

COVID VACCINE Analysis

PHASE -4



1: Performing exploratory data analysis

- ☒ **What Is Exploratory Data Analysis?**
- ☒ Exploratory Data Analysis is a data analytics process to understand the data in depth and learn the different data characteristics, often with visual means. This allows you to get a better feel of your data and find useful patterns in it.
- ☒ You need to know the patterns in your data and determine which variables are important and which do not play a significant role in the output. Further, some variables may have correlations with other variables. You also need to recognize errors in your data.

Importing a Dataset

The screenshot shows a Jupyter Notebook interface running in a browser window. The title bar indicates the window is titled "jupyter Untitled7" and the URL is "localhost:8888/notebooks/Untitled7.ipynb". The notebook contains the following code:

```
from matplotlib import pyplot as plt
%matplotlib inline
```

In the output pane, the command `df = pd.read_csv(r'C:\Users\KGTION\Desktop\Kiran Ratha\country_vaccinations.csv')` is run, followed by `df.head(10)`. The resulting DataFrame is displayed as a table:

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per_hundred
0	Afghanistan	AFG	2021-02-22	0.0	0.0	NaN	NaN	NaN	NaN
1	Afghanistan	AFG	2021-02-23	NaN	NaN	NaN	NaN	NaN	NaN
2	Afghanistan	AFG	2021-02-24	NaN	NaN	NaN	NaN	NaN	NaN
3	Afghanistan	AFG	2021-02-25	NaN	NaN	NaN	NaN	NaN	NaN
4	Afghanistan	AFG	2021-02-26	NaN	NaN	NaN	NaN	NaN	NaN
5	Afghanistan	AFG	2021-02-27	NaN	NaN	NaN	NaN	NaN	NaN
6	Afghanistan	AFG	2021-02-28	NaN	NaN	NaN	NaN	NaN	NaN
7	Afghanistan	AFG	2021-03-01	NaN	NaN	NaN	NaN	NaN	NaN
8	Afghanistan	AFG	2021-03-02	NaN	NaN	NaN	NaN	NaN	NaN
9	Afghanistan	AFG	2021-03-03	NaN	NaN	NaN	NaN	NaN	NaN

Use info

```
In [19]: df.info
Out[19]: <bound method DataFrame.info of
   0    Afghanistan    AFG  2021-02-22      country iso_code       date  total_vaccinations \
   0           Afghanistan    AFG  2021-02-22      AFG  2021-02-22      0.0
   1           Afghanistan    AFG  2021-02-23      NaN
   2           Afghanistan    AFG  2021-02-24      NaN
   3           Afghanistan    AFG  2021-02-25      NaN
   4           Afghanistan    AFG  2021-02-26      NaN
   ..
   86507      Zimbabwe     ZWE  2022-03-25      8691642.0
   86508      Zimbabwe     ZWE  2022-03-26      8791728.0
   86509      Zimbabwe     ZWE  2022-03-27      8845039.0
   86510      Zimbabwe     ZWE  2022-03-28      8924360.0
   86511      Zimbabwe     ZWE  2022-03-29      9039729.0

   people_vaccinated  people_fully_vaccinated  daily_vaccinations_raw \
   0                  0.0                         NaN          NaN
   1                  NaN                         NaN          NaN
   2                  NaN                         NaN          NaN
   3                  NaN                         NaN          NaN
   4                  NaN                         NaN          NaN
   ..
   86507      4814582.0            3473523.0          139213.0
   86508      4886242.0            3487982.0          180006.0
   86509      4918147.0            3493763.0          53311.0
   86510      4975433.0            3501493.0          89321.0
   86511      5053114.0            3510256.0          105369.0

   daily_vaccinations  total_vaccinations_per_hundred \
   0                      NaN                     0.00
   1                   1367.0                     NaN
   2                   1367.0                     NaN
   3                   1367.0                     NaN
   4                   1367.0                     NaN
   ..
   86507      69579.0                     57.59
   86508      83429.0                     58.25
   86509      90629.0                     58.61
   86510      100000.0                    60.00
```

