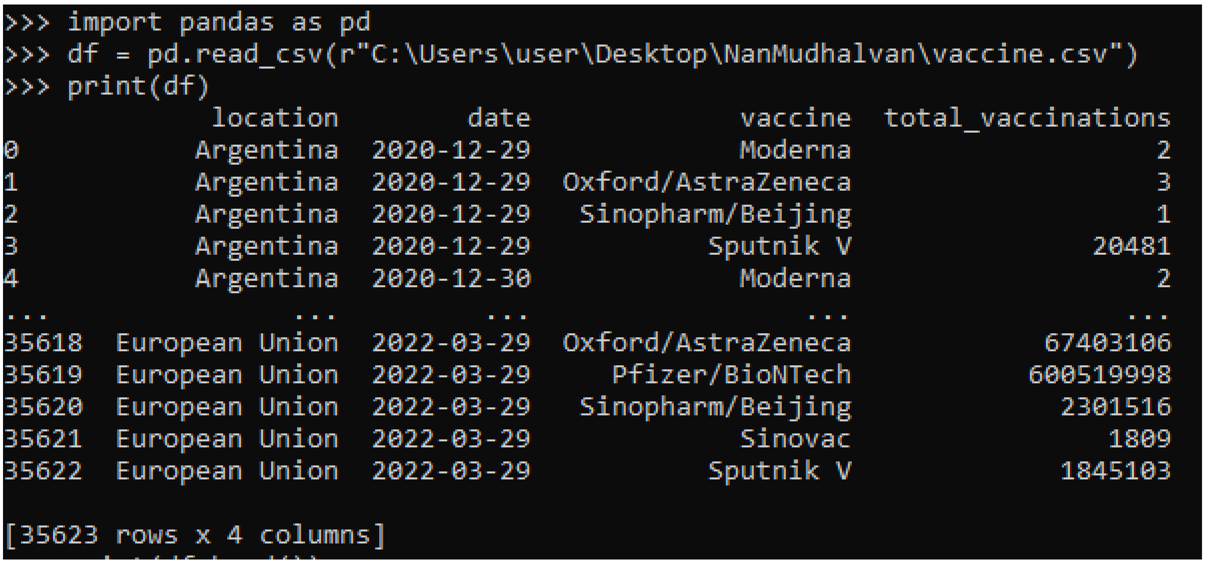
COVID-19 VACCINES ANALYSIS

1.INTRODUCTION:

This phase aims to clean, transform, and engineer features in a way that maximizes the model's ability to capture patterns and make accurate predictions. Through careful data preparation and feature engineering, we enhance the quality of input fed into the models. Effective preprocessing lays the foundation for improved predictive models.**The given dataset has been pre-processed and the outputs are attached with snap shots.**

2. IMPORTING LIBRARIES AND LOADING DATA:

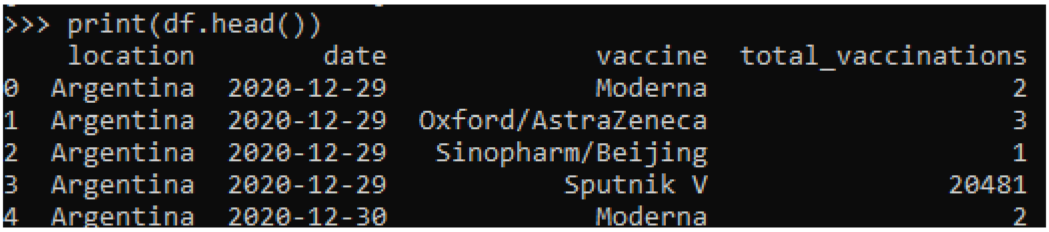
For Pre-Processing the given dataset, the pandas library is used. The given csv file is uploaded to pandas as follows:



```
>>> import pandas as pd
>>> df = pd.read_csv(r"C:\Users\user\Desktop\NanMudhalvan\vaccine.csv")
>>> print(df)
             location        date             vaccine  total_vaccinations
0           Argentina  2020-12-29             Moderna                   2
1           Argentina  2020-12-29  Oxford/AstraZeneca                   3
2           Argentina  2020-12-29   Sinopharm/Beijing                   1
3           Argentina  2020-12-29           Sputnik V               20481
4           Argentina  2020-12-30             Moderna                   2
...               ...         ...                 ...                 ...
35618  European Union  2022-03-29  Oxford/AstraZeneca            67403106
35619  European Union  2022-03-29     Pfizer/BioNTech           600519998
35620  European Union  2022-03-29   Sinopharm/Beijing             2301516
35621  European Union  2022-03-29             Sinovac                1809
35622  European Union  2022-03-29           Sputnik V             1845103

[35623 rows x 4 columns]
```

