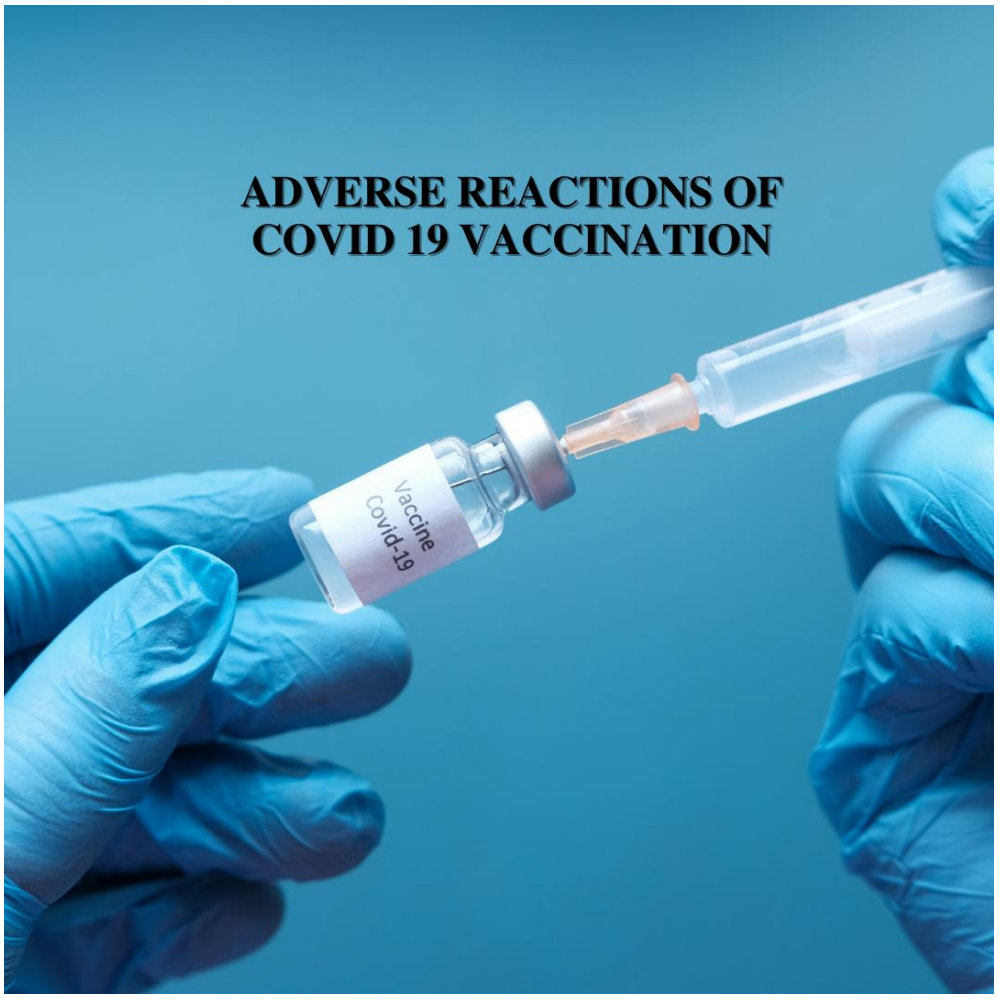


TOPIC: A study on Covid- 19 Vaccines using Natural language processing and Deep learning analysis.

Submitted as Master Dissertation in SIT723

Jasdeep Kaur

STUDENT ID: 220181376
COURSE: Master's in Information & Technology Professional (S779)

Supervised by: Dr. Sasan Adibi

Abstract

Covid vaccination has contributed to cut COVID-related infections severely, prevent deaths, and stem the spread. However, due to potential allergenic compounds present in the vaccine, people have been dealing with adverse reactions after their vaccination doses. This research paper analyses the correlation that exists between covid vaccine and most occurring symptoms post vaccination. To investigate this question, this study examines textual data of the symptoms, word frequency, and a word cloud to identify the most frequently occurring symptoms. Utilizing such data, the aim is to classify and characterize patients that are at risk of such adverse reactions in the hope of reducing patient risk by identifying possible causes of adverse reactions to certain drugs as well as curb the graph of vaccine hesitancy. Additionally, a sentiment analysis has been carried out based on the Twitter Feeds (tokens of Covid and Corona) to predict the sentiments (Positive, Neutral and Negative) by creating word clouds based on the polarity, this in turn uncovered remarkable findings about people's reaction on social media. This is an example of NLP (Natural language Processing) which can also be used for future predicting the polarity of sentiments.

This study also explores the vaccination data to build a prediction model for identifying the mortality rate of patients across three manufacturers namely Pfizer, Moderna and Jannsen. The data has been statistically analysed and machine learning (ML) techniques as well as deep leaning techniques were employed to predict the severity of side effects.

To validate this approach, US dataset has been taken from VAERS (Vaccine Adverse Event Reporting System) which is a passive reporting system, that requires individuals to send in reports of their experiences. VAERS is not intended to determine whether a vaccine caused a health problem but is particularly useful for detecting unusual patterns of side effects. The dataset suggested that vaccines were available from three different manufacturers, and our goal was to determine which of these vaccines was the most effective in terms of lowering mortality rates. Moderna had the most vaccinated patients, followed by Pfizer and Janssen, and in terms of mortality rates, Pfizer had the lowest with 0.52%, followed by Moderna with 0.62%, and then Janssen with 0.65%. According to preliminary analysis, if we talk about effective vaccines, it has to be Moderna and Pfizer because they had a higher number of patients vaccinated and lower mortality rates. According to the EDA, patients spend an average of 5 to 6 days in a hospital, with an average age of 42 to 48 years, while senior citizens aged 62 and up have the highest mortality rates. Using the dataset, a machine learning model has been created to predict patient fatality rates across the three manufacturers. When compared against four machine learning algorithms, LSTM outperformed the others in terms of performance at 95.2% accuracy. While working with an unbiased dataset in which one class is heavier than the other and the data isn't simple enough, variable importance had low scores for the independent variables, and LSTM is proven to be effective in such cases (Bouktif et al. 2018). Even our model performance statistics confirm this.

NLP/Text Analysis was also performed using pre-trained model of VADER (Valence Aware Dictionary for Sentiment Reasoning) with a similar goal of determining which vaccines were effective based on the symptoms data available in the VAERS dataset. Twitter data was also obtained to gain a better understanding of how people are reacting to vaccine manufacturers. Sentiment Analysis carried out on the textual data determined which manufacturers had the lowest Negative Polarity based on Symptoms and which manufacturers had more positive tweets than negative tweets based on Twitter data. Pfizer was outscored by Moderna in both cases.

Index Terms— Covid-19 vaccine adverse reactions, LSTM, machine learning, deep learning, side effects, post vaccination symptoms, sentiment analysis, NLP.

Contents

Introduction

The year 2020 will go down in modern history as one of the most challenging years as far as battling SARS CoV-2, a viral infection that causes acute respiratory illnesses. The virus rapidly spread throughout the world, triggering a worldwide epidemic that lasted until now (Ahamad et al. 2021). The global pandemic has raised concerns about the healthcare system's ability to handle the influx of people due to a serious pandemic (Dong et al. 2020). It has infected over 33 million people and killed about a million worldwide including 200,000 deaths alone in the Unites States (Li and Lu 2020.

The World Health Organization (WHO) declared the release of covid vaccines for emergency use in September 2020 (Kaur and Gupta 2020). Since then, almost 44.77 million people got vaccinated with one or two doses in the US, 653 deaths and 12,697 adverse events had been reported as of 10 February 2021(Li and Lu 2020).

Vaccine hesitancy (VH) refers to a delay in vaccine adoption amongst people. Researchers see it as a public health concern fueled by adverse reactions of vaccine as well as misinformation travelling of the social media in terms of safety and efficacy (Harrison and Wu 2020). Other factors contributing to low vaccine compliance include potential side effects and a lack of faith in vaccine manufacturers (Almufty et al. 2021).

Machine learning has been extensively used in the field of predicting efficacy of a vaccine. The ability to predict how different people will react to vaccination and to understand what best protects people from infection greatly impacts development of future vaccinations (Lee et al. 2016). According to past literature, the LSTM classifier achieves comparable results for detecting vaccination behaviour and that recurrent algorithms outperforms tree-based algorithms(Imran et al. 2020). Natural Language Processing (NLP) on the other hand has been promising in the field of opinion mining via sentiment dictionary to evaluate people's attitude, sentiments and perceptions through computational analysis(Na et al. 2021).

To investigate the perception of US citizens regarding the vaccine this study has followed the above-mentioned approaches for performing analysis. It will discuss past literature under background and will be followed by Sentiment classification based on twitter feeds to build word cloud at an overall level and across the three manufacturers to calculate the NLP summary based on sentiment score (Polarity) distribution. It will then discuss adverse effects or symptoms based on the three different vaccines to identify out of three manufacturers having higher negative symptoms. Lastly, a deep learning will be discussed to predict the mortality of patients post vaccination.

