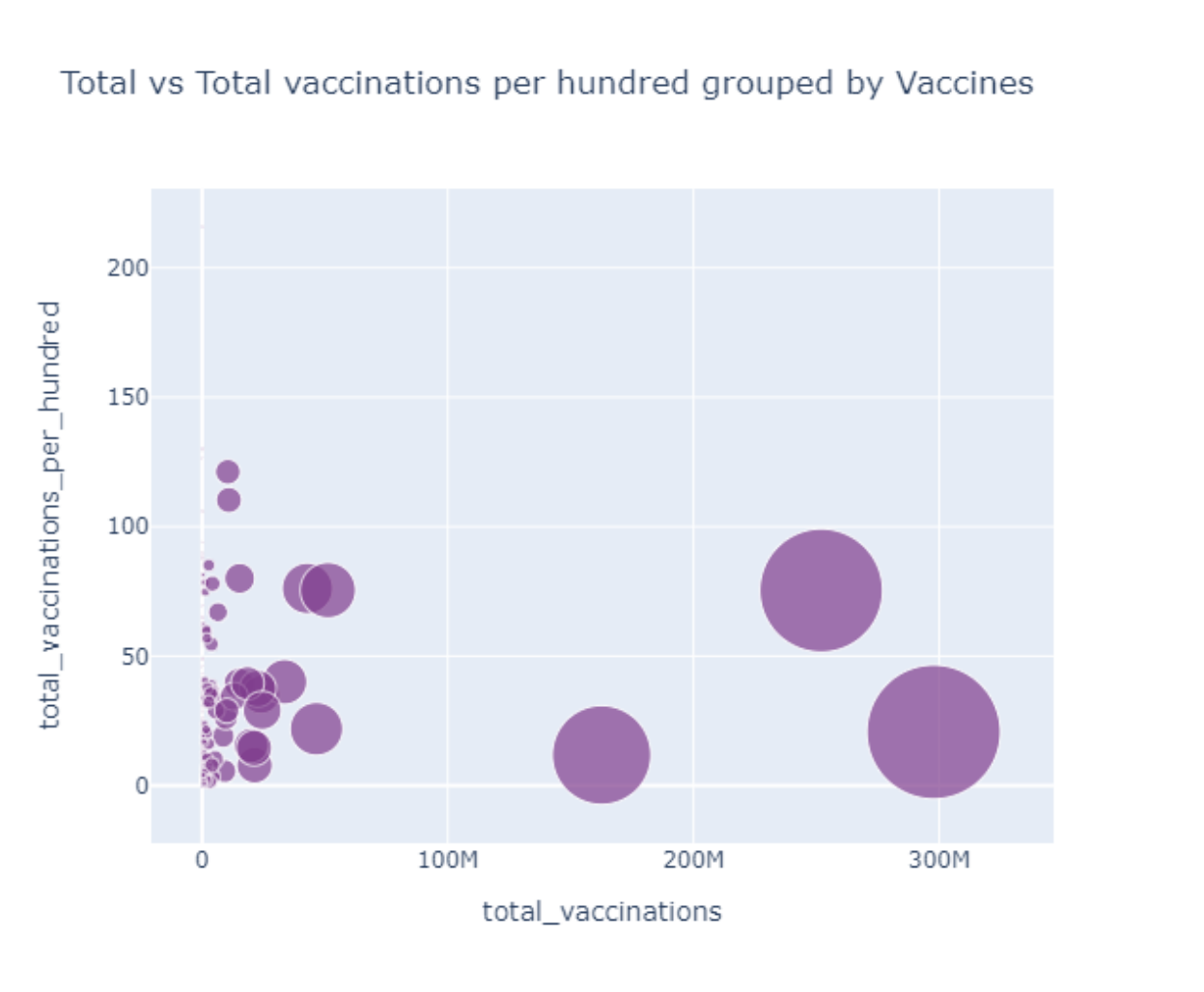
```
fig = go.Choropleth(locations = new_df["country"],locationmode = 'country names', z = new_df['daily_vaccinations'], text= new_df['country'],colorbar = dict(title= "Daily Vaccinations"))

data = [fig]

layout = go.Layout(title = 'Daily Vaccinations per Countries')

fig = dict(data = data,layout = layout)

iplot(fig)
```