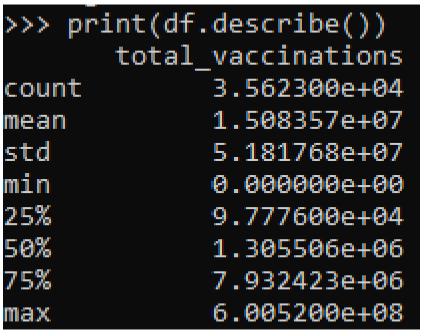
3. UNDERSTANDING THE DATASET:

df.head:



```
>>> print(df.head())
    location        date             vaccine  total_vaccinations
0  Argentina  2020-12-29             Moderna                   2
1  Argentina  2020-12-29  Oxford/AstraZeneca                   3
2  Argentina  2020-12-29   Sinopharm/Beijing                   1
3  Argentina  2020-12-29           Sputnik V               20481
4  Argentina  2020-12-30             Moderna                   2
```

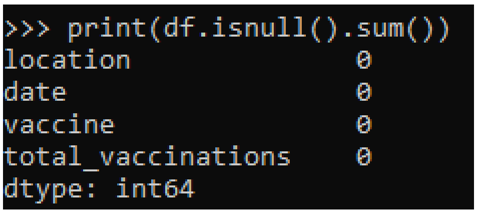
df.describe:



```
>>> print(df.describe())
       total_vaccinations
count        3.562300e+04
mean         1.508357e+07
std          5.181768e+07
min          0.000000e+00
25%          9.777600e+04
50%          1.305506e+06
75%          7.932423e+06
max          6.005200e+08
```

Isnull:

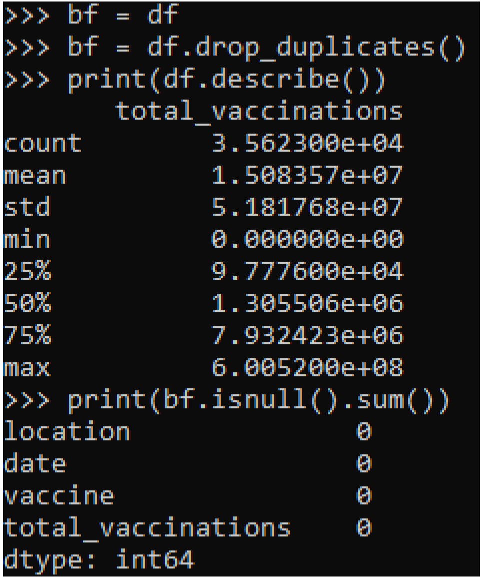
This function is used to identify missing values in the dataset. Since there in no null value, there is no need for handling the missing data.



```
>>> print(df.isnull().sum())
location              0
date                  0
vaccine               0
total_vaccinations    0
dtype: int64
```