Background

There is limited study on potential covid vaccination side effects, linked risk factors and comparison of the three COVID-19 vaccines. A literature study on conventional epidemic (cold, Severe acute respiratory syndrome (SAS), COVID-19) and mRNA vaccines, as shown in Table 1, relates to advantageous nature of these vaccines on viral outbreaks. However, they

are not focussed on complications associated with these vaccines which can be life threatening. This study aims to build a prediction model as a more accurate decision support tool by using deep learning to predict lower mortality rate of covid vaccines based on different vaccine manufacturers.

Authors	Journal	Epidemics	Main content
(Tobaiqy et al. 2021)	Vaccines	Covid -19	Analysis of adverse Reactions of COVID-19 AstraZeneca Vaccine.
(Kaur et al. 2021)	Indian Journal of Clinical Biochemistry 2021	Covid-19	Adverse events reported from COVID-19 vaccine Trials.
(Meyer et al. 2017)	The Journal of Infectious Diseases 2017	Ebola Virus	Modified mRNA-Based Vaccines on Ebola Virus Disease.
(Bahl et al. 2017)	Molecular Therapy	H10N8 and H7N9 Influenza Viruses	Immunogenicity by mRNA Vaccines against H10N8 and H7N9 Influenza Viruses.

Table 1: Literature reviews of epidemics and mRNA vaccines.

A literature study related to impact of deep learning algorithms on vaccine efficacy, such as Table 2, shows that LSTM (Long Short-Term Memory) classification models are the focus.

Authors	Journal	Predicted target	Deep Learning Algorithms
(Tiftikci et al. 2019)	BMC Bioinformatics 2019	adverse drug reactions in drug labels	LSTM
(Challita et al. 2019)	IEEE Wireless Communications 2019 Vol. 26 Issue 1 Pages 28-35		LSTM
Imran et al. 2020	2020 IEEE International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE)	mRNA vaccine degradation	LSTM

Table 2: Literature reviews of Deep learning algorithms

LSTM technique has been incorporated for modelling and compared it with three other algorithms namely Logistic, Decision Tree and Random Forest. The unique part of the analysis is the use of text data (Symptoms) that would have otherwise overlooked in the original model. Textual data has been used and Sentiment analysis was performed to identify the manufacturer with lower negatives of symptoms.

Literature Review

Started in December 2019, pathogenic outbreak severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) has continued to spread around the world now causing 4.46 million deaths across the globe. Occurred in Wuhan City, Hubei Province, China, viral Covid-19 has caused the worst pandemic in the age of globalization leaving society to cope with unprecedented threats and challenges. Aided by vaccination, countries have been trying to quench the virus impact by acquiring immunity through vaccines. While novel covid-19 vaccines have been contributing to the declining graph of mortality, various adverse reactions have been encountered causing potentials risks ultimately leading to vaccine hesitancy (Blumenthal et al.

2021). Wedlund and Kvedar (2021) discussed uninfected people will take the most benefit from the vaccine without getting severe symptoms, on the other hand Ahamad et al. (2021) tried to relate acute reactions from vaccine with patients prior illnesses. Tissot et al. (2021) studied patients with Covid 19 history having at least one adverse reaction to the vaccine. Consequently, a great deal of research has been focussed on understanding allergic reactions on different classes of people obtained from covid vaccines.

Due to limited amount data of vaccinated people, this area has been explored until recently and most of the literature has focussed on possible vaccine treatments for covid mitigation. This study will discuss the relationship between vaccine and reported events by looking at substantial variation of adverse reactions along with sentiment analysis of vaccines on different classes of people in United States. The aim is to understand the extent to which the data analysis enables us to understand which vaccine will be most effective along with predicting the important features or attributes suitable for deciding the efficacy of vaccines on different people. Factors important for increasing the survival of a patient will also be studied across different vaccines and a deep learning model will be devised to mortality rate against three different vaccines. This study will be followed by Deep learning and machine learning algorithms to stimulate association between adverse reactions and efficacy of covid vaccines.

Hatmal et al. (2021) suggests machine learning can be an effective tool to predict the severity of side effects of covid vaccines by analysing the adverse reactions. This research has been carried out in Jordan however, to further appreciate the study, we must investigate, in detail, different methodologies which can be a crucial in predicting efficacy of vaccines.

Whilst researchers have been trying to perform a comprehensive review of these unparalleled initiatives of combating with the deadly virus. However, there are still many uncertainties looming around the efficiency of the vaccines and their side effects. Therefore, predicting life threatening symptoms after vaccination could be beneficial in reducing patient risk and reliability towards vaccine treatments.

While the above studies discuss statistical strategies through a set of classification machine learning algorithms to discuss the negative outcomes of the vaccine. The limitation lies with the precarious nature of the virus and effectual protection through covid vaccines. Hence, further investigation needs to be exercised on data from other countries to explore the possibilities of reducing the risk of poor outcomes.

Most research involves machine learning to understand and predict the implicit impact of vaccine on people around the world Ong et al. (2020), Brooks et al. (2021). Deep learning has been found effective in prediction as well as mitigation of covid vaccine threats (Chen et al. 2021).
After searching some electronic databases like (Google scholar, PubMed, Science direct), descriptive studies were reported on prediction accuracy of covid- 19 analysis by applying Long short-term memory deep neural network (Shahid et al. 2020). It may therefore be advantageous to incorporate LSTM techniques for feature selection to better understand the attributes responsible in deciding which vaccine will be most effective.

Research Design and Methodology

Dataset Part 1: NLP (Text Analysis-Sentiment Analysis)

Text analysis has been carried out using Twitter Feeds. VADER (Valence Aware Dictionary for Sentiment Reasoning) is a text sentiment analysis model that considers both the polarity (positive/negative) and the intensity (strong) of emotion. It's included in the NLTK package and may be used on unlabelled text data right away. The sentimental analysis of VADER is based on a lexicon that maps lexical characteristics to emotion intensities, which are known as sentiment scores. A text's sentiment score may be calculated by adding the intensity of each word in the text. A Sentiment Classification Model based on the Twitter Feeds obtained in September 2021 through tokens of Covid and Corona was carried out on a pre trained model of VADER for predicting the sentiments (Positive, Neutral and Negative) of the tweets based on the sentiment of words. This will give us an idea what impact the Covid had on our lives and how people are reacting to it on social media. I have also created word clouds based on the polarity. This is an example of NLP which can also be used for future predicting the polarity of sentences.

```
1 from nltk.sentiment.vader import SentimentIntensityAnalyzer
2 sia = SentimentIntensityAnalyzer()
```

```
df.head(10)
```

	Id	Timestamp	Source	Retwwet_Count	User_Name	Tweets
0	1441386072357756942	2021-09-24 12:56:33	Twitter for Android	0	Willow A	@CaptainAdvance1 @wewatchu2 Kombucha vaxx coul...
1	1441386055341527041	2021-09-24 12:56:29	Twitter for Android	0	WinterBae❄	@Jyojiriiin Maybe the side effects were kinda ...
2	1441386032859996169	2021-09-24 12:56:24	Twitter Web App	0	Better Masks 4 Melbourne	@_marching_Ents_ The 60+ will have a choice. A...
3	1441386002279387144	2021-09-24 12:56:17	Twitter Web App	0	YinZerAPE	@johnrich How many shares of Moderna, Pfizer d...
4	1441385995732090892	2021-09-24 12:56:15	Twitter Web App	0	Barb Mazzocca	@MikeCruise18 @mikekon71897391 @TomDNaughton W...
5	1441385910474625024	2021-09-24 12:55:55	Echobox	1	Newsweek	The FDA authorized Pfizer booster shots this w...
6	1441385439068246017	2021-09-24 12:54:02	Twitter for Android	0	Joseph	@sedsand @SenatorWong You are way off on Vacci...
7	1441385312463335425	2021-09-24 12:53:32	Twitter for iPad	0	john henry	But no Moderna second jabs... https://t.co/yjy...
8	1441385297829261324	2021-09-24 12:53:29	Twitter for Android	0	Sánchez Acero	@thehill The ones who want you to get their va...
9	1441385272462118924	2021-09-24 12:53:23	Twitter for Android	0	CMDoranASP	@IDstewardship @complexfive @ABXsteward I'm su...