Use describe

```
[86512 rows x 15 columns]>

In [20]: df.describe
Out[20]: <bound method NDFrame.describe of
          country iso_code      date  total_vaccinations  ...
0  Afghanistan   AFG  2021-02-22        0.0
1  Afghanistan   AFG  2021-02-23       NaN
2  Afghanistan   AFG  2021-02-24       NaN
3  Afghanistan   AFG  2021-02-25       NaN
4  Afghanistan   AFG  2021-02-26       NaN
...
86507  Zimbabwe   ZWE  2022-03-25  8691642.0
86508  Zimbabwe   ZWE  2022-03-26  8791728.0
86509  Zimbabwe   ZWE  2022-03-27  8845039.0
86510  Zimbabwe   ZWE  2022-03-28  8934360.0
86511  Zimbabwe   ZWE  2022-03-29  9039729.0

    people_vaccinated  people_fully_vaccinated  daily_vaccinations_raw  ...
0                  0.0                      NaN                    139213.0
1                  NaN                      NaN                     NaN
2                  NaN                      NaN                     NaN
3                  NaN                      NaN                     NaN
4                  NaN                      NaN                     ...
...
86507  4814582.0            3473523.0           100086.0
86508  4886242.0            3487962.0           100086.0
86509  4918147.0            3499763.0           53311.0
86510  4975433.0            3581493.0           89321.0
86511  5053114.0            3510256.0           105369.0

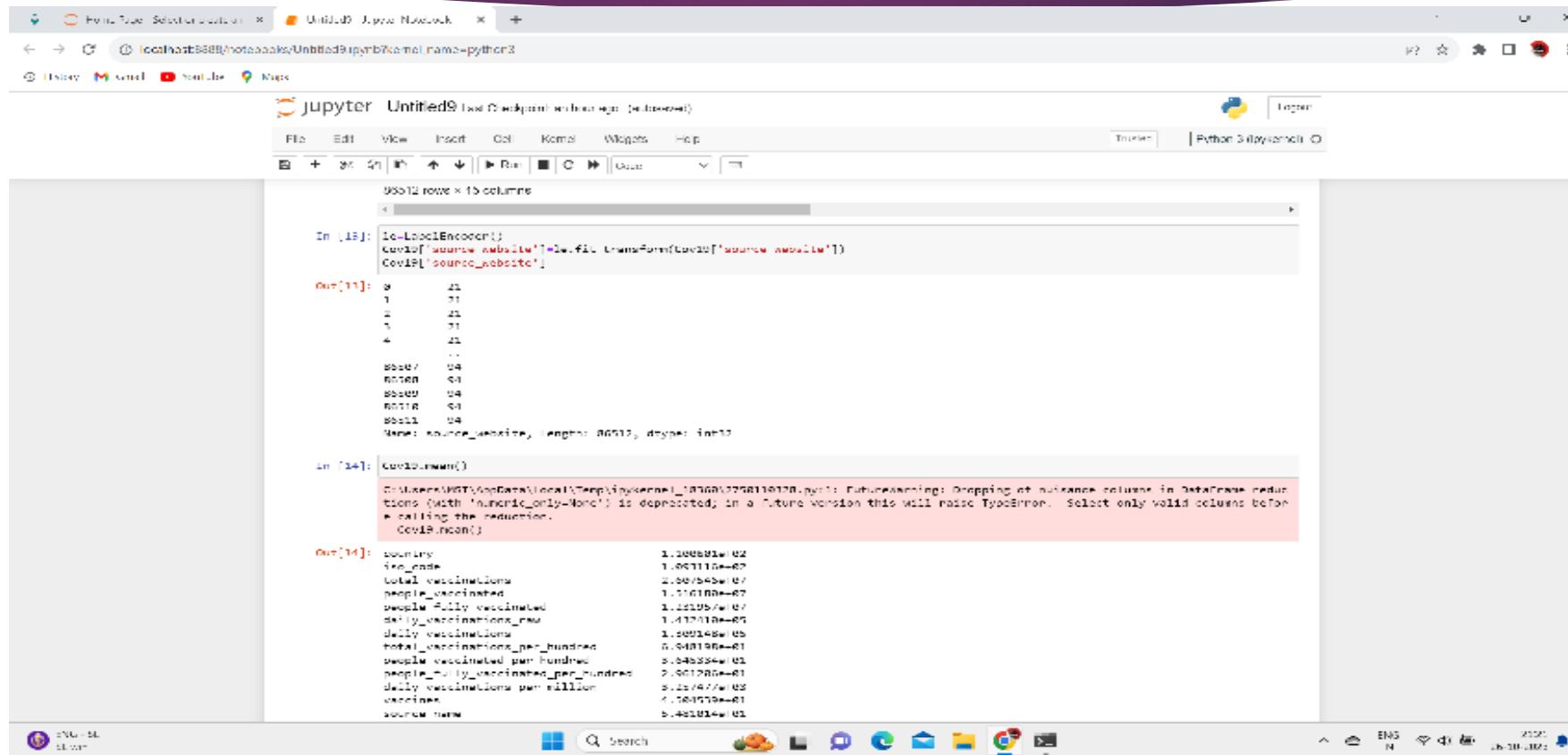
    daily_vaccinations  total_vaccinations_per_hundred  ...
0                   NaN                      0.00
1                 1367.0                      NaN
2                 1367.0                      NaN
3                 1367.0                      NaN
4                 1367.0                      NaN
...
86507                ...                      ...
86508                ...                      ...
86509                ...                      ...
86510                ...                      ...
86511                ...                      ...

25°C Partly cloudy
```

