

COVID-19 VACCINES ANALYSIS

Phase 5: Project Documentation & Submission

In this part you will document your project and prepare it for submission.

Problem Definition:

The problem is to conduct an in-depth analysis of Covid-19 vaccine data, focusing on vaccine efficacy, distribution, and adverse effects. The goal is to provide insights that aid policymakers and health organizations in optimizing vaccine deployment strategies. This project involves data collection, data preprocessing, exploratory data analysis, statistical analysis, and visualization.

Design Thinking:

Data Collection: Collect Covid-19 vaccine data from reputable sources like health organizations, government databases, and research publications.

Data Preprocessing: Clean and preprocess the data, handle missing values, and convert categorical features into numerical representations.

Exploratory Data Analysis: Explore the data to understand its characteristics, identify trends, and outliers.

Statistical Analysis: Perform statistical tests to analyze vaccine efficacy, adverse effects, and distribution across different populations.

Visualization: Create visualizations (e.g., bar plots, line charts, heatmaps) to present key findings and insights.

Data collection

The first step is to collect data that is relevant to the product demand prediction task. Once the data is collected, it needs to be cleaned and prepared for modeling.

The given dataset:

<https://www.kaggle.com/datasets/gpreda/covid-world-vaccination-progress>

Data Preprocessing

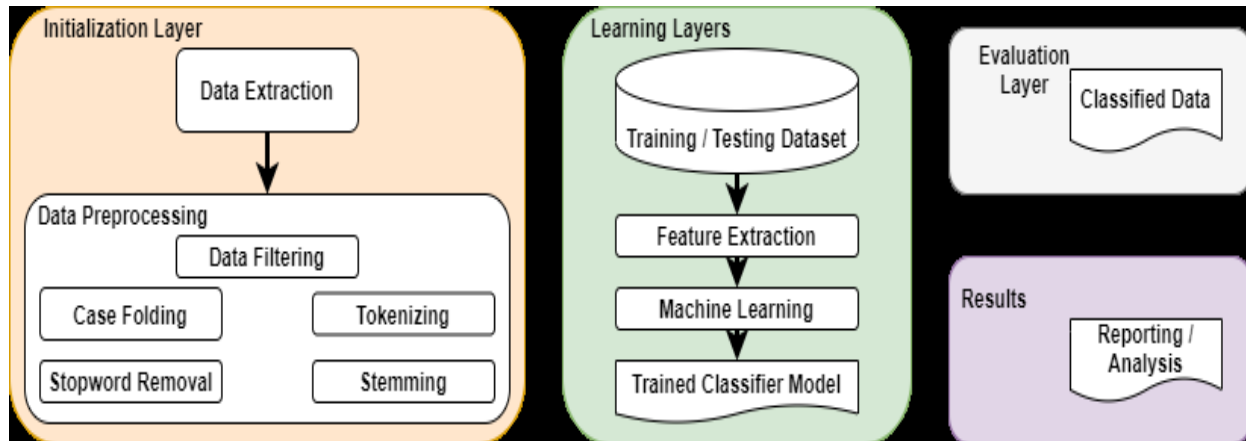
Monitoring the health situation, trends, progress and performance of health systems requires data from multiple sources on a wide variety of health topics. A core component of WHO's support to Member States is to strengthen their capacity to collect, compile, manage, analyze and use health data mainly derived from population-based sources (household surveys, civil registration systems of vital events) and institution-based sources (administrative and operational activities of institutions, such as health facilities).

Exploratory Data Analysis

Covid Explorer model's aim is to provide information on Covid 19 like how you can prevent the Corona Virus, symptoms of Corona Virus or how to book slots for the Corona Virus. You can add the current data related to Covid 19 like a graph indicated the increase or decrease of death rate from this virus, a report is given to show the death rate, the number of vaccines doses given on a particular day, state wise cases are also shown through Application Programming Interface (API) in Covid Explorer model. The model is analyzing and tracking Corona Virus. As per the data analysis this pandemic creates mental health issues but if a model gave up-to date data of the current scenario then stress can be overcome and society can fight against this pandemic. <https://covid19.who.int/> for Data analysis.

Framework

This research presents a framework for sentiment analysis of COVID-19 vaccines. We have used python as a programming language and several libraries for text mining that will be explained.



Statistical Analysis

The available data is limited and is affected by fluctuations i.e. highly variable cases were reported day by day. As a result, Cumulative data is used to predict the number of cases in Pakistan. The cumulative number of COVID-19 confirmed cases, deaths and recoveries are expected to show exponential growth over time. Therefore, we used the simple time series methods of Auto-Regressive Integrated Moving Average (ARIMA) Model to forecast the number of cases, deaths and recoveries for upcoming month. The ARIMA model has higher fitting and forecasting accuracy than exponential smoothing. It captures both the seasonal and non-seasonal forecasting trends. Due to the limited available data, we simply focus on non-seasonal models to describes the pattern (growth) over time. Hence, we assumed that the pattern of current cases will continue in the near future (at least a month). We believe that the ARIMA model, which is the combination of Autoregressive (AR) and Moving Average (MA) fits well to the nature of the available data and provide good forecasting for the short time series data.

Machine learning

Machine learning (ML) is a popular use of artificial intelligence since it automates the system and allows it to learn and improve from diverse experiences without being programmed. Computer programs can teach how to learn by giving them access to data and allowing them to utilize it for learning in ML. The learning process in ML begins with seeing the data through examples or instructions that humans offer; these observations enable ML to look for patterns in order to make the best predictions. Five different ML models were used to train the classifier and evaluate classification performance using the test dataset. These are discussed below.

Machine learning Techniques

- ☐ Random Forest
- ☐ Naive Bayes
- ☐ Decision Tree
- ☐ Logistic Regressions
- ☐ Support Vector Machine

Random Forest

The RF model is an ensemble model that generates high-precision predictions by combining the results obtained from several sub-trees. The supervised ML method known as RF may be used for both classification and regression analysis.

An RF can be represented as:

$$RF = mode\{tR_1, tR_2, tR_3, \dots, tR_n\}$$

$$RF = mode\{\sum_{i=1}^n tR_i\}$$

where $tR_1, tR_2, tR_3, \dots, tR_n$ represent the Decision Trees in RF and n denotes the number of trees.