4. REMOVING DUPLICATES:

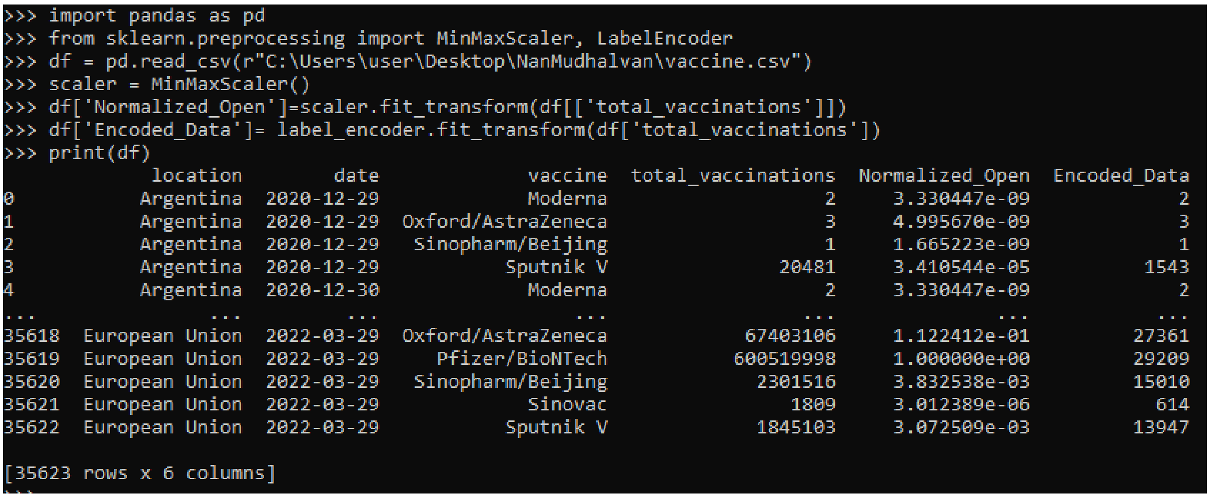
If any row is duplicated in the given dataset, the following code will identify it and remove it. The given dataset does not contain any duplicates and hence the dataset is the same as before.



```
>>> bf = df
>>> bf = df.drop_duplicates()
>>> print(df.describe())
       total_vaccinations
count        3.562300e+04
mean         1.508357e+07
std          5.181768e+07
min          0.000000e+00
25%          9.777600e+04
50%          1.305506e+06
75%          7.932423e+06
max          6.005200e+08
>>> print(bf.isnull().sum())
location              0
date                  0
vaccine               0
total_vaccinations    0
dtype: int64
```

5. DATA TRANSFROMATION:

Normalizing Data:



```
>>> import pandas as pd
>>> from sklearn.preprocessing import MinMaxScaler, LabelEncoder
>>> df = pd.read_csv(r"C:\Users\user\Desktop\NanMudhalvan\vaccine.csv")
>>> scaler = MinMaxScaler()
>>> df['Normalized_Open']=scaler.fit_transform(df[['total_vaccinations']])
>>> df['Encoded_Data']= label_encoder.fit_transform(df['total_vaccinations'])
>>> print(df)
             location        date              vaccine  total_vaccinations  Normalized_Open  Encoded_Data
0           Argentina  2020-12-29              Moderna                   2     3.330447e-09             2
1           Argentina  2020-12-29  Oxford/AstraZeneca                   3     4.995670e-09             3
2           Argentina  2020-12-29   Sinopharm/Beijing                   1     1.665223e-09             1
3           Argentina  2020-12-29           Sputnik V               20481     3.410544e-05          1543
4           Argentina  2020-12-30              Moderna                   2     3.330447e-09             2
...               ...         ...                  ...                 ...              ...           ...
35618  European Union  2022-03-29  Oxford/AstraZeneca            67403106     1.122412e-01         27361
35619  European Union  2022-03-29     Pfizer/BioNTech           600519998     1.000000e+00         29209
35620  European Union  2022-03-29   Sinopharm/Beijing             2301516     3.832538e-03         15010
35621  European Union  2022-03-29              Sinovac                1809     3.012389e-06           614
35622  European Union  2022-03-29           Sputnik V             1845103     3.072509e-03         13947

[35623 rows x 6 columns]
```

Z-Score Standardization (for column – high):

Z-score standardization, also known as "z-score normalization" or "z-score scaling," is a statistical method used to standardize or normalize features in a dataset. It's a process that transforms the features by scaling them to have a mean of 0 and a standard deviation of 1. This makes it easier to compare and analyze variables with different units or scales.