Relation between Total Vaccinations and Total Vaccinations

per Hundred:

```
fig = px.scatter(new_df,x = 'total_vaccinations',y='total_vaccinations_per_hundred',
```

```
size='total_vaccinations', hover_name = 'country',size_max = 50, title="Total vs Total vaccinations per hundred grouped by Vaccines", color_discrete_sequence = px.colors.qualitative.Bold)

fig.show()
```



If you have your cursor over the scatters you will also see the country names, number of total vaccinations and number of total vaccinations per hundred. By this, we observe that:

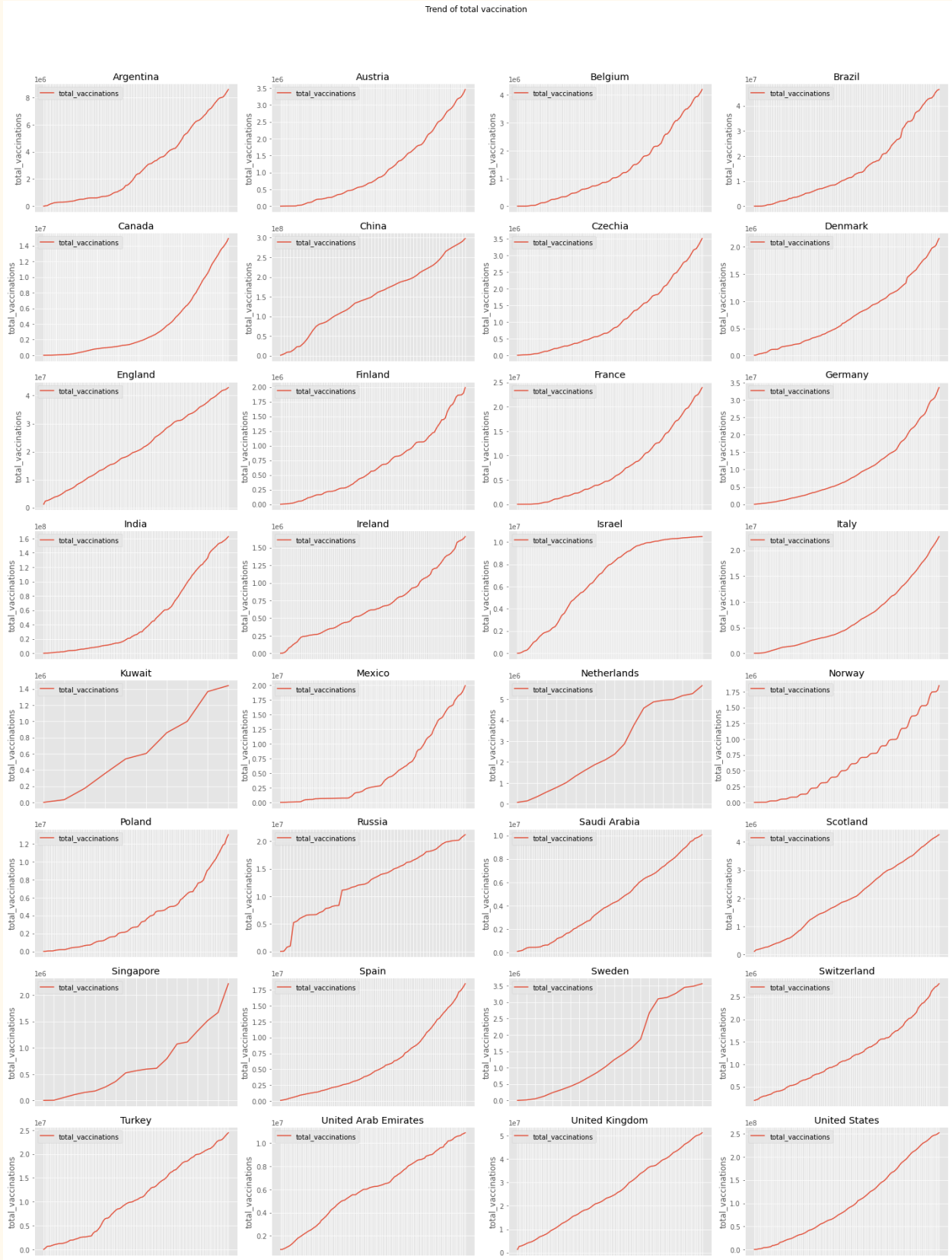
- Although USA & China produce the highest number of vaccinations to their citizens, according to their population this is not much.