Twitter Data – September 2021 (Token → Moderna) | (< 2,000 Tweets)

Dataset Part 2: NLP (Text Analysis-Word cloud)

Another textual analysis has been carried out using the symptoms data from VAERS Dataset. Word Frequency and word cloud are being created to understand which of the symptoms are common based on the vaccines from the three manufacturers. Based on the VAERS dataset, a classification model was created to predict the survival of a patient based on different information of the vaccine administered to the patient. This will also highlight which of the factors are important for increasing the survival of a patient, moreover this can be of utmost importance for a manufacturer for developing a new vaccine. It also helps in identifying which of the manufacturers (out of three manufacturers) have a higher Survival rate. This model can be used later for predicting the survival probability of a patient as well factors influencing the survival probabilities across brands.

```
data_symp_vax_1.head(10)
```

	VAERS_ID	VAX_MANU	VAX_DOSE_SERIES
0	916600	MODERNA	1
1	916601	MODERNA	1
2	916602	PFIZER\BIONTECH	1
3	916603	MODERNA	UNK
4	916604	MODERNA	1
5	916606	MODERNA	1
6	916607	MODERNA	UNK
7	916608	MODERNA	1
8	916609	MODERNA	1
9	916610	MODERNA	1

```
data_2_symptoms.head(10)
```

	VAERS_ID	SYMPTOM1	SYMPTOMVERSION1	SYMPTOM2	SYMPTOMVERSION2	SYMPTOM3	SYMPTOMVERSION3	SYMPTOM4	SYMPTOMVERSION4	SYMPTO
0	916600	Dysphagia	23.1	Epiglottitis	23.1	NaN	NaN	NaN	NaN	
1	916601	Anxiety	23.1	Dyspnoea	23.1	NaN	NaN	NaN	NaN	
2	916602	Chest discomfort	23.1	Dysphagia	23.1	Pain in extremity	23.1	Vsual impairment	23.1	
3	916603	Dizziness	23.1	Fatigue	23.1	Mobility decreased	23.1	NaN	NaN	
4	916604	Injection site erythema	23.1	Injection site pruritus	23.1	Injection site swelling	23.1	Injection site warmth	23.1	
5	916605	Chills	24.0	Confusional state	24.0	Eye inflammation	24.0	Headache	24.0	Labora
6	916605	Pyrexia	24.0	White blood cell count decreased	24.0	NaN	NaN	NaN	NaN	
7	916606	Pharyngeal swelling	23.1	NaN	NaN	NaN	NaN	NaN	NaN	

```
data_3_vax.head(10)
```

	VAERS_ID	VAX_TYPE	VAX_MANU	VAX_LOT	VAX_DOSE_SERIES	VAX_ROUTE	VAX_SITE	VAX_NAME
0	916600	COVID19	MODERNA	037K20A	1	IM	LA	COVID19 (COVID19 (MODERNA))
1	916601	COVID19	MODERNA	025L20A	1	IM	RA	COVID19 (COVID19 (MODERNA))
2	916602	COVID19	PFIZER\BIONTECH	EL1284	1	IM	LA	COVID19 (COVID19 (PFIZER-BIONTECH))
3	916603	COVID19	MODERNA	unknown	UNK	NaN	NaN	COVID19 (COVID19 (MODERNA))
4	916604	COVID19	MODERNA	NaN	1	IM	LA	COVID19 (COVID19 (MODERNA))
5	916605	FLUC4	SEQIRUS, INC.	276563	1	SYR	LA	INFLUENZA (SEASONAL) (FLUCELVAX QUADRIVALENT)
6	916606	COVID19	MODERNA	011J20A	1	IM	LA	COVID19 (COVID19 (MODERNA))

VAERS Data for NLP Analysis

Data Cleaning

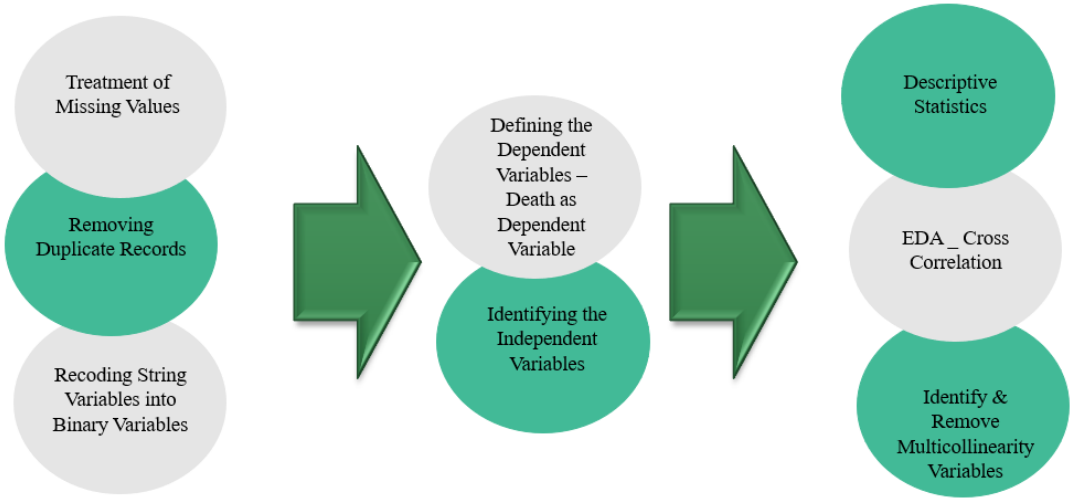
Data was fetched from VAERS website which needed prior data cleaning to make it ready for the analysis. I considered the data which will add value to the analysis and makes it easier to conclude. Hence, 2021 data along with the three manufacturer's data was taken as the objective and target group will be in and around them. I further removed those columns which were unwanted in my analysis for instance, columns which have heavy texts or documents. To add more depth in data cleaning, I also removed data which I couldn't use for modelling like symptoms data. Missing values along with variables with huge number of levels like date variables were imputed.

Pre-processing/EDA

Furthermore, data correction steps along with transformation was performed. For example, replacing the missing values with zeros or average values based on variable types and removing duplicate data if at all present. Next, transforming the characters into dichotomous variables as I could only work on numeric variables for the modelling. Later steps involved identification

of the dependent or target variable which will be the flag to depict patients' survival. This will be considered as a dichotomous variable as I performed a binary classification modelling. The rest of the variables will be independent variable/features used for predicting the survival probability. Descriptive statistics was then carried out for understanding the data patterns and to perform conjunctive analysis on the data to create hypothesis as to what kind of data was involved, distribution of dataset and relationship across the variables.

The objective of the EDA is to study the join effect of the variables and identify the variables which will be important for the model or will have impact on the dependent variable. I also made sure that multicollinearity variables were not present otherwise it would have ended up making the model biased. Multicollinearity happens when independent variables are highly correlated with each other (Blalock 1963). It can be very expensive for a model and should be removed by dropping any one of the multicollinear variables.

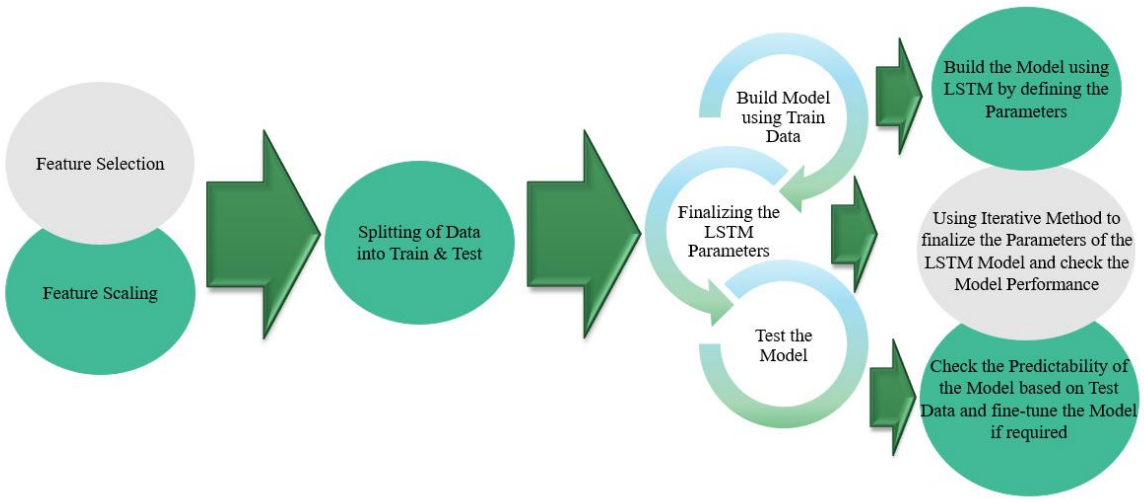


Classifier construction

Constructing classifiers required impactful variables or the variables having an impact on the dependent variable to perform standardization so that variables can be of the same scale to get a better model for analysis.

Using Machine Learning approach, binary classification model was built. I trained the model using the training data and then tested the performance of the model using the testing data. I split the entire data as such that 70% had train data and the rest 30% contained test data.

My binary classification model is all about survival probability, hence it is ought to contain class imbalance i.e., one of the classes will have majority of the data points and are not equally distributed. In such cases, we need to balance the dataset using weighted method or use the SMOTE method for building the model using a balanced dataset. Using these techniques, it generates data and balances the class which had less data points.