2: Statistical analysis

- ☒ Statistical analysis is the collection and interpretation of data in order to uncover patterns and trends. It is a component of data analytics. Statistical analysis can be used in situations like gathering research interpretations, statistical modeling or designing surveys and studies. It can also be useful for business intelligence organizations that have to work with large data volumes.
- ☒ Describe the nature of the data to be analyzed.
- ☒ Explore the relation of the data to the underlying population.
- ☒ Create a model to summarize an understanding of how the data relates to the underlying population.
- ☒ Prove (or disprove) the validity of the model.
- ☒ Employ predictive analytics to run scenarios that will help guide future actions.

- The five basic methods are mean, standard deviation, regression, hypothesis testing, and sample size determination. It is widely used by governments, businesses, banking entities, insurance companies, etc



A screenshot of a Jupyter Notebook interface running in a browser window. The window title is "Untitled9 - Jupyter Notebook". The URL in the address bar is "localhost:8888/notebooks/Untitled9.ipynb?kernel_name=python3". The browser toolbar includes Back, Forward, Stop, Refresh, Home, and other standard icons.

The notebook interface has a menu bar with File, Edit, View, Insert, Cell, Kernel, Widgets, Help, and a toolbar with various icons. A "Trusted" badge is visible next to the kernel name.

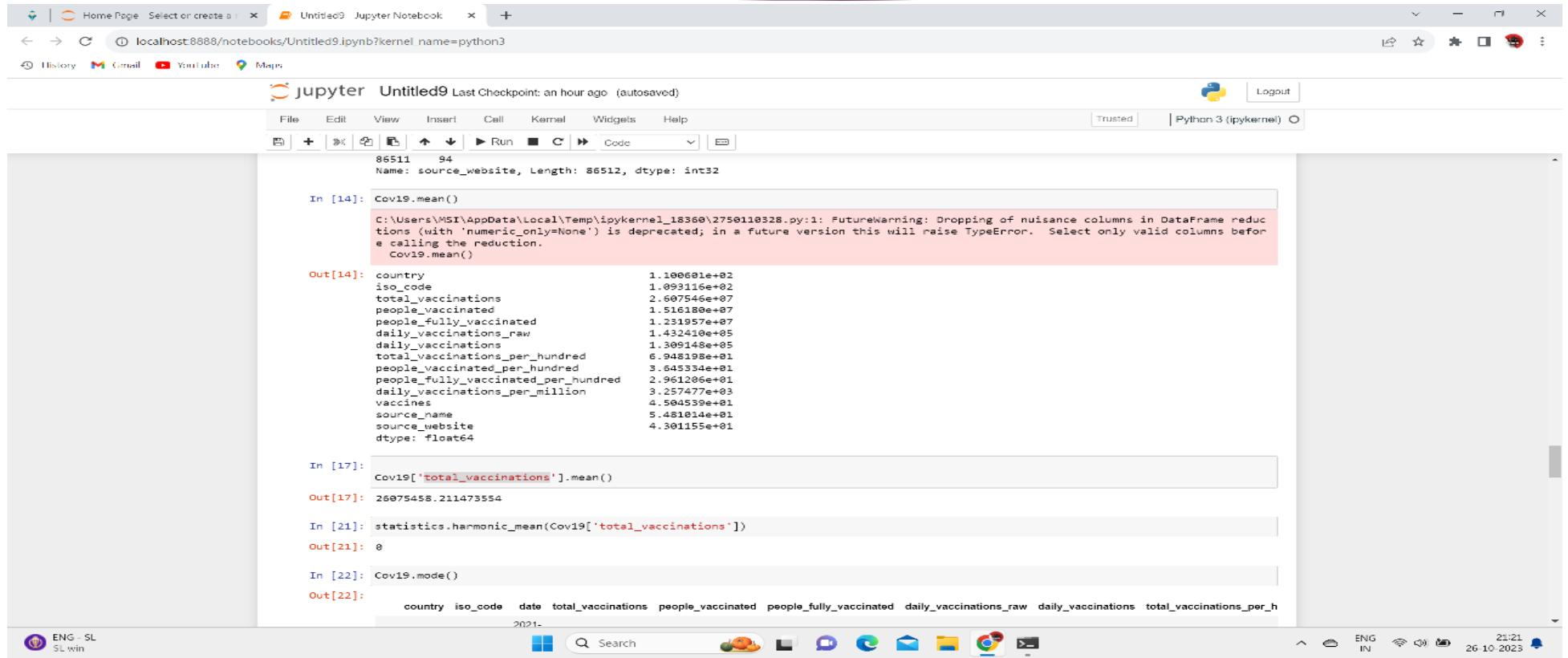
The code cell In [13] contains:In [13]: le=LabelEncoder()
Covid1['source_website']=le.fit_transform(Covid1['source_website'])
Covid1['source_website']

The output cell Out[13] shows the transformed data:Out[13]: 0 21
1 21
2 21
3 21
4 21
.. .
66512 24
Name: source_website, length: 66513, dtype: int32

The code cell In [14] contains:In [14]: Covid1.dropna()

The output cell Out[14] shows the resulting DataFrame:Out[14]: country 1.200582e-02
iso_code 1.051112e-02
total_vaccinations 2.007549e-07
people_vaccinated 1.776100e-07
people_fully_vaccinated 1.221057e-07
daily_vaccinations_raw 1.417011e-07