Naive Bayes

The Bayes Theorem's premise of class conditional independence is used in the NB classification technique. This indicates that the existence of one characteristic in the likelihood of a certain event has no bearing on the presence of another, and each predictor has an equal impact on the outcome. Multinomial NB, Bernoulli NB, and Gaussian NB are the three kinds of NB classifiers. Text categorization, spam detection, and recommendation systems are all applications of this technology.

An NB can be represented as:

$$P(A|B) = \frac{P(B|A) P(A)}{P(B)}$$

Decision Tree

DTs are a technique for non-parametric supervised learning that may be used for classification and regression. DT is a model for ML that may be used for the problem-solving process of regression as well as classification. The purpose of this project is to build a model that can accurately forecast the value of a target variable by gleaning fundamental decision rules from the features of the data. A DT with

multiple branches of varying sizes is used in conjunction with partitioning the dataset into an incremental method of construction.

Logistic Regression

Logistic Regression is a statistical approach to data analysis in which one or more variables are utilized to determine the outcome. When the target variable is categorical, the optimum learning model to utilize is LR, which is the regression model that was used to estimate the likelihood of class members. Linear Regression uses a logistic function to estimate probabilities for the association between the categorical dependent variable and one or more independent variables.

Support Vector Machine

A support vector machine (SVM), which was created by Vladimir Vapnik, is a supervised learning model that can be used to both classify and regress data . On the other hand, the most popular use for it is in the realm of classification problems; in this context, it is used to generate a hyperplane on which the distance between two classes of data points is maximized.

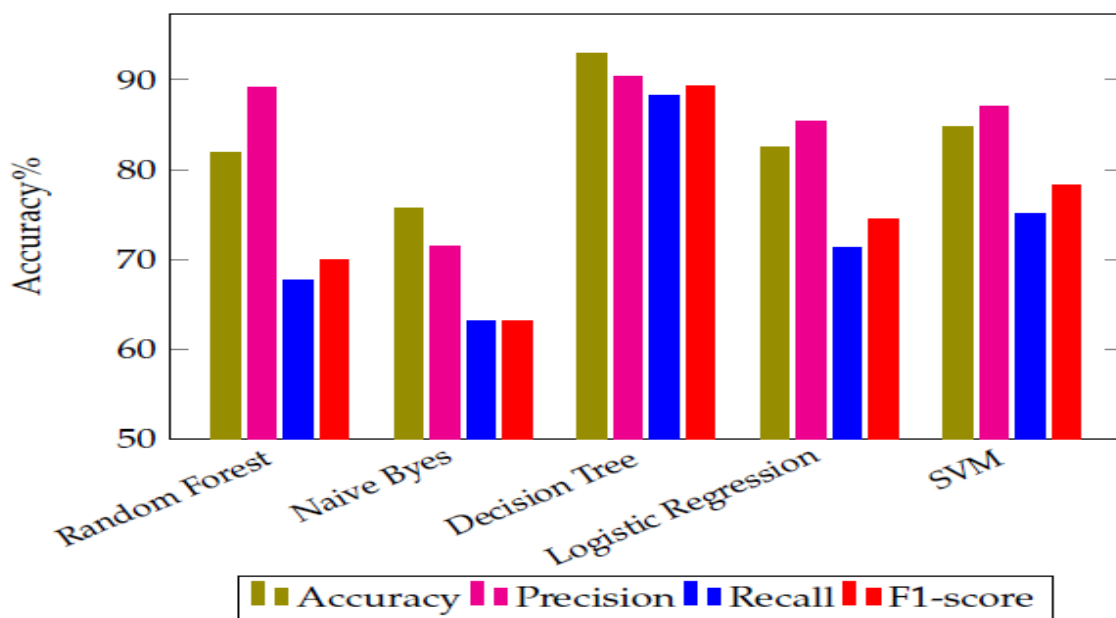
Types of SVM

- > Linear SVM

- > Nonlinear SVM

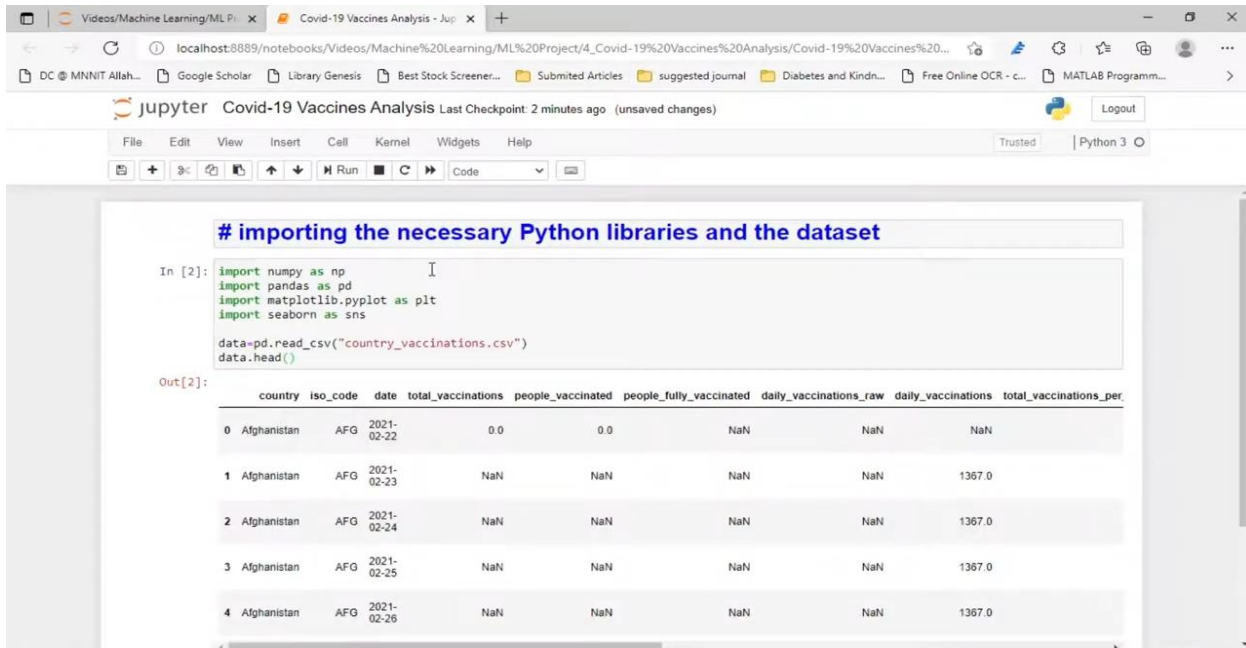
Machine learning Performance on COVID-19 vaccine analysis

