Project 1: Credit Card Fraud Detection

Phase 1: Problem Definition and Design Thinking

In order to effectively test, detect, validate, correct error and monitor control systems against fraudulent activities, businesses entities and organizations rely on specialized data analytics techniques such as data mining, data matching, the sounds like function, regression analysis, clustering analysis etc

Problem Definition:

The Credit Card Fraud Detection Problem includes modeling past credit card transactions with the knowledge of the ones that turned out to be fraud.

Design Thinking:

Data Source:

When we make any transaction while purchasing any product online a good amount of people prefer credit cards. The credit limit in credit cards sometimes helps us me making purchases even if we don't have the amount at that time.

Data Preprocessing:

Data source resulted from preprocessing affects directly the quality of data mining. The methods are not the same according to particular application fields and industries.

Feature Engineering:

Create additional features that could enhance fraud detection, such as transaction frequency and amount deviations

Model Selection:

Choose suitable machine learning algorithms (e.g., Logistic Regression, Random Forest, Gradient Boosting) for fraud detection.

Model Training:

Train the selected model using the preprocessed data.

IBM –NAAN MUDHALVAN - APPLIED DATA SCIENCE

Project name: Credit Card Fraud Detection

Phase 2: Innovation

INTRODUCTION :

Fraud detection and prevention tools involve risk mitigation by monitoring and reviewing user actions. Device fingerprinting, social media lookup, and data enrichment are part of the fraud detection and transaction fraud detection tools. I hope you got a good intuition about isolation forest algorithm and how it achieves high anomaly detection performance reducing swamping and masking by leveraging the subsampling technique.

- Certainly, advanced techniques like Isolation Forest, One-Class SVM, and ensemble methods can be very effective for improving fraud detection accuracy.

- Isolation Forest is great at isolating anomalies, while One-Class SVM can be used to model the normal behavior and detect deviations. Ensembles, such as Random Forest or boosting, can combine multiple models for enhanced performance.

- If you have specific questions about implementing these techniques or need more information, feel free to ask!

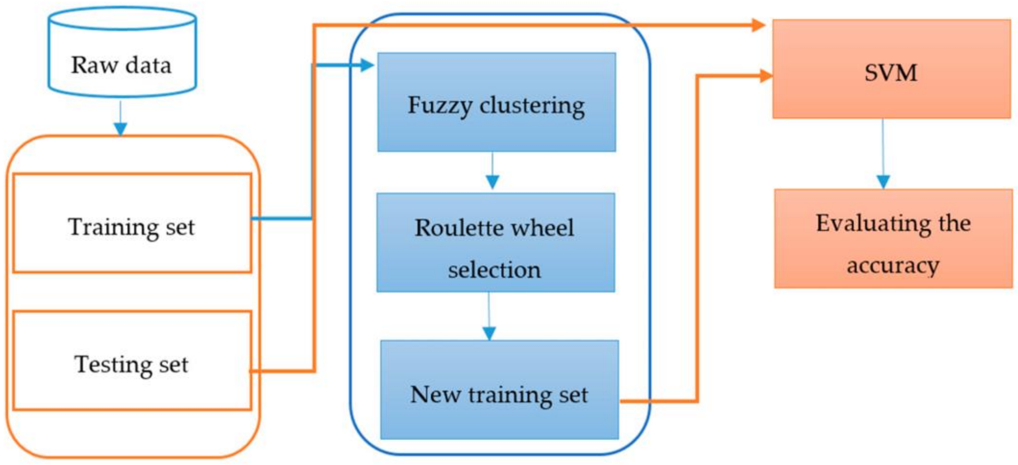
- Advanced Fraud Detection TechniquesUser
  Consider exploring advanced techniques such as anomaly detection algorithms (e.g., Isolation Forest, One-Class SVM) and ensemble methods for improved fraud detection accuracy.

- Certainly, using advanced techniques like Isolation Forest, One-Class SVM, and ensemble methods can enhance fraud detection accuracy. These methods work well for anomaly detection in fraud detection systems.

  Isolation Forest: It’s effective at isolating anomalies by creating a binary tree structure. Anomalies are isolated closer to the root, making them easier to detect.

  One-Class SVM: This algorithm is suitable for one-class classification, making it useful for detecting anomalies in imbalanced datasets, which is common in fraud detection.

  Ensemble Methods: Combining multiple models can improve accuracy. Techniques like Random Forest or Gradient Boosting can be used to create an ensemble of models for more robust fraud detection.



To implement these techniques:

Data Preprocessing: Ensure your data is well-preprocessed and features are properly engineered.

Model Selection: Choose the appropriate anomaly detection algorithm(s) and ensemble methods for your dataset. Experiment with different algorithms to find the best fit.

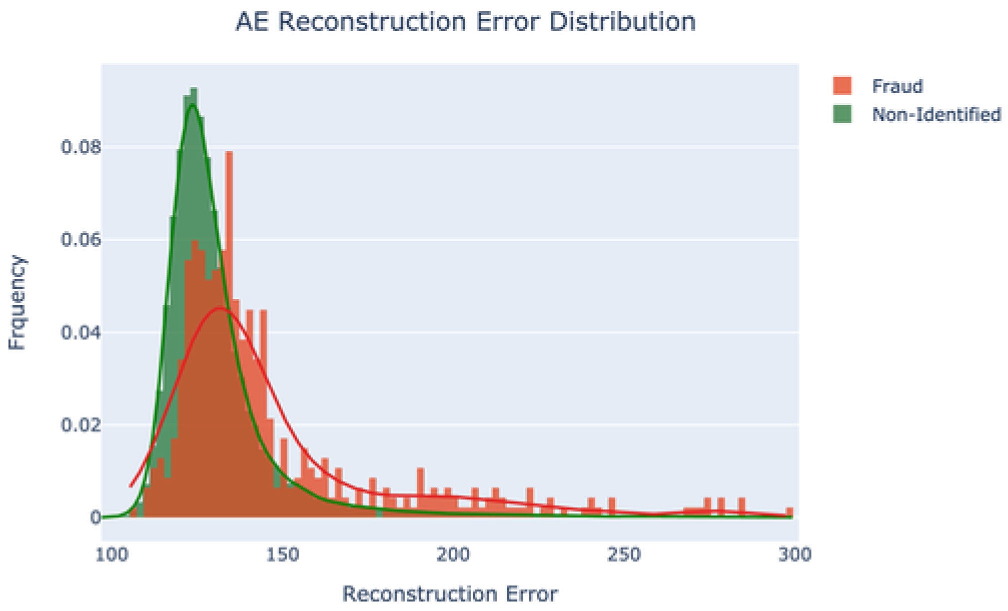
Hyperparameter Tuning: Fine-tune the hyperparameters for each algorithm to optimize performance.

Cross-Validation: Use cross-validation to assess the models' generalization and avoid overfitting.

Monitoring and Updating: Continuously monitor the performance of your fraud detection system and update it as new data becomes available.

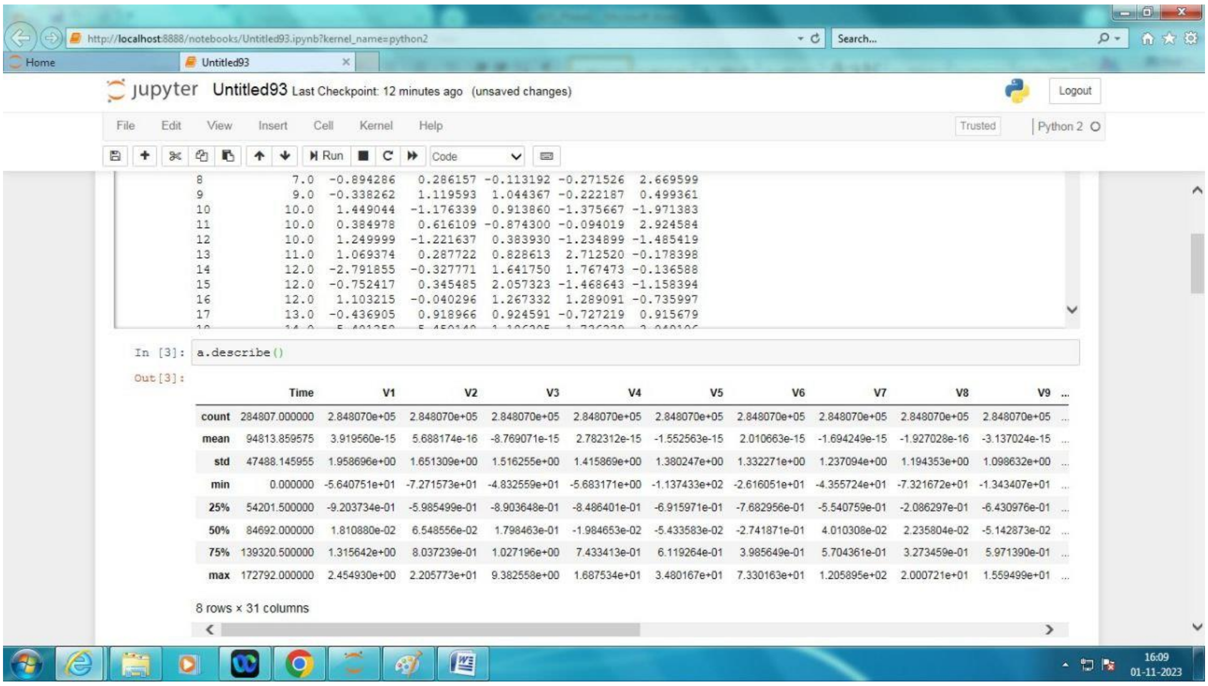
Remember that fraud patterns can change over time, so a robust and adaptable system is essential.

One-class SVM (SGD) – Builds on the one-class SVM algorithm using Stochastic Gradient Descent (SGD). Isolation forest – Uses decision trees to continuously split or divide the data to eventually isolate anomalous data points.