daily_vaccinations 1.200349e-05
total_vaccinations_per_hundred 6.940196e-01
people_vaccinated_per_hundred 5.485529e-01
people_fully_vaccinated_per_hundred 5.901786e-01
daily_vaccinations_per_million 5.157477e-03
excerpts 4.760177e-01
source_name 5.481814e-01

MEAN()



The screenshot shows a Jupyter Notebook interface running on a Windows 10 desktop. The notebook has a single open cell (In [14]) containing the command `Cov19.mean()`. The output (Out[14]) displays various statistics from the `Cov19` DataFrame, including mean values for columns like `country`, `iso_code`, `total_vaccinations`, `people_vaccinated`, etc. A warning message is visible in the output cell, indicating that dropping of nuisance columns in DataFrame reductions is deprecated. Subsequent cells show the execution of `Cov19['total_vaccinations'].mean()` (Out[17]), the harmonic mean of `total_vaccinations` (Out[21]), and the mode of the `Cov19` DataFrame (Out[22]). The status bar at the bottom indicates the system is ENG SL win, the date is 26.10.2023, and the time is 21:21.

```
In [14]: Cov19.mean()
C:\Users\MSI\AppData\Local\Temp\ipykernel_18380\2750110328.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError. Select only valid columns before calling the reduction.
Cov19.mean()

Out[14]:
country           1.106691e+02
iso_code          1.093116e+02
total_vaccinations   2.607546e+07
people_vaccinated    1.516180e+07
people_fully_vaccinated 1.231957e+07
daily_vaccinations_raw 1.432410e+05
daily_vaccinations   1.309148e+05
total_vaccinations_per_hundred 6.948198e+01
people_vaccinated_per_hundred 3.645334e+01
people_fully_vaccinated_per_hundred 2.961206e+01
daily_vaccinations_per_million 3.257477e+03
vaccines           4.584539e+01
source_name         5.481014e+01
source_website      4.301155e+01
dtype: float64

In [17]: Cov19['total_vaccinations'].mean()
Out[17]: 26075458.211473554

In [21]: statistics.harmonic_mean(Cov19['total_vaccinations'])
Out[21]: 0

In [22]: Cov19.mode()
Out[22]:
country iso_code date total_vaccinations people_vaccinated people_fully_vaccinated daily_vaccinations_raw daily_vaccinations total_vaccinations_per_h
```