Training

I worked on binary classification model using the LSTM algorithm which is one of the deep Learning models where we need to define different parameters like epochs, layers, nodes, dropout values, etc. for getting a better model using dataset. Using an iterative method, we will build our model using the 70% training dataset and check for the statistics like confusion matrix (True Positive, False Positive, Precision, Recall, Sensitivity, Specificity, ROC Curve, concordance.

```
: from sklearn.model_selection import train_test_split

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.30, random_state=7)
print(X_train.shape)
print(Y_train.shape)
print(X_test.shape)
print(Y_test.shape)

X_train = X_train.replace([np.inf, -np.inf], 0)
X_test = X_test.replace([np.inf, -np.inf], 0)
```

```
(240443, 84)
(240443,)
(103048, 84)
(103048,)
```

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import KFold
from sklearn.metrics import roc_auc_score
num_trees = [25,50,75,100,125,150,175,200,225,250]
max_depth = [3,4,5,6]
num_cv_splits = 5
```

```
kf = KFold(n_splits=num_cv_splits, random_state=5)
for tree in num_trees:
    for depth in max_depth:
        auc = 0.0
        acc = 0.0
        for train_index, test_index in kf.split(X_train):
            X_train_cv, X_test_cv = X_train.iloc[train_index], X_train.iloc[test_index]
            Y_train_cv, Y_test_cv = Y_train.iloc[train_index], Y_train.iloc[test_index]
            clf = RandomForestClassifier(n_estimators=tree, max_depth=depth, n_jobs = 8, random_state=5)
            clf.fit(X_train_cv,Y_train_cv)
            acc += clf.score(X_test_cv,Y_test_cv)
            pred = clf.predict_proba(X_test_cv)[:,1]
            auc += roc_auc_score(y_true = Y_test_cv, y_score = pred)
        print('num_trees =',tree,'; depth=',depth,'; mean accuracy =',acc/num_cv_splits,'; auc =',auc/num_cv_splits)
```

```
num_trees = 25 ; depth= 3 ; mean accuracy = 0.9885128685053786 ; auc = 0.9003092333118206
num_trees = 25 ; depth= 4 ; mean accuracy = 0.9885128685053786 ; auc = 0.9129976555031017
num_trees = 25 ; depth= 5 ; mean accuracy = 0.9885128685053786 ; auc = 0.916388842114842
num_trees = 25 ; depth= 6 ; mean accuracy = 0.9885170274606491 ; auc = 0.925553235910941
num_trees = 50 ; depth= 3 ; mean accuracy = 0.9885128685053786 ; auc = 0.9078953513786079
num_trees = 50 ; depth= 4 ; mean accuracy = 0.9885128685053786 ; auc = 0.9202721369070523
num_trees = 50 ; depth= 5 ; mean accuracy = 0.9885128685053786 ; auc = 0.9219877873539495
num_trees = 50 ; depth= 6 ; mean accuracy = 0.9885128685053786 ; auc = 0.9280767967707433
num_trees = 75 ; depth= 3 ; mean accuracy = 0.9885128685053786 ; auc = 0.9079991607256641
num_trees = 75 ; depth= 4 ; mean accuracy = 0.9885128685053786 ; auc = 0.9207668806967
num_trees = 75 ; depth= 5 ; mean accuracy = 0.9885128685053786 ; auc = 0.9245166676216078
num_trees = 75 ; depth= 6 ; mean accuracy = 0.9885128685053786 ; auc = 0.9282739187143786
num_trees = 100 ; depth= 3 ; mean accuracy = 0.9885128685053786 ; auc = 0.908636906549918
num_trees = 100 ; depth= 4 ; mean accuracy = 0.9885128685053786 ; auc = 0.9209350523189027
```

Analysis of the model

Lastly, test data was used for predicting the results and calculating the accuracy percentage to finalize the model. The process was rerun to obtain optimal results by changing the parameters of the LSTM model. While working on the feature selection process, it gives us an opportunity to identify the variables which are important or have an impact on the dependent variable. Manufacturers can leverage this information while creating new vaccinations in the future. It will also help us to identify which of these manufacturers have a higher survival rate. The feature selection process has been performed using information value and by using the random forest algorithm.

```
from keras.models import Sequential, Model
from keras.layers import Dense, GRU, Dropout, Flatten, TimeDistributed
from keras.callbacks import ModelCheckpoint, ReduceLROnPlateau
from keras import optimizers
from keras.applications.vgg16 import VGG16

base_model = VGG16(include_top=False, weights='imagenet', input_shape=(120,120,3))
x = base_model.output
x = Flatten()(x)
#x.add(Dropout(0.5))
features = Dense(64, activation='relu')(x)
conv_model = Model(inputs=base_model.input, outputs=features)

for layer in base_model.layers:
    layer.trainable = False

model = Sequential()
model.add(TimeDistributed(conv_model, input_shape=(15,120,120,3)))
model.add(GRU(32, return_sequences=True))
model.add(GRU(16))
model.add(Dropout(0.5))
model.add(Dense(8, activation='relu'))
model.add(Dense(5, activation='softmax'))
```

```
model.fit_generator(train_generator, steps_per_epoch=steps_per_epoch, epochs=num_epochs, verbose=1,
                    callbacks=callbacks_list, validation_data=val_generator,
                    validation_steps=validation_steps, class_weight=None, workers=1, initial_epoch=0)
```

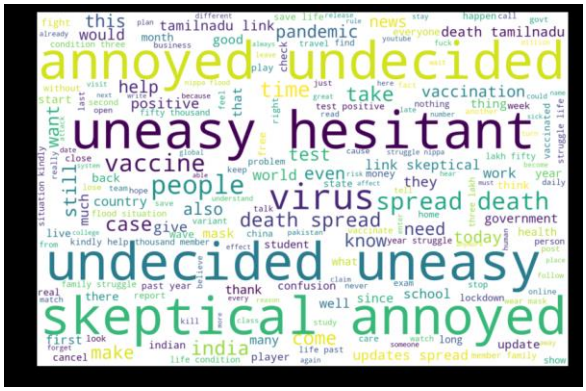
```
Epoch 25/70
42/42 [==============================] - 43s 1s/step - loss: 0.8494 - categorical_accuracy: 0.7529 - val_loss: 1.1134 - val_c
ategorical_accuracy: 0.6100

Epoch 00025: saving model to model_init_conv_lstm_2018-10-0414_07_55.144483/model-00025-0.85085-0.74962-1.11337-0.61000.h5
Epoch 26/70
42/42 [==============================] - 43s 1s/step - loss: 0.8199 - categorical_accuracy: 0.7581 - val_loss: 1.0979 - val_c
ategorical_accuracy: 0.6100

Epoch 00026: saving model to model_init_conv_lstm_2018-10-0414_07_55.144483/model-00026-0.81817-0.75867-1.09791-0.61000.h5