Classifier Name	Accuracy%	Precision%	Recall%	F1-Score%
Random Forest	81.94	89.18	67.76	69.9
Naive Bayes	75.67	71.55	63.19	63.2
Decision Tree	93.0	90.43	88.27	89.24
Logistic Regression	82.5	85.35	71.36	74.47
SVM	84.78	87.0	75.05	78.31



Visualization

Importing the necessary Python libraries and the dataset



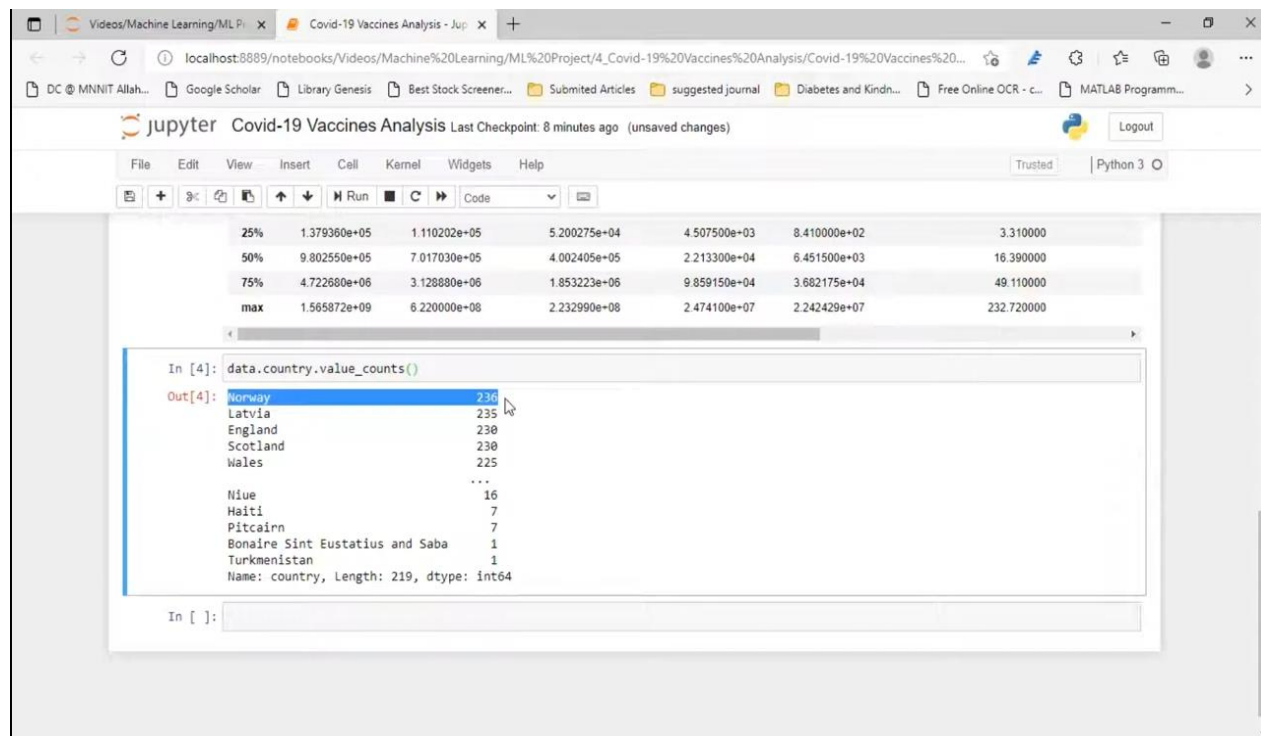
The screenshot shows a Jupyter Notebook titled "Covid-19 Vaccines Analysis". The code cell "In [2]" imports the following libraries: `numpy as np`, `pandas as pd`, `matplotlib.pyplot as plt`, and `seaborn as sns`. It then reads a CSV file named "country_vaccinations.csv" and displays the first five rows of the dataset using `data.head()`.

```
In [2]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

data=pd.read_csv("country_vaccinations.csv")
data.head()
```

The output "Out[2]" shows a DataFrame with the following columns: `country`, `iso_code`, `date`, `total_vaccinations`, `people_vaccinated`, `people_fully_vaccinated`, `daily_vaccinations_raw`, `daily_vaccinations`, and `total_vaccinations_per`.

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per
0	Afghanistan	AFG	2021-02-22	0.0	0.0	NaN	NaN	NaN	
1	Afghanistan	AFG	2021-02-23	NaN	NaN	NaN	NaN	1367.0	
2	Afghanistan	AFG	2021-02-24	NaN	NaN	NaN	NaN	1367.0	
3	Afghanistan	AFG	2021-02-25	NaN	NaN	NaN	NaN	1367.0	
4	Afghanistan	AFG	2021-02-26	NaN	NaN	NaN	NaN	1367.0	



The screenshot shows the same Jupyter Notebook. The code cell "In [4]" executes `data.country.value_counts()`. The output "Out[4]" displays a list of countries and their corresponding counts, sorted in descending order. The top five countries are Norway (236), Latvia (235), England (230), Scotland (230), and Wales (225).

```
In [4]: data.country.value_counts()
```

Out[4]:

Norway	236
Latvia	235
England	230
Scotland	230
Wales	225
...	
Niue	16
Haiti	7
Pitcairn	7
Bonaire Sint Eustatius and Saba	1
Turkmenistan	1

Name: country, Length: 219, dtype: int64

Describe the Data

Home Covid 19 vaccines analysis +

localhost:8888/notebooks/Documents%2FCovid%2019%20vaccines%20analysis%2FCovid%2019%20vaccines%20analysis.ipynb#Describe-the-Data

Jupyter Covid 19 vaccines analysis Last Checkpoint: 14 minutes ago

File Edit View Run Kernel Settings Help Trusted

JupyterLab Python 3 (ipykernel)

