COVID-19 VACCINE ANALYSIS

TEAM MEMBERS

G.DHAYANITHI

K.KALAIYARASAN

T.PARTHASARATHY

K.KARAN

B.MANIYARASU

PHASE 3 SUBMISSION DOCUMENT

PHASE 3: DEVELOPMENT PART 1



INTRODUCTION

* Data Science is focused on interdisciplinary domains and useful for taking decisions.Effective Vaccines are needed to save lots of lives throughout the worldwide epidemics such as COVID19. The community looks to COVID-19 vaccination progression mustbe considered sensibly in directive to know the users sentiments and fears to it. To knowmore about exact information about covid-19 vaccines are from who are taken the vaccination and they are express their opinions. In this research article studied and understands the advantages of social media. Now a Social media has become an important tool for gaining insights about any domain. At the time of Covid-19 pandemic social media applications are playing a key role in users thoughts on various topics sharing. About Vaccination side effects and results confusion is one of the serious issues in realizing herd immunity and suppressing the COVID-19 epidemic. To consider this approach our focus on analyze user opinions on COVID-19 vaccination process. The world face a main corona virus epidemic from the year 2019. The virus infects fast through various ways. All nations lock peoples to avoid the virus. Vaccinations, including Covaxin, Covishield,Pfizer, Moderna, SputnikV have been permitted. This research article, tweet analysisis based on people’s opinions about official covid-19 vaccines on social media Twitter. Datasets collected, Covaxin, Covishield, Pfizer, Moderna, SputnikV. These tweets are preprocessed using Machine learning techniques.
* In this research article studied the users opinion on Pfizer, Modern, AstraZeneca and Johnson & Johnson. The total posts in each nation for time period of month of Jan 2021 to Apr 2021, May 2021 to Aug 2021 and Sept 2021 to Dec 2021 was plot. The use of opinion Analysis impacts on each domain like product analysis, Recommendation system, prediction on healthcare and analytics.
* After declaration of vaccination and governments announces the policy about vaccination. More peoples hesitates about its impact and side effects. In the month of Nov 9, 2020, when the vaccination drive starts and many people are reacting on social media about their effectiveness .

DATASET LINK : https://www.kaggle.com/datasets/gpreda/covid-world-vaccination-progress

NECESSARY STEPS TO FOLLOW:

Start by importing libraries:

Pandas:

Pandas is essential for data manipulation and analysis, particularly for loading and handling datasets.

Program:

import pandas as pd

NumPy:

NumPy is used for numerical computations, and it complements Pandas for handling arrays and mathematical operations.

Program:

import NumPy as np

Scikit-Learn (sklearn):

Scikit-Learn provides tools for machine learning, including dataset splitting, preprocessing, and model evaluation. You'll import specific modules as needed for your analysis.

Program:

from sklearn. model selection import train\_test\_split

from sklearn. Preprocessing import StandardScaler # For data scaling (if needed)

LOAD THE DATASET:

To load a dataset for credit card fraud detection, We can use the Pandas library in Python. Here's how we can load a dataset from a CSV file, which is a common data format:

Program:

import pandas as pd

# Specify the file path to your dataset

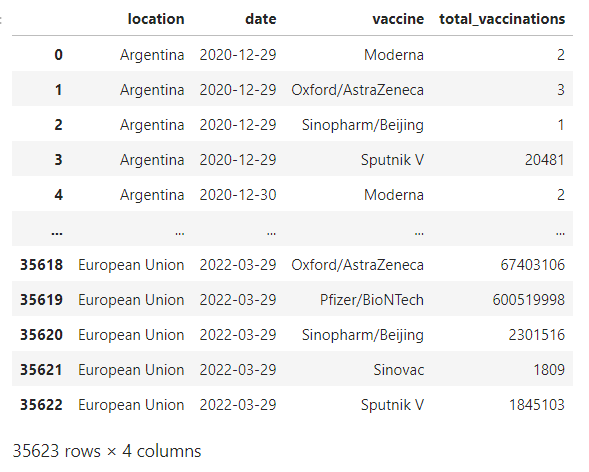
File\_path=("D:/KV STUDY/PHASE

3/country\_vaccinations\_by\_manufacturer.csv")

# Use Pandas to read the CSV file into a DataFrame

df = pd.read\_csv("D:/KV STUDY/PHASE

3/country\_vaccinations\_by\_manufacturer.csv")



# Now, 'df' contains your dataset, and you can start working with it.

In the code above:

Import the Pandas library to work with data.

