

# Surakhsha Kavach : ML based Cross Platform Application for Covid-19 Vulnerability Detection

*Ms.JasmineKaurWadhwa<sup>1</sup>, Ms.SrushtiPatil<sup>2</sup>, Ms.RuchiRaicha<sup>3</sup>, Ms.YamineePatil<sup>4</sup>  
Ms.SonalJain<sup>5</sup>*

Department of Information Technology  
A.P Shah Institute of Technology  
Thane, India

*jasminekaurwadhwa@apsit.edu.in  
srushtipatil@apsit.edu.in  
ruchiraicha@apsit.edu.in  
ympatil@apsit.edu.in  
Sajain@apsit.edu.in*

**Abstract.** Surakhsha Kavach app is the way of detecting vulnerability of covid attack of person. As people were not aware at an early stage so many people were suffering at a large number leading to serious consequences. So The idea of ML based app is to make people aware regarding their vulnerabilities of covid attack which will help people to take precautions at an early stage to avoid further serious consequences and protect themselves and their family. Overall this app will be easily available on any platform which will help people to easily access and take the benefit of it. We have arrived at a conclusion that ML based Surakhsha Kavach App is a much viable solution for the people to take precautionary measures at an early stage.

**Keywords:** Covid-19, Surakhsha Kavach, Machine learning

## 1. Introduction

On December 31, 2019, a novel pathogenic Coronavirus (2019- nCoV) epidemic was first discovered in Wuhan, Hubei Province, South China. On March 11th, 2020, the World Health Organization declared it a Pandemic. COVID-19 is the official name for the Coronavirus illness.

COVID-19 is an infectious disease caused by the Coronavirus 2(SARSCoV2) that causes severe acute respiratory syndrome. World Health Organization (WHO) declared the outbreak a Public Health Emergency of International Concern.

As this covid is becoming a concern for all of us and its spreading very fast. Its necessary for us to take precautions as early as possible.

So Surakhsha kavach app is cross platform application where user can get the vulnerability attack of covid by entering his age and state. After he enters his age and state he gets the predicted vulnerability attack of covid.

This prediction is done using various Machine Learning algorithms like Logistic Regression, Random Forest, Naïve Bayes and Support Vector Machine(SVM). According to the age and state the data is first fitted into training and testing, after getting probability its then classify into 3 classes:-

0 - very less vulnerable

1- vulnerable

2- more vulnerable

For Example: If a user enters age as 60 and state as maharashtra so his chances of getting covid is more as his age is more than 50 and also the covid cases are more so in output there will be message like "You are more vulnerable".

This will help people especially the senior citizens to take precautions at a very early stage and protect themselves from getting covid and severe diseases. The organisation of this paper from hereafter is as follows: Section 2 contains the literature survey done and Section 3 presents the analysis of the project. Section 4 describes the existing System architecture, methods and the solution to that is shown in the proposed System architecture in Section 5. Section 6 contains Futurescope and Section 7 contains Conclusion.