AE Reconstruction Error Distribution
Fraud
Non-Identified
Frquency
0.08
0.06
0.04
0.02
0
100
150
200
250
300
Reconstruction Error

Phase3:Development part1

Importing CSV file :



```
8             7.0 -0.894286  0.286157 -0.113192 -0.271526  2.669599
9             9.0 -0.338262  1.119593  1.044367 -0.222187  0.499361
10           10.0  1.449044 -1.176339  0.913860 -1.375667 -1.971383
11           10.0  0.384978  0.616109 -0.874300 -0.094019  2.924584
12           10.0  1.249999 -1.221637  0.383930 -1.234899 -1.485419
13           11.0  1.069374  0.287722  0.828613  2.712520 -0.178398
14           12.0 -2.791855 -0.327771  1.641750  1.767473 -0.136588
15           12.0 -0.752417  0.345485  2.057323 -1.468643 -1.158394
16           12.0  1.103215 -0.040296  1.267332  1.289091 -0.735997
17           13.0 -0.436905  0.918966  0.924591 -0.727219  0.915679
```

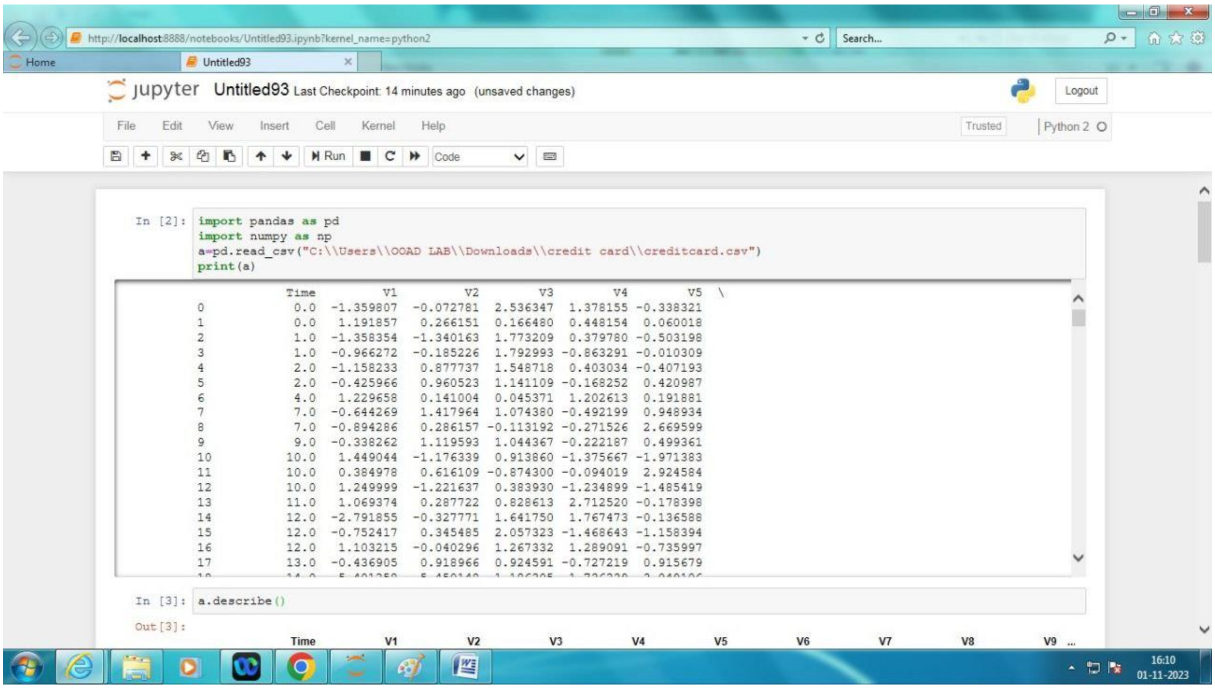
```
In [3]: a.describe()
```

Out[3]:

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...
count	284807.000000	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	...
mean	94813.859575	3.919560e-15	5.688174e-16	-8.769071e-15	2.782312e-15	-1.552563e-15	2.010663e-15	-1.694249e-15	-1.927028e-16	-3.137024e-15	...
std	47488.145955	1.958696e+00	1.651309e+00	1.516255e+00	1.415869e+00	1.380247e+00	1.332271e+00	1.237094e+00	1.194353e+00	1.098632e+00	...
min	0.000000	-5.640751e+01	-7.271573e+01	-4.832559e+01	-5.683171e+00	-1.137433e+02	-2.616051e+01	-4.355724e+01	-7.321672e+01	-1.343407e+01	...
25%	54201.500000	-9.203734e-01	-5.985499e-01	-8.903648e-01	-8.486401e-01	-6.915971e-01	-7.682956e-01	-5.540759e-01	-2.086297e-01	-6.430976e-01	...
50%	84692.000000	1.810880e-02	6.548556e-02	1.798463e-01	-1.984653e-02	-5.433583e-02	-2.741871e-01	4.010308e-02	2.235804e-02	-5.142873e-02	...
75%	139320.500000	1.315642e+00	8.037239e-01	1.027196e+00	7.433413e-01	6.119264e-01	3.985649e-01	5.704361e-01	3.273459e-01	5.971390e-01	...
max	172792.000000	2.454930e+00	2.205773e+01	9.382558e+00	1.687534e+01	3.480167e+01	7.330163e+01	1.205895e+02	2.000721e+01	1.559499e+01	...

8 rows × 31 columns

Preprocessing:



```
In [2]: import pandas as pd
        import numpy as np
        a=pd.read_csv("C:\\Users\\OOAD LAB\\Downloads\\credit card\\creditcard.csv")
        print(a)
```

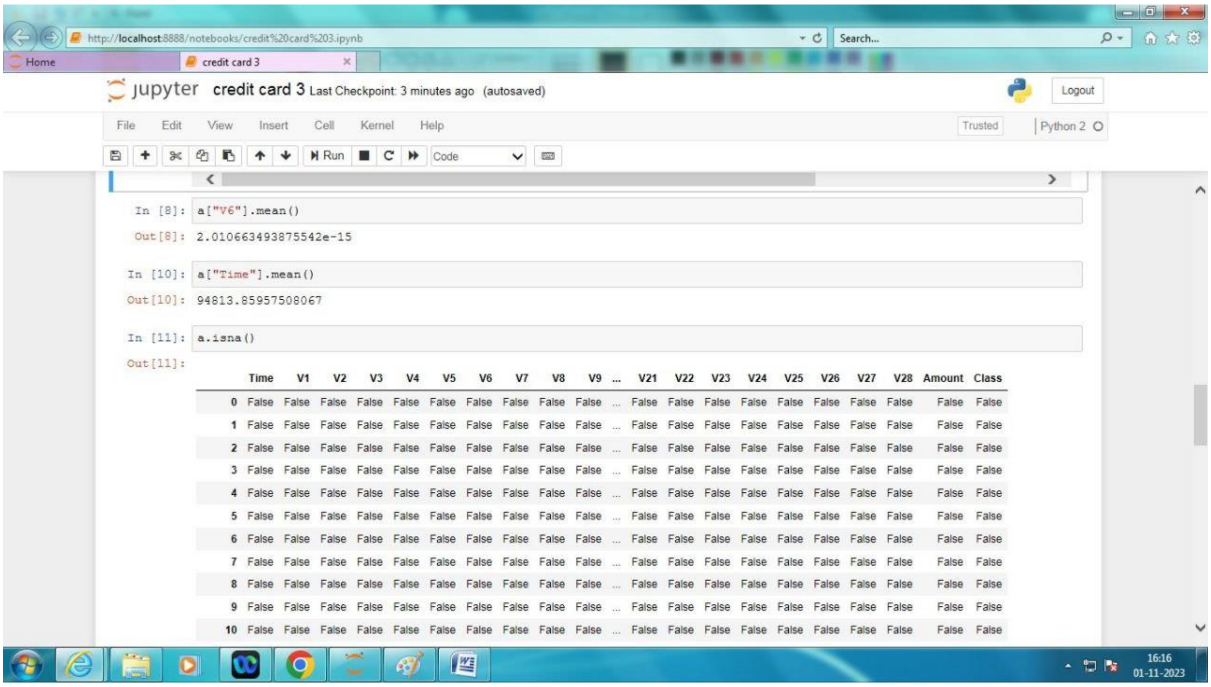
```
             Time        V1        V2        V3        V4        V5  \
0             0.0 -1.359807 -0.072781  2.536347  1.378155 -0.338321
1             0.0  1.191857  0.266151  0.166480  0.448154  0.060018
2             1.0 -1.358354 -1.340163  1.773209  0.379780 -0.503198
3             1.0 -0.966272 -0.185226  1.792993 -0.863291 -0.010309
4             2.0 -1.158233  0.877737  1.548718  0.403034 -0.407193
5             2.0 -0.425966  0.960523  1.141109 -0.168252  0.420987
6             4.0  1.229658  0.141004  0.045371  1.202613  0.191881
7             7.0 -0.644269  1.417964  1.074380 -0.492199  0.948934
8             7.0 -0.894286  0.286157 -0.113192 -0.271526  2.669599
9             9.0 -0.338262  1.119593  1.044367 -0.222187  0.499361
10           10.0  1.449044 -1.176339  0.913860 -1.375667 -1.971383
11           10.0  0.384978  0.616109 -0.874300 -0.094019  2.924584
12           10.0  1.249999 -1.221637  0.383930 -1.234899 -1.485419
13           11.0  1.069374  0.287722  0.828613  2.712520 -0.178398
14           12.0 -2.791855 -0.327771  1.641750  1.767473 -0.136588
15           12.0 -0.752417  0.345485  2.057323 -1.468643 -1.158394
16           12.0  1.103215 -0.040296  1.267332  1.289091 -0.735997
17           13.0 -0.436905  0.918966  0.924591 -0.727219  0.915679
```

```
In [3]: a.describe()
```

Out[3]:

```
         Time     V1     V2     V3     V4     V5     V6     V7     V8     V9 ...
```