Replace 'your\_dataset.csv' with the actual file path to our dataset. Make sure that the CSV file is in the same directory as our Python script, or provide the full path to the file if it's located elsewhere.

The pd.read\_csv(file\_path) function reads the CSV file and stores its contents in a Pandas DataFrame called df. This DataFrame is a two-dimensional table-like data structure that you can manipulate and analyze.

After loading the dataset into a Data Frame, we can perform various data analysis tasks, such as data exploration, preprocessing, and modelling, depending on your specific objectives in credit card fraud detection.

Data processing :

Data processing occurs when data is collected and translated into usable information. Usually performed by a data scientist or team of data scientists, it is important for data processing to be done correctly as not to negatively affect the end product, or data output.

Basic Summary Statistics:

Use Pandas to obtain summary statistics of the dataset, which can give a quick overview of the data, including counts, means, standard deviations, and percentiles.

Program:

print(df.describe())

Data Shape:

Use to see how many Rows and columns in our Dataset.

df**.**shape

(35623, 4)

Dependent and Independent Variables

So, y is referred to as dependent feature or variable of total\_vaccinations and x is referred to as independent features or variables of location,date,vaccine,total\_vaccinations. Any predictive mathematical model tends to divide the observations (data) into dependent/ independent features in order to determine the causal effect.

Program:

x**=**df[['location','date','vaccine','total\_vaccinations']]**.**values

# x for independent variables

X

Out:

array([['Argentina', '2020-12-29', 'Moderna', 2],

['Argentina', '2020-12-29', 'Oxford/AstraZeneca', 3],

['Argentina', '2020-12-29', 'Sinopharm/Beijing', 1],

...,

['European Union', '2022-03-29', 'Sinopharm/Beijing', 2301516],

['European Union', '2022-03-29', 'Sinovac', 1809],

['European Union', '2022-03-29', 'Sputnik V', 1845103]],

dtype=object)

# y for dependent variables

y**=**df[['total\_vaccinations']]**.**values

y

Out:

array([[ 2],

[ 3],

[ 1],

...,

[2301516],

[ 1809],

[1845103]], dtype=int64)

Sklearn.impute

The SimpleImputer class provides basic strategies for imputing missing values. Missing values can be imputed with a provided constant value, or using the statistics (mean, median or most frequent) of each column in which the missing values are located. This class also allows for different missing values encodings.

Program:

**import** sklearn.impute **as** sl

In

**from** sklearn.impute **import** SimpleImputer

In

imputer**=**SimpleImputer(missing\_values**=**np**.**nan,strategy**=**'mean')

In

imputer**=**imputer**.**fit(y[:,0:1])

In

y

output:

array([[ 2],

[ 3],

[ 1],

...,

[2301516],

[ 1809],

[1845103]], dtype=int64)

Label encoder

Label encoding is a technique used in machine learning and data analysis to convert categorical variables into numerical format. It is particularly useful when working with algorithms that require numerical input, as most machine learning models can only operate on numerical data.

Program:

**from** sklearn.preprocessing **import** LabelEncoder

In :

label\_encode\_x**=**LabelEncoder()

In :

y[:,0]**=**label\_encode\_x**.**fit\_transform(y[:,0])

In :

Y

Output:

array([[ 2],

[ 3],

[ 1],

...,

[15010],

[ 614],

[13947]], dtype=int64)

To use machine learning models, analyse feature importance to understand which attributes play a significant role in covid-19 vaccine analysis

One-hot encoder

One-Hot Encoding is another popular technique for treating categorical variables. It simply creates additional features based on the number of unique values in the categorical feature. Every unique value in the category will be added as a feature. One-Hot Encoding is the process of creating dummy variables.

Program:

**from** sklearn.preprocessing **import** OneHotEncoder

In :

onehotencoder**=**OneHotEncoder()

In :

onehotencoder**.**fit\_transform(df**.**location**.**values**.**reshape(**-**1,1))**.**toarray()

output:

array([[1., 0., 0., ..., 0., 0., 0.],

[1., 0., 0., ..., 0., 0., 0.],

[1., 0., 0., ..., 0., 0., 0.],

...,

[0., 0., 0., ..., 0., 0., 0.],

[0., 0., 0., ..., 0., 0., 0.],

[0., 0., 0., ..., 0., 0., 0.]])

LabelEncoder Fit\_transform

LabelEncoder , we can use the fit\_transform function. This function fits the LabelEncoder object to the input data, and then transforms the data into encoded values. By default, fit\_transform assigns a unique numerical value to each category in the input data.