What is the trend of total vaccinations according to countries?

```
def plot_trend(dataframe,feature,title,country):

plt.style.use('ggplot')


plt.figure(figsize=(20,25))

for i,country in enumerate(country):

plt.subplot(8,4,<a

onclick="parent.postMessage({'referent':'.kaggle.usercode.144406

04.62732853.plot_trend..i'}, '*')">i+1)

data = dataframe[dataframe['country'] == country]

sns.lineplot(x=data['date'] ,y=data[feature],label =

feature)

plt.xlabel('')

plt.tick_params(axis='x',which='both',top=False,bottom=False,lab elbottom=False)

plt.title(country)

plt.suptitle(title,y=1.05)

plt.tight_layout()

plt.show()
```

country = ['Argentina', 'Austria', 'Belgium', 'Brazil','Canada','China','Czechia', 'Denmark', 'England','Finland', 'France','Germany','India','Ireland', 'Israel', 'Italy', 'Kuwait','Mexico', 'Netherlands','Norway', 'Poland', 'Russia','Saudi Arabia', 'Scotland','Singapore','Spain', 'Sweden', 'Switzerland', 'Turkey','United Arab Emirates', 'United Kingdom', 'United States']

```
plot_trend(df,'total_vaccinations','Trend of total vaccination',country)
```


Trend of total vaccination
Argentina
Austria
Belgium
Brazil
Canada
China
Czechia
Denmark
England
Finland
France
Germany
India
Ireland
Israel
Italy
Kuwait
Mexico
Netherlands
Norway
Poland
Russia
Saudi Arabia
Scotland
Singapore
Spain
Sweden
Switzerland
Turkey
United Arab Emirates
United Kingdom
United States
total_vaccinations

Modeling the effects of covid 19 vaccines:

The SAIVR model

One of the first attempts to mathematically describe the spread of an infectious disease is due to Kermack and McKendrick [1]. In 1927 they introduced the so-called Susceptible-Infectious-Removed (SIR) model. The SIR model describes the dynamics of a (fixed) population of N.

Individuals are split into three compartments:

1. S(t) is the Susceptible compartment that counts the number of individuals susceptible but still not infected by the disease;
2. I(t) is the Infectious compartment that counts the number of infectious individuals;
3. R(t) is the Removed compartment. It represents the number of those who can no longer be infected either because they recovered and gained long-term immunity or because they passed away.

The model involves two positive parameters, β and γ which govern the flow from one compartment to the other:

- β is the transmission rate or effective contact rate of the disease: an infected individual comes into contact with β other individuals per unit time (the fraction that are susceptible to contracting the disease is S/N);
- γ is the removal rate. γ−1 is the mean number of days who is infected spends in the Infectious compartment.

The SIR model obeys the following system of ordinary differential equations (ODE):

$$\frac{dI}{dt} = \beta I \frac{S}{N} - \gamma I \tag{1a}$$

$$\frac{dS}{dt} = -\beta I \frac{S}{N} \tag{1b}$$

$$\frac{dR}{dt} = \gamma I \tag{1c}$$

Since they often avoid contact tracing due to the absence of symptoms, they can spread the disease while remaining undetected. Furthermore, in December 2020 a global vaccination campaign was started. Vaccinating is a safe way to transfer people from the Susceptible to the Removed compartment bypassing the Infectious one thus reducing the likelihood of an outbreak.

The SAIVR model ODEs read:

$$\frac{dI}{dt} = \beta_1 I\frac{S}{N} + \alpha_2 A\frac{S}{N} + \zeta I\frac{V}{N} - \gamma I, \tag{2a}$$

$$\frac{dA}{dt} = \alpha_1 A\frac{S}{N} + \beta_2 I\frac{S}{N} + \eta A\frac{V}{N} - \gamma A, \tag{2b}$$

$$\frac{dS}{dt} = -\beta I\frac{S}{N} - \alpha A\frac{S}{N} - \delta\frac{S}{N} + (1-\lambda)\,\epsilon V, \tag{2c}$$

$$\frac{dV}{dt} = \delta\frac{S}{N} - \eta A\frac{V}{N} - \zeta I\frac{V}{N} - \epsilon V, \tag{2d}$$

$$\frac{dR}{dt} = \gamma I + \gamma A + \lambda\epsilon V. \tag{2e}$$

The compartment inter-dependencies and flow are presented in Fig. 1 .