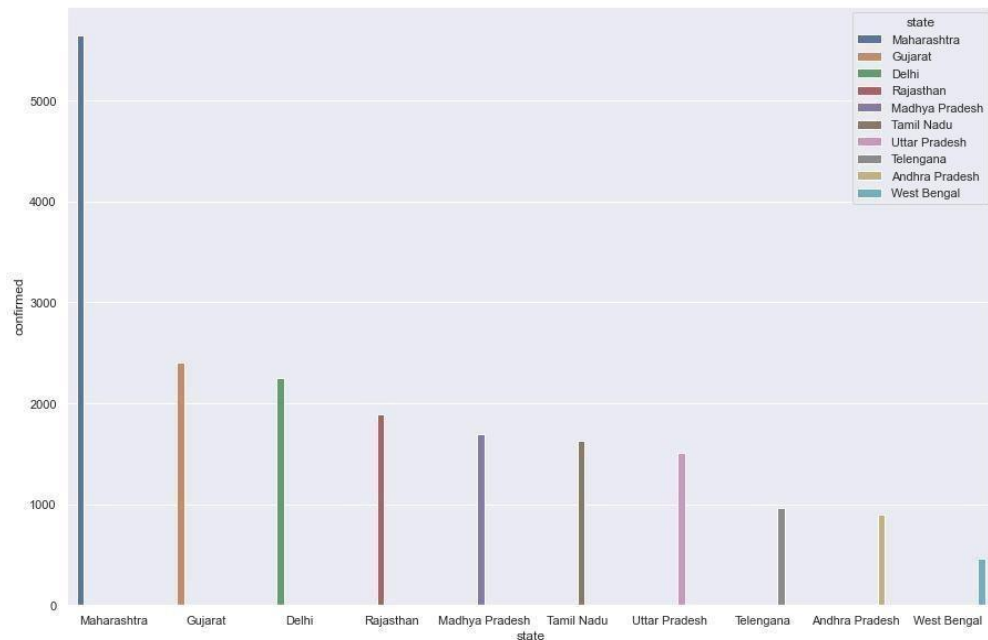
## 2. Literature Survey

1. In the first literature the author has discussed the regression analysis of Covid19 using various machine learning algorithms like polynomial regression, Linear Regression etc [1].
2. In the Second Literature the author has discussed the approach to predict corona virus using various machine Learning algorithms like Support Vector Machine(SVM),K-NN Algorithm,Decision Tree, Random Forest [2].
3. In the Third Literature the author has discussed regarding detection of covid-19 from medical images/Symptoms of Patients using machine learning.This has been carried out using deep learning algorithms [3].

## 3. Analysis

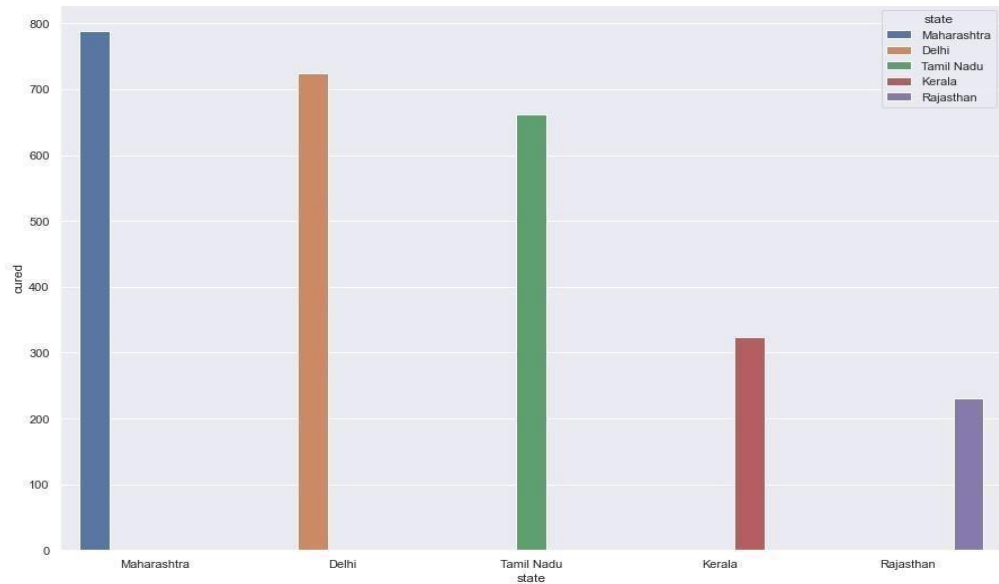
Before going towards existing system architecture we have done the analysis regarding covid cases state-wise.

So we have taken the dataset India\_Covid.csv has been pre-processed which is taken from Kaggle and then we visualized state-wise covid cases as shown below.