MODE()

Home Page Select or create a new notebook Untitled9 Jupyter Notebook

localhost:8888/notebooks/Untitled9.ipynb?kernel_name=python3

History Gmail YouTube Maps

jupyter Untitled9 Last Checkpoint: an hour ago (autosaved)

In [22]: Cov19.mode()

Out[22]:

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per_h
0	149.0	145.0	2021-08-08	7680976.0	54146.0	460292.0	377.0	0.0	
1	Nan	Nan	2021-08-08	Nan	Nan	Nan	Nan	Nan	
2	Nan	Nan	2021-08-10	Nan	Nan	Nan	Nan	Nan	
3	Nan	Nan	2021-08-11	Nan	Nan	Nan	Nan	Nan	
4	Nan	Nan	2021-08-12	Nan	Nan	Nan	Nan	Nan	
5	Nan	Nan	2021-08-13	Nan	Nan	Nan	Nan	Nan	
6	Nan	Nan	2021-08-14	Nan	Nan	Nan	Nan	Nan	
7	Nan	Nan	2021-08-15	Nan	Nan	Nan	Nan	Nan	
8	Nan	Nan	2021-08-16	Nan	Nan	Nan	Nan	Nan	
9	Nan	Nan	2021-08-17	Nan	Nan	Nan	Nan	Nan	
10	Nan	Nan	2021-08-18	Nan	Nan	Nan	Nan	Nan	
11	Nan	Nan	2021-08-19	Nan	Nan	Nan	Nan	Nan	
12	Nan	Nan	2021-08-20	Nan	Nan	Nan	Nan	Nan	
13	Nan	Nan	2021-08-21	Nan	Nan	Nan	Nan	Nan	
14	Nan	Nan	2021-08-22	Nan	Nan	Nan	Nan	Nan	

ENG SL
SL win

Search

21:21 26.10.2023

Median , Variance

The screenshot shows a Jupyter Notebook interface running in a web browser. The title bar indicates the notebook is titled "Untitled9" and is using a Python 3 kernel. The main area displays several code cells and their corresponding outputs:

```
In [23]: statistics.median(Cov19['total_vaccinations'])
Out[23]: 1188693.5

In [24]: statistics.variance(Cov19['total_vaccinations'])
Out[24]: 2.6090190968582216e+16

In [25]: statistics.variance(Cov19['people_fully_vaccinated'])
Out[25]: 5980253699691319.8

In [26]: statistics.stdev(Cov19['total_vaccinations'])
Out[26]: 161524583.17105237

In [27]: Cov19.skew()
C:\Users\MSI\AppData\Local\Temp\ipykernel_18360\486741530.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError. Select only valid columns before calling the reduction.
cov19.skew()

Out[27]: country          0.008257
iso_code         0.024474
total_vaccinations 13.709461
people_vaccinated 11.084284
people_fully_vaccinated 11.764974
daily_vaccinations_raw 14.995365
daily_vaccinations 15.426948
total_vaccinations_per_hundred 0.768833
people_vaccinated_per_hundred 0.312882
people_fully_vaccinated_per_hundred 0.582681
daily_vaccinations_per_million 5.005845
vaccines        -0.174978
source_name      -0.702698
source_website   0.682960
dtype: float64
```

The bottom status bar shows system information including the taskbar, search bar, and system icons.