Epoch 00026: ReduceLROnPlateau reducing learning rate to 0.0002500000118743628.
Epoch 27/70
42/42 [==============================] - 43s 1s/step - loss: 0.8008 - categorical_accuracy: 0.7838 - val_loss: 1.0921 - val_c
ategorical_accuracy: 0.6100
```

Environment

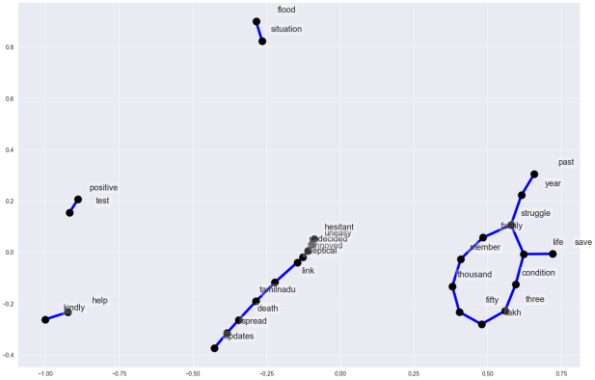
The analysis has taken place in python language, which is compatible with platforms like anaconda, google colab.

Experiments and Results

Sentiment Analysis



Word Cloud of All Twitter Feeds – Overall



Network Graph of All Twitter Feeds – Overall

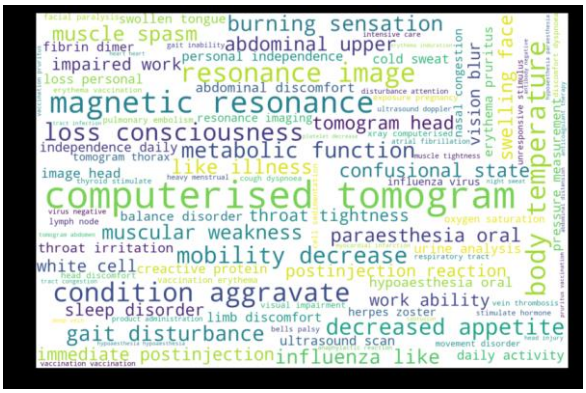
Vaccine Manufacturers	Neutral	Positive	Negative
Janssen	5	2	3
Moderna	158	234	477
Pfizer	152	240	820

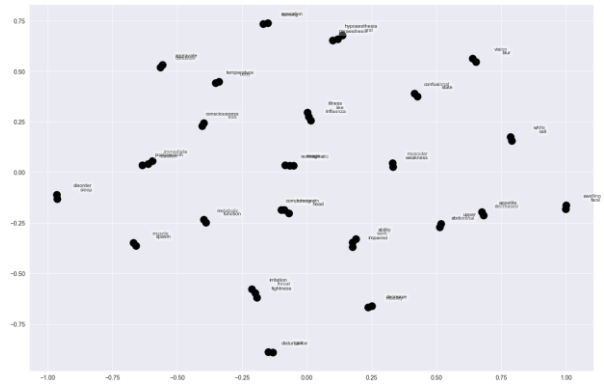
Vaccine Manufacturers	Neutral	Positive	Negative
Janssen	50%	20%	30%
Moderna	18%	27%	55%
Pfizer	13%	20%	68%

Janssen has very few Tweets, so it can be ignored. Moderna has a higher Positive Tweets compared to Pfizer

NLP Summary – Based on Sentiment Score (Polarity) Distribution

Textual Analysis





Textual Analysis – Word Cloud – Overall

Textual Analysis – Network Graph - Overall

Vaccine Manufacturers	Negative - Low	Negative - Medium	Negative - High
Janseen	12,766	6,433	3,508
Moderna	54,052	23,527	14,068
Pfizer	55,042	25,186	16,745

Vaccine Manufacturers	Negative - Low	Negative - Medium	Negative - High
Janseen	56%	28%	15%
Moderna	59%	26%	15%
Pfizer	57%	26%	17%

Moderna has highest Negative Low based on the Symptoms; we can say that Moderna Vaccines have higher effectiveness.

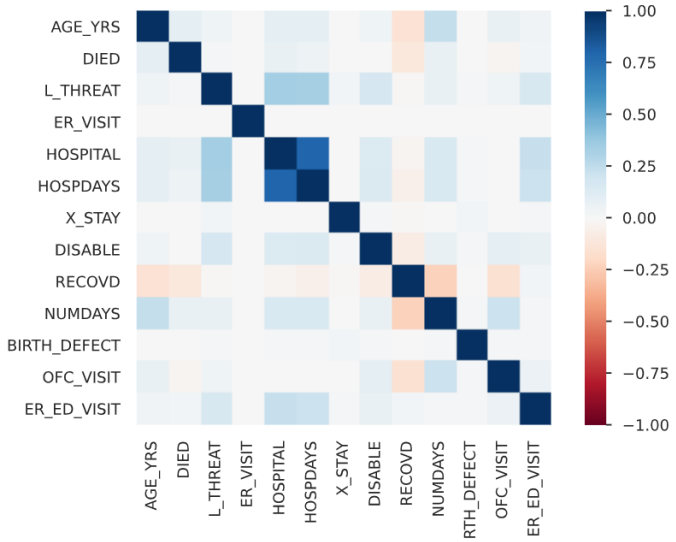
NLP Summary – Based on Sentiment Score (Polarity) Distribution (VAERS Data)

Analysis - Descriptive Statistics

VAX_LOT	JANSSEN	MODERNA	PFIZER\BIONTECH
1	39,576	168,797	137,976
2	236	97,240	112,057
3	35	540	702
4	15	57	59
5	47	81	49
6	5	29	42
7+	16	195	22
UNK	25,230	37,687	36,532

Distribution of Vaccine Lot Across the 3 Manufacturers

	AGE_YRS	DIED	L_THREAT	ER_VISIT	HOSPITAL	HOSPDAYS	X_STAY	DISABLE	RECOVD	NUMDAYS	BIRTH_DEFECT	OFC_VISIT	ER_ED_VISIT
AGE_YRS	1.000000	0.093645	0.042972	0.000515	0.095057	0.037666	0.005499	0.045163	-0.179982	0.005557	-0.002264	0.089068	0.048113
DIED	0.093645	1.000000	0.014986	-0.000964	0.070813	0.030223	0.005462	-0.004123	-0.102772	0.000521	0.001793	-0.024990	0.026470
L_THREAT	0.042972	0.014986	1.000000	0.005976	0.339791	0.133306	0.028282	0.164125	-0.012943	0.003802	0.021902	0.044223	0.158096
ER_VISIT	0.000515	-0.000964	0.005976	1.000000	0.001734	-0.000495	-0.000193	-0.000849	0.003373	-0.000258	-0.000187	-0.003994	-0.002958
HOSPITAL	0.095057	0.070813	0.339791	0.001734	1.000000	0.284962	0.007618	0.140103	-0.025847	0.006659	0.014208	-0.006102	0.228490
HOSPDAYS	0.037666	0.030223	0.133306	-0.000495	0.284962	1.000000	0.002255	0.089731	-0.018559	0.001406	0.003477	0.000369	0.070243
X_STAY	0.005499	0.005462	0.028282	-0.000193	0.007618	0.002255	1.000000	0.012595	-0.008230	-0.000105	0.025899	0.001586	0.014168
DISABLE	0.045163	-0.004123	0.164125	-0.000849	0.140103	0.089731	0.012595	1.000000	-0.068357	0.000475	0.022206	0.095446	0.073852
RECOVD	-0.179982	-0.102772	-0.012943	0.003373	-0.025847	-0.018559	-0.008230	-0.068357	1.000000	-0.005870	-0.000722	-0.144051	0.028815
NUMDAYS	0.005557	0.000521	0.003802	-0.000258	0.006659	0.001406	-0.000105	0.000475	-0.005870	1.000000	0.000040	0.005663	0.003251
BIRTH_DEFECT	-0.002264	0.001793	0.021902	-0.000187	0.014208	0.003477	0.025899	0.022206	-0.000722	0.000040	1.000000	0.020145	0.009395
OFC_VISIT	0.089068	-0.024990	0.044223	-0.003994	-0.006102	0.000369	0.001586	0.095446	-0.144051	0.005663	0.020145	1.000000	0.060948
ER_ED_VISIT	0.048113	0.026470	0.158096	-0.002958	0.228490	0.070243	0.014168	0.073852	0.028815	0.003251	0.009395	0.060948	1.000000




```
#Final Data, joining with Symp and Vax File

data_comb = pd.merge(data_2, data_symp_vax_1,how = 'left', on=['VAERS_ID'], )
```

```
data_comb.head(10)
```

'EDIED	...	V_FUNDBY	PRIOR_VAX	SPLTTYPE	FORM_VERS	TODAYS_DATE	BIRTH_DEFECT	OFC_VISIT	ER_ED_VISIT	VAX_MANU	VAX_DOSE_SERIES
NaN	...	NaN	NaN	NaN	2	01/01/2021	NaN	Y	NaN	MODERNA	1
NaN	...	NaN	NaN	NaN	2	01/01/2021	NaN	Y	NaN	MODERNA	1
NaN	...	NaN	NaN	NaN	2	01/01/2021	NaN	NaN	Y	PFIZER\BIONTECH	1
NaN	...	NaN	got measles from measel shot, mums from mumps ...	NaN	2	01/01/2021	NaN	NaN	NaN	MODERNA	UNK
NaN	...	NaN	NaN	NaN	2	01/01/2021	NaN	NaN	NaN	MODERNA	1
NaN	...	NaN	NaN	NaN	2	01/01/2021	NaN	Y	NaN	0	NaN
NaN	...	NaN	NaN	NaN	2	01/01/2021	NaN	NaN	NaN	MODERNA	1
NaN	...	NaN	NaN	NaN	2	01/01/2021	NaN	NaN	NaN	MODERNA	UNK

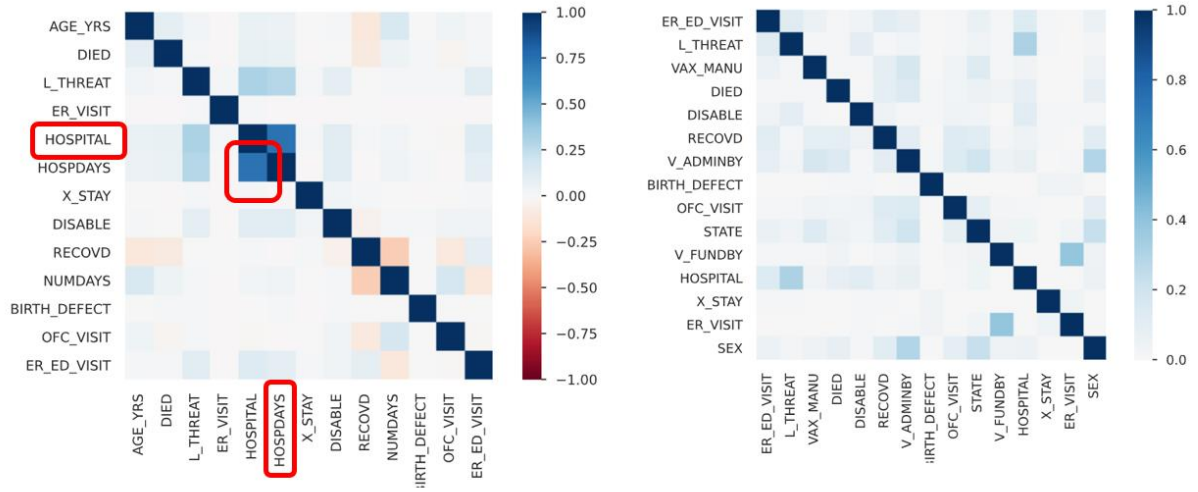
Data Preparation – Combined View