**Fig .1** State-wise Confirmed Cases

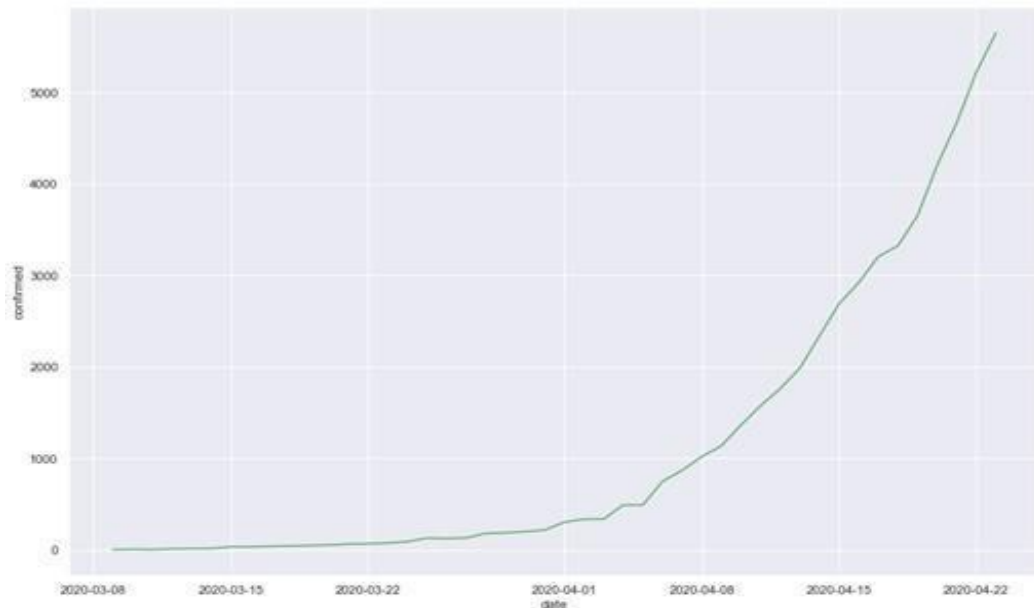
Above Fig 1 shows Maharashtra has the highest confirmed state and West Bengal has the lowest confirmed state.



**Fig.2.** Statewise Cured Cases

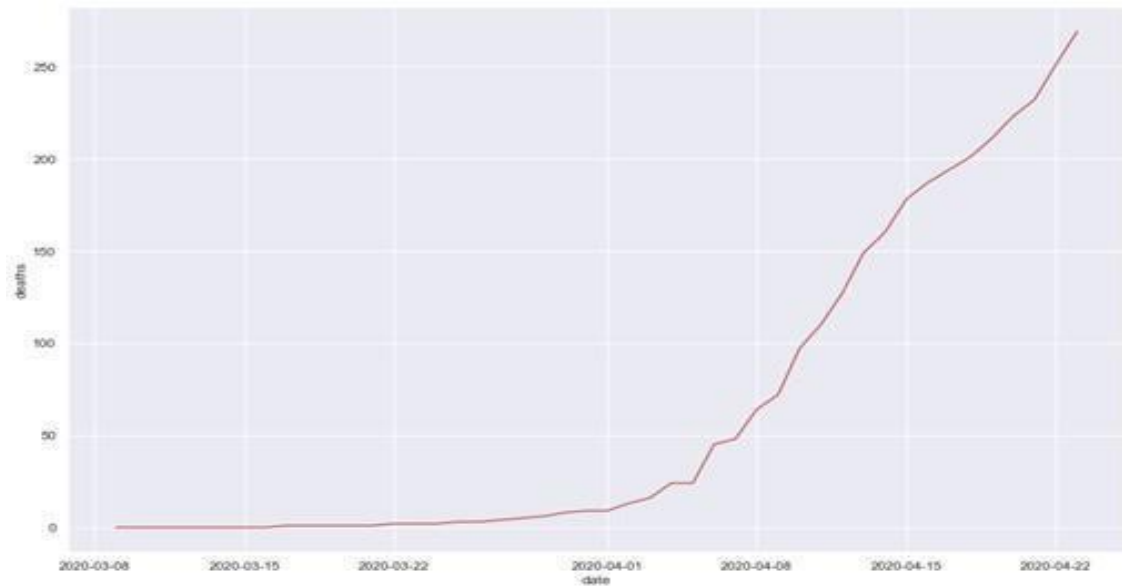
Above Fig.2 shows statewise cured cases in which Maharashtra is the highest cured state and Rajasthan is the lowest cured state. So now we have done analysis of covid cases and death individually of state.