Home Page Select or create a new notebook Untitled9 Jupyter Notebook +

localhost:8888/notebooks/Untitled9.ipynb?kernel_name=python3

History Gmail YouTube Maps

jupyter Untitled9 Last Checkpoint: an hour ago (autosaved)

Logout Trusted Python 3 (ipykernel) O

In [26]: statistics.stdev(Cov19['total_vaccinations'])

Out[26]: 161524583.17105237

In [27]: Cov19.skew()

C:\Users\MSI\AppData\Local\Temp\ipykerne1_18360\486741530.py:1: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is deprecated; in a future version this will raise TypeError. Select only valid columns before calling the reduction.

Cov19.skew()

Out[27]: country 0.008257
iso_code 0.024474
total_vaccinations 13.769461
people_vaccinated 11.084284
people_fully_vaccinated 11.764974
daily_vaccinations_raw 14.995365
daily_vaccinations 15.426948
total_vaccinations_per_hundred 0.768833
people_vaccinated_per_hundred 0.312882
people_fully_vaccinated_per_hundred 0.582681
daily_vaccinations_per_million 5.065845
vaccines -0.174978
source_name -0.702698
source_website 0.682960
dtype: float64

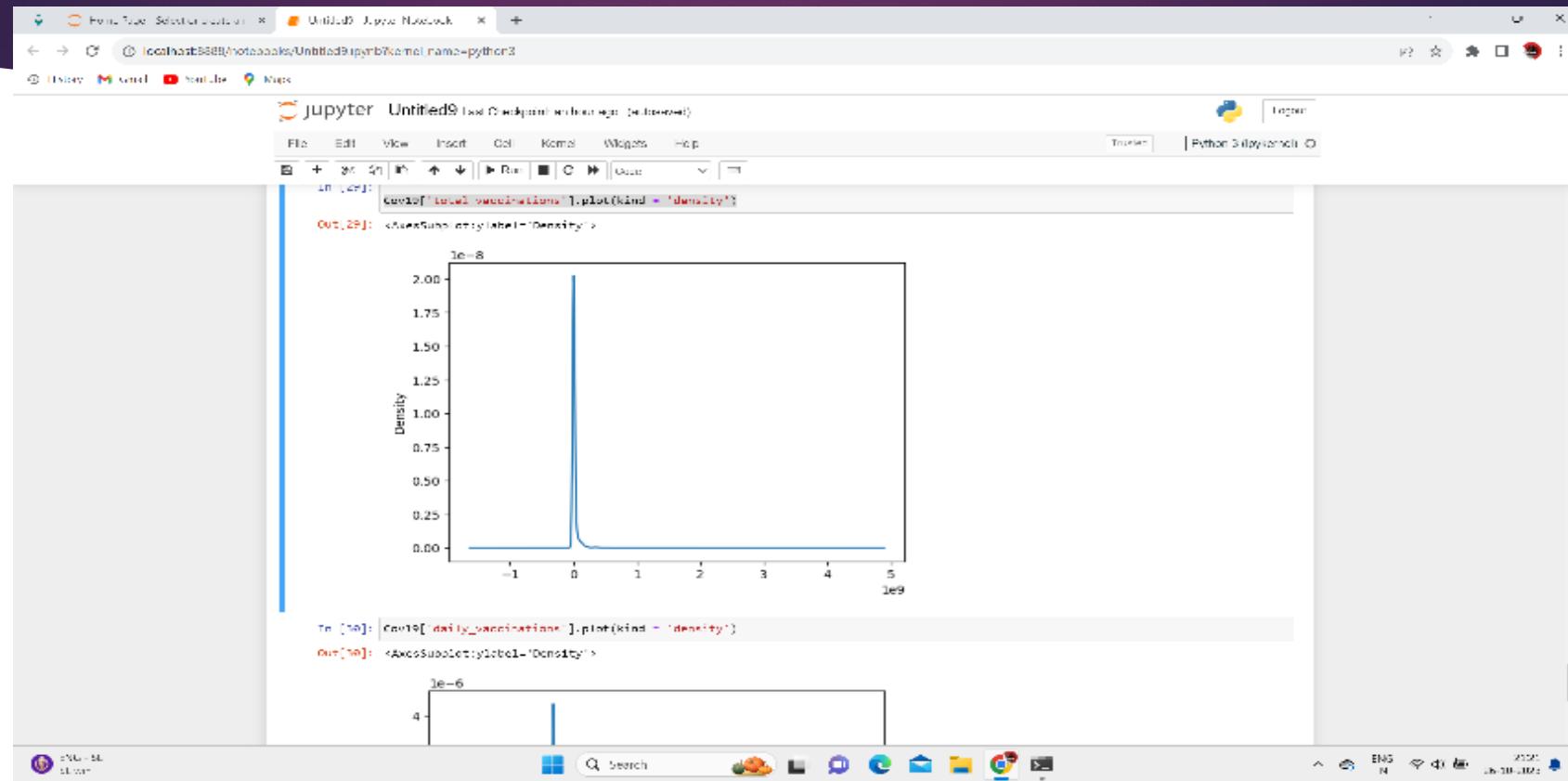
In [29]: Cov19['total_vaccinations'].plot(kind = 'density')