Describe the Data

```
[6]: data.describe()
```

	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per_hundred	people_vaccinated_per_hu
count	4.360700e+04	4.129400e+04	3.880200e+04	3.536200e+04	8.621300e+04	43607.000000	41294.0
mean	4.592964e+07	1.770508e+07	1.413830e+07	2.705996e+05	1.313055e+05	80.188543	40.9
std	2.246004e+08	7.078731e+07	5.713920e+07	1.212427e+06	7.682388e+05	67.913577	29.2
min	0.000000e+00	0.000000e+00	1.000000e+00	0.000000e+00	0.000000e+00	0.000000	0.0
25%	5.264100e+05	3.494642e+05	2.439622e+05	4.668000e+03	9.000000e+02	16.050000	11.3
50%	3.590096e+06	2.187310e+06	1.722140e+06	2.530900e+04	7.343000e+03	67.520000	41.4
75%	1.701230e+07	9.152520e+06	7.559870e+06	1.234925e+05	4.409800e+04	132.735000	67.9
max	3.263129e+09	1.275541e+09	1.240777e+09	2.474100e+07	2.242429e+07	345.370000	124.7

[]:

Home Covid 19 vaccines analysis +

localhost:8888/notebooks/Documents%2FCovid%2019%20vaccines%20analysis%2FCovid%2019%20vaccines%20analysis.ipynb#Describe-the-Data

Jupyter Covid 19 vaccines analysis Last Checkpoint: 14 minutes ago

File Edit View Run Kernel Settings Help Trusted

JupyterLab Python 3 (ipykernel)

Describe the Data

```
[6]: data.describe()
```

	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per_hundred	people_vaccinated_per_hundred	people_fully_vaccinated_per_hundred	daily_vaccinations_per_million
	3.536200e+04	8.621300e+04	43607.000000	41294.000000	38802.000000	86213.000000
	2.705996e+05	1.313055e+05	80.188543	40.927317	35.523243	3257.049157
	1.212427e+06	7.682388e+05	67.913577	29.290759	28.376252	3934.312440
	0.000000e+00	0.000000e+00	0.000000	0.000000	0.000000	0.000000
	4.668000e+03	9.000000e+02	16.050000	11.370000	7.020000	636.000000
	2.530900e+04	7.343000e+03	67.520000	41.435000	31.750000	2050.000000
	1.234925e+05	4.409800e+04	132.735000	67.910000	62.080000	4682.000000
	2.474100e+07	2.242429e+07	345.370000	124.760000	122.370000	117497.000000

[]:

Pre-process the data

The screenshot shows a Jupyter Notebook interface with the following content:

File Edit View Insert Cell Kernel Widgets Help

localhost:8889/notebooks/Videos/Machine%20Learning/ML%20Project/4_Covid-19%20Vaccines%20Analysis/Covid-19%20Vaccines%20... Logout

Trusted Python 3

Pre process the data

In [6]: `#refined our required data
data[["vaccines", "country"]]
df.head()`

Out[6]:

	vaccines	country
0	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/B...	Afghanistan
1	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/B...	Afghanistan
2	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/B...	Afghanistan
3	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/B...	Afghanistan
4	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/B...	Afghanistan