The formula to calculate the z-score for a given data point

X in a feature is: **z=X- μ/ σ**

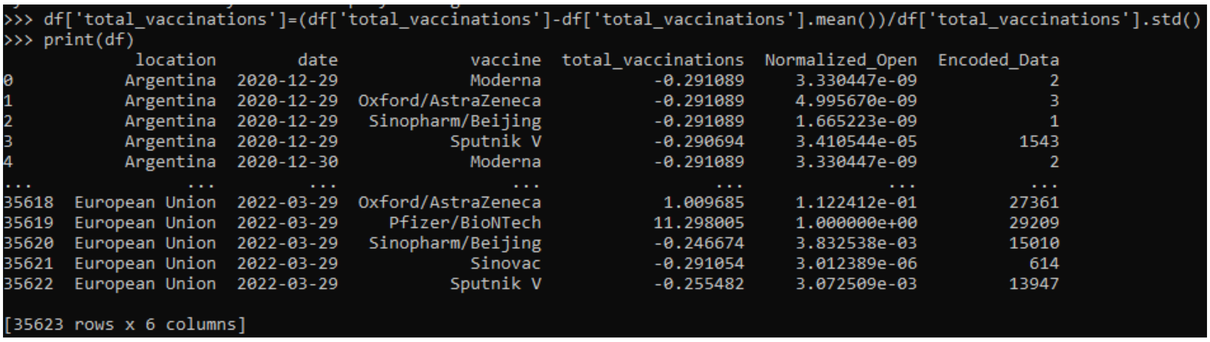
where:

X is an individual data point.

μ is the mean of the feature.

σ is the standard deviation of the feature.

The z-score measures how many standard deviations a data point is from the mean. A positive z-score indicates that the data point is above the mean, while a negative z-score indicates it's below the mean.



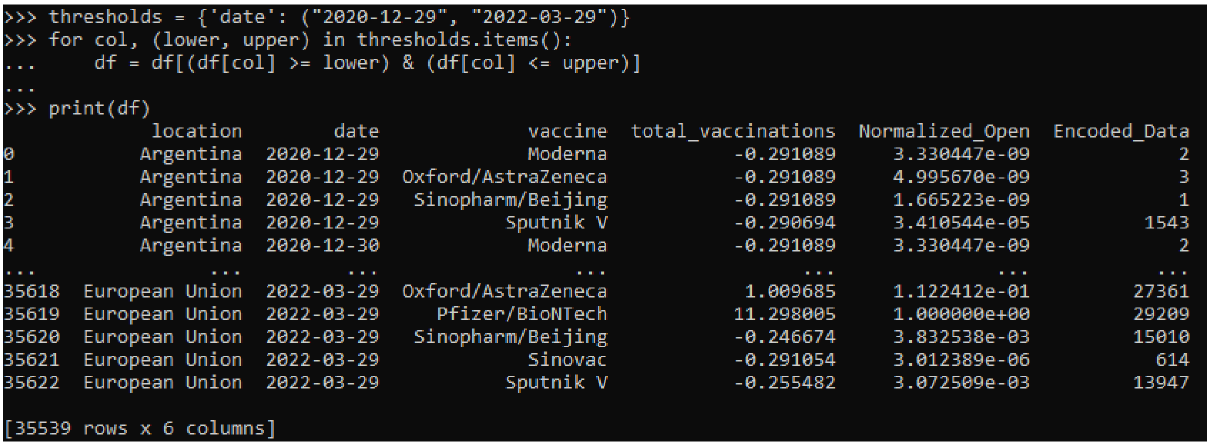
```
>>> df['total_vaccinations']=(df['total_vaccinations']-df['total_vaccinations'].mean())/df['total_vaccinations'].std()
>>> print(df)
             location        date              vaccine  total_vaccinations  Normalized_Open  Encoded_Data
0           Argentina  2020-12-29              Moderna           -0.291089     3.330447e-09             2
1           Argentina  2020-12-29  Oxford/AstraZeneca           -0.291089     4.995670e-09             3
2           Argentina  2020-12-29   Sinopharm/Beijing           -0.291089     1.665223e-09             1
3           Argentina  2020-12-29           Sputnik V           -0.290694     3.410544e-05          1543
4           Argentina  2020-12-30              Moderna           -0.291089     3.330447e-09             2
...               ...         ...                  ...                 ...              ...           ...
35618  European Union  2022-03-29  Oxford/AstraZeneca            1.009685     1.122412e-01         27361
35619  European Union  2022-03-29     Pfizer/BioNTech           11.298005     1.000000e+00         29209
35620  European Union  2022-03-29   Sinopharm/Beijing           -0.246674     3.832538e-03         15010
35621  European Union  2022-03-29              Sinovac           -0.291054     3.012389e-06           614
35622  European Union  2022-03-29           Sputnik V           -0.255482     3.072509e-03         13947

[35623 rows x 6 columns]
```

6. HANDLING OUTLIERS:

Outliers are data points that significantly differ from other observations in a dataset, deviating markedly from the overall pattern or distribution. They can be unusually high or low values that don't conform to the typical behaviour of the dataset.