```
data_comb_2.head(10)
```

	STATE	AGE_YRS	SEX	DIED	L_THREAT	ER_VISIT	HOSPITAL	HOSPDAYS	X_STAY	DISABLE	RECOVD	NUMDAYS	V_ADMINBY	V_FUNDBY	BIRTH_DE
0	TX	33.0	F	0	0	0	0	0.0	0	0	1	2.0	PVT	0	
1	CA	73.0	F	0	0	0	0	0.0	0	0	1	0.0	SEN	0	
2	WA	23.0	F	0	0	0	0	0.0	0	0	2	0.0	SEN	0	
3	WA	58.0	F	0	0	0	0	0.0	0	0	1	0.0	WRK	0	
4	TX	47.0	F	0	0	0	0	0.0	0	0	0	7.0	PUB	0	
6	NV	44.0	F	0	0	0	0	0.0	0	0	1	0.0	PVT	0	
7	KS	50.0	M	0	0	0	0	0.0	0	0	1	1.0	PUB	0	
8	OH	33.0	M	0	0	0	0	0.0	0	0	0	2.0	OTH	0	
9	TN	71.0	F	0	0	0	0	0.0	0	0	0	8.0	PUB	0	
10	VA	18.0	F	0	0	0	0	0.0	0	0	0	1.0	PVT	0	

```
## Replacing Y with 1

data_comb_2['DIED'] = data_comb_2['DIED'].str.replace('Y', '1')
data_comb_2['L_THREAT'] = data_comb_2['L_THREAT'].str.replace('Y', '1')
data_comb_2['ER_VISIT'] = data_comb_2['ER_VISIT'].str.replace('Y', '1')
data_comb_2['HOSPITAL'] = data_comb_2['HOSPITAL'].str.replace('Y', '1')
data_comb_2['X_STAY'] = data_comb_2['X_STAY'].str.replace('Y', '1')
data_comb_2['DISABLE'] = data_comb_2['DISABLE'].str.replace('Y', '1')
data_comb_2['RECOVD'] = data_comb_2['RECOVD'].str.replace('Y', '1')
data_comb_2['RECOVD'] = data_comb_2['RECOVD'].str.replace('N', '0')
data_comb_2['RECOVD'] = data_comb_2['RECOVD'].str.replace('U', '2')
data_comb_2['BIRTH_DEFECT'] = data_comb_2['BIRTH_DEFECT'].str.replace('Y', '1')
data_comb_2['ER_ED_VISIT'] = data_comb_2['ER_ED_VISIT'].str.replace('Y', '1')
data_comb_2['OFC_VISIT'] = data_comb_2['OFC_VISIT'].str.replace('Y', '1')
data_comb_2['BIRTH_DEFECT'] = data_comb_2['BIRTH_DEFECT'].str.replace('Y', '1')
```

Data Preparation – Filtering, Imputing, Treating



EDA - Cross Correlation (Spearman's & Cramer's V(for Categorical))

```
dummydf = pd.DataFrame()
for i in data.columns[obj]:
    print(i)
    dummy = pd.get_dummies(data[i], drop_first=True)
    dummydf = pd.concat([dummydf, dummy], axis=1)
```

```
STATE
SEX
V_FUNDBY
VAX_MANU
```

```
dummydf.head(10)
```

	AK	AL	AR	AS	AZ	CA	CO	CT	Ca	DC	...	XV	M	U	0	OTH	PUB	PVT	UNK	MODERNA	PFIZER\BIONTECH
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	1	0	0	0	0	1	0
1	0	0	0	0	0	1	0	0	0	0	...	0	0	0	1	0	0	0	0	1	0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	1	0	0	0	0	0	1
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	1	0	0	0	0	1	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	1	0	0	0	0	1	0
5	0	0	0	0	0	0	0	0	0	0	...	0	0	0	1	0	0	0	0	1	0
6	0	0	0	0	0	0	0	0	0	0	...	0	1	0	1	0	0	0	0	1	0
7	0	0	0	0	0	0	0	0	0	0	...	0	1	0	1	0	0	0	0	1	0
8	0	0	0	0	0	0	0	0	0	0	...	0	0	0	1	0	0	0	0	1	0

Converted the Categorical variables into dichotomous variables based on their labels, and then included them in the Final Dataset and removed the original categorical variables.

```
data_1.head(5)
```

	Unnamed: 0	AGE_YRS	DIED	L_THREAT	ER_VISIT	HOSPITAL	HOSPDAYS	X_STAY	DISABLE	RECOVD	...	XV	M	U	0	OTH	PUB	PVT	UNK	MODERN
0	0	33.0	0	0	0	0	0.0	0	0	1	...	0	0	0	1	0	0	0	0	
1	1	73.0	0	0	0	0	0.0	0	0	1	...	0	0	0	1	0	0	0	0	
2	2	23.0	0	0	0	0	0.0	0	0	2	...	0	0	0	1	0	0	0	0	
3	3	58.0	0	0	0	0	0.0	0	0	1	...	0	0	0	1	0	0	0	0	
4	4	47.0	0	0	0	0	0.0	0	0	0	...	0	0	0	1	0	0	0	0	

Data Transformation - Creating dummy Variables for Categorical variables

```
Information value of Unnamed: 0 is 0.101826
Information value of AGE_YRS is 1.432297
Information value of L_THREAT is 0.01255
Information value of ER_VISIT is inf
Information value of HOSPITAL is 0.208083

C:\ProgramData\Anaconda3\lib\site-packages\ipykerne

Information value of HOSPDAYS is 0.0
Information value of X_STAY is 0.001383
Information value of DISABLE is 0.001957
Information value of RECOVD is 3.143333
Information value of NUMDAYS is 0.55676
Information value of BIRTH_DEFECT is 0.000212
Information value of OFC_VISIT is 0.07528
Information value of ER_ED_VISIT is 0.048825
Information value of AK is 0.000962
Information value of AL is 0.000266
Information value of AR is 0.003041

C:\ProgramData\Anaconda3\lib\site-packages\ipykerne

Information value of AS is inf
Information value of AZ is 0.003898
Information value of CA is 0.017771
Information value of CO is 0.002519
Information value of CT is 0.000711
Information value of Ca is inf
Information value of DC is 0.000142
Information value of DE is 0.000599
```

```
def iv_woe(data, target, bins=10, show_woe=False):

    #Empty Dataframe
    newDF = pd.DataFrame()

    #Extract Column Names
    cols = data.columns

    #Run WOE and IV on all the independent variables
    for ivars in cols[~cols.isin([target])]:
        if (data[ivars].dtype.kind in 'bifc') and (len(np.unique(data[ivars]))>10):
            binned_x = pd.qcut(data[ivars], bins,  duplicates='drop')
            d0 = pd.DataFrame({'x': binned_x, 'y': data[target]})
        else:
            d0 = pd.DataFrame({'x': data[ivars], 'y': data[target]})
        d = d0.groupby("x", as_index=False).agg({"y": ["count", "sum"]})
        d.columns = ['Cutoff', 'N', 'Events']
        d['% of Events'] = d['Events'] / d['Events'].sum()
        d['Non-Events'] = d['N'] - d['Events']
        d['% of Non-Events'] = d['Non-Events'] / d['Non-Events'].sum()
        d.loc[d['% of Non-Events'] == 0.0,'% of Non-Events'] = 1e-312
        d['WoE'] = np.log(d['% of Events']/d['% of Non-Events'])
        d['IV'] = d['WoE'] * (d['% of Events'] - d['% of Non-Events'])
        print("Information value of " + ivars + " is " + str(round(d['IV'].sum(),6)))
        temp =pd.DataFrame({"Variable" : [ivars],
                            "IV" : [d['IV'].sum()]}, columns = ["Variable", "IV"])
        newDF=pd.concat([newDF,temp], axis=0)

        #Show WOE Table
        if show_woe == True:
            print(d)
```

Feature Selection – Using Information Value

```
def WoE_transform(data, target, bins=10, show_woe=False):

    #Empty Dataframe
    newDF = pd.DataFrame()
    newDF = pd.concat([newDF,data], axis=1)