In []: |

Prepare the Data

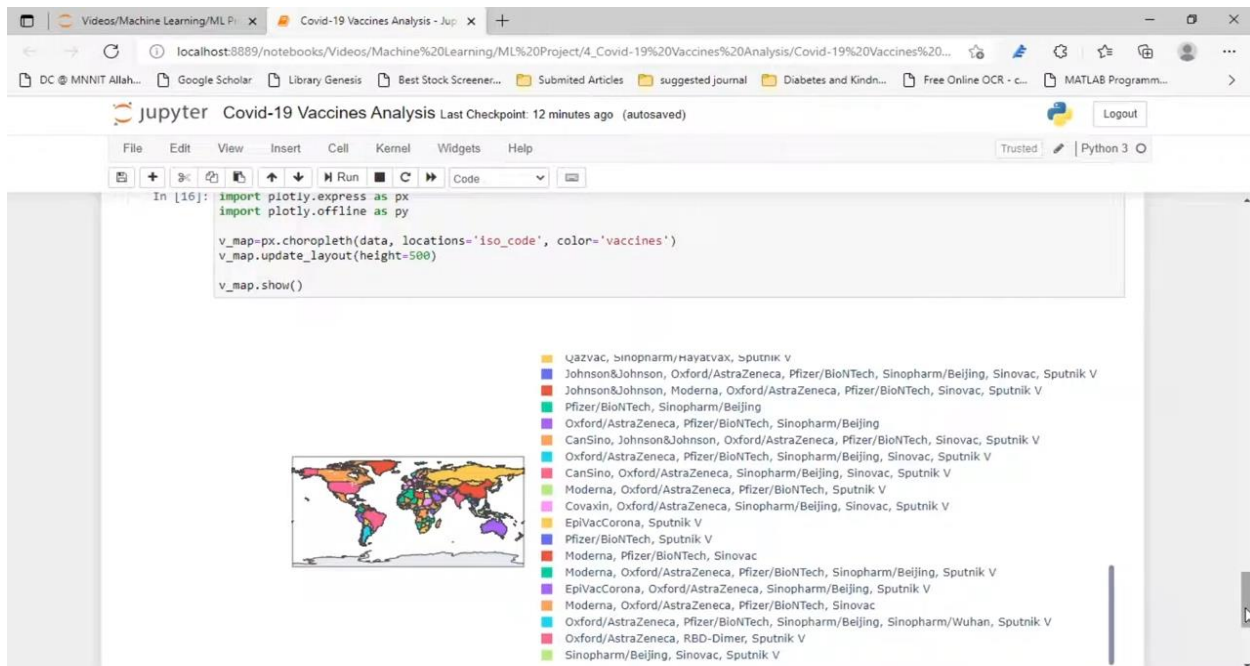
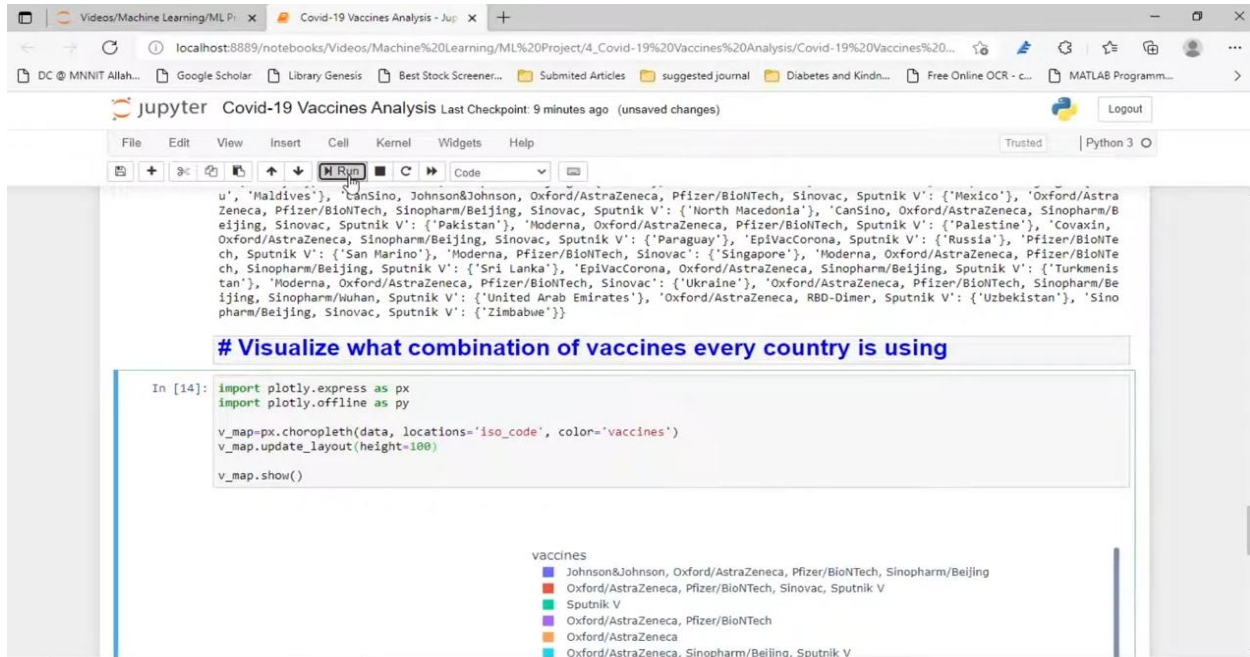
```
Video/Machine Learning/ML Pr... Covid-19 Vaccines Analysis - Jupyter | +
localhost:8889/notebooks/Video/Machine%20Learning/ML%20Project/4_Covid-19%20Vaccines%20Analysis/Covid-19%20Vaccines%20...
DC @ MNIT Allah... Google Scholar Library Genesis Best Stock Screener... Submitted Articles suggested journal Diabetes and Kind... Free Online OCR - c... MATLAB Programm...
jupyter Covid-19 Vaccines Analysis Last Checkpoint 8 minutes ago (unsaved changes) Logout
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3
In [11]: dict={}
for i in df.vaccines.unique():
    dict[i]=[df["country"][j] for j in df[df["vaccines"]==i].index]

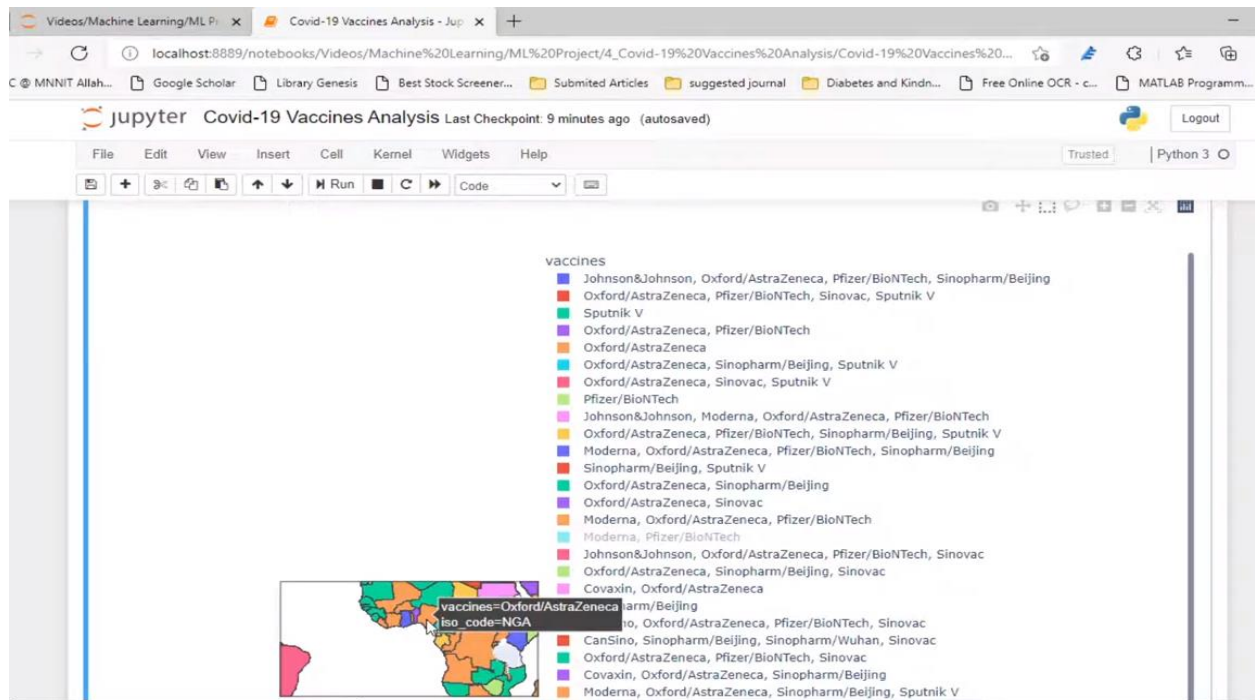
#print(dict)
vaccines={}
for key, value in dict.items():
    vaccines[key]=set(value)

print(vaccines)

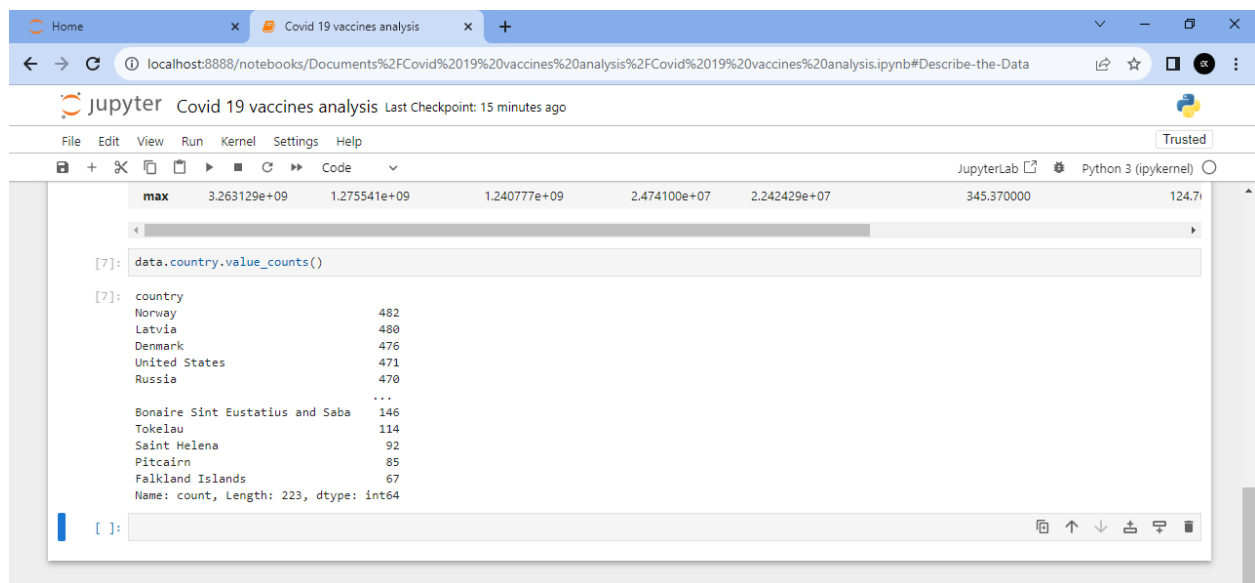
{'Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing': {'Afghanistan'}, 'Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik V': {'Bosnia and Herzegovina', 'Oman', 'Tunisia', 'Albania'}, 'Sputnik V': {'Algeria', 'Guinea'}, 'Oxford/AstraZeneca, Pfizer/BioNTech': {'Costa Rica', 'Cape Verde', 'Andorra', 'Australia', 'Cayman Islands', 'Bermuda', 'Isle of Man', 'Saudi Arabia', 'Slovenia', 'Panama'}, 'Oxford/AstraZeneca': {'Burkina Faso', 'Wallis and Futuna', 'Antigua and Barbuda', 'Bots wana', 'Uganda', 'British Virgin Islands', 'Lesotho', 'Jamaica', 'Saint Helena', 'Democratic Republic