Program:

labelencoder\_y**=**LabelEncoder()

In :

y**=**labelencoder\_y**.**fit\_transform(y)

y

output:

array([ 2, 3, 1, ..., 15010, 614, 13947], dtype=int64)

sklearn model\_selection in train\_test\_split

train\_test\_split is a function in Sklearn model selection for splitting data arrays into two subsets: for training data and for testing data. With this function, you don't need to divide the dataset manually. By default, Sklearn train\_test\_split will make random partitions for the two subsets.

Program:

**from** sklearn.model\_selection **import** train\_test\_split

x\_train,x\_test,y\_train,y\_test**=**train\_test\_split(x,y,test\_size**=**0.2,train\_size**=**0.4,random\_state**=**0)

Input:

x\_train

output:

array([['Germany', '2021-11-19', 'Oxford/AstraZeneca', 12731024],

['European Union', '2021-12-09', 'Sinopharm/Beijing', 2202502],

['South Africa', '2022-03-09', 'Pfizer/BioNTech', 24339805],

...,

['Spain', '2021-08-13', 'Johnson&Johnson', 1916324],

['Ukraine', '2022-02-13', 'Moderna', 3007446],

['Croatia', '2021-07-16', 'Moderna', 299490]], dtype=object)

Input:

x\_test

output:

array([['Argentina', '2021-12-03', 'CanSino', 148761],

['Ireland', '2021-02-26', 'Oxford/AstraZeneca', 72049],

['European Union', '2021-02-18', 'Oxford/AstraZeneca', 535136],

...,

['Austria', '2021-02-26', 'Pfizer/BioNTech', 587225],

['Uruguay', '2021-08-16', 'Pfizer/BioNTech', 1714002],

['Hong Kong', '2021-09-22', 'Sinovac', 3066825]], dtype=object)

Input:

y\_train

output

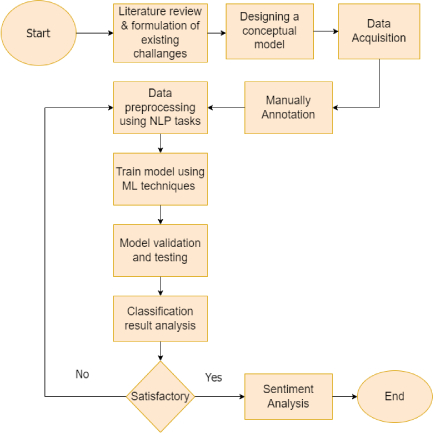
array([22716, 14789, 25546, ..., 14085, 15923, 6310], dtype=int64)

input:

y\_test

output:

array([ 4380, 3108, 8122, ..., 8752, 13680, 16018], dtype=int64)



CONCLUSION

* In this paper, we collected data from twitter and then apply preprocessing for data exploration, classification. Data Preprocessing involves the Removing URLs, DataFiltering, Removing Special Characters, Removal of Retweets, Usernames, Remove Punctuations and symbols, Usage of Web links, Hashtags, Tokenization, Exclamation and question marks, Letter Repetition, Negations.
* Machine learning classifier used and studied the comparative analysis between KNN, Support Vector Machines, Naïve Bayes, Decision Tree algorithms for data classification.
* Finding shows that Decision Tree classifier for Covishield dataset has achieved the highest 97% accuracy with compared to Naïve Bayes, Support Vector Machine, KNN classification methods. Support Vector Machine has lowest Accuracy with 94% for SputnikV.
* COVID-19 Vaccination dataset wise machine learning model evaluation performance studied and got highest and lowest results of Machine learning classifiers. The Support Vector Machine SputnikV dataset got highest accuracy with 94% and Covishield dataset got lowest accuracy with 89%, The Naïve Bayes got highest accuracy for Covishield dataset with 95% and lowest accuracy with 87% for Moderna dataset, The Decision tree got highest accuracy for Covishield dataset with 97% and lowest accuracy with 88% for Pfizer dataset, The KNN got highest accuracy for Covaxin dataset with 96% and lowest accuracy with 88% for SputnikV dataset.
* In Lexicon Based approached Sentiment polarity classification here total 23500 tweets taken for result analysis and predict the vaccination opinions on SputnikV, Covishield, Covishield, Covaxin, Pfizer datasets. Overall here identify the Neutral opinions on Vaccinations. In other side when we focused on positive and negative opinions here Covaxin is more positive compare with all other vaccination datasets according twitter discussion of users insights and negative opinions on Pfizer vaccination datasets.