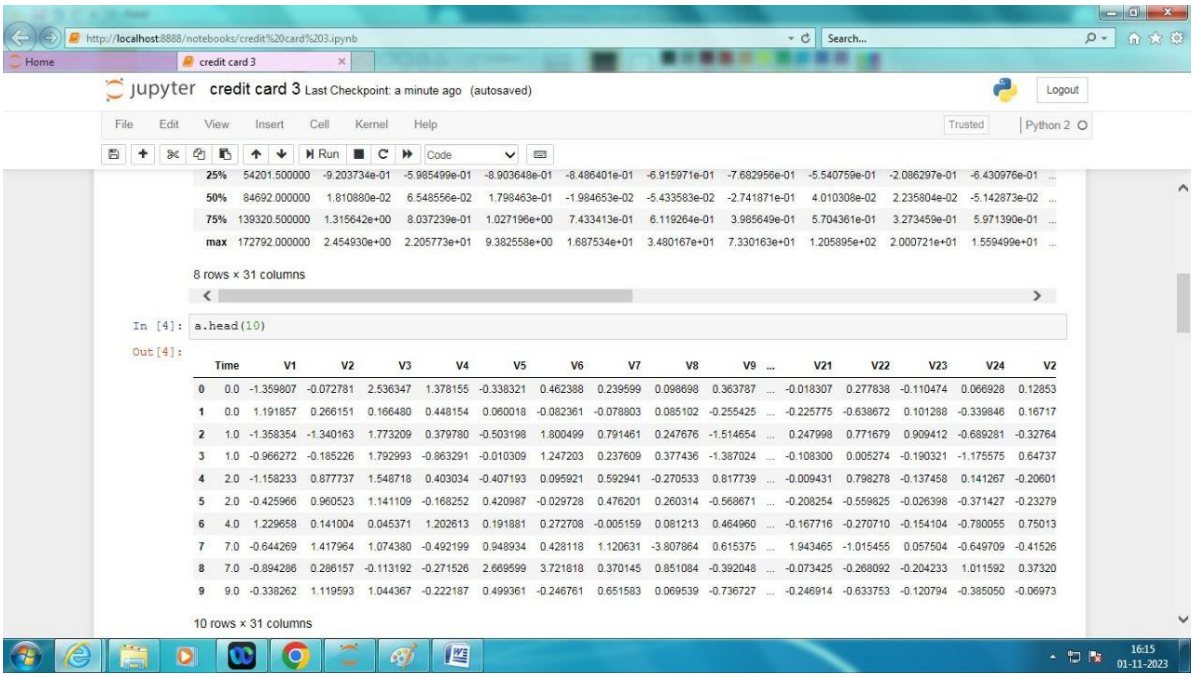
Preforming various operation:



credit card 3 Last Checkpoint: 3 minutes ago (autosaved)

```
In [8]: a["V6"].mean()
Out[8]: 2.010663493875542e-15

In [10]: a["Time"].mean()
Out[10]: 94813.85957508067

In [11]: a.isna()
Out[11]:
```

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25	V26	V27	V28	Amount	Class
0	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
5	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
6	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
7	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
8	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
9	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False
10	False	False	False	False	False	False	False	False	False	False	...	False	False	False	False	False	False	False	False	False	False



credit card 3 Last Checkpoint: a minute ago (autosaved)

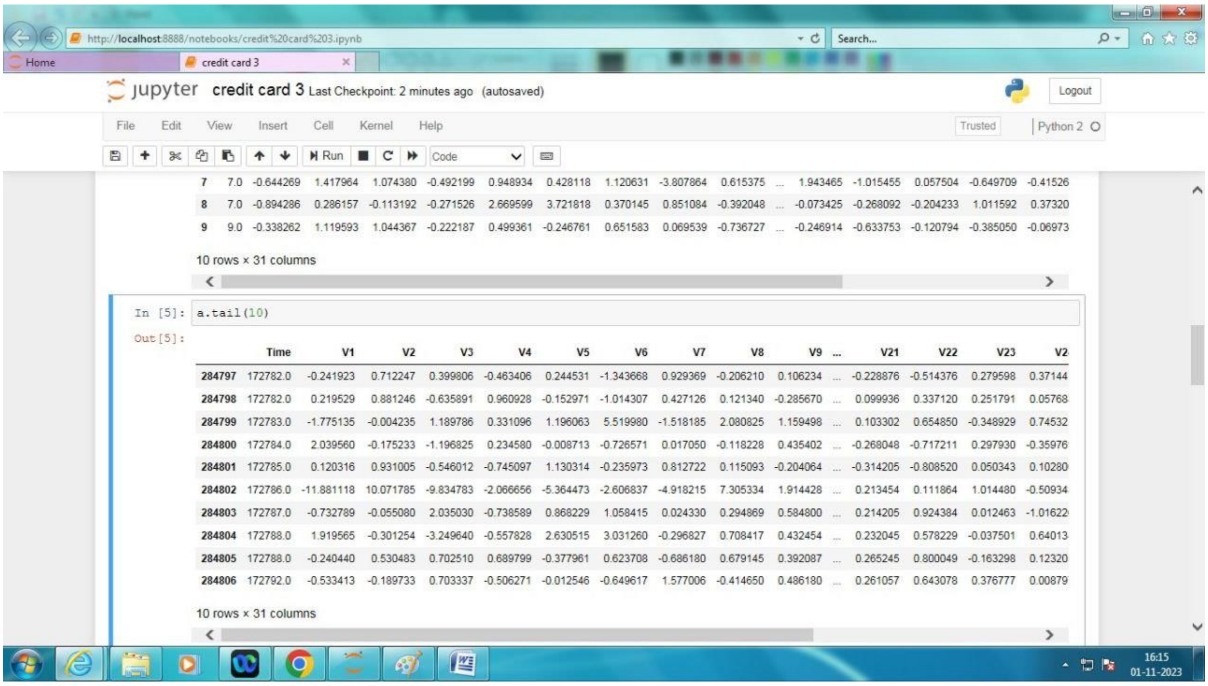
25%	54201.500000	-9.203734e-01	-5.985499e-01	-8.903648e-01	-8.486401e-01	-6.915971e-01	-7.682956e-01	-5.540759e-01	-2.086297e-01	-6.430976e-01	...
50%	84692.000000	1.810880e-02	6.548556e-02	1.798463e-01	-1.984653e-02	-5.433583e-02	-2.741871e-01	4.010308e-02	2.235804e-02	-5.142873e-02	...
75%	139320.500000	1.315642e+00	8.037239e-01	1.027196e+00	7.433413e-01	6.119264e-01	3.985649e-01	5.704361e-01	3.273459e-01	5.971390e-01	...
max	172792.000000	2.454930e+00	2.205773e+01	9.382558e+00	1.687534e+01	3.480167e+01	7.330163e+01	1.205895e+02	2.000721e+01	1.559499e+01	...

8 rows × 31 columns

```
In [4]: a.head(10)
Out[4]:
```

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V2
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928	0.12853
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846	0.16717
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281	-0.32764
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575	0.64737
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141267	-0.20601
5	2.0	-0.425966	0.960523	1.141109	-0.168252	0.420987	-0.029728	0.476201	0.260314	-0.568671	...	-0.208254	-0.559825	-0.026398	-0.371427	-0.23279
6	4.0	1.229658	0.141004	0.045371	1.202613	0.191881	0.272708	-0.005159	0.081213	0.464960	...	-0.167716	-0.270710	-0.154104	-0.780055	0.75013
7	7.0	-0.644269	1.417964	1.074380	-0.492199	0.948934	0.428118	1.120631	-3.807864	0.615375	...	1.943465	-1.015455	0.057504	-0.649709	-0.41526
8	7.0	-0.894286	0.286157	-0.113192	-0.271526	2.669599	3.721818	0.370145	0.851084	-0.392048	...	-0.073425	-0.268092	-0.204233	1.011592	0.37320
9	9.0	-0.338262	1.119593	1.044367	-0.222187	0.499361	-0.246761	0.651583	0.069539	-0.736727	...	-0.246914	-0.633753	-0.120794	-0.385050	-0.06973

10 rows × 31 columns


http://localhost:8888/notebooks/credit%20card%203.ipynb
Search...
Home
credit card 3
jupyter
credit card 3 Last Checkpoint: 2 minutes ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Help
Trusted
Python 2
Run
Code
16:15
01-11-2023

7	7.0	-0.644269	1.417964	1.074380	-0.492199	0.948934	0.428118	1.120631	-3.807864	0.615375	...	1.943465	-1.015455	0.057504	-0.649709	-0.41526
8	7.0	-0.894286	0.286157	-0.113192	-0.271526	2.669599	3.721818	0.370145	0.851084	-0.392048	...	-0.073425	-0.268092	-0.204233	1.011592	0.37320
9	9.0	-0.338262	1.119593	1.044367	-0.222187	0.499361	-0.246761	0.651583	0.069539	-0.736727	...	-0.246914	-0.633753	-0.120794	-0.385050	-0.06973

10 rows × 31 columns

```
In [5]: a.tail(10)
```

Out[5]:

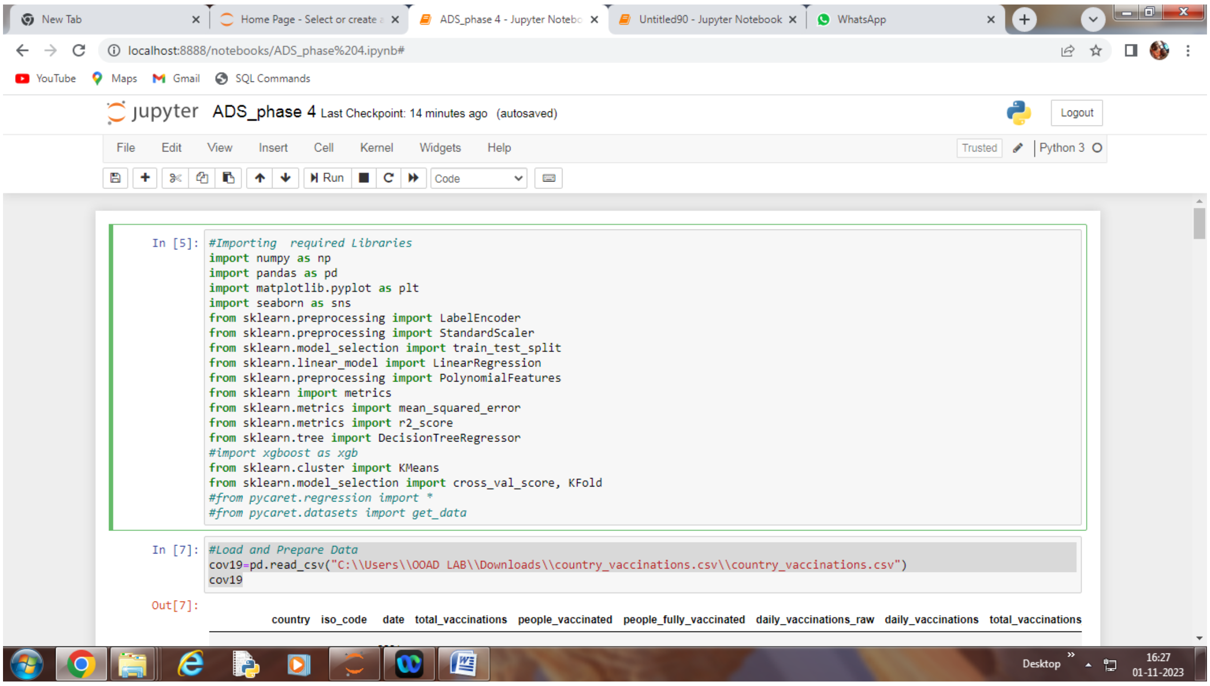
	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V2
284797	172782.0	-0.241923	0.712247	0.399806	-0.463406	0.244531	-1.343668	0.929369	-0.206210	0.106234	...	-0.228876	-0.514376	0.279598	0.37144
284798	172782.0	0.219529	0.881246	-0.635891	0.960928	-0.152971	-1.014307	0.427126	0.121340	-0.285670	...	0.099936	0.337120	0.251791	0.05768
284799	172783.0	-1.775135	-0.004235	1.189786	0.331096	1.196063	5.519980	-1.518185	2.080825	1.159498	...	0.103302	0.654850	-0.348929	0.74532
284800	172784.0	2.039560	-0.175233	-1.196825	0.234580	-0.008713	-0.726571	0.017050	-0.118228	0.435402	...	-0.268048	-0.717211	0.297930	-0.35976
284801	172785.0	0.120316	0.931005	-0.546012	-0.745097	1.130314	-0.235973	0.812722	0.115093	-0.204064	...	-0.314205	-0.808520	0.050343	0.10280
284802	172786.0	-11.881118	10.071785	-9.834783	-2.066656	-5.364473	-2.606837	-4.918215	7.305334	1.914428	...	0.213454	0.111864	1.014480	-0.50934
284803	172787.0	-0.732789	-0.055080	2.035030	-0.738589	0.868229	1.058415	0.024330	0.294869	0.584800	...	0.214205	0.924384	0.012463	-1.01622
284804	172788.0	1.919565	-0.301254	-3.249640	-0.557828	2.630515	3.031260	-0.296827	0.708417	0.432454	...	0.232045	0.578229	-0.037501	0.64013
284805	172788.0	-0.240440	0.530483	0.702510	0.689799	-0.377961	0.623708	-0.686180	0.679145	0.392087	...	0.265245	0.800049	-0.163298	0.12320
284806	172792.0	-0.533413	-0.189733	0.703337	-0.506271	-0.012546	-0.649617	1.577006	-0.414650	0.486180	...	0.261057	0.643078	0.376777	0.00879

10 rows × 31 columns

APPLIED DATA SCIENCE PHASE 4

Project: Covid-19 Vaccines Analysis

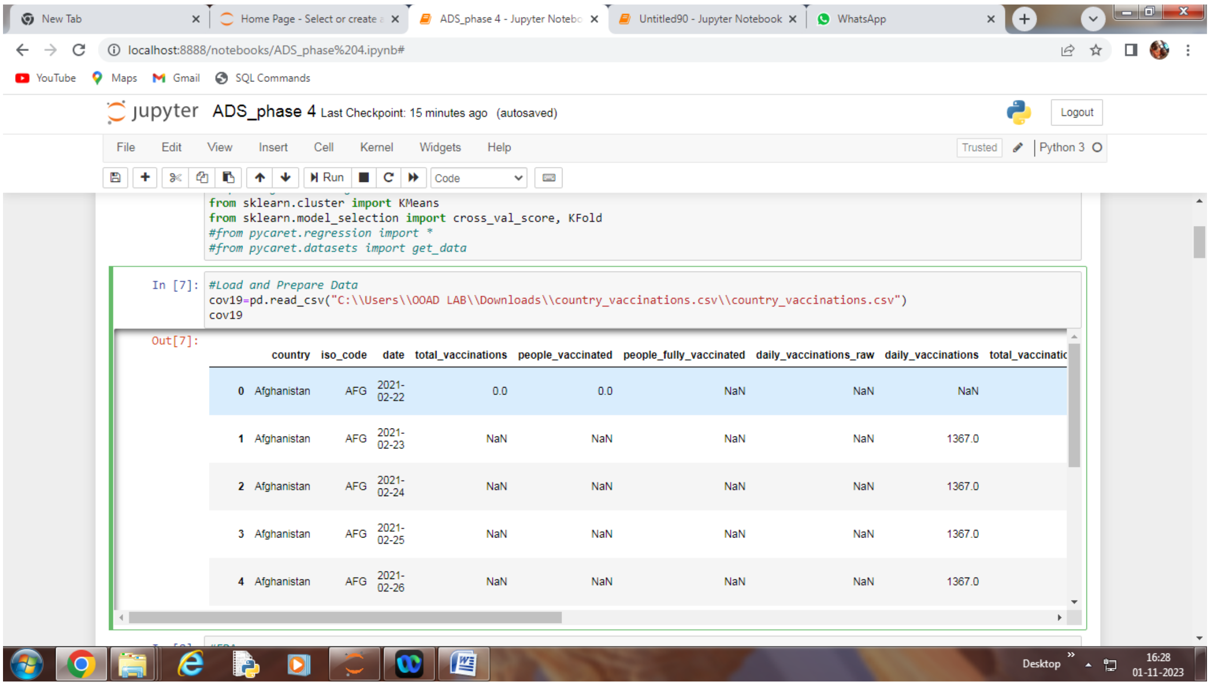
Importing Required Libraries:



New Tab
Home Page - Select or create a
ADS_phase 4 - Jupyter Notebo
Untitled90 - Jupyter Notebook
WhatsApp
localhost:8888/notebooks/ADS_phase%204.ipynb#
YouTube
Maps
Gmail
SQL Commands
jupyter ADS_phase 4 Last Checkpoint: 14 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Desktop
16:27
01-11-2023

```
In [5]: #Importing  required Libraries
        import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns
        from sklearn.preprocessing import LabelEncoder
        from sklearn.preprocessing import StandardScaler
        from sklearn.model_selection import train_test_split
        from sklearn.linear_model import LinearRegression
        from sklearn.preprocessing import PolynomialFeatures
        from sklearn import metrics
        from sklearn.metrics import mean_squared_error
        from sklearn.metrics import r2_score
        from sklearn.tree import DecisionTreeRegressor
        #import xgboost as xgb
        from sklearn.cluster import KMeans
        from sklearn.model_selection import cross_val_score, KFold
        #from pycaret.regression import *
        #from pycaret.datasets import get_data

In [7]: #Load and Prepare Data
        cov19=pd.read_csv("C:\\Users\\OOAD LAB\\Downloads\\country_vaccinations.csv\\country_vaccinations.csv")
        cov19

Out[7]:
           country  iso_code  date  total_vaccinations  people_vaccinated  people_fully_vaccinated  daily_vaccinations_raw  daily_vaccinations  total_vaccinations
```

Load and Prepare Data:



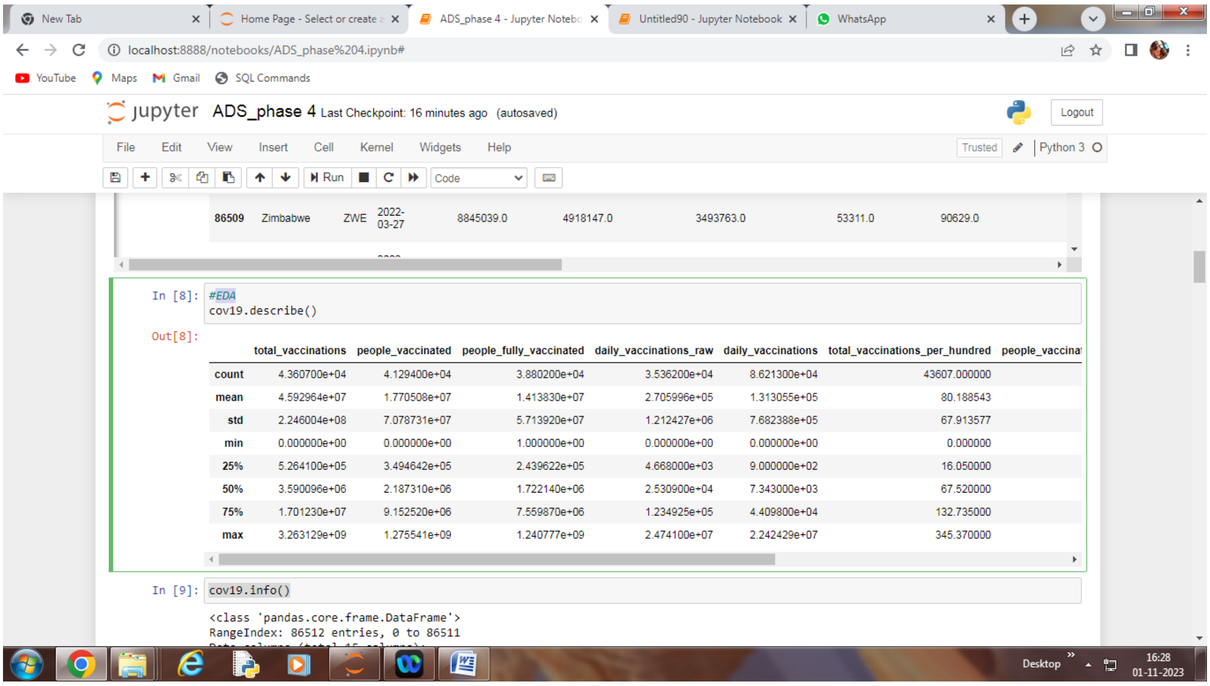
New Tab
Home Page - Select or create a
ADS_phase 4 - Jupyter Notebo
Untitled90 - Jupyter Notebook
WhatsApp
localhost:8888/notebooks/ADS_phase%204.ipynb#
YouTube
Maps
Gmail
SQL Commands
jupyter ADS_phase 4 Last Checkpoint: 15 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
Desktop
16:28
01-11-2023

```
        from sklearn.cluster import KMeans
        from sklearn.model_selection import cross_val_score, KFold
        #from pycaret.regression import *
        #from pycaret.datasets import get_data

In [7]: #Load and Prepare Data
        cov19=pd.read_csv("C:\\Users\\OOAD LAB\\Downloads\\country_vaccinations.csv\\country_vaccinations.csv")
        cov19
```