    #Extract Column Names
    cols = data.columns

    #Run WOE on all the independent variables
    for ivars in cols[~cols.isin([target])]:

        if (data[ivars].dtype.kind in 'bifc') and (len(np.unique(data[ivars]))>10):
            binned_x = pd.qcut(data[ivars], bins,  duplicates='drop')
            d0 = pd.DataFrame({'x': binned_x, 'y': data[target]})

            d = d0.groupby("x", as_index=False).agg({"y": ["count", "sum"]})
            d.columns = ['Cutoff', 'N', 'Events']
            d['% of Events'] = d['Events'] / d['Events'].sum()
            d['Non-Events'] = d['N'] - d['Events']
            d['% of Non-Events'] = d['Non-Events'] / d['Non-Events'].sum()
            d.loc[d['% of Non-Events'] == 0.0,'% of Non-Events'] = 1e-312
            d['WoE'] = np.log(d['% of Events']/d['% of Non-Events'])

            for i in range(d.shape[0]):
                interval = d.iloc[i]['Cutoff']
                left = interval.left
                right = interval.right
                IV_value = d.iloc[i]['WoE']
```

Feature Selection – Using WOE (Weight of Evidence) for taking care of Class Imbalance

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import KFold
from sklearn.metrics import roc_auc_score
c = [0.001,0.01,0.1,1.0,10.0,100.0]
num_cv_splits = 5
kf = KFold(n_splits=num_cv_splits, random_state=5)
for C in c:
    auc = 0.0
    acc = 0.0
    for train_index, test_index in kf.split(X_train):
        X_train_cv, X_test_cv = X_train_z[train_index], X_train_z[test_index]
        Y_train_cv, Y_test_cv = Y_train.iloc[train_index], Y_train.iloc[test_index]
        clf = LogisticRegression(C=C, random_state=5)
        clf.fit(X_train_cv,Y_train_cv)
        acc += clf.score(X_test_cv,Y_test_cv)
        pred = clf.predict_proba(X_test_cv)[:,1]
        auc += roc_auc_score(y_true = Y_test_cv, y_score = pred)
    print('C =',C,'; mean accuracy =',acc/num_cv_splits,'; auc =',auc/num_cv_splits)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:433: FutureWarning: Default solver will be changed
to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:433: FutureWarning: Default solver will be changed
to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:433: FutureWarning: Default solver will be changed
to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:433: FutureWarning: Default solver will be changed
to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:433: FutureWarning: Default solver will be changed
to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)

C = 0.001 ; mean accuracy = 0.9885336632817306 ; auc = 0.9279684038505083
```

Modelling – Logistic Regression

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import KFold
from sklearn.metrics import roc_auc_score,roc_curve,confusion_matrix
depths = [3,4,5,6,7,8,9,10,11]
```

```
num_cv_splits = 5
kf = KFold(n_splits=num_cv_splits, random_state=5)
for depth in depths:
    auc = 0.0
    acc = 0.0
    for train_index, test_index in kf.split(X_train):
        X_train_cv, X_test_cv = X_train.iloc[train_index], X_train.iloc[test_index]
        Y_train_cv, Y_test_cv = Y_train.iloc[train_index], Y_train.iloc[test_index]
        clf = DecisionTreeClassifier(max_depth=depth, random_state=5)
        clf.fit(X_train_cv,Y_train_cv)
        acc += clf.score(X_test_cv,Y_test_cv)
        pred = clf.predict_proba(X_test_cv)[:,1]
        auc += roc_auc_score(y_true = Y_test_cv, y_score = pred)
    print('depth =', depth, '; mean accuracy =', acc/num_cv_splits, '; auc =', auc/num_cv_splits)
```

```
depth = 3 ; mean accuracy = 0.9885128685053786 ; auc = 0.9050642135743112
depth = 4 ; mean accuracy = 0.9886334790730846 ; auc = 0.9183425705942225
depth = 5 ; mean accuracy = 0.988612683950787 ; auc = 0.9237781202237862
depth = 6 ; mean accuracy = 0.9886542737629505 ; auc = 0.9316058253559012
depth = 7 ; mean accuracy = 0.988766566766061 ; auc = 0.9365318202728716
depth = 8 ; mean accuracy = 0.9887748858009242 ; auc = 0.9343187515864082
depth = 9 ; mean accuracy = 0.9885877312570006 ; auc = 0.9317484874888347
depth = 10 ; mean accuracy = 0.9883880989824021 ; auc = 0.9276869168125094
depth = 11 ; mean accuracy = 0.9882841231114552 ; auc = 0.9240789880794212
```

Modelling – Decision Tree

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import KFold
from sklearn.metrics import roc_auc_score
num_trees = [25,50,75,100,125,150,175,200,225,250]
max_depth = [3,4,5,6]
num_cv_splits = 5
```

```
kf = KFold(n_splits=num_cv_splits, random_state=5)
for tree in num_trees:
    for depth in max_depth:
        auc = 0.0
        acc = 0.0
        for train_index, test_index in kf.split(X_train):
            X_train_cv, X_test_cv = X_train.iloc[train_index], X_train.iloc[test_index]
            Y_train_cv, Y_test_cv = Y_train.iloc[train_index], Y_train.iloc[test_index]
            clf = RandomForestClassifier(n_estimators=tree, max_depth=depth, n_jobs = 8, random_state=5)
            clf.fit(X_train_cv,Y_train_cv)
            acc += clf.score(X_test_cv,Y_test_cv)
            pred = clf.predict_proba(X_test_cv)[:,1]
            auc += roc_auc_score(y_true = Y_test_cv, y_score = pred)
        print('num_trees =',tree,'; depth=',depth,'; mean accuracy =',acc/num_cv_splits,'; auc =',auc/num_cv_splits)
```

```
num_trees = 25 ; depth= 3 ; mean accuracy = 0.9885128685053786 ; auc = 0.9003092333118206
num_trees = 25 ; depth= 4 ; mean accuracy = 0.9885128685053786 ; auc = 0.9129976555031017
num_trees = 25 ; depth= 5 ; mean accuracy = 0.9885128685053786 ; auc = 0.916388842114842
num_trees = 25 ; depth= 6 ; mean accuracy = 0.9885170274606491 ; auc = 0.925553235910941
num_trees = 50 ; depth= 3 ; mean accuracy = 0.9885128685053786 ; auc = 0.9078953513786079
num_trees = 50 ; depth= 4 ; mean accuracy = 0.9885128685053786 ; auc = 0.9202721369070523
num_trees = 50 ; depth= 5 ; mean accuracy = 0.9885128685053786 ; auc = 0.9219877873539495
num_trees = 50 ; depth= 6 ; mean accuracy = 0.9885128685053786 ; auc = 0.9280767967707433
num_trees = 75 ; depth= 3 ; mean accuracy = 0.9885128685053786 ; auc = 0.9079991607256641
num_trees = 75 ; depth= 4 ; mean accuracy = 0.9885128685053786 ; auc = 0.9207668806967
num_trees = 75 ; depth= 5 ; mean accuracy = 0.9885128685053786 ; auc = 0.9245166676216078
num_trees = 75 ; depth= 6 ; mean accuracy = 0.9885128685053786 ; auc = 0.9282739187143786
num_trees = 100 ; depth= 3 ; mean accuracy = 0.9885128685053786 ; auc = 0.908636906549918
num_trees = 100 ; depth= 4 ; mean accuracy = 0.9885128685053786 ; auc = 0.9209350523189027
```

Modelling – Random Forest

```
from keras.models import Sequential, Model
from keras.layers import Dense, GRU, Dropout, Flatten, TimeDistributed
from keras.callbacks import ModelCheckpoint, ReduceLROnPlateau
from keras import optimizers
from keras.applications.vgg16 import VGG16

base_model = VGG16(include_top=False, weights='imagenet', input_shape=(120,120,3))
x = base_model.output
x = Flatten()(x)
#x.add(Dropout(0.5))
features = Dense(64, activation='relu')(x)
conv_model = Model(inputs=base_model.input, outputs=features)

for layer in base_model.layers:
    layer.trainable = False

model = Sequential()
model.add(TimeDistributed(conv_model, input_shape=(15,120,120,3)))
model.add(GRU(32, return_sequences=True))
model.add(GRU(16))
model.add(Dropout(0.5))
model.add(Dense(8, activation='relu'))
model.add(Dense(5, activation='softmax'))
```

```
model.fit_generator(train_generator, steps_per_epoch=steps_per_epoch, epochs=num_epochs, verbose=1,
                    callbacks=callbacks_list, validation_data=val_generator,
                    validation_steps=validation_steps, class_weight=None, workers=1, initial_epoch=0)
Epoch 25/70
42/42 [==============================] - 43s 1s/step - loss: 0.8494 - categorical_accuracy: 0.7529 - val_loss: 1.1134 - val_c
ategorical_accuracy: 0.6100

Epoch 00025: saving model to model_init_conv_lstm_2018-10-0414_07_55.144483/model-00025-0.85085-0.74962-1.11337-0.61000.h5
Epoch 26/70
42/42 [==============================] - 43s 1s/step - loss: 0.8199 - categorical_accuracy: 0.7581 - val_loss: 1.0979 - val_c
ategorical_accuracy: 0.6100

Epoch 00026: saving model to model_init_conv_lstm_2018-10-0414_07_55.144483/model-00026-0.81817-0.75867-1.09791-0.61000.h5

Epoch 00026: ReduceLROnPlateau reducing learning rate to 0.0002500000118743628.
Epoch 27/70
42/42 [==============================] - 43s 1s/step - loss: 0.8008 - categorical_accuracy: 0.7838 - val_loss: 1.0921 - val_c
ategorical_accuracy: 0.6100
```