### 1. Maharashtra:



**Fig.3.** Datewise Confirmed Cases in Maharashtra

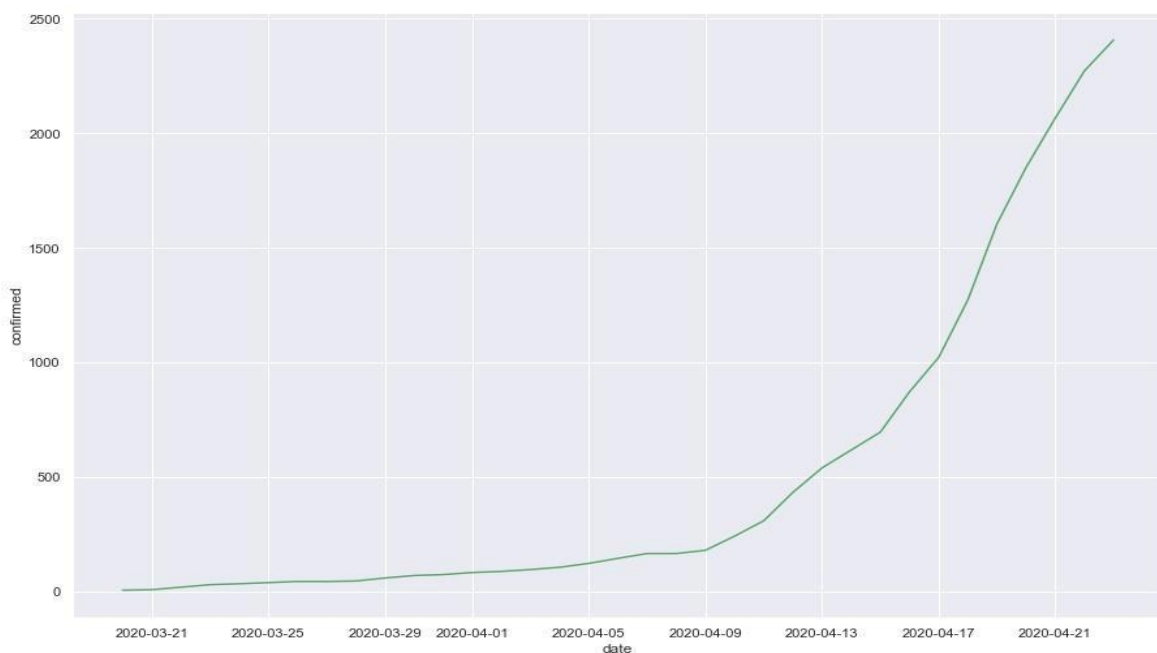
The Above fig shows datewise Confirmed Cases in Maharashtra from 8<sup>th</sup> march 2020 to 22 April 2020.



**Fig.4.** Datewise Death Cases in Maharashtra.

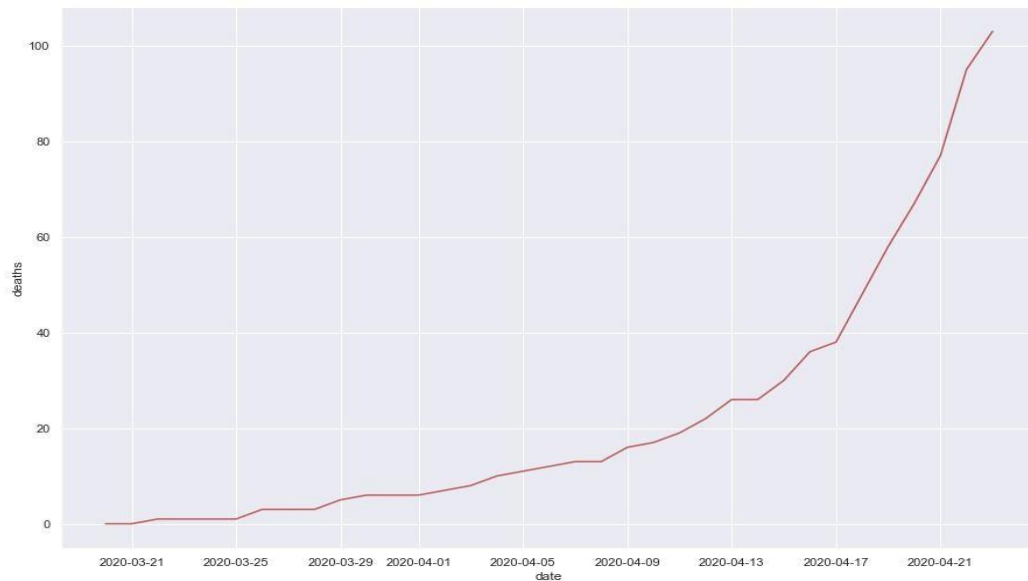
The Above fig shows Datewise Death Cases in Maharashtra from 8<sup>th</sup> march 2020 to 22 April 2020. So we can see that there is an increase in death rate as covid cases are increasing.