The threshold fixed are the end points or outliers, all the values above and below are range are excluded and this process is called handling outliers.

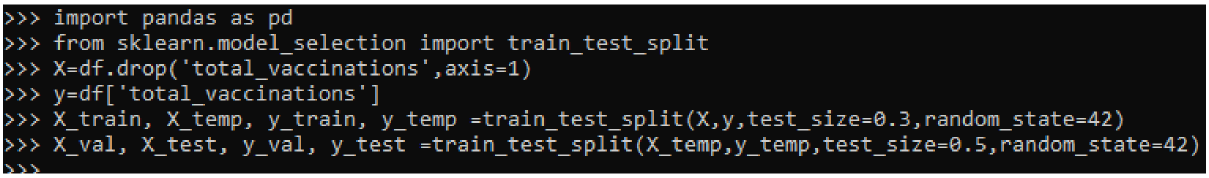
Date fixed as threshold:



```
>>> thresholds = {'date': ("2020-12-29", "2022-03-29")}
>>> for col, (lower, upper) in thresholds.items():
...     df = df[(df[col] >= lower) & (df[col] <= upper)]
...
>>> print(df)
             location        date            vaccine  total_vaccinations  Normalized_Open  Encoded_Data
0           Argentina  2020-12-29            Moderna           -0.291089     3.330447e-09             2
1           Argentina  2020-12-29  Oxford/AstraZeneca          -0.291089     4.995670e-09             3
2           Argentina  2020-12-29  Sinopharm/Beijing           -0.291089     1.665223e-09             1
3           Argentina  2020-12-29          Sputnik V           -0.290694     3.410544e-05          1543
4           Argentina  2020-12-30            Moderna           -0.291089     3.330447e-09             2
...               ...         ...                ...                 ...              ...           ...
35618  European Union  2022-03-29  Oxford/AstraZeneca           1.009685     1.122412e-01         27361
35619  European Union  2022-03-29     Pfizer/BioNTech          11.298005     1.000000e+00         29209
35620  European Union  2022-03-29  Sinopharm/Beijing           -0.246674     3.832538e-03         15010
35621  European Union  2022-03-29            Sinovac           -0.291054     3.012389e-06           614
35622  European Union  2022-03-29          Sputnik V           -0.255482     3.072509e-03         13947

[35539 rows x 6 columns]
```

7. DATA SPLITTING:

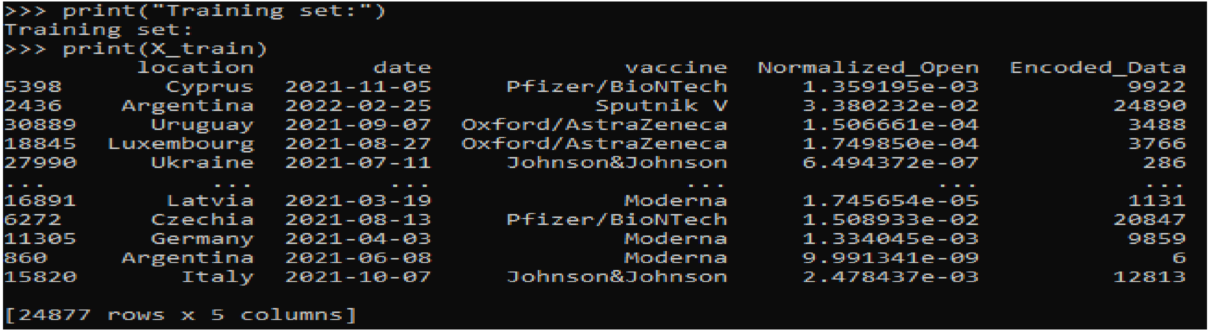
Outliers are data points that significantly differ from other observations in a dataset, deviating markedly from the overall pattern or distribution. They can be unusually high or low values that don't conform to the typical behavior of the dataset.



```
>>> import pandas as pd
>>> from sklearn.model_selection import train_test_split
>>> X=df.drop('total_vaccinations',axis=1)
>>> y=df['total_vaccinations']
>>> X_train, X_temp, y_train, y_temp =train_test_split(X,y,test_size=0.3,random_state=42)
>>> X_val, X_test, y_val, y_test =train_test_split(X_temp,y_temp,test_size=0.5,random_state=42)
>>>
```