Out[7]:

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinatic
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	

EDA:


New Tab
Home Page - Select or create
ADS_phase 4 - Jupyter Notebo
Untitled90 - Jupyter Notebook
WhatsApp
localhost:8888/notebooks/ADS_phase%204.ipynb#
YouTube
Maps
Gmail
SQL Commands
jupyter
ADS_phase 4 Last Checkpoint: 16 minutes ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
Desktop
16:28
01-11-2023

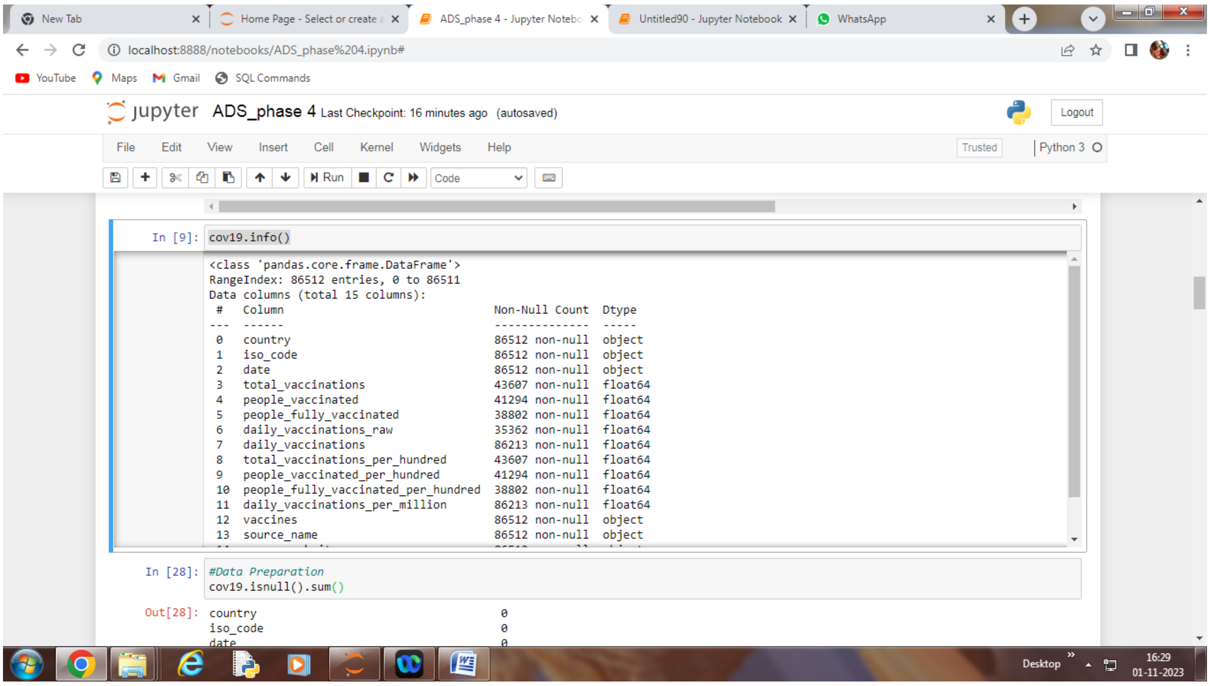
86509	Zimbabwe	ZWE	2022-03-27	8845039.0	4918147.0	3493763.0	53311.0	90629.0

```
In [8]: #EDA
        cov19.describe()
```

Out[8]:

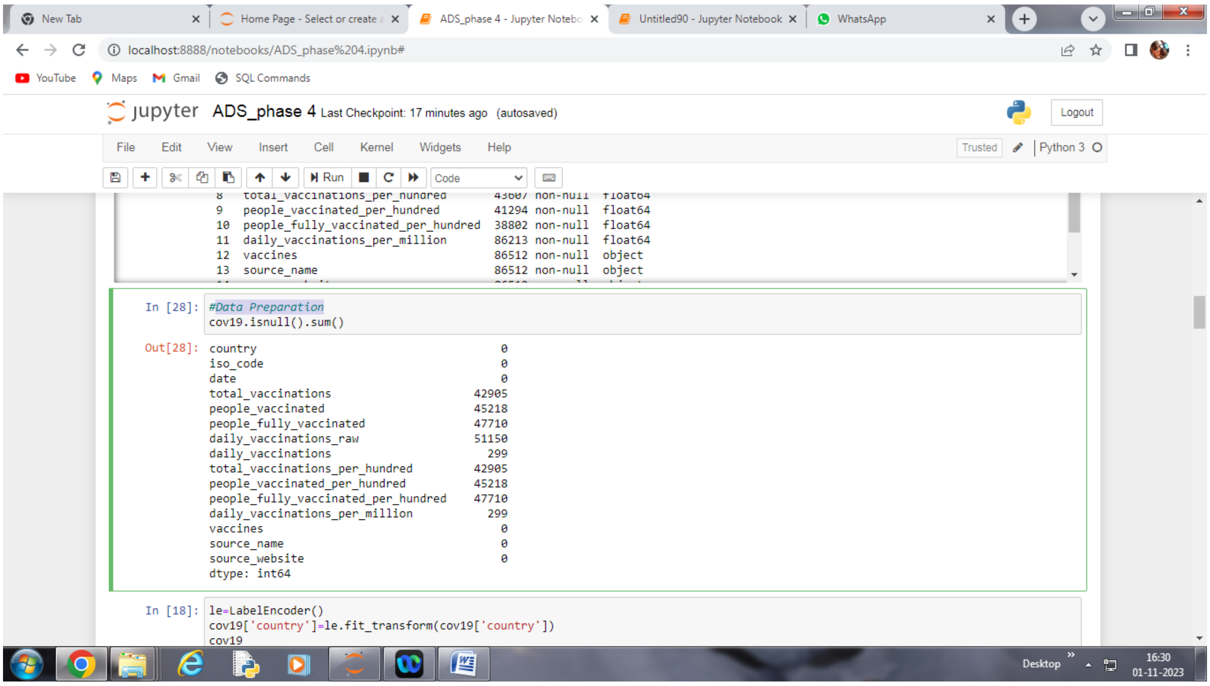
	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per_hundred	people_vaccinat
count	4.360700e+04	4.129400e+04	3.880200e+04	3.536200e+04	8.621300e+04	43607.000000	
mean	4.592964e+07	1.770508e+07	1.413830e+07	2.705996e+05	1.313055e+05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

```
In [9]: cov19.info()

        <class 'pandas.core.frame.DataFrame'>
        RangeIndex: 86512 entries, 0 to 86511
```


New Tab
Home Page - Select or create
ADS_phase 4 - Jupyter Notebo
Untitled90 - Jupyter Notebook
WhatsApp
localhost:8888/notebooks/ADS_phase%204.ipynb#
YouTube
Maps
Gmail
SQL Commands
jupyter
ADS_phase 4 Last Checkpoint: 16 minutes ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
Desktop
16:29
01-11-2023