of Congo', 'Saint Vincent and the Grenadines', 'Solomon Islands', 'Fiji', 'Liberia', 'Haiti', 'Bahamas', 'Madagascar', 'South Sudan', 'Saint Kitts and Nevis', 'Ethiopia', 'Haiti', 'Angola', 'Malawi', 'Saint Lucia', 'Barbados', 'Cote d'Ivoire', 'Vanuatu', 'Sao Tome and Principe', 'Tonga', 'Yemen', 'Samoa', 'Cook Islands', 'Tuvalu', 'Georgia', 'Eswatini', 'Grenada', 'Montserrat', 'Niue', 'Suriname', 'Naur u', 'Nigeria', 'Anguilla', 'Togo', 'Pitcairn', 'Falkland Islands', 'Kosovo'}, 'Oxford/AstraZeneca, Sinopharm/Beijing, Sputnik V': {'Argentina', 'Syria'}, 'Oxford/AstraZeneca, Sinovac, Sputnik V': {'Azerbaijan', 'Armenia'}, 'Pfizer/BioNTech': {'Turks and Caicos Islands', 'New Caledonia', 'New Zealand', 'Gibraltar', 'Monaco', 'Kuwait', 'Aruba'}, 'Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech': {'Austria', 'Malta', 'Bulgaria', 'Cyprus', 'Czechia', 'Netherlands', 'Estonia', 'Iceland', 'Spain', 'Italy', 'Poland', 'Ireland', 'France', 'South Korea', 'Portugal', 'Lithuania', 'Belgium', 'Greece', 'Romania', 'Latvia'}, 'Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V': {'Mongolia', 'Lebanon', 'Bahrain', 'Iraq', 'Montenegro', 'Serbia', 'Moldova', 'Jordan', 'Bolivia'}, 'Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing': {'Vietnam', 'Bangladesh'}, 'Sinopharm/Beijing, Sputnik V': {'Venezuela', 'Kyrgyzstan', 'Belarus'}, 'Oxford/AstraZeneca, Sinopharm/Beijing': {'Mauritania', 'Cameroon', 'Namibia', 'Zambia', 'Mozambique', 'Senegal', 'Myanmar', 'Brunei', 'Trinidad and Tobago', 'Gambia'}}
```