The parameters of the SAIVR model are the following:

-β1 describes the rate at which individuals are exposed to symptomatic infection. An infected symptomatic individual comes into contact and infects β1 susceptible individuals per unit time;

-α1 is the asymptomatic infection rate. An infected asymptomatic individual comes into contact with α1 susceptible individuals per unit time;

-β2 describes the rate at which susceptible individuals become asymptomatic infected after entering in contact with a symptomatic individual;

-$\alpha 2$ describes the rate at which who's susceptible becomes symptomatic after entering in contact with an asymptomatic individual;

-γ retains the same meaning as in the SIR model, representing the mean removal rate. $\gamma{-}1$ is the mean amount of time individuals spend either in the Infectious or Asymptomatic compartments;

-ζ is the rate at which a vaccinated (but still not immune) individual enters in contact with a symptomatic infectious;

-η describes the transmission rate at which who's asymptomatic comes into contact and infects vaccinated (but still not immune) individuals;

-δ is the first shot vaccination rate;

-λ is the vaccine efficacy;

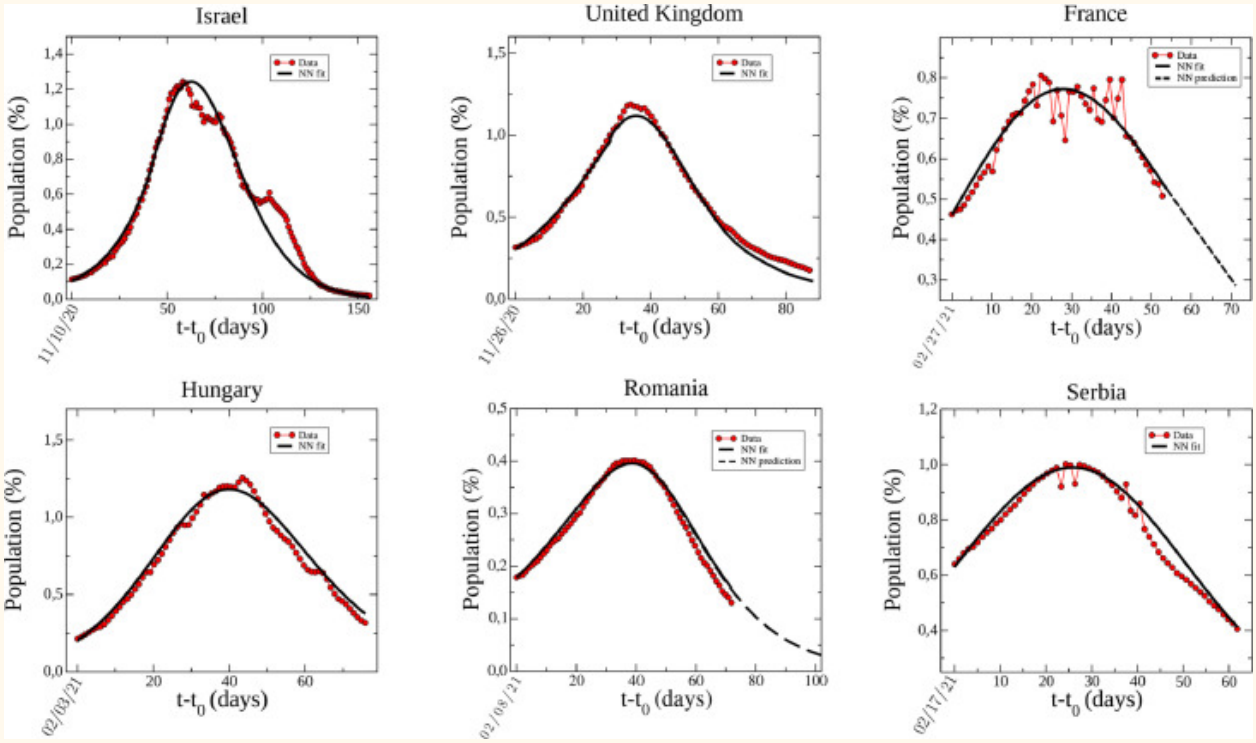
-$\epsilon{-}1$ is the mean amount of time an individual spends in the Vaccinated compartment before reaching immunity and moving to the Removed compartment.

Fitting a dataset

where IData(t) is the infectious population of a given country/state and $\tilde{I}(t)$ is its NN fit.

The machine learning approach presented in this work provides numerical solutions to a nonlinear system of ODEs without statistical error (no data is used in the first part of the process).