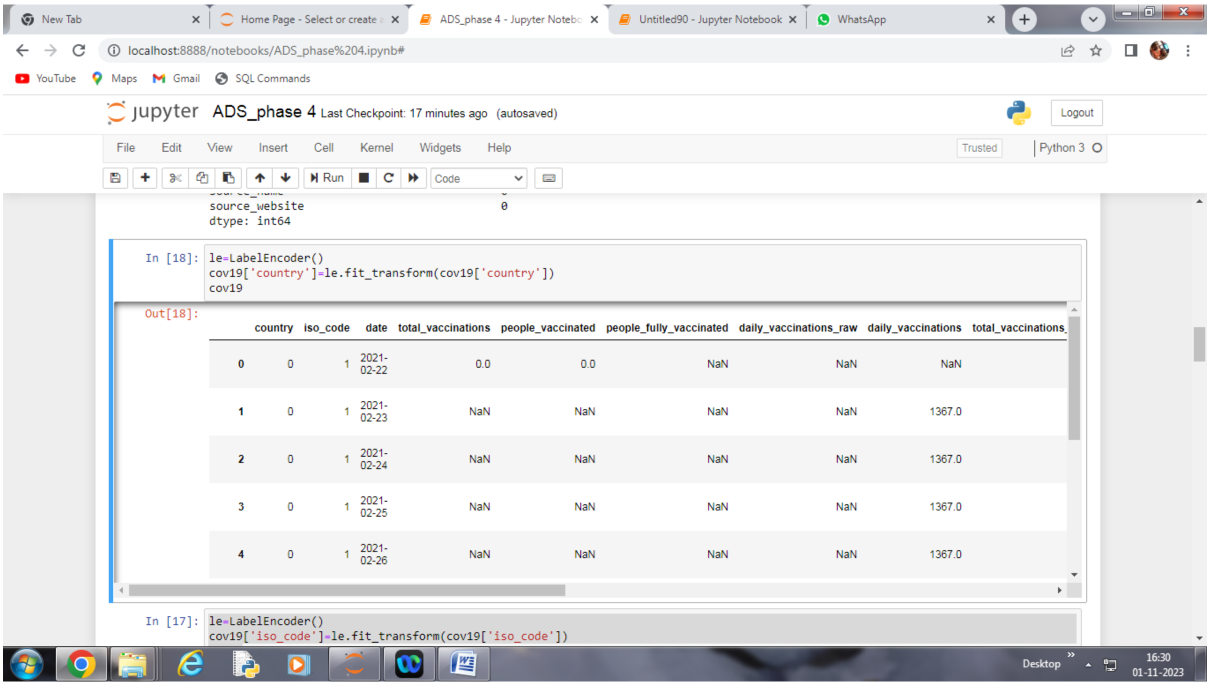
```
In [9]: cov19.info()

        <class 'pandas.core.frame.DataFrame'>
        RangeIndex: 86512 entries, 0 to 86511
        Data columns (total 15 columns):
         #   Column                               Non-Null Count  Dtype
        ---  ------                               --------------  -----
         0   country                              86512 non-null  object
         1   iso_code                             86512 non-null  object
         2   date                                 86512 non-null  object
         3   total_vaccinations                   43607 non-null  float64
         4   people_vaccinated                    41294 non-null  float64
         5   people_fully_vaccinated              38802 non-null  float64
         6   daily_vaccinations_raw               35362 non-null  float64
         7   daily_vaccinations                   86213 non-null  float64
         8   total_vaccinations_per_hundred       43607 non-null  float64
         9   people_vaccinated_per_hundred        41294 non-null  float64
         10  people_fully_vaccinated_per_hundred  38802 non-null  float64
         11  daily_vaccinations_per_million       86213 non-null  float64
         12  vaccines                             86512 non-null  object
         13  source_name                          86512 non-null  object

In [28]: #Data Preparation
         cov19.isnull().sum()

Out[28]: country                              0
         iso_code                             0
         date                                 0
```

Data Preparation:


New Tab
Home Page - Select or create
ADS_phase 4 - Jupyter Notebo
Untitled90 - Jupyter Notebook
WhatsApp
localhost:8888/notebooks/ADS_phase%204.ipynb#
YouTube
Maps
Gmail
SQL Commands
jupyter ADS_phase 4 Last Checkpoint: 17 minutes ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
Desktop
16:30
01-11-2023

```
 8   total_vaccinations_per_hundred       43607 non-null  float64
 9   people_vaccinated_per_hundred        41294 non-null  float64
 10  people_fully_vaccinated_per_hundred  38802 non-null  float64
 11  daily_vaccinations_per_million       86213 non-null  float64
 12  vaccines                             86512 non-null  object
 13  source_name                          86512 non-null  object
```

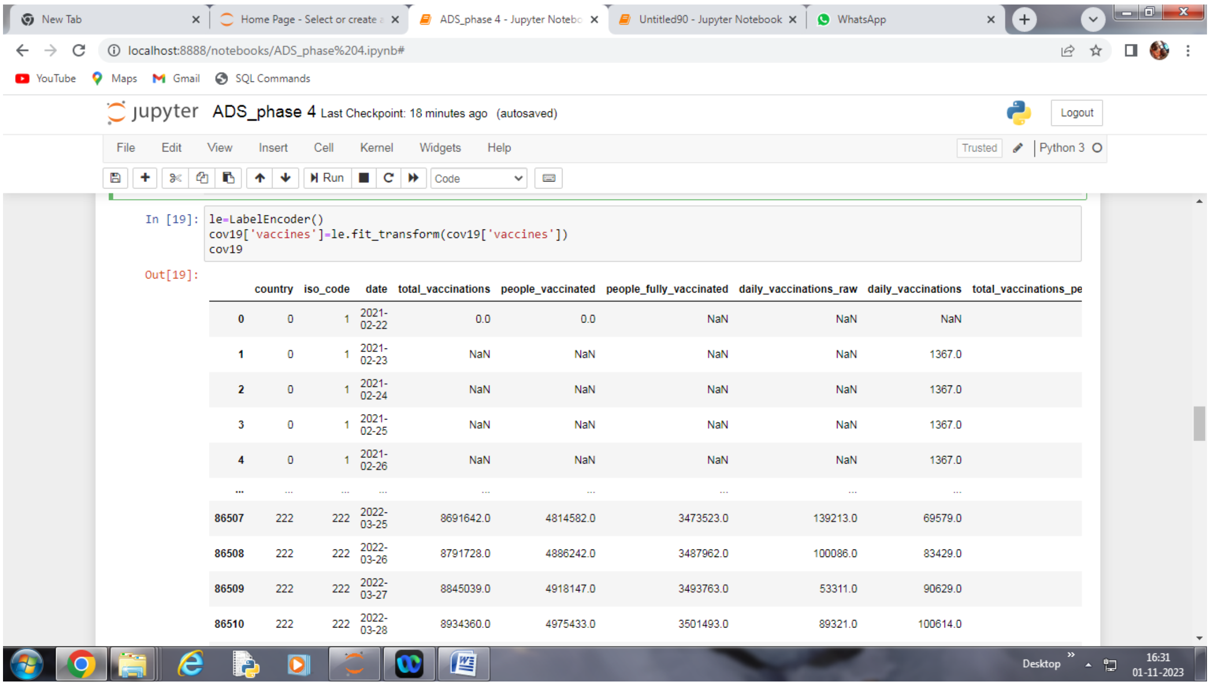
```
In [28]: #Data Preparation
         cov19.isnull().sum()

Out[28]: country                                   0
         iso_code                                  0
         date                                      0
         total_vaccinations                    42905
         people_vaccinated                     45218
         people_fully_vaccinated               47710
         daily_vaccinations_raw                51150
         daily_vaccinations                      299
         total_vaccinations_per_hundred        42905
         people_vaccinated_per_hundred         45218
         people_fully_vaccinated_per_hundred   47710
         daily_vaccinations_per_million          299
         vaccines                                  0
         source_name                               0
         source_website                            0
         dtype: int64

In [18]: le=LabelEncoder()
         cov19['country']=le.fit_transform(cov19['country'])
         cov19
```


New Tab
Home Page - Select or create
ADS_phase 4 - Jupyter Notebo
Untitled90 - Jupyter Notebook
WhatsApp
localhost:8888/notebooks/ADS_phase%204.ipynb#
YouTube
Maps
Gmail
SQL Commands
jupyter ADS_phase 4 Last Checkpoint: 17 minutes ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
Desktop
16:30
01-11-2023

```
         source_website                            0
         dtype: int64

In [18]: le=LabelEncoder()
         cov19['country']=le.fit_transform(cov19['country'])
         cov19

Out[18]:
```

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

```
In [17]: le=LabelEncoder()
         cov19['iso_code']=le.fit_transform(cov19['iso_code'])
```



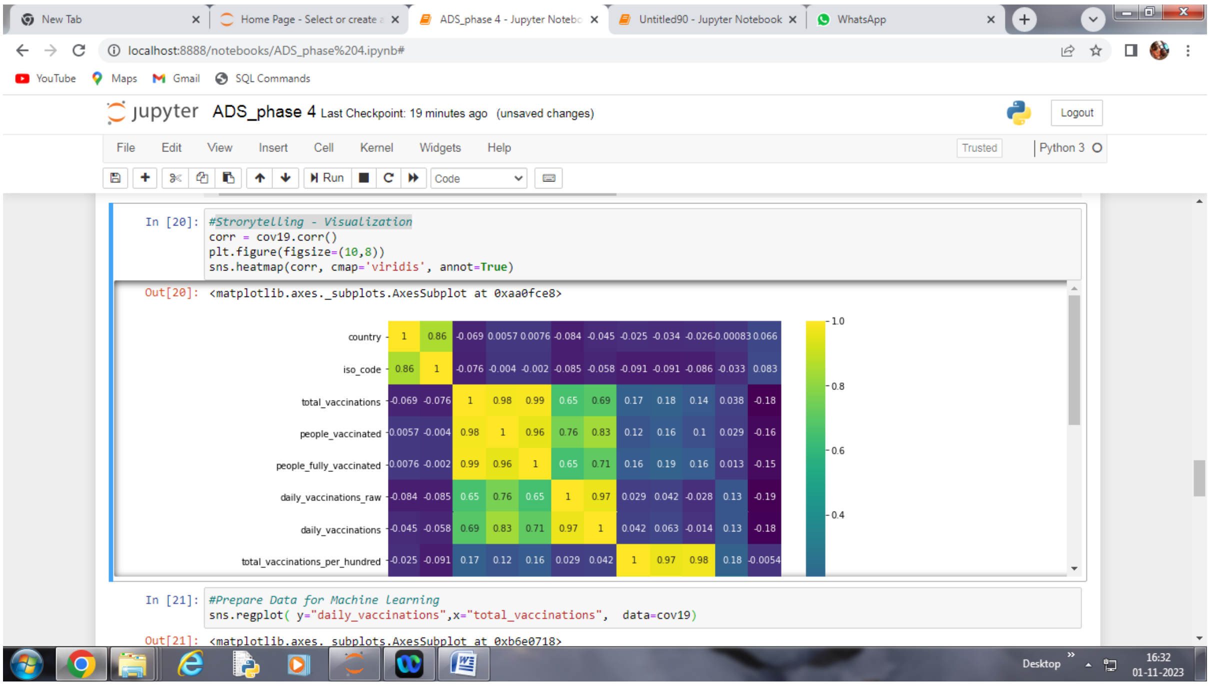
New Tab
Home Page - Select or create
ADS_phase 4 - Jupyter Notebo
Untitled90 - Jupyter Notebook
WhatsApp
localhost:8888/notebooks/ADS_phase%204.ipynb#
YouTube
Maps
Gmail
SQL Commands
jupyter
ADS_phase 4 Last Checkpoint: 18 minutes ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
Desktop
16:31
01-11-2023

```
In [19]: le=LabelEncoder()
         cov19['vaccines']=le.fit_transform(cov19['vaccines'])
         cov19
```