Visualize what combination of vaccines every country is using

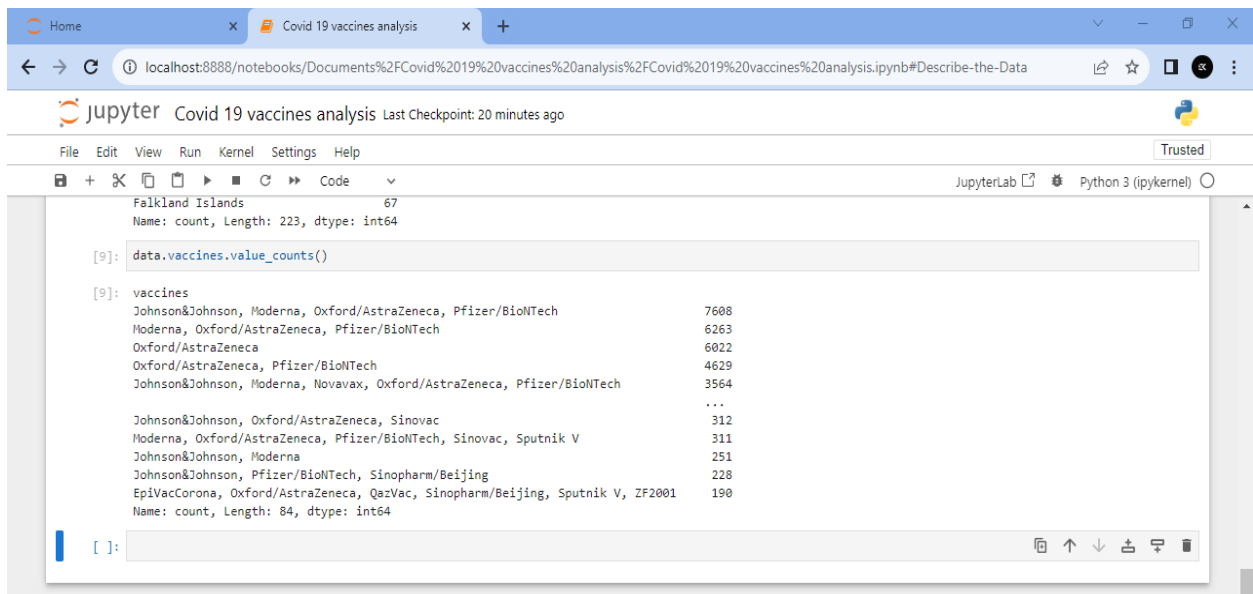




Data of country and value counts



Data of vaccines and value counts

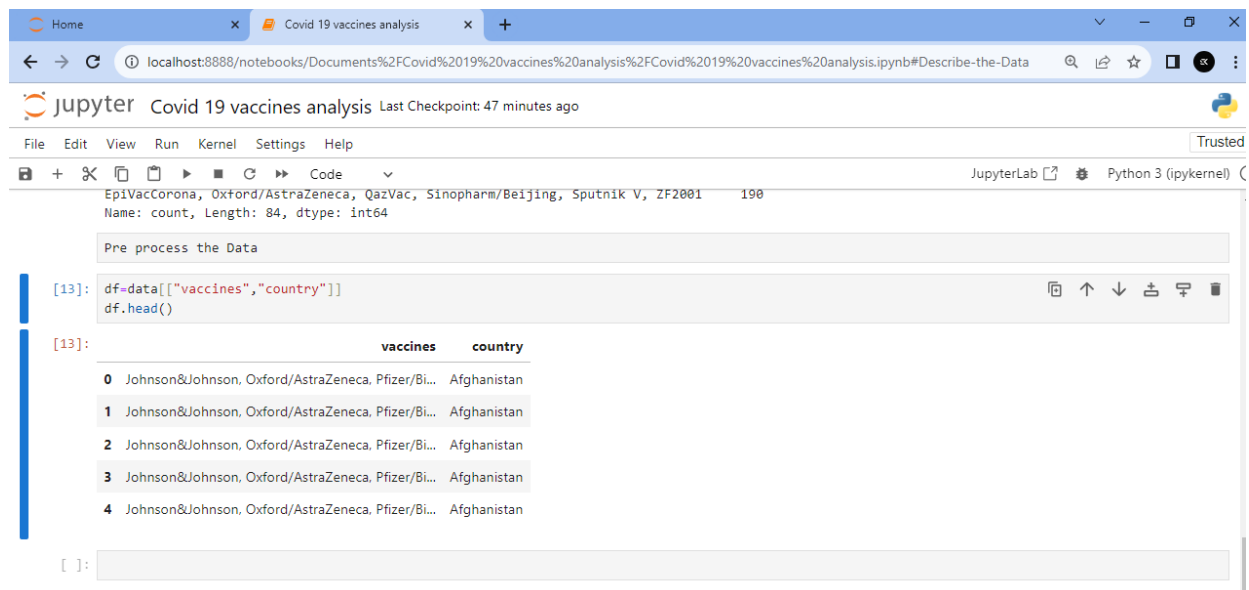


```
Falkland Islands 67
Name: count, Length: 223, dtype: int64

[9]: data.vaccines.value_counts()

[9]: vaccines
Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech    7608
Moderna, Oxford/AstraZeneca, Pfizer/BioNTech                    6263
Oxford/AstraZeneca                                                6022
Oxford/AstraZeneca, Pfizer/BioNTech                              4629
Johnson&Johnson, Moderna, Novavax, Oxford/AstraZeneca, Pfizer/BioNTech 3564
...
Johnson&Johnson, Oxford/AstraZeneca, Sinovac                    312
Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik V    311
Johnson&Johnson, Moderna                                         251
Johnson&Johnson, Pfizer/BioNTech, Sinopharm/Beijing              228
EpiVacCorona, Oxford/AstraZeneca, QazVac, Sinopharm/Beijing, Sputnik V, ZF2001 190
Name: count, Length: 84, dtype: int64
```

Data of vaccines used in country



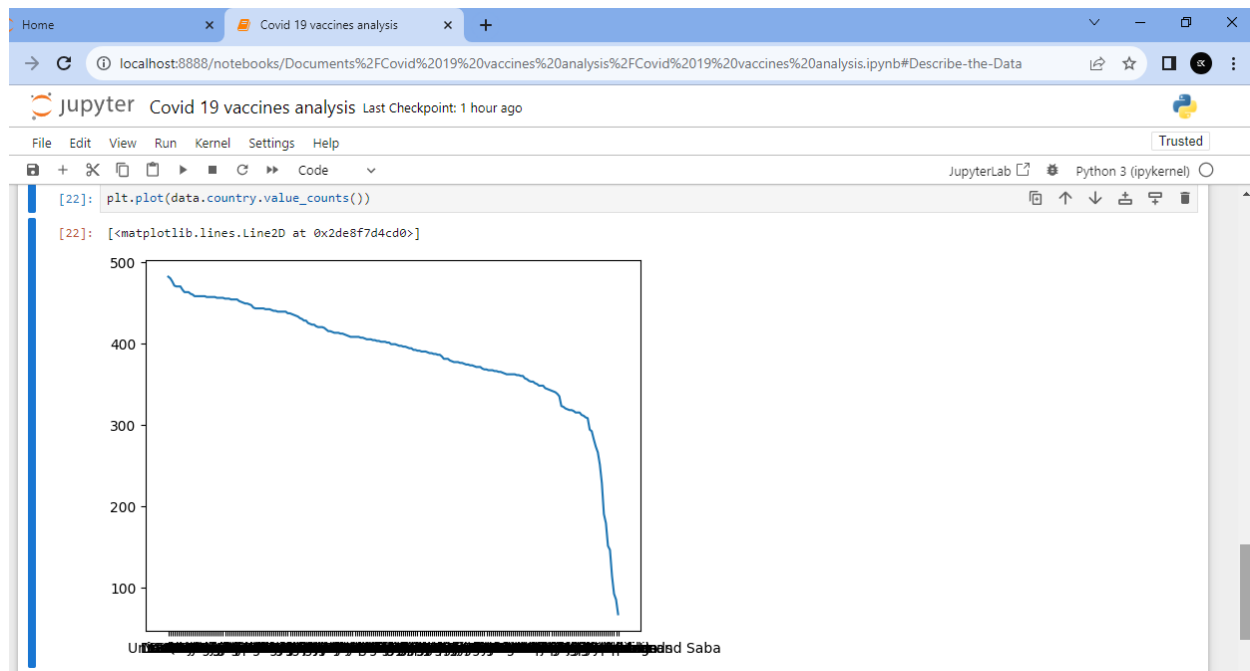
```
EpiVacCorona, Oxford/AstraZeneca, QazVac, Sinopharm/Beijing, Sputnik V, ZF2001 190
Name: count, Length: 84, dtype: int64

Pre process the Data

[13]: df=data[["vaccines", "country"]]
df.head()

[13]:
   vaccines country
0 Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...  Afghanistan
1 Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...  Afghanistan
2 Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...  Afghanistan
3 Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...  Afghanistan
4 Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...  Afghanistan
```