TRAINING SET:

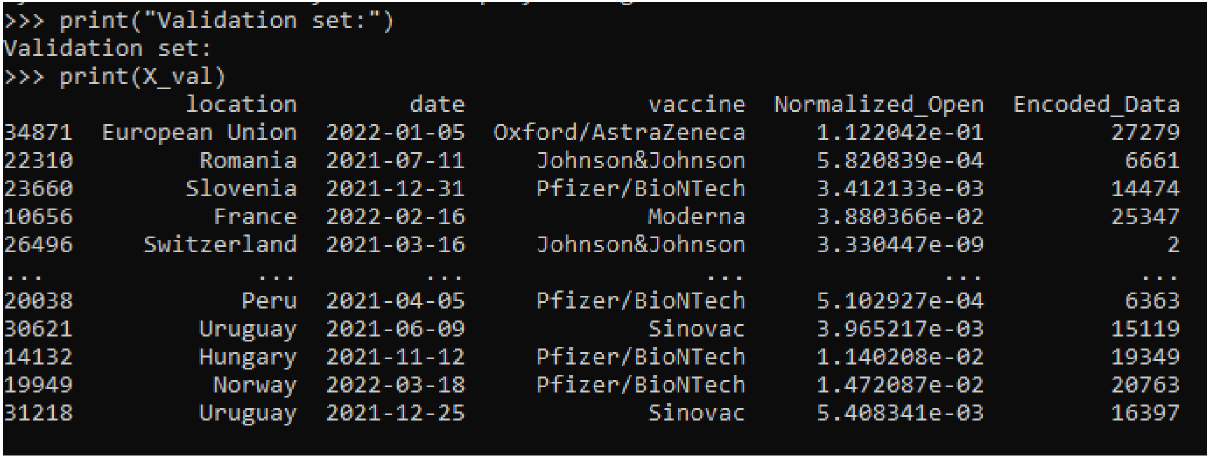
Purpose: Used to train the model, allowing it to learn patterns and relationships in the data.
Size: Largest portion of the dataset (e.g., 70-80%).
Importance: Fundamental for model training, ensuring the model learns from a variety of examples



```
>>> print("Training set:")
Training set:
>>> print(X_train)
         location        date             vaccine  Normalized_Open  Encoded_Data
5398       Cyprus  2021-11-05     Pfizer/BioNTech     1.359195e-03          9922
2436    Argentina  2022-02-25           Sputnik V     3.380232e-02         24890
30889     Uruguay  2021-09-07  Oxford/AstraZeneca     1.506661e-04          3488
18845  Luxembourg  2021-08-27  Oxford/AstraZeneca     1.749850e-04          3766
27990     Ukraine  2021-07-11     Johnson&Johnson     6.494372e-07           286
...           ...         ...                 ...              ...           ...
16891      Latvia  2021-03-19             Moderna     1.745654e-05          1131
6272      Czechia  2021-08-13     Pfizer/BioNTech     1.508933e-02         20847
11305     Germany  2021-04-03             Moderna     1.334045e-03          9859
860     Argentina  2021-06-08             Moderna     9.991341e-09             6
15820       Italy  2021-10-07     Johnson&Johnson     2.478437e-03         12813

[24877 rows x 5 columns]
```

VALIDATION SET:

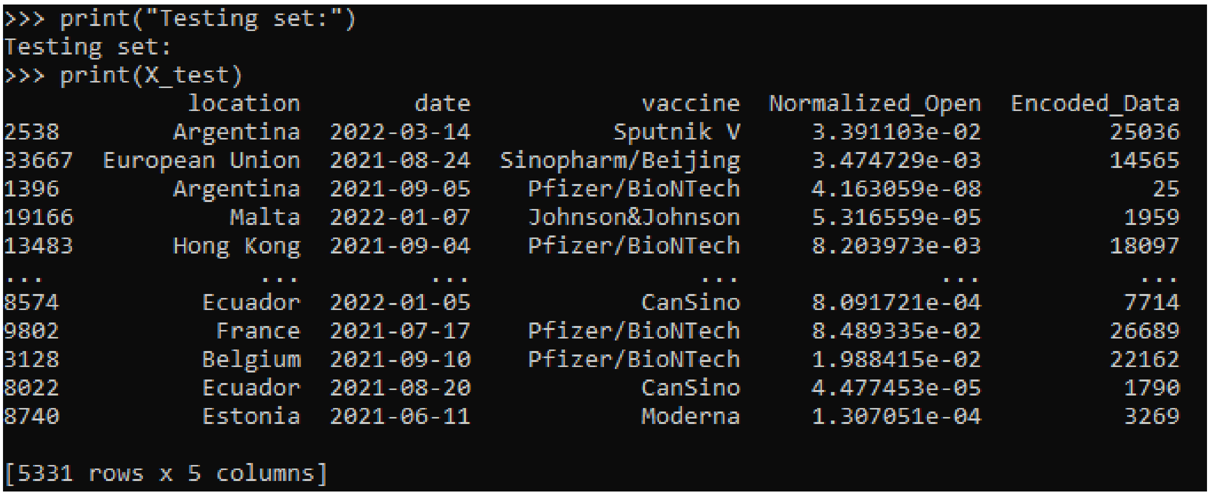
Purpose: Used to fine-tune the model's hyperparameters, aiding in model selection and preventing overfitting.
Size: Smaller portion of the dataset (e.g., 10-15%).
Importance: Helps optimize the model's performance and generalization.



```
>>> print("Validation set:")
Validation set:
>>> print(X_val)
             location        date              vaccine  Normalized_Open  Encoded_Data
34871  European Union  2022-01-05  Oxford/AstraZeneca     1.122042e-01         27279
22310         Romania  2021-07-11     Johnson&Johnson     5.820839e-04          6661
23660        Slovenia  2021-12-31     Pfizer/BioNTech     3.412133e-03         14474
10656          France  2022-02-16             Moderna     3.880366e-02         25347
26496     Switzerland  2021-03-16     Johnson&Johnson     3.330447e-09             2
...               ...         ...                 ...              ...           ...
20038            Peru  2021-04-05     Pfizer/BioNTech     5.102927e-04          6363
30621         Uruguay  2021-06-09             Sinovac     3.965217e-03         15119
14132         Hungary  2021-11-12     Pfizer/BioNTech     1.140208e-02         19349
19949          Norway  2022-03-18     Pfizer/BioNTech     1.472087e-02         20763
31218         Uruguay  2021-12-25             Sinovac     5.408341e-03         16397
```

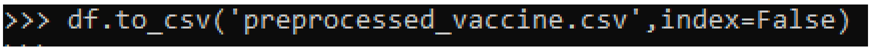
TESTING SET:

Purpose: Used to evaluate the model's performance on unseen data after training and validation.
Size: Smaller portion of the dataset (e.g., 10-15%).
Importance: Provides an unbiased evaluation of the model's performance and generalization to new data.



```
>>> print("Testing set:")
Testing set:
>>> print(X_test)
             location        date             vaccine  Normalized_Open  Encoded_Data
2538        Argentina  2022-03-14          Sputnik V     3.391103e-02         25036
33667  European Union  2021-08-24  Sinopharm/Beijing     3.474729e-03         14565
1396        Argentina  2021-09-05    Pfizer/BioNTech     4.163059e-08            25
19166           Malta  2022-01-07    Johnson&Johnson     5.316559e-05          1959
13483       Hong Kong  2021-09-04    Pfizer/BioNTech     8.203973e-03         18097
...               ...         ...                ...              ...           ...
8574          Ecuador  2022-01-05            CanSino     8.091721e-04          7714
9802           France  2021-07-17    Pfizer/BioNTech     8.489335e-02         26689
3128          Belgium  2021-09-10    Pfizer/BioNTech     1.988415e-02         22162
8022          Ecuador  2021-08-20            CanSino     4.477453e-05          1790
8740          Estonia  2021-06-11            Moderna     1.307051e-04          3269

[5331 rows x 5 columns]
```

8. SAVING:



```
>>> df.to_csv('preprocessed_vaccine.csv',index=False)
```

9. CONCLUSION:

In the third phase, the dataset has been preprocessed, which is fundamental to building accurate and reliable predictive models. This involved handling missing values, scaling, encoding categorical features, and possibly applying other transformations like feature engineering or selection. The preprocessed dataset is now ready for the subsequent phases, where it will be utilized to train and validate models