Out[29]: <AxesSubplot:ylabel='Density'>

ENG - SL SL win

Search

21:21 26.10.2023



3: Visualization

- ☒ Visualization (graphics), the physical or imagining creation of images, diagrams, or animations to communicate a message. Data and information visualization, the practice of creating visual representations of complex data and information.
- ☒ Like:
 - ☒ Linear Regression
 - ☒ Kmeans
 - ☒ Decision Tree
 - ☒ KNN,etc..

Using Linear Regression

The screenshot shows a Jupyter Notebook interface running on a web browser. The title bar indicates the URL is `localhost:8888/notebooks/Untitled8.ipynb?kernel_name=python3`. The notebook has tabs for Home Page, Select or create a..., Machine Learning Tutorial Py, cov19.99%, Untitled8 Jupyter Notebook, and another Untitled8 tab.

The notebook content consists of several code cells:

```
In [58]: LR=LinearRegression()
LR.fit(x_train, y_train)
Out[58]: LinearRegression()

In [59]: LR.intercept_
Out[59]: array([-7968929.46839611])

In [60]: LR.coef_
Out[60]: array([[-4.02667764e+03,  2.01303351e+04,  2.44710269e+00,
   -5.67486912e-01, -4.07149624e+01,  1.02571312e+01,
   9.70753821e+05, -1.00669587e+06, -1.07084487e+06,
   3.63885862e+02,  4.22525315e+04,  5.33569491e+04,
   3.44501497e+04]])

In [61]: coefficients = pd.DataFrame([x_train.columns,LR.coef_]).T
coefficients = coefficients.rename(columns={0: 'Attribute',1: 'Coefficients'})
coefficients
```

The output of cell In [61] is a DataFrame:

	Attribute	Coefficients
0	country	-4026.6776378615473, 20130.3351453798813, 2.44...
1	iso_code	None
2	people_vaccinated	None
3	people_fully_vaccinated	None
4	daily_vaccinations_raw	None
5	daily_vaccinations	None
6	total_vaccinations_per_hundred	None
7	people_vaccinated_per_hundred	None
8	people_fully_vaccinated_per_hundred	None
9	daily_vaccinations_per_million	None
10	vaccines	None

The status bar at the bottom shows the date and time as 23:10 25.10.2023, along with system icons for battery, signal, and network.

Home Page | Select or create a | Machine Learning Tutorial Py | cov19.99% | Kaggle | Untitled8 Jupyter Notebook | +

localhost:8888/notebooks/Untitled8.ipynb?kernel_name=python3

History Gmail YouTube Maps

jupyter Untitled8 Last Checkpoint: 33 minutes ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

In [62]: `y_test`

Out[62]:

	total_vaccinations
49657	1237673.0
43843	3108749.0
84854	4496930.0
28824	1990501.0
11597	588206.0
...	...
32360	128348.0
63524	4922055.0
77482	108084520.0
53989	9313.0
16929	637961.0

21628 rows × 1 columns

In [63]: `y_pred_LR=LR.predict(x_test)`

Out[63]:

```
array([[-2275725.63010717],
       [ 1175176.87498755],
       [-5476102.99220058],
       ...,
       [88590249.83697729],
       [ 3863387.93446322],
       [-5759112.88177419]])
```