Line plot graph for Country and value counts



Data of country and date with daily vaccination

The image shows a JupyterLab interface with a browser window at the top displaying the URL `localhost:8888/notebooks/Documents%2FCovid%2019%20vaccines%20analysis%2FCovid%2019%20vaccines%20analysis.ipynb#Describe-the-Data`. The JupyterLab title bar indicates the notebook is named "Covid 19 vaccines analysis" and the last checkpoint was 2 hours ago. The interface includes a menu bar (File, Edit, View, Run, Kernel, Settings, Help) and a toolbar with icons for file operations and code execution. The code editor shows the following code:

```
[75]: df=data[["country","date","daily_vaccinations",]]
df.head()
```

The output of the code is a table with 5 rows and 3 columns: country, date, and daily_vaccinations. The table is titled "country" and the x-axis is labeled "country".

	country	date	daily_vaccinations
0	Afghanistan	2021-02-22	NaN
1	Afghanistan	2021-02-23	1367.0
2	Afghanistan	2021-02-24	1367.0
3	Afghanistan	2021-02-25	1367.0
4	Afghanistan	2021-02-26	1367.0

Data of Source name and source website

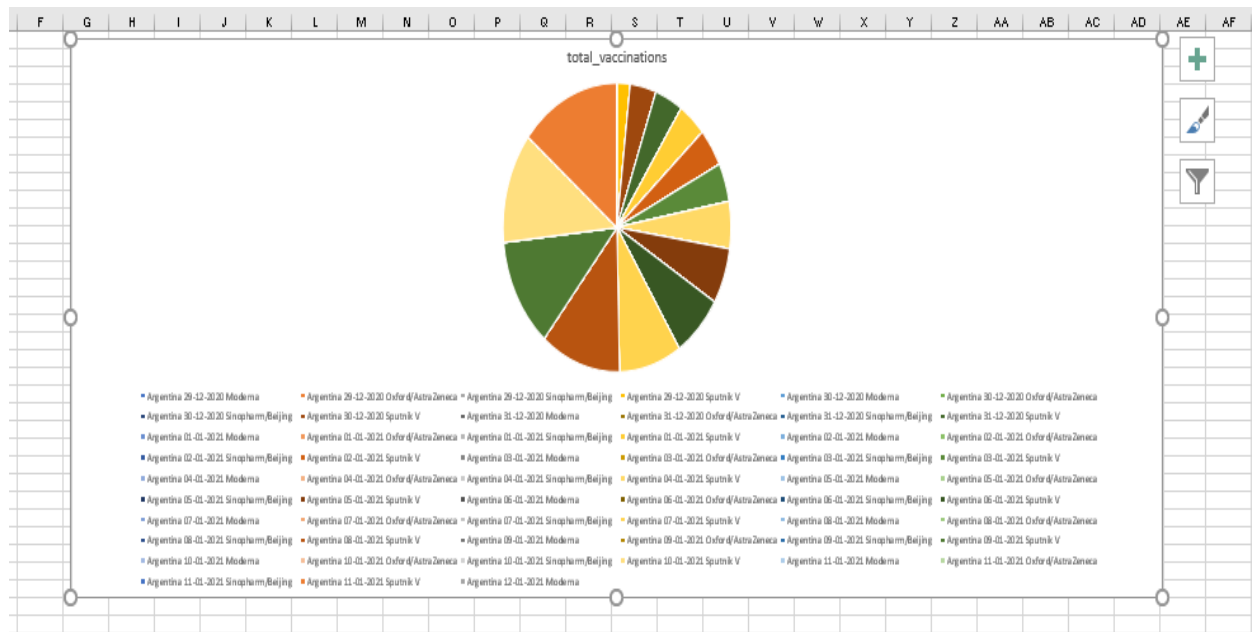
```
Home x Covid 19 vaccines analysis x +
localhost:8888/notebooks/Documents%2FCovid%2019%20vaccines%20analysis%2FCovid%2019%20vaccines%20analysis.ipynb#Describe-the-Data
Jupyter Covid 19 vaccines analysis Last Checkpoint: 1 hour ago
File Edit View Run Kernel Settings Help
+ - [ ] Code Python 3 (ipykernel)
Johnson&Johnson, Moderna 224
Johnson&Johnson, Pfizer/BioNTech, Sinopharm/Beijing 228
EpiVacCorona, Oxford/AstraZeneca, QazVac, Sinopharm/Beijing, Sputnik V, ZF2001 190
Name: count, Length: 84, dtype: int64

[67]: df[df["source_name", "source_website"]]
df.head()

[67]: source_name source_website
0 World Health Organization https://covid19.who.int/
1 World Health Organization https://covid19.who.int/
2 World Health Organization https://covid19.who.int/
3 World Health Organization https://covid19.who.int/
4 World Health Organization https://covid19.who.int/

[ ]:
```

Pie chart for statistical analysis



Results and Discussion

This section presents the accuracy results of sentiment analysis carried out using five distinct methods applied to two distinct datasets, with the second dataset being further subdivided into five distinct vaccination datasets. The accuracy, precision, recall, F1 score, and support measurement are derived from the Random Forest, Naive Bayes, Decision Tree, Logistic Regression, and Support Vector Machine (SVM).