## 2. Gujarat:



**Fig 5.** Datewise Confirmed Cases in Gujarat

The Above fig shows datewise Confirmed Cases in Gujarat from 8<sup>th</sup> march 2020 to 22 April 2020.

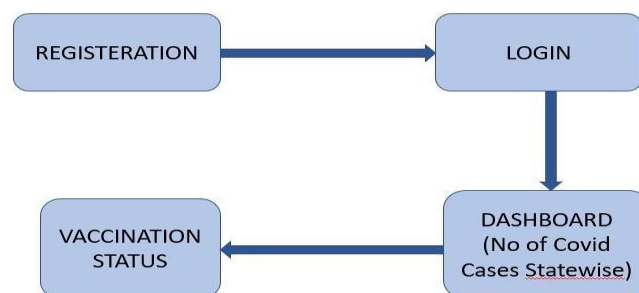


**Fig 6.** Datewise Death Cases in Gujarat

The Above fig shows Datewise Death Cases in Gujarat from 8<sup>th</sup> march 2020 to 22 April 2020. So we can see that there is an increase in death rate as covid cases are increasing.

#### 4. Existing System Architecture

There are many Softwares available which gives information regarding covid cases and vaccination status. People may know that how many cases are around them with the help of tools. They may get the alert notification if there are too many covid cases around them So that a person can take a precautions. However there is no information regarding the percentage attack that one may have of covid which leads people to serious consequences. The Existing System includes-



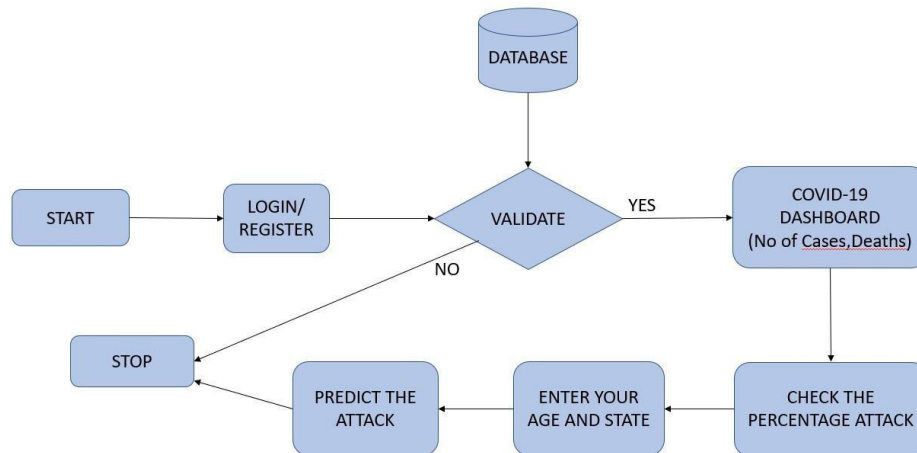
**Fig .7.** Existing System Workflow.

- 1) The Registration
- 2) Login
- 3) User can see covid cases statewide
- 4) User can also see the information regarding vaccination
- 5) But there is no information regarding the Percentage of covid attack that one may have as a precautionary measure.

This project aims to solve the limitations of the existing systems and further develop a new system.

## 5. Proposed System Architecture

Figure 2 Depicts the process of the proposed system. The proposed system is explained below.



**Fig 8.** Proposed System Workflow

### 1) login/Register:

It consist of username and password. this combination is all-inclusive authentication methods used because of convenience to others and low cost of deployment...Once user is login in...he/she can take experience of app.