Modelling – LSTM

Algorithms	Accuracy	AUC	Parameters
Logistic	98.8	94.0	Using KFold with 5 Splits (using c = 0.001, 0.01, 0.1,1,10,100)
Decision Tree	98.8	92.4	Using KFold with Depth = 11
Random Forest	98.8	91.7	Trees = 250, Depth = 3,4,5,6
LSTM	98.8	95.2	Epochs = 120, Batch = 64, Drop out = 0.5

Model Evaluation

Research Questions (RQs)

1. The COVID-19 vaccine: what is the dominant opinion?
2. What do people think about vaccine manufacturers?
3. What are the aftereffects of vaccine and their impact on human populations?
4. What are the adverse effects/symptoms based on the three different vaccines?
5. Out of the 3 Manufacturers identify which of them has higher negative symptoms?
6. What is the correlation across variables causing mortality rate after vaccine?

Conclusion and Future Work

To be added.

References:

Ahamad MM, Aktar S, Uddin MJ, Rashed-Al-Mahfuz M, Azad AKM, Uddin S, Alyami SA, Sarker IH, Liò P, Quinn JMW and Moni MA (2021) 'Adverse effects of COVID-19 vaccination: machine learning and statistical approach to identify and classify incidences of morbidity and post-vaccination reactogenicity', *medRxiv*:2021.2004.2016.21255618, https://doi.org/10.1101/2021.04.16.21255618

Almufty HB, Mohammed SA, Abdullah AM and Merza MA (2021) 'Potential adverse effects of COVID19 vaccines among Iraqi population; a comparison between the three available vaccines in Iraq; a retrospective cross-sectional study', *Diabetes & Metabolic Syndrome: Clinical Research & Reviews*, 15(5):102207, https://doi.org/https://doi.org/10.1016/j.dsx.2021.102207

Bahl K, Senn JJ, Yuzhakov O, Bulychev A, Brito LA, Hassett KJ, Laska ME, Smith M, Almarsson Ö, Thompson J, Ribeiro A, Watson M, Zaks T and Ciaramella G (2017) 'Preclinical and Clinical Demonstration of Immunogenicity by mRNA Vaccines against H10N8 and H7N9 Influenza Viruses', *Molecular Therapy*, 25(6):1316-1327, https://doi.org/https://doi.org/10.1016/j.ymthe.2017.03.035

Blalock HM, Jr. (1963) 'Correlated Independent Variables: The Problem of Multicollinearity', *Social Forces*, 42(2):233-237, https://doi.org/10.1093/sf/42.2.233

Blumenthal KG, Robinson LB, Camargo CA, Jr, Shenoy ES, Banerji A, Landman AB and Wickner P (2021) 'Acute Allergic Reactions to mRNA COVID-19 Vaccines', *JAMA*, 325(15):1562-1565, https://doi.org/10.1001/jama.2021.3976

Bouktif S, Fiaz A, Ouni A and Serhani MA (2018) 'Optimal Deep Learning LSTM Model for Electric Load Forecasting using Feature Selection and Genetic Algorithm: Comparison with Machine Learning Approaches †', *Energies*, 11(7):1636. https://www.mdpi.com/1996-1073/11/7/1636

Brooks NA, Puri A, Garg S, Nag S, Corbo J, Turabi AE, Kaka N, Zemmel RW, Hegarty PK and Kamat AM (2021) 'The association of Coronavirus Disease-19 mortality and prior bacille Calmette-Guerin vaccination: a robust ecological analysis using unsupervised machine learning', *Scientific Reports*, 11(1):774, https://doi.org/10.1038/s41598-020-80787-z

Challita U, Ferdowsi A, Chen M and Saad W (2019) 'Machine Learning for Wireless Connectivity and Security of Cellular-Connected UAVs', *IEEE Wireless Communications*, 26(1):28-35, https://doi.org/10.1109/MWC.2018.1800155

Chen J, Gao K, Wang R and Wei G-W (2021) 'Prediction and mitigation of mutation threats to COVID-19 vaccines and antibody therapies', *Chemical science*, 12(20):6929-6948.

Dong E, Du H and Gardner L (2020) 'An interactive web-based dashboard to track COVID-19 in real time', *The Lancet Infectious Diseases*, 20(5):533-534, https://doi.org/https://doi.org/10.1016/S1473-3099(20)30120-1

Harrison EA and Wu JW (2020) 'Vaccine confidence in the time of COVID-19', *European Journal of Epidemiology*, 35(4):325-330, https://doi.org/10.1007/s10654-020-00634-3

Hatmal MmM, Al-Hatamleh MAI, Olaimat AN, Hatmal M, Alhaj-Qasem DM, Olaimat TM and Mohamud R (2021) 'Side Effects and Perceptions Following COVID-19 Vaccination in Jordan: A Randomized, Cross-Sectional Study Implementing Machine Learning for Predicting Severity of Side Effects', *Vaccines*, 9(6), https://doi.org/10.3390/vaccines9060556

Imran SA, Islam MT, Shahnaz C, Islam MT, Imam OT and Haque M (26-27 Dec. 2020 2020) 'COVID-19 mRNA Vaccine Degradation Prediction using Regularized LSTM Model', in *2020 IEEE International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE)*, accessed. https://doi.org/10.1109/WIECON-ECE52138.2020.9398044

Kaur RJ, Dutta S, Bhardwaj P, Charan J, Dhingra S, Mitra P, Singh K, Yadav D, Sharma P and Misra S (2021) 'Adverse Events Reported From COVID-19 Vaccine Trials: A Systematic Review', *Indian Journal of Clinical Biochemistry*, https://doi.org/10.1007/s12291-021-00968-z

Kaur SP and Gupta V (2020) 'COVID-19 Vaccine: A comprehensive status report', *Virus Research*, 288:198114, https://doi.org/https://doi.org/10.1016/j.virusres.2020.198114

Lee EK, Nakaya HI, Yuan F, Querec TD, Burel G, Pietz FH, Benecke BA and Pulendran B (2016) 'Machine Learning for Predicting Vaccine Immunogenicity', *INFORMS Journal on Applied Analytics*, 46(5):368-390, https://doi.org/10.1287/inte.2016.0862

Li Q and Lu H (2020) 'Latest updates on COVID-19 vaccines', *BioScience Trends*, 14(6):463-466, https://doi.org/10.5582/bst.2020.03445

Meyer M, Huang E, Yuzhakov O, Ramanathan P, Ciaramella G and Bukreyev A (2017) 'Modified mRNA-Based Vaccines Elicit Robust Immune Responses and Protect Guinea Pigs From Ebola Virus Disease', *The Journal of Infectious Diseases*, 217(3):451-455, https://doi.org/10.1093/infdis/jix592

Na T, Cheng W, Li D, Lu W and Li H (2021) 'Insight from NLP Analysis: COVID-19 Vaccines Sentiments on Social Media', *arXiv preprint arXiv:2106.04081*.

Ong E, Wong MU, Huffman A and He Y (2020) 'COVID-19 Coronavirus Vaccine Design Using Reverse Vaccinology and Machine Learning', *Frontiers in Immunology*, 11(1581), https://doi.org/10.3389/fimmu.2020.01581

Shahid F, Zameer A and Muneeb M (2020) 'Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM', *Chaos, Solitons & Fractals*, 140:110212.

Tissot N, Brunel A-S, Bozon F, Rosolen B, Chirouze C and Bouiller K (2021) 'Patients with history of covid-19 had more side effects after the first dose of covid-19 vaccine', *Vaccine*, 39(36):5087-5090, https://doi.org/https://doi.org/10.1016/j.vaccine.2021.07.047

Tobaiqy M, Elkout H and MacLure K (2021) 'Analysis of Thrombotic Adverse Reactions of COVID-19 AstraZeneca Vaccine Reported to EudraVigilance Database', *Vaccines*, 9(4):393. https://www.mdpi.com/2076-393X/9/4/393

Wedlund L and Kvedar J (2021) 'New machine learning model predicts who may benefit most from COVID-19 vaccination', *npj Digital Medicine*, 4(1):59, https://doi.org/10.1038/s41746-021-00425-4