The supervised part is only learning what are the best parameters/conditions of the SAIVR model that fits given data, so the statistical error of the noisy data does not affect the final outcome of the process.



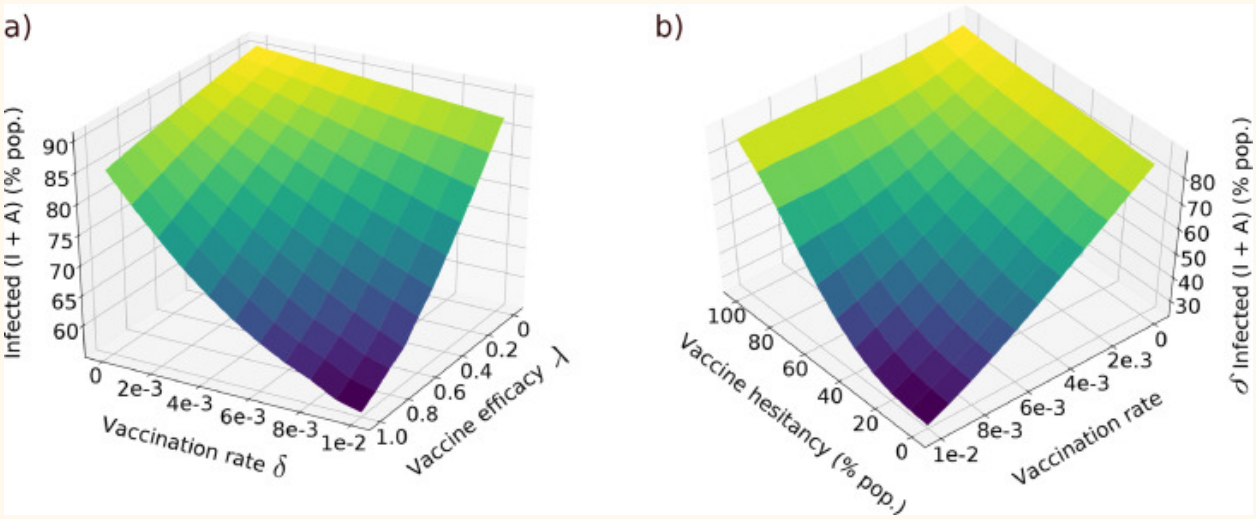
Vaccination efficacy and hesitancy

Fig. 5 presents the total infected (I+A) population under increasing values of vaccination onset times (T0), vaccination daily rates (δ), vaccine efficacy (λ), and vaccine hesitancy/denial population percentage.

In the top panel, the total infected population is shown as a function of the vaccination rate δ and vaccine efficacy λ.

As can be seen, even vaccines with a relatively low efficacy can rapidly reduce the infected population. we show how the number of those infected evolves as a function of δ and the percentage of the population that avoids getting vaccinated.

These findings suggest that vaccine hesitancy, which accounts for a significant proportion of the population might seriously threaten the reach of herd immunity, especially if the situation is worsened by the appearance of more infectious COVID-19 strains.



Key Insights to Optimize Vaccine Deployment Strategies

Priority Groups: Begin with clear prioritization based on risk factors.

Supply Chain Optimization: Ensure a well-managed supply chain to prevent shortages and wastage.

Diverse Distribution: Use various distribution channels, adapting to geographic and demographic needs.

Effective Communication: Communicate transparently to address vaccine hesitancy and concerns.

Healthcare Readiness: Strengthen healthcare facilities for safe and efficient vaccine administration.

Real-Time Monitoring: Continuously track progress and adjust strategies accordingly.

International Collaboration: Collaborate with other nations and organizations for a coordinated effort.

Adaptability: Be flexible in response to evolving circumstances, including new variants. Private

Sector Engagement: Partner with the private sector to leverage resources and expertise.

Community Involvement: Engage communities to ensure equitable access and participation.

CONCLUSION

Hence our project involves data collection, data preprocessing, exploratory data analysis, statistical analysis, and visualization of in-depth analysis of Covid-19 vaccine data, focusing on the vaccine efficacy, distribution and adverse effect.

We used these results to shed light on the impact of the vaccination campaign on the future of the pandemic. We pointed out how vaccine hesitancy is one of the most important hurdles of the campaign and further efforts should be done to support people and give them correct information about vaccines. Because of this, vaccinating the critical number of people that have to be immune in order to prevent future outbreaks (i.e. herd immunity), is likely to be out of reach.