### 2) Covid-19 Dashboard:

Once the user is successfully logged in....he/she can see the daily covid cases.

### 3) Check the vulnerability attack:

The user can click on this button and can check his/her vulnerability attack of covid

### 4) Predict the attack:

After clicking on the vulnerability attack button, enter his/her age and state and then he/she can get the vulnerability attack of covid he may have so that he can take precautionary measures at an early stage to prevent themselves from a serious diseases. The prediction is done based on ML algorithms. Algorithms used are:

#### 1) Random Forest :

Random Forest is a popular machine learning algorithm that is part of a supervised machine learning strategy. It can be used for both classification and regression problems in ML. A fragmented system which contains decision trees for different sets of data provided and takes the measure to improve accuracy of the data speculation is known as Random Forest. This algorithm takes less training time compared to other algorithms.

It also predicts the output with high accuracy. In large databases it works well.

We applied this algorithm on our dataset so the accuracy is 54.44%

```

In [40]: 1 from sklearn.ensemble import RandomForestClassifier
          2
          3 clf_4 = RandomForestClassifier()
          4 clf_4.fit(x_train, y_train)
          5 pred_y_4 = clf_4.predict(x_test)
          6

In [43]: 1 print(accuracy_score(y_test, pred_y_4))
          2 score_svm = round(accuracy_score(pred_y_4,y_test)*100,2)

0.5444118878191712

In [44]: 1 print("The accuracy score achieved using Random Forest is: "+str(score_svm)+" %")

The accuracy score achieved using Random Forest is: 54.44 %

In [ ]: 1

```

**Fig 9.** Random Forest Algorithm Accuracy

## 2) Logistic Regression:

Logistic Regression is a machine learning algorithm that is a part of supervised machine learning algorithm. This algorithm speculates the output of a categorical dependent variable with the help of given set of independent variables. The outcome can be either Yes or No or 0,1 or true or false that is categorical or discrete values.

We applied this algorithm on our dataset so the accuracy is 49.43%.

```

In [30]: 1 from sklearn.linear_model import LogisticRegression#(Logistic Fuction)
          2
          3 lr = LogisticRegression()
          4
          5 lr.fit(x_train,y_train)
          6
          7 Y_pred_lr = lr.predict(x_test)
          8 score_lr = round(accuracy_score(Y_pred_lr,y_test)*100,2)
          9 score_lr

Out[30]: 49.43

In [31]: 1 lr_cm=confusion_matrix(y_test,Y_pred_lr)
          2 plt.title("LogisticRegression_cm")
          3 sns.heatmap(lr_cm,annot=True,cmap="Blues",fmt="d",cbar=False)

Out[31]: <AxesSubplot:title={'center':'LogisticRegression_cm'}>

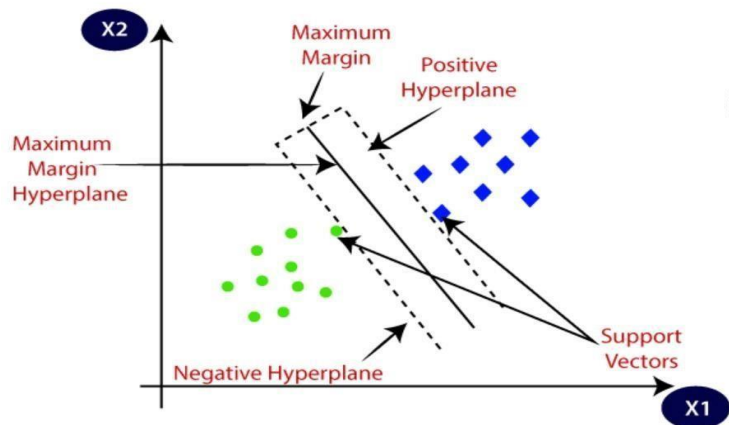
LogisticRegression_cm

```

**Fig 10.** Logistic Regression Accuracy

## 3) Support Vector Machine:

SVM is a machine learning algorithm that comes under supervised Machine Learning algorithm. This algorithm is very helpful in classification problems. In SVM algorithm it divides the datasets into classes so that we can find hyperplane. Hyperplane is a boundary that differentiates the two classes in SVM. The diagram below has 2 distinct categories which is separated by a decision boundary or hyperplane [4].