Out[19]:

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_pe
0	0	1	2021-02-22	0.0	0.0	NaN	NaN	NaN	
1	0	1	2021-02-23	NaN	NaN	NaN	NaN	1367.0	
2	0	1	2021-02-24	NaN	NaN	NaN	NaN	1367.0	
3	0	1	2021-02-25	NaN	NaN	NaN	NaN	1367.0	
4	0	1	2021-02-26	NaN	NaN	NaN	NaN	1367.0	
...	...	...	...	...	...	...	...	...	
86507	222	222	2022-03-25	8691642.0	4814582.0	3473523.0	139213.0	69579.0	
86508	222	222	2022-03-26	8791728.0	4886242.0	3487962.0	100086.0	83429.0	
86509	222	222	2022-03-27	8845039.0	4918147.0	3493763.0	53311.0	90629.0	
86510	222	222	2022-03-28	8934360.0	4975433.0	3501493.0	89321.0	100614.0	

Strorytelling – Visualization:



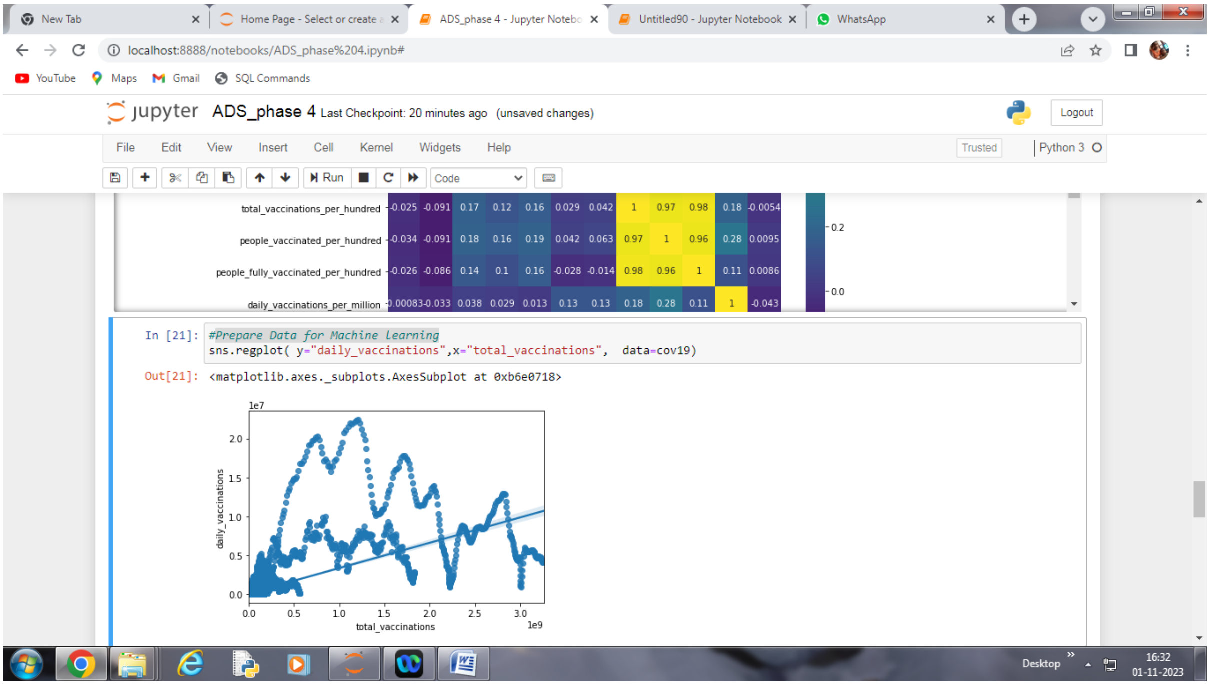
New Tab
Home Page - Select or create
ADS_phase 4 - Jupyter Notebo
Untitled90 - Jupyter Notebook
WhatsApp
localhost:8888/notebooks/ADS_phase%204.ipynb#
YouTube
Maps
Gmail
SQL Commands
jupyter
ADS_phase 4 Last Checkpoint: 19 minutes ago (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
1.0
0.8
0.6
0.4
Desktop
16:32
01-11-2023

```
In [20]: #Strorytelling - Visualization
         corr = cov19.corr()
         plt.figure(figsize=(10,8))
         sns.heatmap(corr, cmap='viridis', annot=True)

Out[20]: <matplotlib.axes._subplots.AxesSubplot at 0xaa0fce8>
```

country	1	0.86	-0.069	0.0057	0.0076	-0.084	-0.045	-0.025	-0.034	-0.026	-0.00083	0.066
iso_code	0.86	1	-0.076	-0.004	-0.002	-0.085	-0.058	-0.091	-0.091	-0.086	-0.033	0.083
total_vaccinations	-0.069	-0.076	1	0.98	0.99	0.65	0.69	0.17	0.18	0.14	0.038	-0.18
people_vaccinated	0.0057	-0.004	0.98	1	0.96	0.76	0.83	0.12	0.16	0.1	0.029	-0.16
people_fully_vaccinated	0.0076	-0.002	0.99	0.96	1	0.65	0.71	0.16	0.19	0.16	0.013	-0.15
daily_vaccinations_raw	-0.084	-0.085	0.65	0.76	0.65	1	0.97	0.029	0.042	-0.028	0.13	-0.19
daily_vaccinations	-0.045	-0.058	0.69	0.83	0.71	0.97	1	0.042	0.063	-0.014	0.13	-0.18
total_vaccinations_per_hundred	-0.025	-0.091	0.17	0.12	0.16	0.029	0.042	1	0.97	0.98	0.18	-0.0054

```
In [21]: #Prepare Data for Machine Learning
         sns.regplot( y="daily_vaccinations",x="total_vaccinations",  data=cov19)

Out[21]: <matplotlib.axes._subplots.AxesSubplot at 0xb6e0718>
```

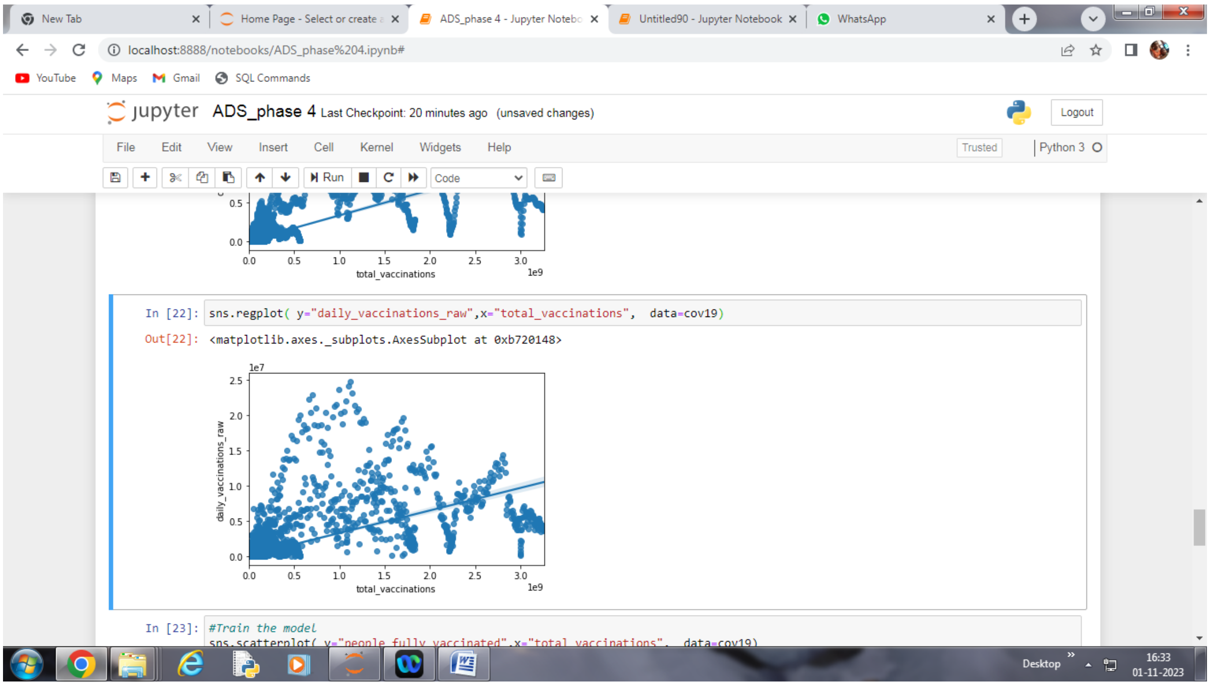
Prepare Data for Machine learning:



New Tab
Home Page - Select or create a
ADS_phase 4 - Jupyter Notebo
Untitled90 - Jupyter Notebook
WhatsApp
localhost:8888/notebooks/ADS_phase%204.ipynb#
YouTube
Maps
Gmail
SQL Commands
jupyter ADS_phase 4 Last Checkpoint: 20 minutes ago (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
0.2
0.0
Desktop
16:32
01-11-2023

total_vaccinations_per_hundred	-0.025	-0.091	0.17	0.12	0.16	0.029	0.042	1	0.97	0.98	0.18	-0.0054
people_vaccinated_per_hundred	-0.034	-0.091	0.18	0.16	0.19	0.042	0.063	0.97	1	0.96	0.28	0.0095
people_fully_vaccinated_per_hundred	-0.026	-0.086	0.14	0.1	0.16	-0.028	-0.014	0.98	0.96	1	0.11	0.0086
daily_vaccinations_per_million	0.00083	-0.033	0.038	0.029	0.013	0.13	0.13	0.18	0.28	0.11	1	-0.043

```
In [21]: #Prepare Data for Machine Learning
         sns.regplot( y="daily_vaccinations",x="total_vaccinations",  data=cov19)

Out[21]: <matplotlib.axes._subplots.AxesSubplot at 0xb6e0718>
```