In [64]: # Model Evaluation

25°C Partly cloudy

Search

23:10 25-10-2023 ENG IN

Home Page Select or create a... Machine Learning Tutorial Py cov19_90%@|Kaggle Untitled8 Jupyter Notebook

localhost:8888/notebooks/Untitled8.ipynb?kernel_name=python3

History Gmail YouTube Maps

jupyter Untitled8 Last Checkpoint: 33 minutes ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Run Code

```
[88590249.83697729],  
[ 3063387.93446321],  
[-5759112.88177419]]
```

In [64]: `print('R^2:',metrics.r2_score(y_test, y_pred_LR))
print('MAE:',metrics.mean_absolute_error(y_test, y_pred_LR))
print('MSE:',metrics.mean_squared_error(y_test, y_pred_LR))
print('RMSE:',np.sqrt(metrics.mean_squared_error(y_test, y_pred_LR)))`

R²: 0.8719183276933511
MAE: 11884018.238719486
MSE: 3581668998225382.5
RMSE: 59847046.69593398

In [65]: `plt.scatter(y_test, y_pred_LR)
plt.xlabel("total vaccines")
plt.ylabel("Predicted total vaccines")
plt.title("TOTAL VACCINES vs Predicted TOTAL VACCINES with LR")
plt.show()`

TOTAL VACCINES vs Predicted TOTAL VACCINES with LR

25°C Partly cloudy

Search

23:10 25.10.2023

2.2 Decision Tree Algorithm

The screenshot shows a Jupyter Notebook interface running on a local host. The notebook has tabs for Home Page, Select or create a, Machine Learning Tutorial Py, cov19.99%, Untitled8, and Jupyter Notebook. The Untitled8 tab is active.

The notebook content includes:

- In [66]: regressor = DecisionTreeRegressor(random_state = 0)
regressor.fit(x_train, y_train)
- Out[66]: DecisionTreeRegressor(random_state=0)
- In [67]: y_pred_DT=regressor.predict(x_test)
y_pred_DT
- Out[67]: array([1.23767300e+06, 3.11151900e+06, 4.50069200e+06, ..., 1.08524424e+08, 9.31300000e+05, 6.37961000e+05])
- In [68]: y_test
- Out[68]:

	total_vaccinations
49657	1237673.0
43843	3108749.0
84854	4498930.0
28824	1990501.0
11597	588206.0
...	...
32360	128348.0
63524	4922055.0
77482	108094820.0
53989	9313.0
16929	637961.0

21628 rows × 1 columns

In [69]: print('R^2:',metrics.r2_score(y_test, y_pred_DT))
print('MAE:',metrics.mean_absolute_error(y_test, y_pred_DT))
print('MSE:',metrics.mean_squared_error(y_test, y_pred_DT))
print('RMSE:',np.sqrt(metrics.mean_squared_error(y_test, y_pred_DT)))

The system tray at the bottom shows: 25°C Partly cloudy, ENG IN, 23:10, 25 10-2023.

Home Page Select or create a | Machine Learning Tutorial Py | cov19.99% | Kaggle | Untitled8 Jupyter Notebook +

localhost:8888/notebooks/Untitled8.ipynb?kernel_name=python3

History Gmail YouTube Maps

jupyter Untitled8 Last Checkpoint: 33 minutes ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

In [69]: `print('R^2:',metrics.r2_score(y_test, y_pred_DT))
print('MAE:',metrics.mean_absolute_error(y_test, y_pred_DT))
print('MSE:',metrics.mean_squared_error(y_test, y_pred_DT))
print('RMSE:',np.sqrt(metrics.mean_squared_error(y_test, y_pred_DT)))`

R^2: 0.9999132840454302
MAE: 156448.73247641945
MSE: 2424928291410.6694
RMSE: 1557215.557143798

In [70]: `plt.scatter(y_test, y_pred_DT)
plt.xlabel("total vaccines")
plt.ylabel("Predicted total vaccines")
plt.title("TOTAL VACCINES vs Predicted TOTAL VACCINES in DT")
plt.show()`

1e9TOTAL VACCINES vs Predicted TOTAL VACCINES in DT

25°C Partly cloudy

Search

23:10 25-10-2023 ENG IN