**Fig 11.** Support Vector Machine.

We applied this algorithm on our dataset so the accuracy is 54.11%.

```
In [34]: 1 from sklearn import svm
2
3 sv = svm.SVC(kernel='rbf')
4
5 sv.fit(x_train, y_train)
6
7 Y_pred_svm = sv.predict(x_test)
8 score_svm = round(accuracy_score(Y_pred_svm, y_test)*100,2)
9
10 print("The accuracy score achieved using RBF SVM is: "+str(score_svm)+" %")

The accuracy score achieved using RBF SVM is: 54.11 %

In [35]: 1 sv_cm=confusion_matrix(y_test,Y_pred_svm)
2 plt.title("SVM_Classification_cm")
3 sns.heatmap(sv_cm,annot=True,cmap="Reds",fmt="d",cbar=False)

Out[35]: <AxesSubplot:title={'center':'SVM_Classification_cm'}>
```

SVM Classification cm

**Fig 12.** Support Vector Machine Accuracy

#### 4) Naive Bayes:

Naive Bayes algorithm is a machine learning algorithm that comes under supervised machine learning algorithm. For solving classification problem Naïve Bayes algorithm is used. This algorithm is based on bayes theorem and is supervised Machine Learning algorithm. The Naïve Bayes is one of the most effective and faster machine learning algorithm that help us to make faster predictions.

It is a system of possible classification, which means that it predicts based on the possibilities of an object.

The formula for Bayes' theorem is given as:

$$P(A|B)=P(B|A)P(A)/P(B)$$

Naïve Bayes Classifier Algorithm

Where,

$P(A|B)$  : Posterior probability

$P(B|A)$  : Likelihood probability

When this algorithm is applied on our dataset the accuracy is 53.66%.



```

In [32]: 1 from sklearn.naive_bayes import GaussianNB
          2
          3 nb = GaussianNB()
          4
          5 nb.fit(x_train,y_train)
          6
          7 Y_pred_nb = nb.predict(x_test)
          8 score_nb = round(accuracy_score(Y_pred_nb,y_test)*100,2)
          9
          10 print("The accuracy score achieved using Naive Bayes is: "+str(score_nb)+" %")

The accuracy score achieved using Naive Bayes is: 53.66 %

In [33]: 1 nb_cm=confusion_matrix(y_test,Y_pred_nb)
          2 plt.title("NaiveBayes_cm")
          3 sns.heatmap(nb_cm,annot=True,cmap="Oranges",fmt="d",cbar=False)

Out[33]: <AxesSubplot:title={'center':'NaiveBayes_cm'}>

```



**Fig 13.** Naïve Bayes Algorithm Accuracy

So overall this age and state data is first fitted into training and testing and then gives probability and then it gets classified into classes like how much the user is vulnerable and then the user gets output according to it.

## 6. Future Scope:

The Future Scope of our proposed system can be:

1. The idea about various languages can be implemented in the application, so that all people who don't understand English can easily access the application.
2. The idea about lockdown reminder is one of the features we can add so that people can know by the notifications on screen.
3. The idea of booking vaccination slots can also be implemented.

## 7. Conclusion:

In our project, the prediction of percentage attack is done using Machine Learning Algorithms. Random forest is suitable as its accuracy is more. We came up with this solution as people are getting infected by this covid at a very large number. So it is necessary to spread awareness and take precautions as soon as possible. This app will save the life of many people and help them to live happy and healthy life.

## References

- [1] E. Gambhir, R. Jain and A. Gupta, "Regression Analysis of COVID-19 using Machine," 2020.
- [2] R. . N. J. K. and J. , "A Comparative Approach To Predict Corona Virus," in *International Conference on Artificial Intelligence and Smart Systems (ICAIS-2021)*.
- [3] A. K. Siddhu, D. A. K. and . D. S. K. , "Detection of COVID19 from Medical Images and/ or Symptoms of Patient," in *International Conference on System Modeling & Advancement in Research Trends*.

[4]

[Online]. Available: <https://medium.com/analytics-vidhya/what-is-machine-learning-3977a3b08384>.