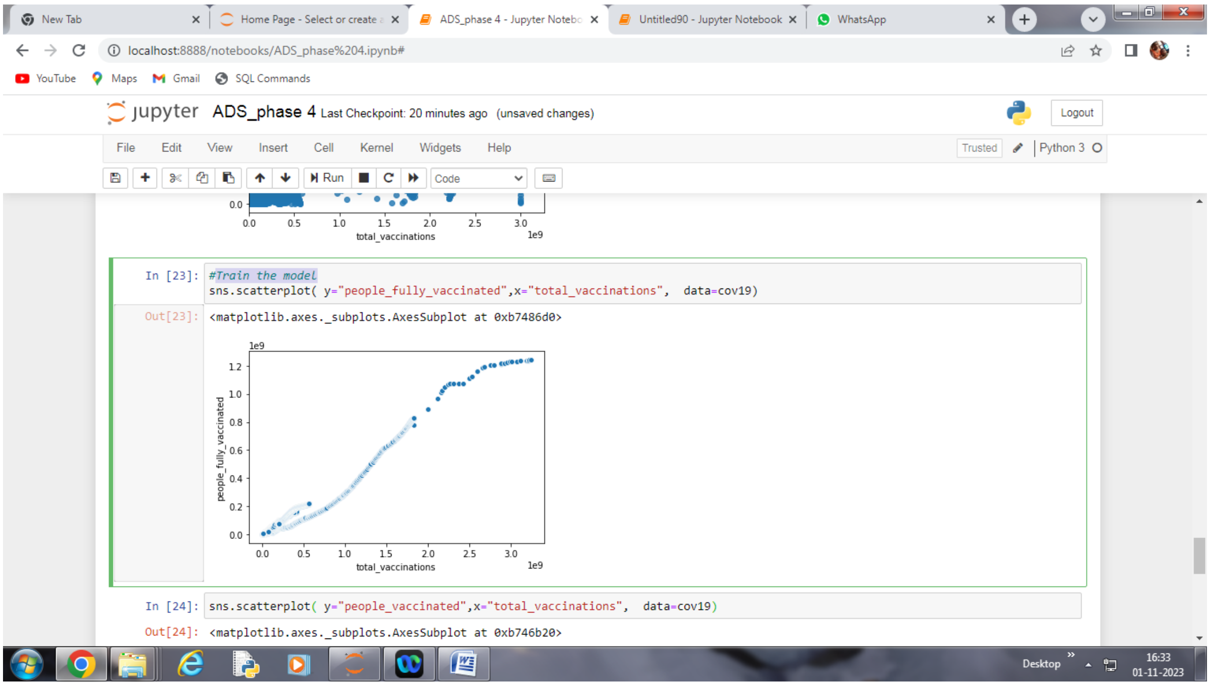


New Tab
Home Page - Select or create a
ADS_phase 4 - Jupyter Notebo
Untitled90 - Jupyter Notebook
WhatsApp
localhost:8888/notebooks/ADS_phase%204.ipynb#
YouTube
Maps
Gmail
SQL Commands
jupyter ADS_phase 4 Last Checkpoint: 20 minutes ago (unsaved changes)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3
Run
Code
0.5
0.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
1e9
total_vaccinations

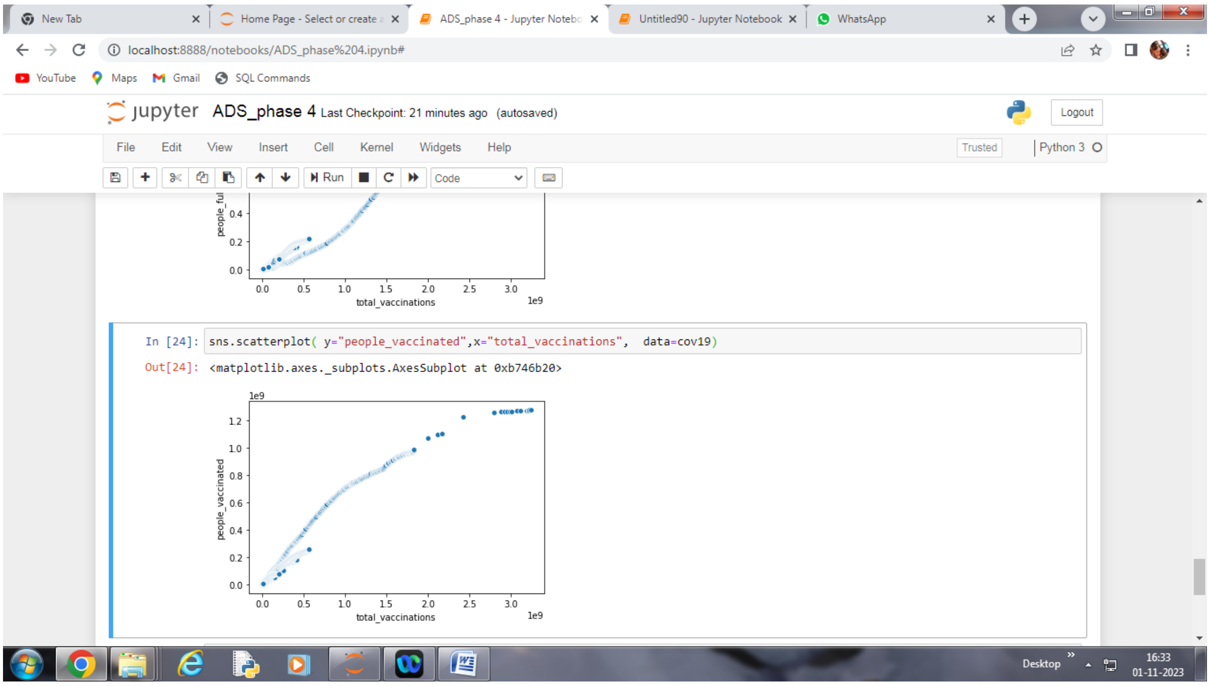
```
In [22]: sns.regplot( y="daily_vaccinations_raw",x="total_vaccinations",  data=cov19)

Out[22]: <matplotlib.axes._subplots.AxesSubplot at 0xb720148>
```

```
In [23]: #Train the model
         sns.scatterplot( y="people_fully_vaccinated",x="total_vaccinations",  data=cov19)
```

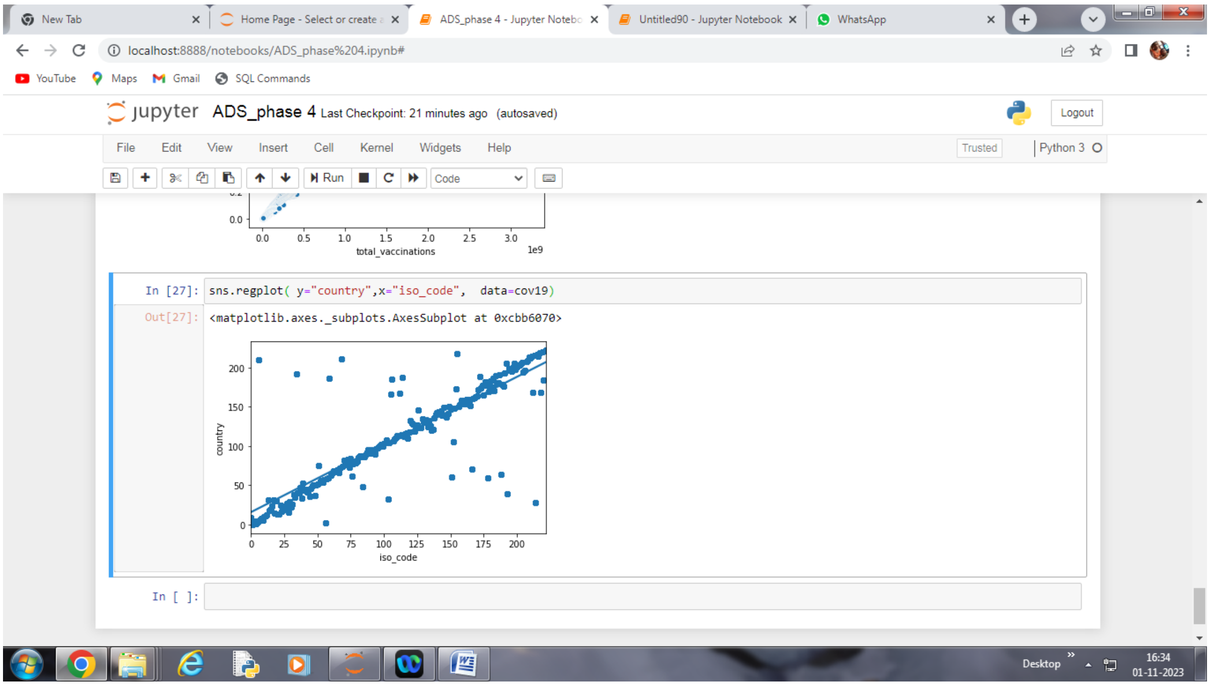
Train the model:



```
In [23]: #Train the model
         sns.scatterplot( y="people_fully_vaccinated",x="total_vaccinations",  data=cov19)

Out[23]: <matplotlib.axes._subplots.AxesSubplot at 0xb7486d0>

In [24]: sns.scatterplot( y="people_vaccinated",x="total_vaccinations",  data=cov19)

Out[24]: <matplotlib.axes._subplots.AxesSubplot at 0xb746b20>
```



```
In [24]: sns.scatterplot( y="people_vaccinated",x="total_vaccinations",  data=cov19)

Out[24]: <matplotlib.axes._subplots.AxesSubplot at 0xb746b20>
```



```
In [27]: sns.regplot( y="country",x="iso_code",  data=cov19)
```

```
Out[27]: <matplotlib.axes._subplots.AxesSubplot at 0xcbb6070>
```

```
In [ ]:
```